Monocular 3D Reconstruction of Locally Textured Surfaces

Aydin Varol, Appu Shaji, Mathieu Salzmann and Pascal Fua

Abstract—
Most recent approaches to monocular non-rigid 3D shape recovery rely on exploiting point correspondences and work best when the whole surface is well-textured. The alternative is to rely either on 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], [29], [33], [38], [39], [44], [49], 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], [23], [28], [48]. Some take advantage of shading information, but typically only to disambiguate the information provided by the interest points or the contours [46]. 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 run time, 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 parametrized 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 Gaps 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).

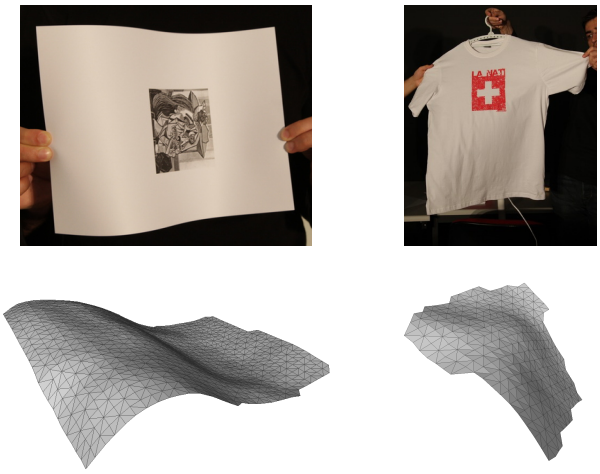


Fig. 1. 3D reconstruction of two poorly-textured deformable surfaces from single images.

We exploit the 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 [33] 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 Prue-segment the surfaces 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 tackles 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 non-rigid 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.

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, this amounts to solving an ill-conditioned linear system [34] and additional constraints must be imposed to obtain an acceptable solution. These may include inextensibility as well as local or global smoothness constraints [13], [29], [33], [38], [49].

Structure-from-Motion methods depend on tracking points across image sequences. This approach was initially introduced in [10] to extend to the non-rigid case earlier structure-from-motion work [40]. 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], [37], [47], motion constraints [1], [26], [31], basis constraints [47], or alternate deformation models [16], [30], [41]. 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 [39], planar [44], 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], [23], [28], [48]. 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 [19]. However, despite many generalizations of the original formulation to account for increasingly sophisticated shading effects, such as interreflections [17], [25], specularities [27], shadows [22], 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 using them in conjunction with deformation models [35], [36]. However, this was only demonstrated for the single light source case. By contrast, our method can operate in more general environments, provided only 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 [46], shading information was used to overcome the twofold ambiguity in normal direction that arises from template matching. In [24], 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.

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 [33].

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.

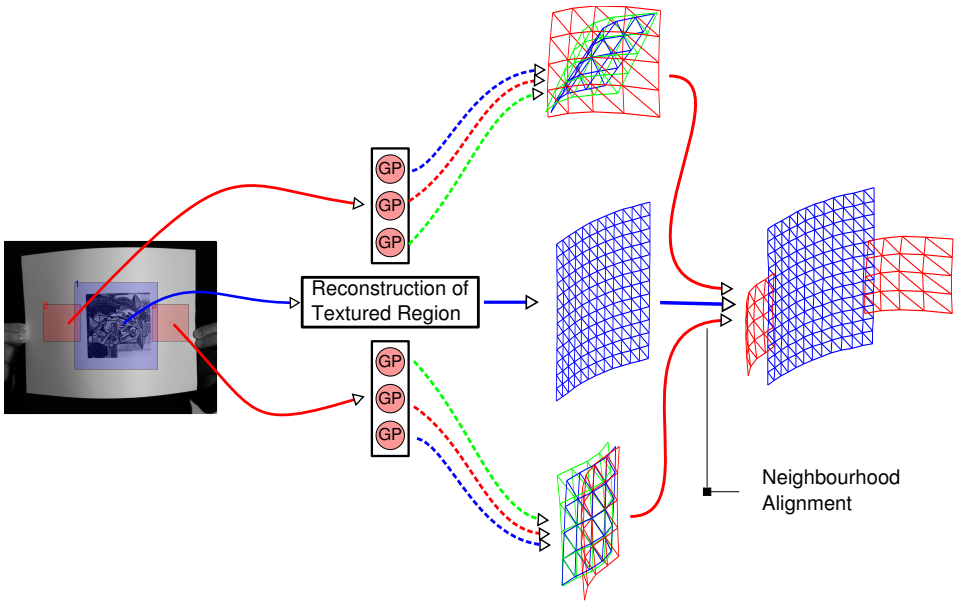


Fig. 2. **Algorithmic flow.** We partition the image into patches, some of which are labeled as textured and others as featureless. We compute the 3D shape of textured patches such as the blue one by establishing point correspondences with a reference image in which the shape is known. We use Gaussian 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 these possible interpretations those that are globally consistent.

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

More specifically, we learn Gaussian Process (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 a Markov Random Field (MRF) that accounts for dependencies between neighboring ones. Finding the maximum a posterior state of the MRF then yields a consistent set of local interpretations.

Although not strictly necessary, textured patches, which can be reconstructed unambiguously, help 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. This being done, we first establish point correspondences with the reference image. Patches that contain enough correspondences are deemed textured and reconstructed using a correspondence-based method we developed in earlier work [33]. We then scan the remaining part of the image, with sliding windows of various sizes and compare each window against the intensity profiles present in the database. Those that are close enough are labeled as featureless patches and the others are simply ignored. Finally, we perform a connected component analysis over the selected patches, and keep the patches that are connected directly or indirectly to the textured ones.

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

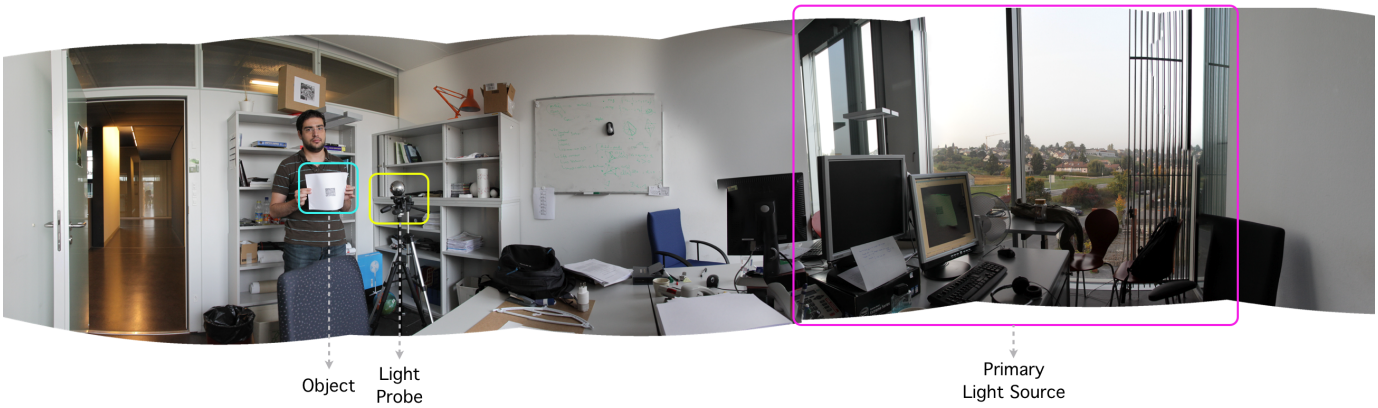


Fig. 3. Panoramic Image of the environment in which we performed our experiments

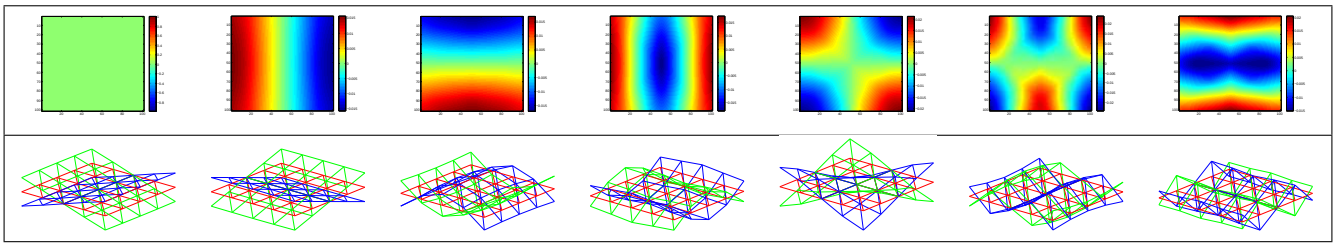


Fig. 4. Intensity and Deformation Modes. **Top Row.** A subset of the low-frequency intensity modes. **Bottom Row.** A subset of the low-frequency deformation modes. The first two encode out-of-plane rotations and the following ones are bending modes.

insensitive to high frequencies in the lighting, for Lambertian objects, we can restrict ourselves to the first nine such coefficients [32]. In practice, this has proved 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 patches, represented by 5×5 grids of 3D points. It was acquired by attaching 3mm wide hemispherical reflective markers to pieces of cloth and paper, which were then waved in front of six infrared Vicon$^{\text{TM}}$ cameras to reconstruct the 3D positions of the markers. For each 3D patch, we use a standard Computer Graphics method [32] 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 an 10201-dimensional space into an 75-dimensional one. It would require a great many samples and be computationally difficult to achieve. Furthermore, high-frequency intensity variations tend to supply relatively little shape information as they 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 $\mathtt{I}_i$ are the intensity modes, the x_i 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 run-time. As a result, the mode weights will remain invariant for similar intensity profiles at different scales.

Similarly, we parametrize 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 $\mathtt{D}_i$ are the deformation modes, the y_i are the modal weights, and N_D is the number of modes. The first few

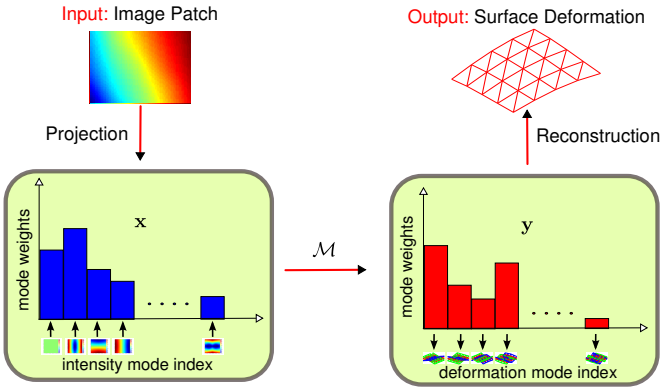


Fig. 5. **Mapping from intensity to surface deformation.** Projecting an intensity patch to the set of orthogonal intensity modes produces a set of intensity modal weights $\mathbf{x}$ that describe its intensity profile. Given a mapping $\mathcal{M}$ from these weights to the deformation modal weights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

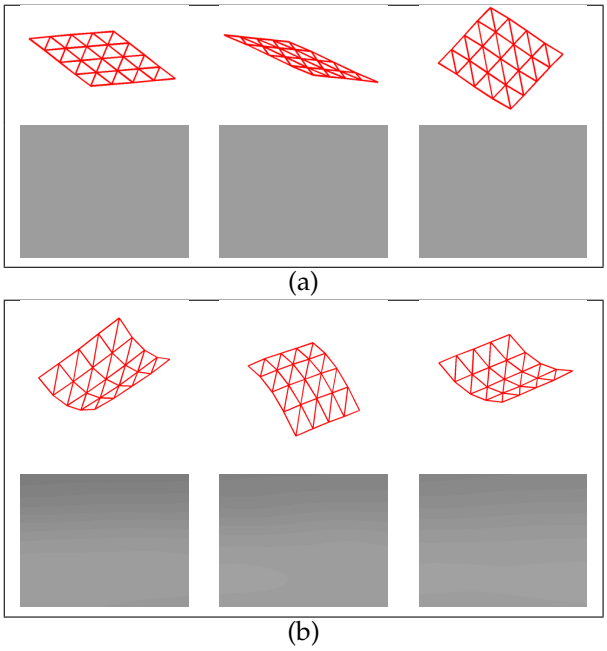


Fig. 6. **Ambiguities for flat (a) and deformed (b) surfaces. First rows** Three different 3D surfaces. **Second rows** Corresponding intensity patches. Even though the 3D shapes are different, their image appearances are almost identical.

deformation modes are depicted by the bottom row of Fig. 4.

After performing this modal decomposition for both the intensity patches and the corresponding 3D surfaces, our database contains for each training sample its intensity modal weights $[x_1, \cdots, x_{N_I}]$ and its deformation modal weights $[y_1, \cdots, y_{N_D}]$.

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

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

that relates intensity weights to deformation weights, as illustrated in 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 Eq. 2.

4.2.1 Gaussian Processes

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

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 input. In our case, we take this function to be the sum of a radial basis function, and a noise term

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

Its shape 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 $p(\mathtt{Y}|\mathbf{x}^1, \cdots, \mathbf{x}^N, \Theta)p(\Theta)$ 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

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

where $\mathbf{k}(\mathbf{x}')$ is the vector of elements $\left[k(\mathbf{x}', \mathbf{x}^1) \cdots k(\mathbf{x}', \mathbf{x}^N)\right]$ [7].

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 [21], 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.

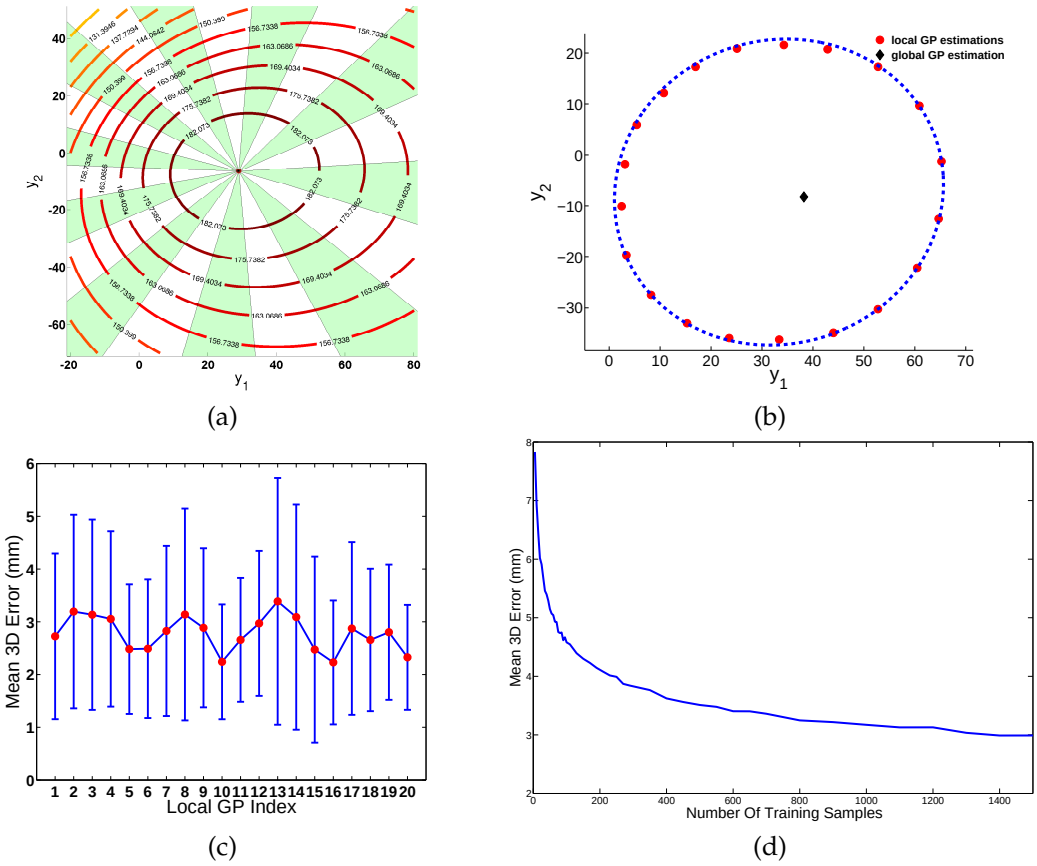


Fig. 7. **Single vs Multiple GPs. (a)** Given a uniform intensity patch, there are infinitely many 3D planar patches that could have generated it. In our scheme, they are parametrized by the y_1 and y_2 weights assigned to the first two deformation modes, which encode out-of-plane rotations. The ovals represent iso-intensity 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 number of training samples. GPs are accurate even when using as few as 1000 samples. In our experiments, for each local GP, we use 1400 samples on average from the training set.

Observing the ambiguous configurations reveals that the ambiguity is particularly severe when the surface patch remains planar and only undergoes rotations. In our scheme, 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. 7(a) we plot the contour curves for the rendered intensities of planar patches in various orientations obtained by densely sampling y_1 and 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 iso-contours in the y_1,y_2 space defines a particular surface normal orientation and, within angular slices, such as those depicted by the alternatively green and white quadrants of Fig. 7(a), 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 [43], 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. 7(b), 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 iso-intensity contour that encodes the family of all orientations producing the

same intensity. In Fig. 7(c), 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 doing the training with 1000 data points and repeated this 100 times. The average reconstruction error of 3 millimeters indicates a good accuracy considering that the average rectangular patch length is 100 millimeters.

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 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. 7(d).

At run time, 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 $\mathcal{S}_p = \{\mathtt{S}_p^1, \cdots, \mathtt{S}_p^{N_{GP}}\}$ of plausible shape interpretations reconstructed up to a scale factor for each patch p, and a single one $\mathtt{S}_{p'}$ 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 a Markov Random Field 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 of an MRF graph. Featureless ones can be assigned one of the N_{GP} labels corresponding to the elements of $\mathcal{S}_p$, while textured ones are assigned their unambiguously recovered shape label.

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

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

where $\mathcal{O}(q)$ is the set of patches overlapping p. The uniary 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. 8(a). 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 uniary and pairwise terms of Eq. 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 re-weighted message passing technique [20] 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 non sub-modular 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

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

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 lies in the overlap area as shown in Fig. 8(b). 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

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

Each sample yields one linear equation of the form of Eq. (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

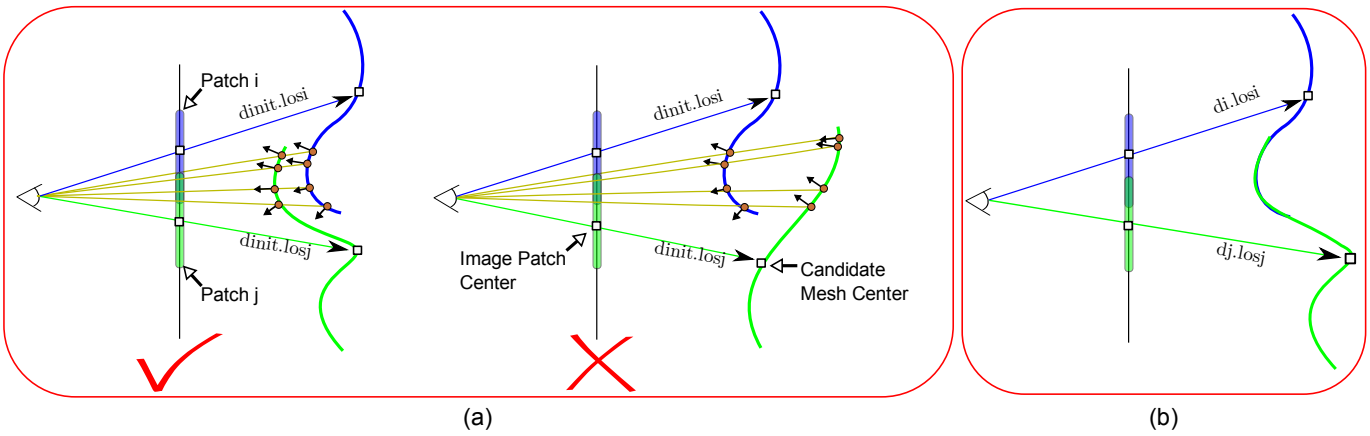


Fig. 8. **Enforcing Shape Consistency (a)** Two different instances of the evaluation of the geometric consistency of patches i and j, shown in blue and green respectively. In both cases, the predicted normals of points along the same lines of sight, drawn in yellow, are compared. Since these points 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.

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 28000 surface patches, or approximately 1400 per GP. They are represented as 5×5 meshes and rendered using the calibrated experiment-specific lighting environment.

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 non-textured parts, we compare our results to those obtained using our earlier technique [33] that relies solely on point correspondences to demonstrate that also using the shape-from-shading information does indeed help. We also compare against pure shape-from-shading algorithms described in [42], [11] 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 in 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 using our earlier texture-based method [33].

We compute 3D reconstruction errors as the mean point-to-surface distances from the reconstructed point clouds to the ground-truth surfaces. The results are shown in the bottom row of Fig. 9. By combining shading and texture clues, our method performs significantly better except for flat surfaces, where both methods return similar results.

In Fig. 10, we compare our results against those of pure shape-from-shading methods [11], [14], [42]. 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.

6.2 Real Images

We applied our reconstruction algorithm on two real sequences of a deforming piece of paper and of a t-shirt, as shown in the top row of Fig. 11 and Fig. 12, respectively.

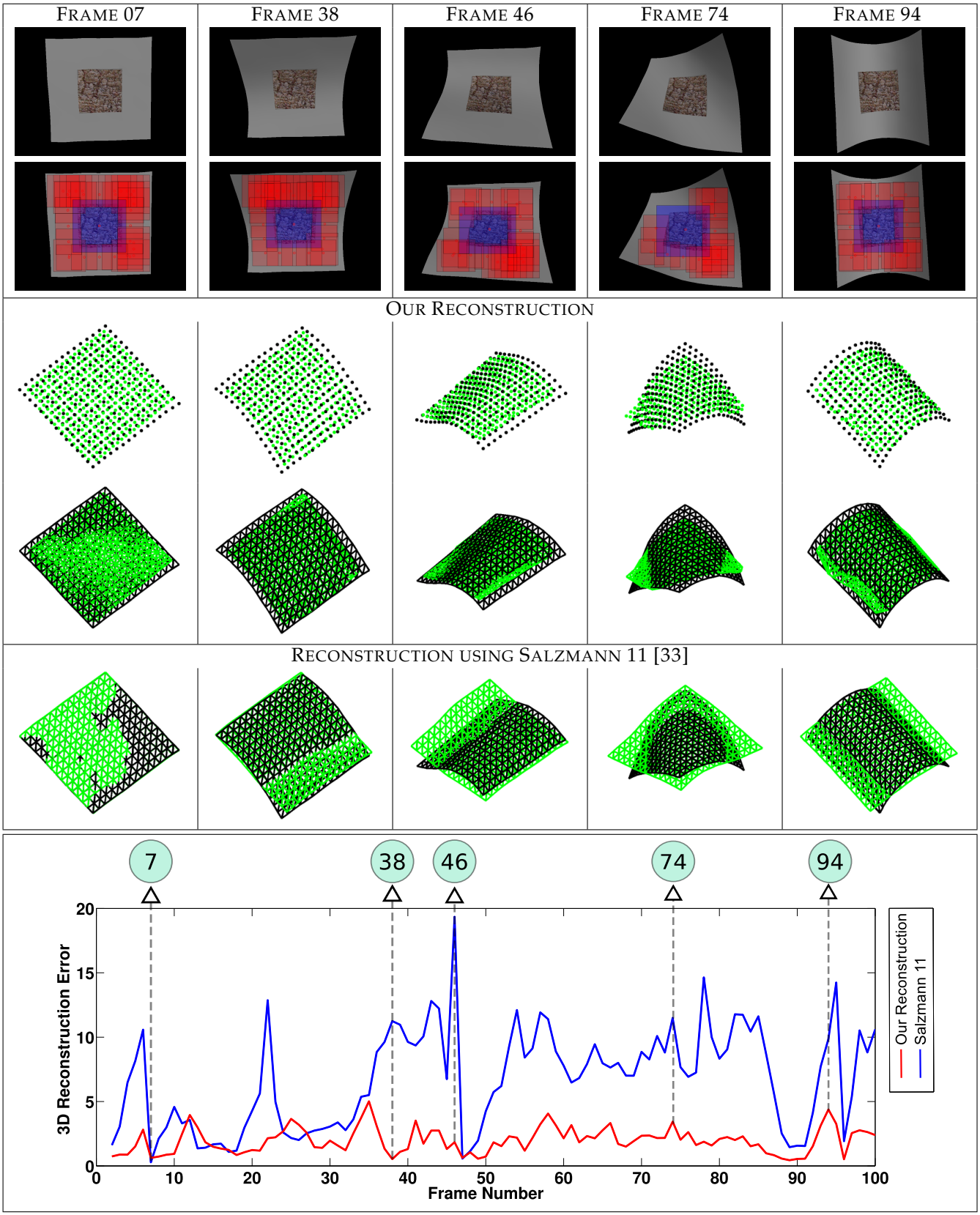


Fig. 9. **Synthetic Sequence. First row** Input images. **Second row** Local Patches. The blue patches indicate the regions where feature correspondences are given, and the red patches are non-textured areas, selected by our patch selection algorithm. **Third row** Reconstructed point cloud (green dots) and ground-truth mesh vertices (black dots) seen from another view point. **Forth row** Estimated triangulation (green) and ground-truth triangulation (black) seen from another view point. **Fifth row** Reconstruction results using the method in [33] (green mesh) and ground-truth triangulation (black). **Bottom row** Reconstruction error of both methods for the first 100 frames of the sequence. Note that the proposed method provides much better reconstructions, except for 6 frames in the sequence.

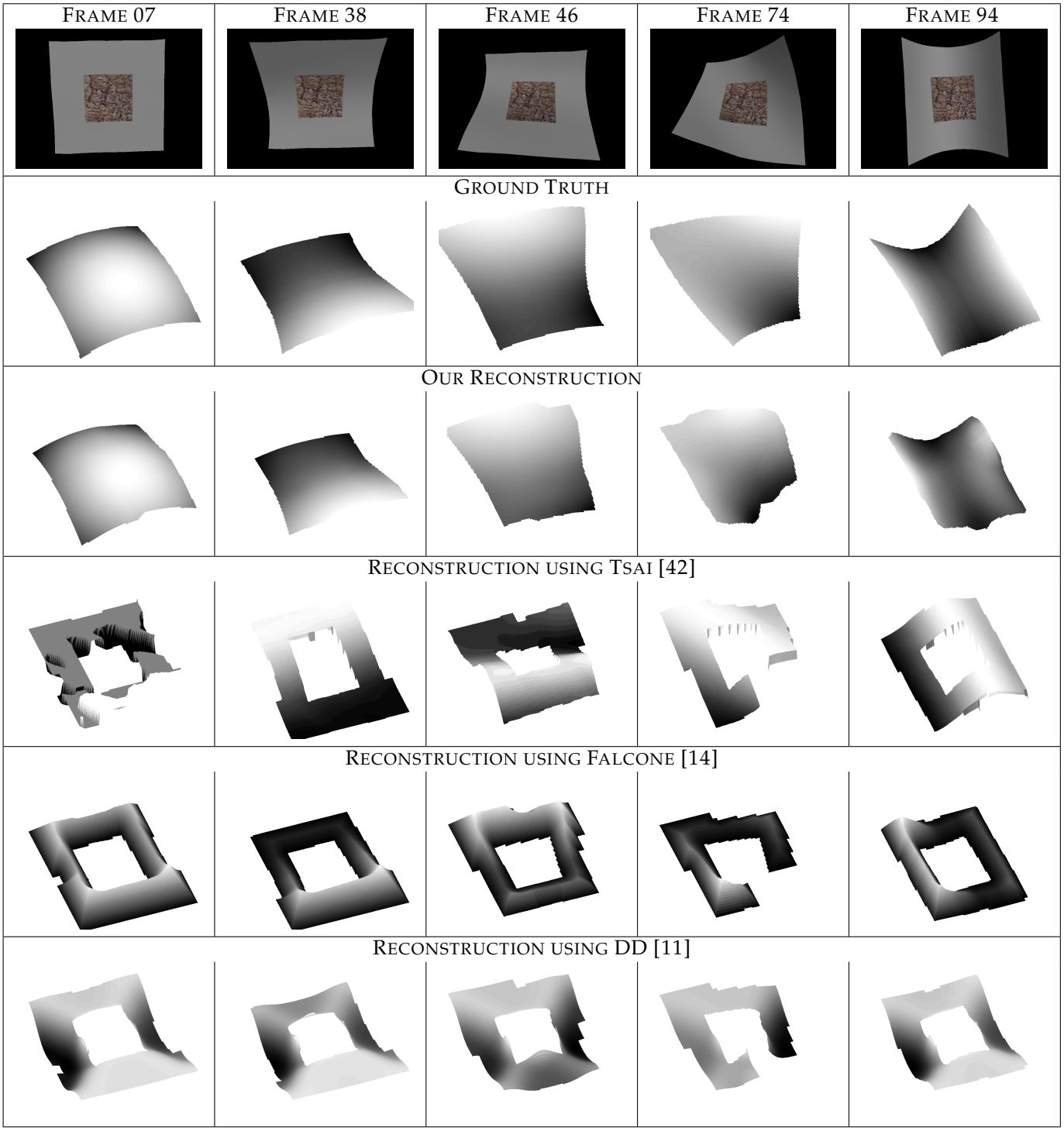


Fig. 10. **Synthetic Sequence. First row:** Input images. **Second row:** Ground truth depth maps **Third row:** Depth maps computed from our reconstructions. **Forth to sixth rows:** Depth maps computed by the methods in [42], [14] and [11], respectively.

These 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 progressively scanning the input image with different patch sizes starting from the largest one. 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 Figures 11 and 12.

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. 11 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 [45]. 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. 13. 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. 13, 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 the shading clues produces much more accurate results than those of these other methods that only rely on one or the other.

7 Conclusion

We have presented an approach to monocular shape recovery that effectively takes advantage of both shape-from-shading cues in non-textured 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, more complex representations of the lighting environment than spherical harmonics could be used to create our training set. Similarly, other, potentially nonlinear parametrization of the patch intensities and deformations could replace the current PCA mode weights.

The main limitation of our current technique is that it requires calibration of the lighting environment prior to the computation. Future work will therefore focus on removing this limitation as well as introducing the constraints imposed by boundary locations into our framework.

References

[1] H. Aanaes and F. Kahl. Estimation of Deformable Structure and Motion. In *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. In *Conference on 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. In *Conference on Computer Vision and Pattern Recognition*, June 2009.

[4] P. Belhumeur, D. Kriegman, and A. Yuille. The Bas-Relief Ambiguity. *International Journal of Computer Vision*, 35(1):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 Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 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*, 9(2), 2006.

[9] M. Brand. A Direct Method of 3D Factorization of Nonrigid Motion Observed in 2D. In *Conference on Computer Vision and Pattern Recognition*, pages 122–128, 2005.

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

[11] P. Daniel and J. D. Durou. From Deterministic to Stochastic Methods for Shape from Shading. In *Asian Conference on Computer Vision*, 2000.

[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*, 2008.

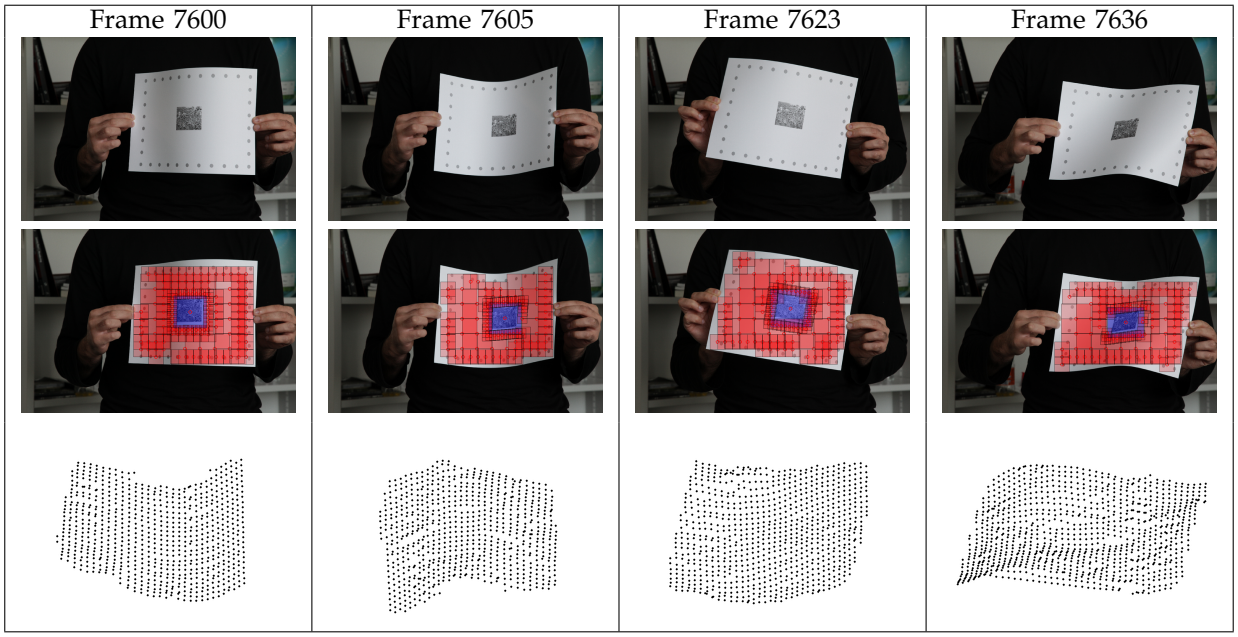
[13] A. Ecker, A.D. Jepson, and K.N. Kutulakos. Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities. In *European Conference on Computer Vision*, October 2008.

[14] M. Falcone and M. Sagona. An algorithm for global solution of the Shape-from-Shading model. In *International Conference on Image Analysis and Processing*, 1997.

[15] J. Fayad, L. Agapito, and A. Del Bue. Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences. In *European Conference on 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. In *British Machine Vision Conference*, 2009.

[17] D.A. Forsyth and A. Zisserman. Reflections on Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):671–679, July 1991.



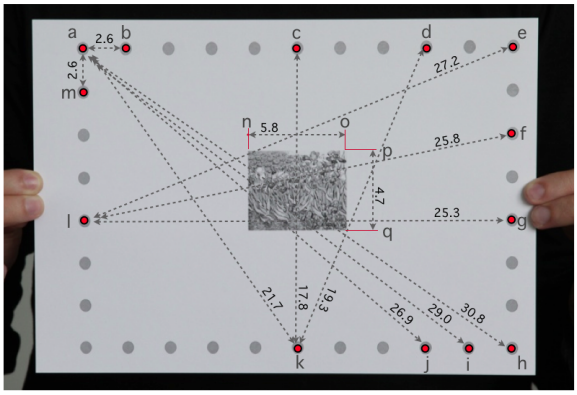
Reference Image	Point Pairs	Ground Truth	Frame Number 7600	7605	7623	7636
	a – b	2.6	2.5	2.6	2.6	3.2
	a – h	30.8	30.6	29.4	31.0	32.6
	a – i	28.9	28.2	27.9	28.9	30.1
	a – j	26.9	26.1	26.1	27.1	27.6
	a – k	21.7	21.2	22.6	22.5	22.1
	a – m	2.6	2.8	2.8	2.9	2.9
	c – k	17.8	18.0	18.0	18.4	18.9
	d – k	19.3	20.2	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. 11. **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 view point. **Fourth row** Geodesic distances between prominent landmarks as identified on the left. Point d in frame 7636 was outside our reconstruction, which explains the missing value in the table.

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

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

[20] V. Kolmogorov. Convergent Tree-reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[21] R. Kozera. Uniqueness in Shape from Shading Revisited. *Journal of Mathematical Imaging and Vision*, 7(2):123–138, 1997.

[22] D.J. Kriegman and P.N. Belhumeur. What Shadows Reveal About Object Structure. In *European Conference on Computer Vision*, pages 399–414, 1998.

[23] J. Liang, D. Dementhon, and D. Doermann. Flattening Curved Documents in Images. In *Conference on Computer Vision and Pattern Recognition*, pages 338–345, 2005.

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

[25] S.K. Nayar, K. Ikeuchi, and T. Kanade. Shape from Interreflections. *International Journal of Computer Vision*, 6(3):173–195, 1991.

[26] S.I. Olsen and A. Bartoli. Implicit Non-Rigid Structure-from-Motion With Priors. *Journal of Mathematical Imaging and Vision*, 31:233–244, 2008.

[27] M. Oren and S.K. Nayar. A Theory of Specular Surface Geometry. *International Journal of Computer Vision*, 24(2):105–124, 1996.

[28] M. Perriollat and A. Bartoli. A Quasi-Minimal Model for Paper-Like Surfaces. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[29] M. Perriollat, R. Hartley, and A. Bartoli. Monocular Template-Based Reconstruction of Inextensible Surfaces. In *British Machine Vision Conference*, 2008.

[30] V. Rabaud and S. Belongie. Re-Thinking Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, June 2008.

[31] V. Rabaud and S. Belongie. Linear Embeddings in Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, June 2009.

[32] R. Ramamoorthi and P. Hanrahan. An Efficient Representation

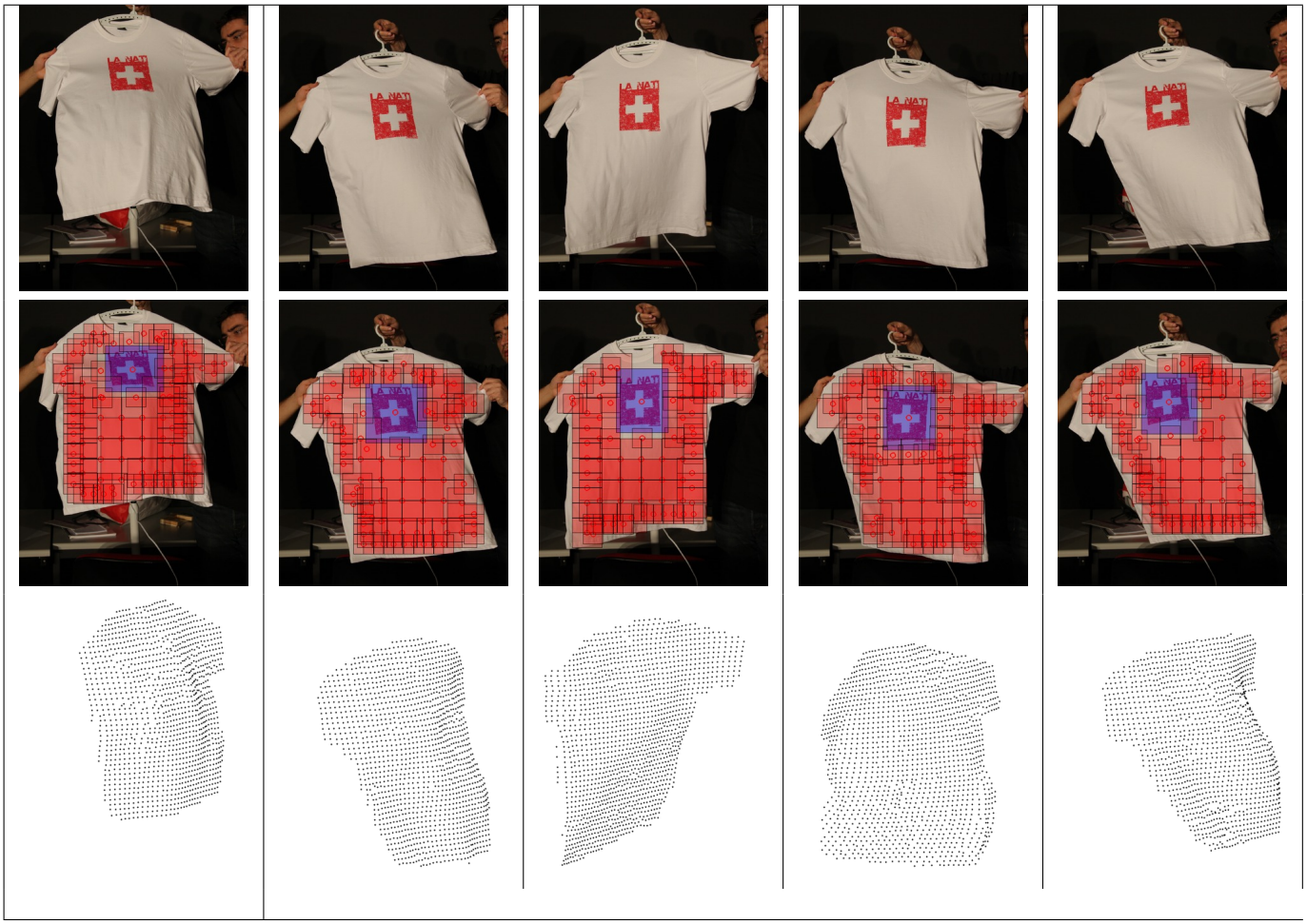


Fig. 12. **T-Shirt Sequence. First row** Input images. **Second row** Local Patches. The blue patch is the one for which enough correspondences are found and the red ones are featureless patches. **Third row** Reconstructed point cloud seen from another view point.

for Irradiance Environment Maps. In *ACM SIGGRAPH*, 2001.

[33] M. Salzmann and P. Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[34] M. Salzmann, V. Lepetit, and P. Fua. Deformable Surface Tracking Ambiguities. In *Conference on Computer Vision and Pattern Recognition*, June 2007.

[35] D. Samaras and D. Metaxas. Incorporating Illumination Constraints in Deformable Models. In *Conference on Computer Vision and Pattern Recognition*, pages 322–329, June 1998.

[36] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc. Variable Albedo Surface Reconstruction from Stereo and Shape from Shading. In *Conference on Computer Vision and Pattern Recognition*, June 2000.

[37] A. Shaji and S. Chandran. Riemannian Manifold Optimisation for Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[38] S. Shen, W. Shi, and Y. Liu. Monocular 3D Tracking of Inextensible Deformable Surfaces Under L2-Norm. In *Asian Conference on Computer Vision*, 2009.

[39] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-Rigid Structure from Locally-Rigid Motion. In *Conference on Computer Vision and Pattern Recognition*, June 2010.

[40] C. Tomasi and T. Kanade. Shape and Motion from Image Streams Under Orthography: A Factorization Method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[41] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid Structure-From-Motion: Estimating Shape and Motion With Hierarchical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

[42] P. S. Tsai and M. Shah. Shape from Shading using Linear Approximation. *Journal of Image and Vision Computing*, pages 69–82, 1994.

[43] R. Urtasun and T. Darrell. Sparse Probabilistic Regression for Activity-Independent Human Pose Inference. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[44] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-Free Monocular Reconstruction of Deformable Surfaces. In *International Conference on Computer Vision*, September 2009.

[45] T. Weise, B. Leibe, and L. Van Gool. Fast 3D Scanning with Automatic Motion Compensation. In *CVPR*, June 2007.

[46] R. White and D.A. Forsyth. Combining Cues: Shape from Shading and Texture. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[47] J Xiao, J.-X. Chai, and T. Kanade. A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. In *European Conference on Computer Vision*, pages 573–587, 2004.

[48] Z. Zhang, C. Tan, and L. Fan. Restoration of Curved Document Images Through 3D Shape Modeling. In *Conference on Computer Vision and Pattern Recognition*, June 2004.

[49] J. Zhu, S. Hoi, C. Steven, Z. Xu, and M.R. Lyu. An Effective Approach to 3D Deformable Surface Tracking. In *European Conference on Computer Vision*, pages 766–779, 2008.

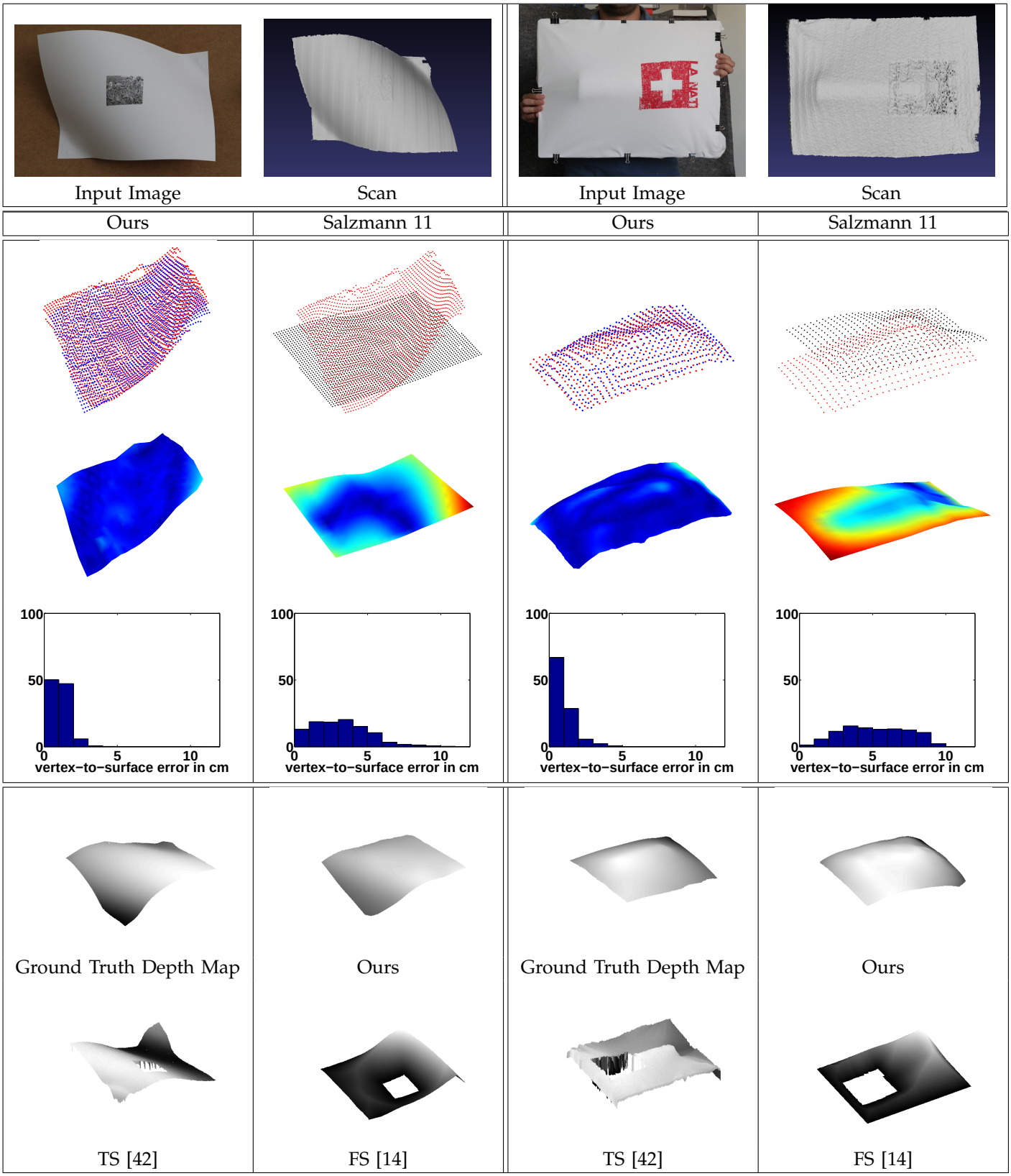


Fig. 13. Accuracy estimation using structured light scans. **Top Row.** Two different surfaces and their corresponding structured light scans. **Middle Row.** From top to bottom, point clouds from the scans (red) and reconstructions by either our algorithm or that of [33] (green), reconstructed 3D surface 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 Row.** Depth maps obtained from our reconstructions and from the methods in [14], [42].