Efficient Groupwise Non-rigid Registration of Textured Surfaces

Kirill A. Sidorov
School of Computer Science
Cardiff University, UK
K.Sidorov@cs.cardiff.ac.uk

Stephen Richmond
School of Dentistry
Cardiff University, UK
richmonds@cardiff.ac.uk

David Marshall
School of Computer Science
Cardiff University, UK
dave@cs.cardiff.ac.uk

Abstract

Advances in 3D imaging have recently made 3D surface scanners, capable of capturing textured surfaces at video rate, affordable and common in computer vision. This is a relatively new source of data, the potential of which has not yet been fully exploited as the problem of non-rigid registration of surfaces is difficult. While registration based on shape alone has been an active research area for some time, the problem of registering surfaces based on texture information has not been addressed in a principled way. We propose a novel, efficient and reliable, fully automatic method for performing groupwise non-rigid registration of textured surfaces, such as those obtained with 3D scanners. We demonstrate the robustness of our approach on 3D scans of human faces, including the notoriously difficult case of inter-subject registration. We show how our method can be used to build high-quality 3D models of appearance fully automatically.

1. Introduction

The non-rigid registration of 2D images in a groupwise fashion has recently drawn significant interest and has been shown to be superior to traditional pairwise methods [4]. The groupwise paradigm has also been applied to the non-rigid registration of surfaces [7]. Recently, video-rate 3D surface scanners, capturing texture as well as shape, have become common. The information content of the textures is very often much higher than that of the shapes alone, especially in the case of facial imagery and other real world objects. In such cases it seems sensible to use texture information for registration. However, the problem of non-rigid registration of deformable surfaces with texture, especially in a groupwise paradigm, has not been addressed in a principled way.

Bending invariants have been used in [17] to compute groupwise correspondences between meshes. However, in [17] the general case of non-rigid registration is not addressed: their model of deformations is only affine and they use use Generalised Procrustes Analysis and nearest-neighbour correspondence to compute the best affine alignment of the bending invariants. A method for non-rigid surface registration, expressed as a graph matching problem, relying on photometric information in addition to shape, is proposed in [19]. However, [19] is a pairwise approach and cannot be readily adopted to the groupwise registration paradigm. In [18], a surface registration method is proposed which is based on detecting features on meshes equipped with scalar fields and using those features for matching (unlike the method in this paper, which is an area-based method and does not require salient features to be present). The method of [18] can only be applied to pairs of meshes and cannot readily be converted to the groupwise paradigm. A texture-based non-rigid surface registration method of [16] is, again, non-groupwise and requires a prior built Active Appearance Model to track features (and so cannot be applied to arbitrary data).

In this paper, a novel, reliable automatic method for groupwise non-rigid registration of textured surfaces is proposed. More specifically, we focus on the registration of textured genus-0 disk-like orientable open surfaces represented by triangulated meshes, which are very common in computer graphics and vision. However, our basic approach could be readily adapted to other types of surfaces (*e.g.* closed surfaces).

The solution to the above problem is important as it allows fully automatic construction of 3D appearance models, with applications in computer graphics and vision, orthodontics, biometrics and security. Large corpora of textured 3D scans (including 3D videos) can now be easily obtained. This is why, a fully automated method for 3D textured surface registration is essential for subsequent analysis and modelling of such data. Constructing 3D appearance models, *e.g.* Morphable Models [1], requires correspondences between analogous features on the surfaces to be known: while in the case of 2D images such correspondences can be defined by manual annotation, this is completely impractical in the 3D case and so an automatic method has to be devised. Since our proposed algorithm achieves this, efficient 3D appearance model is now a reality. We demonstrate this in Section 3, including the difficult case of building an inter-subject appearance model.

We reduce the problem of 3D surface registration to that analogous to image registration. A naïve approach would be to parameterise the surfaces (mapping to a plane) and then register the textures as flat images. However, this naïve approach does not usually work well because it is agnostic of the 3D geometry of the meshes. In other words, small displacements in the flattened textures, induced by the deformation model, do not necessarily correspond to small displacement on the original 3D surfaces and vice versa. The more different the surfaces are from a flat disk, the more pronounced is this effect.

We adopt a more principled approach in which the deformation model is defined on the original meshes and so is explicitly aware of the 3D geometry. The main idea is to maintain the correspondences between surfaces and to operate with textures in a common flat reference space while performing optimisation on the original 3D surfaces. This can be regarded as gradually computing the embeddings of the surfaces into a plane such that they also bring all surfaces into alignment simultaneously. To accomplish this, we begin by computing maximally isometric embeddings which implicitly define correspondences between surfaces and iteratively improve them. Before we proceed, we first review the techniques required by our approach.

Essential to many operations in our method is the ability to compute the geodesic distance between any two points on a mesh. The well established Fast Marching methods [9] on triangulated domains allow for the geodesic distances between points on a mesh ("single source, multiple targets") to be computed very efficiently.

Bending invariants [8], in which Euclidean distances between points correspond to the geodesic distances on the original surfaces, can be used to remove the bending component of the deformation, leaving only the stretching. For example, it was shown empirically [2] that geodesic distances on the surface of a face are significantly less sensitive to changes in expression than Euclidean distances are. This means, therefore, that bending invariants can be used as an excellent starting point with which to initialise our registration algorithm.

Given a matrix of pairwise dissimilarities between items, a technique called classical Multidimensional Scaling (MDS) maps these items to points in $\mathbb{R}^n$ such that the pairwise Euclidean distance between the resulting points approximate the given dissimilarities as closely as possible in the least squares sense. MDS can be used [8] to compute the bending invariants by inputting the matrix of pairwise geodesic distance between points on a mesh, embedding them into $\mathbb{R}^3$. The distance-preserving property of MDS has also been exploited in a number of works on mesh parameterisation and flattening *e.g.* for texture mapping [20].

Given the geodesic distances between every pair of vertices on a mesh it is possible to select k vertices such that the pairwise distances between them are maximised. This can be accomplished with a Farthest Point Sampling (FPS) algorithm [12]. Selection can performed adaptively, *e.g.* seeding more points in areas of higher curvature.

Groupwise non-rigid registration of (textured) surfaces can be regarded as a large optimisation problem. A concern with groupwise methods is that the dimensionality of the space in which the search for the optimal solution is performed, grows very rapidly with the number of samples in the set. This can present a significant obstacle in finding the optimal solution [6]. It has been shown previously [14] that in groupwise registration of images efficient optimisation can be achieved by incrementally composing complex deformation fields out of elementary (*e.g.* piece-wise affine) warps. In this paper, we propose a method to perform a related procedure for groupwise registration (embedding) of 3D surfaces.

The main contribution of this work is a novel method, building on the above ideas, which provides a fully automatic groupwise non-rigid registration of surfaces based predominantly on texture information. Using a novel combination of ideas from geodesic mesh processing and traditional registration methods, we show how to reliably, in a principled manner, solve the problem of registering 3D surfaces in a fashion analogous to the previously solved [14] problem of 2D image registration. The resulting algorithm is computationally efficient, reliable and fully automatic. Additionally, it is readily amenable to a GPU implementation.

2. Our Proposed Method

Our algorithm takes as input as set of N textured meshes, $\mathfrak{M}_i = \{\mathrm{F}_i, \mathrm{V}_i, \mathrm{U}_i, \mathcal{T}_i\}$, where V_i is a $3 \times n_{v_i}$ matrix whose columns are vertex coordinates, similarly U_i is a $2 \times n_{v_i}$ matrix of texture coordinates, F_i is a $3 \times n_{f_i}$ matrix of indices into V_i defining the faces of the mesh, and $\mathcal{T}_i$ is the texture. Note that, in general, meshes may have a different number of vertices, n_{v_i}, and different topologies, as is usually the case with the 3D data provided by scanners.

For every mesh, we maintain a $2 \times n_{v_i}$ matrix E_i of the embedded vertex coordinates in the common reference plane. We aim to find such embeddings E_i that bring the analogous features in all meshes into alignment. Having done this, we can easily recover the correspondences between points on the original 3D meshes, assuming the embedding is bijective.

As in [7, 14], we regard the problem of groupwise registration as an optimisation problem consisting of three components: a mechanism for representing and manipulating deformations (changes in embedding), an objective func-

Figure 1. *Left:* magnitude of influence of several randomly placed individual "deformation disks" (using Eq. (1) with $r = 110$ mm). *Right:* colour-coded mixture of their influence.

tion F measuring the alignment error, and a global minimisation algorithm which optimises F. We address these components below.

2.1. Model of Deformations

Even though in the proposed algorithm we optimise the embeddings of meshes (mappings to the common reference frame), for parallelism with registration literature we shall call the changes to embedding *deformations*.

By analogy with [10] the total deformation is expressed as a sum of elementary warps defined by bounded "deformation disks", in our approach residing on the surface of a mesh (Fig. 1). Let $\mathfrak{G}(\mathfrak{M}, \mathbf{a}, \mathbf{b})$ denote the geodesic distance between points $\mathbf{a}$ and $\mathbf{b}$ on a mesh $\mathfrak{M}$. Using the elementary warp formulation from [3], we define the influence of a disk, with radius r and centre at $\mathbf{c} \in \mathfrak{M}$, on an arbitrary vertex $\mathbf{v} \in \mathfrak{M}$ as a function of the *geodesic* distance along the surface of the mesh $\mathfrak{M}$ from the disk's centre:

$$B(\mathfrak{M}, \mathbf{v}, \mathbf{c}, r) = \begin{cases} 1 - (d/r)^2 \left(1 + \log((d/r)^2)\right), & d < r \\ 0, & d \geq r \end{cases} \tag{1}$$

where $d = \mathfrak{G}(\mathfrak{M}, \mathbf{v}, \mathbf{c})$ and $\log(0) = 0$. Note the desirable property of this representation: since the magnitude of the influence of a disk depends only on the geodesic distance from its centre, there is no need to define a coordinate system on the mesh.

If $\mathbf{p} \in \mathbb{R}^2$ is a vector of parameters controlling the contribution of a disk (displacement at its centre), the coordinates of the embedded vertices E_i are affected by $\Delta\mathrm{E}_i = \mathbf{p}B(\mathfrak{M}, \mathbf{v}, \mathbf{c}, r)$. Given n_b disks, the complete configuration space (the space of all possible deformations) is described by a matrix of parameters $\mathrm{P}_{2\times n_b}$, where the columns are parameters (contributions) of the individual disks. Once the disks $\{\mathbf{c}_j, r_j\}$ are selected, the magnitudes of their influences on vertices of the mesh can be precomputed and stored in the *influence matrix* $\mathrm{Q}_{n_b \times n_v}$, where the j-th row stores the influence of the j-th disk on the vertices:

$$\mathrm{Q}(j, k) = B\left(\mathfrak{M}, \mathrm{V}(:, k), \mathbf{c}_j, r_j\right). \tag{2}$$

Given Q, the effect of all disks together on all vertices of the mesh is simply $\mathrm{D} = \mathrm{PQ}$. The influence of individual deformation disks as well as of their superposition is illustrated in Fig. 1.

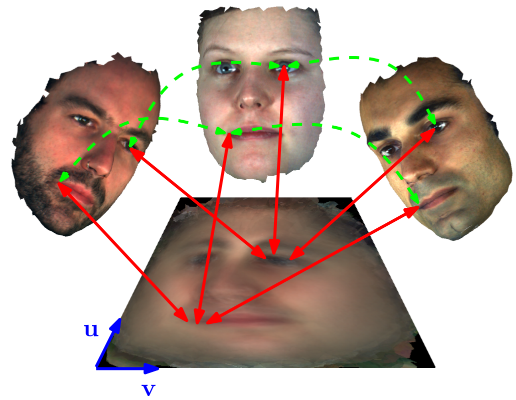

Figure 2. Correspondences via a flat parametric space.

2.2. Objective Function

We next the address the objective function. The purpose of the objective function is to measure how well the correspondences between the analogous features on different surfaces have been established. Instead of operating on the correspondences between surfaces directly, we operate with correspondences between the surfaces and a common reference space (a Euclidean plane, assuming the surfaces can be bijectively embedded in it). Mapping the textures to this reference space enables us to adopt any suitable intensity-based objective function from the 2D image registration literature: the textures in the flat parametric space can be manipulated as ordinary images. This also facilitates GPU-based implementation.

The standard practice in groupwise image registration literature is to maintain an evolving model of pixel colours (*e.g.* average of shape-normalised images) in some reference space to which all samples are aligned. In our case, such a model of texture can also be easily computed by mapping the textures from curved surfaces onto the common reference plane using the estimated correspondences between surfaces and the reference plane and averaging them.

Figure 2 illustrates the idea: vertices of the mesh are mapped to the reference plane, in which the evolving model of texture is maintained. The correspondences between any points on any two meshes can be consistently deduced given the correspondences between the points and the reference plane. In practice, we can represent the mapping between surfaces and the reference plane by specifying for each vertex of a mesh the corresponding coordinates in the reference plane, storing these as columns in a $2 \times n_{v_i}$ matrix E_i, and interpolating between the vertices.

In practice, operations on the textures in the reference plane can be most easily performed on a discrete grid. We now define two operations which map the textured surfaces to such a discrete buffer in the reference plane. Let $\{\mathcal{B}, \mathcal{M}\} = \mathfrak{R}(\mathrm{F}, \mathrm{E}, \mathrm{U}, \mathcal{T})$ denote the result of rasterising a textured mesh, with connectivity defined by faces F, and a 4-channel texture $\mathcal{T}$ into a $w \times h$ buffer $\mathcal{B}$ with 4-component pixels, using columns of E as target vertex coordinates in the reference plane, and columns of U as the texture coordinates. A mask, $\mathcal{M}_{h\times w\times 4}$, which records the pixels of the buffer covered by the rasterised mesh as 1's, with 0's elsewhere is also returned.

Additionally, let $\{\mathcal{B}, \mathcal{M}\} = \mathfrak{R}'(\mathrm{F}, \mathrm{E}, \mathrm{U}, \mathrm{Z}, \mathcal{T})$ similarly denote the result of rasterising a textured mesh, but this time the first three channels of the buffer, $\mathcal{B}(:,:,1{:}3)$, is the result of rendering the textured mesh (equipped with a 3-channel texture), and the fourth channel of the buffer, $\mathcal{B}(:,:,4)$ receives the interpolated values of depth Z, appropriately scaled. Incorporating geometric cues (the depth component) in addition to pixel colours helps to more quickly perform rough alignment in the early stages of registration. This also helps to resolve ambiguities that could arise when parts of the object have similar appearance (*e.g.* left and right eyes). While useful in the early stages of registration, the use of depth information is detrimental in the later stages, where it is not used.

Suppose at some point (at iteration k of the algorithm) the "current" estimate of the model of texture is $\mathcal{R}$. For a mesh $\mathfrak{M}_i$ with the initial embedding E_i and a computed improvement D_{i_k} to this embedding, the quality of alignment (embedding) can be evaluated by comparing the rasterised versions of the embedded mesh with the model $\mathcal{R}$. As advocated in [5], it is preferable to compare the model, warped using the current estimate of the correspondences, with the original undeformed samples: in other words, measuring how well the model of texture "explains" the original samples. Using the above notation, the *local* objective function which evaluates a particular embedding hypothesis $\mathrm{H(P)} = \mathrm{E}_i + \mathrm{D}_{i_k} = \mathrm{E}_i + \mathrm{PQ}_{i_k}$ is

$$C(\mathcal{R}, \mathrm{H(P)}) = S(\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\}, \mathcal{B}_{\mathrm{ref}}), \tag{3}$$

where $\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\} = \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$ is the rasterisation of the *original* flattened mesh, which is to be explained by the deformed model, and $\{\mathcal{B}_{\mathrm{ref}}, \cdot\} = \mathfrak{R}(\mathrm{F}_i, \mathrm{E}_i, \mathrm{H(P)}, \mathcal{R})$ is the model $\mathcal{R}$ warped back to conform to the original mesh in the reference space. Function $S(\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\}, \mathcal{B}_{\mathrm{ref}})$ compares the buffer $\mathcal{B}_{\mathrm{ref}}$ with respect to $\mathcal{B}_{\mathrm{src}}$, such that only pixels masked by $\mathcal{M}_{\mathrm{src}}$ are considered. In our implementation we assume the exponential distribution of pixel intensity errors and use the mean absolute difference between masked pixels for $S(\cdot)$. Note that since $\{\mathcal{B}_{\mathrm{src}}, \mathcal{M}_{\mathrm{src}}\}$ never change, they can be precomputed in advance.

Repeated optimisation of $C(\cdot)$ for one mesh at a time, and evolving the model $\mathcal{R}$ appropriately, optimises the groupwise alignment of the whole ensemble (see discussion in [13, section 3.4.2]), which can be expressed as a *global* objective function:

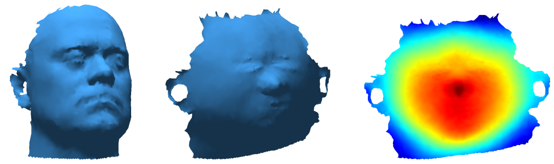

Figure 3. A mesh, its bending invariant, and embedding into $\mathbb{R}^2$.

$$C_{\mathrm{glob}} = \frac{1}{N}\sum_{i=1}^{N} C(\mathcal{R}, \mathrm{E}_i + \mathrm{D}_{i_k}). \tag{4}$$

Note that there is no shape term in our objective function: the smoothness prior of the computed deformations is only implicit in the deformation model. We found that when enough features are present in the textures, no additional shape constraints are needed (see also [5], where the same observation was made, and possible options for the shape smoothness term were also discussed).

2.3. Registration Regime

The first stage of the process is to compute the bending invariants [8] of the meshes using MDS on the pairwise geodesic distances between all vertices in each mesh. The first two components of the bending invariants (or, equivalently, the result of embedding the vertices of the mesh with MDS in $\mathbb{R}^2$) form the initial maximally-isometric embeddings E_i, and the third component, call it *depth*, is kept in a matrix Z_i (Fig. 3).

Note that textures need not necessarily be RGB images, but can in general be any features (vector or scalar) associated with every point on the surface, *e.g.* geometry-based surface descriptors. As advocated in [5], better performance can be achieved if local brightness normalisation is applied to images (textures) and the gradient information is also incorporated as image channels. We also adopt this idea: assume henceforth that textures $\mathcal{T}_i$ are in this form.

The registration begins with a crude alignment of the embedded meshes to a template (say, the first mesh). Since MDS performs the embedding up to a similarity transform, including reflection, this needs to be accounted for. In practice, for the crude alignment we use brute force search to test for the eight possible reflection combinations (by 1 or -1 along each of the three dimensions of the bending invariant) and to approximately estimate rotation (trying all angles in increments of $10°$).

The crude alignment is followed by a groupwise affine alignment stage. This is done in the same fashion as the non-rigid alignment, described below in Alg. 1, except that search is performed for the optimal affine transformation parameters for each embedded mesh, and instead of removing the embedding bias in line 18 the affine parameters are normalised so that the average translation and rotation across the ensemble is 0 and the average scaling is 1.

Henceforth, assume that all E_i are affinely aligned.

Now we address the most important, non-rigid alignment stage. We use the idea from [14] that proved to work well: accumulate the solution additively, gradually composing the resulting optimal embeddings over several iterations.

The non-rigid registration procedure is summarised in Alg. 1. We maintain the improvements to the initial embedding in a matrix D_{i_k} for each mesh. They are initialised to zero (line 1). The iterative body of the algorithm (lines 3–20) is repeated until convergence. Each iteration begins by computing the current estimate of the texture model in the reference plane by rasterising and summing all embedded meshes using the current estimate of the optimal embedding $\mathrm{E}_i + \mathrm{D}_{i_{k-1}}$, lines (5–8). The embedding of each mesh in turn is then improved (lines 9–17). In order to avoid a local minimum around the zero improvement hypothesis, the "current" sample is excluded from the model (lines 11–12), as suggested in [5]. At each iteration a random deformation model is selected, comprising n_b deformation disks on the current 3D mesh. Using the FPS strategy we randomly select n_b mesh vertices with indices $\mathbf{b}_{n_b \times 1}$ as the centres of the disks (line 13). The radii of the deformation disks are chosen such that the adjacent disks overlap by one radius. FPS sampling ensures that the entire area of the mesh is covered evenly. The reason for choosing a random deformation model each is to allow the algorithm the progressively explore the space of all possible deformation models and to exclude to possibility of getting stuck with a poor choice of deformation model, as proposed in [14].

The influence matrix Q describing the effect of the disks on each vertex is then computed (line 14). In practice, to avoid geodesic computations in line 14, we can trade memory for speed. If memory permits, for a given radius r the influences of the deformation disks can be precomputed at each vertex as its potential centre and stored in a sparse matrix. Since the influence of each disk is bounded its influence on most vertices is zero and the above matrix is sparse. (An alternative but less memory efficient way is to precompute the pairwise distances between vertices in each original 3D mesh to avoid geodesic computations in the main loop).

Optimising over all possible parameters $\mathrm{P}_{2\times n_b}$, inducing a hypothetical embedding $\mathrm{H(P)} = \mathrm{E}_i + \mathrm{D}_{i_k} + \mathrm{PQ}$, we compute the optimal improvement $\Delta\mathrm{D}_{i_k}$ to the embedding (line 15). In practice, we found the following optimisation scheme to work well. During the first few iterations, when the disks are large, optimise each disk, one at a time (a 2-dimensional optimisation problem). First we perform brute-force search (as in [5]), trying several displacements within a given evaluation budget and selecting the best one. Then the solution is refined with the Nelder-Mead method. At later stages, when the deformation disks become small and the correspondences are already roughly established, the hypothesis is refined by optimising all disks at once using the stochastic optimiser, Simultaneous Perturbation Stochastic Approximation (SPSA), as proposed in [14]. Its advantage of SPSA is that its performance, in terms of the number of objective function evaluations, is relatively insensitive to the dimensionally of the search space. For details on SPSA see [11, 15]. Finally, the computed improvement is learned (line 16).

As the algorithm approaches the solution and the improvement slows down, the number of deformation disks is increased and their radii are accordingly decreased (line 19), to allow the algorithm to finesse the improvements with a progressively detailed deformation model.

After no further improvement is possible, the algorithm returns the optimal embeddings for each mesh: $(\mathrm{E}_i + \mathrm{D}_{i_{k-1}})$. After the registration is complete, correspondences between any point on one mesh and any point on any other mesh are known via the common reference frame. So, for applications that require only the correspondences to be found nothing else needs to be done. To build an appearance model from the registered meshes are resampled at corresponding locations yielding a set of topologically consistent meshes and corresponding surfaces.

2.4. Removing Embedding Bias

It is possible that during the non-rigid registration stage the correspondences between the surfaces and the common reference plane may become systematically biased, which is equivalent to common reference space becoming distorted. To preclude this from happening, this embedding bias is removed (line 18) by adjusting the improvements D_i, so as to annihilate the bias. First, a point cloud A is formed by concatenating all $(\mathrm{E}_i + \mathrm{D}_{i_k}), \forall i$ from all meshes. Each embedded mesh $\mathfrak{M}_i$ in turn is sampled to determine the displacements at points A due to $\mathfrak{M}_i$ (for points in A that lie on the mesh). The contribution from all meshes is then averaged and subtracted from the displacements D_{i_k}. To save on computation time, bias removal need not necessarily be performed after each iteration, but every few iterations instead.

3. Experiments

In order to validate our approach we conducted registration experiments with artificial and real 3D data, including inter-subject registration. Note that there is no prior art addressing this specific problem, and so a direct comparison with existing methods is impossible.

For all experiments we plotted the values of a various alignment quality measure at each iteration as the algorithm progresses, to monitor improvement. There are the values of the C_{glob} from Eq. (4) (MAD), mean average mutual information and normalised mutual information between the texture model and each shape normalised sample (MI and

Algorithm 1 Perform non-rigid registration of an ensemble of textured meshes.

Require: Textured meshes $\mathfrak{M}_i = \{\mathrm{F}_i, \mathrm{V}_i, \mathrm{U}_i, \mathcal{T}_i\}$, their initial embeddings E_i (affinely aligned), depth components of the bending invariants Z_i, $i \in \{1 \dots N\}$

Initialise: $k \leftarrow 1$; $\mathrm{D}_{i_0} \leftarrow \mathbf{0}, \forall i$
while not happy **do**
  Randomly permute the order of meshes.
  $\mathcal{B}_{\mathrm{sum}} \leftarrow \mathbf{0}_{h\times w\times 4}$; $\mathcal{M}_{\mathrm{sum}} \leftarrow \mathbf{0}_{h\times w\times 4}$
  for $i = 1$ to N **do**
    $\{\mathcal{B}, \mathcal{M}\} \leftarrow \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i + \mathrm{D}_{i_{k-1}}, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$
    $\mathcal{B}_{\mathrm{sum}} \leftarrow \mathcal{B}_{\mathrm{sum}} + \mathcal{B}$; $\mathcal{M}_{\mathrm{sum}} \leftarrow \mathcal{M}_{\mathrm{sum}} + \mathcal{M}$
  end for
  for $i = 1$ to N **do**
    $\{\mathcal{B}_{\mathrm{this}}, \mathcal{M}_{\mathrm{this}}\} \leftarrow \mathfrak{R}'(\mathrm{F}_i, \mathrm{E}_i + \mathrm{D}_{i_{k-1}}, \mathrm{U}_i, \mathrm{Z}_i, \mathcal{T}_i)$
    $\mathcal{M}_{\mathcal{R}} \leftarrow \mathcal{M}_{\mathrm{sum}} - \mathcal{M}_{\mathrm{this}}$
    $\mathcal{R}_{i_k} \leftarrow (\mathcal{B}_{\mathrm{sum}} - \mathcal{B}_{\mathrm{this}})_{\bullet}/\max(1, \mathcal{M}_{\mathcal{R}})$
    $\mathbf{b}_{n_b\times 1} \leftarrow \mathrm{FPS}(\mathfrak{G}(\mathfrak{M}_i, \cdot, \cdot), n_b)$
    $\mathrm{Q} \leftarrow \mathrm{influence}(\mathfrak{M}_i, \mathbf{b})$
    Using $C(\cdot)$ from Eq. (3) and with $\mathrm{H}(\mathrm{P}) = \mathrm{E}_i + \mathrm{D}_{i_k} + \mathrm{PQ}$, optimise w.r.t. P to compute the optimal improvement $\Delta\mathrm{D}_{i_k} \leftarrow (\arg\min_{\mathrm{P}} C(\mathcal{R}_{i_k}, \mathrm{H}(\mathrm{P})))\,\mathrm{Q}$
    $\mathrm{D}_{i_k} \leftarrow \mathrm{D}_{i_{k-1}} + \Delta\mathrm{D}_{i_k}$
  end for
  Remove embedding bias, see Section 2.4.
  If improvement becomes slow, increase n_b.
  $k \leftarrow k + 1$
end while
return $(\mathrm{E}_i + \mathrm{D}_{i_{k-1}})$ — the optimal embedding of V_i into $\mathbb{R}^2$, that brings all meshes, $\mathfrak{M}_i$, into alignment.

NMI), and average pixel stack entropy across the shape normalised ensemble. To visually inspect the quality of registration, we also show the evolution of the model of texture and average shape: as the algorithm establishes the correspondences more and more accurately these converge to a true crisp representation of the underlying structures.

Comparison with the ground truth. For this experiment, one mesh was selected as a template and randomly deformed by selecting 32 control points on it, displacing each control point randomly by ± 24 mm (uniformly distributed) and interpolating the deformation with thin-plate splines. The obtained 64 synthetic meshes (examples shown in Fig. 4), with the ground truth correspondences known, were then registered. Figure 5 shows the evolution of the average shape and texture as the registration progressed. In order to evaluate the accuracy of the registration we computed two measures. The average pairwise distance between corresponding vertices in the aligned meshes was 0.838 mm (median 0.633 mm, $\sigma = 0.773$ mm). The algorithm was stopped after 160 iterations (the results would be improved

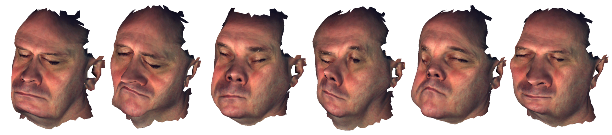

Figure 4. Example meshes from the artificial data set.

Figure 5. Evolution of the mean surface and texture for the artificial data set.

even further if the algorithm was run for longer), see the progress Fig. 12 (bottom left).

We also measured the final spatial errors between every shape-normalised mesh and the template warped to the mean of the shape-normalised meshes. The average pairwise distance between corresponding vertices in the aligned meshes was 0.570 mm (median 0.408 mm, $\sigma = 0.570$ mm).

These results show that our method performed well and converged to within the expected accuracy (subject to the finite number of iterations, flat areas in the texture, and small imperfections due to texture warping).

Within-subject registration. To demonstrate that our algorithm can register a sequence of meshes, we captured a 3D video of two people (PERSON1 and PERSON2) performing various facial actions. We took every fifth frame for each video giving us two sequences of 182 and 221 meshes respectively (examples are shown in Fig. 6 and Fig. 9). The sequences were registered. The progress of registration is shown in Fig. 12 (top row), the evolution of the texture model and the average shape are shown in Fig. 7 and Fig. 10: observe the crisp texture in the final stage of alignment. Having registered the sequences, we built a 3D appearance model for each person. The first three modes of variation are shown in Fig. 8 and 11. The results are excellent, demonstrating the usefulness of our algorithm for automatic 3D appearance model building.

Figure 6. Example meshes from the PERSON1 data set.

Inter-subject registration. To demonstrate that our algorithm can easily handle inter-subject registration, we took a corpus of facial scans of 32 different individuals, 11 of which are women. Some examples from this data set are shown in Fig. 13. Note the degree of variation, both in shape and texture (*e.g.* facial hair). We successfully registered this data set. The progress plot is shown in Fig. 12 (bottom

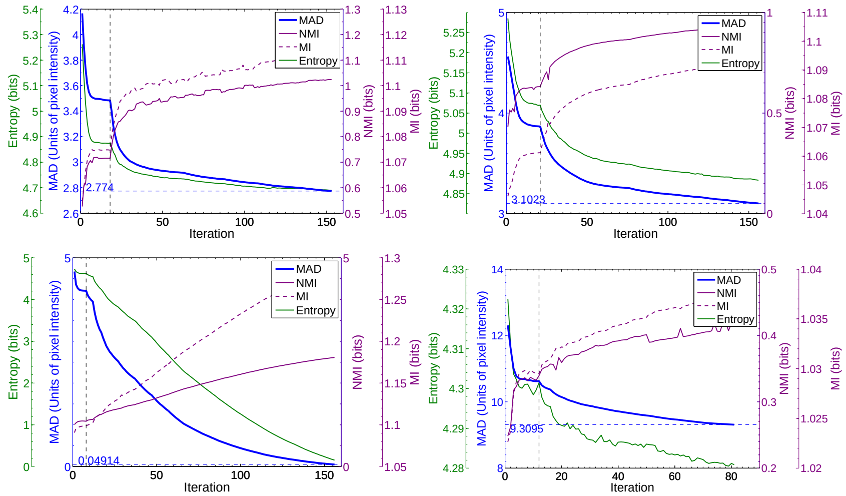

Figure 12. Progress plots for the experiments: PERSON1 (top left), PERSON2 (top right), ground truth artificial data set (bottom left), inter-subject data set (bottom right).

Figure 13. Example meshes from the inter-subject data set.

Figure 7. Evolution of the mean surface and texture for the PERSON1 data set.

Figure 10. Evolution of the mean surface and texture for the PERSON2 data set.

Figure 8. The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered PERSON1 data set.

Figure 11. The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered PERSON2 data set.

Figure 9. Example meshes from the PERSON2 data set.

right), and the evolution of the texture model and average shape in Fig. 15 and Fig. 16. We also built the appearance model from registered samples. The first three modes of variation are shown in Fig. 14. Inter-personal registration is a notoriously challenging problem, with which our algorithm admirably copes.

Figure 14. The first three modes of variation ($\pm 3\sigma$) of the 3D AAM built from the registered inter-subject data set.

Figure 15. Evolution of the texture model in the flat parametric space for the inter-subject data set.

4. Conclusion

We have presented a novel, efficient and reliable, fully automatic method for performing groupwise non-rigid registration of textured surfaces. We have demonstrated its usefulness in accurately establishing correspondences between textured meshes and, especially, in building high quality 3D appearance models. Our method copes with data exhibiting significant variation in shape and texture, such as in the case of notoriously difficult inter-subject registration, with which our algorithm copes admirably.

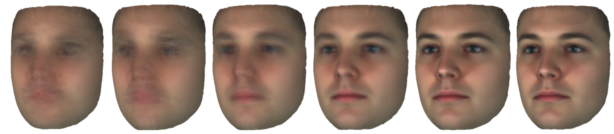

Figure 16. Evolution of the mean surface and texture for the inter-subject data set.

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99*, pages 187–194, 1999.

[2] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant 3D face recognition. In *AVBPA '03*, pages 62–70, 2003.

[3] T. F. Cootes, S. Marsland, C. J. Twining, K. Smith, and C. J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *ECCV '04*, pages 316–327, 2004.

[4] T. F. Cootes, C. J. Twining, V. Petrovic, R. Schestowitz, and C. J. Taylor. Groupwise construction of appearance models using piece-wise affine deformations. In *BMVC '05*, pages 879–888, 2005.

[5] T. F. Cootes, C. J. Twining, V. S. Petrovic, K. O. Babalola, and C. J. Taylor. Computing accurate correspondences across groups of images. *PAMI*, 32(11):1994–2005, November 2010.

[6] R. Davies, C. Twining, and C. Taylor. *Statistical Models of Shape: Optimisation and Evaluation*. Springer Publishing Company, Incorporated, 2008.

[7] R. H. Davies, C. J. Twining, and C. Taylor. Groupwise surface correspondence by optimization: representation and regularization. *Medical Image Analysis*, 12(6):787–796, 2008.

[8] A. Elad and R. Kimmel. Bending invariant representations for surfaces. In *CVPR '01*, pages 168–174, 2001.

[9] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA*, volume 95, pages 8431–8435, 1998.

[10] J. Lötjönen and T. Mäkelä. Elastic matching using a deformation sphere. In W. J. Niessen and M. A. Viergever, editors, *MICCAI*, volume 2208 of *Lecture Notes in Computer Science*, pages 541–548. Springer, 2001.

[11] J. L. Maryak and D. C. Chin. Global random optimization by simultaneous perturbation stochastic approximation. *IEEE Trans. on Automatic Control*, 53(3):780–783, 2008.

[12] G. Peyré and L. Cohen. Geodesic remeshing using front propagation. In *VLSM '03*, pages 33–40, 2003.

[13] K. Sidorov. *Groupwise Non-rigid Registration for Automatic Construction of Appearance Models of the Human Craniofacial Complex for Analysis, Synthesis, and Simulation.* PhD thesis, Cardiff University, 2010.

[14] K. Sidorov, S. Richmond, and D. Marshall. An efficient stochastic approach to groupwise non-rigid image registration. In *CVPR '09*, pages 2208–2213, 2009.

[15] J. C. Spall. Overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[16] S. Wang, X. Gu, and H. Qin. Automatic non-rigid registration of 3d dynamic data for facial expression synthesis and transfer. In *CVPR '08*, pages 1–8, 2008.

[17] S. Wuhrer, C. Shu, J. Boisvert, G. Godin, and P. Xi. Bending invariant meshes and application to groupwise correspondences. In *IEEE Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, pages 382–389, 2009.

[18] A. Zaharescu, E. Boyer, K. Varanasi, and R. P. Horaud. Surface feature detection and description with applications to mesh matching. In *CVPR '09*, pages 373–380, 2009.

[19] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. *CVPR '10*, pages 382–389, 2010.

[20] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Trans. on Visualization and Computer Graphics*, 8(2):198–207, 2002.