Nonlinear Phenomena in Complex Systems, vol. 17, no. 3 (2014), pp. 231 - 241

Using Principles of Fractal Image Compression for Complexity Estimation of the Face Recognition Problem

D. A. Moskvin and V. V. Gluhov
Saint-Petersburg State Polytechnic University,
29 Politechnicheskaya Str, 195251 Saint-Petersburg, RUSSIA
(Received 14 April, 2014)

An approach to solve the problem of identifying individuals in static or dynamic images has been presented. The given approach is based on the methods of a fractal image compression. Algorithms for template image and current image processing have been proposed. For the proposed algorithms the computational complexity is estimated. In contrast with many other recognition algorithms the complexity of the proposed image processing scheme has linear dependence on the size of the image. The disadvantages of this algorithm are high complexity of the template image processing and linear dependence of the recognition speed on the size of the face database. Nevertheless, this algorithm can be effectively parallelized, e.g. with computing on GPU.

PACS numbers: 05.45.Df, 42.30.Sy
Keywords: fractal compression, face detection, face recognition

1. Introduction

Nowadays image recognition techniques are widely used in the security service starting with identification and authentication of computer users and ending with finding out terrorists in crowded areas. Effectiveness of the methods applied is determined by two main parameters. They are accuracy and pattern-recognition time. These parameters are closely connected and, as a rule, are inversely proportional to each other. In other words, the higher recognition accuracy is, the more time is required. Firstly, face recognition differs from other biometric systems. It does not need any special expensive equipment. You need only a personal computer and a usual video camera to solve the majority of problems. Secondly, there is no any physical contact with a device. Also there is no need to touch anything or to expose supersensitive crystalline lens using infrared rays. Most advanced image identification techniques allow providing sufficiently high accuracy (more than 90%), but the speed of image matching makes it possible to use them for visual processing in real-time mode. This is particularly evident during the implementation of different searching tasks in thread-specific data. For example, when we identify a human face on the video obtained from watching cameras in real time mode. In addition, existing image recognition techniques have some severe restrictions on acceptable angles of coverage - the deviation of an analyzed image from the surface of a template image should not exceed 15 degrees. This restriction is caused by using a standard photo taken full-face or half-face (a two-dimensional image) as a template image and it considerably influences the quality of the recognition because modern video control systems, as a rule, take photo of a person to the side at an angle or from above. The obtained images are compared using the original method [1] based on fractal geometry. The main peculiarity providing the high operation speed of the developed technique is that all operations demanding high performance are carried out only on the stage of the template image formation. Direct comparison of two images is performed using minimum operations.

2. Detection and Face Recognition Theory

Let us introduce main definitions:

- face characteristics are typical facial features of a person which uniquely identify him/her;
- a template image is an image obtained by any means and where the object for the comparison is represented;
- a compared image (a current image) is an image which is being analyzed to be compared with an template image image;
- a static image is a single exemplar of an image, mostly obtained by taking photos;
- a dynamic image (video sequence) is a set of static images, rigidly time-ordered, obtained mostly by video recording;
- a metric is a function set on space and returning the distance between two points of this space. If the metric is set in space, the last is called metric space.

The problem of identifying individuals implies two principal stages. The first stage is called "face detection" and involves determining location and size of the face in the picture. Usually this stage is the most time-consuming. After the location of the face is determined (within a certain accuracy), the second stage of identification - "face recognition" starts. To solve this recognition problem we may use statistical, neural network algorithms, Markovian networks (one-dimensional, pseudo-two-dimensional ones), elastic graphs, the wavelet analysis, the analysis of feature points and etc. Also the combination of these methods is possible. A human brain copes with the task of face detection in the pictures more than successful. Thus, it is obvious to determine and use the principles which the human brain uses to solve recognition problems. We may distinguish two directions among methods making such attempts: "top-down" recognition methods based on knowledge and "bottom-up" recognition methods based on features. "Top-down" recognition technique means finding out some set of rules which an image segment should comply with to be classified as a human face. This set of rules is an attempt to formalize some empirical knowledge how exactly the face looks in the pictures and what a person relies on taking a decision if he/she sees a face or not. It is quite easy to build up a set of simple and evident (as we think) features of the face image. For example, a human face is usually symmetric, facial features (eyes, nose, mouth) differ from the skin according to the brightness (usually a drastic change in the brightness corresponds to them), facial features are located in a certain way.

FIG. 1. Interest points of a building image detected using the Hessian matrix.

Allowing for the mentioned features, we may develop an algorithm checking their presence in the image segment. These methods also include recognition technique using predefined template matching. Templates set certain standard sample of the face image, for example, describing features of a certain facial area and their possible relative positions to each other. Face detection by means of the template includes checking each of image areas to be complied with the preset template. The principles of templates and other "top-

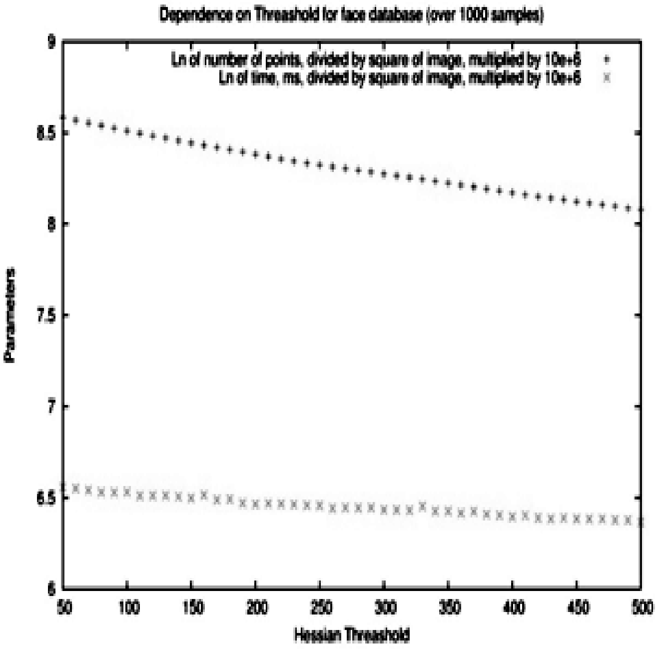

FIG. 2. Dependence of the number of keypoints on the Hessian Threshold.

down" recognition methods are mainly used in previous studies devoted to face detection [2–6]. There were the first attempts to formalize the features of a face image. Moreover, in that time computer powers did not allow using effectively more complicated methods of image recognition. Despite some simplicity of algorithms, we should not underestimate the value of those works, as a lot of methods successfully used today have been developed or adapted to this particular problem. A "bottom-up" recognition technique uses different invariant features of images of faces, based on the hypothesis that if a person may easily recognize a face in the picture without reference to his/her orientation, light conditions and individual features, so there are some signs of presence of faces in the picture with regard to the recording environment. The operation algorithm of "bottom-up" recognition techniques may be shortly described in the following way:

- detection of elements and features typical for an image of a face.
- analysis of detected peculiarities, making decision about the number of faces and their location.

In our work we have chosen the face detection method based on the "bottom-up" principle, namely, the Viola-Jones Object Detection using Haar Cascades [7, 8]. The recognition algorithm, according to the Viola Jones Object Detection, is based on "summing up" of pixels (with certain weight coefficients) under the window sliding on the raster. The recognition in this method is performed according to "precedents". Using the teaching selection, the set of "strong classifiers" is formed, each of them for a square window says: "presumably, there is a face in the window", or - "for certain, there is no face". Thus, for the algorithm to recognize the image in the window as the face, it is necessary for all "strong classifiers" (stages) to answer: «yes, presumably, there is a face B». If at least one of them rejects the window (answered that "for certain, there is no face"), then the algorithm immediately rejects this window, does not use other "strong classifiers" and proceeds to the next window.

3. Estimation of the complexity of the SURF Algorithm of interest point detection in a picture

One of the most popular and fastest algorithms of interest point detection in a picture is the SURF algorithm (Speeded Up Robust Feature) [10]. The method is detecting interest points using the Hessian matrix. The determinant of the Hessian matrix (so called the Hessian) reaches an extremum at points of the maximum change of the brightness gradient. It successfully detects spots, angles and edges of lines (Figure 1). The Hessian is invariant under rotations but is not scale invariant. Thus the SURF algorithm uses different-scale filters for Hessen detection. For each keypoint the direction of the maximum brightness variation (the gradient) and the scale taken from the scale coefficient of the Hessian matrix are calculated. The Gradient at the point is calculated using Haar filters. Let us consider the estimation of the complexity of the SURF algorithm performance. In the algorithm the integral representation of images is used. Suppose the width of the image is equal to w, h is a height, Thr_h is a Hessian threshold. To compute the integral representation we need $h \cdot w$ arithmetic operations. As the SURF uses the binarized approximation of the Laplacian of Gaussian, in order to compute the Hessian (the determinant of the Hessian matrix) we need 10^3 operations for computing convolutions according to associated filters and four operations for the Hessian computing. Four octaves and four filters in each octave are mainly used in the algorithm. Wherein, in adjoining octaves two filters are closed, so it is necessary to compute only ten filters. Also filters of the octave are not computed for all pixels, and the first octave is computed for every second pixel, the second one – for every fourth pixel, the third one – for every eighth pixel, the fourth one – for every sixteenth pixel. As a result, to process the first octave it is necessary to compute $4\frac{h}{2}\frac{w}{2} = h\ w$ Hessians. For processing the second, third and fourth octaves it is necessary to compute $\frac{h\ w}{8}$, $\frac{h\ w}{32}$ and $\frac{h\ w}{128}$ Hessians respectively. To find out the point of a true maximum, the interpolation of determined Hessians of the cube $3 \times 3 \times 3$ of a quadratic function is used [11]. Further, the derivative is computed (the finite-difference method of neighboring points). If it is close to zero, the point of a true maximum is found out. If the derivative is large, which means that there is a shift downwards, we should repeat the stepwise approximation till the derivative becomes less than the preset threshold. The number of points of a true maximum is detected using the Hessian Threshold and in general the linear dependence is not observed. However, when we frequently use the values of the Hessian (100÷500), the dependence of the number of detected points N_{kp} on the Hessian Threshold Thr_h may be defined by the function $N_{kp} \approx \frac{h\ w}{\exp(5.29+0.001Thr_h)}$. The dependence has been revealed in the face database consisting of more than 1000 faces taken as full-face photos (see figure 2). The number of arithmetic operations for computing the point of a true maximum for the cube under consideration is almost constant and approximately equal to 30. Then total number of operations is equal to $30N_{kp}$. The final number of arithmetic operations is approximately equal to $40w\ h+\frac{30h\ w}{\exp(5.29+0.001Thr_h)}$, so the SURF algorithm complexity possesses the linear dependence on the size image.

4. Estimation of the complexity for the methods of the distance detection between the points in the picture

To determine similarity of images, it is necessary to use some metric (the distance) in many-dimensional feature space. The "distance" between points is understood to be their "matching", i.e. some metric set in the feature space (for example, brightness) of the chosen points. When we work with images, the most frequent feature space of a certain point will be a two-dimensional vector of the brightness of the points from the square region outlined around this point. The popular metric is an Euclidean metric (an Euclidean distance). The Euclidean distance is determined as the geometric distance between points in many-dimensional space and is calculated according to the following formula

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

where $p{=}(p_1,\ \ldots,\ p_n)$ and $q{=}(q_1,\ \ldots,\ q_n)$ are two points in the Euclidean space. However, the Euclidean metric does not take into account the correlations between vectors and when operating at images it may produce a nonobjective result (for example, the Euclidean distance between two similar images different in brightness will be large). Therefore, we offer to use the generalization of the Euclidean distance introduced by an Indian statistician P.C. Mahalanobis in 1936 – the Mahalanobis distance [9]. The advantage of the Mahalanobis distance for the work with images is allowing for correlations between variables and the invariance to a scale. The Mahalanobis distance between two random vectors p and q is defined as

$$d_M(p,q) = \sqrt{(p-q)^T S^{-1}(p-q)}$$

where S is a covariance matrix of random vectors p and q. The covariance matrix is a matrix composed of pair-wise covariances of the elements of random vectors. The covariance matrix is determined from the following formula:

$$S = M[(p - Mp)(q - Mq)^T]$$

where M is a expectation function of random value. The covariance matrix is a square symmetric matrix. If the matrix of covariance is a unity matrix, then the Mahalanobis distance is equal to the Euclidean distance. Let us estimate the number of arithmetic operations which is necessary for calculating the Mahalanobis distance. Suppose the width is equal to w, height – h. At the first stage of calculating the Mahalanobis distance we make calculations of the average vectors which need $w(3h-1)$ arithmetic operations. The second stage consists of the calculations of the covariance matrix which have $4w^2h$ arithmetic operations. The third stage consists of the calculations of the reciprocal matrix for the covariance matrix. As the covariance matrix is symmetric, the Cholesky decomposition is used. For calculating the Cholesky decomposition it is necessary $3w + 3\frac{w(w-1)}{2} + \frac{w(w-1)(w-2)}{6}$ arithmetic operations. To convert triangular matrices calculated using the Cholesky decomposition we need $\frac{(w-1)w(w+1)}{3}$ arithmetic operations. The last stage consists of the vector multiplication of the reciprocal matrix which needs $w(w-1)+4$ arithmetic operations. Totally to calculate the Mahalanobis distance we need $w(3h + 4wh + 1 + w + \frac{(w-1)(2w+5)}{3}) + 4 \approx w^2(4h + \frac{2}{3}w) \approx 5hw^2$ arithmetic operations. Rarely the discriminant of the covariance matrix is equal to zero that is the matrix is not invertible and to calculate the distance fewer operations are necessary.

5. The proposed approach to the Face Recognition Problem

The approach introduced in the paper [11] consists of the detection of face typical features and establishing relationships between the detected features. The recognition problem is divided into two main subtasks being an template image processing and a current image processing.

5.1. Template image processing

Processing a template facial image includes the detection of typical features and facial features, recognizing similarities between detected facial features and the face. This work is carried out in several stages. The first stage involves searching keypoints in the picture. Stable interest points can be found, for example, using the SURF algorithm (Speeded Up Robust Feature). The result of the work of the algorithm is the set of the keypoints (the points of maximum brightness variation) which are supposed to be typical for the face being processed. The second stage consists of the construction of domain regions in accordance with the obtained keypoints. A rectangular or square area around each calculated keypoint is detected. The size of domain regions is fixed and they can be crossed and do not cover the whole image. The third stage involves the formation of range regions. One of the methods of image partitioning into range blocks is the quadtree method. At first, the coarse partitioning is performed, for example, subdividing the whole image into four squares (see figure 3). For each range block the algorithm is trying to search the domain and corresponding contracting mapping which in the best way covers the range block. Thus, if covering is within a reasonable error, then this range block is considered to be covered and the algorithm proceeds to the next range

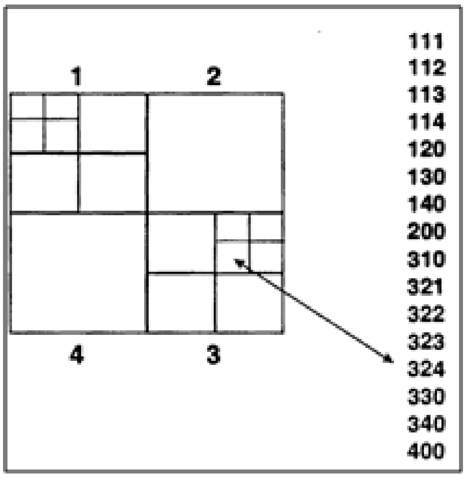

FIG. 3. The partitioning scheme using the quadtree method

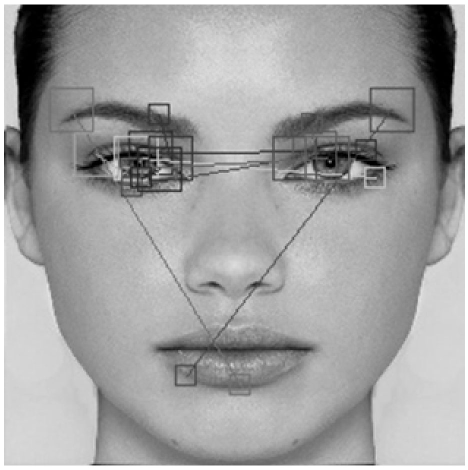

FIG. 4. The example of the detected pairs of domain and range regions

block. If the deviation does not stay within an acceptable error threshold, then the algorithm checks if the maximum depth of the quadtree is achieved. If the maximum depth of the quadtree is not achieved, the algorithm partitions the block into four smaller range blocks and the searching optimal domains and transformations starts again for these new range blocks. The process ends when all range blocks are covered or by means of such a domain and transformation selection providing the deviation within a reasonable error, or achieving the maximum depth of the quadtree. Figure 4 shows the indexation of range regions.

Another method of the construction of range regions is the sliding window method. The size of the sliding method is chosen (for instance, it may be equal to half the size of a domain region). Then the sliding window «moves» in the image from left to right, from top to bottom, and for each position of the sliding window the algorithm searches a domain region and corresponding contracting mapping which covers the window in the best way. A set of covered positions of the sliding window is taken as a set of range regions. At the fourth stage it is necessary to choose one of the existing image classification methods to calculate typical features of domain and range regions. After this the obtained features will be necessary to compare the regions to each other. As domain and range regions are not large with regard to the entire image of a face, then the speed of the analysis of these regions using image classification methods wont be considerable. Taking into account that the calculation of typical features of the detected regions is possible to carry out simultaneously, using parallel computing and the calculations on the graphics processor have become an effective solution for efficiency improvement. As the practice shows, using graphics processors to solve similar tasks considerably increases the analysis rate of the input data and therefore the entire system on the whole. It is possible to use any of the known methods today as the method of classification. For example, the classification based on the intensity histogram, the principal component analysis (PCA) (also known as Eigenfaces), the linear discretionary analysis and etc. Such methods allow indicating the compact set of features typical for regions, and it is possible to match them for a short period of time. If the sizes of ranges are not large this stage may be skipped and regions may be used as the vector. The output of the given stage is an attribute vector typical for the chosen classification algorithm for each domain and range regions. The fifth stage involves searching a range region for each domain region where the distance will be minimal. In the present case by the distance is understood to be the degree of region "matching", in other words the less the distance is the more similar regions are. The features of domain regions are compared with the features of range regions. Moreover, if forming features we have used the method which is not invariant to the rotations of 90° and also to the reflections, then it is necessary to compare them with changed copies of range regions respectively. For each domain region the range region having minimal distance from it is registered. Figure 4 shows the example of the detected pairs of domain and range regions using the Mahalanobis distance.

The computational result at this stage is the pairs of domain and range regions and also the number (or the vector) characterizing the distance of the range region from the domain one. Also in cases described above the result may be the coefficients of the necessary affine transformation of the domain region.

5.2. Current image processing

The current image processing consists of two stages. The first stage involves typical feature extraction in the current image. This stage significantly differs from the feature extraction in the template image processing. Regions similar to those from the current template image with account for all features and affine transformations are detected in the standardized image of a face. Since the face in the current image may be slightly distorted or turned, then it is possible to detect keypoints in a certain neighbourhood of their supposed location. Thus, it is possible to make a decision how the face is distorted using determined keypoints. In accordance with the supposed distortion, the face may be deformed in such a way as to decrease the distortion. Besides, for determined regions attribute vectors are constructed. Then they are compared by forming the distance between regions. The difference between this stage and similar stage in the template image processing is that you do not need to search the pairs of domain and range regions. The complexity of this searching is exponential. The second stage involves the comparison of typical features of template image and current images. In other words, it involves the comparison of the distances between similar pair regions. When the dependencies of regions in the template image and the dependencies of regions in the current image coincide, we may come to a conclusion that all these images belong to the same class. This implies that they are in some way equivalents and according to the problem, it means that a human face that is before the camera corresponds to the face of an individual from the face database. The comparison may be performed using different decision techniques, for example, such as fuzzy logic, the threshold value or neural networks.

5.3. The algorithm of face recognition

Based on the methods of the template image and current image processing described in sections 5.1 and 5.2 we offer to use the following algorithms with following algorithm parameters:

- the sizes of the image of a face: $hFace$ is a height of the image of a face, $wFace$ – the width of the image of a face,
- $scaleDom$ is a dimensional ratio of the linear sizes of a domain region to the sizes of the image,
- $shiftRank$ is a shift of the sliding window of the range region detection relative to the previous position in pixels,
- $pairThreshold$ is a threshold value when selecting the pairs of domain and range regions,
- $attributeThreshold$ is a threshold value for the comparison of the distances between the pairs of domain and corresponding

range regions in the template image and current image.

Input parameters of the algorithm of the template image processing are is an image img of a face and $PersonID$ being a identifier of a certain individual in img. The first stage consists of searching keypoints using the SURF algorithm (without the calculation of descriptors). The second stage involves the formation of rectangular domain regions around keypoints. If a keypoint is close to the edge of the image and the obtained domain region goes beyond the edge, then such a point is not considered. Each domain region is compressed to the sizes of the range region and it means that linear dimensions are halved. In this case each pixel of the compressed region is assigned to the brightness which is equal to the arithmetic average of the brightness of the corresponding square region with the size of 2x2 pixels of the uncompressed region. The third stage involves searching range regions. The sliding window is used for this searching. For each position of the sliding window all compressed domain regions are selected and the Mahalanobis distance between them is calculated. Consequently, no other transformations are applied to them. If the distance between pairs exceeds the preset *pairThreshold*, such a pair is discarded.

1. Reduction the face (img) to the preset sizes $hFace \times wFace$.

2. To find out a lot of keypoints using the algorithm SURF. To set aside those keypoints whose distance to upper and lower edges of the image does not exceed $scaleDom \cdot hFace$,to left and right edges does not exceed $scaleDom \cdot wFace$. Let us set the number of keypoints equal to N_{kp}.

3. If N_{kp} is equal to zero the algorithm ends the work and returns false.

4. To detect square domain regions $\{D_i\}$ around keypoints with the dimension of $(scaleDom \cdot hFace) \times (scaleDom \cdot wFace)$. To find a lot of changed domain regions $\{D'_i\}$ using the compression of detected regions to the size $[\frac{scaleDom \cdot hFace}{2}] \times [\frac{scaleDom \cdot wFace}{2}]$. The compression is performed via assigning to each block the pixel of one pixel containing the arithmetic average of brightness of chosen four pixels (starting with an upper left corner).

5. To select all possible range using the sliding window with the size of $[\frac{scaleDom \cdot hFace}{2}] \times [\frac{scaleDom \cdot wFace}{2}]$. The sliding window moves from left to right and bottom-up with preset shift $shiftRank$. For each position of the sliding window:

   (a) To look over compressed domain regions $\{D'_i\}$:

      i. To calculate the Mahalanobis distance D_M between the chosen window and domain regions.

      ii. If the distance D_M does not exceed $pairThreshold$, the region of the current position of the window (let us set as R_i) and the chosen compressed domain region D'_i are registered in the number of pairs of similar regions DR: $DR = DR \cup \{D'_i,\ R_j\}$.

   (b) If the number of the pairs of regions in the function DR is equal to 0, then it will return false.

.

6. To add a new $FaceID$ into the face database for the preset $PersonID$ and for all pairs of domain and range coordinates xD, yD, xR, yR and also add the sizes hD, wD, hR, wR and the Mahalanobis distance D_M between regions.

7. Return true.

The input parameter of the algorithm of the current image processing is img. The algorithm is selecting all human faces in the face database until it finds out the best matching of the current and template images. Domain and range regions are formed according to the preset coordinates and sizes of the given face. Then domain regions are compressed to the sizes of range regions and after that the Mahalanobis distance between these pairs is calculated. Then the arithmetic average of the determined distances is calculated. For all faces the minimal arithmetic average among all of them is found. If this minimal value exceeds *attributeThreshold*, the face is considered to be unknown. Otherwise, the $PersonID$ of the appropriate person returns.

1. To process the face (img) in accordance with the preset sizes $hFace \times wFace$.

2. For each face from the database of the template image, images referring to a certain individual (searching according to PersonID):

    (a) For each element (face) from the database of the chosen class $PersonID$ (searching according to $FaceID$):

    i. To calculate coordinates of the pairs of domain and range regions xD, yD, xR, yR, sizes hD, wD, hR, wR and coordinatewise vector difference from the database for the present $PersonID$ and $FaceID$.

    ii. To select corresponding pairs $DR = \cup\{D_i,\ R_j\}$ of domain and range regions according to the coordinates xD, yD, xR, yR and sizes hD, wD, hR, wR.

    iii. To compress each domain region D_i to the size of corresponding range regions R_j. The compression is performed in the same way as in the template image processing. Let us set the obtained compressed domain region as D'_i.

    iv. To calculate the Mahalanobis distance $D_{M,i,j}$ between all pairs D_i and R_j.

    v. To calculate the arithmetic average D_M for computing $D_{M,i,j}$.

3. To choose the minimal among all the received values of D_M. If it does not exceed *attributeThreshold*, we should return $PersonID$ of the corresponding class. Otherwise, we should return 0.

6. Estimation of the speed and accuracy of the proposed algorithms

We give the estimation of the speed and accuracy of the template image and current image processing and also some conceptions which let us estimate the recognition accuracy.

6.1. Estimation of the speed of the current image processing

The given stage consists of searching domain and range regions according to the known coordinates, applying affine mapping to domain regions, the calculation of the Mahalanobis distance. Thus, the following operations are the most time-consuming.

1. To process the facial image in accordance with sizes $hFace \times wFace$. Thus, to perform this stage we need the number of arithmetic operations which are roughly equal to the initial size of the image.

2. The compression of detected domain regions is doubled. Let us set the number of domain regions as N_D. Then, for the performance of the given step we need approximately $N_D \cdot scaleDom^2 \cdot hFace \cdot wFace$ arithmetic operations.

3. The most time-consuming stage is the calculation of the Mahalanobis distance for each pair of domain and range regions.

4. The calculation of the arithmetic average among obtained Mahalanobis distances. For this operation it is necessary to perform the number of operations equal to the number of pairs of domain and range regions. With a minor error we may assume that the number of such pairs is equal to N_D.

Taking into consideration the evaluation of the computational complexity of the Mahalanobis distance, the cumulative quantity of arithmetic operations required for the current image processing is roughly equal to $NF \cdot hFace \cdot wFace(1 + N_D \ \ scaleDom^2 + \frac{scaleDom^3 \cdot wFace}{2})$ where NF is a number of faces in the face database. However, matching with each face from the face database is performed independently of other matching, so the given algorithms may be effectively parallelized. Besides, it should be noted that the complexity of matching is linear from the sizes of the image in comparison with a lot of other algorithms (for example, in the main components method the complexity of the current image processing quadratically depends upon the sizes of the image). "Price to pay" for the high speed of the current image processing is the high complexity of the template image processing.

6.2. Speed rating of the template image processing

The complexity of the given stage is considerably higher than the complexity of the current image processing. Main time expenditures account for searching stable domain regions and corresponding range regions. The first stage involves searching keypoints using the SURF algorithm and searching range regions. The estimation of the complexity of the SURF algorithm is given in section 3. After the SURF algorithm is computed, all candidates for range regions are sorted out. Let us set the sizes of the compressed domain or range regions as $h = \frac{scaleDom \cdot hFace}{2}$, $w = \frac{scaleDom \cdot wFace}{2}$. As the shift of the sliding window is equal to $shiftRank$, the number of such candidates is equal to $\frac{(wFace-w)(hFace-h)}{shiftRank^2}$. Using the estimation of the quantity of arithmetic operations while calculating the Mahalanobis distance from section 4, we obtain the final number of the operations to perform this stage: $5h \ \ w^2 N_{kp} \frac{(wFace-w)(hFace-h)}{shiftRank^2}$. As in the processing algorithm of the template image, the direction of the maximum brightness variation and descriptors of interest points do not take into account, resulting complexity may be defined as $40 \ wFace \cdot hFace + \frac{5 \ hFace \cdot wFace}{\exp(5.29+0.001 \ Thr_h)a}$ where $a = 6 + \frac{wFace^3 \cdot hFace^2 \cdot scaleDom^3 \ (1-\frac{scaleDom}{2})^2}{8 \ shiftRank^2}$.

6.3. Estimation of the recognition accuracy

A lot of parameters including the sizes of the attribute space, threshold values, the quality of the image (resolution, face location, the uniformity of illumination) influence the accuracy. It makes sense to perform theoretical estimation of the accuracy of the proposed algorithm in several steps. It is assumed that the face detection has been performed and the image has been standardized. It is necessary to evaluate the probability of the detection if faces belong to the space p and the probability of successful face recognition q. Domain regions typical for the given image are detected in the template image. If the number of such regions is sufficiently large, they describe the chosen face with high probability and without compromising accuracy. Let us set corresponding probabilities as $p^d \approx p$, $q^d \approx q$. Range regions due to their similarity to domain regions, on the one hand, and the small size, on the other hand, provide some increase in the probability of the detection if faces belong to the space p and the decrease in the probability of a successful face recognition q. This change of probability depends on many parameters and more accurate estimates may be obtained by an experimental approach. Let us set $p^r \approx pc$, $q^r \approx \frac{q}{d}$ where parameters c and d may be determined experimentally. Due to the fact that the images of domain regions do not significantly differ from the range regions, the accuracy of the method increases. If we consider that the parameters of the algorithm are fixed, then there probability of the detection may be roughly defined as $p' = 1-(1-p^d)(1-p^r)$ and the probability of a successful face recognition may be defined as $q' = 1-(1-q^d)(1-q^r)$ provided faces belong to the space. Let us estimate the influence of the parameters values on the recognition accuracy. The increase of the sizes of a face ($hFace$, $wFace$) and the decrease of the sliding window shift ($shiftRank$) lead to the increase in accuracy of the method but to decrease in speed. If the sizes of the domain and range regions with regard to the whole image ($scaleDom$) increase, it leads to the decrease in speed and in some value to the increase in the accuracy of the method. But then the accuracy decreases due to the presence of superfluous information about the keypoint. In future it is expected to use different sizes of domain regions according to the size of the region of the keypoint. The parameters $pairThreshold$ (the threshold value while selecting the pairs of domain and range regions) and $attributeThreshold$ (the threshold value for the comparison of the distances between the pairs of domain and corresponding range regions in template image and current images) are the threshold values. Their change leads to the simultaneous increase of one probability and the decrease of other probability from the pair p and q. More accurate estimates of the quality of the recognition should be performed in practice. Thus, we may come to the conclusion that methods and algorithms used in the proposed approach allow us to solve the problem of face recognition with higher accuracy and speed in comparison with other analogs. It enables applying the given approach for processing the streaming video of high-resolution in a real-time mode.

References

[1] D.P. Zegzhda, D.A. Moskvin, U.O. Bosov. Pattern Identification Based on Fractal Compression. Information Security Problems. No.2, 86-90 (2012).

[2] G. Yang, T. S. Huang. Human Face Detection in Complex Background. Pattern Recognition. No. 27, 53-63 (1994).

[3] C. Kotropoulos, I. Pitas. Rule-Based Face Detection in Frontal Views. In: *Conference Proc. Int. Conf. Acoustics*, 1997.

[4] T. Sakai, M. Nagao, S. Fujibayashi. Line Extraction and Pattern Detection in a Photograph. Pattern Recognition. No. 1, 233-248 (1969).

[5] H.E. Craw, J. Lishman. Automatic Extraction of Face Features. Pattern Recognition Letters. No. 5, 183-187 (1987).

[6] V. Govindaraju. Locating Human Faces in Photographs. Int. J. Computer Vision. No. 19, 129-146 (1996).

[7] P. Viola, M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In: *Accepted Conference on Computer Vision and Pattern Recognition*, 2001.

[8] P. Viola, M.l Jones. Robust Real-time Object Detection. In: *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, 2001.

[9] P.C. Mahalanobis. On the Generalized Distance in Statistics. J. Asiat. Soc. Bengal. No.26, 541-588 (1930).

[10] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU). No. 110(3), 346-359 (2008).

[11] M. Brown, D. Lowe. Invariant features from interest point groups. BMVC. No. 11 (2002).