3D Face Landmark Labelling

Clement Creusot
University of York
Department of Computer Science
York, U.K.
creusot@cs.york.ac.uk

Nick Pears
University of York
Department of Computer Science
York, U.K.
nep@cs.york.ac.uk

Jim Austin
University of York
Department of Computer Science
York, U.K.
austin@cs.york.ac.uk

ABSTRACT

Most 3D face processing systems require feature detection and localisation, for example to crop, register, analyse or recognise faces. The three features often used in the literature are the tip of the nose, and the two inner corner of the eyes. Failure to localise these landmarks can cause the system to fail and they become very difficult to detect under large pose variation or when occlusion is present. In this paper, we present a proof-of-concept for a face labelling system, capable of overcoming this problem, as a larger number of landmarks are employed. A set of points containing hand-placed landmarks is used as input data. The aim here is to retrieve the landmark's labels when some part of the face is missing. By using graph matching techniques to reduce the number of candidates, and translation and unit-quaternion clustering to determine a final correspondence, we evaluate the accuracy at which landmarks can be retrieved under changes in expression, orientation and in the presence of occlusions.

Categories and Subject Descriptors

I.4.8 [**Image Processing And Computer Vision**]: Scene Analysis

General Terms

Algorithms,Experimentation

Keywords

3D Face, Labelling, Graph Matching, Registration

1. INTRODUCTION

In anthropology, labelled points (landmarks) are used to compare the shape and size of individuals. Here a labelled point on one individual "corresponds" to the point of the same label on an other individual. In face processing, such landmarks are used to perform both preprocessing (cropping [16], registering, [17]), processing (morphing, mapping), and feature-based face recognition [10]. For a long time landmark labelling has been of secondary interest because it was relatively easy to have a high quality detection/localisation on the available databases containing only frontal non-occluded views. However, as the databases grow bigger and start to include pose variations and occlusions, the problem reappears. The features that are easiest to detect are the tip of the nose and the inner corner of the eyes because of their extremal curvature. However the inner corner of the eyes can easily be occluded by the nose bridge when dealing with orientation, or more generally by spectacles. The tip of the nose will be very rarely occluded, but the curvature attached to it can be seriously altered as part of the neighbourhood of the nose is missing (self-occluded) when processing a non-frontal view. This is a direct consequence of using single-viewpoint 3D sensors, which generate 2.5D data.

No matter how good a landmark descriptor is, there will always be cases where its value cannot be computed, because of missing data (occlusion) or noise in its local neighbourhood. Therefore, in the 3D face data of non-cooperating subjects, a landmarking approach should try to detect as many points as possible. Doing this with landmark-specific recipes leads to complex systems and training procedures, and in this practical sense, generic methods are more preferable. Here we present an approach, which aims to provide general labelling over a wide a region of the face, which is robust to pose variations and occlusions.

After a brief discussion of the state of the art in 3D face landmark detection, our landmark labelling system will be presented in section 3. The experiments to test the system and the results are described in section 4, and a final section is used for conclusions.

2. RELATED WORK

Automatic feature detection on 3D faces started almost 20 years ago but really became an important topic after 2000, when the 3D capture devices start to be more common, and when publicly available 3D face databases appeared. The great majority of feature detection papers deal with landmarks, and a very few with labelled lines or regions.

The problem of feature detection is closely linked to the registration problem. Very often selected points are used to constrain registrations and, conversely, registration is used to map a model on a surface to detect landmarks. In this paper, surface registration methods (the most common of which is Iterative Closest Point(ICP) [1]) are not considered because of their need for relatively clean data and good initial pre-alignment.

Most of the other methods of detection are landmark-

3DOR'10, October 25, 2010, Firenze, Italy.
Copyright 2010 ACM 978-1-4503-0160-2/10/10 ...$10.00.

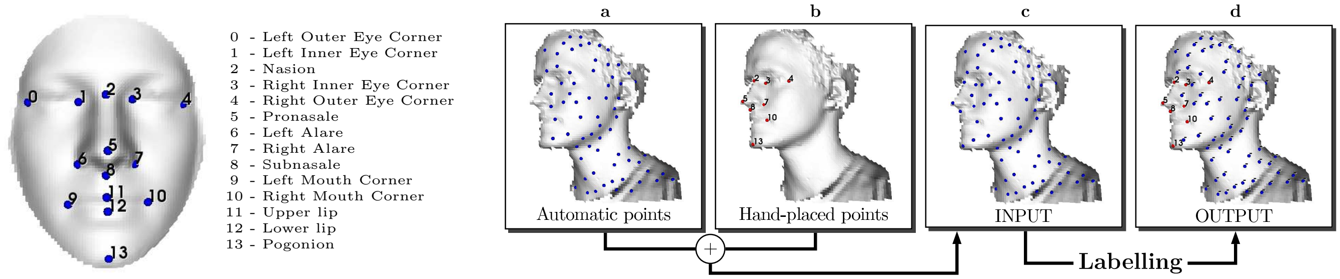

Figure 1: Problem to solve: Given a model and a set of points containing some hand-placed target landmarks, can the labels of the hand-placed landmarks be retrieved? Does it work when large parts of the face are missing? Here the automatic points are detected by taking the points that locally maximise the seeding score for any of the landmarks in the model.

specific. Database-dependent hypotheses are sometimes made, for example that the tip of the nose coordinate has a high value along axis z [2][6], or an extremal value along x when large yaw rotation is present [4].

Most of the methods use, at some point, the notion of local curvature of the surface. The two most often used measures are the HK curvature and the Shape Index [13].

A few methods use shape descriptors (for example Spin Images[12] or Ballon Images[17]) consisting of pose invariant shape histograms that characterize the local region around a landmark. However, the fact that the descriptors may employ extended local regions makes them sensitive to the significant self-occlusions that can occur at extreme head poses in 2.5D face data[17].

Some authors have noticed that the sagittal slice of the face remains identical over orientation changes and therefore can be used to detect the nose. In [8], contours of the mesh are extracted at varying angles until it matches a nose profile signature previously learnt. They reached 98.52% accuracy for the nose tip with variations of angle up to 90 °. Approaches not dealing with orientation have also used transverse slices to detect the nose and its corners [21][16].

In addition to [8], two other papers deal with orientation changes above or close to 45 °. In [15], candidates for the tip of the nose and the corners of eyes are detected using mainly curvature descriptors. As the three points selected can be on the same side of the face, it allows the detection of landmarks with orientation changes. The same authors use the directional maximum along varying direction to detect tip of the nose candidates [14]. Profiles passing through those candidates are then extracted and compared to select the tip of the nose.

The main drawbacks of many current methods are their dependence on specific landmarks being reliably extracted. Most of them will fail if the tip of the nose is not present or not well defined in the face scan data. A summary of recent literature is given in table 1.

3. 3D FACE POINT LABELLING

Figure 1 illustrates the problem to be solved. A set of 14 landmarks on the 3D facial surface forms the basis of our graph model. Our ultimate aim is to be able to detect these landmarks automatically. The problem is complicated by the fact that not all of the landmarks will be visible, due to pose variations caused by self occlusion and other occlusion. As a first step towards this, we want to demonstrate that if the *unlabelled* points that we want to detect are present within a large number of automatically detected points of interest, it is possible to relabel the landmarks correctly, as illustrated in figure 1. Note that the 14 hand-placed landmarks are not necessarily the ones that will be used in our final system, rather they are used as a convenient way of evaluating our labelling system.

Our labelling system is composed of a graph matching stage and a post-processing stage, which uses a scale-adapted rigid registration. The graph matching system consists of two graphs: firstly a fully connected (complete) model graph created from the 14 landmarks in all of the hand-labelled training data set, and secondly a fully connected query graph, generated from the (unlabelled) input points (box c in figure 1). Both of these graphs are attributed, with vertices having N descriptors (eg. curvature) and edges having M descriptors (eg. Euclidean distance).

In more detail, the attributes for each node in a graph are computed using a local spherical neighbourhood of radius $15\,mm$ within the face scan (training or testing set). Most of the attributes are derived from the maximal curvature (k_1) and minimal curvature (k_2) over this neighbourhood and the full list of 5 nodal attributes is:

- Mean Curvature (H): $\frac{k_1+k_2}{2}$
- Gaussian Curvature (K): $k_1 k_2$
- Shape Index (SI): $\frac{1}{2} - \frac{1}{\pi} \arctan \frac{k_1+k_2}{k_1-k_2}$
- Rough Volume (Vol): Sum of the tetrahedron volumes from the centroid of the perimeter to all the triangles inside of the neighbourhood.
- Log Curvedness (LC): $\frac{2}{\pi} \log \sqrt{\frac{k_1^2+k_2^2}{2}}$

The attributes for the edges of the graphs are:

- Euclidean Distance (dist)
- Coarse Geodesic Distance (distG): shortest path on the mesh
- Ratio between the two attributes above (ratioEucliGeod)
- Difference between vertex properties at either end of the edge (ΔH, ΔLC, ΔVol, ΔSI)

The labelling system is composed of an off-line part where the graph model is trained (described later) and an on-line testing part. The online testing is divided into three main processes, as follows:

Table 1: Main recent 3D face landmarking papers and their test case.

Authors	Landmarks	Nose	Eye corners	Database	Size	Crop	Angles	Occl.	Time	Year
Chang,[2]	4	99.4%		FRGC v2	4,485	yes	–	–	–	2006
Lu,[14]	7	98%	–	MSU	300	yes	$0 \sim 45\,^\circ$	–	–	2006
Mian,[16]	1	98.3%	–	FRGC v2	4,950	no	–	–	–	2006
D'Hose,[6]	7	99.89%	99.47%	FRGC v1	277	yes	–	–	$16\,s$	2007
Segundo,[21]	6	99.95%	99.83%	FRGC v2	4,007	yes	–	–	$0.4\,s$	2007
Faltemier,[8]	1	98.52%	–	NDOff2007	7,300	yes	$0 \sim 90\,^\circ$	–	$< 1\,s$	2008
Dibeklioglu,[7]	7	95.61%	96.28%	FRGCv1	943	no	–	–	–	2008
		99.87%	100%	Bosphorus	1,576	yes	–	–	–	
Romero,[19]	3	99.77%	96.82%	FRGC v2	4,013	no	–	–	–	2009
Szeptycki,[22]	9	100%	100%	FRGC v2	1,600	no	–	–	–	2009
Colombo,[5]	3	100%		UND	951	no	–	–	–	2009
	3	89.8%		UND	951	no	–	yes	–	
	3	90.4%		IVL	104	no	–	yes	–	
Pears,[17]	1	99.6%	–	UoY 3D	1,121	no	–	–	–	2010

1. Compute an attributed graph (the query graph) from the unlabelled input points on the facial scan, as described above.
2. Run a graph matching process using the generic model graph of the face. Here we determine initial candidates for each vertex and edge of the query graph. We then iterate "relaxation by elimination" [23] to reduce the number of candidates.
3. Select the best labels using a scale-adapted rigid registration. The method employed here is: (i) Select current best labels using thresholding on scores. (ii) Compute registration transformations using combinations of 3 points. (iii) Cluster rotational and translational components to determine a good scale-adapted rigid registration. (iv) Use this registration to determine the best label assignment.

The graph matcher we have developed is in fact a hypergraph matcher which implies that the processes applied to nodes can also be applied to the edges. The term "element" is used hereafter to generally refer to both vertices and edges.

3.1 Offline Training Process

A set of facial scans that are disjoint from the testing set are selected for training. Using this data, the statistical distribution of each attribute value associated with each element (node or edge) in the model graph is collected and modelled using a Gaussian.

To determine a matching score between an element's attribute *value* in the query graph and an element's attribute *distribution* in the model graph, a normalised probability density function is used, as follows:

$$Score(P^{Query}, P^{Model}) = \exp \frac{-(P^{Query}_{value} - P^{Model}_{\mu})^2}{2 * P^{Model}_{\sigma}{}^2} \qquad (1)$$

where P^{Query}_{value} is the value of the query attribute and P^{Model}_{μ} and P^{Model}_{σ} are the mean and deviation of the trained model attribute distribution.

Note that this equation relates to one attribute, yet an element is described by a N-dimensional vector of attributes. Thus we need a method of composing a match score over this multidimensional space. In order to do this, we find the best linear combination of attributes (for every node/edge in the model graph) that discriminates between elements of the same label and those of a different label. To do this, we apply Linear Discriminant Analysis (LDA) to the training data, an example is given in figure 2. Shown at the bottom of this figure is a blue vertical line that represents the seeding threshold. It is set such that at least 95% of matching landmarks in the training set are above it. When testing, this threshold is used to seed candidate labels for each query point.

The last parameters that need to be determined for the graph matcher are the thresholds used for the elimination decisions. For that, a simple heuristic is used: the thresholds are set to the maximal value that allows all training data to succeed.

The final part of offline modelling in our system generates a rigid face model used in our online post-processing stage. To retrieve relative coordinate positions from the set of statistics on pairwise Euclidean distances, a spring particle simulator is used. The landmarks are considered as particles having a random initial position. All of them are linked by springs with their equilibrium length equal to the mean distance between the two points they represent. The simulator runs until it stabilises in a coherent configuration which is used as the rigid face model.

3.2 Graph Matching

To recap, the graph matcher takes as input a query graph and a model graph and returns, for each node of the query, a list of probable candidate labels in the model with associated scores.

3.2.1 Seeding

First, scores for each possible association are computed by projecting the vector of normalised attribute scores into the LDA space. Only the model elements for which the score reaches a given threshold (blue line in figure 2) are added to the list of candidates.

3.2.2 Elimination rules

The graph matching heuristic used in this paper consists of a loop of elimination processes until the system stabilises. At each iteration (see figure 3), the less probable candidates are eliminated for each element. A candidate is thought to be improbable if its direct neighbourhood gives it very little support. Two kinds of support are considered, the number of matching neighbours and the score attached to those matching. In most graph matchers [3] the neighbours of a vertex will be other vertices. Here we use a hypergraph matcher which implies that the neighbours of a vertex are the edges connected to it and vice versa. The edge's scores help erase node candidates, and the node's scores help erase edge candidates.

The process is first run using static matching scores. Once

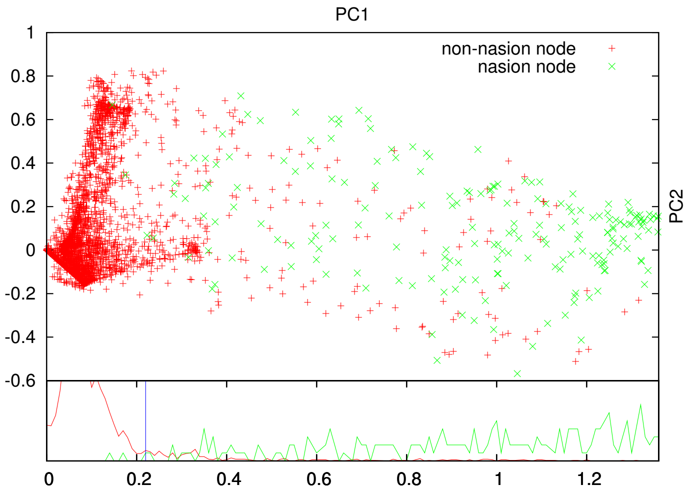

Figure 2: Example (for the nasion) of element correspondences (query-model) plotted in the LDA-reduced space of their score per attribute. The density function below show the separation of the two classes along the main component of the transformed space. The blue vertical line represents the selected seeding threshold.

Table 2: Variables definition

A	Element type $\in \{Vertex, Edge\}$
$\bar{A}$	$\in \{Vertex, Edge\} \setminus A.$
Q_i^A	i^{th} query element of type A
M_i^A	i^{th} model element of type A
$Cand(Q_i^A)$	list of model candidates M_j^A for Q_i^A
$Neigh(X_i^A)$	list of element $X_k^{\bar{A}}$ connected to X_i^A
$Score(Q_i^A, M_j^A)$	Matching score between two elements
$dynamicScore$	Allow score to be updated
$TreshSup(M_j^A)$	Learned Support Threshold
$TreshSco(M_j^A)$	Learned Score Threshold

the system stabilises and stops eliminating candidates, the scores are normalised so that their sum over the candidates of one element is equal to one. The support thresholds are replaced by the ones adapted to the dynamic elimination and then the flag *dynamicScore* is set to *true*. Once this set of iteration stabilises, the graph matching ends and returns the list of candidates for each element.

3.3 Final registration

The output of the graph matcher contains far less possible correspondences than the initial problem (see figure 5) but our objective is a one-to-one correspondence. To make the final decision, the rigid model of the hand-placed model landmarks is used.

Given a set of query landmarks and a set of model landmarks, the registration is defined as the 3D transformation that minimises the mean square distance between the points of corresponding label. A closed-formed solution to this problem is given in [11]. Hereafter, the registration that uses all of the landmarks of the model is referred to as "global" and the registration using a subset of three landmarks is called "triangle registration". The transformations are decomposed in three parts: scale, translation, and rotation.

```
Algorithm 1: Elimination
  For each $Q_i^A$ in Query graph:
      $totalScore = 0.0$
      For each $M_j^A \in Cand(Q_i^A)$:
          $support = 0; score = 0.0$
          For each $M_k^{\bar{A}} \in Neigh(M_j^A)$:
              $sup = 0; sco = 0.0$
              For each $Q_l^{\bar{A}} \in Neigh(Q_i^A)$:
                  If $M_k^{\bar{A}} \in Cand(Q_l^{\bar{A}})$:
                      $sup = sup + 1$
                      $sco = max(sco, Score(Q_l^{\bar{A}}, M_k^{\bar{A}}))$
              If $sup > 0$:
                  $support = support + 1$
                  $score = score + sco$
          $score = score/support$
          If $support < TreshSup(M_j^A)$
             or $score < TreshSco(M_j^A)$:
              $Cand(Q_i^A) = Cand(Q_i^A) \setminus M_j^A$ # Erase candidate
          Else:
              If $dynamicScore$: # Update score
                  $Score(Q_i^A, M_j^A) = Score(Q_i^A, M_j^A) * score$
                  $totalScore = totalScore + Score(Q_i^A, M_j^A)$
      If $dynamicScore$: # Normalise
          For each $M_j^A \in Cand(Q_i^A)$:
              $Score(Q_i^A, M_j^A) = Score(Q_i^A, M_j^A)/totalScore$
```

Figure 3: Elimination procedure called at each iteration of the graph matcher.

For the rotation part, a unit-quaternion representation is used.

Our hypotheses at this stage are the following:

1. A significant proportion of the best query candidates for each model landmark returned by the graph matcher are good matches.
2. The transformation that registers the whole model to the whole face is very similar to the transformation that registers a sub-part of the model (a triangle) with the corresponding sub-part on the query face.
3. Bad correspondences are unlikely to produce a coherent transformation: different sub-parts will be registered in different ways.

The smallest number of points to get a registration is three. However, when the triangle is very flat (close to a straight line), the transformation is less reliable. Triangles with an angle less than 15 degrees are discarded, which represents about 22% of the combinations.

For the clustering used to get the final transformation, a very simple approach is adopted. First the distances between all pairs of elements are determined. A binary distance tree is created by selecting the smallest distance between the already computed sub-tree and the rest of the points. At each step the distances involving the newly created sub-tree are updated using its new centroid. The distance tree is then cut using a distance threshold and/or a critical number of elements per sub-tree.

The final registration is used to assign the definitive labels by selecting the closest query point to each labelled landmark in the registered model.

3.3.1 Unit-Quaternion Clustering

While clustering Euclidean vectors (translations) is not problematic, clustering quaternions that lie on a 4-dimensional sphere requires some precautions. As two quaternions $\dot{q}$ and $-\dot{q}$ represent the same rotation, it is important to check that the dot product between two compared quaternions is posi-

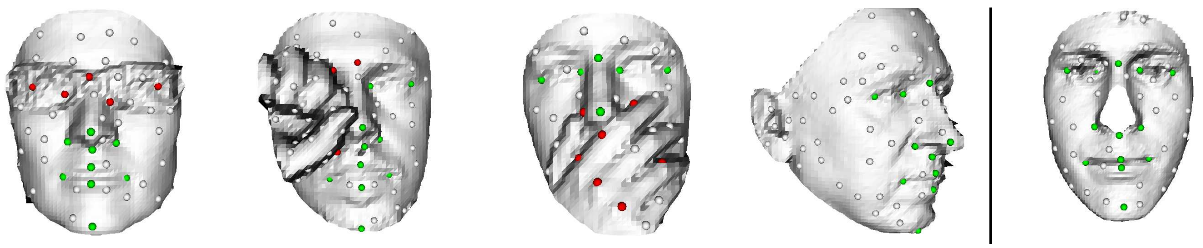

Figure 4: Example of results using the Bosphorus database and one using the FRGC database (right) where the nose is missing. Green dots represent good matches, red dots, false positive.

tive, otherwise we multiply one of them by −1. The metric used for the clustering is the following:

$$d(\dot{q}_1, \dot{q}_2) = \theta = 2 \arccos(\dot{q}_1 . \dot{q}_2) \qquad (2)$$

Finally, we determine the centroid of a subset of quaternions on the 4-sphere. In [9], Gramkow used the Taylor series of $\cos \frac{\theta}{2}$ to approximate the distance for small angles. Under those assumptions, he proved that the centroid is equal to the normalised Euclidean barycentre of the unit-quaternions coordinates. This allows us to compute the unit-quaternion clustering in a simple way.

4. EVALUATION

To test the labelling, a set of hand-placed landmarks are unlabelled and mixed with other automatically detected points that are likely to be landmarks according to their LDA projected scores. The percentage of points found to be the same as the hand-labelled points is used to evaluate the system.

4.1 Database

Tests are performed on two databases. The first one is the full FRGC v2[18] where 200 faces of different individual are selected as our training set and all the rest as a test set (4750 faces). In this proof-of-concept, the faces have been cropped using a 100 mm sphere around the nose prior to the experiment. (We appreciate that this is a slightly easier problem than using the full scan of head and shoulders, and we will address these larger scans in future work.) The landmarks used are a mixture of contributions from [22] and [19].

The second database is the Bosphorus database [20]. It contains 4666 captures of 105 people. Unlike the FRGC, it contains variations in pose and occlusions (hand, hair and spectacles partially covering the face). Our experiments are performed on different subsets, as follows:

N	Neutral Expression
E	Happy/Sad/Surprise/Anger/Disgust
AU	Action Unit Expressions [20]
O	Occlusions (hand, hair and glasses)
YR45	Yaw Rotation 45 ° Right
YL45	Yaw Rotation 45 ° Left
PR	Pitch Rotation up and down
CR	Cross Rotation Pitch+Yaw

In both databases, some landmarks have been detected (eg. subnasale) or refined automatically (eg. alares, nasion) and checked manually.

The automatically detected points on the query scan are extracted by selecting over a small neighbourhood (radius 15 mm) candidate points that maximise the seeding score for any of the landmark in the model. When a candidate is on the border of the neighbourhood, it is not selected. When combining automatically detected points and hand-placed points, the automatic ones that are less than 10 mm away from a hand-placed point are erased.

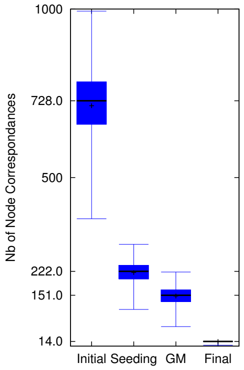

Figure 5: Number of node correspondences at different stages of the system computed on the FRGC test set. At the beginning, after the seeding, after the graph matching, and at the end.

4.2 Results

In controlled conditions on cropped inexpressive frontal faces, our generic system labels the landmarks with an accuracy of 93.8% with higher score for the tip of the nose (97.0%) and subnasale (96.5%), see table 3. The points that are more difficult to label are the outer corner of the eyes and chin (pogonion). This is not surprising as those points don't have a very discriminating shape and can be easily misdetected if the final registration is not perfect.

On a more challenging database like the Bosphorus, we notice that our system does not give as good results as landmark specific techniques (see Figure 4). However, it performs quite consistently when occlusion and change in pose are considered, when most existing techniques will fail with such input. It should be emphasised that the hand-placed landmarks are not necessarily correlated with geometric extrema in this experiment, if the set of target landmark is determine using such saliency, better results may be achieved.

Another interesting discovery is that the subnasale point, which is almost never used in automatic landmarking, is one of the most easily detected point in our experiments.

The time performance of the system when the graph has an average of 50 nodes is 0.67 s for the graph creation, 0.45 s for the graph matching and 0.04 s for the post-processing. The fact that complete graphs are used is costly when the number of nodes increases. Look at different graph topologies (tessellation, etc) can help solve this problem.

Table 3: Results per landmark (0-13) on the FRGC v2 database. The test set is split in two subparts: Neutral(-N) and Expression(-E), shown in the first two rows. The third row shows these results combined.

Train	Test	0	1	2	3	4	5	6	7	8	9	10	11	12	13
train(200)	test-N(3108)	**90.1**	**94.9**	**95.2**	**94.2**	**86.7**	**97.0**	**96.1**	**95.8**	**96.5**	**94.7**	**93.2**	**95.8**	**93.8**	**90.4**
train(200)	test-E(1642)	77.7	84.8	84.7	84.6	74.2	87.8	86.1	85.6	84.6	73.0	71.4	81.1	73.5	68.7
train(200)	test(4750)	85.8	91.4	91.5	90.9	82.4	93.8	92.6	92.3	92.4	87.2	85.7	90.8	86.8	83.0

Table 4: Results per landmark (0-13) on subsets of the Bosphorus database.

Train	Test	0	1	2	3	4	5	6	7	8	9	10	11	12	13
N-train(99)	N-test(200)	**92.5**	**97.0**	**98.0**	**98.0**	**92.5**	**98.5**	**97.5**	**97.5**	**99.0**	**94.5**	**96.0**	**96.0**	**94.5**	**88.5**
N-train(99)	E(453)	77.0	87.1	86.5	87.6	75.0	90.2	88.9	88.3	87.1	68.4	65.5	81.6	71.0	55.4
N-train(99)	AU(2150)	84.8	92.2	91.7	92.5	86.5	92.7	92.0	92.2	91.2	75.7	72.2	82.5	74.1	65.5
N-train(99)	O(381)	68.9	78.8	74.5	78.9	69.9	83.4	82.6	82.7	84.8	82.4	81.5	84.8	81.0	73.0
N-train(99)	PR(419)	84.0	90.1	89.7	89.4	84.7	90.6	90.6	90.9	91.3	88.7	88.7	89.2	88.5	80.1
YR45-train(20)	YR45-test(85)	–	–	83.5	84.7	85.8	88.2	–	91.7	88.2	–	83.5	83.5	80.0	72.2
YL45-train(20)	YL45-test(85)	81.1	83.5	69.4	–	–	86.9	88.2	–	88.0	84.7	–	72.9	76.4	63.4
YR45-train(20)	CR(211)	–	–	70.1	72.3	70.1	77.6	–	79.5	77.7	–	72.0	75.3	71.5	65.2
Mean (3688-3984)		82.4	90.0	87.7	89.2	82.7	90.5	90.6	90.2	89.6	78.1	75.5	83.4	76.9	67.8

5. CONCLUSION AND FUTURE WORK

Our technique shows relatively good results on difficult test cases with changes in pose and occlusion. The main advantages of this technique is that it is landmark-independent and can learn several different models without human intervention and/or addition of specific rules.

The next step in our research will be to evaluate repeatability of automatic candidate detectors, and couple both point detection and labelling systems to produce an automatic landmarker. To our knowledge, this kind of global approach to face landmarking has never been evaluated before. While giving lower results than landmark-specific techniques for the tip of the nose and the eye corners, it is very promising for unconstrained 3D face landmarking when these points are difficult to localise. Finally our method may also be used to landmark other kind of objects such as bones, man-made objects and so on.

6. REFERENCES

[1] P. Besl and N. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] K. I. Chang, K. Bowyer, and P. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, Oct. 2006.

[3] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):749–764, 1995.

[4] D. Colbry, X. Lu, A. K. Jain, and G. Stockman. 3d face feature extraction for recognition. Technical Report MSU-CSE-04-39, Department of Computer Science, Michigan State University, September 2004.

[5] A. Colombo, C. Cusano, and R. Schettini. Gappy pca classification for occlusion tolerant 3d face detection. *J. Math. Imaging Vis.*, 35(3):193–207, 2009.

[6] J. D'Hose, J. Colineau, C. Bichon, and B. Dorizzi. Precise localization of landmarks on 3d faces using gabor wavelets. In *BTAS 2007*, pages 1–6, Sept. 2007.

[7] H. Dibeklioglu, A. Salah, and L. Akarun. 3d facial landmarking under expression, pose, and occlusion variations. In *BTAS08*, pages 1–6, 2008.

[8] T. Faltemier, K. Bowyer, and P. Flynn. Rotated profile signatures for robust 3d feature detection. In *Automatic Face Gesture Recognition*, pages 1–7, 2008.

[9] C. Gramkow. On averaging rotations. *International Journal of Computer Vision*, 42(1):7–16, Apr. 2001.

[10] S. Gupta, M. K. Markey, J. Aggarwal, and A. C. Bovik. Three dimensional face recognition based on geodesic and euclidean distances. In *IS&T/SPIE Symposium on Electronic Imaging: Vision Geometry XV*, 2007.

[11] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[12] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, August 1997.

[13] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image Vision Comput.*, 10(8):557–565, 1992.

[14] X. Lu and A. Jain. Automatic feature extraction for multiview 3d face recognition. In *Automatic Face and Gesture Recognition*, pages 585–590, 2-6 2006.

[15] X. Lu, A. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *Pattern Analysis and Machine Intelligence*, 28(1):31–43, Jan. 2006.

[16] A. S. Mian, M. Bennamoun, and R. A. Owens. Automatic 3d face detection, normalization and recognition. In *3DPVT*, pages 735–742, 2006.

[17] N. Pears, T. Heseltine, and M. Romero. From 3d point clouds to pose-normalised depth maps. *International Journal of Computer Vision*, 89(2):152–176, Sept. 2010.

[18] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. *Computer Vision and Pattern Recognition, 2005*, 1:947–954, June 2005.

[19] M. Romero and N. Pears. Landmark localisation in 3d face data. In *Advanced Video and Signal Based Surveillance, 2009.*, pages 73–78, Sept. 2009.

[20] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3d face analysis. In *Biometrics and Identity Management, 2008*, pages 47–56, 2008.

[21] M. Segundo, C. Queirolo, O. Bellon, and L. Silva. Automatic 3d facial segmentation and landmark detection. *Image Analysis and Processing, 2007*, pages 431–436, Sept. 2007.

[22] P. Szeptycki, M. Ardabilian, and L. Chen. A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking. In *BTAS*, pages 32–37, Sept. 2009.

[23] M. Turner and J. Austin. Graph matching by neural relaxation. *Neural Computing & Applications*, 7(3):238–248, 1998.