3D Landmark Model Discovery from a Registered Set of Organic Shapes

Clement Creusot, Nick Pears, Jim Austin
Department of Computer Science
University of York
{cc595,nick.pears,jim.austin}@york.ac.uk

Abstract

We present a machine learning framework that automatically generates a model set of landmarks for some class of registered 3D objects: here we use human faces. The aim is to replace heuristically-designed landmark models by something that is learned from training data. The value of this automatically generated model is an expected improvement in robustness and precision of learning-based 3D landmarking systems. Simultaneously, our framework outputs optimal detectors, derived from a prescribed pool of surface descriptors, for each landmark in the model. The model and detectors can then be used as key components of a landmark-localization system for the set of meshes belonging to that object class. Automatic models have some intrinsic advantages; for example, the fact that repetitive shapes are automatically detected and that local surface shapes are ordered by their degree of saliency in a quantitative way. We compare our automatically generated face landmark model with a manually designed model, employed in existing literature.

1. Introduction

Often, correspondences are sought between an objects's 3D scan and some generic model of the object class, which has semantic labels. A typical problem of this type is a landmarking problem where, for example, a set of $(position, label)$ pairs are found on an input scan, such that they correspond correctly with the associated points on the model. This is the kind of problem that we are concerned with in this paper, when applied to 3D meshes.

The correspondence search process is usually broken down into a number of stages, namely (i) keypoint detection, (ii) keypoint description and (iii) matching. The detection process aims to take a large, dense set of points over the whole object and produce a much smaller, sparser set of interest-point detections, also known as *keypoints*. In general, this step makes the correspondence search computationally tractable and focuses attention on areas that are different from their neighbors, at least on some local scale. This brings us to the concept of *saliency*, which is broadly defined as something that is locally distinctive, or pronounced, on the shape in question. We acknowledge that the term *saliency* has been use in other works before, but they refer to other definitions than the specific mathematical definition that we give in Sect. 3.4.

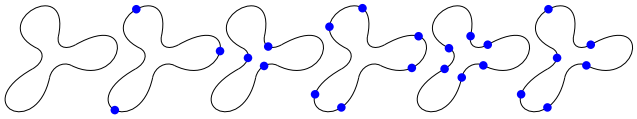


Figure 1. Example of an organically-shaped object and several arbitrary sparse sets of landmarks that can be used as models.

For a wide class of 3D objects that includes those with soft organic shapes (*e.g.* the human face), it is not always clear how to design a keypoint detector, because it is not always clear what the most distinctive surface points are in the first place. Typically, for human faces, extrema of Gaussian curvature have been detected, but this yields a very sparse set of useful keypoints. Others [4] have defined what they believe to be a useful set of locations at which to learn local shape properties, but there is often no obvious geometric justification for these. In many cases it has more to do with the existence of words to describe these locations and the words may even relate to color-texture properties (*e.g.* 'corner of eye') than geometric properties of the 3D mesh.

1.1. Problem Definition and System Outline

In existing literature, landmark models are heuristically-designed and often appear to be arbitrary. In figure 1, we see that an organic shape can be landmarked in a large number of ways. The key question that we address is: *how do we automatically choose a set of points that constitute a good symbolic model for some object class*?

A central problem is that the choice of model points ('landmarks') and the choice of the functional forms of their detectors are inter-related. Therefore, one has the choice of either: (i) manually defining some detector function and computing an optimal set of model points, or (ii) manu-

ally defining a desired set of model points and computing some optimal detector functions. (In general, different model points will have different optimal detector functions.) In this paper, we recognize this inter-dependency and we seek a third way: the automatic computation of *both* the 3D model points *and* the detector functions. The value of doing this is that it may lead to improved landmark models that give faster and more robust model fitting on previously unseen test scans of 3D objects.

Note that our system requires a set of shapes that have been registered by some means: usually some form of iterative closest points (ICP) [2], such as the deformable approach [1] used to develop the Basel Face Model [9]. (This may seem like a circular argument, because the main use of our landmarking output is to register shapes, but this work is a first step in designing a fully automated and optimized closed loop landmarking and registration process.)

In this article, our approach is applied to the specific object class of human faces. This is because we have access to registered face data and dealing more efficiently with human faces has huge potential applications in both industry and academia (*e.g.* face recognition, expression recognition, human-machine interaction). Having said that, the ideas presented in this paper can be applied other classes of organic objects where selecting a set of landmarks to form a model is not straightforward, for example bones or animal faces. However, without access to the appropriate registered datasets, we can make no claims about how well our model discovery process will work for these cases.

Figure 2 shows an outline of model-based 3D shape landmarking systems, where we try to emphasize the difference between what is presented in this paper (blue background in the figure) and what is presented in existing papers in the literature (red background). In this work, we present a model discovery system that tries to optimize the selection of a sparse set of target points (landmarks), whereas, in existing work, the model is manually designed. After we have run our automatic model generation process, we can check whether our framework selects similar locations that are usually selected in manual landmarking processes, such as those used in supervised machine learning scenarios or even manual anthropometry.

In the next section, we present relevant literature. We then present our framework and the datasets used for our experiments. This is followed by results and a comparison between automatic and manually defined models.

2. Related Work

The concept of saliency and the terms *salient point*, *salient region* and *salient object* is defined in many different ways in the literature. Perhaps the most well-known definition is that of Lindeberg [8], where salient points are found in 2D images as local extrema of the difference of Gaussian

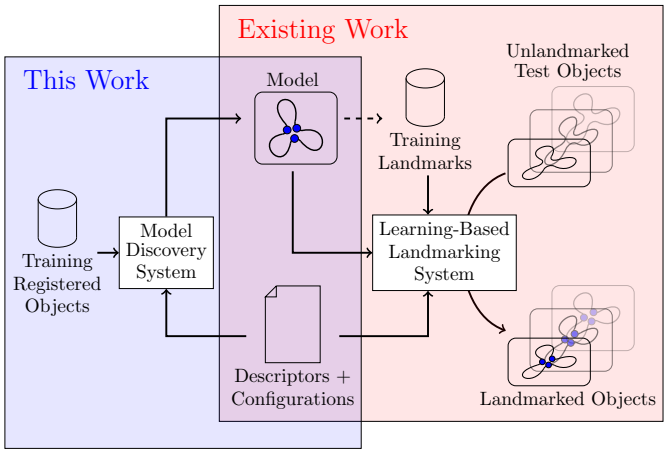


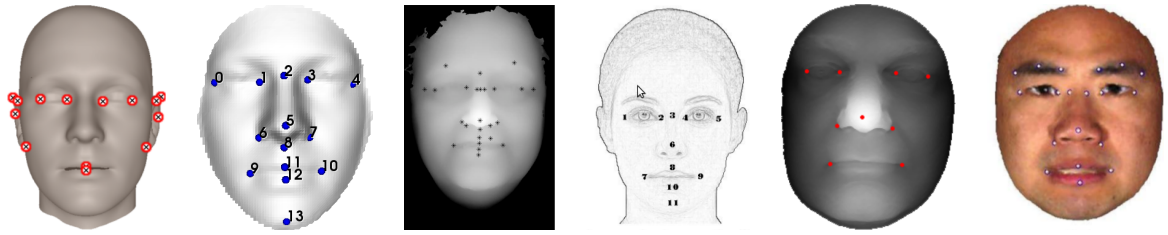
Figure 2. Workflow of a landmarking system. In existing systems (red box), the model is heuristically designed. In contrast, our system (blue box) tries to generate an optimal model for a given landmarking system. Such a model should improve robustness in landmarking instances of previously unseen test objects (right).

(DoG) filter. This approximates the scale normalized Laplacian of Gaussian (LoG) applied to an intensity image. This DoG approach is effectively a multi-scale feature detector that has also been applied to 3D meshes in various ways; for example [3] and [14]. An early discussion of saliency was presented by [6], who describe how a saliency map can be built for 2D images in the context of visual attention. Colors, intensities and orientations are considered over a variety of scales and using center-surround differences (differences between fine and coarse scale), a single saliency map is ultimately generated to guide visual attention.

Thus, although there have been a number of previous studies on 2D image saliency, for example in visual attention studies, image compression and so on, there are very few works that discuss general notions of saliency on meshes, beyond the design of keypoint detectors (eg. DoG-based) and other feature extractors, such as high curvature ridge lines [13]. However, one such paper in this category is that of Lee et al. [7], who define a scale-dependent mesh saliency measure using a center-surround operator on Gaussian-weighted mean curvatures. The motivation for doing this was to provide information for mesh simplification and viewpoint selection and to provide mesh renderings that provide visually appealing results. Their approach is inspired by human perception where, for example, a flat region in the middle of a set of high curvature bumps can be detected as being salient, even though it has near-zero curvature.

3. Automatic Landmark Model Discovery

Our framework aims to learn what surface points can easily be retrieved from a set of registered 3D meshes of



[1] [4] [5] [11] [12] [15]

Figure 3. Examples of existing sets of landmarks (with citations) that are generic and sparse symbolic representations of 3D faces. These are used as *inputs* to landmarking applications. In most cases, the points have been chosen because they can be explained through language to a human operator, not necessarily because of their optimality for a given purpose. In this paper, we aim to find a generic model of 3D faces that is not arbitrarily defined, but is optimized for local feature localization and hence face detection.

a given object class, using a predefined set of local shape descriptors and a definition of vertex locality. Our input of registered 3D meshes is derived from scans of human faces. These meshes are in dense correspondence with respect to their vertices and are generated from either the Basel Face Model (BFM) [9] or the FRGC v2 dataset [10].

Two metrics are presented in this paper that are used to extract our model landmark positions: a *saliency metric* and a *ubiquity metric*. It could be argued that our metrics lack a 'ground truth' for their evaluation, but the essence of this work is 'model discovery': where we are using different types of evidence (saliency and ubiquity) to learn the landmark model in a data-driven way.

In the following three subsections, we outline three essential components of our system: (i) computation of a 'pool' of local shape *descriptor values*, (ii) computation of *descriptor scores* from these raw descriptor values, and (iii) learning optimal functions of such descriptor scores to serve as detector functions for local shapes. Before describing these components, we need to define some notation.

Notation. For some i, $(i \in \{1 \dots V\})$, we define $v_i^k, (k = 1 \dots N)$ as corresponding vertices across N registered training meshes [1]. We use the symbol v_i to represent the vertex indexed by i in the 'template model', $\mathcal{T}$, (a mean mesh) of all of the registered training meshes. Note that the landmark model $\mathcal{L}$ that we wish to extract is a sparse subset of this template model. Also, let d, $(d \in \{1 \dots D\})$ be the index of one particular local shape descriptor (eg. Gaussian curvature) in the pool of D descriptors [2] defined within our system and let $m_d(i_k)$ be that descriptor's value at vertex v_i^k, on training mesh $\mathcal{M}_k$.

3.1. Local Shape Descriptor Values

We do not wish to rigidly prescribe a descriptor or combination of descriptors, as those will often limit the degree of distinctiveness that can be achieved, according to some performance metric definition. Instead, we define a 'pool of D descriptors' as an input to the system. In the experiments presented here, $D = 8$ local shape descriptors are used, as follows: the first and second principal curvature (k_1 and k_2), the Gaussian curvature (K), the mean curvature (H), the Shape Index (SI), the log-curvedness (LC), the local volume (Vol) and the Distance to Local Plane (DLP). These local shape descriptors were computed using implementations provided to us, courtesy of [4].

In the first stage of our system we compute D raw *descriptor values* (magnitudes depend on choice of units) at every vertex, v_i^k, in the registered training data. These raw descriptor values describe the the local shape that is enclosed by a small sphere centered on vertex v_i (we use a 15mm radius).

3.2. Local Shape Descriptor Scores

The probability density function (pdf) of the N descriptor values, $m_d(i_k) \quad (k = 1 \dots N)$, which are associated with descriptor d at vertex v_i in the template model $\mathcal{T}$, can be learnt and approximated by a known function (eg. a Gaussian). Then, for any vertex, v_j^k, on any training mesh, a *descriptor score* can be computed:

$$S_i^d(j_k) = \frac{\mathrm{pdf}_i^d(m_d(j_k))}{\max_x(\mathrm{pdf}_i^d(x))}, \quad (0 < S_i^d(j_k) \leq 1), \qquad (1)$$

where j_k in the above equation indicates that the descriptor score is evaluated at vertex v_j^k.

This score represents how close the descriptor value at vertex v_j^k is to the modeled modal descriptor value at vertex v_i, with respect to descriptor, d. In the case where the modeled pdf is a Gaussian [3], the descriptor score is given as:

$$S_i^d(j_k) = \exp\left(-\frac{(m_d(j_k) - \mu_{i,d})^2}{2\sigma_{i,d}^2}\right), \qquad (2)$$

[1] The experiments presented in this paper use $V = 2000$ vertices in correspondence across $N = 200$ meshes.

[2] We avoid the term 'bag of descriptors' to avoid confusion with the well-known 'bag of features' method.

[3] We found that 7 of 8 local descriptors have distributions that are well approximated by a Gaussian. Only the *shape index* did not, and we found that this was well approximated by an inverse Gaussian distribution.

where $\mu_{i,d}$ is the mean of the descriptor values, $m_d(i_k)$, $(k = 1 \dots N)$, at vertex v_i within $\mathcal{T}$ and $\sigma_{i,d}$ is their standard deviation.

3.3. Landmark Score Functions

Potentially, any vertex v_i in $\mathcal{T}$ can be selected as a landmark. We seek functions of descriptor scores at v_i that make that particular vertex distinctive with respect to its surrounding surface. We call such functions *landmark score functions* and only the most successful of these will be selected for the final landmark model, $\mathcal{L}$. In order to show how the *descriptor scores* are combined into *landmark score* functions, we first need to define two labeled classes of vertex for some given vertex, v_i, in $\mathcal{T}$.

- The *neighboring* class is the set of close neighboring points that are so close that they are similar in shape to the input point. This set of vertices is within a Euclidean inner sphere defined by radius R_A and centered on v_i and is the union of such vertices across the whole of the registered training set.

- The *non-neighboring* class is the set of points surrounding the neighboring points that are used as a reference to which the neighboring class should be compared. This set of vertices is contained within a Euclidean spherical outer shell defined by the two radii R_B and R_C, centered on v_i and taken as the union of such vertices across the whole training set.

We have experimented with a variety of values for (R_A, R_B, R_C) and we usually use $(2, 10, 45)$ in units of millimeters.

In order to determine an optimized landmark score function, whose output values best separate the two classes defined above (i.e. neighboring and non-neighboring vertices), several methods can be used and we elected to use Linear Discriminant Analysis (LDA).

The basic idea is to compute a landmark score function, for every vertex, v_i, in $\mathcal{T}$. Given the two classes of vertex described above, centered on v_i, D descriptor scores are computed, where all D sets of distribution parameters, $(\mu_{i,d}, \sigma_{i,d})$, have been computed at vertex v_i. LDA computes the optimal linear combination of these descriptor scores that best separates the neighboring and non-neighboring classes, in terms of such descriptor scores referenced to the *descriptor value* distributions at vertex, v_i. Thus a vertex-specific linear combination of D *descriptor scores* is used as a *landmark score* function to optimally distinguish each vertex from its neighbors - i.e. each vertex, v_i, has a *different* landmark score function, $\gamma_i(.)$, where:

$$\gamma_i(\mathbf{S}_j) = \mathbf{u}_i^T \mathbf{S}_j = [u_i^1 \dots u_i^D][S_j^1 \dots S_j^D]^T, \qquad (3)$$

where $\mathbf{S}_j$ is a D-dimensional feature vector of descriptor scores at some vertex v_j and $\mathbf{u}_i$ is the unit vector extracted by the LDA process at vertex v_i in $\mathcal{T}$.

In the case of LDA applied naively, the results can be quite unstable, as only one value is generated for per vertex for all the training set. In order to have more meaningful local values, LDA is computed 20 times with different subsets of the training set. The final vector, $\mathbf{u}$, which linearly weights the $D = 8$ descriptor scores in Eq. 3, is the mean of these results.

What we need now is a way to determine whether only the close neighbors of vertex v_i are the points that look like vertex v_i, in terms of the landmark score at v_i, namely $\gamma_i(.)$. Thus we define two metrics aimed at extracting model landmark positions: the saliency metric and the ubiquity metric. These are discussed in the following two subsections.

3.4. The Saliency Metric

Essentially the LDA algorithm is finding the unit vector (direction) in D-dimensional *descriptor score* space that best separates the two vertex classes, neighboring and non-neighboring. For each neighboring or non-neighboring vertex, the LDA-derived descriptor score function projects that vertex's D-dimensional descriptor score down onto that unit vector and generates a normalized score between 0 and 1. If we can quantify the separation between the class-based distributions of these *projected descriptor scores*, then we have a measure of how well separated the distribution of 'close neighbor' detector response scores are with respect to their surrounding vertex scores, for the optimal linear landmark score function $\gamma_i(.)$.

Let p_i^0 and p_i^1 [4] be the distribution of these normalized $\gamma_i(.)$ scores for the neighboring and non-neighboring class respectively. For every threshold t set between $[0, 1]$, the following rates can be computed:

$$\begin{array}{ll} \text{True Negative Rate:} & TNR(t) = \int_0^t p_i^0(x)\,dx \\ \text{False Positive Rate:} & FPR(t) = \int_t^1 p_i^0(x)\,dx \\ \text{False Negative Rate:} & FNR(t) = \int_0^t p_i^1(x)\,dx \\ \text{True Positive Rate:} & TPR(t) = \int_t^1 p_i^1(x)\,dx \end{array} \qquad (4)$$

We define a *saliency metric* as $\Lambda(i) = 1 - g(p_i^0, p_i^1)$ where $g(p_i^0, p_i^1)$ is the oriented intersection of the score distribution of both classes. By integrating over all possible t, the global notion of the intersection I between the two distributions is defined as:

$$g(p_i^0, p_i^1) \quad = \quad \int_0^1 FNR(t).FPR(t)\,dt \qquad (5)$$

The saliency can therefore be expressed as:

$$\begin{array}{rcl} \Lambda(i) & = & 1 - g(p_i^0, p_i^1) \\ & = & 1 - \int_0^1 \int_0^t p_i^1(x).(1 - p_i^0(x))\,dx\,dt \end{array} \qquad (6)$$

[4] These colored symbols correspond to the red and green colored distributions shown in Fig. 4.

This saliency metric can be computed for every single vertex in the template model $\mathcal{T}$, providing us with a *saliency map* over the template's surface. This can be color-mapped for visualization, see Fig. 4, left color map.

3.5. The Ubiquity Metric

Detecting the local saliency is sometimes not enough, if the particular shape in question is commonplace on the object's surface. It is possible to have landmark score functions (detector functions) that are highly locally salient, but ubiquitous in their response: consider, for example, a spike detector over the surface of a sea urchin. An ideal model landmark should not only be locally salient, but also rare. Thus our second metric, a *ubiquity metric*, measures the output of the landmark score function, $\gamma_i(.)$, over the whole object surface, averaged over all registered training data.

Given a landmark scoring function $\gamma_i(.)$ for a given vertex v_i in $\mathcal{T}$, we compute the ubiquity sum function as:

$$U(i) = \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{V} \gamma_i(j_k) \tag{7}$$

In an ideal situation, unlikely to be encountered in practice, only one vertex, v_j^k per training mesh would trigger a non near-zero score for vertex v_i and this score would be 1, achieved whenever $j = i$. Therefore, a vertex perfectly suited to being selected as a landmark would have a ubiquity score close to unity. In reality, many vertices will have non near-zero values and the ubiquity score will be significantly higher than 1. However, this does provide a second method to find potential model landmarks, which is to select those that generate very low ubiquity scores.

We define a *ubiquity map* as the ubiquity metric values over the the full template model $\mathcal{T}$, referenced to some specific vertex's landmark score function, $\gamma_i(.)$. Since there are V vertices in each training mesh, we have V ubiquity maps. A video showing ubiquity maps referenced to every vertex, v_i in $\mathcal{T}$, is presented as supplementary material to this paper. (We emphasize that the color maps generated here are *not* local descriptors.)

3.6. Selecting a Set of Landmarks

Once a saliency map or ubiquity map has been constructed over the set of vertices v_i in $\mathcal{T}$, it is possible to extract local extrema (saliency maxima and ubiquity minima) as the landmarks that constitute our extracted model. As the desired output of such a system is a sparse set of points, the notion of locality should be taken into account at this stage. Furthermore, the object might present some repeated surface features for which we may only want one instance in the model, at least initially.

To avoid using parameters in this part of the system, a simple iterative scheme is implemented. Initially, we have one normalized saliency score map and we detect the single point with the globally maximum value on that map. We then compute the normalized landmark score function associated with that vertex over all vertices of the mesh (again we visualize this as a color map over the template's surface).

If the point was a locally unique point (*e.g.* the nose tip) only one high scoring region (colored as a blue patch on the landmark score map) will appear; if it was a symmetric point (*e.g.* an eye corner), two or more patches will appear. This map is subtracted from the first one and the resulting map is normalized. The second best shape of interest will then be the global maximum over this newly created map. By iterating in this fashion, the set of landmarks created will never contain similar (eg. symmetric) shape and will be relatively sparse.

It is important to note, however, that we may wish to build a configural model of landmarks using multiple instances of the similar local shapes, for example we would usually want to include both the left inner eye corner and the right iner eye corner in a configural model of the human face. In this case, we can compute the landmark score function for each extracted landmark and extract two (or more) strong local maxima from the score map. Indeed, we show such additional (symmetric) detections on the face for the saliency metric (Fig. 7) and the ubiquity metric (Fig. 8).

3.7. System Input Parameters

In addition to the training dataset, consisting of densely registered surfaces, our system requires a number of input parameters that will directly influence the results of our experiments. Here we describe all of them.

One obvious source of variation is the number of descriptors, their nature, and the parameters for their computation. An advantage of our system is that those descriptors do not need to be independent. The correlation between descriptors is taken care of in the LDA-based, two class separation process. Therefore, we do not need to test the system with different subsets of descriptors. The biggest set of descriptors will always give better results. The only concern is their computation time. In this paper, we use 8 different scalar local shape descriptors at a single scale. The neighborhood size is fixed to 15 mm, which previous studies have shown to be adequate for this set of descriptors for hand-placed landmarks [4]. We acknowledge that changing this scale and/or the pool of descriptors can change the nature of the detected landmarks. Our aim is to compare manual and automatic model landmarks; choosing the best set of D descriptors to employ within our framework and choosing the best set of associated scales over which to compute them is another problem altogether.

Another source of variation is the definition of the locality that defines the vertex classes. The differentiation of neighboring and non-neighboring vertices is done using Eu-

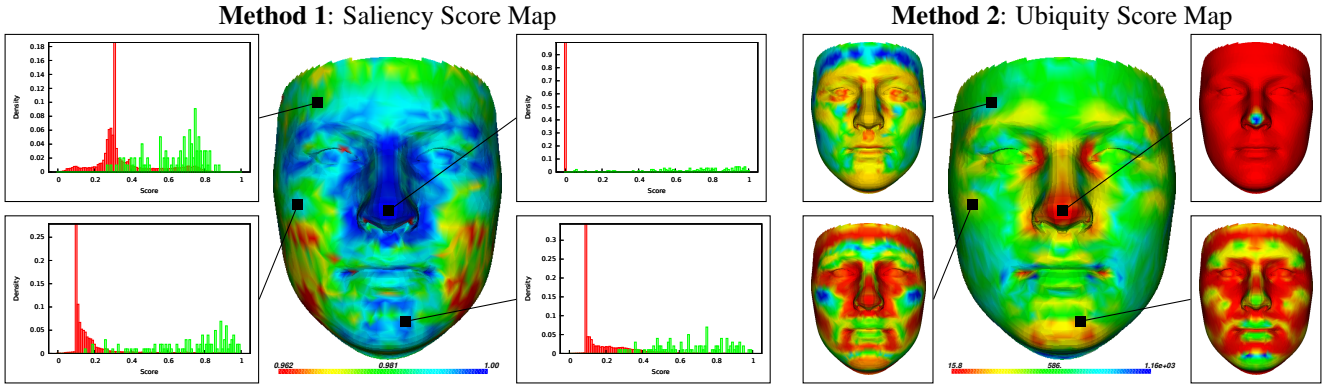


Figure 4. Resulting map for our two methods. For four random points we show, for the first method, a plot of the distribution of the scores associated with the matching neighboring vertices (green) and the non-neighboring vertices (red). The values on the central map are formed as the complement of the distribution intersection. For the second method, we show the associated response score map. The values on the central map are the sum over all vertices of the corresponding response score map.

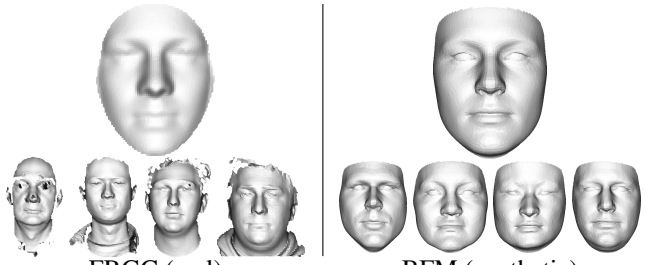


Figure 5. Top: template models, $\mathcal{T}$, derived from the FRGC (left) and BFM (right). Bottom: examples from the 200 training meshes associated with these templates.

clidean spheres and shells at different radii.

A further parameter of the system is the sparsity of the set of landmarks to be detected. The system should know how many landmarks it should be looking for and/or what is the minimum distance between landmarks that can be accepted. The number of landmarks can either be fixed or given as a ratio of the number of vertices in the registered training data. The minimum separation between features is given as a Euclidean distance. In this paper, we present results looking at a maximum of 10 landmarks, with a minimum distance of 10 mm between any two landmarks.

4. Datasets

Two different face datasets are used in our experiments (see Fig. 5). One is synthetic and has been generated from the Basel Face Model (BFM[5]) [9]. The second consists of real scans from the FRGC dataset [10], registered using ICP [2].

[5]Our BFM-derived meshes are not directly from individuals, but the BFM model itself was derived from real face scans.

BFM. We generated 200 random faces from the Basel Face Model (BFM), as well as a null-parameterized face to be used as a (mean) template model over which color maps are generated. To reduce the computation time, the model was cropped using a sphere of radius 100 mm around the nose tip. The inner mouth part was manually erased and the mesh resolution was reduced so that the number of vertices is 2000. The vertex indices on this lower resolution template model, $\mathcal{T}$, were used to reconstruct similar low resolution faces from the 200 meshes in the training data. Every vertex in our 2000 vertex template model has a correspondence in every mesh of the training set.

FRGC. For the FRGC, 200 random faces of different individuals were selected from the whole set. ICP-based registration was used to place cropped versions of the meshes into correspondence. The template model mesh, $\mathcal{T}$, was generated by averaging depth values. The generated mesh is approximately 2000 vertices. For every template model vertex, a correspondence is present in every training mesh by looking at the closest point in the (x, y) plane projection. Compared to the BFM, the registration is approximate for a single mesh. However it provides a good mean response, if enough meshes are present in the training set, and allows us to see if our method can be used without any supervision, as the correspondence computation was fully continuous (i.e. unlike the BFM, no manual anchor points were used for these FRGC registrations).

5. Results

Figure 4 shows maps computed using our two methods. A first comforting result is that both techniques for both datasets find roughly the same regions of the faces as interesting. While the first method finds locally salient points, the second finds globally rarer and more discrimina-

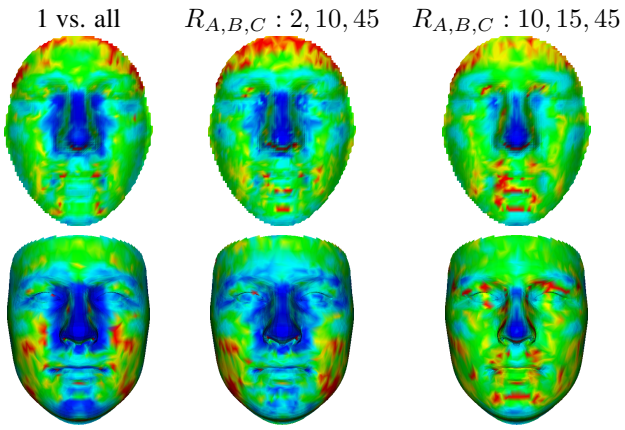


Figure 6. Saliency map for the two datasets with different locality definitions. Regions of maximal saliency are represented in blue.

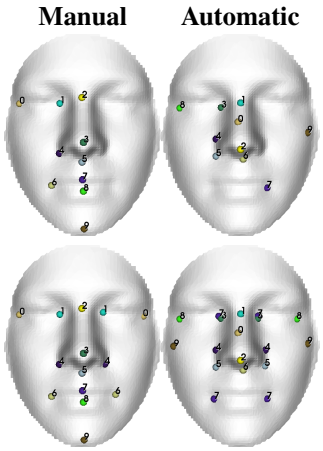


Figure 7. Top left: manual model set of 10 shapes used in the literature, repeated symmetrical shapes not shown. Top right: the 10 best local shapes detected by our automatic method, based on saliency. The numerical labels for the automatic method show the order in which the vertices are extracted, 0 being the most salient and 9 being the least. The bottom line shows the corresponding symmetrical detections (manual and automatic), extractable with the same detector function (landmark score function). With the automatic system, the correlations between the mouth and the eye shape are detected.

tive points. It appears that, on faces, points that are locally salient are also globally rarer. This is something that we humans find obvious in human faces, but that is not true in general; for example, recall the sea urchin. Unfortunately, we do not have access to registered datasets for several different classes of organic shapes.

Figure 6 shows examples of computed saliency map using different locality definitions. The example on the left of the figure is the extreme case, where only vertices $v_i^k,\ k = (1 \dots N)$ are considered as the neighboring class, and $v_j^k,\ (\forall j | j \neq i,\ k = 1 \dots N)$ is the non-neighboring class. The definition of the locality can not be optimized

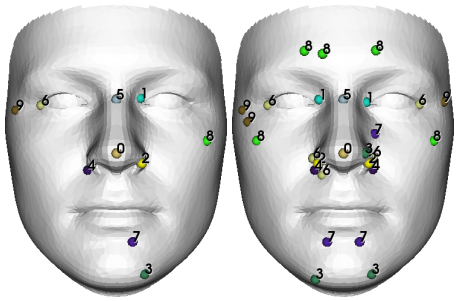


Figure 8. Left: automatic model set of ordered shapes of interest discovered with the ubiquity score map. Right: corresponding symmetrical detection.

within the system and has to be provided in input. For the remainder of this article we use the middle configuration ($R_A = 2$, $R_B = 10$ and $R_C = 45$). When selecting the maxima on this map, 10 shapes of interest can be defined and compared to the 10 manual landmarks commonly used in the literature. One advantage of the automatic detection of the shapes of interest is that the center of areas with similar shape can automatically be labeled.

In Fig. 7, the shapes of interest and corresponding symmetrical points are presented for both the manual and automatic sets of landmark. (Symmetric local shapes are extractable with the same detector functions as their symmetric counterparts.) When comparing both sets (see Fig.9), it can be noted that many of the coarse regions detected are similar in both solutions. For the ones that are different, it appears that the shapes selected by the automatic method have a ubiquity score far lower than the human ones, and are therefore more likely to be more easy to detect and label in a face landmarking system.

Since saliency and rarity seem to be correlated on faces, another way at looking at the problem is to try to find local minima of the ubiquity map. Figure 8 shows minima detected on the ubiquity score map of the BFM dataset and the detected centroids for the associated landmark score response map. Because the response map for the eye and the mouth corner are similar, our extrema detector does not detect the mouth corner as a shape of interest. It is also not detected when looking at the symmetrical shapes, as the response is not high enough at those points. Improved extrema detection techniques should be able to resolve this problem.

6. Conclusions and Future Work

We have presented a way to automatically extract a model set of landmarks to be used as shapes of interest from a registered training set. We discovered some similarities and some differences in the landmark selection made by our automatic process and those typically made by humans. The most important thing is the fact that the landmarks from

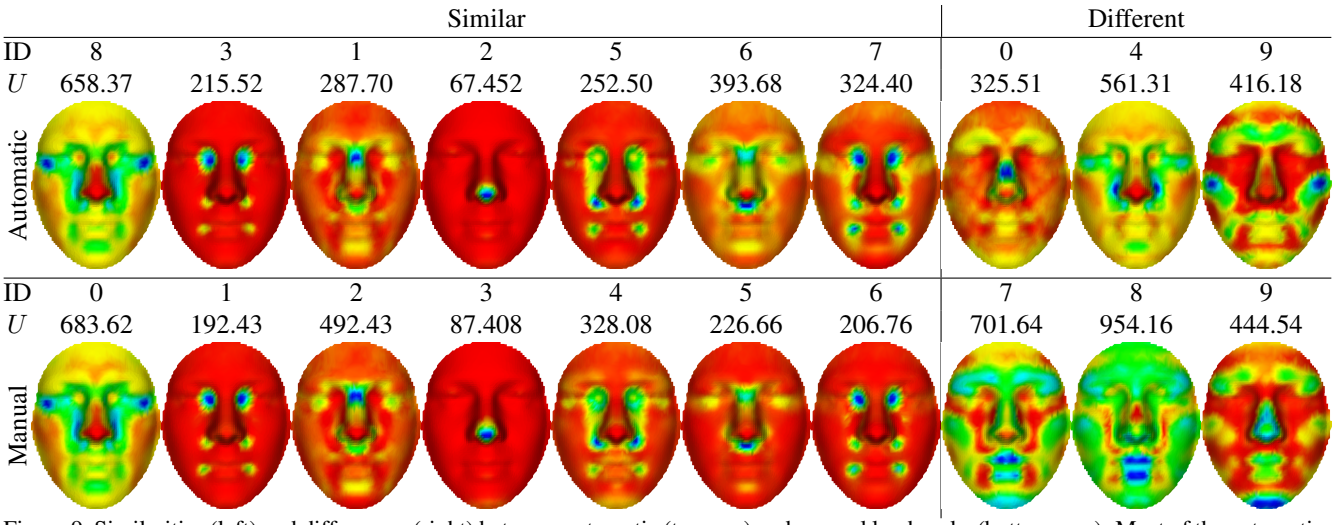


Figure 9. Similarities (left) and differences (right) between automatic (top row) and manual landmarks (bottom row). Most of the automatic landmark are very similar (in a qualitative way) to the ones picked by human specialists. However some of them are different. For the ones that are different, the definition of the automatic ones leads to tighter response maps (lower ubiquity score $U(i)$).

the automatic model, unlike the human ones, are optimized for a particular task with a given set of tools (local shape descriptors).

While our approach eliminates manual supervision for model landmark selections, it still makes two unsubstantiated assumptions. First, that points are the best things to detect on faces. Second, that one model should fit all faces. These were made to solve an otherwise intractable problem. Future work should try to challenge these two points. Different face shapes, ethnic groups, gender, age groups, might be associated with different optimal sets of landmarks. For example, some people have a shallow cup shape at the *ophrion*[6], while others have a perfectly monotonous (round or flat) forehead. The number and nature of the landmarks to be found on faces can vary a lot and it is likely that a highly flexible approach to 3D face landmark modeling is required.

References

[1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *IEEE Int Conf. CVPR*, 2007.

[2] P. Besl and N. McKay. A method for registration of 3d shapes. *IEEE Trans. PAMI*, 14(2):239–256, 1992.

[3] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Comp. Graphics Forum*, 27(2):643–652, 2008.

[4] C. Creusot, N. Pears, and J. Austin. Automatic keypoint detection on 3d faces using a dictionary of local shapes. In *3DIMPVT*, pages 204–211, 2011.

[5] 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 Symp. on Electronic Imaging: Vision Geometry XV*, 2007.

[6] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. PAMI*, 20(11):1254 –1259, 1998.

[7] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24:659–666, July 2005.

[8] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30:79–116, 1998.

[9] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *AVSS*, AVSS '09, pages 296–301, Washington, DC, USA, 2009. IEEE Computer Society.

[10] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face rec. grand challenge. *CVPR*, 1:947–954, 2005.

[11] M. Romero-Huertas and N. Pears. 3d facial landmark localisation by matching simple descriptors. In *IEEE Int. Conf. BTAS*, pages 1–6, 2008.

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

[13] K. Watanabe and A. Belyaev. Detection of salient curvature features on polygonal surfaces. *Comp. Graphics Forum*, 20(3):385–392, 2001.

[14] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *IEEE Int. Conf. CVPR*, pages 373 –380, 2009.

[15] X. Zhao, E. Dellandr anda, L. Chen, and I. A. Kakadiaris. Accurate landmarking of three-dimensional facial data in the presence of facial expressions and occlusions using a three-dimensional statistical facial feature model. *IEEE Trans. Syst. Man, and Cybernetics, Part B: Cybernetics*, 41(5):1417–1428, 2011.

[6] Situated at the center of the forehead.