A Generic Local Deformation Model for Shape Registration

Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, Nikolaos Paragios

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Generic Local Deformation Model for Shape Registration

Yun Zeng — Chaohui Wang — Yang Wang — Xianfeng Gu — Dimitris Samaras — Nikos Paragios

N° 7676

July 2011

ISSN 0249-6399 ISRN INRIA/RR--7676--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

A Generic Local Deformation Model for Shape Registration

Yun Zeng [*] , Chaohui Wang[†‡] , Yang Wang[*] , Xianfeng Gu[*] , Dimitris Samaras[*] , Nikos Paragios[†‡]

Thème : Recalage de Surfaces
Équipe-Projet Gallen

Rapport de recherche n° 7676 — July 2011 — 27 pages

Abstract: In this paper, we propose a new surface registration approach using a generic deformation model, which is efficient to compute and flexible to represent arbitrary local shape deformations. From Riemannian geometry, local deformation at each point of a surface can be characterized by the eigenvalues of a special transformation matrix between two canonically parameterized domains. This local transformation specifies all the deformations (*i.e.*, diffeomorphisms) between surfaces while being independent of both intrinsic (parametrization) and extrinsic (embedding) representations. In particular, we show that existing deformation representations (*e.g.*, isometry or conformality) can be viewed as special cases of the proposed local deformation model. Furthermore, a computationally efficient, closed-form solution is derived in the discrete setting via finite element discretization. Based on the proposed deformation model, the shape registration problem is formulated as a high-order Markov Random Field (MRF) defined on the simplicial complex (*e.g.*, planar or tetrahedral mesh). An efficient high-order MRF optimization algorithm is designed in the paper for such a special structured MRF-MAP problem, which can be implemented in a distributed fashion and requires minimal memory. Finally, we demonstrate the speed and accuracy performance of the proposed approach in the applications of shape registration and tracking.

Key-words: 3D Shape Registration, Riemannian Geometry, High-order Markov Random Fields

[*] Stony Brook University, USA
[†] Laboratoire de Mathématiques Appliquées aux Systèmes, École Centrale de Paris, France
[‡] Equipe GALEN, INRIA Saclay - Île de France, Orsay, France

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Un Modèle Générique de Déformation pour le Recalage de Surfaces

Résumé : Dans ce rapport, nous proposons une nouvelle méthode de recalage de surfaces utilisant un modèle générique de déformation dont l'évaluation est peu coûteuse et qui permet de représenter des déformations locales arbitraires. En géométrie Riemannienne, la déformation locale en chaque point d'une surface peut être caractérisée par les valeurs propres d'une matrice de transformation reliant les parametrisations canoniques des surfaces. Cette transformation locale permet de décrire tous les difféomorphismes possibles entre surfaces tout en étant indépendante des représentations intrinsèques et extrinsèques. En particulier, nous montrons que les modèles de déformation usuels tels que les transformations isométries et conformes sont des cas particuliers de notre modèle. D'autre part, une solution analytique est développée dans un cadre discret en utilisant des éléments finis. Basé sur notre modèle de déformation, le problème de recalage des surfaces est formulé comme un Champs de Markov Aléatoires (MRF) d'ordre élevé défini sur le complexe simplicial. Afin de résoudre le problème de maximisation a posteriori de ce type de champs de Markov, nous introduisons un algorithme d'optimisation pour les potentiels d'ordre élevé à la fois peu gourmand en mémoire et parallélisable. Pour finir, nous évaluons la vitesse et la précision de notre approche dans le cadre d'applications de recalage et de suivi de surfaces.

Mots-clés : Recalage de Surfaces 3D, Géométrie Riemannienne, Champs de Markov Aléatoires d'ordre élevé

Contents

1 Introduction

The problem of shape registration has become increasingly important in computer vision due to the wide availability of acquisition devices [39, 24, 1]. Understanding the transformation between two shapes (*i.e.*, shape matching) is a fundamental task in shape analysis (*e.g.*, classification and recognition) and inference (*e.g.*, registration and tracking). Existing methods of matching shapes are often limited by the assumptions of the underlying deformations. In this paper, we present a generic shape deformation model that can represent arbitrary deformations between two shapes.

Our new deformation model is inspired by the basic ideas in Riemannian geometry [10] in which a shape is equipped with a metric tensor at each point in the parametrization domain. The problem of matching two shapes is hence equivalent to finding correspondences between their parameterized domains (Fig. 1). Different parametrizations represent the same surface (modulo the isometric deformation) if and only if for each correspondence, their metric tensors satisfy certain transformation rules. Hence, for surfaces undergoing isometric deformations, the task of matching becomes finding consistent parametrization such that the transformation rules are satisfied. To characterize more general deformations, at a given point, we consider a special *canonical parametrization* of the shape whose metric tensor *at that point* is Euclidean. Arbitrary deformations can be described by the local distortion of a circle between each pair of correspondence in the canonical parametrization domain [2, 26]. Such distortion can be consequently characterized by the eigenvalues of a canonical transformation relating to the Jacobian matrix between the two canonical parametrization domains (the *canonical distortion coefficient*). In the discrete setting, we consider the common case that a shape is represented as simplicial complex (*e.g.*, a planar or tetrahedral mesh). By assuming the deformation of the shape in the parametrization domain to be piecewise linear, the problem of computing the canonical distortion coefficient at a point becomes computing it at a face. The canonical parametrization at a point is simply equivalent to mapping each face to the 2D domain. Within this setting, the computation of the canonical distortion coefficient for each face becomes solving linear equations with a closed-form solution.

Given the above shape deformation model, the problem of finding the optimal shape registration result that best fits the deformation prior and the observed data can be formulated using the Markov Random Field (MRF) framework [13]. Recent combinatorial methods (*e.g.*, [28, 37]) have demonstrated the superiority of discrete optimization with respect to both solution quality and computational speed. Moreover, in our problem, an MRF formulation allows us to take multiple matching criteria into the same optimization framework. Because the deformation model is defined on each facet (*e.g.*, triangle for planar mesh), the MRF optimization involves high-order cliques. Compared to existing approaches, our method has the following advantages:

- The proposed deformation model is generic and flexible to handle arbitrary deformations. Most of the previous geometry-based deformation models (*e.g.*, isometry and conformality) fall into certain special classes of our model. As a result, our model can accurately describe the deviation from those existing models, which is important in characterizing real-world deformations.
- In the discrete setting, the computation of the canonical distortion coefficient for each basic element (*e.g.*, a triangle for a planar mesh) can be computed efficiently by solving linear equations with a closed-form solution.

- An efficient high-order MRF optimization algorithm is designed and implemented based on the dual-optimization technique ([33]) which requires minimal memory and achieves a good approximate solution.

In summary, our paper proposes a generic deformation descriptor: the canonical distortion coefficient (CDC), and its use in a high-order MRF framework that allows us to impose the deformation of a template as a prior in a Bayesian setting. The novel formulation of the surface registration problem as an MRF problem with a cell complex structure seamlessly brings together Riemannian geometry and modern graphical model. Furthermore, our algorithm can be implemented in distributed hardware that achieves significant speedup[1]. In our experiment, we demonstrate that the MRF optimization technique combined with the generalized deformation model leads to significant improvement for the problem of surface registration and tracking.

This paper is organized as follows: In Sec. 2 we review related work on shape deformation models. The mathematical formulation of our general shape deformation model is presented in Sec. 3, where both continuous concepts and discrete counterparts are discussed. A high-order MRF formulation for shape registration is given in Sec. 4, with the design of an efficient optimization algorithm. In Sec. 5 we show the applications of our new deformation model and optimization technique to the problems of shape registration and tracking. Finally, we conclude our work in Sec. 6.

2 Related work

Accurately modeling surface deformation is a very challenging task due to the large degrees of freedom that arbitrary deformations may present. Previously, the “rigidity" assumption is widely used due to its good approximation to many real-world deformations and simplicity in computation. Existing deformation models either characterize the *rigidity* in the extrinsic (embedding) space or in the intrinsic (parametrization) space.

Assuming two representations of a shape in the embedding space only differ by a global rigid deformation (*i.e.*, rotation and translation), the iterative closest points (ICP) [3] method has been successfully applied for shape registration with various extensions (*e.g.*, [7, 14]). However, global rigidity does not take into account bendable shapes (*e.g.*, garments or rubber bands). Hence the notion of *local rigidity* has been proposed which assumes that the deformation between two local neighborhoods of each correspondence be rigid. Although this increases the degrees of freedom of the deformation, efficient algorithms have been proposed that achieves local optimality for such deformation energy, *e.g.*, [22, 25, 27, 8].

When the shapes undergo large deformations or are represented in different embedding space, intrinsic methods (*e.g.*, [5, 6, 32, 31, 35, 36]) become useful. The basic idea of intrinsic methods is from Riemannian geometry [10], assuming a shape be equipped with a metric measuring the distance between any two points. The notion of rigidity (known as *isometry*) can be best characterized by assuming the metric does not change across two shapes. Hence matching two surfaces becomes finding consistent parametrization that best match the metric. However, finding a globally consistent parametrization is often too restricted. Recently work ([36, 19, 37, 18, 38]) search among multiple parametrization and use multiple matching criteria (*e.g.*, texture

[1]The source code and executable for the CUDA implementation of the high-order MRF optimization can be downloaded from `http://www.cs.sunysb.edu/~yzeng/software_HighorderMRF.html`

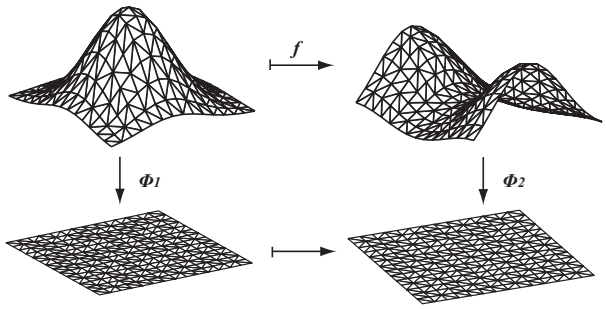


Figure 1: The problem of finding correspondences between two surfaces can be reduced to finding correspondences between their parametrization domain. The problem we solve in this paper is: given a predefined points on the first surface, find the correspondences of only those points on the second one. Efficient candidate selection schemes for surface registration and tracking are described in [37, 38].

or curvature) to improve the matching accuracy, though *their underlying deformation model is still based on isometry assumption*.

3 Canonical distortion coefficient

In this section, we present the mathematical formulation of our deformation model in both continuous and discrete settings.

3.1 Continuous setting

3.1.1 Riemannian metric and parametrization

Let $(\mathcal{M}, g_{\mathcal{M}})$ denote a surface $\mathcal{M}$ equipped with a Riemannian metric $g_{\mathcal{M}}$. In Riemannian geometry ([10]), a surface is described by its local charts $\{(U_\alpha\}$, *i.e.*, $\mathcal{M} = U_\alpha \cup U_\beta \ldots$ and each open subset U_α is in $1-1$ correspondences $\phi_\alpha : U_\alpha \to \mathbb{R}^2$. Here ϕ_α is the local *parametrization*. For any $p \in U_\alpha \subset \mathcal{M}$, a metric tensor is associated to p as a symmetric positive definite matrix:

$$g^\alpha(p) = \begin{pmatrix} g^\alpha_{11}(p) & g^\alpha_{12}(p) \\ g^\alpha_{21}(p) & g^\alpha_{22}(p) \end{pmatrix}. \tag{1}$$

Within such setting, a point on $\mathcal{M}$ may be covered by multiple charts. In order for different local representations to describe the same surface, certain transformation rules must be satisfied, *i.e.*, if a point $p \in U_\alpha \cap U_\beta$, we must have

$$g^\alpha(p) = J_{\alpha\beta}(p)^T g^\beta(q) J_{\alpha\beta}(p). \tag{2}$$

Here $J_{\alpha\beta}$ is the Jacobian matrix of the transformation $x_\alpha(x_\beta)$ ($x_\alpha(x_\beta)$ is the local coordinate system of patch $U_\alpha(U_\beta)$) between charts U_α and U_β, *i.e.*,

$$J_{\alpha\beta} = \begin{pmatrix} \frac{\partial x_\alpha^1}{\partial x_\beta^1} & \frac{\partial x_\alpha^1}{\partial x_\beta^2} \\ \frac{\partial x_\alpha^2}{\partial x_\beta^1} & \frac{\partial x_\alpha^2}{\partial x_\beta^2} \end{pmatrix}. \tag{3}$$

Any local representation satisfying this transformation rule is a valid parametrization of the surface. Therefore, since the metric tensor at any point $p \in \mathcal{M}$ is positive definite, it is always possible to apply a proper linear transformation to its parametrization ϕ_α such that $g^\alpha(p)$ is the identity matrix. We call such a parametrization the *canonical parametrization* for point p:

Definition 1. ***(Canonical parametrization at a point)*** *For any $p \in \mathcal{M}$, a parametrization $\phi_\alpha : \mathcal{U}_\alpha \to \mathbb{R}^2$ is called* canonical parametrization for p *if the metric tensor at p is the identity matrix.*

Please note that although there exists infinite number of such a canonical parametrization for a surface, the parametrization at the particular point is unique. In the following we shall show that focusing on the parametrization only at one point at a time allows us to *characterize arbitrary deformations between two surfaces while regardless of both the intrinsic and extrinsic representations of the surface*, which is the main advantage of our new deformation model.

3.1.2 Diffeomorphisms between two shapes

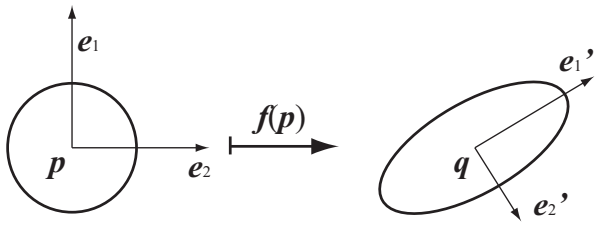
Now we consider arbitrary diffeomorphisms between the parametrization domains of two surfaces (Fig. 1). For any correspondence $p \in \mathcal{U}_\alpha \subset \mathcal{M} \to q \in \mathcal{U}_\beta \subset \mathcal{N}$, the change of metric $g^\alpha(p) \to J_{\alpha\beta}(p)^T g^\beta(q) J_{\alpha\beta}(p)$ reflects how locally a circle is deformed into an eclipse (Fig. 2(a))[2]. In particular, under canonical parametrization for points p and q, the matrix $J_{pq}^T J_{pq}$ accurately characterizes such local deformation, where J_{pq} is the Jacobian at point p. If we only consider the change of shape, *i.e.*, how a circle is deformed into an eclipse regardless its orientation, the eigenvalues λ_1, λ_2 of $J_{pq}^T J_{pq}$ can best describe such change. Therefore, the local deformation between two surfaces can be characterized by the eigenvalues λ_1, λ_2 for each correspondence. Formally, we define:

Definition 2. ***(Canonical distortion coefficient)*** *The eigenvalues of the Jacobian transformation matrix $J_{pq}^T J_{pq}$ between any canonical parametrization at p and q are the* canonical distortion coefficients *between the two points.*

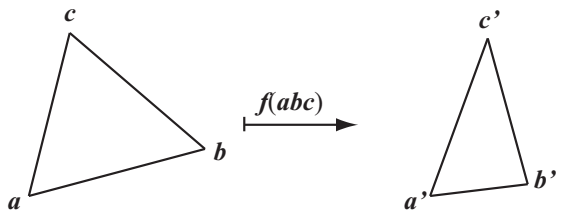
We call the Jacobian matrix J_{pq} between the two points p and q under the canonical parametrization the *canonical Jacobian*. The significance of using canonical Jacobian is that it allows us to *characterize arbitrary deformation by considering the parametrization of the surface at a single point*. Some special cases of deformations can be characterized by the canonical distortion coefficient as follows:

1. In the case of the isometric deformation, a unit circle is mapped to a unit circle, *i.e.*, $\lambda_1 = \lambda_2 = 1$.
2. In the case of the conformal deformation, a unit circle can be mapped to a circle with arbitrary radius [26]. Thus, $\lambda_1 = \lambda_2$.

[2] A rigorous formulation can be found in the Appendix.



(a) Continuous setting



(b) Discrete setting

Figure 2: The finite element method assumes the transformation between facets to be piecewise linear and $f(\vec{ab}) = \vec{a'b'}, f(\vec{ac}) = \vec{a'c'}$. Under the linearity assumption, the Jacobian can be computed in a closed form for each triangle pair $\triangle abc \mapsto \triangle a'b'c'$.

To further connect the canonical distortion coefficient to a general class of diffeomorphisms defined in the complex plane $f : \mathcal{U}_\alpha \rightarrow \mathcal{U}_\beta$, between any canonical parametrization x_α and x_β for p and q respectively, we define

$$\frac{\partial f}{\partial z} = \frac{1}{2}(\frac{\partial x^1_\alpha}{\partial x^1_\beta} + \frac{\partial x^2_\alpha}{\partial x^2_\beta}) + \frac{i}{2}(\frac{\partial x^2_\alpha}{\partial x^1_\beta} - \frac{\partial x^1_\alpha}{\partial x^2_\beta})$$
$$\frac{\partial f}{\partial \overline{z}} = \frac{1}{2}(\frac{\partial x^1_\alpha}{\partial x^1_\beta} - \frac{\partial x^2_\alpha}{\partial x^2_\beta}) + \frac{i}{2}(\frac{\partial x^2_\alpha}{\partial x^1_\beta} + \frac{\partial x^1_\alpha}{\partial x^2_\beta}),$$

The notion of quasi-conformality [2] can be characterized by the following *Beltrami-coefficient*:

$$\mu(z) \equiv \frac{\partial f}{\partial \overline{z}} / \frac{\partial f}{\partial z},$$

which gives us all the information about the conformality of f. Suppose $\lambda_1 \geq \lambda_2$, it can be shown that $|\mu(z)| = (\sqrt{\lambda_1} - \sqrt{\lambda_2})/(\sqrt{\lambda_1} + \sqrt{\lambda_2})$. In particular, f is called *holomorphic* if $\mu(z) = 0$ ([12]), *i.e.*, $\lambda_1 = \lambda_2$, coinciding with the fact that holomorphic function is another description of conformal mapping. Hence the Beltrami-coefficient generalizes conformal mapping and can be partially determined using the canonical distortion coefficient. However, the Beltrami-coefficient is for surface parametrization, where scaling factor is lost. The proposed canonical distortion coefficient preserves the scale information which is important for shape matching. Besides, the canonical distortion coefficient is directly extendable to nD. In this paper, we propose an efficient approach to compute the local deformation based on the canonical distortion coefficient in the context of shape registration.

3.2 Finite element discretization

The basic assumption in finite element analysis [4] is that the continuous space can be approximated using a set of *basic elements* (*e.g.*, polynomial functions defined on each face). Meanwhile, consistencies must be preserved at the boundaries among the basic elements. In this paper, we consider the most common representation of a continuous surface – a triangular mesh, whose basic finite element is a triangular face. In this discrete setting, the canonical distortion coefficient (CDC) is assumed to be constant for each basic element (*i.e.*, $\triangle abc$ shown in Fig. 2).

Thus, the concept of canonical parametrization (Sec. 3.1.1) can be expressed in the following manner: a parametrization of a point p is locally Euclidean at p if the images of any two tangent vectors have the same angle and length. In the discrete setting, this means a triangle $\triangle abc$ keeps all its angles and lengths in its parametrization, which can be achieved by simply mapping the face $\triangle abc$ onto a 2D domain by keeping all its edge lengths.

Finally, we consider the computation of the canonical Jacobian (Sec. 3.1.2). In the continuous setting, the Jacobian matrix at a point p is a linear operation that transforms tangent vectors at p to tangent vectors at q. Given a basic element $\triangle abc$ in the discrete setting, the tangent space at p is equivalent to the linear space spanned by $\triangle abc$. Hence the linear mapping $J(\cdot)$ between two canonical domains should satisfy $J(\vec{ab}) = \vec{a'b'}$ and $J(\vec{ac}) = \vec{a'c'}$. The Jacobian of a linear transformation between two triangles is a 2×2 matrix and can be computed in closed-form. Since $J(\cdot)$ is linear, $J(\vec{bc}) = \vec{b'c'}$ must be satisfied as well, i.e.,

$$\text{Jacobian for mapping } p \rightarrow q \Leftrightarrow \text{Linear transformation matrix for mapping } \vec{ab} \rightarrow \vec{a'b'}, \vec{ac} \rightarrow \vec{a'c'}$$

For clarity, Alg. 1 summarizes the algorithm for computing the canonical distortion coefficient. Note that the computation is in analogy with previous work for surface parametrization (*e.g.*, [20, 21, 23]) since both are based on the same piecewise linear assumption. However, here we derive it from a different continuous setting in the context of shape deformation estimation. Also note that when the shape is n-manifold, the computation of CDC becomes solving n linear equations and eigenvalues.

Algorithm 1: Algorithm for computing the canonical distortion coefficient (CDC) for each triangular face.

Input : $\triangle abc$ and its mapping $\triangle a'b'c'$
Output : CDC for mapping from $\triangle abc$ to $\triangle a'b'c'$.

Step One: Map the triangles $\triangle abc$ and $\triangle a'b'c'$ to 2D and keep their orientation.
Step Two: Compute the 2×2 linear transformation J mapping $\vec{ab}$ to $\vec{a'b'}$ and $\vec{ac}$ to $\vec{a'c'}$.
Step Three: Compute the eigenvalues, λ_1 and λ_2 of $J^T J$.
Step Four: Output λ_1 and λ_2

4 High-order MRF-based shape registration

4.1 MRF formulation for shape registration

Given the canonical distortion coefficient (CDC) defined for each basic element, *i.e.*, the triangular face, one can either deform the original shape (*e.g.*, [25]), or find the

correspondences between two shapes combined with other matching cues (*e.g.*, [37]). Here we consider the problem of finding the mapping f between two shapes $\mathcal{M}$ and $\mathcal{N}$. Similar to [37], we assume a set of n points $\mathcal{V} = \{p_u | p_u \in \mathcal{M}, u = 1, \ldots, n\}$ are sampled on the surface $\mathcal{M}$ and a triangulation of these points are constructed (Fig. 1). Hence the task of shape registration becomes finding the correspondences for the set $\mathcal{V} \subset \mathcal{M}$ on shape $\mathcal{N}$.

To formulate the shape matching problem using graphical models, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{F} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ is a set of faces. For each $u \in \mathcal{V}$, let the random variable $x_u \in \mathcal{L} = \{1, \ldots, L\}$ denote the discrete labeling of all the possible matching candidates of vertex $u \in \mathcal{M}$ on shape $\mathcal{N}$. In the rest of this paper, we use x_u to denote the labeling of vertex u or the point on shape $\mathcal{N}$ when it is clear from the context.

Firstly, let $\text{fea}(\cdot)$ be the feature vector (*e.g.*, texture or shape context) at each point on the shape. We define the cost function $\theta_u(x_u)$ of matching u to x_u:

$$\theta_u(x_u) = |\text{fea}_{\mathcal{M}}(u) - \text{fea}_{\mathcal{N}}(x_u)|^2.$$

Next, we denote by $\lambda_{\mathcal{M}}(u, v, w) = \begin{pmatrix} \lambda_1(u, v, w) \\ \lambda_2(u, v, w) \end{pmatrix}$ as the prior knowledge of CDC that characterizes the deformation for the face (u, v, w) and $\lambda_{\mathcal{N}}(x_u, x_v, x_w) = \begin{pmatrix} \lambda_1(x_u, x_v, x_w) \\ \lambda_2(x_u, x_v, x_w) \end{pmatrix}$ as the CDC computed from deforming $\triangle uvw$ to $\triangle x_u x_v x_w$ (Alg. 1). Hence we define:

$$\theta_{uvw}(x_u, x_v, x_w) = d(\lambda_{\mathcal{M}}(u, v, w), \lambda_{\mathcal{N}}(x_u, x_v, x_w))$$

as the deformation energy. Here $d(\cdot, \cdot)$ is the distance function that is defined according to the application.

Finally, given the feature function $\text{fea}(\cdot)$ and the deformation prior $\lambda_{\mathcal{M}}(\cdot, \cdot, \cdot)$ for each vertex and face, the problem of shape registration becomes solving the best configuration $\mathbf{x}$ that minimizes the following energy:

$$\min_{\mathbf{x}} E(\mathbf{x}) = \sum_{u \in \mathcal{V}} \theta_u(x_u) + \sum_{(u,v,w) \in \mathcal{F}} \theta_{uvw}(x_u, x_v, x_w). \tag{4}$$

To reduce the search space $\mathcal{L}$ and avoid local minima, we adopt a hierarchical optimization scheme in solving the energy function (4). Inspired by [37], sparse feature points are first selected to compute the initial matching with the global constraint $\lambda_1 = \lambda_2 = 1$ for each triangle $(u, v, w) \in \mathcal{F}$. Then, a small set of candidates $\mathcal{L}$ are selected to find the best local match by minimizing the energy function (4).

4.2 An efficient high-order MRF optimization

The high-order potential of Eq. 4 presents the difficulty for solving our surface registration problem. Existing algorithms either transform high-order cliques into pairwise ones (*e.g.*, [15]) or decompose the original problem into a union of sub-problems (*e.g.*, [17]). However, these algorithm are designed for general problems and requires extra memory to store the transformed problem or the dual variables that relates the sub-problems. For the problem of high-order MRF inference, memory efficiency is an important issue since a poorly designed optimization algorithm can easily reach the limit of current hardware. In this paper, we follow the framework by Thomas [33] to design a fast and memory efficient MRF optimization algorithm using the linear programming (LP) relaxation technique.

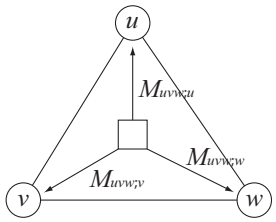


Figure 3: An example of the messages defined for the dual problem of Eq. 5. For each face $\triangle uvw$, the messages are defined from the high-order clique (u, v, w) to each of the nodes u, v and w.

The first step in the LP relaxation of Eq. 4 is to introduce the indicator variables. For any $u \in \mathcal{V}$ and $i \in \mathcal{L}$, we define

$$\tau_{u;i} = \begin{cases} 1 & \text{if } x_u = i \\ 0 & \text{otherwise} \end{cases}.$$

Also for any $(u, v, w) \in \mathcal{F}$ and $(i, j, k) \in \mathcal{L} \times \mathcal{L} \times \mathcal{L}$, we define

$$\tau_{uvw;ijk} = \begin{cases} 1 & \text{if } x_u = i, x_v = j, x_w = k \\ 0 & \text{otherwise} \end{cases}.$$

Similarly, we define $\theta_{u;i} = \theta_u(i)$ and $\theta_{uvw;ijk} = \theta_{uvw}(i, j, k)$. Hence we have the integer LP formulation of the problem of Eq. 4:

$$\begin{aligned}
\min_{\tau} & \sum_{u \in \mathcal{V}} \sum_{i \in \mathcal{L}} \theta_{u;i} \tau_{u;i} + \sum_{(u,v,w) \in \mathcal{F}} \sum_{(i,j,k) \in \mathcal{L}^3} \theta_{uvw;ijk} \tau_{uvw;ijk} \\
\text{s.t.} & \sum_{i} \tau_{u;i} = 1, && \forall u \in \mathcal{V} \\
& \sum_{i,j,k} \tau_{uvw;ijk} = 1, && \forall (u, v, w) \in \mathcal{F} \\
& \sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}, && \forall (u, v, w) \in \mathcal{F} \text{ and } i \in \mathcal{L} \\
& \tau_{u;i}, \tau_{uvw;ijk} \in \{0, 1\}.
\end{aligned}$$

By relaxing the variables to the domain $[0, 1]$, we obtain the dual form of the above LP problem as

$$\begin{aligned}
\max_{M} & \sum_{u} \min_{i} \overline{\theta}_{u;i} + \sum_{(u,v,w) \in \mathcal{F}} \min_{i,j,k} \overline{\theta}_{uvw;ijk} \qquad (5) \\
\text{s.t. } & \overline{\theta}_{u;i} = \theta_{u;i} + \sum_{(u,v,w) \in \mathcal{F}} M_{uvw;u:i}, \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L} \\
& \overline{\theta}_{uvw;ijk} = \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \\
& \forall (u, v, w) \in \mathcal{F} \text{ and } (i, j, k) \in \mathcal{L} \times \mathcal{L} \times \mathcal{L}.
\end{aligned}$$

Here $M_{uvw;u:i}$ is the dual variable (message) corresponding to the constraint $\sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}$. (Fig. 3).

Algorithm 2: Min-sum diffusion algorithm.

repeat
  for each $M_{uvw;u:i}$ **do**
    $M_{uvw;u:i} -= \frac{1}{2}[\theta_{u;i} - \min_{j,k} \theta_{uvw;ijk}]$ and reparameterize $\theta_{u;i}$ and $\theta_{uvw;ijk}$ according to the constraints in Eq. 5.
  end for
until convergence

The dual problem of Eq. 5 can be solved by the simple min-sum diffusion algorithm [33] as shown in Alg. 2 (Note that in [33] their optimization problem is a maximization so the algorithm is called max-sum). It has been shown that at convergence, the solution satisfies J-consistency condition as introduced in [33]. Since after each update of the message, reparameterization is performed, no extra memory is needed for storing all the dual variables $M_{uvw;u:i}$. Hence the memory requirement for the Alg. 3 is the storage for the primal variables, *i.e.*, $O(|V||\mathcal{L}| + |\mathcal{F}||\mathcal{L}|^3)$, which can not be avoided by any algorithm.

Each update of the message in Alg. 2 only involves the parameters in a triangle. Also within each face $\triangle uvw$, the update of each label $M_{uvw;u:i}, i = \{1, \ldots, L\}$ is independent. Hence the algorithm can be significantly accelerated using distributed hardware.

In order to explore the parallelism of the min-sum Algorithm 2, we define the concept of independent face set:

Definition 3. ***(Independent face set)*** *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$, a subset $\mathcal{F}_k \subset \mathcal{F}$ is called* independent face set *if for any $f_i, f_j \in \mathcal{F}_k$, $f_i \cap f_j = \varnothing$.*

The decomposition of a set $\mathcal{F}$ into subsets of independent face sets $\mathcal{F} = \cup_i \mathcal{F}_i$ can be efficiently computed in polynomial time by a simple greedy algorithm. Hence we can implement Alg. 2 in parallel as shown in Alg. 3. The maximal speedup achieved

Algorithm 3: Parallel min-sum diffusion algorithm.

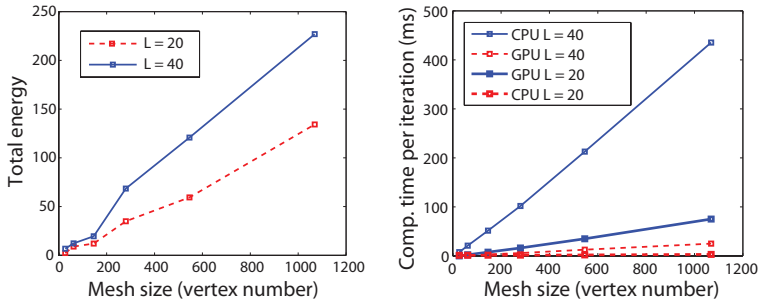
Decompose $\mathcal{F}$ into independent face sets $\cup_i \mathcal{F}_i$
repeat
  for each Independent face set $\mathcal{F}_i$, in parallel for all $(u, v, w) \in \mathcal{F}_i$ and $k \in \mathcal{L}$ **do**
    Update the message $M_{uvw;u:k}$, $M_{uvw;v:k}$ and $M_{uvw;w:k}$ and do reparameterization (Alg. 2).
  end for
until convergence

in Alg. 3 is $\max_i(|\mathcal{F}_i||\mathcal{L}|)$.

4.3 Performance evaluation

We implement Alg. 3 using the NVIDIA® CUDA architecture [16][3]. In approximation algorithms[29], the approximation error (AE) is defined as the gap between the optimal

[3]The source code for CUDA implementation of the MRF optimization algorithm and the executables used in our comparison between CPU and GPU implmentations can be downloaded from `http://www.cs.sunysb.edu/~yzeng/software_HighorderMRF.html`.



(a) (b)

Figure 4: Performance analysis of our MRF optimization algorithm. (a) shows the optimality using the test cases described in Sec. 4.3. (b) shows the speedup using the parallel implementation of Alg. 3. We show the runtime per iteration since different inputs would result in different iteration counts.

integral solution and the solution obtained by the algorithm. The gap between the result by the LP relaxation and the optimal integral solution is often used to upper bound the approximation error. Here we aim to test the true approximation error, as well as the speedup, by designing the test inputs as follows: Given any input mesh, we randomly assign a default labeling l_u for each node $u \in \mathcal{V}$. We define the singleton potentials of Eq. 4 as

$$\theta_u(x_u) = \begin{cases} 0 & \text{if } x_u = l_u \\ rnd(1) & \text{otherwise} \end{cases},$$

where $rnd(1)$ is a random number between $[0, 1]$. Also we define the high-order potentials as

$$\theta_{uvw}(x_u, x_v, x_w) = \begin{cases} 0 & \text{if } (x_u, x_v, x_w) = (l_u, l_v, l_w) \\ rnd(1) & \text{otherwise} \end{cases}.$$

In such case, the optimal solution of Eq. 4 should be $\{l_u | u \in \mathcal{V}\}$. Fig. 4 (a) shows the result of our algorithm using the above designed test cases for different mesh and label size. Note that although the total energy increases with mesh size, the average energy per term (vertex and face) remains significantly low (< 0.01 for all cases). Fig. 4 (b) shows the comparison on average time taken per iteration, between the implementations with and without GPU accelerations. The total number of iterations depends on the energy. In our experiment the algorithm converges within 3000 iterations. Hence our algorithm is both memory and computational efficient, which is important for shape tracking.

5 Experimental results

In this section, we apply our deformation model and optimization technique to the problems of shape registration and tracking. The input to our algorithm is two 3D shapes $\mathcal{M}$ and $\mathcal{N}$, and a triangulated sampling point set $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ on $\mathcal{M}$ (Fig. 6 (a)). The output is the registered result on $\mathcal{N}$ for each point in $\mathcal{V}$ (Fig. 6 (c)). We implement our algorithm on an Intel® Core(TM)2 Duo 3.16G PC with 4G RAM and

Data	Zeng *et.al.* [37]	Our method
Face_Smile	$(2.26, 0.19, 67.83)$	$(1.24, 0.86, 4.2)$
Face_Laugh	$(1.75, 0.12, 111.11)$	$(1.36, 0.82, 11.0)$
Face_Sad	$(1.87, 0.19, 78.62)$	$(1.48, 0.87, 7.52)$

Table 1: Comparison with [37]. Here $(\cdot, \cdot, \cdot)$ denote the average, minimal and maximal area ratios between the original facets and the matched facets.

an NVIDIA® Geforce 9800GTX+ graphics card with 128 CUDA cores. The number of labels L (matching candidates) is set to be 2^n for best hardware performance. In most of our experiments, we set $L = 64$. The computation of all the L^3 possible CDCs for one face takes only $2.0ms$ on average on GPU. Hence the computation of the energy term $\theta_{uvw;ijk}$ for a mesh with 165 vertices and 272 faces takes only 0.5s.

Estimation of deformation prior: In our experiment, we assume that *the deformation priors are similar across different shapes of the same type* (*e.g.*, human faces [30]). The ground truth deformation prior can be obtained by 3D scanning systems with reliable texture information (*e.g.*, markers). As shown in Fig. 5, the two 3D data in (a) and (c) are captured with markers using the system introduced in [32]. Here we select two frames with the largest expression difference to measure the maximal possible change of CDC. Fig. 5 (b), (d), (f) and (g) show the visualization of the distribution of CDC. From the above data set we obtain the allowed bound for human face expression changing from neutral to large deformation as $I_1 = [0.7, 5.66], I_2 = [0.1, 4]$ for λ_1 and λ_2, respectively. For the problem of surface registration, we define a Potts-like energy for the high-order terms in Eq. 4 as follows:

$$\theta_{uvw}(x_u, x_v, x_w) = \begin{cases} 0 & \text{if } \lambda_1 \in I_1 \text{ and } \lambda_2 \in I_2 \\ 10 & \text{otherwise} \end{cases}, \tag{6}$$

where λ_1 and λ_2 are the CDCs obtained by matching $\triangle uvw$ to $\triangle x_u x_v x_w$.

5.1 Shape registration

Fig. 6 shows one of our shape registration results and its comparison with the results of Zeng et.al [37]. The singleton terms are defined similarly as in [37]. To obtain enough matching candidates, re-sampling is made near its original candidates. The high-order graph matching formulation in [37] assumes the two surfaces are isometrically deformed so they have consistent conformal mapping if three correspondences are found. Hence when the deformation is not isometric, the registered points can be significantly distorted (Fig. 6 (b) and (c)). Besides, the optimization technique in [37] requires reducing high-order terms into pairwise ones so the memory required is huge and it only handle label size < 5. In contrast, our method poses deformation constraint onto each triangle and guarantees the consistency condition for the final solution. As a result, unlikely matching is avoided and the results are visually plausible (Fig. 6 (e) and (f)). More quantitative comparisons are given in Table 1. It can been seen that the unnatural deformations presented in [37] are significantly reduced.

To test the accuracy achieved by using a anisometric deformation prior, defined as the canonical distortion coefficient (CDC) in our paper, we design the following experiment. The 3D scan of a highly deformable toy is captured by the system introduced in [32] before and after a large deformation, as shown in Fig. 7 (a) and (b), respectively.

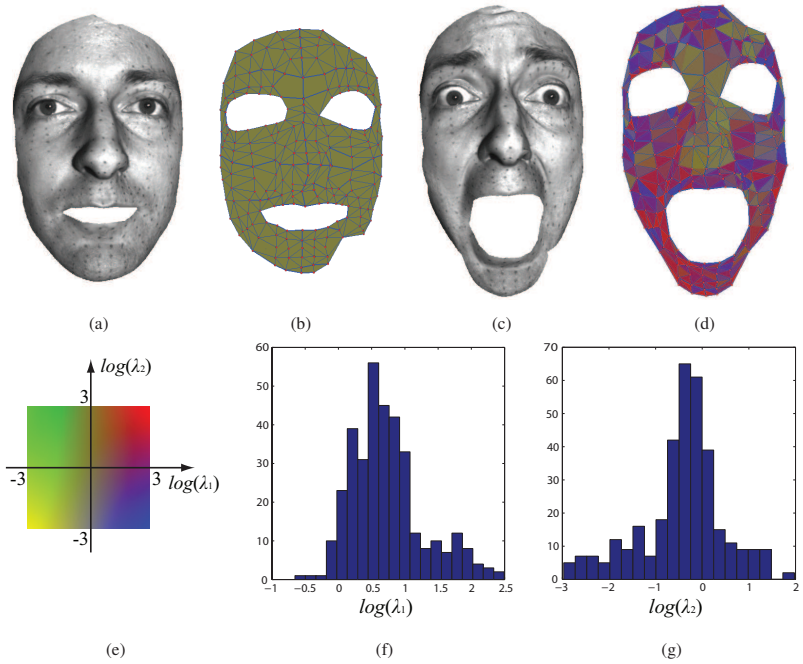


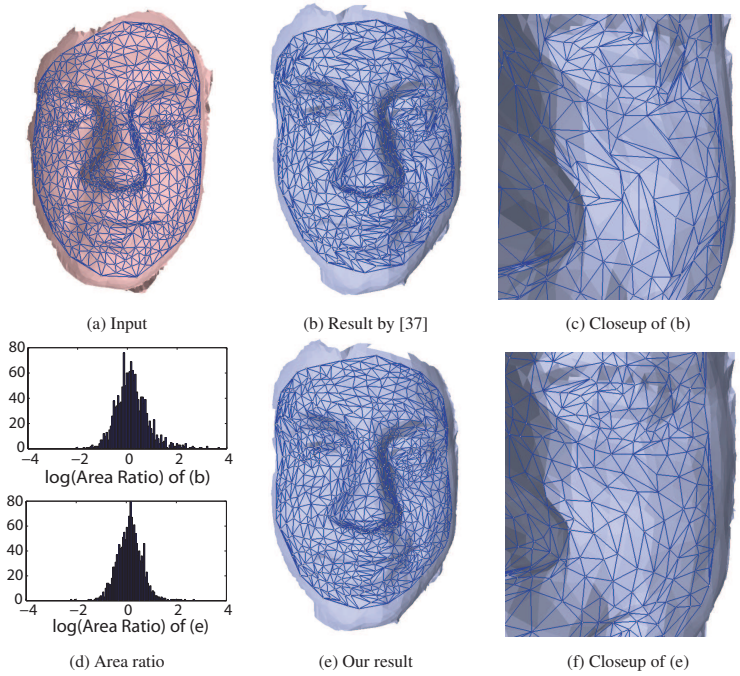
Figure 5: Expression deformation prior is obtained by 3D scanned data with markers. (a) and (b) show the 3D scan of the onset and peak of a facial expression with large shape deformations respectively. (c) and (d) are the corresponding triangular meshes constructed from the 3D scan data. The color coding in (d) shows the deformation intensity as illustrated in (e). The histogram of the canonical distortion coefficient values are shown in (f) and (g).

Outliers are pruned off using the selection tool provided by the MeshLab software [4]. An original set of points and their candidates are selected using the method described in [37]. To establish the ground truth and estimate the deformation prior, we manually select 20 facets and their matches based on the texture information. The two shapes are then matched *without using the texture information*, *i.e.*, in Eq. 4 of in the paper, we use curvature as the singleton term (data likelihood) and the learnt deformation prior as the high-order term (the deformation prior). Fig. 8(b) shows the result using isometric assumption $(\lambda_1, \lambda_2) = (1, 1)$ and Fig. 8(c) shows the result using the learnt CDC prior $(\lambda_1, \lambda_2) = (1.028, 0.993)$. To compare the accuracy achieved by the two assumptions, we compare the average texture difference between the original area covered by the matching points (the blue mesh in Fig. 8) and the matched area.

5.2 Shape tracking

To apply our method to tracking dynamic 3D scanned data, we consider both inter-frame consistency and the consistency between the current frame and the first frame. For the singleton term in Eq. 4, we use the robust metric defined in the paper of [38]. To impose inter-frame consistency, we use the same data set (here we select two con-

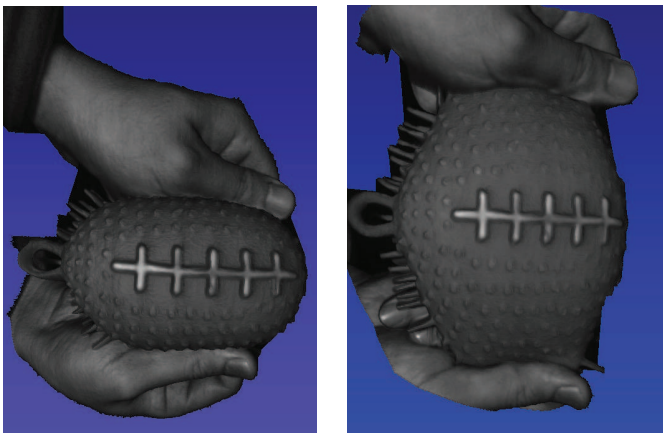
[4] http://meshlab.sourceforge.net/



(a) Input (b) Result by [37] (c) Closeup of (b)

(d) Area ratio (e) Our result (f) Closeup of (e)

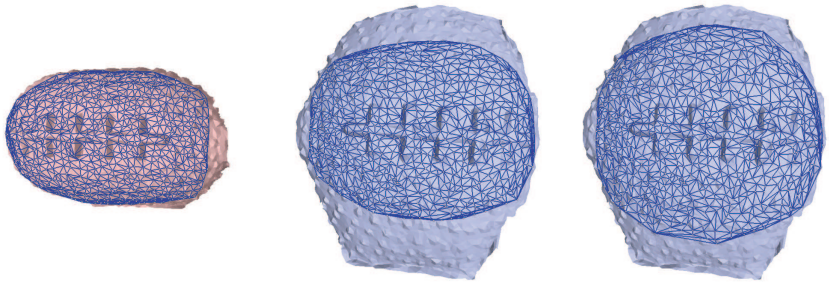
Figure 6: Shape registration result. (a) shows the input mesh with triangulated discrete samples obtained by the method described in [37]. The matching result and its closeup is shown in (b) and (c).



(a) Original shape (b) Deformed shape

Figure 7: The 3D scan of a highly deformable toy.

secutive frames with largest deformation change) and obtain the allowed change of CDC between frames to be $I_1 = [0.874, 1.143]$ and $I_2 = [0.846, 1.182]$ for λ_1 and λ_2



(a) Original shape with discrete samples (b) Result with isometric assumption (c) Result with learnt CDC prior

Figure 8: The comparison between the results of shape registration with isometric assumption (b) and with learnt CDC prior (c). Only curvature and high-order deformation prior are used in the registration.

With isometry assumption	With learnt CDC
0.073 ($(\lambda_1, \lambda_2) = (1, 1)$)	0.005 ($(\lambda_1, \lambda_2) = (1.028, 0.993)$)

Table 2: Comparison between results with and without isometric assumption. The matching is done without using the texture information. The comparison is based on the average texture difference (the gray level is normalized in $[0, 1]$), by warping the source mesh to the matched target mesh.

respectively. Also we handle drift error by imposing the consistency between the first frame and the current frame, using the same face deformation prior learnt in Fig. 5.

Fig. 9 shows tracking results on the BU-4DFE database [34]. A template is constructed in the first frame and tracked in the subsequent frames (same as in [38]. Because of the temporal continuity in consecutive frames, sufficient matching candidates can be obtained by only looking at the neighborhood of each point. In this data set, the texture information is noisy so relying on texture information only can lead to erroneous results. By imposing a simple prior on the bound of the deformation, we have achieved plausible tracking results as shown in Fig. 9.

6 Conclusion

We have presented a generic, geometry-inspired deformation model for characterizing arbitrary diffeomorphisms between shapes. An efficient and accurate algorithm to compute the canonical distortion coefficients (CDCs) is proposed based on finite element analysis in the discrete setting. Searching for the optimal shape registration result given such model is made possible with a high-order MRF framework. An efficient optimization algorithm is designed for such problem. We demonstrated the speed and accuracy performance on 2-manifold shape registration. However, both the deformation model and the optimization algorithm can be easily extended to high-dimensional space.

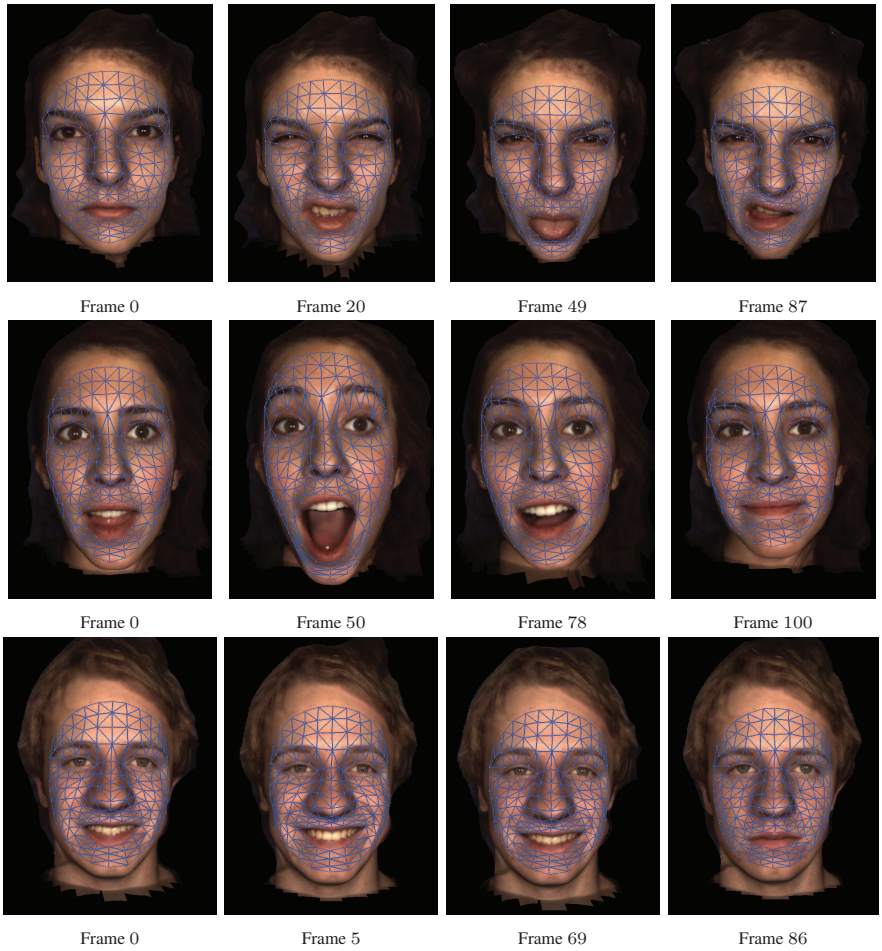


Figure 9: Shape tracking results.

7 Appendix

7.1 A. Theoretical analysis of the canonical distortion coefficient

In this section, we give the rigorous definitions for characterizing arbitrary diffeomorphisms between two surfaces using Riemannian geometry. Specifically, we give the precise meaning of “deformation of a unit circle into an eclipse" mentioned in the paper and explain why it is able to represent arbitrary deformations. We also connect our canonical distortion coefficient to the Beltrami-coefficient defined in quasi-conformal mapping theory. Most of our notations and definitions are adapted from the classic textbooks [9], [12] and [2].

7.1.1 Tangent space, inner product and metric tensor

In Riemannian geometry, the concept of tangent space is the vehicle for studying the intrinsic properties of the geometric shape such as isometry, conformality and diffeomorphism. Intuitively, this is because the tangent space gives us the first-order information of the surface which is linear and hence easy to handle.

Definition 4. *A **tangent vector** at $p \in \mathcal{M}$, call it $\mathbf{X}$, assigns to each coordinate patch (U_α, x_α) holding p, and n-tuple of real numbers*

$$(X_\alpha^i) = (X_\alpha^1, \ldots, X_\alpha^n) \tag{7}$$

such that if $p \in U_\alpha \cap U_\beta$, then

$$X_\beta^i = \sum_{j=1}^{n} [\frac{\partial x_\beta^i}{\partial x_\alpha^j}(p)] X_\alpha^j \tag{8}$$

Here the matrix $(\frac{\partial x_\beta^i}{\partial x_\alpha^j})$ is the Jacobian matrix between the two charts U_α and U_β, as defined in Eq. 8. Hence a tangent vector can be represented by different coordinates under different parametrizations as long as Eq. 8 is satisfied. Also at each point, all the tangent vectors form a linear space, which can be represented by a coordinate basis under any parametrization U_α.

Definition 5. *The **tangent space** to $\mathcal{M}$ at the point $p \in \mathcal{M}$ is the real vector space consisting of all tangent vectors to $\mathcal{M}$ at p. If $x_\alpha = \{x^1, \ldots, x^n\}$ is a coordinate system holding p, then the n vectors*

$$\frac{\partial}{\partial x^1}, \ldots, \frac{\partial}{\partial x^n} \tag{9}$$

*form a basis of this n-dimensional vector space and the basis is called a **coordinate basis**. We denote by $T_p(\mathcal{M})$ as the tangent space at a point $p \in \mathcal{M}$.*

Both the tangent vector and the tangent space are defined at a single point on the surface. If we consider the "bundle" of tangent vectors defined at every point, we can define a vector **field** on an open set U to be the differentiable assignment of a vector $\mathbf{X}$ to each point of U; in terms of local coordinates

$$\mathbf{X} = \sum_{j=1}^{n} X^j(x) \frac{\partial}{\partial x^j}. \tag{10}$$

Given the vector field defined on each point $p \in \mathcal{M}$, we can define a basis in the tangent space $\{e_1(p), \ldots, e_n(p)\}$. Hence we can define the **inner product** $\langle \cdot, \cdot \rangle$ that measures the distance between two tangent vectors at each point.

$$\langle \mathbf{v}, \mathbf{w} \rangle_p = \langle \sum_i \mathbf{e}_i v^i, \sum_j \mathbf{e}_j w^j \rangle_p = \sum_i \sum_j v^i \langle \mathbf{e}_i, \mathbf{e}_j \rangle_p w^j \tag{11}$$

If we define the matrix $G = (g_{ij})_p$ with entries

$$g_{ij} := \langle \mathbf{e}_i, \mathbf{e}_j \rangle_p \tag{12}$$

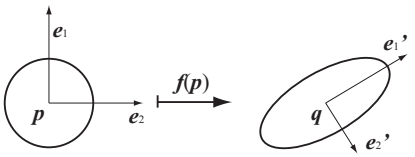


Figure 10: The deformation of a unit circle into an eclipse is defined in the tangent space.

then $\langle \mathbf{v}, \mathbf{w} \rangle = vGw$. The matrix (g_{ij}) is called the **metric tensor**. Note that here G is a function defined at each point on the surface.

A Riemannian metric tensor (g_{ij}) is required to be symmetric positive definite at every point. Hence it is always possible to apply a linear transformation to (g_{ij}) at a given point p such that the new metric tensor $(g'_{ij}) = J_p^T(g_{ij})J_p$ is the identity matrix I, where J_p is an non-degenerate matrix. Therefore, given any coordinate chart $\{U_\alpha, x_\alpha\}$, we can define a new coordinate chart $\{U'_\alpha, x'_\alpha\}$ where $x'_\alpha = J_p x_\alpha$. It can be easily verified that at point p

$$\langle e'_i, e'_j \rangle_p = \langle J_p e_i, J_p e_j \rangle_p = e_i^T J_p^T (g_{ij}) J_p e_j = \delta_i^j. \tag{13}$$

Consequently, the new coordinate chart is called the *canonical parametrization* as defined in Definition 1.

7.1.2 Diffeomorphisms, isometries and conformal maps

Given a diffeomorphism $f : \mathcal{M} \to \mathcal{N}$, the mapping between two tangent spaces $df : T_p(\mathcal{M}) \to T_q(\mathcal{N})$, where $q = f(p)$, is a linear mapping [9]. The mapping f can be fully characterized by its differential df at each point on the surface given an initial correspondence [11]. The advantage of studying the mapping f from its first-order derivative is its simplicity since df is defined on a linear space. Within this setting, the change of the tensor metric, *i.e.*,

$$\langle \mathbf{v}, \mathbf{w} \rangle \to \langle df(\mathbf{v}), df(\mathbf{w}) \rangle_q \tag{14}$$

is determined by the linear transformation df. Hence, under this the linear transformation, a unit circle in $T_p(\mathcal{M})$ is mapped into an eclipse in $T_q(\mathcal{N})$. Under canonical parametrization (Def. 1), such change of metric tensor can be represented as $I \to J_{pq}^T J_{pq}$. Since $J_{pq}^T J_{pq} = O^T \text{diag}(\lambda_1, \lambda_2) O$, this deformation represents both scale change information and angle change information. Here λ_1 and λ_2 represents the scale change along the two axes and the (rotational) orthogonal matrix O represents the angle change. Equivalently, such changes can be fully characterized by the distortion of a unit circle in the tangent space at p into an eclipse in the tangent space at q (Fig. 10). In the following, we show that the canonical distortion coefficient can fully characterize two deformation models that is prevalent in computer vision, *i.e.*, isometry and conformality.

Definition 6. *A diffeomorphism $f : \mathcal{M} \to \mathcal{N}$ is an **isometry** if for all $p \in \mathcal{M}$ and all pairs of tangent vectors $\mathbf{v}, \mathbf{w} \in T_p(\mathcal{M})$, we have*

$$\langle \mathbf{v}, \mathbf{w} \rangle_p = \langle df(\mathbf{v}), df(\mathbf{w}) \rangle_q \tag{15}$$

Hence, under canonical parametrization, the mapping must satisfy

$$J_{pq}^T J_{pq} = I, \tag{16}$$

i.e., $\lambda_1 = \lambda_2 = 1$.

Definition 7. *A diffeomorphism $f : \mathcal{M} \to \mathcal{N}$ is called a **conformal map** if for all $p \in \mathcal{M}$ and all pairs of tangent vectors $\mathbf{v}, \mathbf{w} \in T_P(\mathcal{M})$, we have*

$$\langle df(\mathbf{v}), df(\mathbf{w})\rangle_q = \lambda^2(p)\langle \mathbf{v}, \mathbf{w}\rangle_p, \tag{17}$$

where λ^2 is a nonwhere-zero differentiable function on $\mathcal{M}$.

Again, under canonical parametrization, we have $J_{pq}^T J_{pq} = \lambda^2(p)$, implying $\lambda_1 = \lambda_2$.

7.1.3 Canonical distortion coefficient and quasi-conformal mapping

The above discussion established the link between the popular deformation models (*i.e.*, isometry and conformality) and the canonical distortion coefficient. In fact, the canonical distortion coefficient is also closely related to the more general quasi-conformal mapping [2]. The quasi-conformal mapping studies the deformation between two planes $w = f(z)$, where $z = x + iy$ and $w = u + iv$. At a given point z_0, f induces a linear mapping of the differentials

$$du = u_x dx + u_y dy \tag{18}$$
$$dv = v_x dx + v_y dy \tag{19}$$

which we can also write in the complex form:

$$f_z = \frac{1}{2}(f_x - if_y) \text{ and } f_{\overline{z}} = \frac{1}{2}(f_x + if_y) \tag{20}$$

Note that this transformation is defined on two charts U_α and U_β which is assumed to have Euclidean metric. Hence one can write in classical notation

$$du^2 + dv^2 = Edx^2 + 2Fdxdy + Gdy^2 \tag{21}$$

with

$$E = u_x^2 + v_x^2, \quad F = u_x u_y + v_x v_y, \quad G = u_y^2 + v_y^2 \tag{22}$$

Under canonical parametrizations at q ($f(z_0)$), since it is Euclidean at q, the metric tensor can be represented as $du^2 + dv^2$. Hence we can establish

$$J_{pq}^T J_{pq} = \begin{pmatrix} E & F \\ F & G \end{pmatrix}. \tag{23}$$

In the theory of quasi-conformal mapping, the beltrami-coefficient $\mu(z) = \frac{f_{\overline{z}}}{f_z}$ can be partially determined by the eigenvalues of the matrix $\begin{pmatrix} E & F \\ F & G \end{pmatrix}$, which is equivalent to the canonical distortion coefficient. To see this, we first define

$$D_f = \frac{|f_z| + |f_{\overline{z}}|}{|f_z| - |f_{\overline{z}}|}. \tag{24}$$

From

$$dw = f_z dz + f_{\overline{z}} d\overline{z}, \tag{25}$$

we have

$$(|f_z| - |f_{\overline{z}}|)|dz| \leq |dw| \leq (|f_z| + |f_{\overline{z}}|)|dz|. \tag{26}$$

Hence D_f denotes the ratio of the major to the minor axis of the mapping f at z_0, *i.e.*,

$$D_f = (\frac{\lambda_1}{\lambda_2})^{\frac{1}{2}} \tag{27}$$

Therefore we can easily obtain

$$|\mu(z)| = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1} + \sqrt{\lambda_2}}. \tag{28}$$

7.2 B. Details of high-order MRF optimization algorithm

7.2.1 Derivation of dual optimization

In linear programming (LP), a general form of primal-dual relation is the following

$$\begin{array}{lcl} \min c^T x & & \max b^T y \\ \text{s.t. } Ax = b & \Leftrightarrow & \text{s.t. } A^T y \leq c \\ x \geq 0 & & \end{array} \tag{29}$$

Here c and b are two vectors and A is a matrix.

Now let us consider the LP relaxation of the problem of Eq. 4 as derived in Sec. 4:

$$\begin{aligned} \min_{\tau} & \sum_{u \in \mathcal{V}} \sum_{i \in \mathcal{L}} \theta_{u;i} \tau_{u;i} + \sum_{(u,v,w) \in \mathcal{F}} \sum_{(i,j,k) \in \mathcal{L}^3} \theta_{uvw;ijk} \tau_{uvw;ijk} \\ \text{s.t.} & \sum_i \tau_{u;i} = 1, && \forall u \in \mathcal{V} \\ & \sum_{i,j,k} \tau_{uvw;ijk} = 1, && \forall (u,v,w) \in \mathcal{F} \\ & \sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i}, && \forall (u,v,w) \in \mathcal{F} \text{ and } i \in \mathcal{L} \\ & \tau_{u;i} \geq 0, \tau_{uvw;ijk} \geq 0, \end{aligned} \tag{30}$$

Introducing the dual variables

$$\sum_i \tau_{u;i} = 1 \qquad \rightarrow y_u \tag{31}$$

$$\sum_{i,j,k} \tau_{uvw;ijk} = 1 \qquad \rightarrow y_{uvw} \tag{32}$$

$$\sum_{j,k} \tau_{uvw;ijk} = \tau_{u;i} \qquad \rightarrow M_{uvw;u:i}, \tag{33}$$

by following the primal-dual formulate of Eq. 29, we have the dual problem of the LP problem (Eq. 30):

$$\begin{aligned}
&\max \sum_{u\in\mathcal{V}} y_u + \sum_{(u,v,w)\in\mathcal{F}} y_{uvw} && (34)\\
&\quad y_u - \sum_{(v,w),(u,v,w)\in\mathcal{F}} M_{uvw;u:i} \le \theta_{u;i}, && \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L}\\
&\quad y_{uvw} + M_{uvw;u:i} + M_{uvw;v:j} + M_{uvw;w:k} \le \theta_{\theta_{uvw;ijk}}, \\
&\qquad\qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L}
\end{aligned}$$

If we define the reparametrization

$$\begin{aligned}
&\overline{\theta}_{u;i} := \theta_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} M_{uvw;u:i}, && \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L}\\
&\overline{\theta}_{uvw;ijk} := \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \\
&\qquad\qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L},
\end{aligned}$$

we have for any $u \in \mathcal{V}$

$$y_u \le \overline{\theta}_{u;i} \qquad \forall i \in \mathcal{L} \tag{35}$$

and for all $(u,v,w) \in \mathcal{F}$

$$y_{uvw} \le \overline{\theta}_{uvw;iijk} \qquad \forall (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L} \tag{36}$$

Hence the dual optimization problem of Eq. 34 can be equivalently formulated as

$$\begin{aligned}
&\max_{M} \sum_{u} \min_{i} \overline{\theta}_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} \min_{i,j,k} \overline{\theta}_{uvw;ijk} && (37)\\
&\text{s.t. } \overline{\theta}_{u;i} = \theta_{u;i} + \sum_{(u,v,w)\in\mathcal{F}} M_{uvw;u:i}, && \forall u \in \mathcal{V} \text{ and } i \in \mathcal{L}\\
&\quad \overline{\theta}_{uvw;ijk} = \theta_{uvw;ijk} - M_{uvw;u:i} - M_{uvw;v:j} - M_{uvw;w:k}, \\
&\qquad\qquad \forall (u,v,w) \in \mathcal{F} \text{ and } (i,j,k) \in \mathcal{L}\times\mathcal{L}\times\mathcal{L}.
\end{aligned}$$

7.2.2 An efficient algorithm for finding independent face sets

Alg. 4 shows the algorithm that decomposes a third-order graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ into subsets of independent face sets defined in Sec. 4.2 in the paper. To see to complexity of this algorithm, for each iteration of step two, it take $O(|\max(\mathcal{V}, \mathcal{F})|)$ to traverse all the faces to construct a new independent face set. The total number of iterations depends on the total number of decomposed sets ($< O(|\mathcal{F}|)$). Hence the worst case complexity of this algorithm is $O(|\mathcal{V}||\mathcal{F}|)$. However in practice the number of iterations is expected to be very small.

References

[1] *Microsoft© Kinect, 2010.*

Algorithm 4: Greedy algorithm for independent face sets.

Input : A third-order graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$.
Output : Decomposition of $\mathcal{F}$ into independent face sets $\cup_i \mathcal{F}_i$.

Step One: Initialization.
  for each $f \in \mathcal{F}$ **do**
    visited$[f] \leftarrow$ false
  end for

Step Two: Find maximal independent face set among the un-visited faces.

  $count = 0$
  $i = 0$
  while count $\neq |\mathcal{F}|$ **do**
    $\mathcal{F}_i \leftarrow \varnothing$
    for each $v \in \mathcal{V}$ **do**
      $visited[v] \leftarrow$ false
    end for
    for each $f = (v_1, v_2, v_3) \in \mathcal{F}$ **do**
      if $visited[f]$, $visited[v_1]$, $visited[v_2]$ and $visited[v_3]$ are all false **then**
        $\mathcal{F}_i = \mathcal{F}_i \cup f$
        $visited[v_1] \leftarrow$ true, $visited[v_2] \leftarrow$ true, $visited[v_3] \leftarrow$ true
        $visited[f] \leftarrow$ true
        $count = count + 1$
      end if
    end for
    $i = i + 1$
  end while

[2] Lars V. Ahlfors. *Lectures on Quasiconformal Mappings*. American Mathematical Society, 2 edition, 2006.

[3] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992.

[4] Susanne C. Brenner and Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2007.

[5] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.*, 28:1812–1836, 2006.

[6] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences*, 103:1168–1172, 2006.

[7] Benedict J. Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Trans. Graph.*, 26(3), 2007.

[8] Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. *ACM Trans. Graph.*, 29, July 2010.

[9] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[10] Manfredo P. do Carmo. *Riemannian Geometry*. Birkhäuser, 1992.

[11] Lawrence C. Evans. *Lectures on Quasiconformal Mappings*. American Mathematical Society, 2 edition, 1998.

[12] Theodore Frankel. *The Geometry of Physics–An Introduction*. Cambridge, 2 edition, 2004.

[13] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *TPAMI*, 6(6):721–741, 1984.

[14] Dirk Hahnel, Sebastian Thrun, and Wolfram Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI'03*, pages 915–920, 2003.

[15] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[16] A. Keane. CUDA (compute unified device architecture). 2006.

[17] Nikos Komodakis and Nikos Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009.

[18] Yaron Lipman and Ingrid Daubechies. Surface comparison with mass transportation. Technical report, Princeton University, 2010.

[19] Yaron Lipman and Thomas Funkhouser. Möbius voting for surface correspondence. *ACM Trans. Graph.*, 28(3):1–12, 2009.

[20] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[21] Konrad Polthier. Computational aspects of discrete minimal surfaces. *Global Theory of Minimal Surfaces*, 2005.

[22] Mathieu Salzmann, Julien Pilet, Slobodan Ilic, and Pascal Fua. Surface deformation models for nonrigid 3D shape recovery. *TPAMI*, 29:1481–1487, 2007.

[23] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *SIGGRAPH '01*, pages 409–416, 2001.

[24] A. Shaji, A. Varol, L. Torresani, and P. Fua. Simultaneous point matching and 3D deformable surface reconstruction. In *CVPR*, 2010.

[25] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *SGP*, pages 109–116, 2007.

[26] Kenneth Stephenson. *Introduction to Circle Packing: The Theory of Discrete Analytic Functions*. Cambridge University Press, 2005.

[27] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26, July 2007.

[28] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV08*.

[29] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[30] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24:426–433, 2005.

[31] Sen Wang, Yang Wang, Miao Jin, Xianfeng David Gu, and Dimitris Samaras. Conformal geometry and its applications on 3D shape matching, recognition, and stitching. *TPAMI*, 29(7):1209–1220, 2007.

[32] Y. Wang, M. Gupta, S. Zhang, S. Wang, X.F. Gu, D. Samaras, and P.S. Huang. High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps. In *ICCV*, 2005.

[33] Tomas Werner. Revisiting the linear programming relaxation approach to gibbs energy minimization and weighted constraint satisfaction. *TPAMI*, 32:1474–1488, 2010.

[34] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A high-resolution 3d dynamic facial expression database. In *FGR*, 2008.

[35] Wei Zeng, Dimitris Samaras, and Xianfeng David Gu. Ricci flow for 3D shape analysis. *TPAMI*, 32:662–677, 2010.

[36] Wei Zeng, Yun Zeng, Yang Wang, Xiaotian Yin, Xianfeng Gu, and Dimitris Samaras. 3D non-rigid surface matching and registration based on holomorphic differentials. In *ECCV*, 2008.

[37] Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, 2010.

[38] Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios. Intrinsic dense 3D surface tracking. In *CVPR*, 2011.

[39] Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Trans. Graph.*, 23(3):548–558, 2004.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399