As-Rigid-As-Possible Volumetric Shape-from-Template

Shaifali Parashar, Daniel Pizarro, Adrien Bartoli, Toby Collins

As-Rigid-As-Possible Volumetric Shape-from-Template

Shaifali Parashar[1], Daniel Pizarro[2,1], Adrien Bartoli[1] and Toby Collins[1]
[1]ALCoV-ISIT, UMR 6284 CNRS / Université d'Auvergne, Clermont-Ferrand, France
[2]Geintra Research Group, Universidad de Alcalá, Alcalá, Spain

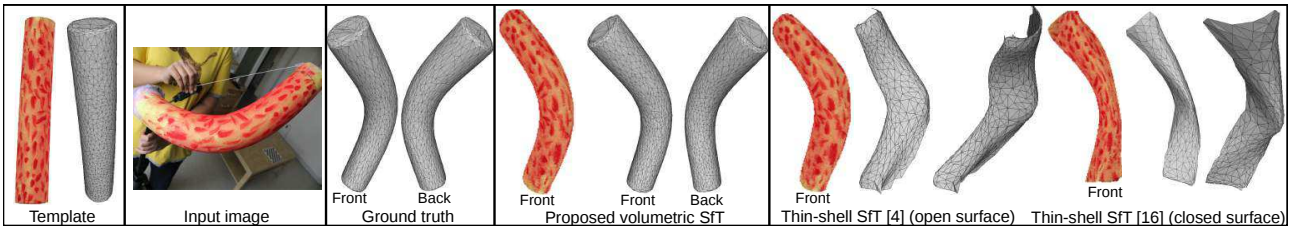

Figure 1: Volumetric SfT versus thin-shell SfT. Existing methods are thin-shell SfT. They use deformation constraints on the object's surface. For instance, [4] uses isometric constraints on the object's visible (front) surface and reconstructs the object partially, while [16] uses isometric constraints on the object's whole closed outer surface and reconstructs it entirely. Volumetric SfT uses deformation constraints on the object's surface *and* interior. This greatly improves reconstruction accuracy and facilitates reconstruction of the object's interior. In this example, the thin-shell SfT methods [4, 16] reach a 3D error of 20 mm and 13 mm respectively on the visible surface, while the proposed volumetric SfT method reaches a 3D error of 7 mm. It reconstructs the non-visible (back) surface, for which no visual data is available, with a 3D error of 17 mm.

Abstract

The objective of Shape-from-Template (SfT) is to infer an object's shape from a single image and a 3D object template. Existing methods are called thin-shell SfT as they represent the object by its outer surface. This may be an open surface for thin objects such as a piece of paper or a closed surface for thicker objects such as a ball. We propose volumetric SfT, which specifically handles objects of the latter kind. Volumetric SfT uses the object's full volume to express the deformation constraints and reconstructs the object's surface and interior deformation. This is a challenging problem because for opaque objects, only a part of the outer surface is visible in the image. Inspired by mesh-editing techniques, we use an As-Rigid-As-Possible (ARAP) deformation model that softly imposes local rigidity. We formalise ARAP isometric SfT as a constrained variational optimisation problem which we solve using iterative optimisation. We present strategies to find an initial solution based on thin-shell SfT and volume propagation. Experiments with synthetic and real data show that our method has a typical maximum relative error of 5% in reconstructing the deformation of an entire object, including its back and interior for which no visual data is available.

1. Introduction

Reconstructing 3D objects from images is an important problem in Computer Vision. It is solved in the case of rigid environments with rigid Structure-from-Motion (SfM). However, rigid SfM fails for deformable objects such as a piece of paper, cloth or the human body. Non-rigid reconstruction is an important current challenge with a wide spectrum of applications in medical imaging and the entertainment industry to name a few. Non-rigid reconstruction encompasses at least two different problems: Non-Rigid Structure-from-Motion (NRSfM) [2, 9, 10], which uses a set of images of a deforming object, and Shape-from-Template (SfT) [21, 4, 16] which uses a single image and a textured 3D template of the object which may be matched to the image. The objective in SfT is to obtain the object's deformed shape in the camera's coordinate frame using a *deformation constraint* formulated from the object's physical material. Existing SfT methods use deformation constraints on the object's outer surface, whose thickness is considered infinitesimal. We thus call them *thin-shell SfT* methods. Thin-shell SfT is very well adapted to thin objects, such as a piece of paper or a balloon, whose outer surfaces may be well approximated by an open or a closed surface. However, while thin-shell SfT handles thicker objects such as the woggle of figure 1 or a foam ball, it does

not fully exploit the strong constraints induced by the object's non-empty interior.

We bring SfT one step further by introducing *volumetric SfT*, defined as an SfT method which uses a deformation constraint on the object's outer surface and interior. An example is shown in figure 1. Volumetric SfT reconstructs the object's interior deformation, which is not reconstructed by thin-shell SfT, and reconstructs the object's outer surface more accurately than thin-shell SfT thanks to the stronger deformation constraint it uses. Volumetric SfT is challenging as only the front part of the object's surface is visible in the image: the object's back surface and interior have to be inferred with no direct visual observations. We propose to instantiate volumetric SfT using the As-Rigid-As-Possible (ARAP) deformation model, which has been used extremely successfully in Computer Graphics [23, 27]. The ARAP model maximises local rigidity while penalising stretching, sheering and compression. More specifically, ARAP has been widely used to perform mesh editing of animated characters [28, 29] because the resulting deformations locally preserve the object's structure.

Contrary to thin-shell SfT, volumetric SfT is largely unexplored. The closest method to volumetric SfT is perhaps [26], where SfT has been combined with silhouette-based reconstruction. The template is first reconstructed from a reference image using a silhouette-based method inspired from [13]. This method reconstructs objects that have a plane of symmetry parallel to the image plane and does not infer concavities, which is also a limitation of most silhouette-based methods [13, 14]. The template is then deformed using a data term based on silhouette, area and orthographic reprojection constraints. The deformation model extends thin-shell isometry by placing virtual nodes in the object's interior, with the objective of preserving the object's volume. The method has a flip ambiguity which cannot be resolved automatically, owing to the orthographic camera model. In contrast, we solve volumetric SfT without restricting the topology of the object and using the perspective camera. By using ARAP, our method preserves the object's interior structure while jointly reconstructing the deformation of the object's full outer surface and interior, as illustrated in figure 2. The scope of SfT is to handle objects such as a piece of paper or cloth whose shape is unpredictable and thus lives in a large space. This is different from methods whose aim is to reconstruct an object-class such as the set of faces, whose shape space can be approximated well by a smaller dimensional statistical model [8, 24, 25].

Technically, we bring two main contributions. First, we show that, unlike thin-shell isometry, imposing volume isometry exactly reduces to global rigidity. This motivates the use of ARAP, which maximises rigidity locally, thus preserving the object's structure, while complying with the reprojection constraints.

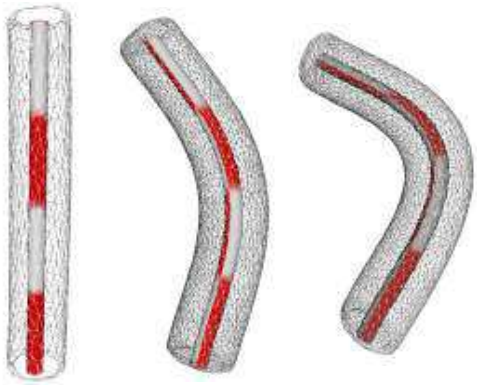

Figure 2: As opposed to thin-shell SfT, volumetric SfT reconstructs the object's interior deformation. In this example using the data from figure 1, a virtual cylinder is placed inside the woggle's template. It is then deformed using the deformation reconstructed by volumetric SfT to aid visualization of the object's reconstructed interior deformation. The second deformation is the one shown in figure 1.

ARAP volumetric SfT involves solving a non-convex constrained variational optimisation problem. We discretise the object's volume and relax the constraints to convert the variational problem into an unconstrained non-linear least-squares optimisation problem. This problem can then be solved with standard numerical solvers such as Levenberg-Marquardt. Second, we contribute with two heuristic initialisation methods. These methods use isometric thin-shell SfT and propagate the result through the object's volume. Experimental results on synthetic and real data show that volumetric SfT improves accuracy to a large extent compared to state-of-the-art thin-shell SfT methods.

2. Previous Work

Almost all existing SfT methods are thin-shell SfT: they use a thin-shell deformation model with physical constraints, such as isometry [21, 22, 4], conformity (angle preserving) [4] and linear elasticity [15, 11]. Thin-shell isometry and conformity are formulated with geometric, thickless surfaces. Thin-shell isometry has been the most studied model and leads to a well-posed and analytically solvable SfT problem [4]. Interestingly, linear elasticity as used in [15, 11] follows the plate theory. It requires the surface model to include thickness, which must however be 'small' so that out-of-plane object deformations may be neglected. In continuum mechanics, this means that the thickness is at least ten times smaller than the object's largest dimension. These methods are thus thin-shell SfT. They require

one to provide the Young modulus of the object's material and, more importantly, boundary conditions expressed as a set of known 3D point coordinates, which may restrict their applicability. We propose volumetric SfT which, in contrast to thin-shell SfT, recovers the deformation of the object's outer surface and interior, does not restrict the object's shape to follow the plate theory, does not require knowing the Young modulus and does not need boundary conditions. A related goal was pursued in [26] where a silhouette-based method was combined with SfT. However, as discussed in the introduction in detail, this method requires stronger image cues, including silhouette and point correspondences, and recovers two-way ambiguous shape solutions. In contrast, volumetric SfT only requires point correspondences and produces a unique solution.

3. Modelling

Notation. We use Greek letters for functions and bold letters for vectors and matrices. Scalars are in regular italics. We use calligraphic letters for sets, and $|\mathcal{A}|$ for the size of set $\mathcal{A}$. We use the operator $\mathbf{J}_\varphi$ for the Jacobian of φ.

Geometric model. Figure 3 shows a general diagram of volumetric SfT extending an existing thin-shell framework [4, 7]. We denote the 3D template as the volume $\mathcal{V}_T \subset \mathbb{R}^3$, the unknown deformed volume as $\mathcal{V}_S \subset \mathbb{R}^3$, and their respective outer surfaces as $\partial\mathcal{V}_T$ and $\partial\mathcal{V}_S$. We denote as $\mathcal{S} \subset \partial\mathcal{V}_S$ the deformed object's visible surface part, *i.e.* the part which is directly observed in the input image $\mathcal{I} \subset \mathbb{R}^2$, and $\mathcal{T} \subset \partial\mathcal{V}_T$ the corresponding part in the template surface. We use a 2D surface parameterisation space $\mathcal{F}$, called the flattened template. This allows us to represent the template's outer surface $\partial\mathcal{V}_T$ by a known invertible embedding $\Delta \in C^2(\mathcal{F}, \mathbb{R}^3)$. In practice, $\mathcal{F}$ and Δ may be computed from $\partial\mathcal{V}_T$ by any flattening method; we use conformal flattening [3]. Using $\mathcal{F}$, the unknown deformed surface $\mathcal{S}$ may be represented by an embedding $\varphi \in C^2(\mathcal{F}, \mathbb{R}^3)$. The deformation between $\mathcal{V}_T$ and $\mathcal{V}_S$ is the unknown mapping $\psi \in C^2(\mathcal{V}_T, \mathbb{R}^3)$.

The task in volumetric SfT is not only to compute the volume $\mathcal{V}_S$ of the deformed object, but to find a full volume deformation function $\psi \in C^2(\mathcal{V}_T, \mathbb{R}^3)$, matching points between the object's template and deformed states. This is a challenging task, as most of $\mathcal{V}_S$ is not directly observed in the image: assuming the object is opaque, the only visual information comes from the outer surface's visible part. The surface embedding φ may of course be directly recovered from the volume deformation ψ computed by volumetric SfT as $\varphi = \psi \circ \Delta$. Depending on the formulation, thin-shell SfT computes either the surface embedding φ [4] or a 3D surface deformation, which is a restriction of ψ to $\partial\mathcal{V}_T$ [16]. The full volume deformation ψ cannot be directly recovered in either case. Our initialisation strategy for volumetric SfT involves inferring ψ from φ through two new

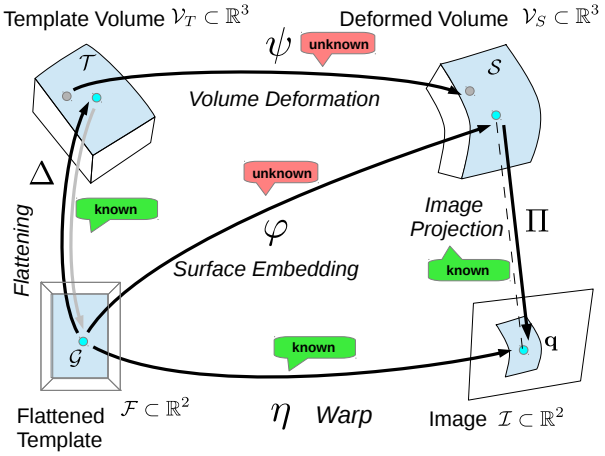

Figure 3: General diagram of volumetric SfT. The visible surface part is shown in blue.

solutions which we name volume interpolation.

Finally, we define as $\eta \in C^2(\mathcal{F}, \mathbb{R}^2)$ the registration warp between $\mathcal{F}$ and the image. The warp η may be estimated automatically from point correspondences [19, 18]. Estimating the warp directly gives two pieces of information. First, it identifies the subset $\mathcal{G} \subset \mathcal{F}$ corresponding to the surface's visible part in the flattened template. Second, it establishes the *reprojection constraint* on φ and ψ as:

$$\eta = \Pi \circ \varphi = \Pi \circ \psi \circ \Delta, \tag{1}$$

where Π denotes perspective projection in coordinates normalised with respect to the camera's intrinsics $\Pi(\mathbf{Q}) = \frac{1}{Q_z}(Q_x \; Q_y)^\top$ with $\mathbf{Q} = (Q_x \, Q_y \, Q_z)^\top$.

Deformation model. Thin-shell isometry allows SfT to resolve the visible surface part uniquely [4] and to extrapolate the non-visible surface part [16]. Applied to an object's interior volume, isometry yields the following differential constraint on the mapping ψ:

$$\mathbf{J}_\psi^\top \mathbf{J}_\psi = \mathbf{I}_{3\times 3}. \tag{2}$$

According to the Mazur-Ulam theorem [20], equation (2) constrains ψ to be a rigid transformation. So as to model deformations, one must relax equation (2). One possibility is the so-called ARAP heuristic [23], which means searching for ψ such that $\left\| \mathbf{J}_\psi^\top \mathbf{J}_\psi - \mathbf{I}_{3\times 3} \right\|_p^2$ is minimised over $\mathcal{V}_T$. We propose to combine ARAP with the reprojection constraint to preserve the object's local structure while driving its deformation to comply with the image constraints.

4. Volumetric SfT

We present ARAP volumetric SfT, which finds the deformation ψ that transforms the volume $\mathcal{V}_T$ into the unknown volume $\mathcal{V}_S$ whose surface is partially observed in $\mathcal{I}$.

4.1. Formulation and Non-Convex Solution

Problem formulation. Combining the reprojection constraint (1) with ARAP leads to the following variational problem:

$$\min_{\psi} \rho \underbrace{\int_{\mathcal{V}_T} \left\| \mathbf{J}_{\psi}^{\top}\mathbf{J}_{\psi} - \mathbf{I} \right\|_p^2 d_{\mathcal{V}_T}}_{\text{ARAP penalty}} + (1-\rho) \underbrace{\int_{\mathcal{G}} \|\eta - \Pi \circ \psi \circ \Delta\|_2^2 \, d_{\mathcal{G}}}_{\text{Reprojection}} . \quad (3)$$

The reprojection constraint is convex, but the ARAP penalty is not. Problem (3) is thus difficult to solve, as it involves integrals and equality constraints. Local analytical solutions, as the ones proposed in [4] for thin-shell SfT, are not applicable at non-visible points since they do not have a data term. This is because the reprojection constraint applies on the visible surface part $\mathcal{S}$ only, corresponding to the subset $\mathcal{G}$ of the flattened template.

Discretisation and optimisation. We evenly discretise the template volume $\mathcal{V}_T$ with a set of 3D points $\mathcal{P}_{\mathcal{V}_T}$. We define the deformation functional $\epsilon_d[\psi]$:

$$\epsilon_d[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{V}_T}|} \sum_{\mathbf{P} \in \mathcal{P}_{\mathcal{V}_T}} \left\| \mathbf{J}_{\psi}^{\top}(\mathbf{P})\mathbf{J}_{\psi}(\mathbf{P}) - \mathbf{I}_{3\times 3} \right\|_p^2 . \quad (4)$$

We write $\epsilon_r[\psi]$ over a regular discretisation $\mathcal{P}_{\mathcal{G}}$ of $\mathcal{G}$:

$$\epsilon_r[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{G}}|} \sum_{\mathbf{p} \in \mathcal{P}_{\mathcal{G}}} \|\eta(\mathbf{p}) - \Pi(\psi(\Delta(\mathbf{p})))\|_2^2. \quad (5)$$

Finally, we optimise the following unconstrained non-linear least-squares problem:

$$\psi \;=\; \arg\min_{\psi} \rho\epsilon_d[\psi] + (1-\rho)\epsilon_r[\psi] \qquad 0 < \rho < 1, \quad (6)$$

where ρ is a weight that balances the ARAP penalty and the reprojection constraint.

In order to find a numerical solution to problem (6), we use a parametric representation of the solution $\tilde{\psi} \in C^2(\mathcal{V}_T \times \mathbb{R}^n, \mathbb{R}^3)$, where n is the dimension of the parameter space. Let $\mathbf{L} \in \mathbb{R}^n$ be the parameter vector and $\mathbf{Q} \in \mathcal{V}_T$, we have $\tilde{\psi}(\mathbf{Q}, \mathbf{L}) \in \mathcal{V}_S$. We have multiple choices for $\tilde{\psi}$ such as the popular linear basis expansion representations (the NURBS [17], the Thin-Plate Splines (TPS) [5], the B-Spline [6], tetrahedron mesh displacements, etc.). We use the TPS representation.

Problem (6) is then optimised using Levenberg-Marquardt. Iterative methods can be highly accurate but because problem (6) is non-convex due to the ARAP penalty, the iterations may converge to a non-global minimum. Therefore, it is important to provide an initial solution close to the global minimum.

4.2. Convex Initialisation

Our initialisation strategy finds an approximate solution ψ_0 to problem (3) in two main steps.

1) **Isometric thin-shell SfT.** We first compute the embedding φ that represents the visible surface $\mathcal{S}$. We approximate the deformation from $\mathcal{T}$ to $\mathcal{S}$ by thin-shell isometry, giving the following problem reformulation:

$$\text{Find } \varphi \text{ s.t. } \begin{cases} \mathbf{J}_{\varphi}^{\top}\mathbf{J}_{\varphi} = \mathbf{J}_{\Delta}^{\top}\mathbf{J}_{\Delta} \\ \eta = \Pi \circ \psi \circ \Delta \end{cases} \text{on } \mathcal{G}, \quad (7)$$

where $\mathbf{J}_{\varphi}$ is a 3×2 matrix. Problem (7) has an analytical solution given in [4].

2) **Volume interpolation.** We use φ to infer ψ representing the full volume deformation. We propose two strategies.

i) Global Smoothness (GS). Our first strategy is based on the assumption that the deformation of the volume is smooth. We can, therefore, formulate the problem as finding the smoothest volumetric deformation such that the deformation at the surface agrees with the solution from thin-shell SfT. We write the discretised transport error:

$$\epsilon_e[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{G}}|} \sum_{\mathbf{p} \in \mathcal{P}_{\mathcal{G}}} \|\varphi(\mathbf{p}) - \psi(\Delta(\mathbf{p}))\|_2^2. \quad (8)$$

Because φ was computed in step *1)*, this is a linear least-squares cost in ψ. We then compute ψ_0 as the solution of the following system:

$$\psi_0 = \arg\min_{\psi} \alpha\epsilon_e[\psi] + (1-\alpha)\epsilon_s[\psi] \quad 0 < \alpha < 1, \quad (9)$$

where $\epsilon_s = \iint \| \frac{d^2\psi}{d\mathbf{p}^2} \|_2^2 d\mathbf{p}$ is a smoothing term called the bending energy and α is a weight balancing the transport error and smoothness. As in the non-convex solution, we use a TPS representation of ψ. The bending energy is then a quadratic function of the TPS parameters [5], making problem (9) linear least-squares, thus convex and easily solvable. GS is a natural way of initialising ψ from φ, but as smoothness is the only constraint it uses to propagate the visible surface deformation, it may spoil the object's inner local structure by causing local shear, shrinking and extension. Our second volume interpolation method addresses this issue.

ii) Local Rigidity (LR). This method is based on the idea that from thin-shell SfT, we can compute a local rigid transform at every point on the visible surface to propagate shape through the object's volume, in an ARAP manner. The key idea is to initialise ψ on the surface's visible part from φ, and use local rigidity to iteratively 'complete' ψ. This is implemented by iteratively drawing local rigid transformations to locally extrapolate the deformation. Concretely, we first find correspondences to all points in $\mathcal{P}_{\mathcal{V}_T}$ (which may be seen as a discretisation of ψ) and then fit a continuous parametric representation of ψ. We write the corresponding point of $\mathbf{P} \in \mathcal{P}_{\mathcal{V}_T}$ as $\mathbf{Q}(\mathbf{P})$.

We first use Delaunay triangulation of the point set $\mathcal{P}_{\mathcal{V}_T}$ to define a tetrahedral mesh. A given tetrahedron has four

vertices, which we denote as $\mathbf{P}_{n_1}$, $\mathbf{P}_{n_2}$, $\mathbf{P}_{n_3}$ and $\mathbf{P}_{n_4}$. Drawing a local rigid transformation is achieved by selecting a tetrahedron which has three vertices, say the first three ones, lying in the 'completed' domain of ψ, for which $\mathbf{Q}_{n_i} = \mathbf{Q}(\mathbf{P}_{n_i})$ exist for $i = 1, \ldots, 3$. At the early iterations, this means that these three vertices will have to be in the surface's visible part, and that $\mathbf{Q}_{n_i} = \varphi(\Delta^{-1}(\mathbf{P}_{n_i}))$, $i = 1, \ldots, 3$. From these three correspondences $\{\mathbf{P}_{n_i} \leftrightarrow \mathbf{Q}_{n_i}\}$, $i = 1, \ldots, 3$, we fit a rigid transform Ω in the least-squares sense using [12]. Completing ψ by local rigidity is then simply done by setting $\mathbf{Q}(\mathbf{P}_{n_4}) = \Omega(\mathbf{P}_{n_4})$. At each iteration, we cycle through all tetrahedra with three vertices lying in the completed domain of ψ. This obviously causes the fourth vertex of many tetrahedra to receive multiple predictions, as several tetrahedra may share it as their single unknown vertex. In order to approximate ARAP as best possible, we keep the prediction for which Ω was fitted with the lowest error. We stop the iterations when all points in $\mathcal{P}_{\mathcal{V}_T}$ have been given a correspondence.

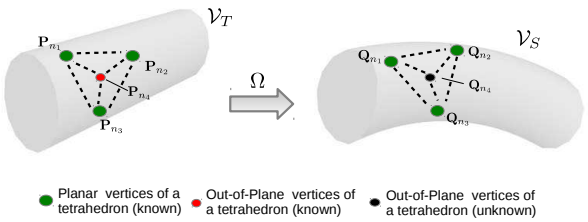

Figure 4: Volume interpolation using Local Rigidity.

We finally define the discretised transport error:

$$\epsilon'_e[\psi] = \frac{1}{|\mathcal{P}_{\mathcal{V}_T}|} \sum_{\mathbf{P} \in \mathcal{P}_{\mathcal{V}_T}} \|\mathbf{Q}(\mathbf{P}) - \psi(\mathbf{P})\|_2^2, \qquad (10)$$

and obtain ψ_0 as the solution of the following optimisation problem:

$$\psi_0 = \arg\min_{\psi} \alpha \epsilon'_e[\psi] + (1-\alpha)\epsilon_s[\psi] \quad 0 < \alpha < 1, \quad (11)$$

where α is a weight balancing the transport error and smoothness. Equation (11) is linear least-squares, thus convex and easily solved.

5. Experimental Results

We report experiments with synthetic data and three sets of real data with different geometries and materials: a woggle, a sponge and an arm. The refinement solution (3) is tested using the L1 and L2 norms, and is then called L1-refinement and L2-refinement respectively. The two initialisation solutions are called GS-initialisation and LR-initialisation. We also compare with two isometric thin-shell SfT methods [4, 16], which were discussed as being representative of the state-of-the-art in the introduction. We use a constant weight $\rho = 0.005$ in the refinement problem (6) (for both L1-refinement and L2-refinement) and a constant weight $\alpha = 0.0001$ in both equation (9) for GS-initialisation and equation (11) for LR-initialisation. We noticed that the algorithms were not very sensitive to these values up to an order of magnitude.

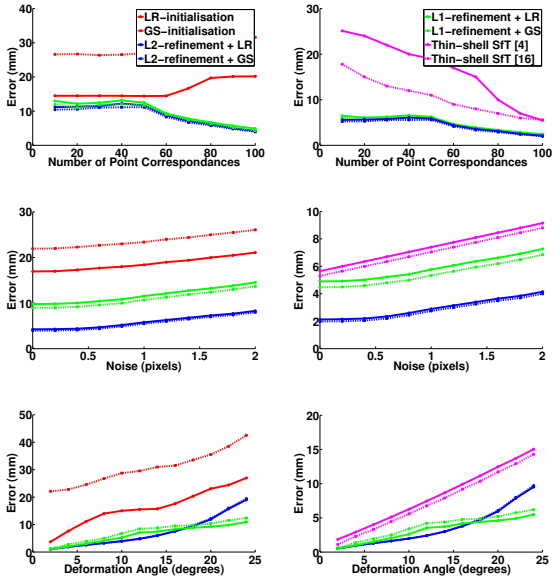

Figure 5: Synthetic data experiments.The graphs on the left show the 3D volume error and the ones on the right show the error on the 3D visible surface.

Experiments with synthetic data.

We test our method for volumetric SfT in various conditions of noise, deformation and correspondences. We simulate a box of dimension $20 \times 20 \times 10$ cm^3 and deform it by bending each of its layers along a vertical rule with some varying bending angle. The higher the bending angle, the more important the box's deformation. If the bending angle is zero, the box is undeformed. We then create a virtual image of the box by projecting it using a perspective camera and add noise in pixels. The default bending angle is 10 degrees.

The results are shown in figure 5, and are averages over multiple runs for each geometric configuration. The three graphs on the left column of figure 5 show the 3D volume error in mm, computed as the RMSE (Root Mean Square Error) over a dense grid of points sampled over the object's outer surface and interior. The results on these graphs thus only concern the proposed volumetric SfT methods. We observe that the refinement methods all do significantly better than all initialisation methods. LR-initialisation does consistently and substantially better than GS-initialisation. This is explained by the fact that LR-initialisation follows

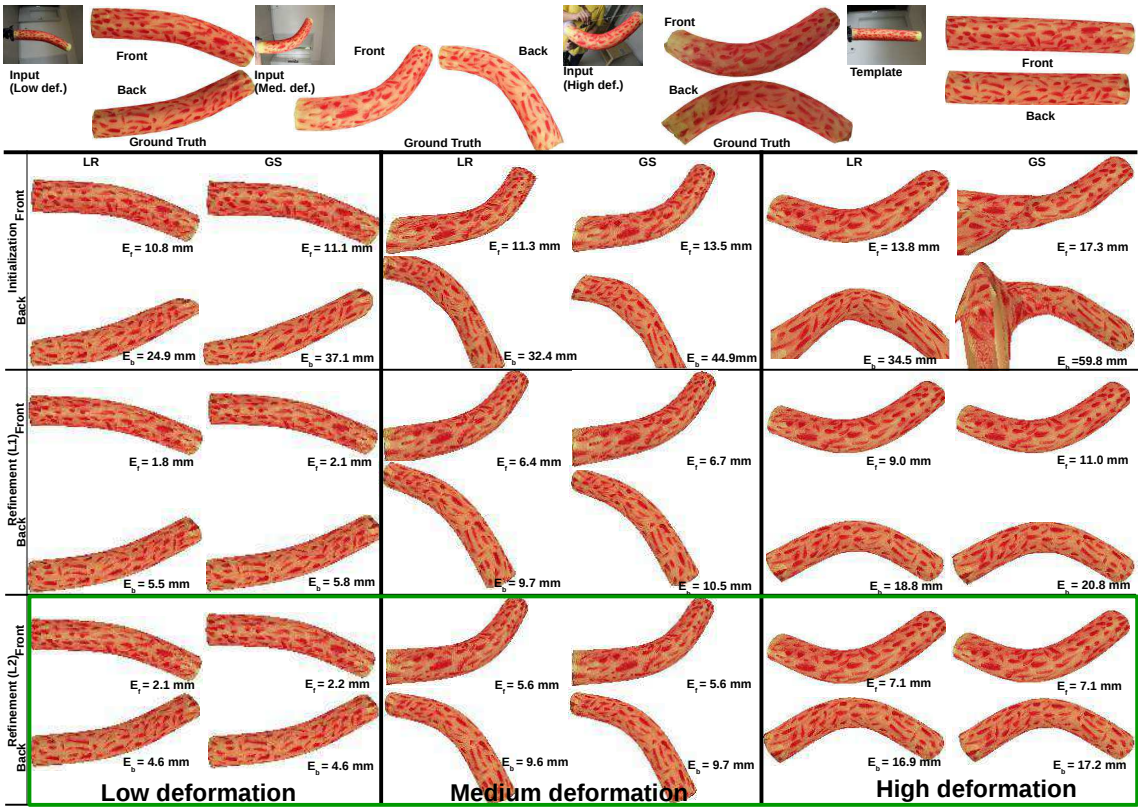

Figure 6: Results on the woggle. The green boxes show the best performing algorithm for each deformation level.

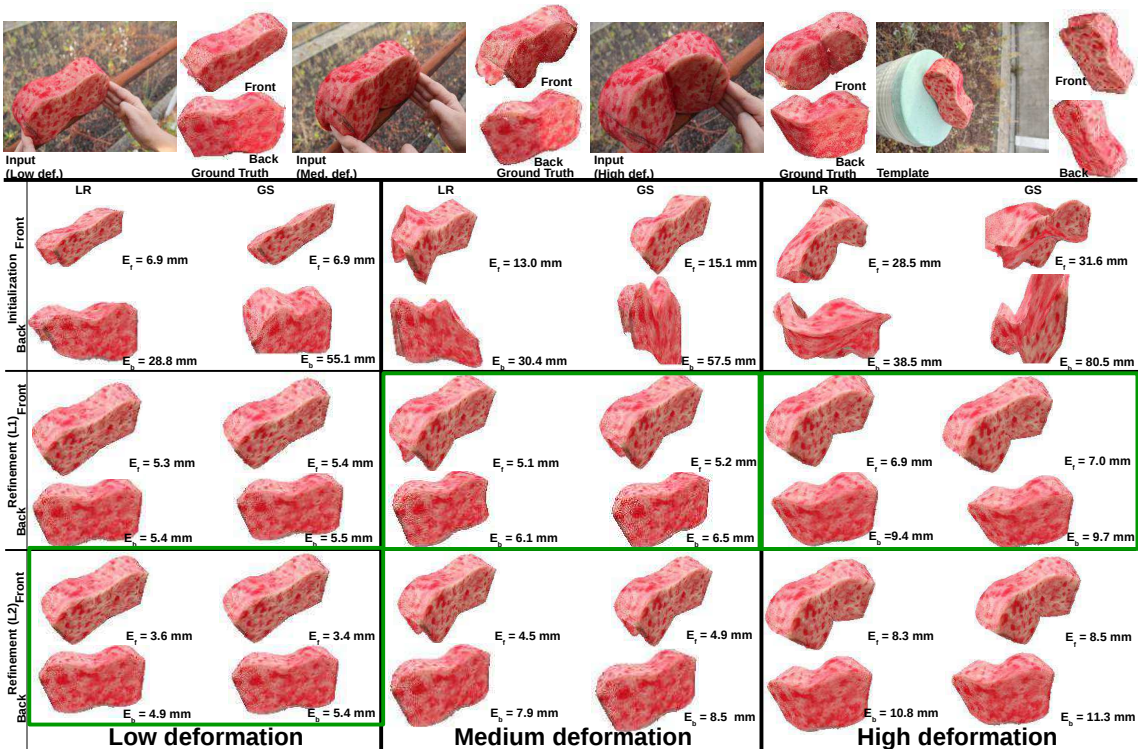

Figure 7: Results on the sponge. The green boxes show the best performing algorithm for each deformation level.

the ARAP methodology for local propagation, while GS-initialisation simply uses smoothness, which is a weaker constraint. L2-refinement does generally better than L1-refinement, except when the deformation increases beyond a certain point. All methods degrade with the amount of deformation and noise. Increasing the number of points improves the refinement methods but slightly degrades the initialisation methods. The three graphs on the right column of figure 5 show the visible surface error in mm, computed as the RMSE over a dense grid of points sampled over the object's visible surface. The same observations which we made for the refinement methods on the 3D volume error can be made, and the general trends also apply to the two tested thin-shell SfT methods. Importantly, we observe that volumetric SfT does consistently and in several case substantially better than thin-shell SfT, even if the measured error concerns only the visible surface part, which is theoretically handled well by both types of methods. This means that the extra constraints used in volumetric SfT compared to thin-shell SfT have a very positive influence on this part of the reconstruction too.

Experiments with real data.

We evaluate the performance of the methods with three real-world objects captured across a range of deformed states.

Test data and ground truth acquisition. The three objects are a foam tube called a woggle (figure 6), a sponge (figure 7) and a human arm (figure 8). We construct the 3D template of each object using Photoscan, a dense rigid SfM package [1]. To achieve this we photograph the objects in a rigid pose from a number of different viewpoints in order to capture the full 3D geometry (we use 47, 55 and 78 images for the three objects respectively). We apply a small amount of manual post-processing to fill holes and make the templates watertight. Then we physically apply forces to the real objects to obtain a set of deformed shapes, which we grouped into three levels: low, medium and high deformation. For each level we compute the ground truth shape by photographing again the deformed object from approximately 50 viewpoints and then running Photoscan. Because Photoscan provides the reconstructed object and the pose for each camera image, this provides us with the ground truth shape of the object's outer surface (including the back surface) in camera coordinates.

Similarly to the vast majority of previous SfT methods, ours takes as input point correspondences between the 3D template and the input image. These can be computed automatically using for instance SIFT combined with outlier detection [19, 18]. However, to keep the results independent of the matching algorithm, we define correspondences manually. For the three objects this gives between 50 to 350 correspondences per image. We click between 30 to 40 correspondences per image and create the others using TPS interpolation [5].

Performance metrics and method comparison. We calculate two types of 3D errors, E_f and E_b, both expressed in mm, for the visible and non-visible surface parts respectively, as the RMSE discrepancy between the true and reconstructed 3D points at the correspondences. For each of the three datasets, and each of the three deformation levels, the top of each figure shows the template, the input image and the ground truth shape. On each figure, the deformation goes through low, medium and high level from left to right. The rows then show the results of both initialisation methods and their use to initialisation both refinement methods, giving a total of six combinations. The 3D errors E_f, including the two thin-shell methods, are finally summarised in table 1. We observe that LR-initialisation gives consistently better results than GS-initialisation, which is in accordance with our observations made on simulated data. The difference becomes very important for stronger deformations. This has a very small impact on the refinement results, for both refinement methods. We can observe small differences between the two refinement methods. However, refinement+LR converges faster than refinement+GS because LR-initialisation is closer to the correct solution. However, none of them is consistently better than the other, even if for the woggle L2-refinement is slightly more accurate, whereas for the sponge and arm L1-refinement is slightly more accurate. The 3D error E_f for volumetric SfT for both refinement methods is consistently smaller than for thin-shell SfT. Depending on the dataset and the deformation level, it is between two and ten times smaller. This confirms our observations made on simulated data that, even if the surface's visible part is handled naturally by both volumetric and thin-shell SfT, the stronger volume deformation constraints used by the former allows it to obtain a much more accurate result.

The results shown in figures 6, 7 and 8 show that the reconstructed object shape is visually close to the true shape. This means that volumetric SfT could allow a user to handle a physical object as a proxy interactor in applications such as virtual shape editing. Quantitatively, the woggle, sponge and arm are 37 cm, 15 cm and 20 cm long, respectively. The relative highest error over the whole reconstructed volume deformation, for the highest level of deformation, is thus smaller than 5%, 7% and 5% of the objects' size, respectively. Our unoptimised MATLAB implementation on a standard desktop with 3.1GHz processor takes between 10 - 25 seconds for the refinement to converge. The computation time for LR-initialisation is 3-5 seconds while for GS-initialisation it is 1-2 seconds.

6. Conclusion and Future Work

We presented volumetric SfT, which reconstructs an object from a single image and a 3D template, by using de-

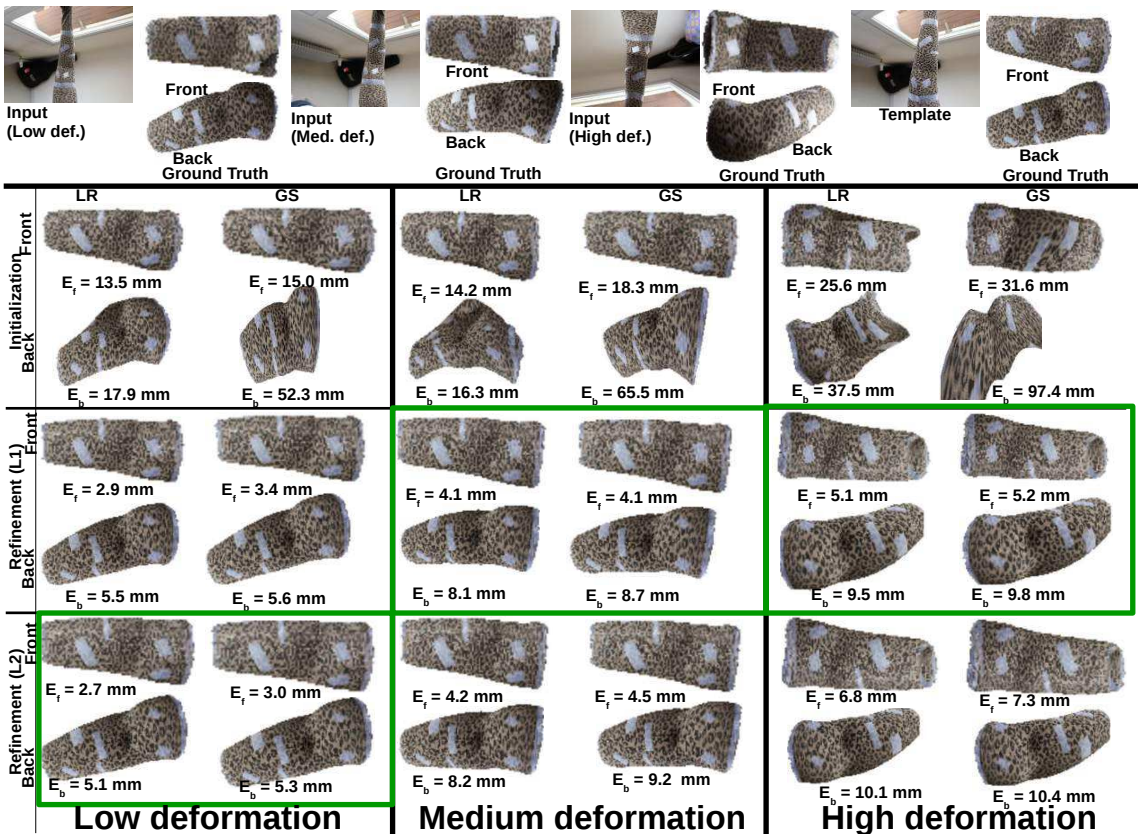

Figure 8: Results on the arm. The green boxes show the best performing algorithm for each deformation level.

Error Analysis (in mm)	Tube Data			Sponge Data			Arm Data		
	Low	Med.	High	Low	Med.	High	Low	Med.	High
Thin-shell SfT [4]	11.8	16.0	19.8	8.0	15.3	31.8	18.5	20.1	33.4
Thin-shell SfT [16]	10.4	11.4	12.7	6.0	25.5	35.4	15.5	18.9	35.7
GS-initialisation	11.1	13.5	17.3	6.9	15.1	31.6	15	18.3	31.6
LR-initialisation	10.8	11.3	13.8	6.9	13.0	28.5	13.5	14.2	25.6
L1-refinement + GS	2.1	6.7	11.0	5.4	5.2	7.0	3.4	4.1	5.2
L1-refinement + LR	1.8	6.4	9.0	5.3	5.1	6.9	2.9	4.1	5.1
L2-refinement + GS	2.2	5.6	7.1	3.4	4.9	8.5	3.0	4.5	7.3
L2-refinement + LR	2.1	5.6	7.1	3.6	4.5	8.3	2.7	4.2	6.8

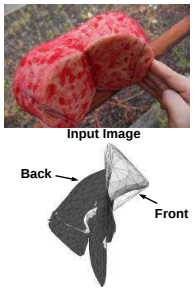

Table 1: (left) 3D visible surface error E_f for the datasets shown in figures 6, 7 and 8. (right) In practice, volumetric SfT always converges with LR or GS-initialisation. But, in order to create conditions of failure, we initialised the refinement at equation (11) very far from the optimal solution. Testing with the top right image led to the local minimum shown bottom right.

formation constraints on the object's outer surface and interior. Previous thin-shell SfT methods use constraints on the object's outer surface only. Volumetric SfT is thus to be used with non-empty and non-flat objects. We proposed an implementation of volumetric SfT using ARAP. Our implementation uses non-convex refinement and has an initialisation procedure following an ARAP propagation of a surface deformation obtained by thin-shell SfT through the object's volume. On method has significantly more accurate results than state-of-the-art isometric thin-shell SfT, reducing the error of an order of magnitude in some cases. ARAP volumetric SfT opens the way to doing Human-Computer Interaction using a proxy object such as a cushion and a simple monocular webcam. Volumetric SfT may also be instantiated with other deformation models, such as biomechanical models.

Acknowledgements. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

[1] Agisoft Photoscan 1.0.4, 2014.

[2] A. Del Bue, F. Smeraldi and L. Agapito. Non-rigid structure from motion using non-parametric tracking and non-linear optimization. In *CVPR*, 2004.

[3] A. Sheffer, B. Lévy, M. Mogilnitsky and A. Bogomyakov. ABF++: Fast and Robust Angle Based Flattening. In *SIGGRAPH*, 2005.

[4] A. Bartoli, Y. Gérard, F. Chadebecq, and T. Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *CVPR*, 2012.

[5] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567–585, 1989.

[6] Carl and D. Boor. *A practical guide to splines.*, volume 27. Springer-Verlag New York, 1978.

[7] A. Chhatkuli, D. Pizarro, and A. Bartoli. Stable Template-Based Isometric 3D Reconstruction in All Imaging Conditions by Linear Least-Squares. In *CVPR*, 2014.

[8] D. Anguelov and P. Srinivasan and D. Koller and S. Thrun and J. Rodgers and J. Davis. Scape: shape completion and animation of people. *ACM Transactions on Graphics*, 24:408–416, 2005.

[9] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *CVPR*, 2012.

[10] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013.

[11] N. Haouchine, J. Dequidt, M. Bergr, and S. Cotin. Single view augmentation of 3D elastic objects. In *ISMAR*, 2014.

[12] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.

[13] M. Oswald and E. Toppe and D. Cremers. Fast and globally optimal single view reconstruction of curved objects. In *CVPR*, 2012.

[14] M. Prasad and A. Zisserman and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *CVPR*, 2006.

[15] A. Malti, R. Hartley, A. Bartoli, and J. Kim. Monocular template-based 3D reconstruction of extensible surfaces with local linear elasticity. In *CVPR*, 2013.

[16] J. Ostlund, T. Ngo, and P. Fua. Monocular 3D Shape Recovery using Laplacian Meshes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[17] L. Piegl. On NURBS: a survey. *IEEE Computer Graphics and Applications*, 11:55–71, 1991.

[18] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *CVPR*, 2005.

[19] D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97:54–70, 2012.

[20] T. Rassias and P. Šemrl. On the Mazur-Ulam theorem and the Aleksandrov problem for unit distance preserving mappings. *Proceedings of the American Mathematical Society*, 118:919–925, 1993.

[21] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, 2009.

[22] M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:931–944, 2011.

[23] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, 2007.

[24] T. Cashman and A. Fitzgibbon. What Shape are Dolphins? Building 3D Morphable Models from 2D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:232 – 244, 2012.

[25] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

[26] S. Vicente and L. Agapito. Balloon shapes: Reconstructing and deforming objects with volume from images. In *3DV*, 2013.

[27] S. Zhang, A. Nealen, and D. Metaxas. Skeleton based as-rigid-as-possible volume modeling. In *EUROGRAPHICS*, 2010.

[28] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H. Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Transactions on Graphics*, 24:496–503, 2005.

[29] M. Zollhfery, E. Sert, G. Greiner, and J. Smuth. Gpu based arap deformation using volumetric lattices. In *EUROGRAPHICS*, 2012.