Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences

Toby Collins and Adrien Bartoli

ALCoV-ISIT, UMR 6284 CNRS/Université d'Auvergne, Clermont-Ferrand, France

Abstract. Template-based methods have been successfully used for surface detection and 3D reconstruction from a 2D input image, especially when the surface is known to deform isometrically. However, almost all such methods require that keypoint correspondences be first matched between the template and the input image. Matching thus exists as a current limitation because existing methods are either slow or tend to perform poorly for discontinuous or unsmooth surfaces or deformations. This is partly because the 3D isometric deformation constraint cannot be easily used in the 2D image directly. We propose to resolve that difficulty by detecting incorrect correspondences using the isometry constraint directly in 3D. We do this by embedding a set of putative correspondences in 3D space, by estimating their depth and local 3D orientation in the input image, from local image warps computed quickly and accurately by means of Inverse Composition. We then relax isometry to inextensibility to get a first correct/incorrect classification using simple pairwise constraints. This classification is then efficiently refined using higher-order constraints, which we formulate as the consistency between the correspondences' local 3D geometry. Our algorithm is fast and has only one free parameter governing the precision/recall trade-off. We show experimentally that it significantly outperforms state-of-the-art.

1 Introduction

An open problem in computer vision is to automatically determine correspondences between two images of a deformable 3D surface. Solving this problem is required in several applications, including estimating the nonrigid shape of the surface (known as template-based 3D reconstruction in the literature [2–5]), as a cue for nonrigid object detection [6, 7], and nonrigid Structure-from-Motion [8, 9]. There are several approaches to this problem, and these can be broadly broken into two main axes. In the first axis are the *Graph-Based Assignment* (GBA) methods [10–12]. These solve the problem by constructing graphs that encode the geometric relationship between correspondences. Solving GBA amounts to an NP-hard binary programming problem, and much of the ongoing research focuses on finding efficient and tight relaxations to this problem. In the second axis are the *Hard Matching with Outlier Detection* (HMOD) methods [5, 6, 1, 4, 13, 14]. HMOD methods work by first matching points using local texture information computed from a keypoint descriptor algorithm. Each point

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 325–340, 2014.

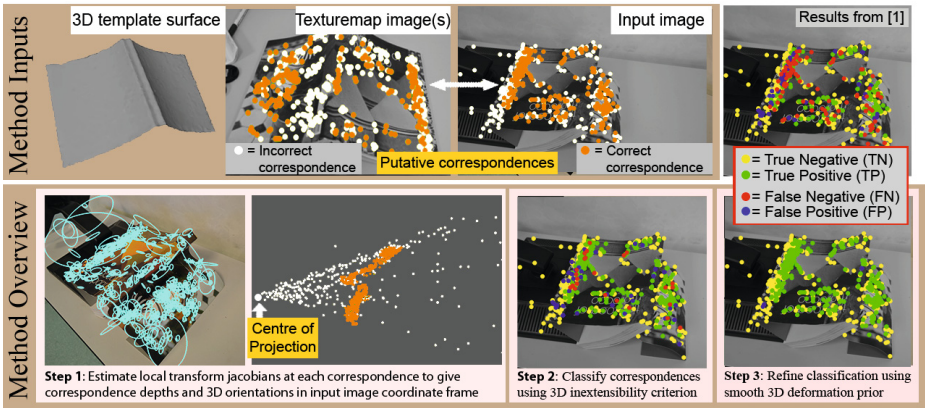


Fig. 1. Summary of the problem tackled and our 3-step solution (example is from our *OpenBook* dataset). As inputs we have 715 2D putative correspondences computed between a 3D template's texturemap and an input image. Of these 429 are correct correspondences (*i.e.* positives) and 286 are incorrect correspondences (*i.e.* negatives). In the bottom-right we show the final output of our method, which correctly classifies 713 correspondences, with 0 false negatives and 2 false positives. In the top-right is the output from [1], which gives 97 false negatives and 85 false positives. Best viewed in colour.

in one image is assigned to the point in the second image with the closest descriptor. Thus, a *hard* correspondence decision is made using only local texture information. A second stage is then performed to determine which correspondences are correct and incorrect by measuring their geometric compatibility via a deformable model. This second stage is sometimes referred to as *outlier detection* in the literature. So far HMOD methods have been preferred over GBA methods for use in template-based 3D reconstruction and nonrigid object detection. The main reason is that they are typically much faster than GBA methods. With efficient implementations the fastest HMOD methods perform in realtime [6] and can handle thousands of feature points, whereas accurate GBA methods are far slower, and may take several minutes to process a few hundred points [10]. Furthermore, unlike HMOD methods, most GBA methods are designed to work when the same features are detected in both images, however this is typically not the case in real conditions with scene clutter or occlusion.

There are three main limitations to state-of-the-art HMOD methods. Firstly, they tend not to be able to handle cases when the number of incorrect correspondences is large (*e.g.* 50% and beyond). This can often occur when dealing with surfaces with poorly discriminative texture, or when the imaging conditions are quite different such as strong lighting change or noise. Secondly, state-of-the-art HMOD methods are either fast, and use a simplified convex model of deformation [13, 6], or use a more realistic physical deformation model, but are either slow to execute and do not scale well to large, complex surfaces with complex topology [5], or cannot handle discontinuous motion such as surface tearing [4]

We present a new HMOD method that does not suffer these limitations and show experimentally that it considerably improves on state-of-the-art (Fig. 1). Our approach is based on using local physical 3D deformation constraints to detect incorrect correspondences. Specifically we use quasi-isometry, which means the amount of stretching induced by the deformation is small. This is a property exhibited by many materials, and which has been exploited before to solve the HMOD problem [5, 4, 15]. However those methods require a costly iterative optimisation process that alternates between registering the surface and detecting incorrect correspondences. We show that the problem can be solved more efficiently using the fact that the deformation of an isometric surface can be locally approximated by smoothly-varying rigid transforms. Our method involves estimating these transforms from the putative correspondences and because it models deformation only locally, and so scales well to large meshes with complex topology, can handle discontinuous surfaces and/or deformation, and is very parallelisable.

2 Previous Work

All prior HMOD methods work by fitting a deformable model using the putative correspondences and detecting incorrect correspondences as those which disagree with the fitted model. The methods differ along two main axes. Along the first axis is the *spatial extent* of the deformable model. *Global methods* work using global deformable models [6, 1, 4, 5] which model the entire deformation of the surface. *Local methods* work by breaking the surface into multiple regions and fitting a local deformable model to each region independently. Along the second axis are *HMOD-3D* and *HMOD-2D* methods which use 3D and 2D deformable models respectively. Previous HMOD-3D methods deform the surface in 3D space using the putative correspondences. Their main advantage is that they can use constraints that have physical meaning which are unaffected by changing the camera viewpoint or camera parameters. All prior HMOD-3D methods are global methods which constrain the surface deformation using isometry [4, 5, 15]. Some of these have proposed detecting incorrect correspondences and fitting the deformable model as a joint optimisation problem [15], however this was very slow and reported to take 15 minutes with examples of only 40 correspondences. Faster HMOD-3D methods work by alternating between registering a mesh of the deforming surface and detecting incorrect correspondences [4, 5]. During optimisation higher confidence is gradually placed on the model's prediction, which leads to more incorrect correspondences being detected. These alternation methods have been shown to work well on very smooth, low complexity surfaces. However the alternation is costly because at each iteration the full 3D shape of the surface is estimated. However these are prohibitively slow to process large, complex 3D surfaces in realtime, and cannot handle discontinuous deformation.

HMOD-2D methods do not model the 3D deformation of the surface. Instead they model the 2D-2D deformation between a single image of the surface

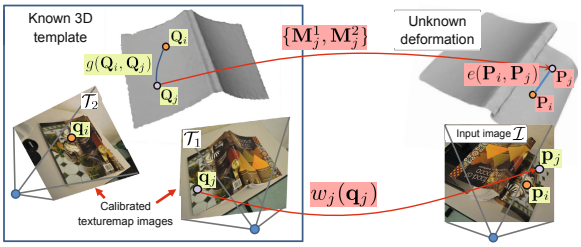
(typically called a *template image*), and the input image. Because they do not involve 3D properties they cannot exploit surface isometry, and must use general assumptions on the 2D-2D flowfield. All prior HMOD-2D methods assume this flowfield is smooth (either globally or piecewise). A global HMOD-2D method was presented in [6] which first proposed the alternation strategy used by [4, 5]. This method is fast but breaks down when the flowfield is discontinuous, which occurs if the surface self-occludes or has sharp edges. Another global method was recently presented [1]. This assumes the 4D correspondence manifold is approximately planar and works by fitting this 4D hyperplane using RANSAC. This works well in some cases, such as simple, smooth bending of paper, but fails for more complex deformations. A local method was presented by [13] which uses affine and low-complexity Thin-Plate Spline (TPS) local models. The method is fast and is highly parallelisable. Because smoothness is assumed only locally, it can handle discontinuous 2D-2D flowfields, however the method does not cope well with correct correspondence ratios below 60%. There are no previous *local HMOD-3D methods*, and our proposed method fills this gap.

3 Problem Setup and Approach Overview

3.1 Problem Setup

Our problem setup is illustrated in Fig. 2. We define a 3D template similarly to the template-based 3D reconstruction literature. The template consists of a 3D mesh model defined in world coordinates which is textured using a set of registered *texturemap images*: $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, .., \mathcal{T}_T\}$. Each $\mathcal{T}_t$ is an RGB photograph of the 3D mesh model with a known pose. We assume the 3D template has been constructed using a 3D acquisition device such as a structured-light scanner with fully-calibrated RGB cameras. We assume the template's silhouette in each $\mathcal{T}_t$ is known, and a set of 2D image features located within the silhouette is provided. We use affine-covariant SIFT features [16] in our experiments but this could be computed with any method. We then perform ray-intersection with the template to compute the 3D positions of the features in world coordinates.

For the input image, 2D features are then computed and putatively matched to the template's features by finding the one with the closest descriptor. An optional step is performed to remove low-confidence correspondences using Lowe's ratio test [17]. We denote the list of 3D-2D putative correspondences between the 3D template and 2D input image with $\mathcal{K} = \{(t_j, \mathbf{q}_j, \mathbf{Q}_j), \mathbf{p}_j\}$. For the j^{th} correspondence, we have a *3D template feature*, denoted by $(t_j, \mathbf{q}_j, \mathbf{Q}_j)$ and a *2D input image feature* $\mathbf{p}_j$. $t_j \in \{1..T\}$ holds the index of the texturemap image from which the 3D template feature was detected, $\mathbf{q}_j \in \mathbb{R}^2$ holds its 2D position in the texturemap image and $\mathbf{Q}_j \in \mathbb{R}^3$ holds its 3D position in world coordinates. $\mathbf{p}_j \in \mathbb{R}^2$ holds the 2D position of the corresponding input image feature. We assume the input image's camera is intrinsically calibrated, and define $\mathbf{p}_j$ in normalised pixel coordinates. The unknown 3D position of $\mathbf{p}_j$ in camera coordinates is denoted by $\mathbf{P}_j \in \mathbb{R}^3$ where $\mathbf{p}_j = \pi(\mathbf{P}_j) + \varepsilon$, where ε denotes



Terminology

Label in $\mathbf{L}$	0	1
Class	Negative	Positive
Correspondence	Incorrect	Correct

Fig. 2. Problem setup illustrated with two putative correspondences. Terms in green and red indicate known and unknown quantities respectively.

measurement noise and $\pi([x, y, z]^\top) \stackrel{\text{def}}{=} \frac{1}{z}[x, y]^\top$ is the normalised perspective projection function.

Our goal is to classify which members of $\mathcal{K}$ are correct and which are incorrect correspondences. We define the positive class to be correct correspondences and the negative class to be incorrect correspondences. The problem is posed as finding the binary label vector $\mathbf{L} \in \{0, 1\}^n$, $n \stackrel{\text{def}}{=} |\mathcal{K}|$ where $\mathbf{L}(j) = 1$ means the j^{th} correspondence is classified positive and $\mathbf{L}(j) = 0$ means it is classified negative (Fig. 2).

3.2 Approach Overview

Our method involves determining $\mathbf{L}$ efficiently using *local* 3D deformation models. We use the fact that for isometric surfaces 3D deformation can be locally approximated by smoothly-varying rigid transforms. The method is broken down into three core steps (Fig. 1). In the first step we take each putative correspondence in $\mathcal{K}$ and *upgrade* it to a 3D-3D correspondence. This is done by estimating the local transform induced by the correspondence, and then inferring the depth of the correspondence in the camera coordinate frame, using a very fast solution inspired by [18]. The transforms are initialised using Affine Covariant Normalisation (ACN) [19], then efficiently refined with Inverse-Compositional iterations [20, 21]. In the second stage we use the 3D-3D correspondences to construct a graph that encodes pairwise inextensibility (in 3D space). Inextensibility is a relaxation of isometry, which says that the Euclidean distance between any two points on an isometric surface should not exceed their geodesic distance (which is known *a priori* from the template). We use this constraint to find an initial labelling $\mathbf{L}_0$ with an approach inspired by [14]. In the third step we refine $\mathbf{L}_0$ with a fast iterative approach by introducing local models with higher-order constraints. Specifically, we enforce that the deformation can be modelled by local, smoothly varying rigid transforms, and estimate these transforms robustly whilst refining $\mathbf{L}$. In practice only a few refinement iterations are needed.

4 Steps 1 and 2: Computing High-Confidence Labels Using Inextensibility in 3D

We show how inextensibility can be used to efficiently upgrade 3D-2D correspondences to 3D-3D correspondences (Step 1). Then we show how to classify correspondences using 3D inextensibility (Step 2).

4.1 Step 1: Upgrading to 3D-3D Correspondences

Principle. We upgrade each correspondence using the constraints that isometry imposes on the local 2D transformation between the template's texturemap image and the input image. This approach is inspired by [18, 22] where it was shown that depth information can be recovered analytically from this transform. Those methods assume that the deformable template and input images are already registered, which was achieved by conformally flattening the template and computing a *global* warp between the flat template and the input image. In our problem we do not know this warp (because knowing it would mean knowing $\mathbf{L}$). Furthermore we want to be able to compute depths for templates with arbitrary topology (including non-flattenable templates). Our solution is to fit a localised warp, but for each correspondence *individually*.

For the j^{th} correspondence $((t_j, \mathbf{Q}_j, \mathbf{q}_j), \mathbf{p}_j) \in \mathcal{K}$, we compute a local warp $w_j : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that transforms the 2D point $\mathbf{q}_j$ in $\mathcal{T}_{t_j}$ to the 2D point $\mathbf{p}_j$ in the input image $\mathcal{I}$ (Fig. 2). Once estimated, by measuring the Jacobian $J_{w_j}(\mathbf{q}_j) \in \mathbb{R}^{2\times 2}$ of the warp, we can compute the depth $z_j \in \mathbb{R}^+$ of $\mathbf{p}_j$ with respect to the input image's camera [18, 22]. Thus we are able to *upgrade* the 3D-2D correspondence to a 3D-3D correspondence, which we denote by the pair $(\mathbf{Q}_j, \mathbf{P}_j)$ with $\mathbf{P}_j \stackrel{\text{def}}{=} z_j[\mathbf{p}_j^\top, 1]^\top$. In addition to $\mathbf{P}_j$, the analytic solution also provides us with two estimates of the rotation matrix that rotates $\mathbf{Q}_j$ to $\mathbf{P}_j$ [23]. This means we have for each correspondence two estimates of the local rigid transform from $\mathbf{Q}_j$ to $\mathbf{P}_j$. We denote these by $\mathcal{M}_j = \{\mathbf{M}_j^1, \mathbf{M}_j^2\}$, $\mathbf{M}_j^s \in SE_3$. We now address the question of how to fit the local warps in order to compute $J_{w_j}(\mathbf{q}_j)$, and hence compute z_j and $\mathcal{M}_j$.

Computing the warp Jacobians. Our approach to compute $J_{w_j}(\mathbf{q}_j)$ is to fit the warp using pixel intensity information surrounding $\mathbf{q}_j$ and $\mathbf{p}_j$. This is summarised in two steps:

1. **Coarse approximation**. We first compute a coarse estimate of $J_{w_j}(\mathbf{q}_j)$ using ACN. By using a feature matching algorithm that performs ACN as part of descriptor extraction then this step is done for us and so is at no additional cost. In our experiments we use VLFeat's `vl_covdet`.
2. **Direct refinement with a local warp**. We then construct a low-complexity 2D-2D parametric warp centred at each $\mathbf{q}_j$. The warp is initialised using the affine transform from Step 1 and refined efficiently by IC iterations [20, 21].

For Step 2 it is important to use low complexity warps. This is necessary to prevent overfitting, improve convergence and to reduce computation time. We

have found good results can be achieved using a TPS warp with only four control points. We define a circular support region centred at $\mathbf{q}_j$ of radius r_j. There is a trade-off in choosing r_j. Too small, and the region may contain insufficient image structure with which to estimate $J_w(\mathbf{q}_j)$. Too large and the motion in the region may be too complex to describe with a simple model. A strategy for selecting r_j is to use the characteristic scale of the feature at $\mathbf{q}_j$. The characteristic scale gives the size of the image region surrounding $\mathbf{q}_j$ with which its descriptor was computed. Because feature descriptors do not normally provide invariance beyond very simple transforms (at most affine transforms), a correct putative correspondence implies the image transform at this scale must be simple. Furthermore if the correspondence is correct then the characteristic scale is large enough to encompass sufficient discriminative image structure, which usually implies there is enough structure with which to estimate $J_{w_j}(\mathbf{q}_j)$. We optimise the TPS parameters using IC iterations, which are extremely fast, using a centre-weighted Normalised Sum-of-Square Difference (NSSD) data cost. We provide implementation details for this optimisation in the supplementary material. After optimisation, we compute $J_{w_j}(\mathbf{q}_j)$ by differentiating the local warp at $\mathbf{q}_j$, from which we compute $\mathbf{P}_j$ and the local rigid transforms $\mathcal{M}_j$.

4.2 Step 2: Classifying Correspondences Using Pairwise Inextensibility

We now use the upgraded 3D-3D correspondences to efficiently gain an initial correspondence labelling $\mathbf{L}_0 \in \{0,1\}^n$ using pairwise *3D* inextensibility constraints (Fig. 2). The approach is inspired by [14], however it it different because in [14] *2D* inextensibility is enforced. The latter can be violated between two correct correspondences when *e.g.* viewing the surface at different depths, different orientations, using different image resolutions or using different focal lengths. By contrast for isometric surfaces, *3D* inextensibility is never violated, and is totally independent of the imaging conditions. We use $g(\mathbf{Q}_i, \mathbf{Q}_j)$ to denote the geodesic distance between points $\mathbf{Q}_i$ and $\mathbf{Q}_j$, and $e(\mathbf{P}_i, \mathbf{P}_j) \stackrel{\text{def}}{=} \|\mathbf{P}_i - \mathbf{P}_j\|_2$ to denote the Euclidean distance between $\mathbf{P}_i$ and $\mathbf{P}_j$. $g(\mathbf{Q}_i, \mathbf{Q}_j)$ can be pre-computed efficiently offline when the 3D template was built, and the online cost of evaluating it is negligible. If i and j are correct correspondences, then in the absence of noise $g(\mathbf{Q}_i, \mathbf{Q}_j) \geq e(\mathbf{P}_i, \mathbf{P}_j)$. This is a relaxation of the isometric constraint $g(\mathbf{Q}_i, \mathbf{Q}_j) = g(\mathbf{P}_i, \mathbf{P}_j)$, which we cannot apply because we do not have measurements of $g(\mathbf{P}_i, \mathbf{P}_j)$. The relaxation is however still powerful because if j is an incorrect correspondence then $\mathbf{P}_j$ tends to be distributed very randomly within the camera's frustum. This is illustrated in Fig. 1 (Step 1). The randomisation of the incorrect correspondences means that when either i, j, or both i and j are incorrect, often $e(\mathbf{P}_i, \mathbf{P}_j)$ will exceed $g(\mathbf{Q}_i, \mathbf{Q}_j)$, and this tells us correspondences i and j are not geometrically compatible. We define a pairwise binary compatibility matrix as follows:

$$\mathbf{U}(i,j) = \begin{cases} 1 & \text{if } g(\mathbf{Q}_i, \mathbf{Q}_j) \geq e(\mathbf{P}_i, \mathbf{P}_j) - \tau_e \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

This compatibility score is more discriminative when $\mathbf{Q}_i$ and $\mathbf{Q}_j$ are close because when $g(\mathbf{Q}_i, \mathbf{Q}_j)$ is smaller the bound on $e(\mathbf{P}_i, \mathbf{P}_j)$ is tighter. τ_e is a tolerance

term used to handle uncertainty caused by the fact that the local warps will have some noise, and hence induce noise in $\mathbf{P}_i$ and $\mathbf{P}_j$. To select τ_e recall that the template has been normalised to fit within the unit cube, and so τ_e does not need to be adapted depending on the template's size. We have found $\tau_e = 5\%$ to work well across all our experiments.

$\mathbf{U}$ can be interpreted as a graph with n nodes, where each node is a correspondence and an edge appears between a pair of nodes if they respect the inextensibility constraint. The set of correct correspondences should therefore form a strongly-connected component in the graph, and so we can estimate $\mathbf{L}$ by establishing which nodes belong to this component. We do this in a similar manner to [14], but because $\mathbf{U}$ is binary the selection process can be simplified because we do not need the eigendecomposition of $\mathbf{U}$. Let $\mathbf{m}_i \in \{0,1\}^n$ denote the i^{th} row of $\mathbf{U}$. First two empty sets are constructed; a set $\mathcal{P} = \emptyset$ holding all positives, and a set $\mathcal{N} = \emptyset$ holding all negatives. We then find $i^* = \arg\max_i \sum_j \mathbf{m}_i(j)$ (*i.e.* the best-connected correspondence) and insert i^* into $\mathcal{P}$. We then find the correspondence which has not yet been classified that has the highest number of connections: $i^* = \arg\max_{i \notin \mathcal{P} \cup \mathcal{N}} \sum_j \mathbf{m}_i(j)$. We test whether i^* is geometrically compatible with $\mathcal{P}$ by computing the compatibility score:

$$c(i^*, \mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} \mathbf{U}(i^*, j) \tag{2}$$

This gives the proportion of members of $\mathcal{P}$ that are geometrically compatible with i^*. We insert i^* into $\mathcal{P}$ if $c(i^*, \mathcal{P}) > \tau_c$, otherwise it is inserted into $\mathcal{N}$. We use $\tau_c = 90\%$, which provides robustness during selection if $\mathcal{P}$ contains some incorrect correspondences. This selection process continues until all correspondences have been assigned to $\mathcal{P}$ or $\mathcal{N}$. We then initialise $\mathbf{L}$ with $\mathbf{L}_0(k) = \mathbb{1}(j \in \mathcal{P})$, where $\mathbb{1}(\cdot)$ denotes the indicator function.

5 Step 3: Fast Label Refinement with Higher-Order Constraints

$\mathbf{L}_0$ serves as an initial classification, but it may contain errors. An example of these errors is shown in Fig. 1 (Step 2). By using a high value of $\tau_c = 90\%$ the number of false positives is usually low. False negatives mainly occur when the local warp of a correspondence fails to converge to the right solution, which can lead to a poor estimate of $\mathbf{P}_j$. The main reasons for this are when *(i)* there is a photoconstancy violation in the local warp's region (such as a specularity) or when *(ii)* the warp's region crosses a discontinuity.

Our classification refinement method is based on the fact that if there is a neighbouring correspondence i which is correct, from $\mathcal{M}_i$ we have two estimates for the local transform that maps the template at point $\mathbf{Q}_i$ to the input image. We can therefore use these to validate whether j is a correct or incorrect correspondence. Assuming rigidity holds locally at i and j, if either $\pi\left(\mathbf{M}_i^1[\mathbf{Q}_j^\top, 1]^\top\right)$ or $\pi\left(\mathbf{M}_i^2[\mathbf{Q}_j^\top, 1]^\top\right)$ is close to $\mathbf{p}_j$, then j is likely to be a correct correspondence. Otherwise j is likely to be an incorrect correspondence. One challenge with doing

this is that we do not know if i is a correct correspondence. Our solution is to use i if it has been classified positive in $\mathbf{L}_0$, but in a way that is robust to false positives.

For each j we construct a list of positives $\mathcal{S}_j = \{i \in 1..n, \mathbf{L}_0(i) = 1, i \neq j\}$. All members of $\mathcal{S}_j$ then vote for the predicted position of $\mathbf{p}_j$. First, for each i we find the rigid transform in $\mathcal{M}_i$ that agrees with $\mathbf{p}_j$ the most. We define this by $\mathbf{M}_i^*(j)$:

$$\mathbf{M}_i^*(j) = \arg\min_{\mathbf{M} \in \mathcal{M}_i} \left\| \pi \left(\mathbf{M} [\mathbf{Q}_j^\top, 1]^\top \right) - \mathbf{p}_j \right\|_2^2 \tag{3}$$

We then compute a robust prediction $\hat{\mathbf{p}}_j$ for $\mathbf{p}_j$. We do this using a weighted median of the individual predictions in $\mathcal{S}_j$ in a neighbourhood of size σ_j:

$$\begin{aligned} \hat{\mathbf{p}}_j &= \operatorname*{wmed}_{i \in \mathcal{S}_j} \left\{ \pi \left(\mathbf{M}_i^*(j) [\mathbf{Q}_j^\top, 1]^\top \right), v_i^j(\sigma_j) \right\} \\ v_i^j &\stackrel{\text{def}}{=} \begin{cases} \exp(-g(\mathbf{Q}_j, \mathbf{Q}_i)^2 / \sigma_j^2) & \text{if } \|\mathbf{Q}_j - \mathbf{Q}_i\|_2 < 3\sigma_j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

v_i^j is a weight function which gives more influence to i if it is close to j. We use a truncated Gaussian for this, which means only a subset of nearby correspondences are used to compute $\hat{\mathbf{p}}_j$ (and thus improve efficiency). The weighted median provides robustness if $\mathcal{S}_j$ has false positives. It also provides robustness if local rigidity holds for some, but not all members of $\mathcal{S}_j$. We then reclassify j according to $\mathbf{L}(j) \leftarrow \mathbb{1}(\|\hat{\mathbf{p}}_j - \mathbf{p}_j\|_2 < \tau_p)$.

The free parameter τ_p governs the degree to which $\hat{\mathbf{p}}_j$ must agree with $\mathbf{p}_j$ for us to classify j as a positive. Thus τ_p provides a recall/precision tradeoff, with a lower τ_p meaning fewer false positives but potentially more false negatives. τ_p is a free parameter which can be set according to the application. We have found a good default value to be $\tau_p = 2\%$.

The weight function's bandwidth is given by σ_j. This should be adapted to reflect the extent of rigidity of the deformation about $\mathbf{Q}_j$. We automatically adapt σ_j using $\mathbf{L}_0$ and a fast minimisation of the prediction error. Specifically, we compute:

$$\sigma_j = \arg\min_{\sigma} \|\hat{\mathbf{p}}_j(\sigma) - \mathbf{p}_j\|_2^2 \tag{5}$$

where $\hat{\mathbf{p}}_j(\sigma)$ denotes the dependency of $\hat{\mathbf{p}}_j$ on σ. We solve Eq. (5) by quantising σ_j in 10 levels in the range 1% to 30%, and using the one that minimises Eq. (5).

We have found that $\mathbf{L}$ can usually be improved further by performing a few reclassification iterations. The algorithm pseudocode is simple and presented in Table 1, Step 3. There are a few cases which must be handled. The first is if j began negative but was reclassified positive. If this occurs it is likely that its warp was not estimated correctly in Step 2, leading to a poor estimate of its 3D position and orientation. We recompute this 3D information using neighbouring positives and a Pose from n Points (PnP) computation. Specifically we take a neighbour i if it is a member of $\mathcal{S}_j$ and $\|\pi \left(\mathbf{M}_i^*(j) [\mathbf{Q}_j^\top, 1]^\top \right) - \mathbf{p}_j\|_2 < \tau_p$ (*i.e.* its transform can predict well the position of j in the image). If there are more than two such neighbours, we recompute $\mathcal{M}_j$ by performing PnP using the correspondences from j and these neighbours. To perform PnP we use RPnP

Table 1. Classifying correct/incorrect 3D-2D correspondences using isometry: algorithm summary

Inputs (§3.1)
- Putative 3D-2D correspondences $\mathcal{K} = \{(t_j, \mathbf{Q}_j, \mathbf{q}_j), \mathbf{p}_j\}$
- Recall/precision threshold τ_p (default to 2% of the image diagonal)

Step 1: Upgrade $\mathcal{K}$ to 3D-3D correspondences (§4.1)
1. For $j = 1 \rightarrow |\mathcal{K}|$ use IC iterations to compute local warp w_j that transforms $\mathbf{q}_j$ to $\mathbf{p}_j$
2. Use $J_{w_j}(\mathbf{q}_j)$ to estimate $\mathbf{P}_j$ and local rigid transforms $\mathcal{M}_j$ mapping $\mathbf{Q}_j$ to $\mathbf{P}_j$

Step 2: Initialise $\mathbf{L}_0$ using 3D-3D pairwise inextensibility constraints (§4.2)
1. Construct compatibility matrix $\mathbf{U} \in \{0,1\}^{|\mathcal{K}|\times|\mathcal{K}|}$
2. Compute $\mathbf{L}_0$ from $\mathbf{U}$ with greedy selection process

Step 3: Refine $\mathbf{L}_0$ using higher-order constraints (§5)
1. $\mathbf{L} \leftarrow \mathbf{L}_0$
2. While $\mathbf{L}$ changes or 10 iterations have not passed **do**
3. For $j = 1 \rightarrow |\mathcal{K}|$
4. Compute $\mathcal{S}_j$, σ_j, $\{w_i^j\}$ and $\hat{\mathbf{p}}_j$ (Eq. (4,5))
5. $\mathbf{L}'(j) \leftarrow \begin{cases} 0 & |\mathcal{S}_j| = 0 \\ \mathbb{1}(\|\hat{\mathbf{p}}_j - \mathbf{p}_j\|_2 < \tau_p) & \text{otherwise} \end{cases}$
6. $\mathbf{L} \leftarrow \mathbf{L}'$

Output class vector $\mathbf{L} \in \{0,1\}^{|\mathcal{K}|}$

[24], and put into $\mathcal{M}_j$ all rigid poses returned by RPnP. We use RPnP because it is fast and can handle cases when the problem is ambiguous (which is often the case when doing local PnP [25]). A second case that must be handled is when $\mathcal{S}_j$ is empty. This occurs when all correspondences excluding j are negative. In practice this only usually happens when the template is not visible in the input image. Thus if $\mathcal{S}_j$ is empty we conclude the template is not visible and set $\mathbf{L}(j) \leftarrow 0$.

6 Experimental Results

We present a range of experiments to compare the performance of our method against state-of-the-art. We compare against [13, 6, 1, 5], which we refer to by Piz-IJCV12, Pil-IJCV08, Tran-ECCV12, and Salz-CVPR09 respectively. We use the authors' original code for Piz-IJCV12, Tran-ECCV12 and Salz-CVPR09, and the implementation of Pil-IJCV08 from [1].

Obtaining ground truth. There are several existing datasets for deformable isometric surfaces (e.g. [26, 27]). However these do not include ground-truth correspondences and are generally quite simple and involve developable surfaces such as sheets of paper or cloth. We have created three new real ground-truth datasets involving more complex surfaces and deformations. Computing dense ground truth correspondences for deforming surfaces is notoriously difficult and tedious [28]. Our approach was based on the idea that although the 3D-2D non-rigid registration problem is hard, when the surface is isometric, registering two

deformed surfaces in 3D is far simpler and can be done automatically or semi-automatically [29, 30]. We captured a test surface in several deformed states and performed dense multiview Structure-from-Motion to obtain a texturemapped 3D template for each deformed state, and the camera parameters for each image. We then semi-automatically co-registered the 3D templates with a method based on [29] to provide us with dense correspondence between the 3D templates, and hence dense registration between different images of the surface in different deformed states.

The OpenBook dataset. The OpenBook dataset comprises four deformed states ($\mathcal{S}_1 \rightarrow \mathcal{S}_4$) of a book cover (Fig. 3 (top row)), with 14 images taken for each deformed state. Images were captured with a standard 1020p point-and-shoot camera and we used Agisoft's Photoscan to perform dense multiview reconstruction. We use $\mathcal{C}_i$ to denote the set of images capturing the i^{th} deformed state. A selection of images from $\mathcal{C}_3$ are shown in Fig. 3 (second row). We then used each state in turn as the 3D template, and used all images for all other deformed states as input images. Thus in this dataset there is a total of $4 \times 3 \times 14 = 168$ different template/input image pairs. To allow a comparison between our method and the HMOD-2D methods we used only features detected in one of the template's texturemap images. This is because the HMOD-2D methods cannot trivially handle features coming from different texturemap images.

We use affine-covariant features using *VLFeat*'s implementation with default parameters. Putative correspondences were found using a Lowe ratio threshold of 1.1 [31]. Typically this resulted in between 200-800 putative correspondences per input image. Correspondences which were within 10 pixels of their ground truth positions were marked as true correspondences, and the rest as false. The proportion of incorrect correspondences in each image had a mean of 62%. For all methods we generated ROC curves by varying each method's detection threshold (we use the same procedure as [1] to do this). For our method the detection threshold is governed by τ_p (§5), with a default of 2.0% of the input image's diagonal). In Fig. 3 (third row) we show the ROC curves, with one ROC curve generated for each deformed state. We can see that our method performs significantly better than all others. At a false negative rate of 4.5% our method successfully classified all incorrect correspondences. The worst performing method is Salz-CVPR09. The reason is because it often eliminates many correct correspondences early in the annealing stage and cannot recover in later iterations. In the fourth row of Fig. 3 we show how the previous methods typically fail. Piz-IJCV12 fails if there is a small number of correct correspondences within each correspondence's neighbourhood. When this occurs a good local 2D model cannot be found, and this leads to false negatives. Tran-ECCV12 fails in general when the image transform is not simple and globally smooth. Pil-IJCV08 also fails when the image transform is not globally smooth. Salz-CVPR09 fails systematically when the incorrect correspondence ratio is beyond approximately 40%. Note that our method can correctly handle correspondences on the book's spine, which proved difficult for the other methods.

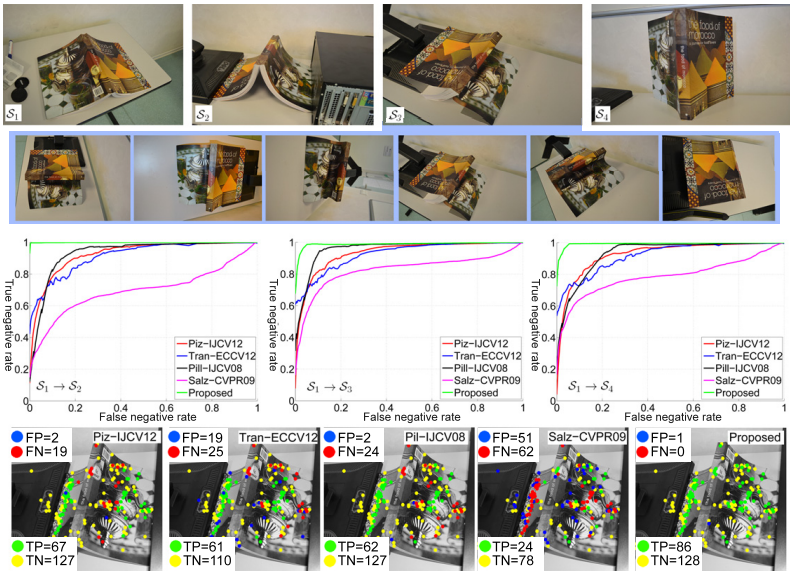
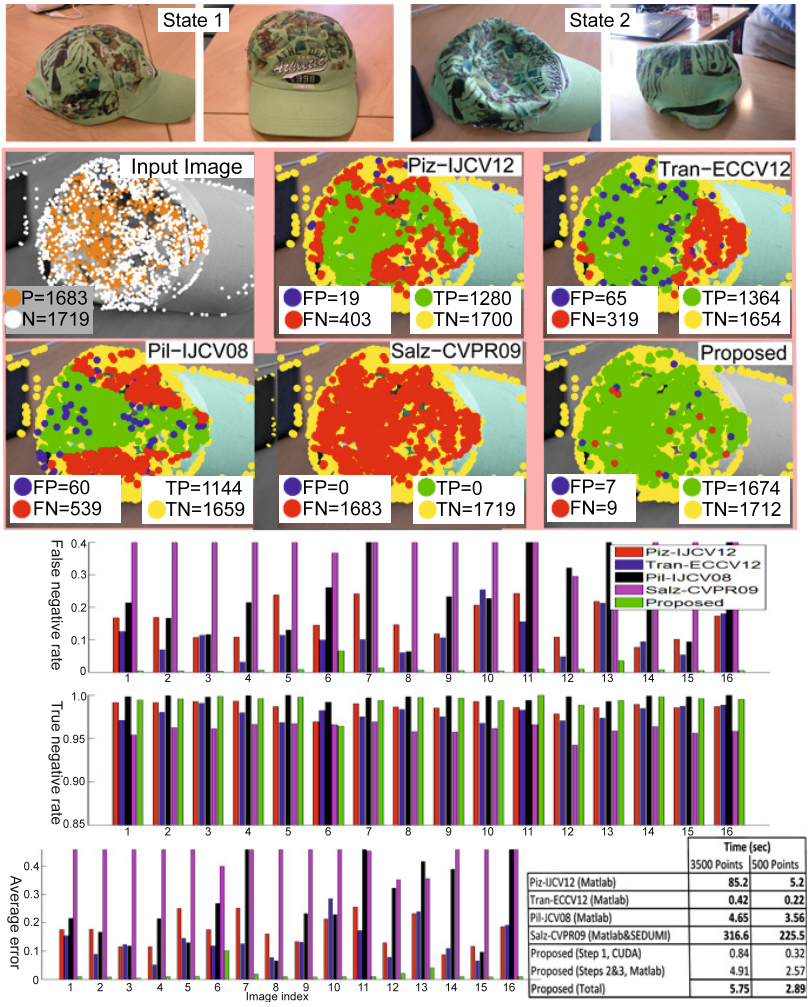


Fig. 3. Results on the OpenBook dataset. There are 86 true and 129 false putative correspondences in the example in the third row. FP and FN denote the number of false positives and false negatives for each method. Best viewed in colour.

The ALCoV Baseball Cap dataset [22]. This dataset consists of two image sets of a baseball cap in two deformed states $\mathcal{S}_1$ and $\mathcal{S}_2$. $\mathcal{C}_1$ and $\mathcal{C}_2$ are of sizes 29 and 16 respectively. We show sample images from $\mathcal{C}_1$ and $\mathcal{C}_2$ in the first row of Fig. 4. The dataset is challenging due to the texture on the cap being repetitive and there being considerable change in illumination. Between 488 and 1,404 affine covariant SIFT features were detected in these images. We used $\mathcal{C}_1$ to build the 3D template, which consists of 12,205 vertices, and used each image in $\mathcal{C}_2$ as an input image. We compared all methods using their default values for detection precision. The results are summarised in the three graphs in Fig. 4 (bottom) showing false negative, true negative and average errors across all 16 input images. We can see that our method performs vastly better than all others in terms of false negative rate, with a mean value of just 1.19%. The true negative rate for our method was joint highest with Pil-IJCV08, however Pil-IJCV08 gives many more false negatives because its deformation model cannot suitably handle the 2D flowfield induced by the cap's deformation, which causes many false negatives. The second and third rows of Fig. 4 show the results on a typical input image from $\mathcal{C}_2$. We present timing information of the methods in Fig. 4 (bottom-right). Note that the implementations are sub-optimal non-parallelised Matlab implementations, and considerable speedups could be made with optimised code. We fully expect our method to be realtime on a standard PC with a good C++/GPGPU implementation.



	Time (sec)	
	3500 Points	500 Points
Piz-IJCV12 (Matlab)	85.2	5.2
Tran-ECCV12 (Matlab)	0.42	0.22
Pil-JCV08 (Matlab)	4.65	3.56
Salz-CVPR09 (Matlab&SEDUMI)	316.6	225.5
Proposed (Step 1, CUDA)	0.84	0.32
Proposed (Steps 2&3, Matlab)	4.91	2.57
Proposed (Total)	5.75	2.89

Fig. 4. Results on the ALCoV Baseball Cap dataset. P and N denote the number of positives and negatives. FP and FN denote the number of false positives and false negatives for each method. Timing information is shown in bottom-right. Best viewed in colour.

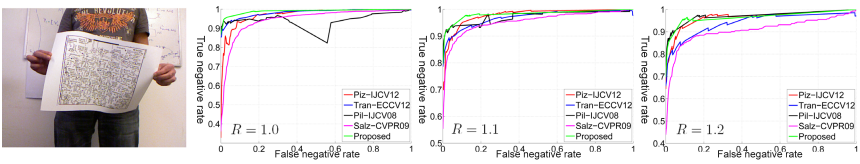


Fig. 5. Results on the CVLAB Bending Paper dataset. Best viewed in colour.

The CVLAB Bending Paper dataset [26]. This dataset is a short video of a deforming sheet of paper lasting 193 frames. Fig. 5 (top left) shows one example frame. The deformation of the paper is very low-frequency, and so we would expect Pil-IJCV08 and Tran-ECCV12 to work well. We used the 3D template that comes with this dataset, which has one texturemap image. We used affine-covariant SIFT features and ran three tests by varying Lowe's ratio threshold using values of $R = 1.0$, $R = 1.1$ and $R = 1.2$. When $R = 1.0$ it means that all putatives are kept (*i.e.* each feature in the input images has a putative correspondence with a feature in the template image). The incorrect correspondence ratios for $R = 1.0$, $R = 1.1$ and $R = 1.2$ (averaged over the whole sequence) are 77.1%, 32.5% and 12.3% respectively. The average number of correct correspondences per frame are 481, 421, 390 respectively. We computed three ROC curve for each R (Fig. 5). The performance difference of our method with respect to Pil-IJCV08 is smaller than the previous datasets, which is expected given the dataset's simple deformation for which Pil-IJCV08 is designed for.

7 Conclusion and Future Work

We have presented a new method to classify correct and incorrect correspondences between a 3D template and a 2D input image of a deformable surface. Our method exploits isometry in an efficient manner. The key to the method's success is turning the putative 3D-2D correspondences to 3D-3D correspondences, and doing this for each correspondence *individually*. This gets us in the position where we can apply 3D inextensibility to obtain an initial classification. This classification is then refined quickly using higher-order geometric consistency between correspondences, which is based on robustly modelling the 3D deformation by smoothly varying rigid transforms. The approach has several advantages. It is very fast because it only uses local estimates of deformation (unlike [5, 4]), can handle discontinuous surfaces and/or deformations, and it has only one important tuning parameter that governs recall and precision, and whose default value of $\tau_p = 2\%$ of the image diagonal gives close to optimal results. We have shown that it significantly outperforms existing methods on more challenging real image datasets with ground truth. We will turn our existing Matlab implementation (which takes a few seconds to run), into a realtime C++/GPGPU implementation and we believe that our algorithm will broaden the use of template-based 3D reconstruction methods. We will be testing new applications of those in our future research.

Acknowledgments. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

1. Tran, Q.-H., Chin, T.-J., Carneiro, G., Brown, M.S., Suter, D.: In defence of ransac for outlier rejection in deformable registration. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 274–287. Springer, Heidelberg (2012)

2. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T.: On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In: CVPR (2012)
3. Salzmann, M., Fua, P.: Linear local models for monocular reconstruction of deformable surfaces. PAMI 33, 931–944 (2011)
4. Östlund, J., Varol, A., Ngo, D.T., Fua, P.: Laplacian meshes for monocular 3D shape recovery. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 412–425. Springer, Heidelberg (2012)
5. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: A convex formulation. In: CVPR (2009)
6. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. IJCV (2008)
7. Alcantarilla, P.F., Bartoli, A.: Deformable 3D reconstruction with an object database. In: BMVC (2012)
8. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI 30 (2008)
9. Taylor, J., Jepson, A.D., Kutulakos, K.N.: Non-rigid structure from locally-rigid motion. In: CVPR (2010)
10. Zhou, F., De la Torre, F.: Deformable graph matching. In: CVPR (2013)
11. Duchenne, O., Bach, F.R., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: CVPR (2009)
12. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 596–609. Springer, Heidelberg (2008)
13. Pizarro, D., Bartoli, A.: Feature-based deformable surface detection with self-occlusion reasoning. IJCV (2012)
14. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV (2005)
15. Shaji, A., Varol, A., Torresani, L., Fua, P.: Simultaneous point matching and 3D deformable surface reconstruction. In: CVPR (2010)
16. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60, 63–86 (2004)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
18. Bartoli, A., Gerard, Y., Chadebecq, F., Collins, T.: On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In: CVPR (2012)
19. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV (2005)
20. Matthews, I., Baker, S.: Active appearance models revisited. IJCV (2004)
21. Brunet, F., Gay-Bellile, V., Bartoli, A., Navab, N., Malgouyres, R.: Feature-driven direct non-rigid image registration. IJCV (2011)
22. Bartoli, A., Collins, T.: Template-based isometric deformable 3D reconstruction with sampling-based focal length self-calibration. In: CVPR (2013)
23. Collins, T., Bartoli, A.: Infinitesimal plane-based pose estimation. IJCV (July 2014)
24. Li, S., Xu, C., Xie, M.: A robust O(n) solution to the perspective N point problem. PAMI (2012)

25. Schweighofer, G., Pinz, A.: Robust pose estimation from a planar target. Pattern Analysis and Machine Intelligence (PAMI) 28, 2024–2030 (2006)
26. Varol, A., Salzmann, M., Fua, P., Urtasun, R.: A constrained latent variable model. In: CVPR (2012)
27. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. In: ICCV (2007)
28. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. IJCV (2011)
29. Huang, Q.X., Adams, B., Wicke, M., Guibas, L.J.: Non-rigid registration under isometric deformations. Comput. Graph. Forum (2008)
30. Tam, G.K.L.: quan Cheng, Z., kun Lai, Y., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., fang Sun, X., Rosin, P.L.: Registration of 3D point clouds and meshes: A survey from rigid to non-rigid. Visualization and Computer Graphics (2013)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)