Shape-from-Template with Curves

Mathias Gallardo[1] **Daniel Pizarro**[2,1] **Toby Collins**[3,1] **Adrien Bartoli**[1]

[1]EnCoV, IP, UMR 6602 CNRS, Université Clermont Auvergne, SIGMA, France

[2]Geintra Research Group, Universidad de Alcalá, Alcalá de Henares, Spain

[3]IRCAD, Strasbourg, France

Corresponding author: Mathias Gallardo

Mathias.Gallardo@gmail.com

August 14, 2019

Accepted by the *International Journal of Computer Vision*

Abstract

Shape-from-Template (SfT) is the problem of using a shape template to infer the shape of a deformable object observed in an image. The usual case of SfT is 'Surface' SfT, where the shape is a 2D surface embedded in 3D, and the image is a 2D perspective projection. We introduce 'Curve' SfT, comprising two new cases of SfT where the shape is a 1D curve. The first new case is when the curve is embedded in 2D and the image a 1D perspective projection. The second new case is when the curve is embedded in 3D and the image a 2D perspective projection. We present a thorough theoretical study of these new cases for isometric deformations, which are a good approximation of ropes, cables and wires. Unlike Surface SfT, we show that Curve SfT is only ever solvable up to discrete ambiguities. We present the necessary and sufficient conditions for solvability with critical point analysis. We further show that unlike Surface SfT, Curve SfT cannot be solved locally using exact non-holonomic Partial Differential Equations (PDE). Our main technical contributions are two-fold. First, we give a stable, global reconstruction method that models the problem as a discrete Hidden Markov Model (HMM). This can generate all candidate solutions. Second, we give a non-convex refinement method using a novel angle-based deformation parameterization. We present quantitative and qualitative results showing that real curve shaped objects such as a necklace can be successfully reconstructed with Curve SfT.

CONTENTS

I. Introduction

A. Background

SfT aims to register and reconstruct the 3D shape of a deforming object from a single input image and a template of the object. Registration and reconstruction are solved by finding the deformation transformation that embeds the template in 3D camera coordinates. A template stores, at the very least, a manifold representing the shape of an object with its texture in a known reference state. For example, the template can be generated from a Computer Aided Design (CAD) model of the object, or reconstructed from data such as Structure-from-Motion (SfM), with a set of rigid views. Various different cases of SfT exist and they can be distinguished by three main properties. These are as follows: *(a)* the dimension x of the template manifold, *(b)* the dimension y of the Euclidean space $\mathbb{R}^y$ that embeds the template manifold in camera coordinates with $y \geq x$, and *(c)* the dimension z of the Euclidean camera image with $z \leq y$. The primary goal of SfT is then to determine the embedding function $\varphi : \mathbb{R}^x \to \mathbb{R}^y$ that embeds the shape template in $\mathbb{R}^y$, from an image of dimension z. We denote a particular SfT case with $\text{SfT}^{x \to y \to z}$. Our notation follows the image formation procedure and first visualizes the template since it is the most discriminative characteristic of the SfT problem. We then obtain that in $\text{SfT}^{x \to y \to z}$ the image where we see the deformed curve is created from an x-D template which is deformed in y-D and then projected in a z-D image. Table I summarizes the important cases. Nearly all SfT methods are named Surface SfT methods because they require a surface template, *i.e.* a thin-shell model without volume. There also exist some Volume SfT methods [Parashar et al., 2015; Collins and Bartoli, 2015]. They require to model the object with a volumetric deformable model, using either continuous models, such as 3D splines [Parashar et al., 2015] or discrete models, such as tetrahedral meshes [Collins and Bartoli, 2015]. We introduce the special case of Curve SfT and reveal its practical uses and hidden complexity.

a) Motivations: The development of SfT with curvilinear templates has not been reported in the literature. In other words, there exist no theory and no reconstruction algorithms for such templates. Our goal is to understand which reconstruction problems can be solved for curvilinear objects and which are the conditions for solvability. Importantly, such templates exist for real physical objects: wires, cables, necklaces and roadlines to name but a few all fall in this case and cannot be reconstructed by any existing method which assumes the object has a surface extent.

	SfT case	Template dimension	Unknown embedding	Observational data	Key references
Volume SfT	$\text{SfT}^{3\to3\to2}$	3D	3D → 3D	2D region in 2D image	[Parashar et al., 2015; Collins et al., 2016]
Surface SfT	$\text{SfT}^{2\to3\to2}$	2D	2D → 3D	2D region in 2D image	[Salzmann and Fua, 2009; Bartoli et al., 2015]
Curve SfT	$\text{SfT}^{1\to3\to2}$	1D	1D → 3D	2D curve in 2D image	Proposed
	$\text{SfT}^{1\to2\to2}$	1D	1D → 2D	2D curve in 2D image	×
	$\text{SfT}^{1\to2\to1}$	1D	1D → 2D	1D straight line in 1D image	Proposed

TABLE I
Taxonomy of SfT cases. $\text{SfT}^{x\to y\to z}$ denotes a case with a template manifold of dimension x, a Euclidean embedding space with dimension y and a perspective camera with image dimension z.

We propose a thorough theoretical study and practical solutions of Curve SfT. As with all SfT cases, Curve SfT requires the use of deformation priors because of the loss of shape information from camera projection. For this, we use the isometry prior. We consider here two main cases of Curve SfT. The first case is when the template is a curve embedded in the 3D space and observed by a regular 2D camera, referred as SfT$^{1\to3\to2}$. A practical example is to reconstruct a thin necklace around a person's neck, given a template of the necklace, as shown at the bottom of figure 1. The second case is similar to the first instance, but the camera is 1D, referred as SfT$^{1\to2\to1}$. It may be created from an orthogonal view of the ground plane, for instance, , as shown at the top of figure 1. At first glance, the use of 1D templates in Curve SfT may seem to make SfT simpler compared with a 2D template in Surface SfT. However, we found that *Curve SfT has fundamental theoretical differences concerning degeneracies, well-posedness and solution uniqueness*. These differences motivate us to propose new theoretic and algorithmic solutions.

b) Theoretical contributions to Curve SfT: We use continuous differential geometry to analyze and derive local solutions, problem well-posedness and ambiguities. We show that the two sub-cases of Curve SfT, which are two problems with different dimensions, can both be written as the same first-order Ordinary Differential Equation (ODE) and solved through an Initial Value Problem (IVP). However, the initial condition required to solve the IVP is a known depth at one point. At first glance, this additional information is generally unavailable. We propose a strategy to solve the IVP by giving an initial condition which is directly obtained from the ODE. This initial condition uses special points of the curve, called the *super critical points*. Through the IVP with the super critical points, the mathematical formulation of Curve SfT gives several solutions which we call *candidate solutions*. We prove the following results:

1) In Curve SfT, the depth of a point is uniquely recoverable if and only if it is a super critical point.
2) Curve SfT is solvable up to a finite number of solutions if and only if there exists at least one super critical point.
3) A section of template falling between two super critical points is recoverable up to a two-fold ambiguity.
4) A Curve SfT problem with N_s super critical points has 2^{N_s+1} discrete candidate solutions.

We also study the solvability of Curve SfT with a method using local non-holonomic solution to our PDE. We prove the following results:

1) Unlike Surface SfT, Curve SfT cannot be solved exactly using local non-holonomic solution to our PDE.
2) By neglecting curvature, it is possible to solve Curve SfT using local non-holonomic solution to our PDE.

c) Technical contributions to Curve SfT: In the literature, there exist three categories of methods to solve Surface and Volume SfT: *(i)* local analytical solutions, *(ii)* convex optimization and *(iii)* non-convex iterative optimization. We give a computational solution for each category for solving Curve SfT. For the category *(i)* method, we first proceed to a similar differential analysis as [Bartoli et al., 2015] and then consider non-holonomic solutions under the assumption of infinitesimal linearity. For the category *(ii)* method, we adapt a convex formulation of Surface SfT, called Maximum Depth Heuristic (MDH) [Perriollat et al., 2011; Salzmann and Fua, 2011]. It uses the inextensibility constraint, a relaxation of the isometry constraint. For the category *(iii)* method, we propose a non-convex continuous formulation that can be optimized efficiently using gradient-based minimization. We achieve this with a novel angle-based parameterization which implicitly models isometric deformations. However, the categories of method *(i)* and *(ii)* only provide a single solution and the category *(iii)* method works from an initial solution. As the theory shows that the Curve SfT problem has several solutions, we introduce a new category *(iv)* of SfT method, which gives all candidate solutions and uses a discrete graphical model. Our method models SfT with a discrete graphical model *without any assumption on the geometry*, *without relaxing isometry* and *without requiring an initial estimate*. This makes it very different to the three categories of method used previously for solving Surface and Volume SfT. Importantly, our category *(iv)* method generates all candidate solutions and thus the true curve, as figure 1 illustrates. We emphasize that all existing categories of method for solving Surface and Volume SfT with isometric deformations and perspective cameras are only able to generate one solution, which is insufficient to solve Curve SfT because of its inherent ambiguity.

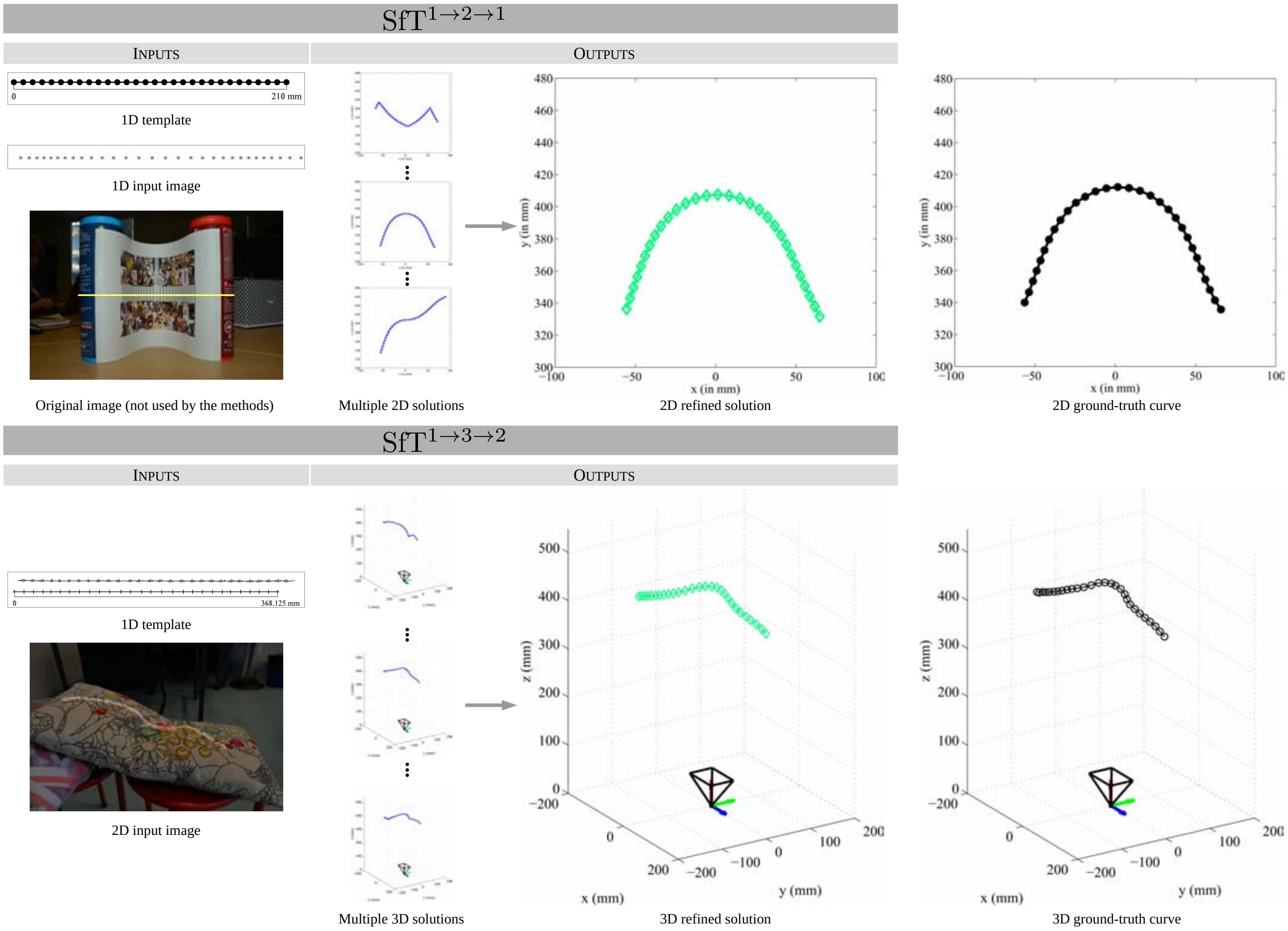


Fig. 1. **Top**: an example of **2D** curve reconstruction from a **1D** input image and a **1D** template using our refined solution of $\mathrm{SfT}^{1\to2\to1}$. It uses the *paper* dataset. The yellow line in the original image corresponds to the 1D input image. **Bottom**: an example of **3D** curve reconstruction from a **2D** input image and a **1D** template using our refined solution of $\mathrm{SfT}^{1\to3\to2}$. It uses the *necklace* dataset. In order to give a better visualization of the necklace on the pillow, we have brightened the region near the necklace by dimming the rest of the image. For our method, inputs are correspondences between the 1D template and the input image: respectively the midpoint of the bars for the *paper* dataset and the center of gravity of the pearls for the *necklace* dataset. For both datasets, we show several candidate solutions obtained by our category *(iv)* method and give the refined version of the best solution.

B. Context

Extension of [Gallardo et al., 2015]: This work is a considerable extension of our conference paper [Gallardo et al., 2015] in four ways. The first way is to extend the solutions and the theoretical analysis to all sub-cases of Curve SfT; in [Gallardo et al., 2015], only $\mathrm{SfT}^{1\to2\to1}$ was addressed. The second way is our discrete graphical method that can generate all candidate solutions; in [Gallardo et al., 2015], only a single solution could be generated. The third way is an improved method to detect critical points which has better stability than the method in [Gallardo et al., 2015]. The fourth way is a larger quantitative evaluation on real and simulated datasets.

Paper layout: The paper is organized as follows. In §II we give further background details on SfT methods and 3D curve reconstruction from images in general. In §III, we model Curve SfT and develop its theoretical study. As $\mathrm{SfT}^{1\to3\to2}$ and $\mathrm{SfT}^{1\to2\to1}$ are solved with the same IVP, we first study $\mathrm{SfT}^{1\to3\to2}$ and then specialize the study to $\mathrm{SfT}^{1\to2\to1}$. In §IV, we discuss the degeneracies and number of solutions for special scenarios of Curve SfT. In §V, we give our multi-solution reconstruction method based on HMM (category *(iv)* method). We also present the refinement (category *(iii)* method) and the single-solution methods (category *(i)* and *(ii)* methods). In §VI, we validate our category *(iv)* method with and without refinement on simulated and real datasets, for both instances of Curve SfT.

Notation guide: We use bold fonts for vectors and consider them by default as column vectors. The transpose of the vector $\mathbf{x}$ is denoted $\mathbf{x}^\top$. By default, we denote x_i the i^{th} component of vector $\mathbf{x}$. We use hats for estimates. Homogeneous coordinates are written with a bar, for instance $\bar{\mathbf{q}} = \begin{pmatrix}\mathbf{q}^\top & 1\end{pmatrix}^\top$. We use Greek letters for functions. The first and second derivatives of a scalar function $\varphi : \mathbb{R}^n \to \mathbb{R}$ are written with primes (*e.g.* φ') and double-primes (*e.g.* φ''). For a vector-valued function $\varphi : \mathbb{R}^n \to \mathbb{R}^m$, we use the Jacobian matrix denoted $\mathbf{J}_\varphi$. For the special case where $\varphi : \mathbb{R} \to \mathbb{R}^m$, we use the Hessian matrix denoted $\mathbf{H}_\varphi$.

II. State-of-the-Art

We first review the SfT state-of-the-art and complement it with some background details on the reconstruction problem of 3D curves from images.

A. Shape-from-Template

Various different cases of SfT exist, Curve SfT, Surface SfT and Volume SfT, but all share the same components. We then discuss in detail the three main components shared by all SfT methods: the template specifications, the data constraints extracted from the input image and the 3D shape inference process.

1) Template Components: The template is the cornerstone of SfT. It brings strong object-specific prior knowledge to the problem. Several types of template exist, but all of them comprise three components: a *shape model*, an *appearance model* and a *deformation model*.

a) Shape Model: The template's shape model represents the object's 3D shape in a fixed reference position. The shape model can be acquired with various ways depending on the application, including SfM methods such as [Wu, 2011; Agisoft, 2014], or structured-light methods such as [David 3D Scanner, 2014], or from a 3D CAD model database such as [TurboSquid, 2016; Warehouse, 2016]. There are two main types of shape models. The first use surface templates [Salzmann and Fua, 2011; Brunet et al., 2014; Collins and Bartoli, 2014; Bartoli et al., 2015; Ngo et al., 2016], where only the object's surface is modeled. The second are with volume templates [Parashar et al., 2015], where the object's surface and interior volume are modeled. Surface templates are the most common and give good approximations for thin or hollow surfaces made for example of paper, cloth and plastic. Surface templates have varied in complexity. The earliest model used algebraic models such as smooth B-splines [Brunet et al., 2014] or thin-plate splines [Chhatkuli et al., 2017; Bartoli et al., 2015]. Most recent models include triangulated meshes, which are conceptually simple, can handle general topologies [Salzmann and Fua, 2009; Ngo et al., 2015; Yu et al., 2015; Collins and Bartoli, 2015; Ngo et al., 2016], and work for surface and volume templates. An important point of mesh models is the trade-off between the density of the model and the computational time for shape inference.

b) Appearance Model: The appearance model is used to describe the photometric appearance of the object. In nearly all cases of Surface and Volume SfT, this is done using a texture-map [Bartoli et al., 2015; Chhatkuli et al., 2017; Salzmann and Fua, 2011; Ngo et al., 2016; Collins and Bartoli, 2015]. A texture-map models the intensity or color of each surface point up to photometric transforms caused by illumination, shading variations, and other photometric factors. In most previous works, the texture-map is generated from one or more images of the object in its reference position [Collins et al., 2014], but it can also come from a CAD model [Collins and Bartoli, 2015].

c) Deformation Model: The deformation model is used to define the transformation of the template's reference shape and the space of possible deformations. Most methods constraint the solution to be in the space of smooth deformations thanks to implicit and explicit smoothing and all methods use physical priors[1] using mathematical models inspired by physical laws.

Implicit and explicit smoothings. Implicit smoothing include thin-plate splines and B-splines, and reduce dimensionality of the deformation space by modeling deformation with a reduced set of control points. For mesh-based shape models, deformation smoothness has been introduced through the mesh laplacian [Sorkine and Alexa, 2007]. In [Ngo et al., 2016], this was used both for smoothing and dimensionality reduction. The idea was to identify the smooth deformation modes (which correspond to eigenvectors with lowest eigenvalues) by performing a modal analysis on the mesh laplacian. For some methods which solve the reconstruction problem by minimizing a cost function, smooth deformations are explicitly enforced with a smoothing term based on an ℓ_2 norm [Brunet et al., 2014; Bartoli and Özgür, 2016]. This norm strongly penalizes non-smooth deformations.

Physical prior: isometry. Isometry and quasi-isometry are the most commonly used priors [Salzmann and Fua, 2011; Chhatkuli et al., 2017; Collins and Bartoli, 2014; Bartoli et al., 2015; Liu-Yin et al., 2016]. They enforce metric constraints by preventing deformations that locally stretch or shrink the object. Isometry means that the geodesic distance between two points on the surface is preserved by deformation. Isometry is also equivalent to saying that the surface's first fundamental form is preserved by deformation. Isometry can be imposed exactly, which means no stretching or shrinking is permitted. Isometry can also be imposed inexactly, meaning that there is non-negligible stretching or shearing, and the model penalizes solutions with increased stretching or shrinking using a penalty function. This is also called *quasi-isometry* in the literature. Isometry and quasi-isometry have been used extensively because they dramatically restrict the solution space, and are applicable for many object classes such as those made of thick rubber, tightly-woven fabrics, paper, cardboard and plastics. The isometric prior is very powerful, and has been shown that if it is imposed exactly and if correspondences are dense, then the problem can be solved uniquely [Bartoli et al., 2015]. The main difficulty with isometry is that it is a non-convex constraint.

Physical prior: inextensibility. This is a relaxation of the isometry prior. It prevents the Euclidean distance between two neighboring surface points from exceeding their geodesic distance, defined on the template. The advantage of the inextensibility constraint is that it is a convex constraint. However, it is too weak to reconstruct geometry accurately and must be combined with additional constraints. This has been done previously using the so-called MDH [Perriollat et al., 2011; Salzmann and Fua, 2009], where a depth maximization constraint is imposed to prevent the reconstructed surface from shrinking arbitrarily. The MDH has been shown to produce very good reconstructions when the perspective effects of the camera are strong.

[1] which is different from statistics-based priors such as the morphable face models of [Blanz and Vetter, 1999]

Other physical priors. Weaker physical priors have also been considered to handle objects that can stretch or shrink as they deform. Examples include the conformal prior (angle preservation) [Malti and Bartoli, 2014; Bartoli et al., 2015] or priors based on elasticity [Malti et al., 2013; Haouchine et al., 2014; Malti et al., 2015; Özgür and Bartoli, 2017]. The problem with using these weaker physical priors is that the SfT problem becomes less well-conditioned. For instance, SfT with the conformal prior is solvable up to a global scale factor and convex/concave ambiguities.

2) Data Constraints in SfT: Data constraints must be extracted from the input image in order to match the template's shape with the object's true shape. By far the most common are motion constraints. Other constraints include contour and shading.

a) Motion Constraints: Motion constraint can be broken down in two types: *correspondences constraints* and *direct constraints*.

Correspondences constraints. Correspondence constraints force 3D points on the template's surface to project at their corresponding 2D points in the input image [Bartoli et al., 2015; Salzmann and Fua, 2011; Brunet et al., 2014; Ngo et al., 2016]. The points used by these constraints are usually obtained by matching features from the template's texture-map and the input image. These constraints have been exploited in various ways: through zeroth-order correspondences [Salzmann and Fua, 2011; Brunet et al., 2014; Ngo et al., 2016], first-order correspondences [Bartoli et al., 2015] or second-order correspondences [Bartoli and Özgür, 2016].

A zeroth-order correspondence is used to constrain the position of a point on the template in camera coordinates. Usually this is implemented using the reprojection error of the correspondence. However, this is a non-convex constraint. It is possible to construct a convex zeroth-order constraint by imposing it in camera coordinates. Zeroth-order correspondences are usually computed using feature-based matching. First-order correspondences require knowing both the position of the correspondence and the local affine transform about the correspondence. A first-order correspondence is used to constrain both the position of a point on the template in camera coordinates, and also first-order properties of the deformation at that point. In practically all cases, the first-order properties relate to isometric deformation which states that the first fundamental form is preserved. We discuss this further in §II-A3. First-order correspondence can be computed in two ways. The first is with a differentiable warp fitted between the template's texture-map and the input image. First-order correspondence can then be computed at any given point by differentiating the warp. The second way is by fitting a local first-order differentiable warp at each correspondence [Collins and Bartoli, 2014]. The main difficulty with first-order correspondences is the need to compute first-order warp derivatives, which is less numerically stable than computing zeroth-order correspondences. Second-order correspondences have also been recently considered [Bartoli and Özgür, 2016] to handle non-isometric deformations.

Correspondence constraints have three main limitations. First, they work well only for densely-textured objects with discriminative texture, which may be a limitation for some man-made objects and natural objects that usually have very weak texture. Second, feature-based matching methods may fail to establish correspondences without errors. Third, the computational time to extract features, compute descriptors and perform the matching can be long without high performance GPUs.

Direct constraints. Direct constraints work by maximizing the photometric agreement, *i.e.* brightness constancy, between the deformed template and the input image [Malti et al., 2011; Yu et al., 2015; Ngo et al., 2015; Collins and Bartoli, 2015]. The main advantage of direct constraints is to provide denser motion constraints than feature correspondences. However, direct constraints present three main limitations: they are highly non-convex, require a good initial estimate and are hard to use in presence of strong photometric changes (induced by complex deformations or complex illuminations) or occlusions.

b) Other Constraints: In SfT, two other constraints have been used: contour constraints [Salzmann et al., 2007; Vicente and Agapito, 2013; Gallardo et al., 2016] and shading constraints [Malti and Bartoli, 2014; Liu-Yin et al., 2016; Gallardo et al., 2016]. Contour constraints force the object's occluding contours align to the corresponding contours in the input image. These constraints do not depend on the template's texture, but may be difficult to use in presence of background clutter or strong template's texture. Shading constraints ensure the respect of the photometric relationship between surface geometry, surface reflectance, illumination, the camera response and pixel intensity. These constraints give dense constraints and work on poorly-textured and textureless surfaces. However, they may be difficult to use in practice because they require good photometric modeling and calibration. Both constraints are highly non-convex constraints and require a good initial estimate. They are also only applicable for surface and volume templates.

3) Inference in SfT: Inference is performed by determining the deformation parameters that mutually satisfy the data constraints and deformation priors. Three inference categories have emerged and we illustrate them in figure 2.

a) Category (i): local analytical solutions using non-holonomic solution to PDE: Category *(i)* methods impose the constraints through a PDE system and solve using non-holonomic solutions. This approach has been used by [Bartoli et al., 2015; Chhatkuli et al., 2017] with the isometric prior and first-order correspondence constraints. Correspondence constraints were computed with a differentiable template-to-image warp which maps the template's texture-map to the input image. With the template-to-image warp, [Bartoli et al., 2015] constructs a first-order PDE system and solves it at each point assuming that the depth and its gradient are independent. These solutions obtained are called non-holonomic solutions. [Bartoli et al., 2015] only uses the depth solution, however, it suffers from instabilities when the projection geometry tends to affine. Figure 2 gives a reconstruction performed by [Bartoli et al., 2015]. [Chhatkuli et al., 2017] solves this issue by improving the stability using the non-holonomic solution of both depth and gradient, which is proven to be stable for both perspective and affine projections. The main advantage of category *(i)* methods is that, as they give an analytical solution at each correspondence, they are fast and they can be parallelized extremely well. The solutions can be used as initial estimate for non-convex refinement which we describe below. The main disadvantage is the need for accurate first-order correspondences.

b) Category (ii): convex optimizations using inextensibility: Category *(ii)* methods work by approximating the inference problem with a convex function. The main idea is to deal with a simpler constraint than isometry which is non-convex in order to obtain quasi-isometry. One solution is to relax isometry to inextensibility, which leads to a convex constraint easier to handle. However, inextensibility is insufficient since a trivial solution is given by putting all correspondences at the camera origin. To prevent this solution, the depths of the correspondences are maximized while simultaneously satisfying inextensibility and zeroth-order correspondence constraints. This technique, called MDH [Perriollat et al., 2011; Salzmann and Fua, 2009; Brunet et al., 2014], shows that it can provide solutions that are often quasi-isometric. Figure 2 gives a reconstruction performed by [Salzmann and Fua, 2009]. Two versions of MDH have been proposed and they differ from the way they solve the problem. The first version uses a fast greedy technique [Perriollat et al., 2011]. The second version is based on the remark that the MDH can be formulated as a Second-Order Cone Programming (SOCP) problem [Salzmann and Fua, 2009; Brunet et al., 2014], and therefore solves globally using efficient methods such as interior point. Despite the improvements of the original formulation [Salzmann and Fua, 2009; Brunet et al., 2014], category *(ii)* methods remain using

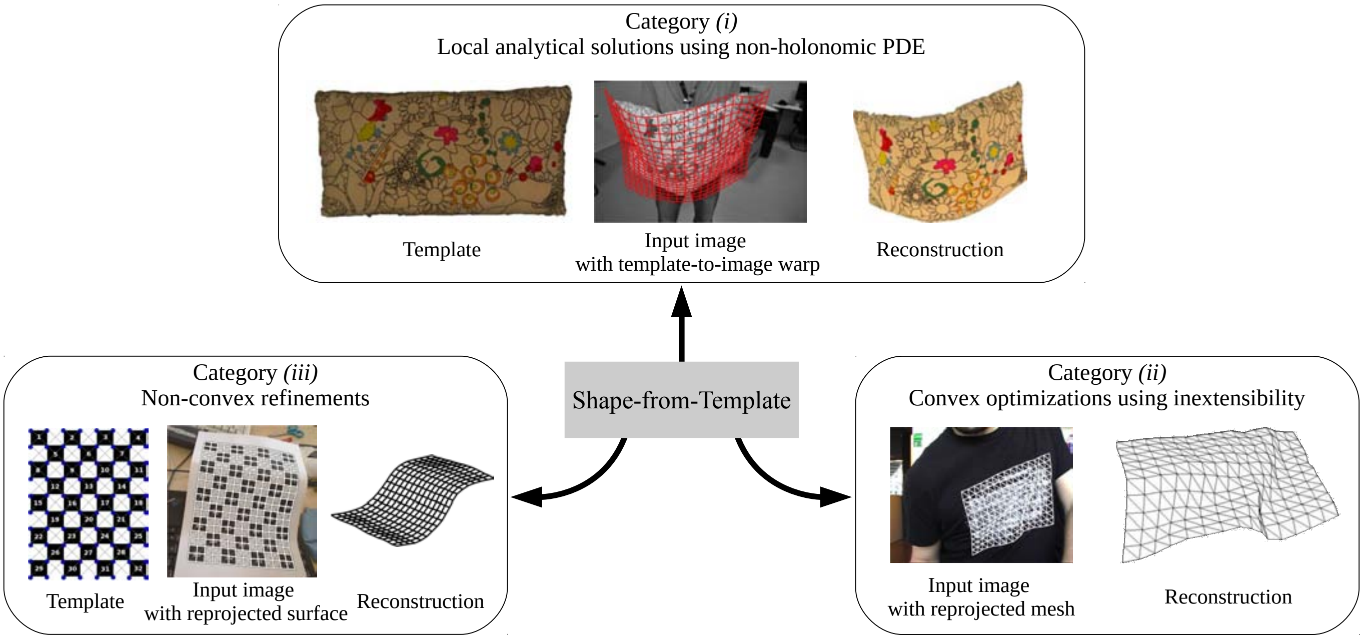


Fig. 2. Illustration of the different categories of SfT methods. Results are taken from the following works (from category *(i)* to category *(iii)*): [Bartoli et al., 2015], [Salzmann and Fua, 2009] and [Brunet et al., 2014].

a relaxation of the isometry, which sacrifices accuracy for obtaining a global solution.

c) Category (iii): non-convex refinements: The third category works by combining the data and prior constraints into a single optimizable non-convex cost function. In general, because of the physical priors, this cost function is non-convex and thus is solved by gradient-based minimization such as Levenberg-Marquardt [Brunet et al., 2014; Liu-Yin et al., 2016] or Gauss-Newton [Ngo et al., 2015; Collins and Bartoli, 2015]. The solutions are holonomic, *i.e.* deformation and deformation derivatives are dependent, which ensures a better conditioning and stability. The advantages of this category are two-fold: there is no relaxation of the physical priors and complex constraints such as shading can be integrated without difficulty. However, category *(iii)* methods present two main challenges: they are non-convex, they may require more computational time than categories *(i)* and *(ii)* methods. They also require to find good weights of the different constraints. To ensure good convergence, they generally require a reasonably accurate initialization. This can be provided by a category *(i)* or *(ii)* method. Also, they may require some optimization techniques to improve the convergence, such as coarse-to-fine optimization [Collins and Bartoli, 2015]. Examples of category *(iii)* methods are [Malti et al., 2011; Malti and Bartoli, 2014; Brunet et al., 2014; Collins and Bartoli, 2015; Ngo et al., 2015; Yu et al., 2015; Liu-Yin et al., 2016].

B. Reconstruction of 3D Curves from Images

The problem of 3D curve reconstruction has been addressed in the two last decades through several approaches [Faugeras and Papadopoulo, 1993; Berthilsson et al., 2001; Mai and Hung, 2010; Martinsson et al., 2007]. These works all assume the curves to be rigid and use multiple images. The images may come from a pair of images [Sbert and F. Sol, 2003], an unorganized set [Berthilsson et al., 2001; Mai and Hung, 2010; Martinsson et al., 2007; Kahl and August, 2003] or a monocular video sequence [Faugeras and Papadopoulo, 1993].

[Sbert and F. Sol, 2003] formulates the problem of curve stereo-reconstruction using an energy functional and obtains using the Euler-Lagrange equations a PDE which is solved using a level-set approach. One drawback is that it requires to know the camera projections.

[Berthilsson et al., 2001] proposes an affine shape method for 3D curves, which optimizes a subspace constraint to align the parameterizations of matched curves over a set of images. It also extends the method of bundle adjustment to 3D curves. [Mai and Hung, 2010] proposes a point-based 3D curve reconstruction method from multiple images taken by uncalibrated cameras. It selects one image as reference, selects a fixed number of 2D 'representative' points along the curve in the reference image and finds their matches in the other images. Then, it minimizes the reprojection error of the 'representative' points by adjusting alternatively the camera projection and the 'representative' points in 2D and 3D. In practice, it requires planar surfaces, which restricts significantly the range of recoverable curves. [Martinsson et al., 2007] reconstructs the curve in 3D for an object observed in a set of images using a CAD model of this object. It proposes a two-stage adaptive reconstruction method which uses 3D NURBS to parameterize the curve. First, it optimizes the 3D NURBS curve with a fixed number of control points by minimizing image contours and gradient intensity constraints. Second, it inserts new control points using the same energy terms and a stop-criterion in order to capture high curvature without over-parameterizing the 3D curve. [Kahl and August, 2003] solves simultaneously the matching and the reconstruction problems. For this, it proposes an optimization problem for a generative model of space curves with two components: a smoothness prior of the curves and a formation model of image curves. Initial 3D curves are estimated using an edge-based trinocular stereovision method [Robert and Faugeras, 1991] and considered as hypotheses regarding the generative model. All hypotheses are then independently refined using the generative model and kept if they respect a set of image conditions. [Kahl and August, 2003] uses B-splines to parameterize the curves and gradient descent for the optimization. One shortcoming is that the relative motion of the cameras is known in order to generate from the 3D curve the 2D curve in each image.

[Faugeras and Papadopoulo, 1993] provides a thorough theoretical study of the structure and the motion of 3D curves: it gives the assumptions under which they can be recovered from a sequence of calibrated images. As it relies on high order spatio-temporal derivatives, the method suffers in practice from numerical problems.

Even if these methods reconstruct 3D curves, their assumptions significantly differ from the ones of Curve SfT, which reconstructs a deformable curve from a single image and a curvilinear template.

III. Problem Modeling and Theoretical Analysis

We now study the two main cases of the Curve SfT problem, referred as $\mathrm{SfT}^{1\to3\to2}$ and $\mathrm{SfT}^{1\to2\to1}$. $\mathrm{SfT}^{1\to3\to2}$ is when the template is a curve embedded in the 3D space and observed by a regular 2D camera. $\mathrm{SfT}^{1\to2\to1}$ is similar to $\mathrm{SfT}^{1\to3\to2}$, but the camera is 1D. We first specify both cases in two particular instances and then propose a theoretical analysis of both instances. For sake of simplicity, we refer both instances respectively by $\mathrm{SfT}^{1\to3\to2}$ and $\mathrm{SfT}^{1\to2\to1}$.

A. $\mathrm{SfT}^{1\to3\to2}$ and $\mathrm{SfT}^{1\to2\to1}$: Two Instances of Curve SfT

We form here the two problem instances of $\mathrm{SfT}^{1\to3\to2}$ and $\mathrm{SfT}^{1\to2\to1}$ using eight problem components and giving the reasons of each component specification. The reason to define these instances is to consider the important problem configurations and to propose a formal and clear definition of the problems which we study.

1) The SfT$^{1\to3\to2}$ instance: (a) Models. In order to solve Curve SfT, two fundamental models are required: the *template* and the *camera projection* model. We use a 1D general function for the template's shape. We model the template's appearance with a known texture-map which is however unused because we consider motion to be given *a priori*, as we state in the component *(f)*. Deformation is modeled quasi-isometrically. The *camera projection* model determines how to reproject the 3D points used by the motion constraint, as explained in §II-A2a. We assume the perspective camera model [Hartley and Zisserman, 2003], which handles well most real-world cameras. *(b) Exploited visual cues.* We use only motion visual cue based on correspondences because it is the main visual cue used in SfT. *(c) Number of required images.* A single image is required because we want to tackle the classical version of SfT. *(d) Expected types of deformations.* We assume quasi-isometric and no tearing. *(e) Scene geometry.* We assume no self or external occlusions, which is a typical assumption in the SfT state-of-the-art. There can be background clutter. *(f) Requirement for correspondences.* We assume to know *a priori* a set of 1D-2D correspondences from the texture-map of the template to the input image. We assume that these are sufficiently dense along the template. *(g) Surface texture characteristics.* We consider well-textured surfaces since it is an usual assumption in SfT. *(h) Known and unknown model parameters.* A template of the surface, as defined in §II-A1, and the camera intrinsics are known. The unknowns are the 3D points of the deformed template in 3D camera coordinates.

2) Specialization to SfT$^{1\to2\to1}$: SfT$^{1\to2\to1}$ differs from SfT$^{1\to3\to2}$ in three components, *(a)*, *(f)* and *(h)*. For *(a)*, SfT$^{1\to2\to1}$ adapts the perspective camera model to a 2D projection. The input image is 1D. For *(f)*, SfT$^{1\to2\to1}$ assumes 1D-1D correspondences between the texture-map and the input image. For *(h)*, the unknowns of SfT$^{1\to2\to1}$ are the 2D points of the deformed template in 2D camera coordinates.

B. SfT$^{1\to3\to2}$: Reconstructing a 3D Curve from a 2D Image and a 1D Template

1) Template and Camera Modeling: The known template is 1D and we write it as $\mathcal{T} \subset \mathbb{R}$. We assume the template is deformed into a smooth unknown curve $\mathcal{S} \subset \mathbb{R}^3$ embedded in 3D. We denote the embedding function that generates $\mathcal{S}$ by $\varphi = (\varphi_x \; \varphi_y \; \varphi_z)^\top \in C^\infty(\mathcal{T}, \mathbb{R}^3)$. The 2D input image $I \subset \mathbb{R}^2$ is a perspective projection of $\mathcal{S}$. We model projection by the pinhole camera Π:

$$\Pi(\mathbf{Q}) = \left(\frac{x}{z} \; \frac{y}{z}\right)^\top \quad \text{where} \quad \mathbf{Q} = (x \; y \; z)^\top. \tag{1}$$

The pinhole camera can be used for general perspective cameras when lens distortion has been corrected and the intrinsic calibration matrix has been standardized to $\mathbf{I}_3$ [Hartley and Zisserman, 2003].

2) Inputs and Outputs: We now give our inputs. *(i)* one RGB input image $I : \mathbb{R}^2 \to \{0, 255\}^3$ showing a deforming curve. *(ii)* a template of the curve, defined using §III-B1. *(iii)* the camera intrinsics of the perspective 3D projection function Π. *(iv)* a set of N 1D-2D correspondences from the texture-map of the template to the input image. We denote the set by $\mathcal{S}_c = \{(u_k, \mathbf{q}_k)\}$ where u_k denotes the correspondence position in $\mathcal{T}$ and $\mathbf{q}_k$ denotes the correspondence position in the input image I. Details for how correspondences are computed for our experimental datasets are given in §VI-B2. We define $\eta \in C^\infty(\mathcal{T}, \mathbb{R}^2)$ as the *template-to-image warp*. As $\mathcal{S}$ has no self-occlusions in I, so η is bijective. As the set of N 1D-2D correspondence points are known between the template and the input image and as these are sufficiently dense, η can be estimated through a smooth interpolation. In practice, we use B-splines to model η.

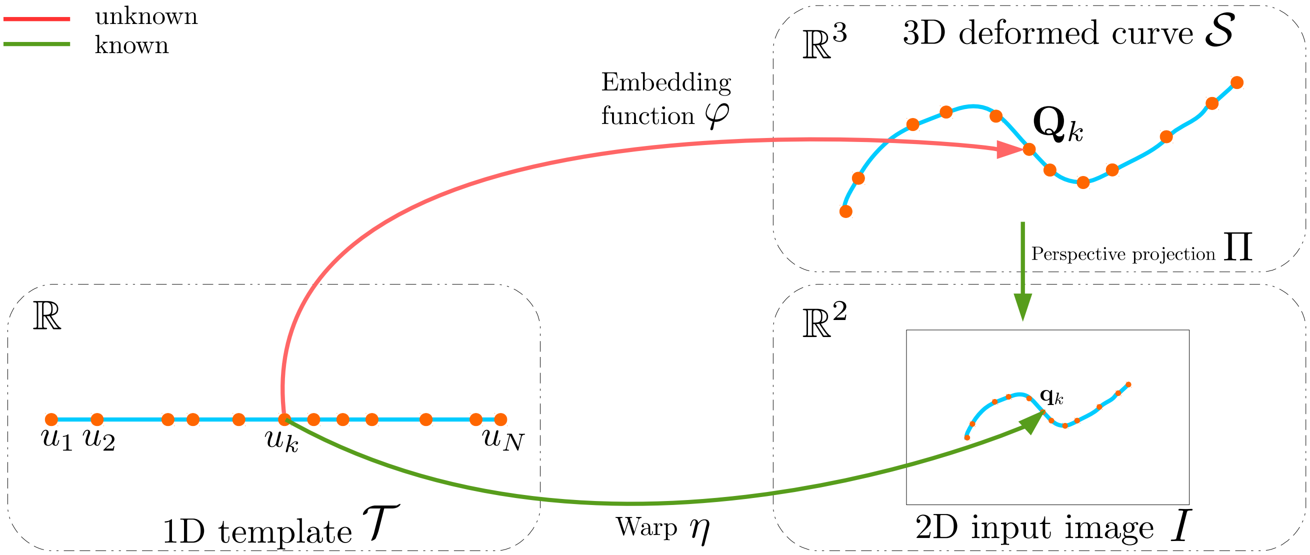


Fig. 3. General modeling of $\mathrm{SfT}^{1\to3\to2}$.

Our solution to $\mathrm{SfT}^{1\to3\to2}$ outputs the 3D points of the deformed template in 3D camera coordinates.

3) Theoretical Analysis: We now express mathematically $\mathrm{SfT}^{1\to3\to2}$ and then propose a differential analysis of the problem. Our geometric modeling is shown in figure 3. It is inspired from [Bartoli et al., 2015], for solving Surface SfT with continuous differential geometry. We recall that we study $\mathrm{SfT}^{1\to3\to2}$ (and $\mathrm{SfT}^{1\to2\to1}$) under the assumption of known correspondences between the template and the input image. This is a reasonable assumption and a mandatory move to understand the theory behind this problem.

a) Problem formulation: $\mathrm{SfT}^{1\to3\to2}$ involves recovering φ, from the warp η and the projection Π. This is constrained by the *isometry* prior and the *reprojection* constraints implied by η. The warp $\eta \in C^{\infty}(\mathcal{T}, \mathbb{R}^2)$ maps the template to a 2D input image. The reprojection constraint is therefore:

$$\eta = \Pi \circ \varphi. \tag{2}$$

The isometry constraint forces the geodesic distance to be preserved between $\mathcal{T}$ and $\mathcal{S}$. It is a first order differential property in φ:

$$\mathbf{J}_{\varphi}^{\top}\mathbf{J}_{\varphi} = 1. \tag{3}$$

From the constraints (2) and (3), we define $\mathrm{SfT}^{1\to3\to2}$ as follows:

$$\text{Find} \quad \varphi \in C^{\infty}(\mathcal{T}, \mathbb{R}^3) \quad \text{s.t.} \quad \begin{cases} \eta = \Pi \circ \varphi & \text{(reprojection)} \\ \mathbf{J}_{\varphi}^{\top}\mathbf{J}_{\varphi} = 1 & \text{(isometry).} \end{cases} \tag{4}$$

b) ODE formulation: We show that equation (4) is equivalent to finding the solution of a first-order non-linear ODE. Using equation (1), we first transform the reprojection constraint into:

$$\varphi = \varphi_z \bar{\eta}, \tag{5}$$

where φ_z is the depth component of φ and $\bar{\eta}$ is the warp in homogeneous coordinates. We differentiate equation (5) once, giving:

$$\mathbf{J}_\varphi = \varphi_z \mathbf{J}_{\bar{\eta}} + \varphi_z' \bar{\eta}. \tag{6}$$

We then substitute equation (6) into the isometry constraint from equation (4) and obtain a first-order non-linear ODE with φ_z as the unknown variable:

$$\varphi_z'^2 \|\bar{\eta}\|^2 + 2\varphi_z \varphi_z' \bar{\eta}^\top \mathbf{J}_{\bar{\eta}} + \varphi_z^2 \mathbf{J}_{\bar{\eta}}^\top \mathbf{J}_{\bar{\eta}} = 1. \tag{7}$$

Using the identities $\bar{\eta}^\top \mathbf{J}_{\bar{\eta}} = \eta^\top \mathbf{J}_\eta$ and $\mathbf{J}_{\bar{\eta}}^\top \mathbf{J}_{\bar{\eta}} = \mathbf{J}_\eta^\top \mathbf{J}_\eta$, we arrive at:

$$\varphi_z'^2 \|\bar{\eta}\|^2 + 2\varphi_z \varphi_z' \eta^\top \mathbf{J}_\eta + \varphi_z^2 \mathbf{J}_\eta^\top \mathbf{J}_\eta = 1. \tag{8}$$

Equation (8) is a first-order ODE. Given a solution to equation (8), the problem is solved and φ can be found as $\varphi = \varphi_z \bar{\eta}$. We now propose a change of variable that greatly simplifies equation (8) and the study of its solutions.

c) Change of variable: We define the function $\varepsilon = \|\bar{\eta}\|$, where $\varepsilon' = \frac{1}{\varepsilon} \eta^\top \mathbf{J}_\eta$. We now define a new scalar function θ with:

$$\theta = \varphi_z \varepsilon. \tag{9}$$

θ can be interpreted as the distance of the curve point to the center of projection. Equation (8) can now be rewritten in terms of θ and θ':

$$\theta'^2 + \xi\theta^2 = 1 \quad \text{with} \quad \xi = \frac{1}{\|\bar{\eta}\|^2} \left(\mathbf{J}_\eta^\top \mathbf{J}_\eta - \frac{1}{\|\bar{\eta}\|^2} \mathbf{J}_\eta^\top \eta \eta^\top \mathbf{J}_\eta \right). \tag{10}$$

We use equation (10) to study the local solvability and the solution space of SfT$^{1\to3\to2}$. This leads to two important results: *(i)* SfT$^{1\to3\to2}$ is not locally solvable exactly and *(ii)* there exist necessary and sufficient conditions for solving SfT$^{1\to3\to2}$ up a discrete number of ambiguities.

d) Local exact solutions: We explore whether local solutions of equation (10) exist using non-holonomic solution analysis. Non-holonomic solutions are based on the creation of new equations by differentiation and a relaxation of the differential dependencies. In our case, this means treating θ and θ' as independent variables. The results are called non-holonomic solutions [Eliashberg and Mishachev, 2002]. The uniqueness of non-holonomic solutions to PDE implies the uniqueness of solutions to the original ODE. Our main motivation is historical as non-holonomic solutions were used successfully in Surface SfT [Bartoli et al., 2015] to prove well-posedness and to give analytic solutions. We prove the following proposition:

Proposition 1 (Impossibility of non-holonomic solutions)**.** *The non-holonomic solution for φ_z in equation* (8) *is under-constrained for any order of differentiation.*

Proof. Equation (10) gives a single constraint for the two unknowns, θ and θ'. Differentiating equation (10) creates extra equations. Differentiating $k-1$ times yields k equations, however, each differentiation introduces one new unknown. For order k, we have a total of $k+1$ unknowns: $\theta, \theta', \ldots, \theta^{(k)}$. As a consequence we have k equations and $k+1$ unknowns, and the problem is under-constrained for any order $k > 0$. □

This proposition is important because it proves that local non-holonomic solutions to the ODE (10) do not exist.

e) Solution space: We now study the *global* solutions to $\text{SfT}^{1\to 3\to 2}$. In general, a single ODE such as equation (10) has an infinite number of solutions. The true curve is one of these solutions. We construct an IVP by adding an initial condition $\theta(u_0) = \theta_0$ to the ODE (10). Our IVP writes as:

$$\begin{cases} \theta'(u) = \Psi(u, \theta(u)) = \sqrt{1 - \xi(u)\theta^2(u)} \\ \theta(u_0) = \theta_0. \end{cases} \tag{11}$$

The value $\theta_0 \in \mathbb{R}^*$ at u_0 is called the *anchor point*. We now use the following two properties.

Proposition 2 (Number of solutions to IVP (11))**.** *For the given IVP* (11)*,*

- *when* $\theta'(u_0) \neq 0$*, the IVP* (11) *has two solutions in a local interval of* u_0
- *when* $\theta'(u_0) = 0$*, the IVP* (11) *has at most two solutions in a local interval of* u_0*.*

Proof. When $\theta'(u_0) \neq 0$, we apply the Picard-Lindelöf theorem and obtain that the IVP (11) has two solutions in a local interval of u_0. This theorem is applicable to the IVP (11) because this IVP respects the Picard-Lindelöf conditions: the function Ψ is uniformly Lipschitz continuous in θ and continuous in u.

When $\theta'(u_0) = 0$, the number of solution is given by [Casillas-Perez and Pizarro, 2017]: there are at most two solutions in a local interval of u_0. The solution space is thus bounded if the anchor point is available. Here, we use the theorem given and proved in [Casillas-Perez and Pizarro, 2017] instead of the Picard-Lindelöf theorem which cannot be applied in this case. This is because, when $\theta'(u_0) = 0$, the function Ψ is not Lipschitz continuous in θ. To solve this, [Casillas-Perez and Pizarro, 2017] shows that at most two analytical solutions can be constructed in a local interval of u_0 such that their first-order derivative is not null at u_0, which allows one to apply the Picard-Lindelöf theorem. □

However, in practice we do not have an anchor point to form the IVP (11), making this approach impractical. We propose a strategy to find solutions of the ODE (10), which obtains very good candidates for the true curve without explicitly needing anchor points. Our strategy is based on finding the so-called *critical points*. Critical points have special geometric properties, especially a unique solution to depth and normal direction, that we exploit to find the true curve.

f) Critical points: We now give the formal definition of critical points and their properties.

Definition 1 (Critical point definition in θ)**.** *Given a solution* $\hat{\theta}$ *of equation* (10)*,* $u_c \in \mathcal{T}$ *is a critical point of* $\hat{\theta}$ *if and only if* $\hat{\theta}'(u_c) = 0$*. Equivalently, from equation* (10)*,* $u_c \in \mathcal{T}$ *is a critical point of* $\hat{\theta}$ *if and only if* $\hat{\theta}^2(u_c)\xi(u_c) = 1$*.*

Proposition 3 (Critical point definition in φ)**.** *Given a solution* $\hat{\varphi}$ *to equation* (4)*,* $u_c \in \mathcal{T}$ *is a critical point if and only if* $\hat{\varphi}^\top(u_c)\,\mathbf{J}_{\hat{\varphi}}(u_c) = 0$*. An intuitive interpretation is that a critical point is the point on the curve where the tangent and the optical ray are orthogonal.*

Proposition 3 is proved in appendix A. It gives a geometric interpretation of the critical points: they represent the points of the curve $\mathcal{S}$ where the tangent and the optical ray are mutually orthogonal. In addition, and directly from definition 1, we have that for a critical point u_c:

$$\hat{\theta}(u_c) = \frac{1}{\sqrt{\xi(u_c)}}. \tag{12}$$

g) Computing super critical points from the ODE: A critical point cannot be determined prior to reconstruction. For instance, from definition 1, one has to know $\hat{\varphi}$, and thus the function $\hat{\theta}$, in order to compute the critical points. For this, we propose to compute a superset of the critical points as this set contains all the critical points of $\hat{\varphi}$. To find this superset, we use equation (10) and define the *super curve*, denoted by φ_s, as follows.

Definition 2 (Super curve)**.** *The super curve $\varphi_s \in C^\infty(\mathcal{T}, \mathbb{R}^3)$ of the ODE* (10) *is defined as $\varphi_s = \frac{1}{\varepsilon\sqrt{\xi}}\bar{\eta}$.*

The super curve is obtained by setting $\theta' = 0$ in equation (10) and solving the resulting algebraic equation for θ:

$$\xi\theta_s^2 = 1 \qquad \Rightarrow \qquad \theta_s = \frac{1}{\sqrt{\xi}}. \tag{13}$$

The *super curve* is not a solution of the ODE (10) except for those points where $\theta_s' = 0$, which form the set of *super critical points*.

Proposition 4 (The set of super critical points)**.** *The set of all critical points contained in any solution of the ODE* (10) *belongs to the set of critical points of its super curve φ_s. We name this set the super critical point set.*

Proposition 4 is proved in appendix B. We now give a way to characterize the super critical points. Precisely, we give three equivalent characterizations. We use them to investigate different methods of super critical point detection.

Proposition 5 (Super critical point identities)**.** *The following are necessary and sufficient conditions for u_s being a super critical point:*

$(\xi'(u_s) = 0) \Leftrightarrow$

$\Big(\|\bar{\eta}(u_s)\|^4 \mathbf{J}_\eta^\top(u_s)\mathbf{H}_\eta(u_s) - \eta^\top(u_s)\mathbf{J}_\eta(u_s)\big(\|\bar{\eta}(u_s)\|^2\eta^\top(u_s)\mathbf{H}_\eta(u_s) +$

$2\|\bar{\eta}(u_s)\|^2\mathbf{J}_\eta^\top(u_s)\mathbf{J}_\eta(u_s) - 2\mathbf{J}_\eta^\top(u_s)\eta(u_s)\eta^\top(u_s)\mathbf{J}_\eta(u_s)\big) = 0\Big)$

$\Leftrightarrow \Big({\varphi_s}^\top(u_s)\mathbf{J}_{\varphi_s}(u_s) = 0\Big)$,

where φ_s is the super curve constructed from equation (8).

Proposition 5 is proven in appendix C.

h) Properties of the solutions of the IVP at a super critical point: Suppose we have one super critical point u_s obtained from the *super curve*. We restate the IVP (11) using the critical point as the initial condition:

$$\begin{cases} \theta'^2 + \xi\theta^2 = 1 \\ \theta(u_s) = \theta_s(u_s). \end{cases} \tag{14}$$

According to [Casillas-Perez and Pizarro, 2017], this problem has two analytical solutions in an open domain $]u_s - \epsilon; u_s + \epsilon[$ sufficiently close to the super critical point with $\epsilon > 0$. One solution is $\theta' = \sqrt{1 - \xi\theta^2}$ corresponding to the case where θ' is positive. The second solution is $\theta' = -\sqrt{1 - \xi\theta^2}$, where θ' is negative. If we combine all pairs of branches on either side of the super critical points we obtain four candidate solutions in the local vicinity of u_s, which are C^1 functions [Casillas-Perez and Pizarro, 2017].

i) Candidate solutions: Suppose that $u_{s_1}, \cdots u_{s_{N_s}}$ is the set of N_s super critical points. We define a candidate solution as the solution of the following MIVP:

$$\begin{cases} \theta'^2 + \xi\theta^2 = 1 \\ \theta(u_{s_1}) = \theta_s(u_{s_1}) \\ \vdots \\ \theta(u_{s_{N_s}}) = \theta_s(u_{s_{N_s}}). \end{cases} \tag{15}$$

We extend the theorem proposed in [Casillas-Perez and Pizarro, 2017] to the case where there are more than one super critical point. We define the interval $\mathcal{I} = [u_{s_1}; u_{s_2}]$, with u_{s_1} and u_{s_2} two consecutive super critical points.

Proposition 6 (Constant sign between two consecutive super critical points)**.** *Given two consecutive super critical points u_{s_1}, u_{s_2} and given a solution $\hat{\theta}$ of equation* (15)*, the sign of $\hat{\theta}'$ remains constant in the interval $\mathcal{I} = [u_{s_1}; u_{s_2}]$.*

Proof. $\hat{\theta}$ has continuous derivatives, $\hat{\theta}'(u_{s_1}) = 0$ and $\hat{\theta}'(u_{s_2}) = 0$ by definition 1. Therefore the function $\hat{\theta}'$ cannot change its sign in the interval $\mathcal{I}$ without passing through a super critical point, which contradicts the fact that there are no super critical points in $\mathcal{I}$. □

Using proposition 6, we obtain the following.

Proposition 7 (Number of candidate solutions between two consecutive super critical points)**.** *Given the interval $\mathcal{I} = [u_{s_1}; u_{s_2}]$ between two consecutive super critical points, there exist two candidate solutions, given by the solutions of equation* (15) *in $\mathcal{I}$.*

Proof. We use the super critical point u_{s_1} to construct an IVP and then apply the theorem from [Casillas-Perez and Pizarro, 2017] on $\mathcal{I}$. This shows the existence of two solutions in $\mathcal{I}$. Because u_{s_2} is also a super critical point, both solutions, which intersect at u_{s_1}, also pass through u_{s_2}. This is respected regardless of using u_{s_1} or u_{s_2} as initial condition. Thus, the two solutions of both IVPs are the same. □

From proposition 7 we can derive the following bound on the number of solutions of the MIVP (15), given the number of super critical points.

Theorem 1 (Number of candidate solutions)**.** *For N_s super critical points in $\mathcal{T}$, there are 2^{N_s+1} candidate solutions to equation* (15).

Proof. Proposition 7 tells us that there are two candidate solutions between two consecutive super critical points. Therefore, the solutions to the MIVP (15) are composed of pieces of curves that connect at the super critical points and correspond to all the possible combinations. For N_s super critical points, we thus have 2^{N_s+1} possible combinations. □

j) Specialization to closed curvilinear templates: All theoretical aspects previously mentioned apply to the case of closed curvilinear templates. However, for this special case, a specific theoretical result can be obtained regarding the lower bound on the number of candidate solutions.

Proposition 8 (Lower bound on the number of candidate solutions for a closed curvilinear template)**.** *If $\mathcal{T}$ is a closed curvilinear template and if $\mathcal{T}$ is deformed smoothly in $\mathbb{R}^3$ by $\varphi \in C^\infty(\mathcal{T}, \mathbb{R}^3)$, then there are at least* 2 *critical points, and so at least* 4 *candidate solutions to equation* (15)*, and the number of solutions is thus always finite.*

Proof. A closed curvilinear template $\mathcal{T}$ implies that function θ is bounded. As the function θ is continuous in $\mathcal{T}$, we can apply the extreme value theorem and obtain that function θ has at least a maximum and a minimum. Therefore, there exist at least two points $(u_1, u_2) \in \mathcal{T}^2$ such that $\theta'(u_1) = 0$ and $\theta'(u_2) = 0$, which gives from definition 1 that u_1 and u_2 are two critical points of φ. From proposition 4, u_1 and u_2 are two super critical points of φ. Thus, from theorem 1, there are at least 4 candidate solutions to equation (15) and thus a finite number of solutions. □

k) Conclusions on the theoretical analysis of SfT$^{1\rightarrow3\rightarrow2}$: We have proved for the SfT$^{1\rightarrow3\rightarrow2}$ problem all properties which we presented in §I-A. The two main outcomes of this analysis are the non-solvability of SfT$^{1\rightarrow3\rightarrow2}$ using local non-holonomic PDE and the finding of the critical points in the data itself, which sufficiently constrain the problem so it can be solved up to a finite set of candidate solutions.

C. SfT$^{1\rightarrow2\rightarrow1}$: Reconstructing a 2D Curve from a 1D Image and a 1D Template

The SfT$^{1\rightarrow2\rightarrow1}$ problem is a straightforward specialization of the SfT$^{1\rightarrow3\rightarrow2}$ problem. *This specialization does not imply that any extra constraint or information are imposed.* We illustrate the specialization in figure 4.

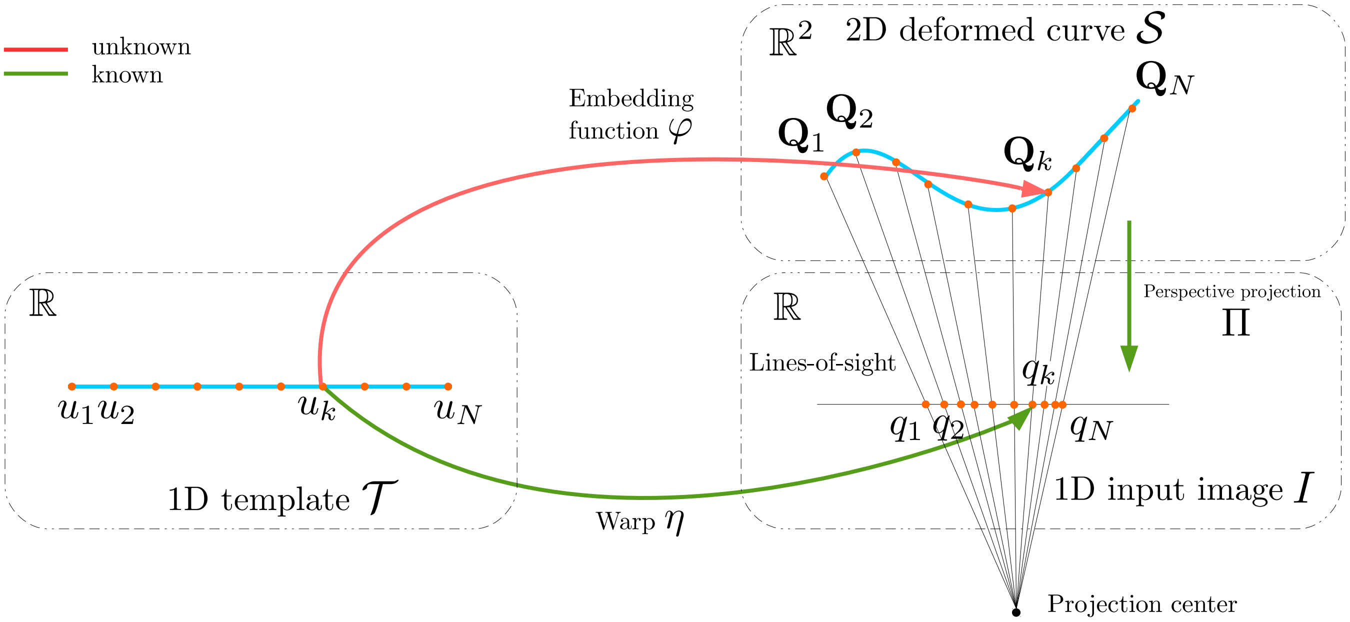


Fig. 4. General modeling of SfT$^{1\rightarrow2\rightarrow1}$.

1) Template and Camera Modeling: Both problems share the same dimension for the known template, $\mathcal{T} \subset \mathbb{R}$. However, the main difference is that the image of the functions, φ, η and Π has a dimension smaller. The template is deformed into an unknown smooth curve $\mathcal{S} \subset \mathbb{R}^2$ embedded in 2D. We parameterize the embedding function by $\varphi = (\varphi_x \; \varphi_y)^\top \in C^\infty(\mathcal{T}, \mathbb{R}^2)$.

The 1D input image $I \subset \mathbb{R}$ is modeled as the perspective projection of $\mathcal{S}$ that we denote with a canonical 1D projection function Π:

$$\Pi(\mathbf{Q}) = \frac{x}{y} \quad \text{where} \quad \mathbf{Q} = (x\ y)^\top. \tag{16}$$

2) Inputs and Outputs: We now give the inputs and the outputs of SfT$^{1\to2\to1}$. *(i)* one 1D input image $I : \mathbb{R} \to \{0, 255\}^3$ showing a deforming curve. *(ii)* a template of the curve, defined using §III-B1. *(iii)* the camera intrinsics of the perspective 2D projection function Π. *(iv)* a set of N 1D-1D correspondences from the texture-map of the template to the input image. We denote the set by $\mathcal{S}_c = \{(u_k, q_k)\}$ where u_k denotes the correspondence position in $\mathcal{T}$ and q_k denotes the correspondence position in the input image I. Details for how correspondences are computed for our experimental datasets are given in §VI-A2.

Our solution to SfT$^{1\to2\to1}$ outputs the 2D points of the deformed template in 2D camera coordinates.

3) Theoretical Analysis: We show that SfT$^{1\to2\to1}$ can be formulated as SfT$^{1\to3\to2}$ in problem (4). More specifically, it follows the same ODE, which allows us to use all properties given in §III-B3. This leads to the same conclusions as for SfT$^{1\to3\to2}$: *(i)* SfT$^{1\to2\to1}$ is not locally solvable exactly and *(ii)*, if $N_s \geq 1$, SfT$^{1\to2\to1}$ has 2^{N_s+1} candidate solutions and an infinite number of solutions otherwise. The differences with SfT$^{1\to3\to2}$ are two-fold: the formula of the ODE coefficients ξ and the critical points definition in φ. We now give these differences.

a) Problem formulation: Similarly to SfT$^{1\to3\to2}$, SfT$^{1\to2\to1}$ involves recovering φ, from the warp η and the projection Π using the isometry prior and the reprojection constraints implied by η. The warp $\eta \in C^\infty(\mathcal{T}, \mathbb{R}^1)$ maps the template to a 1D input image. The reprojection constraint is therefore:

$$\eta = \Pi \circ \varphi. \tag{17}$$

The *isometry* constraint is a first order differential property in φ:

$$\|\varphi'\|_2^2 = 1. \tag{18}$$

From the constraints (17) and (18) we define SfT$^{1\to2\to1}$ as follows:

$$\text{Find} \quad \varphi \in C^\infty(\mathcal{T}, \mathbb{R}^2) \quad \text{s.t.} \quad \begin{cases} \eta = \Pi \circ \varphi & \text{(reprojection)} \\ \|\varphi'\|_2^2 = 1 & \text{(isometry).} \end{cases} \tag{19}$$

b) ODE formulation: We obtain the ODE of SfT$^{1\to2\to1}$ in a similar way as in §III-B3. We first transform the reprojection constraint (17) into:

$$\varphi_y \eta = \varphi_x, \tag{20}$$

where φ_y is the depth component of φ. We differentiate once, giving:

$$\eta' \varphi_y + \eta \varphi_y' = \varphi_x'. \tag{21}$$

By substituting φ_x' from equation (21) in the isometry constraint (18) and expanding, we arrive at:

$$\varphi_y'^2 \eta^2 + 2\varphi_y \varphi_y' \eta \eta' + \varphi_y^2 \eta'^2 + \varphi_y'^2 = 1. \tag{22}$$

This is the same as ODE (10). We then perform the change of variable given in §III-B3c in order to study the ODE solutions.

c) Change of variable: We first define $\varepsilon = \|\bar{\eta}\|$ and thus, $\varepsilon' = \frac{1}{\varepsilon}\eta\eta'$. Introducing ε and ε' in equation (22) we have:

$$\left(\varphi_y'\varepsilon + \varphi_y\varepsilon'\right)^2 - \varphi_y^2\varepsilon'^2 + \varphi_y^2\eta'^2 = 1. \tag{23}$$

We then define the change of variable:

$$\theta = \varphi_y\varepsilon, \tag{24}$$

which allows us to transform equation (23) into one depending on θ and θ':

$$\theta'^2 + \xi\theta^2 = 1 \quad \text{with} \quad \xi = \frac{\eta'^2}{\varepsilon^4}. \tag{25}$$

Finally, two explicit ODEs can be derived from equation (25):

$$\theta' = \pm\sqrt{1 - \xi\theta^2}. \tag{26}$$

Given a solution to equation (25), we recover a solution of the original ODE (22) by simply inverting the change of variable of equation (24).

d) Critical points: We give two propositions regarding the critical points, which differ from $\text{SfT}^{1\to3\to2}$. The reason is that the expression of ξ is different in $\text{SfT}^{1\to2\to1}$.

Proposition 9 (Critical point definition in φ)**.** *Given a solution $\hat{\varphi}$ to equation* (19)*, u_c is a critical point if and only if $\hat{\varphi}^\top(u_c)\ \hat{\varphi}'(u_c) = 0$. An intuitive interpretation is that a critical point is the point on the curve where the tangent and the optical ray are orthogonal.*

Proposition 9 is proven in appendix D.

Similarly to definition 2 and proposition 4, we now introduce the notion of super curve and the set of super critical points. We also give three equivalent ways to characterize the super critical points.

Proposition 10 (Super critical point identities)**.** *A point u_s is a super critical point of equation* (19) *if and only if it is a solution of one of the following equations:*

$(\xi'(u_s) = 0) \Leftrightarrow \left(2\eta(u_s)\eta'^2(u_s) - \left(1 + \eta^2(u_s)\right)\eta''(u_s) = 0\right) \Leftrightarrow \left({\varphi_s}^\top(u_s)\ \varphi_s'(u_s) = 0\right)$,

with φ_s the super curve of equation (25)*.*

Proposition 10 is proven in appendix E.

IV. The Number of Solutions

We have shown that these are very under-constrained problems, described with a single ODE which has an infinite number of solutions. Adding anchor points from the true curve makes the corresponding IVP well-constrained, but it remains impractical. The main contribution of this paper is a method, the HMM-based solution, for finding good candidates of the true curve using only the ODE coefficients. Our method implies finding the so-called super critical points and solving efficiently an MIVP which limits the true curve to belong to a discrete number of candidate solutions. Our HMM method reconstructs each candidate solution and then the true curve, without knowing which reconstructed candidate solution is the true curve. In our experiments, we can note however that the reconstructed true curve does not fit perfectly with the true curve, but is very close to it. Discrepancies may be explained by noisy data, data fitting errors and computational issues. Intuitively our strategy forces the candidate solutions to be smooth as it forces the first derivatives of the depth variable to be small at the super critical points. This also pushes the candidate solutions away from the camera center as our ODE are sum-of-squares of the depth and its first derivatives. In addition, we empirically observed that candidate solutions that do not pass through many of the super critical points tend to be very non-smooth and tend to come very close to the camera center.

In Curve SfT, we face two fundamental cases. In the first case, the true curve does not have any critical point. It is then not recoverable. Mathematically, the ODE (10) has an infinite number of solutions and cannot be upgraded into an IVP. In the second case, the true curve has at least one critical point. It is then recoverable up to a finite number of ambiguities. Mathematically, the ODE (10) has an infinite number of solutions, but can be upgraded into the MIVP (15). In order to guarantee that the true curve belongs to the recovered set of curves, we thus have to enumerate all possible subsets of super critical points. These subsets give the *extended set of candidate solutions*, which we formally define as follows:

Definition 3 (Extended set of candidate solutions)**.** *Given N_s super critical points in $\mathcal{T}$, the extended set of candidate solutions is the set of all candidate solutions of the ODE* (10) *which pass through a combination of the N_s super critical points.*

From theorem (1), we compute the size of the extended set of candidate solutions as follows.

Proposition 11 (Size of the extended set of candidate solutions)**.** *Given N_s super critical points in $\mathcal{T}$, the size of the extended set of candidate solutions is $M_{ext} = 2\left(3^{N_s}\right) - 2$.*

Proof. For each subset of i super critical points selected from the set of super critical points, the number of combinations is given by the binomial coefficient $\binom{N_s}{i}$. From theorem (1), we know that, for each subset of i super critical points, we have 2^{i+1} candidate solutions of the MIVP (15). Then, the number of candidate solutions considering all possible subsets is $M_{ext} = \sum_{i=1}^{N_s} \binom{N_s}{i} 2^{i+1} = N_s 2^2 + ... + N_s 2^{N_s} + 2^{N_s+1}$. This can be simplified using the combinatorial binomial theorem which gives that for $n \geq 0$ and $(x, y) \in \mathbb{R}^2$, $(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}$. We then obtain $M_{ext} = 2\Big(\sum_{i=0}^{N_s} \binom{N_s}{i} 2^i - \binom{N_s}{0} 2^0\Big) = 2\Big((2+1)^{N_s} - 1\Big) = 2\left(3^{N_s}\right) - 2$. □

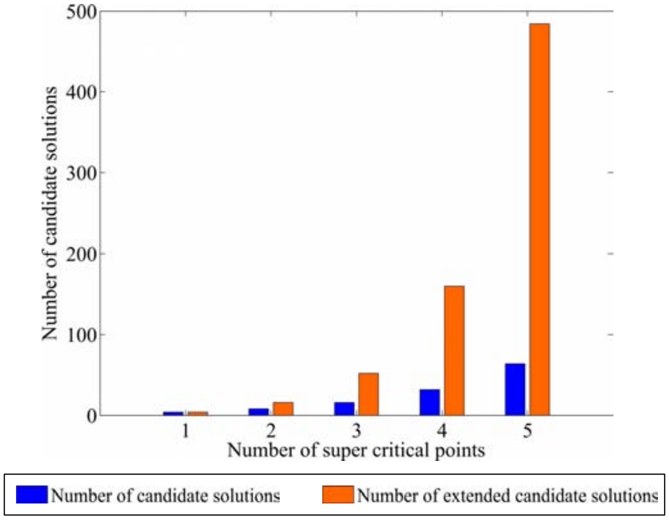
Considering all possible subsets of super critical points, we now give a condition of the recoverability of the true curve and the set of candidate solutions in which the true curve is contained.

Theorem 2 (Recoverability of the true curve)**.** *Given the ODE* (10) *and $N_s \in \mathbb{N}$ super critical points in $\mathcal{T}$,*

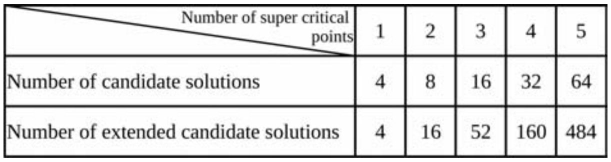
- *if the true curve does not pass through any critical point, it is not recoverable*
- *if the true curve passes through $1 \leq n_c \leq N_s$ critical points, the true curve is recoverable up to a finite number of ambiguities and is contained in the extended set of candidate solutions.*

Proof. We start with the case where the true curve does not pass through any critical point. As setting an *anchor point* as the initial condition of the IVP (11) requires additional information, we do not have any way to set an initial condition. Then we cannot recover the true curve from the ODE (10). We now consider the case where the ODE (10) has $N_s \geq 1$ super critical points and the true curve passes through $1 \leq n_c \leq N_s$ critical points. We can form the MIVP (15) with n_c initial conditions. The true curve is then recoverable and, from definition (3), the true curve is contained in the extended set of candidate solutions. □

From proposition 11, we have that the extended set of candidate solutions is much larger than the set of candidate solutions given by theorem 1. Figure 5 compares how the number of both sets grows as a function of the number of super critical points. We note that the extended set of candidate solutions grows faster with the number of super critical points, as compared to the set of candidate solutions. Therefore, considering the extended set of candidate solutions for the implementation may massively increase the computational burden. This is why we consider in §V the set of candidate solutions.



(a)



Number of super critical points	1	2	3	4	5
Number of candidate solutions	4	8	16	32	64
Number of extended candidate solutions	4	16	52	160	484

(b)

Fig. 5. Size of the extended and normal set of candidate solutions as function of the number of super critical points. (a) bar-plots of the set sizes as functions of the number of super critical points, (b) associated values of the set sizes.

V. Computational Solutions

As mentioned in §II-A3, three categories of method have been proposed to solve SfT. We propose here a method for the three categories and a four method from a new category which uses a graphical model. For each category of method, we first describe in detail the method for SfT$^{1\to3\to2}$ and then give the specialization to SfT$^{1\to2\to1}$.

In the theoretical analysis, we use a general function for the template's shape model. However, such model cannot be used for these four categories. We propose then to adapt the template's shape model for each category. We follow here the same notation of SfT inference categories given in §II-A3. We now give the specific shape model of the template for each category: a differentiable function for category *(i)*, a point set for category *(ii)* and an angle-based parameterization (presented in §V-C1) for category *(iii)*. For the new category *(iv)* based on graphical model, the shape is modeled by a chain.

A. Single-Solution Methods (Categories (i) and (ii))

The following methods are fast and simple solutions to Curve SfT, however they have limited practical use because they compute only a single candidate solution, which may be the wrong one. We describe them for completeness as they follow directly from existing category *(i)* and *(ii)* methods for Surface SfT.

1) A Category (i) Method: SfT$^{1\to3\to2}$ Case. We consider non-holonomic solutions and assume that the deformed curve $\mathcal{S}$ is infinitesimally linear. This is equivalent to $\mathcal{S}$ being a succession of infinitesimal lines, *i.e.* to consider $\mathbf{H}_\varphi(u) = \mathbf{0}_{3\times1}, \forall u \in \mathcal{T}$. By differentiating equation (6) and substituting, we obtain:

$$2\,\varphi_z'\mathbf{J}_{\bar{\eta}} + \varphi_z\mathbf{H}_{\bar{\eta}} = \mathbf{0}_{3\times1} \quad \Rightarrow \quad \varphi_z' = -\frac{1}{2}\varphi_z{\mathbf{J}_{\bar{\eta}}^{-1}}^\top\mathbf{H}_{\bar{\eta}}\,\varphi_z. \tag{27}$$

By substituting φ_z' from equation (8) in equation (27), we obtain two solutions for φ_z:

$$\varphi_z = \pm\frac{1}{\sqrt{\mathbf{J}_{\bar{\eta}}^\top\mathbf{J}_{\bar{\eta}} - \bar{\eta}^\top\mathbf{H}_{\bar{\eta}} + \frac{1}{4}||\bar{\eta}||^2\left({\mathbf{J}_{\bar{\eta}}^{-1}}^\top\mathbf{H}_{\bar{\eta}}\right)^2}}. \tag{28}$$

The infinitesimal linearity assumption can be generalized to higher orders assuming $\varphi^{(k)}(u) = \mathbf{0}_{3\times1}, \forall u \in \mathcal{T}$. This makes φ locally polynomial of finite order. However, unlike for infinitesimal linearity, for $k > 2$, analytical solutions are difficult, but not impossible, to find.

Specialization to SfT$^{1\to2\to1}$. Similarly to SfT$^{1\to3\to2}$, we obtain the analytical solution by differentiating equation (21) and assuming infinitesimal planarity, *i.e.* $\varphi''(u) = \mathbf{0}_{2\times1}$. This produces the solutions:

$$\varphi_y = \pm\frac{2\eta'}{\sqrt{\left(-\eta''\eta + 2\eta'^2\right)^2 + \left(-\eta''\right)^2}}. \tag{29}$$

For both cases, we discard the negative solution because it is behind the camera.

2) A Category (ii) Method: SfT$^{1\to3\to2}$ Case. We model the problem as an SOCP optimization. This is a direct adaptation of the so-called MDH [Perriollat et al., 2011; Salzmann and Fua, 2009], which we presented in §II-A3b. We discretize the template into a chain of N nodes and place the nodes at the correspondence points. We set the positions of each node in the input image at the 2D image correspondences, denoted by $\mathbf{q}_i$, $i \in [1, N]$. The unknowns are then the set of depths $d_i \in \mathbb{R}^+$ at each node. Any pair of nodes is constrained by the following *inextensibility constraint*:

$$\forall(i,j) \in [1,N]^2 \text{ with } i \neq j, \quad \left\| d_i(\mathbf{q}_i^\top\ 1)^\top - d_j(\mathbf{q}_j^\top\ 1)^\top \right\| \leq l_{ij}, \tag{30}$$

where $l_{ij} \in \mathbb{R}^+$ denotes the geodesic distance between nodes i and j, known from the template. The problem is then cast as an SOCP by searching for the maximal depth of each node such that equation (30) is satisfied.

Specialization to SfT$^{1\to2\to1}$. The extension of the MDH to SfT$^{1\to2\to1}$ is trivial and follows the formulation (30) for SfT$^{1\to2\to1}$ where, for each $i \in [1, N]$, q_i is now a 1D point.

B. A Multi-Solution Method with HMM (Category (iv))

1) Overview: We model the problem as a discrete HMM. This overcomes the limitations of category *(i)* and *(ii)* methods because *all* candidate solutions are reconstructed. Note that from now we restrict our search to the set of candidate solutions given in theorem 1 and *not* the extended set of candidate solutions, presented in §IV. This is motivated by two reasons: computing the extended set of candidate solutions increases massively the computational time of the category *(iv)* method and, in practice, we found most of the time that the set of candidate solutions contains a solution very close to the ground-truth.

Our multi-solution method is based on two main remarks:

- Super critical points separate the curve $\mathcal{S}$ (and then the 1D template) into contiguous pieces
- Proposition 7 tells us that each piece is recoverable up to a two-fold ambiguity.

Our method is illustrated by figure 6. From the estimated warp η, we detect the N_s super critical points using a method described in §V-B2. We use $u_j^s \in \mathbb{R}$ to denote the position of the j^{th} super critical point in the template. We use d_j^s to denote its corresponding depth. From proposition 6 we know that between two consecutive super critical points the sign of θ' is constant. We then reconstruct the full template by specifying a sign combination vector $\mathrm{s} = \{s_i\}_{i\in[1,N_s+1]} \in [-1,+1]^{N_s+1}$, where s_i denotes a selection for either the positive solution $\theta' = \sqrt{1-\xi\theta^2}$ or the negative solution $\theta' = -\sqrt{1-\xi\theta^2}$. Therefore, each sign combination vector characterizes uniquely each candidate solution, which gives to this category *(iv)* method the capability to reconstruct all possible candidate solutions. When N_s is small, it is feasible to generate all candidate solutions using every possible s. This is the case with our experimental data where N_s is typically lower than 12. When this is not the case, the template can be generated on demand with a specific s. A pseudo-code of this method is given in algorithm 1 in appendix F.

2) Super Critical Point Detection: All equivalent characterizations of a super critical point given by proposition 5 can be used for detection. They only require one to know the warp η. We have found the method using the roots of ξ' to be the most accurate and the most stable.

3) Graphical Modeling: We setup the HMM as follows. We generate its nodes by discretizing the template into $M+N_s$ 1D points: $\mathcal{U} = \{u_1, u_2, \ldots, u_{M+N_s}\}$. These are made by combining the N_s super critical point positions u_j^s with M uniformly sampled template points spanning the whole template. We order these such that $u_{i+1} \geq u_i$. The position of each node in the input image is denoted by $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{M+N_s}\} \in \mathbb{R}^{M+N_s}$. These are computed using the warp η.

The state of each node holds its unknown depth $d_i \in \mathbb{R}$. We draw d_i from a discrete set of D depth samples, denoted by $\mathcal{D} \in \mathbb{R}^+$. We discuss how $\mathcal{D}$ is created in §V-B5.

The graph's edges are constructed between consecutive nodes, producing $M + N_s - 1$ edges. We use the set $\mathcal{E}_i$ to denote the neighbors of node i. We define the graph's energy using first and second-order potentials. This has the following form:

$$E(\{d_i\}; \mathrm{s}, \mathcal{U}, \mathcal{Q}) = E_{iso}(\{d_i\}; \mathcal{U}, \mathcal{Q}) + \lambda_{cp} E_{cp}(\{d_i\}; \mathcal{Q}) + E_{sign}(\{d_i\}; \mathrm{s}, \mathcal{U}, \mathcal{Q}), \tag{31}$$

The energy term E_{iso} denotes the *isometric energy*, and is a second-order energy between consecutive nodes that enforces the isometry prior. The energy term E_{cp} denotes the *critical point energy*, and is a second-order energy between super critical point nodes and their neighboring nodes. This energy forces the tangent at each super critical point node to be orthogonal its line-of-sight (proposition 3). The last energy term E_{sign} is a first-order energy and forces the gradient signs of the super critical point nodes to agree with a particular sign combination vector s.

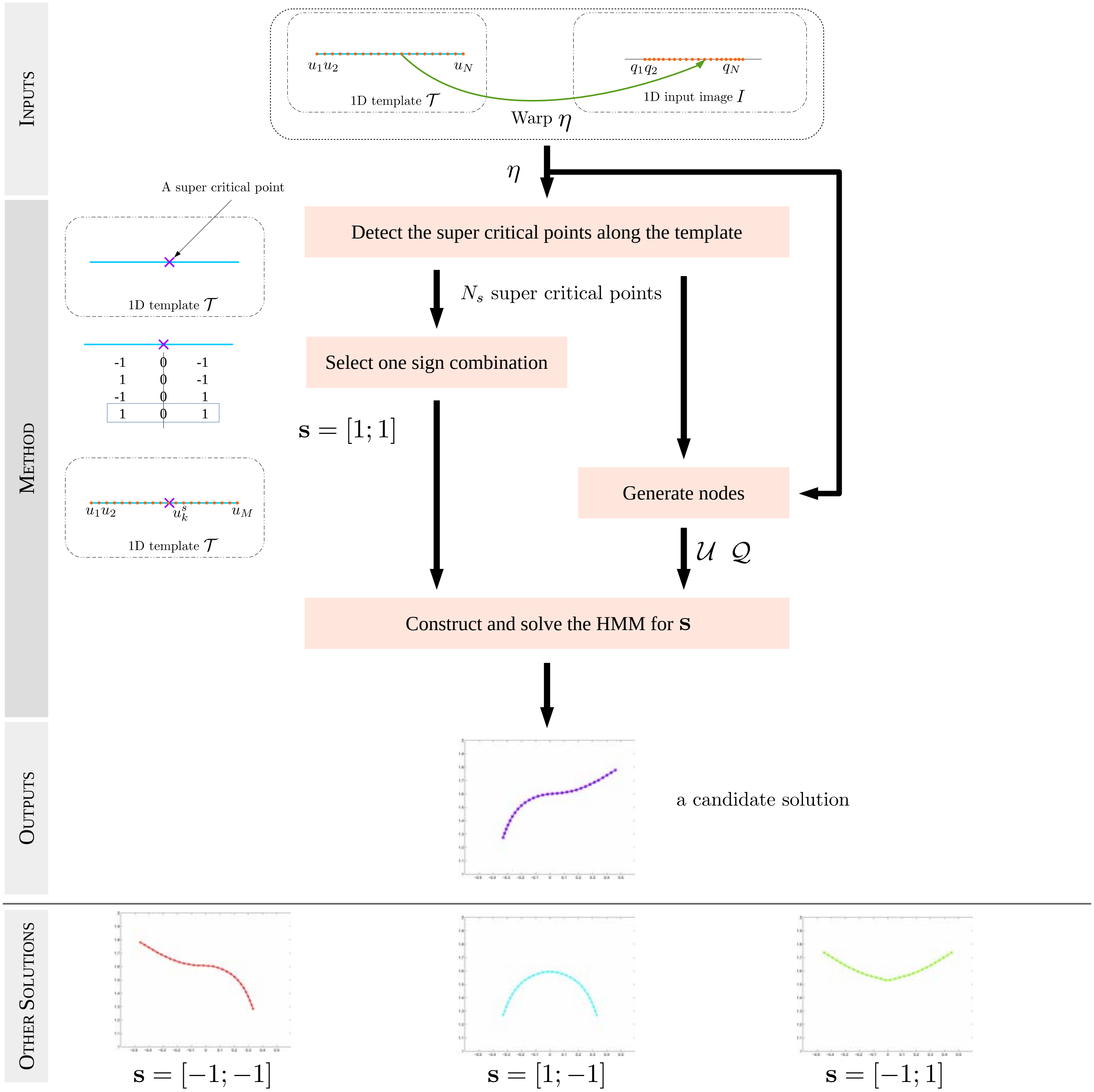


Fig. 6. Reconstruction pipeline of our proposed category *(iv)* method. **Top:** proposed category *(iv)* method for solving SfT$^{1\rightarrow 2\rightarrow 1}$ and so 4 candidate solutions. We illustrate the pipeline with an example where there is $N_s = 1$ super critical point. **Bottom:** we give the other candidate solutions, which are obtained by selecting different sign combinations $\mathbf{s}$.

4) Energy Definitions: The energy term E_{iso} is defined as follows:

$$E_{iso}(\{d_i\};\mathcal{U},\mathcal{Q}) = \sum_{i=1}^{M+N_s} \sum_{j\in\mathcal{E}_i} \left(\left\| d_i \left(\mathbf{q}_i^\top \ 1\right)^\top - d_j \left(\mathbf{q}_j^\top \ 1\right)^\top \right\| - l_{i.j} \right)^2, \tag{32}$$

where $l_{i,j} = |u_i - u_j|$ denotes the distance in the template domain between two neighboring nodes. This prevents the curve from stretching or compressing.

The energy term E_{cp} acts only at super critical point nodes. Suppose node i is a super critical point node. Its critical point energy is as follows:

$$E_{cp}^{i} = \sum_{j \in \mathcal{E}_i} \left| \left(\frac{d_i \left(\mathbf{q}_i^\top \; 1\right)^\top - d_j \left(\mathbf{q}_j^\top \; 1\right)^\top}{\left\| d_i \left(\mathbf{q}_i^\top \; 1\right)^\top - d_j \left(\mathbf{q}_j^\top \; 1\right)^\top \right\|} \right)^\top \frac{d_i \left(\mathbf{q}_i^\top \; 1\right)^\top}{\left\| d_i \left(\mathbf{q}_i^\top \; 1\right)^\top \right\|} \right|. \tag{33}$$

This term models the curve gradient using a finite difference and computes the dot product between the super critical point node gradient and the line-of-sight at the super critical point. We compute E_{cp} as the sum of equation (33) over all super critical point nodes.

The energy term E_{sign} also only acts at super critical point nodes and forces the reconstruction of the candidate solution uniquely characterized by the sign selection. Suppose node i is a super critical point node with sign selection s_i. The sign energy forces the sign of the gradient of θ_i' for node i to agree with the sign selection s_i. For this, it uses equation (9) and the sign selection s_i. For instance, we know that $\theta_i' < 0$, *i.e.* $s_i = -1$, implies that $d_i \left\| \left(\mathbf{q}_i^\top \; 1\right)^\top \right\| > d_{i+1} \left\| \left(\mathbf{q}_{i+1}^\top \; 1\right)^\top \right\|$. Then the sign energy is as follows:

$$E_{sign}^{i} = \sum_{j \in \mathcal{E}_i} \infty \left| \text{sign}\left(u_i - u_j\right) \text{sign}\left(\hat{d}_i \left\| \left(\mathbf{q}_i^\top \; 1\right) \right\| - \hat{d}_j \left\| \left(\mathbf{q}_j^\top \; 1\right) \right\| \right) - s_i \right|. \tag{34}$$

5) Depth Discretization: We define the lower and upper bounds on depth and uniformly sample the bounds using $D = 500$ intervals. The upper bound d_{max} is generated using the category *(ii)* method. We define $d_{max} = \max(\{d_i\})$, with $\{d_i\}$ the depth of all correspondences estimated by the category *(ii)* method. We set the lower bound to $d_{min} = f$, with f the camera focal length. For this, we assume that the curve is beyond the focal length, which is a reasonable assumption in practice.

6) Inference: Because our graph is a chain, global inference can be performed with the Viterbi algorithm [Rabiner, 1989]. Thus, we obtain a candidate solution whose θ' function follows the sign combination $\mathbf{s}$, as figure 6 illustrates.

7) Specialization to SfT$^{1\to 2\to 1}$: The adaption of the full reconstruction pipeline based on HMM is straightforward. As we formulate the reconstruction problem using only depths as unknowns, the HMM energies can be trivially extended to the reconstruction of 2D curves. All other components of the method are kept similar. In figure 6, we present reconstructions in the case of SfT$^{1\to 2\to 1}$ for a simpler understanding of the output from the category *(iv)* method. Figure 6 (bottom) also shows the other candidate solutions with respect to each possible sign combination.

C. Solution Refinement (Category (iii))

We propose a non-convex cost function that models Curve SfT, by balancing the reprojection error with a smoothing prior. Isometry is enforced implicitly using a novel angle-based parameterization. This tends to generate the most accurate results but requires suitable initialization. We provide this with the HMM solution. Figure 7 gives an overview of the method.

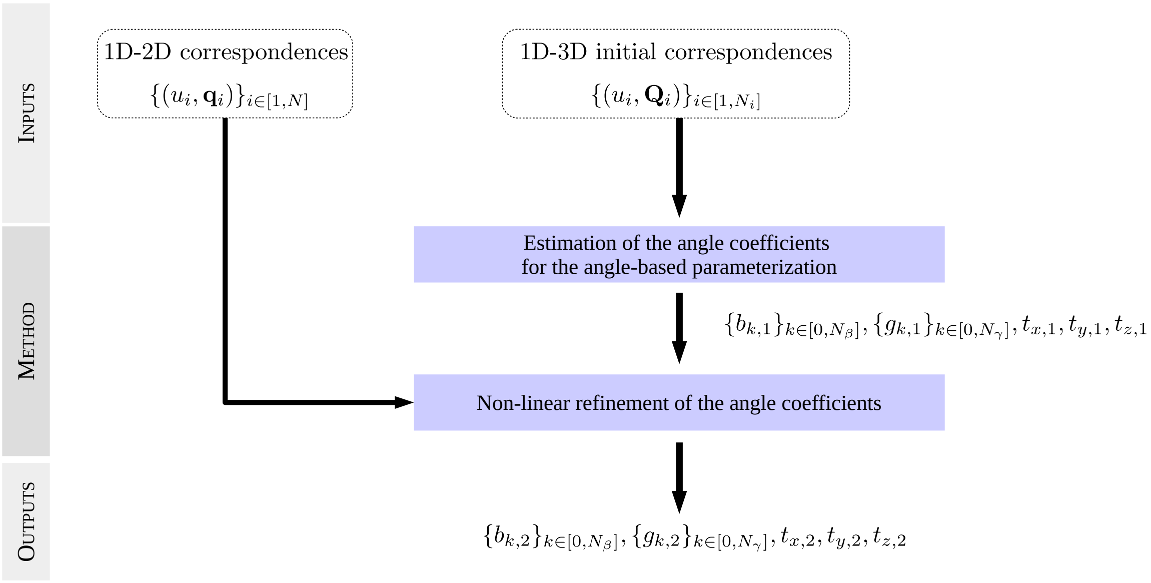

INPUTS
METHOD
OUTPUTS
1D-2D correspondences
$\{(u_i, \mathbf{q}_i)\}_{i\in[1,N]}$
1D-3D initial correspondences
$\{(u_i, \mathbf{Q}_i)\}_{i\in[1,N_i]}$
Estimation of the angle coefficients for the angle-based parameterization
$\{b_{k,1}\}_{k\in[0,N_\beta]}, \{g_{k,1}\}_{k\in[0,N_\gamma]}, t_{x,1}, t_{y,1}, t_{z,1}$
Non-linear refinement of the angle coefficients
$\{b_{k,2}\}_{k\in[0,N_\beta]}, \{g_{k,2}\}_{k\in[0,N_\gamma]}, t_{x,2}, t_{y,2}, t_{z,2}$

Fig. 7. Proposed refinement method for solving SfT$^{1\to3\to2}$.

1) Angle-Based Parameterization: The exact enforcement of isometry has two advantages. First, we do not need to balance it with reprojection and smoothing terms in the cost function, because isometry is always enforced. Second, it reduces the number of optimization variables. However, this parameterization presents one limitation: it cannot model quasi-isometry. We define the parameterization using a spherical parameterization with two angle functions $\beta : \mathcal{T} \to \mathbb{R}$ and $\gamma : \mathcal{T} \to \mathbb{R}$:

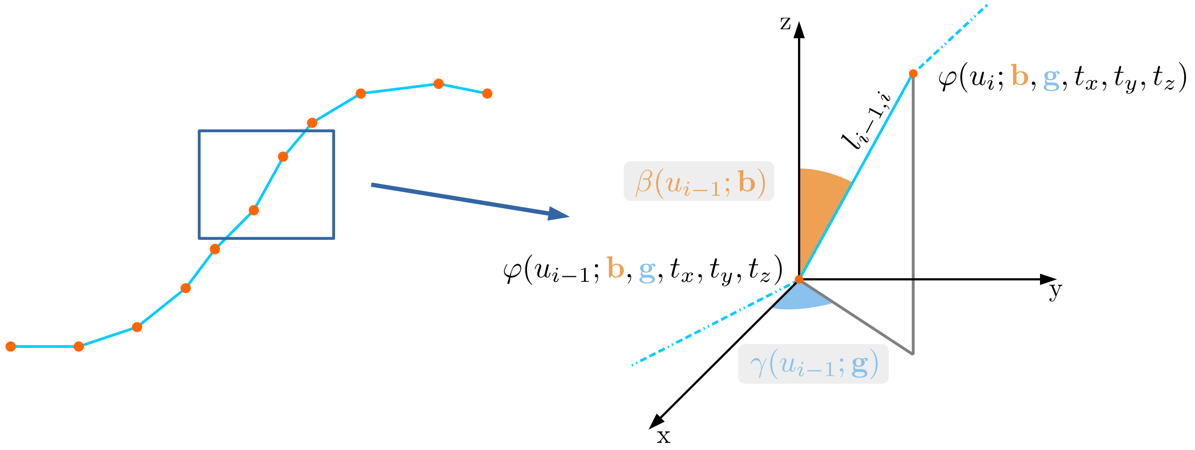
$$\forall u_i \in \mathcal{T}, \quad \varphi(u_i; \mathbf{b}, \mathbf{g}, t_x, t_y, t_z) = \sum_{j=2}^{i} l_{j-1,j} \begin{pmatrix} \sin\beta(u_{j-1}; \mathbf{b}) \ \cos\gamma(u_{j-1}; \mathbf{g}) \\ \sin\beta(u_{j-1}; \mathbf{b}) \ \sin\gamma(u_{j-1}; \mathbf{g}) \\ \cos\beta(u_{j-1}; \mathbf{b}) \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \tag{35}$$

with $l_{j-1,j}$ the length in the template between the $(j-1)^{th}$ and j^{th} correspondences. Figure 8 illustrates the angle-based parameterization. We construct the angle functions β and γ using respectively a degree N_β polynomial and a degree N_γ polynomial:

$$\forall u_j \in \mathcal{T}, \quad \beta(u_j; \mathbf{b}) = \sum_{k=0}^{N_\beta} u_j^k b_k \quad \text{and} \quad \gamma(u_j; \mathbf{g}) = \sum_{k=0}^{N_\gamma} u_j^k g_k, \tag{36}$$

with $\mathbf{b} = \{b_k\}_{k\in[0,N_\beta]}$ the set of polar angle coefficients and $\mathbf{g} = \{g_k\}_{k\in[0,N_\gamma]}$ the set of azimuthal angle coefficients. φ is defined by $N_\beta + N_\gamma + 5$ parameters: the polar angle coefficients $\mathbf{b} = \{b_k\}_{k\in[0,N_\beta]}$, the azimuthal angle coefficients $\mathbf{g} = \{g_k\}_{k\in[0,N_\gamma]}$ and the 3D translation parameters t_x, t_y and t_z.

We now explain how to compute $\mathbf{b}$, $\mathbf{g}$, t_x, t_y and t_z from the N_i initial 1D-3D correspondences which can be obtained from categories *(i)*, *(ii)* and *(iv)* methods. The value of N_i may differ with respect to the method used to initialize: for instance, for our methods from categories *(i)* and *(ii)* $N_i = N$, and for our method from category *(iv)* $N_i = M + N_s$ as explained in §V-B3. We first set the translation vector to the first 3D point of the curve. For the angle coefficients, we compute analytically the polar angle and the azimuthal angle between two 3D points using the trigonometric functions. Then, we fit the sought degree N_β and degree N_γ polynomials to the estimated polar and azimuthal angles respectively.



z
$\varphi(u_i; \mathbf{b}, \mathbf{g}, t_x, t_y, t_z)$
$l_{i-1,i}$
$\beta(u_{i-1}; \mathbf{b})$
$\varphi(u_{i-1}; \mathbf{b}, \mathbf{g}, t_x, t_y, t_z)$
y
$\gamma(u_{i-1}; \mathbf{g})$
x

Fig. 8. The angle-based parameterization for 3D curve. **Left**: a 3D curve parameterized by the angle-based parameterization (35). **Right**: a zoom on the angle-based parameterization. β refers to the polar angle and γ to the azimuthal angle.

2) Refinement: SfT$^{1\to3\to2}$ Case. The refinement is performed by non-linear least-squares optimization of a cost function containing a *reprojection* and a *smoothing* constraints:

$$C(\mathbf{b}, \mathbf{g}, t_x, t_y, t_z) = C_{reproj}(\mathbf{b}, \mathbf{g}, t_x, t_y, t_z) + \lambda_{smooth} C_{smooth}(\mathbf{b}, \mathbf{g}), \tag{37}$$

$$\text{with} \quad C_{reproj}(\mathbf{b}, \mathbf{g}, t_x, t_y, t_z) = \frac{1}{N} \sum_{i=1}^{N} \left\| \Pi \circ \varphi(u_i; \mathbf{b}, \mathbf{g}, t_x, t_y, t_z) - \mathbf{q}_i \right\|^2,$$

$$\text{and} \quad C_{smooth}(\mathbf{b}, \mathbf{g}) = \frac{1}{N-1} \left(\sum_{i=2}^{N} \Big(\beta(u_i; \mathbf{b}) - \beta(u_{i-1}; \mathbf{b}) \Big)^2 + \Big(\gamma(u_i; \mathbf{g}) - \gamma(u_{i-1}; \mathbf{g}) \Big)^2 \right),$$

where $\lambda_{smooth} \geq 0$ is the smoothing weight. This is solved using Levenberg-Marquardt.

Specialization to SfT$^{1\to2\to1}$. The specialization simplifies the angle-based parameterization and the cost function. For the angle-based parameterization, we define φ using only one angle function $\alpha : \mathcal{T} \to \mathbb{R}$:

$$\forall u_i \in \mathcal{T}, \quad \varphi(u_i; \mathbf{a}, t_x, t_y) = \sum_{j=2}^{i} l_{j-1,j} \begin{pmatrix} \cos \alpha(u_{j-1}; \mathbf{a}) \\ \sin \alpha(u_{j-1}; \mathbf{a}) \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{38}$$

where $l_{j-1,j}$ is the length in the template between the $(j-1)^{th}$ and j^{th} correspondences, $\mathbf{a} = \{a_k\}_{k \in [0, N_\alpha]}$ the set of angle coefficients and $\alpha(u_{j-1}; \mathbf{a}) = \sum_{k=0}^{N_\alpha} u_{j-1}^k a_k$, a degree N_α polynomial. φ is then defined by $N_\alpha + 3$ parameters: the angle coefficients $\mathbf{a} = \{a_k\}_{k \in [0, N_\alpha]}$ and the 2D translation parameters t_x and t_y. The cost function includes a smoothing which penalizes non-smooth variations:

$$C(\mathbf{a}, t_x, t_y) = C_{reproj}(\mathbf{a}, t_x, t_y) + \lambda_{smooth} C_{smooth}(\mathbf{a}), \tag{39}$$

$$\text{with} \quad C_{reproj}(\mathbf{a}, t_x, t_y) = \frac{1}{N} \sum_{i=1}^{N} \left\| \Pi \circ \varphi(u_i; \mathbf{a}, t_x, t_y) - q_i \right\|^2,$$

$$\text{and} \quad C_{smooth}(\mathbf{a}) = \frac{1}{N-1} \sum_{i=2}^{N} \Big(\alpha(u_i; \mathbf{a}) - \alpha(u_{i-1}; \mathbf{a}) \Big)^2.$$

VI. Experimental Validation

A. *SfT$^{1\rightarrow 2\rightarrow 1}$ Experiments*

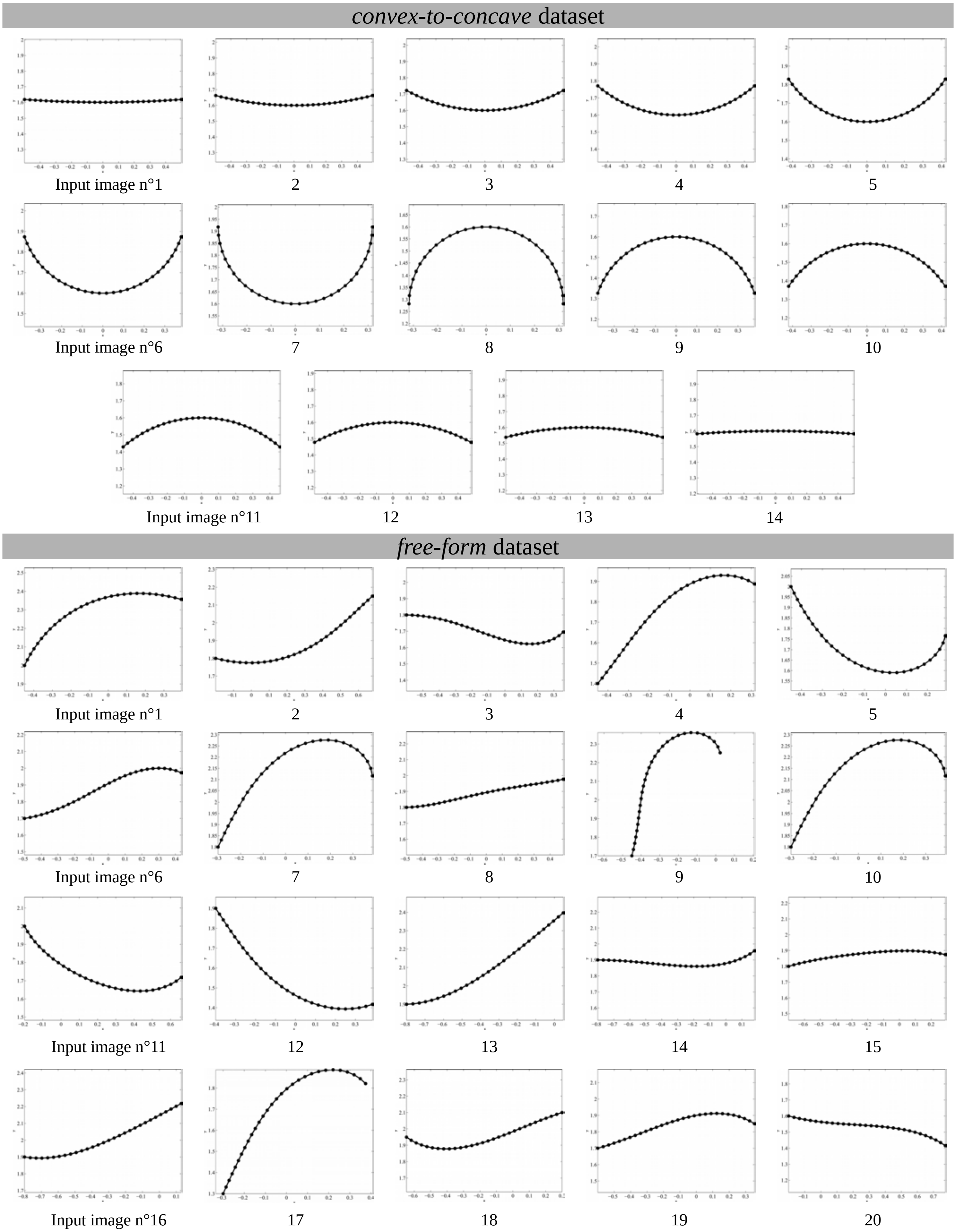
1) Methods: We evaluate thoroughly the category *(iv)* method with and without refinement. As categories *(i)* and *(ii)* methods are single-solution methods, we give only some results for the category *(ii)* method, showing that we can rule it out. The category *(iv)* method is denoted **2DHMM** and the category *(iv)* method followed by iterative refinement is denoted **2DHMM+REF**. The category *(ii)* method, without and with refinement, is denoted respectively **2DMDH** and **2DMDH+REF**.

All methods are implemented in an unoptimized Matlab code and ran on a standard Intel i7 desktop workstation. We use YALMIP [Löfberg, 2004] and SeDuMi [Sturm, 1999] to implement the category *(ii)* method. We use [Schmidt, 2007] to construct and solve the HMMs (category *(iv)*). We perform the non-linear refinement (category *(iii)*) using the Matlab function `lsqnonlin`. We use the curve fitting toolbox of Matlab to detect the critical points as explained in §V-B2.

We compute the template-to-image warp η using an interpolation function and a set of N 1D correspondence points between the template and the input image. We construct η using a spline which is obtained by the Matlab function `spaps` and the curve fitting toolbox. To fix N_α, we fitted the ground-truth 2D curves of the all datasets used in SfT$^{1\rightarrow 2\rightarrow 1}$ using the angle-based parameterization with different polynomial orders for the angle α. We used 9 different values of N_α in the range of $[4, 20]$ and selected the polynomial order which was large enough to model accurately most 2D curves, is not too high to avoid over-fitting and increasing the computational burden in the refinement step. Table V in appendix G gives the hyperparameters for each method for the sake of reproducibility. We remind that a pseudo-code of the HMM-based method is given in algorithm 1 in appendix F.

2) Datasets:

a) Simulated datasets: We evaluated performance with two simulated datasets: the *convex-to-concave* dataset and the *free-form* dataset. The *convex-to-concave* dataset consists of 14 input images that were generated by fixing the middle point of the template at a same depth and in front of the camera center and by decreasing the curvature of the embedded curve in 14 increments, going from convex to concave. Input images n°1 to 7 show convex examples and 8 to 14 show concave examples. The *free-form* dataset consists of 20 input images generated by isometrically deforming the template using a degree 5 polynomial. For the *convex-to-concave* and the *free-form* datasets, the apparent size of the curve in the input images is respectively on average 55% and 47%. For both datasets, we used a 1D template with unit length and added gaussian noise to the correspondences with a standard deviation equal to 1.0 px a for 1D input image of 1920 px. We show some curves of both datasets in figure 11. Critical points are computed by finding where the first derivative of the ground-truth function θ is equal to zero (definition 1). We show in figure 9 the simulated 2D curves for both datasets.



convex-to-concave dataset

Input image n°1 2 3 4 5

Input image n°6 7 8 9 10

Input image n°11 12 13 14

free-form dataset

Input image n°1 2 3 4 5

Input image n°6 7 8 9 10

Input image n°11 12 13 14 15

Input image n°16 17 18 19 20

Fig. 9. Visualization of the ground-truth 2D curves for the two simulated datasets.

b) Real datasets: We tested our methods on two real datasets: the *paper* and the *cable* datasets. The idea behind these real datasets is to test our methods on 2D curves which have a physical meaning. For these two real datasets, the 2D curves which we want to reconstruct are images along a plane of 3D deformations of a paper and a cable. This is illustrated by the figure 17, where we see that the reconstructed 2D curves are images of the 3D deformations visible in the 2D images.

The *paper* dataset was built from a 3D reconstruction of a bent paper (figure 17) generated by Agisoft Photoscan [Agisoft, 2014]. From this reconstruction, 1D input images could be drawn by using black sampling lines on the paper. We used the central line illustrated in yellow. To generate the 1D input image, the camera's y-axis was aligned to the central line. The middle row of the image was thus used as the 1D input image. The *cable* dataset was built using two cameras looking at a cable over a table. One camera took images by aligning its y-axis with the table. The second camera had an over-head view. Its relative pose to the first camera was computed through stereo calibration, and was used to compute the cable's ground-truth shape.

Template construction. In practice, we model the shape of the curvilinear template as a discrete representation of the object as a straight line. The discrete points represent features of the curvilinear object. The features of the template are matched with the input image and the obtained correspondences represent the location of the same physical point in the template and the input image. For the *paper* dataset, the features are the center of the black sampling lines and, for the *cable* dataset, the features are the boundaries of each black markers. Therefore, to construct the template for the *paper* dataset, we measure the distance between the center of each black sampling line and, for the *cable* dataset, we measure the distance between the boundaries of each black marker.

Input images. We used 5 deformations for the *paper* dataset and 10 for the *cable* dataset. For both datasets, the images had a width of 4800 px. We selected manually 30 correspondences for the *paper* dataset and 40 for the *cable* dataset. Note that, in our instantiation of Curve SfT, we assume that there is no mismatches in the correspondences and no self or external occlusions. However, in the other instantiations where the number of correspondences can differ with the input images due to self or external occlusions or where mismatches exist, one can think computing robustly the *template-to-image* warp η using some feature-based matching methods and discretize the template and the input image. For such practical challenges, some methods were proposed to handle large number of mismatches [Lowe, 2004] and discontinuities occurring from self-occlusions [Pizarro and Bartoli, 2012].

3) Evaluation Metrics: We measure accuracy through four metrics: 2D mean point error, normal error, super critical point precision and super critical point accuracy. We emphasize that, as Curve SfT cannot be solved uniquely, we only evaluate the best candidate solution (with lowest 2D mean point error). We measure the super critical point precision and accuracy only for the simulated datasets. This is because the computation of the ground-truth super critical points is more reliable on simulated datasets, where the first derivative of the ground-truth function θ is less noisy.

a) 2D mean point error (MPE): We construct an evaluation grid by uniformly sampling the template in $G = 30$ points and denote it by $\mathcal{G} = \{u_j\}$. We compute the 2D mean point error (in %) between the reconstructed curve $\hat{\varphi}$ and the ground-truth shape φ^* on $\mathcal{G}$ as:

$$MPE(\hat{\varphi}, \varphi^*, \mathcal{G}) = \frac{1}{G} \sum_{j=1}^{G} \frac{\|\hat{\varphi}(u_j) - \varphi^*(u_j)\|}{\|\varphi^*(u_j)\|}. \tag{40}$$

b) 2D normal error (NE): We denote the 2D normal of $\hat{\varphi}$ at a template point u by $\hat{n}(u)$ and the 2D normal of φ^* by $n^*(u)$. In practice, we fit a spline to compute the normals and select the normals with negative y-component. We compute then the 2D normal error (in degrees) between the reconstructed curve $\hat{\varphi}$ and the ground-truth shape φ^* at $\mathcal{G}$ by:

$$NE(\hat{\varphi}, \varphi^*, \mathcal{G}) = \frac{1}{G} \sum_{j=1}^{G} \cos^{-1} \left(\hat{n}^\top(u_j)\, n^*(u_j) \right). \tag{41}$$

We now give details on the normal computation. We first fit a spline over the correspondence points between the template and the input image. For this, we use the Matlab function `spaps` with smoothing parameter of $1\mathrm{e}{-5}$ for the *convex-to-concave*, *free-form* and *paper* datasets, and $1\mathrm{e}{-7}$ for the *cable* dataset. Second, we use the curve fitting toolbox of Matlab to differentiate φ and to obtain their 2D tangents.

c) Super critical point precision: We compute the super critical point precision as the fraction of the number of true super critical points over the number of detected super critical points. We define a true super critical point as a super critical point which is close to a ground-truth super critical point up to T % of the template length. We set $T = 5\%$.

d) Super critical point accuracy (SCPA): We denote by $\{u_j^{s*}\} \in \mathbb{R}^{N_s^*}$ the set of ground-truth super critical points with $u_j^{s*} \in \mathcal{T}$. We denote the closest detected super critical point to each ground-truth super critical point by $\{\hat{u}_j^s\}$. The super critical point accuracy is given by:

$$SCPA(\{\hat{u}_j^s\}, \{u_j^{s*}\}) = \frac{1}{N_s^*} \sum_{j=1}^{N_s^*} \frac{|u_j^s - u_j^{s*}|}{L}, \tag{42}$$

where L is the template length.

4) Results on Simulated Datasets: We emphasize that all reconstructions obtained with **2DHMM** and **2DHMM+REF** and shown in §VI-A4 and §VI-A5 were selected by searching the candidate solution with the lowest 2D mean point error given in §VI-A3. The reason is that the problem is intrinsically ambiguous and there is then no algorithm which finds the true curve among the candidate solutions given by the category *(iv)* method without any additional information different from the input data.

a) Reconstruction accuracy: Figure 10 shows the reconstruction accuracy on the two simulated datasets. Both **2DHMM** and **2DHMM+REF** provide convincing reconstructions, in terms of depth and normals, which we can observe visually in figure 11. **2DHMM+REF** is clearly more accurate. This is explained by two reasons. The main one is that the accuracy of **2DHMM** depends on the accuracy of the super critical point locations. If they are badly localized, **2DHMM** cannot be expected to provide a very accurate solution. By contrast, **2DHMM+REF** is free to optimize the curve without being limited to the accuracy of the super critical point locations. Secondly, there is a smoothing term in **2DHMM+REF** that is not present in **2DHMM**, which penalizes non-smooth curve solutions. The benefit is indicated in the empirical results, which show that the normal error is generally strongly reduced with **2DHMM+REF**. For the *free-form* dataset, we can make two remarks. First, we observe three large 2D mean point errors (input images n°4, 9 and 10). For these configurations, the normal errors are quite similar to the other configurations. One reason is the coupled effect of the noise in the correspondences and the complexity of the scene, due to *e.g.* strong perspective and/or high change of curvature, which figure 9 illustrates. Second, for the input images n°1, 4, 9, 10 and 17 of the *free-form* dataset, we observe that the 2D mean point errors for **2DHMM+REF** is higher than the one for **2DHMM**. At first sight, inaccurate localization of the super critical points may be the source of such behavior because it leads **2DHMM+REF** to a local minimum. However, this contradicts the super critical point accuracy for the input images n°4, 9, 10 and 17 shown in figure 14. Their super critical point accuracy is lower than the median value of all super critical point accuracies, which is 1.77%. This tells us that the closest detected super critical point is significantly close to the ground-truth super critical point. One strong possible reason is the curve modeling used in our category *(iii)* method. In general, our modeling improves the reconstruction, however some errors may occur, which are acceptable since they are not too important and not systematic. We also note that, even if the curve location is not accurate in all input images, the global curvature is well recovered in general. This can be explained by the hard constraint on the tangent at super critical points imposed through E_{sign}. In figure 12, we illustrate the inherent limitation of the category *(ii)* method, **2DMDH** and **2DMDH+REF**, which can only generate a single solution. We see that the initial solution from **2DMDH** is wrong, and that the refinement from **2DMDH+REF** is trapped in an incorrect minimum.

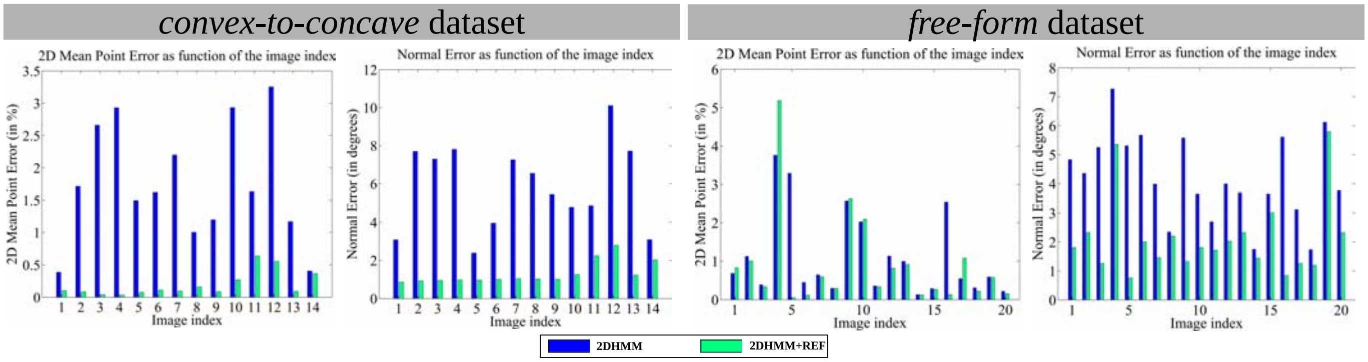


Fig. 10. Reconstruction accuracy of **2DHMM** and **2DHMM+REF**, for the two simulated datasets.

We then tested the influence of *(a)* correspondence density and *(b)* correspondence noise on reconstruction accuracy. For *(a)*, we added gaussian noise with a standard deviation of 1.0 px to between 10 and 100 random correspondences. To do this, for each input image of the two simulated datasets, we generated the correspondences by uniformly sampling N points along the template with $10 \leq N \leq 100$. For *(b)*, we set the number of correspondences at 30 and run our algorithm with 9 different noise levels with a standard deviation between 0 px and 4.0 px. Figure 13 shows the results for both experiments. For both datasets, we note that **2DHMM** is not sensitive to the number of correspondences. This can be explained by the use of a fixed number of nodes in the HMM construction given in §V-B3. The reconstructions with **2DHMM+REF** are slightly better and this is because the refinement uses the real 1D correspondences, as discussed in §V-C. The variation of the **2DHMM** reconstruction accuracy for the *free-form* dataset may be explained by the shape of the 2D curves which are more complex than the ones of the *convex-to-concave* dataset. However, for both datasets, we note that the refined solution **2DHMM+REF** is improved with higher numbers of correspondences, which makes sense as we have more data constraints. Regarding the noise level, we also note the high robustness of **2DHMM** for the *convex-to-concave* dataset, which is due to the fact that its dataset has very simple curves. The *free-form* dataset is more informative: we observe that the increase in noise level degrades the performance of **2DHMM** and **2DHMM+REF**. Globally, we note that in all experiments the refinement improves the reconstruction significantly.

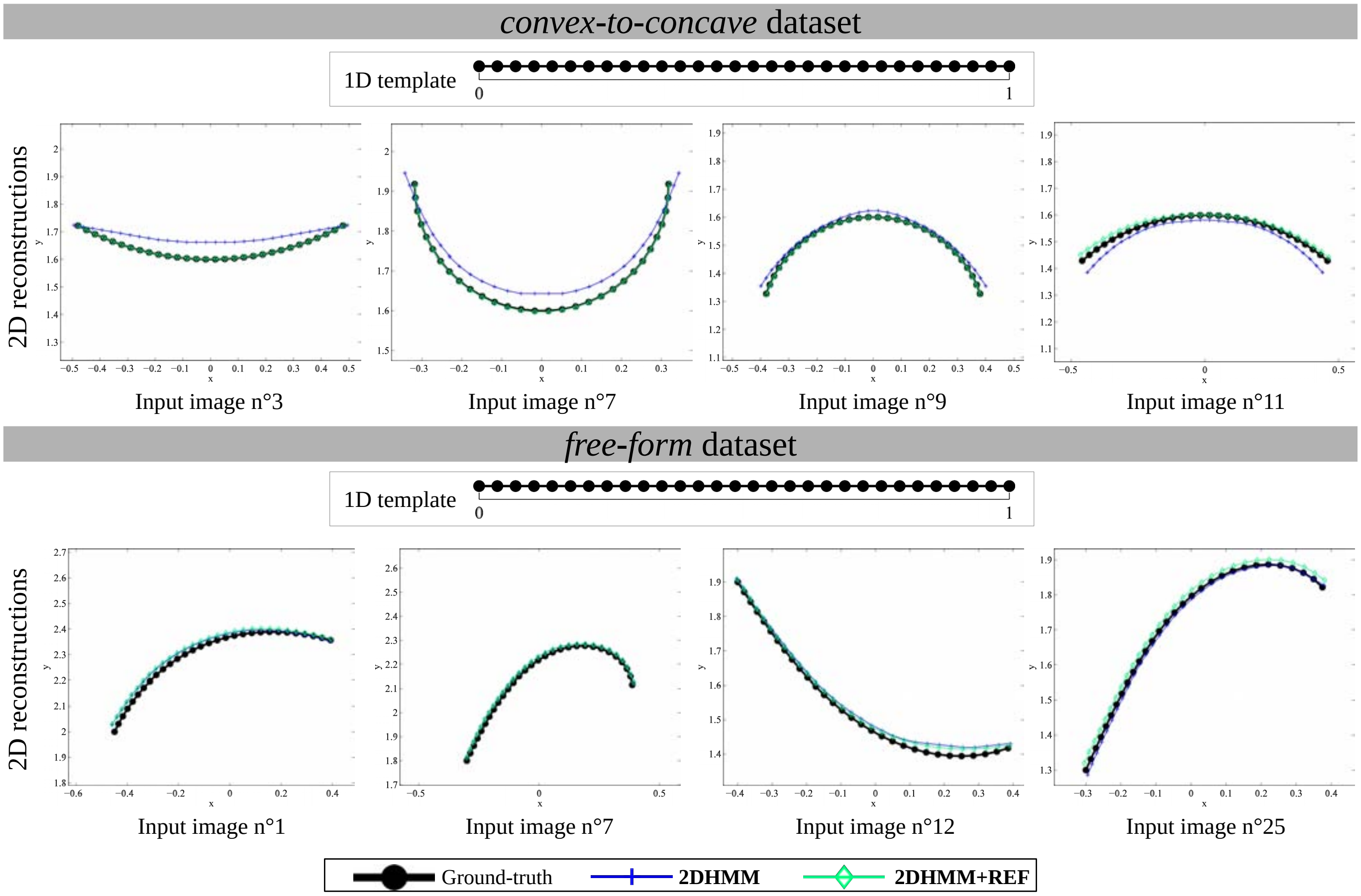


Fig. 11. Visual results of **2DHMM** and **2DHMM+REF**, for the two simulated datasets. We show the 2D reconstructed curves and their ground-truth solutions. As the 2D reconstructed curves are very close to the ground-truth solution, see the digital version of the document for better visualization.

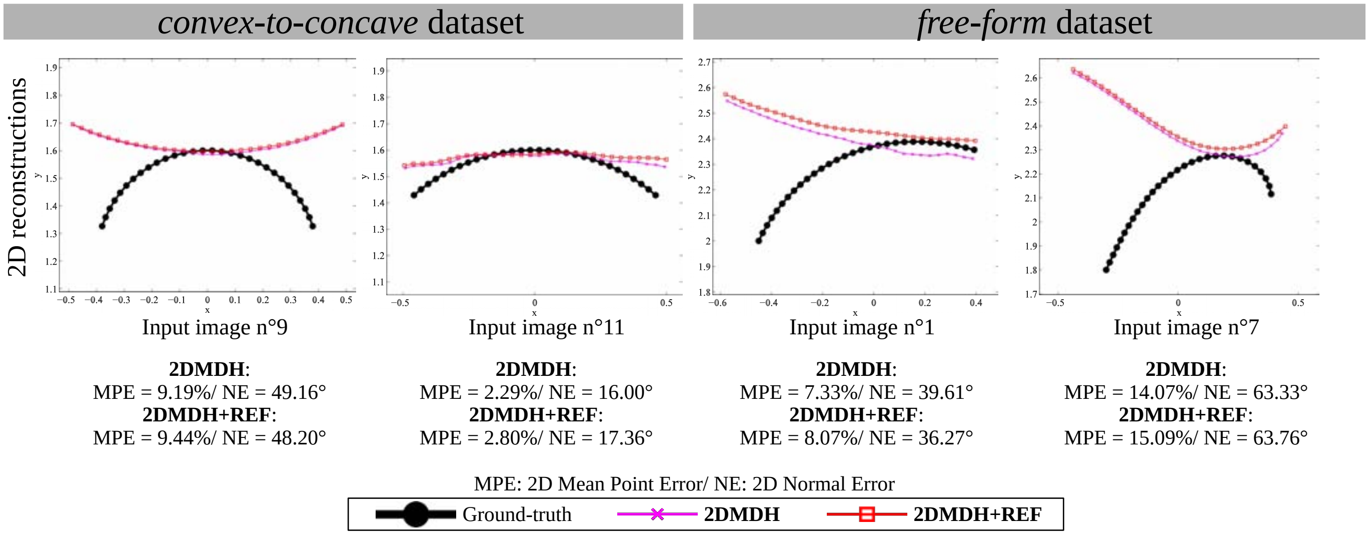


Fig. 12. Visual results and reconstruction accuracy of **2DMDH** and **2DMDH+REF** on a subset of input images used in figure 11. We verify that the category *(ii)* method does not always give the correct solution because it is a single-solution method.

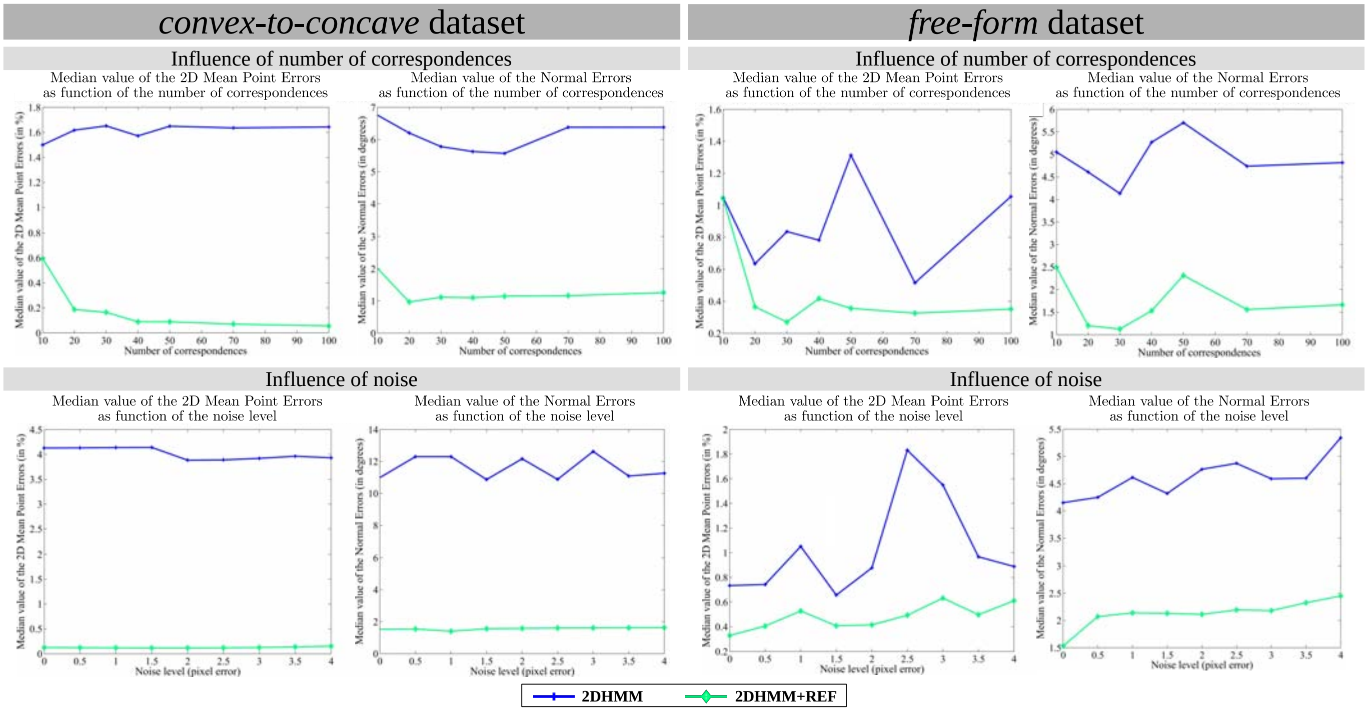


Fig. 13. Experimental analysis of **2DHMM** and **2DHMM+REF**, using the two simulated datasets. We ran both methods with varying numbers of correspondences (with fixed noise level) and for several noise levels (with fixed number of correspondences). We recall that, to compute these errors, we only use the best solution among the multiple ones given by **2DHMM**.

b) Super critical point detection: Figure 14 shows the super critical point precision for the simulated datasets. Our detection method has perfect precision for the *convex-to-concave* dataset and slightly over-detects the super critical points for the *free-form* dataset. Regarding the super critical point accuracy, we can observe a significant difference of the accuracy magnitude between the two simulated datasets: the worst accuracy reaches 0.19% for the *convex-to-concave* dataset and 37.42% for the *free-form* dataset. This can be explained by the very good results of the *convex-to-concave* dataset which are due to the fact that this dataset is smooth, so the interpolation of the warp with the correspondences is very accurate. The order of magnitude of the super critical point accuracy which we met with the other datasets is close to the obtained one with the *free-form* dataset.

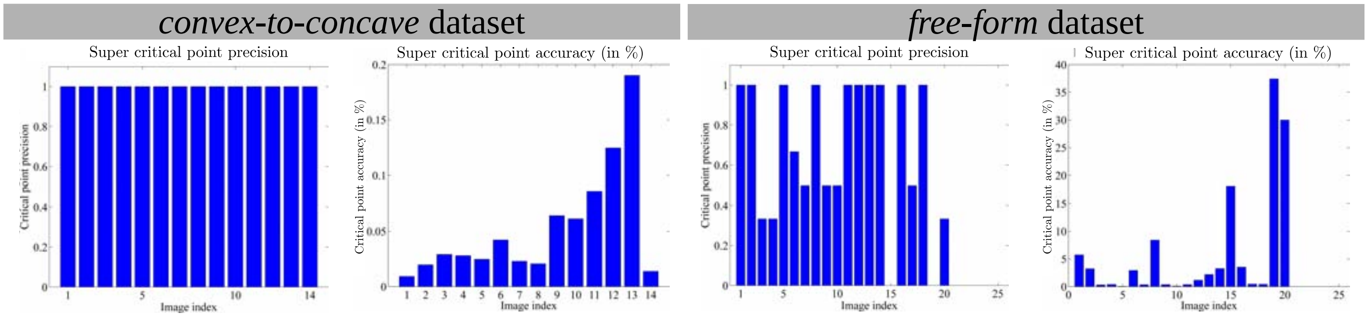


Fig. 14. Super critical point precision and accuracy. We use the two simulated datasets.

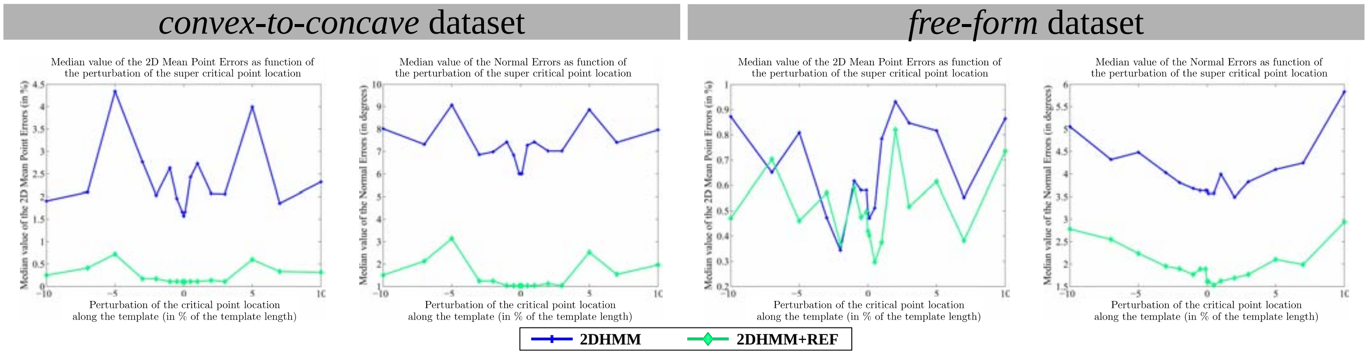


Fig. 15. Reconstruction accuracy as function of the perturbation on the super critical point location along the template. We use the two simulated datasets.

c) Influence of super critical point uncertainty on reconstruction accuracy: Our final test consisted in evaluating how sensitive the reconstructions are to incorrectly-located super critical points. Figure 15 shows the reconstruction accuracy for different perturbation levels of the ground-truth super critical point locations. A perturbation level of $X\%$ means that all super critical points were translated by $X\%$ of the template length along the template. We use the median value of the 2D mean point error and of the 2D normal error to measure the reconstruction accuracy, as we show in the second row of figure 15. We note first that the global minimum is very close to the perturbation level of 0%. We can explain this discrepancy by numerical issues since the ground-truth super critical points are not directly observed, but detected by a process which involves fitting and differentiation of ground-truth data. We then observe that the local minima are strongly symmetric for the *convex-to-concave* dataset. This is because the curves are symmetric around the super critical point, which is located near the midpoint of the curve, *i.e.* the point around which we change the shape of the curve. Super critical point uncertainty does not have a significant impact on the reconstruction accuracy: they degrade the reconstructions by less then 3% 2D mean point error and by less than 3.5 degrees normal error for the *convex-to-concave* dataset, and by less than 1% 2D mean point error and by less than 1.5 degrees normal error for the *free-form* dataset. **2DHMM** and **2DHMM+REF** are more robust on the *free-form* dataset, even if it is a more challenging dataset than the *convex-to-concave* dataset. There is a relationship between the curve's shape and the sensitivity to super critical point localization and this is demonstrated by the difference in performance between the two datasets. It is out of scope to analyze this relationship, but it may be possible with perturbation theory. Our method **2DHMM** and **2DHMM+REF** are able to generate reasonable candidate solutions despite a relatively large error in super critical point locations (until $\pm 10\%$ of the template length).

d) Influence of the number of critical points on reconstruction accuracy: In parallel to the super critical points, we observed that a number of critical points higher than 1 likely means that the curve is complex, *i.e.* strong perspective and/or high curvature. We also observed that the presence of a single critical point likely implies that the curve is simple, *i.e.* constant curvature. From the experiments, we noted that complex curves are more difficult to reconstruct. This is visible while comparing the reconstruction accuracy of the *convex-to-concave* and *free form* datasets, given in figure 9.

A first conclusion would be that a high number of critical points makes the reconstruction more prone to errors. This is not a trivial conclusion because more critical points also implies more "anchor" points and thus better reconstruction. In other words, the more critical points a curve has, the more constrained our HMM-based reconstruction algorithm is, but also the more complex the curve is, the less accurate its reconstruction is. Figure 16 shows the reconstruction accuracy as a function of the number of critical points for some of our datasets. We see that in general the lower reconstruction accuracy is obtained when the true curves have a single critical point, which confirms our first conclusion. We note that the 2D mean point error obtained for the *cable* dataset is lower for the input image with the highest number of critical points, however it is reasonable to not take this result into account because it corresponds to a single input image and the associated normal error is the highest.

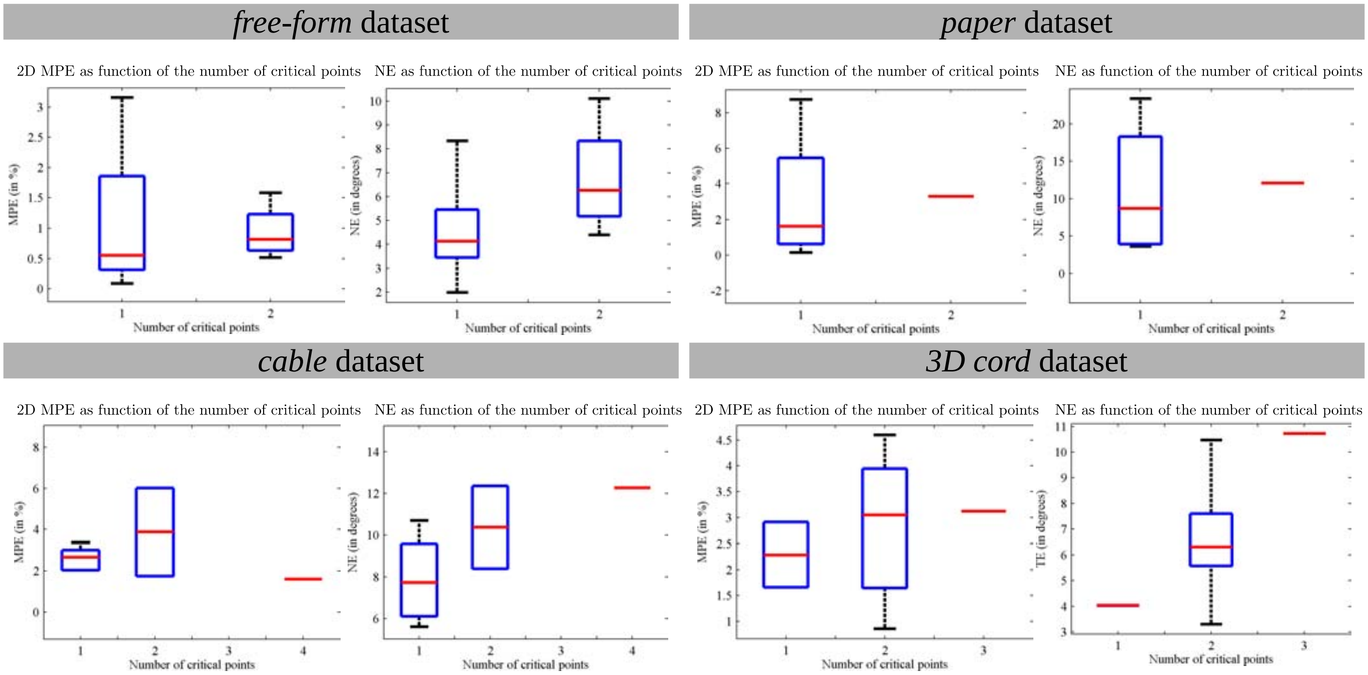


Fig. 16. Reconstruction accuracy as a function of the number of critical points for the *free-form*, the *paper*, the *cable* and the *3D cord* datasets. We do not use the *convex-to-concave* dataset because, for every input image, there is a single critical point. We do not use the *road* dataset because the true curve does not pass through any critical point. We do not use the *necklace* dataset because there is only one input image.

5) Results on Real Datasets: In figure 17, we observe that our methods produce convincing reconstructions. This is coherent with the reconstruction accuracy presented in figure 18. We see that the refinement method **2DHMM+REF** produces the most accurate results: the refinement globally improves the normals orientation. There is no significant difference in 2D mean point error for the *paper* dataset between **2DHMM** and **2DHMM+REF**. However, the small difference can be explained in the same way as we explained the differences of 2D mean point error for the *free-form* dataset in §VI-A4. In figure 19, we show the reconstruction results with the category *(ii)* method with and without refinement, **2DMDH** and **2DMDH+REF**. Note again that it is not capable to provide the correct solution in all cases.

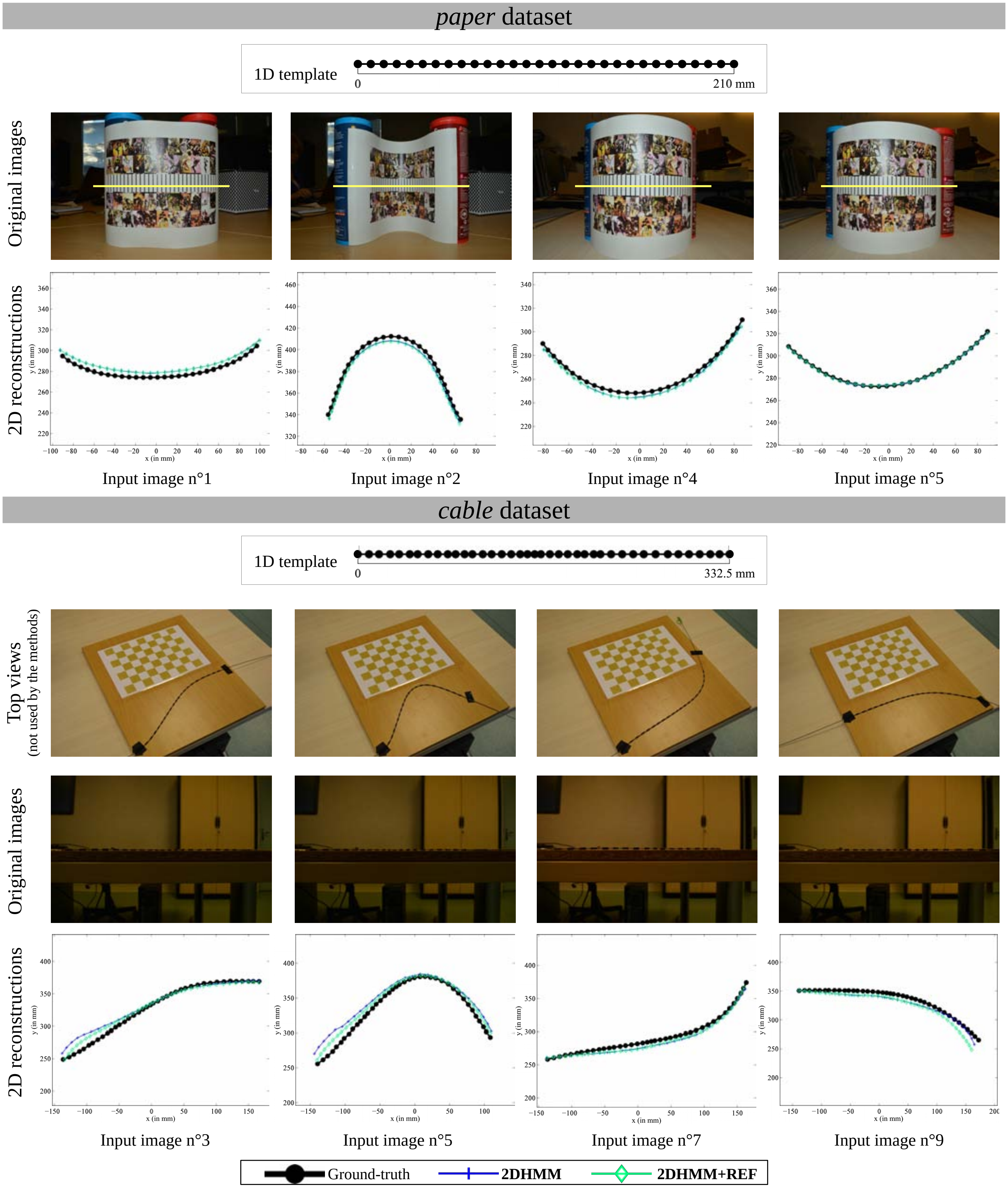


Fig. 17. Visual results of **2DHMM** and **2DHMM+REF**, for the two real datasets. We show 2D reconstructed curves and their ground-truth solutions. For each dataset, the original images correspond to the images from which we get the 1D input images. For the *cable* dataset, we show a top view of the deformed cable for each input image. As the 2D reconstructed curves are very close to the ground-truth solution, we refer readers to the digital version of the document for better visualization.

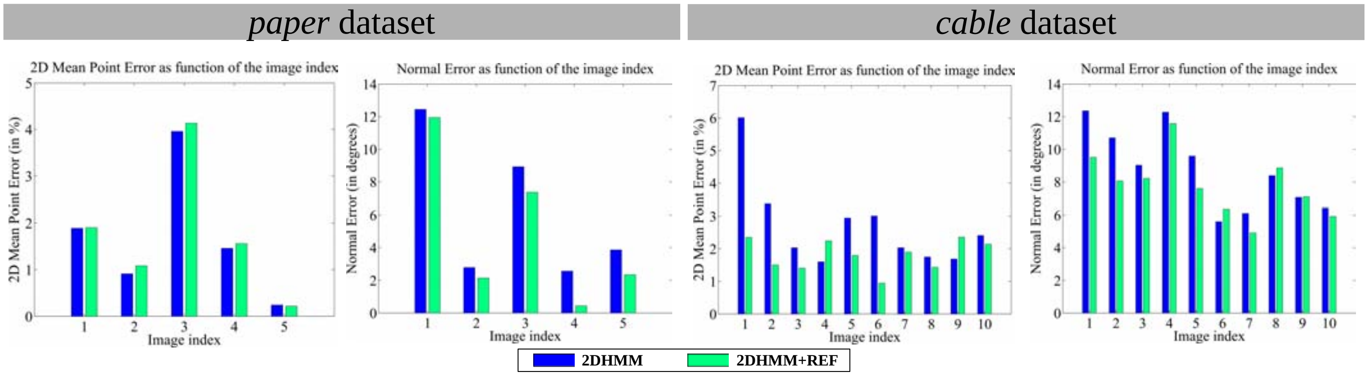


Fig. 18. Reconstruction accuracy of **2DHMM** and **2DHMM+REF**, for the two real datasets.

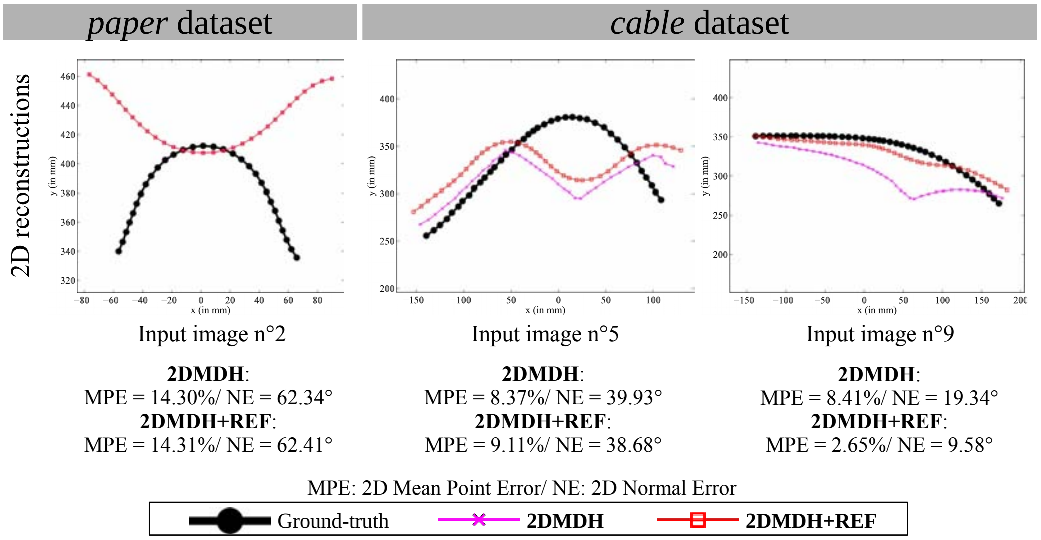


Fig. 19. Visual results and reconstruction accuracy of the category *(ii)* methods, **2DMDH** and **2DMDH+REF**, on a subset of input images used in figure 17.

6) Comparison with a Surface SfT method using the paper dataset: We propose to compare the performance of **2DHMM** and **2DHMM+REF** with the one of a state-of-the-art Surface SfT method. We used the analytical solution from [Chhatkuli et al., 2017] and refined it using [Brunet et al., 2014]. We denote **3DAnRe** this Surface SfT method. For this, we formed from the *paper* dataset the *full paper* dataset which consists of a surface template shown in figure 20 and the five 2D images from which we extracted the 1D input images of the *paper* dataset. The image size is 4800×3200 px. We computed with [Collins et al., 2014] the 2D-2D correspondences between the 2D texture-map of the surface template and each input image. We added to each set of correspondences the 30 correspondences used in the *paper* dataset, as described in §VI-A2. From input image n°1 to input image n°5, we then obtained 894, 354, 766, 889 and 787 correspondences. Figure 20 shows the 3D reconstructions for the same input images used in figure 17 . The reconstruction is evaluated on two different domains. The first domain is a grid of 20×20 points obtained from the interpolation of the 2D-2D correspondences using a B-spline. The second is the 30 correspondences used in the *paper* dataset. The first domain allows us to evaluate the global performance of the Surface SfT method and the second allows us to compare the performance of our Curve SfT methods to the Surface SfT method. We observe that globally **3DAnRe** reconstructs very well the deformations. The wrong deformations visible at the boundary of the paper can be explained by the fact that the regions at the boundary are less constrained. This is because

3DAnRe, as most SfT methods, only uses correspondence constraints which are sparse. We also note that the reconstructions at the 30 correspondences used in the *paper* dataset are accurate. These observations are confirmed by the reconstruction accuracy given in figure 21. The evaluation metrics are direct extensions of the ones presented in §VI-A3. We can see that, for the 30 correspondences used in the *paper* dataset, **3DAnRe** gives globally better reconstructions than our Curve SfT methods, **2DHMM** and **2DHMM+REF**. This is most noticeable in terms of normals. This was expected for two reasons. First, a Surface SfT method uses a surface template, which is a prior much stronger than a curvilinear template. Second, the reconstruction of the 30 correspondences used in the *paper* dataset is more constrained in **3DAnRe** because **3DAnRe** uses data over the whole surface with a factor of 10 to 30 on the number of correspondences.

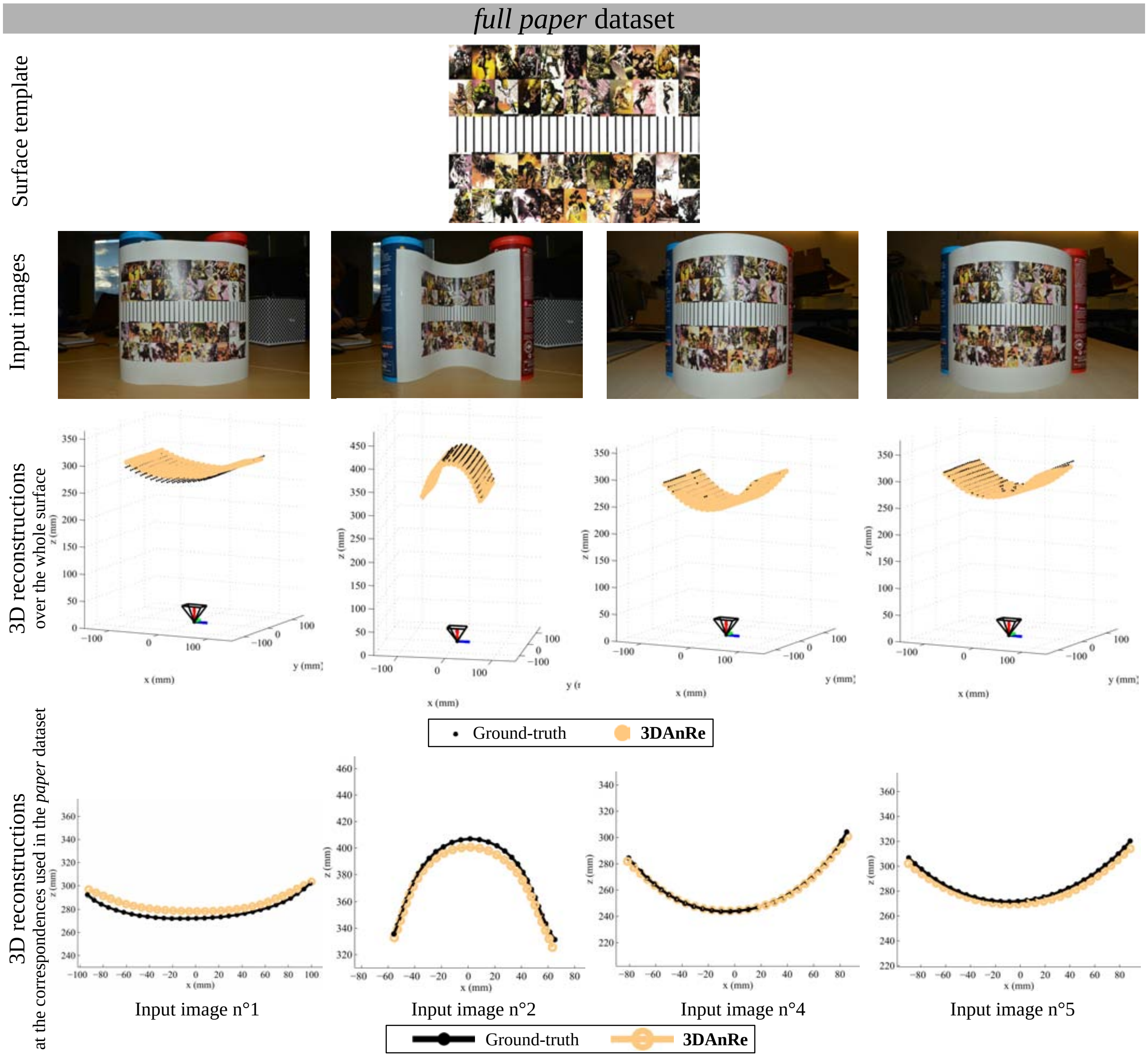


Fig. 20. Visual results of the Surface SfT method **3DAnRe** for the *full paper* dataset. We show the 3D reconstructions for the whole surface and for the correspondences used in the *paper* dataset. We also show their ground-truth solutions. As the 3D reconstructed curves are very close to the ground-truth solution, we refer readers to the digital version of the document for better visualization.

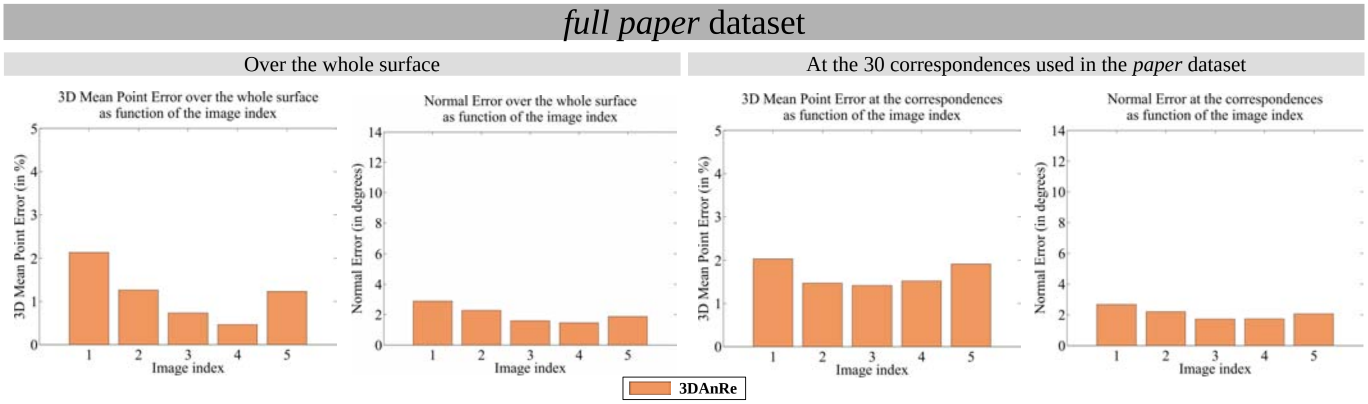


Fig. 21. Reconstruction accuracy of **3DAnRe** for the *full paper* dataset.

7) Timing Information: Table II gives the average processing times for **2DMDH**, **2DHMM** and the 2D non-convex refinement. We refer to §VI-A1 for the implementation details. We computed them by running each method on the 14 input images of the *convex-to-concave* dataset. Note that **2DHMM** comprises two steps: a preprocessing, referred as 2DHMM-1, and the construction and the inference of a single HMM, referred as 2DHMM-2. The step 2DHMM-1 includes common processings to all HMMs, *i.e.* the super critical point detection given in §V-B2, and the depth discretization given in §V-B5. The step 2DHMM-2 constructs and solves a single HMM, as explained in §V-B6, §V-B4 and §V-B3. The step 2DHMM-2 is performed 2^{N_s+1} times with N_s the number of super critical points detected in step 2DHMM-1. We then give the timing for both steps. We emphasize that the fourth row gives the processing time required in average to construct and solve a single HMM. Then, the time required to perform **2DHMM** and **2DHMM+REF** is respectively $t_{\textbf{2DHMM}} = t_{\text{2DHMM-1}} + t_{\text{2DHMM-2}}\, 2^{N_s+1}$ and $t_{\textbf{2DHMM+REF}} = t_{\textbf{2DHMM}} + t_{\text{2DREF}}\, 2^{N_s+1}$. For the *convex-to-concave* dataset, a single super critical point $N_s = 1$ is detected, which leads to a total processing time per input image of $t_{\textbf{2DHMM}} = 189.80$ sec and $t_{\textbf{2DHMM+REF}} = 189.88$ sec. This processing time can be drastically reduced by performing the second step "**2DHMM** - constructing and solving one HMM" in parallel for each candidate solution.

Method	Average processing time (seconds) per input image of the *convex-to-concave* dataset
2DMDH	2.13
2D non-convex refinement (t_{2DREF})	0.02
2DHMM - preprocessing ($t_{\text{2DHMM-1}}$)	2.06
2DHMM - constructing and solving one HMM ($t_{\text{2DHMM-2}}$)	46.93

TABLE II
Average processing time for **2DMDH**, the 2D non-convex refinement and **2DHMM**. For this, we used the *convex-to-concave* dataset which contains 14 input images with 30 correspondences and for which a single super critical point $N_s = 1$ is detected. Therefore, the processing time was averaged using 14 runs for **2DMDH**, $14 + 14(2^{N_s+1}) = 70$ runs for the 2D non-convex refinement, 14 runs for "**2DHMM** - preprocessing" and $14(2^{N_s+1}) = 56$ runs for "**2DHMM** - constructing and solving one HMM".

B. *SfT*$^{1\rightarrow3\rightarrow2}$ *Experiments*

1) Methods: Similarly to §VI-A, we only evaluate the HMM method, with and without refinement. Because the outputs of our methods are 3D curves, we name the methods respectively as **3DHMM** and **3DHMM+REF**. We show results of the category *(ii)* method, without and with refinement, denoted respectively as **3DMDH** and **3DMDH+REF**. We refer to §VI-A for the implementations details. To fix N_β and N_γ, we fitted the ground-truth 3D curves of the *3D cord* dataset using the angle-based parameterization with different polynomial orders for the polar angle β and the azimuthal angle γ. We used 16 different values of N_β and N_γ in the range of $[8, 38]$ and selected the polynomial orders N_β and N_γ using the same criteria as for the 2D angle-based parameterization. Table VI in appendix G gives the hyperparameters for each method for the sake of reproducibility.

2) Datasets:

a) Simulated dataset: We evaluated with a simulated dataset: the *3D cord* dataset. We built it using the software Blender [Blender, 2017]. We simulated a set of 40 spheres linked together by a Bézier curve. Moving the Bézier curve allows us to move the set of spheres to behave as points on a near-isometric curve. We simulated 15 curve deformations, rendered them on images of 960×540 px. The curves are placed on average at 200 mm of the camera center and the focal length is set to 35 mm. We created the 1D template by using the distance between the sphere centers along the Bézier curve. In each input image, we used the projection of the sphere centers as data points and match them with the 1D template. We added to the 2D image correspondences a gaussian noise of $\sigma = 2.0$ px. We show in figure 22 the simulated 2D curves for both datasets. Some examples of input images are shown in figure 22.

b) Real datasets: We tested our methods on two real scenes: the *necklace* and the *road* datasets. The first dataset is composed of one input image of a necklace laid over a pillow. Its 1D template is defined by the distance between the center of mass of the pearls, as shown in figure 27. We selected manually 28 correspondences between the 1D template and the 2D input image. The 2D input image is of size 3600×2800 px. The software Agisoft Lens [Agisoft, 2013] is used to calibrate the cameras and Agisoft Photoscan [Agisoft, 2014] is used to reconstruct the 3D scene for quantitative evaluation. The second dataset is composed of one input image of a road with a varying curvature, as figure 23 shows. Its 1D template is defined by the distance between each transition of road signs, which is standard (0.5 m). We selected manually 63 correspondences between the 1D template and the 2D input image by selecting the left corners of the road signs shown in the 2D input image, shown in figure 23. The 2D input image is of size 4608×3072 px. We acquired the ground-truth using an SfM method based on one essential matrix, triangulation and bundle adjustment [Hartley and Zisserman, 2003] from two views, shown in figure 23. One important feature of this dataset is that its true curve does not pass through any critical point, which means from theorem 2 that its true curve is not recoverable. We use the *road* dataset to illustrate two particular points: how do candidate solutions look like when the true curve is not recoverable and can the true curve be recovered if an initial condition is added in the category *(iv)* method.

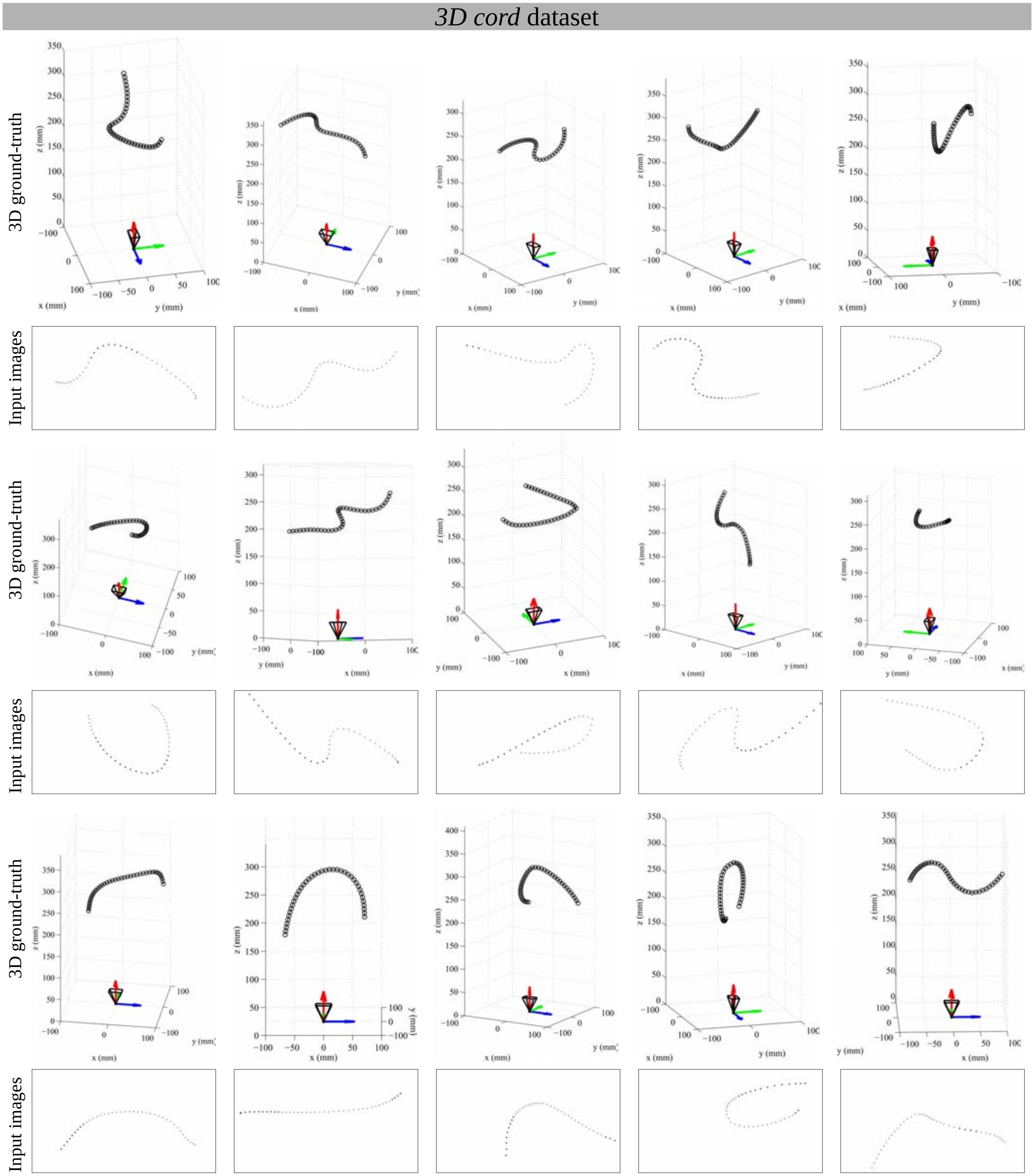
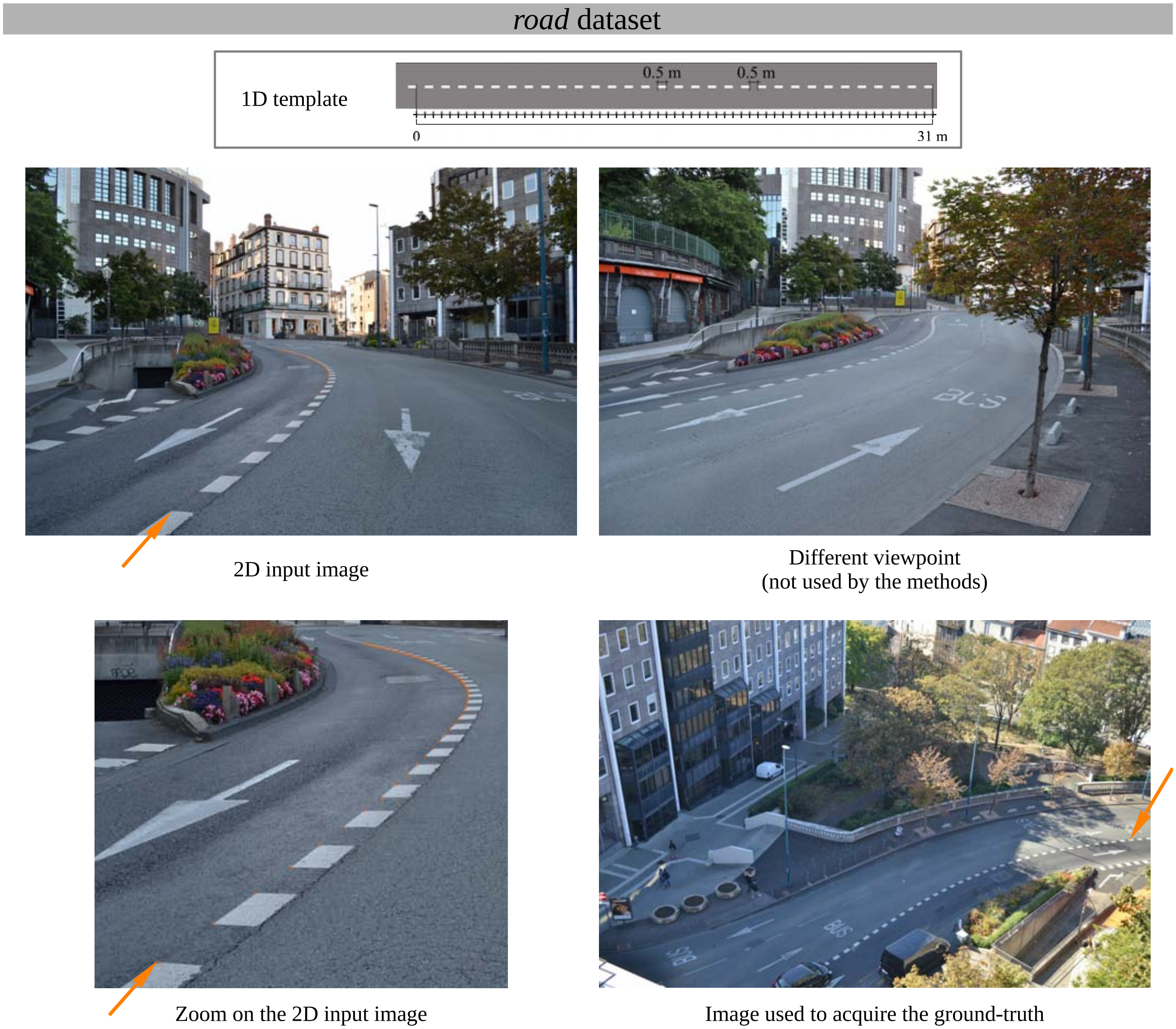


Fig. 22. Visualization of the ground-truth 3D curves for the simulated dataset, *3D cord*. We give the associated 2D input images.



2D input image

Different viewpoint
(not used by the methods)

Zoom on the 2D input image

Image used to acquire the ground-truth

Fig. 23. Visual results of **3DHMM** and **3DHMM+REF**, for the real dataset, *road*. In the first row, we give the 1D template and an illustration of the line of road signs. In the second row, we show the 2D input image and a different viewpoint of the same scene. The correspondences in the 2D input image are shown with orange crosses. In the third row, we show a zoom on the 2D input image in order to see clearly the correspondences and the image used to estimate the ground-truth. For a better understanding of the scene, we indicate the first correspondence with the orange arrows.

3) Evaluation Metrics:

a) 3D mean point error (MPE): This error is the trivial extension of the 2D mean point relative error.

b) 3D tangent error (TE): We use 3D tangent error to evaluate the shape accuracy on the 3D curves because of the ambiguity on the normals of 3D curves (the normal of $\varphi \in C^{\infty}(\mathcal{T}, \mathbb{R}^3)$ is defined up to a rotation about the curve's tangent vector). We denote the 3D tangent of $\hat{\varphi}$ at a template point u by $\hat{t}(u)$ and the 3D tangent of φ^* by $t^*(u)$. Similarly to §VI-A3b, we fit a spline to compute the tangents with a smoothing parameter of $1\mathrm{e}1$ for all datasets. We then compute the 3D tangent error (in degrees) between the reconstructed curve $\hat{\varphi}$ and the ground-truth shape φ^* at $\mathcal{G}$:

$$TE(\hat{\varphi}, \varphi^*, \mathcal{G}) = \frac{1}{G} \sum_{j=1}^{G} \cos^{-1}\left(\hat{t}^{\top}(u_j)\, t^*(u_j)\right). \tag{43}$$

4) Results on Simulated Datasets: We remind that, for the same reason developed in §VI-A, all reconstructions obtained with **3DHMM** and **3DHMM+REF** and shown in §VI-B4 and §VI-B5 were selected by searching the candidate solution with the lowest 3D mean point error given in §VI-B3.

In figure 24, we show the reconstruction errors for our methods, **3DHMM** and **3DHMM+REF**. They perform globally well, but we can note that **3DHMM+REF** performs less well than **2DHMM+REF**, which may be explained by the complexity of the angle-based parameterization of 3D curves compared than the one of 2D curves. In figure 25, we display some 3D reconstructions computed by our methods. We can see the results from different viewpoints and observe that **3DHMM** and **3DHMM+REF** provide shapes which are quite close to the ground-truth shapes. As for SfT$^{1\rightarrow2\rightarrow1}$, we note in figure 26 the limitation of the category *(ii)* method regarding the non-uniqueness of the problem. We show in figure 26 some failure cases of **3DMDH** and **3DMDH+REF**.

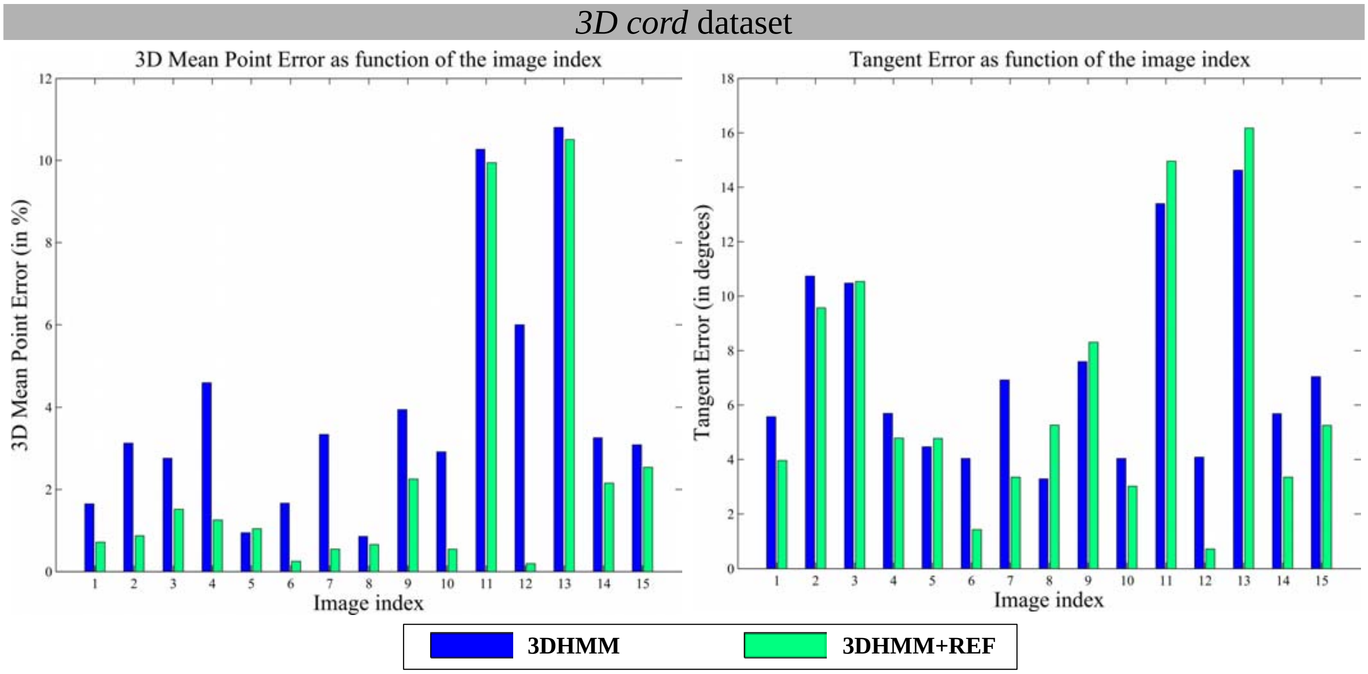


Fig. 24. Reconstruction accuracy of **3DHMM** and **3DHMM+REF**, for the simulated dataset, *3D cord*.

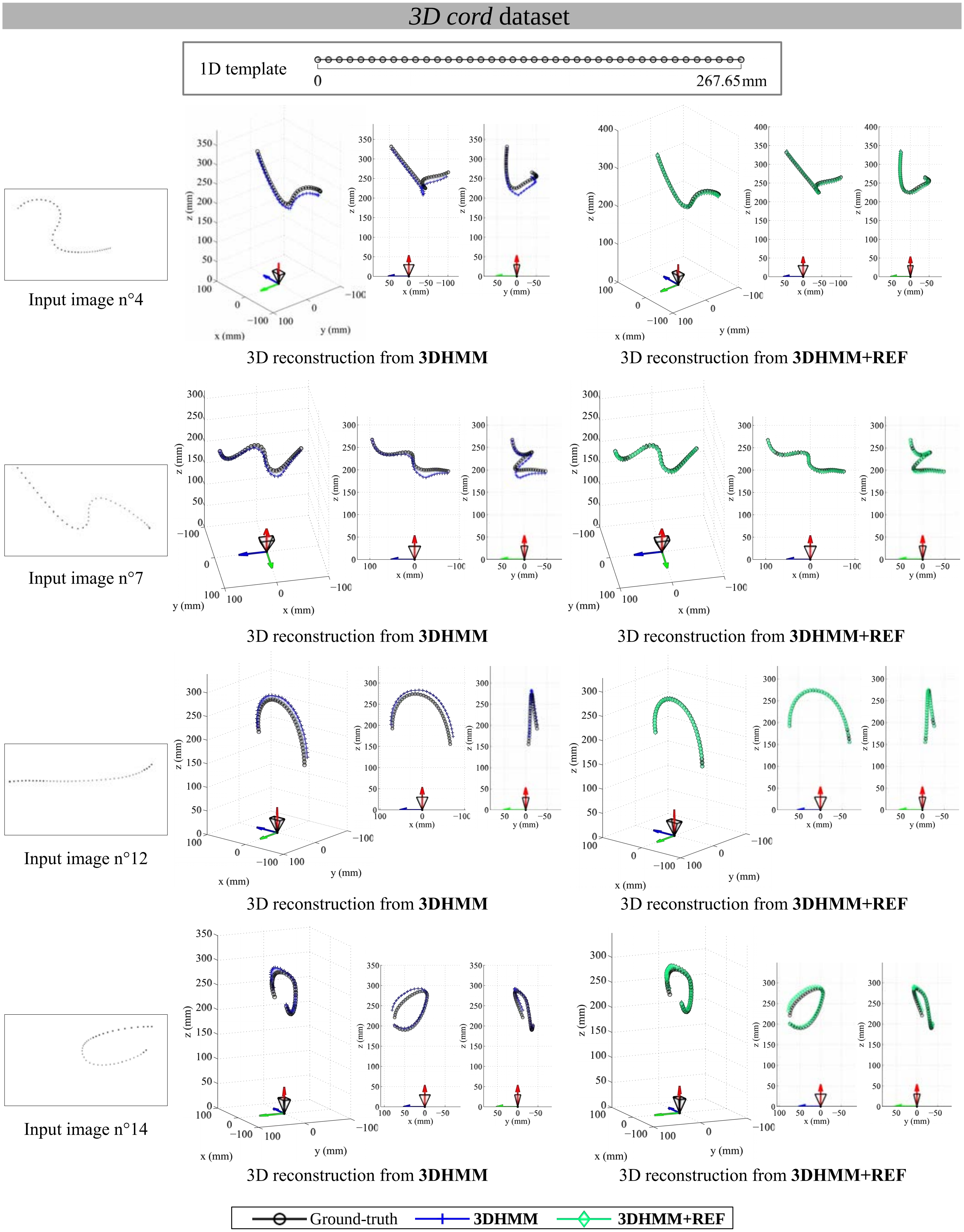


Fig. 25. Visual results of **3DHMM** and **3DHMM+REF**, for the simulated dataset, *3D cord*. We show the 3D reconstructed curves and the ground-truth solutions. Each row corresponds to one input image with the reconstructions given **3DHMM** and **3DHMM+REF**. For each reconstruction, we give three different viewpoints. As the 3D reconstructed curves are very close to the ground-truth solution, we refer the readers to the digital version of the document for better visualization.

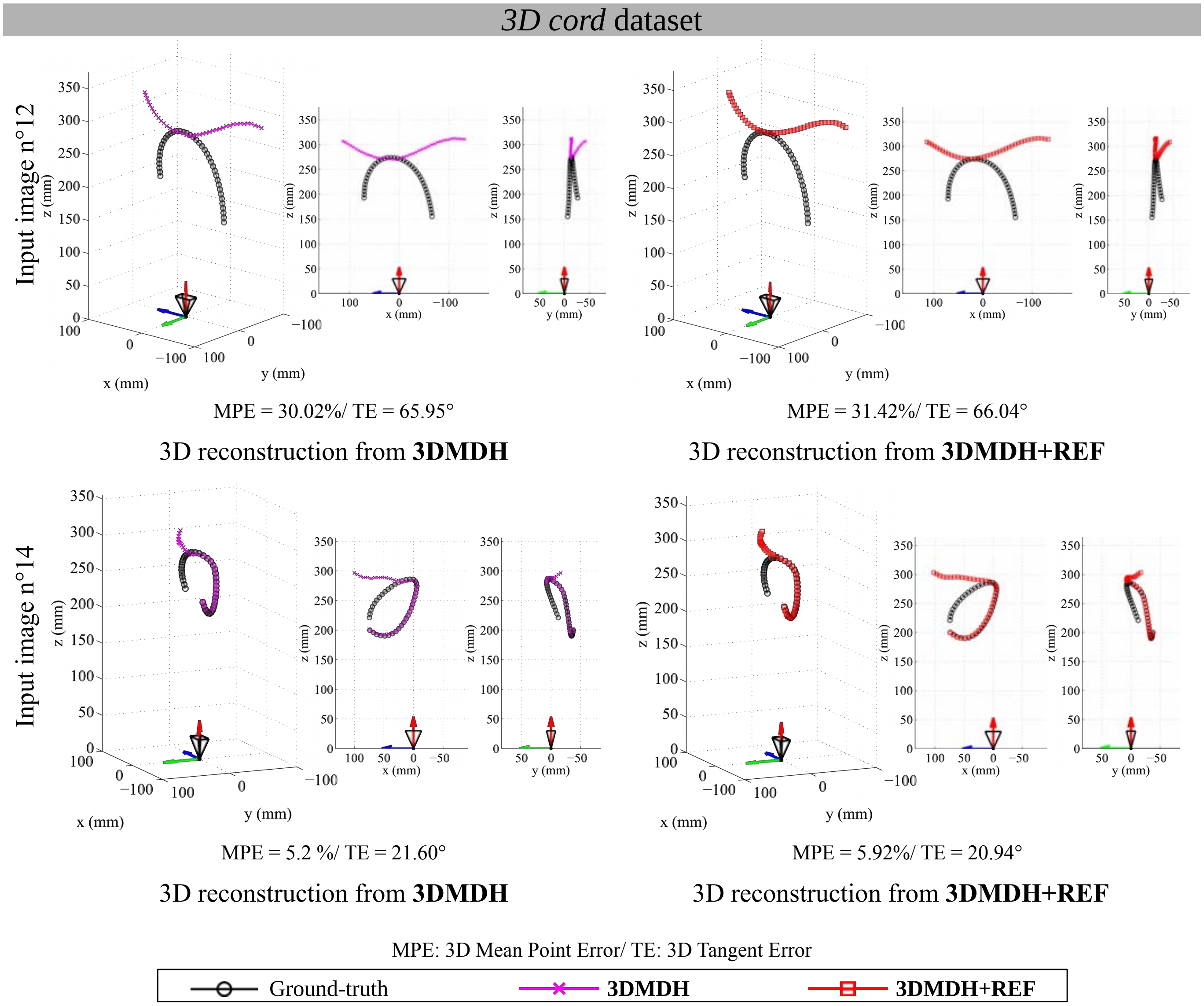
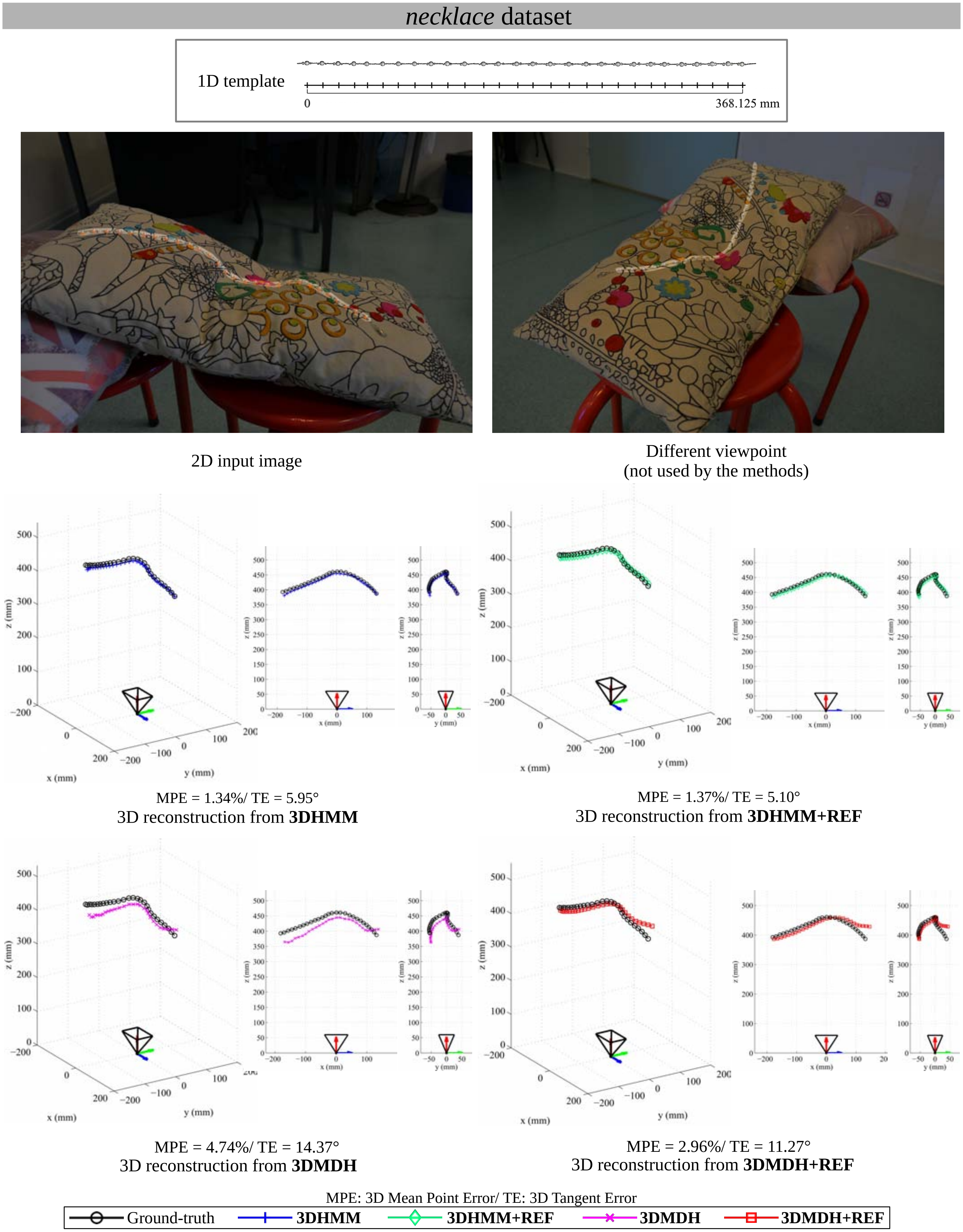


Fig. 26. Visual results and reconstruction accuracy of **3DMDH** and **3DMDH+REF** on the simulated dataset, *3D cord*. Each row corresponds to one input image with the reconstructions given by **3DMDH** and **3DMDH+REF**. The input images are shown in figure 25. For each reconstruction, we give three different viewpoints.

5) *Results on Real Datasets:*

a) Necklace dataset: We give the reconstructions of the *necklace* dataset in figure 27, for the categories *(ii)* and *(iv)* methods with and without refinement. We see that **3DMDH** does not give the correct solution: the curvature at the left end of the curve is wrong. The refinement **3DMDH+REF** cannot change the curvature, which may indicate it is a local minimum. However, we observe that **3DHMM** provides the correct solution, *i.e.* the correct curvature along the curve, which is supported by very good reconstruction accuracy, shown on the bottom of each reconstruction in figure 27.



necklace dataset

1D template

2D input image

Different viewpoint
(not used by the methods)

MPE = 1.34%/ TE = 5.95°
3D reconstruction from **3DHMM**

MPE = 1.37%/ TE = 5.10°
3D reconstruction from **3DHMM+REF**

MPE = 4.74%/ TE = 14.37°
3D reconstruction from **3DMDH**

MPE = 2.96%/ TE = 11.27°
3D reconstruction from **3DMDH+REF**

MPE: 3D Mean Point Error/ TE: 3D Tangent Error

Fig. 27. Visual results and reconstruction accuracy of all methods, **3DHMM**, **3DHMM+REF**, **3DMDH** and **3DMDH+REF**, for the real dataset, *necklace*. In the first row, we show its 1D template and a picture of the necklace used to construct the dataset. In the second row, we show the 2D input image and a different viewpoint of the same scene. The 3D curve to reconstruct corresponds to center of mass of the pearls visible in the 2D input image with orange crosses. In the third row, we give the visual and reconstruction accuracy of **3DHMM** and **3DHMM+REF**. In the fourth row, we give the visual and reconstruction accuracy of **3DMDH** and **3DMDH+REF**. For each reconstruction, we give three different viewpoints.

b) Road dataset: Figure 28 gives the reconstruction of several candidate solutions generated by **3DHMM**. Note that $N_s = 4$ super critical points were detected, while there is no critical point. This over-detection may be explained by the strong perspective of the scene. The reconstruction in the fourth column is the closest curve to the true one among all candidate solutions, however its 3D position error is significantly high. This result illustrates how difficult is to recover the true curve in case where the true curve does not pass through any critical point.

In addition, we explored whether our HMM-based method could solve the *road* dataset correctly using additional boundary conditions. Specifically we introduce anchor points, which are points on the curve for which depth is provided *a priori*. In practice, these may be automatically determined with geometric scene understanding, but this is out of the paper's scope. We implemented this with an additional energy term added to equation (31). We used an energy encoding (*i.e.* a soft constraint) for two important reasons. First, depth is quantized by the model, so it is not possible to force real-valued anchor constraints exactly. Second, there may be noise in the anchor depths, so exactly satisfying the anchor depth is not desirable. Currently we assume that each anchor corresponds with a node in the graph, which leads to a unary anchor energy term. In practice, this works when either the curve is densely sampled by graph nodes, or when the anchor is introduced as a graph node at the time of graph construction. Note that for sparser graphs, it may be possible to encode anchor constraints by interpolating the anchor's adjacent nodes, which would then lead to a second-order energy term. Here, we only considered the unary approach.

Suppose node i is an anchor point. We define its anchor energy is as follows: $E_a(d_i; d_i^*) = (d_i - d_i^*)^2$, with d_i^* denoting the provided node's depth. We denote the category *(iv)* method which uses HMM with anchor energies by **3DHMM'**, while **3DHMM** refers to the category *(iv)* method which uses HMM without anchor energies.

We also add anchor constraints to the refinement of the cost function (37). This is defined as C_a, which acts like E_a and is defined as $C_a(\mathbf{b}, \mathbf{g}, t_x, t_y, t_z; u_i, \mathbf{Q}_i^*) = (\varphi(u_i; \mathbf{b}, \mathbf{g}, t_x, t_y, t_z) - \mathbf{Q}_i^*)^2$, with $\mathbf{Q}_i^*$ denoting the depth of node i. We denote **3DHMM'+REF'** as the process of solving the HMM with anchor terms and refining the solution with anchor terms.

We evaluated performance on the *road* dataset by placing one anchor at three different positions along the curve (first, middle and end nodes). Figure 29 shows the reconstructions and the reconstruction accuracy with an anchor positioned at the first node. The reconstructions from **3DHMM'** and **3DHMM'+REF'** are very accurate, as the reconstruction accuracy underlines. We observe a similar outcome with the anchor placed at the middle and the end node. By comparing the reconstruction accuracy of the best reconstruction without anchor point in figure 28 and the one of the reconstructions with anchor point in table III, we observe that the reconstruction errors obtained using each anchor point are significantly lower than without anchor point. We obtain an average error reduction of 95.04% in MPE and 81.53% in TE between the best reconstruction without anchor point from **3DHMM** and the reconstructions with anchor point from **3DHMM'**. These results demonstrate that ill-posed configurations of Curve SfT, *i.e.* when the true curve does not present any critical point, can be resolved using anchor points, leading also to very accurate reconstructions. However, an interesting future work would be to extend our theory to the well-posedness of Curve SfT in the case of true curves without any critical point, by considering the position and the number of anchor points along the curve.

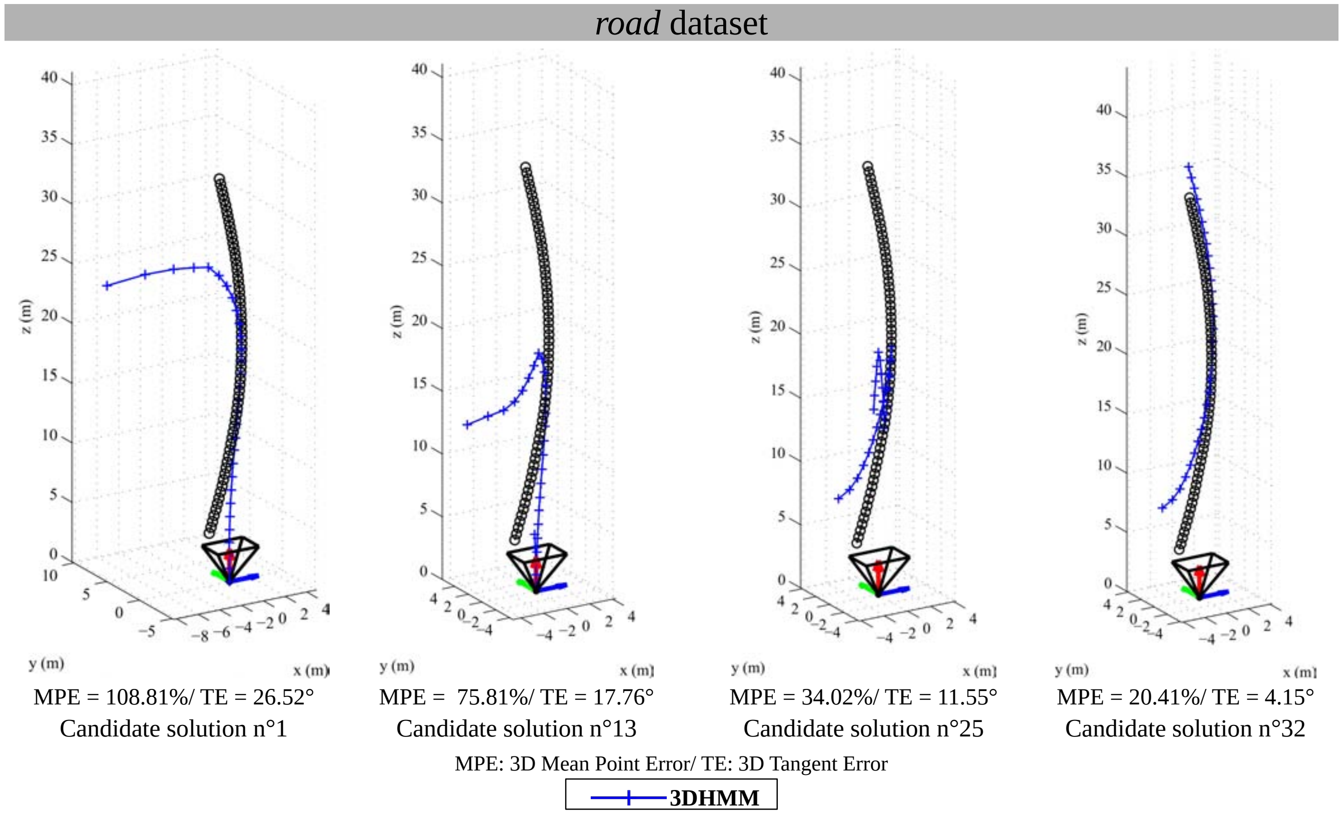


Fig. 28. Visual results and reconstruction accuracy of several candidate solutions given by **3DHMM** for the *road* dataset. The input image of this dataset is given in figure 23. We remind that the true curve for the input image does not pass through any critical point. The number of detected super critical points is $N_s = 4$, which leads to a total of 32 candidate solutions. For a better visualization of the road curvature, we selected a viewpoint far off from the viewpoint of the input image shown in figure 23.

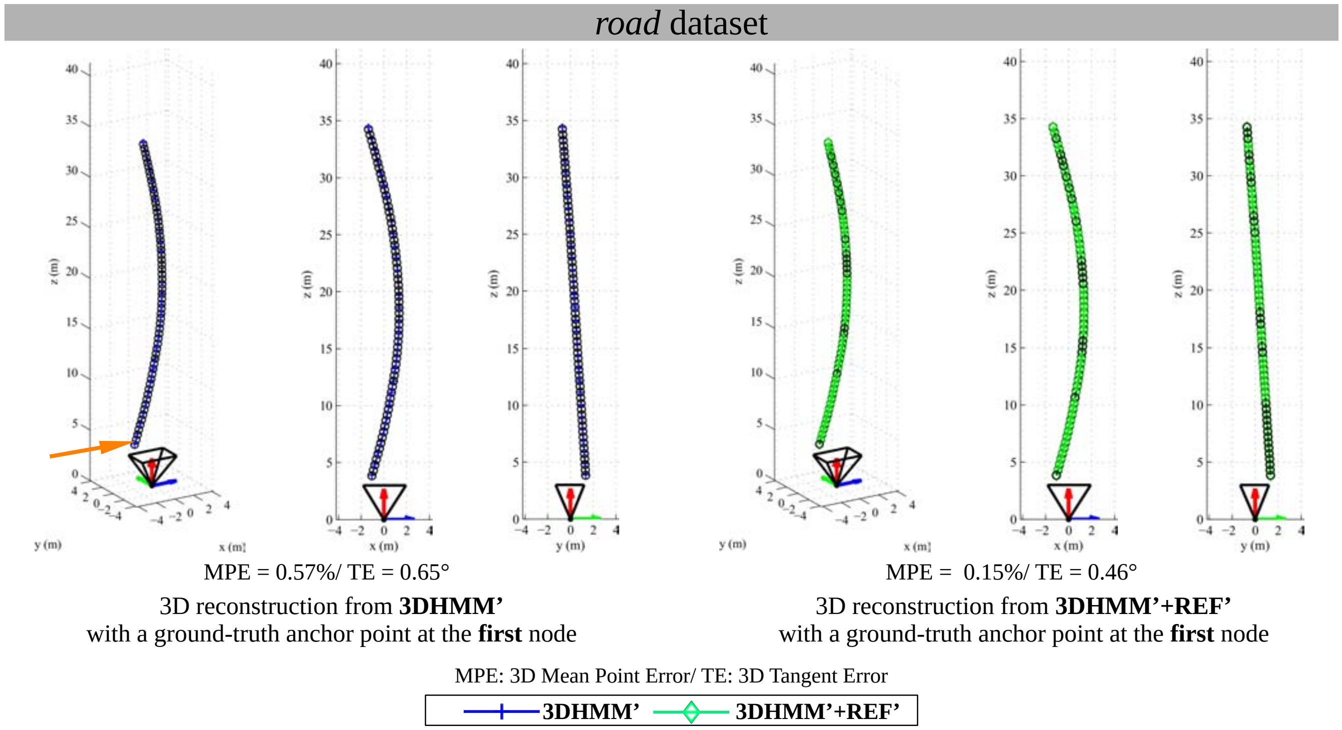


Fig. 29. Visual results and reconstruction accuracy of **3DHMM'** and **3DHMM'+REF'** with the ground-truth anchor point, for the real dataset, *road*. We show the reconstructions obtained when the anchor point is placed the first point of the curve. We give, for both reconstruction methods, three different viewpoints. As the 3D reconstructed curves are very close to the ground-truth solution, we refer the readers to the digital version of the document for better visualization. For a better understanding of the scene, we indicate the first node with an orange arrow.

Location of the anchor point along the curve	**3DHMM'** MPE/ TE	**3DHMM'+REF'** MPE/ TE
First node	$0.57\%/\ 0.65°$	$0.15\%/\ 0.46°$
Middle node	$1.31\%/\ 0.83°$	$0.19\%/\ 0.44°$
End node	$1.16\%/\ 0.82°$	$0.15\%/\ 0.46°$

TABLE III
RECONSTRUCTION ACCURACY OF **3DHMM'** AND **3DHMM'+REF'** AS A FUNCTION OF THE LOCATION OF THE ANCHOR POINT, FOR THE *road* DATASET.

6) Timing Information: Table IV gives the average processing times for **3DMDH**, **3DHMM** and the 3D non-convex refinement. We refer to §VI-B1 for the implementation details. We computed them by running each method on the 15 input images of the *3D cord* dataset. Note again that the fourth row gives the computational time required on average to construct and solve a single HMM. We observe that the processing times for the proposed $\mathrm{SfT}^{1\to3\to2}$ methods are higher than the ones for the proposed $\mathrm{SfT}^{1\to2\to1}$ methods, however the order of magnitude is similar. We explain the differences mainly by the fact that $\mathrm{SfT}^{1\to3\to2}$ has one more dimension in the input data and the reconstruct data, which increases the number of equations for the category *(ii)* method, the number of unknowns to optimize for the category *(iii)* method and the number of computations for the category *(iv)* method.

Method	Average processing time (seconds) per input image of the *3D cord* dataset
3DMDH	3.12
3D non-convex refinement	0.07
3DHMM - preprocessing	2.65
3DHMM - constructing and solving one HMM	73.73

TABLE IV
AVERAGE PROCESSING TIME FOR **3DMDH**, THE 3D NON-CONVEX REFINEMENT AND **3DHMM**. FOR THIS, WE USED THE *3D cord* DATASET WHICH CONTAINS 15 INPUT IMAGES WITH 40 CORRESPONDENCES AND FOR WHICH THE NUMBER OF DETECTED SUPER CRITICAL POINT VARY BETWEEN 2 TO 8. THEREFORE, THE PROCESSING TIME WAS AVERAGED USING 15 RUNS FOR **3DMDH**, $15 + 992 = 1007$ RUNS FOR THE 3D NON-CONVEX REFINEMENT, 15 RUNS FOR "**3DHMM** - PREPROCESSING" AND 992 RUNS FOR "**3DHMM** - CONSTRUCTING AND SOLVING ONE HMM".

C. Limitations and Failure Modes

We discuss here the main limitations and the failure modes of our solutions to Curve SfT. One limitation is that the parameters of our methods are manually set and they may vary for some datasets. However, as the number of tuned parameters is relatively small, this is not a critical issue. Another limitation is that our methods work only for isometric deformations. This assumption is essential since it allows us to prove our theoretical results and construct our computational solutions (category *(i)* to category *(iv)*). An important limitation is our assumption on the correspondences between the template and the input image, which we give in §III-A1. The correspondences should be sufficiently dense so that the warp can be estimated through a smooth interpolation. When this assumption is not met in practice, we face the main failure modes of our computational solutions. For methods of all categories, the failure mode is that there is not enough motion information to infer the whole curvature.

For the HMM solution, the failure mode is that the detection of super critical points and thus the reconstruction accuracy can be significantly impacted.

VII. Conclusion

We have presented a theoretical study of the isometric Curve SfT problem and its implementation to recover respectively 2D and 3D curves using a 1D template. We have revealed the complexity of both problems, SfT$^{1\to2\to1}$ and SfT$^{1\to3\to2}$, thanks to a differential analysis. We have arrived at a deep understanding of Curve SfT using the very informative super critical points which can be detected directly from the input data. The main theoretical outcome is that, when Curve SfT has N_s super critical points, there exist 2^{N_s+1} candidate solutions. We note that usual method categories to solve SfT cannot handle such ambiguities. Methods from categories *(i)* and *(ii)* only estimate one solution among all and methods from category *(iii)* only refines a given solution. We have then proposed a new category *(iv)* of methods which is based on discrete HMM. It estimates all candidate solutions by taking advantage of our theory and more precisely by using the super critical points. Critical points are very informative and have allowed us to constrain specifically each candidate solution of the problem. We have also given several methods to detect the super critical points from the warp function. We have provided an evaluation of the category *(iv)* method, with and without the refinement solution *(iii)*, thanks to simulated and real datasets and have studied the influence of the super critical point detection on the reconstruction accuracy. We have presented satisfying reconstruction accuracy of the category *(iv)* method and its refined version and have illustrated the inherent limitation of the convex optimization method (category *(ii)*), showing inaccurate reconstruction results. These results encourage us to see if such super critical points can be found in other 3D reconstruction problems and how graph-based approaches, such as HMM, can be employed to solve other 3D reconstruction problems. An interesting future work would be to study the different interpolation functions for the warp function and the angle functions.

Acknowledgments. We thank Armine Vardazaryan for her help in creating the simulated dataset *3D cord* and the Conseil Départemental du Puy-de-Dôme which allows us to acquire the ground-truth of the *road* dataset. We also thank Bastien Durix for his valuable discussions about the case of closed curvilinear templates. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

Appendix A
SfT$^{1\to3\to2}$: Proof of Proposition 3 - Critical Point Definition in φ

Proof. We start by writing $\mathbf{J}_{\bar\eta}$ as a function of $\hat\varphi$ from equation (4):

$$\mathbf{J}_{\bar\eta} = \frac{\hat\varphi_z \mathbf{J}_{\hat\varphi} - \hat\varphi'_z \hat\varphi}{\hat\varphi_z^2}. \tag{44}$$

We substitute equation (44) in equation (10), then express ξ as a function of $\hat\varphi$ and $\mathbf{J}_{\hat\varphi}$:

$$\begin{aligned} \xi &= \frac{1}{\|\bar\eta\|^2}\left(\frac{1}{\hat\varphi_z^4}\left(\hat\varphi_z\mathbf{J}_{\hat\varphi} - \hat\varphi'_z\hat\varphi\right)^2 - \frac{1}{\|\bar\eta\|^2}\frac{1}{\hat\varphi_z^6}\left(\hat\varphi_z\mathbf{J}_{\hat\varphi}^\top\hat\varphi - \hat\varphi'_z\hat\varphi^\top\hat\varphi\right)\left(\hat\varphi_z\hat\varphi^\top\mathbf{J}_{\hat\varphi} - \hat\varphi'_z\hat\varphi^\top\hat\varphi\right)\right) \\ &= \frac{1}{\|\bar\eta\|^2}\frac{1}{\hat\varphi_z^4}\left(\left(\hat\varphi_z\mathbf{J}_{\hat\varphi} - \hat\varphi'_z\hat\varphi\right)^2 - \frac{1}{\|\hat\varphi\|^2}\left(\hat\varphi_z\mathbf{J}_{\hat\varphi}^\top\hat\varphi - \hat\varphi'_z\hat\varphi^\top\hat\varphi\right)\left(\hat\varphi_z\hat\varphi^\top\mathbf{J}_{\hat\varphi} - \hat\varphi'_z\hat\varphi^\top\hat\varphi\right)\right). \end{aligned} \tag{45}$$

We expand equation (45) and simplify:

$$\begin{aligned}\xi &= \frac{1}{\|\bar\eta\|^2}\frac{1}{\hat\varphi_z^4}\left(\hat\varphi_z^2\mathbf{J}_{\hat\varphi}^\top\mathbf{J}_{\hat\varphi} - \frac{\hat\varphi_z^2}{\|\hat\varphi\|^2}\mathbf{J}_{\hat\varphi}^\top\hat\varphi\hat\varphi^\top\mathbf{J}_{\hat\varphi}\right)\\ &= \frac{1}{\|\bar\eta\|^2}\frac{1}{\hat\varphi_z^2}\frac{1}{\|\hat\varphi\|^2}\left(\hat\varphi^\top\hat\varphi\mathbf{J}_{\hat\varphi}^\top\mathbf{J}_{\hat\varphi} - \mathbf{J}_{\hat\varphi}^\top\hat\varphi\hat\varphi^\top\mathbf{J}_{\hat\varphi}\right). \qquad (46)\end{aligned}$$

We use definition 1 which gives $\hat\theta^2(u_c)\xi(u_c) = 1$ if and only if u_c is a critical point. For this, we express $\hat\theta^2\xi$ as a function of $\hat\varphi$ and $\mathbf{J}_{\hat\varphi}$:

$$\begin{aligned}\hat\theta^2\xi &= \hat\varphi_z^2\,\|\bar\eta\|^2\frac{1}{\|\bar\eta\|^2}\frac{1}{\hat\varphi_z^2}\frac{1}{\|\hat\varphi\|^2}\left(\hat\varphi^\top\hat\varphi\mathbf{J}_{\hat\varphi}^\top\mathbf{J}_{\hat\varphi} - \mathbf{J}_{\hat\varphi}^\top\hat\varphi\hat\varphi^\top\mathbf{J}_{\hat\varphi}\right)\\ &= \frac{1}{\|\hat\varphi\|^2}\left(\hat\varphi^\top\hat\varphi\mathbf{J}_{\hat\varphi}^\top\mathbf{J}_{\hat\varphi} - \mathbf{J}_{\hat\varphi}^\top\hat\varphi\hat\varphi^\top\mathbf{J}_{\hat\varphi}\right). \qquad (47)\end{aligned}$$

We now replace $\hat\varphi$ by its three components $\hat\varphi_x$, $\hat\varphi_y$ and $\hat\varphi_z$:

$$\hat\theta^2\xi = \frac{1}{\|\hat\varphi\|^2}\left(\left(\hat\varphi_x^2+\hat\varphi_y^2+\hat\varphi_z^2\right)\left(\hat\varphi_x'^2+\hat\varphi_y'^2+\hat\varphi_z'^2\right) - \left(\hat\varphi_x\hat\varphi_x+\hat\varphi_y\hat\varphi_y'+\hat\varphi_z\hat\varphi_z'\right)^2\right). \qquad (48)$$

By expanding equation (48) and simplifying, we obtain:

$$\begin{aligned}\hat\theta^2\xi &= \frac{1}{\|\hat\varphi\|^2}\left(\left(\hat\varphi_x\hat\varphi_y'+\hat\varphi_x'\hat\varphi_y\right)^2+\left(\hat\varphi_x\hat\varphi_z'+\hat\varphi_x'\hat\varphi_z\right)^2+\left(\hat\varphi_y\hat\varphi_z'+\hat\varphi_y'\hat\varphi_z\right)^2\right)\\ &= \frac{\|\hat\varphi\times\mathbf{J}_{\hat\varphi}\|^2}{\|\hat\varphi\|^2}. \qquad (49)\end{aligned}$$

We now reintroduce u_c to use definition 1:

$$\begin{aligned}(u_c \text{ is a critical point}) &\Leftrightarrow \left(\hat\theta^2(u_c)\xi(u_c) = \frac{\|\hat\varphi(u_c)\times\mathbf{J}_{\hat\varphi}(u_c)\|^2}{\|\hat\varphi(u_c)\|^2} = 1\right)\\ &\Leftrightarrow \left(\hat\varphi^\top(u_c)\;\mathbf{J}_{\hat\varphi}(u_c) = 0\right). \qquad (50)\end{aligned}$$

□

Appendix B

SfT$^{1\to3\to2}$: Proof of Proposition 4 - The Set of Super Critical Points

Proof. We first demonstrate that given a solution $\hat\varphi$ and a critical point u_c, then u_c is also a critical point of φ_s. From definition 1, we have $\hat\theta'(u_c) = 0$, which gives:

$$\hat\theta(u_c) = \frac{1}{\sqrt{\xi(u_c)}}. \qquad (51)$$

From equations (51) and (13), we have $\hat\theta(u_c) = \theta_s(u_c)$. Therefore the two curves meet at u_c. To demonstrate that u_c is also a critical point of φ_s, we first differentiate equation (10) to obtain the following second-order ODE:

$$2\theta'\theta'' + \xi'\theta^2 + 2\xi\theta\theta' = 0. \qquad (52)$$

Because $\hat\theta$ is a solution to the ODE (10), at u_c we have by substituting equation (52):

$$\xi'(u_c)\hat\theta(u_c)^2 = 0. \qquad (53)$$

We then differentiate equation (13) to obtain the following constraint on φ_s at u_c:

$$\xi(u_c)\theta_s(u_c)\theta_s'(u_c) + \xi'(u_c)\theta_s(u_c)^2 = 0. \tag{54}$$

We substitute equation (53) into equation (54) and use $\hat{\theta}(u_c) = \theta_s(u_c)$ to obtain:

$$\xi(u_c)\theta_s(u_c)\theta_s'(u_c) = 0. \tag{55}$$

Because $\hat{\varphi}$ is a solution of the ODE (10) and u_c a critical point, we have $\xi(u_c)\hat{\theta}(u_c)^2 = 1$, so $\xi(u_c)$ and $\theta_s(u_c)$ cannot be null. We then have $\theta_s'(u_c) = 0$ and thus u_c is also a critical point of φ_s. □

Appendix C

SfT$^{1\to3\to2}$: Proof of Proposition 5 - Super Critical Point Identities

Proof. Derivation of the first identity. We derive a necessary and sufficient condition on η that is valid at super critical points. We assume $\hat{\varphi}$ is a solution to equation (4) with u_s being a super critical point. We first differentiate equation (10) to form the following ODE:

$$2\theta'\theta'' + \xi'\theta^2 + 2\xi\theta\theta' = 0. \tag{56}$$

We know that $\hat{\theta} = \varepsilon\hat{\varphi}_y$ is a solution to equation (56), and $\hat{\theta}'(u_s) = 0$ from definition 1. We substitute $\hat{\theta}$ in equation (56) and evaluate the result at u_s, obtaining the following:

$$\xi'(u_s)\hat{\theta}^2(u_s) = 0. \tag{57}$$

Derivation of the second identity. We know $\hat{\theta}^2(u_s) \neq 0$, otherwise $\hat{\varphi}$ would pass through the camera's origin at u_s. We also have that $\xi'(u_s) = 0$ from the first super critical point identity. The second identity is found by differentiating ξ as defined in equation (10). To express ξ' as a function of η and its derivatives, we first define two intermediate terms, A_η and B_η, and express ξ' using A_η, B_η and their first derivatives:

$$A_\eta = \mathbf{J}_\eta^\top \mathbf{J}_\eta - \frac{1}{\varepsilon^2} B_\eta \quad \text{and} \quad B_\eta = \mathbf{J}_\eta^\top \eta \eta^\top \mathbf{J}_\eta, \tag{58}$$

$$A_\eta' = \mathbf{H}_\eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \mathbf{H}_\eta - \frac{1}{\varepsilon^4}\left(\varepsilon^2 B_\eta' - 2\varepsilon'\varepsilon B_\eta\right), \tag{59}$$

$$B_\eta' = \mathbf{H}_\eta^\top \eta\eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \mathbf{J}_\eta \eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \eta \mathbf{J}_\eta^\top \mathbf{J}_\eta + \mathbf{J}_\eta^\top \eta\eta^\top \mathbf{H}_\eta. \tag{60}$$

Because η, $\mathbf{J}_\eta$ and $\mathbf{H}_\eta$ are $\mathbb{R}^2$-vector, $\mathbf{H}_\eta^\top \eta\eta^\top \mathbf{J}_\eta = \mathbf{J}_\eta^\top \eta\eta^\top \mathbf{H}_\eta$ and $\mathbf{J}_\eta^\top \mathbf{J}_\eta \eta^\top \mathbf{J}_\eta = \mathbf{J}_\eta^\top \eta \mathbf{J}_\eta^\top \mathbf{J}_\eta$, which simplifies B_η':

$$B_\eta' = 2\mathbf{J}_\eta^\top \eta\eta^\top \mathbf{H}_\eta + 2\mathbf{J}_\eta^\top \eta \mathbf{J}_\eta^\top \mathbf{J}_\eta. \tag{61}$$

From equations (58), (59) and (61), we have:

$$\begin{aligned}
\xi' &= \frac{1}{\varepsilon^4}\left(\varepsilon^2 A'_\eta - 2\varepsilon'\varepsilon A_\eta\right)\\
&= \frac{1}{\varepsilon^4}\left(2\varepsilon^2 \mathbf{J}_\eta^\top \mathbf{H}_\eta - B'_\eta + 2\frac{\varepsilon'}{\varepsilon}B_\eta - 2\varepsilon'\varepsilon \mathbf{J}_\eta^\top \mathbf{J}_\eta + 2\frac{\varepsilon'}{\varepsilon}B_\eta\right)\\
&= \frac{1}{\varepsilon^4}\left(2\varepsilon^2 \mathbf{J}_\eta^\top \mathbf{H}_\eta - 2\mathbf{J}_\eta^\top \eta\eta^\top \mathbf{H}_\eta - 2\mathbf{J}_\eta^\top \eta \mathbf{J}_\eta^\top \mathbf{J}_\eta + \frac{4}{\varepsilon^2}\eta^\top \mathbf{J}_\eta \mathbf{J}_\eta^\top \eta\eta^\top \mathbf{J}_\eta - 2\eta^\top \mathbf{J}_\eta \mathbf{J}_\eta^\top \mathbf{J}_\eta\right). \qquad (62)
\end{aligned}$$

Using $\eta^\top \mathbf{J}_\eta = \mathbf{J}_\eta^\top \eta$, we obtain:

$$\xi' = \frac{2}{\varepsilon^6}\left(\varepsilon^4 \mathbf{J}_\eta^\top \mathbf{H}_\eta - \eta^\top \mathbf{J}_\eta\left(\varepsilon^2 \eta^\top \mathbf{H}_\eta + 2\varepsilon^2 \mathbf{J}_\eta^\top \mathbf{J}_\eta - 2\mathbf{J}_\eta^\top \eta\eta^\top \mathbf{J}_\eta\right)\right), \qquad (63)$$

from which we have that $\xi'(u_s) = 0$ is equivalent to:

$$\begin{aligned}
&\varepsilon^4(u_s)\mathbf{J}_\eta^\top(u_s)\mathbf{H}_\eta(u_s) - \varepsilon^2(u_s)\eta^\top(u_s)\mathbf{J}_\eta(u_s)\eta^\top(u_s)\mathbf{H}_\eta(u_s)\\
&- 2\eta^\top(u_s)\mathbf{J}_\eta(u_s)\left(\varepsilon^2(u_s)\mathbf{J}_\eta^\top(u_s)\mathbf{J}_\eta(u_s) - \mathbf{J}_\eta^\top(u_s)\eta(u_s)\eta^\top(u_s)\mathbf{J}_\eta(u_s)\right) = 0. \qquad (64)
\end{aligned}$$

By substituting ε and ε' in terms of η and removing factors in equation (64), we obtain the following:

$$\begin{aligned}
&\|\bar{\eta}(u_s)\|^4 \mathbf{J}_\eta^\top(u_s)\mathbf{H}_\eta(u_s) - \|\bar{\eta}(u_s)\|^2 \eta^\top(u_s)\mathbf{J}_\eta(u_s)\eta^\top(u_s)\mathbf{H}_\eta(u_s)\\
&- 2\eta^\top(u_s)\mathbf{J}_\eta(u_s)\left(\|\bar{\eta}(u_s)\|^2 \mathbf{J}_\eta^\top(u_s)\mathbf{J}_\eta(u_s) - \mathbf{J}_\eta^\top(u_s)\eta(u_s)\eta^\top(u_s)\mathbf{J}_\eta(u_s)\right) = 0. \qquad (65)
\end{aligned}$$

This only depends on η and its derivatives.

Derivation of the third identity. We use the fact that, at any super critical point u_s, $\hat{\varphi}(u_s) = \varphi_s(u_s)$ (definition 2). We then use proposition 3 which says that the critical points of the super curve φ_s are the points where the tangent of φ_s and the line-of-sight are orthogonal. □

APPENDIX D

SFT$^{1\to 2\to 1}$: PROOF OF PROPOSITION 9 - CRITICAL POINT DEFINITION IN φ

Proof. We start by writing η' in function of $\hat{\varphi}$ from equation (17):

$$\eta' = \frac{\hat{\varphi}'_x \hat{\varphi}_y - \hat{\varphi}_x \hat{\varphi}'_y}{(\hat{\varphi}_y)^2}. \qquad (66)$$

We substitute equation (66) in equation (25) and equation (24) to express ξ and $\hat{\theta}$:

$$\xi = \frac{\left(\hat{\varphi}'_x \hat{\varphi}_y - \hat{\varphi}_x \hat{\varphi}'_y\right)^2}{\left(\hat{\varphi}_x^2 + \hat{\varphi}_y^2\right)^2} \qquad (67)$$

$$\hat{\theta} = \sqrt{\hat{\varphi}_x^2 + \hat{\varphi}_y^2}. \qquad (68)$$

We use definition 1 which gives $\hat{\theta}^2(u_c)\xi(u_c) = 1$ if and only if u_c is a critical point. For this, we express $\hat{\theta}^2\xi$ as a function of $\hat{\varphi}$ and $\hat{\varphi}'$:

$$\begin{aligned}\hat{\theta}^2\xi &= \frac{\left(\hat{\varphi}_x'\hat{\varphi}_y - \hat{\varphi}_x\hat{\varphi}_y'\right)\right)^2}{\hat{\varphi}_x^2 + \hat{\varphi}_y^2} \\ &= \frac{\|\hat{\varphi} \times \hat{\varphi}'\|^2}{\|\hat{\varphi}\|^2}. \end{aligned} \tag{69}$$

We now reintroduce u_c to use definition 1:

$$\begin{aligned}(u_c \text{ is a critical point}) &\Leftrightarrow \left(\hat{\theta}^2(u_c)\xi(u_c) = \frac{\|\hat{\varphi}(u_c) \times \hat{\varphi}'(u_c)\|^2}{\|\hat{\varphi}(u_c)\|^2} = 1\right) \\ &\Leftrightarrow \left(\hat{\varphi}^\top(u_c)\ \hat{\varphi}'(u_c) = 0\right). \end{aligned} \tag{70}$$

□

Appendix E

SfT$^{1\to2\to1}$: Proof of Proposition 10 - Super Critical Point Identites

Proof. We follow the same steps as the proof of proposition 5 and obtain that $\xi'(u_s) = 0$, which is the first characterization. A second one can be found by differentiating the analytical expression of ξ given in equation (25):

$$\xi' = \frac{2\varepsilon^4\eta'\eta'' - 4\varepsilon^3\varepsilon'\eta'^2}{\varepsilon^8}, \tag{71}$$

from which we have that $\xi'(u_s) = 0$ is equivalent to:

$$\frac{\varepsilon^3}{\eta'}\left(\varepsilon(u_s)\eta''(u_s) - 2\varepsilon'(u_s)\eta'(u_s)\right) = 0. \tag{72}$$

By substitution of ε and ε' in terms of η and removing factors in equation (72) we have the second identity:

$$2\eta(u_s)\eta'^2(u_s) - (1 + \eta^2(u_s))\eta''(u_s) = 0, \tag{73}$$

which only depends on η and its derivatives.

For the third characterization, we use the fact that, at any super critical point u_s, $\hat{\varphi}(u_s) = \varphi_s(u_s)$ (definition 2). We then use proposition 9 and obtain that the critical points of the super curve φ_s are the points where the tangent of φ_s and the line-of-sight are orthogonal. □

Appendix F

Reconstruction Algorithm of Proposed Category *(iv)* Method for the SfT$^{1\to3\to2}$ Problem

We give here a pseudo-code of our proposed category *(iv)* method for the SfT$^{1\to3\to2}$ problem, which we present in §V-B. The pseudo-code for the SfT$^{1\to2\to1}$ problem differs from only one notation, the 2D correspondence in the input image $\mathbf{q}_k$ which becomes a scalar q_k as it is a 1D correspondence.

Algorithm 1 Reconstruction algorithm of our proposed category *(iv)* method for the SfT$^{1\rightarrow3\rightarrow2}$ problem.

Inputs:

the template: $\mathcal{T}$
the set of N 1D-2D correspondences: $\mathcal{S}_c = \{(u_k, \mathbf{q}_k)\}$
the smoothing parameter for the *template-to-image* warp: σ_η
the smoothing parameter for the function ξ: σ_ξ
the number of graph nodes: M
the number of depth samples: D

Output:

all candidate solutions: $\{\mathbf{Q}^j\}$

Index convention:

$k \in [1, N]$, where N is the number of 1D-2D correspondences
$j \in [1, N_s]$, where N_s is the number of detected super critical points
$i \in [1, M + N_s]$, where $M + N_s$ is the number of nodes

Notation convention:

$\mathscr{S}_{f,n}$ is the n^{th}-order spline function of the function f
u^s_j is the position of the j^{th} detected super critical point in the template
$\mathbf{s}^j$ is the j^{th} sign combination vector with $\mathbf{s}^j \in [-1, +1]^{N_s+1}$
$\{d^{MDH}_i\}$ is the set of depths estimated using the category *(ii)* method

```
Step 1: HMM Preprocessing
    ▷ Estimate the template-to-image warp:
  $\mathscr{S}_{\eta,0} \leftarrow \texttt{spaps}(\mathcal{S}_c, \sigma_\eta)$
    ▷ Detect super critical points (using the roots of $\xi'$, as explained in §V-B2):
    ▷ Estimate the the first-derivative of the $\xi$ function
  $\mathscr{S}_{\eta,1} \leftarrow \texttt{fnder}(\mathscr{S}_{\eta,0}, 1)$
  $\{\xi_k\} \leftarrow \texttt{computeXi}(\mathscr{S}_{\eta,0}, \mathscr{S}_{\eta,1}, \{u_k\})$        ▷ using equation (10)
  $\mathscr{S}_{\xi,0} \leftarrow \texttt{spaps}(\{(u_k, \xi_k)\}, \sigma_\xi)$
  $\mathscr{S}_{\xi,1} \leftarrow \texttt{fnder}(\mathscr{S}_{\xi,0}, 1)$
    ▷ Computing roots of $\xi'$
  $\{u^s_j\} \leftarrow \texttt{fnzeros}(\mathscr{S}_{\xi,1}, \mathcal{T})$
    ▷ Generate nodes of the HMM (as explained in §V-B3):
  $\{u_l\}_{l\in[1,M]} \leftarrow \texttt{sampleTemplate}(\mathcal{T}, M)$        ▷ as described in §V-B3
  $\mathcal{U} \leftarrow \texttt{addSuperCriticalPoints}(\{u_l\}, \{u^s_j\})$
  $\mathcal{Q} \leftarrow \texttt{fnval}(\mathscr{S}_{\eta,0}, \mathcal{U})$
    ▷ Define the intervals of state, $\mathcal{D}$:
  $\{d^{MDH}_i\} \leftarrow \texttt{MDH}(\mathcal{U}, \mathcal{Q})$        ▷ using category (ii) method described in §V-A2
  $\mathcal{D} \leftarrow \texttt{discretizeDepths}(\{d^{MDH}_i\}, D)$
    ▷ Define all sign combination vectors:
  $\{\mathbf{s}^j\}_{j\in[1,2^{N_s+1}]} \leftarrow \texttt{computeAllSignCombinationVectors}(N_s)$
    ▷ Initialize graph nodes, $\{d^{init}_i\}$:
  $d^{mean} \leftarrow \texttt{mean}(\{d^{MDH}_i\})$
  for $i \in 1$ to $M + N_s$ do
      $d^{init}_i \leftarrow d^{mean}$
  end for
Step 2: Construct and Solve all HMMs
  for $j \leftarrow 1$ to $2^{N_s+1}$ do
      $\mathcal{M} \leftarrow \texttt{computeEnergyMatrix}(\mathcal{D}, \mathbf{s}^j, \mathcal{U}, \mathcal{Q})$        ▷ using the energies defined in §V-B4
      $\{d^j_i\} \leftarrow \texttt{decodeEnergyMatrix}(\mathcal{M}, \{d^{init}_i\})$        ▷ as explained in §V-B6
      for $i \leftarrow 1$ to $M + N_s$ do
          $\mathbf{Q}^j_i \leftarrow d^j_i \bar{\mathbf{q}}_i$
      end for
  end for
```

Appendix G
Hyperparameters for $\mathrm{SfT}^{1\rightarrow2\rightarrow1}$ and $\mathrm{SfT}^{1\rightarrow3\rightarrow2}$ Experiments

		Category *(iv)* method §V-B			Category *(iii)* method §V-C	
	Datasets	Smoothing parameter for η	Smoothing parameter for ξ	Number of nodes M	Polynomial order N_α	Smoothing weight λ_{smooth}
$\mathrm{SfT}^{1\rightarrow2\rightarrow1}$	*convex-to-concave*	9e−6	1e−7	30	12	3e−4
	free-form	9e−8	1e−7	30	12	3e−4
	paper	3e−5	6e−13	30	12	3e−4
	cable	9e−4	6e−13	30	12	3e−4

TABLE V
Hyperparameter values for different category methods to solve $\mathrm{SfT}^{1\rightarrow2\rightarrow1}$ for all datasets.

		Category *(iv)* method §V-B			Category *(iii)* method §V-C	
	Datasets	Smoothing parameter for η	Smoothing parameter for ξ	Number of nodes M	Polynomial order N_β and N_γ	Smoothing weight λ_{smooth}
$\mathrm{SfT}^{1\rightarrow3\rightarrow2}$	*3D cord*	9e−6	1e−7	30	$12, 12$	3e−4
	necklace	9e−4	3e−13	30	$12, 12$	0.03
	road	3e−5	6e−13	30	$12, 12$	3e−4

TABLE VI
Hyperparameter values for different category methods to solve $\mathrm{SfT}^{1\rightarrow3\rightarrow2}$ for all datasets.

References

Agisoft. Agisoft Lens Version 0.4.1 beta 64 bit (build 1718). `http://www.agisoft.com`, 2013.

Agisoft. Agisoft PhotoScan version 1.2.3 build 2331 (64 bit). `http://www.agisoft.com`, 2014.

A. Bartoli and E. Özgür. A Perspective on Non-Isometric Shape-from-Template. In *ISMAR*, 2016.

A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-Template. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, October 2015.

R. Berthilsson, K. Åström, and A. Heyden. Reconstruction of General Curves, Using Factorization and Bundle Adjustment. *International Journal of Computer Vision*, 41(3):171–182, February 2001.

V. Blanz and T. Vetter. Morphable Model for the Synthesis of 3D Faces. In SIGGRAPH, 1999.

Blender. Blender 2.78a. `https://www.blender.org`, 2017.

F. Brunet, R. Hartley, and A. Bartoli. Monocular Template-Based 3D Surface Reconstruction: Convex Inextensible and Nonconvex Isometric Methods. *Computer Vision and Image Understanding*, 125:138–154, August 2014.

D. Casillas-Perez and D. Pizarro. Solutions of Quadratic First-Order ODEs applied to Computer Vision Problems. *ArXiv*, 1710.04265, 2017.

A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins. A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 39(5):833–850, May 2017.

T. Collins and A. Bartoli. Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences. In *ECCV*, 2014.

T. Collins and A. Bartoli. Realtime Shape-from-Template: System and Applications. In *ISMAR*, 2015.

T. Collins, P. Mesejo, and A. Bartoli. An Analysis of Errors in Graph-Based Keypoint Matching and Proposed Solutions. In *ECCV*, 2014.

T. Collins, A. Bartoli, N. Bourdel, and M. Canis. Dense, Robust and Real-time 3D Tracking of Deformable Organs in Monocular Laparoscopy. In *MICCAI*, 2016.

David 3D Scanner. `http://www.david-3d.com/en/products/david4`, 2014.

Y. Eliashberg and N. M. Mishachev. *Introduction to the h-Principle*. Number Grad. Stud. Math. 48. American Mathematical Society, 2002.

O. Faugeras and T. Papadopoulo. A Theory of the Motion Fields of Curves. *International Journal of Computer Vision*, 10 (2):125–156, April 1993.

M. Gallardo, D. Pizarro, A. Bartoli, and T. Collins. Shape-from-Template in Flatland. In *CVPR*, 2015.

M. Gallardo, T. Collins, and A. Bartoli. Using Shading and a 3D Template to Reconstruct Complex Surface Deformations. In *BMVC*, 2016.

N. Haouchine, J. Dequidt, M. O. Berger, and S. Cotin. Single View Augmentation of 3D Elastic Objects. In *ISMAR*, 2014.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.

F. Kahl and J. August. Multiview Reconstruction of Space Curves. In *ICCV*, 2003.

Q. Liu-Yin, R. Yu, L. Agapito, A. Fitzgibbon, and C. Russell. Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading. In *BMVC*, 2016.

J. Löfberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *International Symposium on Computer-Aided Control System Design*, 2004.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91–110, 2004.

F. Mai and Y. S. Hung. 3D Curves Reconstruction from Multiple Images. In *Digital Image Computing: Techniques and Applications*, 2010.

A. Malti and A. Bartoli. Combining Conformal Deformation and Cook-Torrance Shading for 3D Reconstruction in Laparoscopy. IEEE *Transactions on Biological Engineering*, 61(6):1684–1692, June 2014.

A. Malti, A. Bartoli, and T. Collins. A Pixel-Based Approach to Template-Based Monocular 3D Reconstruction of Deformable Surfaces. In *Proceedings of the* IEEE *International Workshop on Dynamic Shape Capture and Analysis at ICCV*, 2011.

A. Malti, R. Hartley, A. Bartoli, and J. Kim. Monocular Template-Based 3D Reconstruction of Extensible Surfaces with Local Linear Elasticity. In *CVPR*, 2013.

A. Malti, A. Bartoli, and R. I. Hartley. A Linear Least-Squares Solution to Elastic Shape-from-Template. In *CVPR*, 2015.

H. Martinsson, F. Gaspard, A. Bartoli, and J. Lavest. Energy-Based Reconstruction of 3D Curves for Quality Control. In *EMMCVPR*, 2007.

T. D. Ngo, S. Park, A. A. Jorstad, A. Crivellaro, C. Yoo, and P. Fua. Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. In *ICCV*, 2015.

T. D. Ngo, J. Östlund, and P. Fua. Template-based Monocular 3D Shape Recovery using Laplacian Meshes. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 38(1):172–187, 2016.

E. Özgür and A. Bartoli. Particle-SfT: A Provably-Convergent, Fast Shape-from-Template Algorithm. *International Journal of Computer Vision*, 123(2):184–205, June 2017.

S. Parashar, D. Pizarro, A. Bartoli, and T. Collins. As-Rigid-As-Possible Volumetric Shape-from-Template. In *ICCV*, 2015.

M. Perriollat, R. Hartley, and A. Bartoli. Monocular Template-Based Reconstruction of Inextensible Surfaces. *International Journal of Computer Vision*, 95(2):124–137, November 2011.

D. Pizarro and A. Bartoli. Feature-Based Deformable Surface Detection with Self-Occlusion Reasoning. *International Journal of Computer Vision*, 97(1):54–70, March 2012.

L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 77(2): 257–286, February 1989.

L. Robert and O. Faugeras. Curve-based stereo: Figural continuity and curvature. In *CVPR*, 1991.

M. Salzmann and P. Fua. Reconstructing Sharply Folding Surfaces: A Convex Formulation. In *CVPR*, 2009.

M. Salzmann and P. Fua. Linear Local Models for Monocular Reconstruction of Deformable Surfaces. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, May 2011.

M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface Deformation Models for Nonrigid 3D Shape Recovery. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1481–1487, August 2007.

C. Sbert and A. F. Sol. 3D Curves Reconstruction Based on Deformable Models. *Journal of Mathematical Imaging and Vision*, 18:211–223, 05 2003.

M. Schmidt. UGM: A Matlab toolbox for probabilistic undirected graphical models. `http://www.cs.ubc.ca/~schmidtm/Software`, 2007.

O. Sorkine and M. Alexa. As-Rigid-As-Possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, pages 109–116, 2007.

J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 available from `http://fewcal.kub.nl/sturm`.

TurboSquid. `http://www.turbosquid.com`, 2016.

S. Vicente and L. Agapito. Balloon Shapes: Reconstructing and Deforming Objects with Volume from Images. In *3DV*, June 2013.

D. Warehouse. `https://3dwarehouse.sketchup.com`, 2016.

C. Wu. VisualSFM: A visual structure from motion system. `http://ccwu.me/vsfm`, 2011.

R. Yu, C. Russell, N. D. F. Campbell, and L. Agapito. Direct, Dense, and Deformable: Template-Based Non-rigid 3D

Reconstruction from RGB Video. In *ICCV*, 2015.