Fuzzy Tolerance Neighborhood Approach to Image Similarity in Content-based Image Retrieval

by

Amir Hossein Meghdadi

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

Doctor of Philosophy
in
Electrical and Computer Engineering

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

Abstract

The problem addressed in this research is how to quantify the notion of similarity between two images with the main focus on content-based image retrieval (CBIR). The main strategy used here is a Near Set approach where each image is considered as a set of *visual elements* that can be described with a set of visual descriptions (features). The similarity between images is then defined as the nearness between sets of elements based on their descriptions.

The main contribution of this thesis, is to define nearness measures between sets of elements based on a *tolerance relation* and a *fuzzy tolerance relation* between pairs of elements. A tolerance relation is used here to describe the limited resolution of the human visual perception to changes in visual stimuli. Also, a fuzzy tolerance relation is adopted here to eliminate the need for a sharp threshold and hence model the gradual changes in perception of similarities by humans. A method for adaptive selection of the threshold is also introduced.

The key idea in defining similarity between images is to consider groups of visual elements that are *almost* similar to each other in description. These groups can be *tolerance classes* or *neighborhoods*. Three novel similarity measures are introduced in this thesis based on this idea. Furthermore, a fuzzy-valued similarity measure is also proposed where similarity is a fuzzy set rather than a real number. Three other similarity measures are also proposed here based on classical distances (namely, Kantorovich, Hausdorff and Mahalanobis) between sets of visual elements. All of the proposed methods are then used as similarity measures in two CBIR experiments. A new method is also proposed to evaluate image retrieval and classification. The proposed method as well as the precision-recall methods are used for evaluation. The results are also compared with other published research papers. An important advantage of the proposed methods is their effectiveness in an unsupervised environment with no prior information about images. Eighteen different features (based on color, texture and edge information) are used in all the experiments regardless of the query image. However, for comparison, a simple feature selection algorithm is used to train the system in choosing a suboptimal set of visual features and the improvement in accuracy of the results is shown.

Keywords: Perception of similarity, image similarity, nearness measure, similarity measure, content-based image retrieval (CBIR), query by content, tolerance spaces, metric spaces, fuzzy sets, fuzzy metric spaces, fuzzy tolerance spaces, fuzzy valued distance.

Acknowledgments

Many thanks to the following people

- James F. Peters (my advisor) for his moral and technical support. For his positive energy and for giving me the motivation and inspiration to complete this thesis.
- My examining committee members, Ekram Hossain, Poorang Irani, Pradeepa Yahampath and Halina Kwaśnicka for their insights and their valuable suggestions.
- My parents, for being my first teachers and lifetime mentors.
- My wife, Shabnam for her love and patience through both the good and difficult times.
- Christopher Henry, Homa Fashandi, Dan Lockery, Sheela Ramanna and Shabnam Shahfar who are some of the past and present members of our research group and more importantly great friends.
- Many thanks to Dan Lockery for his help in final proofreading of this document.

My PhD studies at the University of Manitoba was made possible with financial support from different sources including:

- University of Manitoba Graduate Fellowship (UMGF)
- Government of Manitoba Graduate Scholarship
- Manitoba Health Research Council (MHRC) Studentship
- Edward R. Toporeck Graduate Fellowship in Engineering
- Gordon P. Osler Graduate Scholarship
- University of Manitoba Entrance Scholarships
- MITACS Network of Centers of Excellence for the Mathematical Sciences
- Financial support from my advisors' various research grants.

Contents

Notation

$\mathcal{I}$	Set of images
$\mathcal{X}, \mathcal{Y} \in \mathcal{I}$	Pair of images
O	Set of describable objects
$X, Y \subseteq O$	Subsets of describable objects
$x, y \in O$	Describable objects
$\phi : O \to \mathbb{R}$	A Probe function
$\mathcal{B} = \{\phi_1, \phi_2, ..., \phi_l\}$	A set of probe functions
$\vec{\phi}_{\mathcal{B}}(x) = [\phi_1(x), ..., \phi_l(x)]^T$	Feature vector of the object x
$\Xi_{\mathcal{B}} = R_{\phi_1} \times R_{\phi_2} \times ... \times R_{\phi_l}$	Feature space $\Xi_{\mathcal{B}}$
$\Phi^O_{\mathcal{B}} \subset \Xi_{\mathcal{B}}$	Countable set of feature vectors of objects in O
$\Phi^X_{\mathcal{B}}, \Phi^Y_{\mathcal{B}} \subseteq \Phi^O_{\mathcal{B}}$	Countable subsets of feature vectors of objects in X, Y
$d : \Xi_{\mathcal{B}} \times \Xi_{\mathcal{B}} \to \mathbb{R}^+$	Feature vector distance function
$\sim_{\mathcal{B}} \subseteq O \times O$	Perceptual indiscernibility relation between objects
	$\sim_{\mathcal{B}} = \{(x, y) \in O \times O \parallel \vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y)\}$
$x_{/\sim_{\mathcal{B}}} \subset O$	Equivalence class of all the elements indiscernible from x
$O_{/\sim_{\mathcal{B}}}$	Set of all equivalence classes in O determined by $\sim_{\mathcal{B}}$
$\cong \subseteq O \times O$	A tolerance relation
$\varepsilon \in \mathbb{R}^{+0}$	Tolerance level (threshold)
$\cong_{\mathcal{B},\varepsilon} \subseteq O \times O$	Perceptual tolerance relation on O with respect to d and ε.
	$\cong_{\mathcal{B}} = \{(x, y) \in O \times O \ \mid d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon\}$
$(O, \cong_{\mathcal{B},\varepsilon})$	Tolerance space.

Notation

$n_O^{\cong_{\mathcal{B},\varepsilon}}(x) \in \mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$	Tolerance neighborhood of x in O
$\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$	Set of all tolerance neighborhoods in O
$\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$	Set of maximal pre-calsses of the tolerance space $(O, \cong_{\mathcal{B},\varepsilon})$
$\hat{\cong}_{\mathcal{B},\varepsilon} : O \times O \longrightarrow [0,1]$	Fuzzy perceptual tolerance relation (soft similarity)
$n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}(x)$	Fuzzy tolerance neighborhood of x on O where $\mu_{(n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}})} : O \longrightarrow [0,1]$
$\mathcal{F}(U)$	Family of all possible fuzzy sets defined on a set U
$B_*^{\cong_{\mathcal{B},\varepsilon}}(X)$	Lower approximation of a subset X in a tolerance space
$B^*_{\cong_{\mathcal{B},\varepsilon}}(X)$	Upper approximation of a subset X in a tolerance space
pdf	Probability distribution function of a continuous random variable
cdf	Cumulative distribution function of a continuous random variable
pmf	Probability mass function of a discrete random variable

List of Tables

List of Figures

Chapter 1

Introduction and Classical Measures

The objective of this research is to develop computational methods for quantifying the visual similarity between pairs of digital images. The problem of defining such similarity measures is an important part of content-based image retrieval (CBIR) systems [11, 63]. A solution to this problem generally consists of extracting some visual descriptions (*i.e* features) from images and then comparing the feature values in order to measure how much they resemble each other. The motivation for this stream of research is to work toward reaching the ability of humans in perceiving images and recognizing similarity using computational intelligence. This is one of the most important aspects of the human intelligence in perceiving the world around us. We can detect, organize and explore the world based on similarity between objects or physical stimuli. However, mimicking human intelligence in recognizing similarity is particularly challenging because the correspondence between physical measurable stimuli and its perception (and further psychological experience of the perception) is unknown. Perception of similarity based on visual stimuli has been always an interesting subject for scientists in psychophysics and cognitive psychology and many attempts have been made to model this relation (see Appendix A). This thesis does not include the psychophysics of visual perception. The research problem in this thesis is to develop a computational algorithm for quantifying the level of visual similarity between images. This assumes that some meaningful visual descriptions of the images (features) are provided.

A common classical approach to comparing features is to calculate some form of distance (*e.g.* Minkowski distance: Euclidean or Manhattan distance) either between global or local feature vectors in the feature space. This approach is often called *geometric approach* or *mental distance approach* [42]. For example, in [29, 33], and [34], a weighted Minkowski distance is used to compare global color or texture-based feature vectors. In [27], Euclidean distance is used to compare feature vectors that are extracted using a wavelet decomposition to represent texture. Figure 1.1 for example, is a drawing that shows how each image can be mapped into a point in 3 dimensional feature space. Image dissimilarity can then be viewed as distance between the corresponding points. Measures of distance between histograms have also been used as a measure of dissimilarity. A couple of examples include *histogram intersection* [32] and L_p norm distance between histograms [66]. To learn more about the existing methods in image similarity, the reader is encouraged to refer to [3, 11, 63].

The main approach presented in this research, is based on comparing images as sets of elements in a tolerance space and fuzzy tolerance space. The idea of using tolerance spaces in mathematical modeling of visual perception was introduced by Zeeman in 1962 [80] (inspired by the early qualitative discussions by Henry Poincaré [52]). Zeeman proposed a tolerance view of similarity to represent limitation of human visual acuity in distinction between visual stimuli that are spatially apart [80]. Zeeman's work was not concerned with

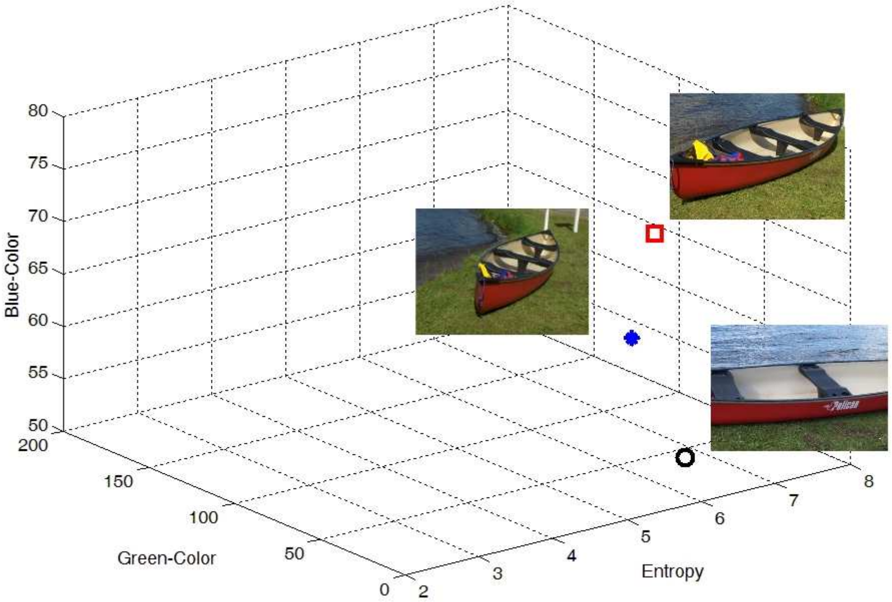


Figure 1.1: Distance based approach to similarity

the concept of visual similarity but rather with a mathematical modeling of how the visual perception is formed. Although Zeemnas's model of the visual perception is very elementary, (even as described by himself in his paper) it has a reasonable basis which has inspired the research in this thesis on the objective of devising new methods to emulate the visual perception in similarity detection. This tolerance space-based approach to perception of similarity is also influenced by the observation about perception made by Ewa Orłowska in 1982, *i.e.*, "*classes defined in an approximation space serve as a formal counterpart of perception*" [45, 46]. The mathematical foundations of tolerance space theory in modeling uncertainty in the real world was further elaborated by Sossinsky in 1986 [65] .

The proposed approach in this thesis can be briefly explained in two levels as follows,

- In a lower level, similarity between image visual elements is modeled in a classical geometric approach where a metric is used to define the distance between local feature vectors as an indication of dissimilarity.

- In a higher level, sets of visual elements are compared to each other by forming tolerance neighborhoods of visually similar elements and measuring distance between sets through analyzing how tolerance neighborhoods cover both images.

Consider each image as a set of *visual elements*. Each visual element is a part of the image (a pixel or a group of pixels) that can be visually perceived and mathematically described by a set of features and can be named as a *describable object*. Figure 1.2 shows an example pair of images, their visual elements displayed individually and in 3D feature space. At a lower level, it is necessary to be able to detect similarities between visual elements based on their description. At this level, parts of image(s) that have similar visual descriptions, can be easily identified. For higher level comparisons it is more challenging to establish the overall similarity between pairs of images based on similarity of elements in the images. Dissimilarity between any two elements can be defined by a common distance function between the corresponding feature vectors in a feature space. However, similarity between two sets of elements is a more complicated task and it is the subject of this research in the following chapters. Figure 1.3 shows pairs of images and their corresponding sets of visual elements in the feature space overlaid on each other.

Figure 1.4, provides an outline of the research in this thesis. The main approach in this research is based on *Near Sets* theory (*i.e* measuring the nearness between sets of describable objects using the nearness between the elements in the sets). Near Set theory, introduced by James F. Peters in 2006 [48, 49, 51] lays out the necessary foundation for defining the similarity between sets of objects based on their description. The methodologies used in this research to represent the uncertainty and imprecise nature of the concept of similarity in the human mind consists of:

1. A tolerance space-based approach is introduced that relaxes the equality requirement of descriptions into an *almost equal requirement* when comparing elements based on their descriptions.

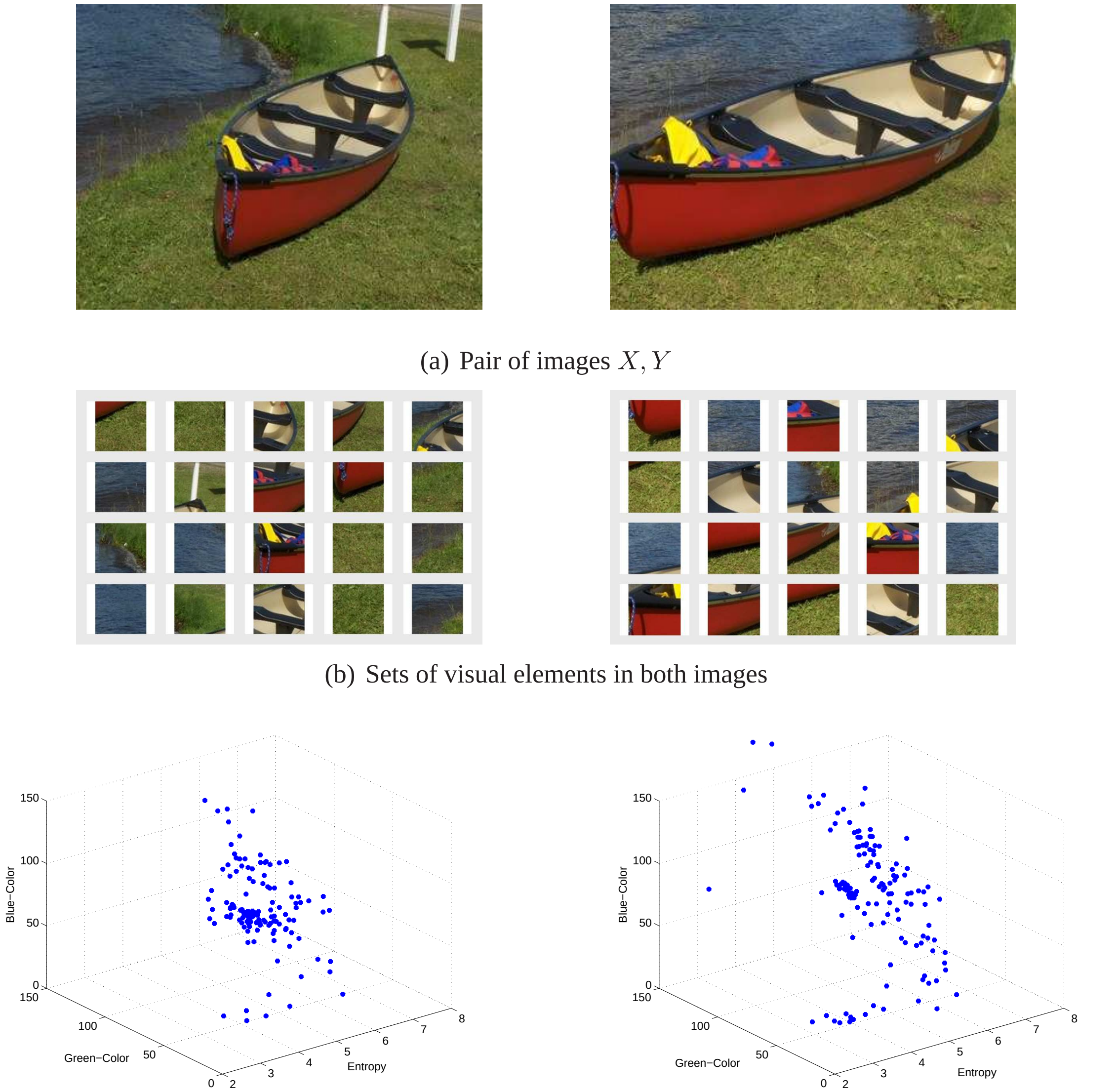
2. A fuzzy set-based approach is used to allow for soft (gradual) transition between *equal*, *almost equal* and *not equal* descriptions.

1.1 Motivation

The main motivation for this research was to bridge the existing gap between the algorithmic, computer-generated similarity measures (mainly based on geometric distance approach) and human judgment of similarity which is both *imprecise* and *fuzzy* by nature and yet more reliable.

1.2 Contributions

The main contribution of this thesis is the proposal, implemention and analysis of methods to use *tolerance space* and *fuzzy set* theories in defining the visual similarity between pairs of images. A complete content based image retrieval system for evaluating the proposed methods is also implemented.

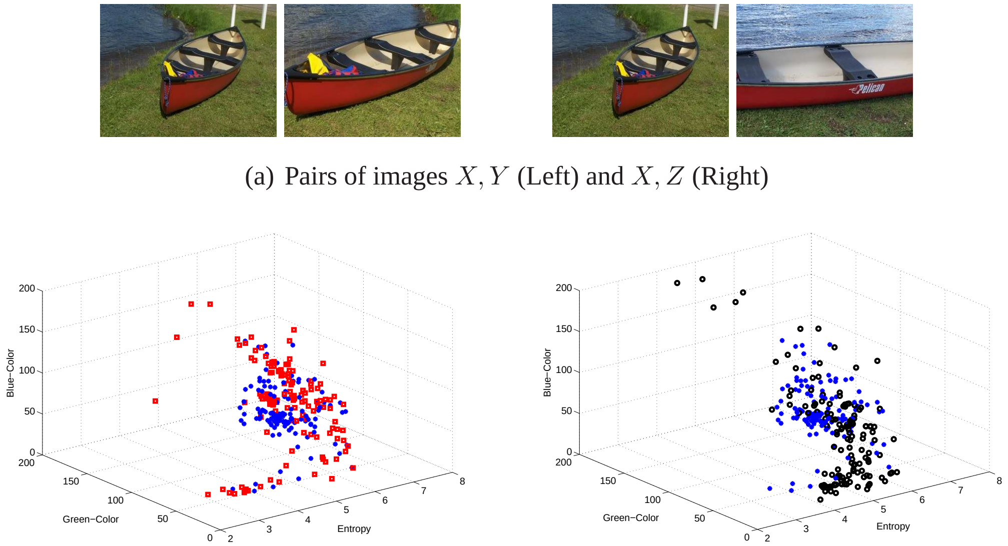


(a) Pair of images X, Y

(b) Sets of visual elements in both images

(c) Sets of visual elements in a 3-D feature space

Figure 1.2: Sample pairs of images viewed as sets of visual elements in the feature space.



(a) Pairs of images X, Y (Left) and X, Z (Right)

(b) Subimages of X, Y (Left) and X, Z (Right), respectively, in the feature space

Figure 1.3: Distance based approach to similarity. Sets of local feature values

The contributions are elaborated in more detail in the list that follows,

- Introducing and implementation of tolerance neighborhood-based methods for comparing images based on their content.

- Introducing a novel fuzzy tolerance relation to describe degree of nearness between visual elements.

- Introducing new distance/nearness measures based on both tolerance space methods (named here as $tcDM$ and $ftcDM$ for conventional and fuzzy tolerance relations, respectively) and also classical methods (Hausdorff, Mahalanobis and Kantorovich distance) in image feature space (named here as $HdNM, gMNM, KdNM$ respectively).

- Introducing and using a probabilistic approach for representing sets of images and analysis of the neighborhood distance as the building block of $tcNM$ tolerance based nearness measure. Providing mathematical proofs and simulation results to show how neighborhood distances depend on distribution of features in sets of images.

- Introducing a method for automatic selection of the tolerance level ε in the tolerance relation based methods.

- Proposing a novel tolerance Rough Set approach to describe a region of interest in an image within a tolerance space used for region-based image similarity.

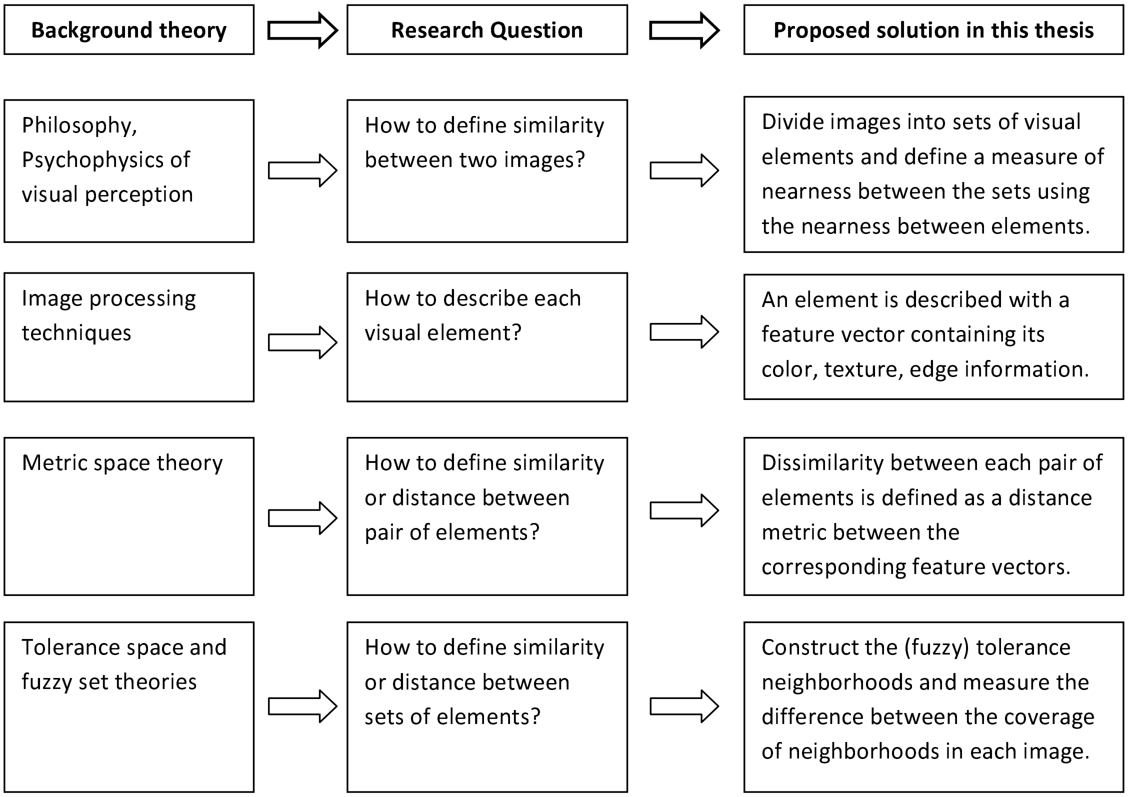
- Introducing the concept of a fuzzy distance function (a fuzzy valued distance) based on a fuzzy tolerance relation (namely $ftcFDM$). The fuzzy distance function represents the distance with a fuzzy set rather than a real number.

- Using the concepts of *narrow-domain* and *broad-domain* in describing a set of images to categorize the search problem (not the set of images) as *broad-target broad-domain*, *narrow-target broad-domain*, and *narrow-target narrow-domain*. [1]

- Implementation of all the proposed nearness/distance measures in a CBIR framework on two controlled test datasets of images. A benchmark publicly available "broad-target broad-domain" dataset and a new personally generated dataset of "narrow-target, broad-domain" images.

- Introducing a new *target discrimination matrix* (TDM) for evaluating the performance of a CBIR (or image classification). Implementing both the new method and the existing performance-recall method on the above systems.

[1] These terms are coined here for the first time, to identify and distinguish between different types of CBIR problems.



Background theory	Research Question	Proposed solution in this thesis
Philosophy, Psychophysics of visual perception	How to define similarity between two images?	Divide images into sets of visual elements and define a measure of nearness between the sets using the nearness between elements.
Image processing techniques	How to describe each visual element?	An element is described with a feature vector containing its color, texture, edge information.
Metric space theory	How to define similarity or distance between pair of elements?	Dissimilarity between each pair of elements is defined as a distance metric between the corresponding feature vectors.
Tolerance space and fuzzy set theories	How to define similarity or distance between sets of elements?	Construct the (fuzzy) tolerance neighborhoods and measure the difference between the coverage of neighborhoods in each image.

Figure 1.4: Outline of the research questions and proposed solutions in 4 steps

- Designing a user friendly program for image analysis and retrieval experiments [2] (POINCaRé: Program for Object and Image Nearness Comparison and Recognition).

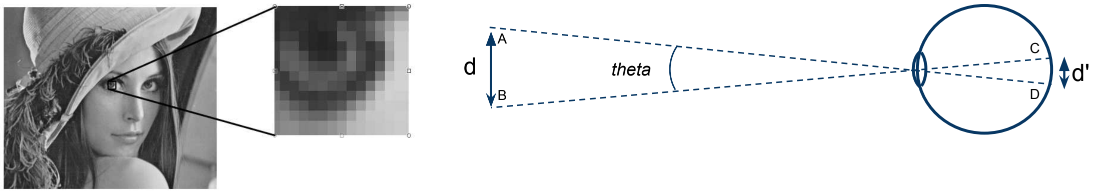
[2] The program is named after Jules Henri Poincaré (1854 - 1912) whose work on the philosophical aspects of the contrast between the mathematical and physical continua laid out the idea of tolerance space theory. POINCaRé is written in MATLAB. A simplified executable version of the program can be downloaded from the Computational Intelligence Laboratory web site. `http://wren.ece.umanitoba.ca/`

1.3 Describing an Image as a Set of Visual Elements

This research is based on a set theoretic approach to image analysis where each image is viewed as a set of *visual elements* (or more generally, *describable objects*). Each visual element can be just a pixel, a pixel and its surrounding pixels or any part of the image that can be visually perceived and described. The reason behind working with a visual element rather than a single pixel, has both a practical and a physical aspect. From a practical point of view, it is easier to consider a small patch of adjacent pixels as a *unit* of visual perception and thus reducing the amount of information needed to represent the image as it is perceived by a human. From a physical point of view, we know that we do not see images in a pixel-based resolution and our local perception of the image is formed by a group of pixels. The size of a visual element thus represents the granularity of the visual system. Figure 1.5(b) shows a simple image of how two distinct points A and B are projected on retina and stimulate the photo-receptor cells named *cone* at points D and C. The two points can be perceived as distinct, if they stimulate two different cones on the retina that are separated by at least one other cones [31]. For a normal healthy human eye with a pupil size of 8 mm in diameter, the angle of resolution is about 1 minute of arc or 1/60 of a degree [15, 31]. The maximum resolution in terms of the distance between pixels on the image, can be calculated using a little geometry. Assuming that the distance between the image and the lens is r, the minimum separable distance on an image will be:

$$d_{min} = 2 \times r \times \tan\left(\frac{1}{120}\right)^{\circ}$$

For the sake of practical simplicity, we consider each visual element to be a small square



(a) An example of a subimage

(b) Spatial resolution of human vision

Figure 1.5: An example of a visual element (Left). Forming an image on retina (Right)

of size p in pixels and we call it a *subimage*. Visual elements may or may not have overlap. The choice of p is optional. A useful guideline for choosing p is to be small enough to represent local details in an image and large enough to limit the number of visual elements for the sake of speed in the algorithm. Figure 1.5(a) shows an image of size 355 by 300 pixels and a sample visual element (describable object) as a square subimage of size 13 by 13 pixels.

1.3.1 Visual elements as describable objects

In this thesis, we adopt the viewpoint and terminology of Near Set theory as described in [51]. The elements of a set in Near Set theory are those that represent something in the physical world and hence they can be perceived and described. Such an element is called here a *describable object* [3]. Describing the element is possible through a set of characteristics (features). A *visual element* (as described in the previous section) is a good example of a *describable object*. A subimage for example is part of an image that can be perceived and described by color or texture. While the subimage contains local visual information about the image, the set of describable objects represents the whole image.

[3] Originally, the term *perceptual object* was used to represent an object that can be perceived and described.

Let $x \in X$ be a describable object in a set. In order to represent features of a describable object, the concept of *probe function* $[50, 51]$ is used here. A probe function is denoted by ϕ and is defined as a function that maps a describable object (such as a visual element x) to a feature value that is denoted by $\phi(x)$. $\mathcal{B}$ is used to denote a set of probe functions that together are used to describe the object.

$$\mathcal{B} = \{\phi_1, \phi_2, ...\phi_l\} \tag{1.1}$$

The set of all feature values for a describable object x can be organized in a vector format as follows (feature vector):

$$\vec{\phi}_{\mathcal{B}}(x) = [\phi_1(x), \phi_2(x), ...\phi_l(x)]^T \tag{1.2}$$

1.3.2 Metric distance between describable objects

Similarity between each pair of describable objects can be measured in a metric approach. In this approach, a distance is defined between each pair of objects in a space called *feature space* as a measure of dissimilarity between the objects. The feature space is the space of all possible feature vectors.

Let $\Xi_{\mathcal{B}}$ be the space of all possible feature vectors defined by a set of probe functions $\mathcal{B}$. Describable objects x, y are represented with their corresponding feature vectors $\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y) \in \Xi_{\mathcal{B}}$. Minkowski distance d of order p between pairs of feature vectors is defined as follows:

$$d : \Xi_{\mathcal{B}} \times \Xi_{\mathcal{B}} \longrightarrow \mathbb{R}^{+}$$

$$d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) = \parallel \vec{\phi}_{\mathcal{B}}(x) - \vec{\phi}_{\mathcal{B}}(y) \parallel_p = \left(\sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|^p \right)^{\frac{1}{p}} \tag{1.3}$$

where $\parallel . \parallel_p$ is called p-norm ($p \in \mathbb{N}$). Minkowski distance is chosen here as an elementary distance between visual elements (describable objects).

Proposition 1.1. $(\Xi_{\mathcal{B}}, d)$ *is a metric space.*

Proof. The proof for $p \in \mathbb{N}$ is as follows:

$(\Xi_{\mathcal{B}}, d)$ has all the required properties of a metric space namely,

$\forall \vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y), \vec{\phi}_{\mathcal{B}}(z) \in \Xi_{\mathcal{B}}$

Part 1) Non-negativity: $d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) \geq 0 \quad .$

Proof directly follows from the definition in Equation 1.3.

Part 2) Identity of indiscernibles: $d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) = 0$ if and only if $\vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y)$.

A) $d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) = 0 \Rightarrow \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|^p = 0$

$\Rightarrow |\phi_k(x) - \phi_k(y)|^p = 0 \quad \forall k \in \{1, 2, ..., l\}$

$\Rightarrow \phi_k(x) = \phi_k(y) \quad \forall k \in \{1, 2, ..., l\} \Rightarrow \vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y)$.

B) $\vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y) \Rightarrow d = 0$ directly follows from definition.

Proof follows from A and B.

Part 3) Triangle inequality: $d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(z)) \leq d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) + d(\vec{\phi}_{\mathcal{B}}(y), \vec{\phi}_{\mathcal{B}}(z))$.

Proof: $\left(d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(z))\right)^p = \sum_{k=1}^{l} |\phi_k(x) - \phi_k(z)|^p = \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y) + \phi_k(y) - \phi_k(z)|^p$

Using the binomial theorem in elementary algebra, we know that $\forall a, b \in \mathbb{R},\ p \in \mathbb{N}, \quad |a +$

$b|^p \leq |a|^p + |b|^p$, therefore:

$$\left(d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(z))\right)^p \leq \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|^p + |\phi_k(y) - \phi_k(z)|^p$$

$$= \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|^p + \sum_{k=1}^{l} |\phi_k(y) - \phi_k(z)|^p = d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y))^p + d(\vec{\phi}_{\mathcal{B}}(y), \vec{\phi}_{\mathcal{B}}(z))^p.$$

Again, using elementary algebra, we know that $\forall r, s, t \in \mathbb{R},\ p \in \mathbb{N} \quad r^p \leq s^p + t^p \Rightarrow r \leq s + t$, therefore:

$$d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(z)) \leq d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) + d(\vec{\phi}_{\mathcal{B}}(y), \vec{\phi}_{\mathcal{B}}(z)) \qquad \square$$

Proposition 1.2. $(\Xi_{\mathcal{B}}, \| \ . \ \|_p)$ *is a normed vector space.*

Proof follows from the fact that $(\Xi_{\mathcal{B}}, d)$ *is a metric space.*

Let $X, Y \subseteq O$ be sets of describable objects and let $\Phi_{\mathcal{B}}^X, \Phi_{\mathcal{B}}^Y, \Phi_{\mathcal{B}}^O \subset \Xi_{\mathcal{B}}$ be sets of feature vectors corresponding to elements of X, Y and O respectively.

Corollary 1.1. *Every non-empty finite set of feature vectors along with the above distance function forms a finite metric space.*

- $(\Phi_{\mathcal{B}}^X, d) \quad d : \Phi_{\mathcal{B}}^X \times \Phi_{\mathcal{B}}^X \rightarrow \mathbb{R}^+$ *is a finite metric space.*

- $(\Phi_{\mathcal{B}}^Y, d) \quad d : \Phi_{\mathcal{B}}^Y \times \Phi_{\mathcal{B}}^Y \rightarrow \mathbb{R}^+$ *is a finite metric space.*

- $(\Phi_{\mathcal{B}}^O, d) \quad d : \Phi_{\mathcal{B}}^O \times \Phi_{\mathcal{B}}^O \rightarrow \mathbb{R}^+$ *is a finite metric space.*

1.4 Image Similarity and Classical Measures of Nearness between Sets

Nearness between sets of describable objects can be defined using a classical method of distance between sets. In this section, 3 distance/nearness functions are introduced based on classical methods of Kantorovich distance, Mahalanobis distance, and Hausdorff distance. While these measures are well-known methods, their application in finding the distance between sets of visual elements, especially the generalized Mahalanobis distance, is part of the contribution of this thesis.

1.4.1 Kantorovich distance based nearness measure (KdNM)

A nearness measure based on the *Kantorovich distance* [12,73] between the histograms of the feature values is introduced in this section. This method assumes a probabilistic nature for images, where the feature values are sample outcomes of a random variable. The underlying probability distribution of the features may be unknown. However, histograms of the feature values are considered as an estimate of the true underlying distribution functions (empirical distributions).

Definition 1.1. Kantorovich Distance

Suppose F and G are distribution functions of random variables μ and ν, respectively. The L_1 norm based Kantorovich *metric [18, 72] is defined as*

$$d_w(\mu,\nu) = \int\limits_{-\infty}^{\infty} |F(x) - G(x)|dx = \int\limits_{0}^{1} |F^{-1}(t) - G^{-1}(t)|dt, \tag{1.4}$$

where F^{-1} and G^{-1} are quantile functions of the distributions. It can be proved that the Kantorovich metric is equal to the Wasserstein metric [18]. Although the original Kantorovich metric uses the L_1 norm, this metric can also be similarly defined using L_p norms ($p > 1$).

Suppose X and Y are sets of describable objects corresponding to a pair of images. For each probe function $\phi_k \in \mathcal{B}$ ($k = 1, 2, ..., l$), feature values are normalized between 0 and 1 and the histogram of the features are calculated to represent empirical density functions. Let $\{b_1, ..., b_j, ...b_{N_b}\}$ be the set of bins in calculation of histograms. For k^{th} feature value, let $H_X^k(b_j)$ be the number of describable objects of image X where its k^{th} feature value belongs to j^{th} bin. Distances between distributions are calculated for all the features. Here, cumulative histograms rather than plain histograms have been used because cumulative histograms tend to be more robust to changes in the bin assignment [63, 66]. The normalized cumulative histogram of k^{th} feature value in image X is defined in Equation (1.5).

$$CH_X^k(b_j) = \left(\sum_{i=1}^{i=j} H_X^k(b_i) \right) / \left(\sum_{i=1}^{i=N_b} H_X^k(b_i) \right). \tag{1.5}$$

Definition of $CH_Y^k(b_j)$ for image Y is similar. Different methods can be used to measure the difference (distance) between histograms [54, 63]. For examples, comparing the statistical moments of histograms individually or in combination with each other such as the Mahalanobis distance [35] or weighted-mean-variance as proposed in [38] that considers the difference between the mean of distributions with reference to the variances. Here, a Kantorovich distance between cumulative histograms is used. Corresponding to each descriptive feature (probe function ϕ_k), a feature specific distance $d_k(X, Y)$ between images X and Y with respect to k^{th} feature, is defined using Equation 1.6.

$$d_k(X,Y) = \sum_{j=1}^{j=N_b} \mid CH_X^k(b_j) - CH_Y^k(b_j) \mid . \tag{1.6}$$

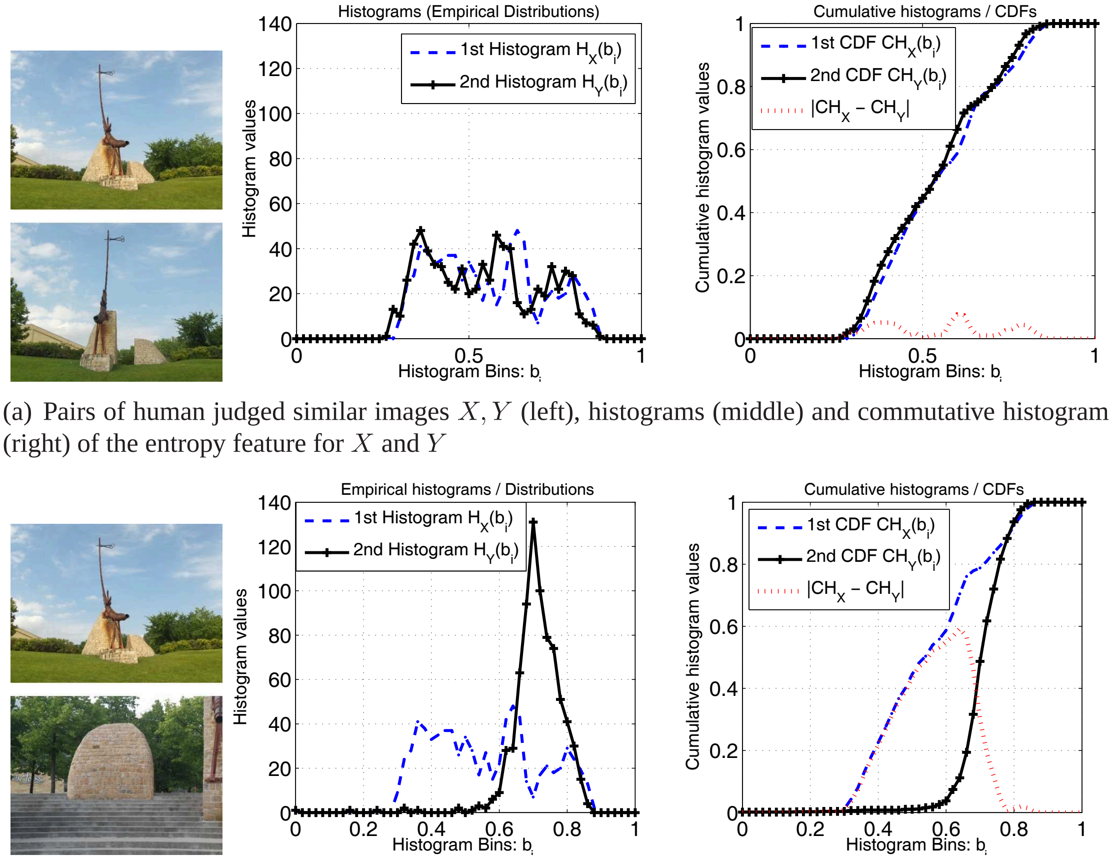
Finally, distance (dissimilarity measure) between sets X and Y is named KDM and defined in Equation 1.7 as the L_p norm ($p = 1$ here) of the distance vector $\vec{d} = [d_1, ..., d_l]^T$ where $l = |\mathcal{B}|$ is the number of feature values. The Kantorovich distance based nearness measure ($KdNM$) is defined by normalizing the distance and converting it to nearness measure as shown in Equation 1.8.

$$KDM(X,Y) = \frac{1}{l}\sum_{k=1}^{l} \mid d_k(X,Y) \mid , \tag{1.7}$$

$$KdNM(X,Y) = 1 - \sqrt{KDM(X,Y)}, \tag{1.8}$$

where $l = |\mathcal{B}|$ is the maximum possible value of $KDM(X,Y)$.

Example 1.1. *Figure 1.6 shows sample pairs of images and their corresponding distribution function of a feature value (“entropy” of subimage). Human judgment on similarity between the first pair of images (X and Y), is higher than the second pair of images (X and Z). Empirical distributions and Empirical cumulative distribution functions (CDF) are also displayed and the difference between the two distribution functions is measured by the area under the function $\mid CH_X^1(b_j) - CH_Y^1(b_j) \mid$ (marked with dotted line). It can be seen that a more similar pair of images (1.1) has less distance between the corresponding histograms.*



(a) Pairs of human judged similar images X, Y (left), histograms (middle) and commutative histograms (right) of the entropy feature for X and Y

(b) Pairs of human judged similar images X, Z (left), histograms (middle) and commutative histograms (right) of the entropy feature in X and Z

Figure 1.6: Difference between histograms for pairs of (a) similar and (b) dissimilar images

1.4.2 Generalized Mahalanobis distance nearness measure (gMNM)

The Mahalanobis distance [35] is a form of distance between two points in the feature space with respect to the variance of the distribution of points. The original Mahalanobis distance is defined between two sample multivariate vectors $\vec{x}$ and $\vec{y}$ as follows [14]

$$D_M(\vec{x}, \vec{y}) = (\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y}), \tag{1.9}$$

where the vectors are assumed to have a normal multivariate distribution with the covariance matrix Σ. This formula can also be used to measure the distance $D_M(\vec{x}, \vec{m})$ between a vector $\vec{x}$ and the mean of the distribution $\vec{m}$. Following the same approach, the Mahalanobis distance can be used to define a distance measure between two separate distributions. Lets assume $\chi_1 = (\vec{m_1}, \Sigma_1)$ and $\chi_2 = (\vec{m_2}, \Sigma_2)$ are two normal multivariate distributions with means $\vec{m_1}$,$\vec{m_2}$ and covariance matrices Σ_1, Σ_2. Let $P(\omega_1)$ and $P(\omega_2)$ represent prior probabilities of the given distributions. A generalized Mahalanobis distance [4, 14] between the two distributions is defined in (1.10).

$$gMD(\chi_1, \chi_2) = \sqrt{(\vec{m_1} - \vec{m_2})^T \Sigma_W^{-1} (\vec{m_1} - \vec{m_2})}, \tag{1.10}$$

where Σ_W^{-1} refers to the within-class covariance matrix defined in (1.11).

$$\Sigma_W = \sum_{i=1,2} \left(P(\omega_i) \sum_{x \in \chi_i} \frac{(x - m_i)(x - m_i)^T}{n_i} \right). \tag{1.11}$$

The above approach is used here to compare distributions of feature values of describable objects in two images. Generalized Mahalanobis distance-based nearness measure ($gMNM$) between two images is defined as follows.

Let X and Y denote sets of describable objects (images). Let $\bar{\Phi}_{\mathcal{B}}^X$ and $\bar{\Phi}_{\mathcal{B}}^Y$ represent the mean feature vector for all the describable objects $x \in X$ and $y \in Y$, respectively. Also, let Σ_X and Σ_Y be the covariance matrices of the multivariate distributions of $\Phi_{\mathcal{B}}^X$ and $\Phi_{\mathcal{B}}^Y$ (feature

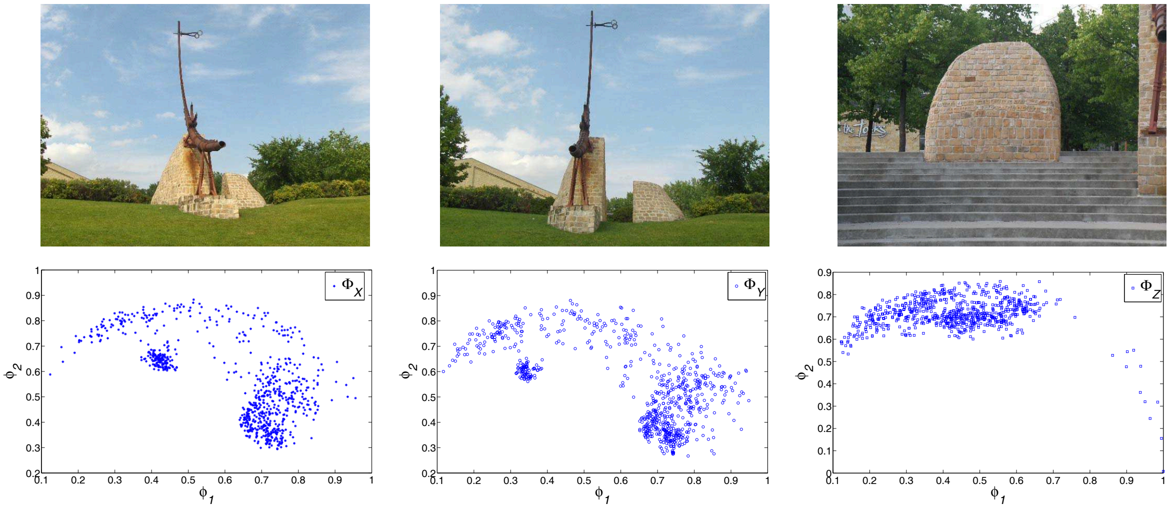


Figure 1.7: An example of 3 images and the corresponding describable objects in a 2D feature space

values), respectively. Then,

$$gMD(X,Y) = \sqrt{(\bar{\Phi}_{\mathcal{B}}^{X} - \bar{\Phi}_{\mathcal{B}}^{Y})^{T}(\Sigma_{X,Y})^{-1}(\bar{\Phi}_{\mathcal{B}}^{X} - \bar{\Phi}_{\mathcal{B}}^{Y})}, \tag{1.12}$$

$$gMNM(X,Y) = \frac{1}{1 + gMD(X,Y)}, \tag{1.13}$$

where

$$\Sigma_{X,Y} = \frac{1}{2}\left(\Sigma_X + \Sigma_Y\right). \tag{1.14}$$

Example 1.2. *Pairs of images in Fig. 1.1 and 1.1 are considered here again. Figure 1.7 shows distribution of subimages in the feature space, where only two probe functions corresponding to average gray level ($\phi_1(.)$) and entropy ($\phi_2(.)$) have been considered. Here, $\bar{\Phi}_{\mathcal{B}}^{X} = [0.6186 \;\; 0.5586]^T$, $\bar{\Phi}_{\mathcal{B}}^{Y} = [0.6161 \;\; 0.5426]$ and $\bar{\Phi}_{\mathcal{B}}^{Z} = [0.4298 \;\; 0.7125]^T$.*

Furthermore, we have

$$\Sigma_X = E[\left(\Phi_{\mathcal{B}}^X - \bar{\Phi}_{\mathcal{B}}^X\right)\left(\Phi_{\mathcal{B}}^X - \bar{\Phi}_{\mathcal{B}}^X\right)^T] = \begin{bmatrix} 0.0289 & -0.0170 \\ -0.0170 & 0.0250 \end{bmatrix}$$

$$\Sigma_Y = E[\left(\Phi_{\mathcal{B}}^Y - \bar{\Phi}_{\mathcal{B}}^Y\right)\left(\Phi_{\mathcal{B}}^Y - \bar{\Phi}_{\mathcal{B}}^Y\right)^T] = \begin{bmatrix} 0.0409 & -0.0200 \\ -0.0200 & 0.0249 \end{bmatrix}$$

$$\Sigma_Z = E[\left(\Phi_{\mathcal{B}}^Z - \bar{\Phi}_{\mathcal{B}}^Z\right)\left(\Phi_{\mathcal{B}}^Z - \bar{\Phi}_{\mathcal{B}}^Z\right)^T] = \begin{bmatrix} 0.0216 & -0.0004 \\ -0.0004 & 0.0053 \end{bmatrix}$$

$$\Sigma_{X,Y} = \frac{1}{2}\left(\Sigma_X + \Sigma_Y\right) = \begin{bmatrix} 0.0349 & -0.0185 \\ -0.0185 & 0.0250 \end{bmatrix}$$

$$\Sigma_{X,Z} = \frac{1}{2}\left(\Sigma_X + \Sigma_Z\right) = \begin{bmatrix} 0.0253 & -0.0087 \\ -0.0087 & 0.0152 \end{bmatrix}$$

$$gMD(X,Y) = 0.0197 \quad gMD(X,Z) = 2.0608$$

This means the distance between X and Z is significantly greater than distance between X and Y.

1.4.3 Hausdorff distance based nearness metric (HdNM)

Hausdorff distance by definition is defined between two finite point sets in a metric space. Assume $d(x,y)$ is a distance function defined between points x and y in a metric space. Let X and Y be sets of points in the space. Hausdorff distance $\rho_H(X,Y)$ between sets X and

Y [16] is defined, as in (1.15).

$$\rho_H(X, Y) = \max\{d_H(X, Y), d_H(Y, X)\}, \tag{1.15}$$

where

$$d_H(X, Y) = \max_{x \in X}\{\min_{y \in Y}\{d(x, y)\}\}, \tag{1.16}$$

$$d_H(Y, X) = \max_{y \in Y}\{\min_{x \in X}\{d(x, y)\}\}. \tag{1.17}$$

$d_H(X, Y)$ and $d_H(Y, X)$ are directed Hausdorff distances from X to Y and from Y to X, respectively. It can be proved that if d is a metric, then ρ_H is a metric as well [16].

Hausdorff distance measure has been used extensively in image comparison for template matching problem [20, 28, 57, 81], where the goal is to find a part of the test image that matches a given template image. In such problems, Hausdorff distance is defined in the spatial domain between the points on the edges in the template image and those of the test image. To the best of the author's knowledge, most of the applications of the Hausdorff distance in image comparison problems has been limited to the template matching. In one case (reference [47]), however, a different approach is taken and a new distance is introduced that is called a perceptually modified Hausdorff distance (PMHD). PMHD is basically different with Hausdorff distance in the sense that the maximum operation in the calculation of directed Hausdorff distances (Equations 1.16 and 1.17) is replaced by a weighted average operation. Moreover, PMHD is defined between statistical signatures of the color features after clustering.

In this thesis, however, the Hausdorff distance is used to define nearness between sets of feature vectors of describable objects. The proposed *Hausdorff distance-based nearness measure* $HdNM$ and its corresponding Hausdorff distance measure HDM is defined here as follows. Let $\mathcal{B}$ be a set of probe functions and let $X, Y \in O$ be sets of describable object. Let $(\Phi_{\mathcal{B}}^{O}, d)$ be a metric space where d is a distance function between feature vectors in $\Phi_{\mathcal{B}}^{O}$. An example of d can be the L_1-norm based or Manhattan distance defined as follows,

$$\forall x, y \in O \quad d(\vec{\phi}(x), \vec{\phi}(y)) = \| \vec{\phi}_{\mathcal{B}}(x) - \vec{\phi}_{\mathcal{B}}(y) \|_1 = \sum_{k=1}^{l} |\phi_k(x) - \phi_k(y)|. \tag{1.18}$$

The Hausdorff distance is then defined using Equations 1.15 to 1.17 and converted to nearness measure ($HdNM$), using Equation 1.20.

$$HDM(X, Y) = \rho_H(\Phi_{\mathcal{B}}^{X}, \Phi_{\mathcal{B}}^{Y}) \tag{1.19}$$

$$HdNM(X, Y) = \frac{1}{1 + HDM(X, Y)}. \tag{1.20}$$

Chapter 2

Tolerance Spaces and Perception of Similarity

Describable objects are identified and recognized by their description. In the previous chapter, it was shown how a mathematical description of a describable object is possible through the use of a feature vector that is created using a set of probe functions. Therefore, the first natural approach to defining similarity between objects is to say that, objects are similar if they have the same (equal) description. This simple and yet fundamental description of similarity is the basis of the near set theory as developed by James Peters in 2006 (See *e.g.* [48, 49, 51]). If two or more describable objects have the same description in terms of their feature vectors, they are *indiscernible* and therefore shall be classified as *similar* with a high level of certainty. This will create an *equivalence relation* which is named *indiscernibility relation* in Near Set and Rough Set literature. An indiscernibility relation partitions the set of objects into *classes* of objects that are indiscernible with respect to their description.

In a more realistic situation, however, if the descriptions are close enough, objects can be considered *almost similar* and thus allowing a small level of error, defined by a *tolerance relation*. Formalizing the concept of similarity between objects in terms of a tolerance relation is highly appealing to intuition. Sossinsky, in his paper in 1986 [65], points to this issue and explains why a tolerance relation is a suitable model to represent similarity both from a mathematical and philosophical point of view. After some formal definitions, it is shown here how equivalence and tolerance relations can be used to describe similarity.

2.1 Descriptive-based Equivalence and Tolerance Relation

In general, an equivalence relation is defined as follows. Note that equivalence relation is different from *equality*. However, the later is a special case of the former.

Definition 2.1. Equivalence Relation
A binary relation $\sim$ *defined on a set* O *is an equivalence relation iff* $\sim: O \times O \to \{1, 0\}$ *and* $\sim$ *has the following properties:*

1. *Reflexivity:* $\forall x \in O \quad x \sim x$
2. *Symmetry:* $\forall x, y \in O \quad x \sim y \Rightarrow y \sim x$
3. *Transitivity:* $\forall x, y, z \in O \quad x \sim y$ *and* $y \sim z \Rightarrow x \sim z$.

Note: A binary relation R on a set O can be defined either as a subset of $O \times O$ or as a mapping from $O \times O$ into the set $\{0, 1\}$. We will use the latter method to extend the definition into fuzzy relations. Therefore, the notations $(x, y) \in \sim$, $\sim (x, y) = 1$, and $x \sim y$ represent the same concept.

Definition 2.2. Descriptive Indiscernibility Relation

A descriptive indiscernibility relation on a set O of describable objects with respect to probe functions in $\mathcal{B}$, is an equivalence relation shown with $\sim_{\mathcal{B}}$ and is defined as follows

$$\sim_{\mathcal{B}} \subseteq O \times O \quad \sim_{\mathcal{B}} = \{(x, y) \in O \mid \vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y)\}. \tag{2.1}$$

Definition 2.3. Equivalence Class

Let $\sim_{\mathcal{B}}$ be an equivalence relation (here, a descriptive indiscernibility relation) defined on a set O. For every element (describable object) $x \in O$, the set of all the elements in O that are indiscernible with x is named the equivalence class associated with x *and is shown as $x_{/\sim_{\mathcal{B}}}$.*

$$x_{/\sim_{\mathcal{B}}} = \{y \in O \mid x \sim y\} \tag{2.2}$$

Corollary 2.1. *The equivalence relation defined on a set O, partitions the set through equivalence classes. The set of all equivalence classes is shown with $O_{/\sim_{\mathcal{B}}}$.*

$$O_{/\sim_{\mathcal{B}}} = \{x_{/\sim_{\mathcal{B}}} \mid \; x \in O\}$$
$$\bigcup_{x \in O} x_{/\sim_{\mathcal{B}}} = O$$
$$\forall x, y \in O \quad (x_{/\sim_{\mathcal{B}}}) \cap (y_{/\sim_{\mathcal{B}}}) = \emptyset$$

Definition 2.4. Tolerance Relation

A binary relation $\cong$ defined on a set O is a tolerance relation iff $\cong: O \times O \to \{1, 0\}$ and $\cong$ has reflexivity and symmetry properties but transitivity is not required.

1. *Reflexivity:* $\forall x \in O \quad x \cong x$

2. *Symmetry:* $\forall x, y \in O \quad x \cong y \Rightarrow y \cong x$

where $x \cong y$ *means* $\cong (x, y) = 1$ *or equivalently* $(x, y) \in \cong$.

Definition 2.5. Tolerance Space

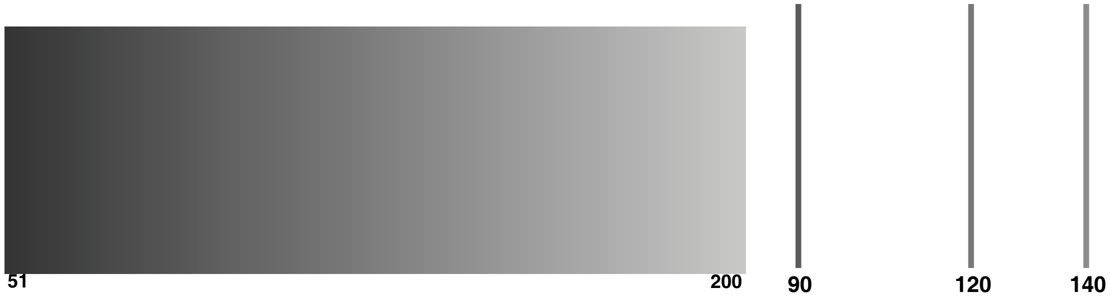
The set O *along with a tolerance relation* $\cong$ *defined on* O *is named a tolerance space and is shown with* $(O, \cong)$.

Why tolerance relation?

Although feature vectors can represent the sensory input of a stimuli, the relation between sensory input and the resulting induced perception in the mind is not known. Human perception does not require two objects to have exactly identical feature vectors to consider them similar. Sensory inputs that are close enough in the feature space, may induce identical perception although they are actually different. Tolerance space theory can be used to model such *approximate* equalities. Figure 2.1(a) consists of 150 vertical bars (bands) of 10 pixels width having a gray level values ranging from 51 in the left side up to 200 in the right side. Figure 2.1(b) for example shows three different individual bands with gray level values of 90, 120 and 140. Although human eye can distinguish between all the 255 different gray levels if looking from a close enough distance, when we are looking at adjacent gray levels we can safely assume they are approximately similar in color. Descriptive tolerance relation is defined as follows to model such approximation.

Definition 2.6. Descriptive-based Distance Tolerance Relation

Let O *be a set of describable objects and let* $\mathcal{B}$ *be a set of probe functions and* $\varepsilon \in \mathbb{R}$. *Let* d *be a distance function such that* $(\Phi^{O}_{\mathcal{B}}, d)$ *is a metric space where* $\Phi^{O}_{\mathcal{B}}$ *is the set of feature vectors corresponding to elements of* O. *The descriptive-based tolerance relation* $\cong_{\mathcal{B}}$ *on* O



(a) gray level spectrum

(b) individual gray levels

Figure 2.1: A collection of different vertical bands with different gray values

is defined as follows:

$$\cong_{\mathcal{B},\varepsilon} \subseteq O \times O \qquad \cong_{\mathcal{B},\varepsilon} = \{(x,y) \in O \times O \mid d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon\}, \tag{2.3}$$

or alternatively

$$\cong_{\mathcal{B},\varepsilon} : O \times O \to \{0,1\} \qquad \cong_{\mathcal{B},\varepsilon} (x,y) = 1 \iff d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon. \tag{2.4}$$

According to the definition of tolerance relation, the relation that is defined in Equation 2.3 or 2.4 is a tolerance relation and hence is named descriptive-based tolerance relation.

In other words, two describable objects are *almost similar* to each other with respect to descriptions given in $\mathcal{B}$, if and only if their corresponding feature vectors with respect to $\mathcal{B}$ have a distance in the feature space which is smaller than ε. Table 2.1 summarizes the two types of similarity between describable objects defined here.

Definition 2.7. Descriptive-based Tolerance Neighborhood

Let $(O, \cong_{\mathcal{B},\varepsilon})$ *be a tolerance space and* $(\Phi^{O}_{\mathcal{B}}, d)$ *be a metric space where* $\mathcal{B}$ *is a set of probe*

Table 2.1: Level of precision in distance based element similarity

Type of similarity	Relation	Notation	Condition
Exact (precise) similarity	Indiscernibility	$\sim_{\mathcal{B}}$	$\vec{\phi}_{\mathcal{B}}(x) = \vec{\phi}_{\mathcal{B}}(y)$
Almost similarity	Tolerance	$\cong_{\mathcal{B},\varepsilon}$	$d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon$

functions. For every element $x \in O$, the descriptive-based tolerance neighborhood of x is shown with $n_O^{\cong_{\mathcal{B},\varepsilon}}(x)$ and defined as follows:

$$n_O^{\cong_{\mathcal{B},\varepsilon}}(x) = \{y \in O \mid d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon\}. \tag{2.5}$$

The set of all tolerance neighborhoods in a tolerance space is shown here with $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$.

$$\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}} = \{n_O^{\cong_{\mathcal{B},\varepsilon}}(x) \mid x \in O\} \tag{2.6}$$

Proposition 2.1. $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$ *is a covering of the set O. Tolerance neighborhoods may have overlap but the union of all tolerance neighborhoods is equal to O. Also, some tolerance neighborhoods may be equal to each other and hence*

$$|\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}| \leq |O|. \tag{2.7}$$

Definition 2.8. Tolerance Pre-class

Let $(O, \cong_{\mathcal{B},\varepsilon})$ be a tolerance space. The set $A \subset O$ is named a pre-class *iff $\forall x, y \in A \Rightarrow x \cong_{\mathcal{B},\varepsilon} y$.*

Definition 2.9. Tolerance Class (Maximal Pre-class)

A maximal pre-class (with respect to inclusion) is named a tolerance class. *Hence, if $(O, \cong_{\mathcal{B},\varepsilon}$*

$)$ *is a tolerance space, the set* $C \subseteq O$ *is a* tolerance class *iff:*

1. $\forall x, y \in C \Rightarrow x \cong_{\mathcal{B},\varepsilon} y$.

2. $\nexists z \in O \mid C \cup \{z\}$ *is a pre-class.*

The set of all tolerance classes of the tolerance space $(O, \cong_{\mathcal{B},\varepsilon})$ *is shown here as* $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$.

Proposition 2.2. $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ *is a covering of the set* O*. Tolerance classes may have overlap but the union of all tolerance classes is equal to* O*. Therefore,*

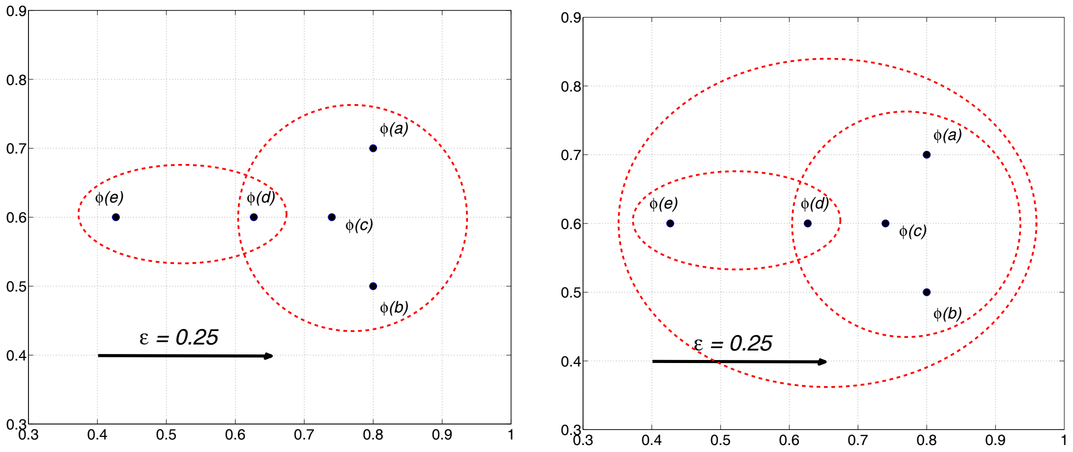
$$|\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}| \leq |O|. \tag{2.8}$$

Example 2.1. *Suppose* $O = \{a, b, c, d, e\}$ *is a set of describable objects with 2 dimensional feature vectors given in set* $\Phi_{\mathcal{B}}^{O} \subset \mathbb{R}^2$*. Let* $(O, \cong_{\mathcal{B},\varepsilon})$ *be a tolerance space defined based on the metric space* $(\Phi_{\mathcal{B}}^{O}, d)$ *where* d *is the Euclidean distance between feature vectors and let* $\varepsilon = 0.25$*. The set of feature vectors will be* $\Phi_{\mathcal{B}}^{O} = \{\vec{\phi}_{\mathcal{B}}(a), \vec{\phi}_{\mathcal{B}}(b), \vec{\phi}_{\mathcal{B}}(c), \vec{\phi}_{\mathcal{B}}(d), \vec{\phi}_{\mathcal{B}}(e)\}$ *and is shown in Fig. 2.2. The tolerance neighborhoods and the tolerance classes of* $(O, \cong_{\mathcal{B},\varepsilon})$ *are listed as follows*

$$\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}} = \{\{a, b, c, d\}, \{a, b, c, d, e\}, \{d, e\}\}$$

$$\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}} = \{\{a, b, c, d\}, \{d, e\}\}$$

This follows from the fact that $n_O^{\cong_{\mathcal{B},\varepsilon}}(a) = n_O^{\cong_{\mathcal{B},\varepsilon}}(b) = n_O^{\cong_{\mathcal{B},\varepsilon}}(c) = \{a, b, c, d\}$*,* $n_O^{\cong_{\mathcal{B},\varepsilon}}(d) = \{a, b, c, d, e\}$ *and* $n_O^{\cong_{\mathcal{B},\varepsilon}}(e) = \{d, e\}$*. Figure 2.2 shows the set of tolerance classes circled (left) and the set of tolerance neighborhoods circled (right). Both* $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$ *and* $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ *cover the set* O *of objects.*



(a) Tolerance classes

(b) Tolerance neighborhoods

Figure 2.2: Tolerance classes and tolerance neighborhoods for an example 2D feature space with only 5 points

Example 2.2. Covering versus partitioning *Figure 2.3 shows a set of 8 describable objects. Each object is a disk where the only feature of interest is the radius of the disk. Part (A) shows how the objects have been divided into 3 clusters based on their description(size). The resulting clusters are disjoint sets that partition the set of elements. Part (B) shows the set of tolerance neighborhoods where the centre element of each neighborhood is marked with black circle.* $\varepsilon = 0.21$ *and a disk belongs to a tolerance neighborhood if the difference between the radius is smaller than* ε. *Part (C) shows the set of tolerance classes in the same tolerance space. It can be seen that neighborhoods and classes can have nonempty intersections and hence provide a covering of the set of elements (not a partitioning).*

Note: 1

In most practical applications, the number of elements is relatively high such that finding all the maximal pre-classes is not feasible by an exhaustive search in the set of subsets. Special algorithms are needed to find all the tolerance classes. This is a topic that has not been

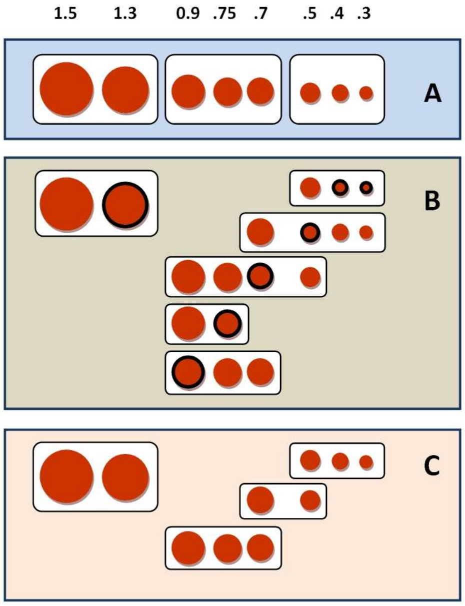


Figure 2.3: An illustrated example of a set of objects (disks) described by their size. (A) partitioning into 3 different clusters, (B) sets of tolerance neighborhoods and (C) sets of tolerance classes. ($\varepsilon = 2.01$)

studied extensively in the literature. In few cases, where an algorithm for finding tolerance classes has been presented, it is either given for special cases [36] or is not able to find *all* the tolerance classes [25] and only finds *a* set of classes that cover the set. In this thesis, an algorithm is proposed to find tolerance classes (Algorithm 1). Figure 2.4 shows an example of a set of points in 2D euclidean space and the tolerance classes that have been identified using this algorithm. The algorithm cannot find all the tolerance classes. However, it finds a set of tolerance classes that can cover the set. In any case, calculation of tolerance classes is extremely computationally expensive. The nearness measures in this thesis are based on tolerance neighborhoods not tolerance classes.

Note 2: In [6], the name tolerance-*class* has been used to denote a tolerance neighborhood (see [6, 67]) and the name tolerance-*block* is given to a maximal preclass (what is actually known as tolerance class). This is contrary to the more commonly accepted terminology in tolerance space theory. In this thesis, tolerance classes and neighborhoods are defined in the classical way and the term *tolerance block* can be used such that it can refer to any of the above.

Algorithm 1: Tolerance class calculation

```
Input  : 1. The set of visual elements $O = \{x_1, x_2, ..., x_N\}$.
         2. A threshold value $\varepsilon$
         3. A distance function $d_\phi$ between feature vectors of elements in $O$.
Output: A set of maximal pre-classes $H = \{T_1, T_2, ...\}$ that cover the set $O$.
initialization: H = ∅ (Start with empty set of pre-classes)
for $i = 1 : N - 1$ do
    for $j = i : N$ do
        if $d_\phi(\vec{x}_i, \vec{x}_j) < \varepsilon$ then
            $H = H \cup \{\{x_i, x_j\}\}$ ;          // Consider $\{x_i, x_j\}$ as a pre-class
        end
    end
end
Level initialization : $L = 2$;
while ($\exists T \in H$ such that $|T| = L$) $\& L < |H|$ do
    for $T_k$ in $H$ do
        // Look in other elements of $O$
        $C = O - T_k$ ;                              // which are not already in $T_k$
        initialize: $new \leftarrow 0$; ;             // No new element is found yet
        initialize: $j \leftarrow 1$; ;         // start with the first element of $C$
        while $new = 0$ do
            $allclear \leftarrow 1, x_j \in C$
            for $x_t \in T_k$ do
                if $d(x_j, x_t) \geq \varepsilon$ then
                    $allclear \leftarrow 0$
                end
            end
            if allclear = 1 then
                $T_k = T_k \cup x_j$ ;         // Add this element to the pre-class
            end
        end
    end
    $L \leftarrow L + 1$
end
```

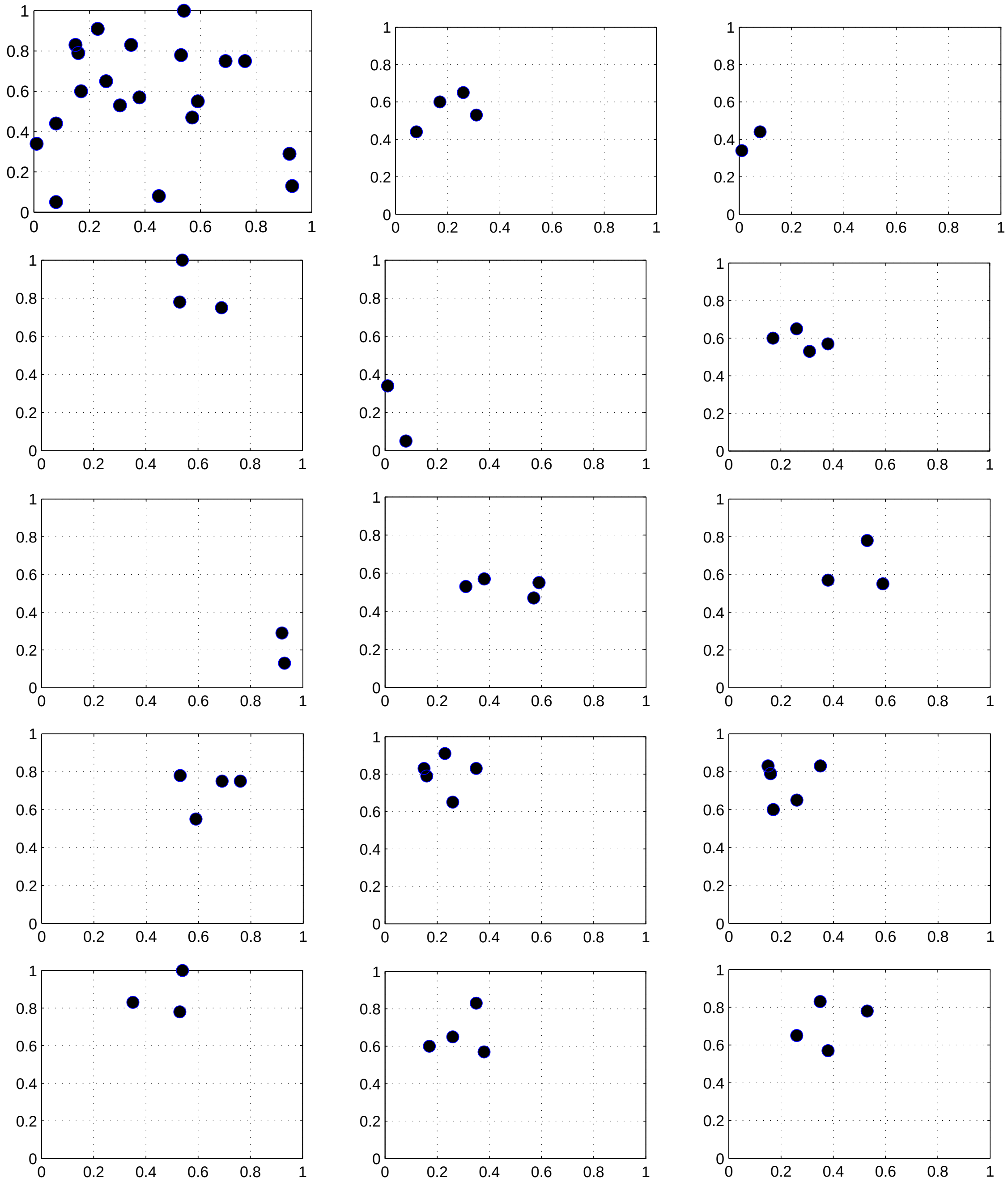


Figure 2.4: An example 2-D feature space (top left) and the corresponding tolerance classes calculated using the algorithm in 1 where $\varepsilon = 0.3$

2.2 New Method: Tolerance Nearness between Sets of Image Visual Elements

In section 1.3.2, it was shown how to define a measure of dissimilarity between describable objects using a distance metric in the feature space. This distance is named here as *elementary distance* and it can be considered as the building block of an image similarity measure. Each image is considered as a set of describable objects (visual elements) and the overal similarity between images is defined as the nearness between sets of visual elements. This is a *Near Sets* approach in which the nearness between sets is defined based on the nearness between elements of the sets [51]. Let $\mathcal{I}$ be a set of images and let $\mathcal{X}, \mathcal{Y} \in \mathcal{I}$ be pairs of images. Let X, Y be sets of visual elements of the images, respectively. A distance measure (DM) and similarity measure (SM) between any two images is then defined as the distance D and nearness N between the sets of corresponding describable objects.

$$DM : \mathcal{I} \times \mathcal{I} \rightarrow [0, \infty) \quad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{I},\ DM(\mathcal{X}, \mathcal{Y}) = D(X, Y) \tag{2.9}$$

$$SM : \mathcal{I} \times \mathcal{I} \rightarrow [0, 1] \quad \forall \mathcal{X}, \mathcal{Y} \in \mathcal{I},\ SM(\mathcal{X}, \mathcal{Y}) = N(X, Y) \tag{2.10}$$

A near set approach by itself just reduces the image similarity problem to the problem of defining nearness between sets of visual elements. This may become possible in different ways. In this thesis, a tolerance space approach as well as a fuzzy tolerance space approach are introduced. In a tolerance space view to image correspondence, nearness between sets of describable objects X, Y is defined by comparing the tolerance blocks (*i.e.* neighborhoods or classes) of *almost similar* objects in a tolerance space that covers both images. Let X and

Y be sets of describable objects and let $O = X \cup Y$. Let $\mathcal{B}$ be a set of probe functions corresponding to visual descriptions of describable objects. Let $(O, \cong_{\mathcal{B},\varepsilon})$ be a tolerance space where $\cong_{\mathcal{B},\varepsilon}$ is defined in Equation 2.3. $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$ is the family of all tolerance neighborhoods and $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ is the family of all tolerance classes in the tolerance space $(O, \cong_{\mathcal{B},\varepsilon})$.

Figure 2.5 is an example of two images divided into visual elements (non-overlapping square subimages of size 20 pixels are considered as visual elements). $\mathcal{B}$ contains 6 probe functions that extract RGB and HSV color components of each visual element. All the feature values are normalized between 0 and 1 and $\varepsilon = 0.8$. Each A_i represents one of the tolerance neighborhoods in $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$ demonstrated in a separate plane in 3D. As one expects, each tolerance neighborhood represents areas of images that have almost the same color. In the next section, it is shown how these tolerance neighborhoods can be used to define similarity.

2.2.1 Neighborhood distances and tolerance covering nearness measure (tcNM)

Tolerance neighborhoods (and tolerance classes) can be used as building blocks of a nearness measure to define similarity between sets of describable objects (visual elements). The main idea behind using tolerance neighborhoods is the conjecture that when we look at two images, we tend to group image elements together based on similarity to the element of interest at the point of gaze. This conjecture can be interpreted as *the principle of similarity* in *Gestalt's theory of perceptual organization* in psychology. [1] This principle states that

[1] Gestalt's principle of organizations is a descriptive theory about how people perceive visual components as organized patterns. This theory is part of a school of thought in psychology.

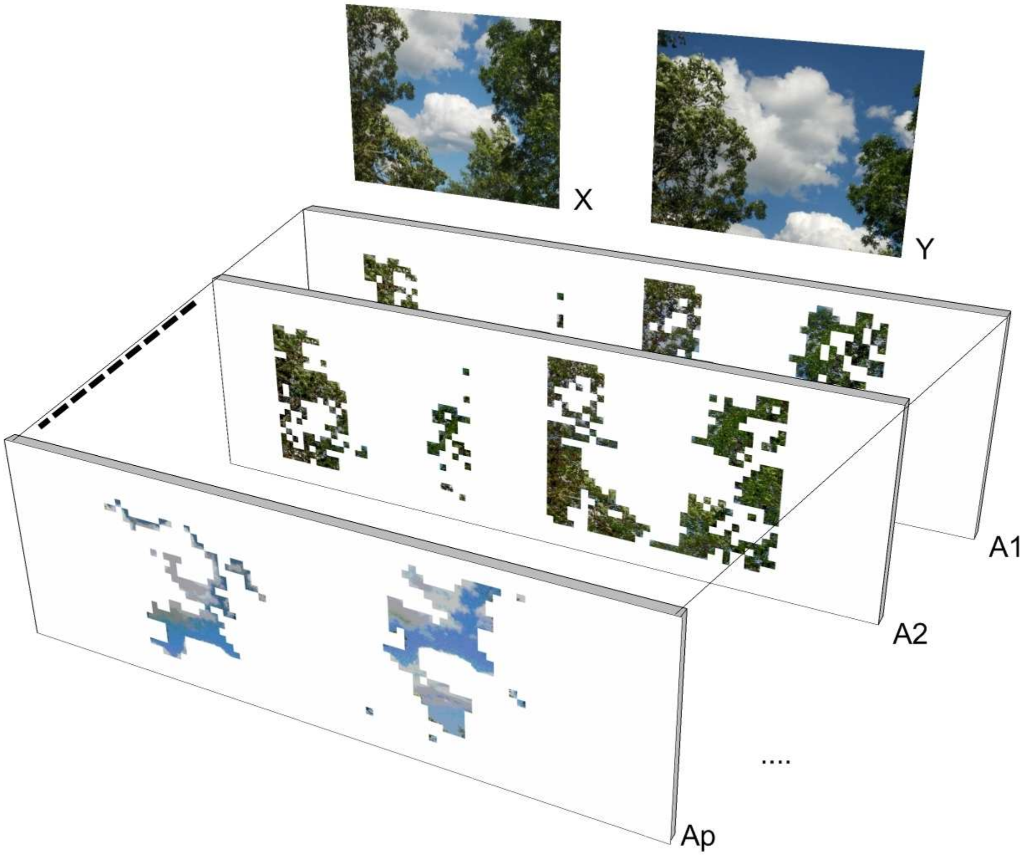


Figure 2.5: Tolerance neighborhoods

things with similar visual characteristics such as shape, size, color or texture, will be seen as belonging together as a group. A suitable mathematical model to describe this group of visually similar elements can be the tolerance neighborhood of elements around the point of gaze. The similarity is then defined here by comparing how these neighborhoods cover images. Figure 2.6 for example shows a sample neighborhood A in a pair of images X and Y. A contains subimages from both X and Y. Part of the neighborhood that belongs to X and Y is shown in the figure and denoted by $A \cap X$ and $A \cap Y$, respectively. The difference between the cardinality of the set $A \cap X$ and the set $A \cap Y$ is used as a measure of difference between the two parts. This difference can be calculated and averaged among all the neighborhoods in the tolerance space to define an overall distance measure between X and Y named here as *tolerance covering distance measure* ($tcDM$). The steps involved in defining $tcDM$ between two images are listed as follows,

- Step 1: Divide both images into sets of visual elements X and Y (visually describable subimages).

- Step 2: Define a set of probe functions $\mathcal{B} = \{\phi_1, \phi_2, ...\phi_l\}$ that can extract some visual features of subimages. Each subimage x can then be represented with its feature vector $\vec{\phi}_{\mathcal{B}}(x) = [\phi_1(x) \;\; \phi_2(x) \;\; ... \;\; \phi_l(x)]^T$.

- Step 3: Define a distance function d between feature vectors in the feature space. L_1-norm-based distance function (Manhattan distance) and Euclidean distance are examples of such distances.

- Step 4: Define a tolerance relation $\cong_{\mathcal{B},\varepsilon}$ between feature vectors based on a tolerance level of error ε to represent similarity in the subimage level. Two visual elements x and y are similar to each other if the above distance between feature vectors $d(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y))$ is smaller than the tolerable level of error threshold, ε.

- Step 5: For each visual element x_0 in the union of all subimages ($x_0 \in X \cup Y$), find the tolerance neighborhood $n_{X \cup Y}(x_0)$ with respect to the tolerance relation $\cong_{\mathcal{B},\varepsilon}$.

- Step 6: For each tolerance neighborhood, define the *neighborhood distance* between

X and Y with respect to x_0 as in Equation 2.11.

$$T_{X,Y}(x_0) = \left| \frac{|n(x_0) \cap X| \; - \; |n(x_0) \cap Y|}{|n(x_0) \cap X| \; + \; |n(x_0) \cap Y|} \right| \tag{2.11}$$

$T_{X,Y}(x_0)$ represents the normalized difference between the size of $n(x_0) \cap X$ and $n(x_0) \cap Y$.

In section 2.3 and using a probabilistic approach, it is shown why $T_{X,Y}(x_0)$ can be used as a measure of dissimilarity between images.

- Step 7: *tolerance covering distance measure* ($tcDM$) is defined in the following equation as an average of the neighborhood distances calculated for all the visual elements of both images. $tcDM$ is a real number between 0 and 1. zero distance represents maximum similarity (equality) and the higher values of distance represent less similarity.

$$tcDM(X,Y) = \frac{1}{|\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}|} \sum_{n(x_0) \in O} T_{X,Y}(x_0) \tag{2.12}$$

where $|\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}|$ is the total number of tolerance neighborhoods in the tolerance space.

- Step 8: Subsequently, $tcNM$ similarity measure is defined as follows,

$$tcNM = 1 - \sqrt{tcDM}. \tag{2.13}$$

Both $tcDM$ and $tcNM$ can be used to compare images. The former is a measure of *dissimilarity* where lower values represent higher similarity. The later is a measure of *similarity* where higher values represent higher similarity. $tcNM$ is a real number between 0 and 1.

The idea of tolerance covering distance measure was inspired by another tolerance space based method namely *tolerance nearness measure* tNM. tNM was first proposed by Henry and Peters and appeared in [24,51]. tNM is defined based on tolerance classes of a tolerance space as follows,

Let $O = X \cup Y$ and let $(O, \cong_{\mathcal{B},\varepsilon})$ be a tolerance space and let $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ be the family of all tolerance classes in O. [2]

$$tNM_{\cong_{\mathcal{B},\varepsilon}}(X,Y) = \frac{\displaystyle\sum_{A \in \mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}} \mathcal{T} \cdot |A|}{\displaystyle\sum_{A \in \mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}} |A|}, \tag{2.14}$$

$$\mathcal{T} = \frac{\min\{\,|A \cap X|, |A \cap Y|\,\}}{\max\{\,|A \cap X|, |A \cap Y|\,\}}. \tag{2.15}$$

where $|A \cap X|$ and $|A \cap Y|$ are part of each tolerance class that are in X and Y, respectively, and $\mathcal{T}$ is a measure of difference between their sizes. As one expects, tNM value is between 0 and 1. In this thesis, a faster version of tNM is implemented that uses tolerance neighborhoods ($\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$) rather than tolerance classes ($\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$). The significance of $tcDM$ (and $tcNM$) compared to tNM can be summarized as follows,

1. tNM is based on tolerance classes in a tolerance space whereas $tcNM$ is based on tolerance neighborhoods. Algorithms for finding tolerance classes such as the one introduced here in Algorithm 1 or the method introduced in [25] have much more

[2] The actual set of tolerance classes which have been used in [25] is not $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ but a subset of $\mathsf{H}_O^{\cong_{\mathcal{B},\varepsilon}}$ that still covers the set O.

computational complexity than finding the neighborhoods. Also, they cannot find all the existing tolerance classes.

2. The performance of $tcDM$ in measuring similarity is shown both theoretically and experimentally in this thesis. A detailed probabilistic approach on describing the correspondence between distribution of feature values in images and their tolerance covering distance measures is derived and validated in Section 2.3.

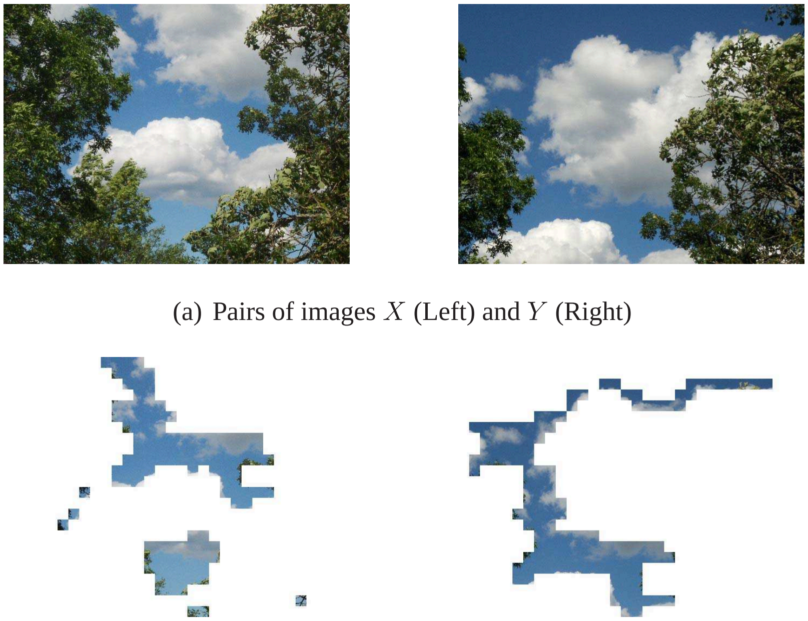
3. A complete side by side comparison of $tcDM$ with the existing methods in content based image retrieval is provided for $tcDM$ (Chapter 5 of this thesis).

4. The generalization of tolerance based methods to fuzzy tolerance based methods is introduced for $tcDM$.

2.2.2 Choosing the value of tolerance threshold ε

The physical and mathematical reason for using a tolerance relation to model the limited acuity and tolerance of human visual perception is intuitive and simple. However, choosing a proper value for the threshold level ε may not seem very obvious and may depend on the situation.

In this section a systematic method for choosing the epsilon value is presented based on the statistical distribution of the distances between the visual elements in a given query image. The method is based on the idea that ε should not be large enough such that most of the pairs of elements fall within ε distance of each other. Moreover, it should not be very small such that most of the pairs of elements are not within ε distance of each other. This can



(a) Pairs of images X (Left) and Y (Right)

(b) A tolerance neighborhood A, $A \cap X$ (Left) and $A \cap Y$ (Right)

Figure 2.6: Sample pair of images X, Y and an example of a tolerance neighborhood that covers part of each image. Here $|A \cap X| = 122$ and $|A \cap Y| = 132$.

be shown using the distribution function (cdf) of the distance between pairs of elements in an image. Figure 2.7(b) shows an empirical pdf (histogram) of the distances between elements of a sample image where the average RGB color components of the visual elements are considered as probe functions (feature values). Fig. 2.7(c) shows distribution of the size of the resulting tolerance neighborhoods in the image, calculated for different values of ε. It can be seen that very small (or very large) values of ε will cause the tolerance neighborhoods to be limited to single elements (or cover the whole image), respectively.

Let D represent a distance metric between visual elements. Considering D as a random variable for an unknown pair of elements, $F_D(d)$ is used to represent the cumulative distribution of D.

$$F_D(d) = Pr(D \leq d), \quad d \in [0, +\infty).$$

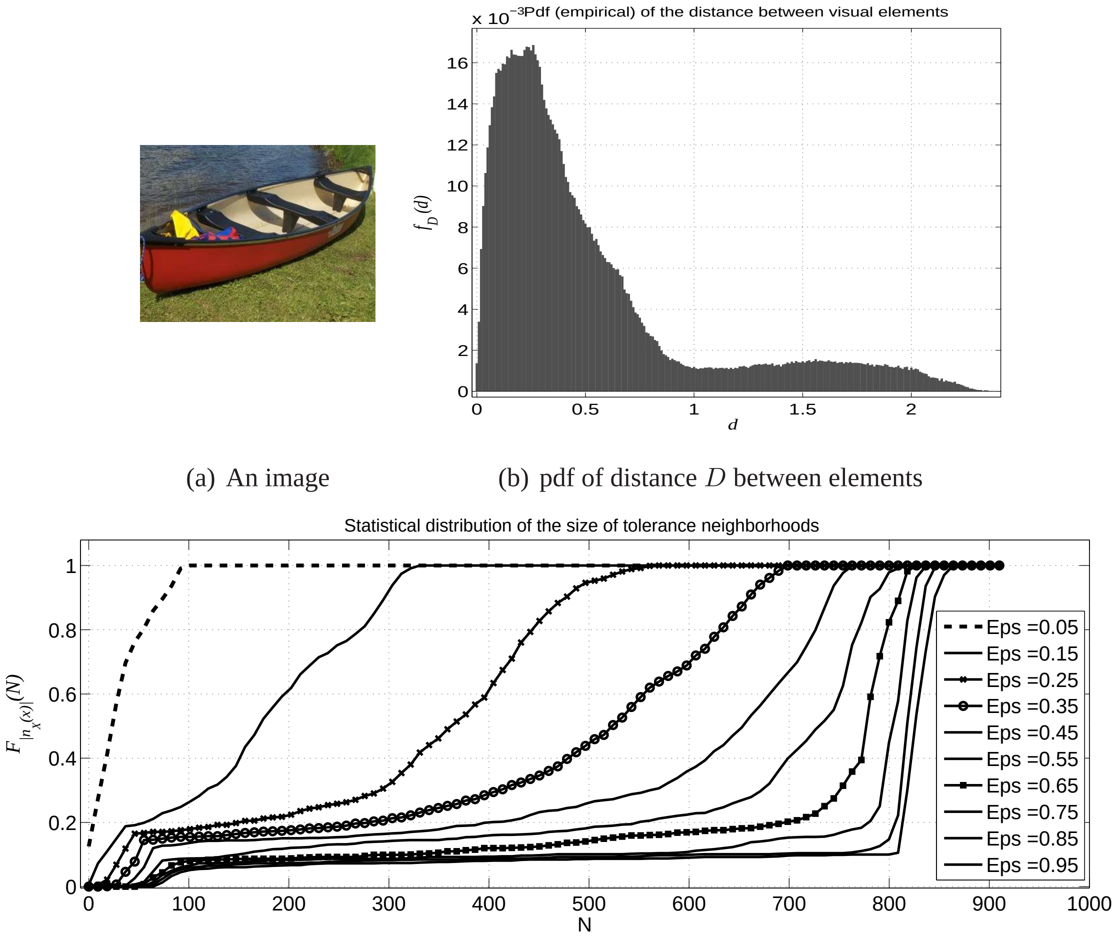
Here ε_τ is defined as the distance value at which F_D equals to τ

$$F_D(\varepsilon_\tau) = \tau, \quad \tau \in [0, 1]. \tag{2.16}$$

For a fixed and arbitrary chosen value of τ, ε_τ can be calculated and adaptively chosen as ε for each query image in any image comparison. Although the method still depends on a parameter (τ), the results are much less sensitive to the selection of τ rather than ε. This method can be used for calculation of both tolerance neighborhoods and tolerance classes. The method for choosing ε can be summarized as follows,

- Take the query image and divide it into visual elements (subimages) $X = \{x_1, x_2, ..., x_N\}$.
- For each subimage x_k, calculate the feature vector $\vec{\phi}_{\mathcal{B}}(x_k)$ using the given probe functions.
- Construct a distance matrix D by finding the distance between all the pairs of feature vectors in the image. $D(k, j) = d(\vec{\phi}_{\mathcal{B}}(x_k), \vec{\phi}_{\mathcal{B}}(x_j))$
- Count the number M of pairs of elements (x, y) where the distance D between their feature vectors is smaller than the threshold level τ.
- For each query image X, the value of $\varepsilon = \varepsilon_\tau(X)$ is calculated using the following formula

$$\varepsilon_\tau(X) = \frac{M}{N^2} \tag{2.17}$$



(a) An image

(b) pdf of distance D between elements

(c) cdf of the size of tolerance neighborhoods

Figure 2.7: An image (a), distribution of distances between pairs of elements (b), and distribution of the size of tolerance neighborhoods at different values of ε (c)

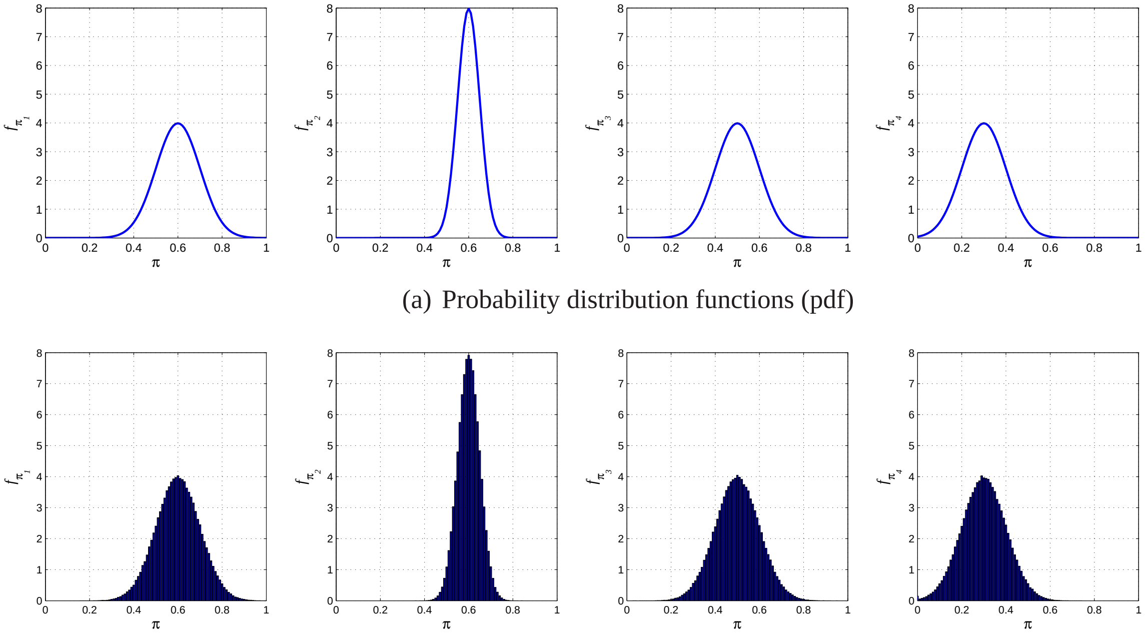
2.3 Mathematical basis for the new measure

In this section, a theoretical basis for the proposed method is introduced using a probabilistic approach to image description. Images are described as sets of visual elements with visual features that are considered as random variables with known probability density functions. The probability density function of the neighborhood distance is then defined based on densities of the feature values.

Let $\mathcal{I}$ be a set of images and suppose $\mathcal{V}$ is the set of all visual elements in images in $\mathcal{I}$. Let $\mathcal{B} = \{\phi_1, \phi_2, ...\phi_l\}$ be a set of probe functions defining visual features of the elements of $\mathcal{V}$. In a probabilistic approach, the value of each probe function at each visual element is a random variable. Let $\pi_1, \pi_2, ...\pi_l$ represent random variables associated with $\phi_1, \phi_2, ...\phi_l$. All feature values are real numbers. Let $f_{\pi_1}(.), f_{\pi_2}(.), ..., f_{\pi_l}(.)$ be the probability density function (pdf) of these random variables and hence by definition:

$$\begin{aligned} Pr(a \leq \pi_1 \leq b) &= \int_a^b f_{\pi_1}(\pi) d\pi \\ Pr(a \leq \pi_2 \leq b) &= \int_a^b f_{\pi_2}(\pi) d\pi \\ &\quad\quad\quad\quad\quad\quad ... \\ Pr(a \leq \pi_i \leq b) &= \int_a^b f_{\pi_i}(\pi) d\pi \end{aligned}$$

Example 2.3. *As an example, Fig. 2.8(a) shows probability distribution functions for feature values in a 4 dimensional feature space. Feature values are assumed to be normal independent random variables with Gaussian distributions. The mean and standard deviation of the distributions are 0.3, 0.6, 0.6, 0.7 and 0.1, 0.15, 0.05, 0.10 respectively. Also, Fig. 2.8(b)* *shows the corresponding empirical distributions (histograms) calculated for 400,000 randomly generated sample feature vectors using the given distributions.*



(a) Probability distribution functions (pdf)

(b) Empirical distributions (histograms)

Figure 2.8: An example of four random variables representing feature values (a) underlying pdf and (b) empirical distribution of the randomly generated samples.

Proposition 2.3. *Let x_0 be an arbitrary visual element and $\vec{\phi}_{\mathcal{B}}(x_0)$ be the corresponding feature vector in the feature space. Let $\vec{\pi} = [\pi_1 \;\; \pi_2 \;\; \ldots \;\; \pi_l]^T$ be a random feature vector with independent feature values where f_{π_i} is the probability density function of the i^{th} feature. Then, $|\pi_i - \phi_i(x_0)|$ is another random variable with the following probability distribution* *function.*

$$\begin{aligned}
\forall i \quad f_{|\pi_i-\phi_i(x_0)|}(\pi) &= f_{\pi_i}(\pi+\phi_i(x_0)) + f_{\pi_i}(-(\pi+\phi_i(x_0))) && \pi \geq 0 \\
&= 0 && \pi < 0
\end{aligned}$$

The proof directly follows from Lemma 1 and Lemma 2 in Appendix B.

Proposition 2.4. *Let $\vec{\pi} = [\pi_1 \ \ \pi_2 \ \ ... \ \ \pi_l]^T$ be a random feature vector and let $\vec{\phi}_{\mathcal{B}}(x_0)$ be the feature vector corresponding to an arbitrary chosen element x_0. Then $\Delta = \parallel \vec{\pi} - \vec{\phi}_{\mathcal{B}}(x_0) \parallel_1$ is another random variable with the following* `pdf` *and* `cdf` *where $\star$ stands for convolution.*

$$f_\Delta(e) = f_{|\pi_1-\phi_1(x_0)|}(e) \star f_{|\pi_2-\phi_2(x_0)|}(e) \ \star ... \star \ f_{|\pi_l-\phi_i(x_0)|}(e) \tag{2.18}$$

$$F_\Delta(e) = \int_{-\infty}^{e} f_{|\pi_1-\phi_1(x_0)|}(\tau) \star f_{|\pi_2-\phi_2(x_0)|}(\tau) \ \star ... \star \ f_{|\pi_l-\phi_l(x_0)|}(\tau) d\tau$$

Proof

$\Delta = \parallel \vec{\pi} - \vec{\phi}_{\mathcal{B}}(x_0) \parallel_1 = \sum_{i=1}^{i=l} |\pi_i - \phi_i(x_0)|$ (according to the definition of the L_1-norm-based distance). Therefore, probability distribution function f_Δ is the convolution of the density functions $|\pi_i - \phi_i(x_0)|$ for $i = 1, 2, ..., l$ and hence the proposition is proved. ■

Definition 2.10. Probability of being in a neighborhood

*Let $P_\varepsilon(x_0, O)$ represent the probability that a randomly selected element $y \in O$ satisfies the inequality $\parallel \vec{\phi}_{\mathcal{B}}(y) - \vec{\phi}_{\mathcal{B}}(x_0) \parallel \leq \varepsilon$ (*i.e. *the distance between feature vectors of x_0 and y is smaller than ε).*

Proposition 2.5. *Let x_0 be an arbitrary visual element with feature vector $\vec{\phi}_{\mathcal{B}}(x_0)$ and let y be a visual element randomly chosen from a set O of elements with a known distribution function of feature values. $P_\varepsilon(x_0, O)$ (as defined above) is calculated as follows.*

$$Pr\left(\| \ \vec{\phi}_{\mathcal{B}}(y) - \vec{\phi}_{\mathcal{B}}(x_0) \ \|_1 \leq \varepsilon\right) = \int_0^\varepsilon f_{|\vec{\phi}_{\mathcal{B}}(y)-\vec{\phi}_{\mathcal{B}}(x_0)|}(e)de \tag{2.19}$$

where $f_{|\vec{\phi}_{\mathcal{B}}(y)-\vec{\phi}_{\mathcal{B}}(x_0)|}(e)$ is calculated using Equation 2.18 in Proposition 2.4.

Proof

$\| \ \vec{\phi}_{\mathcal{B}}(y) - \vec{\phi}_{\mathcal{B}}(x_0) \ \|_1$ is a random variable. By definition, the probability that this random variable is between 0 and ε, is calculated by integrating its pdf from 0 to ε. ■

Proposition 2.6. Probability mass function of the size of tolerance neighborhoods

Let O be a finite set of randomly selected visual elements with known probability density functions and let $n_O(x_0)$ be a tolerance neighborhood of x_0 defined as follows,

$$n_O(x_0) = \{y \in O \mid \quad |\vec{\phi}_{\mathcal{B}}(y) - \vec{\phi}_{\mathcal{B}}(x_0)| \leq \varepsilon\}. \tag{2.20}$$

The size of the tolerance neighborhood $(|n_O(x_0)|)$ is a discrete random variable with the following probability mass function (`pmf`)*,*

$$m_{|n_O(x_0)|}(k) = Pr\ (\ |n_O(x_0)| = k\ ) = \binom{N_O}{k} P_\varepsilon(x_0, O)^k \times (1 - P_\varepsilon(x_0, O))^{(N_O - k)},$$

where $N_O = |O|$ is the size (cardinality) of the set O and $P_\varepsilon(x_0, O)$ is defined as in Equation 2.19.

Proof

The feature values of any element $y \in O$ are random variables with given pdfs. For each element y, the probability that this element belongs to the tolerance neighborhood $n_O(x_0)$ is shown with $P_\varepsilon(x_0, O)$ as defined in Equation 2.19. The size of each tolerance neighborhood is an integer number between 1 and N_O. For each value of $k \in [1, N_O]$, the size of the tolerance neighborhood is equal to k if there are k elements in O that belong to the neighborhood *and* $N_O - k$ elements that do not belong to the neighborhood. The probability of the former event is $P_1 = P_\varepsilon(x_0, O)^k$ and the probability of the later even is $P_2 = (1 - P_\varepsilon(x_0, O))^{(N_O - k)}$. The total probability of having a tolerance neighborhood of size k is then equal to the number of ways one can choose k elements from a set of N_O elements times P_1 times P_2. ■

Definition 2.11. Neighborhood distance $T(x_0)$: Covering of sets by a neighborhood *Let X, Y be two sets of randomly selected visual elements with feature values with known distribution functions and let $O = X \cup Y$. Suppose $x_0 \in X \cup Y$ is an arbitrary element in one of the sets. The corresponding tolerance neighborhood in O is shown with $n_{X \cup Y}(x_0)$. $T(x_0)$ is defined here as follows and named as* neighborhood distance.

$$T(x_0) = \frac{||n_O(x_0) \cap X| - |n_O(x_0) \cap Y||}{|n_O(x_0) \cap X| + |n_Y(x_0) \cap Y|} = \frac{||n_X(x_0)| - |n_Y(x_0)||}{|n_X(x_0)| + |n_Y(x_0)|} \tag{2.21}$$

Theorem 1. *$T(x_0)$ is a discrete random variable with the following probability mass function (*`pmf`*) assuming that the cardinality of the sets X and Y are the same. (The general formula for the case of non equal sizes, is given in the proof section of this proposition)*

$$m_{T(x_0)}(t) = Pr\left(T(x_0) = t\right) =$$

$$= \sum_{k} \left(m_{|n_X(x_0)|}(k\frac{1+t}{1-t}) + m_{|n_X(x_0)|}(k\frac{1-t}{1+t}) \right) m_{|n_Y(x_0)|}(k) \quad \textit{iff} \quad t \geq 0 \tag{2.22}$$

$$= 0 \quad \textit{iff} \quad t \leq 0$$

Proof

Since elements of the sets X and Y are randomly selected, their feature values are random variables and hence the size of tolerance neighborhoods are discrete random variables. Also, since $|n_X(x_0)|$ and $|n_Y(x_0)|$ only take finite discrete values, $T(x_0)$ only takes discrete values and hence $T(x_0)$ is a discrete random variable. $T(x_0)$ can be written as: $T(x_0) = \left|\frac{|n_X(x_0)|-|n_Y(x_0)|}{|n_X(x_0)|+|n_Y(x_0)|}\right| = \left|\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}\right|$

According to Lemma 3 in Appendix B, the probability mass function of the random variable $T(x_0)$ can be written as: $Pr\left(T(x_0) = t\right) = m_{T(x_0)}(t) = m_{\left|\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}\right|}(t)$

$$= m_{\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}}(t) + m_{\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}}(-t) \quad \text{iff} \quad t \geq 0$$

$$= 0 \quad \text{iff} \quad t \leq 0$$

Moreover, the probability mass function $m_{\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}}(t)$ can be re-written as follows:

$$m_{\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1}}(t) = Pr\left(\frac{\frac{|n_X(x_0)|}{|n_Y(x_0)|}-1}{\frac{|n_X(x_0)|}{|n_Y(x_0)|}+1} = t\right) = Pr\left(\frac{|n_X(x_0)|}{|n_Y(x_0)|} = \frac{1+t}{1-t}\right) = m_{\frac{|n_X(x_0)|}{|n_Y(x_0)|}}(\frac{1+t}{1-t}) =$$

$$= \sum_{k} m_{|n_X(x_0)|}(\frac{1+t}{1-t} \times k)\, m_{|n_Y(x_0)|}(k)$$

Therefore, substituting into the above equation,

$Pr\,(T(x_0) = t) = m_{T(x_0)}(t) =$

$$= \sum_k m_{|n_X(x_0)|}(\frac{1+t}{1-t} \times k)\; m_{|n_Y(x_0)|}(k) + \sum_k m_{|n_X(x_0)|}(\frac{1-t}{1+t} \times k)\; m_{|n_Y(x_0)|}(k) \quad \text{iff} \quad t \geq 0$$

$$= 0 \quad \text{iff} \quad t \leq 0$$

For the general case of $|X| = |Y| = N$, the equation can be simplified as

$$\begin{aligned} m_{T(x_0)}(t) &= \sum_{k=0}^{N} \left(m_{|n_X(x_0)|}(k\frac{1+t}{1-t}) + m_{|n_X(x_0)|}(k\frac{1-t}{1+t}) \right) m_{|n_Y(x_0)|}(k) \quad \text{iff} \quad t \geq 0 \\ &= 0 \quad \text{iff} \quad t \leq 0 \end{aligned} \tag{2.23}$$

The proof is complete. ■

In this section, it is shown how the value of the neighborhood distance $T_{X,Y}(x_0)$ depends on distribution functions of the feature values in images X and Y. For any randomly selected pair of query and test images (X and Y) and any given and known visual element x_0, $T(x_0)$ is a random variable with the probability mass function given in Equation 2.22 above. This value is expected to be higher when images X and Y are selected from two different sets of images with different distribution of feature values. This is shown in the next example.

Example 2.4. *Suppose there are two sets of images namely Q and T. Images in each set are considered as similar images with known distribution functions of feature values. Let X (query) and Y (test) be two images (sets of visual elements) randomly selected from Q or T as shown in Fig. 2.9. Each visual element (x) of an image, is described with a 4 dimensional*

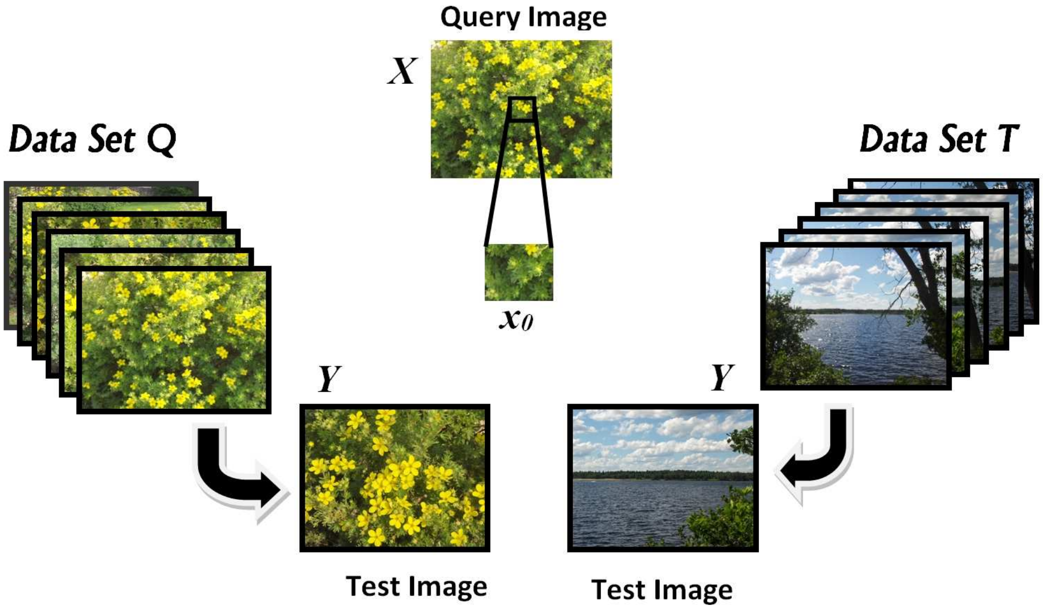
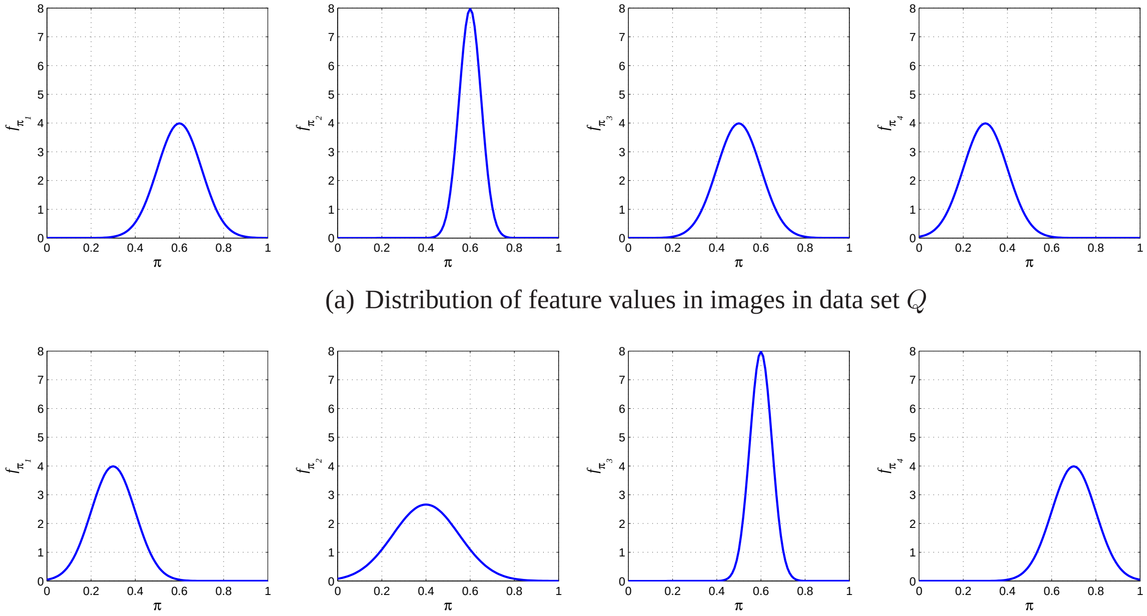


Figure 2.9: Comparing a query image (X) and a test image (Y) where the query image and the test image come from the same dataset of images (Q) or alternatively from two different datasets (Q and T).

*feature vector $\vec{\phi}_{\mathcal{B}}(x) = [\pi_1\ \pi_2\ \pi_3\ \pi_4]^T$. Feature values π_1 to π_4 are considered as independent random variables with normal probability density functions f_{π_1} to f_{π_4} as shown in Fig. 2.10 for Q and T. Now, let x_0 be a sample visual element with a given known feature vector (*e.g. $\vec{\phi}_{\mathcal{B}}(x_0) = [0.5\ \ 0.5\ \ 0.5\ \ 0.5]^T$*). The neighborhood distance $T_{X,Y}(x_0)$ between sets X and Y with respect to x_0 is a random variable with a probability mass function that can be calculated according to Equation 2.22. This function represents the nominal distribution of the values of $T_{X,Y}(x_0)$.*

In order to verify Equation 2.22 in practice, 2000 pairs of sets X and Y have been randomly generated using the given pdfs for feature values where each set contains 200 visual elements simulating an image with 200 subimages. For each pair, the deterministic value



(a) Distribution of feature values in images in data set Q

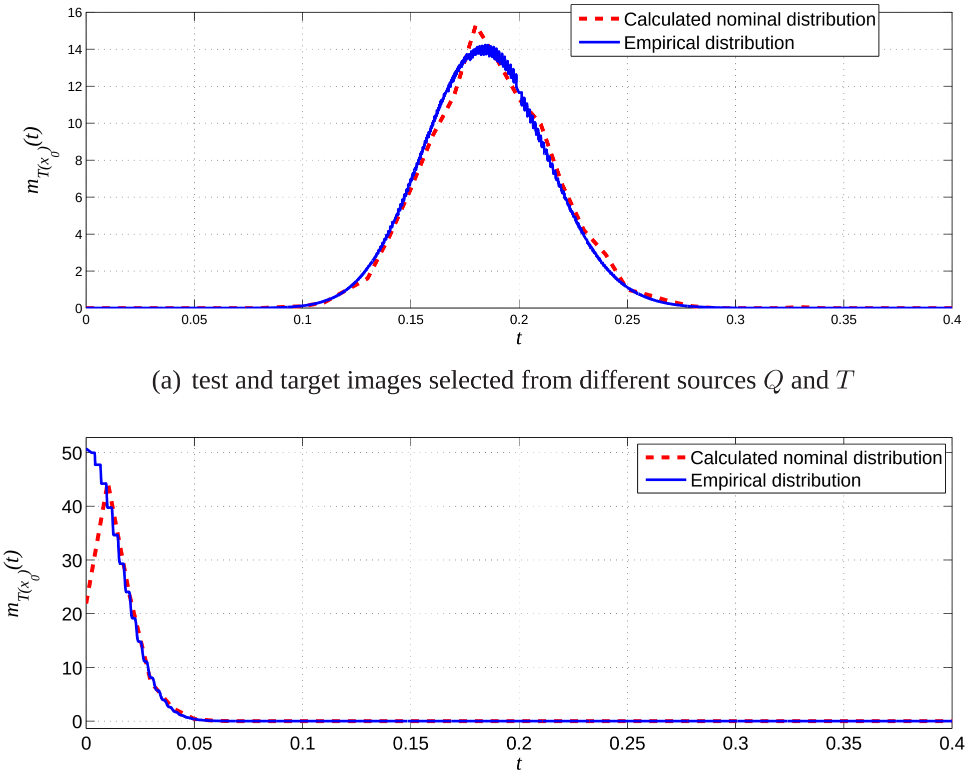
(b) Distribution of feature values in images in dataset T

Figure 2.10: Distributions of feature values in (a) query image and (b) test image, an example.

of neighborhood distance $T_{X,Y}(x_0)$ has been calculated according to Equation 2.11. The empirical distribution of the values of $T_{X,Y}(x_0)$ has been calculated and plotted in Fig. 2.11. Also, the nominal distribution function in Equation 2.22 has been plotted in the same figure. The was done for two difference cases where (a) X and Y are randomly selected from the same data set (Q) and (b) X and Y are randomly selected from different datasets Q and T, respectively.

The following conclusions can be made from the above example and the given results in Fig. 2.11.

- The derived equation for distribution of neighborhood distance in Equation 2.22 in Theorem 2.11 conforms with the empirical distribution of values measured.



(a) test and target images selected from different sources Q and T

(b) test and target sets are selected from the same source Q

Figure 2.11: Nominal and empirical pmfs of $T(x_0)$ between a query image and a test image taken from (a) different data sets and (b) the same data sets of images

- The neighborhood distance $T_{X,Y}(x_0)$ will be significantly smaller (higher similarity), if X and Y have the same distribution of feature values (selected from the same dataset).

The overall distance measure $tcDM$ is simply the average of the neighborhood distances calculated at different visual elements. Therefore, the probability mass function of the actual distance measure can be further calculated by taking the convolution of the individual distribution functions as stated below.

Proposition 2.7. *Distribution of* $tcDM$

Let $X = \{x_1, x_2, ..., x_N\}$ and $Y = \{x_1, x_2, ..., x_N\}$ be sets of visual elements (images) corresponding to two randomly selected images referred to as query (X) and test (Y) image. Also, let (Q) and (T) represent dataset of images with known distribution functions of the feature values. According to Equation 2.12, $tcDM$ is the sum of neighborhood distances (normalized) and hence its probability mass function (pmf) is the convolution of the pmf of neighborhood distances. $tcDM$ is a discrete random variable with values between 0 and 1 with the following probability mass function

$$m_{tcDM}(d) = Pr(tcDM = d) = \frac{1}{N} \times m_{T(x_1)}(d) \star \cdots \star m_{T(x_N)}(d) \tag{2.24}$$

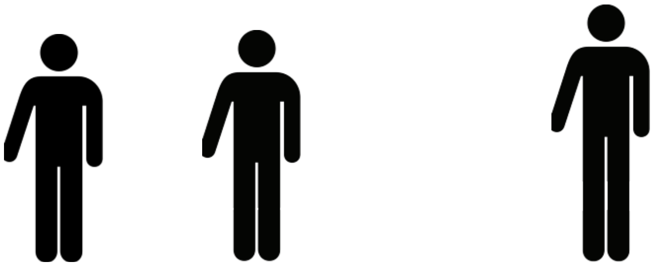
Chapter 3

Fuzzy Tolerance Spaces and Similarity

To be or not to be, that is NOT the question.

In the previous chapters, it was shown how tolerance relation can be used in modeling the existing imprecision in human visual perception of the physical world. Two describable objects can be considered *almost* indiscernible if the difference between their descriptions is smaller than a tolerable level of error (ε). Tolerance relations can be used as a basic framework for modeling this tolerance level of difference in descriptions. Figure 3.1 shows three different images (a),(b) and (c) where the important description (physical feature) is the height of the man depicted in each picture. A first glance at the images shows that the height of (c) is higher than (a) and (b). Also, (a) and (b) seem to have the same height. However, the actual value of the height of (b) is slightly higher than (a). The small difference in their visual feature (height) makes us consider (a) and (b) as *almost* similar. Even if we notice the difference, we may consider this difference insignificant for many purposes. In mathematical language, we say $a \cong_{\mathcal{B},\varepsilon} b$, that means a tolerance relation $\cong_{\mathcal{B},\varepsilon}$ exists between a and b, where

$\mathcal{B} = \{Height(.)\}$ is the only probe function and $\varepsilon = 1cm$. $(\{a, b, c\}, \cong_{\mathcal{B},\varepsilon})$ is a tolerance space. The existing tolerance in overlooking small changes in visual appearances is one aspect of the human perception. However, it is not clear if there is a sharp crisp threshold for this tolerance. In summary, the following three observations can be made about human perception of the notion of similarity.



(a) Amir (b) Ali (c) Farshid

Figure 3.1: An example: the human mind tolerance in perception and description of height

- Exact equality of descriptions is not necessary in order to consider two objects similar. There is always a tolerable level of error in comparing objects by their description. Incorporating the concept of tolerance is not only allowable but also needed to arrive at *approximate* solutions of problems in real world. Tolerance space theory can be used as a framework to incorporate this idea in a computational model of similarity.

- Transition from “similar” to “dissimilar” in human mind is gradual not abrupt. There is no boundary between “similar” and “dissimilar” and it is just the matter of degree of similarity. Therefore, there is an inherent fuzziness in this concept. Fuzzy relation is a solution for incorporating the concept of fuzziness or imprecision in definition of similarity.

- Human judgment of similarity is normally expressed in natural language. Expressions like *very similar*, *almost identical* and *different* are easily used by humans to express similarity. This judgment is also highly subjective and uncertain in nature. A fuzzy-valued similarity/distance measure, will make it possible to utilize human domain knowledge and *approximate reasoning* techniques in a computational algorithm.

The objective of this chapter is to introduce a more general approach based on *fuzzy tolerance relations* that can address all the above aspects in defining the similarity between objects or sets of objects.

Fuzzy sets were introduced in 1965 in a seminal paper by L. A. Zadeh [76]. One of the main advantages of fuzzy sets is their ability in modeling the uncertainty in assigning the degree of membership of an element in a set. The question of *whether or not* an element belongs to a set, is replaced with the question of *how much* the element belongs to the set. This is possible through adopting a so called *fuzzy logic* in which the truth value of a statement can be any real number between 0 and 1 representing the degree of truth. The membership of an element x to a set A is then defined by a membership function $\mu_A(x)$ that represents *degree of membership* of the element in the set. This property of fuzzy sets was a huge breakthrough in computational intelligence by proper handling the uncertainty in defining the boundary of a set.

3.1 Background on Fuzzy Sets and Fuzzy Tolerance Relations

Definition 3.1. Fuzzy Set *[76, 79]*

A fuzzy set $\mathcal{A}$ defined on a universe of discourse U is characterized by a membership function, $\mu_{\mathcal{A}} : U \rightarrow [0, 1]$ that assigns a degree of membership $\mu_{\mathcal{A}}(y)$ to any element $y \in U$. A fuzzy set may be represented as the union of all the pairs of elements of U and their respective membership grades in $\mathcal{A}$. This is shown in either of the following forms

$$\mathcal{A} = \{(y, \mu_{\mathcal{A}}(y)) \mid y \in U,\ \mu_{\mathcal{A}}(y) \in [0, 1]\} \tag{3.1}$$

$$\mathcal{A} = \int_U \mu_{\mathcal{A}}(y)/y \tag{3.2}$$

where $(y, \mu_{\mathcal{A}}(y))$ (alternatively $\mu_{\mathcal{A}}(y)/y$) is named a singleton*, and the fuzzy set is defined as the union of its constituent singletons.*

Example 3.1. *Let $U = [0 \;\; 200]$ represent the set of possible values for height of a person in centimeters. Theory of fuzzy sets allows us to define the concept of "Tall" by defining a fuzzy set ($Tall$) that represents the set of tall people. This fuzzy set is given by the grade of membership of each person to the fuzzy set of tall people using a membership function $\mu_{Tall} : U \rightarrow [0, 1]$. An immediate implication of this, is the subjectivity of the concept. Degree of membership of any person to the fuzzy set of tall people can then be obtained by evaluating the membership function $\mu_{Tall}(x)$ at x, where x is the height of the person. Figure 3.2 shows an example of a membership function defined based on the height.*

Since fuzzy set theory is a generalization of the classical set theory, it rapidly found its way into many areas of mathematics. Fuzzy relations (also introduced in [76]), were one of the first implications of fuzzy sets. A classical relation R defined on a set X is a subset of $X \times X$ where any of the elements of the Cartesian product has a crisp degree of membership (0 or 1) in the set R. Similarly, a *fuzzy relation* $\hat{R}$ defined on a crisp set X is a "fuzzy set"

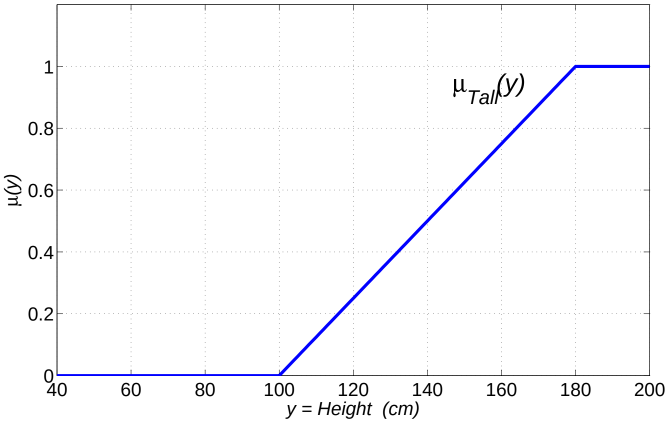


Figure 3.2: An example: Tall membership function

defined as follows where the membership function represents degree of membership of each pair of elements in the relation (*i.e.* the degree to which, the elements are related to each other).

$$\hat{R} = \{((x, y), \mu_{\hat{R}}(x, y)) \quad | \quad (x, y) \in X \times X,\ \mu_{\hat{R}}(x, y) \in [0, 1]\} \tag{3.3}$$

Furthermore, many of the conventional concepts in set theory can be "fuzzified". A conventional equivalence relation is a relation that is *reflexive*, *symmetric* and *transitive* (refer to Chapter 2, Definition 2.1). Therefore, one may defines a fuzzy equivalence relation as follows.

Definition 3.2. Fuzzy Equivalence Relation *[56, 77]*

Let $\hat{R}$ be a fuzzy relation defined on X using the membership function $\mu_{\hat{R}}(x, y)$. $\hat{R}$ is a fuzzy equivalence relation iff it has all the following properties:

- *Reflexivity:* $\forall x \in X \quad \mu_{\hat{R}}(x, x) = 1$
- *Symmetry:* $\forall x, y \in X \quad \mu_{\hat{R}}(x, y) = \mu_{\hat{R}}(y, x)$

- *Transitivity* $\forall x, y, z \in X \quad \mu_{\hat{R}}(x,z) \geq \mu_{\hat{R}}(x,y) \star \mu_{\hat{R}}(y,z)$

where $\star$ represents a triangular norm *(t-norm). A t-norm is a commutative, monotonic and associative binary operation defined on* $[0,1] \times [0,1]$ *into* $[0,1]$. *A simple common example of such function is the "minimum" function also named as Gōdel t-norm.*

NOTE 1: A different definition for reflexivity is also given in [10]: According to this definition, fuzzy binary relation $\hat{R}$ is reflexive iff $\forall x, y \in X, \quad 0 < \mu_{\hat{R}}(x,x) \geq \mu_{\hat{R}}(x,y)$.

NOTE 2: The idea of using equivalence relation to represent similarity, is intuitive and natural. In fact, when Zadeh first introduced fuzzy equivalence relations in 1971 [77], he named them *similarity relations*. Equivalence relations demonstrate the relation between indiscernible objects and it is plausible to consider two indiscernible objects as similar. Fuzzy equivalence relations generalize the idea of similarity from absolute to partial. This can be done by introducing degree of similarity ranging between 0 and 1.

NOTE 3: There are many philosophical debates and discussions on whether a similarity relation should require these three properties (see *e.g* [71]). Transitivity in particular, is not always believed to be necessary for a similarity relation and that was why tolerance relations were introduced as an extension of indiscernibility relation (Chapter 2). Fuzzification of the concept of tolerance relation leads us to the concept of fuzzy tolerance relation as defined in [8, 10].

Definition 3.3. Fuzzy Tolerance Relation *[8, 10]*

A fuzzy tolerance relation is a fuzzy relation which is reflexive and symmetric.

Example 3.2. *Let* $X = \{1, 0.2, -0.5, 0.3\}$ *be a set of numbers and* $\hat{R}$ *is a fuzzy relation defined on* X *by the membership function* $\mu_{\hat{R}}(x,y) = 1 - |x-y| \quad \forall x, y \in X$.

	1	0.2	-0.5	0.3
1	1	0.2	0.5	0.3
0.2	0.2	1	0.7	0.9
-0.5	0.5	07	1	0.3
0.3	0.3	0.9	0.3	1

Figure 3.3: Tabular view of the fuzzy tolerance relation given in Example 3.2

$\hat{R}$ is a fuzzy tolerance relation because it is both reflexive ($\mu_{\hat{R}}(x, x) = 1$) and symmetric $\mu_{\hat{R}}(x, y) = 1 - |x - y| = 1 - |y - x| = \mu_{\hat{R}}(y, x)$. $\hat{R}$ can be represented using a table shown in Fig. 3.3.

3.2 Linguistic and Fuzzy-Valued Distances

The main motivation for using fuzzy set theory in definition of similarity measures in this thesis is to allow a more humanistic natural-language compatible form of distance measures between pairs of images. Humans do not use numbers to express similarity between images. Instead, human-judged similarities are expressed in terms of natural language expressions like *identical*, *very similar*, *partially similar*, *not similar*, etc. Moreover, what someone means by *very similar* (for example), is highly subjective and also depends on the context. In this thesis, three different forms of distance/similarity measures can be considered as explained below. [1] Let $\mathcal{X}, \mathcal{Y} \in \mathcal{I}$ be pairs of images where $\mathcal{I}$ is the infinite set of possible images (or finite set of images under consideration).

- Numerical-valued Distance Measure (NVDM): A distance measure $d(\mathcal{X}, \mathcal{Y}) \in \mathbb{R}^+$

[1] The terms FVDM and LVDM are introduced here for the first time.

where the value of distance is represented with a numerical and real positive number. This is a classical traditional distance that may be bounded or unbounded.

$$d : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{R}^+ \tag{3.4}$$

- Fuzzy-valued Distance Measure (FVDM): Distance is represented with a fuzzy set $\hat{D}$. Here, no single numerical value can represent the distance, but each numerical value d has a degree of membership $\mu_{\hat{D}}(d)$ that represents degree of truth of the statement "*Distance between $\mathcal{X}$ and $\mathcal{Y}$ is d*".

$$\hat{D} :\mathcal{I} \times \mathcal{I} \rightarrow \mathcal{F}(\mathbb{R}^+) \tag{3.5}$$

$$\hat{D}(\mathcal{X}, \mathcal{Y}) = \{(d, \mu_{\hat{D}}(d)) \quad | \quad d \in \mathbb{R}^+,\ \mu_{\hat{D}}(d) \in [0,1]\} \tag{3.6}$$

$\mathcal{F}(\mathbb{R}^+)$ represents the family of fuzzy sets defined on the positive real numbers.

- Linguistic-valued Distance Measure (LVDM): LVDM is a distance that is a *linguistic variable* (see Definition 3.4 below). LVDM can take any value from a set of finite linguistic values usually represented with a fuzzy set. (*e.g.* `Distance = ''very small''`)

$$\hat{D} : \mathcal{I} \times \mathcal{I} \rightarrow \{\texttt{'Very Small', 'Small', 'Large', 'Very Large'}\} \tag{3.7}$$

Definition 3.4. Linguistic Variable

A linguistic variable *is a variable that can take values from natural language terms [78, 79].*

Table 3.1: Different types of distance measures

Distance Type	Variable	Possible Values
Numerical	d	$[0, \inf]$
Fuzzy Valued	$\hat{D}$	$\mathcal{F}([0, \inf])$
Linguistic Valued	$\hat{D}$	*e.g.* { 'Very Small','Small','Large','Very Large' }

For example in the expression `Height = 'Tall'`*, the left hand side (*`Height`*) is a linguistic variable. The right hand side is an expression that prior to the invention of fuzzy sets, existed only in a natural language. However, one can define a fuzzy set that defines the value* `'Tall'` *using a fuzzy membership function (*e.g. *Fig. 3.2). The ambiguity in defining the concept of tall using the actual height of a person in centimeter is modeled through allowing the grade of membership. Other possible values for the linguistic variable* `'Height'` *can be* `'Very tall'`,`'Not tall'` *or* `'Short'`.

3.3 New Method: Fuzzy Tolerance Relations and Similarity

In this section, a descriptive-based fuzzy tolerance relation is introduced as a generalization of the perceptual tolerance relation that was defined in Chapter 2.

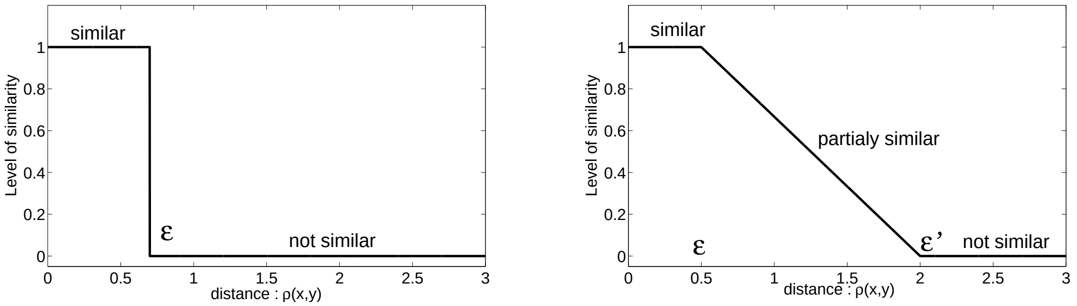
Definition 3.5. Perceptual Fuzzy Tolerance Relation

*Let O be a set of describable objects, $\mathcal{B}$ be a set of probe functions and $\Phi_{\mathcal{B}}^{O}$ be the set of feature vectors corresponding to elements of O. Suppose ρ is a distance metric on $\Phi_{\mathcal{B}}^{O}$ that defines the distance between feature vectors (*e.g *$\rho = \| \cdot \|_2$). Let $\varepsilon < \varepsilon' \in \mathbb{R}$. A perceptual* *fuzzy tolerance relation* $\hat{\cong}_{\mathcal{B},\varepsilon} :\ O \times O \rightarrow [0,1]$ *is defined here as follows.*

$$\begin{aligned} \hat{\cong}_{\mathcal{B},\varepsilon}(x,y) &= 1 \qquad\quad \textit{if} \quad \rho(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon \\ &= \frac{\varepsilon' - \rho(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)}{\varepsilon' - \varepsilon} \quad \textit{if} \quad \varepsilon < \rho(\vec{\phi}_{\mathcal{B}}(x), \vec{\phi}_{\mathcal{B}}(y)) < \varepsilon' \\ &= 0 \quad \textit{Otherwise} \end{aligned} \tag{3.8}$$

$\hat{\cong}_{\mathcal{B},\varepsilon}$ *is a fuzzy tolerance relation since it is reflexive* $(\forall x \in O, \quad \hat{\cong}_{\mathcal{B},\varepsilon}(x,x) = 1)$ *and symmetric* $(\forall x, y \in O \quad \hat{\cong}_{\mathcal{B},\varepsilon}(x,y) = \hat{\cong}_{\mathcal{B},\varepsilon}(y,x))$.

Figure 3.4(b) displays how the transition between similarity and non-similarity is gradual in a fuzzy tolerance relation. This transition in a classical perceptual tolerance relation is sharp and crisp.



(a) Tolerance relation with a crisp transition

(b) A fuzzy tolerance relation with gradual transition

Figure 3.4: Crisp and fuzzy tolerance relations and the transition between similarity and non-similarity

Definition 3.6. Fuzzy Tolerance Neighborhood

Let $(O, \hat{\cong}_{\mathcal{B},\varepsilon})$ *be a fuzzy tolerance space. Fuzzy tolerance neighborhood of an element* $x \in O$ *(shown here with* $n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}(x)$*) is defined as a fuzzy subset of* O *that provides the grade* *of membership of elements in the neighborhood.*

$$n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}(x) = \{(y, \mu_{\left(n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}(x)\right)}(y)) \quad | \quad y \in O\} \tag{3.9}$$

where the membership value of any element y in the fuzzy neighborhood of x, can be defined as

$$\mu_{n_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}(x)}(y) = \hat{\cong}_{\mathcal{B},\varepsilon}(x, y) \tag{3.10}$$

For the sake of simplicity, if $\hat{\cong}_{\mathcal{B},\varepsilon}$ is known, one may use the notation $\hat{n}_O$ to represent the fuzzy tolerance neighborhood. The family of all *fuzzy tolerance neighborhoods* and the family of all *fuzzy tolerance classes* in the tolerance space, are shown here with $\mathsf{N}_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}$ and $\mathsf{H}_O^{\hat{\cong}_{\mathcal{B},\varepsilon}}$, respectively.

3.3.1 Fuzzy Similarity/Distance Measure between Sets of Describable Objects

Fuzzy tolerance covering distance measure: ftcDM

A fuzzy-tolerance-covering-Distance-Measure ($ftcDM$) is proposed here as a numerical valued crisp distance measure obtained using a fuzzy tolerance neighborhood. $ftcDM$ between pairs of images $\mathcal{X}, \mathcal{Y}$ is defined by the following equation.

$$ftcDM(\mathcal{X}, \mathcal{Y}) = \frac{1}{|X| + |Y|} \sum_{x \in (X \cup Y)} \frac{|\parallel \hat{n}_O(x) \star X \parallel - \parallel \hat{n}_O(x) \star Y \parallel|}{\parallel \hat{n}_O(x) \parallel} \tag{3.11}$$

where $\star$ is a fuzzy *t-norm* operation. Note that $\mathcal{X}, \mathcal{Y}$ are pairs of images and X, Y represent sets of describable objects (visual elements) corresponding to images $\mathcal{X}, \mathcal{Y}$. $|X|, |Y|$ are the cardinality of sets X and Y, respectively. $\|\ .\ \|$ represents the cardinality of a fuzzy set and is defined as the sum of the membership values of all the elements in a set (as defined in [13]).

Fuzzy tolerance covering fuzzy distance measure: ftcFDM

A fuzzy-tolerance-covering-Fuzzy-Distance-Measure ($ftcFDM$) is defined here as a fuzzy valued distance measure (FVDM). Therefore, this distance is a fuzzy set rather than a numerical value. $ftcFDM$ between pairs of images $\mathcal{X}, \mathcal{Y}$ is defined as follows. Let X, Y be sets of describable objects corresponding to $\mathcal{X}, \mathcal{Y}$ and $O = X \cup Y$. Let ρ be a distance function between feature vectors where $(\Phi_{\mathcal{B}}^{O}, \rho)$ is a metric space. Furthermore, let $\hat{\cong}_{\mathcal{B},\varepsilon}$ be a fuzzy tolerance relation where $(O, \hat{\cong}_{\mathcal{B},\varepsilon})$ is a fuzzy tolerance space. Let $\star$ be a $t-norm$ operation such as $minimum$. First, element-to-set fuzzy-valued distances are defined.

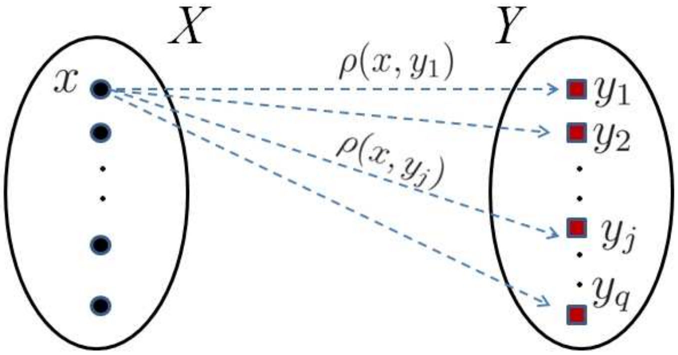


Figure 3.5: Fuzzy distance is defined based on the set of elementary classical distances ρ between an element x and all the element of a set.

Definition 3.7. Element-to-set fuzzy-valued distance

An element to set fuzzy valued distance between an element x and a set Y is a fuzzy set defined *on the set of positive real numbers* $[0,\infty)$ *shown with a membership function* $\mu_{x\to Y}(d)$ *that has the following membership function,*

$$\mu_{x\to Y}(d) = \hat{\cong}_{\mathcal{B},\varepsilon}(x, \operatorname*{argmin}_{y\in Y}\{|d-\rho(x,y)|\}) \quad d\in[0,\infty),\ x\in X. \tag{3.12}$$

In other words, for a particular element x, membership function $\mu_{x\to Y}(d)$ at each point d is defined as the fuzzy tolerance level of those pairs of elements ($\hat{\cong}_{\mathcal{B},\varepsilon}(x,y)$) that have this distance ($\rho(x,y)=d$). If no pair of elements with that value of distance exist, then the closest pair distance to d, (which is $(x, \operatorname{argmin}_{y\in Y}\{|d-\rho(x,y)|\})$) is considered. Figure 3.6 shows a table of pairs of values and how they are used to define $\mu_{x\to Y}$. Similarly, the distance between the element $x\in X$ and set X can be defined as:

$$\mu_{x\to X}(d) = \hat{\cong}_{\mathcal{B},\varepsilon}(x, \operatorname*{argmin}_{s\in X}\{|d-\rho(x,s)|\}) \quad d\in[0,\infty), x\in X$$

d	$\mu_{x\to Y}(d)$
$\rho_1=\rho(x,y_1)$	$\mu_{x\to Y}(\rho_1)=\mu_{\hat{\cong}_{\mathcal{B},\varepsilon}}(d)\vert_{d=\rho_1}$
$\rho_2=\rho(x,y_2)$	$\mu_{x\to Y}(\rho_2)=\mu_{\hat{\cong}_{\mathcal{B},\varepsilon}}(d)\vert_{d=\rho_2}$
$\dots$	$\dots$
$\dots$	$\dots$
$\rho_j=\rho(x,y_j)$	$\mu_{x\to Y}(\rho_j)=\mu_{\hat{\cong}_{\mathcal{B},\varepsilon}}(d)\vert_{d=\rho_j}$
$\dots$	$\dots$
$\rho_q=\rho(x,y_q)$	$\mu_{x\to Y}(\rho_q)=\mu_{\hat{\cong}_{\mathcal{B},\varepsilon}}(d)\vert_{d=\rho_q}$

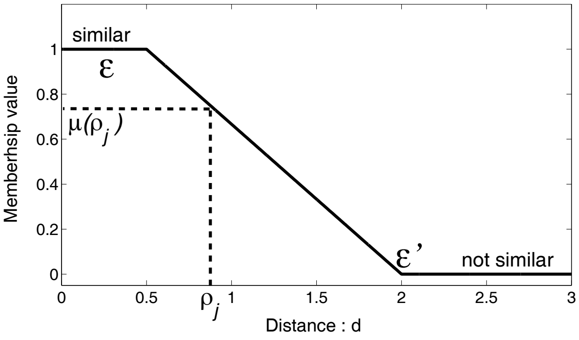


Figure 3.6: The membership function $\mu_{x\to Y}(d)$ at different values of $d=\rho_j$.

Definition 3.8. Directional fuzzy distances

Directional fuzzy distance between images is defined by taking the average (over all the elements of an image) of t-norm *between element-to-set fuzzy distances.*

$$\mu_{X \to Y}(d) = \frac{1}{|X|} \sum_{x_i \in X} \left(\mu_{x_i \to X}(d) \star \mu_{x_i \to Y}(d) \right) \tag{3.13}$$

$$\mu_{Y \to X}(d) = \frac{1}{|Y|} \sum_{y_i \in Y} \left(\mu_{y_i \to Y}(d) \star \mu_{y_i \to X}(d) \right) \tag{3.14}$$

where $\mu_{x_i \to X}(d) \star \mu_{x_i \to Y}(d)$ is a fuzzy set that represents a fuzzy distance between X and Y with respect to the fuzzy tolerance neighborhood centered at x. The overall (directional) distance is then calculated by taking the average of all fuzzy distances over all the elements in X. Finally, fuzzy valued distance $ftcFDM$ is defined as a fuzzy set with a membership function that is obtained by taking the *t-norm* between directional fuzzy distances.

$$ftcFDM(\mathcal{X}, \mathcal{Y}) = \{(d, \mu_{ftcFDM}(d)) \mid \quad d \in [0, \infty), \quad \mu_{ftcFDM}(d) \in [0, 1]\} \tag{3.15}$$

$$\mu_{ftcFDM}(d) = \mu_{X \to Y}(d) \star \mu_{Y \to X}(d) \tag{3.16}$$

Example 3.3. *Let $\mathcal{X}, \mathcal{Y}$ be the images shown in Fig. 3.7(a), where each image is represented by four describable objects (squar subimages) and the only probe function is the average gray scale value of a describable object. Here $\Phi^X_{gray} = \{0, 0.3, 0.7, 0.5\}$ and $\Phi^Y_{gray} = \{0.2, 0.8, 0.6, 0.4\}$ are sets of scalar features. Let $\rho = |.|$ be the Manhattan distance (L_1) metric between feature vectors. Let $\hat{\cong}_{\mathcal{B},\varepsilon}$ be a fuzzy tolerance relation defined as in Equation 3.8 where $\varepsilon = 0.1$ and $\varepsilon' = 0.45$. Distance metric ρ and tolerance relations $\hat{\cong}_{\mathcal{B},\varepsilon}$ between describable objects are shown in Fig. 3.7(b) and Fig. 3.7(c). Here, $tDM = 0.14$*

and $tcDM = 0.08$ *while* $ftcFDM$ *is a fuzzy valued measure shown with the membership function plotted in Fig. 3.7(d).*

Fuzzy distance measure and ordering

It is natural to think of an ordering relation with respect to the concept of distance (/nearness) measure between images. That means some images are closer to each other and some images are further apart. When distance between images is defined as a real number, its values will be a subset of real numbers and hence they can be viewed as an ordered set with conventional ordering relation between real numbers (shown with $<$ and $>$). Therefore, if $d(.)$ is a distance measure between images, then $d(\mathcal{X}, \mathcal{Y}) < d(\mathcal{X}, \mathcal{Z})$ iff $\mathcal{X}, \mathcal{Y}$ are closer to each other (more similar) than $\mathcal{X}, \mathcal{Z}$. This ordering relation is required in an image retrieval application in which images are needed to be *sorted* based on their distances to a query image.

In a fuzzy valued distance/nearness measure, however, this ordering relation may not be well defined. In the case of *fuzzy numbers* (convex, normalized fuzzy sets), there are existing methods for ranking/ordering fuzzy numbers. In a more general case, a fuzzy valued measure is represented with a fuzzy set that may or may not be a fuzzy number. In this case, one can define a *partial ordering relation* between fuzzy valued measures by defining an inclusion relation (being a subset of another set) between fuzzy sets. One, may define a fuzzy set inclusion in different ways (*e.g.* see [13], pages 22-24). In the original paper on fuzzy sets by L.A. Zadeh,, the inclusion (/containment) is defined as follows.

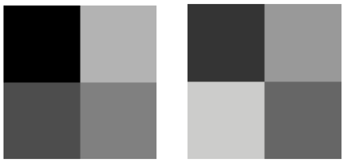
Definition 3.9. *Let* A *and* B *be fuzzy sets defined with membership functions* $\mu_A(x)$ *and* $\mu_B(x)$ *over the universe of discourse* U*, then* A *is a subset of* B *(or* A *is smaller or equal to* B*) iff* $\mu_A(x) \leq \mu_B(x) \quad \forall x \in U$.

This definition creates a *partial ordering relation* because it is possible to have a pair of fuzzy sets where none of them is a subset of the other. However, in this thesis, a full ordering relation is defined between fuzzy valued nearness measures by using the membership value of the fuzzy set $ftcFDM$ at $d = 0$. $ftcFDM_0$ is defined as follows.

$$ftcFDM_0 = 1 - \mu_{ftcFDM}(d)|_{d=0} \tag{3.17}$$

Then we say $ftcFDM(\mathcal{X}, \mathcal{Y}) \leq ftcFDM(\mathcal{X}, \mathcal{Z})$ (or equivalently $\mathcal{X}, \mathcal{Y}$ are closer to each other than $\mathcal{X}, \mathcal{Z}$ with respect to the fuzzy distance measure $ftcFDM$), if and only if
$ftcFDM_0(\mathcal{X}, \mathcal{Y}) \leq ftcFDM_0(\mathcal{X}, \mathcal{Z})$.
The motivation for definition of $ftcFDM_0$ is quite intuitive because $ftcFDM_0(\mathcal{X}, \mathcal{Y})$ can be viewed as the degree of truth of the statement: "*Distance between $\mathcal{X}$ and $\mathcal{Y}$ is zero*" or equivalently it is the degree of truth of the statement "*$\mathcal{X}$ and $\mathcal{Y}$ are the same (completely similar)*".

Example 3.4. *In this example, 4 different pairs of images are compared to each other (Fig. 3.8). Let $\hat{D} = ftcFDM$ be the fuzzy valued distance measure. In each case, membership function of ($\hat{D}$) is plotted. The plots show that $ftcFDM_0$ is zero for the first given pair of images (pair1, identical images). The other pairs of images (2,3 and 4 in order), have less similarity (more distance). These results are consistent with human judgment on the similarity between these images. However, judgment about ranking of pairs 2 and 3 is more likely to depend on human opinion.*



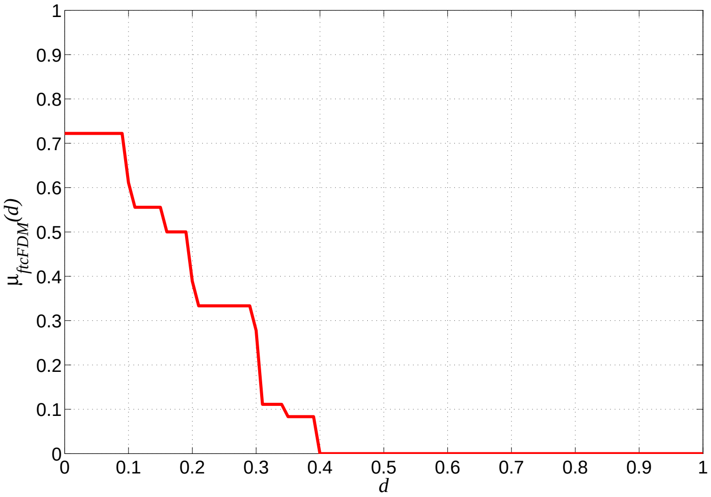
(a) $\mathcal{X}, \mathcal{Y}$, two sets, 4 visual elements each

ρ	x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
x_1	0.0	0.3	0.7	0.5	0.2	0.8	0.6	0.4
x_2	0.3	0.0	0.4	0.2	0.1	0.5	0.3	0.1
x_3	0.7	0.4	0.0	0.2	0.5	0.1	0.1	0.3
x_4	0.5	0.2	0.2	0.0	0.3	0.3	0.1	0.1
y_1	0.2	0.1	0.5	0.3	0.0	0.6	0.4	0.2
y_2	0.8	0.5	0.1	0.3	0.6	0.0	0.2	0.4
y_3	0.6	0.3	0.1	0.1	0.4	0.2	0.0	0.2
y_4	0.4	0.1	0.3	0.1	0.2	0.4	0.2	0.0

(b) Distance Metric

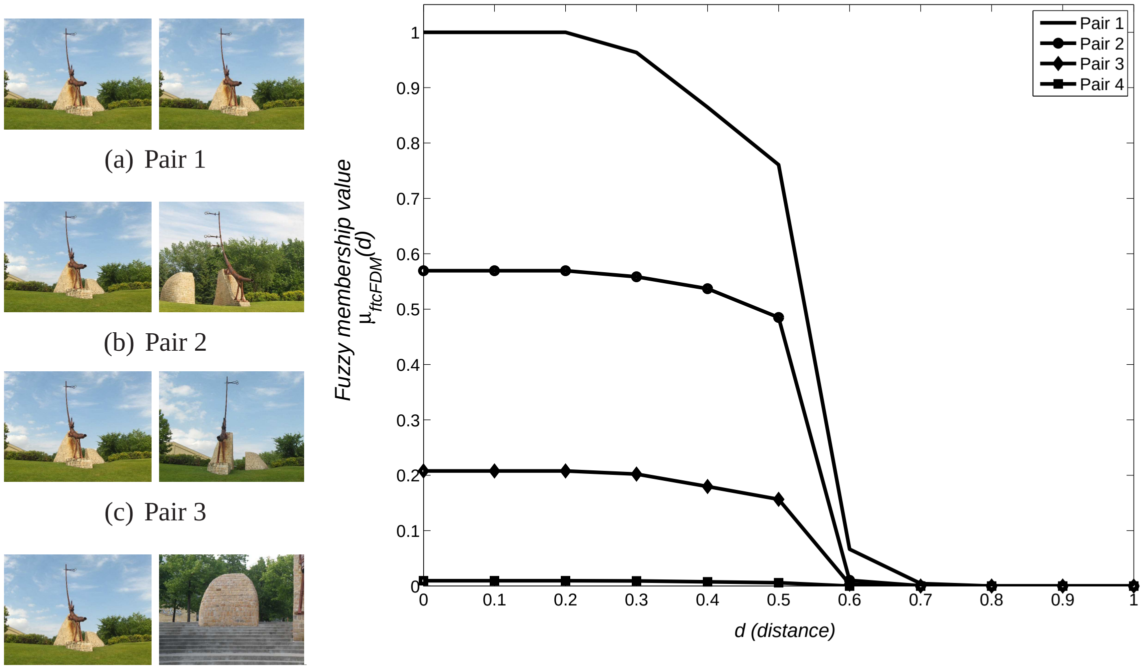
$\hat{\cong}_{\mathcal{B},\varepsilon}$	x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
x_1	1	1/3	0	0	5/9	0	0	1/9
x_2	1/3	1	1/9	5/9	7/9	0	1/3	7/9
x_3	0	1/9	1	5/9	0	7/9	7/9	1/3
x_4	0	5/9	5/9	1	1/3	1/3	7/9	7/9
y_1	5/9	7/9	0	1/3	1	0	1/9	5/9
y_2	0	0	7/9	1/3	0	1	5/9	1/9
y_3	0	1/3	7/9	7/9	1/9	5/9	1	5/9
y_4	1/9	7/9	1/3	7/9	5/9	1/9	5/9	1

(c) Fuzzy tolerance relation



(d) Membership function of the fuzzy valued distance $ftcFDM$

Figure 3.7: An example of two sets of elements (a), the distance metric ρ between elements (b) and fuzzy tolerance relation $\hat{\cong}_{\mathcal{B},\varepsilon}$ on $X \cup Y$ (c) in Example 3.3



(a) Pair 1

(b) Pair 2

(c) Pair 3

(d) Pair 4

(e) Fuzzy valued distances

Figure 3.8: Fuzzy membership functions of the fuzzy valued distances between each pair of given images

Chapter 4

Feature Extraction and Visual Description of Describable Objects

Describing a describable object is possible through a set of probe functions that create a feature vector describing the object. An important question remains that which visual descriptions (features) are important and how to properly extract such features (the choice of probe functions). A proper answer to this question is highly subjective and depends on the application. This chapter deals with the methodologies and algorithms for extraction of some visual features such as color, texture and edge information. The main focus of this thesis, however, is not on feature extraction but rather on new methodologies to accurately define similarity between sets of describable objects (here, subimages that represent visual elements) based on the given descriptions. The only contribution in this chapter is the method of defining visual features corresponding to edges in a subimage, as explained in Section 4.1.5.

Table 4.1: Probe functions in use: $\mathcal{B} = \{\phi_1, \phi_2, ..., \phi_{18}\}$

Probe function	Feature Type	Description
$\phi_1(x)$	Color	Average gray level of pixels
$\phi_2(x)$	Texture	Entropy of the gray level values
$\phi_3(x), \phi_4(x), \phi_5(x)$	Color	R, G and B color components
$\phi_6(x), \phi_7(x), \phi_8(x)$	Color	H, S and V color components
$\phi_9(x)$	Shape	Average intensities of edges
$\phi_{10}(x)$	Shape	Average orientation of edges
$\phi_{11}(x), \phi_{15}(x)$	Texture	Contrast
$\phi_{12}(x), \phi_{16}(x)$	Texture	Correlation
$\phi_{13}(x), \phi_{17}(x)$	Texture	Energy (Uniformity)
$\phi_{14}(x), \phi_{18}(x)$	Texture	Homogeneity

The following sections explain all visual features used in this thesis (see Table 4.1).

4.1 List of Probe Functions

The following probe functions are defined for a small subimage that contains local information about the image content. Each subimage is named a visual element and is considered here as a describable object x.

4.1.1 Average gray level value

The average of gray level values for all the pixels in a given subimage (x), is calculated as the first feature $\phi_1(x)$. In case of color images, the image is first converted into grayscale and then is used to calculate ϕ_1.

4.1.2 Information content (entropy)

Suppose there are N different gray levels possible in an image ($N = 256$ for an 8 bit digital image). Let M be the number of pixels in a subimage. For each subimage, let n_k be the number of pixels that have the k^{th} gray level. The *Shannon* entropy of each subimage is defined as follows,

$$E = -\sum_{k=1}^{N} p_k \log_2(p_k), \tag{4.1}$$

where $p_k = \frac{n_k}{M}$ is the normalized number of pixels that have a gray level value that belongs to the k^{th} level. This definition is based on a probabilistic view of the image where the gray level of the pixels is considered as a random variable. p_k is the probability of a pixel having k^{th} gray level and E (Shanon entropy) is a measure of the uncertainty associated with the random variable. A higher value of entropy indicates more information content in the subimage. $\phi_2(x)$ is used here to represent the entropy of a subimage x.

4.1.3 Color features

$\phi_3(x), \phi_4(x), \phi_5(x)$ are the average *Red*, *Green* and *Blue* color components of the pixels, respectively. $\phi_6(x), \phi_7(x), \phi_8(x)$ are the average *Hue*, *Saturation* and *Intensity* color components of the pixels. RGB color components exist in digital images and HSI components can be obtained using the proper transformations (see [19]).

4.1.4 Texture and statistical features

Texture features in this thesis are defined based on a statistical approach using the gray-level co-occurrence matrix (GLCM) ([23], see also [22, 44]). GLCM (also named graytone spatial-dependence matrix) can be used to define 14 different textural measures [23]. Elements of GLCM consist of the relative frequencies with which two neighboring pixels (separated by a given offset Δx and Δy in pixels) with gray level values i and j, occur in the image [39].

$$C_{\Delta x,\Delta y}(i,j) = \begin{cases} \sum_{p=1}^{n} \sum_{q=1}^{m} 1, & \text{if } I(p,q) = i \text{ and } \; I(p + \Delta\ x, q + \Delta\ y) = j \\ = 0, & \text{otherwise} \end{cases}$$

For notational convenience and with the understanding that the offset $(\Delta x, \Delta y)$ is known, let $p_{i,j}$ denote (i,j)th element of the normalized GLCM matrix ($p_{ij} = C_{\Delta x,\Delta y}(i,j)/(m \times n)$). The following texture features are used in this thesis based on GLCM.

- $\phi_{11}(x)$ (Contrast): Intensity contrast between a pixel and its neighbors (element difference moment of order 2) [19], [26]

$$\sum_i \sum_j (i-j)^k p_{i,j} \tag{4.2}$$

- $\phi_{12}(x)$ (Correlation): Correlation between a pixel and its neighbors is defined ([64]) as

$$\sum_i \sum_j \frac{(ij)p_{i,j} - \mu_x \mu_y}{\sigma_x \sigma_y}. \tag{4.3}$$

where

$\mu_x = \sum\limits_i \sum\limits_j i \cdot p_{i,j}$, $\mu_y = \sum\limits_i \sum\limits_j j \cdot p_{i,j}$
$\sigma_x = \sum\limits_i \sum\limits_j (i-\mu_x)^2 \cdot p_{i,j} \quad \sigma_y = \sum\limits_i \sum\limits_j (j-\mu_y)^2 \cdot p_{i,j}$

- $\phi_{13}(x)$ (Energy/Uniformity) and $\phi_{14}(x)$ (Homogeneity) are defined as ([19, 26])

$$\text{Energy} = \sum_{i,j} p_{ij}^2, \tag{4.4}$$

$$\text{Homogeneity} = \sum_i \sum_j \frac{p_{ij}}{1 + |i - j|}. \tag{4.5}$$

4.1.5 Edge features

Edges contain important information about the shape of objects in an image. There are many methods for detecting the intensity and orientation of a potential edge at each pixel of an image (for example, Canny method [9] or compass edge detectors [59, 60]). Here, the wavelet-based edge detection method by Mallat [37] is used. This method can be used with different wavelet functions at different scales to extract both the intensity and orientations of edges at different levels of details. In each pixel location (x, y), edge intensity and orientation at a given scale (2^j) are defined respectively as follows,

$$M_{2^j}(x, y) = \sqrt{\mid W_{2^j}^1 f_{x,y} \mid^2 + \mid W_{2^j}^2 f_{x,y} \mid^2}, \tag{4.6}$$

$$A_{2^j}(x, y) = \arctan \frac{W_{2^j}^1 f_{x,y}}{W_{2^j}^2 f_{x,y}}. \tag{4.7}$$

where $W_{2^j}^1 f_{x,y}$ and $W_{2^j}^2 f_{x,y}$ are 2-D discrete wavelet transforms of $f_{x,y}$ at each scale 2^j. Edge informations were obtained here using MATLAB wavelet toolbox.

In this thesis, the highest value of edge intensity and its corresponding orientation value (in radians) in each subimage x are chosen as visual features and shown with $\phi_9(x)$ and $\phi_{10}(x)$, respectively. Fig. 4.2 for example, shows an image, a sample subimage in the image and

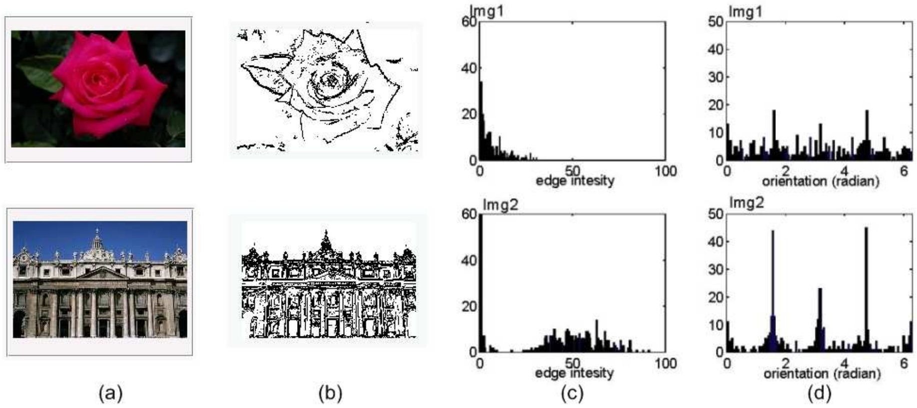
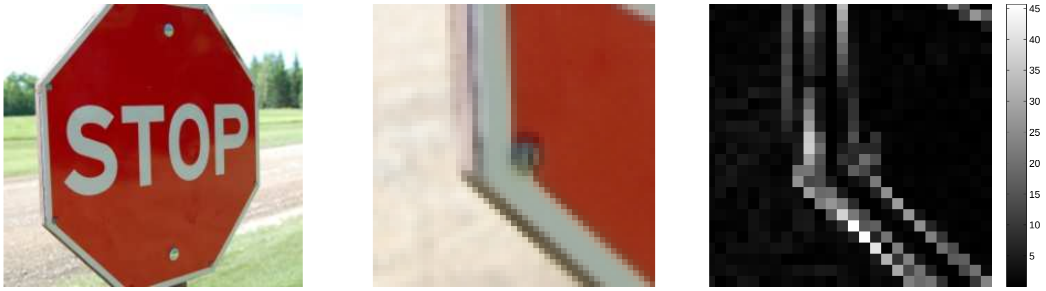


Figure 4.1: Sample images (a), edges after proper thresholding (b), histogram of dominant edge intensity and orientation in subimages (c,d)



(a) Image (b) A subimage (c) Edge intensity

Figure 4.2: An example of an image (a), a sample visual element (b) and the intensity of edges at different pixels in a visual element (c)

the edge intensity of pixels in the subimage. Brighter gray levels represent higher intensity. $\phi_9(x)$ is then defined as the maximum intensity in this subimage and $\phi_{10}(x)$ is defined as the edge orientation of the pixel that has the highest edge intensity.

Chapter 5

Experimental Results

In this chapter, performance of the proposed image similarity/distance measures, is evaluated using experiments in content-based image retrieval. The human judgment of similarity is very subjective and depends on the problem domain. Therefore, evaluation of similarity measures is not a well defined task because there is no ground truth to compare the result with. An exception to this is the trivial requirement for similarity, taking the highest value when an image is compared to itself.

In order to have a consistent method of evaluating the measure, we need a *controlled dataset* where images have already been judged by a human and categorized into groups of similar images. The similarity measure between images within a group is expected to be higher compared to images between different groups. When a particular given image is compared with a set to find those that are similar, we use the term *query image* for the given image of interest and the term *test image* to refer to an image in the set.

5.1 Content-Based Image Retrieval (CBIR)

CBIR is an important application of image similarity measures. In a traditional *text-based* image retrieval system, search and retrieval are based on textual information attached to an image (such as keywords, labels or captions). However, in a CBIR system the search is based on the image content (*i.e.* information about the pixel values in images) to find similar images in a dataset.

In a *query by example*[1] CBIR, the query is an example image and the objective is to search in a set of test images and find those that are similar to the given query. In this thesis, we are dealing with a query by example CBIR problem. A measure is used to calculate the similarity between the query image and each test image in a data set. The images are then sorted based on their similarity to the query image.

Figure 5.1 shows a schematic of a CBIR system where a query image is compared to all test images in a dataset. The images are then sorted based on their similarity to the given query image. Due to the subjective nature of the concept of similarity, it is very hard to evaluate a CBIR system. One common approach to evaluation of a CBIR system is using a controlled test data set of images containing different subsets (target sets). Images in each subset are manually chosen to represent the same (or similar) object or concept. The subset of test images that are similar to a query image is named here as the *target set* of that query image. The retrieval result for each query image is then evaluated by counting the number of images retrieved from the target set (*i.e.* the images that belong to the same subset as the query image). The performance measures *precision* and *recall* are commonly used in CBIR

[1] Other types of query techniques are possible in general CBIR systems. Examples are *query by visual sketch* or *query by providing semantics*.

are defined as follows [43].

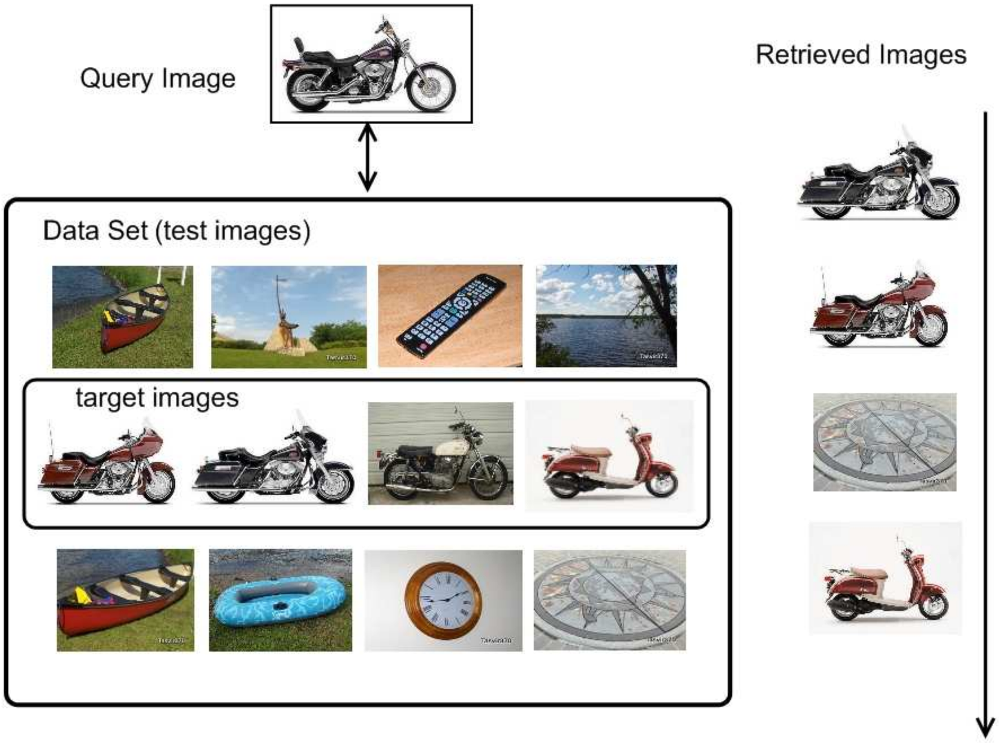


Figure 5.1: Schematic of a content-based image retrieval where a query image is compared to a set of test images.

Let q represent a query image, $A(q)$ is the set of retrieved images based on the relevance to the query and $T(q)$ is the set of target images (all relevant images in the database for the given query image q). Precision $P(q)$ and recall $R(q)$ for this query image are defined in Equations (5.1) and (5.2), respectively.

$$P(q) = \frac{|A(q) \cap T(q)|}{|A(q)|}, \tag{5.1}$$

$$R(q) = \frac{|A(q) \cap T(q)|}{|T(q)|}. \tag{5.2}$$

In other words, precision $P(q)$ in (Equation 5.1) is the fraction of the retrieved images ($A(q)$) that are also in the set of target images (correctly retrieved) and recall $R(q)$ in (Equation 5.2) is defined as the fraction of the target images ($R(q)$) which have been recalled and hence exist in $A(q)$. Higher values of precision at each particular value of recall indicates more accurate retrieval. There are three possible ways of evaluating the performance. They include precision versus the number of images retrieved, recall versus the number of images retrieved and precision versus recall.

When we search for images in a dataset, it is important to know the range of variability of images. This is something that is commonly overlooked in the study of image similarity measures. An exception, is a survey paper by Smeulders *et. al* in [63] where the concept of variability is introduced by defining two types of image domains. A *narrow-domain* set of images has a limited and predictable variability in content and appearance and a *broad-domain* set of images has unlimited and unpredictable variability. For example, if a set of images consists of images of automobiles, it is an example of a *narrow-domain*. An example of a *broad-domain* can be an online image archive such as Picasa or Flickr. In this thesis, this concept has been used to define the type of CBIR problem based on the range of variability for target and test images. Herein, CBIR problems are divided into 3 categories:

- *narrow-target*, *narrow-domain* search problem

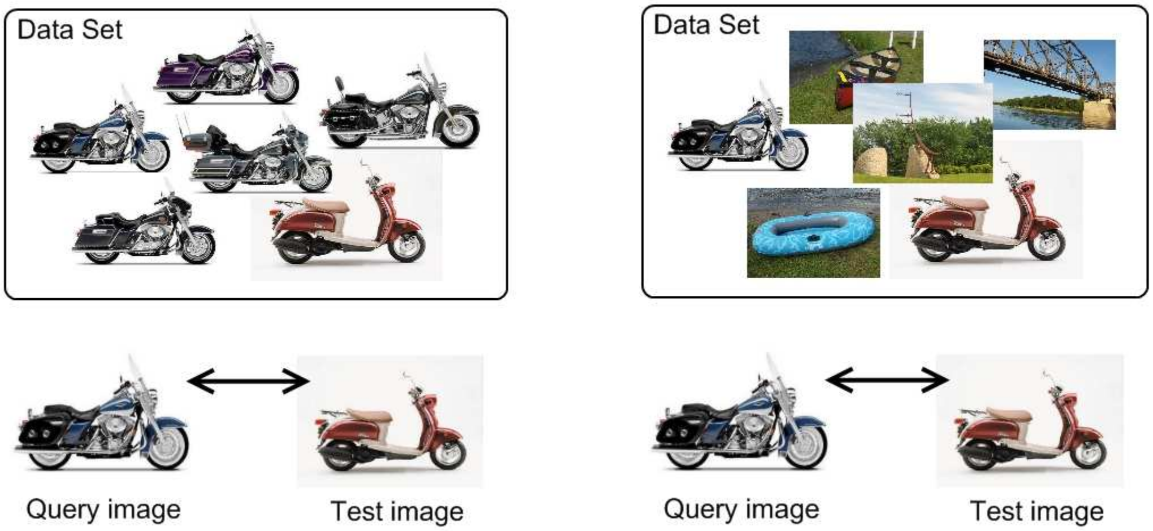
  There is little variability in both the *target images* and *test images*. For example, when the test dataset contains images of the motorcycles as seen in Fig. 5.2(a) and the target images are a subset of test images that have the same shape and model of the query image. Another example is searching in a dataset (test images) of human faces where the query image is a particular face and the target images are faces that resemble it.

- *narrow-target, broad-domain* search problem

  There is little variability in target images but the set of test images has a high variability. An example of such a search problem is searching for images of motorcycles (target images) in a dataset that contains different images randomly selected from a picture archive or the Internet (see Fig. 5.2(b)).

- *broad-target, broad-domain* search problem

  In this type of search problem both *target images* and *test images* have a relatively high variability rate. However, variability of the target images are expected to be less than the test images. An example of such a search problem is looking for images of "buildings" in an image archive of various images.



(a) A narrow-domain dataset of images

(b) A broad-domain dataset of images

Figure 5.2: Human judgment of similarity depends on the variability of the test data set.

Semantic gap

It is important to note that in a CBIR system, images are compared to each other solely based on their content (visual descriptors). However, the association between images is not always based on appearance. In other words, the *concept* behind an image may not be directly related to the visible *content* of the image. This existing gap between visual content and the semantic interpretation of the image is named *semantic gap* [63]. It is important to acknowledge this limitation in a content-based image retrieval system. In order to have a *concept-based* image retrieval system, further techniques such as *human assisted content-annotation* or *relevance feedback* [58, 82] may be useful.

In Sections 5.3 and 5.4, the performance of the proposed measures of image similarity are evaluated in two different types of CBIR problems. Each experiment consists of a number of trials. In each trial, one of the images in the dataset is considered a query image and is compared to the rest of images in the dataset. Experiments were performed using MATLAB. An executable program (POINCaRé) has also been developed with a GUI for simple CBIR experiments (see Appendix C).

5.2 Target Discrimination Matrix (TDM): A new measure for evaluation of image retrieval and classification

Precision and recall have been commonly used in CBIR as standard measures of evaluating data retrieval. In this approach, a retrieved image either belongs to the set of target images (and thus relevant to the query) or is considered irrelevant. This explains why the retrieval accuracy may change significantly if one extends the set of target images to a larger subset of test images. In other words, precision and recall only represent how the "target" images are being retrieved and hence a measure of separability between target sets and the rest of the data set. However, if the image dataset consists of different classes of similar images, it is important to know which target sets have greater similarity to each other.

In this section a new method is proposed to measure how a similarity measure may discriminate between different sets of target images in an image database.

Let $\mathcal{I} = \{I_1, ..., I_N\}$ be a set of test images in a database. Let $C_1, C_2, ..., C_m$ be m different subsets of I such that $C_i \cap C_j = \emptyset \;\forall i, j$ where each C_i consists of visually similar images that represent the same object or concept. The similarity matrix $S = s_{ij} = s(I_i, I_j)$ can be defined as the similarity between images I_i and I_j calculated using a similarity measure such as $tcNM$. A distance matrix can also be defined accordingly.

A reliable similarity measure is expected to have a high value for pairs of images that belong to the same set (within a target set) and low value for pairs of images selected from two different sets (between target sets). In order to quantify this in a statistically meaningful manner, it is proposed here to use a statistical test to compare all the numerical values of distances that belong to the same or different target sets. the Target Discrimination Matrix

(TDM) is introduced here as follows:

Let $C_i, C_j \subset \mathcal{I}$ be sets of target images. Each target set may represent a class of images that have been previously manually classified and can be considered *similar* to each other. Let

$$S_{i,j} = \{s(I_i, I_j) \quad | I_i \in C_i, I_j \in C_j\}$$

be the set of all similarity measures calculated between images in a target set and

$$S_i = \{s(I_p, I_q) \quad | I_p, I_q \in C_i\}$$

is calculated between images from different target sets. The *Welch's t-test* between the two sets of samples (distances) starts by calculating the t parameter as follows [75].

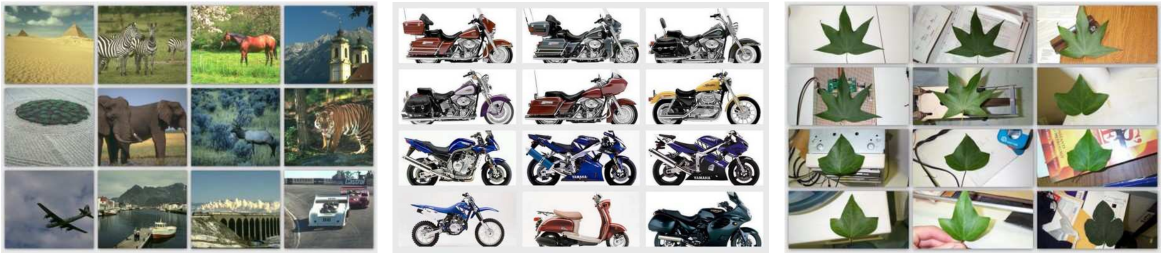
$$t = t(S_i, S_{i,j}) = \frac{\bar{S_i} - \bar{S_{i,j}}}{\sqrt{(\frac{\sigma^2(S_i)}{|S_i|} + \frac{\sigma^2(S_{i,j})}{|S_{i,j}|})}} \tag{5.3}$$

Where $\bar{S_i}$ is the mean (average) of all the elements in S_i and $|S_i|$ represents the number of elements (similarity measures) in S_i. The value of t represents the statistical significance of the difference between samples in S_i and samples in $S_{i,j}$.
the Target Discrimination Matrix (TDM) for this image database is then defined here as

$$TDM(\mathcal{I}) = [t_{i,j}]_{m \times m} \quad | t_{i,j} = t(S_i, S_{i,j}),\ i, j \in \{1, 2, ..., m\} \tag{5.4}$$

where each element of the matrix ($t_{i,j}$) is the output of the *t-test* between the samples in S_i and $S_{i,j}$. Therefore, $t_{i,j}$ represents the significance level of the difference between the similarity of pairs of images selected from the same target C_i and from two separate targets C_i,C_j. The values of the elements of this matrix are 0 (no significant difference) along the diagonal of the matrix and the value of all other elements are expected to be as high as possible.

Example 5.1. *The role of the proposed TDM matrix in demonstrating the difference between classes of similar images is shown here through an experiment. A set of 300 images are considered that consists of 3 different groups (subsets) T_1, T_2 and T_3. The first subset (T_1) consists of 100 randomly selected images from "The Berkeley Segmentation Dataset and Benchmark" images [40] (available online). The second and third subsets of images (T_2 and T_3) consist of 100 images randomly selected from "motorbike" and "leaves" image datasets, respectively. These images are available online at the Computational Vision Laboratory at Caltech* [2]. *Samples of images from subsets T_1, T_2 and T_3 are shown in Fig. 5.3. Images in subset T_1 are pictures of various objects or scenes with a high degree of variability. Subsets T_2 and T_3 are all images of motorbikes and leaves respectively and hence have less variability compared to T_1. All images are 400 pixels wide and the height of images varies between 220 to 330 pixels. Square subimages of size 40×40 pixels with 50% overlap have been used. Color, texture and edge feautures all have been included (see Table 4.1). The experiment consists of comparing each image to the rest of the images and hence $\binom{300}{2}$ pairs of images are*

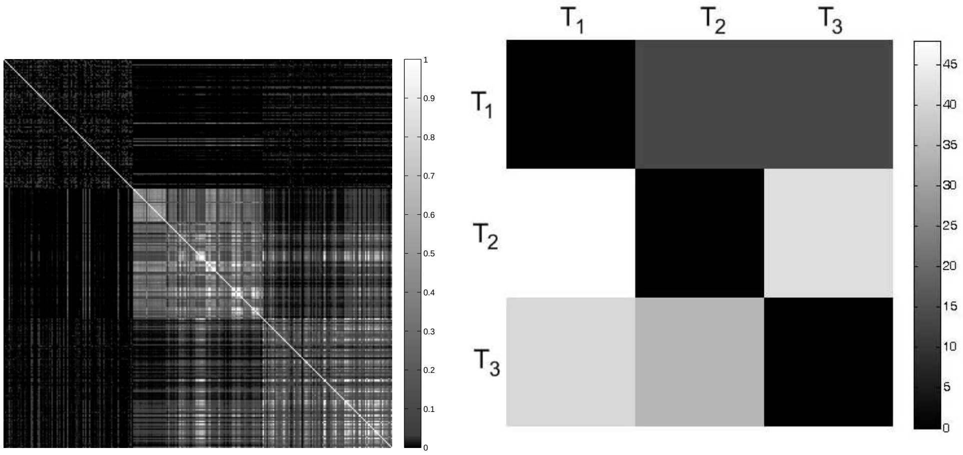


(a) T_1 (b) T_2 (c) T_3

Figure 5.3: An example of a dataset of images containing three different target sets T_1 to T_3

[2] http://www.vision.caltech.edu/archive.html

compared to each other and the value of the $tcNM$ similarity measure has been calculated for each pair according to Equation 2.13 in Section 2.2.1. The similarity matrix is shown in Fig. 5.4(a). There are 3 sets of target images (set T_1,T_2 and T_3) and the Welch's t-test *has been performed between samples of similarity measures corresponding to images from the same or different target sets. Figure 5.4(b) is a graphical representation of the TDM matrix*



(a) Similarity Matrix

(b) TDM matrix

Figure 5.4: A graphical representation of a sample similarity matrix and TDM matrix for its 3 target sets

values. Darker colors represent lower values and brighter colors represent higher values. The figure clearly shows that the difference between the target set T_2 (corresponding to the second row in the TDM matrix) and the other two target sets is more significant.

5.3 Experiment 1: SIMPLIcity Broad-Domain, Broad-Target Dataset

The SIMPLIcity 1000 test images dataset [1, 74] (available for download from [1]) is used here in a broad-domain, broad-target CBIR experiment. This is a controlled test dataset and images are numbered between 0 to 999 and divided into 10 conceptually different categories (named here as target sets C0 to C9). Figure 5.5 displays the first 8 images in each category. Images are 384×256 pixels (dimensions). Any image from the dataset can be selected as a query image and compared to all images in the dataset.

The experiment consists of calculating the similarity measures between each query image and all 1000 test images, resulting in 1,000,000 trials of image comparison. Subsequently, the images will be sorted based on their similarity to the query image. The experiment is performed using each one of the proposed similarity measures in previous chapters. The list of probe functions (visual features) in use is shown in Table 4.1 and is explained in Chapter 4. The size of subimages is 20×20 pixels. The value of ε has been selected automatically for each query image using the method discussed in Section 2.2.2. All feature values have been normalized and scaled between 0 and 1. In addition, for each of the 10 categories a feature selection algorithm is performed as explained in Section 5.3.1 and the results are calculated with an optimal subset of features.

Precision and recall have been calculated for each image in the database (chosen as a query) and the values have been averaged among all queries. Three different methods were used in this experiment to evaluate accuracy of the image retrieval. In the first evaluation method, P_{20} (defined as the precision of the twenty most similar images) was calculated

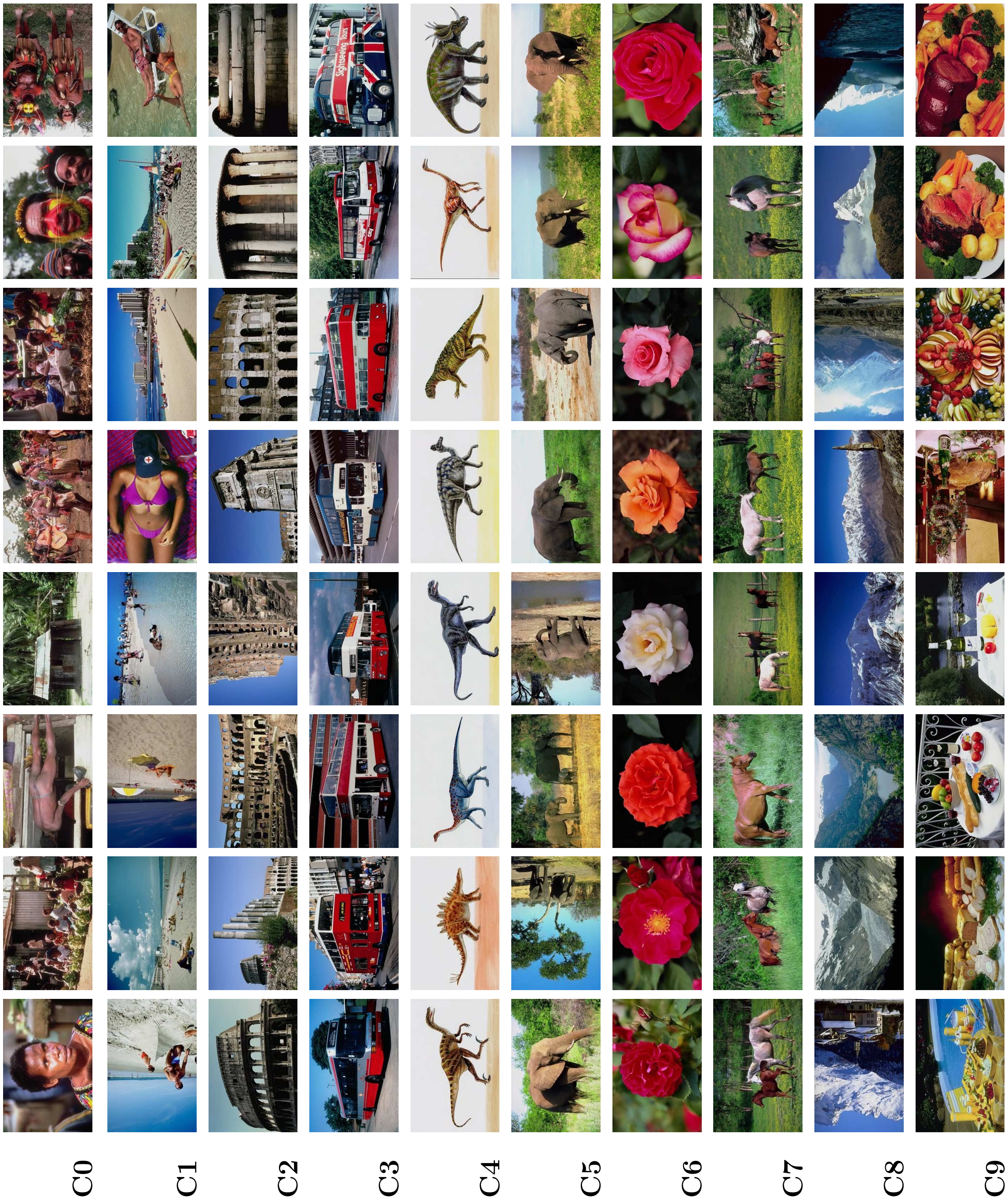


Figure 5.5: Sample images from 10 different categories (target sets) in the SIMPLIcity image dataset

for each query image and the average P_{20} for all query images is used as a measure of accuracy. Table 5.1 shows the average P_{20} precision for all similarity measures (without feature selection) compared with the results published in [27, 29] and [33]. The tolerance based methods $tcNM$, $ftcNM$ and $ftcFNM$ are shown to have better performances. Also, a feature selection method is used in each category and the precision results for $ftcFNM$ are compared to those published in [33] (see Table 5.3).

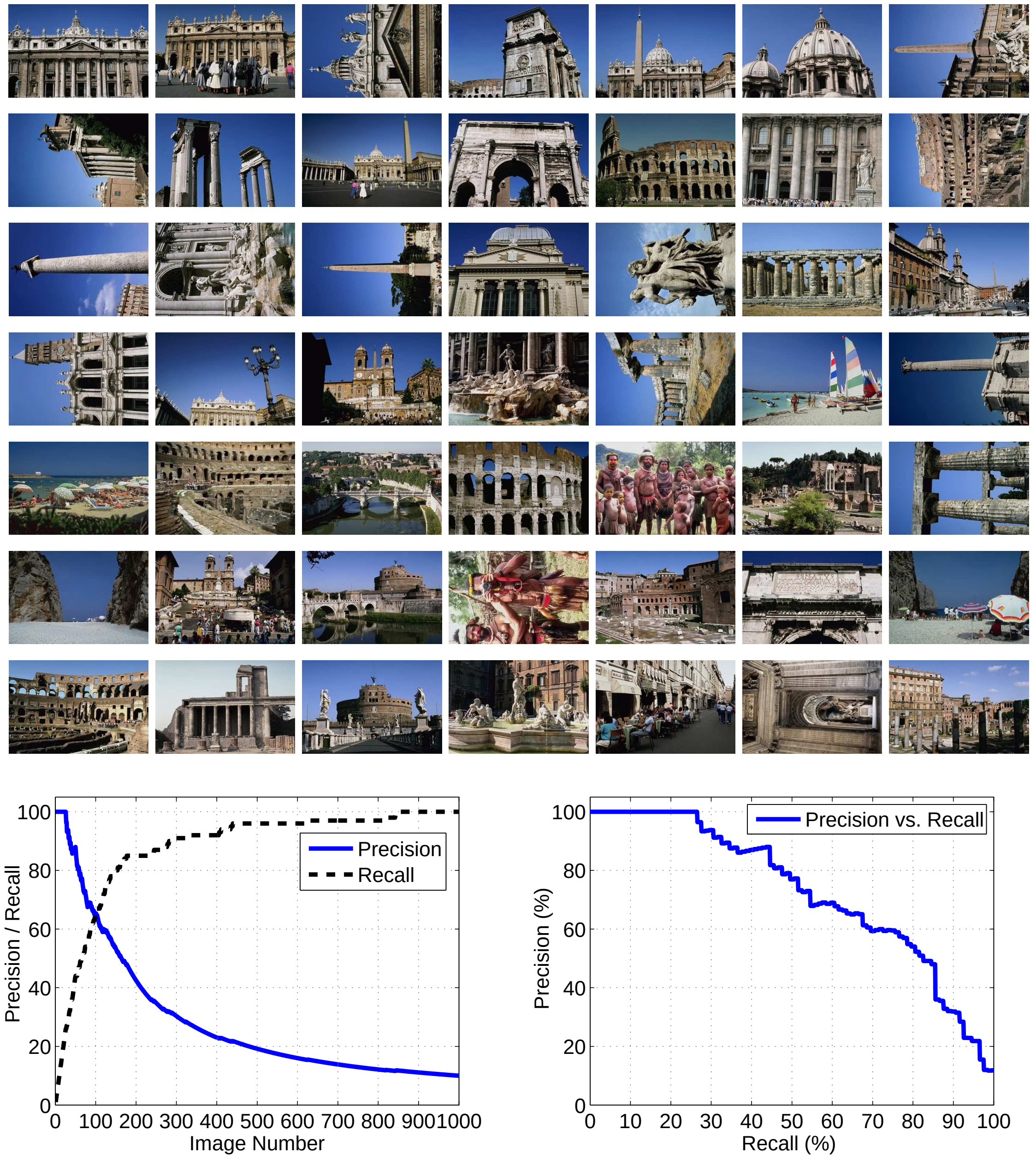
In a second method of retrieval evaluation, both precision and recall were calculated at each number of the k most similar images and the values of precision were plotted against recall. This method is more informative than using the average P_{20} measure because it evaluates the retrieval performance for all relevant images and not just the first 20.

Figure 5.6 shows an example query image (top left) and the 49 most similar images to the given query (sorted) based on $tcNM$ similarity measure. Precision, recall and precision versus recall in this example have been plotted in Fig. 5.6(a) and 5.6(b), respectively.

However, in order to properly evaluate these methods, the results are then averaged among all query images in order to demonstrate overall performance regardless of the chosen query image. Average precision-recall plots for each category have been calculated and plotted in Fig. 5.7 to Fig. 5.11. Vertical bars on each plot represent the standard deviation of variation of precision at each recall rate. Note that the values are not normally distributed and a few outliers contribute to the high values of standard deviation.

5.3.1 Feature Selection

Eighteen different probe functions have been defined here to extract visual features of subimages (visual elements) in an image. Probe functions and their corresponding visual features



(a) Precision and Recall

(b) Precision versus Recall

Figure 5.6: An example query image (top left corner), the most similar images and Precision-Recall plots

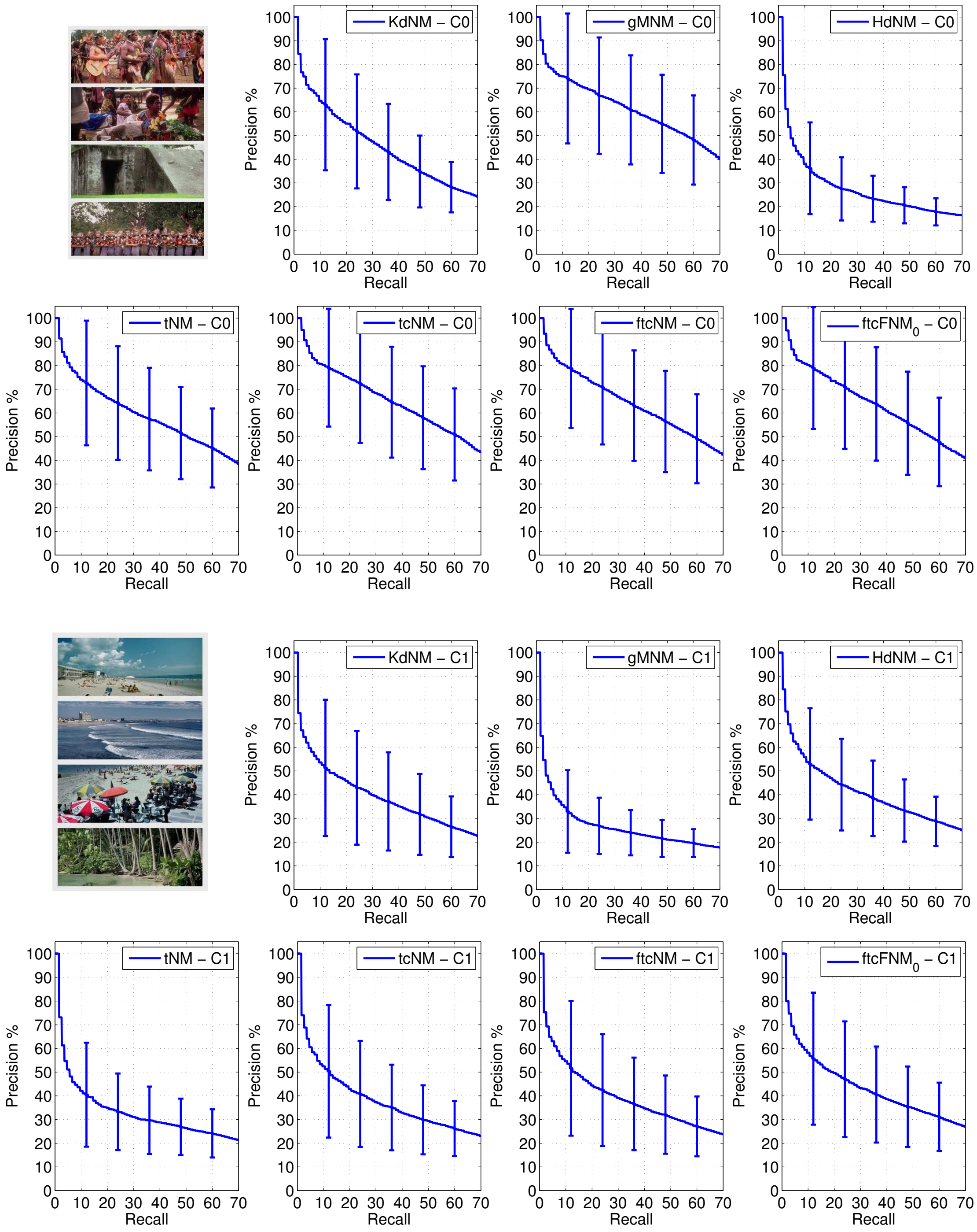


Figure 5.7: Average Precision-Recall plots for images in target sets C0 and C1.

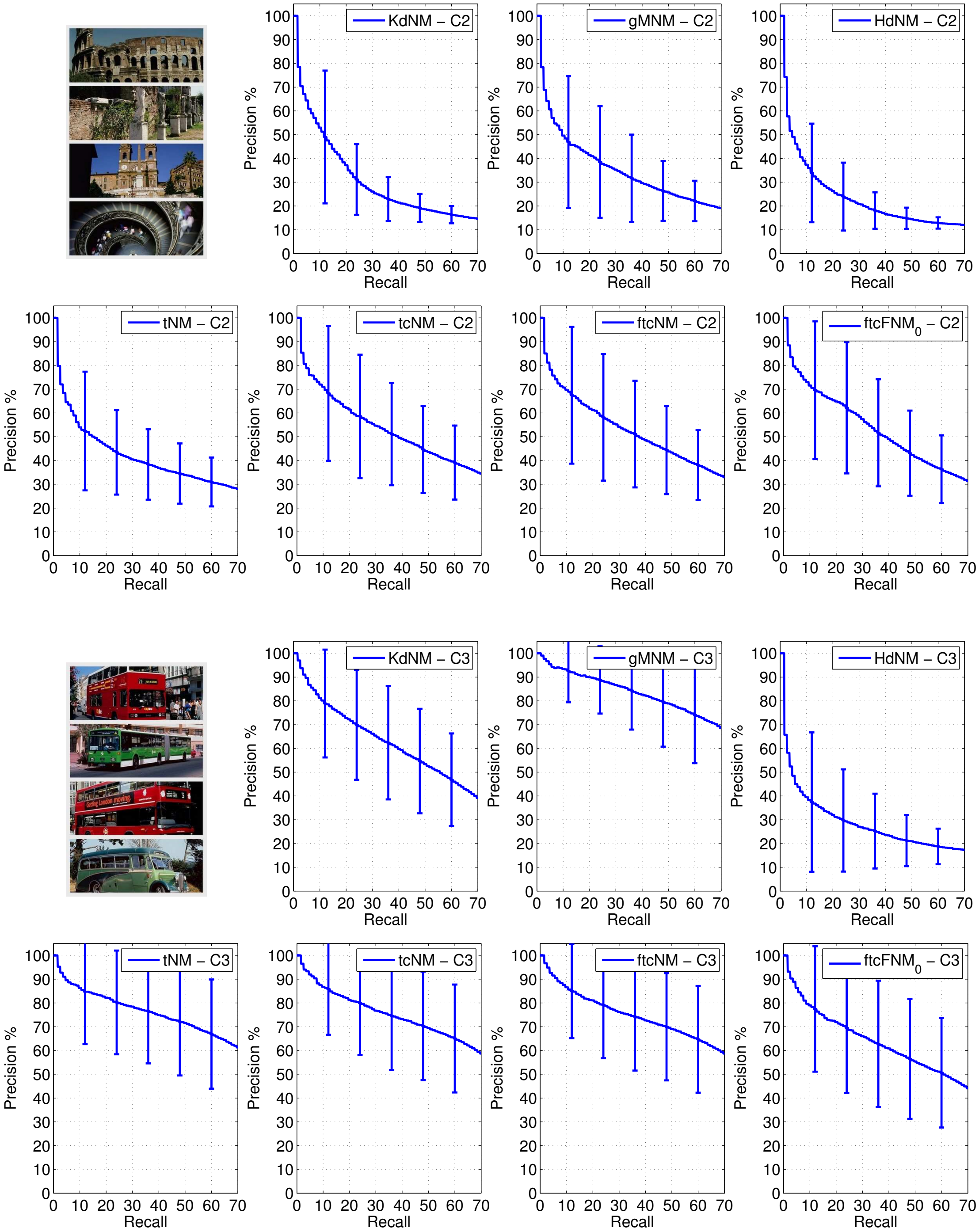


Figure 5.8: Average Precision-Recall plots for images in target sets C2 and C3.

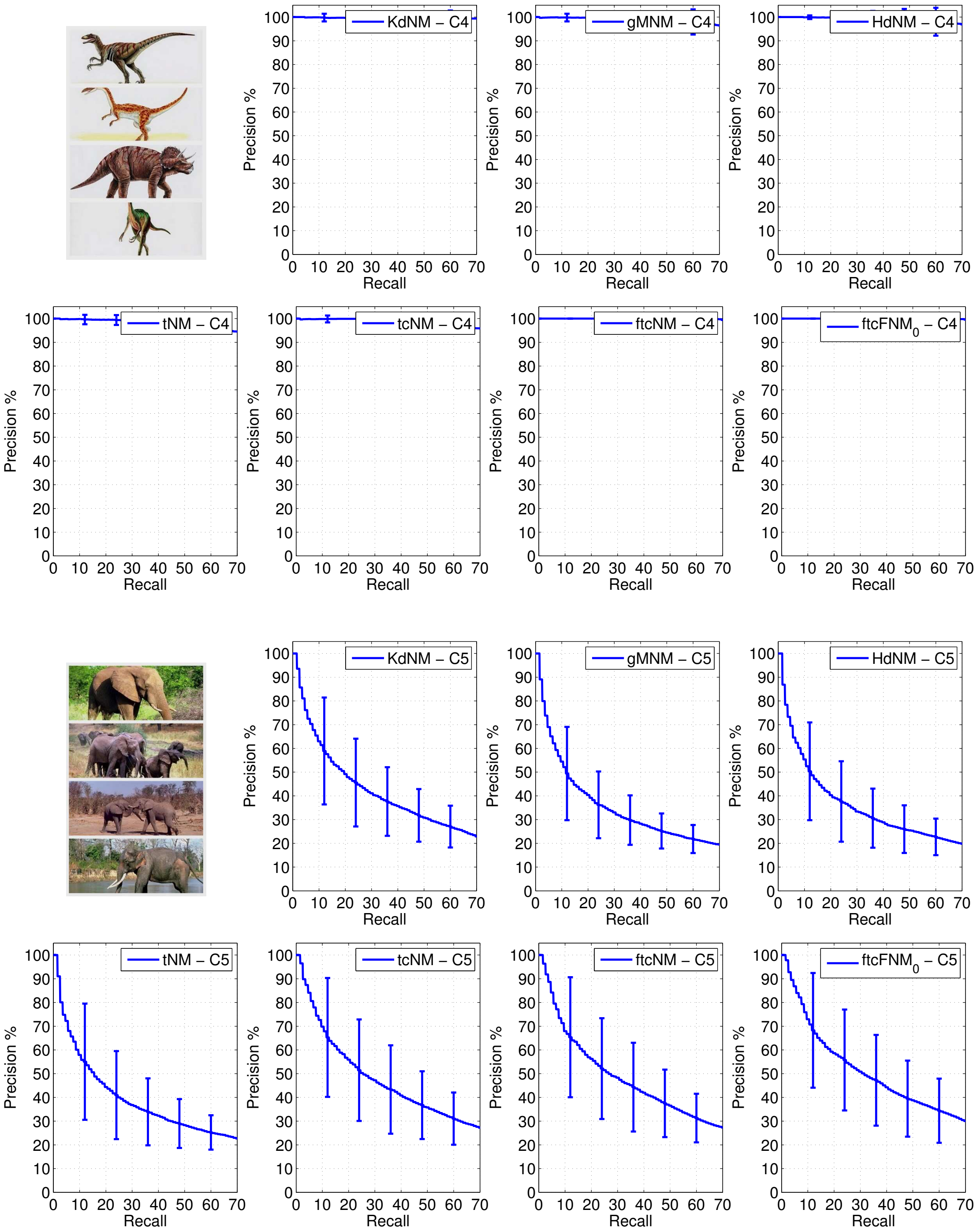


Figure 5.9: Average Precision-Recall plots for images in target sets C4 and C5.

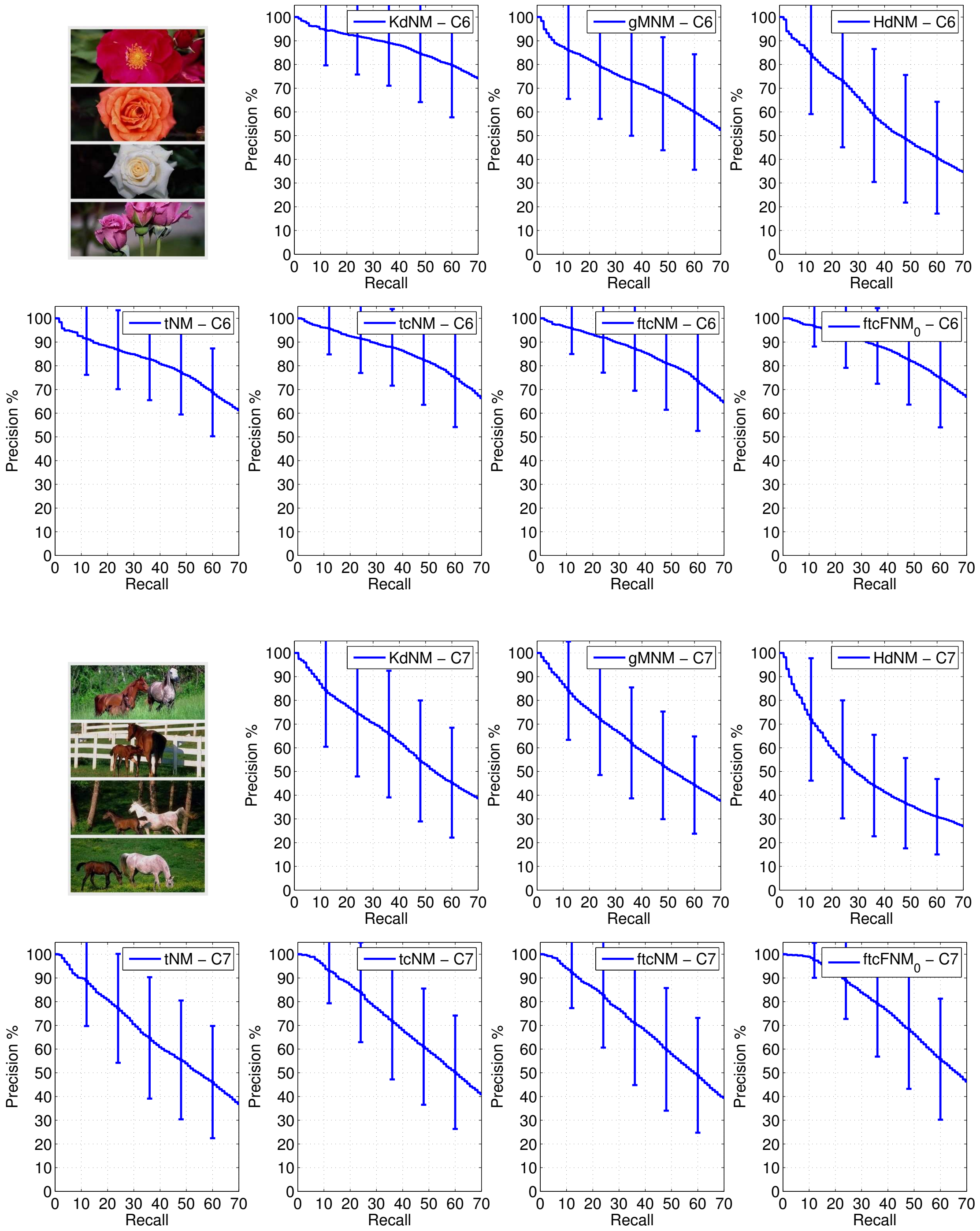


Figure 5.10: Average Precision-Recall plots for images in target sets C6 and C7.

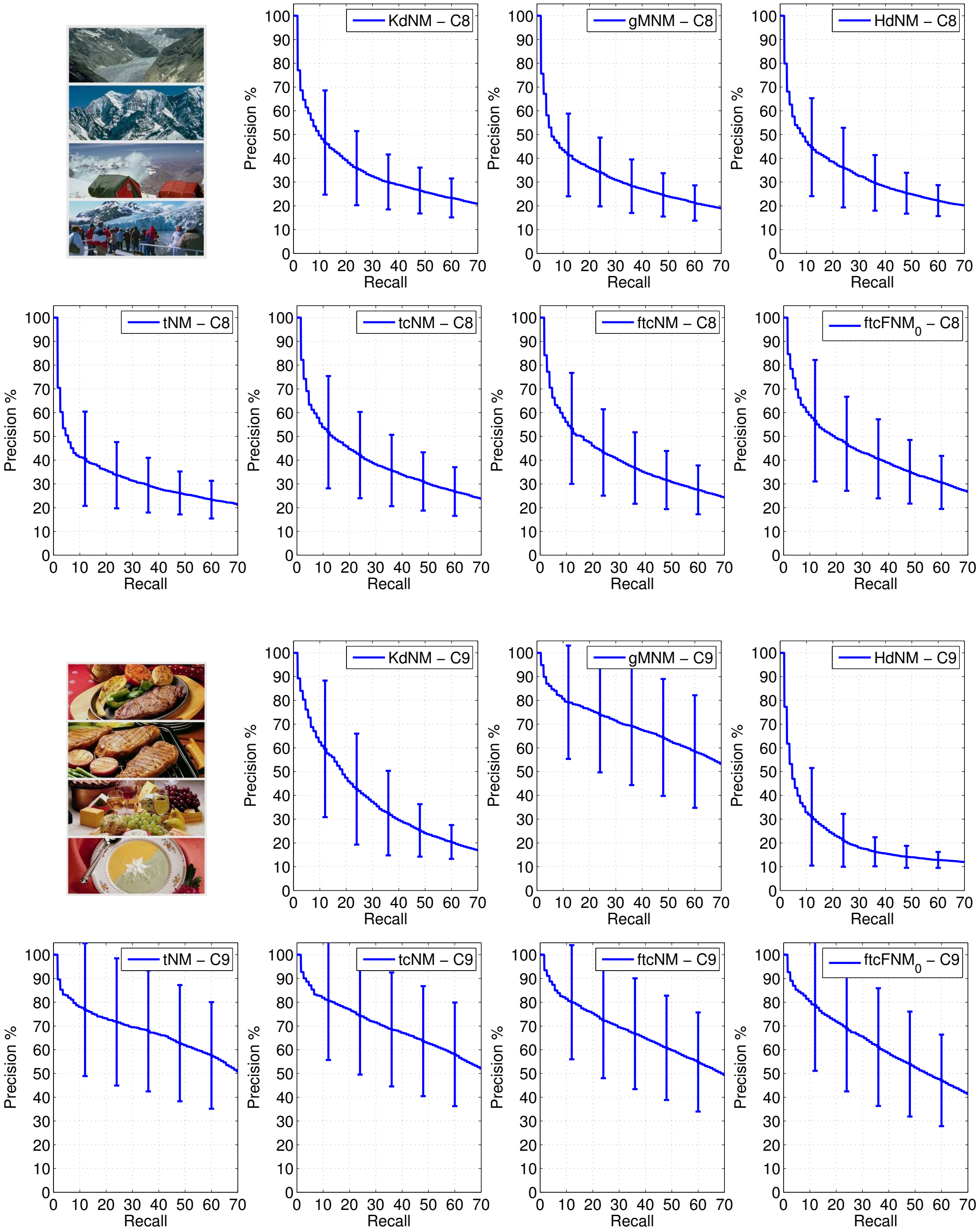
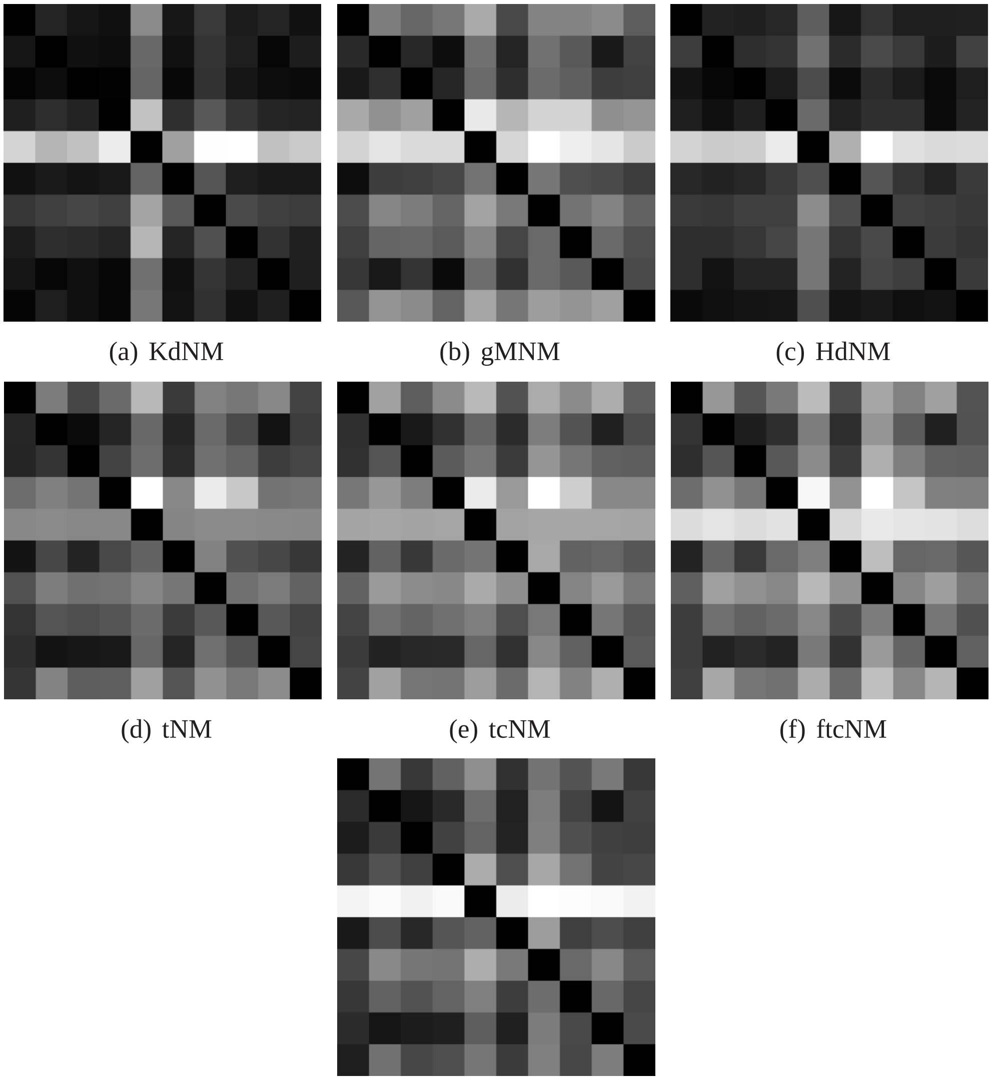


Figure 5.11: Average Precision-Recall plots for images in target sets C8 and C9.



(a) KdNM

(b) gMNM

(c) HdNM

(d) tNM

(e) tcNM

(f) ftcNM

(g) ftcFNM

Figure 5.12: A visual representation of the TDM for 10 target sets in Simplicity data set (calculated for each similarity measure)

Table 5.1: Average precision P_{20} among all query images in each category for the seven proposed methods (feature selection not included)

		CBIR results in			CBIR results in this paper (without feature selection)						
Set		[27]	[29]	[33]	$KdNM$	tNM	$gMNM$	$HdNM$	$tcNM$	$ftcNM$	$ftcFNM_0$
C0		42.40	45.25	68.30	59.35	66.60	69.70	36.95	73.70	71.70	76.05
C1		44.55	39.75	54.00	50.05	36.85	34.40	51.70	41.20	50.10	55.45
C2		41.05	37.35	56.15	47.55	47.50	47.95	36.30	69.70	59.10	65.80
C3		85.15	74.10	88.80	75.90	57.50	89.40	38.00	62.50	81.40	74.05
C4		58.65	91.45	99.25	99.25	99.75	99.15	99.30	99.95	100.00	100.00
C5		42.55	30.40	65.80	56.80	50.25	49.15	50.50	60.60	60.65	66.10
C6		89.75	85.15	89.10	92.55	89.55	83.60	80.15	95.00	93.20	95.30
C7		58.90	56.80	80.25	81.10	83.40	79.75	67.95	89.20	87.35	94.60
C8		26.80	29.25	52.15	46.10	34.25	41.60	44.35	43.15	50.90	54.50
C9		42.65	36.95	73.25	57.00	70.55	76.80	34.50	78.75	72.75	75.05
	Avg.	53.24	52.64	**72.70**	66.57	63.62	67.15	53.97	71.38	72.72	**75.69**

are listed in Table 4.1. However, it is important to note that using all available features does not necessarily imply better retrieval results. For any experiment, a subset $\mathcal{B} \subseteq \mathcal{F}$ of probe functions can be selected and used as a sub-optimal set of features. In this thesis, a Sequential Backward Selection (SBS) algorithm (see: [2, 53, 68]) is used to find a set of sub-optimal probe functions $\mathcal{B}$ selected from all available probe functions (features) in $\mathcal{F}$. This method starts with the set of all features and leaves one out at each level. The choice of feature to be removed at each level depends on the performance function that is defined for each specific problem. It is important to note that in the present experiments, the number of possible features (eighteen) is not very high and the purpose of feature selection is not feature reduction. Features are removed at each level only if their removal increases the performance measure. The algorithm is given in Algorithm 2. The algorithm is performed on a user selected subset

Algorithm 2: Feature Selection Method

```
Input  : 1. The full set of potential probe functions $\mathcal{F}$,
         2. A real valued evaluation performance function $J(B)$.
Output: Selected suboptimal set of probe functions $\mathcal{B} \subseteq \mathcal{F}$.
initialization : $k = 0$;
 Start with the full set $B_k = \mathcal{F}$;
$\Delta J = 0$;
while $\Delta J \geq 0e$ do
    $\phi_k^* = \arg\max_{\phi_j \in B_k} (J(B_k - \{\phi_j\}))$;  Find the worst probe function and leave it out;
    $B_{k+1} = B_k - \{\phi_k^*\}$;
    $\Delta J = J(B_k - \{\phi_k^*\}) - J(B_k)$;
    $k = k + 1$;
end
$\mathcal{B} = B_{k+1}$;
```

of images as training data. In each category, 20 images (out of 100) were handpicked as representatives of images for that category. Half of those images were used for the feature selection algorithm. At each level, the precision of the twenty most similar images (P_{20}) was used as the performance measure $J(B)$ (refer to Algorithm 2) and the average was taken among all selected 10 query images. Table 5.3 shows the selected features (probe functions) for each category. These features are selected by the SBS algorithm (Algorithm 2) based on the selected subset of training images. The performance of $ftcFNM_0$ in each category using only the selected features is shown in Table 5.3 and compared with results published in [33].

Table 5.2: Selected features in each category

		ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}	ϕ_{12}	ϕ_{13}	ϕ_{14}	ϕ_{15}	ϕ_{16}	ϕ_{17}	ϕ_{18}
C0		✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
C1		-	-	-	-	✓	✓	-	-	✓	✓	-	-	-	✓	✓	✓	✓	✓
C2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-	-	-
C3		✓	✓	-	✓	-	✓	-	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
C4		-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
C5		✓	✓	-	✓	-	✓	-	✓	✓	-	✓	✓	-	-	-	-	-	-
C6		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C7		-	-	✓	✓	-	✓	✓	-	-	-	-	✓	-	-	✓	-	-	✓
C8		-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C9		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 5.3: Comparison of average precision P_{20} between the best proposed methods $tcNM$, $ftcFNM_0$ and results published in [33]

Set	Sample	Results reported in [33]	$tcNM$ selected features	$ftcFNM_0$ selected features	$ftcFNM_0$ Accuracy improvement
C0		68.30	75.48	76.05	+ 7.75
C1		54.00	51.38	63.8	+ 9.80
C2		56.15	70.95	71.05	+ 14.90
C3		88.80	85.76	80.1	- 8.70
C4		99.25	100	100	+ 0.75
C5		65.80	57.48	67.45	+ 1.65
C6		89.10	93.48	95.55	+ 6.45
C7		80.25	90.0	95.1	+ 14.85
C8		52.15	50.62	55.50	+ 3.35
C9		73.25	80.67	76.8	+ 3.55
	Avg.	72.70	75.58	78.14	+ 5.44%

5.4 Experiment 2: TasviR-3x70: A Broad-Domain-Narrow-Target Controlled Set of Test Images

In this experiment a controlled test dataset of images has been created to test similarity measures in a *broad-domain, narrow-target* CBIR problem. An HP PhotoSmart R725 point-and-shoot camera was used to take the photos ($640x480$ image size). The photos were later downsampled to 240 by 180 pixels and a subimage size of $20x20$ pixels was used to define visual elements. In order to create this database, photos were taken of different subjects ranging from objects with a clear background to natural scenes or indoor spaces. Therefore, the variability of the database is very high. However, for each subject 3 different pictures were taken, either from a different point of view or by taking a different picture in the same area. The images can be divided into 70 different categories, each representing a different subject where each category consists of 3 images that are very similar with little variability. The target set for each query image is the set of 3 images similar to the query. All proposed similarity measures in the previous chapters have been used to sort images based on their similarity to the query image. Figure 5.14 shows the average recall rate (out of 3) plotted for each target set. Figure 5.15 shows the five most similar images in the dataset for the first 20 images as an example calculated using $gMNM$, tNM and $tcNM$. the TDM matrix (70 × 70) was also calculated. As an example, Fig. 5.16 shows the values of 4^{th} row of the TDM matrix (t in the *t-test* calculated between target set 4 and all the other 70 target sets). The plots clearly show that target set 12 and 37 are less significantly different from target set 4. Sample images from these target sets are shown in Fig. 5.16 and their similarity compared to other target sets is evident.

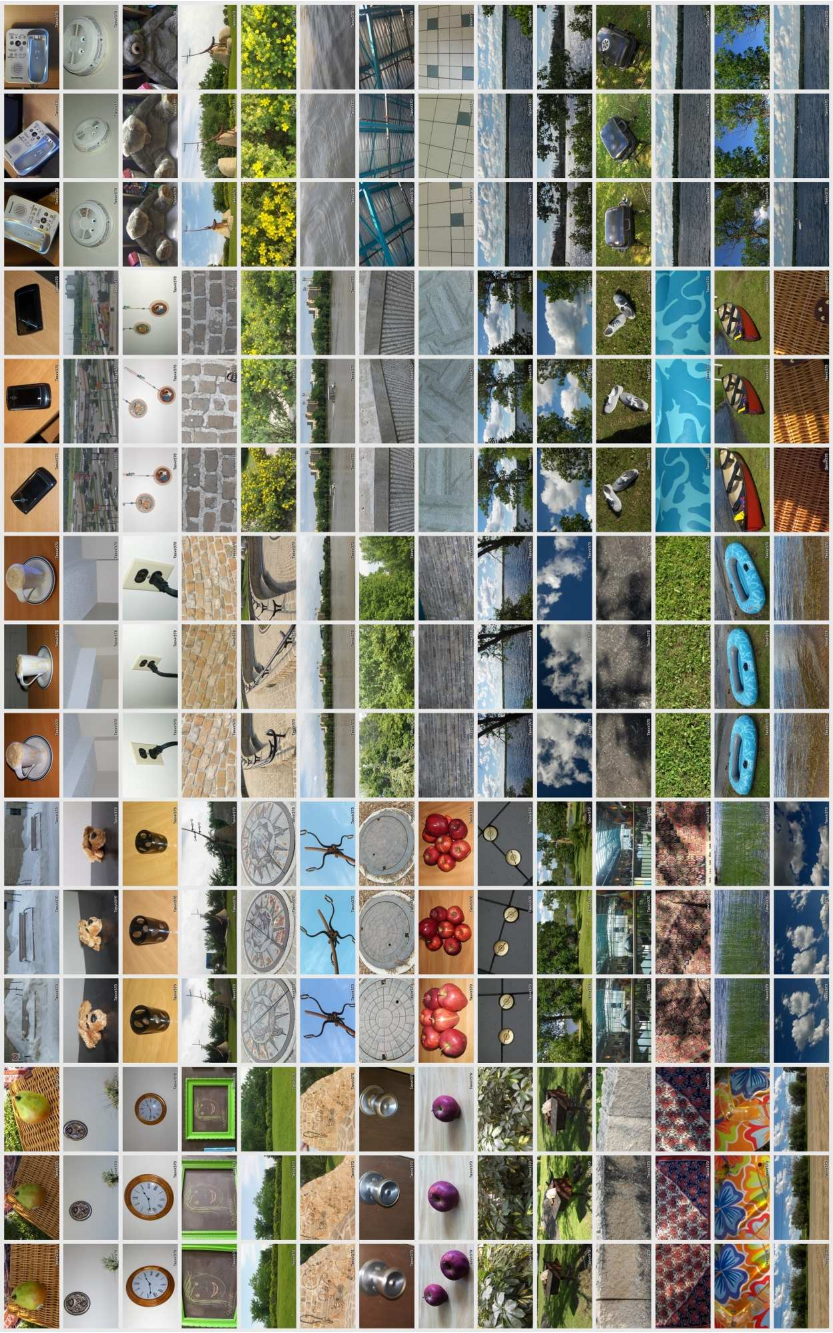
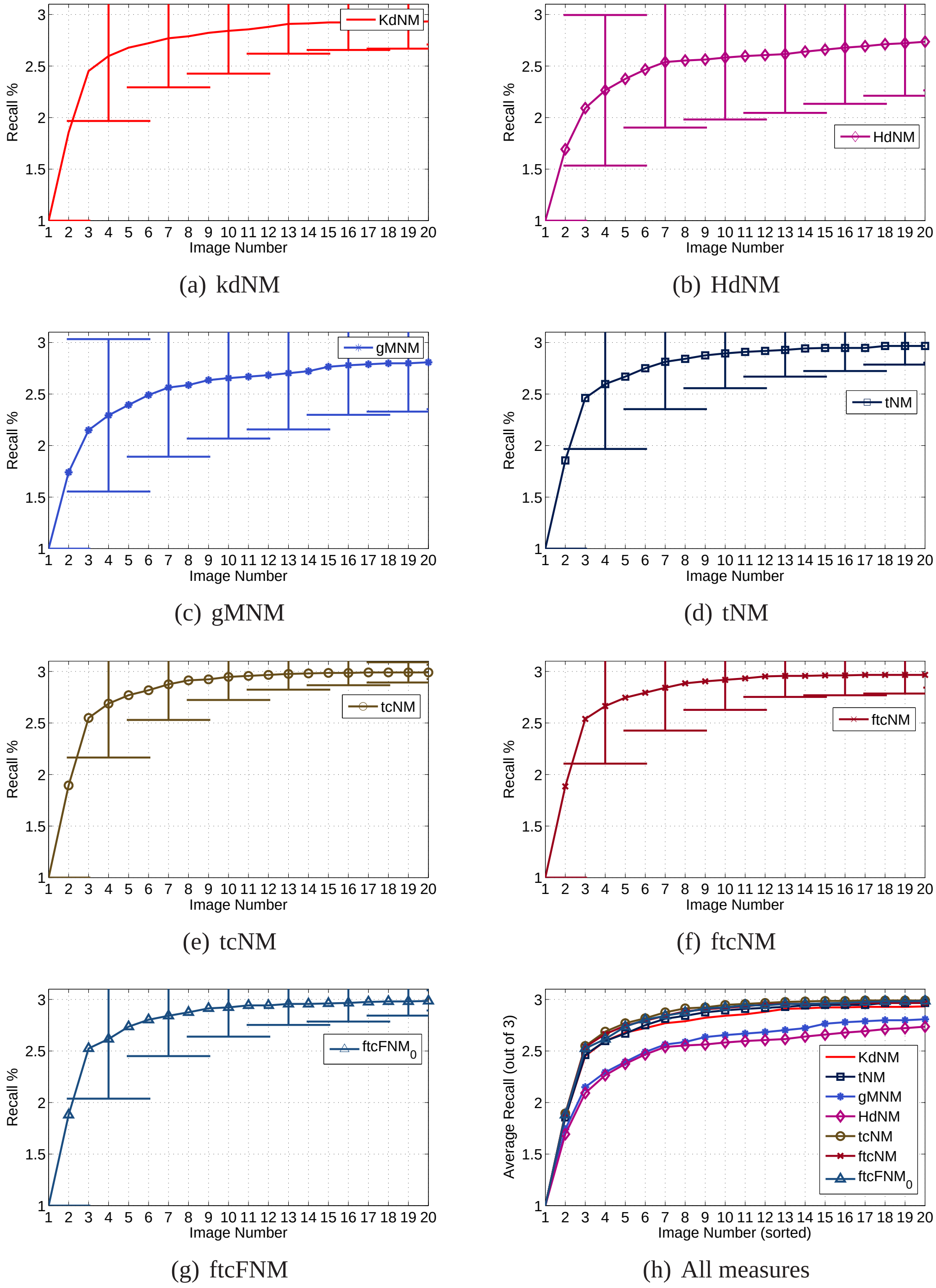


Figure 5.13: A collection of 210 images with 70 target sets (Tasvir-3x70)



(a) kdNM

(b) HdNM

(c) gMNM

(d) tNM

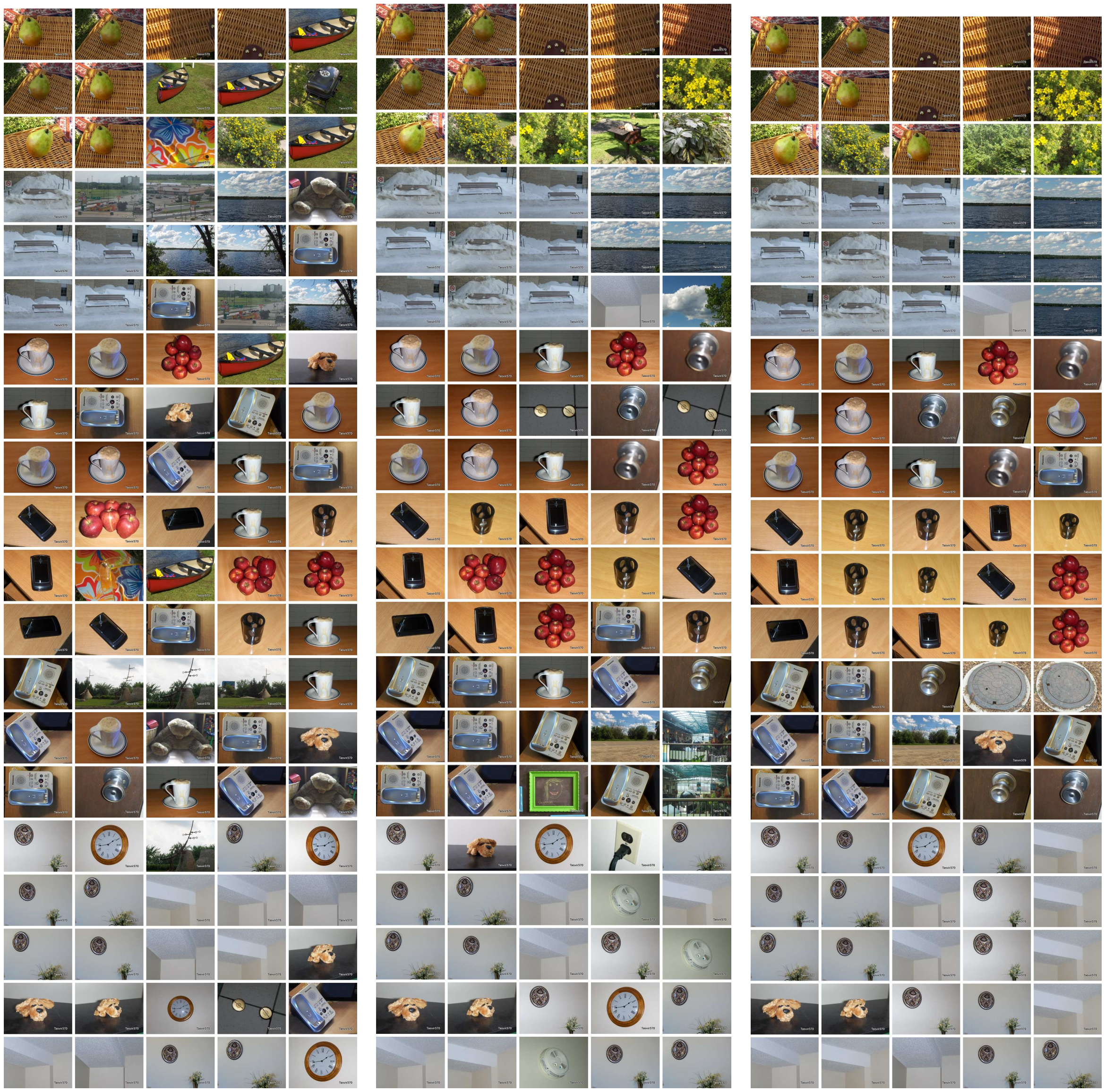
(e) tcNM

(f) ftcNM

(g) ftcFNM$_0$

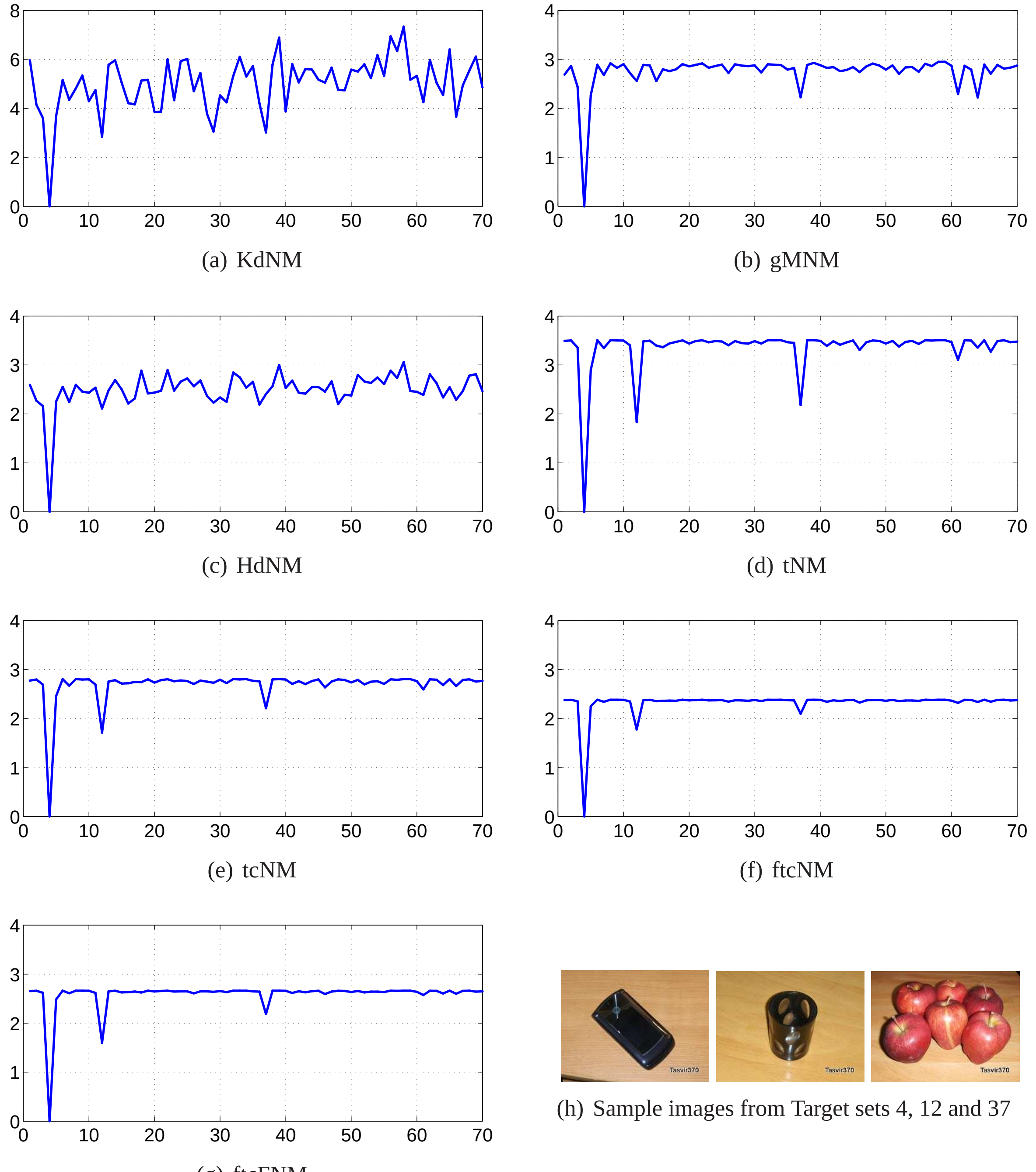
(h) All measures

Figure 5.14: Average recall among all 210 query images calculated using different measures



(a) gMNM (b) tNM (c) tcNM

Figure 5.15: Examples of the three most similar images to a query image obtained using $gMNM, tNM$ and $tcNM$. Each row corresponds to one query image.



(a) KdNM

(b) gMNM

(c) HdNM

(d) tNM

(e) tcNM

(f) ftcNM

(g) ftcFNM

(h) Sample images from Target sets 4, 12 and 37

Figure 5.16: An example of the *t-test* values versus target set number representing the significance of difference between the 4th target set and the rest of the 70 target sets in Experiment 2

5.5 Discussion of the Experimental Results

The importance and significance of the experimental results are discussed in this section. The benefits of the proposed methods based on *tolerance relations* and *fuzzy tolerance relations* (namely tNM, $tcNM$, $ftcNM$ and $ftcFNM$) will be discussed here in comparison to the classical methods that are either introduced and implemented here (namely $KdNM$, $HdNM$ and $gMNM$) or in other literature (see [27,29,33]). Some of the highlights of these experiments are as follow.

1. In all the experiments that have been implemented in this thesis, a near set approach has been adopted. This means that images are divided into visual elements (subimages) and similarity between sets of elements has been introduced and implemented. The global spatial information of images will not be used in this approach. However, this can be beneficial for searching in a broad domain target set of images.

2. Each subimage has been described using only 18 visual features corresponding to color, texture and edge information. In all the experiments, except in Section 5.3.1, (where a simple feature selection has been implemented) feature selection was not used.

3. The adaptive selection of epsilon value (ε) makes the proposed methods parametric but automatic and adaptive. The value of ε is automatically selected for each query image.

4. Experimental CBIR results in Experiment 1 on a standard *broad-domain, broad-target* dataset demonstrates a significant improvement in accuracy of the image retrieval using tolerance based near set methods compared to classical methods of distance between

sets.

5. $tcNM$ outperforms tNM.

6. Comparison of the proposed methods with other existing methods has been shown in Table 5.1 and 5.3. The performance measure that is used here is P_{20} (the precision of the twenty most similar images). This measure is used to have a side by side comparison between the new results and the ones published in [27, 29, 33]. According to this table, the new results are significantly more accurate than the two older methods published in [27, 29]. On average, the new results are also better than the most recent method published in [33].

7. It is important to note that the proposed methods in this thesis (both tolerance based methods and classical distance based methods) are completely unsupervised and require no prior information about the images in the dataset. However, as is stated in [33], their method relies on clustering the pixels of all the images in the database using a K-means clustering algorithm to be used for calculation of color histograms.

8. Using a simple feature selection method, improved the results as shown in Table 5.3. However, feature selection requires training the system with a training set of images from each category and thus it will depend on the training set. High variability of target images in this dataset makes it less robust to the choice of query image. The advantage of the proposed methods in this thesis is their competitive performance even without any feature selection in a completely unsupervised setting.

9. The proposed classical distance based measures are more promising when the target set has low variability (*e.g* target sets C4 (dinosaurs) C6 (flowers) C7 (horses) and C3

(buses) in Experiment 1). In one example, $gMNM$ outperforms all the other methods for target set C3 (buses).

10. The performance of fuzzy tolerance methods (*i.e.* $ftcFNM_0$, $ftcNM$) show an improvement over $tcNM$. However, it is important to note that the reason and the importance of introducing fuzzy tolerance relations is to increase robustness when choosing the epsilon value. The epsilon value is in fact a threshold between similarity and dissimilarity of visual elements based on their distance in feature space. Defining a fuzzy threshold (gradual transition from similar to non-similar) makes the methods more robust to changes in parameters. The automatic method of selecting this threshold (for both tolerance and fuzzy tolerance) makes it adaptive and hence also more robust to changes in the query image.

11. Plots of average *precision versus recall* in Fig. 5.7 to Fig. 5.11 show the full range of precision averaged for all the queries along with the standard deviation of changes in precision. When the target set has high variability, (broad-domain, broad-target CBIR) precision rate drastically drops at high rates of recall. This can be attributed to very high *semantic gap* between some of the target images in each category.

12. Category C4 (dinosaurs) is a very narrow-domain target set and hence can be easily retrieved in broad-domain sets of images. The precision-recall plots show that the average precision for all the methods is very close to 100%. The standard deviation of values is much lower for $ftcNM$ and $ftcFNM_0$.

13. Precision and recall plots demonstrate the performance of the methods in retrieving the target images corresponding to a query image. As long as a target image is ranked in

similarity above a non-target test image, it is not important if the difference between target and non-target is significant. The new proposed TDM matrix provides more information about the separability of target sets. This is done by calculating the statistical significance of the difference between similarity values of different subsets of a test dataset. Figure 5.12 gives more insight about this, by visualizing TDM using gray color coding. It can be seen that target sets C1 and C8 are the most challenging target sets to be retrieved. For example, if the query image is selected from category C1, the most similar results are more likely from not only target sets C1 but also C2 or C8.

14. The performance of the new methods has also been verified on a broad-domain, narrow-target CBIR experiment (Experiment 2). In this experiment, there is almost zero semantic gap between images of the same target set. However, there are occasional strong visual similarities between images that are semantically different.

Chapter 6

Conclusions and Future Work

“The whole is greater than sum of its parts”.

Different similarity measures were introduced in this research to quantify the level of similarity between two images based on their content using visual descriptions. This is an example of a task that is very challenging for computers and yet a basic task for humans. Therefore, the main motivation for this research was to explore methodologies which are inspired by the visual perception of the human mind. The exact mechanism and the neuroscientific basis of the perception especially perception of similarity is not well known. However, we may be able to use some of the intuitive models that can be hypothesized about the way information is perceived in our brain. The following list shows how different mathematical methods were used here inspired by conjectures about the human perception.

- *Visual perception has limited resolution. Objects that are close enough in terms of their visual features, are seen as almost similar*. This hypothesis (that can be easily verified) is the basis for using tolerance space theory in modeling similarity between images at an elementary level.

- *Similarity is not a black and white concept. Human perception of similarity is approximate with no sharp threshold*. This hypothesis was the basis for using fuzzy tolerance relations to improve upon tolerance relations.

- *Human perception in viewing an image is formed by grouping similar parts of the image(s) together*. This conjecture (which is based on the second principle of Gestalt theory of visual perception) [30] as well as the work of Ewa Orłowska [45, 46] is the motivation behind using tolerance classes and tolerance neighborhoods as *groups* of similar elements. Therefore, we can group similar perceptual elements based on their visual descriptions allowing for imprecision and small discrepancies between descriptions.

- *Tolerance classes and tolerance neighborhoods can both be used to group elements of an image together*. In this research, tolerance neighborhoods were used. The accompanying hypothesis is when looking at an image, the elements that are visually similar to the point of gaze will be grouped together based on their similarity to the point of focus. Therefore, the similarity between a given element and the centre is important. Also, using tolerance neighborhoods has significant computational advantages over tolerance classes.

- The purpose of using fuzzy set theory is twofold. The first reason is to eliminate the need for a sharp boundary when forming sets of *similar* elements, allowing a gradual transition between the concepts of *similar* and *dissimilar*. The second reason is to emulate how humans describe similarity between visual stimuli. An automatic com-

> puter algorithm may generate a numerical valued measure of similarity represented by a real number that reflects degree of similarity. Humans, however, can easily recognize similarity and describe it in the form of natural language statements. The fuzzy valued nearness/distance measure proposed in this research is intended to provide a more human interpretation through generating a fuzzy set instead of a numerical value.

The above methodologies were employed to introduce new similarity measures between images that are considered as sets of visual elements. The similarity measures were also tested in CBIR experiments to retrieve images based on similarity. Since the image similarity problem is subjective by nature, there is no direct way of evaluating a nearness measure. However, a content-based image retrieval task using a controlled test dataset of images was used to evaluate the performance of the similarity measures. Here, the final results of the experiments on broad-domain sets of images have an average performance that is higher than the latest published results even without feature selection. This is the main points of strength relating to CBIR in this thesis. The experiments were performed in a completely unsupervised system using some arbitrary chosen visual features such as color and texture. The methods presented in this thesis are universal. For a specific image retrieval problem, feature selection will lead to better results.

Also it is important to note that in all the experiments in this thesis, images are compared only based on their content. There are significant *semantic gaps* between some of the images in Experiment 1 which will contribute to lower precision rates at very high recall values.

6.1 Future Work

Future avenues of this research:

1. To improve upon and expand a region-based distance/similarity measure between a region of interest in one image and another. The method has already been introduced and published by the author [55] introducing a rough set-based boundary (upper and lower approximations) for the region of interest in the tolerance space. This method has not been included in this document because further experimental work is required.

2. To use proposed similarity measures for the *image classification* problem. Having a distance/similarity measure between each pair of images enables us to implement a completely unsupervised two-class classification problem that divides pairs of images into *similar* or *dissimilar* classes. One such experiment has already been presented by the author to be published in [41].

3. Development of a new form of *fuzzy metric spaces* with fuzzy membership functions that are defined as *fuzzy numbers* and studying the topology that may be induced by this new kind of fuzzy metric space.

4. Exploring partial or full ordering relations between fuzzy sets (or fuzzy numbers) used in ranking images based on fuzzy valued distances.

5. Using the proposed fuzzy valued similarity measure in a full knowledge-based system that involves online or offline human interaction.

6. To implement more visual features (probe functions) for comparing images and to incorporate more powerful feature extraction/selection methods.

7. To expand the capabilities of the image retrieval system to include concept based image retrieval.

Appendix A

Psychological Views of the Perception of Similarity

Some of the psychophysical models of similarity have inspired researchers in computer science and artificial intelligence to develop mathematical methods to measure the similarity using the visual information available in images. One of the earliest quantitative studies in psychophysics of perception goes back to 1850 when Gustav Fechner [17] hypothesized that physical quantities and their corresponding psychological experiences are mathematically related. [42]. His one dimensional logarithmic model between the physical intensity and perceived intensity was later challenged by Thurstone in 1927 [69] with a more complex statistical model that is based on discriminal differences between a pairs of stimuli. These studies were limited to the special case of a one dimensional stimuli. Visual perception however, (like many other real situations) requires a combination of stimuli from a set of separate dimensions to achieve a full perception. In 1950, Attneave published a seminal paper [5], in which he studied judgment of similarity between stimuli with multiple dimensions. He was one of the first people who proposed a view of (dis)similarity as distance in a mental space. This mental distance approach (or geometric approach) which was further elaborated by Shepard in 1962 [62], is still one of the basic assumptions used in many of the computational methods for image similarity in computer science. This approach (as was further studies in [7]), assumes a metric structure for the mental space where a similarity function has to satisfy the metric axioms namely *symmetry*, *positivity* and *triangle equality*.

However, metric assumptions have been challenged by later approaches such as featural approach by Tversky in 1977 [70, 71] or recent transformational approach [21]. The *contrast model* as proposed by Tversky assumes a feature matching process for describing similarity rather than a geometric representation. Tversky's model can be viewed [61] as a set-theoretic model in which stimuli has binary features or attributes. Degree of similarity is then defined based on the linear function of their common features. A survey of similarity models from a psychological and computer-science point of view can be found in [42] and [61], respectively. In the present research, the so called mental distance approach is used only to describe the similarity /distance between visual elements while the overall distance between images is defined using a Near Set approach.

Appendix B

Mathematical Proofs

Lemma 1. Absolute value of a random variable

If X is a continuous random variable with probability density function (`pdf`) defined as $f_X(x)$ and (cumulative) distribution function $F_X(x)$, then $|X|$ is another random variable with the following probability distribution function:

$$\begin{aligned} f_{|X|}(x) &= f_X(x) + f_X(-x) && \textit{iff} \quad x \geq 0 \\ f_{|X|}(x) &= 0 && \textit{iff} \quad x < 0 \end{aligned}$$

Proof

Starting with CDF of $|X|$ and $\forall x > 0$

$$F_{|X|}(x) = Pr\left(|X| \leq x\right) = Pr\left(-x \leq X \leq x\right) = F_X(x) - F_X(-x)$$

Taking derivative of both sides

$$f_{|X|}(x) = f_X(x) - (-1) \times f_X(-x) = f_X(x) + f_X(-x) \tag{B.1}$$

$\forall x \leq 0,\ Pr\left(|X| < x\right) = 0$ and hence $F_{|X|}(x) = f_{|X|}(x) = 0$. Lemma is proved. ■

Lemma 2. *If X is a random variable with probability density function* `(pdf)` *defined as $f_X(x)$ and (cumulative) distribution function $F_X(x)$, and if a is a constant, then $X - a$ is another random variable with the following probability distribution function:*

$$f_{(X-a)}(x) = f_X(x + a) \tag{B.2}$$

Proof

Starting with CDF,

$$F_{(X-a)}(x) = Pr\left((X - a) \leq x\right) = Pr\left((X \leq x + a\right) = F_X(x + a)$$

Taking derivative of both sides, $f_{(X-a)}(x) = f_X(x + a)$. Proof is completed.■

Lemma 3. Operations on discrete random variables

Let X and Y be independent discrete random variables with probability mass functions $m_X(x)$ and $m_Y(y)$. Then

1. *The absolute value of the random variable $(|X|)$ is another random variable that has*

the following `pmf`*:*

$$\begin{aligned} m_{|X|}(x) &= m_X(x) + m_X(-x) && \textit{iff} \quad x \geq 0 \\ &= 0 && \textit{iff} \quad x < 0 \end{aligned} \tag{B.3}$$

2. *The sum of the above random variables ($Z = X + Y$) has the following* `pmf`*; where $\star$ stands for convolution.*

$$m_Z(z) = m_X(z) \star m_Y(z) = \sum_{x_k} m_X(x_k) m_Y(z - x_k) \tag{B.4}$$

3. *The difference between the two random variables ($Z = X - Y$) has the following* `pmf`*:*

$$m_Z(z) = \sum_{x_k} m_X(x_k) m_Y(x_k - z) \tag{B.5}$$

4. *The absolute difference between the two random variables: $(Z = |X - Y|)$ has the following* `pmf`*:*

$$\begin{aligned} m_{|X-Y|}(z) &= \sum_{x_k} m_X(x_k) m_Y(x_k - z) + \sum_{x_k} m_X(x_k) m_Y(x_k + z) \quad \textit{iff}\ z \geq 0 \\ &= 0 \quad \textit{iff} \quad z < 0 \end{aligned} \tag{B.6}$$

5. *The ratio of two discrete random variables, $Z = \frac{X}{Y}$ has the following* `pmf`

$$m_{\left(\frac{X}{Y}\right)}(z) = \sum_{x_k} m_X(z \times x_k) m_Y(x_k) \tag{B.7}$$

Proofs

1. Using the definition of pmf and absolute value function:

$$\begin{aligned}
\forall x \geq 0 \quad m_{|X|}(x) &= Pr\left(|X| = x\right) = Pr\left((X = x) \vee (X = -x)\right) \\
&= Pr\left(X = x\right) + Pr\left(X = -x\right) = m_X(x) + m_X(-x) \\
\forall x < 0 \quad m_{|X|}(x) &= Pr\left(|X| = x\right) = 0
\end{aligned}$$

   Proof is complete.■

2. Using the formula for total probability based on conditional probabilities:

$$\begin{aligned}
m_Z(z) &= Pr\left(Z = z\right) = Pr\left(X + Y = z\right) \\
&= \sum_{x_k} Pr\left(X + Y = z \mid X = x_k\right) \times Pr\left(X = x_k\right) \\
&= \sum_{x_k} Pr\left(x_k + Y = z\right) \times Pr\left(X = x_k\right) = \sum_{x_k} m_Y(z - x_k) m_X(x_k)
\end{aligned}$$

   Proof is complete. ■

3. Similarly, for $Z = X - Y$

$$\begin{aligned}
m_Z(z) &= Pr\left(Z = z\right) = Pr\left(X - Y = z\right) \\
&= \sum_{x_k} Pr\left(X - Y = z \mid X = x_k\right) \times Pr\left(X = x_k\right) \\
&= \sum_{x_k} Pr\left(x_k - Y = z\right) \times Pr\left(X = x_k\right) = \sum_{x_k} m_Y(x_k - z) m_X(x_k)
\end{aligned}$$

   Proof is complete. ■

4. Using parts 1 to 3 (above) of this lemma,

$$\begin{aligned}\forall z \geq 0 \quad m_{|X-Y|}(z) &= m_{X-Y}(z) + m_{X-Y}(-z)\\ &= \sum_{x_k} m_Y(x_k - z) m_X(x_k) + \sum_{x_k} m_Y(z - x_k) m_X(x_k)\end{aligned}$$

$\forall z < 0 \quad m_{|X-Y|}(z) = Pr\left(|X-Y| < 0\right) = 0$ Proof is complete. ■

5. Using the formula for total probability based on conditional probabilities:

$$\begin{aligned}m_Z(z) = Pr\left(Z = z\right) = Pr\left(\frac{X}{Y} = z\right) &= \sum_{y_k} Pr\left(\frac{X}{Y} = z \mid Y = y_k\right) \times Pr\left(Y = y_k\right)\\ &= \sum_{y_k} Pr\left(\frac{X}{y_k} = z\right) \times Pr\left(Y = y_k\right) = \sum_{y_k} Pr\left(X = z \times y_k\right) \times Pr\left(Y = y_k\right)\\ &= \sum_{y_k} m_X(zy_k) \times m_Y(y_k)\end{aligned}$$

Proof is complete. ■

Appendix C

POINCaRe

POINCaRe is a computer application developed for image similarity analysis and content based image retrieval. A simplified executable version of POINCaRe can be downloaded from the Computational Intelligence Laboratory web site at University of Manitoba. POINCaRe was originally written in MATLAB but is also available as a standalone executable program. POINCaRe is named after Jules Henri Poincar (1854 - 1912), whose work on the philosophical aspects of the contrast between the mathematical and physical continua laid out the idea of tolerance space theory. POINCaRe can be also read as the initials for: Program for Object and Image Nearness Comparison and Recognition.

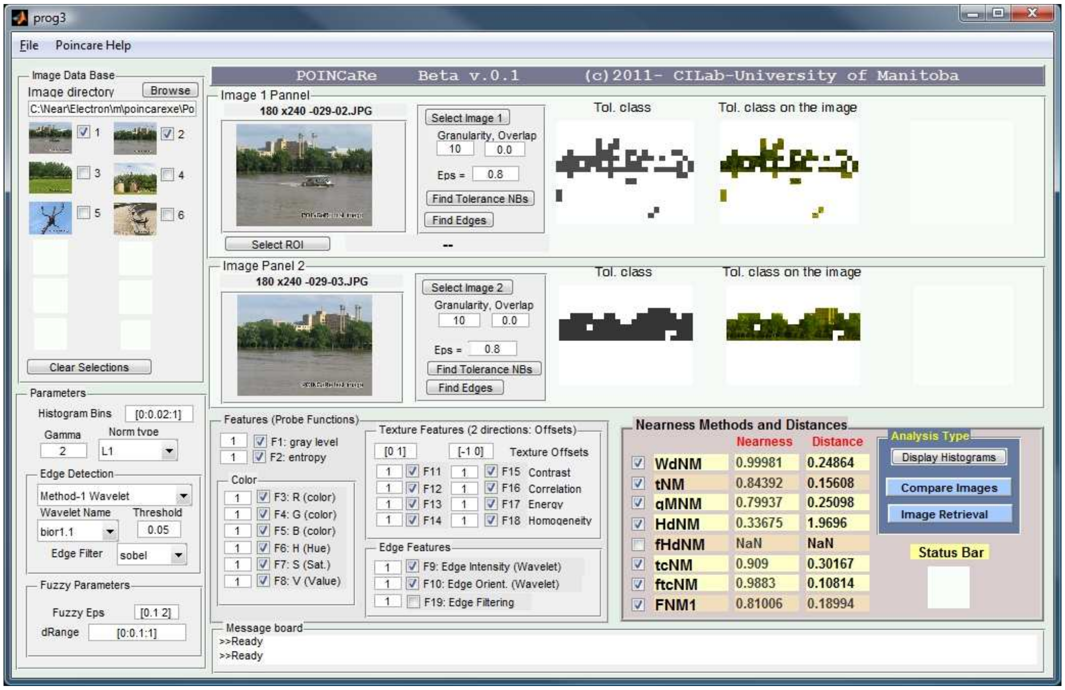
C.1 Program Features

The current released version of POINCaRe (beta 0.1) has the following capabilities:

- Calculating 8 different similarity measures between digital images based on the visual features.

- CBIR: Calculating the similarity between a given query image and a selected directory of images and sorting the images in an HTML file, based on similarity to the query image.
- Specifying a region of interest (ROI) in a query image and comparing the ROI with the test image(s).
- Individual analysis of images such as edge detection, plotting the histogram of local feature values, and finding tolerance neighborhoods in images.

Figure C.1 shows a snapshot of the graphical user interface (GUI) of the program.



POINCaRe Beta v.0.1 (c)2011- CILab-University of Manitoba

Image directory: C:\Near\Electron\m\poincarexe\Po

☑ 1 ☑ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

Image 1 Pannel: 180 x240 -029-02.JPG

Granularity, Overlap: 10, 0.0

Eps = 0.8

Image Panel 2: 180 x240 -029-03.JPG

Granularity, Overlap: 10, 0.0

Eps = 0.8

Parameters

Histogram Bins: [0:0.02:1]

Gamma: 2

Norm type: L1

Edge Detection: Method-1 Wavelet

Wavelet Name: bior1.1

Threshold: 0.05

Edge Filter: sobel

Fuzzy Parameters

Fuzzy Eps: [0.1 2]

dRange: [0:0.1:1]

Features (Probe Functions)

1 ☑ F1: gray level
1 ☑ F2: entropy

Color

1 ☑ F3: R (color)
1 ☑ F4: G (color)
1 ☑ F5: B (color)
1 ☑ F6: H (Hue)
1 ☑ F7: S (Sat.)
1 ☑ F8: V (Value)

Texture Features (2 directions: Offsets)

[0 1] [-1 0] Texture Offsets

1 ☑ F11 1 ☑ F15 Contrast
1 ☑ F12 1 ☑ F16 Correlation
1 ☑ F13 1 ☑ F17 Energy
1 ☑ F14 1 ☑ F18 Homogeneity

Edge Features

1 ☑ F9: Edge Intensity (Wavelet)
1 ☑ F10: Edge Orient. (Wavelet)
1 ☐ F19: Edge Filtering

Nearness Methods and Distances

		Nearness	Distance
☑	WdNM	0.99981	0.24864
☑	tNM	0.84392	0.15608
☑	qMNM	0.79937	0.25098
☑	HdNM	0.33675	1.9696
☐	fHdNM	NaN	NaN
☑	tcNM	0.909	0.30167
☑	ftcNM	0.9883	0.10814
☑	FNM1	0.81006	0.18994

Message board

```
>>Ready
>>Ready
```

Figure C.1: A snapshot of the GUI of POINCaRe beta version 0.1

C.2 POINCaRe Download and Install Instructions

POINCaRe is a standalone executable application that has been implemented with MATLAB but does not need MATLAB to run. However, you need to have the proper version of MATLAB Compiler Runtime (MCR) installed on your computer. You can read about MCR from Mathworks web site. Follow the following steps to install MCR and POINCaRe. If a recent version of MATLAB is installed on your computer, then you already have MCR installed. You can check the version of your MCR and go directly to step 2 below. If there is an MCR problem, you can always come back to step 1 and install a proper MCR.

Step 1: Download and Install MATLAB Compiler Runtime (MCR): Skip this step if an updated version of MATLAB or MCR has already been installed on your computer. You need MCR version 7.15 or higher to run version 0.1 (beta) of POINCaRe. If MATLAB is installed on your computer, you can type: `[major, minor] = mcrversion` at your MATLAB command prompt to see what is the version of MCR on your computer. If you don't have MATLAB or an updated version of MCR, you need to install MCR on your computer (only once). MCRInstaller.exe will install MCR on your computer. Due to licensing issues, MCRInstaller.exe file cannot be uploaded with open access. However, you can obtain this file from any licensed MATLAB distribution that comes with MATLAB compiler. In MATLAB 7.5 (R2007b) and newer, the command (`mcrinstaller`) can be used to determine where the installer is located. You can copy the file into your computer and run it.

Step 2: Download and Install POINCaRe

Currently, there is only a 64 bit version of POINCaRe available for Microsoft Windows.

Check the web site [1]or contact us for updates and new versions. Download the self extracting executable file. Copy the file into a directory in your PC. Double click on the file and the contents will be extracted into the same directory. Double click on the main file `Poincare_win64_b01.exe` to run the program.

C.3 Using the Program

Follow the following steps to use the program

C.3.1 Selecting images

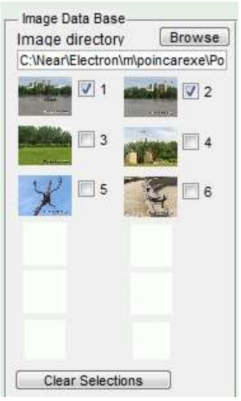
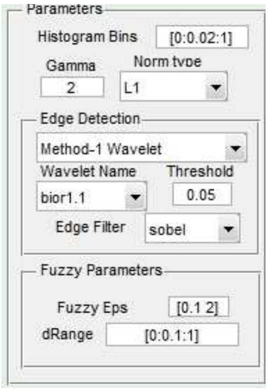


Image directory: C:\Near\Electron\m\poincarexe\Po

☑ 1 ☑ 2
□ 3 □ 4
□ 5 □ 6

Figure C.2: Image Directory



Histogram Bins: [0:0.02:1]
Gamma: 2
Norm tvpe: L1
Method-1 Wavelet
Wavelet Name: bior1.1
Threshold: 0.05
Edge Filter: sobel
Fuzzy Eps: [0.1 2]
dRange: [0:0.1:1]

Figure C.3: Parameters

You can browse and select a directory that contains your images using the controls shown in Fig. C.2. This directory should contain only images. Most of the common image file formats are supported. The images do not have to be the same size. However, since the granularity parameter (subimage size) will be the same for all images, it is strongly suggested

[1] http://wren.ece.umanitoba.ca

to avoid using images with significantly different sizes so that the ratio of the subimage size to the image itself has little variations for the sake of consistency. There is a default directory with 30 sample images located in the same path that the program is installed. These images are selected from Tasvir3x70 dataset. There is no limit in the number of images. However, a thumbnail view of the first 12 images in the directory are shown in the `Image Data Base` panel. Pairs of images can be selected from this panel to be compared.

C.3.2 Selecting Parameters

All of the methods implemented in POINCaRe are based on dividing images into subimages and calculating the local features at each subimage. **Granularity** (in pixels) is the size of square subimages and **Overlap** is a number between 0 and 1 that represents degree of overlap between subimages. Default value of overlap is zero. Figure C.4 shows where to enter the parameters. The current implemented methods require both epsilon values for image 1 and 2 to be the same.

Note that the value of granularity depends on the size of images. As a suggested compromise between accuracy and speed, it is recommended to choose the sub image size such that there are no more than approximately 500 subimages in each image. The Epsilon value (ε) for tolerance based methods can be selected as a fixed value. The method for adaptive selection of epsilon value based on the image data has not been implemented in this simplified version of POINCaRe. Other parameters can also be selected as it can be seen in Fig. C.3 as follows:

- **Histogram Bins**: This is a vector with values between 0 and 1 representing the normalized histogram bins in any method that is based on histogram calculations ($KdNM$

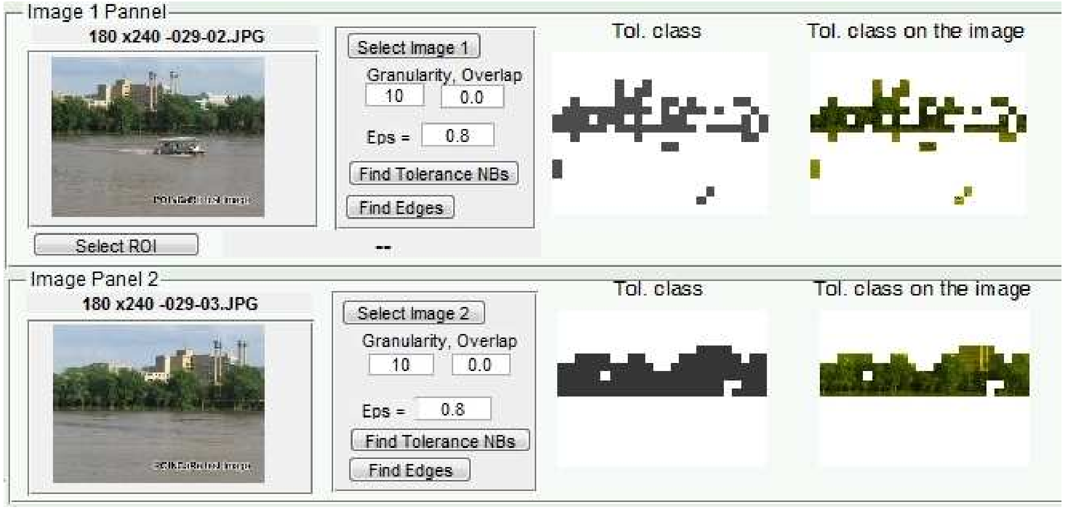


Image 1 Pannel

180 x240 -029-02.JPG

Granularity, Overlap: 10, 0.0

Eps = 0.8

Image Panel 2

180 x240 -029-03.JPG

Granularity, Overlap: 10, 0.0

Eps = 0.8

Figure C.4: POINCaRe: Image Panels

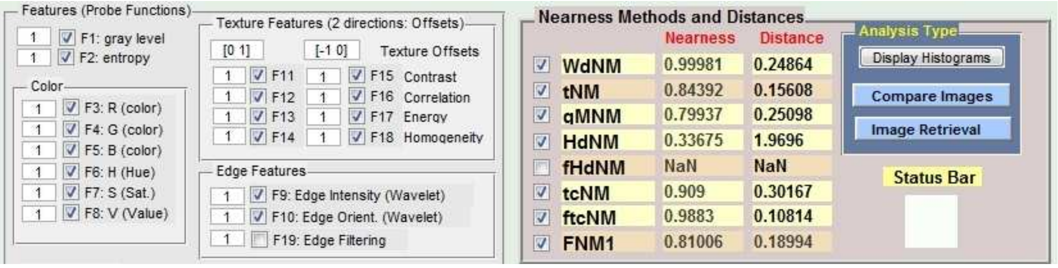
or $WdNM$)

- **Gamma (γ)**: This parameter is a scaling parameter in converting distance measure to similarity measure. if D is a distance measure, similarity or nearness measure (NM) is calculated using the mapping $NM = 1 - D^{\gamma}$ if $D \in [0, 1]$ or $NM = \frac{1}{1+D^{\gamma}}$ if $D \in [0, +\infty)$.

- **Norm Type**: The norm type is the type of the vector norm used in calculation of the distance between visual elements. Default is L_1 norm or Manhattan distance.

- **Wavelet Name** and **Threshold**: Wavelet functions of type **Wavelet Name** will be used for calculating the edge intensity and edge orientations based on the edge detection method in [37]. The threshold value **Threshold** is used to detect an edge if the edge intensity is above the threshold. The parameter **Edge Filter** is the type of filter for another method of edge detection base on the gradient of the image using different operators such as *sobel* or *prewitt* ([19]). This method will be used if the 19th feature (Edge filtering) is selected (see Fig. C.5).

- **Fuzzy Eps** and **dRange**: The parameter **Fuzzy Eps** is a vector that contains the two corner parameters of the fuzzy tolerance relation ε_1 and ε_2 as is shown in Fig. 3.4(b). **dRange** is a vector of values where the membership function of the fuzzy distance μ_{ftcFDM} is calculated.



Features (Probe Functions)

1 ☑ F1: gray level
1 ☑ F2: entropy

Color

1 ☑ F3: R (color)
1 ☑ F4: G (color)
1 ☑ F5: B (color)
1 ☑ F6: H (Hue)
1 ☑ F7: S (Sat.)
1 ☑ F8: V (Value)

Texture Features (2 directions: Offsets)

[0 1] [-1 0] Texture Offsets

1 ☑ F11 1 ☑ F15 Contrast
1 ☑ F12 1 ☑ F16 Correlation
1 ☑ F13 1 ☑ F17 Energy
1 ☑ F14 1 ☑ F18 Homogeneity

Edge Features

1 ☑ F9: Edge Intensity (Wavelet)
1 ☑ F10: Edge Orient. (Wavelet)
1 □ F19: Edge Filtering

Nearness Methods and Distances

		Nearness	Distance
☑	WdNM	0.99981	0.24864
☑	tNM	0.84392	0.15608
☑	qMNM	0.79937	0.25098
☑	HdNM	0.33675	1.9696
□	fHdNM	NaN	NaN
☑	tcNM	0.909	0.30167
☑	ftcNM	0.9883	0.10814
☑	FNM1	0.81006	0.18994

Figure C.5: POINCaRe: Features and Methods Panel

C.3.3 Choosing the features (probe functions) and methods

The user can select up to 19 different features to be used in the feature vector for each visual element (subimage). Features will be later normalized between 0 and 1. For more information on how each feature is calculated, you can refer to Chapter 4. These features describe average color, texture and edge information in each subimage. Note that POINCaRe ver 0.1 does not have the ability to automatically choose the tolerance threshold (ε). Therefore, the user is advised to choose the fixed epsilon value according to the number of selected features in each experiment. Each normalized feature has a range of variation between 0 and

1. Therefore, the range of variation of $d(x,y) = \| \vec{\phi}_{\mathcal{B}}(x) - \vec{\phi}_{\mathcal{B}}(y) \|_1$ (assuming an L_1 norm distance is used) is between 0 and M where M is the number of features.

C.4 Types of Analysis

There are 3 different types of analysis possible in POINCaRe. The first type is individual analysis on the image content of each single image. The second type is pairwise comparison of a pair of images and calculating the similarity and/or distance between images. The last type of analysis is CBIR image analysis by calculating the similarity/distance between a query image and all the images in a directory and sorting the images based on similarity. The program can perform all the above analysis types as follows:

C.4.1 Pairwise image comparison

A pair of images loaded into Image Panel 1 and Image Panel 2 can be directly compared to each other. Each image can be selected by using the `[Select Image]` button in each panel or by choosing the corresponding thumbnail image from the `[Image Data Base]` panel. Similarity between images is calculated using the given parameters in the GUI after the user clicks on `[Compare Images]` button in the `[Analysis Type]` section of the program as shown in Fig. C.5. Selected nearness and distance measure will be shown in this panel. Nearness is a normalized number between 0 and 1 and distance is a non-normalized positive real number.

C.4.2 Histogram of features

Clicking on `[Display Histograms]` will open a new window where the histograms of all the selected feature values for the subimages in each image will be displayed. The histogram bins are used as mentioned earlier. Figure C.6 shows an example of histograms generated by the program when only average R, G and B color components are selected as the features.

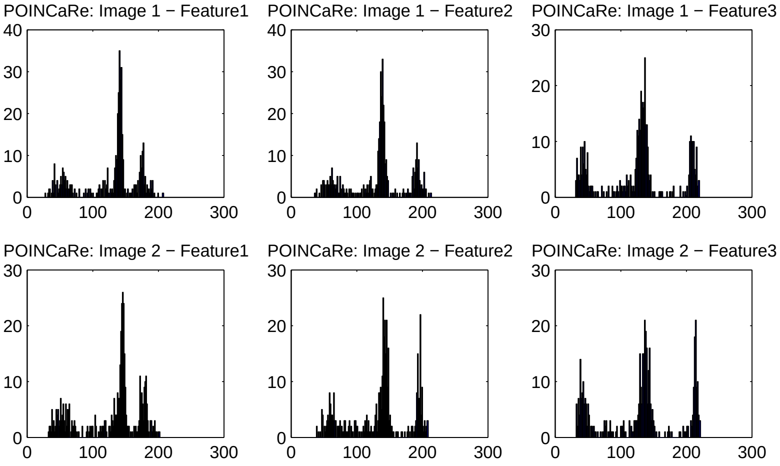


Figure C.6: POINCaRe: Histograms of the subimage average R,G and B for the pair of images shown in Fig. C.1

C.4.3 Finding tolerance neighborhoods and manual selection of a neighborhood

Clicking on `Find Tolerance NBs` for each image, calculates the tolerance neighborhoods and the number of tolerance neighborhoods and a graphical representation of the size of each neighborhood is shown in each image panel. Moreover, by clicking on any point on an image in the image panel, the corresponding tolerance neighborhood around the selected subimage will be calculated and displayed on each image panel. Figure C.4 shows sample tolerance neighborhoods selected this way after clicking on the centre of each image. Note that in this example, only R, G and B features are selected and the resulting tolerance neighborhoods represent parts of the images with almost the same color.

C.4.4 Edge detection

After clicking on `Find Edges`, edge intensity and orientation is calculated at each point and edge intensity id thresholded by the given threshold level to produce a binary image of detected edges shown in the last image window of the image panel.

C.4.5 Selecting a region of interest (ROI)

Instead of image comparison between a query image (image 1) and a test image (image 2), the program can compare part of image 1 (a region of interest as a query image) with image 2. After clicking on `[Select ROI]` in image panel 1, the user can select a region of interest by clicking on the top left and bottom right corner of a region of interest in the image and the ROI will be selected as a set of subimages. Figure C.7 shows the steps needed to select an ROI.

Upper and lower approximation of ROI using tolerance neighborhoods

After selection of an ROI, lower and upper approximations in the tolerance space will be automatically calculated and displayed on GUI. Roughly speaking, lower approximation of an ROI is defined here as the union of all the tolerance neighborhoods (/tolerance classes) which are a proper subset of the ROI. Moreover, upper approximation of an ROI is defined as the union of all the tolerance neighborhoods (/tolerance classes) which have a non-empty intersection with the ROI. Therefore, visual elements in the lower approximation, have very similar descriptions to ROI and objects that do not belong to upper approximation have very different description from ROI. The exact definitions of these approximations and some examples are as follow.

Definition C.1. *Lower Approximation $B_*(ROI)$ and Upper Approximation $B^*(ROI)$*
Let Q be the set of subimages in a query image and Y be the set of subimages in— a test image. $O = Q \cup Y$ is the set of all subimages. Let $ROI \subseteq Q$ be a region of interest in query image and $\mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}}$ be the set of all tolerance neighborhoods in the union of query and test image. Then:

$$B_*(ROI) = \bigcup \left\{ A \in \mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}} \quad \textit{such that} \quad (A \cap Q) \subseteq ROI \right\} \tag{C.1}$$

$$B^*(ROI) = \bigcup \left\{ A \in \mathsf{N}_O^{\cong_{\mathcal{B},\varepsilon}} \quad \textit{such that} \quad (A \cap Q) \cap ROI \neq \emptyset \right\} \tag{C.2}$$

Nearness between ROI and a test image

Region of interest (ROI) can be selected to show an object of interest for example in a query image where the rest of image is not important for image comparison problem. After an ROI is selected and displayed as a query itself, the user can click on [`Compare two images`] button to compare the ROI with the test image.

NOTE: Many nearness measures need a relatively large number of subimages in each image for the comparison to be meaningful. Therefore, selecting a small ROI that contains very few number of subimages, yields a nearness measure which is less meaningful because the set of subimages is not a good representative of the image anymore.

C.4.6 Content-based image retrieval

In an image retrieval experiment, a query image (or a region of interest in the query image) is compared with all the images in an image dataset. The path to the directory of images in image dataset is specified by [`Image directory`] edit box in the top left corner of the GUI. After choosing the path and selecting the query image in `image panel 1`, choosing the required parameters and probe functions, the user can start image retrieval by clicking on [`Image Retrieval`] button. All the images in image directory will be compared against the given query and the results will be stored and saved in an HTML file. Depending on the number of images and the size of subimages, image retrieval may take some time to complete. A status bar and sand-watch icon in the bottom right corner of GUI update the user about the status of the experiment and the time to completion. After all the images are compared, the program automatically opens the default Internet browser and displays the data. The program is tested with Google chrome. If you cannot see the images or if the HTML output file has not been opened for any reason, go to the program directory and open the file: (`Poincare-CBIR.html`) using a different web browser. Figure C.8 is an example of an output file generated after the given query image is compared with 210 test images in Tasvir-3x70 data set. Images are ranked based on their similarity to the query image and the values of nearness and distance are displayed for each image.

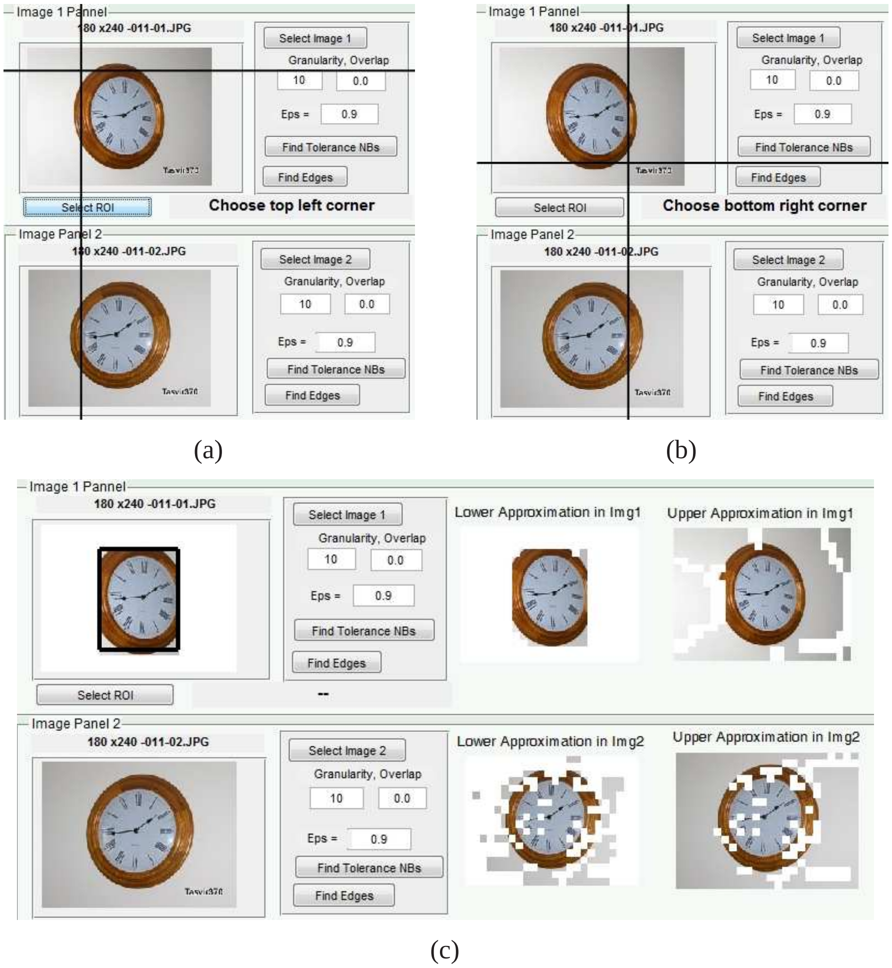
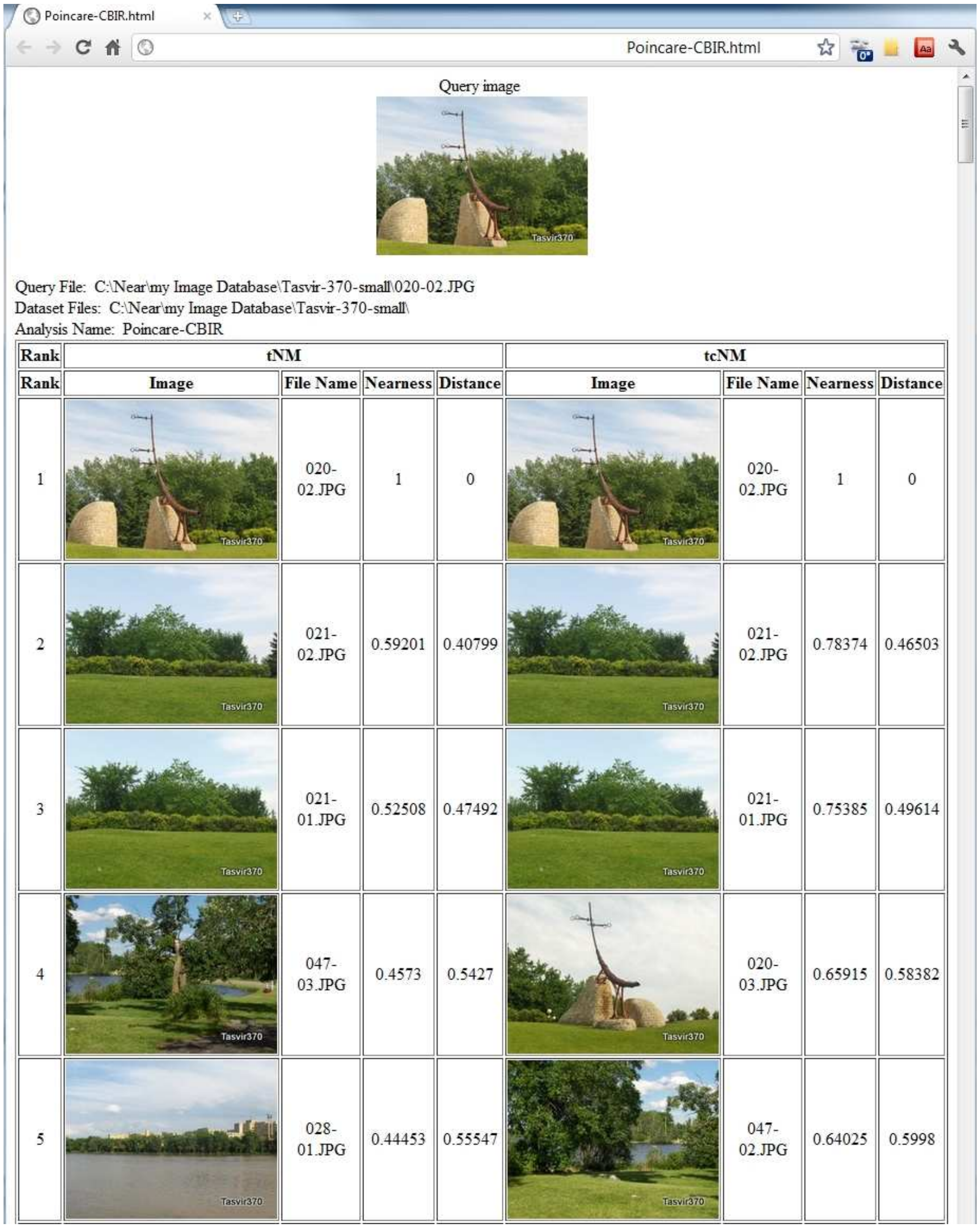


Figure C.7: Steps in selecting an ROI and rough set approximation of ROI with tolerance neighborhoods



Query image

Query File: C:\Near\my Image Database\Tasvir-370-small\020-02.JPG
Dataset Files: C:\Near\my Image Database\Tasvir-370-small\
Analysis Name: Poincare-CBIR

Rank	tNM				tcNM			
Rank	Image	File Name	Nearness	Distance	Image	File Name	Nearness	Distance
1		020-02.JPG	1	0		020-02.JPG	1	0
2		021-02.JPG	0.59201	0.40799		021-02.JPG	0.78374	0.46503
3		021-01.JPG	0.52508	0.47492		021-01.JPG	0.75385	0.49614
4		047-03.JPG	0.4573	0.5427		020-03.JPG	0.65915	0.58382
5		028-01.JPG	0.44453	0.55547		047-02.JPG	0.64025	0.5998

Figure C.8: An example of an output file generated by POINCaRe after CBIR

Bibliography

[1] "Simplicity: Content based image retrieval - image database search engine. availlable at: http://wang.ist.psu.edu/docs/related/."

[2] D. W. Aha and R. L. Bankert, "A comparative evaluation of sequential feature selection algorithms," in *Learning from Data: Artificial Intelligence and Statistics V*, ser. Lecture Notes in Statistics. Springer-Verlag, 1996, ch. 4, pp. 199–206.

[3] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern Recognition*, vol. 35, no. 4, pp. 945–965, 2002.

[4] R. Arai and S. Watanabe, "A quantitative method for comparing multi-agent-based simulations in feature space," in *Multi-Agent-Based Simulation IX: International Workshop, MAPS 2008, Estoril, Portugal, May 12-13, 2008, Revised Selected Papers*. Springer-Verlag, 2009, pp. 154–166, 1538169.

[5] F. Attneave, "Dimensions of similarity," *The American Journal of Psychology*, vol. 63, no. 4, pp. 516–556, 1950.

[6] W. Bartol, J. Mir, K. Piro, and F. Rossell, "On the coverings by tolerance classes," *Information Sciences*, vol. 166, no. 1-4, pp. 193–211, 2004.

[7] R. Beals, D. H. Krantz, and A. Tversky, "Foundation of multidimensional scaling," *Psychological Review*, vol. 75, no. 2, pp. 127–142, 1968,

[8] R. Belohlavek and T. Funiokova, "Similarity and fuzzy tolerance spaces," *J Logic Computation*, vol. 14, no. 6, pp. 827–855, 2004.

[9] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986, 11275.

[10] M. Das, M. K. Chakraborty, and T. K. Ghoshal, “Fuzzy tolerance relation, fuzzy tolerance space and basis,” *Fuzzy Sets and Systems*, vol. 97, no. 3, pp. 361–369, 1998.

[11] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Comput. Surv.*, vol. 40, no. 2, pp. 1–60, 2008, 1348248.

[12] Y. Deng and W. Du, “The kantorovich metric in computer science: A brief survey,” *Electronic Notes in Theoretical Computer Science*, vol. 253, no. 3, pp. 73–82, 2009.

[13] D. DuBois and H. M. Prade, *Fuzzy sets and systems: theory and applications*, ser. Mathematics in science and engineering. Academic Press, 1980.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley New York, 2001.

[15] M. Y. Duker and J. S., *Ophthalmology*, Mosby, St. Louis, 2009.

[16] R. Engelking, *General topology*, ser. Sigma series in pure mathematics. Berlin: Heldermann Verlag, 1989.

[17] G. T. Fechner, *Elemente der psychophysik*. Breitkopf und Hrtel, 1860.

[18] A. L. Gibbs and F. E. Su, “On choosing and bounding probability metrics,” *International Statistical Review / Revue Internationale de Statistique*, vol. 70, no. 3, pp. 419–435, 2002.

[19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2007.

[20] C. Gope and N. Kehtarnavaz, “Affine invariant comparison of point-sets using convex hulls and hausdorff distances,” *Pattern Recognition*, vol. 40, no. 1, pp. 309–320, 2007.

[21] U. Hahn, N. Chater, and L. B. Richardson, “Similarity as transformation,” *Cognition*, vol. 87, no. 1, pp. 1–32, 2003.

[22] R. M. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[23] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 6, pp. 610–621, 1973.

[24] A. E. Hassanien, A. Abraham, J. F. Peters, G. Schaefer, and C. Henry, “Rough sets and near sets in medical imaging: A review,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 13, no. 6, pp. 955–968, 2009.

[25] C. Henry, "Near stes, theory and applications," Ph.D. dissertation, University of Manitoba, 2010.

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

[27] P. W. Huang and S. K. Dai, "Image retrieval by texture similarity," *Pattern Recognition*, vol. 36, no. 3, pp. 665–679, 2003.

[28] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 9, pp. 850–863, 1993.

[29] N. Jhanwar, S. Chaudhuri, G. Seetharaman, and B. Zavidovique, "Content based image retrieval using motif cooccurrence matrix," *Image and Vision Computing*, vol. 22, no. 14, pp. 1211–1220, 2004.

[30] K. Koffka, *Principles of Gestalt psychology*. Routledge, 1999.

[31] G. K. Lang and O. Gareis, *Ophthalmology: A pocket textbook atlas*. Thieme Medical Publishers, 2006.

[32] C. P. Lee, "Robust image segmentation using active contours: Level set approaches," Ph.D. dissertation, North Carolina State University, 2005.

[33] C.-H. Lin, R.-T. Chen, and Y.-K. Chan, "A smart content-based image retrieval system based on color and texture feature," *Image and Vision Computing*, vol. 27, no. 6, pp. 658–665, 2009.

[34] G.-H. Liu, L. Zhang, Y.-K. Hou, Z.-Y. Li, and J.-Y. Yang, "Image retrieval based on multi-texton histogram," *Pattern Recognition*, vol. 43, no. 7, pp. 2380–2389, 2010.

[35] P. C. Mahalanobis, "On the generalized distance in statistics," vol. 2, 1936, pp. 49–55, proc. Nat. Inst. Sci. India 1.

[36] I. M. Makarov, V. Z. Rakhmankulov, A. A. Akhrem, V. I. Sevryuk, and A. V. Sokletkin, "A computer system for integrated analysis and simulation of innovative biosensors," *Computational Mathematics and Modeling*, vol. 13, no. 4, pp. 427–442, 2002.

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

[38] B. S. Manjunath and W. Y. Ma, “Texture features for browsing and retrieval of image data,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, 1996.

[39] D. J. Marceau, P. J. Howarth, J. M. Dubois, and D. J. Gratton, “Evaluation of the grey-level co-occurrence matrix method for land-cover classification using spot imagery,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 4, pp. 513–519, 1990.

[40] 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 *8th Int’l Conf. Computer Vision*, vol. 2, 2001, pp. 416–423.

[41] A. H. Meghdadi and J. F. Peters, “Perceptual tolerance neighborhood-based similarity in content-based image retrieval and classification,” *International Journal of Intelligent Computing and Cybernetics*, vol. 5, no. 3, p. to appear, 2012.

[42] R. D. Melara, “The concept of perceptual similarity: From psychophysics to cognitive psychology,” in *Psychophysical approaches to cognition*, D. Algom, Ed. Amsterdam: North-Holland, 1992, p. 303388.

[43] H. Müller, W. Mller, D. M. Squire, S. Marchand-Maillet, and T. Pun, “Performance evaluation in content-based image retrieval: overview and proposals,” *Pattern Recognition Letters*, vol. 22, no. 5, pp. 593–601, 2001.

[44] P. P. Ohanian and R. C. Dubes, “Performance evaluation for four classes of textural features,” *Pattern Recognition*, vol. 25, no. 8, pp. 819–833, 1992.

[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. Plenum Pres, 1985, pp. 465–482.

[47] B. G. Park, K. M. Lee, and S. U. Lee, “Color-based image retrieval using perceptually modified hausdorff distance,” *J. Image Video Process.*, vol. 2008, no. 2, pp. 1–10, 2008, 1384988.

[48] J. F. Peters, “Near sets. general theory about nearness of objects,” *Applied Mathematical Sciences*, vol. 1, no. 53, pp. 2609–2629, 2007.

[49] ——, “Near sets. special theory about nearness of objects,” *Fundamenta Informaticae*, vol. 75, pp. 407–433, 2007.

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

[51] J. F. Peters and P. Wasilewski, “Foundations of near sets,” *Information Sciences*, vol. 179, no. 18, pp. 3091–3109, 2009.

[52] H. Poincaré, *La Science et l’Hypothèse*. Paris: Ernerst Flammarion, 1902, later ed,; Champs sciences, Flammarion, 1968 & Science and Hypothesis, trans. by J. Larmor, Walter Scott Publishing, London, 1905; cf. Mead Project at Brock University.

[53] P. Pudil, J. Novoviov, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[54] J. Puzicha, “Distribution-based image similarity,” in *State-of-the-Art in Content-Based Image and Video Retrieval [Dagstuhl Seminar, 5-10 December 1999]*. Kluwer, B.V., 2001, pp. 143–164, 725217 143-164.

[55] S. Ramanna, A. H. Meghdadi, and J. F. Peters, “Nature-inspired framework for measuring visual image resemblance: A near rough set approach,” *Theoretical Computer Science*, vol. 412, no. 42, pp. 5926–5938, 2011.

[56] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons Inc, 2004.

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

[58] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, “Relevance feedback: a power tool for interactive content-based image retrieval,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 5, pp. 644–655, 1998.

[59] M. A. Ruzon and C. Tomasi, “Color edge detection with the compass operator,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, p. 166 Vol. 2.

[60] ——, “Edge, junction, and corner detection using color distributions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1281–1295, 2001.

[61] S. Santini and R. Jain, “Similarity measures,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 9, pp. 871–883, 1999.

[62] R. Shepard, “The analysis of proximities: Multidimensional scaling with an unknown distance function. ii,” *Psychometrika*, vol. 27, no. 3, pp. 219–246, 1962.

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

[64] L. K. Soh and C. Tsatsoulis, “Texture analysis of sar sea ice imagery using gray level co-occurrence matrices,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 37, no. 2, pp. 780–795, 1999.

[65] 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.

[66] M. A. Stricker and M. Orengo, “Similarity of color images,” ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, W. Niblack and R. C. Jain, Eds., vol. 2420, 1995, pp. 381–392.

[67] P. Synak and D. Slezak, “Tolerance based templates for information systems: Foundations and perspectives,” in *Advances in Hybrid Information Technology*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4413, pp. 11–19.

[68] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. Elsevier, 2006.

[69] L. L. Thurstone, “A law of comparative judgement,” *Psychological Review*, no. 34, pp. 273–286, 1927.

[70] A. Tversky and I. Gati, “Studies of similarity,” *Cognition and categorization*, pp. 79–98, 1978.

[71] A. Tversky, “Features of similarity,” *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

[72] R. Verde and A. Irpino, “Comparing histogram data using a mahalanobiswasserstein distance,” in *COMPSTAT 2008*, 2008, pp. 77–89.

[73] A. Vershik, “Kantorovich metric: Initial history and little-known applications,” *Journal of Mathematical Sciences*, vol. 133, no. 4, pp. 1410–1417, 2006.

[74] J. Z. Wang, L. Jia, and G. Wiederhold, “Simplicity: semantics-sensitive integrated matching for picture libraries,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 9, pp. 947–963, 2001.

[75] B. L. Welch, “The generalization of ‘student’s’ problem when several different population variances are involved,” *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.

[76] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965, doi: DOI: 10.1016/S0019-9958(65)90241-X.

[77] ——, “Similarity relations and fuzzy orderings,” *Information Sciences*, vol. 3, no. 2, pp. 177–200, 1971.

[78] ——, “The concept of a linguistic variable and its application to approximate reasoning–i,” *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[79] ——, “Outline of a new approach to the analysis of complex systems and decision processes,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[80] E. C. Zeeman, “The topology of the brain and visual perception,” *Topology of 3-manifolds*, vol. 3, pp. 240–248, 1962.

[81] C. Zhao, W. Shi, and Y. Deng, “A new hausdorff distance for image matching,” *Pattern Recognition Letters*, vol. 26, no. 5, pp. 581–586, 2005.

[82] X. S. Zhou and T. S. Huang, “Relevance feedback in image retrieval: A comprehensive review,” *Multimedia Systems*, vol. 8, no. 6, pp. 536–544, 2003.

Author Index

Subject Index