Spatial Keypoint Representation for Visual Object Retrieval

Tomasz Nowak[1], Patryk Najgebauer[1], Jakub Romanowski[1], Marcin Gabryel[1], Marcin Korytkowski[1], Rafał Scherer[1], and Dimce Kostadinov[2]

[1] Institute of Computational Intelligence, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
{tomasz.nowak,patryk.najgebauer,jakub.romanowski,marcin.gabryel,
rafal.scherer,marcin.korytkowski}@iisi.pcz.pl
http://iisi.pcz.pl

[2] University of Geneva, Computer Science Department,
7 Route de Drize, Geneva, Switzerland
http://sip.unige.ch

Abstract. This paper presents a concept of an object pre-classification method based on image keypoints generated by the SURF algorithm. For this purpose, the method uses keypoints histograms for image serialization and next histograms tree representation to speed-up the comparison process. Presented method generates histograms for each image based on localization of generated keypoints. Each histogram contains 72 values computed from keypoints that correspond to sectors that slice the entire image. Sectors divide image in radial direction form center points of objects that are the subject of classification. Generated histograms allow to store information of the object shape and also allow to compare shapes efficiently by determining the deviation between histograms. Moreover, a tree structure generated from a set of image histograms allows to further speed up process of image comparison. In this approach each histogram is added to a tree as a branch. The sub tree is created in a reverse order. The last element of the lowest level stores the entire histogram. Each next upper element is a simplified version of its child. This approach allows to group histograms by their parent node and reduce the number of node comparisons. In case of not matched element, its entire subtree is omitted. The final result is a set of similar images that could be processed by more complex methods.

Keywords: content-based image retrieval, keypoints, histograms.

1 Introduction

In this paper, we discuss the issue of semantically similar images recognition. Search of semantically similar images consists in finding images with related content such as aircrafts, dogs, cars etc. Phase of image processing is performed at the level of its pixels and there are not considered any other ways of image description (image labels, classes of image and others). Processed image is represented by various local features obtained on the basis of pixels, which may

L. Rutkowski et al. (Eds.): ICAISC 2014, Part II, LNAI 8468, pp. 651–662, 2014.

be colors, shapes or keypoints [30][32][34]. There are many methods of image processing, which are usually intended for specific areas of images processing e.g. face recognition [15], fingerprint [35], various symbols and specific objects. Often these methods are created for a specific purpose and usually they reach the goal much more faster and more accurately than humans can do. However, many contemporary methods ignore a very important feature of images which is spatial distribution of image features [4]. The development of appropriate model of the spatial representation of processed image objects constitutes a major challenge. It is especially difficult in the case for visual classification of any objects belonging to different classes. Images can be also classified by various soft computing techniques [1][3][5][6][7][8][16][17][19][20][21][22][23][24][25][28][29]. Novel approach for image retrieval based on nonparametric estimates (see e.g. [12] [26][27]) was presented in [9].

The main problem during image processing is a different perception of the same image by humans and computers. Humans focus on the information carried by the image, remember objects and their names, their relationships and the place in which they are located. Less attention is focused on details in image, trying to simplify and remember generalized information. In many cases humans are unable to reproduce precisely the remembered image, while the computer analyzes the image at a much lower level. Computers do not know what is located in an image, they remember just a group of pixels describing specific image. Pixels are perceived by computers as e.g. three color components pixels. By proper processing this kind of data sets, it is possible to classify processed images.

Most of general purpose algorithms are used to determine automatically interest points also called keypoints. The keypoints are identifiers of image areas which are distinctive from the rest of the image. By applying keypoints it is possible to skip less important parts of the image and focus only on specific areas. However keypoints do not allow search for images which are similar to each other. Important applications of keypoints are finding identical pieces of images, tracking of selected objects in video sequence or to create so-called maps of points that describe local image gradient. Methods based on keypoints often generate a large amount of points that contains only partial information about the image. Their significant and constant amount relative to each image is a problem of searching the common parts of images in their larger group. The disorder is fact that some points from one image does not correspondent to other one which is similar in theory. The search for images related to each other (e.g. two similar objects) with use only keypoints will not work correctly. Not significant change of view point causes that keypoints set will be different than in original image and in result adopting a rigid relationship between points in both images make that will be searched only parts of identical images or relation will be not found. To seek thematically similar objects there is a need to create a specific data structure which will allow to generalize the description of keypoints. In this paper we present a method of generalized keypoints description using histograms. This method of keypoints representation allows to compare similar objects contained in different images, as will be shown in experimental results. In addition, we will present image dictionary built on the basis of histograms, which allows to quick search large image data sets.

The paper is structured as follows. Section 2 shows an overview of popular existing methods of image comparison. Section 3 presents detailed discussion of the problem. Section 4 shows the new method of comparing objects in images. Section 5 shows the results of experiments carried out using the proposed method. Section 6 presents conclusions.

2 Previous Work

Histograms in the processing digital images have a long history. With their help, we can describe many dependencies contained in the image. One of the primary uses of histograms in the process of image processing is to represent the distribution of colors. There are many algorithms based on the analysis of color histograms of the image [11][14][31]. Histograms in the process of image processing are most commonly used to assign the number of pixels in the image to corresponding levels of brightness or color [13]. Histograms are widely considered to be very efficient and concise instrument to provide visual content of a digital image. However, the use of histograms only for the color distribution does not present the image in an unambiguous manner. Many different images may have the same histograms. In this paper we present a method to apply histograms to represent keypoints generated using the SURF algorithm [2]. This approach of keypoints representation will allow to describe their spatial distribution, thus it will be possible to find different objects belonging to the same class. With use of this method we expand the application of the classical approach for comparing images with the keypoints.

To generate keypoints we use the SURF algorithm. The main features of each generated keypoint are: position, orientation, size and descriptor. Algorithms for computing keypoints usually apply a mask defining the local extremes of the image. Such operation is typical for blob detection algorithms (Fig. 1) [33]. The image is processed many times and in each subsequent step the size of the mask is increased, creating a so-called pyramid. This allows to determine the keypoints regardless of their scale. The important advantage of the algorithm is significant acceleration relative to the previous algorithmic solutions. The increase in the efficiency is due to several improvements. In previously used algorithms there were used Gauss masks which required aggregation of all of the pixels repeatedly with a predetermined coefficient. The SURF algorithm is improved by using the so-called Integral Image (1) algorithm which allows rapidly calculating the sum of pixels in a selected area of image. The adopted simplified masks have a negative impact on the accuracy of the calculated descriptors. Integral image is a structure which is represented by the sum of pixels in any rectangular area of the input image I

$$I_{\sum}(x,y) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(x,y) \ , \tag{1}$$

where I - processed image, $I_{\sum}(x, y)$ - the sum of all pixels in the image. Calculation of the sum of the pixels in the selected area of the image (Integral Image) is described by (2)

$$\sum = A - B - C + D \,, \tag{2}$$

where A, B, C and D values are the coordinates of the vertices of the selected rectangular area in the image.

Finally, the algorithm generates descriptor and orientation for each of keypoints. Calculated orientation of the keypoint allows generating the same type of descriptors of keypoint regardless of the global orientation of the entire image. The descriptor (3) is a form of description of the local gradient over which keypoint is located

$$V_{sub} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right] . \tag{3}$$

Object recognition based on keypoints is usually carried out by matching keypoints from two processed images. Matching is carried out by finding the nearest neighbor in the data base of keypoints. The keypoints are typically used to search for exactly the same objects in another image, e.g. to object tracking in video sequences.

3 Problem Description

Keypoints are very useful in situations when the same objects are present in different images. They are resistant to some degree to change in rotation and scale of image. However, keypoints are not sufficient method in situations where there is a need to find objects of similar shape to each other, but not the exact copy of them (e.g. two different cars). This may be the not the keypoints limitation, but generally this is caused by the way in which they are compared. To find a correlation of a pair of points on the two images, all keypoints are compared point by point and the same rule applies to find similar descriptors of similar keypoints. Such an approach makes it impossible in practice to search for images different than processed one (or its fragments).

As it was mentioned above, often a situation occurs where for two the same objects generate different keypoints. To properly compare objects there should be kept an adequate tolerance between calculated descriptors. Increasing the tolerance results in the ability to create groups of similar descriptors. For example, Table 3 shows a matrix in size 4x4 where we present values of the deviations between two V_{sub} descriptors of two similar keypoints. To illustrate similarity of keypoints which descriptors were compared, we present them in Fig. 1. For humans these points seem to be identical while the standard deviation between these keypoints descriptors is 0.4743. For this reason, descriptors can be assumed to be identical when their standard deviation does not exceed 0.5. The simplest and first adopted method for the representation and indexing of keypoints is to generate stable hash codes based on descriptors. Such a solution would be

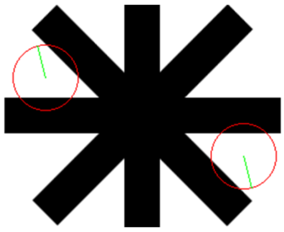

Fig. 1. SURF similar keypoints example with 0.47 value of different between descriptor components

Table 1. Deviation ($\Delta V_{sub}(x, y)$) between V_{sub} values of two similar keypoints

(a/b)	1	2	3	5
1	0.0078	0.0042	0.0619	0.0120
2	0.0293	0.0176	0.0282	0.0103
3	0.0210	0.0306	0.0473	0.0027
4	0.0003	0.0298	0.0181	0.0678

advisable due to the potential use of indexing used in data bases. Unfortunately, the nature of constructing descriptors and previously mentioned deviations between descriptors of keypoints cause many problems. To develop efficient and effective method of searching similar objects, there is a need to simplify description of keypoints enough, to be able to correctly classify processed objects. Frequently those encountered problems in finding similar keypoints stemmed from:

- Descriptor presence in the middle of two hash code values,
- Random deviations of the individual values of descriptors,
- Deviations resulting from changes in orientation, position or scale,
- Distortion or image noise in keypoints areas.

During the generation of the keypoints a set of points is created with various number of them. The number of keypoints depends mainly on the size of image and the amount of details contained in image. For example, the number of generated keypoints of the image in size of 1280x800 pixels in most cases exceeds 1000. During comparison process of two images, the large number of calculated keypoints is the reason of significant slow down.

The easiest way to compare keypoints from different images is all-to-all comparison, but with such a large number of points, this may require a significant time complexity of the algorithm. As an example we can consider two sets of keypoints with 1000 points in each of them. In this situation the algorithm needs to calculate one million of keypoint comparisons. To reduce the number of required comparisons of descriptors we can organize them by sorting. Properly sorted points allow for the omission of part of the descriptors during their comparison.

4 Proposed Method

Given a set of keypoints of the test object generated using the SURF algorithm, we set the new point which is the center of this object. The new point will be a center of the coordinate system, based on which we define the distribution of the remaining keypoints in space. Specifying the position of the new points $(\bar{x}, \bar{y})$ we use dependency

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \ \bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} , \tag{4}$$

where $[x_1, x_2, ..., x_n]$, $[y_1, y_2, ..., y_n]$ are coordinates of the points and $[w_1, w_2, ..., w_n]$ are weights of the points. The next step is calculating angle which was created between the straight line passing through the test keypoints and the X-axis of the coordinate system. It is known that tg angle of inclination graph of a function to the axis X is equal $a = tg\alpha$. Classification of the keypoints we condition from these angles. Assumed intervals from 0^o degrees to 360^o, with the increment every 5^o, for example. $0^o - 5^o$, $5^o - 10^o$ etc. In this way we obtain 72 intervals. In each of the intervals calculated distance between keypoints and the origin (5). Average distance in each of the compartments (6) are moved on a histogram, where the Y axis is normalized average distance, and the X axis the individual invtervals. Storing keypoints in this way causes that similar histograms are generated for similar objects belonging to the same class (Fig.2). Thanks to this solution, we expand the classical approach to search for similar images using keypoints. By generalizing the set of keypoints we created, it is possible to find a similar distribution of points for similar objects. In addition, the transformation of keypoints to histograms allows us to significantly reduce the amount of data that must be processed by a computer.

$$d_i = \sqrt{{x_i}^2 + {y_i}^2}, i = 1, ..., n , \tag{5}$$

$$\bar{d_j} = \frac{\sum_{i=1}^{n} d_i}{n}, j = 1, ..., 72 , \tag{6}$$

where n is the number of keypoints in tested interval. To compare the histograms we used a method of tree representation of the descriptors. This method originally was used in [18] for comparison of image descriptors generated by the SURF algorithm. This method makes it possible to substantially reduce the amount of required operation during the searching similar strings values, with the possibility to take into account the standard deviation between the values of the compared sequences. The advantage of this method is the ability to add new string value without having to rebuild the entire structure. The method has been adapted to handle the histograms by extending the length of the supported numeric strings from 64 as it was in the case of the SURF algorithm to 72 values, what represents a single histogram. The method works on the principle of generalization of string values. It uses the property resulting from the comparison of numerical sequences with taking into account the threshold of allowable deviation between the strings. It can be assumed that if the deviation of two

Fig. 2. Experimental results for the selected classes. As we can see, each of the subject classes generated different histograms, which however, are relatively similar to each other within every class.

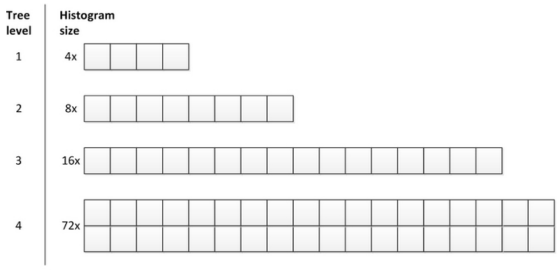

Fig. 3. Type of node descriptor in relations to node tree level

numerical strings is in the allowed range of the deviation between the sums of their values as well.

On the basis of the values of the numeric strings of all histograms, a tree is created which structure is presented in Fig.4. For each histogram is created a sub-tree and placed as the complement in the main tree. Figure 4a presents visualizations of the exemplary completion of a main tree to sub tree of the new histogram. In this process, in the first place searched nodes are shared between the trees, which values the deviation between the string of number is smallest, and does not exceed a given threshold (in the figure they are marked with a bright gray color). Other nodes (indicated in the figure in dark gray) of the sub tree will be copied to the main tree.

Each node of the tree, depending on its level in the hierarchy has a specific subsequence of numbers of histogram as shown in Fig.3, which is a generalization of string values of its child nodes. Free nodes, which are the lowest in the hierarchy tree, keep the original full sequence of 72 values of the image histogram with the name of the image. Then each subsequent node has a string value decreased

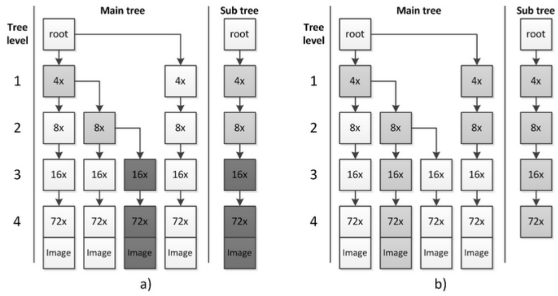

Fig. 4. Visualization of process of single histogram adding to the tree (a) and histogram comparison with the tree (b)

by summing the values of its predecessor. It increases the memory requirements of the method but simultaneously decreasing the number of steps during the image search. In the first step we compare nodes that have histograms that contains sequences of only four values instead of the 72 values of the full image histogram. In addition, each node allows to group other sub nodes if the deviation between their sequences of values does not exceed the threshold. During image searching the method creates a sub tree from the histogram of the searched image as it is in the adding process. Then intersections is performed between the sub tree and the main tree. Then it returns all images from leaves of the main tree that remain from intersection. The final result is a set of similar images.

5 Experimental Results

Experiments were performed on Caltech-101 [10] image database. The Caltech-101 database contains 9.145 images of average resolution of 300x300 pixels. Each image is also assigned to one of 101 image class. Images present various kinds of objects depending on the image category that include animals, plains, everyday objects, etc. For the experiments we limit the number of images per class to 50, because some of class contains much more images in comparison to others. The main task is to find from all the images, images of similar histograms generated from SURF keypoints. Histogram creation process have been discussed in detail in Section 3. After generating histogram the next step is to create from them a tree structure without division into classes. In the case of our experiment, our structure consist 4721 image histograms. The threshold of the maximum allowable deviation between the values of both histograms was set at 5.0. Then each image from the image database is searched in a tree structure. Each time the method finds at least one image that is its duplicate and other similar images. The result of this method is: the number of objects in a given class, the number of retrieved objects and the number of retrieved objects of the same class. The result of the method is also the number of comparison operations performed and number of all combination that would be needed to compare images in the traditional way. The results are presented in Table 2.

Table 2. First column represents class name of image group. Second column presents number of image in each class. Next three column presents dependence between number of matched image form same class to number of all matched images. The last three column presents dependence between number of compared nodes of the tree and number of combination between class image and entire image base.

Class Name	No. of images	No. of matched from same class	No. of matched all images	%	No. of compared node	No. of histgram combination	%
Scissors	39	39	40	97	41764	186069	22
Minaret	50	50	53	94	55418	238550	23
Saxophone	40	44	48	91	47809	190840	25
Faces easy	50	50	56	89	65598	238550	27
Leopards	50	50	56	89	55599	238550	23
Octopus	35	47	53	88	44512	166985	26
Lamp	50	50	58	86	61373	238550	25
Mayfly	40	44	52	84	43585	190840	22
Umbrella	50	50	59	84	59752	238550	25
Metronome	32	32	39	82	36625	152672	23
...							
Brain	50	103	537	19	65693	238550	27
Hedgehog	50	80	404	19	65203	238550	27
Watch	50	84	426	19	72290	238550	30
Cannon	43	48	258	18	55437	205153	27
Crab	50	60	323	18	63903	238550	26
Pizza	50	165	897	18	70091	238550	29
Cougar face	50	62	369	16	73065	238550	30
Gerenuk	34	42	255	16	46552	162214	28
Emu	50	61	404	15	71221	238550	29
Wild cat	34	50	329	15	47016	162214	28
SUM	4721	5561	16468	34	6004834	22523891	27

6 Conclusions and Future Work

On the basis of the experiments it can be concluded that the method can reduce the number of steps during the search in the database. Percent of compared nodes is only 27% of all the nodes which should be compared in a conventional manner. Also, most of the compared nodes are nodes with a smaller histogram than a full 72 values. From experimental results we see that most similar image are in classes like: pizza, brain, dollar_bill. The most unique images in same class and in the entire base are images from classes like: scissors, minaret, saxophone. The proposed method narrows down the number of results and will be useful in a situation when there is a need to quickly select a group of images most similar to the test image, out of a very large group of images.

Acknowledgments. The work presented in this paper was supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union.

References

1. Bartczuk, Ł., Przybył, A., Dziwiński, P.: Hybrid state variables - fuzzy logic modelling of nonlinear objects. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS, vol. 7894, pp. 227–234. Springer, Heidelberg (2013)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. 110(3), 346–359 (2008)
3. Bazarganigilani, M.: Optimized image feature selection using pairwise classifiers. Journal of Artificial Intelligence and Soft Computing Research 1(2), 147–153 (2011)
4. Chang, Y., Wang, Y., Chen, C., Ricanek, K.: Improved image-based automatic gender classification by feature selection. Journal of Artificial Intelligence and Soft Computing Research 1(3), 241–253 (2011)
5. Cpałka, K., Rutkowski, L.: Flexible takagi-sugeno fuzzy systems. In: Proceedings of 2005 IEEE International Joint Conference on Neural Networks, IJCNN 2005, vol. 3, pp. 1764–1769 (July 2005)
6. Cpałka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1851–1857 (2006)
7. Cpałka, K.: A new method for design and reduction of neuro-fuzzy classification systems. IEEE Transactions on Neural Networks 20(4), 701–714 (2009)
8. Cpałka, K., Rutkowski, L.: Flexible takagi sugeno neuro-fuzzy structures for non-linear approximation. WSEAS Transactions on Systems 4(9), 1450–1458 (2005)
9. Duda, P., Jaworski, M., Pietruczuk, L.: On the application of fourier series density estimation for image classification based on feature description. In: Proceedings of the 8th International Conference on Knowledge, Information and Creativity Support Systems, Krakow, Poland, November 7-9, pp. 81–91 (2013)
10. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding 106(1), 59–70 (2007), Special issue on Generative Model Based Vision
11. Gong, Y., Chuan, C.H., Xiaoyi, G.: Image indexing and retrieval based on color histograms. Multimedia Tools Appl. 2(2), 133–156 (1996)
12. Greblicki, W., Rutkowska, D., Rutkowski, L.: An orthogonal series estimate of time-varying regression. Annals of the Institute of Statistical Mathematics 35(1), 215–228 (1983)
13. Irshad, H., Roux, L., Racoceanu, D.: Multi-channels statistical and morphological features based mitosis detection in breast cancer histopathology. In: 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2013)
14. Hafner, J., Sawhney, H.S., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. IEEE Trans. Pattern Anal. Mach. Intell. 17(7), 729–736 (1995)
15. Kisku, D.R., Rattani, A., Grosso, E., Tistarelli, M.: Face identification by sift-based complete graph topology. CoRR abs/1002.0411 (2010)

16. Laskowski, Ł.: Hybrid-maximum neural network for depth analysis from stereo-image. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 47–55. Springer, Heidelberg (2010)
17. Laskowski, L.: A novel continuous dual mode neural network in stereo-matching process. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part III. LNCS, vol. 6354, pp. 294–297. Springer, Heidelberg (2010)
18. Najgebauer, P., Nowak, T., Romanowski, J., Gabryel, M., Korytkowski, M., Scherer, R.: Content-based image retrieval by dictionary of local feature descriptors. In: Proceedings of the 2014 International Joint Conference on Neural Networks, Beijing, July 6-11 (accepted for publication, 2014)
19. Nowicki, R.: Rough-neuro-fuzzy system with micog defuzzification. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1958–1965 (2006)
20. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. International Journal of Applied Mathematics and Computer Science 20(1), 55–67 (2010)
21. Nowicki, R., Rutkowski, L.: Soft techniques for bayesian classification. In: Neural Networks and Soft Computing, pp. 537–544. Springer (2003)
22. Peteiro-Barral, D., Guijarro Bardinas, B., Perez-Sanchez, B.: Learning from heterogeneously distributed data sets using artificial neural networks and genetic algorithms. Journal of Artificial Intelligence and Soft Computing Research 2(1), 5–20 (2012)
23. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
24. Przybył, A., Jelonkiewicz, J.: Genetic algorithm for observer parameters tuning in sensorless induction motor drive, 376–381 (2003)
25. Rutkowski, L., Cpałka, K.: Neuro-fuzzy systems derived from quasi-triangular norms. In: Proceedings of 2004 IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1031–1036 (July 2004)
26. Rutkowski, L.: Sequential estimates of probability densities by orthogonal series and their application in pattern classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-10(12), 918–920 (1980)
27. Rutkowski, L.: Sequential pattern recognition procedures derived from multiple fourier series. Pattern Recognition Letters 8(4), 213–216 (1988)
28. Rutkowski, L., Przybył, A., Cpałka, K., Er, M.: Online speed profile generation for industrial machine tool based on neuro-fuzzy approach. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part II. LNCS, vol. 6114, pp. 645–650. Springer, Heidelberg (2010)
29. Starczewski, J.T.: A type-1 approximation of interval type-2 FLS. In: Di Gesù, V., Pal, S.K., Petrosino, A. (eds.) WILF 2009. LNCS, vol. 5571, pp. 287–294. Springer, Heidelberg (2009)
30. Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision 7, 11–32 (1991)
31. Swain, M.J., Ballard, D.H.: Indexing via color histograms. In: Proceedings of the Third International Conference on Computer Vision, pp. 390–393 (December 1990)
32. Wallraven, C., Caputo, B., Graf, A.: Recognition with local features: The kernel recipe. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003, vol. 2, pp. 257–264. IEEE Computer Society, Washington, DC (2003)

33. Wang, L., Ju, H.: A robust blob detection and delineation method. In: Proceedings of the 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing, ETTANDGRS 2008, pp. 827–830. IEEE Computer Society, Washington, DC (2008)
34. Willamowski, J., Arregui, D., Csurka, G., Dance, C.R., Fan, L.: Categorizing nine visual classes using local appearance descriptors. In: ICPR Workshop on Learning for Adaptable Visual Systems (2004)
35. Zalasiński, M., Cpałka, K.: Novel algorithm for the on-line signature verification. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS, vol. 7268, pp. 362–367. Springer, Heidelberg (2012)