As-Conformal-As Possible Surface Registration

Abstract
We present a non-rigid surface registration technique that can align surfaces whose sizes and shapes are different to each other, while avoiding mesh distortions during deformation. The registration is constrained locally as-conformal-as possible such that angles of triangle meshes are preserved yet local scales are allowed to change. Based on our conformal registration technique, we devise an automatic registration technique and interactive registration technique that can reduce user interventions during template fitting. We demonstrate the versatility of our technique on a wide range of surfaces.

1. Introduction
Recently, there has been a growing interest in creating high-quality digital 3D models from 3D scans of real-world objects, driven by demands from industries such as films and manufacturing, e.g., digital doubles of actors for visual effects, digital 3D models for 3D printing and personal fabrication. Recent progress in 3D scanning allows us to capture high-resolution and highly-detailed 3D geometries even for dynamic scenes. Unfortunately, these 3D scans are not readily usable in such applications because scanned surfaces are partial, incomplete and containing holes/noises. In many applications, a complete mesh model with good topology (neatly aligned connectivity edges) is desirable because it is easier to manipulate and to attach texture and bump maps.

Surface registration is thus an inevitable post-process of 3D scanning, which aligns and merges captured surfaces to generate a complete surface model. One effective way to solve this problem is the technique called template-based model fitting that starts from a template mesh and gradually deform it toward the scan. By fitting a template with good topology, we can obtain a high quality mesh model that can be used in a wide range of digital applications. Non-rigid surface registration is also vital when adopting mesh-based retargeting techniques [SP04] or blendshape techniques [WLVP09] to animation, which demands correspondences between the source model and the target model or blendshape models with identical connectivities.

While many techniques have been proposed, non-rigid registration is still a difficult problem, in particular, because of the following challenges:

- **Geometric and semantic consistency**: Because the orientations, sizes, shapes and poses of two surfaces are often not equivalent to each other, even if correspondences are manually specified, it is difficult to align the models globally while also capturing surface details at fine scale. Furthermore, semantic consistency is crucial for nonrigid registration. For example, in the case of face registration, features around eyes, lips and ears must correspond to each other.

- **Mesh-connectivity preservation**: During registration of two surfaces whose geometries are drastically different, the mesh undergoes large deformation. In such a situation, the mesh is susceptible to shear distortions and fold-overs. Obtaining a clean and high-quality mesh that is usable for applications is very challenging.

- **Less user efforts**: The registration technique should not require a large amount of user efforts for specifying many feature points by hands. However, due to the above two chal-

![Figure 1: As-conformal-as possible (ACAP) nonrigid registration. Given only five correspondences (forehead, cheeks and ears), our technique is able to fit the template to the highly-detailed scan with different size. Notice that the result with high-quality mesh connectivities with no fold-overs and less shear distortions.](image_url)
lenges, this is difficult to achieve and previous work requires to specify dozens of landmarks [SP04, YLSL10].

Early nonrigid registration techniques use smoothness regularization [ACP03, SP04, ARV07] to constrain deformations. This approach is flexible enough to align semantically different shapes e.g., sphere to tooth [YLSL10]. Nevertheless, it loses most of the original details and is also poor at preserving mesh structures during large deformation, which requires to specify many correspondences by hand to work correctly. In contrast, the regularization based on isometric (length-preserving) mapping [LSP08, HAWG08] provides strong constraints on deformations, which is often employed in automatic registration. However, this is not suitable for aligning surfaces with different sizes or shapes. Most often, the template mesh and scans do share semantics, but are different in global scales, pose, local scales, and details; in other words, two surfaces are in the same class e.g. registration of different human individuals. Thus, what we need is somewhere in-between.

In this paper, we present the as-conformal-as possible (ACAP) surface registration technique that is flexible enough to non-rigidly align two surfaces with different details and sizes, yet is able to preserve mesh structure of the template. The key idea of our work is to incorporate conformal mapping into a surface registration framework to preserve angles of triangles in the mesh. Unlike the techniques based on isometric mapping, this allows us to adjust local scale of the model freely and at the same time maintain mesh structures. Furthermore, unlike the techniques based on smoothness regularization, this can produce a good initial shape from a small amount of initial correspondences, such that the user intervention can be minimal. Conformal mappings have been used in parametrization [LPRM02] and correspondence algorithms [LF09], but they are often used for flattening the surfaces, which restricts the surfaces to be near zero genus (or at least the surfaces must have the same genus). We, in turn, define a conformal energy directly in 3D space, enabling registration of incomplete surfaces with complex topology.

The rest of the paper is organized as follows. We first review related work briefly in Section 2. Section 3 describes our core algorithm, ACAP surface registration. Next, we introduce a coarse-to-fine fitting strategy for efficient registration in Section 4. In Section 5, based on the ACAP framework, we propose an automatic registration technique beyond isometric deformations and an interactive registration technique that requires only a small set correspondences. The ACAP surface registration technique is tested on various types of scan data, including different expressions and poses (Section 6). We also compared ACAP with other state-of-the-art techniques.

In summary, our main contributions are the following:

• We derive a nonlinear conformal stiffness term and incorporate this term into the nonrigid registration framework, such that we can handle spatially-varying scale changes of the template and the scan while preserving mesh structures during deformation. We also introduce a linear formulation of the conformal stiffness term. The nonlinear term is robust to large deformation, whereas the linearized version is more efficient.

• We propose an automatic registration technique by employing a nonrigid shape matching technique that is robust to non-isometric deformations and data incompleteness. To the best of our knowledge, this is the first automatic nonrigid registration technique that can handle large non-isometric deformations.

• We propose an interactive registration technique that only requires a small amount of user interactions. Because the conformal stiffness can preserve the overall geometry of the template while changing scale, we can provide a good initial shape with at least three correspondences are given. The user can then specify the point correspondences on-the-fly by observing the deformed model until the satisfactory result can be obtained. Thus, the user interventions can be minimal.

2. Background

2.1. Classes of mappings

Here, we briefly review classes of mappings [FH05] as a guide to choose an appropriate mapping for nonrigid surface registration. Suppose that $S$ is a surface and that $f$ is a mapping from $S$ to a second surface $\tilde{S}$. We consider a 3D to 3D mapping case, where a point at position $S$ is deformed to $\tilde{S}$ by a nonlinear function $f$. $\tilde{S} = f(S)$. We define an orthogonal local frame $dS = [dS_1, dS_2, dS_3]$ at $S$, which is deformed to $d\tilde{S} = [d\tilde{S}_1, d\tilde{S}_2, d\tilde{S}_3]$. A $3 \times 3$ local linear transformation $T$ called deformation gradient is calculated as $T = d\tilde{S} \cdot dS^{-1}$. The rotation-invariant measure of deformation, Cauchy-Green stretch tensor, is defined as $C = T^T T$, which is an analogue of the first fundamental form in 2D. Properties of mappings are described as follows (Table 1).

Isometric mappings A mapping from $S$ to $\tilde{S}$ is isometric or length-preserving if the length of any arc on $S$ is the same as that of the corresponding arc on $\tilde{S}$. When mapping is isometric, $C$ is an identity matrix, $C = I_4$. In other words, deformation is locally rigid (rotation), $T = R$.

Conformal mappings A mapping from $S$ to $\tilde{S}$ is conformal or angle-preserving if the angle of intersection of every pair of intersecting arcs on $S$ is the same as that of the corresponding arcs on $\tilde{S}$. When mapping is conformal, the axes of the local frame must be orthogonal and have the same norm. In terms of stretch, it must satisfy $C = s^2 I_4$, where $s$ is a scale. In other words, a local transformation is scale and rotation $T = s R$ i.e., similarity transformation. Thus, every elementary circle is transformed to an elementary circle and a sphere is transformed to a sphere, but their radius are allowed to change from the original values.

Equireal mappings A mapping from $S$ to $\tilde{S}$ is equireal
if every part of $S$ is mapped onto a part of $\tilde{S}$ with the same area. We can represent this by $\det(T) = 1$. It is scale preserving.

**Harmonic mappings** A mapping from $S$ to $\tilde{S}$ is harmonic if the deformation minimizes the Dirichlet energy:

$$E_D(f) = \frac{1}{2} \| \text{grad}_S f \|^2$$

where $\text{grad}_S$ is the gradient of the surface. Let $f$ be a vector of a function defined on a surface. The solution of the minimization problem is obtained by solving the Laplace equation with some boundary constraints:

$$\Delta_S f = 0$$

where $\Delta_S$ is the Laplace-Bertrami operator. The consequence of this minimization is that the boundary conditions are smoothly interpolated. When mappings are harmonic, because the gradient of a mapping is equivalent to deformation gradient, we are minimizing a local transformation, $\min ||T||^2$.

There are two important implications that describe relationships between the above mappings. First, every isometric mapping is conformal and equiareal (scale-preserving) i.e.,

$$\text{isometric} = \text{conformal} \cap \text{equiareal}$$

Second, conformal mapping is the subspace of harmonic mapping:

$$\text{conformal} \subseteq \text{harmonic}$$

Thus, conformal mappings are always harmonic but the inverse is not true. Not all harmonic mappings are angle preserving.

From these discussion, for the nonrigid surface registration of different shapes (different scale and details), we should avoid isometric mappings due to their scale preserving property. Harmonic mappings are also not desirable because it is not angle-preserving (produce shear distortions). In this paper, we show conformal mappings are most appropriate for registration of the different shapes within the same class.

### 2.2 Previous work

**Nonrigid surface registration techniques** Previous surface registration techniques usually adopts two types of regularizations: isometric mapping and smooth deformation (harmonic mapping and its variants). The advantage of the techniques based on isometric mapping [LSP08, HAWG08, TBW*09] is that they can achieve automatic registration. Li et al. [LSP08] achieved isometric registration using the deformation model of [SSP07] that constrains local linear transformations as orthonormal $T^T T = I_d$. Huang et al. [HAWG08], on the other hand, constrain transformations locally as-rigid-as possible. However, these approaches are incapable of handling the models that have different sizes or that undergo large local stretching. In contrast, the techniques based on the smoothness regularization — harmonic mapping (membrane model) [WLVP09], bi-harmonic mapping (thin-plate model) [YLSL10] and deformation smoothness [ACP03, SP04, ARV07] — are robust to size differences. Nevertheless, they are too weak to preserve mesh structures from shear distortions and lose most of the details of the template. Thus, it requires many landmarks to obtain a good initial shape.

**Conformal mapping** In the mesh parameterization field, conformal or angle-preserving mapping is extensively studied and is used to devise efficient low-distortion flattening methods [LPRM02, GY03, DMA02, KSS06, SSP08]. Recently, an algorithm for 3D volumetric mapping [PP12] is also proposed. Conformal flattening is also utilized in shape correspondence algorithms [LF09, KLF11]. Unfortunately, these techniques require the topology of the template and the target to be the same. Thus, they are not really applicable to real-world 3D scans that contain holes and other artifacts. In contrast, our technique is applicable to such scans by constraining transformations to conformal directly in three dimensional space.

The most similar techniques to ours are Liao et al. [LZW*09] and Papazov et al. [PB11] who achieved surface registration by constraining deformations as similarity transformations. [LZW*09] uses a linear surface deformation model [SCOL*04]. Thus, it does not handle large rotations. [PB11] is based on shape matching, which starts from a rigidly-aligned template and does not incorporate landmark constraints or smoothness regularization (instead, they directly smoothed target positions). Thus, it is difficult to handle large changes in pose or shape and to capture details of the target. In contrast, our formulation is based on conformal mapping, where we optimize affine transformations associated to vertices by enforcing the conformal and smoothness constraint directly on the transformations. This allows us to handle large deformations.

### 3. As-conformal-as possible surface registration

We propose a surface registration technique that is flexible enough to non-rigidly align two surfaces with different sizes and details, yet is able to preserve mesh structures of the template. Our goal is to fit the template to scans that share similar semantics but are different in global scales, pose, local scales, and details (two surfaces are in the same class e.g. registration of different human individuals).
3.1. Notation and overall cost function

The template mesh consists of $n$ vertices and $m$ triangle faces. The vertex positions of the template, $v_1 \ldots v_n$, is denoted by a $n \times 3$ vector, $v = [v_1 \ldots v_n]^T$. The vertex positions of the scan is denoted by $p_1 \ldots p_m$. The registration is expressed as a set of $3 \times 4$ affine transformation matrices $X_i = [T_i, t_i]$ that is associated with each vertex $v_i$ of the template, where $T_i$ is a linear transformation and $t_i$ is a translation. Transformation $X_i$ is concatenated into a single $4n \times 3$ matrix $X = [X_1 \ldots X_n]^T$.

We define the cost function $E(X) = w_{ACAP}E_{ACAP} + w_{reg}E_{reg} + w_CE + w_FE$, where $E_{ACAP}$ constrains deformation as-conformal-as possible, $E_{reg}$ acts as a regularization term to avoid extreme local deformation, $E_C$ penalizes distances between the closest points of template and target surface and $E_F$ penalizes distances between the feature points of template and target surface.

3.2. Conformal stiffness term

Nonlinear conformal energy The $E_{ACAP}$ term penalizes the deviation of each transformation from conformal mapping. Recall that conformal mapping constrains a local transformation as:

$$
T = \begin{bmatrix}
  s & -h_3 & h_2 \\
  h_3 & s & -h_1 \\
  -h_2 & h_1 & s
\end{bmatrix}
$$

where $c_1, c_2$ and $c_3$ are the $3 \times 1$ column vectors of $T$.

Linearization Because $E_{ACAP}$ is nonlinear, it is, in general, expensive to minimize this energy. Following the linear approximation of a linear transformation used in the Laplacian surface editing framework [SCOL'04], we derive the linearized version of $E_{ACAP}$. In [SCOL'04], a linear transformation is approximated as:

$$
T \approx \begin{bmatrix}
  s & -h_3 & h_2 \\
  h_3 & s & -h_1 \\
  -h_2 & h_1 & s
\end{bmatrix}
$$

We instead enforce a constraint on a transformation as:

$$
E_{LACAP} = \sum_i L_{Conformal}(T_i)
$$

LConformal($T_i$) = $\|T_{11} - T_{22}\|^2 + \|T_{22} - T_{33}\|^2 + \|T_{33} - T_{11}\|^2 + \|T_{12} + T_{21}\|^2 + \|T_{23} + T_{32}\|^2 + \|T_{31} + T_{13}\|^2$

This term constrains the diagonals of $T$ to be the same and the off-diagonals to satisfy $T_{ij} + T_{ji} = 0$. Note that this constraint is equivalent to a 3D extension of Cauchy-Riemann equation [PP12], where they relates partial derivatives of function whereas we relate the components of the deformation gradient. This formulation is only valid for a small rotation angle. Thus, we will use this energy at the last stage of registration where deformations should be small.

3.3. Regularization and position constraints

Regularization term We combine two energies for regularization: $E_{reg} = E_{consist} + E_{smooth}$. The role of the first energy is to make the problem well-posed and the second one avoids extreme local deformation. The first energy makes a linear transformation and translation consistent.

$$
E_{consist} = \sum_{i} \sum_{j \in N(i)} \|T_i(v^j_n - v^0_n) + T^0_j + t_i - (v^0_n + t_j)\|^2
$$

where $N(i)$ consists of one-ring neighbors of vertex $i$. $v^0_n$ and $v^0_j$ are vertices of the template. The second energy term serves as a regularizer for the deformation by indicating that the linear transformations of adjacent vertices should agree with one another:

$$
E_{smooth} = \sum_{i} \sum_{j \in N(i)} \|T_i(v^j_n - v^0_n) + T_j(v^0_n - v^0_j)\|^2
$$

Closest point constraints In order to attract the mesh to the scan, we find the closest point matches of the template and the target surface, $C = \{(v_1, p_{idx(1)}) \ldots (v_n, p_{idx(n)})\}$, based on the nearest neighbor search, where $idx(i)$ be the index of the scan point that is matched with vertex $i$. If the distance between the points exceeds $D$ or if the angle between the normals of the points exceeds $\Theta$, we eliminate that point pair from correspondence set $C$.

To avoid extreme deformations in tangential directions, the displacement is projected to the direction of the template normal. Let $\hat{v}_i$ and $\hat{n}_i$ be the position and normal of template vertex $i$ of the current mesh. Then, displacement is $\alpha_i\hat{n}_i$ where $\alpha_i = (p_{idx(i)} - \hat{v}_i) \cdot \hat{n}_i$. The closest point term is defined as:

$$
E_C = \sum_{i \notin C} \|v_i - (\hat{v}_i + \alpha_i\hat{n}_i)\|^2
$$

Feature point constraints The feature point term is established as soft constraints such that we can control the contribution of this term to the overall energy. The feature correspondences are established automatically or by the user. Suppose $n_f$ feature points are specified. Then the feature point term is defined as follows:

$$
E_F = \sum_{j=1}^{n_f} \|v_{idx(l)} - p_l\|^2
$$

where $p_l$ is the position of $l$th feature point and $idx(l)$ is the index of the corresponding vertex of the template model.
3.4. Optimization

The optimization consists of two loops: The outer loop searches for the nearest neighbor points and constructs the closest point term with \( w_{reg} \) set to some value. The inner loop then optimizes the affine transformations at the vertices with the fixed position constraints by minimizing \( E(X) \). Once this is converged, \( w_{reg} \) is halved and the outer loop finds the closest points again.

Nonlinear least squares We minimize the nonlinear energy \( E(X) = w_{ACAP}E_{ACAP} + w_{reg}E_{reg} + w_CE + w_FE_F \) using an iterative Gauss-Newton method [SSP07]. We unrolled \( X \) and define stacked variables by a \( 12n \times 1 \) column vector \( x \). The Gauss-Newton algorithm linearizes the nonlinear problem with Taylor expansion about \( x \):

\[
f(x + \delta) = f(x) + J\delta
\]

The vector \( f(x) \) stacks the squared roots of the cost functions, so that \( f(x)^T f(x) = E(x) = w_{ACAP}E_{ACAP} + w_{reg}E_{reg} + w_CE + w_FE_F \). \( J \) is the Jacobian matrix of \( f(x) \). At each iteration \( t \), we solve a linearized problem and compute an updating vector \( \delta \) to improve the current solution \( x \):

\[
J_t^T J_t \delta_t = -J_t^T f(x_t)
\]

In each Gauss-Newton iteration, we solve the normal equations by Cholesky factorization. We must calculate both the symbolic and numeric factorization of \( J_t \) once after the outer loop finds the closest points. In the inner loop, however, the non-zero structure of \( J_t \) remains unchanged. Thus, we can reuse the symbolic factorization to speed up computations. The inner loop typically takes 6 iterations until convergence.

Linear least squares The linearized conformal registration energy \( E(x) = w_{ACAP}E_{ACAP} + w_{reg}E_{reg} + w_CE + w_FE_F \) is minimized in a linear least squares sense as:

\[
A^T A x = A^T b
\]

where the left hand side \( A \) and the right hand side \( b \) are defined from the constraints.

3.5. Relation to as-similar-as possible energy

As we can see from Table 1, we can also achieve conformal mapping by constraining local linear transformations to similarity transformations, i.e., minimizing the as-similar-as possible (ASAP) energy:

\[
E_{ASAP} = \sum_{i} ||T_i - \tilde{s}_i R_i||_F^2
\]

where scale \( \tilde{s}_i \) and rotation \( R_i \) are calculated from the current transformations. Within our framework, this energy can be minimized by alternating between the optimization of affine transformations and the calculations of similarity transformations using polar decompositions.

Note that \( E_{ACAP} \) and \( E_{ASAP} \) are basically equivalent. However, as they employ different optimization techniques, differences appear in deformation quality, convergence and a single iteration time. In Section 6, we will compare ACAP and ASAP and show that it is better to use ACAP than ASAP for the surface registration problem.

4. Coarse-to-fine fitting strategy

To achieve registration efficiently and robustly, we take a coarse-to-fine fitting strategy. This not only improves performance but also reduces risks of generating fold-overs. We use two slightly different techniques. The first strategy fits a coarse template to the scan, performs subdivision and then fits the resulting mesh to the scan again. The other strategy incorporates space deformation and performs an efficient subspace deformation technique by layering a coarse graph under the dense mesh. Once subspace deformation is done, we perform registration on the original dense resolution.

4.1. Fitting Steps

The fitting steps are summarized as follows:

Step1: Initial fitting The template is first roughly fitted to several feature points. We at least require three correspondences to achieve the initial fitting. At this stage, affine transformations are associated to vertices of a coarse mesh (graph) and the overall size of the template is adjusted. Here, we use the nonlinear conformal constraint to handle large rotations.

Step2: Mid-scale fitting After fitting the template roughly to the scan using feature points, the mesh is deformed gradually toward the scan. Again, affine transformations are associated to vertices of the coarse mesh (graph) and optimized nonlinearly. At this stage, we focus on adjusting local scales and we do not aim at capturing details.

Step3: Subdivision (optional)

Step4: Fine fitting In this stage, the focus is on capturing details. Thus, affine transformations are associated to vertices of the dense (original or subdivided) mesh. Here, to improve efficiency, we minimize the linear conformal registration energy.

4.2. Subdivision

The strategy based on subdivision starts from a coarse mesh with several hundred vertices. We first align the coarse template and the scan using ACAP registration. We then subdivide the deformed mesh to generate a dense result. Any subdivision method can be used here. We use Loop’s subdivision technique [Loo87]. Once we obtain a dense mesh, we perform registration again to attract the vertices of the dense mesh to the scan surface.

4.3. Subspace deformation

To align a dense template efficiently toward a scan, we use a subspace deformation technique called embedded deforma-
tion [SSP07]. The embedded deformation technique layers coarse graphs under the dense mesh and solves the problem in the reduced space, which can significantly speed up computations. An affine transformation is associated with each vertex in the coarse graph. Each vertex in the dense mesh is assigned skinning weights and its deformed position is approximated from transformations of coarse graphs.

The vertex position of the coarse graph is represented by $v_k$. The linear transformation and the translation associated with $v_k$ is denoted by $T_k$ and $t_k$, respectively. Let $v^0_k$ be the vertex position of the coarse mesh in the rest state. The deformed vertex $\bar{v}_i$ is obtained as follows:

$$\bar{v}_i = \sum_{k=1}^{c} w_i^{(k)} [T_k(v^0_i - v^0_k) + v^0_k + t_k]$$

where $c$ is the number of vertices in the coarse graph. $w_i^{(k)}$ is a weight for vertex $i$, controlling how much $v_i$ is influenced by $v_k$.

The coarse graph can be established using mesh simplification techniques or the farthest point sampling strategy. The spatially varying weight $w_i^{(k)}$ for each vertex is computed by a $K$-nearest approach as:

$$w_i^{(k)} = \left(1 - \frac{d_{ik}}{d_{\text{max}}}\right)^2$$

and then normalized to sum to one. Here, $d_{ik}$ is the distance from point $i$ to point $k$ and $d_{\text{max}}$ is the distance to the $K+1$-nearest node. We empirically determined $K=8$.

There are several choices for distance function, e.g., Euclid, geodesic, diffusion. Euclid distances are fast and easiest to compute but its computational cost is high. Diffusion distances mimic geodesics and they are relatively easy to compute. We use Euclid distances for the face model, as it is rarely the case that geodesically far apart regions are close in Euclidean space. As for the body model, we use diffusion distance [dGGV08].

Now the feature point constraints are enforced as:

$$E_F = \sum_{l} ||\mathbf{p}_{\text{idx}}(l) - \mathbf{p}_l||^2$$

With this, the user can specify landmarks anywhere freely on the dense mesh.

### 4.4. Weights and parameters

In every stage, $D = 0.02 \times r_{\text{box}}$, where $r_{\text{box}}$ is the bounding box diagonal, and $\Theta = 90^\circ$ are used. In the initial fitting stage, the closest point term is ignored and a large value for the weight of the landmark term is chosen: $w_{\text{ACAP}} = 1000$, $w_{\text{reg}} = 1000$, $w_{\text{C}} = 0$ and $w_{\text{F}} = 10^5$. In the mid-scale fitting stage, we set $w_{\text{C}} = 10$ and $w_{\text{reg}}$ is halved until $w_{\text{reg}}$ reaches to 1. In the fine fitting stage, we again took the same procedure with $w_{\text{F}} = 1$.

![Figure 2: We provide a simple user interface (UI). The user first selects several (at least three) correspondences. The system outputs an initial fitting result. The user then adjust stiffness using a slider bar (Mid-scale fitting). The user will then add and remove correspondences by observing the result (Feature refinement). If the semantic correspondences are satisfactory, the process is terminated. Finally, the registration is performed off-line on the dense resolution.](image)

### 5. Applications

Based on ACAP surface registration, we propose two use-case scenarios for surface registration that reduces the user interventions during template fitting.

#### 5.1. Automatic registration

The first step of the automatic registration is to establish point-to-point correspondences of two surfaces. In principle, any matching technique can be employed. However, in our case, the shape and size of the target surface differs from that of the template. The target scan also contains holes. We thus use a non-rigid shape matching technique that is robust to data incompleteness and non-isometric deformations [sup].

Once correspondences are established, we feed them into the ACAP registration framework as feature point constraints to achieve initial fitting. We next iteratively deform the mesh towards the target by incorporating the closest point constraint. The above technique allows us to align the template and a scan in different shapes/poses automatically.

#### 5.2. Interactive registration

Although our automatic registration technique can be carried out without any user interactions, it is in practice often the case that fully automatic matching algorithms output low quality correspondences or miss matches (see Fig. 9 Right: mid-scale fitting). Therefore, user specification is still the most reliable way to provide correspondences. Thus, designing means for reducing the effort required for this task is very important. Also, this is also beneficial for modifying low quality correspondences produced by automatic methods. We therefore design an interactive user interface to help the users specify the correspondences between the template and target shapes in an intuitive manner. The proposed interactive system helps users freely add, remove and modify correspondences while observing the result on-the-fly. The user can also change the stiffness interactively with slider bars.

The typical usage is illustrated in Figure 2. The user first
selects several (at least three) correspondences. The system outputs an initial fitting result. The user then adjust stiffness using a slider bar (Mid-scale fitting). The user will then add and remove correspondences by observing the result (Feature refinement). If the semantic correspondences are satisfactory, the process is terminated. Finally, the registration is performed off-line on the dense resolution (Fine fitting).

During the interactive registration, we only optimize transformations of the coarse graph. Also, the closet points are searched only from the vertices of the coarse graph. After interactive registration, we perform surface registration off-line on the dense mesh. These strategy allows for registration of a large mesh. Note that, with the use of subspace deformation, the user can specify feature point constraints freely on the dense mesh and move them interactively.

6. Experiments

We tested our method on a wide variety of surfaces, i.e., face and whole body models with different expressions/poses. Most of the scans contain many holes, occluded regions and noises. Statistics of the models are shown on Table 2. We compare our method with other state-of-the-art methods qualitatively and quantitatively.

Wide range of models  Figures 1, 3, 4 and 7 show the results of interactive registration. Figure 9 shows the result of automatic registration. From the smooth template, shapes of hairs and ears are captured (Figs 1 and 7). Notice how the connectivity is preserved nicely after large deformation (Fig. 1). Furthermore, thanks to the conformal stiffness that automatically adjusts local scales, occluded regions and holes are filled in a visually-pleasing manner. With our registration technique, large deformations due to joint extensions/flexions can be handled without generating low-quality triangles (Figs. 4 and 9).

Comparisons  We compare our technique (ACAP, LCAP and ASAP) with other state-of-the-art algorithms: [SSP07, LZW∗09, PB11, SP04, ARV07]. [SSP07] is an isometric counterpart of ACAP, which we replaced the nonlin-

Table 2: Statistics.

<table>
<thead>
<tr>
<th>Template</th>
<th>Scan</th>
<th>#CV</th>
<th>#DV</th>
<th>#SV</th>
<th>#L</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1</td>
<td>Hair</td>
<td>537</td>
<td>15k</td>
<td>100k</td>
<td>5</td>
<td>Fig. 1</td>
</tr>
<tr>
<td>Head1</td>
<td>Ear</td>
<td>537</td>
<td>15k</td>
<td>50k</td>
<td>7</td>
<td>Fig. 2</td>
</tr>
<tr>
<td>Head1</td>
<td>Female</td>
<td>537</td>
<td>15k</td>
<td>15k</td>
<td>5</td>
<td>Fig. 3</td>
</tr>
<tr>
<td>Head2</td>
<td>Male</td>
<td>570</td>
<td>10k</td>
<td>100k</td>
<td>4</td>
<td>Fig. 3</td>
</tr>
<tr>
<td>Head1</td>
<td>Laugh</td>
<td>537</td>
<td>15k</td>
<td>14k</td>
<td>16</td>
<td>Fig. 3</td>
</tr>
<tr>
<td>SCAPE</td>
<td>Abhijeet</td>
<td>500</td>
<td>12k</td>
<td>80k</td>
<td>5</td>
<td>Fig. 4</td>
</tr>
<tr>
<td>SCAPE</td>
<td>Gorilla</td>
<td>500</td>
<td>–</td>
<td>15k</td>
<td>6</td>
<td>Fig. 10</td>
</tr>
<tr>
<td>Horse</td>
<td>Camel</td>
<td>1000</td>
<td>–</td>
<td>20k</td>
<td>12</td>
<td>Fig. 6</td>
</tr>
<tr>
<td>Ilya</td>
<td>Abhijeet</td>
<td>500</td>
<td>20k</td>
<td>80k</td>
<td>–</td>
<td>Fig. 9</td>
</tr>
<tr>
<td>SCAPE</td>
<td>SCAPE</td>
<td>500</td>
<td>12k</td>
<td>65k</td>
<td>–</td>
<td>Fig. 9</td>
</tr>
</tbody>
</table>

#CV and #DV indicate the number of vertices on a coarse mesh and dense mesh, respectively. #SV indicates the number of scan’s vertices. #L is the number of the landmarks.
Table 3: Quantitative comparisons. D, A and B indicate Data error [%], Angle error [deg] and Bending error [deg], respectively.

<table>
<thead>
<tr>
<th></th>
<th>ACAP</th>
<th>LACAP</th>
<th>ASAP</th>
<th>SSP07</th>
<th>PB11</th>
<th>LZW\textsuperscript{*}09</th>
<th>ARV07</th>
<th>SP04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hair</td>
<td>0.48</td>
<td>3.3</td>
<td>12.0</td>
<td>0.45</td>
<td>5.3</td>
<td>0.51</td>
<td>3.6</td>
<td>11.5</td>
</tr>
<tr>
<td>Ear</td>
<td>0.40</td>
<td>2.9</td>
<td>5.8</td>
<td>0.42</td>
<td>4.6</td>
<td>7.7</td>
<td>0.45</td>
<td>3.7</td>
</tr>
<tr>
<td>Gorilla</td>
<td>0.25</td>
<td>10.7</td>
<td>14.3</td>
<td>0.31</td>
<td>11.9</td>
<td>20.4</td>
<td>0.31</td>
<td>14.4</td>
</tr>
</tbody>
</table>

Figure 6: ACAP vs [SSP07]. a) Gorilla, b) Hair: The intermediate results of Fig.10 with compatible weight values are shown. [SSP07] does not adjust local scales, which could lead to fold-overs, shear distortions, and slow convergence; c) Horse to Camel: The ability of ACAP to adjust local scales can be clearly seen from around the tail.

Figure 7: Linear vs Nonlinear ACAP on fine fitting.

ear conformal stiffness with the isometric one and used the regularization term presented in this paper. [LZW\textsuperscript{*}09, PB11] are as-similar-as possible (conformal). [LZW\textsuperscript{*}09] uses a linear Laplace surface deformation [SCOL\textsuperscript{*}04] based on implicit optimization. [PB11] is based on shape matching, which starts from rigidly aligning the template to the target. It does not require to specify feature points. [SP04, ARV07] are based on the deformation smoothness regularization. The models used for comparisons are shown in Fig.5. Note that global scales, orientations and positions of the models are pre-aligned. We measure 1) Data error, the average distance from vertices of the deformed template to the corresponding points of the scan relative to the bounding box diagonal, 2) Angle error, the average angle deviation from the template and 3) Bending error, the average deviation in dihedral angles.

For the Hair model that requires relatively small deformations for registration, ACAP, LACAP, ASAP and [SSP07] produce visually similar results. In contrast, [PB11, LZW\textsuperscript{*}09, ARV07, SP04] produces the results with shear distortions and fold-overs. For the Gorilla model, the result of ACAP is almost no distortions, whereas others reveal fold-overs and distortions. LACAP, although being linear, produces quite nice results until registration requires large rotations, e.g., fold-overs occurred around the arms when fitting to Gorilla in different arm pose. [SSP07] generates, at a glance, visually similar results to ACAP but the feet of the Gorilla exhibits fold-overs. The reason for this was that [SSP07] could not adjust the scale differences between the legs of Gorilla and Human. Thus, the heels were deeply nailed into the ground (Fig 6a) and the vertices around the ankles were then attracted to the heels. The ability of ACAP to adjust local scales can be clearly seen when fitting the Horse into the Camel where the size of the tail is adjusted (Fig. 6c). [PB11] is quite fast as it does not require to solve (non-) linear systems but cannot handle large changes in pose and shape. [LZW\textsuperscript{*}09] cannot handle large rotations. We also found that [LZW\textsuperscript{*}09] becomes unstable when the model have boundaries and requires to specify additional features there. [ARV07] is prone to shear distortions and it tends not to work well with feature points that are placed in a coplanar arrangement. [SP04] generates an initial shape with extreme shrinkage, which results in large fold-overs in the final result. Table 6 shows quantitative comparisons. Angle err. of ACAP are the smallest among all the techniques. In addition, the Bending error on the Gorilla example is the lowest of all, which reflects the ability of ACAP to reduce risks of producing fold-overs.

In Fig. 7, we also compare fine-fitting results obtained using the linear and nonlinear conformal constraints (LACAP vs ACAP). The results both look very similar almost without visually noticeable differences. In fact, the difference of Angle err. from mid-scale fitting to fine fitting is only 0.15 deg. We therefore believe that LACAP is sufficiently accurate when used in fine fitting.

Performance We implemented the prototype of our algorithm in Matlab with partially written in C/C++ on an Intel Core i7 3.4GHz 64-bit workstation. CHOLMOD [CDHR08] is used for constructing, factorizing and solving Eqs. (1) and (2). The timing is shown on Table 4. The reuse of the symbolic factorization reduces the time required for a single Gauss-Newton iteration for about 35%. The linearized version of ACAP is approximately five times faster than the nonlinear version for the problem size of fine fitting.

Number of feature points required from users The numbers of feature points required from the user are shown
on Table 2. Previous work [SP04, YLSL10] require to specified 20–70 feature points, whereas our technique requires under 20 points. This is because ACAP provides a good initial shape from a few feature points and can avoid extreme distortions during iterative fitting.

**ACAP vs ASAP** ASAP is an alternative formulation of ACAP, but the quality of its results is lower than ACAP (Fig. 10 and Table 6). In particular, distortions around handles are localized (Fig. 10 Bottom). On the other hand, ASAP is approximately twice faster than ACAP because the system matrix of ASAP stays in place during a single outer loop until the closest point term changes. However, the total iteration count of ASAP is about double of ACAP. Thus, the overall computational time is almost equivalent. Consequently, ACAP is preferred over ASAP for the surface registration problem.

**Shape interpolation** Since the models obtained using our technique have identical connectivities, we can interpolate the models and obtain in-between shapes. We used a shape interpolation technique based on mass-springs systems similar to [MWF+12]. The results are shown in Fig. 8.

**Limitations** One limitation of our algorithm is performance due to the use of nonlinear optimization (nonlinear ACAP). Although our technique can achieve (near-) interactive rates by making careful design choices (analytically building Jacobian, reusing symbolic factorization, coarse-to-fine strategy), it is relatively slower than previous work; according to performance comparisons on mid-scale fitting based on our implementations, [PB11] and [LZW09] are approximately 10 and 3 times faster than ACAP, respectively. Performance of our method might be improved by using the linearized version (LACAP) and having the user to specify rotation constraints to handles.

Another limitation is fold-overs. Although ACAP with the coarse-to-fine strategy is quite robust to fold-overs, it does not guarantee to solve this issue, especially for the model with complex topology. Thus, the fold-over removing technique [YLSL10] or bounded distortion mapping [Lip12] would be useful for solving this problem thoroughly.

---

### Table 4: Timings (in seconds)

<table>
<thead>
<tr>
<th>Meshes</th>
<th>Subdivision</th>
<th>Subspace</th>
<th>Meshes</th>
<th>Subdivision</th>
<th>Subspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Coarse</td>
<td>#Dense</td>
<td>#Target</td>
<td>NN search</td>
<td>GN Regist.</td>
<td>NN search</td>
</tr>
<tr>
<td>Subdivision</td>
<td>537</td>
<td>15k(100k)</td>
<td>50k</td>
<td>0.020</td>
<td>0.022</td>
</tr>
<tr>
<td>Subspace</td>
<td>570</td>
<td>10k</td>
<td>100k</td>
<td>0.050</td>
<td>0.023</td>
</tr>
<tr>
<td>NN search</td>
<td>0.08 (0.376)</td>
<td>0.10 (0.8)</td>
<td>4.5 (43)</td>
<td>0.020</td>
<td>0.022</td>
</tr>
<tr>
<td>GN Regist.</td>
<td>0.08 (0.376)</td>
<td>0.10 (0.8)</td>
<td>4.5 (43)</td>
<td>0.08 (0.376)</td>
<td>0.10 (0.8)</td>
</tr>
<tr>
<td>NN search LS</td>
<td>0.13</td>
<td>0.07</td>
<td>2.7</td>
<td>0.13</td>
<td>0.07</td>
</tr>
<tr>
<td>LS Regist.</td>
<td>0.050</td>
<td>0.023</td>
<td>0.26</td>
<td>0.050</td>
<td>0.023</td>
</tr>
</tbody>
</table>

#Coarse, #Dense and #Target indicate the mesh size. "NN search", "GN", "LS" and "Regist." indicate the time required for the nearest neighbor search, a single Gauss Newton iteration, and a single least squares solve and a single iteration of ACAP registration, respectively.

---

**References**


We then iteratively deform the mesh towards the target by incorporating the closest point constraint. Note that the automatic registration technique is not always successful (see the right figure where the ear and nose are deformed semantically incorrect places due to wrong correspondences).

Figure 9: Automatic registration to scans with holes. Correspondences are served as positional constraints of initial fitting.

Figure 10: Comparisons. Experimental setup is shown in Fig. 5.