Inexact Graph Matching using Stochastic Optimization Techniques for Facial Feature Recognition

Roberto Cesar
U. São Paulo
IME-USP, Brazil
cesar@ime.usp.br

Endika Bengoetxea
U. Basque Country,
Dept ATC/KAT,
San Sebastian, Spain
endika@si.ehu.es

Isabelle Bloch
ENST, Dept TSI,
CNRS URA 820, Paris, France
Isabelle.Bloch@enst.fr

Abstract

We propose a formalization of model-based facial feature recognition as an inexact graph matching problem, one graph representing a model of a face and the other an image where recognition has to be performed. The graphs are built from regions and relationships between regions. Both nodes and edges are attributed. A global dissimilarity function is defined based on comparison of attributes of the two graphs, and accounting for the fact that several image regions can correspond to the same model region. This function is then minimized using several stochastic algorithms.

1. Introduction

Recognition of facial features such as eyes, lips, etc. is important in several problems related to face and facial expression recognition [6]. In this paper we express the problem of recognition of facial features as an inexact graph matching problem [7]. One graph represents a model of facial features and their characteristics, while a second graph is computed from an image to be recognized based on an over-segmentation. Due to this over-segmentation, several regions can be assigned to the same feature, i.e. to the same model region. Therefore classical methods for graph or subgraph isomorphisms are not suitable and the problem belongs to the class of inexact graph matching, which is known to be NP-complete [3].

An initial solution for this problem has been proposed using a deterministic tree search algorithm. In this paper we propose stochastic algorithms and show that the results are largely improved based on preliminary experiments. The construction of the graphs for the specific application of facial feature recognition is described in Section 2. Then we express the inexact graph matching problem in Section 3. The proposed algorithms, based on stochastic tree search, genetic algorithms (GAs) and estimation of distribution algorithms (EDAs), are presented in Section 4, and preliminary results are discussed in Section 5.

2. Graph construction

The image where recognition has to be performed is represented as a graph G_D (the data graph), based on an over-segmentation. In our experiments, segmentation was performed using a watershed algorithm, but any other method with similar properties could be used as well. Each region in the segmented image corresponds to a node of the graph. Edges represent spatial relations between any two regions.

The model is represented as a graph G_M (the model graph), where each node corresponds exactly to one facial feature of interest. Edges are defined as for G_D. In our experiments, the model was built manually from a face image. It can be for instance built from one image of a video sequence, and used as a model for all other images of the sequence, or even of another sequence. A tracking procedure, based on Gabor Wavelet Networks (GWN) [8] provides the location of the face in each frame, as well as approximative landmarks of the features of interest, from which rough regions of interest are defined.

We will denote by N_D (resp. N_M) the set of nodes of G_D (resp. G_M), by E_D (E_M) the set of edges of G_D (G_M) (in our experiments, $E_D = N_D \times N_D$ and $E_M = N_M \times N_M$).

Both graphs are attributed, and the recognition will rely on similarity functions between attributes. As an illustration, we use here the following attributes. Let a be a node of G_D or G_M. The node attributes are defined as the average grey level $g(a)$ of the region represented by a (normalized between 0 and 1), a texture index $w(a)$ computed from wavelet coefficients, and the region label $l(a)$ provided by

*This work was partially supported by CAPES-COFECUB (369/01), CNPq-CNRS (91.0125/00-7), FAPESP (99/12765-2), CNPq (300722/98-2).

1051-4651/02 $17.00 © 2002 IEEE

the tracking procedure. The latter attribute aims at restricting the search space.

Let a and b be two nodes of G_D (or of G_M). The edge attribute is defined as the vector $v(a,b) = \frac{\vec{p_a p_b}}{2d_{max}}$ where p_a and p_b are the centroids of the regions represented by a and b respectively, and d_{max} is the largest distance between any two points of the face region. This attribute is not symmetrical in a and b and therefore edges are directed.

Note that this graph construction and the choice of attributes are specific to the considered application and to our current experimentation, while the following two sections are more generic.

3. Recognition as an inexact matching problem

Since in general $|N_D|$ is much larger than $|N_M|$, no exact correspondence can be expected. Therefore, assigning a node of G_M to each node of G_D is an inexact graph matching problem. Recognition can then be performed by searching for a homomorphism between G_D and G_M which satisfies both structural and similarity constraints. A graph homomorphism is a mapping $h : N_D \to N_M$ such that:

$$\forall (a_D^1, a_D^2) \in N_D^2, (a_D^1, a_D^2) \in E_D \Rightarrow (h(a_D^1), h(a_D^2)) \in E_M$$

which imposes a structural constraint on the mapping between edges, and guarantees that each data node has exactly one label (i.e. model node).

Similarity constraints are then used to find the best homomorphism. They are based on comparison of attributes. A good homomorphism will maximize the similarity between attributes of matched nodes and between attributes of matched edges, or, equivalently, minimize a dissimilarity. The dissimilarity between any two nodes $a_D \in N_D$ and $a_M \in N_M$ is defined here as:

$$c_N(a_D, a_M) = \beta |g_D(a_D) - g_M(a_M)| +$$

$$(1-\beta)|w_D(a_D) - w_M(a_M)| \text{ if } l_D(a_D) = l_M(a_M)$$

where g_D, w_D, l_D (g_M, w_M, l_M) are the node attributes of graph G_D (G_M), and β a parameter for tuning the relative importance of grey level and texture index (in our experiments, we set $\beta = 0.7$). If the condition in this equation is not satisfied, then the dissimilarity between a_D and a_M is considered as maximal and the corresponding solution is discarded. This criterion helps the tree-based search algorithms converging fastly to a solution, though it may lead to errors mainly near the boundaries of each label region. Nevertheless, such possible situations do not compromise the results in our experiments.

The dissimilarity between two edges $e_D = (a_D^1, a_D^2)$ of E_D and $e_M = (a_M^1, a_M^2)$ of E_M is defined as:

$$c_E(e_D, e_M) = \|v(a_D^1, a_M^1) - v(a_D^2, a_M^2)\|.$$

Based on these functions, we propose to define a global dissimilarity as:

$$f_1(h) = \frac{\alpha}{|N_D|} \sum_{a_D \in N_D} c_N(a_D, h(a_D)) +$$

$$\frac{1-\alpha}{|E_D|} \sum_{(a_D^1, a_D^2) \in E_D} c_E((a_D^1, a_D^2), (h(a_D^1), h(a_D^2))), \quad (1)$$

where α is a parameter used for tuning the relative importance of node dissimilarity and edge dissimilarity. In our experiments we have set $\alpha = 0.4$.

4. Optimization algorithms

In this Section, we briefly present three types of algorithms aiming at optimizing the global dissimilarity function.

Randomized tree search. The first algorithm is an extension of a deterministic tree search algorithm [2]. The algorithm creates a search tree where each vertex corresponds to a pair (a_D, a_M). It is initialized with a dummy node that is expanded in $|N_M|$ sons $(a_D^1, a_M^1), ...(a_D^1, a_M^{|N_M|})$. The dissimilarity function is computed for each node. Let us assume that (a_D^1, a_M^3) has the smallest value. Then it is analogously expanded in $|N_M|$ sons $(a_D^2, a_M^1), ...(a_D^2, a_M^{|N_M|})$ and the dissimilarity function is calculated for each son (this time involving also dissimilarity between edges (a_D^1, a_D^2) and (a_M^3, a_M^i)), and so forth, until a tree vertex of the form $(a_D^{|N_D|}, a_M^l)$ is reached. All leaves are possible candidates for further expansion. Here we propose to randomize the search by creating a priority list of k candidates ($k = 100$ in our experiments) having the smallest dissimilarity. Then the next leaf to be expanded is chosen randomly among the candidates of this list. This procedure is executed several times, and since different executions may lead to different solutions, we select the best one. This algorithm is very fast and provides much better results than the deterministic one.

Genetic algorithms (GAs). Two types of GAs have also been implemented and tested, in which each individual represents a correspondence hypothesis. The first one is the elitist algorithm (eGA) [5], and consists in keeping the best individual in a population and regenerate all the other individuals for the next generation. The second one is a steady-state algorithm (ssGA) [9], where two individuals are randomly chosen in a generation, undergo cross-over in order to generate a new individual, which is then tested against the worst individual of the generation. The worst of both is then removed.

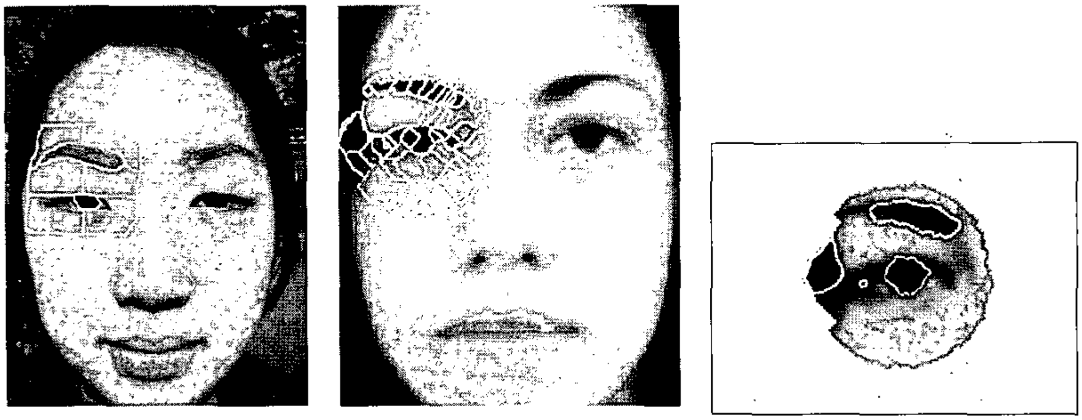

Figure 1. Left: model; middle: over-segmented image (subset); right: results obtained with EDAs.

Estimation of distribution algorithms (EDAs). EDAs [4] are non-deterministic, stochastic heuristic search strategies within the evolutionary computation approaches, where a number of solutions or individuals are created every generation evolving until a satisfactory solution is achieved. The main characteristic of EDAs comparing to other evolutionary search strategies is that the evolution from a generation to the next is done by estimating the probability distribution of the fittest individuals, and afterwards by sampling the induced model. In EDAs, the individuals are not said to contain genes, but variables (again representing correspondence hypotheses) whose dependences have to be analyzed. Furthermore, while in other evolutionary computation heuristics the interrelations between the variables forming the individuals are considered implicitly (i.e. building block hypothesis), in EDAs the interrelations are expressed explicitly through a probability distribution estimated from the fittest individuals of the previous generation. Here, we use different forms of EDAs, described in more detail in [1]: UMDA, PBIL and BSC where all variables are considered to be independent, MIMIC, where pairwise dependences are included, and EBNA (using two different learning scores, BIC and K2), where all possible dependences are considered. The last two categories require the generation of a probabilistic graphical structure in a form of a Bayesian network, showing the dependences between the variables. In the first category, an arc-less structure is always assumed.

5. Results

The proposed method was first tested on an ideal example, where the solution is known. The model was built manually from a real image and contained 8 regions. These regions were further subdivided in order to build the data graph with 23 regions. On this example, all algorithms found the correct solution, except the deterministic tree search and the elitist GA (as was expected).

Now we present a real example, where the model and the data do not correspond to the same person (but under similar acquisition conditions). The model contains 8 regions and is shown in Figure 1 (left). The image is automatically segmented using the watershed algorithm, generating a data graph with 75 nodes corresponding to one of the 4 rough regions provided by the GWN tracking procedure [2]. It is shown in Figure 1 (middle). The best result is shown in Figure 1 (right), and has been obtained by all EDAs. The contours of the recognized eyebrow and pupil (including one small erroneous region on the left of the eye) are shown, as well as those of the hair region, which has no correspondence in the model. Figure 2 shows the histogram of the smallest dissimilarity values obtained with a large search over possible solutions, i.e. the stochastic tree search algorithm has been applied approximately 50000 times. The histogram corresponds to these solutions found by the stochastic tree search algorithm. The best solutions obtained by the stochastic algorithms are also indicated.

Table 1 summarizes the obtained results with all proposed algorithms. The number of wrongly labeled regions is computed based on a manual labeling of the segmented image. All EDAs provide the same solution. Although their computation time is higher (varying from 1 min for UMDA and BSC to 8 min for EBNA$_{BIC}$), they lead to a better optimization. The 5 errors they made are also included in the solutions provided by the other algorithms. The stochastic tree search and ssGA are somewhat worse.

Among the 5 errors, 3 correspond to very tiny regions, for which even the manual result cannot be considered as reliable. One error corresponds to a region in the hair, and exhibits one of the limits of the proposed approach, in cases

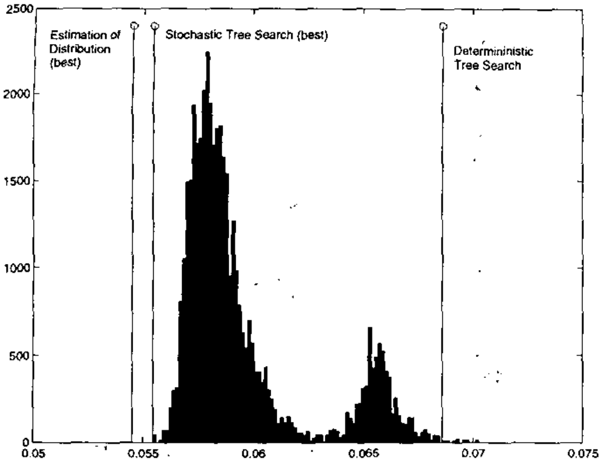

Figure 2. Histogram of the best values found for $f_1(h)$.

Algorithm	$\min_h(f_1(h))$	#,of errors
Deterministic tree search	0.0686	50 (66.0%)
Stochastic tree search	0.0555	13 (17.0%)
cGA	0.0906	54 (71.0%)
ssGA	0.0546	7 (9.3%)
UMDA	0.0545	5 (6.6%)
PBIL	0.0545	5 (6.6%)
BSC	0.0545	5 (6.6%)
MIMIC	0.0545	5 (6.6%)
EBNA$_{BIC}$	0.0545	5 (6.6%)
EBNA$_{K2}$	0.0545	5 (6.6%)

Table 1. Summary of the results.

where the data contain objects that are not represented in the model. An extension of this approach could be to allow for no recognition in such cases. Finally, the last error corresponds to a region at one extremity of the eyebrow, which is labeled as skin. The reason is that the contour between the eyebrow and the skin was not detected by the watershed algorithm and therefore this region contains also some skin.

In the model, the skin is divided in several regions, which are correctly recognized in the data. This result illustrates the influence of edge attributes when node attributes are about the same.

6. Concluding remarks

From our preliminary results, we can conclude that stochastic algorithms are very promising for solving inexact graph matching problems applied to facial feature recognition. Future work aims at extending the experiments to larger size problems. It is worth mentioning that the stochastic tree search algorithm has already been successfully applied to images from different databases, thus increasing our confidence in the method. We also intend to extend the proposed approach to video sequences by taking benefit from the limited variations from frame to frame, as well as to compare it to other graph-based approaches, such as [10].

References

[1] E. Bengoetxea, P. Larranaga, I. Bloch, and A. Perchant. Solving Graph Matching with EDAs Using a Permutation-Based Representation. In P. Larranaga and J. A. Lozano, editors, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, chapter 12, pages 239–261. Kluwer Academic Publisher, Boston, Dordrecht, London, 2001.

[2] R. Cesar and I. Bloch. First Results on Facial Feature Segmentation and Recognition using Graph Homomorphisms. In *VI Simpósio Ibero-Americano de Reconhecimento de Padrões SIARP 2001*, pages 95–99, Florianapolis, Brazil, oct 2001.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New-York, 1979.

[4] P. Larrañaga and J. A. Lozano Eds. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

[5] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlin Heidelberg, 1992.

[6] M. Pantic and L. Rothkrantz. Automatic analysis of facial expressions: the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[7] A. Perchant and I. Bloch. A New Definition for Fuzzy Attributed Graph Homomorphism with Application to Structural Shape Recognition in Brain Imaging. In *IMTC'99, 16th IEEE Instrumentation and Measurement Technology Conference*, pages 1801–1806, Venice, Italy, May 1999.

[8] V. K. R. S. Feris and R. M. Cesar Jr. Efficient real-time face tracking in wavelet subspace. In *Proc. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, 8th IEEE International Conference on Computer Vision (RATFG-RTS - ICCV 2001 - Vancouver, Canada)*, pages 113–118, 2001.

[9] D. Whitley and J. Kauth. GENITOR: A Different Genetic Algorithm. In *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, volume 2, pages 118–130, 1988.

[10] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.