Near Images: A Tolerance Based Approach to Image Similarity and its Robustness to Noise and Lightening

by

Shabnam Shahfar

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

Master of Science
in
Electrical and Computer Engineering

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

Near Images: A Tolerance Based Approach to Image Similarity and Its Robustness to Noise and Lightening

by

Shabnam Shahfar

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

Master of Science
in
Electrical and Computer Engineering

Abstract

This thesis represents a tolerance near set approach to detect similarity between digital images. Two images are considered as sets of perceptual objects and a tolerance relation defines the nearness between objects. Two perceptual objects resemble each other if the difference between their descriptions is smaller than a tolerable level of error. Existing tolerance near set approaches to image similarity consider both images in a single tolerance space and compare the size of tolerance classes. This approach is shown to be sensitive to noise and distortions. In this thesis, a new tolerance-based method is proposed that considers each image in a separate tolerance space and defines the similarity based on differences between histograms of the size of tolerance classes. The main advantage of the proposed method is its lower sensitivity to distortions such as adding noise, darkening or brightening. This advantage has been shown here through a set of experiments.

Keywords: Image similarity measure, perceptual systems, near sets, tolerance spaces, tolerance near sets, histogram-based similarity measures, near images, nearness measure, probe function, feature vector.

Acknowledgments

Many people have helped me during this journey and I would like to thank all of them.

- I am grateful to my advisor, Professor James Peters, for all his support and encouragement, and all the good experience and knowledge that he has passed to me through these years. Without those this journey wouldn't be as pleasant as it was.
- Special thanks to Amir Meghdadi, my husband, for thoughtful research discussions as well as his never ending support and love.
- I would also like to thank Maciej Borkowski, Chris Henry, Dan Lockery, and Homa Fashandi for all the good discussions that we had in the lab and for all their help during my master's studies.
- Many thanks to my parents and my family for their believe in me and for their love and encouragements.
- And last but not least, to the University of Manitoba Graduate Fellowship program for their financial support.

Contents

List of Tables

List of Figures

1 Introduction

Digital imaging and digital image processing have a history of more than 4 decades. Also, computational and storage power of digital computers made it possible to archive, manage and analyze digital images using mathematical algorithms in digital image precessing. Furthermore, the rapid growth of the Internet users and particularly recent popularity of social media were important factors to grow the number of digital images. Photo sharing web service Flickr® alone have received more than 5 billion images uploads, since the launch of this web site in 2004.[5] This huge amount of digital images and their usage is the reason why it is important to develop computational methods for analysis, management and classification of images and the challenging problems that will arise.

One such problem which is considered in this thesis, is how to define a measure of similarity between two digital images. Different approaches may be taken to define the similarity between two images based on visual features such as color and texture. The main approach in this thesis is based on tolerance near set theory and statistical analysis of the size of tolerance classes. Recently, it has been shown that Near set theory [26, 27, 23] can be used in a tolerance space in order to discover affinities between images [7, 8, 33]. Almost all of the different methods for defining similarity involve the following two steps:

- Step 1: Extracting some features of the images and constructing some feature vectors (feature extraction).
- Step 2: Defining the similarity using the given feature vectors (similarity measure).

The focus of this thesis is on the second step and the research is not concerned with finding or extracting the best possible features. This is especially important because the optimal features in each case depend on the application and cannot be generally investigated. Once the feature vectors are obtained, there are also different approaches for defining the similarity between two images (step 2). Traditionally, there are two main approaches [42]. Each approach is supported by different psychological models of how humans perceive similarity:

1. The first approach is to define similarity as a function of distance between feature vectors such that higher values of distance gives lower values of similarity. The type of distance that is used can be anything as simple as the Euclidean distance or other norm based distances [13, 48]. Sometimes other forms of distances such as Hausdorff distance can be used [38]. A Hausdorff distance can be used to compare binary images (as commonly is used, see *e.g* [9]) or to compare sets of feature vectors in feature space as used in this thesis.

2. The second approach is a probabilistic approach where the similarity depends on the distribution of the feature values and hence is a function of the difference between histograms. See *e.g.* [2, 51, 47]. Once the histograms are obtained, one may view the histogram as a set of features and hence this approach can be interpreted as the first approach when feature vectors have been replaced with histograms. This approach has been used many times in literature concerning content based image retrieval that rely on image similarity [42, 41].

Some authors have used both approaches [10]. Some authors also have reported other methods such as clustering algorithm as well as an image recognition neural network [51]. Fuzzy set based approaches have also been used [46]. More information about the existing methods in image similarity can be found in [42, 47, 52].

The key idea in this thesis however, is based on comparing images in a tolerance space of subimages in the feature space. The general idea of using tolerance spaces and near set theory started recently by James F. Peters [31] inspired by the initial ideas of Zeeman [54] on visual perception and Sossinsky [43] on tolerance spaces. A tolerance near set approach to image similarity is based on the idea that visual perception of image similarity forms by tolerance classes of *almost similar* elements in the images. This is the key idea in many recent publications image correspondence problem [8, 33, 18, 40]. The contribution of this thesis is mainly on defining a new measure of nearness between images based on the statistical analysis of the cardinality of tolerance classes. The proposed similarity measure is fully described in section 5.2 and can be described as an alternative to another recently proposed measure by Christopher Henry in [6]. (Section 5.1)

1.1 Applications of image similarity

Finding similarity between images is of a great importance in many applications. Some of the possible applications include the following:

- Management and search through digital archives of images and videos in personal, commercial and public domain image archives. [39, 42]
- Medical applications: analysis, archive and searching medical images. [19]
- Searching for images on Internet, query by image. [39, 42, 4]
- Application in "image registration" problem where similarity between images used to determine similarity between an image and its transformation. [15, 45]

- Image quality assessment: where the goal is to assess the similarity (or differences) between a reference image and a distorted image. [49, 50].
- Classification and retrieval of images based on content based similarity between pair of images. [3, 42]
- Removal of duplicate images from image databases by finding similarity between the image and its duplicate. [11]
- Investigating copyright violations by content based search and finding the source of a given image. [11, 12]

1.2 Contributions

The main contributions of this thesis are:

- Proposing a new tolerance space-based image similarity (nearness) measure that considers each image in its own tolerance space. (TCD nearness measure)
- Implementation of the existing tolerance based methods and the proposed methods using C++.
- Developing a stand alone application with graphical user interface (GUI). This application can be used by users to select and compare digital images and calculate nearness measures (similarities).
- Investigating the performance of the proposed method by performing a set of experiments on measuring similarity between an original image and a distorted image considering three different types of distortion including lightening/darkening, noise and twist.

2 Perceptual systems

A perceptual system is a set of perceptual objects and probe functions where probe functions are used to extract and represent features of the objects [23, 35, 22]. A perceptual object ($x \in O$) is something that can be presented to the senses and recognized by the human mind [20, 23, 22]. For example, a group of pixels in an image can be described as a perceptual object that can be seen and perceived. A probe function can be described as a model of a sensor that senses some visual features of an object such as color, entropy, texture, edges, spatial orientation, etc. Therefore, an image can be considered as a set of perceptual objects in a perceptual system approach to image analysis [40, 23, 18]

Definition 1 Probe function [23, 24]
A *probe function* $\phi(x)$ is a real-valued function representing features of the physical object x. A set of prob functions $\mathbb{F} = \{\phi_1(x), \phi_2(x), ..., \phi_l(x)\}$ can be defined to generate all the features for each object x where $\phi_i : O \to \mathbb{R}$. If only some of the probe functions are used to describe the perceptual objects, then we use $\mathcal{B} \subseteq \mathbb{F}$ to represent the probe functions in use.

Figure 1: A perceptual object is a subimage that can be described with a probe function such as the average gray level

Definition 2 Perceptual System [23]
A *perceptual system* $\langle O, \mathbb{F} \rangle$ is a real valued deterministic information system where O is a non-empty set of *perceptual objects*, while $\mathbb{F}$ is a countable set of *probe functions*.

Example 1 Images and perceptual subimages
An example of a perceptual object can be a pixel or a group of pixels or a region inside an

image . For example, an image can be divided into different subimages and each of these subiamges can be considered as a perceptual object. Figure 2 shows an example image divided into subimages. This image has been divided into square subimages (windows) with the same size. The group of pixels belonging to each subimage creates a perceptual object that can be visually perceived.

(a) Original image (Walnut)

(b) Each subimage is a perceptual object

(c) Gray scale image

(d) Avg gray scale used as a feature value

Figure 2: Images and perceptual objects

The set of probe functions or different features for the perceptual objects represented in an image can include average gray level, color, entropy, edge or texture. For instance, in the example of figure 2, we would like to use only average gray level of the pixels in each subimage (window) as a probe function. Hence, sets $\mathbb{F}$ and $\mathbb{B}$ will be,

$$\begin{aligned}\mathbb{F} &= \{\phi_1(x), \phi_2(x), ..., \phi_l(x)\}\\ &= \{\text{ avg gray level, color, entropy, edge, texture, spatial orientation }\}.\end{aligned}$$

$$\mathbb{B} = \{\phi_1(x)\} = \{\text{ avg gray level of the pixels in each subimage (window)}\}.$$

Figure 2d shows feature values of the perceptual objects in Figure 2b. Every subimage has been replaced with the average gray scale value of all the pixels in the subimage (perceptual object).

This approach to representation and comparison of feature values by probe functions started with the introduction of near sets [25, 28]. Probe functions provide a basis for describing and discovering similarities between sample objects in the context of a perceptual information system. This approach is a generalization of the concept of attributes in the approximation spaces exists in rough set theory [32, 17].

2.1 Probe functions (Visual Features)

Examples of probe functions are displayed in table 1 where each probe function extracts some features of the subimage. The choice of probe function is subjective and depends on the application.

Table 1: Examples of probe functions in use: $\mathbb{B} = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$

Probe function	Name	Equation	Comments and descriptions
$\phi_1(x)$	Average gray level	$\phi_1(x) = \frac{1}{N(x)} \sum_{\mu_i \in x} gr(\mu_i)$	$N(x)$ is the number of pixels in subimage x, μ_i is the i^{th} pixel in subimage x and $gr(\mu_i)$ is the grey level intensity of the pixel μ_i normalized between 0 and 1.
$\phi_2(x)$	Entropy of subimage	$\phi_2(x) = E = -\sum_{i=0}^{i=L} p_i \log p_i$	L is the number of grey levels in image ($L = 255$) and p_i is the probability of having i^{th} level of gray scale for a pixel.
$\phi_3(x)$	Average Red color component	$\phi_3(x) = \frac{1}{N(x)} \sum_{\mu_i \in x} R(\mu_i)$	Average of the Red component (R) of all pixels in the subimage in RGB color space
$\phi_4(x)$	Average Green color component	$\phi_4(x) = \frac{1}{N(x)} \sum_{\mu_i \in x} G(\mu_i)$	Average of the Green component (G) of all pixels in the subimage in RGB color space
$\phi_5(x)$	Average Blue color component	$\phi_5(x) = \frac{1}{N(x)} \sum_{\mu_i \in x} B(\mu_i)$	Average of the Blue component (B) of all pixels in the subimage in RGB color space

2.1.1 Gray level

Digital images are represented with a set of pixels in a two dimensional array and the associated intensity level of each pixel. In gray scale images, there is only one information associated with each pixel and that is the intensity or brightness of the pixel represented with a number in a given range. In a digital image this number is an n-bit binary number and therefore intensities can be represented with 2^n different levels ranging from 0 to $L = 2^n - 1$ (e.g $L = 255$ for an 8 bit digital image). We normalize the values and represent them with a number between 0 and 1 by dividing all the values by L. Let μ_i be a pixel (i^{th} pixel in an image). We use $gr(\mu_i)$, to represent the intensity or gray value of this pixel as it is used in the first two probe functions in table 1. Figure 3a shows an example of a gray scale image.

2.1.2 Entropy

The entropy of a subimage (or an entire image) is defined using the Shannon entropy formula given in equation 1 . Entropy is a statistical measure of randomness.

$$E = -\sum_{i=0}^{i=255} p_i \log p_i \tag{1}$$

where p_i is the probability of the occurrence of gray level i in the subimages. p_i can be defined as $pi = h_i/N$ where h_i is the number of pixels that take the i^{th} gray level and N is total number of pixel in the subimage. h_i can be obtained using the histogram of the pixel intensities. Lets assume we have 256 bins for the histogram corresponding to 256 gray levels ranging from 0 to 255. The value of histogram at i^{th} bin, represents h_i (the number of pixels in the subimage that have i^{th} gray level). A value of 0 is produced where all the pixels have the same intensities and the maximum value of entropy happens when all the intensities have equal probability of occurrence.

2.1.3 Color

Unlike a grayscale image, in a color image, there are 3 numbers assigned to each pixel that show the intensity value of each separate color component. For example, in an RGB color space (which is the standard in computer displays), each pixel has 3 intensity values representing Red, Green and Blue color components. A color image can be viewed as the combination of 3 separate image where the intensity of pixels in each image represents the corresponding color. Figure 3b shows the color image and figures 3c to 3e show the intensity of each separate color channel and hence represent redness, greenness and blueness of each pixel. [1] In table 1, we use $R(\mu_i)$, $G(\mu_i)$ and $B(\mu_i)$ to represent intensities of color channels for the pixel μ_i.

[1]Images are extracted from http://en.wikipedia.org/wiki/File:RGB_channels_separation.png. Images are in *public domain*, no licensing required

(a) grayscale (b) colored (c) R channel (d) G channel (e) B channel

Figure 3: A sample image, (a) gray scale, (b) color image, (c,d,e) three images representing intensities of R, G and B channels, respectively

3 Perceptual Indiscernibility and Tolerance Relations

Tolerance relations can be viewed as a natural generalization of the indiscernibility relations. Sossinsky in his original paper in 1986 writes:
"The exact idea of closeness or of 'resembling', or of 'being within tolerance' is universal enough to appear quite naturally in almost any mathematical setting. It is especially natural in mathematical applications: practical problems, more often than not, deal with approximate input data and only require viable results - results with a tolerable level of error." [43]

In this section tolerance relations are defined and compared with indiscernibility relation. Indiscernibility relation introduced by Z. Pawlak [21] is the key idea in approximation spaces in rough set theory. Indiscernibility and tolerance relations are important and useful in defining measures to compare affinities between pairs of perceptual objects in a perceptual system, for example to compare perceptual images [34]. The term *tolerance space* was introduced by E.C. Zeeman in 1961 in modeling visual perception with tolerances [53]. A tolerance space is a set X supplied with a binary relation $\simeq$ (*i.e.*, a subset $\simeq \subset X \times X$) that is reflexive (for all $x \in X$, $x \simeq x$) and symmetric (for all $x, y \in X$, $x \simeq y$ and $y \simeq x$) but transitivity of $\simeq$ is not required. We use indiscernibility and tolerance relation in the context of perceptual systems.

Definition 3 Perceptual Indiscernibility Relation
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. For every $\mathcal{B} \subseteq \mathbb{F}$ the indiscernibility relation $\sim_{\mathcal{B}}$ is defined as follows:

$$\sim_{\mathcal{B}} = \{(x, y) \in O \times O \mid \ \forall \phi \in \mathcal{B}\,, \quad \| \phi(x) - \phi(y) \| = 0\} \tag{2}$$

If $\mathcal{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\sim_{\{\phi\}}$ we write $\sim_\phi$.

Definition 4 Perceptual Weak Indiscernibility Relation
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. For every $\mathcal{B} \subseteq \mathbb{F}$ the weak indiscernibility relation $\simeq_{\mathcal{B}}$ is defined as follows:

$$\simeq_{\mathcal{B}} = \Big\{ (x, y) \in O \times O \mid \ \exists \phi_i \in \mathcal{B} \ , \ \| \underline{\triangle \phi} \| = 0 \Big\} , \tag{3}$$

If $\mathcal{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\simeq_{\{\phi\}}$ we write $\simeq_{\phi}$.

The set of all perceptual objects in O that are indiscernible to an object $x \in O$ is called an *equivalence class* and is shown as $x_{/\sim_{\mathcal{B}}}$. Note that all the elements in an equivalence class are indiscernible to each other.

Definition 5 Tolerance Relation
A tolerance relation $\mathcal{R} \subseteq X \times X$ on a set X in general, is a binary relation that is reflexive and symmetric but not necessarily transitive [43].

1. $\mathcal{R} \ \subset X \times X$
2. $\forall x \in X, \quad (x, x) \in \mathcal{R}$
3. $\forall x, y \in X, \quad (x, y) \in \mathcal{R} \ \Rightarrow (y, x) \in \mathcal{R}$

The basic idea in a tolerance view of image similarity is to replace the indiscernibility relation in rough sets with a tolerance relation in near sets.

Definition 6 Perceptual Tolerance Relation
Let $\Big\langle O, \mathbb{F} \Big\rangle$ be a perceptual system and let $\epsilon \in \mathbb{R}$ (real numbers). For every $\mathcal{B} \subseteq \mathbb{F}$ the perceptual tolerance relation $\cong_{\mathcal{B}}$ is defined as follows:

$$\cong_{\mathcal{B},\epsilon} = \{(x, y) \in O \times O : \| \ \phi(x) - \phi(y) \ \| \leq \epsilon\}. \tag{4}$$

where $\phi(x) = [\phi_1(x) \ \phi_2(x) \ ... \ \phi_l(x)]^T$ is a feature vector obtained using all the probe functions in $\mathcal{B}$ and $\| \ . \ \|$ is L_2 norm (L_p norm in general). If $\mathcal{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\cong_{\{\phi\}}$ we write $\cong_{\phi}$. Further, for notational convenience, we will write $\cong_{\mathcal{B}}$ instead of $\cong_{\mathcal{B},\epsilon}$ with the understanding that there is a parameter ϵ in definition of the tolerance relation.

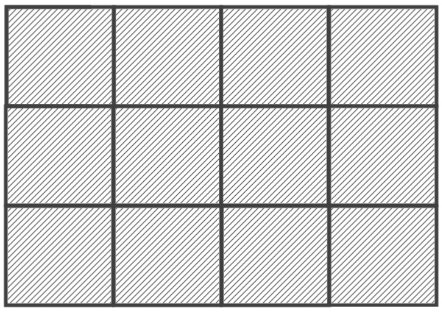
Similar to definition of equivalence classes using indiscernibility relation, a *tolerance class* can be defined as

$$x_{/\cong_{\mathcal{B}}} = \{y \in O \mid y \cong_{\mathcal{B}} x\}. \tag{5}$$

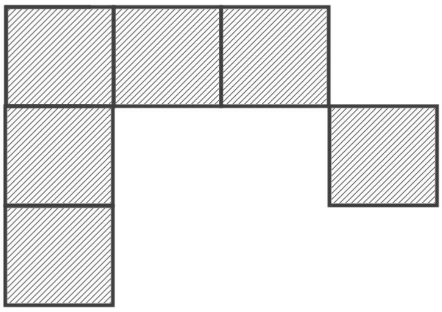
This is a definition of tolerance class that is introduced in [1]. It is important to note that tolerance classes do not partition the set O but rather cover the set (*i.e.* an object can belong to more than one class). The set of all tolerance classes is called a *covering* of O and is defined as:

$$O_{/\cong_{\mathcal{B}}} = \{x_{/\cong_{\mathcal{B}}} \mid x \in O\}. \tag{6}$$

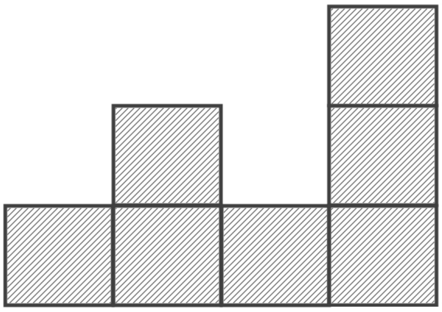
A covering is a family of subsets of O whose union is O and their intersection is not necessarily empty. Figure 4 for instance shows an illustrated example of a set of subimages (windows). Sample tolerance classes are shown in Figures 4b to 4e. Intersection of these tolerance classes is not necessarily empty and they cover the original image (union of all tolerance classes is equal to the original image). Figures 5 and 6 show examples of a real image and its covering with different window sizes. The actual classes are not shown independently. Each subimage is again replaced with its average gray scale value.



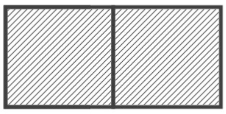
(a) A covering



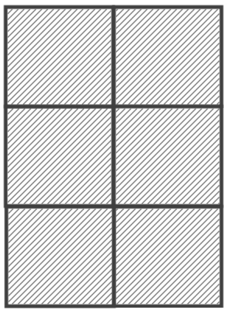
(b) 1st class



(c) 2nd class

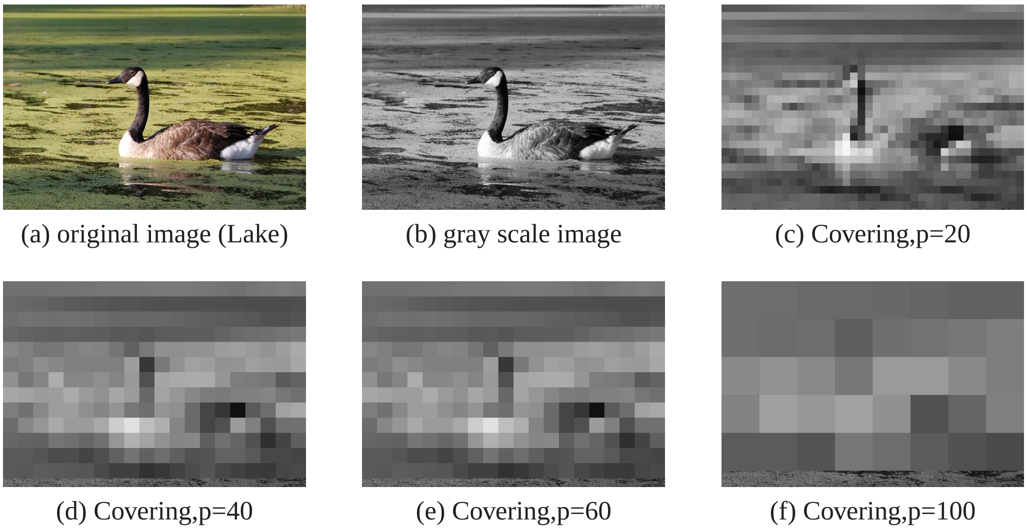


(d) 3rd class



(e) 4th class

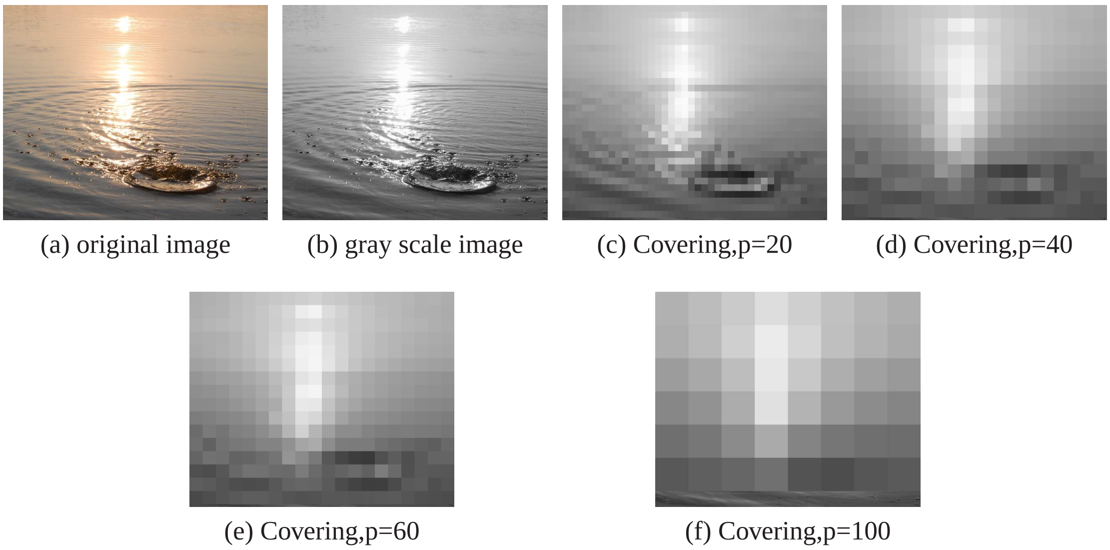
Figure 4: (a) A set of subimages in an unknown image, (b)-(e) examples of tolerance classes (subsets of the original set in (a)) that have non empty intersection and cover the original set



(a) original image (Lake) (b) gray scale image (c) Covering,p=20

(d) Covering,p=40 (e) Covering,p=60 (f) Covering,p=100

Figure 5: Coverings with different subimage sizes (p by p pixels)



(a) original image (b) gray scale image (c) Covering,p=20 (d) Covering,p=40

(e) Covering,p=60 (f) Covering,p=100

Figure 6: Coverings with different subimage sizes (p by p pixels)

4 Near sets

It has been shown [34, 23, 35, 29, 7] that near sets which are a generalization of Rough sets [36] provide a good basis for classification of perceptual objects. Sets of perceptual objects where two or more of the objects have matching descriptions are called near sets [27]. The basic idea in the near set approach to object recognition is to compare object descriptions. Sample perceptual objects $x, y \in O, x \neq y$ are near each other if, and only if x and y have similar descriptions. Figure 7 is an illustration [2] of two near sets X and Y where elements with similar descriptions have been color coded (equivalence classes). Sets X and Y both contain elements that belong to the same equivalence class, and hence, are near each other.

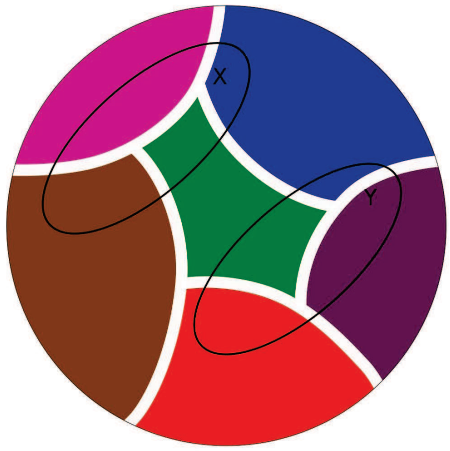


Figure 7: Illustration of near sets X,Y containing elements with the same description.

Definition 7 Weak Nearness Relation[22, 23]
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 $\langle O, \mathbb{F} \rangle$ $(X \underline{\bowtie}_{\mathbb{F}} Y)$ iff there exist $x \in X$ and $y \in Y$ and there is $\mathcal{B} \subseteq \mathbb{F}$ such that $x \sim_{\mathcal{B}} y$, *i.e.*, set X *is weakly near to* a set Y.

Definition 8 Nearness Relation[22, 23]
Let $\langle O, \mathbb{F} \rangle$ be 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 $F_1, F_2 \subseteq \mathbb{F}$ and $f \in \mathbb{F}$ and there are $A \in O_{/\sim_{F_1}}, B \in O_{/\sim_{F_2}}, C \in O_{/\sim_f}$ such that $A, B \subseteq C$.

[2] Image 7 copied from `http://en.wikipedia.org/wiki/Near_sets`

Figure 8 shows the nearness relation in more detail [3] according to definition 8. In (a), two sets X, Y are considered in the space O where O is partitioned into equivalence classes of indiscernible objects using a set of probe functions given in $\mathcal{B} \subset \mathbb{F}$. It can be seen than sets X and Y may have elements that belong to different classes of objects. However, in order for the two sets X and Y to be *near* to each other, it is necessary to find any two subsets of $\mathbb{F}$ namely F_1 and F_2 such that when we partition the set O with F_1 and F_2, there are at least two resulting classes $A \in O_{/\sim_{F_1}}$ and $B \in O_{/\sim_{F_2}}$ that are subsets of X and Y, respectively. This is shown in part (b) and (c) of Figure 8. Moreover, there should exist a probe function $f \in \mathbb{F}$ such that A and B both belong to one of the classes that is created by partitioning O using the probe function f. Part (d) of the figure shows how O is partitioned into equivalent classes where one of them includes both A and B ($A, B \subseteq C$).

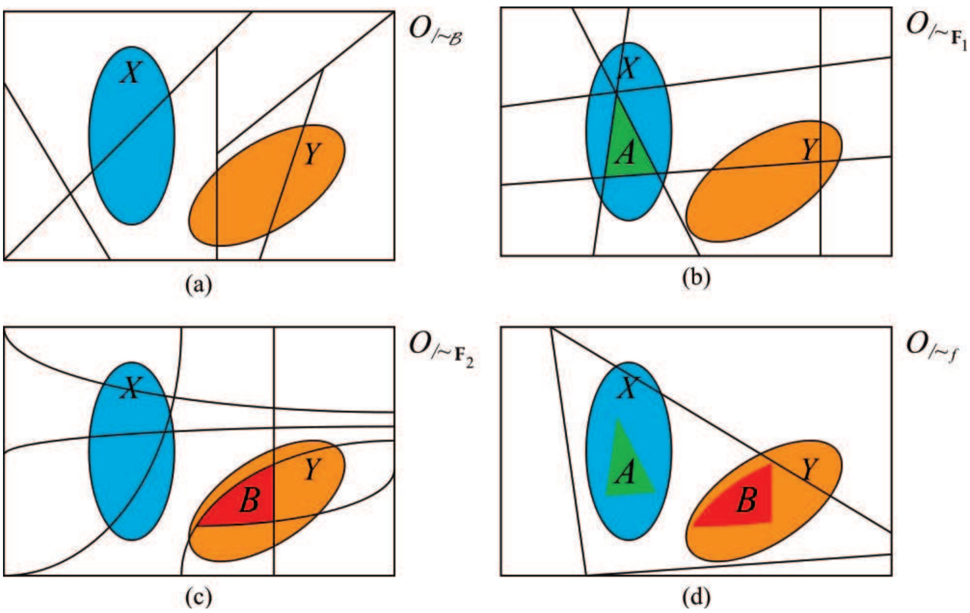


Figure 8: Illustration of near sets X,Y based on definition 8.

Definition 9 Tolerance Nearness[22, 23]
Let $\langle O, \mathbb{F} \rangle$ be a perceptual system and let $X, Y \subseteq O, \epsilon \in \mathbb{R}$. The set X is perceptually near to the set Y within the perceptual system $\langle O, \mathbb{F} \rangle$ ($X \underline{\bowtie}_{\mathbb{F}} Y$) iff there exist $x \in X$, $y \in Y$ and $\epsilon \in \mathbb{R}$ such that $x \cong_{\mathbb{F}} y$.

[3] Image 8 copied from `http://en.wikipedia.org/wiki/Near_sets`

Table 2: Relation Symbols

Symbol	Interpretation
O	Set of perceptual objects,
X	$X \subseteq O$, set of sample objects,
$\mathbb{F}$	A set of probe functions,
$\mathcal{B}$	$\mathcal{B} \subseteq \mathbb{F}$, set of probe functions in use,
$\Re$	reals,
ϕ	$\phi \in \mathcal{B}$, where $\phi_i : O \longrightarrow \Re$, probe function,
$\sim_{\mathcal{B}}$	$\{(x, y) \mid \ f(x) = f(y) \ \forall \phi \in \mathcal{B}\}$, Indiscernibility relation,
$\simeq_{\mathcal{B}}$	$\{(x, y) \in O \times O \mid \ \exists \phi_i \in \mathcal{B} \phi_i(x) = \phi_i(y)\}$, weak indiscernibility,
$\cong_{\mathcal{B},\varepsilon}$	Perceptual tolerance relation,
$x_{/\sim_{\mathcal{B}}}$	$x_{/\sim_{\mathcal{B}}} = \{y \in X \mid \ y \sim_{\mathcal{B}} x\}$ (equivalence class),
$O_{/\sim_{\mathcal{B}}}$	$O_{/\sim_{\mathcal{B}}} = \{x_{/\sim_{\mathcal{B}}} \mid \ x \in O\}$, quotient set,
$\bowtie_{\mathcal{B}}$	nearness relation symbol,
$\underline{\bowtie}_{\mathcal{B}}$	weak nearness relation symbol,
$\underline{\underline{\bowtie}}_{\mathcal{B},\varepsilon}$	Tolerance nearness relation symbol.

Definition 10 Tolerance Near Sets
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $X \subseteq O$. A set X is a tolerance near set iff there is $Y \subseteq O$ such that $X \underline{\underline{\bowtie}}_{\mathbb{F}} Y$. The family of near sets of a perceptual system $\langle O, \mathbb{F}\rangle$ is denoted by $\mathit{Near}_{\mathbb{F}}(O)$.

5 Nearness Measures for Image Analysis and Comparison

Near set theory is a theory about nearness between sets. It provides a framework for comparing sets of perceptual objects based on their descriptions. For example, tolerance near sets [33], suggest how we can use tolerance relations to define nearness between objects in near sets. However, defining an actual numerical nearness measure between two sets depends on the application and is not directly addressed in near set theory. In this section, nearness measures between images are developed and presented in a near set framework using tolerance relations in a tolerance space. The term tolerance space was introduced by E.C. Zeeman in 1961 in modeling visual perception with tolerances. Tolerance relations are viewed as good models of how one perceives, how one sees [53]. Tolerance relations are also considered as a basis for studying similarities between objects [30]. The basic idea behind using tolerance classes is to relax the equivalence relations needed in mathematical world and extend it to tolerance relations where *almost* solutions are possible.
In this section, a new similarity measure based on tolerance classes and near sets, for comparison and analysis of images will be introduced. The new measure called tolerance cardinality distribution nearness measure (TCD) is based on statistical distribution of the size (cardinality) of the tolerance classes in each image. The idea behind it is that if images are similar to each other, they should have corresponding tolerance classes with *almost* the same size and hence similar size distribution functions. TCD will be compared to another tolerance based measure named tNM and a Hausdorff distance based measure named HSDF.

5.1 Tolerance Nearness Measure (tNM)

Tolerance nearness measure (tNM), introduced in [6] is based on the idea that if two images are similar, tolerance classes in the union set of those images will have similar number of subimages from each image. tNM between two images [6, 8] is defined as follows:

Suppose X and Y are sets of perceptual objects (for example, two images), and $Z = X \cup Y$ is the union of X and Y. Let $[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X}$ and $[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y}$ denote the portion of the tolerance class $z_{/\cong_{\mathcal{B},\varepsilon}}$ that belong to X and Y respectively. Then,

$$[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X} \triangleq \{x \in z_{/\cong_{\mathcal{B},\varepsilon}} \mid x \in X\} \subseteq z_{/\cong_{\mathcal{B},\varepsilon}} \tag{7}$$

$$[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y} \triangleq \{y \in z_{/\cong_{\mathcal{B},\varepsilon}} \mid y \in Y\} \subseteq z_{/\cong_{\mathcal{B},\varepsilon}} \tag{8}$$

$$z_{/\cong_{\mathcal{B},\varepsilon}} = [z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X} \cup [z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y} \tag{9}$$

Tolerance nearness measure (tNM) is defined as the weighted average of the closeness between the cardinality (size) of set $[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X}$ and the cardinality of $[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y}$ where the cardinality of $z_{/\cong_{\mathcal{B},\varepsilon}}$ is used as the weighting factor.

$$tNM = \frac{1}{\sum\limits_{z_{/\cong_{\mathcal{B},\varepsilon}}} |z_{/\cong_{\mathcal{B},\varepsilon}}|} \times \sum_{z_{/\cong_{\mathcal{B},\varepsilon}}} \frac{\min(\ |[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X}|\ ,\ |[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y}|\ )}{\max(\ |[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq X}|\ ,\ |[z_{/\cong_{\mathcal{B},\varepsilon}}]_{\subseteq Y}|\ )} \times |z_{/\cong_{\mathcal{B},\varepsilon}}| \tag{10}$$

5.2 Tolerance Cardinality Distribution Nearness Measure (TCD)

5.2.1 Motivation

tNM nearness measure is the first quantitative measure of image similarity in the context of tolerance near sets. While tNM has been shown to be a powerful measure for image similarity, there is a basic problem in this method which was the motivation for defining another measure (TCD) which is explained in the next section. As can be seen from the above equations, tolerance classes in calculation of tNM are formed in the space of all perceptual objects in the union of both images. This creates a problem, if all of the pixels (or perceptual objects) of an image are shifted in the feature space (for example lightening or darkening of an image). In this case, the new shifted pixels are no longer in the same tolerance class with the corresponding pixels in the original image, and hence the value will be significantly changed. This problem is shown as an example in Figure 9. While both of images are expected to be similar, the tNM similarity measure between two images is 0.27 ($\varepsilon = 0.2$, subimage size = 15 pixels and two probe functions in use are average gray level and entropy).

The new proposed measure (TCD) in this thesis, however, is based on obtaining tolerance classes in the tolerance space of each image individually. The correspondence between perceptual objects is then defined by comparing the size of tolerance classes in each tolerance space. This will be done by comparing the histogram of the sizes (cardinality) of tolerance classes in each space. TCD similarity measure between images in Figure 9 (as will be discussed in the following section) will be 0.8, which is intuitively more accurate.

Figure 9: An image under two different lightening conditions, tNM = 0.27, TCD = 0.8

5.2.2 TCD similarity measure

A similarity measure is proposed here based on statistical distribution of the size (cardinality) of the tolerance classes in each image. The size of each tolerance class is defined as the number of perceptual objects (subimages) in that tolerance class. Definition of TCD is based on the basic idea that if images are similar to each other, they should have corresponding tolerance classes with "almost" the same size and hence similar size distribution functions.

Let $X, Y \in O$ represent two images (sets of perceptual objects) where they can be covered by a tolerance relation . Sets of all tolerance classes in X and Y are:

$$X_{/\cong_{\mathcal{B},\epsilon}} = \{x_{/\cong_{\mathcal{B},\epsilon}} \mid \ x \in X\} \tag{11}$$

$$Y_{/\cong_{\mathcal{B},\epsilon}} = \{y_{/\cong_{\mathcal{B},\epsilon}} \mid \ y \in Y\} \tag{12}$$

Let $c(x_{/\cong_{\mathcal{B},\epsilon}})$ and $c(y_{/\cong_{\mathcal{B},\epsilon}})$ represent normalized cardinality of the tolerance classes $x_{/\cong_{\mathcal{B},\epsilon}}$ and $y_{/\cong_{\mathcal{B},\epsilon}}$, respectively.

$$c(x_{/\cong_{\mathcal{B},\epsilon}}) = \frac{\left|x_{/\cong_{\mathcal{B},\epsilon}}\right|}{|X|} \tag{13}$$

$$c(y_{/\cong_{\mathcal{B},\epsilon}}) = \frac{\left|y_{/\cong_{\mathcal{B},\epsilon}}\right|}{|Y|} \tag{14}$$

Suppose $\{b_1, b_2, ..., b_{N_b}\}$ is a set of discrete bins for calculation of the histograms of $c(x_{/\cong_{\mathcal{B},\epsilon}})$ and $c(y_{/\cong_{\mathcal{B},\epsilon}})$ where $x \in X$ and $y \in Y$. The histogram (or empirical distribution function) of $c(x_{/\cong_{\mathcal{B},\epsilon}})$ at bin value b_j is shown as $H_{c_X}(b_j)$ and is defined as the number of tolerance classes with the number of subimages (size) that belongs to j^{th} bin. The cumulative distribution function is then defined as follows:

$$CH_{c_X}(b_j) = \sum_{i=1}^{i=j} H_{c_X}(b_i) \tag{15}$$

$CH_{c_Y}(b_j)$ is similarly defined for image Y. The *Tolerance Cardinality Distribution (TCD)* nearness measure is defined by taking the sum of differences between cumulative histograms as defined in equation 16.

$$TCD = 1 - \left(\sum_{j=1}^{j=N_b} |CH_{c_X}(b_j) - CH_{c_Y}(b_j)| \right) \tag{16}$$

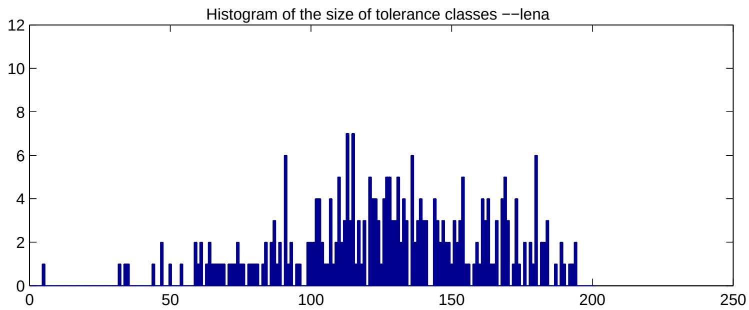
Example 2 Distribution of the size of tolerance Classes

Consider pair of images (A and B) in Figure 10 and pair of images (A and C) in Figure 12. The images are divided into subimages of 20 by 20 pixels and two probe functions have been used to measure the "average grayscale" and "entropy" of each subimage. The tolerance classes have been calculated (using $\varepsilon = .2$). Size of each tolerance class is the number of subimages in the tolerance class. Figure 10c, 10d and 12d show the histograms (empirical distributions) of the size of tolerance classes (not normalized) in images A,B and C, respectively. Moreover, normalized cumulative histograms for images A,B and C are plotted in figures 11c, 11d and 13d, respectively.

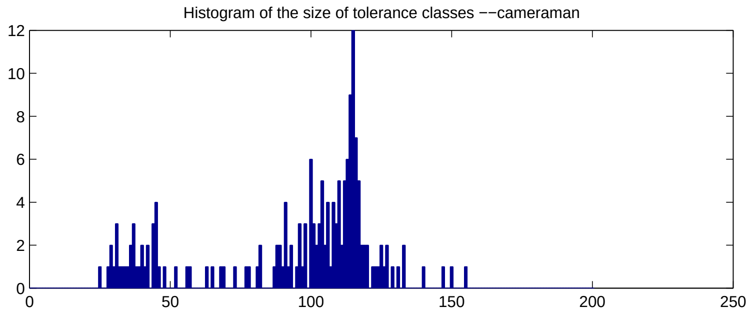
These histograms represent how the tolerance classes are distributed in terms of their size. In order to compare distributions (histograms) between two images, cumulative histograms have been used instead of plain histograms because it has been claimed that this method is less sensitive to the choice of bins for histograms [42]. Dissimilarity between images is calculated by subtracting the cumulative histograms of the corresponding images. Figures 11e and 13e show the absolute value of difference between cumulative histograms calculated at each bin for pairs of images in A,B and pairs of images in A,C respectively. The area under this plots represents the difference between these histograms. Comparing images 11e and 13e reveals that difference between images A and B is more significant than between images A and C (*i.e.* images A and C are more similar to each other). In an extreme case, for two identical images, histograms (and hence cumulative histograms) are the same and the difference would be zero.

(a) Image A

(b) Image B



(c) Histogram of size of classes in Image A

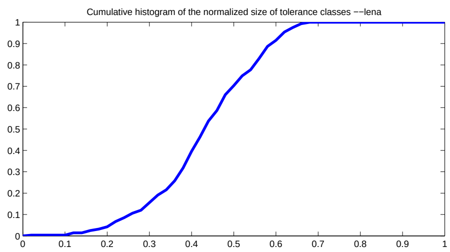


(d) Histogram of size of classes in Image B

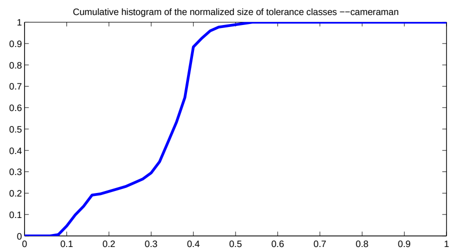
Figure 10: (a),(b): Pair of images (A,B) and (c),(d): histogram of the size of tolerance classes

(a) Image A

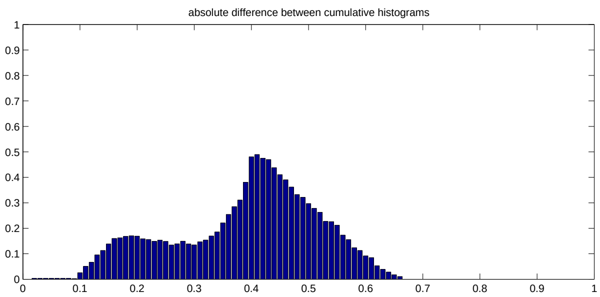
(b) Image B



(c) Cumulative histograms of the size of classes in Image A



(d) Cumulative histograms of the size of classes in Image B

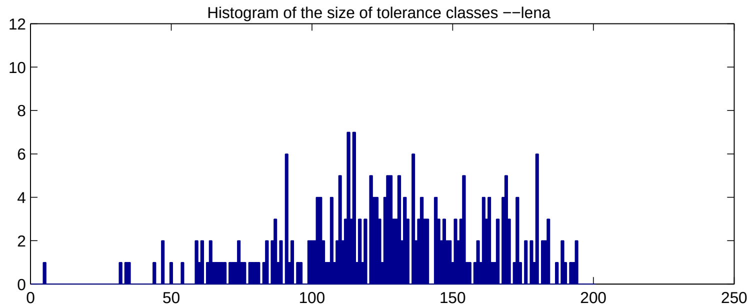


(e) Absolute difference between cumulative histograms

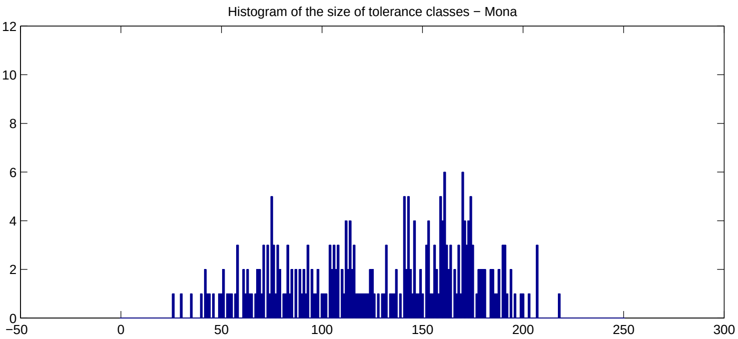
Figure 11: (a),(b): Pair of images (A,B), (c),(d): cumulative histograms of the size of tolerance classes, (e): difference between (c) and (d)

(a) Image A

(b) Image C



(c) Histogram of size of classes in Image A

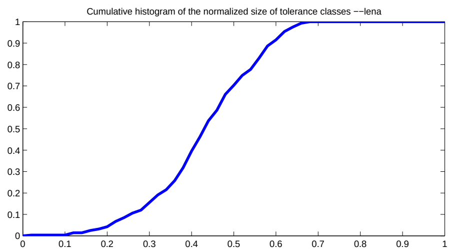


(d) Histogram of size of classes in Image C

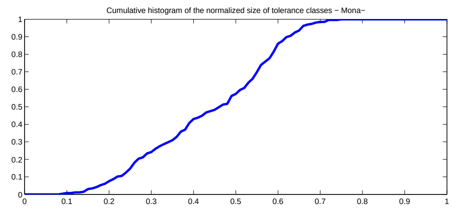
Figure 12: (a),(b): Pair of images (A,C) and (c),(d): histogram of the size of tolerance classes

(a) Image A

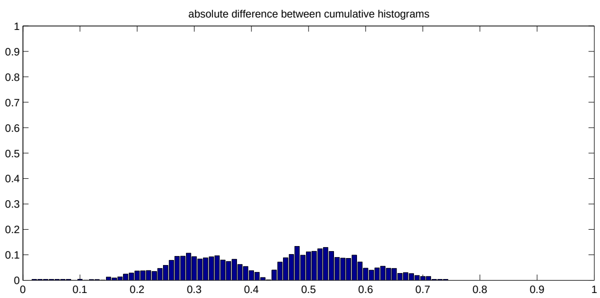
(b) Image C



(c) Cumulative histograms of the size of classes in Image A



(d) Cumulative histograms of the size of classes in Image C

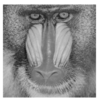
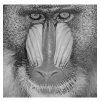


(e) Absolute difference between cumulative histograms

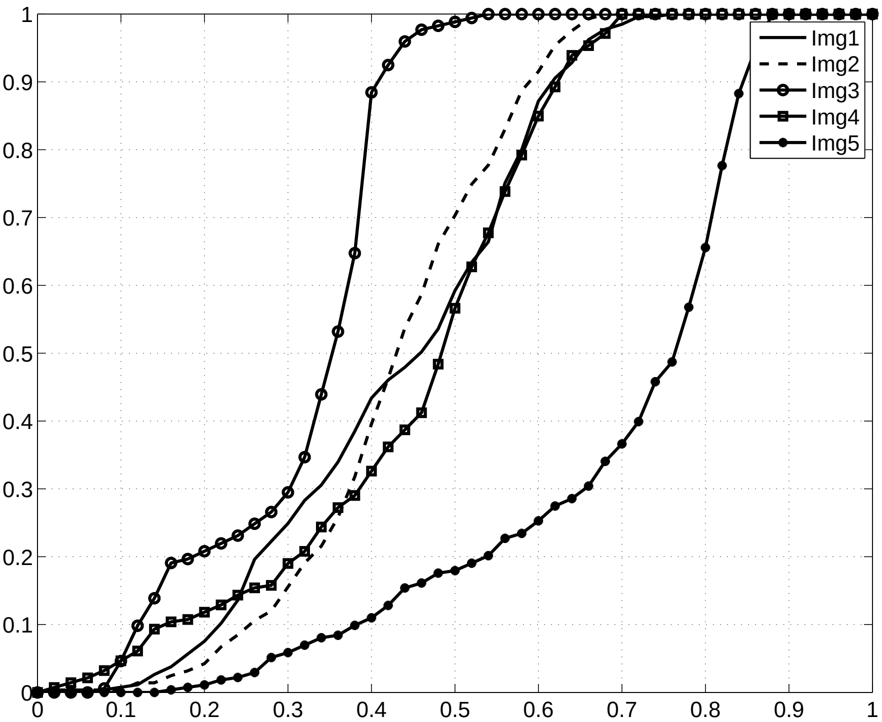
Figure 13: (a),(b): Pair of images (A,C), (c),(d): cumulative histograms of the size of tolerance classes, (e): difference between (c) and (d)

Example 3 Comparing images by histograms of sizes of tolerance classes *In this example, cumulative histograms of the size of tolerance classes are plotted in Figure 14 for 5 different images. It can be seen that while Image 1, Image 2 and Image 4 have very similar distributions (cumulative histograms), Image 3 and Image 5 have completely different histograms, simple subtraction of the plots shows that the maximum distance (dissimilarity) between images of this set is between Image 3 and Image 5. Numerical values of the proposed TCD measure are shown in Table 3 and confirms that the lowest similarity is between Image 3 and Image 5 (cameraman and baboon).*

Table 3: Numerical values of TCD between images in example 3

TCD					
	1	0.66	0.91	0.93	0.86
	0.66	1	0.75	0.59	0.80
	0.91	0.75	1	0.84	0.95
	0.93	0.59	0.84	1	0.79
	0.86	0.80	0.95	0.79	1

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5



(f) Cumulative Histograms

Figure 14: An example of five different images and their corresponding cumulative histograms

5.3 Feature space based Hausdorff nearness measure: HSDF

Hausdorff distance is a known classical distance defined between a pair of sets. It is usually used in image processing for template matching where the spatial distance between sets of points (in image domain) is of interest [9]. Suppose X and Y are two sets of points in a metric space. Assuming that $d(x, y)$ is any distance defined between points x and y in the space, Hausdorff distance $\rho_H(X, Y)$ between sets X and Y is defined as:

$$\rho_H(X, Y) = \max\{d_H(X, Y), d_H(Y, X)\}, \tag{17}$$

where

$$d_H(X, Y) = \max_{x \in X}\{\min_{y \in Y}\{d(x, y)\}\}, \tag{18}$$

$$d_H(Y, X) = \max_{y \in Y}\{\min_{x \in X}\{d(x, y)\}\}. \tag{19}$$

$d_H(X, Y)$ and $d_H(Y, X)$ are directed Hausdorff distances from X to Y and from Y to X, respectively.

In this thesis, we use a standard form of Hausdorff distance in the feature space as used in [37] to define the distance between sets of feature vectors corresponding to subimages in two images. For example, two images in Figure 15a and 15b have been divided into subimages of size 15 by 15 and two features (average gray scale and entropy features) for each subimage has been calculated. Figure 15c shows all the subimages in a 2-D feature space (gray level and entropy) where the set of points with "blue cross" named as X represents subimages in Image 1 (a) and points with "red dots" named as Y represent the feature values for subimages of Image 2 (b). Therefore X and Y represent sets of points in feature space. The difference between these sets of points in the feature space represents the difference between the images in terms of the descriptions of the subimages. For example, it can be seen that Image 1 contains subimages (marked with blue cross sign '+') that have very low values of both gray level and entropy. These subimages correspond to the picture of cameraman body that is dark in color with very little variation and hence low entropy. For every subimage $x \in X$ and $y \in Y$, the L_1 norm distance (Manhattan distance) between subimages is defined as:

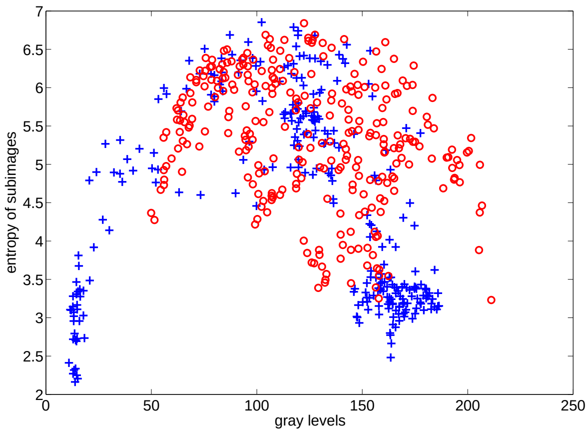
$$d(x, y) = \| \phi(x) - \phi(y) \|_1 = \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|. \tag{20}$$

Using equations 17 to 19, HSDF (Feature Space based Hausdorff Distance) similarity measure is then defined as follows to represent nearness.

$$HSDF = \frac{1}{1 + \rho_H(X, Y)}. \tag{21}$$

(a) Image 1

(b) Image 2



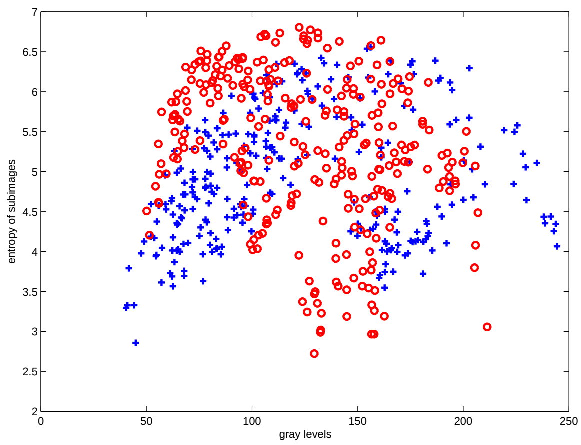
(c) Subimages in feature space

Figure 15: Two sample images (a) and (b) and the point sets of the corresponding subimages in feature space (a: blue cross) and (b: red circles)

Another example is shown in figure 16 with subimages in the feature space

(a) Image 1

(b) Image 2



(c) Subimages in feature space

Figure 16: Two sample images (a) and (b) and the point sets of the corresponding subimages in feature space (a: blue cross) and (b: red circles)

6 Perceptual Image Analysis System: Implementation of a GUI

The perceptual image analysis system discussed in the previous sections and developed as a part of this research, has been implemented using the latest version of "C++ .Net" to date (Visual Studio 2010 [4]) and "Image Magick" library of toolsets for C++ ([16, 44]). By combining both managed and native C++ coding techniques and applying advanced programming skills, it is now a powerful and user friendly tool. The object oriented structure of the program makes it easier to apply further changes and improvements to the program and extend it in the future. The choice of using C++ language over other languages like MATLAB, also gives it the power of speed and more flexibilities to add custom made changes.

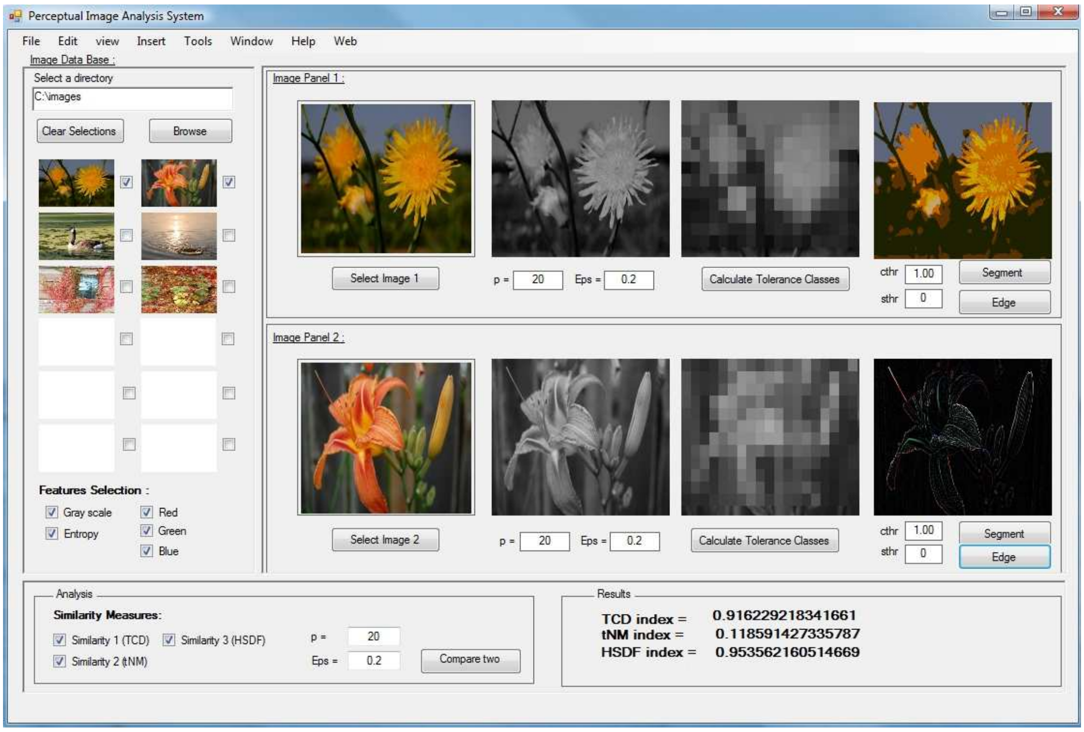


Figure 17: GUI of the "Perceptual Image Analysis" system

The PIA system has been designed to be both user friendly and accurate at the same time. It includes many parts and panels for different purposes and it can be used and

[4]http://www.microsoft.com/visualstudio/

expanded for various applications. Some of its features are explained in the the following paragraphs. Figure 17 shows a snapshot view of the *Perceptual Image Analysis* System (PIA).
The graphical user interface (GUI) of the system has 6 different sections as explained below:

- `Image panel 1` and `Image panel 2`:
  Each panel includes 4 windows that are used for displaying and analyzing the similarity between pairs of images. The first window contains the image that has been selected by the user. The gray scale version of each image is displayed in the second window. The third window displays the *covering* of the selected image. The last window in each panel is used to display the results of image segmentation and edge detection. User can select the parameters such as window size (subimage) and value of *epsilon* (ε)in the tolerance relation.

- `Image database panel`:
  This panel is used to display and manage the set of images used for analysis. Users can select a directory on their computer by clicking on `Browse` button and images in that directory will be selected. The images in that folder will be loaded into the small image boxes in the image database panel and the user will be able to select these images for comparisons. 12 different images will be loaded into the system at the same time and user can choose any pair of those images for later analysis and comparison.

- `Feature selection`:
  This part which is actually inside the image database panel, allows the user to choose from a set of available features (probe functions) to be used for image analysis and similarity calculation.

- `Analysis panel`:
  This panel allows user to choose a method from a list of available measures to calculate similarity between images.

- `Results panel`:
  Calculated similarity measures will be displayed in this part of the program.

A short description of the GUI functionalities are listed below:

- **Loading images**:
  Using the `select image` button located below the first image box, the user can browse in the computer and select the desired image in each panel.

- **Calculating the tolerance classes in the covering**:
  By clicking on the `Calculate Tolerance Classes` button, the *covering* of the image will be displayed in the third image box of the given image panel. Figure 19 shows examples of different images and their coverings created by the PIA software. In these examples, first the original images are loaded and average gray scale feature is chosen. Then, each image is converted to gray scales, tolerance classes are calculated, and the covering of each image is displayed. A covering is the set of all tolerance classes of an image.

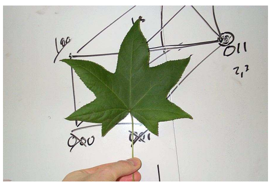
- **Calculating similarity measures**:
  Nearness (similarity) measures will be calculated after the user clicks on `compare two` button. The size of subimages (p) and the value of epsilon in tolerance relation can be also selected before the analysis. The results will be displayed in the results panel.

- **Image Segmentation**:
  By clicking on the `segment` button on each image panel, an image segmentation algorithm is applied to the images and the results will be displayed in the 4th window of the image panel. The segmentation algorithm divides the entire image into non-overlapping areas of the same color/gray level. The algorithm is based on the fuzzy c-means method in [14] and the threshold parameters of the method can also be determined in the program.

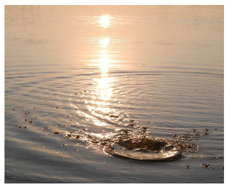
- **Edge detection**:
  By clicking on the `Edge` button in each image panel, edges of the image are detected and displayed in the last window. This functionality has been implemented using Image Magick edge detection routine.

Figure 18: Examples of segmentation and edge detection in the GUI

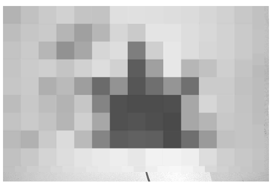
(a) original image (Leaf1)



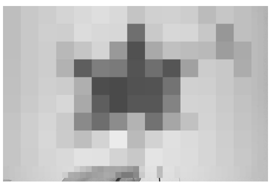
(b) original image (Leaf2)



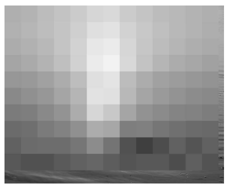
(c) original image (Water)



(d) covering,p=60



(e) covering, p=60



(f) covering, p=60

Figure 19: GUI examples, images and their coverings

7 Experiments on Image Similarity

In this section, the performance of the similarity measures on defining similarity between pairs of images is investigated. A similarity measure $S(X, Y)$ between two images X and Y is expected to have the following properties [2]:

- **Reflexivity**: The similarity measure between two identical images should be one (maximum value). It is clear that the similarity measure given in this thesis all have this property. $S(X, X) = 1$

- **Symmetry**: Intuitively, we require a similarity measure function to be symmetric and the order of images to be compared does not affect similarity. The similarity measures used in this thesis are all symmetrical because the order of images does not affect the similarity measure. This can be shown both theoretically and practically. $S(X, Y) = S(Y, X)$

- **Reaction to noise**: Intuitively, we expect the similarity measure between a noise free image and a noisy version of the same image to be high because they represent the same image. However, the similarity should be decreasing with adding the noise level such that in the presence of extremely high noise the similarity drops accordingly.

- **Reaction to lightening or darkening**: We expect difference in the lightening have very little effect on similarity. Lightening or darkening does not change the content of the image and hence the similarity between an image and its darkened/brightened version should be very high until the content is obscured by the very low/high level of light.

Moreover, in this thesis the effect of distortion of the image is also investigated through some examples.

- **Reaction to distortion**: It is expected that images before and after distortion are similar to each other unless the distortion is very high. The change in similarity also depends on the feature that are used for calculating nearness and also on the level of distortion.

7.1 Reaction to noise, lightening/darkening, and distortion

The effect of noise and lightening/darkening has been investigated here through the following experiments. In the first experiment, “salt and pepper” noise has been added to the image of the cameraman each time with a different intensity. The noise intensity determines the fraction of pixels that will be affected by adding noise. In the second example, the effect of darkening/lightening has been studied by comparing an image to the same image after increasing or decreasing the brightness of the image.

7.1.1 Experiment 1: Sensitivity of the similarity measures to noise

In this experiment, different levels of noise are added to an image and the similarity between the original image and the noisy images has been calculated. Figure 20 shows the original picture in the top left corner and 7 different images obtained by adding 5%, 10% ,15%, 20%, 50%, 75% and 90% salt and peper noise, respectively. The original image is compared with each of the noisy images and the results are shown in figure 20.

Discussion:
Images of this experiment are distorted versions of the same image after adding noise, therefore, as long as the similarity between the original noise free image and the noisy image is clearly visible to human eye, we expect our similarity measure to have high values reflecting the existing similarity. Figure 20 shows how this similarity decreases with increasing the noise level. It can be seen that TCD and HSDF have higher values of similarity at such levels of noise compared to tNM.

7.1.2 Experiment 2: Sensitivity of the similarity measures to lightening/darkening

In this experiment an images is brightened/darkened by multiplying all the pixel intensities with a constant value greater/smaller than one, respectively. The original image is compared with the modified versions of the same image. Modified versions are obtained by multiplying all pixels by a constant number ranging from 0.1 (very dark) to 2 (very bright). The nearness measures are calculated for each pair of images. Figure 21 shows the original image as well as the brightened/darkened images and the corresponding values of the nearness measures.

Discussion:
Similar to the previous experiment, images of this experiment are distorted versions of the same image. However this distortion is in the form of lightening or darkening the image by multiplying all the pixel intensities by a constant value. Therefore, as long as the similarity between the original image and the brightened/darkened image is clearly visible to human eye, we expect our similarity measure to have high values reflecting the existing similarity. Figure 21 shows how this similarity decreases with increasing the brightness (right side) or darkness (left side). It can be seen that TCD and HSDF have higher values of similarity compared to tNM.

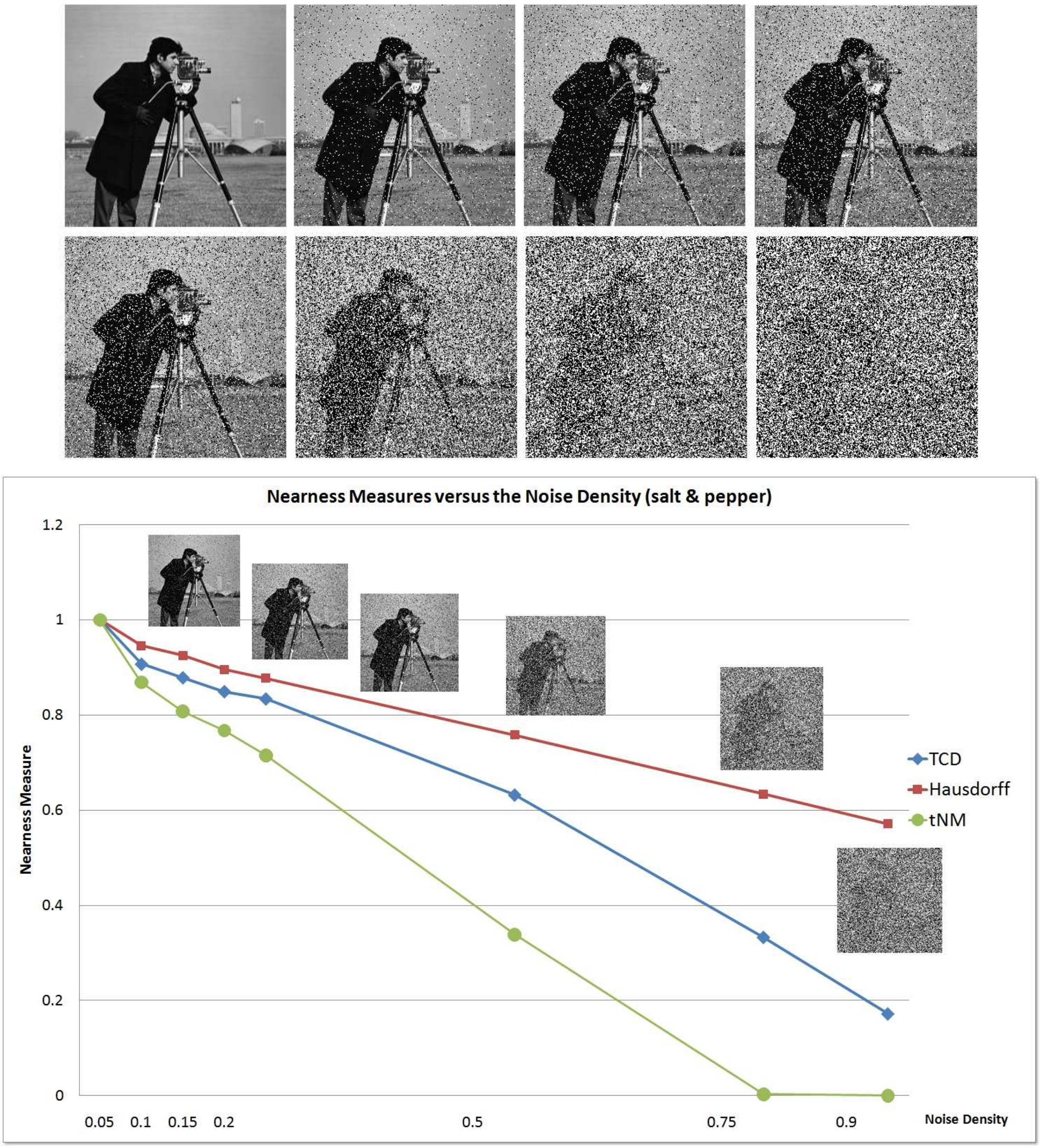


Figure 20: Noisy images with different intensity of noise (Up) and similarity measures versus noise level (Down), $\varepsilon = 0.1$

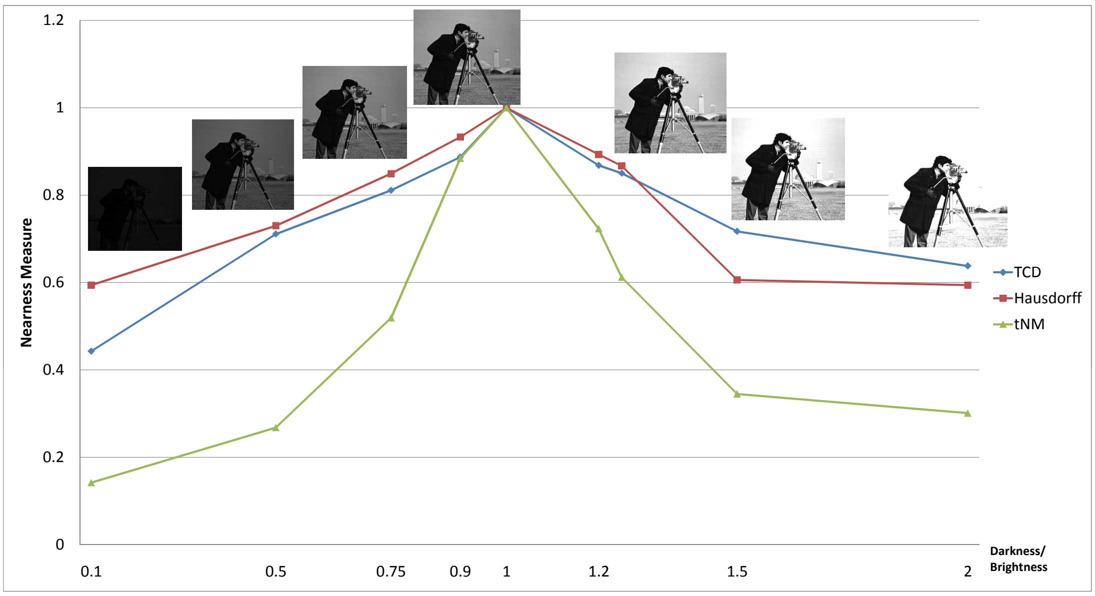


Figure 21: An image under different lighting conditions (Up) and the corresponding similarity measures versus changes in brightness (Down) $\varepsilon = 0.1$

7.1.3 Experiment 3: Sensitivity to brightness for two different image

In this experiment, all the modified versions of two images (under different brightness conditions) are considered and compared with one of the original images as the reference image. The original reference image (Lena) is shown in the top right corner of the plots in Figure 22 and modified versions of both images are shown below the plots. Values of the similarity measures are plotted for different modified versions when compared to the reference image shown in Figure 22. It can be seen how TCD and Hausdorff measures have high values of similarity between the reference image and its modified versions. However, when we are comparing two different images, we expect the similarity to be lower, as shown in the right side plot (for brightness values above 0.75 of the original brightness. Figure 22 shows the value of similarity measure between the images.

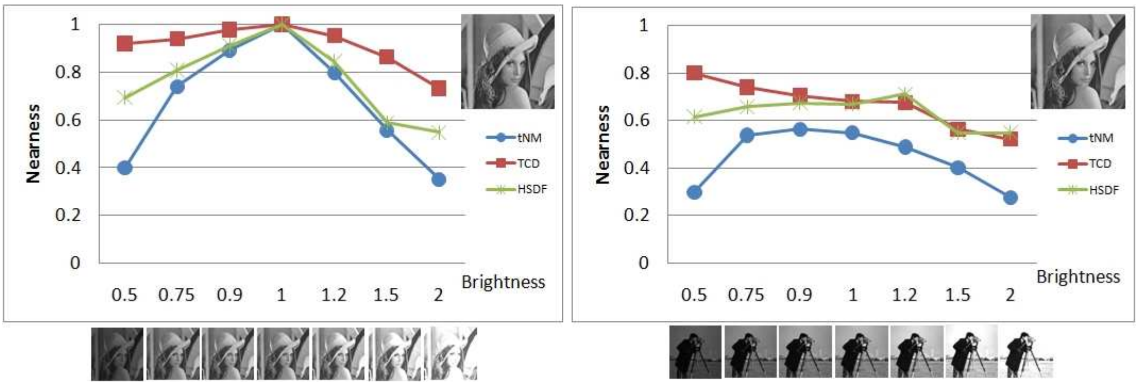


Figure 22: Nearness between different brightness levels of two images and a reference image

Table 4: Comparing an image with itself and with another image under different lightening conditions

Original														
Test image														
TCD	0.92	0.94	0.98	1.00	0.95	0.86	0.73	0.80	0.74	0.71	0.68	0.68	0.56	0.52
Hausdorff	0.69	0.81	0.91	1.00	0.84	0.59	0.55	0.62	0.66	0.67	0.67	0.71	0.55	0.55
tNM	0.40	0.74	0.89	1.00	0.80	0.56	0.35	0.30	0.54	0.56	0.55	0.49	0.40	0.28

It can be seen that again, tNM is very sensitive to the variation in brightness such that when brightness changes (both darkening and lightening), tNM value significantly drops even when image 1 is compared to itself under different lightening conditions. TCD on the other hand shows more similarity when the reference image is compared to its modified versions but is lower (less similar) when compared with a different image. Table 4 shows this information. It is important to note that only average gray level and entropy of subimages are considered as probe functions in this experiment and epsilon value is set to 0.3.

7.1.4 Experiment 4: Comparison of TCD and tNM with respect to sensitivity to noise and lightening conditions

The purpose of this experiment is to compare the proposed nearness measure (TCD) and the recently developed nearness measure (tNM) in terms of their sensitivity to noise and lightening conditions. In experiments 1 and 2, sensitivity of the nearness measures to variations in noise level and lightening was shown for one particular image as an example. In order to investigate this effect, the similarity measures between the original image and the modified version of image is calculated for a larger number of images. In this experiment, the original image is compared with 4 different modified versions of the same image for 100 test images, and the similarity measure is calculated for pairs of original image and each of its modified versions. These images are selected from the Berkeley test image dataset BSDS300 that is available online [5]. Four different variations of each image include darkening, brightening, adding low and high levels of noise to each image. Table 5 shows sample images from the test dataset in the first column as well as modified images in the next columns. Images are darkened by multiplying all the intensity levels (in all 3 color channels) by 0.6. Images are brightened by multiplying all the intensity levels by 1.4. To obtain noisy images, Gaussian noise with zero mean and variance of 0.01 and 0.05 has been added to each image to create low level and high level noisy images, respectively. The similarity measure between the original image and each one of the 4 modified versions of the image is calculated using tNM and TCD nearness measures and plotted in Figure 23 to 26. It can be seen that TCD shows better results than tNM because the similarity measure is consistently higher than tNM with less variation. tNM on the other hand has very low values in some cases that is not desirable because images are expected to by very similar.

[5] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

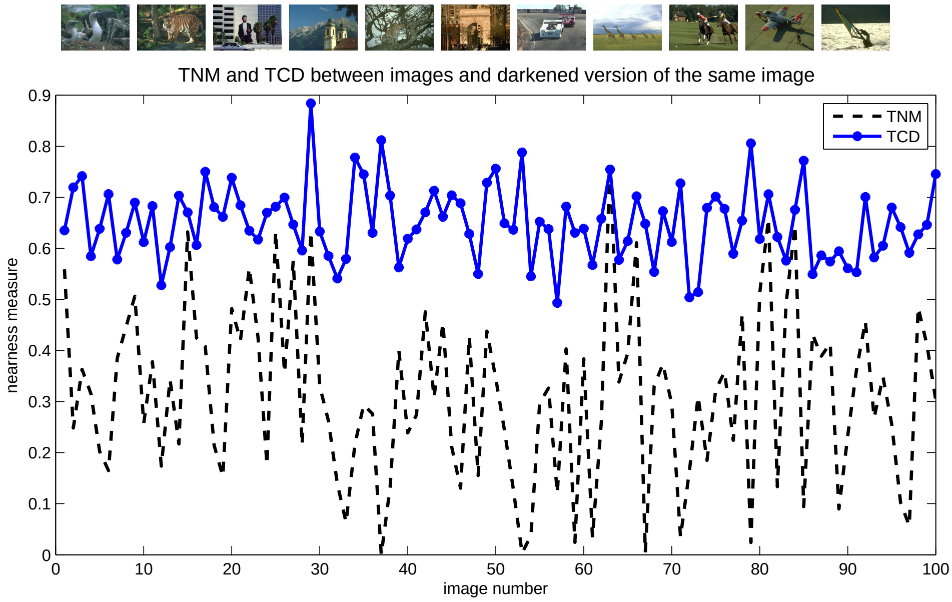


Figure 23: tNM and TCD similarity measures between an original image and the same image after darkening, plotted for all 100 images

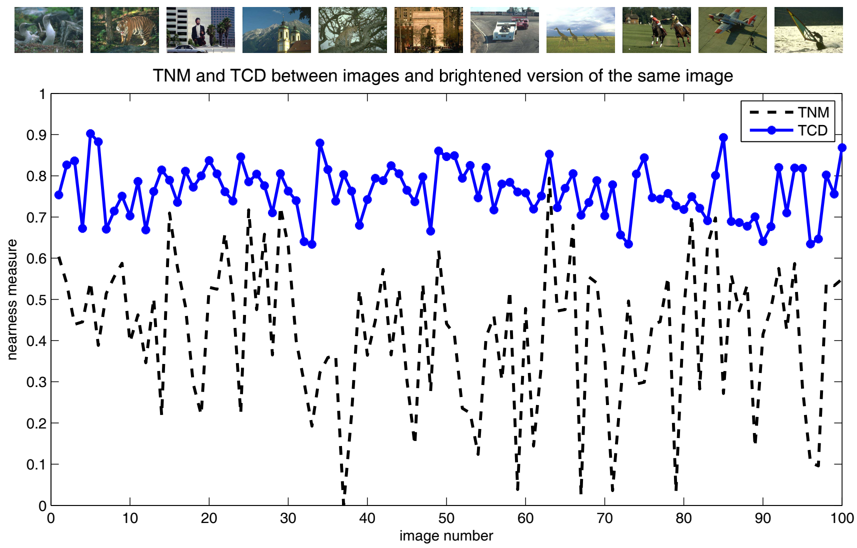


Figure 24: tNM and TCD similarity measures between an original image and the same image after brightening, plotted for all 100 images

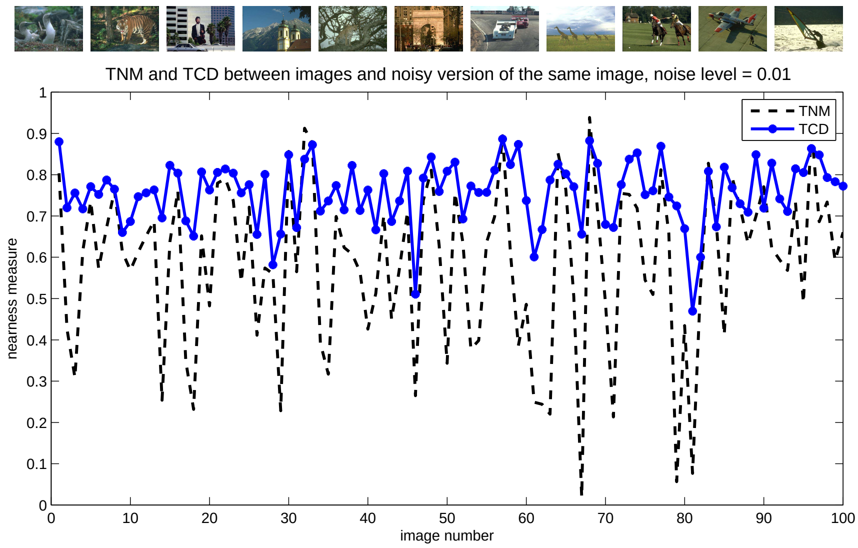


Figure 25: tNM and TCD similarity measures between an original image and the same image after adding noise with variance = 0.01, plotted for all 100 images

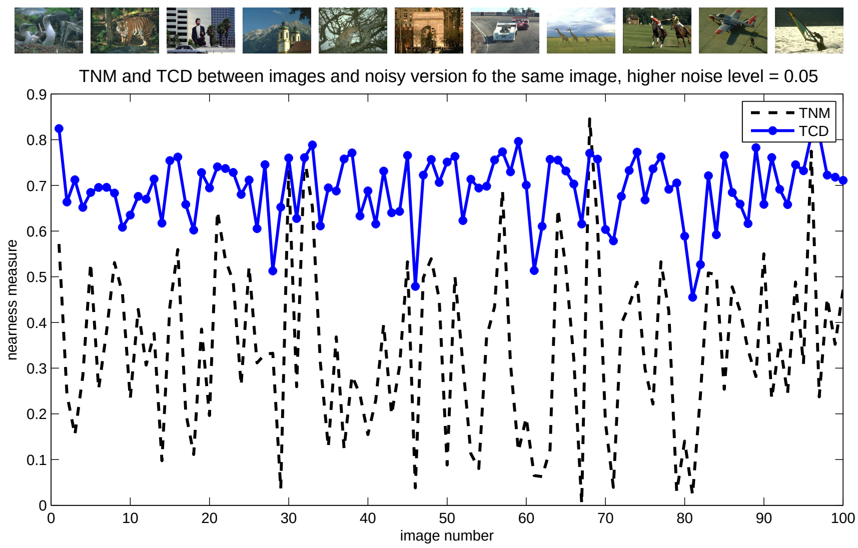
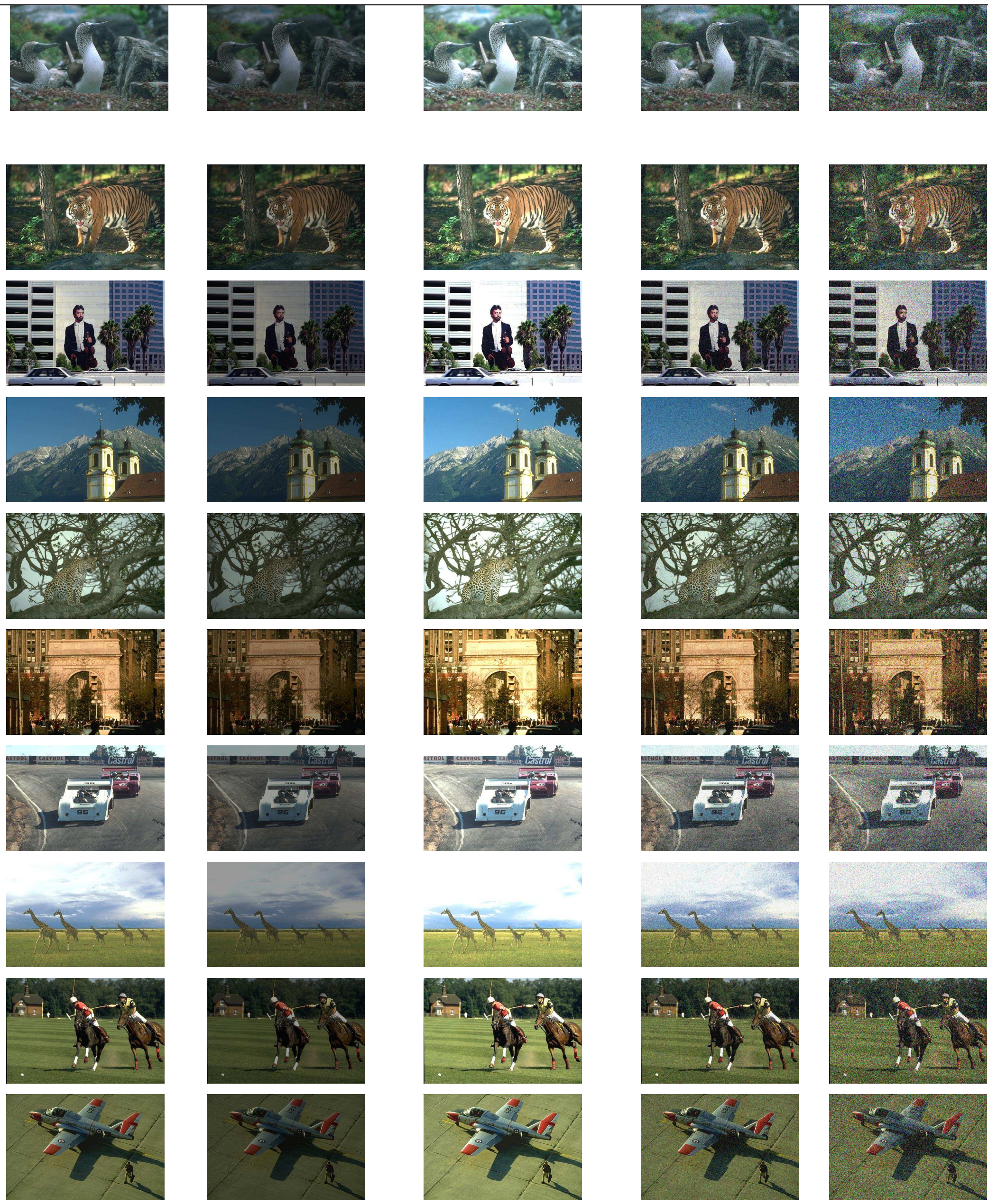


Figure 26: tNM and TCD similarity measures between an original image and the same image after adding noise with variance = 0.05, plotted for all 100 images

Table 5: Examples of images used in experiment 4 and their modified versions

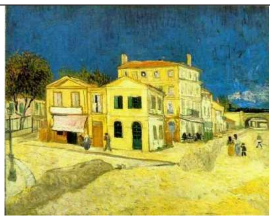
Original image	Darkened	Brightened image	Low noise	High noise



7.1.5 Experiment 5: Effect of distortion on similarity measures

In this section, sample images are compared before and after distortion when the given distortion is created by adding a twist to the images using Pain.Net program which is a free software available online. Table 6 shows an example of images before and after distortion, and the corresponding similarity measures.

Table 6: Examples of pairs of images before and after distortion and their similarity measures, $\varepsilon = 0.1$, $p = 10$.

Pair	Original	Distorted	TCD	tNM	HSDF
A			0.957	0.8025	0.974
B			0.96	0.775	0.984
C			0.9841	0.786	0.9792
D			0.9499	0.7657	0.9555

Discussion
Although distorted images in table 6 may look very different from the original images, it is important to note that this distortion applies a twist transformation on location of

the pixels and does not change the feature vectors. Therefore, we expect the similarity measures to be very high when the image and its distortion are being compared. Table 6 shows the results for TCD, tNM and Hausdorff (HSDF) similarity measures. In this experiment, both TCD and Hausdorff measures have very high values above 0.95 that reflect high level of similarity between images.

7.2 Experiments on similarity measures, toward a higher understanding of image similarity

7.2.1 Experiment 6: More on similarity measures (Van Gogh paintings)

In this experiment the performance of tNM , Hausdorff and TCD measures on image similarity between different images have been studied. Images are selected from paintings of the famous artist, Vincent van Gogh (1853-1890), and similar paintings by other artists. In the first pair, an image has been compared to itself, hence the result is expected to be 1 for all measures. The second pair consists of an image and its distorted version. Images in pair 3 to 8 are paintings that are only slightly different and we expect the similarity measures to give a high value for them since they are visually similar to each other. Table 7 shows pairs of images and the results of nearnesses measures for each pair.

Discussion:
Images in this experiment are carefully selected such that the visual similarity between them is evident to human eye. It is clear that tNM deos not perform well here because for pairs of images such as pairs 3 to 8, tNM value is extremely low and even for almost identical images such as pairs 3 and 8, tNM has low values below 0.25. While Hausdorff distance (HSDF) performs better than tNM, the best results appear to be of TCD where for almost identical images (pairs 3 and 8), TCD is greater than 0.98.

Table 7: Van Gogh paintings and their similarity measures, $\varepsilon = 0.1$, subimage size $p = 20$

Pair	Image A	Image B	TCD	tNM	HSDF
1			1	1	1
2			0.967	0.771	0.966
3			0.933	0.166	0.789
4			0.928	0.230	0.890
5			0.899	0.014	0.643
6			0.853	0.0005	0.6016
7			0.916	0.186	0.909
8			0.96	0.228	0.821

7.2.2 Experiment 7: Comparison of image similarity measures, invariance to rotation and object recognition

In this experiment, all the three different similarity measures (TCD, tNM, HSDF) are used to demonstrate how they can be used to detect similarities. In order to test the measures, an image of an object is compared with images of the same object after a given rotation as well as the images of other objects. A total of 33 images are selected randomly from the ALOI image database (Amsterdam Library of Object Images[6]) corresponding to 11 different objects with images taken from 3 different angles. The images are shown in figure 27. Each image is selected as a reference and compared with all the other images. The three most similar images (based on their nearness measure) are obtained and shown in Table 8 and 9 for all the similarity measures. It is expected that the most similar image to any given image are images of the same object. The number of images of the same object that is among the top 3 results are also counted and shown in Table 10 where the total number of similar images for each nearness measure is calculated at the bottom of the table.

Discussion
In this experiment, it is clear that Hausdorff measure performs very well and it is able to recognize images of the same object among all the images of other objects (95.9 % accurate). TCD is in the second place with an accuracy of 63.6 % and tNM has the lowest performance with an accuracy of 52.5 %.

Figure 27: Sample images of objects from ALOI database

[6]Available at: http://staff.science.uva.nl/ aloi/

Table 8: Similar images in Experiment 7, comparing nearness measures: Part 1

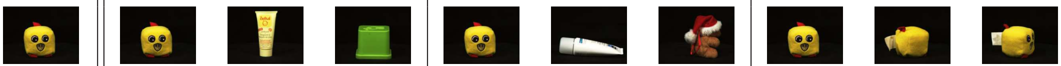
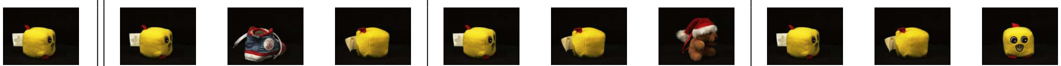
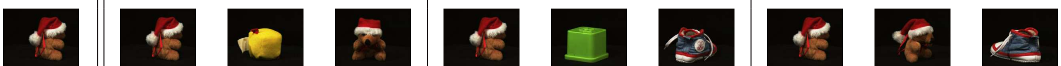
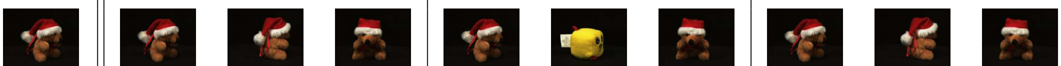
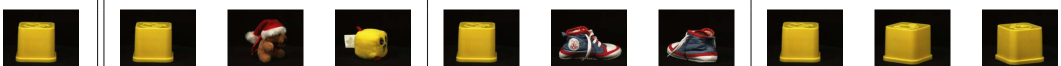
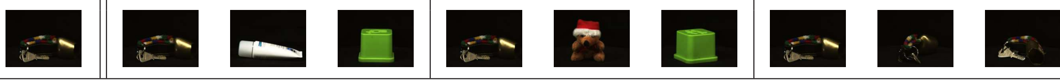
#	Image	TCD			tNM			Hausdorff		
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										

Table 9: Similar images in Experiment 7, comparing nearness measures: Part 2

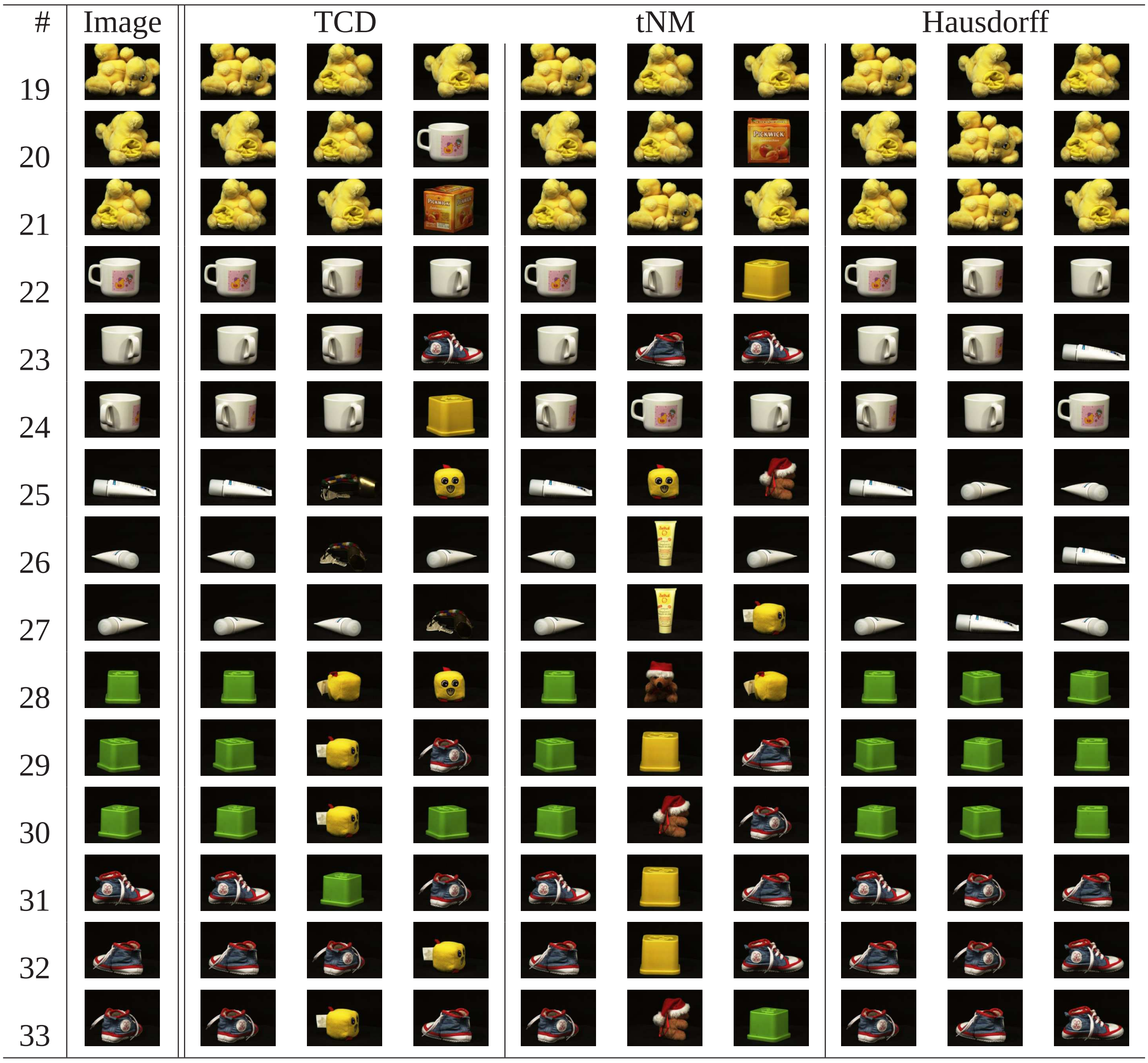


Table 10: Number of similar images in the top 3 results for each nearness measure

#	Image	TCD	tNM	Hausdorff
1		2	1	2
2		2	2	3
3		2	2	3
4		1	1	3
5		2	2	3
6		2	2	3
7		1	1	3
8		3	2	3
9		2	2	3
10		2	1	2
11		2	1	2
12		3	2	3
13		1	1	3
14		2	1	3
15		2	1	3
16		2	1	3
17		2	2	3
18		1	1	3
19		3	3	3
20		2	2	3
21		2	3	3
22		3	2	3
23		2	1	2
24		2	3	3
25		1	1	3
26		2	2	3
27		2	1	3
28		1	1	3
29		1	1	3
30		2	1	3
31		2	2	3
32		2	2	3
33		2	1	3
Sum	All	63	52	95

8 Summary and Conclusions

In this thesis, a solution to the image similarity problem (*i.e* measuring degree of perceptual similarity between pairs of images) was introduced, implemented and demonstrated. The solution is based on tolerance near set theory where the similarity is defined as the degree of nearness between sets of perceptual objects in images. Tolerance classes as sets of perceptually similar subimages are the basic building blocks of this method. The main contribution of this thesis is providing a solution to the problem of comparing tolerance classes associated with images in the form of proposing a new similarity measure based on comparing distribution of the size of tolerance classes in two images. The proposed approach in this thesis is a probabilistic approach where the image content is described by the histogram of the size of tolerance classes and the overall similarity measure is defined based on the difference between histograms. The main advantage of this approach over the existing tolerance near set methods is its lower sensitivity to distortion such as adding noise, darkening or brightening the image.

The problem with the existing approaches to tolerance near sets is that a small shift in the intensity of all the pixels in an image such as adding noise or lightening will affect the behavior of similarity measure to the extent that it does not work as expected. To solve this issue, TCD tolerance-based method that considers each image separately in its tolerance space, was introduced. The main motivation for proposing the new method in this thesis is to increase the performance of the tolerance based methods when an image is compared to a modified or distorted version of itself. A good example of such cases is when an image is lightened/darkened or when noise is added to the image.

Since adding noise or changing the brightness of an image does not change the image content, we expect the similarity measure to remain high after such modifications to the image. However, tNM nearness measure performs poorly in such cases. The reason for this problem is that tNM forms tolerance classes in the tolerance space of union of both images. Therefore, corresponding subimages of the two images will not be grouped with each other in a single tolerance class after changes in pixel intensities due to such modifications. Figure 28 demonstrates the main difference between the proposed approach (TCD) and the recently developed approach (tNM) in a tolerance space framework. In the existing methods using tolerance spaces, both images are considered in one tolerance space where tolerance classes are formed on the union of both images. In the proposed TCD approach however, each image is considered in its own corresponding tolerance space and the set of tolerance classes of each image is compared with the set of tolerance classes of the other image through calculating the difference between cumulative distributions of the size of tolerance classes.

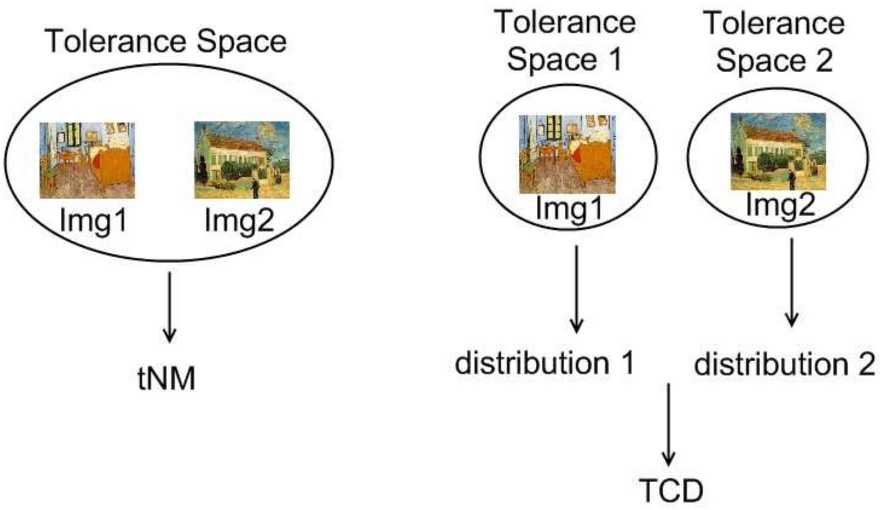


Figure 28: Using tolerance spaces in calculation of TCD and tNM nearness measures

The method used and implemented in this thesis can be summarized as:

1. **Feature extraction**: Extracting features that visually represent the images and are used to form the human perception about an image. This task involves dividing an image into small subimages and extracting the features for each subimage. The choice of features highly depends on the application and it is not the focus of this thesis.

2. **Finding tolerance spaces**: Using tolerance spaces as a basis for formulating the visual perception. This is based on the idea that tolerance classes of perceptually similar objects form the basis for perceiving similarity by humans as discussed in the works of Sossinsky, Zeeman. [43, 54].

3. **Comparing tolerance spaces** Providing a numerical measure of similarity between images by comparing distribution of the size (cardinality) of tolerance classes in both images.

The similarity measure resulting from the proposed method (TCD) was implemented and compared with another tolerance space based method namely tNM. The main difference between tNM and TCD remains in the way they form and compare tolerance spaces.

While tNM defines a tolerance space containing all the elements of both images, TCD defines two separate tolerance spaces associated with each image and then compares them with each other. This is important in overcoming the problem of high sensitivity to noise and distortion that exists in tNM.

References

[1] W. Bartol, J. Mir, K. Piro, F. Rossell, On the coverings by tolerance classes, Information Sciences 166 (1-4) (2004) 193–211.

[2] T. Bilgi, B. De Baets, O. Kaynak, D. Van der Weken, M. Nachtegael, E. Kerre, Using similarity measures for histogram comparison, in: Fuzzy Sets and Systems IFSA 2003, vol. 2715 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2003, pp. 1–9.

[3] A. Bosch, X. Muoz, R. Mart, Which is the best way to organize/classify images by content?, Image and Vision Computing 25 (6) (2007) 778–791, doi: DOI: 10.1016/j.imavis.2006.07.015.

[4] R. Datta, D. Joshi, J. Li, J. Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, ACM Comput. Surv. 40 (2) (2008) 1–60, 1348248.

[5] Flickr-Blog, http://blog.flickr.net/en/2010/09/19/5000000000/.

[6] C. Henry, Perceptual image analysis, International Journal of Bio-Inspired Computation 2 (2) (2009) *to appear*.

[7] C. Henry, J. F. Peters, Image pattern recognition using approximation spaces and near sets, in: Proc. 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 Artificial Intelligence, vol. 4482, 2007.

[8] C. Henry, J. F. Peters, Perception based image classification, Tech. rep., Computational Intelligence Laboratory, University of Manitoba, UM CI Laboratory Technical Report No. TR-2009-016 (2009).

[9] D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, Comparing images using the hausdorff distance, Pattern Analysis and Machine Intelligence, IEEE Transactions on 15 (9) (1993) 850–863.

[10] A. K. Jain, A. Vailaya, Image retrieval using color and shape, Pattern Recognition 29 (8) (1996) 1233 – 1244.

[11] Y. Ke, R. Sukthankar, L. Huston, Efficient near-duplicate detection and sub-image retrieval, vol. 2004, Citeseer, 2004, aCM Multimedia.

[12] Y. Ke, R. Sukthankar, L. Huston, An efficient parts-based near-duplicate and sub-image retrieval system, ACM, 2004, proceedings of the 12th annual ACM international conference on Multimedia.

[13] J. Li, B.-L. Lu, An adaptive image euclidean distance, Pattern Recognition 42 (3) (2009) 349–357, doi: DOI: 10.1016/j.patcog.2008.07.017.

[14] Y. Lim, S. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, Pattern Recognition 23 (9) (1990) 935–952.

[15] B. Lisa Gottesfeld, A survey of image registration techniques, ACM Comput. Surv. 24 (4) (1992) 325–376, 146374.

[16] I. Magick, http://www.imagemagick.org/script/index.php.

[17] A. H. Meghdadi, A perceptual system approach to image analysis: Tolerance classes and lattices, Tech. rep., Computational Intelligence Laboratory, University of Manitoba, E1-526 (2009).

[18] A. H. Meghdadi, J. F. Peters, Perceptual information system approach to image resemblance, in: S. K. Pal, J. F. Peters (eds.), Rough Fuzzy Image Analysis Foundations and Applications, CRC Press, 2010, pp. 8–1 to 8–23.

[19] H. Mller, N. Michoux, D. Bandon, A. Geissbuhler, A review of content-based image retrieval systems in medical applications–clinical benefits and future directions, International Journal of Medical Informatics 73 (1) (2004) 1–23.

[20] J. A. Murray, H. Bradley, W. Craigie, C. Onions, The Oxford English Dictionary, Oxford University Press, London, 1933.

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

[22] J. Peters, S. Ramanna, Affinities between perceptual granules: Foundations and perspectives, in: A. Bargiela, W. Pedrycz (eds.), Human-Centric Information Processing Through Granular Modelling, vol. 182 of Studies in Computational Intelligence, Springer Berlin / Heidelberg, 2009, pp. 49–66.

[23] J. Peters, P. Wasilewski, Foundations of near sets, Information Sciences 179 (18) (2009) 3091–3109.

[24] J. F. Peters, Classification of objects by means of features, in: Proc. IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SCCI 2007), Honolulu, Hawaii, 2007.

[25] 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.

[26] J. F. Peters, Near sets. General theory about nearness of objects, Applied Mathematical Sciences 1 (53) (2007) 2609–2029.

[27] J. F. Peters, Near Sets. Special Theory about Nearness of Objects, Fundamenta Informaticae 75 (1-4) (2007) 407–433.

[28] J. F. Peters, Classification of perceptual objects by means of features, Int. J. of Info. Technology & Intell. Computing 3 (2) (2008) 1–35.

[29] J. F. Peters, Discovery of perceptually near information granules, in: J. T. Yao (ed.), Novel Developements in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation, Information Science Reference, Hersey, N.Y., USA, 2008, p. to appear.

[30] J. F. Peters, Notes on set-theoratic relations, Tech. rep., Computational Intelligence Laboratory Seminar, University of Manitoba, E1-526 (2008).

[31] J. F. Peters, Notes on tolerance relations, Tech. rep. (Nov 2008 2008).

[32] J. F. Peters, Notes on tolerance relations, Tech. rep., Computational Intelligence Laboratory Seminar, University of Manitoba, E1-526 (2008).

[33] J. F. Peters, Tolerance near sets and image correspondence, International Journal of Bio-Inspired Computation 4 (1) (2009) 239–245.

[34] J. F. Peters, S. Ramana, Affinities between perceptual granules: Foundations and perspectives, in: A. Bargiela, W. Pedrycz (eds.), Human-Centric Information Processing Through Granular Modelling, Springer-Verlag, 2008, pp. 409–436.

[35] J. F. Peters, S. Ramanna, Feature selection: A near set approach, in: ECML & PKDD Workshop in Mining Complex Data, Warsaw, 2007.

[36] L. Polkowski, Rough Sets. Mathematical Foundations, Springer, Heidelberg, 2002.

[37] S. Ramanna, A. Meghdadi, Measuring resemblances between swarm behaviours: A perceptual tolerance near set approach, Fundamenta Informaticae 95 (4) (2009) 533–552.

[38] W. Rucklidge, Efficient visual recognition using the Hausdorff distance, Lecture Notes in Computer Science, Springer-Verlag New York, Inc., 1996.

[39] Y. Rui, T. S. Huang, S.-F. Chang, Image retrieval: Current techniques, promising directions, and open issues, Journal of Visual Communication and Image Representation 10 (1) (1999) 39–62, doi: DOI: 10.1006/jvci.1999.0413.

[40] S. Shahfar, A. H. Meghdadi, J. F. Peters, From tolerance near sets to perceptual image analysis, in: S. K. Pal, J. F. Peters (eds.), Rough Fuzzy Image Analysis Foundations and Applications, CRC Press, 2010, pp. 9–1 to 9–16.

[41] S. Siggelkow, Feature histograms for content-based image retrieval, Ph.d. dissertation, University of Freiburg (2002).

[42] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (12) (2000) 1349–1380.

[43] A. B. Sossinsky, Tolerance space theory and some applications, Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications 5 (2) (1986) 137–167.

[44] M. Still, The definitive guide to ImageMagick, Apress, 2005.

[45] M. Svedlow, C. D. McGillem, P. E. Anuta, Image registration: Similarity measure and preprocessing method comparisons, Aerospace and Electronic Systems, IEEE Transactions on AES-14 (1) (1978) 141–150.

[46] D. Van der Weken, M. Nachtegael, E. E. Kerre, Using similarity measures and homogeneity for the comparison of images, Image and Vision Computing 22 (9) (2004) 695–702.

[47] R. C. Veltkamp, M. Tanase, D. Sent, Features in content-based image retrieval systems: a survey, in: State-of-the-Art in Content-Based Image and Video Retrieval [Dagstuhl Seminar, 5-10 December 1999], Kluwer, B.V., 2001, pp. 97–124.

[48] L. Wang, Y. Zhang, J. Feng, On the euclidean distance of images, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (8) (2005) 1334–1339.

[49] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, Image Processing, IEEE Transactions on 13 (4) (2004) 600–612.

[50] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on, vol. 2, 2003.

[51] C.-Y. Yen, K. J. Cios, Image recognition system based on novel measures of image similarity and cluster validity, Neurocomputing 72 (1-3) (2008) 401–412.

[52] C. Young Deok, K. Nam Chul, J. Ick Hoon, Content-based image retrieval using multiresolution color and texture features, Multimedia, IEEE Transactions on 10 (6) (2008) 1073–1084.

[53] E. Zeeman, The topology of the brain and the visual perception, Prentice Hall, New Jersey, 1961, in K.M. Fort, Ed., Topology of 3-manifolds and Selected Topics, 240-256.

[54] E. C. Zeeman, The topology of the brain and visual perception, Topology of 3-manifolds 3 (1962) 240–248.

Author Index

Subject Index