

The Isowarp: the Template-based Visual Geometry of Isometric Surfaces

David Casillas-Perez · Daniel Pizarro · David Fuentes-Jimenez · Manuel Mazo · Adrien Bartoli

Received: date / Accepted: date

Abstract Registration maps or *warps* form a key element in Shape-from-Template (SfT). They relate the template with the input image, which contains the projection of the deformed surface. Recently, it was shown that isometric SfT can be solved analytically if the warp and its first-order derivatives are known. In practice, the warp is recovered by interpolating a set of discrete template-to-image point correspondences. This process relies on smoothness priors but ignores the 3D geometry. This may produce errors in the warp and poor reconstructions. In contrast, we propose to create a 3D consistent warp, which technically is a very challenging task, as the 3D shape variables must be eliminated from the isometric SfT equations to find differential constraints for the warp only. Integrating these constraints in warp estimation yields the *isowarp*, a warp 3D consistent with isometric SfT. Experimental results show that incorporating the isowarp in the SfT pipeline allows the analytic solution to outperform non-convex 3D shape refinement methods and the recent DNN-based SfT methods. The isowarp can be properly initialized with convex methods and its hyperparameters can be automatically obtained with cross-validation. The

isowarp is resistant to 3D ambiguities and less computationally expensive than existing 3D shape refinement methods. The isowarp is thus a theoretical and practical breakthrough in SfT.

Keywords Image warp · SfT · Template-based · Non-Rigid Reconstruction

1 Introduction

This paper is about the Shape-from-Template (SfT) problem, also known as template-based non-rigid 3D reconstruction. The object of interest in SfT is a surface that deforms and from which we take a single *image* with a calibrated camera. The known *template* is a reference shape of the surface (one of its possible deformations) and a texture map that describes its appearance. Solving SfT means recovering the deformed surface seen in the image as a one-to-one embedding from the template. Importantly, SfT requires a deformation prior. *Isometry* is a widely applicable prior, which forces geodesic distances to be preserved, and is a good approximation for the deformation undergone by many common materials such as clothing, tissue or paper. Isometry is a differential concept that involves the embedding function and its derivatives. It has been thoroughly studied in SfT [35, 28, 4, 24, 12] whose specific instance we refer to as *isometric SfT*. Early isometric SfT methods [35, 28] use approximations of isometry, namely the inextensibility constraint and the Maximum Depth Heuristic (MDH). The theoretical properties of these models is not fully understood and their results are inaccurate. Very recently, isometric SfT has been described exactly using the differential geometry of surfaces, image projections and isometric deformations, leading to the *differential SfT* methods. The elements

David Casillas-Perez
Departamento de Teoría de la Señal y Comunicaciones, Universidad Rey Juan Carlos, Fuenlabrada, Spain
E-mail: david.casillas@urjc.es

Daniel Pizarro · David Fuentes-Jimenez · Manuel Mazo · David Casillas-Perez
GEINTRA, Universidad de Alcalá, Alcalá de Henares, Spain
E-mail: dani.pizarro@gmail.com;d.fuentes@edu.uah.es;manuel.mazo@uah.es;david.casillas@edu.uah.es

Adrien Bartoli · Daniel Pizarro
Institut Pascal - CNRS, Université Clermont-Auvergne, Clermont-Ferrand, France
E-mail: adrien.bartoli@gmail.com;dani.pizarro@gmail.com

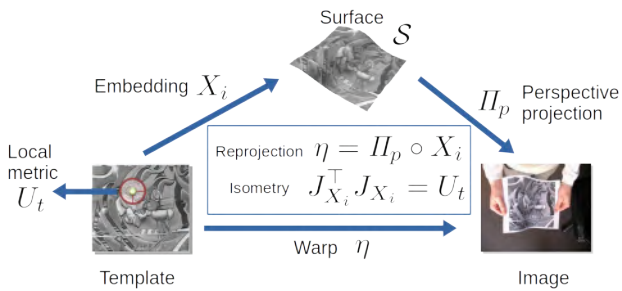


Fig. 1 Differential SfT diagram. Arrows represent locally differentiable functions. The template is represented as a 2D texture map endowed with a local metric U_t , *i.e.* the first fundamental form of the template’s surface. The projection function Π_p represents the perspective camera. The embedding X_i has the same parametrization domain as the template. By following the composition rule, where \circ denotes function composition, the reprojection equation for the warp is given as $\eta = \Pi_p \circ X_i$. Given that J_{X_i} is the Jacobian matrix of X_i , the isometry equation comes from the fact that isometries preserve the local metric and thus $J_{X_i}^\top J_{X_i} = U_t$. The resulting PDE system is called the *isometric SfT system*.

involved in differential SfT (the template domain, deformed surface, image plane) are connected with functions. Their relationship is revealed in a commutative graph shown in Fig. 1. Two important functions contain the unknowns in SfT. *i)* The reconstruction *embedding* X_i , usually described with a scalar *depth function* ρ , embeds the template domain to form the deformed surface and is the sought solution to SfT. *ii)* The *warp* η , which represents the registration between the template and the input image. While X_i is not directly measurable, we can take discrete measurements of η by extracting point correspondences between the template’s texture map and the input image. A recent breakthrough in SfT showed that there exists a direct analytic expression that transforms η and its first order derivatives into X_i [4]. This expression is exact given that η is error free and shows that isometric SfT is well-posed given η . It also means that obtaining η is equivalent to solving isometric SfT. Logically, thus current methods first estimate the warp η and then use the analytic solution for X_i . However, estimating the warp is a difficult regression problem that has been overlooked in the literature. Firstly, the warp is sampled at zeroth-order, by means of point correspondences, which requires interpolation to obtain its derivatives. Secondly, there are no known geometric constraints to make the warp 3D consistent in isometric SfT. This leaves the warp estimation largely disconnected from the 3D geometry, only relying on weaker smoothness priors.

We challenge the current SfT paradigm: we raise the possibility of incorporating 3D geometry directly at warp estimation. This difficult problem has so far

remained open. It raises three main fundamental unanswered questions. *Q1)* What are the equations to guarantee the 3D geometry consistency of a warp? *Q2)* Can we transform these equations in a prior to be used in warp estimation? *Q3)* To which extent this new prior improves the quality of the warp and ultimately of the 3D reconstruction?

This paper brings answers to these questions. Our main contributions are then threefold. *i)* We show that the warp in isometric SfT, hereinafter the *isometric warp* or simply the *isowarp*, is the solution of a non-linear system of second-order PDEs that depend on the template’s known first fundamental form. This represents a breakthrough in our theoretical understanding of SfT that reveals the mathematical properties of the warp *independently of the surface shape*. *ii)* We propose a method to compute the isowarp from point correspondences using unconstrained iterative optimization. We show how to initialize the solution and automatically select the hyperparameters. *iii)* We show that given an isowarp, which is exactly compatible with the SfT equations, the analytic reconstruction is also exact, producing isometric solutions. Beyond the theoretical advances, the isowarp has important practical advantages: *1)* it allows the analytic solution [4] to achieve a higher reconstruction quality than existing nonlinear refinement methods, clearly outperforming warps based on smoothing priors, namely the bending energy [8] or the Schwarzian derivatives [32], see Fig. 2, *2)* it is resistant to 3D ambiguities, *3)* it involves fewer variables and has a fast convergence and *4)* its hyperparameters can be optimally obtained with *cross-validation*, which is not possible with the existing shape refinement methods.

The isowarp was first published for the weak-perspective camera [31]. For this particular camera model it is formulated as a simple linear equation system. In contrast, we find the isowarp for the perspective camera, which is tremendously more challenging owing to its nonlinear nature, and considerably more important in terms of practical use.

We recall that the isowarp is a fundamental contribution, in the sense that it establishes a fundamental new multiple image model in computer vision. While multiple image models are very well understood for rigid objects (they are, in particular, the inter-image homography for a planar scene and the fundamental matrix for a non-planar scene [19]), they are still largely unknown for deformable objects. Some deformable objects can extend while they deform and this creates a complex and largely underconstrained image geometry. However, many objects deform closely to isometries, preserving their geodesics. While modeling this



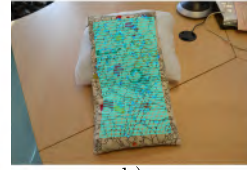

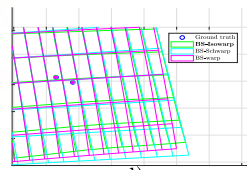
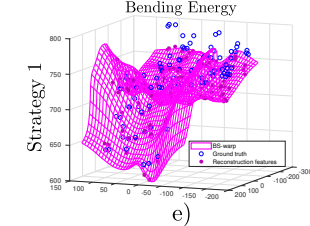
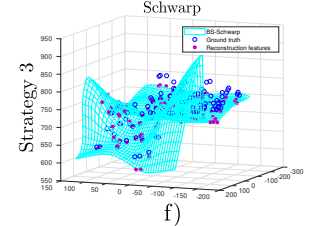
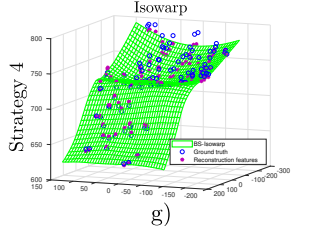
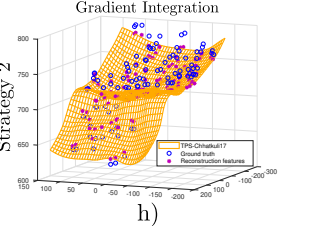
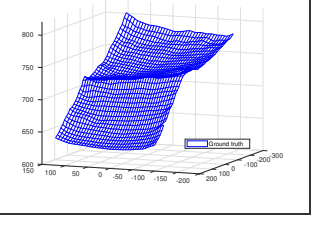
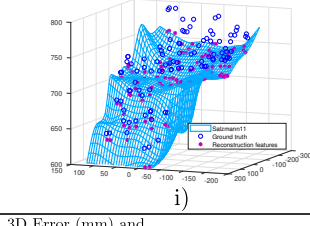
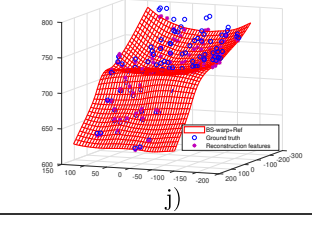
Registration				
Template	Bending Energy	Schwarp	Isowarp	Corner Zoom
				
	a)	b)	c)	d)
2D Error (px):	0.70	0.86	0.93	
Reconstruction				
Analytic Methods				
Direct Depth Methods			Depth Gradient Method	
Bending Energy	Schwarp	Isowarp	Gradient Integration	
				
e)	f)	g)	h)	
3D Error (mm) and Isometry Factor:	12.92 - 78.91%	15.56 - 190.37%	3.43 - 1.29%	9.07 - 4.92%
MDH	Refinement Method	Ground Truth	Isometry Factor	
Salzmann11	3D Refinement		$\left(\frac{1}{ W } \sum_{p \in W} \frac{\ U_t(p) - J_{X_t}^T J_{X_t}(p)\ _F}{\ U_t(p)\ _F} \right)^{1/2}$	
				
i)	j)			
3D Error (mm) and Isometry Factor:	12.85 - 58.57%	4.91-2.08%		

Fig. 2 Results of isometric SfT for different warp models and reconstruction methods on a real example of a cushion. *a)*, *b)* and *c)* are the warps obtained with *BS-warp*, *BS-Schwarp* and *BS-Isowarp* respectively; and *e)*, *f)*, *g)*, *h)*, *i)* and *j)* show the 3D reconstruction results of *BS-warp*, *BS-Schwarp*, *BS-Isowarp*, *TPS-Chhatkuli17*, *Salzmann11* and *BS-warp+Ref* respectively, whose names are defined later in the paper. The reconstruction ground truth is obtained using dense SfM from multiple images [1]. The experiment uses a 3D template and the warps are computed using 100 SURF point correspondences [6]. We give an isometry factor, computed for a set of points W in the template. We give 2D and 3D generalization errors over a set of 1200 correspondences, different from those used in estimation. In registration, we find seemingly very similar results for the different warps. In reconstruction however, our isowarp leads to the best results, outperforming the rest of warp-based methods, the MDH and even 3D refinement. The reason is that, even if the isowarp does not achieve a drastic improvement in registration, it intrinsically enforces geometric constraints, giving structural coherence to the derivatives, directly leading to outperforming 3D reconstruction results without 3D refinement.

property using differential equations has been largely exploited and understood in the recent years by combining a perspective camera and a representation of the 3D surface [4, 12], it is far from trivial to create the mathematical object which establishes the relationship between images independently of the actual 3D surface. This object is the isowarp and is shown to be governed by a differential equation, given for the first time in this paper, as equation (11).

The paper is organized as follows. Section 2 presents the state-of-the-art in SfT. Section 3 describes the geometric model of isometric SfT. Section 4 derives the

isowarp equations. Section 5 gives the isowarp algorithm. Section 6 gives experimental results. Finally, Section 7 concludes.

2 Previous Work

We review the state-of-the-art in SfT with special attention to differential isometric SfT methods and the techniques they use to compute the warp. We also review the contributions of the paper with respect to the existing methods.

2.1 General SfT Methods

The general landscape of SfT methods is pictured in Fig. 3. We first distinguish between *iterative-refinement* and *initialization-free* methods. Iterative-refinement methods are based on non-convex optimization and they require accurate initialization to avoid falling into incorrect local minima. We have refinement methods based on *particles* [26] and those directly optimizing a cost function with local descent methods [18,9,39], such as the popular *Levenberg-Marquardt*. These methods are accurate if they converge to the correct local minimum. Initialization-free methods do not require an initial guess of the solution. In this category we distinguish between *local* and *global* methods. Local methods are based on *differential* [4,12] or *piece-wise* [15] models of the surface, recovering point-wise depth, surface normals, or planes [15]. Usually these methods count with analytic solutions that do not require optimization albeit a last step of surface integration and/or stitching, which is usually a Linear Least-Squares (LLS) problem. Global methods recover the surface in a single step (in this sense all the aforementioned iterative-refinement methods are global). In this category we find differential boundary conditioned problems [11] and methods based on convex optimization, such as the MDH [35,28], methods based on l_1 optimization [23] and methods based on LLS [22].

In terms of deformation priors we distinguish between *isometric* and *non-isometric* methods. The isometric model has been extensively studied [35,28,4,12,9] and counts with methods in all previous categories. Other popular models that are non-isometric are based on conformity [4], Laplacian smoothness [24], linear [22,23] and non-linear elasticity [18,3] and statistical learning, such as the low-rank shape model [2,39]. In isometric SfT, the state-of-the-art strategy is to use local methods, in particular differential SfT solutions [4,12], to initialize iterative-refinement methods [9].

In the recent years, DNN-based SfT methods have appeared [33,17,36,16]. These methods learn the function that maps the input image to the template and to the object depth from a training dataset. They solve both registration and reconstruction in a single step, which is a strong advantage against the classical methods based on a pre-computed registration. However, DNN-based methods require a large amount of data with multiple deformations to be trained and are computationally demanding. Their 3D results over a test set strongly depend on the similarity of deformations used during the training, a disadvantage not present in non-DNN methods. Most DNN-based SfT methods are *object-specific* [33,17,16], which means their networks

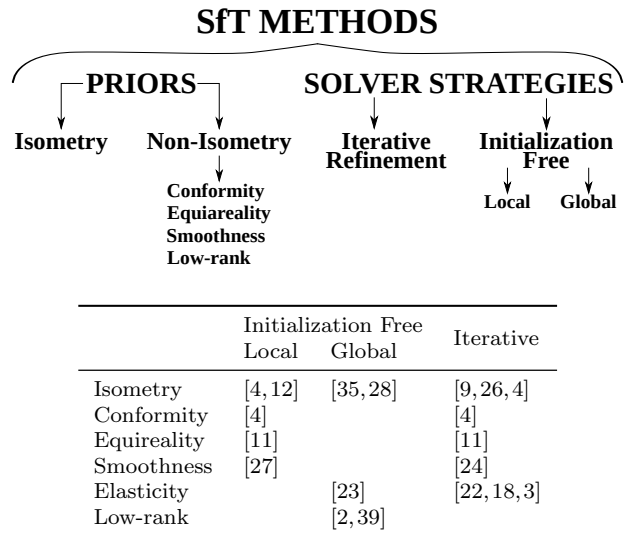


Fig. 3 Classification of SfT methods. The classification is made by the deformation priors and between *initialization-free* and *iterative-refinement* methods.

are trained for reconstructing a specific template (with a specific texture). Hence, they need to be retrained for each new object. In contrast, the *object-generic* DNN-based SfT method [36] works even if the texture of the object was not present in the training dataset. However, object-specific methods are more accurate. DNN-based SfT methods are not scalable in the sense that they have fixed output sizes, both in terms of the output meshes in [33,17,36] and dense maps [16].

2.2 Warp Estimation

Warp functions are defined with different function representations. We find linear basis expansions, such as the thin-plate spline [8], the tensor-product B-spline [34], finite elements [29] or finite differences as in optical flow methods [20]. In SfT [4,12,32], the warp is estimated from point correspondences extracted with either keypoint detection methods [21,6] or optical flow methods [37]. The false correspondences are removed using specific robust deformable registration methods [30,29]. The measurements are thus discrete and scattered in the image, especially in wide-baseline conditions. Fitting a warp function under these conditions requires priors. Existing methods rely on a *smoother*, which is a prior that penalizes high frequencies. The smoother plays an important role as it directly controls how the warp interpolates data and how its derivatives are shaped. The most common smoother is based on penalizing the norm of the second-order derivatives of the warp and is known as the *bending energy* [8,20,40]. The main problem of the bending en-

ergy is that it makes the warp locally affine [8], obliterating local perspective effects which contain crucial information for the reconstruction [32]. In fact, in the limit where the bending energy is exactly enforced the warp is a mere affine function (see the magenta warp in Fig 10.d) for an illustration). Another smoother, which has been recently proposed, is based on the *Schwarzian derivatives* [25], which are projective differential invariants preserved by planes. When used to compute a warp, called the *schwarp*, it helps preserving local perspective effects. However, the schwarp also distorts the local structure of the warp by assuming that the surface is locally flat. When imposed exactly, the schwarp becomes a homographic function.

There also exist rational warp representations that capture the local projective structure. For instance we find the generalized TPS warps [5] and the NURBS warp [10]. They use a 3D form of bending energy expressed in homogeneous coordinates. They tend to be unstable due to their rational structure [32].

None of the previous warp estimation methods includes constraints to make the warp consistent with the 3D geometry. A warp resulting from these methods is likely to produce non-isometric and unstable reconstructions, as exemplified by Fig. 2.

2.3 Differential Isometric SfT Methods

Differential SfT methods study solutions to the *isometric SfT system*, a system of non-linear first-order PDEs [4], see Fig. 1. These equations depend on the unknown reconstruction embedding X_i , uniquely defined with the depth function ρ , and the warp η . These methods find analytic solutions for ρ , and its derivatives $\nabla\rho = (\rho_u, \rho_v)^\top$, in terms of η , η_u and η_v and the template’s known local metric U_t . Using these analytic solutions, there exist several strategies depending on how the warp is computed. Fig. 4 shows the three main existing strategies based on the SfT analytic solutions and the one we propose, strategy 4. We describe them next in detail. They are also mentioned in Fig. 2.

Strategy 1. Direct depth solution from any warp. This strategy was first proposed in [4] along with the first analytic solution and well-posedness properties of isometric SfT. This solution assumes that ρ and its derivatives $\nabla\rho$ are independent algebraic variables in the isometric SfT system, not related via differentiation, leading to the *non-holonomic solution*. This solution is applied in [4] to warps computed from point correspondences with the bending energy as smoother.

Strategy 2. Depth’s gradient integration from any warp. The next step in differential SfT methods was proposed in [12], where the non-holonomic depth’s gradient was used to recover the surface via integration. This strategy is significantly more stable against the amount of perspective and more accurate than strategy 1. Strategies 1 and 2 use the same solution to the isometric SfT system. They differ if the warp has errors. The recovered surface is not exactly isometric to the template. If the warp is error free, the two strategies converge to the same exact solution.

Strategy 3. Direct depth solution from a locally projective warp. In this strategy a schwarp is first obtained to apply the analytic solution. This warp better captures the local perspective effects than the usual warps. This strategy involves non-convex optimization to retrieve the warp. In practice, the improvement of schwarp in 3D reconstruction is however limited and degrades with increasing surface curvature.

Proposed strategy 4. Direct depth solution from the isowarp. We propose a new reconstruction strategy and method in isometric SfT. We derive a set of warp constraints from the 3D isometry equations. We then derive a new method that forces any warp representation, such as the B-spline, to comply with the isowarp equations. We show that using the isowarp followed by the analytic depth solution achieves 3D refinement quality reconstructions. This has many practical advantages in terms of accuracy and computational complexity since the isowarp does not suffer the possible ambiguities arising from the 3D to 2D projection characteristic of the 3D refinement methods and involves less unknown variables.

Final 3D refinement. Strategies 1 and 2 are usually used as initialization in a final 3D refinement step based on non-convex optimization to achieve state-of-the-art reconstruction quality. It appears as a final step in [4, 12], where strategies 1 and 2 are first proposed, for stabilizing the results and achieving competitive reconstruction accuracy. The 3D refinement method converges well to the sought solution from strategies 1 and 2. Indeed, both refined solutions are similar. However, 3D refinement falls into a local minimum with lesser quality initialization and is computationally costly. Besides, the non-convex cost function involves several terms, including the isometry constraint, the reprojection term and a smoother, and combines them using unknown hyperparameters. Obtaining these hyperparameters with cross-validation represents an open problem without the access to 3D ground-truth. This is

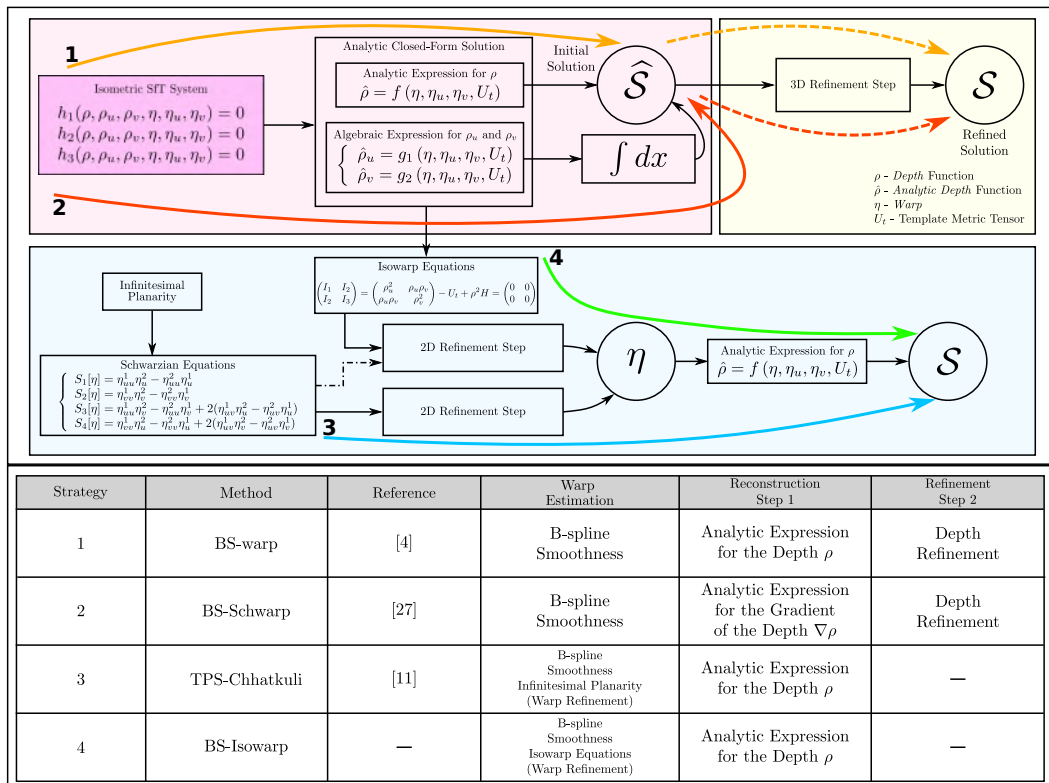


Fig. 4 Differential methods for isometric SfT. We show 4 main strategies represented by numbered arrows in the figure. Arrows 1 and 2 represent the methods [4] and [12] which use the analytic expression for the depth ρ and its derivatives ρ_u and ρ_v respectively. Arrow 3 represents the schwarp method [32], which expands the effort to obtain a quality warp in a 2D refinement step to improve the 3D reconstruction. Arrow 4 represents our isowarp method, which imposes the isowarp equations to obtain a warp fully compatible with isometry.

mainly because the 2D metrics that can be used without ground-truth data, such as the 2D reprojection error, do not completely constrain all the degrees-of-freedom of the surface solution, leading to suboptimal sets of hyperparameters. As a consequence, the hyperparameters in refinement methods cannot be adjusted automatically and must thus be manually tuned or kept fixed to predefined suboptimal values. In contrast, the hyperparameters for the isowarp can be obtained automatically by cross-validation, and we give an algorithm that implements this step.

3 Problem Formulation

3.1 Notation

We use script upper-case characters such as \mathcal{S} to denote regular surfaces in \mathbb{R}^3 and subsets of the usual topology of \mathbb{R}^2 , such as \mathcal{U} which represents the surface parametrization domain. Surface parametrizations are represented by pairs (\mathcal{U}, X_t) , where \mathcal{U} is the domain set of the function and $X_t : \mathcal{U} \rightarrow \mathbb{R}^3$ is the parametrization

function of the surface, represented by an upper-case letter with a single subscript. The subscript signals the associated surface. We use greek upper-case characters for maps between surfaces, *e.g.* the isometric map Ψ , and greek lower-case characters for maps between subsets of \mathbb{R}^2 , *e.g.* the warp η . Mapping composition is represented by the \circ symbol. The operators $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ refer to the Euclidean norm and the dot product, respectively. Along the paper, we consider row vectors of the Euclidean vector space \mathbb{R}^3 and we represent them with tuples as (x, y, z) . Cases where a column vector is required will be indicated through the transpose operator using the superscript $(\cdot)^\top$. Image points are denoted by the coordinate vectors (u, v) . A canonical basis is used in both cases. Operators $\text{tr}()$ and $\text{det}()$ are the trace and the determinant of a square matrix respectively. Operators and functionals are defined with upper-case letters and square brackets. For instance $D[\eta]$ denotes an error metric over the function η . We use $\stackrel{\Delta}{=}$ to denote equality definitions.

3.2 Geometric Setup

The geometric model of isometric SfT is illustrated in Fig. 5. Let the subset $\mathcal{T} \subset \mathbb{R}^3$ be a regular parametric

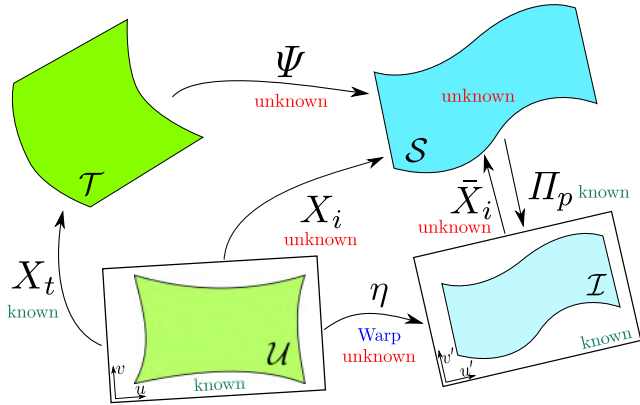


Fig. 5 The diagram shows the differential geometric model we use to study isometric SfT. The known template \mathcal{T} is transformed by an isometric map Ψ into the unknown surface \mathcal{S} , which is then projected with a perspective camera Π_p to create the image \mathcal{I} . In this diagram one can find identities by following different paths that connect the domains. The arrows represent functions and we can thus follow them by function composition (for instance we have $X_i = \bar{X}_i \circ \eta$). See Section 3.2 for details.

surface known as the *template surface* or simply the *template*, and let (\mathcal{U}, X_t) be one of its parametrizations. The template and its parametrization are known. The subset $\mathcal{S} \subset \mathbb{R}^3$ is the regular parametric surface that results from composing the template with an unknown *isometric map* Ψ . We review isometric maps and their properties in Section 3.4.

The observed *image* of surface \mathcal{S} is represented by $\mathcal{I} \subset \mathbb{R}^2$ and captured by a perspective camera. We assume that the restriction of perspective projection map $\Pi_p|_{\mathcal{S}}$ to \mathcal{S} is bijective with image \mathcal{I} . This assumption ensures the surface is not self-occluded and thus that a point is uniquely projected into image \mathcal{I} .

The *warp* is represented by $\eta = (\eta^1, \eta^2)^\top \in C^2(\mathcal{U}, \mathcal{I})$ in Fig. 5. It is a two-diffeomorphism between the open sets $\mathcal{U}, \mathcal{I} \subset \mathbb{R}^2$, the parametric domain of the template and the image respectively. A two-diffeomorphism is a twice continuously differentiable map with differentiable inverse map too. The diffeomorphic character of the warp ensures that the parametrizations (\mathcal{U}, X_t) and (\mathcal{I}, \bar{X}_i) of \mathcal{S} are compatible. We derive constraints that η has to fulfill to become an isowarp in Section 4.

3.3 Camera Model and Image Embedding

We assume that the image plane is at $z = 1$ in camera coordinates, which is achieved by working in standardized retinal coordinates. The perspective projection of a point (x, y, z) is then given by:

$$\Pi_p: \quad \mathbb{R}^3 \setminus \{(x, y, z) \in \mathbb{R}^3 \mid z = 0\} \rightarrow \mathbb{R}^2$$

$$(x, y, z)^\top \mapsto \left(\frac{x}{z}, \frac{y}{z}\right)^\top. \quad (1)$$

The bijective projection assumption permits the unknown surface \mathcal{S} to be parametrized from the image domain using a scalar depth function. It is called the *image embedding* of the surface (\mathcal{I}, \bar{X}_i) . The image domain \mathcal{I} however may be brought to the template domain \mathcal{U} using the warp η , which maps between them in a one to one manner. This way, we construct a surface embedding whose domain is the template's. In other words, this function maps a template point to its corresponding 3D point on the deformed surface \mathcal{S} directly in camera coordinates. This is visualized in Fig. 5 and follows the usual geometric setup of SfT, as described in [4,11] for instance. As a result, the deformed surface \mathcal{S} can be expressed in terms of a depth function $\rho: \mathcal{U} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ as:

$$X_i = \bar{X}_i \circ \eta = \frac{\rho}{\zeta \circ \eta} (\eta^\top, 1)^\top, \quad (2)$$

where $\zeta(u', v') = \sqrt{1 + u'^2 + v'^2}$. Appendix A shows the detailed derivation of this image embedding which was first used in [4].

As will be seen, working with the parametrization (\mathcal{U}, X_i) of \mathcal{S} has two principal advantages. First, it allows us to compute the *first fundamental form*, also known as the *metric tensor*, over the same parametrization domain as the template, which is essential to obtain the isowarp equations. Second, it greatly simplifies the form of these equations.

3.4 Isometric Maps

An isometric map between two surfaces $\Psi: \mathcal{S}_1 \rightarrow \mathcal{S}_2$ is a diffeomorphism that preserves the intrinsic distance (the geodesic distance) and, consequently, the angles and the areas. Formally, isometries are defined in terms of the pushforward function.

Definition 1 (Isometric map) A diffeomorphism $\Psi: \mathcal{S}_1 \rightarrow \mathcal{S}_2$ between surfaces in \mathbb{R}^3 is an isometric map (also called isometry) if for each point $\mathbf{p} \in \mathcal{S}_1$, the pushforward function $d\Psi_{\mathbf{p}}: T_{\mathbf{p}}\mathcal{S}_1 \rightarrow T_{\Psi(\mathbf{p})}\mathcal{S}_2$ of Ψ is a linear isometric function. It means that:

$$\langle \mathbf{w}, \mathbf{h} \rangle = \langle d\Psi_{\mathbf{p}}(\mathbf{w}), d\Psi_{\mathbf{p}}(\mathbf{h}) \rangle, \quad (3)$$

where $\mathbf{w}, \mathbf{h} \in T_{\mathbf{p}}S_1$ are any two vectors of the tangent plane of S_1 at \mathbf{p} . Two regular surfaces \mathcal{S}_1 and \mathcal{S}_2 are isometric to one another if there exists an isometric map between them.

Isometric maps can also be characterized in terms of differential geometry. The next theorem shows that it is possible to find parametrizations of isometric surfaces with common domain that preserve the coefficients of the first fundamental form.

Theorem 1 *A diffeomorphism $\Psi: S_1 \rightarrow S_2$ between two surfaces S_1 and S_2 in \mathbb{R}^3 with parametrizations (\mathcal{U}, X_1) and (\mathcal{U}, X_2) respectively, is an isometric map if and only if it preserves the coefficients of the first fundamental form computed from these parametrizations.*

The proof of the theorem is a well known result that can be found in Do Carmo's book [13]. It is important to emphasize that the coefficients of the first fundamental form computed from both parametrizations (\mathcal{U}_1, X_1) and (\mathcal{U}_2, X_2) of \mathcal{S}_1 and \mathcal{S}_2 respectively, are not necessarily equal even if they are isometric surfaces. A common domain $\mathcal{U}_1 = \mathcal{U}_2 = \mathcal{U}$ is required to apply Theorem 1. This is one of the reasons why we use the parametrization (2) of surface \mathcal{S} .

4 The Isowarp

We review how to obtain the depth and depth's gradient non-holonomic solutions, as was first shown in [4]. We then impose the differential consistency to those solutions by differentiating the non-holonomic depth expression and substituting it back into the SFT equations. This allows us to find the isowarp equations which will characterize the isowarp. As a preliminary, we formally define the isowarp.

Definition 2 (Isowarp) A warp $\eta \in C^2(\mathcal{U}_1, \mathcal{U}_2)$ between the domains $\mathcal{U}_1, \mathcal{U}_2 \subset \mathbb{R}^2$ of two surfaces \mathcal{S}_1 and \mathcal{S}_2 with a generic parametrization (\mathcal{U}_1, X_1) and a perspective parametrization $(\mathcal{U}_2, \bar{X}_2)$ respectively (see Fig. 5), is an *isowarp* if and only if the surfaces are related by an isometric map.

Note that using the identity $X_2 = \bar{X}_2 \circ \eta$ the surface \mathcal{S} can be expressed through the parametrization (\mathcal{U}_1, X_2) (2) which shares the domain with \mathcal{S}_1 . In terms of isometric SFT and according to the definition, the template \mathcal{T} can be defined by any generic parametrization (\mathcal{U}, X_t) , but the deformed surface \mathcal{S} must be projected by the perspective projection (1). Consequently, there exist a parametrization referred from the template domain as $(\mathcal{U}, X_i = \bar{X}_i \circ \eta)$ (2).

4.1 The Non-Holonomic Solution

Theorem 1 ensures that the coefficients of the first fundamental form of two isometric surfaces are equal when they are computed from parametrizations that share a common domain. Fig. 5 shows that \mathcal{U} is the common domain of the parametrizations (\mathcal{U}, X_t) and (\mathcal{U}, X_i) of surfaces \mathcal{T} and \mathcal{S} respectively. Consequently, denoting the first fundamental form of these parametrizations as U_t and U_i , we obtain the following matrix PDE:

$$U_t = \begin{pmatrix} E_t & F_t \\ F_t & G_t \end{pmatrix} = \begin{pmatrix} E_i & F_i \\ F_i & G_i \end{pmatrix} = U_i, \quad (4)$$

where E_t, F_t and G_t are the known coefficients of the first fundamental form of the template \mathcal{T} . The matrix expression (4) represents the isometry equation for a generic parametrization (\mathcal{U}, X_i) of surface \mathcal{S} . Calculating the coefficients $E_i = \langle \frac{\partial X_i}{\partial u}, \frac{\partial X_i}{\partial u} \rangle$, $F_i = \langle \frac{\partial X_i}{\partial u}, \frac{\partial X_i}{\partial v} \rangle$ and $G_i = \langle \frac{\partial X_i}{\partial v}, \frac{\partial X_i}{\partial v} \rangle$ from the perspective parametrization (\mathcal{U}, X_i) (2), and reordering terms we achieve the following system of PDEs:

$$\nabla \rho \nabla \rho^\top + \rho^2 H[\eta] = U_t, \quad (5)$$

where $\nabla \rho$ is the gradient vector of the depth function ρ and $H[\eta]$ is the following first-order differential operator:

$$H[\eta] = \frac{1}{\|\omega\|^2} \left(J_\eta^\top J_\eta - \frac{1}{\|\omega\|^2} (J_\eta^\top \eta \eta^\top J_\eta) \right), \quad (6)$$

where J_η is the full rank Jacobian matrix of the warp η , and $\omega: \mathcal{U} \rightarrow \mathbb{R}^3$ is $\omega^\top = (\eta^\top, 1)$. Note that $H[\eta]$ depends only on the warp function η and its derivatives J_η . Evaluated at any point of the warp domain, $H[\eta](u, v)$ is a symmetric positive-definite matrix. Expression (5) shows the isometry equations customized for the perspective parametrization. The left-hand side of the equation is composed by a sum of two matrices, $\nabla \rho \nabla \rho^\top$ and $\rho^2 H$, whose ranks are 1 and 2 respectively. Following [4], there exists an analytic closed-form expression $\hat{\rho}[\eta]$ for the depth function ρ :

$$\hat{\rho}[\eta] = \sqrt{\frac{\text{tr}(U_t H^{-1}[\eta])}{2} - \sqrt{\frac{(\text{tr}(U_t H^{-1}[\eta]))^2}{4} - \det(U_t H^{-1}[\eta])}}}. \quad (7)$$

Using $\hat{\rho}[\eta]$ in a reconstruction algorithm does not guarantee that the resulting surface is isometric, unless η would be noise-free, which is never the case in practice.

4.2 The Isowarp Equations

Equation (7) constitutes the non-holonomic depth solution. Observe that $\hat{\rho}[\eta]$ only depends on the warp η via $H[\eta]$ and the coefficients of the first fundamental form U_t of the template. The non-holonomic depth $\hat{\rho}[\eta]$ is a differentiable operator and we can thus compute its derivatives:

$$\hat{\rho}_u[\eta] \triangleq \frac{\partial \hat{\rho}[\eta]}{\partial u}, \quad \hat{\rho}_v[\eta] \triangleq \frac{\partial \hat{\rho}[\eta]}{\partial v}, \quad (8)$$

where the new differential operators $\hat{\rho}_u[\eta]$, $\hat{\rho}_v[\eta]$ depend on the second-order derivatives of η .

We rewrite equation (5) to find the following PDE system:

$$\nabla_{\rho} \nabla_{\rho}^{\top} = \begin{pmatrix} \rho_u^2 & \rho_u \rho_v \\ \rho_u \rho_v & \rho_v^2 \end{pmatrix} = U_t - \rho^2 H[\eta]. \quad (9)$$

Substituting equations (7) and (8) with $\hat{\rho}[\eta]$ and its derivatives $\hat{\rho}_u[\eta]$, $\hat{\rho}_v[\eta]$ in equation (9) we obtain the isowarp equations. They are three PDEs that relate the warp η and its partial derivatives up to second-order η_u , η_v , η_{uu} , η_{uv} , η_{vv} . Moving all terms in equation (9) to the left-hand side we obtain their matrix form:

$$\begin{pmatrix} \hat{\rho}_u^2[\eta] & \hat{\rho}_u[\eta]\hat{\rho}_v[\eta] \\ \hat{\rho}_u[\eta]\hat{\rho}_v[\eta] & \hat{\rho}_v^2[\eta] \end{pmatrix} - U_t + \hat{\rho}^2[\eta]H[\eta] = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \quad (10)$$

As a result and due to symmetry we obtain the following PDE system:

$$\begin{cases} I_1[\eta] = \hat{\rho}_u^2[\eta] - E_t + \hat{\rho}^2[\eta]H^{11}[\eta] = 0 \\ I_2[\eta] = \hat{\rho}_u[\eta]\hat{\rho}_v[\eta] - F_t + \hat{\rho}^2[\eta]H^{12}[\eta] = 0 \\ I_3[\eta] = \hat{\rho}_v^2[\eta] - G_t + \hat{\rho}^2[\eta]H^{22}[\eta] = 0, \end{cases} \quad (11)$$

where $H^{ij}[\eta]$ denotes the (i, j) th entry of H . The PDE system (11) represents the *isowarp equations*, a set of 3 non-linear second-order PDEs $I_1[\eta]$, $I_2[\eta]$ and $I_3[\eta]$ in 12 variables: $(\eta^1, \eta_u^1, \eta_v^1, \eta_{uu}^1, \eta_{uv}^1, \eta_{vv}^1)$ and $(\eta^2, \eta_u^2, \eta_v^2, \eta_{uu}^2, \eta_{uv}^2, \eta_{vv}^2)$. Appendix B gives code to derive the isowarp equations using Matlab's symbolic engine. The following theorem formally characterizes the isowarp from system (11).

Theorem 2 *A warp function $\eta \in C^2(\mathcal{U}_1, \mathcal{U}_2)$ between the domains $\mathcal{U}_1, \mathcal{U}_2 \subset \mathbb{R}^2$ of two surfaces \mathcal{S}_1 and \mathcal{S}_2 with a generic parametrization (\mathcal{U}_1, X_1) and a perspective parametrization (\mathcal{U}_2, X_2) (2) respectively, is an isowarp if and only if it satisfies the system of isowarp equations (11).*

Proof It is a consequence of the isowarp Definition 2 and of the development followed to derive the isowarp equations in Sections 4.1 and 4.2.

Importantly, the isowarp equations only have η as free variable: the other variables are known from the template, as we managed to eliminate the unknown 3D surface. Obviously, the system of isowarp equations (11) does not uniquely determine the isowarp. It represents constraints the warp must verify to be an isowarp. In other words, the system (11) characterizes the set of image warps which feasibly represent the projection of any isometric deformation of a known surface. The isowarp algorithm described next shows how to combine these equations to a data term to actually estimate a warp.

5 Isowarp Algorithm

We propose a method to compute the isowarp from a set of N point correspondences $C = \{(p_i, q_i) \in \mathcal{U} \times \mathcal{I} \mid 1 \leq i \leq N\}$ between the template's domain defining \mathcal{U} and an input image \mathcal{I} . It is based on a 2D refinement process which enforces the isowarp equations. An iterative estimation algorithm is proposed to automatically select the hyperparameters by cross-validation.

Originally, the isowarp algorithm assumes that no self-occlusion occurs, hence, that a diffeomorphic map exists between the image and the sought reconstruction. However, this is not a hard assumption, as we can still use the isowarp strategy when the warp is piecewise diffeomorphic, excluding self-occluded areas from the reconstruction. The method developed in [30] which estimates piecewise smooth warps by deactivating the smoothing terms in self-occluded areas, can be used with the isowarps. The method also removes the possible mismatches.

5.1 Overview of the BS-Isowarp Algorithm

Figure 6 shows an overview of the complete BS-Isowarp Algorithm. We estimate the isowarp following 3 steps using a B-spline representation, leading to the BS-Isowarp. We first compute a BS-warp as initialization, as this warp has a convex solution, for which we give a full-fledged algorithm including automatic hyperparameter selection in Section 5.2, along with a short description of the original variational problem. Interestingly, this BS-warp algorithm serves as a gentle introduction to the BS-Isowarp algorithm, whose layout we give in Algorithm 1. This algorithm first calls the BS-warp algorithm, then selects the BS-Isowarp hyperparameters and finalizes the isowarp. These last two steps depend on two essential algorithms, *BS-Isowarp-Hyperparameters* and *BS-Isowarp-Base*. The BS-Isowarp-Base algorithm is given first, in Section 5.3,

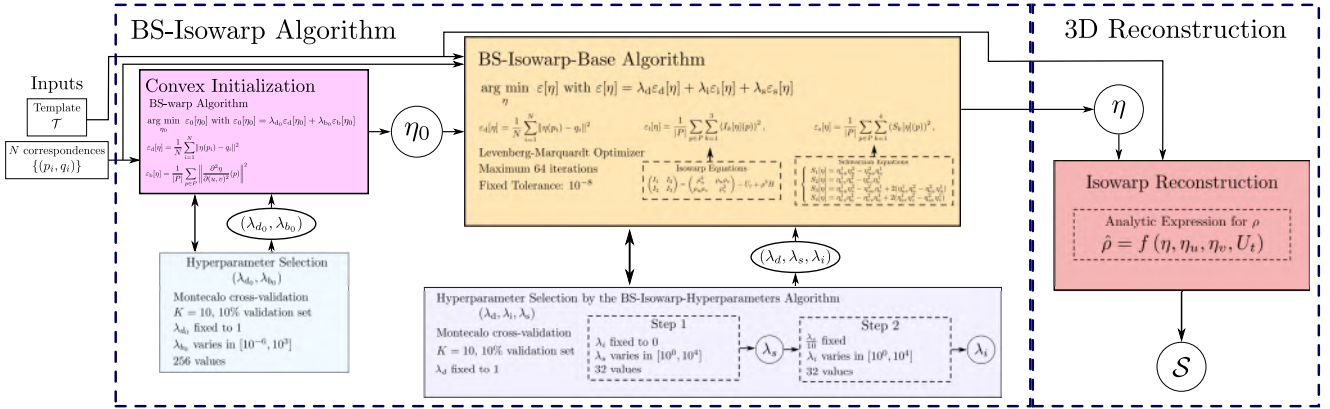


Fig. 6 Overview of the BS-Isowarp Algorithm. Dotted squares separate the isowarp estimation steps from the reconstruction block. The BS-Isowarp Algorithm has three main steps. Firstly, a convex problem is solved to retrieve the BS-warp η_0 (magenta box), which includes automatic hyperparameter selection (blue box), see Section 5.2. The BS-warp serves as an initialization to the BS-Isowarp-Base algorithm (yellow box), which establishes a variational problem that enforces the isowarp equations, see Section 5.3. This Algorithm repeatedly calls the BS-Isowarp-Hyperparameters Algorithm (grey box), which implements a cross-validation mechanism to select the hyperparameters, see Section 5.4.

along with the variational problem. Specifically, this algorithm estimates the BS-Isowarp given the hyperparameters. The BS-Isowarp-Hyperparameters algorithm is then given in Section 5.4, as it depends on the BS-Isowarp-Base algorithm. It implements a cross-validation mechanism to select the hyperparameters.

Algorithm 1: BS-Isowarp

Input : Template \mathcal{T} , image \mathcal{I} , point correspondences C .

Output: Control points w of the B-spline.

- 1 Initialization: $w_0 = \text{BS-warp}(\mathcal{T}, \mathcal{I}, C)$.
 - 2 Hyperparameters: $(\lambda_d, \lambda_i, \lambda_s) = \text{BS-Isowarp-Hyperparameters}(\mathcal{T}, \mathcal{I}, C, w_0)$.
 - 3 Finalization: $w = \text{BS-Isowarp-Base}(\mathcal{T}, \mathcal{I}, C, w_0, \lambda_d, \lambda_i, \lambda_s)$.
-

5.2 Convex Initialization by the BS-warp Algorithm

We obtain the initial estimate η_0 by solving the following classical variational problem:

$$\arg \min_{\eta_0} \varepsilon_0[\eta_0] \text{ with } \varepsilon_0[\eta_0] = \lambda_{d_0} \varepsilon_d[\eta_0] + \lambda_{b_0} \varepsilon_b[\eta_0]. \quad (12)$$

The functional $\varepsilon_d[\eta]$ is the data term, given by the mean square transfer error of the point correspondences:

$$\varepsilon_d[\eta] = \frac{1}{N} \sum_{i=1}^N \|\eta(p_i) - q_i\|^2. \quad (13)$$

The functional $\varepsilon_b[\eta]$ is the bending energy which penalizes the warp second-order derivatives, given by:

$$\varepsilon_b[\eta] = \frac{1}{|P|} \sum_{p \in P} \left\| \frac{\partial^2 \eta}{\partial(u, v)^2}(p) \right\|^2. \quad (14)$$

Given the hyperparameters $(\lambda_{d_0}, \lambda_{b_0})$, the estimation problem (12) is convex and solved with LLS. We represent η using a B-spline model. This converts the variational problem (12) into a linear least-squares problem depending on a set w_0 of control points. Optimal values for the hyperparameters λ_{d_0} and λ_{b_0} are chosen by applying *Monte Carlo cross-validation* [14]. We fix the number of splits to $K = 10$ and choose 90% of random correspondences to create each warp proposal. The remaining 10% correspondences are used as validation set. As validation metric we measure the mean square transfer error over the validation set. We vary λ_{b_0} in the range $[10^{-3}, 10^6]$, which we determined experimentally, and fix λ_{d_0} to 1 as only their ratio matters. Thanks to the convexity of problem (12), we can evaluate a large number of values $n_{b_0} = 256$. The BS-warp algorithm is given in Algorithm 2.

5.3 Variational Problem and Base Solution by the BS-Isowarp-Base Algorithm

The isowarp algorithm relies on the following variational problem:

$$\arg \min_{\eta} \varepsilon[\eta] \text{ with } \varepsilon[\eta] = \lambda_d \varepsilon_d[\eta] + \lambda_i \varepsilon_i[\eta] + \lambda_s \varepsilon_s[\eta]. \quad (15)$$

Compared with the classical formulation (12), it has an extra functional $\varepsilon_i[\eta]$. This functional models the

Algorithm 2: BS-warp

Input : Template \mathcal{T} , image \mathcal{I} , point correspondences C .

Output: Control points w_0 of the B-spline.

- 1 Hyperparameters: Set $\lambda_{d_0} = 1$. Use Monte Carlo cross-validation to infer λ_{b_0} in the range $[10^{-3}, 10^6]$ with $n_{b_0} = 256$. Solve problem (12) by LLS for each value of λ_{b_0} and keep the one which minimizes the validation metric.
- 2 Finalization: Solve (12) by LLS with all correspondences C .

Algorithm 3: BS-Isowarp-Base Algorithm

Input : Template \mathcal{T} , image \mathcal{I} , point correspondences C , initial control points w_0 , hyperparameters $(\lambda_d, \lambda_i, \lambda_s)$.

Output: Control points w of the B-spline.

- 1 Optimization: Run the Levenberg-Marquardt algorithm to minimize $\varepsilon[\eta]$ (15) with respect to the control points w starting from w_0 .

isowarp constraint measured at a dense regular grid of points $P \subset \mathcal{U}$ with $|P|$ points in the template domain:

$$\varepsilon_i[\eta] = \frac{1}{|P|} \sum_{p \in P} \sum_{k=1}^3 (I_k[\eta](p))^2, \quad (16)$$

where $I_k[\eta]$ is an isowarp equation (11). While $\varepsilon_i[\eta]$ imposes that the warp is an isowarp, it does not impose it to be smooth. This makes sense as the isometric surface deformation it captures is not necessarily smooth. We thus use a smoother $\varepsilon_s[\eta]$, which we choose as the schwarp [32],

$$\varepsilon_s[\eta] = \frac{1}{|P|} \sum_{p \in P} \sum_{k=1}^4 (S_k[\eta](p))^2, \quad (17)$$

where the Schwarzian equations [32] are:

$$\begin{cases} S_1[\eta] = \eta_{uu}^1 \eta_u^2 - \eta_{uu}^2 \eta_u^1 \\ S_2[\eta] = \eta_{vv}^1 \eta_v^2 - \eta_{vv}^2 \eta_v^1 \\ S_3[\eta] = \eta_{uv}^1 \eta_v^2 - \eta_{uv}^2 \eta_u^1 + 2(\eta_{uv}^1 \eta_u^2 - \eta_{uv}^2 \eta_u^1) \\ S_4[\eta] = \eta_{vv}^1 \eta_u^2 - \eta_{vv}^2 \eta_u^1 + 2(\eta_{uv}^1 \eta_v^2 - \eta_{uv}^2 \eta_v^1) \end{cases} \quad (18)$$

The hyperparameters λ_d , λ_i and λ_s are scalars that control the importance of each functional.

Assuming known hyperparameters, the BS-Isowarp-Base algorithm estimates η by solving the variational problem (15) given an initial estimation η_0 . Similarly to the BS-warp algorithm, we use a B-spline representation and estimate its control points denoted w . The problem is however non-convex. We use the Levenberg-Marquardt which finishes if a minimum is reached with a fixed tolerance of 10^{-8} or a maximum number of iterations $n_f = 64$ has passed. The BS-Isowarp-Base algorithm is given in Algorithm 3.

5.4 Hyperparameter Selection by the BS-Isowarp-Hyperparameters Algorithm

The estimation of the optimal hyperparameters $(\lambda_d, \lambda_i, \lambda_s)$ also relies on Monte Carlo cross-validation

in this case, with the same data split and validation metric as for the BS-warp. Obviously, only 2 out of 3 are independent hyperparameters and we thus fix λ_d to 1. However, for obvious computational efficiency reasons, we cannot try all possible combinations for the two remaining free hyperparameters. We thus estimate each one in turn, keeping the other one fixed. We propose a heuristic but very effective way to conduct this two-round process. We start with λ_s and end with λ_i . We first vary λ_s in the range $[10^0, 10^4]$, while nullifying λ_i and call the base warp algorithm for each value. We fix the number of values to $n_s = 32$. We then vary λ_i in the range $[10^0, 10^4]$ sampled by $n_i = 32$ values while fixing λ_s to 10% of the obtained value. This two-round process may lead to a suboptimal local minimum. Intuitively, since the Schwarzian equations (18) tend to oversmooth warps, reducing or even vanishing their second-order derivatives, they artificially diminish the value of the isowarp functional. Our strategy is to decrease the hyperparameter λ_s to a fixed ratio with respect to its initial optimum. We choose 10% as a heuristic ratio with very good results in practice. Remark that the Schwarzian derivatives (18) [32] form a smoother, not a constraint which should be fulfilled. We provide an experimental investigation on hyperparameter sensitivity in Section 6.2.3 and justify the chosen ranges numerically. We give our BS-Isowarp-Hyperparameters algorithm in Algorithm 4.

5.5 The Complete Isowarp Reconstruction Algorithm

Surface reconstruction is achieved in a final step by evaluating the non-holonomic depth equation (7) using the obtained isowarp and the template. The importance of this algorithm is that it is able to automatically adjust its hyperparameters. We show in the experiments of Section 6 that this strategy is consistently better than 3D refinement methods: the strategy reaches 3D refinement quality reconstruction but involves fewer variables and does not suffer from ambiguities derived from the 3D to 2D projection that can affect the convergence of iterative methods. The reprojection error may not be

Algorithm 4: BS-Isowarp-Hyperparameters Algorithm

Input : Template \mathcal{T} , image \mathcal{I} , point correspondences C , initial control points w_0 .

Output: Optimal hyperparameters $(\lambda_d, \lambda_i, \lambda_s)$.

- 1 Hyperparameter λ_s : Set $\lambda_d = 1$ and $\lambda_i = 0$. Use Monte Carlo cross-validation to infer λ_s in the range $[10^0, 10^4]$ with $n_s = 32$. Solve problem (15) by BS-Isowarp-Base($\mathcal{T}, \mathcal{I}, C, w_0, 1, 0, \lambda_s$) for each value of λ_s and keep the one which minimizes the validation metric.
 - 2 Hyperparameter λ_i : Set $\lambda_d = 1$ and $\lambda_s = \frac{\lambda_s}{10}$. Use Monte Carlo cross-validation to infer λ_i in the range $[10^0, 10^4]$ with $n_i = 32$. Solve the problem (15) by BS-Isowarp-Base($\mathcal{T}, \mathcal{I}, C, w_0, 1, \lambda_i, \lambda_s$) for each value of λ_i and keep the one which minimizes the validation metric.
-

used to select the hyperparameters related to the depth, as explained in Section 2.3.

In other words, cross-validation cannot be implemented with 3D refinement methods to select hyperparameters as there is no sensible validation metric independent of the ground-truth. In practice the hyperparameters of 3D refinement methods are thus chosen manually.

6 Experimental Results

We present experimental results to compare the isowarp against state-of-the-art in registration and 3D reconstruction.

6.1 Compared Methods and Error Metrics

We use *BS* or *TPS* at the beginning of each method’s name to indicate whether the warp and depth functions are represented using the B-spline or the Thin-Plate Spline. In terms of registration we compare our method, called *BS-Isowarp*, with two different state-of-the-art warp estimation methods: *BS-warp* [34] and *BS-Schwarp* [32]. Table 1 summarizes the main characteristics of all evaluated warp estimation methods. We

Method	Constraint	Smoother	Convex	Initialization
BS-warp	-	Bending energy	Yes	N/A
BS-Schwarp	-	Schwarzian Eqs. (18)	No	BS-warp
BS-Isowarp	Isowarp Eqs. (11)	Schwarzian Eqs. (18)	No	BS-warp

Table 1 Characteristics of the evaluated warp methods.

provide the 2D error metric $\mathcal{E}_{\text{gen}}^{\text{warp}}$ (19), computing the transfer error over a set of correspondences Q , the test

set, which is different from the set of correspondences used to estimate the warp $\tilde{\eta}$:

$$\mathcal{E}_{\text{gen}}^{\text{warp}}[\tilde{\eta}] = \sqrt{\frac{1}{|Q|} \sum_{(p,q) \in Q} \|\tilde{\eta}(p) - q\|^2}. \quad (19)$$

The 2D error metric (19) allows us to evaluate overfitting and warp quality. In synthetic experiments, where ground truth is available, we also provide 2D error measures of first (20) and second-order (21) derivatives with respect to the ground truth warp η . We use a dense set of points W chosen to cover the template domain to compute the relative metrics:

$$\mathcal{E}_{\text{gen}'}^{\text{warp}}[\eta, \tilde{\eta}] = \sqrt{\frac{1}{|W|} \sum_{p \in W} \frac{D[\eta - \tilde{\eta}](p)}{D[\eta](p)}} \quad (20)$$

$$\mathcal{E}_{\text{gen}''}^{\text{warp}}[\eta, \tilde{\eta}] = \sqrt{\frac{1}{|W|} \sum_{p \in W} \frac{D^2[\eta - \tilde{\eta}](p)}{D^2[\eta](p)}}, \quad (21)$$

where:

$$D[\eta](p) = \left\| \frac{\partial \eta}{\partial(u, v)}(p) \right\|^2 \quad (22)$$

$$D^2[\eta](p) = \left\| \frac{\partial^2 \eta}{\partial(u, v)^2}(p) \right\|^2. \quad (23)$$

We also evaluate the influence of warp quality in 3D reconstruction. We compare the four different reconstruction strategies described in Fig. 4. For simplicity, we keep the same acronyms used in warp estimation for the reconstruction methods based on the direct depth solution (7). Otherwise a different name is given to the reconstruction method. We use *BS-warp* and *BS-warp+Ref* for methods using strategy 1. The latter uses non-linear shape refinement [9] initialized with the former. *TPS-Chhatkuli17* uses the original implementation [12] of strategy 2 based on TPS warps. We omit refinement in strategy 2 as it consistently gives very similar reconstruction results to *BS-warp+Ref* in all our experiments. *BS-Schwarp* represents strategy 3 and *BS-Isowarp* the proposed strategy 4. Besides, we compare the four strategies with the MDH method [35], named *Salzmann11*. The original method [35] reconstructs the given correspondences. We added a final step of surface fitting with a TPS in order to compute 3D error metric. Table 2 summarizes the principal properties of all the evaluated isometric SFT methods.

Method	Warp Method	Strategy	3D Ref.	Init. by
BS-warp	BS-warp	1	No	N/A
BS-warp+Ref	BS-warp	1	Yes	BS-warp
TPS-Chhatkuli17	TPS-warp	2	No	N/A
BS-Schwarz	BS-Schwarz	3	No	N/A
BS-Isowarp	BS-Isowarp	4	No	N/A
Salzmann11	Correspondences	-	No	N/A

Table 2 Characteristic of the evaluated isometric SfT methods.

We provide the 3D error metric $\mathcal{E}_{\text{gen}}^{\text{3D}}$ (24) of the reconstruction for all evaluated methods:

$$\mathcal{E}_{\text{gen}}^{\text{3D}}[\tilde{X}] = \sqrt{\frac{1}{|R|} \sum_{(p,Q) \in R} \|\tilde{X}(p) - Q\|^2}, \quad (24)$$

where R is a set of pairs of points with p in the template domain and Q its corresponding ground truth 3D point, different from the ones used to reconstruct.

We use Monte Carlo cross-validation to choose the best hyperparameter in the *BS-warp*, *BS-Schwarz* and *BS-Isowarp* methods as explained in Section 5. The hyperparameters of *TPS-Chhatkuli17*, *Salzmann11* and *BS-warp+Ref* are manually adjusted by varying them in intervals around the values recommended by the authors in their published codes, using the average reconstruction error as a metric to find a suitable value.

6.2 Synthetic Data Experiments

We test the registration and reconstruction methods using synthetically generated surfaces and their image projections. We describe next the simulation conditions and the experimental setup. Besides, we present the results of the *BS-Isowarp* in a hyperrealistic synthetic experiment with a shoe, in Section 6.2.6.

6.2.1 Simulation Setup

We generate a set of 100 images from rigid and deformable surfaces. We simulate an A4 paper sheet as a planar template. In the rigid case, we rotate and translate the template to emulate rigid transformations. In the deformable case, the template is isometrically transformed into part of a cylinder and is then rotated and translated. We emulate a pin-hole camera with a resolution of 640×480 pixels and a focal length of 500 pixels. We randomly generate 100 point correspondences between the images for warp estimation. The test set we use to compute the error metrics from equations (19), (20), (21) and (24) consists of 1500 randomly generated correspondences. We fix the number of B-spline control points to 12. We add noise, independently to each image coordinate, with a Gaussian distribution

with a 5 pixel standard deviation. We measure an average error by repeating each experimental condition 100 times.

6.2.2 Examples

We show registration and reconstruction results for a rigid and a deformable, synthetically generated examples. Fig. 7 illustrates the planar template, the point correspondences and the images for both experiments. Fig. 8 and 9 show the registration and reconstruction

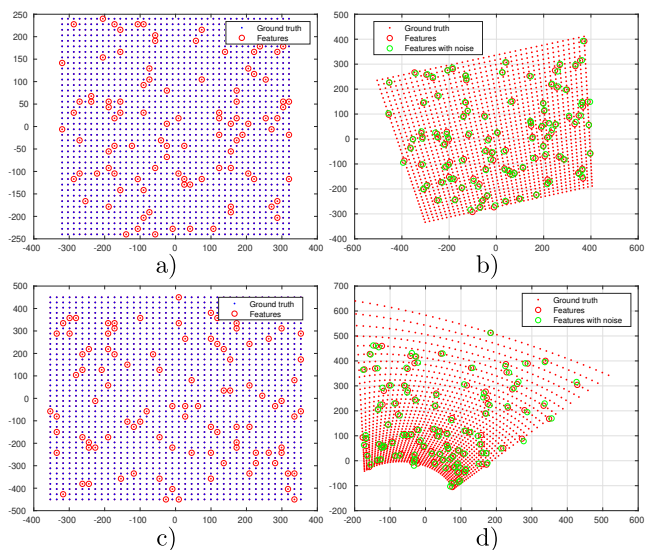


Fig. 7 Experiments with synthetic data. *a)* and *b)* show a planar template transformed rigidly. *c)* and *d)* show the deformable case where the template is isometrically transformed into a cylinder. The red circles show the ground-truth correspondences and the green circles show the noisy correspondences.

results we obtain for the rigid case. We observe that the quality of all warps is very high in the rigid case. Table 3 gives the 2D error and the first and second-order derivative errors. There are no significant differences between the methods in 2D error and first-order derivative error. However we can notice that *BS-Isowarp* overcomes *BS-Schwarz* and *BS-warp* in second-order derivative error. Observe that *BS-warp* produces a non-uniform error

Method	2D Error	1st Derivative	2nd Derivative
BS-warp	2.80	1.42	26.85
BS-Schwarz	2.58	1.09	14.15
BS-Isowarp	2.10	0.81	8.71

Table 3 2D error and relative first and second-order derivative error for all evaluated warp methods in the rigid case.

pattern in the warp in contrast to *BS-Schwarz* and *BS-Isowarp*. In terms of reconstruction Table 4 gives the

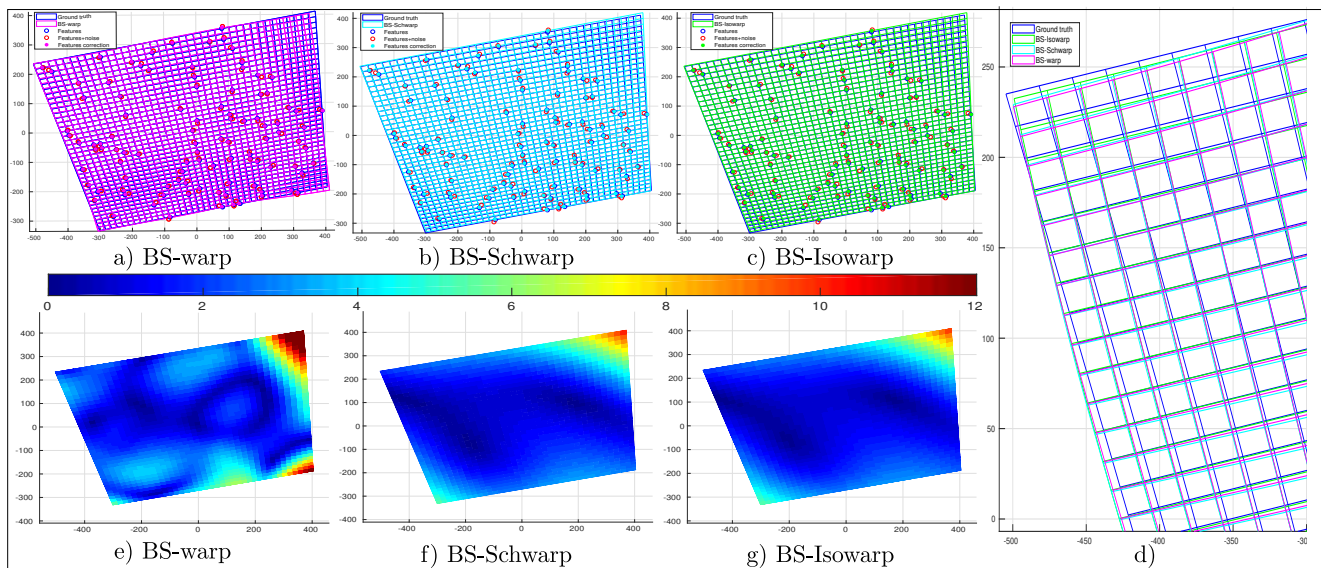


Fig. 8 Registration results in the rigid case. *a)*, *b)* and *c)* show the warps obtained with *BS-warp*, *BS-Schwarp* and *BS-Isowarp* respectively. *d)* shows a comparison of the behaviour of each method in the top left hand corner. *e)*, *f)* and *g)* represent a heat map for each warp that shows the 2D error in pixels at each point, see Section 6.2.2 for more details.

3D error for this rigid example. *BS-Isowarp* obtains the best 3D reconstructions in the rigid case closely followed by *BS-Schwarp*. *BS-warp* does not achieve an accurate 3D reconstruction and produces high frequency reconstruction errors. *TPS-Chhatkuli17* does not perform as well as expected in this case. The multiple hyperparameters to choose explain this result. *BS-warp+Ref* achieves accurate reconstruction, though inferior to *BS-Isowarp*. *Salzmann11* obtains the worst reconstruction due to its sensitivity to noise in the correspondences.

Method	Rigid case	Deformable case
BS-warp	5.13	29.94
BS-Schwarp	2.38	37.99
BS-Isowarp	2.25	4.32
TPS-Chhatkuli17	6.10	51.87
BS-warp+Ref	4.37	6.89
Salzmann11	39.62	84.66

Table 4 3D error for all evaluated methods in the rigid and deformable cases. The errors are given in millimeters.

In the deformable case, Fig. 10 and 11 illustrate the registration and reconstruction results respectively. *BS-Isowarp* produces better registration results than *BS-warp* and *BS-Schwarp*. This is clearly visible in the upper right corner where the surface curvature is pronounced (see Fig. 10. *d)*). Table 5 gives the 2D error and the first and second-order derivative errors for this experiment. Fig. 11 shows the reconstruction results of all evaluated methods. *BS-Isowarp* achieves the best reconstruction results followed by *BS-warp+Ref*. The

Method	2D Error	1st Derivative	2nd Derivative
BS-warp	4.71	4.24	22.03
BS-Schwarp	4.30	3.83	25.68
BS-Isowarp	2.57	1.90	14.21

Table 5 2D error and relative first and second-order derivative error for all evaluated warp methods in the deformable case.

other reconstruction methods behave similarly and are not able to produce accurate reconstructions.

The performance of the evaluated reconstruction methods in our synthetic experiments may at first sight look worse than what the original paper reported [12]. However, bear in mind that our experiments use few point correspondences with high noise to test the interpolation capabilities of the different warp methods. In our experiments smooth registration warps fail to accurately recover the first-order derivatives, which leads to poor results in the *BS-warp* and *TPS-Chhatkuli17* methods.

6.2.3 Hyperparameter Sensitivity

We measure the sensitivity of the hyperparameters used in *BS-warp*, *BS-Schwarp* and *BS-Isowarp*. Table 6 describes the main hyperparameter for each evaluated method. We use the synthetic deformation from the previous section. For each warp estimation method, *BS-warp*, *BS-Isowarp* and *BS-Schwarp*, we vary the hyperparameter in the range $[10^0, 10^4]$ for *BS-Isowarp* and *BS-Schwarp* and $[10^{-3}, 10^6]$ for *BS-warp*. For each hyperparameter value we compute the warp. The first row

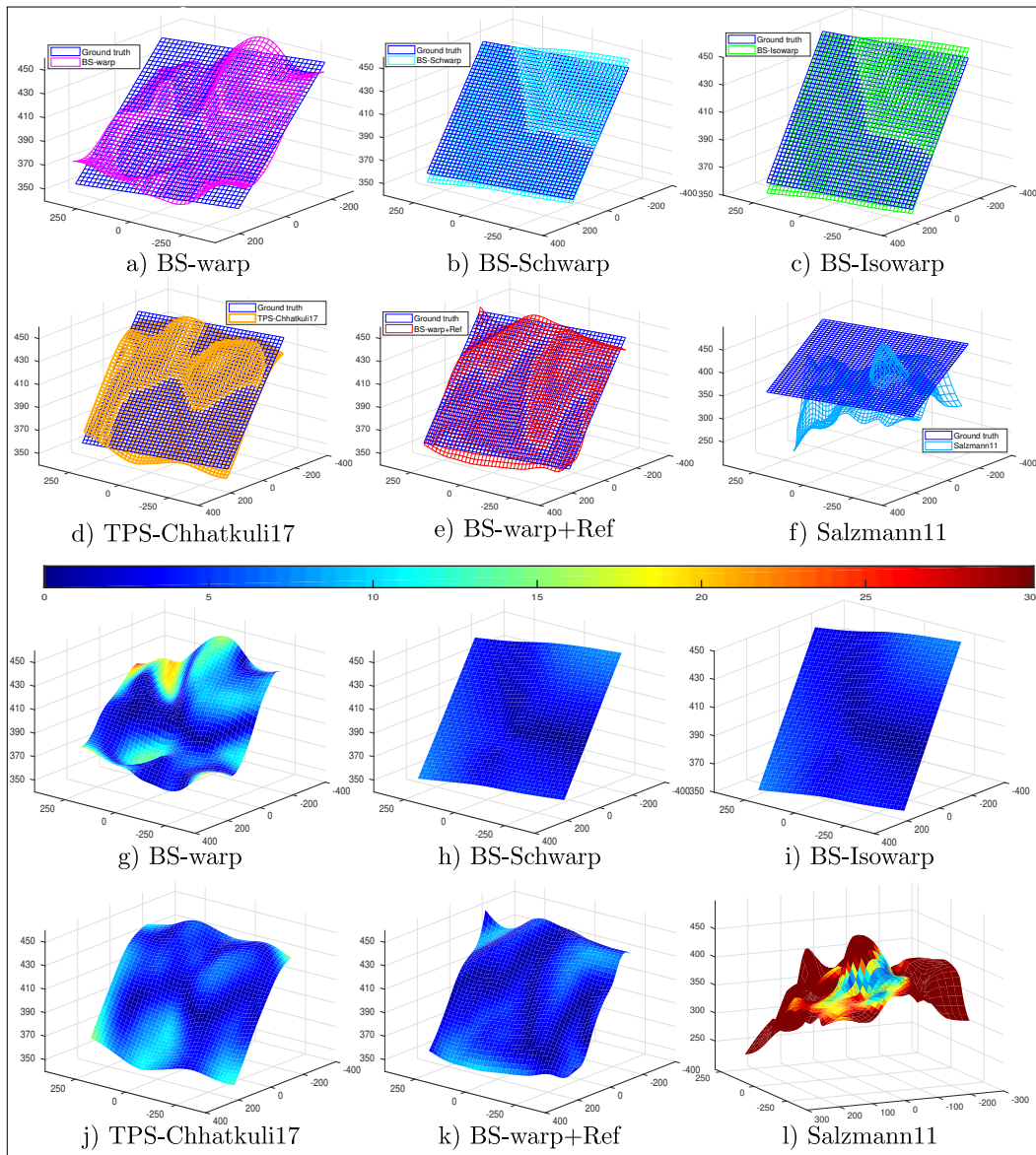


Fig. 9 Reconstruction results in the rigid case. *a)*, *b)*, *c)*, *d)*, *e)* and *f)* show the 3D reconstruction results of *BS-warp*, *BS-Schwarp*, *BS-Isowarp*, *TPS-Chhatkuli17*, *BS-warp+Ref* and *Salzmann11* respectively. *g)*, *h)*, *i)*, *j)*, *k)* and *l)* show a heat map for each warp that shows the 3D error at each point, see Section 6.2.2 for more details.

Method	Penalizer	Symbol
BS-warp	Bending energy	λ_b
BS-Schwarp	Schwarzian equations	λ_s
BS-Isowarp	Isowarp equations	λ_i

Table 6 Principal hyperparameter for each evaluated warp.

of Fig. 12 shows the 2D error in function of the hyperparameter for all methods. The second row of Fig. 12 shows the same experiment by overlaying the results with different noise standard deviations in the interval from 1 to 5 pixels. We observe that all the evaluated methods present a large basin, of at least two orders of magnitude the hyperparameter value, where the 2D

error measurements are similar. This means that hyperparameter selection is feasible and meaningful.

6.2.4 The Effect of Noise

We evaluate the behavior of the different algorithms against measurement noise. We vary the noise standard deviation linearly between 0 and 5 pixels in 10 steps.

Fig. 13.a) shows the results for the rigid case. *BS-Schwarp* achieves the best performance in both 2D error and first and second-order derivative error, followed by *BS-Isowarp*. Note that in this case, the surface is planar and thus the *BS-Schwarp* is an exact model. However, in terms of reconstruction error *BS-Isowarp* obtains

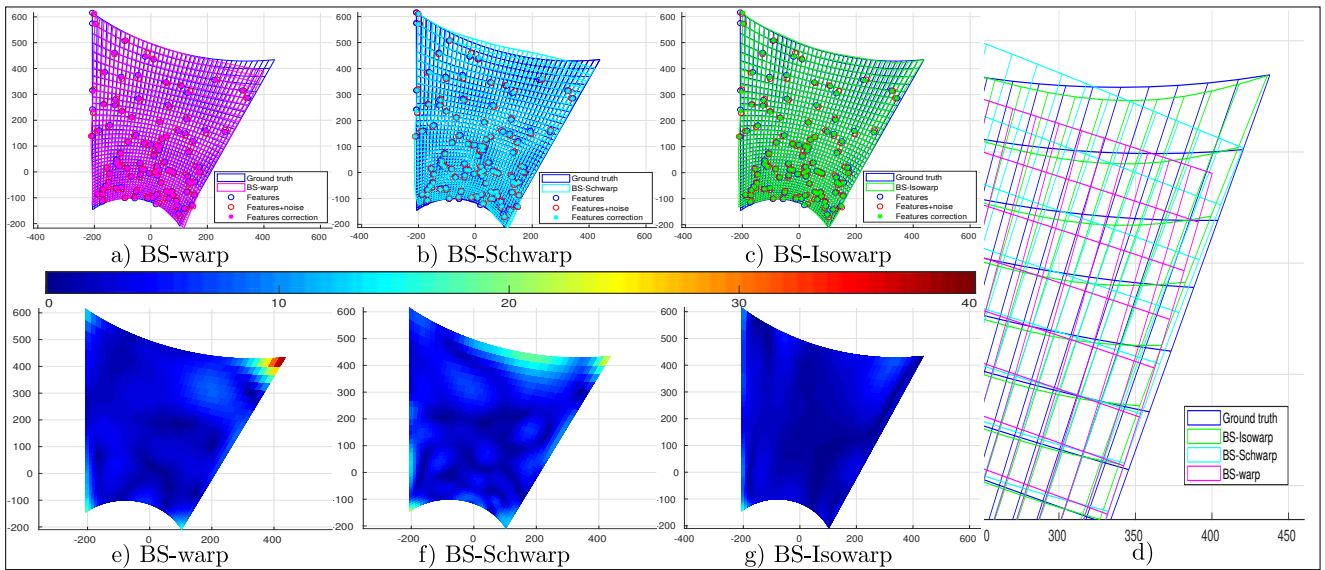


Fig. 10 Registration results in the deformable case. *a)*, *b)* and *c)* are the warps obtained with *BS-warp*, *BS-Schwarp* and *BS-Isowarp* respectively. *d)* shows a comparison of the behaviour of each method in a corner. *e)*, *f)* and *g)* represent a 2D error heat map for each warp. See Section 6.2.2 for more details.

slightly better results than *BS-Schwarp*. This example shows that the isowarp is able to properly integrate the 3D geometry, which leads to better reconstructions even if it does not achieve the best registration results. *BS-warp* is consistently worse than the other methods in both registration and reconstruction. *BS-warp+Ref* achieves good results closed to *BS-Isowarp* and *BS-Schwarp*. *TPS-Chhatkuli17* outperforms *BS-warp* but is not on par with the other methods.

Fig. 13.*b)* shows the results for the deformable case. *BS-Isowarp* clearly outperforms *BS-Schwarp* and *BS-warp* in terms of registration. It reduces by at least half the 2D error. *BS-Isowarp* also produces the best reconstructions, closely followed by *BS-warp+Ref*. It achieves strikingly better reconstruction results than the other non-refinement methods, reducing by about 20 times the 3D error, even when no noise is added to the correspondences. *TPS-Chhatkuli17* obtains worse results than *BS-warp*. This is caused by the factors explained in Section 6.2.2. Finally, *BS-Schwarp* obtains the worst results in 3D reconstruction. This is probably due to the overly constraining local planarity assumption.

6.2.5 The Effect of Curvature

We evaluate the effect of surface curvature on warp estimation. We proceed by increasing the radius of the cylinder between 1m to 16m. The higher the radius the lower the curvature. We add Gaussian noise with 5 pixels standard deviation. Fig. 14 shows the behavior of the evaluated methods with the amount of curvature. As expected, *BS-Schwarp* obtains better registration

results with lower curvature. *BS-Isowarp* achieves similar registration results than *BS-Schwarp* and *BS-warp* obtains the worst results in all cases.

In 3D reconstruction, *BS-warp*, *BS-Schwarp* and *TPS-Chhatkuli17* are very sensitive to curvature, showing worse results with higher curvature. *BS-Isowarp* and *BS-warp+Ref* show resistance against curvature variation. Besides, *BS-Isowarp* has better reconstruction results than the refinement method *BS-warp+Ref*. This shows that our isowarp equations are exact irrespective of the surface curvature.

6.2.6 Shoe Experiment

We tested the *BS-Isowarp* with a hyperrealistic recreation of a shoe, the *Shoe dataset*, from a volumic template introduced in [16]. The shoe template was built with a triangular mesh (around 5200 mesh faces) using dense SfM (Agisoft Photoscan [1]). The *Shoe dataset* is composed of random quasi-isometric deformations simulated by Blender [7], together with the corresponding rendered images with a resolution of 960×540 pixels and a focal length of 530 pixels. Approximately 5000 point correspondences are drawn between the template and the input image.

For the shoe experiment, we choose 850 point correspondences to build warps. The other available correspondences are used to compute 2D and 3D errors. Fig. 15 shows the results of isometric SFT for different warp models and reconstruction methods. All the evaluated methods present similar behavior in terms of 2D error. In reconstruction, *BS-Isowarp* achieves the best

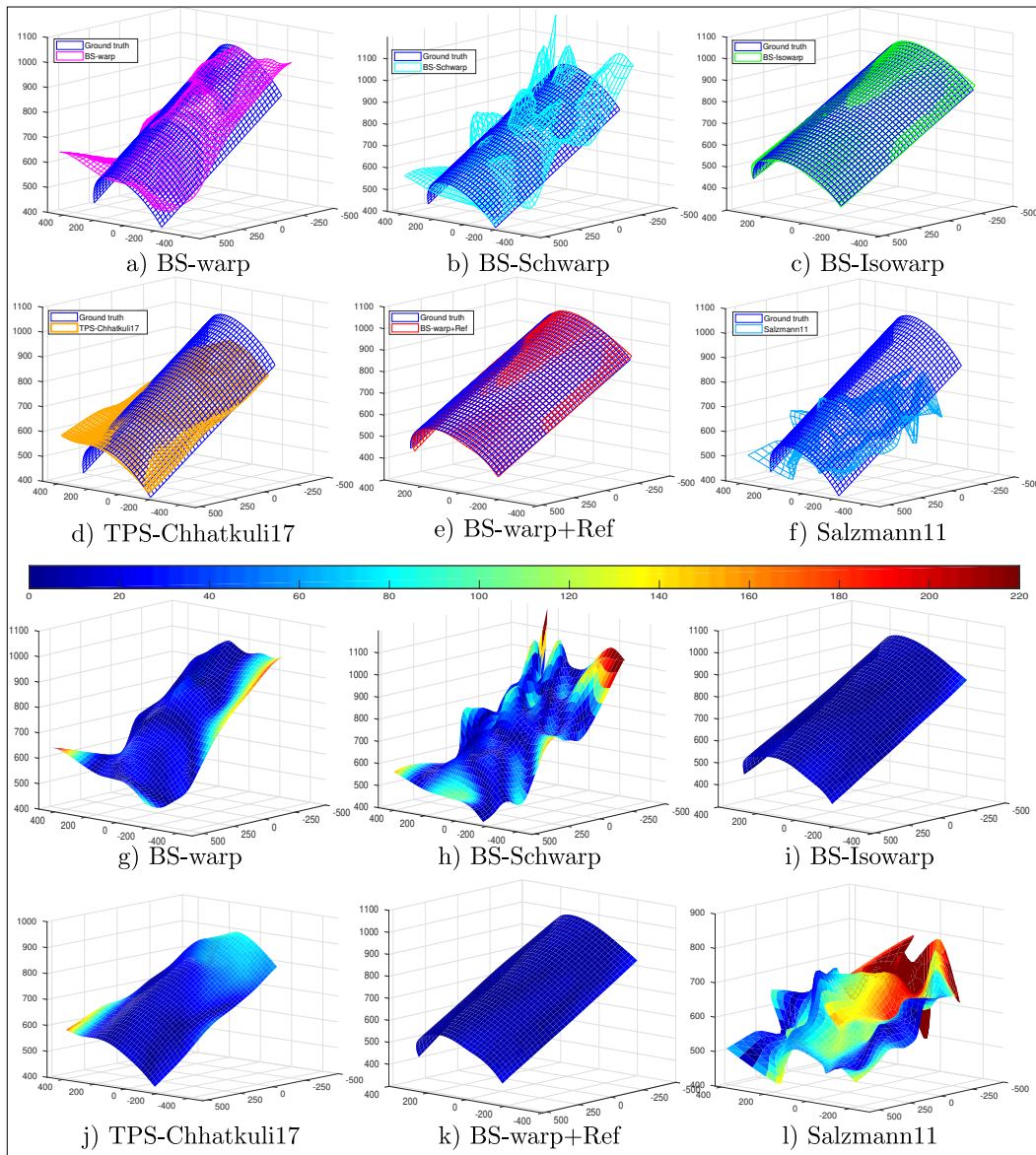


Fig. 11 Reconstructions achieved in the deformable case. *a)*, *b)*, *c)*, *d)*, *e)* and *f)* show the 3D reconstruction results of *BS-warp*, *BS-Schwarz*, *BS-Isowarp*, *TPS-Chhatkuli17*, *BS-warp+Ref* and *Salzmann11* respectively. *g)*, *h)*, *i)*, *j)*, *k)* and *l)* show a heat map for each warp that shows the 3D error at each point, see Section 6.2.2 for more details.

results, see Fig. 15g). It fits the surface shape better than any other evaluated methods. *TPS-Chhatkuli17* and the 3D refinement method *BS-warp+Ref* achieve good results in 3D reconstruction but still far from *BS-Isowarp*, see Fig. 15h) and Fig. 15j). The refinement method gets stuck in a local minimum, which prevents it from providing a better surface reconstruction. *BS-warp*, *Salzmann11* and *BS-Schwarz* give worse reconstructions (see Fig. 15e), Fig. 15i) and Fig. 15f)). The *BS-warp* and *BS-Schwarz* solutions present strong ripples while *Salzmann11* obtains a smooth surfaces but far from the ground truth.

6.3 Real Data Experiments

We propose experiments with three real datasets.

The *Zooming dataset* [12] consists of several deformations of a folded A4 piece of paper from different viewpoints and focal lengths. Specifically, the focal length varies between 1348 and 3937 pixels in 9 different values. The image resolution is 1728×1552 pixels. The dataset provides between 7 to 10 different images for each zoom level. Feature correspondences between the template and the image are computed using SIFT [21], eliminating possible mismatches with [30]. We obtain around 1200 point correspondences for each view. The

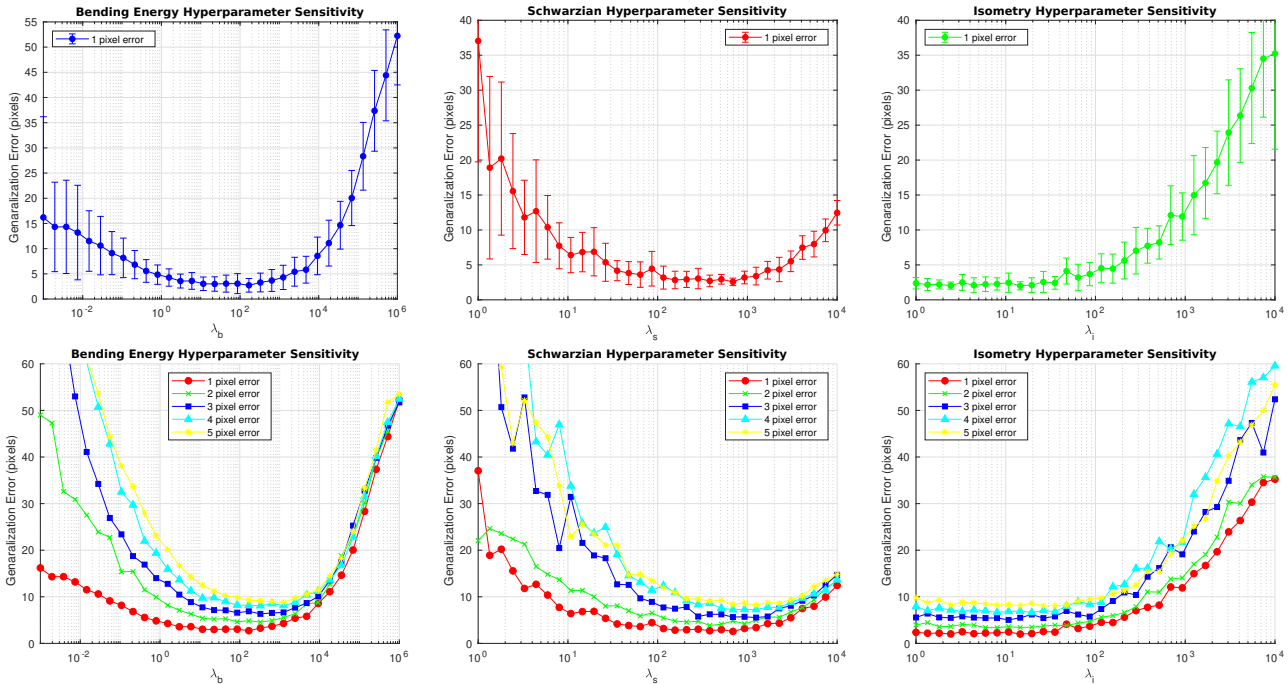


Fig. 12 Sensitivity of the evaluated warps to their main hyperparameter, see Table 6. The first row shows 2D error mean and standard deviation against hyperparameter value for a 1 pixel noise. The second row shows 2D error for a noise from 1 to 5 pixels with a 1 pixel step. See Section 6.2.3 for details.

template and groundtruth shapes are computed using dense Structure-from-Motion (SfM) [1].

The *Cushion dataset* [12] uses a non-developable template that deforms isometrically. Image resolution is 3456×2304 pixels and feature correspondences are extracted using SURF [6] eliminating possible mismatches with [30]. We obtain around 1800 point correspondences for each view in this experiment. The camera focal length is 2700 pixels. As in the previous dataset, templates and reconstruction ground truth are obtained with dense SfM [1].

The *Can dataset* [12] also uses a non-developable template, a can made of cardboard deforming isometrically. In this experiment, a 3D template model is thus considered. The image resolution is 4800×3200 pixels with a focal length of 11000 pixels. Keypoint correspondences are extracted by combining SIFT and KAZE, giving around 7000 correspondences. The template and reconstruction ground truth were obtained with dense SfM [1].

6.3.1 Zooming Experiment

We select one image from each zoom level of the dataset with a similar viewpoint. We choose 100 point correspondences from each image to compute the warp. The other correspondences are used as the test set to compute the 2D error and 3D error. Fig. 16 shows the

warps computed from each selected frame. We show the original image, and the results obtained with *BS-warp*, *BS-Schwarz* and *BS-Isowarp*. We include the 2D error in pixels. All methods achieve similar results in terms of 2D error, which is consistent with the experiments using synthetic data. Fig. 17 shows the reconstruction results. Our *BS-Isowarp* achieves the best results followed by the 3D refinement method *BS-warp+Ref*. *TPS-Chhatkuli17* is third in reconstruction quality and the obtained error is coherent with the results given in [12]. *BS-Schwarz* obtains worse reconstruction results than *BS-warp*. We point out that *BS-warp*, *TPS-Chhatkuli17* and *BS-warp+Ref* obtain similar 3D error results than the ones reported in [12]. However, as explained in Section 6, we report generalization 2D and 3D errors by using a set of point correspondences different from the one used to compute the warp (similarly to the training and test sets used in machine learning). On the contrary, the results in [12] (*TPS-Chhatkuli17*) use all the available correspondences for computing the warps and report reconstruction errors at those same correspondences (in the analogy with machine learning, they use the exact same training and test sets). This can lead to discrepancies between our results and theirs. Nevertheless, despite these methodological differences, our results do not differ that much from the reported values in the literature. In some reconstructions, such as Zooms 8 and 9, there might be

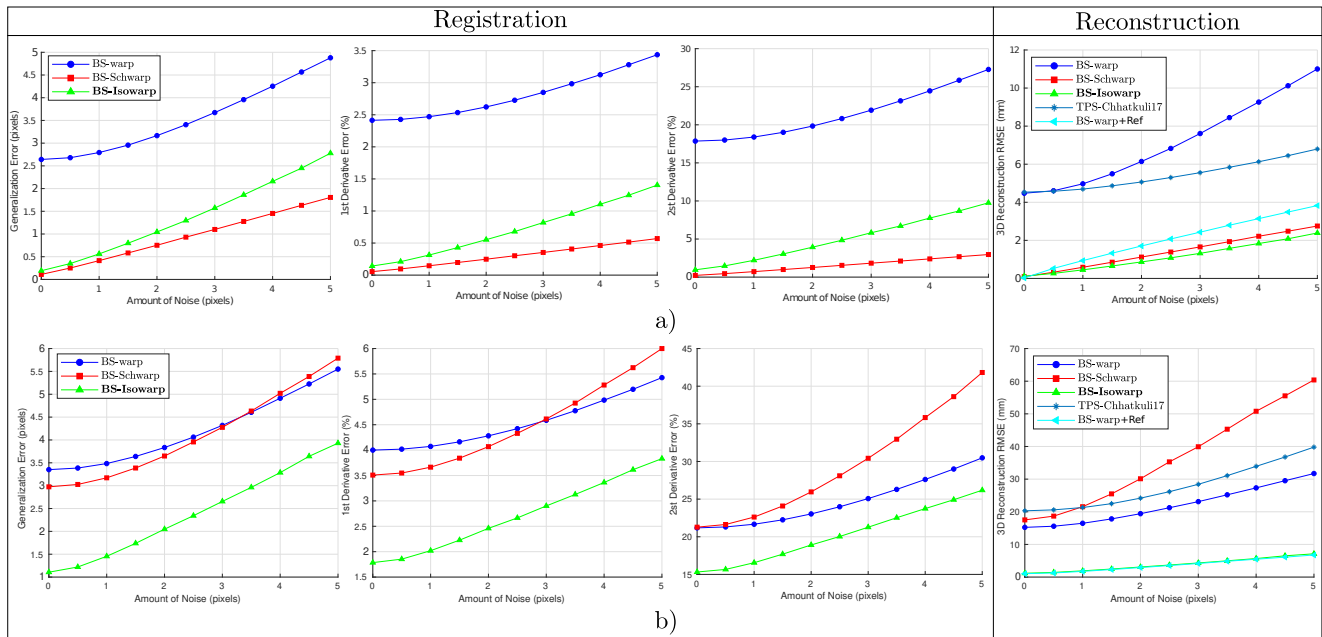


Fig. 13 Effect of noise. The registration and reconstruction results are given for the rigid case (first row) and the deformable case (second row). The first three columns show the 2D error, including 1st and 2nd derivatives error. The last column shows the reconstruction 3D error.

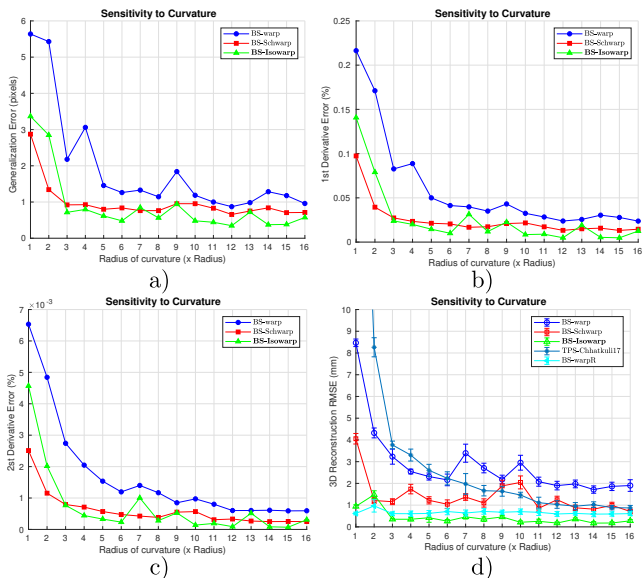


Fig. 14 Sensitivity to curvature. *a)*, *b)* and *c)* show 2D error and relative first and second-order derivatives errors respectively. *d)* shows 3D error.

noticeable qualitative differences between the reconstructed surface and the provided 3D error, especially in corners (see *BS-warp+Ref*). Although qualitative assessments are highly debatable, we have observed that *TPS-Chhatkuli17* often misreconstructs the corners of the surface and generally oversmooths the reconstruction. Although it might look good at first glance, it does not fit the ground-truth surface properly. Con-

cerning 3D refinement results, we have observed that it has offset errors, which are visible in the last two rows, and fails to recover the surface curvature well. The *BS-Isowarp* looks slightly less smooth in the last two cases. Still, it does not suffer offset errors and correctly estimates the curvature, especially in the problematic corners. This explains why the *BS-Isowarp* obtains better reconstruction error than the other methods.

Fig. 18 presents the results of 2D and 3D error averaged over the set of all different viewpoints provided by the *Zooming dataset*. All methods obtain good results in 2D error, which is under 1.2 pixels. In 3D reconstruction, *BS-Isowarp* achieves the best results followed by *BS-warp+Ref*. *TPS-Chhatkuli17* is consistently better than *BS-warp*, *Salzmann11* and *BS-Schwarp*. *Salzmann11* overestimates the depth due to the inextensibility relaxation of isometry.

6.3.2 Cushion Experiment

We choose 100 point correspondences to build warps. The other available correspondences from the database are used to compute the 2D error and 3D error. Figure 2 presents the results achieved in this dataset. All evaluated methods present similar behavior in terms of 2D error, the same behavior than in the previous experiments. In reconstruction, *BS-Isowarp* achieves the best results, see Fig. 2*g)* followed by the 3D refinement method *BS-warp+Ref* presented in Fig. 2*j)*. *TPS-Chhatkuli17* achieves good results in 3D reconstruction

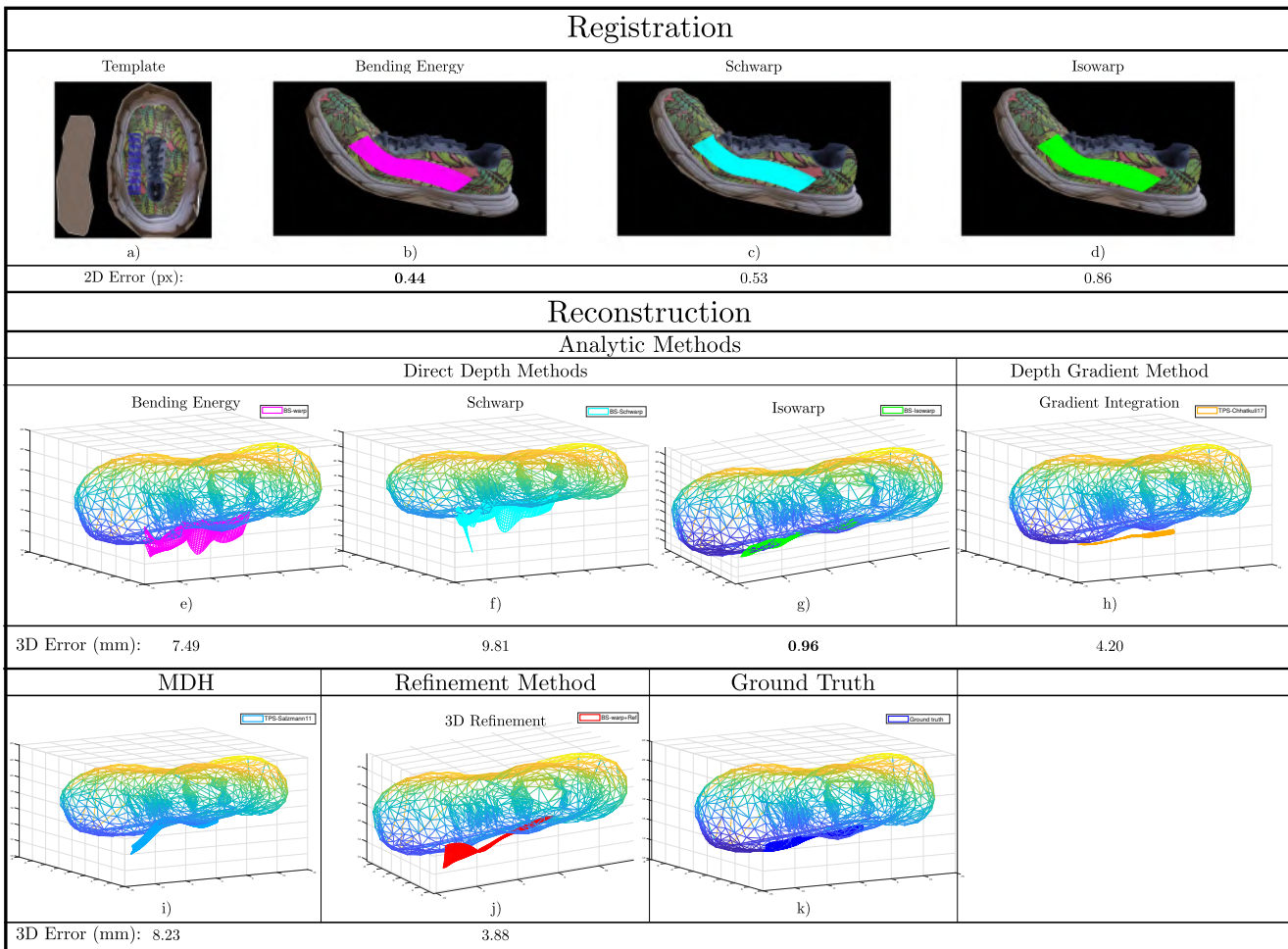


Fig. 15 Results of isometric SfT for different warp models and reconstruction methods on the shoe dataset. *a)*, *b)* and *c)* are the warps obtained with *BS-warp*, *BS-Schwarp* and *BS-Isowarp* respectively; *e)*, *f)*, *g)*, *h)*, *i)* and *j)* show the 3D reconstruction results of *BS-warp*, *BS-Schwarp*, *BS-Isowarp*, *TPS-Chhatkuli17*, *Salzmann11* and *BS-warp+Ref*. The reconstruction ground truth in *k)* is obtained using dense SfM from multiple images [1].

but is still far from *BS-Isowarp* or *BS-warp+Ref*, see Fig. 2*h*). Both *BS-warp*, *Salzmann11* and *BS-Schwarp* give worse reconstructions (see Fig. 2*e*), Fig. 2*i*) and Fig. 2*f*)).

6.3.3 Can Experiment

In this experiment, we use 850 point correspondences to build the warps. The other available correspondences are used to compute the 2D error and 3D error. Fig. 19 shows the results achieved in this dataset. As in the Cushion or Zooming dataset experiments, all the evaluated methods present similar behavior in terms of 2D error. In reconstruction, *BS-Isowarp* achieves the best results, see Fig. 19*h*), closely followed by the 3D refinement method *BS-warp+Ref* presented in Fig. 19*k*). *TPS-Chhatkuli17* achieves very good results in 3D reconstruction but with a larger error than *BS-Isowarp* and *BS-warp+Ref*, especially in the top cor-

ners, see Fig. 19*i*). Their reconstructions deteriorate in the edges and corners. Both *BS-warp*, *Salzmann11* and *BS-Schwarp* give worse reconstructions (see Fig. 19*f*), Fig. 19*j*) and Fig. 19*g*)). *Salzmann11* especially suffers the non-developable template of this experiment.

We also test the *BS-Isowarp* against occlusions using the Can dataset. The simulated occlusion consists of a circular object placed at the image center. We increase its radius from 100 to 1000 pixels removing the correspondences in the occlusion area. Table 7 shows the number of feature correspondences and the 3D error obtained by our method. Fig 20 shows specific 3D reconstructions corresponding to the occlusion with radius 300, 600 and 900 pixels.

We observe that *BS-Isowarp* is robust against occlusions when the occlusion radius varies between 100 and 700 pixels. The 3D error increases but the reconstructed shape does not substantially changes. The smoothing term and the remaining sparse correspondences are

Radius (pixels)	100	200	300	400	500	600	700	800	900	1000
Nb. corresp.	836	803	754	693	656	606	525	427	288	199
3D Error (mm)	3.48	3.53	3.64	3.67	3.79	3.96	4.51	7.20	18.43	20.1

Table 7 3D error obtained by *BS-Isowarp* for the occlusion experiment in the Can dataset.

enough to obtain a suitable warp, which results in a good reconstruction. However, with more severe occlusions, the isowarp degenerates and *BS-Isowarp* fails to correctly reconstruct the surface.

6.3.4 Comparison with DNN-based Methods

We compare *BS-Isowarp* with recent DNN-based SfT methods: *DeformNet* [33], *DeepSfT* [16], and *IsMo-GAN* [36]. *DeformNet* and *DeepSfT* are object-specific SfT methods, meaning that they are trained to reconstruct a specific template, both in shape and in texture map. In contrast, *IsMo-GAN* is an object-generic method trained with a dataset of thin-shell deformable objects with varying texture. In terms of object representation, *DeformNet* and *IsMo-GAN* use a regular mesh to represent the object’s shape, and *DeepSfT* recovers a pixel-based depth map.

We compare these methods using the publicly available Kinect paper dataset [38], commonly used as a baseline in SfT methods. Briefly, the dataset comprises 191 images of a deforming paper acquired with the Kinect sensor. It provides ground-truth for both registration and reconstruction with point correspondences and their 3D coordinates. The implementation of *DeformNet* [33] is not publicly available, but it provides error metrics in this dataset. We reproduced the same training conditions as reported in *DeformNet*, where 15% of the frames were used to fine tune the neural network, to adapt *DeepSfT* and *DeformNet* to this dataset. Although *IsMo-GAN* does not need in principle to be retrained, we fine-tuned it with the same training set to produce a fair comparison between all DNN-based SfT methods. We use 250 correspondences in our *BS-Isowarp* and *BS-warp+Ref* methods and leave the rest as ground truth to compute 3D errors. Importantly, *DeformNet* [33] uses a Procrustes step, which computes the best 3D similarity transformation, to align the reconstruction with the ground-truth shape. This step should not be necessary for isometric SfT since it recovers the surface without ambiguities in the camera reference frame. In our understanding, this reveals some DNN methods’ inability to precisely recover the scale, rotation, and translation of the object. Nevertheless, we have included the Procrustes step in our experiments to give a fair comparison with their reported results on this dataset.

Table 8 summarizes the obtained 3D error for each method. *BS-Isowarp* obtains the best result with

Method	IsMo-GAN	DeepSfT	DeformNet	BS-Isowarp	BS-warp+Ref
K. Paper	10.21	6.18	4.79	1.78	1.79

Table 8 Comparison of DNN-based SfT method with *BS-Isowarp* in the Kinect paper dataset. The 3D error is given in millimeters.

1.78 mm of 3D error closely followed by the refinement method *BS-warp+Ref* 1.79 mm. Both methods retrieve quite similar reconstruction surfaces. *DeformNet* is the DNN-based SfT method with the best result 4.79 mm, far from the previous ones. The object-generic method *IsMo-GAN* obtains the worst result. Without the final Procrustes step, our *BS-Isowarp* obtains a 3D error of 5.48 mm which is still comparable with the results reported by *DeformNet*. Unfortunately, the authors of *DeformNet* do not report the 3D error without the Procrustes step, which will naturally increase this 3D error. We point out that retrieving the scale and distance to the image plane to the surface is not an ambiguous problem with isometric deformations and a perspective camera model [4], hence, it should not be necessary to use the Procrustes step.

Figure 21 shows the reconstruction of two different frames (frames 140 and 155 specifically) of the Kinect paper dataset, which presents strong deformation. *BS-Isowarp* and *BS-warp+Ref* obtains similar 3D errors, see Figs 21.c), d), g) and h). *DeepSfT* and *IsMo-GAN* obtain worse reconstructions compared with the previous one. *IsMo-GAN* struggles in corners and edges of the reconstruction, see Figs 21.a) and f). *DeepSfT* suffers from high-frequency noise which is a known problem in dense methods, see Figs 21.b) and d).

Overall, our *BS-Isowarp* method outperforms the aforementioned DNN-based methods in the Kinect paper dataset, without requiring training. A critical assessment of these results however leads to the observation that a direct comparison between classical and DNN-based methods is not entirely fair for two opposite reasons. The first reason is that DNN-based methods use all the image data, whereas many classical methods, including the *BS-Isowarp*, depend on the quality and quantity of the point correspondences they use as input. In that regard, our results with the so-called classical methods could be improved with better point correspondences or registration strategy. The second reason is that DNN-based methods strongly depend on the amount and type of training data. Yet, it is important to realise that there does not exist a universal DNN-based SfT method that can operate without fine-

tuning on frames extracted from the test dataset, which currently represents a major issue.

7 Conclusion

We have studied the theoretical properties of warps in isometric SfT, finding differential constraints they must fulfill. This is a theoretical breakthrough in our understanding of the problem. On the practical side, this allows us to find a warp, called the isowarp, that is 3D consistent with the physics of the problem. This is a change of paradigm in isometric SfT as we have shown that with the isowarp, one outperforms the best existing reconstruction methods by using the analytic direct depth solution [4]. Besides, computing the isowarp has several practical advantages over the other reconstruction methods based on iterative refinement, namely lower complexity, higher accuracy, convex initialization, and automatic hyperparameter tuning. Isowarp initiates a new line of reconstruction methods which imposes 3D geometric constraints on the warps. We expect the isowarp to be an essential component in future SfT methods.

A Derivation of the Image Embedding

We assume that the image plane is at $z = 1$ in camera coordinates, which is achieved by working in retinal coordinates. The perspective projection of a point (x, y, z) is then given by:

$$\Pi_p: \mathbb{R}^3 \setminus \{(x, y, z) \in \mathbb{R}^3 \mid z = 0\} \rightarrow \mathbb{R}^2$$

$$(x, y, z)^\top \mapsto \begin{pmatrix} x \\ y \\ z \end{pmatrix}^\top. \quad (25)$$

The inverse of the restriction $\Pi_p|_{\mathcal{S}}: \mathcal{S} \rightarrow \mathbb{R}^2$ of Π_p is the *image embedding*. It forms a depth based parametrization of the surface (\mathcal{I}, X_i) expressed in terms of the depth function $\rho: \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$X_i(u', v') = \rho(u', v') (u', v', 1)^\top, \quad (26)$$

where u' and v' represent the image coordinates. Alternatively to ρ , we define the Euclidean distance between the camera's projection origin and the surface point as $\tilde{\rho}: \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$\tilde{\rho}(u', v') = \rho(u', v') \zeta(u', v'), \quad (27)$$

where $\zeta(u', v') = \sqrt{1 + u'^2 + v'^2}$. The perspective parametrization (\mathcal{I}, X_i) can be then expressed in terms of $\tilde{\rho}$ as:

$$X_i(u', v') = \frac{\tilde{\rho}(u', v')}{\zeta(u', v')} (u', v', 1)^\top. \quad (28)$$

Now, we can define the surface \mathcal{S} from the template parametrization domain \mathcal{U} by composing the previous parametrization X_i and the warp function η as follows:

$$\bar{X}_i(u, v) = X_i \circ \eta = \frac{\tilde{\rho}(\eta(u, v))}{\zeta(\eta(u, v))} (\eta(u, v), 1)^\top, \quad (29)$$

where u and v are template domain coordinates.

Defining the depth function $\bar{\rho}: \mathbb{R}^2 \rightarrow \mathbb{R}$ as the depth function $\tilde{\rho}$ in u, v coordinates by the composition $\bar{\rho} = \tilde{\rho} \circ \eta$, we obtain:

$$\bar{X}_i(u, v) = X_i \circ \eta = \frac{\bar{\rho}(u, v)}{\zeta(\eta(u, v))} (\eta(u, v), 1)^\top. \quad (30)$$

Working with the parametrization (\mathcal{U}, \bar{X}_i) of \mathcal{S} has two principal advantages. First, it allows us to compute the *first fundamental form*, also known as the *metric tensor*, over the same parametrization domain as the template, which is essential to obtain the isowarp equations. Second, it greatly simplifies these equations.

B Derivation of the Isowarp Equations

We give Matlab code to establish the Isowarp equations (11). These equations are too lengthy to be reproduced fully expanded. However we recall that, importantly, they depend on the known template and the unknown warp η only. More specifically, they are quadratic of the second-order in η .

```

1 %% Isowarp Equations Matlab Code
2 clc; clear all;
3
4 syms u v % Variables
5 syms W1(u,v) W2(u,v) % Warp
6 % Template's First Fundamental Form
7 syms U11(u,v) U12(u,v) U22(u,v)
8
9 U = [U11, U12; U12, U22];
10
11 W1u = diff(W1,u); W2u = diff(W2,u);
12 W1v = diff(W1,v); W2v = diff(W2,v);
13
14 Jw = [W1u, W1v; W2u, W2v];
15 JtJ = transpose(Jw)*Jw;
16 wrpMod2 = (1 + W1^2 + W2^2);
17 JwTW = transpose(Jw)*([W1;W2]);
18 JwTW = JwTW * transpose(JwTW);
19 H = (1/wrpMod2) * (JtJ - (1/wrpMod2)*JwTW);
20
21 S = U*inv(H); St = trace(S); Sd = det(S);
22 rho = sqrt(St/2-sqrt(St^2/4-Sd));
23 rho_u = diff(rho,u);
24 rho_v = diff(rho,v);
25
26 L = (U - H*rho^2); L = L(u,v);
27 L11 = L(1,1); L12 = L(1,2); L22 = L(2,2);
28
29 disp('Calculating EQs...')
30 EQ1 = simplify(rho_u^2 - L11);
31 EQ2 = simplify(rho_u*rho_v - L12);
32 EQ3 = simplify(rho_v^2 - L22);

```

Acknowledgements This research has received funding from the Spanish Ministry of Education and Culture under the scholarship FPU, the Spanish Ministry of Economy, Industry and Competitiveness under the project ARTEMISA (TIN2016-80939-R) and the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

1. Agisoft Photoscan 1.0.4 (2014). URL <http://www.agisoft.ru/products/photoscan>.
2. Agudo, A., Agapito, L., Calvo, B., Montiel, J.M.: Good vibrations: A modal analysis approach for sequential non-rigid structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1558–1565 (2014)
3. Agudo, A., Moreno-Noguer, F., Calvo, B., Montiel, J.M.M.: Sequential non-rigid structure from motion using physical priors. *IEEE transactions on pattern analysis and machine intelligence* **38**(5), 979–994 (2016)
4. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., Pizarro, D.: Shape-from-template. *IEEE transactions on pattern analysis and machine intelligence* **37**(10), 2099–2118 (2015)
5. Bartoli, A., Perriollat, M., Chambon, S.: Generalized thin-plate spline warps. *International Journal of Computer Vision* **88**(1), 85–110 (2010)
6. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision, pp. 404–417. Springer (2006)
7. Blender Online Community: Blender - a 3D modelling and rendering package. Blender Foundation, Blender Institute, Amsterdam. URL <http://www.blender.org>
8. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (6), 567–585 (1989)
9. Brunet, F., Bartoli, A., Hartley, R.I.: Monocular template-based 3D surface reconstruction: Convex inextensible and nonconvex isometric methods. *Computer Vision and Image Understanding* **125**, 138–154 (2014)
10. Brunet, F., Bartoli, A., Navab, N., Malgouyres, R., et al.: NURBS warps. In: British machine vision conference (2009)
11. Casillas-Perez, D., Pizarro, D., Fuentes-Jimenez, D., Mazo, M., Bartoli, A.: Equiareal Shape-from-Template. *Journal of Mathematical Imaging and Vision* (2018). DOI 10.1007/s10851-018-0862-5
12. Chhatkuli, A., Pizarro, D., Bartoli, A., Collins, T.: A stable analytical framework for isometric shape-from-template by surface integration. *IEEE transactions on pattern analysis and machine intelligence* **39**(5), 833–850 (2017)
13. Do Carmo, M.P.: *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications (2016)
14. Dubitzky, W., Granzow, M., Berrar, D.P.: *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media (2007)
15. Fayad, J., Russell, C., Agapito, L.: Automated articulated structure and 3d shape recovery from point correspondences. In: 2011 International Conference on Computer Vision, pp. 431–438. IEEE (2011)
16. Fuentes-Jiménez, D., Casillas-Pérez, D., Pizarro-Pérez, D., Collins, T., Bartoli, A.: Deep shape-from-template: Wide-baseline, dense and fast registration and deformable reconstruction from a single image. *CoRR abs/1811.07791* (2018). URL <http://arxiv.org/abs/1811.07791>
17. Golyanik, V., Shimada, S., Varanasi, K., Stricker, D.: Hdm-net: Monocular non-rigid 3d reconstruction with learned deformation model. In: International Conference on Virtual Reality and Augmented Reality, pp. 51–72. Springer (2018)
18. Haoachine, N., Dequidt, J., Berger, M.O., Cotin, S.: Single view augmentation of 3D elastic objects. In: ISMAR, pp. 229–236. IEEE (2014)
19. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. Cambridge university press (2003)
20. Horn, B.K., Schunck, B.G.: Determining optical flow. *Artificial intelligence* **17**(1-3), 185–203 (1981)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2), 91–110 (2004). DOI 10.1023/B:VISI.0000029664.99615.94. URL <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
22. Malti, A., Bartoli, A., Hartley, R.: A linear least-squares solution to elastic shape-from-template. In: CVPR, pp. 1629–1637 (2015)
23. Malti, A., Herzet, C.: Elastic shape-from-template with spatially sparse deforming forces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3337–3345 (2017)
24. Ngo, D.T., Östlund, J., Fua, P.: Template-based monocular 3d shape recovery using laplacian meshes. *IEEE transactions on pattern analysis and machine intelligence* **38**(1), 172–187 (2016)
25. Ovsienko, V., Tabachnikov, S.: What is the schwarzian derivative. *Notices of the AMS* **56**(1), 34–36 (2009)
26. Özgür, E., Bartoli, A.: Particle-SfT: A provably-convergent, fast shape-from-template algorithm. *International Journal of Computer Vision* **123**(2), 184–205 (2017)
27. Parashar, S., Pizarro, D., Bartoli, A.: Local deformable 3d reconstruction with cartan’s connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**(12), 3011–3026 (2020). DOI 10.1109/TPAMI.2019.2920821
28. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. *International journal of computer vision* **95**(2), 124–137 (2011)
29. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision* **76**(2), 109–122 (2008)
30. Pizarro, D., Bartoli, A.: Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision* **97**(1), 54–70 (2012)
31. Pizarro, D., Bartoli, A., Collins, T.: Isowarp and conwarp: Warps that exactly comply with weak-perspective projection of deforming objects. In: BMVC (2013)
32. Pizarro, D., Khan, R., Bartoli, A.: Schwarzp: Locally projective image warps based on 2d schwarzian derivatives. *International Journal of Computer Vision* **119**(2), 93–109 (2016). DOI 10.1007/s11263-016-0882-9
33. Pumarola, A., Agudo, A., Porzi, L., Sanfeliu, A., Lepetit, V., Moreno-Noguer, F.: Geometry-aware network for non-rigid shape prediction from a single view. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2018)
34. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging* **18**(8), 712–721 (1999). DOI 10.1109/42.796284
35. Salzmann, M., Fua, P.: Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(5), 931–944 (2011)
36. Shimada, S., Golyanik, V., Theobalt, C., Stricker, D.: Ismo-gan: Adversarial learning for monocular non-rigid

- 3d reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0 (2019)
37. Sundaram, N., Brox, T., Keutzer, K.: Dense point trajectories by GPU-accelerated large displacement optical flow. In: European conference on computer vision, pp. 438–451. Springer (2010)
 38. Varol, A., Salzmann, M., Fua, P., Urtasun, R.: A constrained latent variable model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2248–2255. Ieee (2012)
 39. Yu, R., Russell, C., Campbell, N.D., Agapito, L.: Direct, dense, and deformable: Template-based non-rigid 3D reconstruction from RGB video. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 918–926 (2015)
 40. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l1 optical flow. In: F.A. Hamprecht, C. Schnörr, B. Jähne (eds.) Pattern Recognition, pp. 214–223. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)

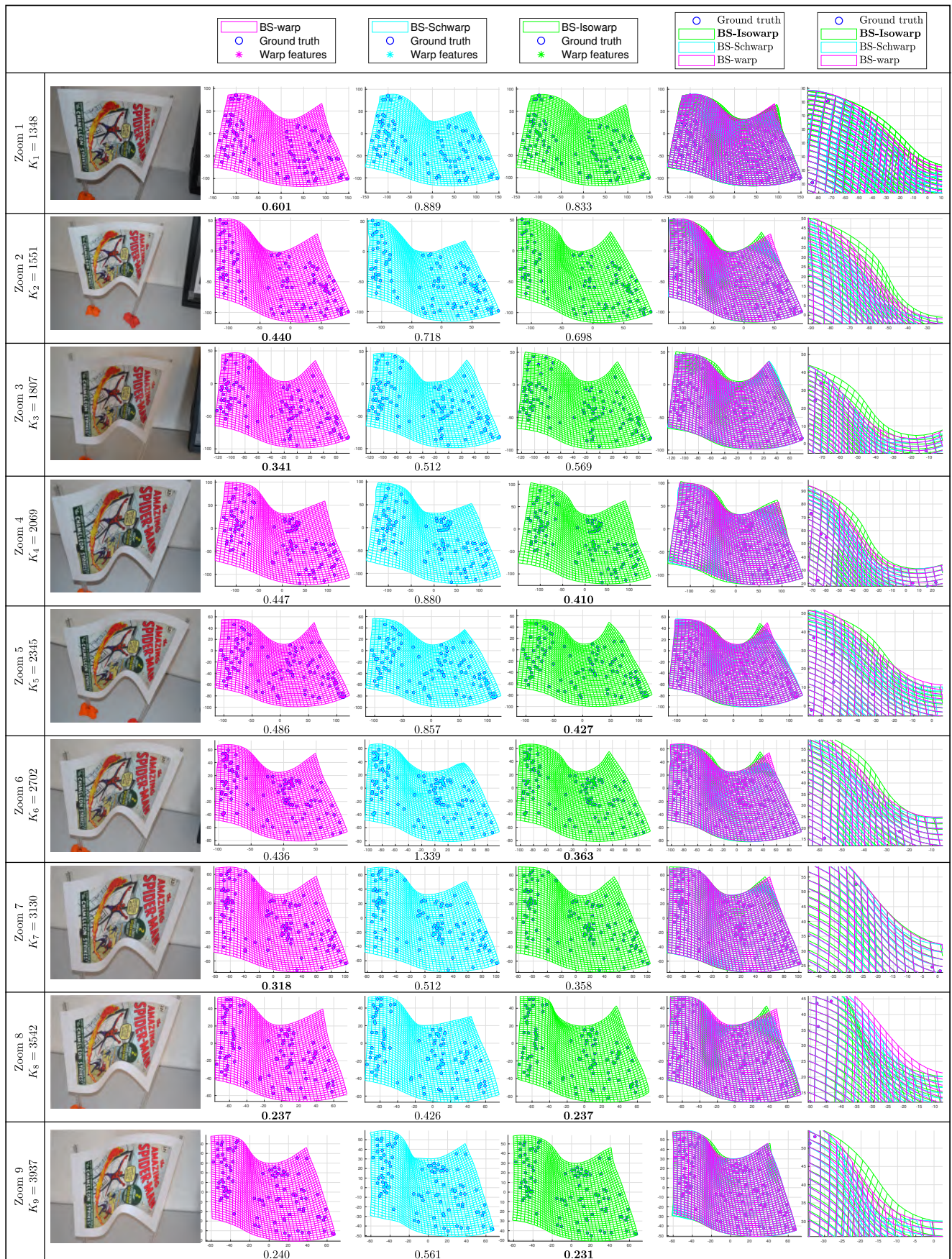


Fig. 16 Registration results for the *Zooming* experiment. *BS-warp*, *BS-Schwarp* and *BS-Isowarp* are evaluated. Each row show one view from the dataset with different values of focal length. The 2D error is given under each warp in pixels.

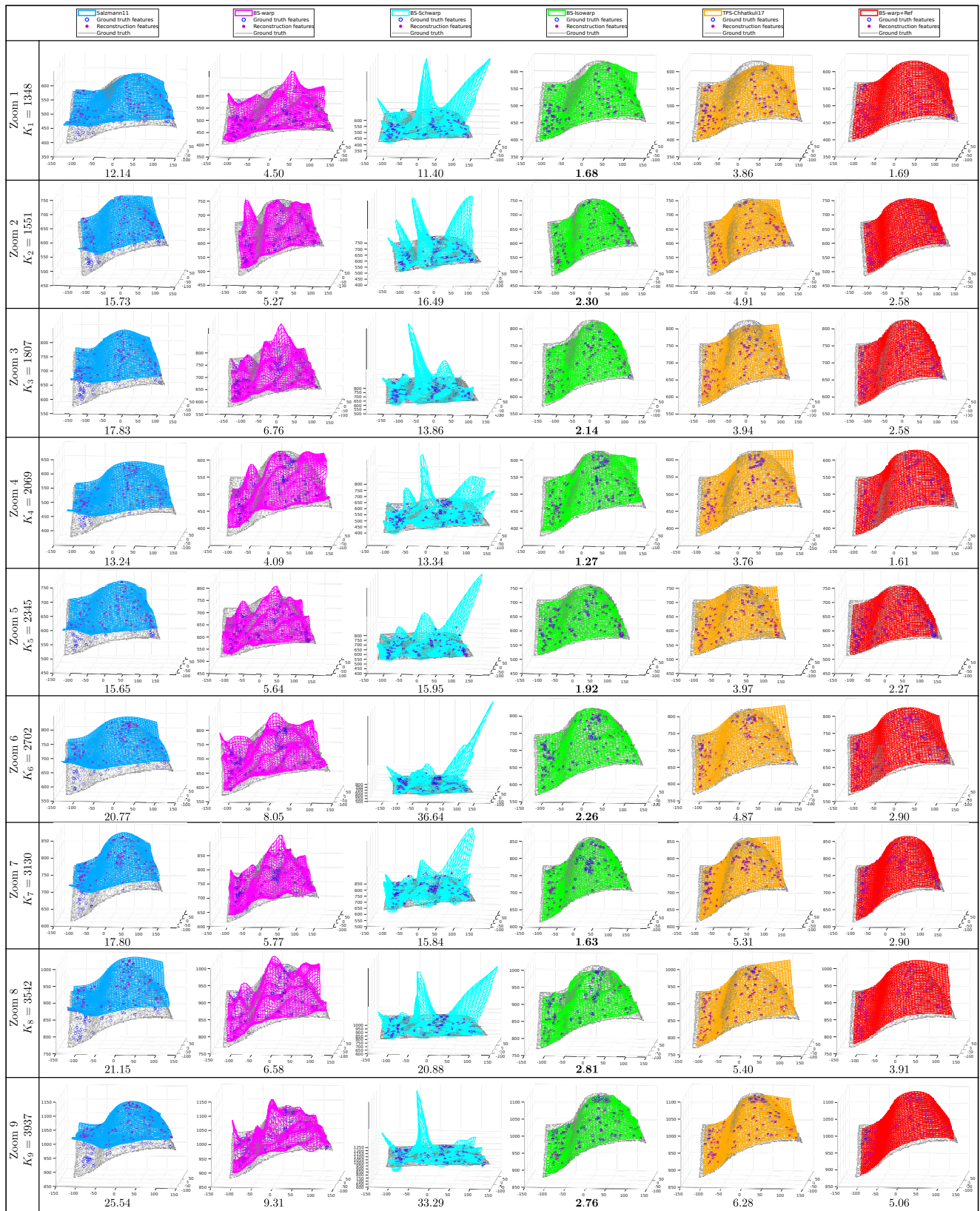


Fig. 17 3D reconstruction results from all evaluated methods in the *Zooming* experiment. *Salzmänn11*, *BS-warp*, *BS-Schwarp*, *BS-Isowarp*, *TPS-Chhatkuli17* and *BS-warp-Ref* are compared with nine images from the *Zooming* dataset with different values of focal length. Ground truth and reconstructed point correspondences are also shown. The 3D error is given under each 3D reconstruction in millimeters.

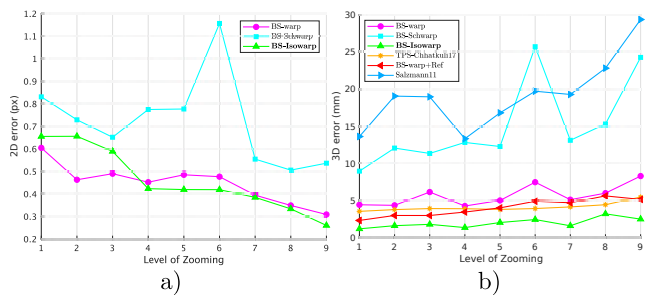


Fig. 18 *Zooming* dataset average registration and reconstruction errors. a) average 2D error for each zoom level. b) 3D error average for each zoom level. See Section 6.3.1 for more details.

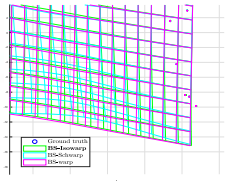




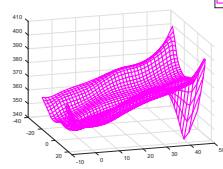
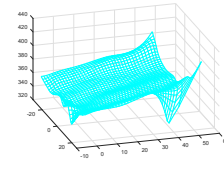
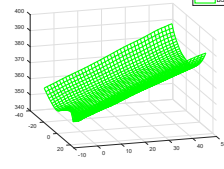
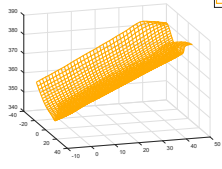
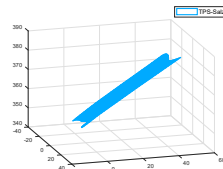
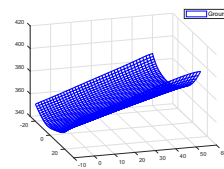
Registration				
Template	Bending Energy	Schwarp	Isowarp	
				
a)	b)	c)	d)	e)
2D Error (px):	0.56	0.96	0.49	
Reconstruction				
Analytic Methods				
Direct Depth Methods			Depth Gradient Method	
Bending Energy	Schwarp	Isowarp	Gradient Integration	
				
f)	g)	h)	i)	
3D Error (mm):	7.49	9.81	3.07	5.28
MDH	Refinement Method		Ground Truth	
	3D Refinement			
j)	k)		l)	
3D Error (mm):	23.84	3.52		

Fig. 19 Results of isometric SFT for different warp models and reconstruction methods on a real example of a can. *b)*, *c)* and *d)* are the warps obtained with *BS-warp*, *BS-Schwarp* and *BS-Isowarp* respectively; and *f)*, *g)*, *h)*, *i)*, *j)* and *k)* show the 3D reconstruction results of *BS-warp*, *BS-Schwarp*, *BS-Isowarp*, *TPS-Chhatkuli17*, *Salzmann11* and *BS-warp+Ref*. The reconstruction ground truth in *l)* is obtained using dense SfM from multiple images [1].

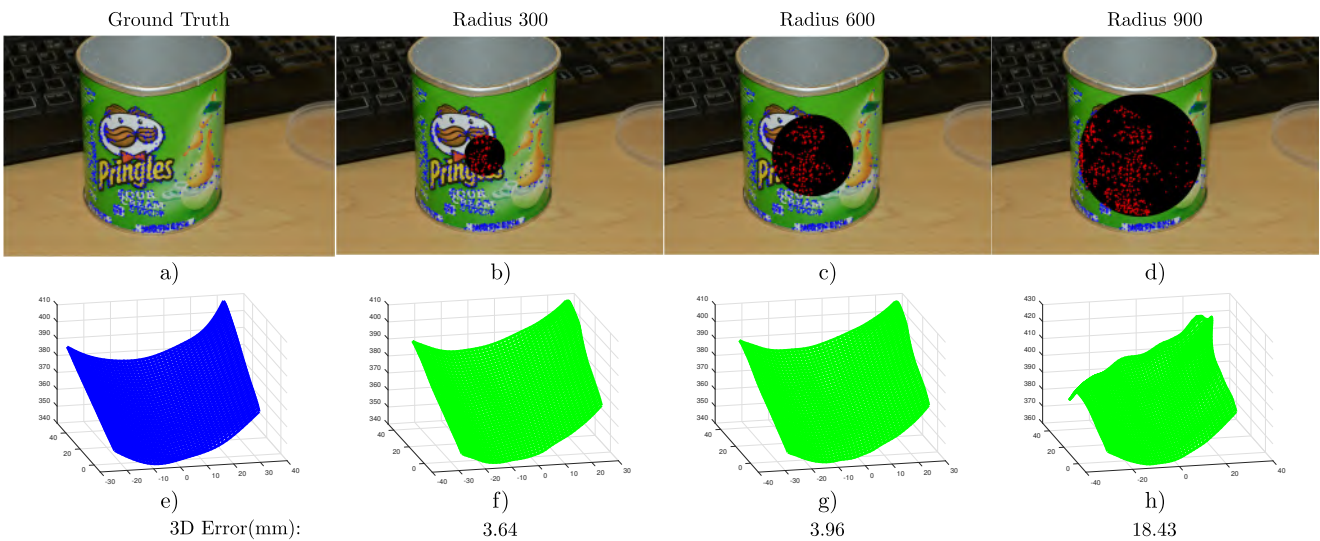


Fig. 20 Robustness of *BS-Isowarp* against occlusions for the Can dataset. *b)*, *c)* and *d)* show the simulated circular occlusion (black hole) corresponding to the radii 300, 600 and 900 pixels. The used features correspondences are shown in blue and the removed ones in red. *f)*, *g)* and *h)* show the 3D reconstruction result of *BS-Isowarp*. The reconstruction ground truth is shown in *e)*.

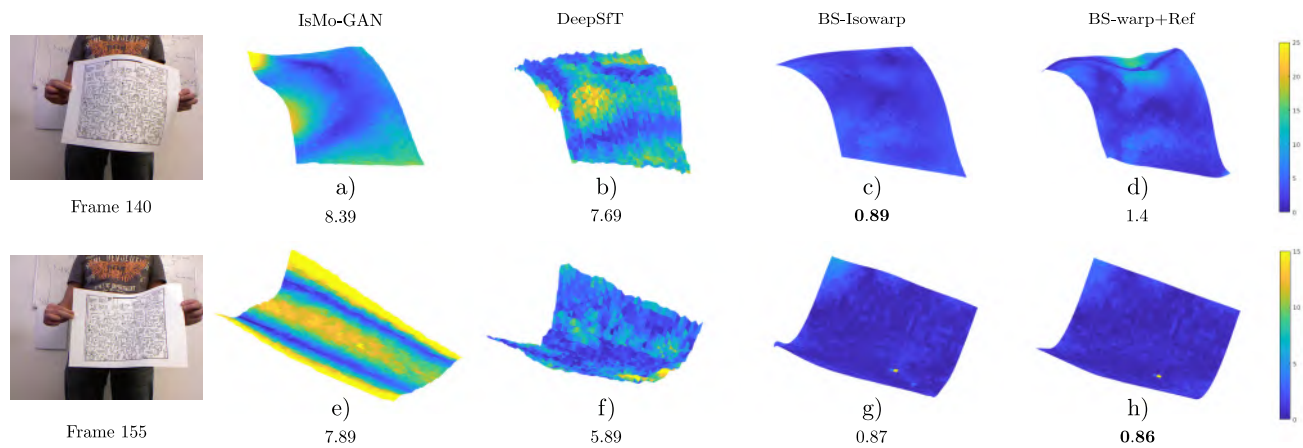


Fig. 21 Frame reconstructed from the Kinect paper dataset. The 3D error is given under each 3D reconstruction in millimeters.