Automatic Object Classification and Image Retrieval by Sobel Edge Detection and Latent Semantic Methods

Vesna Zeljkovic[1] and Pavel Praks[2]

[1] School of Engineering & Computing Sciences, New York Institute of Technology (NYIT), Nanjing Campus
[2] Dept. of Applied Mathematics, VSB-Technical University of Ostrava, Czech Republic
dr.zeljkovic@gmail.com, pavel.praks@vsb.cz

Abstract. We perform in this paper a comparative study of ability of the proposed novel image retrieval algorithms to provide automated object classification invariant of rotation, translation and scaling. We analyze simple cosine similarity coefficient methods and the SVD-free Latent Semantic method with an alternative sparse representation of color images. Considering applied cosine similarity coefficient methods, the two following approaches were tested and compared: i) the processing of the whole image and ii) the processing of the image that contains edges extracted by the application of the Sobel edge detector. Numerical experiments on a real database sets indicate feasibility of the presented approach as automated object classification tool without special image pre-processing.

Keywords: Object Classification, Image Retrieval, Sparse Image Representation, SVD-free Latent Semantic Method, Cosine Similarity Coefficient, Sobel Edge Detector.

1 Introduction

Automatic object recognition and classification is very important and has numerous applications, such as image retrieval and robot navigation.

Rapid development of information technologies provides users an easy access to a large amount of multimedia data, for instance images and videos. Unfortunately, wide popular text retrieval techniques, which are based on keyword matching, are not efficient for describing rich multimedia context. Recently, wavelets and various methods of numerical linear algebra are successfully used for automated information retrieval and identification tasks [10-15]. Moreover, genetic programming is used as a tool for image feature synthesis and recognition [16, 17]. In this paper, a comparison of modified Sobel edge detection and Latent Semantic methods with an alternative sparse representation of color images for automatic object classification and retrieval is presented.

L. Atzori, J. Delgado, and D. Giusto (Eds.): MOBIMEDIA 2011, LNICST 79, pp. 204–216, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

2 Proposed Methods for Automatic Object Classification

We propose in this paper three different methods invariant of rotation, translation or scaling of the classified objects that successfully perform object classification on set of three different groups of objects Dinosaurs, Mummies and Sculls represented by images taken under various rotational, scaling and zooming conditions.

2.1 Sobel Edge Detector and Similarity Coefficient Methods

We applied two techniques for automatic object classification. We used Sobel edge filtered images for similarity computation in the first method and in the second method we applied simple cosine similarity coefficient on plain gray images with the goal to classify them.

The first technique implies procedure with an image converted to gray image with the extracted edges using Sobel edge detection method [1-7]. The idea behind this method is to significantly reduce the amount of data and filter out useless information, while preserving the important structural properties of an image and the targeted object.

Every image is processed as a two-dimensional $m \times n$ matrix image. We apply the two-dimensional Sobel masks to gray images. The Sobel operator performs a 2-D spatial gradient measurement on an image. It is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The Sobel edge detector uses a pair of 3×3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). After that the magnitude of the gradient is calculated. In the next step we applied cosine similarity coefficient [8]-[11] in order to extract the image containing the most similar object in the database.

In the second approach, we convert color images to gray scale images and process them. Then we apply simple cosine similarity coefficient [8]-[11] as in the first method.

In our initial study, we applied for both techniques image de-noising and pre-processing by wavelet filter application. Our numerical results pointed out, that the application of de-noising methods does not have any influence of the proposed algorithms to perform more successful object recognition. This additionally slowed down the algorithm so we concluded to omit that pre-processing stage.

We applied different edge detection functions and we have concluded, based on the obtained results that the Sobel edge detector gave the most clear and emphasized edge extracted results for the first proposed method.

In the current computer implementation of the proposed object recognition procedures, no pre-processing of images is assumed. The presented numerical experiments indicate optimistic application of the proposed techniques for object recognition and classification.

The colors of images are coded in Matlab (tm) as non-negative integral numbers and we did not use any scaling. The application of the proposed procedures can be written in Matlab as follows.

```
% Input:
% A ... the m × n document matrix
% Output:
% sim ... the vector of similarity coefficients
[m,n] = size(Image);
```

1. Calculate the gray image presentation for both proposed techniques:

```
Gray = rgb2gray(Image);
```

2. Apply Sobel edge detector on gray scale Image

```
ImageSobel = edge(Gray,'sobel');
```

3. Compute the similarity coefficients between two inspected images

```
xx = ImageSobel '*ImageSobel0;
%for the first method
%or
xx = Gray '*Gray0;
%for the second method
xx= xx/(norm(ImageSobel0)*norm(Sobel));
sim(i) = 1-acos(xx);
```

The proposed two algorithms give at the output the similarity coefficients *sim*. The absolute value of *i*-th element of *sim* coefficient is a measure of the similarity between two compared images.

Both algorithms give acceptable and competitive results. They are efficient, easy for implementation and fast enough for real application.

2.2 Latent Semantic Indexing Method

The Latent Semantic Indexing method (LSI) [12] was originally developed for automated text retrieval because of efficient matching of polysemy and synonymy. Moreover, LSI can be extended image retrieval [13-15]. A raster $m \times n$ image can be represented as a sequence of $m \times n$ pixels. Elements of this sequence represent colors of the original image. In order to achieve sparsity character of LSI-based image descriptor, FFT or similar technique with quantization can be applied [14]. The image preprocessing and retrieval can be done by the following steps:

```
Procedure IP [Image Preprocessing]
Input:
N images with the same resolution m × n,
Output: mn x N document matrix A
for j=1:N {for j-th image}
```

• Step A: Represent j-th image as a sequence of one-dimensional signal [14]. Let symbol A denote a $mn \times N$ term-document matrix related to mn keywords (pixels) in N images. The (i, j)-element of the term-document matrix A represents the color of i-th position in the j-th image document:

```
A(:,j)= reshape(j-th image,m*n,1)
```

• Step B: Sparse representation of images by DST transformation leaving unchanged top 1 % coefficient. The remaining 99 % unsignificant coefficients are set to zero (a quantization).

```
A(:,j)= dst(A(:,j)); A(1,j)= 0;
A(:,j)= quantize(A(:,j),0.01);
end;
```

After Step B, image database is represented by the sparse $mn \times N$ document matrix A.

• Step C: Latent Semantic Indexing.

Following [12, 14] the Latent Semantic Indexing method can be written as:

```
Procedure LSI [Latent Semantic Indexing]
function sim = lsi(A,q,k)
Input:
A . . . the mn × N matrix
q . . . the query vector
k . . . Compute k largest singular values and vectors; k ≤ N
Output: sim . . . the vector of similarity coefficients
[m,n] = size(A);
```

1. Compute the co-ordinates of all images in the k-dim space by the partial SVD of a document matrix A.

```
[U,S,V] = svds(A,k);
```

{Compute the k largest singular values of A; The rows of V contain the co-ordinates of images.}

2. Compute the co-ordinate of a query vector q

```
qc = q' * U * pinv(S);
```

{The vector q_c includes the co-ordinate of the query vector q; The matrix pinv(S) contains reciprocals of nonzeros singular values (a pseudoinverse). For more details please see Fig. 5 of [12].}

3. Compute the similarity coefficients between the query vector and images.

```
for j = 1:N  Loop over all images
sim(j)=(qc*V(j,:)')/(norm(qc)*norm(V(j,:)));
end;
```

{Compute the similarity coefficient for *i*-th image; *V(j, :)* denotes the *j*-th row of *V*.}

The procedure LSI returns to a user the vector of similarity coefficients *sim*. The *j*-th element of the vector *sim* contains a value which indicates a measure of a semantic similarity between the *j*-th document and the query document.

3 Numerical Results

The collection of three different groups, each containing 24 images, in total 24×3=72 color images was analyzed. The sample images representing versatility in scale, rotation and distance from the camera for all three groups is presented in Figure 1. The dimensions of images varied so all of them were set to the same width of 2000 pixels and the height of 2000 pixels. So the each picture is characterized by 4,000,000 attributes. For example, the name "D_3.jpg" implies that the third image from Dinosaurs group of images is considered. The analyzed image database is available for research purposes under an e-mail request.

a) Dinosaur

b) Mummy

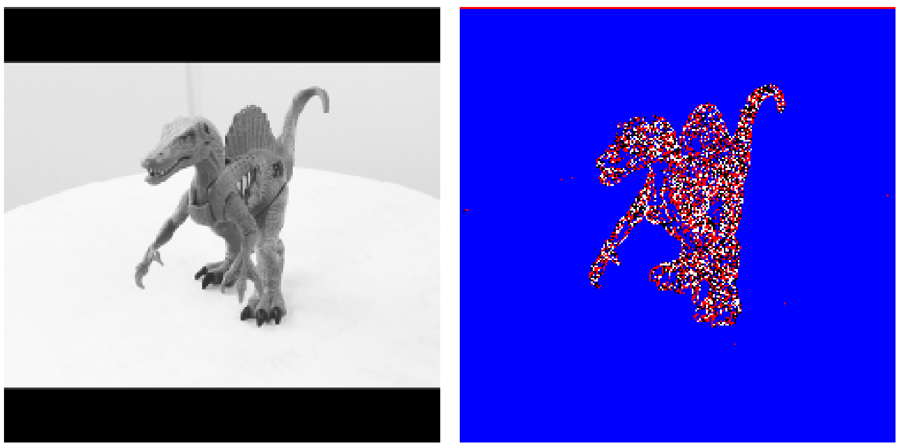
c) Scull

Fig. 1. Image database examples

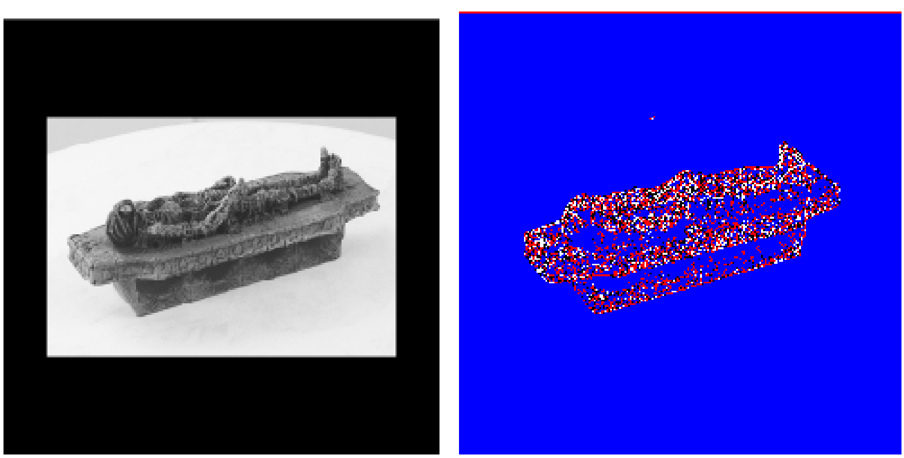
The queries were represented as images from the collection.

3.1 Results for Sobel Edge Detector and Similarity Coefficient Methods

Figure 2 represents: a) Gray scale converted and rescaled Dinosaur image No.1 and the same image after Sobel edge detector, b) Gray scale converted and rescaled Mummy image No.1, and the same image after Sobel edge detector and c) Gray scale converted and rescaled Scull image No.1, and the same image after Sobel edge detector. It can be observed that the filtered images contain an emphasized unique structure and edge elements present in the classified object. The Sobel edge extraction application enables us to extract useful information necessary for further object comparison and identification. It is obvious that all three filtered images also contain the background edges that could be drawback in object classification due to introduction of emphasized non useful information.

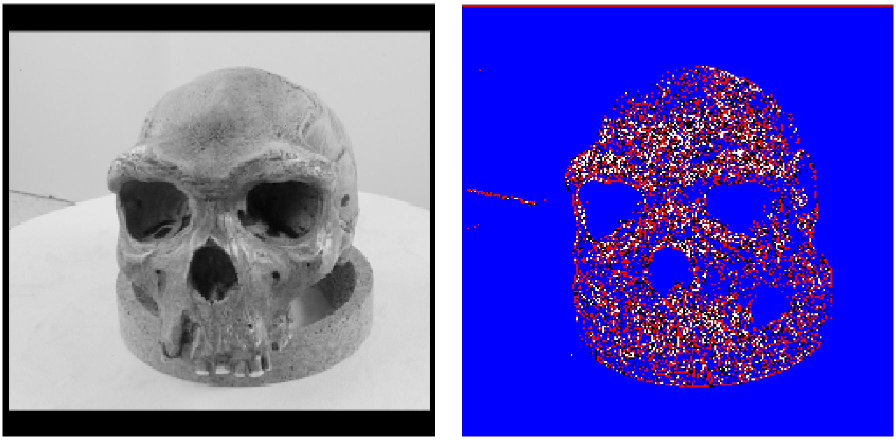


a) Dinosaur



b) Mummy

Fig. 2. Gray scale images and Sobel edge detector applied on database images



c) Scull

Fig. 2. (*continued*)

Table 1 contains the results obtained for the Dinosaur group of images used as query with both proposed algorithms.

Table 1. Dinosaur image retrieval result

	Method II		Method I	
Image	Similar	Similarity	Similar	Similarity
D_0	D_1	0.792098	D_22	-0.477922
D_1	D_2	0.825600	D_9	-0.462840
D_2	D_6	0.827910	M_11	-0.452429
D_3	D_2	0.793506	D_11	-0.467122
D_4	D_3	0.781580	D_13	-0.467995
D_5	D_2	0.818380	M_16	-0.468481
D_6	D_2	0.827910	D_20	-0.462955
D_7	D_0	0.781627	D_15	-0.480580
D_8	D_9	0.794382	D_7	-0.483005
D_9	D_10	0.801764	D_1	-0.462840
D_10	D_14	0.844410	M_11	-0.462095
D_11	D_10	0.809539	D_3	-0.467122
D_12	D_11	0.783882	D_3	-0.478209
D_13	D_11	0.791611	D_4	-0.467995
D_14	D_10	0.844410	D_10	-0.467348
D_15	D_14	0.795961	D_6	-0.463719
D_16	D_20	0.846660	M_21	-0.467024
D_17	D_16	0.839282	M_3	-0.479152
D_18	D_17	0.835905	M_16	-0.481522
D_19	D_16	0.834931	D_5	-0.480134
D_20	D_16	0.846660	D_6	-0.462955
D_21	D_20	0.840781	M_21	-0.488171
D_22	D_21	0.829590	D_0	-0.477922
D_23	D_16	0.838982	D_1	-0.473084

The first column in Table 1 represents the query image, second column represents the most similar image retrieved by each method and the third column is the maximum similarity coefficient determined by the applied technique.

Analyzing Table 1 we can conclude that we have 100% correct classification results obtained with the second method. The compared objects are recognized correctly. There are seven misclassifications obtained by the first method. Dinosaur images 2, 5, 10, 16, 17, 18 and 21 are wrongly classified as Mummy images 11, 16, 11, 21, 3, 16 and 21. The first method gave 70.83% correct classifications for the Dinosaurs group.

Table 2 contains the results obtained for the Mummy group of images used as query with both proposed algorithms.

Table 2. Mummy image retrieval result

	Method II		Method I	
Image	Similar	Similarity	Similar	Similarity
M_0	M_4	0.8098	M_22	-0.473274
M_1	M_5	0.8175	M_23	-0.473435
M_2	M_6	0.8638	M_6	-0.456423
M_3	M_7	0.8465	M_7	-0.449088
M_4	M_0	0.8098	M_18	-0.475947
M_5	M_1	0.8175	M_23	-0.474928
M_6	M_2	0.8638	M_2	-0.456423
M_7	M_3	0.8465	M_21	-0.448823
M_8	M_12	0.8213	M_22	-0.464342
M_9	M_13	0.7947	M_0	-0.485602
M_10	M_14	0.8489	M_14	-0.469516
M_11	M_15	0.8636	M_15	-0.448240
M_12	M_8	0.8213	M_18	-0.466055
M_13	M_9	0.7947	M_4	-0.486284
M_14	M_10	0.8489	M_10	-0.469516
M_15	M_11	0.8636	M_11	-0.448240
M_16	M_20	0.8793	M_20	-0.451026
M_17	M_21	0.8741	M_21	-0.444759
M_18	M_22	0.8662	M_22	-0.448271
M_19	M_23	0.8663	M_23	-0.454096
M_20	M_16	0.8793	M_16	-0.451026
M_21	M_17	0.8741	M_17	-0.444759
M_22	M_18	0.8662	M_18	-0.448271
M_23	M_19	0.8663	M_19	-0.454096

We obtained even better results for both methods for the Mummy group of images used as query images. The compared Mummy images are recognized correctly. Analyzing Table 2 we can conclude that we have 100% correct classification results obtained with both proposed methods for the Mummy query group.

Table 3 contains the results obtained for the Scull group of images used as query with both proposed algorithms.

Table 3. Scull image retrieval result

	Method II		Method I	
Image	Similar	Similarity	Similar	Similarity
S_0	S_7	0.7866	S_14	-0.492076
S_1	S_2	0.7869	S_16	-0.477059
S_2	S_1	0.7869	S_17	-0.477816
S_3	S_4	0.7853	S_18	-0.487566
S_4	S_5	0.7980	S_20	-0.480737
S_5	S_6	0.8294	S_20	-0.478156
S_6	S_5	0.8294	S_21	-0.475674
S_7	S_6	0.8002	S_22	-0.486436
S_8	S_15	0.8368	S_15	-0.476013
S_9	S_10	0.8483	S_10	-0.482384
S_10	S_11	0.8604	S_20	-0.474668
S_11	S_10	0.8604	M_17	-0.478968
S_12	S_11	0.8470	D_1	-0.478980
S_13	S_12	0.8367	S_21	-0.483276
S_14	S_15	0.8190	S_15	-0.478761
S_15	S_8	0.8368	S_8	-0.476013
S_16	S_17	0.7937	S_1	-0.477059
S_17	S_16	0.7937	S_2	-0.477816
S_18	S_19	0.7938	S_3	-0.487566
S_19	S_20	0.8010	S_4	-0.481828
S_20	S_21	0.8252	S_10	-0.474668
S_21	S_20	0.8252	S_6	-0.475674
S_22	S_21	0.8044	S_21	-0.484460
S_23	S_22	0.7932	S_16	-0.491731

The first column in Table 3 represents the query image, second column represents the most similar image retrieved by each method and the third column is the maximum similarity coefficient determined by the applied technique.

Analyzing Table 3 we can conclude that we have 100% correct classification results obtained with the second method as for the previous two query groups Dinosaur and Mummy. The compared objects are recognized correctly. There are only two misclassification obtained by the first method. Scull images 11 and 12 are wrongly classified as Mummy image 17 and Dinosaur image 1, respectively. The first method gave 91.67% correct classifications for the Scull group.

3.2 Results Obtained with Latent Semantic Indexing Method

In the first step, the input image was rescaled to 320×200 pixels. It means that the each image was represented by the 320×200=64 000 features. Moreover, in order to assume a color representation for the image retrieval, each image was represented by the RGB color model. In contrary to previous research [13-15], an alternative sparse coding of color images was implemented as a sequence of quantized FFT representations of RGB components. It means that the each image is represented by 3 × 64 000 sparse features, which gives us 192 000 sparse features of an image in summary. Thanks to the LSI preprocessing quantization, these color image features remain sparse. In other words, both the quantization and the user defined coefficient *k*=8 significantly reduced memory requirements of LSI [14]. As a result, the document matrix of 72 images allocated only 0.82 MB of the computer memory. Moreover, LSI can be implemented very effectively by solving a partial symmetric eigen-problem (so called the SVD-free approach [13]). For this reason, LSI computations required less than 0.1 seconds.

Image retrieval results are presented by decreasing order of similarity. In all cases, the query image is situated in the upper left corner, see Figures 3-6. The similarity of the query image and the retrieved image is written in parentheses. In order to achieve well arranged results, only 9 most significant images are presented.

LSI image retrieval results seem to be very promising. Images were correctly classified in all cases except one case, see Figure 6. It gives us the probability of failure 1/81=1.23 %. The remaining results are classified well: the most similar image is from the same group as the query.

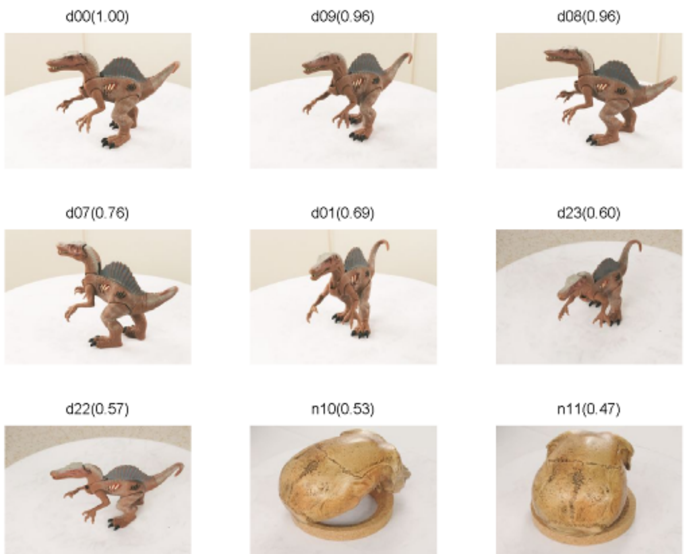


Fig. 3. An example of LSI image retrieval results: experiments with Dinosaur query image

Fig. 4. An example of LSI image retrieval results: experiments with Mummy query image

Fig. 5. An example of LSI image retrieval results: experiments with Scull query image

Fig. 6. An example of LSI image retrieval results: experiments with Dinosaur query image: failure recognition

4 Conclusion

Three approaches to the recognition and classification of different objects in various images are presented in this article. The results of the recognition test are promising and they show the ability of presented algorithms to successfully recognize various objects in real images. Of course, the quality of images and proper localization can influence the resulting errors by inaccurate localization. The proper localization will be subject to future work.

The proposed methods also show to be rotation, translation and scale invariant which open their potential application in wide range of areas.

We have applied two different methods to get proper object comparison and identification. The two following algorithms were tested and compared: the processing of the image with extracted edges and the processing of the whole image. As it can be observed from presented numerical results, both algorithms have shown compatible, accurate and comparable results. The advantage of the second proposed method is its simplicity, effectiveness, 100% correct classification results and practical implementation and realization.

An SVD-free sparse LSI algorithm with an alternative sparse representation of color images is presented as a tool for the object classification problem. In our experiments, the LSI algorithm was numerical stable. Results seem to be very robust: There is only one incorrectly detected case and it gives us 98.77% correct classification results.

Future research would concentrate on combining explicit mapping from presented low-level image features to semantic abstractions, which can be used for computer based interpretation of query images.

Acknowledgments. This work was partially supported by the Ministry of Education, Youth and Sport of the Czech Republic (Project No. 1M06047). Special thanks to Ruzica Vlajkovic and Milos Zeljkovic.

References

1. Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C.: Real-time volume graphics, pp. 112–114. AK Peters Ltd. (2006)
2. Jähne, B., Scharr, H., Körkel, S.: Principles of filter design. In: Handbook of Computer Vision and Applications. Academic Press (1999)
3. Farid, H., Simoncelli, E.P.: Differentiation of discrete multi-dimensional signals. IEEE Transactions on Image Processing 13(4), 496–508 (2004)
4. Kroon, D.: Numerical Optimization of Kernel Based Image Derivatives, Short Paper University Twente (2009)
5. Scharr, H.: Optimal Second Order Derivative Filter Families For Transparent Motion Estimation. In: 15th European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland, September 3-7 (2007)
6. Duda, R., Hart, P.: Pattern Classification and Scene Analysis, pp. 271–272. John Wiley and Sons
7. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall (2001)
8. Singhal, A., Buckley, C., Mitra, M.: Pivoted Document Length Normalization, NSF grant supported study, Cornell University Ithaca NY
9. Garcia, E.: Patents on Duplicated Content and Re-Ranking Methods. In: SES 2005: Advanced Track Issues: The Patent Files, San Jose, August 8-11 (2005)
10. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining, ch. 8, p. 500. Addison-Wesley (2005) ISBN: 0-321-32136-7
11. Zeljković, V., Praks, P.: A Comparative Study of Automated Iris Recognition Using the Biorthogonal Wavelets and the SVD-Free Latent Semantic Methods. In: ZNALOSTI 2007 Annual Conference on Knowledge Acquisition, Czecho-Slovak Knowledge Technology Conference, Ostrava, Czech Republic, February 21-23, pp. 143–154 (2007) ISBN: 978-80-248-1279-3
12. Berry, W.M., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. SIAM Review 37, 573–595 (1995)
13. Praks, P., Dvorský, J., Snášel, V., Černohorský, J.: On SVD-free latent semantic indexing for image retrieval for application in a hard industrial environment. In: IEEE ICIT 2003 (2003)
14. Praks, P., Kučera, R., Izquierdo, E.: The sparse image representation for automated image retrieval. In: IEEE ICIP 2008, pp. 25–28 (2008), doi:10.1109/ICIP.2008.4711682
15. Praus, P., Praks, P.: Hierarchical clustering of RGB surface water images based on MIA-LSI approach. Water SA 36(1), 143–150 (2010)
16. Krawiec, K., Bhanu, B.: Visual learning by evolutionary and coevolutionary feature synthesis. IEEE Trans. Evol. Comput. 11(5), 635–650 (2007)
17. Watchareeruetai, U., Takeuchi, Y., Matsumoto, T., Kudo, H., Ohnishi, N.: Redundancies in linear GP, canonical transformation, and its exploitation: a demonstration on image feature synthesis. Genet. Program Evolvable. 12, 49–77 (2011), doi:10.1007/s10710-010-9118-x