

Monocular Template-Based 3D Surface Reconstruction: Convex Inextensible and Nonconvex Isometric Methods

F. Brunet A. Bartoli R. I. Hartley
Ubleam ISIT, UMR 6284 CNRS-UdA ANU-NICTA
Toulouse, France Clermont-Ferrand, France Canberra, Australia

Corresponding author: Adrien Bartoli, Adrien.Bartoli@gmail.com

Abstract

We study the 3D reconstruction of an isometric surface from point correspondences between a template and a single input image. The template shows the surface flat and fronto-parallel. We propose three new methods. The first two use a convex relaxation of isometry to inextensibility. They are formulated as Second Order Cone Programs (SOCP). The first proposed method is point-wise (it reconstructs only the input point correspondences) while the second proposed method uses a smooth and continuous surface model, based on Free-Form Deformations (FFD). The third proposed method uses the ‘true’ nonconvex isometric constraint and the same continuous surface model. It is formulated with Nonlinear Least-Squares and can thus be solved with the efficient Levenberg-Marquardt minimization method. The proposed approaches may be combined in a single pipeline whereby one of the convex approximations is used to initialize the nonconvex method. Our contributions solve two important limitations of current state of the art: our convex methods are the first ones to handle noise in both the template and image points, and our nonconvex method is the first one to use ‘true’ isometric constraints. Our experimental results on simulated and real data show that our convex point-wise method and our nonconvex method outperform respectively current initialization and refinement methods in 3D reconstructed surface accuracy.

Code release. We intend to release our code under the GPL license.

Research highlights:

- A convex SOCP based method with noise in template and image
- The use of a continuous surface model based on an FFD
- A nonconvex, statistically founded, and truly isometric reconstruction method

1 Introduction

Monocular 3D surface reconstruction of deformable objects is a challenging problem which has gained interest over the past decade. Without additional priors, this is an ill-posed problem since there is an infinite number of deformable 3D surfaces with exactly the same projection. In this paper, we study monocular 3D reconstruction of *deformable but isometric surfaces* from a single input image and under two common assumptions [21, 26, 17]. First, we assume that point correspondences are available between the input image and a template image; the latter shows the surface flat and fronto-parallel. Second, we assume that the camera follows the pin-hole model with known intrinsic parameters [13]. This type of reconstruction algorithms may be used for thin-shell objects which may be physically flattened and which smoothly deform without stretching. Examples of such objects and deformations are a sheet of paper being gently bent, a flag

waving in the wind and a piece of cloth on a moving body. However, non physically flattening objects such as faces, even if they deform quasi-isometrically, cannot be handled by this type of algorithms.

Different types of constraints have been proposed to disambiguate the problem of monocular 3D reconstruction of deformable surfaces. These constraints can be divided into two main categories: *statistical constraints* and *physical constraints*. Constraints coming from both categories may of course be combined [1, 7]. Statistical constraints include the very popular low-rank shape model applicable to template-free 3D reconstruction and leading to nonrigid factorization [4, 1, 3, 7, 15, 27, 28]. This constraint has been used in the template-based case by a priori learning a typical shape basis [10, 22, 24, 21]. Physical constraints include temporal and spatial 3D surface priors [12, 19] and isometric deformations [23, 26, 21, 17].

Methods for monocular 3D reconstruction of deformable surfaces can be categorized by type of constraints but also by type of 3D surface model. The *point-wise surface model* uses a sparse set of 3D points, each of which generally associated to a known point correspondence [17]. The *mesh-based surface model* uses a set of connected, typically triangular facets [23, 21]. The *continuous surface model* uses a smooth surface embedding which maps the template space to 3D [17, 1].

The isometric deformation constraint is widely applicable and stable for 3D reconstruction. For instance, it approximates well the deformation of materials such as paper and fabric [23, 26, 21, 17]. An isometric deformation preserves geodesic distances on the 3D surface across time. However, computing geodesics is a hard problem; it is consequently even harder to constrain their length in a 3D reconstruction algorithm. Two main ways have been proposed to implement the isometry constraint. The first way is the *inextensibility relaxation* that leads to convex inequality constraints on the 3D points' coordinates. It upper bounds the Euclidean distance by the template geodesic distance for pairs of 3D points [23, 21, 26, 17]. The second way is the *Euclidean approximation* that leads to quartic constraints on the 3D points' coordinates. It preserves the Euclidean distance instead of the geodesic distance for pairs of closely spaced 3D points [25].

Current algorithms' limitations and proposed contributions. This paper proposes three new methods to improve template-based 3D reconstruction with the isometric deformation constraint. It extends our conference paper [6].

First, the current literature misses a convex algorithm solving the inextensibility relaxation and handling noise in both the template and the input image. We show that the maximum depth heuristic [17] with the inextensibility relaxation can be cast as an SOCP [2]. While previous work use iterative local optimization and model noise in the template image only [17] or in the input image only [23, 21, 26], our formulation models noise in both images. Our first implementation of this formulation uses a point-wise surface model; our second implementation uses a continuous surface model. In the latter case, we propose using an FFD with a compact B-spline kernel as opposed to the dense TPS kernel [17, 1].

Second, the current literature misses a formulation of the isometric deformation constraint adapted to nonconvex optimization. We show that using a continuous surface model allows one to gracefully formulate the isometric deformation constraint as a Nonlinear Least-Squares cost function on the unknown continuous surface model.

We report extensive experimental results on simulated and real data. They show that our point-wise convex inextensibility method outperforms current initialization methods in terms of 3D reconstruction accuracy, but that directly including the continuous surface model degrades accuracy. Our nonconvex isometric method gives the most accurate results of all compared methods by a substantial margin. We give a careful analysis of these results and draw general conclusions.

Paper organization. We review related work in §2. We present our embedding-based continuous surface modeling in §3. We give our convex methods using inextensibility in §4 and our nonconvex method in §5. We report experimental results in §6 and conclude in §7.

Notation. We write scalars in italics, vectors in bold and matrices in sans-serif fonts. $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ contains the camera's intrinsic parameters (the camera lies at the world coordinate frame's origin) and \mathbf{p}_k^\top is the k th row of matrix \mathbf{K} . The n point correspondences are written $(\mathbf{q}_i, \mathbf{q}'_i)$ with $i \in \{1, \dots, n\}$ the point index, $\mathbf{q}_i \in \mathbb{R}^2$ is the template point and $\mathbf{q}'_i \in \mathbb{R}^2$ the input image point. The homogeneous coordinates of point \mathbf{q}_i are written $\bar{\mathbf{q}}_i \in \mathbb{R}^3$. The sightline corresponding to point \mathbf{q}'_i is given by $\mathbf{u}_i = \frac{\mathbf{K}^{-1}\bar{\mathbf{q}}'_i}{\|\mathbf{K}^{-1}\bar{\mathbf{q}}'_i\|}$, where $\|\cdot\|$

stands for the two-norm of a vector, matrix or tensor. The 3D reconstructed surface is written $\mathcal{S} \subset \mathbb{R}^3$. The depth of the i th reconstructed 3D point $\mathbf{Q}_i \in \mathcal{S}$ is written $\mu_i \in \mathbb{R}_+$. The geodesic distance between two points $\mathbf{A} \in \mathcal{S}$ and $\mathbf{B} \in \mathcal{S}$ as measured on \mathcal{S} is written $d_G(\mathbf{A}, \mathbf{B})$, and the Euclidean distance is written $d_E(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|$. The ‘reference’ geodesic distance easily measured from the flat fronto-parallel template image is written $d_{\mathbf{A}, \mathbf{B}}$. For the i th and j th point in particular it is written $d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$. \hat{x} means the true value of x (for $x \in \{\mathbf{q}'_i, \mathbf{q}_i, \mathbf{Q}_i, \mathbf{u}_i, \mu_i, d_{ij}\}$).

2 Related Work

We review existing work on template-based isometric deformable 3D surface reconstruction. Mathematically, the *isometry constraint* can be stated as:

$$d_G(\mathbf{A}, \mathbf{B}) = d_{\mathbf{A}, \mathbf{B}} \quad \forall \mathbf{A} \in \mathcal{S} \quad \forall \mathbf{B} \in \mathcal{S}. \quad (1)$$

This constraint is nonconvex and has not received a formulation that easily fits in optimization since the geodesic distance $d_G(\mathbf{A}, \mathbf{B})$ on the deformed surface \mathcal{S} does not have a general closed-form expression. Two main formulations can be found in the literature: the inextensibility relaxation and the Euclidean approximation.

The convex inextensibility relaxation. The principle that underlies the convex inextensibility relaxation [21, 17] is simple: the Euclidean distance between two points on an isometrically deforming surface cannot grow larger than the geodesic distance. The isometry constraint (1) can thus be relaxed to the following *inextensibility constraint*:

$$d_E(\mathbf{A}, \mathbf{B}) \leq d_{\mathbf{A}, \mathbf{B}} \quad \forall \mathbf{A} \in \mathcal{S} \quad \forall \mathbf{B} \in \mathcal{S}. \quad (2)$$

It is worth noting that (1) \Rightarrow (2) but that the opposite is not true, which is why the inextensibility constraint (2) is a relaxation of the isometry constraint (1). This relaxation applied to the set of point correspondences was used in conjunction with the *maximum depth heuristic*. This heuristic consists in forcing each 3D point \mathbf{Q}_i to lie on the sightline \mathbf{u}_i by setting $\mathbf{Q}_i = \mu_i \mathbf{u}_i$ and maximizing the depth of the reconstructed 3D points by an ad hoc iterative algorithm [17]. This algorithm accounts for noise in the template image by slightly overestimating the geodesic distances d_{ij} , replacing them by $d_{ij} + \varepsilon_\tau$ where ε_τ is an upper bound on the error made in the template point positions. No proof of convergence was given for this algorithm, though in practice it seems to always converge in a few iterations. Another implementation of the inextensibility constraint with the maximum depth heuristic was proposed in [23, 21]. This implementation solves an SOCP where the cost function combines the depth of the reconstructed points and the negative reprojection error is maximized. It uses a mesh-based surface model and applies the inextensibility constraint to the mesh vertices. This cost function models noise in the input image point positions but not in the template point positions. A similar approach was explored in [26].

The nonconvex Euclidean approximation. The Euclidean approximation replaces the geodesic distance in the isometry constraint (1) by the Euclidean distance for a set of closely spaced point pairs [22, 23, 29, 26]:

$$d_E(\mathbf{A}, \mathbf{B}) \approx d_{\mathbf{A}, \mathbf{B}} \quad \forall \mathbf{A} \in \mathcal{S} \quad \forall \mathbf{B} \in \mathcal{S} \quad d_{\mathbf{A}, \mathbf{B}} \leq \epsilon, \quad (3)$$

where $\epsilon \in \mathbb{R}_+$ is a small constant. Note that enforcing $d_E(\mathbf{A}, \mathbf{B}) = d_{\mathbf{A}, \mathbf{B}}$ in the Euclidean approximation (3) would not allow the surface to deform at all, which is why $d_E(\mathbf{A}, \mathbf{B}) \approx d_{\mathbf{A}, \mathbf{B}}$ is used instead. In practice, the constraint is embedded as a nonconvex penalty in a compound cost function, and generally leads to a quartic cost term. Although this approximation works well in some cases it performs poorly when the surface bends too strongly. Note that the Euclidean approximation (3) is a stronger constraint than the inextensibility relaxation (2), though the former does not imply the latter.

Summary. To summarize, existing work lacks two main important pieces. First, it lacks a convex solution which would handle noise in both the template and input images. Second, it lacks a simple formulation of the isometric constraint which could be easily used in a nonconvex optimization program. A direct consequence is that no algorithm was proposed to exploit the ‘true’ isometric constraints. Our contributions address these shortcomings.

3 Parametric Surface Modeling

We propose to use the template space $\Omega \subset \mathbb{R}^2$ as a parameterization space for the surface \mathcal{S} to be reconstructed. This is in contrast to almost all previous works, which merely estimate the depth of a small set of point correspondences from the input image. This way, we are able to solve minimization problems over a set of continuous surfaces. We parameterize the set of admissible surfaces by Free-Form Deformations (FFD) [20] based on uniform cubic B-splines [8]. In practice, this amounts to defining an embedding $\mathcal{W}_\ell : \Omega \rightarrow \mathcal{S}$ that maps the template space to the unknown surface as an FFD, thus parametrized by a grid of $n_u \times n_v$ 3D control points ℓ_{jk} with $j \in \{1, \dots, n_u\}$ and $k \in \{1, \dots, n_v\}$, which act as ‘attractors’ for the surface. For a template point $\mathbf{q} = (u, v)^\top$ the surface point is explicitly given as:

$$\mathcal{W}_\ell(\mathbf{q}) = \sum_{j=1}^{n_u} \sum_{k=1}^{n_v} \ell_{jk} N_j(u) N_k(v). \quad (4)$$

The functions N_j are the B-spline basis functions [8] which are cubic polynomials. If point $\mathbf{q}_i = (u_i, v_i)^\top$ is fixed and known then the surface point $\mathcal{W}_\ell(\mathbf{q}_i)$ is expressed as a linear combination of the points ℓ_{jk} , and hence can be written in the form $\mathcal{W}_\ell(\mathbf{q}_i) = \mathbf{W}_i \ell$, where \mathbf{W}_i is a $3 \times 3p$ matrix ($p = n_u n_v$) depending only on point \mathbf{q}_i , and ℓ is the vector obtained by concatenating all the control points ℓ_{jk} . Using an FFD parameterization has three major advantages:

- **Linear continuous surface model.** A 3D point is given as a linear function of the control point vector ℓ . This is especially important in our application, where we will show how to directly reconstruct a continuous surface with a convex formulation.
- **Local control.** The control points have a bounded (local) influence. The polynomials N_j have a local support in cubic B-splines; the matrix \mathbf{W}_i is thus sparse, which is important for computational efficiency.
- **Analytical convex smoothing expression.** The surface’s bending energy, which we use as a smoothing term, has a convex, Linear Least-Squares expression, in terms of the control points. This is again important to facilitate a convex formulation (see below).

We define our smoothing term using the bending energy as:

$$\mathcal{E}_s(\ell) = \sum_{m=1}^3 \int_{\Omega} \left\| \frac{\partial^2 \mathcal{W}_\ell^m}{\partial \mathbf{q}^2} \right\|^2 d\mathbf{q} \quad (5)$$

where $\mathcal{W}_\ell^m(\mathbf{q})$ is the m th coordinate of the point. With the FFD, there exists a simple and Linear Least-Squares closed-form expression for the bending energy:

$$\mathcal{E}_s(\ell) = \|\mathbf{B}^{1/2} \ell\|^2 = \ell^\top \mathbf{B} \ell \quad (6)$$

where $\mathbf{B} \in \mathbb{R}^{3p \times 3p}$ is a symmetric, positive, and semi-definite matrix which can be easily computed from the second derivatives of the B-spline basis functions [5].

4 Convex Inextensible Methods

We propose a convex formulation of the inextensibility relaxation with the maximum depth heuristic that, as opposed to [17, 23, 21, 26], accounts for noise in both the template and the input images. As in [23, 21], our approach is formulated as an SOCP, but it does not require one to tune the relative influence of the negative reprojection error and the depths in the cost function.

4.1 Prerequisite: A Convex Formulation of Template Noise

We first formulate the inextensibility relaxation with the maximum depth heuristic as an SOCP. This first formulation is point-wise: it only reconstructs the point correspondences to \mathbf{Q}_i , using them via their sightlines \mathbf{u}_i , $i \in \{1, \dots, n\}$. Importantly, noise is only accounted for in the template image. For any pair of points, the inextensibility constraint $\|\mathbf{Q}_i - \mathbf{Q}_j\| \leq d_{ij} + \varepsilon_\tau$ can be written:

$$\|\mu_i \mathbf{u}_i - \mu_j \mathbf{u}_j\| \leq d_{ij} + \varepsilon_\tau. \quad (7)$$

The maximum depth heuristic leads to the following SOCP:

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \sum_{i=1}^n \mu_i \\ \text{subject to} \quad & \|\mu_i \mathbf{u}_i - \mu_j \mathbf{u}_j\| \leq d_{ij} + \varepsilon_\tau \quad \forall (i, j) \in \mathcal{E} \\ & \mu_i \geq 0 \quad i \in \{1, \dots, n\} \end{aligned} \quad (8)$$

where $\boldsymbol{\mu}^\top = (\mu_1 \dots \mu_n)$, and $\mathcal{E} \subset \{1, \dots, n\}^2$ is a set of pairs of points to which the inextensibility constraint is applied. This is merely a reformulation of [17] as an SOCP. It thus shows that this formulation is convex, and does not require an iterative ad hoc minimization algorithm to be solved, as was proposed in [17].

4.2 A Convex Formulation of Template and Input Image Noise

We now suppose that the inaccuracies are expressed in terms of image measurements.

4.2.1 Point-Wise Surface Model

Suppose that points are measured in the image with a maximum error of $\varepsilon_\mathcal{I}$. This leads to:

$$\|\hat{\mathbf{q}}'_i - \mathbf{q}'_i\| \leq \varepsilon_\mathcal{I}, \quad \forall i \in \{1, \dots, n\}. \quad (9)$$

Since we are searching for the true 3D point \mathbf{Q}_i we can write:

$$\hat{\mathbf{q}}'_i = \frac{1}{\mathbf{p}_3^\top \mathbf{Q}_i} \begin{pmatrix} \mathbf{p}_1^\top \mathbf{Q}_i \\ \mathbf{p}_2^\top \mathbf{Q}_i \end{pmatrix}. \quad (10)$$

Equation (9) can thus be rewritten as:

$$\left\| \frac{1}{\mathbf{p}_3^\top \mathbf{Q}_i} \begin{pmatrix} \mathbf{p}_1^\top \mathbf{Q}_i \\ \mathbf{p}_2^\top \mathbf{Q}_i \end{pmatrix} - \mathbf{q}'_i \right\| \leq \varepsilon_\mathcal{I}. \quad (11)$$

Adding the inextensibility constraints and the maximization of the depths (which are given by $\mathbf{p}_3^\top \mathbf{Q}_i$) and we arrive at the following SOCP problem:

$$\begin{aligned} \max_{\mathbf{Q}} \quad & \mathbf{p}_3^\top \sum_{i=1}^n \mathbf{Q}_i \\ \text{subject to} \quad & \left\| \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \end{bmatrix} \mathbf{Q}_i - \mathbf{q}'_i \mathbf{p}_3^\top \mathbf{Q}_i \right\| \leq \varepsilon_\mathcal{I} \mathbf{p}_3^\top \mathbf{Q}_i \quad \forall i \in \{1, \dots, n\} \\ & \|\mathbf{Q}_i - \mathbf{Q}_j\| \leq d_{ij} + \varepsilon_\tau \quad \forall (i, j) \in \mathcal{E} \\ & \mathbf{p}_3^\top \mathbf{Q}_i \geq 0 \quad \forall i \in \{1, \dots, n\} \end{aligned} \quad (12)$$

where \mathbf{Q} is the concatenation of the 3D points \mathbf{Q}_i , for $i \in \{1, \dots, n\}$. This is the first proposed convex formulation which handles noise both in the template via ε_τ and in the image via $\varepsilon_\mathcal{I}$.

4.2.2 Continuous Surface Model

We extend the above formulation (12) to incorporate the FFD-based continuous surface model proposed in §3, expressing \mathbf{Q}_i in terms of the control points $\boldsymbol{\ell}$, according to $\mathbf{Q}_i = \mathbf{W}_i \boldsymbol{\ell}$. We show that one may solve for $\boldsymbol{\ell}$ directly using SOCP. As in the previous approach, the linear smoothing term may be included in the estimation scheme. It results in the following SOCP formulation:

$$\begin{aligned}
& \max_{\boldsymbol{\ell}, r} && \mathbf{p}_3^\top \sum_{i=1}^n \mathbf{W}_i \boldsymbol{\ell} - r \\
\text{subject to} &&& \left\| (\mathbf{W}_i \boldsymbol{\ell})^\top \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix} - (\mathbf{W}_i \boldsymbol{\ell})^\top \mathbf{p}_3 \mathbf{q}'_i{}^\top \right\| \leq \varepsilon_x \mathbf{W}_i \boldsymbol{\ell} \mathbf{p}_3 && \forall i \in \{1, \dots, n\} \\
&&& \|(\mathbf{W}_i - \mathbf{W}_j) \boldsymbol{\ell}\| \leq d_{ij} + \varepsilon_\tau && \forall (i, j) \in \mathcal{E} \\
&&& (\mathbf{W}_i \boldsymbol{\ell})^\top \mathbf{p}_3 \geq 0 && \forall i \in \{1, \dots, n\} \\
&&& \lambda \|\mathbf{B}^{1/2} \boldsymbol{\ell}\| \leq r
\end{aligned} \tag{13}$$

where λ is the regularization weight (set to $\lambda = 10^{-4}$ in our experiments since the surface is already regularized by inextensibility, see the discussion in §5.4) and r is an additional variable added to the SOCP problem for minimizing the regularization term. Compared to the previous approach, this one reduces the number of steps to get an initial solution since no surface fitting is required as a post-processing. The reprojection error is now a hard constraint that the FFD must satisfy.

5 A Nonconvex Isometric Method

Although the maximum depth heuristic used in the previous section gives reasonable initialization results, it is merely a heuristic, not based on a valid principle related to surface properties. We therefore consider next a new formulation based on the ‘true’ isometry constraints. We start by presenting the principles underlying the 3D reconstruction of a continuous surface model under the isometric assumption expressed by equation (1). We then present an estimation method based on point correspondences and our FFD-based surface model. In particular, we will see that this leads to a Nonlinear Least-Squares problem for the continuous surface model’s control points.

5.1 Using ‘True’ Isometry Constraints

Let the surface be modelled as a function $\mathcal{W} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, mapping the planar template to 3-dimensional space. The inextensibility constraint is equivalent to having the continuous surface model \mathcal{W} to be a local isometry at every point over its domain. This condition may be expressed in terms of its Jacobian matrix. Let $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{3 \times 2}$ be the Jacobian matrix $\frac{\partial \mathcal{W}}{\partial \mathbf{q}}$ evaluated at point \mathbf{q} . The map \mathcal{W} is an isometry at \mathbf{q} if the columns of $\mathbf{J}(\mathbf{q})$ are orthonormal. This local isometry can be enforced for the whole surface with the following Nonlinear Least-Squares constraint:

$$\int_{\Omega} \|\mathbf{J}(\mathbf{q})^\top \mathbf{J}(\mathbf{q}) - \mathbf{I}_2\|^2 d\mathbf{q} = 0. \tag{14}$$

In practice, we discretize equation (14) as:

$$\mathcal{E}_i(\mathcal{W}) = \sum_{j=1}^{n'} \|\mathbf{J}(\mathbf{g}_j)^\top \mathbf{J}(\mathbf{g}_j) - \mathbf{I}_2\|^2, \tag{15}$$

where $\{\mathbf{g}_j\}_{j=1}^{n'}$ is a dense set of n' 2D points in the template image space taken on a fine and regular grid (for instance, a grid of size 30×30). Note that the continuous surface model’s partial derivatives are computed analytically [5]. This term $\mathcal{E}_i(\mathcal{W})$ measures the departure from isometry of the continuous surface model \mathcal{W} . Our problem is then to minimize this quantity, over all possible surfaces, subject to the projection constraints, namely that point $\mathcal{W}(\mathbf{q}_i)$ projects close to the image point \mathbf{q}'_i , $i \in \{1, \dots, n\}$.

5.2 General Estimation Scheme

The general estimation scheme we use for implementing the isometry constraints consists in minimizing a cost function of the following form:

$$\min_{\boldsymbol{\mu}, \boldsymbol{\ell}} \mathcal{E}_d(\boldsymbol{\mu}, \boldsymbol{\ell}) + \alpha \mathcal{E}_i(\boldsymbol{\ell}) + \beta \mathcal{E}_s(\boldsymbol{\ell}), \quad (16)$$

where \mathcal{E}_d , \mathcal{E}_i , \mathcal{E}_s are the *data*, *isometry*, and *smoothing* terms respectively. The data term ensures the consistency of the reconstructed surface with the data. It is derived in the next section. The isometry term has been discussed directly above. The smoothing term promotes surfaces whose curvature is smallest, in order to cope with, for instance, a lack of point correspondences. A more detailed discussion is provided in the section after next. The relative influence of these three terms is controlled with the weights $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$. In practice, α weighs the relative influence of the isometric deformation prior. Its optimal value is thus application dependent, since a large value will cause the reconstructed surface to be almost exactly applicable to the template, while a smaller value will relax isometry and let the surface to locally stretch. In our experiments, we have chosen to favor strongly isometric surfaces, since handling this type of surfaces has been our main goal in this paper, and set $\alpha = 10^6$. A discussion on how to choose the value of β is provided in §5.4.

As a consequence of the above discussion, all terms to be minimized are Linear or Nonlinear Least-Squares. We thus use the Levenberg-Marquardt algorithm to solve the minimization problem. This algorithm has proven extremely efficient for solving general geometric 3D computer vision problems [13]. Being iterative it requires an initial estimate, which can be directly provided by one of our convex algorithms as discussed in §5.5.

5.3 The Data-Term

We give an estimation scheme for smooth surfaces that uses point correspondences. Instead of considering a hard upper-bounded constraint, as required by our SOCP approach in §4.2.2, which would require constrained Nonlinear Least-Squares optimization, we include the reprojection error into the cost function, leading to an unconstrained problem.

To simplify the formulation of the reprojection error, we introduce the depths μ_i as ancillary variables, for reasons that shall become clear shortly. This is not strictly necessary, but this reduces the degree of the reprojection error term and convexifies the overall cost function. Replacing \mathbf{Q}_i by $\mathbf{W}_i \boldsymbol{\ell}$ and minimizing the left-hand side of equation (11) gives an expression for the reprojection error associated with point $i \in \{1, \dots, n\}$. However, the resulting expression is nonlinear with respect to the parameters $\boldsymbol{\ell}$. A linear reformulation is obtained by introducing the depths $\boldsymbol{\mu}$ of the data points in the optimization problem, leading to the following expression of the data term:

$$\mathcal{E}_d(\boldsymbol{\mu}, \boldsymbol{\ell}) = \sum_{i=1}^n \left\| \mathcal{W}_{\boldsymbol{\ell}}(\mathbf{q}_i) - \mu_i \mathbf{K}^{-1} \bar{\mathbf{q}}'_i \right\|^2, \quad (17)$$

which measures the distance between the point $\mathcal{W}_{\boldsymbol{\ell}}$ on the surface and the point at depth μ_i along the viewing ray defined by $\bar{\mathbf{q}}'_i$.

5.4 The Smoothing Term

The point correspondences and the hypothesis of an inextensible surface may not be sufficient constraints. For instance, imagine that there is no point correspondences in a corner of the surface. In this case, nothing says how the surface behaves. The corners of the surface can bend freely as long as they do not extend or shrink (like the corners of a piece of paper). To overcome this difficulty, we add a third term (the smoothing term) in our cost function that favours non-bending surfaces. Note that usually, such terms are used to compensate for the undesirable effects of under-fitting and over-fitting. Doing so is usually a problem because it requires one to determine a correct value for the weight associated to the smoothing term (value β in equation (16)). This balances the effective complexity of the surface against the complexity of the data.

Its value is here not critical. Indeed, the complexity of the surface is limited by the fact that it is inextensible. Any small value (but large enough not to be negligible, for instance $\beta = 10^{-4}$) is thus suitable for the weight of the smoothing term.

5.5 Initialization from a Point-Wise Surface Model

An initial estimate of the control point vector ℓ and of the reconstructed point correspondences μ is required to start minimizing the nonconvex cost. It can be directly provided by one of our convex algorithms.

5.5.1 Initialization from our Point-Wise Surface Model Algorithm

Our convex point-wise surface model based algorithm of §4.2.1 directly provides the depth μ of the reconstructed point correspondences. The control point vector ℓ is then obtained using Linear Least-Squares. The idea is to consider the discrepancy between the estimated 3D points \mathbf{Q}_i , $i \in \{1, \dots, n\}$ and the surface estimate, measured by the following equation:

$$\min_{\ell} \sum_{i=1}^n \|\mathcal{W}_{\ell}(\mathbf{q}_i) - \mathbf{Q}_i\|^2 \Leftrightarrow \min_{\ell} \sum_{i=1}^n \|\mathbf{W}_i \ell - \mathbf{Q}_i\|^2. \quad (18)$$

We also include the Linear Least-Squares smoothing term (5) in the minimization, giving:

$$\min_{\ell} \sum_{i=1}^n \|\mathbf{W}_i \ell - \mathbf{Q}_i\|^2 + \lambda \|\mathbf{B}^{1/2} \ell\|^2. \quad (19)$$

In this equation, $\lambda \in \mathbb{R}_+$ controls the desired amount of regularization, similarly to the smoothing term discussed in §5.4. We fix this parameter to $\lambda = 10^{-4}$ since the FFD function is internally regularized. It thus only needs external regularization in areas with severely sparse data, where external regularization will not compete with any other terms. A short calculation gives the following closed-form solution:

$$\ell = \left(\sum_{i=1}^n \mathbf{W}_i^{\top} \mathbf{W}_i + \lambda \mathbf{B} \right)^{-1} \sum_{i=1}^n \mathbf{W}_i^{\top} \mathbf{Q}_i.$$

5.5.2 Initialization from our Convex Continuous Surface Model Algorithm

Our convex continuous surface model based algorithm of §4.2.2 directly provides the control point vector ℓ . The depth μ of the reconstructed point correspondences is then obtained, for each point independently, by solving the point's error following the data term (17):

$$\min_{\mu_i} \|\mathcal{W}_{\ell}(\mathbf{q}_i) - \mu_i \mathbf{K}^{-1} \bar{\mathbf{q}}'_i\|^2.$$

This is a convex, Linear Least-Squares problem which has the following closed-form solution:

$$\mu_i = (\bar{\mathbf{q}}'_i{}^{\top} \mathbf{K}^{-\top} \mathbf{K}^{-1} \bar{\mathbf{q}}'_i)^{-1} \bar{\mathbf{q}}'_i{}^{\top} \mathbf{K}^{-\top} \mathcal{W}_{\ell}(\mathbf{q}_i).$$

6 Experimental Results

We test and compare several algorithms from state of the art for template-based surface reconstruction. The details of these are given in table 1.

6.1 Experiments on Synthetic Data

In our experiments, we first use synthetic data in order to assess the reconstructed surfaces against ground truth, and thus to measure some 3D surface error.

Shortname	Description	Surface model
Perriolnit	Iterative ad hoc minimization of the maximum depth heuristic [17]	Point-wise
PerrioRef	Nonlinear refinement of the 3D points obtained with Perriolnit [17]	Point-wise
Salz	A convex formulation similar to PointSOCP handling noise with a 3D error [21]	Triangular mesh
PointSOCP	Our convex point-wise method of §4.2.1 handling noise with hard constraints	Point-wise
FFDInit	The surface reconstructed from PointSOCP as described in §5.5.1	Continuous FFD
FFDSOCP	Our direct convex surface reconstruction method of §4.2.2 followed by §5.5.2	Continuous FFD
FFDRefPoints	Our optimal nonconvex surface reconstruction algorithm of §5	Continuous FFD

Table 1: Summary of the algorithms tested and compared experimentally.

6.1.1 Synthetic Surface Generation and Measured Errors

Images. We generated isometric surfaces using the code provided by [16]. The pieces of paper are of size A4 (*i.e.* $297 \times 210\text{mm}^2$) and they are represented by a triangular mesh with 79,158 faces. The input images are simulated by projecting the deformed piece of paper with a virtual perspective camera placed at approximately 1m of the surface and with a focal length of 36mm. As shown in figure 1, different texture-maps were used. A total number of 539 different images were generated using this process. A simple model of illumination (flat shading) was used to create the images with a single light source placed near the optical center simulating a flash light. The template image size ranges from $1,000 \times 707$ pixels to $2,291 \times 1,620$ pixels and always has the aspect ratio of an A4 sheet. The input image size is fixed to $1,024 \times 768$ pixels. In the generation process, special care was taken to get high quality images by using, in particular, an anti-aliasing mechanism (the input images were first generated at a resolution of $4,096 \times 3,072$ pixels and then subsampled to their final resolution of $1,024 \times 768$ pixels).

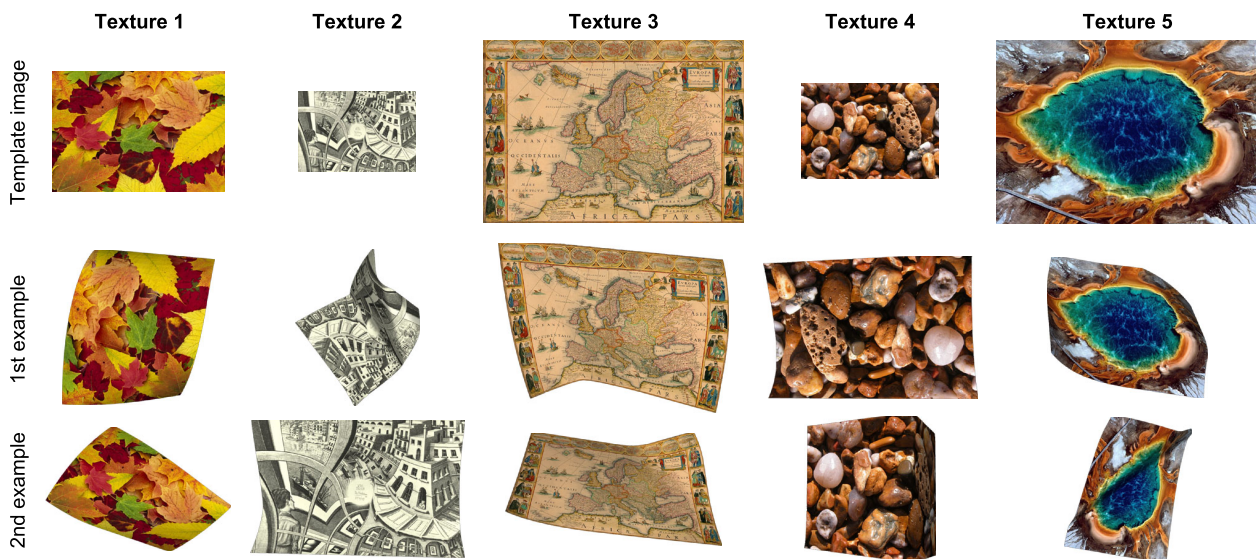


Figure 1: Examples of generated synthetic images. This figure shows the five texture-maps we used. The first row shows the template image. The relative size of these thumbnails corresponds to the true relative size of the template images. The second and third rows show examples of generated input images.

Point correspondences. We used two types of point correspondences. The isometric surface generator [17] provides a surface represented as a triangular mesh with 79,158 faces whose vertices are located on a regular grid of size 238×168 . The first type of point correspondences we used were extracted from these vertices (varying the number of extracted points and, if necessary, adding some noise on their location). The

second type of point correspondences we used were SIFT keypoints automatically extracted from the generated images and matched using the FBDSO algorithm [18]. In this second case, the point correspondences are of course very realistic even though the images were synthetically generated.

Measured reconstruction errors. The discrepancy between the reconstructed and the true 3D surfaces was quantified with two measures, depending on the surface model used by the algorithms. The *point-wise reconstruction error* (PWRE), denoted e_p , can be measured for all the algorithms. It is defined by:

$$e_p = \frac{1}{n} \sum_{i=1}^n \|\mathbf{Q}_i - \hat{\mathbf{Q}}_i\|, \quad (20)$$

where n is the number of point correspondences, \mathbf{Q}_i is the i th reconstructed 3D point and $\hat{\mathbf{Q}}_i$ is the true 3D location of that point. For algorithms that use more complex surface models, such as a triangular mesh or an FFD-based continuous surface model, we measured the *surface reconstruction error* (SRE), denoted e_s . It is defined as the integral difference between the reconstructed surface \mathcal{W}_ℓ and the true surface $\hat{\mathcal{W}}$:

$$e_s = \int_{\Omega} \|\mathcal{W}_\ell - \hat{\mathcal{W}}\| d\mathbf{q}. \quad (21)$$

6.1.2 Reconstruction Errors with Generated Point Correspondences

In this experiment, we used all the 539 randomly generated paper sheets and we varied two parameters, while measuring PWRE and SRE:

- **Point correspondence noise σ .** We added Gaussian noise to the position of the point correspondences with three different standard deviations σ : 0, 1 and 2 pixels, keeping $n = 247$. The results are in figure 2.
- **Number of point correspondences n .** We subsampled the point correspondences with three set sizes: 165, 247 and 368 points, keeping $\sigma = 1$ pixel. The results are in figure 3.

We observe that the iterative algorithms Perriolnit and Perrioref perform well in the absence of noise, but that their performances degrade substantially with noise. In the SOCP-based algorithms (Salz, PointSOCP, FFDSOCP and FFDnit), Salz works best for very small amounts of noise $\sigma \approx 0$ pixels, and closely follows PointSOCP and FFDnit. We can see that FFDSOCP gives poor results in most cases. This comes from the fact that this algorithm has to meet hard reprojection error constraints and the continuous surface model may lack flexibility; this may be due to an overweighted smoothing term or an underestimated number of control points. Finally, we can see that FFDRefPoints obtains the best results, except in the absence of noise, but the error is marginal at this point. This algorithm has the nice property of being rather insensitive to noise and number of point correspondences.

6.1.3 Reconstruction Errors with Matched Keypoint Correspondences

This experiment completes the previous one by drawing error statistics from automatically matched keypoint correspondences. Those were extracted as SIFT points [14] and matched using local consistency with the FBDSO method [18]. Contrarily to generated point correspondences, one cannot vary parameters such as noise. However, the whole setup is more realistic, and closely represents the type of results expected on real data, while providing ground truth for error assessment. The results are in figure 4. Clearly, things are consistent with the previous set of experiments, confirming their validity. We have that FFDRefPoints performs best, followed by Salz with a discrepancy of a little less than an order of magnitude. The other five tested methods follow, all more or less at an order of magnitude of Salz, at the exception of FFDnit which stays a bit closer to Salz than the others.

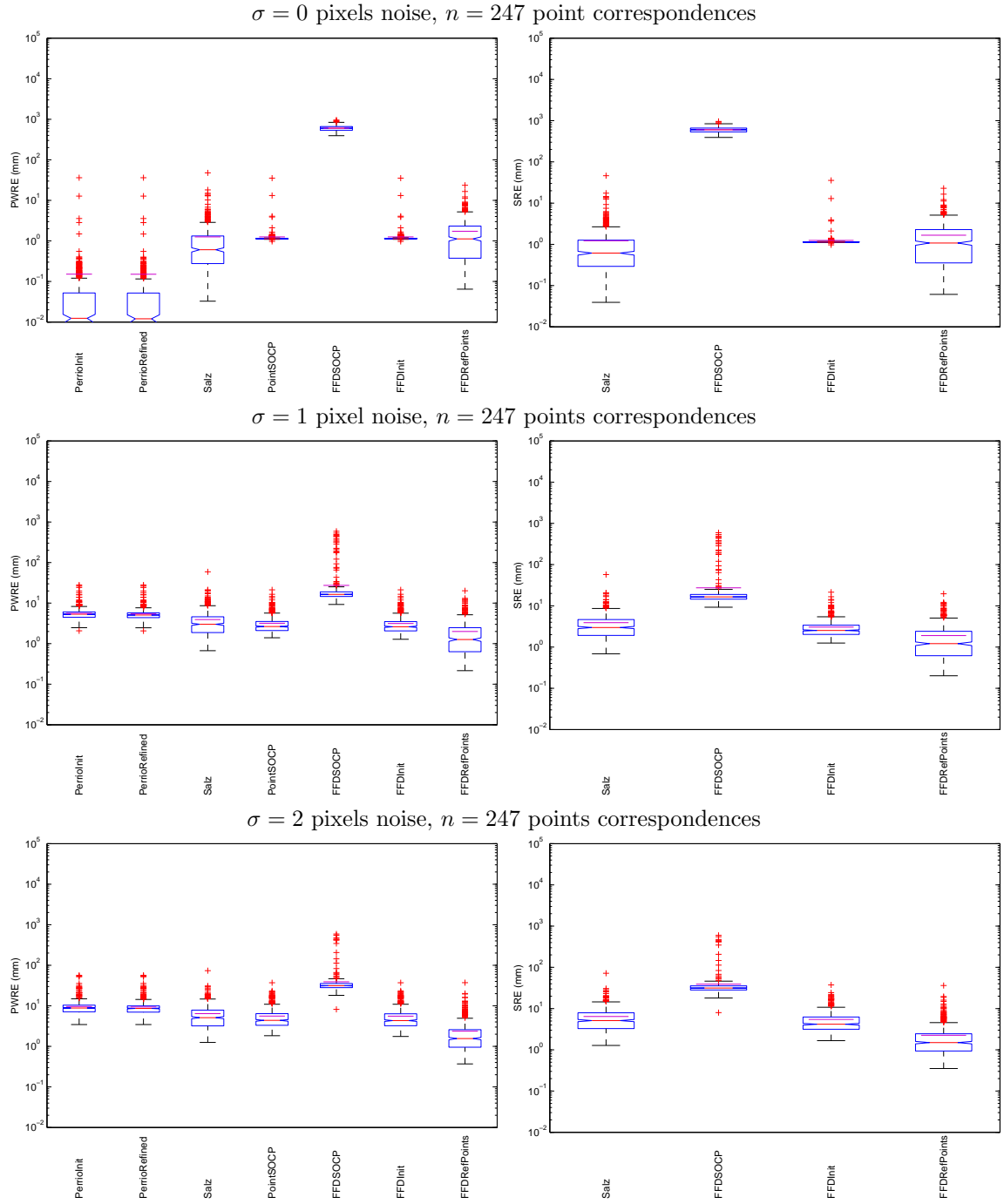
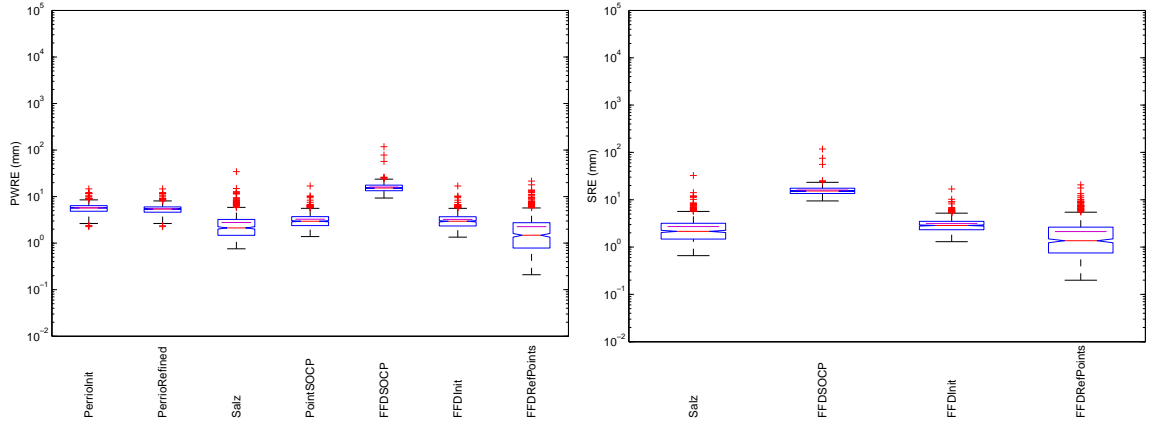


Figure 2: PWRE (left column) and SRE (right column) for $n = 247$ point correspondences for $\sigma = \{1, 2, 3\}$ pixels image noise. We used boxplots with whiskers to show the three quartiles (thus including the median), the min and max errors, and the outliers shown by the red plus signs.

$\sigma = 1$ pixel noise, $n = 165$ point correspondences



$\sigma = 1$ pixel noise, $n = 368$ point correspondences

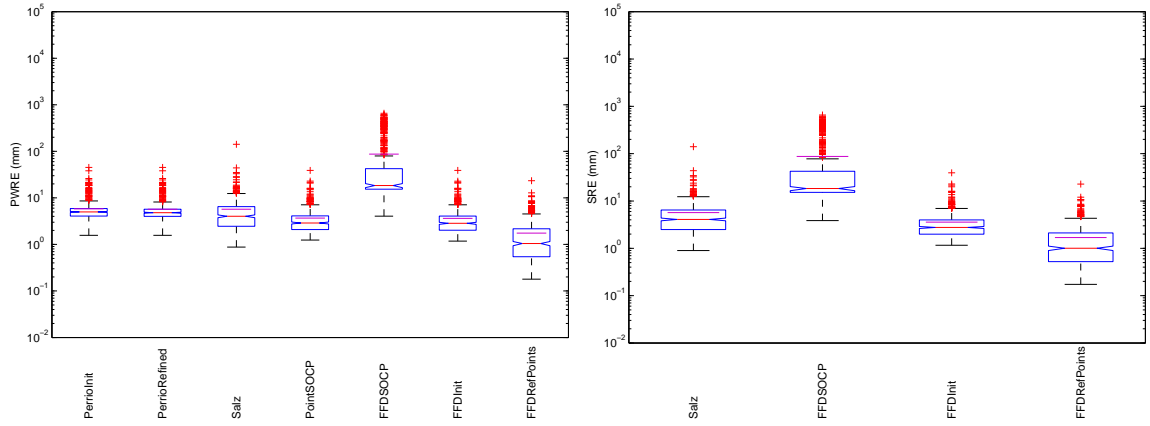


Figure 3: PWRE (left column) and SRE (right column) for $n = \{165, 368\}$ point correspondences for $\sigma = 1$ pixel image noise.

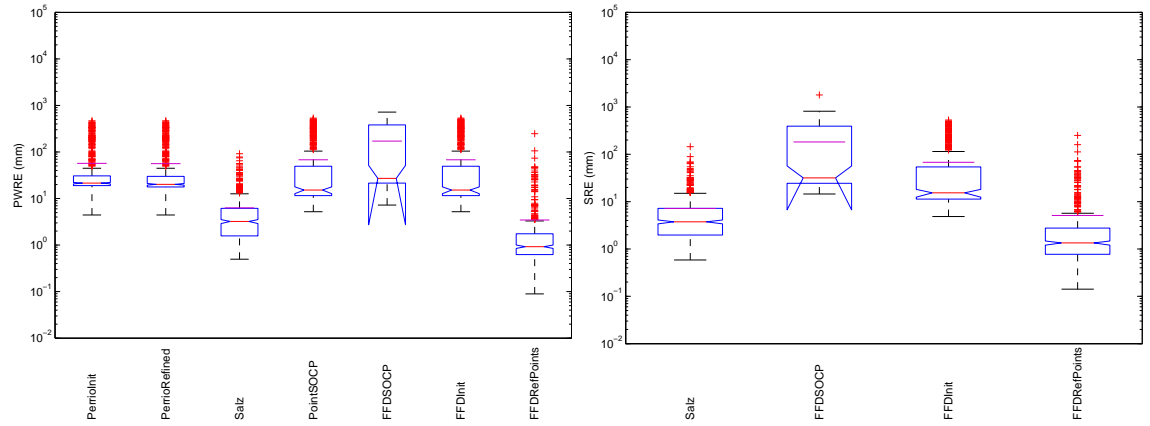


Figure 4: PWRE (left) and SRE (right) using automatically matched SIFT keypoints.

6.1.4 Computation Time

For all experiments, we have monitored the computation time of all algorithms. We observed that, as expected, computation time is almost independent of image noise, but is strongly affected by the number of point correspondences. We therefore show representative results only for this parameter, in figure 5. We observe that SOCP-based methods are quite slow, as well as our nonconvex `FFDRefPoints` method. `Salz` is an order of magnitude faster than the other SOCP methods. This is due to the reduced variable set which it optimizes over, thanks to a learnt surface model. `PerrioInit` and `PerrioRef` are extremely fast, as well as the surface initialization method `FFDInit`.

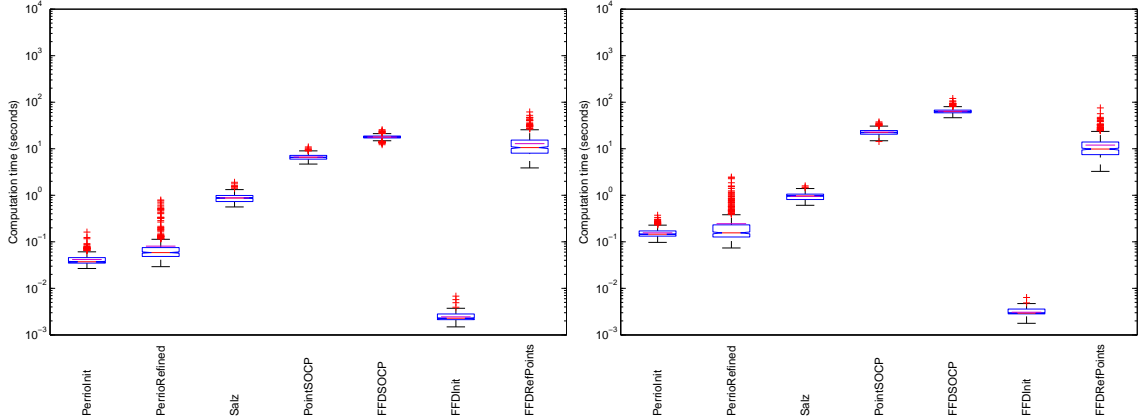


Figure 5: Computation time for $\sigma = 1$ pixel image noise and 165 (left) and 368 (right) point correspondences.

6.1.5 Conservation of Length

In this experiment, we measured to which extent our algorithms reconstruct an isometric surface. We evaluated this quantitatively by measuring the discrepancy between straight lines in the template and the geodesic corresponding to its endpoints in the reconstructed surface. We used the same data as in the previous experiments. For each surface, we randomly chose 10,000 pairs of points in the template image. For each pair of points $(\mathbf{g}_i, \mathbf{g}_j)$, the length l_{ij}^{3D} of the geodesic between the 3D points $\mathcal{W}_\ell(\mathbf{g}_i)$ and $\mathcal{W}_\ell(\mathbf{g}_j)$ was computed as:

$$l_{ij}^{3D} = \sum_{k=1}^{n_g} \left\| \mathcal{W}_\ell\left(\mathbf{g}_i + \frac{k}{n_g} \|\mathbf{g}_j - \mathbf{g}_i\|\right) - \mathcal{W}_\ell\left(\mathbf{g}_i + \frac{k-1}{n_g} \|\mathbf{g}_j - \mathbf{g}_i\|\right) \right\|, \quad (22)$$

where n_g is the number of intermediate points used for the approximation (we use $n_g = 200$ since we experimentally observed that the approximation stabilizes for values of n_g greater than 180). The length l_{ij}^{3D} of the geodesic is plotted against the reference length l_{ij}^{2D} in figures 6 and 7 against noise and number of points respectively.

We make the same observations on both sets of graphs. We observe that `FFDInit` violates isometry to a small extent. `FFDSOCP` however, despite using the same inextensibility relaxation of isometry as `FFDInit`, violates isometry to a large extent. This is explained by the fact that the continuous surface model, incorporated directly in `FFDSOCP`, may not be flexible enough to be able to force isometry. Finally, our nonconvex method `FFDRefPoints` respects isometry to a very good extent, and thus can be said as reconstructing truly isometric surfaces, as required.

6.1.6 Gaussian Curvature

The Gaussian curvature is the product of the two principal curvatures (which are the reciprocal of the radius of the osculating circle). For an isometric and developable surface, the Gaussian curvature vanishes. However,

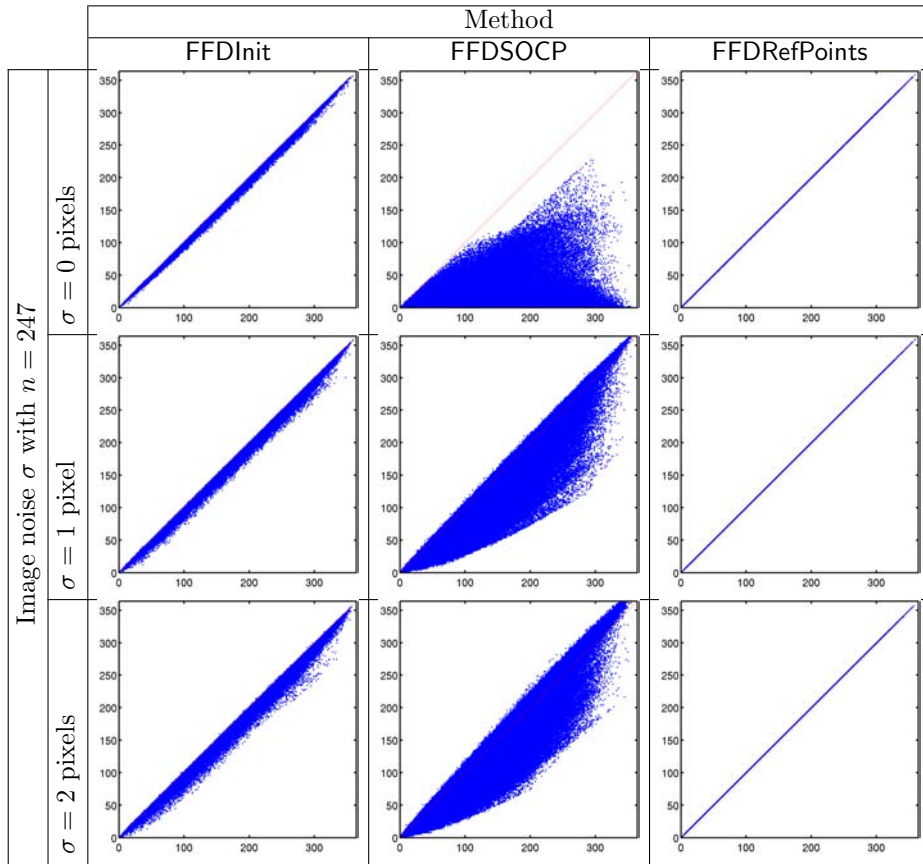


Figure 6: Plots of the length of geodesics measured on the reconstructed surfaces against the true length for $\sigma \in \{0, 1, 2\}$ pixels image noise and $n = 247$ point correspondences. The red diagonal line is the place where all the blue points should lie for an exactly isometric surface.

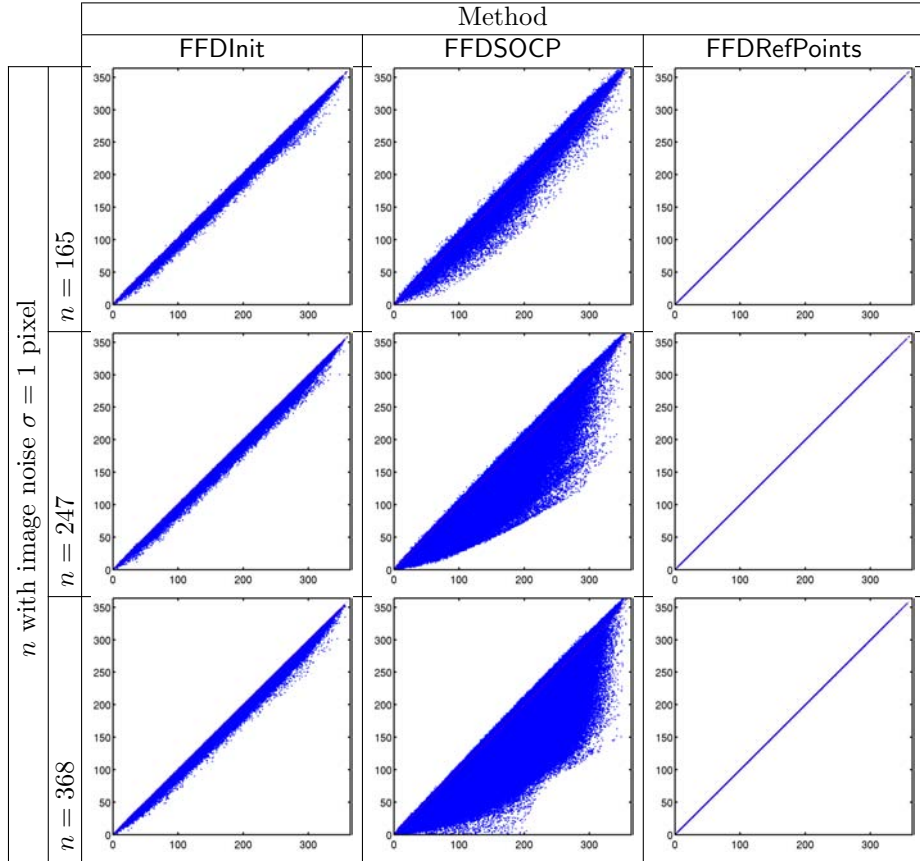


Figure 7: Plots of the length of geodesics measured on the reconstructed surfaces against the true length for and $\sigma = 1$ pixel image noise and $n \in \{165, 247, 368\}$ point correspondences. The red diagonal line is the place where all the blue points should lie for an exactly isometric surface.

the Gaussian curvature is a highly nonlinear function of the surface parameters, and it would be difficult to minimize efficiently compared to the isometry constraint. In this experiment, we checked if this property was satisfied by the continuous surfaces reconstructed with FFDSOCP, FFDInit, and FFDRefPoints. We used the same 539 reconstructed surfaces as in the previous experiments. The Gaussian curvature, denoted κ , was computed for 10,000 randomly chosen points on the surface with the formula $\kappa = \frac{\det(\mathbf{II})}{\det(\mathbf{I})}$, where \mathbf{I} and \mathbf{II} are the first and the second fundamental forms of the parametric surface [11]. The results of this experiment are reported in table 2. It shows that, on average, the Gaussian curvature of the surfaces reconstructed using FFDRefPoints are consistently close to 0. It also shows that FFDRefPoints gives Gaussian curvatures which are two orders of magnitude lower than those obtained with FFDInit. These results demonstrate that the surfaces reconstructed with our approach FFDRefPoints are truly isometric, but also preserves the local surface structure of the template. On the other hand, FFDSOCP does not preserve well this local surface structure.

	Mean	Std deviation	Median	Minimum	Maximum
FFDSOCP	2.82×10^{-4}	6.16×10^{-2}	4.03×10^{-5}	1.31×10^{-11}	8.03×10^1
FFDInit	4.81×10^{-5}	3.30×10^{-3}	1.15×10^{-5}	3.14×10^{-12}	6.91×10^0
FFDRefPoints	4.94×10^{-7}	1.42×10^{-6}	1.51×10^{-7}	3.24×10^{-14}	2.35×10^{-4}

Table 2: Statistics on the absolute value of the Gaussian curvature for 539 reconstructed surfaces and 10,000 points per surface. The number of point correspondences and the noise intensity are fixed to intermediate values ($n = 247$ and $\sigma = 1$ pixel).

6.1.7 Hyperparameter Tuning and Sensitivity Analysis

This set of experiments on synthetic data is intended to test the sensitivity of the algorithms on the various hyperparameters they involve (such as the noise level). All the experiments were conducted on the same synthetic data as previously with $n = 247$ point correspondences.

Method PointSOCP. In the PointSOCP method, we study the influence of the parameters $\varepsilon_{\mathcal{T}}$ and $\varepsilon_{\mathcal{I}}$ of equation (12). These parameters account for the noise in the template and the input images respectively. For each level of noise σ added to the data point, we search the best values of $\varepsilon_{\mathcal{T}}$ and $\varepsilon_{\mathcal{I}}$, *i.e.* the values that minimize PWRE. The results are showed in figures 8 and 9. These figures show that setting correctly $\varepsilon_{\mathcal{I}}$ and $\varepsilon_{\mathcal{T}}$ may be critical. Figure 9 shows that there is an almost linear relationship between the standard deviation of the noise added to the data point and the supposed noise of the algorithm. Both figures show that, as a rule of thumb, $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}} = 2\sigma$ is an optimal choice, and that in practice, σ being generally unknown, overestimating is better than underestimating $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}}$.

Methods Perriolnit, PerrioRef. These two methods have a single parameter to be set: $\varepsilon_{\mathcal{T}}$, the supposed noise in the template image. The influence of this parameter is tested using exactly the same principle as for the PointSOCP algorithm. The results are displayed in figures 10 and 11. We draw the same conclusions as for PointSOCP: the results do depend on the hyperparameter. As a rule of thumb, for both Perriolnit and PerrioRef, $\varepsilon_{\mathcal{T}} = 4\sigma$ seems to be a quasi optimal choice, and $\varepsilon_{\mathcal{T}}$ is better overestimated.

Method Salz. As described in [21], the method Salz has a weight w_d that controls the relative influence of two terms in a cost function to minimize (one of these terms accounts for the consistency between the reprojection of the 3D points and their actual location in the input image, the second one quantifies the distance between the 3D points and their corresponding line of sight). In [21], the authors recommend to set w_d to $\frac{2}{3}$. We studied the influence of this parameter using a principle similar to the one used previously with two noticeable differences. First, for computational complexity purposes, we added a Gaussian noise to the data points only in the input image (which explains why we used standard deviations twice as large as in the previous cases). Second, the quantification of the quality of a reconstructed surface is evaluated with SRE instead of PWRE. The results are shown in figures 12 and 13. These figures show that a value of 2 for

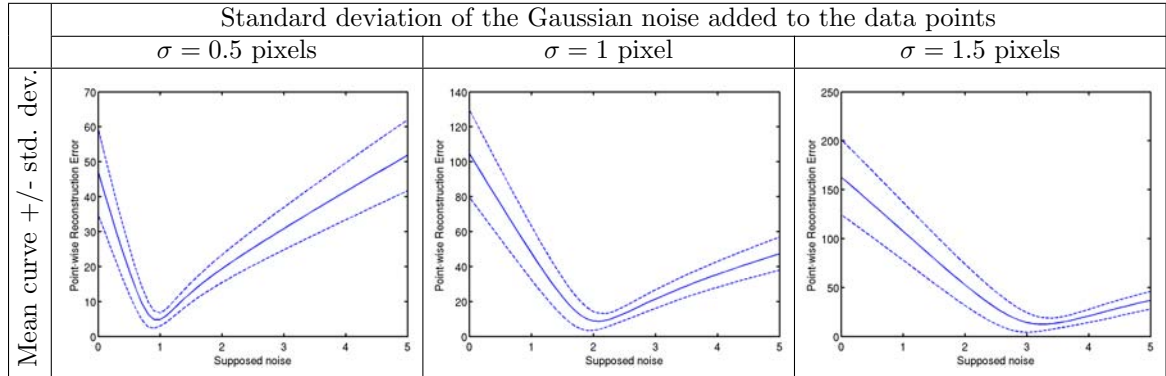


Figure 8: Influence of the $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}}$ hyperparameter in PointSOCP. Evolution of PWRE in function of the noise added to the data points (columns of the table) and of the values of $\varepsilon_{\mathcal{T}}$ and $\varepsilon_{\mathcal{I}}$ (abscissa of the plots). We used the same noise intensity in the template and the input images to alterate the data points. We further considered that $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}}$. The first row shows the evolution of PWRE for all the 539 test cases. The second row shows the average curve (solid line) and the standard deviation (dashed lines).

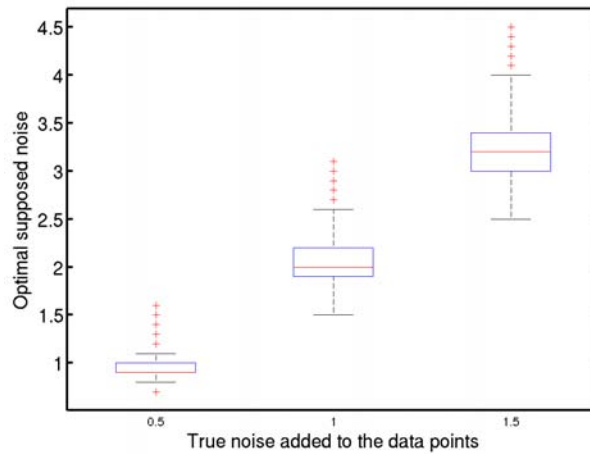


Figure 9: Influence of the $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}}$ hyperparameter in PointSOCP. Optimal supposed noise $\varepsilon_{\mathcal{T}} = \varepsilon_{\mathcal{I}}$ in function of the noise added to the data points.

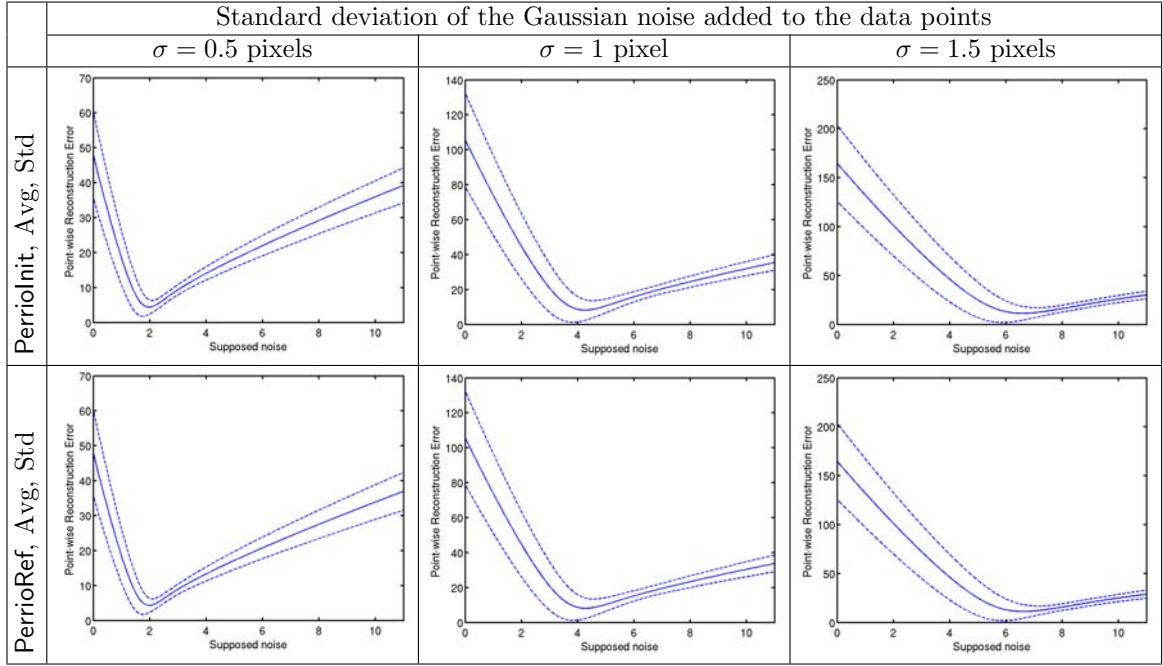


Figure 10: Influence of the $\varepsilon_{\mathcal{T}}$ hyperparameter in Perriolnit and Perrioref. Evolution of PWRE in function of the noise added to the data points (columns of the table) and of the values for $\varepsilon_{\mathcal{T}}$ (abscissa of the plots). We used the same noise intensity in the template and the input images to alterate the data points. The first and third rows show the evolution of PWRE for all the 539 test cases for the Perriolnit and Perrioref algorithms respectively. The second and fourth rows show the average curve (solid line) and the standard deviation (dashed lines) for the Perriolnit and Perrioref algorithms respectively.

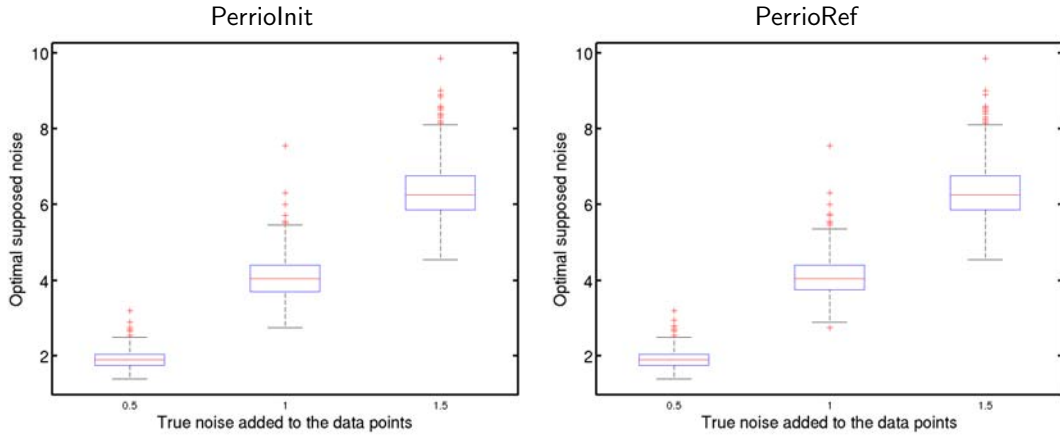


Figure 11: Influence of the $\varepsilon_{\mathcal{T}}$ hyperparameter in Perriolnit and Perrioref. Optimal supposed noise $\varepsilon_{\mathcal{T}}$ in function of the noise added to the data points.

w_d is preferable to $\frac{2}{3}$. Besides, when overestimated, w_d does not degrade the results much compared to its optimal value.

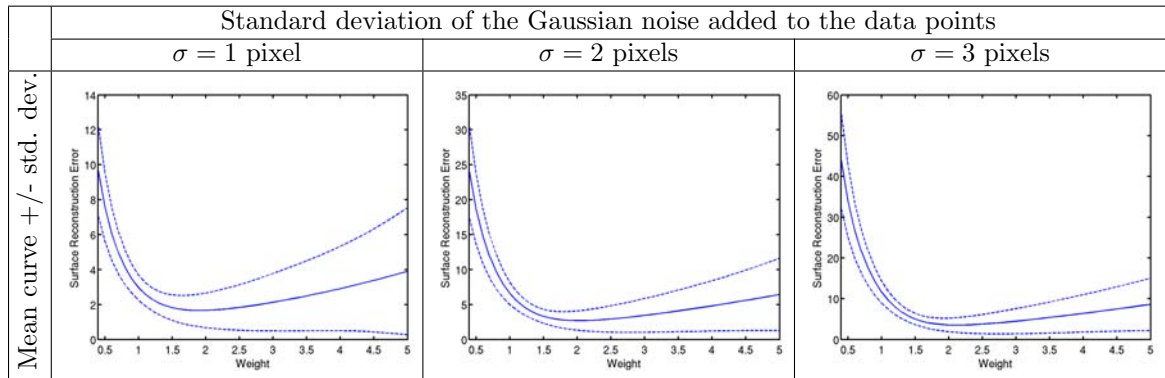


Figure 12: Influence of the w_d hyperparameter in Salz. Evolution of SRE in function of the noise added to the data points (columns of the table) and of the values for the weight w_d (abscissa of the plots) of the Salz algorithm. Here, only the points in the input image are distorted. The first row shows the evolution of SRE for all the 539 test cases. The second row shows the average curve (solid line) and the standard deviation (dashed lines).

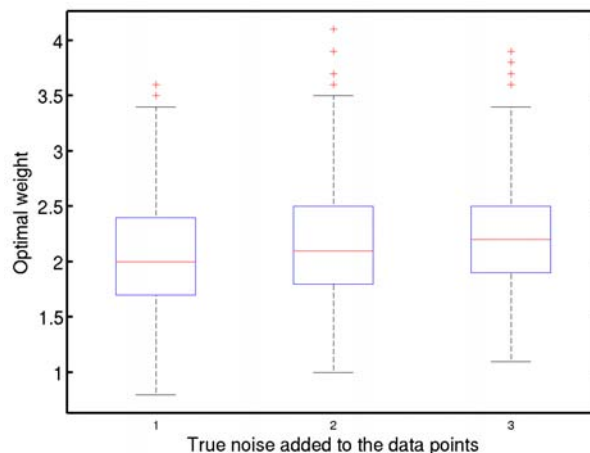


Figure 13: Influence of the w_d hyperparameter in Salz. Optimal weight w_d in function of the noise added to the data points.

6.2 Experiments on Real Data

We compared our algorithms to existing ones on several real datasets. The three first were provided by other authors and we created the last one. Our nonconvex method FFDDRefPoints gives the best results in all cases. We deal with the problem of establishing point correspondences with different state-of-the-art methods, from direct video stream tracking [9], robust fitting of a warp function and local consistency based filtering [18]. The first method is only applicable for videos, but is probably the best choice in this case. The second and third methods are applicable to two images. The second method works very well if the images are similar to each other, while the third one is specifically designed to the wide-baseline case, where the input image may be very different from the template.

The checkerboard dataset [17]. This first dataset, provided by [17], shows a bent piece of paper printed with a checkerboard pattern. For this dataset, the point correspondences were established manually by clicking. Ground truth was available from stereo reconstruction, and allowed use to compute PWRE for all algorithms:

PerrioRef	PointSOCP	Salz	FFDInit	FFDRefPoints
2.39mm	2.26mm	4.74mm	2.26mm	1.99mm

A visualization of the reconstruction surfaces is shown in figure 14. All methods give decent qualitative visual results. However, in terms of 3D accuracy, Salz gave the worst results, with a 3D error about twice as large as the other methods. PerrioRef, PointSOCP and FFDInit gave similar 3D errors, while FFDRefPoints gave the lowest average 3D error, below 2mm.

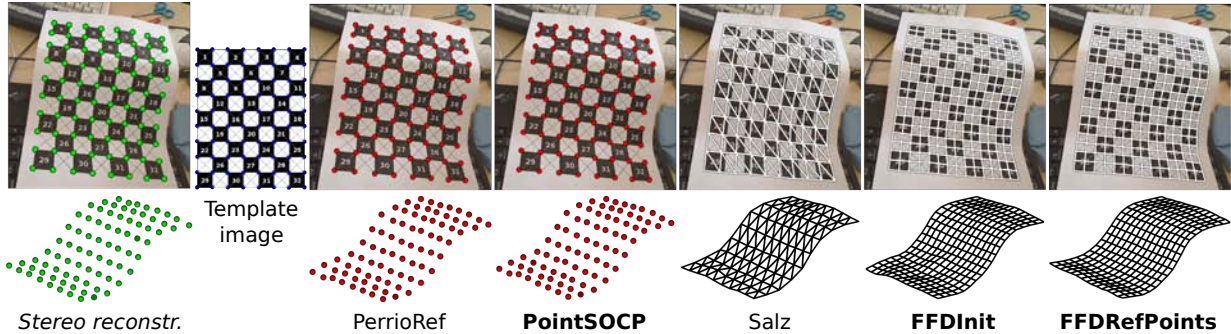


Figure 14: **The checkerboard dataset.** Reconstruction results obtained with several. First row: input image along with a reprojection of the reconstructed 3D surface. Second row: reconstructed surface from a different point of view. Note that the stereo reconstruction (first column) is not a monocular algorithm: it is used as ground truth.

The self-occluded paper dataset [9]. This second dataset, provided by [9], shows a piece of paper with a printed natural texture and bent in a way that creates a strong self-occlusion (about half of the surface is hidden in the input image). This makes this dataset difficult for the algorithms since the hidden part of the surface has to be inferred from the visible parts only from isometry. The point correspondences were obtained by sampling an image warp function which was fitted using a direct method to a video stream. The reconstruction results are shown in figure 15. For this dataset, we had no ground truth, and could not report quantitative results. However, we can observe that all methods recover the paper’s global shape to some extent. However, only Salz, FFDInit and FFDRefPoints were able to reconstruct the self-occluded part. Our nonconvex method, FFDRefPoints, clearly does better in recovering the paper’s shape than the others, even in the self-occluded part, which we consider as a very successful result.

The deforming paper dataset [22]. This third dataset, provided by [22], shows a piece of paper being smoothly bent in front of the camera. The deformation does not create self-occlusions or external occlusions. However, a difficulty is that the point correspondences provided in this datasets contain noise and a few mismatches. Another difficulty is that the imaging conditions are close to affine, while all tested algorithms explicitly use a perspective camera model. We discard them prior to applying the reconstruction methods, by robustly fitting a 2D FFD warp [20, 18] using an M-estimator, and sampling it over a (25×25) regular grid. This process reduces the amount of noise in the point positions. The initial correspondences and the estimated warp can be seen in figure 16. The reconstruction results are showed in figure 17. We do not have groundtruth for this dataset. However, the surface deformation is so simple that the reconstruction quality can be easily assessed visually, at least qualitatively. We observe that the reconstruction of all methods captures the surface’s bulk shape for all frames. All methods then generate artefacts on at least some of the frames. These happen at different places of the surface. The only exception is FFDRefPoints, which visually produces the best surface reconstruction for all frames, closely followed by PointSOCP. The largest artefacts seems to be provided by PerrioRef, FFDInit and FFDSOCP, while PerrioRef and Salz do slightly better.

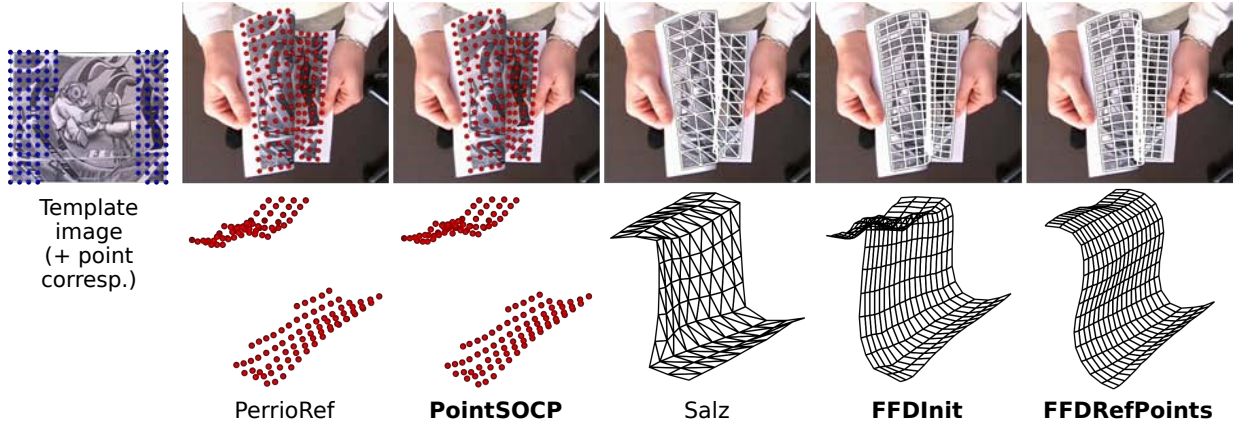


Figure 15: **The self-occluded paper dataset.** Reconstruction results for several algorithms in the presence of a self-occlusion (the point correspondences were automatically extracted using [9]). Note how our algorithm FFDRefPoints recovers a reasonable shape for the occluded part.

The zooming paper dataset. We propose this new dataset, which is original in that it provides several calibrated levels of zoom for the same surface and the same camera (we here show four levels of zoom in increasing order). The stronger the zoom, the more affine the imaging conditions become, and thus the more challenging the data become for the tested perspective algorithms. The template was matched to the input image entirely automatically using FBDS [18]. The intuition in FBDS is that, for a surface, points close in space will tend to move ‘together’; mismatches thus form locally inconsistent motion fields. The correspondence results, as well as the estimated image warp, are shown in figure 18. The 3D reconstruction results are shown in figure 19. All methods but FFDRefPoints find the surface’s bulk shape. Because of the high density of point correspondences, PerrioRef, Salz, PointSOCP and FFDInit obtain very neat and smooth results, up to some small artefacts. Our refinement method FFDRefPoints clearly gives the best results in terms of visual impression.

7 Conclusion

We have presented new approaches for monocular 3D reconstruction of inextensible surfaces imaged by a perspective camera. First, we have proposed an SOCP formulation of the problem that accounts for noise in both the template and the input images. Second, we have designed an algorithm that directly integrates a continuous surface model based on Free-Form Deformations within the SOCP formulation. Third, we have given a nonconvex algorithm which, using Nonlinear Least-Square refinement, obtains the continuous surface model parameters by refining an initial guess. We have given a thorough experimental evaluation of the sensitivity of our methods and others from the literature to their hyperparameters, and have compared all methods in terms of reconstruction accuracy on simulated and real datasets. This has allowed us to draw the following important conclusions.

(i) SOCP methods are generally working reasonably in terms of accuracy, though they are slow, and do not scale well with the number of point correspondences or surface model complexity. They are sensitive to the selected noise level, and a bit less to the smoothing parameter. (ii) Incorporating a surface model in the SOCP framework has not worked well in most of our experiments, and we do not recommend to do so. (iii) Finally, nonlinear refinement is important, and improves the result of the tested methods by a substantial margin. We have showed that our formulation of nonlinear refinement produces truly isometric deformations, as opposed to initialization methods. Being able to enforce this type of physical constraints is one of the keys to increase 3D accuracy.

In our future work, we are planning to use the image color information directly to further improve 3D reconstruction. We would want to initialize a so-called direct approach from an existing feature-based method, so as to be able to use the whole image content, which we believe is particularly important for tasks

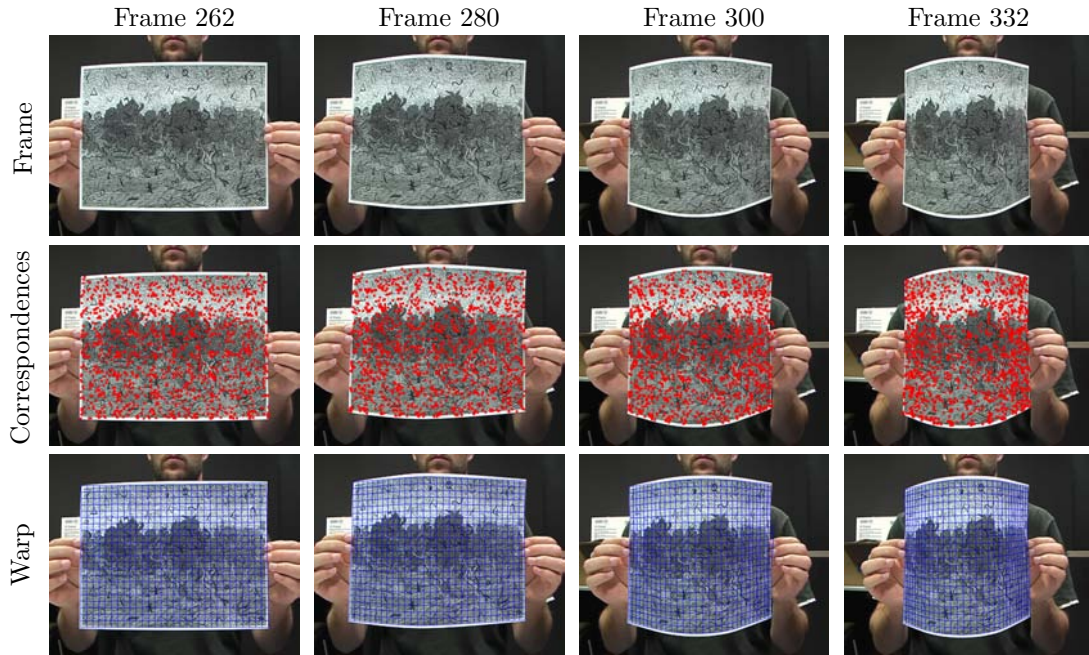


Figure 16: **The deforming paper dataset.** Sample frames, point correspondences and estimated image warps.

such as reconstructing a deformable surface, which by definition requires localized information.

Acknowledgement. Florent Brunet has been partly funded by Conseil Régional d’Auvergne and Richard Hartley by NICTA and the Australian Government through the Australian Research Council. This research has received funding from the EU’s FP7 through the ERC research grant 307483 FLEXABLE.

References

- [1] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S.I. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [3] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2696, 2000.
- [5] F. Brunet, V. Gay-Bellile, A. Bartoli, N. Navab, and R. Malgouyres. Feature-driven direct non-rigid image registration. *International Journal of Computer Vision*, 93(1):33–52, May 2011.
- [6] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *Asian Conference on Computer Vision*, 2010.
- [7] A. Del Bue. A factorization approach to structure from motion with shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [8] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, 1993.

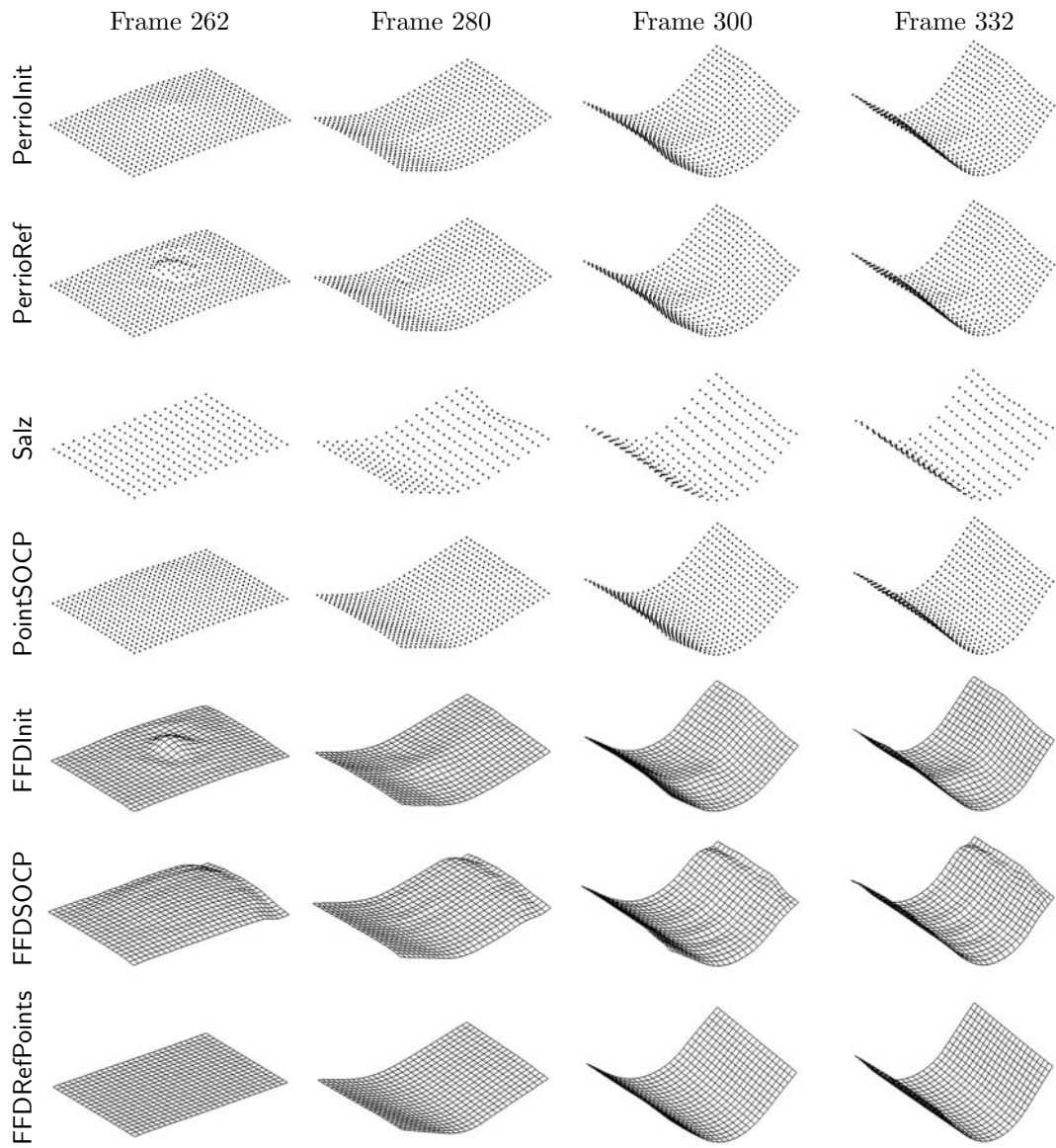


Figure 17: **The deforming paper dataset.** 3D reconstruction results for all compared methods.



Figure 18: **The zooming paper dataset.** Sample frames, point correspondences and estimated image warps.

- [9] V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:87–104, 2008.
- [10] V. Gay-Bellile, M. Perriollat, A. Bartoli, and P. Sayd. Image registration by combining thin-plate splines with a 3D morphable model. In *International Conference on Image Processing*, 2006.
- [11] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, chapter The Gaussian and Mean Curvatures, pages 373–380. CRC Press, 1997.
- [12] N. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. 3D structure recovery and unwarping of surfaces applicable to planes. *International Journal of Computer Vision*, 66(3):261–281, March 2006.
- [13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.
- [14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [15] S.I. Olsen and A. Bartoli. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision*, 31(2-3):233–244, 2008.
- [16] M. Perriollat and A. Bartoli. A computational model of bounded developable surfaces with application to image-based 3D reconstruction. *Computer Animation and Virtual Worlds*, 24(5):459–476, September 2013.
- [17] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*, 95(2):124–137, 2011.
- [18] D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

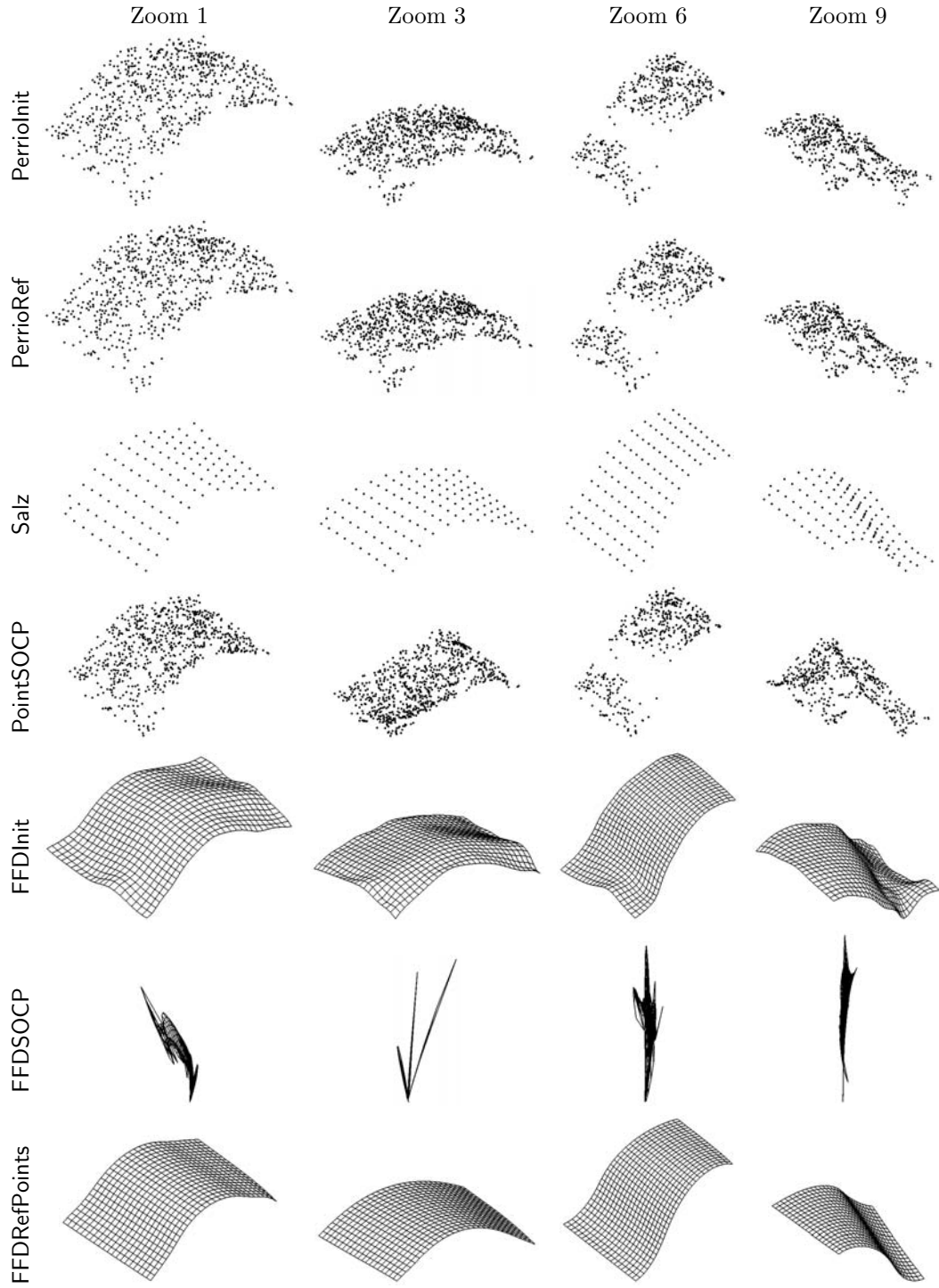


Figure 19: **The zooming paper dataset.** 3D reconstruction results for all compared methods.

- [19] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Single view reconstruction of curved surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
- [20] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.
- [21] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [22] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-D tracking. In *IEEE International Conference on Computer Vision*, 2007.
- [23] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. In *European Conference on Computer Vision*, 2008.
- [24] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [25] S. Shen, W. Shi, and Y. Liu. Monocular template-based tracking of inextensible deformable surfaces under L_2 -norm. In *Asian Conference on Computer Vision*, 2009.
- [26] S. Shen, W. Shi, and Y. Liu. Monocular 3-D tracking of inextensible deformable surfaces under L_2 -norm. *IEEE Transactions on Image Processing*, 19:512–521, 2010.
- [27] 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.
- [28] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, 2006.
- [29] J. Zhu, S.C.H. Hoi, and M.R. Lyu. Nonrigid shape recovery by gaussian process regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.