Near set Evaluation And Recognition (NEAR) System V3.0

Christopher Henry
James F. Peters
ch.henry@uwinnipeg.ca
jfpeters@ee.umanitoba.ca
University of Manitoba
Computational Intelligence Laboratory
ENGR E1-526 Engineering & Information Technology Complex
75A Chancellor's Circle
Winnipeg, Manitoba R3T 5V6
Tel.: 204.474.9603
Fax.: 204.261.4639

UM CI Laboratory Technical Report
Number TR-2012-019
January 9, 2012

University of Manitoba

Computational Intelligence Laboratory
URL: http://wren.ee.umanitoba.ca

Project no. 2011-019

Near set Evaluation And Recognition (NEAR) System V3.0

Christopher Henry
James F. Peters
ch.henry@uwinnipeg.ca
jfpeters@ee.umanitoba.ca
University of Manitoba
Computational Intelligence Laboratory
ENGR E1-526 Engineering & Information Technology Complex
75A Chancellor's Circle
Winnipeg, Manitoba R3T 5V6
Tel.: 204.474.9603
Fax.: 204.261.4639

CI Laboratory TR-2012-019

January 9, 2012

This research work has been funded by Manitoba Hydro grants T137, T247, T260, T270, T277, and by the Natural Sciences & Engineering Research Council of Canada (NSERC) grant 185986, NSERC Postgraduate Doctoral Fellowship PGS-D3, University of Manitoba Faculty of Engineering grant, and Canadian Arthritis Network grant SRI-BIO-05.

Contents

Abstract

This report introduces version V3.0 of the Near set Evaluation And Recognition (NEAR) system. The goal of the NEAR system is to extract perceptual information from images using near set theory, which provides a framework for measuring the perceptual nearness of objects. The contributions of this report are an introduction to the NEAR system as an application of near set theory to image processing, a feature-based approach to solving the image correspondence problem, and a first step toward automating the extraction of perceptual information from images where there is interest in measuring the degree of resemblance between images. This new version of the NEAR system includes region-of-interest and image neighbourhood analysis. For the first time, topological structures in digital images are considered. Various distance functions are implemented to quantify the degree nearness or apartness of pairs of digital images.
Keywords: Digital images, distance functions, features, image analysis, metrics, near sets, nearness, neighbourhoods, pattern recognition, perception, topological structures.

1 Introduction

The goal of the NEAR system is to demonstrate applications of the near set theory presented in [1–15]. The system implements a Multiple Document Interface (MDI) (see, *e*.g., Fig. 1), where each separate processing task is performed in its own child frame. The objects (in the near set sense) in this system are subimages of the images being processed and the probe functions (representing features) are image processing functions defined on the subimages. The system was written in C++ and was designed to facilitate the addition of new processing tasks and probe functions[1]. Currently, the system performs a number major tasks, namely, displaying equivalence and tolerance classes for an image, performing segmentation evaluation, measuring the nearness of two images, determining the nearness of neighbourhoods of points in pairs of regions of interest (ROIs), and displaying the locations of equivalence class subimages for a selected feature. This report is organized as follows: Section 2 gives some background on near set theory, and Section 3 demonstrates the application of near set theory to images. Finally, Sections 4-7 describe the operation of the GUI.

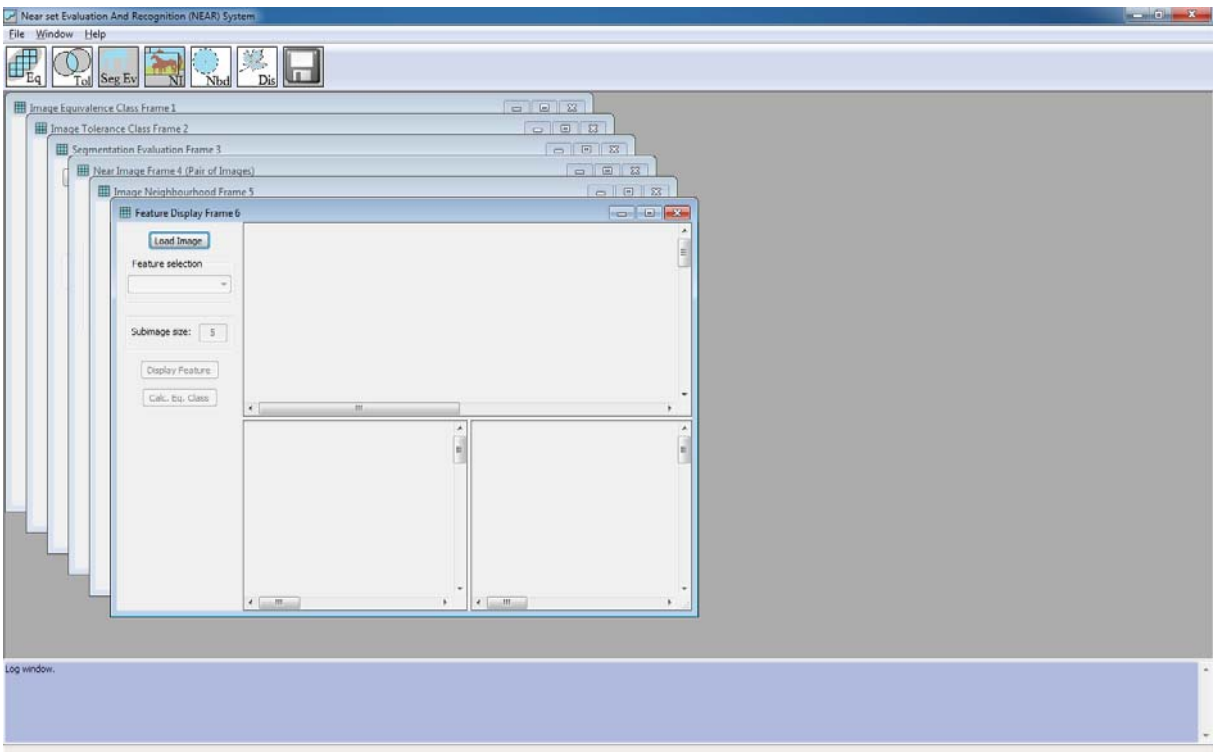


Figure 1: NEAR system GUI.

2 Near sets

Near set theory focuses on nonempty sets of perceptual objects with matching descriptions. Specifically, let O represent a nonempty set of available objects. The description of an object $x \in O$ is given by

$$\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_l(x)),$$

where l is the length of the description and each $\phi_i(x)$ is a probe function for a particular feature in the description of the object x. The notion of a probe function in near sets is inspired by Monique Pavel [18],

[1]Parts of the Graphical User Interface (GUI) were inspired by the GUI reported in [16] and the wxWidgets example in [17].

where a probe function that is invariant relative to a transformation of the images results in matching function (feature) values. In the near set approach, a real-valued funcion $\phi : O \longrightarrow \Re$, O a set of images, is a *probe function* if, and only if, ϕ represents an image feature with values that are in the description of a perceptual object, in particular, in the description of an image [19–21]. Furthermore, a set $\mathbb{F}$ can be defined that represents all the probe functions used to describe an object x. Next, a perceptual system S can be defined as $S = \langle O, \mathbb{F}, \{Val_{\phi_i}\}_{\phi_i \in \mathbb{F}}\rangle$, where $\mathbb{F}$ is the set of all possible probe functions that take as the domain objects in O, and $\{Val_{\phi_i}\}_{\phi_i \in \mathbb{F}}$ is the value range of a function $\phi_i \in \mathbb{F}$. For simplicity, a perceptual system is abbreviated as $\langle O, \mathbb{F}\rangle$ when the range of the probe functions is understood. It is the notion of a perceptual system that is at the heart of the following definitions.

Definition 1 Normative Indiscernibility Relation *[19] Let $\langle O, \mathbb{F}\rangle$ be a perceptual system. For every $\mathcal{B} \subseteq \mathbb{F}$, the normative indiscernibility relation $\sim_{\mathcal{B}}$ is defined as follows:*

$$\sim_{\mathcal{B}} = \{(x, y) \in O \times O : \parallel \phi(x) - \phi(y) \parallel = 0\},$$

where $\|\cdot\|$ represents the l^2 norm. If $\mathcal{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\sim_{\{\phi\}}$ we write $\sim_\phi$.

Defn. 1 is a refinement of the original indiscernibility relation given by Pawlak in 1981 [22]. Using the indiscernibility relation, objects with matching descriptions can be grouped together forming granules of highest object resolution determined by the probe functions in $\mathcal{B}$. This gives rise to an elementary set (also called an equivalence class)

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

defined as a set where all objects have the same description. Similarly, a quotient set is the set of all elementary sets defined as

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

Defn. 1 provides the framework for comparisons of sets of objects by introducing a concept of nearness within a perceptual system. Sets can be considered near each other when they have "things" in common. In the context of near sets, the "things" can be quantified by granules of a perceptual system, *i.e.*, the elementary sets. The simplest example of nearness between sets sharing "things" in common is the case when two sets have indiscernible elements. This idea leads to the definition of a weak nearness relation.

Definition 2 Weak Nearness Relation [2]
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and nonempty subsets $X, Y \subseteq O$. A set X is weakly near to a set Y within the perceptual system $\langle O, \mathbb{F}\rangle$ ($X \underline{\bowtie}_{\mathbb{F}} Y$) iff there are $x \in X$ and $y \in Y$ and there is $\mathcal{B} \subseteq \mathbb{F}$ such that $x \sim_{\mathcal{B}} y$. In the case where sets X, Y are defined within the context of a perceptual system as in Defn2, then X, Y are weakly near each other.

An examples Defn. 2 is given in Fig. 2 where the grey lines represent equivalence classes. The sets X and Y are weakly near each other in Fig. 2 because they both share objects belonging to the same equivalence class.

Defn. 2 can be used to define a Nearness Measure (NM) between two nonempty sets X and Y [5, 9]. Let $Z = X \cup Y$ and let the notation

$$[z_{/\sim_{\mathcal{B}}}]_X = \{z \in z_{/\sim_{\mathcal{B}}} \mid z \in X\},$$

denote the portion of the elementary set $z_{/\sim_{\mathcal{B}}}$ that belongs to X, and similarly, use the notation

$$[z_{/\sim_{\mathcal{B}}}]_Y = \{z \in z_{/\sim_{\mathcal{B}}} \mid z \in Y\},$$

to denote the portion that belongs to Y. Further, let the nonempty sets X and Y be weakly near each other using Defn. 2. Then, a NM between X and Y is given by

$$NM_{\sim_\mathcal{B}} = \left(\sum_{z_{/\sim_\mathcal{B}} \in Z_{/\sim_\mathcal{B}}} |z_{/\sim_\mathcal{B}}| \right)^{-1} \sum_{z_{/\sim_\mathcal{B}} \in Z_{/\sim_\mathcal{B}}} |z_{/\sim_\mathcal{B}}| \frac{\min(|[z_{/\sim_\mathcal{B}}]_X|, |[z_{/\sim_\mathcal{B}}]_Y|)}{\max(|[z_{/\sim_\mathcal{B}}]_X|, |[z_{/\sim_\mathcal{B}}]_Y|)} \tag{1}$$

The idea behind Eq. 1 is that sets that are similar should have similar number of objects in each equivalence class. Thus, for each equivalence class obtained from $Z = X \cup Y$, Eq. 1 counts the number of objects that belong to X and Y and takes the ratio (as a proper fraction) of their cardinalities. Furthermore, each ratio is weighted by the total size of the equivalence class (thus giving importance to the larger classes) and the final result is normalized by dividing by the sum of all the cardinalities. The range of Eq.1 is in the interval [0,1], where a value of 1 is obtained if the sets are equivalent and a value of 0 is obtained if they have no elements in common.

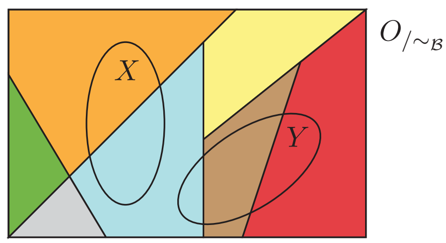
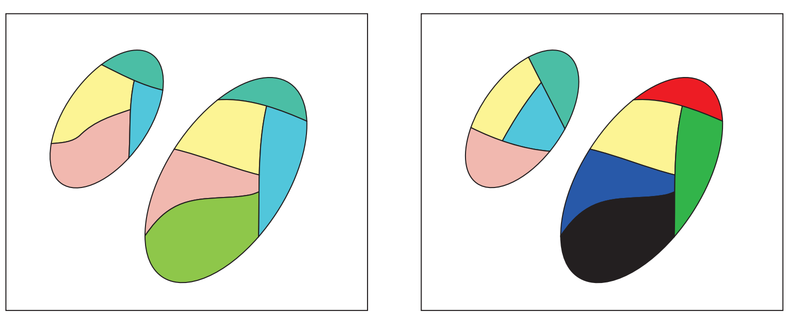


Figure 2: Example of Defn. 2.

As an example of the degree of nearness between two sets, consider Fig.3 in which each image consists of two sets of objects, X and Y. Each colour in the figures corresponds to an elementary set where all the objects in the class share the same description. The idea behind Eq. 1 is that the nearness of sets in a perceptual system is based on the cardinality of equivalence classes that they share. Thus, the sets in Fig. 3(a) are closer (more near) to each other in terms of their descriptions than the sets in Fig.3(b).



(a) (b)

Figure 3: Example of degree of nearness between two sets: (a) High degree of nearness, and (b) low degree of nearness.

2.1 Tolerance relation

A perception-based approach to discovering resemblances between images leads to a tolerance class form of near sets that models human perception in a physical continuum viewed in the context of image tolerance spaces. A tolerance space-based approach to perceiving image resemblances hearkens back to the observation about perception made by Ewa Orłowska in 1982 [23] (see, also, [24]), *i.e.*, classes defined in an approximation space serve as a formal counterpart of perception.

The term *tolerance space* was coined by E.C. Zeeman in 1961 in modeling visual perception with tolerances [25]. A tolerance space is a set X supplied with a binary relation $\simeq$ (*i.e.*, a subset $\simeq \subset X \times X$) that is reflexive (for all $x \in X$, $x \simeq\ x$) and symmetric (*i.e.*, for all $x, y \in X$, $x \simeq\ y$ implies $y \simeq x$) but transitivity of $\simeq$ is not required. For example, it is possible to define a tolerance space relative to subimages of an image. This is made possible by assuming that each image is a set of fixed points. Let O denote a set of perceptual objects (*e.g.*, gray level subimages) and let $gr(x)$ = average gray level of subimage x. Then define the tolerance relation

$$\simeq_{gr} = \{(x, y) \in O \times O \mid\ |gr(x) - gr(y)| \leq \varepsilon\},$$

for some tolerance $\varepsilon \in \Re$ (reals). Then $(O, \simeq_{gr})$ is a sample tolerance space. The tolerance ε is directly related to the exact idea of closeness or resemblance (*i.e.*, being within some tolerance) in comparing objects. The basic idea is to find objects such as images that resemble each other with a tolerable level of error. Sossinsky [26] observes that main idea underlying tolerance theory comes from Henri Poincaré [27]. Physical continua (*e.g.*, measurable magnitudes in the physical world of medical imaging [9]) are contrasted with the mathematical continua (real numbers) where almost solutions are common and a given equation have no exact solutions. An *almost solution* of an equation (or a system of equations) is an object which, when substituted into the equation, transforms it into a numerical 'almost identity', i.e., a relation between numbers which is true only approximately (within a prescribed tolerance) [26]. Equality in the physical world is meaningless, since it can never be verified either in practice or in theory. Hence, the basic idea in a tolerance space view of images, for example, is to replace the indiscerniblity relation in rough sets [28] with a tolerance relation in partitioning images into homologous regions where there is a high likelihood of overlaps, *i.e.*, non-empty intersections between image tolerance classes. The use of image tolerance spaces in this work is directly related to recent work on tolerance spaces (see, *e.g.*, [3, 4, 7–9, 29–34]).

When dealing with perceptual objects (especially, components in images), it is sometimes necessary to relax the equivalence condition of Defn. 1 to facilitate observation of associations in a perceptual system. This variation is called a tolerance relation that defines yet another form of near sets [3, 4, 7] and is given in Defn. 3.

Definition 3 Tolerance Nearness Relation *[3]*
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $\epsilon \in \mathbb{R}$. For every $\mathcal{B} \subseteq \mathbb{F}$ the tolerance relation $\cong_{\mathcal{B}}$ is defined as follows:

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

If $\mathcal{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\cong_{\{\phi\}}$ we write $\cong_{\phi}$. Further, for notational convince, we will write $\cong_{\mathcal{B}}$ instead of $\cong_{\mathcal{B},\epsilon}$ with the understanding that ϵ is inherent to the definition of the tolerance relation.

As in the case with the indiscernibility relation, a tolerance class can be defined as

$$x_{/\cong_{\mathcal{B}}} = \{y \in O \mid y \cong_{\mathcal{B}} x'\ \forall\, x' \in x_{/\cong_{\mathcal{B}}}\}. \tag{2}$$

Note, Defn. 3 covers O instead of partitioning it because an object can belong to more than one class. As a result, Eq. 2 is called a tolerance class instead of an elementary set. In addition, each pair of objects x, y in a tolerance class $x_{/\cong_{\mathcal{B}}}$ must satisfy the condition $\parallel \phi(x) - \phi(y) \parallel \leq \epsilon$. Next, a quotient set for a given a tolerance relation is the set of all tolerance classes and is defined as

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

Table 1: Tolerance Class Example

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

Notice that the tolerance relation is a generalization of the indiscernibility relation given in Defn.1 (obtained by setting $\epsilon = 0$). As a result, Defn. 2 and Eq. 1 can be redefined with respect to the tolerance relation[2].

The following simple example highlights the need for a tolerance relation as well as demonstrates the construction of tolerance classes from real data. Consider the 20 objects in Table1 that where $|\phi(x_i)| = 1$. Letting $\epsilon = 0.1$ gives the following tolerance classes:

$$\begin{aligned} X_{/\cong_{\mathcal{B}}} \quad = \quad & \{\{x_1, x_8, x_{10}, x_{11}\}, \{x_1, x_9, x_{10}, x_{11}, x_{14}\}, \\ & \{x_2, x_7, x_{18}, x_{19}\}, \\ & \{x_3, x_{12}, x_{17}\}, \\ & \{x_4, x_{13}, x_{20}\}, \{x_4, x_{18}\}, \\ & \{x_5, x_6, x_{15}, x_{16}\}, \{x_5, x_6, x_{15}, x_{20}\}, \\ & \{x_6, x_{13}, x_{20}\}\} \end{aligned}$$

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

3 Perceptual image processing

Near set theory can be easily applied to images. For example, define a RGB image as $f = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$, where $\mathbf{p}_i = (c, r, R, G, B)^{\mathrm{T}}$, $c \in [1, M]$, $r \in [1, N]$, $R, G, B \in [0, 255]$, and M, N respectively denote the width and height of the image and $M \times N = T$. Further, define a square subimage as $f_i \subset f$ with the following conditions:

$$\begin{aligned} & f_1 \cap f_2 \ldots \cap f_s = \emptyset, \\ & f_1 \cup f_2 \ldots \cup f_s = f, \end{aligned} \tag{3}$$

where s is the number of subimages in f. The approach taken in the NEAR system is to restrict all subimages to be square except when doing so violates Eq. 3. For example, the images in the Berkeley Segmentation Dataset [35] often have the dimension 321×481. Consequently, a square subimage size of 25 will produce 6240 square subimages, 96 subimages of size 1×5, 64 subimages of size 5×1 and 1 subimage consisting of a single pixel. Next, O can be defined as the set of all subimages, *i.e.*, $O = \{f_1, \ldots, f_s\}$, and $\mathbb{F}$ is a set of functions that operate on images (see, *e.g.* Table 2 for examples of probe functions used in the NEAR system or [36] for other examples). Once the set $\mathcal{B}$ has been selected, the elementary sets are simply created by grouping all objects with the same description and the quotient set is made up of all the elementary sets.

[2] The two relations were treated separately in the interest of clarity.

Finally, a simple example of these concepts is given in Fig. 4 where the left image contains an octagon with a radius of 100 pixels located at the centre of the 400×400 image, and the right image contains the elementary sets obtained using $\mathcal{B} = \{\phi_{\text{avg}}(f_s)\}$ and a subimage size of 10×10.

Table 2: Example probe functions

ϕ_i	Description
$\phi_{\text{avg}}(f_s)$	Average greyscale value of subimage
$\phi_{\text{IC}}(f_s)$	Information content of subimage [2]
$\phi_{\text{NormR}}(f_s)$	Average normalized R value of subimage [36]


(a) (b)

Figure 4: Example of near set theory in the context of image processing: (a) Original image, and (b) elementary sets obtained from (a) using $\phi_{\text{avg}}(f_s)$.

Observe that three elementary sets are obtained in Fig. 4(b), namely, the light grey background, the dark grey octagon interior, and the black squares along the diagonals. The black squares are created by subimages that contain both black and white pixels (in the original image) and are located only on the diagonals due to the subimage size and shape, and the position and radius of the hexagon. All other subimages are uniformly white or black. Thus, we are presented with perceptual information in the form of three equivalence classes when restricted to only being able to describe the original image with the probe function $\mathcal{B} = \{\phi_{\text{avg}}(f_s)\}$ and a subimage size of 10×10. This example clearly demonstrates that perceptual information obtained from the application of near set theory is represented by the elementary sets (formed by the grouping of objects with similar descriptions), and the information gained is always presented with respect to the probe functions contained in $\mathcal{B}$.

4 Equivalence class frame

This frame calculates equivalence classes using the Perceptual Indiscernibility Relation Perceptual Indiscernibility relation in Definition 1, *i*.e., given an image X, it will calculate $X_{/\sim_{\mathcal{B}}}$, where the objects are Subimage subimages of X. An example using this frame is given in Fig. 5 and is obtained by the following steps:

1. Click *Load Image* button and select an image.
2. Click the *Set Parameters* button.

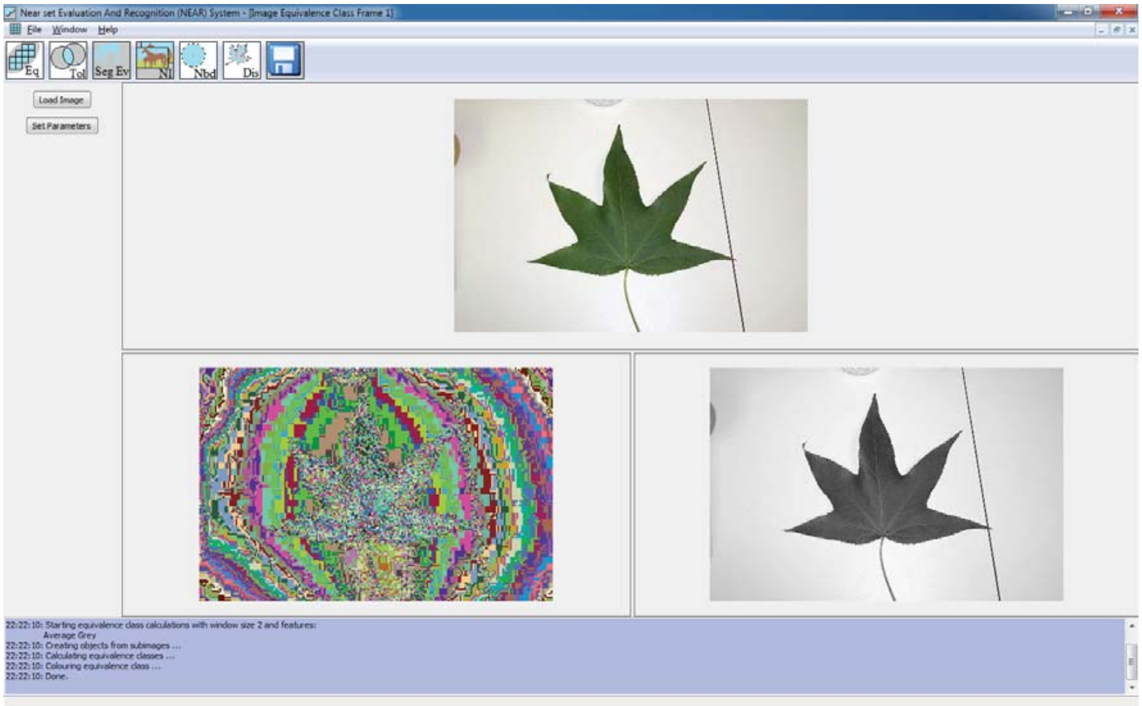


Figure 5: Sample run of the equivalence class frame using a window size of 2×2 and $\mathcal{B} = \{\phi_{\text{Average Grey}}$ (Image shown in NEAR system used with permission [37]).

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

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

5. Select features.

6. Click *Run*.

The result is given in Fig. 5, where the bottom left window contains an image of the equivalence classes, and each colour represents a single class. The bottom right window is used to display equivalence classes by clicking in any of the three images. The coordinates of the mouse click determine the equivalence class that is displayed. The results may be saved by clicking on the save button.

5 Tolerance class frame

This frame finds tolerance classes using the Perceptual Tolerance Relation perceptual tolerance relation in **Definition**, *i*.e., given an image X, this frame finds $H_{\cong_{\mathcal{B},\varepsilon}}(O)$, where the objects are subimages of X Subimage (see **Section** for further explanation). An example using this frame is given in Fig. 6 and is obtained by the following steps:

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

2. Click the *Set Parameters* button.

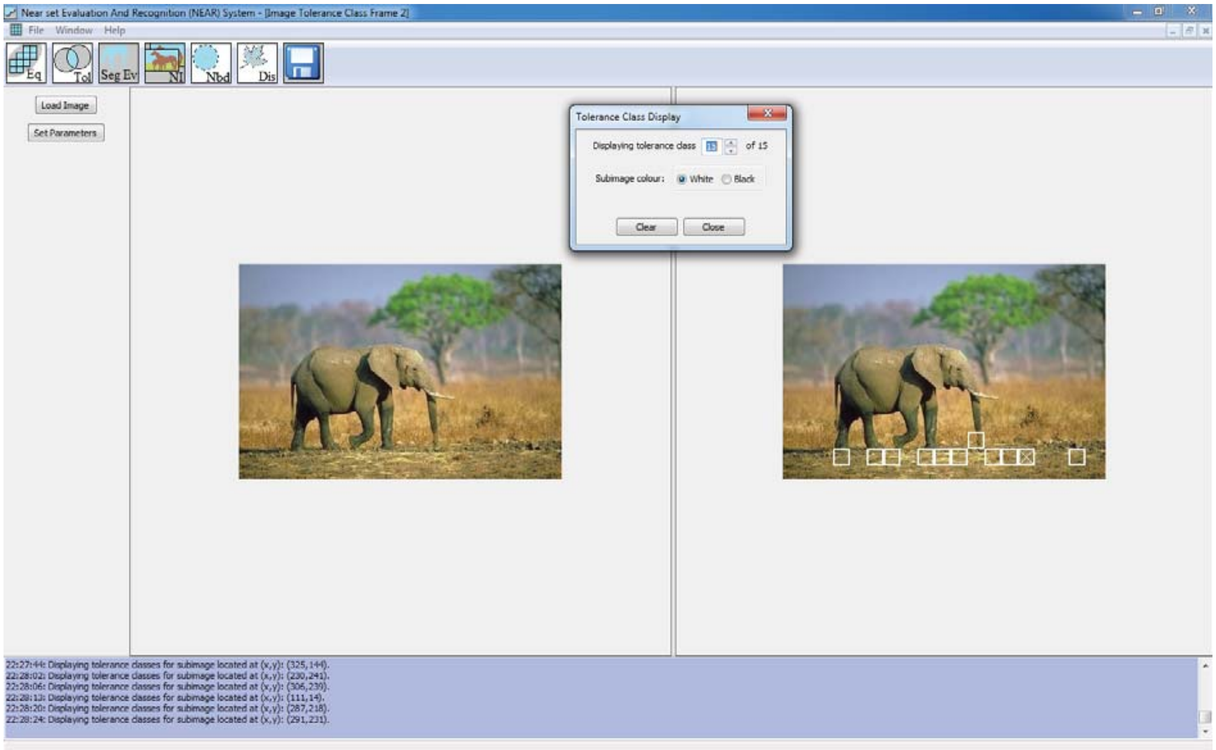


Figure 6: Sample run of the tolerance class frame using a window size of 20×20, 18 features used to generate the results in this thesis, and $\varepsilon = 0.7$ (Image shown in NEAR system used with permission [38, 39]).

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

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

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

6. Select features.

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

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

9. Click *Run*.

The result is given in Fig. 6 where the left side is the original image, and the right side is used to display the tolerance classes. Since the tolerance relation covers an image instead of partitioning the image, the tolerance classes are displayed upon request. For instance, by clicking on either of the two images, a window appears letting the user display each tolerance class containing the subimage selected by the mouse. Further, the subimage containing the mouse click contains an ‘X’, and the subimages can be coloured white or black.

6 Segmentation evaluation frame

This frame performs segmentation evaluation using Perceptual Morphology perceptual morphology as described in **Section**, where the evaluation is labelled the Near Set IndexNear Set Index (NSI). For instance, given a set of Probe Functions probe functions $\mathcal{B}$, an image, and a segmentation of the image (labelled A), this frame can perform the perceptual erosion or dilation using $B = O_{/\sim_{\mathcal{B}}}$ as the structuring element. Also, the NSI is calculated if perceptual erosion was selected. A sample calculation using this frame is given in Fig. 7 and is obtained by the following steps:

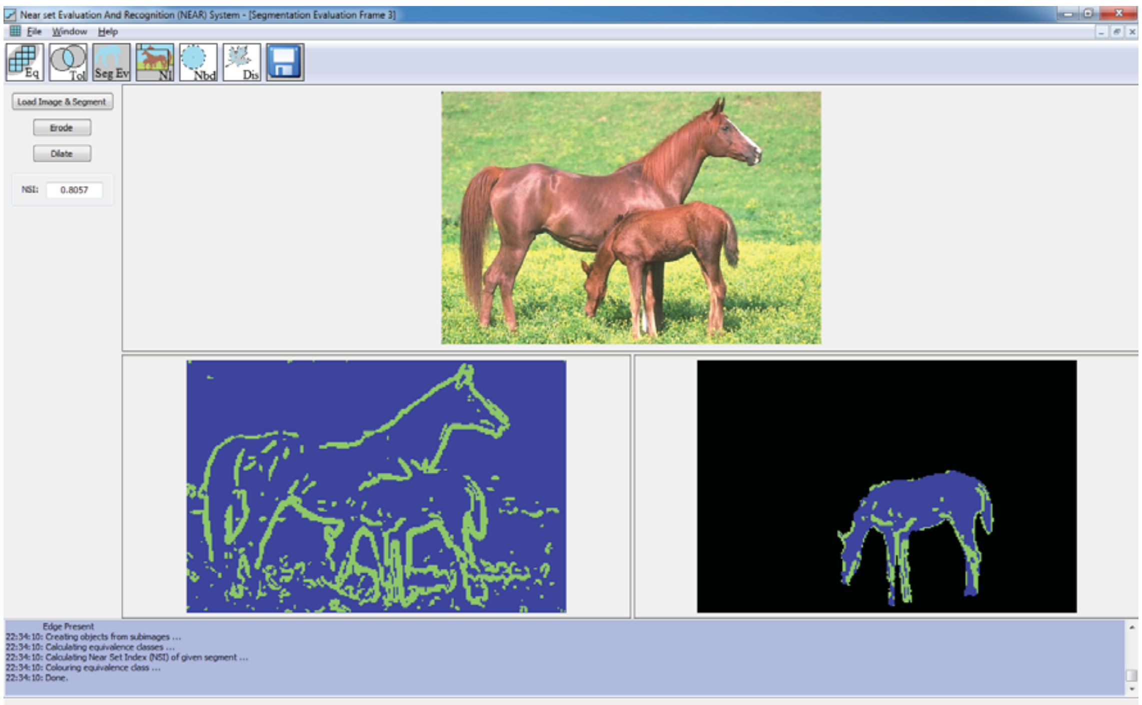


Figure 7: Sample run of the segmentation evaluation frame using a window size of 2×2, and $\mathcal{B} = \{\phi_{\text{Edge Present}}\}$ (Image shown in NEAR system used with permission [35]).

1. Click *Load Image & Segment* button.
2. Select an image click *Open*.
3. Select segmentation image and click *Open*. Image should contain only one segment and the segment must be white $(255, 255, 255)$ and the background must be black $(0, 0, 0)$. The image is displayed in the top frame, while the segment is displayed in the bottom right (make sure this is the case).
4. Click either *Erode* to perform perceptual erosion and segmentation evaluation, or *Dilate* to perform perceptual dilation (no evaluation takes place during dilation).
5. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.
6. Select number of features (maximum allowed is 24).
7. Select features.

8. Click *Run*.

The result is given in Fig. 7 where the bottom left window contains the an image of the equivalence classes where each colour represents a different class. The bottom right window contains either the erosion or dilation of the segmentation. Clicking on any of the three images will display the equivalence class containing the mouse click in the bottom right image. The NSI is also displayed on the left hand side (if applicable).

7 Near image frame

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

7.1 Evaluating a pair of images

The steps involved in comparing a pair of images are as follows, and sample output for this process is given in Fig. 8.

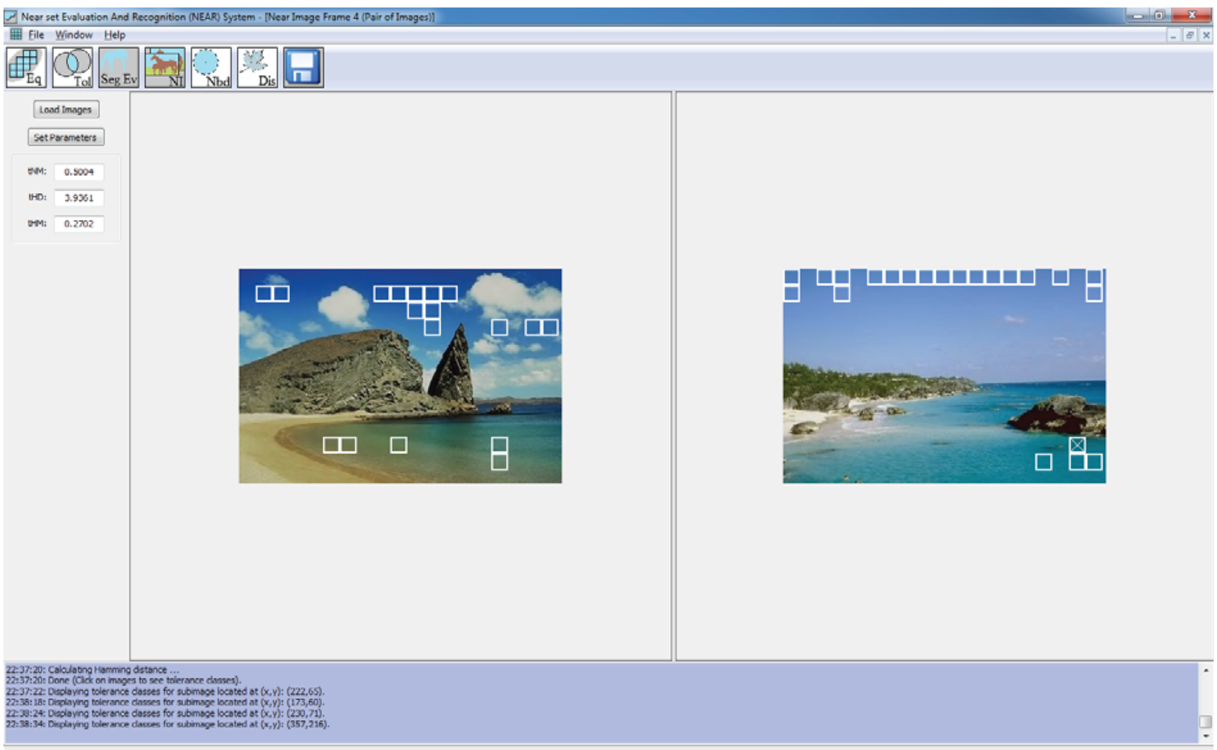


Figure 8: Sample run comparing a pair of images using a window size of 20×20, 18 features used to generate the results in this thesis, and $\varepsilon = 0.7$ (Image shown in NEAR system used with permission [38, 39]).

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

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

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

4. Click the *Set Parameters* button.

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

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

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

8. Select features.

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

10. Click *Run*.

The result is given in Fig. 8 where the left side contains the first image, and the right side contains the second image. Clicking in any of the two images will bring up a window that allows the user to view each tolerance class containing the subimage selected by the mouse. Further, the subimage containing the mouse click is marked with an 'X', and the subimages can be coloured white or black. Also, the similarity of the images is evaluated using the measures described in this thesis, where the results are displayed on the left hand side.

7.2 Comparing a query image with a directory of images

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

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

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

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

4. Click the *Set Parameters* button.

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

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

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

8. Select features.

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

10. Click *Run*.

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

8 Image neighbourhood and region of interest frame

The problem solved in this NEAR system frame is *how to measure the nearness of sets of neighbourhoods*. This frame is also used to measure the similarity of images, however, the user must target specific portions of each image by defining Regions of Interest (ROI). This part of the NEAR system constructs a collection of open neighbourhoods that cover a particular ROI. In effect, the NEAR system topologises a part of an image represented by a ROI. The following details the additional input parameters for this frame, a sample run, the algorithm used to produce the output, and a discussion the results.

Remark 1 The solution to the neighbourhood nearness problem stems from the work by M. Katětov [40] and S. Tiwari [41] on merotopic spaces. M. Katětov observed that merotopic spaces are obtained by topologising certain parts of a nonempty set. The term *mero* comes from the Greek word *meros* (part). Historically, a consideration of merotopic distance starts with a study of approach spaces (see, *e.g.*, [41–43]). Usually, an approach space distance function $\delta : X \times \mathcal{P}(X) \longrightarrow [0, \infty]$ that maps a member of a set and a subset (part) of a set to a number in $[0, \infty]$. For more about this problem in near set theory, see, *e.g.*, [12, 13, 44, 45].

8.1 Additional Parameters

The user is presented with the following additional parameters for this frame.

- Nearness Type: Determines how the Čech distance [46] is used to form the result. Indicating *Strong* results in looking for neighbourhoods with the smallest Čech distance, while setting the nearness type to *Weak* will select neighbourhoods with the largest Čech distance less than ε.
- Dilation Amount and Iteration: *Dilation Iteration* sets the number of iterations for performing calculations. After each calculation, the ROI is dilated (enlarged) by the number of pixels specified by the *Dilation Amount*.

Remark 2 For a source of examples of what is known ε-approach nearness on a nonempty set X, consider the gap distance function introduced by Čech in his 1936–1939 seminar on topology [46]. For an introduction to ε-approach nearness, see, *e.g.*, [10, 11].

The steps involved in comparing a pair of images are as follows, and sample output for this process is given in Fig. 9.

1. Select the *New Image Neighbourhood* icon, select File→New near image window, or press Alt+B.
2. Click *Load Images* button and select two images.
3. Click the *Set Parameters* button.
4. Select window size. The value is taken as the square root of the area for a square subimage, *e*.g., a value of 5 creates a subimage containing 25 pixels.

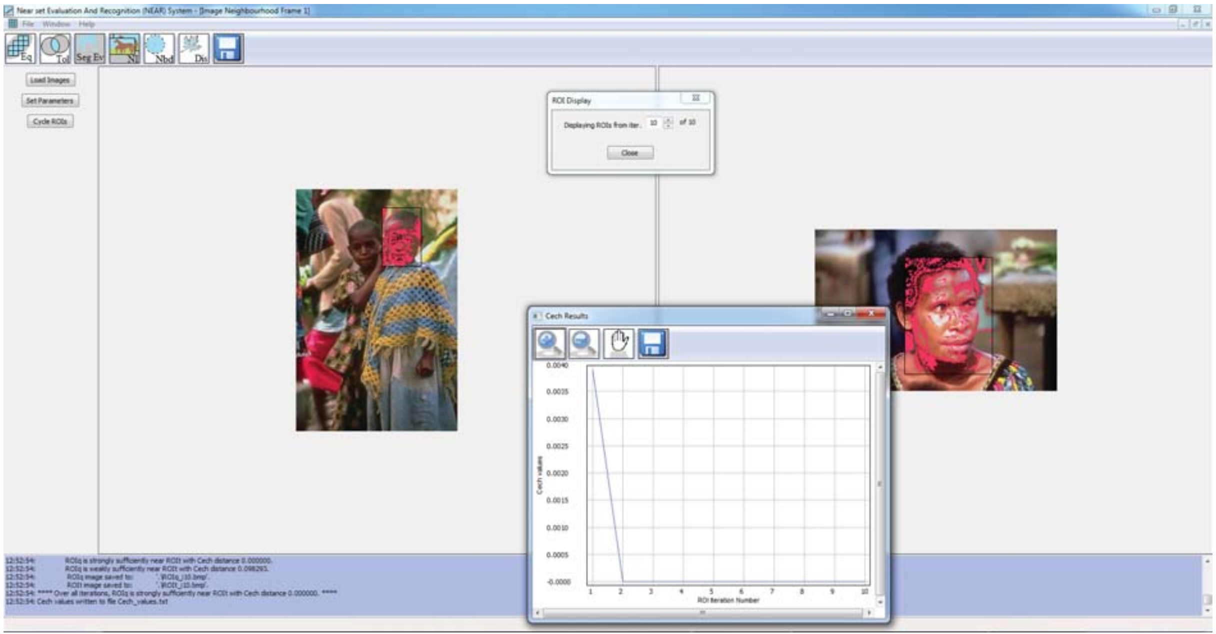


Figure 9: Sample run comparing a pair of images using a window size of 1×1, average R, G, and B features, $\varepsilon = 0.1$, strong nearness type, dilation amount of 1, and 10 dilation iterations.

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

6. Select the nearness type.

7. Select the dilation amount and number of iterations.

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

9. Select features.

10. Click *Run*.

Completing Step 10 causes the system to calculates probe function values only. Then, the user is prompted to select two ROIs, one for each image. This can be accomplished clicking and dragging with the mouse on each image. Note, the ROI for the left (resp. right) image is labelled ROI_q (ROI_t). Once the second ROI has been selected, the system will calculated the similarity of images according to the following algorithm.

1. Initialize the output by setting $bROI_q \leftarrow \emptyset$ and $bROI_t \leftarrow \emptyset$.

2. Initialize the smallest distance, $sd \leftarrow \infty$.

3. Go through all subimages of ROI_q. By default, the window size is set to 1. Thus, with default settings, the word *pixels* could replace *subimages* in the following steps.

4. For each subimage, $p_q \in ROI_q$ perform the following:.

   (a) Find $N(p_q)$.

   (b) Look for $p_t \in ROI_t$ such that $p_q \cong_{\mathcal{B},\epsilon} p_t$.

(c) If a subimage p_t can be found in Step 4b, then find $N(p_t)$. Otherwise, go back to Step 4.

(d) Calculate the Čech distance between $N(p_q)$ and $N(p_t)$ defined as

$$D_{\rho_{\|\cdot\|}}(N(p_q), N(p_t)) = \inf\{\rho_{\|\cdot\|}(a, b) : a \in N(p_q), b \in N(p_t)\},$$

where

$$\rho_{\|\cdot\|}(a, b) = \sum_{i=1}^{l} |\phi_i(a) - \phi_i(b)|.$$

(e) If $D_{\rho_{\|\cdot\|}}(N(p_q), N(p_t)) < sd$, then set $bROI_q \leftarrow N(p_q)$ and $bROI_t \leftarrow N(p_t)$.

(f) Else, if $D_{\rho_{\|\cdot\|}}(N(p_q), N(p_t)) == sd$, then $bROI_q \cup N(p_q)$ and $bROI_t \cup N(p_t)$.

(g) Loop back to Step 4.

Note, the above algorithm is for nearness type *Strong*. If nearness type *Weak* is selected, the following steps need to be changed.

- Replace Step 2 with: Initialize the largest distance, $ld \leftarrow 0$, and replace all instances of sd with ld.
- Replace the test in Step 4e with: $D_{\rho_{\|\cdot\|}}(N(p_q), N(p_t)) > ld$ && $D_{\rho_{\|\cdot\|}}(N(p_q), N(p_t)) \leq \varepsilon$.

Finally, the above algorithm is repeated *Dilation Iteration* times, where, on each iteration, the ROIs are increased by *Dilation Amount* pixels.

The result of a sample run is given in Fig. 9. After the calculations terminate, a pop-up window is presented to the user to allow them to cycle through the best ROIs for each iteration. If this window is closed, it can be reopened by clicking on the *Cycle ROIs* button. A plot is also present to the user containing the best Čech values after each iteration. Finally, all the best ROIs are saved as images in the same directory as the exe.

Figure 10: Sample run of the feature display frame (Image shown in NEAR system used with permission [35]).

9 Feature display frame

This frame is used to display the output of processing an image with a specific Probe Function probe function. An example using this frame is given in Fig. 10 and is obtained by the following steps:

1. Click *Load Image* button and select an image.
2. Select features.
3. Select probe function.
4. Click *Display feature*.

References

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

[2] ——, “Near set index in an objective image segmentation evaluation framework,” in *Proceedings of the GEOgraphic Object Based Image Analysis: Pixels, Objects, Intelligence*, 2008, pp. 1–8.

[3] J. F. Peters and S. Ramanna, *Affinities between perceptual granules: Foundations and Perspectives*. Berlin: Springer-Verlag, 2009, pp. 49–66.

[4] J. F. Peters, *Discovery of perceptually near information granules*. Hersey, N.Y., USA: Information Science Reference, 2009, p. *in press*.

[5] C. Henry and J. F. Peters, “Perception image analysis,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 3/4, pp. 271–281, 2010.

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

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

[8] ——, “Corrigenda and addenda: Tolerance near sets and image correspondence,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 5, pp. 310–318, 2010.

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

[10] J. Peters and S. Naimpally, “Approach spaces for near families,” *Gen. Math. Notes*, vol. 2, no. 1, pp. 159–164, 2011.

[11] J. Peters and S. Tiwari, “Approach merotopies and near filters. Theory and application,” *Gen. Math. Notes*, vol. 3, no. 1, pp. 15 Pages, to appear, 2011.

[12] J. Peters, "Sufficiently near sets of neighbourhoods," in *Lecture Notes in Artificial Intelligence 6954*, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds. Springer, 2011, pp. 17–24.

[13] ——, "ε-Near collections," in *Lecture Notes in Artificial Intelligence 6954*, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds. Springer, 2011, pp. 533–542.

[14] J. Peters and S. Naimpally, "Applications of near sets," *Amer. Math. Soc. Notices*, 2012, *to appear*.

[15] J. Peters and S. Tiwari, "Completing extended metric spaces: An alternative approach," *Applied Math. Letters*, 2012, *to appear*.

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

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

[18] M. Pavel, ""shape theory" and pattern recognition," *Pattern Recognition*, vol. 16, no. 3, pp. 349–356, 1983.

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

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

[21] ——, "Classification of perceptual objects by means of features," *International Journal of Information Technology & Intelligent Computing*, vol. 3, no. 2, pp. 1 – 35, 2008.

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

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

[24] ——, *Semantics of Vague Concepts*. London/NY: Plenum Pres, 1985, pp. 465–482.

[25] E. C. Zeeman, *The topology of the brain and the visual perception*. New Jersey: Prentice Hall, 1965, pp. 240–256.

[26] 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.

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

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

[29] W. Bartol, J. Miró, K. Pióro, and F. Rosselló, "On the coverings by tolerance classes," *Inf. Sci. Inf. Comput. Sci.*, vol. 166, no. 1-4, pp. 193–211, 2004.

[30] S. N. Gerasin, V. V. Shlyakhov, and S. V. Yakovlev, "Set coverings and tolerance relations," *Cybernetics and Sys. Anal.*, vol. 44, no. 3, pp. 333–340, 2008.

[31] M. Schroeder and M. Wright, "Tolerance and weak tolerance relations," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 11, pp. 123–160, 1992.

[32] Y. A. Shreider, "Tolerance spaces," *Cybernetics and Systems Analysis*, vol. 6, no. 12, pp. 153–758, 1970.

[33] A. Skowron and J. Stepaniuk, "Tolerance approximation spaces," *Fundamenta Informaticae*, vol. 27, no. 2/3, pp. 245–253, 1996.

[34] Z. Zheng, H. Hu, and Z. Shi, "Tolerance relation based granular space," *Lecture Notes in Computer Science*, vol. 3641, pp. 682–691, 2005.

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

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

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

[38] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

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

[40] M. Katětov, "On continuity structures and spaces of mappings," *Comment. Math. Univ. Carolinae*, vol. 6, pp. 257–278, 1965.

[41] S. Tiwari, "Some aspects of general topology and applications. Approach merotopic structures and applications, supervisor: M. khare," Ph.D. dissertation, Department of Mathematics, Allahabad (U.P.), India, Jan. 2010.

[42] R. Lowen, *Approach Spaces: The Missing Link in the Topology-Uniformity-Metric Triad*. Oxford, UK: Oxford Mathematical Monographs, Oxford University Press, 1997, viii + 253pp.

[43] R. Lowen, D. Vaughan, and M. Sioen, "Completing quasi metric spaces: an alternative approach," *Houstan J. Math.*, vol. 29, no. 1, pp. 113–136, 2003.

[44] C. Henry and S. Ramanna, "Parallel computation in finding neighbourhoods," in *Lecture Notes in Artificial Intelligence 6954*, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds. Springer, 2011, pp. 523–532.

[45] M. Wolski, "Gauges, pregauges and completions: Some theoretical aspects of near and rough set approaches to data," in *Lecture Notes in Artificial Intelligence 6954*, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds. Springer, 2011, pp. 543–552.

[46] E. Čech, *Topological Spaces, revised Ed. by Z. Frolik and M. Katětov*. London: John Wiley & Sons, 1966, 893pp.