

PERCEPTUAL SYSTEMS APPROACH TO MEASURING IMAGE RESEMBLANCE

Irina Bocearov¹
Mihai Zaharescu²

ABSTRACT

This projects implements a particular approach in measuring image resemblance called Perceptual Systems Approach, based on Near Sets Theory where image resemblance is viewed as a form of nearness between sets of perceptual objects. This application offers a simple interface that is easy to use and understand. The user only needs to upload two images and in a few seconds he will have the result he was interested in.

KEYWORDS: Perceptual resemblance, image resemblance, resemblance measure

1. INTRODUCTION

The best way to motivate the necessity on this application is to show the actuality and completeness of the used approach.

One solution to the problem of image resemblance is to consider a tolerance space form of near sets that model human perception in a physical continuum. Therefore, a perceptual image processing approach has been rapidly growing out of the near set theory for the past several years. Near sets are disjoint sets that resemble each other, especially resemblance defined within perceptual representative spaces (a.k.a., tolerance spaces). This way, this approach takes care, first of all, of the human perception of the world.

This paper is organized as follows. Section 2 presents related works to help establish a context for this research. Section 3 presents the working of the implementation. Section 4 presents the conclusions and some future work is presented in Section 5.

2. RELATED WORK

Near sets, introduced in 1980s by Zdzislaw Pawlak are useful for solving problems based on human perception [1] that arise in areas such as image analysis [2,3], image processing [2,4], face recognition [5], ethology [6], as well as engineering and science problems

As an illustration of the degree of nearness between two sets, consider an example of the Henry color model for varying degrees of nearness between sets. The two pairs of ovals in Figures 1 and 2 contain colored segments. Each segment in the figures corresponds to an equivalence class where all pixels in the class have matching descriptions, i.e., pixels with matching colors. Thus, the ovals in Figure 1 are closer to each other in terms of their descriptions than the ovals in Figure 2.

¹ Engineer, irina.bocearov@cti.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

² Engineer, mihai.zaharescu@cs.pub.ro, "Politehnica" University of Bucharest, 060042 Bucharest, Romania

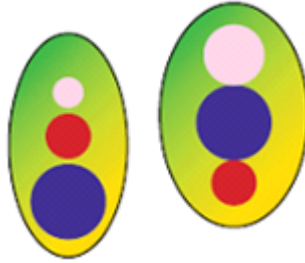


Figure 1. Very Near Sets

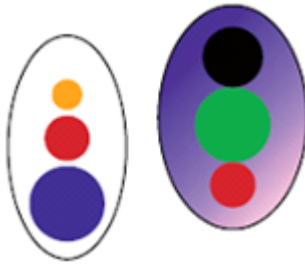


Figure 2. Minimally Near Sets

When investigating the perceptual basis Orłowska suggested that approximation spaces are the formal counterpart of perception or observation [8]. In [9] it is explained that the brain identifies relationships between object features in the process of forming the perceptions.

The idea of using tolerance relations [10] in formalizing the concept of perceptual resemblance between images was introduced in 2008 [11] as a result of collaboration with Z. Pawlak in 2002 on describing the nearness of perceived objects [12].

This project implements the theory described in [7].

3. THE APPLICATION

Each step described in [7] provides input for the future steps, producing tolerance classes of the images and corresponding tolerance matrix used in the next steps to define similarity measures. As a probe function, this project used the average gray value of the sub-images.

The following sections present some results by emphasizing the difference between pair of images with a good resemblance and images which are different from a perceptual point of view. Tests have been made for different sizes of grayscale [14][15] images, sub-images, tolerance range ε to compare results and to analyze which factors are most influent.

Images with a good resemblance

For two input images of size 300 x 300 px, the application generate the tolerance ordered matrix which is displayed after the black and white version of the image(in case the image is colored). From here on, the following notation will be used: X for the first image and Y for the second one.



Figure 3. Image X and corresponding tolerance ordered matrix



Figure 4. Image Y and corresponding tolerance ordered matrix

It is the moment to compare these two images from a personal, perceptual point of view and to see if the application meets our perception sense. If they match, the application must have a good implementation and provide the right results. As we can see these two images are similar, let's check the results.

For a visual interpretation of the results, the application provides a graphical representation of the empirical distribution function, but what the user must see is if the two lines corresponding to each image are similar or not. As our two images are supposed to be similar the resulting histogram is:

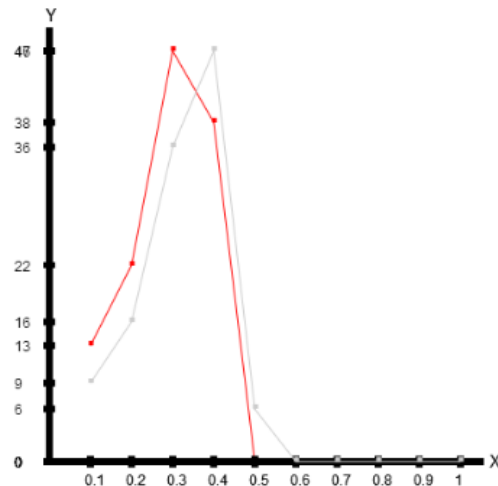


Figure 5. Empirical histograms/ Distribution for image X and Y

It is not necessary to know much about tolerance classes or empirical distribution to understand that these two images have a good degree of similarity. The numerical values organized in a Table 1.

Results

p	eps	#Tol-X	#Tol-Y	TOD	HSM	tNM
20	20	218	209	0.9800	0.9800	0.9305

Table 1. able of results

TOD, HSM and tNM were explained in the previous section and have a range of 0 to 1 with a higher value for a better resemblance. p means the size of the sub-images in pixels (20 x 20), eps represents the tolerance range and #Tol-X and #Tol-Y represents number of tolerance classes for image X and Y. Again results shows that images have a good resemblance. For comparison, section 4.2 presents the results for two different images.

Images that do not resemble

As for the previous example, after loading the two images their black and white versions are displayed, as well as the ordered tolerance matrix.



Figure 6. Image X and corresponding tolerance ordered matrix

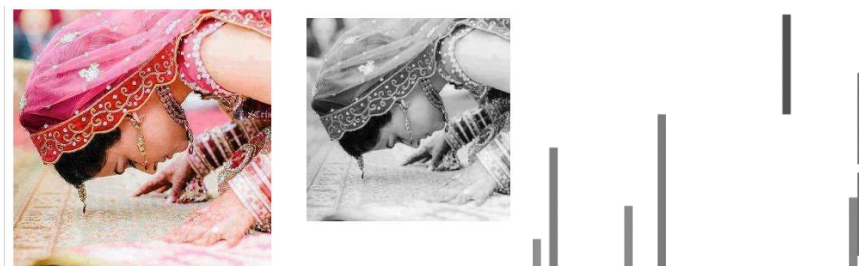


Figure 7. Image Y and corresponding tolerance ordered matrix
The resulting histogram is:

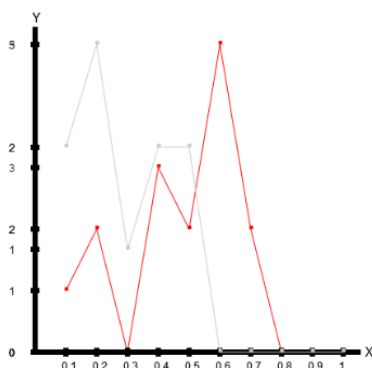


Figure 8. Empirical histograms/ Distribution for image X and Y
As we can see in resulted histogram those two images are not similar from a perceptual point of view. Table 2 presents the numerical results:

Results						
p	eps	#Tol-X	#Tol-Y	TOD	HSM	tNM
100	20	192	185	0.9000	0.9000	0.7326

Table 2. Table of results

We can observe smaller values for tNM and TOD, HSM indicators. To highlight the algorithm performance, next will be presented an example which compares two absolutely different images and an example with two equal images.

Images with a very poor similarity



Figure 9. Image X and corresponding tolerance ordered matrix



Figure 10. Image Y and corresponding tolerance ordered matrix

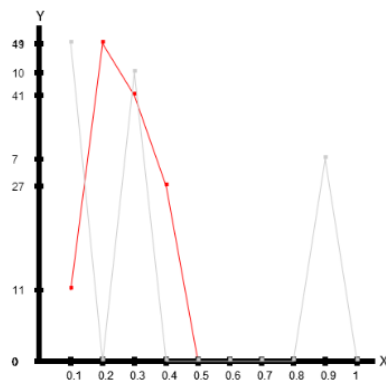


Figure 11. Empirical histograms/ Distribution for image X and Y

Results						
p	eps	#Tol-X	#Tol-Y	TOD	HSM	tNM
20	20	239	256	0.4914	0.4914	0.2003

Table 3. Table of results

Two equal images



Figure 12. Image X and Y and corresponding tolerance ordered matrix

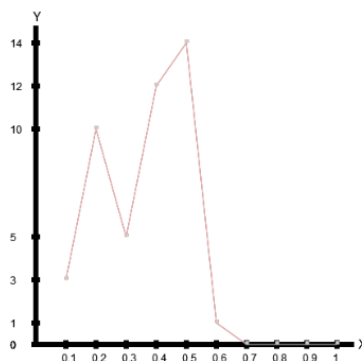


Figure 13. Empirical histograms/ Distribution for image X and Y

As it is shown in Figure 13 the two lines overlaps, it was expected because the images are identical.

Results						
p	eps	#Tol-X	#Tol-Y	TOD	HSM	tNM
20	20	256	256	-Infinity	-Infinity	1.000

Table 4. Table of results

The values of indicators also show that our images are identical. Also the pictures have the same number of tolerance classes.

Next, for more tests of the application a number of 5 different pictures where inserted into a database and the input images are now extracted from a selected database. One of the images will be modified so that it can be another input image resulting 6 images where 2 of them are almost the same images. The fifth image is obtained by drawing a line over the pair image. The idea is to test if the application will find those two images almost the same. The user selects the image and for each of another 4 images will be applied algorithms to compute the values of the indicators and the best result will meets the pictures we want to find (the duplicate). The pictures will be named as 1, 1' (duplicate image), 2, 3, 4. Next figures show the results:



Figure 14. Original image and the modified duplicate. (1 and 1')

The results of the comparison of the duplicate image with other 4 images are presented next. The minimum values of indicators will indicate the image we need to find.

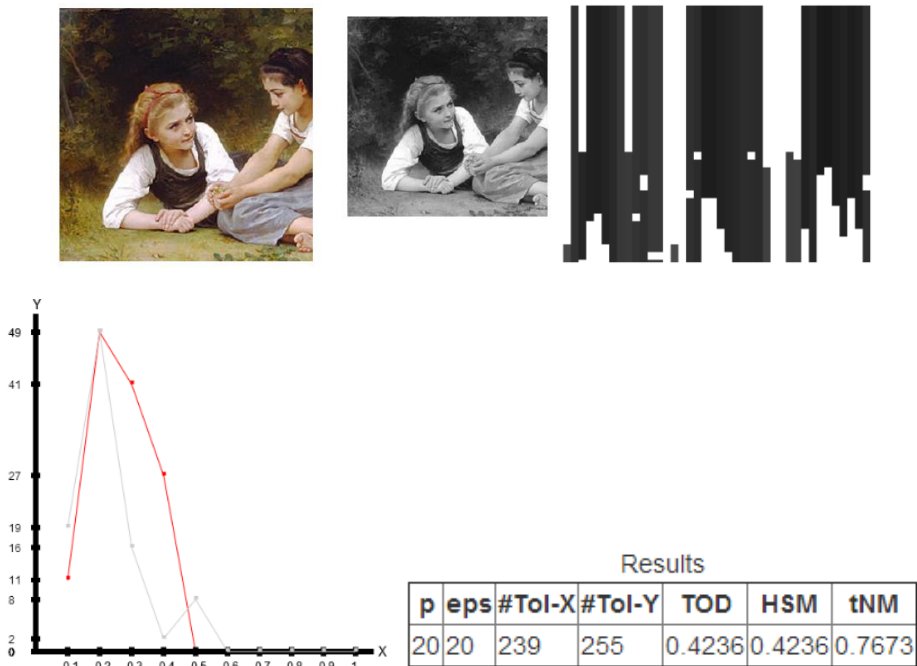


Figure 15. Results of comparison of image 1 and 2.

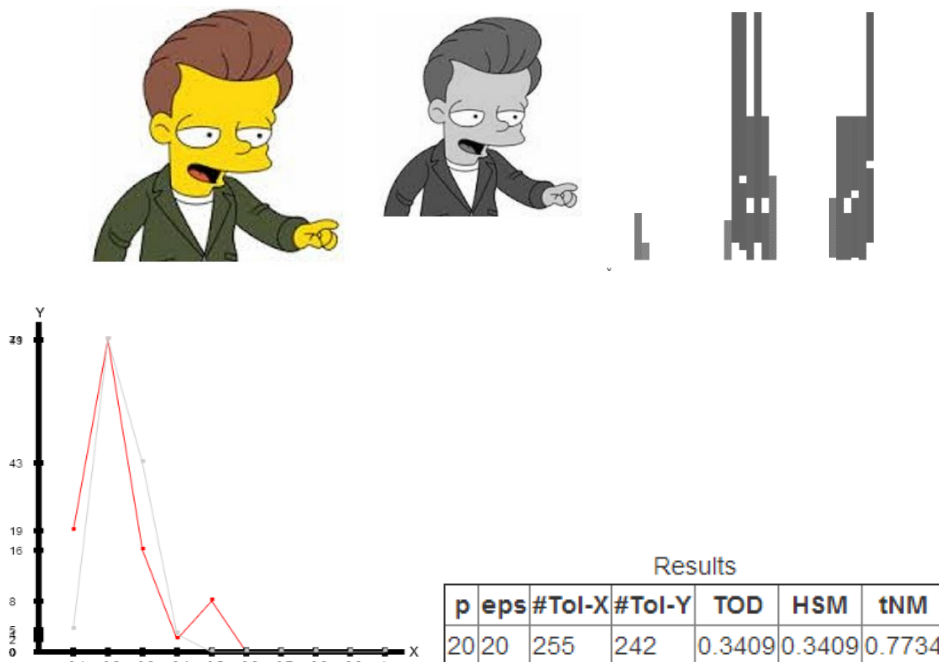


Figure 16. Results of comparison of image 1 and 3.

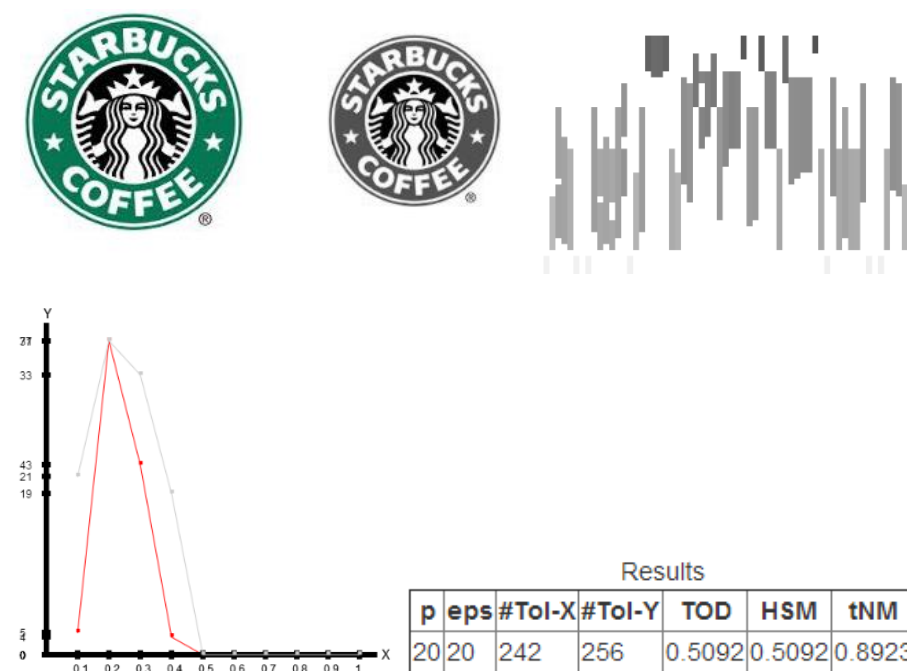


Figure 17. Results of comparison of image 1 and 3.

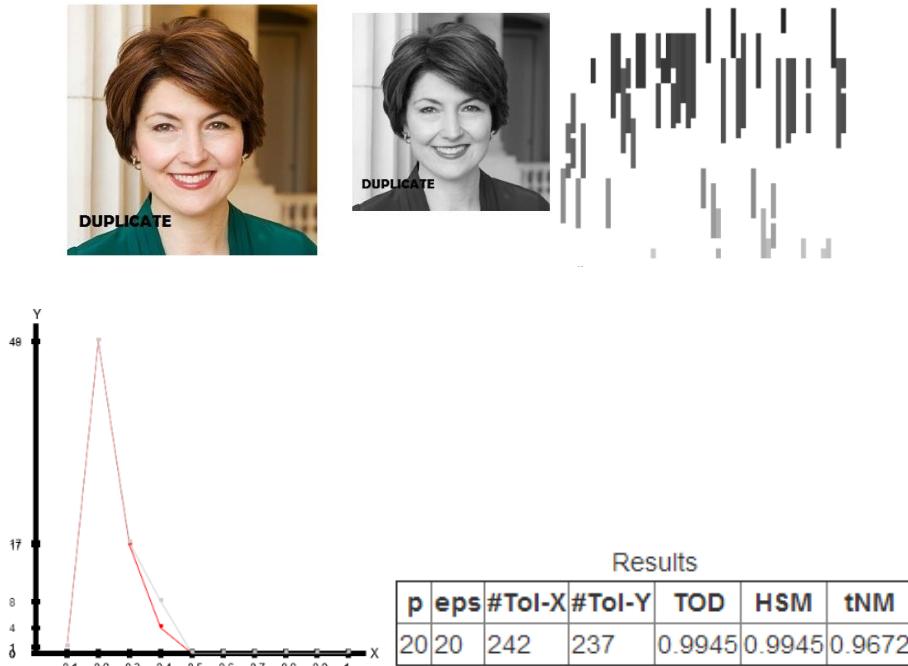


Figure 18. Results of comparison of image 1 and 1'.

By collecting all the resulted data and making the minimum of the value of the indicators application easily find the duplicated image. Even on the distribution diagram it's easy to see that image are the same but at one point there is a difference, in our case the text "DUPLICATE".

4. CONCLUSIONS

The main goal of this project was to provide an application with a simple, easy to use interface which offers in a few seconds some visible results, easy to understand and to gather the information about the image resemblance, based upon the article [7] for perceptual similarity.

Three forms of image resemblance measures are considered, the tolerance overlap distribution (TOD), HSM histogram similarity measure and a tolerance near set resemblance measure tNM. Because of the use of only one probe function (average gray level) values of the TOD and HSM are the same, so it can be considered two forms of measure. The results reported in this paper suggest that the tNM measure is more accurate than the TOD and HSM. The empirical distribution histogram gives the best visual results.

5. FUTURE WORK.

To increase the precision of the results, for the future work application could be improved by adding some more probe functions like image entropy, texture and color information.

Another improvement would be to eliminate the requirements concerning the input image size, the possibility to upload images of different sizes. Also, the algorithm could be optimized to decrease the time of the execution for larger images.

This system can work in conjunction with image merge and stitching algorithms in order to classify images containing the same scene. [13][16][17]

6. REFERENCES

- [1] Peters JF, Wasilewski P. Foundations of near sets. *Information Sciences*. 2009; 179(18):3091–3109
- [2] Review Rough sets and near sets in medical imaging: a review. Hassanien AE, Abraham A, Peters JF, Schaefer G, Henry C *IEEE Trans Inf Technol Biomed*. 2009 Nov; 13(6):955–68.
- [3] Peters JF, Puzio L, Szturm T. Measuring nearness of rehabilitation hand images with finely tuned anisotropic wavelets. In: Choraś RS, Zabłudowski A, editors. *Image Processing & Communication Challenges*. Warsaw, Poland: Academy Publishing House; 2009. pp. 342–349
- [4] Peters JF. Tolerance near sets and image correspondence. *International Journal of Bio-Inspired Computation*. 2009; 1:239–245.
- [5] Gupta S, Patnaik K. Enhancing performance of face recognition systems by using near set approach for selecting facial features. *Journal of Theoretical and Applied Information Technology*. 2008; 4:433–441.
- [6] Ramanna S, Meghdadi AH. Measuring resemblances between swarm behaviours: a perceptual tolerance near set approach. *Fundamenta Informaticae*. 2009; 95(4):533–552.
- [7] Amir H. Meghdadi, James F. Peters. Perceptual Systems Approach to Measuring Image Resemblance
- [8] Orlowska, Ewa, and Zdzislaw Pawlak. 1984. Representation of nondeterministic information. *Theoretical computer science* 29(1-2):27–39.
- [9] Peters, J.F. Notes on tolerance relations. Computational Intelligence Laboratory Seminar. See J.F. Peters, “Tolerance near sets and image correspondence”, *Int. J. of Bio-Inspired Computing* 4 (1), 2009, 239–245
- [10] Sossinsky, A. B. 1986. Tolerance space theory and some applications. *Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications* 5(2):137–167.
- [11] Peters, J.F. 2008. Notes on perception. Computational Intelligence Laboratory Seminar.
- [12] Pawlak, Zdzislaw, and James Peters. 2002, 2007. *Jak blisko (how near)*. Systemy Wspomagania Decyzji I:57, 109. ISBN 83-920730-4-5.
- [13] Costin-Anton Boiangiu, Ion Bucur, Andrei Cristian Spataru, “Statistical Approaches Used in Automatic Merge of Scanned Images”. *Annals of DAAAM for 2008, Proceedings of the 19th International DAAAM Symposium*, Editor B. Katalinic, Published by DAAAM International (Vienna, Austria), pp. 0129–0130, Trnava, Slovakia, October 22–25, 2008
- [14] Andrei Tigora, Costin-Anton Boiangiu, “Image recoloring using iterative refinement,” in *Proceedings of the 1st International Conference on Image Processing and Pattern Recognition (IPPR '13)*, Budapest, Hungary, December 10–12, 2013, pp. 204–209.
- [15] Andrei Tigora, Costin-Anton Boiangiu, “Image Color Reduction Using Iterative Refinement”, *International Journal of Mathematical Models and Methods in Applied Sciences*, Volume 8, 2014, pp. 203–207.
- [16] Costin-Anton Boiangiu, Andrei Cristian Spataru, Andrei Iulian Dvornic, Ion Bucur, “Usual Scenarios and Suitable Approaches Used in Automatic Merge of Scanned Images”, *International Journal of Computers*, Issue 1, Volume 2, 2008, pp. 340–349.
- [17] Costin-Anton Boiangiu, Andrei Cristian Spataru, Andrei Iulian Dvornic, Ion Bucur. “Merge Techniques for Large Multiple-Pass Scanned Images”. *Proceedings of the 1st WSEAS Int. Conf. on Visualisation, Imaging and Simulation (VIS '08)*, WSEAS Press, pp. 72 – 76, Bucharest, Romania, November 7–9, 2008.