FITTING 3D MORPHABLE MODELS USING IMPLICIT REPRESENTATIONS

Curzio Basso and Alessandro Verri
DISI, Università di Genova, Italy

Keywords: 3D Morphable Models, non-rigid registration, implicit surface representations.

Abstract: We consider the problem of approximating the 3D scan of a real object through an affine combination of examples. Common approaches depend either on the explicit estimation of point-to-point correspondences or on 2-dimensional projections of the target mesh; both present drawbacks. We follow an approach similar to (Ilic and Fua, 2003) by representing the target via an implicit function, whose values at the vertices of the approximation are used to define a robust cost function. The problem is approached in two steps, by approximating first a coarse implicit representation of the whole target, and then finer, local ones; the local approximations are then merged together with a Poisson-based method. We report the results of applying our method on a subset of 3D scans from the Face Recognition Grand Challenge v.1.0.

1 INTRODUCTION

We consider the problem of approximating a *target* 3D surface with an affine combination of (registered) examples, under the assumption that both the examples and the target belong to the same object class. When the assumption holds, the affine space generated by the examples represents a *model* of the object class, usually known as 3D Morphable Model in case of faces (Blanz and Vetter, 1999). The model is parameterized by the coefficients of the affine combination. We consider the class of human faces; our goal is to find the best possible combination of the available examples that approximates a given target. This is essentially a problem of non-rigid registration, where the available deformations are constrained by the model; it can arise not only when building a 3D Morphable Model (Allen et al., 2003; Basso and Vetter, 2006), but also for performing 3D shape analysis as done for images by (Blanz and Vetter, 2003).

In general, we might approach the problem in different ways, depending on the representation of the target surface:

- the target is explicitly represented as a triangular mesh, and the problem is solved in $\mathbb{R}^3$;
- the target is projected to $\mathbb{R}^2$, as depth map or cylindrical projection;
- the target is represented implicitly in $\mathbb{R}^3$.

The first strategy is essentially based on the Iterated Closest-Point (ICP) registration algorithm (Besl and McKay, 1992; Turk and Levoy, 1994). Although originally used for rigid registration, it can be extended to non-rigid registration and to the case at hand. This was demonstrated in (Allen et al., 2003), where the possibility of using examples-based models was also mentioned. The drawback of this type of approaches is the need of an explicit estimation of the point-to-point correspondence between the model and the target. Letting aside the problems due to holes in the target, the search for corresponding points in $\mathbb{R}^3$ might be computationally inefficient.

The second approach, adopted by (Blanz and Vetter, 1999) and (Basso and Vetter, 2006), avoids the explicit estimation of point-to-point correspondences, by minimizing the difference between the cylindrical projections to $\mathbb{R}^2$ of the target and the model. As well as avoiding the direct search for corresponding points, the projection to $\mathbb{R}^2$ has the advantage of reducing the problem dimensionality. However, this comes at the cost of loosing some information about the 3D sur-

Basso C. and Verri A. (2007).
FITTING 3D MORPHABLE MODELS USING IMPLICIT REPRESENTATIONS.
In *Proceedings of the Second International Conference on Computer Graphics Theory and Applications - GM/R*, pages 45-52
DOI: 10.5220/0002077200450052
Copyright © SciTePress

faces, since the projection is not a parameterization of the original surface. Moreover, the projection is usually non-linear and occlusions frequently occur.

In this paper we investigate the possibility of solving the problem representing the target implicitly in $\mathbb{R}^3$. In doing so, we avoid the explicit correspondence estimation, while still working in the original domain of the data. Although it has never been applied to our specific problem, (Ilic and Fua, 2003) have used an implicit representation to reconstruct surfaces from uncalibrated video sequences. In their work, however, it is not the target data that are represented implicitly, but rather the model, a deformable template that has to be matched to the data. Also related to ours is the work of (Steinke et al., 2005), where the problem of registering the target with a fixed template (not an example-based model) was solved representing both surfaces implicitly. The implicit representations where found using Support Vector Machines (SVM).

Our work is based on a simpler implicit representation of the target mesh, based on Radial Basis Functions (RBF), while the model is still represented explicitly. The implicit representation defines a sort of potential field in the space surrounding the target surface, and can be used to define a robust cost function. In order to reduce the complexity of computing the RBF for large meshes, we first minimize the cost function over a low resolution implicit representation, and afterward we minimize it over higher resolution patches of the target. Finally, the different patches are blended together with a Poisson-editing approach (Yu et al., 2004). In our work we assume that the target is already coarsely aligned with the model.

2 BACKGROUND

In the following two subsection we briefly expose the basic notions our method builds upon: three-dimensional Morphable Models, and implicit representations of surfaces.

2.1 3D Morphable Models

A triangular mesh is defined by a graph $M = (V,E)$; V is a set of n vertices in $\mathbb{R}^3$ and E is a set of edges connecting the vertices. For convenience we write V as a matrix

$$V = [v_1 \dots v_n] = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ z_1 & \dots & z_n \end{bmatrix} \in \mathbb{R}^{3\times n}. \quad (1)$$

A set of m registered meshes, denoted by M_i, will share the same connectivity E but have different vertices positions V_i. We can naturally define the subspace of their affine combinations, with coefficients $a = (a_1, \dots, a_m) \in \mathbb{R}^m$, as

$$M(a) = (V(a), E), \quad (2)$$

with

$$V(a) = \sum_{i=1}^{m} a_i V_i \quad \text{and} \quad \sum a_i = 1. \quad (3)$$

The affine constraint is needed in order to avoid scaling effects.

Note that rewriting eq. (3) in terms of the barycenter

$$\overline{V} = \frac{1}{m}\sum_{i=1}^{m} V_i, \quad (4)$$

we can eliminate the constraint on the sum of the coefficients:

$$V(a) = \overline{V} + \sum_{i=1}^{m} a_i (V_i - \overline{V}). \quad (5)$$

Three-dimensional Morphable Models (3DMM) use a representation for the affine combination based on the Principal Component Analysis (PCA) of the subspace spanned by the $V_i - \overline{V}$. Without going into the details of the PCA (Hastie et al., 2001, pp. 62-63), we only report that for 3DMM eq. (5) is written as

$$V(\alpha) = \overline{V} + \sum_{i=1}^{m-1} \alpha_i U_i, \quad (6)$$

where the U_i are the principal components and the coefficient vector a is replaced by $\alpha \in \mathbb{R}^{m-1}$. See figure 1 for a simple two-dimensional example.

Assuming the examples are correctly registered, any rigid transformation has been factored out from the model. It can be explicitly included by defining a rotation matrix R and a translation vector t which are applied to $V(\alpha)$:

$$V(\alpha,\rho) = R(\rho) \cdot \left\{ \overline{V} + \sum_{i=1}^{m-1} \alpha_i U_i \right\} + t(\rho) \otimes 1_n^T, \quad (7)$$

where the last term is simply an n-times repetition of the column vector t. R and t are parameterized by a vector $\rho \in \mathbb{R}^6$, holding the coefficients of the transformation (three for the rotation and three for the translation). Taking into account the rigid transformation, the model is defined by $M(\alpha,\rho) = (V(\alpha,\rho), E)$.

2.2 Implicit Representations of 3D Surfaces

An implicit representation of a given surface $T \subset \mathbb{R}^3$ is a function $F_T : \mathbb{R}^3 \to \mathbb{R}$ such that the surface is one

of its level sets. That is, $F_T(v) = h$ for each $v \in T$, and $F_T(v) \neq h$ otherwise, where h is a constant, typically zero. Clearly, an example of implicit representation is the Euclidean distance

$$F_T(v) = \min_{w \in T} \|v - w\|. \tag{8}$$

In order to define a cost function based on such an implicit representation, we are interested in an analytic form for the implicit function F_T; we build it following the lines of (Turk and O'Brien, 1999; Carr et al., 2001). Given a certain 3D mesh, they look for a function $F(v)$ which is zero on the vertices and different from zero on a set of off-surface points. The off-surface points are required to avoid the trivial solution of a function identically zero over the whole space. They are chosen to lie on the normal to the surface at the mesh vertices. Let us denote by w_j the vertices and the off-surface points. Then, given a radial basis function $\phi(x)$, there exist a choice of scalar weights d_j and of a degree one polynomial $P(v)$ such that the function

$$F(v) = \sum_{j=1}^{n} d_j \phi(v - w_j) + P(v), \tag{9}$$

satisfies the constraints $F(w_j) = h_j$ and is also smooth, in the sense that minimizes the energy

$$E = \int_{\Omega \subset \mathbb{R}^3} F_{xx}^2 + F_{yy}^2 + F_{zz}^2 + 2F_{xy}^2 + 2F_{yz}^2 + 2F_{zx}^2, \tag{10}$$

a generalization to $\mathbb{R}^3$ of the thin-plate energy. In figure 1 we show such an example of $F_T(v)$, where T is a poly-line in 2D.

In practice, the unknown vectors d and p (coefficients of the polynomial) are the solutions of a linear system. Given the matrices

$$\Phi = \begin{bmatrix} \phi_{11} & \dots & \phi_{1n} \\ \vdots & & \vdots \\ \phi_{n1} & \dots & \phi_{nn} \end{bmatrix} \tag{11}$$

and

$$A = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix} \tag{12}$$

one has to solve the system

$$\begin{bmatrix} \Phi & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d \\ p \end{bmatrix} = \begin{bmatrix} h \\ 0 \end{bmatrix}. \tag{13}$$

Note that when the input mesh is large, the above linear system, if dense, might become intractable. A possible solution is to induce sparsity by choosing a basis with compact support, but this on turn creates extrapolation problems. In order to maintain sparsity and achieve good extrapolation behavior an option is to define multiple levels of resolution, as done by (Ohtake et al., 2003). Without resorting to a compact support basis, one can use Fast Multipole Methods to reduce both the storage and the computational cost, as done in (Carr et al., 2001).

3 LOW RESOLUTION APPROXIMATION

Given a 3DMM $M(\alpha, \rho)$ as defined in section 2.1, and a target mesh T, we formulate the approximation problem as the optimization

$$\alpha^\star, \rho^\star = \arg\min_{\alpha,\rho} D(M(\alpha, \rho), T), \tag{14}$$

where D is a suitable function measuring the approximation cost.

Assuming we have an implicit representation F_T for the target mesh, we can define the cost of approximating T by $M(\alpha, \rho)$ as

$$D(M(\alpha, \rho), T) = \frac{1}{N} \sum_{i=1}^{N} \ell(F_T(v_i)), \tag{15}$$

where the v_i are the vertices of $M(\alpha, \rho)$, and ℓ might be a quadratic loss or an M-estimator. The choice of an M-estimator might be necessary when parts of the model M are not present in the target T and influence the correct approximation of the rest. Observe that in the above definition we are essentially treating the values $F_T(v_i)$ as residuals of the approximation. In fact, if F_T was the Euclidean distance function and $\ell(x) = x^2$, then D would correspond to an ℓ_2 norm.

3.1 Optimization Scheme

In order to find the minimum of the cost function (15), we use a modified Newton's method. For the sake of clarity, in the following discussion we denote by θ the vector of all model coefficients, without distinctions between α and ρ. Recall that the *exact* Newton's method consists in iteratively updating θ by adding the solution p of the linear system

$$\nabla^2 D(\theta) \cdot p = -\nabla D(\theta), \tag{16}$$

where $\nabla^2 D(\theta)$ denotes the Hessian matrix of D at θ. The exact method converges to a minimum of D when sufficiently close to it, but it does not in general; nevertheless, there are a number of standard modifications that make it more efficient and robust. We employ a simple scheme, which keeps the Hessian matrix sufficiently positive definite by adding a multiple

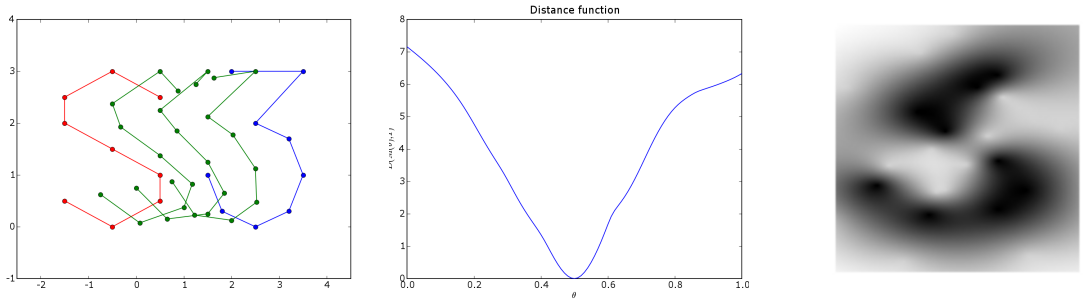

Figure 1: On the left, an example of a 2D morphable model $M(\alpha)$: two poly-lines, in red and blue, representing the characters 'S' and '3'. The green curves are linear combinations of the two examples, and can be written as the average shape plus a deformation along the only principal direction. The amount of the deformation is given by the coefficient α, whose values for the three green curves are respectively $\alpha = \{-0.5, 0, 0.5\}$. If we consider the average shape ($\alpha = 0$), it generates the implicit representation on the right, where the gray-levels correspond to the values of $F_T(x)$, as defined by eq. (9) with $\phi(x) = \|x\|$. The figure in the middle shows the corresponding cost function $D(M(\alpha), T)$ as defined by eq. (15) with $\ell(x) = x^2$.

of the identity when required (Nocedal and Wright, 1999, Ch.6.3). Another modification to the exact method is that the update length is not unitary, but it is determined by a backtracking procedure which reduces the length if the update does not reduce D (Nocedal and Wright, 1999, Ch.3.1).

Having the exact form of the implicit representation allows us to compute analytically the gradient and the Hessian matrices of the cost function:

$$\frac{\partial D}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N} \ell_i' \left(\nabla F_i \cdot \frac{\partial v_i}{\partial \theta_j} \right), \tag{17}$$

and

$$\frac{\partial^2 D}{\partial \theta_j \partial \theta_k} = \frac{1}{N}\sum_{i=1}^{N} \ell_i'' \left(\nabla F_i \cdot \frac{\partial v_i}{\partial \theta_j} \right) \left(\nabla F_i \cdot \frac{\partial v_i}{\partial \theta_k} \right)$$
$$+ \frac{1}{N}\sum_{i=1}^{N} \ell_i' \left(\frac{\partial v_i}{\partial \theta_j} \cdot \nabla^2 F_i \cdot \frac{\partial v_i}{\partial \theta_k} + \nabla F_i \cdot \frac{\partial^2 v_i}{\partial \theta_j \partial \theta_k} \right). \tag{18}$$

In the above equations we used ℓ_i' and ℓ_i'' to denote the first and second derivatives of the loss function computed at $F_T(v_i)$; ∇F_i and $\nabla^2 F_i$ denotes the gradient vector and the Hessian matrix of F_T with respect to the spatial coordinates, computed at v_i.

In practice we choose ℓ to be the Tukey estimator, that is

$$\ell(x) = \begin{cases} \frac{c^2}{6}\left[1 - \left(1 - (x/c)^2\right)^3\right] & \text{if} |x| \le c \\ c^2/6 & \text{otherwise} \end{cases} \tag{19}$$

with derivatives

$$\ell'(x) = \begin{cases} x\left[1 - (x/c)^2\right]^2 & \text{if} |x| \le c \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

$$\ell''(x) = \begin{cases} \left[1 - 5(x/c)^2\right]\left[1 - (x/c)^2\right] & \text{if} |x| \le c \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Note that the effectiveness of the Tukey estimator in reducing the influence of model vertices not present in the target depends on the scale of the constant c: the smaller the constant, the smaller the influence of missing vertices. On the other hand, small c's require a good pre-alignment of the model with the target. In all our experiments we used $c = 5.0$.

The gradient and Hessian of F_T depends on the choice of the basis ϕ. Following (Carr et al., 2001), we use the biharmonic basis function $\phi(x) = \|x\|$, which results in the smoothest solution to the interpolation problem among all radial basis functions. Such a function with non-compact support is also more suitable to inter- and extrapolation. The derivatives are easily computed

$$\nabla\phi(x) = x/\phi, \tag{22}$$
$$\nabla^2\phi(x) = \left(I_3 - \nabla\phi \cdot \nabla\phi^T\right)/\phi. \tag{23}$$

Accordingly, we have

$$\nabla F_i = \sum_{j=1}^{n} \frac{d_j}{\|v_i - w_j\|}(v_i - w_j) + (p_1, p_2, p_3), \tag{24}$$

$$\nabla^2 F_i = \sum_{j=1}^{n} \frac{d_j}{\|v_i - w_j\|}\left(I_3 - \frac{(v_i - w_j)(v_i - w_j)^T}{\|v_i - w_j\|^2}\right). \tag{25}$$

3.2 Regularizing Prior

In many cases, in particular to avoid overfitting, it might be convenient to add to the cost function (15) a regularization term which penalizes excessively large model coefficients:

$$D_R = D + \eta_\alpha E(\alpha) + \eta_\rho E(\rho), \tag{26}$$

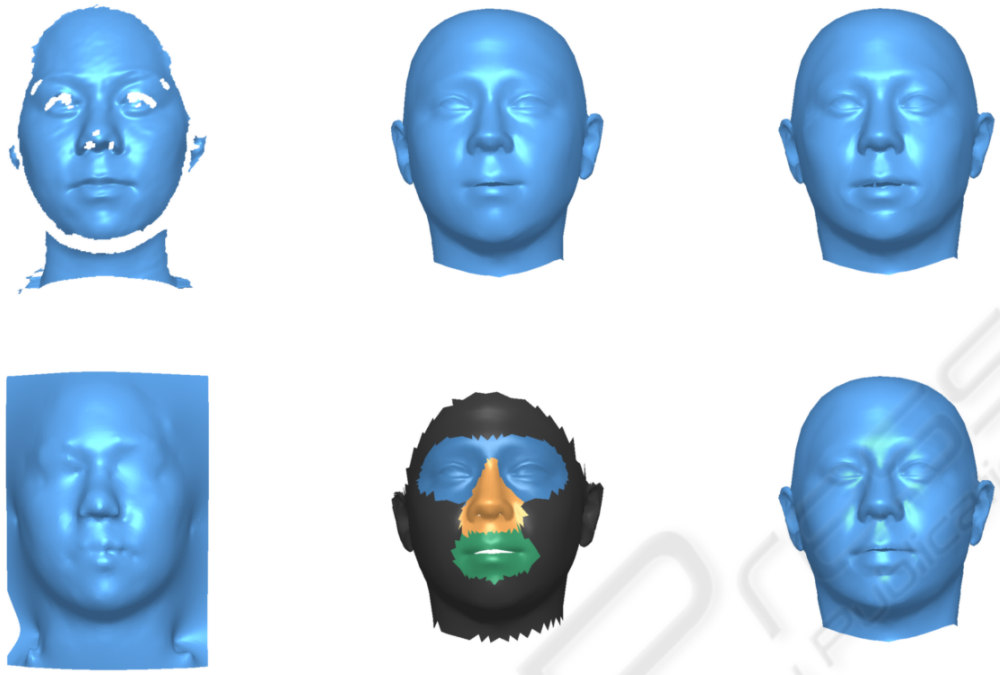

Figure 2: Examples of input data and the results of each approximation step. On the left column are shown the target (top image) and the implicit representation of its subsampling (bottom). Fitting the model to the latter results in the shape in the middle column, top image. The next step consist in separately fitting the segments of the model, shown in the bottom-center image, to corresponding parts of the target. The result is shown in the right column: on the top are the different approximations stitched together, on the bottom the result of blending them.

where the parameters η_α and η_ρ weight the effect of the regularization on the shape and the rigid parameters, respectively.

A standard way to choose the regularization terms consists in deriving them from prior probabilities. In the case of the shape coefficients, the PCA model assumes for α a normal distribution with unit variance; for the rigid coefficients we also assume a Gaussian distribution with zero mean, but with empirically chosen variances. In both cases, we can define a regularization term proportional to the inverse log-likelihood. For a generic coefficient θ with variance σ^2 we have

$$-\log p(\theta) = \frac{\theta^2}{2\sigma^2} + const. \tag{27}$$

so that we set (Σ_ρ is the diagonal covariance matrix for the rigid coefficients)

$$E(\alpha) = \frac{1}{2}\|\alpha\|^2 \quad \text{and} \quad E(\rho) = \frac{1}{2}\rho^T\Sigma_\rho^{-1}\rho. \tag{28}$$

4 SEGMENTED APPROXIMATION

As noted in section 2.2, a direct solution of the interpolation problem by solving the linear system is computationally and storage intensive. In fact, for high resolution target meshes, the system is too large to be allocated in memory. Rather than following one of the methods mentioned in section 2.2, we decided to overcome the problem by adopting a simpler multi-resolution approach (refer to figure 2 for examples of the intermediate steps outputs):

1. **Coarse Approximation.** We first compute a coarse approximation of the target. We select a subset of the target mesh vertices, sampled with probabilities proportional to the areas of the adjoining triangles, and use them as constraints of the interpolation problem. The resulting implicit surface is fit with the procedure described in the previous section.
2. **Partitioning.** Once we have obtained the coarse approximation of the target, we can achieve a higher resolution by repeating the process on

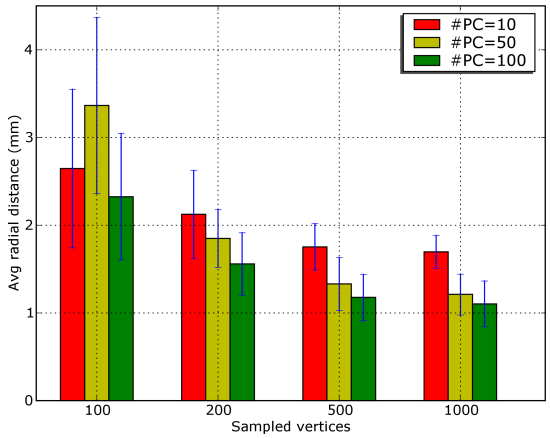

Figure 3: Dependency of the approximation on the number of principal components used by the model and the number of vertices sampled from the target surface.

smaller patches of the target and finally merging the local approximations. To this aim, we manually defined four regions on the model topology; the regions are visible in figure 2, middle image of the bottom row. Given the coarse approximation, we compute the bounding box of each region, we expand it in all directions by a fixed length (2 cm in our experiments), and select all the target's vertices falling inside the box. This results in four overlapping subsets of the target vertices, each one associated with a different segment of the model.

3. **Finer Approximations.** For each subset of the target vertices we build an (approximate) implicit representation by sampling its vertices as done in step 1. Although the subsets are again subsampled, the ratio between the number of sampled vertices and the total number of vertices clearly increases, resulting in a more precise representation. The implicit surfaces are fit again with the usual procedure.

4.1 Blending

It is clear that the approximation method explained in the previous section provides local results, which do not match precisely at their boundaries. In order to merge them smoothly, we use a variational approach akin to the Poisson-based mesh editing of (Yu et al., 2004), to which we refer for more details. The main idea is to keep fixed the positions of the vertices in the interiors of the segments and let the other vertices relax to the positions which minimize an elastic energy. The procedure is as follows:

1. identify the boundaries of the patches;
2. for each vertex, compute its minimum distance from the boundaries, by fast marching (Kimmel and Sethian, 1998);
3. define the interior as the set of vertices with distance greater than a certain threshold (we used 0.5 cm);

Steps 1 to 3 define the mesh interior Ω and its complement, the boundary region $\overline{\Omega}$: an area with given width that surrounds the segments boundaries. We denote by $\partial\Omega$ the set of vertices in the interior Ω that are connected to any vertex in the boundary region $\overline{\Omega}$. While the vertices in Ω are fixed, the positions of the vertices in $\overline{\Omega}$ are obtained by solving the Poisson equation:

4. for all the vertices in $\overline{\Omega}$, find their position solving a discretized Poisson equation using as boundary conditions the positions of the vertices in $\partial\Omega$ and as guidance field the gradient of the coarse approximation.

In practice, one solves the Poisson equation for the field of displacements from the coarse approximation to the fine one. This formulation yields a sparse linear system of equations of the type $\Delta d_i = 0$, where Δ is the discrete Laplacian operator over the mesh (Taubin, 1995), and d_i is the displacement of the i-th vertex in $\overline{\Omega}$. The system is solved under the boundary constraints at $\partial\Omega$, where the displacements are known.

5 RESULTS

The above method has been tested using as model a sub-sampled version of the mixed expression-identity 3D Morphable Model built with the algorithm described in (Basso and Vetter, 2006). The subsampling is simple, since the reference template was built by subdivision of a low-resolution one. Out of around 40k vertices at full resolution, we retained approximately 2.5k. As well as reducing the number of vertices, we also discarded the expression shape components and all the texture components. The targets were a set of 165 range scans, randomly sampled from the range data distributed for the Face Recognition Grand Challenge v.1.0 (Phillips et al., 2005). The scans were distributed with a list of landmark positions, which we used to pre-align the faces to the model. The approximations have been performed using 99 model components and 500 sampled vertices for building the implicit representations. We should remark that these are preliminary experiments, and a more in-depth study will follow.

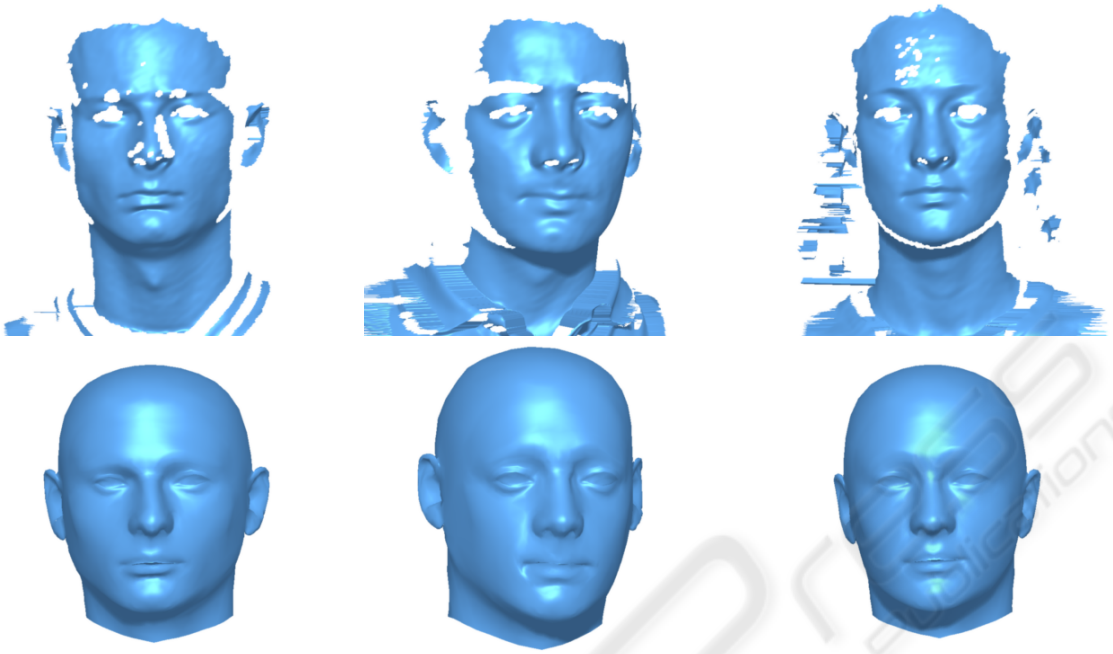

Figure 4: Some examples of approximations (bottom row, the original are on the top row). The average per-vertex radial errors for the three examples are, respectively: 0.80, 0.86 and 0.80 mm.

We evaluated the goodness of the fit by cylindrically projecting the target and the approximation, and computing the average absolute error (a radial distance) on the four segments. The rest of the head model was not taken into account because of the hair and clothes often present in the scans. Of the 165 results, we rejected three of them in which the error was more than three standard deviations larger than the average. In the remaining 163 results, the average radial error was 1.09 mm on the blended high-resolution result, with an average improvement with respect to the low resolution approximation of 0.39 mm. Some of the results are shown in figure 4.

We also ran a small experiment to assess the dependency of the approximation on the number of principal components used and the number of vertices sampled from the target surface. We repeatedly computed the low resolution approximations for a small subset of test data (only four examples), varying the two aforementioned parameters. The results are in figure 3. The dependency on the number of principal components used in the model behaves as expected: up to a point, the more components are used the better the approximation performance. This is not true anymore when input is too noisy, as it probably occurs when the number of sampled vertices is only 100. More interestingly, the dependency on the number of sampled vertices shows that an increase on the sample rate pays off only up to a point, while the improvement from 500 to 1000 is minimal.

6 CONCLUSION

We have presented a method which makes simple use of the implicit representation of a surface to find its optimal approximation in terms of an affine combination of examples. Implicit representations are appealing in general, because they are topology-free and typically quite robust to holes in the data. In our setting, they also offer the advantage of completely avoiding the problem of estimating correspondences.

As we saw, however, the use of implicit representations poses serious computational problems when dealing with high-resolution meshes. Therefore, we have proposed to tackle the problem in a multi-resolution fashion, and we have shown how this approach can provide good results without computing full-resolution representations. We should remark that, in assessing the method, we considered its performance only in absolute terms, without comparing it with respect to other, already published algorithms.

This is certainly a deficiency of our work, and we will have to correct it in the future. Nevertheless, we believe that the absolute performance achieved by our method on the Face Recognition Grand Challenge range data is a strong indicator of its applicability to a real-world scenario.

The sub-sampling of the original surface is further motivated by the consideration that the model has also a fixed level of resolution. The resolution of the model does not only dependent on the number of vertices of its mesh, but especially on the number of training examples and of components used. It seems therefore reasonable to tweak the resolution of the target so that it matches the one of the model. On the other hand, sub-sampling poses problems. This is particularly clear when considering the approximation results obtained on targets containing clothes or hairs. In the best case these data are irrelevant, in the worst they are harmful, since they will cause distortions in the low-resolution implicit representation, which might be enhanced by an unlucky sub-sampling. Preprocessing of the targets which removes these data would certainly improve the method's performance and stability.

The future development will focus on two problems: first, the choice of the optimal segments in which the model is partitioned, and second, the integration of a texture model in the approximation scheme.

ACKNOWLEDGEMENTS

This work has been partially supported by the FIRB project RBIN04PARL.

REFERENCES

Allen, B., Curless, B., and Popovic, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. ACM SIGGRAPH 03*.

Basso, C. and Vetter, T. (2006). Registration of expressions data using a 3d morphable model. *Journal of Multimedia*, 1(4):37–45.

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

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proc. ACM SIGGRAPH 99*, pages 187–194. ACM Press.

Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proc. of ACM SIGGRAPH 2001*.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistic. Springer.

Ilic, S. and Fua, P. (2003). Implicit meshes for modeling and reconstruction. In *Proc. Conf. Computer Vision and Pattern Recognition 2003*.

Kimmel, R. and Sethian, J. (1998). Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA*, 95(15):8431–8435.

Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer Series in Operations Research. Springer.

Ohtake, Y., Belyaev, A., and Seidel, H.-P. (2003). A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 153, Washington, DC, USA. IEEE Computer Society.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W. (2005). Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 947–954, Los Alamitos, CA, USA. IEEE Computer Society.

Steinke, F., Scholkopf, B., and Blanz, V. (2005). Support vector machines for 3d shape processing. *Computer Graphics Forum*, 24(3):285–294.

Taubin, G. (1995). A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, New York, NY, USA. ACM Press.

Turk, G. and Levoy, M. (1994). Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, New York, NY, USA. ACM Press.

Turk, G. and O'Brien, J. F. (1999). Shape transformation using variational implicit functions. In *Proc. of ACM SIGGRAPH 99*.

Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., and Shum, H.-Y. (2004). Mesh editing with poisson-based gradient field manipulation. In Slothower, D., editor, *Proc. ACM SIGGRAPH 04*, pages 641–648. ACM Press.