Shape-from-Template with Camera Focal Length Estimation

Toby Collins and Adrien Bartoli

Abstract One of the major and open research objectives in computer vision is to automatically reconstruct the 3D shape of a deformable object from a monocular image. Shape-from-Template (SfT) methods use prior knowledge embodied in a template that provides the object's 3D shape in a known reference position, and a physical model that constrains deformation. SfT methods have shown great success in recent years; however, accurate methods require an intrinsically calibrated camera. This is an important practical limitation because the intrinsics of many real cameras are not available, so they must be estimated with a dedicated calibration process. In this chapter, we present a novel SfT method that handles unknown focal length (a critical intrinsic of the perspective camera). The other intrinsics such as the principal point and aspect ratio are assumed to take canonical values, which is valid for many real cameras. We call this problem fSfT and we solve it by gradient-based optimization of a large-scale non-convex cost function. This is not trivial for two main reasons. Firstly, it requires suitable initialization, and we present a multi-start approach using a small set of candidate focal lengths (typically fewer than three are required). We combine this with a mechanism to avoid repeated exploration of the search space from different starts. Furthermore, we present cost normalization strategies, allowing the same cost function weights to be used in a diverse range of cases. This is crucial to make the method practical for real-world use. The method has been evaluated on twelve public datasets and it significantly outperforms a previous state-of-the-art fSfT method in both focal length and deformation accuracy.

Toby Collins
Faculté de Médecine, Institut Pascal, 28, place Henri Dunant, 63001 Clermont-Ferrand, France
e-mail: toby.collins@gmail.com.

Adrien Bartoli
Faculté de Médecine, Institut Pascal, 28, place Henri Dunant, 63001 Clermont-Ferrand, France
e-mail: adrien@bartoli.gmail.com

The research described in this chapter was conducted by Toby Collins while at the Institut Pascal as part of his Ph.D thesis.

1 Introduction

1.1 Shape-from-Template (SfT)

Reconstructing the 3D shape of a deformable object from a monocular image is a central and open problem in computer vision. It is usually much harder compared to reconstructing rigid objects because of the significantly larger problem space and much weaker constraints. To make deformable reconstruction well-posed, prior knowledge is required. In many previous works, including this chapter, the prior knowledge is embodied in a *template* [42, 5, 37, 3, 36, 44, 58]. The template provides a textured 3D geometric model of the object in a reference shape (usually implemented as a surface mesh), and it also constrains how the object can physically deform from its reference shape. The template can be acquired by various means, for example, using a computer assisted design (CAD) model, a 3D scanner, or using a reconstruction from monocular images viewing the object at rest with dense multi-view stereo (MVS). The approach of solving monocular deformable reconstruction with a template is often called *Shape-from-Template* (SfT) in the literature, or equivalently *template-based monocular deformable reconstruction*. We use SfT in this chapter.

SfT is ill-posed if the template can deform arbitrarily because of the loss of depth information resulting from camera projection. To overcome this, most methods use a quasi-isometic template that prevents deformations that significantly stretch or shrink the template. This deformation model is valid for many objects of interest including those made of leather, plastic, stiff rubber, paper and cardboard and tightly-woven fabrics. Crucially, a quasi-isometric template can guarantee that SfT is well-posed with a calibrated perspective camera [42, 3]. SfT has various applications that include Augmented Reality (AR) with deformable objects, Human-Computer Interaction (HCI) with deforming objects, and AR-guided surgery [36, 11, 55]. We illustrate two of these applications in Figures 1 and 2.

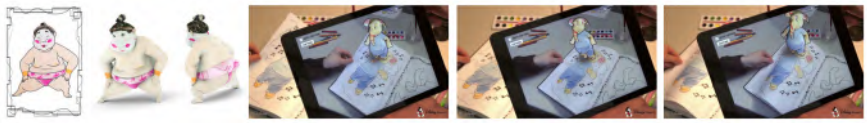


Fig. 1: An HCI and AR application of SfT from [30]. This is an interactive educational game, implemented on a tablet, for coloring a virtual 3D cartoon model using a real coloring book. A page from the book is colored with a pencil and SfT is used to register images of the page with a paper template. The registration from SfT allows the color from the image to be transferred to the paper template. Using a known association between the template and the cartoon model, the color is then transferred to the cartoon model. Because SfT also provides the 3D deformation of the paper template, the cartoon model can be virtually positioned on the paper sheet in real-time.

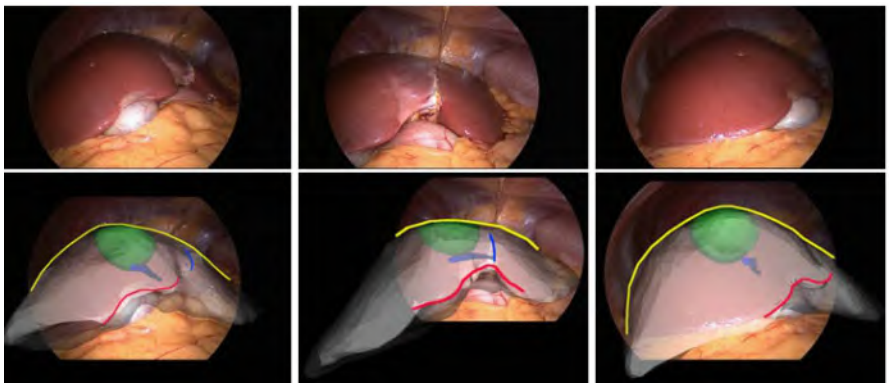


Fig. 2: An AR application of SfT from [25]. This is a prototype system to assist laparoscopic surgery of the liver using AR guidance, for safer liver resection. The top row of images shows three frames from a laparoscopic video of a liver. The bottom row shows the images augmented with hidden anatomical structures, including a tumor shown in green. This has been achieved using SfT, with a template constructed from a pre-operative CT image of the liver. The template is registered with laparoscopic images using SfT with contour and shading constraints.

1.2 Chapter innovations

SfT has been studied extensively with a perspective camera that is fully calibrated [42, 5, 37, 3, 36, 44, 58]. Calibrated intrinsics are required to relate camera coordinates with image coordinates. However, requiring known intrinsics is an important limitation in many real-world applications. A camera may have fixed and unknown intrinsics, or time-varying and unknown intrinsics *e.g.* if the camera zooms in or out. Neither situation can be handled by these methods. With fixed and unknown intrinsics, a classical camera calibration is usually performed with a rigid calibration target such as a checkerboard [59]. However, this has several limitations. The calibration process requires user interaction and time, it adds inconvenience to the user, and a calibration target may not be available. Furthermore, *a priori* calibration is only suitable when the camera intrinsics are fixed, which is restrictive.

This work describes a novel SfT algorithm that jointly estimates focal length and the template's 3D deformation from a single image. We refer to this problem as *focal length and Shape-from-Template* (fSfT). The other intrinsics are assumed to take canonical values. fSfT is an important problem because many real cameras can be modeled accurately with negligible skew, an aspect ratio of one, a principal point at the image center and negligible lens distortion. The only unknown intrinsic is the focal length. Our approach also works if the non-focal length intrinsics have known

non-canonical values, computed with a calibration process, however, this is a less common use case.

We solve fSfT by designing and optimizing a large-scale non-convex cost function $c(f, \boldsymbol{\theta})$ where f is the unknown focal length and $\boldsymbol{\theta}$ is the unknown template deformation. The form of c is similar to cost functions used by the most accurate SfT methods that require calibrated cameras [11, 36]. The cost function includes a data cost to register the template, and a deformation cost to penalize non-isometric deformation. We cannot optimize c with guaranteed global optimality. Nevertheless, we present a solution that works very well in practice, using local (iterative) optimization, combining a well-designed initialization strategy, careful cost modeling, and fast optimization. The principal novel characteristics and advantages of the approach are as follows:

1. We model *all* deformation constraints provided by the template in the cost function using a mesh-based physical deformation model. The results we obtain are significantly more accurate compared to the analytical method [4].
2. Precise initialization is not required in general. Initialization can be performed either using the analytical fSfT method [4] or using a very small number of focal length samples (three or fewer). We introduce a mechanism to improve computational efficiency to avoid repeated optimization in the same region of search space from different initializations.
3. We apply normalization techniques to the cost function, which greatly reduces the need to tune cost weights. Such tuning is a known issue in cost optimization approaches, and thanks to normalization, the same weights can be used for any problem instance. In our experimental evaluation, the same weights are used in *all* test cases, covering different object shapes, mesh discretization, textures, deformations, and imaging conditions. The ability to use the same weights in all conditions represents a significant advance towards a practical SfT solution.

1.3 Chapter organization

The remainder of this chapter is organized as follows. In §2 we summarize previous approaches for solving SfT and fSfT, and we discuss their main limitations. In §3 we describe our fSfT method in detail. In §4 we present the experimental results and in §5 we present our conclusion and directions for future research.

2 Related works

2.1 SfT approaches

We categorize prior SfT methods into three main groups: *i*) closed-form methods that do not require an initialization, *ii*) optimization-based methods, that are generally more accurate than closed-form methods but require an initialization, and more recently *iii*) convolutional neural network (CNN)-based methods. We now review these three categories.

2.1.1 Closed-form solutions

There are two main ways to solve SfT in closed-form. The first relaxes the isometric constraint to inextensibility [37, 42, 6], which allows the surface to shrink but not stretch. The problem is then cast as finding the deformation that maximizes the depth of matched points such that the Euclidean distances between surface points do not exceed their geodesic distances (as defined by the template). The problem is convex and has been solved using a greedy approach [42] and with second order cone programming (SOCP) using the interior point method [42, 37]. When the perspective effects are strong and there are many points, these methods can be very accurate. However, performance deteriorates when perspective effects and/or number of points are reduced [8].

The second main way to solve SfT in closed-form uses 1^{st}-order non-holonomic partial differential equations (PDEs) [3]. A PDE is setup at each surface point that relates surface depth, normals, camera projection and registration functions to 1^{st}-order. By imposing the isometric constraint, the PDE can be solved analytically by treating depth and normals as independent functions (a problem relaxation). The approach can also solve conformal (angle preserving) deformation [3] up to an arbitrary scale factor and convex/concave ambiguities. The PDE approach is very fast and it can be parallelized trivially. However, an accurate registration is normally required, which can be hard with poorly-textured surfaces.

2.1.2 Optimization-based solutions

The main disadvantage of the closed-form methods is to relax physical constraints, yielding sub-optimal solutions. In contrast, optimization-based solutions can exploit all available physical constraints. They take as input an initial sub-optimal solution, and perform iterative numerical optimization of a non-convex cost function [28, 58, 36, 11, 58, 31, 44]. Practically all methods use a pseudo *Maximum a Posteriori* cost function consisting of prior and data terms. The prior term nearly always penalizes non-isometric deformation. The data term penalizes disagreement between the deformed template and image evidence, such as the reprojection of point matches

[42, 37, 8, 3], patch-based matches [11], pixel-level photo-consistency [35, 58, 31] or contours [22, 54, 12, 17]. The main advantage of optimization-based methods is that complex cost functions can be used with no known closed-form solution. When properly initialized, they generally produce the most accurate solutions. Initialization can be performed using a closed-form method, or in the case of video data, with the solution from the previous frame (also called *frame-to-frame tracking*). There are three main open challenges with optimization-based solutions. The first is to increase the convergence basin, to reduces the dependency on good initialization. Methods such as coarse-to-fine optimization with multi-resolution meshes [58], or advanced schemes using geometric multi-grid [11] have proved useful. The second challenge is to reduce the cost of optimization for real-time solutions. This has been achieved with dimensionality reduction and GPU implementations such as [11]. The third challenge is designing a cost function that works well in a broad range of settings without requiring fine-tuning of hyper-parameters.

2.1.3 CNN-based solutions

CNNs have been used with great success for solving monocular reconstruction problems with deformable objects, such as 3D human pose estimation [32, 21], surface normal reconstruction [1, 56] and monocular depth estimation [14, 19, 27]. These works have stimulated recent progress for solving SfT with CNNs [40, 20, 15, 16]. The main idea is to train a CNN to learn the function that maps a single RGB image with known camera intrinsics to the template's deformation parameters. The CNNs in these works are trained using supervised learning with labeled data *i.e.* pairs of RGB images with the corresponding deformation parameters. Acquiring labeled data is a main practical challenge and it is practically impossible to obtain with real data. For this reason, these works rely heavily on simulated labeled data generated by rendering software such as Blender. On one hand, this offers a way to generate an enormous amount of training data. On the other hand, this opens up new challenges to ensure that the training data represents the variability and realism of real-world images. The so-called render gap is a term used to express the difference in realism between simulated and real data, and it affects the ability of the CNN to generalize well to real data. In SfT, we additionally face the problem that the space of possible deformations can be exceptionally large, making it difficult to cover the deformation space sufficiently with training data. For this reason, these works have been shown to work with objects undergoing simple, smooth deformation with a low-dimensional deformation space such as bending paper sheets or smoothly deforming cloth. Furthermore, these works require intrinsically calibrated cameras. There has been some recent progress for combining labeled simulated data with partially labeled real data in order to reduce the render gap [15]. The real data is acquired by a standard RGBD camera. This data does not contain sufficient information to train the CNN with supervised learning because RGBD images provide depth but not registration information. Consequently, the CNN is trained with a combination of supervised learning (to learn the template's depth) and unsupervised learning

(to learn the template's registration). Unsupervised learning is implemented using a photometric loss similar to multi-scale normalized cross-correlation.

While [15] marks a good step forward to solving SfT with CNNs, is requires calibrated RGBD data, so it is not applicable for solving fSfT. Furthermore, it requires a CNN to be trained specifically for each template, which is a strong practical and computational limitation. This directly contrasts our approach to fSfT, which does not require a computationally-intensive training process for each template, making it much easier to apply in real applications. Very recently, a CNN-based approach has been presented that eliminates the need to train for a specific template texture [16]. This is promising work, however it only works for flat, rectangular surfaces such as a sheet of paper. This contrasts our approach to fSfT which handles templates with any shape or texture.

2.2 fSfT solutions

There have been a few previous approaches to solve fSfT [2, 4]. An approach using affine correspondences (ACs) [33] has been presented using focal length sampling [2]. Given a focal length sample, the depth of each AC can be estimated using the AC's motion with plane-based pose estimation [23, 49, 59, 10]. A good focal length sample should produce reconstructions that satisfy the isometric assumption (specifically, that the Euclidean distance between reconstructed neighboring ACs is similar to their geodesic distance that is known *a priori* from the template). The method densely samples candidate focal lengths, and for each candidate, reconstruction compatibility is tested with the isometric assumption. However, this approach has several shortcomings. Firstly, it requires precisely registered ACs, which is difficult to achieve in practice, and it normally requires iterative registration refinement that is computationally expensive. Secondly, it cannot compute focal length in closed-form. Thirdly, it was only shown to work well with strongly-textured surfaces with many ACs.

An improved fSfT method that estimates focal length analytically has also been presented [4]. Using point correspondences, a local smooth warp is fitted to point neighborhoods, and focal length is then estimated at each point using a 2^{nd}-order PDE. In a final step, focal length estimates are robustly combined from multiple neighborhoods. This approach is fast and works well for smooth, well-textured surfaces. However, it is sub-optimal because it does not apply geometric constraints acting across point neighborhoods. These are essential to obtain accurate solutions especially when point correspondences are sparse.

fSfT has similarities with the problem of texton-based shape-from-texture with focal length estimation [13]. In texton-based shape-from-texture, the goal is to reconstruct a surface whose texture consists of repeated units known as textons. If the textons are small, such as the circular dots on a polka dot dress, they can each be modeled well by a plane. When the texton's metric shape is known *a priori* (for example, knowing that the textons are circular), each texton can be reconstructed

with plane-based pose estimation. The whole surface can then be reconstructed by interpolation or surface normal integration. [13] propose two analytical approaches to texton-based shape-from-texture with unknown focal length. The first solves f using the fact that the Euclidean distances between neighboring textons is approximately preserved by isometric deformation, producing a unique solution to f with a minimum of two textons. The second finds f that yields an integrable surface. There are similarities between [13] and the fSFT solutions, where each texton can be considered as a local template or a single AC. Additionally, both [13] and [4] use a weak-perspective approximation to obtain an analytical solution. The main limitation of [13] is to only solve texton reconstruction, and it cannot handle general objects or textures unlike the proposed fSfT method.

3 Methodology

3.1 Problem modeling

3.1.1 Template geometry and deformation parameterization

The setup is illustrated in Figure 3. The template surface $\mathcal{R} \subset \mathbb{R}^3$, defined in object coordinates, is modeled using a discrete texture-mapped triangulated surface mesh, called the *template mesh*. The template mesh has connected and non-overlapping triangle faces that model the surface piecewise linearly. It consists of vertices $\mathcal{V}$, edges $\mathcal{E}$ and faces $\mathcal{F}$. We define as $\boldsymbol{y}_{i\in[1,V]} \in \mathcal{R}$ the known 3D position of vertex i in object coordinates, where V is the number of vertices. We define as $\boldsymbol{\theta}_{\text{ref}}$ the known 3D positions of all vertices in object coordinates, corresponding to the template's reference shape:

$$\boldsymbol{\theta}_{\text{ref}} \stackrel{\text{def}}{=} \text{stk}\left(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_V\right) \tag{1}$$

where stk is the stacking operator that concatenates its arguments into a column vector. We define as $\boldsymbol{x}_{i\in[1,V]} \in \mathbb{R}^3$ the unknown 3D position of vertex i in camera coordinates, and we define as $\boldsymbol{\theta}$ the unknown positions of all vertices in camera coordinates:

$$\boldsymbol{\theta} \stackrel{\text{def}}{=} \text{stk}\left(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_V\right) \tag{2}$$

We define as $g(\boldsymbol{p};\boldsymbol{\theta}) : \mathcal{R} \rightarrow \mathbb{R}^3$ the spatial transformation of a surface point $\boldsymbol{p} \in \mathcal{R}$ from object coordinates to camera coordinates. This is parameterized using $\boldsymbol{\theta}$ with barycentric interpolation. Specifically, $\boldsymbol{p}$ is uniquely associated to an enclosing mesh triangle, and transformed according to the motion of the triangle's three vertices:

$$g(\boldsymbol{p};\boldsymbol{\theta}) : \mathcal{R} \rightarrow \mathbb{R}^3 \stackrel{\text{def}}{=} w_1\boldsymbol{x}_i + w_2\boldsymbol{x}_j + w_3\boldsymbol{x}_k \tag{3}$$

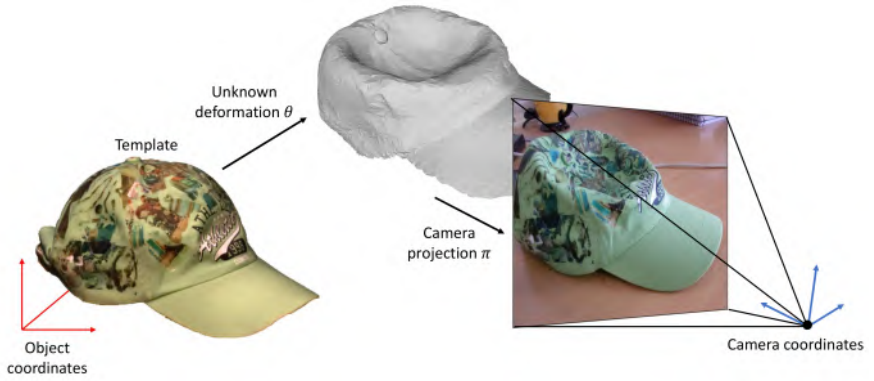


Fig. 3: SfT illustrated with a deformable cap [2]. The goal of SfT is to determine the unknown deformable 3D transform $\boldsymbol{\theta}$ that maps the template to camera coordinates, using *(i)* visual information in the image and *(ii)* deformation prior knowledge embodied in the template. The goal of fSfT is to solve SfT and jointly calibrate the camera's unknown focal length.

where i, j and k denote the three indices of the enclosing triangle, and $0 \leq w_1, w_2, w_3 \leq 1$ are the known barycentric weights associated with point $\boldsymbol{p}$ such that $w_1 + w_2 + w_3 = 1$.

3.1.2 Cost function

General form

We model fSfT with a non-convex cost function $c(\boldsymbol{\theta}, f) : \Theta \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ that maps deformation parameters $\boldsymbol{\theta}$ and focal length f to a positive real cost. We recall that $\boldsymbol{\theta}$ has been defined in Equation 2 as the unknown 3D positions of the template's vertices in camera coordinates.

We use a cost function inspired from the SfT literature with special attention to cost normalization to ensure it works well for a broad variety of problem instances (templates, deformations, viewpoints, textures *etc.*). The cost function is a weighted combination of three terms as follows:

$$c(\boldsymbol{\theta}, f; \mathcal{P}, \mathcal{Q}) = c_{\text{data}}(\boldsymbol{\theta}, f; \mathcal{P}, \mathcal{Q}) + \lambda_{\text{iso}} c_{\text{iso}}(\boldsymbol{\theta}) + \lambda_{\text{reg}} c_{\text{reg}}(\boldsymbol{\theta}) \tag{4}$$

The terms c_{data}, c_{iso} and c_{reg} are the data, isometric and regularization costs respectively. The terms λ_{iso} and λ_{reg} are weights that balance the influence of c_{iso} and c_{reg}, and are important hyper-parameters. Note that f influences only c_{data} directly and it influences the other terms indirectly via $\boldsymbol{\theta}$. In contrast, $\boldsymbol{\theta}$ influences all terms directly. As a pre-processing step, we normalize the template's size by a scale factor s so its

total surface area is 1 unit. This makes the cost function invariant to templates of different sizes. After optimization, the deformed template is recovered in its original size by scaling the solution to $\boldsymbol{\theta}$ by $\frac{1}{s}$. We now summarize our implementations of each term.

Data cost

The data cost c_{data} forces registration between the template's surface and the image from point correspondences. We denote the correspondences by the ordered sets $\mathcal{P} \in \mathbb{R}^{3N}$ and $\mathcal{Q} \in \mathbb{R}^{2N}$, where each point $\mathcal{P}(i \in [1, N]) \in \mathbb{R}^3$ is a point on the template surface in object coordinates, and $\mathcal{Q}(i \in [1, N]) \in \mathbb{R}^2$ is the point in image coordinates. Each point correspondence is related as follows:

$$\pi(g(\mathcal{P}(i), \boldsymbol{\theta}); f) = \mathcal{Q}(i) + \epsilon_i \tag{5}$$

where $\epsilon_i \in \mathbb{R}^2$ is unknown measurement noise and $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the perspective projection function that depends on the unknown focal length f:

$$\pi\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}; f\right) \stackrel{\text{def}}{=} \frac{f}{z}\begin{pmatrix} x \\ y \end{pmatrix} \tag{6}$$

Various approaches can be used to determine point correspondences and our method is not tied to a specific approach. One of the most common approaches used in previous SfT methods is image keypoint matching (also known as interest point matching), where standard methods such as SIFT [29] or learning-based methods such as LIFT [57] may be used. First, keypoints are detected in one or more images of the template, which are then back projected onto the template's surface to determine their barycentric coordinates. A second set of keypoints are then detected in the input image, and the two keypoint sets are matched based on keypoint descriptors to generate $\mathcal{P}$ and $\mathcal{Q}$. Often keypoint matching methods generate mismatches, which are point correspondences that do not physically correspond to the same surface point up to noise. In a pre-processing step, we remove mismatches with a dedicated method. Mature methods exist that can be used without knowledge of camera intrinsics. Possible methods include [36] where the template is fitted directly in 2D using a stiff-to-flexible annealing scheme, RANSAC-based model fitting such as [52], or methods based on motion consistency between neighboring points [39]. Our approach can be used with any combination of point matching and outlier detection methods, and we give our implementation choices for these in the experimental section of this chapter.

Outlier rejection methods are not always perfect, and our SfT method includes robustness built-into c_{data} to handle a small proportion of residual mismatches. This is implemented with the Huber M-estimator ρ_h, and the data cost writes as follows:

$$
\begin{aligned}
&c_{\text{data}}(\boldsymbol{\theta}, f; \mathcal{P}, \mathcal{Q}) = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \tfrac{1}{\sigma^2} \rho\left(\pi(g(\mathcal{P}(i); \boldsymbol{\theta}), f) - \mathcal{Q}(i)\right) && (a)\\
&\rho(\text{stk}(x, y)) \stackrel{\text{def}}{=} \rho_h(x) + \rho_h(y) && (b)\\
&\rho_h(z) \stackrel{\text{def}}{=} \begin{cases} \frac{1}{2} z^2 & \text{if } |z| < k \\ k(|z| - \frac{1}{2} k) & \text{otherwise} \end{cases} && (c)
\end{aligned}
\tag{7}
$$

The M-estimator acts to reduce the influence of correspondences with large residuals, which are commonly caused by mismatched points. The term σ is an estimate of the noise standard deviation. Its value depends on several factors, including the method used to generate point correspondences, image resolution and image noise. Unless σ is known, we use the following as default:

$$\sigma = \frac{1}{640} \max(w, h) \tag{8}$$

The image resolution (w, h) is taken into account in Equation (8) as σ is scaled by $\max(w, h)$. The denominator (640) is merely intended to help interpret σ relative to VGA resolution. The default defined in Equation (8) corresponds to a noise standard deviation of 1 pixel at VGA resolution. The value k is the Huber constant, set to a default $k = 10\,\sigma$.

Isometric cost

The isometric cost is implemented using a discrete approximation of the elastic strain energy E_{strain} of continuous surfaces [50]:

$$E_{strain} = \int_{\mathcal{R}} \|\mathbf{I}_{\mathcal{R}} - \mathbf{I}_{\mathcal{S}}\|_F^2 \, \mathrm{d}\mathcal{R} \tag{9}$$

Where $\mathbf{I}_{\mathcal{R}}$ and $\mathbf{I}_{\mathcal{S}}$ are the first fundamental forms of $\mathcal{R}$ (the template's surface in object coordinates) and $\mathcal{S}$ (the template's surface in camera coordinates) respectively. Penalizing E_{strain} encourages a deformation to preserve the first fundamental form, equivalent to penalizing non-isometric deformation. We use a discrete approximation of E_{strain} using a Finite Element Model (FEM) with Constant Strain Triangles (CSTs). This is a well-known model from mechanics that is suitable for relatively stiff (quasi-isometric) materials. Furthermore, using a FEM with CSTs gives a consistent discretization of the continuous strain energy. That is, under appropriate refinement conditions and norms, it is largely invariant to the mesh discretization and it converges to the continuous energy E_{strain}. This is important for our purposes because it eliminates the need to tune the cost's weight λ_{iso} according to the mesh discretization (number of vertices, placement of vertices and the triangulation). This is not true for the majority of membrane-like costs used in the SfT, which often use inconsistent isometric costs, such as those based on the preservation of mesh edge lengths [36, 6, 17] or the popular As-Rigid-As-Possible (ARAP) cost from [47]. The ARAP cost was shown to not be a consistent scheme in [26].

We compute the isometric cost c_{iso} as the discrete approximation of Equation (9) using CSTs. This is implemented by a weighted sum of strain energies from each triangle and the implementation details are given in Appendix 2. The isometric cost is a cubic expression in $\boldsymbol{\theta}$, which makes the minimization of c a large-scale non-convex problem.

Regularization cost

The regularization cost c_{reg} is convex and it encourages smooth deformation. Various implementations could be used, and we use a simple one using the moving least squares energy [46], also used in [11]. First the mesh is divided into overlapping *cells* where each cell describes the local motion of the mesh. We define one cell per vertex, containing all neighboring vertices connected by a mesh edge. The cell's motion is determined by the movement of its constituent vertices. Regularization is imposed by encouraging the cell's motion from object to camera coordinates to be described with an affine transform that is specific to each cell. This is implemented by penalizing the residual of the least squares affine motion of each cell. It is straightforward to show that the residuals are linear in $\boldsymbol{\theta}$, making c_{reg} convex and quadratic in $\boldsymbol{\theta}$. However, unlike c_{iso}, c_{reg} is not consistent, which means it depends strongly on the mesh discretization. The reason is similar for why the ARAP mesh energy is not consistent as discussed in [26]. Indeed, constructing a consistent and convex regularization cost with surface meshes is not trivial, and it has not been achieved before in the SfT literature. We handle this using normalization as follows. We apply a global reweighing to c_{reg} so that a small deformation from the rest state induces approximately the same cost irrespective of the template's discretization:

$$c_{\text{reg}} \leftarrow \frac{1}{\left\|\boldsymbol{J}_{\text{reg}}\right\|_F^2} c_{\text{reg}} \tag{10}$$

where $\boldsymbol{J}_{\text{reg}}$ is the Jacobian matrix of c_{reg}.

3.1.3 Cost normalization summary and weight hyper-parameters

In the cost definitions above, normalization has been used to significantly reduce the need to tune the cost weight hyper-parameters λ_{iso} and λ_{reg}. Specifically, normalization has made c strongly invariant to four sources of variability: template scale, template discretization, number of correspondences and image resolution. The influence of template scale is handled by rescaling the template to have unit area. The influence of template discretization is handled by two techniques. The first technique, used to normalize c_{iso}, involves the use of a consistent discrete approximation of a continuous surface function (strain energy) with an FEM, which achieves good discretization invariance by construction. The second technique, used to normalize c_{reg}, involves re-weighting the cost term by the magnitude of its Jacobian. This results in

a small deformation from the rest state having approximately the same regularization cost irrespective of the discretization.

Invariance to the number of correspondences N is achieved by rescaling c_{data} inversely in N in Equation (8). Image resolution invariance is achieved in Equation (8) by rescaling the residual error of each point inversely by the image size. Note that image resolution invariance is often achieved in SfT methods by defining residual errors in retinal coordinates (also called normalized pixel coordinates). However, this does not work for fSfT because it yields a trivial solution with the focal length at infinity.

Thanks to these normalization techniques, we use the same weights λ_{iso} and λ_{reg} for *all* templates and test datasets, where mesh resolutions vary considerably from $O(100)$ to $O(1,000)$ vertices, number of point correspondences vary from $O(10)$ to $O(1,000)$, and image resolutions vary from VGA to high definition (3600×2400 pixels). In all experiments, we use a default of $\lambda_{\text{iso}} = 1583$ and $\lambda_{\text{reg}} = 1\text{e} - 3$, found experimentally.

3.2 Optimization

3.2.1 Approach overview

The cost function c defined in Equation (4) is non-convex in the unknowns f and $\boldsymbol{\theta}$, arising from the non-convexity of both c_{iso} and c_{data}. Concerning c_{data}, the non-convexity is from the depth division of π. Concerning c_{iso}, the non-convexity is because c_{iso} is quartic in $\boldsymbol{\theta}$ as detailed in §2.2 of the appendix. Although f appears only directly in c_{data}, it has an indirect influence on c_{data} and c_{iso} by its connection to $\boldsymbol{\theta}$ in c_{data}.

Our goal is to determine f and $\boldsymbol{\theta}$ by optimizing the following large-scale non-convex optimization problem, which does not admit a closed-form solution:

$$\arg\min_{\boldsymbol{\theta}, f} c(\boldsymbol{\theta}, f; \mathcal{P}, \mathcal{Q}) \tag{11}$$

We propose an approach based on multi-start local (iterative) optimization that proves very effective in practice. We run local optimization from one or more initializations (also called starts), and the solution yielding the lowest overall cost is returned. To reduce computational cost, we propose a mechanism to terminate repeated search of the same search region from different initializations. There is a trade-off in having a larger number of initializations, which increase computational cost but may also increase the chances of finding the global minimum. This trade-off is explored in the experimental section of the chapter.

We now describe how the initialization set is generated and then describe the multi-start optimization algorithm.

3.2.2 Generating the initialization set

We define an initialization set $\mathcal{I}$ as $S \geq 1$ pairs: $\mathcal{I} = \{(f_1, \boldsymbol{\theta}_1), \ldots (f_S, \boldsymbol{\theta}_S)\}$, with each pair being an initial focal length and a corresponding initial deformation. We generate $\mathcal{I}$ by exploiting the fact that given an initial focal length, we can initialize deformation reasonably well using an existing closed-from SfT method. We therefore first generate a set of initial focal lengths, then we pass each of these, together with the other camera intrinsics, the template, and the point correspondences, to a closed-form SfT method, to generate the initial deformations.

Focal length generation

We compare two approaches to generate initial focal lengths. The first approach generates one focal length, estimated analytically from the set of point correspondences [4]. This method works best with relatively dense correspondences and smooth, well-textured surfaces. The second approach, which does not depend on the correspondences, works by focal length sampling.

We sample focal lengths using the opening angle representation, which is invariant to image resolution. The focal length f and lens opening angle ψ are related as follows:

$$\tan\left(\frac{\psi}{2}\right) = \frac{s}{2f}, \quad s \stackrel{\text{def}}{=} \max(w, h) \tag{12}$$

where w and h denote the image width and height in pixels respectively. In real-world SfT applications, lens opening angles are limited by two factors: *(i)* the physical limits of camera hardware, and *(ii)* theoretical limits and well-posedness of our problem. Concerning *(i)*, the distribution of opening angles of real cameras has been studied previously [45]. The distribution is mono-modal with a mode of approximately 50° and a maximum of approximately 100°, equivalent to a short focal length of $f \approx \frac{1}{2}\max(h, w)$ px. This sets a focal length lower bound in practice. In contrast, *(ii)* sets a focal length upper bound in practice for the following reason. A smaller opening angle (longer focal length) reduces the field-of-view, which in turn causes the viewing rays to become more parallel. When the viewing rays are almost parallel (known as quasi-affine projection), it can be difficult to stably estimate focal length with noise, which is a known result from camera calibration with rigid objects. Consequently, fSfT will not be solvable in real-world cases if the opening angle is very small. As such, we restrict the range of opening angle samples to $20^\circ \leq \psi \leq 100^\circ$. We note that this range is more than sufficient to cover all public datasets that have been used to test previous SfT methods. We found that in practice, we do not need to densely sample focal lengths and good results can be achieved with as few as three samples (20°, 50° and 80°), corresponding to a narrow, average and wide field-of-view.

Deformation generation

Given an initial focal length, there are several closed-form SfT methods that could be used to initialize deformation. We compare two of these. The first is the so-called *Maximum Depth Heuristic* method, referred to as MDH. [42, 37]. This makes a convex relaxation of the isometric constraints, leading to a second-order cone programming (SOCP) problem that can be solved with a depth maximization heuristic [37] or with the interior point method [42]. In this work, we use the interior-point method implemented in Sedumi [48]. The second approaches uses a perspective-n-point (PnP) method, referred to as PnP, which gives the best-fitting rigid pose. We also compare using both MDH and PnP (generating two initial deformations for each initial focal length). This generates twice as many initializations, however it is handled efficiently by detecting repeated search during optimization, described in the following section. We find that this works better than using either MDH or PnP alone and we give implementation details of MDH and PnP in §4 of the appendix.

3.2.3 Optimization process and pseudo-code

Our optimization process is summarized in pseudo-code in Algorithm 1. Each initialization from the initialization set is processed (either in parallel or sequentially) and local optimization is performed in two steps (lines 7 and 8). At line 7, deformation is optimized with focal length fixed, and at line 8, they are both optimized jointly. These two steps are used to improve convergence especially when the initial focal length is far from the true solution. We implement local optimization with Gauss-Newton with backtracking line search until some termination criteria are satisfied, denoted by T_1 and T_2 respectively. When all initializations have been processed, the solution with lowest cost is taken, and a final refinement is performed with local optimization using termination criteria T_3.

To prevent unnecessary repeated search from different initializations, we maintain a search history $\mathcal{H}$ that holds all the solutions that have been found from a previous initialization (line 9). During the local optimization stages (lines 7, 8 and 13), we continually measure the distance of the current estimate $\hat{f}$ and $\hat{\boldsymbol{\theta}}$ to the closest member of $\mathcal{H}$ using a distance function $d_{\mathcal{H}}((\hat{\boldsymbol{\theta}}, \hat{f}), \mathcal{H})$. We terminate local optimization early if $d_{\mathcal{H}}((\hat{\boldsymbol{\theta}}, \hat{f}), \mathcal{H}) \leq \tau_{\mathcal{H}}$ where $\tau_{\mathcal{H}}$ is a threshold. The distance function is designed to tell us when the current estimate is likely to converge on a solution that already exists in $\mathcal{H}$ (and therefore when we should terminate local optimization). We measure distance in terms of surface normal dissimilarity as follows:

$$d_{\mathcal{H}}((\boldsymbol{\theta}, f), \mathcal{H}) = \min_{(\boldsymbol{\theta}', f') \in \mathcal{H}} \max_{t \in [1,T]} \text{abs}\left(\measuredangle\left(\boldsymbol{n}_t(\boldsymbol{\theta}), \boldsymbol{n}_t(\boldsymbol{\theta}')\right)\right) \tag{13}$$

where $\boldsymbol{n}_t(\boldsymbol{\theta})$ is the surface normal for triangle t generated by $\boldsymbol{\theta}$, and $\measuredangle(\boldsymbol{a}, \boldsymbol{b})$ is the angle in degrees between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

The following conditions are used in the termination criteria:

Algorithm 1 fSfT Optimization

Require:
$\{(f_1, \boldsymbol{\theta}_1), \ldots (f_S, \boldsymbol{\theta}_S)\}$ ▷ initialization set
$c(\boldsymbol{\theta}, f) : \Theta \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ ▷ cost function

function FSFT_OPTIMIZE($\{(f_1, \boldsymbol{\theta}_1), \ldots (f_S, \boldsymbol{\theta}_S)\}, c$)
  $c^* \leftarrow \infty$ ▷ lowest cost found so far
  $\mathcal{H} \leftarrow \emptyset$ ▷ search history
  $f^* \leftarrow 0$, $\boldsymbol{\theta}^* \leftarrow \mathbf{0}$ ▷ best solution with cost c^*
  for $s \in [1, S]$ **do**
    initialize estimates: $\hat{f} \leftarrow f_s, \hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}_s$
    locally optimize c w.r.t. $\hat{\boldsymbol{\theta}}$ until stopping criteria T_1 satisfied.
    locally optimize c w.r.t. $\hat{\boldsymbol{\theta}}$ and $\hat{f}$ until stopping criteria T_2 satisfied.
    update history: $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\hat{f}, \hat{\boldsymbol{\theta}})\}$
    if $c(\hat{f}, \hat{\boldsymbol{\theta}}) < C^*$ **then**
      $(f^*, \boldsymbol{\theta}^*) \leftarrow (\hat{f}, \hat{\boldsymbol{\theta}})$
      $c^* \leftarrow c(f^*, \boldsymbol{\theta}^*)$
  Final refinement: locally optimize c initialized with $(f^*, \boldsymbol{\theta}^*)$ until stopping criteria T_3 satisfied.
  return f^* and $\boldsymbol{\theta}^*$

S1 **Maximum iterations**: The number of iterations τ_{step} has been performed
S2 **Small parameter update**: The relative change of all unknowns is below a threshold τ_Δ
S3 **Small cost update**: The relative change of c is below a threshold τ_c
S4 **Out-of-bounds focal length**: $\hat{f}$ is out of bounds: $\hat{f} \leq f_{min}$ or $\hat{f} \geq f_{max}$
S5 **Repeated search**: The current solution is similar to one already in the search history: $d_{\mathcal{H}}((\hat{\boldsymbol{\theta}}, \hat{f}), \mathcal{H}) \leq \tau_{\mathcal{H}}$.

S1-S3 are standard in local optimization. S4 is used to terminate early if optimization is converging on a focal length solution that is clearly wrong. Normally this happens either when the problem is degenerate or when optimization has been very poorly initialized.

The termination criteria T_1, T_2 and T_3 in Algorithm 1 are instantiated by defining thresholds τ_{step}, τ_Δ, τ_c, f_{min}, f_{max} and $\tau_{\mathcal{H}}$. The same values are used in all experiments and are given in Table 1 of the Appendix. We use $f_{\text{min}} = 0.1w$ and $f_{\text{max}} = 1000w$ in all cases, where w is the image width. We highlight why the bounds are different to the focal length bounds defined in §3.2.2. Those bounds concerned the sampling range for initializing focal length, with opening angles between 20° and 100°. However, there could be cases where the true focal length lies out of these bounds. For that reason, the range of permissible focal lengths during optimization is larger than the range considered for initialization. The range of $0.1w$ to $1000w$ is arbitrary, and it is probably overly broad in practice.

4 Experimental results

4.1 Datasets

We evaluate our method on 12 public datasets of quasi-isometrically deforming objects from the existing SfT literature (Figure 4), with total of 310 test images. These datasets represent a range of real-world challenges, in particular strong deformation and weak texture. Each dataset has a set of images of a deformable object, a template, and point correspondences in each image. We give full dataset details, including the number of images per dataset, focal lengths and number of points correspondences in §5 of the Appendix.

The first four datasets ('Spider-man' [9], 'Kinect paper' [53], 'Van Gogh paper' [43] and 'Hulk' [7]) are of smoothly deforming paper sheets that are relatively well textured. The Spider-man dataset has images taken at 9 different focal lengths with opening angles from 24.8° to 65.3°. The other datasets have a fixed focal lengths with opening angles 62.4°, 44.5°, and 66.1° respectively. The next four datasets ('Cap' [2], 'Bedsheet' [41], 'Kinect t-shirt' [53] and 'Handbag' [18]) are of deforming objects made of cloth. These datasets have fixed focal lengths with opening angles of 53.3°, 44.5°, 62.4° and 50.9° respectively. The next two datasets ('Floral paper' [18] and 'Fortune teller' [18]) are of creased paper objects with sparse texture, making them especially difficult objects. The next dataset ('Bending cardboard' [44]) is of a smoothly deforming cardboard sheet with very sparse texture. The final dataset ('Pillow cover' [18]) is of a deforming pillow cover made of fabric with sparse texture. Outlier-fee point correspondences are provided with six of the datasets (Spider-man, Hulk, Handbag, Floral paper, Fortune teller and Pillow cover). We generated point correspondences for the other datasets ourselves. The images from the Kinect paper, Van Gogh paper, Bedsheet, Kinect t-shirt, and Bending cardboard are from video clips, so point correspondences were make by tracking keypoints over time. We used KLT feature tracking [51], which worked well in practice because the objects deform relatively smoothly with limited motion blur. Forward-backwards consistency checking was used to detect and remove outliers tracks. Point correspondences for the cap dataset were computed by hand using an interactive graphical user interface.

Several of the datasets have images where the object is flat and facing the camera (the Kindet paper, Van Gogh paper, Bedsheet, Kinect t-shirt and Bending cardboard datasets). These cases are unsolvable because of the ambiguity between focal length and surface depth. We therefore exclude an image from the evaluation if all surface normals approximately align with the optical axis (we use a threshold of 5°).

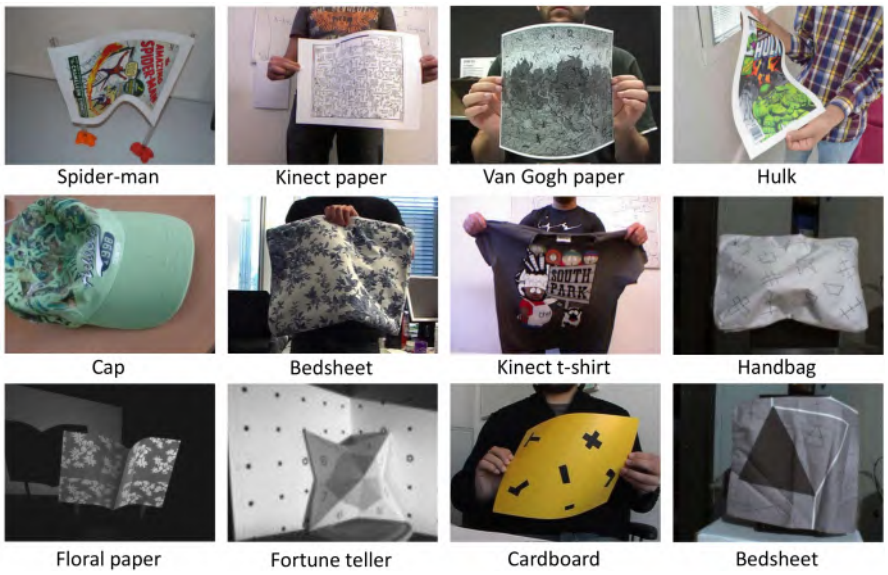


Fig. 4: The 12 public datasets used for evaluation. One representatative image per dataset is shown.

4.2 Evaluation metrics

We evaluate solution accuracy for each image with two metrics. The first is Focal Length Percentage error (FLPE) and the second is Shape Error (SE). FLPE is defined as follows:

$$\text{FLPE}(\hat{f}, f) \stackrel{\text{def}}{=} 100 \times \frac{\left|\hat{f} - f^{gt}\right|}{f^{gt}} \tag{14}$$

where $\hat{f}$ is the estimate focal length and f^{gt} is the ground-truth focal length (all datasets provide ground-truth focal lengths). SE is computed for all datasets with ground truth (Spider-man, Kinect paper, Hulk, Cap, Kinect t-shirt, Handbag, Floral paper, Fortune teller and Bedsheet) as follows. For each image and each point correspondence, we evaluate the Euclidean distance between the reconstructed 3D point in camera coordinates $\hat{\boldsymbol{q}} \in \mathbb{R}^3$ and ground truth $\boldsymbol{q}^{gt} \in \mathbb{R}^3$. The *Reconstruction Error* (RE) is defined as $\text{RE}(\hat{\boldsymbol{q}}, \boldsymbol{q}^{gt}) \stackrel{\text{def}}{=} \|\hat{\boldsymbol{q}} - \boldsymbol{q}^{gt}\|$.

RE has been used extensively for SfT evaluation. However, it has an important limitation for fSfT evaluation. We now explain this, motivating the use of an adapted metric, called the Shape Error (SE). The isometric prior penalizes stretching and shrinking of the template. This fixes the scale ambiguity that would otherwise exist between f and the template's scale. However, in cases where the perspective effects are weak (*i.e.* when the viewing rays of the point correspondences were approximately parallel), an ambiguity emerges between f and the template's average depth $\bar{d}$. That

is, one can obtain a similar image by reducing f by a scale factor α and increasing $\bar{d}$ by $\frac{1}{\alpha}$. This ambiguity is well-known in the case of rigid objects and was previously identified in fSfT [4]. In terms of evaluation, this highlights a shortcoming of RE: in cases with weak perspective effects, it may be possible to reconstruct the shape of the template accurately, but not possible to precisely determine $\bar{d}$ and f. A method able to accurately reconstruct shape in these cases would receive a high RE error, which would be unfair.

To handle this, we adapt RE to make it insensitive to a global shift in average depth. First, a least-squares translation t_z is computed along the camera's optical axis to align the reconstructed 3D point correspondences with their ground truths. SE is then computed as follows:

$$\mathrm{SE}\left(\hat{\boldsymbol{q}}, \boldsymbol{q}^{gt}\right) \stackrel{\text{def}}{=} \frac{100}{S} \times \left\|\hat{\boldsymbol{q}} + t_z - \boldsymbol{q}^{gt}\right\|_2 \tag{15}$$

The denominator S is used to make SE independent of the template's size. We set this as the maximum spatial range of the template's rest shape with respect to its 3 spatial coordinates. Consequently an SE of 1 corresponds to approximately 1% of the template's size. We emphasize that this is to help interpret results, and it is not linked to a reconstruction scale ambiguity.

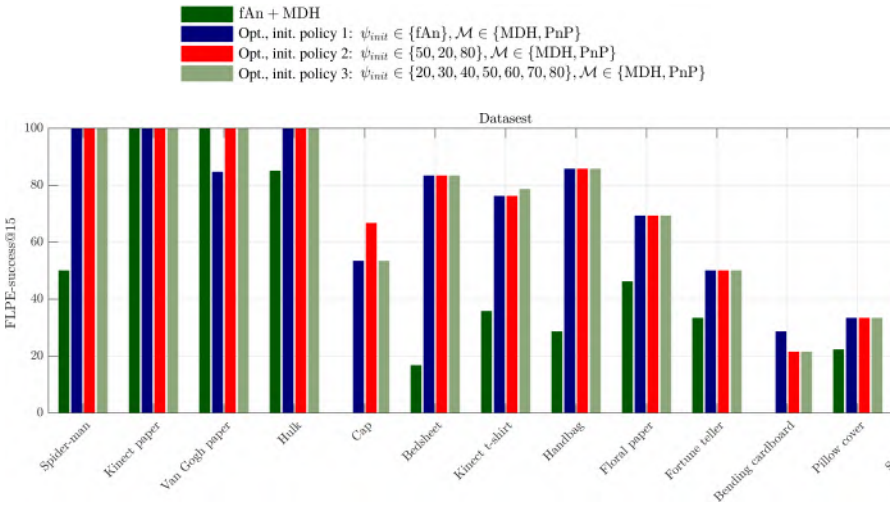
4.3 Success rates

We compare performance using *success rates*, which are the proportion of images for which a method returns a solution with an error less than a threshold τ. We use FLPE-success@τ to denote the FLPE success rate using a threshold τ. Similarly, we use SE-success@τ to denote the SE success rate. We use a few different thresholds to assess how often very accurate results are achieved (smaller τ) and how often results in the right 'ballpark' are achieved (larger τ). Success rate was selected because it is a robust statistic, required to handle the fact that in some instances fSfT can be weakly-posed, leading to extreme FLPE and SE values.

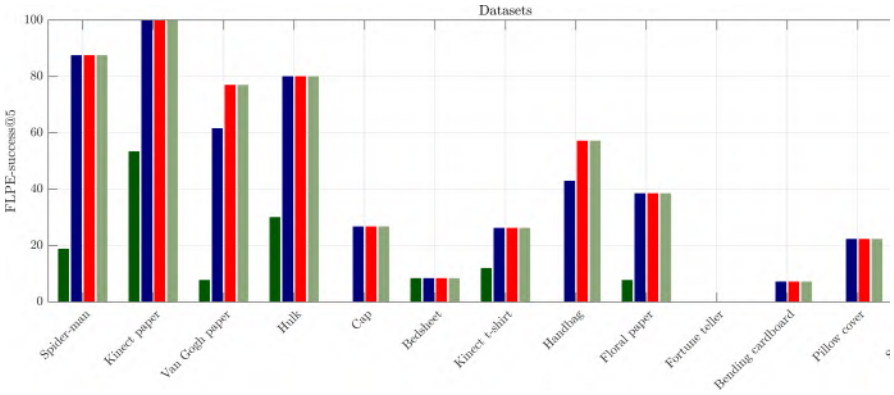
4.4 FLPE and SE results

We evaluate our approach with three different policies for creating the initialization set. There is a trade-off between using a larger initialization set (increasing computational cost) and a smaller initialization set (reducing computational cost but potentially reducing the chance of finding the global optimum). In this section we test three initialization policies that are specified by a set of lens opening angles, defined as Ψ_{init}, and a set of closed-form SfT methods, defined as $\mathcal{M}$:

- Policy 1: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} = \{\mathrm{MDH}, \mathrm{PnP}\}$
- Policy 2: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} = \{\mathrm{MDH}, \mathrm{PnP}\}$



(a)

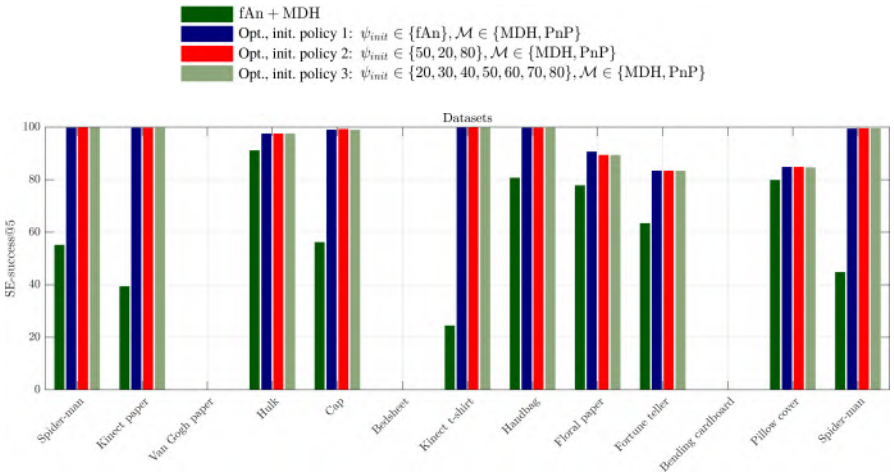


(b)

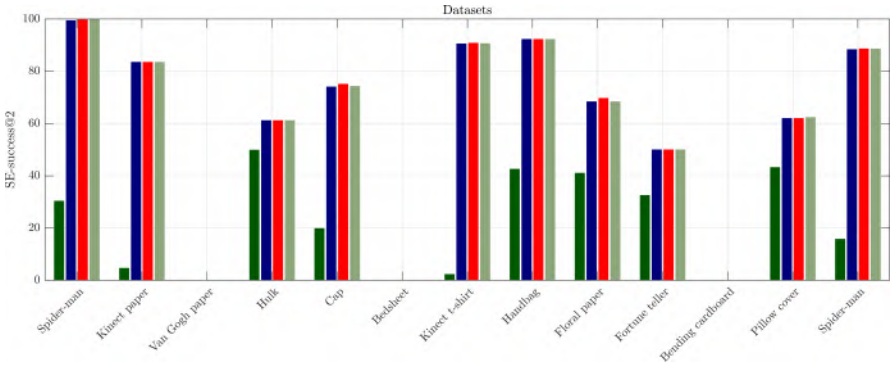
Fig. 5: Focal Length Percentage Error (FLPE) results for the analytical method (denoted as 'fAn+MDH') and optimization-based method (denoted as 'Opt.') using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods $\mathcal{M}$ used to initialize deformation. (a) shows FLPE success rates at 15% and (b) shows FLPE success rates at 5%.

- Policy 3: $\Psi_{init} = \{20, 30, 40, 50, 60, 70, 80\}$, $\mathcal{M} = \{\text{MDH, PnP}\}$

We use ψ^{An} to denote the opening angle estimated by the analytical method . For each focal length initialization, we generate two deformation initializations using MRD and PnP. The number of initializations S for policies 1,2 and 3 are therefore 2, 6, and 14 respectively. We compare results against the analytical method to solve focal



(a)



(b)

Fig. 6: (a-b) shows the Shape Error (SE) of the method (denoted as 'fAn+MDH') and optimization-based method (denoted as 'Opt.') using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods $\mathcal{M}$ used to initialize deformation. Van Gogh paper, Bedsheet and Bending cardboard dataset have no errors because they do not contain ground truth 3D information. For this reason there are no bars associated with them.

length, combined with the MDH method to compute deformation. This combination is denoted as **FAn+MDH**.

We consider FLPE below 15% to be a good result for fSfT and FLPE below 5% to be an exceptional result. Thus, we evaluate both FLPE-success@15 and FLPE-success@5, shown in Figure 5(a) and (b) respectively. Similarly, Figure 6(a) and Figure 6(b) show SE-success@5% and SE-success@2% respectively.

We first consider FLPE-success@15 in Figure 5(a). We observe the following points:

1. The performance of **FAn+MDH** is very good for the Kinect and Van Gogh Paper dataset, where FLPE-success@15 is 100.0%. **FAn+MDH** achieves a relatively high FLPE-success@15 of 80.0% for the Hulk dataset. Recall that these dataset are smoothly deforming paper sheets with dense texture. These results indicate that the analytical method can estimate the focal length well in these cases.
2. For the other dataset (Spider-man, Cap, Bedsheet Kinect t-shirt, Handbag, Floral paper and Bending cardboard), **FAn+MDH** performs relatively poorly and much worse than optimization-based method (with any initialization policy). Indeed for the Cap and Bending cardboard datasets **FAn+MDH** has FLPE-success@15 of 0.0%: Therefore it was not able to find a focal length within 15% of ground truth in any of the images of those dataset. These results indicate that the analytical method does not work well in more difficult cases when texture is sparse and/or when deformation is complex.
3. There is little difference between policies 2 and 3. They achieve FLPE-success@15 of 100% for datasets with smoothly deforming, well textured objects (Spider-man, Kinect paper, Van Gogh and Hulk datasets). They achieve FLPE-success@15 above 60% for the Cap, Bedsheet, Kinect t-shirt, Handbag and Floral paper datasets. Considering the challenges associated with these datasets including strong complex and non-isometric deformation, this is a strong result. Furthermore, it indicates that *(i)* initialization with three fixed focal length samples (short, medium, far) achieves similar or better performance compared to policy 1, and *(ii)* there appears to be very little benefit in using more than three focal length samples.
4. For the Cap dataset, policy 2 has a higher success rate than policy 3. This may seem surprising because policy 3 initializes with more starts, including all starts in policy 2, so we may think policy 3 should always do better. This is not necessarily the case. The reason is because there exists an image in the Cap dataset with a spurious solution that has a lower cost compared to the true solution. This was located using policy 3 but not with policy 2. However, because in all other datasets the performance of policies 2 and 3 are practically identical, we see that this kind of events is extremely rare.
5. Performance is clearly strongly dataset dependent. The Bending cardboard and Pillow cover datasets have the lowest performance among all dataset. Recall that these datasets are very challenging because the Bending cardboard has extremely sparse correspondences, and the Pillow cover has many views that are approximately fronto-parallel (making fSfT poorly conditioned).

We now consider FLPE-success@5 in Figure 5(b). We observe the following:

6. Because of the much more stringent success threshold of 5%, we observer lower success rates for most datasets. Nevertheless, 100% success rate is achieved by the optimization-based method for the Kinect paper dataset with all initialization policies. Success rates above 78% are achieved for the Spider-man, Van Gogh

paper and Hulk dataset with the optimization-based method and all initialization policies. This shows that we can solve fSfT with the optimization-based method and achieve very high accuracy (FLPE below 5%) for strongly isometric and well-textured objects.
7. For less isometric and/or weakly textured objects (those other than Spider-man, Kinect paper, Van Gogh paper and Hulk datasets), it is very challenging to solve fSfT consistently with high accuracy and FLPE below 5%.
8. Unlike FLPE-success@15, **FAn+MDH** achieves significantly lower success rates for FLPE-success@5 with the Kinect paper, Van Gogh and Hulk datasets compared to the optimization-based method. This indicates that the analytical method can achieve focal lengths in the right ballpark (< 15% error) for isometric well-textured objects, but it is not as precise as the optimization-based method.

We now consider Shape Error (SE) shown in Figure 6. Recall that Van Gogh, Bedsheet and Bending cardboard datasets do not have ground truth 3D information so SE cannot be measured. We observe the following points:

9. Very similar SE results are achieved for the different initialization policies for each dataset. This agrees with the FLPE results.
10. SE-success@5 is above 80% for all datasets with the optimization-based method with all initialization policies. Recall that an SE of 5 occurs when the Euclidean error at each reconstructed point is within 5% relative to the size of the object template. Thus, SE-success@5 above 80% is a strong result.
11. Unlike the FLPE results, the simpler dataset (Spider-man, Kinect paper and Hulk) do not have systematically better SE results compared to other datasets. Indeed, the Pillow cover dataset, which had the second lowest FLPE success rate among all dataset, has very similar SE-success@2 as the Hulk dataset. This highlights the intrinsic difficulty of the Pillow-cover dataset. It has little variation in depth, leading to weak perspective effects. This causes an ambiguity between the distance of the object to the camera and focal length, explaining why its shape can be estimated well but the focal length cannot.
12. The shape error of **FAn+MDH** is similar to the optimization-based method for the Hulk dataset, but it is generally significantly worse for the other datasets. Recall that the Hulk dataset has relatively strong perspective effects thanks to the short focal length and deformation is smooth with well-distributed point correspondences.

4.5 Results visualizations

We visualize results of the optimization-based method (policy 2) in Figures 8, 9, 10. The figures are laid out in the same way, with four representative images per dataset. Below the images are the corresponding deformation solutions rendered from the camera's viewpoint and shaded to show the shapes. We give the corresponding

FLPE with each image below each render. For all datasets with ground-truth, point correspondences are colored-mapped using their shape error. For datasets without ground-truth, point correspondences are colored in green. We can see the method has been able to estimate deformations very well, especially considering the complex deformations exhibited by the handbag, floral paper, fortune teller and cap datasets. Note that the floral paper and fortune teller datasets have strongly creased objects with relatively few point correspondences. We see our method is able to recover the general shape well despite these challenges. In some datasets, a relatively large FLPE was obtained (mainly the cardboard and pillow-cover datasets), while the shape appears to be reconstructed well. These results suggest fSfT is weakly-posed in those cases, with an ambiguity between focal length and surface distance,

4.6 Convergence basin

As only a few initializations are required with widely-spread focal lengths to achieve good results, this suggests that the cost function's convergence basin is relatively wide. We provide a graphical illustration of the convergence basin in figure 7 using the first image from the Floral paper dataset as a typical example. These graphs are generated by sampling 20 focal lengths with opening angles ranging from 1° (extreme tele-photo) to 170° (extreme wide-angle), including the ground-truth focal length. For each sample, Algorithm 1 is run using two SfT methods (MDH and PnP), and the cost of the final solution returned by the algorithm is recorded. Figure 7 (top) shows the final costs plotted against the initial focal lengths (expressed as opening angles). Figure 7 (bottom) shows the FLPE of the final solution plotted against initial focal length. The ground truth focal length is illustrated by the vertical lines. For initial focal lengths between 1° and 100°, the same FLPE and final cost are achieved (FLPE of 1.2%). This clearly demonstrates a relatively wide convergence basin with respect to initial focal length.

4.7 Results summary.

The results show that the optimization-based method generally achieves far better accuracy compared to the analytical method. Therefore, in practical applications, the analytical method should be considered as a way to initialize the optimization-based method (as done in policy 1), and not as a competitive approach. Initialization with the analytical method appears to achieve similar accuracy compared to initialization with focal length sampling (policies 2 and 3). Furthermore, there appears to be no benefit in initializing with more than three focal length samples (policy 2 versus policy 3). We have implemented the method in Matlab a standard x64 Linux workstation computer, and it run in approximately 5 seconds using initialization policy 2 (6 initializations). We confidently believe it can be run in real-time with a C++ and

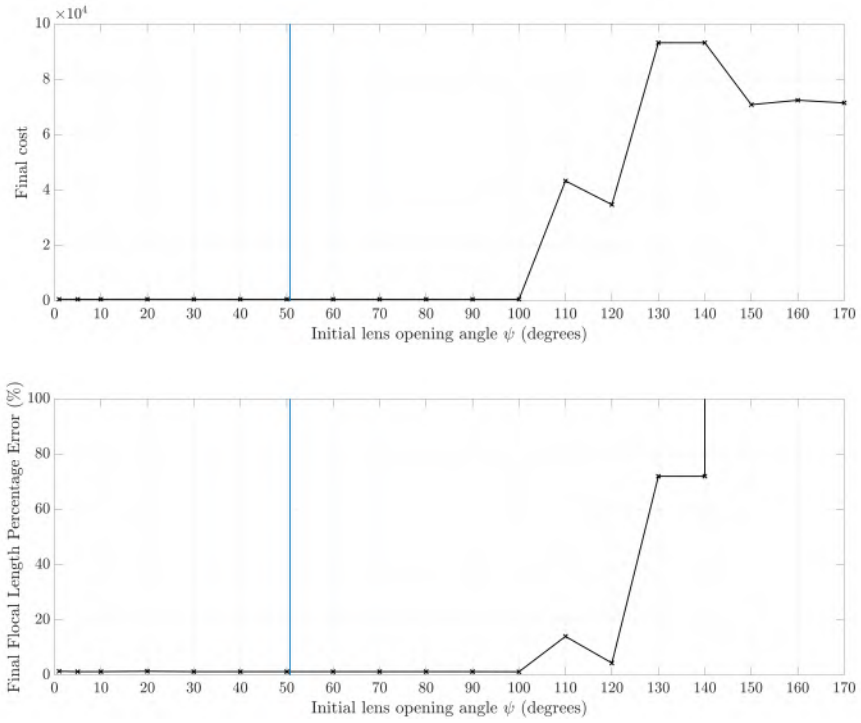


Fig. 7: Graphical illustration of the cost function's convergence basin with respect to initial focal length (expressed in opening angles). 20 different initial focal lengths are tested using Algorithm 1, ranging from 1° (extreme tele-photo) to 170° (extreme wide-angle). The top graph shows final cost and the bottom graph shows final FLPE against initial focal length respectively. ground-truth focal length is shown as a vertical line.

CUDA implementation based on [11]. A more detailed analysis of computation time with different initialization policies is provided in §7 of the appendix.

4.8 Additional initialization sensitivity experiments

We conducted an additional experiment to further investigate the impact of the initialization policy on computation time and accuracy. The public datasets were limiting for this because the range of opening angles was not very large (24.8° to 65.3°). Consequently, one focal length sample at 50.0° worked very well in practice. We therefore augmented the datasets by adding simulated digital zoom variability. We also added synthetic point correspondence noise. Because of space restrictions,

we give further details for the augmented datasets and results in §6 of the appendix, and we summarize the findings here:

13. In the original datasets (without simulated digital zoom or point correspondence noise), initialization with the analytical solution had very similar performance as initialization with one focal length sample at 50.0°. However, when zoom augmentation was added, initialization with the analytical solution outperformed one focal length sample at 50.0°.
14. Initializing with three opening angles of 20.0°, 50.0° and 80.0° gives slightly better performance compared to initializing with analytical solution.
15. There is a clear benefit in using both PnP and MDH for initializing deformation, compared to using just MDH. Because the optimization algorithm had an early termination mechanism to avoid repeated search, the computational cost of using both PnP and MDH is only approximately 50% greater than just using MDH.
16. There is little benefit in using more than three focal length samples.

We therefore recommend initializing with three opening angles at 20.0°, 50.0° and 80.0° as the default, and using both PnP and MDH for initializing deformation for each opening angle. We refer the reader to §6 of the appendix for the experiment details and quantitative performance statistics.

4.9 Isometric weight sensitivity

We end the experiments section by investigating the sensitivity of the isometric weight λ_{iso} on solution accuracy, and showing the positive effects our normalization techniques have had in restricting the range of good isometric weights. We test 10 different isometric weights ranging from $\lambda_{\text{iso}} = 1000 \times \lambda'_{\text{iso}}$ to $\lambda_{\text{iso}} = \frac{1}{100} \times \lambda'_{\text{iso}}$, given in the legend in Figure 11. For each isometric weight, we optimize the cost function using Algorithm 1, initialized using initialization policy 3 (3 opening angles at 20.0°, 50.0° and 80.0°, and using both PnP and MDH for initializing deformation). The results are shown in Figure 11. We see that λ_{iso} strongly influences performance, where excessively large or small weights compared to λ'_{iso} lead to poor performance. This is seen for all dataset groups and performance metrics. Performance is generally uni-modal in λ_{iso}, and peaking at, or very close to λ'_{iso}. Using a fixed weight in the range $\frac{1}{2}\lambda'_{\text{iso}} \leq \lambda_{\text{iso}} \leq 2\lambda'_{\text{iso}}$ leads to similar performance where $\lambda_{\text{iso}} = \lambda'_{\text{iso}}$ is generally the best. The results also shows the importance of normalization. If for example we changed the template's size by a factor of 10, then without scale normalization, the influence of the isometric cost (Equation 3.1.2) would be 10 times greater, and λ'_{iso} would not longer be a good weight: results would be equivalent to using a weight of $10\lambda'_{\text{iso}}$, with much worse results. Similarly, if we were to not normalize the number of point correspondences N, a change in N by a factor of 10 would also lead to significantly worse performance equivalent to $\lambda_{\text{iso}} = \lambda'_{\text{iso}}/10$. These results strongly indicate our normalization strategies are effective, because the same isometric weight can be used in all datasets with near optimal performance.

5 Conclusion

fSfT is a challenging problem that must be solved in real-world SfT applications when the camera's focal length (a key intrinsic) is unknown. We have modeled and solved fSfT with a large-scale non-convex cost function that is optimized with multi-start Gauss-Newton with a mechanism to avoid repeated search. The method has received a relatively large evaluation comprising 12 public datasets with various challenges that include sparse texture and creased surfaces. The method is considerably more accurate compared to the analytical solution [4] in all datasets. It uses a carefully-designed cost function with normalization techniques that allow the same deformation weights to be used for all datasets, making our solution practical for real-world use without the need to tune cost weights at run-time. In future work, we aim to combine the method with deep learning-based dense image registration and to extend the method to multiple views sharing a common focal length. We also aim to study the practical and theoretical limits of solving SfT with more unknown intrinsics.

Matching quality is an important factor that affects results, and a deeper analysis would be useful. There are three main aspects to match quality: match density, match noise and incorrect matches (also called 'outliers' or 'mis-matches'). By including objects with different amounts of texture, match density is varied in our experiments. As shown in Figure 13, the poorly textured objects (Floral paper, Fortune teller, Bending cardboard and Pillow cover) tend to have higher reconstruction error than well-textured objects (with denser matches). Fortune teller represents a 'breakpoint' because of the combination of very sparse matches and highly discontinuous deformation (folds). The matches do not provide enough motion information for consistent and accurate focal length estimation. This is shown in Figure 5a, where focal length estimates with 15% error were attained in 50% of the Fortune teller images. Concerning mis-matches, as with most prior works in SfT, we assume that the majority of mis-matches have been detected and removed by a dedicated 'outlier rejection' method, then the remaining matches are passed to our fSfT method. In particular, [38] and its extension to arbitrary template meshes [36] has shown excellent robustness in a tracking-by-detection setting. A dedicated evaluation of the robustness of the complete pipeline, using different outlier rejection methods, would be interesting to perform for future work. We also aim to conduct a more detailed analysis of the influence of point noise on reconstruction accuracy.

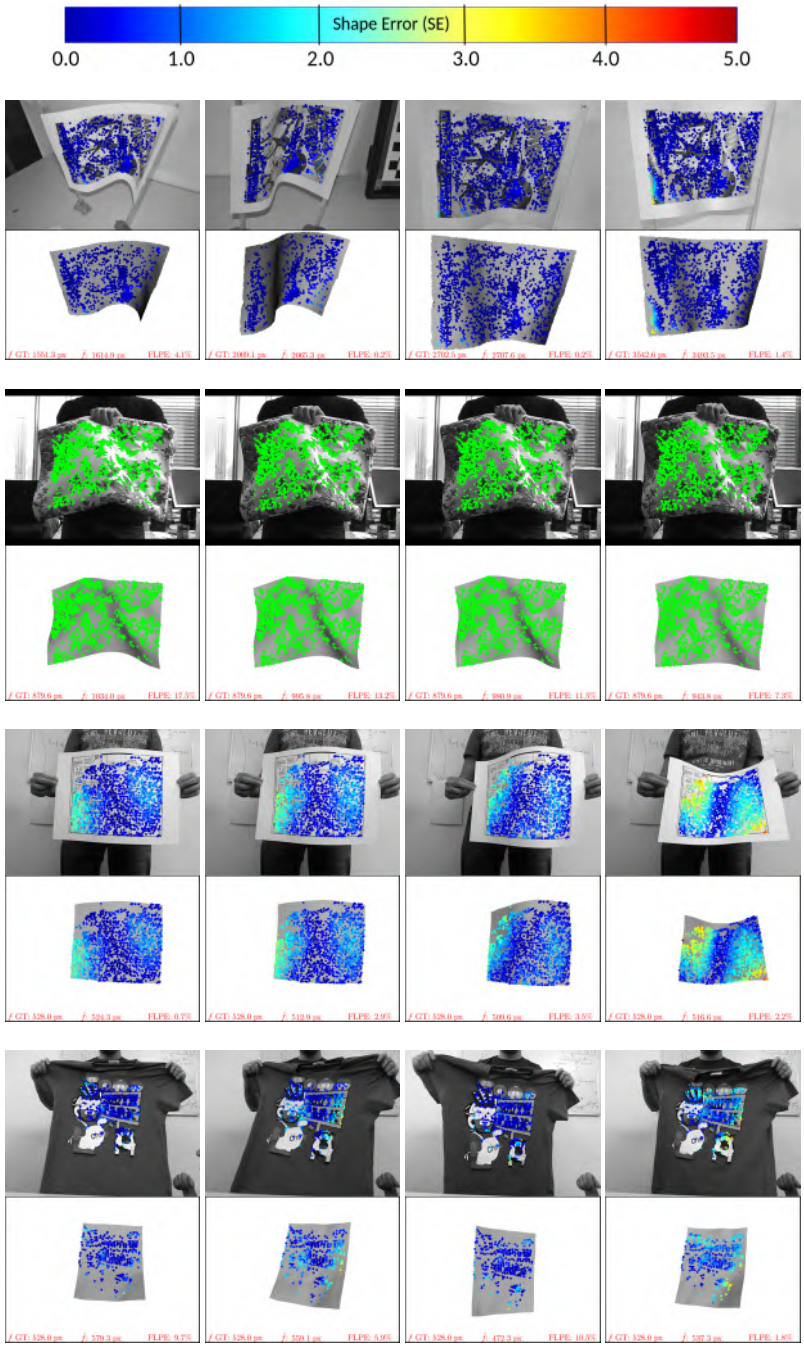


Fig. 8: fSfT results with our method. Four representative images from the Spiderman, Bedsheet, Kinect paper and Kinect t-shirt datasets are shown (top to bottom). The estimated deformations are shown below each image as shaded renders. Point correspondences are visualized and color-mapped according to their SE. The Bedsheet dataset does not have ground-truth 3D, so point correspondences are shown in green. Focal length information and FLPE is given below each render.

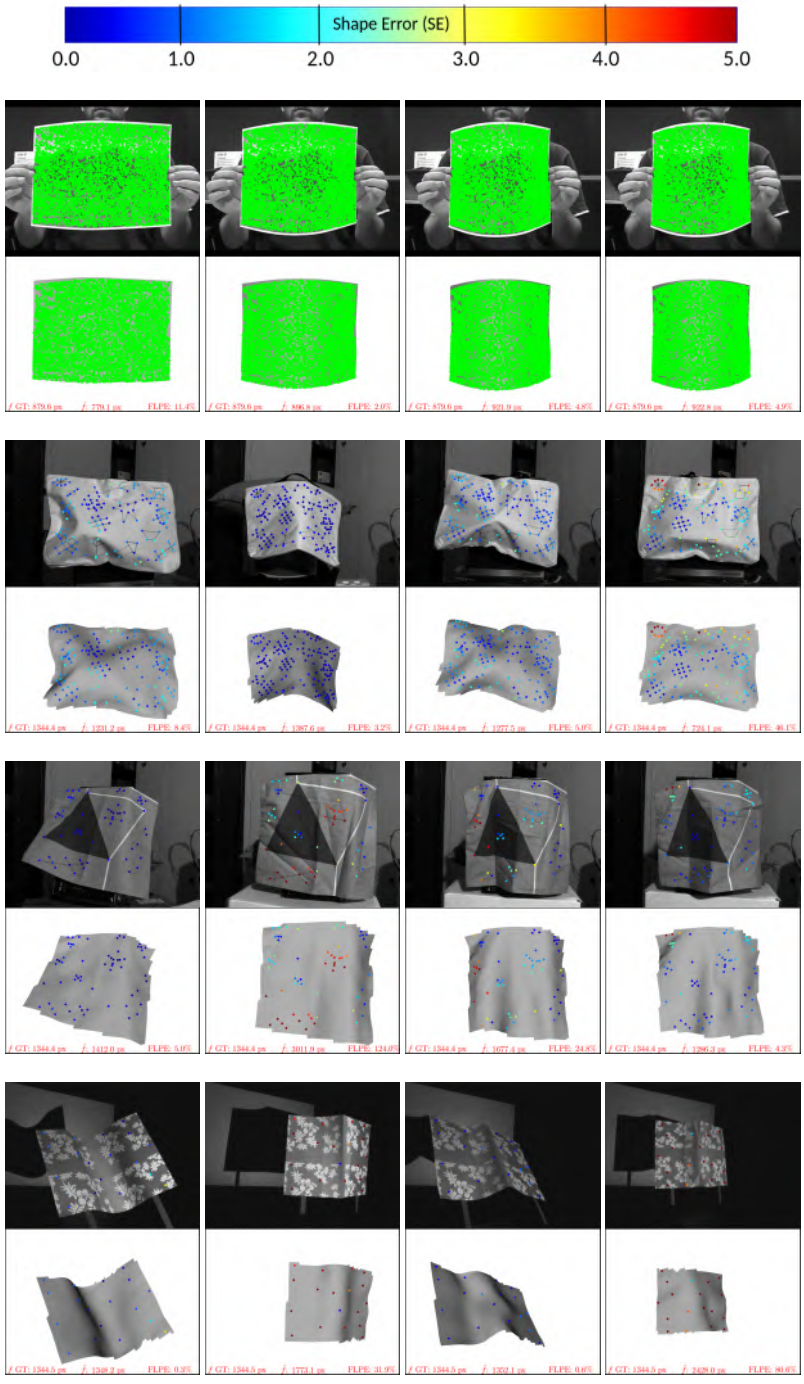


Fig. 9: fSfT results with our method. Four representative images from the Van Gogh paper, Handbag, Pillow cover and floral paper datasets are shown (top to bottom). The estimated deformations are shown below each image as shaded renders. Point correspondences are visualized and color-mapped according to their SE. The Van Gogh paper dataset does not have ground-truth 3D, so point correspondences are shown in green. Focal length information and FLPE is given below each render.

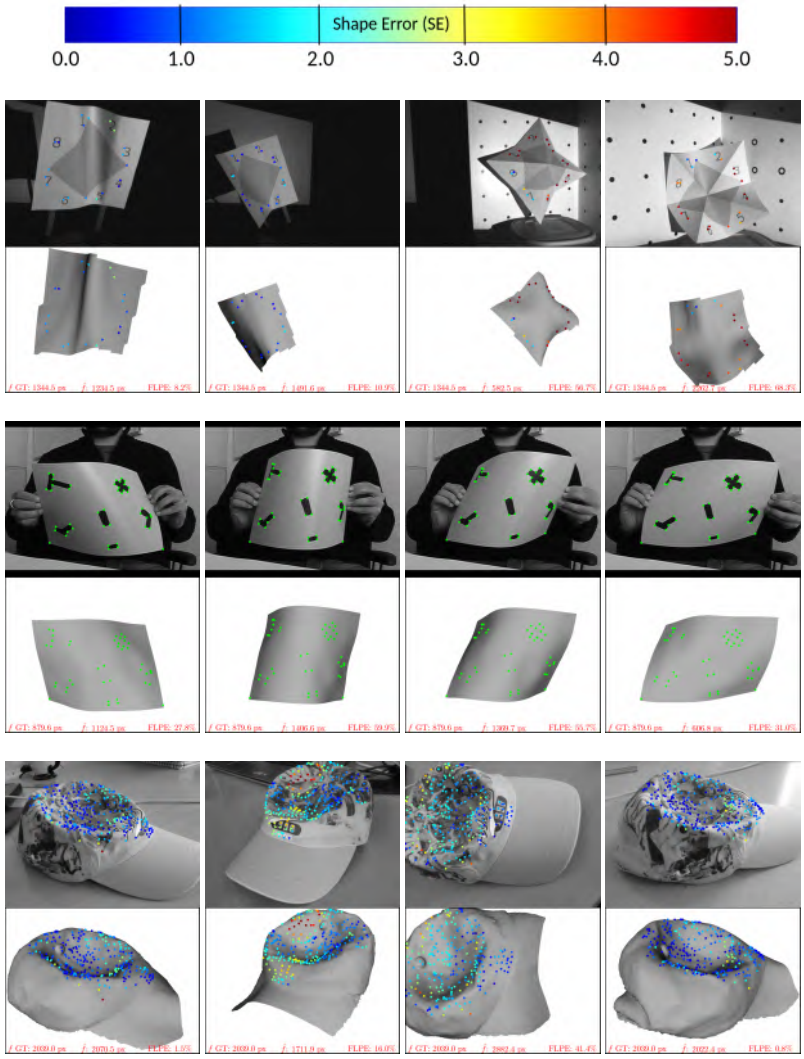


Fig. 10: fSfT results with our method. Four representative images from the Fortune teller, Bending cardboard and Cap datasets (top to bottom). The estimated deformations are shown below each image as shaded renders. Focal length information and FLPE is given below each render.

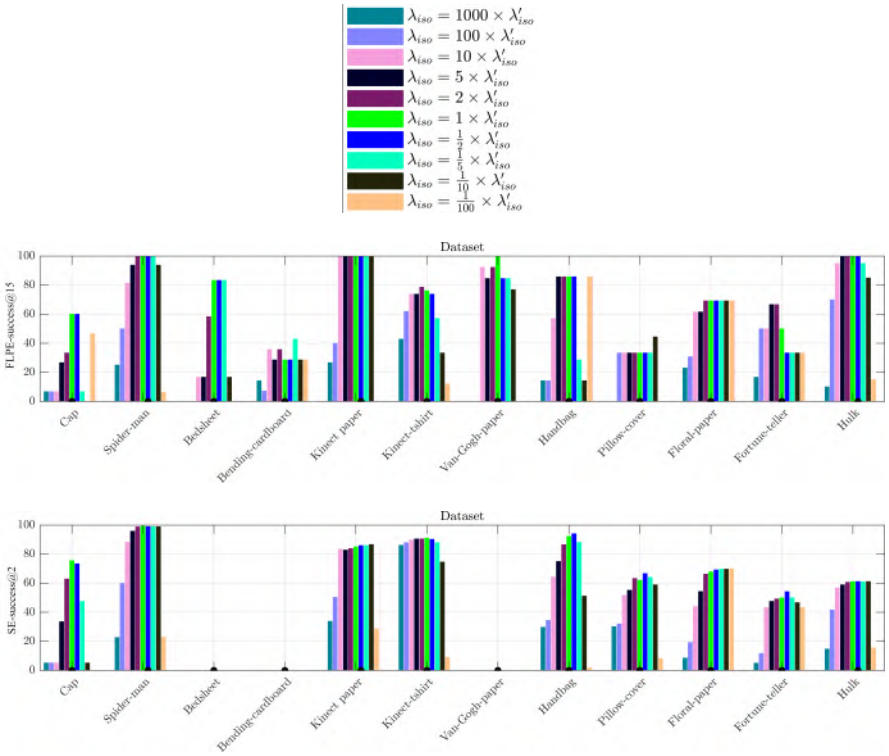


Fig. 11: FLPE performance (top) and SE performance (bottom) with different isometric weights. Note that Bedsheet, Bending cardboard and Van Gogh paper do not have ground truth 3D, so SE for those datasets cannot be measured.

References

1. Bansal, A., Russell, B., and Gupta, A. Marr revisited: 2d-3d alignment via surface normal prediction. pp. 5965–5974.
2. Bartoli, A., and Collins, T. Template-based isometric deformable 3d reconstruction with sampling-based focal length self-calibration. *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), 1514–1521.
3. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., and Pizarro, D. Shape-from-template. ieee *Transactions on Pattern Analysis and Machine Intelligence 37*, 10 (Oct 2015), 2099–2118.
4. Bartoli, A., Pizarro, D., and Collins, T. A robust analytical solution to isometric shape-from-template with focal length calibration. In *International Conference on Computer Vision (ICCV)* (2013).
5. Brunet, F., Hartley, R., and Bartoli, A. Monocular Template-Based 3D Surface Reconstruction: Convex Inextensible and Nonconvex Isometric Methods. *Computer Vision and Image Understanding 125* (August 2014), 138–154.
6. Brunet, F., Hartley, R., Bartoli, A., Navab, N., and Malgouyres, R. Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces. In *Asian Conference on Computer Vision (ACCV)* (2010).
7. Chhatkuli, A., Pizarro, D., and Bartoli, A. Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *British Machine Vision Conference (BMVC)* (2014).
8. Chhatkuli, A., Pizarro, D., and Bartoli, A. Stable Template-Based Isometric 3D Reconstruction in All Imaging Conditions by Linear Least-Squares. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).
9. Chhatkuli, A., Pizarro, D., Bartoli, A., and Collins, T. A stable analytical framework for isometric shape-from-template by surface integration. ieee *Transactions on Pattern Analysis and Machine Intelligence 39*, 5 (May 2017), 833–850.
10. Collins, T., and Bartoli, A. Infinitesimal plane-based pose estimation. *International Journal of Computer Vision 109*, 3 (2014), 252–286.
11. Collins, T., and Bartoli, A. Realtime Shape-from-Template: System and Applications. In *International Symposium on Mixed and Augmented Reality (ISMAR)* (2015).
12. Collins, T., Bartoli, A., Bourdel, N., and Canis, M. Robust, real-time, dense and deformable 3D organ tracking in laparoscopic videos. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2016).
13. Collins, T., Durou, J.-D., Gurdjos, P., and Bartoli, A. Single view perspective shape-from-texture with focal length estimation: A piecewise affine approach. *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)* (2010).
14. Eigen, D., and Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision (ICCV)* (2015), pp. 2650–2658.
15. Fuentes-Jimenez, D., Casillas-Perez, D., Pizarro, D., Collins, T., and Bartoli, A. Deep shape-from-template: Wide-baseline, dense and fast registration and deformable reconstruction from a single image, 2018. arXiv:1811.07791.
16. Fuentes-Jimenez, D., Pizarro, D., Casillas-Perez, D., Collins, T., and Bartoli, A. Texture-generic deep shape-from-template. *IEEE Access 9* (2021), 75211–75230.
17. Gallardo, M., Collins, T., and Bartoli, A. Can we Jointly Register and Reconstruct Creased Surfaces by Shape-from-Template Accurately? In *European Conference on Computer Vision (ECCV)* (2016).
18. Gallardo, M., Collins, T., and Bartoli, A. Dense Non-Rigid Structure-from-Motion and Shading with Unknown Albedos. In *International Conference on Computer Vision (ICCV)* (2017).
19. Garg, R., Kumar, B. V., Carneiro, G., and Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)* (2016).

20. Golyanik, V., Shimada, S., Varanasi, K., and Stricker, D. Hdm-net: Monocular non-rigid 3d reconstruction with learned deformation model.
21. Güler, R. A., Neverova, N., and Kokkinos, I. Densepose: Dense human pose estimation in the wild. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 7297–7306.
22. Ilic, S., Salzmann, M., and Fua, P. Implicit meshes for effective silhouette handling. *International Journal of Computer Vision 72* (2007), 159–178.
23. Ke, T., and Roumeliotis, S. I. An efficient algebraic solution to the perspective-three-point problem. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4618–4626.
24. Kimmel, R., and Sethian, J. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America 95* (08 1998), 8431–5.
25. Koo, B., Özgür, E., Le Roy, B., Buc, E., and Bartoli, A. Deformable Registration of a Preoperative 3D Liver Volume to a Laparoscopy Image Using Contour and Shading Cues. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2017).
26. Levi, Z., and Gotsman, C. Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation. ieee *Transactions on Visualization and Computer Graphics 21*, 2 (Feb. 2015), 264–277.
27. Liu, F., Shen, C., Lin, G., and Reid, D. I. Learning depth from single monocular images using deep convolutional neural fields. ieee *Transactions on Pattern Analysis and Machine Intelligence* (2016), 2024–2039.
28. Liu-Yin, Q., Yu, R., Agapito, L., Fitzgibbon, A., and Russell, C. Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading. In *British Machine Vision Conference (BMVC)* (2016).
29. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (Nov. 2004), 91–110.
30. Magnenat, S., Ngo, D., Zünd, F., Ryffel, M., Noris, G., Roethlin, G., Marra, A., Nitti, M., Fua, P., Gross, M., and Sumner, R. W. Live texturing of augmented reality characters from colored drawings. ieee *Transactions on Visualization and Computer Graphics 21* (2015), 1201–1210.
31. Malti, A., Bartoli, A., and Collins, T. A pixel-based approach to template-based monocular 3d reconstruction of deformable surfaces. In *International Conference on Computer Vision Workshops* (2011), pp. 1650–1657.
32. Martinez, J., Hossain, R., Romero, J., and Little, J. J. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision (ICCV)* (2017).
33. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. A comparison of affine region detectors. *International Journal of Computer Vision 65* (2005), 2005.
34. MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.
35. Ngo, T. D., Park, S., Jorstad, A. A., Crivellaro, A., Yoo, C., and Fua, P. Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. In *International Conference on Computer Vision (ICCV)* (2015).
36. Ostlund, J., Varol, A., Ngo, T., and Fua., P. Laplacian Meshes for Monocular 3D Shape Recovery. *European Conference on Computer Vision (ECCV)* (2012).
37. Perriollat, M., Hartley, R., and Bartoli, A. Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision 95*, 2 (Nov 2011), 124–137.
38. Pilet, J., Lepetit, V., and Fua, P. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision 76*, 2 (Feb 2008), 109–122.
39. Pizarro, D., and Bartoli, A. Feature-Based Deformable Surface Detection with Self-Occlusion Reasoning. *International Journal of Computer Vision 97*, 1 (March 2012), 54–70.
40. Pumarola, A., Agudo, A., Porzi, L., Sanfeliu, A., Lepetit, V., and Moreno-Noguer, F. Geometry-aware network for non-rigid shape prediction from a single view. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), IEEE Computer Society, pp. 4681–4690.

41. Salzmann, M., and Fua, P. Reconstructing sharply folding surfaces: A convex formulation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2009), pp. 1054–1061.
42. Salzmann, M., Hartley, R., and Fua, P. Convex Optimization for Deformable Surface 3D Tracking. In *International Conference on Computer Vision (ICCV)* (2007).
43. Salzmann, M., Moreno-Noguer, F., Lepetit, V., and Fua, P. Closed-form solution to non-rigid 3d surface registration. In *European Conference on Computer Vision (ECCV)* (2008), pp. 581–594.
44. Salzmann, M., Urtasun, R., and Fua, P. Local Deformation Models for Monocular 3D Shape Recovery. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).
45. Sattler, T., Sweeney, C., and Pollefeys, M. On sampling focal length values to solve the absolute pose problem. In *European Conference on Computer Vision (ECCV)* (Cham, 2014), pp. 828–843.
46. Schaefer, S., McPhail, T., and Warren, J. Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3 (July 2006), 533–540.
47. Sorkine, O., and Alexa, M. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 109–116.
48. Sturm, J. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software 11–12* (1999), 625–653. Version 1.05 available from `http://fewcal.kub.nl/sturm`.
49. Sturm, P. F., and Maybank, S. J. On plane-based camera calibration: A general algorithm, singularities, applications. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (1999), vol. 1, pp. 432–437 Vol. 1.
50. Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. Elastically deformable models. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 205–214.
51. Tomasi, C., and Kanade, T. Detection and tracking of point features, 1991.
52. Tran, Q.-H., Chin, T.-J., Carneiro, G., Brown, M. S., and Suter, D. In Defence of RANSAC for Outlier Rejection in Deformable Registration. In *European Conference on Computer Vision (ECCV)* (2012).
53. Varol, A., Salzmann, M., Fua, P., and Urtasun, R. A constrained latent variable model. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).
54. Vicente, S., and Agapito, L. Balloon Shapes: Reconstructing and Deforming Objects with Volume from Images. In *International Conference on 3D Vision* (2013).
55. Vávra, P., Roman, J., Zonča, P., Ihnát, P., Němec, M., Jayant, K., Habib, N., and El-Gendi, A. Recent development of augmented reality in surgery: A review. *Journal of Healthcare Engineering 2017* (08 2017), 1–9.
56. Wang, X., Fouhey, D. F., and Gupta, A. Designing deep networks for surface normal estimation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 539–547.
57. Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. Lift: Learned invariant feature transform. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 467–483.
58. Yu, R., Russell, C., Campbell, N. D. F., and Agapito, L. Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video. In *International Conference on Computer Vision (ICCV)* (2015), pp. 918–926.
59. Zhang, Z. A flexible new technique for camera calibration. ieee *Transactions on Pattern Analysis and Machine Intelligence 22* (2000), 1330–1334.

Appendix

1 Overview

This appendix is organized into 7 sections. §2 describes the implementation of the isometric cost defined in §3.1.2 with a triangulated mesh. §3 gives the default values of the termination condition thresholds used in Algorithm 1 and in all experiments. §4 gives the implementation details of the MDH and PnP SfT methods. §5 provides additional numerical details of each dataset. §6 provides the details for the initialization sensitivity experiment described in §4.8. §7 provides additional computational cost analysis.

2 Discrete quasi-isometric cost implementation

2.1 Triangle geometry and embedding functions

A constant strain triangle (CST) is a triangular element whose stress and strain fields are constant in the triangle's domain. Each triangle is associated with a flat 2D triangular domain $\Omega_t \subset \mathbb{R}^2$ where $t \in [1, T]$ is the triangle index. We define Ω_t by three 2D vertices denoted as $u_t^1 \in \mathbb{R}^2$, $u_t^2 \in \mathbb{R}^2$ and $u_t^3 \in \mathbb{R}^2$. We define as $\boldsymbol{y}_t^i \in \mathbb{R}^3$ and $\boldsymbol{x}_t^i \in \mathbb{R}^3$ the position of the i^{th} vertex of triangle t in object and camera coordinates respectively. We define as $\zeta_t : \Omega_t \rightarrow \mathbb{R}^3$ the embedding of triangle t into object coordinates. We define as $\phi_t : \Omega_t \rightarrow \mathbb{R}^3$ the embedding of triangle t into camera coordinates. Recall that ζ_t is known from the template and ϕ_t to be estimated by fSfT.

2.2 Cost

The isometric cost c_{iso} is constructed by the following discrete approximation of E_{strain}:

$$c_{\text{iso}} = \sum_{t=1}^{T} a_t \left\| G\left(J(\zeta_t)\right) - G\left(J(\phi_t)\right) \right\|_F^2 \approx E_{strain} \tag{16}$$

where $G(\boldsymbol{X}) \stackrel{\text{def}}{=} \boldsymbol{X}^\top \boldsymbol{X}$ is the Gramian operator, a_t is the known surface area of the t^{th} triangle and $J(f)$ denotes the Jacobian of a function f. Taken strictly, Equation (16) equates the first fundamental form of the two triangle embeddings.

We position the vertices $\boldsymbol{u}_t^{i\in[1,3]}$ such that the triangle in object coordinates is isometric with Ω_t as follows:

$$\boldsymbol{u}_t^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{R}_t \, \boldsymbol{y}_t^i \tag{17}$$

where $\boldsymbol{R}_t$ is any 3D rotation that aligns the triangle's normal in object coordinates to the z axis. As a consequence, $G(J(\zeta_t)) = \boldsymbol{I}_2$. Returning to Equation (16), we are left with expressing $J(\phi_t)$. Because ϕ_t is linear within Ω_t from the definition of a CST, as a local affine embedding $J(\phi_t) \in \mathbb{R}^{3\times 2}$ is constant and it is found from the following linear system of 6 equations:

$$J\left(\phi_t\right)\left(\boldsymbol{u}_t^i - \frac{1}{3}\left(\boldsymbol{u}_t^1 + \boldsymbol{u}_t^2 + \boldsymbol{u}_t^3\right)\right) = \left(\boldsymbol{x}_t^i - \frac{1}{3}\left(\boldsymbol{x}_t^1 + \boldsymbol{x}_t^2 + \boldsymbol{x}_t^3\right)\right), \quad \forall i \in [1,3] \tag{18}$$

The solution to $J(\phi_t)$ in Equation (18) is a linear expression in the unknown vertex positions $\boldsymbol{x}_t^1$, $\boldsymbol{x}_t^2$ and $\boldsymbol{x}_t^3$. We recall that from the definition of $\boldsymbol{\theta}$ in Equation (2), $\boldsymbol{x}_t^1$, $\boldsymbol{x}_t^2$ and $\boldsymbol{x}_t^3$ are contained within $\boldsymbol{\theta}$ ($\boldsymbol{\theta}$ holds the positions of all vertices in camera coordinates). As a consequence, $J(\phi_t)$ is linear in $\boldsymbol{\theta}$, and therefore Equation (16) is quartic in $\boldsymbol{\theta}$.

3 Optimization termination conditions

	τ_{step}	τ_Δ	τ_c	$\tau_{\mathcal{H}}$
T_1	10	1e − 5	1e − 5	20
T_2	20	1e − 5	1e − 5	20
T_3	100	1e − 5	1e − 5	20

Table 1: Default termination values used in Algorithm 1

4 SfT implementation details

4.1 MDH

MDH is considered one of the best closed-form SfT methods. Given an initial opening angle ψ_s with focal length f_s, and the set of N point correspondences $\mathcal{P}$ and $\mathcal{Q}$, we compute $\boldsymbol{\theta}_s$ in two stages as follows. In the first stage we reconstruct the depths $\mathcal{Z} = \{z_1, z_2, \ldots, z_N\}$ of $\mathcal{Q}$ by solving a convex relaxation of SfT following [42]. Specifically, we maximize the depth of each point such that the Euclidean distance e_{ij} between any two points $(i, j) \in [1, N]^2$ does not exceed their geodesic distance d_{ij}. The geodesic distances are known *a priori* from the template and they can be computed efficiently using *e.g.* Fast Marching [24]. The following SOCP problem is

then solved:

$$\begin{aligned} &\max \ \textstyle\sum_{i=1}^{N} z_i \quad \text{s.t.} \\ &\forall (i,j) \in \mathcal{N} \ \ \left\| \tfrac{z_i}{f_s} \operatorname{stk}\left(\mathcal{Q}(i), f_s\right) - \tfrac{z_j}{f_s} \operatorname{stk}\left(\mathcal{Q}(j), f_s\right) \right\|_2^2 \leq d_{ij} \end{aligned} \tag{19}$$

The set $\mathcal{N}$ defines pairs of point correspondences, constructed with a K-nearest neighbor (KNN) graph with a default of $\min(N, 15)$ neighbors. We solve Equation (19) quickly using the interior point method from Mosek [34]. When N is not large ($N \leq 500$), this typically takes between 100 and 500 ms on the benchmark computer. If $N > 500$ we reduce the problem size by randomly sub-sampling 500 correspondences without replacement using furthest point sampling, and we ignore the remaining points. This normally has little effect on reconstruction accuracy.

In the second stage we compute $\boldsymbol{\theta}_s$ from $\mathcal{Z}$ and f_s. We solve a regularized linear least squares system that finds a smooth 3D deformation of the template mesh that fits the reconstructed point correspondences in camera coordinates. This problem is as follows:

$$\begin{aligned} \boldsymbol{\theta}_s &= \arg\min_{\boldsymbol{\theta}} \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \left\| g(\mathcal{P}(i); \boldsymbol{\theta}) - \tfrac{z_i}{f} \operatorname{stk}(Q(i), f_s) \right\|_2^2 + \lambda c_{\text{reg}}(\boldsymbol{\theta}) \quad (a) \\ &= \arg\min_{\boldsymbol{\theta}} \| \boldsymbol{A}_{mdh} \boldsymbol{\theta} - \boldsymbol{b}_{mdh} \|_2^2 \qquad (b) \end{aligned} \tag{20}$$

where λ is a regularization weight. Equation (20-b) is equivalent to Equation (20-a) where we have rearranged the problem to a standard LLS format with known terms $\boldsymbol{A}_{mdh}$ and $\boldsymbol{b}_{mdh}$. The matrix $\boldsymbol{A}_{mdh}$ does not depend on f_s nor $\mathcal{Z}$. We exploit this by solving Equation (20) with a factorization of $\boldsymbol{A}_{mdh}$. Importantly, the factorization can be done once and be reused for any f_s or $\mathcal{Z}$. We weight c_{reg} using the normalization technique described in §3.1.3, and we use the same λ for all problem instances (we use a default of $\lambda = 100$ in all experiments).

The factorization can be solved very quickly when the number of mesh vertices V is small (a few hundred) using sparse Cholesky factorization. However for larger meshes it become unreasonably expensive. We deal with this by applying dimensionality reduction by eliminating high-frequency deformation components from the problem. We implement this with linear bases using a modal analysis of the design matrix of the regularization cost. This reduces the problem to a smaller dense linear system that is solved efficiently with Cholesky factorization (we use Eigen's LDLT implementation).

4.2 PnP

PnP estimates the rigid pose of the template in camera coordinates from a focal length sample f_s, and the point correspondences $\mathcal{P}$ and $\mathcal{Q}$. Despite not estimating deformation, we find this is a surprisingly effective and fast initialization method for fSfT. When $\mathcal{P}$ is co-planar we use IPPE [10], otherwise we use OpenCV's

SolvePnP method that implements the direct linear transform (DLT) initialization and Levenberg-Marquardt refinement.

5 Dataset descriptions

Additional dataset descriptions are provided in Tables 2, 3 and 4.

	Cap	Handbag	Pillow-cover	Spider-man
Object material	Fabric	Fabric	Fabric	Paper
Template geometry	3D open	3D open	3D open	Flat open
Number of template vertices (V)	4854	1098	1368	2918
Number of template triangles (T)	9502	2063	2587	5000
Video (vid.) or image collection (col.)	col.	col.	col.	col.
Number of images (M)	15	7	9	79
Image resolution ($w \times h$)	2048×1536	1280×960	1280×960	1728×1152
Correspondences per image (N)	266	150	63	1176 ± 468
Focal length (px)	2039	1344.0	1344.0	$1348.4 \rightarrow 3937.9$
Focal length (% of w)	99.6	105.0	105.0	$78.0 \rightarrow 277.9$
Lens opening angle (°)	53.3	50.9	50.9	$65.3 \rightarrow 24.8$
Has ground truth 3D	Yes	Yes	Yes	Yes

Table 2: Cap, Pillow-cover, Handbag and Spider-man dataset statistics.

6 Additional initialization sensitivity experiments

6.1 Initialization policies

We test 8 initialization policies in this experiment. The first 3 policies are the same as defined previously and we introduce 5 new policies as follows:

- Policy 1: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 2: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 3: $\Psi_{init} = \{20, 30, 40, 50, 60, 70, 80\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$

	Floral paper	Fortune teller	Hulk	Bending cardboard
Object material	Paper	Paper	Foam	Cardboard
Template geometry	3D open	3D open	Flat open	Flat open
Number of template vertices (V)	1248	936	122	609
Number of template triangles (T)	2342	1747	200	1120
Video (vid.) or image collection (col.)	col.	col.	col.	vid.
Number of images (M)	13	6	20	18 (87)
Image resolution ($w \times h$)	1280×960	1280×960	4928×3264	720×576
Correspondences per image (N)	18	20	20	52
Focal length (px)	1344.0	1344.0 → 3937.9	3784.9	879.6
Focal length (% of w)	105.0	105.0 → 277.9	76.8	122.2
Lens opening angle (°)	50.9	50.9	66.1	44.5
Has ground truth 3D	Yes	Yes	Yes	No

Table 3: Floral paper, Fortune teller, Hulk and Bending cardboard dataset statistics.

	Bedsheet	Kinect t-shirt	Kinect paper	Van Gogh paper
Object material	Fabric	Fabric	Paper	Paper
Template geometry	Flat open	Flat open	Flat open	Flat open
Number of template vertices (V)	1271	1089	1089	1189
Number of template triangles (T)	2400	2048	2048	2240
Video (vid.) or image collection (col.)	vid.	vid.	vid.	vid.
Number of images (N)	14 (68)	63 (313)	33 (100)	24 (71)
Image resolution ($w \times h$)	720×576	640×480	640×480	720×576
Correspondences per image N	1393	367	1228	4665
Focal length (px)	879.6	528.0	528.0	879.6
Focal length (% of w)	122.2	82.5	82.5	122.2
Lens opening angle (°)	44.5	62.4	62.4	44.5
Has ground truth 3D	No	Yes	Yes	No

Table 4: Bedsheet, Kinect t-shirt, Kinect paper and Van Gogh paper dataset statistics.

- Policy 4: $\Psi_{init} = \{50\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 5: $\Psi_{init} = \{50\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$
- Policy 6: $\Psi_{init} = \{\psi^{An}\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 7: $\Psi_{init} = \{20, 50, 80\}$, $\mathcal{M} \in \{\text{MDH}\}$
- Policy 8: $\Psi_{init} = \{\psi^{GT}\}$, $\mathcal{M} \in \{\text{MDH}, \text{PnP}\}$

Policies 4 and 5 have one focal length sample whose opening angle is 50°. We compare them to evaluate the benefit of initializating with two SfT methods (MDH and PnP) compared with one (MDH). This is similarly done with policies 6 and 1, and policies 7 and 2.

Policies 6 and 1 have one focal length sample which is from the analytical method. Policies 7 and 2 have three focal length samples and policy 3 has 7 focal length samples. Policy 8 has one focal length sample, which is the ground truth with opening angle denoted by ψ^{GT}. Of course, we cannot use policy 8 in practice because it requires the ground truth. However, we use it to compare how well the other policies perform compared to the ideal of initializing with the ground truth focal length.

6.2 Dataset versions

We use six dataset versions in this experiment as follows:

- **v1**: No augmentation (original datasets)
- **v2**: Zoom augmentation and noise augmentation with $\sigma = 0.16w$
- **v3**: Zoom augmentation and noise augmentation with $\sigma = 0.32w$
- **v1+SF**: v1 with Solvable Filtering
- **v2+SF**: v2 with Solvable Filtering
- **v3+SF**: v3 with Solvable Filtering

We now describe zoom augmentation, noise augmentation and Solvable Filtering.

Zoom augmentation implementation

Zoom augmentation is implemented as follows. For each image in each dataset, we convert the point correspondences to retina coordinates then we projected them back to image coordinates using a simulated intrinsic matrix with a random focal length f_{rand}, with principal point at the image center and zero skew. We compute f_{rand} independently for each image, using an opening angle ψ_{rand} drawn with uniform probability in the range 10° to 90°, producing a wide range of focal lengths. We illustrate examples of images with simulated digital zoom for the Cap dataset in Figure 12.

Fig. 12: 8 representative images from the Cap dataset where zoom augmentation is applied.

Noise augmentation implementation

Noise augmentation is implemented to simulate increasingly adverse conditions, by adding noise to each point correspondence and by reducing the number of point correspondences. Reducing points is required because additional noise has a smaller influence on accuracy when there are many points. For each image in each dataset, we retain N' points sampled randomly and without replacement where N' is drawn uniformly in the range $[\min(N, 75),\ \min(N, 100)]$ where N is the original number of points in the image. Noise is added by randomly perturbing each of the retained image points by Gaussian I.I.D. noise of standard deviation σ (px). We test $\sigma = 0.16w$ and $\sigma = 0.32w$ where w is the image width. These are equivalent to a standard deviation of 1px and 2px respectively at 640×480 resolution. The latter can be considered strong noise.

Solvable Filtering (SF) implementation

There may exist problem instances that are not solvable by *any* fSfT method. This is a limitation because such instances dilute the effect of different initialization policies on the performance metrics. To deal with this, we also measure performance on a sub-set of problem instances for which the optimization-based method succeeds (we define success if the FLPE is below 15%.) To implement this, we run the optimization-based method using *all* initializations contained in policies 1-8, and we filter out the problem instance if its corresponding FLPE was above 15%. By only evaluating on the filtered set of problem instances, we could answer the question: How well does an initialization policy perform given that the problem is solvable using at least one of the initialization policies?

6.3 Results

Focal length results are shown in Figure 13, where Figure 13(a) shows FLPE-success@15 and Figure 13(b) shows FLPE-success@5 averaged across all datasets. We make the following observations:

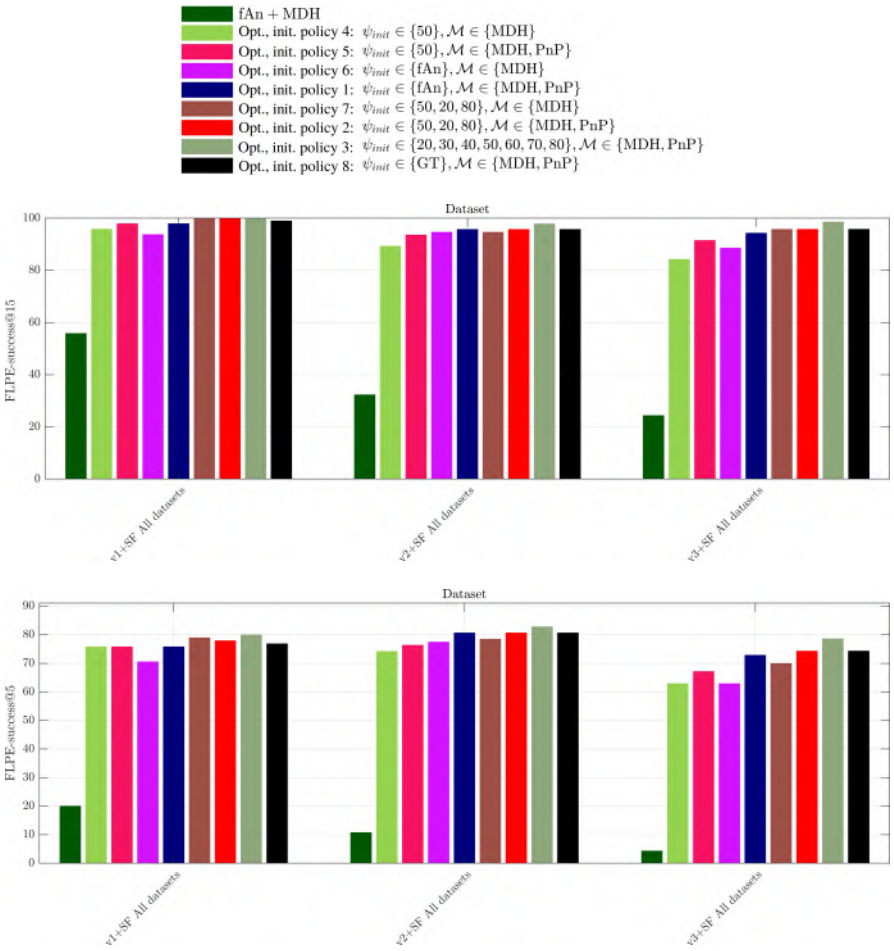


Fig. 13: Focal Length Percentage Error (FLPE) performance of the analytical method (denoted as 'fAn+MDH') and optimization-based method (denoted as 'Opt.') using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods $\mathcal{M}$ used to initialize deformation.

17. There is a strong trend where using more focal length samples in the initialization policy reduces focal length error. There is a slight improvement using policy 3 (with 7 samples) compared to policy 2 (with 3 samples). By contrast there is a strong benefit using policy 2 compared to policy 5 (with one sample). This illustrates diminishing returns where increasing the number of focal length samples has less of a benefit on solution accuracy.
18. The benefit of more focal length samples is less in v1+SF compared to v2+SF and v3+SF. This is because in v1+SF we do not apply zoom augmentation, so the focal lengths in v1+SF have opening angles in the range $24.8^\circ \leq \psi \leq 65.3^\circ$. In these cases the benefits of using more focal length samples is less pronounced compared to one sample at 50°.
19. There is a strong trend where using two SfT methods for initialization (MDH and PnP) improves performance compared to one SfT method (MDH). Recall that these methods operate very differently: MDH estimates deformation, and although it works well in general, there are cases when it does not estimate shape well thanks to the convex relaxation. By contrast, PnP does not estimate deformation, so the initialization it provides is the rigid pose that best fits the data. Adding the PnP solution appears to improve robustness in cases when the MDH solution cannot give a good initial estimate.
20. Initializing with the analytical method (policy 1) performs worse than initializing with a fixed opening angle of 50° (policy 5) for v1+SF. However, for v2+SF and v3+SF, we see a benefit where policies 6 and 1 outperform policies 4 and 5. Recall that v2+SF has zoom augmentation, and it has a much larger variation in opening angles compared to the original datasets without zoom augmentation (v1+SF). Therefore, when there is larger variation in opening angles, the analytical method is able to provide a better initialization compared to using a fixed opening angle of 50°. By contrast, in v1+SF, where the range of possible opening angles spans 40.5° with a midpoint at 45.0°, using a single opening angle of 50° performs better than using the opening angle from the analytical method.
21. Initializing with the ground truth focal length (policy 8) performs approximately the same as policy 2 (three focal length samples) for all dataset versions. This shows that accurate focal length initialization is not required by Algorithm 1.
22. Initializing with policy 8 performs worse in general than initializing with policy 3 (7 focal length samples). This can seem counter intuitive and we study the cause in more detail below. In short, the reason is because when we initialize with multiple focal length samples, we introduce shape diversity into the initialization set as a side effect. This diversity can help to locate the global optimum. We call this the *diverse initialization effect*.

The shape error results are shown in Figure 14, where Figure 14(a) shows SE-success@5 and Figure 14(b) shows SE-success@2. We observe all the same performance trends as we have observed for FLPE. The diverse initialization effect is also present.

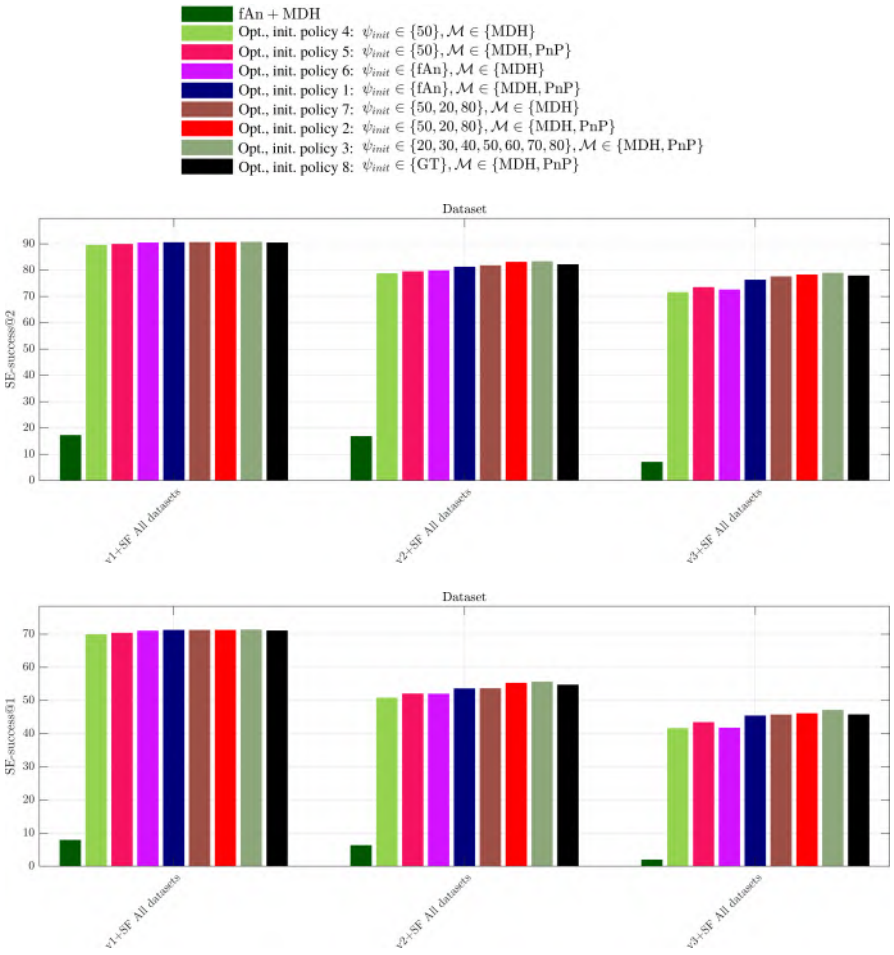


Fig. 14: Shape Error (SE) performance of the analytical method (denoted as 'fAn+MDH') and optimization-based method (denoted as 'Opt.') using different initialization policies. The initialization policies are defined in terms of the set ψ_{init} of initial focal lengths and the set of SfT methods $\mathcal{M}$ used to initialize deformation.

7 Computation cost analysis

We compare the computational cost of the different initialization policies by measuring the average number of optimization iterations (Gauss-Newton steps) required by Algorithm 1 using each policy. We use this instead of computation time because it is invariant to the implementation platform, and it is roughly proportional to computation time because the cost of executing each Gauss-Newton iteration is

approximately constant at each iteration. The results are shown in Figure 15 where we observe the following:

23. There is a clear increase in computational cost using policies with a larger initialization set.
24. There is practically no difference in the computational cost of initializing using one focal length sample (policies 4 and 5) and using the analytical method's focal length estimate (policies 6 and 1). This indicates that the number of iterations required for convergence is not highly sensitive to the accuracy of the initial focal length estimate.
25. The early termination criteria used in Algorithm 1 to avoid repeated search of solution space are proving effective. Without them, we would be seeing a doubling in the number of optimization iterations from policies using MDH to policies using both MDH and PnP. For example, the extra cost from policy 7 to policy 2 is between 22.2% and 32.7% depending on the dataset version. Without early termination the additional cost would be approximately 100%.
26. There is a slight increase in computational cost from v1 to v2 (and v1+SF to v2+SF) for all policies. This indicates that increasing noise also increases the number of iterations required for convergence.

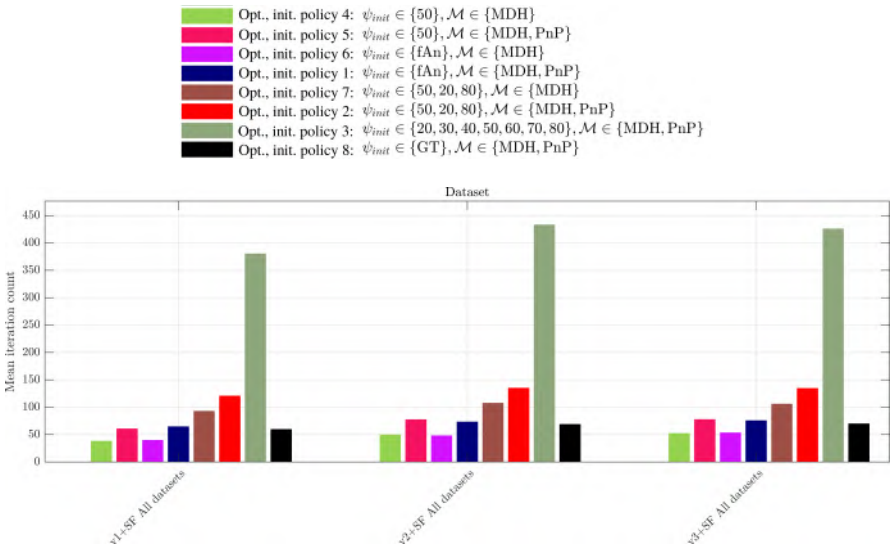


Fig. 15: Computational cost the optimization-based method with different initialization policies. This is expressed in the average number of Gauss-Newton iterations required for Algorithm 1 to converge.

z