

Particle-SfT: a Provably-Convergent, Fast Shape-from-Template Algorithm

Erol Özgür Adrien Bartoli

Clermont Université, France

Corresponding author: Erol Özgür

Erol0zgur@gmail.com

September 28, 2016

Submitted to the *International Journal of Computer Vision*

Abstract

The Shape-from-Template (SfT) problem is to recover the 3D shape of a deformable object from a single image, given a 3D template and a deformation constraint. We propose Particle-SfT, a new SfT algorithm which handles isometric and non-isometric deformations. We build Particle-SfT upon a particle system guided by deformation and reprojection constraint projections. Reconstruction is achieved by evolving particles to a globally attractive equilibrium, while taking observable external forces such as gravity into account, if any. Particle-SfT may be used to refine an existing initial shape. However, in practice we simply use the template as initial guess. This is because, as opposed to the existing refining methods, Particle-SfT has an extremely wide convergence basin. Particle-SfT is also faster than the existing refining methods. This is because it moves pieces of the shape's mesh independently to achieve larger step size by optimal constraint projection. We proved its convergence to a fixed-point. We experimented it with synthetic and real data. It has the same accuracy as the best performing isometric method and consistently outperforms all existing elastic methods in almost all cases, while being much faster.

Contents

1	Introduction	3
2	Previous Work	4
3	Problem Formulation	6
4	Constraints and Mappings	8
4.1	Reprojection	8
4.2	Deformation	9
4.3	Joint Reprojection and Deformation	12
5	Algorithm	15
6	Convergence Analysis	16
7	Experimental Results	21
7.1	Isometric Surfaces	21
7.2	Elastic Surfaces	25
7.3	Hidden Parts from Gravity	33
8	Conclusions	35

1 Introduction

A deformable object, like this paper that may be gently bent between your fingers, may take many different shapes whose projection produces the same image. Inferring the 3D shape of an object from a single image thus requires more constraints. SfT uses a 3D template of the object and a deformation constraint as priors to recover the unknown shape from the given input image. SfT has been well studied for the isometric deformation (Bartoli et al., 2015; Brunet et al., 2014; Chhatkuli et al., 2014; Östlund et al., 2012; Salzmann and Fua, 2011) and marginally studied for elastic deformations (Haouchine et al., 2014; Malti et al., 2013, 2015). Because SfT is fundamentally a non-convex problem, an SfT method is either *initializing* or *refining*. An initializing method computes a suboptimal solution, which is iteratively improved by a refining method. One of the main current open problems in SfT is finding a refining method which would be fast and have a large convergence basin. This problem exists for both the isometric and non-isometric deformation cases.

We propose *Particle-SfT*, an SfT algorithm which uses a particle model guided by optimal constraint projections. We design our particle model specifically to embed the two fundamental constraint sets required in SfT to converge to the observed shape, the *deformation constraints* and the *reprojection constraints*. We assume that keypoint correspondences between the template and the image are given. These may be computed prior to shape reconstruction using for instance SIFT (Lowe, 1999) followed by (Pilet et al., 2008; Pizarro and Bartoli, 2012). Our SfT method then proceeds in two steps: instantiation and evolution. We first instantiate particles at the keypoint correspondences and, if necessary, add extra particles so that a homogeneous distribution is obtained on the template’s surface. Each particle has a position, a mass and is subject to a set of internal and external constraints and forces. The internal constraints are used to represent SfT’s deformation constraints. They are created by ‘sewing’ the particles to each other along a specific pattern, which is specifically designed to prevent effects such as folding of the object’s surface. The external constraints are used to represent SfT’s reprojection constraints. They are created as strong attractors between each particle and its corresponding sight-line traced from the perspective camera, modeling projection to the input image. We then evolve the particle system by recursively applying optimal constraint projections until equilibrium.

Particle-SfT has three strong features. First, it can be controlled both in position and in force, and can thus evolve under any known external forces such as gravity. Second, it can represent materials of various stiffness. In other words, Particle-SfT can represent very stiff objects such as a piece of paper and extensible objects such as cloth, in a unified framework. Existing algorithms are all specific to a type of deformation, whether isometric or elastic. In Particle-SfT, the internal constraints are expressed by two stiffness parameters, one for stretching and one for bending. In practice, in the isometric case, this reduces

to a single parameter, the one which controls the bending stiffness, since there is no stretching. In the elastic case both stiffness parameters are lower than one and may usually be chosen equal. The design of Particle-SfT would allow it to handle objects with inhomogeneous stiffness, should a stiffness map be provided in the template. Reconstructing extensible objects requires one to provide boundary conditions, which fit easily as external constraints in Particle-SfT. Third, Particle-SfT is fast. This is because it handles the constraints independently and optimally. Therefore it takes large and precise steps. Existing methods use local approximation of all the constraints (for instance using Gauss-Newton (Brunet et al., 2014)). This limits them to take smaller steps and to converge more slowly.

From a practical standpoint, Particle-SfT has many advantages. First, it has an extremely wide convergence basin due to the globally attractive equilibrium of the involved iterations. Second, Particle-SfT is algorithmically simple, has linear complexity in the number of particles and the number of iterations, and its constraints can be separated into groups of parallelizable isolated constraints (*i.e.*, constraints that do not have common particles). Third, Particle-SfT yields similar results in accuracy to the best performing isometric SfT method and outperforms all existing elastic SfT methods in almost all cases. Importantly, existing elastic SfT methods all require one to provide a very good initialization whereas Particle-SfT converges to a sensible solution even when initialized very far from the solution. Third, Particle-SfT runs about 15 times faster than the refining isometric SfT method (Brunet et al., 2014) and at least 4 times faster than the refining elastic methods (Haouchine et al., 2014; Malti et al., 2013).

2 Previous Work

Shape-from-Template. SfT methods may be grouped by the type of deformation constraint they implement on the object’s surface. Many use *isometry*, which preserves geodesic distances (Bartoli et al., 2015; Brunet et al., 2014; Chhatkuli et al., 2014; Östlund et al., 2012; Perriollat et al., 2011; Salzmann and Fua, 2009, 2011; Salzmann et al., 2008; Varol et al., 2012), and is by far the most studied model in SfT. Some complement this constraint by a learnt statistical model (Salzmann and Fua, 2011). Others consider *non-isometric models*, namely conformity, which preserves angles (Bartoli et al., 2015), linear elasticity (Malti et al., 2013, 2015) and non-linear elasticity (Haouchine et al., 2014). The latter three methods depend on the material’s Young modulus. Stretchable surfaces were also reconstructed using shading (Moreno-Noguer et al., 2009).

SfT methods may be *initializing* or *refining*. Initializing methods have a convex formulation (Malti et al., 2015; Östlund et al., 2012; Perriollat et al., 2011; Salzmann and Fua, 2009, 2011; Salzmann et al., 2008; Varol et al., 2012) or are analytical solutions (Bartoli et al., 2015; Chhatkuli et al., 2014). Refining

methods use non-convex numerical optimization (Brunet et al., 2014; Haouchine et al., 2014; Malti et al., 2013). Initializing methods are typically less accurate than refining methods, and are often used to provide an initialization to a refining method. Particle-SfT is a refining method.

Physics- and particle-based methods. In terms of physics-based models, the Finite Element Method (FEM) was used in tracking (Kass et al., 1988; Metaxas, 1993), to reconstruct beam-like structures (Ilić and Fua, 2007), and in Non-Rigid Structure-from-Motion (Agudo et al., 2014) (NRSfM). Particle dynamics was used to track particle like rigid objects (*e.g.*, billiard balls, basketballs) and articulated body motion in 3D using a learnt model (Salzmann and Urtasun, 2011), and in NRSfM (Agudo and Moreno-Noguer, 2015).

NRSfM and SfT are related but different problems. The NRSfM methods proposed in (Agudo and Moreno-Noguer, 2015; Agudo et al., 2014, 2016) are template-free but reconstruct small incremental deformations. The NRSfM methods in (Agudo and Moreno-Noguer, 2015; Agudo et al., 2014) need at least three sequential images. They work with the orthographic camera model and have non-convex cost functions requiring careful initialization for Levenberg-Marquardt refinement. The particle-based NRSfM method in (Agudo and Moreno-Noguer, 2015) uses forces (Newtonian dynamics). The NRSfM method in (Agudo et al., 2016) works with the perspective camera model, needs two sequential images with previously reconstructed shape, and uses forces. Our method uses the perspective camera, a template, a single image, projections (projective dynamics) and reconstructs a large deformation.

In computer graphics, physics-based models (Provot, 1996) and projective particle-based models (Müller et al., 2006) were used to produce highly realistic object deformations. The convergence of projective particle dynamics approaches was however not proved so far.

We use a projective particle-based model for the first time in SfT, and prove the convergence of Particle-SfT, the proposed companion algorithm, using the fixed-point theory (Agarwal et al., 2001) and contractive mappings (Rhoades, 1977).

Particle Shape-from-Template compared to previous methods. The Particle-SfT algorithm is refining, iterative and provably convergent. It is simple, fast and versatile (it handles isometric and non-isometric deformations) compared to other SfT methods. It is guided by optimal constraint projections and, if available, by external forces.

We present our contributions as follows. First, we give the reprojection and deformation constraints and their optimal projections. These projections are the key elements of Particle-SfT but they cannot be directly used as the deformation constraint is not position attractive. Second, we combine these reprojection and deformation constraint projections to obtain a particle-pairwise joint constraint mapping. This yields an

attractive behavior on the particle pairs towards the sought-after shape of the deformed object. We then put forward three propositions and a new fixed-point theorem about the joint constraint mappings. These propositions and the new fixed-point theorem are necessary to show the convergence of Particle-SfT. We prove the convergence of Particle-SfT in Proposition 3, whose premises are established in Propositions 1 and 2, and Theorem 1. Proposition 1 proves the asymptotic regularity of a joint constraint mapping and this guarantees the minimization of its related cost function. Proposition 2 shows that a joint constraint mapping satisfies a generalized contractiveness condition which allows us to study the convergence of the set of joint constraint mappings. The new fixed-point theorem, Theorem 1, unifies a family of asymptotically regular mappings satisfying the same generalized contractiveness condition. It gives the necessary conditions on the uniqueness of the fixed-point solution. This shows that the solution is not always unique. Finally, we experimentally validate Particle-SfT against existing methods.

3 Problem Formulation

We first give the notation and assumptions, then define the particle system and finally formulate the problem.

Notation. We use the following notation:

- $n \in \mathbb{N}$ is the number of particles.
- $\mathbf{x} \in \mathbb{R}^3$ is a 3D point, $\underline{\mathbf{x}} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top]^\top \in \mathbb{R}^{3n}$ is a 3D shape vector and $\hat{\underline{\mathbf{x}}} \in \mathbb{R}^{3n}$ is its estimate. Positions are in meters.
- \mathcal{M} is a closed and bounded (*i.e.*, compact) subset of the Euclidean space \mathbb{R}^{3n} , and the metric $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}_+ = [0, \infty)$ is defined as:

$$d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{y}_i\|, \quad \forall \underline{\mathbf{x}}, \underline{\mathbf{y}} \in \mathcal{M} \subset \mathbb{R}^{3n} \quad (1)$$

where $\|\cdot\|$ is the Euclidean norm.

- $\mathbf{u} \in \mathbb{R}^2$ is a 2D image point and $\underline{\mathbf{u}} = [\mathbf{u}_1^\top, \dots, \mathbf{u}_n^\top]^\top \in \mathbb{R}^{2n}$ is an image shape vector. Image point coordinates are in pixels.
- $\mathbf{f} \in \mathbb{R}^3$ is a 3D external force vector and $\underline{\mathbf{f}} = [\mathbf{f}_1^\top, \dots, \mathbf{f}_n^\top]^\top \in \mathbb{R}^{3n}$ is a shape force vector. Forces are in Newtons. Forces are given.
- $m \in \mathbb{R}_+$ is the mass of a particle and $\underline{\mathbf{m}} = [m_1, \dots, m_n]^\top \in \mathbb{R}_+^n$ is the shape mass vector. Masses are in kilograms.

- $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is the intrinsic parameter matrix of the perspective camera model for the input image. The intrinsic parameter matrix is known.
- $\mathcal{E} = \{ij\}$ is the edge set containing the indices of connected pairs of particles.
- $\mathcal{S} = \{s_{ij}\}$ is the set of the correction strengths of the edges defined in \mathcal{E} .
- $\mathcal{R} \subset \mathcal{M} \subset \mathbb{R}^{3n}$ is the reprojection constraint set of the particles.
- $\mathcal{D}_{ij} \subset \mathcal{M} \subset \mathbb{R}^{3n}$ is a deformation constraint set for one pair of particles $ij \in \mathcal{E}$.
- $\mathcal{T} = (\underline{\mathbf{x}}_\tau, \rho_\tau, s_\tau, h_\tau)$ is the template with the shape $\underline{\mathbf{x}}_\tau$, density ρ_τ , elasticity s_τ (e.g., Young's modulus) and surface thickness h_τ parameters.
- $\mathcal{C} = \{(\mathbf{K}, \underline{\mathbf{u}}), \underline{\mathbf{f}}\}$ is the external constraint parameter set.

Particle system. The particle system $\mathcal{P} = (\underline{\mathbf{x}}, \mathcal{E}, \mathcal{S}, \underline{\mathbf{m}})$ is a model of the object. It imitates the behavior of the object by obeying the object's internal deformation constraints as well as the object's external constraints such as reprojection in the image and forces due to gravity. We instantiate the particle system's shape with the template's shape, *i.e.*, $\underline{\mathbf{x}} = \underline{\mathbf{x}}_\tau$, the edge set \mathcal{E} with the procedure described in Section 4.2, the correction strength set \mathcal{S} as a function of the template's elasticity parameter s_τ (the function is assumed to be known; this is discussed in section 8), and masses $\underline{\mathbf{m}}$ from the template's density parameter ρ_τ and the triangles issued of the particles' Delaunay triangulation (Marton et al., 2009). More precisely, the mass of a particle is determined by summing one third of the mass of all triangles containing this particle. A triangle's mass is the product of the material's density and the triangle's volume. In practice, for a set of particles homogeneously distributed on the template's surface, a particle's mass can be approximated well enough by dividing the object's known mass by the number of particles.

Problem statement. We look for a shape $\underline{\mathbf{x}}$ of particles \mathcal{P} that best fits the deformation constraints and that satisfies the reprojection constraints:

$$\min_{\underline{\mathbf{x}} \in \mathcal{R}} \left(\sum_{ij \in \mathcal{E}} \min_{\underline{\mathbf{y}} \in \mathcal{D}_{ij}} d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) \right) \quad (2)$$

In this formulation, we chose to enforce perfect reprojection constraints. This is because they are provided by a direct measurement of the sought-after shape of the deformed object from an image. The deformation constraints however are approximate because, in reality, an object never deforms isometrically. There is always a non-isometric counterpart in the deformation, even if small. Enforcing perfect isometry on the

deformed object would strongly push the object towards its rest shape. Another reason why we do not want to enforce perfect isometry is that the particle-based model is a discrete approximation of a surface and must not preserve the edges' length exactly to be able to deform. Lastly, our experiments show that the obtained precision on the reconstructed 3D shapes is good, which confirms our modeling choice. We also prefer working on the constraints that are applied on the whole shape, although they affect only pairs of particles. This simplifies the formulation and derivation of equations. We solve problem (2) through an iterative algorithm, considering it as a common fixed-point problem of the joint constraint mappings *i.e.*, reprojection and deformation constraint projections. We prove the convergence of our algorithm to a fixed point which is also a stable equilibrium point of problem (2). The solution provided by our algorithm is also robust to noise on the image points. This is because particles are connected to each other through multiple edges across the triangles of a triangulated surface mesh (shown in red in figure 2, left) forming a subset of the deformation constraints of the object. These larger distance connections across triangles make the particles less sensitive to depth errors induced by image noise, compared to the shorter ones (shown in black in figure 2, left).

4 Constraints and Mappings

4.1 Reprojection

Constraint. We define the reprojection constraints between the particles' positions and the shape reprojection set \mathcal{R} as follows:

$$f_i(\underline{\mathbf{x}}) = \|\mathbf{x}_i - T_{\mathbf{r}_i} \mathbf{x}_i\| = 0, \quad i \in [1, n] \quad (3)$$

where $T_{\mathbf{r}} \in \mathbb{R}^{3 \times 3}$ is an orthogonal projection of a particle onto its sight-line $\mathbf{r} \in \mathcal{R}$. We express a sight-line set as $\mathbf{r}_i = \{\mathbf{O} + \theta \bar{\mathbf{p}}_i \mid \theta \in \mathbb{R}\}$ where $\mathbf{O} = [0, 0, 0]^T$ is the optical center of the camera and $\bar{\mathbf{p}}_i = \mathbf{p}_i / \|\mathbf{p}_i\| \in \mathbb{R}^3$ are the known unit sight-line vectors obtained from the image shape vector $\underline{\mathbf{u}}$ as $\mathbf{p}_i = \mathbf{K}^{-1} [\mathbf{u}_i^T, 1]^T$, where \mathbf{u}_i is the position of the i -th particle in the input image. Each sight-line \mathbf{r} is a convex and linear set.

Mapping. The orthogonal projection $T_{\mathbf{r}}$, which solves the reprojection constraint (3) in one step by bringing the particle at $\mathbf{x} \notin \mathbf{r}$ back onto its sight-line \mathbf{r} , is defined as:

$$T_{\mathbf{r}} = \bar{\mathbf{p}} \bar{\mathbf{p}}^T \quad (4)$$

Note that $T_{\mathbf{r}} \mathbf{x}$ always exists. Since a sight-line set \mathbf{r} is convex and $T_{\mathbf{r}}$ is its orthogonal projection (*i.e.*, a nearest point mapping), then $T_{\mathbf{r}} \mathbf{x}$ is a unique point. We handle the reprojection constraints of a pair of particles $p_i \in \mathcal{P}$ and $p_j \in \mathcal{P}$ in a shape $\underline{\mathbf{x}} \in \mathbb{R}^{3n}$ as:

$$\underline{\mathbf{x}} = T_{\mathbf{R}_{ij}} \underline{\mathbf{x}} \quad (5)$$

where $\mathbf{T}_{\mathbf{R}_{ij}} = \text{diag}(\mathbf{I}_3, \dots, \mathbf{I}_3, T_{\mathbf{r}_i}, \mathbf{I}_3, \dots, \mathbf{I}_3, T_{\mathbf{r}_j}, \mathbf{I}_3, \dots, \mathbf{I}_3) \in \mathbb{R}^{3n \times 3n}$ with \mathbf{I}_3 being a 3×3 identity matrix.

4.2 Deformation

Constraint. We define a deformation constraint between a pair of connected particles at \mathbf{x}_i and \mathbf{x}_j of a shape $\underline{\mathbf{x}}$ through the following implicit convex signed distance function:

$$g_{ij}(\underline{\mathbf{x}}) = \|\mathbf{x}_i - \mathbf{x}_j\| - \ell_{ij} = 0, \quad ij \in \mathcal{E} \quad (6)$$

where ℓ_{ij} is the known distance of the pair, obtained from the template. This deformation constraint tells us how far a deformed pair is from the closest equilibrium and also in which relative configuration (a positive value for a stretched pair and a negative value for a shrunk pair). An equilibrium is a relative placement of particles $p_i \in \mathcal{P}$ and $p_j \in \mathcal{P}$ anywhere in \mathbb{R}^3 such that $\|\mathbf{x}_i - \mathbf{x}_j\| = \ell_{ij}$.

Mapping. Whenever $g_{ij}(\underline{\mathbf{x}}) \neq 0$, we would like to find the projection step $\Delta\underline{\mathbf{x}}$ which solves the deformation constraint as $g_{ij}(\underline{\mathbf{x}} + \Delta\underline{\mathbf{x}}) = 0$. To do so, we fix one particle of a pair, say p_i at \mathbf{x}_i , with respect to the other. This allows us to illustrate the convex signed distance function $g_{ij}(\underline{\mathbf{x}})$ as in figure 1. Remark that the shape of $g_{ij}(\underline{\mathbf{x}})$ between the positions of particles becomes linear. This is also true the other way around when the roles of particles p_i and p_j are exchanged. Fixing one particle therefore lets us express $g_{ij}(\underline{\mathbf{x}} + \Delta\underline{\mathbf{x}}) = 0$

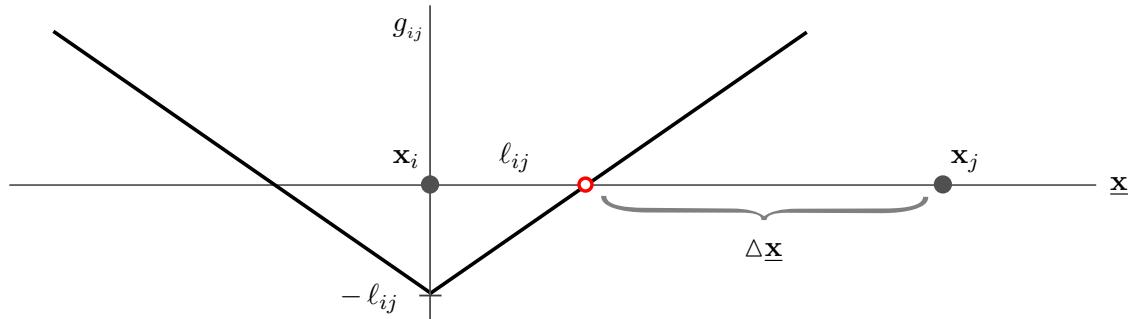


Figure 1: Convex implicit signed distance function as deformation constraint.

exactly with a first-order representation as:

$$g_{ij}(\underline{\mathbf{x}} + \Delta \underline{\mathbf{x}}) = g_{ij}(\underline{\mathbf{x}}) + \nabla g_{ij}(\underline{\mathbf{x}})^\top \Delta \underline{\mathbf{x}} = 0 \quad (7)$$

Remark also that the gradient of $g_{ij}(\underline{\mathbf{x}})$ at a particle's position is piecewise constant, and especially it does not vary between any point and the closest solution, *e.g.*, the red circle in figure 1. We can thus solve constraint (7) with an exact projection step along the constant gradient direction:

$$\Delta \underline{\mathbf{x}} = -\frac{g_{ij}(\underline{\mathbf{x}})}{\|\nabla g_{ij}(\underline{\mathbf{x}})\|^2} \nabla g_{ij}(\underline{\mathbf{x}}) \quad (8)$$

We now propose to solve constraint (6) in one step by moving both particles simultaneously with respect to each other by sharing the projection step (8) rather than moving only one with respect to the other. This is possible because the gradient vectors of $g_{ij}(\underline{\mathbf{x}})$ at the particles' positions are aligned on the same 3D line passing through the particles' positions \mathbf{x}_i and \mathbf{x}_j . We share the projection step between the particles with respect to their mass ratios $\alpha_i = m_j/(m_i + m_j)$ and $\alpha_j = m_i/(m_i + m_j)$. The heavier a particle, the smaller its α and therefore the smaller its displacement. Remark that $\alpha_i + \alpha_j = 1$. Therefore the sum of the displacements of both particles remains equal to the exact projection step length. We rewrite equation (8) explicitly for the displacement of particles p_i and p_j only as:

$$\begin{bmatrix} \Delta \mathbf{x}_i \\ \Delta \mathbf{x}_j \end{bmatrix} = \begin{bmatrix} -\alpha_i \frac{g_{ij}(\underline{\mathbf{x}})}{\|\nabla_i g_{ij}(\underline{\mathbf{x}})\|^2} \nabla_i g_{ij}(\underline{\mathbf{x}}) \\ -\alpha_j \frac{g_{ij}(\underline{\mathbf{x}})}{\|\nabla_j g_{ij}(\underline{\mathbf{x}})\|^2} \nabla_j g_{ij}(\underline{\mathbf{x}}) \end{bmatrix} \quad (9)$$

where $\nabla_i g_{ij}(\underline{\mathbf{x}}) = (\mathbf{x}_i - \mathbf{x}_j)/\|\mathbf{x}_i - \mathbf{x}_j\| \in \mathbb{R}^3$ is the gradient vector of g_{ij} with respect to the particle at \mathbf{x}_i , and where $\nabla_j g_{ij}(\underline{\mathbf{x}}) = -\nabla_i g_{ij}(\underline{\mathbf{x}}) \in \mathbb{R}^3$ is the gradient vector of g_{ij} with respect to the particle at \mathbf{x}_j . We also scale the displacements of particles with a correction strength $s_{ij} \in (0, 1]$ to introduce an elastic behavior on their deformation. The correction strength is directly related to the object material's stiffness, and is provided with the template as it represents a characteristic of the object, similarly to Young's modulus or any other mechanical parameter. Consequently, we design for the deformed pair of particles two weighted gradient-based orthogonal projections ${}^j \mathbf{T}_i \in \mathbb{R}^{3 \times 3}$ and ${}^i \mathbf{T}_j \in \mathbb{R}^{3 \times 3}$ to solve their deformation constraint (6) using (9) as:

$${}^j \mathbf{T}_i = \left(\mathbf{I}_3 - s_{ij} \alpha_i g_{ij}(\underline{\mathbf{x}}) \nabla_i g_{ij}(\underline{\mathbf{x}}) \frac{\mathbf{x}_i^\top}{\|\mathbf{x}_i\|^2} \right) \quad (10)$$

$${}^i \mathbf{T}_j = \left(\mathbf{I}_3 - s_{ij} \alpha_j g_{ij}(\underline{\mathbf{x}}) \nabla_j g_{ij}(\underline{\mathbf{x}}) \frac{\mathbf{x}_j^\top}{\|\mathbf{x}_j\|^2} \right) \quad (11)$$

Remark that the gradient vectors of the connected particles have unit norm, which was therefore crossed out from the denominators of the projections (10) and (11). If $s_{ij} = 1$, then the projections are isometric and otherwise they are non-isometric. The deformation constraint (6) is then solved in one step when the pair of connected particles is projected through mappings (10) and (11) at the same time:

$$\underline{\mathbf{x}} = \mathbf{T}_{D_{ij}} \underline{\mathbf{x}} \quad (12)$$

where $\mathbf{T}_{D_{ij}} = \text{diag}(\mathbf{I}_3, \dots, \mathbf{I}_3, {}^j\mathbf{T}_i, \mathbf{I}_3, \dots, \mathbf{I}_3, {}^i\mathbf{T}_j, \mathbf{I}_3, \dots, \mathbf{I}_3) \in \mathbb{R}^{3n \times 3n}$ is an orthogonal projection mapping of only one pair in a shape $\underline{\mathbf{x}}$. The deformation constraint set \mathcal{D}_{ij} , which first appeared in equation (2), can then be defined explicitly, using $\mathbf{T}_{D_{ij}}$ as:

$$\mathcal{D}_{ij} = \{\underline{\mathbf{x}} \mid \underline{\mathbf{x}} = \mathbf{T}_{D_{ij}} \underline{\mathbf{x}}, \underline{\mathbf{x}} \in \mathcal{M}\} \quad (13)$$

We note that if $\mathbf{x}_i \neq \mathbf{x}_j$ then ${}^j\mathbf{T}_i \mathbf{x}_i$ and ${}^i\mathbf{T}_j \mathbf{x}_j$ exist and are unique points in \mathbb{R}^3 . Otherwise if $\mathbf{x}_i = \mathbf{x}_j$, then ${}^j\mathbf{T}_i \mathbf{x}_i$ and ${}^i\mathbf{T}_j \mathbf{x}_j$ still exist but this time they are set valued mappings in \mathbb{R}^3 . More explicitly they map to concentric sphere sets and the center is the singular configuration. Remark that equation (5) keeps a pair away from the singular configuration of the deformation constraint.

Meshing. We used Delaunay triangulation (Marton et al., 2009) to obtain the edges defining the A-constraints shown in black in figure 2, left. The A-constraints preserve the topology of the particles and control the in-plane shear deformation of the model. We then complete the mesh by adding edges across the triangles, to form the B-constraints shown in red in figure 2, left. This was inspired by a procedure used for the mass-spring model (Provot, 1996). This procedure was using a rectangular regular grid which we extended here to handle a non-regular particle system. The B-constraints control both out-of-plane bending and in-plane shear deformation of the model. Intuitively, B-constraints capture effects such as curvature, as they connect particles to their second-order neighborhood. Without them the model would be prone to fold, especially at its corners. Another important advantage of the B-constraints is to make the particles less sensitive to depth ambiguities which might occur due to noise on the image points. This is because they have larger distance connections than A-constraints.

Parallelization. The constraints can be separated into groups of parallelizable isolated constraints (*i.e.*, constraints that do not have common particles). Figure 2, right illustrates parallelizable constraints. Each color, except black, represents a group of isolated constraints which can be solved in one step in parallel. Groups are solved sequentially. For the larger meshes one typically uses in practice, the constraints are

parallelizable to a good extent.

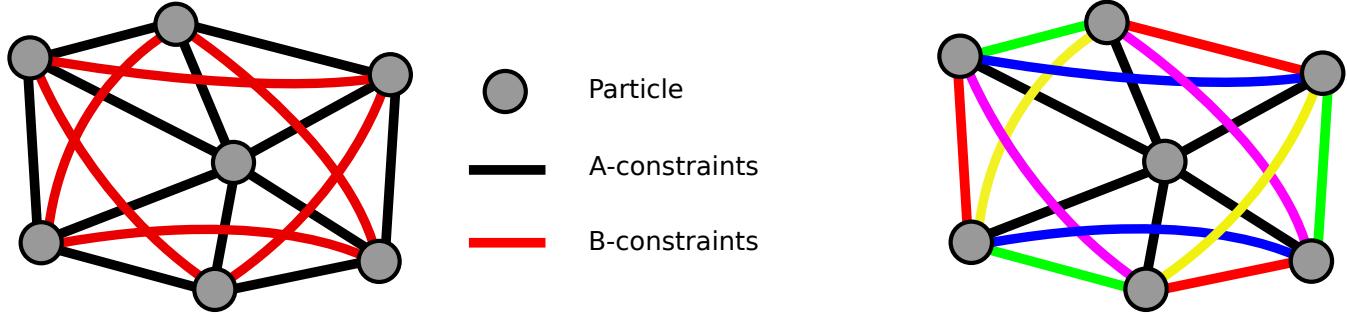


Figure 2: Meshing for deformation constraints (left). Parallelizable isolated constraints in red, green, blue and purple colors (right).

4.3 Joint Reprojection and Deformation

Motivation. The motivation behind using a joint constraint mapping is that the reprojection constraints are *position-attractive* whereas the deformation constraints are not. This is because deformation is a relative configuration among the particles, that may hold independently from the particles’ positions in space. Applying each type of constraint independently on the particles makes convergence analysis tremendously difficult, because of the non-attractive behavior of the deformation constraints. Therefore, we combine a deformation constraint for a pair of particles with the reprojection constraints of the same pair in order to obtain a new *position-attractive constraint* on the pair, including both reprojection and deformation constraints.

Mapping. A joint constraint mapping $T_{ij} : \mathcal{M} \rightarrow \mathcal{M}$ is made up of the successive projections of a deformation constraint and then the reprojection constraints of a pair of particles:

$$\underline{x}_{k+1} = T_{ij} \underline{x}_k \quad (14)$$

with $T_{ij} = T_{R_{ij}} T_{D_{ij}} \in \mathbb{R}^{3n \times 3n}$. The joint constraint mapping T_{ij} does not fall into a singular configuration of the deformation constraints if the initial shape is not singular. Therefore it yields continuous, uniquely defined constraint mappings. Subsequently, from Brouwer’s fixed-point theorem (Brouwer, 1910), this implies the existence of fixed points of the joint constraint mappings.

We now give Proposition 1 on a property of a joint constraint mapping which is one of the premises of Proposition 3, for the proof of convergence of Particle-SfT.

Proposition 1. A joint constraint mapping T_{ij} is asymptotically regular on \mathcal{M} and convergent. It also minimizes its associated deformation constraint when applied successively on a shape.

In order to prove Proposition 1, we need Lemma 1.

Lemma 1. If $g_{ij}(\underline{\mathbf{x}}_k) \rightarrow 0$, then $d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) \rightarrow 0$.

Proof. Here d is the metric (1), the sum of distances between the corresponding particles. We write the explicit expression of the metric for the deformation mapping using projections (10) and (11) as:

$$d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) = \| s_{ij} \alpha_i g_{ij}(\underline{\mathbf{x}}_k) \nabla_i g_{ij}(\underline{\mathbf{x}}_k) \| + \| s_{ij} \alpha_j g_{ij}(\underline{\mathbf{x}}_k) \nabla_j g_{ij}(\underline{\mathbf{x}}_k) \| \quad (15)$$

where $s_{ij}, \alpha_i, \alpha_j \in \mathbb{R}_+$. Remember that $\nabla_j g_{ij}(\underline{\mathbf{x}}_k) = -\nabla_i g_{ij}(\underline{\mathbf{x}}_k)$ and subsequently their absolute values are equal. We rewrite equation (15) by replacing $\|\nabla_j g_{ij}(\underline{\mathbf{x}}_k)\|$ with $\|\nabla_i g_{ij}(\underline{\mathbf{x}}_k)\|$ as:

$$d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) = s_{ij} |g_{ij}(\underline{\mathbf{x}}_k)| \|\nabla_i g_{ij}(\underline{\mathbf{x}}_k)\| (\alpha_i + \alpha_j) \quad (16)$$

where $\alpha_i + \alpha_j = 1$ by definition and $\|\nabla_i g_{ij}(\underline{\mathbf{x}}_k)\| = 1$ by construction. We are then left with:

$$d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) = s_{ij} |g_{ij}(\underline{\mathbf{x}}_k)| \quad (17)$$

We conclude from equation (17) that if $g_{ij}(\underline{\mathbf{x}}_k) \rightarrow 0$, then $d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) \rightarrow 0$, and vice versa. \square

Proof of Proposition 1. An asymptotically regular mapping is convergent (Belluce and Kirk, 1969). We prove the asymptotic regularity of the joint constraint mapping T_{ij} using Lyapunov's direct method. Let $V : \mathcal{M} \rightarrow \mathbb{R}$ be a Lyapunov candidate function defined over the deformation error as:

$$V(\underline{\mathbf{x}}_k) = \frac{1}{2} g_{ij}(\underline{\mathbf{x}}_k)^2 \quad (18)$$

Here V is a positive semi-definite and radially unbounded function:

$$V = 0 \iff g_{ij}(\underline{\mathbf{x}}_k) = 0 \quad (19)$$

$$V > 0 \iff g_{ij}(\underline{\mathbf{x}}_k) \neq 0 \quad (20)$$

$$V \rightarrow \infty \iff |g_{ij}(\underline{\mathbf{x}}_k)| \rightarrow \infty \quad (21)$$

Afterward if $\dot{V} < 0$ when $V > 0$, and if $\dot{V} = 0$ when $V = 0$, then $V \rightarrow 0$ (Lyapunov, 1892). Subsequently this implies that $g_{ij}(\underline{\mathbf{x}}_k) \rightarrow 0$ and from Lemma 1 that $d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) \rightarrow 0$. Differentiating V with respect

to time yields:

$$\dot{V} = g_{ij}(\underline{\mathbf{x}}_k) \dot{g}_{ij}(\underline{\mathbf{x}}_k) \quad (22)$$

where:

$$\dot{g}_{ij}(\underline{\mathbf{x}}_k) = \nabla_i g_{ij}(\underline{\mathbf{x}}_k)^\top \dot{\mathbf{x}}_i + \nabla_j g_{ij}(\underline{\mathbf{x}}_k)^\top \dot{\mathbf{x}}_j \quad (23)$$

Considering that time takes unit discrete steps, we estimate $\dot{\mathbf{x}}_i$ and $\dot{\mathbf{x}}_j$ from the displacement $\underline{\mathbf{x}}_{k+1} - \underline{\mathbf{x}}_k$ where $\underline{\mathbf{x}}_{k+1} = T_{ij} \underline{\mathbf{x}}_k$, as:

$$\dot{\mathbf{x}}_i = -\alpha_i g_{ij}(\underline{\mathbf{x}}_k) T_{\mathbf{r}_i} \nabla_i g_{ij}(\underline{\mathbf{x}}_k) \quad (24)$$

$$\dot{\mathbf{x}}_j = -\alpha_j g_{ij}(\underline{\mathbf{x}}_k) T_{\mathbf{r}_j} \nabla_j g_{ij}(\underline{\mathbf{x}}_k) \quad (25)$$

Substituting equations (23), (24) and (25) into equation (22) yields:

$$\dot{V} = -\lambda g_{ij}(\underline{\mathbf{x}}_k)^2 \leq 0 \quad (26)$$

with λ being:

$$\lambda = s_{ij} \left(\alpha_i \nabla_i g_{ij}(\underline{\mathbf{x}}_k)^\top T_{\mathbf{r}_i} \nabla_i g_{ij}(\underline{\mathbf{x}}_k) + \alpha_j \nabla_j g_{ij}(\underline{\mathbf{x}}_k)^\top T_{\mathbf{r}_j} \nabla_j g_{ij}(\underline{\mathbf{x}}_k) \right) \quad (27)$$

We note that λ is positive because $s_{ij}, \alpha_i, \alpha_j \in (0, 1]$ and $T_{\mathbf{r}_i}, T_{\mathbf{r}_j}$ are orthogonal projections with two distinct eigenvalues $\{0, 1\}$ which cancel only some coordinates of the gradient vectors $\nabla_i g_{ij}(\underline{\mathbf{x}}_k)$ and $\nabla_j g_{ij}(\underline{\mathbf{x}}_k)$. We also know that $\|\nabla_i g_{ij}(\underline{\mathbf{x}})\| = 1$ and $\|\nabla_j g_{ij}(\underline{\mathbf{x}})\| = 1$. Subsequently this yields the following inequalities $0 < \nabla_i g_{ij}(\underline{\mathbf{x}}_k)^\top T_{\mathbf{r}_i} \nabla_i g_{ij}(\underline{\mathbf{x}}_k) < 1$ and $0 < \nabla_j g_{ij}(\underline{\mathbf{x}}_k)^\top T_{\mathbf{r}_j} \nabla_j g_{ij}(\underline{\mathbf{x}}_k) < 1$. Consequently, we have $\dot{V} \leq 0$.

Here \dot{V} is uniformly continuous everywhere in \mathcal{M} except at the optical center \mathbf{O} of the camera. In equation (26), if $g_{ij}(\underline{\mathbf{x}}_k) \neq 0$, then $\dot{V} < 0$, and if $g_{ij}(\underline{\mathbf{x}}_k) = 0$, then $\dot{V} = 0$. It follows that both $V \rightarrow 0$ and $\dot{V} \rightarrow 0$, consequently $g_{ij}(\underline{\mathbf{x}}_k) \rightarrow 0$ and therefore $d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) \rightarrow 0$. From this result, we make the following conclusion:

$$d(\underline{\mathbf{x}}_k, T_{D_{ij}} \underline{\mathbf{x}}_k) > d(\underline{\mathbf{x}}_{k+1}, T_{D_{ij}} \underline{\mathbf{x}}_{k+1}), \quad \forall k \geq 0 \quad (28)$$

for $\underline{\mathbf{x}}_k, \underline{\mathbf{x}}_{k+1} \in \mathcal{R}$ and $\underline{\mathbf{x}}_{k+1} = T_{ij} \underline{\mathbf{x}}_k$. Then we note that:

$$d(T_{R_{ij}} \underline{\mathbf{x}}, T_{R_{ij}} \underline{\mathbf{y}}) < d(\underline{\mathbf{x}}, \underline{\mathbf{y}}), \quad \forall \underline{\mathbf{x}}, \underline{\mathbf{y}} \in \mathcal{M} \quad (29)$$

for $\underline{x} \in \mathcal{R}$ and $\underline{y} \notin \mathcal{R}$. Figure 3 illustrates this case on two particles. We use equation (29) to write the

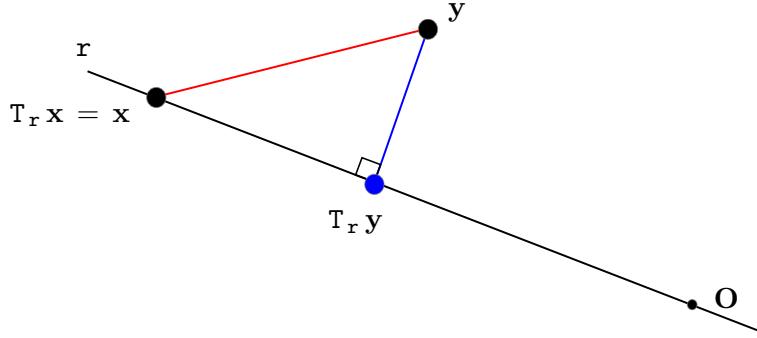


Figure 3: Projection of two particles at $\underline{x} \in \mathcal{R}$ and $\underline{y} \notin \mathcal{R}$ onto the sight-line r which passes through the optical center \mathbf{O} .

following relation:

$$d(\underline{x}_k, T_{D_{ij}} \underline{x}_k) > d(T_{R_{ij}} \underline{x}_k, T_{R_{ij}} T_{D_{ij}} \underline{x}_k) \quad (30)$$

which is true when $\underline{x}_k \neq T_{ij} \underline{x}_k$ or $g_{ij}(\underline{x}_k) \neq 0$. Remark that:

$$T_{R_{ij}} T_{D_{ij}} \underline{x}_k = T_{ij} \underline{x}_k = \underline{x}_{k+1} \quad (31)$$

and when $\underline{x}_k \in \mathcal{R}$ that:

$$T_{R_{ij}} \underline{x}_k = \underline{x}_k \quad (32)$$

Therefore equation (30) can be rewritten as:

$$d(\underline{x}_k, T_{D_{ij}} \underline{x}_k) > d(\underline{x}_k, \underline{x}_{k+1}) \quad (33)$$

Since $d(\underline{x}_k, T_{D_{ij}} \underline{x}_k) \rightarrow 0$ for all $\underline{x} \in \mathcal{M}$, then $d(\underline{x}_k, \underline{x}_{k+1}) \rightarrow 0$ which implies $d(T_{ij}^k \underline{x}, T_{ij}^{k+1} \underline{x}) \rightarrow 0$. Furthermore, the fixed points of T_{ij} are globally attractive since V is radially unbounded, and globally stable equilibrium points of V since both $V = 0$ and $\dot{V} = 0$ whenever $g_{ij}(\underline{x}_k) = 0$. \square

5 Algorithm

The algorithm in figure 4 finds a stable equilibrium shape for problem (2). Line 01 initializes the shape, velocity, external forces and the masses of the particles as well as the iteration counter of the main loop. The initial shape must be a valid shape (non-singular). In practice, any initial shape in front of or behind the camera is a good starting point. Line 02 starts the main loop. Line 03 exerts the external forces, if known, on the shape with weight $\beta \in [0, 1]$. In line 03, if the masses of particles are approximated, then β

also helps smoothing out these effects. Line 04 predicts the future pose using the damped velocities of the particles with the weight vector $\underline{\kappa} \in (\mathbf{0}, \mathbf{1})$. Parameter $\underline{\kappa} = \mathbf{1} - \underline{\mu}$ weights the prediction using $\underline{\mu} \in \mathbb{R}^n$ the damping coefficient vector. $\underline{\mu}$ is made of $\mu_i \approx 2\sqrt{s_{min}} \min(m_i, 0.25 - \epsilon)$ coefficients where each critically damps a particle's motion (Goldstein, 1980). $s_{min} = \min \{ s_{ij} \mid ij \in \mathcal{E} \}$ is the minimum of all the elasticity parameters. If $0 < s_{ij} < 1$ then the connected pair behaves elastically, and if $s_{ij} \rightarrow 1$ then the connected pair behaves isometrically. Line 05 applies successively the joint constraint mappings. Line 06 computes the new velocity. Line 07 updates the shape. Line 08 ends the algorithm either when the root-mean-squared (RMS) value of the shape velocity is lower than a very small positive threshold value ϵ or when the maximum number of loop iteration $maxiter$ is reached. Finally, the shape might escape from one side to the other due to the prediction step in line 04. Line 09 brings the shape in front of the camera if required: if it is behind, then it is reflected about the optical center to its correct symmetrical pose. In all our experiments, we set $\epsilon = 10^{-6}$ and $maxiter = 10^4$.

Procedure : Particle-SfT(\mathcal{P}, \mathcal{C})

Inputs : Particle system \mathcal{P} , constraints \mathcal{C}

Output : Reconstructed shape \underline{x}

```

01:  $\underline{x} = \text{shape}(\mathcal{P}), \quad \underline{v} = \mathbf{0}, \quad \underline{f} = \text{forces}(\mathcal{C}), \quad \underline{m} = \text{masses}(\mathcal{P}), \quad k = 0$ 
02: do
03:    $\underline{v} = \underline{v} + \beta \underline{f} / \underline{m}$ 
04:    $\widehat{\underline{x}} = \underline{x} + \underline{\kappa} \underline{v}$ 
05:   for each  $ij$  in  $\mathcal{E}$  do,  $\widehat{\underline{x}} = T_{ij} \widehat{\underline{x}}$ , end
06:    $\underline{v} = \widehat{\underline{x}} - \underline{x}$ 
07:    $\underline{x} = \widehat{\underline{x}}$ 
08: until  $RMS(\underline{v}) \leq \epsilon$  or  $k++ \geq maxiter$ 
09: if  $\underline{x}$  is behind the camera then  $\underline{x} = -\underline{x}$ 

```

Figure 4: The particle-SfT algorithm.

6 Convergence Analysis

Proposition 1 showed that a joint constraint mapping is asymptotically regular and therefore convergent, and has globally attractive fixed-points. Furthermore, the fixed-points are globally stable equilibrium points of the associated deformation constraint of a joint constraint mapping. Problem (2) is the sum of the deformation constraints associated to these joint constraint mappings. We now propose a new fixed-point theorem and then two more propositions to develop the convergence proof of Particle-SfT. The new fixed-point theorem, Theorem 1, shows that a family of asymptotically regular mappings satisfying the same

generalized contractiveness condition is convergent and has fixed-points. Theorem 1 also gives the uniqueness condition for the fixed-point solution. The solution may not always be unique. Afterward, Proposition 2 is given. It states that a joint constraint mapping satisfies the generalized contractiveness condition proposed in Theorem 1. Finally, Proposition 3 is given. It combines Propositions 1 and 2, and Theorem 1, and proves the convergence of Particle-SfT as given in figure 4.

Theorem 1. Let (\mathcal{M}, d) be a compact metric space and $\{Z_h\}_{h=1}^{\infty}$ be a sequence of self-mappings of \mathcal{M} . Let each mapping $Z_h : \mathcal{M} \rightarrow \mathcal{M}$ be asymptotically regular with a fixed point \mathbf{x}_h and satisfy the following generalized contractiveness condition for all $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ and $h \geq 1$:

$$\gamma (d(\mathbf{x}, \mathbf{y}) + d(\mathbf{x}, Z_h \mathbf{x}) + d(\mathbf{y}, Z_h \mathbf{y}) + d(\mathbf{x}, Z_h \mathbf{y}) + d(\mathbf{y}, Z_h \mathbf{x})) \geq d(Z_h \mathbf{x}, Z_h \mathbf{y}) \quad (34)$$

where $0 \leq \gamma < \frac{1}{2}$. Suppose that the sequence $\{Z_h\}$ converges pointwise to a mapping $Q : \mathcal{M} \rightarrow \mathcal{M}$ and that \mathbf{x}^* is any cluster point of the sequence $\{\mathbf{x}_h\}$ of fixed points of $\{Z_h\}$. Then \mathbf{x}^* is a fixed point of mapping Q . Furthermore, if $0 \leq \gamma < \frac{1}{3}$, then \mathbf{x}^* is the unique fixed point of mapping Q .

Proof. Since \mathbf{x}^* is a cluster point of the sequence $\{\mathbf{x}_h\}$, there is a subsequence $\{\mathbf{x}_{h_i}\}$ of $\{\mathbf{x}_h\}$ which converges to \mathbf{x}^* as $h_i \rightarrow \infty$. By contradiction, assume that $Q \mathbf{x}^* \neq \mathbf{x}^*$. This subsequently implies $d(\mathbf{x}^*, Q \mathbf{x}^*) > 0$. We then write the following relation using the triangular inequality twice:

$$d(\mathbf{x}^*, Q \mathbf{x}^*) \leq d(\mathbf{x}^*, Z_{h_i} \mathbf{x}_{h_i}) + d(Z_{h_i} \mathbf{x}_{h_i}, Z_{h_i} \mathbf{x}^*) + d(Z_{h_i} \mathbf{x}^*, Q \mathbf{x}^*) \quad (35)$$

and using equation (34) this leads to:

$$\begin{aligned} d(\mathbf{x}^*, Q \mathbf{x}^*) &\leq d(\mathbf{x}^*, Z_{h_i} \mathbf{x}_{h_i}) \\ &+ \gamma (d(\mathbf{x}_{h_i}, \mathbf{x}^*) + d(\mathbf{x}_{h_i}, Z_{h_i} \mathbf{x}_{h_i}) + d(\mathbf{x}^*, Z_{h_i} \mathbf{x}^*) + d(\mathbf{x}_{h_i}, Z_{h_i} \mathbf{x}^*) + d(\mathbf{x}^*, Z_{h_i} \mathbf{x}_{h_i})) \\ &+ d(Z_{h_i} \mathbf{x}^*, Q \mathbf{x}^*) \end{aligned} \quad (36)$$

Since \mathbf{x}_{h_i} is a fixed-point of Z_{h_i} then $Z_{h_i} \mathbf{x}_{h_i} = \mathbf{x}_{h_i}$. Therefore this implies $d(\mathbf{x}^*, Z_{h_i} \mathbf{x}_{h_i}) = d(\mathbf{x}^*, \mathbf{x}_{h_i})$ and $d(\mathbf{x}_{h_i}, Z_{h_i} \mathbf{x}_{h_i}) = 0$. Hence equation (36) becomes:

$$\begin{aligned} d(\mathbf{x}^*, Q \mathbf{x}^*) &\leq d(\mathbf{x}^*, \mathbf{x}_{h_i}) \\ &+ \gamma (d(\mathbf{x}_{h_i}, \mathbf{x}^*) + 0 + d(\mathbf{x}^*, Z_{h_i} \mathbf{x}^*) + d(\mathbf{x}_{h_i}, Z_{h_i} \mathbf{x}^*) + d(\mathbf{x}^*, \mathbf{x}_{h_i})) \\ &+ d(Z_{h_i} \mathbf{x}^*, Q \mathbf{x}^*) \end{aligned} \quad (37)$$

Since $\mathbf{x}_{h_i} \rightarrow \mathbf{x}^*$ and $\mathbf{z}_{h_i} \rightarrow \mathbf{Q}$ as $h_i \rightarrow \infty$, then $d(\mathbf{x}^*, \mathbf{x}_{h_i}) \rightarrow 0$ and $d(\mathbf{z}_{h_i} \mathbf{x}^*, \mathbf{Q} \mathbf{x}^*) \rightarrow 0$ and $d(\mathbf{x}^*, \mathbf{z}_{h_i} \mathbf{x}^*) \rightarrow d(\mathbf{x}^*, \mathbf{Q} \mathbf{x}^*)$ and $d(\mathbf{x}_{h_i}, \mathbf{z}_{h_i} \mathbf{x}^*) \rightarrow d(\mathbf{x}^*, \mathbf{Q} \mathbf{x}^*)$. Hence equation (37) yields:

$$d(\mathbf{x}^*, \mathbf{Q} \mathbf{x}^*) \leq 2\gamma d(\mathbf{x}^*, \mathbf{Q} \mathbf{x}^*) \quad (38)$$

This is a contradiction since $0 \leq \gamma < \frac{1}{2}$. It follows that $\mathbf{x}^* = \mathbf{Q} \mathbf{x}^*$. To see that \mathbf{x}^* is unique, let there be another fixed point, say \mathbf{y}^* , with $\mathbf{Q} \mathbf{y}^* = \mathbf{y}^*$. By contradiction, assume that $\mathbf{y}^* \neq \mathbf{x}^*$ which implies $d(\mathbf{x}^*, \mathbf{y}^*) > 0$. We then write the following relation using the given generalized contractiveness condition:

$$d(\mathbf{x}^*, \mathbf{y}^*) = d(\mathbf{Q} \mathbf{x}^*, \mathbf{Q} \mathbf{y}^*) \leq \gamma (d(\mathbf{x}^*, \mathbf{y}^*) + d(\mathbf{x}^*, \mathbf{Q} \mathbf{x}^*) + d(\mathbf{y}^*, \mathbf{Q} \mathbf{y}^*) + d(\mathbf{x}^*, \mathbf{Q} \mathbf{y}^*) + d(\mathbf{y}^*, \mathbf{Q} \mathbf{x}^*)) \quad (39)$$

and this yields:

$$d(\mathbf{x}^*, \mathbf{y}^*) \leq 3\gamma d(\mathbf{x}^*, \mathbf{y}^*) \quad (40)$$

which is a contradiction for $0 \leq \gamma < \frac{1}{3}$. It then follows that $\mathbf{y}^* = \mathbf{x}^*$ for $0 \leq \gamma < \frac{1}{3}$. \square

We now give Proposition 2 on contractiveness of a joint constraint mapping.

Proposition 2. *A joint constraint mapping T_{ij} conforms to the generalized contractiveness condition (34) proposed in Theorem 1 with $0 \leq \gamma < \frac{1}{2}$.*

Proof. We can write the following relations using the triangular inequality:

$$d(T_{ij} \underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) \leq d(T_{ij} \underline{\mathbf{x}}, \underline{\mathbf{y}}) + d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{y}}) + d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) \quad (41a)$$

$$d(T_{ij} \underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) \leq d(T_{ij} \underline{\mathbf{x}}, \underline{\mathbf{x}}) + d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) \quad (41b)$$

Summing equations (41a) and (41b), we obtain:

$$d(T_{ij} \underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) \leq \frac{1}{2} (d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) + d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{x}}) + d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{y}}) + d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) + d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{x}})) \quad (42)$$

If $\underline{\mathbf{x}} \neq \underline{\mathbf{y}}$, then this becomes exactly the generalized contractiveness condition (34) with $\gamma \in [0, \frac{1}{2}]$:

$$d(T_{ij} \underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) \leq \gamma (d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) + d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{x}}) + d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{y}}) + d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) + d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{x}})) \quad (43)$$

If $\underline{\mathbf{x}} = \underline{\mathbf{y}}$, then equation (43) turns to be:

$$0 \leq 4\gamma d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{x}}) \quad (44)$$

which is still valid with $\gamma \in [0, \frac{1}{2}]$. We give an example which illustrates the worst case scenario. The worst case, $\gamma \rightarrow \frac{1}{2}$, is observed when $\underline{\mathbf{x}} \neq \underline{\mathbf{y}}$, the line segments $\mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{y}_i - \mathbf{y}_j$ intersect, both $g_{ij}(\underline{\mathbf{x}}) < 0$ and $g_{ij}(\underline{\mathbf{y}}) < 0$ and the intersection angle $\varphi \leq \pi - \theta$, where θ is the angle between the sight-lines of a pair. Figure 5 illustrates a configuration of pairs in the worst case. In figure 5, $d(T_{ij} \underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) = a + b + c + d + e + f$,

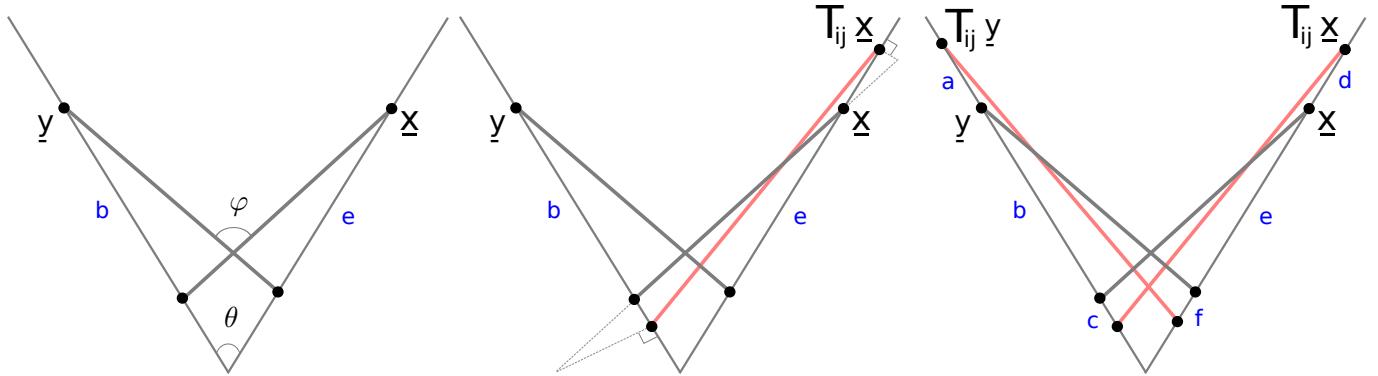


Figure 5: A configuration of pairs in the worst case. Grey pairs show the initial configuration on the sight-lines. Red pairs show the configuration after the joint constraint mappings.

$d(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = b + e$, $d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{x}}) = c + d$, $d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{y}}) = a + f$, $d(\underline{\mathbf{x}}, T_{ij} \underline{\mathbf{y}}) = a + b + e + f$, $d(\underline{\mathbf{y}}, T_{ij} \underline{\mathbf{x}}) = b + c + d + e$. Hence the worst case conforms to the generalized contractiveness condition for all $\underline{\mathbf{x}}, \underline{\mathbf{y}} \in \mathcal{M}$ with $\gamma \in [0, \frac{1}{2}]$:

$$a + b + c + d + e + f \leq \gamma (2a + 3b + 2c + 2d + 3e + 2f) \quad (45)$$

This concludes the proof. \square

We finally give Proposition 3, built upon Propositions 1 and 2 and Theorem 1.

Proposition 3. *Particle-SfT is convergent.*

Proof. Let one cycle of joint constraint mappings in line 05 form a mapping $\mathbf{C} \triangleq \prod_{ij \in \mathcal{E}} T_{ij}$. Then $\{\mathbf{C}_k\}_{k=1}^{\infty}$ can be considered as a sequence composed of sufficiently many mappings which are obtained by cycling the sequence of joint constraint mappings $\{T_{ij}\}_{ij \in \mathcal{E}}$. Thanks to Propositions 1 and 2, the sequence $\{\mathbf{C}_k\}_{k=1}^{\infty}$ conforms to Theorem 1, therefore it is convergent. We next write one main loop iteration in the algorithm from lines 03 to 07 as a two-step update method:

$$\underline{\mathbf{y}}_k = \underline{\mathbf{x}}_k + \text{diag}(\underline{\boldsymbol{\kappa}}) (\underline{\mathbf{x}}_k - \underline{\mathbf{x}}_{k-1}) + \text{diag}(\underline{\boldsymbol{\kappa}}) \beta \underline{\mathbf{a}} \quad (46a)$$

$$\underline{\mathbf{x}}_{k+1} = \mathbf{C}_k \underline{\mathbf{y}}_k \quad (46b)$$

with the shape prediction matrix $\text{diag}(\underline{\boldsymbol{\kappa}}) = \text{diag}(\kappa_1 \mathbf{I}_3, \dots, \kappa_n \mathbf{I}_3) \in \mathbb{R}^{3n \times 3n}$ and shape acceleration

vector $\underline{\mathbf{a}} = \underline{\mathbf{f}} / \underline{\mathbf{m}} \in \mathbb{R}^{3n}$. We rewrite equations (46a) and (46b) as the equivalent two-step update:

$$\underline{\mathbf{x}}_{k+1} = C_k \underline{\mathbf{y}}_k \quad (47a)$$

$$\underline{\mathbf{y}}_{k+1} = \underline{\mathbf{x}}_{k+1} + \text{diag}(\underline{\kappa}) (\underline{\mathbf{x}}_{k+1} - \underline{\mathbf{x}}_k) + \text{diag}(\underline{\kappa}) \beta \underline{\mathbf{a}} \quad (47b)$$

By substituting equation (47a) into equation (47b), we obtain:

$$\underline{\mathbf{x}}_{k+1} = C_k \underline{\mathbf{y}}_k \quad (48a)$$

$$\underline{\mathbf{y}}_{k+1} = \text{diag}(\mathbf{1} + \underline{\kappa}) C_k \underline{\mathbf{y}}_k - \text{diag}(\underline{\kappa}) \underline{\mathbf{x}}_k + \text{diag}(\underline{\kappa}) \beta \underline{\mathbf{a}} \quad (48b)$$

with $\text{diag}(\mathbf{1} + \underline{\kappa}) = \text{diag}((1 + \kappa_1) \mathbf{I}_3, \dots, (1 + \kappa_n) \mathbf{I}_3) \in \mathbb{R}^{3n \times 3n}$. We rewrite equations (48a) and (48b) in matrix form as:

$$\begin{bmatrix} \underline{\mathbf{y}}_{k+1} \\ \underline{\mathbf{x}}_{k+1} \\ \beta \underline{\mathbf{a}} \end{bmatrix}_{\underline{\mathbf{z}}_{k+1}} = \begin{bmatrix} \text{diag}(\mathbf{1} + \underline{\kappa}) C_k & -\text{diag}(\underline{\kappa}) & \text{diag}(\underline{\kappa}) \\ C_k & 0 & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{y}}_k \\ \underline{\mathbf{x}}_k \\ \beta \underline{\mathbf{a}} \end{bmatrix}_{\underline{\mathbf{z}}_k} \quad (49)$$

where $\underline{\mathbf{z}} \in \mathcal{M} \times \mathcal{M} \times \mathbb{R}^{3n}$ is an augmented state vector. Equation (49) can be rewritten as:

$$\begin{bmatrix} \underline{\mathbf{y}}_{k+1} \\ \underline{\mathbf{x}}_{k+1} \\ \beta \underline{\mathbf{a}} \end{bmatrix}_{\underline{\mathbf{z}}_{k+1}} = \underbrace{\begin{bmatrix} \text{diag}(\mathbf{1} + \underline{\kappa}) & -\text{diag}(\underline{\kappa}) & \text{diag}(\underline{\kappa}) \\ \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix}_A}_{\mathbf{A}} \underbrace{\begin{bmatrix} C_k & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix}_B}_{\mathbf{B}} \begin{bmatrix} \underline{\mathbf{y}}_k \\ \underline{\mathbf{x}}_k \\ \beta \underline{\mathbf{a}} \end{bmatrix}_{\underline{\mathbf{z}}_k} \quad (50)$$

The last expression can be rewritten by expanding the mapping \mathbf{B} as:

$$\underline{\mathbf{z}}_{k+1} = \mathbf{A} \left(\prod_{ij \in \mathcal{E}} \text{diag}(\mathbf{T}_{ij}, \mathbf{I}, \mathbf{I}) \right) \underline{\mathbf{z}}_k \quad (51)$$

where the identity matrix $\mathbf{I} \in \mathbb{R}^{3n \times 3n}$ is by definition asymptotically regular and satisfies the generalized contractiveness condition (34) with $\gamma \in [0, \frac{1}{3}]$. The linear mapping \mathbf{A} has eigenvalues $\underline{\kappa}$, repeated 3 times, and 1, repeated $6n$ times. If $\underline{\kappa} \in (\mathbf{0}, \mathbf{1})$, then \mathbf{A} is also asymptotically regular, satisfies the generalized contractiveness condition (34) with $\gamma \in [0, \frac{1}{3}]$, and makes the algorithm take larger steps by extrapolation. Consequently, the augmented state sequence $\{\underline{\mathbf{z}}_k\}_{k=1}^\infty$ conforms to Theorem 1, and is therefore convergent. Since each $\underline{\mathbf{x}}$ in $\underline{\mathbf{z}}$ is obtained from a C mapping, then $\underline{\mathbf{x}}$ is a cluster fixed-point of the $\{\mathbf{T}_{ij}\}_{ij \in \mathcal{E}}$ mappings. \square

7 Experimental Results

We conduct three different types of experiments with **Particle-SfT**. First, we reconstruct isometric surfaces and compare the results with state-of-the-art methods. Second, we reconstruct elastic surfaces and again compare the results with state-of-the-art methods. Finally, we exploit the known gravity vector to reconstruct the shape of an object which is deformed by gravitational forces. In all experiments, we compute the 3D error in terms of RMS between the reconstructed shape $\underline{\mathbf{x}}$ and the ground truth shape $\underline{\mathbf{x}}^*$:

$$RMS(\underline{\mathbf{x}}, \underline{\mathbf{x}}^*) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}_i^*\|^2} \quad (52)$$

Synthetic data experiments are repeated four times and their 3D errors are averaged.

7.1 Isometric Surfaces

General points. In order to reconstruct isometric shapes with **Particle-SfT**, we have two sets of parameters to choose, the A-constraint and B-constraint strengths. Since there should be no stretching in the isometric case, we set all the A-constraint strengths as $s_A = 1$, and all the B-constraint strengths used for bending as $s_B = 0.99$. We compare the results of **Particle-SfT** with (Bartoli et al., 2015; Brunet et al., 2014; Chhatkuli et al., 2014; Östlund et al., 2012; Salzmann and Fua, 2011). The compared SfT methods can be grouped in two categories: four initializing methods **Chhatkuli14** (Chhatkuli et al., 2014), **Bartoli12i** (Bartoli et al., 2015), **Ostlund12** (Östlund et al., 2012), **Salzmann11** (Salzmann and Fua, 2011) and one refining method **Brunet10** (Brunet et al., 2014). We recall that **Particle-SfT** is a refining method, but we will see that in practice it does not need an initial guess to estimate the shape, as opposed to **Brunet10** which requires an initial guess close to the optimal shape. We start **Particle-SfT** from the known template shape in all experiments. We also remark that all the state-of-the-art isometric SfT methods, except **Salzmann11**, have at least two parameters to be tuned (*e.g.*, the smoothness weight).

Synthetic Data [developable surface] We simulate 8 different isometric deformations of a flat template. We then generate synthetic images of size 640×480 pixels (see figure 6) of these deformations using a pinhole camera model with focal length 500 pixels and principal point at (320, 240).

Noise and number of correspondences. In order to test the robustness to noise of the SfT methods, we first randomly generate 100 point correspondences between the template and the image, and inject different levels of Gaussian noise to the positions of the image points with standard deviation σ varying from 0 to 2.4 pixels with a step of 0.2 pixels. We then run the SfT methods on these noisy data. The left-most graph

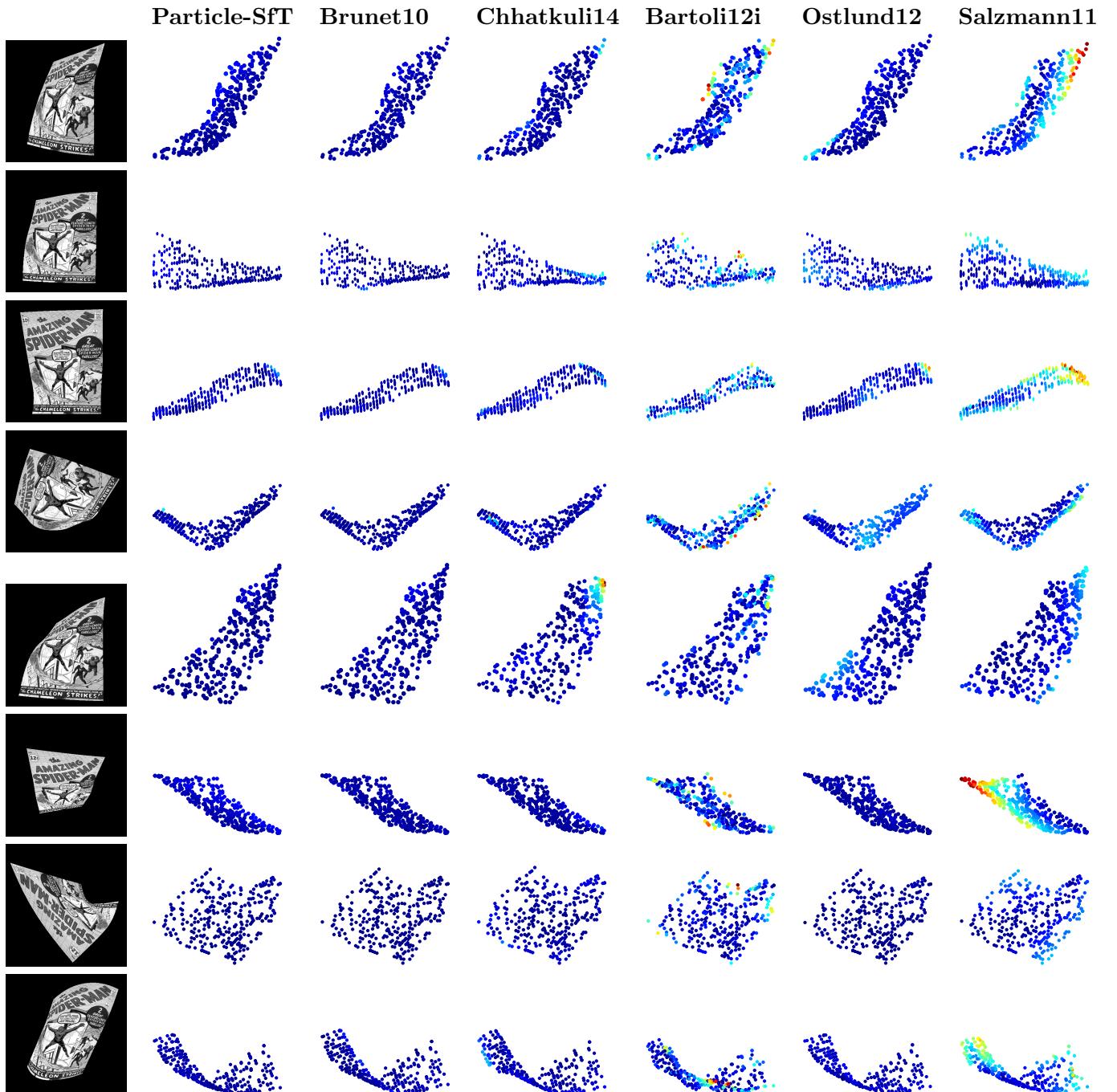


Figure 6: Synthetic images of 8 isometric deformations and the 3D reconstructions seen from a different viewpoint. In this figure the reconstructions are performed with 300 random point correspondences under a 1 pixel Gaussian noise. The colder a point's color, the smaller its reconstruction error.

in figure 7 shows the 3D errors of the SfT methods versus noise level. We see that **Particle-SfT** yields the most accurate results with **Brunet10** under different noise levels, and outperforms the other methods.

Second, we randomly generate N point correspondences varying from 50 to 300 points with a step of 50 points between the template and the image. We also inject Gaussian noise to the positions of these image points with standard deviation $\sigma = 1$ pixel. We then run the SfT methods on these data. The right-most graph in figure 7 shows the 3D errors of the SfT methods versus the number of point correspondences. We see that **Particle-SfT** yields the most accurate results again with **Brunet10** under different numbers of correspondences, and outperforms the other methods.

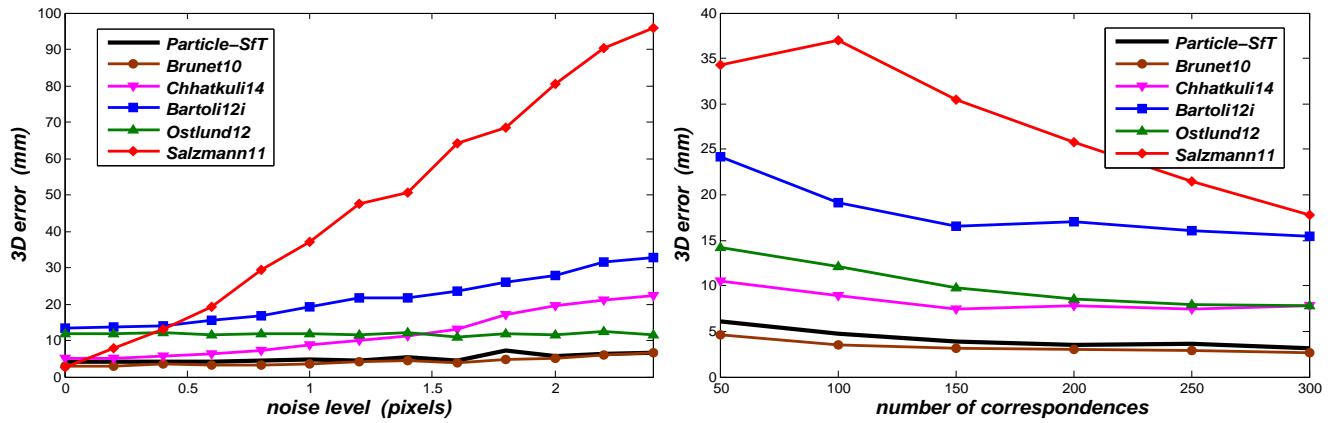


Figure 7: 3D errors of isometric SfT methods on synthetic data versus Gaussian noise levels (left) and number of correspondences (right).

Real Data [paper sheet] We use the CVLab’s paper dataset (Varol et al., 2012). This dataset consists of 191 images of a paper sheet being deformed. The images are taken with a fixed Kinect while the paper sheet is being moved and deformed. Some 1,300 point features are detected per image. The left-most graph in figure 8 compares the SfT methods in terms of 3D error versus image number. Table 1 gives the mean 3D error values and run times of the compared methods. We observe from figure 8 and table 1 that **Particle-SfT** outperforms the other methods in most of the frames, and yields the lowest mean 3D error over the whole CVLab’s paper dataset. **Particle-SfT** reconstructs shapes about 15 times faster than the refining method **Brunet10**.

Convergence basin. We compare **Brunet10** with **Particle-SfT** in terms of convergence basin using the CVLab’s paper dataset. To do so, we generate initial guesses by perturbing the known ground-truth shapes. Afterwards we let **Brunet10** and **Particle-SfT** start from these guesses. A 1% perturbation level corresponds to 1° degree of rotation about a random axis, to a displacement along the same random axis with length 1% of the distance from the center of gravity of the ground-truth shape to the camera and to

	Initializing methods				Refining methods	
	Chhatkuli14	Bartoli12i	Ostlund12	Salzmann11	Brunet10	Particle-SfT
error (mm)	3.97	4.56	4.82	7.47	3.86	3.78
time (sec)	0.50	0.15	6.00	1.10	22.33	1.50

Table 1: Mean errors and run times for CVLab’s paper dataset.

a Gaussian noise with standard deviation 1 mm on the position coordinates. We increase the perturbation level from 0% to 100% with a step of 5 percentage points per shape. We repeat this experiment three times over the whole CVLab’s paper dataset. The right-most graph in figure 8 shows the mean 3D errors versus perturbation level. We observe that **Brunet10** misses the solutions after 25% perturbation on the ground-truth shapes whereas **Particle-SfT** still finds the expected solutions. The mean standard deviation of the errors at convergence is computed for **Brunet10** as 215.44 mm and for **Particle-SfT** as 0.95 mm.

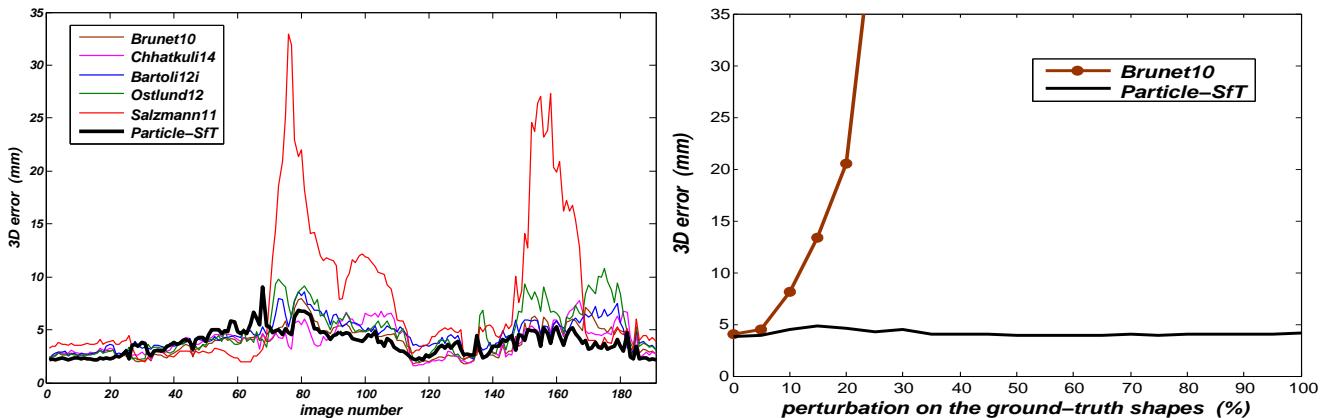


Figure 8: 3D errors of isometric SfT methods on the CVLab’s paper dataset (left), mean 3D errors of isometric refining SfT methods versus perturbation level on the ground-truth shapes of CVLab’s paper dataset used as initial guesses (right).

Real Data [cushion with non-planar template] **Particle-SfT** also works with non-planar templates. We apply **Particle-SfT** to a cushion dataset whose template is non-planar. This dataset contains images of 4 different deformations of a cushion, the cushion’s non-planar 3D template and the template image. The deformations are quasi-isometric on the surface of the cushion. The images are $3,456 \times 2,304$ pixels. The focal length of the camera is about 2,700 pixels. Some 1,000 feature correspondences are detected between the template image and an image of a deformation by combining SIFT (Lowe, 1999) and KAZE (Alcantarilla et al., 2012). In figure 9, the first column shows the template image and its non-planar 3D template, and the rest of the columns show the input images, the 3D reconstructions (black circles) over the ground-truth shapes (green dots) and the 3D errors obtained by applying **Particle-SfT** directly with the non-planar

3D template. The results for the compared SfT methods can be found in table 2. We observe in table 2 that for the mean 3D errors over the whole cushion dataset **Particle-SfT** yields the second best result, after **Chhatkuli14**. **Particle-SfT**'s result is very close to **Chhatkuli14**, less than half a millimeter. Both **Chhatkuli14** and **Bartoli12i** need flattening of the non-planar 3D template. **Chhatkuli14** uses point correspondences and requires the first-order differential structure around these points.

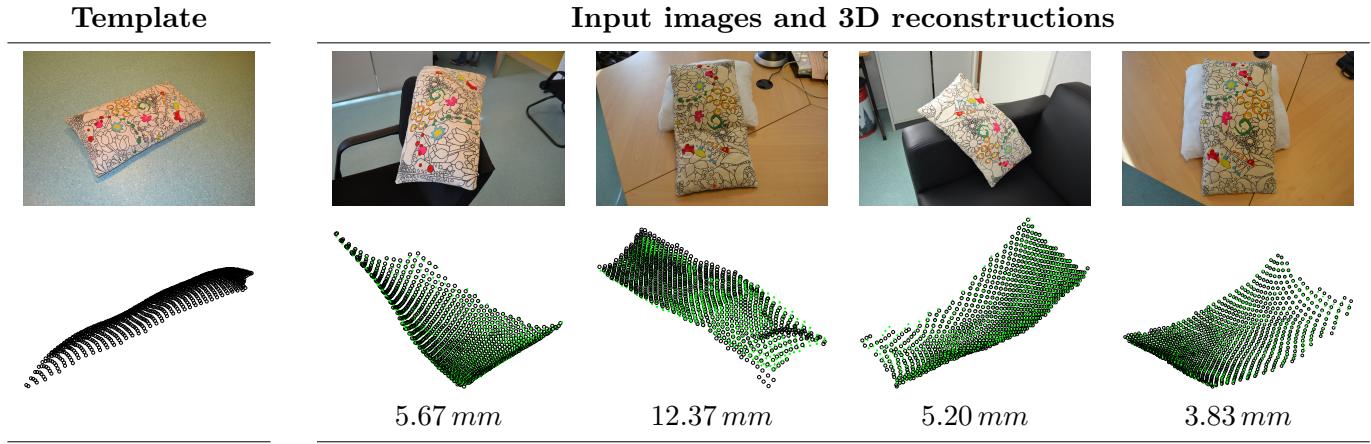


Figure 9: Cushion dataset with non-planar template. The 3D reconstructions (black circles) of **Particle-SfT** are shown with the ground-truth shapes (green dots). The 3D errors are also given below the reconstructions.

	Initializing methods				Refining methods	
	Chhatkuli14	Bartoli12i	Ostlund12	Salzmann11	Brunet10	Particle-SfT
cushion #1	4.25	10.52	5.02	7.08	5.49	5.67
cushion #2	10.48	17.21	9.76	12.99	17.32	12.37
cushion #3	4.74	19.41	4.70	7.69	7.89	5.20
cushion #4	6.07	13.16	12.46	8.36	4.61	3.83
mean	6.39	15.08	7.99	9.03	8.83	6.76

Table 2: 3D errors in millimeters for the cushion dataset.

Remark. **Brunet10** is initialized with **Bartoli12i** in all our experiments. The results are similar in accuracy and speed of convergence when **Brunet10** is initialized with **Chhatkuli14**. This was shown in the results of (Chhatkuli et al., 2014). The **Bartoli12i+Brunet10** method corresponds to the ReD method and the **Chhatkuli14+Brunet10** method corresponds to the ReJ method in (Chhatkuli et al., 2014).

7.2 Elastic Surfaces

General points. In order to reconstruct elastic shapes with **Particle-SfT**, we set the distance correction strengths as $0 < s_A < 1$ and $0 < s_B < 1$. All the A-constraint strengths s_A use the same value for a

given material. Similarly all the B-constraint strengths s_B use the same value for a given material but might be different to what A-constraints use. In practice, the values of s_A and s_B were found on test data and included as known parameters in the template. We need to add boundary conditions into the deformation constraints to resolve the unknown shape. In our case, these boundary conditions are known points \mathbf{x}_i^* of the ground-truth shape as for the other elastic SfT methods (Haouchine et al., 2014; Malti et al., 2013, 2015). Those known points are integrated as deformation constraints by setting the position of the corresponding particles simply as $\mathbf{x}_i = \mathbf{x}_i^*$ in each iteration. We compare the results of **Particle-SfT** with (Bartoli et al., 2015; Haouchine et al., 2014; Malti et al., 2013, 2015). The first elastic SfT method, **Haouchine14** (Haouchine et al., 2014), is based on non-linear elasticity and the second elastic SfT method, **Malti13** (Malti et al., 2013), on linear elasticity. They both minimize a non-convex energy function. **Haouchine14** and **Malti13** are refining elastic SfT methods. The third elastic SfT method, **Malti15** (Malti et al., 2015), is based on linear elasticity and minimizes a convex energy function. **Malti15** is an initializing elastic SfT method. **Malti15** uses reprojection boundary constraints and is designed to work with a rigid frame (undeformed border points) around the shape. We adapt **Malti15** to work with the known 3D boundary points of the ground-truth shape and with a non-rigid frame around the shape. The fourth SfT method, **Bartoli12c** (Bartoli et al., 2015), is for conformal deformations and yields a set of analytic solutions up to scale which are later refined through non-linear optimization. **Bartoli12c** does not need boundary constraints, however its solutions need to be correctly scaled with respect to the ground-truth shape for comparison purposes. We choose the best solution from the solution set of **Bartoli12c**.

All the state-of-the-art elastic SfT methods use the material's Young modulus and/or Poisson's ratio. The parameters of all the compared SfT methods are tuned by trial and error to give the best results. The ground-truth shapes of real data experiments are obtained using the PhotoScan software. PhotoScan reconstructs the 3D scene from multiple images using Structure-from-Motion.

Synthetic Data [partially stretched surface] We use Blender to simulate a rubber like material. The template shape is a rectangular flat surface and its size is 300×400 mm. We partially stretch this surface from its longest side in a continuous manner up to 66% extensibility ratio as shown in figure 10. We sequentially pick 50 ground-truth shapes from this continuous deformation. We then generate synthetic images of size $1,024 \times 1,024$ pixels of these ground-truth shapes using a pinhole camera model with focal length 770 pixels and principal point at (512, 512). In order to reconstruct the shapes, we choose 20 boundary points where the pulling forces are applied on the deformed ground-truth shapes. Figure 11 shows the reconstructed shapes from the noisy synthetic image with extensibility ratio 66% from figure 10. We observe that **Particle-SfT** yields a better reconstructed shape than the other methods. **Malti13** and **Haouchine14** give similar

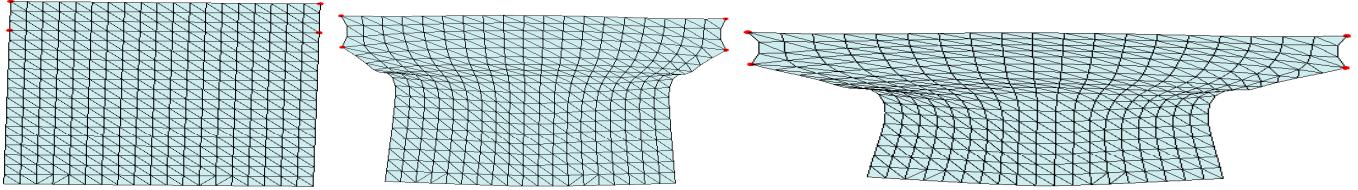


Figure 10: Partially stretched elastic material. Three frames (#1, #25, #50) from a continuous deformation sequence. From left to right the extensibility ratios are 0%, 33%, 66%. This data can be downloaded from <http://isit.u-clermont1.fr/~ab/Research>.

results, less accurate at the lower-right corner of the shape compared to **Particle-SfT**. **Bartoli12c** gives a worse result, since it can only handle conformal deformations but not general elastic ones. **Malti15** gives the worst result, since it is a convex approximation with a linear elasticity model of a non-convex problem with nonlinear behavior.

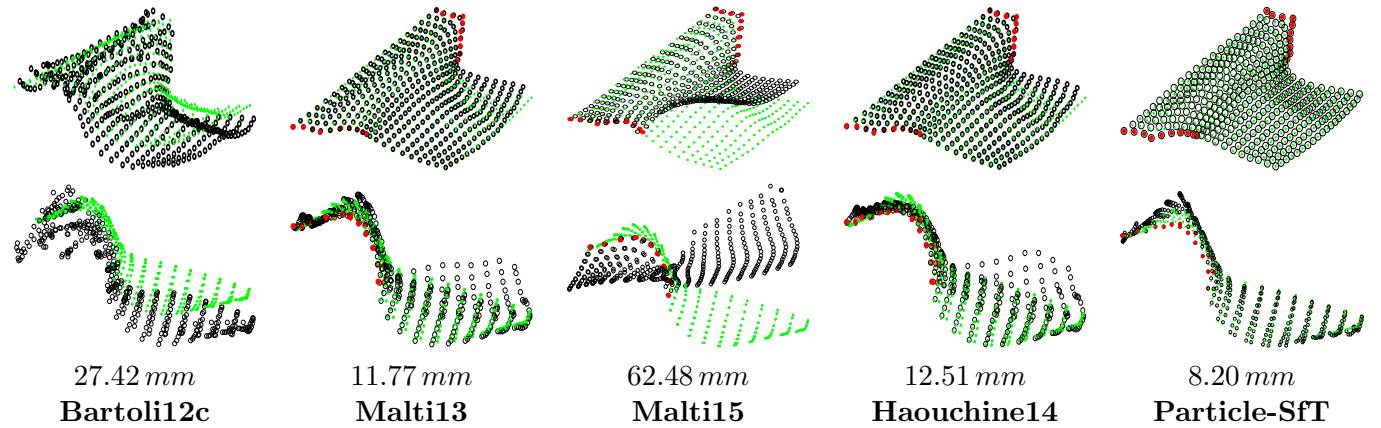


Figure 11: Reconstructed shapes (black circles) using 20 boundary points (red dots) for the ground-truth shape (green dots) with extensibility ratio 66%. Each column shows the same reconstructed shape from two different viewpoints, and gives the 3D errors in millimeters (mm).

Noise and number of correspondences. We test the robustness of the shape reconstruction methods against three varying parameters: extensibility ratio, noise level and number of correspondences. The extensibility ratio varies from 0% to 66% with a step of about 1.32%. The standard deviation σ of Gaussian noise varies from 0 to 2 pixels with a step of 0.2 pixels. Correspondences are selected randomly. The number of correspondences varies from 50 to 300 points with a step of 50 points. While one of the parameters is varied, the other two are kept fixed. Fixed default values are 66% extensibility ratio, 100 random correspondences and $\sigma = 1$ pixel. Figure 12 shows the 3D errors of the SfT methods versus extensibility ratio, Gaussian noise level and number of correspondences. Table 3 lists the mean of 3D errors and run times of the compared methods on the varying extensibility ratio experiments whose results are shown in figure 12. We observe from figure 12 and table 3 that **Particle-SfT** yields about 3 times more accurate results than the other methods in almost all configurations except at very small extensibility ratios. On less deformed surfaces all

methods perform with about the same order of accuracy.

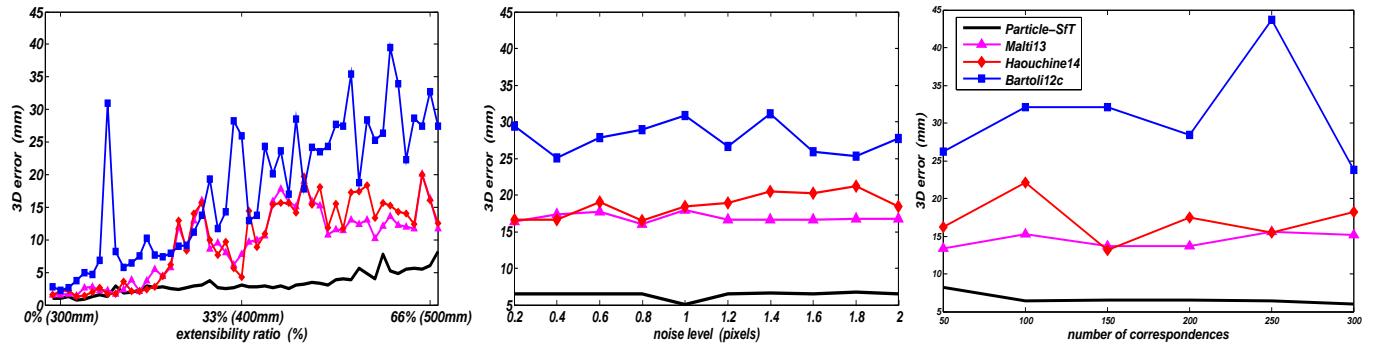


Figure 12: 3D errors of elastic SfT methods versus extensibility ratio, Gaussian noise level, and number of correspondences. The 3D error for **Malti15** is off the graphs’ axis limits.

Initializing methods		Refining methods		
Bartoli12c	Malti15	Haouchine14	Malti13	Particle-SfT
error (mm)	17.88	51.07	10.02	9.46
time (sec)	580.47	1.18	10.90	121.80

Table 3: Mean error and run time for the varying extensibility ratio results of figure 12. The run time for **Bartoli12c** was extremely high due to the built-in refinement of its multiple solutions (Bartoli et al., 2015).

Change of boundary locations. We reconstruct the shape which has extensibility ratio 66% seen in figure 10 from the noisy synthetic image. We choose, this time, the 20 boundary points on the lateral sides of the deformed ground-truth shape where no pulling forces are applied. For all elastic methods, except **Malti15**, the reconstructions are now worse than the previous case where the boundary points are chosen about the locations of the pulling forces. Figure 13 shows the results of **Bartoli12c**, **Malti13**, **Malti15**, **Haouchine14** and **Particle-SfT** for this new case. Since **Bartoli12c** does not need boundary conditions, its result is the same as in figure 11 and here becomes the most accurate solution when no strong priors on boundary conditions are exploited. However remark that **Bartoli12c** needs the correct scale with respect to the ground-truth shape and one should know how to choose the best solution from its multiple solutions. We observe that **Particle-SfT** yields once more a better reconstructed shape than the other elastic methods. However the difference in results is less significant because the new boundary locations introduce ambiguities for all elastic methods.

Real Data [foot] We use a sock for foot size 37 as a template. We then put this sock on a foot whose size is 41, and take its image to be used as input for SfT. In figure 14, the left box shows the template and the input images of this sock. We match 137 point correspondences between the template image and the

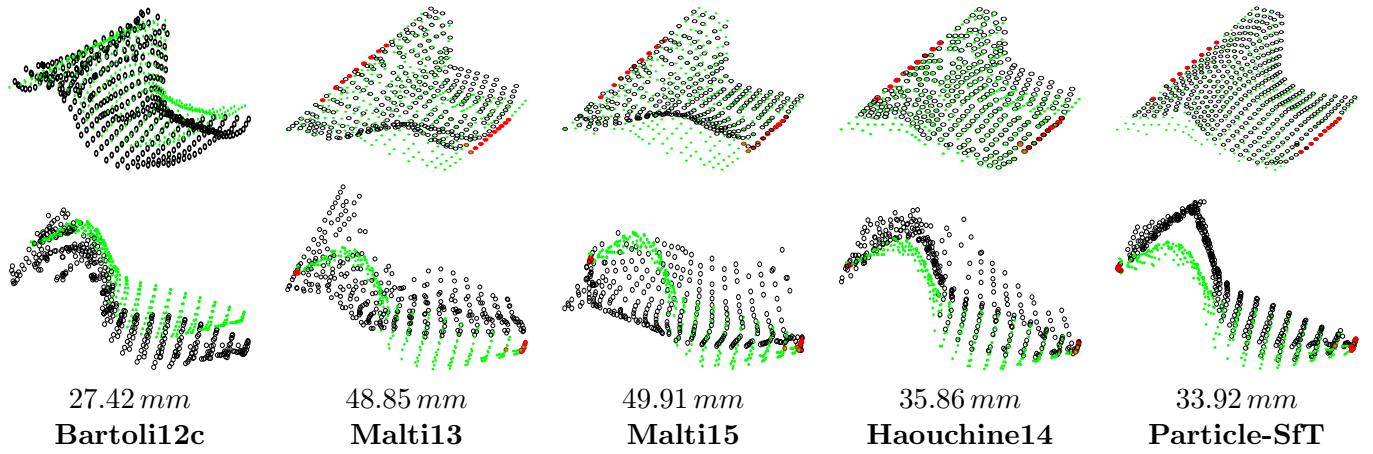


Figure 13: Reconstructed shapes (black circles) with extensibility ratio 66% and the ground-truth shapes (green dots). The 20 boundary points (red dots) are chosen on the lateral sides of the ground-truth shape where no pulling forces are applied. Each column shows the same reconstructed shape from two different viewpoints, and gives the 3D errors in millimeters (mm).

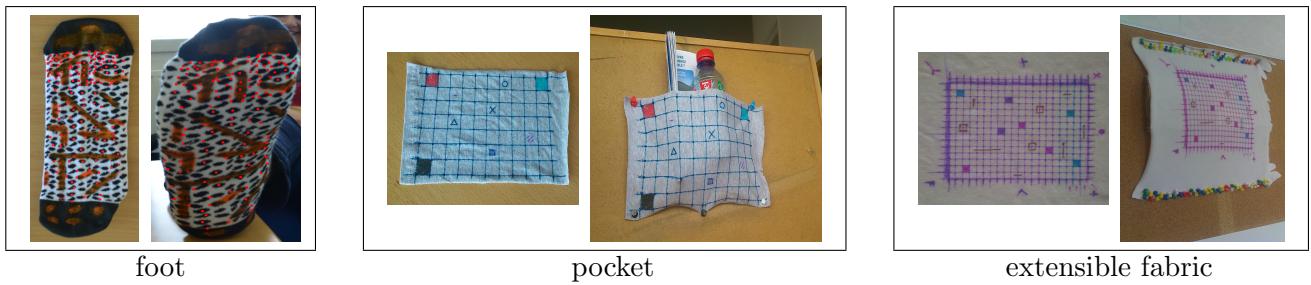


Figure 14: Real elastic datasets. In each box, the left image shows the template image and the right image shows the input image used in SfT.

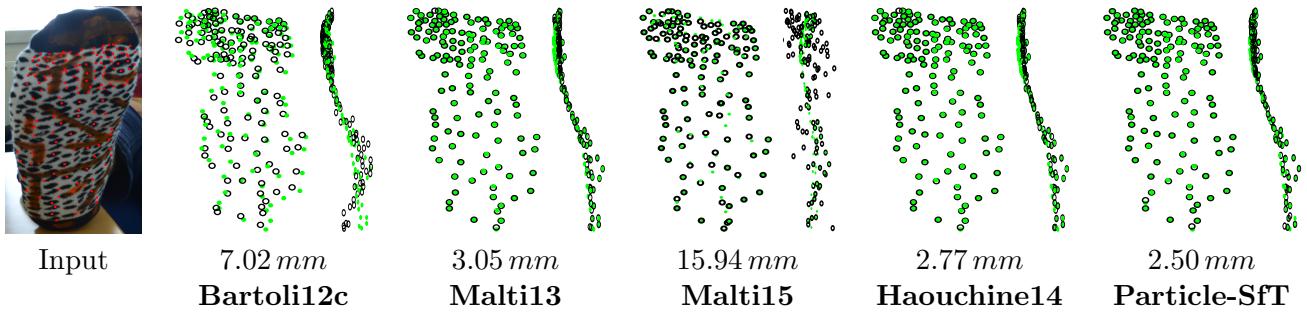


Figure 15: SfT results for an underfoot surface reconstruction of foot size 41 wearing a sock for foot size 37. The first column shows the input image. Red markers on the input image show the point correspondences used in SfT. The last 5 columns show the reconstructed shapes (black circles) on the ground-truth shapes (green dots) with front and side views. The 3D reconstruction errors are also given for each method.

input image. We choose 37 of the 137 points as the known boundary conditions from the border points of the ground-truth shape. Figure 15 shows the reconstructed shapes of the compared methods and lists the 3D errors. We observe that all the refining methods perform better than the initializing methods, and that the performances of refining methods are similar in accuracy. However, **Particle-SfT** outperforms all the other methods.

Real Data [pocket] We use a piece of elastic cloth, whose size is 20×14 cm, to make a pocket for holding objects. The cloth is fixed on a cork board with 5 pins, where it was laid flat as in the template image shown in figure 14, middle. We then insert a bottle of water and a magazine in the pocket. The distances between the fixing pins before the objects were put inside and after, are the same. The elastic deformation on the pocket due to inserted objects is stronger at the borders between the pins. The elastic deformation on the surface, where the grid is drawn, is moderate. The pocket grid contains 88 points. We use these 88 points as correspondences for reconstruction. 25 boundary points obtained from the left, bottom and right sides of the grid, are used as boundary constraints. Figure 16 shows the reconstructed shapes for the compared methods and lists the 3D errors. We observe that again all the refining methods perform better than the initializing methods, and their performances are similar in accuracy. Once more, **Particle-SfT** outperforms the other methods.

Real Data [extensible fabric] We draw a regular grid on a piece of extensible fabric. The size of the grid is 15×20 cm. We then pin this fabric from its upper and lower parts down to a fixed cork board, and insert an object between the fabric and the board to create extension and curvature, as shown in figure 14, right. The grid extension is 33 % along the vertical axis between the upper and lower parts. This extension causes shrinking of 20 % of the grid along the lateral axis. The grid contains 336 points. We use these 336 points as correspondences for reconstruction. 42 of them are used as boundary constraints. These 42

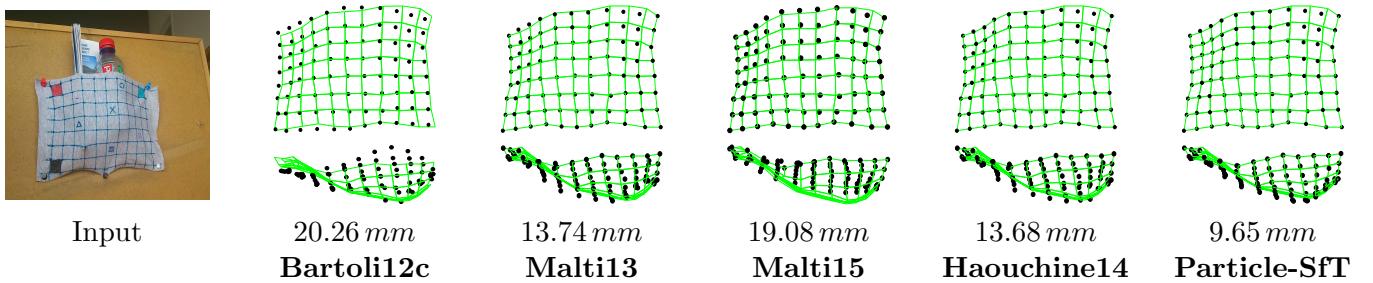


Figure 16: SfT results for the pocket data. The input image shows a pocket filled by a bottle of water and a magazine. The last 5 columns show the reconstructed shapes (black dots) on the ground-truth shapes (green mesh) with front and bottom views. The 3D reconstruction errors are also given for each method.

boundary points are located on the upper and lower edges of the grid. Figure 17 shows the reconstructed shapes of the compared methods and lists the 3D errors. We observe that again all the refining methods perform better than the initializing methods, and their performances are similar in accuracy. This time, **Particle-SfT** yields the second best result, after **Haouchine14**.

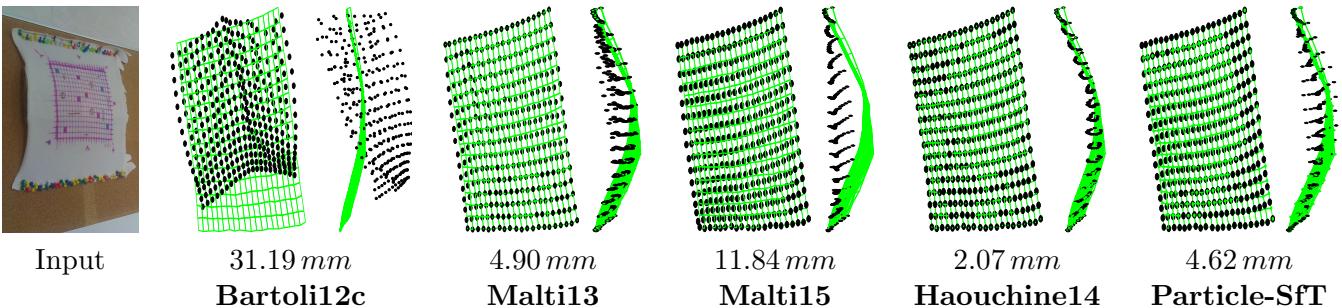


Figure 17: SfT results for a piece of fabric which has large lateral stretching and a mild central pushing deformation. The left most column shows the deformed fabric. The last 5 columns show the reconstructed shapes (black dots) on the ground-truth shapes (green mesh) with front and side views. The 3D reconstruction errors are also given for each method.

As a general conclusion, we observe from figures 15, 16 and 17, that the refining methods **Malti13**, **Haouchine14** and **Particle-SfT** reconstruct shapes quite well. The extensibility ratio results with synthetic data in figure 12 also confirm that conclusion. In synthetic data results, we see that **Particle-SfT** outperforms the other methods. In figures 15 and 16, in real data results, **Particle-SfT** also outperforms the other methods and improves the results of other refining elastic methods of at least 15%. Only in figure 17 **Particle-SfT** yields the second best result, after **Haouchine14**.

Convergence basin. We compare the refining methods **Malti13** and **Haouchine14** with **Particle-SfT** in terms of convergence basin using the simulated rubber like material shown in figure 10 with extensibility ratio 66% and using the real elastic datasets (foot, pocket, extensible fabric) shown in figure 14. For real elastic datasets, we use the same correspondences and boundary points as for the above elastic SfT experi-

ments. For the simulated rubber, we take its synthetic image and randomly select 120 point correspondences between the template and the image. We also add Gaussian noise with $\sigma = 1$ pixel to the image points. We choose 20 of the 120 points as the boundary points located exactly at the same places as shown in figure 11. We then generate initial guesses by perturbing the ground-truth shapes of the real elastic datasets and the simulated rubber. Afterwards we let **Malti13**, **Haouchine14** and **Particle-SfT** start from these guesses. A guess with 1% perturbation level is equal to the ground-truth shape multiplied by a scale factor 1. It is multiplied by 10 for 10% perturbation level and so on. We increase the perturbation level from 1% to 100% with a step of 10%. Figure 18 shows the datasets used in the convergence basin experiment and the mean 3D error of SfT methods versus perturbation level on the ground-truth shapes used as initial guesses. We observe that **Haouchine14** and **Particle-SfT** come back to the expected solutions, while **Malti13** starts diverging after 30% perturbation level.

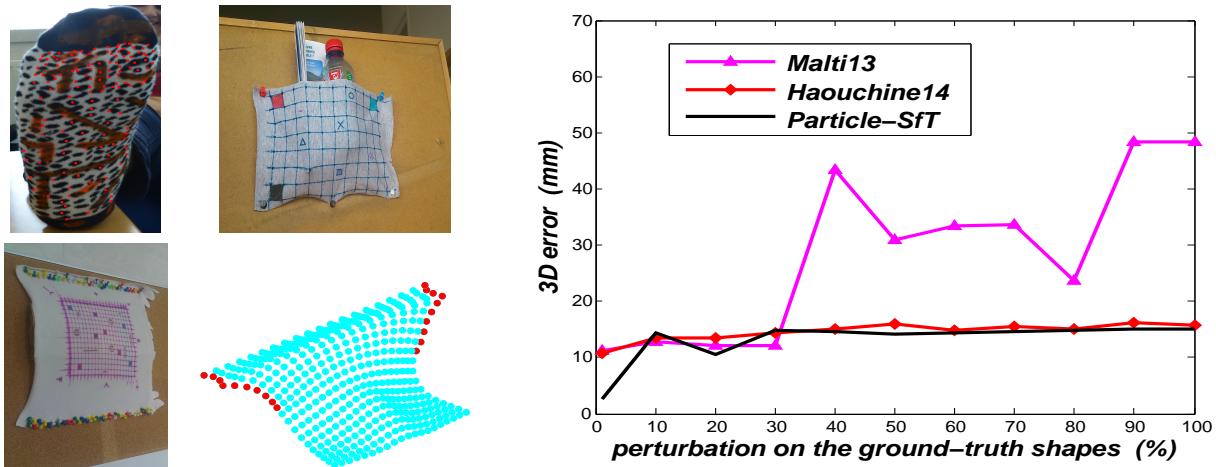


Figure 18: 3D errors of SfT methods versus perturbation on the ground-truth shapes used as initial guesses.

We repeat this experiment for another deformed shaped. This time, the rubber like material is pushed up from below by a cube by about 200 mm with its corners kept fixed. We take a synthetic image of this ground-truth shape and randomly select 200 point correspondences between the template and the image. We also add Gaussian noise with $\sigma = 1$ pixel to the image coordinates. We choose 35 of 200 points as the boundary points which correspond to the fixed corners of the rubber material and to the border contact points between the pushing cube and the rubber material. We generate initial guesses by perturbing this ground-truth shape. Afterwards we let **Malti13**, **Haouchine14** and **Particle-SfT** start from these guesses. A guess with 1% perturbation level is equal to the ground-truth shape multiplied by a scale factor 1.01. It is multiplied by 1.02 for 2% perturbation level and so on. We increase the perturbation level from 0% to 100% with a step of 1%. The graph in figure 19 shows the 3D errors versus perturbation level for this inner pushing. We observe that **Malti13** and **Haouchine14** miss the solution after 10% perturbation on

the ground-truth shape whereas **Particle-SfT** still finds the expected solution. This is a special case where **Haouchine14** cannot refine the initial solution in any case, while **Malti13** can only refine it for very small perturbations. The low error observed at the beginning of the error graph in figure 19 for **Haouchine14** is because of the initial guess being equal to the exact ground-truth shape without any perturbation.

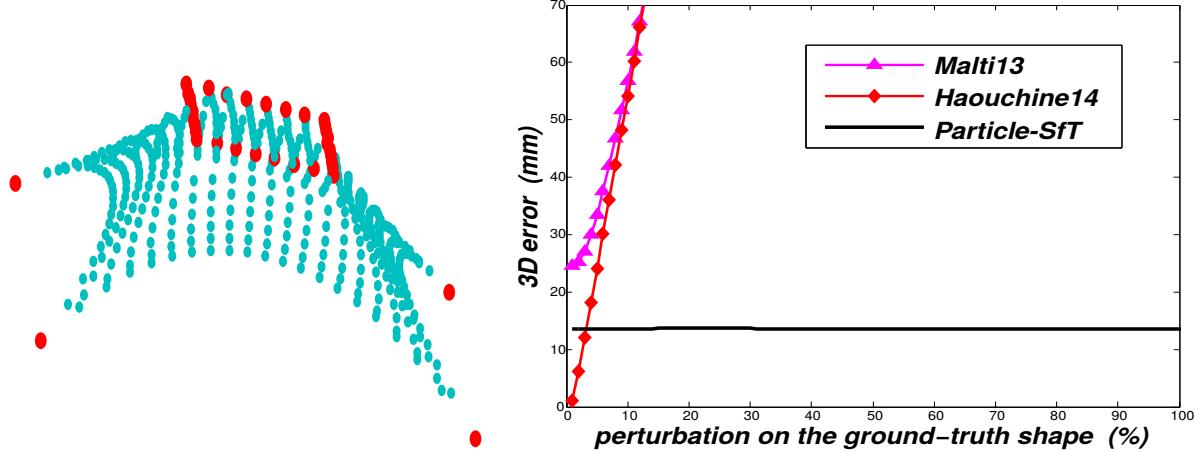


Figure 19: 3D errors of SfT methods versus perturbation on the ground-truth shape used as an initial guess.

7.3 Hidden Parts from Gravity

In gravity based experiments, we consider objects with textureless, hidden or occluded parts in the image. We assume that these parts are deformed by the gravitational force. Our objective is then to reconstruct the full shape of the object from its partly textureless/hidden/occluded image and the known gravity vector. We use a smartphone’s camera to take the input image of the deformed object and the smartphone’s accelerometer to measure the direction of the gravity vector. We achieve the full shape reconstruction by setting the force vector in **Particle-SfT** to the gravity vector for all the textureless/hidden/occluded particles. The reconstruction errors in the experiments below may be explained by the particle model not modeling the deformed object perfectly and the noise on the gravity vector measured by the accelerometer. **Particle-SfT** is the first SfT method which allows one to use gravity.

Gravity experiment 1. In this scenario, we have a piece of cloth laid on a table with a visible part and a hidden part in the image. The hidden part is deformed by the gravitational force. The deformation correction strengths are chosen as $s_A = 1$ and $s_B = 0.2$. Figure 20 shows this experimental scenario. The cloth’s size is 30×30 cm. An 11 by 11 regular grid is sketched onto the surface of the cloth. In the input image we use the visible 77 grid points as point correspondences between the template and input image. Figure 20 contains respectively the template shape of the cloth, the input image, the fully reconstructed shape and an image revealing the whole cloth. We also reconstruct the ground-truth shape from the right

image shown in figure 20 and compare it to the shape reconstructed from the input image. The 3D error between the corresponding ‘hidden part’ of the ground-truth shape and the reconstructed shape is found as 8 mm. This corresponds to a 2.6% error with respect to the size of the cloth.

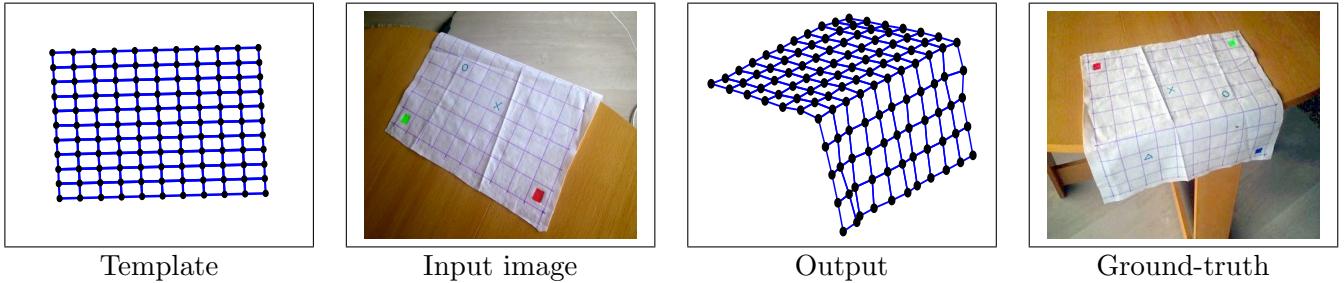


Figure 20: Particle-SfT recovers the hidden part using gravity as measured by a smartphone.

Gravity experiment 2. In this scenario, we have a piece of cloth hanged on a laundry holder. The cloth has a visible part and an occluded part in the image. The occluded part is deformed by the gravitational force. The deformation correction strengths are chosen as $s_A = 1$ and $s_B = 0.1$. Figure 21 shows this experimental scenario. The cloth’s size is 40×20 cm. We have 36 point correspondences between the template and input image. Figure 21 in its first column shows the input image and template, in the second and third columns the reconstructed shape (black mesh) overlaid on the ground-truth shape (green mesh) without gravity and with gravity from two different viewpoints, and in the last column an image revealing the whole cloth and its reconstructed shape which is used as ground-truth shape for comparison. We compare the ground-truth shape with the shapes reconstructed without and with gravity from the occluded input image. The 3D errors are listed below the reconstructions in figure 21. We observe that **Particle-SfT** with gravity reconstructs the correct shape to a good extent but misses it without gravity.

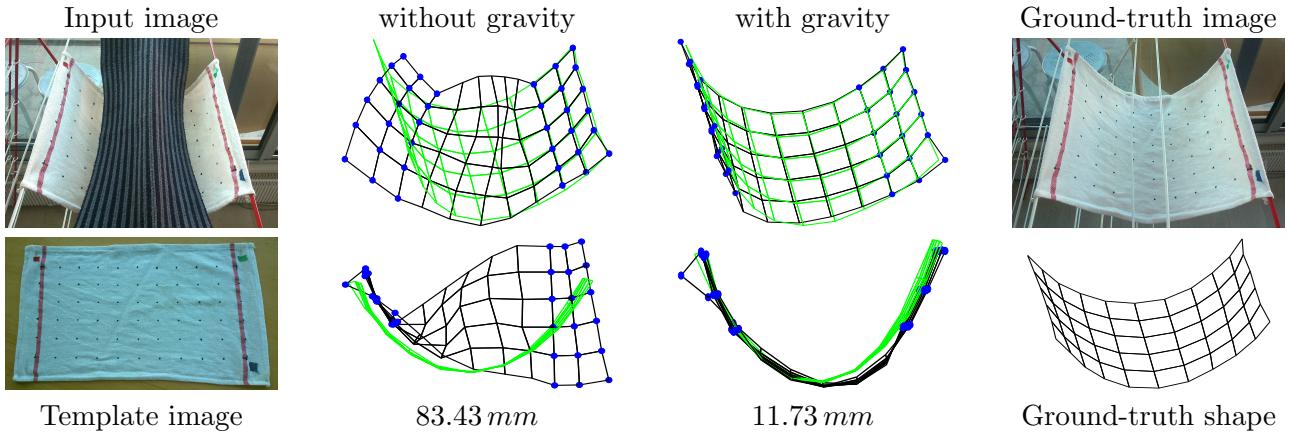


Figure 21: Particle-SfT recovers occluded part using gravity as measured by a smartphone. The second and third columns show the reconstruction (black mesh) overlaid on the ground-truth shape (green mesh) without and with gravity respectively from two different viewpoints.

8 Conclusions

We have proposed Particle-SfT, the first SfT algorithm guided by constraint projections and forces, and capable of reaching a highly accurate solution without an initial guess. Because Particle-SfT has low complexity and is generic in the types of deformation constraints it uses, it may be used in place of any other existing refinement algorithm. Our experimental results show that Particle-SfT reaches the same accuracy that the best performing isometric SfT method while consistently outperforming existing non-isometric SfT methods in almost all cases. The core of Particle-SfT is to enforce constraint subsets independently and optimally, allowing it to converge faster. Particle-SfT is provably convergent.

Remark that the template of Particle-SfT includes more information than previous SfT methods so that it can cope well with both isometric and elastic deformations. We list in table 4 the template’s exact data structure and how each field is obtained. The template’s shape and texturemap are obtained from multiple images of the object at rest using Structure-from-Motion. The density of the material of which the object is made is known from the material’s properties. The object’s surface thickness can be precisely measured beforehand. Only the stretching and bending correction strengths need tuning. We set $s_A = 1$ for the isometric deformations since there is no stretching, and therefore only s_B needs tuning. We tune both s_A and s_B for the elastic deformations. One of the perspectives of our future work is to be able to compute these two correction strengths automatically from the material’s mechanical parameters such as Young’s modulus and/or Poisson’s ratio since those parameters encode the material’s elasticity behavior. Our second perspective is to extend Particle-SfT to volumetric objects, and the third one is to investigate

Data	Shape+Texturemap	Density	Thickness	Stretching s_A	Bending s_B
How are the values obtained?	any rest shape with texture via SfM (PhotoScan software)	known from material’s properties	measured from the object	isometric case: set $s_A = 1$, elastic case: tuned $0 < s_A < 1$	tuned for both isometric and elastic cases $0 < s_B < 1$

Table 4: The template’s data structure in Particle-SfT.

Particle-SfT for objects made of mixed materials, thus having inhomogeneous stiffness.

Acknowledgments. We thank the authors of (Bartoli et al., 2015; Brunet et al., 2014; Chhatkuli et al., 2014; Malti et al., 2015; Östlund et al., 2012; Salzmann and Fua, 2011; Varol et al., 2012) for making their source codes and/or datasets available to us. This research has received funding from the EU’s FP7 through the ERC research grant 307483 FLEXABLE.

References

- R. Agarwal, M. Meehan, and D. O'Regan. Fixed point theory and applications. *Cambridge University Press*, 2001.
- A. Agudo and F. Moreno-Noguer. Simultaneous pose and non-rigid shape with particle dynamics. *CVPR*, 2015.
- A. Agudo, L. Agapito, B. Calvo, and J. Montiel. Good vibrations: a modal analysis approach for sequential non-rigid structure from motion. *CVPR*, 2014.
- A. Agudo, F. Moreno-Noguer, B. Calvo, and J. Montiel. Sequential non-rigid structure from motion using physical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):979–994, 2016.
- P. F. Alcantarilla, A. Bartoli, and A. J. Davison. KAZE features. *ECCV*, 2012.
- A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-Template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, 2015.
- L. Belluce and W. Kirk. Some fixed point theorems in metric and banach spaces. *Canadian Mathematical Bulletin*, 12(4):481–491, 1969.
- L. Brouwer. Über abbildungen von mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1910.
- F. Brunet, A. Bartoli, and R. Hartley. Monocular template-based 3D surface reconstruction: convex inextensible and nonconvex isometric methods. *Computer Vision and Image Understanding*, 125:138–154, 2014.
- A. Chhatkuli, D. Pizarro, and A. Bartoli. Stable template-based isometric 3D reconstruction in all imaging conditions by linear least-squares. *CVPR*, 2014.
- H. Goldstein. Classical mechanics. *Addison-Wesley*, 1980.
- N. Haouchine, J. Dequidt, M. Berger, and S. Cotin. Single view augmentation of 3D elastic objects. *ISMAR*, 2014.
- S. Ilić and P. Fua. Non-linear beam model for tracking large deformation. *ICCV*, 2007.
- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- D. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.

- A. M. Lyapunov. The general problem of the stability of motion (in russian). *Doctoral dissertation, Univ. Kharkov*, 1892.
- A. Malti, R. Hartley, A. Bartoli, and J. Kim. Monocular template-based 3D reconstruction of extensible surfaces with linear elasticity. *CVPR*, 2013.
- A. Malti, A. Bartoli, and R. Hartley. A linear least-squares solution to elastic shape-from-template. *CVPR*, 2015.
- Z. Marton, R. Rusu, and M. Beetz. On fast surface reconstruction methods for large and noisy point clouds. *ICRA*, 2009.
- D. Metaxas. Constrained deformable superquadratics and nonrigid motion tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.
- F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed form. *CVPR*, 2009.
- M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Workshop in Virtual Reality Interactions and Physical Simulation VRIPHYS*, 2006.
- J. Östlund, A. Varol, T. D. Ngo, and P. Fua. Laplacian meshes for monocular 3D shape recovery. *ECCV*, 2012.
- M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*, 95(2):124–137, 2011.
- J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.
- D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.
- X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Graphics Interface, Canad. Inf. Processing Society*, pages 147–154, 1996.
- B. Rhoades. A comparison of various definitions of contractive mappings. *Tran. of the American Mathematical Society*, 226:257–290, 1977.
- M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. *CVPR*, 2009.

- M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.
- M. Salzmann and R. Urtasun. Physically-based motion models for 3D tracking: a convex formulation. *ICCV*, 2011.
- M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. *ECCV*, 2008.
- A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. *CVPR*, 2012.