Cybernetics and Systems: An International Journal, XX:YYY-ZZZ
Copyright © Taylor & Francis Group, LLC
ISSN: XXXX-XXX print/XXX-XXX online
DOI: XX.XXXX.XXXXXXX.XXXXXX

Neighbourhood-based Vision Systems

CHRISTOPHER J. HENRY and JAMES F. PETERS
Department of Electrical and Computer Engineering, University of Manitoba, Canada

The problem presented in this paper is how to find similarities between digital images useful in design cybernetic vision systems. The solution to this problem stems from a neighbourhood based vision system. A neighbourhood is viewed in the context of a covering of a visual space defined by tolerance relations. A consideration of neighbourhoods and tolerance classes leads to a highly practical tolerance near set approach in vision systems. The contribution of this article is an algorithm for finding tolerance classes, a new measure for quantifying the similarity between tolerance classes, and a practical application of the tolerance space approach.

KEYWORDS: metric, tolerance space, vision system

INTRODUCTION

This article presents a new algorithm for finding tolerance classes that represents a first step toward the application of the tolerance near set approach in vision systems. Tolerance classes are groups of objects that have similar features, and are defined with respect to a tolerance space (introduced by Zeeman during the 1960s (Zeeman 1965; Zeeman and Buneman 1968)). Tolerance spaces are related to both physical and visual spaces, which are the domain of visual systems. As defined by (Wagner 2006), a physical space is the space revealed by instrumentation and is independent of the observer, while a visual space is a non-objective interpretation by the observer of the physical space based on the perception of external stimuli. For instance, it is impossible to determine the accuracy of a person's judgement of the properties of a physical room, whereas these values can be obtained exactly through measurement. In the former case, for example, it could be that some

Address correspondence to James F. Peters, Department of Electrical and Computer Engineering, 75A Chancellor's Circle, University of Manitoba, Winnipeg MB, R2T 5V6 Canada. E-mail: jfpeters@ee.umanitoba.ca

shadows caused the observer to misjudge the actual shape of the room. Wagner's definition of a visual space describes the environment of a visual system, which is an artificial approach to mimicking the human visual system. Generally, these systems consist of a sensing device (such as a camera) that generates an image, or stream of images, as well as a processing unit that interprets the images and makes decisions. As a result, a visual system operates in a visual space since the judgements are based only on the output of the sensors.

A tolerance space relates objects to one another based on similarities of appearance or description rather than on equivalence. Tolerance spaces were inspired by the imperfections of the senses (see, *e.g.*, (Henry 2010b)) and are inherent to visual spaces, a fact observed by Zeeman when he noted that a single eye cannot identify a 2D Euclidean space because the Euclidean plane has an infinite number of points. Rather, we see things only within a certain tolerance (Zeeman 1965; Henry 2010b), which is implicit in the definition of a visual space given above. Further, tolerance near sets are disjoint set of objects that have similar descriptions. The focus of this article is to present a new algorithm for finding tolerance classes, and to demonstrate the use of tolerance near sets for finding similarities between digital images, an approach that can be used in the design of vision systems. Also, a practical application of the tolerance space approach is presented by way of a method for determining the resemblance between images extracted from videos sequences showing non-arthritic and arthritic hand-finger movements. This article is organized as follows: The next section presents related works, and is followed by a section reviewing tolerance spaces and tolerance near set theory. Next, the nearness metrics are presented for evaluating the similarity of images, succeeded by a discussion on applications to vision systems and the experimental results.

RELATED WORKS

The results presented in this paper are obtained from a tolerance near set metric for measuring the similarity of images called the tolerance nearness measure (tNM), and a new measure called the tolerance Hamming measure (tHM). Further, we propose, that these results suggest that the presented metric would be useful in vision systems. The nearness measure was crated out of a need to determine the degree that near sets resemble each other, a need which arose during the application of near set theory to the practical applications of image correspondence and content-based image retrieval. Specifically, the nearness measure was introduced by Henry and Peters in (Hassanien et al. 2009, Section VII.A, pp. 964-965) where it was given as a solution to the problem of image resemblance of MRI images. At the same time, the nearness measure was also introduced in (Henry and Peters 2009). Since then, the notation of the nearness measure has been refined (as reported in (Henry and Peters 2010)) and it has been applied to the problems of image resemblance and correspondence (Meghdadi et al. 2009; Peters et al. 2009; Peters and Puzio 2009; Peters 2009c, 2010, 2009b,a) which is closely related to content-based image retrieval (Henry and Peters 2010), *i.e.* the problem of retrieving images based on the perceived objects within the image rather than based on semantic terms associated with the image. The nearness measure has also been applied to patterns of observed swarm behaviour stored in tables called ethograms, where the goal is to measure the resemblance between the behaviours of different swarms (Ramanna and Meghdadi 2009).

Metrics are essential to measuring the similarity of images (which is common in content-based image retrieval). For example, (Yang et al. 2010) identify a problem in the medical community that a retrieval system needs to take into account both visual and semantic features. For instance, a tire has the same visual appearance as a donut, but they do not belong in the same category. This problem is addressed by using a training set that includes side information, *i.e.* information on the semantic relationship between training data. Then, distance metric learning is performed to learn a distance function from training data. Another example of a similarity metrics is presented in (Zheng et al. 2003) where image features of colour histogram, image texture, Fourier coefficients, and wavelet coefficients are combined into an image signature and the metric is measured as the vector dot product between signatures. Other examples of metrics include the multiresolution tangent distance for use in image alignment applied to the problems of image retrieval and mosaic creation (Vasconcelos and Lippman 2005), or the metric presented in (Mojsilović and Rogowitz 2004) that relates low-level image features to high-level image semantics.

As was mentioned in the introduction, a vision system is one that that mimics the power and capability of the human sense of sight (*i.e.* the ability to detect light) combined with some type of cognition, perception, or interpretation of the stimulus. We propose that the approach to measuring the similarities of images presented in this article could be useful in the design of a visual system. While a complete survey of vision systems is outside the scope of this article, the following examples are presented to give an idea as to the various types of vision systems. (Bakhtari and Benhabib 2007) present a vision system with the goal to position multiple cameras to identify and track multiple objects of interest in dynamic multiobject environments. (Hussmann and Liepert 2009) use 3-D time of flight (rather than stereo vision) to control a robot in a simulation of loading a container ship. The visual system generates range data to the objects that need to be loaded onto a ship, and performs segmentation of an image generated from range date to identify the centre of gravity and the rotation angle (information necessary to grab the simulated containers). Finally, another example of a vision system is the CogV system presented in (Zhang and Tay 2009) which mimics saccade and vergence movements in a binocular camera system to identify objects of interest in the field of view.

PRELIMINARIES

The approach taken in this article is made possible using tolerance spaces introduced by E.C. Zeeman during the 1960s (Zeeman 1965; Zeeman and Buneman 1968) as well as recent work on tolerance near sets (Peters 2009c; Hassanien et al. 2009; Henry 2010b) and hand images (Ferrer et al. 2009). The term *tolerance space* was coined by Zeeman in 1962 in modelling visual perception with tolerances (Zeeman 1965). A tolerance space $\langle X, \simeq \rangle$ consists of a set X and a binary relation $\simeq$ on X ($\simeq \subset X \times X$) that is reflexive (for all $x \in X$, $x \simeq x$) and symmetric (for all $x, y \in X$, if $x \simeq y$, then $y \simeq x$) but transitivity of $\simeq$ is not required. Every tolerance relation determines some specific subsets of the space $\langle X, \simeq \rangle$. A set $A \subset X$ is a *preclass* of the relation $\simeq$ if and only if for all $x, y \in A$, $x \simeq y$ (*i.e.* $A \times A \subset \simeq$). All precalsses of a given tolerance relation are naturally ordered by inclusion. Preclasses maximal with respect to inclusion are called *tolerance classes*.

A specific tolerance relation can be defined as follows. Let $\mathcal{B}$ denote a set of real-valued functions (called probe functions) that represent object features and let $\phi_i \in \mathcal{B}, \phi_i :\longrightarrow \Re$. The *description* of an object x is a vector given by $\boldsymbol{\phi}_{\mathcal{B}}(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_k(x))$, where k is the length of $\boldsymbol{\phi}_{\mathcal{B}}(x)$ and $\phi_i(x)$ represents a feature value of x. Let $\varepsilon \in (0, +\infty)$. A tolerance relation $\cong_{\mathcal{B},\epsilon}$ is defined by

$$\cong_{\mathcal{B},\varepsilon} = \{(x,y) \in X \times X : \parallel \boldsymbol{\phi}_{\mathcal{B}}(x) - \boldsymbol{\phi}_{\mathcal{B}}(y) \parallel_2 \leq \varepsilon\},$$

where $\parallel \cdot \parallel_2$ is the L_2 norm (*i.e.*, Euclidean distance). Every tolerance relation determines two useful sets, namely, neighbourhood and tolerance class. The *neighbourhood of a point* $x \in X$ with respect to a tolerance $\cong_{\mathcal{B},\varepsilon}$ is a set $N_{\cong_{\mathcal{B},\varepsilon}}(x) = \{y \in X : \; y \cong_{\mathcal{B},\varepsilon} x\}$. Observe that $(x, y) \in \cong_{\mathcal{B},\varepsilon}$ for objects $y \in N(x)$. By contrast, for a tolerance class $A \subset \cong_{\mathcal{B},\varepsilon}$, for every $x, y \in A$, we have $(x, y) \in \cong_{\mathcal{B},\varepsilon}$. Let $\mathrm{H}^{\varepsilon}_{\mathcal{B}}(X)$ denote the family of all tolerance classes of relation $\cong_{\mathcal{B},\varepsilon}$ on the set X.

Near sets are disjoint sets that resemble each other, and tolerance near sets are near sets defined using the tolerance relation. Resemblance is determined by measuring the distance between object descriptions. For instance, a tolerance class X resembles (is near) a tolerance class Y if, and only if there are $x \in X$ and $y \in Y$ such that $x \cong_{\mathcal{B},\varepsilon} y$. If this is the case, tolerance classes X and Y are considered tolerance near sets.

TOLERANCE CLASS CALCULATION

The practical application of the tolerance space approach to measuring resemblance between digital images rests on our ability to find tolerance classes efficiently. An algorithm useful in finding tolerance classes in a tolerance relation $\cong_{\mathcal{B},\varepsilon}$ stems from Proposition 1.

Proposition 1. *Given a tolerance space $\langle X, \cong_{\mathcal{B},\varepsilon} \rangle$, all tolerance classes containing $x \in X$ are subsets of neighbourhood $N(x)$.*

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

Algorithm:

1. Take an element $z \in Z$ and find $N_{\cong_{\mathcal{B},\varepsilon}}(z)$.

2. Add z to a new tolerance class C. Select an object $z' \in N_{\cong_{\mathcal{B},\varepsilon}}(z)$ such that $z' \neq z$.

3. Add z' to C. Find neighbourhood $N_{\cong_{\mathcal{B},\varepsilon}}(z')$ using only objects from $N_{\cong_{\mathcal{B},\varepsilon}}(z)$. Do not include z in $N_{\cong_{\mathcal{B},\varepsilon}}(z')$. Select a new object $z'' \in N_{\cong_{\mathcal{B},\varepsilon}}(z')$ such that $z'' \neq z'$. Relabel $z \leftarrow z'$, $z' \leftarrow z''$ and $N_{\cong_{\mathcal{B},\varepsilon}}(z) \leftarrow N_{\cong_{\mathcal{B},\varepsilon}}(z')$.

4. Repeat step 3 until a neighbourhood of only 1 element is produced. When this occurs, add the last element to C, and then add C to $\mathrm{H}^{\varepsilon}_{\mathcal{B}}(Z)$.

5. Perform step 2 (and subsequent steps) until each object in $N_{\cong_{\mathcal{B},\varepsilon}}(z)$ has been selected at the level of step 2.

6. Perform step 1 (and subsequent steps) for each object in Z.

7. Delete any duplicate classes.

Finally, note the following. We used an added heuristic for step 2 to reduce the computation time of the algorithm. Namely, an object from $N_{\cong_{\mathcal{B},\varepsilon}}(z)$ can only be selected as z' in step 2 if it has not already been added to a tolerance class created from $N_{\cong_{\mathcal{B},\varepsilon}}(z)$ (*i.e.*, this rule is reset each time step 1 is visited). In addition, the Fast Library for Approximate Nearest Neighbours (Muja 2009) was used to find all the neighbourhoods in this algorithm.

A visual example of a sample run of this algorithm if given in Fig. 1 to help clarify the pseudo code given above. In this case, an example of a neighbourhood containing 21 objects in a 2D feature space is given, where the position of all the objects are labelled by the numbers 1 to 21, the neighbourhood is defined with respect to the object labelled 1, and $\varepsilon = 0.1$. Moreover, as per the definition of a neighbourhood, the distance between all the objects and object 1 is less than or equal to $\varepsilon = 0.1$, but that not all pairs of objects in the neighbourhood of x satisfy the tolerance relation. Also, notice that in each figure, the area shaded grey represents objects that satisfy the tolerance relation with the bold object(s), and the bold object(s) represent a pre-class.

To begin with, Fig. 1a represents Step 1 of the algorithm with $z = 1$. Step 2 is given in Fig. 1b, where $z' = 20$. Steps 3 & 4 are given in Fig. 1c-1f. Observe that in Fig. 1f $|N_{\cong_{\mathcal{B},\varepsilon}}(3)| = 1$ since all the other bold object in the grey area have been added to C, and, as such, are not allowed to be included in subsequent neighbourhoods. Step 5 can be explained as follows. Fig. 1 shows the sequence of steps for selecting $z = 20$ (the closest object to 1) at the level of Step 2. Hence, Step 5 states that each object in the neighbourhood of 1 (except 1 itself) should be selected at Step 2. Moreover, the heuristic given after the algorithm states that any object added to a tolerance class derived from the neighbourhood of 1 should not be considered at Step 2. As a result, in this example, the objects $\{3, 6, 10, 15, 16\}$ should not be considered again at Step 2 for finding tolerance classes derived from the neighbourhood of object 1. Lastly, note that Step 1 must be performed for all objects in Z.

NEARNESS METRICS

Applying near set theory to the problem of image resemblance requires a method for determining the degree in which two tolerance near sets are similar. Let X and Y be disjoint sets and let $Z = X \cup Y$. Then a tolerance nearness metric (tNM) (Henry 2010a) is given by Eq. 1.

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

The idea behind Eq. 1 is that similar sets should have tolerance classes that are evenly divided between X and Y. This is measured by counting the number of objects that belong to sets X and Y for each tolerance class $C \in \mathrm{H}^{\varepsilon}_{\mathcal{B}}(Z)$, and then compare these counts as a proper fraction. The measure is simply a weighted average of all of the fractions.

In Eq. 2, we introduce a second similarity measure for comparison with Eq. 1 defined as

$$tHM_{\cong_{\mathcal{B}}}(X,Y) = 1 - \frac{1}{|H_{\cong_{\mathcal{B}}}(Z)|} \cdot \sum_{C \in H_{\cong_{\mathcal{B}}}(Z)} \mathbf{1}(|\text{avgn}(C \cap X) - \text{avgn}(Y \cap X)| \leq \varepsilon), \tag{2}$$

where $\mathbf{1}(\cdot)$ is the indicator function and avgn$(C \cap X)$ is the average feature vector used to describe objects in $C \cap X$. Here, the idea is that, for similar sets, the average feature vector of the portion of a tolerance class (obtained from $Z = X \cup Y$) that lies in X should have values similar to the average feature vector of the portion of the tolerance class that lies in Y. The tHM measure was

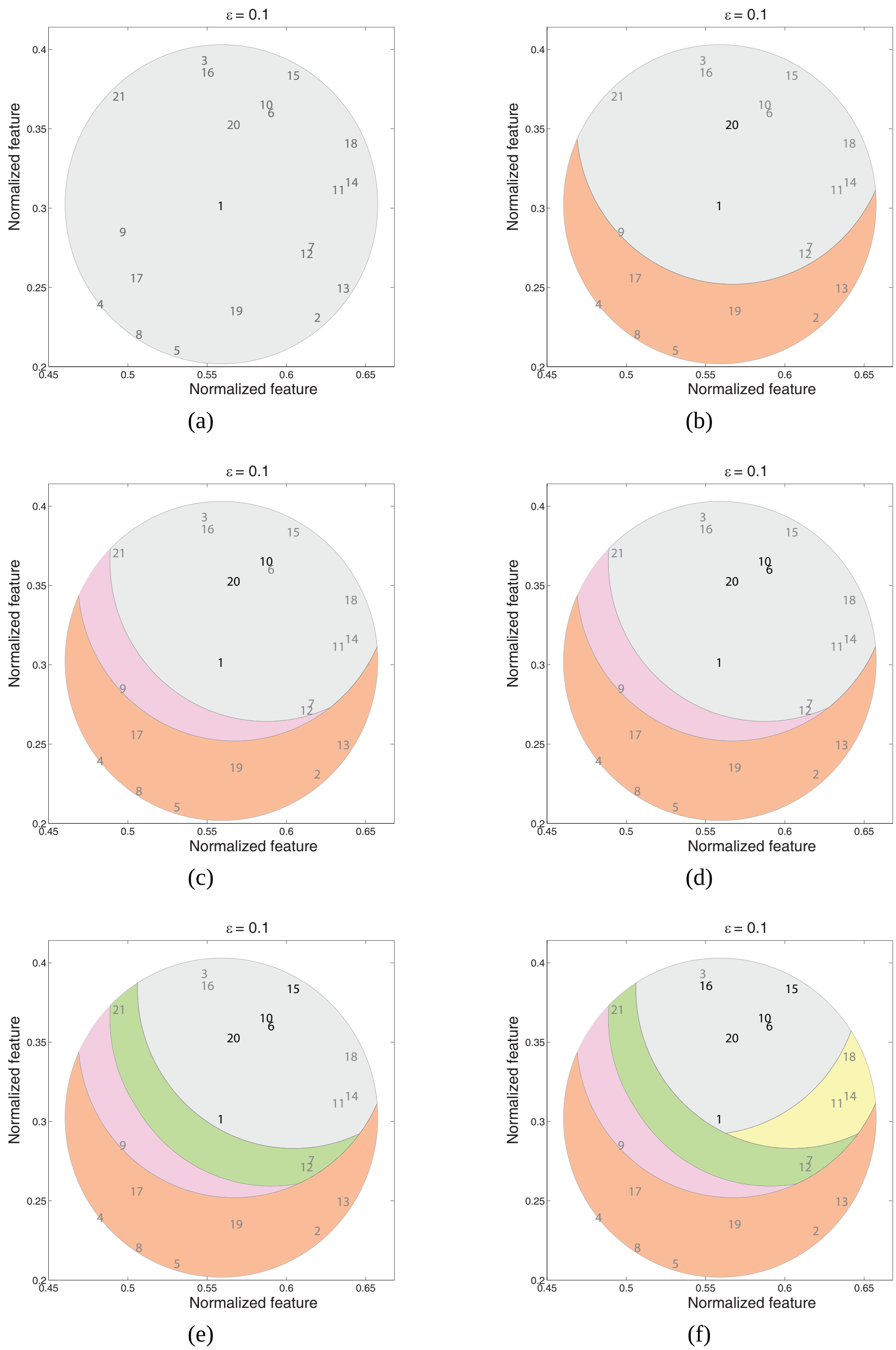

FIGURE 1 Visualization of Algorithm. (a) $N(1)$, (b) $N(20)$, created using only objects from $N(1)$, (c) $N(10)$, created using only objects from $N(20)$ (which was created using only objects from $N(10)$), (d) $N(6)$, again created using only objects from $N(10)$, *etc.*, (e) $N(15)$, and (f) $N(16)$.

inspired by the Hamming measure in (Ferrer et al. 2009), and since the Hamming measure is not defined in terms of sets, we have not included it in our resemblance measurement experiments.

VISION SYSTEMS APPLICAITONS

As was mentioned, one of the goals of this paper is to demonstrate that the proposed approach would work well in a vision system. Currently, our approach only measures the similarity of images. However, this approach could be combined with a real time image acquisition system to produce a vision system. For instance, the ALiCE II system we reported in (Peters et al. 2006) (and shown in Fig. 2) is an example of a system that would benefit from the approach presented in this article. ALiCE II is an autonomous line-crawling robot designed to inspect hydro-electric power transmission equipment. Currently, the system includes simple target tracking based on correlation between input frames and template images of equipment ment for inspection. This system could be improved using the approach for measuring similarities of images presented in this paper to give a full vision system capable of identifying damaged hydro-electric equipment. Here the idea is that the robot would contain a database of images representing equipment which needed to be inspected, and only images identified as similar would be further inspected by ALiCE II.

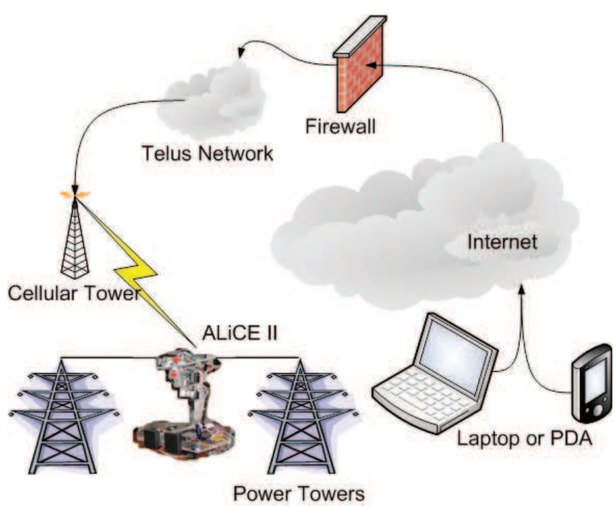

FIGURE 2 Figure demonstrating application of proposed image resemblance method in the ALiCE II vision system (Peters et al. 2006).

EXPERIMENTAL RESULTS

This section presents a practical application of the tolerance near set approach for measuring the similarity of images. Specifically, Eq. 1 & 2 are used to perform image retrieval on a database of 98 hand-finger images. This collection of images includes normal as well as rheumatoid arthritis patient hand-finger movements. The images were extracted from video sequences obtained from a telerehabilitation system that monitors patient hand-finger motion during rehabilitation exercises (see, *e.g.*, (Szturm et al. 2008)). It is important to have some way to compare nuances in normal versus arthritic hand-finger movements. Hence, the interest in measuring the similarities between the hand images. Before performing the experiments, the images were preprocessed so that only the hand is contained in the image (see, *e.g.* Fig. 3). The approach was to consider each image as a query image and to use precision/recall plots to determine the effectiveness of the nearness measures. Implementation of the measures is accomplished in the following manner. Define a RGB image as $f = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_T\}$, where $\mathbf{p}_i = (c, r, R, G, B)^{\mathrm{T}}$, $c \in [1, M]$, $r \in [1, N]$, $R, G, B \in [0, 255]$, and M, N respectively denote the width and height of the image and $M \times N = T$. Further, define a square subimage as $f_i \subset f$ such that $f_1 \cap f_2 \ldots \cap f_s = \emptyset$, and $f_1 \cup f_2 \ldots \cup f_s = f$, where s is the number of subimages in f. Next, label the query image and the current image for comparison as X and Y respectively, and view each image as a set of subimages. In this experimentonly one probe function was used, namely the average orientation of lines (obtained using Mallat's multiscale edge detection algorithm (Mallat and Zhong 1992)) within a subimage. Finally, the precision/recall plot comparing the two measures is given in Fig. 4. It can be observed that tNM measure has the best precision versus recall for image queries from the patient database.

CONCLUSION

This article has presented a new algorithm for finding tolerance classes based on the insight provided by Proposition 1. Moreover, a new measure of similarity was introduced for comparison with the tNM, and a practical application of the nearness measures was demonstrated by way of correspondence experiments on images obtained during rehabilitation exercises for arthritic patients.

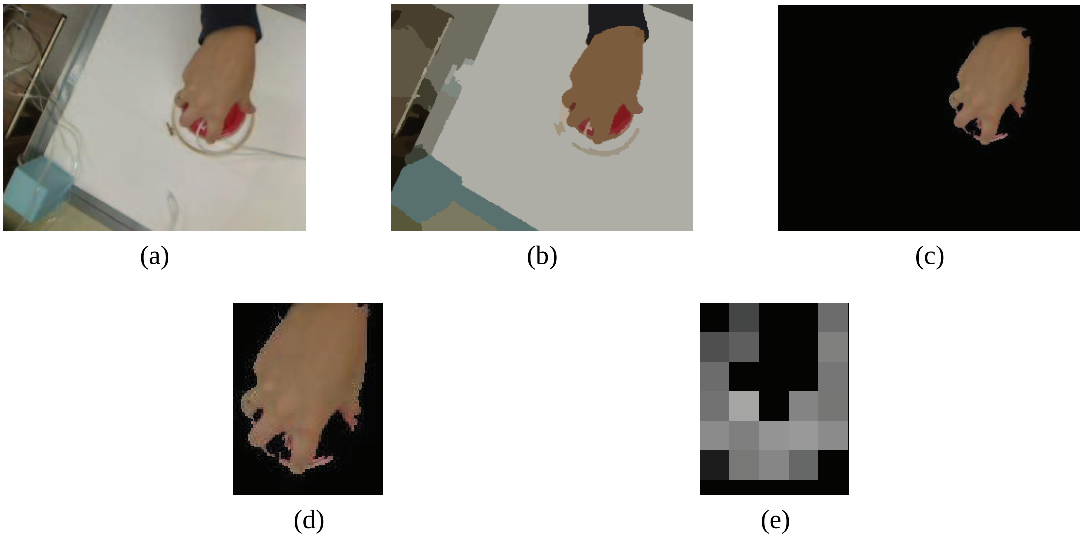

FIGURE 3 Figure showing preprocessing required to create tolerance classes and calculate nearness measure. (a) Original image, (b) segmented image, (c) hand segment only, (d) cropped image to eliminate useless background, and (e) final image used to obtain tolerance classes. Each square represents an object where the colour (except black) represents the average orientation of a line segment within that subimage.

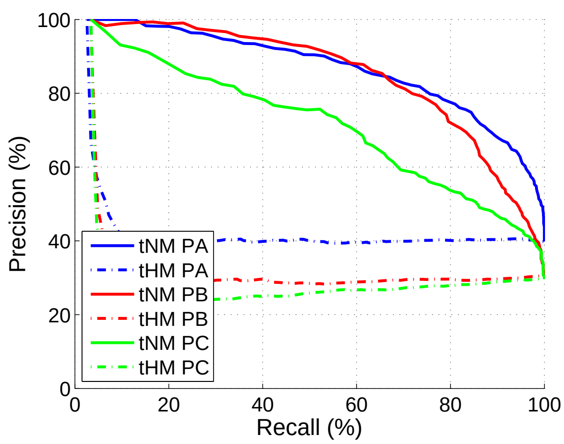

FIGURE 4 Precision/recall plot comparing tNM with tHM. PA, PB, and PC refer to three patients and PB is the patient with arthritis. Note, an ideal plot would show a precision of 100% until recall reached 100% showing that only images from the same category as the query image were returned first

While the new measure did not perform as well as the tNM measure, the results presented here provide an important first step to developing a new metric for similarity evaluation in vision systems. Finally, future work will consist of incorporating the tolerance near set approach presented here in a real-time vision system.

ACKNOWLEDGEMENTS

This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research grant 185986, Manitoba Centre of Excellence Fund (MCEF) grant, Canadian Network Centre of Excellence (NCE) and Canadian Arthritis Network (CAN) grant SRI-BIO-05.

REFERENCES

Bakhtari, A., and B. Benhabib. 2007. An active vision system of multitarget surveillance in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 37(1):190–198.

Ferrer, M. A., A. Morales, and L. Ortega. 2009. Infrared hand dorsum images for identification. *IET Electronic Letters* 45(6):306–308.

Hassanien, A. E., A. Abraham, J. F. Peters, G. Schaefer, and C. Henry. 2009. Rough sets and near sets in medical imaging: A review. *IEEE Transactions on Information Technology in Biomedicine* 3(6):955–968.

Henry, C. 2010a. Near set Evaluation And Recognition (NEAR) system. In *Rough fuzzy analysis foundations and applications*, ed. S. K. Pal and J. F. Peters, *accepted*. CRC Press, Taylor & Francis Group. ISBN 13: 9781439803295.

———. 2010b. Near Sets: Theory and Applications. Ph.d. dissertation, University of Manitoba, CAN.

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

———. 2010. Perception-based image classification. *International Journal of Intelligent Computing and Cybernetics accepted*.

Hussmann, S., and T. Liepert. 2009. Three-dimensional tof robot vision system. *IEEE Transactions on Instrumentation and Measurement* 58(1):141–146.

Mallat, S., and S. Zhong. 1992. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(7):710–732.

Meghdadi, A. H., J. F. Peters, and S. Ramanna. 2009. Tolerance classes in measuring image resemblance. In *Knowledge-based and intelligent information and engineering systems*, vol. 5712, 127–134. Santiago, Chile.

Mojsilović, A., and B. E. Rogowitz. 2004. Semantic metric for image library exploration. *IEEE Transactions on Multimedia* 6(6):828–838.

Muja, M. 2009. FLANN - Fast Library for Approximate Nearest Neighbors. Http://www.cs.ubc.ca/ mariusm/index.php/FLANN/FLANN.

Peters, J. F. 2009a. Discovering affinities between perceptual granules. l_2 norm-based tolerance near preclass approach. *Advances in Man-Machine Interactions and Soft Computing* 59:43–54.

———. 2009b. Fuzzy sets, near sets, and rough sets for your computational intelligence toolbox. In *Foundations of compuational intelligence*, ed. A. Abraham and F. Herrera, vol. 2. Springer, Heidelberg.

———. 2009c. Tolerance near sets and image correspondence. *International Journal of Bio-Inspired Computation* 1(4):239–245.

———. 2010. Corrigenda and addenda: Tolerance near sets and image correspondence. *International Journal of Bio-Inspired Computation* 2(5):*in press*.

Peters, J. F., M. Borkowski, C. Henry, D. Lockery, and D. S. Gunderson. 2006. Line-crawling bots that inspect electric power transmission line equipment. In *Proceedings of the third international conference on autonomous robots and agents (icara 2006)*, 39–45. Palmerston North, New Zealand.

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

Peters, J. F., L. Puzio, and T. Szturm. 2009. Measuring nearness of rehabilitation hand images with finely-tuned anisotropic wavelets. In *Image processing & communication challenges*, ed. R. S. Choraś and A. Zabludowski, 342–349. Warsaw: Academy Publishing House.

Ramanna, S., and A. H. Meghdadi. 2009. Measuring resemblances between swarm behaviours: A perceptual tolerance near set approach. *Fundamenta Informaticae* 95:533–552.

Szturm, T., J. F. Peters, C. Otto, N. Kapadia, and A. Desai. 2008. Task-specific rehabilitation of finger-hand function using interactive computer gaming. *Archives of Physical Medicine and Rehabilitation* 89(11): 2213–2217.

Vasconcelos, N., and A. Lippman. 2005. A multiresolution manifold distance for invariant image similarity. *IEEE Transactions on Multimedia* 7(1):127–142.

Wagner, M. 2006. *The geometries of visual space*. Mahwah, New Jersey, USA: Lawrence Erlbaum Associates, Inc.

Yang, L., R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. H. Hoi, and M. Satyanarayanan.

2010. A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1):30–44.

Zeeman, E. C. 1965. The topology of the brain and the visual perception. In *Topoloy of 3-manifolds and selected topices*, ed. K. M. Fort, 240–256. New Jersey: Prentice Hall.

Zeeman, E. C., and O. P. Buneman. 1968. Tolerance spaces and the brain. In *Towards a theoretical biology*, ed. C. H. Waddingto, 140–151. London: Aldine. Published in C.H. Waddington (Ed.), Towards a Theoretical Biology. The Origin of Life, Aldine Pub. Co.

Zhang, X., and A. L. P. Tay. 2009. A physical system for binocular vision through saccade generation and vergence control. *Cybernetics and Systems* 40:549–568.

Zheng, L., A. W. Wetzel, J. Gilbertson, and M. J. Becich. 2003. Design and analysis of a content-based pathology image retrieval system. *IEEE TRansactions on Information Technology in Biomedicine* 7(4): 249–255.