Monocular 3D Reconstruction of Locally Textured Surfaces

Aydin Varol, Appu Shaji, Mathieu Salzmann, and Pascal Fua, *Fellow*, *IEEE*

Abstract—Most recent approaches to monocular nonrigid 3D shape recovery rely on exploiting point correspondences and work best when the whole surface is well textured. The alternative is to rely on either contours or shading information, which has only been demonstrated in very restrictive settings. Here, we propose a novel approach to monocular deformable shape recovery that can operate under complex lighting and handle partially textured surfaces. At the heart of our algorithm are a learned mapping from intensity patterns to the shape of local surface patches and a principled approach to piecing together the resulting local shape estimates. We validate our approach quantitatively and qualitatively using both synthetic and real data.

Index Terms—Deformable surfaces, shape recovery, shape from shading.

1 Introduction

Many algorithms have been proposed to recover the 3D shape of a deformable surface from either single views or short video sequences. The most recent approaches rely on using point correspondences that are spread over the entire surface [13], [15], [31], [35], [41], [42], [47], [52], which requires the surface to be well textured. Others avoid this requirement by exploiting contours, but can only handle surfaces such as a piece of paper, where the boundaries are well defined [18], [24], [30], [51]. Some take advantage of shading information, but typically only to disambiguate the information provided by the interest points or the contours [49]. This is largely because most traditional shape-from-shading techniques can only operate under restrictive assumptions regarding lighting environment and surface albedo.

In this paper, we propose a novel approach to recovering the 3D shape of a deformable surface from a monocular input by taking advantage of shading information in more generic contexts. This includes surfaces that may be fully or partially textured and lit by arbitrarily many light sources. To this end, given a lighting model, we propose to learn the relationship between a shading pattern and the corresponding local surface shape. At runtime, we first use this knowledge to recover the shape of surface patches and then enforce spatial consistency between the patches to produce a global 3D shape.

More specifically, we represent surface patches as triangulated meshes whose deformations are parameterized as weighted sums of deformation modes. We use spherical harmonics to model the lighting environment, and calibrate this model using a light probe. This lets us shade and render realistically deforming surface patches that we use to create a database of pairs of intensity patterns and 3D local shapes. We exploit this data set to train Gaussian Process (GP) mappings from intensity patterns to deformation modes. Given an input image, we find featureless surface patches and use the GPs to predict their potential shapes, which usually yields several plausible interpretations per patch. We find the correct candidates by linking each individual patch with its neighbors in a Markov Random Field (MFR).

We exploit texture information to constrain the global 3D reconstruction and add robustness. To this end, we estimate the 3D shape of textured patches using a correspondence-based technique [35] and add these estimates into the Markov Random Field. In other words, instead of treating texture as noise as in many shape-from-shading approaches, we exploit it as an additional source of information.

In short, our contribution is an approach to shape-from-shading that can operate in a much broader context than earlier ones: We can handle indifferently weak or full perspective cameras, the surfaces can be partially or fully textured, we can handle any lighting environment that can be approximated by spherical harmonics, there is no need to presegment the surface, and we return an exact solution as opposed to one up to a scale factor. While some earlier methods address subsets of these problems, we are not aware of any that tackle them all.

We demonstrate the effectiveness of our approach on synthetic and real images, and show that it outperforms state-of-the-art texture-based shape recovery and shape-from-shading techniques.

2 Related Work

Recent advances in nonrigid surface reconstruction from monocular images have mostly focused on exploiting textural information. These techniques can be roughly classified into Template-based approaches and Structure-from-Motion methods.

- *A. Varol, A. Shajiis, and P. Fua are with the EPFL-IC-Computer Vision Laboratory, Station 14, CH-1015 Lausanne, Switzerland. E-mail: {aydin.varol, appu.shaji, pascal.fua}@epfl.ch.*
- *M. Salzmann is with NICTA, 7 London Circuit, Canberra ACT 2600, Australia. E-mail: mathieu.salzmann@nicta.com.au.*

Manuscript received 19 Nov. 2010; revised 8 July 2011; accepted 25 Aug. 2011; published online 7 Oct. 2011.
Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference. IEEECS Log Number TPAMI-2010-11-0880.
Digital Object Identifier no. 10.1109/TPAMI.2011.196.

Template-based methods start from a reference image in which the 3D surface shape is known. They then establish point correspondences between the reference image and an input image from which the unknown 3D shape is to be recovered. Given such correspondences, reconstruction amounts to solving an ill-conditioned linear system [36] and additional constraints must be imposed to obtain an acceptable solution. These may include inextensibility constraints as well as local or global smoothness constraints [13], [31], [35], [41], [52].

Structure-from-Motion methods depend on tracking points across image sequences. This approach was initially introduced in [10] to extend to the nonrigid case earlier structure-from-motion work [43]. Surface shapes are represented as linear combinations of basis shapes, which are estimated together with the weights assigned to them and the camera pose. This is again an ill-posed problem, which requires additional constraints. They include orthonormality constraints designed to ensure that the recovered camera motion truly is a rotation [3], [9], [40], [50], motion constraints [1], [28], [33], basis constraints [50], or alternate deformation models [16], [32], [44]. More recently, it has been proposed to split the global reconstruction into a series of local ones, which can then be patched together into a consistent interpretation. The local surface deformations can be modeled as isometric [42], planar [11], [47], or quadratic [15].

While these correspondence-based techniques are effective when the texture is sufficiently well spread across the surface, they perform less well when the texture is sparser or even absent. In the case of developable surfaces, this limitation can be circumvented by using information provided by boundaries, which is sufficient to infer the full 3D shape [18], [24], [30], [51]. Nevertheless, this approach does not extend to cases where the contours are not well defined. For those, in the absence of texture, the natural technique to use is shape-from-shading [20]. However, despite many generalizations of the original formulation to account for increasingly sophisticated shading effects, such as interreflections [17], [27], specularities [29], shadows [23], or non-Lambertian materials [2], most state-of-the-art solutions can only handle a subset of these effects and, therefore, only remain valid in tightly controlled environments. Shape-from-shading techniques have been made more robust by exploiting deformation models [38], [39]. However, this was only demonstrated for the single light source case. By contrast, our method can operate in more general environments, only provided that a light model expressed in terms of spherical harmonics can be estimated.

A more practical solution to exploiting shading is to use it in conjunction with texture. In [49], shading information was used to overcome the twofold ambiguity in normal direction that arises from template matching. In [26], the inextensibility constraints mentioned earlier were replaced with shading equations, which allowed the reconstruction of stretchable surfaces. However, these techniques still require the presence of texture over the whole surface. By contrast, our proposed framework can exploit very localized texture in conjunction with shading to reconstruct the entire surface.

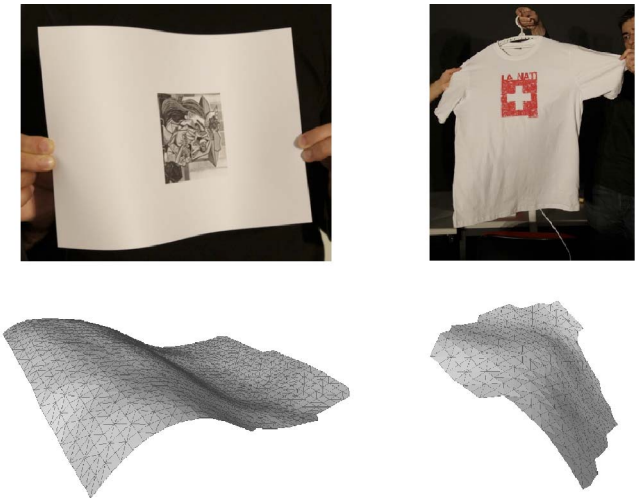


Fig. 1. 3D reconstruction of two poorly textured deformable surfaces from single images.

3 Method Overview

Our goal is to recover the 3D shape of deforming surfaces such as those shown in Fig. 1 from a single *input image*, given a *reference image* in which the shape is known, a calibrated camera, and a lighting model. We assume that the surface albedo is constant, except at textured regions, and measure it in the reference image. Our approach relies on several insights.

- The deformations of local surface patches are simpler to model than those of the whole surface.
- For patches that are featureless, one can learn a relationship between gray-level variations induced by changes in surface normals and 3D shape that holds even when the lighting is complex.
- For patches that fall on textured parts of the surface, one can use preexisting correspondence-based techniques [35].

This patch-based approach allows the use of different techniques for different patches depending on the exact nature of the underlying image. In practice, the local reconstruction problems may have several plausible solutions and obtaining a global surface requires a final step to enforce global geometric consistency across the reconstructed patches.

The algorithm corresponding to our approach is depicted by Fig. 2. Its two key steps are the estimation of local 3D surface shape from gray-level intensities across image patches followed by the enforcement of global geometric consistency. We outline them briefly below and discuss them in more details in the two following sections.

3.1 Estimating the Shape of Local Patches

While we can reconstruct the 3D shape of textured patches by establishing correspondences between the feature points they contain and those points in the reference image [35], this can obviously not be done for featureless ones. For those, we infer shape from shading-induced gray-level variations. Since there is no simple algebraic relationship between intensity patterns and 3D shape

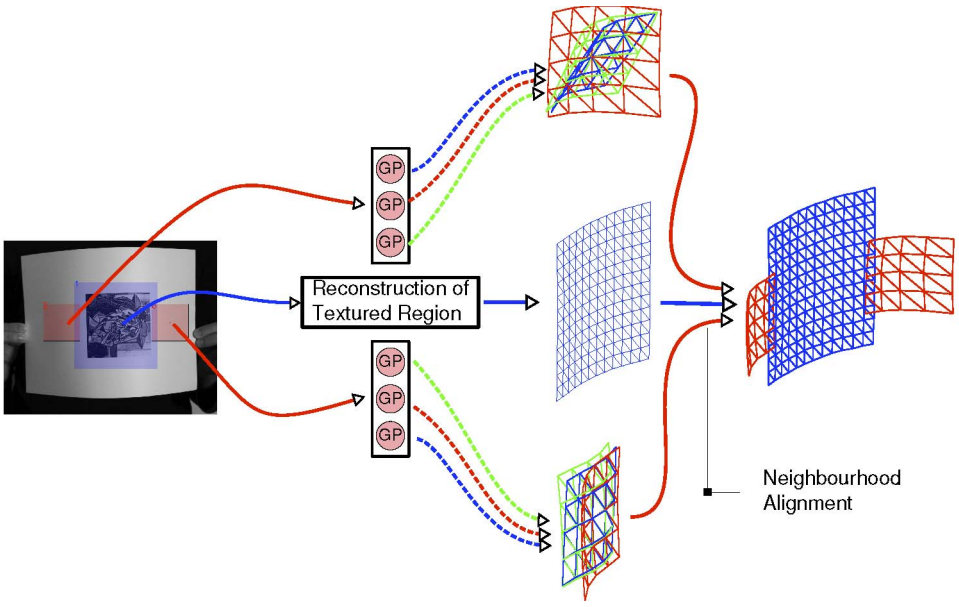


Fig. 2. Algorithmic flow. We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian Processes trained on synthetic data to predict plausible 3D shapes for featureless patches such as the red ones. Finally, neighborhood alignment of the patches is done using a Markov Random Field to choose, among all possible local interpretations, those that are globally consistent.

when the lighting is complex, we use a Machine Learning approach to establish one.

More specifically, we learn GP mappings from intensity variations to surface deformations using a training set created by rendering a set of synthetically deformed 3D patches shaded using the known lighting model. As we will see, this is a one-to-many mapping since a given intensity pattern can give rise to several interpretations.

3.2 Enforcing Overall Geometric Consistency

Because there can be several different interpretations for each patch, we must select the ones that result in a consistent global 3D shape. To this end, we link the patches into an MRF that accounts for dependencies between neighboring ones. Finding the maximum a posteriori state of the MRF then yields a consistent set of local interpretations.

Although not strictly necessary, textured patches, which can be reconstructed accurately in most cases, help to better constrain the process. In essence, they play the role of boundary conditions, which are always helpful when performing shape-from-shading type computations.

4 Estimating Local Shape

As outlined above, our method begins by reconstructing local surface patches from intensity profiles, which we do using a statistical learning approach. To this end, we calibrate the scene lighting, create a training database of deformed 3D patches and corresponding intensity profiles, and use GPs to learn the mapping between them.

4.1 Generating Training Data

Since shading cues are specific to a given lighting environment, we begin by representing it in terms of spherical harmonics coefficients that we recover using a spherical light probe. As scene irradiance is relatively insensitive to high frequencies in the lighting, for Lambertian objects lit by far lighting sources we can restrict ourselves to the first nine such coefficients [34]. In practice, this has proven sufficient to operate in an everyday environment, such as our office pictured in Fig. 3, which is lit by large area lights and extended light sources.

To populate our training database, we take advantage of the availability of a set of realistically deforming surface

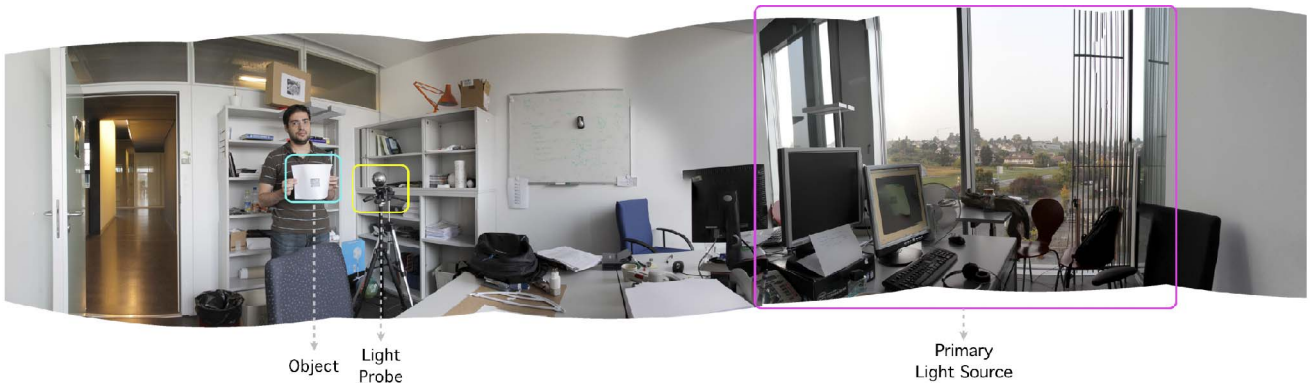


Fig. 3. Panoramic image of the environment in which we performed our experiments.

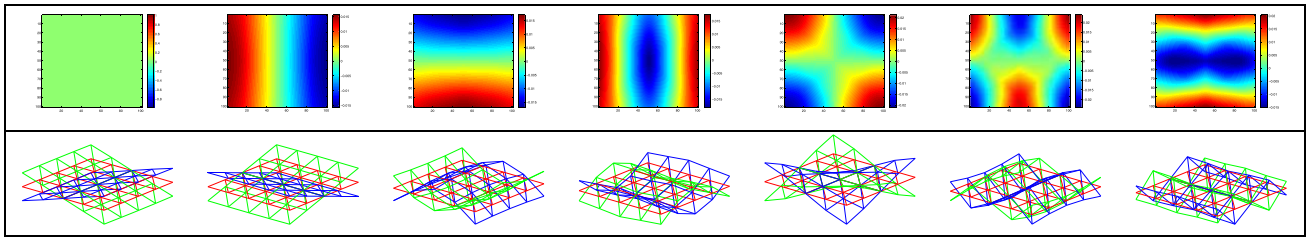


Fig. 4. Intensity and deformation modes. Top row: A subset of the low-frequency intensity modes. Bottom row: A subset of the low-frequency deformation modes. The first two encode out-of-plane rotations and the following ones are bending modes.

patches, represented by 5×5 grids of 3D points. It was acquired by attaching 3 mm wide hemispherical reflective markers to pieces of cloth, which were then waved in front of six infrared Vicon cameras to reconstruct the 3D positions of the markers. For each 3D patch, we use a standard Computer Graphics method [34] to render the patches as they would appear under our lighting model.

As a result, our training database contains pairs of 2D intensity profiles and their corresponding 3D shapes. In practice, we use 101×101 intensity patches and 5×5 3D patches, which could mean learning a mapping from a 10,201D space into an 75D one. It would require data with a large number of samples andwould be computationally difficult to achieve. Furthermore, as Lambertian surfaces evenly scatter the incoming light, they can be viewed as low-pass filters over the incident illumination. Thus, the high-frequency intensity variations tend to supply relatively little shape information and are mostly induced by noise. We therefore reduce the dimensionality of our learning problem by performing Principal Component Analysis (PCA) on both the intensity patches and the corresponding 3D deformations, and discarding high-frequency modes.

Performing PCA on the intensity patches produces an orthonormal basis of *intensity modes* and a *mean intensity patch*, as depicted by the top row of Fig. 4. Each intensity mode encodes a structured deviation from the mean intensity patch. More formally, a square intensity patch $\mathtt{I} \in \mathbb{R}^{\mathtt{w} \times \mathtt{w}}$ of width $\mathtt{w}$ can be written as

$$\mathtt{I} = \mathtt{I}_0 + \sum_{i=1}^{N_I} x_i \mathtt{I}_i, \tag{1}$$

where $\mathtt{I}_0$ is the mean intensity patch, the $\mathtt{I}_i$s are the intensity modes, the x_is are the modal weights that specify the intensity profile of the patch, and N_I denotes the number of modes. Note that, even though we learn the modes from patches of width $\mathtt{w}$, we are not restricted to that size because we can uniformly scale the modes to the desired size at runtime. As a result, the mode weights will remain invariant for similar intensity profiles at different scales.

Similarly, we parameterize the shape of a 3D surface patch as the deformations of a mesh around its undeformed state. The shape can thus be expressed as the weighted sum of *deformation modes*:

$$\mathtt{D} = \mathtt{D}_0 + \sum_{i=1}^{N_D} y_i \mathtt{D}_i, \tag{2}$$

where $\mathtt{D}_0$ is the undeformed mesh configuration, the $\mathtt{D}_i$s are the deformation modes, the y_is are the modal weights, and N_D is the number of modes.

The modes are obtained by performing PCA over vectors of vertex coordinates from many examplars of inextensibly deformed surface patches, obtained from motion capture data [37], such as those depicted by Fig. 4. Since they are naturally ordered by increasing levels of deformation, the first three always correspond to translations in the X, Y, and Z directions and the next three to a linear approximation of rotations around the three axes. We discard the in-plane deformation modes because they do not affect local patch appearance.

This being done, for each training sample we now have intensity modal weights $[x_1, \ldots, x_{N_I}]$ and deformation modal weights $[y_1, \ldots, y_{N_D}]$.

4.2 From Intensities to Deformations

Our goal is to relate the appearance of a surface patch to its 3D shape. In our context, this means using our database to learn a mapping

$$\mathcal{M} : [x_1, \ldots, x_{N_I}] \mapsto [y_1, \ldots, y_{N_D}] \tag{3}$$

that relates intensity weights to deformation weights, as illustrated by Fig. 5. Given $\mathcal{M}$, the 3D shape of a patch that does not belong to the database can be estimated by computing its intensity weights as the dot product of the vector containing its intensities and the intensity modes, mapping them to deformation modes, and recovering the 3D shape from (2).

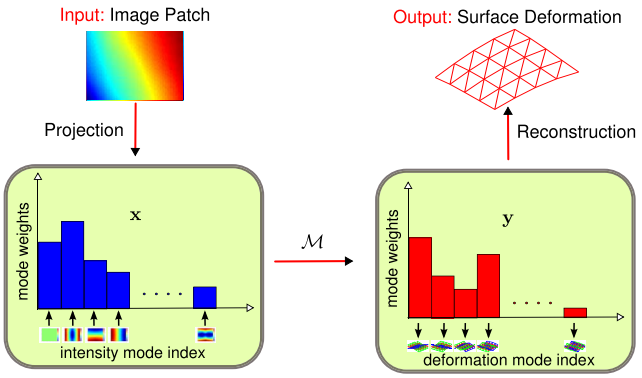


Fig. 5. Mapping from intensity to surface deformation. Projecting an intensity patch to the set of orthogonal intensity modes produces a set of intensity modal weights $\mathbf{x}$ that describe its intensity profile. Given a mapping $\mathcal{M}$ from these weights to the deformation modal weights $\mathbf{y}$, we reconstruct the shape of the patch in 3D.

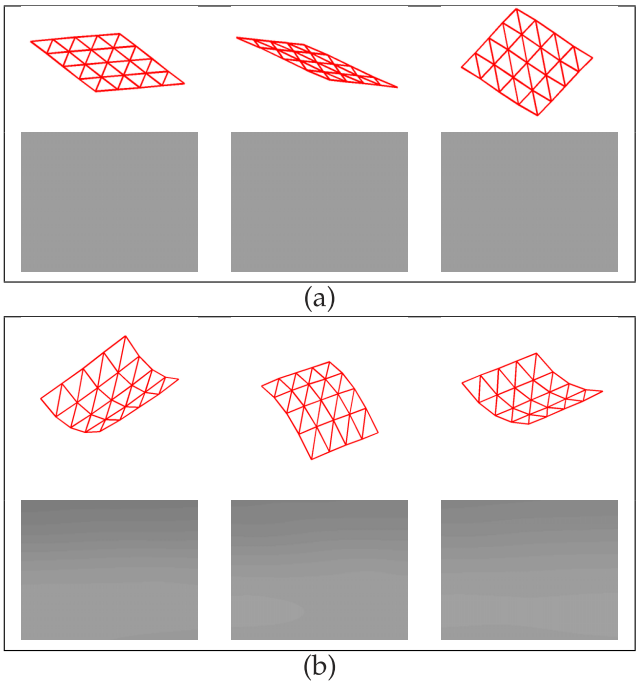


Fig. 6. Ambiguities for (a) flat and (b) deformed surfaces. First rows: Three different 3D surfaces. Second rows: Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical.

4.2.1 Gaussian Processes

Given N training pairs of intensity and deformation modes $[(\mathbf{x}^1, \mathbf{y}^1), \ldots, (\mathbf{x}^N, \mathbf{y}^N)]$, our goal is to predict the output $\mathbf{y}' = \mathcal{M}(\mathbf{x}')$ from a novel input $\mathbf{x}'$. Since the mapping from $\mathbf{x}$ to $\mathbf{y}$ is both complex and nonlinear, with no known parametric representation, we exploit the GP's ability to predict $\mathbf{y}'$ by nonlinearly interpolating the training samples $(\mathbf{y}^1 \cdots \mathbf{y}^N)$.

A GP mapping assumes a Gaussian process prior over functions, whose covariance matrix $\mathtt{K}$ is built from a covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ evaluated between the training inputs. In our case, we take this function to be the sum of a radial basis function and a noise term:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_0 \exp\left\{-\frac{\theta_1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|_2\right\} + \theta_2 \ . \tag{4}$$

It depends on the hyper parameters $\Theta = \{\theta_0, \theta_1, \theta_2\}$. Given the training samples, the behavior of the GP is only function of these parameters. Assuming Gaussian noise in the observations, they are learned by maximizing $p(\mathtt{Y}|\mathbf{x}^1, \ldots, \mathbf{x}^N, \Theta)p(\Theta)$ with respect to Θ, where $\mathtt{Y} = [\mathbf{y}^1 \cdots \mathbf{y}^N]^T$.

At inference, given the new input intensity patch coefficients $\mathbf{x}'$, the mean prediction $\mu(\mathbf{x}')$ can be expressed as

$$\mu(\mathbf{x}') = \mathtt{Y}\mathtt{K}^{-1}\mathbf{k}(\mathbf{x}'), \tag{5}$$

where $\mathbf{k}(\mathbf{x}')$ is the vector of elements $[k(\mathbf{x}', \mathbf{x}^1) \cdots k(\mathbf{x}', \mathbf{x}^N)]$ [7]. We take $\mathbf{y}'$ to be this mean prediction.

4.2.2 Partitioning the Training Data

The main difficulty in learning the mapping $\mathcal{M}$ is that it is not a function. Even though going from deformation to intensity can be achieved by a simple rendering operation, the reverse is not true. As shown in Fig. 6, many different 3D shapes can produce identical, or nearly identical, intensity profiles. These ambiguities arise from multiple phenomena, such as rotational ambiguity, convex-concave ambiguity [22], or bas-relief ambiguity [4].

As a result, many sets of deformation weights can correspond to a single set of intensity weights. Since GPs are not designed to handle one-to-many mappings, training one using all the data simultaneously produces meaningless results.

Observing the ambiguous configurations reveals that the ambiguity is particularly severe when the surface patch remains planar and only undergoes rotations. Recall that the principal components of out-of-plane rotations are encoded by the first two deformation modes, which are depicted at the bottom left of Fig. 4, and the corresponding y_1 and y_2 weights. In Fig. 7a, we plot the contour curves for the rendered intensities of planar patches in various orientations obtained by densely sampling y_1, y_2 space. This shows that there are infinitely many combinations of y_1 and y_2 that represent a planar patch with the same intensity. Since y_1 and y_2 encode the amount of out-of-plane rotation, a line emanating from the center of the isocontours in the y_1, y_2 space defines a particular surface normal orientation and, within angular slices such as those depicted by the alternating green and white quadrants of Fig. 7a, the surface normal of the corresponding patch remains within a given angular distance of an average orientation. We can therefore reduce the reconstruction ambiguities by splitting the y_1, y_2 space into such angular slices and learning one local GP per slice. In practice, we use 20 local GPs to cover the whole space. This resembles the clustering scheme proposed in [46], but with a partitioning scheme adapted to our problem. Other schemes, such as defining boxes in the y_1 and y_2 dimensions, would, of course, have been possible. However, since the dominant source of ambiguity appears to be the average surface normal that is encoded by the ratio of y_1 to y_2, we experimentally found our angular partitioning to be more efficient than others.

In Fig. 7b, we demonstrate the benefit of using local GPs over a global one to reconstruct a uniform flat patch from its intensities. The predictions from multiple GPs correctly sample the isointensity contour that encodes the family of all orientations producing the same intensity. In Fig. 7c, we consider the case of a deformed patch and plot the mean and variance values of the vertex-to-vertex distances between the prediction and ground truth. For each slice we tested 100 unique patch deformations while training over 1,000 data points. We repeated this 100 times. The average reconstruction error of 3 millimeters is small considering that the average patch side is 100 millimeters long. This indicates that, within each partition, there is a one-to-one correspondence between intensity and deformation mode weights. Otherwise, the GP mapping could not produce this accuracy.

One attractive feature of GPs is that they can be learned from a relatively small training set. We estimate the required size empirically by measuring the accuracy of the mapping, given by the average vertex-to-vertex distance between the prediction and ground truth data, as a function of the number of training samples. For a given size, we draw 100 independent subsets of samples of that size from

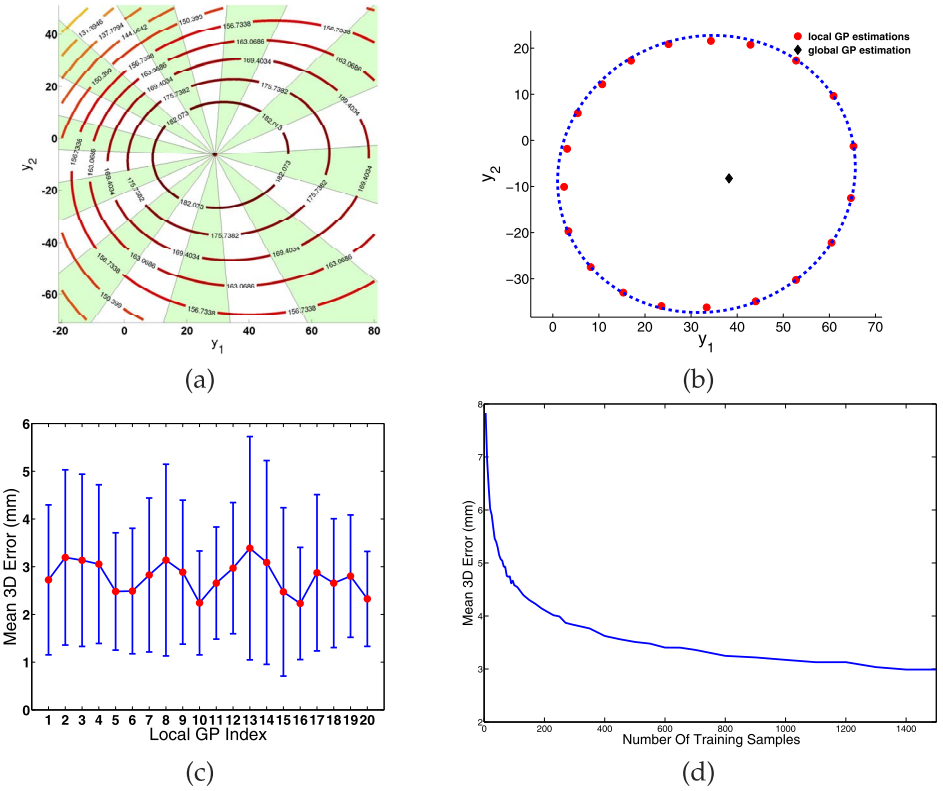


Fig. 7. Single versus multiple GPs. (a) Given a uniform intensity patch, there are infinitely many 3D planar patches that could have generated it. In our scheme, they are parameterized by the y_1 and y_2 weights assigned to the first two deformation modes, which encode out-of-plane rotations. The ovals represent isointensity values of these patches as a function of y_1 and y_2. (b) If we train a GP using *all* the training samples simultaneously, it will predict the same erroneous surface orientation depicted by the black dot for any uniform intensity patch. If we first partition the training samples according to angular slices shown in green and white in (a) and train a GP for each, we can predict the patch orientation, shown as blue dots, which are much closer to the true orientations, shown in red. (c) Mean and variance of the vertex-to-vertex distance between the predicted patch deformations and the ground-truth shapes for each local GP. (d) Accuracy of a local GP as a function of the number of training samples. GPs are accurate even when using as few as 1,000 samples. In our experiments, for each local GP, we use 1,400 samples, on average, from the training set.

our training set. For each subset, we test the accuracy using 100 other instances from the test set. The resulting mean error is depicted by Fig. 7d.

4.3 Local Reconstructions from an Input Image

At runtime, we first identify the textured patches by extracting SIFT interest points and establishing point correspondence with the reference image. They are used to recover their 3D-shape using the correspondence-based method [35], which we briefly summarize in Appendix A1, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.196. We then scan the remainder of the image multiple times with square sliding windows of varying sizes, starting with a large one and progressively decreasing its size. During each scan, the windows whose intensity variance is greater than a threshold are discarded. The remaining ones are projected into the learned intensity mode space and retained if their mode-space distance to their nearest neighbor in the training set is smaller than a threshold. In successive scans, we ignore areas that are completely subsumed by previously selected windows. Finally, we run a connected component analysis and keep only the patches that are connected directly or indirectly to the textured one. In all our experiments, we keep the maximum acceptable standard deviation of intensities in a patch to be 30 units and mode-space distance to be 10.

Given a set of featureless patches and N_{GP} Gaussian Processes, one for each angular partition of the training data, we therefore predict N_{GP} shape candidates per patch, represented as 5×5 meshes. We initially position them in 3D with their center at a fixed distance along the line of sight defined by the center of the corresponding image patch.

5 Enforcing Global Consistency

Local shape estimation returns a set $\mathcal{S}_p = \{\mathbb{S}_p^1, \cdots, \mathbb{S}_p^{N_{GP}}\}$ of plausible shape interpretations reconstructed up to a scale factor for each patch p, and a single one $\mathbb{S}_{p'}$ for each textured patch p'. To produce a single global shape interpretation, we go through the two following steps.

First, we choose one specific interpretation for each featureless patch. To this end, we use an MRF to enforce global consistency between the competing interpretations in a way that does not require knowing their scales. Second, we compute the scale of each patch, or equivalently its distance to the camera, by solving a set of linear equations. In the remainder of this section, we describe these two steps in more details.

5.1 Selecting One Shape Interpretation per Patch

To select the correct interpretation for individual patches, we treat each one as a node in an MRF graph. Featureless patches can be assigned one of the N_{GP} labels corresponding

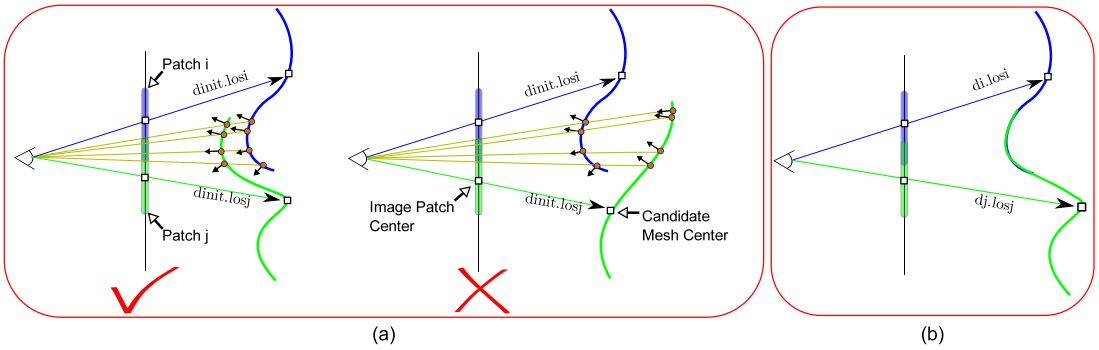


Fig. 8. Enforcing shape consistency. (a) Two different instances of the evaluation of the geometric consistency of patches i and j, shown in blue and green, respectively. In both cases, the predicted normals of points along the same lines of sight, drawn in yellow, are compared. Since these points have the same projections, their normals should agree. Thus, the patches on the left are found to be more consistent than those on the right. (b) Moving patches along their respective lines of sight. The patches i and j are moved to distances d_i and d_j from the optical center so as to minimize the distance between them in their regions of overlap.

to the elements of $\mathcal{S}_p$, while textured ones are assigned their recovered shape label.

We take the total energy of the MRF graph to be the sum over all the featureless local patches:

$$\mathtt{E} = \sum_p \left(\mathtt{E}_1(\mathtt{S}_p) + \frac{1}{2} \sum_{q \in \mathcal{O}(p)} \mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q) \right), \tag{6}$$

where $\mathcal{O}(p)$ is the set of patches that overlap p. The unary terms $\mathtt{E}_1$ favor shapes whose shaded versions match the image as well as possible. The pairwise terms $\mathtt{E}_2$ favor geometric consistency of overlapping shapes.

In practice, we take $\mathtt{E}_1(\mathtt{S}_p)$ to be the inverse of the normalized cross correlation score between the image patch and the rendered image of the 3D shape. To evaluate the pairwise term $\mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q)$ for overlapping patches p and q, we shoot multiple camera rays from the camera center through their common projection area, as shown in Fig. 8a. For each ray, we compare the normals of the two 3D shapes and take $\mathtt{E}_2(\mathtt{S}_p, \mathtt{S}_q)$ to be the mean L2 norm of the difference between the normals.

Note that both the unary and pairwise terms of (6) can be evaluated without knowing the scale of the patches, which is essential in our case because it is indeed unknown at this stage of the computation. We use a tree reweighted message passing technique [21] to minimize the energy. In all of our experiments, the primal and dual programs returned the same solution [5], which indicates the algorithm converged to a global optimum even though the energy includes nonsubmodular components.

5.2 Aligning the Local Patches

Having assigned a specific shape $\mathtt{S}_p$ to each patch, we now need to scale these shapes by moving them along their respective lines of sight, which comes down to computing the distances d_p from the optical center to the patch centers. In the camera referential, the line of sight defined by the center of patch p emanates from the origin and its direction is

$$\mathrm{los}_p = \frac{\mathtt{A}^{-1}\mathbf{c}_p}{\|\mathtt{A}^{-1}\mathbf{c}_p\|_2}, \tag{7}$$

where $\mathtt{A}$ is the 3×3 matrix of internal camera parameters and $\mathbf{c}_p$ represents the projective coordinates of the patch center.

To enforce scale consistency between pairs of overlapping patches p and q, we consider the same point samples as before, whose projections lie in the overlap area as shown in Fig. 8b. Let $[x_p, y_p, z_p]^T$ and $[x_q, y_q, z_q]^T$ be the 3D coordinates of the vectors connecting such a sample to the centers of p and q, respectively. Since they project to the same image location, we must have

$$d_p\big(\mathrm{los}_p^T + [x_p, y_p, z_p]\big) = d_q\big(\mathrm{los}_q^T + [x_q, y_q, z_q]\big). \tag{8}$$

Each sample yields one linear equation of the form of (8). Thus, given enough samples, we can compute all the d_p up to a global scale factor by solving the resulting system of equations in the least-squares sense. If there is at least one textured patch whose depth can be recovered accurately, the global scale can be fixed and this remaining ambiguity resolved.

5.3 Post Processing

The alignment yields a set of overlapping 3D shapes. To make visual interpretation easier, we represent them as point clouds which are computed by linearly interpolating the z values of the vertices of all the local solutions on a uniformly sampled xy grid. For display purposes, we either directly draw these points or the corresponding Delaunay triangulation.

6 Results

In this section, we demonstrate our method's ability to reconstruct different kinds of surfaces. In all these experiments, we learned 20 independent GPs by partitioning the space of potential surface normals, as discussed in Section 4.2. For training purposes, we used 28,000 surface patches, or approximately 1,400 per GP. They are represented as 5×5 meshes and rendered using the calibrated experiment-specific lighting environment. The calibration and the training process jointly take approximately 2 hours to complete on a standard machine with a 2.4 GHz processor.

In the remainder of this section, we first use synthetic data to analyze the behavior of our algorithm. We then demonstrate its performance on real data and validate our results against ground-truth data.

Since our images contain both textured and nontextured parts, we compare our results to those obtained using our

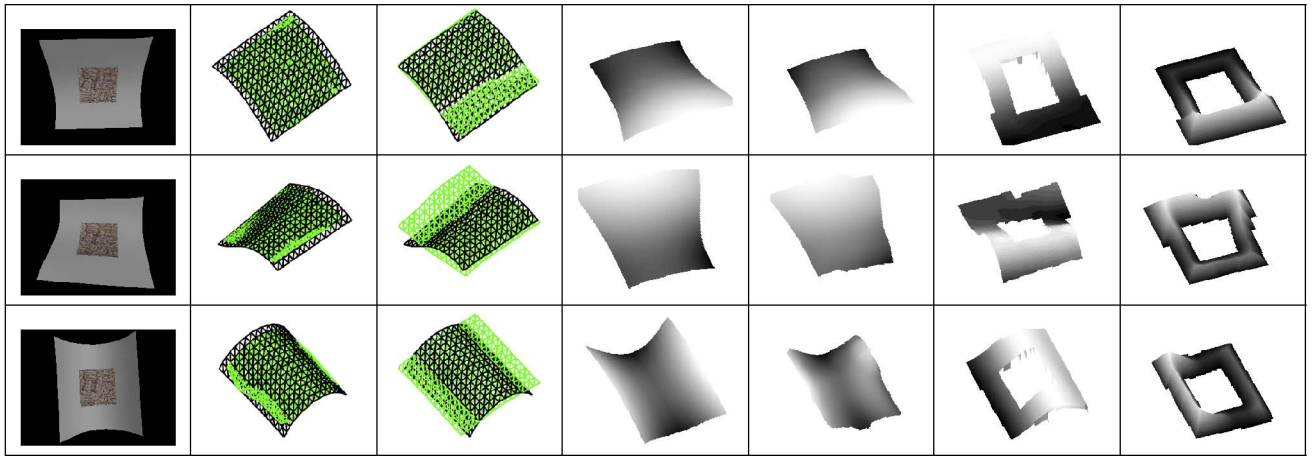


Fig. 9. Synthetic experiments. First column: Input images. Second column: Estimated triangulation (green) and ground-truth triangulation (black) seen from another view point. Third column: Reconstruction results using the method in [35] (green mesh) and ground-truth triangulation (black). Fourth column: Ground truth depth maps. Fifth column: Depth maps computed from our reconstructions. Sixth and last columns: Depth maps computed by the methods of [45] and [14], respectively.

earlier technique [35] that relies solely on point correspondences to demonstrate that also using the shading information does indeed help. We also compare against pure shape-from-shading algorithms described in [45] and [14] that are older but, as argued in [12], still representative of the state of the art, and whose implementations are available online.

6.1 Synthetic Images

We first tested the performance of our algorithm on a synthetic sequence created by rendering 100 different deformations of a piece of cardboard obtained using a motion capture system. Note that this is not the same sequence as the one we used for learning the intensity to deformation mapping discussed in Section 4. The entire sequence is rendered using the lighting parameters corresponding to a complicated lighting environment such as the one shown in Fig. 3. To this end, we use a set of spherical harmonics coefficients computed for that particular lighting environment. In addition, the central part of the surface is artificially texture mapped. Fig. 9 depicts a subset of these synthetic images, the 3D reconstructions we derive from them, and 3D reconstructions obtained with our earlier texture-based method [35].

By combining shading and texture cues, our method performs significantly better except for flat surfaces, where both methods return similar results. In the same figure, we also compare our results against those of pure shape-from-shading methods [14], [45]. Our algorithm computes a properly scaled 3D surface, but these methods only return a normalized depth map. For a fair comparison, we, therefore, computed normalized depth maps from our results. Furthermore, although our method does not require it, we provided manually drawn masks that hide the background and the textured parts of the surfaces to make the task of the shape-from-shading methods easier. As can be seen, in addition to being correctly scaled, our reconstructions are also considerably more accurate.

We compute 3D reconstruction errors as the mean point-to-surface distances from the reconstructed point clouds to the ground-truth surfaces for all the frames in this sequence. The results are shown in Fig. 10. In Appendices A2 and A3, available in the online supplemental material, we show our 3D reconstruction errors ordered with respect to the complexity of the surface deformations, and demonstrate the robustness of our approach to image noise.

In theory, there is no guarantee that the reconstruction of the textured patch is correct, which could lead to reconstruction failure. An incorrect reconstruction will result in a gross error, especially since our algorithm tries to enforce global consistency with respect to this erroneous configuration. In practice, this only rarely occurs, and in all the correspondence-based experiments reported here, the algorithm we used [35] returned a valid reconstruction for the textured area. Nevertheless, our method can also handle cases when there are multiple interpretations for the textured patches by adding them as additional labels to our MRF. To demonstrate this, we generated multiple candidates for the textured patches using the sampling scheme proposed in [25]. As shown in Fig. 11, our algorithm picks the right one from the candidate reconstructions.

6.1.1 Robustness to Lighting Environment

To show that our algorithm is robust to lighting changes, we rendered images of the same surface under three different lighting arrangements with either frontal, on left, or on righ, lighting. As shown in Fig. 12, the three reconstructions that we obtained were all similar and close to the ground truth.

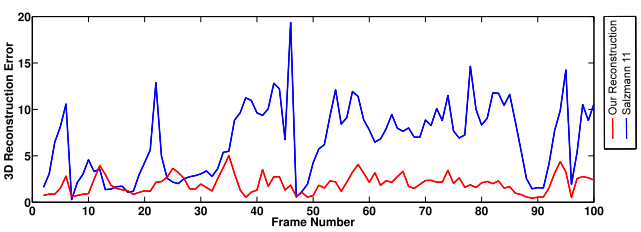


Fig. 10. Reconstruction error of both methods for all the frames in the sequence. Note that the proposed method provides much better reconstructions, except for six frames in the sequence.

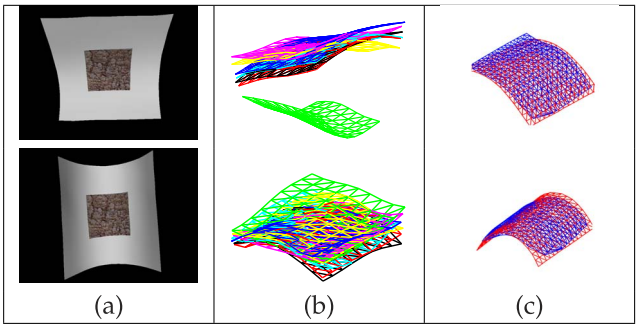


Fig. 11. Reconstruction with multiple 3D hypotheses for the textured patch. (a) Input images. (b) 3D hypotheses for the textured patches. (c) 3D reconstructions of the surfaces.

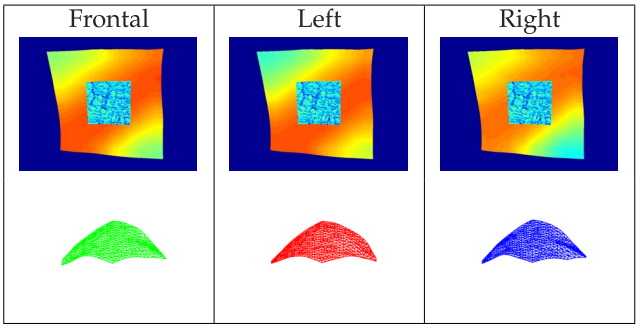
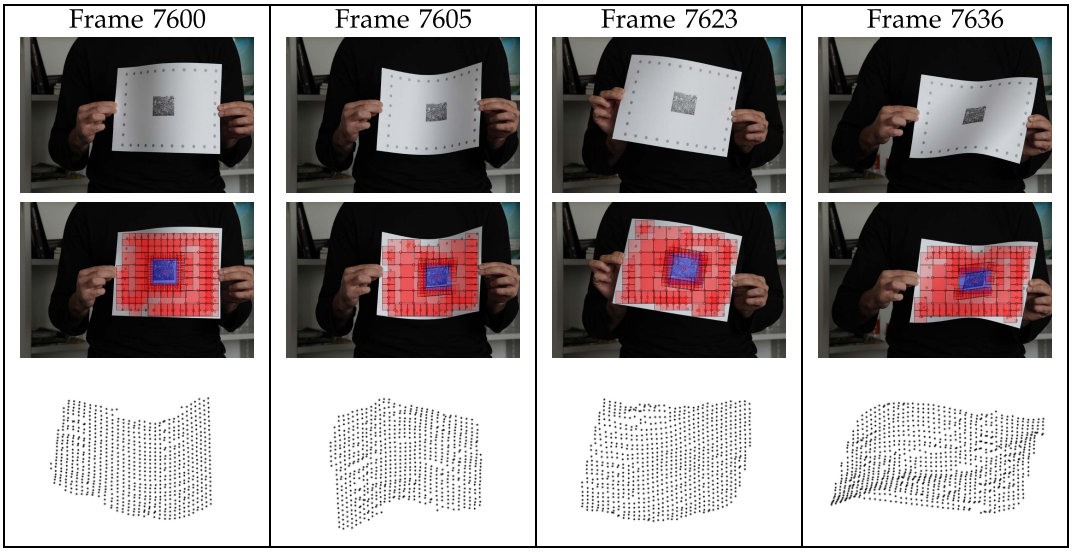


Fig. 12. Robustness to lighting environment. The surface is lit by three different lighting schemes. Top row: Intensity variation in the surface. Bottom row: Reconstructed surfaces.

6.2 Real Images

As the nature of the deformations varies considerably with respect to the surface material type, we applied our reconstruction algorithm to two surfaces with very different physical properties: the piece of paper of Fig. 13 and the T-shirt of Fig. 14. The deformation of the latter is significantly less constrained than that of the former. Note that because we only model the deformations of small patches that are then assembled into global surfaces, we can handle complex global deformations. However, as will be discussed below, folds that are too sharp may result in self-shadowing, which is not handled in our current implementation.



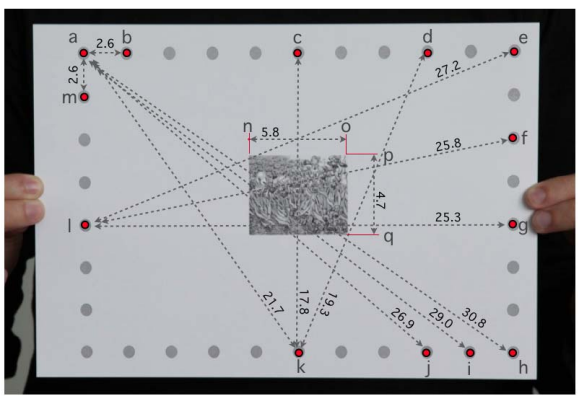
Reference Image	Point Pairs	Ground Truth	Frame Number 7600	7605	7623	7636
	a – b	2.6	2.5	2.6	2.6	3.2
	a – h	30.8	30.6	29.4	31.0	32.6
	a – i	28.9	28.2	27.9	28.9	30.1
	a – j	26.9	26.1	26.1	27.1	27.6
	a – k	21.7	21.2	22.6	22.5	22.1
	a – m	2.6	2.8	2.8	2.9	2.9
	c – k	17.8	18.0	18.0	18.4	18.9
	d – k	19.3	20.2	20.0	20.2	NA
	e – l	27.2	27.3	26.3	26.9	28.5
	f – l	25.8	26.3	24.9	25.2	27.8
	g – l	25.3	25.6	24.3	24.8	27.5
	n – o	5.8	5.9	5.9	5.9	6.1
	p – q	4.7	4.8	4.9	4.8	5.1
All distances are in cm	Avg Error		0.36	0.65	0.35	1.03

Fig. 13. Paper sequence. First row: Input images. Second row: Local patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. Third row: Reconstructed point cloud seen from another viewpoint. Fourth row: Geodesic distances between prominent landmarks as identified on the left. Point d in frame 7,636 was outside our reconstruction, which explains the missing value in the table.

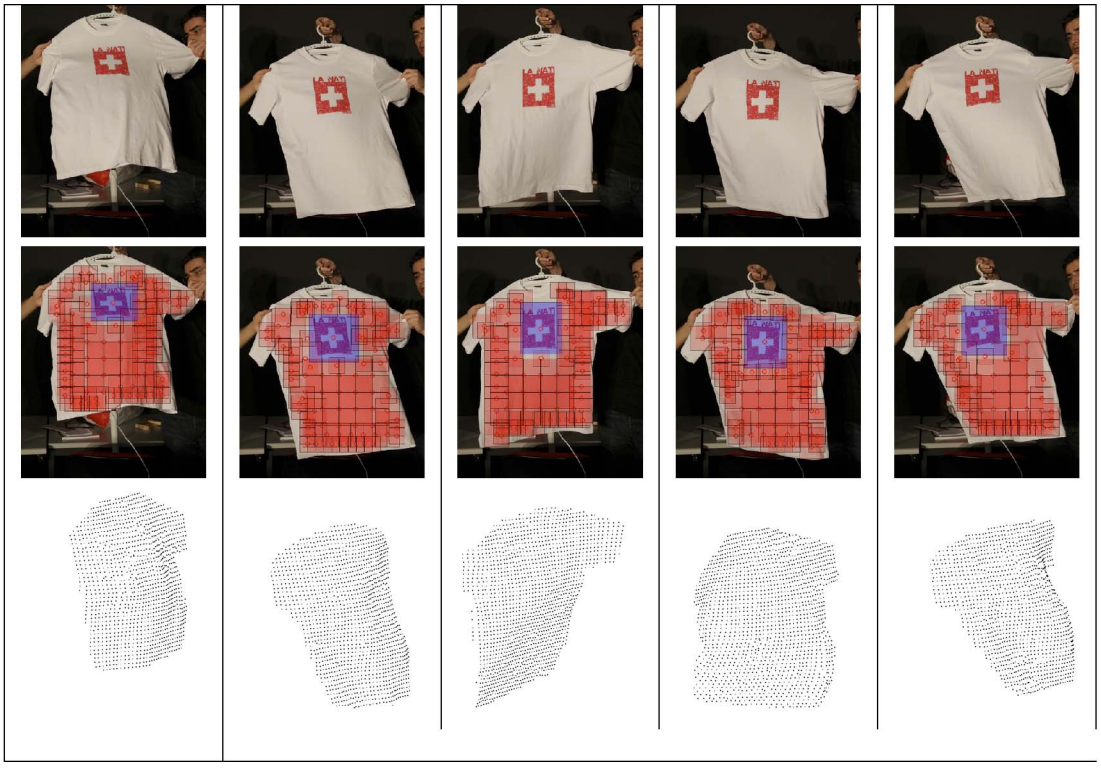


Fig. 14. T-shirt sequence. First row: Input images. Second row: Local patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. Third row: Reconstructed point cloud seen from another viewpoint.

The real sequences were captured by a Single-Lens Reflex (SLR) camera and recorded in raw format. The linear images were then extracted from the raw image files and the image intensities linearly scaled so that they cover most of the observable intensity range. The image resolution was approximately 5 mega-pixels.

The image patches of Section 4 were selected by the patch selection algorithm. In practice, we used square patches whose size ranges from 401 to 101 pixels with a 100 pixels step. We show the textured and textureless image patches selected by this procedure in the second rows of Figs. 13 and 14.

6.3 Validation

To quantitatively evaluate our algorithm's accuracy, we performed two different sets of experiments involving real data, which we detail below.

6.3.1 Preservation of Geodesic Distances

The geodesic distances between pairs of points, such as the circles on the piece of paper at the bottom of Fig. 13, remain constant no matter what the deformation is because the surface is inextensible. As shown in the bottom-right table, even though we do not explicitly enforce this constraint, it remains satisfied to a very high degree, thus indicating that the global deformation is at least plausible.

In this example, the ground-truth geodesic distances were measured when the sheet of paper was lying flat on a table. To compute the geodesic distances on the recovered meshes, we used an adapted Gauss-Seidel iterative algorithm [8].

6.3.2 Comparison against Structured Light Scans

To further quantify the accuracy of our reconstructions, we captured surface deformations using a structured light scanner [48]. To this end, we fixed the shape of the same piece of paper and T-shirt as before by mounting them on a hardboard prior to scanning, as shown at the top of Fig. 15. Because of the physical setup of the scanner, we then had to move the hardboard to acquire the images we used for reconstruction purposes. To compare our reconstructions to the scanned values, we therefore used an ICP algorithm [6] to register them together.

In the remainder of Fig. 15, we compare the output of our algorithm to that of the same algorithms as before. These results clearly indicate that our approach to combining texture and shading cues produces much more accurate results than those of these other methods that only rely on one or the other.

6.4 Limitations

The main limitation of our current technique is that, outside of the truly textured regions, we assume the surface to be Lambertian and of constant albedo. As a result, we cannot reconstruct shiny objects, such as the balloon shown in Fig. 16a. However, given the Bidirectional Reflectance Distribution Function (BRDF) of the surface points, our framework could, in theory, be extended to such non-Lambertian surfaces.

Like most other shape-from-shading methods, ours is ill-equipped to handle self-shadows and occlusions. The effect of the latter can be mitigated to a certain extent by using temporal models. The self-shadows produced by sharp folds, such as the ones shown in Fig. 16b, violate our basic

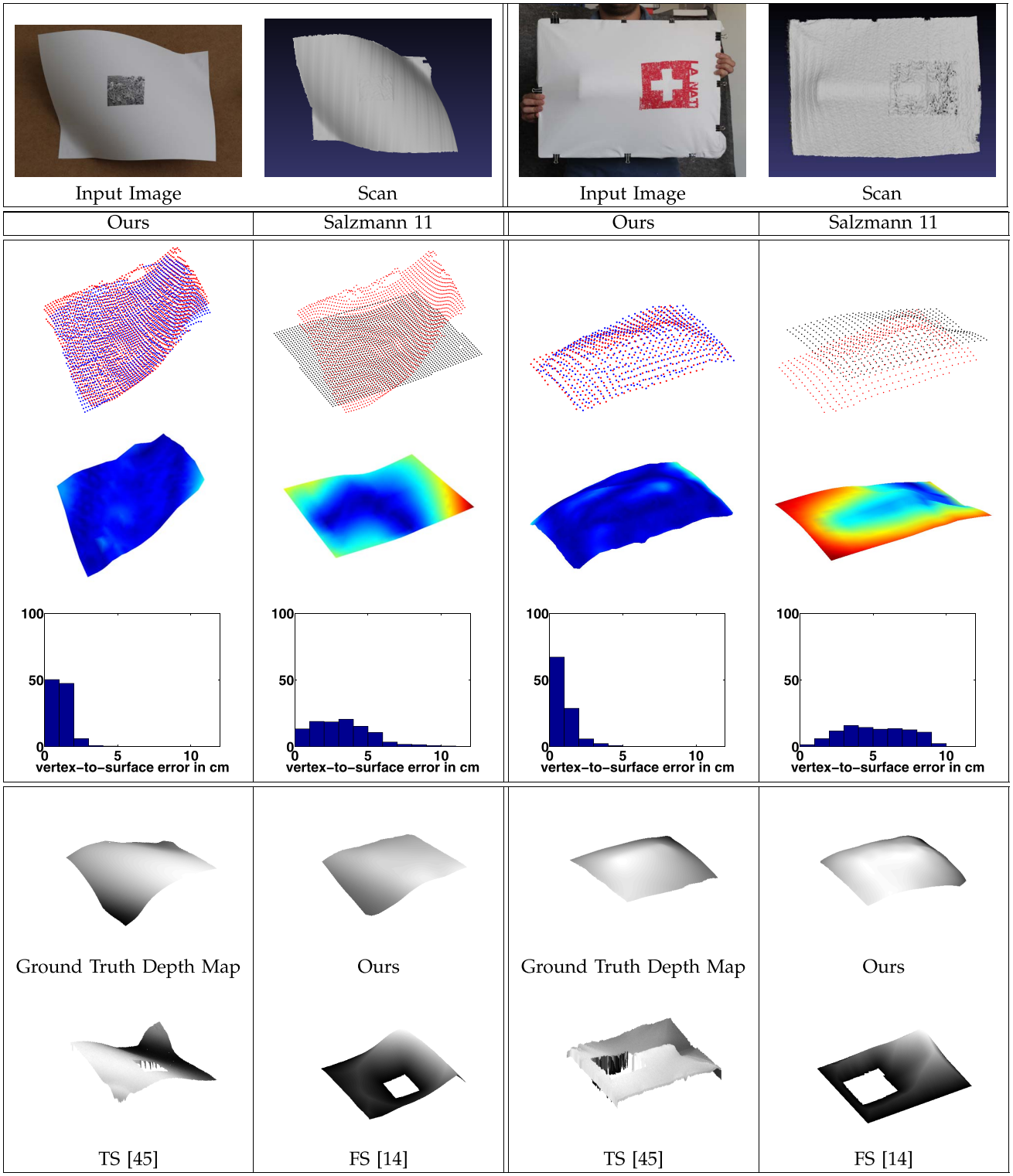
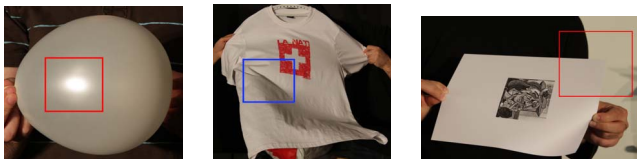


Fig. 15. Accuracy estimation using structured light scans. Top row: Two different surfaces and their corresponding structured light scans. Middle block: From top to bottom, point clouds from the scans (red) and reconstructions by either our algorithm or that of [35] (green), reconstructed 3D surface rendered using a color going from blue to red as the vertex-to-surface distance to the ground-truth increases, and corresponding histogram of vertex-to-surface distances. Bottom block: Depth maps obtained from our reconstructions and from the methods in [14] and [45].

assumptions that shading only depends on 3D shape, and could be handled by a separately trained generative model [19]. Addressing these issues is a topic for future research.

Failure may also occur when background image regions are extracted by our patch selection algorithm. Fortunately, this rarely occurs, i.e., when the background is made up of uniform regions with albedo very similar to that of the surface to be reconstructed, as shown in Fig. 16c. In such cases, the background patches look very similar to those of our training set and will not be filtered out.



(a) (b) (c)

Fig. 16. Failure modes. (a) Non-Lambertian surface. (b) Folds that create self-shadows. (c) Background albedo very similar to the surface.

7 Conclusion

We have presented an approach to monocular shape recovery that effectively takes advantage of both shading cues in nontextured areas and point correspondences in textured ones under realistic lighting conditions and under full perspective projection. We have demonstrated the superior accuracy of our approach compared to state-of-the-art techniques on both synthetic and real data.

Our framework is general enough so that each component could be replaced with a more sophisticated one. For instance, representations of the lighting environment more sophisticated than spherical harmonics could be used to create our training set. Similarly, other, potentially nonlinear parametrizations of the patch intensities and deformations could replace the current PCA mode weights.

Acknowledgments

This work has been supported in part by the Swiss National Science Foundation. In addition, the authors would like to thank Fethallah Benmansour for providing the code for computing the geodesic distance on a triangulated surface, Thibaut Weise for letting them use his structured light 3D scanner system, and Jean-Denis Durou for providing the implementation of various Shape from Shaping methods online.

References

[1] H. Aanaes and F. Kahl, "Estimation of Deformable Structure and Motion," *Proc. Vision and Modelling of Dynamic Scenes Workshop,* 2002.

[2] A. Ahmed and A. Farag, "A New Formulation for Shape from Shading for Non-Lambertian Surfaces," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* June 2006.

[3] I. Akhter, Y. Sheikh, and S. Khan, "In Defense of Orthonormality Constraints for Nonrigid Structure from Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2009.

[4] P. Belhumeur, D. Kriegman, and A. Yuille, "The Bas-Relief Ambiguity," *Int'l J. Computer Vision,* vol. 35, no. 1, pp. 33-44, 1999.

[5] D.P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1999.

[6] P. Besl and N. Mckay, "A Method for Registration of 3D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 239-256, Feb. 1992.

[7] C.M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[8] F. Bornemann and C. Rasch, "Finite-Element Discretization of Static Hamilton-Jacobi Equations Based on a Local Variational Principle," *Computing and Visualization in Science,* vol. 9, no. 2, pp. 57-69, 2006.

[9] M. Brand, "A Direct Method of 3D Factorization of Nonrigid Motion Observed in 2D," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 122-128, 2005.

[10] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering Non-Rigid 3D Shape from Image Streams," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2000.

[11] T. Collins and A. Bartoli, "Locally Affine and Planar Deformable Surface Reconstruction from Video," *Proc. Int'l Workshop Vision, Modeling and Visualization,* pp. 339-346, 2010.

[12] J.-D. Durou, M. Falcone, and M. Sagona, "Numerical Methods for Shape from Shading: A New Survey with Benchmarks," *Computer Vision and Image Understanding,* vol. 109, pp. 22-43, 2008.

[13] A. Ecker, A.D. Jepson, and K.N. Kutulakos, "Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities," *Proc. European Conf. Computer Vision,* Oct. 2008.

[14] M. Falcone and M. Sagona, "An Algorithm for Global Solution of the Shape-from-Shading Model," *Proc. Int'l Conf. Image Analysis and Processing,* 1997.

[15] J. Fayad, L. Agapito, and A. Del Bue, "Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences," *Proc. 11th European Conf. Computer Vision,* 2010.

[16] J. Fayad, A. Del Bue, L. Agapito, and P.M.Q. Aguiar, "Non-Rigid Structure from Motion Using Quadratic Deformation Models," *Proc. British Machine Vision Conf.,* 2009.

[17] D.A. Forsyth and A. Zisserman, "Reflections on Shading," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 7, pp. 671-679, July 1991.

[18] N.A. Gumerov, A. Zandifar, R. Duraiswami, and L.S. Davis, "Structure of Applicable Surfaces from Single Views," *Proc. European Conf. Computer Vision,* May 2004.

[19] M. Han, W. Xu, H. Tao, and Y. Gong, "An Algorithm for Multiple Object Trajectory Tracking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 864-871, June 2004.

[20] B.K.P. Horn and M.J. Brooks, *Shape from Shading.* MIT Press, 1989.

[21] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1568-1583, Oct. 2006.

[22] R. Kozera, "Uniqueness in Shape from Shading Revisited," *J. Math. Imaging and Vision,* vol. 7, no. 2, pp. 123-138, 1997.

[23] D.J. Kriegman and P.N. Belhumeur, "What Shadows Reveal about Object Structure," *Proc. European Conf. Computer Vision,* pp. 399-414, 1998.

[24] J. Liang, D. Dementhon, and D. Doermann, "Flattening Curved Documents in Images," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 338-345, 2005.

[25] F. Moreno-Noguer, J. Porta, and P. Fua, "Exploring Ambiguities for Monocular Non-Rigid Shape Estimation," *Proc. European Conf. Computer Vision,* Sept. 2010.

[26] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fu, "Capturing 3D Stretchable Surfaces from Single Images in Closed Form," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2009.

[27] S.K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from Interreflections," *Int'l J. Computer Vision,* vol. 6, no. 3, pp. 173-195, 1991.

[28] S.I. Olsen and A. Bartoli, "Implicit Non-Rigid Structure-From-Motion with Priors," *J. Math. Imaging and Vision,* vol. 31, pp. 233-244, 2008.

[29] M. Oren and S.K. Nayar, "A Theory of Specular Surface Geometry," *Int'l J. Computer Vision,* vol. 24, no. 2, pp. 105-124, 1996.

[30] M. Perriollat and A. Bartoli, "A Quasi-Minimal Model for Paper-Like Surfaces," *Proc. BenCos: Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images,* 2007.

[31] M. Perriollat, R. Hartley, and A. Bartoli, "Monocular Template-Based Reconstruction of Inextensible Surfaces," *Proc. British Machine Vision Conf.,* 2008.

[32] V. Rabaud and S. Belongie, "Re-Thinking Non-Rigid Structure from Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2008.

[33] V. Rabaud and S. Belongie, "Linear Embeddings in Non-Rigid Structure from Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2009.

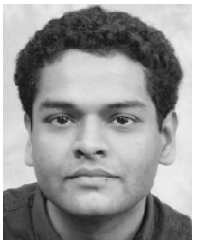
[34] R. Ramamoorthi and P. Hanrahan, "An Efficient Representation for Irradiance Environment Maps," *Proc. ACM Siggraph,* 2001.

[35] M. Salzmann and P. Fua, "Linear Local Models for Monocular Reconstruction of Deformable Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 5, pp. 931-944, May 2011.

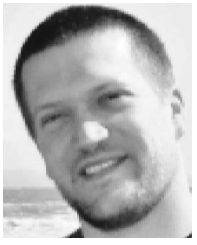
[36] M. Salzmann, V. Lepetit, and P. Fua, "Deformable Surface Tracking Ambiguities," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2007.

[37] M. Salzmann, R. Urtasun, and P. Fua, "Local Deformation Models for Monocular 3D Shape Recovery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2008.
[38] D. Samaras and D. Metaxas, "Incorporating Illumination Constraints in Deformable Models," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 322-329, June 1998.
[39] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc, "Variable Albedo Surface Reconstruction from Stereo and Shape from Shading," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2000.
[40] A. Shaji and S. Chandran, "Riemannian Manifold Optimisation for Non-Rigid Structure from Motion," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[41] S. Shen, W. Shi, and Y. Liu, "Monocular 3D Tracking of Inextensible Deformable Surfaces under L2-Norm," *Proc. Asian Conf. Computer Vision,* 2009.
[42] J. Taylor, A.D. Jepson, and K.N. Kutulakos, "Non-Rigid Structure from Locally-Rigid Motion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2010.
[43] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method," *Int'l J. Computer Vision,* vol. 9, no. 2, pp. 137-154, 1992.
[44] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid Structure-From-Motion: Estimating Shape and Motion With Hierarchical Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 5, pp. 878-892, May 2008.
[45] P.S. Tsai and M. Shah, "Shape from Shading Using Linear Approximation," *J. Image and Vision Computing,* vol. 12, pp. 487-498, 1994.
[46] R. Urtasun and T. Darrell, "Sparse Probabilistic Regression for Activity-Independent Human Pose Inference," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.
[47] A. Varol, M. Salzmann, E. Tola, and P. Fua, "Template-Free Monocular Reconstruction of Deformable Surfaces," *Proc. Int'l Conf. Computer Vision,* Sept. 2009.
[48] T. Weise, B. Leibe, and L. Van Gool, "Fast 3D Scanning with Automatic Motion Compensation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2007.
[49] R. White and D.A. Forsyth, "Combining Cues: Shape from Shading and Texture," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2006.
[50] J. Xiao, J.-X. Chai, and T. Kanade, "A Closed-Form Solution to Non-Rigid Shape and Motion Recovery," *Proc. European Conf. Computer Vision,* pp. 573-587, 2004.
[51] Z. Zhang, C. Tan, and L. Fan, "Restoration of Curved Document Images through 3D Shape Modeling," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* June 2004.
[52] J. Zhu, S. Hoi, C. Steven, Z. Xu, and M.R. Lyu, "An Effective Approach to 3D Deformable Surface Tracking," *Proc. European Conf. Computer Vision,* pp. 766-779, 2008.

Aydin Varol received the BSc degrees in both computer and mechanical engineering in 2005 from Koc University, Turkey. Afterward, he received the MSc degree from the same university in 2007. He then joined the Computer Vision Laboratory, EPFL (Swiss Federal Institute of Technology), where he is currently working toward the PhD degree. His research interests focus on monocular nonrigid surface reconstruction.



Appu Shaji received the BTech degree in computer science in 2002 from Cochin University of Science and Technology and the PhD degree in computer science in 2009 from the Indian Institute of Technology Bombay, Mumbai. He then joined the Computer Vision Lab at EPFL (Swiss Federal Institute of Technology) in 2009 as a postdoctoral fellow. His research interests include nonrigid shape recovery, structure from motion, image registration, and optimization techniques for computer vision.



Mathieu Salzmann received the BSc and MSc degrees in computer science in 2004 and the PhD degree in computer vision in 2009 from EPFL (Swiss Federal Institute of Technology). He then joined the International Computer Science Institute and the Electrical Engineering and Computer Science Department at the University of California Berkeley as a postdoctoral fellow and in 2010 at TTI Chicago as a research assistant professor. Currently, he is working as a researcher at NICTA in Canberra, Australia. His research interests include nonrigid shape recovery, human pose estimation, machine learning, and optimization techniques for computer vision.

Pascal Fua received the engineering degree from the Ecole Polytechnique, Paris, in 1984 and the PhD degree in computer science from the University of Orsay in 1989. He joined EPFL (Swiss Federal Institute of Technology) in 1996, where he is now a professor in the School of Computer and Communication Science. Before that, he worked at SRI International and at INRIA Sophia-Antipolis as a computer scientist. His research include shape modeling and motion recovery from images, human body modeling, and optimization-based techniques for image analysis and synthesis. He has (co)authored more than 150 publications in referred journals and conferences. He has been an associate editor of the *IEEE Transactions for Pattern Analysis and Machine Intelligence* and has been a program committee member and an area chair of several major vision conferences. He is a fellow of the IEEE.