RR-XXXX

Easy Rigging of Face by Transfer of Skinning Parameters

Ludovic Dutreve†, Alexandre Meyer†, Veronica Orvalho‡, Saïda Bouakaz†

†Université de Lyon, CNRS,
Université Lyon 1, LIRIS, UMR5205, F-69621, France.

‡Faculdade de Cincias da Universidade do Porto.

15/03/2010

Abstract Preparing a facial mesh to be animated requires a laborious manual rigging process. The rig specifies how the input animation data deforms the surface and allows artists to manipulate a character. We present a method that automatically rigs a facial mesh based on Radial Basis Functions (*RBF*) and linear blend skinning approach. Our approach transfers the skinning parameters (feature points and their envelopes, ie. point-vertex weights), of a reference facial mesh (source) - already rigged - to the chosen facial mesh (target) by computing an automatic registration between the two meshes. There is no need to manually mark the correspondence between the source and target mesh. As a result, inexperienced artists can automatically rig facial meshes and start right away animating their 3D characters. Last, we show how a rigged facial mesh is ready to be animated driven by motion capture data.

1 Introduction

Modeling 3D faces is becoming more and more automatic and common with systems based on photos [1] or on user-friendly interactions [2]. Consequently, many applications in the area of games or virtual reality offer to novices the capacity to generate or customize a 3D facial avatar. Indeed, avatar with better capacity to represent his owner in the virtual community will provide a better immersion. However, setup a facial mesh in order to animate it requires fastidious manual rigging to specify how the input animation data deforms the surface. We aspire a system that automatize this task to make animation more accessible for children, educators, researchers, and other non-expert animators. Notice that we address the missing portion of automatically rig a face, we do not target the motion transfer issue.

Since rigging fully automatically a facial mesh is already a challenging problem, we have chosen to focus on a traditional approaches for the real-time facial mesh deformation: Linear blend skinning (LBS) [3], also known as skinning or enveloping or skeletal subspace deformation [4]. Indeed, with Blendshape [5] which consists on morph between key meshes, they remains the most popular methods used in practice in realtime 3D engines. We avoid blendshape-based methods because manipulating and transferring the key shapes are far less tractable than skinning parameters.

Skinning binds controllers with a mesh in order to deform the mesh according to the controllers deformations. Usually, these controllers are bones or skeleton for the body and feature points ($\mathcal{FP}$) or landmarks for the face (as the ones defined by the Facial Definition Parameters of the MPEG4 standard [6]). A vertex may be influenced by several $\mathcal{FP}$. In this paper, we call skinning parameters: the $\mathcal{FP}$ and their influences on the mesh vertices (point-vertex influences or weights). In order to prove the concept of transferring parameters, we focus only on the common skinning technique, in the same way the *Pinocchio* system [7] did for automatically rig a character. Nevertheless, more accurate skinning techniques based on examples [8–10] may also be considered as an extension of our work as long as they are based on $\mathcal{FP}$ and on weights parameters.

Usually, the rigging process to create realistic and expressive human face animation means that an user has to manually place the $\mathcal{FP}$ on the face and specify which parts of the face are attached to which points. [11] shows that only the landmarks positioning can take around 2 minutes. We have designed our technique on usage simplicity, automatism and aim at provid-

Address(es) of author(s) should be given

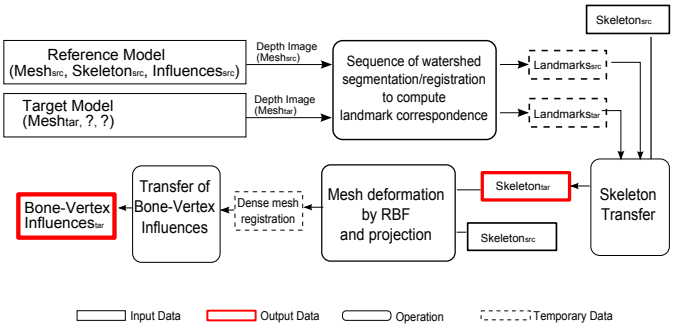


Fig. 1: Overview of our method. We rig a target mesh by transferring the skinning parameters from a reference mesh previously rigged (source). We firstly compute a landmark correspondence to transfer the $\mathcal{FP}$. Followed, by a dense registration to transfer the point-vertex influences.

ing real-time animation. We propose a method which takes as input a mesh of face (target) and provides by transfer as output a $\mathcal{FP}$ set and the associated point-vertex influences without human intervention. These skinning parameters are transferred from a reference mesh (source) already rigged. Comparing to previous transfer approaches focusing on face [12], our main contribution is the automatic registration which provides a landmark correspondence between the two meshes (See Section 5). Once a 3D target mesh is rigged, it may be directly animated by any $\mathcal{FP}$-based animations coming for instance from a simple motion capture system based on webcam as illustrated in our results (See Figure 9). Since our rigging is based only on common skinning, it may also be used by any 3D engines and major 3D Softwares like 3D Studio Max or Maya.

2 Related Work

Our problem is related to a wide range of sub-problems for which it is not reasonably possible to be exhaustive. First, we overview approaches working on semi-automatic rigging of characters. Second, since our approach is based on transfer we review some of surface registration methods which count a wide range of approaches.

2.1 Automatic Rigging

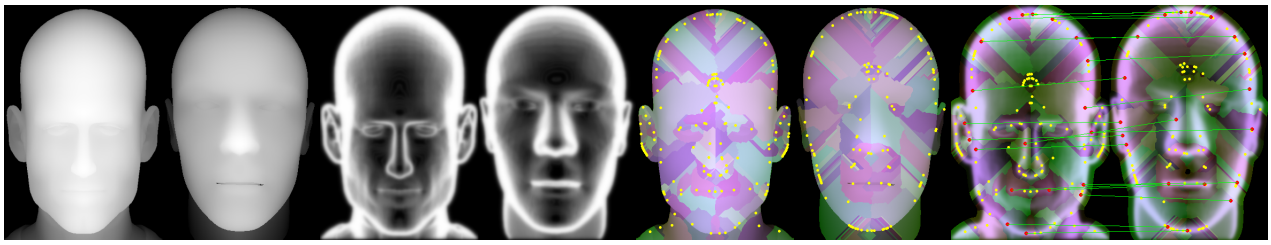
Previous research on character rigging (body or face) has mostly provided tools to help professional animators. For instance, Maya, 3D Studio Max or Blender software assign bone-vertex influences (weights) of skinning based solely on the vertex proximity to the bone and thus let the user to improve the result. Few methods have targeted fully automatic approaches except the *Pinocchio* system [7] proposing an easy, automatic rigging of 3D body characters based on a simulation of heat diffusion of each bone. Our work follows the same goal for the case of face than what *Pinocchio* proposed on character body. We differ by the fact we propose a transfer-based approach more suitable to face whereas they propose a more *procedural* approach dedicated to body. Indeed, for instance the case of the upper lip which are independent to the move of near $\mathcal{FP}$ of the lower lip are more easily treated by transfer. In the case of face, Costa-Teixeira *et al.* in [12] proposed a method dedicated to professional animators to transfer the rigging from one face to another after computing a registration based on landmarks manually positioned. Our approach follows the same principle but our registration provides automatically the landmark correspondence on both models avoiding manual intervention. Since we aim to create animations on the fly, automating the landmark positioning process becomes crucial.

2.2 Surface Correspondence and Registration

Finding a correspondence between two surfaces represented by meshes is central to various aspects of computer graphics: modeling from 3D scanner/range images, compression, information transfer, texturing or morphing. Given a source and a target surface, the goal of the surface registration is to find a transformation that optimally transforms points from the source surface to the target surface. Registration methods can take different forms depending on the nature of the data: static objects *vs* deformable dynamic objects, surface represented by range images *vs* meshes, manufactured objects *vs* human bodies *vs* faces. Our problem of automatically transfer facial animation parameters has to be classified on the problem of registering two different instances of static objects of the same family: variation of face.

Our approach is related to range images registration methods [13], often addressed by variant of the Iterative Closest Points algorithm [14], in the sense we compute the first part of the registration using a depth image representation of the surface which may be seen as a range image. Nevertheless, we differ because our input data are not two views of a same static object. The two depth images are frontal view of two different face models. Thus, even if they share the same nature of face they may slightly differ.

Non rigid registrations in the case of low deformation is a common problem in body/face motions cap-



(a) Depth images (b) Preprocessing (c) Watershed segmentation (d) Seed registration

Fig. 2: One step of our landmark based registration is as follow: (a) the two depth images of the meshes (which can be computed as a Z-buffer). (b) The depth images are preprocessed in order to raise specific features, here we have computed a Laplacian filter. (c) The transformed depth images are segmented by the watershed algorithm, yellow points are the watershed seeds (the local maxima). (d) Using the segmented regions, our algorithm registers watershed seeds and add them to the landmark list (red points). The algorithm iterates with different preprocessing on the depth images.

tures which may try to fit a template [15] model to scanned surface. In the case of face, methods [16,5] are often based on variational approaches minimizing an error criteria based on surfaces differences. [17] employs a morphable shape model, which is computed from a database of laser-scans faces. These approaches support the presence of holes and poorly sampled data and provide a quit good dense correspondence between the two surfaces. Nevertheless, their performances depend strongly on the similitude between the model and the database. Thus, for the sake of generality, we preferred working on an approach without *a priori* on the surfaces which allows to register two cartoon faces as illustrated on Figure 5 (c) and (d). Notice that in this context, it is often convenient [18,19] to work in the 2D parameter domain of the surface. Similarly, we base our registration on 2D representation to efficiently deal with meshes of various resolutions and to take advantage of several image-based descriptors computation.

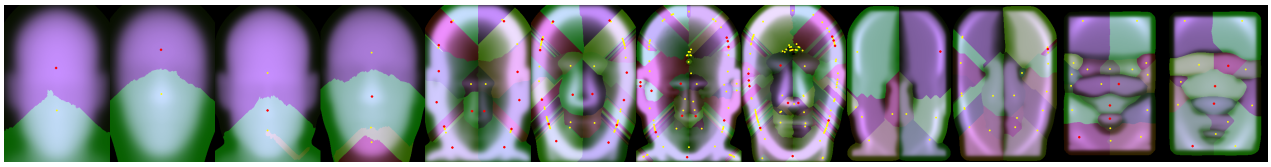
Registrations of surfaces with large variations is tricky to solve in a fully automatic manner. The case of articulated body with large variation of curvature seems more trackable as testifies the recent work [20] which works greatly and fully automatically on articulated models. For facial models, the problem often solved by human-input is conveniently defined in two levels. The large scale correspondence is often given by human-provided landmark points [21,12], on both models. And, the dense registration which defines the correspondence for each surface point is computed by low-deformation non-rigid registration based on radial basis functions *RBF* as described by Noh *et al.* in [22] in the context of deformations transfer. We use a similar approach which has proven to work for the dense registration but we automatize the large scale part where user provides landmarks.

3 Overview

Starting with a facial mesh, creating and placing by hand each component of the rig quickly becomes impractical when complexity grows. Our method automatically registers a reference facial mesh (source) to the facial mesh to rig (target) in order to transfer the rigging parameters with no human intervention. The reference facial mesh was previously rigged by the skinning method: $\mathcal{FP}$ and the point-vertex fitting are set. The transferred parameters are the $\mathcal{FP}$ positions and the point-vertex influences which link the mesh to the $\mathcal{FP}$. The result is a rigged facial mesh ready to be animated by skinning. The transfer is done in four phases of registration/transfer as shown on Figure 1. The method we present addresses the correspondence issue using a new approach registering the result of a sequence of watershed segmentations described on Section 5. Then, these landmarks are used to transfer the $\mathcal{FP}$ from the source to the target. The third phase (Section 6) computes a dense registration which associates each vertex of the target mesh to a surface point of the source mesh. And finally, the point-vertex influences are transferred.

4 Radial Basis Functions for Mapping Between Meshes

Both surface registration and skinning parameters transfer use Radial Basis Functions (*RBF*). *RBF* allow to compute a regression between two sets of scattered data. In our cases, *RBF* are used to define a transformation from a source to a target surfaces by interpolating source and target points sets. Let suppose we have two points sets, a source one X and a target one Y, with



(a) (b) (c) (d) (e) (f)

Fig. 3: A sequence of the watershed segmentation/registration process. The red points are registered watersheds seeds which are kept as landmarks whereas the yellow ones are not used. Each step corresponds to a preprocessing on the depth image. On (a) and (b), the preprocessing is a Gaussian filter with different radius. On (c) and (d), the preprocessing is a Laplacian filter followed by a Gaussian filter with different radius. On (e) and (f), the preprocessing is a vertical and horizontal anti-symmetric filters.

a correspondence between points: i^{th} point of X corresponds to i^{th} point of Y. The RBF are of the form:

$$\mathbf{y} = F(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{w_i} \varphi(||\mathbf{x} - \mathbf{x_i}||)$$

with $\mathbf{y} \in Y$ a learning point in the target space, $\mathbf{x} \in X$ a point in the source space, N the size of the both points sets, φ the radial function, $\mathbf{w_i}$ the weights for the i^{th} point and F the interpolation function we want to learn. Learning the interpolation (*i.e.* finding weights for each source points and for each axe) consists on solving three linear systems of N equations, if points are in 3D. [22] provides good explanations of how the system can be solved. More details about the use of RBF are given in Section 5 and Section 6. Intuitively, once the learning is computed we can use the RBF to transform a 3D point P_{source} from the source space to the target space, we note $P_{target} = RBF(P_{source})$.

5 Surface Correspondence and Feature-Points Transfer

The first issue of our transfer approach is the computation of the landmark correspondence on both meshes. This delicate task is in charge of registering main areas of the faces which may be tricky if the anthropometric lengths are slightly different. We propose a general framework on Section 5.1.1 which may be refined by any kind of surface descriptors as explained on Section 5.1.2.

To find this landmark correspondence, our approach runs a sequence of watershed segmentation/registration steps performed on depth images. We have chosen to register the two faces using a depth images representation for several reasons. We start from the observation that a frontal part of a face is 2.5D, so the depth image representation should keep the important information for registration. Depth image allows to be easily independent on the mesh resolution which is essential for the usefulness of the method. A low polygon facial model may be registered with a high polygon model. Comparing to complex remeshing, computing the depth map representation of a face can be done easily using for instance the result of a frontal projection (for instance the Z-buffer of an orthogonal rendering). Notice that cylindrical projection may also be used. And finally, working on regular 2D grid (image) is more convenient and more efficient in computation time than working directly on the mesh. It allows us to take advantage of several image-based descriptors.

Each segmentation/registration step adds new landmarks to the set. The landmarks registered are some seeds of the watershed segmentation regions. Figure 2 shows one step of preprocessing, segmentation and landmark registration. Since the segmentations are computed using the global image information, our method can localize the landmark correspondence more accurately than local detection/grouping methods as for instance the SIFT descriptor algorithm would have done be selecting only landmarks on the border (See Figure 5.(a)). The assumption of our method is that objects of a same family, in our case human face variation, will provide mostly similar segmentation. Then, segmentation is more easy to register as shown by our criteria described on Section 5.1.1 than registering directly meshes. Good arguments around this are given in [23] in the context of using segmentation for matching images.

Our approach runs iteratively a sequence of registration steps which differ by the preprocessing done on the depth images before the segmentation. This preprocessing consists of computing specific descriptor on the source and target depth images in order to combine the advantages of several descriptors. Each descriptor raises some particular locations of the two surfaces like edges or symmetric areas. Figure 3 shows the sequence of registration with different preprocessing. This preprocessing is related to the classical problem of com-

puting a good descriptors and may be augmented by an descriptor.

5.1 Automatic Landmark Correspondence Determination

At the beginning of the sequence, any landmark are defined. After each step, a *RBF* transformation is recomputed with the whole set of landmarks. Each step of the watershed sequence/registration adds some landmarks which are watershed seeds registered with a watershed seed of the other surface using the two segmented regions and the *RBF* computed at the previous step.

5.1.1 One Step of Watershed Segmentation

The concept of watersheds comes from the field of topography, referring to the division of a landscape in several basins or water catchment regions. Consider two oceans on both sides of a mountain. On rainy days, all the drops of rain that fall on one side of the mountain flow into one ocean, while rain falling on the other side of the division will flow into the other ocean. From this point of view, we can consider an image and in particular our depth image as a topographic surface [24] where each pixel is a point situated at some altitude as a function of its gray level. We will see in the next Section that the depth image may be preprocessed in order to bring up desired features before being used as topographic surface. To segment the surface represented by the depth image, we gradually immerse the surface in a water container. Previously, a hole has been made in each of the local minima. The water will begin to flow through the holes, first through those with less altitude but gradually reaching those with a greater altitude. So in the end every point will be assigned to a minimum and the surface will be divided into its catchment basins. The local minimum point of each basin is the representant of the basin we call the seed. In term of depth image, the seed is a 2D pixel position and each pixel is assigned to a watershed which is a segmented region of pixels.

In this section, we assume that k steps of the sequence of watershed segmentation/registration has already be done and has provided a set of landmarks. These landmarks define the transformation $RBF_k^{src->tar}$, just noted RBF, which transforms a point of the source image to the target image. By inverting the landmarks, we compute $RBF_k^{tar->src}$ noted RBF^{-1}. We consider the next step of the sequence, the $k+1^{th}$ watershed segmentation/registration. The watershed segmentation computes on the two depth images, two sets of regions with their associated seeds

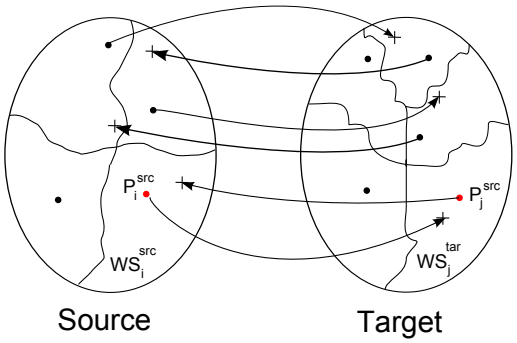
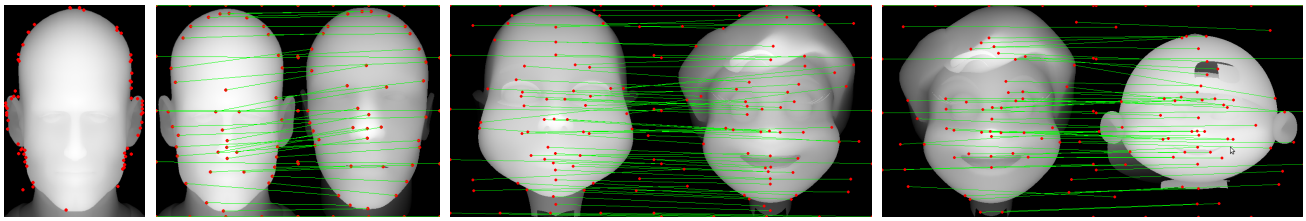


Fig. 4: During the step $k+1$ of our iterative registration, two segmented regions (watersheds) match if their seed falls into the converse watershed after being transformed by the RBF of the step k.

(See Figure 2). A seed of the source segmentation will be selected as landmark if it can be associated with a seed of the target segmentation. The correspondence criteria between two watershed is described on Figure 4. Two watersheds WS_i^{src} and WS_j^{tar} match if the seed P_i^{src} (resp. P_i^{src}) falls into the watershed WS_j^{tar} (resp. WS_i^{src}) after being transformed by the RBF (resp. RBF^{-1}) defined by the previous steps. *i.e.* if $RBF(P_i^{src}) \in WS_j^{tar}$ and $RBF^{-1}(P_j^{tar}) \in WS_i^{src}$ we add P_i^{src} and P_j^{tar} to the set of landmarks defining the correspondence. For instance, on Figure 4, the bottom right watershed regions match, top ones not. At the end of this step, we recompute the RBF with the added landmarks for the next step. This criteria is fast to compute and has the advantage to quickly registers similar regions. According to our tests, it is enough efficient to not considering an other criteria more complicated to compute.

5.1.2 Sequence of Segmentation as a Descriptor Problem

The watershed segmentation applied on depth image combined with the fast to compute criteria to register regions has the advantage to provide a global approach which can be easily iterated with different preprocessing. Indeed, with this approach our goal is to take advantage of various descriptors to catch different regions of a face, for instance regions of high curvature or of high symmetry, *etc.*. Thus, in this Section, we propose different preprocessing to apply on the depth images. which is related to the classical issue of choosing pertinent descriptors for image or surface registration [26, 13]. Since our approach is based on a sequence of registrations, as many descriptors as needed can be used. Nevertheless, the choose of descriptors order have to be



(a) SIFT descriptors (b) (c) (d)

Fig. 5: On (a), the landmarks selected by the SIFT descriptors algorithm [25], only on the border, are not enough accurate to guaranty a good registration. On (b), (c) and (d), our sequence of segmentations/registrations provides a set of 2D pair of landmarks on both depth images. The reprojection of these 2D landmarks on the 3D surface is used to define the 3D *RBF* transformation from the source mesh to the target one. (c) and (d) illustrates our registration on cartoon faces.

from coarse to fine, for instance see the progression of descriptor described on Figure 3.

The first idea of preprocessing is to apply Gaussian filter on the depth images to remove the issue of small regions during the segmentation. This will register the global appearance (local maxima) of the face, mainly the nose and often the forehead (See Figure 3 (a) and (b)). It is also possible to invert the depth in order to register local minima points like the ones localized on the border of the eye near the up of the nose. The Laplacian will register edges in general like border of the face or of the nose (See Figure 3 (c), (d) and (e)). Since a face included many horizontal or vertical symmetric regions, an interesting descriptor is the symmetric one.This list is not exhaustive and reader may define other interesting descriptors. In practice, our different steps of preprocessing are the ones described Figure 3 with which we have computed results of Figure 7.

5.2 Feature Points Transfer

The sequence of watershed segmentations/registrations previously described provides as output two sets of final landmarks defining a correspondence as illustrated on Figure 5. These data are used to map the two faces, they are used as learning points for a *RBF* interpolation $RBF_{final}^{src->tar}$ as described in Section 4. We use this *RBF* to transfer each $\mathcal{FP}$ position of the source face to the target face. Figure 7 illustrates this process on several targets. The error values describe the difference between the transferred $\mathcal{FP}$ positions and the manually positioning ones (See Section 7 for more details).

6 Transfer of Point-Vertex Influences

Rigging the target mesh with common skinning techniques requires to attach each vertex to one or more $\mathcal{FP}$ with influence values. As for the $\mathcal{FP}$, we perform this task by transferring this information from the source mesh.

We then define a dense registration which associates each vertex V of the target mesh to a point P of the source surface, meaning P is a 3D position on a triangle of the source mesh and is not necessary a vertex.

6.1 Dense Surface Registration

In order to do densely register the two meshes, we use the transformation $RBF_{final}^{tar->src}$ defined by the landmarks from the mesh to rig (target) to the reference mesh (source). Because the set of landmarks is not sufficient to reflect every details of the mesh, this transformation does not match precisely the two meshes, meaning it does not guaranty that all transformed vertices fall onto the surface (See Figure 8 on the left). Before computing a dense registration, we crop the target head to the face. The goal is to work only with the vertices of the face, since back-head vertices will not be influenced by facial $\mathcal{FP}$. Every removed vertices will be attach to the root head bone. This results in a decrease of computational cost, and avoid possible registration errors due to a lack of landmark points away from the face. The crop is done automatically, using the distance between the vertex and the nearest landmarks which has to be less than a distance dependant on the anthropometric measures of the face. In our case, we found empirically that the distance between the two eyes is a good value.

To get the dense registration, each vertex V of the cropped target mesh is transformed by the *RBF* interpolation: $V' = RBF_{final}^{tar->src}(V)$. Then, each interpolated target vertex V' is projected on the source mesh and falls into a point P on the triangle t_{src} which defines our dense registration as illustrated on Figure 8 on the right. Since a head geometry is somehow close

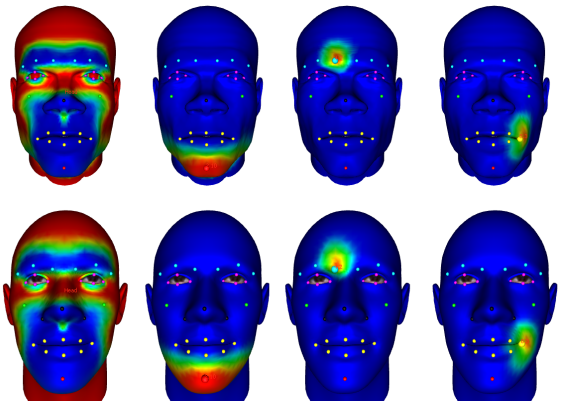


Fig. 6: The top row shows how some point-vertex influences on the reference face. The bottom row shows the transferred $\mathcal{FP}$ influences on the target face.

to a sphere, we apply a spherical projection with the head root bones as projection center.

6.2 Feature Points Influences Transfer

Each target vertex V is registered with a point P on a source triangle t_{src} defined by the three vertices: $t_{src} = (V_1, V_2, V_3)$. We compute the $\mathcal{FP}$ influences of V by interpolating the ones of the vertices V_1, V_2 and V_3 according to the position P on t_{src}. The interpolation is done with the Inverse Distance Weighting (IDW) :

$$IDW_{V_i \in \{t_{src}\}}(V_i) = \frac{1}{||V_i - P||} \times \frac{1}{\sum_{k=1}^{3} \frac{1}{||V_k - P||}}$$

$\mathcal{FP}$ influences values of t_{src} are blend using IDW coefficients to get $\mathcal{FP}$ influences of V. Figure 6 shows an example of influences transfer for some $\mathcal{FP}$.

Sometimes conflicts happened for vertices influenced by the wrong side of the mouth, for example, a top lip vertex could be influenced by a bottom lip points. This problem does not occur around the eyes if we assume that target faces are in a neutral pose, with opened yes. However, it could be solved in a same way than for the lips. For each vertex influenced by a lip $\mathcal{FP}$, we compute its geodesic distance with the current $\mathcal{FP}$ and with the mouth symmetric one. If this distance is lower for the opposite $\mathcal{FP}$ than for the current $\mathcal{FP}$, influence is swapped. Notice, that we assume that the target mesh is enough dense to be properly animated, otherwise a mesh refinement may be applied.

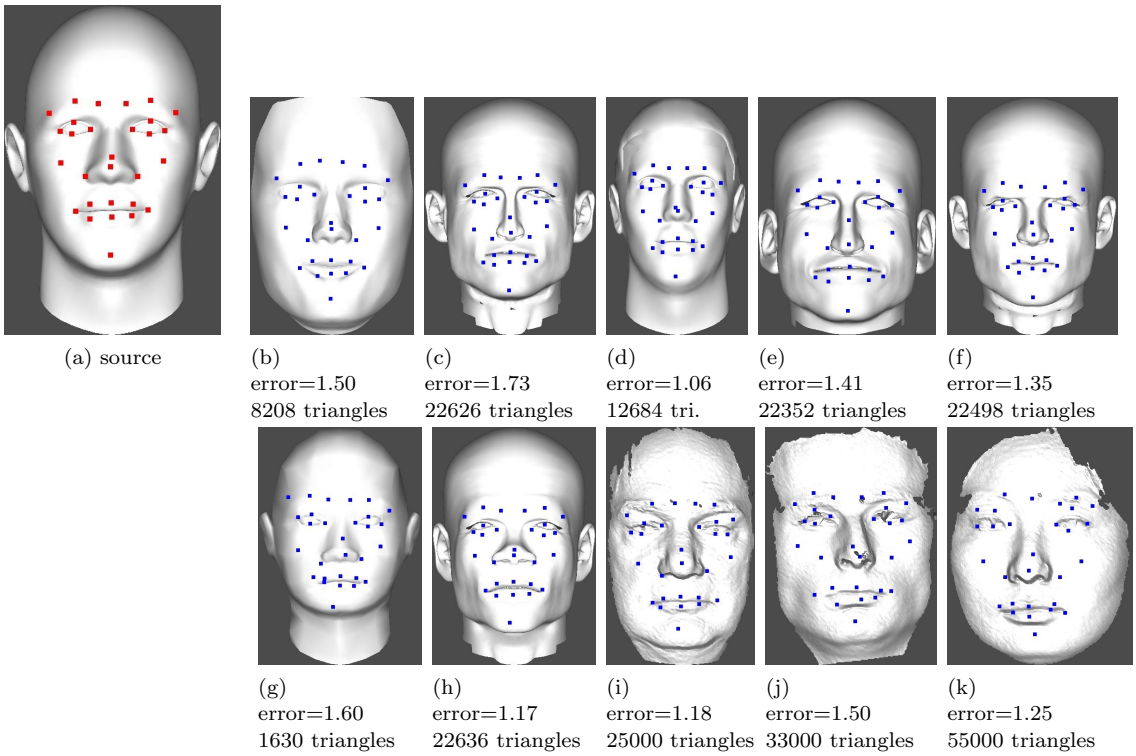
7 Results

We have tested our registration and skinning transfer on several facial models, which are human variation with a large scale of characteristics (See Figure 7). To illustrate the capacity of our transfer to deal with different mesh resolutions, the 3D mesh models count from about thousand triangles to several tens of thousand triangles. Some models are the whole head and some are only the face. Some models are computer generated, others are scanned faces (the three on the bottom right). The only constraint we ask to the models is that the face has to look in the Z axis direction to be able to compute the depth image of the frontal view. We also tested our registration on cartoon faces illustrated on Figure 5 (c) and (d). To measure the quality of the landmark correspondence computation, we have manually defined a $\mathcal{FP}$ set for each 3D mesh. These sets are composed by twenty points set on representative positions for animation: around the nose, the mouth and each eyes (see red points on Figure 7.(a)). We define an error criteria (see error values on Figure 7) by summing the distance between each $\mathcal{FP}$ of the transferred rig and the manually set ones normalized by the diagonal $Diag_{bbox}$ of the manually set ones bounding box:

$$error = \sum_{i=1}^{n_{\mathcal{FP}}} \frac{\mathcal{FP}_i^{manual} - \mathcal{FP}_i^{transferred}}{Diag_{bbox}}$$

Notice that the result of the automatic $\mathcal{FP}$ transfer may be manually edited if the user feels the need, without changing the rest of the method. Our results on Figure 7, 9 and on the provided video are not manually adjusted.

Computing the depth image representation of the mesh is instantaneous using the GPU capacity by using the Z-buffer of the frontal view. The landmark correspondence determination is computationally dependant on the depth image resolution and on the number of steps in the sequence of segmentation/registrations. Usually in our tests, we used depth images of 512×512 with the segmentations/registration described on Figure 3. Landmarks computation takes about 20s on a Athlon X2 3800+ with 2Go of RAM. The dense registration is computationally dependent on the number of triangles of the two meshes. Our implementation, not optimized, takes for instance about 3s for a models with 1800 triangles or a minute for a mesh with 12000 triangles. Once the dense registration is done, the skinning parameters transfer is instantaneous. And finally to illustrate our results, Figure 9 shows an animation applied on faces automatically rigged by our method in a minute without human intervention. The animation is provided on the fly by a motion capture system using



(a) source

(b) error=1.50 8208 triangles

(c) error=1.73 22626 triangles

(d) error=1.06 12684 tri.

(e) error=1.41 22352 triangles

(f) error=1.35 22498 triangles

(g) error=1.60 1630 triangles

(h) error=1.17 22636 triangles

(i) error=1.18 25000 triangles

(j) error=1.50 33000 triangles

(k) error=1.25 55000 triangles

Fig. 7: Examples of $\mathcal{FP}$ set transfer. The mesh on the left is the source (reference) model which was previously rigged including setting of the red $\mathcal{FP}$. On the right, the blue $\mathcal{FP}$ are automatically transferred on the mesh to rig from the source mesh. To measure the quality of the registration we have manually set the $\mathcal{FP}$ on the target meshes and compute an error value as describe in Section 7

a simple webcam and the Lucas-Kanade [27] computer vision marker tracker. Notice that any other motion capture system may be used and that the markers on the face may be different than the $\mathcal{FP}$using retargeting adaptation [28].

8 Conclusion

We have proposed a fully automatic method to rig and animate a facial mesh by transferring the $\mathcal{FP}$ and the skinning influences from a reference mesh previously rigged. We have tested our approach on several meshes and provided an error criteria to evaluate the quality of the landmark correspondence determination which is the most sensitive part of our approach. Animations of faces rigged by our method prove the viability of such an automatic transfer approach. By using depth image representation of the mesh we have been able to use image approach to 3D problem, which make the implementation more easier, let use several descriptors with an approach based on segmentation, and allows to easily deal with mesh of different resolutions. Indeed, running a re-meshing, several preprocessing and segmentation on two input meshes would have taken more computation time if we have kept a classical mesh representation.

As future work, we would like to extend the evaluation of our approach by computing an error more dedicated to animation. For instance, we could measure the difference between the transferred point-vertex influences to the ones defined by an artist. Although registering faces of the same class is possible, fully automatic transfer of rigging from a generic reference human face to a large range of faces like cartoon or animals is still delicate because of the landmark correspondence issue. A solution would be to rig several types of reference faces (human, cartoon, animal, *etc.*) as preprocess and when a new face has to be rigged the transfer could be done from the nearest reference face. Indeed, we have test our registration approach only on human face variation but nothings prevents to use it with any surface with similar characteristics.

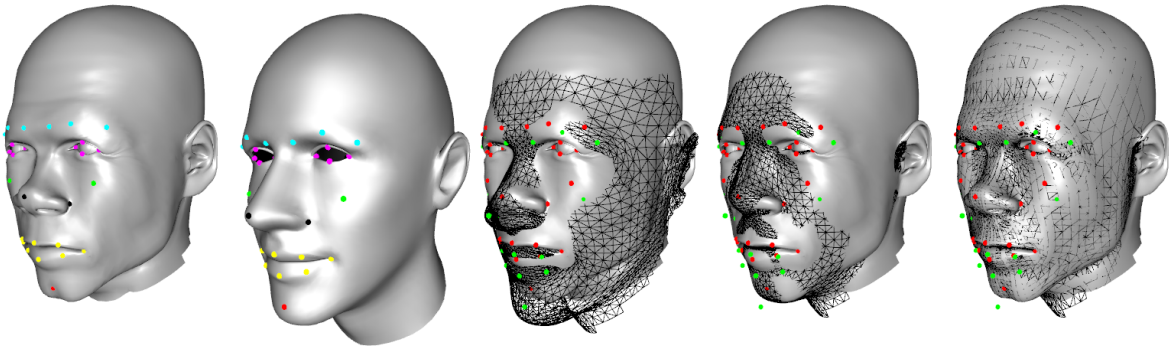


Fig. 8: Illustration of the dense registration based on the landmark correspondence. The two left images show the two meshes to register with their $\mathcal{FP}$. The third image shows the superposition of the two meshes (target mesh in wireframe). The fourth image shows the superposition of the source mesh on the target mesh transformed only by the *RBF* interpolation. The right image shows the dense registration after the spherical projection, each target vertex is associate to a triangle of the source mesh.

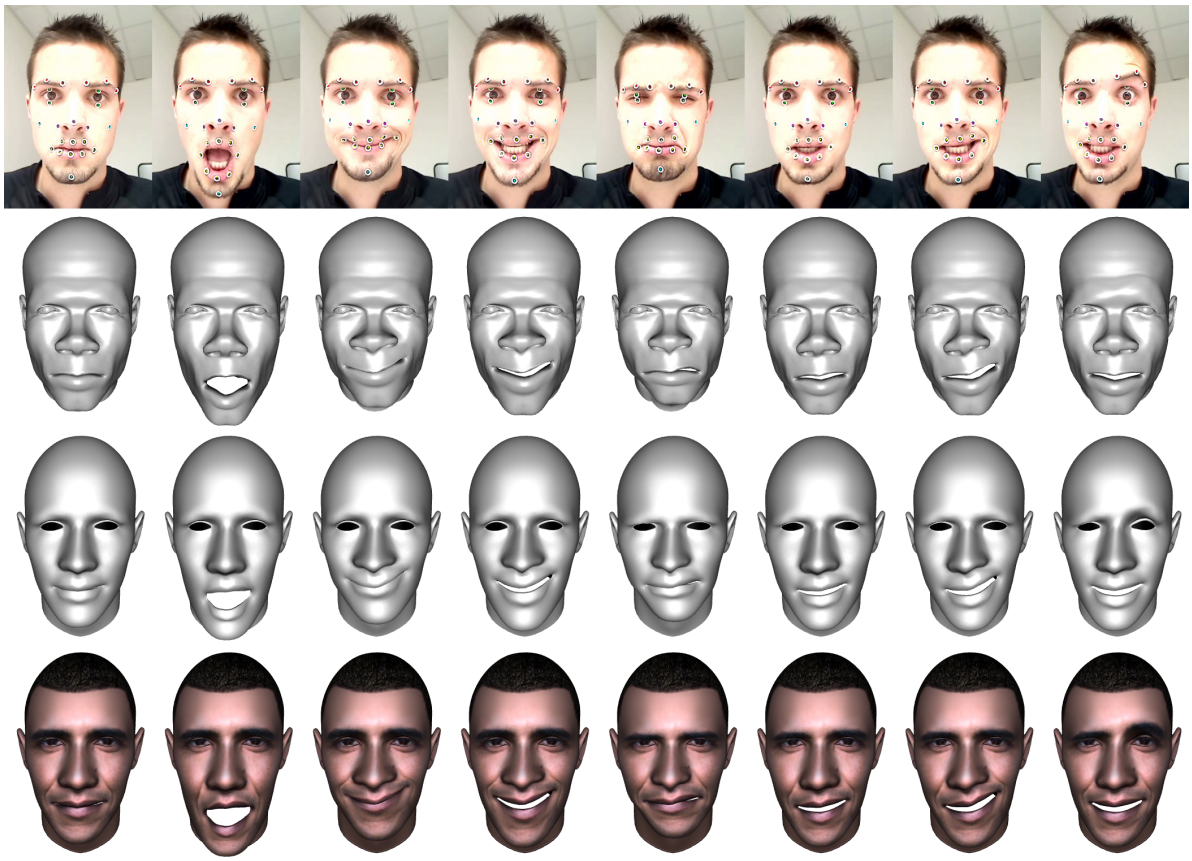


Fig. 9: First line, motion capture animation. Second line, reference model previously rigged. Third and fourth lines, the animation is applied on a face automatically rigged by our method, *i.e.* rigging parameters are transferred from the reference model.

References

1. Geoegios Stylianou and Andreas Lanitis. Image based 3d face reconstruction: a survey. *International Journal of Image and Graphics*, 2009.
2. Nikolaos Ersotelos and Feng Dong. Building highly realistic facial modeling and animation: a survey. *The Visual Computer*, 24(1):13–30, 2007.
3. N. Magnenat-Thalmann, R. Laperrire, , and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface*, 1988.
4. J.P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, 2000.
5. Z. Deng and J. Y. Noh. *Computer Facial Animation: A Survey*, chapter 1, pages 1–28. Springer, Ulrich Neumann, Zhigang Deng (eds.), September 2007.
6. J.L. Zhong. Flexible face animation using mpeg-4/snhc parameter streams. In *IEEE International Conference on Image Processing*, pages II: 924–928, 1998.
7. Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.
8. Xiaohuan Corina Wang and Cary Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 129–138, New York, NY, USA, 2002. ACM.
9. Alex Mohr and Michael Gleicher. Building efficient, accurate character skins from examples. *ACM Trans. Graph.*, 22(3):562–568, 2003.
10. Robert Y. Wang, Kari Pulli, and Jovan Popović. Real-time enveloping with rotational regression. *ACM Trans. Graph.*, 26(3):73, 2007.
11. Softimage. Pick the landmarks with face robot. *http://facerobot.wiki.softimage.com/index.php/Stage_3:_Pick_the_Landmarks*, 2009.
12. Verónica Costa Teixeira Orvalho, Ernesto Zacur, and Antonio Susin. Transferring the rig and animations from a character to different face models. *Computer Graphics Forum*, 27(8):1997–2012, 2008.
13. Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5):578–596, 2007.
14. Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
15. Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.
16. Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, 2004.
17. P. Eisert D. Schneider. Algorithms for automatic and robust registration of 3d head scans. *Journal of Virtual Reality and Broadcasting 2009*, 2009. `http://iphome.hhi.de/eisert/papers/jvrb09.pdf`.
18. Nathan Litke, Marc Droske, Martin Rumpf, and Peter Schröder. An image processing approach to surface matching. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, 2005.
19. Wei Zeng, Yun Zeng, Yang Wang, Xiaotian Yin, Xianfeng Gu, and Dimitris Samaras. 3d non-rigid surface matching and registration based on holomorphic differentials. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 1–14, Berlin, Heidelberg, 2008. Springer-Verlag.
20. Yaron Lipman and Thomas Funkhouser. Mobius voting for surface correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.
21. Fr'ed'eric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 75–84, New York, NY, USA, 1998. ACM.
22. Jun yong Noh and Ulrich Neumann. Expression cloning. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 277–288, 2001.
23. H. Arora Varsha Hedau and N. Ahuja. Matching images under unstable segmentation. In *IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, 2008.
24. Alan P. Mangan and Ross T. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, 1999.
25. David G. Lowe. Distinctive image features from scale-invariant keypoints,. *International Journal of Computer Vision*, *http://people.cs.ubc.ca/~lowe/keypoints/*.
26. Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 2003.
27. Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, 1981.
28. Frederic Pighin and J. P. Lewis. Facial motion retargeting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 2, New York, NY, USA, 2006. ACM.