Functional Faces: Groupwise Dense Correspondence using Functional Maps

Chao Zhang[1], William A.P. Smith[1], Arnaud Dessein[2], Nick Pears[1] and Hang Dai[1]

[1]Department of Computer Science, The University of York, UK
{cz679, william.smith, nick.pears, hd816}@york.ac.uk
[2]IMB/LaBRI, Université de Bordeaux, France
arnaud.dessein@u-bordeaux.fr

Abstract

In this paper we present a method for computing dense correspondence between a set of 3D face meshes using functional maps. The functional maps paradigm brings with it a number of advantages for face correspondence. First, it allows us to combine various notions of correspondence. We do so by proposing a number of face-specific functions, suited to either within- or between-subject correspondence. Second, we propose a groupwise variant of the method allowing us to compute cycle-consistent functional maps between all faces in a training set. Since functional maps are of much lower dimension than point-to-point correspondences, this is feasible even when the input meshes are very high resolution. Finally, we show how a functional map provides a geometric constraint that can be used to filter feature matches between non-rigidly deforming surfaces.

1. Introduction

Computing dense correspondence between a collection of faces (whether images or 3D meshes) is a fundamental problem in face modelling. It arises in a number of settings including statistical face modelling [2], face morphing [13], motion capture [31], performance driven animation [8] and face transfer [28]. Dense correspondence allows a collection of faces to be treated as a vector space, enabling analysis with, for example, PCA.

There are two distinct classes of dense face correspondence problem. The first version of the problem (which we term *within-person correspondence*) is to compute correspondence between meshes of the same face exhibiting different expressions. This is a non-rigid registration problem which, in principle, has a well defined correct solution. Given enough information, it should be possible to uniquely describe a local region of one scan and find its corresponding region in another (for example, if shape or texture resolves skin pores, these could be explicitly put into correspondence). A special case (and more constrained version) of this problem is where the meshes come from a motion sequence where the face non-rigidly deforms over time. In this context, temporal consistency means that dense correspondence can be viewed as a tracking problem.

The second version of the problem (which we term *between-person correspondence*) is computing dense correspondence between meshes of different faces (i.e. faces with different identities). This is a much harder problem and arguably not well defined. In general, correspondence is a hypothesis of equivalence and defining the objective of the correspondence requires a definition of equivalence. Defining a meaningful notion of equivalence may only be possible in a sparse or low frequency sense. For example, sparse fiducial points can be identified across different faces [25] or it may be meaningful to talk of correspondence between face parts [18] (for example, nose-to-nose or eyebrow-to-eyebrow). In this case, the correspondence in the remaining regions of the face is interpolated (either explicitly or implicitly). An alternate view is to impose some external desirable criterion on the correspondence. For example, we may require that the correspondence is smooth [2] or that it is optimal with respect to an information theoretic measure (e.g. minimum description length [7]). Although practically useful, these choices are in some sense arbitrary and are not justified by any fundamental property of face populations.

In both within- and between-person correspondence, another important distinction is between pairwise and groupwise methods. Pairwise methods compute correspondence between each face in the collection and a reference shape. This includes all template-based methods. On the other hand, groupwise methods explicitly optimise functions that measure the quality of the correspondence across the whole set of shapes simultaneously. The advantage of this is that the result is not dependent on a choice of reference shape or the order in which samples are processed. Also, groupwise information can help resolve ambiguities that would be present in pairwise correspondence. Historically, groupwise approaches to computing correspondence have had limited practical application. This is because the dimensionality of the problem space grows very rapidly with the number of samples in the set. Leading to a very high dimensional nonlinear optimisation problem.

1.1. Contributions

In this paper we present a method for dense 3D face correspondence that addresses the scalability of groupwise methods and elegantly handles multiple notions of equivalence between faces. A recent paradigm shift in non-rigid shape analysis is due to "functional maps". The idea is to correspond real-valued functions on the mesh rather than points on the mesh directly [21]. A functional map can be converted to a point-to-point correspondence and they have recently been shown to perform very well for point-to-point shape matching [24]. In this paper, we apply functional maps specifically to the problem of dense 3D face correspondence. In doing so we make a number of contributions.

First, we propose a *groupwise* variant of functional maps. The functional representation overcomes the computational cost associated with groupwise methods in two ways. First, the maps are of much lower dimensionality than the meshes themselves. We show that functional maps of dimension 30 are adequate for high quality correspondence between face meshes containing 250k vertices. Second, the maps can be composed meaning that a minimal subset of maps can be optimised with the remainder being constructed from compositions of these maps.

Second, we heuristically design a set of real-valued functions that are appropriate specifically to the problem of face correspondence. Key to the functional map approach is to select a set of functions that describe intrinsic properties of the surfaces being matched. Previous work has focussed on intrinsic shape properties. However, face texture provides a rich modality for descriptive functions that capture intrinsic properties of the face. We propose a number of functions that are appropriate for face correspondence and discuss how these may be adapted for within-person or between-person correspondence problems.

Finally, we show how functional maps provide a constraint that can be used to filter potential feature matches between faces. Playing much the same role as the fundamental matrix in multiview image feature matching, we use a functional map to remove outliers in dense but noisy feature matches. This enables us to extract a significantly improved point-to-point correspondence than the original ICP approach of Ovsjanikov *et al.* [21].

We apply our method to both between- and within-person facial correspondence problems. This includes high resolution, high quality static facial expression scans where we are able to obtain a correspondence that allows qualitatively convincing expression interpolation.

1.2. Related Work

Dense Face Correspondence A popular approach to computing dense 3D face correspondence is to use a deformable template where, for example, local affine deformations are permitted. These techniques include Optimal Step Non-rigid ICP [1], which assigns an affine transformation to each vertex and minimises the difference in the transformation of neighbouring vertices (i.e. regularises). This algorithm was used by Paysan *et al.* [23] to build the Basel Face Model. The Global Correspondence Optimisation algorithm [17], optimises a non-linear function using the Levenberg-Marquardt (LM) algorithm. Instead of heuristically determining approximated corresponding points, the algorithm simultaneously solves for correspondences, confidence weights, and deformation field within a single (LM) optimisation. Another approach, Coherent Point Drift [19], formulates the non-rigid registration problem as a Gaussian Mixture Model (GMM) estimation problem (using the EM algorithm). Here, one point set represents the GMM centroids, and the other represents the data points. They iteratively fit the GMM centroids by maximizing the likelihood and find the posterior probabilities of centroids, which provide the correspondence probability.

Groupwise Correspondence The problem of groupwise correspondence appears as an important step in shape matching. Recent advances have also shown that this approach often leads to significant improvements compared to pairwise approaches. Davies *et al.* [7] proposed a groupwise method based on optimising an information theoretic objective. In the area of human body shape registration, co-registration [9] aims to solve the multiple shape registration problem by doing shape modelling and alignment simultaneously. Nguyen *et al.* [22] perform groupwise refinement of maps by identifying "bad" maps (those contained in cycles far from identity) and replace them by compositions of other maps. Taking advantage of the correspondence matrix being low-rank and semidefinite Huang and Guibas [11] use consistent semidefinite programming for multiple shape matching. However, point-based methods do not scale well when the shapes to be matched are densely sampled.

Functional Maps on Shape (Image) Collections Map-based representation have received lots of attention recently in both shape matching [10] and image segmentation [29]. Our approach falls in this category. A network of functional maps [10] is proposed to compute consistent functional maps within heterogeneous shape collections where shapes are only partially similar. Although not explicitly designed to tackle groupwise optimisation, Kovnatsky *et al.* [15] seek a pair of bases in which one tries to achieve a diagonal map matrix. Very recently, groupwise functional maps have been considered via non-smooth optimisation over a product of Stiefel manifolds [16]. A basis-free formulation has also been considered [14].

2. Functional Maps

Let $T : M \to N$ be a bijective mapping between manifolds M and N. Given a scalar function on M, $f : M \to \mathbb{R}$, we obtain a corresponding function on N, $g : N \to \mathbb{R}$, by composition $g = f \circ T^{-1}$. The induced transformation $T_F : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$ is the functional representation of the mapping T, where $\mathcal{F}(\cdot, \mathbb{R})$ is a generic space of real valued functions.

Let $\{\phi_i^M\}$ and $\{\phi_i^N\}$ be bases for $\mathcal{F}(M, \mathbb{R})$ and $\mathcal{F}(N, \mathbb{R})$ respectively (in the discrete case, these are basis vectors). We follow Ovsjanikov *et al.* [21] and use as our basis the eigenfunctions of the Laplace-Beltrami operator. Any function $f : M \to \mathbb{R}$ can be represented as a linear combination of basis functions $f = \sum_i a_i \phi_i^M$. The functional mapping T_F can be expressed in terms of these bases as:

$$T_F\left(\sum_i a_i \phi_i^M\right) = \sum_j \sum_i a_i c_{ij} \phi_i^N, \tag{1}$$

where c_{ij} is a possibly infinite matrix of real coefficients.

Given a pair of functions $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$, the correspondence between f and g can be written simply as $\mathbf{Ca} = \mathbf{b}$ with $\mathbf{C}$ being the matrix representation of the functional map and $\mathbf{a}$ and $\mathbf{b}$ the representation of f and g in the chosen bases of M and N. When $\mathbf{C}$ is unknown but a number of corresponding functions on M and N are provided, $\mathbf{C}$ can be found given enough constraints of type $\mathbf{Ca}_i = \mathbf{b}_i$. This is the so called function preservation constraint. In Section 4 we design functions specifically for computing functional maps between faces.

3. Groupwise functional maps

Now consider a collection of faces, i.e. a set of face surfaces M_i $(i \in 1 \dots s)$. If we compute pairwise functional maps between all pairs of faces, we ignore the context provided by the collection as a whole. Previously, this has been addressed by post-processing the pairwise maps to encourage cycles of compositions of maps to be close to the identity. Instead, we replace pairwise map inference with a groupwise objective function that measures the quality of the functional maps across the whole set of faces simultaneously. We show how to optimise all maps simultaneously whilst guaranteeing inversion and transitivity constraints are satisfied (i.e. any cycle of maps will be guaranteed to be equal to the identity). Moreover, we describe a variant that enforces orthonormality of the maps as a hard constraint. Although strict orthonormality implies isometric deformations, we find this to be a useful constraint, even for between-person correspondence.

3.1. Groupwise optimisation

We denote by $\mathbf{C}_{j\leftarrow i}$ (or $\mathbf{C}_{ji}$ for short) the functional map from shape i to j. We denote by $\mathbf{a}_{ip}$ the representation of the pth function in the chosen basis of M_i with $i \in 1 \dots s$ and $p \in 1 \dots t$. The set of unknown functional map matrices is $\mathcal{C} = \{\mathbf{C}_{ij} | i, j \in [1..s] \wedge i \neq j\}$.

The groupwise objective function is as follows:

$$\varepsilon(\mathcal{C}) = \sum_i \sum_j \sum_p \|\mathbf{C}_{ij}\mathbf{a}_{jp} - \mathbf{a}_{ip}\|^2. \tag{2}$$

This is quadratic in the unknown functional map matrices. It could be solved using linear least squares with independent matrix variables. However, in order to satisfy the cycle consistency constraint, we need to impose two constraints between maps leading to a non-convex optimisation.

First, we require that maps in either direction between a pair of faces are the inverse of each other:

$$\forall i, j \in [1..s], \mathbf{C}_{ij} = \mathbf{C}_{ji}^{-1}. \tag{3}$$

Second, the maps are subject to a transitivity constraint:

$$\forall i, j, k \in [1..s], \mathbf{C}_{ki} = \mathbf{C}_{kj}\mathbf{C}_{ji} \tag{4}$$

This ensures that all 3-cycles are identity and, by construction, all n-cycles are also identity.

If we optimise over all transformations and explicitly enforce these two constraints, then this leads to a constrained optimisation with quadratic equality constraints, i.e. a non-convex quadratically-constrained quadratic programming problem. We propose instead to optimise over a minimal subset of the transformations which allows us to express the problem as an unconstrained optimisation yet to guarantee that all constraints are satisfied.

We optimise $\mathbf{C}_{k1}$ for all $k > 1$, where shape 1 is an arbitrarily chosen reference shape. We emphasise that the solution is independent of the chosen reference shape. In order to evaluate the objective function in (2), we can compute the map between any pair of shapes i and j, in terms of maps from shape 1 by using the construction:

$$\mathbf{C}_{ji} = \mathbf{C}_{j1}\mathbf{C}_{1i} = \mathbf{C}_{j1}\mathbf{C}_{i1}^{-1}. \tag{5}$$

Hence, we can construct the set of maps $\mathcal{C} = \{\mathbf{C}_{ij} | i, j \in [1..s] \wedge i \neq j\}$ from the reduced set $\mathcal{R} = \{\mathbf{C}_{k1} | k \in [2..s]\}$ by applying the appropriate construction from (5) which we denote $\mathcal{C}(\mathcal{R})$. In doing so, we guarantee that the constraints in (3) and (4) are satisfied.

3.2. Hard orthonormality constraint

The Stiefel manifold $V_k(\mathbb{R}^n)$ is the set of all orthonormal k-frames in $\mathbb{R}^n$, i.e. the set of all $n \times k$ orthonormal matrices:

$$V_k(\mathbb{R}^n) = \{\mathbf{X} \in \mathbb{R}^{n \times k} | \mathbf{X}^T\mathbf{X} = \mathbf{I}_k\}. \tag{6}$$

If we require our functional map matrices to be orthonormal then $\mathbf{C}_{ij} \in V_t(\mathbb{R}^t), \forall i, j \in [1..s]$.

We can use the same optimisation strategy as in the previous section. Since the product of two orthogonal matrices is an orthogonal matrix, the construction in (5) ensures that any $\mathbf{C}_{ij}$ will be orthonormal, so long as $\mathbf{C}_{k1}$ for all $k > 1$ is orthonormal. There is also an efficiency saving in evaluating the objective function and its gradient since the matrix inverse can be replaced with a matrix transposition. In order to guarantee that $\mathbf{C}_{k1}$ for all $k > 1$ are orthonormal, we optimise over a product of $s - 1$ Stiefel manifolds:

$$\underset{\mathcal{R}\in\prod_{i=1}^{s-1} V_t(\mathbb{R}^t)}{\arg\min} \quad \varepsilon(\mathcal{C}(\mathcal{R})). \tag{7}$$

The objective function could be written as :

$$\varepsilon = \sum_{i,j} \|\mathbf{C}_{ij}\mathbf{P}_j - \mathbf{P}_i\|_F^2 + \alpha \sum_{i,j} \|\mathbf{Q}_j\mathbf{C}_{ij} - \mathbf{C}_{ij}\mathbf{Q}_i\|_F^2 \tag{8}$$

where the first term encodes the functional preservation constraint with $\mathbf{P}_i$ the corresponding functions in shape i, and the second term enforces operator commutativity constraint with $\mathbf{Q}_i$ the low rank approximation of the Laplacian operator. The weight α is used to adjust the importance. $\mathbf{C}_{ij} = \mathbf{T}_i^T\mathbf{T}_j$ is the matrix representation of functional maps from shape j to i, constructed from the minimal set of optimisation variables $\mathbf{T}_i$.

3.3. Implementation and optimisation

Optimisation on a product manifold of Stiefel manifolds is implemented with the Manopt toolbox [3]. Specifically, groupwise cost objective function and the gradient w.r.t. the minimal set of matrix variables are provided. The trust-regions solver is used to solve the manifold optimisation problem. The weight α is set to be 1 in all cases. In our experiment, the optimisation problem involving two 30×30 matrices takes less than 50 seconds to converge on an Intel Core 3.40GHz. Note that the optimisation problem is non-convex since the Stiefel manifold is not a convex set. Hence, a good initialisation is important.

To initialise, we begin by solving the problem without orthogonality constraints, transform each map $\mathbf{C}_{k1}$ to its closest orthogonal matrix and then refine by performing manifold optimisation of the groupwise objective function. Again, an alternative unbiased initialisation would be to use identity matrices. A further alternative is to use each shape in turn as the reference shape and then take the solution which gives lowest error overall (in other words, use a number of different initialisations and pick the best result).

4. Candidate Functions

Key to the functional map framework is the choice of a set of functions that are assumed to be approximately preserved between shapes. In the case of faces, each function can be seen to capture a different notion of equivalence between faces. The functional map framework attempts to satisfy all of these notions simultaneously via the function preservation constraint. In general, we aim to choose smooth functions that will be well approximated by the low frequency basis provided by the truncated eigenfunctions of the Laplace-Beltrami. Below, we propose a range of functions suitable for face correspondence with reference to Figure 1 where we visualise each function on a sample face from the 3DRFE database [26].

4.1. Shape

Prior work on functional maps has almost exclusively used functions derived from intrinsic shape properties. Many of these functions are appropriate for dense face correspondence, both within- and between-person. Expression changes are known to be approximate isometries [4]. This means that functions dependent on geodesic distances or intrinsic curvature should be approximately preserved under expression changes. Between-person, we still expect correspondence between shape features. For example, between regions of high curvature such as brow, nose and lips.

Curvature We use mean (g) and Gaussian (h) curvature and some functions derived from these intrinsic curvatures. Specifically, we follow [6] and use the logarithm of the absolute value of the mean (i) and Gaussian (j) curvature. Shape index [12] is a continuous characterisation of local surface shape and previous work [20] has shown that it is useful for face shape matching problems. Shape index is derived from the principal curvatures and is shown in (k).

Boundaries During face shape capture, certain regions of the face may not be captured. For example, many capture methods do not accurately capture the interior of the mouth, interior of the nostrils or the highly specular eye surface. These regions may be missing from the scan or manually removed. The internal boundaries caused by these holes provides a useful cue for correspondence. For example, in (n) we show a function derived from the mouth boundary (geodesic distance from the closest mouth boundary point).

Shape signatures Following the original work of Ovsjanikov *et al.* [21], we also use the Wave Kernel Signature (l) and Heat Kernel Signature (m) to provide shape-dependent functions. By varying the time parameter, these signatures provide a sequence of functions (we use 100 time steps for the WKS and 30 for the HKS).

4.2. Texture

We are not aware of any previous work using texture-derived functions for computing functional maps. For dense face correspondence, texture provides a rich source of intrinsic face properties and correspondence cues.

Texture and edges First, there is the raw texture itself, i.e. we use the raw RGB channels (p)-(r) as scalar functions on the mesh. For within-person correspondence, we

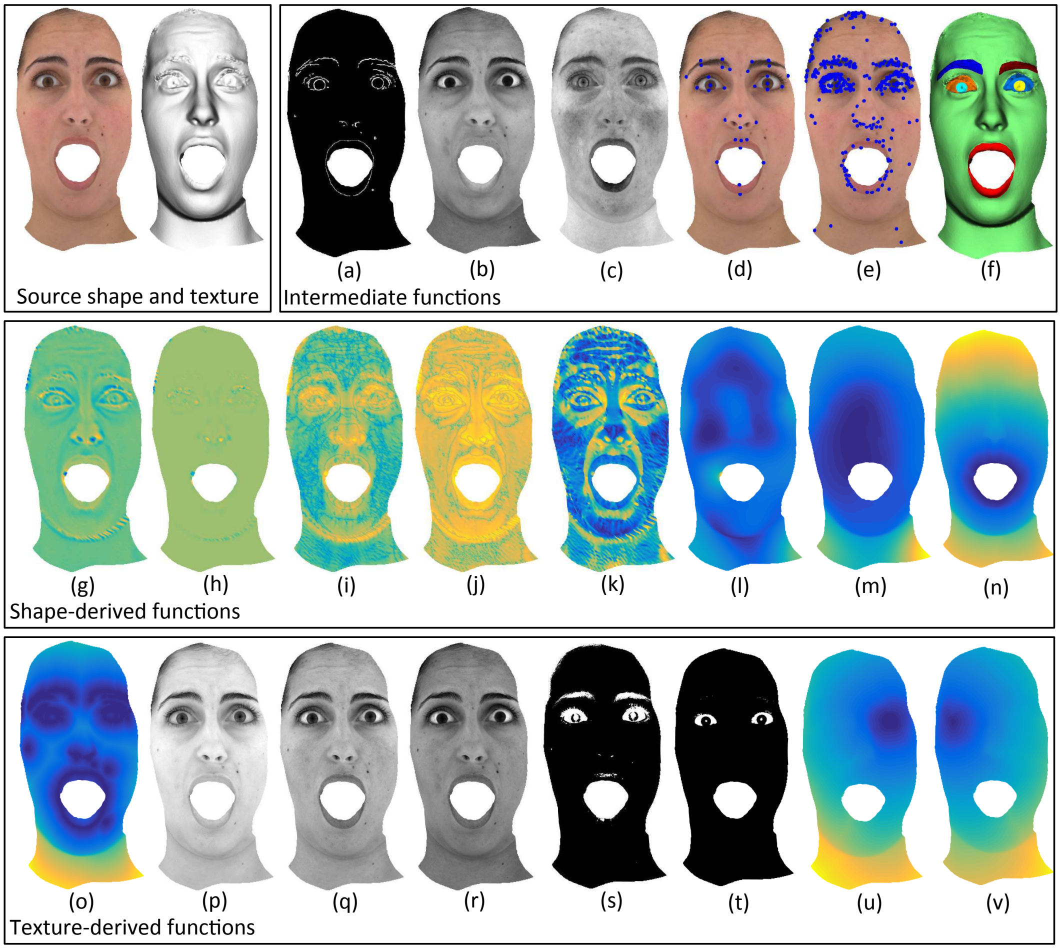

Figure 1. Candidate functions for functional map face matching. See text for details.

expect this to remain exactly constant apart from changes in appearance due to lighting. The 3DRFE dataset that we use was captured in a lightstage meaning that texture is diffuse albedo - an intrinsic property of the surface. Between-person, we do not expect texture to remain constant although there are still useful cues. For example, lips will be redder than the rest of the skin and the whites of the eyes have a consistent colour between people. Nevertheless, it may be more appropriate to preserve texture *edges* rather than texture itself. Intuitively, this will encourage features such as the lips and eyebrows to have consistent boundaries. To do so, we apply an edge detector to the texture in UV space and transfer the binary edges to mesh (a). To represent the edges using a smooth function, we apply a geodesic distance transform to the edges (o) so that each vertex is assigned a value corresponding to the geodesic distance to the closest texture edge vertex.

Segmentations In the same vein, texture provides a useful cue for segmenting a face into semantically meaningful regions (f). One robust way to do this for faces is to use a biophysically inspired colour transformation to identify non-skin regions. Tsumura *et al.* [27] have shown that Independent Components Analysis can be used to estimate melanin and haemoglobin maps from face textures. We show the two ICA channels in (b) and (c) corresponding to melanin and haemoglobin respectively. The haemoglobin map can be used to segment the lips. Further, by thresholding the relative error between the original colour values and their reconstruction using two ICA channels, we can compute binary segmentations (s) that highlight eyebrows and eyes (i.e. non-skin regions). Another simple segmentation is threshold the image saturation which allows the white of the eyes to be located (t). Segmentations can either be used directly as indicator functions or transformed into smooth functions by computing geodesic distance maps. For example, (u) shows geodesic from the right iris segment.

Landmarks Perhaps the most powerful use for texture is in the accurate detection and matching of landmark points. Between-person, landmarks are fiducial points with anthropometric meaning (manually or automatically labelled). The automatic labelling of such points in images has recently received a lot of attention [25]. Applying a facial landmarking algorithm in UV texture space or to a rendered image of the textured mesh, provides fiducial points that can be mapped to the corresponding vertex (d). Since each landmark has the same meaning they are, by design, groupwise consistent. Landmarks can be represented as smooth functions using geodesic distance (v). For within-person correspondence, local features (e.g. SIFT) provide landmarks. In (e), we show SIFT features detected in UV space and mapped to the mesh. We ensure groupwise consistency by removing matches that are not cycle consistent and use the functional map to filter feature matches (see Section 5).

5. Filtering feature matches

Functional maps provide an ideal means to filter tentative feature matches between non-rigidly deforming surfaces for which functional correspondence can be established. We use this to find robust feature matches between two or more meshes of the same face in different expressions.

From an initial estimate of the functional map between two faces, we can filter potential features matches. Suppose that we believe vertex i in face mesh M may correspond to vertex j in face mesh N and that we have an estimate of the functional transformation matrix $\mathbf{C}$ between M and N, then we expect the distance in the embedded functional space to be small:

$$\|\mathbf{C}\mathbf{\Phi}_i^M - \mathbf{\Phi}_j^N\| < \epsilon, \tag{9}$$

where $\mathbf{\Phi}_i^M$ is the ith column of the matrix containing the eigenfunctions of the Laplace-Beltrami of face mesh M.

This is of great practical utility for within-subject correspondence over non-rigid expressions. A high resolution texture map provides numerous local features, for example computed using SIFT. With no information about the non-rigid deformation, there is no obvious way to filter greedy pairwise features matches in order to be geometrically consistent. Equation 9 allows us to restrict the region in which a match is considered reliable. Given a feature at a vertex in face mesh M, we find all vertices whose distance in the functional embedding space is under the threshold. This is illustrated in Figure 2. By choosing an appropriate threshold we can remove unreliable matches. While in principle, this constraint could be used to filter matches between features of any kind (including between-subject), we use the constraint for within-subject correspondence where it is reasonable to expect local features such as SIFT to be preserved between expressions.

In practice, we apply the following steps. First, in order to find landmarks for computing initial functional maps, we extract SIFT features and perform nearest neighbour matching with a conservative matching threshold. This yields a small number of high quality matches. Next, we relax the peak response and distance ratio thresholds yielding a larger number of noisy matches. These are filtered according to Equation 9 to remove geometrically inconsistent matches. These medium dense correspondences can be used to re-estimate the functional map. Finally, we can use the constraint to locally search for less distinctive matches yielding much denser correspondence. These dense matches are used in the point-to-point conversion method in Section 6.

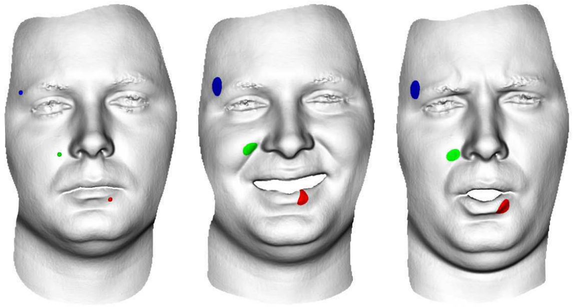

Figure 2. Filtering feature matches: Given feature points on one face (left), we can use the functional maps computed to the other two faces to restrict the region of possible matches (middle and right). A Euclidean distance threshold in the functional embedding space leads to useful feature matching constraints under non-rigid deformation.

In order to remain groupwise consistent, we must enforce some additional constraints in the feature matching. First, we use only mutually nearest neighbour matches between features. Second, we check for cycle consistency over all 3-cycles in the shape collection. Any features which do not satisfy all 3-cycles are removed. One advantage of matching features groupwise in this manner is that we can get a better estimate of the closest non-match distance for each feature. This allows us to use the same multi-image thresholding criterion as Brown *et al.* [5], further improving robustness.

6. Point-to-point conversion

Ultimately, for applications such as face morphing, statistical modelling or motion capture, we wish to compute a dense point-to-point correspondence. A functional map can be transformed to a dense point-to-point correspondence in a straightforward manner. The naive approach is to pass an indicator function through the map and select the vertex with maximal value. Concretely, to find the point $y \in N$ that corresponds to $x \in M$, we chose y as the point at which $g(y)$ obtains a maximum, where $g = T_F(\delta_x)$ and δ_x the delta function around $x \in M$. This is computationally expensive. A better alternative is to perform nearest neighbour (NN) matching in the embedded functional space.

Observe that the delta function δ_x has coefficients $a_i = \phi_i^M(x)$ in the basis of M. Hence, given a matrix $\mathbf{\Phi}^M \in \mathbb{R}^{t \times v}$, where v is the number of vertices, and a functional map matrix $\mathbf{C}$, the image of all delta functions centred at points of M is given by $\mathbf{C}\mathbf{\Phi}^M$. Finding corresponding maxima amounts to performing NN matching between the columns of $\mathbf{C}\mathbf{\Phi}^M$ and $\mathbf{\Phi}^N$, i.e. in t-dimensional space.

This is the approach taken in the original work of Ovsjanikov *et al.* [21] and we use the same method for finding point-to-point maps for between-subject correspondence. We choose one of the meshes to provide the reference topology, apply NN matching between each mesh and the reference in the functional embedding space and use this correspondence to consistently remesh each input mesh. We note however that NN matching in a relatively high dimensional space of potentially hundreds of thousands of vertices is computationally expensive, even using approximate NN.

For within-subject correspondence, we propose an alternate approach for point-to-point conversion which yields higher quality correspondence. First, we use the functional map and the filtering method in Section 5 to increase the number of landmark correspondences substantially. Next, we add additional point-to-point correspondences by performing NN matching in the functional embedding space between segment boundaries. So, for example, the interior mouth boundary provides two sets of vertices between which we perform NN matching in the embedded space. In a UV texture space, we can then warp each mesh onto a reference using a thin plate spline (TPS) warp computed from the landmark and boundary correspondences. Finally, we can apply subpixel refinement by computing optical flow between the warped texture images. The texture images can be enriched with additional channels containing any of the functions used to compute the functional map. Finally, we warp the mesh coordinate functions onto the reference topology and resample to yield a consistent remeshing. This method provides much improved results but is also much faster since NN matching is only performed on small subsets of vertices and the only additional cost is applying a TPS warp and (optionally) optical flow in texture space.

7. Experiments

We use several datasets for our evaluation. For between-person correspondence we use the 10 out-of-sample meshes provided with the Basel Face Model [23]. These meshes have been put into dense correspondence using non-rigid ICP fitting of a template mesh and include textures, enabling qualitative comparison. For within-person correspondence we use a subset of the 3DRFE [26] dataset. These models were captured using a lightstage and are of very high quality. The resolution of each mesh is over 250k vertices and the texture is diffuse albedo, obtained using cross-polarised illumination. These meshes have genus 1 as the mouth has been manually removed. For quantitative evaluation, we choose 6 scans of the same person from the BU3D [30] database. In all cases, we use functional maps of dimension 30 × 30.

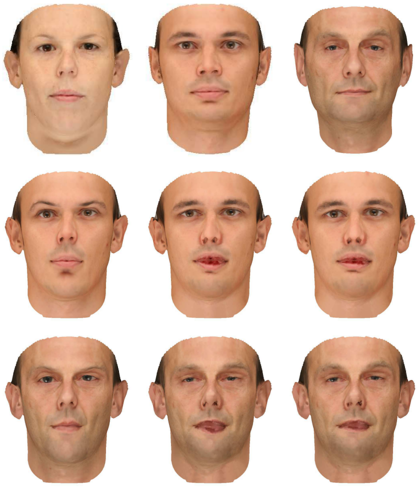

Figure 3. Texture transfer results on BFM meshes [23]. Row 1: original textures. Rows 2 and 3: textures transferred from shape 2 and 3 to shape 1 respectively. Col. 1: non-rigid ICP [1]; col. 2: pairwise functional maps; col. 3: groupwise functional maps.

7.1. Between-person correspondence

In order to evaluate between-person correspondence we visualise texture transfer results allowing qualitative assessment of the correspondence quality. In Figure 3 we compare texture transfer results for the pairwise version of our proposed method, the groupwise version and non-rigid ICP [1]. Non-rigid ICP introduces some artefacts around the eyes and eyebrows where the result lacks symmetry. The functional map results are improved in the eye regions but show some artefacts around the lips, though the groupwise variant slightly reduces these. We believe that the source of these errors is the reliance on nearest neighbour matching in the functional embedding space, particularly as the deformations are non-isometric in this case.

7.2. Within-person correspondence

We now evaluate within-person correspondence. Again, in Figure 4 we provide qualitative evaluation via texture transfer and provide comparison with non-rigid ICP [1]. Under large non-rigid deformations, non-rigid ICP introduces large artefacts in the eyebrows and lips. Groupwise functional maps with simple nearest neighbour matching improves in the eyebrow region but still shows problems in the lip region. Groupwise functional maps with our proposed point-to-point conversion using filtered feature matches provides a convincing texture transfer.

Following [8] we can evaluate this correspondence quantitatively under the assumption that texture remains constant under expression changes. By measuring the variance of the colour at each vertex over the expression changes, we get a measure of correspondence quality. We expect the variance to be zero under perfect correspondence. In other

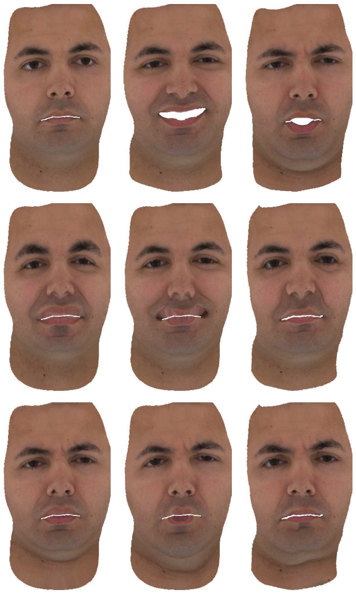

Figure 4. Texture transfer results on 3DRFE [26]. Row 1: original textures. Rows 2 and 3: texture transfer results from shape 2 and 3 to shape 1 respectively. Col. 1: Non-rigid ICP [1]; col 2: groupwise functional map with nearest neighbour point-to-point; col 3. groupwise functional map with feature match warping.

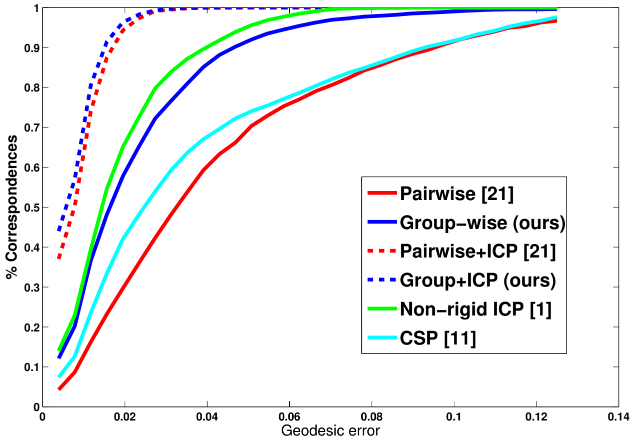

Figure 5. Quantitative comparison of our method with the state-of-the-art methods.

words, a small value of the variance indicates a better correspondence. The variances averaged over colour channels and vertices are reported in Table 1. The functional map methods outperform non-rigid ICP, while requiring several orders of magnitude less running time. The qualitative improvement of our feature match warping method is evident again in the quantitative results. In Figure 5, we also evaluate the correspondence quantitatively and compare to the state of the art methods using 6 meshes from the BU3D [30] database. Our groupwise method outperforms pairwise functional maps [21] and CSP [11] with a large margin. With ICP refinement step, our groupwise method could further improve the performance over [21].

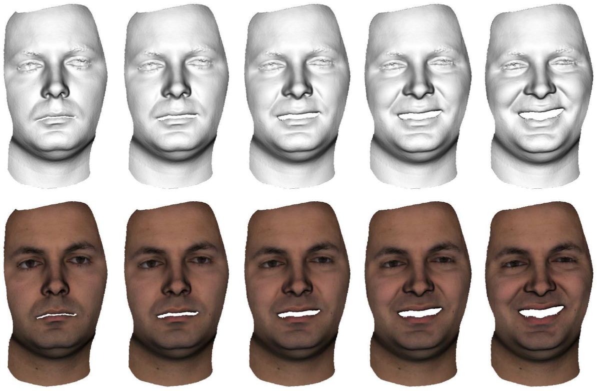

Figure 6. Expression morphing: four step linear interpolation between neutral and smiling face.

Finally, in Figure 6 we show an application of our correspondence by interpolating between expressions. The interpolation is convincing and free of artefacts. See supplementary material for additional results and video.

8. Conclusions

In this paper we have presented a groupwise extension to the functional maps paradigm and designed functions specifically suited to the problem of face correspondence. The method is efficient, flexible and can be used to provide a constraint that enables dense, robust feature matches to be found and exploited during point-to-point conversion.

In future work, we would like to explore additional data modalities for computing intrinsic functions. For example specular albedo and bump/displacement maps. One particularly interesting idea is to use displacements due to expressions (computed as a within-person correspondence problem) as a function for between-person correspondence. Intuitively, the idea is that the displacement of a vertex under semantically equivalent deformations is a feature that can be used for correspondence. We would also like to use the correspondence for statistical modelling where the quantitative evaluation of the model provides another indirect way of evaluating dense correspondence. Finally, we would like to develop a weighting scheme where optimal function weights are found by optimising such an indirect measure of correspondence quality. This will help identify which functions are important for high quality correspondence.

Method	Texture consistency	Running time
Non-rigid ICP[1]	.0103	6 hours
Ours (with NN)	.0101	55 seconds
Ours (Sec.6)	.0099	120 seconds

Table 1. Texture consistency and running time on 3DRFE subset

References

[1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *Proc. CVPR*, pages 1–8, 2007.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, pages 187–194, 1999.

[3] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[5] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proc. CVPR*, pages 510–517, 2005.

[6] É. Corman, M. Ovsjanikov, and A. Chambolle. Supervised descriptor learning for non-rigid shape matching. In *ECCV 2014 Workshops*, pages 283–298. Springer, 2014.

[7] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. Medical Imaging*, 21(5):525–537, 2002.

[8] G. Fyffe, A. Jones, O. Alexander, R. Ichikari, and P. Debevec. Driving high-resolution facial scans with video performance capture. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 34(1), 2014.

[9] D. A. Hirshberg, M. Loper, E. Rachlin, and M. J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Proc. ECCV 2012*, pages 242–255. Springer, 2012.

[10] Q. Huang, F. Wang, and L. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graphic.*, 33(4):36, 2014.

[11] Q. Huang and L. Guibas. Consistent shape maps via semidefinite programming. *Computer Graphics Forum*, 32(5):177–186, 2013.

[12] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10:557–565, 1992.

[13] P. Korshunov and T. Ebrahimi. Using face morphing to protect privacy. In *Proc. AVSS*, pages 208–213, 2013.

[14] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proc. CVPR*, 2015.

[15] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library, 2013.

[16] A. Kovnatsky, K. Glashoff, and M. M. Bronstein. Madmm: a generic algorithm for non-smooth optimization on manifolds. *arXiv:1505.07676*, 2015.

[17] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5):1421–1430, 2008.

[18] M. Mauro, K. Khan, and R. Leonardi. Multi-class semantic segmentation of faces. In *Proc. ICIP*, 2015.

[19] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[20] P. Nair and A. Cavallaro. Matching 3d faces with partial data. In *Proc. BMVC*, 2008.

[21] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graphic.*, 31(4):30, 2012.

[22] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. An optimization approach to improving collections of shape maps. *Computer Graphics Forum*, 30(5):1481–1491, 2011.

[23] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *Proc. AVSS*, pages 296–301. IEEE, 2009.

[24] E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. International Symposium on Vision, Modeling and Visualization*, 2015.

[25] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proc. ICCV workshop on Automatic Facial Landmark Detection in-the-Wild Challenge*, pages 397–403, 2013.

[26] G. Stratou, A. Ghosh, P. Debevec, and L. Morency. Effect of illumination on automatic expression recognition: a novel 3D relightable facial database. In *Proc. Face and Gesture*, pages 611–618, 2011.

[27] N. Tsumura, N. Ojima, K. Sato, M. Shiraishi, H. Shimizu, H. Nabeshima, and Y. Miyake. Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin. *ACM Trans. Graphic.*, 22(3):770–779, 2003.

[28] D. Vlasic, M. Brand, H. H-P. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 24(3):426–433, 2005.

[29] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *Proc. ICCV*, pages 849–856. IEEE, 2013.

[30] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *Proc. Face and Gesture*, pages 211–216. IEEE, 2006.

[31] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, 23(3):548–558, 2004.