Model-based scene recognition using graph fuzzy homomorphism solved by genetic algorithm

Aymeric Perchant⋆, *Claudia Boeres*[1)]⋆, *Isabelle Bloch*⋆, *Michel Roux*⋆ *and Celso Ribeiro*†

⋆ Ecole Nationale Supérieure des Télécommunications, Dpt. TSI, CNRS URA 820, 46 rue Barrault, 75634 PARIS Cedex 13, France,

† Catholic University of Rio de Janeiro, Dpt. of Computer Science, R. Marquês de São Vicente 225, Rio de Janeiro 22453-900, Brazil

perchant@tsi.enst.fr, boeres@tsi.enst.fr (or boeres@pep.ufrj.br), bloch@tsi.enst.fr, mroux@tsi.enst.fr, celso@inf.puc-rio.br

Abstract:

We present in this paper a new formalism for non bijective graph matching, based on fuzzy homomorphism. The search of the optimal solution is performed by a genetic algorithm adapted to this formalism. This type of matching is used for model-based scene recognition, where objects and their relations are represented as graphs, in the scene and in the model. Our method deals with imperfect matching, where no graph or subgraph isomorphism can be expected.

1 Introduction

In many complex picture processing tasks, the recognition of individual components of the scene is very difficult because of the similarity between the objects, leading to a poor discrimination, and because of their variability. These features prevent recognition based on object characterization only. When the scene components are embedded in a highly reliable set of structural links, we may expect a global interpretation of the scene, in particular if a structural reference model for it is available. Such situations are encountered in aerial imaging, computer vision or medical imaging, when strong a priori information exists on the structure of the scene to be recognized. The underlying structure calls for representations of the scene and of the model in terms of graphs, and for an interpretation process in terms of graph morphisms (most often graph or sub-graph isomorphisms in the literature). The issue of scene recognition modeled with graphs is then to extract the graph structure using observed

[1)]Supported by a Brazilian sponsor scholarship (CAPES)

relations between (previously) segmented objects, and to use the reference model represented as a graph to recognize individual objects based on the structural information carried by the graph.

The literature on the use of graphs in image processing is growing, and covers different types of graphs. Association graphs often used in bipartite graph matching ([12] is a reference on the subject) are a model of the matching itself. Here, in the structural recognition problem we address, we rather consider graphs as representations of objects and their links. Graphs for data representation can be found at different levels. Pixel graphs widely used in Markov model (e.g. [8]) are limited to basic pixel neighborhood relations. Other low-level pyramidal graphs (trees) such as classical quadtrees (or as in [1] where a pyramidal model is used for object tracking in a sequence) are useful for segmentation purpose, but not recognition purpose when the underlying structure is not pyramidal. At intermediate level, graphs can also be built on clustering results, but one main problem is that the segmentation by clustering algorithms do not create connected objects, but a set of regions with common properties (e.g. [7]). To have structured individual objects, nodes of the graphs can represent connected regions resulting from a segmentation or clustering process, and arcs can represent relationships. The most used relationship is spatial adjacency (e.g. [20]). Other relations can be used to build the graph, like distance relations or other similarity measures between objects (e.g. [11] where the author uses a feature similarity based graph). In the main domains of application of such graphs (character recognition, e.g. in [5], where Chinese character are modeled with fuzzy attributed graphs of stroke adjacency, or face recognition, e.g. [22]), the graphs are generally small (i.e. less than 30 nodes).

Complex scene recognition problems partly belong to these two last cases, but the goal is different, and thus the methods will be. Our recognition aim is to compare one model to one scene, and not one character to all the alphabet. All our model will be much more complex (bigger graph, more relations between objects), but we have one global matching to do, and not several little ones. This complexity will bring inaccuracies throughout all the model, and will forbid the search of a perfect (i.e. bijective) relation between the model and the scene. Handling all these inaccuracies is mandatory for such a problem, because of their compulsory presence.

The contribution of this paper is, first of all, the new formalization of this problem, as a graph fuzzy homomorphism. This formalization enables to cope with the inevitable inaccuracies of the data. The second contribution is the handling of these defects by using fuzzy sets in the graph model and in the morphism between the model and the scene, and by using a generic homomorphism instead of an isomorphism. Finally we aim at proposing an original genetic algorithm using our model to solve the morphism part.

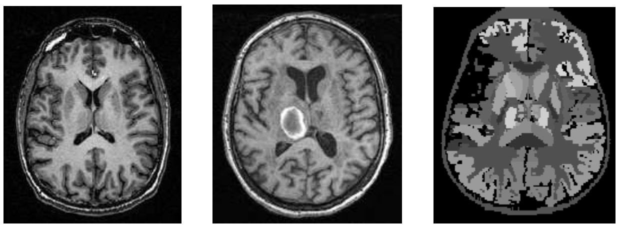
Section 2 will describe more precisely the specific characteristics of the problem. The models and their constructions will be explained in Section 3. The proposed fuzzy homomorphism will afterwards be introduced in Section 4. Section 5 will define a genetic algorithm using the fuzzy homomorphism formalization and Section 6 will show our first results. Section 7 will conclude about the paper, and about the first medical imaging application given as our basic example through the paper.

2 Problem description

The problem considered in this paper is the problem of structural object recognition by graph morphism from one complex scene to one complex model. We have seen in Section 1 that it may happen that the structure of the objects of the scene is mandatory for the recognition process. But the structure is not unique, and several structure types can be extracted, using adjacency graph, or distance graph. A natural choice is to use attributed graphs, or fuzzy attributed graphs. A node represents an object and holds attributes of this object. An arc between two objects holds several attributes describing their relations. The choice of the relations are highly problem-dependent and will not be discussed in this paper.

Inaccuracies constitute one main characteristic of the problem. Objects are segmented from the scene: all the problems of object segmentation will be reflected in the scene model. For instance, these problems can be undersegmentation, oversegmentation, unexpected object found in the scene, or expected object not found, great deformation of objects, and misplacement of one object in the object structure. Because our problem deals with complex scene recognition it is impossible to avoid absolutely all these drawbacks of the scene. For example, in aerial imaging, road extraction can be corrupted, clouds can hide some roads, poor camera calibration can misplace all the objects, bad focusing can blur the road and modify attributes such as width of the road, bad segmentation can miss some part of a road [13]. Similar problems occur in other applications.

The model of the scene is itself often inaccurate too. It can be incomplete, or objects attributes can be imprecise. Another problem can originate from the possible genericity of the chosen model. For example, in medical imaging, one purpose is to segment anatomical structures in the brain from Magnetic Resonance Imaging (MRI). Because everybody should have the same anatomy, one generic model can be chosen. But we also know that brain structures show inter-individual variability in the shapes and the placements of the structures. Though, these types of attributes cannot be exact attributes. If the brain scene is pathological, then the pathology is one more object. This pathology can be in place of another disappeared object. This example will be kept for the Sections 3 and 4, and is exactly the same example as in [16], where details about the application objectives can be found. To illustrate these difficulties, Figure 1 presents a slice of three different volumes: Figure 1-a is a normal brain, Figure 1-b is a pathological brain (with a tumor), Figure 1-c is the representation of a brain atlas where one grey level corresponds to one unique connected structure. Middle dark structures are much bigger on Figure 1-b than Figure 1-a, white hypersignal structure (tumor) is neither in the atlas (1-c) nor in the normal brain (1-a). The neophyte reader can appreciate himself the numerous differences between these images.



(a) (b) (c)

Figure 1: MR image examples: (a) is an axial slice of a normal brain, (b) is an axial slice of a pathological brain with a tumor, (c) is an axial slice of a brain atlas.

We can separate the inaccuracies in two parts: the first one concerns the model itself, and the way both the scene representation and the reference model are built (as graphs here). We will see that these two graph structures can be different. The second one concerns the fact that the morphism between these two structures cannot be a one-to-one mapping (i.e. an isomorphism). The first type of inaccuracies will be handled by two features of our method: the attributes describing objects and their relations will be fuzzy sets, and the graph building will be based upon the possible defect of the problem. This part is explained in Section 3. The second type of inaccuracies concerns the mapping between these possible two different structures: we chose to handle it by introducing the new concept of graph homomorphism. Moreover, because the attributes are fuzzy, the homomorphism will be itself described by fuzzy sets. This part is explained in Section 4.

3 Graph-based model and construction of the graph structure

A type of graph that is now widely used in related problems is the Fuzzy Attributed Graph (FAG) defined for example in [5]. The definition and implementation of the fuzzy attributes are specific to the problem at hand. For example, in our brain medical imaging example, possible attributes for the relations between brain structures are: fuzzy adjacency [4], fuzzy distance [3] and fuzzy relative position [2].

Depending on the imperfections of the scene, or of the reference model, and on the object relation attributes, the construction of the graph is different. The common point is that we have one node for one region (either from the reference model or the segmented scene). Differences occur when creating the arcs, because of the chosen attributes, or the defect of the segmentation for the scene. The possibilities can be divided in five cases described below, and are useful when the scene is complex enough (more than 30 objects).

The first case is the *complete graph*: a complete graph structure is created, and for each arc, the attributes linking two objects are computed. This case can be useful in undersegmentation cases, because all relations are useful. The second case is the *simplification of a complete graph*: arcs of which attributes are not significant are taken out of the graph. This case is useful when undersegmentation occurs sparsely. The third case is the *local building*: when relations are not significant between all the objects of the whole scene, the arcs are created by propagation of a set of local relation attributes. Adjacency is a simple example of such case. This building method is useful when there is a lot of objects and when the segmentation has no real defect. The fourth case is the *k-local transitive closure on local building*: when homogeneous oversegmentation occurs. This operation is the following for $k = 1$ and is close to the classical operation of transitive closure of a set:

```
copy graph G to graph G'
For all nodes u of G do
    For all neighbors v of u do
        For all neighbors w of v do
            add (u,w) to G'.
```

For k greater than 1, the process is iterated. The value of k is chosen as a function of the importance of the oversegmentation. The fifth case is the *k-local conditional transitive closure on local building*, when heterogeneous oversegmentation occurs. The process is to conditionally add an arc by transitivity closing only if it links two oversegmented regions. This last case is possible only if we have a way of estimating the local degree of oversegmentation. The method described in this paper is able to deal with all cases.

The next step is to compute the attributes of the nodes and the arcs. This step can be made during the graph structure building, depending on the method. In our example of medical imaging [16], the reference model is an atlas based upon the manual segmentation of a normal brain by a radiologist. The scene is homogeneously oversegmented, and objects can have shape modifications. Thus, the building of the reference graph is a local building, and the scene graph is a 2-local transitive closure on local building using fuzzy adjacency. The other arc attributes have been given in Section 3.

4 Fuzzy homomorphism of graphs

This part aims at defining the basic definition of a generic fuzzy graph homomorphism. Fuzzy morphism was first introduced by Rosenfeld in [18]. We need two graph structures to define the morphism: $G_i = (N_i, E_i)$, where N_i is a vertex set, $E_i \subseteq N_i \times N_i$ is an arc set, and $i \in \{1, 2\}$

refers to the graphs. Letters with σ index will refer to nodes, and letters with μ index will refer to arcs, according to Rosenfeld's notations of fuzzy graphs [18]. A fuzzy homomorphism from the graph G_1 to the graph G_2 is the pair of membership functions (ρ_σ, ρ_μ) defined on the vertices and on the arcs:

$$\rho_\sigma = \{{\rho_\sigma}^{u_1} : N_2 \to [0,1],\ u_1 \in N_1\}$$
$$\rho_\mu = \{{\rho_\mu}^{(u_1,v_1)} : E_2 \to [0,1],\ (u_1,v_1) \in E_1\}$$
$$\text{such that } \forall (u_1,v_1) \in E_1,\ {\rho_\mu}^{(u_1,v_1)}(u_2,v_2) > 0 \Rightarrow {\rho_\sigma}^{u_1}(u_2) > 0 \text{ and } {\rho_\sigma}^{v_1}(v_2) > 0 \qquad (1)$$

Equation 1 only guarantees that the correspondence between arcs implies a non-zero correspondence between their extremities. Depending on the problem, other hypotheses on the morphism can be added, like Equation 2 below. This last property is explained in [16] and is useful if the graph structure must be kept during the algorithm that searches the best fuzzy homomorphism.

$$\forall (u_1,v_1) \in E_1 : supp({\rho_\sigma}^{u_1}) \neq \emptyset \text{ and } supp({\rho_\sigma}^{v_1}) \neq \emptyset$$
$$\Rightarrow \exists (u_2,v_2) \in supp({\rho_\sigma}^{u_1}) \times supp({\rho_\sigma}^{v_1}) \text{ such that: } (u_2,v_2) \in supp({\rho_\mu}^{(u_1,v_1)}), \qquad (2)$$

where $supp$ denotes the support of a fuzzy set, i.e. $supp({\rho_\sigma}^{u_1}) = \{u_2 | {\rho_\sigma}^{u_1}(u_2) \neq 0\}$.

The aim of this definition is to work on both the vertices and the arcs as a whole structure, whereas other definitions work only on vertices and consider the arcs as a guide to match two graphs. The morphism is expressed in terms of degree in order to handle possible variations of the attributes between the two graphs. This definition is given for a classical graph structure. The use of FAG or other graph types does not modify this definition because only the structures are matched. The way two nodes (or two arcs) are matched and the degree of the matching must be computed using the attributes. The fitness function of the genetic algorithm (explained in the following part) is a degree of the matching.

In our application, ρ_σ and ρ_μ are computed as functions of similarity measures between attributes of nodes (respectively arcs). Unlike in relaxation-based methods for matching optimization [16], where ρ_σ and ρ_μ are iteratively updated, they are kept fixed in the proposed genetic algorithm method.

5 Graph morphism problem solved by genetic algorithms

The research in Genetic Algorithms (GA) is increasing since the sixties, when John Holland and his group of Michigan University started to develop studies of these algorithms resulting in the publication of the book *Adaptation in Natural and Artificial Systems*, 1975 [10].

Basically, the GA starts its evolution with a random generation of a population of p *individuals* (also denoted as *chromosomes*), usually represented as binary vectors (each one of them represents a feasible solution of the treated problem). This population is submitted to an iterative process composed of three principal steps: evaluation (according to a fitness function), selection of the best individuals and application of the genetic operators: *crossover* (reconfiguration of the chromosomes) and *mutation* (random change of components of a chromosome). This process is repeated until a defined *termination criterion* is reached. In many applications, the way of representing the individuals and also the definition of each operation surely depends on the knowledge that one has of the problem.

This description of GA is denoted in [21] as a *canonical genetic algorithm* and refers to the model first introduced and investigated by Holland and widely used in several applications until nowadays. Despite this algorithm looks simple and very random, in [10] Holland and his group develop some arguments to explain its robustness as a search method.

Genetic Algorithms belong to a broader class of algorithms, called Evolutionary Computing. Typically, in this types of algorithms, the aim is to generate new sample solutions in a search space, using principles of natural evolution and inheritance. This family of algorithms has recent interesting developments in graph morphism [6, 19]. This section is concerned with the proposal of utilization of the canonical approach as an optimization tool for solving the Graph Morphism problem. For this, we will define all data and parameters needed by the *canonical genetic algorithm*, considering the specific features of the problem.

Chromosomes: A solution is represented as a set of vertex associations established between the compared graphs. Each individual of the population is defined as a matrix c of $|N_1| \times |N_2|$ binary values.

Population: In the first iteration of our GA, the population is randomly generated. We can improve the initial population, allowing only a limited number of associations to each vertex of G_1. In our experiments, we set this number as $2 * \frac{|N_2|}{|N_1|}$. To achieve feasibility ($c_{ij} = 1$ only if there exists an association between u_1^i and u_2^j with weight $\rho_\sigma^{u_1^i}(u_2^j) > 0$), a repairing step is needed to eliminate associations ($c_{ij} = 1$) for which $\rho_\sigma^{u_1^i}(u_2^j) = 0$. Several issues about the feasibility of solutions generated by GA are addressed in [15]. Here, we use the method of always conserving feasibility: at the end of each GA iteration, we verify if there exists any infeasible individual in the population and we repair it.

Population size is a parameter. Empirical results suggest that not very large populations are adequate [17]. In our experimental tests, we fixed population size as 50 individuals.

Fitness function: The fitness function to be maximized is defined as

$$f = \frac{\alpha}{|N_1| \times |N_2|} \left[\sum_{i=1}^{|N_1|} \sum_{j=1}^{|N_2|} (1 - |c_{ij} - \rho_\sigma^{u_1^i}(u_2^j)|) \right] + \frac{(1-\alpha)}{|E_1| \times |E_2|} \left[\sum_{k=1}^{|E_1|} \sum_{l=1}^{|E_2|} (1 - |c_{ij} c_{i'j'} - \rho_\mu^{e_1^l}(e_2^k)|) \right] \quad (3)$$

where α is a parameter used to weigh each term of f, c_{ij} is a component of a chromosome c, $e_1^l = (u_1^i, u_1^{i'})$, and $e_2^k = (u_2^j, u_2^{j'})$.

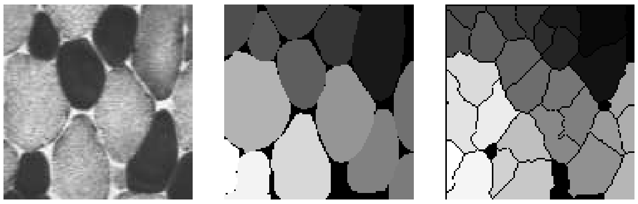
The value of f associated to each chromosome measures the correspondence established between the compared graphs. The first part of the right-hand side of Equation 3 represents an average of the node contribution to the correspondence. A similar analysis can be done for the terms which compose the second part of the right-hand side of Equation 3, where only the compared edges which satisfy the constraint represented by Equation 1 are considered.

Operators: The operations of selection, crossover and mutation used here are the classical ones, as defined in [14, 9].

Termination: The termination criterion usually adopted is a fixed number of iterations, or even a specific number of iterations after which the fitness value of the best individuals is unchanged. We consider both criteria in our experimental tests.

6 Results

This first test was performed on a real simple 2D image example (Figure 2-a) in order to understand the behavior of the method. This image was oversegmented in 28 regions (Figure 2-c) and compared with a model image of 14 well defined regions (Figure 2-b). The model graph has 14 vertices and 26 edges and the image graph has 28 vertices and 62 edges. We used grey level as node attribute; fuzzy distance and fuzzy adjacency between regions as edge attributes. In the results presented here, we fixed population size of 50 chromosomes, Mutation probability (p_m) of approximately $\frac{1}{|N_1| \times |N_2|}$ and a maximum of 100 iterations.



(a) (b) (c)

Figure 2: Test images: slice of a muscle. (a) is the original image, (b) is the atlas, and (c) is the automatical segmentation.

Figures 3 shows the best results obtained by the canonical genetic algorithm for the example described, with $\alpha = 0.2$ (most influence of edges). The crossover probabilities p_c are showed in Figure 3. With maximal p_c value, 50% of model regions are fully recognized while others 29% are partially recognized. Figure 3 shows that 71% of the regions can be recognized, but not necessarily with the same p_c value. In the way the individuals were defined, good correspondences for some model nodes can be eventually disrupted with high probabilities for crossover. It can explain good results achieved with lower p_c values. Modifications of the classical crossover for this problem are proposed in future work. The evolution time spent

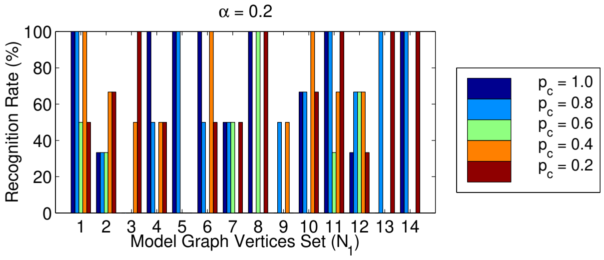


Figure 3: Recognition results obtained for the 28 regions segmented image

by the algorithm for this small example was five seconds on a SUN Ultra-2 station machine.

We believe that the low quality of solution is due to the non discriminant aspect of attributes. Because of this, good fitness values of chromosomes can not correspond to really good graph associations and the GA has not a good development. Also, the spatial image structure is not enough detected by the fitness function, because we evaluate only pair of nodes correspondences (considering edge information) instead of subgraph correspondence. Our current work aims at defining new evaluation functions by considering both morphism properties (Section 4) and using the idea of subgraph correspondence. Also, others vertice and edge attributes are introduced in order to better describe the images.

7 Conclusion

We have presented a new generic definition of graph fuzzy homomorphism. Minimal properties have been proposed in order to keep graph structure through the morphism. Several properties of this definition still need to be explored. The search of the best graph morphism was performed with a genetic algorithm adapted to this problem. First results are promising. Future work aims at refining the genetic algorithm and at performing more extensive tests on real data, in order to evaluate the method.

References

[1] Pascal Bertolino and Stéphane Ribas. Image Sequence Using a Single Evolutionary Graph Pyramid. In *Graph Based Representations in Pattern Recognition*, pages 93–100. Springer-Verlag, 1998.

[2] I. Bloch. Fuzzy Relative Position between Objects in Image Processing: a Morphological Approach. *Pattern Analysis Machine Intelligence*, 21(5), 1999.

[3] I. Bloch. On Fuzzy Distances and their Use in Image Processing under Imprecision. *Pattern Recognition*, 32(11), 1999.

[4] I. Bloch, H. Maître, and M. Anvari. Fuzzy Adjacency between Image Objects. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5(6):615–653, 1997.

[5] K. P. Chan and Y. S. Cheung. Correction to "Fuzzy-Attribute Graph with Application to Chinese Character Recognition". *IEEE Transactions on Systems, Man and Cybernetics*, 22(2):402–410, Mars.–Avr. 1992.

[6] Andrew D. J. Cross, Richard C. Wilson, and Hancock Edwin R. Inexact Graph Matching using Genetic Search. *Pattern Recognition*, 30(6):953–970, 1997.

[7] Christophe Demko, Pierre Loonis, and El-Hadi Zahzah. Isomorphism of Fuzzy Structures: a New Method for Image Classification. In *Proceedings of the 9th Scandinavian Conference on Image Analysis*, volume 1, pages 297–304, Uppsala, Sweden, June 1995.

[8] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[9] D. E. Goldberg. *Genetics Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[11] H. Jahn. A Graph Structure for Grey Value and Texture Segmentation. In *Graph Based Representations in Pattern Recognition*, pages 73–82. Springer-Verlag, 1998.

[12] L. Lovàsz and M. D. Plummer. *Matching Theory.* Mathematics studies, Annals of Discrete Mathematics. North-Holland, Elsevier science, 1986.

[13] H. Maître, I. Bloch, H. Moissinac, and C. Gouinaud. Cooperative Use of Aerial Images and Maps for the Interpretation of Urban Scenes (invited conference). In *Ascona Workshop "Automatic Extraction of Man-Made Objects from Aerial and Space Images"*, pages 297–306, Ascona, Suisse, 1995.

[14] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1992.

[15] Z. Michalewicz. Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1:177–206, 1995.

[16] Aymeric Perchant and Isabelle Bloch. A New Definition for Fuzzy Attributed Graph Homomorphism with Application to Structural Shape Recognition in Brain Imaging. In *IEEE Instrumentation and Measurement Technology Conference proceedings, Venice, Italy*, volume 3, pages 1801–1806, May 1999.

[17] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, 1993.

[18] A. Rosenfeld. *Fuzzy Sets and their Applications*, chapter Fuzzy Graphs. Academic Press, New York, 1975. L.A. Zadeh and K.S. Fu and M. Shimara. Eds.

[19] Montek Singh, Amitabha Chatterjee, and Santanu Chaudhury. Matching Structural Shape Description using Genetic Algorithm. *Pattern Recognition*, 30(9):1451–1462, Sept. 1997.

[20] Milan Sonka, Satish K. Tadikonda, and Steve M. Collins. Knowledge-Based Interpretation of MR Brain Images. *IEEE transactions on medical imaging*, 15(4):443–451, August 1996.

[21] D. Whitley. A Genetic Algorithm Tutorial. Technical report, Computer Science Department, Colorado State University, 1997.

[22] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face Recognition by Elastic Bunch Graph Matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.