A Constrained Latent Variable Model *

Aydin Varol[1] Mathieu Salzmann[2,3] Pascal Fua[1] Raquel Urtasun[3]

[1] École Polytechnique Fédérale de Lausanne (EPFL), CVLab, Switzerland

[2] NICTA, 7 London Circuit, Canberra ACT 2600, Australia

[3] Toyota Technological Institute at Chicago, Chicago, IL 60637

aydin.varol@epfl.ch, mathieu.salzmann@nicta.com.au, pascal.fua@epfl.ch, rurtasun@ttic.edu

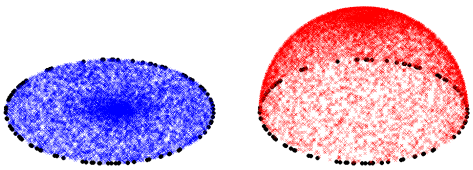
Abstract

Latent variable models provide valuable compact representations for learning and inference in many computer vision tasks. However, most existing models cannot directly encode prior knowledge about the specific problem at hand. In this paper, we introduce a constrained latent variable model whose generated output inherently accounts for such knowledge. To this end, we propose an approach that explicitly imposes equality and inequality constraints on the model's output during learning, thus avoiding the computational burden of having to account for these constraints at inference. Our learning mechanism can exploit non-linear kernels, while only involving sequential closed-form updates of the model parameters. We demonstrate the effectiveness of our constrained latent variable model on the problem of non-rigid 3D reconstruction from monocular images, and show that it yields qualitative and quantitative improvements over several baselines.

1. Introduction

Latent variable models have been widely used in a variety of computer vision problems, such as image classification [13, 32] and non-rigid pose estimation [24, 27, 20]. However, as effective as they are, they suffer from the fact that they ignore prior knowledge that might be available for the specific problem at hand. In particular, nothing prevents commonly-employed latent variable models from generating configurations that violate known constraints.

In this paper we propose a novel non-linear latent variable model whose output explicitly accounts for the inherent constraints of the problem. To this end, we learn a non-linear mapping from the latent space to the output space such that the generated outputs comply with equality and inequality constraints expressed in terms of the problem variables. We make use of unlabeled examples to enforce the constraints, while minimizing the prediction error of labeled ones. To allow for kernel-based mappings, we introduce a primal-dual optimization framework, where the mapping is learned by sequential closed-form updates. Our approach is completely generic and could be used in many different contexts, such as image classification to impose separation of the classes, and articulated tracking to constrain the space of possible poses.



Standard LVM Our constrained LVM

Figure 1. **Generating 3D points on a hemisphere.** (Left) Predictions from random samples in the latent space using an unconstrained latent variable model. (Right) Predictions from the same samples using our constrained latent variable model. Both models were learned using only the black dots as labeled training samples.

To illustrate the benefits of our model, we consider a toy case where the output is a single 3D point constrained to lie on a hemisphere. We learned a constrained latent variable model and its unconstrained version using the black dots close to the great circle of the hemisphere as labeled training examples. Fig. 1 shows the predictions of the unconstrained and constrained models from random samples on the latent space. Note how much the predictions are improved by learning a constrained model.

While this corresponds to an extreme case of poorly sampled training points, a similar scenario could easily occur locally on more complex output spaces. In particular, to demonstrate the effectiveness of our approach on a challenging real-world problem, we consider the task of non-rigid surface reconstruction from monocular inputs. This is known to be a very ambiguous problem due to the fact that any point on a line of sight reprojects at the same image location, thus making depth estimation very ill-posed. Furthermore, the partial lack of identifiable texture on the surface makes the use of shape regularizers necessary to produce reasonable reconstructions. Latent variable models are commonly used for such regularization [3, 5, 19, 20, 9]. Our experimental evaluation shows that our constrained latent variable model produces more accurate reconstructions than the standard linear subspace models and the increasingly popular Gaussian Process Latent Variable Model (GPLVM) [11], which corresponds to the unconstrained

*This work was partially supported by the Swiss National Science Foundation and NSF ID 1017626.

version of our model. This evaluation was performed in a variety of scenarios including real images of different materials captured with the Microsoft Kinect, providing ground-truth 3D measurements. The code to reproduce our experiments , our novel dataset and other supplementary materials are publicly available[1].

2. Related Work

In many computer vision problems, one would like a parameterization that models our prior knowledge of the task at hand and satisfies known constraints in terms of the variables of the problem. While such parameterizations exist for some specific problems, they are not generally available for all tasks. Latent variable models have come as an alternative to such parameterizations, where one tries to learn the properties of the variables of interest from training data.

Latent variable models effectively provide a compact representation of the problem at hand and improve robustness to noise and other sources of ambiguities by regularization. For instance, linear subspace models have been widely applied to problems such as human pose estimation [24], or non-rigid reconstruction [3]. More recently, non-linear latent variable models, such as the GPLVM [11], have become popular for computer vision applications [27, 20]. Unfortunately, these models only try to capture properties of interest from training data and completely ignore our knowledge of the problem at hand.

Several attempts at incorporating prior knowledge in latent variable models have been made. However, most approaches are restricted to encoding this knowledge in the latent space. For instance, sparse coding [13, 32] encourages sparsity of the latent representation; the Gaussian Process Dynamical Model (GPDM) [30, 28] allows to model dynamics in the latent space, although the dynamics are learned rather than explicitly encoded in the model; [29] introduces topological constraints to the GPLVM, but once more acting directly on the latent space; [1] imposes physics-based constraints on the latent space in the form of differential equations. However, imposing constraints on the latent variables does not guarantee that equivalent ones are satisfied in the output space. Therefore, these models will still produce outputs that violate known constraints.

To the best of our knowledge, [21] represents the only attempt to learn a predictor that implicitly satisfies constraints on the output variables. However, the method is limited to linear and quadratic equality constraints. Furthermore, it suffers from ambiguities in the predicted output that can only be overcome in specific scenarios.

In the context of non-rigid reconstruction, both linear [3, 5, 4, 31, 25, 10, 12] and non-linear [19, 20, 9, 8] latent variable models have been employed. However, since existing models are unable to make use of the known physical properties of the surface, they produce shapes that violate important constraints, and therefore look unnatural. An alternative approach is to directly encode the physical properties of the system. Physics-based approaches have made use both of the Finite Element Method [17, 16, 15, 14, 26, 2] and of more intuitive constraints, such as inextensibility [7, 23, 6, 18]. Unfortunately, while physics-based approaches have the advantage of explicitly encoding prior knowledge, they involve solving high-dimensional optimization problems. Furthermore, since constraints such as inextensibility are only local, the resulting methods typically require the surface to be well-textured.

Attempts at coupling latent variable models and constraints for deformable shape recovery have been made [22]. However, these methods are limited to linear subspace models and to specific constraints. Furthermore, and more importantly, they incorporate the constraints on top of a latent variable model that still allows for invalid configurations. It would seem more effective to exploit the constraints while learning the model, thus yielding a latent variable model that only generates physically-plausible deformations. This, in essence, is what we propose in this paper.

3. Learning a Constrained LVM

In this section, we present our approach to learning a latent variable model that incorporates constraints on the generated outputs. In particular, we focus on the problem of learning the mapping from a given latent space to the output space under equality and inequality constraints. Note that the latent space itself can be obtained with any available technique, such as PCA, or Isomap. Our mapping can thus be seen as a predictor from the latent space to the output space. We learn this mapping by minimizing a prediction error on labeled examples, for which the true output is known, while simultaneously enforcing constraints on unlabeled ones, for which the output is unknown. In the remainder of this section, we first derive the primal form of our learning problem. We then exploit duality to kernelize our approach, and thus be able to make use of the nonlinear kernels (e.g., RBF) that have been proven more effective than linear ones for many computer vision tasks.

3.1. Primal Optimization Problem

Let $\mathcal{X} \subseteq \mathbb{R}^m$ be a given latent space, and $\mathcal{Y} \subseteq \mathbb{R}^D$ be the output space of interest, such as the space of non-rigid 3D surfaces. Given a latent variable $\mathbf{x} \in \mathcal{X}$, our latent variable model can be encoded as a mapping of the form

$$\hat{\mathbf{y}} = \mathbf{W}\phi(\mathbf{x}) \ , \tag{1}$$

such that $\hat{\mathbf{y}} \in \mathcal{Y}$. $\mathbf{W} \in \mathbb{R}^{D \times d}$ is the parameter matrix that defines the mapping, and $\phi(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}^d$ is the feature map of the latent variable $\mathbf{x}$.

[1] Publicly available at http://cvlab.epfl.ch/software/clvm/index.php

Let $\mathcal{L}$ be the set of labeled training examples containing N pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ of latent variables $\mathbf{x}_i$ and associated continuous multi-dimensional labels (outputs) $\mathbf{y}_i$. Furthermore, let $\mathcal{U} = \mathcal{U}_E \cup \mathcal{U}_I$ be the set of unlabeled training examples $\bar{\mathbf{x}}_j$ subject to equality ($\mathcal{U}_E$) and inequality ($\mathcal{U}_I$) constraints. We formulate learning as a constrained optimization problem, where a loss function l is minimized on the labeled training set $\mathcal{L}$ subject to constraints on the unlabeled set $\mathcal{U}$. This can be written as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{\{\mathbf{x}_i,\mathbf{y}_i\}\in\mathcal{L}} l(\mathbf{W},\mathbf{x}_i,\mathbf{y}_i) + \gamma\mathcal{R}(\mathbf{W}) \qquad (2) \\ \text{s. t.} \quad & C(\mathbf{W},\bar{\mathbf{x}}_u) = 0 \qquad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D(\mathbf{W},\bar{\mathbf{x}}_v) \leq 0 \qquad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned}$$

where $\mathcal{R}$ is the regularizer on $\mathbf{W}$ with weight γ, and $C(\mathbf{W},\bar{\mathbf{x}}_u)$, resp. $D(\mathbf{W},\bar{\mathbf{x}}_v)$, is a vector function encoding all N_E equality constraints, resp. N_I inequality constraints, defined with respect to the prediction of the unlabeled data $\bar{\mathbf{x}}_u$, resp. $\bar{\mathbf{x}}_v$.

The problem of Eq. 2 is very general, and different loss functions, regularizers and constraints can be utilized. Here, we consider the case of the square loss, Frobenius norm regularizer, and arbitrary nonlinear constraints on the predictions. The optimization problem therefore becomes

$$\begin{aligned} \min_{\mathbf{W}} \quad & \frac{1}{2}\|\mathbf{W}\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 \qquad (3) \\ \text{s. t.} \quad & C\left(\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right) = 0 \quad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D\left(\mathbf{W}\phi(\bar{\mathbf{x}}_v)\right) \leq 0 \quad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned}$$

where $\phi(\mathbf{x}) = [\phi(\mathbf{x}_1)\cdots\phi(\mathbf{x}_N)]$, and $\mathbf{Y} = [\mathbf{y}_1\cdots\mathbf{y}_N]$ is the matrix of labeled training outputs.

If the constraints are non-convex, so is the optimization problem in Eq. 3. We therefore transform it so that we can solve it as a sequence of closed-form updates. First, we rewrite the inequality constraints as equalities by introducing slack variables $\boldsymbol{\epsilon}$. This yields the optimization problem

$$\begin{aligned} \min_{\mathbf{W},\boldsymbol{\epsilon}} \quad & \frac{1}{2}\|\mathbf{W}\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon}\|_2^2 \qquad (4) \\ \text{s. t.} \quad & C\left(\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right) = 0 \qquad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E \\ & D\left(\mathbf{W}\phi(\bar{\mathbf{x}}_v)\right) + \boldsymbol{\epsilon}^{(v)} \odot \boldsymbol{\epsilon}^{(v)} = 0 \quad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned}$$

where $\odot$ is the Hadamard (elementwise) product, $\boldsymbol{\epsilon}^{(v)}$ contains the slack variables associated with example v, and $\frac{\alpha}{2}\|\boldsymbol{\epsilon}\|_2^2$ encodes potential additional knowledge about the problem. This, for instance, is useful in conjunction with the inequality constraints of [22], where only small deviations from equalities are expected.

As a second step, we perform a first order Taylor expansion of the constraints. Given an initial solution for the parameters $\mathbf{W}$ and $\boldsymbol{\epsilon}$, we iteratively linearize the constraints around the current solution, and update the parameters by solving the linearized problem. At each iteration t of this procedure, the linearized problem can be written as

$$\begin{aligned} \min_{\delta\mathbf{W},\delta\boldsymbol{\epsilon}} \quad & \frac{1}{2}\|(\mathbf{W}_t + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 \qquad (5) \\ & + \frac{\gamma}{2}\|\mathbf{W}_t + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ \text{s. t.} \quad & C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u) = 0 \qquad \forall\bar{\mathbf{x}}_u \in \mathcal{U}_E \ , \\ & D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) \\ & \quad + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)} = 0 \qquad \forall\bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned}$$

where $\mathbf{W}_t$ and $\boldsymbol{\epsilon}_t$ are the current estimates of $\mathbf{W}$ and $\boldsymbol{\epsilon}$, respectively. $C_t^{(u)}$ is the value of the equality constraints for unlabeled example u at the current prediction $\hat{\mathbf{y}}_{u,t}$, and $\mathbf{G}_u$ is the $N_E \times D$ matrix containing the gradient of these constraints with respect to $\hat{\mathbf{y}}_{u,t}$. Similarly, $D_t^{(v)}$ and $\mathbf{Q}_v$ encode the value and gradient of the inequality constraints for unlabeled example v at the current prediction $\hat{\mathbf{y}}_{v,t}$. The solution to the problem in Eq. 5 can be obtained in closed-form by solving a linear system in $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$.

3.2. Kernel-based Mappings

The primal formulation of our latent variable model only allows for linear mappings from the feature map of the latent space to the output space. While some degree of nonlinearity can be encoded in the feature map, it results in the rapid growth of the number of parameters to optimize. This makes our primal formulation computationally expensive and more prone to overfitting. Furthermore, for many kernels (e.g., RBF), the feature maps cannot be explicitly computed.

We therefore need to kernelize our approach to take advantage of such kernels. To this end, we exploit duality. We start by first writing the Lagrangian of the minimization problem in Eq. 5, and then make use of the Karush-Kuhn-Tucker (KKT) conditions to derive a solution for the Lagrange multipliers. This yields an optimization method similar to the one in Section 3.1, where we iteratively linearize the constraints around the current prediction of the unlabeled data, solve for the Lagrange multipliers of the dual linearized problem, and update the prediction. Importantly, we show that the Lagrange multipliers can be obtained in closed-form, thus yielding a sequence of closed-form updates similar to the one in the primal formulation.

More specifically, the Lagrangian of the minimization problem in Eq. 5 can be expressed as

$$\begin{aligned} L = & \frac{1}{2}\|(\mathbf{W}_t + \delta\mathbf{W})\phi(\mathbf{x}) - \mathbf{Y}\|_F^2 + \frac{\gamma}{2}\|\mathbf{W}_t + \delta\mathbf{W}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{\epsilon} + \delta\boldsymbol{\epsilon}\|_2^2 \\ & + \sum_u \left[C_t^{(u)} + \mathbf{G}_u\delta\mathbf{W}\phi(\bar{\mathbf{x}}_u)\right]^T \boldsymbol{\lambda}_u^E \\ & + \sum_v \left[D_t^{(v)} + \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)} + \mathbf{Q}_v\delta\mathbf{W}\phi(\bar{\mathbf{x}}_v) + \boldsymbol{\epsilon}_t^{(v)} \odot \delta\boldsymbol{\epsilon}^{(v)}\right]^T \boldsymbol{\lambda}_v^I \ , \end{aligned}$$

where $\boldsymbol{\lambda}_u^E \in \mathbb{R}^{N_E}$ and $\boldsymbol{\lambda}_v^I \in \mathbb{R}^{N_I}$ are the Lagrange multipliers associated with the equality and inequality constraints for unlabeled examples u and v, respectively.

To find an optimal solution to our problem, we first make use of the KKT stationarity condition, which, in our case, states that the solution for $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$ must satisfy $\frac{\partial L}{\partial \delta \mathbf{W}} = 0$ and $\frac{\partial L}{\partial \delta \boldsymbol{\epsilon}} = 0$, respectively.

Claim 1 *Solving the KKT stationarity conditions yields*

$$\begin{aligned} \delta\mathbf{W} &= \mathbf{A}\mathbf{Z} - \mathbf{W}_t, \\ \delta\boldsymbol{\epsilon}^{(v)} &= -\left(\frac{1}{\alpha}\boldsymbol{\lambda}_v^I + \mathbf{1}\right) \odot \boldsymbol{\epsilon}_t^{(v)} \ , \quad \forall \bar{\mathbf{x}}_v \in \mathcal{U}_I \ , \end{aligned} \tag{6}$$

respectively, where

$$\mathbf{A} = \left[\mathbf{M} - \sum_u \mathbf{G}_u^T \boldsymbol{\lambda}_u^E \mathbf{K}_{u,:} - \sum_v \mathbf{Q}_v^T \boldsymbol{\lambda}_v^I \mathbf{K}_{v,:}\right] \mathbf{B}^{-1}, \tag{7}$$

$$\mathbf{Z} = \begin{bmatrix} \phi(\mathbf{x}) & \cdots & \phi(\bar{\mathbf{x}}_{u'}) & \cdots & \phi(\bar{\mathbf{x}}_s) & \cdots & \phi(\bar{\mathbf{x}}_{v'}) & \cdots \end{bmatrix}^T,$$

$$\mathbf{B} = \mathbf{K}_{:,\mathcal{L}}\mathbf{K}_{\mathcal{L},:} + \gamma \mathbf{K}_{:,:} \ ,$$

$$\mathbf{M} = \mathbf{Y}\mathbf{K}_{\mathcal{L},:} \ ,$$

with u',s and v' the indices of the unlabeled data in $\mathcal{U}_E \backslash \mathcal{U}_I$, $\mathcal{U}_E \cap \mathcal{U}_I$, and $\mathcal{U}_I \setminus \mathcal{U}_E$, respectively. The kernel $\mathbf{K}_{:,:}$ is defined as

$$\mathbf{K}_{:,:} = \mathbf{Z}\mathbf{Z}^T = \left[\begin{array}{c|c} \mathbf{K}_{\mathcal{L},\mathcal{L}} & \mathbf{K}_{\mathcal{L},\mathcal{U}} \\ \hline \mathbf{K}_{\mathcal{U},\mathcal{L}} & \mathbf{K}_{\mathcal{U},\mathcal{U}} \end{array}\right] \ , \tag{8}$$

and can be computed via any kernel function, e.g., RBF.

Proof: In supplementary material. □

The KKT stationarity conditions define a solution for our variables $\delta\mathbf{W}$ and $\delta\boldsymbol{\epsilon}$ in terms of the Lagrange multipliers $\boldsymbol{\lambda}_u^E$ and $\boldsymbol{\lambda}_v^I$. To find a solution for these Lagrange multipliers, we make use of the KKT primal feasibility condition, which states that the constraints should be satisfied at the optimal value of the parameters.

Claim 2 *The solution to the constraints encoded by the KKT primal feasibility condition takes the form $\boldsymbol{\lambda} = \mathbf{S}^{-1}\mathbf{r}$, where*

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}^E \\ \boldsymbol{\lambda}^I \end{bmatrix} \ , \quad \mathbf{S} = \left[\begin{array}{c|c} \mathbf{S}^{E,E} & \mathbf{S}^{E,I} \\ \hline \mathbf{S}^{I,E} & \mathbf{S}^{I,I} \end{array}\right] \ , \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}^E \\ \mathbf{r}^I \end{bmatrix} \ ,$$

and

$$\begin{aligned} \mathbf{S}_{u,a}^{E,E} &= \mathbf{G}_u \mathbf{G}_a^T (\mathbf{K}_{a,:}\mathbf{B}^{-1}\mathbf{K}_{:,u}) \ , \\ \mathbf{S}_{u,b}^{E,I} &= \mathbf{G}_u \mathbf{Q}_b^T (\mathbf{K}_{b,:}\mathbf{B}^{-1}\mathbf{K}_{:,u}), \\ \mathbf{S}_{v,a}^{I,E} &= \mathbf{Q}_v \mathbf{G}_a^T (\mathbf{K}_{a,:}\mathbf{B}^{-1}\mathbf{K}_{:,v}), \\ \mathbf{S}_{v,b}^{I,I} &= \mathbf{Q}_v \mathbf{Q}_b^T (\mathbf{K}_{b,:}\mathbf{B}^{-1}\mathbf{K}_{:,v}) + \delta_{v,b}\, diag\left(\frac{1}{\alpha}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)}\right), \\ \mathbf{r}_u^E &= \mathbf{G}_u \mathbf{M}\mathbf{B}^{-1}\mathbf{K}_{:,u} - \mathbf{G}_u \hat{\mathbf{y}}_{u,t} + C_t^{(u)}, \\ \mathbf{r}_v^I &= \mathbf{Q}_v \mathbf{M}\mathbf{B}^{-1}\mathbf{K}_{:,v} - \mathbf{Q}_v \hat{\mathbf{y}}_{v,t} + D_t^{(v)} - \frac{1}{2}\boldsymbol{\epsilon}_t^{(v)} \odot \boldsymbol{\epsilon}_t^{(v)}, \end{aligned}$$

with $\delta_{v,b}$ the Kronecker delta.

Proof: In supplementary material. □

In the two claims above, we have shown how to obtain in closed-form the Lagrange multipliers that give the optimal solution to the problem in Eq. 5. Note that this requires having a prediction for the unlabeled examples $\hat{\mathbf{y}}_{v,t}$ at the current iteration t. Furthermore, at inference, to make use of our latent variable model, we need to be able to compute the prediction for a new input. To address these points, we now define the form of the prediction in our model.

Claim 3 *Prediction for any input $\mathbf{x}_*$ in our kernelized model can be done in closed-form, and can be written as*

$$\hat{\mathbf{y}}_* = \mathbf{A}\mathbf{K}_{:,*} \ , \tag{9}$$

where $\mathbf{K}_{:,*} = \mathbf{Z}\phi(\mathbf{x}_*) = \left[\begin{array}{c|c} \mathbf{K}_{*,\mathcal{L}} & \mathbf{K}_{*,\mathcal{U}} \end{array}\right]^T$.

Proof: In supplementary material. □

Since we follow the same linearization strategy as in Section 3.1, learning still consists of a succession of updates based on the current prediction for the unlabeled inputs. Therefore, we can derive an algorithm that iteratively linearizes the constraints around the current prediction, solves for the Lagrange multipliers and refines the prediction. This scheme is summarized in Algorithm 1. Note that each step can be done in closed-form. Note also that, even though Claim 1 defines the update $\delta\mathbf{W}$ in terms of the Lagrange multipliers, $\mathbf{W}$ is never explicitly computed in our algorithm, thus making it fully kernelized.

Algorithm 1 Learning a constrained latent variable model

Initialize $\hat{\mathbf{y}}_{u,0}$ and $\hat{\mathbf{y}}_{v,0}$ using an unconstrained predictor
Initialize $\boldsymbol{\epsilon}_0$ to non-zero values
for $t = 1$ to #iters **do**
  Compute $\mathbf{G}_u$, $C_t^{(u)}$, $\mathbf{Q}_v$, $D_t^{(v)}$ from $\hat{\mathbf{y}}_{u,t-1}$, $\hat{\mathbf{y}}_{v,t-1}$
  Compute $\mathbf{S}$ from Claim 2
  Compute $\mathbf{r}$ from Claim 2
  Compute $\boldsymbol{\lambda}_u^E$ and $\boldsymbol{\lambda}_v^I$ using $\boldsymbol{\lambda} = \mathbf{S}^{-1}\mathbf{r}$
  Compute $\mathbf{A}$ using Eq. 7
  Compute $\hat{\mathbf{y}}_{u,t}$ and $\hat{\mathbf{y}}_{v,t}$ using Eq. 9
  Compute $\boldsymbol{\epsilon}_t$ using Eq. 6
end for

4. Experimental Evaluation

We demonstrate the effectiveness of our constrained latent variable model at reconstructing deformable surfaces from monocular images. We compare our results to those obtained using a linear subspace model and an unconstrained version of our non-linear model, which corresponds to a GPLVM. In all cases, reconstruction is obtained by optimizing the latent variable so as to minimize an image-based loss function. Errors are given in terms of both the 3D reconstruction errors and the constraint violation. Reconstruction errors are computed as the average point-to-point distance between ground-truth and reconstructed shapes. Constraint violation is taken to be the mean value

			Reconstruction error [mm]				Constraint violation [mm]			
			$V = 20$	$V = 60$	$V = 100$	$V = 150$	$V = 20$	$V = 60$	$V = 100$	$V = 150$
Cardboard	Equality	PCA	31.6±4.45	31.6±4.45	31.6±4.45	31.6±4.45	3.19±0.64	3.19±0.64	3.19±0.64	3.19±0.64
		Unconstrained	4.64±0.29	4.64±0.29	4.64±0.29	4.64±0.29	0.98±0.05	0.98±0.05	0.98±0.05	0.98±0.05
		Ours	**4.48±0.31**	**4.28±0.33**	4.30±0.36	4.27±0.39	**0.81±0.02**	**0.63± 0.01**	**0.50±0.02**	**0.41±0.02**
	Ineq.	PCA	31.6±4.45	31.6±4.45	31.6±4.45	31.6±4.45	3.43±0.76	3.43±0.76	3.43±0.76	3.43±0.76
		Unconstrained	4.64±0.29	4.64±0.29	4.64±0.29	4.64±0.29	0.94±0.04	0.94±0.04	0.94±0.04	0.94±0.04
		Ours	4.52±0.31	4.34±0.30	**4.25± 0.28**	**4.12±0.27**	**0.77±0.03**	**0.56±0.02**	**0.43±0.02**	**0.34±0.02**
Cloth	Equality	PCA	16.2±2.19	16.2±2.19	16.2±2.19	16.2±2.19	3.05±0.26	3.05±0.26	3.05±0.26	3.05±0.26
		Unconstrained	4.36±0.20	4.36±0.20	4.36±0.20	4.36±0.20	1.33±0.04	1.33±0.04	1.33±0.04	1.33±0.04
		Ours	4.30±0.19	4.29±0.12	4.27±0.21	4.44±0.17	**1.20±0.04**	**1.03± 0.04**	**0.88±0.02**	**0.75±0.02**
	Ineq.	PCA	16.2±2.19	16.2±2.19	16.2±2.19	16.2±2.19	3.43±0.29	3.43±0.29	3.43±0.29	3.43±0.29
		Unconstrained	4.36±0.20	4.36±0.20	4.36±0.20	4.36±0.20	1.14±0.08	1.14±0.08	1.14±0.08	1.14±0.08
		Ours	**4.25±0.17**	**4.10±0.14**	**3.99± 0.17**	**3.95±0.16**	**1.02±0.09**	**0.87±0.06**	**0.73±0.03**	**0.63±0.02**

Table 1. **Predicting shapes from known latent variables.** Reconstruction error and constraint violation as a function of the number of unlabeled examples V for a fixed $N = 50$. Note that the constraint violation measure is different for inequalities and for equalities.

of $C(\mathbf{x}_*)$ for equalities and the mean value of $D(\mathbf{x}_*)$ for all violated inequality constraints. All the quantitative results are expressed in millimeters.

In the remainder of this section, we first describe our learning setup and the different types of constraints used in our experiments. We then present our results on synthetic data and real images of surfaces made of different materials. Our results include a quantitative evaluation of reconstructions obtained from real images captured with a Microsoft Kinect, whose output depth we treat as ground-truth.

4.1. Learning Setup

To learn the models, we make use of two publicly available datasets obtained with a motion capture system[2]. The first one consists of the 3D locations of markers placed as a 9×9 grid on a piece of cardboard, thus forming a square mesh of size 160×160mm with 208 edges. The second one consists of similar measurements on a piece of cloth, represented by a 9×7 mesh of size 160×120mm with 158 edges. The cardboard dataset exhibits simpler deformations than the cloth one. To obtain latent spaces for each dataset independently, we performed PCA on the 3D marker locations. We used 12 and 30 latent variables for the cardboard and cloth datasets, respectively, which covers more than 95% of the variance of the data. In all the experiments, we used an RBF kernel for our model and its unconstrained version. We set both regularization weights γ and α in Eq. 5 to 0.001.

We investigated the use of length constraints as both equalities and inequalities. Under the former, the length of the edges connecting the mesh vertices must remain constant. The latter allow these lengths to decrease to model the fact that two vertices may come closer to each other if folds appear between them, but cannot be further apart than the geodesic distance along the surface. The equalities have been shown to be appropriate for smoothly deforming surfaces, and the inequalities for surfaces undergoing more complex deformations [22]. To confirm this, we predicted shapes given the true latent variables of 500 test examples.

[2]Publicly available at http://cvlab.epfl.ch/data/dsr/

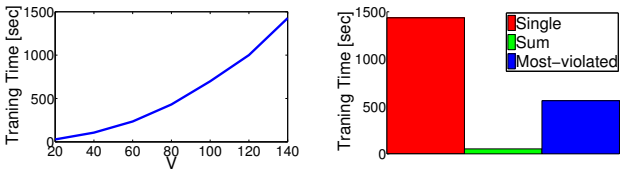


Figure 2. **Training times.** (Left) Training time as a function of the number of validation samples when using all constraints. (Right) Training times for different constraint selection strategies for a fixed number of validation samples $V = 140$.

Table 1 depicts reconstruction and constraint errors averaged over the test samples as a function of the number of unlabeled examples V. Note that the constraint violation measure is different for inequalities and equalities. For the cardboard dataset, both constraint types perform well. For the cloth dataset where sharper folds occur, inequality constraints are more appropriate; increasing V improves both reconstruction and constraint satisfaction for inequalities, whereas it only improves constraint satisfaction for equalities. Note that our predictions are more accurate than those of the baselines.

Our implementation can handle up to 60K constraints on a standard PC. However, this still limits us in the number of unlabeled examples that we can use. To increase this number, we implemented a different strategy to encode the constraints, which involves summing over individual ones. This yields new constraints of the form $\tilde{C}(\mathbf{x}) = \sum_j C_j(\mathbf{x}) = 0$, and reduces the number of constraints for each unlabeled sample, which lets us use more of them. In practice, we define these sums of constraints as the sums of all individual vertical or horizontal constraints on the rectangular grid, which amounts to preserving the length of a complete horizontal or vertical line as opposed to that of individual edges.

The constraints we use are sparse in nature, since each one only depends on two mesh vertices. Thus, the linear system to obtain the Lagrange multipliers is sparse as well, which allows for the use of efficient sparse solvers. Fig. 2(left) depicts training time as a function of the number of unlabeled examples for the cardboard dataset. To im-

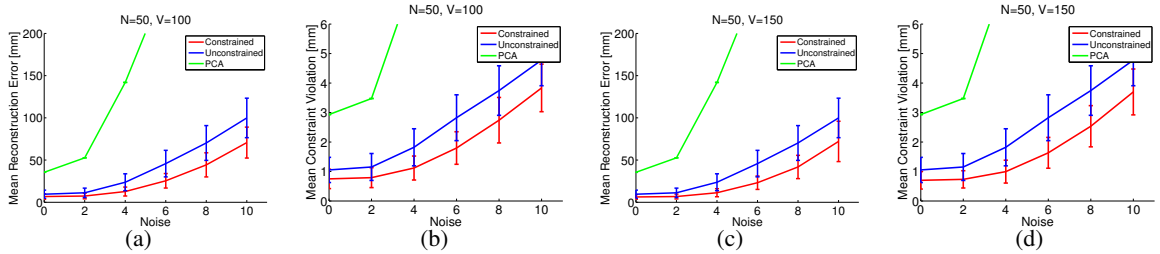


Figure 3. **Reconstructing a piece of cardboard from synthetic data.** (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using equality constraints.

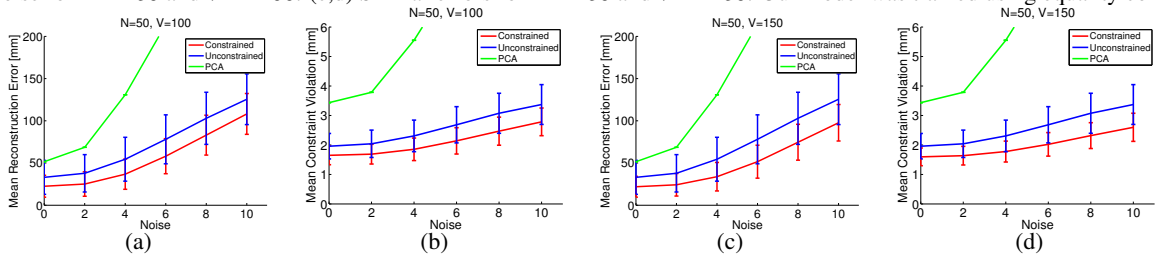


Figure 4. **Reconstructing a piece of cloth from synthetic data.** (a,b) Reconstruction error and constraint violation as a function of image noise for $N = 50$ and $V = 100$. (c,d) Similar errors for $N = 50$ and $V = 150$. Our model was trained using inequality constraints.

prove efficiency, we can also rely on different strategies to account for constraints. Summing constraints, as described above, is one such strategy. Another one consists in adding the most violated constraints to a set of active constraints at each learning iteration. Training times for these different strategies are given in Fig. 2(right). Note that summing constraints decreases the training time dramatically since it effectively reduces the number of constraints for the same number of unlabeled samples. Iteratively adding constraints to an active set also noticeably reduces training time.

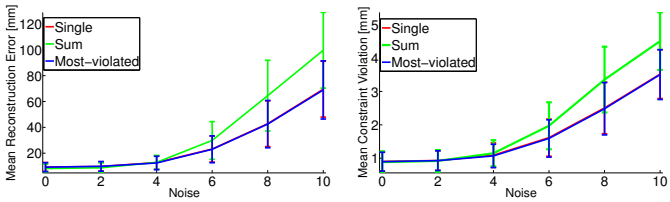


Figure 5. **Reconstructing a piece of cardboard using different constraint selection strategies.** Reconstruction error and constraint violation as a function of noise for $N = 100$ and $V = 100$. The curves for the single and most-violated strategies overlap.

4.2. Synthetic Data

We first used the two motion capture datasets to generate synthetic data. To this end, we sampled the barycentric coordinates of the ground-truth meshes and projected the resulting 3D points with a known camera, thus creating 2D image measurements. We then added Gaussian noise with standard deviation ranging between 0 and 10 to these measurements. At test time, we optimized the latent variables, as well as the global rotation and translation, so that the predicted 3D shape minimizes the reprojection error with respect to the noisy image measurements. For both datasets, we learned the models with $N = 50$ labeled examples and either $V = 100$ or $V = 150$ unlabeled ones, and tested them on 300 samples. For each noise value, we used 5 different train/test partitions. Figs. 3 and 4 depict errors as a function of the image noise standard deviation for the cardboard and cloth datasets, respectively. Note that our constrained model consistently outperforms PCA and the unconstrained model for both reconstruction error and constraint violation. Error bars on the plots represent ± 1 standard deviation over the 5 different partitions. Note that the PCA model was learned from all the data, and therefore does not depend on the partition. This remains a valid comparison, since only the baseline has access to more data. Fig. 5 depicts similar errors for our different constraint selection strategies. Note that, while being faster to train, summing constraints yields less accurate reconstructions for a given V.

4.3. Real Images with Ground-truth

To evaluate our model's accuracy in realistic conditions, we performed experiments where the images were captured with a Microsoft Kinect, which also provides us with ground-truth 3D information, only used for evaluation purposes. We captured deformations of two different materials: a piece of paper and a t-shirt. As before, we used reprojection error as image loss, but used SIFT to compute the correspondences between a reference image in which the 3D shape is known and the other images of the sequence. For both materials, we learned the models with $N = 20$ labeled and $V = 50$ unlabeled examples from the cardboard dataset, and, as before, used 5 different training sets. Fig. 6 depicts images of both sequences with our reconstructions. Tables 2 and 3 show errors averaged over the frames of the sequence when using either equality, or inequality constraints. Here, reconstruction error was computed between the predicted 3D locations of the feature points and their ground-truth Kinect locations. As with synthetic data, our model outperforms the baselines for both constraint types.

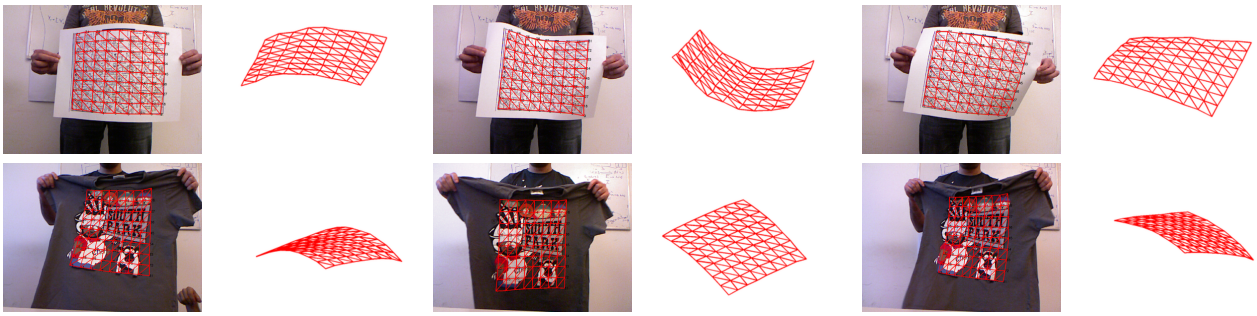


Figure 6. **Real images with ground-truth.** (Top) Images of a deforming piece of paper with reconstructed meshes seen from a different viewpoint. (Bottom) Similar images for a deforming t-shirt. In both cases, we show our reconstructions reprojected on the images.

	Equality		Inequality	
	Reconstr. Error [mm]	Constraint Viol. [mm]	Reconstr. Error [mm]	Constraint Viol. [mm]
PCA	11.68 ± 0.00	1.71 ± 0.00	11.68 ± 0.00	2.09 ± 0.00
Unconstrained	9.35 ± 1.03	1.10 ± 0.23	9.35 ± 1.03	0.96 ± 0.09
Ours	$\mathbf{7.23 \pm 0.76}$	$\mathbf{0.78 \pm 0.03}$	$\mathbf{8.03 \pm 0.56}$	$\mathbf{0.71 \pm 0.13}$

Table 2. **Reconstructing a piece of paper.** Reconstruction error was computed with respect to the Kinect ground-truth.

	Equality		Inequality	
	Reconstr. Error [mm]	Constraint Viol. [mm]	Reconstr. Error [mm]	Constraint Viol. [mm]
PCA	18.44 ± 0.00	0.97 ± 0.00	18.44 ± 0.00	0.84 ± 0.00
Unconstrained	15.50 ± 1.78	0.92 ± 0.22	15.50 ± 1.78	0.75 ± 0.14
Ours	$\mathbf{14.79 \pm 0.84}$	$\mathbf{0.73 \pm 0.05}$	$\mathbf{14.35 \pm 0.90}$	$\mathbf{0.60 \pm 0.07}$

Table 3. **Reconstructing a deforming t-shirt.** Reconstruction error was computed with respect to the Kinect ground-truth.

4.4. Real Images without Ground-truth

For qualitative evaluation, we also applied our model to reconstructing two sequences of deforming cloth surfaces[3]. In both cases, we used $N = 500$ labeled examples. Since the deformations in the first sequence are relatively simple, we could use a small number of unlabeled examples (V=150), and thus exploit all individual edge equality constraints when learning our model. For the second sequence, which contains more complex deformations, we used sums of constraints which let us employ more unlabeled examples (V=1500). In both experiments, the image-based loss was taken as the normalized cross-correlation between the texture under the optimized mesh and the texture in a reference image. In Fig. 7, we compare our results with those obtained with the baselines for the first sequence. Note that the shapes reconstructed with our model better correspond to the ones in the images. Fig. 8 depicts our reconstructions on the second sequence. For reasons of space, we do not include the baselines' results for this sequence. However, we encourage the reader to look at the full comparison in the videos given as supplementary material. Since we have no ground-truth for these sequences, we can only evaluate constraint violation. Fig. 9 depicts this error for all the frames in the sequences. Note that our method clearly outperforms the baselines in terms of constraint satisfaction.

5. Conclusion

In this paper, we have introduced a constrained latent variable model that encodes prior knowledge about the desired output in the form of equality and inequality constraints. We have shown that our approach can be kernelized, thus allowing for the use of non-linear kernels that have been proven effective in many computer vision tasks. Using both synthetic and real data, we have demonstrated that our model outperforms the commonly-employed ones for the purpose of monocular 3D surface reconstruction, which is such an ambiguous problem that using constraints effectively is a requirement for success. Furthermore, our formalism is extremely general. In future work, we therefore plan to extend our approach to different loss functions, regularizers, and constraints, in order to apply our model to a wide range of computer vision problems.

[3] Publicly available at http://cvlab.epfl.ch/data/dsr/

References

[1] M. Alvarez, D. Luengo, and N. Lawrence. Latent force models. In *AISTATS*, 2009. 2

[2] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popovic, and S. M. Seitz. Estimating Cloth Simulation Parameters from Video. In *SCA*, 2003. 2

[3] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, 1999. 1, 2

[4] M. Brand. Morphable 3D Models from Video. *CVPR*, 2001. 2

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *CVPR*, 2000. 1, 2

[6] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces. In *ACCV*, 2010. 2

[7] A. Ecker, A. Jepson, and K. Kutulakos. Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities. In *ECCV*, 2008. 2

[8] J. Fayad, L. Agapito, and A. Del Bue. Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences. In *ECCV*, 2010. 2

[9] J. Fayad, A. Del Bue, L. Agapito, and P. M. Q. Aguiar. Non-Rigid Structure from Motion Using Quadratic Deformation Models. In *BMVC*, 2009. 1, 2

[10] R. Hartley and R. Vidal. Perspective Nonrigid Shape and Motion Recovery. In *ECCV*, 2008. 2

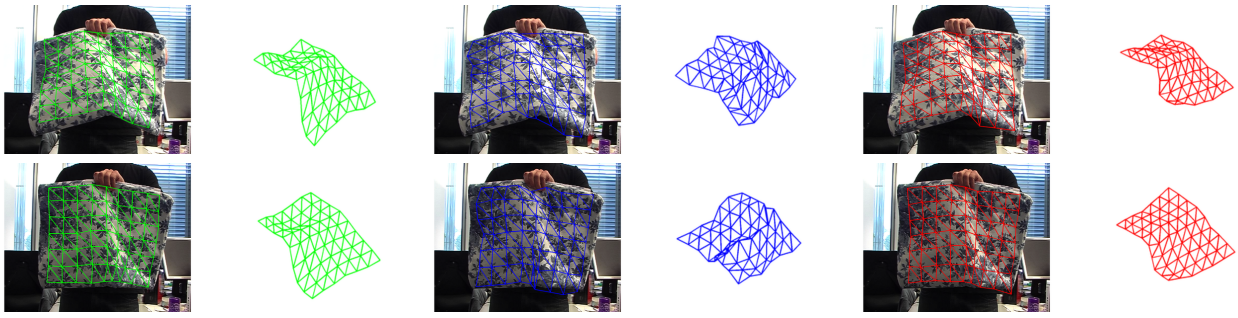


Figure 7. **Reconstructing a deforming bed sheet.** Comparison of our reconstructions with those of two baselines for two frames of the sequence. In each frame, we show the reconstructed mesh reprojected on the original image, as well as a side view of the mesh.

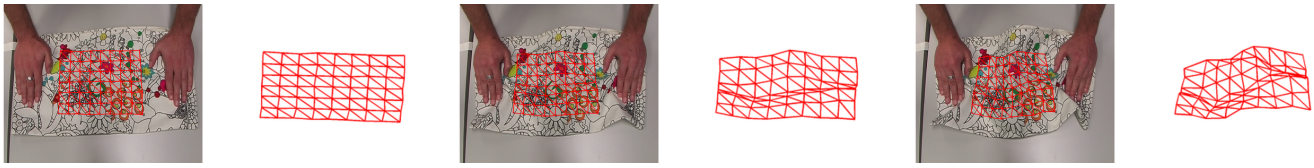


Figure 8. **Reconstructing a deforming cushion cover.** Reconstructions obtained with our model for 3 frames reprojected on the original image, and seen from a different viewpoint.

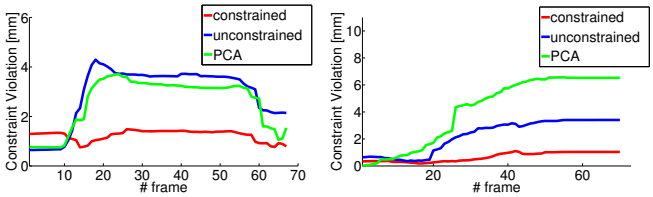


Figure 9. **Constraint violation.** Comparison of our model with two baselines for (a) the bed sheet and (b) the cushion sequences.

[11] N. D. Lawrence. Probabilistic Non-Linear Principal Component Analysis with Gaussian Process Latent Variable Models. *JMLR*, 2005. 1, 2

[12] X. Llado, A. Del Bue, and L. Agapito. Non-Rigid Metric Reconstruction from Perspective Cameras. *IVC*, 2010. 2

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative Learned Dictionaries for Local Image Analysis. In *CVPR*, 2008. 1, 2

[14] T. Mcinerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *ICCV*, 1993. 2

[15] D. Metaxas and D. Terzopoulos. Constrained Deformable Superquadrics and Nonrigid Motion Tracking. *PAMI*, 1993. 2

[16] C. Nastar and N. Ayache. Frequency-Based Nonrigid Motion Analysis. *PAMI*, 1996. 2

[17] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Shape Modeling and Recognition. *PAMI*, 1991. 2

[18] M. Perriollat, R. Hartley, and A. Bartoli. Monocular Template-Based Reconstruction of Inextensible Surfaces. *IJCV*, 2010. 2

[19] V. Rabaud and S. Belongie. Re-Thinking Non-Rigid Structure from Motion. In *CVPR*, 2008. 1, 2

[20] M. Salzmann, R. Urtasun, and P. Fua. Local Deformation Models for Monocular 3D Shape Recovery. In *CVPR*, 2008. 1, 2

[21] M. Salzmann and R. Urtasun Implicitly Constrained Gaussian Process Regression for Monocular Non-Rigid Pose Estimation In *NIPS*, 2010. 2

[22] M. Salzmann and P. Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. *PAMI*, 2011. 2, 3, 5

[23] S. Shen, W. Shi, and Y. Liu. Monocular 3D Tracking of Inextensible Deformable Surfaces Under L2-Norm. In *ACCV*, 2009. 2

[24] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic Tracking of 3D Human Figures Using 2D Image Motion. In *ECCV*, 2000. 1, 2

[25] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid Structure-From-Motion: Estimating Shape and Motion with Hierarchical Priors. *PAMI*, 2008. 2

[26] L. Tsap, D. Goldgof, and S. Sarkar. Nonrigid Motion Analysis Based on Dynamic Refinement of Finite Element Models. *PAMI*, 2000. 2

[27] R. Urtasun, D. Fleet, A. Hertzman, and P. Fua. Priors for People Tracking from Small Training Sets. In *ICCV*, 2005. 1, 2

[28] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *CVPR*, 2006. 2

[29] R. Urtasun, D. Fleet, A. Geiger, and J. Popović, T. Darrell, and N. Lawrence. Topologically-Constrained Latent Variable Models. In *ICML*, 2008. 2

[30] J. Wang, D. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. In *NIPS*, 2005. 2

[31] J. Xiao and T. Kanade. Uncalibrated Perspective Reconstruction of Deformable Structures. In *ICCV*, 2005. 2

[32] J. Yang, K. Yu, and T. Huang. Efficient Highly Over-Complete Sparse Coding using a Mixture Model. In *ECCV*, 2010. 1, 2