An Approach to Solving Face Detection Task

V.V. Krasnoproshin
Belarusian State University, Minsk, Belarus,
krasnoproshin@bsu.by

E.V. Koblov
Belarusian State University, Minsk, Belarus,
kablov@bsu.by

Abstract: *In this paper a pattern-recognition based approach to solving face detection task is proposed. All main steps are revealed and algorithms for their solution are built. It is shown that these algorithms in the aggregate effectively solve the face detection task. The conditions when the exact solution can be obtained are determined.*

Keywords**:** *face detection, ellipse search, image fragmentation, pattern recognition*.

I. INTRODUCTION

Currently so-called biometrical systems of human identification became popular [1]. Such systems use unique biological characteristics which unambiguously define a person. The face is considered to be one of such characteristics.

An important task during implementation of technology of biometrical identification is detection of faces on an arbitrary input image [2]. This task is currently far from its solution [3]-[6]. Methods used for its solution usually have a universal nature, very high computational complexity (about O (n^3), where n – number of pixels in the input image) and are not invariant to shooting condition.

In this paper an efficient (with O(n) complexity) approach to finding face-like areas is proposed Also the conditions on which it's possible to find exact solution of face detection task are determined.

II. STATEMENT OF PROBLEM OF FACE DETECTION

Let's define the face detection problem in general. An image of an arbitrary scene is given where different objects are present (including people). It's necessary to detect only those fragments of image which correspond to human faces.

Let's consider an approach to solving this problem based on its reduction to the pattern recognition problem. The letter can be defined as follows [7]:

A set of permissible object is given which is divided into ℓ non-overlapping classes. The division is not completely defined and is given either by enumeration of some objects (by precedents) or by some properties of class (by rules). It's necessary to determine (recognize) to which class arbitrary permissible object belongs, given the conditions.

Thus, to define and solve a face detection task in the terms of pattern recognition it's necessary to define an appropriate set of permissible objects, divide this set into classes and construct a recognition algorithm.

Let's suppose that the face detection task is defined in the terms of pattern recognition and the corresponding recognition algorithm has been constructed. The process of face detection on an arbitrary image requires performing the following steps:

- Division of image into fragments
- Description of fragments in the terms of permissible objects
- Recognition of belonging of permissible objects to classes.

It's obvious that to each of these steps a separate subtask is corresponding. The main of them is the recognition subtask, however, the efficiency of solution of face detection task depends on solution of each of them. It's necessary to construct such algorithms for solving each of the enumerated subtasks which in the aggregate would solve the original face detection problem. Let's consider each of them in detail.

III. RECOGNITION PROBLEM

When constructing recognition algorithm we'll use a well-known hypothesis that the face has an elliptic shape which semi-axes are in the ratio of 75%. Then it's possible to use ellipse as a geometric primitive which mostly corresponds to permissible objects. In this case an object can be described in terms of ellipse parameters i.e. permissible object $F = (x_0, y_0, a, b, \theta)$ where (x_0, y_0) – ellipse center, θ - rotation angle and a, b – big and small semi-axes.

A set of all permissible objects can be divided into two classes: all objects which satisfy the condition $\left|\frac{b}{a} - 0{,}75\right| \le \varepsilon$, where ε is some threshold value we'll refer to class K1, and all others to class K2.

Now, let's construct an algorithm which determines belonging of an arbitrary permissible object F to one of the given classes.

Let's introduce two predicates:

$$P_1(F) = \begin{cases} 1, & if\ |b/a - 0{,}75| \le \varepsilon, \\ 0, & otherwise \end{cases}$$

$$P_2(F) = \begin{cases} 1, & if\ |b/a - 0{,}75| > \varepsilon, \\ 0, & otherwise \end{cases}$$

Then recognition algorithm can be written down as $A = R(R_1, R_2) \circ r$, where $R_1=<F, P_1>$, $R_2=<F, P_2>$ - recognition operators, $r = max(R_1, R_2)$ – decision rule.

It's easy to notice that complexity of the constructed algorithm is equal to the complexity of the calculation of predicates which comes to O(1) and accuracy of solution depends on the chosen hypothesis and threshold value (ε).

Describing the algorithm, it's also possible to note that it's invariant to the geometric transforms of scale and rotation.

IV. DESCRIPTION PROBLEM

To describe image fragments in terms of permissible objects it's necessary to solve two tasks:

- Approximate fragment with a geometric primitive (in this case ellipse);
- Check the permissibility of the constructed approximation (degree of correspondence of fragment to shape of constructed ellipse).

When solving the first task we'll use the following considerations. It's well-known that ellipse center (x_0, y_0) coincides with its center of mass, rotation angle θ is determined by ellipse orientation and semi-axes a and b are half of height width of ellipse boundary rectangle. Since a human face has an elliptic shape, these properties can be used for calculation of parameters of approximating ellipse.

Algorithm 1 (of approximation):

Step 1. *(ellipse center (x_0, y_0))*

Let an image fragment U_i be given by its characteristic function $c_i(x,y) = \begin{cases} 1, (x,y) \in U_i \\ 0, (x,y) \notin U_i \end{cases}$. Center of mass by definition can be obtained using first moments:

$$x_i = \frac{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} k c_i(k,l)}{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} c_i(k,l)} \qquad y_i = \frac{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} l c_i(k,l)}{\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} c_i(k,l)}$$

Step 2. *(ellipse rotation angle θ)*

To find θ we'll use second moments [8]:

$$\alpha = \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (k - x_i)^2 c_i(k,l)$$

$$\beta = 2\sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (k - x_i)(l - y_i) c_i(k,l)$$

$$\gamma = \sum_{k=0}^{m-1}\sum_{l=0}^{n-1} (l - y_i)^2 c_i(k,l)$$

Rotation angle can be obtained to ($\pi/2$) using expression $tg 2\theta = \beta/(\alpha - \gamma)$, when $\beta \neq 0$ or $\alpha \neq \gamma$ as follows:

$$\theta = \frac{\arctan(\beta/(\alpha - \gamma))}{2}$$

Step 3. *(Calculation of ellipse semi-axes)*

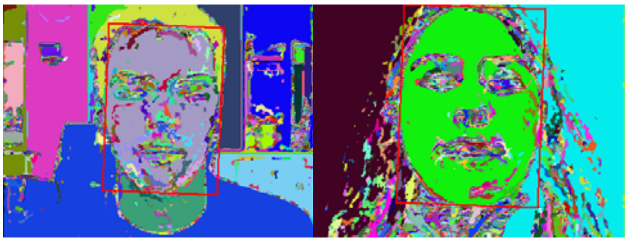

Fig. 1. Calculation of boundary fragment of a face-like fragment.

To find ellipse semi-axes let's put down a fragment into a rectangle considering already known center and orientation of the fragment. Co-ordinates of pixels of the fragment aligned to ordinate axis can be obtained using formulas:

$$\begin{cases} x^* = x_0 + (x - x_0)\cos\theta - (y - y_0)\sin\theta \\ y^* = y_0 + (x - x_0)\sin\theta + (y - y_0)\cos\theta \end{cases}$$

Then coordinates of diagonal points of the rectangle can be found as follows:

$$left = \min_{c_i(x,y)\neq 0}(x^*) \qquad top = \min_{c_i(x,y)\neq 0}(y^*)$$

$$right = \max_{c_i(x,y)\neq 0}(x^*) \qquad bottom = \max_{c_i(x,y)\neq 0}(y^*)$$

Step 4. *(a more accurate definition of a rotation angle)*

Considering semi-axes found we can correct value of the rotation angle:

$$\theta = \begin{cases} \theta, \ if \ \ 2a = right - left \\ \theta + \dfrac{\pi}{2}, \ otherwise \end{cases} .$$

Fig. 2. Ellipse approximations of face-like fragments.

It's easy to see that the algorithm finds an approximating ellipse for any fragment. At the same time, the accuracy of approximation essentially depends on the shape of the fragment: the closer fragment to ellipse the better approximation is. These properties of the algorithm relies, on the one hand, regulated degree of feasibility of hypothesis that the face has an elliptic shape and on the other hand pick out permissible objects. To do this it's enough to estimate accuracy of the constructed approximation.

Fig. 3. Ratio of approximating ellipse and real fragment.

Let's designate number of fragment pixels inside and outside ellipse as *Pin* and *Pout* correspondingly, and general number of pixels in ellipse as *C*.

To check the accuracy of approximation let's introduce the following rule:

$$Pin / C > T_1 \text{ and } Pout / C < T_2,$$

where T_1 and T_2 are some threshold values.

This algorithm of description of permissible objects does not make any limitations to the source image and always constructs a description of permissible objects. It's also easy to prove that it has a linear complexity.

V. FRAGMENTATION PROBLEM

It's necessary to develop an algorithm which divides a source image into fragments and obtain conditions on which algorithm is working correctly (i.e. areas with faces correspond to different fragments).

To describe an algorithm let's introduce necessary notions and definitions. Let I be an original image. Pixels (i_1, j_1) and (i_2, j_2) we'll call *neighboring* if the following condition is met:

$$\max(|i_1 - i_2|, |j_1 - j_2|) = 1,$$

Two neighboring pixels (i_1, j_1) и (i_2, j_2) we'll call *close*, if inequality is true:

$$\|I(i_1, j_1) - I(i_2, j_2)\| < \nu,$$

where ν is a threshold value which defines image detailed elaboration.

When fragmenting the image we'll require that only close pixels composed a fragment. The main idea of the algorithm is based on a marking procedure, which assigns an identifier to each pixel of the image. At that, pixels belonging to one fragment should have the same identifier. As a result of the algorithm a matrix of identifiers (U_{mn}) which represents all image fragments is constructed

Now let's describe fragmentation algorithm. Let L be some queue. The image is scanned and all non-marked pixels are added to L. After adding a regular pixel, its closed pixels are discovered and added to L. At this original image is extracted from L. This procedure is repeated until L is empty. Adding a pixel *(i, j)* to L we'll designate as $(i, j) \to L$, and extraction - $(i, j) \leftarrow L$. Then algorithm can be written down as follows:

Algorithm 2: (of image fragmentation)

Step 1. *(primary step)*

$U_{mn} = \{0\}, L = \{\varnothing\}$ *id=1*

Step 2. *(common step)*

The image is searched from left to right from top to bottom ($i = \overline{0, m-1}$, $j = \overline{0, n-1}$) for a regular non-marked pixel (i, j) ($U(i, j) = 0$). If there are no non-marked pixels go to step 7.

Step 3.

$L = \{(i, j)\}$, $U(i, j) = id$, $id = id + 1$.

Step 4.

While $L \neq \{\varnothing\}$ do step 5. Otherwise go to step 2.

Step 5.

$(i, j) \leftarrow L$ - extract regular pixel from the queue and compare it to the neighboring non-marked pixels.

Step 6.

Let (i_k, j_k) be k-th neighboring pixel ($k = \overline{1, c}$, where c – number of neighbors). If $\|I(i, j) - I(i_k, j_k)\| < \nu$, then $U(i_k, j_k) = U(i.j)$, $(i_k, j_k) \to L$

Step 7.

End of algorithm.

As a result of the algorithm matrix *U* which contains all image fragments will be constructed.

Fig. 4. Example of fragmentation algorithm working.

It is proved that for correct working of fragmentation algorithm it's enough that image meets the following conditions:

1. $\forall(i_1, j_1) \in A, (i_2, j_2) \notin A$, such that

$$\max(|i_1 - i_2|, |j_1 - j_2|) \le 1 \Rightarrow \|I(i_1, j_1) - I(i_2, j_2)\| \ge \nu$$

2. $\forall(i_1, j_1), (i_2, j_2) \in A$, such that

$$\max(|i_1 - i_2|, |j_1 - j_2|) \le 1 \Rightarrow \|I(i_1, j_1) - I(i_2, j_2)\| < \nu,$$

As for algorithm complexity, it comes to O (N), where N is the number of pixels in image.

VI. CONCLUSION

A described technology has a linear complexity which allows its usage in real time systems. Moreover, it is invariant to scale and rotation of a face and can be used for images with arbitrary background.

Decomposition into elemental subtasks allowed to obtain conditions of algorithm application and to manage its accuracy and complexity. However, critical moment of the technology is a fragmentation task. In the case when it's impossible to isolate an elliptic fragment of a face (for instance when face is partially visible, the face is not in frontal position, etc.) additional research is necessary which eliminates this limitation. This can be achieved for instance by skin segmentation, directional filtration, borders accentuation, severance of fragment bottlenecks, etc. All this allows to meet conditions of divisibility which allow usage of fragmentation algorithm.

REFERENCES

1] Pankanti S., Bolle R.M., Jane E. *Biometry: Future of identification* // Open systems, №03, 2000.

[2] Rowley H. A., Baluja S. and Kanade T. *Neural Network-Based Face Detection* // IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 20, 2001.

[3] Brilyuk D., Starovoitov V. *Human face recognition and neuronetworking methods*. Institution of technical cybernetics of National Academy of Sciences of Belarus, Minsk, 2001.

[4] Vezhnevets V. *"Method For Localization Of Human Faces In Color-Based Face Detectors And Trackers"* // Proceedings of the Third International Conference on Digital Information Processing And Control In Extreme Situations, Minsk, 2002, pp. 293.

[5] Zorin S.A., Matveev I.A., Murynin A.B., Senkov R.V, Tsurkov V.I. *Face relief reconstruction in task of automatic identification of personality.* Russian Academy of Sciences, Moscow, 1997.

[6] Yang, M.-H., Kriegman, D. J., Ahuja, N. *Detecting Faces in Images: A Survey*. IEEE Transitions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 1, 2002. pp. 34-56.

[7] Aderikho K., Kiselevsky L., Kostyukevich S., Krasnoproshin V. *Physical foundations of remote sensing.* Minsk: Universitetskoe, 1991, pp. 293.

[8] B.K.P. Horn. Robots vision. – M.: Mir, 1989

[9] Krasnoproshin V., Koblov E., *Face-like ellipse search based on texture segmentation*. // Proceedings of 8th International conference on Pattern Recognition and Information Processing, Minsk, 2005, pp. 390-393.