3D FACIAL EXPRESSION RECOGNITION USING SWARM INTELLIGENCE

Iordanis Mpiperis[1,2], *Sotiris Malassiotis*[1], *Vassilios Petridis*[2] *and Michael G. Strintzis*[1,2]

[1] Informatics and Telematics Institute
Centre for Research and Technology Hellas
Thermi-Thessaloniki 57001 Greece

[2]Electrical and Computer Engineering Department
Aristotle University of Thessaloniki
Thessaloniki 54124 Greece

ABSTRACT

In this paper, we present a novel approach for 3D facial expression recognition which is inspired by the advances of ant colony and particle swarm optimization (ACO and PSO respectively) in the field of data mining. Anatomical correspondence between faces is first established using a generic 3D face model which is deformed elastically to match the facial surfaces. Surface points are then used as a basis for classification according to a set of classification rules, which are discovered by an ACO/PSO-based rule discovery algorithm. The performance of the proposed algorithm has been evaluated on the BU-3DFEDB facial expression database where a total recognition rate of 92.3% was achieved.

Index Terms— Pattern recognition, intelligent systems

1. INTRODUCTION

Facial expression recognition is a challenging task which has received growing interest within the research community over the past years. While, however, many works have examined expression recognition from still images or video [1], a few researchers have addressed the problem using 3D facial information. Nevertheless, 3D expression recognition is expected to gain ground in the future, since depth can handle more effectively the main limitation factors of the current 2D state-of-the-art systems, i.e. head pose and illumination changes.

In this paper we present a new approach to the expression classification problem using purely depth information. We propose a classifier based on a set of rules which are discovered following the principles of swarm intelligence, in particular of Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The term "swarm intelligence" is used to describe a set of optimization algorithms inspired by the collective behaviour of social animals, such as insects, birds and fish. These algorithms consist of decentralized simple agents which interactively move in a high combinatorial search space to find an optimum or suboptimum solution. ACO has been introduced by Dorigo [2] and has been applied successfully to NP-hard problems where shortest distance paths are sought [3, 4]. In this work we use the framework of Ant-Miner [5], a modification of the original ACO able to discover classification rules. PSO, which simulates the cyclic "dancing" movement of flocks of birds, has been developed by Eberhart and Kennedy [6] and has been used initially for nonlinear function optimization. Its ability of moving in a high dimensional space avoiding local minima has been exploited soon in a variety of applications from training neural networks to data mining and protein classification [7]. However, the potential of these techniques in facial expression recognition has not been explored yet.

In this work, we use ACO and PSO to discover the classification rules which best discriminate the classes of prototypical facial expressions. First we fit a 3D deformable model to each input 3D facial scan under anatomical constraints and then the set of discovered rules is applied to the parameters describing the above model. The performance of the proposed algorithm has been evaluated on the BU-3DFEDB database [8] where a total recognition rate of 92.3% has been achieved.

This paper follows Ekman's approach to the classification of expressions. Ekman [9] proposed six prototypical emotions, *anger*, *fear*, *disgust*, *happiness*, *sadness* and *surprise*, each of which corresponds to a unique facial expression. Although the uniqueness and the universality of these emotions are still under dispute within the psychology community, most of researchers follow the above classification scheme. To this end, Wang et al [10] propose an algorithm where expressions are classified according to the distribution of several surface geometry descriptors. These descriptors are based on principal curvatures of different regions of the facial surface which are delimited according to the neuro-anatomic knowledge of the configuration of facial muscles and their dynamics. The authors evaluate their algorithm on the BU-3DFEDB database and they report a maximum 83.6% recognition rate achieved with a Linear Discriminant Analysis classifier. To the best of our knowledge, this is the only work which addresses the problem using purely 3D data and it is used in this paper for comparison.

The rest of the paper is organized as follows: In the next section we describe the procedure to establish anatomical cor-

This work was supported by research project "PASION - Psychologically Augmented Social Interaction over Networks" (contract No FP6-027654) under Information Society Technologies (IST) priority of the 6th Framework Programme of the European Community.

respondence among faces, in Section 3 we present the ACO/PSO framework for the construction of the classification rules and finally in Section 4 we evaluate the performance of the proposed algorithm by experimenting on the BU-3DFEDB database.

2. ESTABLISHING POINT CORRESPONDENCE

Our goal is to establish point-to-point correspondence among 3D facial surfaces so that alignment of anatomical facial features be guaranteed. Since 3D facial information is usually given in the form of a point cloud, we represent facial surfaces as deformations of a common generic 3D mesh. To achieve this, we model the facial surface as a subdivision surface similarly to [11]. Thus, each facial surface is represented by a fixed number of mesh vertices having the same topology among all faces.

A triangular 3D mesh M_0 with N_0 vertices, neutral expression and average identity is used as a base-mesh. This base-mesh is subdivided according to a subdivision rule resulting in a more dense mesh, whose vertices may be written as a linear combination of the base-mesh vertices. Subdivision may continue infinitely, converging to a smooth continuous surface which is a function of the base-mesh and the subdivision rule. However, in practice, we do not have to make infinite subdivisions since after a few levels (e.g. 3 in our experiments) the mesh becomes dense enough to approximate the subdivision surface. This final mesh M with N vertices $\mathcal{V} = \{\mathbf{v}_1 \dots \mathbf{v}_N\}$, where $\mathbf{v}_i = [x_i \ y_i \ z_i]^T$ are the coordinates of each vertex, serves as the generic face model which is deformed so that it resembles the cloud of facial points.

Since the subdivision surface has to be fitted to the 3D cloud of points guaranteeing anatomical correspondence, a set of landmarks corresponding to anatomically salient points of the face has to be defined both on the mesh and the cloud of points. Let $\mathbf{p}_i, i = 1 \dots K$ denote the points of the cloud and $\mathbf{y}_i, i = 1 \dots L$ the associated landmarks, e.g. $\mathbf{y}_1$ corresponds to the left eye leftmost point, $\mathbf{y}_2$ to the left eye rightmost point and so on. Similarly, we select a subset of vertices on M that anatomically correspond to the L landmarks. Thus, we can define easily a table $c(i)$ that maps each landmark index i to the corresponding vertex index of M.

Model fitting is formulated as an energy minimization problem which gives rise to two opposed force fields: An external field of attractive forces towards the cloud and an internal field of elastic forces which oppose to vertex displacements. The deformation energy is formulated as the weighted sum

$$E_{def} = \lambda_1 E_c + \lambda_2 E_{mc} + \lambda_3 E_{cm} + \lambda_4 E_e \tag{1}$$

E_c is the sum of the squared distances between landmarks and corresponding vertices given by

$$E_c = \sum_{i=1}^{L} \left(\mathbf{y}_i - \mathbf{v}_{c(i)}\right)^2 \tag{2}$$

E_{mc} is the sum of the squared directed distances from each mesh M vertex to the nearest point of the cloud whose index is returned by function $mc(\cdot)$, while E_{cm} is the sum of the inversely directed distances, from each point of the cloud to the nearest mesh vertex whose index is returned by function $cm(\cdot)$. Using both terms leads to a smoother force field between the mesh and the point cloud. Analytically we have

$$E_{mc} = \sum_{i=1}^{N} \left(\mathbf{v}_i - \mathbf{p}_{mc(i)}\right)^2 \tag{3}$$

$$E_{cm} = \sum_{i=1}^{K} \left(\mathbf{p}_i - \mathbf{v}_{cm(i)}\right)^2 \tag{4}$$

The above energy terms are responsible for the attractive forces, while internal elastic forces stem from E_e, which is a measure of the elastic energy of the mesh edges that penalizes their non parallel displacements. It is used as a regularization term which prevents mesh triangles from being folded and it is given by the equation

$$E_e = \sum_{i=1}^{N} \frac{1}{N_i} \sum_{j \in \mathcal{N}_i} \left(\mathbf{v}_i - \mathbf{v}_j - \mathbf{v}_i^0 + \mathbf{v}_j^0\right)^2 \tag{5}$$

where $\mathcal{N}_i$ is the set of $\mathbf{v}_i$'s neighbours, N_i is its cardinality and $\mathbf{v}_i^0$, $\mathbf{v}_j^0$ are the initial positions of the vertices.

The coordinates of vertices $\mathbf{v}_i$ that minimize E_{def} can be found by differentiating Eq. 1 with respect to $\mathbf{v}_i$ and setting the partial derivatives equal to zero. Differentiation leads to a linear system of equations which can be solved easily using Singular Value Decomposition. The estimated parameters $\mathbf{v}_i$ are subsequently used for expression recognition.

3. CLASSIFICATION USING SWARM INTELLIGENCE

Classification rules discovered using ACO and PSO are expressed in the form of IF-THEN rules:

IF<*term1* AND ... *termN*> THEN<*class*>

where the terms in the rule antecedent (IF part) are triples of the form <*attribute, operator, value*> and the class in the rule consequent (THEN part) is one of the six possible expressions. These rules are stored in a sequentially accessed list and expressions are classified to the class predicted by the first valid rule. In case no rule is valid, the expression is classified to a default class. Using this framework, the rule classifier actually divides the hyperspace of attributes to a number of hypercubes and classifies each attribute vector to the class associated with the hypercube the vector belongs to. Thus nonlinear shapes of the manifolds of the classes can be captured effectively leading to increased recognition performance.

ACO and PSO are involved in the rule construction step and more precisely in the construction of the rule antecedent.

ACO can solve problems where a shortest distance path on a graph is sought. The solution is found by "ants" which lay a pheromone trail while moving through the nodes of the graph. When an ant has to choose between several nodes for its next move, it is more likely to choose the node with the highest amount of pheromone. Since shortest graph edges are traversed more quickly, more ants will pass through them and thus more pheromone will be accumulated. At the same time accumulated pheromone will encourage more and more ants to follow shortest edges until all ants finally move through the shortest path. We also note that pheromone evaporates through time so that search of the space and avoidance of local minima be feasible.

Within this context, attributes are allowed to get discrete values which represent the nodes of the graph. Each ant constructs the antecedent part of its rule term by term. The value of the next attribute is selected according to: a) the pheromone of each possible value; b) the heuristic index of each possible value, which depends on the distribution of classes conditional on the value in question. The higher the amount of pheromone of a value, the higher the probability of being chosen and the more uniformly distributed the classes are conditional on this value, the smaller the probability of being chosen. Once a rule is constructed, it is pruned to remove terms that may have been introduced unduly due to the stochastic character of the rule construction. Pruning usually improves the quality Q of the rule which is defined as the product of sensitivity and specificity

$$Q = \frac{TP}{TP+FN} \times \frac{TN}{FP+TN} \quad (6)$$

[1]

Then, the pheromone of the terms of the temporarily best rule is increased, while the pheromone of the other terms is decreased. This process is iterated until an optimum rule is found. Then, the training samples covered by this rule are removed from the training set and the process is repeated to discover the next rule. When a few training samples remain in the training set, rule discovery is terminated and the default rule which assigns to the class of the majority of the samples is constructed.

The disadvantage of ACO is that it applies to discrete attributes. Although discretizing is not difficult to perform, it leads to the loss of order that continuous attributes possess and may be important to classification. To overcome this problem we have also explored PSO which can handle continuous data.

In PSO framework, the rule antecedent is represented by a particle moving in a high dimensional attribute space. The movement of the id-th particle is defined by its own experience, that is its past best position $\mathbf{b}_{id}$, and by the experience of its most successful neighbour, that is the local best position $\mathbf{b}_{loc}$. Its position $\mathbf{p}_{id}^t$ and its velocity $\mathbf{v}_{id}^t$ at iteration t are given by equations

$$\mathbf{v}_{id}^t = c_1\left(\mathbf{v}_{id}^{t-1} + c_2(\mathbf{b}_{id} - \mathbf{p}_{id}^{t-1}) + c_3(\mathbf{b}_{loc} - \mathbf{p}_{id}^{t-1})\right) \quad (7)$$
$$\mathbf{p}_{id}^t = \mathbf{p}_{id}^t + \mathbf{v}_{id}^t \quad (8)$$

where c_1, c_2 and c_3 are constants used to bound the velocity and prevent particles from oscillating without converging. From the above equations it can be seen that particles actually search the space between the best past position of their own and of their neighbours hoping to find better positions whose quality is defined again by Eq. 6.

During rule construction, the antecedent part is converted to a vector consisting of two dimensions per attribute, one for the lower bound and one for the upper. For instance, if there are two attributes a_1 and a_2 then we seek for a rule

$$\text{IF } l_1 \le a_1 \le u_1 \text{ AND } l_2 \le a_2 \le u_2 \text{ THEN } C$$

and the vector to be optimized consists of the four bounds $\mathbf{p} = [l_1\ u_1\ l_2\ u_2]^T$. A number of particles is let to move through the search space until they converge to an optimum position/rule. This rule is then pruned and added to the rule list, while the samples it covers are removed from the training set. Subsequent rules are found by iterating the whole process.

4. PERFORMANCE EVALUATION

The proposed algorithm is evaluated on the BU-3DFEDB database [8] which contains 56 female and 44 male subjects displaying the six universal expressions in four levels of intensity, *low*, *middle*, *high* and *highest*. For each subject, there is also a 3D face scan with neutral expression thus resulting in a total number of 2,500 face scans in the database. Each facial scan is also associated with a set of feature points located on the eyes, the eyebrows, the nose, the mouth and the face boundary, which have been detected manually and are used as landmarks during the establishment of point correspondence.

In our experiments we use the 10-fold cross-validation approach. First, we set all the faces of the database in anatomical correspondence as described in Section 2. Thus each facial surface is approximated by a 3D mesh with 2,500 vertices whose coordinates are stacked in the vector to be classified. Then, at each experiment we use 10 subjects chosen randomly as the validation set and the rest subjects as the training set assuring that each subject is included in the validation set at least once. Principal Components Analysis (PCA) is also applied to reduce data dimensionality and remove redundancy. Keeping the 98% of the data variance results in a 80D vector representation. Linear Discriminant Analysis (LDA) is then used to find a linear transformation which maximizes the intra-class scatter matrix and meanwhile minimizes the inter-class scatter matrix. This step results in a final 5D vector representation of facial surfaces which is the input of the ACO and PSO rule discovery algorithms. We remind that a discretization stage is also required before ACO is applied.

[1] TP=true positive, FP=false positive, TN=true negative, FN=false negative.

Table 1. Expression recognition rates in %.

	Wang [10]						ACO						PSO					
In/Out	Ang.	Dis.	Fea.	Hap.	Sad.	Sur.	Ang.	Dis.	Fea.	Hap.	Sad.	Sur.	Ang.	Dis.	Fea.	Hap.	Sad.	Sur.
Ang.	**80**	1.7	6.3	0.0	11.3	0.8	**58**	0	0	0	42	0	**75.3**	0	0.4	0	24.3	0
Dis.	4.6	**80.4**	4.2	3.8	6.7	0.4	0	**96.5**	3	0.5	0	0	0	**100**	0	0	0	0
Fea.	0	2.5	**75**	12.5	7.9	2.1	0	0	**100**	0	0	0	0	0	**100**	0	0	0
Hap.	0	0.8	3.8	**95**	0.4	0	0	1.3	1	**97.3**	0.4	0	0	0	0	**100**	0	0
Sad.	8.3	2.5	2.9	0	**80.4**	5.8	37.6	0	0	0	**62.4**	0	20.7	0	0.2	0	**79.1**	0
Sur.	1.7	0.8	1.2	0	5.4	**90.8**	0	0	1	0	0	**99**	0	0	0	0	0	**100**

During testing, each test face is subjected to the transformations defined by the PCA and LDA steps and then its expression is classified according to the discovered rules. The most time consuming part of recognition is searching for nearest neighbours during point correspondence establishment which is encountered by a space partition technique [12] that accelerates mesh fitting and allows real-time recognition.

Experimental results are summarized in Table 1, where rates reported in [10] are also included for comparison. As it is shown, the highest confusion occurs between the angry and sad expression. This is because these expressions differ mainly in the eyebrow configuration which however cannot be captured accurately when depth information is used. Nevertheless, a total recognition rate of 92.3% using the PSO classifier was achieved.

5. CONCLUSION

In this paper we proposed the use of ACO and PSO to extract optimum facial expression classification rules. These classifiers were applied to anatomically aligned vector representations of the facial surfaces obtained by deforming elastically a generic 3D face model. The high performance of the proposed approach was demonstrated on the BU-3DFEDB database where it was shown that further improvement can be achieved by focusing on the angry and sad expression discrimination problem.

6. REFERENCES

[1] M. Pantic and L. J. M. Rothkrantz, “Automatic analysis of facial expressions: The state of the art,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 22, no. 12, pp. 1424–1445, December 2000.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, “The Ant System: Optimization by a colony of cooperating agents,” *IEEE Trans. on Systems, Man and Cybernetics-Part B*, vol. 26, pp. 1–13, 1996.

[3] M. Dorigo and G. Di Caro, “The Ant Colony Optimization Meta-Heuristic,” in *New Ideas in Optimization*, David Corne, Marco Dorigo, and Fred Glover, Eds., pp. 11–32. McGraw-Hill, London, 1999.

[4] G. Di Caro and M. Dorigo, “AntNet: A mobile agents approach to adaptive routing,” Tech. Rep. IRIDIA/97-12, IRIDIA, Université Libre de Bruxelles, Belgium, 1997.

[5] R.S. Parpinelli, H.S. Lopes, and A.A. Freitas, “Data mining with an ant colony optimization algorithm,” *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, August 2002.

[6] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conf. on Neural Networks*, 1995, pp. 1942–1948.

[7] N. Holden and A. A. Freitas, “A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data,” in *Proc. 2005 IEEE Swarm Intelligence Symposium*, 2005, pp. 100–107.

[8] L. Yin, X. Wei, Y. Sun, J. Wang, and Matthew J. Rosato, “A 3D facial expression database for facial behavior research,” in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, April 2007, pp. 211–216.

[9] P. Ekman and W.V. Friesen, *Facial Action Coding System (FACS): Manual*, CA: Consulting Psychologists Press, Palo Alto, 1978.

[10] J. Wang, L. Yin, X. Wei, and Y. Sun, “3D facial expression recognition based on primitive surface feature distribution,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1399–1406.

[11] C. Mandal, H. Qin, and B. C. Vemuri, “Novel FEM-based dynamic framework for subdivision surfaces,” *Computer-Aided Design*, vol. 32, no. 8, pp. 479–497, 2000.

[12] J. L. Bentley, “K-d trees for semidynamic point sets,” in *Proc. Sixth Annual Symposium on Computational Geometry*, 1990, pp. 187–197.