Marcin IWANOWSKI, Bartłomiej ZIELIŃSKI, Sebastian STYGAR
Warsaw University of Technology, LINGARO Sp. z o.o.

Comparing images using color blob approach

Abstract. *The paper presents novel approach to measuring image similarity. It is based on detection of regions of dominant colors, called color blobs, and computing their numerical description. The color image is transformed into a set of layered blobs characterized by both their color and some other parameters referring to their shape. In order to compare images, the numerical description of their layers of blobs is used to compute image dissimilarity measure. The results of experiments presented in the paper prove that the proposed method allows effective image comparison and can be used in image-based data search.*

Streszczenie. *W artykule przedstawiono nowe podejście do porównywania obrazów. Jest ono oparte na wykrywaniu obszarów kolorów dominujących (plam koloru) oraz na wyznaczaniu ich opisu numerycznego. Obraz kolorowy jest przekształcany w zbiór plam, połączonych w warstwy, definiowanych przez kolor oraz parametry odnoszące się do kształtu. Na podstawie opisów numerycznych warstw wyznaczana jest miara niepodobieństwa międzyobrazowego. Przedstawione w artykule wyniki przeprowadzonych eksperymentów dowodzą, że proponowana metoda pozwala na efektywne porównywanie obrazów i może być skutecznie wykorzystywana m.in. w wyszukiwaniu danych obrazowych.*
*(**Porównywanie obrazów z wykorzystaniem metody plam koloru**)*

Keywords: pattern recognition, image processing, image comparison, image similarity
Słowa kluczowe: rozpoznawanie wzorców, przetwarzanie obrazów, porównywanie obrazów, podobieństwo obrazów

Introduction

Image comparison is used in various computer science branches dealing with digital images. It is crucial for content-based image database and Internet search. Usually in those areas one provides a reference image and appropriate algorithm is searching the given data sources and comparing the stored images with the reference one. In some cases, however, not the images themselves are compared, but the measures computed upon the images. There are two reasons for using such measures. First, they allow to focus on the important properties of image content. Direct computation of image similarity (image-to-image) is particularly sensitive to changes of composition of elements of visual scene. The extraction of image features allows concentrating on the image content. The second reason for using measures is the computational efficiency. Instead of comparing images consisting of millions of pixels, one compares vectors consisting of, at most, few dozens of numbers. Having in mind that the image search task requires comparing, in some applications, huge number of images, the replacement of image by its measure allows significant improvement of computational effectiveness. What, however, should be mentioned, the metric-based comparison requires necessary preprocessing step, namely the computation of a measure of images from the database. This step is often called image indexing.

There are several methods of comparing color images. Some assume that considered images are of the same size and proportions. Comparison is then performed pixel-wise, and the similarity value relies on relation between pairs of corresponding pixels. Mean square error is an example of such approach. The main drawbacks of pixel-based methods are necessity to concern about images of different proportions and sensitivity to scaling and rotation. There exist algorithms which initially determine the pixel correspondence with one-to-many relation possible. It certainly is a profitable solution, however, determining the correspondence is usually a difficult task. Different approaches of comparing images have been proposed so far. For example, [1] presents image similarity indices based on distances of pixel intensity. Other methods take advantage of graphs [2] or trees [3], in order to assess image dissimilarity.

The paper presents a novel approach to image comparison. It is based on the assumption that the content of the image consists of objects characterized by their shape, position and color. Following this assumption, in order to produce image measure, at first the color image is transformed into a set of connected components characterized by both their color and some other parameters referring to their shape. In order to obtain these components, called blobs, at first the color quantization is performed of filtered image, next regions of homogeneous colors are individually morphologically filtered to preserve only the most important ones – color blobs. Blobs constitute a base for extraction of numerical data. Blobs of the same color are grouped into blob layers. The numerical description of a blob layer consists of color data as well as their spatial properties. Due to the fact that images may be described by different numbers of dominant colors, a special comparison scheme is proposed. It allows to find layer-to-layer correspondence depicting pairs of similar properties on two images. As a result, one gets a scalar value that describes the dissimilarity of two images.

Fig. 1. Original image (left) and filtered image (right).

Experiments that aimed at evaluating the efficiency of proposed method were performed. The experiments were conducted on the test set of images consisting of images from Kodak image database [4], both untouched and with special distortions. The results of test are presented in the paper in a separate section.

Detection of color blobs

Objects that are present within the image are characterized by their color, shape and position. Based on this observation we introduce a *color blob*, as a description of an object of visual scene presented in the image. Color blobs are thus geometrical descriptors of image content. In order to extract them, the three-stage procedure is used. It consists of image blurring, color quantization and morphological filtering.

The aim of the first step – image blurring – is to remove noise and local color variations. Such disturbances, if not removed, may result in incorrect results of processing in the next stages. Image blurring is performed by means of linear low-pass filter. In particular, local averaging or Gaussian filters may be used. The size of the filters should be carefully

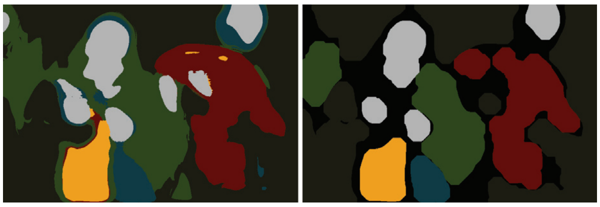

Fig. 2. Image with quantized colors (left) and color blobs (right).

chosen. On one hand, the filter size should be as large as possible. On the other, important image objects should be preserved, which limits size of the filter. Filtering itself might significantly modify object boundaries by introducing some shape blurring. This is usually perceived as a negative effect of linear low-pass filters, but in our case it has a positive influence – some shape simplification must be obtained what makes the approach insensitive to small variations of objects shape. Blurring causes such a shape simplification, which is particularly important in view of the next processing stage – color quantization.

The blurred image is next processed in order to find the most substantial colors. This is performed by means of color quantization. Various quantization approaches may be used, in experiments carried-out, the minimum variance method was applied[1]. As a result the image consisting of only at most given number of colors is obtained. These colors are referring to image objects.

Depending on the properties of image object that allow them to be recognized as a separate entities, different color parameters may be used during the quantization: one, two or all three color components in given color space that may be either Cartesian RGB, YUV or cylindrical e.g. HSV. For example, in case image objects are distinguishable based on their color hue value, the HSV color space should be used and the quantization should be performed based on scalar hue value.

As a result of quantization, we obtain an image having only limited number of colors. Consequently, a single image region characterized by relatively uniform (but not exactly the same) color of the input image has been transformed into a single-color region in quantized image. Apart from regions that refer to actual objects, also some small regions might appear. Inconsiderable size of a color region suggests its insignificant importance. Moreover, such regions can disrupt other – visually crucial – ones, and therefore should be removed. In addition to remove shape variations of the region, the shape simplification should be performed in this step. In order to obtain both results, removal and simplification, the morphological filter of opening is used. It consists of erosion followed by dilation [6, 7]. The shape and size of the structuring element should be chosen based on the spatial characteristics of blobs. In our experiments the disk-shaped element of radius equal to approx. 5% of smaller of image sizes was taken. Image obtained as a result of morphological filtering contains color blobs referring to most significant image objects. All described above processing steps are shown in Figs. 1 and 2.

Numerical description of blobs

Color blobs constitute a geometrical pixel-based description of the image content. The final goal is, however, to get the numerical description of this content in the form of a feature vector. In order to produce such description, blobs are grouped into blob layers containing blobs of the same color. Next, we extract layer features describing their properties.

Various binary image features might be used to describe blobs or their layers. In the current study, features are computed for each layer separately. Let S be the set of coordinates of pixels belonging to given blob layer. Let, moreover, the size of image along principal axes be equal to x_m and y_m. The numerical features that are used are following.

1. Size – the number of pixels belonging to $|S|$. In order to obtain the scale invariance, the size is divided by the number of pixels of the image, so finally we get:

(1) $$s = \frac{|S|}{x_m \cdot y_m}.$$

2. Layer color c. Depending on the meaning of color information in the image, understanding process and the quantization type, various approaches may be used. First of all, a single color component may be taken into consideration, for example hue layer from HSV color space and then $c = h$. In case quantization is performed in RGB color space, one may take simply the RGB-triple of values $c = [r, g, b]$. On the other hand, an image may be transformed, so that pixels are represented by scalars rather than vectors. Such a transformation can be achieved by applying a norm. For example (Euclidean and Mahattan norm, resp.):

(2) $$c = \sqrt{{c_1}^2 + {c_2}^2 + {c_3}^2},$$

(3) $$c = |c_1| + |c_2| + |c_3|,$$

where c_1, c_1 and c_3 stand for color components. Moreover, in RGB color space, a well-known norm can be used:

(4) $$c = 0.299\,r + 0.587\,g + 0.114\,b,$$

where r, g and b stand for values of color pixel in bands red, green and blue, respectively.

3. Centroid of a layer – average coordinates of pixels belonging to S (its center of gravity). Centroid is a vector consisting of two coordinates: $q = [q_x, q_y]$. To get the relative position of the layer, each coordinate of the centroid is divided by the appropriate image size:

(5) $$q_x = \frac{\sum_{p \in S} p_x}{|S| \cdot x_m}\ ;\ q_y = \frac{\sum_{p \in S} p_y}{|S| \cdot y_m},$$

where $p = [p_x, p_y]$ describes the coordinates of a pixel.

4. Average distance to the centroid among pixels belonging to the layer. As in the previous cases the issue of scale-invariance must be considered. In this case the value is divided by the length of diagonal of the image:

(6) $$a = \frac{\sum_{p \in S} dist(p, q)}{|S| \cdot \sqrt{x_m^2 + y_m^2}},$$

where $dist(p, q)$ stands for the distance between pixel p and centroid q.

5. Standard deviation of distances to the centroid, also normalized by using the length of image diagonal:

[1]The choice of the quantization method has an influence on the results of blob detection. This issue is however out of the scope of the current study.

$$(7) \qquad d = \sqrt{\frac{\sum_{p \in S} \left(dist(p,q) - a\right)^2}{(|S|-1) \cdot (x_m^2 + y_m^2)}}.$$

Concluding above, the entire feature vector that characterizes i-th blob is defined as follows:

$$(8) \qquad v = [s, c, q, a, d].$$

Note that q is represented by two values (q_x and q_y) and c by either single value (hue, luminance) or a triple of RGB components. The length of v equals thus either 6 of 8, depending on color representation.

Comparing two blob layer feature vectors

The elements of blob layer feature vectors are not of the same kind. For example, color value should be treated differently depending on how the color is described. It may also happen that, for particular image kind, considering human perception of image objects, the importance of particular features may be different to the others. Having in mind these restrictions, a special vector comparison scheme is proposed. Let $v = [s, c, q, a, d]$ and $v' = [s', c', q', a', d']$ be two feature vectors to be compared, $m(v, v')$ is the dissimilarity measure which shows how distant, in layer feature space, is one layer from another. The dissimilarity measures for particular features are computed as follows.

1. Size difference factor is calculated based on the following measure:

$$(9) \qquad m_s = \max\left(\frac{s}{s'}, \frac{s'}{s}\right).$$

2. The comparison of layers' color depends on the color representation. In case of the pixels represented as vectors, the dissimilarity measure may be equal to the mean value:

$$(10) \quad m_c = \frac{1}{3}\left(|c_1 - c_1{'}| + |c_2 - c_2{'}| + |c_3 - c_3{'}|\right).$$

In case of pixel represented as scalars, simple absolute value of difference is taken:

$$(11) \qquad m_c = |c - c'|.$$

If color is represented by hue value, due its cylindrical character, the appropriate measure should be computed as:

$$(12) \qquad m_c = \min\left(|h - h'|, h_{max} - |h - h'|\right),$$

where h_{max} stands for the highest possible hue value.

3. To compare centroids, one computes the Euclidean distance between them:

$$(13) \qquad m_q = \sqrt{(q_x - q'_x)^2 + (q_y - q'_y)^2}.$$

4. Finally, average distance to centroid and standard deviation of distances are compared by taking the absolute value of difference:

$$(14) \qquad m_a = |a - a'| \ ; \ m_d = |d - d'|.$$

The combination of all above measures is next computed in order to get single dissimilarity measure of two vectors using the following equation:

$$(15) \quad m(v, v') = (w_c \cdot m_c + w_g \cdot m_g + w_a \cdot m_a + w_d \cdot m_d) \cdot (s + s') \cdot m_s,$$

where $[w_c, w_g, w_a, w_d]$ are weights that are assigned manually which indicated the importance of particular features. The result of comparison of sizes is applied in a different way to other measures. Its usage promotes layers of similar size, and penalizes the cases when one layer is much larger than the other one.

Image comparison

The image preprocessing and feature extraction steps described in the previous sections result in numerical description of image blob layers. It consists of set of feature vectors as defined by Eq. (8) that constitute a complete description of the content replacing the underlying image for the comparative purposes.

The number of feature vectors depends on the number of layers. Moreover, due to both color quantization process and morphological filtering, it may happen that some layers disappear and consequently number of feature vectors decrease. It strongly influences the comparison process because one have to deal with the problem of comparing images characterized by different numbers of feature vectors.

The proposed approach is based on the dissimilarity matrix D containing the dissimilarities between layers from both images: $D_{i,j} = m(v_i, v'_j)$. This matrix is further processed in order to get single dissimilarity measure of two images. We consider layer-to-layer relation and express image dissimilarity as a sum of dissimilarities of layer pairs. If the number of layers in compared images is equal, we have one-to-one assignment. Let $a(i)$ be the assignment function that assigns i-th layer of the first image to $a(i)$-th one of the second image. Let A be the set of all possible assignments. The image dissimilarity measure m of two images I and I' is defined as:

$$(16) \qquad m(I, I') = \min_{a \in A} \left\{ \sum_{i=1}^{n} D_{i,a(i)} \right\},$$

where n stands for the number of layers present on each of images. According to (16) the dissimilarity measure of two images is based on finding such a pair assignment, that the sum of their dissimilarities is minimal.

Nevertheless, the number of layers in two images may be different. In such a case, some of them remain unpaired. The presence of such layers should also influence the image dissimilarity – they simply make images less similar one to another. In the proposed approach, this influence is modeled by means of penalty term that is added to the dissimilarity value computed according to the Eq. (16). The penalty of an unpaired layer i is calculated according to the formula:

$$(17) \qquad P = s_i \frac{1}{n} \sum_{j=1}^{n} m(v_i, v_j),$$

where n stands for the number of layers in that image where the number is greater. For images of different number of layers, we find thus such a assignment, that the final dissimilarity value, which takes into account the penalties for unpaired blobs, is minimal.

Testing procedure

In order to evaluate proposed solution, we created a dataset of images, basing on Kodak image dataset [4]. Apart from original images, our dataset contains distorted images. Assuming pixel value $c \in \langle 0, 255 \rangle$, the distortions enclose:

- brightness change by $l = 20$ or $l = -20$, according to the formula: $c' = c + l$, where c' stands for output pixel value,
- contrast change with factors $f = 0.8$ or $f = 1.2$, according to the formula: $c' = f(c - 255/2) + 255/2$, where c' stands for output pixel value,
- gamma-correction with exponent and 0.5 or 2,
- contamination with Gaussian white noise of mean 0 and variance 3901.5,
- contamination with salt and pepper noise of density 0.1,
- sharpening,
- blurring.

This way, the dataset consists of 264 images in 24 classes.

In this study, we take advantage of some standard measures for comparison needs. We apply the following measures for scalar pixel images (I and I'): mean square error (MSE), root mean square error (RMSE), average difference (AD), maximum difference (MD), defined, respectively, as:

(18)

$$MSE(I, I') = \frac{\sum_{i=1}^{x_m} \sum_{j=1}^{y_m} \left[I(i,j) - I'(i,j) \right]^2}{x_m \cdot y_m},$$

$$RMSE(I, I') = \sqrt{MSE(I, I')},$$

$$AD(I, I') = \frac{\sum_{i=1}^{x_m} \sum_{j=1}^{y_m} |I(i,j) - I'(i,j)|}{x_m \cdot y_m},$$

$$MD(I, I') = \max_{i=1 \ldots x_m, j=1 \ldots y_m} \left\{ |I(i,j) - I'(i,j)| \right\}.$$

We performed experiments to compare presented method with rival algorithms. For every approach, the testing procedure ran as follows. First, every pair of images was compared. In effect, we obtained a dissimilarity matrix. Next, the matrix was clustered with use of k-medoids algorithm [5]. This way, data was grouped into unlabeled clusters, representing, possibly, similar images. If the dissimilarity measure was reliable, then – truly – clusters should contain images from the same classes. We find label arrangement, so that percentage of correctly clustered images is highest, and treat the value as expression of dissimilarity measure quality.

Test results

The following tables contain results for both classical pixel-pixel measures and proposed approach. Table 1 presents results for standard approaches, for different color spaces and components. Table 2 shows results obtained for the proposed method, operating of hue band from HSV color space, with equal weight values $w_c = w_g = w_a = w_d$, for different numbers of layers in an image.

By adjusting weight values one can achieve better performance. In the experiments, we obtained result of 98.86, for desired number of layers equal 6, and different weights configuration.

Table 1. Percentage results of standard metrics.

Space (Component)	Formula	Metric			
		MSE	$RMSE$	AD	MD
RGB	(4)	88.64	88.64	84.47	92.05
	(2)	89.02	89.02	83.33	92.05
	(3)	88.26	88.26	84.47	91.29
HSV	(2)	87.50	88.64	89.39	30.68
	(3)	87.88	89.02	90.15	51.89
R	—	91.29	92.05	88.64	85.61
G	—	89.39	90.15	85.61	85.23
B	—	87.12	87.88	84.09	82.58
H	—	75.76	91.29	92.42	32.20
S	—	81.06	81.44	79.55	50.00
V	—	91.67	91.67	87.88	85.61

Table 2. Percentage results of proposed method for equal weights, for different numbers of blob layers.

Number of blob layers			
3	4	5	6
90.15	94.70	94.32	98.48

Conclusions

In this paper a novel approach to image comparison was proposed. It is based on detection of blobs of dominant colors that represents the image objects. The layers of blobs are transformed into their numerical description – feature vectors. Based on it the similarity measure between images is computed using the similarity matrix. The numerical description obtained is then used to compare images. The proposed method is flexible – it allows considering various features that refer to blobs' shapes. Also, thanks to weights assigned to every feature separately, one can adjust the algorithm to particular tasks, focusing e.g. on some shape properties rather that color, or reversely. The results of experiments presented in the paper show that the proposed method allows effective image comparison. The presented approach can be used to compare images in e.g. image-based data search.

Acknowledgements

This work was co-financed by the European Union within European Regional Development Fund, through grant Innovative Economy (UDA POIG.01.04.00-14-013/10-00).

BIBLIOGRAPHY

[1] V. Di Gesù and V. Starovoitov, "Distance-based functions for image comparison", *Pattern Recognition Letters* vol. 20, 1999.
[2] R. Baeza-Yates and G. Valiente, "An image similarity measure based on graph matching", *Seventh International Symposium on String Processing and Information Retrieval*, 2000.
[3] B. Zieliński and M. Iwanowski, "Comparing image objects using tree-based approach", *Computer Vision and Graphics*, Lecture Notes in Computer Science, vol. 7594, 2012.
[4] "Kodak lossless true color image suite", http://r0k.us/graphics/kodak (accessed in Dec. 2012)
[5] L. Kaufman and P. J. Rousseeuw, "Finding groups in data: an introduction to cluster analysis", Probability and Statistics, New York: Wiley–Interscience, 1990.
[6] M. Iwanowski, "Metody morfologiczne w przetwarzaniu obrazow cyfrowych", *EXIT*, 2009.
[7] P. Soille, "Morphological image analysis", *Springer Verlag*, 1999, 2004.

Authors: Marcin Iwanowski, Ph.D., D.Sc.; Bartłomiej Zieliński, M.Sc.; Institute of Control and Industrial Electronics, Faculty of Electrical Engineering, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland, email: marcin.iwanowski@ee.pw.edu.pl, bartlomiej.zielinski@ee.pw.edu.pl
Sebatian Stygar, M.Sc., Lingaro Sp. z o.o. ul. Puławska 99a, 02-595 Warszawa, Poland