Intrinsic Dense 3D Surface Tracking

Yun Zeng[1], Chaohui Wang[2,3], Yang Wang[1], Xianfeng Gu[1], Dimitris Samaras[1], Nikos Paragios[2,3]
[1]Computer Science Department, Stony Brook University, New York, USA
[2]Laboratoire MAS, Ecole Centrale Paris, Châtenay-Malabry, France
[3]Equipe GALEN, INRIA Saclay - Île-de-France, Orsay, France

Abstract

This paper presents a novel intrinsic 3D surface distance and its use in a complete probabilistic tracking framework for dynamic 3D data. Registering two frames of a deforming 3D shape relies on accurate correspondences between all points across the two frames. In the general case such correspondence search is computationally intractable. Common prior assumptions on the nature of the deformation such as near-rigidity, isometry or learning from a training set, reduce the search space but often at the price of loss of accuracy when it comes to deformations not in the prior assumptions. If we consider the set of all possible 3D surface matchings defined by specifying triplets of correspondences in the uniformization domain, then we introduce a new matching cost between two 3D surfaces. The lowest feature differences across this set of matchings that cause two points to correspond, become the matching cost of that particular correspondence. We show that for surface tracking applications, the matching cost can be efficiently computed in the uniformization domain. This matching cost is then combined with regularization terms that enforce spatial and temporal motion consistencies, into a maximum a posteriori (MAP) problem which we approximate using a Markov Random Field (MRF). Compared to previous 3D surface tracking approaches that either assume isometric deformations or consistent features, our method achieves dense, accurate tracking results, which we demonstrate through a series of dense, anisometric 3D surface tracking experiments.

1. Introduction

Dynamic 3D data has become increasingly popular with the advances in 3D reconstruction techniques [1, 12, 20, 26, 33]. An important prerequisite for most applications is to register the 3D data among frames. For applications such as facial expression analysis/transfer [7, 23], dense and accurate registration is highly desired for capturing subtle details. However, achieving dense, accurate registration remains challenging when there is noise, large deformations and lack of reliable features. In this paper, we address the challenging problem of tracking a deformable template from dynamic, markerless 3D data.

According to the well-known Riemann uniformization theorem [8], any simply-connected surface with a Riemannian metric can be conformally deformed onto one of three canonical spaces: the sphere, the plane and the hyperbolic disk. By using uniformization, 3D geometric problems are naturally converted to 2D ones, which in general simplifies computation. Most importantly, when the deformations between surfaces are isometric, matching between two surfaces can be greatly simplified in the uniformization domain by only searching for a few correspondences [15].

Previously work [25, 26, 29, 30] relied on consistent feature boundaries or points to determine prescribed sparse correspondences, in order to match two surfaces in the uniformization domain. Once the sparse correspondences are established, a single conformal energy is minimized to match between the whole surfaces. To avoid relying on consistent feature points, Lipman *et al.* [16] observed that when two surfaces are isometrically deformed, only three correspondences are needed to determine a unique conformal mapping which is described by a Möbius transformation. Sparse correspondences are optimized by voting from different Möbius transformations induced by different combinations of correspondences. Recently, based on the fact that every three correspondences determine a unique conformal mapping, leading to a candidate matching point for every point on the surface, Zeng *et al.* [31] formulated a high-order graph matching problem to search for the optimal dense matching result by combining multiple matching criteria. Despite its success in combining multiple matching criteria to handle more than isometric deformations, the singleton term that defines textural and geometric similarities for each matching candidate is evaluated only pointwise and hence such a term is sensitive to noise. Most recently, Lipman *et al.* [15] proposed a new distance function that compares two neighborhoods (*i.e.*, disks) around two points in the disk uniformization domain, which improves the robustness of the matching cost for each candidate correspondence. Nevertheless, this distance cannot handle general surface matching when the surfaces have inconsistent boundaries or are anisometrically deformed, because comparing two neighborhoods directly in the uniformization domain is not straightforward since a disk is no longer mapped to a disk under Möbius transformations [18].

In this paper, we define a new distance that compares the neighborhoods between any candidate matching pair even when the two surfaces have inconsistent boundaries and are not isometrically deformed. Since every three correspondences determine a unique conformal mapping between the two surfaces, we can define a matching cost based on feature differences for every such possible match. However, globally searching for the best three correspondences that match the two surfaces is limited because the two surfaces can be exactly matched only when they are isometrically deformed and have consistent boundaries. Hence, we define a cost function for a particular correspondence by the lowest feature differences across the set of transformations that cause the two points to match, which only involves searching for the correspondences of another two points on the surface. By restricting the comparison of feature differences only between the neighborhoods of the correspondence, we can handle surfaces with inconsistent boundaries or anisometric deformations. A matching cost between two neighborhoods can be therefore efficiently computed since only one conformal mapping is needed for one surface and the other conformal mappings induced from different correspondence matches are computed in a closed form.

With the above mentioned matching cost for any candidate correspondence, it is not enough to simply output such locally best match for each point due to multiple optima, noisy data and numerical errors. Therefore, regularization is necessary for a plausible result. In this paper, we formulate the surface tracking problem in a unified probabilistic inference framework that takes into account spatio-temporal consistency as well as the possibility for drift error. We show that such an inference problem can be approximated by standard MRF optimization methods in a discrete setting and occlusion can be appropriately handled. Combinatorial methods based on graphical models have become popular due to their capabilities of solving for more complicated deformations [10, 11, 21, 24] and avoiding local optimal solutions [13, 14]. Besides, occlusion handling can be conveniently modeled in the same framework [22].

In summary, the primary contributions of this paper are a robust intrinsic distance function for measuring the cost of matching two points and a unified framework for intrinsic 3D surface tracking. To achieve a robust 3D tracking system, our framework includes an intrinsic spatial deformation prior that constrains consistency in local deformations among neighboring points as well as drift and occlusion handling. Our tracking method is computed in the uniformization domain, so it is robust to large deformations and scale changes. Compared to existing tracking algorithms such as [3, 26], we do not require prescribed feature detectors and do not rely on consistent boundaries. Therefore our method is able to handle surface tracking under challenging situations as shown in our experiments with a deforming sponge. Unlike the system of Weise *et al.* [28], we do not require a pre-defined training set for PCA learning, which is important for accurately tracking both large previously unseen variations in object deformation as well as subtle but significant differences in the case of facial expression changes. Quantitative results show that our algorithm achieves a high level of accuracy.

The remainder of this paper is organized as follows. The new distance of matching correspondences is defined in Sec. 2. In Sec. 3 we introduce our probabilistic 3D surface tracking framework. The implementation details are given in Sec. 4. Experimental results and validation are part of Sec. 5. Finally we conclude our work in Sec. 6.

2. A robust correspondence matching distance

We assume a shape is represented in a metric feature space $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$, where $\mathcal{M}$ is a compact connected and complete Riemannnian surface, $d_{\mathcal{M}} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a measure of distances between pairs of points on $\mathcal{M}$, and $f_{\mathcal{M}} : \mathcal{M} \to \mathbb{R}^n$ is the mapping of each point on $\mathcal{M}$ into the feature space (such as curvatures, texture, *etc*.). Previously, a number of metrics have been proposed for measuring the similarities between any two shapes based on geometric information only, *e.g*., the geodesics distance [4], the diffusion distance [5] and distances based on the conformal factor [15, 30]. To compare two shapes $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$ and $(\mathcal{N}, d_{\mathcal{N}}, f_{\mathcal{N}})$, we denote the set of possible mappings (*e.g*., diffeomorphism) between them as $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$. A distance between any two shapes $(\mathcal{M}, d_{\mathcal{M}}, f_{\mathcal{M}})$ and $(\mathcal{N}, d_{\mathcal{N}}, f_{\mathcal{N}})$ can be defined as follows:

$$d^{\mathcal{T}}(\mathcal{M},\mathcal{N}) = \inf_{t\in\mathcal{T}_{\mathcal{M}\to\mathcal{N}}} \int_{\mathcal{M}} |f_{\mathcal{M}}(x) - f_{\mathcal{N}}(t(x))|dx. \quad (1)$$

Such a definition resembles the sum of absolute differences (SAD) metric commonly used in motion estimation. Furthermore, in the context of surface registration, it provides a flexible way of handling a wide range of deformations between surfaces. For example, when the feature is the conformal factor [30] and the mappings $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ are restricted to Möbius transformations in the uniformization domain, it handles isometric or near-isometric surface matching. To deal with more general deformations, we may use other features such as texture and curvature [31].

In this paper, based on Eq. 1, we consider the following distance function for measuring the quality of a correspondence between a point $p \in \mathcal{M}$ and a point $q \in \mathcal{N}$:

$$d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p,q) = \inf_{\substack{t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}} \\ t(p)=q}} \int_{\mathcal{M}} |f_{\mathcal{M}}(x) - f_{\mathcal{N}}(t(x))| dx, \quad (2)$$

which is defined by the cost of matching the two surfaces by fixing the particular correspondence. When there is no mapping in the group $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ that maps p to q, we define the distance to be infinite. Since the distance function of Eq. 2 is defined on the whole surface, when the transformation group $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ is confined to mappings with bounded area distortions, a small deviation from the true matching that minimizes Eq. 1 would cause the energy measure to deviate significantly from the optimum, which guarantees the robustness of this distance measure. Hence such a distance is more robust compared to distances based on local features. It is easy to see that $d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p,q) \geq d^{\mathcal{T}}(\mathcal{M},\mathcal{N})$ for any $p \in \mathcal{M}, q \in \mathcal{N}$ and the lower bound is achieved when p is mapped to q under the transformation that minimizes the energy of Eq. 1. In the problem of dense surface registration, the goal is to find on $\mathcal{N}$ the correspondences of a point set $P = \{p_i | p_i \in \mathcal{M}, i = 1, \ldots, n\}$. Since we have

$$d^{\mathcal{T}}(\mathcal{M},\mathcal{N}) \leq \frac{\sum_{p \in P} d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p, t(p))}{|P|}, \forall t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}}, \quad (3)$$

the problem of shape registration can be formulated as

$$\arg\inf_{t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}}} \sum_{p \in P} d^{\mathcal{T}}_{\mathcal{M},\mathcal{N}}(p, t(p)). \quad (4)$$

In the following, we show how the distance function can be efficiently approximated in the uniformization domain.

2.1. Approximation in the uniformization domain

Although the distance defined in Eq. 2 gives us a robust way to evaluate the matching cost between points, it is in general computationally intractable to evaluate such a distance function directly in the 3D embedding space since it involves searching among all possible matchings between two surfaces given a correspondence. For example, in [32] the matching cost given a few sparse correspondences is measured by deforming the whole surface to the target based on a particular deformation energy; minimizing such an energy is computational possible for only approximately 10 correspondences. In this paper, we propose an efficient way to approximate the distance function of Eq. 2 by considering a subset of the mapping set $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ (note that such an approximation is only used for evaluating the correspondence cost and the global shape registration problem of Eq. 4 is considered in a general set of mappings).

In order to take into account mappings between two surfaces with inconsistent boundaries, we consider a neighborhood $N(p)$ of p and the points on its boundary $\partial N(p) = \{p_1, \ldots, p_r\}$. For each possible mapping of the neighboring points $p_1, \ldots, p_r \in \partial N(p)$, the distance function of Eq. 2 can be approximated by warping the neighborhood $N(p)$ to the neighborhood of the target. Directly computing such warping is very costly. However, if we notice that a mapping between the two surfaces can be computed by specifying a few feature correspondences and optimizing a conformal energy to find the mappings of all the other points [25, 26, 30], we then have an efficient way to approximate the distance function of Eq. 2. This motivates us to consider the mappings of the neighborhood $N(p)$ in the uniformization domain $\mathcal{U} \subset \mathbb{C}$, where $\mathbb{C}$ denotes the complex domain.

Formally, we denote the uniformization (conformal mapping) of any surface $\mathcal{M}$ as $\phi_{\mathcal{M}} : \mathcal{M} \to \mathcal{U}$. Also we consider the set of mappings $\mathcal{T}^{\text{UNI}}$ that is induced by specifying three correspondences between two surfaces in the uniformization domain [16]. For any point $p \in \mathcal{M}$, we define the image of a point p as $\text{Img}(p) = \{t(p) | t \in \mathcal{T}_{\mathcal{M}\to\mathcal{N}}\}$, where $\mathcal{T}_{\mathcal{M}\to\mathcal{N}}$ can be arbitrary diffeomorphisms and we only require that $\mathcal{T}^{\text{UNI}} \subset \mathcal{T}_{\mathcal{M}\to\mathcal{N}}$. For any two points $p_1, p_2 \in \partial N(p)$, $q \in \text{Img}(p)$, $q_1 \in \text{Img}(p_1)$ and $q_2 \in \text{Img}(p_2)$, let us denote by $\text{Mo} : pp_1p_2 \to qq_1q_2$ the Möbius transformation that maps (p, p_1, p_2) to (q, q_1, q_2) in $\mathcal{U}$. We then approximate the distance of Eq. 2 in the uniformization domain as follows:

$$d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(p,q) = \inf_{\substack{p_1,p_2 \in \partial N(p), q_1 \in \text{Img}(p_1), q_2 \in \text{Img}(p_2), \\ \text{Mo}: pp_1p_2 \to qq_1q_2}} \frac{\int_{\phi_{\mathcal{M}}(N(p)) \subset \mathcal{U}} |f_{\mathcal{M}}(\phi^{-1}_{\mathcal{M}}(z)) - f_{\mathcal{N}}(\phi^{-1}_{\mathcal{N}}(\text{Mo}(z)))| dz}{Area(N(p))}. \quad (5)$$

Here $\phi_{\mathcal{M}}(N(p))$ denotes the mapping of the neighborhood $N(p)$ to the uniformization domain and $Area(N(p))$ denotes the area of the neighborhood. When the feature is only based on geometry (*e.g.*, the conformal factor [15] or the gaussian curvature), the distance measures the deviation from isometric deformation. However, when the feature is based on other measures such as texture, the distance measures how accurately the source surface has been deformed onto the target surface, not necessarily by an isometric deformation. Therefore, our definition is general and subsumes the isometric deformation as a special case.

3. Probabilistic 3D surface tracking

The intrinsic distance $d^{\text{UNI}}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ measures the likelihood of matching between individual points. Such a data term is then combined with regularization terms towards solving the problem of surface registration. To this end, we investigate the 3D surface tracking problem in a probabilistic framework which takes into account geometric and textural similarities, spatio-temporal consistencies as well as error drift.

Let us denote by $\mathcal{M}^{1:t} \equiv \{\mathcal{M}^1, \dots, \mathcal{M}^t\}$ the dynamic 3D data up to time t, and $\mathbf{x}^{1:t} \equiv \{\mathbf{x}^i \subset \mathcal{M}^i | i = 1, \dots, t\}$ as the trajectory of the given initial dense points $\mathbf{x}^0 = \{x_i^0 \in \mathbb{R}^3 | i = 1, \dots, n\}$ where $\mathbf{x}^t = \{x_i^t \in \mathbb{R}^3 | i = 1, \dots, n\}$. In order to utilize the intrinsic measure defined in the previous section, we assume the initial points $\mathbf{x}^0$ are represented as a 2-manifold mesh, *i.e.*, a planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The task of tracking the trajectory of $\mathbf{x}^0$ at time t given the dynamic data $\mathcal{M}^{1:t}$ and the previous trajectory $\mathbf{x}^{1:t-1}$ therefore becomes the MAP problem

$$\arg\max_{\mathbf{x}^t} p(\mathbf{x}^t | \mathcal{M}^{1:t}, \mathbf{x}^{1:t-1}). \tag{6}$$

Here we assume $\mathcal{M}^{1:t'}$ to be independent of $\mathbf{x}^t$ given $\mathbf{x}^{1:t'}$ whenever $t' < t$. From the Bayes' theorem, we have

$$\begin{aligned} & p(\mathbf{x}^t | \mathcal{M}^{1:t}, \mathbf{x}^{1:t-1}) \\ &= \frac{p(\mathcal{M}^{1:t} | \mathbf{x}^{1:t}) p(\mathbf{x}^t | \mathbf{x}^{1:t-1})}{p(\mathcal{M}^{1:t} | \mathbf{x}^{1:t-1})} \\ &\propto p(\mathcal{M}^{1:t} | \mathbf{x}^{1:t}) p(\mathbf{x}^t | \mathbf{x}^{1:t-1}) \\ &= p(\mathcal{M}^t | \mathbf{x}^{1:t}, \mathcal{M}^{1:t-1}) p(\mathcal{M}^{1:t-1} | \mathbf{x}^{1:t}) p(\mathbf{x}^t | \mathbf{x}^{1:t-1}) \\ &\propto \underbrace{p(\mathcal{M}^t | \mathbf{x}^{1:t}, \mathcal{M}^{1:t-1})}_{\text{Data fidelity}} \underbrace{p(\mathbf{x}^t | \mathbf{x}^{1:t-1})}_{\text{Spatio-temporal prior}} . \end{aligned} \tag{7}$$

Here $p(\mathcal{M}^t | \mathbf{x}^{1:t}, \mathcal{M}^{1:t-1})$ denotes the data likelihood defined by intrinsic similarities. $p(\mathbf{x}^t | \mathbf{x}^{1:t-1})$ denotes the spatio-temporal priors that ensure the smoothness of the result. In the following we discuss each of the two terms in detail.

3.1. Data fidelity terms

The data fidelity terms consider the fidelity of the $3D$ data $\mathcal{M}^{1:t}$ given the tracking results $\mathbf{x}^{1:t}$. For dense tracking, we assume the tracking points $\mathbf{x}^0$ are dense enough to capture the detailed geometry of the surfaces and thus the trajectory $\mathbf{x}^{1:t'}$ and the data $\mathcal{M}^{1:t'}$ are independent of the trajectory $\mathbf{x}^t$ given $\mathbf{x}^{t-1}$ and $\mathcal{M}^{t-1}$ when $t' < t - 1$. We have

$$\begin{aligned} & p(\mathcal{M}^t | \mathbf{x}^{1:t}, \mathcal{M}^{1:t-1}) \\ &= \frac{p(\mathcal{M}^t, \mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})}{p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})} \\ &\propto p(\mathcal{M}^t, \mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}) \\ &= p(\mathcal{M}^t | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}) \times \\ & \quad p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, \mathcal{M}^t). \end{aligned}$$

Here $p(M^t | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1})$ denotes the registration between successive frames and $p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, \mathcal{M}^t)$ denotes the consistency between the current frame and the previously tracked frames, which avoids loss of tracking caused by accumulation of local registration errors.

3.1.1 Geometry and texture similarities

Intrinsic comparison takes into account both geometry and texture (if available) consistencies between frames. We define the inter-frame data similarity term as follows:

$$\begin{aligned} & p(\mathcal{M}^t | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}) \\ & \propto \prod_{i=1}^{n} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t, \mathcal{M}^{t-1}}(x_i^t, x_i^{t-1}) | 0, \sigma_{data}). \end{aligned} \tag{8}$$

3.1.2 Error drift term

The intrinsic distance $d^{\text{UNI}}_{\mathcal{M}^t, \mathcal{M}^{t'}}(x_i^t, x_i^{t'})$ allows us to consider the likelihood of a correspondence between time t and t'. We assume that $\mathbf{x}^t$ is consistent with the previously tracked frames if it "agrees" with the majority of them. For each point, we select the median distance of the set of the point to its previous matches. This is similar to the use of median filter, and formulated as

$$\begin{aligned} & p(\mathbf{x}^{1:t-2}, \mathcal{M}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}, \mathcal{M}^{t-1}, \mathcal{M}^t) \\ & \propto \prod_{i=1}^{n} \text{median}_{t' \in \{1, \dots, t-2\}} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t, \mathcal{M}^{t'}}(x_i^t, x_i^{t'}) | 0, \sigma_{drift}) \end{aligned} \tag{9}$$

However, computing consistency between the current frame and all the previous frames is very costly. An approximate sampling scheme is to consider a subset of $\{1, \dots, t-2\}$, namely, $\mathcal{I}$. Hence Eq. 9 can be approximated as follows:

$$\prod_{i=1}^{n} \text{median}_{t' \in \mathcal{I}} \mathcal{N}(d^{\text{UNI}}_{\mathcal{M}^t, \mathcal{M}^{t'}}(x_i^t, x_i^{t'}) | 0, \sigma_{drift}). \tag{10}$$

3.2. Spatio-temporal priors

The probability $p(\mathbf{x}^t | \mathbf{x}^{1:t-1})$ represents the prior knowledge of the trajectory $\mathbf{x}^{1:t}$, and also regularizes the tracking result. We decompose the probability into two terms:

$$\begin{aligned} p(\mathbf{x}^t | \mathbf{x}^{1:t-1}) &= \frac{p(\mathbf{x}^t | \mathbf{x}^{t-1}) p(\mathbf{x}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1})}{p(\mathbf{x}^{1:t-2} | \mathbf{x}^{t-1})} \\ &\propto p(\mathbf{x}^t | \mathbf{x}^{t-1}) p(\mathbf{x}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}). \end{aligned} \tag{11}$$

Here $p(\mathbf{x}^t | \mathbf{x}^{t-1})$ denotes the spatial deformation consistency between consecutive frames and $p(\mathbf{x}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1})$ denotes the dynamic motion consistency.

3.2.1 Intrinsic spatial deformation prior

The spatial prior $p(\mathbf{x}^t | \mathbf{x}^{t-1})$ takes into account the plausible deformation of the surface between frames. In the definition of the distance in Eq. 5, only the matching costs for each correspondence $x_i^{t-1} \mapsto x_i^t$ are considered. There is no constraint on the consistency between two neighboring correspondences $x_i^{t-1} \mapsto x_i^t$ and $x_j^{t-1} \mapsto x_j^t$ where

$(i, j) \in \mathcal{E}$. Since each of the distance functions (Eq. 5) takes into account the locally best Möbius transformation mapping a neighborhood of x_i^{t-1} to a neighborhood of x_i^t, it is reasonable to assume that such a locally optimal transformation also maps x_i^{t-1}'s neighbor x_j^{t-1} to a position nearby x_j^t. Let $\mathrm{Mo}_{p,q}^{opt}$ denote the optimal Möbius transformation that results in the distance defined in Eq. 5. To constrain the deformation consistency between neighboring points $(i, j) \in \mathcal{E}$, we define the following distance in the uniformization domain:

$$d_{i \to j}^t = |(\mathrm{Mo}_{x_i^{t-1}, x_i^t}^{opt}(\phi^{t-1}(x_j^{t-1}))) - \phi^t(x_j^t)|. \qquad (12)$$

Here $\phi^t(\cdot)$ denotes the uniformization of the data $\mathcal{M}^t$. This distance measures how close $\phi^t(x_j^t)$ is to the point z, where z is obtained by transforming $\phi^{t-1}(x_j^{t-1})$ using the optimal transformation $\mathrm{Mo}_{x_i^{t-1}, x_i^t}^{opt}(\cdot)$ that maps x_i^{t-1} to x_i^t. When this distance is small, it means such optimal transformation produces consistent mappings with the neighbors. Formally, we define,

$$p(\mathbf{x}^t | \mathbf{x}^{t-1}) \propto \prod_{(i,j) \in \mathcal{E}} \mathcal{N}((d_{i \to j}^t + d_{j \to i}^t)/2 | 0, \sigma_{spa}). \qquad (13)$$

3.2.2 Dynamic motion prior

The dynamic prior imposes temporal consistency of each vertex i by assuming the curve traced by each vertex i to be smooth, *i.e.*, we assume the acceleration to be small. If we define the angle between the vectors $x_i^t - x_i^{t-1}$ and $x_i^{t-1} - x_i^{t-2}$ as Ang_i^t, the dynamic prior can be defined as

$$p(\mathbf{x}^{1:t-2} | \mathbf{x}^t, \mathbf{x}^{t-1}) \propto \prod_{i=1}^{n} \mathcal{N}(\mathrm{Ang}_i^{t-1} | \mathrm{Ang}_i^t, \sigma_{dyn}). \qquad (14)$$

In practice, the smoothness assumption is only applicable when the motion between two frames is not too large.

4. Implementation details

The above mentioned objective function (Eq. 7) does not have a closed form in continuous space, making global optimization difficult. Instead, in this paper, we employ the discrete MRF framework [9] to take into account the objectives discussed in the previous section. For each frame at time t, we select L matching candidates for each point $x_i^t, i = 1, \ldots, n$. As a result, by applying the $-\log$ operator to the probability of Eq. 7, our tracking problem is equivalent to solving for the optimal configuration $\mathbf{x}^t \in L^n$:

$$\arg\min_{\mathbf{x}^t} \sum_{i \in \mathcal{V}} \theta_i(x_i^t) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i^t, x_j^t), \qquad (15)$$

where the energy functions $\theta_i(x_i^t)$, $\theta_{ij}(x_i^t, x_j^t)$ are defined according to the probabilistic framework discussed in Sec. 3. In this section, we give the detailed definition of the energy.

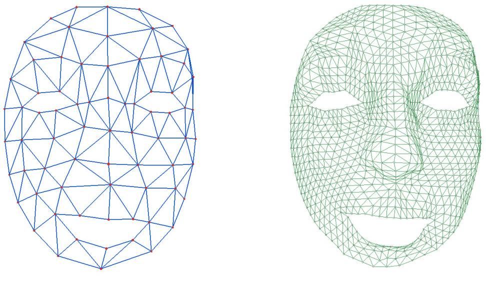

(a) Base mesh (b) Dense mesh

Figure 1. Mesh template. The base mesh (a) is resampled to produce the dense mesh (b).

4.1. Initialization

In the first frame $\mathcal{M}^0$, an initial mesh template $\mathbf{x}^0 = \{x_1^0, \ldots, x_n^0\}$ is constructed. The template can be constructed either automatically [28] or manually [3]. In our experiments, we constructed the template using the retopology tool provided in MeshLab[1]. In this way, we specify less than 100 base vertices and obtain a dense mesh with around 1000 vertices (Fig. 1).

4.2. Candidate selection

To decide the matching candidates for each node x_i^t in the next frame $t+1$, we consider *the embedding space* neighborhood, *the view space* neighborhood and *the intrinsic space* neighborhood. (1) For the embedding space neighborhood, we uniformly sample L_1 points on $\mathcal{M}^{t+1}$ in the neighborhood of each x_i^t within radius R. (2) For the view space neighborhood, we project the point x_i^t to an image plane where x_i^t is visible. Then we back-project L_2 neighboring points of the the projection of x_i^t on the image plane back to the surface $\mathcal{M}^{t+1}$. For 3D data obtained from the depth map of 2D images, the selection of the neighbors on the image plane is done in a hierarchical manner in order to take into account large deformations as in [11]. (3) For the intrinsic space neighborhood, we randomly select L_3 triplets of initial correspondences (by closest point registration or results from previous iteration) among the vertices of the base mesh and obtain a correspondence for each point from every triplets of correspondences [31]. Such intrinsic space sampling can achieve sub-sample accuracy [31]. In our experiments, we consider $L = 64$ candidates for each point, $L_1 = 24$ from embedding space sampling, $L_2 = 25$ from view space sampling and $L_3 = 15$ from intrinsic space sampling. R is set to be 0.1 of the diameter of the mesh $\mathcal{M}^0$.

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

4.3. Computation of intrinsic distance

The computation of the intrinsic distance $d^{\mathrm{UNI}}_{\mathcal{M},\mathcal{N}}(p,q)$ for any correspondence $p \mapsto q$ involves selection of the neighborhood $N(p)$, sampling of points $p_1,\ldots,p_r \in \partial N(p)$ and the numerical approximation of the integration in Eq. 5 from the $N(p)$ to all its possible mappings on the target surface in the uniformization domain. We select the boundary points $p_1,\ldots,p_r$ among the vertices of the base mesh (Fig. 1(a)). If p is not on the boundary of the template and $p_a p_b p_c$ is the face of the base mesh that covers p, we choose $N(p)$ to be the largest triangle among the three triangles $\triangle p p_a p_b$, $\triangle p p_b p_c$ and $\triangle p p_c p_a$ and we evaluate the integration of Eq. 5 only in the region covered by the selected triangle. If p is on the boundary, we choose $N(p)$ to be the two largest triangles among its neighboring triangles mentioned above. In our implementation, we select 81 sampling points in each neighborhood. The distance function can be efficiently computed in parallel because each of the functions $d^{\mathrm{UNI}}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ is independent of the others.

4.4. Occlusion handling

Occlusions are handled by introducing an additional label $\{Occ\}$ for each vertex x_i^t, with a cost function:

$$d^{\mathrm{occ}}(x_i^t, x_i^{t'}) = \begin{cases} 0 & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) < \delta, x_i^t \neq Occ \\ E_1 & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) > \delta, x_i^t = Occ \\ E_2 & \text{otherwise} \end{cases}.$$

Here t' is the last frame before time t that the point i is visible and we set $E_1 = 1$, $E_2 = 10$ and $\delta = 0.05$. Intuitively, when the cost of matching a correspondence $x_i^{t'} \mapsto x_i^t$ is higher than a threshold δ, it is likely that i is occluded at frame t, where a constant penalty E_1 is imposed. When occlusion occurs at a point, we set its default position as the position computed by the ICP algorithm [2] registering the un-occluded points.

4.5. Composite MRFs and optimization

Combining the energy functions defined above, we optimize for the MAP problem of Eq. 6 under the discrete MRF optimization framework. The singleton terms include the data-fidelity, dynamic motion consistency, error drift and occlusion handling priors, *i.e.*, for all $i \in \mathcal{V}$, we define

$$\theta_i(x_i^t) = \begin{cases} \frac{d^{\mathrm{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t-1}}(x_i^t, x_i^{t-1})^2}{\sigma_{data}} + \frac{(\mathrm{Ang}_i^{t-1} - \mathrm{Ang}_i^t)^2}{\sigma_{dyn}} \\ + \frac{(\mathrm{median}_{t' \in \mathcal{T}} d^{\mathrm{UNI}}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}))^2}{\sigma_{drift}} \\ \quad & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) < \delta, x_i^t \neq Occ \\ E_1 & \text{if } d^{UNI}_{\mathcal{M}^t,\mathcal{M}^{t'}}(x_i^t, x_i^{t'}) > \delta, x_i^t = Occ \\ E_2 & \text{otherwise} \end{cases}$$

The pairwise terms include the spatial deformation prior in Sec. 3.2.1 and the smoothness of the occluded part, *i.e.*,

$$\theta_{ij}(x_i^t, x_j^t) = \begin{cases} \frac{(d^t_{i\to j} + d^t_{j \to i})^2}{4\sigma_{spa}} & \text{if } x_i^t, x_j^t \neq Occ \\ 0 & \text{if } x_i^t = x_j^t = Occ \\ E_3 & \text{otherwise} \end{cases} \quad (16)$$

In our experiments, $E_3 = 1$. Intuitively, such an energy encourages the smoothness of the occluded part. We employ the TRW-S algorithm [13] for the optimization. The energy is optimized iteratively until convergence or up to a maximum allowed number of iterations. For the drift handling term of Eq. 10, we randomly select 5 frames from previous tracking results $\{1,\ldots,t-1\}$. The weights of the energy are selected as $\sigma_{data} = 1$, $\sigma_{dyn} = 500$, $\sigma_{drift} = 2$, $\sigma_{spa} = 20$.

5. Results

Data: We test our tracking system on a dynamic face data set captured by the 3D scanning system described in [27]. The data set consists of four actors with 24 different facial expressions, including coy flirtation, devious smirk, soft affection and fake smile, *etc*. Each of the expressions is captured with a frame rate of $24fps$ for around $10-20$ seconds. The number of vertices for each frame is $79,000$ on average with only gray-scale textural information (the gray level is normalized as in $[0,1]$). Some of the captured raw 3D data frames suffer from scale ambiguities. In such cases, we remove the dynamic prior defined in Sec. 3.2.2. In our experiments, we use texture as the feature for computing the intrinsic distance (Eq. 5) as it is less sensitive to anisometric deformation. Fig. 2 shows four different sequences from four different actors[2]. Fig. 3 shows a tracking result in a very challenging situation (largely inconsistent boundaries, occlusions and anisometric deformations between frames). For this example, the average texture difference per point between every frame and the first frame is 0.0235. The maximal average area ratio change (to the first frame) is 1.26 and the maximal percentage of surface occlusion occurring between two frames is around 30%.

Analysis of intrinsic distance function: A key factor to achieve high accuracy of surface tracking is the distance function $d^{UNI}_{\mathcal{M},\mathcal{N}}(\cdot,\cdot)$ defined in Sec. 2. To see the sensitivity of the distance in distinguishing subtle differences in correspondences for a given point p, we sample 7×7 closest neighboring points in the next frame as matching candidates in the embedding space (Sec. 4.2). Fig. 4(a) shows the evaluation of the 49 values of the function $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ for different points on the surface. We compare the distance with simple per-point texture difference.

Furthermore, we compare our cost function with the result obtained by the optical flow algorithm in [17] based on [6]. We project the left part of the face to a 640×480 perspective view selected to maximize visibility and apply

[2] See the complete sequences: http://www.cs.sunysb.edu/∼ial/.

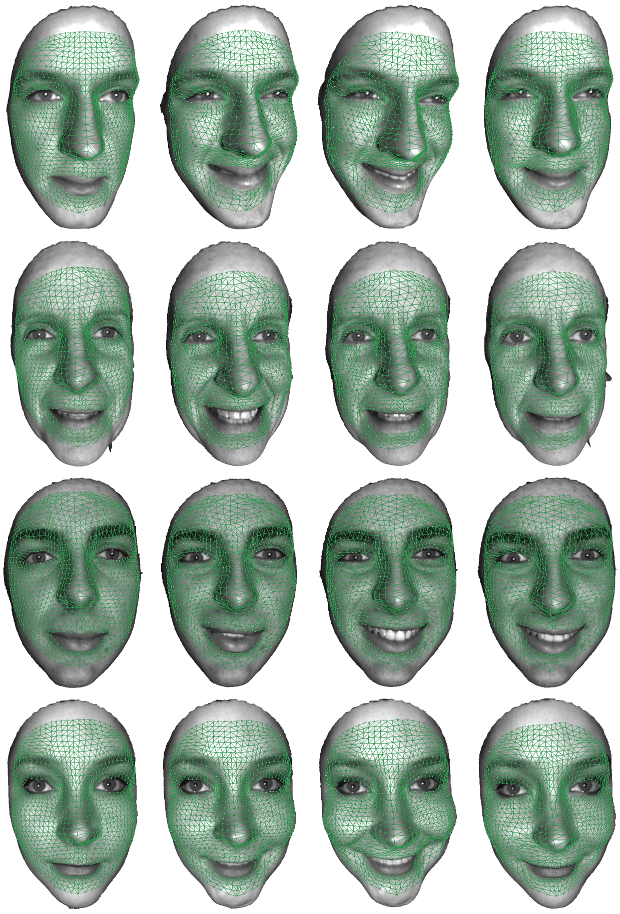


Figure 2. Tracking results selected from our data set.

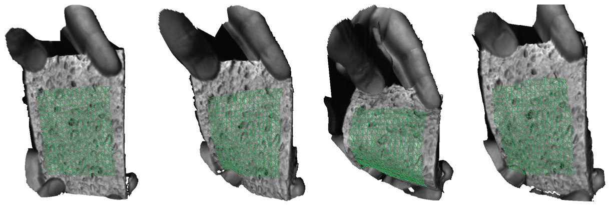
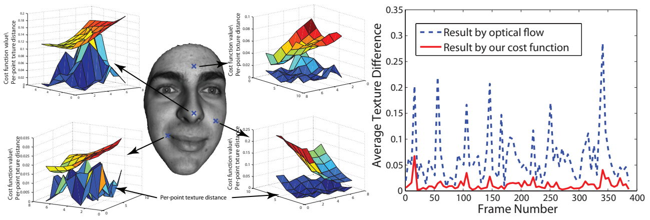


Figure 3. A challenging result with both anisometric deformation and inconsistent boundaries.



(a) $d^{UNI}_{\mathcal{M},\mathcal{N}}(p,\cdot)$ (b)

Figure 4. (a) Our cost function of Eq. 5 *vs.* per-point texture distance on distinguishing subtle differences in correspondences and (b) comparison with optical flow method for inter-frame registration (details of comparison are described in the text).

the optical flow algorithm to establish correspondences between two frames. For the template points that belong to the projected part, we compute the cost function and choose the correspondence with lowest cost as the matching result. We linearly interpolate the correspondence of other points within the template that are visible. We compare the average texture per-point differences based on the correspondences obtained by optical flow and by our method (Fig. 4 (b) shows the comparison for one sequence). It can be seen that when the deformation between two frames is large, the optical flow degrades more significantly than our method.

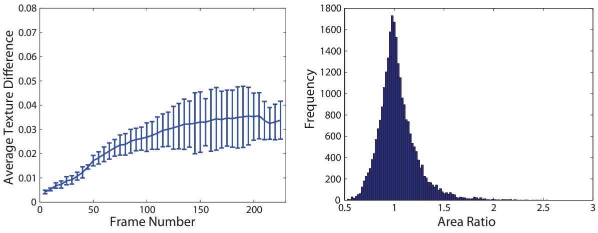
Error and performance analysis: Fig. 5(a) shows the error evaluation based on the complete 24 tracking results. The error measures the average per-point texture difference compared to the first frame. Fig. 5(b) shows the amount of area ratio change (anisometry) between the first frame the the current frame for $23,000$ randomly selected triangles among the tracking results.

We compare the influence of the regularization terms in the optimization of Eq. 15. The error is evaluated based on the average texture difference between every frame and the first frame. Fig. 6 (a) shows the comparisons for sequence A_coyfirtion. Even by considering the data term only, our method is more accurate for most frames than a previous intrinsic surface tracking method based on Harmonic maps [26]. Fig 6 (b) is the comparison for 5 more sequences on the average per-point texture difference.

Computation time: Our algorithm is implemented on an Intel® Core(TM) 2 Duo 3.16G PC with 4G RAM and a NVIDIA® Geforce 9800GTX+ graphics card with 128 CUDA cores. The preprocessing (mesh loading, nearest neighbor search data structure construction, candidate selection) takes 2–3s. The computation of the mid-edge uniformization [19] for each mesh takes less than 1s using GPU implementation. With the hardware acceleration described in Sec. 4.3, the computation of the $L = 64$ cost functions from Eq. 5 for one tracking point takes only 3ms on average. The MRF optimization using the TRW-S algorithm [13] takes around 1–3s for the 1000 template points described in Sec. 4.1 with 65 (64 for matching candidate and 1 for occlusion) labels per point. Therefore tracking one frame with 5 look-back frames takes only $18 - 25$s for each iteration. In our experiments, we observe that the algorithm often converges within 5 iterations.

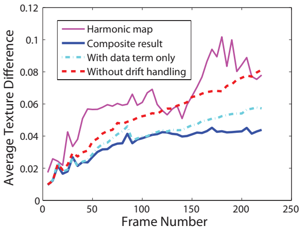
6. Conclusion

We proposed a new cost function that compares two neighborhoods of a correspondence by searching among all the possible mappings in the uniformization domain. This cost function is then combined with regularization terms that take into account spatio-temporal consistency, error drift and occlusion problems, into a unified 3D tracking framework. By employing existing MRF optimization techniques and hardware accelerations, our algorithm becomes



(a) (b)

Figure 5. (a) shows the evaluation of the average per-point texture difference between every frame and the first frame on the whole 24 data set. (b) is the frequency of the area ratio of randomly selected triangles between current frame and the first frame, which shows that a significant number of them deforms anisometrically.



Method \ Data	B_F	D_C	B_C	A_C	C_A
HM	0.0810	0.0987	0.0796	0.0983	0.0704
D.Term	0.0573	0.0423	0.0505	0.0452	0.0564
W.o. Drift	0.0364	0.0315	0.0431	0.0428	0.0396
Our Res.	0.0317	0.0262	0.0150	0.0299	0.0321

(a) Per-frame error for one sequence (b) Average error for 5 sequences

Figure 6. Results show influence of the regularization term used in the optimization of Eq. 15 and comparison with previous intrinsic tracking method used in Harmonic maps [26].

practical for applications where high accuracy is essential. In the near future, we would like to apply our algorithm to track additional dynamic 3D databases and explore applications such as facial expression analysis/transfer, *etc*.

Acknowledgements

We thank Przemyslaw Szeptycki for help with data capture and many useful discussions. This work was partially supported by grants from: NSF: CCF-TF 0830550, CCF-0916235, IIS: III 0916286, ONR N000140910228, NSF: CNS-0959979, CNS-0627645, NIH 1R01EB0075300A1.

References

[1] *Microsoft© Kinect, 2010.*

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

[3] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Trans. Graph.*, 29(4):1–10, 2010.

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

[5] M. M. Bronstein and A. M. Bronstein. Shape recognition with spectral distances. *TPAMI*, 33:1065–1071, 2011.

[6] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61:211–231, 2005.

[7] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2), 2001.

[8] H. M. Farkas and I. Kra. *Riemann Surfaces*. Springer, 2004.

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

[10] B. Glocker, H. Heibel, N. Navab, P. Kohli, and C. Rother. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *ECCV*, 2010.

[11] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through MRFs and efficient linear programming. In *Medical Image Analysis*, volume 12, pages 731–741, 2008.

[12] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *ICCV*, 2007.

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2006.

[14] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *CVIU*, 112(1):14–29, 2008.

[15] Y. Lipman and I. Daubechies. Surface comparison with mass transportation. Technical report, Princeton University, 2010.

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

[17] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[18] T. Needham. *Visual Complex Analysis*. Oxford University Press, 1999.

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

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

[21] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient MRF deformation model for non-rigid image matching. *CVPR*, 2007.

[22] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPRW '04*, volume 12, page 189, 2004.

[23] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24:426–433, 2005.

[24] C. Wang, M. M. Bronstein, and N. Paragios. Discrete minimum distortion correspondence problems for non-rigid shape matching. In *Research Report 7333, INRIA*, 2010.

[25] S. Wang, Y. Wang, M. Jin, X. D. Gu, and D. Samaras. Conformal geometry and its applications on 3D shape matching, recognition, and stitching. *TPAMI*, 29(7):1209–1220, 2007.

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

[27] Y. Wang, X. Huang, C. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. High resolution acquisition, learning and transfer of dynamic 3-D facial expressions. In *Computer Graphics Forum*, pages 677–686, 2004.

[28] T. Weise, H. Li, L. Van Gool, and M. Pauly. Face/off: live facial puppetry. In *SCA '09*, pages 7–16, 2009.

[29] W. Zeng, D. Samaras, and X. D. Gu. Ricci flow for 3D shape analysis. *TPAMI*, 32:662–677, 2010.

[30] W. Zeng, Y. Zeng, Y. Wang, X. Yin, X. Gu, and D. Samaras. 3D non-rigid surface matching and registration based on holomorphic differentials. In *ECCV*, 2008.

[31] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In *CVPR*, 2010.

[32] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-driven shape correspondence. *Computer Graphics Forum (SGP)*, 27(5):1431–1439, 2008.

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