Nevoscopy:

Three-Dimensional Computed Tomography of Nevi and Melanomas in situ by Transillumination to Detect Early Cutaneous Malignant Melanomas

Ph.D. THESIS

by

ATAM PRAKASH DHAWAN

Department of Electrical Engineering
University of Manitoba
Winnipeg, Canada R3T 2N2

ISBN 0-315-48114-5

NEVOSCOPY: THREE-DIMENSIONAL COMPUTED TOMOGRAPHY OF NEVI AND MELANOMAS IN SITU BY TRANSILLUMINATION TO DETECT EARLY CUTANEOUS MALIGNANT MELANOMAS

BY

ATAM PRAKASH DHAWAN

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Atam P. Dhawan

Department of Electrical and Computer Engineering
University of Cincinnati, 812 Rhodes Hall,
Cincinnati, Ohio 45221 - 0030 , U.S.A.
Phone: 513 - 475 - 4386

July 1, 1988

Ms. Helen Agar
Faculty of Graduate Studies
University of Manitoba
500 University Center
Winnipeg, Manitoba
Canada R3T 2N2

Dear Ms. Agar:

This is regarding the copyrighted material of the Appendix-B of my Ph.D. thesis "Nevoscopy: Three-Dimensional Computed Tomography of Nevi and Melanoma in situ by Transillumination to Detect Early Cutaneous Malignant Melanoma". All the four papers whose copies have been enclosed in the Appendix-B are written by me. As the first author of these papers: "Nevoscopy: Three-Dimensional Computed Tomography of Nevi and Melanoma in situ by Transillumination", published in IEEE Trans. on Medical Imaging, June 1984; "Wiener Filtering For Deconvolution of Geometric Artifacts in Limited-View Image Reconstruction", published in the Proceedings 1984 I.S.M.I.I. Conference; "Image Restoration by Two-Dimensional Deconvolution in Limited-View Reconstruction", published in the Technical Digest, Topical Meeting in Industrial Applications of CT and NMR Imaging; and "Computed Tomography By Transillumination To Detect Early Melanoma", published in the Proceedings of the Sixth Annual Conference of IEEE EMBS Society, I hereby give my permission to the University of Manitoba to copy them as the Appendix-B of my thesis.

Please note a change in address as written above. Thank You.

Sincerely

Atam P. Dhawan

Dedicated To My Father

Chandar Bhan Dhawan

ACKNOWLEDGEMENT

I express my sincere thanks to the Canadian Commonwealth Scholarship and Fellowship Committee for giving me the opportunity to study in Canada and for supporting me as a Canadian Commonwealth Fellow for my Ph.D. degree. I feel myself fortunate to have Dr. Richard Gordon as my advisor. His invaluable advice, suggestions and encouragement during this work deserve much more appreciation than I can express in words. I would also like to thank to Drs. Rangaraj M. Rangayyan, Ed Shewdyk, James S. Henderson, Ross Brown and Waldemar H. Lehn for their advice from time to time. I appreciate the suggestions and help given by Dr. John Taylor, Dr. Frank Baldwin, Dr. Eileen Murray, Dr. Barry Anderson and Linda Nahnybida in understanding the medical aspects of the problem and in carrying out the histology. I would also like to express my appreciation to John Brexel for construction of the nevoscope, and John Dunning and the staff of the Computer Department for Health Sciences (CDHS) of the Faculty of Medicine for their help in image processing and computer simulation.

Last, but not least, a few words for those who patiently listened my problems and always encouraged me to complete this research work. First of all, I would like to thank my wife Nilam Dhawan without whom I would not have been able to do this work. I thank my father Chandar Bhan Dhawan and my mother Shanti Dhawan for inspiring me to study hard to do useful research. They encouraged me even on my minor

acheievements. I thank Dorothy Linklater, Jack Linklater and Leslie J. Gordon with a deep appreciation for the encouragement and help they have given me in last years. I also thank Dr. R.M. Mathur, Dr. E. Kuffel, and Dr. J. B. Sutherland for providing me funds to attend conferences.

CONTENTS

PAGE

LIST OF FIGURES

PAGE

ABSTRACT

Malignant melanoma is the most lethal skin cancer. It often develops from a pre-existing mole (nevus) or the pigment (tanning) cells of the upper layer of the skin. The mortality rate of malignant melanoma is increasing at a faster rate than any other cancer in North America except lung cancer in women. At the current pace 1 person out of 250 in the United States will develop malignant melanoma. But, nearly all melanomas can be cured if detected early and promptly removed surgically. Changes in thickness, size, color and shape have been found to be the best features to recognize melanoma in its early, curable phase. Unfortunately there is no method available at present to detect these changes in situ over time.

This thesis presents a new non-invasive method based on three-dimensional (3D) computed tomography (CT) of nevi, to scan the skin to detect early melanomas. The nevus is transilluminated using fibre optics directed into the surrounding skin. Three images of the transilluminated nevus are obtained at different angles by an optical device called a "nevoscope". The 3D reconstruction of the nevus is then obtained using new limited-view CT algorithms. Thus, without excision, a baseline 3D structure is established for each nevus. Some time later, the procedure is repeated. A quantitative and objective analysis performed on consecutive 3D reconstructions would bring out any changes in thickness, size, color, and structure of the nevus. Any changes detected would be taken as an early warning of probable malignant melanoma.

Computed tomography of nevi, in the approach presented, is a special case of limited-view CT. Images reconstructed from a few views (only three, in this case, as compared to 180 views in commercial CT scanners) are characterized by systematic distortion and streaking artifacts along the view angles used. A new CT algorithm, based on the 2D deconvolution of the point spread function of the reconstruction process, is presented. It incorporates a priori knowledge derived from the known projections. It is shown that if a linear (or a quasi-linear) reconstruction algorithm is used, the 2D deconvolution algorithm substantially removes the distortion and artifacts in limited-view CT.

Results of nevoscopy performed on a nevus of a cadaver are presented. The sectional computed reconstructions of the cadaver nevus correlate well with their corresponding histological sections.

Chapter 1

INTRODUCTION

Melanoma is the most malignant primary cutaneous tumor. Often the origin of a melanoma is a pigmented nevus. Metastases may occur early and the malignant transformation is accompanied by pigmentary increase, induration, and ulceration at later stages. It is the leading killer in dermatological disease. The number of deaths caused by melanoma was estimated to be 8,000 per year in North America in 1980 (Ariel, 1981). The occurance rate of malignant melanoma is rising faster than almost any other cancer in North America, doubling every 10 years (Kopf et al., 1984). Currently over 17,000 new cases of melanoma are anticipated in the United States per year (Sober,1983,a,b,c; cf. Gumport et al., 1981). Authorities project that 1 in every 250 persons will develop this threatening malignancy, if the current trend of increase in incidence rate continues (Kopf et al., 1984). The increase in mortality rate from melanoma in Australia, Canada, and USA has been found to be more than 100% from 1960 to 1970 in a survey conducted by World Health Organization (cf. Ariel, 1981). The mortality rate of malignant melanoma in the United States is increasing at a faster rate than any other cancer except lung cancer in women (Rigel et al., 1984).

The average age at which melanoma develops is in the mid-thirties for patients with atypical moles and often occurs in the twenties (Kopf et al., 1984). Fortunately nearly all patients with melanoma may be saved and cured if detected early and treated promptly. Unfortunately, in most cases of death (because of melanoma), malignant melanomas are not detected early enough.

The detection of melanoma in its early, curable phase is now primarily left up to the patient or the general practitioner. The change from a benign to a malignant lesion produces only a few warning symptoms that are often not observed by the patient (Wick et al., 1980). The classic signs that herald the transformation are changes in size, shape and color, development of inflammation, bleeding, ulceration, and appearance of pigment around the lesion.

Thickness in millimeters is the best prognostic factor for clinical stage 1 (early) melanoma patients even when regional lymph nodes contain metastases (Day et al.,1982 a,b,c,d). But until now this could not be measured directly without excision and histological sectioning of the nevus. Consequently, an attempt has been made to correlate other signs and symptoms with tumor thickness during the earliest stages of tumor growth. These signs are primarily increase in size and change of color (cf. Wick et al., 1980). Bleeding, ulceration, tenderness, and itching occur less frequently (Sober et al.,1983,a,b). The primary signs are rather subtle to expect a patient to notice, and since they involve what may be small changes in one or a few of what may be a large number of benign nevi, the probability of detection by the unsuspecting patient may be quite low (Cassileth et al., 1982). Moreover, a significant fraction of melanomas occur on the back, a difficult part to observe oneself. The general practitioner is in a way in a worse position than the patient, because she or he will generally have even less memory of the previous state of each nevus.

The thickness - survival relation is best characterized as a step function with the four natural thickness groups (1) less than 0.85 mm, (2) 0.85 mm to 1.69 mm, (3) 1.70 mm to 3.64 mm, (4) greater than or equal to 3.65 mm (Sober et al., 1983). (These groups have been formed on the basis of survival rates.)

In clinical stage 1 patients, the primary tumor thickness and anatomical location predict the survival rate better than do the thickness and histological determination of regional node metastases (Day et al., 1982; Sober et al., 1983). The survival rate for patients with primary tumor clinical stage 1, less than 1.69 mm thick, is excellent (greater than 95%). But patients with melanomas 2.75 mm thick on the hands or feet have a survival rate close to zero. Sober et al. (1983) found that nearly all patients with stage 1 primary tumor thickness greater than or equal to 3.65 mm anywhere on the body died because of melanoma within the following eight years.

The thickness of a tumor is not only important for predicting the survival rate, but also important for recognizing the cutaneous melanomas in the early curable phase. Sober et al., (1983 a,b), Day et al. (1982) and Breslow (1975 a,b) have demonstrated that the level of invasion and thickness of the lesion are highly correlated.

To recognize the melanoma in its early phase, the other important features are increase in size, change in color, and increase in elevation. These three signs are the most useful in detecting early lesions. Many other salient features, such as bleeding, tenderness, and lack of healing of the skin lesion occur late, and less frequently, in the natural history of melanoma. For the lesions of thickness greater than or equal to 3.65 mm, increase in size was noted in more than 72% of the cases studied. In 58% of the cases, color change was noted and in 82% of cases, increase in elevation of the primary tumor was found as the symptom or increase in sign (Sober, Day et al., 1983 a,b).

In patients with familial melanomas more than one primary is not uncommon (Clark et al.,1978). These patients often have multiple dysplastic nevi which differ from ordinary nevi in size, color, and variability and are precursors of melanoma. The lifetime risk of melanoma may approach 100% for those people with dysplastic nevi who are from melanoma-prone families (families with two or more first degree relatives having cutaneous melanomas) (Kopf et al., 1984). Melanomas in people from those families may arise within the dysplastic nevi or <u>de novo</u>. Changes detected would allow early excision of those nevi developing malignant melanoma.

The objective of this research is to develop a non-invasive technique to detect suspect nevi by 3D imaging of the skin lesion of the patient. The 3D reconstructions of all nevi would be performed through suitable CT techniques to conduct quantitative and objective analysis for the above mentioned features (thickness, size, color, structure, and elevation). Changes in the values of these parameters over time could be used to ascertain which nevi are suspected to be malignant. This method is completely non-invasive and could be used to detect melanoma in its early phase to save the life of the patient by excising the malignant lesion.

To create images of a nevus or melanoma that could be used in a 3D reconstruction process, the dermal object is transilluminated using fibre optics to direct light into the surrounding skin. One gets a direct impression of depth when viewing transilluminated nevi in a stereomicroscope (Gordon & Dhawan, 1983). The nevus looks as if it were backlit. Images of a transilluminated nevus may be recorded using an optical device, called a "nevoscope". A prototype nevoscope has been developed. It both transilluminates the dermal object and produces three images: an image at 45 deg, a vertical image at 90 deg, and a glancing image at 180 deg. These images are used for the 3D reconstruction.

Various limited-view reconstruction algorithms (Rangayyan & Gordon, 1982, Gordon & Rangayyan, 1983, Gordon, Dhawan & Rangayyan, 1984; Dhawan et al., 1984a, b, c; Dhawan et al., 1985), have been studied and modified for this specific application.

The ordinary reconstruction algorithms, like ART (Algebraic Reconstruction Techniques: Gordon, 1974) introduce a systematic geometric distortion and streaking artifacts along the view angles used. The reconstructed images are severely distorted when only a few views (such as three, in this study), are available for the reconstruction. The geometric distortion and other artifacts are caused by the 2D point spread function of the reconstruction process.

A new CT algorithm (Dhawan et. al., 1984a, b; Dhawan et al., 1985) has been developed which substantially removes the systematic geometric distortion and artifacts in the limited-view reconstruction. This algorithm performs 2D deconvolution via Wiener filtering incorporating a priori knowledge derived from the known projections.

The performance of these algorithms was tested by reconstructing simulated nevi from three views only (Dhawan et al., 1984a, b, c). Various computed sectional reconstructions of a nevus were obtained from the projections computed at those sections. The reconstructions of a simulated nevus and an actual nevus from only three views are both given in this report.

A quantitative analysis is performed on these three-dimensional reconstructions of a nevus for the prognostic variables such as thickness, size, and elevation. The structure of the reconstructed nevus may also be examined and compared to determine the correlation between reconstruction and histology, whenever excision is planned.

In order to check the relaibility of this method and to examine the correlation between reconstruction and histology, nevoscopy was carried out on a cadaver. Images of a transilluminated nevus were recorded in situ on a cadaver using nevoscope. The nevus was then excised, fixed, and sectioned for direct comparison between the reconstructed and actual sections. The procedures and the problems associated with this are discussed later in this thesis.

NOTE: The bulk of this thesis consists of my own work. I studied the problem of early detection of malignant melanoma in detail, which is reported in the next chapter. I developed "nevoscopy" from the concept level to the practical implementation, designed the required instrumentation (nevoscope), performed nevoscopy on a cadaver, did histology of the cadaver nevus, and compared the computed reconstructions with the histological sections (Dhawan et al., 1984d, e). The work on the development of an algorithm to isolate skin lesion's image from the surrounding skin was started by Daniel Martin, a medical student, but completed by me. The "geometric deconvolution" algorithm was developed by Drs. Gordon and Rangayyan but I modified its implementation and then used it for the reconstructions of nevus. I also worked on the geometric deconvolution algorithm to show that it improves the projections of the initial limited-view reconstruction which are used to obtain the final reconstruction (Gordon, Dhawan & Rangayyan, 1984). Later, I developed the 2D deconvolution algorithm for removing the artifacts in limited-view reconstructions (Dhawan et al., 1984a, b, c). These are reported in this thesis from Chapter 3 to Chapter 8.

Chapter 2

CUTANEOUS MALIGNANT MELANOMA: PROGNOSTIC GUIDELINES AND EARLY DETECTION

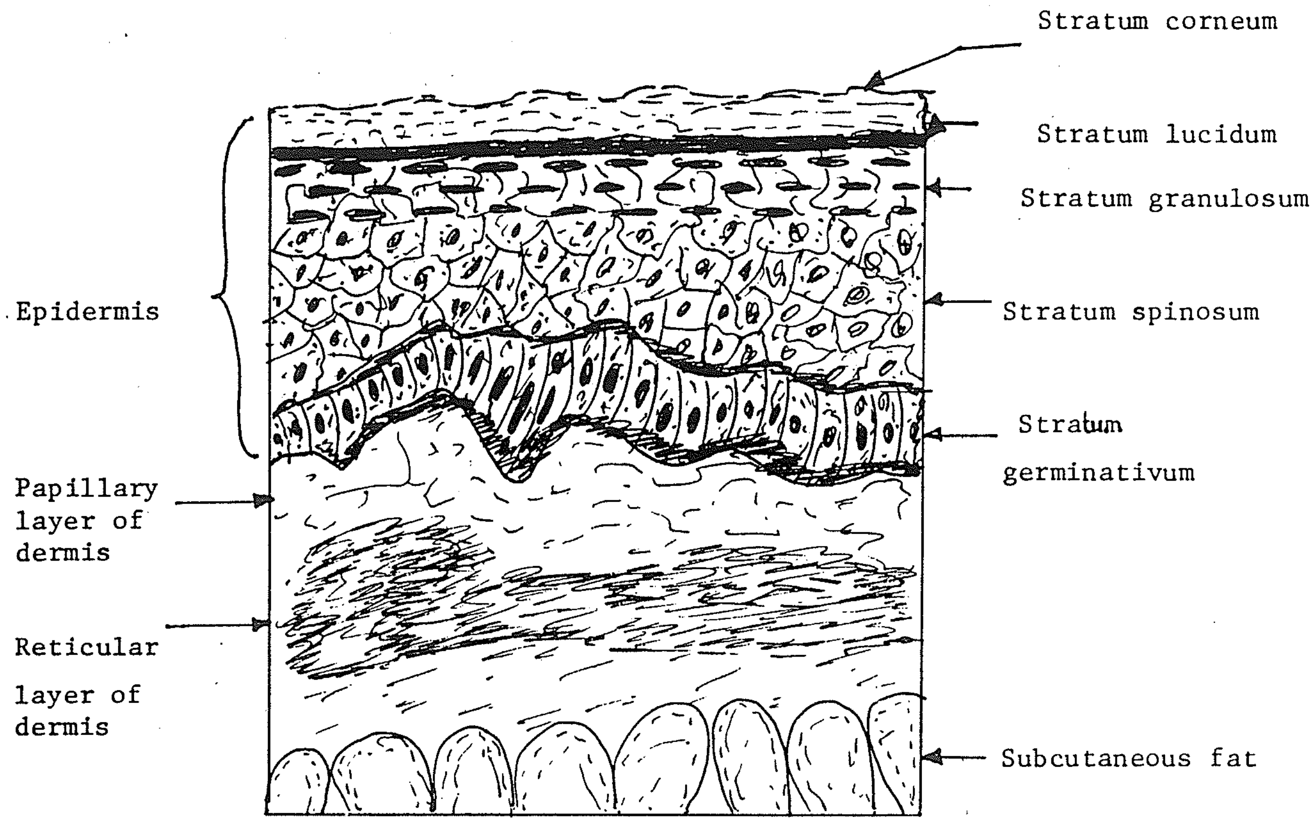
Malignant melanoma is the most lethal skin cancer. It often develops from a pre-existing birthmark, mole or a nevus. It is a pigmented growth that usually is characterized by variations of brown, black, gray, blue and even red and white. Although it begins as a slow initial growth of tumor cells on the surface of the skin or mole, malignant melanoma, after a period of time that may range from months to years, sends down "roots" into deeper layers of the skin where some of these microscopic extensions may enter blood vessels or lymph channels and spread to other areas of the body as metastases. If removed in early stages of growth, melanoma can usually be cured and recurrences prevented. Once it spreads or metastasizes beyond the skin, malignant melanoma can be fatal. Therefore, it is imperative that malignant melanoma be diagnosed early and treated promptly. Only 5% of all skin cancers are melanomas, but this 5% accounts for more than 70% of all deaths from skin cancer. In Canada, at least 80% of all reported deaths from skin cancers result from malignant melanoma alone (Canadaian Cancer Society Report, October 1982).

The Nevus and the Malignant Melanoma

A nevus is an accumulation of pigment cells in the skin. A nevus may be skin-coloured or pigmented. Basically, a nevus is a benign tumor arising from neural crest cells present since the embryonic period, which have differentiated into pigmented cells (Stewart et al., 1978). The cells composing it are called nevus cells. In adults these cells are present in collections in the epidermis and in the dermis, having migrated to these locations during the developmental period and then proliferated. They are intimately associated with the pigment-producing melanocytes and are of the same neural origin. They may form vari-sized growths, with varying degrees of brown pigmentation, commonly referred to as moles or "beauty marks".

In brief, the biology of pigment cells (melanocytes), which are present in large numbers in the skin, hair, eyes, and ears, is as follows. Cutaneous melanocytes arise from the neural crest, migrate to the epidermis during embryogenesis, and settle among basal keratinocytes. Each melanocyte extends numerous slender processes called dendrites, along intercellular spaces. Melanocytes synthesize the pigment melanin which is a highly stable polymer of oxidation products of tyrosine. There are three different melanins: eumelanin, phaeomelanin, and neuromelanin. Eumelanin is the brown-black pigment of skin and hair and most responsible for skin color. Phaeomelanin is a red-yellow pigment found in hair in human. Neuromelanin is present in neurons of the central nervous system, in adrenal medulla, and other areas of chromaffin system. Melanin has a variety of important functions such as protection of skin against sunburning, protection

Figure 2.0. Layer structure of skin.



against photodegradation of essential nutrients in the blood (such as folic acid), decreasing the formation of skin cancers, etc. Within the cytoplasm of the pigment cell, melanin is synthesized inside cytoplasmic organelles called melanosomes. The dendritic processes of the melanocytes project in between keratinocytes so that a single melanocyte supplies melanosomes to its group of 36 keratinocytes. These melanosomes migrate centrifugally along the dendritic processes of the melanocytes and are then transfered to or captured by keratinocytes. The size and the number of melanosomes transferred into keratinocytes determines to a great degree the color of person's skin. Melanocytes and melanin within the keratinocytes are transported through the epidermis and melanin is excreted by desquamation of the stratum corneum. Usually, in normal human skin, no melanosomes are present in the dermis except in mongolian spots. Melanosomes may be found in the dermis, contained in either melanocytes and/or nevus cells which produce melanosomes and retain them in their cytoplasm, or phagocytized in melanophages. When melanocytes proliferate into the dermis, as with malignant melanoma, large numbers of melanosomes also appear in the dermis.

Nevi or moles are common to all races, and most people have more than a dozen nevi (moles) somewhere on the skin. Published studies have reported numbers of nevi varying from 14 nevi/person to 40 nevi/person in Caucasians, on average (Bondi, 1984). They have a remarkable variation in appearance, since they may occur as flat, slightly elevated, dome-shaped, hair-bearing, smooth, or rough-surfaced lesions. Their color ranges from normal skin color through tan, gray, and dark brown to black. They vary from 1 or 2 mm in size to lesions covering a large part of the body surface. Their common characteristic is the possession of nevus cells or melanocytes appearing as rounded or polygonal, pale-staining cells in the dermis. The specific location of nevus cells in the dermis is used in the classification of nevi (Mishima, 1970; Mishima,1967; Rower, 1978).

In approximately 75%-85% of melanoma patients questioned, the nevus was present for a variable period before it underwent malignant transformation (Ariel, 1981). Most nevi appear during the early years of life (childhood and puberty). It is believed that these seldom undergo malignant transformation (Dubreuilh, 1912). Nevi that appear later in life are more suspicious. Dubreuilh termed them "precancerous circumscribed melanosis". Becker, in 1954, emphasized the danger of delayed nevi, which he called "malignant lentigo" (Becker,1954,1958).

Atypical (dysplastic) moles are associated with a higher risk of developing melanoma on normal appearing skin or in the moles themselves. Microscopic changes of dysplasia (abnormal growth of tissue) can be found even in minor portions of moles. Dysplastic nevi are often 3 mm, or larger in diameter, but 2 mm, histologically confirmed dysplastic nevi have also been found (Crutcher, 1984). The common features of dysplastic nevi include haphazard speckles of brown, dark brown or black hues, reddish brown or slightly pinkish background, erythematous (pinkish) areas, irregular and/or fuzzy outline that fades into normal skin, and irregular pigmentation. Numerous nevi, even hundreds, are sometimes present, but few or even solitary small atypical moles are common.

A congenital nevus is a melanocytic nevus that is present at birth. Not all pigmented lesions present at birth represent melanocytic nevi. Congentital nevi may be small (less than 1.5 cm in diameter), or medium (1.5-20 cm in diameter), or large (greater than 20 cm in diameter). Small congenital nevi may not be recognized at birth. Also, the clinical features of small congenital nevi may be the same as those of acquired melanocytic nevi. For example, they may have a smooth surface and uniform pigmentation. Congenital nevi occur in approximately 1 to 2.5% of the U.S. population (Rigel et al., 1984). Most of these nevi are small (less than 1.5 cm). It is still unclear to what extent melanoma develops in small and medium-sized congenital nevi (Kopf et al., 1984). The lifetime risk of melanoma in patients with large congenital nevi has been estimated to be 5 to 10% (Kopf et al., 1984).

Malignant melanoma is a neoplastic disorder of melanocytes with the potential of widespread invasion, metastasis, and subsequent mortality. The different types of melanoma exhibit different shades of color, sizes, shapes, and potential for invasion and metastasis. Initial growth of tumor cells usually takes place superficially across the surface of the skin or mole (Ariel, 1981). The early evolutionary changes in most of the cases last for a relatively long period (6 months to more than 5 years), during which, if they are recognized, cure is generally possible. However, once deep invasion has occurred, the survival rate drops drastically. Melanoma spreads not only by local extension and through regional lymphatics, but also by the blood stream. This accounts for the high mortality rate in invasive lesions.

Malignant change is indicated by changes in size, shape, or color, ulceration, pruritis, bleeding, irregularity of the margin, or satelittosis.

The hereditary nature of malignant melanoma is frequently commented upon in relation to family occurrence. Miller & Pack (1962) reported an example of familial malignant melanoma; a white girl died at the age of 36 because of malignant melanoma. The patient's maternal grandfather had died at the age of 42 years of a malignant melanoma originating on the lower leg, and her maternal uncle also died of melanoma. The patient's female infant had a black congenital nevus situated on the left leg at a location identical to her mother's melanoma. In 1962, more cases of familial melanomas were studied by Pack medical group (Pack, 1962). In most reported cases, a 67% to 85% incidence of preexisting moles was found in patients with familial melanomas. Pack believed that all melanomas arise from preexisting nevi, many of which may be invisible to the naked eye for some period (Pack,1961; Pack, 1948; Pack & Devis, 1962). Thus, inherited tendencies for developing malignant melanomas are not uncommon. Clark, Greene et al., (1978) described an atypical nevus labelled the B.K. mole (after the first two families investigated). They described the so-called inherited B.K. syndrome, in which certain nevi with specific characteristics were more prone to the development of malignant melanoma.

The notion that dysplastic nevi (DN) represent a class of melanocytic nevi which are precursors of cutaneous malignant melanoma has its origin in studies of melanoma-prone families (Greene, 1984). Greene, in the 43rd Annual Meeting of American Academy of Dermatology, reported that 51 family members with cutaneous malignant melanoma out of 401 members of 14 melanoma-prone families were examined. 48 (94%) of them had biopsy proven dysplastic nevi. Among 111 primary CMM (Cutaneous Malignant Melanoma) tumors in which an assessment could be made, 78 (70%) had histological evidence of a dysplastic nevus at a margin of the melanoma. It was also reported at this meeting that the lifetime risk of melanoma approaches 100% for those people with dysplastic nevi who are from melanoma-prone families (i.e. families with two or more first-degree relatives having cutaneous melanomas) (Rigel et al., 1984). Melanomas from these families may arise within the dysplastic nevi or de novo. The early and proper treatment of patients with familial malignant melanoma may be life saving. They may thus represent a group in which the effectiveness of nevoscopy can be rapidly evaluated.

The growing mortality from melanoma is due largely to the increase in population, increase in expected years of life, and the absolute increase in the numbers of melanoma (Ariel, 1981). The increased incidence of melanoma among the white races can be noted throughout the world. In Queensland it quadrupled in 30 years. In Norway, Britain, the United States, and Canada, the incidence has doubled in the last 10 years. The occurance rate of malignant melanoma is increasing faster than any other cancer in the United States except lung cancer in women. The mortality has also shown a relentless increase. At the current pace, 1 person in 250 in the United States will develop malignant melanoma (The Skin Cancer Foundation, 1984). Currently, over 17,000 new cases of malignant melanoma are seen annually in the United States alone out of which more than 8,000 persons die annually largely because their melanomas were detected too late.

Atypical (dysplastic) moles are associated with an increased risk of developing melanoma on normal appearing skin or in the moles themselves. Among Caucasians in the United States, the lifetime risk of developing cutaneous melanoma is currently about 0.53% (i.e. 1 in 185). Almost 5% of Caucasians may have dysplastic nevi and 40% to 50% of their close relatives may be effected. The National Cancer Institute (USA) reported that 18% of individuals with dysplastic nevi will eventually develop melanoma (Kopf et al., 1984). The risk is greater and may approach 100% for those people having one or more relatives with melanomas. This marked increase in incidence has not been observed in other skin cancers. In fact, in some countries, the mortality from other skin cancers exclusive of melanoma has declined (Elwood & Lee, 1979). Elwood and Lee have estimated a rise of about 9% per year in mortality, so that the death rate has doubled in the past 10 years. These rates are real and not due to better reporting or other factors in evaluation. If there were a common factor, all tumors rates would rise; however, such is not the case, for the increase in incidence and mortality is greater among younger people than those over 65 years of age. The fact is that people born in more recent years have both a higher incidence rate and a higher death rate from malignant melanoma, at all ages, than those born earlier (Lee &

Carter, 1970). The average age at which melanoma develops is in the mid thirties and cases often occur in the twenties (Kopf et al., 1984).

Classification of Melanoma

The modern assessment and treatment of malignant melanoma is based on a classification devised in 1967 by Clark (1967) and almost simultaneously by McGovern (1967) in Australia. Malignant melanomas are classified into (1) lentigo meligna melanoma, (2) superficial spreading melanoma, (3) nodular melanoma, and (4) acral lentiginous melanoma. Acral lentiginous melanoma has only recently been recognized as a histologic entity (Lupulescu et al., 1973).

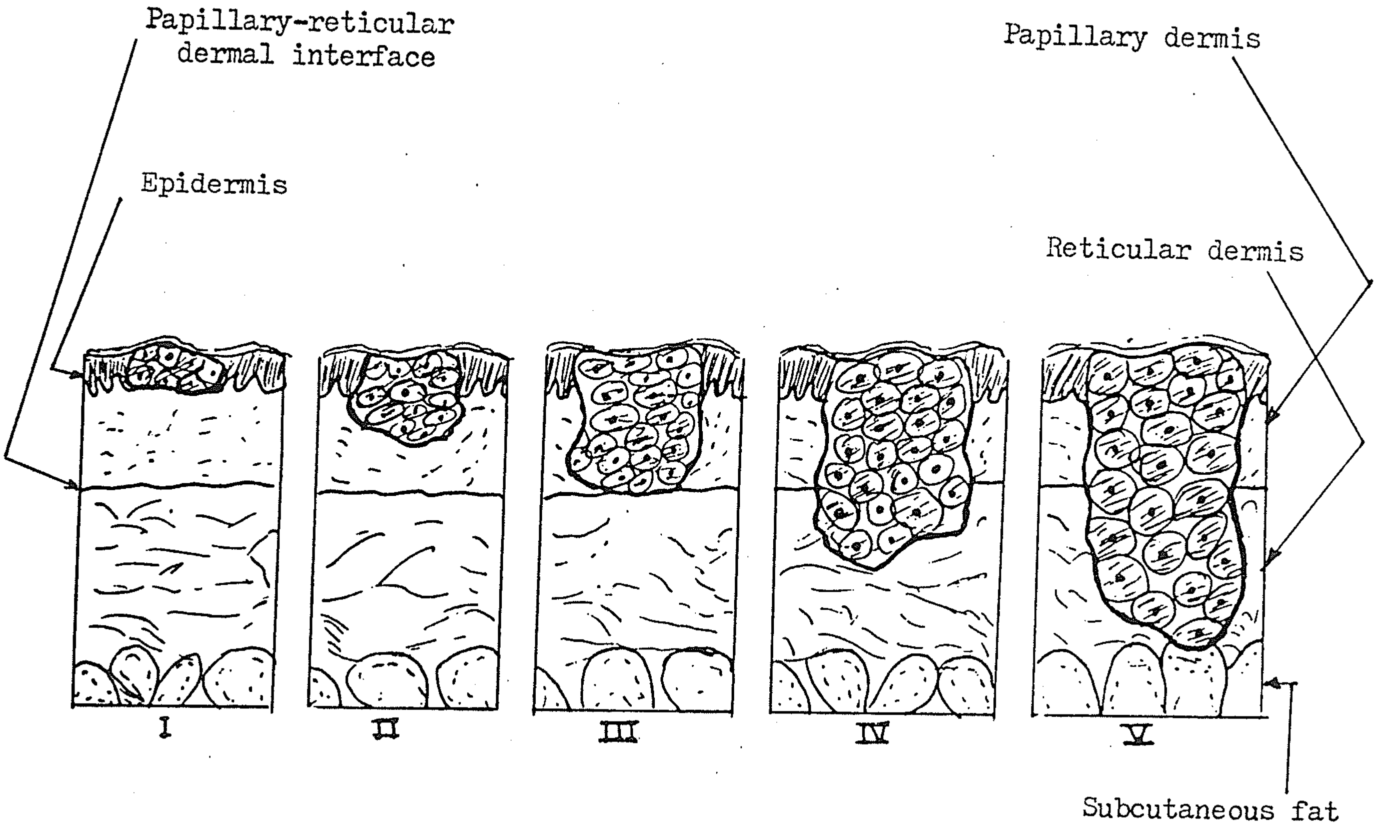
Some years ago, it was believed that malignant melanoma never occured in children before puberty (Ariel, 1981). Often the pathologist used to ask the age of the patient before making his diagnosis. If the patient's age was before puberty, the diagnosis would usually be benign nevus, if after puberty, as malignant melanoma. Woringer (1939) described a pinkish, slightly raised, lesion of the hand of an eight year-old girl, whose histology revealed unusual nevus cells. Webster, Stevenson, and Stout (1944) described pink elevated lesions in children that had the microscopic appearance of malignancy, but which very rarely metastasized. In 1948, Spitz called such lesions in children "juvenile melanomas" (Spitz, 1948). Spitz described the benign juvenile melanoma as a distinct entity with characteristic histological features. Kopf (1966) called them "benign juvenile melanomas" to emphasize the benign nature of the tumor, and to segregate the entity from the malignant metastasizing neoplasm. A few other synonyms are: spindle cell, epitheliod cell, and round cell juvenile melanoma, pseudomelanoma, juvenile nevus, and Spitz nevus. A typical benign juvenile melanoma is a papillar, pinkish lesion usually occuring on the cheek of a child, but it may occur anywhere in the body. The color of the lesion may vary from a tan to a deep brown. The juvenile melanomas usually vary in shape, increase in size (the average size is 7 to 10 mm), and then remain stationary. Kopf (1966) accumulated reports from various authors of 4,556 nevi excised in children, out of which 71 (1.6%) were benign juvenile melanomas. At present these lesions are assumed benign with the same potential of formimg malignant melanoma as do the usual compound nevi, which is rare. However, some authors believe that benign juvenile melanomas do have an increased potential of becoming malignant, especially after adulthood (Attie & Khalif, 1964).

Levels of invasion of these tumors are very important in regard to surgical or radiation therapy, and are directly related to survival. Originally, the depth of invasion was determined by the five levels suggested by Clark et al. (1967). They are as follows:

Level 1: intra-epidermal (*in situ*, pre-invasive); confinement of the malignant melanoma cells to the epidermis and its appendages.

Level 2: invasion into the papillary layer of the dermis,

Figure 2.1. Levels of invasion in melanomas



with at most only a few melanoma cells extending to the interface between papillary and reticular dermis.

Level 3: extension of the tumor cells throughout the papillary dermis, filling it and impinging upon the reticular dermis without, however invading it.

Level 4: invasion into the reticular dermis.

Level 5: invasion into subcutaneous fat.

Later, Breslow (1975), Cassileth & Clark (1982) and Sober et al. (1983) found that the level of invasion of melanoma is highly correlated with the thickness of the lesion (cf. Wick et al., 1980; Sober et al., 1983 a,b,c).

Melanomas are now usually staged into four categories (Ariel, 1981). Stage 1 represents a primary lesion without evidence of metastases to regional lymph nodes. These would include small satellites within the immediate vicnity of the primary melanoma. Stage 2 represents those that have metastases to the regional lymph nodes. Stage 3 are those that, in addition to metastases to regional lymph nodes, also have metastases to the afflicted limb: either satellites or metastases intransit. (Intransit(s) are defined as tissue metastases that result from implantation and growth of "in-transit" cells between primary and regional nodes.) Stage 4 represents blood-borne metastases. In addition to the above, some experts stage melanomas as following:

Stage 1: Localized primary melanoma

A. Primary intact

B. Primary locally excised

C. Mutiple primaries

Stage 2: Local recurrance or primary lesion with local peripheral nodules less than or equal to 3 cm.

Stage 3: Regional metastases

A. In-transit(s)

B. Node(s)

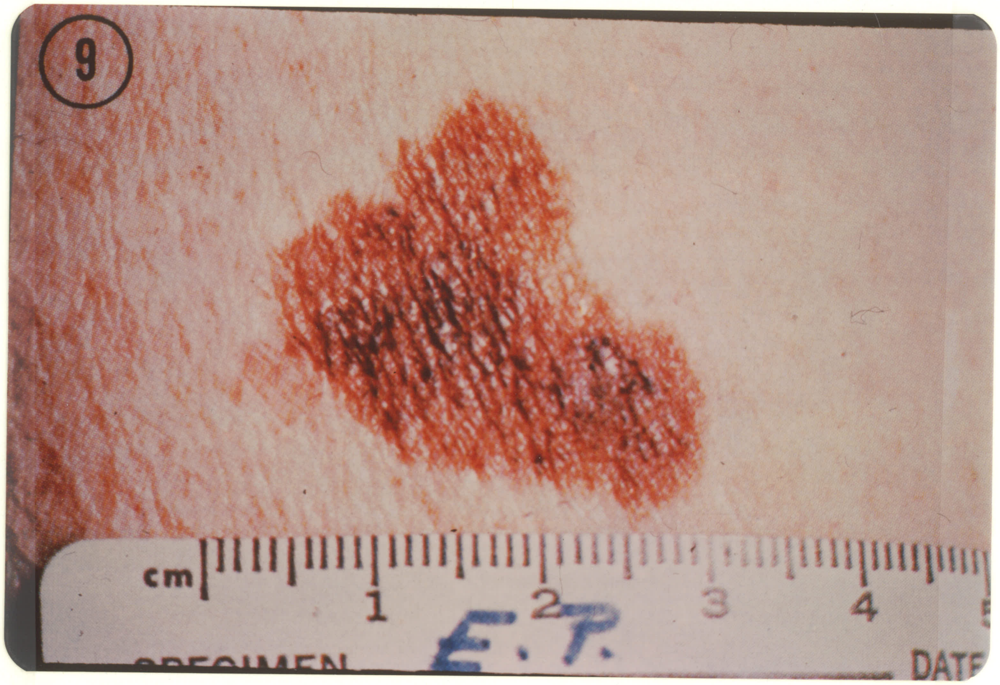
AB. Intransits, plus nodes

Stage 4: Systemic metastases

Lentigo Meligna Melanoma

Lentigo meligna melanoma constitutes about 8% of the malignant melanomas and follows a slow, steady progression through enlargement, mottling, darkening, and eventual thickening and nodule formation. In this, a focus of invasive melanoma develops in a pre-existing, usually long standing intra-epidermal lesion, also known as Hutchinson's freckle. These lesions affect persons over 50 years of age. They are usually quite large, flat freckles of varied color, irregularly

Figure 2.2. Lentigo Maligna Melanoma.



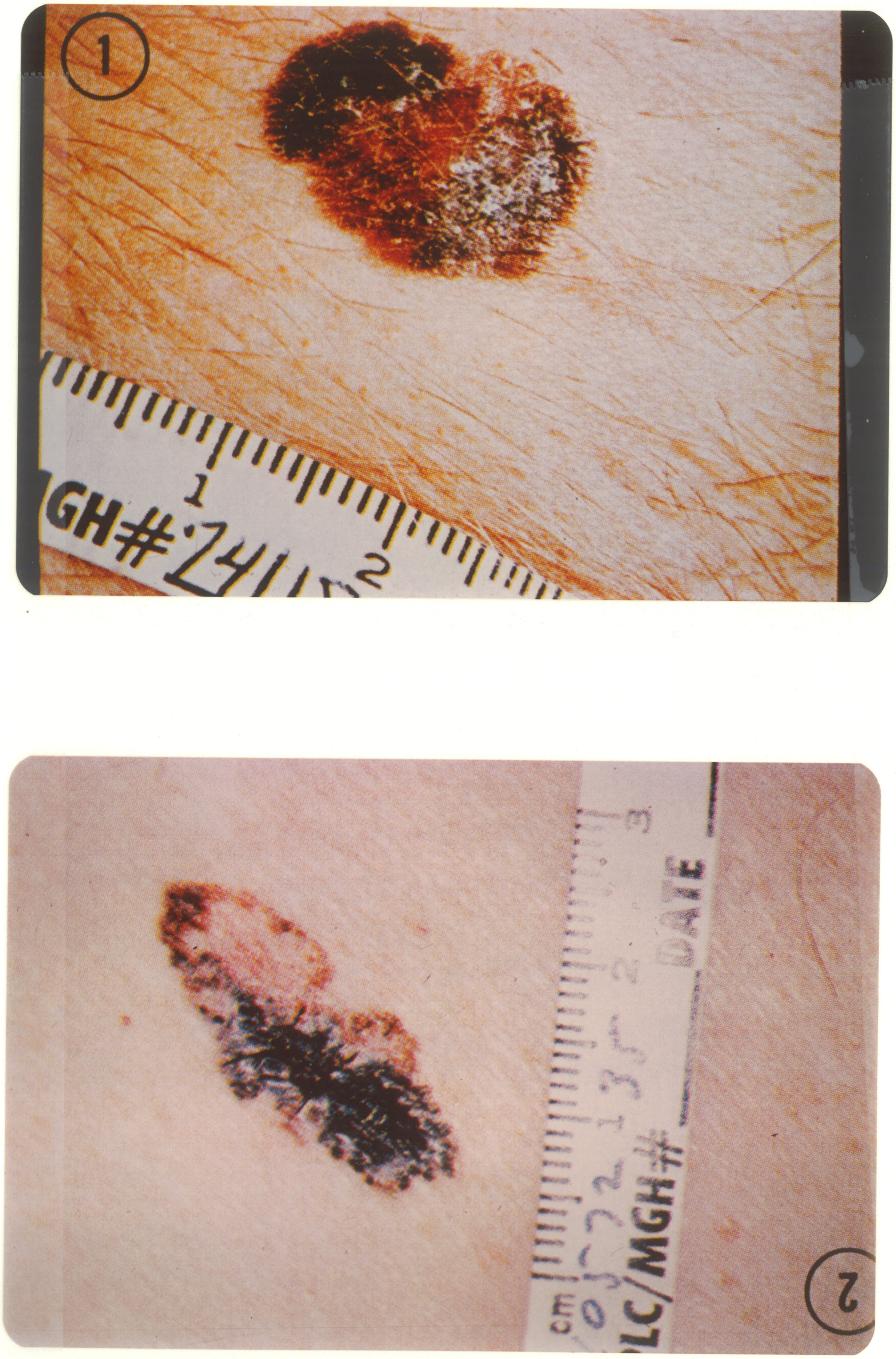
pigmented, and occur virtually exclusively in sun-exposed areas, commonly in the malar region of the face. The size is 3-6 cm or more with an irregularly scattered black nodular portion of size ranging from a few mm to 1-2 cm, on the surface. The color shades include tan, brown, black, and blue-gray. The survival rate is high, lying between 90% and 94% (Clark & Mihm et al., 1969).

Superficial Spreading Melanoma

This lesion constitutes 70% of malignant melanomas. It begins in young to middle age and has no predilection for specific areas of the skin. It may be flat or slightly elevated. The pre-invasive lesion is variable in color, with a fairly discrete but irregularly shaped outline. It tends to remain superficial and grow horizontally for several years, after which it develops a vertical growth phase with nodules becoming apparent. The superimposed invasive lesion is an elevated nodule, often with a verrucous surface, showing loss of normal skin marking (Mihm et al., 1973). The coloration of superficial spreading melanoma often includes a variegated assortment of pink, red, blue, black, tan, and white. Most of these melanomas arise from prexistent nevi. During the early epidermal phase, the pathologist may call them nevi with atypical cells, or active junction nevi, or atypical melanocytic hyperplasia.

The histological appearance of superficial spreading melanoma lesions is characterized by involvement of the full thickness of the

Figure 2.3. Superficial Spreading Melanoma.

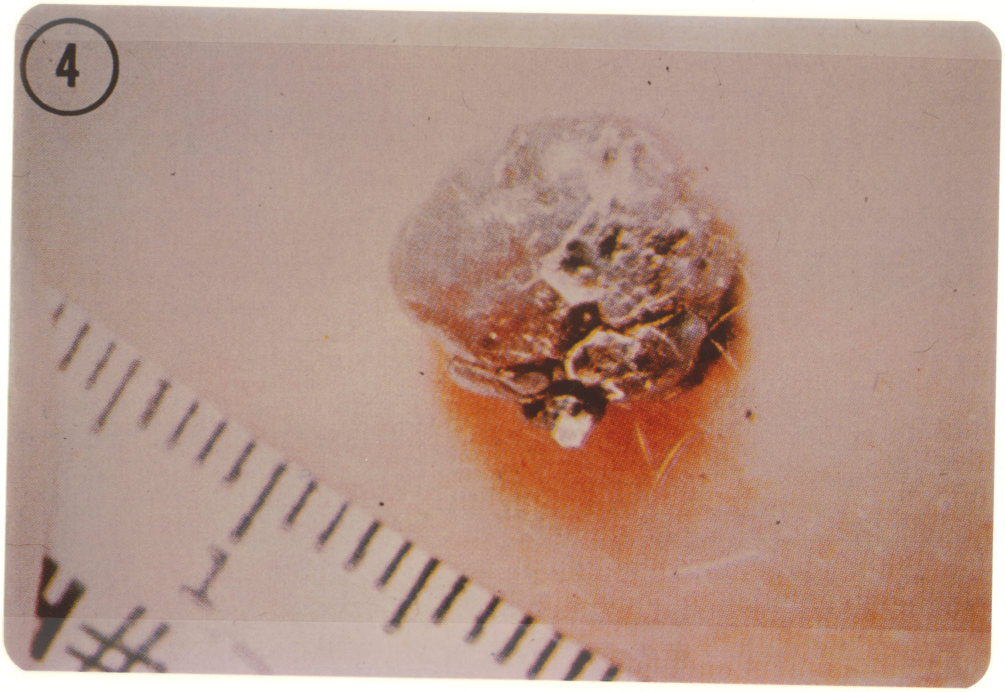


epidermis by malignant melanocytic cells, usually containing melanin pigment, and showing large nucleoli and mitotic activity (Stewart et al., 1978). Development into an invasive malignant melanoma is usually indicated by the appearance of papules and nodules or by diffused induration. Ulceration, if it occurs, is a late feature. The prognosis is predictable from the stage of the disease and the depth of penetration, and depends less on other factors, which are discussed later in this chapter. The 5-years survival rate of patients in stage-1 averages about 65% (Clark et al., 1969; Clark et al., 1979; McGovern et al., 1970; Sober et al., 1983; Day et al., 1982 a,b,c,d).

Nodular Melanoma

The characteristic nodular melanoma is an invasive lesion that grows more rapidly than other melanomas. It is pigmented, round to irregular, slightly elevated, embedded beneath the surface and constitutes about 15%-20% of the malignant melanomas. The color usually consists of shades of gray and gray-blue on a background of brown to black. A pigmented halo is not uncommon. The surface is smooth, and rarely verrucous. The border is less irregular than other melanomas. Microscopically typical melanoma cells occur at the junction of rete ridges and dermis as well as invading well into the dermis. The mitotic rate is higher than seen in the two previous categories of malignant melanoma. Lymphatic or blood vessel invasion can frequently be found in these lesions, and the incidence of lymph node and hematogenous metastases from these tumors is very high.

Figure 2.4. Nodular Melanoma.



Perineural invasion may also be encountered in the primary lesion. It increases in size very rapidly and often undergoes ulceration. When nodular melanoma is treated in stage-1, the 5-year survival rate lies between 50% and 60% (Clark et al., 1969).

Acral Lentiginous Melanoma

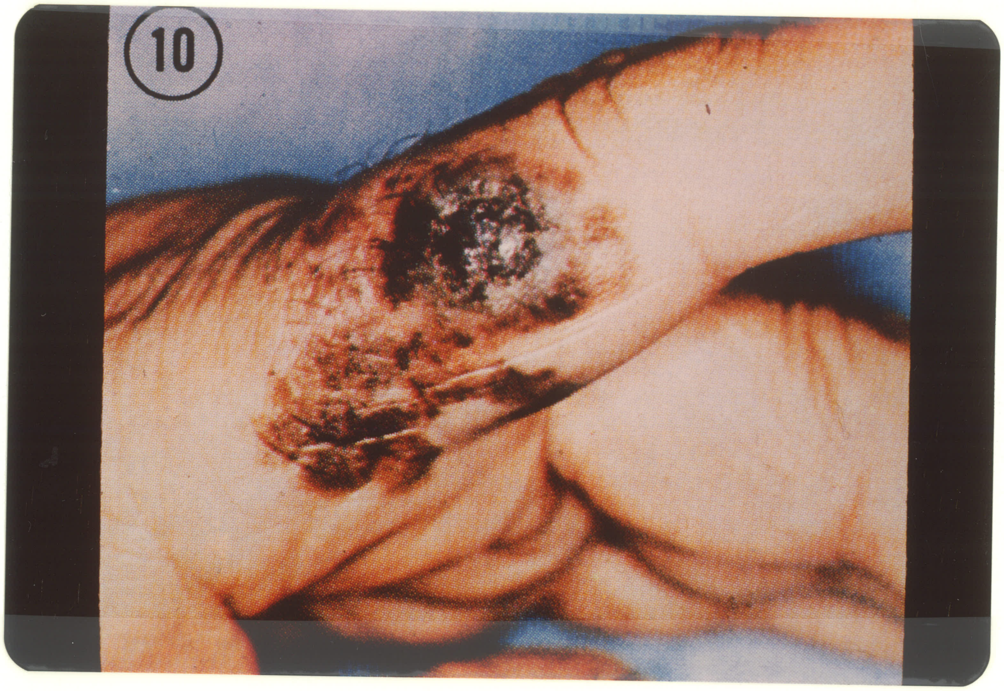
Acral lentiginous melanoma occurs on the hairless skin of the palms and soles and in the ungual and periungual regions. It is characterized not only by its specific location but also by its usually short period of in situ growth before invasive growth occurs. (Clark et al., 1979). Even though it accounts for only 8% of all malignant melanomas, it is the most common type in black patients (Pinkus & Mehregan, 1981).

Clinically, acral lentiginous melanoma shows uneven pigmentation with an irregular, often indefinite border (Lupulescu et al., 1973). Tumefaction and ulceration, as well as metastases, often occur within a short time, resulting in very low survival rates of only 11% to 15% (Coleman et al., 1980; Fleming et al., 1975).

General Histological Aspects of Malignant Melanoma

Whatever the subtype, melanoma cells have certain properties in common by which they can be recognized even in the absence of pigment

Figure 2.5. Acral Lentiginous Melanoma.



production.

Melanoma cells are usually large, polygonal, or spindle-shaped with an amphophilic, grayish cytoplasm and large nuclei. The large nuclei often have very prominent nucleoli. The nucleoli may be acidophilic and sometimes are so large that they resemble inclusion bodies. Melanoma cells grow in clusters but do not cohere (Ariel, 1981). Mitoses can almost always be demonstrated in considerable numbers in malignant melanomas (Clark et al., 1975).

Pigment production by melanoma cells can occur in an irregular manner. Pigmented cells in the deep portions of melanoma are not uncommon. This production of pigment in deeply situated cells is usually helpful in the diagnosis of doubtful cases.

Many, if not all malignant melanomas are underlain by a zone of chronic inflammatory cell infiltration. This is cited as indicative of the malignant nature of the lesion. However, inflammatory cells are also present in the vicinity of "juvenile melanomas", and thus should not give rise to erroneous diagnosis of malignant melanoma.

Prognostic Factors and Survival Rate Relationship

Clark et al., (1969) subdivided the invasion of the papillary dermis into a deep group in which melanoma cells accumulate at the junction of the reticular and papillary dermis (level 3), and a superficial group in which they do not (level 2). The overall mortality of patients with level 2 to level 5 lesions was 8%, 35%, 46%, and 52% respectively. The problem of this method is that it depends on the identification of the junction between the papillary and reticular dermis. In some parts of the body this is simple, but in many the interface is vague and the distinction is very difficult or impossible to make. There are also other problems with this method, which could not make it practical in many cases. The correlation between the maximal tumor thickness and the rate of metastasis has been found to be excellent (Breslow, 1970,1975). The rate of metastasis as a function of thickness is shown in Table I. This shows that the rate of metastasis increases with the increase in thickness: for lesions with the thickness less than or equal to 0.75 mm it is 1%, but for the lesions more than 3.00 mm thick there is a high metastasis rate of 84%. The correlation of maximal thickness and the rate of metastasis in the study of melanomas from all body sites was found to be very close with a correlation coefficient of 0.931 (Breslow, 1975).

Simultaneous statistical analysis of the thickness and level of invasion (Wanebo, 1975) revealed the maximal thickness to be superior to the level of invasion in predicting mortality. It has been found now that once thickness is measured, there is little that can be added by determining the level (Geelhoed & Breslow, 1977). This was also the conclusion of various studies of melanomas of the extremities: 95% of the available prognostic information was recovered when thickness was measured first, leaving only 5% to be recovered by determining the level (Breslow, 1978). Maximal tumor thickness has been found to be an independent variable and "the single most important determinant for

Table I

THE RATE OF METASTASIS AS

A FUNCTION OF PRIMARY TUMOR THICKNESS

Thickness (mm)	Number of cases	%Metastasis
0-0.75	54	1
0.76-1.50	27	33
1.51-2.25	19	33
2.26-3.00	13	69
greater than 3.00	25	84

Taken from Ariel (1981).

stage 1 disease" (Day et al., 1982).

More recently Balch et al. (1980); Balch et al. (1979); Day et al. (1981 a,b,c,d); Day et al. (1982a,b,c,d,e); Wick et al. (1980); Sober et al. (1979); Sober et al. (1980); and Sober et al. (1983a,b,c,d); have found that: (1) thickness in millimeters, of primary tumor, and (2) anatomical location of the primary tumor are the two most important variables which highly correlate with the survival time of the patient. The survival rate for cutaneous melanoma with distant metastases (clinical stage 3) has been found zero, because all these patients eventually died of disease within five to sixteen months. Approximately 30% of clinical stage 2, pathological stage 2 patients live for five years, and 10-year survival is uncommon. Balch et al. (1980) had no 10-year survivors among this group. In clinical stage 1 patients, the survival rate depends on many prognostic factors (McGovern, 1976). They include thickness of the lesion, location of the lesion, sex of the patient, age of the patient, and number of mitoses per square mm in a standard section. The probability of death from clinical stage 1 melanoma has been tabulated in Table II by primary tumor thickness. Table III shows the probability of death in the first seven years after diagnosis of clinical stage 1 melanoma by tumor thickness and anatomical location. Some of the important facts regarding survival and prognosis are as following: (1) Specific location and primary tumor thickness are better than any other combinations for predicting recurrence and death; (2) Clinical stage 1 melanomas less than 0.85 mm. thick have an excellent patient survival rate (greater than 95%); (3) Clinical

Table II

PROBABILITY OF DEATH FROM CLINICAL

STAGE 1 MELANOMA BY PRIMARY TUMOR THICKNESS

Thickness in mm	Two & half years in percent	Five years in percent	Seven & half years in percent
less than 0.85 (N=190)	1	1	1
0.85-1.69 (N=178)	3	6	7
1.70-3.64 (N=151)	12	24	31
greater than equal to 3.65 (N=79)	43	59	82

N is the number of cases investigated.

Taken from Sober et al. (1983).

Table III

PROBABILITY OF DEATH (%) FROM MELANOMA IN THE FIRST SEVEN YEARS AFTER DIAGNOSIS OF CLINICAL STAGE 1 MELANOMA

Thickness (mm)	Non-BANS	Non-BANS	Non-BANS	Hands & Feet	BANS
	Extremities	Head	Trunk		
	Excluding	and			
	Hands & Feet	Neck			
0-0.85	0	0	0	0	0
0.85-1.69	0	0	3	0	22
1.70-3.64	14	36	23	40	42
greater than equal to 3.65	17	35	78	100	67

The table shows seven-year death rate from melanoma determined by the life table analysis of 598 clinical stage 1 patients from the Massachusetts General Hospital and New York University Medical Center. BANS=upper Back, posterolateral Arm, posterior and lateral Neck, and posterior Scalp.

Taken from Sober et al. (1983).

stage 1 melanomas from 0.85 mm to 1.69 mm thick anywhere on the body excluding the posterolateral portion of the arm rarely metastasize and have the same excellent patient survival rate; (4) Clinical stage 1 patients with melanomas greater than or equal to 3.65 mm thick on the trunk and those with melanomas greater than 2.75 mm thick on the hands or feet nearly all die of melanoma; and (5) The clinical stage 1 patients most likely to benefit from elective regional node dissection are those with melanomas from 0.85 mm to 1.69 mm on the upper back, posterolateral upper arm, posterior and lateral neck, and posterior scalp, and most patients with melanomas 1.70 mm to 3.64 mm thick. However, an early total excision may save or at least prolong the life of the patient of any category (Gumport et al., 1981).

Early Detection of Primary Malignant Melanoma

Many salient features of malignant melanoma such as bleeding, tenderness, and lack of healing occur very late. But for early detection of malignant melanoma (level 1 to level 2), change in size, change in color, and change in elevation are found to be the most often seen features. Since primary tumor thickness is a more accurate predictor of prognosis than level of invasion, and level of invasion and thickness of lesion are highly correlated, it has been found by Sober et al. (1983) that thickness, size, color, and elevation form the best combination of frequently seen symptoms for the detection of early melanomas. Table IV shows the percentage of patients who reported the presence of a symptom or change in sign in a study of

Table IV

PERCENT OF PATIENTS PRESENTING WITH THE FOLLOWING SIGNS AND SYMPTOMS BY PRIMARY TUMOR THICKNESS IN MM.

Thickness	0-0.84	0.85-1.69	1.70-3.64	>=3.65
Size	55	50	51	72
Color Change	49	48	47	58
Elevation	46	51	59	82
Bleeding	13	25	45	63
Ulcer	5	15	33	45
Tenderness	8	7	14	19
Itching	12	20	28	46

Taken from Sober et al. (1983).

598 patients conducted by Sober et al. (1983) while Table V shows the percentage of patients presenting with the signs and symptoms by level of invasion (Wick et al., 1980).

The melanoma danger signs, as described by the Skin Cancer Foundation, New York, are any of the following in localized area of the skin.

1). Change in size: sudden increase in size is of special concern, slow change is much more common. This change in size may be found unevenly in all the three dimensions. For example, in the case of superficial spreading melanoma, horizontal spread of cells is common in the very early phase which is immediately followed by lateral spread. But in the case of nodular melanoma, the lateral (vertical) growth is seen more frequently.

2). Change in color: slow or sudden darkening of brown or black shades, or the mixing of shades of red, white, and blue.

3). Change in shape or elevation: finding of an irregular border with or without notches of a lesion which used to have a regular border. The elevation above the surface or the total thickness which is more precised and relaible, is another important sign. The irregularity of the border of mole is usually accompanied by change in shape of the mole (for example from round to oval shape).

Table V

PERCENTAGE OF PATIENTS PRESENTING WITH THE FOLLOWING SIGNS AND SYMPTOMS BY LEVEL OF INVASION

	LEVEL			
	II	III	IV	V
Size	71	68	64	83
Color	55	55	54	64
Elevation	56	65	67	80
Bleeding	10	26	42	54
Ulcer	4	13	27	43
Tenderness	9	11	15	18
Itching	24	25	36	43

Ref.: Wick et al., 1980.

4). Change in surface characteristics: flaking, scaliness, oozing, crusting, ulceration, bleeding, appearance of a nodule or bulging, etc. These signs may be seen in malignant melanomas but usually are observed late or very late.

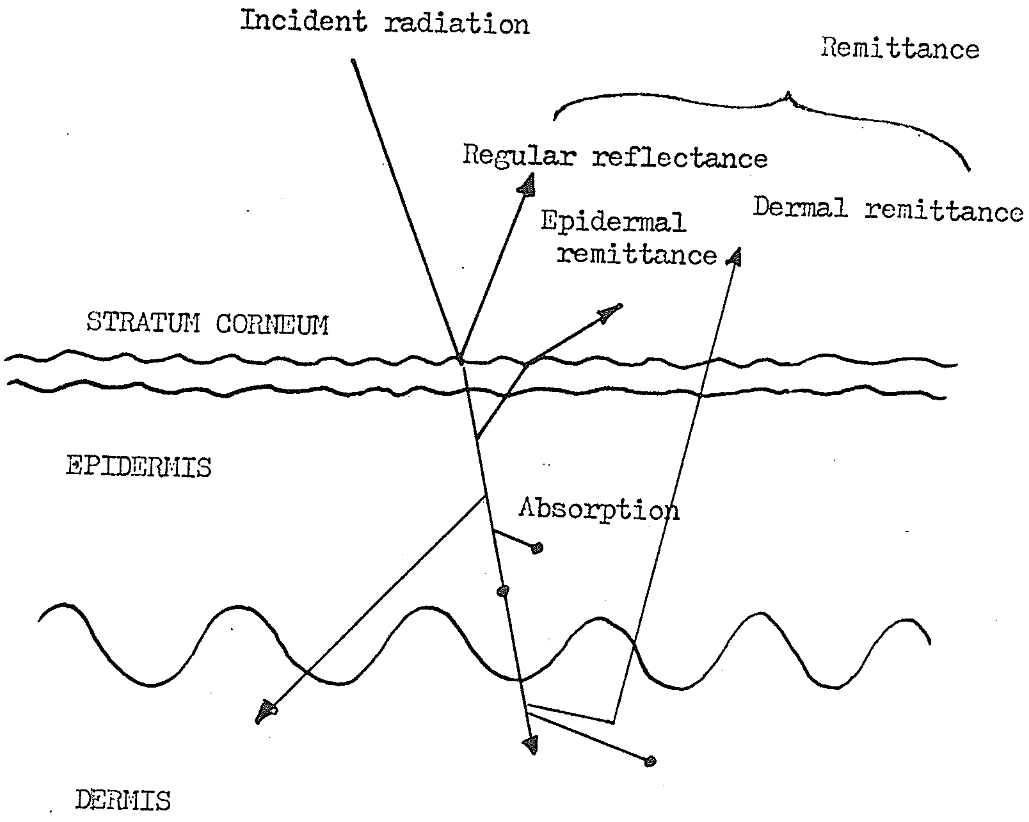
Thus, from these facts and investigations, it is evident that changes observed in the three-dimensional size or shape (growth of cells in all the dimensions) and the color of the skin lesion over time are the most useful features in recognizing early malignant melanoma. Total thickness is the most common and strongest prognostic factor found in all prognostic models. This suggests that early detection of primary cutaneous malignant melanoma (superficial spreading melanoma, and nodular melanoma, in particular) may be feasible by three dimensional computed tomography of lesions and by observing the above mentioned parameters over time. This will allow the possible excision of the lesions in time to save the life of the patients.

Chapter 3

PROCEDURES AND METHODS: TRANSILLUMINATION AND THE NEVOSCOPE

Computed Tomography by transillumination is a case of the inverse radiative transfer problem, and received its first mathmetical solution in this form by Chandrasekhar (1960 a,b). Here, it has been treated in a simpler fashion, in effect considering that the nevus cells act as absorbing objects seen against a bright (transilluminated) background (cf. Anderson et al., 1981). Dermal scattering plays a major role in determining the depth to which radiation of various wavelengths penetrates the dermis (Hardy et al., 1956; Bachem & Reed, 1929). In general, longer wavelengths across the UV-visible-near infrared spectrum penetrate the dermis to a greater extent than do shorter wavelengths. The absorption and scattering in the dermis layers also determine the appearance of different colors in skin lesions. For example, the appearance of blue skin nevi can be explained based upon this fact. The average dermal pathlength and depth of penetration of remitted longer wavelengths (such as red) light is larger than that of shorter wavelengths (such as blue) light. This is because of increased scattering at shorter wavelengths. In blue nevi, melanin is pathologically deposited fairly deep in the dermis. Blue light encounters less of the dermally deposited melanin

Figure 3.1. Schematic diagram of optics of human skin..



than red light, and may therefore suffer less absorption. Such differential scattering is the only means by which a pigment, such as melanin, which absorbs shorter wavelengths more strongly than longer wavelengths, can produce blue color (Anderson & Parrish, 1981).

The Optics of Human Skin: An Overview

Whenever skin is involved as the site for photo-biological reaction, its optical properties play a major role in affecting the response. Radiation passes through the stratum corneum before reaching tissues, and hence the thickness, composition, and morphology of the stratum corneum is always a modifying factor. Having reached viable tissue, the radiation is scattered and absorbed by structures and chromophores which vary dynamically and between individuals (Anderson & Parrish, 1980).

To understand this, consider the optics of normal skin as shown in Figure 3.1. At nearly perpendicular incidence, a small fraction of an incident radiation is reflected due to the change in refractive index between air (ref. index = 1.0) and stratum corneum (ref. index = 1.55) (Scheuplein, 1964). For normally incident radiation, this regular reflectance of an incident beam from normal skin is always between 4% and 7% over the entire spectrum from 250-3000 nm, for both white and black skin (Parrish et al., 1978). This same air-tissue optical interface also causes internal reflections of diffused, back-scattered radiation. Because the surface of the stratum corneum is not smooth and planar, skin reflectance does not maintain an image. Also, a beam of collimated incident radiation, upon passing through this surface and into the skin, is refracted. These effects are similar to those which make ground glass translucent, compared with the transparency of polished glass.

Within any of the layers of the skin, the 93% to 96% of the incident radiation not returned by regular reflectance may be absorbed or scattered. These two process taken together essentially determine the penetration of radiation into the skin, as well as remittance of scattered radiation from the skin. Scattering results from inhomogeneities in a medium's refractive index, corresponding to physical inhomogeneities. The spatial distribution and intensity of scattered light depends upon the size and shape of the inhomogeneities relative to the wavelengths, and upon the difference of the refractive index between the medium and the inhomogeneities (Anderson et al., 1980). For molecules or small particles with dimension less than one tenth of the wavelengths, scattering is generally weak, polarized, and varies inversely with the 4th power of wavelength (Rayleigh scattering). For particles with dimension on the same order as the wavelength, scattering is much stonger and forward-directed. When the particle size greatly exceeds the wavelength (so called Mie scattering), scattering is again diminished and becomes highly forward-directed. All these types of scattering occur within the skin (Parrish et al., 1978).

Kubelka (1954) presented a model for radiation transfer in a turbid, absorbing medium. Anderson & Parrish (1981) utilized the Kubelka-Munk model in the problem of optical radiation transfer of skin to measure the transmittance and remittance of skin layers. They considered the dermis as a turbid tissue matrix and found that with the dermis, optical scattering was an inverse function of wavelength and largely defined the depth of optical penetration.

In the visible part of the spectrum, melanin is essentially the only pigment affecting the transmittance of normal human epidermis. This causes a wide range of discernable skin colors from "black" to "white". The 300nm transmittance of full thickness suction-separated epidermis including the basal cell layer varies by 2 to 3 orders of magnitude from Caucasian to darkly pigmented Negro individuals (Hais, 1969). The lack of significant absorption by melanin for wavelengths longer than 1,100 nm, and increased absorption at shorter wavelengths, have been observed (Anderson et al., 1981).

For dermis the transmittance is both higher and forward-directed for longer wavelengths over the region between 0.5 and 1.25 um. On the other hand, scattering is of major importance in the dermis. Dermal scattering plays a major role in determining the depth to which radiation of various wavelengths penetrates the dermis. Scattering makes it possible for a black pigment such as melanin to produce different colors. Melanin absorbs shorter wavelengths more strongly than longer wavelengths. The optical scattering is also an inverse function of wavelength.

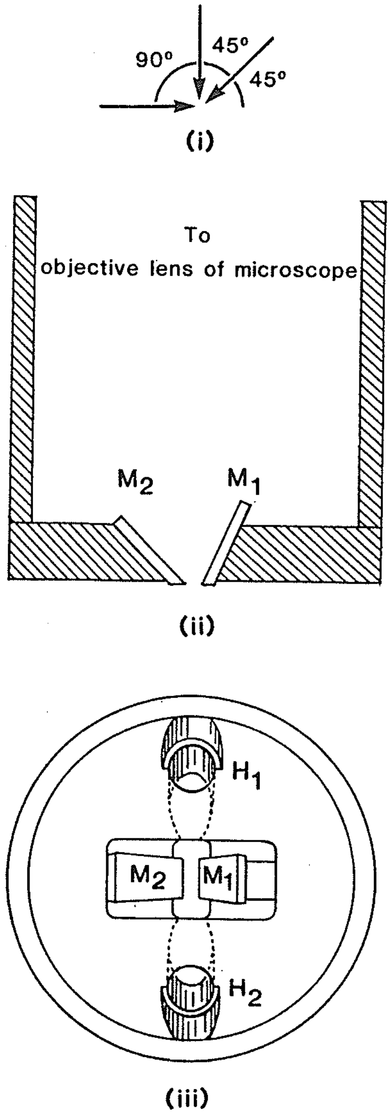
Anderson found that up to 1% of the 605 to 850 nm wavelength region penetrates the entire human chest wall, post mortem (Anderson et al., 1981). More than 25% of the incident radiation density of wåvelengths ranging from 400 to 800 nm can penetrate approximately 0.5 to 3 mm deep for fair Caucasian skin (cf. Anderson et al., 1981). These data are based on a highly simplified model of human dermis (Kubelka-Munk model for radiation transfer in turbid, absorbing medium).

Thus, the stratum corneum and epidermis provide an optical barrier primarily by absorption of radiation, and to a lesser degree, by optical scattering and reflection. The dermis may be considered a turbid tissue matrix within which optical scattering largely defines the depth of penetration and apparent color of skin lesions. Moreover, the absorption bands present in the remittance spectra of skin can be used to monitor or analyze the pigmentation responses (Anderson et al., 1981).

The Nevoscope

The prototype nevoscope has been built out of a plexiglass cylinder machined to fit around the objective lens of a Wild M8 stereomicroscope. Nylon screws permit it to be moved up and down along the tube of the objective lens for focussing. A plexiglass plate screws on to the bottom of the cylinder. This plate (Figure 3.2) contains a trapezoidal slot. Two front surface mirrors (9 x 19 x 1.8 mm, Efstonscience, Toronto) were glued onto the slanted parts of the slot at intended angles of 45 deg and 22.5 deg from the vertical. The actual angles were measured by reflection of a beam of light and were found to be 45 deg and 23 deg respectively. Light for

Figure 3.2. View angles and line diagram of nevoscope.

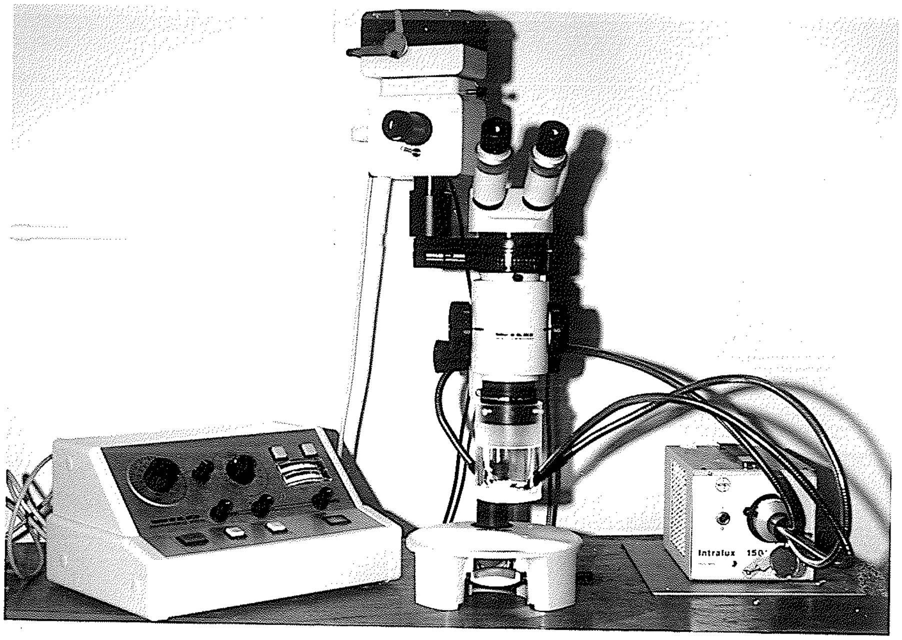


(i)

(ii)

(iii)

Figure 3.3. Nevoscope mounted on a stereomicroscope.



imaging was provided by a Volpi (Switzerland) heat filtered illuminator using Philips #6423 15V, 150W bulbs. Two of its three fiber optics bundles were directed in a plane perpendicular to the slot through two holes drilled at 45 deg through the cylinder and the plate (Figure 3.2). These were allowed to protrude slightly (about 2 mm) so that they would dent the skin and make direct contact with it. (The protruding edges of the mirrors were ground down with a carborundum stone to allow the skin to bulge inwards next to the silvered surfaces.) Some of the superficially backscattered light was blocked off with black tape on the lateral walls of the trapezoidal slot. The transilluminated image, so obtained, was recorded on 35 mm Panatonic-X (Kodak) film and then the film was digitized using a Sony (Japan) video camera AVC-3260 with a Nikon 55 mm macro lens. Digitization was accomplished at a six bit resolution averaging 8 consecutive frames by Grinnell GMR-27 frame buffer.

For surface illumination, the third optics fiber bundle was directed at the nevus through the transparent cylinder. The transillumination was not blocked.

The nevoscope with the stereomicroscope is shown in Figure 3.3.

Chapter 4

PROCEDURES AND METHODS:

HISTOLOGICAL SERIAL RECONSTRUCTION OF SKIN LESION

Since it was important to study and compare the histological sections of a skin lesion (nevus or melanoma) with the computed 3D reconstruction of the same lesion, nevoscopy was performed on a cadaver. A nevus on a cadaver was first trnasilluminated in situ using the nevoscope and then excised the tissue to obtain histological serial sections.

After excision, the tissue was fixed to prevent postmortem changes and to minimize shrinkage and distortion during subsequent procedures. The tissue was dehydrated after fixation and then embedded for sectioning.

The lesion's left top corner was marked on the skin of the cadaver before the tissue was excised. To orient the tissue properly for cutting sections perpendicular to the skin, the tissue was first trimmed into a cubic shape. A dissecting needle was pierced across the dermis to the left of the lesion. A fine wire was pushed through the needle and left in place during fixation to make a hole across the dermis. The fixative used just after excision of tissue was 10% neutral buffered formalin. After keeping the tissue in formalin for

24 hours, it was dehydrated in 30%, 50%, 70%, and 90% alcohol for 15 minutes each. Then it was put in absolute alcohol for 20 minutes and in methanol for 20 minutes. It was infiltrated in "solution A" consisting of glycol methacrylate, butoxyethanol and benzoyl peroxide overnight. Embedding was done with a JB-4 Plastic Embedding Kit with a solution A plus "solution B" consisting of Polyethylene glycol 400 and N-dimethylaniline.

After embedding the specimen in plastic, serial sections were cut using a Sorvall JB-4 (Connecticut) Microtome with a glass knife. Sections were cut with a thickness of 5 microns and then mounted on plain slides and dried at 60 deg C for 48 hours. Standard staining procedure with hematoxylin was used (Humason, 1979). After staining, sections were mounted on the slides with Protex mounting coverslips.

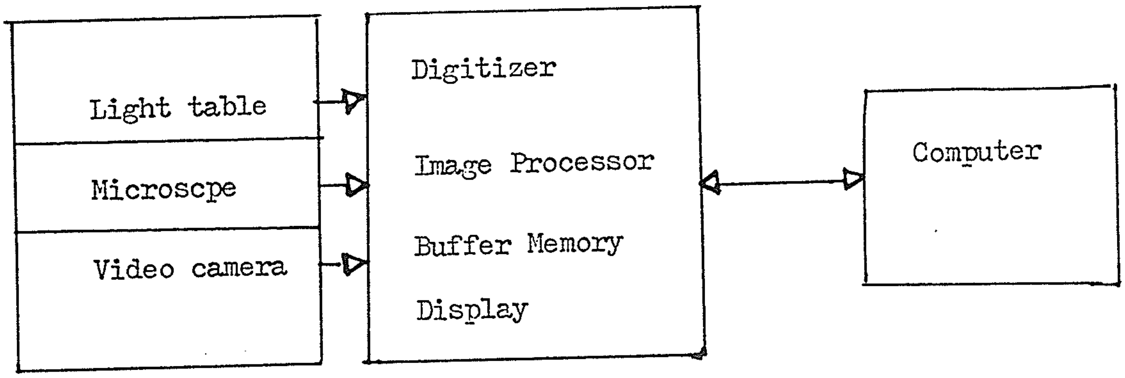
Chapter 5
PROCEDURE AND METHODS:
IMAGE PROCESSING SYSTEM

The image processing system used consists of three major components: A) the image input devices such as a light table, microscope, video camera, etc., B) an image processing buffer with digitizer and display, and C) a minicomputer or computer with peripherals. A schematic block diagram of the image processing system is shown in Figure 5.1.

All programs for image processing except the two-dimensional Fast Fourier Transform (FFT) and Wiener filter were implemented on a Control Data Canada (Toronto) Cyber 171 computer at the Medical College campus of the University of Manitoba, Winnipeg, Canada in the Pascal language (University of Minnesota compiler). The two-dimensional FFT and Wiener filter programs were written in the FORTRAN IV language. File handling of images was performed with the Cyber Control Language (CCL) and NOS operating system commands. Documented programs are given in Appendix A.

The images were displayed on a Conrac (Covina, Calif.) color television monitor via a 512 x 480 pixel Grinnell GMR-27 (San Jose, Calif.) frame buffer with 10 bit planes connected to the Cyber

Figure 5.1. Block diagram of image processing system.



computer via two CAMAC crates linked by a serial highway. Image enhancement, thresholding, and boundary tracing were carried out using the CDHS Manitoba Video Processor (MVP) software. Digitization was accomplished at 6 bits density resolution with averaging of 8 consecutive video frames by the frame buffer. The video images were acquired with a Sony video camera (AVC-3260) using a Nikon (Japan) 55 mm macro lens on a C-mount adapter.

The grey scale 0-63 for acquired 6 bit resolution video images was transformed linearly to 0-255 (8 bits). The Fourier magnitude spectrum images were transformed to a higher grey scale (to enhance the details) before their photographs were taken. Photographs were taken with a Lang System Videoslide 35 mm camera unit using 32 and 400 ASA Kodak black and white films.

Chapter 6

PROCEDURES AND METHODS: IMAGING OF PIGMENTED SKIN LESION

Algorithm to Isolate Melanotic Lesion from Surrounding Skin

Before a subset of an image (that part of the picture which contains only the object of interest such as pigmented lesion in the photograph of lesion on the skin) can be characterized, it is necessary to isolate it from the background. A computer algorithm has been developed to separate the pigmented lesion from the background of surrounding skin (Dhawan et al., 1985). This approach is different from those ordinarily used in automated cytology to separate the image of a cell from its background (Wied et al., 1976), because pigmented lesions (nevi or melanomas) may have faint irregular margins which must be included unlike the relatively well defined boundaries of cells. The algorithm is based on an image processing method called "Region Expansion" along with an adaptive histogram technique.

The histogram techniques for image enhancement and image segmentation are well known in image processing (Gonzalez & Wintz, 1977; Hall, 1979; Rosenfeld & Kak, 1982). Image segmentation by histogram thresholding (Rosenfeld, 1969; Hall, 1972) is an approach based on dividing the grey scale into bands and using thresholds to determine regions or boundary points. The accuracy of this approach is entirely based on the computation of thresholds. The problem of finding such thresholds for optimum results was discussed by Chow & Kaneko, (1972) for outlining boundaries of the left ventricle in cardioangiograms. Their approach included preprocessing by logarithm transformation of every pixel, subtraction of images and averaging of several angiograms before computation of optimal thresholds. The method finds the thresholds for a region of interest and the remaining regions of the image. It then assigns a threshold to every point in the image by interpolating the original thresholds of the region of interest and remaining regions. But these methods do not isolate an irregular region of interest from its background and their performance depends on how well the region of interest is defined. This approach is much simpler and allows both sensitive and specific isolation of a pigmented lesion from the surrounding skin.

The algorithm takes as input the image to be partitioned into lesion and backgound, a rectangular subset of the image containing only the lesion, and another rectangular subset containing only an area of the background. All pixels touching the rectangular subset containing lesions are considered for inclusion in the lesion. The nearest neighbors of the included pixels then have their nearest neighbors tested for inclusion in the lesion, etc., until the rectangle has been "grown out" to include the whole lesion.

The decision to include a given pixel in the lesion is made if the normalized histogram for the part of the region processed so far has a greater number of pixels with that same grey level than the background does. The pixel must also be adjacent to at least one pixel which is already considered to be part of the lesion (considering only connected regions). The algorithm is as follows:

Step I: Acquire two subimages, one consisting of pixels lying entirely within the lesion, the other consisting entirely of the background.

Step II: Compute the histogram of the lesion subimage.

Step III: Compute the histogram of the background subimage.

Step IV: Compute a list of coordinates of the pixels which will constitute the initial border of the lesion subimage.

Step V: For each pixel in the list (first generated in Step IV) check each of its 8-nearest neighbors (ignoring those that already belong to the lesion) to see whether the neighbor ought to belong to the lesion. If for the grey level of the pixel being checked the percentage of the lesion's current normalized histogram is greater than the background's current histogram,

then

A. assign the pixel to the lesion:

1) add it to the histogram of the lesion and recompute the normalized histogram;

2) add the pixel's coordinates to the list (started in step IV) of pixels to be searched for neighbors belonging to the region.

otherwise

B. assign the pixel to the background:

1) add it to the histogram of the background and recompute the normalized histogram of the background;

Step VI: When the list of boundary pixels (pixels to be checked for inclusion in the lesion) is exhausted, expansion of the lesion subimage stops.

Since the algorithm continually changes the histograms of the lesion and background, called an "adaptive histogram" method. The technique is analogous to the region growing method (Rosenfeld & Kak, 1982) which tests small groups of points called cells, for inclusion in an object, and accepts them if their textures resemble that of the object sufficiently closely. But it appears to be distinct from other boundary finding or region growing algorithms (Zucker, 1976).

Chapter 7

PROCEDURES AND METHODS:

3D IMAGE RECONSTRUCTION FROM A LIMITED NUMBER OF VIEWS

Image Reconstruction: An Overview

The problem of reconstructing three-dimensional objects from a set of two dimensional (2D) projected images has been dealt with in many ways in different fields. The process of reconstruction creates an image from projections of the three dimensional (3D) object obtained onto a two-dimensional surface such as film. Thus, the reconstruction problem is: given a subset of all possible projections of an object, estimate its internal density distribution. A 3D reconstruction problem can usually be reduced to a 2D reconstruction problem. This is accomplished by using projection data taken across parallel planar cross-sections of the object, and stacking the reconstructed two dimensional cross-sections in the third dimension. But the problem becomes critical when the projection data are very limited (Herman, 1980). The reconstruction algorithm takes projection data as input to estimate the original structure based on the given data (projections observed from different angles). In continuous space, the projections are, to a first approximation, simply time integrals of the image (object), measured for different ray positions and angles. (The set of projections is also known as Radon transform of the image, named after the originator of the principle). In practice, however, discrete measurements are made, and the solution to the problem can be formulated variously as convolving, backprojecting and summing the given projection data; completing the corresponding Fourier space, solving a set of simultaneous equations; optimal estimation of the image satisfying the constraints imposed by the given projections, etc. Each of these methods has its own advantages and disadvantages which determines its suitability to a particular reconstruction problem. In the case of limited data, the reconstruction is very poor, if proper numerical methods are not adopted during the reconstruction procedure. The usual methods of reconstruction from projection such as summation, Fourier transform, and other algebraic formulations cannot give satisfactory results without major modifications in the procedures, and often fail even then.

Summation is the simplest algorithm for reconstruction. The principle underlying this technique is the same as in classical tomography: the density of each point is estimated by adding all the ray sums (estimated time integrals of the densities) of all the rays through the point. The planar version of the summation method was discovered in electron microscopy by Vainshtein (1970) and Gordon et al. (1970). Because the projections are spread back across the reconstruction plane this method is also called "back projection" (Crowther et al., 1970). (See also Gaarder & Herman, 1972). The accuracy of the summation method is limited as it blurs out the sharp features of the original image.

In the Fourier transform method the projections are transformed into Fourier space to obtain some of the values of the full Fourier spectrum of the unknown image. Each projection yields values on a radial line in Fourier space. Other unknown values of the full Fourier spectrum are obtained by interpolation from the values at the known radial lines. After interpolation the reconstruction is obtained by taking the inverse Fourier transform. The Fourier transform method was first discussed by Bracewell (1958) for the reconstruction of brightness distributions from strip integrals in radio astronomy, and suggested by DeRosier & Klug (1968) for the reconstruction of 3D structures from electron micrographs. Crowther et al. (1970) and DeRosier (1971) discussed the mathematical aspects of the approach with examples showing electron micrograph reconstructions. The problems associated with this method are the computational difficulties of interpolation and selection of a cutoff to the (infinite) Fourier plane. Any type of interpolation procedure cannot provide exact values of Fourier components, even within the desired resolution.

Radon (1917) discussed the reconstruction problem analytically in the form of a set of integral equations and presented an analytical solution. In this method the relation between the image and its projections is expressed by a set of integral equations which are then solved analytically. The image elements are estimated based on the analytical solution. Several methods of numerically evaluating the integrals to estimate the analytical solution are available. These include the methods of Bracewell & Riddle (1967), Ramachandran & Lakshminarayan (1971a,b), Gilbert (1972a,b), and Vainshtein & Orlov (1972). Different approximations are used in these methods which yield different estimates of the structure to be reconstructed. Because this method of reconstruction is simply a numerical method for evaluating the analytical solution of the integral equations, the numerical implementaion may yield different results in practice than what are expected in theory.

In series expansion methods it is assumed that an image, which is to be reconstructed, can be sufficiently approximated by a linear combination of some pre-determined basis images. The unknown coefficients in this linear combination are estimated from the equations obtained by expressing the projections of the unknown image as linear combinations of the projections of basis images. Thus, for example, if one assumes that the object lies within some square region and can be approximated by combining identical little square regions, then he can call the little square region (pixel) as a basis function. The digitized version of the image can now be obtained by defining average values of the image in the little squares. Series expansion is a feasible approach to the reconstruction problem because the projection operation is linear. Gordon et al. (1970) proposed Algebraic Reconstruction Techniques (ART) which are iterative methods of this class. Herman et al. (1973) proved that unconstrained additive ART converges to the solution with the smallest variance. With noisy projections, images produced by ART in successive iterations are, at first, better and better estimates of the image to be reconstructed, but then become progressively worse. A workable convergence criterion based on the reconstruction error was proposed by Herman et al. (1973). (See also Gordon, 1974).

The selection of a particular technique and its performance relative to others depends on the object, the number and angles of projections, and how the projection data are collected.

In this thesis, a distinctive approach to the reconstruction problem has been presented. It involves two stages. First, a limited-view reconstruction by one of the above algorithms (which are linear) is carried out. Second, the 2D point spread function of that algorithm is deconvolved.

For the reconstruction problem in nevoscopy, one has a few images of the transilluminated dermal object: The prototype nevoscope gives only three images of the dermal object: one diagonal image at 45 deg., one vertical image at 90 deg., and one glancing image at 180 deg. Thus, though only three views are available to derive only three projections per sectional reconstruction, a wide angular range of 135 deg is covered for imaging the lesion. This makes the interpolations discussed below perform better. The problem of reconstructing an image of the object from only three views is thus an extreme case in limited-view reconstruction. Some of the recently developed limited-view CT reconstruction algorithms were chosen and modified for this application (Dhawan et al., 1984a,b; Dhawan et al., 1985; Rangayyan & Gordon, 1982; Rangayyan & Gordon, 1983a,b; Gordon & Rangayyan, 1983; Gordon et al., 1970). These techniques and algorithms will now be discussed.

The Algebraic Reconstruction Technique

The Algebraic Reconstruction Techninique (ART) is an iterative procedure which starts with an initial estimate of the image and updates the pixels so as to satisfy the given projection data. A brief description of ART from a digital image processing point of view is given below (cf. Gordon & Herman, 1974 a,b).

The image reconstruction problem can be posed as follows: Given a set of projections R(l,k) at angles θ_ℓ, for a number of views l= 1...P, each view having a number of rays k= -K...K, compute an image p(i,j) such that the raysums S(l,k) of {p(i,j)} are as close to R(l,k) as possible. The ART algorithm updates pixels belonging to individual rays of a view to meet the raysum criterion as

$$p^{q+1}(i,j) = p^{q}(i,j) + (R(l,k)-S^{q}(l,k)/N(l,k) \tag{1}$$

where N(l,k) is the number of pixels in the ray (l,k) and q refers to the iteration number. The above operation, performed over all views, constitutes one cycle, and a number of such cycles will have to be executed before all raysum constraints are met. A suitable convergence criterion can be set up based on the error of reconstruction defined as

$$E^{q} = \sum_{l}\sum_{k} |R(l,k)-S^{q}(l,k)| / \sum_{l}\sum_{k} R(l,k)$$

$$\tag{2}$$

Equation (1) represents additive ART, so called because the correction applied is additive. As this can lead to negative values being assigned to pixels, an additional constraint is essential. This is done by setting a pixel to zero whenever the value given by equation (1) is negative.

The multiplicative version of ART is defined as

$$p^{q+1}(i,j) = p^{q}(i,j)R(l,k)/S^{q}(l,k) \tag{3}$$

which has the advantage that negative pixel values are not encountered. The initial values of $p0(i,j)$ must all be set positive.

The raywidth at different projection angles is defined as

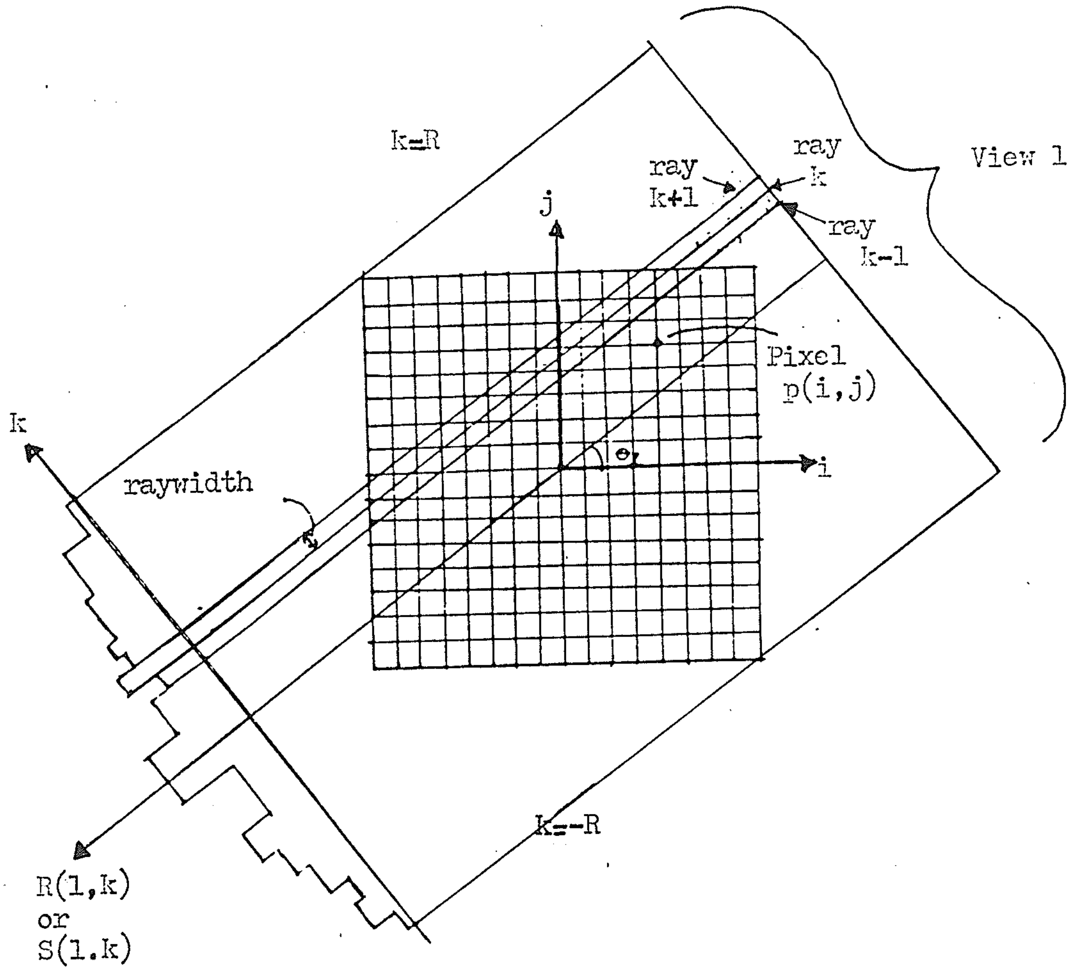
$$W(l) = \max\{|\cos(\theta_l)|, |\sin(\theta_l)|\} \tag{4}$$

so that each ray crosses one and only one pixel per row or column. This way, a plot of $N(l,k)$ versus k will be smooth.

The SPARTAF Algorithm

The SPARTAF algorithm (Streak Preventive ART via Adaptive Filtering) prevents streaks and other artifacts that arise in reconstructions made from very few views (Rangaraj & Gordon, 1982). A

Figure 7.1. Ray geometry of projections.



brief description of the algorithm follows. Let a(i,j) denote a linear combination of the immediate neighbors of p(i,j) which belong to the rays to the left and right of the ray passing through p(i,j) (see Figure 7.1). In order to prevent the occurrence of streaks, one can iteratively minimize

$$\sum \{ p(i,j) - a(i,j) \}^2 \tag{5}$$

with respect to p(i,j), subject to the constraint

$$\sum p(i,j) = R(1,k) \tag{6}$$

where the summations are over all pixels belonging to ray (1,k). Applying Lagrange's method of undetermined multipliers, the following algorithm is obtained:

$$p^{q+1}(i,j) = a(i,j) + (R(1,k) - \sum a(i,j))/N(1,k) \tag{7}$$

where the summation is over all a(i,j) associated with the pixels of ray (1,k). The multiplicative version of the above equation can be expressed as

$$p^{q+1}(i,j) = a(i,j)R(1,k)/\sum a(i,j) \tag{8}$$

The important feature of the above algorithm is that it tends to keep the difference between adjacent ray pixels to a minimum, satisfying tha raysum constraint at the same time (Equation 6).

In order to avoid loss of details, the neighborhood factor a(i,j) has to be appropriately defined at each location using the information available at that cycle in such a way as to both prevent streaking and avoid indiscriminate blurring of features. For this three contrast measures are defined to help determine a(i,j):

$$T1=[|p^{-1}(i,j)-p(i,j)|+|p^{+1}(i,j)-p(i,j)|]/[p^{-1}(i,j)+p^{+1}(i,j)+2p(i,j)]$$

$$T2=[|l(i,j)-p(i,j)|+|r(i,j)-p(i,j)|]/[l(i,j)+r(i,j)+2p(i,j)]$$

$$T3=|l(i,j)-r(i,j)|/[l(i,j)+r(i,j)] \tag{9}$$

where p(i,j) is a pixel belonging to ray (l,k), $p^{-1}(i,j)$ and $p^{+1}(i,j)$ are neighbors of p(i,j) along the ray (l,k), l(i,j) is the average of neighbors of p(i,j) belonging to ray (l,k+1), and r(i,j) is the average of neighbors of p(i,j) belonging to ray (l,k-1) (see Figure 7.1). The pixels used are in the 8-neighborhood of p(i,j) when the raywidth is defined as in Equation (4).

Three options are provided in the definition of a(i,j) using the above contrast measures:

```
if ($T2 > W1$) and ($T1 < W2$)
 1.then {possible streak}
  if $T3 > W3$
   1a. then {edge}
    $a(i,j)$ = nearer of $l(i,j)$ and $r(i,j)$ to $p(i,j)$
    1b. else {streak}
      $a(i,j) = 0.5(l(i,j)+r(i,j))$
  2. else {proceed with ART}
    $a(i,j) = p^{q}(i,j)$
```

where W1, W2, and W3 are thresholds defined for the three contrast measures to detect streaks and edges in the reconstructed image. By setting the contrast thresholds to the maximum possible value of 1, the algorithm may be forced to follow regular ART.

Algorithms For Limited-View CT: An Overview

In many applications of Computed Tomography, it may not be possible to acquire projection data at all angles as required by most of the commonly used algorithms. In such a "limited-view" or "limited-data" situation one faces an ill-posed problem in attempting to reconstruct an image from an incomplete set of projections. Many techniques have been proposed to solve this problem employing diverse theories such as signal recovery, image restoration, constrained deconvolution, constrained optimization, etc.

When a small number of projections is used, the reconstructed image suffers from severe artifacts such as streaking and geometric distortion. Further, the solution obtained will be non-unique. For instance, if a basis function approach is used, the set of simultaneous equations to be solved will be highly underdetermined

(Gordon, 1973; Smith et al., 1977; Smith el al., 1982; Snyder & Cox, 1977; Duerinckx & Macovski, 1980). The lack of information, however, may be partially made up in fact by adequate a priori knowledge about the image or object. Such knowledge may be used to impose constraints on the solutions or the reconstruction procedure. The use of more specific knowledge about the object, when available, regularizes the ill-posed problem but makes the reconstruction procedure object-dependent and may not be generally applicable directly to other cases. On the other hand, constraints like positivity may be directly applied to almost all cases, but they may not help much in improving the image.

A large number of algorithms available for limited-view CT are based on Image-Fourier Space Methods which use the projection (Fourier slice) theorem. The limited-angle problem may be characterized by a cone of missing information in Fourier space corresponding to the angular region (in the object domain) where no projections are available. This comes about because the Fourier slice theorem shows that the one-dimensional (1D) Fourier transform of a 1D projection of a two-dimensional (2D) image is simply a radial slice of the 2D Fourier transform of the 2D image at a corresponding angle. Many authors have proposed iterative image Fourier-space revision methods for extrapolation of known information into the missing cone based on the methods of Gerchberg, (1974) and Papoulis, (1975) for super-resolution and signal recovery. Tam & Perez-Mendez (1981a,b) and Sato et al. (1981) have proposed iterative image revision between image and Fourier spaces. Byrne & Fitzgerald (1982) applied these methods in tomography for reconstruction from partial data. More specific a priori information, if available, may be applied as constraints to the revised image estimate and its transform. It should be noted that such a procedure may be highly object-dependent and non-linear.

Tam (1982) has suggested the use of multispectral imaging in limited-angle reconstruction. This method separates out individual component substances in the object through multiple energy scans and also uses non-linear constraints in reconstruction. The use of orthogonal expansion algorithms has also been suggested by Inouye (1979) to estimate the missing cone components in the Fourier space. This, in fact, is another method for extrapolating the available data in the Fourier space employing the property of analytic continuity. The method of orthogonal expansion, though giving the exact limited-data reconstruction in theory, is very sensitive to the presence of noise as well as the numerical inaccuracies in the given projection data. The same problems are encountered with almost every image-Fourier space revision method. Also, they are good for parallel ray geometry only.

When fan beam geometry is used, the missing projections do not correspond to missing slices in the Fourier domain. Nassi et al. (1982) proposed an iterative reconstruction-reprojection algorithm which estimates the missing views at each iteration based on reprojection. The method uses convolution-backprojection for image reconstruction. The authors call it a "software substitute of scanning process". In the process of reprojection specific a priori information is used as a constraint to the estimation of the revised image and its reprojections. This makes the iterative reconstruction reprojection method object-dependent. The iterative convolution backprojection algorithm (Medoff et al., 1983) and iterative reconstruction-reprojection techniques presented by Heffernan & Robb (1983) are other examples of image-projection space iterative revision methods used in limited-view CT.

In order to reduce the streaking artifacts and geometric distortion to reveal extra details from limited-view reconstruction, one requires a safe inversion technique (one that gives a stable and unique solution). The restoration algorithms may be based on linear or non-linear methods incorporating constraints based on a priori information about the object. The linear restoration methods include "geometric deconvolution" (Gordon & Rangayyan, 1983; Gordon, Dhawan & Rangayyan, 1984) and the 2D deconvolution method (Dhawan et al., 1984a, b, c). I shall discuss these methods in detail in the next sections. The non-linear restoration methods apply to the reconstruction of non-negative objects. One of these methods is Burg's maximum entropy method (Burg, 1975). The Burg method is a non-iterative method that provides a positive reconstruction consistent with the data. The maximum entropy method may be used as an inversion technique in both image restoration (Frieden, 1972) and image reconstruction (Wernecke & D'Addario, 1977; Gull & Daniell, 1978). Also, it may be applied iteratively to CT (Minerbo, 1979; Kemp, 1980; Burch et al., 1983). Burch et al. (1983) assumed the limited-view reconstruction as available data and the convolution of an original object with a spatially variant point spread function with the addition of noise. The solution is obtained by maximizing the entropy function of a set of numbers which represent the pixel values of the image. The a priori information is applied to reach the unique feasible solution that fits the actual data. Since the positivity constraint is inherent in entropy maximization, the maximum entropy restoration methods may provide lower noise (Burch et al., 1983). (See also Gordon et al., 1984; and Gordon, 1985). There may be some variations in defining the entropy, but the underlining principle of maximizing a function incorporating some constraints remains the same in all the algorithms (Kikuchi & Soffer, 1979; Jaynes, 1982; Frieden, 1980a, b). The main disadvantages of these methods are they are very time consuming. That makes processing of large size images computationally very expensive.

The Baysian approach also allows one to incorporate a priori information to estimate the missing components of the unknown source functions (Hanson & Wecksung, 1983). The Baysian maximum a posteriori (MAP) reconstruction method presented by Hanson & Wecksung (1983) is equivalent to a minimum-variance linear estimate with non-stationary mean and covariance ensemble characterizations. The authors claim that if the a priori information is specific enough, reasonable estimates of the null-space components of the source function can be obtained, therby reducing the artifacts in the reconstruction. Burg's maximum entropy method has also been aplied to computed tomography problems by Byrne et al. (1983) in a non-linear restoration model and compared with other linear restoration models extending the minimum-energy band limited extrapolation methods based on Gerchberg-Papoulis methods.

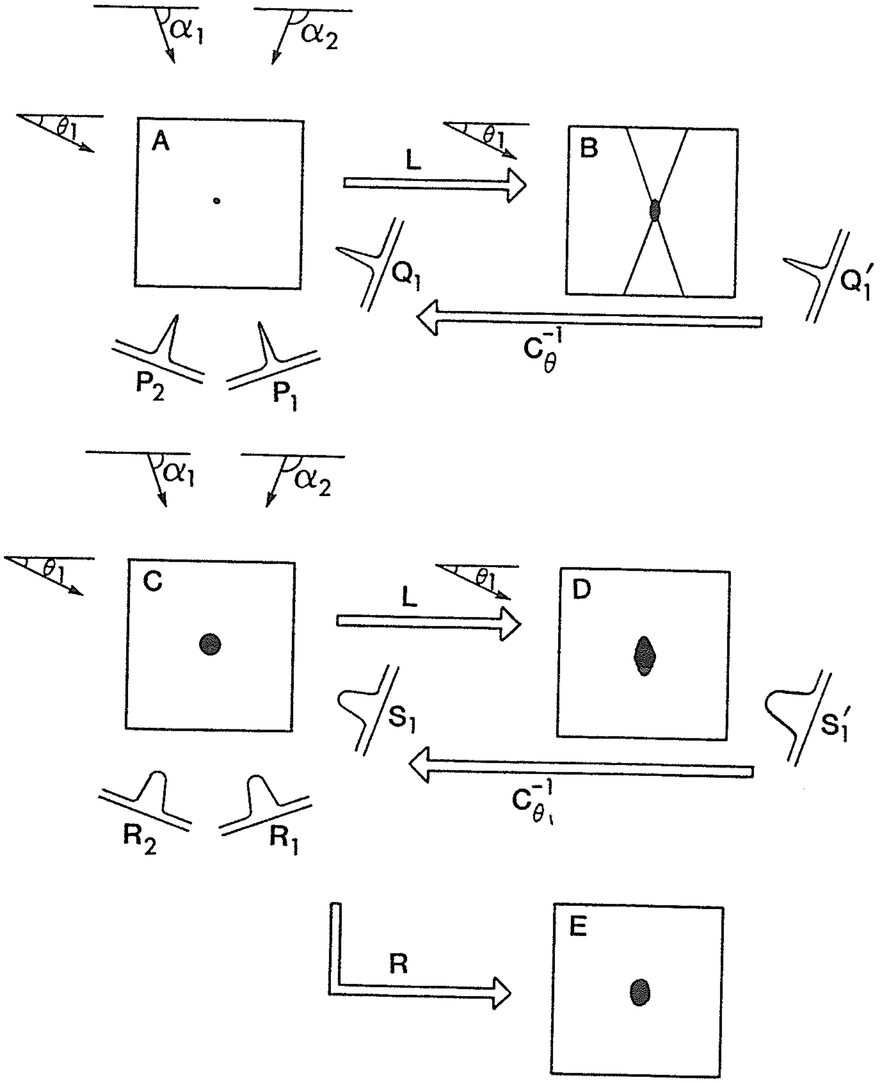
The other approach for the optimum estimator is using the principle of maximum likelihood which becomes one of the entropy forms under different conditions of prior knowledge. Frieden (1983) has discussed various maximum likelihood (ML) estimators such as maximum Shanon entropy, maximum Kikuchi-Soffer entropy, and maximum Burg entropy.

Geometric Deconvolution

Images reconstructed from a limited number of projections suffer from a systematic geometric distortion along the direction of the rays used, and poor contrast at angles not used (Gordon, 1973; Gordon & Rangayyan, 1983). This distortion appears because of the 2D point spread function of the reconstruction process. By applying the projection theorem, the problem of removing this distortion may be reduced to that of estimating the one-dimensional spread function (in the projection domain) and deconvolving projections computed for a complementary set of new angles from the initial reconstruction (Gordon & Rangayyan, 1983). An initial reconstruction is obtained using the available limited number of projections with a linear reconstruction algorithm such as multiplicative ART.

The procedure of geometric deconvolution can be explained as follows: I first reconstruct a known test pattern A (Figure 7.2) using projection data P1,P2,...,Pl from a limited set of known views at

Figure 7.2. A schematic representation of geometric deconvolution.



angles $\alpha 1$, $\alpha 2, \ldots, \alpha l$. The reconstruction is performed using a linear computed tomography algorithm L. If the test pattern is a point, as shown in Figure 7.2, I can obtain the point spread function of the reconstruction process, as in image B. A projection Q1' of B at a new angle $\theta 1$ is now calculated. The projection Q1 of test image A is also calculated. I can now compute $C_{\theta_1}^{-1}$, the function that deconvolves Q1', to give Q1. Next I reconstruct the unknown image C, using the linear algorithm L and its known projections R1, R2,...,Rl at angles $\alpha 1$, $\alpha 2, \ldots, \alpha l$ obtaining image D, with the characteristic geometric distortion. A projection S1' at the angle $\theta 1$, for which I have no measured projection, is calculated from image D. S1' is then deconvolved using $C_{\theta_1}^{-1}$, to obtain an estimated projection S1, at angle $\theta 1$, of the unknown object C. This operation is performed for the whole complementary set of angles chosen. The combined set of projections is then used for reconstruction by another computed tomography algorithm R. Algorithm R may be nonlinear, object dependent, and may incorporate a priori information. It could also give different weights to the measured and estimated projections. The result is image E, a reconstruction of the unknown object with some of the geometric distortion of image D removed. Geometric deconvolution allows any combination of any linear and nonlinear computed tomography algorithms. In the examples given in this thesis, I used multiplicative ART for both reconstruction stages in geometric deconvolution.

Image Restoration by 2D Deconvolution in Limited-View CT

The geometric distortion and streaking artifacts are due to the 2D point spread function of the reconstruction process for the given set of projection angles (Gordon, 1973). I have the following problem: Given a distorted reconstruction, compute a distortion-free and noise-free reconstruction of the image. A direct approach would be to deconvolve the 2D point spread function (Gordon, 1972). This may be done via Wiener filtering. Klug & Crowther (1972) and King et al. (1983) have pointed out the possibility of application of the Wiener filter in image reconstruction from projections but have not explained how it may be implemented in limited-view reconstruction tomography. If I restrict myself to a linear reconstruction algorithm, the problem reduces to that of estimating the point spread function of the system for a given set of projections and performing an appropriate deconvolution. I will show that geometric distortion and streaking artifacts in limited-view reconstruction can be reduced substantially by 2D Wiener deconvolution performed using a priori knowledge derived from the known projections. I have used the unconstrained multiplicative Algebraic Reconstruction Technique (ART) (Gordon et al., 1974a, b) for all the images to obtain their limited-view reconstructions which were then used in 2D Wiener deconvolution. (These results are discussed in the next chapter.) The unconstrained multiplicative ART is an iterative reconstruction algorithm. It is "quasi-linear" because of its slightly spatially variant point spread function (Andrews & Hunt, 1977). For now I will treat the problem as if this reconstruction algorithm were strictly linear.

TWO-DIMENSIONAL DECONVOLUTION METHOD

I perform 2D deconvolution on the limited-view reconstruction of an unknown image via Wiener filtering to remove distortion and obtain a noise-free restored image. I make use of a priori knowledge derived from the given projections in the formulation of the Wiener filter.

Assume that the reconstruction process produces a degraded image g(x,y) from the given projections of an unknown image f(x,y), modeled (Gonzalez & Wintz, 1977) as:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y) \tag{10}$$

where h(x,y) is the two-dimensional (2D) point spread function of the reconstruction process and n(x,y) is the noise introduced. The "*" designates 2D deconvolution. The image restoration problem can be written as that of obtaining an approximation $\hat{f}(x,y)$ to f(x,y), given g(x,y) and a knowledge of h(x,y) and n(x,y). The Wiener filter estimate gives $\hat{F}(u,v)$, the Fourier transform of $\hat{f}(x,y)$, as:

$$\hat{F}(u,v) = [|H(u,v)|^2 / (|H(u,v)|^2 + W(u,v))] G(u,v) / H(u,v) \tag{11}$$

where G(u,v) is the Fourier transform of g(x,y), H(u,v) is the system transfer function, and W(u,v) is the noise-to-signal ratio function in the frequency domain. Wiener filtering may be used if I can estimate H(u,v) and W(u,v).

The system transfer function H(u,v) is obtained from the reconstruction of a point image by the chosen reconstruction algorithm for the given set of angles. These angles are the same as those for which the projections of the unknown image are available. I compute the 2D Fourier spectra of the point image and its reconstruction, say J(u,v) and K(u,v), respectively. The 2D system transfer function is now computed as:

$$H(u,v) = K(u,v)/J(u,v) \tag{12}$$

where the noise is assumed to be zero.

One should expect K(u,v) to be the same as the system function H(u,v) (when J(u,v) is unity for an ideal impulse). But the system transfer function differs from K(u,v) because of the finite extent of the so called point image used in image processing, which consists of a single pixel. The Fourier spectrum of a pixel image constructed by assigning a higher density level to a single pixel than the background is the Fourier spectrum of a square function because of the finite dimensions of the pixel.

The noise-to-signal ratio function W(u,v) requires an estimate of both the noise N(u,v) and the signal S(u,v). The signal spectrum S(u,v) is estimated in the following way. I obtain the 2D Fourier transform of the limited-view reconstruction of the unknown image by the chosen reconstruction algorithm from the known projections. From the Fourier slice theorem, I know that the one-dimensional (1D) Fourier transform of a projection corresponds to a radial slice of the

2D Fourier transform of the image, at a corresponding angle. I compute the average of the known projections, and take its 1D Fourier transform. From this I obtain the average 1D Fourier spectrum in the projection domain. I fill the 2D Fourier spectrum of the limited-view reconstruction using this average "slice" to compute an image spectrum model. Thus the use of a priori information is to roughly fill in Fourier space with this average spectrum. (Other interpolation schemes, such as those designed for the simpler Fourier algorithm, could be used: Lutz, 1975.) Let this filled-in spectrum be the estimate of S(u,v).

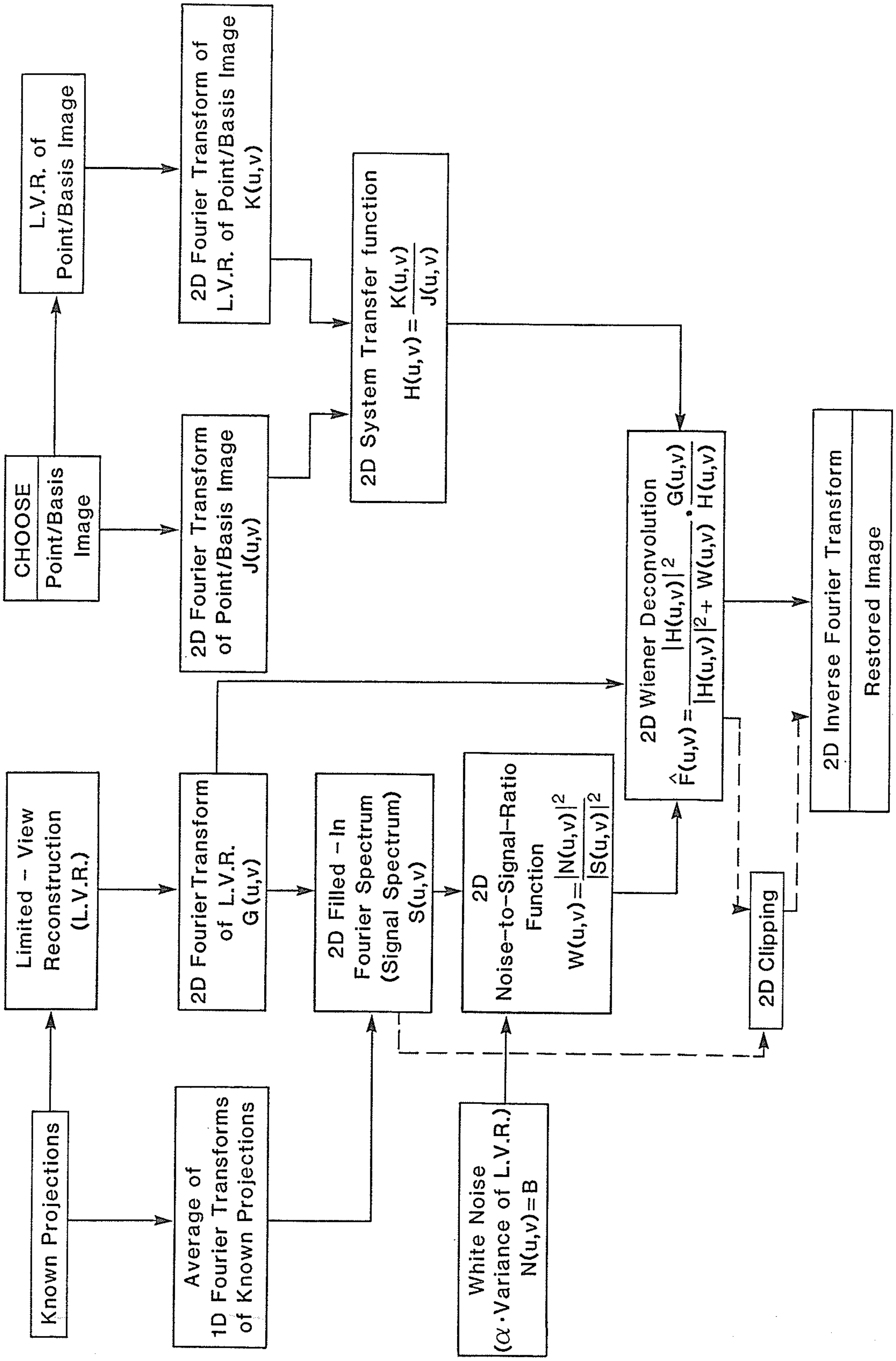
I assume a white noise spectrum N(u,v), introduced in the reconstruction process, with a total energy equal to a small percentage of the total energy in the image and to be a constant B independent of u and v. A 2D noise-to-signal-ratio function is now computed as:

$$W(u,v) = |N(u,v)|^2 / |S(u,v)|^2 \tag{13}$$

This is used in the Wiener filter Equation (11).

The restored image may be obtained by taking the inverse Fourier transform of $\hat{F}(u,v)$. I have found from the experiments that the choice of the value of N(u,v)=B is very critical in achieving optimum restoration. Lack of knowledge of the 2D noise spectrum and values of B other than the one that gives optimum restoration cause some excessive values in the Fourier spectrum the $\hat{F}(u,v)$. These values may be clipped using S(u,v) as a 2D magnitude clipping function. I take

Figure 7.3. A schematic flow-diagram of 2D Wiener deconvolution.


Known Projections
Limited - View Reconstruction (L.V.R.)
Average of 1D Fourier Transforms of Known Projections
CHOOSE
Point/Basis Image
L.V.R. of Point/Basis Image
2D Fourier Transform of L.V.R. of Point/Basis Image K(u,v)
2D Fourier Transform of Point/Basis Image J(u,v)
2D System Transfer function
$H(u,v)=\frac{K(u,v)}{J(u,v)}$
2D FourierTransform of L.V.R. G(u,v)
2D Filled - In Fourier Spectrum (Signal Spectrum) S(u,v)
2D Noise-to-Signal-Ratio Function
$W(u,v)=\frac{|N(u,v)|^2}{|S(u,v)|^2}$
White Noise (α·Variance of L.V.R.) N(u,v)=B
2D Wiener Deconvolution
$\hat{F}(u,v)=\frac{|H(u,v)|^2}{|H(u,v)|^2+W(u,v)}\cdot\frac{G(u,v)}{H(u,v)}$
2D Clipping
2D Inverse Fourier Transform
Restored Image

the inverse Fourier transform of the clipped spectrum $\hat{F}(u,v)$ to yield the restored image. Fast Fourier transform (FFT) techniques are used throughout the procedure.

Further, the system transfer function may also be obtained from a basis image b(x,y) based on some a priori information about the geometrical shape of the object to be reconstructed (Burch et al., 1983; Andrews & Hunt, 1977). For example, the basis image may be a disc image or an ellipse image for the objects having cellular structures. The function H(u,v) can be obtained as follows:

$$H(u,v) = D(u,v)/B(u,v) \tag{14}$$

where B(u,v) is the Fourier spectrum of basis image b(x,y), and D(u,v) is the Fourier spectrum of the reconstruction of the basis image (assumed to be spatially invarient). The noise has been assumed to be zero here as in Eq. 12.

Chapter 8

RESULTS AND DISCUSSION

3D Reconstruction: Simulation

To test the ability to reconstruct the third dimension using computed tomography from only three views, I used the simulated patterns shown in Figures 8.1 and 8.3. Each pattern, idealizing a vertical cross section through a nevus of a given thickness and elevation as an ellipse, was reconstructed using the SPARTAF algorithm, multiplicative ART, multiplicative ART with geometric deconvolution, and multiplicative ART followed by Wiener deconvolution. Although the net angular range (45 deg-180 deg) is not so narrow, the number of views is so small as to leave significant anisotropic resolution (cf. Jaman et al., 1985). The original set of angles used in SPARTAF, and multiplicative ART was 45, 90, and 180 deg. In geometrical deconvolution the angles used for the complementary set were 22.5, 67.5, 112.5, 157.5 deg giving eight views equally spaced by 22.5 deg. To remove the geometric artifacts from the limited-view reconstruction obtained from unconstrained multiplicative ART, I used a Wiener filter incorporating _a priori_ knowledge derived from the projections obtained from the three known views. A comparison of the reconstructions obtained by various algorithms is given in this

Figure 8.1. Computer simulations of reconstructions of elliptical "nevi". (a). Test pattern of eccentricity 0.78, arbitrary densities of 20, 200 and 2 were assigned to skin, nevus and air, respectively. The densities are reversed compared to actual nevi. Reconstructions were made of the test pattern with angles of 45 deg., 90 deg. and 180 deg. The SPARTAF algorithm was used. Reconstruction is shown at the right side.

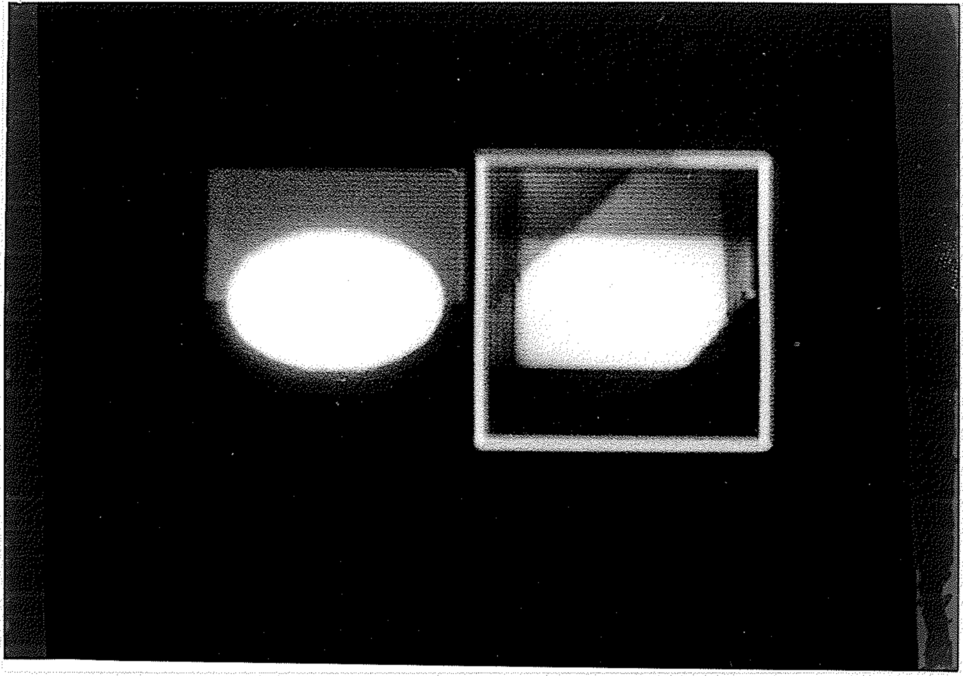


Figure 8.1(b). Thresholded reconstruction of the test pattern and its reconstruction by SPARTAF as in (a). A threshold of 110 units was applied.

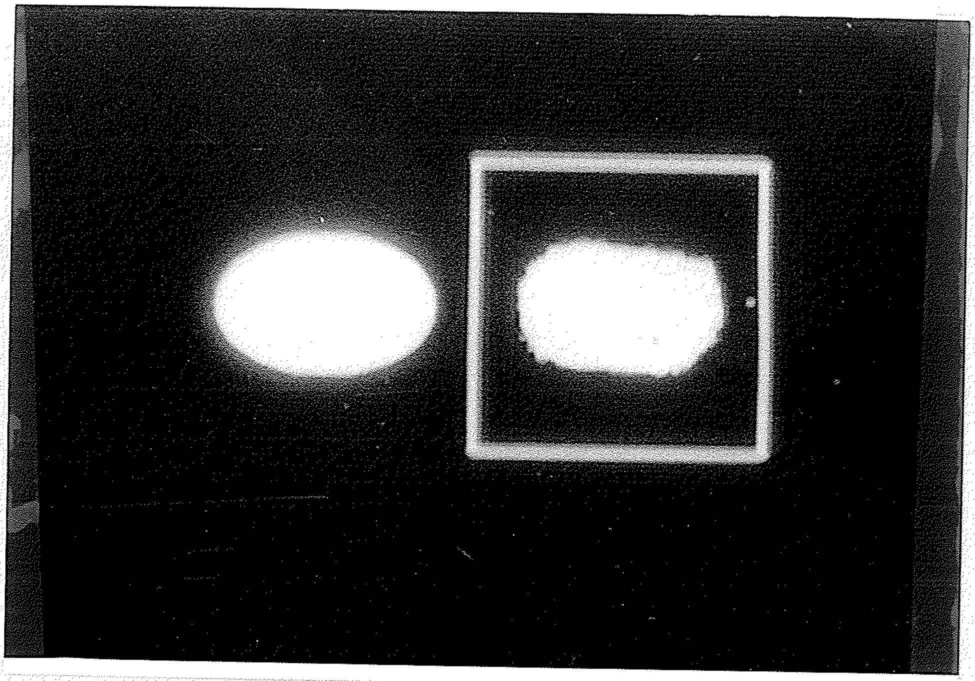
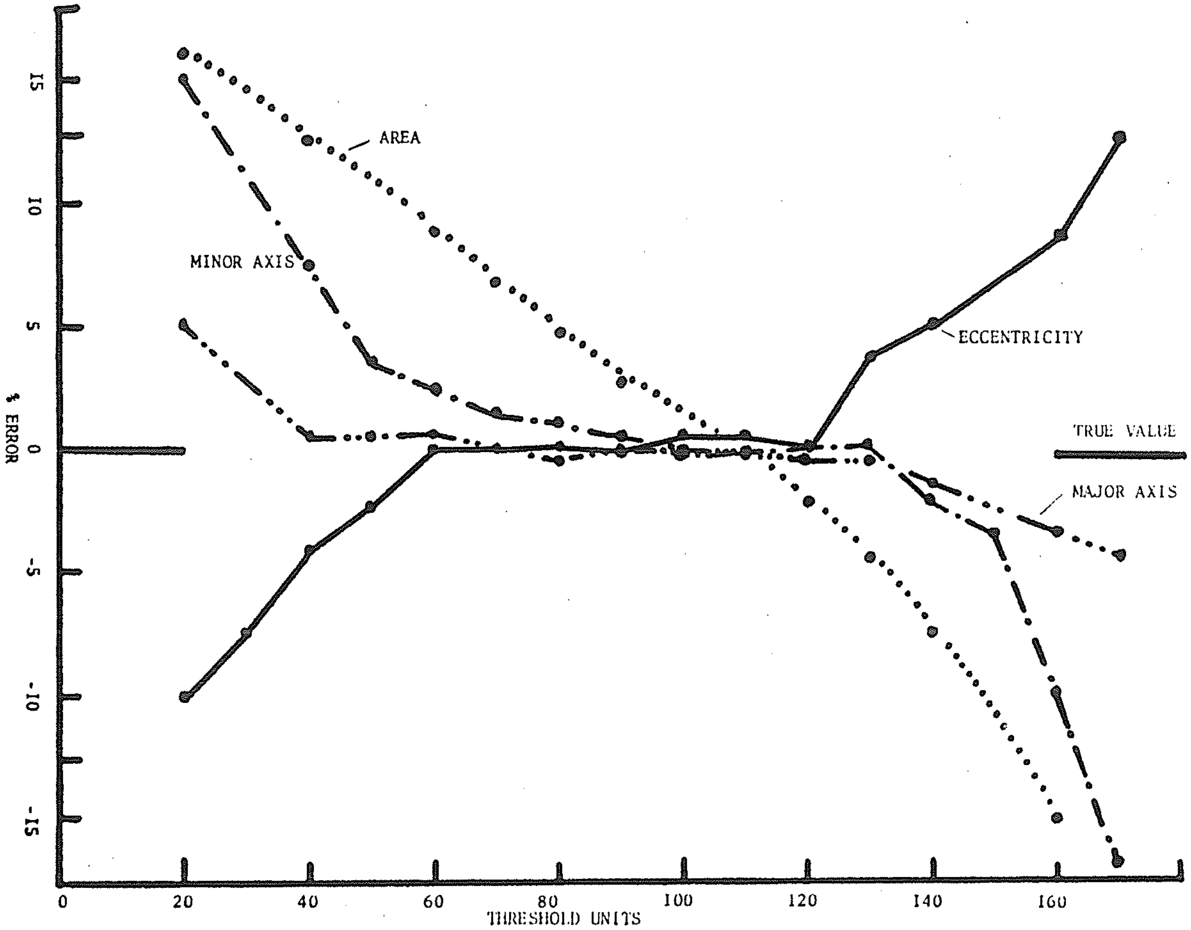


Figure 8.2. Measurements on SPARTAF reconstruction vs thesholds.



section.

I took the criterion of success of reconstruction to be a reasonably accurate estimation of the width and thickness of the nevus from the reconstructed images. To obtain such measurements from the simulations, I had to choose a threshold for contouring the reconstructions. A fit to an ellipse with horizontally and vertically oriented axes was calculated, by fitting it to a boundary trace of the thresholded reconstruction. For this, the thresholded reconstructed image was first converted into a binary image, and then the boundary was traced by chain coding (Rosenfeld & Kak, 1982). The major and minor axes were estimated by counting the number of pixels within the boundary along the x and y axes, respectively. Then, the area was estimated by counting the total number of pixels within the traced boundary.

As can be seen from Figure 8.2, the reconstructed major and minor axes of this fit to an ellipse match those of the test pattern over a broad range of threshold values. Thus the choice of a threshold, anywhere within this range, is not critical. I found that selection of a threshold was required when the SPARTAF algorithm was used. But when multiplicative ART with geometric deconvolution was used, this problem was taken care of by applying a suitable constraint (threshold from the curves shown in Figure 8.2) during the reconstruction procedure. Pixels with less than the threshold value at any stage in the reconstruction procedure were set to zero. Note that in multiplicative ART, a pixel, once set to zero, remains so.

Figure 8.3-8.5. Test patterns of different geometric structures are at extreme left. Reconstructions of the test patterns using multiplicative ART with angles of 45 deg., 90 deg. and 180 deg. are in the middle. Reconstuctions of test patterns using multiplicative ART with geometric deconvolution, where the complimentary angles fill in every 22.5 deg. are at the extreme right.

Figure 8.3.

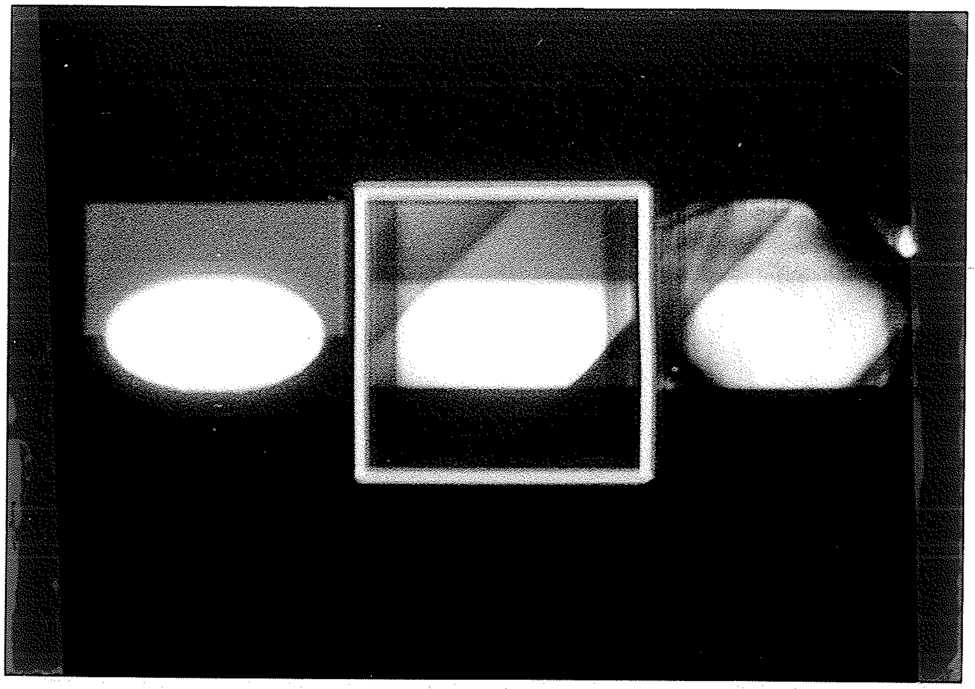


Figure 8.4. Test pattern-2 and its reconstructions by multi. ART only and multi. ART with geometric deconvolution.

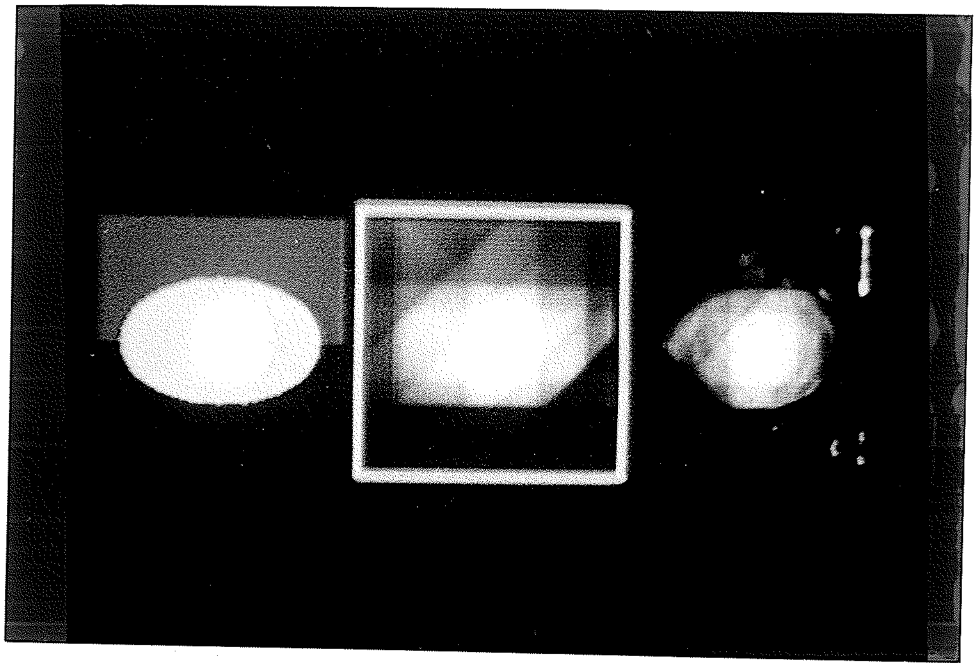
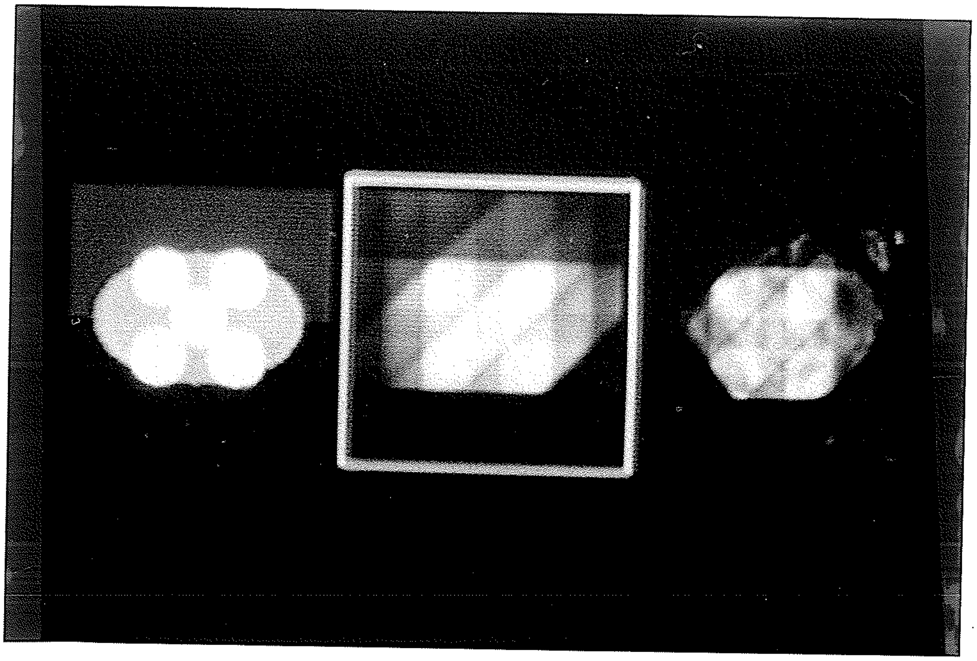


Figure 8.5. Test pattern-3 and its reconstructions by multi. ART only and multi. ART with geometric deconvolution.



I also carried out tests on a pattern simulating a section of a nevus. This test pattern, as shown in Figure 8.4, has an ellipse with discs. The discs, representing melanin structure, were assigned a density of 200 on a 0-255 grey scale. The ellipse representing the nevus was assigned a density of 170 with backgrounds of 20 and 2 representing skin and air respectively.

Multiplicative ART with geometrical deconvolution appeared to give better reconstructions than the SPARTAF and plain multiplicative ART algorithms, as is evident from Figures 8.3, 8.4 and 8.5. It also retained the internal structure of the test pattern when compared with the reconstruction by multiplicative ART only. Thus it is evident that in spite of introducing noise and blurring out the boundary of the test image, the 1D deconvolution method of the geometric deconvolution algorithm improved the reconstructed image obtained from ART.

Image Restoration by 2D Wiener Deconvolution

I first obtained the 2D point spread function of the reconstruction process by reconstructing a point image by multiplicative ART using the given set of angles. The "point image" was chosen to be a pixel of grey level value 255 in the center of a background of 25 on a 101X101 raster. I shall refer such a "point image" as a "single pixel image" in the next discussion. I observed, from a comparison of the Fourier spectrum of the test image and its reconstruction, that the phase values at radial sections corresponding

Figure 8.6. (a) Fourier spectrum of a point image. (b) Fourier spectrum of the reconstruction of the point image using multiplicative ART using eight views at 55, 65,..., 125 deg. (c) Amplitude spectrum of the system transfer function.

Figure 8.6 (a).

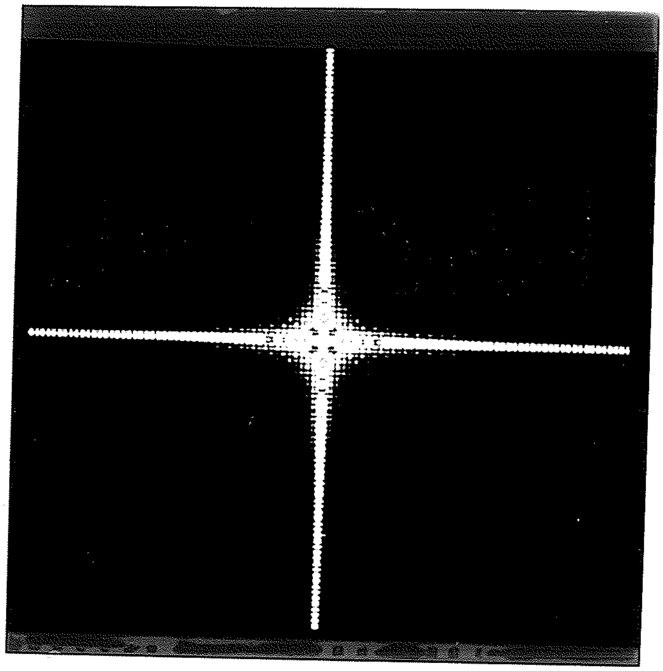


Figure 8.6 (b).

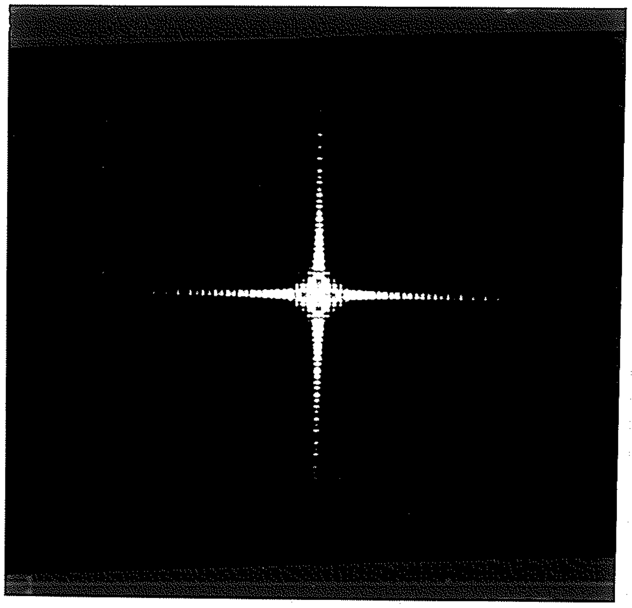
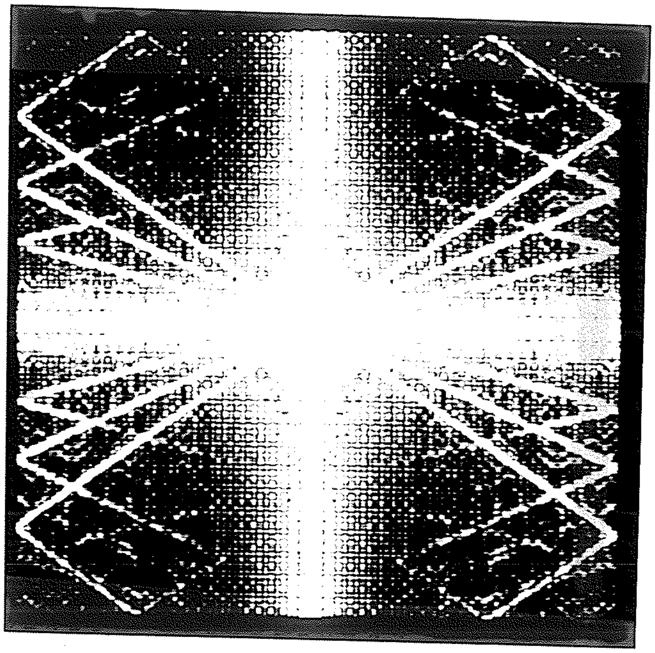


Figure 8.6 (c).



to the data projections were similar in the lower spatial frequency region. I obtained the system transfer function H(u,v) from Eq. 12 and found that the phase values at radial sections corresponding to the known projections in the low spatial frequency region were nearly equal to zero.

To illustrate these aspects, the Fourier transform of a single pixel image and its limited-view reconstruction using multiplicative ART with eight views at 55, 65, 75,..., 125 deg are shown in Figures 8.6(a) and 8.6(b). The Fourier spectrum displays were enhanced by a linear lookup transformation of pixel values to see and photograph the details. Figure 8.6(a) shows the superposition of 2D sinc patterns (the Fourier transform of square windows), because of the square shape of the pixel and the square support of the image. Thus the Fourier transform of single pixel image differs substantially from the Fourier transform of an impulse function. As the Fourier transform of the limited-view reconstruction of a single pixel image is also affected by the same components (Fig. 8.6(b)), it cannot be directly taken to be the system transfer function H(u,v). The system transfer function therefore has to be obtained from Eq.12. This is shown in Figure 8.6(c). The contribution of the square dimensions of the pixel and the square support of the image size are now removed. The distinct sinc patterns are not seen in Figure 8.6(c). Instead, the streaks, at the corresponding angles used in the reconstruction process, may be seen clearly. Thus, the system transfer function, so obtained, has the spread function of the reconstruction process for the set of angles used.

Figure 8.7. A test image.

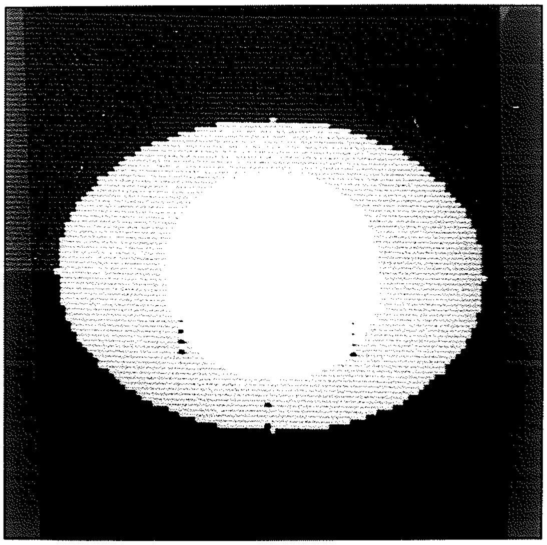


Figure 8.8. Limited-view reconstruction of the test image by multiplicative ART using eight views at 55, 65,..., 125 deg.

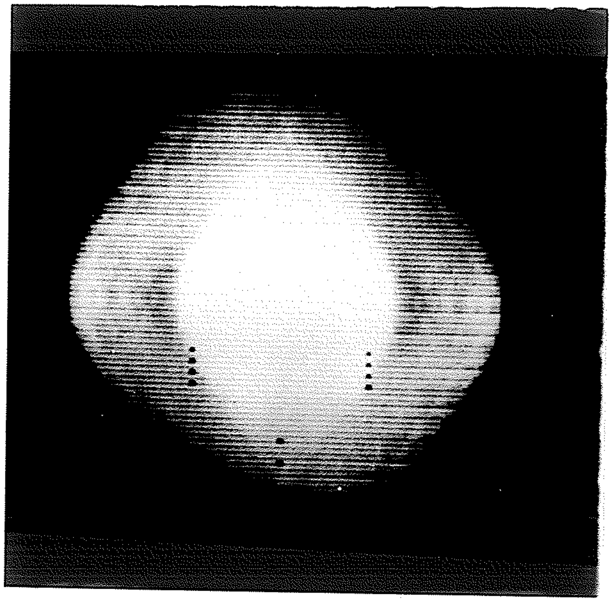


Figure 8.9. Fourier spectrum of the test image.

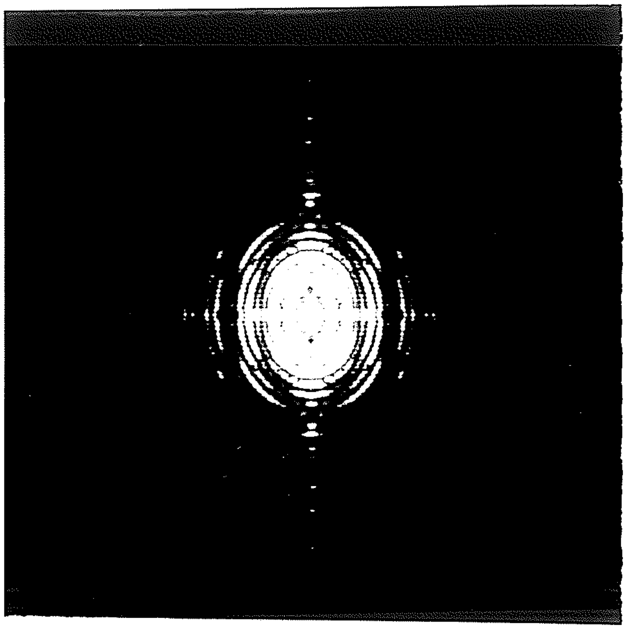
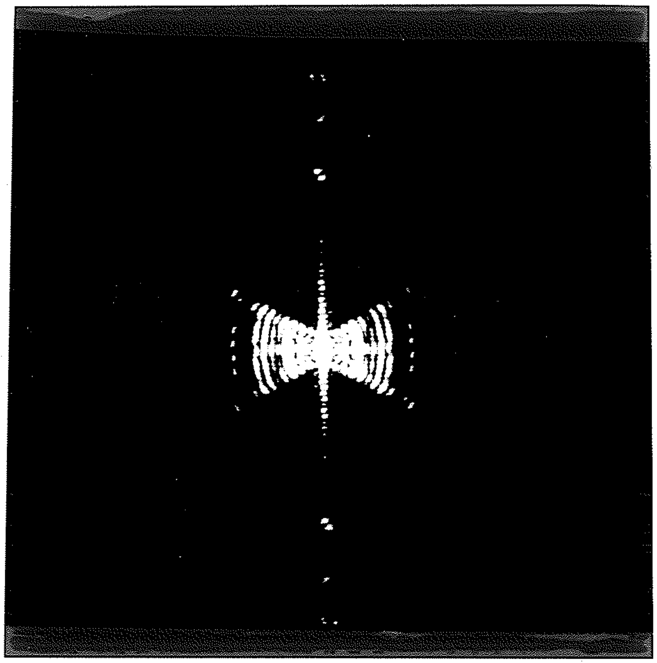


Figure 8.10. Fourier spectrum of the limited-view reconstruction shown in Figure 8.9.



I used a test image consisting of a disc inside an ellipse with the grey levels of 255 and 175 respectively on a grey level scale of 0-255. The background in the test image was assigned a grey level value of 20. The test image and its limited-view reconstruction, obtained by unconstrained multiplicative ART using eight views at 55, 65, 75, 85, 95, 105, 115, and 125 deg are shown in Figures 8.7 and 8.8 respectively. The characteristic geometric distortion in the limited-view reconstruction can be seen along the view angles used in the reconstruction process. The Fourier spectra of the test image (Figure 8.7) and its reconstruction (Figure 8.8) are shown in Figures 8.9 and 8.10. It is evident from a comparison of Figures 8.9 and 8.10 that the Fourier spectrum of the limited-view reconstruction has no significant spatial frequency information along radial lines (lines through the origin or dc value in Fourier space) corresponding to the angles not used in the reconstruction process. It should be noted that the 1D Fourier transform of the 1D projection at the angle 0 (with respect to horizontal axis) is the same as the components along the radial line inclined at (90+0) deg. with respect to horizontal axis in the 2D Fourier transform of the image.

The Fourier spectrum of the limited-view reconstruction, obtained as explained above, was used to compute the 2D noise-to-signal ratio function W(u,v) assuming a white noise spectrum N(u,v) with a constant value equal to 2.0% of the variance of the reconstructed image g(x,y). Figures 8.11(a) and 8.11(b) show the filled spectrum and the 2D noise-to-signal ratio function. I performed 2D deconvolution using Eq. 11. The spectrum thus obtained is shown in Figure 8.12, which

Figure 8.11. (a) The "filled-in" spectrum of the limited-view reconstruction. This is used to obtain the 2D noise-to-signal ratio function, and also as a clipping function. (b) The 2D noise-to-signal ratio function.

Figure 8.11 (a).

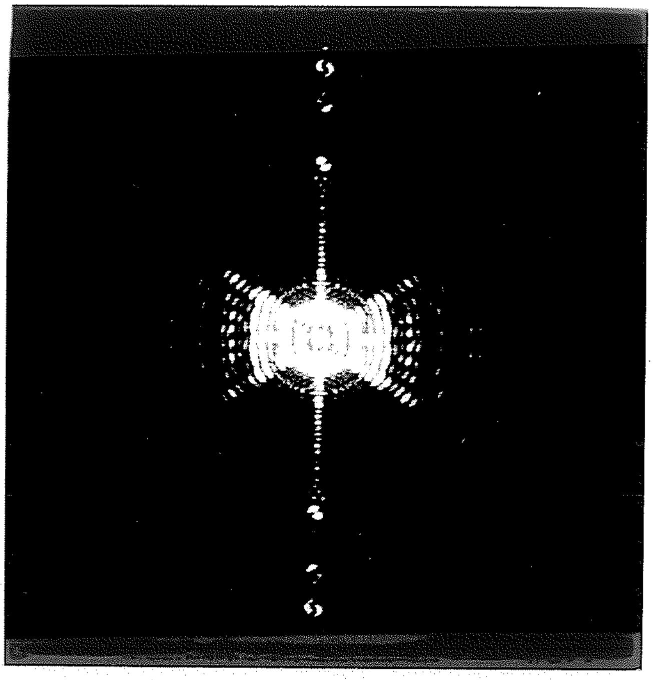


Figure 8.11 (b).

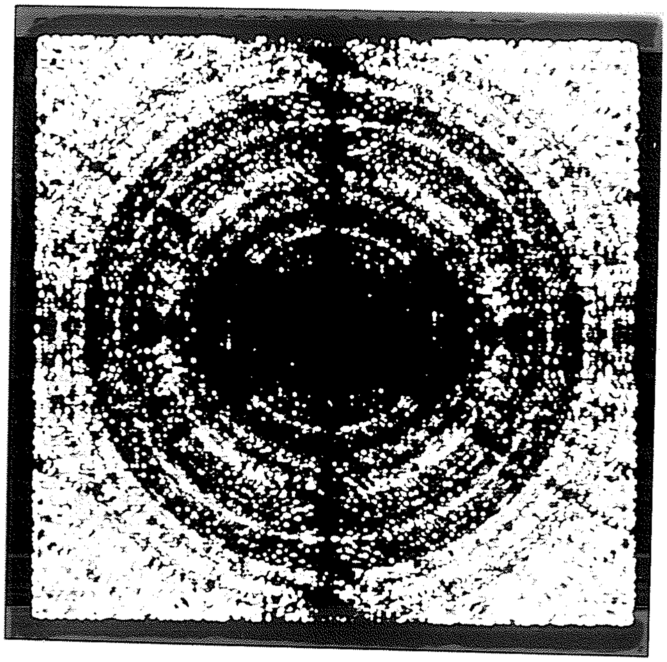
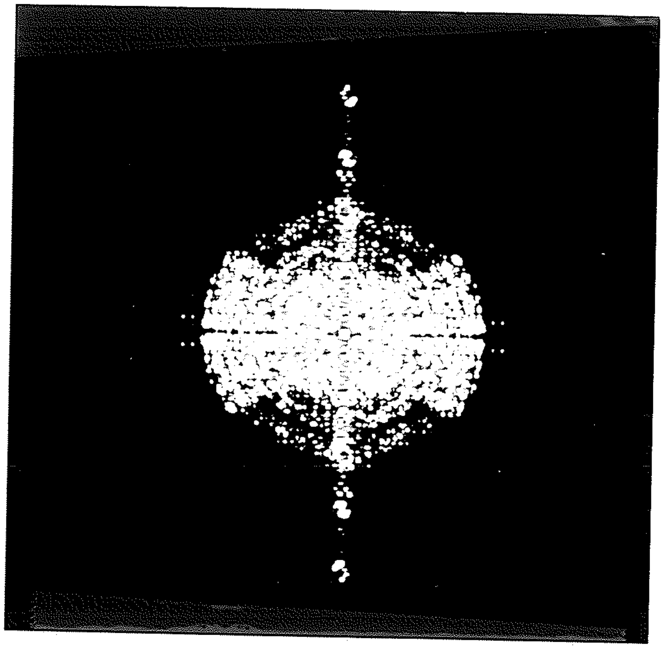


Figure 8.12. The deconvolved spectrum obtained after Wiener deconvolution. The system transfer function H(u,v) was obtained from the 2D point spread function (shown in Figure 8.6(c)).



shows some excessively large values as bright spots. However, the deconvolved spectrum shows spectral components in regions for which no information was available in the Fourier spectrum of the limited-view reconstruction (Figure 8.9). The 2D noise-to-signal ratio function W(u,v) in Eq. 11 can be regarded as a modification function which controls 1/H(u,v) in order to provide optimum restoration. To achieve optimal restoration W(u,v) should be obtained using the exact model of the noise function N(u,v) in Eq. 13. which is not available. Since I have assumed the noise to be white, different values of the constant B = N(u,v) will give different restored images. Unfortunately, there is no theory to serve as a guide in the optimal choice of B at present. (An interactive approach, to obtain the optimal choice of B could be investigated in the future. Such an approach could be based on the difference of the variance between the measured and computed projections.) To obtain an optimal or a sub-optimal value of N(u,v), one has to try a few values and then the clipping procedure may be used to clip the extreme values in the deconvolved spectrum. Thus, the critical dependence of the restoration on N(u,v) may be reduced to a certain extent.

The clipping procedure used is as follows:

$$\begin{aligned} &\text{if } |X(f)/X(0)| > |Y(f)/Y(0)| \\ &\text{then } X(f) = X(f)\,|Y(f)/Y(0)|/|X(f)/X(0)| \end{aligned} \tag{15}$$

where X(f) and Y(f) are the values (in complex numbers) of $\hat{F}(u,v)$ and clipping function respectively, and f is spatial frequency

Figure 8.13. The deconvolved spectrum after clipping.

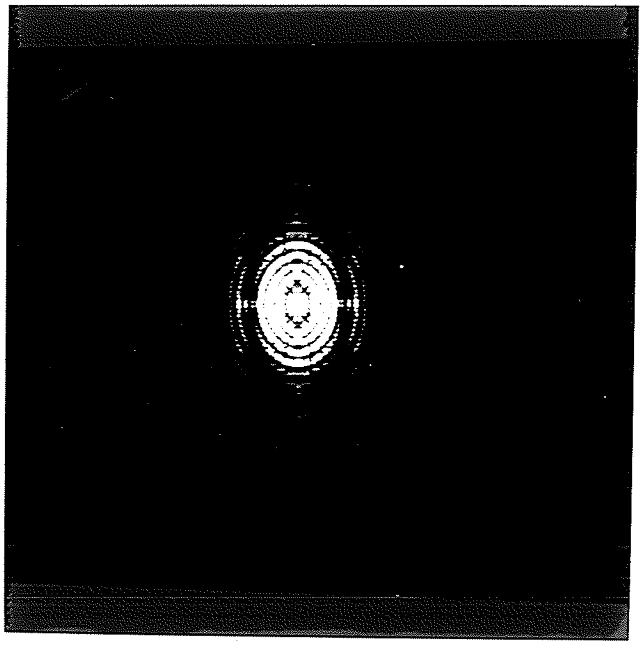


Figure 8.14. The restored image obtained by taking the inverse FFT of the deconvolved spectrum.

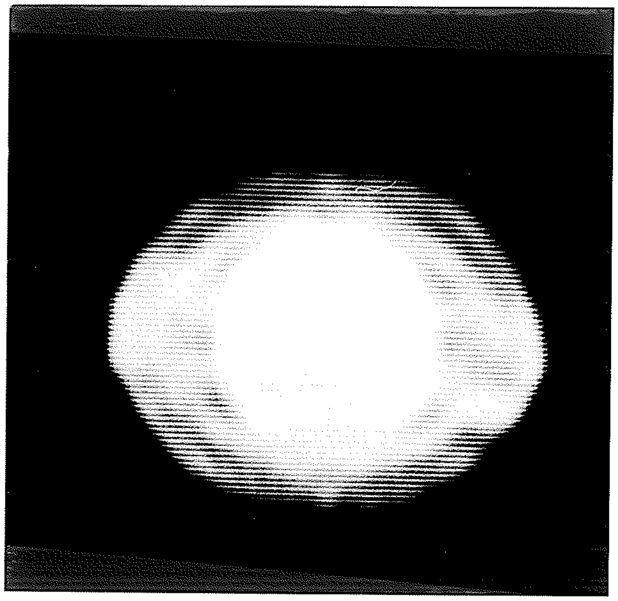
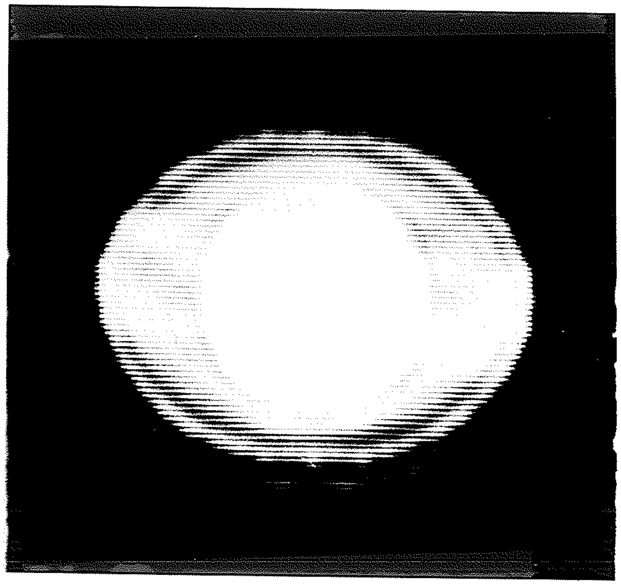


Figure 8.15. The restored image when a disc image was used as a basis function to obtained the system transfer function.



(f = u^2+v^2 where u and v are the rectangular co-ordinates). The deconvolved spectrum after clipping is shown in Figure 8.13. The noise in the higher frequency region is almost removed in the clipped spectrum, which looks smoother than the deconvolved spectrum shown in Figure 8.12. The inverse FFT of the clipped spectrum, which is the restored image, is shown in Figure 8.14. In the restored image it is evident that most of the geometric distortion and artifacts are removed by the 2D deconvolution performed via Wiener filtering. It is interesting to note that this reconstruction is obviously better than multiplicative ART, which maximizes entropy (Lent, 1976).

As a matter of fact, the 2D deconvolution provides a better way of interpolation and modification of spatial frequency information than Fourier algorithms. I also found from the experiments that better results are obtained when the system transfer function H(u,v) is obtained from a basis image instead of a point image. The system transfer function, obtained from a basis function may provide better deconvolved spatial frequency information, if the basis image has some information about the object. Thus, the system transfer function, so obtained, incorporates some a priori information about the geometrical shape of the object. For example, the basis image may be a disc for objects having circular structures. (See Andrews & Hunt, 1977.) The function H(u,v) can be obtained from Equation 14.

The image restored from the limited-view reconstruction (Figure 8.8), using the system transfer function obtained from a disc image of radius 6 pixels, is shown in Figure 8.15. It is evident from a comparison of the two restored images (shown in Figures 8.14 and

Figure 8.16. A test image taken from a Sony video camera.

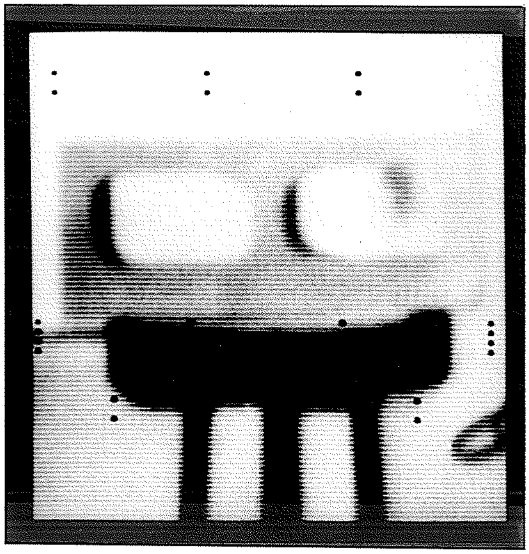


Figure 8.17. The limited-view reconstruction of the test image shown in Figure 8.16 using multiplicative ART and the same eight views used for Figure 8.8.

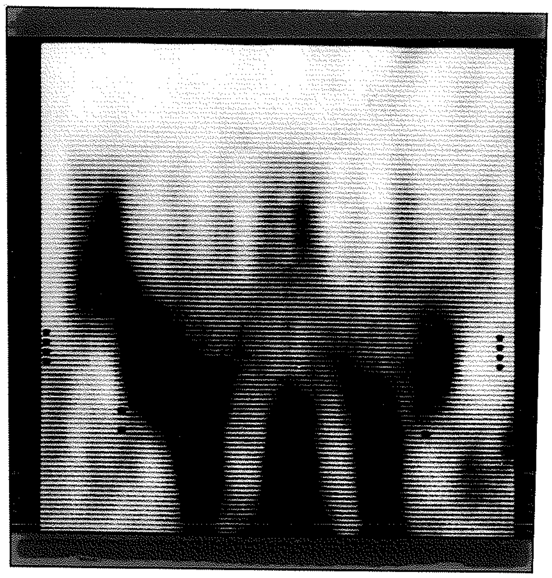


Figure 8.18. A basis image used to obtain the system transfer function.

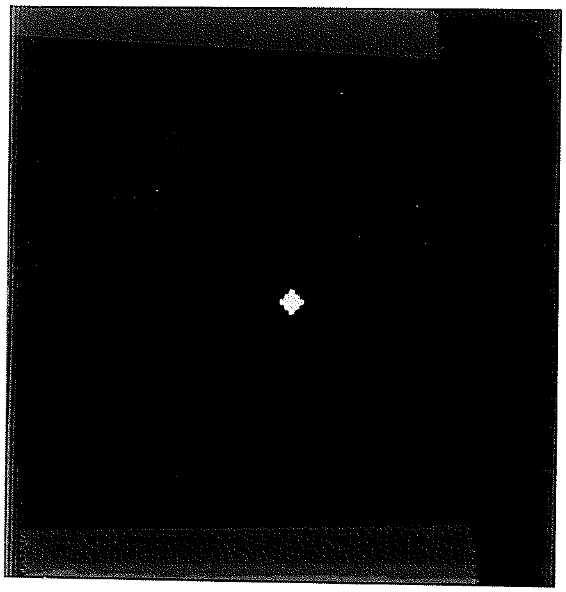
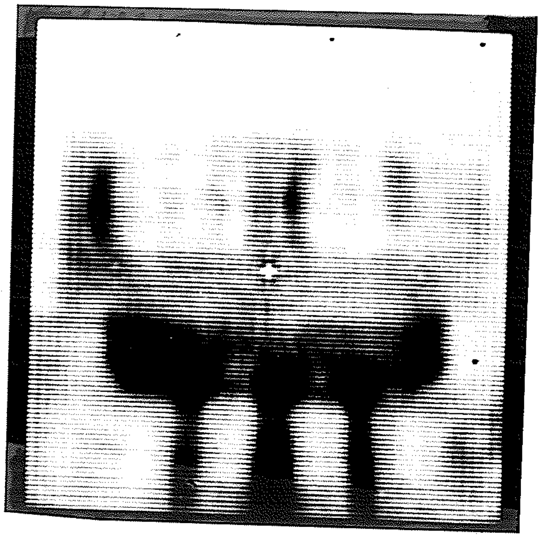


Figure 8.19. The restored image from the reconstruction shown in Figure 8.17.



8.15) that restoration is better when a disc is used to obtain the system transfer function. The circle is restored better in Figure 8.15.

To test the ability of the method in the restoration of general images, I applied the algorithm to the images shown in Figures 8.16 and 8.20. These images were acquired by a Sony video AVC-3260 camera and digitized on a Grinnell GMR-27 image processing system. The size of the stored images was 127X127 pixels. Figure 8.17 shows the limited-view reconstruction of the image shown in Figure 8.16. The reconstruction was obtained by multiplicative ART using the same eight views used in the earlier example (with the viewing angles at 55,65,75,...,125 deg). I used a disc of radius two pixels as the basis image to obtain the system transfer function for this problem. The basis image is shown in Figure 8.18. Note that the disc of radius two pixels is a rectangle. The system transfer function was then obtained from Eq. 14. The restored image obtained by this method is shown in Figure 8.19, which clearly shows details not seen in the limited-view reconstruction, and lacks the streaks typical of standard algorithms.

I also found from the experiments that sometimes application of a low pass filter during the clipping procedure may smooth the image and improve the restoration. This is advantageous when one is unable to find a suitable value of white noise (represented by N(u,v) in Eq. 13) and the image contains uniform grey levels. For example, the image of a face (shown in Figure 8.20) was reconstructed by multiplicative ART using only five views at viewing angles 45, 67.5, 90, 112.5, and 135 deg. The reconstruction is shown in Figure 8.21. The restoration

Figure 8.20. An image of a face having well distributed grey levels.

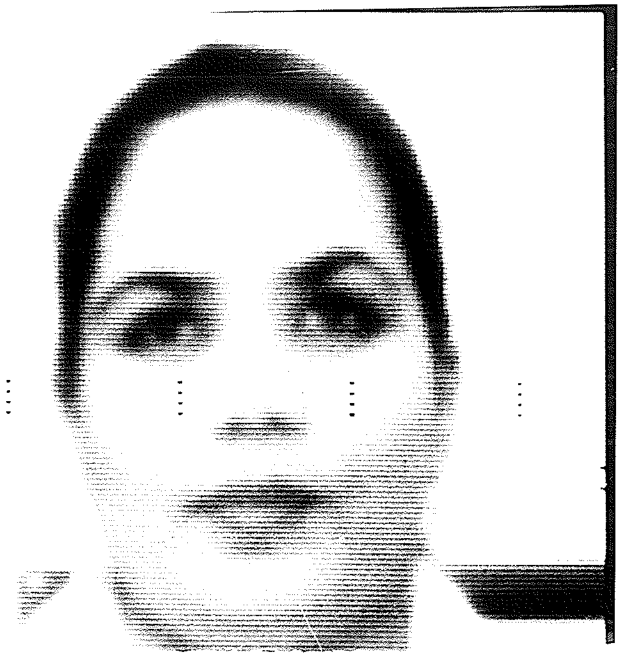


Figure 8.21. The limited-view reconstruction of the face image by multiplicative ART using only five views at 45, 67.5, 90, 112.5 and 135 deg.

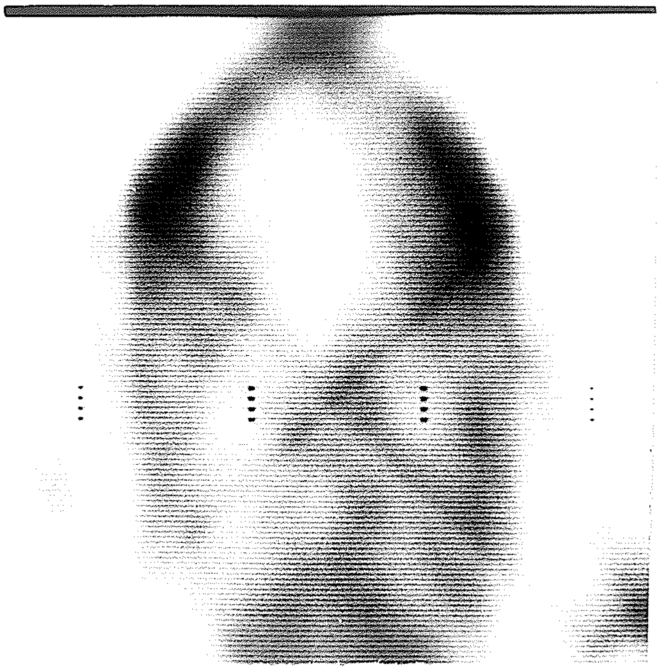
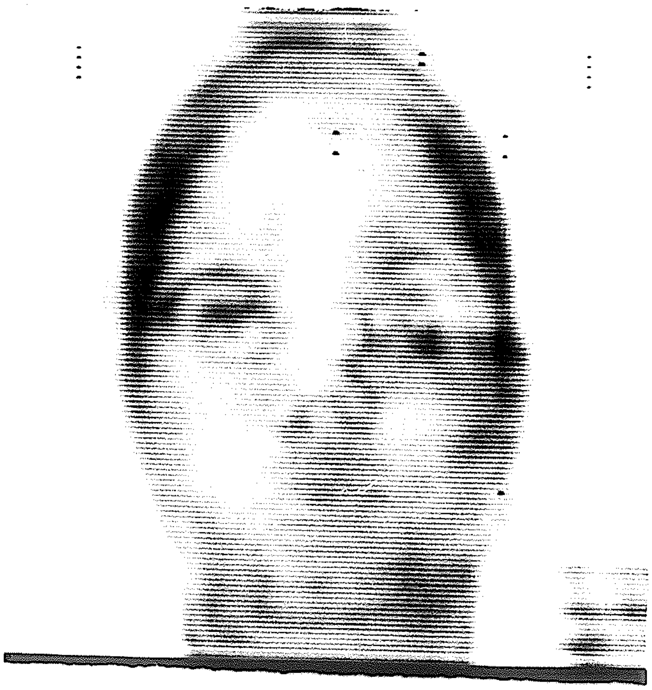


Figure 8.22. The restored image of face reconstruction.



was performed using the same basis image as in the last example (shown in Figure 8.18), but with additional exponential filtering (Gonzalez & Wintz, 1977) after the clipping. Exponential low pass filtering was used with the cut-off frequency locus at a radial distance of 40 from the origin (dc value) in the frequency plane of the 256x256 clipped spectrum F(u,v). The restored image is shown in Figure 8.22. The comparison between Figures 8.21 and 8.22 clearly shows the magnitude of improvement. The limited-view reconstruction hardly shows any information about the object but most of the details of the face can be seen in the restored image.

Restoration For Nevoscopy by 2D Wiener Deconvolution

I have shown in the last section that the 2D Wiener deconvolution algorithm not only removes artifacts in limited-view reconstructions, but, also brings out image features lost by other algorithms. I have only three views available (45, 90, and 180 deg) from the prototype nevoscope to obtain reconstructions of the nevus. I obtained a limited-view reconstruction of the simulated vertical section of the nevus (an ellipse of grey level 170 with four discs of grey level 250 on a square raster of grey level 20). The test· pattern is shown in Figure 8.23. The reconstruction obtained by multiplicative ART for the given angles (45, 90, and 180 deg) is shown in Figure 8.24. The restored image, after performing 2D deconvolution via Wiener filter as explained in previous sections, is shown in Figure 8.25. The system

Figure 8.23. A test pattern representing a vertical section of a nevus.

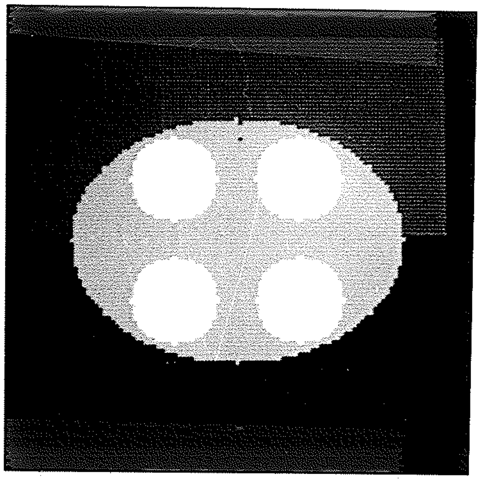


Figure 8.24. The limited-view reconstruction of the test pattern shown in Figure 8.23 by multiplicative ART using only three views at 45, 90, and 180 deg.

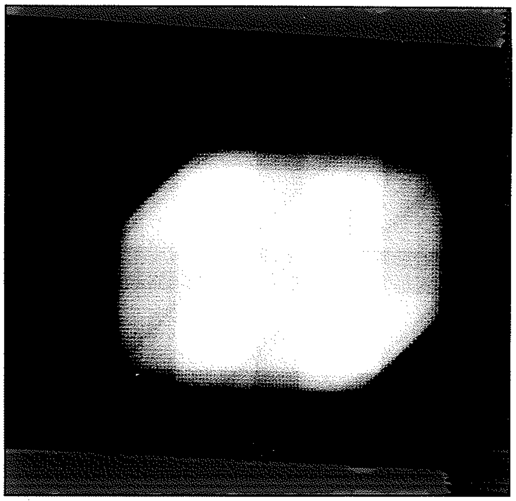
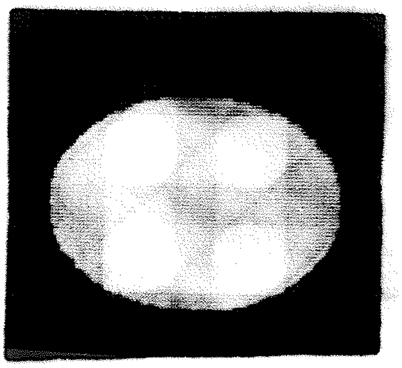


Figure 8.25. The restored image using 2d Wiener deconvolution.



transfer function was obtained from the basis function of a disc of radius of six pixels. It should be noted that the vertical minor axis of the ellipse was measured and found to be the same as in the reconstruction obtained by geometric deconvolution, and in the reconstruction obtained by multiplicative ART followed by 2D Wiener deconvolution. The reconstructed minor axis was found to be 4.16% bigger than the true minor axis (in # of pixels). The major axis was improved in case of Wiener filtering as compared to geometric deconvolution (3.58% bigger against 7.75% smaller in number of pixels).

Images of Transilluminated Nevus

Nevi viewed by transillumination give a direct impression of depth when seen in a stereomicroscope. The nevoscope, attached to a microscope, yields the images shown in Figure 8.26 with incident illumination. These images of a transilluminated nevus, obtained through the nevoscope with surface illumination and external light removed, are shown in Figure 8.27. In incident illumination, due to the bulging of the skin into the space between the two mirrors, often one of the reflections cannot be seen. Nevertheless both reflected views are seen in transillumination, demonstrating that transillumination is providing a means of seeing through the skin. The three images of a transilluminated nevus obtained by the nevoscope (shown in Figures 8.26 and 8.27) are, from left to right, an image at 45 deg, the vertical image at 90 deg, and a glancing image at 180 deg,

Figure 8.26. Incident illumination image of a nevus (on a cadaver) and its reflections in the two mirrors of the nevoscope. The scale is indicated by the separation of the mirrors, which is 3.3 mm.

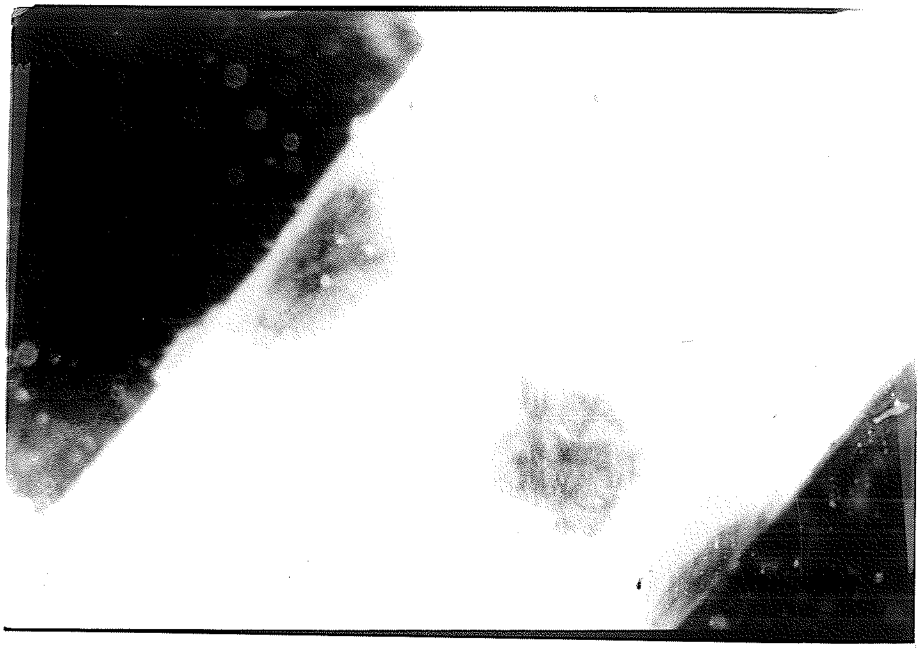
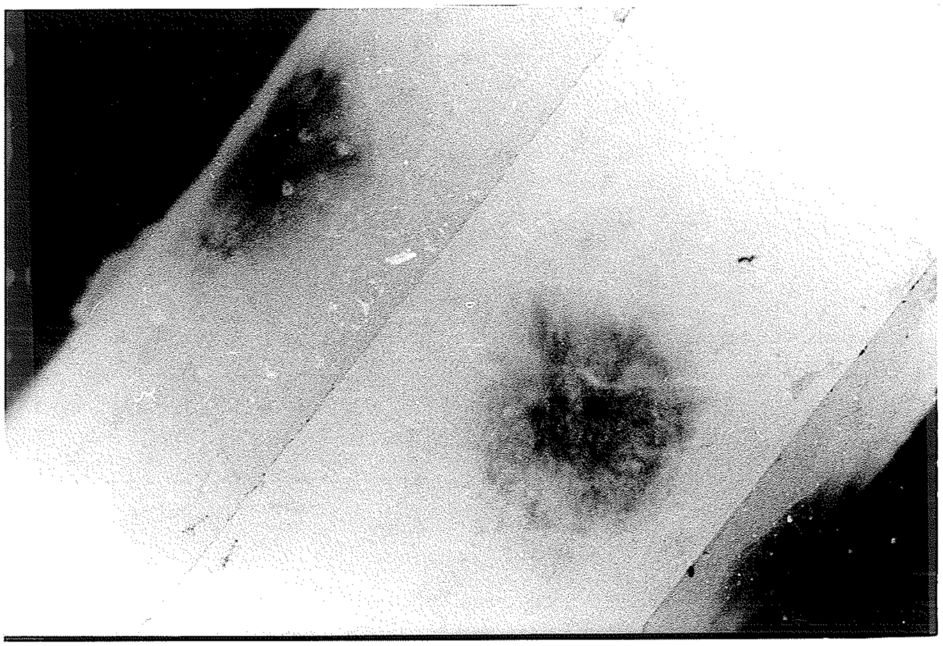


Figure 8.27. Transilluminated image of the same nevus and its reflections. The direct image is in the center.



respectively. (Note that the viewing angles are double that at which the mirrors are mounted.)

3D Computed Reconstruction of Nevi

To see if reconstructions obtained from 3D computed tomography of a transilluminated nevus are correlated with their histology, I performed the method of imaging on a nevus in situ on a cadaver. A stand was built with an arm to hold the stereomicroscope. The length of the arm of the stand was made adjustable through clamps to reach the specific location on the cadaver. A nevus on the left arm of the cadaver (65 years old male) was selected for nevoscopy. The nevoscope mounted on the objective lens of the microscope was positioned on the cadaver nevus. For illumination, the fiber optics cables were inserted in the base plate of the nevoscope. The camera with exposure control unit was hooked up with the stereomicroscope and nevoscopy was performed. The nevus was then excised and histological serial sections were obtained. The details of the procedure of excision and serial sectioning have already been discussed in Chapter 4.

The images of the transilluminated cadaver nevus were then digitized and processed by the algorithm discussed in Chapter 6 to isolate melanotic lesion images from the surrounding skin and the background. The images were then extracted in to 280X110 pixel arrays which were used to compute the projections. These projections were used to obtain the 3D reconstruction using multiplicative ART with geometric deconvolution, and then by multiplicative ART followed by

Wiener filtering deconvolution. The reconstruction were obtained at the scan line numbers 20, 40, 50, 60, 70, and 80 (along the Y-axis) by multiplicative ART with geometrical deconvolution. These reconstructions are shown in Figures 8.28(a), (b), (c), (d), (e), and (f) respectively. From the separation of the mirrors in the digitized image I calculated the pixel width (or pixel-dimension) which is as follows:

Separation of mirrors = 3.3 mm.

Separation of mirrors in digitized image = 120 pixels

Pixel width = 0.0275 mm.

I computed the vertical thickness of the nevus cells in pixels for each section reconstruction and obtained their corresponding thicknesses in mm. I also computed, using the pixel width, that the sections at line number 20, 40, 50, 60, 70, and 80 correspond to the sections 0.055, 0.605, 0.88, 1.155, 1.43, and 1.705 mm away from the left edge of the lesion along the cadaver arm or from the bottom edge of the digitized images (parallel to the mirror axis). Note that the mirror axis serves as a guide for the orientation of the nevus. The sections, with their computed vertical thicknesses of melanin cells, are tabulated in Table VI.

I selected two sections, one at section line 50 and other at section line 70, for the comparison with the corresponding histological sections. I also performed the Wiener filter algorithm on

Figure 8.28. Reconstructions of the cadaver nevus. Six reconstructions are shown at section lines 20, 40, 50, 60, 70, and 80 in (a), (b), (c), (d), (e) and (f) respectively. The most distal section is 8(f). The scale bar represents 200 microns. The corresponding reconstructions to the sections shown in Figures 8.30(a) and 8.30(b) are shown in Figures 8.28(c) and 8.28(e) respectively. Their computed thicknesses are 0.495 mm and 0.565 mm respectively.

Figure 8.28 (a).

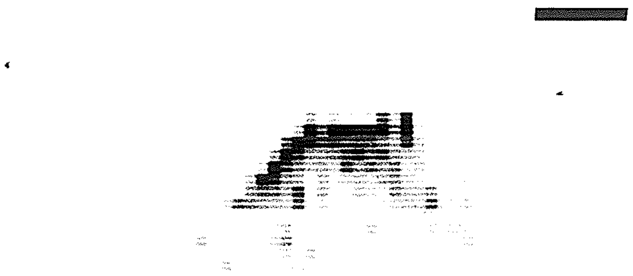


Figure 8.28 (b).

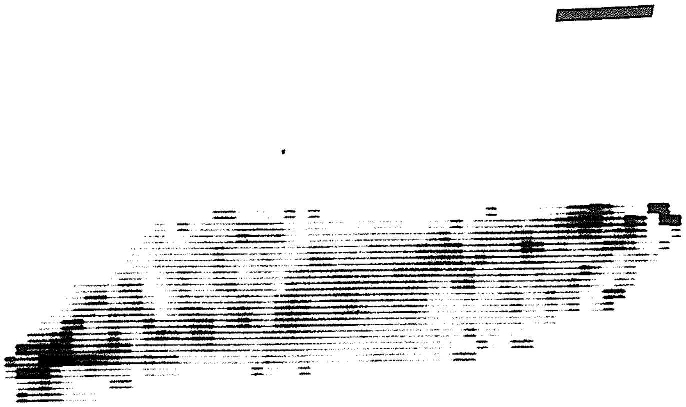


Figure 8.28 (c).

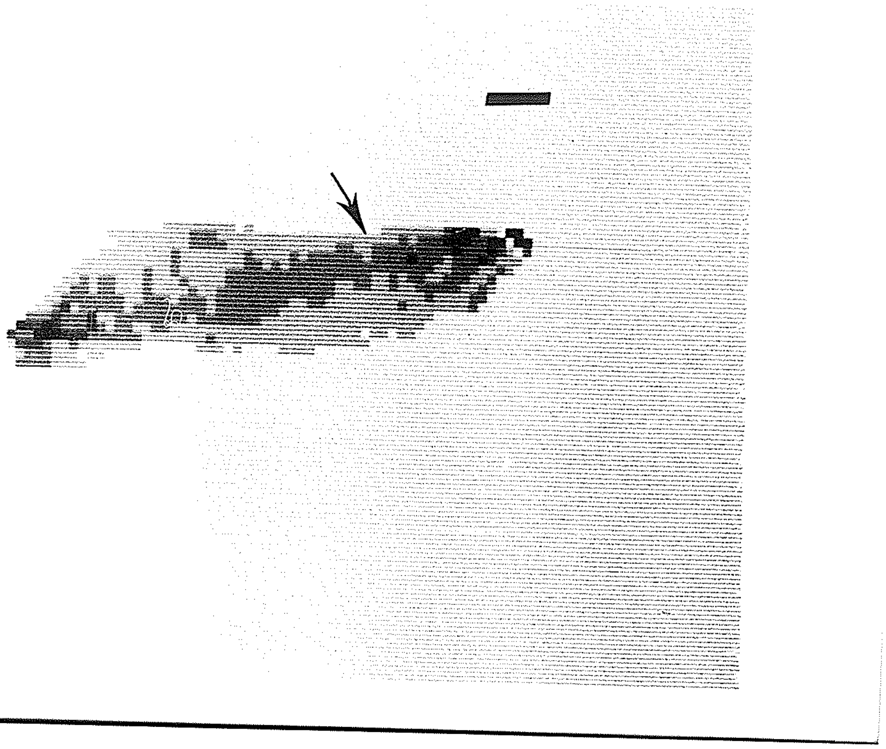


Figure 8.28 (d).

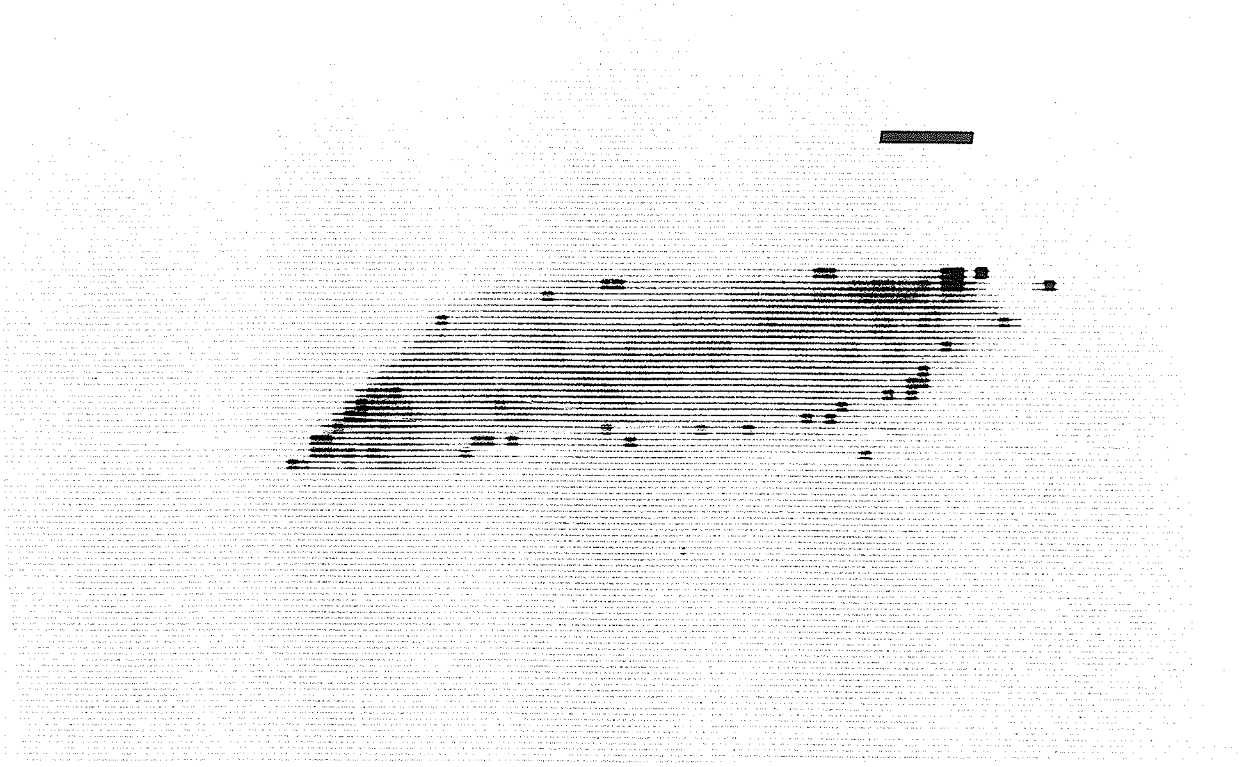


Figure 8.28 (e).

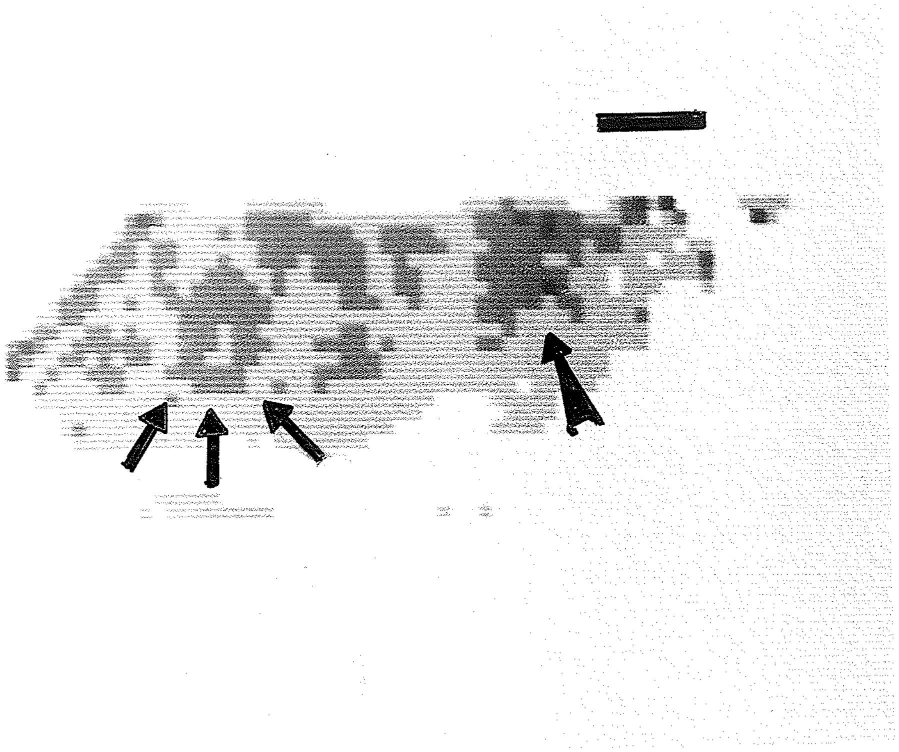


Figure 8.28 (f).

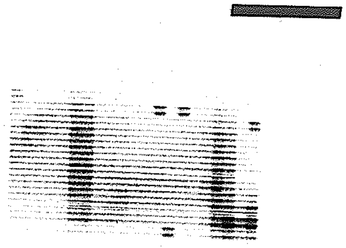


Figure 8.29. Two restored reconstructions (using Wiener deconvolution) at section lines 50, and 70 are shown in (a) and (b) respectively. The scale bar represents 200 microns. Their computed thicknesses are found to be the same as in Figures 8.28(c) and 8.28(e) respectively.

Figure 8.29 (a).

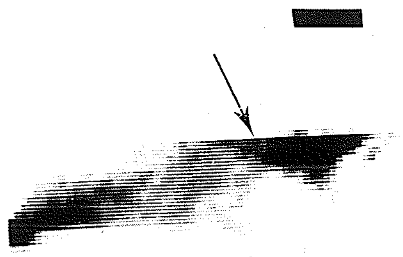
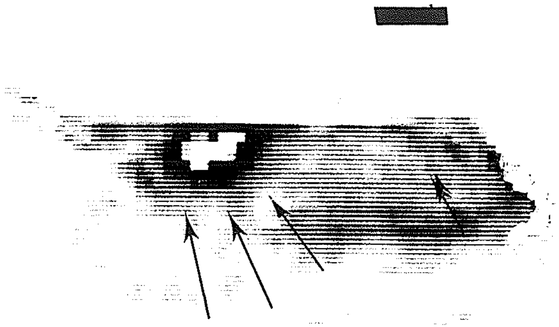


Figure 8.28 (b).



the computed reconstructions (obtained from multiplicative ART only) of these two sections. The restored computed reconstructions are shown in Figures 8.29(a) and 8.29(b), which are reconstructions at section lines 50 and 70 respectively. The comparison between Figure 8.28(c) and Figure 8.29(a), and between Figures 8.28(e) and 8.29(b) shows that the Wiener filter algorithm has improved the reconstruction by removing the geometrical structure caused by the limited geometry of three viewing angles. Details of comparisons are discussed later in this Chapter.

Histological Serial Sectioning of Nevus

The width of the lesion excised from the cadaver was found to be 1.76 mm (from the left edge to the right edge parallel to the mirror edges). Serial sections were cut with a thickness of 5 microns. Thus 352 sections were supposed to be obtained. But only 330 sections were obtained on the slides. The remaining 22 sections were the lost or damaged during the process or because of shrinkage of the tissue. A serial number record was maintained for all sections including lost or damaged sections. I marked a few sections and their related positions in the digitized images so that corresponding sections could be obtained from reconstruction. The actual thickness of the melanin part of the serial sections was measured from the top melanin layer to the deepest cluster of melanin cells with a micrometer. Pictures of the sections were taken through a stereomicroscope. Two histological sections 0.88 mm and 1.43 mm away from the left edge of the nevus are shown in Figures 8.30 (a) and 8.30(b). The computed reconstructions of

Figure 8.30. Two histological sections corresponding to the section lines 50 and 70 are shown in (a) and (b) respectively. These sections are at 0.088 mm away from the left edge of nevus. Measured thickness of nevus is 0.512 mm. The scale bar represents 100 microns. The arrows marked A in Figures 8.28(c), 8.29(a) and 8.30(a) show similar regions in the reconstruction, restored reconstruction and the corresponding histological section.

Figure 8.30(a).

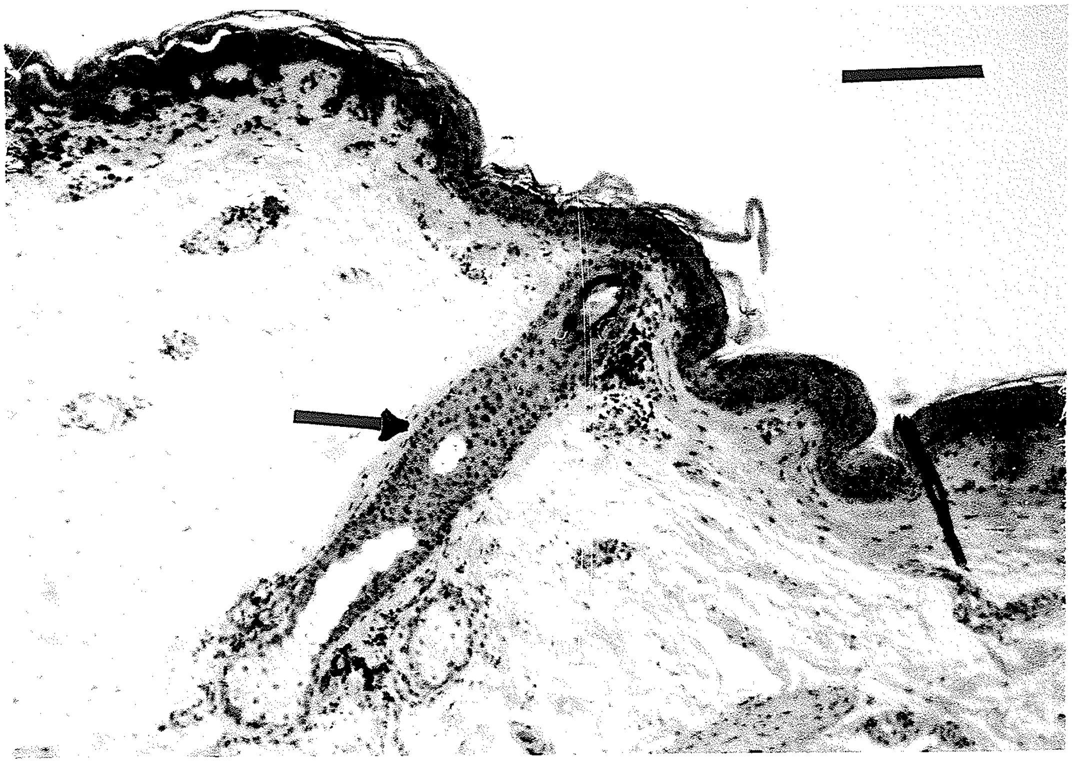
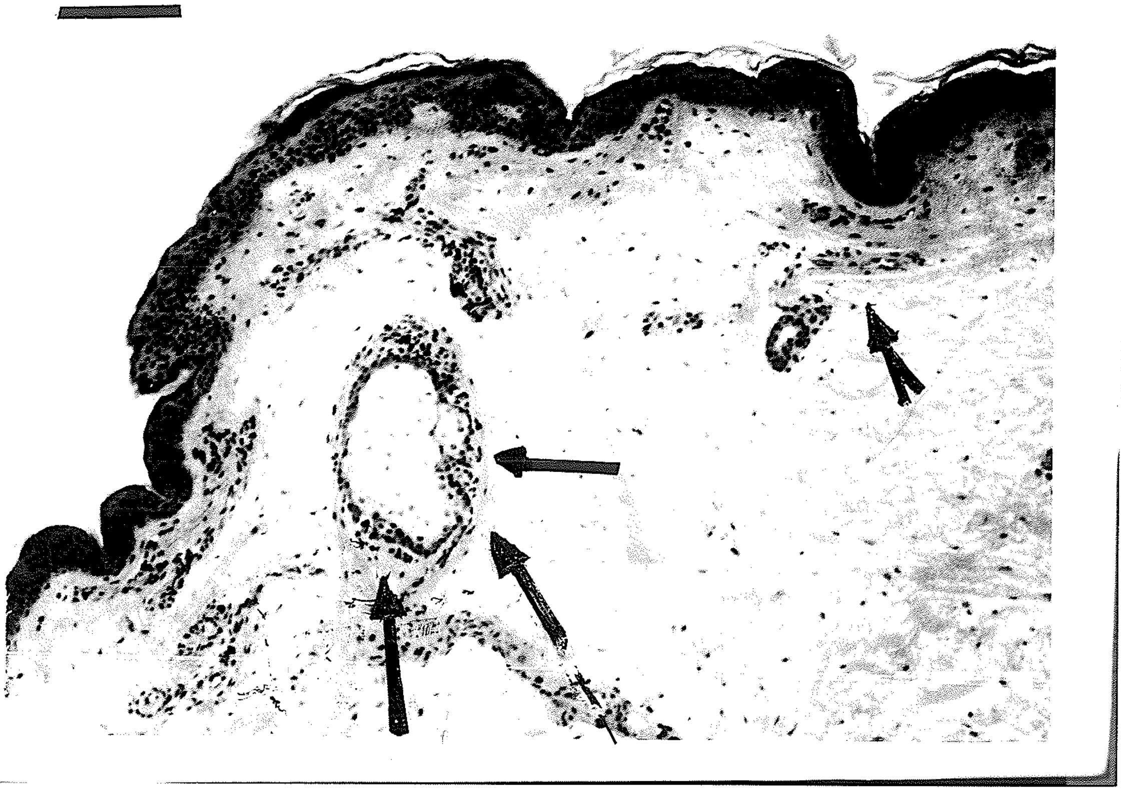


Figure 8.30(b). Histological serial section of the nevus at 1.43 mm far from the left edge of the nevus. Measured thickness is 0.586 mm. The scale bar represents 100 microns. The arrows marked B in Figures 8.28(e), 8.29(b) and 8.30(b) show similar regions in the reconstruction, restored reconstruction and the corresponding histological section.

Figure 8.30 (b).



the same sections, obtained by Wiener filtering, are shown in Figures 8.29 (a) and 8.29(b) respectively, and those obtained by geometrical deconvoution are shown in Figures 8.28(c) and 8.28(e) respectively. I estimate the error in this correspondence to be ± 4 sections.

Comparison of Computed Reconstruction with Histology

A comparison of the actual thicknesses of the nevus in the serial sections with their thicknesses obtained from reconstructions of the corresponding sections is shown in Table VI. The percentage error of thickness of the reconstruction at each section is also shown in this table. This shows that the error is positive at the edges of the lesion while it is negative in-between. This could be because of slightly out of focus mirror images (particularly the 180 deg view) and a scattering effect at the edges. Even then the error in computing the vertical thickness is around 5 % or less.

Comparison of histological sections in Figures 8.30(a) and 8.30(b) with the corresponding computed reconstructions in Figures 8.28(c) and 8.28(e) and also with Figures 8.29(a) and 8.29(b) shows some similarity in the pattern of the melanin. The regions marked by arrows in Figures 8.30(a) and 8.28(c), and in Figures 8.30(b) and 8.28(e) are found to be alike and at the same place. Further these similarities are also found at the same places in the restored computed reconstructions (Figures 8.29(a) and 8.29(b)).

Table VI

COMPARISON OF VERTICAL THICKNESSES OF RECONSTRUCTED AND HISTOLOGICAL SECTIONS OF A CADAVER NEVUS

Distance of section away from left edge of nevus.	Line representing section in digitized images.	Histological thickness	Computed thickness	% error
0.055 mm	20	0.262 mm	0.275 mm	+4.96
0.605 mm	40	0.458 mm	0.44 mm	-4.09
0.88 mm	50	0.512 mm	0.495 mm	-3.32
1.155 mm	60	0.54 mm	0.523 mm	-3.15
1.43 mm	70	0.586 mm	0.565 mm	-3.58
1.705 mm	80	0.34 mm	0.358 mm	+5.29

Width of nevus measured parallel to mirror axis = 1.76 mm

Pixel dimension: 1 pixel = 0.0275 mm

(The restored computed reconstructions are also found to be less noisy than those obtained by geometric deconvolution when compared with their corresponding histological sections). Thus, not only thickness, but some of the details of the morphology are recoverable in limited-view computed tomography of transilluminated nevi.

Two controllable factors could lead to significant improvements in the reconstructed images of nevi. First, the 180 deg view of a nevus is out of focus because of its longer optical path length to the microscope. An objective lens of short focal length or a properly designed correction plate could bring all three images to a common focus. Second, only three views were used. A nevoscope with more mirrors or rotatable mirrors could be designed. Some details about this are given in next chapter.

Chapter 9

FURTHER STUDIES AND RESEARCH

I started this research program aiming at the development of a technique for automated detection of the skin cancer melanoma. I have come up with a new non-invasive method for obtaining different views of skin lesions by transillumination. I have developed reconstruction algorithms suitable for obtaining the 3D reconstructions from very few views. The 3D computed reconstructions and the histology were found to correlate.

Further studies and research are required in order to develop a complete procedure that is reproducible, reliable, and could be used by a dermatologist easily. The following are the problems and areas, beyond this thesis, that I intend to study in detail to develop the required equipment and procedure for automated skin-scanning:

1. Transillumination as an inverse radiative transfer problem for pigmented human skin: The optics of normal and pigmented human skin is to be studied to formulate the problem mathematically. The mathematical model will be used in objective and quantitative analysis for more accurate 3D computed reconstruction of nevi or melanomas in order to provide the best information about changes in diagnostic and prognostic variables.

2. Optics of the nevoscope: The images yielded by the reflections through the mirrors of the nevoscope suffer by being out of focus because of different optical path lengths. This problem can be solved simply by placing a very short focal length lens at a distance more than the focal length of the lens from the base plate of the nevoscope. Thus all the images will be focussed in the same plane.

3. Objective and quantitative analysis of prognostic variables (size, thickness, color, and structure) on 3D computed reconstruction of skin lesions: An appropriate calibration procedure for the objective and quantitative analysis has to developed to perform correct measurements repeatedly.

4. Multi-color analysis: To study the changes in color of the lesion, color picture processing is required. This may be done by using color filters (red, green, and blue) and obtaining different color filtered images separately. These images will be used to obtain different color filtered reconstructions. The comparative analysis of color-filtered reconstructions would provide information about the changes in color of the lesion.

5. Growth of normal nevi: Since melanoma is relatively rare, and it is unethical to study its growth *in situ*, I will use the proposed method on children to study the growth of normal nevi in normal young children. Any planned excisions of suspect nevi will also be reconstructed *in situ* by nevoscopy for comparison with their histological sections.

6. Results with real melanomas: To study the results of this method regarding its repeatability, reliability, and diagnostic

capability, I intend to use this method on melanoma patients.

7. Automated Skin-scanner: I also intend to study the possibility of developing a robot-vison system for automated scanning of the skin of patients. A robot arm holding a solid state (CCD) camera will be made to move along the contours of the body of the patient by a computer-controlled position control system. The robot-vision system will first find the nevi on the body and determine their positional coordinates. The robot-arm would then record the images of the transilluminated nevus through a nevoscope hooked up with a second CCD camera. The images will then be stored and the reconstructions will be obtained. The recent 3D reconstructions would be compared with previous 3D reconstructions of the same nevus to detect the changes.

I also intend to develop a hand-hold nevoscope which will transilluminate the skin-lesion and provide different images of transilluminated lesion. The instrument may have a number of mirrors placed at different angles around the lesion covering all angles. The instrument will also have an optical system with a micrometer to focus the different vertical layers of nevus cells as well as to make surface (2D size) measurements. Thus an appropriate measurement of the depth of the nevus cells would be obtained along with the measurements of the 2D size of the lesion. This set may be used by a dermatologist to look into the moles, to record images of a transilluminated nevus on film (when hooked up with a camera), and to decide which lesions should be checked through 3D nevoscopy or biopsy.

Chapter 10

CONCLUSION

In this thesis, a detailed study on prognostic variables and early detection of cutaneous malignant melanoma has been presented. Based on several investigations, thickness, size and color are found to be the best features for detection of early melanoma. A new non-invasive method for obtaining different images of a transilluminated nevus has also been presented and the required instrumentation has been developed. It has been shown that these images are capable of providing sufficient information for 3D reconstruction. Three-dimensional reconstructions obtained from these images using the reconstruction and restoration techniques provide details about the internal structure of the nevi. Reconstructions of several sections permitted quantitative measurements of the desired variables (such as size, thickness, elevation, etc.) without excision. Thus a baseline 3D shape and structure can be established by this procedure. Any changes found in these specified variables, when the procedure is repeated after some time, could indicate possible development of malignancy.

The possibility of detecting suspect nevi and melanomas by scanning the skin through an automated skin-scanner has been pointed out for future work. This has been proposed for the automated detection of malignant melanoma in its early curable phase. Methods of pattern recognition, computer vision, image processing and artificial intelligence would have to be developed to enable the skin scanner to locate the same lesions at different times, determine their nature (papule, pustule, nevus, melanoma, etc.), and calculate if significant changes have occurred.

Nevoscopy could be generalized for the investigation of other dermatological problems. For instance, the problem of measuring skin thickness without using ionizing radiation (Tan, 1981) might be solvable by transillumination, especially if grazing images were recorded at different wavelengths and subtracted from one another (a multispectral approach). A general purpose instrument for skin investigation might be called a "3D dermascope". It could be carried by a computer controlled robot, which then becomes a "skin scanner".

Nevoscopy could also be carried out using emitted infrared light. This would be a form of microthermography. Microthermography (or infrared microscopy, reviewed by Puchtler, Meloan & Pascha, 1980; Focht, 1981) might show up the earliest stages of reddening or erythema associated with melanoma. Of course, infrared transillumination of nevi could also be considered (cf. Todor, 1978a,b; Morton, 1981).

The investigation of the pathology of the skin, which is the largest and most accessible organ, has tended to be along traditional lines. It is hoped that this approach towards automated investigation and detection of malignant melanoma will open new research areas in automation of diagnostic methods in dermatology.

GLOSSARY

Acral Lentiginous Melanoma a form of malignant melanoma that usually occurs on the hairless skin of the palms and soles. The most common type of malignant melanoma in black people.

ART a class of reconstruction algorithms which reconstruct the unknown object from the measured projections iteratively.

Atypical deviating from the normal (abnormal).

Atypical Moles abnormal moles or pigmented nevi. Usually referred to as dysplastic nevi.

Basal Cell Carcinoma raised, translucent, pearly nodules which may crust, ulcerate, and sometime bleed. They occur most often on the exposed areas of the body. They grow slowly and rarely spreads, but if left untreated it can cause considerable problems. Easily cured, if detected early.

Benign Tumor a mass of abnormal cells that may grow locally but does not spread or metastasize. Also see cancer.

Cancer a disease in which abnormal cells in some organ or tissue go out of control, growing and increasing in number forming a mass which is called a tumor. The tumors, which are not benign, not only enlarge locally but also have the potential to invade and destroy the normal tissue around them and to spread to distant parts of the body. These are called malignant tumors. On the other hand, the benign tumors either do not grow or grow and enlarge only at the site. Distant spread of a cancer occurs when malignant cells detach themselves from the original or primary tumor, are carried to other parts of the body through the blood or lymphatic vessels and establish themselves in the new site as an independent or secondary cancer. A tumor that spreads in this manner is said to have metastasized and the secondary tumor (or tumors) is called a metastasis (or metastases).

Congenital Nevi melanocytic nevi that are present at birth. They may be small (less than 1.5 cm in diameter) or larger than 20 cm in diameter. Large congenital nevi are also called "giant congenital" nevi.

Corium the dermis: the layer of the skin deep to the epidermis, consisting of a deep bed of vascular connective tissue; called also cutis-vera.

Cutaneous pertaining to the skin; of or on the skin.

Cytoplasm protoplasm (unicellular animal organism) of a cell other than that of the nucleus.

Dendritic processes which project between the basal cells of the epidermis.

Dermatology the medical speciality concerned with the diagnosis and treatment of diseases of the skin.

Dermis An alternative for corium. The dermal components of the skin are conventionally devided into two layers: papillary layer which is adjacent to the basement memberane of epidermis, and includes ridges, and papillae of fine connective fibres, loops of capillaries and some nerve endings; and reticular layer which contains coarse, dense, interlacing fibres, a few reticular fibres, collagen fibres and numerous elastic fibres, blood vessels, some sweet glands and hair follicles.

Desquamation shedding of the superficial layer of the skin in scales or shreds.

Desmosome intercellular structure of epidermal cells of all levels. Bundles of intracytoplasmic filaments loop to attach to the disc surface at sites of contact between adjacent cells. Half desmosomes are at points of intimate contact of basal cells to the basement memberane.

Dysplasia Abnormal development of tissue.

Dysplastic Nevus a melanocytic nevus with disordered growth. It usually has faint to moderately intense pink pigment as the background color, irregularity of pigmentation, haziness of outline, and a slightly pebbled surface. It varies in size from a few mm to larger than 5 mm.

Endoplasm zone of granular cytoplasm found the nucleus of many unicellular organisms.

Embryo in man, the developing organism from about two weeks after fertilization to the end of seventh or eighth week.

Embryonic of or pertaining to the embryo.

Epidermis the outermost and nonvascular layer of the skin, derived from embryonic ectoderm, varying in thickness from 1/200 to 1/20 inch, and made up, from within outward, of five layers: (1) a basal layer (stratum basale epidermidis), composed of columnar cells arranged perpendicularly; (2) a spinous layer (stratum spinosum epidermidis), composed of rounded or polygonal pigmented cells; (3) a granular layer (stratum granulosum epidermidis), composed of flattened granular cells; (4) a clear layer (stratum lucidum epidermidis), consisting of several layers of transparent nucleated cells; and (5) a horny layer (stratum corneum epidermidis), composed of flattered horny cells.

Epithelium cells that cover body surfaces and mucous membranes and that line lands.

Erythema Faint to moderately intense pink (distinct from reddish-brown) pigment as the background color or as a peripheral blush about darkly pigmented more advanced lesions.

Familial affecting several members of the same family.

Fix to make firmly attached or set.

Fourier Transformation a method to translate a signal that is observable as intensity over time or space into a signal of intensity versus frequency by resolving the composite sine and cosinewaves that constitute the intensity versus time signal. The Fast Fourier Transform (FFT) is a standard computer algorithm to perform Fourier transformation fast.

Freckle Tan, small or large, irregular macule that appears with sun exposure, often fades away with avoidance. Histologically, there are no more melanocytes in the bassal cell layer, but they are larger and have more melanin.

Grey Level density or light intensity level of the image.

Histogram a graph of the frequency of the various grey levels in an image. The abscissa represents the grey levels, and the ordinate represents the number of pixels having a given grey level. A normalized histogram has the ordinate converted to the percentage of pixels in the image.

Histology microscopical anatomy.

Histopathology the histology of diseased tissues.

Juvenile Melanoma a papillar, pinkish lesion most commonly found on the cheek of a child, but can occur anywhere in the body. The lesion is usually found in children as benign. It metastasizes rarely. Also known as benign juvenile melanoma, spindle cell nevus, Spitz nevus, etc.

Keratinization the process of differentiation by which epidermal cells transform themselves into the firm compact barrier of the outer stratum corneum.

Lentigo Tan, 1 to 5 mm macule. Usually it does not change with season or sun.

Lesion any pathological or traumatic discontinuity of tissue or loss of function of a part.

Lymph a transparent, slightly yellow liquid of alkaline reaction, found in the lymphatic vessels and derived from the tissue fluids. Under the microscope, the lymph is seen to consist of a liquid portion and of cells, most of which are lymphocytes.

Macule a flat circumscribed discoloration of the skin or mucous memberane up to 1 cm in its longest diameter.

Malignant tending to become progressively worse and to result in death. Having the properties of anaplasia, invasion, and metastasis; said of tumors.

Malignant Tumor a mass of abnormal cells that spreads and metastasizes. See cancer.

Melanin the dark amorphous pigment of the skin, hair, and various tumors.

Melanocyte the cell responsible for the synthesis of melanin; melanocytes constitute 5 to25 per cent of epidermal cells.

Melanoma a tumor made up of melanin-pigmented cells. When used alone the term refers to malignant melanoma.

Metastasis a growth of pathological microorganisms or of abnormal cells distant from the site primarily involved by the morbid process. In other words "seed" of cancer that may spread from the original site to other areas. Also see cancer.

Mitosis a method of indirect division of a cell, consisting of a daughter nuclei normally receive identical complements of the number of chromosomes characteristic of the somatic cells of the species.

Mole a pigmented nevus, smooth-surfaced papule that is usually skin colored to brown and is commonly found on the face, but may appear any where in the body at birth, during adolescence, or during pregnancy. Morphologic pertaining to shape or form.

Nevus a mole. The classification of nevi includes intradermal nevi, junctional nevi, and active junctional nevi depending on the location of melanocytes found in the dermis. Special forms of nevus are Ito's nevus, Becker's nevus, halo nevus, blue nevus, and Mongolian spot.

Nevus Cells specialized epithelial cells containing varying amount of melanin.

Organelle specialized part of a protozoon (unicellular animal organism) that performs a special function; one of many intracellular cytoplasmic structures.

Papillaris the mesodermal components of the skin are conveniently divided into pars papillaris consisting of papillae and subpapillary layer, pars reticularis, and subcutaneous fat tissue or hyperderm. The pars papillasis contains relatively more cells and vessels then does the pars reticularis. The frame of the pars papillaris is made of collagen fibres and thin bundles.

Papillary Layer see papillaris; also see dermis.

Papule a solid elevated lesion of the skin or mucous memberane up to 1 cm in its longest diameter.

Pigmentation coloration, especially abnormally increased coloration, by melanin.

Pigmented Nevus Flat, tan to black, irregularly shaped, multiple macules, usually 1 to 10 mm in size. Also see mole.

Pixel smallest picture element, usually a square in digitized images.

Precursor a lesion that has a significantly higher rate of evolving into melanoma than would be expected by randon chance alone.

Projection shadow of the object from a specific angle. In computed tomography, projections are records or photographs taken from various directions using transmitted radiation.

Pruritis itching, a sensation that elicits the desire to scratch.

Pustule a vesicle or bulla containing pus rather than clear fluid.

Reticularis consists of a three-dimensional meshwork of collagen bundles which are accompanied by thick interconnected elastic fibres and ribbons in similar three-dimensional arrangement. Reticulam fibres are present in dense array just below the epidermis and are arranged vertical to the interface.

Reticular Layer see reticularis; also see dermis.

Rete ridges network of projected fibres.

Satellitosis accumulation of neurological cells about neurons; seen whenever neurons are damaged.

Scar first red then pale avascular, smooth hyaline wound repair; may be flat, depressed, elevated, or hypertrophic.

Skin Cancer cancer of the skin, classified according to the cells involved: basal cell carcinoma, squamous cell carcinoma, and malignant melanoma. Approximately 95% of all skin cancers fall in the first two classifications, but at least 75% - 80% of the reported deaths from all skin cancers occur because of malignant melanoma alone. Almost all lesions can be cured, if detected early enough.

Squamous Cell Carcinoma raised, pink, opaque nodules or patches which frequently ulcerate in the center. They most often appeared on exposed areas of the body, and occasionally can spread rapidly. Curable easily, if detected early.

SPARTAF a reconstruction algorithm which reduces the streaking artifacts, spelled as Streak Preventive Algebraic Reconstruction Technique via Adaptive Filtering.

Stratum basale see epidermis.

Stratum corneum see epidermis.

Stratum lucidum see epidermis.

Stratum granulasum see epidermis.

Stratum spinosum see epidermis.

Syndrome set of symptoms and signs occuring together in a complex.

Transillumination shining light on an object allowing some light to pass through.

Tumor a solid, elevated lesion of the skin or mucous memberane, with the added dimension of depth in tissue.

Ulcer

loss of tissue from a surface, leaving an uncovered wound.

Verrucous surface rough surface.

Vesicle a fluid-filled, superficial, elevated lesion of the skin or mucous memberane, usually less than 1 cm in diameter.

Wiener Deconvolution the two-dimensional deconvolution with the point spread function of the algorithm, performed via Wiener filter.

REFERENCES

[1] Anderson, R.R., J. Hu & J.A. Parrish, Optical radiation transfer in the human skin and application in *in vivo* remittance spectroscopy, Proceeding of the Symposium on Bioenginering and the Skin, Cardiff. Wales, July 19-21, 1979.

[2] Anderson, R.R., M.J. LeVine & J.A. Parrish, Selective modification of the optical properties of psoriatic vs. normal skin, Book of Abstracts, 8th International Photobiology Congress, Strasbourg France, p.152, 1980.

[3] Anderson, R.R. & J.A. Parrish, The optics of human skin, J. Invest. Dermatol. 77, 13-19, 1981.

[4] Andrews, H.C. & B.R. Hunt, Digital Image Restoration, Englewood Cliffs, New Jersey: Prentice-Hall, 1977.

[5] Ariel, I.M., Malignant Melinoma, New York: Appleton-Century-Crofts, 1981.

[6] Attie, J.N. & Khalif, R.A., Melanotic Tumors, Springfield, Charles B. Thomson, Ill., 1964.

[7] Bachem, A & C.I. Reed, The transparency of live and dead animal tissue to ultraviolet light, Am. J. Physiol. 90 600-606, 1929.

[8] Balch, C.M., T.M. Murad, S.J. Soong, et al., Tumor thickness as a

guide to surgical management of clinical stage 1 melanoma patients, Cancer 43 883-888, 1979.

[9] Balch, C.M., J.A. Wilkerson, T.M. Murad, et al., The prognostic significance of ulceration of cutaneous melanoma, Cancer 45 3012-3017, 1980.

[10] Becker, S.W., Pitfalls in the diagnosis and treatment of melanoma, Arch. Dermatol. Syphilol. 69 11, 1954.

[11] Becker, S.W., Black lesions of the skin, Calif. Med. 88 228, 1958.

[12] Bondi, E.E., The common acquired nevus: A melanoma precursor?, presented at the 43rd Annual Meeting of American Academy of Dermatology, Washington D.C., Dec. 1-6, 1984.

[13] Bracewell, R.N., Strip integration in radio astronomy, Aust. J. Phys., 9(2) 198-217, 1956.

[14] Bracewell, R.N. & A.C. Riddle, Inversion of fan-beam scan in radio astronomy, Astrophys. J. 150(2), 427-434, 1967.

[15] Breslow, A., Thickness, cross-sectional area, and depth of invasion in the prognosis of cutaneous melanoma, Ann. Surg. 172 902, 1970.

[16] Breslow, A., Tumor thickness, level of invasion and mole dissection in stage 1 cutaneous melanoma, Ann. Surg. 182 572, 1975.

[17] Breslow, A., Tumor thickness, level of invasion and node

dissection in stage 1 cutaneous melanoma, Ann. Surg. 182 572-575, 1975.

[18] Breslow, A., N. Cascinelli, E.P. van der Esch, and A. Morabito, Stage 1 melanoma of the limbs: Assessment of prognosis by level of invasion and maximum thickness, Tumori 64 273-284, 1978.

[19] Burch, S.F., S.F. Gull & J. Skilling, Image restoration by a powerful maximum entropy method, Computer Vision, Graphics, Image Proc. 23, 113-128, 1983.

[20] Burg, J.P., Maximum entropy spectrum analysis, Ph.D. Thesis, Stanford University, Palo Alto, Calif., 1975.

[21] Byrne, C.L. & R.M. Fitzgerald, Reconstruction from parallel information with application to tomography, SIAM, Applied Math. (42) 4 933-940, 1982.

[22] Byrne, C.L., R.M. Fitzgerald, M. A. Fiddy, T.J. Hall & A.M. Darling, Image restoration and resolution enhancement, J. Opt. Soc. Am. 73, pp. 1481-1487, 1983.

[23] Cassileth, B.R., W.H. Clark, Jr., R.M. Heiberger, V. March & A. Tenaglia, Relationship between patients' early recognition of melanoma and depth of invasion, Cancer 49(1), 198-200, 1982.

[24] Chandrasekhar, S., Radiative Transfer, New York: Dover Publ., 1960.

[25] Chandrasekhar, S., Plasma Physics, Chicago: University of Chicago Press, 1960.

[26] Chow, C.K. & T. Kaneko, Automated boundary detection of the left ventricle from cineangiograms, Comp. and Biomed. Res. 5, 388-410, 1972.

[27] Clark, W.H.Jr., A classification of malignant melanoma in man, correlated with histogenesis and biological behavior, In Montagna W: Advances in Biology of Skin. The Pigmented System, New York, Pergamon Press, pp. 621-647, 1967.

[28] Clark, W.H.Jr., L. Form & E.A. Bernardino, The histogenesis and biological behavior of primary human malignant melanomas of the skin, Cancer Res. 29 705, 1969.

[29] Clark, W.H.Jr. & M.C. Mihm, Lentigo maligna and lentigo maligna melanoma, Am. J. Pathol. 55 39-67, 1969.

[30] Clark, W.H.Jr., A.M. Ainsworth, E.A. Bernardino, C. Yang, M.C. Mihm & R.J. Reed, The developmental biology of primary human malignant melanoma, Semin. Oncol. 2 83-103, 1975.

[31] Clark, W.H.Jr., R.R. Reimer, M. Greene, A.M. Ainsworth & M.J. Mastrangelo, Origin of familial malignant melanomas from heritable melanocytic lesion, Arch. Dermatol. 114 732-738, 1978.

[32] Clark, W.H.Jr., E.A. Bernardino, R.J. Reed & A.W. Kopf, Acral lentiginous melanomas, In Clark, W.H. (eds) Human Malignant Melanoma, New York, Grune & Stratton pp. 109-124, 1979.

[33] Clark, W.H.Jr., L.I. Goldman & M.J. Mastrangelo, Human Malignant Melanoma, Grune & Stratton, New York, 1979.

[34] Coleman, W.P., P.R. Loria, R.J. Reed, et al., Acral lentiginous melanoma, Arch. Dermatol. 116 773-776, 1980.

[35] Crowther, R.A., D.J. Derosier & A. Klug, The reconstruction of a three-dimensional structure from projections and its application to elactron microscopy, Proc. Ray. Soc., Ser. A 317, 319-340, 1970.

[36] Crutcher, W.A., Dysplastic nevi: clinical aspects, presented at the 43rd Annual Meeting of American Academy of Der matology, Washin¬gton D.C., Dec. 1-6, 1984.

[37] Day, C.L., Jr., T.J. Harrist, F. Gorstein, A.J. Sober, R.A. Lew, R.J. Friedman, B.S. Pasternack, A.W. Kopf, T.B. Fitzpatrick & M.C. Mihm, Jr., Malignant melanoma, prognostic significance of "microscopic satellites" in the reticular dermis and subcutaneous fat, Ann. Surg. 194(1), 108-112, 1981.

[38] Day, C.L., Jr., A.J. Sober, A.W. Kopf, R.A. Lew, M.C. Mihm, Jr., P. Hennessey, F.M. Golomb, M.N. Harris, S.L. Gumport, J.W. Raker, R.A. Malt, A.B. Cosimi, W.C. Wood, D.F. Roses, F. Gorstein, A. Postel, W.B.N. Grier, M.N. Mintzis & T.B. Fitzpatrick, A prognostic model for clinical stage I melanoma of the upper extremity, the importance of anatomic subsites in predicting recurrent disease, Ann. Surg. 193(4), 436-440, 1981.

[39] Day, C.L., Jr., A.J. Sober, A.W. Kopf, R.A. Lew, M.C. Mihm, Jr., F.M. Golomb, P. Hennessey, M.N. Harris, S.L. Gumport, J.W. Raker, R.A. Malt, A.B. Cosimi, W.C. Wood, D.F. Roses, F. Gorstein, T.B. Fitzpatrick & A. Postel, A prognostic

model for clinical stage I melanoma of the lower extremity, location on foot as independent risk factor for recurrent disease, Surgery 89(5), 599-603, 1981.

[40] Day, C.L., A.J. Sober, R.A. Lew, et al., Malignant melanoma patients with positive nodes and relatively good prognosis, Cancer 47 955-962, 1981.

[41] Day, C.L., Jr., M.C. Mihm, Jr., A.T. Sober, T.B. Fitzpatrick & R.A. Malt, Narrower margins for clinical stage 1 malignant melanoma, New England J. Med. 306(8), 479-482, 1982.

[42] Day, C.L., Jr., M.C. Mihm, Jr., R.A. Lew, A.W. Kopf, A.J. Sober & T.B. Fitzpatrick, Cutaneous malignant melanoma: prognostic guidelines for physicians and patients, CA-A Cancer J. Clinicians 32(2), 113-122, 1982.

[43] Day, C.L., Jr., M.C. Mihm, Jr., R.A. Lew, M.N. Harris, A.W. Kopf, T.B. Fitzpatrick, T.J. Harrist, F.M. Golomb, A. Postel, P. Hennessey, S.L. Gumport, J.W. Raker, R.A. Malt, A.B. Cosimi, W.C. Wood, D.F. Roses, F. Gorstein, D. Rigel, R.J. Friedman, M.M. Mintzis & A.J. Sober, Prognostic factors for patients with clinical stage I melanoma of intermediate thickness (1.51-3.99 mm), a conceptual model for tumor growth and metastasis, Ann. Surg. 195(1), 35-43, 1982.

[44] Day, C.L., Jr., R.A. Lew, M.C. Mihm, Jr., A.J. Sober, M.N. Harris, A.W. Kopf, T.B. Fitzpatrick, T.J. Harrist, F.M. Golomb, A. Postel, P. Hennessey, S.L. Gumport,

J.W. Raker, R.A. Malt, A.B. Cosimi, W.C. Wood, D.F. Roses, F. Gorstein, D. Rigel, R.J. Friedman, M.M. Mintzis & R.W. Grier, A multivariate analysis of prognostic factors for melanoma patients with lesions ≥ 3.65 mm in thickness, the importance of revealing alternative Cox models, Ann. Surg. 195(1), 44-49, 1982.

[45] Day, C.L., Jr., M.C. Mihm, Jr., A.J. Sober, M.N. Harris, A.W. Kopf, T.B. Fitzpatrick, R.A. Lew, T.J. Harrist, F.M. Golomb, A. Postel, P. Hennessey, S.L. Gumport, J.W. Raker, R.A. Malt, A.B. Cosimi, W.C. Wood, D.F. Roses, F. Gorstein, D. Rigel, R.J. Friedman & M.M. Mintzis, Prognostic factors for melanoma patients with lesions 0.76-1.69 mm in thickness, an appraisal of "thin" level IV lesions, Ann. Surg. 195(1), 30-34, 1982.

[46] Day, C.L., M.C. Mihm, Jr., A.J. Sober, T.B. Fitzpatrick, R.A. Malt, A.W. Kopf, R.A. Lew & T.J. Harrist, Skin lesions suspected to be melanoma should be photographed: gross morphological features of primary melanoma associated with metastases, JAMA 248(9), 1077-1081, 1982.

[47] DeRosier, D.J. & A. Klug, Reconstruction of three dimensional structure from electron micrographs, Nature (London) 217, 130-134, 1968.

[48] DeRosier, D.J., The reconstruction of three dimensional images from electron micrographs, Contemp. Phusics 12(5), 437-452, 1971.

[49] Dhawan, A.P., D. Martin & R. Gordon, Computerized imaging of

pigmented skin lesion: an adaptive histogram algorithm to isolate melanotic lesions from surrounding skin, J. Invest. Dermatol., to be sumitted, 1985.

[50] Dhawan, A.P., R. Gordon & R.M. Rangayyan, Computed tomography by transillumination to detect early melanoma, IEEE Trans. Biomed. Eng. BME-31, 574, 1984.

[51] Dhawan, A.P., R.M. Rangayyan & R. Gordon, Wiener filtering for deconvolution of geometric artifacts in limited-view image reconstruction, Proc. 1984 International Symposium on Medical Images and Icons, July 23-27, Arlington, IEEE Computer Society, pp. 168-172, 1984.

[52] Dhawan, A.P., R.M. Rangayyan & R. Gordon, Image restoration by deconvolution in computed tomography, Scientific Exhibits at Annual Meeting of Radiological Society of North America, Nov. 25-30, Washington D.C., 1984.

[53] Dhawan, A.P., R.M. Rangayyan & R. Gordon, Image restoration by two-dimensional deconvolution in limited-view reconstruction, In: Technical Digest, Topical Meeting on Industrial Applications of Computed Tomography and NMR Imaging, ed. R. Gordon et al., Washington, D.C.: Optical Society of America, pp. TuA5-1 to TuA5-3, 1984.

[54] Dhawan, A.P., R. Gordon & R.M. Rangayyan, Computed tomography by transillumination to detect early melanoma, Proc. Frontier of Engineering and Computing in Health Care, 6th Ann. Conf. of IEEE Eng.

in Medicine and Biology Society, 518-522, 1984.

[55] Dhawan, A.P., R. Gordon & R.M. Rangayyan, Nevoscopy: three-dimensional computed tomography of nevi and melanomas in situ by transillumination, IEEE Trans. Med. Imaging MI-3(2), 54-61, 1984.

[56] Dhawan, A.P., R.M. Rangayyan & R. Gordon, Image restoration by Wiener deconvolution in limited-view computed tomography, Applied Optics (May 15), in press, 1985.

[57] Dubreuilh, M.W., On circumscribed precancerous melanoma, De La Melanose Circonscrite Precancereuse, Ann. Derm. Syph. 3, 129-205, 1912.

[58] Duerinckx, A.J. & A Mecovski, Information and artifact in computed tomography image statistics, Medical Physics 7, pp. 127-134, 1980.

[59] Elwood, M. & A.H. Lee, Data on the epidemiology of malignant melanoma, In Clark, W. Jr. (eds) Human Malignant Melanoma, New York, Grune & Stratton pp. 261-272, 1979.

[60] Fleming, I.D., Barnawell Jr., P.E. Burlison, et al., Skin cancer in black patients, Cancer 35 600-605, 1975.

[61] Focht, M.W., Infrared microscopy for examination of inks and documents, International Adv. Nondestructive Testing 7, 311-325, 1981.

[62] Frieden, B.R., Restoring with maximum likelihood and maximum entropy, J. Opt. Soc. Am. 62, pp. 511-518, 1972.

[63] Frieden, B.R., Image restoration using a norm of maximum information, Optical Eng. 19(3), 290-296, 1980.

[64] Frieden, B.R., Statistical models for the image restoration problems, Computer Graphics Image Processing 12, pp. 40-59, 1980.

[65] Frieden, B.R., Unified theory for estimating frequency-of-occurrence laws and optical objects, J. Opt. Soc. Am. 73(7), 927-938, 1983, entropy.

[66] Gaarder, N.T. & G.T. Herman, Algorithms for reproducing objects from their x-rays, Comput. Graphics Image Process. 1 97-106, 1972.

[67] Gerchberg, R.W., Super-resolution through error energy reduction, Opt. Acta 21, pp. 709-720, 1974.

[68] Gilbert, P., Iterative methods for 3-dimensional reconstruction of an object from projections, J. Theor. Biol. 36(1), 105-, 1972.

[69] Gilbert, P.F.C., Reconstruction of a 3-dimensional structure from projections and its application to electron microscopy. 2. Direct methods, Proc. Roy. Soc. B182, 89-, 1972.

[70] Gonzalez, R.C. & P. Wintz, Digital Image Processing, Reading: Addison-Wesley Publ. Co., 1977. $28,

[71] Gordon, R., R. Bender & G. T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography., J. Theor. Biol. 29, 471-481., 1970.

[72] Gordon, R., A proposal for deconvolution of reconstructions.,

Workshop on Information Treatment in Electron Microscopy, Basel, 2pp., 1972.

[73] Gordon, R., Artifacts in reconstructions made from a few projections., Proceedings of the First International Joint Conference on Pattern Recognition, Oct. 30 to Nov. 1, Washington, D. C., IEEE Computer Society, Northridge, California, pp. 275-285., 1973.

[74] Gordon, R., & G. T. Herman, Three dimensional reconstruction from projections: A review of algorithms., Int. Rev. Cytol. 38, 111-151., 1974.

[75] Gordon, R., A tutorial on ART (Algebraic Reconstruction Techniques)., IEEE Trans. Nucl. Sci. NS-21, 78-93, 95., 1974.

[76] Gordon, R. & R.M. Rangayyan, Geometric deconvolution: a meta-algorithm for limited view computed tomography, IEEE Trans. Biomed. Eng. 30(12), 806-810, 1983.

[77] Gordon, R. & A.P. Dhawan, Three-dimensional computed tomography for nevi *in situ* by transillumination, Presented at Fourth International Conference on Automation of Diagnostic Cytology, Montreal, paper 66, June 24-25, 1983.

[78] Gordon, R., ed., Industrial Computed Tomography and NMR Imaging, Applied Optics, special issue (May 15) in press, 1985.

[79] Gordon, R. et al., eds., Topical Meeting on Industrial Applications of Computed Tomography and NMR Imaging, Washington, D.C.: Optical Society of America 1984.

[80] Gordon, R., A.P. Dhawan & R.M. Rangayyan, In defense of geometric deconvolution for computed tomography, IEEE Trans. Biomed. Eng., submitted, 1984.

[81] Greelhoed, G.W., A. Breslow, W.S. McCune, Malignant melanoma: Correlation of long term follow-up clinical staging, level of invasion, and thickness of primary tumor, Am. Surg. 43 77-85, 1977.

[82] Greene, M.H., Dysplastic nevi - precursor aspects, presented at the 43rd Annual Meeting of American Academy of Dermatology, Washington D.C., Dec. 1-6, 1984.

[83] Gull, S.F. & G.J. Daniell, Image reconstruction from incomplete and noisy data, Nature 272, 686-690, 1978.

[84] Gumport, S.L., M.N. Harris, D.F. Roses & A.W. Kopf, The diagnosis and management of common skin cancers, Cancer 31(2), pp. 79-90, 1981.

[85] Hais, I.M. & A. Strych, Increase in urocanic acid concentration in human epidermis following insolation, Coll Czech. Chem. Comm. 34, pp. 649-655, 1969.

[86] Hall, E.L., Automated computer diagnosis applied to lung cancer, Proc. 1972 Int. Conf. on Cybernetics and Society, New Orleans, 1972.

[87] Hall, E.L., Computer Image Processing and Recognition, New York: Academic Press, 1979.

[88] Hanson, K.M. & G.W. Wecksung, Baysian approach to limited-angle reconstruction in computed tomogrphy, J. Opt. Soc. Am. 73, pp.

1501-1509, 1983.

[89] Hardy, J.D., H.T. Hammell & D. Murgatroyd, Spetral transmittance and reflectance of excised human skin, J. Appl. Physiol. 9 257-264, 1956.

[90] Heffernan, P.B. & R.A. Robb, Image reconstruction from incomplete projection data: Iterative reconstruction-projection technique, IEEE Trans. Biomed. Eng. 30(12) pp. 828-831, 1983.

[91] Herman, G.T., A. Lent & S.W. Rowland, ART: mathematics and applications, report on mathematical foundations and on the applicability to real data of algebraic reconstruction techniques, J. Theor. Biol. 42(1), 1-32, 1973.

[92] Herman, G.T., Image Reconstruction from Projections: the Fundamentals of Computerized Tomography, New York: Academic Press, 1980.

[93] Humason, G.L., Animal Tissue Techniques, San Francisco: W.H. Freeman and Co., 4th ed., 1979.

[94] Inouye, T., Image reconstruction with limited angle projection data, IEEE Trans. Nucl. Sci. NS-26(2), 2666-2669, 1979.

[95] Jaman, K.A., R. Gordon & R.M. Rangayyan, Display of 3D anisotropic data from limited-view computed tomography, Computer Vision, Graphics, and Image Processing, submitted, 1985.

[96] Jaynes, E.T., On the rationals of maximum entropy methods, IEEE Proceedings 70, pp. 939-952, 1982.

[97] Kemp, M.C., Maximum entropy reconstruction in emission tomography, Medical Radionuclide Imaging 1, pp. 313-323, 1980.

[98] Kikuchi, R. & B.H. Soffer, Maximum entropy image restoration I: the entropy expression, J. Opt. Soc. Am. 67, pp. 1656-1665, 1977.

[99] King, M.A., R.B. Schwinger & P.W. Doherty, Fast Wiener digital post-processing of SPECT images, J.of Nuclear Med. 24(5), pp. 81-82, 1983.

[100] Klug, A. & R.A. Crowther, Three dimensional image reconstruction from the viewpoint of information theory, Nature 238, 435-440, 1972.

[101] Kopf, A.W. & R. Andrade, Benign juvenile melanoma, in Year Book of Dermatology, 1965-1966, Year Book of Med. Pub., 1966.

[102] Kopf, A.W. & O. Braun-Falco, directors, special forum, Malignant Melanoma, The 43rd Annual Meeting of American Academy of Dermatology, 1984.

[103] Kubelka, P., New contributions to the optics of intensely lightscattering materials Part II:nonhomogeneous layers, J. Opts. Soc. Am. 44 330-335, 1954.

[104] Lee, J. A. & A.P. Carter, Secular trends in mortality from melignant melanoma, J. Natl Ca Inst. 45 91-100, 1970.

[105] Lent, A., Maximum entropy and multiplicative art, Preprint: State University of New York at Buffalo, Dept. of Computer Science, 1976.

[106] Lupulescu, A.P., H. Pinkus, D.J. Brimingham, et al., Lentigo maligna of the fingertip, Arch. Dermatol. 107 717, 1973.

[107] Lutz, P.H., Fourier image reconstruction incorporating three single interpolation techniques, Department of Computer Science Technical Report 104, State University of New York at Buffalo, N.Y., 1975.

[108] for persons at high risk, The Skin Cancer Foundation, 475 Park Ave. South, New York, 1984.

[109] McGovern, V.J., R.A. Caldwell, C.A. Duncan, et al., Moles and melignant melanoma-terminology and classification, Med. J. Aus. 1 123, 1967.

[110] McGovern, V.J., The classification of melanoma and its relationship to prognosis, Pathology 2 85-98, 1970.

[111] McGovern, V.J., Malignant Melanoma: Clinical and Histological Diagnosis, John Wiley & Sons, New York, 1976.

[112] Medoff, B.P., W.R. Brody, M. Nass$ & A. Macovski, Iterative convolution backprojection algorithms for image reconstruction from limited data, J. Opt. Soc. Am. 73(11), pp. 1493-1500, 1983.

[113] Mihm, M.C., Jr., T.B. Fitzpatrick, M.M.L. Brown, J.W. Raker, R.A. Malt & J.S. Kaiser, Early detection of primary cutaneous malignant melanoma, a color atlas, New England J. Med. 289(19), 989-996, 1973.

[114] Miller, T.R. & G.T, Pack, The familial aspect of malignant melanoma, Arch. Dermatol. 86 35-39, 1962.

[115] Minerbo, G., Maximum entropy reconstruction from cone beam projection data, Comput. Biol. 9(1), 29-37, 1979.

[116] Mishima, Y., Cellular and subcellular activities in the ontogeny of nevocytic and melanocytic melanomas, In Montagna, W. & F. Hu (eds): The Pigmented System, Advances in Biology of Skin, vol 8, Oxford, New York, Pergamon, p.545, 1967.

[117] Mishima, Y., Cellular blue nevus: melanogenic activity and malignant transformation, Arch. Dermatol. 101 104, 1970.

[118] Morton, R. & S.S. Miller, Infrared transillumination using photography and television (videoscopy), J. Audiovisual Media Med. 4, 86-90, 1981.

[119] Nassi, M., W.R. Brody, B.P. Medoff & A. Macovski, Iterative reconstruction-reprojection: an algorithm for limited data cardiac-computed tomography, IEEE Trans. Biomed. Eng. BME-29(5), pp. 333-341, 1982.

[120] Pack, G.T., A clinical study of pigmented nevi and melanomas, , In: The Biology of Melanomas, vol.4, New York: New York Academy of Science, pp. 52-70, 1948.

[121] Pack, G.T. & J. Davis, The pigmented mole, Postgrad. Med. 27, 370-382, 1960.

[122] Pack, G.T. & T.R. Miller, Metastatic melanoma with intermediate

primary site, J. Amer. Med. Assoc. 176, 55-57, 1961.

[123] Papoulis, A., A new algorithm in spectral analysis and band-limited extrapolation, IEEE Trans. Circuits and Systems CAS-22, pp. 735-742, 1975.

[124] Parrish, J.A., R.R. Anderson, F. Urbach & D. Pitts, UV-A: Biological Effects of Ultraviolet Radiation with Emphasis on Human Responses to Longwave Ultraviolet, New York, Plenum Press, 1978.

[125] Pinkus, H. & A.H. Mehregan, A Guide to Dermatohistopathology, Appleton Century Crofts, New York, pp. 351-398, 1981.

[126] Puchtler, H., S.N. Meloan & L.D. Paschal, Infrared fluorescence microscopy of stained tissues: principles and technic, Histochem. 68, 211-230, 1980.

[127] Radon, J., On the determination of functions from their integrals along certain manifolds, Uber die Bestimmung von Funktionen durch ihre integralwerte langs gewisser Mannigfaltigkeiten, Ber. Saechs. Akad. Wiss. Leipzig, 69, 262-272, 1917.

[128] Ramachandran, G.N. & A.V. Lakshminarayan, Three dimensional reconstruction from radiographs and electron micrographs: application of convolutions instead of Fourier transform, Proc. Nat. Acad. Sci. U. S. 68(9), 2236-2240, 1971.

[129] Ramachandran, G.N. & A.V. Lakshminarayan, Three dimensional reconstruction from radiographs and electron micrographs: Part III Description & application of the convolution method, Pure Appl. Phys.

9, 997-1003, 1971.

[130] Rangayyan, R.M. & R. Gordon, Streak preventive image reconstruction with ART and adaptive filtering, IEEE Trans. Med. Imaging MI-1(3), 173-178, 1982.

[131] Rangayyan, R.M. & R. Gordon, Geometric deconvolution of artifacts in limited view computed tomography, In: Digest of Papers, Topical Meeting on Signal Recovery and Synthesis with Incomplete Information and Partial Constraints, Washington, D.C.: Optical Society of America, pp. FA2-1 to FA2-4, 1983.

[132] Rigel, D.S., R.J. Friedman & R. Rodriguez-Sains, eds., Recognizing and managing dysplastic and congenital nevi, The Melanoma Letter 2(3), The Skin Cancer Foundation, N.Y., 1984.

[133] Rosenfeld, A., Picture Processing by Computer, New York: Academic Press, 1969.

[134] Rosenfeld, A. & A.C. Kak, Digital Picture Processing, New York: Academic Press, 2nd ed., 2 vols., 1982.

[135] Rower, J.M., R.D. Carr & E.D. Lowney, Progressive cribriform and zosteriform hyperpigmentation, Arch. Dermatol. 114 98, 1978.

[136] Sato, T., S.J. Norton, M. Linzer, O. Ikeda & M. Hirama, Tomographic image reconstruction from limited projections using interactive revision in image and transform spaces, Applied Optics 20(3), 395-399, 1981.

[137] Scheuplein, R.J., A survey of some fundamental aspects of the

absorption and reflection of light by tissue, J. Soc. Cosmet. Chem. 15 111-122, 1964.

[138] Smith, K.T., D.C. Solomon & S.L. Wagner, Practical and mathematical aspects of the problem of reconstructing objects from radiographs, Bull. Amer. Math. Soc. 83, pp. 1227-1270, 1977.

[139] Smith, K.T., Reconstruction formulas in computed tomography, In: Computed Tomography, ed. L.A. Shepp, Providence, Rhode Island: American Mathematical Society, pp. 7-24, 1982.

[140] Snyder, D.L. & J. R. Cox, An overview of reconstruction tomography and limitations imposed by a finite number of projections, in Reconstructive Tomography in Diagnostic Radiology and Nuclear Medicine, eds. M.M. Ter-Pogossian et al., University Park Press, Baltimore, 1977.

[141] Sober, A.J., T.B. Fitzpatrick, M.C. Mihm, T.G. Wise, B.J. Pearson, W.H. Clark & A.W. Kopf, Early recognition of cutaneous melanoma, JAMA 242 2795-2799, 1979.

[142] Sober, A.J., M.D. Thomos, B. Fitzpatrick & M.C. Mihm, Primary melanoma of the skin: recognition and management, Am. Acad. Dermatol. 2 179-197, 1980.

[143] Sober, A.J., C.L. Day, A.W. Kopf & T.B. Fitzpatrick, Detection of "thin" primary melanomas, Ca-Cancer J. Clinicians 33(3), 160-163, 1983.

[144] Sober, A.J., Diagnosis and management of skin cancer, Cancer

51(12), 2448-2452, 1983.

[145] Sober, A.J., C.L. Day, T.H. Fitzpatrick, R.A. Lew, A.W. Kopf, M.C. Mihm, Early death from clinical stage I melanoma, J. Invest. Dermatol. 80(6, Suppl.), 50s-52s, 1983.

[146] Sober, A.J., C.L. Day, T.B. Fitzpatrick, R.A. Lew, A.W.Kopf & M.C. Mihm, Factors associated with death from melanoma from 2 to 5 years following diagnosis in clinical stage I patients, J. Invest. Dermatol. 80 (6 Suppl.) 53s-55s, 1983.

[147] Spitz, S., Melanomas in childhood, Am. J. Pathology 24, 591, 1948.

[148] Stewart, W.D., J.L. Danto & S. Maddin, Dermatology: Diagnosis and Treatment of Cutaneous Disorders, The C.V.Mosby Company, St. Louis, Fourth Edition, 1978.

[149] Tam, K.C. & V. Perez-Mendez, Limited-angle three dimensional reconstruction using Fourier transform iterations and Radon transform iterations, Opt. Eng. 20, pp. 586-589, 1981.

[150] Tam, K.C. & V. Perez-Mendez, Tomographic imaging with limited-angle input, J. Opt. Soc. Am. 71, pp. 582-592, 1981.

[151] Tam, K.C., The use of multispectral imaging in limited-angle reconstruction, IEEE Trans. Nuclear Science NS-29(1), pp. 512-515, 1982.

[152] Tan, C.Y., Comparison of xeroradiographic and ultrasound detection of corticosteroid induced dermal thinning, J. Invest.

Dermatol. 76, 126-128, 1981.

[153] Todor, G.A., Transillumination in infra-red light as a method of location of intraocular neoplasms in the pre-equatorial zone, Russian, Oftalmol. Zh. 33(8), 624-625, 1978.

[154] Vainshtein, B.K., Finding the structure of objects from projections, Kristallografiya 15(5), 894-902, 1970.

[155] Vainshtein, B.K. & S.S. Orlov, Theory of the recovery of functions from their projections, Kristallgrafiya 17(2), 253-257, 1972.

[156] Wanebo, H.J., J.G. Fortner, J. Woodruff, et al., Selection of stage 1 melanoma by depth of microinvasion: use of the combined microstage technique, Ann. Surg. 182 302-313, 1975.

[157] Webster, J.P., T.W. Stevenson & A.P. Stout, The surgical treatment of malignant melanomas of the skin, Surg. Clin. North Am. 24, 319, 1944.

[158] Wernecke, S.J. & L. R. D'Addario, Maximum entropy image reconstruction, IEEE Trans. Comput. C-26, pp. 686-690, 1977.

[159] Wick, M.M., A.J. Sober, T.B. Fitzpatrick, M.C. Mihm, A.W. Kopf, W.H. Clark & M.S. Blois, Clinical characteristics of early cutaneous melanoma, Cancer 45, 2684-2686, 1980.

[160] Wied, G.L., G.F. Bahr & P.H. Bartels, eds., The Automation of Uterine Cancer Cytology, Chicago: Tutorials of Cytology, 1976.

[161] Woringer, F., A propos d'un naevus actromique de la joue chez une filletre de 8 ans, Bull Soc. Fr. Dermatol Syphiligr 46, 550, 1939.

[162] Zucker, S., Region growing: childhood and adolescence, Computer Graphics Image Proc. 5, 382-399, 1976.

Dhawan, A.P., Nevoscopy: Three dimensional CT by transillumination to detect early melanoma, presented at the SCAMC Student Paper Competition in Eighth Annual Symposium on Computer Application in Medical Care, Washington D.C., November 4, 1984, winner of the "Martin N. Epstein Award" and the first prize (US $ 1,000).

APPENDIX-A

LISTING OF PROGRAMS

```
{       Z$NOBELLS$B$READP       ART1$T$XN$XL/end./$XN$XCOPYD      ARTO
*$FILE,ARTO,R$T$XCOPY           ART1             *$FILE,ART1,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/versio

The  above  XEDIT  commands  create two backup files and copy compiler
error messages to the end, as a comment, if any. }

{
      }
      {
      Title:ART:PLAIN ART
      Author:M.R.Rangaraj
      Address:Department of Pathology, University of Manitoba,
            Medical College, 770 Bannatyne avenue,
            Winnipeg, Manitoba, Canada R3E OW3.

                                  Program description :
      This program returns reconstructions of a given picture by
      summation ( back projection ) and additive/multiplicative ART.
      Parallel ray geometry is used throughout. Projections( views )
      of the picture are first computed and then using them, the
      picture is reconstructed. Parameters of the picture and
      reconstruction method to be used are to be entered in the
      constants list and within the main program as well.

      }
      {

                                    Program Statement

      }
      program artp(input/,inpic,output,prompt,lstfrm,nxtfrm,calcom,mvp1,mvp2,
      mvp3) ;
      {
                                  Use of External Files

      input:   terminal input
      output:  debugging output and Pascal dump on error exits
      prompt:  terminal output
      lstfrm:  last frame recorded
      nxtfrm:  next frame to be output by this program
      calcom:  output file for PASPLOT commands
      inpic: input test picture
      mvp1:  test picture returned in MVP format 5
      mvp2:  reconstruction by summation
      mvp3:  ART reconstruction

      Execute as:
      /ARTX,INPUT,INPIC,ARTU,OUTPUT,ARTI,ARTO,ARTZ,MVP1,MVP2,
      MVP3.
      }
      {

                                      Associated Files

      ARTC   for execution
ARTM   for batch compilation
ARTS for batch job submission
}
{

                    Compiler Include Directives

}
{$I'PASPLOT'}
{

                              Labels

}
{

                             Constants

}
Const
{A constants:}
{ for additive ART set artadd to true ; multiplicative ART set artmul to true ;
for constrained ART set artconstr to true ; set others to false }
artadd=false;artmul=true; artconstr=false ;
{B constants:}
{C constants:}
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
{ max (+-) y/j index of picture ( origin at center of picture ) }
mj=63 ;
{N constants:}
{ No. of rows and columns of picture and no. of bits per word in MVP output file
}
norow=127 ; nocol=127 ; nobits=8 ;
{
ni:max(+-) x/i index of picture; np:no. of projections/views to be used;
nr:max(+-) index no. of rays; nr2:total no. of rays per view(2*nr+1) ;
nprmax:max no. of pixels a ray can cross - raywidth in pixels * max of
norow,nocol .
}
ni=63;np=5; nr=127; nr2=255;nprmax=127;
{O constants:}
{P constants:}
pi = 3.1415926536;
{Q constants:}
{R constants:}
{S constants:}
{T constants:}
```

```
{U constants:}
{V constants:}
version_number = 1;
{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                                 Types

     Primary types, defined in terms of constants,  are  given  first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                             Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
twod=array[integer,integer] of real ;
twod1=array[1..np,-nr..nr] of real ;
twod2=array[-ni..ni,-mj..mj] of real ;
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
                             Compound Types
}
{

                               Variables

}
```

```
Var
{A variables:}
averagepic: text ;
angl,avg:real ;
{B variables:}
bufferpic: text ;
{C variables:}
counti:integer ;
countj:integer ;
cosl:real ;
calcom: calcomfile; {Needed by PASPLOT}
{D variables:}
{E variables:}
eucliderr : real ;
error: real ;
{F variables:}
{G variables:}
{H variables:}
{I variables:}
i,ip,imin,imax,inpr,iteration:integer ;
inpic: text ;
{J variables:}
j,jj1,jj2,jj3,jj4,jmin,jmax,jp:integer ;
{K variables:}
k:integer ;
{L variables:}
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
mvp1,mvp2,mvp3:text ;
{N variables:}
nxtfrm: text; {See program statement}
npr:integer ;
{O variables:}
{P variables:}
preerr:real ; {error at previous iteration }
{ pseudo projection of ray }
pseudop: real ;
{ picture array -ni..ni, -mj..mj }
pic:twod2 ;
{ angles of projections : to be entered in main program }
prangl: array[1..np] of real ;
{ first dimension refers to view no. and second to ray no. }
projection:text ;
pixel:array[1..2,1..nprmax] of integer ;
prompt: text; {Call to program should be
{}              /ARTX,,ARTU,OUTPUT, etc., to
{}             place any bugs into file ARTU and have the prompts
{}             come out on OUTPUT. All prompts are to file PROMPT
{}             which thereby becomes OUTPUT.}
{Q variables:}
{R variables:}
raysum:real ;
{ ray width : modify in main program at all locations if necessary }
rw:real ;
{S variables:}
```

```
spot:real ;
sinl:real ;
scale: real; {Scale factor for plotting}
{T variables:}
{ threshold for ray boundary = 0.5 * ray width }
thr:real ;
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
version_date: alfa; {Date compiled}
version_time: alfa; {Time compiled}
val:real ;
valelp:real ;
{W variables:}
{X variables:}
xinfo:scaleinfo ;
{Y variables:}
yinfo:scaleinfo ;
{Z variables:}
{

                           Forward Declarations

}
{Forward declarations of procedures and functions:}
{A forward declarations:}
function Amaxl(a,b: real): real; forward;
function Aminl(a,b: real): real; forward;
{B forward declarations:}
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
function Factorial(n: integer): real; forward;
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
function Max0(a,b: integer): integer; forward;
function Min0(a,b: integer): integer; forward;
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
```

```
{

                    Procedures and Functions

}
{Procedures and functions:}
{A procedures and functions:}
{

                              Absmal

}
function absmal(one,two:integer):integer ;
{ returns no. with smaller absolute value
and sign of the first no. }
  begin { absmal }
  if one < 0 then
    begin
    if abs(one) < two then absmal:=one else absmal:=-two
    end
  else
    begin
    if one < two then absmal:=one else absmal:=two
    end
  end ; { absmal }
{

                              Amaxl

}
function Amaxl{(a,b: real): real};
  begin {Amaxl}
  if a > b then Amaxl := a else Amaxl := b;
  end; {Amaxl}
{

                              Aminl

}
function Aminl{(a,b: real): real};
  begin {Aminl}
  if a < b then Aminl := a else Aminl := b;
  end; {Aminl}
{B procedures and functions:}
{D procedures and functions:}
{

                               del

}
function del(i,j,k:integer;sinl,cosl:real):boolean ;
{ del : true if pixel(i,j) belongs to ray (l,k) }
var dist,jy:real ;
  begin  { del  }
  jy:=i*sinl + k*rw  ; jy:= jy/cosl ;
  dist:= abs(j-jy)*abs(cosl);
  if dist<=thr then del:=true else del:=false
  end ;  { del }
{E procedures and functions:}
{

                            euclid_error ;

}
procedure euclid_error ;
{ computes the Euclidean distance between the original picture
and its reconstruction }
  begin { euclid_error }
  eucliderr:=0.0 ;
  reset(inpic) ;
  for j:=-mj to mj do
    begin { row }
    for i:=-ni to ni do
      begin { pixels }
      read(inpic,val) ;
      val:=val-pic[i,j];
      eucliderr:=eucliderr + val*val ;
      end ; { pixels }
    readln(inpic) ;
    end ; { row }
  eucliderr:=eucliderr/(norow*nocol) ;
  end ; { euclid_error }
{F procedures and functions:}
{

                              Factorial

}
function Factorial{(n: integer): real};
  begin {Factorial}
  if n = 0 then Factorial := 1
  else Factorial := n * Factorial(n-1);
  end; {Factorial}
{

                                formvp

}
procedure formvp( var fig:dynamic twod;nrow,ncol,bits:integer; var mvpfil:text);
{ writes picture in MVP format 5 in file mvpfil }
var ij,jk,newval,twelv:integer ; maxel,minel,lim:real ;okay:boolean ;
  begin { for mvp }
  okay:=true ;
  if (nrow<1) or (nrow>512) or (ncol<1) or (ncol>512) then
    begin { error 2 }
    writeln(prompt,'Maximum picture size permitted is 512 X 512') ;
    writeln(output,'Maximum picture size permitted is 512 X 512') ;
    okay:=false ;
    end ; { error 2 }
  if (bits<1) or (bits>12) then
    begin { error 3 }
    writeln(prompt,'MVP resolution limited to 1..12 bits') ;
  writeln(output,'MVP resolution limited to 1..12 bits') ;
  okay:=false ;
  end ; {error 3 }
 if okay then
  begin { execute formvp }
  maxel:=0 ;
   case bits of  { resolution }
   1: lim:=1 ;
   2: lim:=3 ;
   3: lim:=7 ;
   4: lim:=15 ;
   5: lim:=31 ;
   6: lim:=63 ;
   7: lim:=127 ;
   8: lim:=255 ;
   9: lim:=511 ;
   10:lim:=1023 ;
   11:lim:=2047 ;
   12:lim:=4095 ;
   end ;    { resolution }
  ij:=low(fig,1) ; jk:=low(fig,2) ; minel:=fig[ij,jk] ;
  for ij:=low(fig,1) to high(fig,1)do
  for jk:=low(fig,2) to high(fig,2) do
   begin { minmax }
   if maxel<fig[ij,jk] then maxel:=fig[ij,jk] ;
   if minel>fig[ij,jk] then minel:=fig[ij,jk] ;
   end ; { minmax }
  writeln(mvpfil,'PICTURE FILE FOR DISPLAY USING MVP.') ;
  writeln(mvpfil,'ENDOFHEADR',ncol:5,nrow:5,bits:5,1:5) ;
  for jk:=low(fig,2) to high(fig,2) do
   begin { row }
   twelv:=0 ;
   for ij:=low(fig,1) to high(fig,1) do
    begin { col }
    {linear transformation from 0-maxel to 0-lim}
    newval:=round(lim*fig[ij,jk]/maxel) ;
    if newval<0 then newval:=0 ;
    write(mvpfil,newval:5) ;
    twelv:=twelv+1 ;
    if twelv=12 then
     begin { end line }
     twelv:=0 ;
     writeln(mvpfil) ;
     end ; { end line }
    end ; { col }
   writeln(mvpfil) ;
   end ; { row }
  end ; { execute formvp }
 end ; { for mvp }
{G procedures and functions:}
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{

                    jminmax
}
procedure jminmax(n1,n2,n3,n4 :integer ; var min,max:integer ) ;
{ returns min and max of four integers given }
  begin { jminmax }
  min:=n1 ; max:=n1 ;
  if n2<min then min:=n2 ;
  if n3<min then min:=n3 ;
  if n4<min then min:=n4 ;
  if n2>max then max:=n2 ;
  if n3>max then max:=n3 ;
  if n4>max then max:=n4 ;
  end ; { jminmax }
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                              Max0

}
function Max0{(a,b: integer): integer};
  begin {Max0}
  if a > b then Max0 := a else Max0 := b;
  end; {Max0}
{

                              Min0

}
function Min0{(a,b: integer): integer};
  begin {Min0}
  if a < b then Min0 := a else Min0 := b;
  end; {Min0}
{N procedures and functions:}
{O procedures and functions:}
{P procedures and functions:}
{Q procedures and functions:}
{R procedures and functions:}
{

                           raypixels

}
procedure raypixels(l,k:integer ) ;
{
Parallel ray geometry. Returns nc. of pixels covered by ray(l,k) in npr,
their coords in pixel[1,i],pixel[2,i] and the pseudo projection in
pseudo. Raywidth rw, threshold thr, sin and cos of view angle sinl,cosl
to be defined in main program before calling raypixels.
Calls absmal,del,jminmax,min0,max0.
}
  begin { raypixels }
  npr:=0 ; pseudop:=0 ;
  { get limits of search in y/j direction }
  jjl:=round((-ni*sinl+(k-1)*rw)/cosl) ;
```

```
 jj2:=round((-ni*sinl+(k+1)*rw)/cosl) ;
 jj3:=round((ni*sinl+(k-1)*rw)/cosl) ;
 jj4:=round((ni*sinl+(k+1)*rw)/cosl) ;
 jj1:=absmal(jj1,mj) ; jj2:=absmal(jj2,mj) ;
 jj3:=absmal(jj3,mj) ; jj4:=absmal(jj4,mj) ;
 jminmax(jj1,jj2,jj3,jj4,jmin,jmax) ;
 for j:=jmin to jmax do
  begin { row }
  { get limits of search in x/i direction }
  jj1:=round((j*cosl-(k-1)*rw)/sinl) ; jj1:=absmal(jj1,ni-1) ;
  jj2:=round((j*cosl-(k+1)*rw)/sinl) ; jj2:=absmal(jj2,ni-1) ;
  imin:=min0(jj1,jj2)-1 ; imax:=max0(jj1,jj2)+1;
  for i:=imin to imax do
   begin  { column }
   if del(i,j,k,sinl,cosl) then
    begin  { store ray pixels }
    npr:=npr+1 ; pseudop:=pseudop+pic[i,j] ;
    pixel[1,npr]:=i ; pixel[2,npr]:=j ;
    end;  { store ray pixels }
   end ; { column }
  end ; { row }
 end ; { raypixels }
{S procedures and functions:}
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{

               Main Block

}
 begin {artp}
 rewrite(calcom) ;
 reset(inpic) ;
 writeln(prompt,' START TIME ' ,clock:10) ;
 for j:=-mj to mj do
  begin
  for i:=-ni to ni do
  pic[i,j]:=0 ;
  end ;
 for j:=-mj to mj do
  begin { get pic }
  for i:=-ni to ni do
  if (i=0) and (j=0) then pic[i,j]:=255 else pic[i,j]:=25 ;
  end ;
{
  read(inpic,pic[i,j]) ;
  readln(inpic) ;
  end ;
}
{
   begin {elps}
   valelp:=i*i/1600+j*j/900 ;
   if valelp<=1.0 then pic[i,j]:=255 else
   if j<0 then pic[i,j]:=10 else pic[i,j]:=20 ;
   end ; {elps}
```

```
    end ; { get pic }
}
{
      begin {rectangle}
      pic[i,j]:=100 ;
      if j>27 then pic[i,j]:=160 ;
      if i>35 then pic[i,j]:=160 ;
      if j<-10 then pic[i,j]:=160 ;
      if i<-30 then pic[i,j]:=160 ;
      valelp:=(j-10)*(j-10)/100+i*i/400 ;
      if valelp<=1 then pic[i,j]:=150 ;
      end ; {rectangle}
    end ; {get pic}
}
{
  for j:=-mj to mj do
    begin {circle}
    for i:=-ni to ni do
      begin
      valelp:=j*j/400+i*i/400 ;
      if valelp<=1.0 then pic[i,j]:=255 ;
      end ;
    end ; {circle}
}
{
  for j:=-mj to mj do
    begin
    for i:=-ni to ni do
      begin
      valelp:=(j-15)*(j-15)/100+(i-15)*(i-15)/100 ;
      if valelp<=1.0 then pic[i,j]:=255 ;
      end ;
    end ;
  for j:=-mj to mj do
    begin
    for i:=-ni to ni do
      begin
      valelp:=(j+15)*(j+15)/100+(i-15)*(i-15)/100 ;
      if valelp<=1.0 then pic[i,j]:=255 ;
      end ;
    end ;
  for j:=-mj to mj do
    begin
    for i:=-ni to ni do
      begin
      valelp:=(j+15)*(j+15)/100+(i+15)*(i+15)/100 ;
      if valelp<=1.0 then pic[i,j]:=255 ;
      end ;
    end ;
  for j:=-mj to mj do
    begin
    for i:=-ni to ni do
      begin
      valelp:=(j-15)*(j-15)/100+(i+15)*(i+15)/100 ;
      if valelp<=1.0 then pic[i,j]:=255 ;
      end ;
    end ;
}
{
  counti:=1 ;
  countj:=1 ;
  for j:=-mj to mj do
    begin
    countj:=countj+1 ;
    for i:=-ni to ni do
      begin
      counti:=counti+1 ;
      if (counti<=10) and (countj<=10) then pic[i,j]:=127 else
      if (counti>10) and (countj>10) then pic[i,j]:=127 else
      pic[i,j]:=200 ;
      if counti=20 then counti:=1 ;
      if countj=20 then countj:=1 ;
      end ;
    end ;
  for j:=-mj to mj do
    begin
    for i:=-ni to ni do
      begin
      valelp:=i*i/1600+j*j/625 ;
      if valelp>1.0 then pic[i,j]:=20 ;
      end ;
    end ;
}
  rewrite(mvpl) ;
  formvp(pic,2*mj+1,2*ni+1,8,mvpl) ;
  { define projection angles in degrees }
  for l:=1 to np do prangl[l]:=45+(l-1)*22.5 ;
{
  prangl[1]:=45 ;
  prangl[2]:=90 ;
  prangl[3]:=180 ;
}
  { }
  {     ..........##########################................   }
  { }
  { compute projections }
  rewrite(projection);
  avg:=0 ;
  for l:= 1 to np do
    begin  { np projections }
    angl:= prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    { ray width rw may be multiplied by an integer to use wider rays ( more
    pixels per ray : modify nprmax ). Perform corrections at all occurrences
    of
    rw}
    rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:= -nr to nr do
      begin  { ray }
      raypixels(l,k) ;
      write(projection,pseudop) ;
      avg:=avg + pseudop ;
      end ;  { ray }
    writeln(projection) ;
    end ;  { np projections }
  reset(projection) ;
  avg:=avg/(norow*nocol*np) ;
  writeln(prompt,' AFTER PROJECTIONS ',clock:10) ;
  { }
  {   ..............#########################.................   }
  { }
  { Reconstruction by  ART }
  for i:=-ni to ni do
  for j:=-mj to mj do pic[i,j]:=avg ;
  iteration:=0 ;error:=0.0 ; preerr:=9999999.9 ;
  repeat  { ART }
  reset(projection) ;
  if iteration>0 then preerr:=error ;
  iteration:=iteration+1 ;
  for l:= 1 to np do
    begin  { projection  }
    angl:=prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:=-nr to nr do
      begin  { ray }
      read(projection,raysum) ;
      raypixels(l,k) ;
      for inpr:=1 to npr do
        begin { correction }
        ip:=pixel[1,inpr] ; jp:=pixel[2,inpr] ;
        if artadd then
          begin { additive ART }
          pic[ip,jp]:=pic[ip,jp]+(raysum-pseudop)/npr ;
          { if constrained ART, check for negative values }
          if (artconstr) and (pic[ip,jp]<0) then pic[ip,jp]:= 0 ;
          end ; { additive ART }
        if artmul then
          begin { multiplicative ART }
          if pseudop<>0.0 then
          pic[ip,jp]:= pic[ip,jp]*raysum/pseudop
          else pic[ip,jp]:=0.0 ;
          end ; { multiplicative ART }
        end ; { correction }
      end ; { ray }
    readln(projection) ;
    end;  { projection }
{
    euclid_error ;
}
  writeln(prompt,' ITERATION ',iteration:3,' EUCLIDEAN ERROR ',euclidderr) ;
  { a suitable criterion on the error may be included here to stop ART }
  until (iteration=5);  { ART }
  rewrite(mvp3) ;
  formvp(pic,2*mj+1,2*ni+1,8,mvp3)  ;
  writeln(prompt,' CPU TIME USED ',clock:10,' MILLISECONDS.') ;
  end.  { artp }
```

```
{      Z$NOBELLS$B$READP     SPARTI$T$XN$XL/end./$XN$XCOPYD     SPARTO
*$FILE,SPARTO,R$T$XCOPY          SPARTI         *$FILE,SPARTI,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/versio

project-  3d  reconsruction  of  nevi  ;  atam prakash dhawan graduate
student UNIVERSITY OF MANITOBA The above  XEDIT  commands  create  two
backup  files  and  copy  compiler  error  messages  to  the end, as a
comment, if any. }

{
      }
      {
      Title: SPART: WEIGHTED ART WITH AVERAGING
      Author:M.R.Rangaraj
      Address:Department of Pathology, University of Manitoba,
            Medical College, 770 Bannatyne avenue,
            Winnipeg, Manitoba, Canada R3E 0W3.

                                    Program description :
      This program returns reconstructions of a given picture by
      summation ( back projection ) and additive/multiplicative ART
      combined with a selective neighborhood averaging to prevent streaks.
      Parallel ray geometry is used throughout. Projections( views )
      of the picture are first computed and then using them, the
      picture is reconstructed. Parameters of the picture and
      reconstruction method to be used are to be entered in the
      constants list and within the main program as well.

      }
      {

                                      Program Statement

      }
      program spartp(input/,inpic,limits,output,prompt,lstfrm,nxtfrm,calcom,
      mvp1,mvp2,mvp3,view1,view2,view3,view4,view5,view6,view7,view8) ;
      {
                                    Use of External Files

      input:   terminal input
      output:  debugging output and Pascal dump on error exits
      prompt:  terminal output
      lstfrm:  last frame recorded
      nxtfrm:  next frame to be output by this program
      calcom:  output file for PASPLOT commands
      inpic: input test picture
      mvp1:  test picture returned in MVP format 5
      mvp2:  reconstruction by summation
      mvp3:  ART reconstruction

      Execute as:
      /SPARTX,INPUT,INPIC,LIMITS,SPARTU,OUTPUT,SPARTI,SPARTO,SPARTZ,
      MVP1,MVP2,MVP3.
      }
      {
```

```
                         Associated Files

SPARTC   for execution
SPARTM   for batch compilation
SPARTS for batch job submission
}
{

                    Compiler Include Directives

}
{$I'PASPLOT'}
{

                              Labels

}
{

                            Constants

}
Const
{A constants:}
{ for additive ART set artadd to true ; multiplicative ART set artmul to true ;
for constrained ART set artconstr to true ; set others to false }
artadd=true ; artmul=false ; artconstr=true ;
{B constants:}
{C constants:}
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
{ max (+-) y/j index of picture ( origin at center of picture ) }
mj=50 ;
{N constants:}
{ No. of rows and columns of picture and no. of bits per word in MVP output file
}
norow=101 ; nocol=101 ; nobits=8 ;
{
ni:max(+-) x/i index of picture; np:no. of projections/views to be used;
nr:max(+-) index no. of rays; nr2:total no. of rays per view(2*nr+1) ;
nprmax:max no. of pixels a ray can cross - raywidth in pixels * max of
norow,nocol .
}
ni=50;np=3;nr=110;nr2=221;nprmax=101;
{O constants:}
{P constants:}
pi = 3.1415926536;
{Q constants:}
```

```
{R constants:}
{S constants:}
{T constants:}
{U constants:}
{V constants:}
version_number = 1;
{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                                 Types

     Primary types, defined in terms of constants,  are  given  first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                             Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
twod=array[integer,integer] of real ;
twod1=array[1..np,-nr..nr] of real ;
twod2=array[-ni..ni,-mj..mj] of real ;
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
                             Compound Types
}
{
```

Variables

```
}
Var
{A variables:}
along,across : real ; {contrast along & across a ray}
averagepic:text ;
avgwt:real ; { weighting factor for neighbors' average }
angl,avg:real ;
{B variables:}
bufferpic : text ;
{C variables:}
cosl:real ;
calcom: calcomfile; {Needed by PASPLOT}
{D variables:}
{E variables:}
euclidderr:real ;
error: real ;
valelp:real ;
{F variables:}
{G variables:}
{H variables:}
{I variables:}
i,ip,imin,imax,inpr,iteration:integer ;
inpic: text ;
{J variables:}
j,jj1,jj2,jj3,jj4,jmin,jmax,jp:integer ;
{K variables:}
k:integer ;
{L variables:}
limits:text ; { input file containing streak contrast thresholds }
ltrt:real ; {left to right contrast across a ray}
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
midpix:real ;
mvpl,mvp2,mvp3:text ;
{N variables:}
nsmooth,nart,nsharpen : integer ;
nextray:array[1..nprmax]of real ; {neighboring pixels of next ray }
nnexpix,nprepix : integer ; { number of neighboring pixels in next & previous
rays }
{ string of average of neighbor's of a pixel }
neighborhood: array [1..nprmax] of real ;
{ average of neighbor array over all pixels of a ray }
neighbor_total : real ;
nxtfrm: text; {See program statement}
npr:integer ;
{O variables:}
{P variables:}
pixpre,pixnex : real ; {previous & next pixels along ray}
prevray:array[1..nprmax] of real ; { neighboring pixels of previous ray }
projl:integer ;
preerr: real ; { error of previous iteration }
picavg: real ; { average density of picture }
{ pseudo projection of ray }
```

```
pseudop: real ;
{ picture array -ni..ni, -mj..mj }
pic:twod2 ;
{ angles of projections : to be entered in main program }
prangl: array[1..np] of real ;
{ first dimension refers to view no. and second to ray no. }
projection:text ;
pixel:array[1..2,1..nprmax] of integer ;
prompt: text; {Call to program should be
{}              /SPARTX,,SPARTU,OUTPUT, etc., to
{}             place any bugs into file SPARTU and have the prompts
{}             come out on OUTPUT. All prompts are to file PROMPT
{}             which thereby becomes OUTPUT.}
{Q variables:}
{R variables:}
raysum: real ;
{ ray width : modify in main program at all locations if necessary }
rw:real ;
{S variables:}
sinl:real ;
scale: real; {Scale factor for plotting}
{T variables:}
thresh1,thresh2 : real ;
{ threshold for ray boundary = 0.5 * ray width }
thr:real ;
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
view1,view2,view3,view4,view5,view6,view7,view8: text ;
version_date: alfa; {Date compiled}
version_time: alfa; {Time compiled}
val:real ;
{W variables:}
weight : real ; { weighting factor for average }
{X variables:}
x:array[1..nr2] of real ;
xinfo:scaleinfo ;
{Y variables:}
y:array[1..nr2] of real ;
yinfo:scaleinfo ;
{Z variables:}
{

                         Forward Declarations

}
{Forward declarations of procedures and functions:}
{A forward declarations:}
function Amax1(a,b: real): real; forward;
function Amin1(a,b: real): real; forward;
{B forward declarations:}
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
function Factorial(n: integer): real; forward;
```

```
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
function Max0(a,b: integer): integer; forward;
function Min0(a,b: integer): integer; forward;
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
{

                       Procedures and Functions

}
{Procedures and functions:}
{A procedures and functions:}
{

                                Absmal

}
function absmal(one,two:integer):integer ;
{ returns no. with smaller absolute value
and sign of the first no. }
  begin { absmal }
  if one < 0 then
    begin
    if abs(one) < two then absmal:=one else absmal:=-two
    end
  else
    begin
    if one < two then absmal:=one else absmal:=two
    end
  end ; { absmal }
{

                                Amax1

}
function Amax1{(a,b: real): real};
  begin {Amax1}
  if a > b then Amax1 := a else Amax1 := b;
  end; {Amax1}
```

```
{

                                Aminl

}
function Aminl{(a,b: real): real};
  begin {Aminl}
  if a < b then Aminl := a else Aminl := b;
  end; {Aminl}
{B procedures and functions:}
{C procedures and functions:}
{D procedures and functions:}
{

                                 del

}
function del(i,j,k:integer;sinl,cosl:real):boolean ;
{ del : true if pixel(i,j) belongs to ray (l,k) }
var dist,jy:real ;
  begin  { del  }
  jy:=i*sinl + k*rw  ; jy:= jy/cosl ;
  dist:= abs(j-jy)*abs(cosl);
  if dist<=thr then del:=true else del:=false;
  end ;  { del }
{E procedures and functions:}
{

                            euclid_error ;

}
procedure euclid_error ;
{ computes the Euclidean distance between the original picture in file inpic
and its reconstruction }
  begin { euclid_error }
  eucliderr:=0.0 ;
  reset(inpic) ;
  for j:=-mj to mj do
    begin { row }
    for i:=-ni to ni do
      begin { pixels }
      read(inpic,val) ;
      if pic[i,j]<>-999 then
      val:=val-pic[i,j];
      eucliderr:=eucliderr + val*val ;
      end ; { pixels }
    readln(inpic) ;
    end ; { row }
  eucliderr:=eucliderr/(norow*nocol);
  end ; { euclid_error }
{F procedures and functions:}
{

                                filter

}
```

```
procedure filter ;
  begin { filter }
  for j:=-mj+1 to mj-1 do
  for i:=-ni+1 to ni-1 do
  if pic[i,j]<>-999 then
    begin { operations }
    avg:=0 ;
    for jj1:=-1 to 1 do
    for jj2:=-1 to 1 do
      begin { neighbors }
      jj3:=i+jj1 ; jj4:=jj2+j ;
      if pic[jj3,jj4]<>-999 then
      avg:=avg + pic[jj3,jj4] ;
      end ; { neighbors }
    neighbor_total:=avg ;
    avg:=avg - pic[i,j] ;
    avg:= avg/8 ;
    if neighbor_total=0 then val:=0
    else val:= abs(pic[i,j]-avg)/neighbor_total ;
    if val>=thresh2 then pic[i,j]:=avg ;
    end ; { operations }
  end ; { filter }
{

                                        Factorial

}
function Factorial{(n: integer): real};
  begin {Factorial}
  if n = 0 then Factorial := 1
  else Factorial := n * Factorial(n-1);
  end; {Factorial}
{

                                          formvp

}
procedure formvp( var fig:dynamic twod;nrow,ncol,bits:integer; var mvpfil:text);
{ writes picture in MVP format 5 in file mvpfil }
var ij,jk,newval,twelv:integer ; maxel,minel,lim:real ;okay:boolean ;
  begin { for mvp }
  okay:=true ;
  if (nrow<1) or (nrow>512) or (ncol<1) or (ncol>512) then
    begin { error 2 }
    writeln(prompt,' MAXIMUM PICTURE SIZE PERMITTED IS 512 X 512') ;
    writeln(output,'Maximum picture size permitted is 512 X 512') ;
    okay:=false ;
    end ; { error 2 }
  if (bits<1) or (bits>12) then
    begin { error 3 }
    writeln(prompt,' MVP RESOLUTION LIMITED TO 1..12 BITS') ;
    writeln(output,'MVP resolution limited to 1..12 bits') ;
    okay:=false ;
    end ; {error 3 }
  if okay then
    begin { execute formvp }
  maxel:=0 ;
   case bits of  { resolution }
   1: lim:=1 ;
   2: lim:=3 ;
   3: lim:=7 ;
   4: lim:=15 ;
   5: lim:=31 ;
   6: lim:=63 ;
   7: lim:=127 ;
   8: lim:=255 ;
   9: lim:=511 ;
   10:lim:=1023 ;
   11:lim:=2047 ;
   12:lim:=4095 ;
   end ;    { resolution }
  ij:=low(fig,1) ; jk:=low(fig,2) ; minel:=fig[ij,jk] ;
  for ij:=low(fig,1) to high(fig,1)do
  for jk:=low(fig,2) to high(fig,2) do
  if fig[ij,jk]<>-999 then
   begin { minmax }
   if maxel<fig[ij,jk] then maxel:=fig[ij,jk] ;
   if minel>fig[ij,jk] then minel:=fig[ij,jk] ;
   end ; { minmax }
  writeln(mvpfil,'PICTURE FILE FOR DISPLAY USING MVP.') ;
  writeln(mvpfil,'ENDOFHEADR',ncol:5,nrow:5,bits:5,1:5) ;
  for jk:=low(fig,2) to high(fig,2) do
   begin { row }
   twelv:=0 ;
   for ij:=low(fig,1) to high(fig,1) do
    begin { col }
    if fig[ij,jk]=-999 then newval:=0
    else
    newval:=round(lim*fig[ij,jk]/maxel) ;
    if newval<0 then newval:=0 ;
    write(mvpfil,newval:5) ;
    twelv:=twelv+1 ;
    if twelv=12 then
     begin { end line }
     twelv:=0 ;
     writeln(mvpfil) ;
     end ; { end line }
    end ; { col }
   writeln(mvpfil) ;
   end ; { row }
  end ; { execute formvp }
 end ; { for mvp }
{G procedures and functions:}
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{

                jminmax

}
procedure jminmax(n1,n2,n3,n4 :integer ; var min,max:integer ) ;
```

```
{ returns min and max of four integers given }
  begin { jminmax }
  min:=n1 ; max:=n1 ;
  if n2<min then min:=n2 ;
  if n3<min then min:=n3 ;
  if n4<min then min:=n4 ;
  if n2>max then max:=n2 ;
  if n3>max then max:=n3 ;
  if n4>max then max:=n4 ;
  end ; { jminmax }
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                                 Max0

}
function Max0{(a,b: integer): integer};
  begin {Max0}
  if a > b then Max0 := a else Max0 := b;
  end; {Max0}
{

                                 Min0

}
function Min0{(a,b: integer): integer};
  begin {Min0}
  if a < b then Min0 := a else Min0 := b;
  end; {Min0}
{N procedures and functions:}
{O procedures and functions:}
{P procedures and functions:}
{Q procedures and functions:}
{R procedures and functions:}
{

                              raypixels

}
procedure raypixels(l,k:integer ) ;
{
Parallel ray geometry. Returns no. of pixels covered by ray(l,k) in npr,
their coords in pixel[1,i],pixel[2,i] and the pseudo projection in
pseudo. Raywidth rw, threshold thr, sin and cos of view angle sinl,cosl
to be defined in main program before calling raypixels.
Calls absmal,del,jminmax,min0,max0.
}
  begin { raypixels }
  npr:=0 ; pseudop:=0 ;
  { get limits of search in y/j direction }
  jj1:=round((-ni*sinl+(k-1)*rw)/cosl) ;
  jj2:=round((-ni*sinl+(k+1)*rw)/cosl) ;
  jj3:=round((ni*sinl+(k-1)*rw)/cosl) ;
  jj4:=round((ni*sinl+(k+1)*rw)/cosl) ;
```

```
 jj1:=absmal(jj1,mj) ; jj2:=absmal(jj2,mj) ;
 jj3:=absmal(jj3,mj) ; jj4:=absmal(jj4,mj) ;
 jminmax(jj1,jj2,jj3,jj4,jmin,jmax) ;
 for j:=jmin to jmax do
  begin { row }
  { get limits of search in x/i direction }
  jj1:=round((j*cosl-(k-1)*rw)/sinl) ; jj1:=absmal(jj1,ni-1) ;
  jj2:=round((j*cosl-(k+1)*rw)/sinl) ; jj2:=absmal(jj2,ni-1) ;
  imin:=min0(jj1,jj2)-1 ; imax:=max0(jj1,jj2)+1;
  for i:=imin to imax do
   begin  { column }
   if del(i,j,k,sinl,cosl) and (pic[i,j] <> -999) then
    begin  { store ray pixels }
    npr:=npr+1 ; pseudop:=pseudop+pic[i,j] ;
    pixel[1,npr]:=i ; pixel[2,npr]:=j ;
    end;  { store ray pixels }
   end ; { column }
  end ; { row }
 end ; { raypixels }
{S procedures and functions:}
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{

              view_average

}
procedure view_average ;
{ produces a picture which is the average of reconstructions
at end of each view }
 begin { view average }
 if l=1 then
  begin { initiate averagepic }
  rewrite(averagepic) ;
  for j:=-mj to mj do
   begin { row }
   for i:=-ni to ni do
   write(averagepic,pic[i,j]) ;
   writeln(averagepic) ;
   end ; { row }
  end  { initiate averagepic }
 else
  begin { add next view picture }
  reset(averagepic) ;
  rewrite(bufferpic) ;
  for j:=-mj to mj do
   begin { row }
   for i:= -ni to ni do
    begin { pixel }
    read(averagepic,val) ;
    val:=val + pic[i,j] ;
    write(bufferpic,val) ;
    end ; { pixel }
   readln(averagepic) ;
   writeln(bufferpic) ;
   end ; { row }
  rewrite(averagepic) ;
  reset(bufferpic) ;
  for j:= -mj to mj do
   begin { swap files }
   for i:=-ni to ni do
    begin { interchange }
    read(bufferpic,val) ;
    write(averagepic,val) ;
    end ; { interchange }
   readln(bufferpic) ;
   writeln(averagepic) ;
   end ; { swap files }
  end ; { add next view picture }
 if l=np then
  begin { transfer picture }
  reset(averagepic) ;
  for j:=-mj to mj do
   begin { put picture in pic }
   for i:=-ni to ni do
    begin { divide }
    read(averagepic,val) ;
    pic[i,j]:=val/np ;
    end ; { divide }
   readln(averagepic) ;
   end ; { put picture in pic }
  end ; { transfer picture }
 end ; { view average }
{W procedures and functions:}
{X procedures and functions:}
{Z procedures and functions:}
{

               Main Block

}
 begin {spartp}
 rewrite(calcom) ;
 reset(inpic) ;
 rewrite(projection) ;
 writeln(prompt,' START TIME ' ,clock:10) ;
 for j:=-mj to mj do
  begin { get pic }
  for i:=-ni to ni do
   begin {elps}
   valelp:=i*i/1600+j*j/625 ;
   if valelp<=1.0 then pic[i,j]:=255 else pic[i,j]:=20 ;
   end {elps}
  end ; { get pic }
 rewrite(mvpl) ;
 formvp(pic,2*mj+1,2*ni+1,8,mvpl) ;
 { define projection angles in degrees }
 for l:=1 to np do
 prangl[l]:=l*45 ;
 { }
 {    .........#########################................   }
{ }
{ compute projections }
rewrite(projection) ; picavg:=0 ;
for l:= 1 to np do
 begin  { np projections }
 angl:= prangl[l]*pi/180.0 ;
 sinl:=sin(angl) ; cosl:=cos(angl) ;
 { ray width rw may be multiplied by an integer to use wider rays ( more
 pixels per ray : modify nprmax ). Perform corrections at all occurrences of
 rw}
 rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
 for k:= -nr to nr do
  begin  { ray }
  raypixels(l,k) ;
  write(projection,pseudop) ;
  picavg:=picavg + pseudop ;
  end ;  { ray }
 writeln(projection) ;
 end ;  { np projections }
picavg:=picavg/(np*norow*nocol) ;
writeln(prompt,' AFTER PROJECTIONS ',np:5,clock:10) ;
{ }
{   ..............#########################................   }
{ }
{ Reconstruction by a linear combination of selective averaging and ART }
reset(limits) ;
read(limits,thresh1,thresh2) ;
writeln(prompt,' STREAK CONTRAST THRESHOLDS ',thresh1:10:4,thresh2:10:4) ;
for i:=-ni to ni do
for j:=-mj to mj do pic[i,j]:=picavg ;
iteration:=0 ;preerr:=9999999.0 ; error:= 0 ;
reset(projection) ;
{ starting picture for ART computed from first projection only}
l:=1 ;
angl:= prangl[l]*pi/180.0 ;
sinl:= sin(angl) ; cosl:= cos(angl) ;
rw:= amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
for k:= -nr to nr do
 begin { ray pixels }
 raypixels(l,k) ;
 read(projection,raysum) ;
 for inpr:= 1 to npr do
  begin { define pixels }
  ip:= pixel[1,inpr] ; jp:= pixel[2,inpr] ;
  pic[ip,jp]:= raysum/npr ;
  end ; { define pixels }
 end ; { ray pixels }
 repeat  { ART }
 reset(projection) ;
 iteration:=iteration+1 ; projl:= 1;
 if iteration=1 then projl:=2 else projl:=1 ;
 for l:= projl to np do
  begin  { projection  }
  nsmooth:=0 ; nart:=0 ; nsharpen:=0 ;
  angl:=prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:=cos(angl) ;
```

```
rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
for k:=-nr to nr do
  begin { rays }
  read(projection,raysum) ;
  if raysum>0.0 then
    begin { neglect zero rays }
    raypixels(l,k) ;
    for inpr:=1 to npr do
      begin { neighborhood operations}
      ip:=pixel[1,inpr] ; jp:=pixel[2,inpr] ;
      nprepix:=0 ; nnexpix:=0 ;
      prevray[inpr]:=0 ; nextray[inpr]:=0 ;
      for i:= 1 to 3 do
      for j:= 1 to 3 do
        begin { search 8-neighborhood }
        jj1:= ip+i-2; jj2:= jp+j-2 ;
        if (abs(jj1)<=ni) and (abs(jj2)<=mj) then
          begin { test distance }
          val:= (jj1*sinl + (k-1)*rw)/cosl ;
          val:= abs((jj2-val)*cosl) ;
          if val<=thr then
            begin { previous ray pixel }
            nprepix:=nprepix + 1 ;
            val:=pic[jj1,jj2] ;
            if val=-999 then val:=0.0 ;
            prevray[inpr]:= prevray[inpr] + val ;
            end ; { previous ray pixel }
          val:= (jj1*sinl + (k+1)*rw)/cosl ;
          val:= abs((jj2-val)*cosl) ;
          if val<=thr then
            begin { next ray pixel }
            nnexpix:= nnexpix + 1 ;
            val:=pic[jj1,jj2] ;
            if val=-999 then val:=0.0 ;
            nextray[inpr]:= nextray[inpr] + val ;
            end ; { next ray pixel }
          end ; { test distance }
        end ; { search 8-neighborhood }
      if nprepix=0 then prevray[inpr]:=0.0
      else prevray[inpr]:= prevray[inpr]/nprepix ;
      if nnexpix=0 then nextray[inpr]:=0.0
      else nextray[inpr]:= nextray[inpr]/nnexpix ;
      end ; { neighborhood operations }
    for inpr:=1 to npr do
      begin { compute weighted neighborhood }
      midpix:= pic[pixel[1,inpr],pixel[2,inpr]] ;
      if inpr=1 then pixpre:=pic[pixel[1,npr],pixel[2,npr]]
      else pixpre:=pic[pixel[1,inpr-1],pixel[2,inpr-1]] ;
      if inpr=npr then pixnex:=pic[pixel[1,1],pixel[2,1]]
      else pixnex:=pic[pixel[1,inpr+1],pixel[2,inpr+1]] ;
      val:=nextray[inpr] + prevray[inpr] + 2*midpix ;
      if val<>0.0 then
      across:= (abs(nextray[inpr]-midpix)+abs(prevray[inpr]-midpix))/val
      else across:=0.0 ;
      val:= nextray[inpr] + prevray[inpr] ;
      if val<>0.0 then ltrt:=abs(nextray[inpr]-prevray[inpr])/val
      else ltrt:=0.0 ;
      val:=pixpre + pixnex + 2*midpix ;
      if val<>0.0 then
      along:= (abs(midpix-pixnex) + abs(midpix-pixpre))/(2*val)
      else along:=0.0 ;
      if (nnexpix=0)or(nprepix=0) then
        begin {avoid false contrast}
        across:=0.0; along:=0.0;
        end ; {avoid false contrast}
      if (across>thresh1)and(along<thresh1) then
        begin { streak }
        if ltrt>thresh2 then
          begin { edge}
          nsharpen:=nsharpen+1 ;
          if abs(midpix-nextray[inpr])<abs(midpix-prevray[inpr]) then
          neighborhood[inpr]:=nextray[inpr]
          else neighborhood[inpr]:=prevray[inpr] ;
          end { edge }
        else { bump }
          begin { smoothing }
          nsmooth:=nsmooth+1;
          neighborhood[inpr]:= 0.5*(nextray[inpr] + prevray[inpr]) ;
          end ; { smoothing }
        end { streak }
      else { nothing }
        begin
        nart:=nart+1 ;
        neighborhood[inpr]:=midpix ;
        end ;
      end ; { compute weighted neighborhood }
     neighbor_total:=0.0 ;
     for inpr:=1 to npr do
     neighbor_total:=neighbor_total + neighborhood[inpr] ;
     for inpr:=1 to npr do
      begin { correction }
      ip:=pixel[1,inpr] ; jp:=pixel[2,inpr] ;
      if neighbor_total<>0.0 then
      pic[ip,jp]:= neighborhood[inpr]*raysum/neighbor_total
      else pic[ip,jp]:=0.0 ;
      { if constrained ART, check for negative values }
      if (artconstr) and (pic[ip,jp]<0) then pic[ip,jp]:= 0 ;
      end;{ correction }
     end ; { neglect zero rays }
    end ; { rays }
   readln(projection) ;
   filter ;
   if iteration=5 then view_average ;
   writeln(prompt, ' PROJN ', 1:2, ' SHARPEN ', nsharpen:10, ' SMOOTH ',
   nsmooth:10, ' ART ', nart:10);
   end;  { projection }
{
  euclid_error ;
  writeln(prompt,' EUCLIDEAN ERROR ',eucliderr) ;
}
  { a suitable criterion on the error may be included here to stop ART }
  until iteration=5 ;  { ART }
```

```
formvp(pic,2*mj+1,2*ni+1,8,mvp3) ;
writeln(prompt,' CPU TIME USED ',clock:10,' MILLISECONDS.') ;
end.  { spartp }
```

```
{      Z$NOBELLS$B$READP     PICPR1$T$XN$XL/end./$XN$XCOPYD     PICPRO
*$FILE,PICPRO,R$T$XCOPY        PICPR1          *$FILE,PICPR1,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/versio

The  above  XEDIT  commands  create two backup files and copy compiler
error messages to the end, as a comment, if any. }

{
      }
      {
      Title:PICPR:PICTURE PROJECTION
      Author: Atam P.Dhawan
      Address:Department of Electrical Engineering
            University of Manitoba
            Winnipeg, Manitoba, Canada R3T 2N2.

                              Program description :
      This program computes the projections of the input image at
      specified angles.
      Parallel ray geometry is used throughout. Projections( views )
      are stored in file MVP1
      in format 5.

      }
      {

                                Program Statement

      }
      program
      picprp(input/,inpic,output,prompt,lstfrm,nxtfrm,calcom,convfn,mvpl,);
      {
                              Use of External Files

      input:   terminal input
      output:  debugging output and Pascal dump on error exits
      prompt:  terminal output
      lstfrm:  last frame recorded
      nxtfrm:  next frame to be output by this program
      calcom:  output file for PASPLOT commands
      inpic: input test picture
      mvpl:  projections of the image in format 5.

      Execute as:
      /PICPRX,INPUT,INPIC,PICPRU,OUTPUT,PICPRI,PICPRO,PICPRZ,MVP1,MVP2,
      MVP3.
      }
      {

                                  Associated Files

      PICPRC   for execution
      PICPRM   for batch compilation
      PICPRS for batch job submission
      }
      {
```

```
                    Compiler Include Directives

}
{$I'PASPLOT'}
{$I'MATH'}
{
                              Labels

}
{
                             Constants

}
Const
{A constants:}
{ for additive ART set artadd to true ; multiplicative ART set artmul to true ;
for constrained ART set artconstr to true ; set others to false }
artadd=false;artmul=true; artconstr=true;
{B constants:}
{C constants:}
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
{ max (+-) y/j index of picture ( origin at center of picture ) }
mj=63 ;
{N constants:}
{ No. of rows and columns of picture and no. of bits per word in MVP output file
}
norow=127 ; nocol=127 ; nobits=8 ;
{
ni:max(+-) x/i index of picture; np:no. of projections/views to be used;
nr:max(+-) index no. of rays; nr2:total no. of rays per view(2*nr+1) ;
nprmax:max no. of pixels a ray can cross - raywidth in pixels * max of
norow,nocol .
}
ni=63;np=8; nr=127; nr2=255;nprmax=127;
nft=512 ; mft=9 ;
{O constants:}
{P constants:}
pi = 3.1415926536;
{Q constants:}
{R constants:}
{S constants:}
{T constants:}
{U constants:}
{V constants:}
```

```
version_number = 1;
{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                                 Types

     Primary types, defined in terms of constants,  are  given  first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                             Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
twod=array[integer,integer] of real ;
twod1=array[1..np,-nr..nr] of real ;
twod2=array[-ni..ni,-mj..mj] of real ;
twod3=array[-nr..nr,1..np] of real ;
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
                             Compound Types
}
{

                                Variables

}
Var
```

```
{A variables:}
averagepic: text ;
angl,avg:real ;
{B variables:}
bufferpic: text ;
{C variables:}
convfn:text ;
cosl:real ;
calcom: calcomfile; {Needed by PASPLOT}
{D variables:}
dcpower:real ;
{E variables:}
eucliderr : real ;
error: real ;
{F variables:}
{G variables:}
{H variables:}
{I variables:}
i,ip,imin,imax,inpr,iteration:integer ;
inpic: text ;
{J variables:}
j,jj1,jj2,jj3,jj4,jmin,jmax,jp:integer ;
{K variables:}
k:integer ;
{L variables:}
ll:integer ;
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
maxfac:real ;
mvp1,mvp2,mvp3:text ;
{N variables:}
nfth:integer ;
nxtfrm: text; {See program statement}
npr:integer ;
{O variables:}
{P variables:}
pjnum: array[1..np] of integer ;
preerr:real ; {error at previous iteration }
{ pseudo projection of ray }
pseudop: real ;
{ picture array -ni..ni, -mj..mj }
pic:twod2 ;
picpr:twod3 ;
{ angles of projections : to be entered in main program }
prangl: array[1..np] of real ;
{ first dimension refers to view no. and second to ray no. }
projection:twod1 ;
pixel:array[1..2,1..nprmax] of integer ;
prompt: text; {Call to program should be
{}               /PICPRX,,PICPRU,OUTPUT, etc., to
{}              place any bugs into file PICPRU and have the prompts
{}              come out on OUTPUT. All prompts are to file PROMPT
{}              which thereby becomes OUTPUT.}
{Q variables:}
{R variables:}
```

```
{ ray width : modify in main program at all locations if necessary }
rw:real ;
{S variables:}
suppd:integer ;
sum1,sum2:real ;
spot:real ;
sin1:real ;
scale: real; {Scale factor for plotting}
{T variables:}
{ threshold for ray boundary = 0.5 * ray width }
thr:real ;
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
version_date: alfa; {Date compiled}
version_time: alfa; {Time compiled}
val:real ;
{W variables:}
{X variables:}
xreal,ximag,xabs:array[1..nft]of real ;
x:array[1..nr2] of real ;
xinfo:scaleinfo ;
{Y variables:}
y:array[1..nr2] of real ;
yinfo:scaleinfo ;
{Z variables:}
{

                              Forward Declarations

}
{Forward declarations of procedures and functions:}
{A forward declarations:}
function Amax1(a,b: real): real; forward;
function Amin1(a,b: real): real; forward;
{B forward declarations:}
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
function Factorial(n: integer): real; forward;
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
function Max0(a,b: integer): integer; forward;
function Min0(a,b: integer): integer; forward;
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
```

```
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
{

                       Procedures and Functions

}
{Procedures and functions:}
{A procedures and functions:}
{

                                Absmal

}
function absmal(one,two:integer):integer ;
{ returns no. with smaller absolute value
and sign of the first no. }
  begin { absmal }
  if one < 0 then
    begin
    if abs(one) < two then absmal:=one else absmal:=-two
    end
  else
    begin
    if one < two then absmal:=one else absmal:=two
    end
  end ; { absmal }
{

                                Amax1

}
function Amax1{(a,b: real): real};
  begin {Amax1}
  if a > b then Amax1 := a else Amax1 := b;
  end; {Amax1}
{

                                Amin1

}
function Amin1{(a,b: real): real};
  begin {Amin1}
  if a < b then Amin1 := a else Amin1 := b;
  end; {Amin1}
{B procedures and functions:}
{

                               boas_kac

}
```

```
procedure boas_kac ;
{ Boas & Kac clipping procedure }
  begin { boas_kac }
  dcpower:=sqrt(xreal[1]*xreal[1] + ximag[1]*ximag[1]) ;
  nfth:= nft div 2 ;
  nfth:= nfth + 1;
  suppd:=0 ;
  for k:=2 to nfth do
    begin { peak suppression }
    val:= sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
    maxfac:= pi/(2 + nfth div k) ;
    maxfac:= cos(maxfac) ;
    if val>(maxfac*dcpower) then
      begin { clipping }
      xreal[k]:= xreal[k]*maxfac*dcpower/val ;
      ximag[k]:= ximag[k]*maxfac*dcpower/val ;
      suppd:=suppd + 1;
      end ; { clipping }
    end ; { peak suppression }
  end ; { boas_kac }
{C procedures and functions:}
{D procedures and functions:}
{

                                 del

}
function del(i,j,k:integer;sinl,cosl:real):boolean ;
{ del : true if pixel(i,j) belongs to ray (l,k) }
var dist,jy:real ;
  begin  { del  }
  jy:=i*sinl + k*rw  ; jy:= jy/cosl ;
  dist:= abs(j-jy)*abs(cosl);
  if dist<=thr then del:=true else del:=false
  end ;  { del }
{E procedures and functions:}
{

                            euclid_error ;

}
procedure euclid_error ;
{ computes the Euclidean distance between the original picture
and its reconstruction }
  begin { euclid_error }
  eucliderr:=0.0 ;
  reset(inpic) ;
  for j:=-mj to mj do
    begin { row }
    for i:=-ni to ni do
      begin { pixels }
      read(inpic,val) ;
      val:=val-pic[i,j];
      eucliderr:=eucliderr + val*val ;
      end ; { pixels }
    readln(inpic) ;
    end ; { row }
  euclidierr:=euclidierr/(norow*nocol) ;
  end ; { euclid_error }
{F procedures and functions:}
{

                                   ftbtr

}
function ftbtr(jbtr,mft:integer):integer;
var ibtr,jbtr1,jbtr2,kbtr:integer ;
  begin { ftbtr }
  jbtr1:=jbtr ; kbtr:=0 ;
  for ibtr:= 1 to mft do
    begin
    jbtr2:=jbtr1 div 2 ;
    kbtr:=kbtr*2 + (jbtr1-2*jbtr2) ;
    jbtr1:=jbtr2 ;
    end ;
  ftbtr:=kbtr ;
  end ; { ftbtr }
{

                                   FFT

}
procedure FFT;
{ Fast Fourier Transform program from E.O.Brigham
Translated from Fortran
Arrays xreal and ximag must contain real and imaginary parts
of input signal. The same arrays will contain on return the
real and imaginary parts of the Fourier Transform- the input
data are lost. NFT and MFT must contain no. of FFT samples
required, and NFT=2**MFT. }
var kft,lft,ift,kft1,kft1n2,nft2,mft1,jft,tupwr:integer ;
var pft,arg,cft,sft,treal,timag:real ;
  begin { fft }
  nft2:=nft div 2 ; mft1:=mft-1 ; kft:=0 ;
  for lft:=1 to mft do
    begin { loop 100 }
    while kft<nft do
      begin { loop 102 }
      for ift:=1 to nft2 do
        begin { loop 127 }
        tupwr:=1 ;
        for jft:=1 to mft1 do
        tupwr:=tupwr*2 ;
        pft:= ftbtr(kft div tupwr,mft) ;
        arg:=6.283185*pft/nft ;
        cft:=cos(arg) ; sft:=sin(arg) ;
        kft1:=kft+1 ; kft1n2:=kft1+nft2 ;
        treal:= xreal[kft1n2]*cft + ximag[kft1n2]*sft ;
        timag:= ximag[kft1n2]*cft - xreal[kft1n2]*sft ;
        xreal[kft1n2]:=xreal[kft1]-treal ;
        ximag[kft1n2]:=ximag[kft1]-timag ;
        xreal[kft1]:= xreal[kft1] + treal ;
    ximag[kft1]:= ximag[kft1] + timag ;
    kft:=kft+1 ;
    end ; { loop 127 }
   kft:= kft + nft2 ;
   end ; { loop 102 }
  kft:= 0 ; mft1:= mft1-1 ;
  nft2:= nft2 div 2 ;
  end ; { loop 100 }
 for kft:= 1 to nft do
  begin { loop 103 }
  ift:= ftbtr(kft-1,mft) + 1 ;
  if ift>kft then
   begin { inner loop of 103 }
   treal:= xreal[kft] ;
   timag:= ximag[kft] ;
   xreal[kft]:=xreal[ift] ; ximag[kft]:=ximag[ift] ;
   xreal[ift]:=treal ; ximag[ift]:=timag ;
   end ; { inner loop of 103 }
  end ; { loop 103 }
 end ; { fft }
{

                Factorial

}
function Factorial{(n: integer): real};
 begin {Factorial}
 if n = 0 then Factorial := 1
 else Factorial := n * Factorial(n-1);
 end; {Factorial}
{

                 formvp

}
procedure formvp( var fig:dynamic twod;nrow,ncol,bits:integer; var mvpfil:text);
{ writes picture in MVP format 5 in file mvpfil }
var ij,jk,newval,twelv:integer ; maxel,minel,lim:real ;okay:boolean ;
 begin { for mvp }
 okay:=true ;
 if (nrow<1) or (nrow>512) or (ncol<1) or (ncol>512) then
  begin { error 2 }
  writeln(prompt,'Maximum picture size permitted is 512 X 512') ;
  writeln(output,'Maximum picture size permitted is 512 X 512') ;
  okay:=false ;
  end ; { error 2 }
 if (bits<1) or (bits>12) then
  begin { error 3 }
  writeln(prompt,'MVP resolution limited to 1..12 bits') ;
  writeln(output,'MVP resolution limited to 1..12 bits') ;
  okay:=false ;
  end ; {error 3 }
 if okay then
  begin { execute formvp }
  maxel:=0 ;
   case bits of  { resolution }
   1: lim:=1 ;
   2: lim:=3 ;
   3: lim:=7 ;
   4: lim:=15 ;
   5: lim:=31 ;
   6: lim:=63 ;
   7: lim:=127 ;
   8: lim:=255 ;
   9: lim:=511 ;
   10:lim:=1023 ;
   11:lim:=2047 ;
   12:lim:=4095 ;
   end ;    { resolution }
  ij:=low(fig,1) ; jk:=low(fig,2) ; minel:=fig[ij,jk] ;
  for ij:=low(fig,1) to high(fig,1)do
  for jk:=low(fig,2) to high(fig,2) do
   begin { minmax }
   if maxel<fig[ij,jk] then maxel:=fig[ij,jk] ;
   if minel>fig[ij,jk] then minel:=fig[ij,jk] ;
   end ; { minmax }
  writeln(mvpfil,'PICTURE FILE FOR DISPLAY USING MVP.') ;
  writeln(mvpfil,'ENDOFHEADR',ncol:5,nrow:5,bits:5,1:5) ;
  for jk:=low(fig,2) to high(fig,2) do
   begin { row }
   twelv:=0 ;
   for ij:=low(fig,1) to high(fig,1) do
    begin { col }
    {linear transformation from 0-maxel to 0-lim}
    newval:=round(lim*fig[ij,jk]/maxel) ;
    if newval<0 then newval:=0 ;
    write(mvpfil,newval:5) ;
    twelv:=twelv+1 ;
    if twelv=12 then
     begin { end line }
     twelv:=0 ;
     writeln(mvpfil) ;
     end ; { end line }
    end ; { col }
   writeln(mvpfil) ;
   end ; { row }
  end ; { execute formvp }
 end ; { for mvp }
{G procedures and functions:}
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{

                jminmax

}
procedure jminmax(n1,n2,n3,n4 :integer ; var min,max:integer ) ;
{ returns min and max of four integers given }
 begin { jminmax }
 min:=n1 ; max:=n1 ;
 if n2<min then min:=n2 ;
```

```
  if n3<min then min:=n3 ;
  if n4<min then min:=n4 ;
  if n2>max then max:=n2 ;
  if n3>max then max:=n3 ;
  if n4>max then max:=n4 ;
  end ; { jminmax }
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                                 MaxO

}
function MaxO{(a,b: integer): integer};
  begin {MaxO}
  if a > b then MaxO := a else MaxO := b;
  end; {MaxO}
{

                                 MinO

}
function MinO{(a,b: integer): integer};
  begin {MinO}
  if a < b then MinO := a else MinO := b;
  end; {MinO}
{N procedures and functions:}
{O procedures and functions:}
{

                              other_half

}
procedure other_half ;
{duplicates second half of Fourier Transform array}
  begin { other_half }
  nfth:= nft div 2;
  nfth:= nfth + 1 ;
  for k:= nft downto nfth+1 do
    begin { other half of spectrum }
    xreal[k]:= xreal[nft-k+2] ;
    ximag[k]:=ximag[nft-k+2] ;
    end ; { other half of spectrum }
  end ; { other_half }
{P procedures and functions:}
{

                                 plot

}
procedure plot ;
{ prepares 8 plots in one page : works for np=8 only  }
  begin  { plotter instructions }
  plotopen(calcom,longx,nearside,11.0,7.0,'RANGARAJ') ;
  setscale(1.0) ;
```

```
 for l:=1 to np do
   begin { np plots }
   for i:= 1 to nr2 do
     begin
     x[i]:= i ; y[i]:= projection[l,i-nr-1] ;
     end ;
     case l of
     1:move(1.5,0) ;
     2:move(1.5,1.5) ;
     3:move(1.5,3.0) ;
     4:move(1.5,4.5) ;
     5:move(5.5,0) ;
     6:move(5.5,1.5) ;
     7:move(5.5,3.0) ;
     8:move(5.5,4.5) ;
     end ; { case }
   drawinteger(l,3,0.2,0.0) ;
   drawstring(' PROJN ANGLE',-1,0.2,0.0);
   drawreal(prangl[l],7,2,0.2,0.0) ;
     case l of
     1:move(1.5,0.3) ;
     2:move(1.5,1.8) ;
     3:move(1.5,3.3) ;
     4:move(1.5,4.8) ;
     5:move(5.5,0.3) ;
     6:move(5.5,1.8) ;
     7:move(5.5,3.3) ;
     8:move(5.5,4.8) ;
     end ; { case }
   scaledata(x,nr2,4,xinfo) ;
   scaledata(y,nr2,1.0,yinfo) ;
   drawline(x,y,nr2,xinfo,yinfo,lineonly,1) ;
   end ; { np plots }
 writeln(prompt,'THE PLOT IS ',plotclose:5:2,' INCHES LONG.')
 end ; { plotter instructions }
{Q procedures and functions:}
{R procedures and functions:}
{

                              raypixels

}
procedure raypixels(l,k:integer ) ;
{
Parallel ray geometry. Returns no. of pixels covered by ray(l,k) in npr,
their coords in pixel[1,i],pixel[2,i] and the pseudo projection in
pseudo. Raywidth rw, threshold thr, sin and cos of view angle sinl,cosl
to be defined in main program before calling raypixels.
Calls absmal,del,jminmax,min0,max0.
}
 begin { raypixels }
 npr:=0 ; pseudop:=0 ;
 { get limits of search in y/j direction }
 jj1:=round((-ni*sinl+(k-1)*rw)/cosl) ;
 jj2:=round((-ni*sinl+(k+1)*rw)/cosl) ;
 jj3:=round((ni*sinl+(k-1)*rw)/cosl) ;
 jj4:=round((ni*sinl+(k+1)*rw)/cosl) ;
 jj1:=absmal(jj1,mj) ; jj2:=absmal(jj2,mj) ;
 jj3:=absmal(jj3,mj) ; jj4:=absmal(jj4,mj) ;
 jminmax(jj1,jj2,jj3,jj4,jmin,jmax) ;
 for j:=jmin to jmax do
  begin { row }
  { get limits of search in x/i direction }
  jj1:=round((j*cosl-(k-1)*rw)/sinl) ; jj1:=absmal(jj1,ni-1) ;
  jj2:=round((j*cosl-(k+1)*rw)/sinl) ; jj2:=absmal(jj2,ni-1) ;
  imin:=min0(jj1,jj2)-1 ; imax:=max0(jj1,jj2)+1;
  for i:=imin to imax do
   begin  { column }
   if del(i,j,k,sinl,cosl) then
    begin  { store ray pixels }
    npr:=npr+1 ; pseudop:=pseudop+pic[i,j] ;
    pixel[1,npr]:=i ; pixel[2,npr]:=j ;
    end;  { store ray pixels }
   end ; { column }
  end ; { row }
 end ; { raypixels }
{S procedures and functions:}
{

                 spectra

}
procedure spectra ;
{computes average power spectral envelope of the
original eight projections in array xabs }
 begin { spectra }
 for ll:=1 to 3 do
  begin
  if ll=1 then l:=2 ;
  if ll=2 then l:=4 ;
  if ll=3 then l:=8 ;
  end ;
  begin { 3 projections }
  for k:=1 to nr2 do
  xreal[k]:= projection[l,k-nr-1] ;
  for k:= nr2+1 to nft do xreal[k]:=0 ;
  for k:=1 to nft do ximag[k]:=0 ;
  fft;
  for k:=1 to nft do
  xabs[k]:= xabs[k] + sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
  end ; { 3 projections }
 for k:=1 to nft do
 xabs[k]:= xabs[k]/3 ;
 end ; { spectra }
{

                 speclip

}
procedure speclip ;
{clips given spectra to spectral average envelope in array xabs}
 begin { speclip }
```

```
  dcpower:=sqrt(sqr(xreal[1])+sqr(ximag[1])) ;
  for k:=1 to nft do
    begin { check }
    val:= sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
    val:=val*xabs[1]/(dcpower*xabs[k]) ;
    if val>1 then
      begin { clip }
      xreal[k]:= xreal[k]/val ;
      ximag[k]:= ximag[k]/val ;
      end ; { clip }
    end ; { check }
  end ; { speclip }
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{

                             view_average

}
procedure view_average ;
{ produces a picture which is the average of reconstructions
at end of each view }
  begin { view average }
  if l=1 then
    begin { initiate averagepic }
    rewrite(averagepic) ;
    for j:=-mj to mj do
      begin { row }
      for i:=-ni to ni do
      write(averagepic,pic[i,j]) ;
      writeln(averagepic) ;
      end ; { row }
    end  { initiate averagepic }
  else
    begin { add next view picture }
    reset(averagepic) ;
    rewrite(bufferpic) ;
    for j:=-mj to mj do
      begin { row }
      for i:= -ni to ni do
        begin { pixel }
        read(averagepic,val) ;
        val:=val + pic[i,j] ;
        write(bufferpic,val) ;
        end ; { pixel }
      readln(averagepic) ;
      writeln(bufferpic) ;
      end ; { row }
    rewrite(averagepic) ;
    reset(bufferpic) ;
    for j:= -mj to mj do
      begin { swap files }
      for i:=-ni to ni do
        begin { interchange }
        read(bufferpic,val) ;
        write(averagepic,val) ;
        end ; { interchange }
      readln(bufferpic) ;
      writeln(averagepic) ;
      end ; { swap files }
    end ; { add next view picture }
  if l=np then
    begin { transfer picture }
    reset(averagepic) ;
    for j:=-mj to mj do
      begin { put picture in pic }
      for i:=-ni to ni do
        begin { divide }
        read(averagepic,val) ;
        pic[i,j]:=val/np ;
        end ; { divide }
      readln(averagepic) ;
      end ; { put picture in pic }
    end ; { transfer picture }
  end ; { view average }
{W procedures and functions:}
{

                                 weight

}
procedure weight ;
{weight the spectrum }
  begin { weight }
  for k:=2 to nfth do
    begin { weighting }
    val:= 1 - sqrt(k/nfth) ;
    xreal[k]:= val*xreal[k] ;
    ximag[k]:= val*ximag[k] ;
    end ; { weighting }
  end ; { weight }
{X procedures and functions:}
{

                                 xerror

}
procedure xerror ;
{computes error of reconstruction comparing projection data}
  begin { error }
  error:=0.0 ;
  for l:= 1 to np do
    begin  { np projections }
    angl:= prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:= -nr to nr do
      begin  { ray }
      raypixels(l,k) ;
      error:=error + abs(projection[l,k]-pseudop) ;
      end ;  { ray }
  end ;  { np projections }
 error:=error/(norow*nocol*np) ;
 end ; { error }
{Y procedures and functions:}
{Z procedures and functions:}
{

                   Main Block

}
 begin {picprp}
 rewrite(calcom) ;
 reset(inpic) ;
 writeln(prompt,' START TIME ' ,clock:10) ;
 for i:=-ni to ni do
 for j:=-mj to mj do
 pic[i,j]:=0 ;
 for j:=-mj to mj do
  begin { get pic }
  for i:=-ni to ni do
  read(inpic,pic[i,j]) ;
  readln(inpic) ;
  end ; { get pic }
 { define projection angles in degrees }
 for l:=1 to np do prangl[l]:=55+(l-1)*10 ;
 { }
 {    .........#########################................   }
 { }
 { compute projections }
 for l:= 1 to np do
 for k:=-nr to nr do projection[l,k]:= 0 ;
 for k:=-nr to nr do
 for l:=1 to np do picpr[k,l]:=0 ;
 for l:= 1 to np do
  begin  { np projections }
  angl:= prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:=cos(angl) ;
  { ray width rw may be multiplied by an integer to use wider rays ( more
  pixels per ray : modify nprmax ). Perform corrections at all occurrences of
  rw}
  rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
  for k:= -nr to nr do
   begin  { ray }
   raypixels(l,k) ;
   projection[l,k]:= pseudop ;
   end ;  { ray }
  end ;  { np projections }
 for k:=-nr to nr do
 for l:=1 to np do picpr[k,l]:=projection[l,k] ;
 formvp(picpr,np,2*nr+1,8,mvpl)  ;
 writeln(prompt,' CPU TIME USED ',clock:10,' MILLISECONDS.') ;
 end.  { picprp }
```

```
{       Z$NOBELLS$B$READP       CTR1$T$XN$XL/end./$XN$XCOPYD      CTRO
*$FILE,CTRO,R$T$XCOPY           CTR1            *$FILE,CTR1,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/versio

The  above  XEDIT  commands  create two backup files and copy compiler
error messages to the end, as a comment, if any. }

{
      }
      {
      Title:CTRP: Reconstructive Tomography from radiographs
      Author:M.R.Rangaraj
      Modified by: Atam P. Dhawan
      Address:Department of Pathology, University of Manitoba,
            Medical College, 770 Bannatyne avenue,
            Winnipeg, Manitoba, Canada R3E OW3.

                                Program description :
      This program performs computed tomography from radiographs.
      Scan lines are read from the given set of radiographs at
      a specified sectional level and projection data are computed
      from them. Reconstructions of the section are computed by
      the summation (back projection) and ART techniques. The parameters
      of the scan lines, reconstruction matrix, and the ART algorithm
      are to be specified in the constants and variables lists.
      Parallel ray geometry is used throughout.

      }
      {

                                  Program Statement

      }
      program ctrp(input/,output,prompt,lstfrm,nxtfrm,calcom,mvp1,mvp2,
      p1,p2,p3) ;
      {
                                Use of External Files

      input:   terminal input
      output:  debugging output and Pascal dump on error exits
      prompt:  terminal output
      lstfrm:  last frame recorded
      nxtfrm:  next frame to be output by this program
      calcom:  output file for PASPLOT commands
      mvp1:  reconstruction by summation
      mvp2:  ART reconstruction
      p1..p11: Input radiographic views
      params: Input parameters file (tomo level, scanline width,
              x and y coords of calibn. pt. )

      Execute as:
      /CTRX,INPUT,INPIC,CTRU,OUTPUT,CTRI,CTRO,CTRZ,PARAMS,MVP1,MVP2,
      P1..P11.
      }
      {
```

```
                         Associated Files

CTRC   for execution
CTRM   for batch compilation
CTRS for batch job submission
}
{

                     Compiler Include Directives

}
{$I'PASPLOT'}
{

                              Labels

}
{

                             Constants

}
Const
{A constants:}
{ for additive ART set artadd to true ; multiplicative ART set artmul to true ;
for constrained ART set artconstr to true ; set others to false }
{ for labelling of zero ray pixels set artzero=true ; doubly constrained
ART set artupconstr=true and enter upper pixel limit in artuplim }
artadd=false; artmul=true ; artconstr=true ;
artzero=true ; artupconstr=true ; artuplim=63.0 ;
{B constants:}
{C constants:}
{ define threshold at which projection data ends can be chopped
and also no. of points to be left out before the point when
the chopping threshold ( suggested value : 5% total excursion )
is reached }
chopthr=10.0 ; choptail=10 ;
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
{ max (+-) y/j index of picture ( origin at center of picture ) }
mj=50 ;
{N constants:}
{ No. of rows and columns of picture and no. of bits per word in MVP output file
}
norow=101 ; nocol=101 ; nobits=8 ;
{
ni:max(+-) x/i index of picture; np:no. of projections/views to be used;
nr:max(+-) index no. of rays; nr2:total no. of rays per view(2*nr+1) ;
```

```
nprmax:max no. of pixels a ray can cross - raywidth in pixels * max of
norow,nocol .
}
ni=50;np=3; nr=110; nr2=221;nprmax=101;
{O constants:}
{P constants:}
pi = 3.1415926536;
scanpix=121 ;
numpix=121 ;
teex=15 ;
{Q constants:}
{R constants:}
{S constants:}
{T constants:}
{U constants:}
{V constants:}
version_number = 1;
{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                              Types

     Primary types, defined in terms of constants,  are  given  first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                            Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
oned= array[1..512] of real ;
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
twod=array[integer,integer] of real ;
twod1=array[1..np,-nr..nr] of real ;
```

```
twod2=array[-ni..ni,-mj..mj] of real ;
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
                              Compound Types
}
{

                               Variables

}
Var
{A variables:}
averagepic:text ;
angl,avg:real ;
{B variables:}
bufferpic:text ;
{C variables:}
cosl:real ;
calcom: calcomfile; {Needed by PASPLOT}
{D variables:}
{ scan line read from radiograph }
dens : oned;
{E variables:}
edgelt,edgert:integer ;
error: real ;
{F variables:}
{G variables:}
{H variables:}
{I variables:}
i,ip,imin,imax,inpr,iteration:integer ;
inpic: text ;
{J variables:}
j,jj1,jj2,jj3,jj4,jmin,jmax,jp:integer ;
{K variables:}
k:integer ;
{L variables:}
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
mvp1,mvp2:text ;
{N variables:}
normspot: real ; { for normalisation of views }
nextint: integer ;
nxtfrm: text; {See program statement}
npr:integer ;
{O variables:}
omit:boolean; {true if ray to be omitted from ART}
{P variables:}
params:text ; {input parameters file }
projn: array[-nr..nr] of real ;
preerr:real ; { error of previous reconstruction }
```

```
{Input files containing radiographic data of the different views}
p1,p2,p3 :text ;
{ pseudo projection of ray }
pseudop: real ;
{ picture array -ni..ni, -mj..mj }
pic:twod2 ;
{ angles of projections : to be entered in main program }
prangl: array[1..np] of real ;
{ first dimension refers to view no. and second to ray no. }
projection:twod1 ;
pixel:array[1..2,1..nprmax] of integer ;
prompt: text; {Call to program should be
{}               /CTRX,,CTRU,OUTPUT, etc., to
{}              place any bugs into file CTRU and have the prompts
{}              come out on OUTPUT. All prompts are to file PROMPT
{}              which thereby becomes OUTPUT.}
{Q variables:}
{R variables:}
{ ray width : modify in main program at all locations if necessary }
rw:real ;
{S variables:}
spotx,spoty:integer ; {x,y coords of calibn pt}
{
scanpix:integer; {width of scan line}
}
section:integer; {tomo level desired}
shiftp: array[1..np]of real ;
spot: real ; { for normalisation of views }
step: real ;
sinl:real ;
scale: real; {Scale factor for plotting}
{T variables:}
{
teex,teey,toff:integer; {coords of T point on radiograph and offset desired}
}
teey,toff:integer;
trunk: integer ;
{ threshold for ray boundary = 0.5 * ray width }
thr:real ;
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
version_date: alfa; {Date compiled}
version_time: alfa; {Time compiled}
val:real ;
{W variables:}
{X variables:}
x:array[1..nr2] of real ;
xinfo:scaleinfo ;
{Y variables:}
y:array[1..nr2] of real ;
yinfo:scaleinfo ;
{Z variables:}
{

                        Forward Declarations
}
{Forward declarations of procedures and functions:}
{A forward declarations:}
function Amaxl(a,b: real): real; forward;
function Aminl(a,b: real): real; forward;
{B forward declarations:}
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
function Factorial(n: integer): real; forward;
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
function Max0(a,b: integer): integer; forward;
function Min0(a,b: integer): integer; forward;
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
{

                        Procedures and Functions

}
{Procedures and functions:}
{A procedures and functions:}
{

                                 Absmal

}
function absmal(one,two:integer):integer ;
{ returns no. with smaller absolute value
and sign of the first no. }
  begin { absmal }
  if one < 0 then
    begin
    if abs(one) < two then absmal:=one else absmal:=-two
    end
  else
    begin
    if one < two then absmal:=one else absmal:=two
    end
  end ; { absmal }
{

                              Amax1

}
function Amax1{(a,b: real): real};
  begin {Amax1}
  if a > b then Amax1 := a else Amax1 := b;
  end; {Amax1}
{

                              Amin1

}
function Amin1{(a,b: real): real};
  begin {Amin1}
  if a < b then Amin1 := a else Amin1 := b;
  end; {Amin1}
{B procedures and functions:}
{C procedures and functions:}
{

                             chop_sides

}
procedure chop_sides ;
{ chops sides of projection data by detecting starting
points of the object from either side- chopthr and choptail
must be specified in constants list. Also subtracts film fig. }
  begin { chop sides }
  for l:=1 to np do
    begin { projections }
    val:=0 ;
    for k:=-nr to -nr+9 do val:=val+projection[l,k] ;
    k:= -nr + 9 ;
      repeat
      k:=k+1 ; val:= val + projection[l,k] ;
      until (k=0) or (abs(projection[l,k] - val/(k+nr+1))>chopthr) ;
    avg:= val/(k+nr+1) ; i:= k-choptail ;
    writeln(prompt,' PROJN ',l:3,' LEFT CHOP ',i:5) ;
    val:=0 ;
    for k:=-nr to i do projection[l,k]:=0 ;
    for k:=nr downto nr-9 do val:=val + projection[l,k] ;
    k:=nr-9 ;
      repeat
      k:=k-1 ; val:= val + projection[l,k] ;
      until (k=0) or (abs(projection[l,k] - val/(nr-k+1))>chopthr) ;
    avg:= 0.5*(avg + val/(nr-k+1)) ; i:= k + choptail ;
    writeln(prompt,' PROJN ',l:3,' RIGHT CHOP ',i:5,' FOG ',avg:6:2) ;
    for k:=nr downto i do projection[l,k]:=0 ;
    for k:=-nr to nr do projection[l,k]:=projection[l,k]-avg ;
    for k:=-nr to nr do
    if projection[l,k]<0 then projection[l,k]:=0 ;
  end ; { projections }
 end ; { chop sides }
{
                     centroid_plot

}
procedure centroid_plot ;
{ this procedure computes the centroids of the projections
and draws the rays that represent the centroids. All these rays
should intersect at one single point if the registration is perfect }
var moved :boolean ;
 begin { plot centroid }
 plotopen(calcom,longx,nearside,100.0,100.0,'RAJ') ;
 setscale(0.05) ; setorigin(50.0,50.0) ;
 move(-50.0,50.0); draw(50.0,50.0);
 draw(50.0,-50.0); draw(-50.0,-50.0);
 draw(-50.0,50.0); move(0.0,0.0);drawspecial(1,2.0,0.0) ;
 for l:=1 to np do
  begin { projection }
  angl:=prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:=cos(angl) ;
  rw:=amaxl(abs(sinl),abs(cosl)) ;
  avg:=0.0 ; val:=0.0 ;
  for k:=-nr to nr do
   begin { compute centroid }
   avg:=avg+projection[l,k]*k ;
   val:=val+projection[l,k] ;
   end ; {compute centroid }
  if not (val=0) then
  val:=avg/val ;
  writeln(prompt,' CENTROID ', val:6:2) ;
  moved:=false ;
  spot:= (50.0*cosl-rw*val)/sinl ;
  writeln(prompt,' COORDS ',spot:10:4,' 50.0 ') ;
  if abs(spot)<50.0 then
   begin
   moved:=true ;
   move(spot,50.0) ;
   end ;
  spot:=(-50.0*cosl-rw*val)/sinl ;
  writeln(prompt,' COORDS ', spot:10:4, ' -50.0 ') ;
  if abs(spot)<50.0 then
   begin
   if moved then draw(spot,-50.0)
   else
    begin
    moved:=true ;
    move(spot,-50.0) ;
    end ;
   end ;
  spot:=(50.0*sinl + rw*val)/cosl ;
  writeln(prompt,' COORDS  50.0 ',spot:10:4) ;
  if abs(spot)<50.0 then
   begin
   if moved then draw(50.0,spot)
   else
```

```
        begin
        moved:=true ;
        move(50.0,spot) ;
        end ;
      end ;
    spot:=(-50.0*sinl + rw*val)/cosl ;
    writeln(prompt,' COORDS -50.0 ',spot:10:4) ;
    if abs(spot)<50.0 then draw(-50.0,spot) ;
    end ; { projection }
  writeln(prompt,' CENTROID  PLOT DONE ',plotclose:6:2) ;
  end ; { plot centroid }
{

                         centroid-correction

}
procedure centroid_correction ;
{ This procedure computes the intersections of all the centroid rays by
solving the equations of the rays two at a time. The mid point of all these
intersections is taken to be the new centroid and the shift required for
all projections to bring their centroids to the ray passing through
this mid point is computed and projection data shifted accordingly }
var a1,a2,b1,b2,c1,c2,midx,midy,sum,wtdsum:real ; cntroid:array[1..np]of real ;
  begin { centroid correction }
  for l:= 1 to np do
    begin { projection }
    wtdsum:=0.0 ; sum:=0.0 ;
    for k:=-nr to nr do
      begin { rays }
      sum:=sum+projection[l,k] ;
      wtdsum:=wtdsum+projection[l,k]*k ;
      end ; { rays }
    if not (sum=0.0) then
    cntroid[l]:=wtdsum/sum
    else cntroid[l]:= 0.0 ;
    end ; { projection }
  midx:=0.0 ; midy:= 0.0 ;
  for l:= 1 to np do
    begin { projection }
    angl:= prangl[l]*pi/180.0 ;
    sinl:= sin(angl) ; cosl:=cos(angl) ;
    a1:=sinl ; b1:=-cosl ;
    c1:=amax1(abs(sinl),abs(cosl))*cntroid[l] ;
    for i:= 1 to np do
      begin { intersection with other centroid rays }
      if not (i=l) then
        begin { take up this projection }
        angl:= prangl[i]*pi/180.0 ;
        sinl:=sin(angl) ; cosl:= cos(angl) ;
        a2:=sinl ; b2:=-cosl ;
        c2:= amax1(abs(sinl),abs(cosl))*cntroid[i] ;
        { solve for the intersection of the two centroid rays }
        midx:=midx + (b1*c2-b2*c1)/(b2*a1-b1*a2) ;
        midy:=midy + (a1*c2-a2*c1)/(b1*a2-b2*a1) ;
        end ; { take up this projection }
      end ; { intersection with other projections }
  end ; { projection }
 midx:= midx/(np*(np-1)) ; midy:= midy/(np*(np-1)) ;
 for l:= 1 to np do
  begin { compute shift required }
  angl:=prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:= cos(angl) ;
  rw:= amaxl(abs(sinl),abs(cosl)) ;
  { shift required = centroid - index of the ray that passes through
  midx,midy which is the average of all intersections
  of the np centroid rays }
  shiftp[l]:= cntroid[l] - (midy*cosl - midx*sinl)/rw ;
  end ; { compute shift required }
 { shift projection data appropriately }
 for l:=1 to np do
  begin { projections }
  k:=-nr ;
  while k<=nr do
   begin { shift }
   i:=k+round(shiftp[l]) ;
   if i<-nr then i:=-nr ;
   if i>nr then i:=nr ;
   projn[k]:=projection[l,i] ;
   k:=k+1 ;
   end ; { shift }
  for k:=-nr to nr do
  projection[l,k]:=projn[k] ;
  end ; { projections }
 end ; { centroid correction }
{D procedures and functions:}
{

                del

}
function del(i,j,k:integer;sinl,cosl:real):boolean ;
{ del : true if pixel(i,j) belongs to ray (l,k) }
var dist,jy:real ;
 begin  { del  }
 jy:=i*sinl + k*rw  ; jy:= jy/cosl ;
 dist:= abs(j-jy)*abs(cosl);
 if dist<=thr then del:=true else del:=false
 end ;  { del }
{E procedures and functions:}
{F procedures and functions:}
{

               Factorial

}
function Factorial{(n: integer): real};
 begin {Factorial}
 if n = 0 then Factorial := 1
 else Factorial := n * Factorial(n-1);
 end; {Factorial}
{
}
procedure formvp( var fig:dynamic twod;nrow,ncol,bits:integer; var mvpfil:text);
{ writes picture in MVP format 5 in file mvpfil }
{ pixel values not modified in any manner : all transformations
for enhancement to be done outside, but provision is included
here for such modifications }
var ij,jk,newval,twelv:integer ; maxel,minel,lim:real ;okay:boolean ;
  begin { for mvp }
  okay:=true ;
  if (nrow<1) or (nrow>512) or (ncol<1) or (ncol>512) then
    begin { error 2 }
    writeln(prompt,'Maximum picture size permitted is 512 X 512') ;
    writeln(output,'Maximum picture size permitted is 512 X 512') ;
    okay:=false ;
    end ; { error 2 }
  if (bits<1) or (bits>12) then
    begin { error 3 }
    writeln(prompt,'MVP resolution limited to 1..12 bits') ;
    writeln(output,'MVP resolution limited to 1..12 bits') ;
    okay:=false ;
    end ; {error 3 }
  maxel:=0 ;
    case bits of  { resolution }
    1: lim:=1 ;
    2: lim:=3 ;
    3: lim:=7 ;
    4: lim:=15 ;
    5: lim:=31 ;
    6: lim:=63 ;
    7: lim:=127 ;
    8: lim:=255 ;
    9: lim:=511 ;
    10:lim:=1023 ;
    11:lim:=2047 ;
    12:lim:=4095 ;
    end ;    { resolution }
   ij:=low(fig,1) ; jk:=low(fig,2) ; minel:=fig[ij,jk] ;
   for ij:=low(fig,1) to high(fig,1)do
   for jk:=low(fig,2) to high(fig,2) do
     begin { minmax }
     if maxel<fig[ij,jk] then maxel:=fig[ij,jk] ;
     if minel>fig[ij,jk] then minel:=fig[ij,jk] ;
     end ; { minmax }
   {exclude this section if picture normalized later }
 {
   if maxel>lim then
     begin { error 4 }
     writeln(prompt,' MVP BIT LIMIT EXCEEDED.') ;
     writeln(output,' MVP BIT LIMIT EXCEEDED.') ;
     okay:=false ;
     end ; { error 4 }
 }
   if okay then
     begin { execute formvp }
  writeln(mvpfil,'PICTURE FILE FOR DISPLAY USING MVP.') ;
  writeln(mvpfil,'ENDOFHEADR',ncol:5,nrow:5,bits:5,1:5) ;
  for jk:=low(fig,2) to high(fig,2) do
   begin { row }
   twelv:=0 ;
   for ij:=low(fig,1) to high(fig,1) do
    begin { col }
    { linear transformation from 0-maxel to 0-lim }
    newval:=round(lim*fig[ij,jk]/maxel) ;
    write(mvpfil,newval:5) ;
    twelv:=twelv+1 ;
    if twelv=12 then
     begin { end line }
     twelv:=0 ;
     writeln(mvpfil) ;
     end ; { end line }
    end ; { col }
   writeln(mvpfil) ;
   end ; { row }
  end ; { execute formvp }
 end ; { for mvp }
{G procedures and functions:}
{

                getrow

}
procedure getrow(var dens:oned; var  view : text ) ;
{ reads rows at same plane from all radiographs,
given as 'section' rows down from cross point 'teey'}
 begin { getrow }
 reset(view) ;
 j:=15 ;
 for i:=1 to 512 do
 dens[i]:=0 ;
 for i:=1 to j-1 do readln(view) ;
 for j:=1 to 5 do
  begin { average 5 rows }
  for i:=1 to numpix do
   begin { read pixels }
   read(view,val) ;
   dens[i]:= dens[i] + val ;
   end ; { read pixels }
  readln(view) ;
  end ; { average 5 rows }
 for i:=1 to numpix do dens[i]:= dens[i]/5 ;
 for i:=1 to numpix do
  begin
  if dens[i]<48 then dens[i]:=0 ;
  if dens[i]>=48 then dens[i]:=dens[i]*3 ;
  end ;
 end ; { get row }
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{
```

```
                              jminmax

}
procedure jminmax(n1,n2,n3,n4 :integer ; var min,max:integer ) ;
{ returns min and max of four integers given }
  begin { jminmax }
  min:=n1 ; max:=n1 ;
  if n2<min then min:=n2 ;
  if n3<min then min:=n3 ;
  if n4<min then min:=n4 ;
  if n2>max then max:=n2 ;
  if n3>max then max:=n3 ;
  if n4>max then max:=n4 ;
  end ; { jminmax }
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                               Max0

}
function Max0{(a,b: integer): integer};
  begin {Max0}
  if a > b then Max0 := a else Max0 := b;
  end; {Max0}
{

                               Min0

}
function Min0{(a,b: integer): integer};
  begin {Min0}
  if a < b then Min0 := a else Min0 := b;
  end; {Min0}
{N procedures and functions:}
{O procedures and functions:}
{P procedures and functions:}
{

                               plot

}
procedure plot ;
{ prepares plots of projections in one page : works for np<=12}
  begin  { plotter instructions }
  plotopen(calcom,longx,nearside,11.0,11.0,'RANGARAJ') ;
  setscale(1.0) ;
  for l:=1 to np do
    begin { np plots }
    for i:= 1 to nr2 do
      begin
      x[i]:= i ; y[i]:= projection[l,i-nr-1] ;
      end ;
      case l of
      1:move(1.5,0) ;
      2:move(1.5,2.8) ;
      3:move(1.5,5.8) ;
{
      4:move(5.5,1.0) ;
      5:move(1.5,2.0) ;
      6:move(5.5,2.0) ;
      7:move(1.5,3.0) ;
      8:move(5.5,3.0) ;
      9:move(1.5,4.0) ;
      10:move(5.5,4.0) ;
      11:move(1.5,5.0) ;
      12:move(5.5,5.0) ;
}
      end ; { case }
    drawinteger(1,3,0.1,0.0) ;
    drawstring(' PROJN ANGLE',-1,0.1,0.0);
    drawreal(prangl[1],7,2,0.1,0.0) ;
      case 1 of
      1:move(1.5,0.2) ;
      2:move(1.5,2.85) ;
      3:move(1.5,5.85) ;
{
      4:move(5.5,1.2) ;
      5:move(1.5,2.2) ;
      6:move(5.5,2.2) ;
      7:move(1.5,3.2) ;
      8:move(5.5,3.2) ;
      9:move(1.5,4.2) ;
      10:move(5.5,4.2) ;
      11:move(1.5,5.2) ;
      12:move(5.5,5.2) ;
}
      end ; { case }
    scaledata(x,nr2,6,xinfo) ;
    scaledata(y,nr2,3.0,yinfo) ;
    drawline(x,y,nr2,xinfo,yinfo,lineonly,1) ;
    end ; { np plots }
  writeln(prompt,'THE PLOT IS ',plotclose:5:2,' INCHES LONG.')
  end ; { plotter instructions }
{Q procedures and functions:}
{R procedures and functions:}
{

                              raypixels

}
procedure raypixels(l,k:integer ) ;
{
Parallel ray geometry. Returns no. of pixels covered by ray(l,k) in npr,
their coords in pixel[1,i],pixel[2,i] and the pseudo projection in
pseudo. Raywidth rw, threshold thr, sin and cos of view angle sinl,cosl
to be defined in main program before calling raypixels.
Calls absmal,del,jminmax,min0,max0.
}
  begin { raypixels }
```

```
 npr:=0 ; pseudop:=0 ;
 { get limits of search in y/j direction }
 jj1:=round((-ni*sinl+(k-1)*rw)/cosl) ;
 jj2:=round((-ni*sinl+(k+1)*rw)/cosl) ;
 jj3:=round((ni*sinl+(k-1)*rw)/cosl) ;
 jj4:=round((ni*sinl+(k+1)*rw)/cosl) ;
 jj1:=absmal(jj1,mj) ; jj2:=absmal(jj2,mj) ;
 jj3:=absmal(jj3,mj) ; jj4:=absmal(jj4,mj) ;
 jminmax(jj1,jj2,jj3,jj4,jmin,jmax) ;
 for j:=jmin to jmax do
  begin { row }
  { get limits of search in x/i direction }
  jj1:=round((j*cosl-(k-1)*rw)/sinl) ; jj1:=absmal(jj1,ni-1) ;
  jj2:=round((j*cosl-(k+1)*rw)/sinl) ; jj2:=absmal(jj2,ni-1) ;
  imin:=min0(jj1,jj2)-1 ; imax:=max0(jj1,jj2)+1;
  for i:=imin to imax do
   begin  { column }
   if(del(i,j,k,sinl,cosl))and(pic[i,j]<>-999) then
    begin  { store ray pixels }
    npr:=npr+1 ; pseudop:=pseudop+pic[i,j] ;
    pixel[1,npr]:=i ; pixel[2,npr]:=j ;
    end;  { store ray pixels }
   end ; { column }
  end ; { row }
 end ; { raypixels }
{

             read_projections

}
procedure read_projections ;
 begin { read projections }
 for l:= 1 to np do
 for k:=-nr to nr do projection[l,k]:= 0 ;
 for l:= 1 to np do
  begin  { np projections }
  edgelt:=1 ; edgert:=numpix-1 ;
  angl:= prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:= cos(angl) ;
  rw:= amax1(abs(sinl),abs(cosl)) ;
   case l of
   1 : getrow(dens,p1) ;
   2 : getrow(dens,p2) ;
   3 : getrow(dens,p3) ;
{
   4 : getrow(dens,p4) ;
   5 : getrow(dens,p5) ;
   6 : getrow(dens,p6) ;
   7 : getrow(dens,p7) ;
   8 : getrow(dens,p8) ;
   9 : getrow(dens,p9) ;
   10: getrow(dens,p10) ;
   11: getrow(dens,p11) ;
}
   end ; { case }
  for k:= -nr to nr do
```

```
      begin  { rebinning }
      step:=-rw*k ;
{
      { correction for cassette skew }
      step:=-rw*k/abs(sin(angl)) ;
      { shift required to match centre of reconstruction matrix to centre
      of scan line from film }
      step:= step + mj*cosl/sinl ;
}
      { scaling to match size of image on radiograph to reconstruction matrix }
      step:= step*scanpix/nocol ;
      { measurement done w.r.t. 'teex' point on radiograph }
      trunk:= trunc(step + teex) ;
      if (trunk<edgelt) or (trunk>edgert) then projection[l,k]:=0 else
      projection[l,k]:= (dens[trunk] + dens[trunk+1])*0.5;
      end ;  { rebinning }
    end ;  { np projections }
  end ; { read projections }
{S procedures and functions:}
{

                          scale_projections

}
procedure scale_projections ;
{ scales projections such that their integrals are all same }
  begin { scale }
  avg:=0 ;
  for k:=-nr to nr do
  avg:= avg + projection[1,k] ;
  for l:=2 to np do
    begin { normalization }
    val:=0 ;
    for k:=-nr to nr do
    val:= val + projection[l,k] ;
    for k:=-nr to nr do
    projection[l,k]:= projection[l,k]*avg/val ;
    end ; { normalization }
  end ; { scale }
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{

                             view_average

}
procedure view_average ;
{ produces a picture which is the average of reconstructions
at end of each view }
  begin { view average }
  if l=1 then
    begin { initiate averagepic }
    rewrite(averagepic) ;
    for j:=-mj to mj do
      begin { row }
      for i:=-ni to ni do
      write(averagepic,pic[i,j]) ;
      writeln(averagepic) ;
      end ; { row }
    end  { initiate averagepic }
  else
    begin { add next view picture }
    reset(averagepic) ;
    rewrite(bufferpic) ;
    for j:=-mj to mj do
      begin { row }
      for i:= -ni to ni do
        begin { pixel }
        read(averagepic,val) ;
        val:=val + pic[i,j] ;
        write(bufferpic,val) ;
        end ; { pixel }
      readln(averagepic) ;
      writeln(bufferpic) ;
      end ; { row }
    rewrite(averagepic) ;
    reset(bufferpic) ;
    for j:= -mj to mj do
      begin { swap files }
      for i:=-ni to ni do
        begin { interchange }
        read(bufferpic,val) ;
        write(averagepic,val) ;
        end ; { interchange }
      readln(bufferpic) ;
      writeln(averagepic) ;
      end ; { swap files }
    end ; { add next view picture }
  if l=np then
    begin { transfer picture }
    reset(averagepic) ;
    for j:=-mj to mj do
      begin { put picture in pic }
      for i:=-ni to ni do
        begin { divide }
        read(averagepic,val) ;
        pic[i,j]:=val/np ;
        end ; { divide }
      readln(averagepic) ;
      end ; { put picture in pic }
    end ; { transfer picture }
  end ; { view average }
{W procedures and functions:}
{X procedures and functions:}
{

                                xerror

}
procedure xerror ;
{computes error of reconstruction comparing projection data}
```

```
  begin { error }
  error:=0.0 ;
  for l:= 1 to np do
    begin  { np projections }
    angl:= prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:= -nr to nr do
      begin  { ray }
      raypixels(l,k) ;
      error:=error + abs(projection[l,k]-pseudop) ;
      end ;  { ray }
    end ;  { np projections }
  error:=error/(norow*nocol*np) ;
  end ; { error }
{Y procedures and functions:}
{Z procedures and functions:}
{

                               zero_rays

}
procedure zero_rays ;
  begin { zero rays }
  for l:=1 to np do
    begin { projections }
    angl:=prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:=-nr to nr do
    if projection[l,k]=0.0 then
      begin { label zero ray pixels }
      raypixels(l,k) ;
      for inpr:=1 to npr do
      pic[pixel[1,inpr],pixel[2,inpr]]:=-999 ;
      end ; { label zero ray pixels }
    end ; { projections }
  end ; { zero rays }
{

                               Main Block

}
  begin {ctrp}
  rewrite(calcom) ;
  writeln(prompt,' START TIME ' ,clock:10) ;
  prangl[1]:=45.0;
  prangl[2]:=90.0;
  prangl[3]:=180.0;
  read_projections ;
{
  chop_sides ;
}
  scale_projections ;
  centroid_plot ;
  {shift projection data such that all centroid rays intersect at one point }
```

```
centroid_correction ;
centroid_plot ;
plot ;
writeln(prompt,' AFTER PROJECTIONS ',clock:10) ;
{ }
{   ..............#########################.................    }
{ }
{ reconstruction by summation }
for i:=-ni to ni do
for j:=-mj to mj do
pic[i,j]:=0 ;
for i:=-ni to ni do
for j:=-mj to mj do
for l:= 1 to np do
  begin  { summation  }
  angl:=prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:=cos(angl) ;
  k:=round(j*cosl - i*sinl) ;
  pic[i,j]:=pic[i,j] + projection[l,k] ;
  end ; { summation } ;
for i:=-ni to ni do
for j:=-mj to mj do
pic[i,j]:=pic[i,j]/np ;
xerror ;
writeln(prompt,' SUMMATION ERROR : PROJECTION ',error) ;
rewrite(mvpl) ;
formvp(pic,2*mj+1,2*ni+1,8,mvpl) ;
writeln(prompt,' AFTER SUMMATION ',clock:10 ) ;
{ }
{ .................... ############### .....................}
{REconstruction by ART}
avg:=0.0 ;
for l:=1 to np do
for k:=-nr to nr do avg:=avg+projection[l,k] ;
avg:=avg/(nocol*norow*np) ;
for i:=-ni to ni do
for j:=-mj to mj do pic[i,j]:=avg ;
iteration:=0 ;error:=0.0 ; preerr:=99999999.9 ;
if artzero then { label zero ray pixels }
zero_rays ;
  repeat  { ART }
  if iteration>0 then preerr:=error ;
  iteration:=iteration+1 ; error:=0 ;
  for l:= 1 to np do
    begin  { projection  }
    angl:=prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amaxl(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:=-nr to nr do
      begin { rays }
      omit:=false ;
      if artzero and (projection[l,k]=0.0) then omit:=false ;
      if not omit then
        begin  { ray }
        raypixels(l,k) ;
        for inpr:=1 to npr do
```

```
     begin { correction }
     ip:=pixel[1,inpr] ; jp:=pixel[2,inpr] ;
     if artadd then  { additive ART }
     pic[ip,jp]:=pic[ip,jp]+(projection[1,k]-pseudop)/npr ;
     if artmul then  { multiplicative ART }
      begin
      if pseudop<>0.0 then
      pic[ip,jp]:= pic[ip,jp]*projection[1,k]/pseudop
      else pic[ip,jp]:=0.0 ;
      end ;
     { if constrained ART, check for negative values }
     if (artconstr) and (pic[ip,jp]<0) then pic[ip,jp]:= 0 ;
     { clip at upper limit if doubly constrained }
     if (artupconstr)and(pic[ip,jp]>artuplim) then pic[ip,jp]:=artuplim ;
     end;{ correction }
    end ; { ray }
   end ; { rays }
  end;  { projection }
 xerror ;
 writeln(prompt,' ITERATION ',iteration:3,' PROJECTION ERROR ',error) ;
 { a suitable criterion on the error may be included here to stop ART }
 until (iteration=5)or(error>0.90*preerr) ;  { ART }
if artzero then { set zero ray pixels to zero }
for j:=-mj to mj do
for i:=-ni to ni do
if pic[i,j]=-999 then pic[i,j]:=0.0 ;
rewrite(mvp2) ;
formvp(pic,norow,nocol,nobits,mvp2) ;
writeln(prompt,' CPU TIME USED ',clock:10,' MILLISECONDS.') ;
end.  { ctrp }
```

```
{      Z$NOBELLS$B$READP     DCONV1$T$XN$XL/end./$XN$XCOPYD     DCONVO
*$FILE,DCONVO,R$T$XCOPY          DCONV1          *$FILE,DCONV1,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/version_/


          The  above  XEDIT  commands  create  two  backup  files  and copy
      compiler error messages to the end, as a comment, if any.

          This file may be printed out using the commands:

            GET,FORMPRT/UN=APPLIB

            CALL,FORMPRT(I=DCONVP)
      }

      {
      Title....DCONV
      Author: Rangaraj M. Rangayyan   Atam P. Dhawan
      Address: Department of Electrical Engineering,
           University of Manitoba, Winnipeg, Canada R3T 2N2

                              Purpose

           This program computes inverse point spread functions  (PSF)  from
      the  original and reconstructed projections. The corresponding PSF for
      each view is stored in the file dconfn. The input files  for  original
      and reconstructed projections are projo and projr respectively.
      }
      {

                            Program Statement

      }
      program Dconv(input/,output,prompt,projo,projr,dconfn,lstfrm,nxtfrm,calcom);
      {
                          Use of External Files

      input....terminal input
      output...debugging output and Pascal dump on error exits
      prompt...terminal output
      lstfrm...last frame recorded
      nxtfrm...next frame to be output by this program
      calcom...batch PASPLOT output, remove for interactive PASPLOT
      projo...input file: original projections.
      projr...input file: reconstructed projections
      dconfn..output file: inverse point spread functions

      Execute as:
      /DCONVX,INPUT,DCONVU,DCTPUT,DCONVI,DCONVO,DCONVZ.
      }
      {

                            Associated Files

      DCONVO  oldest backup of DCONVP
      DCONV1  most recent backup of DCONVP
DCONVC  interactive initialization, then submits DCONVS,
         CALL,DCONVC(FRAME=DCrun_number0)
DCONVD  dayfile formed during batch runs.
DCONVE  XEDCT commands for showing key parameters before display
DCONVL  for printout CALL,SUBMITC(NAME=DCONVL)
DCONVM  for compilation CALL,SUBMITC(NAME=DCONVM)
DCONVP  this file (Pascal source)
DCONVS  for batch execution CALL SUBMITC(NAME=DCONVS)
DCONVU  error messages, if any
DCONVX  executable binary.
DCONVZ  for Tektronix-type display CALL,DCONVZ(FRAME=DCrun_numbern)
DCrun_number0  initial parameters for a run
DCrun_numberD  dayfile for a run
DCrun_numberI  direct access version of DCrun_number0, etc.,
               for MVP input via the GETFILE command.
DCrun_numberN  to stop batch run PURGE,DCrun_numberN
DCrun_numberO  parameters for last recorded frame of a run
DCrun_numberU  error messages from a run
DCrun_numberY  accumulated frames from a run, direct access file
DCrun_numberZ  plot file for a frame from a run
SUBMITE        contains root of file names (set up by DCONVC)
USERE          contains user number and password
}
{

                      Compiler Include Directives

}
{$I'PASPLOT'}
{

                                Labels

}
{

                               Constants

}
Const
{A constants:}
{B constants:}
{C constants:}
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
mft=9 ;
{N constants:}
nft=512 ; npjn=8 ; nrays=221 ;
```

```
{O constants:}
{P constants:}
plotting = true;  {Indicates if standard (true) or interactive
(false)
{}                 PASPLOT package has been included.}
{Q constants:}
{R constants:}
{S constants:}
{T constants:}
{U constants:}
{V constants:}
version_number = 1; version_date = 'MAR 13, 1982';

{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                                  Types

     Primary  types,  defined  in terms of constants, are given first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                              Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
```

```
                              Compound Types
}
{

                               Variables

}
Var
{A variables:}
{B variables:}
batch: boolean;  {Indicates if batch (true) or interactive (false)
run}
{C variables:}
calcom: CALCOM_FILE;  {File which receives the batch PASPLOT plot}
current_date: alfa; {Today's date}
current_time: alfa; {Now}
{D variables:}
denom:real ;
dconfn:text ;
{E variables:}
{F variables:}
{G variables:}
{H variables:}
{I variables:}
{J variables:}
{K variables:}
k:integer ;
{L variables:}
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
min,max:real ;
{N variables:}
nxtfrm: text; {See program statement}
{O variables:}
{P variables:}
projo,projr:text ;
plot_scale: real; {Scale factor for plotting}
prompt: text; {Call to program should be
{}              /DCONVX,,DCONVU,DCTPUT, etc., to
{}             place any bugs into file DCONVU and have the prompts
{}             come out on DCTPUT. All prompts are to file PROMPT
{}             which thereby becomes DCTPUT.}
{Q variables:}
{R variables:}
ran_seed_1, ran_seed_2: integer; {Random # seeds for use with RANDOM}
run_number: integer;  {Number assigned to files: DCrun_number etc.}
{S variables:}
sum1,sum2,sum3:real ;
{T variables:}
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
{W variables:}
{X variables:}
xreal2,ximag2:array[1..nft] of real ;
```

```
xinfo:scaleinfo ;
xreal,xreall,ximag,ximagl,x:array[1..nft] of real ;
{Y variables:}
yinfo:scaleinfo ;
y:array[1..nft] of real ;
{Z variables:}
zeropts:integer ;
{

     In the University of  Minnesota  Pascal,  only  single  variables
which  are  global  are  included  in  dumps.  For this reason we have
permitted a GLOBAL statement which  makes  a  local  variable  global.
Duplicated  names, however, are not permitted. These GLOBAL statements
are gathered together in the file GLOBALP and brought  back  into  the
program here:
}
{$I'GLOBAL'/'GLOBALP'}
{

                          Forward Declarations

}
{Forward declarations of procedures and functions:}
{A forward declarations:}
{B forward declarations:}
procedure Bomb_out (bomb_condition: alfa);  forward;
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
{

                        Procedures and Functions

}
{Procedures and functions:}
```

```
{A procedures and functions:}
{B procedures and functions:}
{
                               Bomb_out

     Bomb_out  allows  the user to stop the program on detection of an
unwanted condition. The condition must be exactly 10  characters  long
(lower case letters count for two). Example:

Bomb_out('BAD PARAM ');
}
procedure Bomb_out {(bomb_condition: alfa)};
  begin {Bomb_out}
  writeln(prompt, 'Program terminated because of condition: ',
  {}              bomb_condition);
  writeln(output, 'Program terminated because of condition: ',
  {}              bomb_condition);
  halt(bomb_condition);
  end; {Bomb_out}
{C procedures and functions:}
{D procedures and functions:}
{E procedures and functions:}
{F procedures and functions:}
{

                                ftbtr

}
function ftbtr(jbtr,mft:integer):integer;
var ibtr,jbtr1,jbtr2,kbtr:integer ;
  begin { ftbtr }
  jbtr1:=jbtr ; kbtr:=0 ;
  for ibtr:= 1 to mft do
    begin
    jbtr2:=jbtr1 div 2 ;
    kbtr:=kbtr*2 + (jbtr1-2*jbtr2) ;
    jbtr1:=jbtr2 ;
    end ;
  ftbtr:=kbtr ;
  end ; { ftbtr }
{

                                 FFT

}
procedure FFT;
{ Fast Fourier Transform program from E.O.Brigham
Translated from Fortran
Arrays xreal and ximag must contain real and imaginary parts
of input signal. The same arrays will contain on return the
real and imaginary parts of the Fourier Transform- the input
data are lost. NFT and MFT must contain no. of FFT samples
required, and NFT=2**MFT. }
var kft,lft,ift,kft1,kft1n2,nft2,mft1,jft,tupwr:integer ;
var pft,arg,cft,sft,treal,timag:real ;
  begin { fft }
```

```
 nft2:=nft div 2 ; mft1:=mft-1 ; kft:=0 ;
 for lft:=1 to mft do
  begin { loop 100 }
  while kft<nft do
   begin { loop 102 }
   for ift:=1 to nft2 do
    begin { loop 101 }
    tupwr:=1 ;
    for jft:=1 to mft1 do
    tupwr:=tupwr*2 ;
    pft:= ftbtr(kft div tupwr,mft) ;
    arg:=6.283185*pft/nft ;
    cft:=cos(arg) ; sft:=sin(arg) ;
    kft1:=kft+1 ; kft1n2:=kft1+nft2 ;
    treal:= xreal[kft1n2]*cft + ximag[kft1n2]*sft ;
    timag:= ximag[kft1n2]*cft - xreal[kft1n2]*sft ;
    xreal[kft1n2]:=xreal[kft1]-treal ;
    ximag[kft1n2]:=ximag[kft1]-timag ;
    xreal[kft1]:= xreal[kft1] + treal ;
    ximag[kft1]:= ximag[kft1] + timag ;
    kft:=kft+1 ;
    end ; { loop 101 }
   kft:= kft + nft2 ;
   end ; { loop 102 }
  kft:= 0 ; mft1:= mft1-1 ;
  nft2:= nft2 div 2 ;
  end ; { loop 100 }
 for kft:= 1 to nft do
  begin { loop 103 }
  ift:= ftbtr(kft-1,mft) + 1 ;
  if ift>kft then
   begin { inner loop of 103 }
   treal:= xreal[kft] ;
   timag:= ximag[kft] ;
   xreal[kft]:=xreal[ift] ; ximag[kft]:=ximag[ift] ;
   xreal[ift]:=treal ; ximag[ift]:=timag ;
   end ; { inner loop of 103 }
  end ; { loop 103 }
 end ; { fft }
{G procedures and functions:}
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                minmax

}
procedure minmax ;
{ extracts min and max values of
power spectrum in array y[1..nft] and prints them }
 begin { minmax }
 min:=99999999.99 ; max:=0 ;
```

```
  for k:=1 to nft do
  if y[k]<min then min:=y[k]
  else
  if y[k]>max then max:=y[k] ;
  writeln(prompt,' MIN AND MAX VALUES OF POWER SPECTRUM ',min,max) ;
  end ; { minmax }
{N procedures and functions:}
{O procedures and functions:}
{P procedures and functions:}
{

                                 plot

}
procedure plot ;
  begin  { plotter instructions }
  setscale(1.0) ;
  move(1,1) ;
  scaledata(x,nft,6,xinfo) ;
  scaledata(y,nft,4,yinfo) ;
  drawline(x,y,nft,xinfo,yinfo,lineonly,1) ;
  end ; { plotter instructions }
{Q procedures and functions:}
{R procedures and functions:}
{S procedures and functions:}
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{W procedures and functions:}
{X procedures and functions:}
{Y procedures and functions:}
{Z procedures and functions:}
{

                              Main Block

}
  begin {Dconv}
  reset(projo) ; reset(projr) ; rewrite(dconfn) ;
  plotopen(calcom,longx,nearside,11.0,7.0,'RANGARAJ') ;
  for k:= 1 to nft do
  x[k]:=k ;
  for l:= 1 to npjn do
    begin { projections }
    zeropts:=0 ;
    for k:= 1 to nrays do
    read(projo,xreal[k]) ;
    readln(projo) ;
    for k:= nrays+1 to nft do
    xreal[k]:= 0 ;
    for k:=1 to nft do y[k]:=xreal[k] ; plot ; { original projection }
    sum1:=0 ; for k:=1 to nft do sum1:=sum1 + xreal[k] ;
    for k:= 1 to nft do ximag[k]:= 0 ;
    fft ;
    for k:= 1 to nft do
      begin { store FT }
  y[k]:=sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
  xreal1[k]:= xreal[k] ; ximag1[k]:= ximag[k] ;
  end ; { store FT }
plot ; { power spectrum of original projection }
minmax ;
for k:= 1 to nrays do read(projr,xreal[k]) ;
readln(projr) ;
for k:=nrays+1 to nft do xreal[k]:= 0 ;
sum2:=0 ; for k:=1 to nft do sum2:=sum2 + xreal[k] ;
for k:=1 to nft do xreal[k]:=xreal[k]*sum1/sum2 ;
for k:=1 to nft do y[k]:=xreal[k] ; plot ; { projection of reconstruction }
for k:= 1 to nft do ximag[k]:= 0 ;
fft ;
for k:=1 to nft do
  begin { store FT }
  xreal2[k]:=xreal[k] ; ximag2[k]:=ximag[k] ;
  end ; { store FT }
for k:= 1 to nft do
  begin { divide FTs }
  denom:= xreal[k]*xreal[k] + ximag[k]*ximag[k] ;
  y[k]:=sqrt(denom) ;
  if denom=0 then
    begin
    zeropts:=zeropts + 1 ;
    xreal[k]:=0 ; ximag[k]:=0 ;
    end
  else
    begin
    xreal[k]:= (xreal[k]*xreal1[k] + ximag[k]*ximag1[k])/denom ;
    ximag[k]:= (xreal[k]*ximag1[k] - ximag[k]*xreal1[k])/denom ;
    end ;
  end ; { divide FTs }
plot ; { power spectrum of reconstruction }
minmax ;
for k:=1 to nft do
write(dconfn,xreal[k],ximag[k]) ;
writeln(dconfn) ;
for k:= 1 to nft do
  begin { prepare for IFT }
  y[k]:= sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
  xreal1[k]:=xreal[k] ; ximag1[k]:=ximag[k] ;
  xreal[k]:= xreal[k]/nft ; ximag[k]:= -ximag[k]/nft ;
  end ; { prepare for IFT }
plot ; { PSF inverse }
minmax ;
fft ;
for k:=1 to nft do
  begin { prepare for deconvolution }
  xreal[k]:=xreal2[k]*xreal1[k]/nft ; ximag[k]:=-ximag1[k]*ximag2[k]/nft ;
  end ; { prepare for deconvolution }
fft ;
for k:= 1 to nft do y[k]:=xreal[k] ;
plot ; { deconvolved projection }
sum3:=0 ; for k:=1 to nft do sum3:= sum3 + xreal[k] ;
writeln(prompt,' PROJECTION INTEGRALS ',sum1,sum2,sum3) ;
writeln(prompt,' ZERO DENOMINATOR POINTS ',zeropts:6) ;
```

```
    writeln(prompt,' FINISHED PROJECTION ',1:2) ;
    end ; { projections }
  end. { Dconv }
```

```
{      Z$NOBELLS$B$READP      CART1$T$XN$XL/end./$XN$XCOPYD      CARTO
*$FILE,CARTO,R$T$XCOPY          CART1          *$FILE,CART1,R$XB$READP
PU$T$XN$XL/end./$P*$XN$L/versio

The  above  XEDIT  commands  create two backup files and copy compiler
error messages to the end, as a comment, if any. }

{
      }
      {
      Title:CART:CONVOLUTION ART
      Author:M.R.Rangaraj
      Modification and rewritten by Atam P. Dhawan, Department of Electrical eng.
      Address:Department of Pathology, University of Manitoba,
            Medical College, 770 Bannatyne avenue,
            Winnipeg, Manitoba, Canada R3E OW3.

                              Program description :
      This program returns reconstructions of a given picture by
      additive/multiplicative ART.
      Parallel ray geometry is used throughout. Projections( views )
      of the picture are first computed and then using them, the
      picture is reconstructed. Parameters of the picture and
      reconstruction method to be used are to be entered in the
      constants list and within the main program as well.
      THe projections are convolved with the inverse PSF
      supplied to get rid of distortions due to limited
      angle coverage.

      }
      {

                                Program Statement

      }
      program
      cartp(input/,inpic,output,prompt,lstfrm,nxtfrm,calcom,convfn,mvp1,mvp2,mvp3) ;
      {
                              Use of External Files

      input:   terminal input
      output:  debugging output and Pascal dump on error exits
      prompt:  terminal output
      lstfrm:  last frame recorded
      nxtfrm:  next frame to be output by this program
      calcom:  output file for PASPLOT commands
      inpic: input test picture
      mvp1:  test picture returned in MVP format 5
      mvp2:  reconstruction by summation
      mvp3:  ART reconstruction

      Execute as:
      /CARTX,INPUT,INPIC,CARTU,OUTPUT,CARTI,CARTO,CARTZ,MVP1,MVP2,
      MVP3.
      }
      {
```

```
                         Associated Files

CARTC   for execution
CARTM   for batch compilation
CARTS for batch job submission
}
{

                     Compiler Include Directives

}
{$I'PASPLOT'}
{$I'MATH'}
{

                              Labels

}
{

                             Constants

}
Const
{A constants:}
{ for additive ART set artadd to true ; multiplicative ART set artmul to true ;
for constrained ART set artconstr to true ; set others to false }
artadd=false;artmul=true; artconstr=true;
{B constants:}
{C constants:}
{D constants:}
{E constants:}
{F constants:}
{G constants:}
{H constants:}
{I constants:}
{J constants:}
{K constants:}
{L constants:}
{M constants:}
{ max (+-) y/j index of picture ( origin at center of picture ) }
mj=50 ;
{N constants:}
{ No. of rows and columns of picture and no. of bits per word in MVP output file
}
norow=101 ; nocol=101 ; nobits=8 ;
{
ni:max(+-) x/i index of picture; np:no. of projections/views to be used;
nr:max(+-) index no. of rays; nr2:total no. of rays per view(2*nr+1) ;
nprmax:max no. of pixels a ray can cross - raywidth in pixels * max of
norow,nocol .
}
ni=50;np=8; nr=110; nr2=221;nprmax=101;
nft=512 ; mft=9 ;
{O constants:}
```

```
{P constants:}
pi = 3.1415926536;
{Q constants:}
{R constants:}
{S constants:}
{T constants:}
{U constants:}
{V constants:}
version_number = 1;
{W constants:}
{X constants:}
{Y constants:}
{Z constants:}
{

                                   Types

     Primary types, defined in terms of constants,  are  given  first,
followed by compound types, which are defined in terms of other types.
}
Type
{
                                Primary Types
}
{A types:}
{B types:}
{C types:}
{D types:}
{E types:}
{F types:}
{G types:}
{H types:}
{I types:}
in_file = (input_file, lstfrm_file);
{J types:}
{K types:}
{L types:}
{M types:}
{N types:}
{O types:}
{P types:}
{Q types:}
{R types:}
{S types:}
{T types:}
twod=array[integer,integer] of real ;
twod1=array[1..np,-nr..nr] of real ;
twod2=array[-ni..ni,-mj..mj] of real ;
{U types:}
{V types:}
{W types:}
{X types:}
{Y types:}
{Z types:}
{
                                Compound Types
```

```
}
{
                              Variables
}
Var
{A variables:}
averagepic: text ;
angl,avg:real ;
{B variables:}
bufferpic: text ;
{C variables:}
convfn:text ;
cosl:real ;
calcom: calcomfile; {Needed by PASPLOT}
{D variables:}
dcpower:real ;
{E variables:}
euclidErr : real ;
error: real ;
{F variables:}
{G variables:}
{H variables:}
{I variables:}
i,ip,imin,imax,inpr,iteration:integer ;
inpic: text ;
{J variables:}
j,jj1,jj2,jj3,jj4,jmin,jmax,jp:integer ;
{K variables:}
k:integer ;
{L variables:}
ll:integer ;
l:integer ;
lstfrm: text; {See program statement}
{M variables:}
maxfac:real ;
mvpl,mvp2,mvp3:text ;
{N variables:}
nfth:integer ;
nxtfrm: text; {See program statement}
npr:integer ;
{O variables:}
{P variables:}
pjnum: array[1..np] of integer ;
preerr:real ; {error at previous iteration }
{ pseudo projection of ray }
pseudop: real ;
{ picture array -ni..ni, -mj..mj }
pic:twod2 ;
{ angles of projections : to be entered in main program }
prangl: array[1..np] of real ;
{ first dimension refers to view no. and second to ray no. }
projection:twodl ;
pixel:array[1..2,1..nprmax] of integer ;
prompt: text; {Call to program should be
```

```
{}              /CARTX,,CARTU,OUTPUT, etc., to
{}             place any bugs into file CARTU and have the prompts
{}             come out on OUTPUT. All prompts are to file PROMPT
{}             which thereby becomes OUTPUT.}
{Q variables:}
{R variables:}
{ ray width : modify in main program at all locations if necessary }
rw:real ;
{S variables:}
suppd:integer ;
sum1,sum2:real ;
spot:real ;
sinl:real ;
scale: real; {Scale factor for plotting}
{T variables:}
{ threshold for ray boundary = 0.5 * ray width }
thr:real ;
testing: boolean; {Flag for debugging}
{U variables:}
{V variables:}
version_date: alfa; {Date compiled}
version_time: alfa; {Time compiled}
val:real ;
{W variables:}
{X variables:}
xreal,ximag,xabs:array[1..nft]of real ;
x:array[1..nr2] of real ;
xinfo:scaleinfo ;
{Y variables:}
y:array[1..nr2] of real ;
yinfo:scaleinfo ;
{Z variables:}
{

                        Forward Declarations

}
{Forward declarations of procedures and functions:}
{A forward declarations:}
function Amax1(a,b: real): real; forward;
function Amin1(a,b: real): real; forward;
{B forward declarations:}
{C forward declarations:}
{D forward declarations:}
{E forward declarations:}
{F forward declarations:}
function Factorial(n: integer): real; forward;
{G forward declarations:}
{H forward declarations:}
{I forward declarations:}
{J forward declarations:}
{K forward declarations:}
{L forward declarations:}
{M forward declarations:}
function Max0(a,b: integer): integer; forward;
function Min0(a,b: integer): integer; forward;
```

```
{N forward declarations:}
{O forward declarations:}
{P forward declarations:}
{Q forward declarations:}
{R forward declarations:}
{S forward declarations:}
{T forward declarations:}
{U forward declarations:}
{V forward declarations:}
{W forward declarations:}
{X forward declarations:}
{Y forward declarations:}
{Z forward declarations:}
{

                       Procedures and Functions

}
{Procedures and functions:}
{A procedures and functions:}
{

                                Absmal

}
function absmal(one,two:integer):integer ;
{ returns no. with smaller absolute value
and sign of the first no. }
  begin { absmal }
  if one < 0 then
    begin
    if abs(one) < two then absmal:=one else absmal:=-two
    end
  else
    begin
    if one < two then absmal:=one else absmal:=two
    end
  end ; { absmal }
{

                                Amax1

}
function Amax1{(a,b: real): real};
  begin {Amax1}
  if a > b then Amax1 := a else Amax1 := b;
  end; {Amax1}
{

                                Amin1

}
function Amin1{(a,b: real): real};
  begin {Amin1}
  if a < b then Amin1 := a else Amin1 := b;
  end; {Amin1}
```

```
{B procedures and functions:}
{

               boas_kac

}
procedure boas_kac ;
{ Boas & Kac clipping procedure }
  begin { boas_kac }
  dcpower:=sqrt(xreal[1]*xreal[1] + ximag[1]*ximag[1]) ;
  nfth:= nft div 2 ;
  nfth:= nfth + 1;
  suppd:=0 ;
  for k:=2 to nfth do
    begin { peak suppression }
    val:= sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
    maxfac:= pi/(2 + nfth div k) ;
    maxfac:= cos(maxfac) ;
    if val>(maxfac*dcpower) then
      begin { clipping }
      xreal[k]:= xreal[k]*maxfac*dcpower/val ;
      ximag[k]:= ximag[k]*maxfac*dcpower/val ;
      suppd:=suppd + 1;
      end ; { clipping }
    end ; { peak suppression }
  end ; { boas_kac }
{C procedures and functions:}
{D procedures and functions:}
{

                del

}
function del(i,j,k:integer;sinl,cosl:real):boolean ;
{ del : true if pixel(i,j) belongs to ray (l,k) }
var dist,jy:real ;
  begin  { del  }
  jy:=i*sinl + k*rw  ; jy:= jy/cosl ;
  dist:= abs(j-jy)*abs(cosl);
  if dist<=thr then del:=true else del:=false
  end ;  { del }
{E procedures and functions:}
{

              euclid_error ;

}
procedure euclid_error ;
{ computes the Euclidean distance between the original picture
and its reconstruction }
  begin { euclid_error }
  eucliderr:=0.0 ;
  reset(inpic) ;
  for j:=-mj to mj do
    begin { row }
    for i:=-ni to ni do
```

```
      begin { pixels }
      read(inpic,val) ;
      val:=val-pic[i,j];
      eucliderr:=eucliderr + val*val ;
      end ; { pixels }
    readln(inpic) ;
    end ; { row }
  eucliderr:=eucliderr/(norow*nocol) ;
  end ; { euclid_error }
{F procedures and functions:}
{

                                ftbtr

}
function ftbtr(jbtr,mft:integer):integer;
var ibtr,jbtr1,jbtr2,kbtr:integer ;
  begin { ftbtr }
  jbtr1:=jbtr ; kbtr:=0 ;
  for ibtr:= 1 to mft do
    begin
    jbtr2:=jbtr1 div 2 ;
    kbtr:=kbtr*2 + (jbtr1-2*jbtr2) ;
    jbtr1:=jbtr2 ;
    end ;
  ftbtr:=kbtr ;
  end ; { ftbtr }
{

                                 FFT

}
procedure FFT;
{ Fast Fourier Transform program from E.O.Brigham
Translated from Fortran
Arrays xreal and ximag must contain real and imaginary parts
of input signal. The same arrays will contain on return the
real and imaginary parts of the Fourier Transform- the input
data are lost. NFT and MFT must contain no. of FFT samples
required, and NFT=2**MFT. }
var kft,lft,ift,kft1,kft1n2,nft2,mft1,jft,tupwr:integer ;
var pft,arg,cft,sft,treal,timag:real ;
  begin { fft }
  nft2:=nft div 2 ; mft1:=mft-1 ; kft:=0 ;
  for lft:=1 to mft do
    begin { loop 100 }
    while kft<nft do
      begin { loop 102 }
      for ift:=1 to nft2 do
        begin { loop 101 }
        tupwr:=1 ;
        for jft:=1 to mft1 do
        tupwr:=tupwr*2 ;
        pft:= ftbtr(kft div tupwr,mft) ;
        arg:=6.283185*pft/nft ;
        cft:=cos(arg) ; sft:=sin(arg) ;
    kft1:=kft+1 ; kft1n2:=kft1+nft2 ;
    treal:= xreal[kft1n2]*cft + ximag[kft1n2]*sft ;
    timag:= ximag[kft1n2]*cft - xreal[kft1n2]*sft ;
    xreal[kft1n2]:=xreal[kft1]-treal ;
    ximag[kft1n2]:=ximag[kft1]-timag ;
    xreal[kft1]:= xreal[kft1] + treal ;
    ximag[kft1]:= ximag[kft1] + timag ;
    kft:=kft+1 ;
    end ; { loop 101 }
   kft:= kft + nft2 ;
   end ; { loop 102 }
  kft:= 0 ; mft1:= mft1-1 ;
  nft2:= nft2 div 2 ;
  end ; { loop 100 }
 for kft:= 1 to nft do
  begin { loop 103 }
  ift:= ftbtr(kft-1,mft) + 1 ;
  if ift>kft then
   begin { inner loop of 103 }
   treal:= xreal[kft] ;
   timag:= ximag[kft] ;
   xreal[kft]:=xreal[ift] ; ximag[kft]:=ximag[ift] ;
   xreal[ift]:=treal ; ximag[ift]:=timag ;
   end ; { inner loop of 103 }
  end ; { loop 103 }
 end ; { fft }
{

               Factorial

}
function Factorial{(n: integer): real};
 begin {Factorial}
 if n = 0 then Factorial := 1
 else Factorial := n * Factorial(n-1);
 end; {Factorial}
{

                formvp

}
procedure formvp( var fig:dynamic twod;nrow,ncol,bits:integer; var mvpfil:text);
{ writes picture in MVP format 5 in file mvpfil }
var ij,jk,newval,twelv:integer ; maxel,minel,lim:real ;okay:boolean ;
 begin { for mvp }
 okay:=true ;
 if (nrow<1) or (nrow>512) or (ncol<1) or (ncol>512) then
  begin { error 2 }
  writeln(prompt,'Maximum picture size permitted is 512 X 512') ;
  writeln(output,'Maximum picture size permitted is 512 X 512') ;
  okay:=false ;
  end ; { error 2 }
 if (bits<1) or (bits>12) then
  begin { error 3 }
  writeln(prompt,'MVP resolution limited to 1..12 bits') ;
  writeln(output,'MVP resolution limited to 1..12 bits') ;
  okay:=false ;
  end ; {error 3 }
 if okay then
  begin { execute formvp }
  maxel:=0 ;
   case bits of  { resolution }
   1: lim:=1 ;
   2: lim:=3 ;
   3: lim:=7 ;
   4: lim:=15 ;
   5: lim:=31 ;
   6: lim:=63 ;
   7: lim:=127 ;
   8: lim:=255 ;
   9: lim:=511 ;
   10:lim:=1023 ;
   11:lim:=2047 ;
   12:lim:=4095 ;
   end ;    { resolution }
  ij:=low(fig,1) ; jk:=low(fig,2) ; minel:=fig[ij,jk] ;
  for ij:=low(fig,1) to high(fig,1)do
  for jk:=low(fig,2) to high(fig,2) do
   begin { minmax }
   if maxel<fig[ij,jk] then maxel:=fig[ij,jk] ;
   if minel>fig[ij,jk] then minel:=fig[ij,jk] ;
   end ; { minmax }
  writeln(mvpfil,'PICTURE FILE FOR DISPLAY USING MVP.') ;
  writeln(mvpfil,'ENDOFHEADR',ncol:5,nrow:5,bits:5,1:5) ;
  for jk:=low(fig,2) to high(fig,2) do
   begin { row }
   twelv:=0 ;
   for ij:=low(fig,1) to high(fig,1) do
    begin { col }
    {linear transformation from 0-maxel to 0-lim}
    newval:=round(lim*fig[ij,jk]/maxel) ;
    if newval<0 then newval:=0 ;
    write(mvpfil,newval:5) ;
    twelv:=twelv+1 ;
    if twelv=12 then
     begin { end line }
     twelv:=0 ;
     writeln(mvpfil) ;
     end ; { end line }
    end ; { col }
   writeln(mvpfil) ;
   end ; { row }
  end ; { execute formvp }
 end ; { for mvp }
{G procedures and functions:}
{H procedures and functions:}
{I procedures and functions:}
{J procedures and functions:}
{

               jminmax
```

```
}
procedure jminmax(n1,n2,n3,n4 :integer ; var min,max:integer ) ;
{ returns min and max of four integers given }
  begin { jminmax }
  min:=n1 ; max:=n1 ;
  if n2<min then min:=n2 ;
  if n3<min then min:=n3 ;
  if n4<min then min:=n4 ;
  if n2>max then max:=n2 ;
  if n3>max then max:=n3 ;
  if n4>max then max:=n4 ;
  end ; { jminmax }
{K procedures and functions:}
{L procedures and functions:}
{M procedures and functions:}
{

                                 MaxO

}
function MaxO{(a,b: integer): integer};
  begin {MaxO}
  if a > b then MaxO := a else MaxO := b;
  end; {MaxO}
{

                                 MinO

}
function MinO{(a,b: integer): integer};
  begin {MinO}
  if a < b then MinO := a else MinO := b;
  end; {MinO}
{N procedures and functions:}
{O procedures and functions:}
{

                              other_half

}
procedure other_half ;
{duplicates second half of Fourier Transform array}
  begin { other_half }
  nfth:= nft div 2;
  nfth:= nfth + 1 ;
  for k:= nft downto nfth+1 do
    begin { other half of spectrum }
    xreal[k]:= xreal[nft-k+2] ;
    ximag[k]:=ximag[nft-k+2] ;
    end ; { other half of spectrum }
  end ; { other_half }
{P procedures and functions:}
{

                                 plot
```

```
}
procedure plot ;
{ prepares 8 plots in one page : works for np=8 only  }
  begin  { plotter instructions }
  plotopen(calcom,longx,nearside,11.0,7.0,'RANGARAJ') ;
  setscale(1.0) ;
  for l:=1 to np do
    begin { np plots }
    for i:= 1 to nr2 do
      begin
      x[i]:= i ; y[i]:= projection[l,i-nr-1] ;
      end ;
      case l of
      1:move(1.5,0) ;
      2:move(1.5,1.5) ;
      3:move(1.5,3.0) ;
      4:move(1.5,4.5) ;
      5:move(5.5,0) ;
      6:move(5.5,1.5) ;
      7:move(5.5,3.0) ;
      8:move(5.5,4.5) ;
      end ; { case }
    drawinteger(l,3,0.2,0.0) ;
    drawstring(' PROJN ANGLE',-1,0.2,0.0);
    drawreal(prangl[l],7,2,0.2,0.0) ;
      case l of
      1:move(1.5,0.3) ;
      2:move(1.5,1.8) ;
      3:move(1.5,3.3) ;
      4:move(1.5,4.8) ;
      5:move(5.5,0.3) ;
      6:move(5.5,1.8) ;
      7:move(5.5,3.3) ;
      8:move(5.5,4.8) ;
      end ; { case }
    scaledata(x,nr2,4,xinfo) ;
    scaledata(y,nr2,1.0,yinfo) ;
    drawline(x,y,nr2,xinfo,yinfo,lineonly,1) ;
    end ; { np plots }
  writeln(prompt,'THE PLOT IS ',plotclose:5:2,' INCHES LONG.')
  end ; { plotter instructions }
{Q procedures and functions:}
{R procedures and functions:}
{

                              raypixels

}
procedure raypixels(l,k:integer ) ;
{
Parallel ray geometry. Returns no. of pixels covered by ray(l,k) in npr,
their coords in pixel[1,i],pixel[2,i] and the pseudo projection in
pseudo. Raywidth rw, threshold thr, sin and cos of view angle sinl,cosl
to be defined in main program before calling raypixels.
Calls absmal,del,jminmax,min0,max0.
}
```

```
 begin { raypixels }
 npr:=0 ; pseudop:=0 ;
 { get limits of search in y/j direction }
 jj1:=round((-ni*sinl+(k-1)*rw)/cosl) ;
 jj2:=round((-ni*sinl+(k+1)*rw)/cosl) ;
 jj3:=round((ni*sinl+(k-1)*rw)/cosl) ;
 jj4:=round((ni*sinl+(k+1)*rw)/cosl) ;
 jj1:=absmal(jj1,mj) ; jj2:=absmal(jj2,mj) ;
 jj3:=absmal(jj3,mj) ; jj4:=absmal(jj4,mj) ;
 jminmax(jj1,jj2,jj3,jj4,jmin,jmax) ;
 for j:=jmin to jmax do
  begin { row }
  { get limits of search in x/i direction }
  jj1:=round((j*cosl-(k-1)*rw)/sinl) ; jj1:=absmal(jj1,ni-1) ;
  jj2:=round((j*cosl-(k+1)*rw)/sinl) ; jj2:=absmal(jj2,ni-1) ;
  imin:=min0(jj1,jj2)-1 ; imax:=max0(jj1,jj2)+1;
  for i:=imin to imax do
   begin  { column }
   if del(i,j,k,sinl,cosl) then
    begin  { store ray pixels }
    npr:=npr+1 ; pseudop:=pseudop+pic[i,j] ;
    pixel[1,npr]:=i ; pixel[2,npr]:=j ;
    end;  { store ray pixels }
   end ; { column }
  end ; { row }
 end ; { raypixels }
{S procedures and functions:}
{

                  spectra

}
procedure spectra ;
{computes average power spectral envelope of the
original eight projections in array xabs }
 begin { spectra }
 for ll:=1 to 3 do
  begin
  if ll=1 then l:=2 ;
  if ll=2 then l:=4 ;
  if ll=3 then l:=8 ;
  end ;
  begin { 3 projections }
  for k:=1 to nr2 do
  xreal[k]:= projection[l,k-nr-1] ;
  for k:= nr2+1 to nft do xreal[k]:=0 ;
  for k:=1 to nft do ximag[k]:=0 ;
  fft;
  for k:=1 to nft do
  xabs[k]:= xabs[k] + sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
  end ; { 3 projections }
 for k:=1 to nft do
 xabs[k]:= xabs[k]/3 ;
 end ; { spectra }
{
```

```
                               speclip

}
procedure speclip ;
{clips given spectra to spectral average envelope in array xabs}
  begin { speclip }
  dcpower:=sqrt(sqr(xreal[1])+sqr(ximag[1])) ;
  for k:=1 to nft do
    begin { check }
    val:= sqrt(xreal[k]*xreal[k] + ximag[k]*ximag[k]) ;
    val:=val*xabs[1]/(dcpower*xabs[k]) ;
    if val>1 then
      begin { clip }
      xreal[k]:= xreal[k]/val ;
      ximag[k]:= ximag[k]/val ;
      end ; { clip }
    end ; { check }
  end ; { speclip }
{T procedures and functions:}
{U procedures and functions:}
{V procedures and functions:}
{

                             view_average

}
procedure view_average ;
{ produces a picture which is the average of reconstructions
at end of each view }
  begin { view average }
  if l=1 then
    begin { initiate averagepic }
    rewrite(averagepic) ;
    for j:=-mj to mj do
      begin { row }
      for i:=-ni to ni do
      write(averagepic,pic[i,j]) ;
      writeln(averagepic) ;
      end ; { row }
    end  { initiate averagepic }
  else
    begin { add next view picture }
    reset(averagepic) ;
    rewrite(bufferpic) ;
    for j:=-mj to mj do
      begin { row }
      for i:= -ni to ni do
        begin { pixel }
        read(averagepic,val) ;
        val:=val + pic[i,j] ;
        write(bufferpic,val) ;
        end ; { pixel }
      readln(averagepic) ;
      writeln(bufferpic) ;
      end ; { row }
    rewrite(averagepic) ;
  reset(bufferpic) ;
  for j:= -mj to mj do
   begin { swap files }
   for i:=-ni to ni do
    begin { interchange }
    read(bufferpic,val) ;
    write(averagepic,val) ;
    end ; { interchange }
   readln(bufferpic) ;
   writeln(averagepic) ;
   end ; { swap files }
  end ; { add next view picture }
 if l=np then
  begin { transfer picture }
  reset(averagepic) ;
  for j:=-mj to mj do
   begin { put picture in pic }
   for i:=-ni to ni do
    begin { divide }
    read(averagepic,val) ;
    pic[i,j]:=val/np ;
    end ; { divide }
   readln(averagepic) ;
   end ; { put picture in pic }
  end ; { transfer picture }
 end ; { view average }
{W procedures and functions:}
{

                weight

}
procedure weight ;
{weight the spectrum }
 begin { weight }
 for k:=2 to nfth do
  begin { weighting }
  val:= 1 - sqrt(k/nfth) ;
  xreal[k]:= val*xreal[k] ;
  ximag[k]:= val*ximag[k] ;
  end ; { weighting }
 end ; { weight }
{X procedures and functions:}
{

                xerror

}
procedure xerror ;
{computes error of reconstruction comparing projection data}
 begin { error }
 error:=0.0 ;
 for l:= 1 to np do
  begin  { np projections }
  angl:= prangl[l]*pi/180.0 ;
  sinl:=sin(angl) ; cosl:=cos(angl) ;
  rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
  for k:= -nr to nr do
    begin  { ray }
    raypixels(l,k) ;
    error:=error + abs(projection[l,k]-pseudop) ;
    end ;  { ray }
  end ;  { np projections }
 error:=error/(norow*nocol*np) ;
 end ; { error }
{Y procedures and functions:}
{Z procedures and functions:}
{

               Main Block

}
 begin {cartp}
 rewrite(calcom) ;
 reset(inpic) ;
 writeln(prompt,' START TIME ' ,clock:10) ;
 for i:=-ni to ni do
 for j:=-mj to mj do
 pic[i,j]:=0 ;
 for j:=-mj to mj do
   begin { get pic }
   for i:=-ni to ni do
   read(inpic,pic[i,j]) ;
   readln(inpic) ;
   end ; { get pic }
 for j:=-mj to mj do
   begin
   for i:=-ni to ni do
   if pic[i,j]<30 then pic[i,j]:=2 ;
   end ;
 rewrite(mvpl) ;
 formvp(pic,2*mj+1,2*ni+1,8,mvpl) ;
 { define projection angles in degrees }
 for l:=1 to np do prangl[l]:=l*22.5 ;
 { }
 {    .........#########################...............   }
 { }
 { compute projections }
 for l:= 1 to np do
 for k:=-nr to nr do projection[l,k]:= 0 ;
 for l:= 1 to np do
   begin  { np projections }
   angl:= prangl[l]*pi/180.0 ;
   sinl:=sin(angl) ; cosl:=cos(angl) ;
   { ray width rw may be multiplied by an integer to use wider rays ( more
   pixels per ray : modify nprmax ). Perform corrections at all occurrences of
   rw}
   rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
   for k:= -nr to nr do
     begin  { ray }
     raypixels(l,k) ;
     projection[l,k]:= pseudop ;
    end ;  { ray }
  end ;  { np projections }
reset(convfn) ;
for k:=1 to nft do xabs[k]:=0 ;
spectra ;
for l:=1 to np do
  begin { convolution }
  sum1:=0 ; for k:=-nr to nr do sum1:=sum1 + projection[l,k] ;
  for k:=1 to nr2 do xreal[k]:=projection[l,k-nr-1] ;
  for k:=nr2+1 to nft do xreal[k]:=0 ;
  for k:=1 to nft do ximag[k]:=0 ;
  fft ;
  for k:=1 to nft do
    begin { multiply spectra }
    read(convfn,val) ; xreal[k]:=xreal[k]*val ;
    read(convfn,val) ; ximag[k]:=ximag[k]*val ;
    end ; { multiply spectra }
  speclip ;
  for k:=1 to nft do
    begin { prepare for IFT }
    xreal[k]:= xreal[k]/nft ;
    ximag[k]:= -ximag[k]/nft ;
    end ; { prepare for IFT }
  fft ;
  for k:=-nr to nr do
  if projection[l,k]>0 then projection[l,k]:=xreal[k+nr+1] ;
  for k:=-nr to nr do
  if projection[l,k]<0 then projection[l,k]:=0 ;
  sum2:=0 ; for k:=-nr to nr do sum2:= sum2+ projection[l,k] ;
  for k:=-nr to nr do projection[l,k]:= projection[l,k]*sum1/sum2 ;
  end ; { convolution }
writeln(prompt,' AFTER PROJECTIONS ',clock:10) ;
{ }
{   ..............#########################.................   }
{ }
{ Reconstruction by  ART }
iteration:=0 ;error:=0.0 ; preerr:=9999999.9 ;
  repeat  { ART }
  if iteration>0 then preerr:=error ;
  iteration:=iteration+1 ;
  for ll:= 1 to np do
    begin  { projection  }
    l:=ll ;
    angl:=prangl[l]*pi/180.0 ;
    sinl:=sin(angl) ; cosl:=cos(angl) ;
    rw:=amax1(abs(sinl),abs(cosl)) ; thr:=0.5*rw ;
    for k:=-nr to nr do
      begin  { ray }
      raypixels(l,k) ;
      for inpr:=1 to npr do
        begin { correction }
        ip:=pixel[1,inpr] ; jp:=pixel[2,inpr] ;
        if artadd then
          begin { additive ART }
          pic[ip,jp]:=pic[ip,jp]+(projection[l,k]-pseudop)/npr ;
          { if constrained ART, check for negative values }
```

```
     if (artconstr) and (pic[ip,jp]<0) then pic[ip,jp]:= 0 ;
     end ; { additive ART }
    if artmul then
     begin { multiplicative ART }
     if pseudop<>0.0 then
     pic[ip,jp]:= pic[ip,jp]*projection[1,k]/pseudop
     else pic[ip,jp]:=0.0 ;
     end ; { multiplicative ART }
    end ; { correction }
   end ; { ray }
  end;  { projection }
 euclid_error ; xerror ;
 writeln(prompt,' ITERATION ',iteration:3,' PROJECTION ERROR ',error) ;
 writeln(prompt,' EUCLIDEAN ERROR ',eucliderr) ;
 { a suitable criterion on the error may be included here to stop ART }
 until (iteration=5)or(error>0.90*preerr) ;  { ART }
rewrite(mvp3) ;
formvp(pic,2*mj+1,2*ni+1,8,mvp3)  ;
writeln(prompt,' CPU TIME USED ',clock:10,' MILLISECONDS.') ;
end.  { cartp }
```

```
      PROGRAM FFT2D(INPUT,OUTPUT,INPIC,FTREAL,FTIMAG,
     1XFT,YFT,ZFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=INPIC,
     2TAPE1=FTREAL,TAPE2=FTIMAG,TAPE7=XFT,TAPE8=YFT,TAPE9=ZFT)
C23456789
CCCCCCC--------------------------------------------------------------
CCCCCCC THIS PROGRAM COMPUTES 2D FAST FOURIER TRANSFORM OF THE
CCCCCCC  INPUT IMAGE STORED IN FILE INPIC. THE OUTPUT FILES
CCCCCCC   FTREAL AND FTIMAG CONTAIN THE REAL AND IMAGINARY
CCCCCCC    PART OF THE FOURIER TRANSFORM. THE PROGRAM HAS
CCCCCCC     BEEN ADAPTED FROM DIGITAL IMAGE PROCESSING BOOK BY
CCCCCCC      GONZALEZ AND WINTZ.
CCCCCCC--------------------------------------------------------------
CCCCCCC           AUTHOR - ATAM P. DHAWAN
CCCCCCC--------------------------------------------------------------
      DIMENSION ROW(512),INDEX1(512),INDEX2(512),INDEX3(512)
      COMPLEX X(512),Y(512)
      DIMENSION HEADR(10)
      DATA MVPMAX,NFT,MFT/255,256,8/
      NFT2=NFT*2
      CALL OPENMS(7,INDEX1(1),NFT2,0)
      CALL OPENMS(8,INDEX2(1),NFT2,0)
      CALL OPENMS(9,INDEX3(1),NFT2,0)
      READ(3,10) (HEADR(I), I=1,10), NCOL,NROW,NBIT
10    FORMAT(10A1,3I5)
      WRITE(6,10) (HEADR(I), I=1,10), NCOL,NROW
      NBIT=8
      NITN=NFT-NCOL
      NOIN=NITN/2+1
      NJTN=NFT-NROW
      NOJN=NJTN/2+1
      DO 50 J=1,NROW
      JJ=J+NOJN
C     READ ONE ROW OF IMAGE
      READ(3,*) (ROW(I),I=1,NCOL)
      DO 30 I=1,NFT
30    X(I)=(0.0,0.0)
      DO 40 I=1,NCOL
      II=I
      INP=I+NOIN
40    X(INP)= ROW(II)*(1.0,0.0)
C     ROW TRANSFORM
      CALL FFT(X,MFT)
      CALL FOLD(X,NFT)
      CALL WRITMS(7,X,NFT2,JJ)
50    CONTINUE
      DO 80 I=1,NFT
C     READ COLUMNS
      REWIND(7)
      DO 60 J=1,NROW
       JJ=J+NOJN
      CALL READMS(7,Y,NFT2,JJ)
60    X(JJ)=Y(I)
      NROW1=NROW+1
```

```
       NROW1=NROW1+NOJN
       DO 69 J=1,NOJN
       X(J)=(0.0,0.0)
   69 CONTINUE
      DO 70 J=NROW1,NFT
70    X(J)=(0.0,0.0)
C     COLUMN TRANSFORM
      CALL FFT(X,MFT)
      CALL FOLD(X,NFT)
      CALL WRITMS(8,X,NFT2,I)
80    CONTINUE
C      ROTATE TRANSFORM IMAGE
      DO 100 I=1,NFT
      REWIND(8)
      DO 90 J=1,NFT
      CALL READMS(8,Y,NFT2,J)
90    X(J)=Y(I)
      CALL WRITMS(9,X,NFT2,I)
100   CONTINUE
      REWIND(9)
      WRITE(1,105) (HEADR(I),I=1,10),NFT,NFT
      WRITE(2,105) (HEADR(I),I=1,10),NFT,NFT
105   FORMAT(10A1,2I5)
      DO 150 J=1,NFT
      CALL READMS(9,X,NFT2,J)
      DO 110 I=1,NFT
      ROW(I)=REAL(X(I))
110   CONTINUE
      WRITE(1,120) (ROW(!),I=1,NFT)
120   FORMAT(10G13.5)
      DO 130 I=1,NFT
      ROW(I)=AIMAG(X(I))
130   CONTINUE
      WRITE(2,140) (ROW(I),I=1,NFT)
140   FORMAT(10G13.5)
150   CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  HEADR   10616B             UND              REAL            10      NBIT    10637B
  I       10634B                              INTEGER                 NCOL    10635B
  II      10651B                              INTEGER                 NFT     10631B
  INDEX1   1616B                              INTEGER        512      NFT2    10633B
  INDEX2   2616B                              INTEGER        512      NITN    10640B
  INDEX3   3616B                              INTEGER        512      NJTN    10642B
  INP     10652B                              INTEGER                 NOIN    10641B
  J       10644B                              INTEGER                 NOJN    10643B
  JJ      10646B                              INTEGER                 NROW    10636B
  MFT     10632B                              INTEGER                 NROW1   10655B
  MVPMAX  10630B                              INTEGER                 ROW       616B
```

```
             PROGRAM FFT2D        73/74   OPT=0                         FTN 5.0+508          85/01
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  X        4616B                                 COMPLEX       512       Y         6616B


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----       -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL           1      INTRINSIC         READMS                    4       SUBROUTINE
  FFT                      2      SUBROUTINE        REAL       GENERIC        1       INTRINSIC
  FOLD                     2      SUBROUTINE        WRITMS                    4       SUBROUTINE
  OPENMS                   4      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-A

     10    444B     FORMAT          25            70  INACTIVE    DO-TERM        60            110
     30  INACTIVE   DO-TERM         37            80  INACTIVE    DO-TERM        65            120
     40  INACTIVE   DO-TERM         41            90  INACTIVE    DO-TERM        71            130
     50  INACTIVE   DO-TERM         46           100  INACTIVE    DO-TERM        73            140
     60  INACTIVE   DO-TERM         53           105    446B      FORMAT         77            150
     69  INACTIVE   DO-TERM         58


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FFT2D      50B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------       -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ                         TAPE7
  TAPE2   FMT/SEQ                         TAPE8
  TAPE3   FMT/SEQ                         TAPE9
  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH         10665B  =  4533
  CM STORAGE USED             61500B  =  25408
  COMPILE TIME                 1.314 SECONDS
```

```
C
C
    SUBROUTINE FFT(F,LN)
    COMPLEX F(1),U,W,T,CMPLX
    PI=4.0*ATAN(1.0)
    N=2**LN
    NV2=N/2
    NM1=N-1
    J=1
    DO 3 I=1,NM1
    IF(I.GE.J) GO TO 1
    T=F(J)
    F(J)=F(I)
    F(I)=T
1   K=NV2
2   IF(K.GE.J) GO TO 3
    J=J-K
    K=K/2
    GO TO 2
3   J=J+K
    DO 5 L=1,LN
    LE=2**L
    LE1=LE/2
    U=(1.0,0.0)
    W=CMPLX(COS(PI/LE1), -SIN(PI/LE1))
    DO 5 J=1,LE1
    DO 4 I=J,N,LE
    IP=I+LE1
    T=F(IP)*U
    F(IP)=F(I)-T
4   F(I)=F(I)+T
5   U=U*W
    DO 6 I=1,N
6   F(I)=F(I)/FLOAT(N)
    RETURN
    END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  F           1   DUMMY-ARG                  COMPLEX          1     LN          2   DUMMY-ARG
  I        270B                              INTEGER                N        264B
  IP       302B                              INTEGER                NM1      266B
  J        267B                              INTEGER                NV2      265B
  K        272B                              INTEGER                PI       263B
  L        273B                              INTEGER                T        261B
  LE       275B                              INTEGER                U        255B
  LE1      276B                              INTEGER                W        257B
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ATAN      GENERIC        1      INTRINSIC
  CMPLX     GENERIC        2      INTRINSIC
  COS       GENERIC        1      INTRINSIC
  FLOAT     REAL           1      INTRINSIC
  SIN       GENERIC        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

      1     55B                    15             4  INACTIVE   DO-TERM         31
      2     57B                    16             5  INACTIVE   DO-TERM         32
      3     67B     DO-TERM        20             6  INACTIVE   DO-TERM         34


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FFT         5B      2


--STATISTICS--

  PROGRAM-UNIT LENGTH           306B  =   198
  CM STORAGE USED             57500B  =  24384
  COMPILE TIME                 0.704 SECONDS
```

```
      1                     SUBROUTINE FOLD(X,NFT)
      2                     COMPLEX X(1),Z(512)
      3                     NFTH=NFT/2
      4                     DO 800 K=1,NFT
      5              800    Z(K)=X(K)
      6                     DO 810 K=1,NFTH
      7              810    X(K)=Z(K+NFTH)
      8                     NFTH1=NFTH+1
      9                     DO 820 K=NFTH1,NFT
     10              820    X(K)=Z(K-NFTH)
     11                     RETURN
     12                     END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  K        2076B                             INTEGER                NFTH1    2101B
  NFT          2   DUMMY-ARG                 INTEGER                X            1   DUMMY-ARG
  NFTH     2075B                             INTEGER                Z          75B
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    800  INACTIVE   DO-TERM         5
    810  INACTIVE   DO-TERM         7
    820  INACTIVE   DO-TERM        10
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FOLD        5B      2
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH           2105B  =  1093
  CM STORAGE USED              57500B  =  24384
  COMPILE TIME                  0.269 SECONDS
```

```
      PROGRAM SYSTFN(INPUT,OUTPUT,PICR,PICI,RECONR,RECONI,
     1XREAL,XIMAG,MVP1,DAT,XFT,YFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=PICR,TAPE4=PICI,
     2TAPE11=RECONR,TAPE12=RECONI,TAPE1=XREAL,TAPE2=XIMAG,
     3TAPE8=MVP1,TAPE15=DAT,TAPE7=XFT,TAPE13=YFT)
CCCCCCC--------------------------------------------------------------
CCCCCCC   THIS PROGRAM COMPUTES THE SYSTEM TRANSFER FUNCTION.
CCCCCCC    INPUTS FILES ARE PICR AND PICI WHICH ARE REAL AND
CCCCCCC     IMAGINARY PART OF THE 2D FOURIER TRANSFORM OF
CCCCCCC      POINT/BASIS IMAGE, RESPECTIVELY. OTHER PAIR
CCCCCCC       OF INPUT FILES ARE RECONR AND RECONI WHICH
CCCCCCC        ARE REAL AND IMAGINARY PART OF THE 2D FOURIER
CCCCCCC         TRANSFORM OF THE RECONSTRUCTED POINT/BASIS
CCCCCCC          IMAGE, RESPECTIVELY.  THE OUTPUT FILES
CCCCCCC           ARE XREAL, XIMAG, MVP1 AND DAT. XREAL
CCCCCCC            AND XIMAG CONTAIN THE REAL AND IMAGINARY
CCCCCCCC             PART OF THE SYSTEM TRANSFER FUNCTION RESPECTIVELY.
CCCCCCC              MVP1 IS THE MVP IAMGE FILE OF THE AMPLITUDE
CCCCCCC               SPECTRUM OF THE SYSTEM TRANSFER FUNCTION.
CCCCCCC-------------------------------------------------------------------
CCCCCCC              AUTHOR - ATAM P. DHAWAN
CCCCCCC-------------------------------------------------------------------
      DIMENSION ROWR1(512),ROWR2(512),ROWI1(512),ROWI2(512)
      DIMENSION INDEX1(512),HEADR(10)
      DIMENSION INDEX2(512),XXAB(512)
      COMPLEX X(512),Y(512),W(512),Z(512)
      NFT=256
      NHFT=NFT/2+1
      MVPMAX=255
      NFT2=NFT*2
      PI=ATAN(1.0)*4
      CALL OPENMS(7,INDEX1(1),NFT2,0)
      CALL OPENMS(13,INDEX2(1),NFT2,0)
      READ(3,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(4,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(11,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(12,10)  (HEADR(I),I=1,10),NCOL,NROW
   10 FORMAT(10A1,2I5)
      NBIT=8
      WRITE(6,10)  (HEADR(I),I=1,10),NROW,NCOL
      DO 501 I=1,NFT
      ROWR1(I)=0.0
      ROWR2(I)=0.0
      ROWI1(I)=0.0
      ROWI2(I)=0.0
  501 CONTINUE
      DO 30 I=1,NFT
      X(I)=(0.0,0.0)
      Y(I)=(0.0,0.0)
      Z(I)=(0.0,0.0)
      W(I)=(0.0,0.0)
      XXAB(I)=0.0
   30 CONTINUE
      TENG=0.0
      DO 50 J=1,NROW
      READ(3,*)  (ROWR1(I),I=1,NCOL)
      READ(4,*)  (ROWI1(I),I=1,NCOL)
      READ(11,*)  (ROWR2(I),I=1,NCOL)
      READ(12,*)  (ROWI2(I),I=1,NCOL)
      DO 40 I=1,NCOL
      X(I)=CMPLX(ROWR1(I),ROWI1(I))
      Y(I)=CMPLX(ROWR2(I),ROWI2(I))
   40 CONTINUE
      DO 5 I=1,NCOL
      XXAB(I)=CABS(X(I))*CABS(X(I))
      TENG=TENG+XXAB(I)
      IF(J-129)7,8,7
    8 IF(I-129)9,13,9
   13 WRITE(6,14)X(I)
   14 FORMAT("DCVALUE",2F16.5)
      DCVAL=ROWR1(I)
      TENG=TENG-XXAB(I)
    9 CONTINUE
    7 CONTINUE
    5 CONTINUE
       DO 41 I=1,NCOL
      Z(I)=Y(I)/X(I)
   41  CONTINUE
      CALL WRITMS(7,Z,NFT2,J)
      IF(J.EQ.129)GO TO 298
      GO TO 299
  298 DCVALC=CABS(Z(129))
  299 CONTINUE
   50 CONTINUE
      REWIND(7)
      WRITE(1,110)  NFT,NFT
      WRITE(15,130)DCVAL,TENG,DCVALC
      WRITE(2,110)  NFT,NFT
  110 FORMAT("ENDOFHEADR",2I5)
      DO 150 J=1,NFT
      CALL READMS(7,Z,NFT2,J)
      DO 120 I=1,NFT
       ROWR1(I)=REAL(Z(I))
  120 CONTINUE
      WRITE(1,130)  (ROWR1(K),K=1,NFT)
  130 FORMAT(10G13.5)
      DO 140 I=1,NFT
      ROWI1(I)=AIMAG(Z(I))
  140 CONTINUE
      WRITE(2,130)  (ROWI1(K),K=1,NFT)
  150 CONTINUE
      REWIND(7)
      WRITE(6,4)
    4 FORMAT('BEFORE MVP')
      XMAX=0
      XMIN=XMAX
      DO 310 I=1,NFT
      CALL READMS(7,Z,NFT2,I)
      DO 600 K=1,NFT
      W(K)=Z(K)
600   CONTINUE
      CALL WRITMS(13,W,NFT2,I)
      DO 320 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      IF(ROWR1(J).GT.XMAX)XMAX=ROWR1(J)
      IF(ROWR1(J).LT.XMIN)XMIN=ROWR1(J)
 320  CONTINUE
 310  CONTINUE
       WRITE(6,88)XMAX,XMIN
 88    FORMAT(2F12.5)
      WRITE(8,111)
 111  FORMAT('PICTURE FILE FOR MVP')
      I=1
      WRITE(8,222)NFT,NFT,NBIT,I
 222   FORMAT('ENDOFHEADR',4I5)
      SCLF=MVPMAX/(XMAX-XMIN)
      NFT=256
      REWIND(13)
      DO 850 JN=1,NFT
      CALL READMS(13,W,NFT2,JN)
      DO 360 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      INDEX1(J)=0.49+(ROWR1(J)-XMIN)*SCLF
 360  CONTINUE
      WRITE(8,333)  (INDEX1(K),K=1,NFT)
 333  FORMAT(12I5)
850   CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  DCVAL    20161B                            REAL                    NROW    20150B
  DCVALC   20163B                            REAL                    PI      20145B
  HEADR     6127B          UND               REAL             10     ROWI1    3127B
  I        20146B                            INTEGER                 ROWI2    4127B
  INDEX1    5127B                            INTEGER         512     ROWR1    1127B
  INDEX2    6141B                            INTEGER         512     ROWR2    2127B
  J        20155B                            INTEGER                 SCLF    20175B
  JN       20176B                            INTEGER                 TENG    20154B
  K        20166B                            INTEGER                 W       14141B
  MVPMAX   20143B                            INTEGER                 X       10141B
  NBIT     20151B                            INTEGER                 XMAX    20170B
  NCOL     20147B                            INTEGER                 XMIN    20171B
  NFT      20141B                            INTEGER                 XXAB     7141B
  NFT2     20144B                            INTEGER                 Y       12141B
  NHFT     20142B                            INTEGER                 Z       16141B
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----       -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL           1       INTRINSIC         OPENMS                     4       SUBROUTINE
  ALOG10    REAL           1       INTRINSIC         READMS                     4       SUBROUTINE
  ATAN      GENERIC        1       INTRINSIC         REAL      GENERIC          1       INTRINSIC
  CABS      REAL           1       INTRINSIC         WRITMS                     4       SUBROUTINE
  CMPLX     GENERIC        2       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF         -LABEL-A

     4    677B      FORMAT        104            41   INACTIVE    DO-TERM        78           298
     5  INACTIVE    DO-TERM        75            50   INACTIVE    DO-TERM        84           299
     7    313B                     74            88     702B      FORMAT        120           310
     8  INACTIVE                   68           110     672B      FORMAT         89           320
     9    312B                     73           111     704B      FORMAT        122           333
    10    665B      FORMAT         38           120   INACTIVE    DO-TERM        94           360
    13  INACTIVE                   69           130     675B      FORMAT         96           501
    14    667B      FORMAT         70           140   INACTIVE    DO-TERM        99           600
    30  INACTIVE    DO-TERM        53           150   INACTIVE    DO-TERM       101           850
    40  INACTIVE    DO-TERM        63           222     710B      FORMAT        125


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SYSTFN     70B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------       -NAME--- PROPERTIES-------------       -NAME--- PROPERTIES-

  TAPE1   FMT/SEQ                         TAPE15  FMT/SEQ                         TAPE6   FMT/SEQ
  TAPE11  FMT/SEQ                         TAPE2   FMT/SEQ                         TAPE7
  TAPE12  FMT/SEQ                         TAPE3   FMT/SEQ                         TAPE8   FMT/SEQ
  TAPE13                                  TAPE4   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           20201B  =  8321
  CM STORAGE USED               61500B  =  25408
  COMPILE TIME                   1.994 SECONDS
```

```
      PROGRAM FPROJN(INPUT,OUTPUT,PICPR,XREAL,XIMAG,XFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=PICPR,TAPE1=XREAL,
     2TAPE2=XIMAG,TAPE7=XFT)
CCCCCCC-----------------------------------------------------
CCCCCCC  THIS PROGRAM FPROJN CREATES AVERAGE FOURIER SPECTRUM
CCCCCCC    OF THE KNOWN PROJECIONS WHICH ARE STORED IN INPUT
CCCCCCC     FILE PICPR. THE OUTPUT FILES XREAL AND XIMAG
CCCCCCC      CONTAIN REAL AND IMAGINARY PART OF THE
CCCCCCC       AVERAGE FOURIER SPECTRUM  RESPECTIVELY.
CCCCCCC-------------------------------------------------------
CCCCCCC        AUTHOR - ATAM P. DHAWAN
CCCCCCC-------------------------------------------------------
      DIMENSION PICRO(512),PICR1(512),INDEX1(512)
      DIMENSION ROWR(512),ROWI(512)
      COMPLEX X(512),Y(512),Z(512)
      DIMENSION HEADR(10)
      DATA NFT,MVPMAX,MFT/256,255,8/
       NFT2=NFT*2
      READ(3,10)  (HEADR(I),I=1,10),NCOL,NROW,NBIT
   10 FORMAT(10A1,3I5)
      DO 80 I=1,NCOL
      PICRO(I)=0.0
      PICR1(I)=0.0
   80 CONTINUE
      DO 90 I=1,NFT
      Y(I)=(0.0,0.0)
   90 CONTINUE
      DO 100 I=1,NROW
      READ(3,444)  (PICR1(J),J=1,NCOL)
  444 FORMAT(12F5.0)
      DO 140 K=1,NFT
      X(K)=(0.0,0.0)
  140 CONTINUE
      DO 150 JJ=1,NCOL
  150 X(JJ)=PICR1(JJ)*(1.0,0.0)
      CALL FFT(X,MFT)
      CALL FOLD(X,NFT)
      DO 999 NM=1,NFT
      Y(NM)=Y(NM)+X(NM)
  999 CONTINUE
  100 CONTINUE
      DO 143 I=1,NFT
      Y(I)=Y(I)/NROW
CCCCCC IF YOU WANT TO REDUCE WEIGHT OF DC VALUE, DO HERE
C      IF(I.EQ.129)Y(I)=Y(I)/2
CCCCCC FOR EXAMPLE DC VALUE IS REDUCED TO HALF
  143 CONTINUE
      K=1
      WRITE(1,111)  (HEADR(I),I=1,10),NFT,K
      WRITE(2,111)  (HEADR(I),I=1,10),NFT,K
  111 FORMAT(10A1,2I5)
      DO 200 I=1,NFT
      ROWR(I)=REAL(Y(I))
      ROWI(I)=AIMAG(Y(I))
   55           200  CONTINUE
   56                WRITE(1,222)  (ROWR(I),I=1,NFT)
   57                WRITE(2,222)  (ROWI(I),I=1,NFT)
   58           222  FORMAT(10G13.5)
   59                STOP
   60                END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  HEADR    13364B           UND              REAL           10       NFT2     13401B
  I        13402B                            INTEGER                 NM       13416B
  INDEX1    2364B           UND              INTEGER       512       NROW     13404B
  J        13411B           UND              INTEGER                 PICRO      364B
  JJ       13414B                            INTEGER                 PICR1     1364B
  K        13412B                            INTEGER                 ROWI      4364B
  MFT      13400B                            INTEGER                 ROWR      3364B
  MVPMAX   13377B                            INTEGER                 X         5364B
  NBIT     13405B                            INTEGER                 Y         7364B
  NCOL     13403B                            INTEGER                 Z        11364B          U
  NFT      13376B                            INTEGER


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL            1       INTRINSIC
  FFT                       2       SUBROUTINE
  FOLD                      2       SUBROUTINE
  REAL      GENERIC         1       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-A

     10     300B     FORMAT         20          111     304B     FORMAT         51          200
     80  INACTIVE    DO-TERM        24          140  INACTIVE    DO-TERM        33          222
     90  INACTIVE    DO-TERM        27          143  INACTIVE    DO-TERM        47          444
    100  INACTIVE    DO-TERM        41          150  INACTIVE    DO-TERM        35          999


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FPROJN     40B      0
```

```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ
  TAPE2   FMT/SEQ
  TAPE3   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH         13422B  =  5906
  CM STORAGE USED             57500B  =  24384
  COMPILE TIME                 0.874 SECONDS
```

```
      SUBROUTINE FFT(F,LN)
      COMPLEX F(1),U,W,T,CMPLX
      PI=4.0*ATAN(1.0)
      N=2**LN
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I=1,NM1
      IF(I.GE.J) GO TO 1
      T=F(J)
      F(J)=F(I)
      F(I)=T
 1    K=NV2
 2    IF(K.GE.J) GO TO 3
      J=J-K
      K=K/2
      GO TO 2
 3    J=J+K
      DO 5 L=1,LN
      LE=2**L
      LE1=LE/2
      U=(1.0,0.0)
      W=CMPLX(COS(PI/LE1), -SIN(PI/LE1))
      DO 5 J=1,LE1
      DO 4 I=J,N,LE
      IP=I+LE1
      T=F(IP)*U
      F(IP)=F(I)-T
 4    F(I)=F(I)+T
 5    U=U*W
      DO 6 I=1,N
 6    F(I)=F(I)/FLOAT(N)
      RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  F           1   DUMMY-ARG                   COMPLEX         1      LN          2   DUMMY-ARG
  I        270B                               INTEGER                N        264B
  IP       302B                               INTEGER                NM1      266B
  J        267B                               INTEGER                NV2      265B
  K        272B                               INTEGER                PI       263B
  L        273B                               INTEGER                T        261B
  LE       275B                               INTEGER                U        255B
  LE1      276B                               INTEGER                W        257B
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ATAN      GENERIC        1      INTRINSIC
  CMPLX     GENERIC        2      INTRINSIC
  COS       GENERIC        1      INTRINSIC
  FLOAT     REAL           1      INTRINSIC
  SIN       GENERIC        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

      1      55B                    13            4  INACTIVE    DO-TERM        29
      2      57B                    14            5  INACTIVE    DO-TERM        30
      3      67B     DO-TERM        18            6  INACTIVE    DO-TERM        32


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FFT          5B     2


--STATISTICS--

  PROGRAM-UNIT LENGTH           306B  =  198
  CM STORAGE USED             57500B  =  24384
  COMPILE TIME                 0.691 SECONDS
```

```
      1                       SUBROUTINE FOLD(X,NFT)
      2                       COMPLEX X(1),Z(512)
      3                       NFTH=NFT/2
      4                       DO 800 K=1,NFT
      5                 800   Z(K)=X(K)
      6                       DO 810 K=1,NFTH
      7                 810   X(K)=Z(K+NFTH)
      8                       NFTH1=NFTH+1
      9                       DO 820 K=NFTH1,NFT
     10                 820   X(K)=Z(K-NFTH)
     11                       RETURN
     12                       END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  K         2076B                             INTEGER                NFTH1     2101B
  NFT          2   DUMMY-ARG                  INTEGER                X            1   DUMMY-ARG
  NFTH      2075B                             INTEGER                Z          75B


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    800   INACTIVE   DO-TERM        5
    810   INACTIVE   DO-TERM        7
    820   INACTIVE   DO-TERM       10


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FOLD        5B     2


--STATISTICS--

  PROGRAM-UNIT LENGTH          2105B  =  1093
  CM STORAGE USED             57500B  =  24384
  COMPILE TIME                 0.265 SECONDS
```

```
      PROGRAM AVROT(INPUT,OUTPUT,INPROJR,INPROJI,MVP1,XREAL,
     1XIMAG,PRJABS,XFT,YFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=INPROJR,TAPE2=INPROJI,
     2TAPE3=MVP1,TAPE8=XREAL,TAPE9=XIMAG,TAPE4=PRJABS,TAPE7=XFT,
     3TAPE15=YFT)
C-----------------------------------------------------------------
C          THIS PROGRAM COMPUTES 2D RADIALLY SYMMETRIC SPECTRUM
C             OF ONE-DIMENSIONAL AVERAGE SPECTRUM OF KNOWN
C                     PROJECTIONS.
C           INPUT FILES ARE INPROJR AND INPROJI WHICH ARE OUTPUTS
C             OF PROGRAM FPROJN, AND REAL AND IMAGINARY PART
C                OF THE 1D AVERAGE SPECTRUM OF PROJECTIONS,
C                  RESPECTIVELY.  OUTPUT FILES ARE XREAL AND
C                   XIMAG WHICH ARE REAL AND IMAGINARY PART
C                    OF THE 2D RADIALLY SYMMETRIC AVERAGE
C                      SPECTRU, RESPECTIVELY.
C--------------------------------------------------------------------
C                    AUTHOR - ATAM P. DHAWAN
C====================================================================
      DIMENSION X(512),Y(512),INDEX(512),HEADR(10),INDEX2(512)
      DIMENSION INDEX1(512),ROWR(512),ROWI(512)
      COMPLEX PROJN(512),Z(512),DCVAL,DCVALC
      NFT=256
      NFT2=NFT*2
      WRITE(6,2)
  2   FORMAT('START OF PROGRAM')
      MVPMAX=255
      NBIT=8
      NHFT=NFT/2+1
      CALL OPENMS(7,INDEX1(1),NFT2,0)
      CALL OPENMS(15,INDEX2(1),NFT2,0)
      READ(1,10)  (HEADR(I), I=1,10), NCOL,NROW
      READ(2,10)  (HEADR(I), I=1,10), NCOL,NROW
  10  FORMAT(10A1,2I5)
      DCVALC=(0.0,0.0)
      DCVAL=(0.0,0.0)
      READ(1,*)  (ROWR(I),I=1,NCOL)
      READ(2,*)  (ROWI(I),I=1,NCOL)
       DO 17 I=1,NFT
       PROJN(I)=(0.0,0.0)
  17   CONTINUE
      DO 23 I=1,NCOL
      PROJN(I)=CMPLX(ROWR(I),ROWI(I))
  23  CONTINUE
C     WRITE(6,444)  (PROJN(I),I=1,NFT)
      DO 60 I=1,NFT
      X(I)=0.0
      Y(I)=0.0
      Z(I)=(0.0,0.0)
  60  CONTINUE
      DO 70 I=1,NFT
      X(I)=CABS(PROJN(I))
  70  CONTINUE
      WRITE(4,444)  (X(I),I=1,NFT)
      DO 80 I=1,NFT
      X(I)=0.0
   80 CONTINUE
      DO 100 I=1,NFT
      DO 150 J=1,NFT
      X(J)=((NHFT-I)*(NHFT-I)+(NHFT-J)*(NHFT-J))**0.5
      IF(I.LT.129)GO TO 130
      IF(J.LT.129)GO TO 134
      Y(J)=X(J)+NHFT
      GO TO 140
  134  Y(J)=NHFT-X(J)
       GO TO 140
  130 Y(J)=NHFT-X(J)
 140  IF(Y(J).LT.0.)Y(J)=0.0
      N1=Y(J)
       NN1=N1+1
      N2=Y(J)*1000.0
      N3=N2-N1*1000
      IF(NN1.GT.256)PROJN(N1)=(0.0,0.0)
      IF(NN1.GT.256)PROJN(NN1)=(0.0,0.0)
      Z(J)=PROJN(N1)+(PROJN(NN1)-PROJN(N1))*N3/1000
  150 CONTINUE
C     WRITE(6,8)I,J,NFT
  8   FORMAT(3I5)
      CALL WRITMS(7,Z,NFT2,I)
      IF(I.EQ.129)GO TO 701
      GO TO 702
 701  DO 806 K=1,NFT
      DCVAL=Z(K)
      IF(K.EQ.129)DCVALC=DCVAL
 806  CONTINUE
  702 CONTINUE
  100 CONTINUE
      WRITE(6,704)DCVALC
 704  FORMAT("DCCVAL",2F14.6)
      REWIND(7)
      DO 705 J=1,NFT
      CALL READMS(7,Z,NFT2,J)
      DO 706 I=1,NFT
      Z(I)=Z(I)/DCVALC
 706  CONTINUE
      CALL WRITMS(15,Z,NFT2,J)
 705  CONTINUE
      REWIND(15)
      WRITE(8,77)  (HEADR(I),I=1,10),NFT,NFT
      WRITE(9,77)  (HEADR(I),I=1,10),NFT,NFT
  77  FORMAT(10A1,2I5)
      DO 400 J=1,NFT
      CALL READMS(15,Z,NFT2,J)
      DO 410 I=1,NFT
      ROWR(I)=REAL(Z(I))
 410  CONTINUE
      WRITE(8,444)  (ROWR(I),I=1,NFT)
 444  FORMAT(10G13.5)
```

```
      DO 420 I=1,NFT
      ROWI(I)=AIMAG(Z(I))
 420  CONTINUE
      WRITE(9,444)  (ROWI(I),I=1,NFT)
 400  CONTINUE
      REWIND(15)
      WRITE(6,4)
   4  FORMAT('BEFORE MVP')
      XMAX=0
      XMIN=XMAX
      DO 310 I=1,NFT
      CALL READMS(15,Z,NFT2,I)
      DO 320 J=1,NFT
      Z(J)=Z(J)*DCVALC
      X(J)=ALOG10(CABS(Z(J))+1)
      IF(X(J).GT.XMAX)XMAX=X(J)
      IF(X(J).LT.XMIN)XMIN=X(J)
  320 CONTINUE
  310 CONTINUE
       WRITE(6,88)XMAX,XMIN
  88   FORMAT(2F12.5)
      REWIND(15)
      WRITE(3,111)
  111 FORMAT('PICTURE FILE FOR MVP')
      I=1
      WRITE(3,222)NFT,NFT,NBIT,I
  222  FORMAT('ENDOFHEADR',4I5)
      SCLF=MVPMAX/(XMAX-XMIN)
      DO 350 I=1,NFT
      CALL READMS(15,Z,NFT2,I)
      DO 360 J=1,NFT
      Z(J)=Z(J)*DCVALC
      X(J)=ALOG10(CABS(Z(J))+1)
      INDEX(J)=0.49+(X(J)-XMIN)*SCLF
  360 CONTINUE
      WRITE(3,333)  (INDEX(K),K=1,NFT)
  333 FORMAT(12I5)
  350  CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  DCVAL    14227B                              COMPLEX                   MVPMAX   14235B
  DCVALC   14231B                              COMPLEX                   NBIT     14236B
  HEADR     4215B             UND              REAL             10       NCOL     14241B
  I        14240B                              INTEGER                   NFT      14233B
  INDEX     3215B                              INTEGER         512       NFT2     14234B
  INDEX1    5227B                              INTEGER         512       NHFT     14237B
  INDEX2    4227B                              INTEGER         512       NN1      14254B
  J        14251B                              INTEGER                   NROW     14242B
  K        14257B                              INTEGER                   N1       14253B
```

```
           PROGRAM AVROT       73/74    OPT=0                         FTN 5.0+508          85/01
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  N2       14255B                            INTEGER                 X         1215B
  N3       14256B                            INTEGER                 XMAX     14266B
  PROJN    10227B                            COMPLEX         512     XMIN     14267B
  ROWI      7227B                            REAL            512     Y         2215B
  ROWR      6227B                            REAL            512     Z        12227B
  SCLF     14272B                            REAL


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----       -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL             1     INTRINSIC         OPENMS                    4      SUBROUTINE
  ALOG10    REAL             1     INTRINSIC         READMS                    4      SUBROUTINE
  CABS      REAL             1     INTRINSIC         REAL       GENERIC        1      INTRINSIC
  CMPLX     GENERIC          2     INTRINSIC         WRITMS                    4      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF         -LABEL-A

      2     775B     FORMAT          26          100  INACTIVE   DO-TERM        87           360
      4    1014B     FORMAT         116          111   1021B     FORMAT        132           400
      8    1003B     FORMAT          78          130    277B                    67           410
     10    1001B     FORMAT          34          134    272B                    65           420
     17  INACTIVE    DO-TERM         41          140    303B                    68           444
     23  INACTIVE    DO-TERM         44          150  INACTIVE   DO-TERM        76           701
     60  INACTIVE    DO-TERM         50          222   1025B     FORMAT        135           702
     70  INACTIVE    DO-TERM         53          310  INACTIVE   DO-TERM       127           704
     77    1010B     FORMAT         101          320  INACTIVE   DO-TERM       126           705
     80  INACTIVE    DO-TERM         57          333   1030B     FORMAT        145           706
     88    1017B     FORMAT         129          350  INACTIVE   DO-TERM       146           806


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  AVROT       60B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------        -NAME--- PROPERTIES------------        -NAME--- PROPERTIES-

  TAPE1   FMT/SEQ                         TAPE3   FMT/SEQ                        TAPE7
  TAPE15                                  TAPE4   FMT/SEQ                        TAPE8   FMT/SEQ
  TAPE2   FMT/SEQ                         TAPE6   FMT/SEQ                        TAPE9   FMT/SEQ
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH          14275B  =  6333
  CM STORAGE USED              61500B  =  25408
  COMPILE TIME                  2.228 SECONDS
```

```
C---------------------------------------------------------------
      PROGRAM FILIN(INPUT,OUTPUT,RECONSR,RECONSI,PRSPR,PRSPI,
     1XREAL,XIMAG,MVP1,DAT,XFT,YFT,ZFT,
     2TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=RECONSR,TAPE2=RECONSI,
     3TAPE3=PRSPR,TAPE4=PRSPI,TAPE7=XREAL,TAPE8=XIMAG,
     4TAPE9=MVP1,TAPE10=XFT,TAPE13=YFT,TAPE14=ZFT,TAPE15=DAT)
C-----------------------------------------------------------------
C       THIS PROGRAM COMPUTES FILLED-IN (INTERPOLATED) FOURIER
C        SPECTRUM OF THE LIMITED-VIEW RECONSTRUCTION WITH THE
C       AVERAGE SPECTRUM OF THE KNOWN PROJECTIONS. THE INPUT
C        FILES ARE RECONSR, RECONSI, PRSPR AND PRSPI. RECONSR
C       AND RECONSI ARE REAL AND IMAGINARY PART OF THE 2D FOURIER
C        TRANSFORM OF THE LIMITED-VIEW RECONSTRUCTION, AND PRSPR
C       AND PRSPI ARE THE REAL AND IMAGINARY PART OF THE 2D AVERAGE
C        FOURIER SPECTRUM RESPECTIVELY.
C         THE OUTPUT FILES XREAL AND XIMAG CONTAIN THE REAL AND
C        IMAGINARY PART OF THE FILLED-IN SPECTRUM. MVP1 IS THE
C         MVP IMAGE FILE OF THE FILLED-IN SPECTRUM.
C------------------------------------------------------------------
C
C                    AUTHOR - ATAM P. DHAWAN
C
C===================================================================
      DIMENSION INDEX1(512),HEADR(10),ROWR1(512),INDEX2(512)
      DIMENSION ROWI1(512),ROWR2(256),ROWI2(256)
      DIMENSION XXAB(512),INDEX3(512)
      COMPLEX X(512),W(512),Y(512)
       NFT=256
      NBIT=8
      MVPMAX=255
      NHFT=NFT/2+1
      NFT2=NFT*2
      WRITE(6,2)
  2   FORMAT("START OF PROGRAM")
      CALL OPENMS(10,INDEX1(1),NFT2,0)
      CALL OPENMS(13,INDEX2(1),NFT2,0)
      CALL OPENMS(14,INDEX3(1),NFT2,0)
      READ(1,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(2,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(3,10)  (HEADR(I),I=1,10),NCOL,NROW
      READ(4,10)  (HEADR(I),I=1,10),NCOL,NROW
  10  FORMAT(10A1,2I5)
      WRITE(6,10)  (HEADR(I),I=1,10),NCOL,NROW
      DO 100 I=1,NFT
      ROWR1(I)=0.0
      ROWI1(I)=0.0
      ROWR2(I)=0.0
      ROWI2(I)=0.0
  100 CONTINUE
      DO 101 I=1,NFT
      X(I)=(0.0,0.0)
      XXAB(I)=0.0
      Y(I)=(0.0,0.0)
      W(I)=(0.0,0.0)
  101 CONTINUE
      NCOUNT=0
      READ(15,*)DCVALP,TENGP,DCVALM
      DO 110 J=1,NROW
      READ(1,*)  (ROWR1(I),I=1,NCOL)
      READ(2,*)  (ROWI1(I),I=1,NCOL)
      READ(3,*)  (ROWR2(I),I=1,NCOL)
      READ(4,*)  (ROWI2(I),I=1,NCOL)
      DO 120 I=1,NCOL
      X(I)=CMPLX(ROWR1(I),ROWI1(I))/DCVALP
C     W(I)=CMPLX(ROWR2(I),ROWI2(I))/DCVALP
      W(I)=CMPLX(ROWR2(I),ROWI2(I))
 120  CONTINUE
      DO 130 K=1,NCOL
      X1ABS=CABS(X(K))
      X2ABS=CABS(W(K))
      IF(X1ABS-X2ABS)505,506,506
 505  X(K)=X(K)*X2ABS/X1ABS
C505   X(K)=W(K)
C     X(K)=X(K)*2.25
      NCOUNT=NCOUNT+1
C     WRITE(6,14)X1ABS,X2ABS
      GO TO 507
 506  CONTINUE
 507  CONTINUE
  130  CONTINUE
      DO 367 I=1,NFT
      X(I)=X(I)*DCVALP
  367 CONTINUE
      CALL WRITMS(10,X,NFT2,J)
  110 CONTINUE
      REWIND(10)
      TENGC=0.0
      DCVALC=0.0
      DO 901 J=1,NFT
      CALL READMS(10,X,NFT2,J)
      DO 5 I=1,NCOL
      XXAB(I)=CABS(X(I))*CABS(X(I))
      TENGC=TENGC+XXAB(I)
      IF(J-129)7,8,7
  8   IF(I-129)9,13,9
  13  WRITE(6,14)X(I)
  14  FORMAT("DCVALUE",2F16.5)
      DCVALC=REAL(X(I))
      TENGC=TENGC-XXAB(I)
   9  CONTINUE
   7  CONTINUE
   5  CONTINUE
  901  CONTINUE
      WRITE(6,909)TENGP,TENGC
 909  FORMAT("TOTALENERGYPICCON",2F16.5)
      REWIND(10)
      DO 902 J=1,NFT
      CALL READMS(10,X,NFT2,J)
      DO 903 I=1,NFT
      Y(I)=X(I)
  903 CONTINUE
      CALL WRITMS(13,Y,NFT2,J)
  902 CONTINUE
      WRITE(7,111)NFT,NFT
      WRITE(8,111)NFT,NFT
  111 FORMAT("ENDOFHEADR",2I5)
      REWIND(13)
      DO 150 J=1,NFT
      CALL READMS(13,X,NFT2,J)
      DO 160 I=1,NFT
      ROWR1(I)=REAL(X(I))
      ROWI1(I)=AIMAG(X(I))
  160 CONTINUE
      WRITE(7,222)  (ROWR1(K),K=1,NFT)
      WRITE(8,222)  (ROWI1(K),K=1,NFT)
  222 FORMAT(10G13.5)
  150 CONTINUE
      REWIND(13)
      WRITE(6,4)
    4 FORMAT('BEFORE MVP')
      WRITE(6,666)NCOUNT
  666 FORMAT("NUMBERCOUNT",I7)
      XMAX=0
      XMIN=XMAX
      DO 310 I=1,NFT
      CALL READMS(13,X,NFT2,I)
      DO 600 K=1,NFT
      W(K)=X(K)
 600  CONTINUE
      CALL WRITMS(14,W,NFT2,I)
      DO 320 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      IF(ROWR1(J).GT.XMAX)XMAX=ROWR1(J)
      IF(ROWR1(J).LT.XMIN)XMIN=ROWR1(J)
  320 CONTINUE
  310 CONTINUE
       WRITE(6,88)XMAX,XMIN
  88   FORMAT(2F12.5)
      WRITE(9,333)
  333 FORMAT('PICTURE FILE FOR MVP')
      I=1
      WRITE(9,444)NFT,NFT,NBIT,I
  444  FORMAT('ENDOFHEADR',4I5)
      SCLF=MVPMAX/(XMAX-XMIN)
      REWIND(14)
      DO 850 JN=1,NFT
      CALL READMS(14,W,NFT2,JN)
      DO 360 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      INDEX1(J)=0.49+(ROWR1(J)-XMIN)*SCLF
  360 CONTINUE
      WRITE(9,555)  (INDEX1(K),K=1,NFT)
    163             555 FORMAT(1215)
    164             850 CONTINUE
    165                 STOP
    166                 END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  DCVALC   16340B                            REAL                     NHFT     16314B
  DCVALM   16326B                            REAL                     NROW     16320B
  DCVALP   16324B                            REAL                     ROWI1     4311B
  HEADR     2277B          UND               REAL             10      ROWI2     5711B
  I        16316B                            INTEGER                  ROWR1     2311B
  INDEX1    1277B                            INTEGER         512      ROWR2     5311B
  INDEX2    3311B                            INTEGER         512      SCLF     16354B
  INDEX3    7311B                            INTEGER         512      TENGC    16337B
  J        16327B                            INTEGER                  TENGP    16325B
  JN       16355B                            INTEGER                  W        12311B
  K        16332B                            INTEGER                  X        10311B
  MVPMAX   16313B                            INTEGER                  XMAX     16347B
  NBIT     16312B                            INTEGER                  XMIN     16350B
  NCOL     16317B                            INTEGER                  XXAB      6311B
  NCOUNT   16323B                            INTEGER                  X1ABS    16334B
  NFT      16311B                            INTEGER                  X2ABS    16335B
  NFT2     16315B                            INTEGER                  Y        14311B


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL           1      INTRINSIC         OPENMS                    4      SUBROUTINE
  ALOG10    REAL           1      INTRINSIC         READMS                    4      SUBROUTINE
  CABS      REAL           1      INTRINSIC         REAL      GENERIC         1      INTRINSIC
  CMPLX     GENERIC        2      INTRINSIC         WRITMS                    4      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF         -LABEL-A

       2     765B     FORMAT         34         110  INACTIVE    DO-TERM       85           444
       4    1007B     FORMAT        130         111   1002B      FORMAT       116           505
       5  INACTIVE    DO-TERM       102         120  INACTIVE    DO-TERM       67           506
       7     430B                   101         130  INACTIVE    DO-TERM       80           507
       8  INACTIVE                   95         150  INACTIVE    DO-TERM      127           555
       9     427B                   100         160  INACTIVE    DO-TERM      123           600
      10     771B     FORMAT         42         222   1005B      FORMAT       126           666
      13  INACTIVE                   96         310  INACTIVE    DO-TERM      146           850
      14     773B     FORMAT         97         320  INACTIVE    DO-TERM      145           901
      88    1015B     FORMAT        148         333   1017B      FORMAT       150           902
     100  INACTIVE    DO-TERM        49         360  INACTIVE    DO-TERM      161           903
     101  INACTIVE    DO-TERM        55         367  INACTIVE    DO-TERM       83           909
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FILIN      74B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------      -NAME--- PROPERTIES-------------      -NAME--- PROPERTIES-

  TAPE1   FMT/SEQ                        TAPE15  FMT/SEQ                        TAPE6   FMT/SEQ
  TAPE10                                 TAPE2   FMT/SEQ                        TAPE7   FMT/SEQ
  TAPE13                                 TAPE3   FMT/SEQ                        TAPE8   FMT/SEQ
  TAPE14                                 TAPE4   FMT/SEQ                        TAPE9   FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH         16360B  =  7408
  CM STORAGE USED             61500B  =  25408
  COMPILE TIME                 2.333 SECONDS
```

```
      PROGRAM WTFR(INPUT,OUTPUT,INPROJR,INPROJI,MVP1,XREAL,
     1XIMAG,PRJABS,XFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=INPROJR,TAPE2=INPROJI,
     2TAPE3=MVP1,TAPE8=XREAL,TAPE9=XIMAG,TAPE4=PRJABS,TAPE7=XFT)
C====================================================================
C             THIS PROGRAM COMPUTES 2D NOISE-TO-SIGNAL-RATIO FUNCTION
C               FROM FILLED-IN SIGNAL SPECTRUM AND WHITE NOISE
C             ASSUMPTION. INPROJR AND INPROJI ARE REAL AND IMAGINARY
C               PART OF THE FILLED-IN SIGNAL SPECTRUM (OUTPUT OF
C             FILIN PROGRAM). VALUES OF PERCENTAGE OF VARIANCE SHOULD BE
C               PUT IN THE PROGRAM. REAL AND IMAGINARY PART OF THE
C             2D NOISE-TO-SIGNAL-RATIO FUNCTION ARE STORED IN
C               OUTPUT FILES XREAL AND XIMAG RESPECTIVELY.
C               MVP1 IS THE MVP IMAGE FILE OF THE AMPLITUDE SPECTRUM
C                      OF 2D NOISE-TO-SIGNAL RATIO FUNCTION.
C====================================================================
C
C                   AUTHOR: ATAM P. DHAWAN
C
C======================================================================
      DIMENSION X(512),Y(512),INDEX(512),HEADR(10)
      DIMENSION INDEX1(512),ROWR(512),ROWI(512)
      COMPLEX PROJN(512),Z(512),APROJ(512)
      NFT=256
      NFT2=NFT*2
      WRITE(6,2)
  2   FORMAT('START OF PROGRAM')
      MVPMAX=255
      NBIT=8
      NHFT=NFT/2+1
      CALL OPENMS(7,INDEX1(1),NFT2,0)
      READ(1,10) (HEADR(I), I=1,10), NCOL,NROW
      READ(2,10) (HEADR(I), I=1,10), NCOL,NROW
  10  FORMAT(10A1,2I5)
      DO 555 J=1,NFT
      READ(1,*) (ROWR(I),I=1,NCOL)
      READ(2,*) (ROWI(I),I=1,NCOL)
       DO 17 I=1,NFT
       Z(I)=(0.0,0.0)
       APROJ(I)=(0.0,0.0)
  17   CONTINUE
      DO 23 I=1,NCOL
      APROJ(I)=CMPLX(ROWR(I),ROWI(I))
  23  CONTINUE
      PERCN=0.0000005
      VARPIC=604.0
      ETAS=PERCN*VARPIC/256.0
      DO 45 I=1,NCOL
      Z(I)=ETAS/(APROJ(I)*APROJ(I))
  45  CONTINUE
C     WRITE(6,8)I,J,NFT
  8   FORMAT(3I5)
      CALL WRITMS(7,Z,NFT2,J)
 555  CONTINUE
```

```
      REWIND(7)
      WRITE(8,77)  (HEADR(I),I=1,10),NFT,NFT
      WRITE(9,77)  (HEADR(I),I=1,10),NFT,NFT
   77 FORMAT(10A1,2I5)
      DO 400 J=1,NFT
      CALL READMS(7,Z,NFT2,J)
      DO 410 I=1,NFT
      ROWR(I)=REAL(Z(I))
  410 CONTINUE
      WRITE(8,444)  (ROWR(I),I=1,NFT)
  444 FORMAT(10G13.5)
      DO 420 I=1,NFT
      ROWI(I)=AIMAG(Z(I))
  420 CONTINUE
      WRITE(9,444)  (ROWI(I),I=1,NFT)
  400 CONTINUE
      REWIND(7)
      WRITE(6,4)
    4 FORMAT('BEFORE MVP')
      XMAX=0
      XMIN=XMAX
      DO 310 I=1,NFT
      CALL READMS(7,Z,NFT2,I)
      DO 320 J=1,NFT
      X(J)=ALOG10(CABS(Z(J))+1)
      IF(X(J).GT.XMAX)XMAX=X(J)
      IF(X(J).LT.XMIN)XMIN=X(J)
  320 CONTINUE
  310 CONTINUE
       WRITE(6,88)XMAX,XMIN
   88  FORMAT(2F12.5)
      REWIND(7)
      WRITE(3,111)
  111 FORMAT('PICTURE FILE FOR MVP')
      I=1
      WRITE(3,222)NFT,NFT,NBIT,I
  222  FORMAT('ENDOFHEADR',4I5)
      SCLF=MVPMAX/(XMAX-XMIN)
      DO 350 I=1,NFT
      CALL READMS(7,Z,NFT2,I)
      DO 360 J=1,NFT
      X(J)=ALOG10(CABS(Z(J))+1)
      INDEX(J)=0.49+(X(J)-XMIN)*SCLF
  360 CONTINUE
      WRITE(3,333)  (INDEX(K),K=1,NFT)
  333 FORMAT(12I5)
  350  CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  APROJ    12701B                          COMPLEX        512       NHFT     14705B
  ETAS     14717B                          REAL                     NROW     14710B
  HEADR     3667B           UND            REAL            10       PERCN    14715B
  I        14706B                          INTEGER                  PROJN     6701B             U
  INDEX     2667B                          INTEGER        512       ROWI      5701B
  INDEX1    3701B                          INTEGER        512       ROWR      4701B
  J        14711B                          INTEGER                  SCLF     14730B
  K        14733B           UND            INTEGER                  VARPIC   14716B
  MVPMAX   14703B                          INTEGER                  X          667B
  NBIT     14704B                          INTEGER                  XMAX     14724B
  NCOL     14707B                          INTEGER                  XMIN     14725B
  NFT      14701B                          INTEGER                  Y         1667B             U
  NFT2     14702B                          INTEGER                  Z        10701B


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL            1      INTRINSIC        OPENMS                     4      SUBROUTINE
  ALOG10    REAL            1      INTRINSIC        READMS                     4      SUBROUTINE
  CABS      REAL            1      INTRINSIC        REAL      GENERIC          1      INTRINSIC
  CMPLX     GENERIC         2      INTRINSIC        WRITMS                     4      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-A

        2    502B      FORMAT          27          77     512B      FORMAT          58         350
        4    516B      FORMAT          73          88     521B      FORMAT          85         360
        8    510B      FORMAT          52         111     523B      FORMAT          88         400
       10    506B      FORMAT          34         222     527B      FORMAT          91         410
       17  INACTIVE    DO-TERM         41         310   INACTIVE    DO-TERM         83         420
       23  INACTIVE    DO-TERM         44         320   INACTIVE    DO-TERM         82         444
       45  INACTIVE    DO-TERM         50         333     532B      FORMAT         100         555


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  WTFR        54B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------      -NAME--- PROPERTIES------------

  TAPE1   FMT/SEQ                      TAPE7
  TAPE2   FMT/SEQ                      TAPE8    FMT/SEQ
  TAPE3   FMT/SEQ                      TAPE9    FMT/SEQ
  TAPE6   FMT/SEQ
```

```
--STATISTICS--

  PROGRAM-UNIT LENGTH         14734B  =  6620
  CM STORAGE USED             61500B  =  25408
  COMPILE TIME                 1.517 SECONDS
```

```
C     PROGRAM FOR WIENER FILTER OPERATION BY A.P. DHAWAN
      PROGRAM WIENER(INPUT,OUTPUT,PICR,PICI,CONR,CONI,
     1XREAL,XIMAG,WTREAL,WTIMAG,MVP1,DAT,XFT,YFT,
     1TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=PICR,TAPE4=PICI,
     2TAPE11=CONR,TAPE12=CONI,TAPE1=XREAL,TAPE2=XIMAG,TAPE9=WTREAL,
     3TAPE10=WTIMAG,TAPE8=MVP1,TAPE15=DAT,TAPE7=XFT,TAPE13=YFT)
CCCCCCC----------------------------------------------------------------
CCCCCCC THIS PROGRAM PERFORMS 2D WIENER DECONVOLUTION. INPUT FILES
CCCCCCC   ARE PICR, PICI, WTREAL, WTIMAG, CONR AND CONI.
CCCCCCC PICR AND PICI ARE REAL AND IMAGINARY PART OF FOURIER TRANSFORM
CCCCCCC  OF LIMITED-VIEW RECONSTRUCTION. WTREAL AND WTIMAG ARE REAL AND
CCCCCCC   IMAGINARY PART OF 2D NOISE-TO-SIGNAL RATIO FUNCTION. CONR
CCCCCCC   AND CONI ARE THE REAL AND IMAGINARY PART OF THE SYSTEM
CCCCCCC    TRANSFER FUNCTION.  THE OUTPUT FILES ARE MVP1, XREAL, XIMAG
CCCCCCC       AND DAT. MVP1 IS THE MVP IMAGE ( IN FORMAT 5) OF
CCCCCCC       THE  AMPLITUDE SPECTRUM OF THE DECONVOLVED DATA.  DAT FILE
CCCCCCC         CONTAINS THE STATISTICS ABOUT THE DC VALUE
CCCCCCC            AND TOTAL ENERGY OF THE DECONVOLVED SPECTRUM.
CCCCCCC---------------------------------------------------------------------
CCCCCCC              AUTHOR - ATAM P. DHAWAN
CCCCCCC---------------------------------------------------------------------
      DIMENSION ROWR1(512),ROWR2(512),ROWI1(512),ROWI2(512),V(512)
      DIMENSION INDEX1(512),HEADR(10),ROWR3(512),ROWI3(512),HMAC(512)
      DIMENSION INDEX2(512),XXAB(512),S(512)
      COMPLEX X(512),Y(512),Z(512),W(512)
      NFT=256
      NHFT=NFT/2+1
      MVPMAX=255
      NFT2=NFT*2
      PI=ATAN(1.0)*4
      CALL OPENMS(7,INDEX1(1),NFT2,0)
      CALL OPENMS(13,INDEX2(1),NFT2,0)
      READ(3,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(4,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(11,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(12,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(9,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(10,10) (HEADR(I),I=1,10),NCOL,NROW
   10 FORMAT(10A1,2I5)
      NBIT=8
      WRITE(6,10) (HEADR(I),I=1,10),NROW,NCOL
      DO 501 I=1,NFT
      ROWR1(I)=0.0
      ROWR2(I)=0.0
      ROWR3(I)=0.0
      ROWI1(I)=0.0
      ROWI2(I)=0.0
      ROWI3(I)=0.0
  501 CONTINUE
      DO 30 I=1,NFT
      X(I)=(0.0,0.0)
      Y(I)=(0.0,0.0)
      Z(I)=(0.0,0.0)
      XXAB(I)=0.0
      W(I)=(0.0,0.0)
      S(I)=0.0
      V(I)=0.0
      HMAC(I)=0.0
  30  CONTINUE
      TENG=0.0
      DO 50 J=1,NROW
      READ(3,*)  (ROWR1(I),I=1,NCOL)
      READ(4,*)  (ROWI1(I),I=1,NCOL)
      READ(11,*)  (ROWR2(I),I=1,NCOL)
      READ(12,*)  (ROWI2(I),I=1,NCOL)
      READ(9,*)  (ROWR3(I),I=1,NCOL)
      READ(10,*)  (ROWI3(I),I=1,NCOL)
      DO 40 I=1,NCOL
      X(I)=CMPLX(ROWR1(I),ROWI1(I))
      Y(I)=CMPLX(ROWR2(I),ROWI2(I))
       W(I)=CMPLX(ROWR3(I),ROWI3(I))
      S(I)=CABS(W(I))
      HMAC(I)=CABS(Y(I))
   40 CONTINUE
      DO 5 I=1,NCOL
      XXAB(I)=CABS(X(I))*CABS(X(I))
      TENG=TENG+XXAB(I)
      IF(J-129)7,8,7
  8   IF(I-129)9,13,9
   13 WRITE(6,14)X(I)
   14 FORMAT("DCVALUE",2F16.5)
      DCVAL=ROWR1(I)
      TENG=TENG-XXAB(I)
    9 CONTINUE
    7 CONTINUE
    5 CONTINUE
       DO 41 I=1,NCOL
C     VALF=CABS(W(I))
C     IF(VALF.LE.0.001)W(I)=CMPLX(0.001,0.0)
      DIST=((NHFT-I)*(NHFT-I)+(NHFT-J)*(NHFT-J))**0.5
C     IF(DIST.LT.60.0)S(I)=S(I)*0.1
      S(I)=S(I)*0.01
      V(I)=HMAC(I)*HMAC(I)/(HMAC(I)*HMAC(I)+S(I))
C     IF(DIST.LT.30.0)GO TO 198
      Z(I)=X(I)*V(I)/Y(I)
C     Z(I)=Z(I)*1.3830
C     GO TO 199
C 198 Z(I)=(X(I)*V(I))/HMAC(I)
C      IF(DIST.GT.15.0)GO TO 771
C      GO TO 772
C771  IF(DIST.LT.64.0)Z(I)=X(I)
  199 IF(DIST.GT.120.)Z(I)=(0.0,0.0)
  772 CONTINUE
   41  CONTINUE
      CALL WRITMS(7,Z,NFT2,J)
      IF(J.EQ.129)GO TO 298
      GO TO 299
  298 DCVALC=CABS(Z(129))
  299 CONTINUE
   50 CONTINUE
      REWIND(7)
      WRITE(1,110)  NFT,NFT
      WRITE(15,130)DCVAL,TENG,DCVALC
      WRITE(2,110)  NFT,NFT
  110 FORMAT("ENDOFHEADR",2I5)
      DO 150 J=1,NFT
      CALL READMS(7,Z,NFT2,J)
      DO 120 I=1,NFT
       ROWR1(I)=REAL(Z(I))
  120 CONTINUE
      WRITE(1,130)  (ROWR1(K),K=1,NFT)
  130 FORMAT(10G13.5)
      DO 140 I=1,NFT
      ROWI1(I)=AIMAG(Z(I))
  140 CONTINUE
      WRITE(2,130)  (ROWI1(K),K=1,NFT)
  150 CONTINUE
      REWIND(7)
      WRITE(6,4)
    4 FORMAT('BEFORE MVP')
      XMAX=0
      XMIN=XMAX
      DO 310 I=1,NFT
      CALL READMS(7,Z,NFT2,I)
      DO 600 K=1,NFT
      W(K)=Z(K)
  600 CONTINUE
      CALL WRITMS(13,W,NFT2,I)
      DO 320 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      IF(ROWR1(J).GT.XMAX)XMAX=ROWR1(J)
      IF(ROWR1(J).LT.XMIN)XMIN=ROWR1(J)
  320 CONTINUE
  310 CONTINUE
       WRITE(6,88)XMAX,XMIN
   88  FORMAT(2F12.5)
      WRITE(8,111)
  111 FORMAT('PICTURE FILE FOR MVP')
      I=1
      WRITE(8,222)NFT,NFT,NBIT,I
  222  FORMAT('ENDOFHEADR',4I5)
      SCLF=MVPMAX/(XMAX-XMIN)
      NFT=256
      REWIND(13)
      DO 850 JN=1,NFT
      CALL READMS(13,W,NFT2,JN)
      DO 360 J=1,NFT
      ROWR1(J)=ALOG10(CABS(W(J))+1)
      INDEX1(J)=0.49+(ROWR1(J)-XMIN)*SCLF
  360 CONTINUE
      WRITE(8,333)  (INDEX1(K),K=1,NFT)
  333 FORMAT(12I5)
```

```
    163              850  CONTINUE
    164                   STOP
    165                   END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS--BLOCK-----P

  DCVAL    25316B                             REAL                    PI        25302B
  DCVALC   25321B                             REAL                    ROWI1      3264B
  DIST     25320B                             REAL                    ROWI2      4264B
  HEADR     7264B           UND               REAL              10    ROWI3     10276B
  HMAC     11276B                             REAL             512    ROWR1      1264B
  I        25303B                             INTEGER                 ROWR2      2264B
  INDEX1    6264B                             INTEGER          512    ROWR3      7276B
  INDEX2   12276B                             INTEGER          512    S         14276B
  J        25312B                             INTEGER                 SCLF      25333B
  JN       25334B                             INTEGER                 TENG      25311B
  K        25324B                             INTEGER                 V          5264B
  MVPMAX   25300B                             INTEGER                 W         23276B
  NBIT     25306B                             INTEGER                 X         15276B
  NCOL     25304B                             INTEGER                 XMAX      25326B
  NFT      25276B                             INTEGER                 XMIN      25327B
  NFT2     25301B                             INTEGER                 XXAB      13276B
  NHFT     25277B                             INTEGER                 Y         17276B
  NROW     25305B                             INTEGER                 Z         21276B


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----       -NAME------TYPE--------ARGS------CLASS-----

  AIMAG     REAL            1       INTRINSIC        OPENMS                     4       SUBROUTINE
  ALOG10    REAL            1       INTRINSIC        READMS                     4       SUBROUTINE
  ATAN      GENERIC         1       INTRINSIC        REAL      GENERIC          1       INTRINSIC
  CABS      REAL            1       INTRINSIC        WRITMS                     4       SUBROUTINE
  CMPLX     GENERIC         2       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-A

     4    1005B     FORMAT        130           50  INACTIVE    DO-TERM       110           298
     5  INACTIVE    DO-TERM        86           88    1010B     FORMAT        146           299
     7     362B                    85          110    1000B     FORMAT        115           310
     8  INACTIVE                   79          111    1012B     FORMAT        148           320
     9     361B                    84          120  INACTIVE    DO-TERM       120           333
    10     773B     FORMAT         39          130    1003B     FORMAT        122           360
    13  INACTIVE                   80          140  INACTIVE    DO-TERM       125           501
    14     775B     FORMAT         81          150  INACTIVE    DO-TERM       127           600
    30  INACTIVE    DO-TERM        59          199  INACTIVE                  102           772
    40  INACTIVE    DO-TERM        74          222    1016B     FORMAT        151           850
    41  INACTIVE    DO-TERM       104
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  WIENER     100B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------      -NAME--- PROPERTIES-------------      -NAME--- PROPERTIES-

  TAPE1   FMT/SEQ                       TAPE15  FMT/SEQ                       TAPE6   FMT/SEQ
  TAPE10  FMT/SEQ                       TAPE2   FMT/SEQ                       TAPE7
  TAPE11  FMT/SEQ                       TAPE3   FMT/SEQ                       TAPE8   FMT/SEQ
  TAPE12  FMT/SEQ                       TAPE4   FMT/SEQ                       TAPE9   FMT/SEQ
  TAPE13

--STATISTICS--

  PROGRAM-UNIT LENGTH          25337B  =  10975
  CM STORAGE USED              61500B  =  25408
  COMPILE TIME                  2.349 SECONDS
```

```
C PROGRAM CONVERTS RECTANGULAR TO POLAR COORDINATES.
C LOG10( MAG +1 ) IS USED.
CCCCCCC----------------------------------------------------------
CCCCCCC           ATAM P. DHAWAN
CCCCCCC=============================================================
C           INPUT FILES ARE FTREAL AND FTIMAG. REAL AND IMAGINARY
C             PART OF THE RECTANGULAR DATA ARE STORED IN FTREAL
C           AND FTIMAG RESPECTIVELY. OUTPUT IN POLAR FORM IS
C             OBTAINED IN FTMAG (MAGNITUDE) AND FTPHAS (PHASE).
CCCCCCCC==============================================================
C23456789
      PROGRAM TOPOL(FTREAL,FTIMAG,FTMAG,FTPHAS,
     &TAPE1=FTREAL,TAPE2=FTIMAG,TAPE3=FTMAG,TAPE4=FTPHAS)
      DIMENSION ROWR(512),ROWI(512),HEADR(10)
      READ(1,10) (HEADR(I),I=1,10),NCOL,NROW
      READ(2,10) (HEADR(I),I=1,10),NCOL,NROW
10    FORMAT(10A1,2I5)
      WRITE(3,10) (HEADR(I),I=1,10),NCOL,NROW
      WRITE(4,10) (HEADR(I),I=1,10),NCOL,NROW
      DO 40 J=1,NROW
      READ(1,*) (ROWR(I),I=1,NCOL)
      READ(2,*) (ROWI(I),I=1,NCOL)
      DO 20 K=1,NCOL
      XMAG=SQRT(ROWR(K)*ROWR(K) + ROWI(K)*ROWI(K))
      XIMAG=ROWI(K)
      IF(XMAG .NE. 0.)THEN
      ROWI(K)=ATAN2(ROWI(K),ROWR(K))
      ELSE
      ROWI(K)=0
      END IF
      ROWR(K)=ALOG10(XMAG + 1.)
C     ROWR(K)=XMAG
20    CONTINUE
      WRITE(3,30) (ROWR(I),I=1,NCOL)
      WRITE(4,30) (ROWI(I),I=1,NCOL)
30    FORMAT(10G13.5)
40    CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  HEADR    2223B             UND              REAL           10      NROW     2237B
  I        2235B             UND              INTEGER                ROWI     1223B
  J        2240B                              INTEGER                ROWR      223B
  K        2242B                              INTEGER                XIMAG    2245B
  NCOL     2236B                              INTEGER                XMAG     2244B
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ALOG10    REAL           1       INTRINSIC
  ATAN2     GENERIC        2       INTRINSIC
  SQRT      GENERIC        1       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10     141B    FORMAT          17
     20  INACTIVE   DO-TERM         33
     30     143B    FORMAT          36
     40  INACTIVE   DO-TERM         37


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  TOPOL       30B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ
  TAPE2   FMT/SEQ
  TAPE3   FMT/SEQ
  TAPE4   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          2246B  =  1190
  CM STORAGE USED             57500B  =  24384
  COMPILE TIME                 0.518 SECONDS
```

```
C PROGRAM TOMVP CONVERTS A FORMAT-FREE REAL-VALUED FORTRAN PICTURE FILE
C TO MVP FORMAT 1215.
CCCCCCC----------------------------------------------------------------
CCCCCCC             ATAM P. DHAWAN
CCCCCCC=================================================================
      PROGRAM TOMVP(INPIC,OUTPIC,
     &TAPE1=INPIC,TAPE2=OUTPIC)
      DIMENSION ROW(512),IROW(512),HEADR(10)
      REAL MAX,MIN
      DATA MVPMAX,NBIT/255,8/
      MAX=0.
      MIN=1.0E99
      READ(1,10) (HEADR(I),I=1,10),NCOL,NROW
10    FORMAT(10A1,2I5)
      DO 30 J=1,NROW
        READ(1,*) (ROW(I),I=1,NCOL)
        DO 20 I=1,NCOL
          IF(ROW(I).GT.MAX)MAX=ROW(I)
          IF(ROW(I).LT.MIN)MIN=ROW(I)
20      CONTINUE
30    CONTINUE
C---------------------------------------------------------------------------
      REWIND(1)
      READ(1,10) (HEADR(I),I=1,10),NCOL,NROW
      WRITE(2,40)
40    FORMAT('PICTURE FILE FOR MVP')
      WRITE(2,50)NCOL,NROW,NBIT,1
50    FORMAT('ENDOFHEADR',4I5)
      SCALE=MVPMAX/(MAX-MIN)
      DO 80 J=1,NROW
        READ(1,*) (ROW(I),I=1,NCOL)
        DO 60 I=1,NCOL
          IROW(I)=0.49 + (ROW(I)-MIN)*SCALE
60      CONTINUE
        WRITE(2,70) (IROW(I),I=1,NCOL)
70      FORMAT(12I5)
80    CONTINUE
      STOP
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS--BLOCK-----P

  HEADR    2235B             UND              REAL          10      MVPMAX   2251B
  I        2253B                              INTEGER               NBIT     2252B
  IROW     1235B                              INTEGER      512      NCOL     2254B
  J        2256B                              INTEGER               NROW     2255B
  MAX      2247B                              REAL                  ROW       235B              U
  MIN      2250B                              REAL                  SCALE    2261B
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

     10    152B     FORMAT          14           50    160B     FORMAT          28
     20  INACTIVE   DO-TERM         20           60  INACTIVE   DO-TERM         34
     30  INACTIVE   DO-TERM         21           70    163B     FORMAT          36
     40    154B     FORMAT          26           80  INACTIVE   DO-TERM         37


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  TOMVP      20B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE1   FMT/SEQ
  TAPE2   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          2264B  =   1204
  CM STORAGE USED             57500B  =   24384
  COMPILE TIME                 0.556 SECONDS
```

APPENDIX-B

SELECTED PUBLICATIONS

1884–1984 A CENTURY OF ELECTRICAL PROGRESS

IEEE TRANSACTIONS ON

JUNE 1984 VOLUME MI-3 NUMBER 2 (ISSN 0278-0062)

PAPERS

Nevoscopy: Three-Dimensional Computed Tomography of Nevi and Melanomas *In Situ* by Transillumination

ATAM P. DHAWAN, RICHARD GORDON, AND RANGARAJ M. RANGAYYAN

Abstract—The thickness of a malignant nevus has been found to be an important prognostic factor for patients with melanoma. We have designed a new method of imaging nevi that permits their thickness to be measured *in situ*. Using fiber optics directed into the surrounding skin, we transilluminate the nevus. Three images are picked up by a digitizing TV camera: the vertical image (90°), a glancing image (180°), and one at 45°, obtained by using two front-silvered mirrors held next to the nevus in a "nevoscope." The digitized images are used in a computed tomography algorithm to calculate approximate vertical cross sections of the nevus. The algorithm is one we recently developed to permit reconstruction from a very few projections.

Our method is completely noninvasive. It may be used to check all the nevi on a patient. Without excisions, we could establish a baseline three-dimensional shape for each nevus, follow any changes in time, and obtain an early warning of increase in thickness and possible malignancy.

Introduction

THE detection of melanoma in its early, curable stage is now primarily left up to the patient or the general practitioner. The best prognostic factor is the thickness of the suspect nevus ("beauty mark") [1]-[5]. But until now this could not be measured directly without excision and histological sectioning. Consequently, attempts have been made to correlate other signs and symptoms with tumor thickness during the earliest stages of tumor growth. These signs are primarily increase in size and change of color [6]. Bleeding, ulceration, tenderness, and itching occur less frequently [7]. The primary signs are rather subtle to expect a patient to notice, and since they involve what may be small changes in one of a few of what may be a large number of benign nevi, the probability of detection by the unsuspecting patient may be quite low [8]. Moreover, significant fraction of melanomas occur on the back, a difficult part to observe oneself. The general practitioner is in a way in a worse position than the patient, because she or he will generally have even less memory of the previous state of each nevus. Failure to detect melanoma early leaves the patient with a metastasizing cancer that is among the most malignant tumors [1].

We have begun a research program aimed at automatic detection of early melanoma [9]. We hope to replace the current dependence on human memory of nevi and their changes by quantitative recording and interpretation of computerized images. In our first approach [10] we developed a new method of boundary detection that isolates the image of a nevus from that of the surrounding skin. In this paper we will 1) show a new method for looking at nevi by transillumination; 2) demonstrate the prototype of a new device that we call a "nevoscope" that obtains three views of a nevus simultaneously; 3) show how these three views may be combined, using new computed tomography algorithms that we have developed, to produce an approximate three-dimensional reconstruction of the nevus from which we can measure the thickness and width and obtain some details of its internal structure; and 4) show preliminary results comparing our computed tomography reconstructions with histological sections of the same nevus excised from a cadaver after nevoscopy.

Materials and Methods

We built our nevoscope out of a plexiglass cylinder machined to fit around the objective lens of a Wild M8 stereomicroscope. Nylon screws permit it to be moved up and down along the tube of the objective lens for focusing. A plexiglass plate screws into the bottom of the cylinder. This plate (Fig. 1) contains a trapezoidal slot. Two front surface mirrors (9 X 19 X 1.8 mm) were glued onto the slanted parts of the slot at intended angles of 45° and 22.5° from the vertical. The actual angles were measured by reflection of a beam of light and were found to be 45° and 23°, respectively. Light for imaging was provided by a Volpi heat filtered illuminator using Philips 15 V, 150 W bulbs. Two of its three fiber optics bundles were directed in a plane perpendicular to the slot through two holes drilled at 45° through the cylinder and the plate (Fig. 1). These were allowed to protrude slightly (2 mm) so that they would dent the skin and make direct contact with it. (The protruding edges of the mirrors were ground down with a carborundum stone to allow the skin to bulge inwards next to the silvered surfaces.) Some of the superficially backscattered light was blocked off with black tape on the walls of the trapezoidal slot. The transilluminated image, so obtained, was recorded on 35 mm Panatonic-X (Kodak) film and then the film was digitized as described below.

For surface illumination, the third fiber optics bundle was

Manuscript received May 23, 1984.

A. P. Dhawan and R. M. Rangayyan are with the Department of Electrical Engineering, University of Manitoba, Winnipeg, Man., Canada R3T 2N2.

R. Gordon is with the Departments of Pathology, Radiology, Electrical Engineering, and Zoology, University of Manitoba, Winnipeg, Man., Canada R3E OW3.

0278-0062/84/0600-0054$01.00 © 1984 IEEE

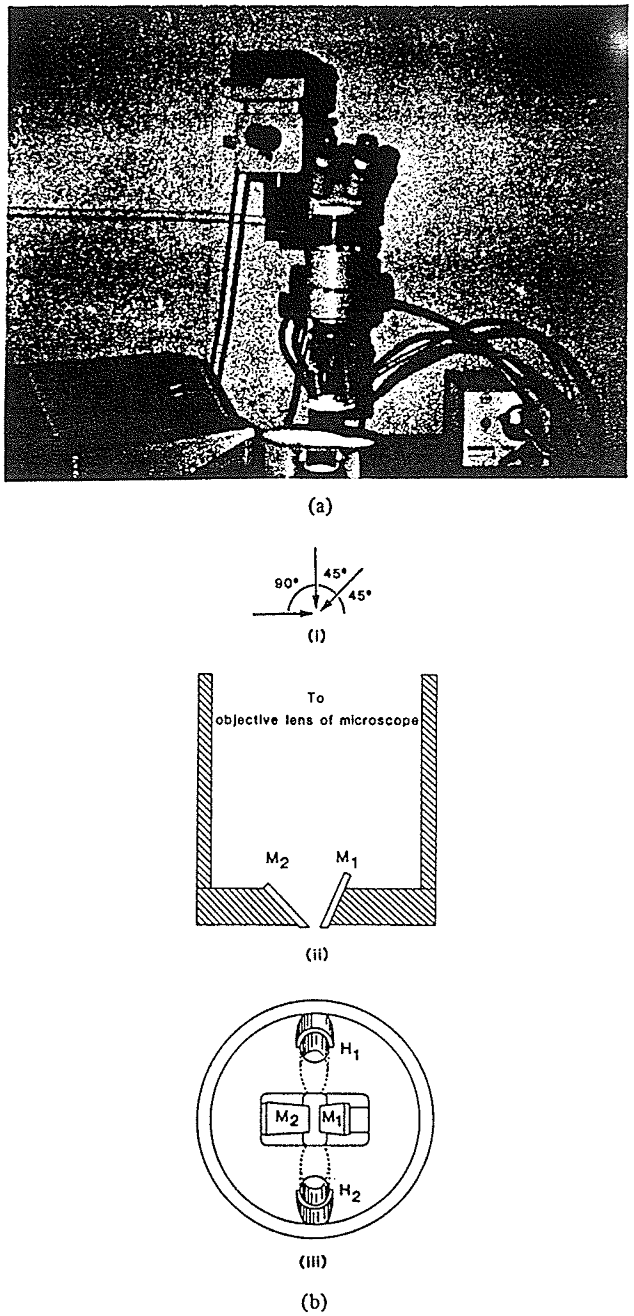


Fig. 1. (a)The prototype nevoscope: side view, as mounted on a stereomicroscope. (b) (i) represents viewing angles of images yielded by nevoscope, (ii) represents sectional view, and (iii) represents mirror bearing plate of the nevoscope. M_1 and M_2 are the front silvered mirrors. H_1 and H_2 are holes for fiber optics bundles to transilluminate the skin.

directed at the nevus through the transparent cylinder. The transillumination was not blocked.

Computed tomograms of nevi and simulated nevi were displayed on a Conrac color television monitor via a 512 X 480 pixel Grinnell frame buffer with 10 bit planes connected to a Cyber computer via a CAMAC crate. Image enhancement, thresholding, and boundary tracing were carried out using our Manitoba Video Processor (MVP) software. Digitization was accomplished at 6 bits density resolution with averaging of 8 consecutive video frames by the frame buffer. The video image was acquired with a Sony video camera using a Nikon 55 mm macro lens on a C-mount adapter.

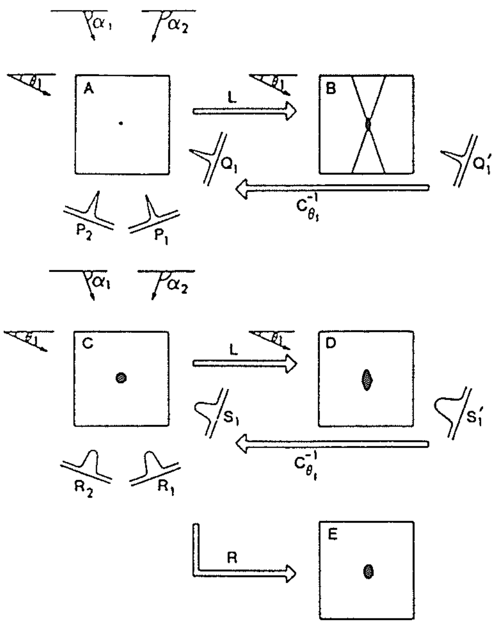


Fig. 2. A schematic representation of geometric deconvolution.

For the reconstructions of the simulated nevi and nevi images, three computed tomography algorithms were used: multiplicative ART (algebraic reconstruction technique) [12], streak preventive ART with adaptive filtering or SPARTAF [13], and multiplicative ART with geometric deconvolution [14], [15]. Multiplicative ART is an ordinary iterative computed tomography algorithm. SPARTAF is an ART-like algorithm oriented towards prevention of streaks via optimization of a cost function based on features of streaks, subject to the constraints of given projection data. This object dependent method employs pattern recognition of streaks and adaptive filtering during iterative reconstruction by ART.

In geometric deconvolution we first reconstruct a known test pattern A (Fig. 2) using projection data $P_1, P_2, \cdots, P_l$ from a limited set of known views at angles $\alpha_1, \alpha_2, \cdots, \alpha_l$. The reconstruction is performed using a linear computed tomography algorithm L. If the test pattern is a point, as shown in Fig. 2, we obtain the point-spread function of the reconstruction process, as in image B. A projection Q_1' of B at a new angle θ_1 is now calculated. The projection Q_1 of test image A is also calculated. We can now compute $C_{\theta_1}^{-1}$, the function that deconvolves Q_1', to give Q_1. Next we reconstruct our unknown image C, using the linear algorithm L and its known projections $R_1, R_2, \cdots, R_l$ at angles $\alpha_1, \alpha_2, \cdots, \alpha_l$ obtaining image D, with the characteristic geometric distortion. A projection S_1' at the angle θ_1, for which we have no projection known, is calculated from image D. S_1' is then deconvolved using $C_{\theta_1}^{-1}$, to obtain an estimated projection S_1, at angle θ_1, of the unknown object C. This operation is performed for the complementary set of angles chosen. The combined set of projections is then used for reconstruction by another computed tomography algorithm R. Algorithm R may be nonlinear, object dependent, and may incorporate *a priori* information. The result is image E, a reconstruction of

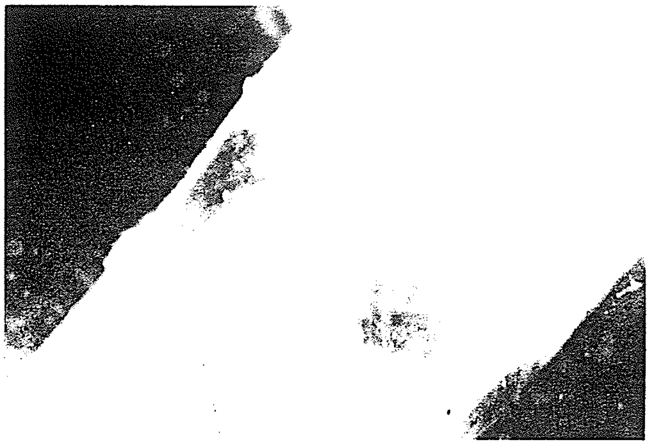


Fig. 3. Incident illumination image of a nevus (on a cadaver) and its reflections in the two mirrors of the nevoscope. The scale is indicated by the separation of the mirrors, which is 3.3 mm.

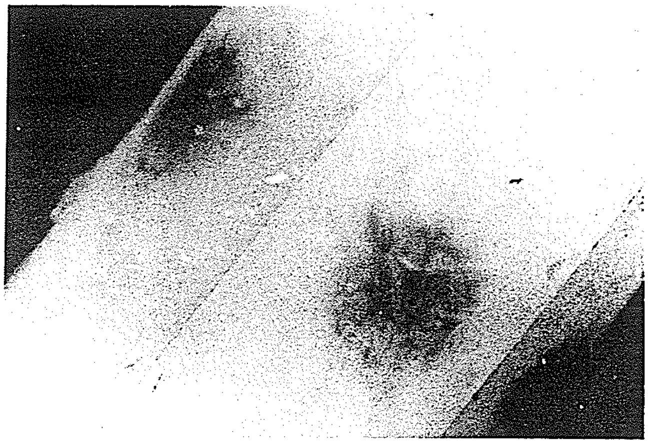


Fig. 4. Transilluminated image of the same nevus and its reflections. The direct image is in the center.

the unknown object with some of the geometric distortion of image D removed. Because geometric deconvolution allows any combination of linear and nonlinear computed tomography algorithms, we refer to it as a "meta-algorithm." In this paper we have used multiplicative ART for both reconstruction stages in geometric deconvolution.

To compare the computed reconstructions with histological sections, we used our method of imaging on a cadaver. A nevus on the cadaver was transilluminated and photographed using the nevoscope and then exised for histological sectioning. The lesion's left top corner was marked on the skin of the cadaver before the tissue was excised. To orient the tissue properly for cutting sections perpendicular to the skin, we first trimmed the tissue to a cubic shape. A dissecting needle was pierced across the dermis to the left of the lesion. A fine wire was pushed through the needle and kept in place during fixation to make a hole across the dermis. The fixation used just after excision of tissue was 10 percent neutral buffered formaline. The excised tissue was embedded in plastic after fixation. 330 serial sections were cut from one side of the nevus to the other across the skin (total width 1.76 mm) at a thickness of 5 μm on a Sorvall microtome using a glass knife. Standard staining with hematoxylin was used [16]. After staining the sections were mounted on slides serially with Protex coverslips. The actual thickness of melanin on each serial section was measured by a micrometer. Thickness was measured from the melanin cells which were found on the surface of the skin to the deepest cluster of melanin cells. We selected a few section levels randomly and found the thickness of melanin part of the nevus in each case. The results were then compared with those obtained from the reconstruction.

Results

Images of nevi obtained by transillumination give a direct impression of depth when viewed in a stereomicroscope. The images obtained of a nevus on a cadaver are shown in Figs. 3 and 4. In incident illumination, due to the bulging of the skin into the space between the two mirrors, often one of the reflections cannot be seen. Nevertheless, both reflected views are seen in transillumination, demonstrating that transillumination is providing a means of seeing *through* the skin.

To confirm our ability to reconstruct the third dimension using computed tomography from only three views, we carried out tests on the images shown in Fig. 5. (Note that the viewing angles are double the angles at which the mirrors are mounted.) Each image, idealizing a vertical cross section through a nevus of a given thickness and elevation as an ellipse, was reconstructed using the SPARTAF algorithm [13], multiplicative ART, and multiplicative ART with geometric deconvolution. We used 45°, 90°, and 180° as the original set of angles in multiplicative ART and SPARTAF algorithms. The angles used for the complementary set in geometric deconvolution were 22.5°, 67.5°, 112.5°, 135°, and 157.5°. Even with geometric deconvolution, the number of views is so small as to result in significant anisotropic resolution.

We took our criterion of success of reconstruction to be a reasonably accurate estimation of the width and thickness of the nevus from the reconstructed images. To obtain such measurements from the simulations, we chose a threshold for contouring the reconstructions. A fit to an ellipse with horizontally and vertically oriented axes was calculated by fitting it to a boundary trace of the threshold reconstruction. For this, the reconstructed image was first converted into a binary image, and then the boundary was traced by chain coding. The major and minor axes were calculated by counting the number of pixels within the boundary along the x and y axes, respectively. Then, the area under the curve (ellipse) was calculated by counting the total number of pixels within the boundary. As can be seen from Fig. 6, the reconstructed major and minor axes of this fit to an ellipse match those of the test pattern over a broad range of threshold values. Thus the choice of a threshold, anywhere within this range, is not critical. We found that selection of a threshold was required when the SPARTAF algorithm was used. But when multiplicative ART with geometric deconvolution was used, this problem was taken care of by applying a suitable constraint during the reconstruction procedure. Pixels with less than the threshold value at any stage is the reconstruction procedure were set to zero. Note that in multiplicative ART, a pixel, once set to zero, remains so.

We also found through the results of our simulation that multiplicative ART with geometric deconvolution gave a better reconstruction of the test pattern contour without ap-

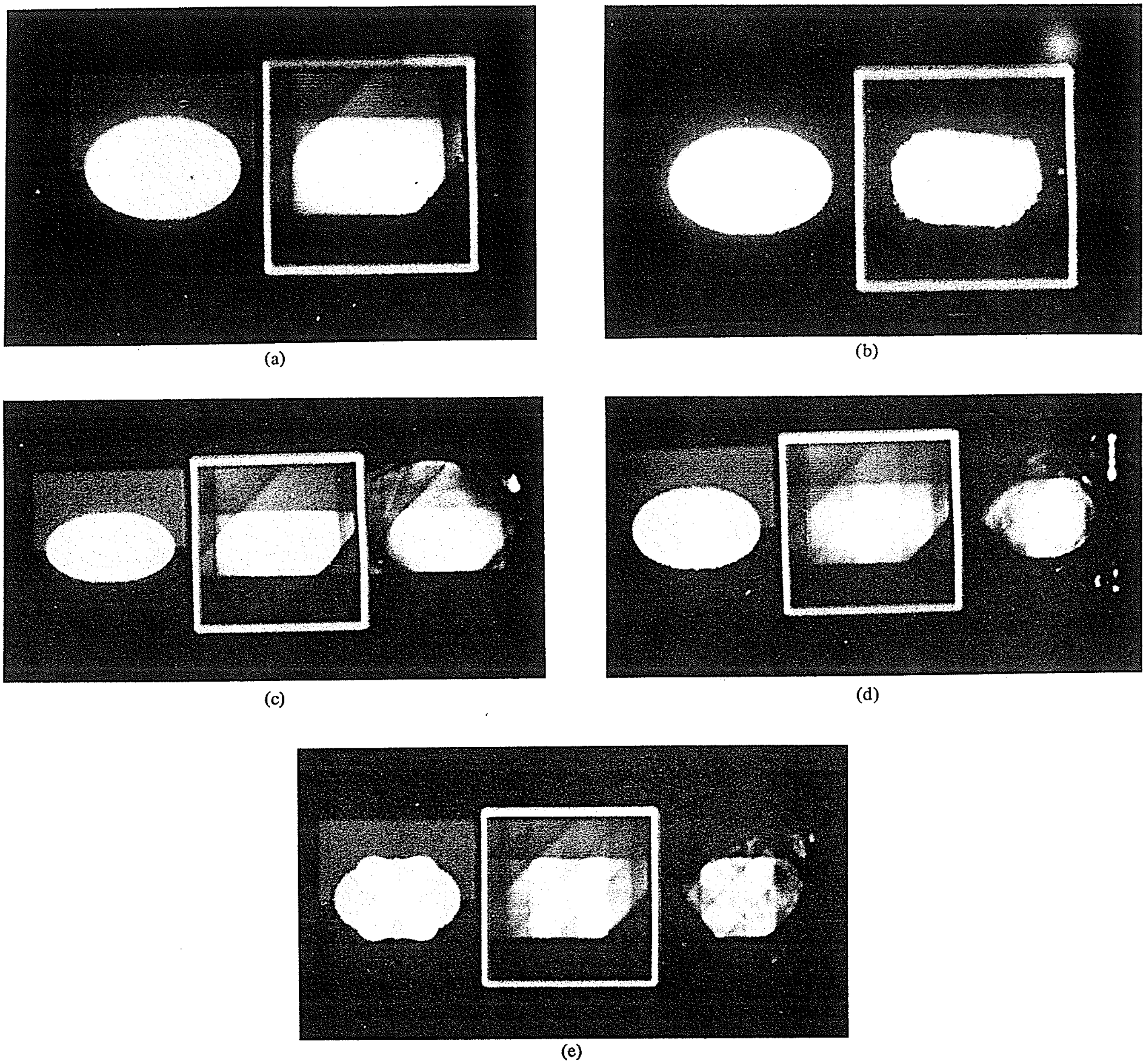


Fig. 5. Computer simulations of reconstructions of elliptical "nevi." (a) Test pattern of eccentricity of 0.78, arbitrary densities of 20, 200, and 2 were assigned to skin, nevus, and air, respectively. Reconstructions were made of the test pattern with angles of 45°, 90°, and 180°. The SPARTAF algorithm was used. (b) Threshold reconstruction of the test pattern and its reconstruction by SPARTAF as in (a). A threshold of 110 units was applied. (c), (d), (e) Show test patterns of different geometric structures at extreme left. Reconstructions of the test patterns using multiplicative ART with angles of 45°, 90°, and 180° are in the middle. Reconstruction of the test patterns using multiplicative ART with geometric deconvolution, where the complimentary angles fill in every 22.5°, are at the extreme right.

plying any external threshold. The ellipse fitting method verified that such reconstruction yielded values of the axes and area very close to their actual values (within 2 percent). Multiplicative ART with geometric deconvolution appeared to give the best reconstruction, retaining the internal structure of nevi when compared with the images produced by the SPARTAF and multiplicative ART algorithms. To test this observation objectively, we designed test patterns with some internal structure (circles with different densities inside an ellipse) and reconstructed them with the three algorithms. The comparison showed that multiplicative ART with geometric deconvolution, in spite of introducing some noise because of the inverse filtering used, not only reconstructed the test patterns with better contours, but retained the internal structure as well. The other algorithms blurred out the internal structure (Fig. 5).

From the digitized images of a transilluminated nevus, we computed the pixel dimension by equating the number of pixels between mirror lines (seen in Fig. 4) to the actual separation of the mirrors in the nevoscope (3.3 mm). From this we identified the corresponding histological serial sections. Two sections, one 0.88 mm away and the other 1.705 mm away from the left edge of the nevus are shown in Fig. 7(a) and (b), respectively. These sections are shown with a magnification of 180. The computed reconstructions of the transilluminated cadaver nevus at section lines 20, 40, 50, 60, 70, and 80 (corresponding to the histological sections 0.055,

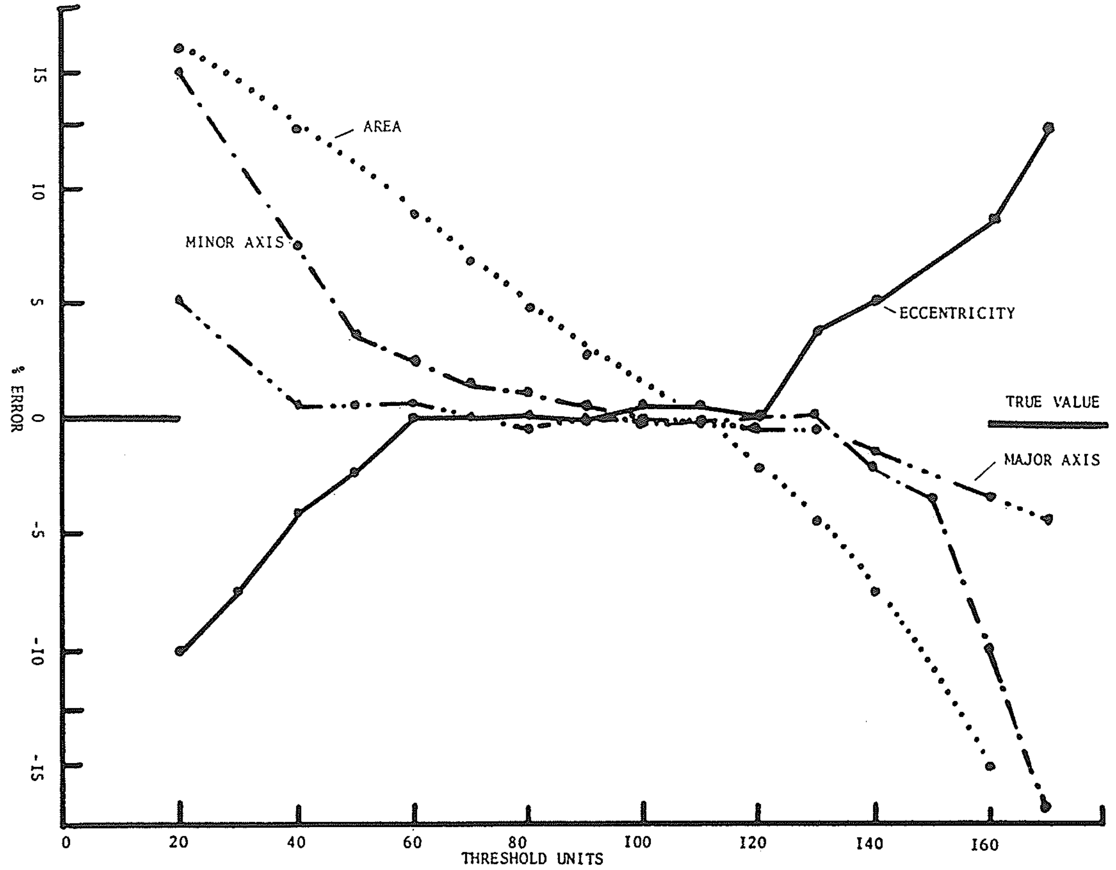


Fig. 6. Plots of the measured major and minor axes of the elliptical fit to the contoured reconstruction of Fig. 5(a). The eccentricity of the ellipse (c/a, where c is the distance from the center to a focus and a is the semimajor axis) and the area of the ellipse are also shown versus threshold, along with their true values.

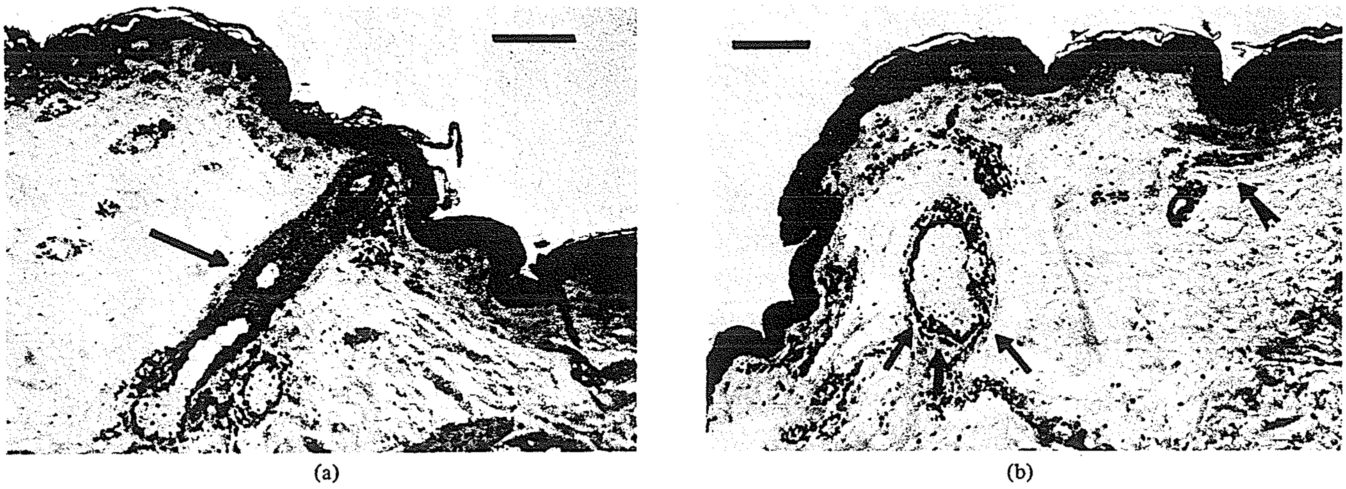


Fig. 7. (a) Histological serial section of the nevus at 0.088 mm away from the left edge of nevus. Measured thickness of nevus is 0.512 mm. The scale bar represents 100 μm. (b) Histological serial section of the nevus at 1.43 mm far from the left edge of the nevus. Measured thickness is 0.586 mm. The scale bar represents 100 μm.

0.605, 0.88, 1.155, 1.43, and 1.705 mm away from left edge of nevus) are shown in Fig. 8(a), (b), (c), (d), (e), and (f), respectively, zoomed by a factor of 4. We estimate the error in this correspondence to be ± 3 sections. Reconstructions corresponding to the sections shown in Fig. 7(a) and (b) are shown in Fig. 8(c) and (e), respectively. Thicknesses of these reconstructions were computed as 0.495 and 0.565 mm, which are 3.32 and 3.58 percent less than those observed by histology. Table I shows the percentage error in measuring thickness at different section levels.

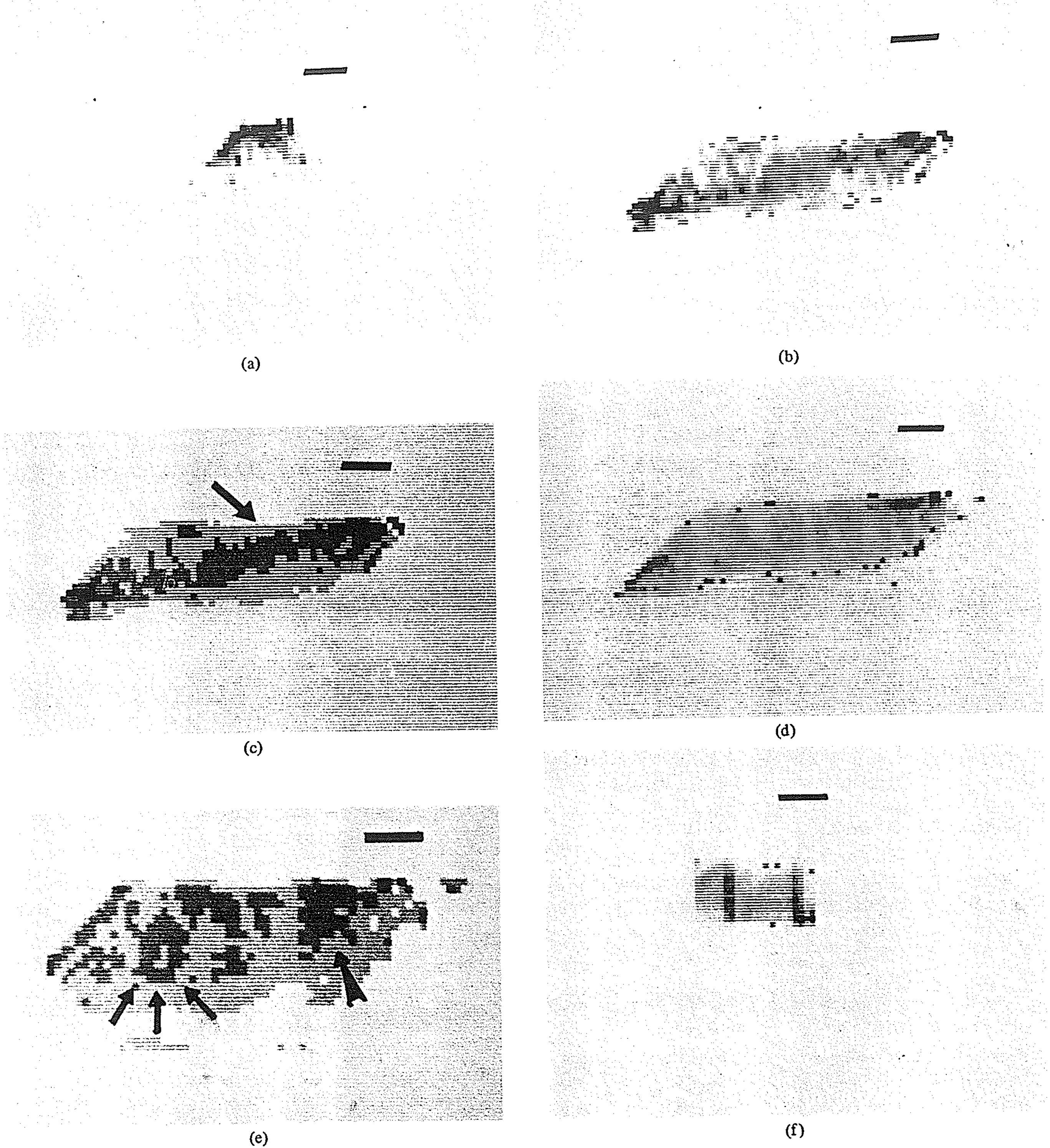


Fig. 8. (a), (b), (c), (d), (e), and (f) Reconstructions of the transilluminated nevus of Fig. 4. Reconstructions have been obtained at section lines 20, 40, 50, 60, 70, and 80 of the 101 × 101 pixels digitized transilluminated nevus images. The most distal section is shown in Fig. 8(f). The scale bar represents 200 μm. The corresponding reconstructions to the sections shown in Fig. 7(a) and (b) are shown in Fig. 8(c) and (e), respectively. Their computed thicknesses are 0.495 and 0.565 mm, respectively.

Discussion

A comparison of Figs. 7(a) and 8(c) shows the similarity between the histological sections and computed reconstructions. The areas marked by arrows in these figures look alike. Similarly, areas marked by arrows in Fig. 7(b) and 8(e) look alike. Note that no circular melanin cell structured is found in the histological section of Fig. 7(a) or in the corresponding reconstruction. But a ring is seen in both the histological

TABLE I
ACTUAL AND COMPUTED THICKNESS OF DIFFERENT SECTIONS ALONG WITH PERCENTAGE ERROR

Distance of section away from left edge of nevus	Line representing section in digitized images	Histological thickness	Computed thickness	% error
0.055 mm	20	0.262 mm	0.275 mm	+4.96
0.605 mm	40	0.458 mm	0.44 mm	-4.09
0.88 mm	50	0.512 mm	0.495 mm	-3.32
1.155 mm	60	0.54 mm	0.523 mm	-3.15
1.43 mm	70	0.586 mm	0.565 mm	-3.58
1.705 mm	80	0.34 mm	0.358 mm	+5.29

Width of nevus measured parallel to mirror axis = 1.76 mm.
Pixel dimension: 1 pixel = 0.0275 mm.

section in Fig. 7(b) and in the corresponding reconstruction [Fig. 8(e)]. Thus, not only the thickness, but some of the details of the morphology are also recoverable by limited-view computed tomography of transilluminated nevi.

Three controllable factors could lead to significant improvements in our reconstructed images of nevi. First, the 180° view of a nervus is out of focus because of its longer optical path length to the microscope. We are developing a correction plate to bring all three images to a common focus. Second, use of inverse filtering in geometric deconvolution introduces noise in the reconstruction. We are developing a new algorithm which uses two-dimensional Wiener deconvolution to remove noise and geometric artifacts in limited-view reconstructions. Third, only three views were used. We are designing a nevoscope with more mirrors or rotatable mirrors.

We have presented a new method for imaging skin lesions to detect changes in thickness, size, and shape of nevi without excising them. This noninvasive technique could be used to detect the early stages of the transformation of a nevus into melanoma, when it is easy to save the life of the patient.

Computed tomography by transillumination is a case of the inverse radiation transfer problem, and received its first mathematical solution in this form by Chandrasekhar [17]. We have treated it in a simpler fashion, in effect assuming that the nevus acts as an absorbing object seen against a bright (transilluminated) background. A more precise treatment, using the mathematics of radiative transfer, might lead to more precise reconstructions [18].

Nevoscopy could be generalized for the investigation of other dermatological problems. For instance, the problem of measuring skin thickness without using ionizing radiation [19] might be solvable by transillumination, especially if grazing images were recorded at different wavelengths and subtracted from one another (a multispectral approach). A general-purpose instrument for skin investigation might be called a "3-D dermascope." It could be carried by a computer controlled robot, which then becomes a "skin scanner." Methods of pattern recognition, computer vision, image processing, and artificial intelligence would have to be developed to enable the skin scanner to locate the same lesion at different times, determine its nature (papule, pustule, nevus, melanoma, etc.), and calculate if significant changes have occurred.

Nevoscopy could also be carried out using emitted infrared light. This would be a form of microthermography. Microthermography (or infrared microscopy) [20], [21] might show up the earliest stages of reddening or erythema often associated with melanoma. Of course, infrared transillumination of nevi could also be considered [22]-[24].

The skin is the largest and most accessible organ, yet investigation of its pathology has tended to be along traditional lines. We hope our efforts here point to a new research direction and will be a step towards automated detection of skin cancer in its early stages.

ACKNOWLEDGMENT

This work has been funded in part by grants from the Natural Sciences and Engineering Council of Canada, Manitoba Medical Services Foundation, and the Dean of Engineering, University of Manitoba. A. P. Dhawan is supported by the Canadian Commonwealth Scholarship and Fellowship Committee. We thank Mr. J. Brexel for construction of the nevoscope, and Mr. J. Dunning and Mr. W. Paley for assistance in image processing. Thanks are also due to Dr. J. Taylor, Dr. F. Baldwin, and L. Nahnybida for their help in histology.

REFERENCES

[1] C. L. Day, Jr., M. C. Mihm, Jr., R. A. Lew, A. W. Kopf, A. J. Sober, and T. B. Fitzpatrick, "Cutaneous malignant melanoma: Prognostic guidelines for physicians and patients," *CA-Cancer J. Clinicians*, vol. 32, no. 2, pp. 113-122, 1982.

[2] C. L. Day, Jr., M. C. Mihm, Jr., R. A. Lew, M. N. Harris, A. W. Kopf, T. B. Fitzpatrick, T. J. Harrist, F. M. Golomb, A. Postel, P. Hennessey, S. L. Gumport, J. W. Raker, R. A. Malt, A. B. Cosimi, W. C. Wood, D. F. Roses, F. Gorstein, D. Rigel, R. J. Friedman, M. M. Mintzis, and A. J. Sober, "Prognostic factors for patients with clinical stage I melanoma of intermediate thickness (1.51-3.99 mm), A conceptual model for tumor growth and metastasis," *Ann. Surg.*, vol. 195, no. 1, pp. 35-43, 1982.

[3] C. L. Day, Jr., R. A. Lew, M. C. Mihm, Jr., A. J. Sober, M. N. Harris, A. W. Kopf, T. B. Fitzpatrick, T. J. Harrist, F. M. Golomb, A. Postel, P. Hennessey, S. L. Gumport, J. W. Raker, R. A. Malt, A. B. Cosimi, W. C. Wood, D. F. Roses, F. Gorstein, D. Rigel, R. J. Friedman, M. M. Mintzis, and R. W. Grier, "A multivariate analysis of prognostic factors for melanoma patients with lesions $\geqslant$ 3.65 mm in thickness, The importance of revealing alternative C_{ox} models," *Ann. Surg.*, vol. 195, no. 1, pp. 44-49, 1982.

[4] C. L. Day, Jr., M. C. Mihm, Jr., A. J. Sober, M. N. Harris, A. W. Kopf, T. B. Fitzpatrick, R. A. Lew, T. J. Harrist, F. M. Golomb A. Postel, P. Hennessey, S. L. Gumport, J. W. Raker, R. A. Malt, A. B. Cosimi, W. C. Wood, D. F. Roses, F. Gorstein, D. Rigel, R. J. Friedman, and M. M. Mintzis, "Prognostic factors for melanoma patients with lesions 0.76-1.69 mm in thickness, An appraisal of thin level IV lesions," *Ann. Surg.*, vol. 195, no. 1, pp. 30-34, 1982.

[5] C. L. Day, M. C. Mihm, Jr., A. J. Sober, T. B. Fitzpatrick, R. A. Malt, A. W. Kopf, R. A. Lew, and T. J. Harrist, "Skin lesions suspected to be melanoma should be photographed: Gross morphological features of primary melanoma associated with metastases," *J. Amer. Med. Assoc.*, vol. 248, no. 9, pp. 1077-1081, 1982.

[6] M. M. Wick, A. J. Sober, T. B. Fitzpatrick, M. C. Mihm, A. W. Kopf, W. H. Clark, and M. S. Blois, "Clinical characteristics of early cutaneous melanoma," *Cancer*, vol. 45, pp. 2684-2686, 1980.

[7] A. J. Sober, C. L. Day, A. W. Kopf, and T. B. Fitzpatrick, "Detection of thin primary melonomas," *CA-Cancer J. Clinicians*, vol. 33, no. 3., pp. 160-163, 1983.

[8] B. R. Cassileth, W. H. Clark, Jr., R. M. Heiberger, V. March, and A. Tenaglia, "Relationship between patients' early recognition of melanoma and depth of invasion," *Cancer*, vol. 49, no. 1, pp. 198-200, 1982.

[9] R. Gordon, "The future of computers in medicine," *Univ. Manitoba Med. J.*, vol. 53, no. 1, pp. 27-31, 1983.

[10] A. P. Dhawan, D. Martin, and R. Gordon, "Computerized imaging of pigmented skin lesions: An adaptive histogram algorithm to isolate melanotic lesions from surrounding skin," *Invest. Dermatol.*, to be published.

[11] A. Rosenfeld, and A. C. Kak, *Digital Picture Processing.* New York: Academic, 1982.

[12] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *J. Theor. Biol.*, vol. 29, pp. 471-81, 1970.

[13] R. M. Rangayyan and R. Gordon, "Streak preventive image reconstruction with ART and adaptive filtering," *IEEE Trans. Med. Imaging*, vol. MI, pp. 173-178, Nov. 1982.

[14] —, "Computed tomography from a few ordinary radiographs for teleradiology," *Med. Physics*, vol. 10, pp. 687-690, 1983.

[15] R. Gordon and R. M. Rangayyan, "Geometric deconvolution: A meta-algorithm for limited view computed tomography," *IEEE Trans. Biomed. Eng.*, vol. BME-30, pp. 806-810, 1983.

[16] G. L. Humason, *Animal Tissue Techniques*, fourth ed. San Francisco, CA: Freeman, 1979.

[17] S. Chandrasekhar, *Radiative Transfer.* New York: Dover, 1960.

[18] A. J. Devaney, "Inverse scattering as a form of computed tomography," *SPIE*, vol. 358, 1982, in press.

[19] C. Y. Tan, "Comparison of xeroradiographic and ultrasound detection of corticosteroid induced dermal thinning," *J. Invest. Dermatol.*, vol. 76, pp. 126-128, 1981.

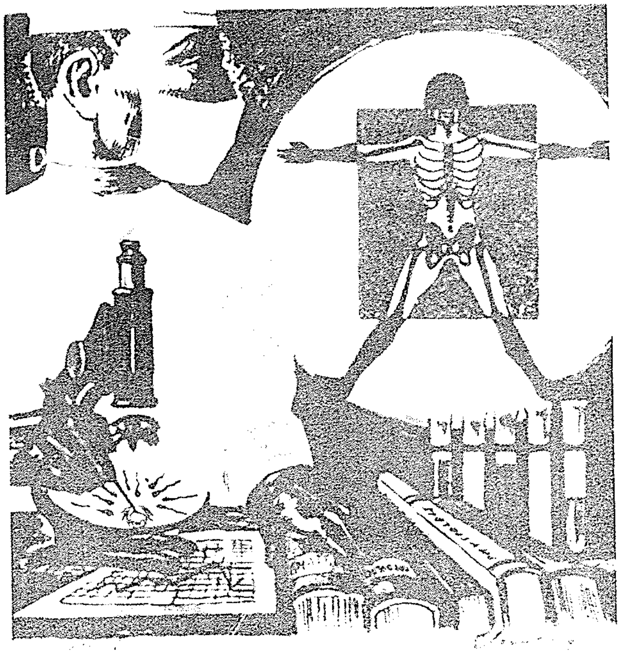
[20] H. Puchtler, S. N. Meloan, and L. D. Paschal, "Infrared fluorescence microscopy of stained tissues: Principles and technique," *Histochem.*, vol. 68, pp. 211-230, 1980.

[21] M. W. Focht, "Infrared microscopy for examination of inks and documents," *Int. Adv. Nondestructive Testing*, vol. 7, pp. 311-325, 1981.

[22] G. A. Todor, "Transillumination in infrared light as a method of location of intraocular neoplasms in th preequatorial zone" (in Russian), *Oftalmol. Zh.*, vol. 33, no. 8, pp. 624-625, 1978.

[23] G. A. Todor *et al.*, "Diaphanoscopic photography in infrared rays" (in Russian), *Oftalmol. Zh.*, vol. 34, no. 3, pp. 179-1981, 1978.

[24] R. Morton *et al.*, "Infrared transillumination using photography and television (videodioscopy)," *J. Audiov. Media Med.*, vol. 4, no. 3, pp. 86-90, 1981.



PROCEEDINGS
ISMII '84

IEEE COMPUTER SOCIETY

INTERNATIONAL SYMPOSIUM ON MEDICAL IMAGES AND ICONS

Hyatt Regency, Arlington, Virginia
July 24-27, 1984

FEATURING: Med**PACS:** Picture Archiving and Communication Systems
Med**PICS:** Picture Interpretation Computers and Systems
Med**GRAPH:** Computer Graphics

EDITED BY
André Duerinckx, Ph.D.
Murray H. Loew, Ph.D.
Judith M. S. Prewitt, Ph.D.

Sponsored by

IEEE Computer Society
Technical Committee on Computational Medicine

1884 1984 A CENTURY OF ELECTRICAL PROGRESS THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

ISBN 0-8186-0544-8
IEEE CATALOG NO. 84CH2047-9
LIBRARY OF CONGRESS NO. 84-80907
COMPUTER SOCIETY ORDER NO. 544

IEEE COMPUTER SOCIETY PRESS

WIENER FILTERING FOR DECONVOLUTION OF GEOMETRIC ARTIFACTS IN LIMITED-VIEW IMAGE RECONSTRUCTION

Atam P. Dhawan*, Rangaraj M. Rangayyan* and R. Gordon**

Departments of Electrical Engineering*, Botony** and Radiology**

University of Manitoba, Winnipeg, Canada R3T 2N2

ABSTRACT

Computed Tomography (CT) images reconstructed using a limited number of projections, measured over a narrow angle range, are characterized by approximately elliptical distortion along the view angles used, and poor contrast at angles not used. This systematic geometric distortion is caused by the two dimensional point spread function of the reconstruction process. In this paper, we show that such geometric distortion and other artifacts introduced in the reconstruction process can be reduced substantially by deconvolution performed via Wiener filtering using a priori knowledge derived from the given projections. The two-dimensional system transfer function used in the deconvolution is obtained from the reconstruction of a test image by a linear reconstruction algorithm (unconstrained multiplicative Algebraic Reconstuction Technique).

INTRODUCTION

Images reconstructed using a limited number of projections suffer from a systematic geometric distorion due to the two dimensional (2D) point spread function of the reconstruction process. We show that such geometric distortion and other artifacts[1] introduced in the reconstruction process can be reduced substantially by deconvolution performed via Wiener filtering using a priori knowledge derived from the given projections. The use of a priori knowledge in restoring the image obtained from limited data has been investigated by various authors.[2-6] In these methods an attempt has been made to fill the missing spatial frequency information in the reconstructed image. The use of Fourier expansion minimization methods is suggested by Inouye[5], Tam and Parez-Mendez[6], and others to compute interpolated projections. These are the iterative schemes which estiamte the missing views. These methods are computaionally complex and time consuming. Moreover they may not restore the image well if the number of known views is very small. We proposed a simple method of removing the geometrical distortion in our previos paper[7] by geometric deconvolution. There, deconvolution was performed via inverse filtering using one-dimensional projections instead of the two-dimensional image. The results, however, were noisy. Also, the method required a second reconstruction process.

The ordinary reconstruction process introduces a systematic geometric distortion and other artifacts in the reconstructed image where the projection information provided is incomplete. We have the following problem: Given a distorted reconstruction, compute a distortion-free and noise-free reconstruction of the image. In other words, deconvolve the geometric distortion and artifacts[8] If we restrict ourselves to a linear reconstruction algorithm such as unconstrained multiplicative Algebraic Reconstruction Technique (ART)[9,10] the problem reduces to estimating the point spread function of the system for a given set of projections and performing appropriate deconvolution.

TWO-DIMENSIONAL DECONVOLUTION METHOD

We perform 2D deconvolution on the limited-view reconstruction of an unknown image via Wiener filtering to remove distortion and obtain a noise-free restored image. We make use of a priori knowledge derived from the given projections in the formulation of the Wiener filter. The limited-view reconstruction should be obtained from a linear reconstruction algorithm.

Let us assume that the reconstruction process produces a degraded image g(x,y) from the given projections[11] of an unknown image f(x,y), modeled as:

$$g(x,y) = h(x,y)*f(x,y) + n(x,y) \tag{1}$$

where h(x,y) is the two-dimensional (2D) point spread function of the reconstruction process and n(x,y) is the noise introduced. The image restoration problem can be written as that of obtaining an approximation $\hat{f}(x,y)$ to f(x,y),

CH2047-9/84/0000/0168$01.00©1984 IEEE

given g(x,y) and a knowledge of h(x,y) and n(x,y). The Wiener filter estimate gives $\hat{F}(u,v)$, the Fourier transform of $\hat{f}(x,y)$, as:

$$\hat{F}(u,v) = [H^2(u,v)/(H^2(u,v)+W(u,v))]G(u,v)/H(u,v) \quad (2)$$

where G(u,v) is the Fourier transform of g(x,y), H(u,v) is the system transfer function, and W(u,v) is the noise-to-signal ratio function in frequency domain.

The system transfer function H(u,v) is obtained from the reconstruction of a point image by the chosen linear reconstruction algorithm for the given set of angles. These angles are the same as those for which the projections of the unknown image are available. We compute the Fourier spectra of the point image and its reconstruction, say J(u,v) and K(u,v) respectively. The 2D system transfer function (or modulation transfer function MTF) is now computed as:

$$H(u,v) = K(u,v)/J(u,v) \quad (3)$$

where the noise is assumed to be zero.

Ideally, K(u,v) should be the same as the system function H(u,v) (when J(u,v) is unity for an ideal impulse). But the system transfer function differs from K(u,v) because of the finite extent of the point image, which introduces a residual phase error in K(u,v).

The approach we have taken is as follows: To begin with, we obtain an initial reconstruction of the unknown image by a linear reconstruction algorithm (unconstrained multiplicative ART) from the known projections. To formulate the Wiener filter for deconvolution, we make use of <u>a priori</u> knowledge derived from the original projection data. From the Fourier slice theorem, we know that the one-dimensional (1D) Fourier transform of a projection corresponds to a radial slice of the 2D Fourier transform of the image, at a corresponding angle. We compute the average of the known projections, and take its 1D Fourier transform. From this we obtain the average 1D Fourier spectrum in the projection domain. We fill the 2D Fourier spectrum of the limited-view reconstruction using this average "slice" to compute an interpolated spectrum. Let this spectrum be denoted by S(u,v). We assume a white noise spectrum N(u,v) with a total energy equal to a small percentage of the total energy in the image, as computed from its interpolated energy spectral density $|S(u,v)|^2$. A 2D noise-to-signal-ratio function is now computed as:

$$W(u,v) = |N(u,v)|^2/|S(u,v)|^2 \quad (4)$$

This is used in the Wiener filter Equation (2).

The restored image may be obtained by taking the inverse Fourier transform of $\hat{F}(u,v)$. It has been found from our experiments that choice of the value of N(u,v) is very critical to achieve optimum restoration. Values of N(u,v) other than the one that gives optimum restoration cause some excessive values in the Fourier spectrum $\hat{F}(u,v)$. These values may be clipped using the interpolated Fourier spectrum of limited-view reconstruction S(u,v) as a 2D clipping function. After clipping we take the inverse Fourier transform of the spectrum to yield the restored image. Fast Fourier transform (FFT) techniques are used throughout the procedure.

RESULTS AND DISCUSSIONS

We first obtained the 2D point spread function of the reconstruction process by reconstructing a point image by multiplicative ART using the given set of angles. We observed from a comparison of the Fourier spectra of the point image and its reconstruction that the phase values at radial sections corresponding to the projections used were similar in the lower spatial frequency region. But, when K(u,v) was used in place of H(u,v) in Eq. 2 for deconvolution, different phase values were observed for the same radial sections for the deconvolved spectrum (Fourier spectrum of the restored image). This resulted in a noisy image when the inverse FFT was taken. We then obtained the system transfer function H(u,v) from Eq. 3 and found that the phase values at radial sections corresponding to the known projections in the low spatial frequency region were zero. We were able to perform proper deconvolution when H(u,v) thus obtained was used in Eq. 2.

We used a test image consisting of a disc in an ellipse with the grey level of 255 and 175 respectively on a 0-255 grey level scale. The background in the test image was assigned grey level value of 20. The test iamge and its limited-view reconstruction obtained by unconstrained multiplicative ART using eight views at 55, 65, 75, 85, 95, 105, 115, and 125 deg. are shown in Figures 1 and 2 respectively. The characteristic geometric distortion in the limited-view reconstruction can be seen along the view angles used in the reconstruction process. The Fourier spectra of the test image (Figure 1) and its reconstruction (Figure 2) are shown in Figures 3 and 4. It is evident from a comparison of Figures 3 and 4 that the Fourier spectrum of the limited-view reconstruction has no significant spatial frequency information along radial sections corresponding to the angles not used in the reconstruction process. We computed the system transfer function H(u,v) from Eq. 3. The interpolated Fourier spectrum of the

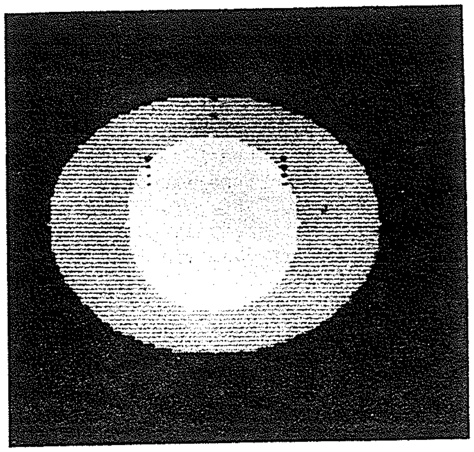


Figure 1. A test image.

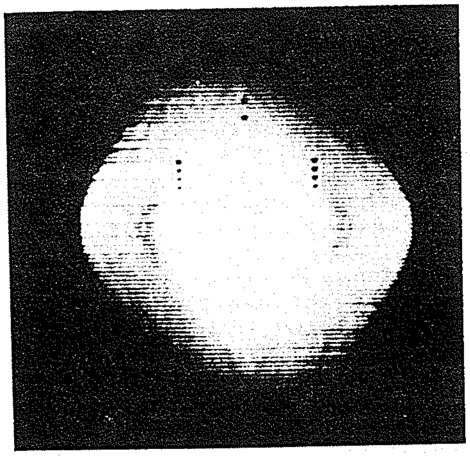


Figure 2. Limited-view reconstruction of the test image by multiplicative ART using eight views at 55, 65, 75,....,125 deg.

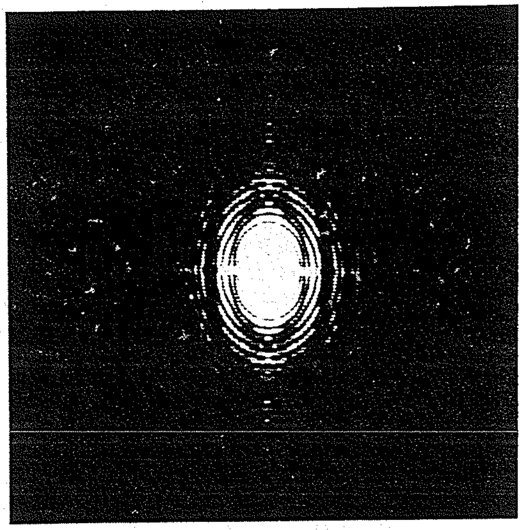


Figure 3. Fourier spectrum of the test image.

limited-view reconstruction obtained as explained above, was used to compute the 2D noise-to-signal ratio function W(u,v) assuming a white noise spectrum N(u,v) with a constant value equal to 2.0% of the variance of the reconstructed image g(x,y). Figures 5 and 6 show the interpolated spectrum and the 2D noise-to-signal ratio function. We performed 2D deconvolution using Eq. 2. The spectrum thus obtained is shown in Figure 7, which shows some excessively large values as bright spots. However, the deconvolved spectrum shows the spectral components in regions for which no information was available in the Fourier spectrum of the limited-view reconstruction (Figure 4). The 2D noise-to-signal ratio function W(u,v) in Eq. 2 can be regarded as a modification function which smooths 1/H(u,v) in order to provide optimum restoration. To achieve optimal restoration W(u,v) must be obtained using the exact model of the noise function N(u,v) in Eq. 4. which is difficult. Since we have assumed the noise to be white, different values of N(u,v) (which is a constant, in case of white noise) will give different restoration performances. To obtain an optimal or a sub-optimal value of N(u,v), one has to try a few values and then the clipping procedure may be used to clip the noise in the deconvolved spectrum. Thus, the critical dependence on N(u,v) may be reduced to a certain extent. The deconvolved spectrum after clipping is shown in Figure 8. The noise in the higher frequency region is almost removed in the clipped spectrum which looks smoother than the deconvolved spectrum shown in Figure 7. The inverse FFT of the clipped spectrum, which is the restored image, is shown in Figure 9. In the restored image it is evident that most of the geometric distortion and artifacts are removed by the 2D deconvolution performed via Wiener filtering.

We also found from our experiments that better results were obtained when the system transfer function H(u,v) was obtained from a basis image containing a disc instead of a point iamge. Thus the system transfer function may be obtained from a basis image b(x,y) based on some *a priori* information about the geometrical shape of the object to be reconstructed[12,13]. For example, the basis image may be a disc for objects having circular structures. The function H(u,v) can be obtained as follows:

$$H(u,v) = D(u,v)/B(u,v) \tag{5}$$

where B(u,v) is the Fourier spectrum of basis image b(x,y), and D(u,v) is the Fourier spectrum of the reconstruction of the basis image. The noise has been assumed to be zero here as in Eq. 3. The image restored from the limited-view reconstruction (Figure 2) using the system

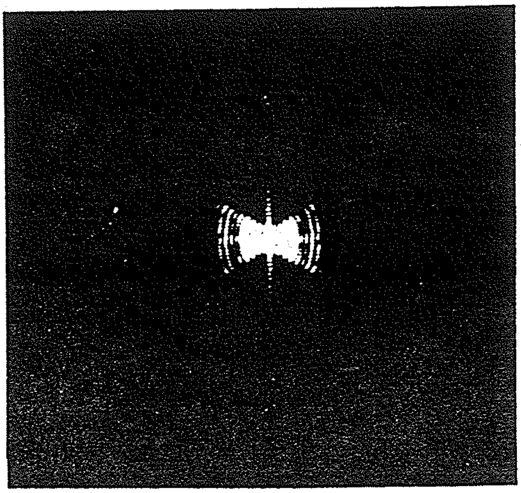


Figure 4. Fourier spectrum of the reconstructed image, shown in Figure 2.

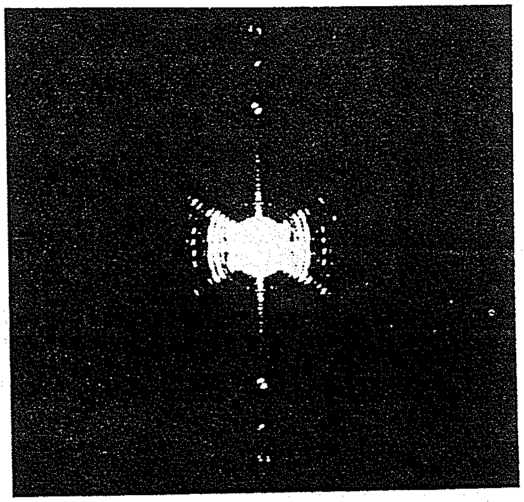


Figure 5. The interpolated spectrum of Figure 4 which is used to obtain the the 2D noise-to-signal ratio function, and also used as a clipping function.

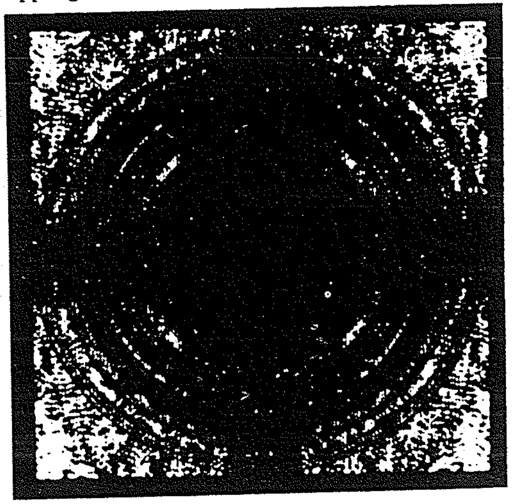


Figure 6. The 2D noise-to-signal ratio function.

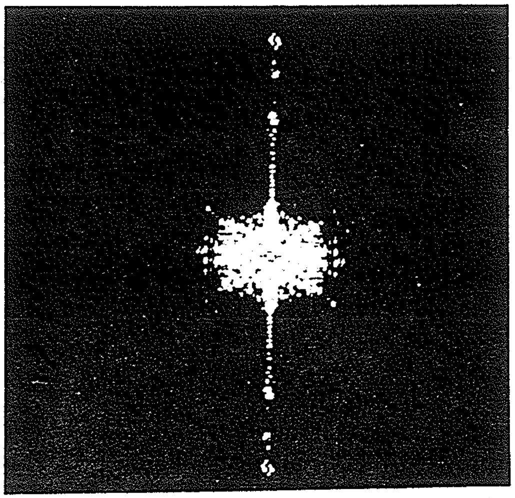


Figure 7. The deconvolved spectrum obtained after Wiener deconvolution. The system transfer function H(u,v) was obtained from the 2D point spread function.

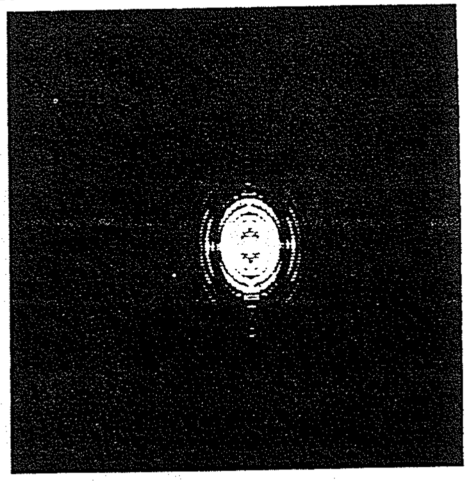


Figure 8. The deconvolved spectrum after clipping.

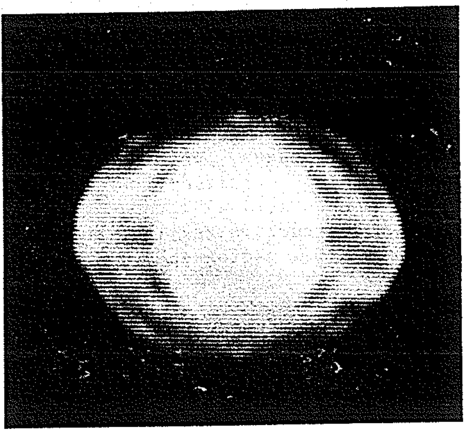


Figure 9. The restored image obtained by taking the inverse FFT of the spectrum shown in Figure 8.

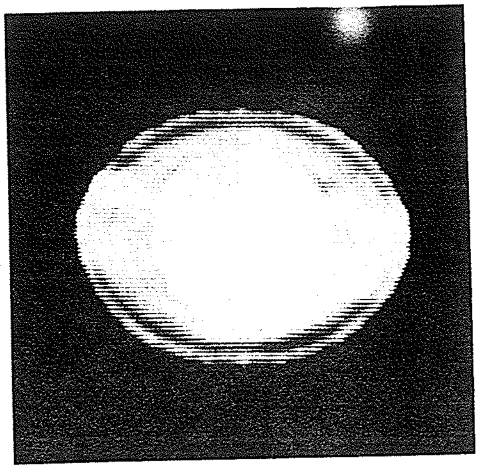


Figure 10. The restored image when a disc image was used as a basis function to obtain the system transfer function H(u,v).

transfer function obtained from a disc image of radius 6 pixels, is shown in Figure 10. It is evident from a comparison of the two restored images (shown in Figures 9 and 10) that restoration is better when a disc is used to obtain the system transfer function. The circle is restored better in Figure 10.

CONCLUSION

We have described a deconvolution method for removing geometric distortion and artifacts in limited-view reconstruction by using the Wiener filter method. We believe the success achieved here is due to our novel method for use of a priori knowledge derived from the known projections, both in obtaining the noise-to-signal ratio function for use in the Wiener filter as well as the clipping procedure. We are now applying these techniques to restoration of reconstructions obtained by simple back projection method. Our methods should improve images obtained in many applications of limited-view computed tomography, such as electron microscopy, industrial non-destructive testing, and computed tomography from a few radiographs.

ACKNOWLEDGEMENTS

A.P. Dhawan was supported by Canadian Commonwealth Fellowship Plan. We thank Paul Soble, whose suggestions have led to some of the techniques developed here. This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Gordon, R., Artifacts in reconstructions made from a few projections., Proceedings of the First International Joint Conference on Pattern Recognition, Oct. 30 to Nov. 1, Washington, D. C., IEEE Computer Society, Northridge, California, pp. 275-285., 1973.

[2] Sato, T., S.J. Norton, M. Linzer, O. Ikeda & M. Hirama, Tomographic image reconstruction from limited projections using interactive revision in image and transform spaces, Applied Optics 20(3), 395-399, 1981.

[3] Digest, Optical Society of America Topical Meeting on Signal Recovery and Synthesis with Incomplete Information, Lake Tahoe, Nevada, Jan. 12-14,1983.

[4] Hall, T.J., A.M. Darling and M.A. Fiddy, Image reconstruction and restoration incorporating prior knowledge, Optics Letters (10) 7 467-468, 1982.

[5] Inouye, T., Image reconstruction with limited angle projection data, IEEE Trans. Nuclear Science NS-26 2 pp.2666-2669, 1979.

[6] Tam, K.C. & V. Parez-Mendez, Limits to image reconstruction from restricted angular input, IEEE Trans. NS-28 179-183, 1981.

[7] Gordon, R. & R.M. Rangayyan, Geometric deconvolution: a meta-algorithm for limited view computed tomography, IEEE Trans. Biomed. Eng., 30(12), 806-810, 1983.

[8] Gordon, R., A proposal for deconvolution of reconstructions., Workshop on Information Treatment in Electron Microscopy, Basel, 2pp., 1972.

[9] Gordon, R., A tutorial on ART (Algebraic Reconstruction Techniques)., IEEE Trans. Nucl. Sci. NS-21, 78-93, 95., 1974.

[10] Gordon, R., & G. T. Herman, Three dimensional reconstruction from projections: A review of algorithms., Int. Rev. Cytol. 38, 111-151., 1974.

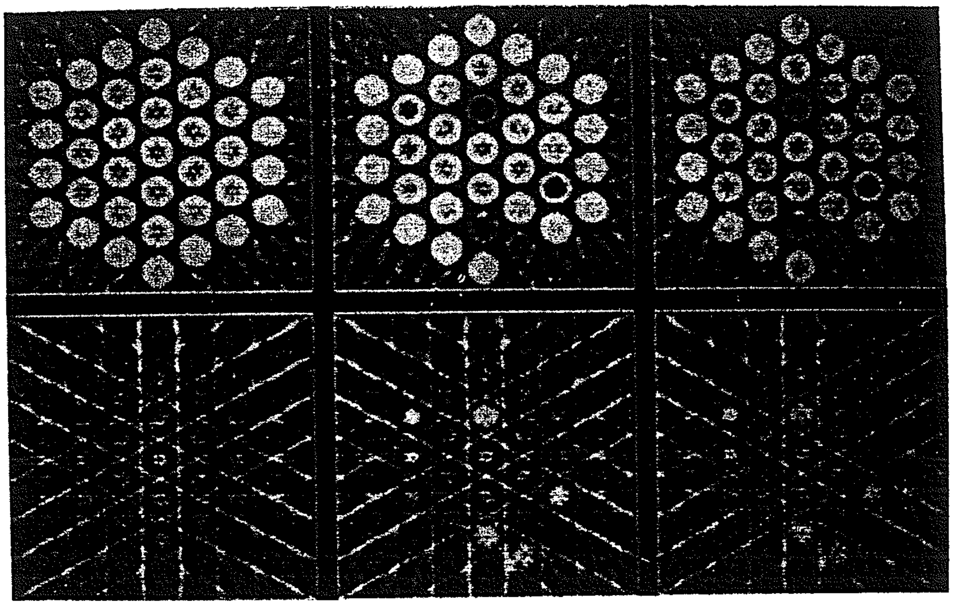
[11] Gonzalez, R.C. & P. Wintz, Digital Image Processing, Addison-Wesley, 1977.

[12] Burch, S.F., S.F. Gull & J. Skilling, Image restoration by a powerful maximum entropy method, Computer Vision, Graphics, Image Proc. 23, 113-128, 1983.

[13] Andrews, H.C. & B.R. Hunt, Digital Image Restoration, Englewood Cliffs, Prentice-Hall, 1977.

TOPICAL MEETING ON

INDUSTRIAL APPLICATIONS OF COMPUTED TOMOGRAPHY AND NMR IMAGING



TECHNICAL DIGEST

AUGUST 13-14, 1984
Hecla Island, Manitoba, Canada

Image Restoration by Two-Dimensional Deconvolution in Limited-View Reconstruction

A.P. Dhawan, R.M. Rangayyan and R. Gordon

University of Manitoba, Winnipeg, Canada R3T 2N2

The use of a priori knowledge in restoring images reconstructed using limited data has been investigated by various authors [1-4] In these methods an attempt has been made to fill the missing spatial frequency information in the reconstructed image. The use of an optical feedback system [2] and the use of a Fourier expansion minimization method [5] have been suggested to recover the lost information. The main disadvantge of these techniques is that they are very time consuming.

The ordinary reconstruction process introduces systematic distortion and artifacts in the reconstructed image. We have the following problem: Given a distorted reconstruction, compute a distortion and noise-free reconstruction of the image. In other words, deconvolve the geometric distortion and artifacts If we restrict ourselves to a linear reconstruction algorithm such as unconstrained multiplicative Algebraic Reconstruction Technique [ART] [6] the problem reduces to estimating the point spread function of the system for a given set of projections and performing appropriate deconvolution. (see also [7])

Let us assume that the reconstruction process produces a degraded image g(x,y) from the given projections of an unknown image f(x,y), modeled as (see [8]):

$$g(x,y) = h(x,y)*f(x,y) + n(x,y) \tag{1}$$

where h(x,y) is the two-dimensional (2D) point spread function of the reconstruction process and n(x,y) is the noise introduced. The image restoration problem can be written as that of obtaining an approximation $\hat{f}(x,y)$ to f(x,y), given g(x,y) and a knowledge of h(x,y) and n(x,y). The Wiener filter estimate gives $\hat{F}(u,v)$, the Fourier transform of f(x,y), as:

$$\hat{F}(u,v) = [H^2(u,v)/(H^2(u,v)+W(u,v))]G(u,v)/H(u,v) \tag{2}$$

The system transfer function H(u,v) is obtained from the reconstruction of a point image by the chosen linear reconstruction algorithm for the given set of angles. These angles are the same for which the projections of the unknown image are available. We compute Fourier spectra of the point image and its reconstruction, say J(u,v) and K(u,v) respectively. The 2D system transfer function (or modulation transfer function MTF) is now computed as:

$$H(u,v) = K(u,v)/J(u,v) \tag{3}$$

Ideally, K(u,v) should be the same as system function H(u,v) (when

$J(u,v)$ is unity for an ideal impulse). But system transfer function differs from $K(u,v)$ because of finite extent of the point image which introduces residual phase error in $K(u,v)$.

The approach we have taken is as follows: To begin with, we obtain an initial reconstruction of the unknown image by a linear reconstruction algorithm (unconstrained multiplicative ART) from the known projections. Figures 1 and 2 show a test pattern and its reconstruction respectively. To formulate the Wiener filter for deconvolution, we make use of a priori knowledge derived from the original projections data. From the Fourier slice theorem, we know that the one-dimensional (1D) Fourier transform of a projection corresponds to a radial slice of the 2D Fourier transform of the image, at a corresponding angle. We compute the average of the known projections, and take its 1D Fourier transform. From this we obtain the average 1D Fourier spectrum in the projection domain. We fill the 2D Fourier spectrum of the limited-view reconstruction using this average "slice" to compute an interpolated spectrum. Let this spectrum be denoted by $S(u,v)$. We assume a white noise spectrum $N(u,v)$ with a total average energy equal to a small percentage of the total energy in the image, as computed from its interpolated energy spectral density $|S(u,v)|^2$. A 2D noise-to-signal-ratio function is now computed as:

$$W(u,v) = |N(u,v)|^2 / |S(u,v)|^2 \tag{4}$$

This is used in the Wiener filter Equation 2.

The restored image may be obtained by taking the inverse Fourier transform of $\hat{F}(u,v)$. We found from our experiments that choice of the value of $N(u,v)$ (a constant in case of white noise assumption) is very critical in achieving the optimum restoration. Values of $N(u,v)$ other than one that gives optimum restoration cause some excessive values in Fourier spectrum $\hat{F}(u,v)$. These values may be clipped using interpolated Fourier spectrum of limited-view reconstruction as 2D clipping function. After clipping we take inverse Fourier transform of the spectrum to yield distortion-free restored image. Fast Fourier transform techniques are used throughout the procedure. Figure 3 shows the restored image where most of the distortion has been removed. We are trying to use this deconvolution technique for restoration of images in limited-view CT of mammography, angiography and non-destructive testing.

REFERENCES

[1] Sato, T., S.J. Norton, M. Linzer, O. Ikeda & M. Hirama, Tomographic image reconstruction from limited projections using interactive revision in image and transform spaces, Applied Optics 20(3), 395-399, 1981.

[2] Sato, T., K. Saski, Y. Nakamura, M. Linzer & S.J. Norton, Tomographic image reconstruction from limited projections using coherent optical feedback, Applied Optics 20(17), 3073-3076, 1981.

[3] Byrne, C.L. & R.M. Fitzgerald, Reconstruction from parallel information with application to tomography, SIAM, Applied Math. (42) 4 933-940, 1982.

[4] Hall, T.j., A.M. Darling and M.A. Fiddy, Image reconstruction and restoration incorporating prior knowledge, Optics Letters (10) 7 467-468, 1982.

[5] Inouye, T., Image reconstruction with limited angle projection data, IEEE Trans. Nuclear Science NS-26 2 pp.2666-2669, 1979.

[6] Gordon, R., A tutorial on ART (Algebraic Reconstruction Techniques)., IEEE Trans. Nucl. Sci. NS-21, 78-93, 95., 1974.

[7] Gordon, R & R.M. Rangayyan, Geometric deconvolution: A meta-algorithm for limited-view computed tomography, IEEE Trans. Biomedical Eng., 30(12), 806-810,1983

[8] Gonzalez, R.C., P. Wintz, Digital Image Processing, Addison-Wesley Publishing Company, 1977

FIGURE CAPTIONS

Figure 1. Test image.

Figure 2. Limited-view reconstruction of test image by unconstrained multiplicative ART using projections at 55, 65, 75,...,125 deg.

Figure 3. Restored image.

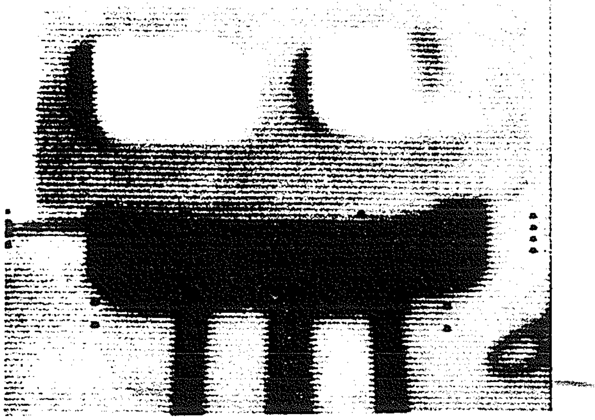


Figure 1

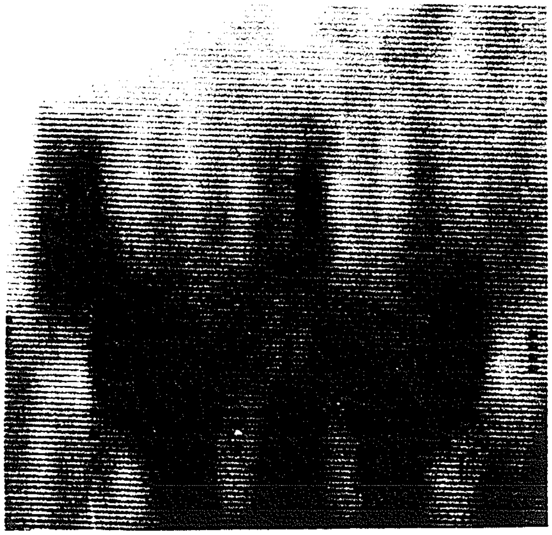


Figure 2

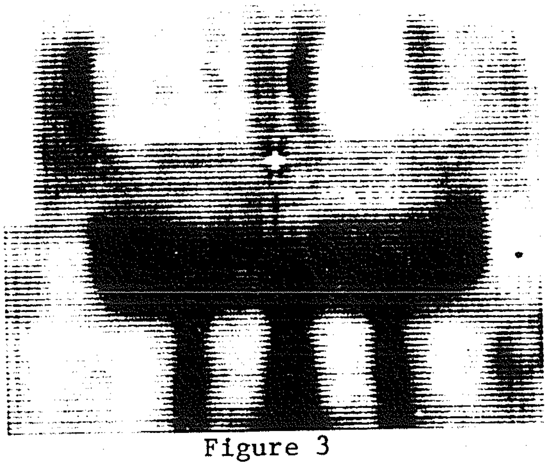
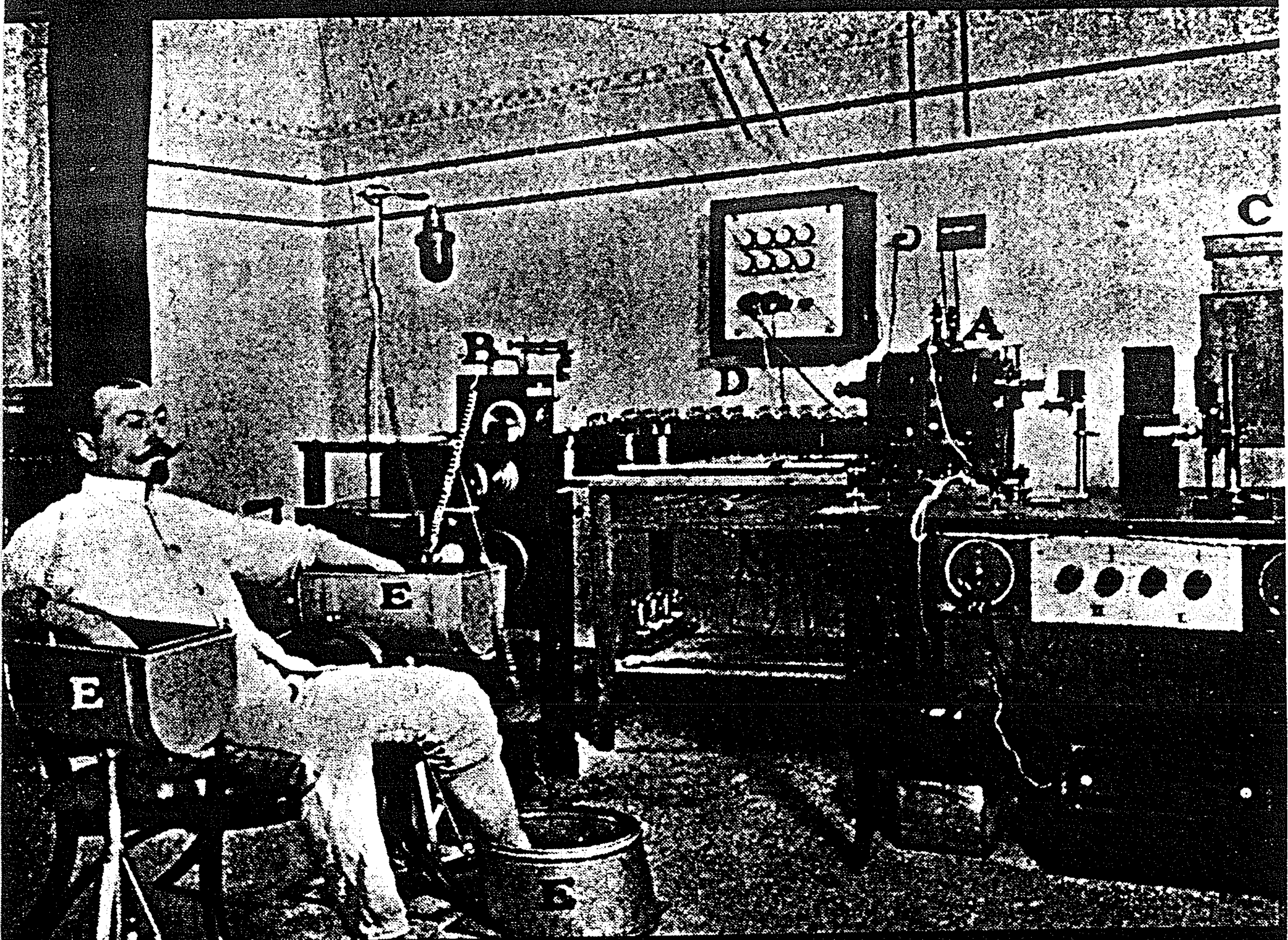


Figure 3

Frontiers of Engineering and Computing in Health Care — 1984

Los Angeles, CA — 15-17 September 1984



Proceedings — Sixth Annual Conference

IEEE Engineering in Medicine and Biology Society

1884 1984
A CENTURY OF ELECTRICAL PROGRESS

84CH2058-6

COMPUTED TOMOGRAPHY BY TRANSILLUMINATION TO DETECT EARLY MELANOMA

Atam P. Dhawan*, Richard Gordon** and Rangaraj M.Rangayyan*

Departments of Electrical Engineering* and Radiology**, University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

Melanoma is the most malignant primary cutaneous tumor and the leading killer in dermatological disease. It has been found that changes in thickness, size, and color are the best features to detect melanomas in the early, curable phase. We present a non-invasive method based on three-dimensional (3D) computed tomography (CT) of a nevus _in situ_ by transillumination to detect melanoma in early stages. We transilluminate a nevus using fibre optics directed into the surrounding skin. We obtain three images of the transilluminated nevus at different angles by an optical device we developed called a "nevoscope". 3D reconstruction of the nevus is then obtained using our recently developed limited-view CT algorithms. A quantitative analysis may be performed on consecutive 3D reconstructions to bring out any changes in thickness, size, color, and structure of the nevus over time.

INTRODUCTION

Melanoma is the most malignant primary cutaneous tumor among dermatological diseases. Often the origin of a melanoma is a pigmented nevus. The malignant transformation is accompanied by pigmentary increase, induration, and ulceration at later stages [1]. The change from a benign to a malignant lesion produces only a few warning symptoms that are often not observed by the patient [2]. The primary signs are increase in thickness and size, and change in color. Bleeding, ulceration, tenderness, and itching occur less frequently [1]. The thickness of the lesion is found to be the best prognostic factor for clinical stage 1 (early) mealnoma patients even when regional lymph nodes contain metastases [3]. But until now this could not be measured without excision and histological sectioning of the nevus.

It has been found that primary tumor thickness and the anatomical location are the best features to predict the survival relationship in clinical stage 1 patients [2], when cure is more probable. Level of invasion and thickness of the lesion have been found highly correlated [3].

We have developed an optical device called a "nevoscope" to obtain three images (a vertical image at 90 deg., a glancing image at 180 deg., and one at 45 deg.) of the nevus which is transilluminated using the fiber optics directed into the surrounding skin. These images are picked up by a digitizing TV camera. The digitized images are used in computed tomography (CT) algorithms to obtain vertical cross-sections of the nevus. We use our recently developed CT algorithm which removes geometric distortion and other artifacts in limited-view reconstruction. This algorithm performs two-dimensional deconvolution via Wiener filter incorporating _a priori_ knowledge derived from the known projections. We perform a quantitaive analysis to obtain the thickness of these sections. We performed our method on a cadaver and compared the computed reconstructions of a nevus on the cadaver, with the histology of the nevus.

MATERIALS AND METHODS

We built our nevoscope out of a plexiglass cylinder machined to fit around the objective lens of a Wild M8 stereomicroscope. Nylon screws permit it to be moved up and down along the tube of the objective lens for focussing. A plexiglass plate screws into the bottom of the cylinder. This plate (Figure 1) contains a trapezoidal slot. Two front surface mirrors (9 x 19 x 1.8 mm) were glued onto the slanted parts of the slot at intended angles of 45 deg. and 22.5 deg. from the vertical. The actual angles were measured by reflection of a beam of light and were found to be 45 deg. and 23 deg. respectively. Light for imaging was provided by a Volpi heat filtered illuminator using Philips 15V, 150W bulbs. Two of its three fiber optics bundles were directed in a plane perpendicular to the slot through two holes drilled at 45 deg through the cylinder and the plate (Figure 1). These were allowed to protrude slightly (about 2 mm) so that they would dent the skin and make direct contact with it. (The protruding edges of the mirrors were ground down with a carborundum stone to allow the skin to bulge inwards next to the silvered surfaces.) Some of the superficially backscattered light was blocked off with black tape on the walls of the trapezoidal slot. The transilluminated

CH2058-6/84/0000-0518 $1.00 © 1984 IEEE

image, so obtained, was recorded on 35 mm Panatonic-X (Kodak) film and then the film was digitized as described below.

The images were displayed on a Conrac color television monitor via a 512 x 480 pixel Grinnell frame buffer with 10 bit planes connected to a Cyber 171 computer via a CAMAC crate. Image enhancement, thresholding, and boundary tracing were carried out using our Manitoba Video Processor (MVP) software. Digitization was accomplished at 6 bits density resolution with averaging of 16 consecutive video frames by the frame buffer. The video image was acquired with a Sony video camera using a Nikon 55 mm Macro lens on a C-mount adapter.

For the reconstruction of the simulated nevi and nevi images, unconstrained multiplicative ART (Algebraic Reconstruction Technique) [4] was used. Multiplicative ART is an ordinary iterative computed tomography algorithm. Ordinary reconstruction algorithms like ART introduce systematic geometric distortion and artifacts in the reconstructed image. We used our recently developed two-dimensional (2D) deconvolution technique to remove geometric distortion in the image reconstructed from limited number of views. This 2D deconvolution is based on the Wiener filter formulation incorporating a priori knowledge derived from the known projections and may be described in brief, as follows:

Let us assume that the reconstruction process produces a degraded image g(x,y) from the given projections of an unknown image f(x,y), modeled as [6]:

$$g(x,y) = h(x,y)*f(x,y)+n(x,y)$$

where h(x,y) is 2D point spread function of the reconstruction process and n(x,y) is the noise introduced. In the Fourier domain, the Wiener filter estimate gives the F(u,v) as:

$$F(u,v) = [H^2(u,v)/(H^2(u,v)+W(u,v))]G(u,v)/H(u,v)$$

where G(u,v) is the Fourier spectrum of the limited-view reconstruction, W(u,v) is a 2D noise-to-signal ratio function, and H(u,v) is the system transfer function.

The system transfer function H(u,v) is obtained from the reconstruction of a disc image of radius four pixels by multiplicative ART for the given set of angles. These angles are the same for which the projections of the unknown image are available. Then we compute the Fourier spectra of the disc image and its reconstruction, say J(u,v) and K(u,v) respectively. The system transfer function (or modulation transfer function MTF) is now computed as:

$$H(u,v) = K(u,v)/J(u,v)$$

(assuming zero noise).

The noise-to-signal ratio function is obtained as follows: We compute the average of the known projections, and take its 1D Fourier transform. From this we obtain the average 1D Fourier spectrum in the projection domain. We fill the 2D Fourier spectrum of the limited-view reconstruction using this average "slice" to compute an interpolated spectrum. Let this spectrum be denoted by S(u,v). We assume a white noise spectrum N(u,v) with a total average energy equal to a small percentage of the total energy in the image, as computed from its interpolated energy spectral density $|S(u,v)|^2$. The 2D noise-to-signal ratio function is now computed as:

$$W(u,v) = |N(u,v)|^2/|S(u,v)|^2$$

This is used in the Wiener filter formulation.

The excessively large values of the estimated Fourier spectrum F(u,v), caused by computational problems of deconvoluion, are clipped by using the interpolated spectrum as the 2D clipping function. The image is obtained by taking the inverse Fourier transform of the clipped spectrum F(u,v).

To compare the computed reconstructions with histological sections, we used our method of imaging on a cadaver. A nevus on the cadaver was transilluminated and photographed using the nevoscope and then excised for histological sectioning. The lesion's left top corner was marked on the skin of the cadaver before the tissue was excised. To orient the tissue properly for cutting sections perpendicular to the skin, we first trimmed the tissue to a cubic shape. A dissecting needle was pierced across the dermis to the left of the lesion. A fine wire was pushed through the needle and kept in place during fixation to make a hole acrosss the dermis. The fixative used just after excision of tissue was 10% neutral buffered formaline. The excised tissue was embedded in plastic after fixation. 330 serial sections were cut from one side of the nevus to the other across the skin (total width 1.76 mm.) at a thickness of 5 microns on a Sorvall microtome using a glass knife. Standard staining with hematoxylin was used [5]. After staining the sections were mounted on slides serially with Protex coverslips. The actual thickness of melanin on each serial section was measured by a micrometer. The thickness was measured from the melanin cells which were found on the surface of the skin to the deepest cluster of melanin cells. We selected a few section levels randomly and found the thickness of the melanin part of the nevus in each case. The results were then compared with those obtained from the reconstruction.

RESULTS

To confirm our ability to reconstruct the third dimension using CT algorithms from only three views, we carried our test on a test pattern of a simulated section of the nevus. This test pattern as shown in Figure 3(a) has an ellipse with four discs. The discs representing melanin structure were assigned a density of 200 on a 0-255 grey scale. The ellipse representing the nevus was assigned a density of 170 with backgrounds of 20, and 2 representing skin and air. The test pattern, its reconstruction using multiplicative ART, and the restored image by Wiener deconvolution are shown in Figures 3(b) and 3(c). The angles used in multiplicative ART were 45, 90, and 180 deg. It can be seen that most of the distortion of the ART reconstruction is removed in the restored image.

From the digitized images of a transilluminated nevus, we computed the pixel dimension by equating the number of pixels between mirror lines (seen in Figure 2) to the actual separation of the mirrors in the nevoscope (3.3 mm). From this we identified the corresponding histological serial sections. Two sections, one 0.88 mm away and the other 1.705 mm away from the left edge of the nevus are shown in Figures 4 and 5 respectively. These sections are shown with a magnification of 180. The computed reconstructions of the transilluminated cadaver nevus were obtained at section lines 20, 40, 50, 60, 70, and 80 (corresponding to the histological sections 0.055mm, 0.605 mm, 0.605 mm, 0.88 mm, 1.155 mm, 1.43 mm, and 1.705 mm away from the left edge of the nevus). Two reconstructions corresponding to the sections shown in Figures 4 and 5 are shown in Figures 6 and 7 respectively, zoomed by a factor of 4. We estimate the error in this correspondence to be ± 3 sections. Thicknesses of these reconstructions were computed as 0.495 mm and 0.565 mm which are 3.32% and 3.58% less than those observed by histology. Table 1 shows the percentage error in measuring thickness at different section levels.

Discussion

A comparison of Figures 4 and 6 shows the similarity between the histological sections and computed reconstructions. The areas marked by arrows in these Figures look alike. Similarly areas marked by arrows in Figures 5 and 7 look alike. Note that no circular melanin cell structure is found in the histological section of Figure 4 nor in the corresponding reconstruction. But a ring is seen in both the histological section in Figure 5 and in the corrsponding reconstruction (Figure 7). Thus not only the thickness, but some of the details of the morphology are also recoverable by limited-view computed tomography of transilluminated nevi.

Two controllable factors could lead to significant improvements in our reconstructed images of nevi. First, the 180 deg. view of the transilluminated nevus is out of focus because of its longer optical path length to the microscope. We are developing a correction plate to bring all three images to a common focus. Second, only three views were used. We are designing a nevoscope with more mirrors or rotatable mirrors.

The skin is the largest and most accessible organ, yet investigation of its pathology has tended to be along traditional lines. We hope our efforts here point to a new research direction and will be a step towards automated detection of skin cancer in its early stages.

REFERENCES

[1] Day, C.L., Jr., M.C. Mihm, Jr., R.A. Lew, A.W. Kopf, A.J. Sober & T.B. Fitzpatrick, Cutaneous malignant melanoma: prognostic guidelines for physicians and patients, CA-A Cancer J. Clinicians 32(2), 113-122, 1982.

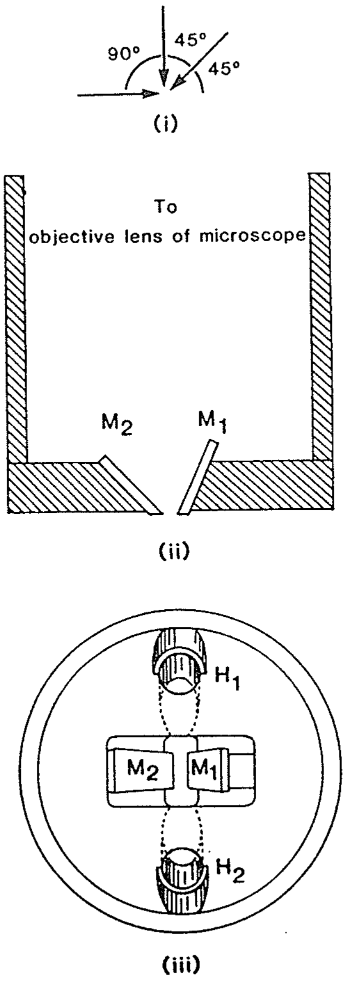
[2] Wick, M.M., A.J. Sober, T.B. Fitzpatrick, M.C. Mihm, A.W. Kopf, W.H. Clark & M.S. Blois, Clinical characteristics of early cutaneous melanoma, Cancer 45, 2684-2686, 1980.

[3] Sober, A.J., C.L. Day, A.W. Kopf & T.B. Fitzpatrick, Detection of "thin" primary melanomas, Ca-Cancer J. Clinicians 33(3), 160-163, 1983.

[4] Gordon, R., A tutorial on ART (Algebraic Reconstruction Technique), IEEE Trans. Nucl. Sci. NS-21, 78-93, 1974.

[5] Humason, G.L., Animal Tissue Techniques, Fourth edition, W.H. Freeman & Company, San Francisco, 1979.

[6] Gonzalez, R.C. & P. Wintz, Digital Image Processing, Addison-Wesely Publishing Company, 1977.

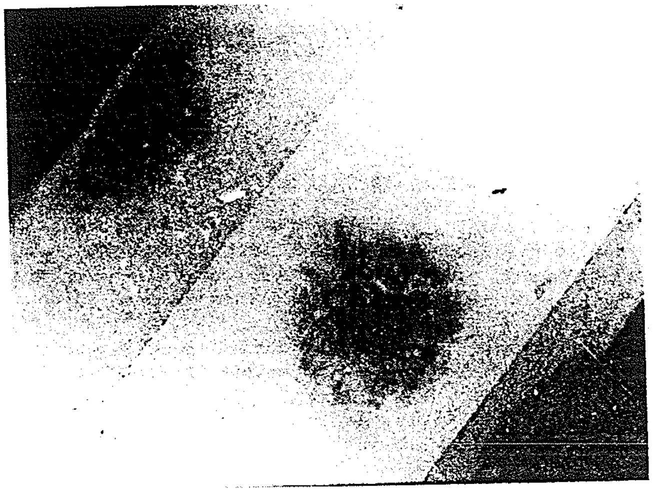


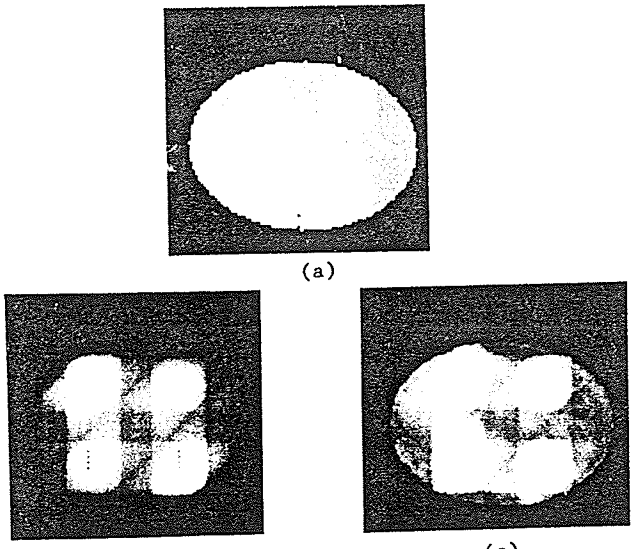
(i)

(ii)

(iii)

Figure 1.(i). Viewing angles of images yielded by nevoscope. (ii). sectional view, and (iii). mirrors bearing plate of nevoscope. M_1 and M_2 are the front silvered mirrors. H_1 and H_2 are holes for fibre optics bundles to transilluminate the skin.





(a)

(b)

(c)

Figure 3. (a). Test image of simulated nevus, (b) its reconstruction by multiplicative ART, and (c) image restored by Wiener filter.

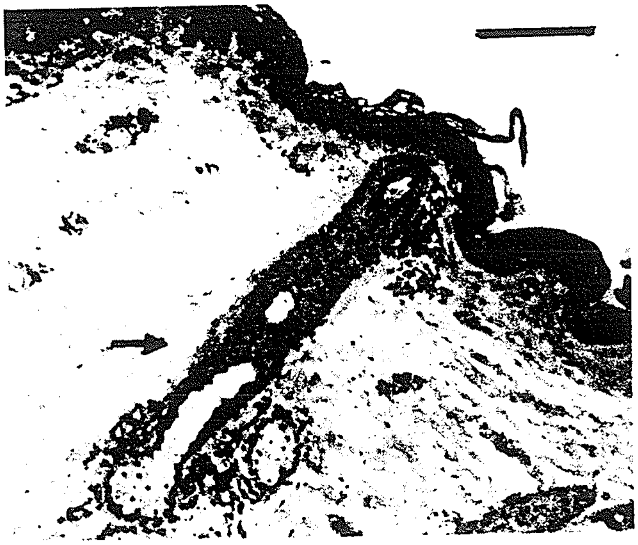


Figure 4. Histological serial section of the the nevus at 0.088 mm away from the left edge of nevus. Measured thickness of nevus is 0.512 mm. The scale bar represents 100 microns.

Figure 2. Images of transilluminated nevus as yielded by nevoscope. The scale is indicated by the separation of mirrors, which is 3.3 mm.

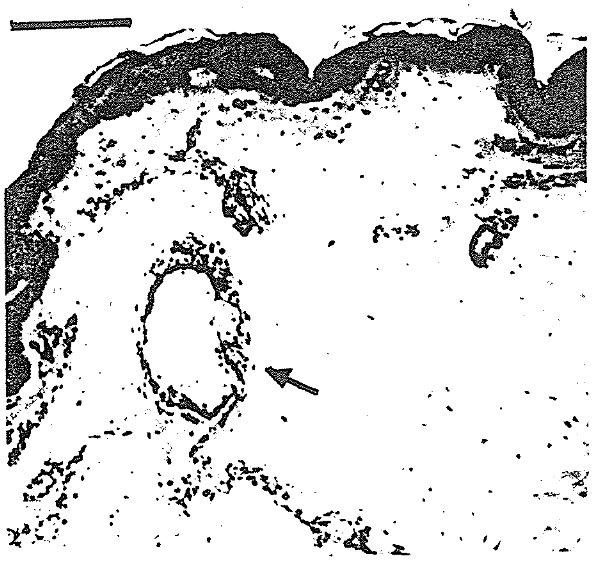


Figure 5. Histological serial section of the nevus at 1.43 mm far from the left edge of the nevus. Measured thickness is 0.586 mm. The scale bar represents 100 microns.

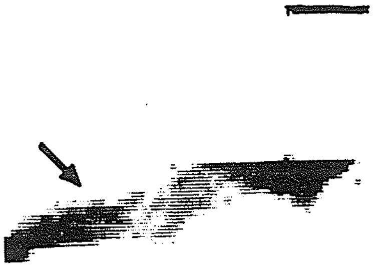


Figure 6. Reconstruction of the section shown in Figure 4. The scale bar represents 200 microns. The computed vertical thickness is 0.495 mm.

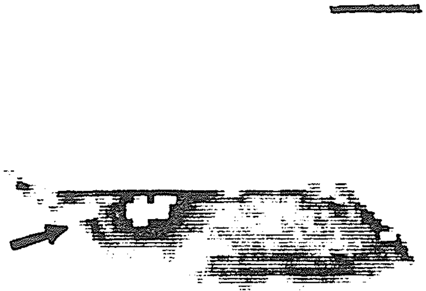


Figure 7. Reconstruction of the section shown in Figure 5. The scale bar represents 200 microns. The computed vertical thickness is 0.565 mm.

Distance of section away from left edge of nevus.	Line representing section in digitized images.	Histological thickness	Computed thickness	% error
0.055 mm	20	0.262 mm	0.275 mm	+4.96
0.605 mm	40	0.458 mm	0.44 mm	-4.09
0.88 mm	50	0.512 mm	0.495 mm	-3.32
1.155 mm	60	0.54 mm	0.523 mm	-3.15
1.43 mm	70	0.586 mm	0.565 mm	-3.58
1.705 mm	80	0.34 mm	0.358 mm	+5.29

Width of nevus measured parallel to mirror axis = 1.76 mm

Pixel dimension - 1 pixel = 0.0275 mm

Table 1. Table showing actual and computed vertiacl thickness of sections along with percentage error.