STUDIA INFORMATICA 2009
Volume 30 Number 2A (83)

Michał KAWULOK
Politechnika Śląska, Instytut Informatyki
Janusz SZYMANEK
Future Processing, Research Lab

ALGORITHM FOR PRECISE FRONTAL FACE DETECTION

Summary. The paper presents an algorithm for frontal face detection. At first a set of face candidates is selected based on ellipse detection with the Hough transform. Subsequently, every candidate is verified and for positive verification the detection precision is improved, which is particularly important for face recognition purposes. Results of conducted experiments, which are discussed in the paper, confirm high speed and effectiveness of the algorithm.

Keywords: face detection, generalized Hough transform, Support Vector Machines

ALGORYTM PRECYZYJNEJ DETEKCJI TWARZY FRONTALNYCH

Streszczenie. Artykuł opisuje algorytm detekcji twarzy frontalnych. W pierwszym etapie przeprowadzana jest detekcja elips za pomocą transformaty Hougha, mająca na celu wyłonienie zbioru kandydatów na twarze. Następnie każdy z kandydatów jest weryfikowany i w przypadku pozytywnej weryfikacji stosowane są zabiegi podnoszące precyzję detekcji. W artykule zostały przedstawione wyniki badań eksperymentalnych potwierdzające wysoką skuteczność i szybkość działania algorytmu.

Słowa kluczowe: detekcja twarzy, uogólniona transformata Hougha, Mechanizm Wektorów Wspierających

1. Introduction

Face detection is a commonly performed and very important task in our daily communication and therefore it is one of the most investigated areas of computer vision [17]. It is the first and indispensable step of automatic face recognition [18], but its applications are also concerned with other areas of face analysis including surveillance tracking, human-computer interfaces and entertainment purposes. Face analysis may be used for image indexing purposes and utilized in multimedia databases. By definition it is aimed at determining whether there are any faces in the image and in case of a positive answer, finding their exact locations. If faces are oriented frontally, which is assumed in the presented approach, their position is unambiguously defined by central points of eyes. Face localization is a simplified case of face detection and analyzes an image with an assumption that it contains exactly one face.

1.1. Related work

There are many various techniques for face detection and localization which utilize common properties of human faces [17]. There are some methods which are based on heuristic rules to define a face or facial features. These rules can be applied in top-down manner to narrow down the searched area and potential locations of faces [9, 16], but usually they occur ineffective for detecting multiple faces. It is also possible to detect facial features first and then try to match them into faces. For example, eyes can be detected with morphological operations [5] and then matched into potential face regions.

There are also many algorithms which detect faces and skin regions based on skin color. These methods can be divided into parametric models which operate based on fixed decision rules defined for various color spaces [7] and statistical models which require appropriate training [8]. Skin color detection may be helpful in detecting human faces, but it is extremely sensitive to lighting condition changes and its reliability is not high enough if it is used alone. There are some methods which combine skin color detection with other methods, for example an approach proposed by Hsu et al. [7] utilizes skin color detection, morphological operations and ellipse detection.

Face appearance can be also learned from a representative training set to make it possible to determine whether a given image region presents a face or not. Such approach was presented in the Eigenfaces method for face detection and recognition [15]. At first, principal component analysis is performed over a set of face images to create a face space. In order to determine whether an image presents a face, it is projected onto the face space and the decision is made depending on the projection error. Such classification can be also performed with learning machines such as neural networks [14] or Support Vector Machines [11]. Such approach may be effective, but the whole image must be analyzed with varying size of scanning window which seriously affects the performance.

1.2. Outline of the proposed approach

In the presented research face detection is performed in luminance channel based on geometric properties of human faces. We utilized and developed the idea proposed by Maio and Maltoni [9] for face localization. In that approach ellipses are detected in a directional image with generalized Hough transform to find approximate face location and then direction histograms are used within the ellipse to determine position of eyes and mouth. Ellipse detection proposed by Maio and Maltoni [9] is described in section 2. We adapted this technique for preliminary selection of potential locations of faces (called face candidates). After that, we repeat ellipse detection inside the primarily detected ellipses to find candidates for eye sockets. These candidates are paired with simple heuristic rules to obtain face candidates which are normalized and verified with the Support Vector Machines (SVM) [3]. In case of a positive verification, the positions of eyes are updated to maximize the SVM response. Verification of face and eye images with the SVM is addressed in section 3. Detailed description of the proposed algorithm is presented in section 4 and experimental results are shown and discussed in section 5. Section 6 concludes the paper.

2. Ellipse detection

Ellipse detection proposed by Maio and Maltoni [9] is performed in directional image with a generalized Hough transform [1, 5]. In this section we describe the techniques which are used for transforming an intensity image into a directional one and how the ellipse detection is performed.

2.1. Directional image

Detection of directions and edges in images is a widely investigated area of image processing and there have been many methods proposed for solving this task. In general it is performed by low-level image analysis based on first-order derivative of brightness. Larger values of image gradient induce higher probability of observing an edge. However, the main difficulty is concerned with uncertainty of what should be regarded as an edge, which usually cannot be solved without extracting higher-level information from the image. It is therefore necessary to select the most appropriate low-level methods depending on the application.

In our approach we adapted an algorithm proposed by Donahue and Rokhlin for determining the averaged tangent direction [4]. Its main idea is that tangent direction is approximated for a group of gradient vectors calculated in the input image.

For every pixel $I(m,n)$ the surface normal vector is calculated based on a group of neighboring pixels. We achieved the best results if for every 3x3 group of pixels the gradient direction was calculated in this way:

$$g(m,n) = \begin{bmatrix} a + b + 2I_{m,n+1} - 2I_{m,n-1} \\ a - b - 2I_{m-1,n} - 2I_{m+1,n} \end{bmatrix} \quad \begin{matrix} a = I_{m-1,n+1} - I_{m-1,n-1} \\ b = I_{m+1,n+1} - I_{m+1,n-1} \end{matrix} \tag{1}$$

This operation (1) transforms the input image into a gradient image, in which tangent directions, perpendicular to the gradient vectors, are calculated. Every 3x3 group of gradient vectors ($\{\boldsymbol{g}_i\}$) is averaged with one tangent direction ($\mathbf{u}$) based on 25 gradients in 5x5 neighborhood. Tangent vector is a unit vector which minimizes the error function:

$$\delta(\mathbf{u}) = \sum_k (\mathbf{g}_k \cdot \mathbf{u})^2 \ . \tag{2}$$

The function is minimized analytically and tangent phase and magnitude are found. If variance of the gradient directions is high within a group of pixels, the error of averaging (2) is high as well and detected tangent direction is not reliable. This error is normalized to eliminate length of the gradient vectors:

$$E = \delta_{\min}(\mathbf{u}) / \sum_k (\mathbf{g}_k)^2 \ . \tag{3}$$

The directions, for which the error is greater than a threshold value (E_{th}), are considered as unreliable and rejected. Every detected tangent direction is characterized by its location, angle and magnitude defined as a sum of squared contributing gradients. The magnitude means how strong a direction is and only the directions with high magnitudes are selected.

2.2. Hough transform

The generalized Hough transform [1] is a technique for detecting parameterized shapes in digital images. It works by transforming the input image (usually a directional image) to a parameter space, which is called the accumulator. Every pixel or direction is considered as a part of potential shape and contributes to the accumulator by adding the most probable values of the parameters. Local maxima in the accumulator indicate the most probable values of parameters which define shape location.

In the analyzed case ellipses of fixed size (defined by the length of the semi-axes a and b) are detected in the image and location of an ellipse is defined by its central point. Hence, after the transformation the accumulator shows the most probable locations of ellipse centers. In order to detect ellipses of variable size, lengths of the semi-axes are defined as ranges rather than fixed values ($a_{\min} = \rho_r\, a$, $a_{\max} = \rho_e\, a$, $b_{\min} = \rho_r\, b$, $b_{\max} = \rho_e\, b$, where ρ_r and ρ_e are reduction and expansion factors). It is assumed that every direction can belong to two potential ellipses, so it increases two areas in the accumulator based on predefined templates. Due to optimization reasons, the templates are generated for 256 directions before starting the processing. Examples of such templates used for head detection are presented in Fig. 1. For every direction an appropriate template is selected from the set and added to the accumulator in the direction's location. When the accumulator is generated, ellipse central points are selected as local maxima of the accumulator which exceed a given threshold value.

Fig. 1. Templates for ellipse detection with Hough transform
Rys. 1. Szablony wykorzystywane do detekcji elips za pomocą transformaty Hougha

3. Support Vector Machines for verification

The Support Vector Machines (SVM) [3] is a learning machine which solves two-group classification problems and can be enhanced to multi-class cases as well.

Learning aims at determining an optimal hyperplane which separates a classified, linearly separable training data set: $(l_1,\mathbf{u_1})$, ..., $(l_n,\mathbf{u_n})$, $l_i \in \{-1, 1\}$, where $\mathbf{u_i}$ are vectors in N-dimensional input space and l_i are class labels. The hyperplane defined by $\mathbf{w}_0 \cdot \mathbf{u} + b_0 = 0$ is found by maximizing the margin ρ which separates the classes:

$$\rho(\mathbf{w},b) = \min_{\{u:l=1\}}\left[l \cdot \frac{\mathbf{u}\cdot\mathbf{w}}{\|\mathbf{w}\|}\right] + \min_{\{u:l=-1\}}\left[l \cdot \frac{\mathbf{u}\cdot\mathbf{w}}{\|\mathbf{w}\|}\right], \tag{4}$$

where $\|\cdot\|$ is a vector length. An optimal normal vector $\mathbf{w_0}$ which maximizes the margin can be expressed as a linear combination of vectors from the training set ($\mathbf{w}_0 = \sum_{i=1}^{n} y_i \alpha_i^0 \mathbf{u}_i$), where α_i^0 are non-negative Lagrange multipliers which are obtained during the optimization process. Each vector from the training set is associated with one α and those vectors with non-zero α are termed support vectors and are used for classification.

When the training process is finished, the SVM allows for classification of vector $\mathbf{u}$, based on the calculated decision surface:

$$f(\mathbf{u}) = \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{u},\mathbf{u}_i), \tag{5}$$

where $K(\cdot,\cdot)$ is a kernel function which increases the data dimensionality to a space, in which the data are linearly separable. The most popular kernel functions include: linear, polynomial and radial basis functions (RBF). For the verification the RBF kernel was used:

$$K(\mathbf{u},\mathbf{v}) = \exp(-\|\mathbf{u}-\mathbf{v}\|^2 / \sigma^2), \tag{6}$$

where σ is a width parameter.

For the verification purposes the SVM was trained with two classes of normalized images presenting the positive and negative cases. It was trained independently for every verification task. Normalization includes geometrical transformations, after which the images are of the same size and central points of eyes are located in fixed positions. Also histogram equalization is performed, which is a commonly used technique in face processing area [18]. Examples of normalized face candidates are presented in Fig. 2. Positive cases (upper row) were normalized based on real eye positions pointed by experts and false candidates (bottom row) were extracted based on ellipse detection in complex background.

Fig. 2. Examples of face candidates
Rys. 2. Przykłady obrazów kandydatów na twarze

4. Proposed algorithm for face detection

The proposed algorithm can be divided into three main stages: detection of candidates, verification and precision improvement. At first we perform ellipse detection to find potential areas where human faces may be located. Then, eye sockets are detected within the detected head-ellipses. The eyes are paired to generate face candidates which are verified with the SVM. Finally, position of eyes is verified and corrected with the SVM and iris detection is performed. If it is necessary to detect rotated faces, the directional image is rotated and the detection process is repeated. Outline of the algorithm is presented in Fig. 3.

4.1. Detection of candidates

Ellipse detection which we used in our approach is described in section 2. Before the detection the image is processed with median and gaussian 3x3 filters in order to minimize false edge detection resulting from noise. In our approach we search for ellipses of size 32x40 pixels with reduction and expansion coefficients $\rho_r = 0.8$ and $\rho_e = 1.15$. The directional image is thresholded with $E_{th} = 0.25$ and the associated magnitude image is filtered with Canny edge detector to eliminate redundant directions (the directions with small magnitude are rejected). Depending on minimal and maximal relative size of faces that are to be detected, the image is scaled in a pyramid manner from the smallest to the largest, so that all sizes of ellipses are covered. An example of ellipse detection is presented in Fig. 4 (from left: original and filtered image, directional image before and after Canny edge detector applied in the magnitude image, accumulator and detected ellipses).

Fig. 3. Proposed algorithm for face detection
Rys. 3. Zaproponowany algorytm detekcji twarzy

Fig. 4. Detection of head ellipses
Rys. 4. Detekcja elips głowy

After head detection every ellipse is scaled to a size of 120x150 pixels and its histogram is equalized. Inside the head ellipses the ellipse detection is repeated to find candidates for eye sockets. Eye sockets are characterized by elliptical shapes, but it may be observed that the edges are stronger at the inner sides near the nose. Based on this observation, half-ellipses (right and left) are detected in the image and two accumulators (left and right) are created separately. Ellipse centers are extracted as local maxima in three accumulators: left, right and in the multiplication of both. The detection is performed for half-ellipses of a constant size of 10x5 pixels with $\rho_r = 0.3$ and $\rho_e = 1.3$. Example of eye socket detection with this procedure and generation of a face candidate (described in section 4.2) is presented in Fig. 5.

4.2. Verification of the candidates

Central point of every detected eye-socket ellipse is considered as an eye candidate. After that every pair of eye candidates is regarded as a potential face and it is verified based on simple heuristic rules. Distance between the points must be greater than 25% of the head ellipse width and an angle of a line which crosses these points must be smaller than 30°. Hence, for every head ellipse a set of face candidates (pairs of eye centers) is prepared.

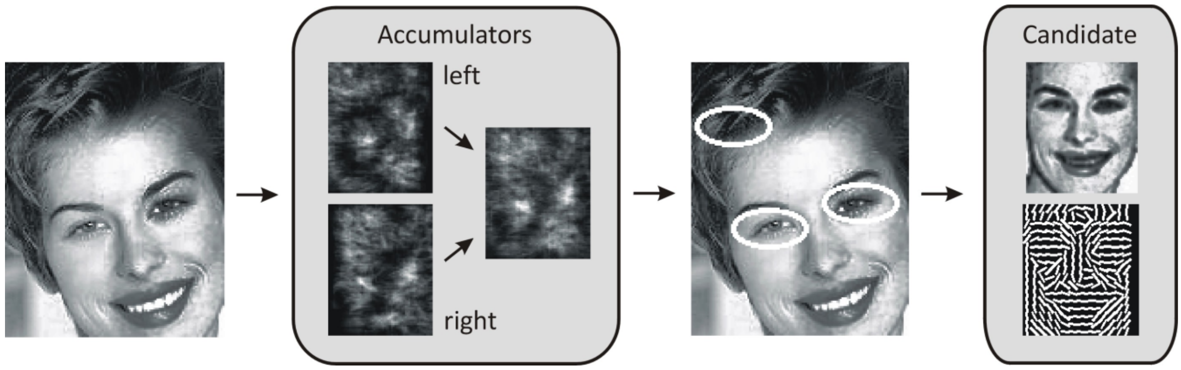

Fig. 5. Detection of eye sockets and generation of face candidates
Rys. 5. Detekcja oczodołów i przygotowanie kandydatów na twarze

Every face candidate is normalized to a size of 16x19 pixels as described in section 3 and two-level verification is performed with the SVM. At the first level the intensity images are verified and in case of a positive response the image is transformed into a directional image as specified in section 2, every direction is represented as a vector in Cartesian coordinate system and such images are verified again with the SVM. The SVM was trained independently for intensity and directional images with a width parameter $\sigma = 0.9$ and $\sigma = 3$, respectively. An example of normalized candidate is presented in Fig. 5.

4.3. Precision improvement

If a face candidate is positively verified, the eye positions are updated iteratively based on the SVM with RBF kernel ($\sigma = 1$) trained with eye and non-eye images normalized to a size of 16x24 pixels. Every eye position is modified in eight directions by a step s_i as presented in Fig. 6 (left), verified and moved to a position, for which the verifier's response is maximal. If a current position gives maximal response, the step is decreased ($s_{i+1} = s_i /2$). The process is started with the step $s_0 = 4$ and repeated until it equals 1.

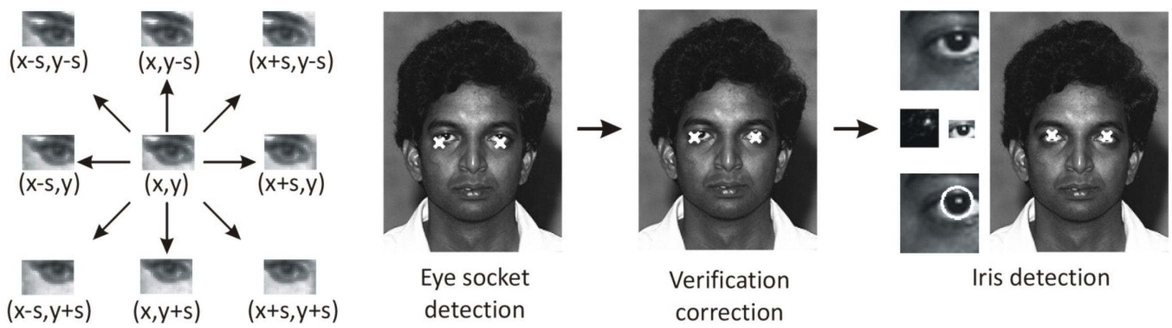

Fig. 6. Eye position update (left) and improvement process (right)
Rys. 6. Modyfikacja położenia oczu (po lewej) oraz proces poprawy precyzji (po prawej)

The second stage of the precision improvement is aimed at iris detection. This is performed by detecting circles with the algorithm described in section 2 ($a = b = 7$, $\rho_r = 0.8$ and $\rho_e = 1.15$) in normalized images of eyes. A square area around an initial eye position of size equal to 0.45 of the distance between the eyes is normalized to 72x72 pixels. Eye position is moved to the maximal value of the accumulator and eye image is verified with the SVM. For a positive response, the eye position is updated. An example of improvement with the iris detection is presented in Fig. 6 (right). The processing steps are presented in a column of small images (from top: clipped right eye at initial position, accumulator, normalized image for verification, detected ellipse).

5. Experimental results

Usually face detection effectiveness is measured with false positive and false negative rates. This is a proper approach as long as the aim is to count faces in images, while the detection precision is irrelevant. However, for face analysis purposes the detection error is propagated further and therefore it is crucial to take the precision into account for algorithm assessment. For every correctly detected face we define the relative precision error as:

$$\delta_d = (\Delta_l + \Delta_r)/(2D), \tag{7}$$

where Δ_l and Δ_r are the distances in pixels between real and detected positions for left and right eye respectively and D is the distance between real positions of eyes.

In our experiments we used face images from Notre-Dame database [12] for training purposes and we validated our algorithm for 3657 images from Feret face image database [13] containing frontal faces. We investigated detection effectiveness given by false positives and false negatives, detection precision and time of execution. Additionally, we measured dependence between face detection precision and face recognition rate with Fisherfaces method based on linear discriminant analysis [2]. We implemented our algorithms in C++ using Adaptive Vision Studio environment (available at www.adaptive-vision.com) and the tests were performed on 2GHz Pentium processor.

For face recognition tests the database is split into a gallery which contains one image per person and a query set which contains images of people whose images are in the gallery. For every image in the query set the most similar image from the gallery is found and the classification is correct if identities of these two images are the same. Effectiveness of face recognition is defined as percentage of correctly classified face images from the query set.

For Feret database 99.42% of faces were correctly detected with one false positive case. Detection precision and corresponding face recognition effectiveness were measured for three cases: based on eye-sockets detection without the precision improvement, after verification maximization and with iris detection. The results are presented in Tab. 1. It may be observed that the detection precision has significant influence on recognition effectiveness. Even a slight precision error of 4% causes a 10-percentage drop in recognition effectiveness.

Table 1

Detection precision test for Feret database

	Eye socket detection	Verification correction	Iris detection	Perfect detection
Relative precision error	7.95 %	5.8 %	4.04 %	-
Recognition effectiveness	64.6 %	70.0 %	73.4 %	82.4 %

Face detection is often used in real-time applications, where time of processing is crucial. The time is strongly dependent on size of faces that are to be detected and the algorithm definitely slows down for small faces (more ellipses are detected and further processed). We measured average times for various relative size of faces and the results are presented in Tab. 2. The relative size is a height of the smallest face which should be detected related to the image height. For the size of 0.2 time of processing is 52 ms, which allows the detector to work at 20 frames per second.

Table 2

Detection time for various sizes of faces (Feret database)

Relative size	0.05	0.1	0.15	0.2	0.25	0.3	0.5
Time [ms]	997	139	58	52	42	32	24

The proposed algorithm works correctly for face images, in which two eyes are visible, even if a face is slightly rotated around vertical axis. Such faces should be selected as face candidates after ellipse detection, but it depends on the verifier whether they will be accepted or rejected. If slightly rotated faces are placed within positive cases in the training set, the verifier will be more tolerant for rotation.

The main drawback of the algorithm is that it is not suitable for detecting small faces. Not only does it consume much more time, but also the effectiveness drops when eye socket ellipses cannot be detected due to their small size. However, face recognition algorithms usually are more robust for faces of higher resolution, so this drawback of the detector does not affect face analysis applications much. Several examples of correct detection for difficult cases are presented in Fig. 7. In the upper row detection result for some face images with glasses, closed eyes and occlusions from AR database [10] are shown. In the bottom row images with rotated and small faces from CMU database [14] are presented.

6. Conclusions and future work

In this paper we introduce a novel algorithm for fast and precise frontal face detection. It utilizes existing methods of head detection based on generalized Hough transform and verification with the SVM. In our approach we use the head detection to narrow down the image area in which we search for eye sockets to find candidates that are further verified. Furthermore, the proposed two-stage method for precision improvement proved to be crucial in face recognition applications. Effectiveness of the proposed approach has been validated with experiments.

Possible directions of future works include decreasing the precision error and conforming the algorithm to detect small faces. The precision may be improved with wavelet analysis of eye area or detection of eyelids. Another problem concerned with detecting small faces is the detection time. However, this may be solved by parallelization of the algorithm in order to utilize multi core processors.

Fig. 7. Successful detection for difficult cases
Rys. 7. Przykłady poprawnej detekcji dla trudnych przypadków

BIBLIOGRAPHY

1. Ballard D. H.: Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition, 1981, Vol. 13(2), p. 111÷122.
2. Belhumeur P. N., Hespanha J. P., Kriegman D. J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, Vol. 19, p. 711÷720.
3. Cortes C., Vapnik V.: Support-Vector Networks. Machine Learning, 1995, Vol. 20(3), p. 273÷297.

4. Donahue M. J., Rokhlin S. I.: On the use of level curves in image analysis. Image Understanding, 1993, Vol. 57, p. 185÷203.
5. Duda R. O., Hart P. E.: Pattern Classification and Scene Analysis. John Wiley, 1973.
6. Han C.-C., Liao H.-Y. M., Yu K.-C., Chen L.-H.: Fast Face Detection via Morphology-Based Pre-Processing, Proc. Ninth Int'l Conf. Image Analysis and Processing, 1998, p. 469÷476.
7. Hsu R. L., Abdel-Mottaleb M., Jain A. K.: Face detection in color images. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, Vol. 24(5), p. 696÷706.
8. Jones M. J., Rehg J. M.: Statistical color models with application to skin detection. International Journal of Comp. Vision, 2002, Vol. 46(1), p. 81÷96.
9. Maio D., Maltoni, D.: Fast face location in complex backgrounds. Face Recognition from Theory to Applications, Nato Asi Series, Springer-Verlag, 1997, p. 568÷577.
10. Martinez A. M., Benavente R.: The AR face database. Technical Report 24, CVC, 1998.
11. Osuna E., Freund R., Girosi, F.: Training Support Vector Machines: An Application to Face Detection. Proc. IEEE Conf. Comp. Vision and Pattern Recogn., 1997, p. 130÷136.
12. Phillips P. J., Flynn P. J., Scruggs T., Bowyer K. W., Chang J., Hoffman K., Marques J., Min J., Worek W.: Overview of the Face Recognition Grand Challenge. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2005, Vol. 1, p. 947÷954.
13. Phillips P. J., Wechsler H., Huang J., Rauss P.: The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing J., 1998, Vol. 16(5), p. 295÷306.
14. Rowley H., Baluja S., Kanade T.: Neural Network-Based Face Detection. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1996, p. 203÷208.
15. Turk M., Pentland A.: Face Recognition Using Eigenfaces. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1991, p. 586÷591.
16. Yang G., Huang T. S.: Human Face Detection in Complex Background. Pattern Recognition, 1994, Vol. 27(1), p. 53÷63.
17. Yang M.-H., Kriegman D. J., Ahuja N.: Detecting faces in images: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, Vol. 24, p. 34÷58.
18. Zhao W., Chellappa R., Phillips P. J., Rosenfeld A.: Face recognition: A literature survey. Tech. Report CARTR-948, Center for Automation Research, University of Maryland, College Park, 2000.

Recenzent: Prof dr hab. inż. Ewa Piętka

Wpłynęło do Redakcji 31 stycznia 2009 r.

Omówienie

Zadanie detekcji twarzy sprowadza się do podjęcia decyzji, czy w zadanym obrazie występują ludzkie twarze oraz wskazania ich dokładnego położenia w przypadku odpowiedzi pozytywnej. W artykule został zaprezentowany algorytm detekcji twarzy frontalnych w kanale luminancji obrazów cyfrowych. W analizowanym przypadku położenie twarzy definiowane jest jednoznacznie przez położenie środków oczu. Zaproponowany algorytm składa się z trzech podstawowych etapów: wyłonienia zbioru kandydatów na twarze, weryfikacji kandydatów oraz operacji zwiększających precyzję detekcji.

Pierwszym krokiem algorytmu jest detekcja elips głowy za pomocą uogólnionej transformaty Hougha. Następnie wewnątrz każdej elipsy głowy poszukiwane są elipsy reprezentujące oczodoły, które są łączone w pary w celu wyłonienia zbioru kandydatów. Kandydaci są normalizowani i poddawani dwustopniowej weryfikacji za pomocą odpowiednio wytrenowanego Mechanizmu Wektorów Wspierających. Jeżeli odpowiedź weryfikatora jest pozytywna, to kandydat uznawany jest za twarz.

Ostatnim krokiem jest zwiększenie precyzji detekcji już wykrytych twarzy, przeprowadzane dwustopniowo. Najpierw iteracyjnie modyfikowane jest położenie każdego oka w celu maksymalizacji odpowiedzi klasyfikatora obrazu oka, a następnie przeprowadzana jest detekcja tęczówki, również za pomocą transformaty Hougha.

W ramach prac przeprowadzone zostały badania eksperymentalne, które potwierdziły wysoką skuteczność zaproponowanych rozwiązań. Zbadany został także wpływ precyzji detekcji na skuteczność rozpoznawania twarzy wykonywanego za pomocą metody Fisherfaces. Otrzymane wyniki pozwoliły na wyciągnięcie wniosku, że precyzja detekcji jest istotnym czynnikiem wpływającym na skuteczność rozpoznawania twarzy, a niedokładność detekcji obniża szanse na poprawne rozpoznanie twarzy.

Wśród możliwych kierunków przyszłych badań należy wymienić opracowanie dodatkowych mechanizmów zwiększających precyzję detekcji, na przykład przez wykrycie powiek lub z wykorzystaniem analizy falkowej. Badania mogą zostać ponadto pokierowane w stronę zwiększenia skuteczności funkcjonowania algorytmu dla niewielkich i niewyraźnie widocznych twarzy.

Addresses

Michał KAWULOK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, mkawulok@polsl.pl.

Janusz SZYMANEK: Future Processing, ul. Rokitnicka 67, 41-936 Bytom, Polska, jszymanek@future-processing.com.