Near Sets: Theory and Applications

by

Christopher James Henry

A Thesis
submitted to the Faculty of Graduate Studies,
in Partial Fulfilment of the Requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada

Near Sets: Theory and Applications

by

Christopher James Henry

A Thesis
submitted to the Faculty of Graduate Studies,
in Partial Fulfilment of the Requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Abstract

The focus of this research is on a tolerance space-based approach to image analysis and correspondence. The problem considered in this thesis is one of extracting perceptually relevant information from groups of objects based on their descriptions. Object descriptions are represented by feature vectors containing probe function values in a manner similar to feature extraction in pattern classification theory. The motivation behind this work is the synthesizing of human perception of nearness for improvement of image processing systems. In these systems, the desired output is similar to the output of a human performing the same task. Thus, it is important to have systems that accurately model human perception. Near set theory provides a framework for measuring the similarity of objects based on features that describe them in much the same way that humans perceive the similarity of objects. In this thesis, near set theory is presented and advanced, and work is presented toward a near set approach to performing content-based image retrieval. Furthermore, results are given based on these new techniques and future work is presented. The contributions of this thesis are: the introduction of a nearness measure to determine the degree that near sets resemble each other; a systematic approach to finding tolerance classes, together with proofs demonstrating that the proposed approach will find all tolerance classes on a set of objects; an approach to applying near set theory to images; the application of near set theory to the problem of content-based image retrieval; demonstration that near set theory is well suited to solving problems in which the outcome is similar to that of human perception; two other near set measures, one based on Hausdorff distance, the other based on Hamming distance.

Keywords: Description, near sets, tolerance near sets, tolerance space, perception, probe functions, feature values, nearness measure, content-based image retrieval (CBIR).

Acknowledgements

Many thanks the to following people for their role to in the completion of this thesis.

- My adviser, James F. Peters, for his council and advice, steadfast encouragement, great discussions, insights, valued feedback, and, most importantly, positive energy. Thank you for being a great role model and a great person.
- My examination committee, Miroslaw Pawlak, Pradeepa Yahampath, Robert Thomas, and Som A. Naimpally, for their insights and helpful discussions.
- Sheela Ramanna for collaborating on coauthored papers.
- Maciej Borkowski, Homa Fashandi[1], Dan Lockery, Amir H. Meghdadi, Leszek Puzio, Shabnam Shafar, and Piotr Wasilewski. All these people have at one point have been a part of our research group, and they have all provided many great discussions, helpful insights, and warm friendships.
- Wes Mueller and Manitoba Hydro for their generous support.
- The Natural Sciences and Engineering Research Council of Canada (NSERC).
- My parents, family, and friends for continuing to be part of my strong foundation and their constant support.
- Lastly, my beautiful wife Tanya Henry for her love and never ending encouragement, our little peanut Rhyla Henry, and baby number two that is on the way.

This research has been supported by grant T277 from Manitoba Hydro, NSERC grant 185986, Manitoba Centre of Excellence Fund (MCEF) grant, Canadian Network Centre of Excellence (NCE) and Canadian Arthritis Network (CAN) grant SRI-BIO-05.

[1]Special thanks to Homa for being the first to bring to the group the idea that successful content-based image retrieval depends on texture-based features.

Contents

List of Tables

List of Figures

1 Introduction

The focus of this thesis is on perceptual nearness theory and applications. The view of the perception of nearness presented in this thesis combines the basic understanding of perception in psychophysics with a view of perception found in Merleau-Ponty's work [1]. That is, perception of the nearness of objects (*i.e.*, in effect, our knowledge about objects) depends on sensor signals gathered by our senses. The proposed approach to perception is feature-based and is similar to the one discussed in the introduction of [2]. In this view, our senses are likened to probe functions, *i.e.*, mappings of sensations to values assimilated by the mind. A human sense modelled as a probe measures the physical characteristics of objects in our environment. The sensed physical characteristics of an object are identified with object features. It is our mind that identifies relationships between object feature values to form perceptions of sensed objects [1]. In this thesis, it is conjectured that perception, *i.e.* human perception of nearness, can be quantified through the use of near sets by providing a framework for comparing objects based on object descriptions. Objects that have similar appearance (*i.e.*, objects with similar descriptions) are considered *perceptually near each other*. Sets are considered near each other when they have "things" (perceived objects) in common. Specifically, near sets facilitate measurement of similarity between objects based on feature values (obtained by probe functions) that describe the objects. This approach is similar to the way humans perceive objects (see, *e.g*, [3]) and as such facilitates the creation of perception-based systems.

This thesis is divided into two main components, namely a section presenting (and contributing to) near set theory, and a section demonstrating that near set theory can be used in applications where the result is similar to that of human perception. Specifically, near set theory is applied to the problem of content-based image retrieval, which is a research area focusing on the retrieval of images from a database based on the perceptual content of the images, as well as segmentation evaluation, an application concerned with measuring the quality of an image partition to capture the perceptual objects contained within the

image. Implicit in most applications concerning digital images is human perception of images, and the objects we perceive that the images contain. A system that employs image processing either purposely or inadvertently mimics the human visual system due to fact that these systems ultimately aide visual perception by people. Thus a generalization of definitions for image processing leads to the manipulation of digital images for the purpose of extracting or enhancing perceptual information contained in images.

The term *perception* appears in the literature in many different places with respect to the processing of images. For instance, the term is often used for demonstrating that the performance of methods are similar to results obtained by human subjects (as in [4]), or it is used when the system is trained from data generated by human subjects (as in [5]). Thus, in these examples, a system is considered perceptual if it mimics human behaviour. Another illustration of the use of perception is in the area of semantics with respect to queries [6, 7]. For instance, [7] focuses on queries for 3-D environments, *i.e.*, performing searches of an online virtual environment. Here the question of perception is one of semantics and conceptualization with regard to language and queries. For example, users might want to search for a tall tree they remembered seeing on one of their visits to a virtual city.

Other interpretations of *perception* are tightly coupled to psychophysics, *i.e.* perception based on the relationship between stimuli and sensation [8]. For example, [9] introduces a texture perception model. The texture perception model uses the antagonistic view of the human visual system in which our brain processes differences in signals received from rods and cones rather than sense signals, directly. An image-feature model of perception has been suggested by Mojsilovic *et al.* [10], where it is suggested that humans view or recall an image by its dominant colours only, and areas containing small, non-dominant colours are averaged by the human visual system. Other examples of the term perception defined in the context of psychophysics have also been given [11–17].

Perception as explained by psychologists [18, 19] is similar to the understanding of perception in psychophysics. In a psychologist's view of perception, the focus is more on the mental processes involved rather than interpreting external stimuli. For example, [19]

presents an algorithm for detecting the differences between two images based on the representation of the image in the human mind (*e.g.*, colours, shapes, and sizes of regions and objects) rather than on interpreting the stimuli produced when looking at an image. In other words, the stimuli from two images have been perceived and the mind must now determine the degree of similarity.

Much work has been reported based on the perceptual approach presented by near set theory [20–25], an outgrowth of the rough set approach to obtaining approximate knowledge of objects that are known imprecisely [26–30]. The perceptual approach to nearness presented in this thesis is an outgrowth of research into applications of near set theory [31–33], as well as, application of near sets to the image processing problems of segmentation evaluation [34, 35], image correspondence [33, 36–39]. Briefly, disjoint sets containing objects with matching descriptions are called near sets. The discovery of near sets begins with the selection of probe functions that provide a basis for describing and discerning affinities between sample objects (see, *e.g.*, [20, 40, 41]). A *probe function* is a real-valued function representing a feature of physical objects. The perceptual approach of near set theory is based on the idea that our mind identifies relationships between object features to form perceptions of sensed objects. As was mentioned, our senses gather the information of the objects we perceive and map sensations to values assimilated by the mind. Thus, our senses can be likened to perceptual probe functions in the form of a mapping of stimuli from objects in our environment to sensations (values used by the mind to perceive objects). It is this idea of probe functions that is at the heart of near sets.

This thesis presents the theory and applications of near sets. The approach is by way of extracting perceptually relevant information from a set of objects, where each object has an associated feature vector describing object features (perceived object characteristics such as colour). It is the information contained in these feature vectors that is used to extract perceptual information from classes of objects and to measure the similarity among them. The contributions presented in this thesis are:

- The introduction of a nearness measure to determine the degree that near sets resem-

ble each other,

- A systematic approach to finding tolerance classes, together with proofs demonstrating that the proposed approach will find all tolerance classes on a set of objects,
- An approach to applying near set theory to images,
- The application of near set theory to the problem of content-based image retrieval,
- Demonstration that near set theory is well suited to solving problems in which the outcome is similar to that of human perception,
- Two other near set measures are considered, one based on Hausdorff distance, the other based on Hamming distance.

This thesis is organized as follows: Section 2.1 contains background information on near set theory and image processing; Section 3 presents tolerance near sets, and introduces the nearness measure tNM; Section 5 presents an application of near set theory to the problem of content-based image retrieval; and Section 6 concludes the thesis. In addition, Appendix A presents another application of near set theory by way of segmentation evaluation, and Appendix B presents the NEAR System, a GUI that contains an implementation of all the code used to generate the results presented in this thesis, and is freely available at [42].

2 Background

2.1 Near Sets

Nearness is an intuitive concept that allows us to function in our daily lives. At a young age, we become adept at identifying the similarity of objects in our environment, and can quickly assess the degree of similarity. In fact, our day-to-day conversations are full of adverbs and adjectives used to encapsulate the nearness of "things" in our environment. Phrases like "he was about as tall as you are" or "those suits look similar" serve to demonstrate the frequency with which we are making analogies to objects that are not the same, but share some common characteristics. However, it was only recently, relatively speaking, that this idea of nearness was first explored mathematically. Frigyes Riesz first published a paper in 1908 on the nearness of two sets, initiating a field of study which has now become known as proximity spaces [43, 44]. Proximity spaces axiomatically characterize the proximity relation, a relation that, in brief, provides a framework for identifying the nearness of a point to a set and the nearness of two sets, where nearness is based on the spatial relationship between objects, rather than in terms of the descriptions associated with the objects. In contrast, near set theory is concerned with the nearness of objects based on their descriptions.

The introduction of near set theory was significantly influenced by work on rough set theory established by Z. Pawlak in 1981 [26], and by the work of E. Orłowska on approximation spaces [45, 46]. Briefly, a set X is considered a rough set if X cannot be reproduced by the union of cells in a partition, where the partition is defined by an equivalence relation on object descriptions, called the indiscernibility relation. A set that is considered rough can be approximated using this relation. The main concept from rough set theory present in near set theory is the notion of indiscernibility of objects. Near set theory was founded on the idea that two disjoint sets of objects are near each other, if they contain objects that have the same descriptions, which can be identified using the indiscernibility relation. The principal difference between rough set theory and near sets (as reported in [25, Section 6.1,

pp. 18]) is that near sets can be discovered without the approximation of sets.

Near set theory was inspired by a collaboration in 2002 by Z. Pawlak and J. F. Peters on a poem entitled "How Near" [47]. The poem's theme is the human perception of nearness, conveying imagery about the proximity of snow flakes to trees, and the nearness of icicles to the ground. At the same time, work began on applying rough set theory to the problem of measuring the similarity of images, *i.e.* image correspondence, and content-based image retrieval, the problem of retrieving images by their content rather than strings of text associated with the image. It was these events which led to the first two publications on near set theory in [21, 48]. These papers represent the introduction of near theory, *e.g.* the introduction of the definitions fundamental to the field of near set theory, and mark a transition from focusing on the approximation of a single set, as in rough set theory, to discovering the nearness/similarity of disjoint sets based on object descriptions. This can be seen by the discovery of near sets in approximation spaces, *i.e.* the approximation of one set by another set [21], the introduction of a nearness approximation space (also in [21]), and the approximation of the nearness of objects in [48]. Notice that these papers are still using the terminology in the approximation of sets that are qualitatively near each other, a fact that shows the influence of rough set theory in the introduction of near sets.

Subsequently, there have been many publications in the area of near set theory [25]. While, near set theory was motivated by the image correspondence problem, the first few papers on near set theory had as their aim the application of near sets to the problem of discovering affinities between perceptual information granules, where a perceptual granule is a non-empty set containing objects with common descriptions [20, 23, 24]. As will become apparent in this thesis, sets that are near each other are examples of information granules (see, *e.g.*, definition in [23]). Other examples of the application of near set theory include: identification of features in an automated facial feature extraction procedure [49]; image correspondence [33, 36–39] (still using approximation spaces [50]); adaptive learning [31] and near set theory applied to patterns of observed swarm behaviour stored in tables called ethograms, where the goal is to measure the resemblance between the behaviours of differ-

ent swarms [51]; and finally image morphology and segmentation evaluation [34, 35].

Finally, it is worth mentioning that the papers [20, 48] also contain a nice discussion on the differences between features and attributes, which is an important distinction between the traditional applications of rough set theory, and the application of near set theory presented in this thesis. Namely, an attribute is a partial function measuring some characteristic of an object. In contrast, a feature is a perceived characteristic of an object, and there can be more than one feature for a given characteristic, *e.g.* colour can be described by the RGB colour model or the HSV colour model.

2.1.1 Perceptual Systems

A logical starting point for a discussion on near set theory begins with establishing a basis for describing elements of sets. All sets in near set theory consist of perceptual objects.

Definition 1. Perceptual Object. *A* perceptual object *is something perceivable that has its origin in the physical world.*

A perceptual object is anything in the physical world with characteristics observable to the senses such that they can be measured and are knowable to the mind. Examples of perceptual objects include patients, components belonging to a manufacturing process, and camera images. Here, the term *perception* is considered relative to measurable characteristics called the object's features.

In keeping with the approach to pattern recognition suggested by M. Pavel [52], the features of an object are quantified by probe functions.

Definition 2. Probe Function [21, 40]. *A* probe function *is a real-valued function representing a feature of a perceptual object.*

In this work, probe functions are defined in terms of digital images such as: colour, texture, contour, spatial orientation, and length of line segments along a bounded region. In the context of near set theory, objects in our visual field are always presented with respect to the selected probe functions. Moreover, it is the probe functions that are used to measure

characteristics of visual objects and similarities among perceptual objects, making it possible to determine if two objects are associated with the same pattern without necessarily specifying which pattern (as is the case when performing classification).

Next, a perceptual system is a set of perceptual objects, together with a set of probe functions.

Definition 3. Perceptual System [25]. *A* perceptual system $\langle O, \mathbb{F} \rangle$ *consists of a non-empty set* O *of sample perceptual objects and a non-empty set* $\mathbb{F}$ *of real-valued functions* $\phi \in \mathbb{F}$ *such that* $\phi : O \rightarrow \mathbb{R}$.

The notion of a perceptual system admits a wide variety of different interpretations that result from the selection of sample perceptual objects contained in a particular sample space O. Two examples of perceptual systems are: a set of images together with a set of image processing probe functions, or a set of results from a web query together with some measures (probe functions) indicating, *e.g.*, relevancy or distance (*i.e.* geographical or conceptual distance) between web sources.

Combining Definitions 1 & 2, the description of a perceptual object within a perceptual system can be defined as follows.

Definition 4. Object Description. *Let* $\langle O, \mathbb{F} \rangle$ *be a perceptual system, and let* $\mathcal{B} \subseteq \mathbb{F}$ *be a set of probe functions. Then, the* description *of a perceptual object* $x \in O$ *is a feature vector given by*

$$\boldsymbol{\phi}_{\mathcal{B}}(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_l(x)),$$

where l *is the length of the vector* $\boldsymbol{\phi}_{\mathcal{B}}$*, and each* $\phi_i(x)$ *in* $\boldsymbol{\phi}_{\mathcal{B}}(x)$ *is a probe function value that is part of the description of the object* $x \in O$.

Note, the idea of a feature space is implicitly introduced along with the definition of object description. An object description is the same as a feature vector as described in traditional pattern classification [53]. The description of an object can be considered a point in an l-dimensional Euclidean space $\mathbb{R}^l$ called a feature space. As was mentioned in the

introduction, near set theory is concerned with the nearness of objects based on their descriptions. Thus, the relationship between objects is discovered in a feature space that is determined by the probe functions in $\mathcal{B}$.

2.1.2 Perceptual Indiscernibility Relations

Building on the foundational definitions of a perceptual system and the description of an object, this section introduces the perceptual indiscernibility relation and its relationship to near sets. Near set theory originated with the indiscernibility relation, an equivalence relation defined with respect to object descriptions (see *e.g.* [25, 48]). Recall, a relation on two sets is a subset of their Cartesian product, and an equivalence relation is any relation that is reflexive, symmetric, and transitive.

Definition 5. Perceptual Indiscernibility Relation [25, 26]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. For every $\mathcal{B} \subseteq \mathbb{F}$ the* perceptual indiscernibility relation $\sim_{\mathcal{B}}$ *is defined as follows:*

$$\sim_{\mathcal{B}} = \{(x, y) \in O \times O : \forall \phi_i \in \mathcal{B} \, . \, \phi_i(x) = \phi_i(y)\}.$$

The perceptual indiscernibility relation is a variation of the one given by Z. Pawlak in 1981 [26]. Furthermore, notice that equivalence is defined with respect to the description of an object, *i.e.* objects are considered equivalent when the features used to describe them are the same.

Using the indiscernibility relation (together with the probe functions in $\mathcal{B}$), a set of objects can be partitioned into classes of objects with matching descriptions such that each class has the highest possible object resolution under the indiscernibility relation. These classes are called elementary sets or equivalence classes and are given in Definition 6.

Definition 6. Equivalence Class. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $x \in O$. For a set $\mathcal{B} \subseteq \mathbb{F}$ an* equivalence class *is defined as*

$$x_{/\sim_{\mathcal{B}}} = \{x' \in O \mid x' \sim_{\mathcal{B}} x\}.$$

Observe that a single object is sufficient to label the class, since all objects in a class have the same description. Moreover, the set of all equivalence classes induced by the partition of a set using the indiscernibility relation is called a quotient set .

Definition 7. Quotient Set. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. For a set $\mathcal{B} \subseteq \mathbb{F}$ a* quotient set *is defined as*

$$O_{/\sim_{\mathcal{B}}} = \{x_{/\sim_{\mathcal{B}}} \mid x \in O\}.$$

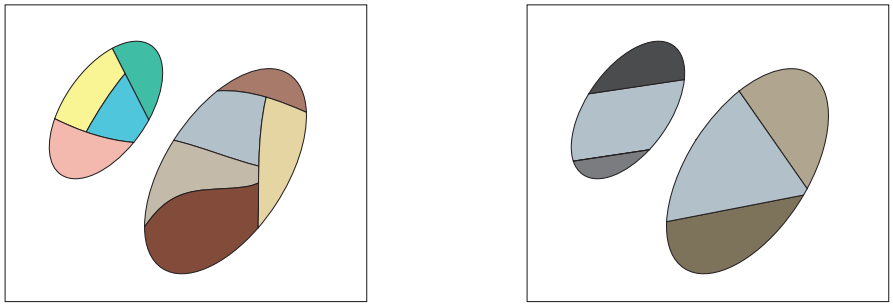
Similar to the indiscernibility relation, another equivalence relation can be defined such that only a single probe function $\phi_i \in \mathcal{B}$ is required for equivalence.

Definition 8. Weak Perceptual Indiscernibility Relation [25, 54]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system, and let $\phi_i \in \mathbb{F}$. Then, the* weak perceptual indiscernibility relation $\simeq_{\phi_i}$ *is defined as follows:*

$$\simeq_{\phi_i} = \{(x, y) \in O \times O : \exists \phi_i \in \mathbb{F} \centerdot \phi_i(x) = \phi_i(y)\}.$$

The weak indiscernibility relation can provide new information or relationships about a set of objects for a given application. For instance, consider a scenario where near set theory is applied to an existing problem or process where objects are already being associated with feature values; examples include a problem already formulated in terms of near sets without using the weak perceptual indiscernibility relation, or problems in other areas such as pattern recognition or image analysis. In other words, a set of probe functions, $\mathcal{B}$, has already been selected to solve the problem. In such a scenario, the perceptual indiscernibility relation could produce a quotient set as given in Fig. 1a (where each colour represents a different class), indicating these two sets of objects, representing some perceptual information in the original problem domain, are not related to each other. However, selecting a single $\phi_i \in \mathcal{B}$ could produce the equivalence classes shown in Fig. 1b, where it is now apparent that there is some relationship between these two sets of objects. Also, as described in [24], Definition 8 can be used to discover similarities in a manner similar

to humans, namely, we identify similarity between objects using only a subset of all the possible features associated with an object.



(a) (b)

Figure 1: Example demonstrating the practical application of Definition 8. (a) Quotient set created using Definition 5 showing no relationship between the two sets, and (b) quotient set created using Definition 8 showing a relationship between the two sets.

2.1.3 Near Sets and the Nearness Relation

Definition 5 provides the framework for comparisons of sets of objects by introducing a concept of nearness within a perceptual system. Sets can be considered near each other when they have "things" in common. In the context of near sets, the "things" can be quantified by objects or equivalence classes. The simplest example of nearness between sets sharing "things" in common is the case when two sets have indiscernible elements. This idea leads to the definition of a weak nearness relation.

Definition 9. Weak Nearness Relation [25]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X, Y \subseteq O$. A set X is weakly near to a set Y within the perceptual system $\langle O, \mathbb{F} \rangle$ $(X \underset{\mathbb{F}}{\bowtie} Y)$ iff there are $x \in X$ and $y \in Y$ and there is $\phi_i \in \mathbb{F}$ such that $x \simeq_{\mathcal{B}} y$. In the case where sets X, Y are defined within the context of a perceptual system, then X, Y are weakly near each other.*

An example of disjoint sets that are weakly near each other is given in Fig. 2a, where each colour represents an equivalence class. These sets are weakly near each other since both sets share objects belonging to the same equivalence class. As a practical example of weakly near sets, consider a database of images where each image is described by some

feature vector, *i.e.* the images are considered perceptual objects and the feature vectors are the object descriptions. Examples of features are the values of different colour models [55] or moments [56]. In this case, two disjoint sets of images are weakly near each other if each set contains one or more images with descriptions that match an image in the other set.

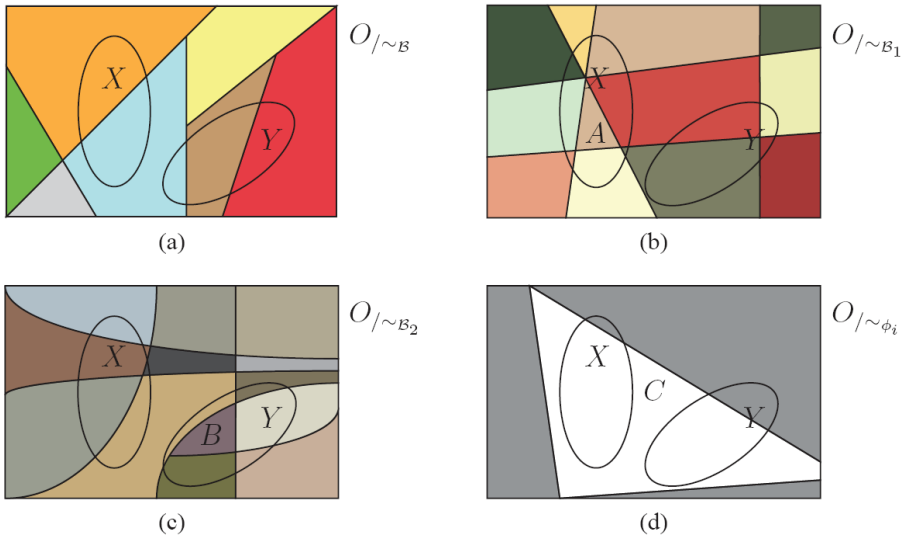


Figure 2: Examples of Definitions 9 & 10, where each colour represents an equivalence class. (a) Example of Definition 9, (b) example of $O_{/\sim_{\mathcal{B}_1}}$, (c) example of $O_{/\sim_{\mathcal{B}_2}}$, and (d) example of $O_{/\sim_{\phi_i}}$ showing (together with (b) and (c)) that sets X and Y are near to each other according to Definition 10.

Next, the notion of nearness in Definition 9 can be strengthened by considering equivalence classes rather than objects which is the case in the following definition.

Definition 10. Nearness Relation [25]. *Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $X, Y \subseteq O$. A set X is near to a set Y within the perceptual system $\langle O, \mathbb{F}\rangle (X \bowtie_{\mathbb{F}} Y)$ iff there are $\mathcal{B}_1, \mathcal{B}_2, \subseteq \mathbb{F}$ and $\phi_i \in \mathbb{F}$ and there are $A \in O_{/\sim_{\mathcal{B}_1}}, B \in O_{/\sim_{\mathcal{B}_2}}, C \in O_{/\sim_{\phi_i}}$ such that $A \subseteq X, B \subseteq Y$, and $A, B \subseteq C$. If a perceptual system is understood, than a set X is near to a set Y.*

The concept of the nearness relation can be further explained as follows. First, recall that within a perceptual system there is a set of probe functions, $\mathbb{F}$, where each probe function describes the objects in a different manner. Further, each set in the family of subsets of $\mathbb{F}$ (*i.e.* each $\mathcal{B} \in \mathbb{F}$) produces different partitions of the sets X and Y, where each partition

presents different perceptual information. In other words, the selection of $\mathcal{B} \in \mathbb{F}$ constrains our ability to describe the objects in the sets X and Y in same manner as if one were told they could only describe fruit by colour or by shape, *etc*. Consequently, the sets X and Y are near each other if there are three subsets of $\mathbb{F}$ that respectively produce an equivalence class in X and Y that are subsets of an equivalence class that covers both the sets X and Y. As an intuitive example, let O be the set of fruit found in the produce section of the local grocery store, let X be the set of apples and bananas, and let Y be the set of strawberries and pears. Selecting $\mathbb{F}_1$ as a single probe function identifying the shape of the fruit, $\mathbb{F}_2$ as a single probe function identifying "bite sized" fruit, and f as a single probe function identifying fruit colour, gives an equivalence class containing round apples from X, an equivalence class containing strawberries from Y, and an equivalence class containing both the red apples from X and the red strawberries from Y. Thus, using Definition 10 the sets X and Y are near each other. Furthermore, notice that, in this example, the sets X and Y are weakly near each other using only the probe function f. This suggests that one approach to determine if two sets are near each other would be first to partition the sets using a set of probe functions, and then use Definition 8 to "cycle" through the probe functions in $\mathcal{B}$ looking for a partition that is a superset of the equivalence classes from the two sets. Also, Definition 10 does not put any restriction on the sets $\mathbb{F}_1$ and $\mathbb{F}_2$ being disjoint. As a result, these sets could share probe functions or even be the same. Lastly, a visualization of Definition 10 is also given in Fig. 2. Sets X and Y are near to each other in Fig.'s 2b-2d, since the sets $A \in O_{/\sim_{\mathbb{F}_1}}$ in Fig. 2b and $B \in O_{/\sim_{\mathbb{F}_2}}$ in Fig. 2c are subsets of $C \in O_{/\sim_f}$ given in Fig. 2d.

Next, as given in the following definition, sets X, Y are near sets if they satisfy the nearness relation.

Definition 11. Perceptual Near Sets [25] *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system, let $X, Y \subseteq O$ denote disjoint sets. Sets X, Y are near sets iff $X \bowtie_{\mathbb{F}} Y$.*

Lastly for completeness, a formalization of identifying similarities among objects is given by way of Definition 12 that is a principle for determining the nearness of objects.

Definition 12. Nearness Description Principle (NDP) [21,25]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $x, y \in O$. Objects $x,\ y$ are* perceptually near each other *within $\langle O, \mathbb{F} \rangle$ (or, more concisely,* near each other*), if and only if $\{x\} \bowtie_{\mathbb{F}} \{y\}$. In other words, objects $x,\ y$ are* near each other within $\langle O, \mathbb{F} \rangle$ *if and only if there exists $\phi \in \mathbb{F}$ such that $x \sim_\phi y$, i.e., objects $x,\ y$ are indiscernible with respect to the family of probe functions of the perceptual system $\langle O, \mathbb{F} \rangle$.*

Note, as mentioned in [25], the nearness of objects is always with respect to Definition 9 (without the adjective weak) since Definition 10 would require 1-element equivalence classes, a case which cannot be guaranteed.

2.2 Approximate Nearest Neighbours

Given a set of points $P = \{p_1, \ldots, p_n\}$ in an d-dimensional vector space X and a query point $q \in X$, the nearest neighbour search problem is defined as finding the point in P that is closest to q [57]. This problem arises in many research areas, especially in computer vision, and for high dimensional data, there is no known algorithm that performs much better than a linear search of the data points in P [57]. As a result, α-approximate[2] nearest neighbour searching has been introduced where query times can be reduced by orders of magnitude while sill achieving near-optimal accuracy. An α-approximate nearest neighbour to a query point $q \in X$ is defined as $p \in X$ if $dist(p, q) \leq (1 + \alpha) dist(p^*, q)$ where p^* is the true nearest neighbour [57].

Although there are a number of recent contributions to the fast calculation of approximate nearest neighbours [58], the results presented in this thesis were obtained using the Fast Library for Approximate Nearest Neighbours (FLANN) [59], since it is a library easily added to any C++ program, and because of the option for automatic optimization. FLANN uses two data structures to efficiently perform approximate nearest neighbour searches, namely, the randomized kd-tree algorithm and the hierarchical k-means tree algorithm [57].

[2]Note: the symbol α is being used instead of ϵ (as is traditional in the literature) to avoid confusion with the tolerance relation introduced below.

A kd-tree organizes the data using a binary tree where the tree nodes are points from P. Since points belong to a d-dimensional vector space, each node must have an associated splitting dimension (*i.e.* a dimension used to divide subsequent nodes in the tree). The next data point added to the tree is assigned to either the left or right child node depending on whether its value in the splitting dimension is less than or greater than the value of the current node. The kd-tree algorithm used in FLANN is called randomized because the splitting dimension for each node is selected randomly from the first D dimensions that have the greatest variance [57]. The other data structure used is the hierarchical k-means tree. This structure is created by recursion, *i.e.* the set of data is partitioned into K regions using the k-means clustering algorithm and then each region is again partitioned into K regions *etc.* The recursion is terminated when there are less than K data points in a region [57].

FLANN is the ideal library for performing approximate nearest neighbour searching because of the option for automatic optimization. The choice of algorithm used for approximate nearest neighbour searching is highly dependent on the dataset [57]. Consequently, the FLANN library has an option to select automatically the search algorithm and to optimize the input parameters of the selected algorithm. Both options are based on the points in P. Optimization is guided by a set of parameters specified by the user in the following equation

$$\text{cost} = \frac{s + w_b b}{(s + w_b b)_{\text{opt}}} + w_m m,$$

where s is the search time for the number of vectors in the sample dataset, b is the build time, $m = m_t/m_d$ is the ratio of memory used for the tree and memory used to store the data, w_b is the importance of build time over search time, and w_m is the importance of memory overhead [57]. Setting $w_b = 0$ means that the fastest search time is desired, and similarly, setting $w_m = 0$ means that faster search time is more important than memory requirements. Additionally, optimization is also performed based on the desired precision (percentage of query points for which the correct nearest neighbour is found) of the results from a nearest neighbour search (see [57] for more details). To generate the results presented here, a target precision of 0.8 was used together with $w_b = w_m = 0$.

2.3 Image Processing

The application of near set theory demonstrated in this thesis is in the area of image correspondence and content-based image retrieval. Briefly, the problem of image correspondence is the unaided process of assessing the degree in which one image resembles another. Similarly, content-based image retrieval is the problem of retrieving images from a database based on the content of the image, *e.g.* colour, shapes, texture, objects *etc.*, rather than on some semantic description or set of key words associated with the image [60]. Each of these applications deal with assessing the similarity of images, which proves to be a natural arena for near set theory. As will be described latter, the approach is to consider portions of the images as perceptual objects and to use image processing techniques as probe functions. Consequently, the following sections describe image processing techniques that were used to generate the results presented in this thesis.

2.3.1 Normalized RGB

The normalized RGB values is a feature described in [61], and the formula is given by

$$N_X = \frac{X}{R_T + G_T + B_T},$$

where the values $R_T, G_T,$ and B_T are respectively the sum of R, G, B components of the pixels in each subimage, and $X \in [R_T, G_T, B_T]$.

2.3.2 Entropy

Shannon introduced entropy (also called information content) as a measure of the amount of information gained by receiving a message from a finite codebook of messages (see [62] for a comprehensive presentation of entropy). The idea was that the gain of information from a single message is proportional to the probability of receiving the message. Thus, receiving a message that is highly unlikely gives more information about the system than a message with a high probability of transmission. Formally, let the probability of receiving

a message i of n messages be p_i, then the information gain of a message can be written as

$$\Delta I = \log(1/p_i) = -\log(p_i), \tag{1}$$

and the entropy of the system is the expected value of the gain and is calculated as

$$H = -\sum_{i=1}^{n} p_i \log(p_i).$$

Work in [63,64] shows that Shannon's definition of entropy has some limitations. Shannon's definition of entropy suffers from the following problems: it is undefined when $p_i = 0$; in practise, the information gain tends to lie at the limits of the interval $[0, 1]$; and statistically speaking, a better measure of ignorance is 1 - p_i rather than $1/p_i$ [63]. As a result, a new definition of entropy can be defined with the following desirable properties:

P1: $\Delta I(p_i)$ is defined at all points in $[0, 1]$.

P2: $\lim_{p_i \to 0} \Delta I(p_i) = \Delta I(p_i = 0) = k_1, k_1 > 0$ and finite.

P3: $\lim_{p_i \to 1} \Delta I(p_i) = \Delta I(p_i = 1) = k_2, k_2 > 0$ and finite.

P4: $k_2 < k_1$.

P5: With increase in p_i, $\Delta I(p_i)$ decreases exponentially.

P6: $\Delta I(p)$ and H, the entropy, are continuous for $0 \leq p \leq 1$.

P7: H is maximum when all p_i's are equal, *i.e.* $H(p_1, \ldots, p_n) \leq H(1/n, \ldots, 1/n)$.

With these in mind, [63] defines the gain in information from an event as

$$\Delta I(p_i) = e^{(1-p_i)},$$

which gives a new measure of entropy as

$$H = \sum_{i=1}^{n} p_i e^{(1-p_i)}.$$

2.3.3 Mean Shift Segmentation Algorithm

Image segmentation is the process of partitioning an image into regions such that each region represents a perceptual object within the image. The mean shift algorithm (introduced in [65]) segments an image using kernel density estimation, a nonparametric technique for estimating the density of a random variable. Nonparametric techniques are characterized by their lack of assumptions about the density and differ from parametric techniques which assume a parametric form of a given density and then estimate parameters that describe the density, such as mean or variance [53]. The estimate of the density is calculated from the number of observations within a volume in d-dimensional space centred on $\mathbf{x}$, and a kernel that weights the importance of the observations [53]. Formally, given n observations of a random variable $\mathbf{X} \in \mathbb{R}^d$, the kernel density estimate of the pdf of $\mathbf{X}$ is given by

$$f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i), \tag{2}$$

where

$$K_{\mathbf{H}}(\mathbf{x}) = \mid \mathbf{H} \mid^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x}),$$

the matrix $\mathbf{H}$ is called the bandwidth, and the function $K(\cdot)$ is the kernel used to perform the estimation. The kernel defines how the observed data points influence the estimate. For example, all data within the volume contribute equally using the uniform kernel, whereas the Gaussian kernel gives more weight to the observations closest to $\mathbf{x}$. Often kernels are specified using profile notation written as

$$K(\mathbf{x}) = c_k(\| \mathbf{x} \|^2), \tag{3}$$

where c is an constant ensuring that the kernel integrates to one [65]. As an example, the Epanechnikov kernel is given as

$$K_E(\mathbf{x}) = \begin{cases} ck_E(\|\mathbf{x}\|), & \|\mathbf{x}\| \leq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where

$$k_E(x) = \begin{cases} 1 - x^2, & 0 \leq x \leq 1, \\ , 0, & x > 1. \end{cases}$$

Similarly, the bandwidth is used to define the size of the d-dimensional volume around $\mathbf{x}$ for which the observations, $\mathbf{x}_i$, are included in the estimate of $f(\mathbf{x})$. Common choices for the bandwidth include

$$\mathbf{H} = \textbf{diag}[h_1^2, \ldots, h_d^2],$$

where each component of the vector is assigned a separate area of influence, and

$$\mathbf{H} = h^2\mathbf{I},$$

where each component has the same area of influence [65]. Using the latter approach and the kernel profile notation given in Eq. 3, Eq. 2 can be rewritten as

$$\hat{f}_K(\mathbf{x}) = \frac{c_k}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{5}$$

As was mentioned, the main idea behind this algorithm is finding the modes of the density from observations in the form of an image. These modes lie at the zeros of the gradient. The gradient of a function $f(\mathbf{x})$ is defined as

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n} \right).$$

The zeros of the gradient $\nabla \hat{f}(\mathbf{x}) = \mathbf{0}$ can be found by exploiting the linearity of Eq. 5 [1] giving

$$\hat{\nabla} f_K(\mathbf{x}) \equiv \nabla \hat{f}_K(\mathbf{x}) = \frac{2c_k}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) k' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\| \right).$$

In other words, the density gradient estimate $\hat{\nabla} f_K(\mathbf{x})$ is calculated as the gradient of the density estimate $\hat{f}_K(\mathbf{x})$. Next, a new kernel $G(\cdot)$ is defined as

$$G(\mathbf{x}) = c_g g(\|\mathbf{x}\|^2), \tag{6}$$

where

$$g(x) = -k'(x)$$

yields

$$\begin{aligned} \hat{\nabla} f_K(\mathbf{x}) &= \frac{2c_k}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) g \left(\left\| \frac{\mathbf{x} - \mathbf{x_i}}{h} \right\|^2 \right), \\ &= \frac{2c_k}{nh^{d+2}} \left[\sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \cdot \\ &\quad \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right] \end{aligned} \tag{7}$$

which is at the heart of the mean shift algorithm. The second term in Eq. 7 is given by

$$\mathbf{m}_G(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x},$$

and is the mean shift vector that always points in the direction of maximum increase in density [65]. This can be seen by defining a new density estimate using the kernel in Eq. 6 as

$$\hat{f}_G(\mathbf{x}) = \frac{c_g}{nh^d} \sum_{i=1}^{n} g \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right).$$

Then,

$$\hat{\nabla} f_K(\mathbf{x}) = \hat{f}_G(\mathbf{x}) \frac{2c_k}{h^2 c_g} \mathbf{m}_G(\mathbf{x}),$$

which yields

$$\mathbf{m}_G(\mathbf{x}) = \frac{1}{2} h^2 c \frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})}, \tag{8}$$

where $c = c_g/c_k$. Thus, as was mentioned, Eq. 8 shows that the mean shift is a normalized vector pointing in the direction of the maximum increase increase in the density at location $\mathbf{x}$ [65]. The segmentations used in this thesis were created using an implementation of Eq. 8 called EDISON [66], a system for which both the source code and binaries are freely available online. A sample segmentation produced by the EDISON system is given in Fig. 3. Finally, note, the choice of h (actually h_s and h_r) used to generate the segmentations in this thesis was selected based on trial and error using the EDISON system, *i.e.*, the values were selected by experimenting on a few sample images before segmenting the entire database.

(a) (b)

Figure 3: Example demonstrating the mean shift segmentation algorithm [65]. (a) Sample image, and (b) Segmentation of (a) using the EDISON system [66].

2.3.4 Multiscale Edge Detection

Mallat's multiscale edge detection method uses Wavelet theory to find edges in an image [67, 68]. Edges are located at points of sharp variation in pixel intensity. These points can be identified by calculating the gradient of a smoothed image (*i.e.* an image that has

been blurred). Then, edge pixels are defined as those that have locally maximal gradient magnitudes in the direction of the gradient. Formally, define a 2-D smoothing function $\theta(x, y)$ such that its integral over x and y is equal to 1, and converges to 0 at infinity. Using the smoothing function, one can define the functions

$$\psi^1(x, y) = \frac{\partial\theta(x, y)}{\partial x} \quad \text{and} \quad \psi^2(x, y) = \frac{\partial\theta(x, y)}{\partial y},$$

which are, in fact, wavelets given the properties of $\theta(x, y)$ mentioned above. Next, the dilation of a function by a scaling factor s is defined as

$$\xi_s(x, y) = \frac{1}{s^2}\xi(\frac{x}{s}, \frac{y}{s}).$$

Thus, the dilation by s of ψ^1, and ψ^2 is given by

$$\psi^1_s(x, y) = \frac{1}{s^2}\psi^1(\frac{x}{s}, \frac{y}{s}) \quad \text{and} \quad \psi^2_s(x, y) = \frac{1}{s^2}\psi^2(\frac{x}{s}, \frac{y}{s}).$$

Using these definitions, the wavelet transform of $f(x, y) \in L^2(\mathbb{R}^2)$ at the scale s is given by

$$W^1_s f(x, y) = f * \psi^1_s(x, y),$$

and

$$W^2_s f(x, y) = f * \psi^2_s(x, y),$$

which can also be written as

$$\begin{pmatrix} W^1_s f(x, y) \\ W^2_s f(x, y) \end{pmatrix} = s \begin{pmatrix} \frac{\partial}{\partial x}(f * \theta_s)(x, y) \\ \frac{\partial}{\partial y}(f * \theta_s)(x, y) \end{pmatrix} = s\nabla(f * \theta_s)(x, y).$$

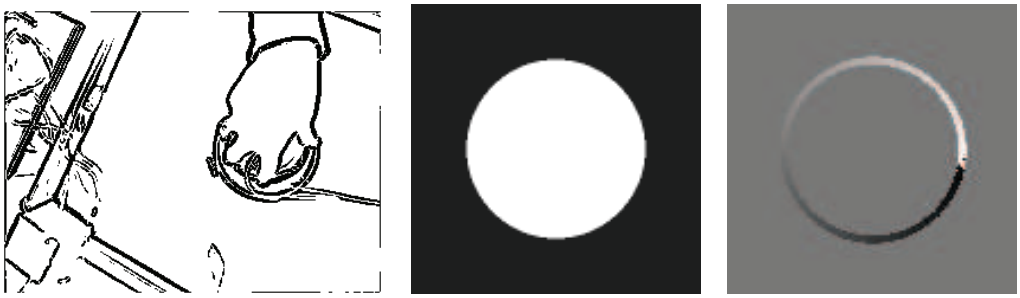
Next, the modulus and angle of the gradient vector are defined respectively as

$$M_s f(x, y) = \sqrt{|W^1_s f(x, y)|^2 + |W^2_s f(x, y)|^2},$$

and

$$A_s f(x,y) = \tan^{-1}(W_s^2 f(x,y)/W_s^1 f(x,y)).$$

These equations can be used to detect an edge and calculate its orientation. Edge pixels are those belonging to the modulus maximum, defined as pixels with modulus greater than the two neighbours in the direction indicated by $A_s f(x,y)$, and the orientation of an edge pixel is simply given by the angle of the gradient (see [67] for specific implementation details). Examples of my own implementation of Mallatt's edge detection and edge orientation methods are given in Fig. 4.



(a) (b)

Figure 4: (a) Example demonstrating implementation of Mallat's multiscale edge detection method [67]. (b) Example of finding edge orientation using the same method. White represents 0 radians and black 2π radians.

2.3.5 Normalized Probabilistic Rand Index

The normalized probabilistic rand index (introduced in [69] and summarized here) is a non-parametric technique for evaluating the performance of an image segmentation algorithm. This index is a supervised technique since evaluation is performed with respect to ground truth images. A supervised measure provides a nice benchmark for testing purposes, since human perceptual grouping is inherent to the evaluation of the segmentation due to the use of ground truth images. Furthermore, the normalized probabilistic index was selected due to its use of multiple ground truth images when evaluating a proposed segmentation, thus taking into account multiple perceptual sources.

The NPR index has its roots in the rand index, a measure developed based on the idea of counting pixel pairs that have the same segmentation labels. First, define an image $X =$

$\{x_1, \ldots, x_N\}$ of N pixels and two segmentations of X, S and S', where each segmentation respectively assigns labels l_i and l'_i to the pixels in X. Then the rand index is given as

$$R(S, S') = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i \neq j}} \Big[\Pi\Big(l_i = l_j \wedge l'_i = l'_j\Big) + \Pi\Big(l_i \neq l_j \wedge l'_i \neq l'_j\Big) \Big],$$

where Π is the identity function and $\binom{N}{2}$ is the number of unique pixel pairs in X. Next, the rand index was extended to allow for the use of more ground truth images in the evaluation of a proposed segmentation. The idea is that observing the same pixel pair in each ground truth image is considered a Bernoulli trial with the two outcomes being either they have the same label or they do not. Then, the set of observations across all ground truth segmentations forms a Bernoulli distribution with expected value p_{ij}. Thus, given a set of manual segmentations $\{S_1, \ldots, S_K\}$, a segment for evaluation S_{test}, and a label of x_i denoted as l_i^S where S denotes the segment used to label x_i, the probabilistic rand index is defined as

$$PR(S_{\text{test}}, \{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})],$$

where

$$c_{ij} = \Pi(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}).$$

Finally, the normalized probabilistic rand index extends the probabilistic rand index by normalizing with respect to its baseline. The selected baseline is the expected value of probabilistic rand index. Consequently, the NPR index is defined as

$$\text{NPR} = \frac{\text{PR} - \mathbb{E}[\text{PR}]}{\max(\text{PR}) - \mathbb{E}[\text{PR}]},$$

where the maximum value is taken to be 1 and the expected value of the PR index is given

as

$$\begin{aligned}\mathbb{E}\Big[PR(S_{\text{test}},\{S_k\})\Big] =& \frac{1}{\binom{N}{2}}\sum_{\substack{i,j\\ i<j}}\Big\{\mathbb{E}\Big[\Pi(l_i^{S_{\text{test}}}=l_j^{S_{\text{test}}})\Big]p_{ij}\\ &+\mathbb{E}\Big[\Pi(l_i^{S_{\text{test}}}\neq l_j^{S_{\text{test}}}\Big](1-p_{ij})\Big\},\\ =& \frac{1}{\binom{N}{2}}\sum_{\substack{i,j\\ i<j}}\Big[p'_{ij}p_{ij}+(1-p'_{ij})(1-p_{ij})\Big],\end{aligned}$$

where $p'_{ij}=\mathbb{E}[\Pi(l_i^{S_{\text{test}}}=l_j^{S_{\text{test}}}]$. To make the baseline representative of perceptually consistent groupings, p'_{ij} is estimated from segmentations of all images for all unordered pairs. In other words, given Φ as the number of images in the database used for testing, p'_{ij} is defined as

$$p'_{ij}=\frac{1}{\Phi}\sum_{\phi}\frac{1}{K_\phi}\sum_{k=1}^{K_\phi}\Pi(l_i^{S_k^\phi}=l_j^{S_k^\phi}).$$

2.3.6 Grey Level Co-occurrence Matrices

Image texture is an important part of perceiving images. Texture is difficult to describe, and is generally associated with a region of the image, rather than restricted to a specific pixel. Generally, there are statistical and structural approaches to identifying texture [70]. The textural features used in this thesis are based on second order measures, as reported in [71–73], where the approach is considered second-order, since the measures are not derived directly from the pixel values themselves, but rather on statistics generated from relationships between groups of two pixels given by a grey-level co-occurrence matrix. In other words, the features are based on the average spatial relationship between pixel values [71].

In general, the grey level co-occurrence matrix is defined with respect to the angle and distance between pixel pairs. However, to keep things simple, the grey level co-occurrence matrix will first be defined with respect to horizontally adjacent pixels, which corresponds to an angle of 0° and a distance $d=1$ in the traditional literature. Using the notation given in [71], let $L_x=\{1,2,\ldots,N_x\}$ and $L_y=\{1,2,\ldots,N_y\}$ respectively denote the

horizontal and vertical spatial domains of a grey level image quantized to N_g levels, *i.e.* the grey levels in an image are in the set $G = \{0, 1, \ldots, N_g - 1\}$. Then, $L_y \times L_x$ is the set of all pixel coordinates belonging to an image I, where $I : L_y \times L_x \rightarrow G$, and the grey level co-occurrence matrix is given as

$$P(i,j) = |\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) : \\ m - k = 0, n - l = 1, I(k,l) = i, I(m,n) = j\}|. \quad (9)$$

For clarity, an example of Eq. 9 is given graphically in Fig. 5. One can add the degree and distance to Eq. 9, by the following simple modification,

$$P(i,j,d,0^\circ) = |\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) : \\ m - k = 0, |n - l| = d, I(k,l) = i, I(m,n) = j\}|.$$

For angles $45^\circ, 90^\circ,$ and 135°, see [71]. Finally, the following textural features can be derived from the grey level co-occurrence matrix,

Maximum Probability	$\max_{i,j}(p_{ij})$,
Contrast	$\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 p_{ij}$,
Uniformity (also called Energy)	$\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p_{ij}^2$, and
Homogeneity	$\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{p_{ij}}{1+\lvert i-j \rvert}$,

where $p_{ij} = P(i,j)$ divided by the sum of the elements in P. In brief, the maximum probability returns the strongest response of P, contrast measures the intensity contrast between a pixel and its neighbour, uniformity is the angular second moment, and homogeneity measures the spatial closeness of the distribution of elements in P to the diagonal (see [74] for further details).

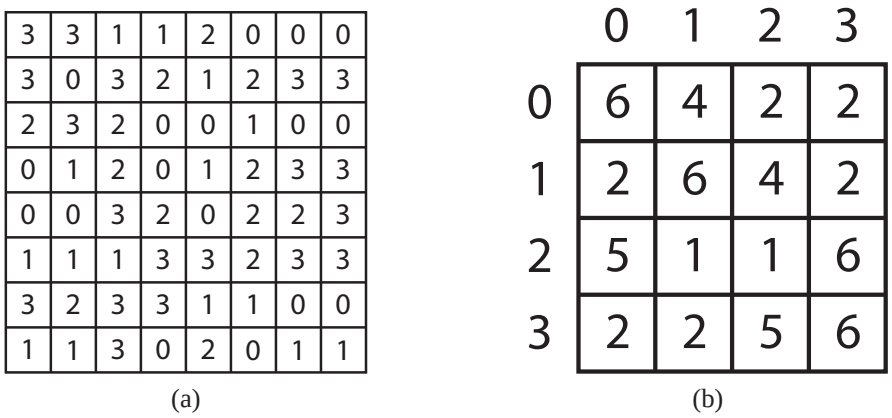


Figure 5: Example demonstrating the creation of a grey level co-occurrence matrix. (a) Quantized image, and (b) grey level co-occurrence matrix of 0° and $d = 1$.

2.3.7 Zernike Moments

There are many approaches to analyzing shapes contained in images. For example, researchers in the computer vision community at Brown University are, to say the least, working on areas such as measuring the similarity between two shapes, modelling shapes from a pattern theory perspective, shape representation and perceptual grouping [75]. Another entire research area is that of the statistical theory of shape pioneered by David Kendall [76]. However, in this thesis, Zernike moments are used to provide region-based descriptors of an image that are invariant with respect to rotation and reflections, where the notion of the term "moment" referred to here is that of the general theory of moments which appears in areas of mathematics, physics, and statistics [56]. Moreover, a small set of Zernike moments can characterize the global shape of a pattern effectively, where the lower order moments represent the global shape, and the higher order moments represent the detail [77–79].

As given in [56], for a continuous image function $f(x, y)$, the Zernike moment of order n with repetition m is defined as

$$A_{nm} = \int\int_D f(x, y) V^*_{nm}(x, y) dx dy, \tag{10}$$

where the double integral is defined over the unit disk $D = \{(x, y) : x^2 + y^2 \leq 1\}$, n is a non-negative integer, and m is an integer that makes result of $n - |m|$ even and non-negative.

In Eq. 10, $V_{nm}(x, y)$ is a Zernike function defined as

$$V_{nm}(x, y) = R_{nm}(\rho)e^{jm\theta},$$

where $\rho = \sqrt{x^2 + y^2}$, $\theta = \tan^{-1}(y/x)$, and the radial Zernike polynomial $R_{nm}(\rho)$ is defined by

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)! p^{n-2s}}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!}.$$

As explained in [56], Eq. 10 cannot be applied directly to digital images. Consequently, a mapping of the digital image must occur. Let $F(i, j), i = 1, \ldots, N, j = 1, \ldots, N$ denote an $N \times N$ image, then $F(i, j)$ can be mapped onto a function $f(x_i, y_i)$ defined on $[-1, 1]^2$ according to

$$f(x_i, y_i) = F(i, j), i = 1, \ldots, N, j = 1, \ldots, N,$$

where $x_i = (2i - N - 1)/N$ and $y_j = (2j - N - 1)/N$. Note, it can be assumed, without loss of generality, that $f(x_i, y_i)$ is a function with all its pixels inside the unit circle [56]. Moreover, since the image is not analog, but actually a discrete function, the following approximation can be used to calculate the Zernike moments from sampled data

$$\widetilde{A}_{nm} = \sum_i \sum_j w_{nm}(x_i, y_j) f(x_i, y_j), \tag{11}$$

where i and j are taken such that $(x_i, y_j) \in D$,

$$w_{nm}(x_i, y_j) = \int_{x_i - \frac{\Delta}{2}}^{x_i + \frac{\Delta}{2}} \int_{y_j - \frac{\Delta}{2}}^{y_j + \frac{\Delta}{2}} V^*_{nm}(x, y) dx dy,$$

and $\Delta = 2/N$ is the pixel width/height. Finally, $w_{nm}(x_i, y_j)$ can be approximated by a simple one-point numerical integration formula

$$w_{nm}(x_i, y_j) \approx \Delta^2 V^*_{nm}(x_i, y_j). \tag{12}$$

Note, it was shown in [56] that using Eq. 11 & 12 is a highly inaccurate approach to computing Zernike moments due to both the geometric error caused by the difference between the total area covered by the pixels in Eq. 11 and the actual area of the unit circle, as well as the error due to the approximation of $w_{nm}(x_i, y_j)$ in Eq. 12. Instead, a method for calculating Zernike moments in polar coordinates (rather than the Cartesian method given above) is given that eliminates the previously mentioned errors. Nevertheless, Eq. 11 & 12 were still used to generate rotationally invariant features due to the following reasons. First, only low order moments were used (*e.g.* $n \leq 4$), and evidence in [56] demonstrated that the results of using only low orders of Zernike moments produced magnitudes with acceptable level of errors, both in comparisons of the magnitudes on a constant image and for use in reconstructing images. Also, others have reported success using low order Zernike moments for content-based image retrieval (see, *e.g.* [80, 81]), and implementation of Eq. 11 & 12 is simple and fast.

2.3.8 CIELUV Colour Space

The CIE 1976 $L^*u^*v^*$ Colour Space (also written CIELUV) is a uniform colour space where the Euclidean distances between points in the space is proportional to human perception of differences in colour [82]. In contrast, the RGB colour space represents a non-uniform space with respect to the human visual system. The L^*u*v* colour components are given (in terms of the XYZ colour components) by the following equations [83]:

$$\begin{aligned} L^* &= 116\Big(\frac{Y}{Y_n}\Big)^{1/3} - 16, \Big(\frac{Y}{Y_n}\Big) > 0.008856, \\ L^* &= 903.3\Big(\frac{Y}{Y_n}\Big), \Big(\frac{Y}{Y_n}\Big) \leq 0.008856, \\ u^* &= 13L^*(u' - u'_n), \\ v^* &= 13 * L^*(v' - v'_n), \end{aligned}$$

where

$$u' = 4X/(X + 15Y + 3Z), \qquad u'_n = 4X_n/(X_n + 15Y_n + 3Z_n),$$
$$v' = 9Y/(X + 15Y + 3Z), \qquad v'_n = 9Y_n/(X_n + 15Y_n + 3Z_n),$$

and Y_n, X_n, and Z_n are based on the reference white point. For the results presented in this thesis, the D50 reference white point was used giving values of $Y_n = 1, X_n = 0.964221$, and $Z_n = 0.825211$. Similarly, the XYZ colour components can be calculated using

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.006 & 1.116 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

3 Tolerance Near Sets

Disjoint sets containing objects with similar descriptions are near sets. Similarity is determined quantitatively via some description of the objects. Near set theory provides a formal basis for identifying, comparing, and measuring resemblance of objects based on their descriptions, *i.e.* based on the features that describe the objects. The discovery of near sets begins with identifying feature vectors for describing and discerning affinities between sample objects. Objects that have, in some degree, affinities in their features are considered *perceptually near* each other. Groups of these objects, extracted from the disjoint sets, provide information and reveal patterns of interest.

Tolerance near sets are near sets[3] defined by a description-based tolerance relation. Tolerance relations provide a view of the world without transitivity [84]. Consequently, tolerance near sets provide a formal foundation for *almost solutions*, solutions that are valid within some approximation, which is required for real world problems and applications [84]. In other words, tolerance near sets provide a basis for a quantitative approach for evaluating the similarity of objects without requiring object descriptions to be exact.

Sossinsky addresses the question of "Why Tolerance?", *i.e.*, why consider the tolerance relation at all [84]. One answer, which is the main focus of this thesis, is that practical applications deal with imprecise data and that solutions only need to be accurate to a degree, *i.e.* to within some tolerance. Other answers to the question of "why tolerance?" (outside the scope of this thesis) consider the fact that tolerance fits quite nicely in other areas of mathematics, and that tolerance can be convenient for many existing mathematical studies.

3.1 History

The idea of tolerance first surfaced in Poincaré's work in 1905 in which he reflects on experiments performed by Ernst Weber in 1834, and Gustav Fechner's insight in 1850 [84–87]. Weber (Fechner's doctoral supervisor) was interested in the sense of touch and kinethesis

[3] See Section 2.1 for a comprehensive introduction to near set theory.

and was the first to discover that the sense of touch consisted of pressure, temperature, and pain [87]. Moreover, Weber performed experiments to determine our ability to perceive two pin points on the surface of our skin. He would blindfold a subject and use a compass, with increasing distances between the pin points, to determine the threshold at which a person can perceive the points as two. He labelled this point as the two-point threshold. Weber discovered that our skin is partitioned into regions based on the domain of sensory nerves, and we perceive the pin points as one if they fall into the same region. For instance, he found the sensitivity to be the highest on the tip of the tongue, followed by the volar side of the finger tip. In contrast, the least sensitive part was the upper part of the spine and the middle of the upper arm [88]. While Weber was not investigating the concept of tolerance, this idea can be easily identified in his experiments. For instance, his results indicate that there is a specific tolerance on the distance between two points under which they are perceived as one (*i.e.* indistinguishable), which varies with the density of nerves in our skin. If for a specific region on the skin this distance was labelled ε, then any distance less than ε would be classified in the same category, namely a single pin prick.

Both Weber and Fechner were also interested in our ability to detect the difference in a pair of weights [87]. Specifically, Weber performed two types of experiments, either the weights were placed on a subject's hands or a subject was asked to lift the weights. In each case a standard weight was applied to one hand, and the subject was asked to determine if the unknown weight was heavier, the same, or lighter than the standard. The results of these experiments led Weber to introduce a term called *just noticeable differences*, the point at which a person is able to perceive the difference in a stimulus presented to the senses, and to introduce a law (labelled Weber's Law by Fechner) stating that detection of a stimulus is directly dependent on the amount of stimulus already present [88]. For example, one must speak with greater volume when having a conversation in a noisy room versus a quiet room. Fechner in his interest in the relationship between the physical world and our sensations, built on Weber's law [86]. He realized that just noticeable differences represented a psychological difference and developed a law that states that sensation is

proportional to the logarithm of the stimulus.

Poincaré's work on tolerance was inspired by Fechner, but the key difference is Poincaré's work marked a shift from stimuli and sensations to an abstraction in terms of sets together with an implicit idea of tolerance. Although the general idea of tolerance is present in an essay titled *Space and the Senses* published in 1913 [89], the concept of a tolerance space is directly implied in a discussion on Fechner's law (with respect to the weight experiments) in the book *Science and Hypothesis* [85], a fact identified by Sossinsky when he states Poincaré "discerned" a tolerance space but did not write out the mathematics [84]. A tolerance space $\langle X, \xi \rangle$ consists of a set X and a binary relation ξ on X ($\xi \subset X \times X$) that is reflexive (for all $x \in X$, $x \xi x$) and symmetric (for all $x, y \in X$, if $x \xi y$, then $y \xi x$) but transitivity of ξ is not required [84, 90]. The idea of a tolerance space is apparent when Poincaré states:

> It has, for instance, been observed that a weight A of 10 grammes and a weight B of 11 grammes produced identical sensations, that the weight B could no longer be distinguished from a weight C of 12 grammes, but that the weight A was readily distinguished from the weight C. Thus the rough results of the experiments may be expressed by the following relations: $A = B, B = C, A < C$, which may be regarded as the formula of the physical continuum. But here is an intolerable disagreement with the law of contradiction, and the necessity of banishing this disagreement has compelled us to invent the mathematical continuum. We are therefore forced to conclude that this notion has been created entirely by the mind, but it is experiment that has provided the opportunity. We cannot believe that two quantities which are equal to a third are not equal to one another, and we are thus led to suppose that A is different from B, and B from C, and that if we have not been aware of this, it is due to the imperfections of our senses [85].

By separating the three weights into two sets $\{w_{10}, w_{11}\}$ and $\{w_{11}, w_{12}\}$, Poincaré has implicitly identified a tolerance space $\langle W, \simeq_{wt,\varepsilon} \rangle$, where $wt : W \to \mathbb{R}$ and for a sensation sensitivity threshold ε (implicit in what Poincare writes), a tolerance relation can be written as

$$\simeq_{wt,\varepsilon} = \{(x, y) \in W \times W : \| \, wt(x) - wt(y) \, \|_1 \leq \varepsilon\},$$

where $\| \cdot \|_1$ is the L_1 norm.

Next, the idea of tolerance is formally introduced by Zeeman [90] with respect to the

brain and visual perception. Zeeman makes the observation that a single eye cannot identify a 2D Euclidean space because the Euclidean plane has an infinite number of points. Instead, we see things only within a certain tolerance. Of particular importance is the first formal definition of a tolerance space and the tolerance relation. Continuing, Sossinsky [84] presents homology and homotopy theories on tolerance spaces and gives practical applications. Finally, the contributions of this thesis involve tolerance near sets introduced by Peters [36, 91], which combines near set theory (see, *e.g.*, Section 2.1) with the ideas of tolerance spaces and relations.

4 Perceptual Systems[4]

A logical starting point for a discussion on near set theory begins with establishing a basis for describing elements of sets. All sets in near set theory consist of perceptual objects.

Definition 13. Perceptual Object. *A* perceptual object *is something perceivable that has its origin in the physical world.*

A perceptual object is anything in the physical world with characteristics observable to the senses such that they can be measured and are knowable to the mind. Examples of perceptual objects include patients, components belonging to a manufacturing process, and camera images. Here, the term *perception* is considered relative to measurable characteristics called the object's features.

In keeping with the approach to pattern recognition suggested by M. Pavel [52], the features of an object are quantified by probe functions.

Definition 14. Probe Function [21, 40]. *A* probe function *is a real-valued function representing a feature of a perceptual object.*

In this work, probe functions are defined in terms of digital images such as: colour, texture, contour, spatial orientation, and length of line segments along a bounded region. In the

[4]The theory presented in this Section is a reproduction of Section 2.1.1 and is reproduced here in the interest of clarity for those who skip directly to this section.

context of near set theory, objects in our visual field are always presented with respect to the selected probe functions. Moreover, it is the probe functions that are used to measure characteristics of visual objects and similarities among perceptual objects, making it possible to determine if two objects are associated with the same pattern without necessarily specifying which pattern (as is the case when performing classification).

Next, a perceptual system is a set of perceptual objects, together with a set of probe functions.

Definition 15. Perceptual System [25]. *A* perceptual system $\langle O, \mathbb{F} \rangle$ *consists of a non-empty set* O *of sample perceptual objects and a non-empty set* $\mathbb{F}$ *of real-valued functions* $\phi \in \mathbb{F}$ *such that* $\phi : O \to \mathbb{R}$.

The notion of a perceptual system admits a wide variety of different interpretations that result from the selection of sample perceptual objects contained in a particular sample space O. Two examples of perceptual systems are: a set of images together with a set of image processing probe functions, or a set of results from a web query together with some measures (probe functions) indicating, *e.g.,* relevancy or distance (*i.e.* geographical or conceptual distance) between web sources.

Combining Definitions 13 & 14, the description of a perceptual object within a perceptual system can be defined as follows.

Definition 16. Object Description. *Let* $\langle O, \mathbb{F} \rangle$ *be a perceptual system, and let* $\mathcal{B} \subseteq \mathbb{F}$ *be a set of probe functions. Then, the* description *of a perceptual object* $x \in O$ *is a feature vector given by*

$$\boldsymbol{\phi}_{\mathcal{B}}(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_l(x)),$$

where l *is the length of the vector* $\boldsymbol{\phi}_{\mathcal{B}}$*, and each* $\phi_i(x)$ *in* $\boldsymbol{\phi}_{\mathcal{B}}(x)$ *is a probe function value that is part of the description of the object* $x \in O$.

Note, the idea of a feature space is implicitly introduced along with the definition of object description. An object description is the same as a feature vector as described in traditional pattern classification [53]. The description of an object can be considered a point

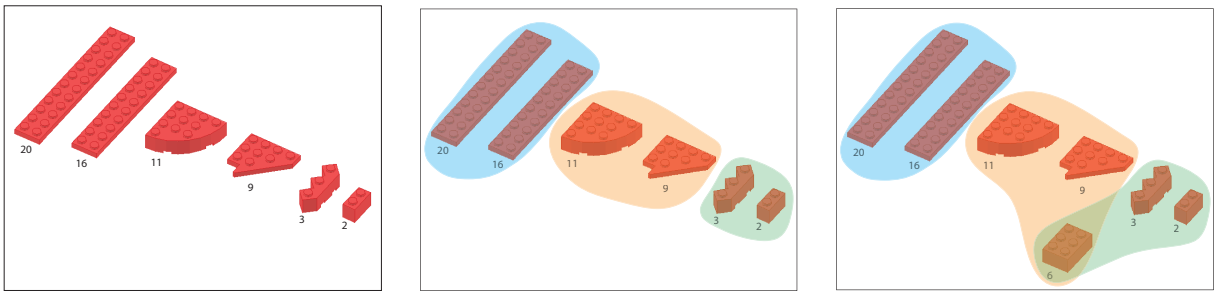
in an l-dimensional Euclidean space $\mathbb{R}^l$ called a feature space. As was mentioned in the introduction, near set theory is concerned with the nearness of objects based on their descriptions. Thus, the relationship between objects is discovered in a feature space that is determined by the probe functions in $\mathcal{B}$.

4.1 Perceptual Tolerance Relation

The introduction to a tolerance view of near sets grew out of a need for a relation less restrictive than the equivalence condition of the indiscernibility relation to facilitate observation of similar objects and associations in a perceptual system. Specifically, this approach is useful in real world problems (especially when dealing with components in images) where a probe function value for two objects perceived to be "the same" is rarely an exact match. As a simple example, consider Fig. 6 along with a probe function that returns the number of pegs on each block. Using the indiscernibility relation on these blocks produces six different classes, each containing only one object, and no new information is revealed. However, allowing classes to be formed where the number of pegs on each block in the class must be within five of all the other blocks produces the sets of objects shown in Fig. 6b. The result is three classes of objects that present perceptual information about the relationship of these blocks to one another (with respect to the selected probe function) that was not present using the indiscernibility relation. Namely, these sets now represent the concept of small, medium, and large blocks. Moreover, by relaxing the equivalence relation, objects can belong to more than one class as shown in Fig. 6c. This matches human descriptions of objects where, for example, one could say "in terms of the number of pegs, the block with six pegs is kind of small," which is reflected by the block belonging to both the medium and small classes.

As was mentioned in Section 3.1, a tolerance space can be defined as follows.

Definition 17. Tolerance Space [84, 90] *Let O be a set of sample perceptual objects, and let ξ be a binary relation (called a tolerance relation) on X ($\xi \subset X \times X$) that is reflexive*



(a) (b) (c)

Figure 6: Example showing the need to relax the equivalence condition of Definition 5. (a) Set of six objects together with the number of pegs of each object, (b) classes formed by grouping objects where the difference in the number of pegs is less than five, and (c) example demonstrating that objects can belong to more than one class when the equivalence relation is relaxed.

(for all $x \in X$, $x \xi x$) and symmetric (for all $x, y \in X$, if $x \xi y$, then $y \xi x$) but transitivity of ξ is not required. Then a tolerance space is defined as $\langle X, \xi \rangle$.

Thus, a specific tolerance relation is given in Definition 18.

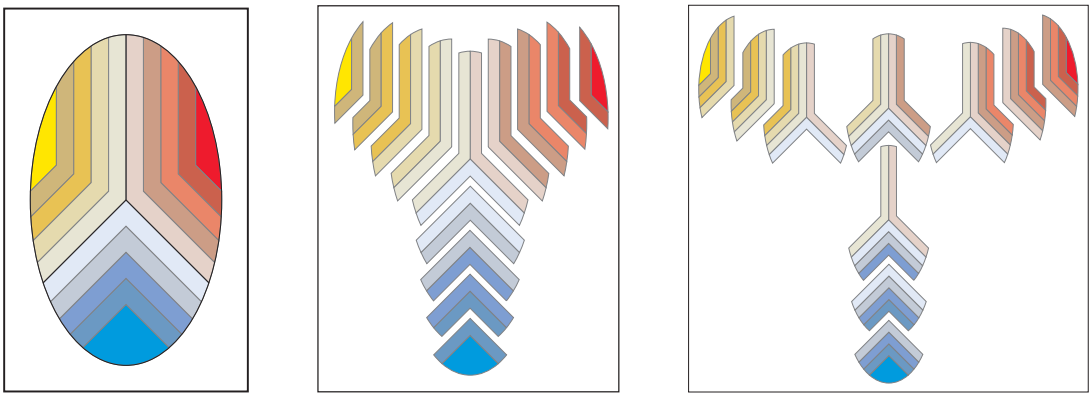
Definition 18. Perceptual Tolerance Relation [36, 91] (see [33, 39] for applications). *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \mathbb{R}$. For every $\mathcal{B} \subseteq \mathbb{F}$, the* perceptual tolerance relation $\cong_{\mathcal{B},\epsilon}$ *is defined as follows:*

$$\cong_{\mathcal{B},\epsilon} = \{(x, y) \in O \times O : \| \boldsymbol{\phi}(x) - \boldsymbol{\phi}(y) \|_2 \leq \varepsilon\},$$

where $\| \cdot \|_2$ is the L^2 norm. For notational convenience, this relation is written $\cong_{\mathcal{B}}$ instead of $\cong_{\mathcal{B},\varepsilon}$ with the understanding that ε is inherent to the definition of the tolerance relation.

Notice the relation $\cong_{\mathcal{B},\epsilon}$ is defined with respect to the description of a pair of objects, *i.e.*, objects resemble each other when the feature vectors used to describe them are within some epsilon value. Furthermore, the perceptual tolerance relation differs from the indiscernibility relation (see Section 2.1) by allowing $\| \cdot \|_2 \leq \varepsilon$ instead of requiring $\| \cdot \|_2 = 0$. In fact, Definition 18 is a generalization of the indiscernibility relation, a special case occurring for $\varepsilon = 0$, a fact that is, in part, highlighted in the following simple examples on the effect of the perceptual tolerance relation.

As defined in Section 2.1, an equivalence class is a set of objects that satisfy the indiscernibility relation, *i.e.* the descriptions of objects in an equivalence class are all the same. A visualization of a set of equivalence classes is given in Fig. 7a where the oval represents a set of objects and each colour represents an equivalence class. Moreover, in this figure, the position of class is also relevant, which is not always the case. Here, the distance between the object descriptions in feature space increases with the distance between classes in the image. In accordance, Fig. 7b & 7c represents the classes that result from a low and high value of epsilon, which respectively correspond to the object descriptions of two adjacent equivalence classes satisfying Definition 18, and the object descriptions of four adjacent classes satisfying Definition 18. Observe that low values of epsilon tend to produce a large number of small classes, and high value of epsilon tend to produce a small number of large classes.



(a) (b) (c)

Figure 7: Example highlighting the effect of the perceptual tolerance relation. (a) A set of objects represented by the oval, each colour represents an equivalence class (see Section 2.1), and the distance between the object descriptions in feature space increases with the distance between classes (b) the classes produced by a "low" value of epsilon such that the descriptions of two adjacent equivalence classes satisfy Definition 18, and (c) the classes produced by a "high" value of epsilon such that the descriptions of four adjacent equivalence classes satisfy Definition 18.

The next example demonstrates the effect of the perceptual tolerance relation on real data. Consider Table 1 that contains 20 objects with $|\phi(x_i)| = 1$. Letting $\varepsilon = 0.1$ gives the

Table 1: Tolerance Class Example

x_i	$\phi(x)$	x_i	$\phi(x)$	x_i	$\phi(x)$	x_i	$\phi(x)$
x_1	.4518	x_6	.6943	x_{11}	.4002	x_{16}	.6079
x_2	.9166	x_7	.9246	x_{12}	.1910	x_{17}	.1869
x_3	.1398	x_8	.3537	x_{13}	.7476	x_{18}	.8489
x_4	.7972	x_9	.4722	x_{14}	.4990	x_{19}	.9170
x_5	.6281	x_{10}	.4523	x_{15}	.6289	x_{20}	.7143

following classes:

$$\{\{x_1, x_8, x_{10}, x_{11}\}, \{x_1, x_9, x_{10}, x_{11}, x_{14}\},$$
$$\{x_2, x_7, x_{18}, x_{19}\},$$
$$\{x_3, x_{12}, x_{17}\},$$
$$\{x_4, x_{13}, x_{20}\}, \{x_4, x_{18}\},$$
$$\{x_5, x_6, x_{15}, x_{16}\}, \{x_5, x_6, x_{15}, x_{20}\},$$
$$\{x_6, x_{13}, x_{20}\}\}$$

Observe that each pair of objects in each of the above classes satisfies the condition $\| \phi(x) - \phi(y) \|_2 \leq \varepsilon$, and that almost all of the objects appear in more than one class. Moreover, there would be twenty classes, if the indiscernibility relation were used since there are no two objects with matching descriptions.

In each of the previous examples, there was a need for structures that correspond to the equivalence class under the tolerance relation, *i.e.* there is a need to define a method by which objects are grouped together when transitivity no longer applies. In an equivalence class, an object is added to a class if its description matches the description of the objects already in the class, which by definition are all the same. However, the lack of transitivity gives rise to the two very different classes given in the following definitions.

Definition 19. Neighbourhood. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $x \in O$. For a*

set $\mathcal{B} \subseteq \mathbb{F}$ *and* $\varepsilon \in \mathbb{R}$*, a* neighbourhood *is defined as*

$$N(x) = \{y \in O : x \cong_{\mathcal{B},\epsilon} y\}.$$

An example of a neighbourhood in a 2D feature space is given in Fig. 8, where the position of all the objects are given by the numbers 1 to 21, and the neighbourhood is defined with respect to the object labelled 1. Notice that the distance between all the objects and object 1 is less than or equal to $\epsilon = 0.1$, but that not all the pairs of objects in the neighbourhood of x satisfy the tolerance relation.

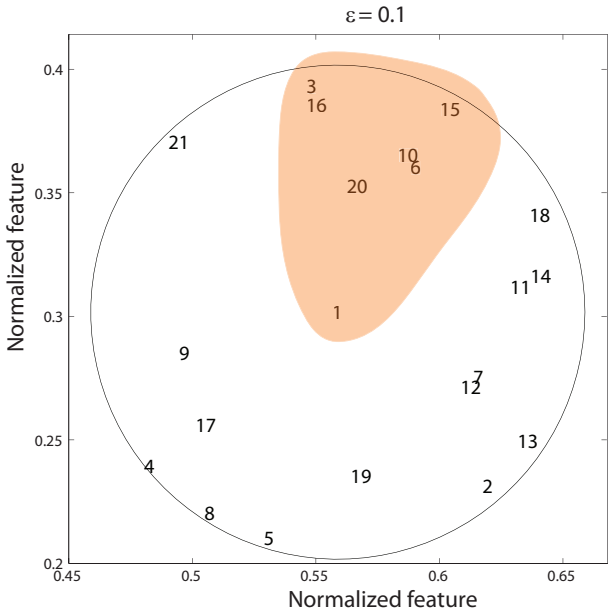


Figure 8: Example demonstrating the difference between a neighbourhood and a tolerance class in 2 dimensional feature space. The neighbourhood is all the objects within the circle, and the tolerance class is shown in orange.

In contrast, all the pairs of objects within a pre-class must satisfy the tolerance relation as given in the next definitions.

Definition 20. Pre-Class. *Let* $\langle O, \mathbb{F}\rangle$ *be a perceptual system. For* $\mathcal{B} \subseteq \mathbb{F}$ *and* $\varepsilon \in \mathbb{R}$*, a set* $X \subseteq O$ *is a* pre-class *iff* $x \cong_{\mathcal{B},\epsilon} y$ *for any pair* $x, y \in X$.

Definition 21. Tolerance Class. *A maximal pre-class with respect to inclusion is called a* tolerance class.

An example of a tolerance class is given by the set of objects coloured orange in Fig. 8 since no object can be added to the set and still satisfy the condition that any pair $x, y \in X$ must be within ϵ of each other. Also, the example given in Fig. 7 was created with tolerance classes in mind, and the classes formed from the data in Table 1 are clearly tolerance classes.

As was mentioned in the introduction to this section, objects can belong to more than one tolerance class. Consequently, the following notation is required to differentiate between classes and facilitate discussions in subsequent sections. The set of all tolerance classes using only the objects in O is given by $H_{\cong_{\mathcal{B},\varepsilon}}(O)$ (also called the cover of O), a single tolerance class is represented by $C \in H_{\cong_{\mathcal{B},\varepsilon}}(O)$, and the set of all tolerance classes containing an object x is denoted by $C_x \subset H_{\cong_{\mathcal{B},\varepsilon}}(O)$.

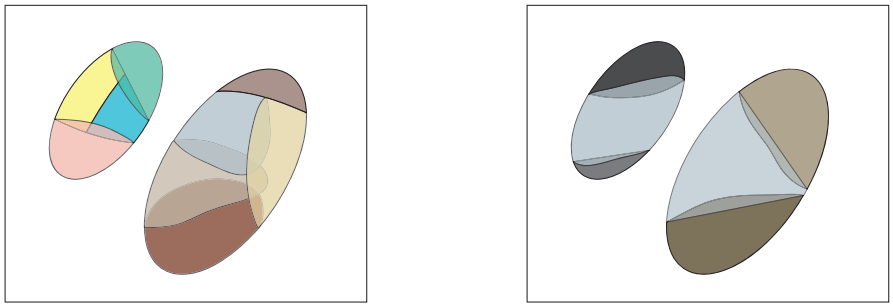
Finally, this section is concluded by introducing another tolerance relation similar to the weak indiscernibility relation given in Section 2.1, where resemblance is defined with respect to a single probe function.

Definition 22. Weak Perceptual Tolerance Relation *[24] Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \mathbb{R}$, $\phi_i \in \mathbb{F}$. Then, the* weak perceptual tolerance relation $\approx_{\mathcal{B},\epsilon}$ *is defined as follows:*

$$\approx_{\mathcal{B},\epsilon} = \{(x, y) \in O \times O : \exists \phi_i \in \mathbb{F} \; . \; |\phi_i(x) - \phi_i(y)| \leq \varepsilon\}.$$

The weak tolerance relation can provide new information or relationships about a set of objects for a given application. For instance, consider a scenario where near set theory is applied to an existing problem or process where objects are already being associated with feature values; examples include a problem already formulated in terms of near sets without using the weak perceptual tolerance relation, or problems in other areas such as pattern recognition or image analysis. In other words, a set of probe functions, $\mathcal{B}$, has already been selected to solve the problem. In such a scenario, the perceptual tolerance relation could produce a covering as given in Fig. 9a (where each colour represents a difference tolerance

class), indicating these two sets of objects, representing some perceptual information in the original problem domain, are not related to each other. However, selecting a single $\phi_i \in \mathcal{B}$ could produce the tolerance classes shown in Fig. 9b where it is now apparent that there is some relationship between these two sets of objects.



(a) (b)

Figure 9: Example demonstrating the practical application of Definition 22. (a) Covering of two sets showing no relationship between, and (b) covering created using Definition 22 showing a relationship between the two sets.

4.2 Tolerance Near Sets

The term *relationship* was mentioned in the explanation of Fig. 9; however, a definition on which the relationship is based was not given (although it may have been clear from the context). Recall that sets of objects that have similar descriptions are called near sets, and a method for determining similarity was provided by way of the perceptual tolerance relation (and to a lesser degree with the weak perceptual tolerance relation). Consequently, the following two definitions enunciate the fundamental notion of nearness between two sets and provide the foundation of the results presented in this thesis.

Definition 23. Tolerance Nearness Relation [36, 91]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X, Y \subseteq O, \varepsilon \in \mathbb{R}$. A set X is near to a set Y within the perceptual system $\langle O, \mathbb{F} \rangle$ $(X \underset{=\!=\mathbb{F}}{\bowtie} Y)$ iff there exists $x \in X$ and $y \in Y$ and there is $\mathcal{B} \subseteq \mathbb{F}$ such that $x \cong_{\mathcal{B},\epsilon} y$.*

Definition 24. Tolerance Near Sets [36, 91]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $\varepsilon \in \mathbb{R}, \mathcal{B} \subseteq \mathbb{F}$. Further, let $X, Y \subseteq O$, denote disjoint sets with coverings determined by the tolerance relation $\cong_{\mathcal{B},\epsilon}$, and let $H_{\cong_{\mathcal{B},\varepsilon}}(X), H_{\cong_{\mathcal{B},\varepsilon}}(Y)$ denote the set of tolerance classes*

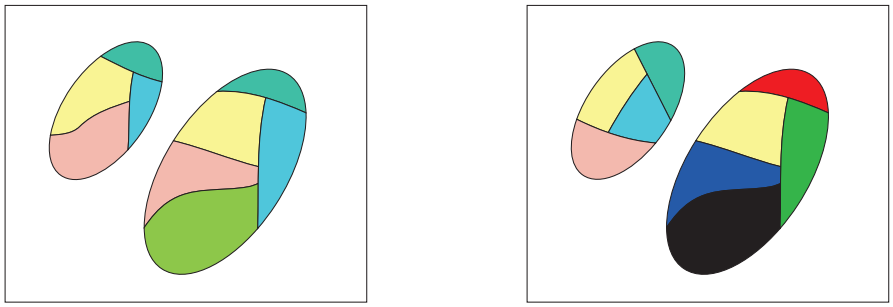
for X, Y*, respectively. Sets* X, Y *are* tolerance near sets *iff there are tolerance classes* $A \in H_{\cong_{\mathcal{B},\varepsilon}}(X), B \in H_{\cong_{\mathcal{B},\varepsilon}}(Y)$ *such that* $A \underline{\bowtie}_{\mathbb{F}} B$.

Observe that two sets $X, Y \subseteq O$ are tolerance near sets, if they satisfy the tolerance nearness relation. Also, notice that Tolerance near sets are a variation of the original definition of near sets using the indiscernibility relation [21]. Moreover, the original definition of tolerance near sets given in [36, 91] defines nearness in terms of pre-classes (as opposed to tolerance classes as given in Definition 24), however the results presented in this thesis are obtained using tolerance classes, and so the definition was adjusted accordingly. Finally, an example of tolerance near sets is given in Fig. 9b, where the colours represent different tolerance classes, and classes with the same colour represent the situation where $A \underline{\bowtie}_{\mathbb{F}} B$.

4.3 Nearness Measure

The nearness measure was created out of a need to determine the degree that near sets resemble each other, a need which arose during the application of near set theory to the practical applications of image correspondence and content-based image retrieval. Specifically, the nearness measure was introduced by Henry and Peters in [92]. At the same time, the nearness measure was also introduced by Henry and Peters in [33, Section VII.A, pp. 964-965] where it was given as a solution to the problem of image resemblance of MRI images. Since then, the notation of the nearness measure has been refined (as reported in [93]) and it has been applied to the problems of image resemblance and correspondence [36–38, 91, 94–96] which is closely related to content-based image retrieval [93], *i.e.* the problem of retrieving images based on the perceived objects within the image rather than based on semantic terms associated with the image. The nearness measure has also been applied to patterns of observed swarm behaviour stored in tables called ethograms, where the goal is to measure the resemblance between the behaviours of different swarms [51]. In each of these applications, the nearness measure has been applied to problems formulated either in terms of near sets and the indiscernibility relation or as tolerance near sets and the perceptual tolerance relation.

The catalyst for the creation of the nearness measure was the observation that under Definition 24 disjoint sets are either considered near, or not, *i.e.* there is no middle ground. However, the practical application of near set theory requires a method for quantifying the nearness of sets, as in, for example, retrieving images from a database that are similar to a query image. In this case, a positive measure of similarity points to the resemblance between a query image and the retrieved images. In essence, the nearness measure presents a systematic approach to determining the degree of similarity between a pair of disjoint sets, an idea that can be visualized by asking "which pair of sets in Fig. 10 are more similar?"



(a) (b)

Figure 10: Example of degree of nearness between two sets, where each colour corresponds to an equivalence class. (a) High degree of nearness, and (b) low degree of nearness.

The nearness measure was first proposed in working with the indiscernibility relation and equivalence classes. The approach was that the degree of nearness of sets in a perceptual system is determined by the cardinalities of the equivalence classes that have the same description (an idea that is visualized in Fig. 11). For sets that are considered "more similar" as in Fig. 10a, there should be more pairs of equivalence classes (from the respective sets) that have matching descriptions. Consequently, the nearness measure is determined by counting the number of objects in equivalence classes that have matching descriptions. Thus, the sets in Fig. 10a are closer (more near) to each other in terms of their descriptions than the sets in Fig. 10b. Moreover, this notion can be generalized to tolerance classes as is the case in the following definition.

Definition 25. Nearness Measure [33,93]. *Let $\langle O, \mathbb{F} \rangle$ be a perceptual system, with $\varepsilon \in \mathbb{R}$, and $\mathcal{B} \subseteq \mathbb{F}$. Furthermore, let X and Y be two disjoint sets and let $Z = X \cup Y$. Then a*

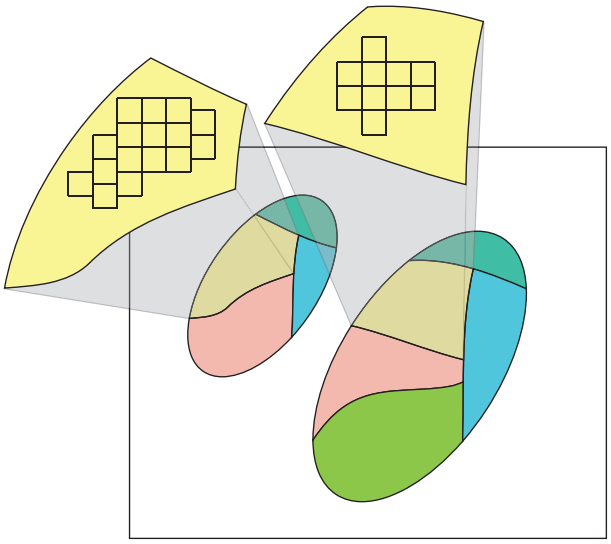


Figure 11: Visualization of nearness measure based on equivalence classes and the indiscernibility relation. Similar images should produce equivalence classes that are evenly divided between X and Y. This is measured by counting the number of objects that belong to sets X and Y for each equivalence class, and comparing the results.

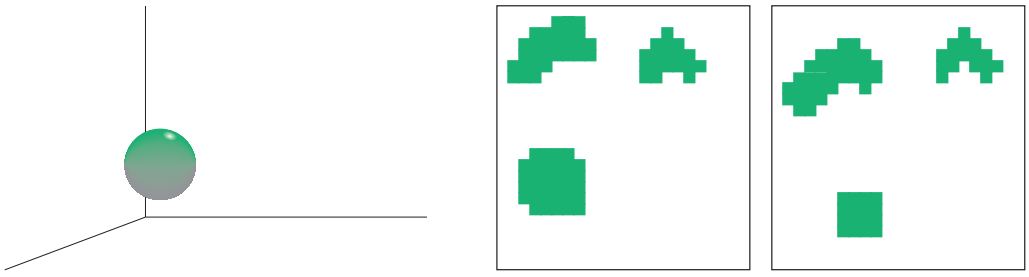
nearness measure *between two sets is given by*

$$tNM_{\cong_{\mathcal{B},\varepsilon}}(X,Y) = \left(\sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C| \right)^{-1} \cdot \sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C| \frac{\min(|C \cap X|, |[C \cap Y|)}{\max(|C \cap X|, |C \cap Y|)}.$$

As was explained, the idea behind Definition 25 is that similar sets should produce equivalence classes with matching descriptions. However, the addition of the perceptual tolerance relations subtly adds to the complexity of calculating the measure. The main idea stays the same, namely, similar sets should produce classes that are evenly divided between the two sets X and Y. It is the approach to calculating the measure that is important with the addition of the tolerance relation. For instance, using the indiscernibility relation it is simply a matter of determining the equivalence classes of objects in both sets and then comparing the description of each equivalence class in set X to the description of each equivalence class in set Y. In contrast, the process of calculating the measure under the perceptual tolerance relation involves first finding the tolerance classes of all objects in the union of X and Y. This approach is best because of the fact that all objects within a tolerance class must satisfy the tolerance relation. Because of this fact, a comparison

of two tolerance classes cannot be made directly without comparing all the objects in one class with all the objects in the other class. As a result, a more efficient approach is to find the tolerance classes of the union of X and Y, and then determine which portion of each tolerance class (form the covering of Z) belongs to X and Y, which is why C is intersected with X and Y in above equation.

In any event, the measure is calculated by counting the number of objects that belong to sets X and Y for each tolerance class, and then comparing these counts as a proper fraction (guaranteed by the min and max functions). Then, the final value of the measure is simply a weighted average of all the fractions. A weighted average was selected to give preference to larger tolerance classes with the idea that a larger tolerance class contains more perceptually relevant information. Calculating the proper fraction for a single tolerance class C is shown graphically in Fig. 12, where Fig. 12a is a single sample tolerance class in a 3D feature space, and Fig. 12b contains two disjoint sets of objects, where the objects are represented by small square blocks laid out in rows and columns, and only the members of the tolerance class in Fig. 12a are shown in green. Observe that a tolerance class in feature space can be distributed throughout the sets, and that the nearness measure would compare the number of objects from the tolerance class in set X to the number of objects from the tolerance class in set Y. In this case, the ratio would be close to 1 because the number of objects in both sets X and Y are nearly the same.



(a) (b)

Figure 12: Example relating tolerance class objects to their coordinates within a pair of images. (a) Tolerance class in 3 dimensional feature space. (b) Two disjoint sets of objects, where the objects are represented by small square blocks laid out in rows and columns, and only the members of the tolerance class in Fig. 12a are shown in green.

The nearness measure produces values in the interval $[0, 1]$, where, for a pair of sets

X, Y, a value of 0 represents no resemblance in terms of the probe functions in $\mathcal{B}$, and a value of 1 indicates the sets X, Y completely resemble each other, a fact that can be seen by calculating the nearness measure on a single set, *i.e.* $tNM_{\cong_{\mathcal{B},\varepsilon}}(X, X) = 1$. In addition, the nearness measure provides a methodical approach for determining whether two sets are tolerance near sets as seen in the following proposition and its corollary.

Proposition 1. *A nearness measure of* $tNM_{\cong_{\mathcal{B},\varepsilon}}(X, Y) = 0$ *indicates that the sets* X, Y *are not tolerance near sets.*

Proof. A $tNM_{\cong_{\mathcal{B},\varepsilon}}(X, Y) = 0$ is produced by either the empty set (*i.e.* $Z = X \cup Y = \emptyset$) or each tolerance class in $H_{\cong_{\mathcal{B},\varepsilon}}(Z)$ is either completely a subset of X or completely a subset of Y. In other words, there is no tolerance class that consists of objects from both X and Y. Consequently, there is no $x \in X$ and $y \in Y$ such that $x \cong_{\mathcal{B},\epsilon} y$, and so, by Definitions 23 & 24, the sets X and Y are not tolerance near sets. □

Corollary 1. *A nearness measure of* $tNM_{\cong_{\mathcal{B},\varepsilon}}(X, Y) > 0$ *indicates that the sets* X, Y *are tolerance near sets.*

Proof. A $tNM_{\cong_{\mathcal{B},\varepsilon}}(X, Y) > 0$ is produced, if there is at least one tolerance class $C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)$ such that $X \cap C \neq \emptyset$ and $Y \cap C \neq \emptyset$. Consequently, there must be $x \in X$ and $y \in Y$ that satisfies $x \cong_{\mathcal{B},\epsilon} y$, and so, by Definitions 23 & 24, the sets X and Y are tolerance near sets. □

Next, a concrete example of calculating the nearness measure is given with the aid of Fig. 13. Assume each of Fig. 13a - 13e is a set of objects, where each square represents 100 objects, and the different colours correspond to tolerance classes (with each object belonging to only one class). Thus, Fig. 13a consists of 400 objects distributed into three classes of 200, 100, and 100 objects, respectively. Further, assume all pairs of objects taken from two different classes that share the same colour will satisfy the perceptual tolerance relation[5]. The results of comparing the set in Fig. 13a with itself and then the remaining

[5]Generally, the approach is not to compare separate coverings (as in this example), but to calculate the nearness measure based on the covering obtained from the union of the sets being measured. However, the example was presented in this manner in the interest of clarity.

four other sets is given in the caption of Fig. 13. Notice the nearness measure ranges from 1, produced by calculating the measure on a single set, to 0, produced by sets that are completely dissimilar. For the most part, the results of calculating the nearness measure in this example matches intuition. For example, Fig. 13a & 13b differ only by the lower left tolerance class, thus one would naturally come to the conclusion that these two sets are 75% similar. In contrast, when considering Fig. 13a & 13c, one might venture that half of objects of each set belong to classes that are shared by both sets. Consequently, the nearness measure should be 0.5. However, this does not accurately reflect the perceptual information contained in each of these classes. For instance, using the data given in Table 2, the tolerance nearness measure between Fig.'s 13a & 13b is calculated as

$$tNM_{\cong_{\mathcal{B},\varepsilon}}(\text{8a}, \text{8b}) = \frac{1}{800}\Big(1 \cdot 400 + 1 \cdot 200\Big) = 0.75,$$

and the nearness measure between Fig.'s 13a & 13c is calculated as

$$tNM_{\cong_{\mathcal{B},\varepsilon}}(\text{8a}, \text{8c}) = \frac{1}{800}\Big(0.5 \cdot 300 + 1 \cdot 200\Big) = 0.4375.$$

Observe, $tNM_{\cong_{\mathcal{B},\varepsilon}}(\text{8a}, \text{8c}) = 0.4375$ since the black tolerance class consisting of 200 objects represents half of the perceptual information of the set in Fig. 13c (and a quarter of the total perceptual information in both sets), yet it is not reflected at all by the set in Fig. 13a. Furthermore, the grey tolerance class (consisting of objects from both classes) represents roughly 38% of the perceptual information in both sets, yet only 1/3 of the class is common to both sets. Consequently, while half of each set shares the same label, less than half of the perceptual information in the union of these two sets is common to both, a fact that is reflected in the nearness measure.

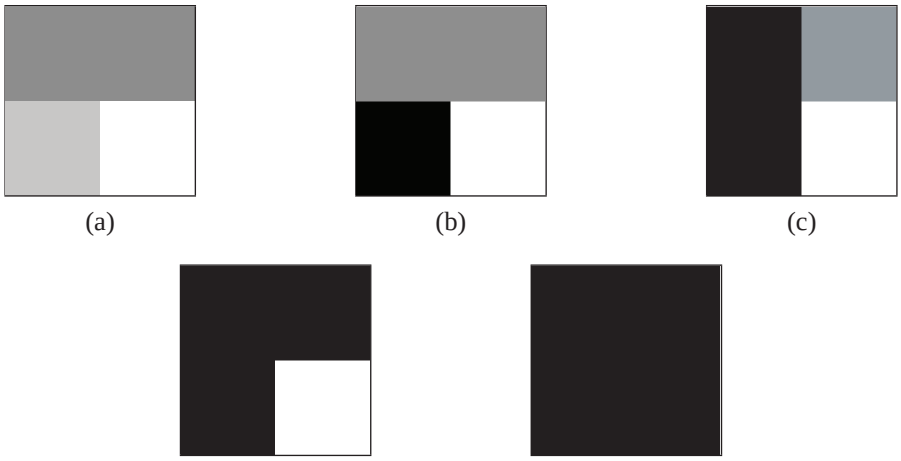


Figure 13: Example of calculating $tNM_{\cong_{\mathcal{B},\varepsilon}}(X,Y)$ by comparing the set in (a) with itself and the remaining four. $tNM_{\cong_{\mathcal{B},\varepsilon}}(X,Y) = \{1, 0.75, 0.4375, 0.25, 0\}$ for comparing set (a) with itself and with the sets (b), (c), (d), and (e).

Table 2: $NM_{\cong_{\mathcal{B}}}$ Calculation Example

Covering	Tolerance Class	TC Size	Object in X	Objects in Y	TC Ratio
		400	200	200	1
		100	100	0	0
		100	0	100	0
		200	100	100	1
		300	200	100	0.5
		100	100	0	0
		200	0	200	0
		200	100	100	1

4.4 Finding classes

The practical application of the nearness measure rests on the ability to efficiently find all the classes for a set $Z = X \cup Y$. In the case where $\varepsilon = 0$, the process is straightforward, *i.e.*, the first object is assigned to a tolerance class (which is an equivalence class since, $\varepsilon = 0$), then the description of each subsequent object is compared to objects in each existing tolerance class. If a given object's description does not match any of the descriptions of the existing tolerance classes, then a new class is created. Thus, the algorithm runtime ranges from order $O(|Z|^2)$ in the worst case, which occurs when none of the object descriptions match, to $O(|Z|)$, which occurs when all the object descriptions are equivalent. In practise,

the runtime is somewhere between these two extremes.

The approach to finding tolerance classes in the case where $\varepsilon \neq 0$ is based on the observations presented in the following Propositions.

Proposition 2. *All tolerance classes containing $x \in O$ are subsets of the neighbourhood of x, $N(x)$.*

Proof. Given a tolerance space $\langle O, \cong_{\mathcal{B},\varepsilon} \rangle$ and tolerance class $A \in H_{\cong_{\mathcal{B},\varepsilon}}(O)$, then $(y, z) \in \cong_{\mathcal{B},\varepsilon}$ for every $y, z \in A$. Let $N_{\cong_{\mathcal{B},\varepsilon}}(x)$ be a neighbourhood of $x \in O$ and assume $x \in A$. For every $y \in A, (x, y) \in \cong_{\mathcal{B},\varepsilon}$. Hence, $A \subset N_{\cong_{\mathcal{B},\varepsilon}}(x)$. As a result, $N_{\cong_{\mathcal{B},\varepsilon}}(x)$ is superset of all tolerance classes containing x. □

Proposition 3. *Let $z_1, \ldots, z_n \in Z$ be a succession of objects, called query points, such that $z_n \in N(z_{n-1}) \backslash z_{n-1}$, $N(z_n) \subseteq N(z_{n-1}) \backslash z_{n-1} \subseteq \ldots \subseteq N(z_1) \backslash z_1$. In other words, the series of query points, $z_1, \ldots, z_n \in Z$, is selected such that each subsequent object z_n (where $z_n \neq z_{n-1}$) is obtained from the neighbourhood $N(z_{n-1})$, that is created only using objects from the previous neighbourhood. Then, under these conditions, the set $\{z_1, \ldots, z_n\}$ is a pre-class.*

Proof. For $n \geq 2$, let $S(n)$ be the statement that $\{z_1, \ldots, z_n\}$ is a pre-class given the conditions in Proposition 3.

BASE STEP ($n = 2$): Let $z_1 \in Z$ be the first query point, and let $N(z_1)$ be the first neighbourhood. Next, let z_2 represent the next query object. Since z_2 must come from $N(z_1)$, and all objects in $x \in N(z_1)$ satisfy the tolerance relation $z_1 \cong_{\mathcal{B},\epsilon} x$, $S(2)$ holds.

INDUCTIVE STEP: Fix some $k \geq 2$ and suppose that the inductive hypothesis holds, *i.e.*, $\{z_1, \ldots, z_k\}$ is a pre-class, and choose z_{k+1} from $N(z_k) \backslash z_k$. Since $N(z_k)$
$\subseteq N(z_{k-1}) \backslash z_{k-1} \subseteq \ldots \subseteq N(z_1) \backslash z_1$, z_{k+1} must satisfy the perceptual tolerance relation with all the objects in $\{z_1, \ldots, z_k\}$. Consequently, $\{z_1, \ldots, z_{k+1}\}$ is also a pre-class.

Therefore, by MI, $S(n)$ is true for all $n \geq 2$. □

Corollary 2. *Let $z_1, \ldots, z_n \in Z$ be a succession of objects, called query points, such that $z_n \in N(z_{n-1}) \backslash z_{n-1}$, $N(z_n) \subseteq N(z_{n-1}) \backslash z_{n-1} \subseteq \ldots \subseteq N(z_1) \backslash z_1$. In other words, the series of query points, $z_1, \ldots, z_n \in Z$, is selected such that each subsequent object z_n (where $z_n \neq z_{n-1}$) is obtained from the neighbourhood $N(z_{n-1})$, that is created only using objects from the previous neighbourhood. Then, under these conditions, the set $\{z_1, \ldots, z_n\}$ is a tolerance class if $|N(z_n)| = 1$.*

Proof. Since the cardinality of $N(z_1)$ is finite for any practical application, and the conditions given in Corollary 2 dictate that each successive neighbourhood will be smaller than the last, there is a n such that $|N(z_n)| = 1$. By Proposition 3 the series of query points $\{z_1, \ldots, z_n\}$ is a pre-class, and by Proposition 2 there are no other objects that can be added to the class $\{z_1, \ldots, z_n\}$. As a result, this pre-class is maximal with respect to inclusion, and by definition is called a tolerance class. □

The above observations are visualized in Fig. 14 using the example first introduced in Fig. 8, where the following conventions are used in the figures. In this case, only the first 21 objects of Z are shown, where each object is represented by a number from 1 to 21. A neighbourhood is represented by the colour grey. For example, the entire circle in Fig.14a is grey because it represents the neighbourhood of object 1, *i.e.* $N(1)$. Similarly, the grey portion of Fig. 14b represents the neighbourhood of object 20 only using the objects from $N(1)$ excluding object 1. In the above propositions and proofs this is labelled as $N(20) \subseteq N(1) \backslash 1$. Also, note these figures gives examples of the portion of the set of objects not contained in a neighbourhood. These areas are represented by the colours red, pink, green, and yellow; and an example can be seen in Fig. 14b where the area shaded red is the part of $N(1)$ that does not satisfy the tolerance relation with 20. Next, objects coloured black (as opposed to blue) are objects that have been added to the potential tolerance class (called a pre-class) and are not considered part of the neighbourhood. For instance, in Fig. 14b $\{1\}$ is black, in Fig. 14c $\{1, 20\}$ are coloured black, in Fig. 14d $\{1, 20, 10\}$ are coloured black, *etc.* Moreover, objects that are coloured blue satisfy the tolerance relation with all the black objects in the potential tolerance class (again, also known as a pre-class), but not

necessarily with all the other objects coloured blue. As an example, the objects coloured blue in Fig. 14c all satisfy the tolerance relation with $\{1, 20\}$, but they do not all satisfy the tolerance relation with each other, as is the case for objects 14 and 21.

Starting with the the proof of Proposition 3, a visual example of the base step in is given in Fig. 14a & 14b. Here, Fig. 14a contains $N(z_1)$, and according to Proposition 3, another query point $z_2 \in N(z_1)\backslash z_1$ is selected (*i.e.*, z_2 can be any object in $N(z_1)$ except z_1). Here, $z_2 = 20$ is selected because it is the next object closest to z_1. Since $z_1 \cong_{\mathcal{B},\epsilon} z_2$, the class $\{z_1, z_2\}$ is a pre-class. Continuing on, an example of the inductive step from the proof of Proposition 3 is given in Fig. 14e. In this case, there are $k = 5$ objects and $\{z_1, \ldots, z_5\} = \{1, 20, 10, 6, 15\}$. The area shaded grey represents $N(z_5)\backslash z_5 \subset, \ldots, \subset N(z_1)\backslash z_1$, along with the black coloured query points $\{z_1, \ldots, z_5\}$[6]. Note, while $z_5 = 15$ is not coloured black, it is considered a query point because it is the object used to create the current neighbourhood, and is added to the pre-class. It is coloured black in the subsequent figure. Notice that, as was mentioned above, all the blue objects in the grey area satisfy the tolerance with all the query points, but that the grey area does not represent a pre-class. Moreover, any new query point selected from $N(z_5)\backslash z_5 = \{16, 18, 3, 14, 11\}$ will also satisfy the tolerance relation with all the query points $\{z_1, \ldots, z_5\}$. Finally, Fig. 14f demonstrates the idea behind Corollary 2. In this figure, the area shaded grey represents the neighbourhood of $z_7 = 3$ along with all black query points. Observe that (besides the black query points) the shaded area only contains one object, namely z_7. Also, note that there are no more objects that will satisfy the tolerance relation with all the objects in the shaded area. As a result, the set $\{z_1, \ldots, z_7\}$ is a tolerance class.

Using Propositions 2 & 3 and Corollary 2, Algorithm 1 gives the pseudocode for an approach for finding all the tolerance classes on a set of objects Z. The general concept of the algorithm is, for a given object $z \in Z$, to recursively find all the tolerance classes containing z. The first step, based on Proposition 2, is to set z as a query point and to find the neighbourhood $N(z)$. Next, consider the nearest neighbour of z from the neighbour-

[6]According to the conditions given in Proposition 3 queries points are not included in subsequent neighbourhoods.

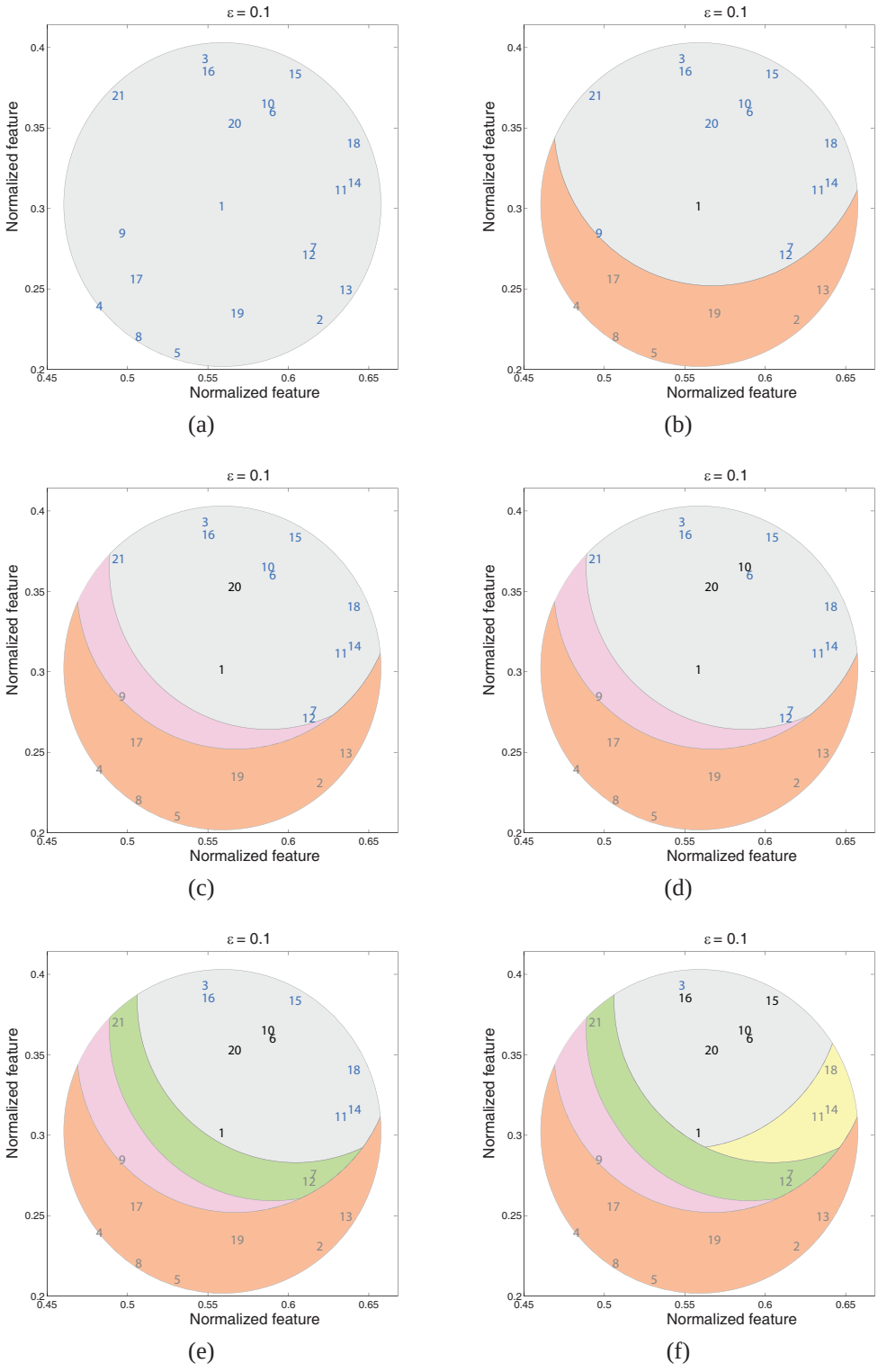


Figure 14: Visualization of Propositions 2 & 3 and Corollary 2. (a) $N(1)$, (b) $N(20) \subseteq N(1)\backslash 1$, (c) $N(10) \subseteq N(20)\backslash 20 \subseteq N(1)\backslash 1$, (d) $N(6) \subseteq N(10)\backslash 10 \subseteq N(20)\backslash 20 \subseteq N(1)\backslash 1$, (e) $N(15) \subseteq N(6)\backslash 6 \subseteq N(10)\backslash 10 \subseteq N(20)\backslash 20 \subseteq N(1)\backslash 1$, and (f) $N(3) \subseteq \cdots \subseteq N(1)\backslash 1$.

hood $N(z)$ as a query point and find its neighbourhood only considering objects in $N(z)$. Continue this process until the result of a query produces a neighbourhood with cardinality 1[7]. Then, the series of query points becomes the tolerance class. The tolerance class originally given in Fig. 8 was produced using this algorithm, and the intermediate steps of this algorithm are visualized in Fig. 14. As was mentioned, the tolerance class is then the series of query points given as

$$C = \{1, 20, 10, 6, 15, 16, 3\},$$

where the sequence of neighbourhoods is given in Table 3 (notice that results of the nearest neighbour search are obtained in order of increasing distance). Finally, it is important to note for this algorithm (and Algorithm 3) that the input is specified as $Z = X \cup Y$ since it is assumed here that the user wants to find tolerance classes to calculate the degree of nearness (*i.e.* tNM) between two sets X and Y. As was mentioned in Section 4.3, it is best to find the tolerance classes on the union of two sets X, Y, rather than find the tolerance classes on the individual sets when calculating tNM.

Algorithm 1: Algorithm for finding tolerance classes

Input : $Z = X \cup Y$
Output: $H_{\cong_{\mathcal{B},\varepsilon}}(Z)$

```
H_Z ← ∅;
for (i = 0; i < |Z|; i++) do
    C ← ∅;
    findNN(Z, C, i, H_{≅_{B,ε}}(Z));
end
```

While Algorithm 1 finds all the tolerance classes of objects from a set $Z = X \cup Y$, it is not very efficient. As a result, Algorithm 1 was modified, producing Algorithm 3. While the performance of this algorithm was much better than that of Algorithm 1, its runtime is still poor. For the worst case scenario, occurring when each pair of objects in Z satisfies the tolerance relation, the runtime is of order $O(|Z|^3 T)$, where T is the complexity of finding an object's neighbourhood from among the other $|Z| - 1$ objects. However, this runtime

[7]The result of a query will always be at least 1 since the tolerance relation is reflexive.

Procedure `findNN(`$N(z)$`)`, C, i, $H_{\cong_{\mathcal{B}}}(Z)$)

```
1  if (|N(z)| = 1) then
2      C ← C ∪ N(z);
3      H_{≅_B}(Z) ← H_{≅_B}(Z) ∪ C;
4  else
5      z' ← N^i(z);
6      N(z) ← N(z)\N^i(z);
7      C ← C ∪ z';
8      find N(z') only using the objects in N(z);
9      for (j = 0; j < |N(z')|; j++) do
10         findNN(N(z'), C, j, H_{≅_B}(Z));
```

Table 3: Intermediate steps of Algorithm 3 using data in Fig. 8

z'	$N(z')$
1	$\{1, 20, 12, 7, 9, 6, 19, 10, 17, 11, 14, 16, 18, 3, 13, 21, 15, 2, 5, 8, 4\}$
20	$\{20, 10, 6, 16, 3, 15, 21, 18, 11, 14, 7, 12, 9\}$
10	$\{10, 6, 15, 16, 3, 18, 11, 14, 21, 7, 12\}$
6	$\{6, 15, 16, 3, 18, 11, 14, 7, 12, 21\}$
15	$\{15, 16, 18, 3, 14, 11\}$
16	$\{16, 3\}$
3	$\{3\}$

can be significantly reduced by the addition of a simple heuristic. For example, step 3 can be changed so that an object from $N(z)$ can only be selected as z' in step 3 (*i.e.,* this rule is reset each time step 2 is visited), if it has not already been added to a tolerance class created from the original neighbourhood $N(z)$. As a result of this modification, the runtime in the worst case is now $O(|Z|^2T)$. Moreover, it should be noted that the algorithm is rarely run on worst case data. The worst case suggests that either the epsilon value is much too large, or that the data is so clustered that, from a perceptual point of view, every pair of objects in the set resembles each other. In either case, the data is not interesting from a nearness measure or image correspondence perspective. Lastly, the runtime on typical data is of order $O(|Z|cT)$, where $c \leq |Z|$ is a constant based on the object $z \in Z$ that has the largest neighbourhood.

Next, consider the affect of using Algorithm 3 (and the added heuristic) instead of using Algorithm 1. The main difference (besides improved runtime) is that the new algorithm

Algorithm 3: Algorithm for finding tolerance classes

Input : $Z = X \cup Y$
Output: $H_{\cong_{\mathcal{B}}}(Z)$

1 $H_{\cong_{\mathcal{B},\varepsilon}}(Z) \leftarrow \emptyset$;
2 Take an element $z \in Z$ and find $N(z)$;
3 Add z to a new tolerance class C. Select an object $z' \in N_{\cong_{\mathcal{B},\varepsilon}}(z)$;
4 Add z' to C. Find neighbourhood $N(z')$ using only objects from $N(z)$. Select a new object $z'' \in N(z')$. Re-label $z' \leftarrow z''$ and $N(z) \leftarrow N(z')$;
5 Repeat step 4 until a neighbourhood of only 1 element is produced. When this occurs, add the last element to C, and then add C to $H_{\cong_{\mathcal{B},\varepsilon}}(Z)$;
6 Perform step 3 (and subsequent steps) until each object in $N(z)$ has been selected at the level of step 3;
7 Perform step 2 (and subsequent steps) for each object in Z;
8 Delete any duplicate classes;

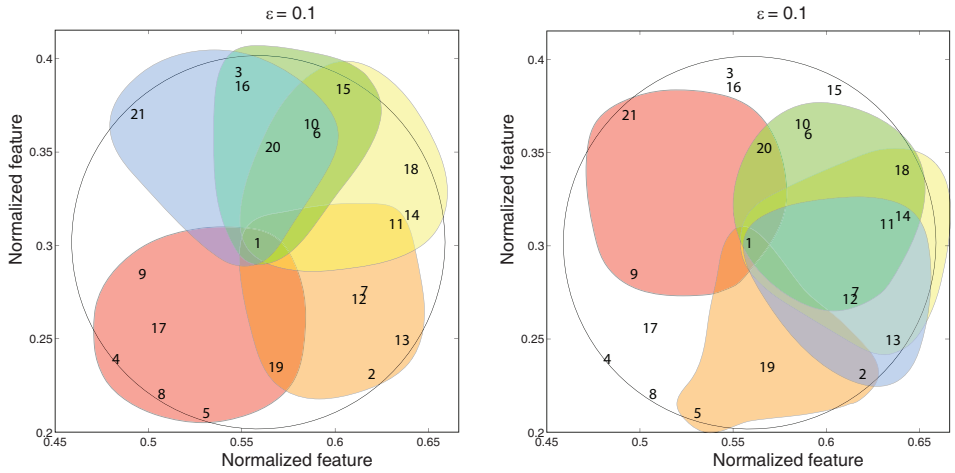
Table 4: Tables showing classes found and not found by Algorithm 3 (and the added heuristic).

Found Classes
$\{1, 2, 7, 11, 12, 13, 19\}$
$\{1, 3, 6, 10, 15, 16, 20\}$
$\{1, 3, 6, 10, 16, 20, 21\}$
$\{1, 4, 5, 8, 9, 17, 19\}$
$\{1, 6, 10, 11, 14, 15, 18, 20\}$

Missing Classes
$\{1, 2, 5, 19\}$
$\{1, 2, 7, 11, 12, 13, 14\}$
$\{1, 6, 7, 10, 11, 12, 14, 18, 20\}$
$\{1, 7, 11, 12, 13, 14, 18\}$
$\{1, 9, 20, 21\}$

does not find all the tolerance classes containing objects in Z; however the result is still a covering of Z. For example, Fig. 15a and Table 4 show the classes found and not found by Algorithm 3 (and the added hueristic). Another concern is whether the new algorithm will affect the outcome of the nearness measure. I conjecture that perceptually the difference does not matter because the result of the algorithm is still a covering of Z. Moreover, it is usually the case that an object belongs to more than one class, and that the extra classes would not significantly alter the perceptual information contained in the tolerances classes that are discovered. To test this conjecture, an experiment was conducted on the objects in Fig. 15 to provide an empirical argument indicating that using Algorithm 3 to find tolerance classes does not significantly affect the nearness measure between two sets of objects.

To perform this experiment, the 21 objects in Fig. 15 needed to be separated into two disjoint sets. Then, the nearness measure was calculated using only the tolerance classes



(a) (b)

Figure 15: Example of a tolerance classes not produced by Algorithm 3 (and the added heuristic). (a) Classes found by Algorithm 3 (and the added heuristic), and (b) classes not found by the algorithm.

Table 5: Error in nearness measure tNM using Algorithm 3 to find tolerance classes.

Error	% of Combinations Below Error
0.2	99.9%
0.15	98%
0.1	87%
0.05	55%

found by Algorithm 3 (and the added heuristic), as well as using all the tolerance classes. As one could guess, these objects can be separated into two disjoint sets in many different ways. The approach was to set the number of objects in each disjoint set, then calculate the nearness measures for all combinations of dividing these objects into two sets of specific cardinalities. This process was then repeated for each unique set of cardinalities between the two disjoint sets, *e.g.* two sets consisting of 1 and 20 elements, 2 and 19 elements, 3 and 18 elements, *etc.* The results of this comparison are given in Fig. 16, and Table 5. Notice that, for the objects in Fig. 15, the difference between calculating the nearness measure using all the classes and the classes found by Algorithm 3 is less than 0.2 for 99.9% of all combinations of objects divided into two dijoint sets. Thus, as a result of this experiment, Algoirthm 3 (and its reduced runtime) was used to generate the results presented in this thesis.

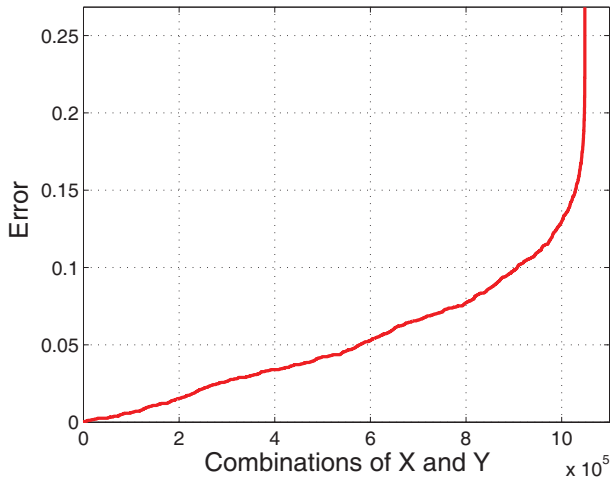


Figure 16: Plot showing error in nearness measure tNM using Algorithm 3 (and the added heuristic) to find tolerance classes.

Finally, this section is concluded by mentioning a few observations about the algorithms. First, both the original algorithm and the improved heuristic algorithm produce a set of classes that contain duplicates (more so in the case of the first algorithm). Consequently, it is necessary to remove duplicate classes before calculating the nearness measure. Also, the runtime of these algorithms can be significantly improved by approximate nearest neighbour searching, which is why the runtime was presented partially in terms of the complexity of finding neighbourhoods. The results presented in this thesis were obtained using the approximate nearest neighbour algorithm reported in [57], and discussed in Section 2.2. Lastly, these algorithms lend themselves to parallel processing techniques, and the results in this paper were also obtained using multi-threading on a quad core processor.

5 Application of Near Sets

As demonstrated in Section 3, tolerance near sets provide a systematic framework for measuring the similarity of objects and sets of objects, based on object descriptions, and in a manner similar to the way people perceive objects and the affinities between them. Thus, one might conjecture that near set theory is a natural choice for problems and research areas where the desired solution or result matches that of human perception. Indeed, the goal of this section is to demonstrate the validity of this conjecture, namely, that near set theory is well suited to solving problems in a manner similar to the problem-solving approach by humans in discerning similarities between objects.

The choice of an application for this demonstration is in the area of content-based image retrieval (see, *e.g.*, [60]), where the goal is to retrieve images from databases based on content of an image rather than on some semantic string or keywords associated with the image. The content of the image is determined by image processing functions that characterize qualities such as colour, texture, shape of objects in the images, and edges. Notice, that this approach is conducive to producing feature vectors, and as such, is an application in which near set theory can be easily applied. Furthermore, tolerance near sets together with the nearness measure on classes of objects derived from two perspective images provides a quantitative approach for accessing the similarity of images. Thus, the focus of this chapter is to demonstrate a practical application of near set theory in terms of content-based image retrieval, and to show that this framework does indeed produce results similar to those produced by human perception. The outline of the chapter is as follows: Section 5.1 demonstrates an approach to applying near set theory to images, Section 5.2 presents initial results toward applying near set theory to the problem of content-based image retrieval, Section 5.3 discusses the selection of ε, Section 5.3 presents other methods of measuring the nearness of two sets in a perceptual system, and Sections 5.5 & 5.6 present the results of performing content-based image retrieval on the SIMPLIcity image database using the near set approach.

5.1 Perceptual Image Analysis

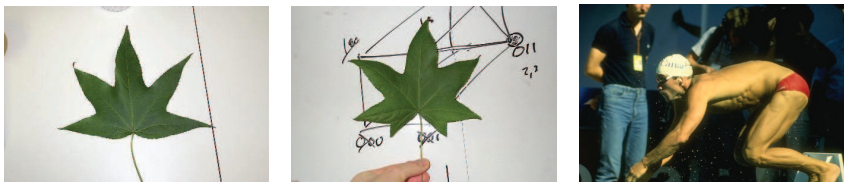
Near set theory can easily be applied to images by partitioning an image into subimages and considering each subimage as an object in the near set sense, *i.e.* each subimage is a perceptual object, and each object description consists of the values obtained from techniques of image processing on the subimage (see, *e.g.* Fig. 17). Moreover, this technique of partitioning an image, and assigning feature vectors to each subimage is an approach that has also been traditionally used in content-based image retrieval.

Formally, define a RGB image as $f = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$, where $\mathbf{p}_i = (c, r, R, G, B)^{\mathrm{T}}$, $c \in [1, M]$, $r \in [1, N]$, $R, G, B \in [0, 255]$, and M, N respectively denote the width and height of the image and $M \times N = T$. Further, define a square subimage as $f_i \subset f$ such that $f_1 \cap f_2 \ldots \cap f_s = \emptyset$, and $f_1 \cup f_2 \ldots \cup f_s = f$, where s is the number of subimages in f. Next, O can be defined as the set of all subimages, *i.e.*, $O = \{f_1, \ldots, f_s\}$, and $\mathbb{F}$ is a set of image processing descriptors or functions that operate on images. Then, the nearness of two images can be discovered by partitioning each of the images into subimages and letting these represent objects in a perceptual system, *i.e*, let the sets X and Y represent the two images to be compared where each set consists of the subimages obtained by partitioning the images. Then, the set of all objects in this perceptual system is given by $Z = X \cup Y$.

Figure 17: Example demonstrating the application of near set theory to images, namely the image is partitioned into subimages where each subimage is considered a perceptual object, and object descriptions are the results of image processing techniques on the subimage (Image used with permission [97]).

5.2 Initial Results

This section presents the initial results (published in [93]) demonstrating that near set theory can successfully be applied to the problem of content-based image retrieval. The adjective "initial" is stressed since these results are based on performing content-based image retrieval on a database containing only two categories that are quite different from each other, namely the database consists of images are from the Berkeley Segmentation Dataset [97] and the Leaves Dataset [98] (see, *e.g.*, Fig. 18). Nevertheless, these results represent an important first step to demonstrating the application of near set theory to the problem of content-based image retrieval.



(a) (b) (c)

Figure 18: Sample images. (a), (b) Leaves Dataset (Images used with permission [98]), and (c) Berkeley Segmentation Dataset (Image used with permission [97]).

To begin with, Fig. 19 is a plot of tNM values comparing the nearness of Fig.'s 18a & 18b and Fig.'s 18a & 18c using the normalized green value from the RGB colour model and Pal's entropy (see Sections 2.3.1 & 2.3.2). Furthermore, the results were obtained using $\varepsilon = 0, 0.01, 0.05, 0.1$ (note, the perceptual indiscernibility relation is used for $\varepsilon = 0$), and a subimage size of 10×10. Observe that the two leaf images produce higher nearness measure values than Fig. 18a and the Berkeley image because the leaf images produce objects that have more in common in terms of their descriptions (using the probe functions in $\mathcal{B}$). These results match our perception of the similarity between these three images. Also, note that the values using the perceptual indiscernibility relation are quite similar (near zero). In practise features values tend not to be exactly equal thus producing lower nearness measure values. As shown by the results, this problem can be overcome by using the perceptual tolerance relation.

The plot given in Fig. 19 suggests that the nearness measure would be useful in mea-

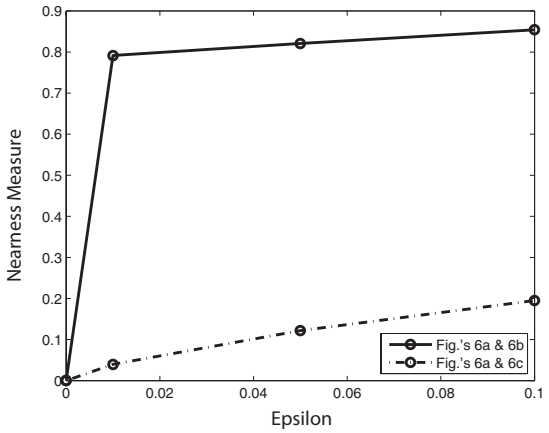


Figure 19: Plot showing NM values comparing Fig.'s 18a & 18b and Fig.'s 18a & 18c for $\varepsilon = 0, 0.01, 0.05, 0.1$.

suring the similarity of images. To investigate this property further, the Berkeley Segmentation Dataset and the Leaves Dataset were used to perform content-based image retrieval. Specifically, the image in Fig. 18a was selected as the query image and was compared to 200 images, 100 from both the leaves and Berkeley datasets, respectively[8]. The ideal result is for the highest 100 tNM values to be associated with the 100 leaf images. The results were generated using the same parameters as in Fig. 19, and they are compared using precision versus recall plots. Precision/recall plots are the common metric for evaluating content-based image retrieval systems where precision and recall are defined as

$$\text{precision} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{retrieved images}\}},$$

and

$$\text{recall} = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{relevant images}\}}.$$

In the idea case (described above), precision would be 100% until recall reached 100%, at which point precision would drop to # of images in query category / # of images in the database. In this case, the final value of precision will be 50% since there are two categories each containing 100 images.

[8]Note, the number of pixels in the leaf images were decimated by a factor of 4 to be closer in size to the Berkeley images, *i.e.*, their dimension was reduced from 896×592 to 448×296.

The results of these comparisons are given in Fig. 20. Notice, tNM produces a precision/recall plot with 73 images retrieved from the leaves dataset before a Berkeley image is selected. These results match intuition in that, at some level, our mind assesses similarity by comparing the descriptions of the objects we are considering, and that the comparison is not based on exact values (*i.e.*, the equivalence of features) but rather our mind easily allows some tolerance in making these comparisons.

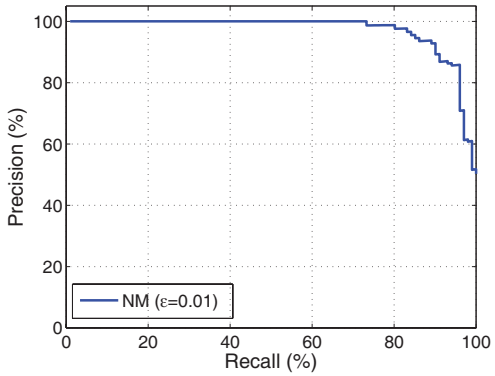


Figure 20: Precision versus recall plot obtained using Fig. 18a as a query image compared to 100 images from the Leaves Dataset [98] and Berkeley Segmentation Dataset [97].

5.3 Parameter Adjustment

While the results in the previous section are positive, some more work needs to be accomplished before applying near set theory to databases with multiple classes containing images with resemblance across categories. First, an investigation into selecting the value of ε needs to be performed. For normalized feature values, the largest distance between two objects occurs when one object has a feature vector (object description) of all zeros, and the other has a feature vector of all ones. As a result, ε is in the interval $[0, \sqrt{l}]$, where l is the length of the feature vectors. In any given application, there is always an optimal ε when performing experiments using the perceptual tolerance relation. For instance, a value of $\varepsilon = 0$ produces little or no pairs of objects that satisfy the perceptual tolerance relation, and a value of $\varepsilon = \sqrt{l}$, means that all pairs of objects satisfy the tolerance relation. Conse-

quently, ε should be selected such that the objects that are relatively[9] close in feature space satisfy the tolerance relation, and the rest of the pairs of objects do not. The selection of ε is straightforward when a metric is available for measuring the success of the experiment. In this instance, the value of ε is selected based on the best result of the evaluation metric. Fortunately, in this case, precision versus recall plots, defined in the context of image retrieval, can be used to evaluate the effectiveness of ε.

To demonstrate the selection of ε, a small database is used containing a collection of hand-finger movement images from three patients. One of the patients has rheumatoid arthritis, while the other two do not. Here, the goal is to perform content-based image retrieval and separate the images into three categories, one for each patient. The images were extracted from video sequences obtained from a telerehabilitation system that monitors patient hand-finger motion during rehabilitation exercises (see, *e.g.*, [99]). An example of the type of images obtained directly from the video is given in Fig. 21a. These images needed to be further processed to remove the common background (*e.g.* all the images contain the white desktop, the square blue sensor, *etc*) that would produce results indicating all the images were similar. Therefore, the mean shift segmentation algorithm (see Section 2.3.3) was used to create a segment containing only the hand in each image. The resultant segmented image is given in Fig. 21b where objects with similar colour are now grouped together into segments. The next step was to use the segment representing the hand as a mask to separate the hand from the original image (given in Fig. 21c). Next, notice the absence of the majority of the black background (representing the masked pixels in the original image) in Fig. 21d. Each image was cropped to an image containing only the hand because the output of probe functions on the black background would be the same for each image, and therefore, not contribute to differentiating the images.

Next, perceptual objects are created by partitioning the image into subimages, and, in this case, only one probe function was used, namely the average orientation of lines within a subimage. For example, the orientation can be determined (using the process given in

[9]Here, distance of "objects that are relatively close" will be determined by the application.

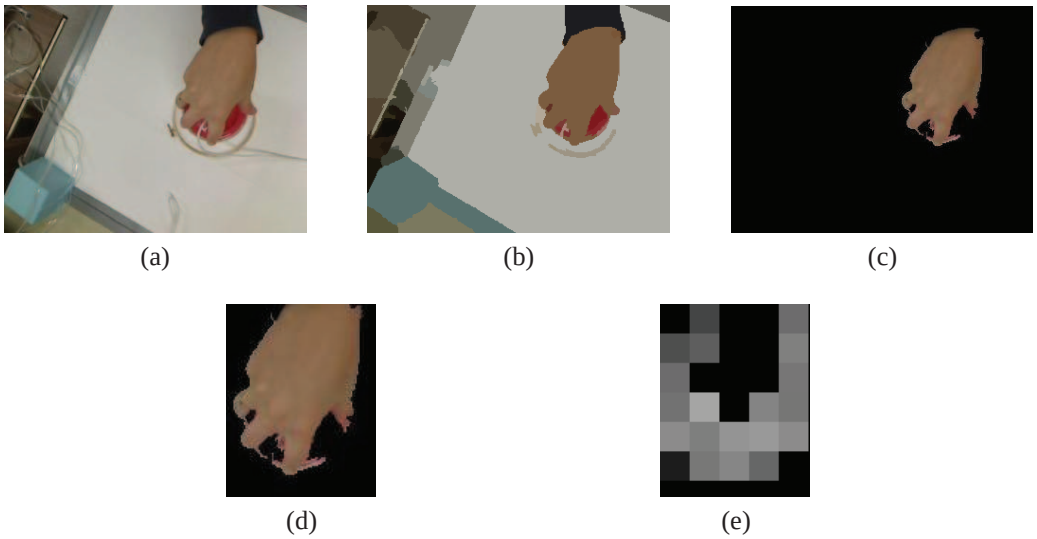


Figure 21: Figure showing preprocessing required to create tolerance classes and calculate nearness measure. (a) Original image, (b) segmented image, (c) hand segment only, (d) cropped image to eliminate useless background, and (e) final image used to obtain tolerance classes. Each square represents an object where the colour (except black) represents the average orientation of a line segment within that subimage.

Section 2.3.4) for each pixel considered part of a line detected in an image. Then, the probe function takes an average of all the orientations for pixels belonging to edges within a specific subimage. An example of the output of this probe function is given in Fig. 21e.

As was mentioned, these images will be used to demonstrate the selection of ε for a given application of tolerance near sets. Specifically, an image belonging to one of the three patients is used as a query image, and then the images are ranked in descending order based on their nearness measure with the query image. For example, the database of images used in this section contains 98 images, of which 30 are from the patient with arthritis, and respectively, 39 and 29 of them are form two patients without arthritis. Then, each image is in turn selected as the query image, and a nearness between the query image and every other image in the database is determined. Subsequently, a tolerance ε can be selected based on the number of images that are retrieved from the same category as the query image before a false negative occurs (*i.e.* before an image from a category other than the query image occurs).

Using this approach, Fig. 22 contains a plot showing the number of images retrieved before the precision dropped below 90% for a given value of ε. The image (out of all

possible 98 images) that produced the best query results is given in red, and the average is given in blue. Notice the best results in the average case occur with tolerance $\varepsilon = 0.05$, which is close to the $\varepsilon = 0.07$ in the best case. This plot suggests that retrieval of images in this database improves with a slight easing of the equivalence condition, but not much. Lastly, note that ε is also low due to the use of approximate nearest neighbour searching (see, *e.g.* Section 2.2) because the nearest neighbour of an object p can be p^* such that $dist(p,q) \leq (1+\alpha)dist(p^*,q)$, where q is the actual nearest neighbour. The effect is an increase in ε because it is now possible to include objects in the neighbourhood that are at a distance of $(1+\alpha)\varepsilon$.

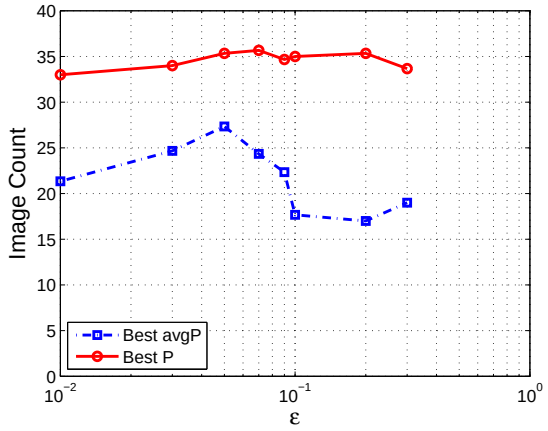


Figure 22: Plot giving the number of images retrieved before the precision falls below 90%.

Verifying the validity of selecting ε in this manner can be accomplished by both the visualization of the nearness measure for all pairs of images in the experiment, and by observing the precision recall plots directly. First, an image can be created where the height and width are equal to the number of images in the database, each pixel corresponds to the nearness measure from the comparison of two images, and the colours black and white correspond to a nearness measure of 0 and 1 respectively. For example, an image of size 98×98 can be created like the one in Fig. 23a where patient B is the one with arthritis, and each pixel corresponds to the nearness measure between two pairs of images in the database. Notice that a checkered pattern is formed with a white line down the diagonal. The white line corresponds to the comparison of an image with itself in the

database, naturally producing a nearness measure of 1. Moreover, the lightest squares in the image are formed from comparisons between images from the same patient, and that the darkest squares are formed from comparisons between the arthritis and healthy images. Also notice, that the boundaries in Fig. 23c & 23d are more distinct than for images created by other values of ε suggesting $\varepsilon = 0.05$, or $\varepsilon = 0.07$ is the right choice of ε. Similarly, square corresponding to patient C has crisp boundaries in Fig. 23a & 23h, and is also the brightest area of the figure, suggesting that a value of $\varepsilon = 0.3$ would also be a good choice for images belonging to patient C.

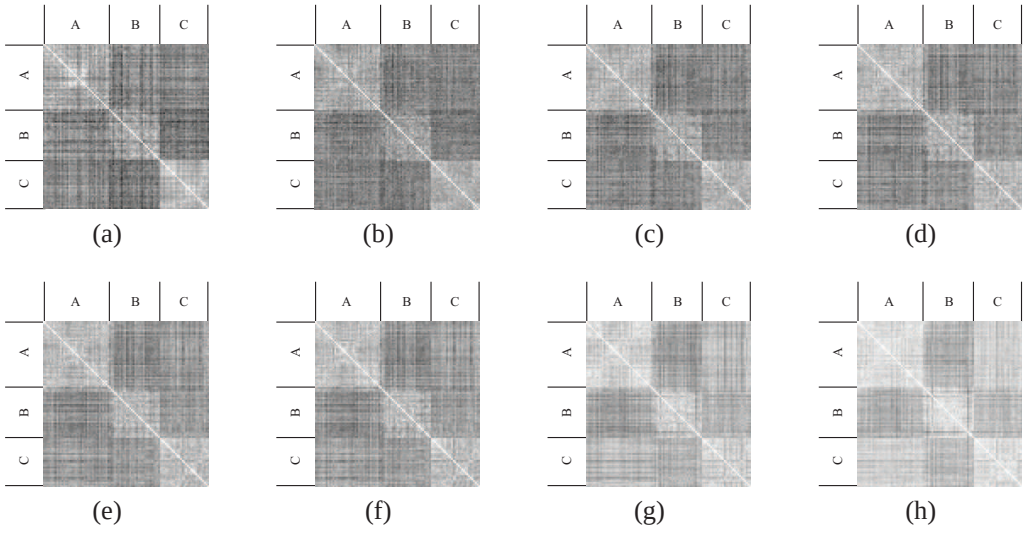
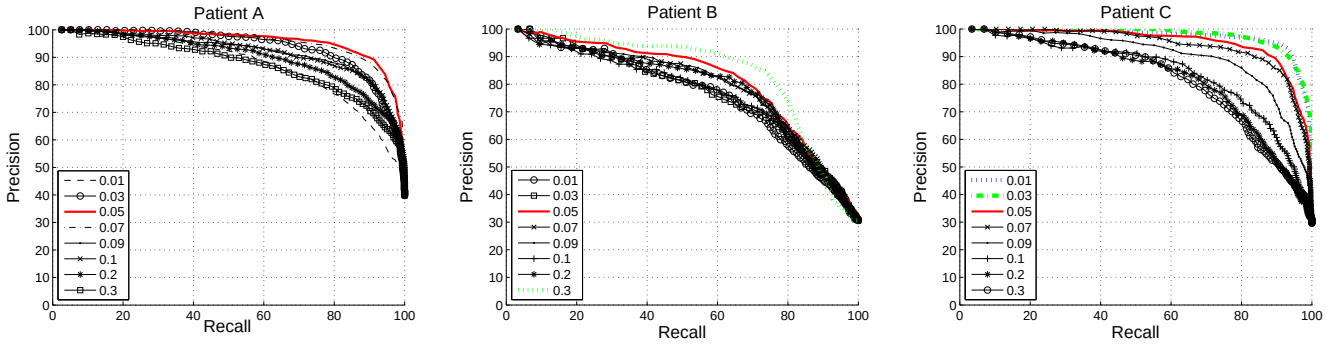


Figure 23: Images of nearness measure obtained from comparing the 98 images from three subjects to each other. (a) - (h) Visualization of nearness measure using $\varepsilon \in \{0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3\}$. Patients B has arthritis, while A and C do not.

Next, Fig. 24 gives plots of the average precision versus recall for each patient. These plots were created by fixing a value of ε, and calculating precision versus recall for each image belonging to a patient. Then, the average of all the precision/recall values for a specific value of ε are added to the plot for each patient. The results for selecting $\varepsilon = 0.05$ are given in red, and in the case of patients B and C, the choice of ε that produce a better result than $\varepsilon = 0.05$ are also highlighted.



(a) (b) (c)

Figure 24: Plots showing the average precision recall plots for patients A-C.

5.4 Other measures

This section introduces two additional measures for determining the degree that near sets resemble each other. These measures were created out of a need for making comparisons of the results generated by the nearness measure. Here, one of two approaches could have been investigated. Namely, the nearness measure could be compared with a content-based image retrieval system or measure that is currently regarded as the best approach for a database with given characteristics. Or, the nearness measure could be compared with measures that determine nearness in a manner comparable to tNM. Since the focus of this thesis is to demonstrate that the application of near sets to the problem of content-based image retrieval is possible, where the results match that of human perception, the latter approach was taken. As a result, approaches were created, based on existing theories, that measure the distance between sets.

5.4.1 Hausdorff Distance

The Hausdorff distance is used to measure the distance between sets in a metric space [100–102], and is defined as

$$d_H(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\},$$

where $\sup$ and $\inf$ refer to the supremum and infimum, and $d(x, y)$ is the distance metric (in this case it is the L^2 norm). The distance is calculated by considering the distance from a single element in a set X to every element of set Y, and the shortest distance is selected as the infimum (see, *e.g.*, Fig. 25). This process is repeated for every $x \in X$ and the largest distance (supremum) is selected as the Hausdorff distance of the set X to the set Y. This process is then repeated for the set Y because the two distances will not necessarily be the same. Keeping this in mind, the measure tHD [93] is defined as

$$tHD_{\cong_{\mathcal{B},\varepsilon}}(X,Y) = \left(\sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C| \right)^{-1} \cdot \sum_{C \in H_{\cong_{\mathcal{B},\varepsilon}}(Z)} |C|(\sqrt{l} - d_H(C \cap X, C \cap Y)).$$

Observe, that low values of the Hausdorff distance correspond to a higher degree of resemblance than larger distances. Consequently, the distance is subtracted from the largest distance $\sqrt{l}$. Also, notice that the performance of the Hausdorff distance is poor for low values of ε, since, as tolerance classes start to become equivalence classes (*i.e.* as $\varepsilon \to 0$), the Hausdorff distance approaches 0 as well. Thus, if each tolerance class is close to an equivalence class, the resulting distance will be zero, and consequently the measure will produce a value near to 1, even if the images are not alike. In contrast, as ε increases, the members of classes tend to become separated in feature space, and, as a result, only classes with objects that have objects in X that are close to objects in Y will produce a distance close to zero. What does this imply? If for a larger value of ε, relatively speaking, the set of objects $Z = X \cup Y$ still produces tolerance classes with objects that are tightly clustered, then this measure will produce a high measure value. Notice, that this distinction is only made possible if ε is relaxed. Otherwise, all tolerance classes will be tightly clustered.

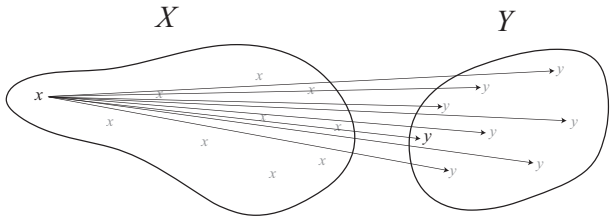


Figure 25: Example demonstrating a single step in determining the Hausdorff distance between two sets.

The Hausdorff distance is a natural choice for comparison with the tNM nearness measure because it measures the distance between sets in a metric space. Recall, that tolerance classes are sets of objects with descriptions in l-dimensional feature space. The nearness measure evaluates the split of a tolerance class between sets X and Y, where the idea is that a tolerance class should be evenly divided between X and Y, if the two sets are similar (or the same). In contrast, the Hausdorff distance measures the distance between two sets. Here the distance being measured is between the portions of a tolerance class in sets X and Y. Thus, two different measures can be used on the same data, namely the tolerance classes obtained from the union of X and Y.

5.4.2 Hamming Measure

The Hamming measure introduced in this section was inspired by the Hamming measure in [103], and since the Hamming measure is not defined in terms of sets, it was modified to give the following

$$tHM_{\cong_{\mathcal{B}}}(X,Y) = \frac{1}{|H_{\cong_{\mathcal{B}}}(Z)|} \cdot \sum_{C \in H_{\cong_{\mathcal{B}}}(Z)} \mathbf{1}(\parallel \text{avgn}(C \cap X) - \text{avgn}(C \cap Y) \parallel_2 \leq th),$$

where $\mathbf{1}(\cdot)$ is the indicator function and avgn$(C \cap X)$ is the average feature vector used to describe objects in $C \cap X$. For example, the average feature vector can be calculated by adding all the values for a specific feature in the feature vector in $C \cap X$, and then dividing by the number of objects. The idea behind this measure is that, for similar sets, the average feature vector of the portion of a tolerance class (obtained from $Z = X \cup Y$) that lies in X should have values similar to the average feature vector of the portion of the tolerance class that lies in Y. Observe, that if $th = \varepsilon$, this function will simply count the number of classes that are not singletons, *i.e.* classes that contain more than one element, since all objects have descriptions whose distances are less than ε. If $th = \varepsilon$, than this measure will perform best for low levels of ε, since only sets that resemble each other will contain classes with cardinality greater than one. Otherwise, this measure will perform in a similar manner to

tHD, namely, this measure will produce high values for classes which have objects in X that are close to objects in Y with respect to th.

5.4.3 Application

The method of applying these measures to the image correspondence problem is the same as that described in Section 5.1. To reiterate, consider Fig. 26 where each rectangle represents a set of subimages (obtained by partitioning the original images X and Y) and the coloured areas represent some of the obtained tolerance classes[10]. Recall, as mentioned in Section 4.3, the tolerance classes are created based on the feature values of the subimages, and consequently, do not need to be situated geographically near each other (as shown in Fig. 26). In the case of the nearness measure, the idea is that similar images should produce tolerance classes with similar cardinalities. Consequently, the cardinality of the portion of a tolerance class belonging to set X is being compared with the cardinality of the portion of the tolerance class belonging to set Y (represented in Fig. 26 as sets with the same colour). The Hausdorff and Hamming measures take the same approach, but rather consider the Hausdorff distance or the average vector distance between the portions of the classes.

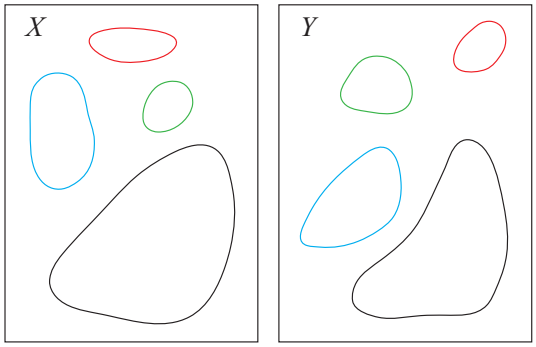


Figure 26: Graphical representation of the application of the nearness measures to a pair of images.

[10]The tolerance relation covers both images, but not all the classes are shown in the interest of clarity

5.5 SIMPLIcity Image Database

This section presents the application of near set theory to the problem of content-based image retrieval on the SIMPLIcity image database [104,105], a database of images containing 10 categories with 100 images in each category. The categories are varied with different objects and scenes, and images in different categories can also resemble each other (see, *e.g.* Fig. 27). While the goal is to retrieve images in a manner similar to that of human perception, one must recall from Section 3, perceptual information is always presented with respect to the probe functions contained in $\mathcal{B}$ just as our senses define our perception of the world. For example our ability to view light in the visible spectrum rather than infra red or microwaves spectra defines our perception of the world just as the selection of probe functions constrains the amount of perceptual information available for extraction from a set of objects. Thus, the ability of a near set-based system to assess similarity of images in a manner that mimics human perception of nearness, is completely dependent on the features selected to describe the objects (subimages). This dependence is not a new concept, and is present in any research area that is dependent on feature extraction and feature value vectors (see, *e.g.*, [53]). Moreover, the precision versus recall plots measure the ability of a system to return images from the correct category. However, since images across categories can also be similar (depending on the features used to describe the images), the results will be presented using both precision versus recall plots, as well as showing the top 10 images that match a given query image.

Before performing retrieval, both ε and the probe functions need to be selected. A common approach to performing content-based image retrieval is to use features based on colour, and texture [60, 80, 106, 107]. Also, another type of feature that is commonly used in pattern recognition, image analysis, ophthalmology, medical imaging, optical engineering, and watermarking are Zernike moments, since they provide region-based descriptors of an image that are invariant with respect to rotation and reflections [56]. Accordingly, the results of image retrieval in this section were obtained using 18 features, namely 4 texture features obtained from the grey-level co-occurrence matrix of a subimage (see, *e.g.* Sec-

Figure 27: Examples of each category of images. (a) - (d) Categories 0 - 3, and (e) - (i) categories 5 - 9 (Images used with permission [104, 105]).

tion 2.3.6), the first and second moments of u and v in the CIELUV colour space (see, *e.g.*, Section 2.3.8), an edge based feature (see, *e.g.* Section 2.3.4), and the Zernike moments of order 4, excluding $\widetilde{A}_{00}$ (see, *e.g.*, Section 2.3.7).

While the selection of ε should have followed the approach outlined in Section 5.3, this was not the case due the runtime of the algorithm used to find the tolerance classes and the size of the database. The approach was to perform a comprehensive comparison of each images in the database with each other image. Since the result of comparing two images with the nearness measure is symmetric, *i.e.* $tNM(X,Y) = tNM(Y,X)$, comparing each image to each other image involves $(1000)(1001)/2$ comparisons. Some of average runtimes for finding tolerance classes on objects obtained from a pair of images in the SIMPLIcity database (using a subimage of 20×20) is given in Fig. 28, where the runtime ranges from 0.3 sec. for $\varepsilon = 0.2$ to 606 sec. for $\varepsilon = 1.5$. Since this section is devoted to demonstrating a practical application of near set theory, the value of ε was selected based on performing

the experiment in a reasonable amount of time, rather than on the value of ε that preformed best on a small subset of the database, with the goal of showing that even for small values of ε this approach is quite powerful. The runtime of the experiment was further reduced by removing the category of images shown in Fig. 28. They were removed because correctly retrieving images from this category was trivial, as demonstrated by the precision versus recall plot of a random image from this category in Fig. 28. A more realistic experiment would be to identify the different type of dinosaurs and perform retrieval among this single category. In any event, this category was removed to reduce the total time to perform the experiment.

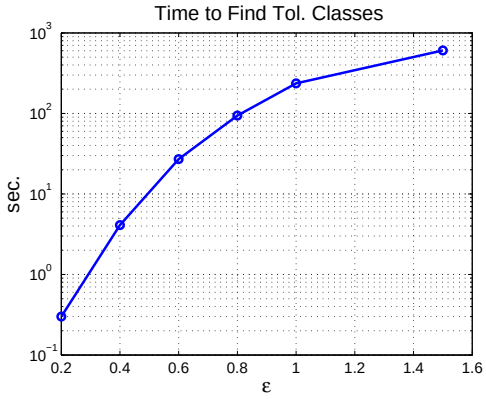


Figure 28: Plot of ε versus time to find all the tolerance classes on a pair of images in the SIMPLIcity image database, using the algorithm given in Section 4.4 on a quad-core 2.4 GHz machine, running a multi-threaded application.

(a) (b) (c)

Figure 29: Examples of images from category 4 (Images used with permission [104, 105]).

5.6 Discussion

The results of applying near set theory to the problem of content-based image retrieval on the SIMPLIcity image database are presented in this section. Fig. 31 contains three plots,

Figure 30: Precision versus recall plot for image 412.jpg.

one for each measure, giving the precision versus recall data for each category, and Fig. 32 gives the precision versus recall plots for all three measures grouped by category. These plots were generated by comparing each image in the SIMPLIcity database to each other image. Moreover, the precision versus recall plots for a specific category were generated by taking the average of all the precision and recall values for each image in that category. Next, Fig. 33 - 41 give the ten highest ranked images, using the tNM nearness measure, for each category based on the query image that produced the best results, determined by the largest area under the precision recall curve. Also given in these figures is the largest tolerance class obtained between the query image and the second highest ranked image (the first ranked image being the query image). These results were generated with $\varepsilon = 0.2$ (giving a total runtime of approximately $(900)(901)(0.3)/2 = 1.4$ days), a subimage size of 20×20 and the 18 probe functions identified in the previous section.

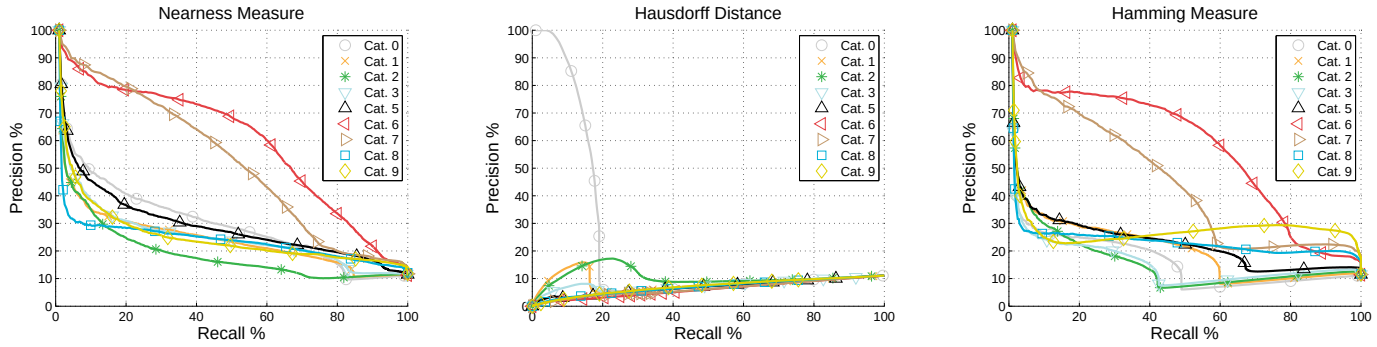


Figure 31: Precision versus recall plots for all three measures on the simplicity database. (a) Nearness measure, (b) Hausdorff distance, and (c) Hamming measure.

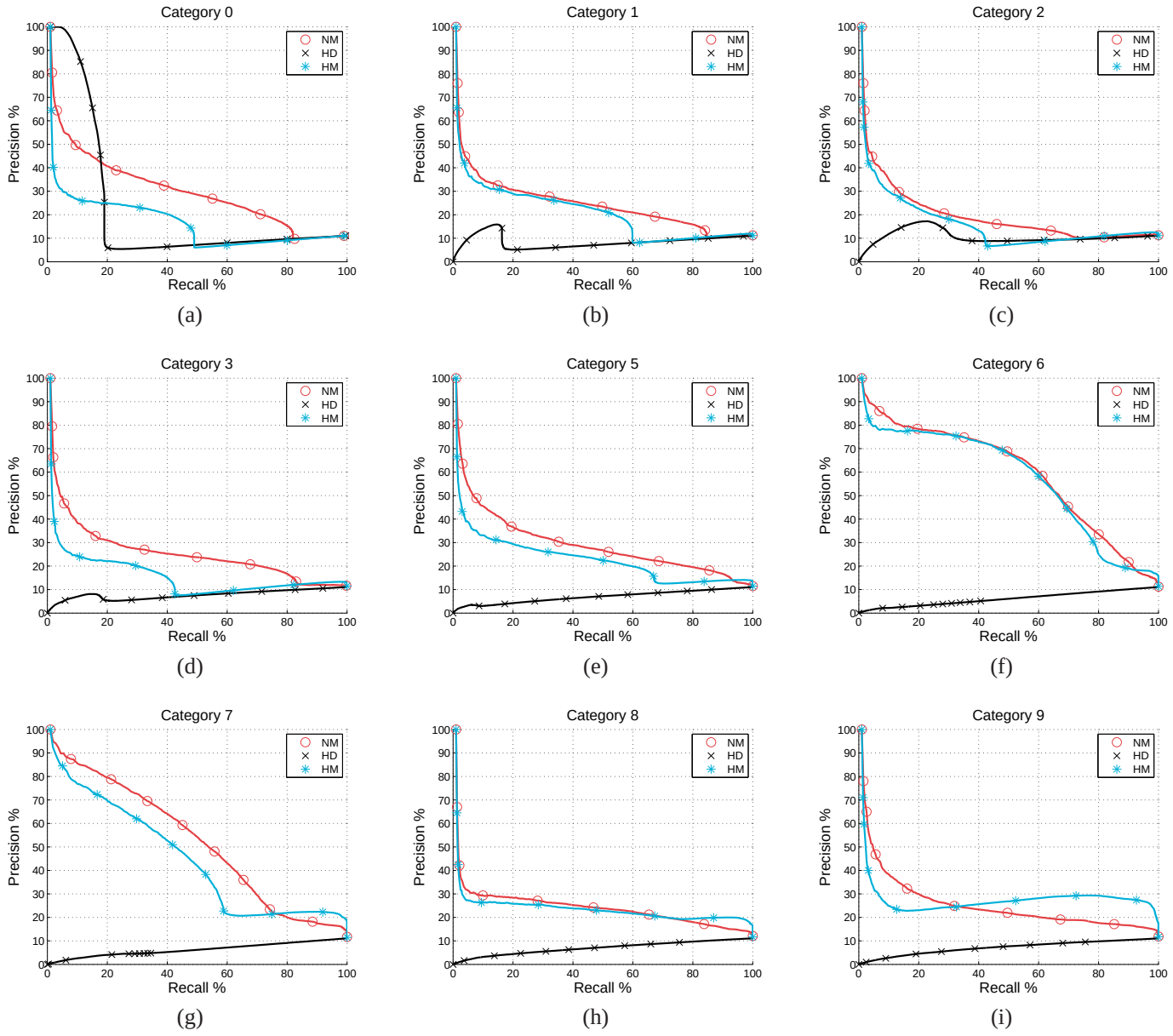


Figure 32: Precision versus recall plots for all three measures grouped by category. (a) - (i) Categories 0 - 9 (excluding category 4).

Figure 33: Results of best query from Category 0 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

Figure 34: Results of best query from Category 1 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

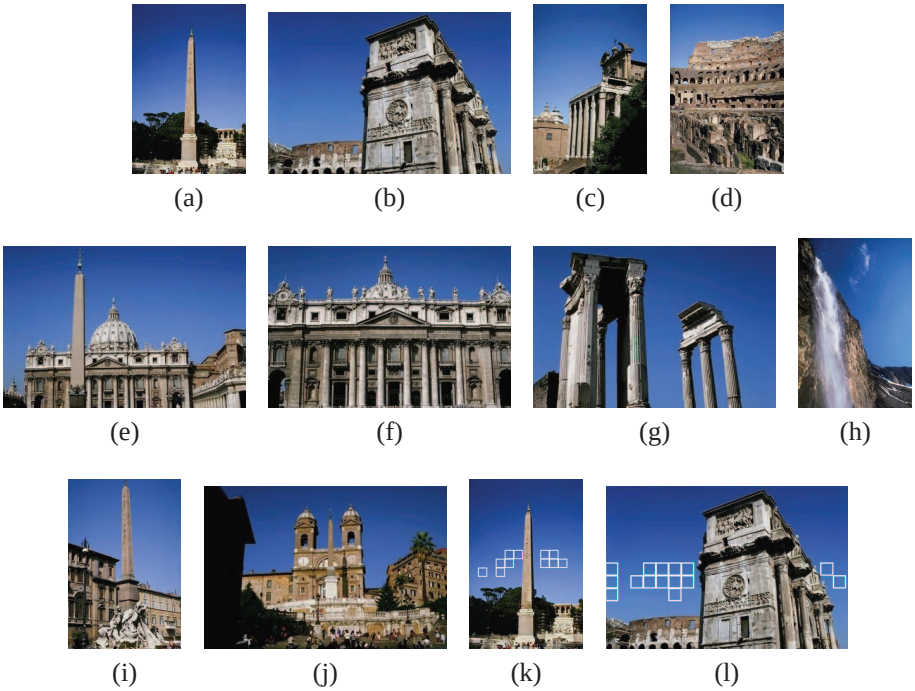


Figure 35: Results of best query from Category 2 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

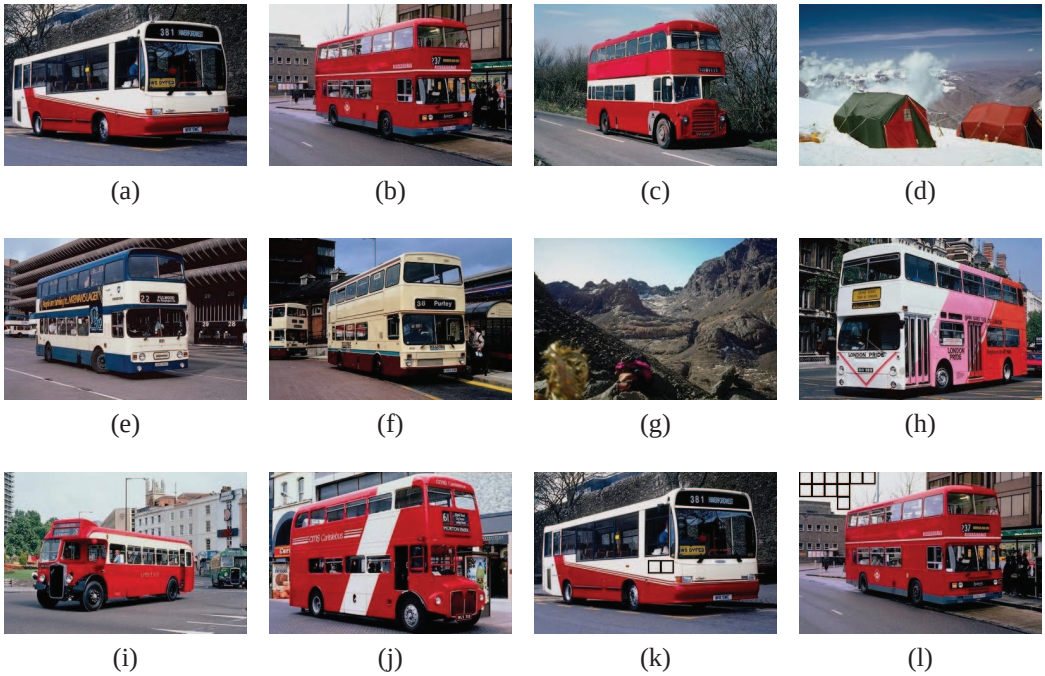


Figure 36: Results of best query from Category 3 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by black boxes) between images (a) & (b) (Images used with permission [104, 105]).

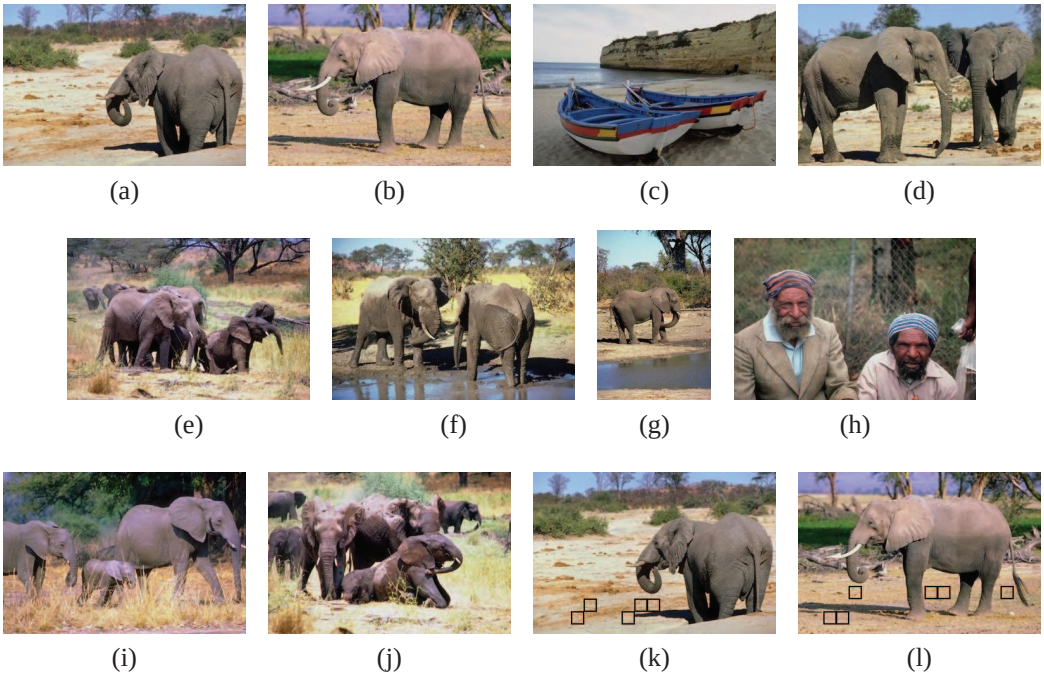


Figure 37: Results of best query from Category 5 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by black boxes) between images (a) & (b) (Images used with permission [104, 105]).

Figure 38: Results of best query from Category 6 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

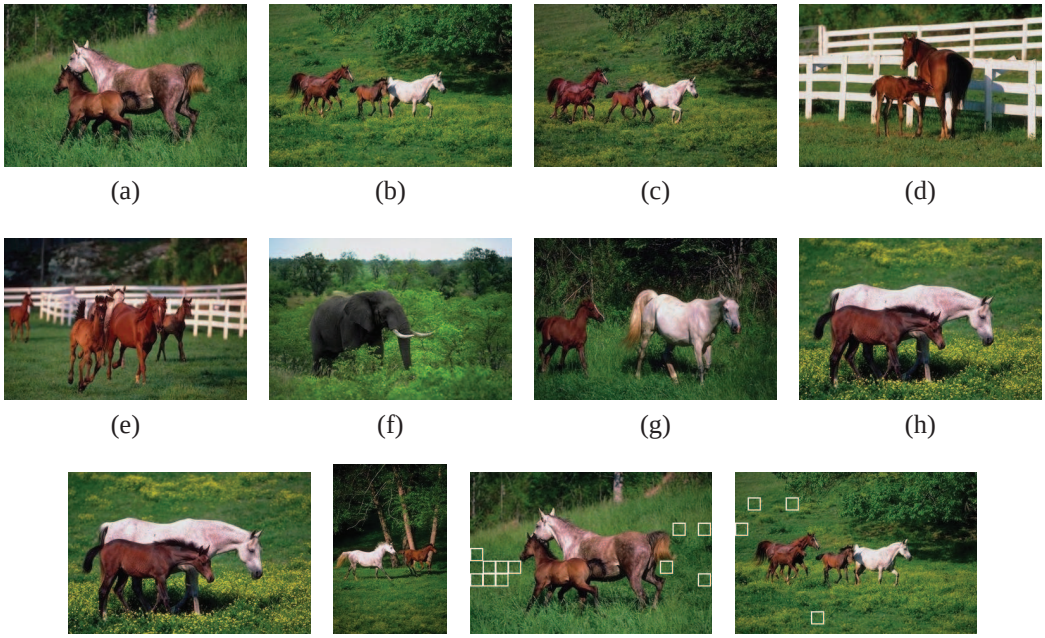


Figure 39: Results of best query from Category 7 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

Figure 40: Results of best query from Category 8 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by black boxes) between images (a) & (b) (Images used with permission [104, 105]).

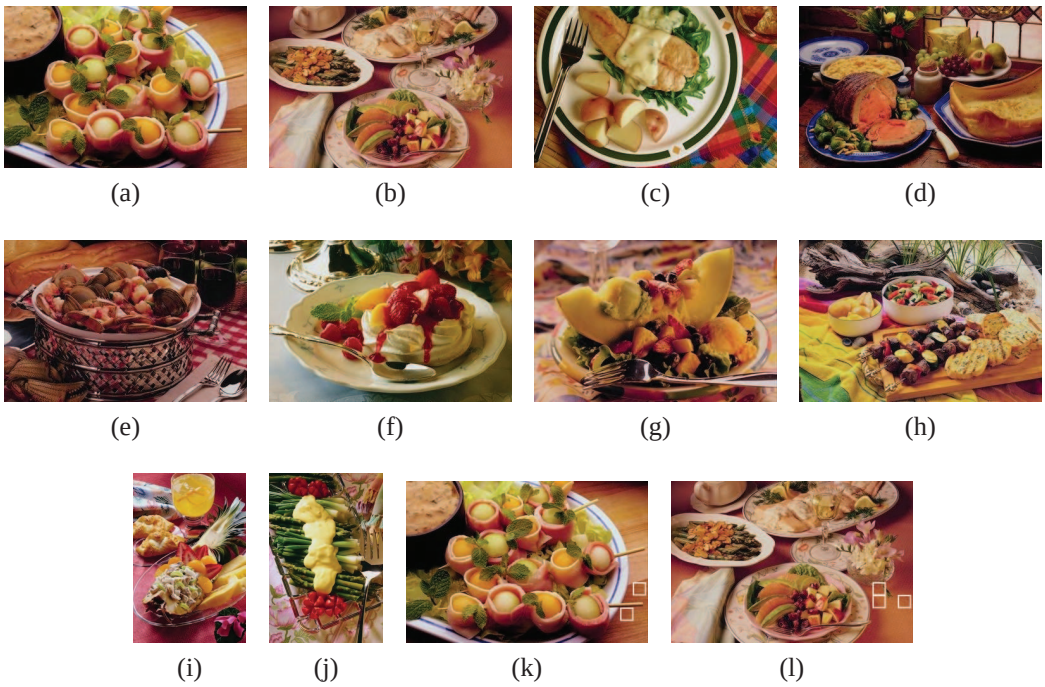


Figure 41: Results of best query from Category 9 using tNM. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

Starting with Fig. 31, the following observations can be made. First, the measure based on the Hausdorff Distance performs quite poorly; *i.e.* it does so poorly that the plots for nearly every category start with a precision of 0, meaning that the query image was not the highest ranked image. This is due to the observation made in Section 5.4.1, that the Hausdorff distance does not perform well for low values of ε. An improvement of tHD can be seen in Fig. 42 for a specific query image[11]. Notice, that as ε increases from 0.2 to 0.7, the results do improve, *i.e.*, the query image is now given the highest ranking by the measure, and the area under the curve has increased, implying that a greater number of images from the correct category are initially retrieved for $\varepsilon = 0.7$ than for $\varepsilon = 0.2$. For example, if the images are ranked according to the tHD measures, there are 80 images in the first half that are retrieved from the correct category using $\varepsilon = 0.7$, compared with 6 with $\varepsilon = 0.2$. Furthermore, as suggested by the results published in [93], these results would probably continue to improve with a larger value of ε (if the testing time was not prohibitive), before they begin to decline.

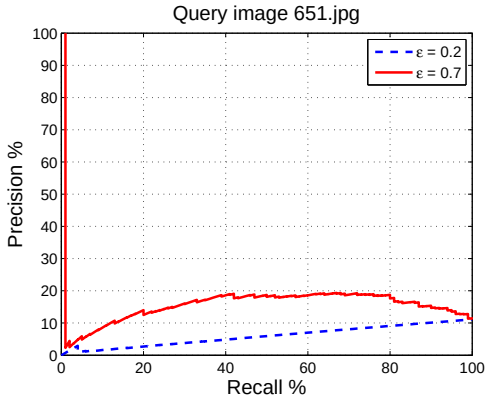


Figure 42: Plot demonstrating improvement of tHD as ε increases.

The other observations that can be made about Fig. 31 is that tNM and tHM produce similar results, and that categories 6 and 7 seem to produce the best results. This can easily be explained by observing that category 6 is quite different from all the other categories since these images are close-up shots of flowers, and both these categories are the only

[11]This image was selected since it was the query image that produced the best results in Category 6 using tNM.

ones where the background in all 100 images have similar colour and texture. Next, the explanation of why the results of the other categories are so poor is as follows. As can be seen in Fig. 33 - 41, especially Fig. 37 & 40, there are images in different categories that are similar to each other, especially describing the images using the 18 probe functions mentioned in the previous section. Thus, queries will produce results that are similar to the query image in terms of the selected probe functions, but the images may be from different categories, which drives down the precision of the query. However, as can be seen in the results most of these images are perceptually similar, both in terms of our subjective observations, and, obviously, by the probe functions selected to perform this experiment. Moreover, these results can be improved by increasing ε. Notice, that in terms of the maximum distance between objects, $\sqrt{18}$, the choice of ε is quite small. Thus, the query images in Fig. 37 & 40 that produced the poorest results were retested with $\varepsilon = 0.7$. The results of these queries are given in Fig. 43 & 44. Notice the the improvement. Another approach to improving the precision versus recall plots would be to change the selection of probe functions, although, since the database is so varied, there is bound to be images that are retrieved from other categories.

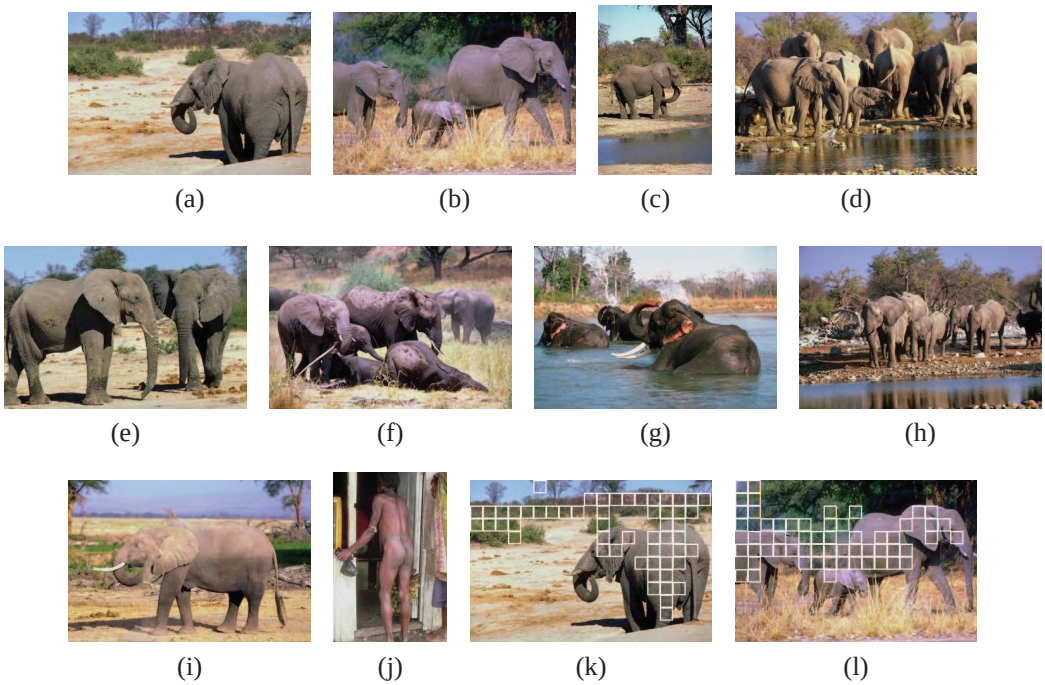


Figure 43: Results of best query from Category 5 using tNM with $\varepsilon = 0.7$. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

Figure 44: Results of best query from Category 8 using tNM with $\varepsilon = 0.7$. (a) Query image, (b) - (j) closest images using tNM, and (k) & (l) largest tolerance class (shown by white boxes) between images (a) & (b) (Images used with permission [104, 105]).

As was mentioned, Fig. 32 gives the results of each measure grouped by category. Furthermore, since the selection of ε was so low, the value of th was set to ε. Thus, tHD in this case is giving a measure of the number of tolerance classes that were not singletons. Observe that the tNM nearness measure performs slightly better than tHD for all categories, with the possible exception of category 9. For this reason, Fig. 33 - 41 are the best query images from each category using tNM.

Lastly, a question one might ask is "Why do all the subimages belonging to all the largest tolerance classes in Fig. 33 - 41 tend to be of uniform colour, especially since the selected probe functions take into account colour, texture, and local geometry?" The answer is due to the selection of $\varepsilon = 0.2$, a choice that was based on time constraints rather than on optimal performance. Selecting $\varepsilon = 0.2$ means that most of the tolerance classes are quite close to equivalence classes, *i.e.* the objects of a single tolerance class are quite close in feature space. Consequently, ε is not large enough to account for the differences in the descriptions of subimages containing texture in the image that is perceptually similar. Note, there are classes that have texture, they are just not the largest, which is depicted in Fig. 33 - 41. However, by increasing ε the subimages that contain texture start to form larger classes, as evidence by Fig. 43k & 43l and Fig. 44k & 44l.

5.6.1 Future Work

The goal of this thesis was to contribute to near set theory, and to provide practical applications demonstrating near set theory is well suited to problems where the desired solution is based on human perception of objects. While this goal was achieved, there are still some open problems to consider for future work.

Improve Algorithm Runtime While the results in this thesis are promising, there is one significant drawback to the tolerance near set approach, namely, the time required to determined tolerance classes. First, it should be noted that generally one does not compare every image in a database to every other image. This approach was taken to perform a comprehensive test using the nearness measures. That being said, there are two solutions

that may, either independently or together, significantly reduce the computation time, and consequently, warrant investigation. The first approach involves increasing the amount of processors available for performing the calculations. For instance, most modern computers have Graphics Processing Units (GPUs) with many cores (processors). For example, the machine used to produce the results presented in this thesis has a GPU with 128 cores operating at 600 MHz. Even if it is possible to reduce the execution time by a factor of 10, then the time required to perform the test that generated the results from SIMPLIcity database goes from approximately 1.4 days to 3 hours.

The other approach is based on preprocessing the images in the database prior to performing queries[12]. For instance, an approach that should be investigated is to find the tolerance classes for each image independently, prior to any queries being performed, and store only the average feature vector values created from all the objects in each tolerance class. Then, during an actual query, instead of finding tolerance classes on the objects $Z = X \cup Y$, where X is the query image, and Y is an image belonging to the database, it may be a good idea to create neighbourhoods using the stored average feature vectors as neighbourhood centres, and the objects from the query image to populate the neighbourhoods. This would significantly reduce the computation time in two ways. First, tolerance classes only need to calculated once for each image in the database, rather than for each image in the database during each new query. Second, during a query, only n comparisons need to be made, where n is the number of objects generated from the query image. This method would definitely be faster, and is likely to produce comparable results since it is still based on tolerance classes. It may even produce better results in scenarios where larger values ε can be used without penalty of extremely large computation times.

Other Content-Based Image Retrieval Methods As was mentioned above, the focus of the thesis was to advance near set theory, and to provide an application of near set theory which demonstrates both its utility, and that solutions based on the near set approach pro-

[12] A simple solution (which was used to generate the results in this thesis) would be to compute and store the feature values for each sub image ahead of time so that they do not need to be computed anew during each query. However, this was not the major bottleneck in generating these results.

duce results matching human perception. That being said, future work definitely warrants contrasting the application of the near set approach to the problem of content-based image retrieval should with other methods to identify its strengths and weakness. Furthermore, preliminary work toward this aim is presented here. To reiterate, the strength of the proposed approach is its apparent ability to measure the similarity of objects, and, in this case, images, in a manner similar to human perception. Moreover, initial testing demonstrates that the near set approach is competitive with other existing content-based image retrieval systems as demonstrated below in Fig. 45 & 46. Note, Fig. 45 represents the query image from the category that produced the worst results using $\epsilon = 0.2$ and the 18 probe functions used to generate the above results, and Fig. 46 represents the query image from the category that produced the best results using the same parameters. Finally, this work is considered preliminary since this comparison was not exhaustive in that there are many other methods not considered, and the selected methods used default settings.

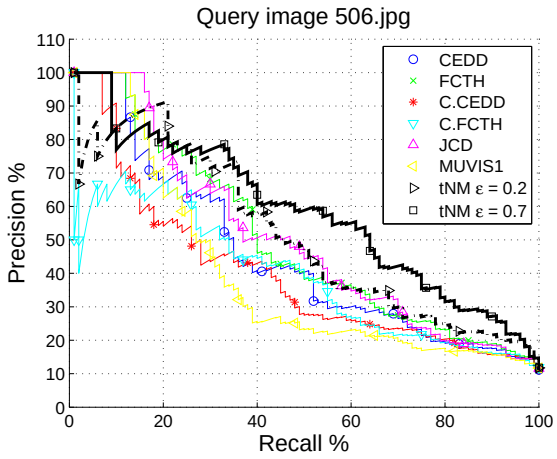


Figure 45: img(Anaktisi) system [108](CEDD [109] FCTH [110] JCD [111] C.CEDD & C.FCTH [108]) MUVIS system [112–114]

Other Distance Measures

While the L^2 norm was the distance used to define the tolerance relation in the first papers published on tolerance near sets (see, *e.g.* [33, 36]), an interesting area for future work is an investigation into using different distance measures for defining other tolerance relations. To that end, the plot in Fig. 47 demonstrates a few tests using Minkowski dis-

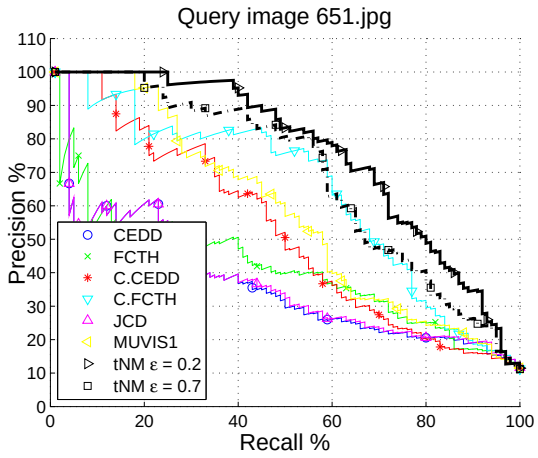


Figure 46: img(Anaktisi) system [108](CEDD [109] FCTH [110] JCD [111] C.CEDD & C.FCTH [108]) MUVIS system [112–114]

tances of orders other than 2 (which is the L^2 norm). This work is considered preliminary since there are many other distance measures that could be considered.

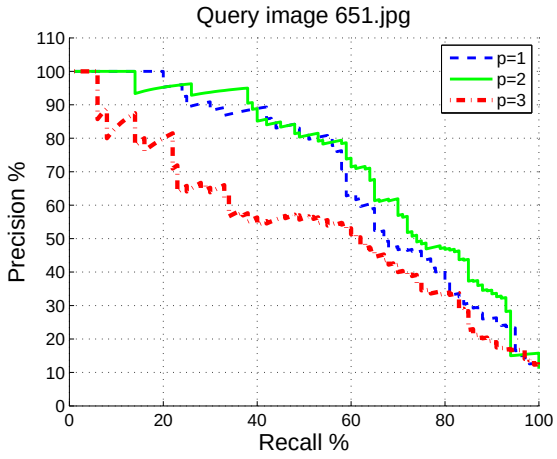


Figure 47: Plot of results from CBIR using distance measures other than the L^2 norm.

Curse of Dimensionality The curse of dimensionality refers the problem of distance between feature vectors in a high-dimensional hypercube. As reported in [115–117], any two randomly picked vectors in a high-dimensional hypercube tend to have a constant distance from each other due to this curse of dimensionality. Furthermore, with each additional dimension the spread of clusters in feature space is increased, but also the distance between two previously well-separated clusters is decreased. As a result, this "curse" could be a

problem for the near set approach, and definitely warrants future work. However, notice that the dimensionality of the approach taken in the thesis is low, only 18, and that a general rule of thumb is to pick a selection of important features rather than consider as many features as possible. Moreover, the FLANN library used in this thesis has been used with feature vectors that have dimensionality in the thousands [59].

Invariance Invariance is a property of a mathematical object that does not change when the object is transformed in some manner. A frequent use of invariant properties arises in image processing, where the desired output of a feature value, system, or process produces the same result if an image is, *e.g.*, translated, rotated, or scaled. Invariance is not a desirable property in the perception of objects. For instance, if some transformation is performed on the objects such that the probe functions used to observe the objects produce different values, then this transformation needs to be reflected in the distance measure. The near set approach does not start out with predefined classes of objects, and then define relations with specific measures, choose specific probe functions, or define nearness measures to conform to these groupings. Instead, the probe functions define our "view" of the objects, and this view determines the classes of objects that are similar. This is similar to our senses which define our perception of the world. For example our ability to view light in the visible spectrum rather than infra red or microwaves spectra defines our perception of our environment. In the case of near set theory, the probe functions were selected for a reason, *i.e.* their particular view of the objects is valuable. If a transformation occurs on the objects, such that the output of the probe functions is different, then this change needs to be reflected in assessing the similarity of the sets of objects. However, invariance could be further investigated in the near set approach to content-based image retrieval. In this case, it is desirable to return images that are similar to the query image except for some transformation like rotation or scaling of the objects in the image. This was one of the reasons for choosing to use Zernike moments which are invariant to rotation and reflection. Dealing with the problem of invariance in the near set approach to content-based image retrieval is a matter of choose probe functions that are invariant to a particular

transformation present in the application.

Image Noise

Image resolution and noise have not been addressed in this thesis, which are problems that arise in practical applications. Philosophically speaking, if the objects being compared are obtained from images that contain defects (such as noise, or blurring) and these defects are reflected in the probe functions, then this approach should assess other images containing defects (and similar content) as more similar than those images without. This is intended result and is not a short coming of the near set approach. Recall probe functions define our view of the objects, and are the basis for making judgements on similarity. An analogous example is our senses. Surely our senses have noise, *i.e.* we are not viewing our environment as it truly exists, however, we are still able to assess similarity based on the output of our senses. Practically speaking, this problem falls into the realm of image processing. If one has an image that contains defects, but wants to retrieve images similar to the content and without noise, then one must either pre-process the query image to remove the defects, or to use features like the ones reported in [56] which are robust to noise. Either way this an application specific problem (which is why it was not addressed in the thesis), but still warrants future investigation.

6 Conclusion

The focus of this research is on a tolerance space-based approach to image analysis. This approach was chosen because coverings of finite non-empty sets, determined by tolerance relations, tend to reveal resemblances between disjoint sets. In keeping with this insight about resemblance made by Sossinsky, this thesis introduces a tolerance near set approach that includes: the introduction of a nearness measure to determine the degree that near sets resemble each other (see, *e.g.* Section 4.3); a systematic approach to finding tolerance classes (see, *e.g.* Section 4.4), together with proofs demonstrating that the proposed approach will find all tolerance classes on a set of objects (also in Section 4.4); an approach to applying near set theory to images (see, *e.g.* Section 5 and Appendix A); the application of near set theory to the problem of content-based image retrieval (also in Section 5); demonstration that near set theory is well suited to solving problems in which the outcome is similar to that of human perception (see, results in Section 5.6); and two other near set measures are considered, one based on Hausdorff distance, the other based on Hamming distance (see, *e.g.* Section 5.4).

The results presented in Section 5 demonstrate that the near set approach is a powerful tool in content-based image retrieval applications applications, even for small values of ε. Moreover, these results suggest that the near set approach, and more specifically, the tolerance near-set approach, is suitable for applications where the desired outcome is close to the human perception of nearness, where a problem can be formulated in terms of sets of objects together with feature value vectors describing the objects. In fact, in terms of perception, the near set approach is advantageous, since it provides a formal framework for the quantification of the perception of similarities of objects based on a manner in which people perceive objects and the affinities between them, since people tend to grasp not single objects, but classes of them [45].

In summary, near sets themselves reflect human perception of nearness, *i.e.*, they resemble collections of object-comparison measurements that are byproducts of perceiving

and, possibly, recognizing objects in the environment. Although a consideration of human perception itself is outside the scope of this thesis, it should be noted that a rather common sense view of perception underlies the basic understanding of near sets (in effect, perceiving means identifying objects with common descriptions). And perception itself can be understood in Maurice Merleau-Ponty's sense, where perceptual objects are those objects captured by the senses. However, it is important to note that these methods are dependent on probe function selection and that poor selection of probe functions results in less perceptually relevant information. This is to be expected and is similar to the feature extraction problem in pattern recognition and other feature dependent disciplines. This thesis has demonstrated that near set theory is a powerful approach to solving problems based on the human perception of nearness.

A Other Applications of Near Sets

This section presents a new form of morphological image processing based on perception rather than geometry, which was introduced by Henry and Peters (see, *e.g.*, [35]). First, a review of traditional mathematical morphology is presented, followed by a discussion on perceptual morphology. Gonzalez and Woods [55] define morphology as the study of form and structure in complex biological organisms. Similarly, they define mathematical morphology in terms of set theory where the sets consist of tuples containing pixel coordinates in a binary image. In contrast, perceptual morphology, inspired by traditional morphology, is an approach where the operations are based on objects defined in near set sense.

A.1 Mathematical Morphology

The following review uses the same notation given in [55]. Mathematical morphology starts with the assumption that an image is a set of points (pixels), that are represented by two dimensional vectors. With this view of an image, it is then possible to consider set theory-based operations on images, namely, dilation $\oplus$ and erosion $\ominus$. Let the $\hat{\cdot}$ operator

represent the reflection of a set B, defined as

$$\hat{B} = \{w \mid w = -b, \forall\, b \in B\},$$

and let the operator $(\cdot)_z$ denote the translation of a set by point $z = (z_1, z_2)^{\mathrm{T}}$, that is,

$$(A)_z = \{c \mid c = a + z, \forall\, a \in A\}.$$

Then, the dilation of an image A by a structuring element B is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\},$$

and, similarly, an erosion operation $A \ominus B$ is defined as

$$A \ominus B = \{z \mid (B)_z \subseteq A\},$$

where A usually indicates the image to be transformed, and the structuring element B consists of a geometric arrangement of pixels in which the shape and binary value of the pixels play an important role in the transformation of A. A simple example of dilation and erosion is given in Fig. 48 (see both [55, 118] for more examples). Notice that dilation and erosion operators respectively cause the set A to grow and reduce in size, where the terms grow and reduce are defined with respect to area since the objects in the sets correspond to pixels coordinates.

(a) (b) (c)

Figure 48: Example of mathematical morphology. (a) Segment obtained from an image in the Berkeley Segmentation Dataset [97], (b) dilation of (a) using square a structuring element of width 5, and (c) erosion of (a) using the same structuring element. (Original unsegmented image used with permission [97])

A.2 Perceptual Morphology

If set theory is the language of mathematical morphology, then near set theory is language of perceptual morphology. The central idea between the two approaches is the same, *i.e.*, both methods increase or decrease set membership based on comparison with a structuring element. The difference lies in the definition of the objects, in particular, the perceptual approach defines objects in the near set sense. Thus, objects can be anything as long as it is possible to define probe functions that operate on the objects under consideration. In terms of image processing, the goals are the same. Mathematical morphology is used for identifying structure [55], whereas the near set approach is used to identify perceptual properties in images. The latter is similar to the former in that identification of structure facilitates human perception of images [13].

The proposed approach to perceptual morphology is based on mathematical morphology, yet it takes advantage of the perceptual information inherent to near set theory. Again, let the set of objects be represented by O, and the quotient set is $O_{/\sim_{\mathcal{B}}}$, where $\mathcal{B}$ is a set of probe functions on objects in O selected from $\mathbb{F}$. Now, define a set $A \subseteq O$ such that it has some *a priori* perceptual meaning associated with it, *i.e.* this set has definite meaning in a perceptual sense outside of the probe functions in $\mathcal{B}$. Next, let the quotient set represent the structuring element from traditional mathematical morphology, in other words let $B = O_{/\sim_{\mathcal{B}}}$[13]. As will be seen shortly, the quotient set is used as the structuring element in perceptual morphology, since it contains the perceptual information necessary to augment the set A in a perceptually meaningful way. This perceptual information is in the form of elementary sets (collections of objects with matching descriptions), since we perceive objects by the features that describe them and that people tend to grasp not single objects, but classes of them [45].

[13]The quotient set is being relabelled only to be notationally consistent with traditional mathematical morphology defined in Section A.1.

Keeping the above in mind, perception-based dilation is defined as

$$A \oplus B = \{x_{/\sim_{\mathcal{B}}} \in B \mid x_{/\sim_{\mathcal{B}}} \cap A \neq \emptyset\}, \tag{13}$$

and the perception-based erosion is defined as

$$A \ominus B = \bigcup_{x_{/\sim_{\mathcal{B}}} \in B} \{x_{/\sim_{\mathcal{B}}} \cap A\}. \tag{14}$$

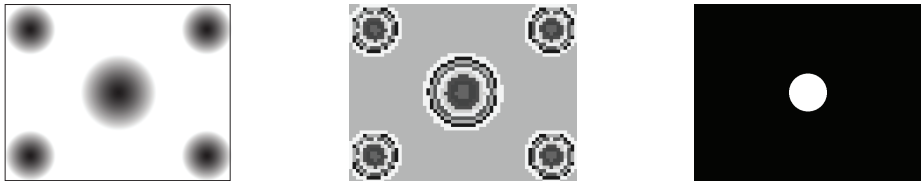
Notice, the set A is grown perceptually by the structuring element B (and consequently by the probe functions in $\mathcal{B}$) using the dilation operator by including objects in the result that have similar descriptions to those contained in A. In other words, the dilation operation perceptually enhances the set A by including the full membership of the elementary sets that have at least one object in A. Conversely, the erosion operation essentially masks the set B (using A) by including in the result only the portions of the elementary sets contained in A already. Thus, perceptual information can be reduced if the entire elementary set is not contained in the result.

By way of example the above concepts are illustrated with images. Let O contain subimages as set elements, *i.e.* perceptual objects, and let A be a subset of the subimages[14]. Then, the structuring element, $B = O_{/\sim_{\mathcal{B}}}$, can be viewed as an image where each class is assigned a unique colour (or grey value). Similarly, the results of dilation and erosion can be viewed as images as well where the objects in the result are assigned the same colour as the objects in B, and the rest of the image can be coloured white representing an absence of objects (since it is most likely the case that not all of the objects in O will be included in the result).

Fig. 49 is an example of these techniques applied to a simple greyscale image containing five circles in which a gradient operation from white to black was applied as shown in Fig. 49a. The gradient was used so that there is no crisp boundary to indicate the start

[14]Sometimes a set A will be given in terms of a segment of the original image (as in Fig. 49c). In this case it is necessary to rasterize the set by converting from a set with pixel granulation to a set with subimage granulation.

of the circle. Next, Fig. 49b contains the elementary sets created using a subimage size of 10×10 and $\mathcal{B} = \{\phi_{\text{avg}}(f_s)\}$ where each colour represents a different set. Notice that each circle has similar elementary sets since they are identical (except for the centre circle which is slightly larger). Next, Fig. 49c contains the set A representing *a priori* perceptual information, in this case it is a segment representing the point at which the centre circle is predominately black. The result of perceptually growing the segment using the dilation operator defined in Eq. 13 is given in Fig. 49d[15]. Notice the results show that the segment A was enlarged to include the other four circles that are perceptually similar with respect the probe functions in $\mathcal{B}$. Thus, more perceptual information was gained about the segment represented by A. Similarly, the result of perceptually reducing A using Eq. 14 is given in Fig. 49e. The perceptual information is reduced since all the information contained in $B = O_{/\sim_{\mathcal{B}}}$ is not being used. For instance, there is no representation of the other four circles in the result, which are perceptually similar to the centre circle. However, the operation is still useful in that it gives us perceptual information (in the form of elementary sets) about A with respect to the probe functions in $\mathcal{B}$.



(a) (b) (c)

(d) (e)

Figure 49: Example of perceptual morphology on simple greyscale image. (a) Original image, (b) quotient set of (a), (c) perceptual segmentation of the centre circle in (a), (d) perceptual dilation of (b), and (e) perceptual erosion of (b).

[15]Note, the white area of Fig.'s 49d & 49e and Fig.'s 50d & 50e do not represent an elementary set. This area is meant to be a background used to indicate a lack of elementary sets that are found in the same location in the image of the quotient set.

Another example of perceptual morphology is given in Fig. 50. In this case, Fig 50a contains an image from the Berkeley Segmentation Dataset [97] and Fig. 50c is a single ground truth segment from the same dataset. As before, Fig. 50b contains the elementary sets, this time created using a subimage of size 2×2 and $\mathcal{B} = \{\phi_{\text{IC}}(f_s), \phi_{\text{NormG}}(f_s)\}$. Notice, for the most part, the elementary sets represent perceptual concepts of the image, *e.g*, the horses tend to share the same elementary sets which differ from those of the background. Next, the set A represented by the segment of the younger horse is perceptually dilated in Fig. 50d. Again, the result now includes other areas of the image that are perceptually similar (with respect to probe functions in $\mathcal{B}$) including the other horse and parts of the background. Recall, that we may not find the background perceptually similar to the horse but the similarity occurs using only the probe functions in $\mathcal{B}$, *i.e.* information content and the normalized green value from the RGB colour model. Similar to the last example, the perceptual reduction caused by the erosion operator occurs due to the lack of inclusion of the full elementary sets in the result. Although, as was mentioned before, this result still contains perceptually valuable information of the original set A as will be seen in the next section.

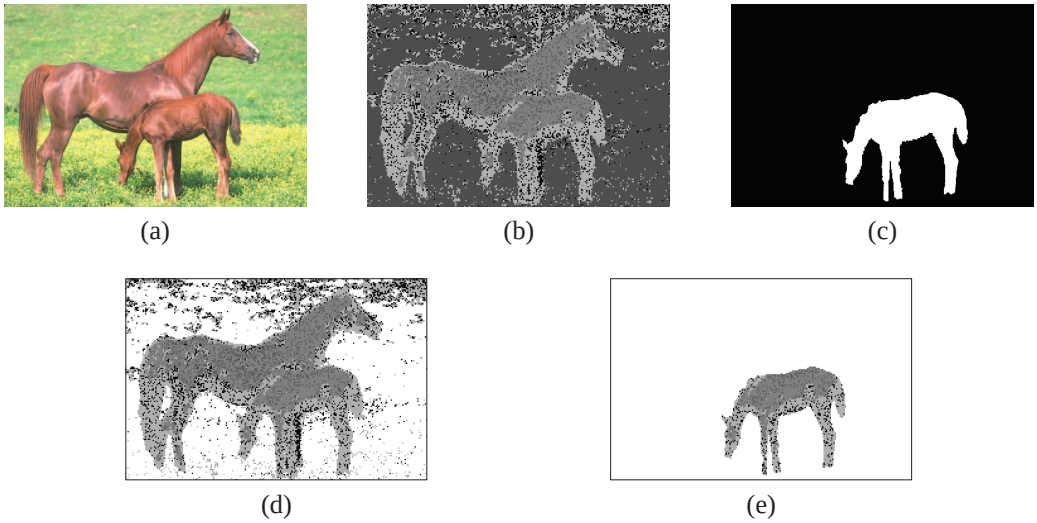


Figure 50: Example of perceptual morphology on image from Berkeley Segmentation Dataset [97]. (a) Original image (Used with permission [97]), (b) quotient set of (a), (c) perceptual segmentation of younger horse in (a), (d) perceptual dilation of (b), and (e) perceptual erosion of (b).

A.3 Segmentation Evaluation

This section presents an application of perceptual morphology in the form of a new method for segmentation evaluation called the Near Set Index (NSI) first introduced by Henry and Peters in [34, 35]. The NSI index was created out of a need for an unsupervised segmentation evaluation method that could be used in real world systems yet still based on human perception rather than characteristics of ideal segmentations that is popular in most unsupervised segmentation evaluation methods.

There are generally three classes of segmentation evaluation techniques, namely, analytic, empirical discrepancy, and empirical goodness methods [119, 120]. Analytic methods generally perform evaluation solely by examining the algorithm without considering the resultant segmentations. These methods are concerned with processing complexity and strategy and are not useful for evaluating the perceptual relevance of a given segmentation. The next category is empirical discrepancy (also called relative or supervised evaluation) and is characterized by the comparison of the test segmentation with ground truth images (segmentations performed by people or experts). These methods are popular because they evaluate segmentations based on the perceptual groupings created by people which is the end result of any segmentation algorithm. Unfortunately, it is not realistic to assume that systems incorporating image segmentation will have access to ground truth images and so these methods are generally used for the comparison of segmentation algorithms during the design phase. The last type of segmentation category is called empirical goodness (also known as stand-alone or unsupervised). These methods are based on some properties that ideal segmentations should contain. Generally, it is more difficult for these evaluation methods to be based on human perception of objects due to the lack of ground truth input or a formal framework for the quantification of perception and similarities of objects.

The are many examples of empirical goodness methods for segmentation evaluation. For example, [120] implements a co-evaluation framework in which multiple unsupervised methods are combined with learning algorithms to take advantage of different measures. Examples of the measures they consider are ones based on squared colour error ratio to

segment area [121, 122], on entropy (information content) of an image and the minimum description length principle (MDL) [123], on the geometric shape of a segment (*e.g.* compactness, circularity and elongation), and a contrast measure between the inside and outside of a segment [124]. Similarly, [125] present a review of six unsupervised methods based on image features such as segmentation contrast, standard deviation, and colour error. Likewise, [119] is another often sited survey of segmentation evaluation techniques also describing unsupervised measures where again the unsupervised methods are based solely on image characteristics. Notice that all these methods suffer because of the lack of the perceptual information contained in a ground truth image or the lack of a formal framework for the quantification of perception and similarity of objects introduced by near set theory. This is a problem we attempt to rectify with the introduction of the NSI.

A.4 Near Set Index

This section introduces a method for segmentation evaluation using perceptual morphology presented above. The goal of image segmentation is to partition an image into disjoint regions such that each one reflects some perceptual component of the image[16]. Since it has been observed that the quotient set captures perceptual information of objects [46], it makes sense to use the quotient set to measure the quality of a segmentation, *i.e.*, the degree to which a segmentation represents an image component.

Let f represent an RGB image and let A represent *a priori* information in the form of an image segment. Then, the result of perceptual erosion can be used to evaluate the quality of the segment since it only contains perceptual information about the set A. Further, since this set should represent a perceptual component within f and the quotient set represents perceptual information about the subimages in O, it should be possible to select probe functions in $\mathcal{B}$ such that the elementary sets begin to represent these components. As such, a good measure of segmentation quality is the variability of the classes contained in the

[16]Here we refer to the *objects* contained in images as components to avoid confusion with *objects* in the near set sense.

perceptual erosion of A. In general, the measure of variability of objects that take on labels from a discrete set is called the information content, and it takes on values in the interval $[0, \log_2 L]$ where L is the number of different labels the objects can assume [126, 127]. A value of 0 is produced when the objects contains all the same labels and the highest value occurs when each label occurs with equal frequency. Thus, for this application, low values of information content of the erosion of A corresponds to good segmentations and *vice versa*. This leads to the following definition:

Definition 26. Near Set Index *[35]. Let A represent a single image segment for evaluation, and let $B = O_{/\sim_{\mathcal{B}}}$ represent the quotient set obtained using the probe functions in $\mathcal{B}$. Then, the* Near Set Index *(NSI) is the information content of the perception-based erosion of A.*

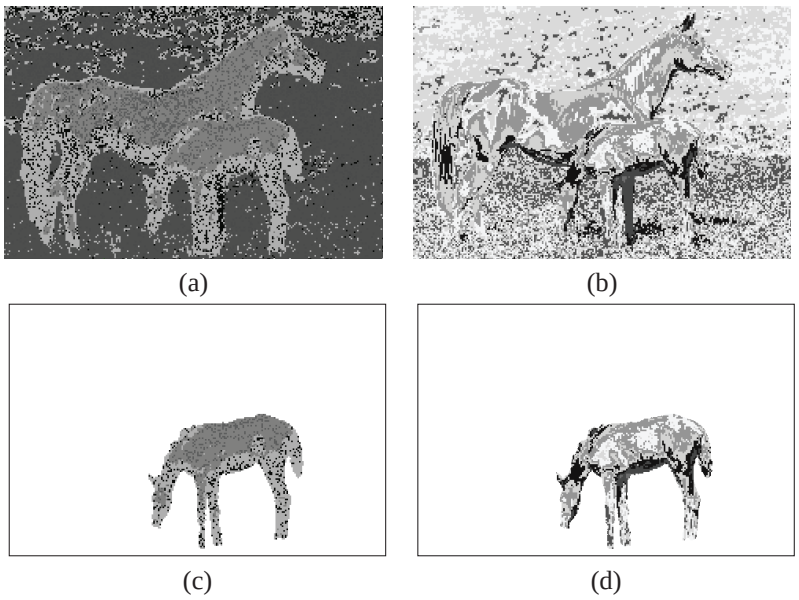


Figure 51: Example of quotient set and perception based erosion of Fig. 50a using different probe functions. (a) Fig. 50b created using probe functions $\mathcal{B} = \{\phi_{\text{IC}}(f_s), \phi_{\text{NormG}}(f_s)\}$ and repeated here for comparison, (b) quotient set obtained using $\mathcal{B} = \{\phi_{\text{avg}}(f_s)\}$, (c) perceptual based erosion using (a) as the SE, and (d) perceptual based erosion using (b) as the SE (Original unmodified image used with permission [97]).

A.5 Segmentation Evaluation Examples

To demonstrate these concepts two examples are given in Fig.'s 51 & 52 and the resultant information content is given in Table 6. The first example shows that poor choices of $\mathcal{B}$ lead to poor segmentations evaluations demonstrating this method is dependent on the selection of probe functions in the same manner that pattern recognition is dependent on feature selection. Notice that the elementary sets of Fig. 51a tend to capture the perceptual components of Fig. 50a better than those in Fig. 51b. Consequently, the information content of the erosion given in Fig. 51d is higher (and so worse) than that of Fig. 51c. Thus, the same segment can have different NSI values depending on the choice of $\mathcal{B}$. This example was given to highlight the need for careful probe function selection for a given application. The next example (given in Fig. 52) demonstrates the ability of the NSI to evaluate different segmentations. The segmentation given in Fig. 52a is the same as Fig. 50c shifted to the right, and the segmentation given in Fig. 52c was created by placing boxes of approximately the right size over the horse. Both of these segmentations are bad in the sense that they do not capture the perceptual component representing the smaller horse in the image as well as the segmentation in Fig. 50c. This is reflected by the information content values given in Table 6 which are higher than that of Fig. 51c.

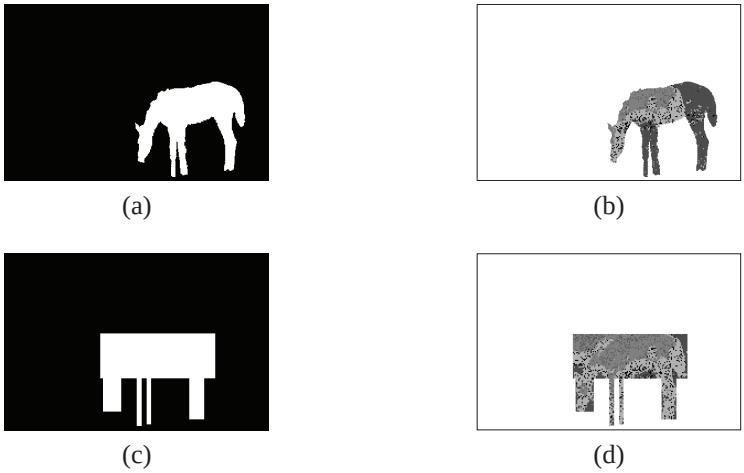


Figure 52: Segmentations of the smaller horse in Fig. 50a and their perception-based erosions. (a) Sample segmentation, (b) erosion of (a), (c) sample segmentation, and (d) erosion of (b) (Original unmodified image used with permission [97]).

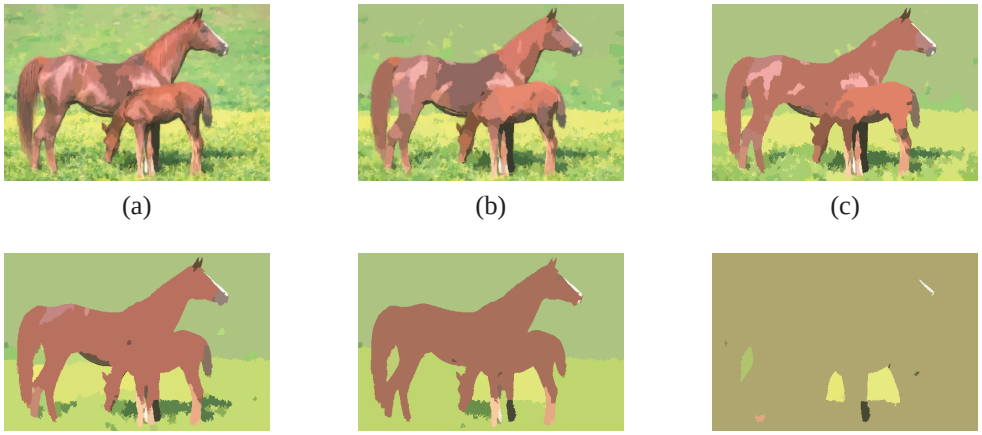
Table 6: Information content of perceptual-based erosion

Erosion image	Information content
Fig. 51c	1.6619
Fig. 51d	2.5682
Fig. 52b	2.1954
Fig. 52d	2.0797

Next, further demonstration of the ability of the NSI index to evaluate an image segmentation, S_{test}, is given by way of comparison with the normalized probabilistic rand index (see Section 2.3.5). But first the NSI must be extended to handle more than one segment. This is easily accomplished by letting the NSI of a proposed image segmentation be the information content of the worst segment in the image, *i.e.* let the NSI be the value of the highest information content resulting from the erosions of all the segments in S_{test}. The idea being that a proposed segmentation is only as good as its worst region. Using this approach, the NSI was used to evaluate the segmentations from an example given in [69] and repeated in Fig. 53. The results of evaluating the segmentations in Fig. 53 using both the NPR and the NSI index are reported in Fig. 54. Notice all three measures give similar evaluations of the segmentations of Fig. 53, *i*.e., Fig.'s 53c-53e are all ranked the highest (with Fig. 53e ranked the best). Additionally, the NSI index gives similar (poor) ratings for both the over and under segmentations shown in Fig.'s 53a & 53f, rather than treating one case much worse than the other (a rather nice result sine both are usually equally bad outcomes).

B Near System

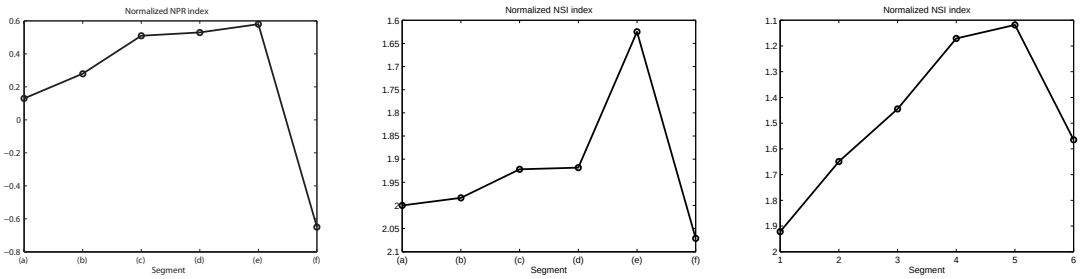
This section introduces the Near set Evaluation and Recognition (NEAR) system, a system developed to demonstrate practical applications of near set theory in the problems of image segmentation evaluation [34, 35] and image correspondence [92, 93]. This system was also used to demonstrate and visualize concepts from near set theory reported in [20, 23–25, 33,



(a) (b) (c)

(d) (e) (f)

Figure 53: Mean shift segmentation of Fig. 50a with h_s (spatial) = 7 and h_r (range) = 3,7,11,15,19,23 for (a), (b), (c), (d), (e), and (f) (Original unsegmented image used with permission).



(a) (b) (c)

Figure 54: Results of evaluating the segmentations given in Fig. 53: (a) NPR index, and (b), (c) NSI on a window size of 2×2 using respectively $\mathcal{B} = \{\phi_{\mathrm{H_{Shannon}}}(f_s), \phi_{\mathrm{NormG}}(f_s)\}$ and $\mathcal{B} = \{\phi_{\mathrm{H_{Pal}}}(f_s), \phi_{\mathrm{NormG}}(f_s)\}$.

36, 40, 48, 50]. It was motivated by a need for a freely available software tool (available at [42]) that can provide results for research and to generate interest in near set theory.

The functionality of the NEAR system is given in the following sections. This system implements a Multiple Document Interface (MDI) (see, *e.g.*, Fig. 55) where each separate processing task is performed in its own child frame. The perceptual objects in this system are subimages of the images being processed and the probe functions are image processing functions defined on the subimages. The system was written in C++ and was designed to facilitate the addition of new processing tasks and probe functions[17]. Currently, the sys-

[17] Parts of the Graphical User Interface (GUI) were inspired by the GUI reported in [66] and the wxWidgets example in [128].

tem performs siz major tasks: displaying equivalence and tolerance classes for an image; performing segmentation evaluation; measuring the nearness of two images; performing content-based image retrieval; and displaying the output of processing an image using individual probe functions.

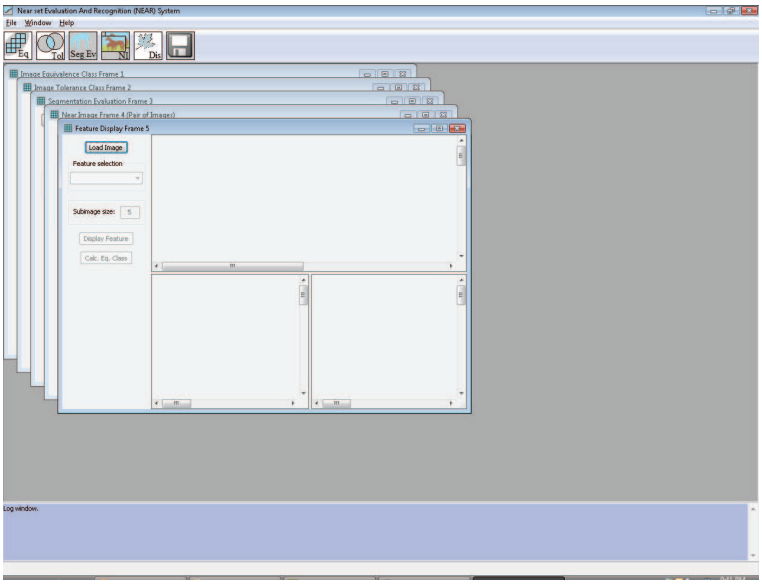


Figure 55: NEAR system GUI.

B.1 Equivalence class frame

This frame calculates equivalence classes using the Perceptual Indiscernibility relation in Definition 5, *i*.e., given an image X, it will calculate $X_{/\sim_B}$, where the objects are subimages of X (see Section 2.1 for further explanation). An example using this frame is given in Fig. 56 and is obtained by the following steps:

1. Click *Load Image* button and select an image.

2. Click the *Set Parameters* button.

3. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

4. Select number of features (maximum allowed is 24).

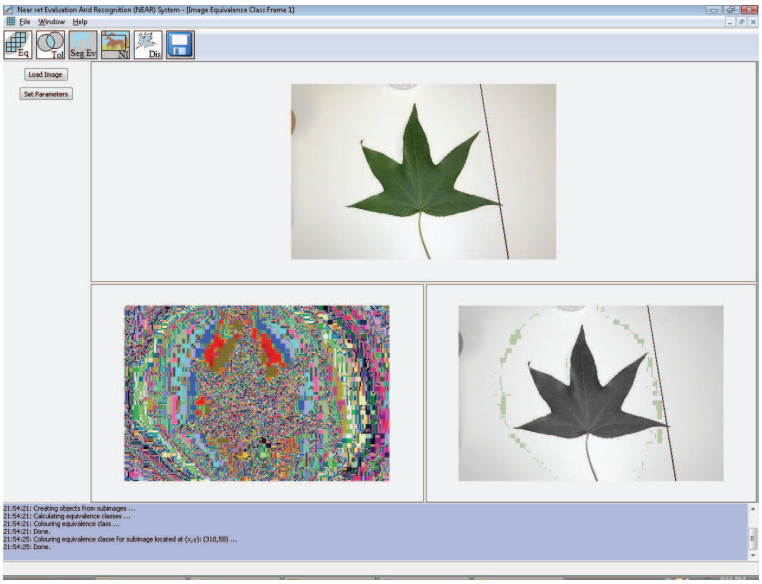


Figure 56: Sample run of the equivalence class frame using a window size of 2×2 and $\mathcal{B} = \{\phi_{\text{Average Grey}}$ (Image shown in NEAR system used with permission [98]).

5. Select features.

6. Click *Run*.

The result is given in Fig. 56, where the bottom left window contains an image of the equivalence classes, and each colour represents a single class. The bottom right window is used to display equivalence classes by clicking in any of the three images. The coordinates of the mouse click determine the equivalence class that is displayed. The results may be saved by clicking on the save button.

B.2 Tolerance class frame

This frame finds tolerance classes using the perceptual tolerance relation in Definition 18, *i*.e., given an image X, this frame finds $H_{\cong_{\mathcal{B},\varepsilon}}(O)$, where the objects are subimages of X (see Section 3 for further explanation). An example using this frame is given in Fig. 57 and is obtained by the following steps:

1. Click *Load Image* button and select an image.

2. Click the *Set Parameters* button.

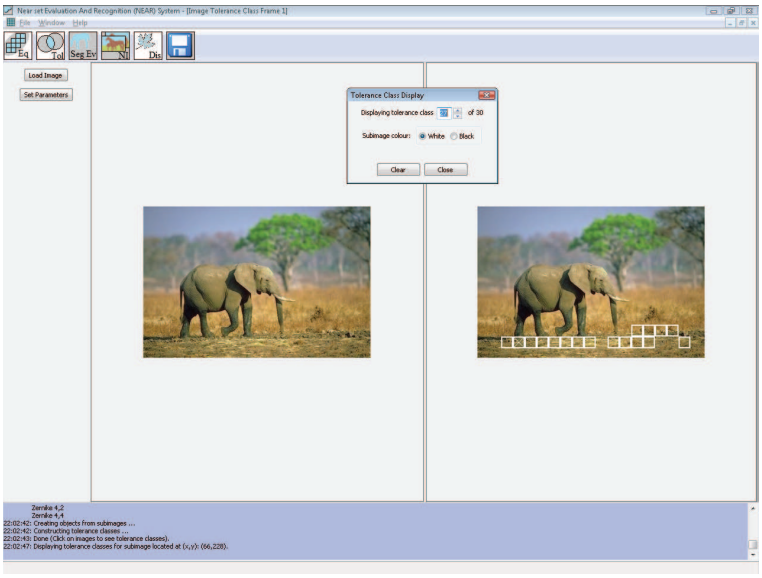


Figure 57: Sample run of the tolerance class frame using a window size of 20×20, 18 features used to generate the results in this thesis, and $\varepsilon = 0.7$ (Image shown in NEAR system used with permission [104, 105]).

3. Select window size. The value is taken as square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

4. Select ε, a value in the interval $[0, \sqrt{l}]$, where l is the number of features (length of object description).

5. Select number of features (maximum allowed is 24).

6. Select features.

7. Click on *FLANN Parameters* tab, and select the FLANN parameters for calculating tolerance classes.

8. Select ϵ, a value in the interval $[0, \sqrt{\text{Num. features}}]$.

9. Click *Run*.

The result is given in Fig. 57 where the left side is the original image, and the right side is used to display the tolerance classes. Since the tolerance relation covers an image instead of partitioning the image, the tolerance classes are displayed upon request. For

instance, by clicking on either of the two images, a window appears letting the user display each tolerance class containing the subimage selected by the mouse. Further, the subimage containing the mouse click contains an 'X', and the subimages can be coloured white or black.

B.3 Segmentation evaluation frame

This frame performs segmentation evaluation using perceptual morphology as described in Section A, where the evaluation is labelled the Near Set Index (NSI). For instance, given a set of probe functions $\mathcal{B}$, an image, and a segmentation of the image (labelled A), this frame can perform the perceptual erosion or dilation using $B = O_{/\sim_{\mathcal{B}}}$ as the structuring element. Also, the NSI is calculated if perceptual erosion was selected. A sample calculation using this frame is given in Fig. 58 and is obtained by the following steps:

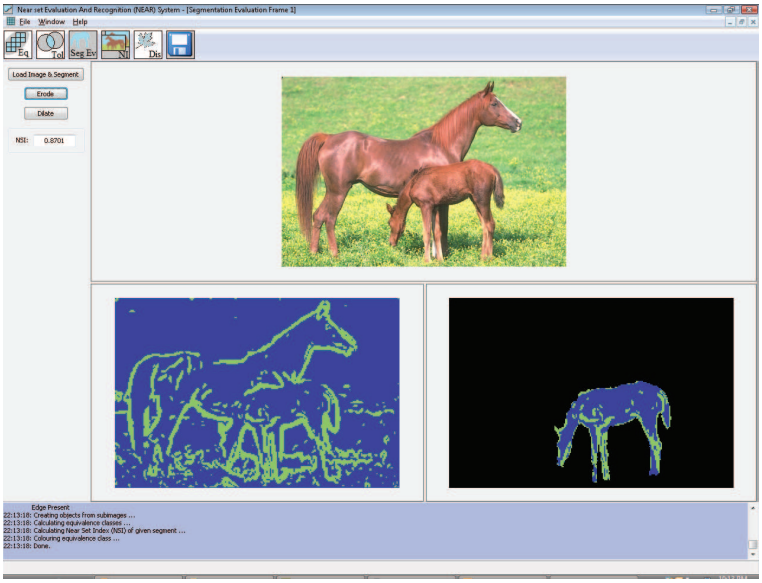


Figure 58: Sample run of the segmentation evaluation frame using a window size of 2×2, and $\mathcal{B} = \{\phi_{\text{Edge Present}}\}$ (Image shown in NEAR system used with permission [97]).

1. Click *Load Image & Segment* button.

2. Select an image click *Open*.

3. Select segmentation image and click *Open*. Image should contain only one segment and the segment must be white $(255, 255, 255)$ and the background must be black $(0, 0, 0)$. The image is displayed in the top frame, while the segment is displayed in the bottom right (make sure this is the case).

4. Click either *Erode* to perform perceptual erosion and segmentation evaluation, or *Dilate* to perform perceptual dilation (no evaluation takes place during dilation).

5. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

6. Select number of features (maximum allowed is 24).

7. Select features.

8. Click *Run*.

The result is given in Fig. 58 where the bottom left window contains the an image of the equivalence classes where each colour represents a different class. The bottom right window contains either the erosion or dilation of the segmentation. Clicking on any of the three images will display the equivalence class containing the mouse click in the bottom right image. The NSI is also displayed on the left hand side (if applicable).

B.4 Near image frame

This frame is used to calculate the similarity of images using the measures given in this thesis. The use has the option of comparing a pair of images (and viewing the resulting tolerance classes), or comparing a query image to an entire directory of images. The following two subsections outline the steps involved under both options.

B.4.1 Evaluating a pair of images

The steps involved in comparing a pair of images are as follows, and sample output for this process is given in Fig. 59.

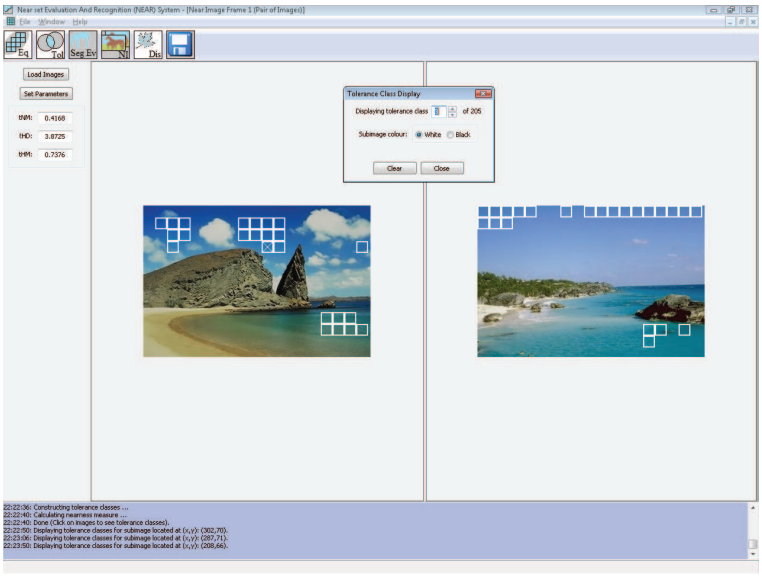


Figure 59: Sample run comparing a pair of images using a window size of 20×20, 18 features used to generate the results in this thesis, and $\varepsilon = 0.7$ (Image shown in NEAR system used with permission [104, 105]).

1. Select the *New near image window* icon, select File→New near image window, or press Alt+N.

2. Select *A pair of images* (the default value) from the *Select type of Comparison* window, and click OK.

3. Click *Load Images* button and select two images.

4. Click the *Set Parameters* button.

5. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

6. Select ε, a value in the interval $[0, \sqrt{l}]$, where l is the number of features (length of object description).

7. Select number of features (maximum allowed is 24).

8. Select features.

9. Click on *FLANN Parameters* tab, and select the FLANN parameters for calculating tolerance classes.

10. Click *Run*.

The result is given in Fig. 59 where the left side contains the first image, and the right side contains the second image. Clicking in any of the two images will bring up a window that allows the user to view each tolerance class containing the subimage selected by the mouse. Further, the subimage containing the mouse click is marked with an 'X', and the subimages can be coloured white or black. Also, the similarity of the images is evaluated using the measures described in this thesis, where the results are displayed on the left hand side.

B.4.2 Comparing a query image with a directory of images

The steps involved in comparing a query image with a directory containing images is as follows.

1. Select the *New near image window* icon, select File→New near image window, or press Alt+N.

2. Select *Query image with a directory of images* from the *Select type of Comparison* window, and click OK.

3. Click *Load Query Image + Dir.* button and select an image plus a directory containing images for comparison with query image.

4. Click the *Set Parameters* button.

5. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

6. Select ε, a value in the interval $[0, \sqrt{l}]$, where l is the number of features (length of object description).

7. Select number of features (maximum allowed is 24).

8. Select features.

9. Click on *FLANN Parameters* tab, and select the FLANN parameters for calculating tolerance classes.

10. Click *Run*.

The result is the left side contains the query image, and the right side contains an image from the directory. Clicking in any of the two images will bring up a window that allows the user to view the images from the directory in the order they were ranked by the selected similarity measure. In addition, three output files are created containing the similarity measure of each image in the database, sorted from most similar to least similar. Finally, three figures are also displayed plotting the similarity measures vs. images in the directory for all three measures. Note, the results are sorted from best to worst, so the output files are also required to relate the abscissae to actual image files.

B.5 Feature display frame

This frame is used to display the output of processing an image with a specific probe function. An example using this frame is given in Fig. 60 and is obtained by the following steps:

1. Click *Load Image* button and select an image.

2. Select features.

3. Select probe function.

4. Click *Display feature*.

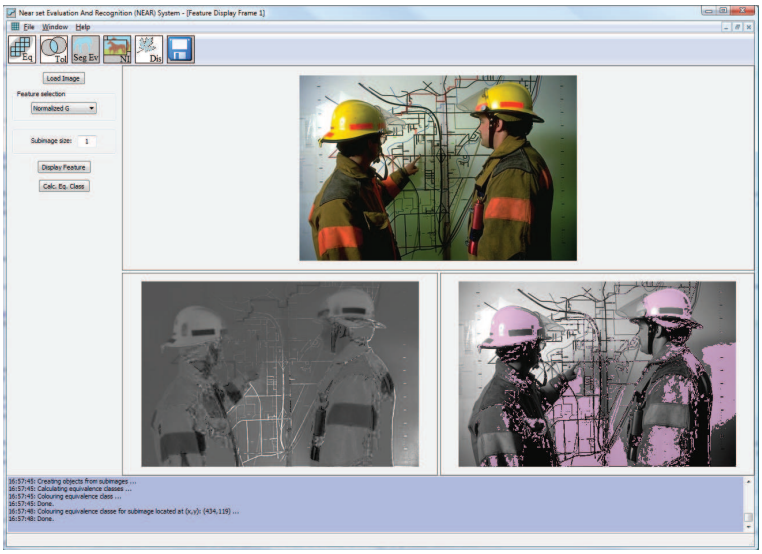


Figure 60: Sample run of the feature display frame (Image shown in NEAR system used with permission [97]).

References

[1] M. Merleau-Ponty, *Phenomenology of Perception*. Paris and New York: Smith, Callimard, Paris and Routledge & Kegan Paul, 1945.

[2] D. Calitoiu, B. J. Oommen, and D. Nussbaum, "Desynchronizing a chaotic pattern recognition neural network to model inaccurate perception," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 3, pp. 692–704, 2007.

[3] M. Fahle and T. Poggio, *Perceptual Learning*. Cambridge, MA: The MIT Press, 2002.

[4] E. D. Montag and M. D. Fairchild, "Pyschophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 977–989, 1997.

[5] I. El-Naqa, Y. Yang, N. P. Galatsanos, R. M. Nishikawa, and M. N. Wernick, "A similarity learning approach to content-based image retrieval: application to digital mammography," *IEEE Transactions on Medical Imaging*, vol. 23, no. 10, pp. 1233–1244, 2004.

[6] M. Rahman, P. Bhattacharya, and B. C. Desai, "A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, no. 1, pp. 58–69, 2007.

[7] J. I. Martinez, A. F. G. Skarmeta, and J. B. Gimeno, "Fuzzy approach to the intelligent management of virtual spaces," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 36, no. 3, pp. 494–508, 2005.

[8] V. Bruce, P. R. Green, and M. A. Georgeson, *Visual perception: physiology, psychology, and ecology*. Hove, East Sussex, UK: Psychology Press, 1996.

[9] T. V. Papathomas, R. S. Kashi, and A. Gorea, "A human vision based computational model for chromatic texture segregation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 27, no. 3, pp. 428–440, 1997.

[10] A. Mojsilovic, H. Hu, and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1238–1248, 2002.

[11] N. Balakrishnan, K. Hariharakrishnan, and D. Schonfeld, "A new image representation algorithm inspired by image submodality models, redundancy reduction, and learning in biological vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1367–1378, 2005.

[12] A. Qamra, Y. Meng, and E. Y. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 379–391, 2005.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assesment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[14] L. Dempere-Marco, H. Xiao-Peng, S. L. S. MacDonald, S. M. Ellis, D. M. Hansell, and G.-Z. Yang, "The use of visual search for knowledge gathering in image decision support," *IEEE Transactions on Medical Imaging*, vol. 21, no. 7, pp. 741–754, 2002.

[15] S. Kuo and J. D. Johnson, "Spatial noise shaping based on human visual sensitivity and its application to image coding," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 509–517, 2002.

[16] B. A. Wandell, A. El Gamal, and B. Girod, "Common principles of image acquisition systems and biological vision," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 5–17, 2002.

[17] T. A. Wilson, S. K. Rogers, and M. Kabrisky, "Perceptual-based image fusion for hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 1007–1017, 1997.

[18] A. Hoogs, R. Collins, R. Kaucic, and J. Mundy, "A common set of perceptual observables for grouping, figure-ground discrimination, and texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 458–474, 2003.

[19] N. G. Bourbakis, "Emulating human visual perception for measuring difference in images using an SPN graph approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 2, pp. 191–201, 2002.

[20] J. F. Peters, "Classification of objects by means of features," in *Proceedings of the IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SCCI 2007)*, Honolulu, Hawaii, 2007, pp. 1–8.

[21] ——, "Near sets. General theory about nearness of objects," *Applied Mathematical Sciences*, vol. 1, no. 53, pp. 2609–2629, 2007.

[22] J. F. Peters, A. Skowron, and J. Stepaniuk, "Nearness of objects: Extension of approximation space model," *Fundamenta Informaticae*, vol. 79, no. 3-4, pp. 497–512, 2007.

[23] J. F. Peters, "Discovery of perceptually near information granules," in *Novel Developments in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation*, J. T. Yao, Ed. Hersey, N.Y., USA: Information Science Reference, 2009, p. *in press*.

[24] J. F. Peters and S. Ramanna, "Affinities between perceptual granules: Foundations and perspectives," in *Human-Centric Information Processing Through Granular Modelling*, A. Bargiela and W. Pedrycz, Eds. Berlin: Springer-Verlag, 2009, pp. 49–66.

[25] J. F. Peters and P. Wasilewski, "Foundations of near sets," *Elsevier Science*, vol. 179, no. 1, pp. 3091–3109, 2009.

[26] Z. Pawlak, "Classification of objects by means of attributes," Institute for Computer Science, Polish Academy of Sciences, Tech. Rep. PAS 429, 1981.

[27] ——, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, pp. 341–356, 1982.

[28] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, pp. 3–27, 2007.

[29] ——, “Rough sets: Some extensions,” *Information Sciences*, vol. 177, pp. 28–40, 2007.

[30] ——, “Rough sets and boolean reasoning,” *Information Sciences*, vol. 177, pp. 41–73, 2007.

[31] J. F. Peters, S. Shahfar, S. Ramanna, and T. Szturm, “Biologically-inspired adaptive learning: A near set approach,” in *Frontiers in the Convergence of Bioscience and Information Technologies*, Korea, 2007.

[32] J. F. Peters and S. Ramanna, “Feature selection: A near set approach,” in *ECML & PKDD Workshop in Mining Complex Data*, Warsaw, 2007, pp. 1–12.

[33] A. E. Hassanien, A. Abraham, J. F. Peters, G. Schaefer, and C. Henry, “Rough sets and near sets in medical imaging: A review,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 3, no. 6, pp. 955–968, 2009.

[34] C. Henry and J. F. Peters, “Near set index in an objective image segmentation evaluation framework,” in *Proceedings of the GEOgraphic Object Based Image Analysis: Pixels, Objects, Intelligence*, University of Calgary, Alberta, 2008, pp. 1–8.

[35] ——, “Perception image analysis,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 3/4, pp. 271–281, 2010.

[36] J. F. Peters, “Tolerance near sets and image correspondence,” *International Journal of Bio-Inspired Computation*, vol. 1, no. 4, pp. 239–245, 2009.

[37] A. H. Meghdadi, J. F. Peters, and S. Ramanna, “Tolerance classes in measuring image resemblance,” in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 5712, Santiago, Chile, 2009, pp. 127–134.

[38] J. F. Peters, L. Puzio, and T. Szturm, “Measuring nearness of rehabilitation hand images with finely-tuned anisotropic wavelets,” in *Image Processing & Communication Challenges*, R. S. Choraś and A. Zabludowski, Eds. Warsaw: Academy Publishing House, 2009, pp. 342–349.

[39] C. Henry, “Near set Evaluation And Recognition (NEAR) system,” in *Rough Fuzzy Analysis Foundations and Applications*, S. K. Pal and J. F. Peters, Eds. CRC Press, Taylor & Francis Group, 2010, p. *accepted*, iSBN 13: 9781439803295.

[40] J. F. Peters, “Classification of perceptual objects by means of features,” *International Journal of Information Technology & Intelligent Computing*, vol. 3, no. 2, pp. 1 – 35, 2008.

[41] A. Skowron and J. F. Peters, “Rough-granular computing,” in *Handbook on Granular Computing*, V. K. W. Pedrycz and A. Skowron, Eds. N.Y., U.S.A: John Wiley & Sons, Inc., 2008, pp. 285–328.

[42] J. F. Peters, “Computational intelligence laboratory,” 2009, http://wren.ece.umanitoba.ca/.

[43] S. A. Naimpally, “Near and far. A centennial tribute to Frigyes Riesz,” *Siberian Electronic Mathematical Reports*, vol. 6, pp. A.1–A.10, 2009.

[44] S. A. Naimpally and B. D. Warrack, “Proximity spaces,” in *Cambridge Tract in Mathematics No. 59*. Cambridge, UK: Cambridge University Press, 1970.

[45] E. Orłowska, “Semantics of vague concepts. Applications of rough sets,” Institute for Computer Science, Polish Academy of Sciences, Tech. Rep. 469, 1982.

[46] ——, “Semantics of vague concepts,” in *Foundations of Logic and Linguistics. Problems and Solutions*, G. Dorn and P. Weingartner, Eds. London/NY: Plenum Pres, 1985, pp. 465–482.

[47] Z. Pawlak and J. F. Peters, “Jak blisko (how near),” *Systemy Wspomagania Decyzji*, vol. I, pp. 57, 109., 2002.

[48] J. F. Peters, “Near sets. Special theory about nearness of objects,” *Fundamenta Informaticae*, vol. 75, no. 1-4, pp. 407–433, 2007.

[49] S. Gupta and K. Patnaik, “Enhancing performance of face recognition systems by using near set approach for selecting facial features,” *Journal of Theoretical and Applied Information Technology*, vol. 4, no. 5, pp. 433–441, 2008.

[50] C. Henry and J. F. Peters, “Image pattern recognition using near sets,” in *Proceedings of the Eleventh International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computer (RSFDGrC 2007), Joint Rough Set Symposium (JRS07), Lecture Notes in Computer Science*, vol. 4482, 2007, pp. 475–482.

[51] S. Ramanna and A. H. Meghdadi, “Measuring resemblances between swarm behaviours: A perceptual tolerance near set approach,” *Fundamenta Informaticae*, vol. 95, pp. 533–552, 2009.

[52] M. Pavel, *Fundamentals of Pattern Recognition*. NY: Marcel Dekker, Inc., 1993.

[53] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley, 2001.

[54] E. Orłowska, “Incomplete information: Rough set analysis,” in *Studies in Fuzziness and Soft Computing 13*. Heidelberg, Germany: Physica-Verlag, 1998.

[55] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Toronto: Prentice-Hall, 2002.

[56] M. Pawlak, *Image analysis by moments: reconstruction and computational aspects*. Wrocław: Wydawnictwo Politechniki, 2006.

[57] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal, 2009, pp. 331–340.

[58] G. Shaknarovich, T. Darrel, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. The MIT Press, 2006.

[59] M. Muja, “FLANN - Fast Library for Approximate Nearest Neighbors,” 2009, http://www.cs.ubc.ca/ mariusm/index.php/FLANN/FLANN.

[60] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[61] J. Marti, J. Freixenet, J. Batlle, and A. Casals, “A new approach to outdoor scene description based on learning and top-down segmentation,” *Image and Vision Computing*, vol. 19, pp. 1041–1055, 2001.

[62] T. M. Cover and J. A. Thomas, "Elements of information theory." New York: John Wiley & Sons, Inc., 1991.

[63] N. R. Pal and S. K. Pal, "Entropy: A new definition and its applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 5, pp. 1260 – 1270, 1991.

[64] ——, "Some properties of the exponential entropy," *Information Sciences*, vol. 66, pp. 119–137, 1992.

[65] D. Comaniciu, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[66] C. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 4, Quebec City, 2002, pp. 150–156.

[67] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, 1992.

[68] S. Mallat, *A Wavelet Tour of Signal Processing*. California: Academic Press, 1999.

[69] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, 2007.

[70] H. Tamura, M. Shunji, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[71] R. M. Haralick, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[72] ——, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[73] P. Howarth and S. Ruger, "Robust texture features for still-image retrieval," *IEE Proceedings Vision, Image, & Signal Processing*, vol. 152, no. 6, pp. 868–874, 2005.

[74] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ, USA: Person/Prentice Hall, 2008.

[75] M. J. Black and B. B. Kimia, "Guest editorial: Computational vision at brown," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 5–11, 2003.

[76] D. G. Kendall, D. Barden, T. K. Crane, and H. Le, *Shape and Shape Theory*. Chichester, England: John Wiley & Sons Ltd, 1999.

[77] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1988.

[78] A. Khotanzad and Y. H. Hong, "Invariant image reconstruction by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[79] W. Y. Kim and Y. S. Kim, "A region-based shape descriptor using Zernike moments," *Signal Processing: Image Communication*, vol. 16, pp. 95–102, 2000.

[80] R. S. Choraś, T. Andrysiak, and M. Choraś, "Integrated color, texture and shape information for content-based image retrieval," *Pattern Analysis & Applications*, vol. 10, no. 4, pp. 333–343, 2007.

[81] P. Toharia, O. D. Robles, A. Rodriguez, and L. Pastor, "A study of Zernike invariants for content-based image retrieval," in *Advances in Image and Video Technology*. Berlin Heidelberg: Springer-Verlag, 2007, vol. LNCS 4872, pp. 944–957.

[82] J. M. Kasson and W. Plouffe, "An analysis of selected computer interchange color spaces," *ACM Transactions on Graphics*, vol. 11, no. 4, pp. 373–405, 1992.

[83] K. Nallaperumal, M. S. Banu, and C. C. Christiyana, "Content based image indexing and retrieval using color descriptor in wavelet domain," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 3, 2007, pp. 185–189.

[84] A. B. Sossinsky, "Tolerance space theory and some applications," *Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications*, vol. 5, no. 2, pp. 137–167, 1986.

[85] H. Poincaré, *Science and Hypothesis*. Brock University: The Mead Project, 1905, L. G. Ward's translation.

[86] L. T. Benjamin, Jr., *A Brief History of Modern Psychology*. Malden, MA: Blackwell Publishing, 2007.

[87] B. R. Hergenhahn, *An Introduction to the History of Psychology*. Belmont, CA: Wadsworth Publishing, 2009.

[88] G. T. Fechner, *Elements of Psychophysics, vol. I*. London, UK: Holt, Rinehart & Winston, 1966, H. E. Adler's trans. of Elemente der Psychophysik, 1860.

[89] H. Poincaré, *Mathematics and Science: Last Essays*. N. Y.: Kessinger Publishing, 1963, J. W. Bolduc's trans. of Dernières Pensées, 1913.

[90] E. C. Zeeman, "The topology of the brain and the visual perception," in *Topoloy of 3-manifolds and selected topices*, K. M. Fort, Ed. New Jersey: Prentice Hall, 1965, pp. 240–256.

[91] J. F. Peters, "Corrigenda and addenda: Tolerance near sets and image correspondence," *International Journal of Bio-Inspired Computation*, vol. 2, no. 5, p. *in press*, 2010.

[92] C. Henry and J. F. Peters, "Perception based image classification," Computational Intelligence Laboratory, University of Manitoba, Tech. Rep., 2009, uM CI Laboratory Technical Report No. TR-2009-016.

[93] ——, "Perception-based image classification," *International Journal of Intelligent Computing and Cybernetics*, p. *accepted*, 2010.

[94] J. F. Peters and L. Puzio, "Anisotropic wavelet-based image nearness measure," *International Journal of Computational Intelligence Systems*, vol. 2-3, pp. 168–183, 2009.

[95] J. F. Peters, "Fuzzy sets, near sets, and rough sets for your computational intelligence toolbox," in *Foundations of Compuational Intelligence*, A. Abraham and F. Herrera, Eds. Springer, Heidelberg, 2009, vol. 2.

[96] ——, "Discovering affinities between perceptual granules. l_2 norm-based tolerance near preclass approach," *Advances in Man-Machine Interactions and Soft Computing*, vol. 59, pp. 43–54, 2009.

[97] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Visison*, vol. 2, 2001, pp. 416–423, Database URL: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/.

[98] M. Weber, "Leaves dataset: Images taken in and around caltech." Computational Vision at California Institute of Technology, 2003, Permission recived July 2008. Database URL: www.vision.caltech.edu/archive.html.

[99] T. Szturm, J. F. Peters, C. Otto, N. Kapadia, and A. Desai, "Task-specific rehabilitation of finger-hand function using interactive computer gaming," *Archives of Physical Medicine and Rehabilitation*, vol. 89, no. 11, pp. 2213–2217, 2008.

[100] F. Hausdorff, *Grundzüge der mengenlehre*. Leipzig: Verlag Von Veit & Comp., 1914.

[101] ——, *Set theory*. New York: Chelsea Publishing Company, 1962.

[102] W. Rucklidge, *Efficient Visual Recognition Using Hausdorff Distance*. Springer-Verlag, 1996.

[103] M. A. Ferrer, A. Morales, and L. Ortega, "Infrared hand dorsum images for identification," *IET Electronic Letters*, vol. 45, no. 6, pp. 306–308, 2009.

[104] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001, Permission recived May 2010. Database URL: http://wang.ist.psu.edu/docs/related/.

[105] J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003, Permission recived May 2010. Database URL: http://wang.ist.psu.edu/docs/related/.

[106] A. Grigorova, F. G. B. De Natale, C. Dagli, and T. S. Huang, "Content-based image retrieval by feature adaptation and relevance feedback," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1183–1192, 2007.

[107] J. C. Caicedo, F. A. González, E. Triana, and E. Romero, "Design of medical image database with content-based retrieval capabilities," in *Advances in Image and Video Technology*, D. Mery and L. Rueda, Eds. Berlin: Springer-Verlag, 2007, vol. LNCS 4872, pp. 919–931.

[108] K. Zagoris, "img(Anaktisi)," 2010, http://orpheus.ee.duth.gr/anaktisi/.

[109] S. A. Chatzichristofis and Y. S. Boutalis, "CEDD: Color and edge directivity descriptor - a compact descriptor for image indexing and retrieval," in *Proceedings of the 6th International Conference in Advance Research on Computer Vision Systems ICVS 2008*, ser. Lecture Notes in Computer Science (LNCS). Santorini, Greece: Springer, 2008, pp. 312–322.

[110] ——, "FCTH: Fuzzy color and texture histogram - a low level feature for accurate image retrieval," in *Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Services*. Klagenfurt, Austria: IEEE Computer Society, 2008.

[111] S. A. Chatzichristofis and A. Arampatzis, "Late fusion of compact composite descriptors for retrieval from heterogeneous image databases," in *Proceedings of the 5th International Multi-Conference on Computing in the Global Information Technology, ICCGI*. IEEE Computer Society, 2010.

[112] S. Kiranyaz, "Advanced techniques for content-based management of multimedia databases," Ph.D. dissertation, Tampere University of Technology, Finland, 2005.

[113] E. Guldogan, "Improving content-based image indexing and retrieval performance," Ph.D. dissertation, Tampere University of Technology, Finland, 2009.

[114] M. Gabbouj, "MUVIS a system for content-based indexing and retrieval in multimedia databases," 2010, http://muvis.cs.tut.fi/index.html.

[115] G. Zervas and S. M. Ruger, "The curse of dimensionality and document clustering," in *IEE Colloguium on Microengineering in Optics and Optoelectronics*, vol. 187, 1999, pp. 19/1–19/3.

[116] K. Beyer and J. Goldstein, "When is nearest neighbor meaningful," in *International Conference on Database Theory*, 1999.

[117] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Proceedings of the ICDT Conference*, 2001, pp. 420–434.

[118] E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing*. SPIE Press, 2003, vol. TT59.

[119] Y. J. Zhang, "A survey on evaluation methods for image segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335–1346, 1996.

[120] H. Zhang, J. E. Fritts, and S. A. Goldman, "A co-evaluation framework for improving segmentation evaluation," in *SPIE Defense and Security Symposium - Signal Processing, Sensor Fusion, and Target Recognition XIV*, 2005, pp. 420–430.

[121] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative evaluation of color image segmentation results," *Pattern Recognition Leters*, vol. 19, pp. 741–747, 1998.

[122] J. Liu and Y.-H. Yang, "Multi-resolution color image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7, pp. 689–700, 1994.

[123] H. Zhang, J. E. Fritts, and S. A. Goldman, "A fast texture feature extraction method for region-based image segmentation," in *Proceedings of IS&T/SPIE's 16th Annual Symposium on Image and Video Communication and Processing*, vol. 5685, 2005.

[124] P. L. Correia and F. Pereira, "Objective evaluation of video segmentation quality," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 156–200, 2003.

[125] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger, and P. Marche, "Unsupervised evaluation of image segmentation application to multi-spectrual images," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 1, 2004, pp. 576 – 579.

[126] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. UK: Cambridge University Press, 2003.

[127] T. Seemann, "Digital image processing using local segmentation," Ph.D. dissertation, School of Computer Science and Software Engineering, Monash University, 2002.

[128] wxWidgets, "wxwidgets cross-platform gui library v2.8.9," 2009, www.wxwidgets.org.

Author Index

Subject Index