Near set Evaluation And Recognition (NEAR) System

Christopher Henry
James F. Peters, Supervisor
{chenry,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-2009-015
May 27, 2009

University of Manitoba

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

Project no. 2009-015

Near set Evaluation And Recognition (NEAR) System

Christopher Henry
James F. Peters, Supervisor
{chenry,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-2009-015

May 27, 2009

Abstract

This report presents 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.

1 Introduction

The goal of the NEAR system is to demonstrate applications of the near set theory presented in [1–9]. 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 (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 five major tasks, namely, displaying equivalence and tolerance classes for an image, performing segmentation evaluation, measuring the nearness of two images, and displaying the output of processing an image using an individual probe functions. 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 5-8 describe the operation of the GUI.

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.

[1]Parts of the Graphical User Interface (GUI) were inspired by the GUI reported in [10] and the wxWidgets example in [11].

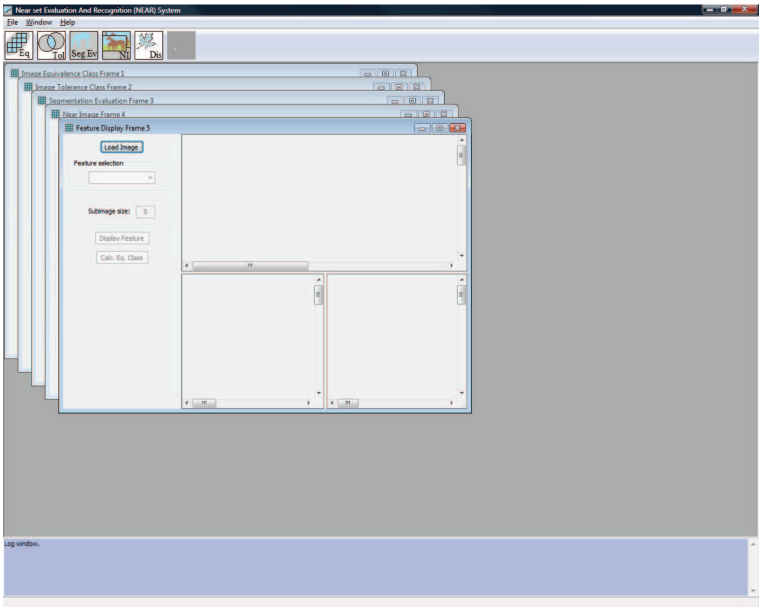


Figure 1: NEAR system GUI.

2 Near sets

Near set theory focuses on sets of perceptual objects with matching descriptions. Specifically, let O represent the set of all 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 that describes the object x. The notion of a probe function in near sets is inspired by Monique Pavel [12], where a probe function that is invariant relative to a transformation of the images results in matching function (feature) values. In a 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 [13–15]. Furthermore, a set $\mathbb{F}$ can be defined that represents all the probe functions used to describe an object x. Next, a perceptual information 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 *[13] 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 [16]. 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 let $X, Y \subseteq O$. A set X is weakly near to a set Y within the perceptual system $\langle O, \mathbb{F}\rangle$ ($X \bowtie_{\mathbb{F}} Y$) 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 Defn 2, 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.

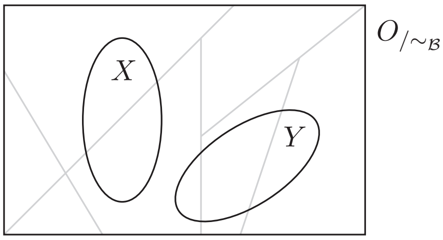


Figure 2: Example of Defn. 2.

Defn. 2 can be used to define a Nearness Measure (NM) between two sets X and Y [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

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

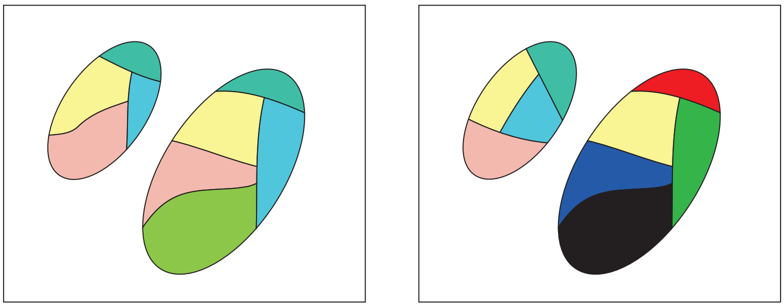
to denote the portion that belongs to Y. Further, let the sets X and Y be weakly near each other using Defn. 2. Then, a NM between X and Y is given by

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

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.

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 [17] (see, also, [18]), *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 [19]. 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 [20] observes that main idea underlying tolerance theory comes from Henri Poincaré [21]. 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) [20]. 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 [22] 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, 8, 9, 23–28]).

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, 8] 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

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

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 Table 1 that where $|\phi(x_i)| = 1$. Letting $\epsilon = 0.1$ gives the following tolerance classes:

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

Observe that each object in a tolerance class satisfies the condition $\parallel \phi(x) - \phi(y) \parallel \leq \epsilon$, 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.

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

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

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 [29] 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.* Section 4 for examples of probe functions used in the NEAR system or [30] 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. 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.

(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 blue background, the orange octagon interior, and the green squares along the diagonals. The green 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 Probe functions

This section describes the probe functions used in the NEAR system, and gives example NEAR system output images processed using these probe functions.

4.1 Average greyscale value

Conversion from RGB image to greyscale is accomplished using Magick++, the object-orientated C++ API to the ImageMagick image-processing library [31]. First, an RGB image is converted to greyscale using

$$Gr = 0.299R + 0.587G + 0.114B, \tag{4}$$

and then the values are averaged over each subimage. An example is given in Fig. 5.

(a) (b) (c)

Figure 5: Example of average greyscale probe function: (a) Original image [32], (b) average greyscale over subimages of size 5×5, and (c) average greyscale over subimages of size 10×10.

4.2 Normalized RGB

The normalized RGB values is a feature described in [30], and the formula is given by

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

where the values R_T, G_T, and B_T are respectively the sum of R, G, B components of the pixels in each subimage, and $X \in [R_T, G_T, B_T]$. See Fig. 6 for an example using this probe function. Note, these images were produces by finding the normalized value and multiplying it by 255.

(a) (b) (c)

Figure 6: Example of normalized RGB probe function: (a) Original image [29], (b) normalized R over subimages of size 5×5, and (c) normalized R over subimages of size 10×10.

4.3 Shannon's entropy

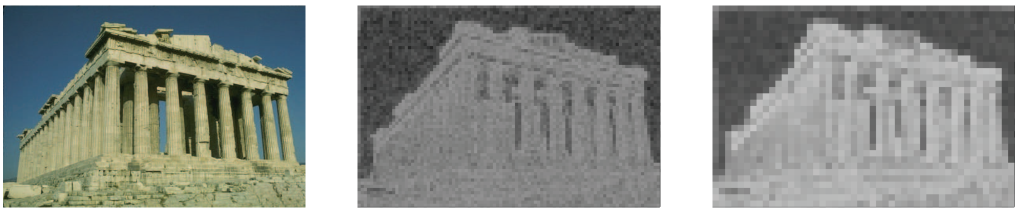
Shannon introduced entropy (also called information content) as a measure of the amount of information gained by receiving a message from a finite codebook of messages [33]. The idea was that the gain of information from a single message is proportional to the probability of receiving the message. Thus, receiving a message that is highly unlikely gives more information about the system than a message with a high probability of transmission. Formally, let the probability of receiving a message i of n messages be p_i, then the information gain of a message can be written as

$$\Delta I = \log(1/p_i) = -\log(p_i), \tag{5}$$

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

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

This concept can easily be applied to the pixels of a subimage. First, the subimage is converted to greyscale using Eq. 4. Then, the probability of the occurrence of grey level i can be defined as $p_i = h_i/T_s$, where h_i is the number of pixels that take a specific grey level in the subimage, and T_s is the total number of pixels in the subimage. Information content provides a measure of the variability of the pixel intensity levels within the image and takes on values in the interval $[0, \log_2 L]$ where L is the number of grey levels in the image. A value of 0 is produced when an image contains all the same intensity levels and the highest value occurs when each intensity level occurs with equal frequency [34]. An example of this probe function is given in Fig. 7. Note, these images were formed by multiplying the value of Shannon's entorpy by 32 since $L = 256$ (thus giving a maximum value of 8).



(a) (b) (c)

Figure 7: Example of Shannon's entropy applied to images: (a) Original image [29], (b) Shannon's entropy applied to subimages of size 5×5, and (c) Shannon's entropy applied to subimages of size 10×10.

4.4 Pal's entropy

Work in [33, 35] shows that Shannon's definition of entropy has some limitations. Shannon's definition of entropy suffers from the following problems: it is undefined when $p_i = 0$; in practise the information gain tends to lie at the limits of the interval $[0, 1]$; and statistically speaking, a better measure of ignorance is 1 - p_i rather than $1/p_1$ [33]. As a result, a new definition of entropy can be defined with the following desirable properties:

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

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

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

P4: $k_2 < k_1$.

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

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

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

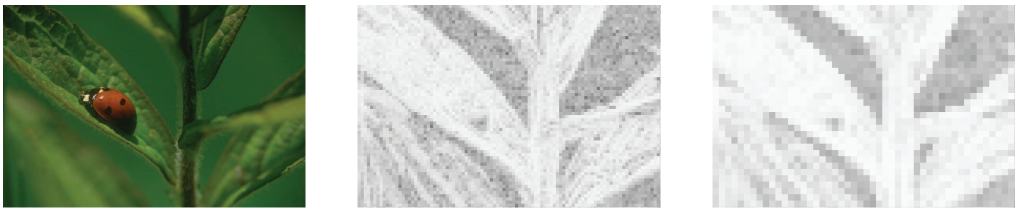
With these in mind, [33] defines the gain in information from an event as

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

which gives a new measure of entropy as

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

Pal's version of entropy is given in Fig. 8. Note, these images were formed by first converting the original image to greyscale, calculating the entropy for each subimage, and multiplying this value by 94 (since the maximum of H is $e^{1-1/256}$).



(a) (b) (c)

Figure 8: Example of Pal's entropy applied to images: (a) Original image [29], (b) Pal's entropy applied to subimages of size 5×5, and (c) Pal's entropy applied to subimages of size 10×10.

4.5 Edge based probe functions

The edge based probe functions integrated in the NEAR system incorporate an implementation of Mallat's Multiscale edge detection method based on Wavelet theory [36]. The idea is that edges in an image occur at points of sharp variation in pixel intensity. Mallat's method calculates the gradient of a smoothed image using Wavelets, and defines edge pixels as those that have locally maximal gradient magnitudes in the direction of the gradient.

Formally, define a 2-D smoothing function $\theta(x, y)$ such that its integral over x and y is equal to 1, and converges to 0 at infinity. Using the smoothing function, one can define the functions

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

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

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

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

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

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

$$W_s^1 f(x, y) = f * \psi_s^1(x, y) \quad \text{and} \quad W_s^2 f(x, y) = f * \psi_s^2(x, y),$$

which can also be written as

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

Finally, edges can be detected by calculating the modulus and angle of the gradient vector defined respectively as

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

and

$$A_s f(x, y) = \text{argument}(W_s^1 f(x, y) + iW_s^2 f(x, y)),$$

and then finding the modulus maximum defined as pixels with modulus greater than the two neighbours in the direction indicated by $A_s f(x, y)$ (see [36] for specific implementation details). Examples of Mallatt's edge detection method obtained using the NEAR system are given in Fig. 9.

4.5.1 Edge present

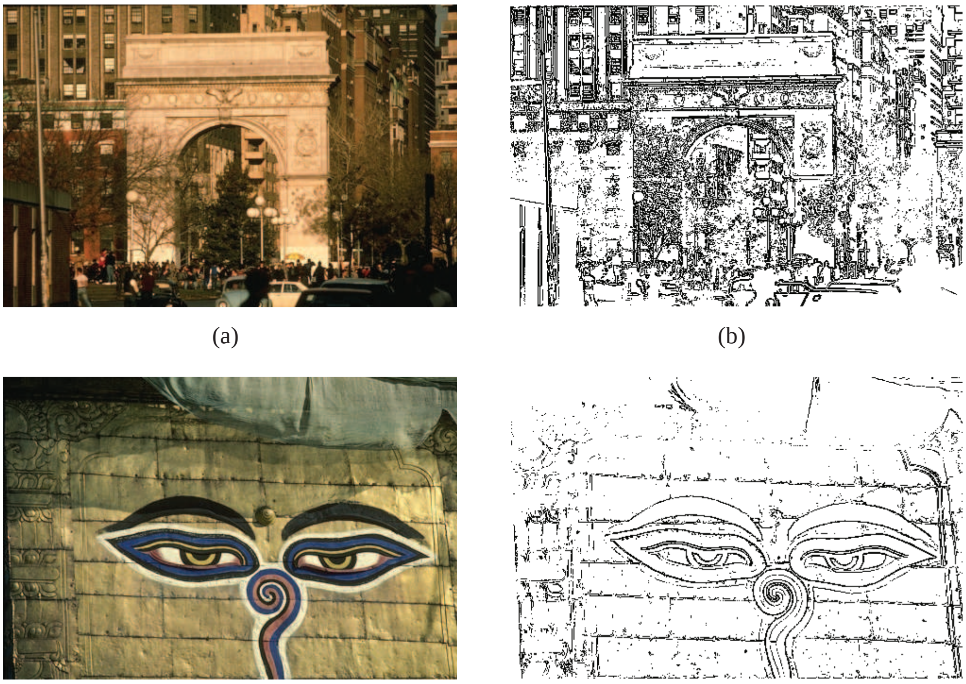
This prob function simply returns true if there is an edge pixel contained in the subimage (see, *e.g.*, Fig. 10).

4.5.2 Number of edge pixels

This probe function returns the total number of pixels in a subimage belonging to an edge (see, *e.g.*, Fig. 11).

4.5.3 Edge orientation

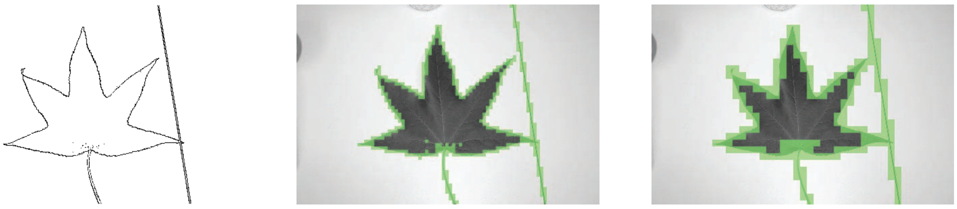
This probe function returns the average orientation of subimage pixels belonging to an edge (see, *e.g.*, Fig. 12).



(a) (b)

(c) (d)

Figure 9: Example of NEAR system edge detection using Mallat's method: (a) Original image, (b) edges obtained from (a), (c) original image, and (d) obtained from (c).



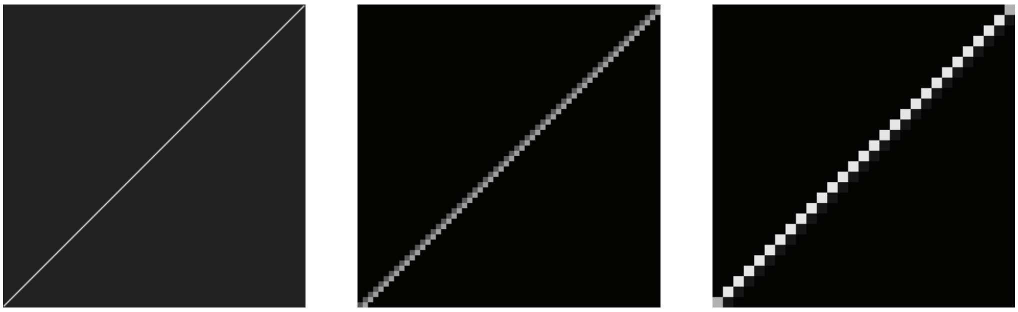
(a) (b) (c)

Figure 10: Example of edge present probe function: (a) Edges obtained from Fig. 5(a), (b) Application to image with subimages of size 5×5, and (c) Application to image with subimages of size 10×10.

5 Equivalence class frame

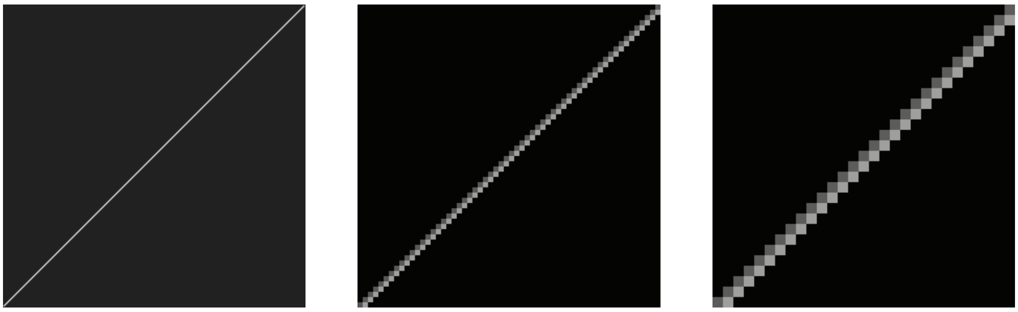
This frame calculates equivalence classes using the Indiscernibility relation of Defn. 1, *i*.e., given an image X, it will calculate $X_{/\sim_{\mathcal{B}}}$ where the objects are subimages of X. See Section 3 for an explanation of the theory used to obtain these results. A sample calculation using this frame is given in Fig. 13 and was obtained by the following steps:

1. Click *Load Image* button.
2. Select number of features (maximum allowed is four).
3. Select features (see Section 4 for a list of probe functions).



(a) (b) (c)

Figure 11: Example of number of edge pixels probe function: (a) Original image, (b) Application to image with subimages of size 5×5, and (c) Application to image with subimages of size 10×10.



(a) (b) (c)

Figure 12: Example of average orientation probe function: (a) Original image, (b) Application to image with subimages of size 5×5, and (c) Application to image with subimages of size 10×10.

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 of 25 pixels.

5. Click *Run*.

The result is given in Fig. 13 where the bottom left window contains an image of the equivalence classes where 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.

6 Tolerance class frame

This frame calculates tolerance classes using the Tolerance relation of Defn. 3, *i*.e., given an image X, it will calculate $X_{/\cong_{\mathcal{B}}}$ where the objects are subimages of X. This approach is similar to the one given in Section 3 with the exception that Defn. 1 is replaced with Defn. 3. A sample calculation using this frame is given in Fig. 14 and was obtained by the following steps:

1. Click *Load Image* button.

2. Select number of features (maximum allowed is four).

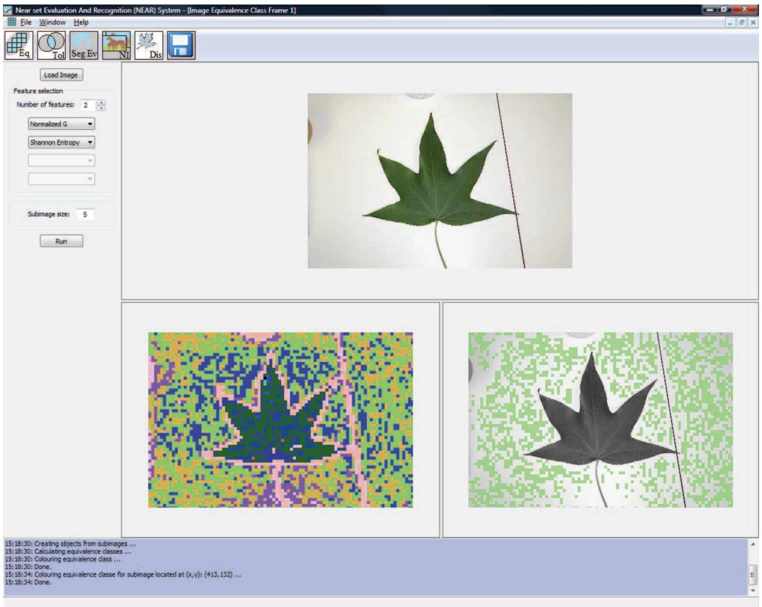


Figure 13: Sample run of the equivalence class frame using a window size of 5×5 and $\mathcal{B} = \{\phi_{\text{NormG}}, \phi_{\text{H}_{\text{Shannon}}}\}$.

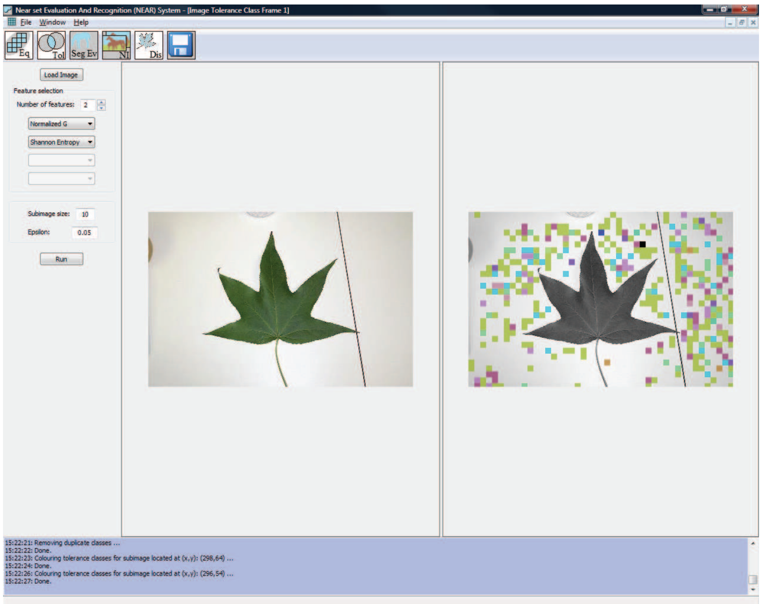


Figure 14: Sample run of the tolerance class frame using a window size of 10×10, $\mathcal{B} = \{\phi_{\text{NormG}}, \phi_{\text{H}_{\text{Shannon}}}\}$, and $\epsilon = 0.05$.

3. Select features (see Section 4 for a list of probe functions).

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 of 25 pixels.

5. Select ϵ, a value in the interval $[0, 1]$.

6. Click *Run*.

The result is given in Fig. 14 where the left side is the original image, and the right side is used to display the tolerance classes. Since the tolerance relation does not partition an image, the tolerance classes are displayed upon request. For instance, by clicking on either of the two images, all the tolerance classes are displayed that are within ϵ of the subimage containing the coordinates of the mouse click. Further, the subimage containing the mouse click is coloured black.

7 Segmentation evaluation frame

This frame performs segmentation evaluation using perceptual morphology as described in [2, 6], where the evaluation is labelled the Near Set Index (NSI). For instance, given a set of probe functions $\mathcal{B}$, and an image A, this frame can perform the perceptual erosion or dilation using $B = O_{/\sim_{\mathcal{B}}}$ as the SE. Also, the NSI is calculated if perceptual erosion was selected. A sample calculation using this frame is given in Fig. 15 and was obtained by the following steps:

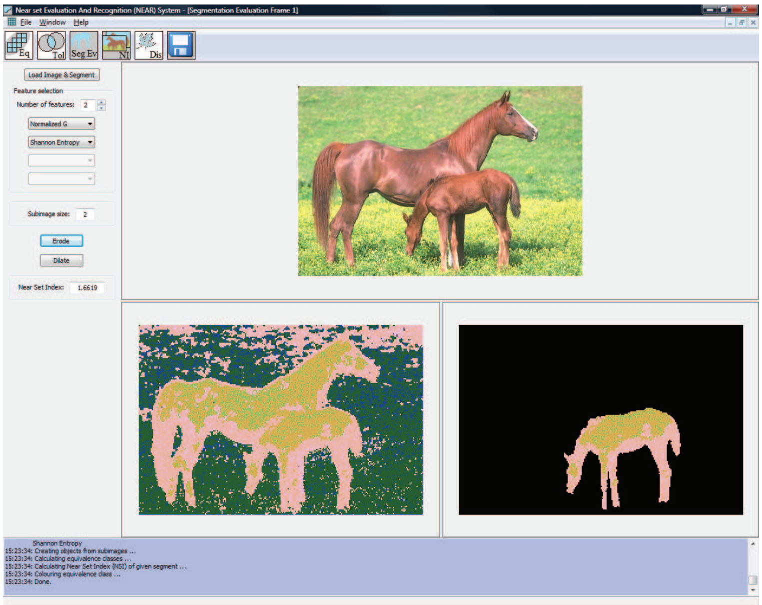


Figure 15: Sample run of the segmentation evaluation frame using a window size of 2×2, and $\mathcal{B} = \{\phi_{\text{NormG}}, \phi_{H_{\text{Shannon}}}\}$.

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. Select number of features (maximum allowed is four).
5. Select features (see Section 4 for a list of probe functions).

6. 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 of 25 pixels.

7. Click *Erode* to perform perceptual erosion and segmentation evaluation. Click *Dilate* to perform perceptual dilation (no evaluation takes place during dilation).

The result is given in Fig. 15 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 segments erosion or dilation. 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.

8 Near image frame

This frame is used to calculate the nearness of two images using the nearness measure from Eq. 1 defined in Section 2. A sample calculation using this frame is given in Fig. 16 and was obtained by the following steps:

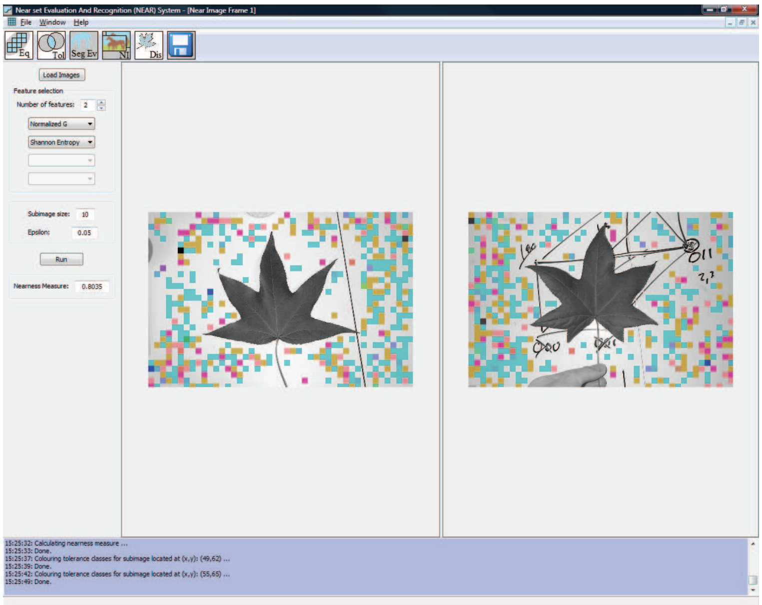


Figure 16: Sample run of the near image frame using a window size of 10×10, $\mathcal{B} = \{\phi_{\text{NormG}}, \phi_{\text{H}_{\text{Shannon}}}\}$, and $\epsilon = 0.05$.

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

2. Select number of features (maximum allowed is four).

3. Select features (see Section 4 for a list of probe functions).

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 of 25 pixels.

5. Select ϵ, a value in the interval $[0, 1]$.

6. Click *Run*.

The result is given in Fig. 16 where the left side contains the first image, and the right side contains the second image. Clicking in any of the two images will display the tolerance classes from both images near to the subimage selected by the mouse click. The subimage matching the coordinates of the mouse click is coloured black and all subimages that are near to the black subimage are displayed using a different colour for each class. The NM is also displayed on the left hand side.

9 Feature display frame

This frame is used to display the output of processing an image with a specific probe function. A sample calculation using this frame is given in Fig. 17 and was obtained by the following steps:

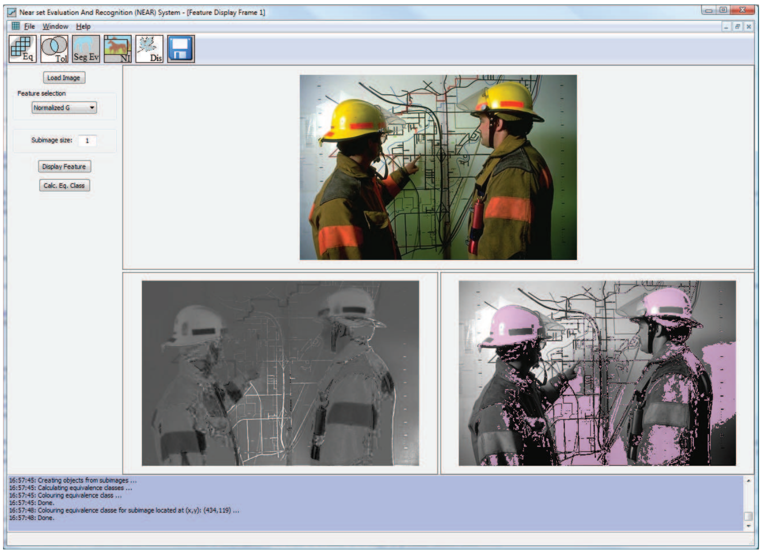


Figure 17: Sample run of the feature display frame.

1. Click *Load Image* button and select an image.
2. Select features (see Section 4 for a list of probe functions).
3. Select probe function
4. Click *Display feature*.

References

[1] C. Henry and J. F. Peters, "Image pattern recognition using approximation spaces and 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 Artificial Intelligence*, 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*, University of Calgary, Alberta, 2008, p. to appear.

[3] J. F. Peters, “Affinities between perceptual granules: Foundations and perspectives,” in *Human-Centric Information Processing Through Granular Modelling*, A. Bargiela and W. Redrycz, Eds. Berline: Springer-Verlag, 2008.

[4] ——, “Discovery of perceptually near information granules,” in *Novel Developements in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation*, J. T. Yao, Ed. Hersey, N.Y., USA: Information Science Reference, 2008, p. to appear.

[5] C. Henry and J. F. Peters, “Perception based image classification,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, p. submitted, 2008.

[6] ——, “Perceptual image analysis,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, p. *to appear*, 2008.

[7] J. F. Peters and P. Wasilewski, “Foundations of near sets,” *Elsevier Science*, pp. submited, revised, resubmitted, 2008.

[8] J. F. Peters, “Tolerance near sets and image correspondence,” *International Journal of Bio-Inspired Computation*, vol. 4, no. 1, p. *in press*, 2009.

[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*, p. accepted, 2009.

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

[11] wxWidgets, “wxwidgets cross-platform gui library v2.8.9,” 2009, www.wxwidgets.org.

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

[13] J. F. Peters, “Near sets. special theory about nearness of objects,” *Fundamenta Informaticae*, vol. 76, pp. 1–27, 2007.

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

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

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

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

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

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

[20] Sossinsky1986, "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.

[21] H. Poincaré, *Mathematics and Science: Last Essays, trans. by J. W. Bolduc*. N. Y.: Kessinger, 1913.

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

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

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

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

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

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

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

[29] 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.

[30] 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.

[31] Magick++, "Imagemagick image-processing library," 2009, www.imagemagick.org.

[32] M. Weber, "Leaves dataset," 1999, url: www.vision.caltech.edu/archive.html.

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

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

[35] N. R. Pal and S. K. Pal, "Some properties of the exponential entropy," *Information Sciences*, vol. 66, pp. 119–137, 1992.

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