Face Recognition by Face Bunch Graph Method

JIRI STASTNY*, VLADISLAV SKORPIL**
* Department of Automation and Computer Science,
** Department of Telekommunications,
Brno University of Technology,
Purkynova 118, 612 00 Brno,
CZECH REPUBLIC,

Abstract: - Face Bunch Graph method method uses a simple comparison function both for the localization and the recognition of faces. The input data for the two processes are so-called Jets, which represent image properties in the neighbourhood of a face bunch graph (FBG) node. The algorithm described below was chosen for implementation because of its very good results and because of the application of the same representation for both searching for and comparing images.

Key-Words: - Face Bunch Graph, recognition, Jets, algorithm, Gabor wavelet, similarity function

1 Introduction

The problem the algorithm is to solve is searching for faces in individual examined pictures and assigning them to a specific face included in a database, in spite of the differences in the face expression, head position, the position and size of images. Further, it is necessary to distinguish an object that momentarily looks different from what is stored in the database. Here is the need to suppress differences and emphasize the values being distinguished. This is generally possible only on the assumption that some auxiliary information about the structure of the sought object and about its expected changes is available.

The system being described has an important backbone structure (so-called face bunch graph), which reflects the fact that images of coherent objects have a tendency to be inverted, change their size, rotate and be deformed in the image plane. The basic object is represented by a labeled graph – the edges carry information about the distance and the nodes are assessed by wavelet coefficients grouped into Jets. The stored model graphs can be applied to new images with the aim of generating a graph for a new picture, which can then be incorporated in the gallery and become a model graph. The wavelets used are resistant to changes in illumination, small shifts and deformations. Model graphs can be shifted a little, and their size, orientation and deformation can be changed during the comparison process. This will simultaneously offset a considerable part of the differences between the pictures being compared.

2 Problem Formulation

The representation of local properties is based on the Gabor wavelet transform, see [1]. The Gabor wavelets are biologically motivated convolutions of cores in the form of a planar wavelet limited by the Gauss envelope function. A set of convolution coefficients for cores with different orientations and frequencies in one picture pixel is referred to as the Jet.

The Jet describes a small area of information in image $I(\vec{x})$ in the neighbourhood of a given pixel $\vec{x} = (x, y)$. The calculation is based on the wavelet transform defined as the convolution:

$$J_j(\vec{x}) = \int I(\vec{x'}) \psi_j(\vec{x} - \vec{x'}) d^2\vec{x'} \qquad (1)$$

with a family of Gabor cores:

$$\psi_j(\vec{x'}) = \frac{k_j^2}{\sigma^2} e^{\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)} \left[e^{\left(i\vec{k_j}\vec{x}\right)} - e^{\left(-\frac{\sigma^2}{2}\right)} \right] \qquad (2)$$

in the form of planar waves with wave vector k, limited by the Gauss envelope function. The calculation is performed for a discrete set of 5 different frequencies $v = 0, ...,4$ and 8 different orientations $\mu = 0,...,7$

$$\vec{k_j} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\varphi_\mu \\ k_v \sin\varphi_\mu \end{pmatrix}, \quad k_v = 2^{-\frac{v+2}{2}}\pi, \quad \varphi_\mu = \mu\frac{\pi}{8} \qquad (3)$$

with subscript $j = \mu + 8v$. This sampling covers uniformly the frequency space being examined. The width σ/k of the Gauss curve is controlled by parameter $\sigma = 2\pi$.

2.1 Face bunch graphs

A set of initial points is defined for faces, e.g. the pupil, mouth edges, nose tip, upper and lower ear rims. Labelled graph G represents a face consisting of N nodes in these initial points in positions $\overrightarrow{x_n}, n = 1,...,N$, and E edges between them. The nodes are labelled with Jets J_n. The edges are labelled with distance vectors $\overrightarrow{\Delta x_e} = \overrightarrow{x_n} - \overrightarrow{x_{n'}}, e = 1,...,E$, where edge e links node n' with n. The edges are therefore labelled with two-dimensional vectors. This face graph is adapted to the object because the nodes are selected from specific points of the face, the *initial points.*

In order to find initial points in the new pictures of faces it is necessary to have a general face representation over individual face models. This representation should capture a wide spectrum of possible variations in the look of the face, e.g. differently set eyes, mouth, nose, different shapes of chin, differences given by sex, age, race, etc. It is evident that it would be too demanding to capture every property by a single graph. Instead, we combine sets of individual graphs into a stack-like structure called "face bunch graph" (FBG), see Fig. 1. Each model has the same mesh structure and the nodes refer to identical initial points. A set of Jets referring to one initial point is called the bunch. An eye bunch, for example, may contain the Jets from closed, female and male eyes, etc. in order to cover these local diversities. In the course of localizing initial points in a face that have not yet been recorded the procedure described in (2.1) selects the best corresponding Jet (called local expert). The latter is selected from the bunch for every single initial point. Each bunch graph has at its disposal a combination of Jets and each bunch graph thus covers considerably more diversities in a face than captured in the set-up of individual graphs as such.

The face bunch graph serves as the representation of common face. It is designed to incorporate all the possible variants in the look of a face. A FBG combines information from different face graphs. Its nodes are labelled with Jet sets called bunches, and its edges are labelled with average distance vectors. During comparison to an image, the best fitting Jet in each bunch, indicated by grey shading, is selected independently.
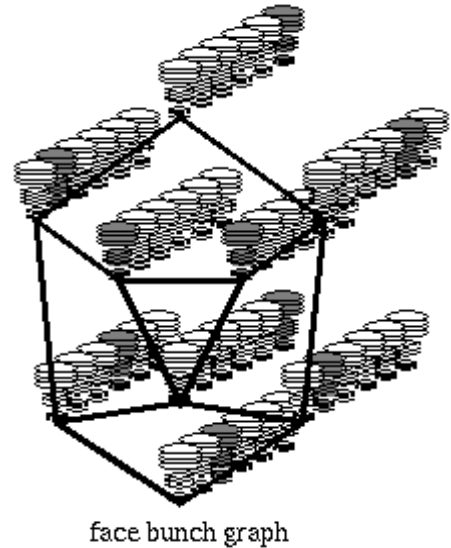

face bunch graph

Fig. 1

So far the representation of individual faces and basic data about faces has been described by face bunch graphs, or FBG. Now a description of how to generate these graphs will be given. The simplest method is to produce them manually. This method is used to generate the basic graphs for the system. If the system contains FBG (also in the case that it contains a single manually defined model), the graphs for new images can be generated automatically. In the beginning, when the FBG contains only a few faces, it is necessary to check and correct the resulting comparisons. If, however, the FBG has been filled sufficiently (ca. 70 graphs), the comparison can be relied on and extensive galleries of model graphs can be generated automatically.

Manual graph is generated in three steps. In the first step a set of initial points from a given image is designated. Most of them are in easy-to-determine locations, where they can be measured simply, such as the right and left pupil, mouth corners, nose tip, upper and lower ear rims, crown of the head, tip of the chin. These points have been selected to enable easy and precise manual localization. Other initial points are located around centres of importance, where they can be determined precisely and simply. This allows the automatic selection of initial points in areas where precisely definable properties and shapes are absent, e.g. the cheeks or forehead. In the next step the between initial points are depicted and the evaluation of edges is automatically calculated as a difference in the distances between nodes. Finally, using the Gabor wavelet transform (1), Jets for the respective nodes are determined.

In short, the set of initial points should cover the whole face uniformly. However, it depends on the concrete case, when it may be necessary to duly emphasize

certain areas via additional nodes. For example, when searching for faces we place more nodes on the silhouette because with a homogeneous background the face contour is a good clue for searching. For face recognition, on the other hand, we place more nodes in the inner part of the face since this is important for recognition. Using more nodes has a tendency to achieve better results since a larger amount of information is used. But the resulting effect is suppressed if the nodes are too close and the corresponding Gabor coefficients become very similar because of overlapping nuclei. However, the computation demands increase linearly with increasing number of nodes. The optimum number of nodes is then a compromise between the recognition speed and power.

In the image graph method the function that calculates the similarity between the image graph and FBG in the corresponding initial point plays a key role. Important are the similarity of Jets and the curvature of mesh graphs with respect to the mesh in the FBG. For image graph G^I with nodes n = 1,...,N, with edges e = 1,...,E and for FBG with model graphs m = 1,...,M, the similarity is defined as:

$$S_B\left(G^I, B\right) = \frac{1}{N} \sum_n \max_m \left(S\phi\left(J_n^I, J_n^{Bm}\right)\right) - \frac{\lambda}{E} \sum_e \frac{\left(\overrightarrow{\Delta x_e^I} - \overrightarrow{\Delta x_e^B}\right)^2}{\left(\overrightarrow{\Delta x_e^B}\right)^2} \qquad (4)$$

where λ gives the relative importance of the Jet and the metric structure. J_n are Jets in nodes n, and Δx_e is the distance vector used as label v of edge e. FBG offers several Jets for each initial point and the best Jet is chosen and used in the comparison.

2.2 Searching procedure

The aim of the image graph method is to find in the pictures tested the initial face points and to extract them into an image graph that maximizes the agreement with FBG as defined in (4). In practice it is necessary to apply a heuristic algorithm so that the calculation may approximate optimum in reasonable time. In the beginning, a rough estimate is used, which is made more specific using the FBG degrees of freedom: shift, change in scale, ratio of edge and local deformation. Information about the phase is used similarly, and the focus degree of expected shift is increased: no phase, phase with focus 1, then phase with focus 1 up to 5.

Step 1 - Searching for an approximate face position:
For this step, so-called "average graph" is calculated as the arithmetic mean of values over all Jets of the corresponding initial point of FBG. An image graph is made with the same mesh as given by FBG and the graph is shifted along the examined picture in steps of 4 pixels. The results of this phase are coordinates, which maximize function S_B (4). The procedure is repeated in the vicinity of coordinates with maximum value of S_B in steps of 1 pixel. The coordinates found in this way serve as the initial point for step 2.

Step 2 - Exact determination of position and size:
The FBG is now used without averaging. The calculation proceeds for 4 different positions ($\pm3, \pm3$) pixels around the position found in Step 1. In each individual position the graph is examined increased or reduced by a factor of 1.18 compared with the FBG size. This has no effect on the shape of graph mesh since all vectors x, e and b are transformed identically (for $\lambda = \infty$). For each of the 8 variations the best fitting Jet is selected for each node and the calculation of its shift, corresponding to (5), is completed. The shift estimate is obtained by maximizing similarity S_Φ in its Taylor series:

$$S_\phi\left(J, J'\right) \approx \frac{\sum_j a_j a'_j \left[1 - 0.5\left(\phi_j - \phi'_j - \overrightarrow{d} \overrightarrow{k_j}\right)^2\right]}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}} \qquad (5)$$

This is done with the focus value equal to 1, which means that the shift values may be as much as 8 pixels. The meshes are then re-calculated and shifted so as to minimize the power of the sum of all shifts. The best points from all 8 variations serve as the initial points for the next step.

Step3 - Exact determination of the size and ratio of sides:
The same process is applied as in Step 2 but different positions are examined separately for axis x and axis y. The focus value is gradually increased from 1 to 5.

Step 4 - Local deformation:
Local deformation: In pseudorandom sequences the position of each node is changed such that it approximates similarity with FBG. The metric similarity is now taken into consideration by setting ($\lambda = 2$) and using the vectors x_e obtained in Step 3. In this step, only those positions are compared for which the estimated shift vector is d<1 (see expression (5)). For these local deformations, the focus is again increased from 1 to 5.

The resulting graph is called the image graph and it is stored as a model of each face from the image. At this moment the face position in the picture under examination is known. This procedure assumes that the face exists in the examined picture, roughly in the expected position. For other different face positions (half-profile, profile) the algorithm can be applied with an appropriate change in the graph shape. The possibility of establishing the existence or non-existence of face in the picture examined can be provided for by the threshold value of similarity function S_B (4).

2.3 Recognition

At this moment the position of face is known (the coordinates of initial points in the picture). The face can now be transformed into standard dimensions and overlapped with a graph of denser mesh (a greater number of initial points in the face part). This newly created graph is again labelled with Jets and can be stored in the database. After creating model graphs from the image gallery, and image graphs from the pictures examined, recognition with relatively few demands on computation time is possible by merely comparing the image graph with all model graphs and selecting the one with greatest degree of similarity. Comparing with a gallery of 250 pictures (model graphs stored in the database) takes less than a second on a current computer. The similarity function used to compare the graphs is an average of the values of Jet-couple agreement. If G^I is an image graph, then G^M is a model graph and node n_n in the model graph corresponds to node n' in the image graph. Graph similarity is then defined as:

$$S_G\left(G^I, G^M\right) = \frac{1}{N'}\sum_{n'} S_a\left(J_{n'}^I, J_{n_n'}^M\right) \tag{3}$$

The Jet similarity function has been used without phase. Experiments have shown that it has a greater resolving power because it has a higher resistance to changes in the face expression.

3 Problem Solution

As part of this work, the USC algorithm (Face recognition by the bunch graph method) was implemented. A library of classes was created together with an application enabling the testing of results obtained for the partially automatic variant of the logarithm. The implementation was conducted in the Python language (www.python.org) using the wxPython library (www.wxpython.org) and the PIL (Python Imaging Library – www.pythonware.com). The transferability of the application and library to an arbitrary computation platform without any need for recompiling is thus guaranteed.

Fig. 2

The main window of the application (Fig. 2) is divided into 2 parts. The left part displays the picture being examined while the right part displays the content of database (gallery) and the comparison result.

3.1 Evaluation

The test was carried out in two phases. In the first phase the effect of the number of pictures on the modification of the mesh shape in the bunch graph was investigated. Altogether, 26 and 50 model graphs included in the bunch graph were tested. The graphs had been calculated from pictures that had not been used for recognition (see Table 1).

	26 model graphs	50 model graphs
Average value of similarity function S_B	0,556603	0,562189
Number of manual corrections	77	80

Table 1

In the second phase the effect of mesh density on correct recognition of the face picture was tested. This test concerned 100 examined pictures and 100 pictures held in the gallery. The gallery contained 100 pictures from the preceding evaluation. The test pictures were compared with the gallery pictures (see Table 2).

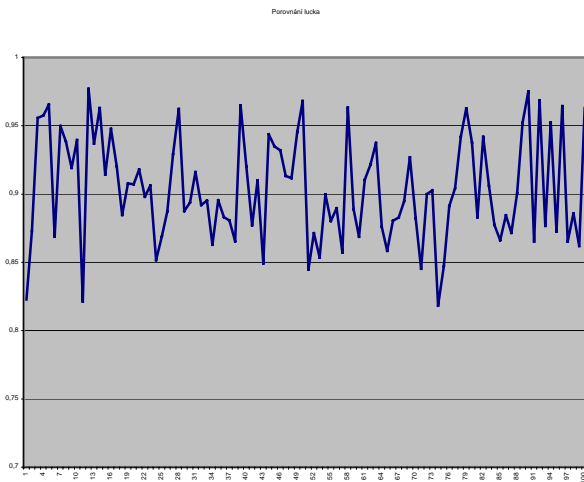|	26 model graphs/ 9 mesh nodes	50 model graphs/ 17 mesh nodes
Average value of similarity function S_B	0,904098	0,895163
Number of correctly determined pictures	73	86

Table 2

Porovnání lucka

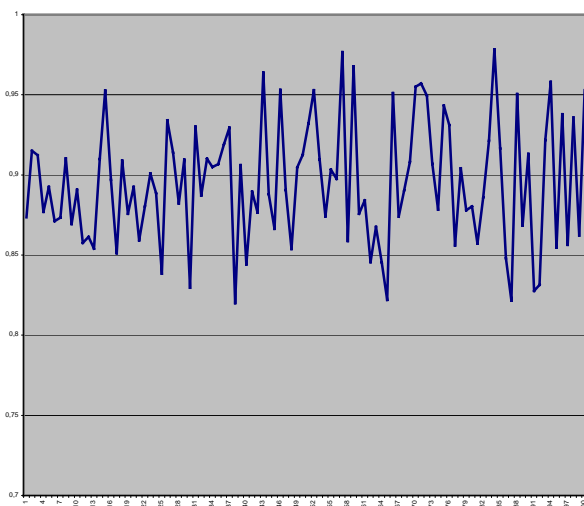Fig. 3 Waveform of agreement function value for 9 graph nodes

Fig. 4 Waveform of agreement function value for 17 graph nodes

4 Conclusion

The algorithm has shown relatively good results with respect to the number of graph mesh nodes but the results obtained in deforming the graph with respect to the face in the picture do not allow immediate practical application. A considerable improvement can be expected if the number of graph mesh nodes is increased. The evaluation of the comparison does not yield the results given in [1] but the success rate increases in dependence on the number of mesh nodes. Although the computation of the Jets properties is demanding as regards processor time, the search and comparison processes are elegant and fast when compared with other methods. It is recommended that further work should concentrate on the description of graph shape and its density.

References:
[1] P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, *The FERET Evalution Methodology for Face-Recognition Algorithms,* Technical report NISTIR 6264, 1999
[2] B. Maghaddam, C. Nastar, A. Pentland, Bayesian Face Recognition using Deformable Intensity Surfaces, *M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 371,* 1996
[3] S. A. Rizvi, P. J. Phillips, H. Moon, *The FERET Verification Testing Protocol for Face Recognition Algorithms,* Technical report NISTIR 6264,. 1998
[4] L. Wiskott, J. M. Fellows, N. Krüger, *Face recognition by elastic bunch graph matching.* Inteligent Biometric Techniques in Fingerprint and Face Recognition, ISBN 0-8493-2055-0, 1999, pp. 355-396.
[5] R. Polikar, The Wavelet Tutorial. *http://engineering.rowan.edu/~polikar/wavelet.html*