The Visual Computer (2019) 35:935–948
https://doi.org/10.1007/s00371-019-01670-1

ORIGINAL ARTICLE

Huber-L_1-based non-isometric surface registration

Tao Jiang[1] · Xiaosong Yang[1] · Jianjun Zhang[1] · Feng Tian[1] · Shuang Liu[1] · Nan Xiang[1] · Kun Qian[2]

Published online: 10 May 2019

Abstract

Non-isometric surface registration is an important task in computer graphics and computer vision. It, however, remains challenging to deal with noise from scanned data and distortion from transformation. In this paper, we propose a Huber-L_1-based non-isometric surface registration and solve it by the alternating direction method of multipliers. With a Huber-L_1-regularized model constrained on the transformation variation and position difference, our method is robust to noise and produces piecewise smooth results while still preserving fine details on the target. The introduced as-similar-as-possible energy is able to handle different size of shapes with little stretching distortion. Extensive experimental results have demonstrated that our method is more accurate and robust to noise in comparison with the state-of-the-arts.

Keywords Surface registration · Huber-L_1 · Non-isometric

1 Introduction

Surface registration has drawn intensive attention in computer graphics and computer vision with massive applications such as film production and computer games. With the development of geometry acquisition technology, 3D scanning systems allow us to capture high-resolution and highly detailed 3D geometries even for dynamic scenes. However, these scanned data often contain noise, especially from commodity depth sensors such as Microsoft Kinect. To tackle this problem, surface registration is introduced to deform a high quality model (template) so that it aligns with the scanned shape (target). Often, the size and details of the template differ from that of the target. Hence, registration methods which are robust to noise and capable of handling different sizes and details have been investigated.

According to the type of deformation mapping [6], the surface registration is generally categorized into two groups: rigid registration and non-rigid registration. The rigid registration [2,3] aims to find a rigid-body transformation between two shapes, and thus, it cannot handle deformable (non-rigid) shapes. The non-rigid registration, including isometric registration and similar registration, is to find a set of local transformations that align two shapes. The isometric registration which deforms the shape in an as-rigid-as-possible (ARAP) manner has been widely used between isometric shapes [7,10,25]. However, it is not capable of handling shapes with different sizes since it tries to preserve the length of edges. To address this limitation, similar registration approaches [8,29] introduce a scale factor into each local transformation and formulate as an as-similar-as-possible (ASAP) energy, which allows local scalability to handle the size difference.

For the accuracy and robustness of registration, the transformation variation and position difference constraints are usually formulated as a smoothness term and a data term, respectively, to measure the smoothness of the neighboring transformation and the closeness of registration shapes, respectively. Most works [1,8,10] use the classic squared L_2-norm on both constraints (L_2-L_2). However, the smoothness term in L_2-norm tends to penalize large transformation variation. It is not suitable for articulated models where large deformation variations exist at their joints. This could also be seen in image processing where discontinuities are allowed to highlight the sharp edges in image denoising [5]. The classic model in image denoising is ROF model [18], where the total variation (TV) term is an L_1-norm-based regularization and the data term is in squared L_2-norm (TV-L_2). Based on ROF model, Yang et al. [27] propose a sparse non-rigid registration method with an L_1-norm regularized on the smoothness term. However, ROF tends to produce over-regularized results as

✉ Xiaosong Yang
xyang@bournemouth.ac.uk

[1] Bournemouth University, Bournemouth, UK

[2] King's College London, London, UK

the L_2-norm strives to distribute errors evenly, thus fitting the result evenly on the noisy parts. To tackle this issue, TV-L_1 model [30] is proposed to efficiently remove the outliers while preserving fine details. Based on TV-L_1 model, Yang et al. [11] propose a dual sparsity registration approach on both position and transformation sparsity, allowing the positional error to concentrate on small regions. However, TV-L_1 model tends to produce piecewise constant results as shown in Werlberger et al. [26]. We propose a Huber-L_1-based non-isometric surface registration to reduce the staircasing artifacts known from TV. A Huber-norm is applied on the transformation variation, which significantly reduces artificial discontinuities and produces piecewise smooth results. Meanwhile, an L_1-norm is applied on the position difference, solving the over-regularized issue that appeared in L_2-norm. The ASAP energy is introduced to handle shapes of different sizes and poses. To reduce the risks of surface foldover, we adopt Laplacian energy to smooth the template on both primal and dual domains. We compare Huber-L_1 to other state-of-the-art models in the experiments on clean, noisy and real scanned data, demonstrating the advantage and robustness of our method.

The main contributions of this work are summarized as follows:

- We propose a Huber-L_1-based non-isometric registration method regularized on transformation variation and position difference. The Huber-L_1 model is solved by the alternating direction method of multipliers (ADMM) with each energy term being represented in matrix form. The proposed model is robust to noise and produces piecewise smooth results with the target's fine details being well preserved.
- We incorporate ASAP energy in the registration method, which is not only able to handle shapes with different sizes but can also reduce local stretch and distortion. Since ASAP energy preserves the overall geometry of the template while allowing local scaling, it demands a few user efforts to provide a good initial shape estimation.
- We introduce Laplacian energy to relax the template on both primal and dual domains, improving the template mesh quality and decreasing the foldover occurrence.

2 Related work

Over last two decades, surface registration has been deeply researched. A complete survey can be referred in [24]. Here, we briefly review the related works on three categories: rigid registration, isometric registration and non-isometric registration.

Rigid registration The most classic algorithm in rigid registration is iterative closest point (ICP) [2]. It alternates between closest point searching and optimal transformation solving. To improve the converge rate of ICP, Low et al. [13] used a point-to-plane metric in which the energy function is the sum of the squared distance between a point and the tangent plane at its correspondence point. One main issue for ICP and its variants is that they are sensitive to outliers and missing data. To tackle this issue, Bouaziz et al. [3] propose a new formulation of ICP using sparsity inducing norms. However, methods in this category cannot deal with deformable shapes registration.

Isometric registration Huang et al. [7] present a non-rigid registration of a pair of partially overlapping surfaces, constraining transformation locally as-rigid-as-possible (ARAP). Rouhani et al. [17] propose a non-rigid registration between two clouds of points, where the source set is clustered into small patches and then deformed rigidly to align with the target. Sussmuth et al. [23] slide the template mesh along the time–space surface in ARAP manner to reconstruct animated meshes from a series of time-deforming point clouds. Wand et al. [25] take a set of time-varying unstructured sample points as input and reconstruct a single shape and a local deformation field. Li et al. [10] introduce a registration algorithm for partial range scans of deforming shapes. They deform local features as rigidly as possible to avoid shearing and stretching artifacts. Yang et al. [11] propose a dual-sparsity-based non-rigid registration, which adds orthogonality constraints on the local transformation to preserve local rigidity.

Non-isometric registration Based on shape matching [15], Papazov et al. [16] compute shape transitions based on local similarity transforms which allow to model not only as-rigid-as-possible deformations but also local and global scalability. Liao et al. [12] reconstruct complete 3D deformable models over time by a single depth camera. They constrain local transformation to isotropic scales and rotations. Amberg et al. [1] propose an optimal step non-rigid ICP algorithm for surface registration, where an affine transformation is assigned to each vertex and a stiffness term is used to minimize the difference in the transformation of neighboring vertices. Sumner et al. [20] compute the set of affine transformations induced by the deformation of the source mesh, and the smoothness term is added to constrain the transformation matrices for adjacent triangles to be equal. Yeh et al. [28] introduce a template fitting method for 3D surface meshes and optimize the local transformation by applying bi-Laplacian energy. Yoshiyasu et al. [29] present a non-isometric surface registration method that restricts deformations locally as conformal as possible. However, the discretized energy they adopted is not a consistent discretization of a continuous energy and thus may produce foldover and distortion. To tackle this issue, Jiang et al. [8] propose a consistent as-similar-as-possible surface registration method. They apply the L_2-norm regularization on both the transformation variation and position difference (L_2–L_2), which results in consistent discretization for surfaces and improves the quality of the surface deformation and registration. However, the approach tends to produce over-smoothed results given a noisy target. Yang et al. [27] propose a sparse non-rigid registration method with an L_1-norm regularized on the transformation variation and an L_2-norm regularized on the position difference (L_1-L_2). The L_1-norm-regularized smoothness term usually results in artificial discontinuities, and the L_2-norm-regularized data term often loses fine details while smoothing the noise [5]. To tackle these issues, we regularize the smoothness term in Huber-norm, which significantly reduces the artificial discontinuities. Unlike most work using the squared L_2-norm in position difference constraints, we apply an L_1-norm-regularized data term, which is not only able to remove noise but also preserve fine details.

3 Surface registration

In this paper, we adopt a coarse-to-fine fitting strategy to implement the whole registration process in three steps: coarse fitting, mid-scale fitting and fine fitting. For each step, different energy terms are used and combined, which will be introduced at their first appearance. The comparison results with and without some energy terms will be illustrated to stress the significance of these energies.

3.1 Notations

Suppose the template mesh is composed of n vertices $\mathcal{P} \triangleq \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, where $\mathbf{p}_i \triangleq [x_i, y_i, z_i]^\top$ is a 3D vertex position in Euclidean coordinate. In coarse fitting step, n is the vertex number of coarse template mesh. The vertices of the target are denoted as $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$. For non-rigid registration, a 3×4 affine transformation matrix $\mathbf{A}_i \triangleq [\mathbf{X}_i, \mathbf{t}_i]$ is associated with each vertex $\mathbf{p}_i$ of the template, where $\mathbf{X}_i$ is a 3×3 linear transformation matrix and $\mathbf{t}_i$ is a 3×1 translation vector. For simplification, we concatenate $\mathbf{p}_i, \mathbf{q}_i, \mathbf{X}_i, \mathbf{t}_i$ into a $n \times 3$ matrix $\mathbf{P} \triangleq [\mathbf{p}_1 \ldots \mathbf{p}_n]^\top$, a $m \times 3$ matrix $\mathbf{Q} \triangleq [\mathbf{q}_1 \ldots \mathbf{q}_m]^\top$, a $3n \times 3$ matrix $\mathbf{X} \triangleq [\mathbf{X}_1 \ldots \mathbf{X}_n]^\top$ and a $n \times 3$ matrix $\mathbf{T} \triangleq [\mathbf{t}_1 \ldots \mathbf{t}_n]^\top$, respectively. Similarly, the vertices on the template dual mesh $\mathcal{P}^*$ are denoted by $\mathbf{P}^* \triangleq [\mathbf{p}_1^* \ldots \mathbf{p}_{n^*}^*]^\top$, where n^* is the number of the vertices on the template dual mesh, which is also equal to the number of triangle faces on the template primal mesh. Again, a translation vector will be assigned to each dual vertex, all of which can be concatenated as $\mathbf{T}^* \triangleq [\mathbf{t}_1^* \ldots \mathbf{t}_{n^*}^*]^\top$.

3.2 Coarse fitting

Instead of fitting the fine template surface from the beginning, a coarse mesh extracted from the origin template mesh is used to fit for efficiency, as the coarse mesh involves less unknown parameters. We employ the farthest point sampling approach [14] to sample certain number of vertices to approximately represent the shape of the template (Fig. 1a). Note that all the sampled vertices are the subset of the original vertex set. The geodesic remeshing technique [22] is then applied to generate the coarse mesh out of the samples vertices (Fig. 1b). Afterward, the feature points between the coarse template mesh and the target are specified by users (Fig. 1d). The coarse template is then fitted to the specified feature points to approximate the overall size of the target (Fig. 1e). The total energy in coarse fitting step is composed of four energies: E_f penalizes the distances between the feature points of template and target surface; E_ASAP constrains deformation ASAP; E_r penalizes the transformation variation; E_l penalizes the edge lengths difference locally:

$$E_\mathrm{coarse}(\mathbf{X}, \mathbf{T}) = w_\mathrm{f} E_\mathrm{f} + w_\mathrm{ASAP} E_\mathrm{ASAP} + w_\mathrm{r} E_\mathrm{r} + w_\mathrm{l} E_\mathrm{l}. \quad (1)$$

Each energy term will be introduced as follows, and w_f, w_ASAP, w_r and w_l are the weights which represent the influence of each energy term.

3.2.1 Feature point constraints

To pull feature points on the template toward their correspondence on the target, we define the feature point constraint energy as:

$$E_\mathrm{f}(\mathbf{T}) = \frac{1}{2} \sum_{i \in \mathcal{F}} \|\mathbf{p}_i + \mathbf{t}_i - \mathbf{q}_{\mathrm{idx}(i)}\|_F^2,$$

where $\mathcal{F}$ is the template index set of the feature points, $\mathbf{p}_i$ is the position of ith feature point on the template, $\mathrm{idx}(i)$ is the index of the corresponding feature point on the target. To fit this energy (and other energies subsequently) into ADMM [4] for optimization, we need to rewrite the energy (and other energies subsequently introduced) in matrix form. We define two sparse matrices $\mathbf{C}_\mathrm{f}$, $\mathbf{D}_\mathrm{f}$ which select the feature point pairs between the template and the target. Assuming the rth feature point pair is $\mathbf{p}_i$ on the template and $\mathbf{q}_{\mathrm{idx}(i)}$ on the target, then

$$\mathbf{C}_\mathrm{f}(r, s) = \begin{cases} 1, & \text{if} \quad s = i \\ 0, & \text{if} \quad s \neq i \end{cases}, \quad \mathbf{D}_\mathrm{f}(r, t) = \begin{cases} 1, & \text{if} \quad t = \mathrm{idx}(i) \\ 0, & \text{if} \quad t \neq \mathrm{idx}(i) \end{cases}.$$

Therefore, we can rewrite $\mathrm{E_f}$ as

$$E_\mathrm{f}(\mathbf{T}) = \frac{1}{2} \|\mathbf{C}_\mathrm{f}(\mathbf{P} + \mathbf{T}) - \mathbf{D}_\mathrm{f}\mathbf{Q}\|_F^2, \quad (2)$$

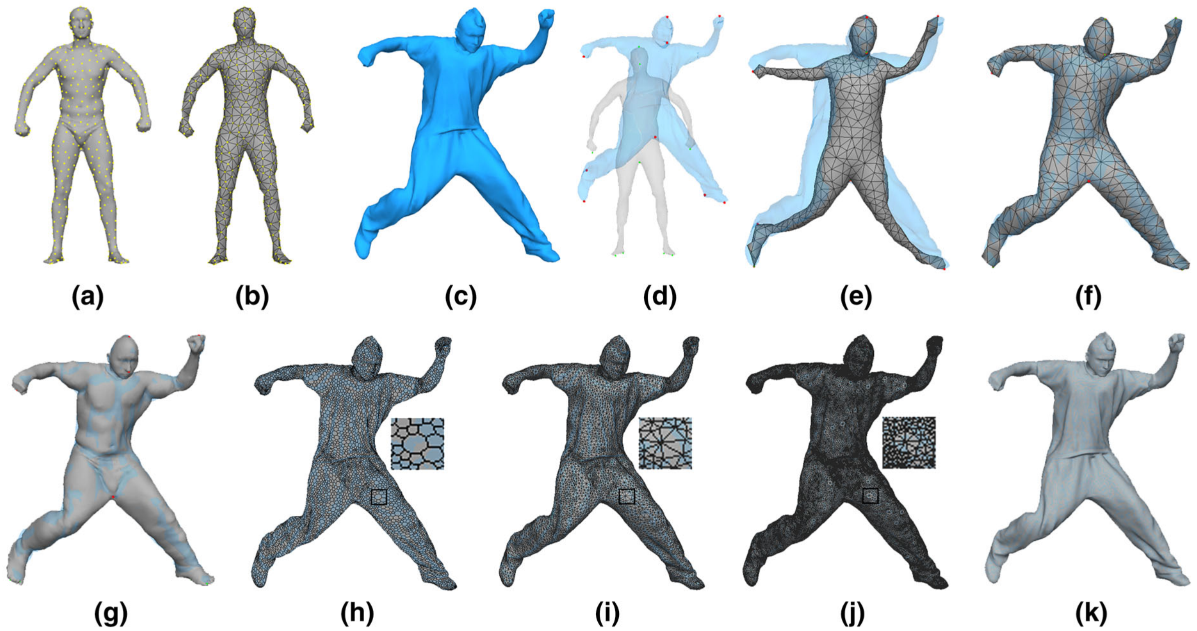


Fig. 1 Huber-L_1 algorithm: **a** sampled points (yellow dots); **b** remeshing from the sampled points as embedded coarse mesh; **c** input of target surface; **d** 9 feature points specified by user (red dots for target and green dots for template); **e** coarse fitting; **f** mid-scale fitting; **g** reconstructed through embedded deformation; **h** fine fitting on the dual domain; **i** fine fitting on the primal domain; **j** subdivision; **k** the final result

3.2.2 ASAP energy

The transformation from the template to the target can involve large deformation. To prevent shear distortion (Fig. 2), we constrain the deformation matrix $\mathbf{X}_i$ to an orthogonal rotation matrix $\mathbf{R}_i$. As we deal with non-isometric registration, the size between the template and the target can be different. Therefore, we add a scalar s_i to make the deformation scalable. The ASAP energy is defined as

$$E_{\text{ASAP}}(\mathbf{X}) = \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2,$$
$$\text{s.t. } \mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{I}_3$ is a 3×3 identity matrix and $\det(\cdot)$ is the determinant of a matrix. To write the energy in matrix form, we concatenate s_i, $\mathbf{R}_i$ into a $n \times 1$ vector $\mathbf{s} = [s_1 \ldots s_n]^\top$ and a $3n \times 3$ matrix $\mathbf{R} \triangleq [\mathbf{R}_1 \ldots \mathbf{R}_n]^\top$. The ASAP energy can then be expressed as:

$$E_{\text{ASAP}}(\mathbf{X}) = \frac{1}{2}\|\mathbf{X} - (\text{diag}(\mathbf{s}) \otimes \mathbf{I}_3)\mathbf{R}\|_F^2,$$
$$\text{s.t. } \mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \quad (3)$$

where $\text{diag}(\cdot)$ returns the block-wise diagonal matrix of each row vector in the input matrix and $\otimes$ denotes the operator of Kronecker product.

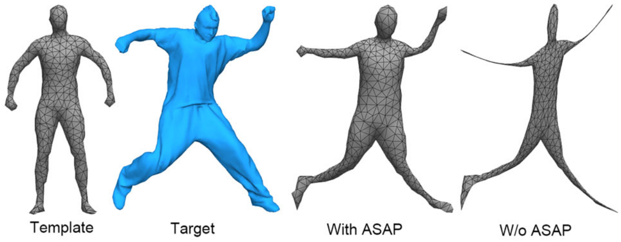


Fig. 2 Coarse fitting results with and without ASAP energy. Without ASAP energy, the model can easily get shear distortion

3.2.3 Regularization

We assign an affine transformation to each vertex. These transformations are not independent with each other, and the nearby transformations should have overlapping influence. Therefore, the computed transformations should be consistent with respect to one another. In practice, articulated animals only have large transformation deviation at joints, so the transformation on the parts between joints should be piecewise smooth with only small deviations. To produce a piecewise smooth result (Fig. 3), we apply a Huber-norm-regularized model constrained on the transformation variation. This metric behaves like an L_2-norm below a certain threshold ε and like an L_1-norm above. The regularization energy is defined as:

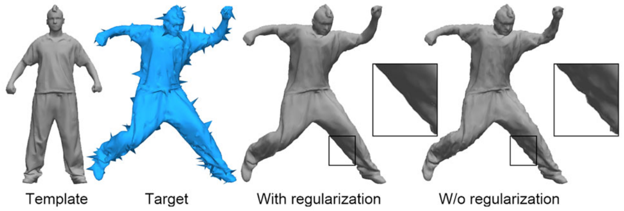


Fig. 3 Fitting results with and without regularization energy. The method with regularization energy is more robust against outliers and produces piecewise smoother result

$$E_{\mathrm{r}}(\mathbf{X},\mathbf{T}) = \sum_{\mathbf{e}_{ij}\in\mathcal{E}} \|\mathbf{X}_i(\mathbf{p}_j - \mathbf{p}_i) + \mathbf{p}_i + \mathbf{t}_i - (\mathbf{p}_j + \mathbf{t}_j)\|_{\varepsilon},$$

where $\mathbf{e}_{ij}$ is a directional edge from vertex i to vertex j, $\mathcal{E}$ is the half-edge set of the template mesh. Note that an edge has two opposite directional half-edges. $\|\cdot\|_{\varepsilon}$ is the Huber-norm defined as:

$$\|x\|_{\varepsilon} = \begin{cases} \frac{\|x\|_2^2}{2\varepsilon}, & \text{if} \quad \|x\|_1 \le \varepsilon \\ \|x\|_1 - \frac{\varepsilon}{2}, & \text{otherwise} \end{cases},$$

$\varepsilon > 0$ is a threshold, $\|\cdot\|_1, \|\cdot\|_2$ are the L_1, L_2 norm, respectively. In order to express the regularization energy with respective to $\mathbf{X}$ and $\mathbf{T}$, we introduce a selection matrix $\mathbf{J} \in \{1\}^{|\mathcal{E}|\times n}$ and a directional differential matrix $\mathbf{K} \in \{-1, 1\}^{|\mathcal{E}|\times n}$. Specifically, each row of $\mathbf{J}$ and $\mathbf{K}$ corresponds to a half-edge in $\mathcal{E}$ and each column of them corresponds to a vertex in $\mathcal{P}$. Without loss of generality, we assume the r-th rows in $\mathbf{J}$ and $\mathbf{K}$ are associated with the half-edge $\mathbf{e}_{ij}$. Each row in $\mathbf{J}$ denoted by $\mathbf{J}(r,\cdot)$ only has one nonzero entry, whose column corresponds to the start vertex $\mathbf{p}_i$ of the half edge $\mathbf{e}_{ij}$, i.e., $\mathbf{J}(r,i) = 1$. Each row in $\mathbf{K}$ denoted by $\mathbf{K}(r,\cdot)$ contains two nonzero entries. The entry linked to the reference vertex $\mathbf{p}_i$ is set at -1, while the one linked to the neighboring vertex $\mathbf{p}_j$ is set at 1, i.e., $\mathbf{K}(r,i) = -1$, $\mathbf{K}(r,j) = 1$. Then, the regularization term can be rewritten as:

$$E_{\mathrm{r}}(\mathbf{X},\mathbf{T}) = \|\mathrm{diag}(\mathbf{KP})(\mathbf{J}\otimes\mathbf{I}_3)\mathbf{X} - \mathbf{KP} - \mathbf{KT})\|_{\varepsilon}. \quad (4)$$

3.2.4 Laplacian energy

To improve the mesh quality, we apply the uniform Laplacian operator on the vertex position, enforcing each vertex to strive to lie in the centroid of its one-ring neighbors and thus the edge lengths strive to be locally equalized. The Laplacian energy is defined as follows:

$$E_{\mathrm{l}}(\mathbf{T}) = \frac{1}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2, \quad (5)$$

where $\mathbf{L}$ is the uniform Laplacian matrix corresponding to the mesh connectivity:

$$\mathbf{L}(i,j) = \begin{cases} -\frac{1}{d_i}, & \text{if} \quad \text{vertices } i \text{ and } j \text{ are neighbors,} \\ 1, & \text{if} \quad i = j, \\ 0, & \text{otherwise,} \end{cases}$$

where d_i is the valence of vertex i.

3.3 Mid-scale fitting

This fitting step deforms the coarse mesh as close as possible to the target. Apart from those constraints adopted in the previous step, the data constraint is also applied to attract the coarse mesh toward the target gradually (Fig. 1f). The energy in this step is denoted by:

$$E_{\mathrm{mid}}(\mathbf{X},\mathbf{T}) = E_{\mathrm{coarse}}(\mathbf{X},\mathbf{T}) + w_{\mathrm{d}}E_{\mathrm{d}}. \quad (6)$$

Data constraint

To pull the template toward the target, we need to determine the reliable correspondences between the template and the target. For each vertex $\mathbf{p}_i$ on the template, we project it onto the target along its normal direction. The projection $\mathbf{c}_i$ is regarded as a reliable correspondence only if:

- $\mathbf{c}_i$ is inside a triangle of the target.
- The distance between $\mathbf{c}_i$ and $\mathbf{p}_i$ is under a threshold α.
- The angle between the normals at $\mathbf{c}_i$ and $\mathbf{p}_i$ is under a threshold Θ.

We apply L_1 norm on the position difference to allow a small fraction of regions with large positional error, which is more robust against outliers and fits to the target's details better. The data term is defined as:

$$E_{\mathrm{d}}(\mathbf{T}) = \sum_{i\in\mathcal{C}} \|\mathbf{p}_i + \mathbf{t}_i - \mathbf{c}_i\|_1,$$

where $\mathcal{C}$ is the template index set of the correspondence. In order to express the data term in regard to $\mathbf{T}$, we stack all $\mathbf{c}_i$ into a matrix $\mathbf{C}$. Similar to $\mathbf{C}_{\mathrm{f}}$, we define a sparse matrix $\mathbf{C}_{\mathrm{d}}$ to indicate the corresponding vertices on the template. Therefore, we can rewrite the data term as:

$$E_{\mathrm{d}}(\mathbf{T}) = \|\mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C})\|_1. \quad (7)$$

3.4 Fine fitting

In this step, the dense mesh is first reconstructed from the mid-scale fitting result via embedded deformation method [21] (Fig. 1g). To avoid foldovers and improve the mesh quality, the dual-domain relaxation strategy [28] is applied. We alternatively perform the relaxation algorithm between the primal domain and the dual domain. The reconstructed primal mesh is first transformed into its corresponding dual mesh, and then, the relaxation algorithm is applied on the dual domain (Fig. 1h). The energy in the dual domain is defined as:

$$E_{\mathrm{dual}}(\mathbf{T}^*) = w_{\mathrm{r}}E_{\mathrm{r}}(\mathbf{T}^*) + w_{\mathrm{d}}E_{\mathrm{d}}(\mathbf{T}^*) + w_{\mathrm{l}}E_{\mathrm{l}}(\mathbf{T}^*). \tag{8}$$

After that, the relaxation algorithm is applied in the primary domain (Fig. 1i), whose energy can be expressed as:

$$E_{\mathrm{primal}}(\mathbf{T}) = w_{\mathrm{c}}E_{\mathrm{c}}(\mathbf{T}) + w_{\mathrm{l}}E_{\mathrm{l}}(\mathbf{T}), \tag{9}$$

where E_{c} is the consistent energy transforming the template from the dual domain to the primary domain.

Consistency constraint

It is well known that one dual mesh corresponds to a primal mesh and their vertices should be consistent: the ith dual vertex should be equal to the centroid vertex of the ith triangle of the primal mesh (Fig. 4). The consistency energy is defined as:

$$E_{\mathrm{c}}(\mathbf{T}) = \frac{1}{2}\sum_{i=1}^{n^*} \| \frac{1}{3}(\mathbf{p}_{i_1} + \mathbf{t}_{i_1} + \mathbf{p}_{i_2} + \mathbf{t}_{i_2} + \mathbf{p}_{i_3} + \mathbf{t}_{i_3}) - \mathbf{p}_i^* \|^2,$$

where $\{i_1, i_2, i_3\}$ are the indices of the primal vertices participating in the ith triangle of the primal mesh. Again, in matrix form, we could rewrite this energy as:

$$E_{\mathrm{c}}(\mathbf{T}) = \frac{1}{2}\|\mathbf{C}_{\mathrm{c}}(\mathbf{P} + \mathbf{T}) - \mathbf{P}^*\|_F^2, \tag{10}$$

where $\mathbf{C}_{\mathrm{c}}$ is a $n^* \times n$ matrix:

$$\mathbf{C}_{\mathrm{c}}(i, j) = \begin{cases} \frac{1}{3}, & \text{if} \quad j \in \{i_1, i_2, i_3\}, \\ 0, & \text{otherwise}, \end{cases}$$

If the resolution of the template mesh is insufficient to fit the target tightly, a uniform or adaptive subdivision approach can be employed. Here, we adopt 1–4 uniform subdivision method [9] to subdivide the template (Fig. 1j), and then, the dual-domain relaxation algorithm is performed on the subdivided template mesh again.

4 Optimization

The optimization process for each fitting step is quite similar. We take the mid-scale step as an example to explain the optimization process as it includes all kinds of norm appeared in the algorithm. Expanding each term in (6) gives us:

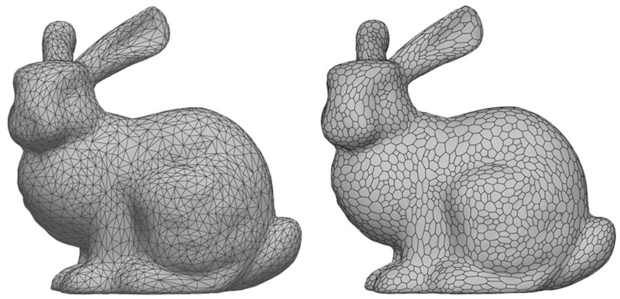


Fig. 4 The Stanford bunny's primary mesh and its corresponding dual mesh

$$\begin{aligned} E_{\mathrm{mid}}(\mathbf{X}, \mathbf{T}) = \; & \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P} + \mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 \\ & + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 \\ & + w_{\mathrm{r}}\|\mathrm{diag}(\mathbf{KP})(\mathbf{J} \otimes \mathbf{I}_3)\mathbf{X} - \mathbf{KP} - \mathbf{KT}\|_{\varepsilon} \\ & + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P} + \mathbf{T})\|_F^2 + w_{\mathrm{d}}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C}\|_1, \\ \text{s.t. } & \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \end{aligned} \tag{11}$$

To solve the non-differential Huber-norm and L_1-norm, we introduce two auxiliary variables $\mathbf{F}, \mathbf{G}$ to change the minimization of (11) into the following form:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{T},\mathbf{F},\mathbf{G}} \; & \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P}+\mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 \\ & + w_{\mathrm{r}}\|\mathbf{F}\|_{\varepsilon} + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P} + \mathbf{T})\|_F^2 + w_{\mathrm{d}}\|\mathbf{G}\|_1, \\ \text{s.t. } & \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \\ & \mathbf{F} = \mathbf{HX} - \mathbf{KP} - \mathbf{KT}, \mathbf{G} = \mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C}, \end{aligned} \tag{12}$$

where $\mathbf{H} = \mathrm{diag}(\mathbf{KP})(\mathbf{J} \otimes \mathbf{I}_3)$ is introduced for conciseness.

To solve the constrained minimization (12), we transform the original problem to iterative minimization of its augmented Lagrangian form:

$$\begin{aligned} & L(\mathbf{X}, \mathbf{T}, \mathbf{F}, \mathbf{G}, \{\mathbf{R}_i\}, \{s_i\}, \mathbf{Y}, \rho) \\ & \quad = \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P} + \mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{q}\|_F^2 \\ & \quad + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 \end{aligned}$$

$$
\begin{aligned}
&+ w_{\mathrm{r}}(\|\mathbf{F}\|_{\varepsilon} + \langle \mathbf{Y}_1, \mathbf{HX} - \mathbf{KP} - \mathbf{KT} - \mathbf{F}\rangle_F \\
&+ \frac{\rho_1}{2}\|\mathbf{HX} - \mathbf{KP} - \mathbf{KT} - \mathbf{F}\|_F^2) \\
&+ \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2 + w_{\mathrm{d}}(\|\mathbf{G}\|_1 \\
&+ \langle \mathbf{Y}_2, \mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C} - \mathbf{G}\rangle_F \\
&+ \frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C} - \mathbf{G}\|_F^2), \\
\text{s.t. } & \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \qquad (13)
\end{aligned}
$$

where ρ_1 and ρ_2 are positive constants, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are Lagrangian multipliers and $\langle\cdot,\cdot\rangle_F$ is the Forbenius inner product. We solve this problem by using the ADMM algorithm, the detailed derivation of which is referred to "Appendix A."

Now we could summarize the optimization in mid-scale fitting step in Algorithm 1. The optimization consists of two loops: the outer loop adjusts the weights and searches the correspondences to construct $\mathbf{E}_{\mathrm{d}}$, while the inner loop optimizes the translation $\mathbf{T}$ for each template vertex. Once the inner loop converges, $\mathbf{T}$ will be used to update the template vertex position, and then, a new outer iteration starts again. In this algorithm, l and k represent the indices of the outer and inner iteration, respectively. The optimization problems in coarse fitting and fine fitting steps can be solved in the same way.

Algorithm 1 Mid-scale fitting step

```
while not converged do
  Adjust weights in (11) and construct $\mathbf{E}_{\mathrm{d}}$
  while not converged do
    Compute $\mathbf{F}^{(k)}$ by solving Eq. (15).
    Compute $\mathbf{G}^{(k)}$ by solving Eq. (16).
    Compute $\mathbf{R}_i^{(k)}$ by solving Eq. (17).
    Compute $s_i^{(k)}$ by solving Eq. (18).
    Compute $\mathbf{X}^{(k)}$ by solving Eq. (19).
    Compute $\mathbf{T}^{(k)}$ by solving Eq. (20).
    $\mathbf{Y}_1^{(k)}=\mathbf{Y}_1^{(k-1)} + \rho_1(\mathbf{HX}^{(k)} - \mathbf{KP} - \mathbf{KT}^{(k)}- \mathbf{F}^{(k)})$.
    $\mathbf{Y}_2^{(k)} = \mathbf{Y}_2^{(k-1)} + \rho_2(\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k)})–\mathbf{C}–\mathbf{G}^{(k)})$.
  end while
  Update the template by $\mathbf{P}^{(l)} = \mathbf{P}^{(l-1)} + \mathbf{T}^{(l-1)}$.
end while
```

5 Experiments

We evaluate the performances of the proposed approach by comparing with state-of-the-art algorithms on clean datasets, noisy datasets and real scans, respectively. The number of feature points, vertices and faces of models used in each example is shown in Table 1. All the algorithms are implemented in MATLAB, and all the statistics are measured on an Intel Xeon E5 3.4 GHz 64-bit workstation with 16GB of RAM.

Table 1 The number of feature points (#FP), vertices (#V), faces (#F) of the template and the target models in the examples

Name	#FP	Template		Target	
		$\#V$	$\#F$	$\#V$	$\#F$
Bouncing	9	12,500	24,996	10,002	20,000
Camel	24	6608	13,200	9469	18,934
Crane	11	12,500	24,996	10,002	20,000
Dog	0	25,290	50,528	25,290	50,528
Gorilla	0	25,438	50,868	25,438	50,868
Head	10	1669	3298	281,581	562,554

5.1 Parameters and weights

In the data constraints, we set $\Theta = 90°$, $\alpha = 0.05r_{\mathrm{box}}$, where r_{box} is the length of the target bounding box diagonal. In the regularization term, $\epsilon = 0.1$ is used, $\rho = 1$ for the optimization. As for the weights, we use $w_{\mathrm{ASAP}} = 1$, $w_{\mathrm{r}} = 10$, $w_{\mathrm{f}} = 1000$, $w_{\mathrm{l}} = 10$ for the whole coarse fitting step to highlight the feature point constraints. In the mid-scale step, w_{d} is initialized to 0.1 and then increased by 0.1 for each outer iteration. w_{l} is initialized to 100 and then decreased by 10 for each outer iteration until reaching 1. We gradually increase w_{d} and reduce w_{l} so that the good quality of the template mesh can be maintained during the registration to avoid the foldover occurrence. The other weights are the same as those in the coarse fitting step. In the fine fitting step, we set $w_{\mathrm{d}} = 0.1$, $w_{\mathrm{r}} = 0.1$ for (8); $w_{\mathrm{c}} = 1000$ for (9); w_{l} is initialized to 100 and then decreased by 10 for each outer iteration until reaching 1 through the whole fine fitting step; w_{d} is increased by 0.1 for each outer iteration as the same reason in the mid-scale step.

5.2 Results on clean data

We compare our method with state-of-the-art non-isometric registration methods (Fig. 5): consistent as-similar-as-possible surface registration (CASAP) [8], as-conformal-as-possible surface registration (ACAP) [29], the shape matching-based registration method that minimizes the as-similar-as-possible energy (SM-ASAP) [16] and the registration method that utilizes the point-based deformation smoothness regularization (PDS) [1]. To evaluate the registration accuracy quantitatively, we follow the criterion used in [8,11] and measure (1) distance error, which is the average distance from the vertices of the deformed template to the corresponding points on the target relative to the target bounding box diagonal, (2) intersection error, which is the number of self-intersecting faces, (3) Hausdorff error, the largest distance between two shapes with respective to the target bounding box diagonal. The statistics data can be

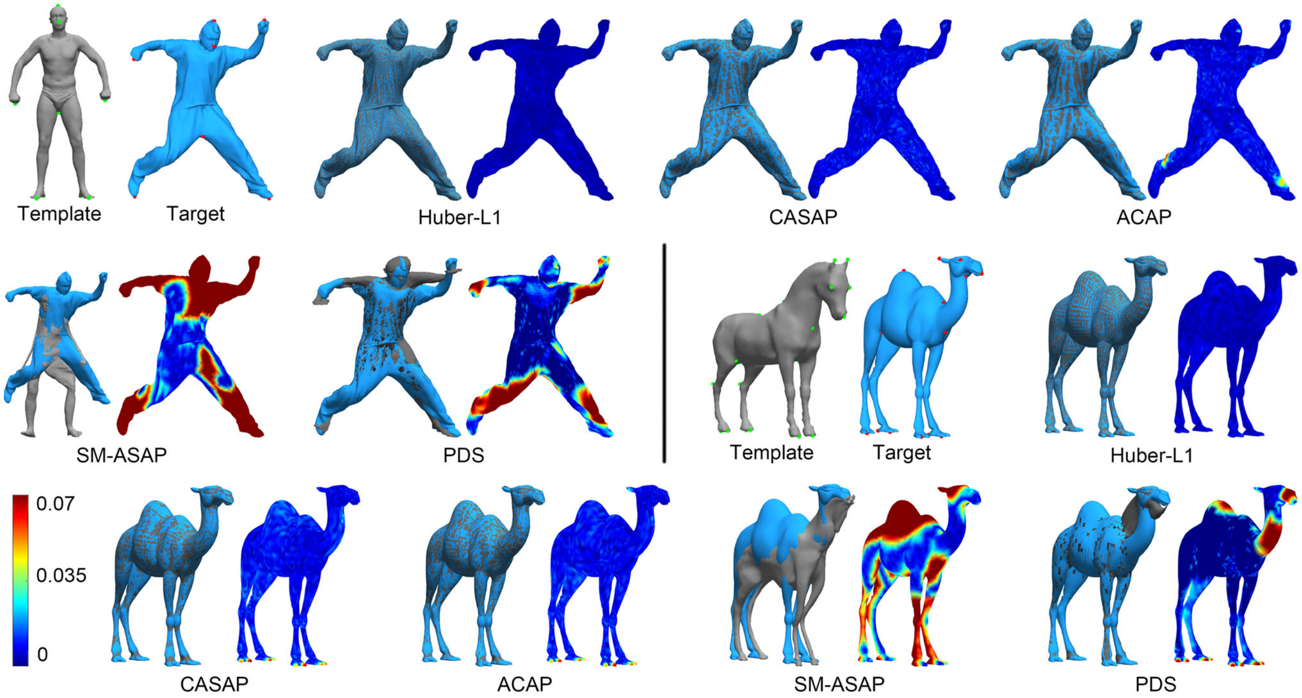


Fig. 5 Comparison of non-isometric surface registration approaches on clean data. The color bar denotes the distance from the registration result to the target

Table 2 Quantitative evaluation in the bouncing and camel examples. D, I and H indicate distance error [%], intersection error and Hausdorff error [%], respectively

	Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
Bouncing					
D	**2.1556e−05**	4.3619e−05	4.9460e−05	0.0010	1.8525e-04
I	**0**	242	494	909	5639
H	**0.0113**	0.0273	0.0784	0.7279	0.1619
Camel					
D	**7.1209e−05**	1.8185e−04	1.8074e−04	9.8233e−04	2.9304e−04
I	**0**	**0**	76	85	3884
H	**0.0172**	0.0704	0.0710	0.3222	0.1238

Best values are highlighted in bold

found in Table 2. It is obvious that our method Huber-L_1 produces least errors without any foldover generated. Although CASAP and ACAP can fit the template close to the target, as shown at the left ankle of the bouncing model, they still suffer from the foldover issue when the deformation is dramatic. SM-ASAP does not require feature points, but it is only able to handle surfaces with close initial alignment and similar poses. PDS allows for affine transformation for each vertex, which makes it too weak against shear distortions as shown at both arms of the bouncing model. Table 3 shows the iteration steps and time taken by each method. The total time cost in SM-ASAP and PDS is relatively small; however, their results are undesirable with large errors and many foldovers. The total time used in the rest of three methods is almost on the same level, but our method gets the best result with no foldover occurred.

5.3 Results on noisy data

In this subsection, we set up two different experiments to demonstrate the robustness of our method. In both experiments, the targets are polluted with noise along the normal direction of each vertex by multiplying the standard deviation of the average length of the edges in the target.

First, we compare our method with the start-of-the-arts on the noisy data in Fig. 6. The quantitative evaluation is shown in Table 4. Affected by the noise, CASAP and ACAP get poor initial shape estimation and regard some noise as correspondence, which makes parts of template fitted to noise as shown at the right waist of CASAP and the left arm of ACAP. SM-ASAP and PDS still produce poor results as on the clean data. Thanks to the dual relaxation and Huber-L_1 regularization, our method is robust against noise and achieves more

Table 3 Iteration steps and time (in seconds) in the bouncing and camel example. #O and #I indicate the number of outer iteration steps and total inner iteration steps, respectively. "Inner" indicates the average time required for each inner iteration step. "Total" represents the total fitting time

	Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
Bouncing					
#O	24	54	73	**9**	500
#I	2064	3477	577	**37**	1608
Inner	0.068	**0.034**	0.277	0.252	0.035
Total	140.199	118.044	159.687	**9.346**	56.281
Camel					
#O	24	54	73	**9**	500
#I	2037	3352	562	**34**	1582
Inner	0.065	**0.032**	0.272	0.261	0.034
Total	132.405	107.264	152.864	**8.874**	54.328

Best values are highlighted in bold

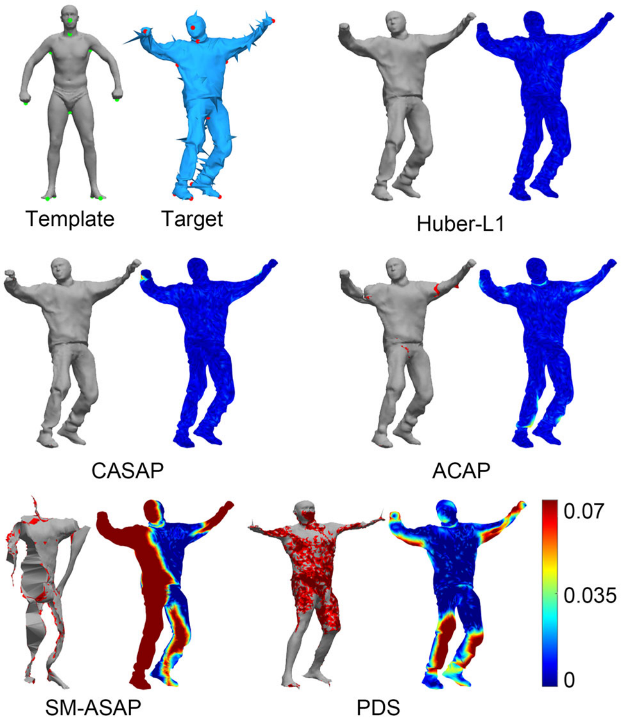


Fig. 6 Comparison of non-isometric surface registration approaches on noisy data. The self-intersection faces on the template are colored in red

accurate results than other methods without any foldover generated.

Second, instead of comparing with a particular method (e.g., CASAP), we compare our Huber-L_1 with different norms applied on the regularization term and data term, including TV-L_1, TV-L_2 and L_2–L_2, as shown in Fig. 7. To ensure fair comparison, the correspondences among different approaches are the same and have been given as priors. As the L_2-norm is easily affected by the outliers, TV-L_2 and L_2-L_2 produce larger errors than Huber-L_1 and TV-L_1, especially at the places with large deformation (seen at the shoulder and butt of the gorilla in SNR and L_2–L_2). Due to the L_1-norm's tendency on favoring sparse solution, the effect caused by the regularizer leads to piecewise constant solutions (shown at the tail of the dog in TV-L_1). This effect can be reduced significantly by using a quadratic penalization for small gradient magnitudes while sticking to linear penalization for larger magnitudes to maintain the discontinuity properties known from TV, which is the Huber-norm we adopted here. The quantitative evaluation is shown in Table 5. As the benchmark models also involve foldovers, we ignore the intersection error in the table. With Huber-L_1 regularization scheme, our method is not only robust against noise but also produces piecewise smooth results with least errors.

5.4 Results on real scans

Finally, we compare our method with the state-of-the-art approaches on real scan in Fig. 8. The quantitative evaluation is shown in Table 6. In terms of distance error and Hausdorff error, CASAP and ACAP have competitive results with our method. However, they still produce foldovers at the canthus and nostril and especially generate poor results around

Table 4 Quantitative evaluation in the crane example

	Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
Crane					
D	**4.7397e−05**	5.2738e−05	5.6536e−05	9.4177e−04	1.6583e−04
I	**0**	143	453	8965	5862
H	**0.0227**	0.0729	0.0469	0.5691	0.1258

Best values are highlighted in bold

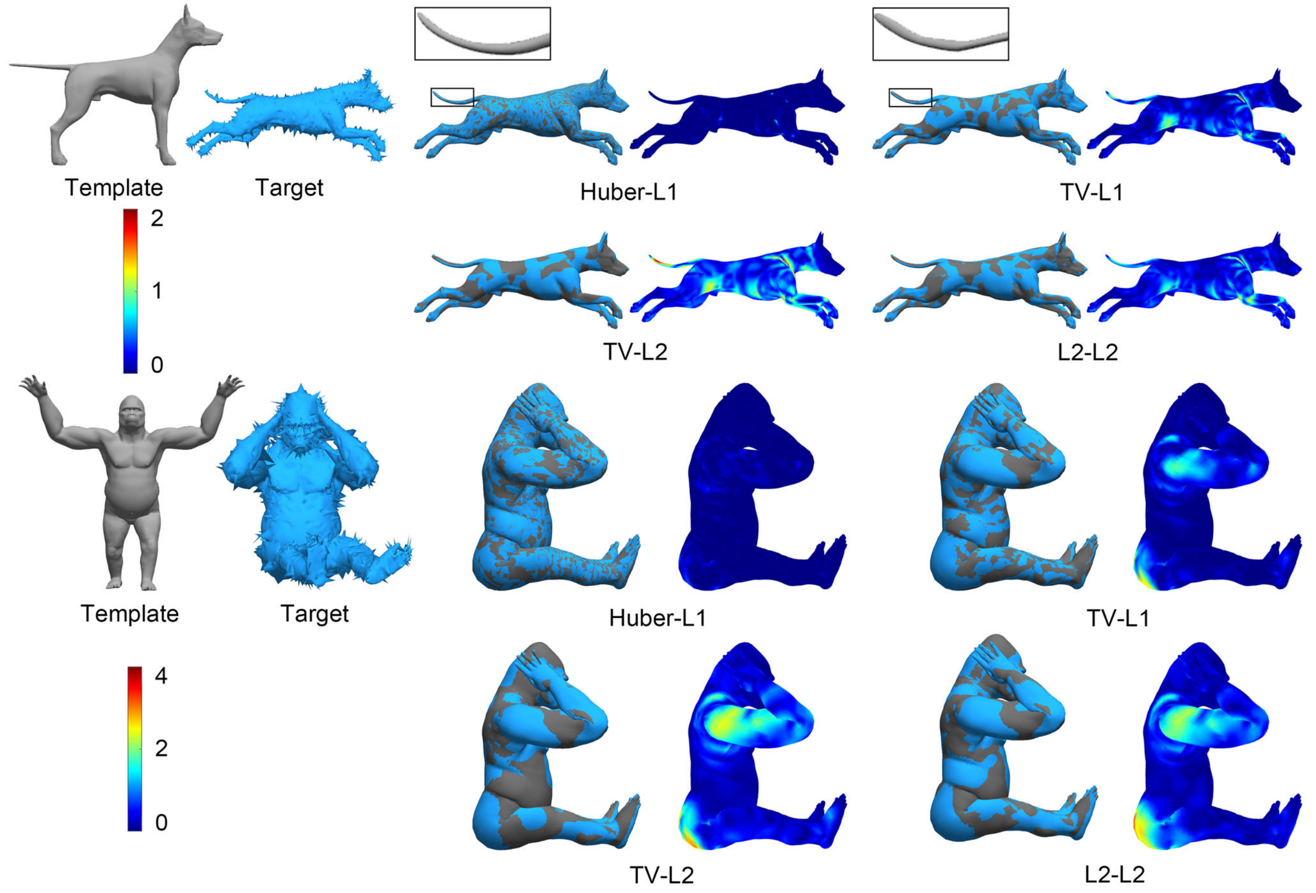


Fig. 7 Comparison of different norms applied on the regularization term and data term on noisy data

Table 5 Quantitative evaluation in the dog and gorilla examples

	Huber-L_1	TV-L_1	TV-L_2	L_2-L_2
Dog				
D	**1.0441e−06**	5.2783e−06	1.0682e−05	1.5101e−05
H	**1.6409**	2.9771	3.9666	2.6419
Gorilla				
D	**2.3474e−06**	2.8365e−06	1.6977e−05	2.3565e−05
H	**4.1795**	4.5067	5.5522	5.1793

Best values are highlighted in bold

the noisy boundary and the missing parts of the target (seen from the side view). Compared to CASAP and ACAP, SM-ASAP and PDS produce less foldovers; however, they have larger distance errors and Hausdorff errors so that their results even look dissimilar to the target intuitively. On the contrary, Huber-L_1 produces least errors with no foldover generated, which demonstrates our method is more robust to a noisy and incomplete target.

6 Conclusions

We have proposed a novel non-isometric surface registration approach based on the Huber-L_1 model. The Huber-norm regularizes on transformation variation, which is robust to noise and produces piecewise smooth result. The position difference is regularized in L_1-norm, which preserves the fine details on the target while smoothing the noise. The ASAP energy allows us to handle shapes in different sizes. To improve the efficiency and robustness of registration, a coarse-to-fine strategy is adopted. The Laplacian energy relaxes the template on the primal and dual domain, reducing self-intersection and foldover occurrence and improving the mesh quality. The experiments on various data have shown our method is more robust and accurate than other state-of-the-art approaches.

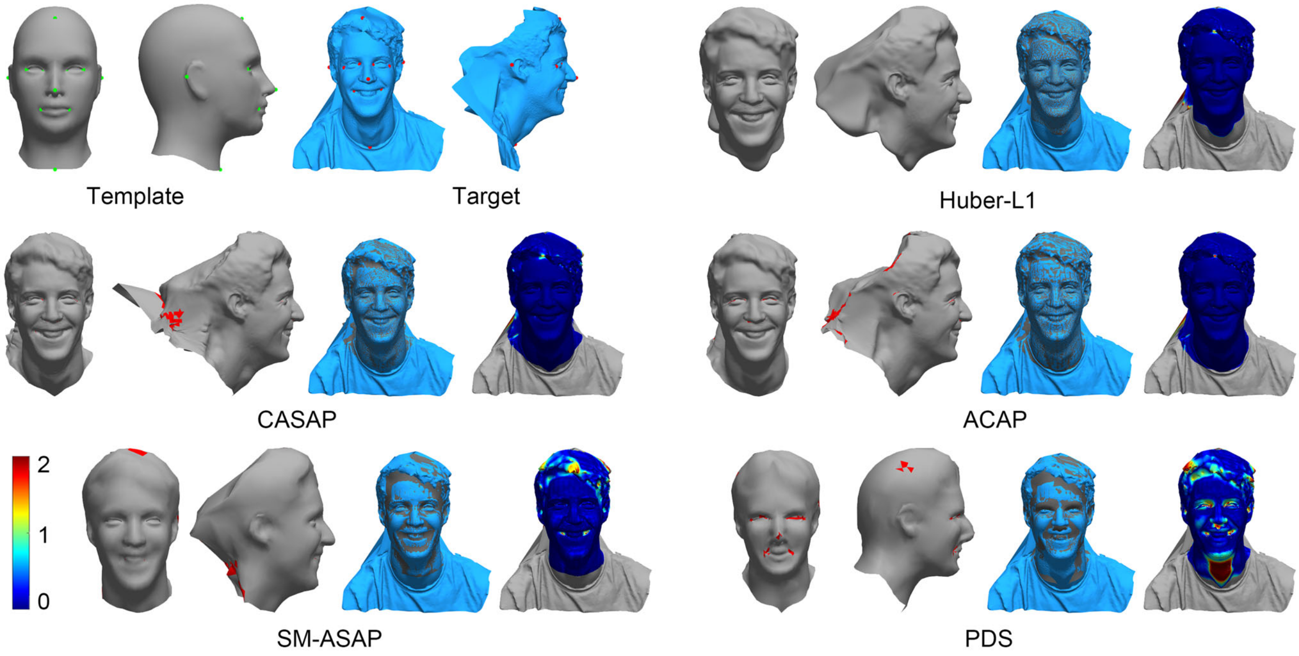


Fig. 8 Comparison of non-isometric surface registration approaches on real scan

Table 6 Quantitative evaluation in the head example

	Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
Head					
D	**3.6954e−06**	4.0682e−04	3.1503e−04	0.0016	0.0048
I	**0**	1287	2345	103	185
H	**3.1624**	3.2413	3.2244	3.2463	3.2516

Best values are highlighted in bold

In the future, we will detect the inversion of the elements to prevent the occurrence of foldover completely during registration. In addition, if the feature points can be detected automatically from deep learning, the whole registration process will be implemented without any user intervention. This would be another interesting research topic.

Acknowledgements We would like to thank Gabriel Peyre and Alec Jacobson for their help in rendering experimental results in MATLAB.

Funding This study was funded by EU H2020 under the REA Grant Agreement (Grant Number 691215)

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Appendix A

The ADMM algorithm is employed to optimize unknown variables in (13) alternatively. The kth iteration can be summarized as follows:

$$\left\{\begin{array}{rcl}
\mathbf{F}^{(k)} & = & \arg\min_{\mathbf{F}} w_{\mathrm{r}}(\|\mathbf{F}\|_{\varepsilon} + \langle \mathbf{Y}_1^{(k-1)}, \mathbf{HX}^{(k-1)} - \mathbf{KP} - \mathbf{KT}^{(k-1)} \\
 & & -\mathbf{F}\rangle_F + \frac{\rho_1}{2}\|\mathbf{HX}^{(k-1)} - \mathbf{KP} - \mathbf{KT}^{(k-1)} - \mathbf{F}\|_F^2), \\
\mathbf{G}^{(k)} & = & \arg\min_{\mathbf{G}} w_{\mathrm{d}}(\|\mathbf{G}\|_1 + \langle \mathbf{Y}_2^{(k-1)}, \mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k-1)}) \\
 & & -\mathbf{C} - \mathbf{G}\rangle_F + \frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k-1)}) - \mathbf{C} - \mathbf{G}\|_F^2), \\
\mathbf{R}_i^{(k)} & = & \arg\min_{\mathbf{R}_i} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X}_i^{(k-1)} - s_i^{(k-1)}\mathbf{R}_i\|_F^2 \\
 & & \text{s.t. } \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \\
s_i^{(k)} & = & \arg\min_{s_i} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X}_i^{(k-1)} - s_i\mathbf{R}_i^{(k-1)}\|_F^2, \\
\mathbf{X}^{(k)} & = & \arg\min_{\mathbf{X}} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X} - (\mathrm{diag}(\mathbf{s}^{(k-1)}) \otimes \mathbf{I}_3)\mathbf{R}^{(k-1)}\|_F^2 \\
 & & +w_{\mathrm{r}}(\langle \mathbf{Y}_1^{(k-1)}, \mathbf{HX} - \mathbf{KP} - \mathbf{KT}^{(k-1)} - \mathbf{F}^{(k)}\rangle_F \\
 & & +\frac{\rho_1}{2}\|\mathbf{HX} - \mathbf{KP} - \mathbf{KT}^{(k-1)} - \mathbf{F}^{(k)}\|_F^2), \\
\mathbf{T}^{(k)} & = & \arg\min_{\mathbf{T}} \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P} + \mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 \\
 & & +w_{\mathrm{r}}(\langle \mathbf{Y}_1^{(k-1)}, \mathbf{HX}^{(k)} - \mathbf{KP} - \mathbf{KT} - \mathbf{F}^{(k)}\rangle_F \\
 & & +\frac{\rho_1}{2}\|\mathbf{HX}^{(k)} - \mathbf{KP} - \mathbf{KT} - \mathbf{F}^{(k)}\|_F^2) + \frac{w_1}{2}\|\mathbf{L}(\mathbf{P} + \mathbf{T})\|_F^2 \\
 & & +w_{\mathrm{d}}(\langle \mathbf{Y}_2^{(k-1)}, \mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C} - \mathbf{G}^{(k)}\rangle_F \\
 & & +\frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C} - \mathbf{G}^{(k)}\|_F^2) \\
\mathbf{Y}_1^{(k)} & = & \mathbf{Y}_1^{(k-1)} + \rho_1(\mathbf{HX}^{(k)} - \mathbf{KP} - \mathbf{KT}^{(k)} - \mathbf{F}^{(k)}), \\
\mathbf{Y}_2^{(k)} & = & \mathbf{Y}_2^{(k-1)} + \rho_2(\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k)}) - \mathbf{C} - \mathbf{G}^{(k)}).
\end{array}\right. \tag{14}$$

The $\mathbf{F}$-subproblem has the following closed solution:

$$\mathbf{F}^{(k)} = \frac{\varepsilon\rho_1}{\varepsilon\rho_1+1}\left(\mathbf{HX}^{(k-1)} - \mathbf{KP} - \mathbf{KT}^{(k-1)} + \frac{\mathbf{Y}_1^{(k-1)}}{\rho_1}\right) + \frac{1}{\varepsilon\rho_1+1} S_{\frac{\varepsilon\rho_1+1}{\rho_1}}\left(\mathbf{HX}^{(k-1)} - \mathbf{KP} - \mathbf{KT}^{(k-1)} + \frac{\mathbf{Y}_1^{(k-1)}}{\rho_1}\right), \tag{15}$$

where S is the soft thresholding operator acting on each element of the given matrix:

$$S_\kappa(a) = \begin{cases} a-\kappa, & \text{if} \quad a > \kappa \\ 0, & \text{if} \quad |a| \le \kappa \\ a+\kappa, & \text{if} \quad a < -\kappa \end{cases}.$$

The $\mathbf{G}$-subproblem can be solved as:

$$\mathbf{G}^{(k)} = S_{\frac{1}{\rho_2}}\left(\mathbf{C}_\mathrm{d}(\mathbf{P} + \mathbf{T}^{(k-1)}) - \mathbf{C} + \frac{\mathbf{Y}_2^{(k-1)}}{\rho_2}\right), \tag{16}$$

Following the work of [19], we solve the $\mathbf{R}_i$-subproblem by using singular value decomposition of $\mathbf{X}_i$:

$$\mathbf{U}_i\Sigma_i\mathbf{V}_i^T = \mathrm{svd}(\mathbf{X}_i), \mathbf{R}_i^{(k)} = \mathbf{V}_i\mathbf{U}_i^T \tag{17}$$

If $\det(\mathbf{R}_i) < 0$, we change the sign of the column of $\mathbf{U}_i$ corresponding to the smallest singular value.

Dividing the s_i-subproblem by s_i and setting its derivative to zero yields:

$$s_i^{(k)} = \frac{\langle \mathbf{X}_i^{(k-1)}, \mathbf{X}_i^{(k-1)} \rangle_F}{\langle \mathbf{R}_i^{(k)}, \mathbf{R}_i^{(k)} \rangle_F} = \frac{\langle \mathbf{X}_i^{(k-1)}, \mathbf{X}_i^{(k-1)} \rangle_F}{3} \tag{18}$$

In order to solve the $\mathbf{X}$-subproblem, we first zero its derivative and then write the equation in its equally stacked form:

$$\mathbf{A}_x\mathbf{X}^{(k)} = \mathbf{B}_x, \tag{19}$$

where

$$\mathbf{A}_x = \begin{pmatrix} \mathrm{sqrt}(w_{\mathrm{ASAP}})\mathbf{I} \\ \mathrm{sqrt}(w_\mathrm{r})\mathbf{H} \end{pmatrix},$$

$$\mathbf{B}_x = \begin{pmatrix} \mathrm{sqrt}(w_{\mathrm{ASAP}})(\mathrm{diag}(\mathbf{s}^{(k)}) \otimes \mathbf{I}_3)\mathbf{R}^{(k)} \\ \mathrm{sqrt}(w_\mathrm{r})(\mathbf{F}^{(k)} - \frac{\mathbf{Y}^{(k-1)}}{\rho_1} + \mathbf{KP} + \mathbf{KT}^{(k-1)}) \end{pmatrix},$$

$\mathrm{sqrt}(\cdot)$ is the square root function.

Similarly, $\mathbf{T}$ can be obtained by solving the following equation:

$$\mathbf{A}_t\mathbf{T}^{(k)} = \mathbf{B}_t, \tag{20}$$

where

$$\mathbf{A}_t = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{f})\mathbf{C}_\mathrm{f} \\ \mathrm{sqrt}(w_\mathrm{r})\mathbf{K} \\ \mathrm{sqrt}(w_\mathrm{d})\mathbf{C}_\mathrm{d} \\ \mathrm{sqrt}(w_\mathrm{l})\mathbf{L} \end{pmatrix},$$

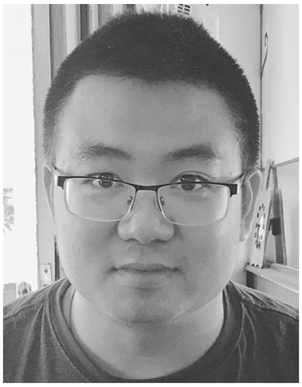
$$\mathbf{B}_t = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{f})(\mathbf{D}_\mathrm{f}\mathbf{Q} - \mathbf{C}_\mathrm{f}\mathbf{P}) \\ \mathrm{sqrt}(w_\mathrm{r})(\mathbf{HX}^{(k)} - \mathbf{KP} - \mathbf{F}^{(k)} + \frac{\mathbf{Y}^{(k-1)}}{\rho_1}) \\ \mathrm{sqrt}(w_\mathrm{d})(\mathbf{G}^{(k)} - \frac{\mathbf{Y}_2^{(k-1)}}{\rho_2} + \mathbf{C} - \mathbf{C}_\mathrm{d}\mathbf{P}) \\ \mathrm{sqrt}(w_\mathrm{l})(-\mathbf{L} * \mathbf{P}) \end{pmatrix}.$$

References

1. Amberg, B., Romdhani, S., Vetter, T.: Optimal step nonrigid ICP algorithms for surface registration. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR'07, pp. 1–8. IEEE (2007)
2. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
3. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. In: Computer graphics forum, vol. **32**, pp. 113–123. Wiley (2013)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
5. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imag. Vis. **40**(1), 120–145 (2011)
6. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. In: Dodgson, N., Floater, M.S., Sabin, M. (eds.) Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer, Berlin (2005)
7. Huang, Q.X., Adams, B., Wicke, M., Guibas, L.J.: Non-rigid registration under isometric deformations. In: Computer Graphics Forum, vol. 27, pp. 1449–1457. Wiley (2008)
8. Jiang, T., Qian, K., Liu, S., Wang, J., Yang, X., Zhang, J.: Consistent as-similar-as-possible non-isometric surface registration. Vis. Comput. **33**(6–8), 891–901 (2017)
9. Lee, A., Moreton, H., Hoppe, H.: Displaced subdivision surfaces. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 85–94. ACM Press/Addison-Wesley Publishing Co. (2000)
10. Li, H., Sumner, R.W., Pauly, M.: Global correspondence optimization for non-rigid registration of depth scans. In: Computer Graphics Forum, vol. 27, pp. 1421–1430. Wiley (2008)
11. Li, K., Yang, J., Lai, Y., Guo, D.: Robust non-rigid registration with reweighted position and transformation sparsity. IEEE Trans. Vis. Comput. Graphics **25**(6), 2255–2269 (2019)
12. Liao, M., Zhang, Q., Wang, H., Yang, R., Gong, M.: Modeling deformable objects from a single depth camera. In: IEEE 12th International Conference on Computer Vision, 2009, pp. 167–174. IEEE (2009)
13. Low, K.L.: Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration, vol. 4. University of North Carolina, Chapel Hill (2004)
14. Moenning, C., Dodgson, N.A.: Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory (2003)

15. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. ACM Trans. Gr. **24**, 471–478 (2005)
16. Papazov, C., Burschka, D.: Deformable 3D shape registration based on local similarity transforms. In: Computer Graphics Forum, vol. 30, pp. 1493–1502. Wiley (2011). https://doi.org/10.1111/j.1467-8659.2011.02023.x
17. Rouhani, M., Boyer, E., Sappa, A.D.: Non-rigid registration meets surface reconstruction. In: 2014 2nd International Conference on 3D Vision (3DV), vol. 1, pp. 617–624. IEEE (2014)
18. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
19. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Symposium on Geometry Processing, vol. 4, p. 30 (2007)
20. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Trans. Gr. **23**, 399–405 (2004)
21. Sumner, R.W., Schmid, J., Pauly, M.: Embedded deformation for shape manipulation. ACM Trans. Gr. **26**, 80 (2007)
22. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Trans. Gr. **24**, 553–560 (2005)
23. Süßmuth, J., Winter, M., Greiner, G.: Reconstructing animated meshes from time-varying point clouds. In: Computer Graphics Forum, vol. 27, pp. 1469–1476. Wiley (2008)
24. Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L.: Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. IEEE Trans. Vis. Comput. Gr. **19**(7), 1199–1217 (2013)
25. Wand, M., Adams, B., Ovsjanikov, M., Berner, A., Bokeloh, M., Jenke, P., Guibas, L., Seidel, H.P., Schilling, A.: Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. ACM Trans. Gr. **28**(2), 15 (2009)
26. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic huber-l1 optical flow. In: BMVC, vol. 1, p. 3 (2009)
27. Yang, J., Li, K., Li, K., Lai, Y.K.: Sparse non-rigid registration of 3D shapes. In: Computer Graphics Forum, vol. 34, pp. 89–99. Wiley (2015). https://doi.org/10.1111/cgf.12699
28. Yeh, I.C., Lin, C.H., Sorkine, O., Lee, T.Y.: Template-based 3d model fitting using dual-domain relaxation. IEEE Trans. Vis. Comput. Gr. **17**(8), 1178–1190 (2011)
29. Yoshiyasu, Y., Ma, W.C., Yoshida, E., Kanehiro, F.: As-conformal-as-possible surface registration. In: Computer Graphics Forum, vol. 33, pp. 257–267. Wiley (2014). https://doi.org/10.1111/cgf.12451
30. Zach, C., Pock, T., Bischof, H.: A duality based approach for real-time TV-l 1 optical flow. In: Joint Pattern Recognition Symposium, pp. 214–223. Springer (2007)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Tao Jiang received the B.Sc. and M.Sc. degrees from the University of Electronic Science and Technology of China (UESTC). He is currently a Ph.D. student in the Faculty of Media & Communication at Bournemouth University. His research interests include computer vision and computer graphics, specifically, SLAM, shape deformation and registration.

Xiaosong Yang is currently an Associate Professor in the National Centre for Computer Animation, Bournemouth University, UK. He received his bachelor (1993) and master degrees (1996) in computer science from Zhejiang University (P. R. China) and Ph.D. (2000) in computing mechanics from Dalian University of Technology (P. R. China). He worked as PostDoc (2000–2002) in the Department of Computer Science and Technology of Tsinghua University for two years and as Research Assistant (2001–2002) at Chinese University of Hong Kong. His research interests include deep learning, computer vision, computer animation, motion capture and synthesis, VR&AR, special effects and game development, digital health, data mining, medical visualization. He has published over 70 papers in journals and refereed conferences.

Jianjun Zhang is Professor of Computer Graphics at the National Centre for Computer Animation, Bournemouth University. He is Head of the National Research Centre for Computer Animation. His research interests include computer graphics, computer animation, physically based simulation, geometric modeling, medical simulation and visualization.

Feng Tian received the Ph.D degree from Xi'an Jiaotong University, China. Currently he is an Associate Professor Bournemouth University (BU), UK. He was an Assistant Professor in Nanyang Technological University in Singapore before joining BU in 2009. His current research interests include computer graphics, computer games, augmented reality, machine learning and image processing.

Shuang Liu received his B.S. degree in computer science from the Hebei University of Technology (2014) and Ph.D. (2018) from the National Centre for Computer Animation at Bournemouth University. His research interests include computer vision and computer animation.

Nan Xiang received his B.S. degree from the School of Software, Nanchang University, China, in 2012, and the MA degree from School of Animation and Digital Arts, Communication University of China, in 2017. He is currently a Ph.D student at the National Centre for Computer Animation, Bournemouth University, UK. His research interests include virtual reality and 3D reconstruction.

Kun Qian is an research associate at Bioengineering Department of King's College, London. His research mainly focuses on MRI compatible virtual reality, virtual surgery, eye tracking, visual effect and game technologies. Kun obtained Ph.D from National Center for Computer Animation, Bournemouth University on the topic of virtual surgery. He received his bachelor's degree in applied mathematics and master's degree in computer science from the University of Electronic Science and Technology of China (UESTC).