Using Shading and a 3D Template to Reconstruct Complex Surface Deformations

Mathias Gallardo
Mathias.Gallardo@gmail.com
Toby Collins
Toby.Collins@gmail.com
Adrien Bartoli
Adrien.Bartoli@gmail.com

ISIT, UMR 6284 CNRS
Université d'Auvergne
Clermont-Ferrand, France

Abstract

The goal of Shape-from-Template (SfT) is to register and reconstruct the 3D shape of a deforming surface from a single image and a known deformable 3D template. Most SfT methods use only motion information and require well-textured surfaces which deform smoothly. Consequently they are unsuccessful for poorly-textured surfaces with complex deformations such as creases. We propose to combine shading and motion to handle these cases. There exist a few previous works which also exploit shading. However these do not provide an integrated solution, and they assume the surface and its deformations are globally smooth. Furthermore, most require a training phase and *a priori* photometric calibration for surface reflectance, camera response and/or light, which is often not possible. We propose an integrated solution without these shortcomings by jointly reconstructing the surface and performing photometric auto-calibration, considering an unknown light source which is constant and fixed in the camera coordinates. We evaluate with qualitative and quantitative results and show that it is possible to accurately register and reconstruct poorly-textured surfaces with complex deformations without any *a priori* photometric calibration.

1 Introduction, Background and Contributions

The problem of reconstructing a deformable 3D surface from 2D images has been addressed with three main approaches: Shape-from-Template (SfT) which uses a single image and a 3D deformable template, Shape-from-Shading (SfS) which uses a single image only, and Non-Rigid Structure-from-Motion (NRSfM) which uses multiple images. The most practical and robust approach is currently SfT, because the 3D template provides strong constraints on the problem [4, 18] and can be solved uniquely. In SfT, the template is a textured 3D model of the surface in a rest position, and the problem is solved by determining the 3D deformation that transforms the template into camera coordinates. The limitation of nearly all existing SfT methods is that they use only motion constraints, either by matching features such as SIFT between the template and the image [14, 18], or by densely matching at the pixel level [4, 12, 21]. These methods cannot therefore accurately reconstruct deformations at textureless regions (where motion information is unobtainable). On the other hand, SfS works exclusively with shading information through the photometric relationship between surface normals, light and pixel intensities [9, 22]. Shading is the most important visual cue for inferring high-frequency deformation at textureless regions [15]. SfS works by estimating the depths of each image pixel without the use of a 3D template, and can be used to reconstruct textureless surfaces. However SfS is often very difficult to use in practice for several reasons. Firstly it is a weakly-constrained problem and usually requires a complete *photometric calibration* of the scene *a priori*. A photometric calibration involves modeling surface reflectance (diffuse reflection, albedos, specular reflection, *etc.*), light, camera response and determining the model parameters. This is required in SfS because the problem is generally ill-posed if either surface reflectance, light or camera response parameters are unknown. Furthermore, SfS can only reconstruct up to scale, it cannot reconstruct self-occluded surfaces, and often suffers from discrete convex/concave ambiguities. Furthermore, in many applications such as Augmented Reality (AR) registration is required as well as reconstruction, which unlike SfT is not given by SfS.

There exist some related works on combining the advantages of SfS with SfT [11, 13, 20], however they have three main shortcomings. Firstly they do not formulate the problem as an integrated problem that combines all constraints. Secondly they require the surface and its deformations to be smooth, so the full power of shading constraints are not exploited. Thirdly, they require a photometric calibration *a priori* (either a complete calibration [11, 20] or a partial one [13]). [20] works by first segmenting the surface into textured and textureless regions, then the textured regions are independently reconstructed by an SfT method. Secondly, local textureless patches on the surface are independently reconstructed by applying a trained SfS regressor. In a final stage, the patches are stitched together to form the final surface. The regressor is trained from simulated images of the patch using a set of smooth training deformations, which requires *a priori* photometric calibration at both training and test times (and it has to be the same), so it cannot *e.g.* handle changes in illumination or camera exposure. [13] works by first transforming the template to camera coordinates using motion constraints (coming from point correspondences). The transform is not unique but up to a low-dimensional set of solutions. In a final stage, shading information is used to disambiguate the correct solution. Its main limitation is that the correct solution must be contained in the solution set, which is only possible if the surface and its deformations are extremely smooth. [11] requires complete photometric calibration and can also only reconstruct simple, smooth surfaces. Another drawback of [11, 20] is they assume the surface has a single albedo which is very restrictive.

We propose a novel, fully-integrated approach to combine shading constraints with SfT. This combines all the advantages of SfS and SfT. As with SfT, we use the 3D template to provide strong physical constraints on the surface's 3D shape and use shading constraints to reveal the complex deformations. The problem is solved by optimizing the template's shape using motion, shading and physical deformation constraints, whilst jointly performing photometric auto-calibration required to use shading. Our approach shows that it is possible to reconstruct a surface with complex, non-smooth deformations at all visible regions (both textured and textureless), and *without any a priori* photometric calibration. This has not been possible with previous SfS and SfT approaches. In §2, we present the general problem and specialize it to the case of a constant light source rigidly attached to the camera. In §3, we present our optimization framework. In §4, we validate our method with quantitative and qualitative results.

2 Problem Modeling

2.1 Template Definition

We define the template as three components: a *shape model*, a *deformation model* and a *reflectance model*. The shape model is a standard, texture-mapped thin shell 3D mesh model in a known reference pose consisting of a set of M 3D vertices $\mathbf{y} \triangleq \{\mathbf{y}_1, ..., \mathbf{y}_M\} \in \mathbb{R}^{3\times M}$ and F faces $\mathscr{F} \triangleq \{f_1, ..., f_F\}, f_k \in [1, M]^3$. The deformation model transforms each vertex to 3D camera coordinates. In order to capture complex deformations we do not use a low-dimensional deformation model (which is typically used in SfT [13, 14]), but instead model the position of each vertex $i \in \{1 \ldots M\}$ in camera coordinates by $\mathbf{x}_t^i \in \mathbb{R}^3$, where t denotes time. Therefore our task is to determine the set $\mathbf{x}_t \triangleq \left\{\mathbf{x}_t^1, ..., \mathbf{x}_t^M\right\} \in \mathbb{R}^{3\times M}$ at any time t.

The *reflectance model* defines the reflectance of each point on the template's surface. We use a Lambertian model, which gives a good approximation of many surfaces, and we handle model deviations due to *e.g.* specular reflections with a robust data term (see §2.3). Because the template is texture-mapped, we also have a texture-map image $\mathcal{T}(\mathbf{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ which models the intensity of any point $\mathbf{u}$ on the template's surface. In the SfT problem, $\mathcal{T}$ is typically generated from photographs of the template in its rest position [5], but it can also come from a CAD model [4]. The values in $\mathcal{T}$ do not in general correspond to surface albedo: If $\mathcal{T}$ is generated from photographs then it is formed from a complex process that mixes camera responses, albedos, illumination, and image blending (to merge the photographs). If $\mathcal{T}$ comes from a CAD model then it does not usually consider physical aspects that alter the albedo (*e.g.* surface roughness). However, to apply shading constraints we need the surface albedos. We define an *albedo texture-map* $\mathcal{A}(\mathbf{u}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ as a texture-map image that gives the surface albedos. Thus our task is to transform $\mathcal{T}$ into $\mathcal{A}$. To simplify the problem we assume that $\mathcal{A}$ is piecewise constant, which is valid for many surfaces (particularly man-made ones). $\mathcal{A}$ is therefore given by $\mathcal{A}(\mathbf{u}) : \mathbb{R}^2 \rightarrow \{\alpha_1, ..., \alpha_K\}$ where α_k denotes the k^{th} unknown albedo value, where K is also unknown. We transform a point $\mathbf{u} \in \mathcal{A}$ on the photometric texture-map to camera coordinates according to $\mathbf{x}_t$ with a barycentric interpolation, which is a linear interpolation of the positions of the three vertices surrounding $\mathbf{u}$. We use the function $f(\mathbf{u}; \mathbf{x}_t) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ to represent the transform of $\mathbf{u}$ to camera coordinates according to $\mathbf{x}_t$ (which is linear in $\mathbf{x}_t$), and $n(\mathbf{u}; \mathbf{x}_t) : \mathbb{R}^2 \rightarrow \mathbb{S}\mathbb{S}_{31}$ to represent its unit surface normal.

2.2 Illumination and Camera Models

We assume that the scene is illuminated by an unknown illumination which is constant over time and fixed in the camera coordinates. This is the setup both for a camera/light rig such as an endoscope, and a non-rig where the illumination and camera are not physically connected but do not move relative to each other whilst the images are being acquired. We use $\mathbf{l}$ to denote the unknown illumination coefficients. In this work we use first and second-order spherical harmonic models, which are very common models in SfS, with 4 and 9 parameters respectively. We use the function $r(\mathbf{n}, \mathbf{l}) : \mathbb{S}\mathbb{S}_{31} \rightarrow \mathbb{R}^+$ to denote the surface irradiance (the amount of light received by the surface per unit area) for a normal vector $\mathbf{n}$ according to $\mathbf{l}$. Note that for spherical harmonics r is linear in $\mathbf{l}$. The camera response function $I \triangleq g_t(R) : \mathbb{R} \rightarrow \mathbb{R}$ transforms the image irradiance R to pixel intensity I. We assume g_t is unknown and time-varying, which allows us to handle changes due to camera shutter speed and/or exposure. Similar to most works in SfS, we assume g_t is linear, which gives $I = \beta_t R$ with $\beta_t \in \mathbb{R}^+$. We assume the camera is perspective and intrinsically calibrated for all images (which is the typical assumption in SfT). All pixel positions are therefore given in normalized pixel coordinates. We use $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ to denote the projection from camera coordinates to normalized pixel coordinates.

2.3 The Integrated Cost Function

We first write the integrated cost function and then describe each term in detail. The cost function consists of image data terms (motion and shading terms) and physical deformation prior terms. For surfaces which have disc topology such as sections of cloth or sheets of paper, we also introduce a boundary data term similar to [19]. This encourages the surface's boundary to lie close to image edges and is useful for poorly textured surfaces. For a time t we use $\mathbf{x}_t \in \mathbb{R}^{3\times M}$ to denote the template's unknown deformed vertices. The cost function C_t for a single intensity image $\mathcal{I}_t$ is denoted by:

$$\begin{split} C_t(\mathbf{x}_t,\mathbf{l},\alpha_1,\ldots,\alpha_K,\beta_t;\mathcal{I}_t) \triangleq C_{shade}(\mathbf{x}_t,\mathbf{l},\alpha_1,\ldots,\alpha_K,\beta_t;\mathcal{I}_t) + \\ \lambda_{motion}C_{motion}(\mathbf{x}_t;\mathcal{I}_t) + \lambda_{bound}C_{bound}(\mathbf{x}_t;\mathcal{I}_t) + \lambda_{iso}C_{iso}(\mathbf{x}_t) + \lambda_{smooth}C_{smooth}(\mathbf{x}_t). \end{split} \tag{1}$$

The terms C_{shade}, C_{motion} and C_{bound} are shading, motion and boundary data terms respectively. The terms C_{smooth} and C_{iso} are physical deformation prior terms, which encourage the deformation to be smooth (C_{smooth}) and to not significantly stretch or shrink as it deforms (typically referred as an isometric prior in SfT, given by C_{iso}). The terms λ_{motion}, λ_{bound}, λ_{iso} and λ_{smooth} are positive weights that we select by hand and keep constant for all experiments. We do not assume dependency between the camera responses and surface deformations across different images, which allows us to handle both unorganised image sets and images from video streams. Given a set of N input images we define the complete cost function C as the sum of costs from each image: $C \triangleq \sum_{t=1}^{N} C_t$. In this work we consider batch sets, where the N images are provided as inputs at once, rather than incremental sets, where the images come sequentially from a capture device.

The shading term. The shading term robustly encodes the Lambertian relationship between albedo, surface irradiance, pixel intensity and camera response. We evaluate this at each pixel in the photometric texture-map $\mathcal{A}$, which gives:

$$C_{shade}(\mathbf{x}_t,\mathbf{l},\alpha_1,\ldots,\alpha_K,\beta_t;\mathcal{I}_t) \triangleq \sum_{\mathbf{u}\in\mathcal{A}} \rho\left(\beta_t\,\mathcal{A}(\mathbf{u})\,r\left(n(\mathbf{u};\mathbf{x}_t);\mathbf{l}\right) - \mathcal{I}_t\left(\pi\circ f(\mathbf{u};\mathbf{x}_t)\right)\right). \tag{2}$$

The function $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is an *M-estimator* which is used to enforce similarity between the modeled and measured pixel intensities, while also allowing for some points to violate the model (caused by specular reflection and other unmodeled factors). When the residual of such points is not too high, we find that an M-estimator is very effective for handle them. We also use M-estimators in some of the other cost function terms, and defer the exact choice to the implementation section.

The motion term. We assume that a set of 2D points in the texture-map image $\mathcal{S}_c = \{\mathbf{u}_1,\ldots,\mathbf{u}_S\}$ are putatively matched to each input image. Details for how this is done for our experimental datasets are given in §4. We denote the matching position of the j^{th} texture-map point in image t by $\mathbf{q}_t^j \in \mathbb{R}^2$. In general not all points will have matches due to self and/or external occlusions or feature detection failures. We represent this with the indicator matrix $\mathbf{M} \in \{0,1\}^{S \times N}$ where $\mathbf{M}(j,t) = 1$ means that the j^{th} point has a match in image t and $\mathbf{M}(j,t) = 0$ otherwise. We also assume there may be a small fraction of mis-matches due to ambiguous texture regions or other factors, which we handle by an M-estimator. The motion term robustly encourages the texture-map points to project to their matches and is given by:

$$C_{motion}(\mathbf{x}_t; \mathcal{S}_c, \mathbf{M}) \triangleq \sum_{\mathbf{u}_j \in \mathcal{S}_c} \mathbf{M}(j,t) \rho(\pi \circ f(\mathbf{u}_j; \mathbf{x}_t) - \mathbf{q}_j). \tag{3}$$

The boundary term. This constraint is based on the boundary term in [8] and works for surfaces with disc topology. It is used to encourage the surface's boundary to project closely to image edges. We discretize the boundary to obtain a set $\mathscr{B} \triangleq \{\mathbf{u}_{k \in [1,Q]}\}$ of Q boundary points defined on the texture-map image. For each input image we define an boundariness map $\mathcal{B}_t \triangleq \exp(-|\nabla \mathcal{I}_t|/\sigma)$ where $\nabla \mathcal{I}$ is the gradient of $\mathcal{I}$ and σ is the bandwidth. The boundary term is defined as follows:

$$C_{bound}(\mathbf{x}_t; \mathcal{B}_t) \triangleq \sum_{\mathbf{u}_j \in \mathscr{B}} \rho\left(\mathcal{B}_t(\pi \circ f(\mathbf{u}_j; \mathbf{x}_t))\right), \tag{4}$$

where an M-estimator is used to handle the fact that sometimes there may be little contrast difference between the surface and background structures.

Deformation priors. For the isometric cost (C_{iso}), we use a standard definition which penalises deviation of the template mesh's edges between the rest and deformed positions [10]. This is an appropriate physical constraint for many types of surfaces including plastics, cloths and paper, and strongly constrains deformation. We circumvent noise in the shading term using a robust smoothing regularizer C_{smooth} based on [8]. This uses the thin-plate energy, and encourages smoothness while also permitting high bending at creased regions:

$$C_{smooth}(\mathbf{x}_t) \triangleq \sum_{i \in [1,M]} \rho\left(\frac{\partial^2}{\partial \mathbf{u}^2} f(\mathbf{u}_i; \mathbf{x}_t)\right), \tag{5}$$

where $\mathbf{u}_i$ denotes the 2D position of the i^{th} vertex in the texture-map image. Note that most previous regularizers used in SfT cannot achieve this because they penalize surface bending with an ℓ_2 norm, which cannot handle sharp creases.

3 Solution

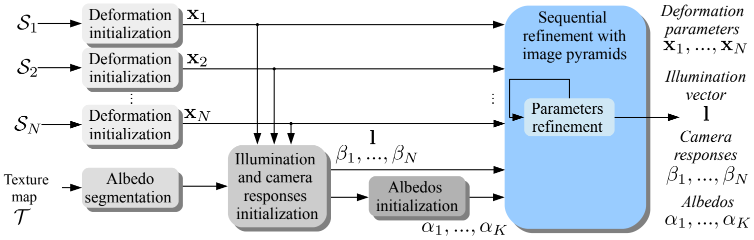
The integrated cost function C is large scale (depending on the mesh density and number of views there may be hundreds of thousands of unknowns), and it is highly non-linear which makes it challenging to solve. We present a solution using a fast cascaded initialization strategy followed by iterative gradient-based numerical optimization. A schematic of the whole process is illustrated in figure 1. Note that the problem always has a global photometric scale ambiguity between albedos, camera response and illumination strength because of their trilinear product in C_{shade}. This can be fixed by arbitrarily setting $\beta_1 = \alpha_1 = 1$.

3.1 Initialization

We propose a cascaded initialization strategy shown in Figure 1. This works by successively introducing the motion then boundary terms to obtain an initial estimate for the deformation parameters. Next the illumination and camera response parameters are estimated through a robust model-sampling based approach. This leverages the fact that at smooth surface regions, deformation can usually be estimated well at point correspondences without needing any boundary or shading terms. Given deformations at the correspondences, we initialize the photometric parameters by inverting the shading equation using pixel intensities and the estimated surface normals *around each correspondence*. At the final stage the albedo texture-map is initialized by first segmenting the intensity texture-map through an intrinsic image decomposition, and then estimating the albedo values for each segment by inverting the shading equation using the deformation, illumination and camera response estimates.

Initializing deformation. We use an existing global method to estimate $\{\mathbf{x}_t\}$ from point correspondences [3]. Because global methods such as [3] relax the isometric constraint (in [3] it is done using non-holonomic partial differential equations), the solutions can usually be improved by iterative non-linear optimization. We perform this by iteratively optimizing our cost function with the non-motion data term weights (λ_{shade} and λ_{bound}) set to zero, using Gauss-Newton iterations and backtracking line-search. Note that because the shading term is not used $\mathbf{x}_t$ can be optimized independently for each image and in parallel. We then improve the initial solution at regions far from any point correspondences by introducing the boundary term into the cost function. This is again done with Gauss-Newton iterations and for each image independently. To improve convergence we construct $\mathcal{B}_t$ using an image pyramid (we found that three octaves provide good convergence), and sequentially optimize with each pyramid level until convergence.

Initializing illumination and camera responses. Using the initial deformation estimates, we first estimate the surface normal for each of the texture-map points in $\mathcal{S}_c$. These normals are given in 3D camera coordinates and denoted by $\mathbf{n}_t^j$, with t being the image index and j being the point index. For each point in $\mathcal{S}_c$ we also model the average albedo within a small square window (in our experiments we use 11×11 windows), which we denote by $a_j \in \mathbb{R}$. Recall that because we do not yet know the albedo texture-map, we do not yet know a_j. We use $I_t^j \in \mathbb{R}$ to denote the average pixel intensity within the local window in the t^{th} image. Our goal is to estimate the illumination vector $\mathbf{l}$, the camera response terms $\{\beta_t\}$ and the albedo values $\{a_j\}$ by inverting the Lambertian shading equation $\beta_t \, a_j \, r(\mathbf{n}_t^j; \mathbf{l}) = I_t^j$. This is a hard non-convex inverse problem. It can be greatly simplified if $\{\beta_t\}$ is known, because with spherical harmonic models r is linear in $\mathbf{l}$, so it becomes linear by dividing through by a_j. In some cases $\{\beta_t\}$ can be estimated from EXIF tags. In other cases, if the image background does not change between images we can approximate $\{\beta_t\}$ by taking the ratio of pixel intensities in the background between different images. When neither is possible, we can make a different assumption: that the camera response terms are roughly similar in a few of the images, but we do not know which ones *a priori*. We then can tackle the problem with random sample consensus. To do this we require a low dimensional illumination model, so we use first-order spherical harmonics (with 4 dimensions). We will first describe how the unknown parameters are computed from a minimal sample of a single texture-map point and 4 images. We will then describe how this fits into a RANSAC framework. Given a single texture-map point j and 4 images where β_t is assumed constant, we solve $\mathbf{l}$ up to scale with an exact linear system (recall that the scale is never recoverable but it is not actually important to us). Given $\mathbf{l}$, each correspondence is then used to estimate β_t by taking intensity ratios I_t^j / I_1^j. Recall that $\beta_1 = 1$ (to fix the photometric scale ambiguity), so after rearranging we



$\mathcal{S}_1$
Deformation initialization
$\mathbf{x}_1$
$\mathcal{S}_2$
Deformation initialization
$\mathbf{x}_2$
$\mathcal{S}_N$
Deformation initialization
$\mathbf{x}_N$
Texture map $\mathcal{T}$
Albedo segmentation
Illumination and camera responses initialization
$\mathbf{l}$
$\beta_1, ..., \beta_N$
Albedos initialization
$\alpha_1, ..., \alpha_K$
Sequential refinement with image pyramids
Parameters refinement
Deformation parameters $\mathbf{x}_1, ..., \mathbf{x}_N$
Illumination vector $\mathbf{l}$
Camera responses $\beta_1, ..., \beta_N$
Albedos $\alpha_1, ..., \alpha_K$

Figure 1: Schematic of our approach to optimize the cost function C.

have $\beta_t = I_t^j r(\mathbf{n}_1^j;\mathbf{l})/I_1^j r(\mathbf{n}_t^j;\mathbf{l})$. Because each texture-map point produces a value for β_t, we compute β_t robustly by taking the median across all texture-map points. We then compute a_j. Each image provides an estimate with $a_j = I_t^j/(\beta_t\, r(\mathbf{n}_t^j;\mathbf{l}))$, so a robust estimate is computed by taking the median across the images.

The second component for RANSAC is to validate the parameters through consensus. We do this by computing the number of point matches where the shading equation is satisfied up to noise: $|\beta_t\, a_j\, r(\mathbf{n}_t^j;\mathbf{l}) - I_t^j| \leq \tau$. This requires an acceptance tolerance τ and in all experiments we use $\tau = 0.04$. The third component is the random selection, which was done by first randomly selecting the texture-map point from $\mathcal{S}_c$ (with uniform probability), then selecting four images where it had a match (with uniform probability). We terminated RANSAC if either 50% consensus was reached or an iteration limit $L = 20,000$ iterations had passed. On a standard Intel i7 desktop workstation $L = 20,000$ takes a few seconds to reach wit sub-optimal Matlab code. The choice of L depends on the likelihood of drawing 4 images with approximately the same camera response, which is difficult to know *a priori*.

Initializing albedos. Recall that we assume the surface albedos are piecewise constant. We first perform an intensity-based segmentation of the template's texture-map using an intrinsic image decomposition method [?] to obtain the reflectance image. We cluster the reflectance image into piecewise constant albedos by using the Mean Shift algorithm [?] and threshold the pixels number of each region to remove the textured regions. This is designed to be an oversegmentation, and within each segment we assume the albedo is constant. Thus if there are K segments, then the albedo set $\{\alpha_1, \ldots \alpha_K\}$ has size K. For each segment, we estimate its corresponding albedo by optimizing C_{shade}, using only the texture-map $\mathcal{T}$ and the photometrically-calibrated texture-map $\mathcal{A}$ pixels contained within the segment. To improve convergence, we also use an image pyramid (three octaves such as for the boundary constraint) of blurred irradiance images. For convex M-estimators such as the (ℓ_1-ℓ_2) M-estimator, this is a convex problem that can be solved globally. Note that this initialization corresponds to transform the texture-map $\mathcal{T}$ into the photometrically-calibrated texture-map $\mathcal{A}$ and is based on Lambertian surface assumption. Other reflectance models will require to estimate extra-parameters or maybe to change the initialization process.

3.2 Refinement

Having initialized we refine C using Gauss-Newton iterations with line-search. For optimizing illumination we either keep to the first-order spherical harmonics model or switch to the second-order model (we present results for both cases). Note that selecting the best illumination model is non-trivial and we leave this to future work. Unlike the initialization process, the deformation parameters $\mathbf{x}_t$ are all linked in the optimization through the shading term. Thus the problem size during refinement grows with the number of views. Because we use a triangulated mesh parametrization, all constraints are sparse with respect to $\mathbf{x}_t$. Therefore each Gauss-Newton iteration requires solving a large sparse linear system. We have found that for dense meshes with vertices of order $\mathcal{O}(10^4)$ one can solve this directly with sparse Cholesky in reasonable time with up to ten views (typically done in under a few minutes in Matlab on a desktop PC). For larger meshes and more views, direct solvers become impractical and one must resort to iterative solvers such as conjugate gradient.

4 Experimental Results

Method comparison. We evaluated our method with three real-world datasets which mostly respect the Lambertian assumption. These exhibit complex non-smooth deformations without any *a priori* photometric calibration, so cannot be handled by SfS methods nor previous methods combining shading and SfT [11, 13, 20] (code for these methods is not publically available). We compared with four competitive SfT methods with publically available code [1, 3, 14, 17], denoted respectively by **ReD12**, **ReJ14**, **MDH09** and **LM16**. We evaluated our method in five cases to fairly assess the benefits of using shading. The first two are with a first-order spherical harmonic light model which is either calibrated *a priori* or uncalibrated, denoted by **DaK** and **DaU** respectively. The second two are with a second-order spherical harmonic light model that is either calibrated *a priori* or uncalibrated, denoted by **Sh9K** and **Sh9U**. The fifth is when shading is omitted by assigning $\lambda_{shade} = 0$, denoted by **NoS**.

Datasets. We computed very high quality ground truth surfaces with sub-millimeter accuracy using a structured-light 3D scanner [6]. This has considerably higher precision than is used in previous SfT method, which mostly use the Microsoft Kinect. Images were captured by a 1288 × 964 px Point Grey camera [16]. We computed the illumination calibration using multiple images of a MacBeth chart. We have tested with three datasets: *floral paper* (8 images and 20 manual correspondences), shown in Figure 3, rows $n°1$ and $n°2$, *paper fortune teller* (4 images and 24 manual correspondences), shown in Figure 3, rows $n°3$ and $n°4$, and *floral video* (8 images and 1000 correspondences), shown in Figure 3, row $n°5$. For *floral video*, 8 frames were extracted from a one minute video (uniformly sampled over time), and correspondences were generated with a dense point tracking covering the textured regions.

For each dataset, a template was constructed using a grid of 100×100 mesh vertices (we found that this number of points is sufficient to accurately reconstruct creases). This was texture-mapped using a photograph of the template in an undeformed rest state. For all methods we manually set their free parameters to achieve the best performance on all datasets (for our method these are the weight terms in C). In our method, we found that respectively the (ℓ_1-ℓ_2) M-estimator for motion, boundary and smoothing terms and Huber M-estimator for shading term give good reconstructions. We found that using shading in half of the texture-map pixels gives good reconstructions. The weights of the different constraint are normalized (λ_{motion} by the number of correspondences, λ_{bound} by the number of boundary points, λ_{shade} by the number of shading points, λ_{iso} by the number of edges and λ_{smooth} by the area of the surface) and set to $\lambda_{motion} = 4.1667$, $\lambda_{bound} = 0.0017$, $\lambda_{iso} = 0.6667$, $\lambda_{smooth} = 1.6667\ e^{-14}$ and the extra-parameter of Huber M-estimator for shading constraint

$k_{shade} = 1\ e^{-3}$. To measure accuracy we used four metrics: *(i)* the depth error over the entire surface (in %), *(ii)* the normal error over the entire surface (in degrees), *(iii)* the depth error at regions near to surface creases (to within approximately 10 mm of a crease), and *(iv)* the normal error at regions near to surface creases.

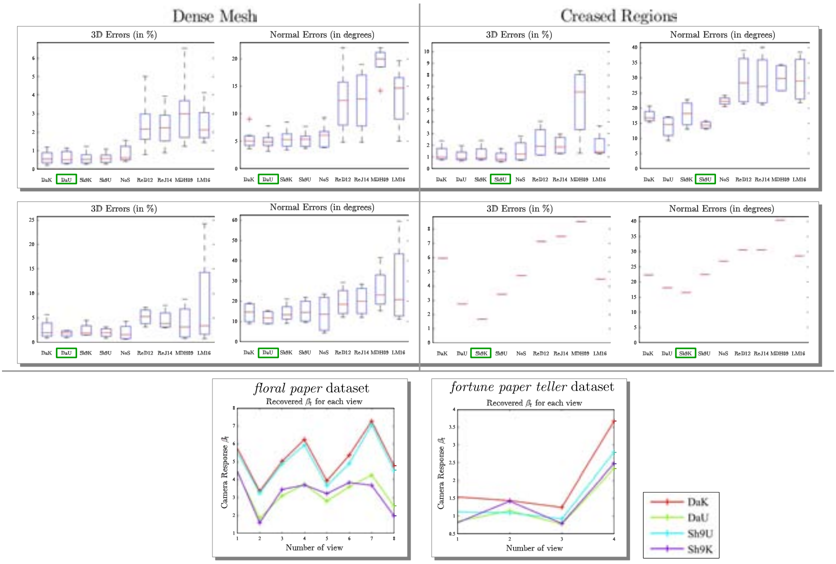


Figure 2: Numerical results for the three datasets. **First row**: *floral paper* dataset. **Second row**: *paper fortune teller* dataset. We indicate by a green rectangle the method which produces the lowest median value. **Third row**: recovering the camera response β_t for *floral paper* and *paper fortune teller* datasets. Best viewed in colour.

Results. For the three datasets, we present in figure 3 the 3D reconstructions produced by all the methods. **ReD12**, **ReJ14** and **LM16** produce smooth surfaces and do not reconstruct the creases because these methods interpolate the surface between correspondences. **MDH09** forms non-smooth deformations but not in the good regions. The reason is that the reconstructed creases are a by-product of the inextensibility constraint used in [17] which is a relaxation of the isometry constraint. We observe that **NoS** succeeds to create creases when creases are revealed by the surface boundaries. However, these reconstructed folds are not sharp enough and the creases which cannot be guessed from the boundaries are not reconstructed, as the *paper fortune teller* in figure 3 illustrates. Our method, **Sh9K** and **Sh9U**, produces significantly good results compared to the state-of-the-art methods: the creases are well registered and reconstructed. We note that the rendered solutions of **DaK** and **DaU** are similar to the ones of **Sh9K** and **Sh9U**. Figure 2 shows the shape error of the compared methods for one input image and for the *floral paper* and *paper fortune teller* datasets. Our methods, **DaK**, **DaU**, **Sh9K** and **Sh9U**, produce the best precision on 3D errors and normal errors. In particular, we note that the shading improves notably the normals error. The numerical results are coherent with the renderings in figure 3. We note that using second-order spherical harmonics, **Sh9K** and **Sh9U**, improves lightly the reconstructions since there are more variables which gives more degrees of freedom for the minimization of the cost function Eq.(1). In figure 2, we also show that we recover the camera responses β_t and the recovered values are similar in the four versions of our method. We show also a qualitative result with the *floral video*, where we can note the shading contribution to reveal the crease.

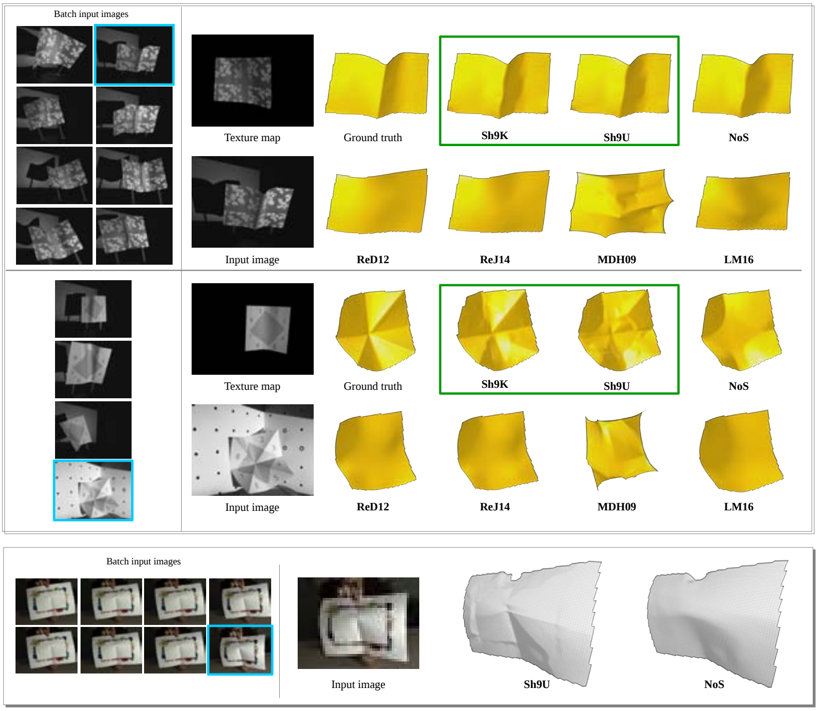


Figure 3: Renders of results. **Rows** $n^\circ 1$ **and** $n^\circ 2$: input image $n^\circ 5$ of the *floral plane* dataset. **Rows** $n^\circ 3$ **and** $n^\circ 4$: input image $n^\circ 4$ of the *paper fortune teller* dataset. **Rows** $n^\circ 5$: input image $n^\circ 8$ of the *floral video* dataset.

5 Conclusion

We have presented an integrated approach to reconstruct deformable surfaces from 2D images using a 3D template, shading and motion constraints. We have shown that with our approach it is possible to reconstruct complex non-smooth deformations without any *a priori* photometric calibration, which was not possible with previous methods in SfT or SfS. In future works, we will investigate the impact of different image transforms, such as steerable filters, on convergence. We also aim to extend the approach to analyse degenerate scenes and to handle shadows, non-Lambertian reflectance, self-occlusions, surfaces which can significantly stretch as they deform and more real world applications such as endoscopic images.

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

References

[1] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-Template. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, 2015.

[2] S. Bell, K. Bala, and N. Snavely. Intrinsic Images in the Wild. *ACM Trans. on Graphics (SIGGRAPH)*, 33(4), 2014.

[3] A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins. A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[4] T. Collins and A. Bartoli. Realtime Shape-from-Template: System and Applications. In *International Symposium on Mixed and Augmented Reality*, 2015.

[5] T. Collins, P. Mesejo, and A. Bartoli. An Analysis of Errors in Graph-Based Keypoint Matching and Proposed Solutions. In *European Conference on Computer Vision*, 2014.

[6] David 3D Scanner. http://www.david-3d.com/en/products/david4, 2014.

[7] K. Fukunaga and L. Hostetler. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE *Transactions on Information Theory*, 21(1):32–40, Jan 1975.

[8] M. Gallardo, T. Collins, and A. Bartoli. Can we Jointly Register and Reconstruct Creased Surfaces by Shape-from-Template Accurately? In *European Conference on Computer Vision*, 2016.

[9] B. K.P. Horn. Shape from Shading: A Method for Obtainig the Shape of a Smooth Shape of a Smooth Opaque Object from One View. Technical report, Cambridge, MA, USA, 1970.

[10] Slobodan Ilić, Mathieu Salzmann, and Pascal Fua. Implicit Meshes for Effective Silhouette Handling. *International Journal of Computer Vision*, 72(2):159–178, 2006.

[11] A. Malti and A. Bartoli. Combining Conformal Deformation and Cook-Torrance Shading for 3D Reconstruction in Laparoscopy. IEEE *Transactions on Biomedical Engineering*, 61(6):1684–1692, June 2014.

[12] A. Malti, A. Bartoli, and T. Collins. A Pixel-Based Approach to Template-Based Monocular 3D Reconstruction of Deformable Surfaces. In *Proceedings of the* IEEE *International Workshop on Dynamic Shape Capture and Analysis at ICCV*, pages 1650–1657, November 2011.

[13] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D Stretchable Surfaces from Single Images in Closed Form. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

[14] Dat Tien Ngo, Jonas Östlund, and Pascal Fua. Template-based Monocular 3D Shape Recovery using Laplacian Meshes. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 38(1):172–187, 2016.

[15] A. Pentland. Shape Information From Shading: A Theory About Human Perception. In *Computer Vision., Second International Conference on*, pages 404–413, Dec 1988.

[16] Point Grey. Flea2G 1.3 MP Color Firewire 1394b (Sony ICX445). https://www.ptgrey.com/.

[17] M. Salzmann and P. Fua. Reconstructing Sharply Folding Surfaces: A Convex Formulation. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

[18] M. Salzmann and P. Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

[19] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

[20] A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3D Reconstruction of Locally Textured Surfaces. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 34(6), 2012.

[21] Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. Direct, Dense, and Deformable: Template-Based Non-rigid 3D Reconstruction from RGB Video. In *International Conference on Computer Vision*, 2015.

[22] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from Shading: A Survey. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.