Beyond Feature Points: Structured Prediction for Monocular Non-rigid 3D Reconstruction

Mathieu Salzmann   Raquel Urtasun

NICTA   TTI Chicago

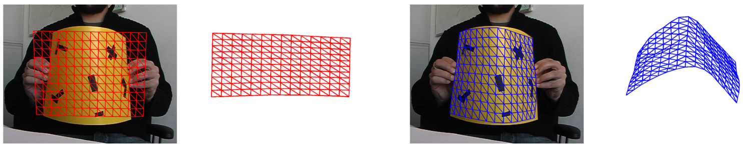
Abstract. Existing approaches to non-rigid 3D reconstruction either are specifically designed for feature point correspondences, or require a good shape initialization to exploit more complex image likelihoods. In this paper, we formulate reconstruction as inference in a graphical model, where the variables encode the rotations and translations of the facets of a surface mesh. This lets us exploit complex likelihoods even in the absence of a good initialization. In contrast to existing approaches that set the weights of the likelihood terms manually, our formulation allows us to learn them from as few as a single training example. To improve efficiency, we combine our structured prediction formalism with a gradient-based scheme. Our experiments show that our approach yields tremendous improvement over state-of-the-art gradient-based methods.

1 Introduction

Monocular non-rigid surface reconstruction has received increasing attention in recent years. Existing approaches to tackling this problem can be classified into (i) non-rigid structure-from-motion techniques [4, 27, 8] that exploit the availability of multiple images of different deformations to reconstruct both 3D points and camera motion, and (ii) template-based methods [23, 18, 5] that rely on a reference image with known 3D shape to perform reconstruction from a single additional image of the deformed surface. In most cases, the aforementioned methods are specifically designed to handle feature point correspondences, and as a consequence, cannot make use of richer image information, such as full surface texture, or surface boundaries. More importantly, these methods become unsuitable when too few feature points can be reliably detected and matched.

Several attempts have been proposed to leverage more complex image likelihoods [20, 21]. However, the resulting methods rely on gradient-based optimization schemes that can easily get trapped in the many local maxima of these complex, non-smooth likelihoods. As a consequence, these methods have only been used either for frame-to-frame tracking, where the previous frame provides a good initialization [20], or when large amounts of training data are available to learn a discriminative predictor that produces a good initialization [21].

In contrast, in this paper we propose to employ a global optimization framework to exploit complex image likelihoods for monocular non-rigid reconstruction. As our optimization is more global than gradient-based methods, it is also more robust to local maxima, thus yielding accurate reconstructions even in the absence of a good initialization, as illustrated in Fig. 1. More specifically, we



Gradient-based method

Our approach

Fig. 1. Reconstructing a piece of cardboard from a single input image. (Left) Reconstruction obtained with a gradient-based method. (Right) Our reconstruction.

represent a surface as a triangulated mesh and formulate the 3D reconstruction problem as inference in a conditional Markov random field (CRF), where the variables to recover are the rotations and translations of the individual mesh facets. To handle such continuous variables, we adopt particle convex belief propagation [16] as our inference algorithm: We iteratively draw random samples around the current solution for each variable, compute the MAP estimate of the discrete CRF defined by these samples using convex belief propagation [12], and update the current solution with this MAP estimate. This strategy lets us effectively explore the 3D shape space even when no good initialization is provided. Furthermore, given very few training pairs of images and 3D shapes, we employ a structured prediction learning algorithm [11] to find the weights of the individual terms in the likelihood, thus avoiding having to set them manually as is traditionally done in 3D reconstruction algorithms (e.g., [20, 18, 21, 5]).

To reduce the computational burden of performing global optimization on large graphs (i.e., fine meshes), we introduce a coarse-to-fine scheme that combines the advantages of global optimization and gradient-based approaches. Our strategy consists in first performing structured prediction with a coarse mesh, and then using the coarse solution as initialization to a gradient-based method. Since our coarse structured prediction yields a good initial shape estimate, this strategy has proven very effective in practice. We demonstrate the benefits of our approach in a variety of scenarios ranging from well-textured surfaces to very poorly-textured ones. Comparison against gradient-based techniques clearly shows that our approach is much better adapted to 3D reconstruction from a single image than state-of-the-art methods.

2 Related Work

Monocular non-rigid 3D shape recovery is a very challenging problem with many ambiguities due to noisy measurements, as well as to the wide range of deformations that objects may undergo. Throughout the years, approaches to tackling this problem have evolved, starting from the early methods that attempted to model the physical behavior of deformable surfaces [13, 17, 14, 15], to the more recent ones that tried to learn this behavior from data [6, 3].

In recent years, two main trends have emerged for non-rigid 3D shape recovery: Non-rigid structure-from-motion (NRSfM) and template-based reconstruction. NRSfM techniques [4, 27, 25, 1, 8] work under the assumption that multiple images of the surface undergoing different deformations are available. These methods try to recover the 3D locations of feature points, as well as the camera motion. As in our approach, [24] also reconstructs individual triangles, but in the NRSfM setting. [19] utilizes discrete optimization for NRSfM. However, their discrete problem is not directly for reconstruction purposes, but only to assign feature points to local patches. As opposed to NRSfM, template-based approaches [23, 18, 5] work with a single input image, but assume that the camera is calibrated and that a reference image with known surface shape is available. A successful shape prior in these methods is to encourage the surface to deform isometrically. Our work falls into the template-based category and exploits a similar isometry prior. However, whereas all the above-mentioned methods rely on feature points, our approach lets us exploit much richer image information.

Techniques that employ different sources of information, such as shading [26] or contours [10], have been developed. However, contour-based approaches are only applicable to a specific class of surfaces, and shape-from-shading methods make strong assumptions on the lighting conditions. More directly related to our approach are the methods of [20, 21], where general image losses were also employed. However, due to the non-convexity of such losses and the use of a gradient-based method, [20] was only applied in a frame-to-frame tracking scenario. Furthermore, both techniques heavily rely on the availability of relatively large amounts of training data to learn either a deformation model [20], or a discriminative predictor to initialize a gradient-based method [21]. While we also exploit training data to learn the weights of the different terms in our likelihood, we require much fewer training examples. Furthermore, we utilize a global optimization method, which lets us reconstruct surfaces from individual images.

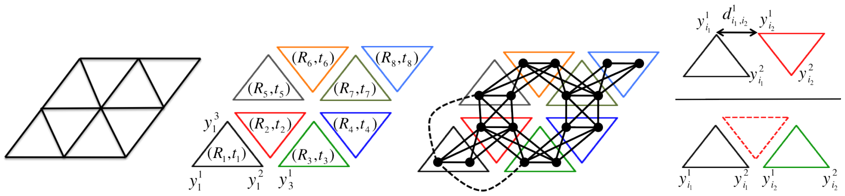
3 Structured Prediction for Non-rigid Surfaces

In this section, we introduce our surface parametrization and then present our structured prediction approach to non-rigid 3D reconstruction. Finally, we describe the gradient-based method used to refine the structured prediction results.

3.1 Surface Parametrization

We represent non-rigid surfaces as triangulated meshes, and, following a popular and effective trend [23, 18, 5], encourage the surface to deform isometrically by preserving the distances between neighboring mesh vertices. Furthermore, as our method falls into the template-based category, we assume that we are given a reference image in which the 3D shape of the surface is known.

Since the mesh already forms a graph, it might seem natural to use the 3D vertex positions as variables. However, some image likelihoods, such as template matching, are defined over a facet. Therefore, employing a parametrization in terms of 3D vertices will require 3-way potentials, i.e., terms that involve three variables. As the complexity of message passing inference in graphical models is a function of the order of the potentials, as well as of the cardinality of the label set for each random variable, employing 3-way potentials is computationally prohibitive, thus making this parametrization unappealing. Instead, as illustrated in Fig. 2(a,b), we parametrize the surface in terms of the rotations and translations of the mesh facets, which, as shown below, only requires pairwise potentials.



(a) (b) (c) (d)

Fig. 2. Structured prediction with a mesh. (a) Triangulated mesh. (b) Parametrization in terms of facet rotations and translations. (c) Graphical model. Note that, to avoid clutter, only two longer range (dashed) edges are shown. (d) Illustration of the facet coherence potential (top) and the smoothness potential (bottom).

More specifically, the 3D location of the k^{th} vertex of facet i is given by

$$\mathbf{y}_i^k = \mathbf{R}_i(\tilde{\mathbf{y}}_i^k - \tilde{\mathbf{c}}_i) + \mathbf{t}_i \triangleq \mathbf{R}_i\bar{\mathbf{y}}_i^k + \mathbf{t}_i \ , \tag{1}$$

where $\mathbf{R}_i$ and $\mathbf{t}_i$ are the rotation matrix and translation of the facet, respectively, $\tilde{\mathbf{y}}_i^k$ is the location of the k^{th}vertex of facet i in the reference mesh, and $\tilde{\mathbf{c}}_i$ is the centroid of facet i in the reference mesh. We represent the rotation $\mathbf{R}_i$ in terms of a 3D vector of Euler angles $\boldsymbol{\theta}_i$. Note that other parameterizations, such as quaternions are also possible. The location of a 3D mesh vertex can then be obtained by averaging the above locations over all the facets that contain this vertex. Of course, this requires preventing the rotations and translations of these facets from disagreeing over the location of the shared vertex. As will be shown in the next section, this can be expressed as a pairwise potential.

3.2 Non-rigid 3D Reconstruction as Inference in a Graphical Model

Given our parametrization in terms of facet rotations and translations, we now describe our approach to non-rigid 3D reconstruction. We formulate monocular shape recovery as an inference problem in a CRF, where the random variables are continuous. The joint distribution over the random variables can be factorized into a product of non-negative potentials

$$p(\mathbf{z}) = p(\mathbf{R}, \mathbf{t}) = Z^{-1} \prod\nolimits_i \psi_i(\mathbf{z}_i) \prod\nolimits_\alpha \psi_\alpha(\mathbf{z}_\alpha) \ , \tag{2}$$

where $\mathbf{z} = (\mathbf{R}, \mathbf{t})$ is the set of all random variables, with $\mathbf{R}$ and $\mathbf{t}$ containing the rotations and translations for all facets, and Z is the partition function. The potentials $\psi_i(\mathbf{z}_i)$ and $\psi_\alpha(\mathbf{z}_\alpha)$ encode functions over single variables and groups of variables, respectively. Inference is performed by computing the MAP estimate

$$\mathbf{z}^* = \text{argmax}_{\mathbf{z}} \prod\nolimits_i \psi_i(\mathbf{z}_i) \prod\nolimits_\alpha \psi_\alpha(\mathbf{z}_\alpha) \ . \tag{3}$$

To solve our inference problem over continuous variables, we rely on particle convex belief propagation (PCBP) [16]. PCBP is an iterative algorithm that works as follows: Particles are sampled around the current solution for each random variable. These samples act as labels in a discrete CRF which is solved to convergence using convex belief propagation [12]. The current solution is then updated with the MAP estimate returned by convex BP. This process is repeated for a fixed number of iterations. In practice, we use the distributed message passing algorithm of [22] to solve the discrete CRF at each iteration.

Algorithm 1 depicts PCBP for our formulation of non-rigid 3D reconstruction. In the algorithm, we denote by $\hat{\mathbf{R}}_i$ and $\hat{\mathbf{t}}_i$ the discretized variables, which are grouped in the set $\hat{\mathbf{z}}$. At each iteration, to increase the accuracy of the reconstruction, we decrease the values of the standard deviations σ_r and σ_t of the Gaussian distributions from which the discretized random variables are drawn.

Algorithm 1 PCBP for non-rigid 3D reconstruction

```
Set $N$, $\eta_r$ and $\eta_t$
Initialize $\mathbf{R}_i$ and $\mathbf{t}_i$ from the template mesh $\forall i$, as well as $\sigma_r$ and $\sigma_t$
for $s = 1$ to #iters do
    Draw $N$ random samples of Euler angles $\boldsymbol{\theta}_i^j \propto \mathcal{N}(0, \sigma_r)$, $\forall i$
    Compute the candidate discretized rotations $\hat{\mathbf{R}}_i^j = \mathbf{R}_i^j(\boldsymbol{\theta}_i^j)\mathbf{R}_i$, $\forall i, j$
    Draw $N$ random samples of candidate discretized translations $\hat{\mathbf{t}}_i^j \propto \mathcal{N}(\mathbf{t}_i, \sigma_t)$, $\forall i$
    Solve the discrete CRF: $(\hat{\mathbf{R}}^*, \hat{\mathbf{t}}^*) = \mathrm{argmax}_{\hat{\mathbf{z}}} \prod_i \psi_i(\hat{\mathbf{z}}_i) \prod_\alpha \psi_\alpha(\hat{\mathbf{z}}_\alpha)$
    Update $\mathbf{R}_i \leftarrow \hat{\mathbf{R}}_i^*$ and $\mathbf{t}_i \leftarrow \hat{\mathbf{t}}_i^*$, $\forall i$
    Update $\sigma_r \leftarrow \eta_r \sigma_r$ and $\sigma_t \leftarrow \eta_t \sigma_t$
end for
```

An artifact of using discretized variables with non-smooth potentials is that a solution around a local maximum might have a higher value than one around the global maximum. The iterative scheme will then re-sample around this relatively bad solution and, with decreasing σ_r and σ_t, potentially be driven away from the global maximum. To circumvent this issue, we introduce a scheme that keeps track of multiple solutions at each iteration of PCBP. Given all the discrete candidates for all the variables, we find an approximate MAP solution using convex BP. We then remove the labels corresponding to this solution and find an approximate solution to the MAP problem defined by the remaining labels. This can be done in an iterative manner, thus yielding M solutions around which we can then sample N/M values for the next iteration of PCBP. Note that even if we could solve the NP-hard discrete inference problem exactly, these solutions would not necessarily truly be the M best ones, since combinations of their labels are not considered as potential solutions (e.g., the second solution cannot contain labels used in the first solution). However, this allows for more variety in the candidate solutions, and the labels can potentially be combined at the next PCBP iteration. Note that other algorithms, such as [9, 2], could also be used to generate candidate solutions. In the last iteration of PCBP, we only compute a single MAP estimate, which we take as our final reconstruction.

In the remainder of this section, we describe the different potentials that we used in our experiments. In particular, we define three types of image potentials to handle feature point correspondences, template matching and surface boundary likelihoods. These likelihoods are the ones typically used in gradient-based methods [20, 21]. Additionally we employ two types of shape potentials encoding coherence of the facets and surface smoothness. Taken together, these potentials yield a graph such as the one depicted by Fig. 2(c). For clarity, we describe the potentials in the log domain, i.e., $w^T\phi = log(\psi)$. We define a weight for each type of potential, and as described later, learn the weights using [11].

Feature Point Correspondences*:* Although our main focus is to go beyond feature point correspondences, we show that our formulation also remains pairwise in this case. We make use of the template mesh to establish correspondences between a 3D point expressed in barycentric coordinates with respect to the facet it lies on and a 2D point in the input image. In the camera referential, the fact that a 3D point j on facet i reprojects at image location (u^j, v^j) can be written as

$$\mathbf{A}\sum_{k=1}^{3} b_j^k \mathbf{y}_i^k = \mathbf{A}\sum_{k=1}^{3} b_j^k \left(\mathbf{R}_i \bar{\mathbf{y}}_i^k + \mathbf{t}_i\right) = d^j \left(u^j\ v^j\ 1\right)^T \ , \tag{4}$$

where b_j^k is the barycentric coordinate of point j with respect to the k^{th}vertex $\mathbf{y}_i^k$ of facet i to which the point belongs, $\mathbf{A}$ is the matrix of known internal camera parameters, and d^j is an unknown scalar encoding depth.

We define pairwise potentials $\phi^r_{\alpha_i}(\mathbf{R}_i, \mathbf{t}_i)$ by summing the negative reprojection errors of each detected feature point belonging to one particular facet. To this end, let us define the projection of point j on facet i as

$$\hat{u}^j(\mathbf{R}_i, \mathbf{t}_i) = \frac{\mathbf{A}_1 \sum_{k=1}^{3} b_j^k \left(\mathbf{R}_i \bar{\mathbf{y}}_i^k + \mathbf{t}_i\right)}{\mathbf{A}_3 \sum_{k=1}^{3} b_j^k \left(\mathbf{R}_i \bar{\mathbf{y}}_i^k + \mathbf{t}_i\right)} \ , \quad \hat{v}^j(\mathbf{R}_i, \mathbf{t}_i) = \frac{\mathbf{A}_2 \sum_{k=1}^{3} b_j^k \left(\mathbf{R}_i \bar{\mathbf{y}}_i^k + \mathbf{t}_i\right)}{\mathbf{A}_3 \sum_{k=1}^{3} b_j^k \left(\mathbf{R}_i \bar{\mathbf{y}}_i^k + \mathbf{t}_i\right)} \ , \tag{5}$$

where $\mathbf{A}_k$ is the k^{th} row of $\mathbf{A}$. The potential for facet i can then be written as

$$\phi^r_{\alpha_i}(\mathbf{R}_i, \mathbf{t}_i) = -\sum_{j \in \mathcal{F}(i)} \left\| \left(\hat{u}^j(\mathbf{R}_i, \mathbf{t}_i) - u^j \ , \ \hat{v}^j(\mathbf{R}_i, \mathbf{t}_i) - v^j\right) \right\|_2^2 \ , \tag{6}$$

where $\mathcal{F}(i)$ is the set of feature points belonging to facet i. This potential is pairwise, as it is a function of the rotation and translation of a single facet.

Template Matching*:* For template matching, each facet in the reference mesh is treated as a template. We compute the normalized cross-correlation between the texture under the facet in the reference image and the texture under the deformed facet in the input image. This can be done by sampling the barycentric coordinates of the facet and retrieving the intensity values at the 2D image locations corresponding to the projected sampled 3D facet points. In our formalism, the intensity values for facet i can be stored in a vector $\mathbf{q}_i$, such that each element j is given by $\mathbf{q}_i^j = I\left(\hat{u}^j(\mathbf{R}_i, \mathbf{t}_i), \hat{v}^j(\mathbf{R}_i, \mathbf{t}_i)\right)$, where $I(u, v)$ is the intensity value at image location (u, v), and $(\hat{u}^j, \hat{v}^j)$ are the projections of the points at the sampled barycentric coordinates.

Let $\hat{\mathbf{q}}_i$ and $\tilde{\mathbf{q}}_i$ be the mean subtracted vectors of intensity values in the input image and in the reference image, respectively. A template matching potential for facet i can then be written as

$$\phi^t_{\alpha_i}(\mathbf{R}_i, \mathbf{t}_i) = \left(\hat{\mathbf{q}}_i^T \tilde{\mathbf{q}}_i\right) \left(\sum\nolimits_j \left(\hat{\mathbf{q}}_i^j\right)^2 \sum\nolimits_j \left(\tilde{\mathbf{q}}_i^j\right)^2 \right)^{-1/2} \ . \tag{7}$$

As for correspondences, this potential only depends on the rotation and translation of a single facet, and is therefore pairwise. Note that this potential truly depends on the locations of 3 vertices. Therefore, had we used vertex locations to parametrize our problem instead of facet rotations and translations, we would not be able to decompose this term in a sum of unary and pairwise potentials.

Surface Boundary*:* To account for object boundaries, we make use of the distance transform D of the edge image obtained from the input image with Canny's algorithm. D encodes the distance of each pixel to the closest edge, which has the advantage of being smoother than the edge image itself. We sample the barycentric coordinates of the boundary mesh edges, and project the resulting 3D points in D. Given the barycentric coordinates b^k_j of points sampled on an edge belonging to facet i, we can then write the edge potential

$$\phi^e_{\alpha_i}(\mathbf{R}_i, \mathbf{t}_i) = -D\big(\hat{u}^j(\mathbf{R}_i, \mathbf{t}_i), \hat{v}^j(\mathbf{R}_i, \mathbf{T}_i)\big) \ , \tag{8}$$

where $(\hat{u}^j, \hat{v}^j)$ are the projected sampled barycentric coordinates, which now only depend on the 2 vertices that define the mesh edge (i.e k ranges up to 2 in Eq. 5). Once again, this potential depends on a single facet, and is thus pairwise.

Facets Coherence*:* As mentioned in Section 3.1, optimizing the rotations and translations of the mesh facets independently may lead to disagreements over the location of the vertices shared by neighboring facets. As a consequence, 3D vertices belonging to multiple facets, computed by averaging the locations predicted by the facets, will be distant from the individual predictions. To prevent this, we include a potential that encourages facets sharing an edge to agree on the predictions of the two vertices defining the edge. Since this involves two facets, it may seem that the resulting potential will be of order 4 (i.e., 2 rotations and 2 translations). However, as shown below, our formulation has the advantage of decomposing the potential into a sum of unary and pairwise terms.

Let i_1 and i_2 be the indices of two facets sharing a mesh edge, as illustrated in Fig. 2(d). Let us denote by $\mathbf{y}^1_{i_1}$ and $\mathbf{y}^1_{i_2}$ the first pair of corresponding vertices in the two facets. The squared distance between these corresponding points can be written as

$$(d^1_{i_1,i_2})^2 = \left\|\mathbf{y}^1_{i_1} - \mathbf{y}^1_{i_2}\right\|^2_2 = \left\|\mathbf{R}_{i_1}\bar{\mathbf{y}}^1_{i_1} + \mathbf{t}_{i_1} - \mathbf{R}_{i_2}\bar{\mathbf{y}}^1_{i_2} - \mathbf{t}_{i_2}\right\|^2_2 \ . \tag{9}$$

By expanding the previous squared distance, we obtain

$$\begin{aligned}(d^1_{i_1,i_2})^2 = {} & \bar{\mathbf{y}}^{1^T}_{i_1}\mathbf{R}^T_{i_1}\mathbf{R}_{i_1}\bar{\mathbf{y}}^1_{i_1} + \mathbf{t}^T_{i_1}\mathbf{t}_{i_1} + \bar{\mathbf{y}}^{1^T}_{i_2}\mathbf{R}^T_{i_2}\mathbf{R}_{i_2}\bar{\mathbf{y}}^1_{i_2} + \mathbf{t}^T_{i_2}\mathbf{t}_{i_2} + 2\bar{\mathbf{y}}^{1^T}_{i_1}\mathbf{R}^T_{i_1}\mathbf{t}^1_{i_1} \\ & + 2\bar{\mathbf{y}}^{1^T}_{i_1}\mathbf{R}^T_{i_1}\mathbf{R}_{i_2}\bar{\mathbf{y}}^1_{i_2} + 2\bar{\mathbf{y}}^{1^T}_{i_1}\mathbf{R}^T_{i_1}\mathbf{t}_{i_2} + 2\mathbf{t}^T_{i_1}\mathbf{R}_{i_2}\bar{\mathbf{y}}^1_{i_2} + 2\mathbf{t}^T_{i_1}\mathbf{t}_{i_2} + 2\bar{\mathbf{y}}^{1^T}_{i_2}\mathbf{R}^T_{i_2}\mathbf{t}_{i_2} \ .\end{aligned} \tag{10}$$

Note that the first and third terms are constant due to the properties of rotation matrices. More importantly, note that all the terms are only functions of at most two variables. Therefore, the resulting potential obtained by summing the squared distances of both pairs of corresponding vertices, written as

$$\phi^c_{\alpha_{i_1,i_2}}(\mathbf{R}_{i_1}, \mathbf{t}_{i_1}, \mathbf{R}_{i_2}, \mathbf{t}_{i_2}) = -(d^1_{i_1,i_2})^2 - (d^2_{i_1,i_2})^2 \ , \tag{11}$$

is a sum of unary and pairwise terms.

Surface Smoothness*:* In addition to enforcing coherence of the facets, one might also want to encode some knowledge about the possible surface deformations. A classical example of this was introduced in the Active Contour Model [13], where the contour is encouraged to remain smooth by penalizing a quadratic function that approximates the sum of the square of the curvature along the contour. Following a similar idea, and assuming that the mesh forms a regular grid, we enforce smoothness by encouraging two aligned edges (i.e., horizontal or vertical edges in the grid) to remain straight.

Let i_1 and i_2 be the indices of two facets, each of which contains one of two aligned edges, as illustrated in Fig. 2(d). Furthermore, without loss of generality, let us assume that $\mathbf{y}_{i_1}^2$ and $\mathbf{y}_{i_2}^1$ correspond to the vertex shared by both facets. An energy encoding the squared curvature of these two edges can be written as

$$\begin{aligned} c_{i_1,i_2}^2 &= \left\| -\mathbf{R}_{i_1}\bar{\mathbf{y}}_{i_1}^1 - \mathbf{t}_{i_1} + \mathbf{R}_{i_1}\bar{\mathbf{y}}_{i_1}^2 + \mathbf{t}_{i_1} + \mathbf{R}_{i_2}\bar{\mathbf{y}}_{i_2}^1 + \mathbf{t}_{i_2} - \mathbf{R}_{i_2}\bar{\mathbf{y}}_{i_2}^2 - \mathbf{t}_{i_2} \right\|_2^2 \\ &= \left\| -\mathbf{R}_{i_1}\bar{\mathbf{y}}_{i_1}^1 + \mathbf{R}_{i_1}\bar{\mathbf{y}}_{i_1}^2 + \mathbf{R}_{i_2}\bar{\mathbf{y}}_{i_2}^1 - \mathbf{R}_{i_2}\bar{\mathbf{y}}_{i_2}^2 \right\|_2^2 \ , \end{aligned} \tag{12}$$

where we computed the location of the vertex shared by the two edges as the average over both facet predictions. Note that the translation variables have cancelled each other out. As a consequence, it is obvious that this decomposes into a sum of terms that involve at most two variables. Therefore, we can write the smoothness potential

$$\phi_{\alpha_{i_1,i_2}}^s(\mathbf{R}_{i_1}, \mathbf{R}_{i_2}) = -c_{i_1,i_2}^2 \ , \tag{13}$$

which is purely pairwise, since the unary terms involving the rotations become constant (i.e., as before, $\bar{\mathbf{y}}_{i_1}^{1^T}\mathbf{R}_{i_1}^T\mathbf{R}_{i_1}\bar{\mathbf{y}}_{i_1}^1 = cst$).

Other shape regularizers have been used for 3D reconstruction and could possibly be incorporated into our formalism. However, as shown in our experiments, these general potentials are sufficient to perform accurate 3D reconstruction.

3.3 Learning the Potential Weights

Given a few training examples where both image and ground-truth 3D shape are available, structured prediction methods can also be used to learn the weights of the different potentials of interest. This is in contrast with most existing approaches to non-rigid 3D reconstruction where the weights are typically set manually. We rely on the family of structured prediction problems introduced in [11] to learn our weights. In particular, we make use of their CRF formulation with ℓ_2 regularization (i.e., following the notation of [11], $\epsilon = 1$ and $p = 2$). Since this formulation is designed for discrete variables, we draw N sample rotations and translations for each facet, and keep them fixed for the entire procedure.

In addition to the potentials defined above, learning the weights requires a loss function encoding the error of a configuration with respect to the ground-truth reconstruction. Here, we use a squared point-to-point distance. More specifically, for each facet i, the loss can be written as

$$\Delta(\mathbf{R}_i, \mathbf{t}_i) = \sum_{k=1}^{3} \left\| \mathbf{R}_i\bar{\mathbf{y}}_i^k + \mathbf{t}_i - \check{\mathbf{y}}_i^k \right\|_2^2 \ , \tag{14}$$

where $\check{\mathbf{y}}_i^k$ is the ground-truth location of the vertex corresponding to the k^{th} vertex of facet i. It can easily be checked that this loss also consists of a sum of unary and pairwise terms. See [11] for more details on the learning method.

As shown in our experimental evaluation, only very few training examples are required to learn the potential weights. This is in contrast with reconstruction techniques that exploit learned deformation models, such as [20], which typically require many more training examples. This makes our approach more practical to deploy in general scenarios.

3.4 Shape Refinement with Gradient-based Optimization

Performing PCBP on large graphs (i.e., fine meshes) can quickly become computationally prohibitive. To overcome this issue, we follow a simple coarse-to-fine strategy: We first compute an initial solution on a coarse mesh using the structured prediction approach described above, and then refine this solution using a gradient-based method. Since structured prediction provides us with a good initial shape estimate, a gradient-based method becomes very well suited. More specifically, we follow the gradient-based approach of [23] for inextensible surfaces, which directly optimizes the 3D locations of the mesh vertices. This approach was extended in [21] to handle more general image likelihoods than the reprojection error of feature points for which it was originally designed.

Let $\mathbf{y}$ be the $3N_v$-dimensional vector of mesh vertices, initialized with our subdivided coarse structured prediction. We refine the 3D surface shape by solving the optimization problem

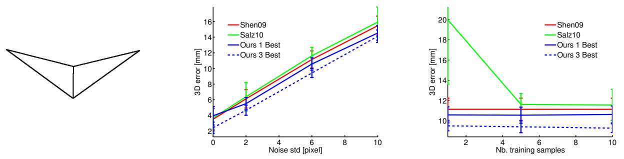
$$\begin{aligned} &\min_{\mathbf{y}} -\sum_i w_i \phi_i'(\mathbf{y}) - \sum_\alpha w_\alpha \phi_\alpha'(\mathbf{y}) \\ &\text{s. t. } \|\mathbf{y}^j - \mathbf{y}^k\|_2^2 = l_{j,k}^2 \;\; \forall (j,k) \in \mathcal{E} \;, \end{aligned} \tag{15}$$

where $l_{j,k}$ is the known reference distance between vertices $\mathbf{y}^j$ and $\mathbf{y}^k$, and $\mathcal{E}$ is the set of mesh edges. ϕ_i' and ϕ_α' are the same potentials as for structured prediction, but expressed in terms of the mesh vertices.

Following [23, 21], we obtain the solution to this optimization problem by iteratively linearizing the constraints and performing a few (i.e., 100 in practice) gradient descent steps in the null space of the linearized constraints. This scheme is carried out until convergence, or until a maximum number of iterations has been reached. More details on the overall procedure can be found in [23, 21].

4 Experimental Evaluation

We demonstrate the effectiveness of our method in various scenarios including feature point correspondences, as well as more complex image likelihoods with well- and poorly-textured surfaces. For all our experiments, we ran 20 iterations of PCBP, and initialized $\sigma_r = \pi/8$ and $\sigma_t = 10$, with $\eta_r = \eta_t = 0.75$. We used $N = 100$ states, except for the real images where $N = 200$. For the first iteration, we used the reference shape as initialization, thus yielding identity rotation matrices and translations corresponding to the centroids of the facets.



(a) (b) (c)

Fig. 3. Reconstructing a 2×2 mesh from correspondences. (a) Sample deformed mesh. 3D error as a function of (b) the 2D input noise, and (c) the number of training examples. Note that with few training examples, Salz10 performs poorly. In contrast, our approach performs well independently of the number of training examples.

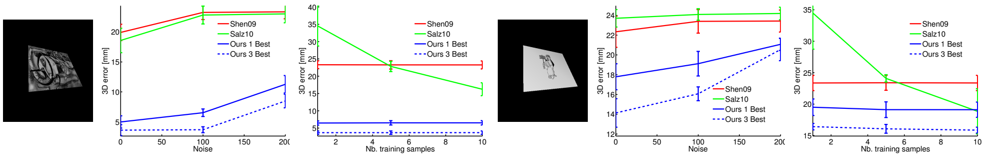
At each iteration, we kept either $M = 1$ or $M = 3$ solutions around which to re-sample. Corresponding results are denoted by Ours 1 Best and Ours 3 Best.

We compare our results against two baselines. The first one, later denoted by Shen09, corresponds to [23] initialized with the reference shape, with the extension of [21] to allow for more general image likelihoods than feature point reprojection error. The second baseline, later denoted by Salz10, follows the method of [21] and uses a Gaussian process (GP) predictor to initialize the shape before gradient-based optimization. To learn the GP predictor, we used the same training shapes as to learn the potential weights, and employ either noisy 2D point locations, or PHOG descriptors as input. To confirm that a simple coarse-to-fine optimization scheme is not enough to solve the problem, we also compare our results with a coarse-to-fine version of [23], denoted by Shen09 CTF. For all the baselines, we used the same image likelihoods as for our method, together with the weights learned with our CRF formulation.

In the remainder of this section, we present our results on synthetic data, motion capture data, and real images. 3D reconstruction errors are computed as the mean vertex-to-vertex distance between the ground-truth meshes and the reconstructions, averaged over 100 test images and for 5 train/test partitions.

Synthetic Data: As a first example, we consider the case of a 100 × 100mm mesh made of two facets, whose common edge act as a hinge, as depicted by Fig. 3(a). Deformations of this mesh were generated by randomly setting the angle between the two facets, as well as the global motion of the mesh. In this scenario, neither smoothness potential nor coarse-to-fine scheme were used.

To evaluate the performance of our approach on the popular problem of 3D reconstruction from feature point correspondences, we projected the deformed meshes in a 512×512 image using a known camera, added zero mean Gaussian noise with standard deviations $\{0, 2, 6, 10\}$ pixels to the 2D projections of the vertices, and used these noisy 2D locations as image measurements. We learned our potential weights and the GP predictor of Salz10 with $\{1, 5, 10\}$ training examples. Fig. 3(b,c) depict the 3D reconstruction errors as a function of the 2D measurement noise and of the number of training examples. Our approach outperforms the baselines, especially when keeping multiple solutions throughout the PCBP iterations. Note that with few training examples, Salz10 performs



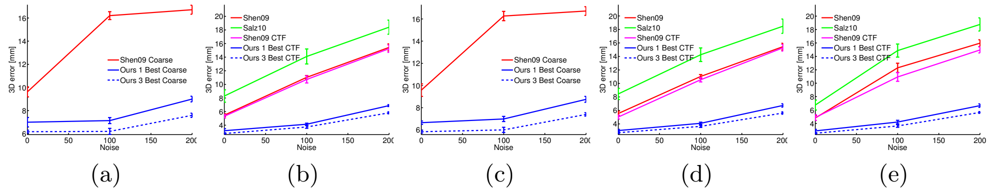
(a) (b) (c) (d) (e) (f)

Fig. 4. Reconstructing a 2×2 mesh from well- and poorly-textured images. (a) Sample well-textured input image. 3D error as a function of (b) the 2D input noise, and (c) the number of training examples. (d-f) Similar figures for the poorly-textured case. Note that our results are much more accurate than the baselines.

quite poorly. In contrast, our approach is very robust to the number of training examples; Even a single one is enough for us to learn the potential weights.

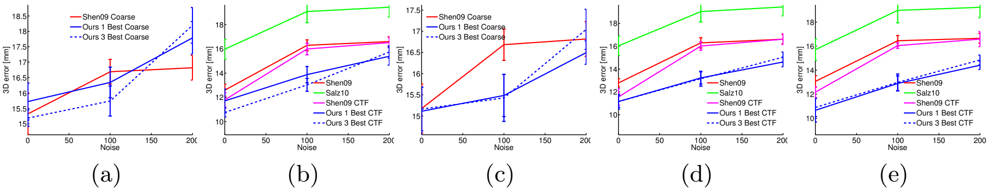
While feature point correspondences are an interesting source of information, our goal here is to address the problem of using more complex image likelihoods. To this end, we applied two different textures to the deformed meshes to create synthetic images such as those depicted in Fig. 4(a,d). We then added uniform random noise to the image intensities with maximum values of $\{0, 100, 200\}$. For all approaches, we used template matching and boundary likelihoods to reconstruct the surfaces. Fig. 4(b,c,e,f) depict the 3D errors as a function of the noise variance and of the number of training examples. In the well-textured case, our method yields a huge improvement over the baselines, thus fully showing the benefits of global optimization over local one. While improvement for the poorly-textured images is slightly smaller, it remains quite large. The lack of texture yields more ambiguities, which explains why keeping multiple solutions throughout PCBP yields significantly better results.

Motion Capture Data: The second set of experiments was performed using data obtained with a motion capture system [7]. The data consists of 3D reconstructions of reflective markers placed in a 9×9 regular grid of 160×160mm on a piece of cardboard deformed in front of 6 infrared cameras. Therefore, as opposed to the previous experiments, the deformations come from a real surface. Since no images are provided with the 3D data, we synthesized well- and poorly-textured images as before. In this experiment, we made use of our coarse-to-fine scheme, and performed our initial structured prediction with a 3×3 mesh. We used 5 training examples to learn the potential weights. We performed reconstruction with and without the smoothness prior to evaluate the performance of our algorithm when relying only on image information, in addition to the facet coherence term which is equivalent to the distance constraints of the baselines. Furthermore, since for the same deformation, a fine mesh is actually smoother than a coarse one, we also computed results by increasing the smoothness weight manually for refinement. Note that this was also performed for the baselines. Fig. 5(a,b) depict the 3D errors with no smoothness for the well-textured surface with a coarse mesh and after refinement, respectively. Our approach yields much more accurate reconstructions than the baselines. In Fig. 5(c-e), we show the 3D errors when using the smoothness term. Note that with this nice texture,



(a) (b) (c) (d) (e)

Fig. 5. Reconstructing a piece of cardboard from well-textured images. 3D error when (a) using a coarse (3×3) mesh and no smoothness, and (b) refining the results of (a) with a gradient-based method. (c-d) Similar results as (a-b) but with smoothness. (e) 3D errors when manually increasing the influence of the smoothness term for refinement. Shen09 and Salz10 were directly obtained using a fine mesh. Note that our coarse results give a much better initialization for the refinement step.



(a) (b) (c) (d) (e)

Fig. 6. Reconstructing a piece of cardboard from poorly-textured images. Similar plots as in Fig. 5. Note that here, the smoothness term has more influence on our results. Interestingly, increasing smoothness does not help the baselines significantly.

smoothing has very little effect on the results. Fig. 6 depicts similar results for a poorly-textured surface; Without smoothness, our coarse results are roughly on par with Shen09. Interestingly, however, we outperform the baselines after refinement. This shows that our coarse results still provide a better initialization than the coarse version of Shen09. Note that with this poorly-textured surface, smoothness improves reconstruction, which seems natural since image information is much weaker. This, however, is not noticeably the case for the baselines.

Real Images: Finally, to show that our approach can also be applied to real images, we used two sequences of different deforming materials [7]. While these are video sequences, all the images were treated independently and initialized from the template mesh to illustrate the fact that our approach can perform reconstruction from a single input image. Since no training data is available for these surfaces, we used a single training example consisting of the template mesh with reference image to learn the potential weights. In Fig. 7, we visually compare our reconstructions to those of Shen09. We do not show the results of Salz10, since with the template mesh as single training example, it would always predict the reference shape, and thus perform the same as Shen09. For the well-textured surface, Shen09 manages to reconstruct fairly large deformations. However, as illustrated by the two leftmost columns of the figure for two very similar frames, it is less consistent than our approach. For the poorly-textured surface, the baseline is completely unable to cope with large deformations. Our approach, however, still manages to reconstruct the surface. In the rightmost column of the figure, we show a failure case of our approach, where the facet

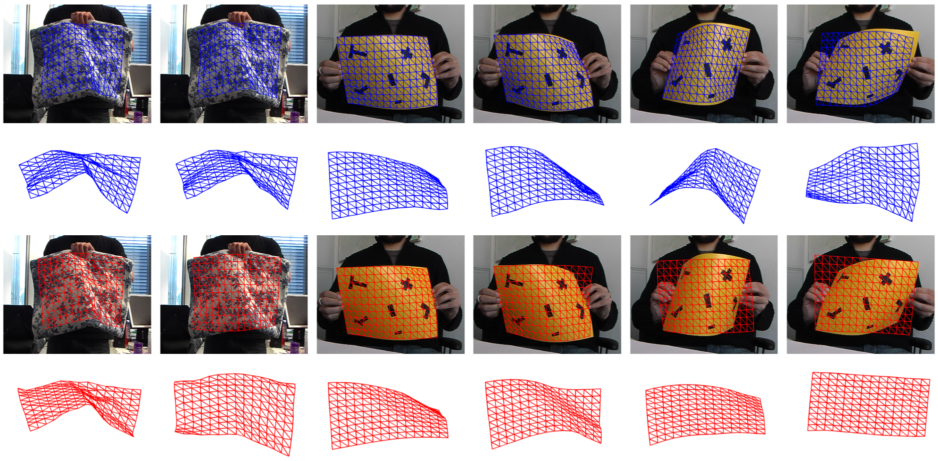


Fig. 7. Reconstructing surfaces from real images. From top to bottom: Our reconstructions reprojected on the original images, side view of our reconstructions, reconstructions obtained with Shen09 CTF reprojected on the original images, side view of those reconstructions. For a well-textured surface, the baseline manages to reconstruct fairly large deformations, but is less consistent than our approach, as illustrated for two very similar frames. For a poorly-textured surface, the baseline only manages to reconstruct small deformations, whereas our approach can deal with much larger ones. The rightmost column shows a failure of our method due to an ambiguity in the facet reconstruction and to the use of a coarse mesh.

orientation is ambiguous. Furthermore, the topology of the coarse mesh makes it harder to bend the surface along this diagonal. Note, however, that as opposed to the baseline, we still recover some degree of surface deformation.

5 Conclusion

We have introduced an approach to non-rigid 3D reconstruction of a potentially poorly-textured surface from a single image when no good initialization is available. To this end, we have formulated reconstruction as a structured prediction problem, and have shown that the popular image likelihoods decompose into unary and pairwise potentials, thus making inference algorithms practical for our purpose. We have demonstrated the benefits of our approach over state-of-the-art gradient-based methods in various scenarios, and have shown tremendous improvement over existing baselines. The current main limitation of our technique comes from the computational burden of performing structured prediction with large graphs. However, as research in that field advances, our approach will be applicable to denser and denser meshes. Studying these advances, as well as other image information such as shading, will be the focus of our future work.

References

1. I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid Structure from Motion in Trajectory Space. In *NIPS*, 2008.

2. D. Batra, P. Yadollahpour, A. Guzman-Rivera and G. Shakhnarovich. M-Best Modes: Extracting Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, 2012.
3. V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, 1999.
4. C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *CVPR*, 2000.
5. F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces. In *ACCV*, 2010.
6. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active Appearance Models. In *ECCV*, 1998.
7. http://cvlab.epfl.ch/data/dsr/.
8. J. Fayad, L. Agapito, and A. Del Bue. Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences. In *ECCV*, 2010.
9. M. Fromer, and A. Globerson. An LP view of the M best problem. In *NIPS*, 2009.
10. N.A. Gumerov, A. Zandifar, R. Duraiswami, and L.S. Davis. Structure of Applicable Surfaces from Single Views. In *ECCV*, 2004.
11. T. Hazan and R. Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *NIPS*, 2010.
12. T. Hazan, and A. Shashua. Norm-Product Belief Propagation: Primal-Dual Message-Passing for LP-Relaxation and Approximate Inference. In *IT*, 2011.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *IJCV*, 1988.
14. T. Mcinerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *ICCV*, 1993.
15. D. Metaxas and D. Terzopoulos. Constrained Deformable Superquadrics and Nonrigid Motion Tracking. *PAMI*, 1993.
16. J. Peng, T. Hazan, D. McAllester, and R. Urtasun. Convex Max-Product Algorithms for Continuous MRFs with Applications to Protein Folding. *ICML*, 2011.
17. A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *PAMI*, 1991.
18. M. Perriollat, R. Hartley, and A. Bartoli. Monocular Template-Based Reconstruction of Inextensible Surfaces. *IJCV*, 2010.
19. C. Russell, J. Fayad, and L. Agapito. Energy Based Multiple Model Fitting for Non-Rigid Structure from Motion. In *CVPR*, 2011.
20. M. Salzmann, R. Urtasun, and P. Fua. Local Deformation Models for Monocular 3D Shape Recovery. In *CVPR*, 2008.
21. M. Salzmann and R. Urtasun Combining Discriminative and Generative Methods for 3D Deformable Surface and Articulated Pose Reconstruction. In *CVPR*, 2010.
22. A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun Distributed Message Passing for Large Scale Graphical Models. In *CVPR*, 2011.
23. S. Shen, W. Shi, and Y. Liu. Monocular 3D Tracking of Inextensible Deformable Surfaces Under L2-Norm. In *ACCV*, 2009.
24. J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-Rigid Structure from Locally-Rigid Motion. In *CVPR*, 2010.
25. L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid Structure-From-Motion: Estimating Shape and Motion with Hierarchical Priors. *PAMI*, 2008.
26. A. Varol, A. Shaji, M. Salzmann, and P. Fua. Monocular 3D Reconstruction of Locally Textured Surfaces. *PAMI*, 2011.
27. J. Xiao and T. Kanade. Uncalibrated Perspective Reconstruction of Deformable Structures. In *ICCV*, 2005.