Fast Shape-from-Template Using Local Features

Mahmoud Famouri[a], Adrien Bartoli[b], Zohreh Azimifar[a,*]

[a]*School of Electrical and Computer Engineering, Shiraz University.*
[b]*Institut Pascal, University Clermont Auvergne / CNRS / SIGMA*

Abstract

Reconstructing the 3D surface of an object using only a single image is a challenging task, which has recently attracted attention. In this paper, a template-based approach is presented to reconstruct the surface of an isometric deformable object. The proposed approach brings a solution for a class of computer vision problems named Shape-from-Template (SfT). In SfT, the goal is to solve single-image reconstruction for an object given its 3D template model in some rest shape. To this end, corresponding keypoints between the template and the so-called deformed image are first established. Then, a very fast method is used to estimate the first-order differential flow around the extracted keypoint-pairs as an affine transformation. This is done using the keypoint-pairs' surrounding texture patch. In our method, we estimate this affine transformation using the keypoint pairs' closest neighbors. This is both faster and more stable. Finally, the depth of each keypoint in the deformed image is estimated from its associated affine transformation. The robustness of keypoint matching is essential to the process. Indeed, outliers defeat depth estimation dramatically. We propose two new approaches to detect and remove the possible outliers based on geometrical properties of the matched keypoints. These two geometrical outlier removal approaches are faster than existing ones and can be used with almost any image descriptor. Experimental results show that the proposed approaches are very effective and outperform existing ones.

Key words: Fast Shape-from-Template, Deformable Surface, Geometrical Outlier Removal, Affine Transformation Estimation, Depth Estimation, Local Texture

1. Introduction

The 3D reconstruction of surfaces has an important role in computer vision. A surface may be either rigid, which means it does not deform over time, or non-rigid. Stereo vision, multi-view geometry and Structure-from-Motion deal only with rigid surfaces [9]. It has been shown [1, 2] that the shape of rigid objects can be uniquely recovered using camera motion. On the other hand, many objects have flexible and non-rigid surfaces which reshape in time. Recovering the shape of deformed surfaces is thus very important when dealing with some applications in robotics [3, 4] and medical imaging [5, 6, 7].

A large body of research has been devoted to surface reconstruction. It was shown that in some configurations a solution can be found from a single image only and a template, following a setup called Shape-from-Template (SfT). In this regard, Moreno-Noguer *et al.* [10] proposed an approach to reconstruct elastic surfaces using shading and Salzmann *et al.* [11] use isometry and temporal smoothness. More recently, Bartoli *et al.* [12] showed that using the isometry condition one can solve for depth uniquely from point correspondences augmented by the local image deformation at first order. Amongst the many different approaches for shape recovery, some assume that the deformation is inextensible [14, 15, 16] and some do not [17, 18, 19].

In this paper, we propose a new approach to solve SfT for isometric surfaces.Instead of using corresponding keypoints to estimate the global warp function between the template and the deformed image as in [12], we propose to directly use the corresponding keypoints to estimate depth locally. Being motivated by the derivation given in [12], but to avoid solving a complex objective function globally which is time-consuming, we propose an algorithm to estimate the depth of each matched keypoint pair independently. Indeed, instead of estimating the warp globally, we estimate it locally for each matched pair with respect to both the local texture and

*Corresponding author. Tel./fax:+98 71 36474605.
E-mail address: azimifar@cse.shirazu.ac.ir, z.azimifar@gmail.com.

the neighboring matches. This warp is locally represented by an affine transformation that approximates the warp to first order. It is not the first time that the local warp is approximated as affine, see for instance Bachelder *et al.* [20]. In SfT however, this local transformation gives enough information to solve the problem exactly and very fast. The challenge is to thus to estimate these affine transformations very fast and to filter out the outliers in the correspondences, which were originally removed during global warp estimation. We thus propose two new geometrical methods to eliminate the outliers. These Geometrical Outlier approaches (GOR) are not only fast but are also good at detecting the outliers.

The remainder of this paper has the following sections. Existing approaches are first studied in Section 2. Since the proposed algorithm makes use of our previous work, Section 3 reviews the derivation of the isometric local SfT. Our main contribution for outlier removal and depth estimation are presented in Section 4. Experimental results of depth estimation and comparison with state-of-the-art are given in Section 5. Finally, conclusions and future research remarks are provided in Section 6.

2. Previous Work

Over the last decade, many different techniques have been proposed to represent and reconstruct the deformation of non-rigid surfaces. Salzmann and Fua [8] classified these methods into three main classes, namely physics-based, learning-based and regularize-based. Bartoli *et al.* [12] proposed a specific type of monocular surface reconstruction for deformable surfaces and named it SfT. It is noticed that reconstructing a deformable surface is an ill-posed problem. Therefore, some constraints should be considered to solve the problem uniquely. Isometry is a constraint that comes from physics and it can be used to tackle the problem of ambiguity associated with deformable surface reconstruction. Indeed, isometry means that deformation should be consistent against extendibility and the geodesic distance should not be changed during the deformation. Essentially, for a non-elastic surface like paper, we can consider this constraint:

$$\left\|P_i^t - P_j^t\right\|_G = \left\|P_i - P_j\right\|_G, \forall i, j \in 1..N \qquad (1)$$

where, P_i^t and P_i are keypoints of i^{th} matched pair between template and deformed image, respectively. Term $\|.\|_G$ denotes the geodesic distance and N is the number of corresponding pairs. Since computing the geodesic distance is not easy, the isometry constraint can be relaxed and reduced to the following constraint:

$$\left\|P_i^t - P_j^t\right\|_G \geq \left\|P_i - P_j\right\|_E, \forall i, j \in 1..N \qquad (2)$$

where $\|.\|_E$ denotes the Euclidean distance. This constraint indicates that the Euclidean distance between corresponding points should not be more than the Geodesic one. Violation to this constraint means that the surface is stretched. Most recent approaches use this reduced inequality constraint instead of original isometry constraint [21, 22, 23, 25]. Since our proposed method presents a solution for such a configuration, we briefly review the state-of-the-arts in this field, next.

Salzmann and Fua [21] used linear local models for well-textured surfaces. In their approach, first, they divided the deformed image to several smaller image patches and then for each patch, an objective function was defined with some distance equality and inequality constraints. After solving the objective functions and estimating the shape of each patch, the authors merged the calculated local surfaces of the patches using a linear model. It is noticed that for better accuracy, some overlaps are considered between the patches. Indeed, they model the global surface with these overlapped local patches. For low texture regions, another method [24] was proposed. In this method, by addressing low texture patches, the authors model intensities and describe an intensity profile. Afterwards, the deformation model is calculated from the intensity model. Finally, a Markov Random Field (MRF) model is used to align the textureless regions to the textured regions. Recently, Ngo *et al.* [23] presented a Laplacian formulation of [24]. They showed that this new representation leads to a well-conditioned system meanwhile the problem's size is reduced.

Locher *et al.* [25] proposed a discrete objective function to recover the shape of deformed surfaces. In this paper, the authors assumed that the position of the camera with respect to the template does not change and the deviations of keypoints from their original positions in the template are less than two centimeters. Then, the algorithm tries to minimize the sum of deviations from the original positions for some randomly selected matched pairs with respect to inextensibility inequality constraints. After solving the objective function, the depth information of matched pairs is used to estimate the depth of the control points. This estimation for control points is repeated several times with different matched pairs and then the mean values are chosen as the final depth values of control points. This method has drawbacks which are due to the assumptions taken

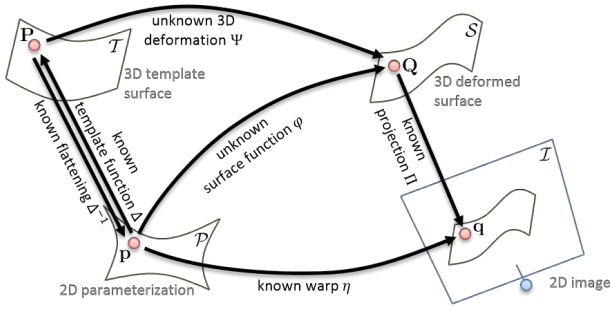


Figure 1: Geometrical modeling of SfT (courtesy of [12]).

by the algorithm. First, the depth values for the candidate pairs are discretized and the deviation from the template position could not be more than a predefined threshold (two centimeters) and the second, is that the distance and orientation of surface and camera should not be changed.

Another approach which uses a global objective function is proposed by Bartoli *et al.* [12, 13]. the authors showed that the depth of each point can be uniquely estimated using the position of the point on the template image and deformed image and the partial derivative between them. Since our proposed method is based on this theorem, we prefer to review this work here. Details of this method appear in Section 3. It is noticeable that all the methods which have been proposed for SfT, define a global objective function to estimate the warp function between the template and deformed images, but, in this study, we propose a local method which is faster than the global ones.

3. Shape-from-Template

Figure 1 demonstrates a geometric modeling of the SfT. In this configuration, it is supposed that the template image $\mathcal{P} \subset \mathbb{R}^2$ and its 3D surface $\mathcal{T} \subset \mathbb{R}^3$ are known. Also, $\mathcal{I} \subset \mathbb{R}^2$ which is the image of deformed surface $\mathcal{S} \subset \mathbb{R}^3$ is known. The goal is to recover the surface $\mathcal{S}$ which is an isometric deformation of surface $\mathcal{T}$. Recovering the surface $\mathcal{S}$ can be done by estimating one of the unknown functions Ψ or φ. It is noticeable that the warp function η, the invertible function Δ and the projection function Π are known. Indeed, Δ maps every 3D template point P on the surface $\mathcal{T}$ to its corresponding point p on the template image $\mathcal{P}$, and η maps every point p on template image $\mathcal{P}$ to its corresponding point q on the deformed image $\mathcal{I}$. Also, Π which is projection function, maps every 3D point Q from deformed surface $\mathcal{S}$ to its corresponding point q on the deformed image $\mathcal{I}$.

There are two different types of constraints in this model: reprojection constraint and deformation constraint. The reprojection constraint can be achieved by $\Pi \circ \varphi = \eta$ and $\varphi = \Psi \circ \Delta$. The deformation constraint comes from isometry. Since we know that for an isometric deformation, distances are preserved, then we can model deformation constraints by $J_{\Psi,\mathcal{T}}^{\top} J_{\Psi,\mathcal{T}} = I$ where the notation $J_{\Psi,\mathcal{T}}$ refers to the first-order derivative operator acting on the tangent $space^2$ of $\mathcal{T}$. Therefore, the isometric SfT problem can be stated as:

$$\begin{array}{l} Find \;\; \varphi : \mathcal{P} \rightarrow \mathbb{R}^3 \\ s.t. \left\{ \begin{array}{c} \Pi \circ \varphi = \eta \\ J_{\Psi,\mathcal{T}}^{\top} J_{\Psi,\mathcal{T}} = I \end{array} \right. \end{array} \tag{3}$$

The reprojection constraint $\Pi \circ \varphi = \eta$ can be simply rewritten [12] as:

$$\varphi = \gamma \tilde{\eta} \tag{4}$$

where γ is the unknown depth function and $\tilde{\eta}$ is the homogenous presentation of η and:

$$\tilde{\eta} \propto \left[\begin{array}{c} \eta \\ 1 \end{array} \right] \tag{5}$$

Bartoli*et al.* [12] showed that the deformed embedding function φ for every point p of the deformed image can be found using:

$$\varphi(p) = sqrt\Bigg(\Big(J_{\Delta}^{\top}(p) J_{\Delta}(p) \left(J_{\eta}^{\top}(p) J_{\eta}(p) - \frac{1}{\|\tilde{\eta}(p)\|_2^2} J_{\eta}^{\top}(p) \eta(p)^{\top} J_{\eta}(p)\right)^{-1}\Big)\Bigg) \tilde{\eta}(p) \tag{6}$$

The equation infers an image points depth given its correspondence in the template and the local affine transformation that holds between the template and the image at this point. In this paper, we use this equation to estimate the 3D position of each keypoint in the deformed image.

4. Proposed Approach

In this section, novel approaches for outlier removal and depth estimation are presented. Figure 2 demonstrates flow diagram of the proposed method. Since it is assumed that the template image and its 3D surface are known, the template image's keypoints are extracted prior to starting the main algorithm. This not only saves the time but also, leads to select the keypoints which are more repetitive and robust against noise and deformation. Then, corresponding points between the deformed

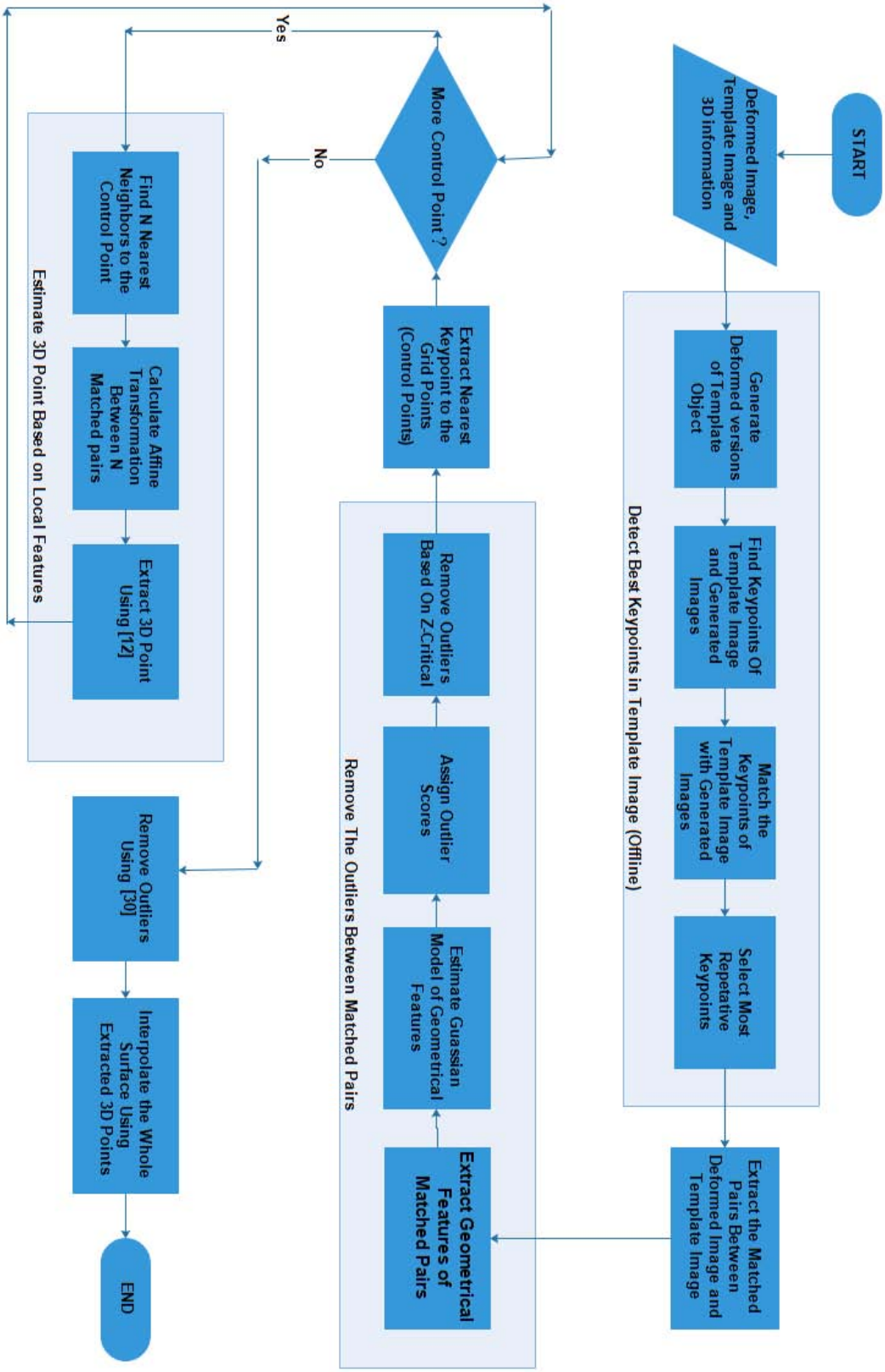


Figure 2: Flowchart of the proposed pointwise SfT method.

image and template image are extracted. Here, our new outlier removal approach is applied to remove outliers between the corresponding points.

Since, it is very time consuming to estimate the depths of all individual points on deformed object, we estimate the depth of just a few number of points called mesh points. These mesh points should cover the entire surface of template object since they are used for interpolating the depth of other keypoints. Indeed, the locations of mesh points are predefined and can be selected with respect to the template object. In this paper, the mesh points are selected uniformly like a grid from the template image and, thus, we call them grid points. To estimate the 3D surface of the deformed object, first the depths of predefined grid points from the template image are estimated. Since, it is not guaranteed to have the position of corresponding points of the grid points on deformed images, the 3D position of the nearest corresponding pairs to the grid points are estimated instead of grid points.

Therefore, for each nearest corresponding pair c to grid points, corresponding points N are extracted from the local neighborhood of c on template image. Based on the corresponding points N, the affine transformation between the points of pair c is calculated. Then, the estimated affine transformation is substituted in (5) to estimate the depth of the point on the deformed surface.

An outlier removal approach discussed in [32] is also applied here to remove the noisy 3D points. In this method, the outliers are detected in the 3D space, with respect to isometry constraints. Indeed, some pairwise constraints are considered for every two extracted 3D points in a small neighborhood. Next, the 3D points which are not consistent with the others are labeled as outliers. The consistency between two 3D points comes from the fact that the Euclidean distance between two points should not be larger than the geodesic one which is calculated based on the template image. If the Euclidean distance of two extracted 3D points becomes greater than their geodesic distance in the template surface, then these two points are not consistent.

Finally, the 3D positions of grid points are interpolated using all the remaining extracted 3D points. A detailed description of the above process is given next. All the experiments in this section are conducted on Spiderman dataset whose template and deformed images are demonstrated in Figure 3.

(a)

(b)

Figure 3: Spiderman dataset. (a) Template Image. (b) Deformed Image.

4.1. Best Keypoint Selection

Finding corresponding points between two images plays a fundamental role in image processing and computer vision, especially in stereo vision. It is used in a wide variety of applications such as object detection, depth estimation and image registration. Recently, different types of image descriptors are proposed to address this problem [26, 27, 28, 29, 30]. In this paper, since, the template image is available, a training algorithm is used to select best keypoints from the template image.

Selecting the best keypoints leads to more accurate and faster results. To do so, first, we extract all the possible keypoints using SIFT descriptor (here the other image descriptors like ORB can be used instead of SIFT).

Then, several affine deformations with various affine parameters are applied on the template image. This is followed by extracting keypoints from the new deformed images. The extracted keypoints from deformed images are matched with the extracted keypoints from the template image. For each template keypoint, the number of correctly matched pairs from the deformed images are calculated. Finally, the template keypoints

which have the most number of true peers on deformed images are selected based on a predefined threshold and are named as the best template keypoints. The predefined threshold can be selected by two different strategies. In the first strategy, the threshold is chosen so that a certain number of keypoints are extracted among all the keypoints. The second strategy which is used in our experiments, is to select the keypoints whos matching score passes the threshold. Here, we keep those keypoints that are matched exactly at least one time and remove the others. This means the threshold is set to one.

In fact, such keypoints are more repetitive and are more likely to be extracted and matched when the surface is deformed. Only these best template keypoints are kept in the memory for next steps. This part of proposed algorithm is done offline to accelerate the method.

4.2. Geometrical Outlier Removal

Many image descriptors have been proposed recently. Although, some of these methods are very effective, some outliers always appear between their extracted pairs. Here, a global method is presented to remove the outliers of matched pairs with respect to geometrical properties of the matched pairs. It is noticed that since the proposed method just uses the geometrical properties of matched pairs, it can be used along the other outlier removal approaches. Indeed, to be invariant with respect to rotation and scale, most of the image descriptors assign an orientation and scale to each extracted keypoint.

These geometrical properties of keypoints are extracted just to normalize the descriptors such that they become invariant to orientation and scale changes. When the keypoint matching process takes place, just the descriptors are used. Of course, it is logical, because there is no relation between the size and orientation of the keypoints in one image. But, when the corresponding points between two images are extracted, some meaningful relations between the values of orientations and sizes of the truly matched pairs will happen.

For example, all the keypoints of the image are rotated and scaled in the same way when the viewpoint changes. With respect to these observations, we propose to use a Geometrical Outlier Removal approach to detect and remove the outliers. Two GOR-based algorithms are presented in this paper. Algorithm 1 and Algorithm 2 describe these methods step by step. In these algorithms, first, the geometrical properties of each matched pair are extracted, then the geometrical features are modeled and following model fitting, the outliers are removed. Further details about these methods are presented next.

4.2.1. Geometrical Feature Extraction

Each keypoint has some geometrical properties, including the keypoint's position, orientation and scale. The scale and orientation of each keypoint are extracted with respect to its local texture. Therefore, we do not expect any dependency between geometrical properties of two keypoints within an image, even between two near keypoints. But there is a direct dependency between the geometrical properties of correctly matched pairs between two images. For example, regardless of the values of geometrical properties of keypoints, scale ratio and the difference of orientation offset of matched pairs are likely to be the same for all the correctly matched pairs. This is because the geometrical properties of keypoints on an object or a surface change in the same way, even when there is a deformation on the surface of the object and these values obey the same distribution. The geometrical properties of a corresponding pair can be achieved using the geometrical properties of its keypoints as follows:

$$F = \begin{cases} f_1 = x_1 - x_2 \\ f_2 = y_1 - y_2 \\ f_3 = s_1 / s_2 \\ f_4 = \sin(o_1 - o_2) \\ f_5 = \cos(o_1 - o_2) \end{cases} \tag{7}$$

here, x, y, s, o are respectively x coordinate, y coordinate, scale and orientation of a keypoint from the first image or the second one.

4.2.2. Modeling Outliers Using Robust Gaussian

We propose a Robust Gaussian Model for Geometrical Outlier Removal (RGM_GOR). In this method first, we only model the inliers and based on this model an outlier score is assigned to each matched pair. Afterward, a Z-Score threshold is used to filter the outliers. In this method, since an outlier score is assigned to each matched pair, we can rank the pairs. The pairs with smaller outlier scores are more likely to be inliers.

Of course, someone may claim that just a single Robust Gaussian model is not enough to model the inliers and it is better to model inliers using a mixture of Gaussians. In Sec. 4.2.3, we will show that there is not any significant achievement by using mixture models, however, our second outlier removal approach employs a mixture of Gaussians to model inliers.

- Outlier Score Assignment:

After finding the geometrical properties of matched pairs, a Gaussian model is considered to describe these geometrical properties. A Gaussian model is determined by its mean and covariance. Of course, here, to be robust against outliers, instead of calculating mean and covariance matrix, median and robust covariance are used. The robust covariance is calculated as:

$$\begin{aligned} \mu &= med\,(F) \\ \Sigma &= med\left(\left(F^i - \mu\right) * \left(F^j - \mu\right)\right) \end{aligned} \tag{8}$$

where, F is determined in (6).

Now, the Mahalanobis distance between each matched pair and the center of the distribution is considered as the outlier score for the matched pair:

$$Score(F^i) = \left(F^i - \mu\right)^\top \Sigma^{-1} \left(F^i - \mu\right) \tag{9}$$

- Filtering the Outliers:

The matched pairs are sorted with respect to their outlier scores. Our approach to remove the outliers is based on the Z-Score filtering. Indeed, the Z-Score of each pair is calculated with respect to all the outlier scores and the pairs with the Z-Sores more than Z-Critical, are marked as outliers. Of course, we can select as much as inliers pairs we need from the first of the sorted list. Indeed, the pairs with larger outlier scores are more likely to be an outlier. Algorithm 1 describes the details of RGM_GOR.

4.2.3. Modeling Outliers Using Gaussian Mixture Model

The second approach to filter outliers is to model the inliers and outliers, simultaneously. Since the keypoints are matched with respect to their descriptors, for the outlier pairs, the keypoints may be matched anywhere with random position, rotation and scale. Therefore, the geometrical features of outlier pairs are random. It is expected that their distribution obey a Gaussian model with wide variance. Figure 4 verifies the wider variance of outlier features than inlier ones. Hence, the problem can be converted to a clustering problem to find the best Gaussian mixture models that can be fitted on geometrical features. Indeed, we tried to fit a Gaussian mixture model on geometrical features and filter the models with a large determinant of covariance matrix which are expected to be the outlier model. Here, Bayesian Robust Mixture Model (BRMM) [38] is used to find the models and their parameters. BRMM is an adaptive method which can find the true number of models. Moreover, this method is robust against outliers, too.

Algorithm 1 Geometrical Outlier Removal (RGM_GOR)

Input:
I_t // *Template Image*
I_i // *Deformed Image*
Descriptor // *Image Descriptor Like SIFT or ORB*
Output:
M // *MatchedPairs*
S // *OutlierScores*

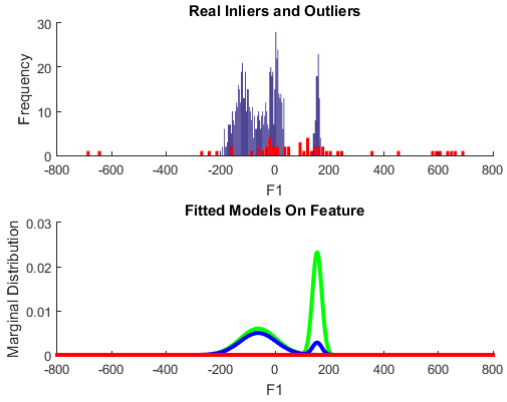
$M = FindMatchedPairs(I_t, I_i, Descriptor)$
for M^t **in** M **do**
  $F^t_1 \leftarrow M^t_{X_1} - M^t_{X_2}$
  $F^t_2 \leftarrow M^t_{Y_1} - M^t_{Y_2}$
  $F^t_3 \leftarrow M^t_{S_1} - M^t_{S_2}$
  $F^t_4 \leftarrow \sin(M^t_{O_1} - M^t_{O_2})$
  $F^t_5 \leftarrow \cos(M^t_{O_1} - M^t_{O_2})$
end for
$\mu \leftarrow Median(F)$
$\Sigma \leftarrow Median\left(\left(F^i - \mu\right) * \left(F^j - \mu\right)\right)$
for F^t **in** F **do**
  $S^t = (F^t - \mu)\,\Sigma^{-1}\,(F^t - \mu)^\top$
end for
while *true* **do**
  $Z = ZScore(S)$
  if $Max(Z) > ZCritical(S)$ **then**
    Remove $Max(S)$ from M, S
  else
    Break
  end if
end while
Sort M, S with respect to S

Return M, S

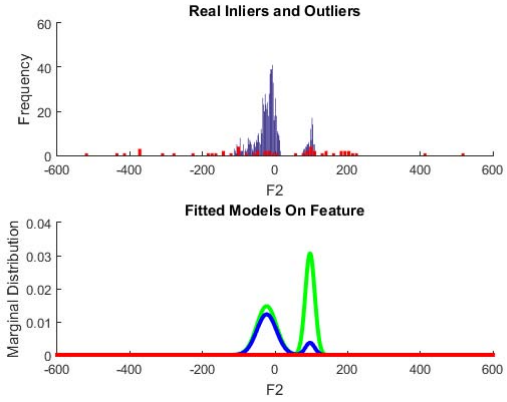
Most of the times, only two models are fitted on geometrical features, one describes inliers and the other one describes outliers. This observation also validates our previous RGM_GOR approach which models the inliers just by using one robust Gaussian model. Figure 4 shows the marginal distribution of estimated inlier model for the dataset 'paper14.jpg' which has large deformation and located far from the camera. Here, sometimes more than one model are fitted on inliers (based on initial condition), however, more often just one Gaussian model is fitted using BRMM.

We call this approach Bayesian Robust Mixture Model for Geometrical Outlier Removal (BRMM_GOR). BRMM_GOR directly models both outliers and inliers. The matched pairs which belong to outlier model are removed and in this method no rank list for the inlier pairs is assumed.

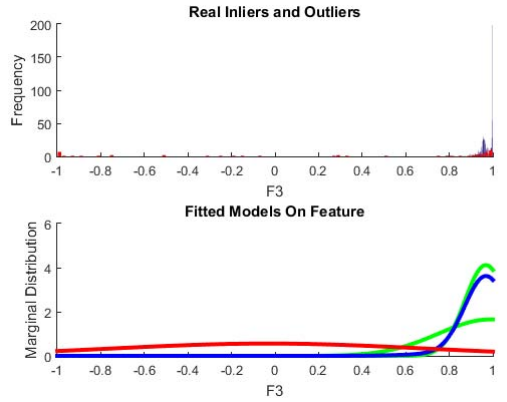
The results show that most of the time, there is just one Gaussian model for inliers and one Gaussian model for outliers. We classify each model as inlier or outlier with respect to the determinant of its covariance matrix. Indeed, the models with the determinant of covariance bigger than 10^5 are classified as outlier models and otherwise as inlier models. Algorithm 2 describes the proposed BRMM_GOR in details.



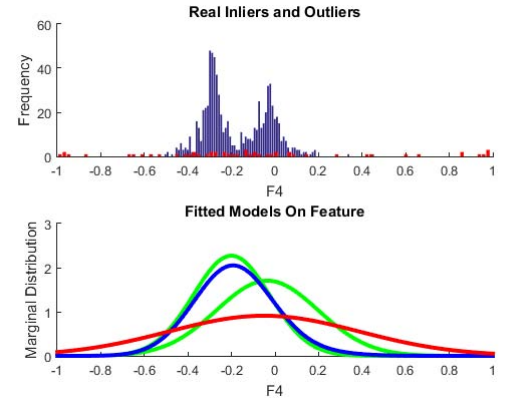
(a) F1, displacement on X axis



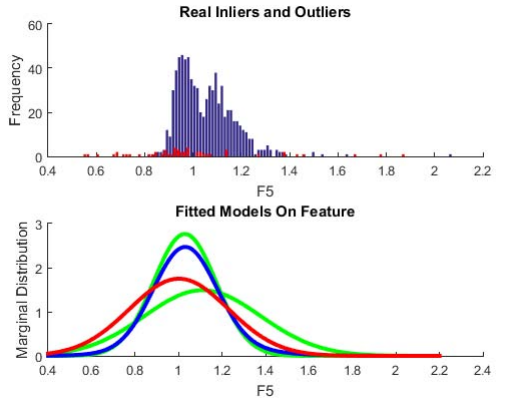
(b) F2, displacement on Y axis



(c) F3, scale ratio



(d) F4, sine of orientation offset



(e) F5, cosine of orientation offset

Figure 4: Marginal distributions of fitted models using BRMM. The blue bars and lines show the distribution of inliers and the red ones show the distribution of outliers. The green lines show the marginal distribution of each inlier model.

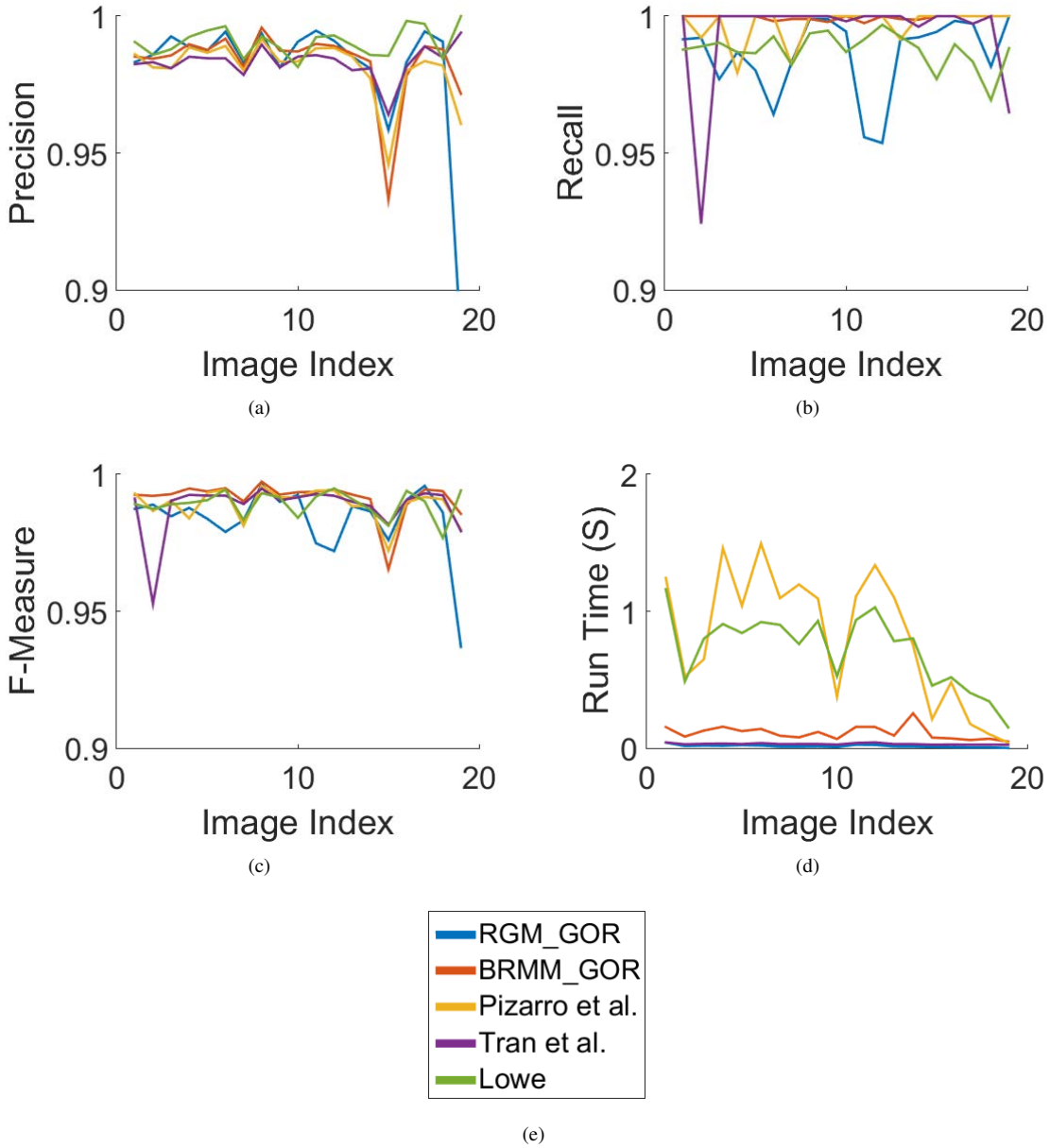


Figure 5: Comparing our method with Tran *et al.* [36], Pizarro *et al.* [37] and Lowe [29] on artificially deformed Spiderman datasets.

Algorithm 2 Geometrical Outlier Removal (BRMM_GOR)

Input:
I_t // *Template Image*
I_i // *Deformed Image*
Descriptor // *Image Descriptor Like SIFT or ORB*
Output:
M // *MatchedPairs*

$M = FindMatchedPairs(I_t, I_i, Descriptor)$
for M^t **in** M **do**
  $F^t_1 \leftarrow M^t_{X_1} - M^t_{X_2}$
  $F^t_2 \leftarrow M^t_{Y_1} - M^t_{Y_2}$
  $F^t_3 \leftarrow M^t_{S_1} - M^t_{S_2}$
  $F^t_4 \leftarrow \sin(M^t_{O_1} - M^t_{O_2})$
  $F^t_5 \leftarrow \cos(M^t_{O_1} - M^t_{O_2})$
end for
$Models = BRMM(F)$
for *Model* **in** *Models* **do**
  if $Det(Model) < 1.0e5$ **then**
    Add *Model* to *Inliers_Models*
  end if
end for

Return $M \in Inliers\_Models$

Table 1: Comparing RGM_GOR with SIFT on artificially deformed Spiderman datasets (ratio test = 0.7).

Image number	# SIFT Outliers	# SIFT Inliers	# RGM_GOR Outliers	# RGM_GOR Inliers	Recall	Precision	F-Measure
1	44	1065	18	1056	0.991	0.983	0.987
2	36	633	9	628	0.992	0.985	0.988
3	43	824	6	805	0.976	0.992	0.984
4	39	1063	12	1049	0.986	0.988	0.987
5	39	960	12	941	0.980	0.987	0.983
6	38	1088	6	1049	0.964	0.994	0.979
7	37	958	16	942	0.983	0.983	0.983
8	41	954	6	953	0.998	0.993	0.996
9	47	944	18	943	0.998	0.981	0.990
10	40	535	5	532	0.994	0.990	0.992
11	33	1176	6	1124	0.955	0.994	0.974
12	34	1273	11	1214	0.953	0.991	0.971
13	47	943	14	935	0.991	0.985	0.988
14	53	774	15	768	0.992	0.980	0.986
15	55	348	15	346	0.994	0.958	0.976
16	37	587	10	586	0.998	0.983	0.990
17	26	360	2	359	0.997	0.994	0.995
18	26	326	3	320	0.981	0.990	0.986
19	25	171	23	171	1	0.881	0.936
Average	39	788	10	774	0.985	0.981	0.983

Table 2: Comparing BRMM_GOR with SIFT on artificially deformed Spiderman datasets (ratio test = 0.7).

Image number	# SIFT Outliers	# SIFT Inliers	# BRMM_GOR Outliers	# BRMM_GOR Inliers	Recall	Precision	F-Measure
1	44	1065	16	1065	1.000	0.985	0.993
2	36	633	10	633	1.000	0.984	0.992
3	43	824	12	824	1.000	0.986	0.993
4	39	1063	12	1063	1.000	0.989	0.994
5	39	960	12	960	1.000	0.988	0.994
6	38	1088	9	1086	0.998	0.992	0.995
7	37	958	19	957	0.999	0.981	0.990
8	41	954	4	953	0.999	0.996	0.997
9	47	944	12	942	0.998	0.987	0.993
10	40	535	8	535	1.000	0.985	0.993
11	33	1176	12	1173	0.997	0.990	0.994
12	34	1273	14	1273	1.000	0.989	0.995
13	47	943	13	942	0.999	0.986	0.993
14	53	774	13	773	0.999	0.983	0.991
15	55	348	25	348	1.000	0.933	0.965
16	37	587	13	587	1.000	0.978	0.989
17	26	360	4	360	1.000	0.989	0.994
18	26	326	4	326	1.000	0.988	0.994
19	25	171	5	171	1.000	0.972	0.986
Average	39	788	11	788	0.999	0.983	0.991

4.2.4. Outliers Removal Results

Here, SIFT is used for detection and description of keypoints. Also, to match the descriptors, we use ratio test which is originally proposed by David Lowe [29]. To evaluate our methods, several isometric deformations are applied artificially to Spiderman template image and 19 different deformed images are provided. Here, we used deformation toolbar [35] to apply isometric deformations. Since the warp function is known for these deformed images, it is possible to check if a matched pair is true positive or false positive. Tables 1 and 2 show the results of RGM_GOR and BRMM_GOR on artificially deformed Spiderman dataset, respectively. The values for recall and precision are calculated with respect to the extracted pairs by the original SIFT algorithm.

Moreover, we compared the result of our proposed algorithms with some of the state-of-the-art methods. Here, works proposed by Tran *et al.* [36], Pizarro *et al.* [37] and Lowe [29] are compared with RGM_GOR and BRMM_GOR. Figure 5 compares the results using the artificially deformed Spiderman dataset individually

Table 3: Comparing RGM_GOR, BRMM_GOR, Tran *et al.* [36], Pizarro *et al.* [37] [29] and Lowe on artificially deformed Spiderman datasets (ratio test = 0.7) with respect to average recall and precision.

Method	Mean Recall	Mean Precision	Mean F-Measure	Mean Run Time
RGM_GOR	0.986	0.981	0.983	0.0138
BRMM_GOR	0.999	0.983	0.991	0.114
Tran *et al.*	0.983	0.994	0.988	0.0357
Pizarro *et al.*	0.981	0.997	0.989	0.850
Lowe	0.990	0.987	0.989	0.704

for each image and Table 3 compares all the methods with respect to average values.

It should be noted that the reported run time in this table does not include the corresponding point extraction time. Indeed, the corresponding point extraction step must be done fairly by all methods and its run time overhead is identical for all. Moreover, several local image feature descriptors such as ORB, SIFT, SURF and recently LIFT [30] exist which can be used to extract the keypoints in images. All of them have GPU-based and CPU-based implementations with their own time complexity and accuracy. Someone can select the corresponding point extraction method with respect to the problem definition and limitations. Since, selecting a proper corresponding point extraction method is not our main concern in this paper and it has the same overhead for all the outlier removal methods, we do not report run time of this step in our experiments.

Moreover, the number of outliers when using SIFT algorithm can be controlled by applying different thresholds when applying ratio test. Indeed, the probability that a matched pair is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest which is called ratio test. Therefore, by applying bigger threshold on ratio test we allow more outliers and inliers to be selected.

The value of the ratio test threshold is selected with respect to the number of required corresponding points. Lowe [29] suggested a ratio test threshold of 0.8. He demonstrated experimentally that the best value for the ratio test threshold lies between 0.7 and 0.8. Here we selected 0.7. Also, Tables 4 and 5 show the results of RGM_GOR and BRMM_GOR with respect to the number of outliers by applying various ratio test thresholds on Spiderman dataset. Moreover, in Table 6, RGM_GOR and BRMM_GOR are compared with the state-of-the-art methods with respect to average values when different values of ratio test are applied on Spiderman dataset.

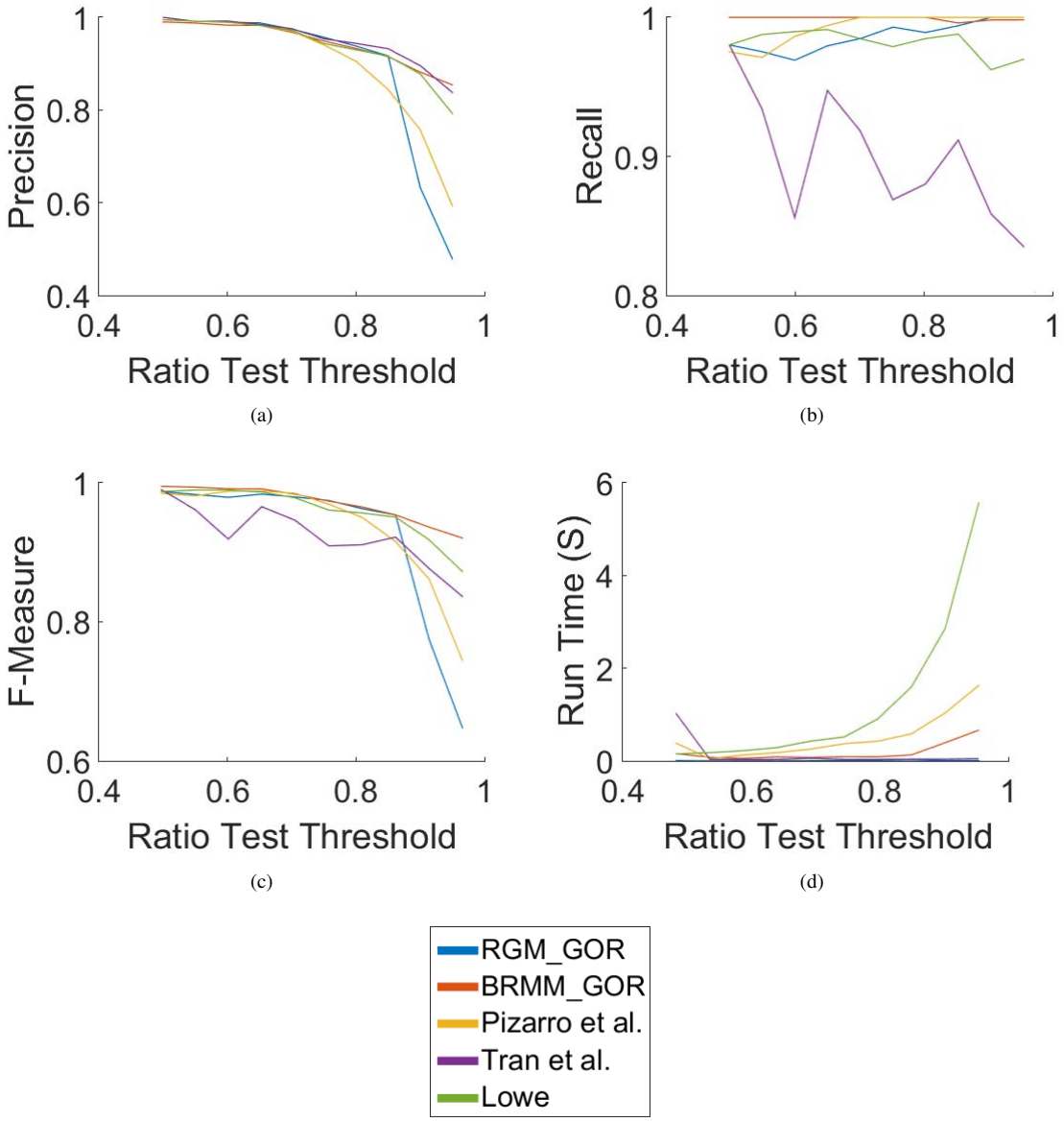
Table 4: The efficiency of RGM_GOR on Spiderman dataset using different ratio thresholds.

Ratio Test Threshold	# SIFT Outliers	# SIFT Inliers	# RGM_GOR Outliers	# RGM_GOR Inliers	Recall	Precision	F-Measure
0.5	2	202	1	194	0.960	0.995	0.977
0.55	3	243	2	235	0.967	0.992	0.979
0.6	7	292	3	279	0.955	0.989	0.972
0.65	14	342	3	325	0.950	0.991	0.970
0.7	28	392	9	384	0.980	0.977	0.978
0.75	53	427	17	421	0.986	0.961	0.973
0.8	94	459	26	453	0.987	0.946	0.966
0.85	163	499	42	490	0.982	0.921	0.951
Avg.	46	357	13	348	0.9710	0.9715	0.9709

In order to measure the performance and the robustness of the proposed method, we have generated another dataset with 10 completely different images and various textures. Indeed, a random deformation is applied to each of the images and 10 pair images are provided to evaluate the performance of proposed method. Table 7 shows the comparison between SIFT and the RGM_GOR outlier removal and Table 8 shows the same for BRMM_GOR. Since each image in this dataset has its own texture, therefore, it is concluded that the

Table 5: The efficiency of BRMM_GOR on Spiderman dataset using different ratio thresholds.

Ratio Test Threshold	# SIFT Outliers	# SIFT Inliers	# BRMM_GOR Outliers	# BRMM_GOR Inliers	Recall	Precision	F-Measure
0.5	2	202	2	202	1.000	0.990	0.995
0.55	3	243	3	243	1.000	0.988	0.994
0.6	7	292	3	292	1.000	0.990	0.995
0.65	14	342	6	342	1.000	0.983	0.991
0.7	28	392	13	392	1.000	0.968	0.984
0.75	53	427	23	427	1.000	0.949	0.974
0.8	94	459	32	459	1.000	0.935	0.966
0.85	163	499	46	497	0.996	0.915	0.954
Avg.	46	357	16	356	0.9995	0.9636	0.9811



(a) (b) (c) (d) (e)

Figure 6: The effect of outliers percentage on outlier removal. Comparing our method with Tran *et al.* [36], Pizarro *et al.* [37] and Lowe [29] on Spiderman dataset with different ratio tests.

Table 6: Comparing the proposed methods with the state-of-the-art methods on Spiderman dataset with different ratio test threshold

method	# Average Outliers	# Average Inliers	Average Precision	Average Recall	Average F-Measure
BRMM_GOR	16	356	0.9995	0.9636	0.9811
RGM_GOR	13	348	0.9710	0.9715	0.9709
Tran *et al.*	12	320	0.8999	0.9712	0.9339
Pizarro *et al.*	24	355	0.9909	0.9524	0.9704
Lowe	16	351	0.9856	0.9652	0.9751

proposed geometrical outlier removal approaches especially BRMM_GOR do cope with a variety of image textures. The results show that BRMM_GOR have better performance than RGM_GOR on this dataset too. Table 9 compares the average performance of proposed methods with the other state-of-the-art outlier removal methods on this dataset.

As it can be seen from Figure 6, the proposed outlier removal approaches are very effective in removing the outliers. It should be noticed that we can find more inliers by employing higher ratio threshold without any concern about outliers.

4.2.5. Discussion

In this paper two methods are introduced to remove outliers from corresponding points between two images. We aim to maintain the tradeoff between accuracy and speed by proposing the two approaches, such that the first method, RGM_GOR is super-fast with a reasonable accuracy, while the second method, BRMM_GOR performs very accurately with an accepted speed. Indeed, BRMM_GOR is proposed since the geometrical features do not always obey a Gaussian distribution, especially when the surface deformation is significant or there is more than one object in the scene. This is why BRMM_GOR has better performance than RGM_GOR on the artificial datasets. However, it runs at a lower speed. The contrast between BRMM_GOR and RGM_GOR is the tradeoff between accuracy and speed. Any prior knowledge about the dataset can help one select the most efficient method.

Although it may come to mind that geometrical features introduced here, neither are necessary for outlier removal, nor impose a positive effect when in-plane rotation occurs, the experimental results indicate the positive effectiveness of these features. Figure 7 shows the in-plane rotation dataset and Table 10 compares the result of proposed method when displacement features are used and when they are not.

Another important challenge is when two or more objects appear in the scene and they move and rotate independently. Figure 8 shows such an example. Table 11 compares performance of the proposed algorithm with

Table 7: Comparing RGM_GOR with SIFT on randomly deformed images with different textures (ratio test = 0.7).

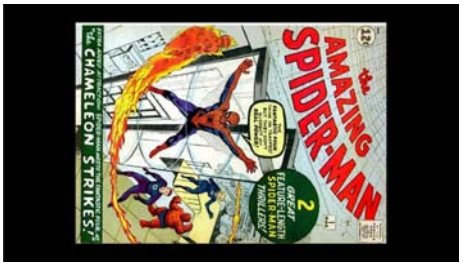
Image number	# SIFT Outliers	# SIFT Inliers	# RGM_GOR Outliers	# RGM_GOR Inliers	Recall	Precision	F-Measure
1	25	893	20	864	0.968	0.977	0.972
2	16	838	15	807	0.963	0.982	0.972
3	33	1610	25	1566	0.984	0.973	0.978
4	16	376	12	360	0.957	0.968	0.963
5	26	1174	23	1146	0.976	0.980	0.978
6	46	845	38	804	0.951	0.955	0.953
7	39	728	29	703	0.966	0.960	0.963
8	33	856	28	822	0.960	0.967	0.964
9	52	1003	47	956	0.953	0.953	0.953
10	38	1200	33	1147	0.956	0.972	0.964
Average	32	952	27	928	0.962	0.970	0.966

Table 8: Comparing BRMM_GOR with SIFT on randomly deformed images with different textures (ratio test = 0.7).

Image number	# SIFT Outliers	# SIFT Inliers	# RGM_GOR Outliers	# BRMM_GOR Inliers	Recall	Precision	F-Measure
1	25	893	14	893	1	0.985	0.992
2	16	838	6	836	0.998	0.993	0.995
3	33	1610	12	1610	1	0.993	0.996
4	16	376	7	375	0.997	0.982	0.989
5	26	1174	12	1173	0.999	.990	0.995
6	46	845	21	845	1	0.976	0.988
7	39	728	18	726	0.997	0.976	0.986
8	33	856	13	856	1	0.985	0.992
9	52	1003	9	1001	0.998	0.991	0.995
10	38	1200	9	1199	0.999	0.993	0.996
Average	32	952	12	951	0.999	0.986	0.993

Table 9: Comparing RGM_GOR, BRMM_GOR, Tran *et al.* [36], Pizarro *et al.* [37] and Lowe [29] on randomly deformed images (ratio test = 0.7) with respect to average recall and precision.

Method	Mean Recall	Mean Precision	Mean F-Measure	Mean Run Time
RGM_GOR	0.9623	0.9699	0.9661	0.0139
BRMM_GOR	0.9987	0.9863	0.9925	0.1258
Tran *et al.*	0.9984	0.9652	0.9815	0.0365
Pizarro *et al.*	0.9886	0.9847	0.9866	0.9731
Lowe	0.9875	0.9867	0.9871	0.677



(a)

(b)

Figure 7: In-plane rotation dataset. (a) Template image (b) Deformed image.

the state-of-the-art methods applied on this dataset. As it was expected, the proposed method has good consequence on these datasets.

Low image resolution is another challenge which should be discussed here. Indeed, the image resolution directly affects the number of detected keypoints and as a result the number of matched pairs between the images. The lower the resolution, the lower the number of matched pairs. With respect to the rows 17, 18 and 19 of Table 1 and Table 2, it can be seen that the proposed outlier removal algorithms have fair performance even when the matched pairs are not too many.

4.3. Surface Reconstruction

Surface reconstruction consists of two steps. The first step is to estimate the affine transformation between

(a)

(b)

Figure 8: Multi object dataset. (a) Template image (b) Deformed image.

matched pairs of the template and deformed images and the second step is estimating the depth of the matched pairs.

- Affine Transformation Estimation:

As it is shown in Section 3, the depth of each point can be estimated by its locations and first-order differential independently. Here, to calculate the Jacobean between the keypoints of each pair p, N nearest corresponding points to p are extracted. Then, using these

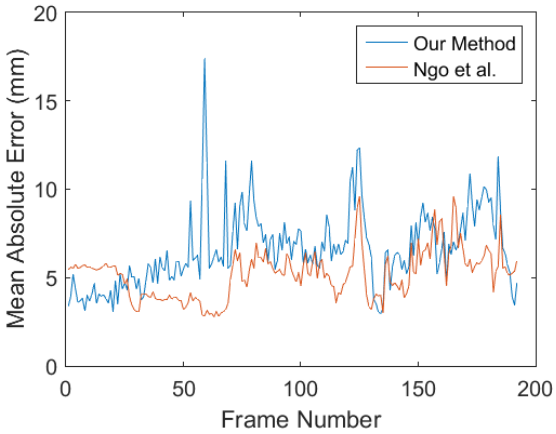


Figure 9: Comparing the proposed method and the one in [23] with respect to the mean absolute error on kinect_paper dataset.

Table 10: The influence of using displacement features on in-plane rotation image. BRMM_GOR* is modified version of BRMM_GOR which does not use displacement features. True number of inliers is 284 and true number of outlier is 86

method	# Outliers	# Inliers	Precision	Recall	F-Measure
BRMM_GOR	79	291	0.9691	0.9930	0.9809
BRMM_GOR*	82	288	0.9722	0.9859	0.9790
Tran *et al.*	69	301	0.9435	1	0.9709
Pizarro *et al.*	52	318	0.8931	1	0.9435
Lowe	78	292	0.9623	0.9894	0.9757

points, the affine transformation between the local patch is estimated.

It is noticed that since an image consists of discretized pixels, the matched pairs are not exactly peer to peer and usually some small displacement is likely to happen between them. Therefore, to address robustness against these displacement noises between corresponding points, the RANSAC algorithm is used to estimate the affine transformation between matched pairs N, accurately. Finally, the estimated affine transformation is assigned to the pair p.

- Depth Estimation:

Eq. 5, which is proposed in [12], is used to estimate the depth of the matched pairs. However, the estimated depths may be noisy, since the affine transformation between two keypoints are effected by displacement noises or some outliers may remain as the result of keypoint matching.

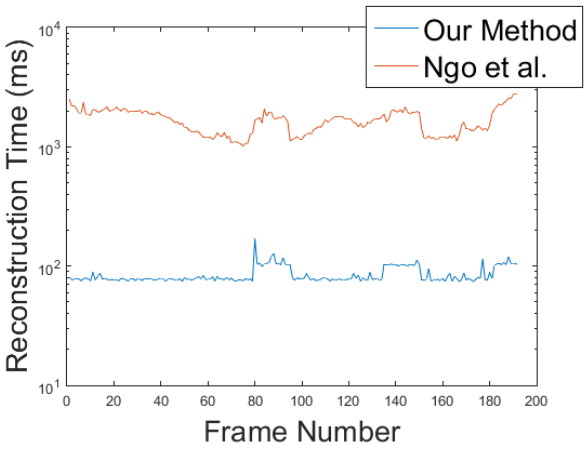


Figure 10: Comparing the proposed method and the one in [23] with respect to reconstruction time on Kinect_paper dataset.

Table 11: BRMM_GOR can model multi-modal distributions and remove outliers. True number of inliers is 2358 and true number of outlier is 131

method	# Outliers	# Inliers	Precision	Recall	F-Measure
BRMM_GOR	102	2387	0.9849	0.9970	0.9909
Tran *et al.*	588	1901	0.9826	0.7922	0.8772
Pizarro *et al.*	497	1992	0.9669	0.8168	0.8855
Lowe	126	2363	0.9932	0.9953	0.9943

Algorithm 3 Pointwise SfT

Input:
I_t // *Template Image*
I_i // *Deformed Image*
T_m // *Template 3D Model*
Descriptor // *Image Descriptor Like SIFT or ORB*
N // *Neigbours*
Output:
D // *Depth of Grid Points* C_p

$M = BRMM\_GOR(I_t, I_i, Describtor)$
for M^t **in** M **do**
  $N^t \leftarrow NearestNeigbour(M^t, N)$ // N Nearest Neighbours of M^t
  $A^t \leftarrow CalcAffine(N)$ // Calculates Affine Transformation Using RANSAC
  $P^t \leftarrow EstimateDepth(M^t, A^t)$ // Estimates 3D Position of M^t using [12]
end for
$P \leftarrow RemoveOutliersIn3D(P)$ // Remove the outliers using [32]
$D \leftarrow Interpolate(P, C_p)$ // Estimate 3D position of Grid Points.
Return D

Therefore, the proposed approach in [32] is used to remove the keypoints whose depths are not consistent.

The extracted 3D points do not cover all the deformed surface completely, therefore, thin plate spline method [33] is used to interpolate the final deformed surface. The details of our proposed method are explained in Algorithm 3.

5. Experimental Results

To evaluate the proposed method, we measured its performance on real and artificial datasets. We applied isometric deformation on Spiderman template image to generate new artificial datasets. We compared our algorithm with the one proposed in [23] on the 'kinect_paper' [24] dataset in term of accuracy and efficiency. It is noticed that the original implementation which is also available on their homepage is considered for comparisons. The mean absolute error of grid points

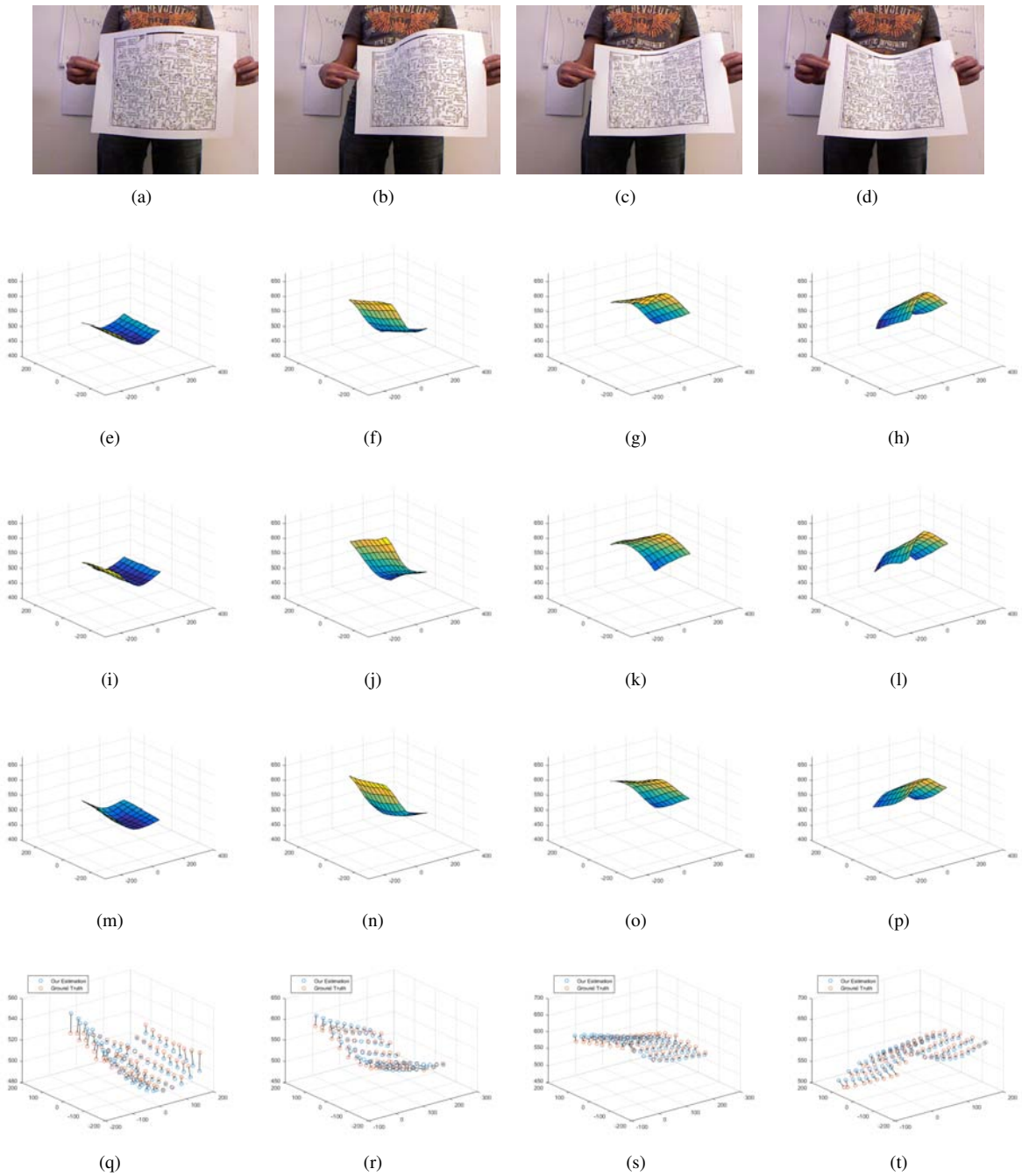
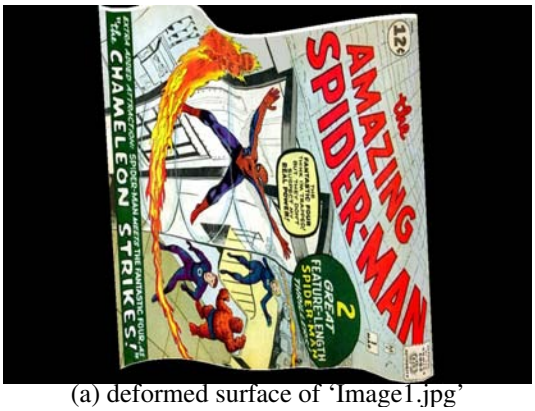
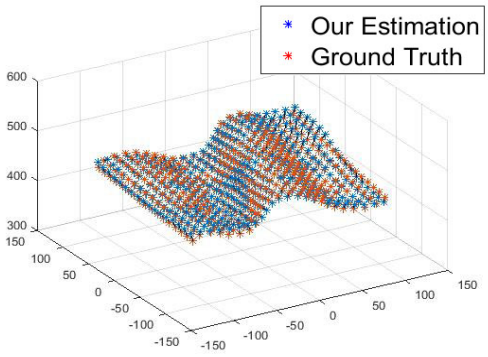


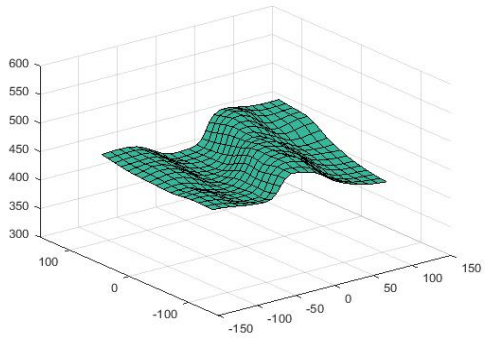
Figure 11: Comparison between the proposed method and the one in [23] (a-d) The deformed images corresponding to the files 'frame_041.png', 'frame_071.png', 'frame_081.png' and 'frame_101.png' respectively. (e-h) Ground truths. (i-l) Estimated surfaces by the method proposed by Ngo *et al.* [23] (m-p) Estimated surfaces by proposed method. (q-t) The visualized errors of our estimation.



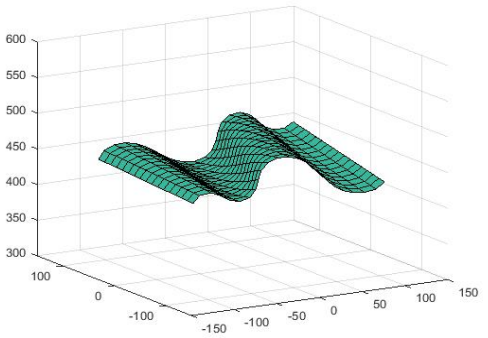
(a) deformed surface of ‘Image1.jpg’



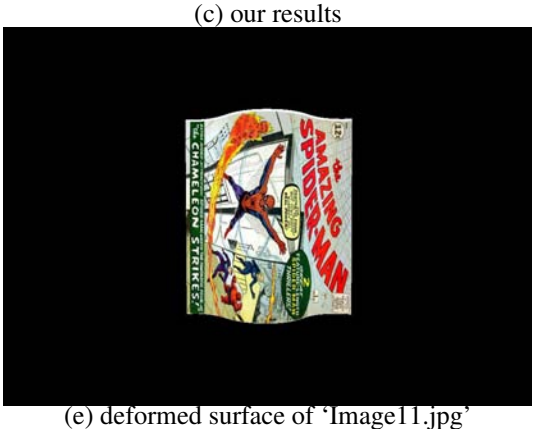
(b) visualizes the error



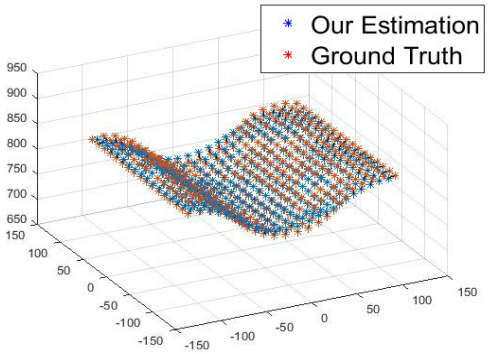
(c) our results



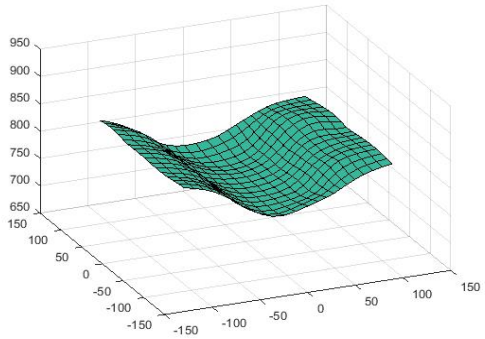
(d) the ground truth



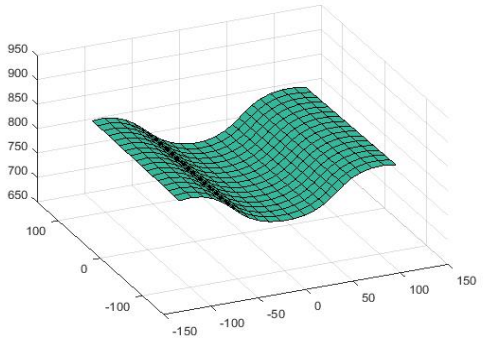
(e) deformed surface of ‘Image11.jpg’



(f) visualizes the error

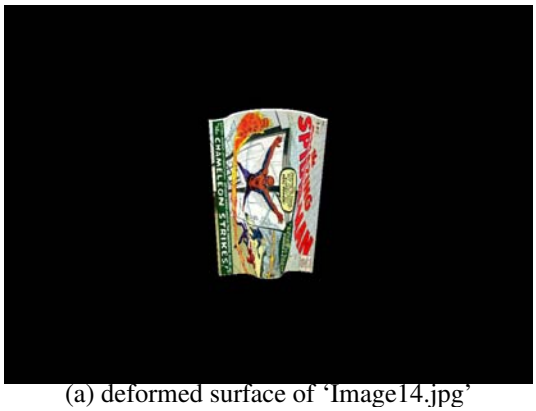


(g) our results

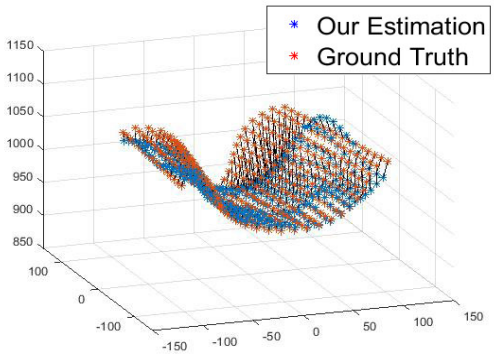


(h) the ground truth

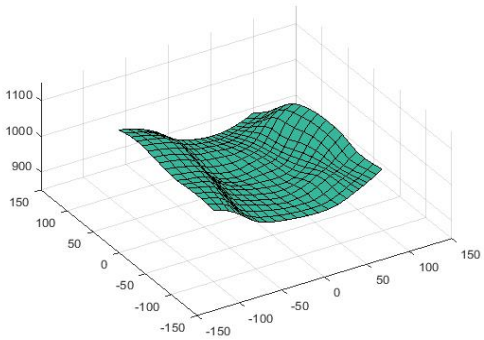
Figure 12: The results of proposed method on artificially deformed Spiderman datasets ‘Image1.jpg’ and ‘Image11.jpg’.



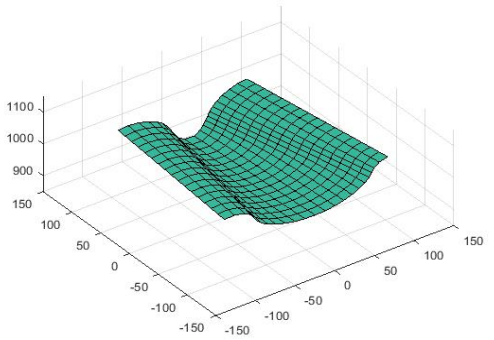
(a) deformed surface of 'Image14.jpg'



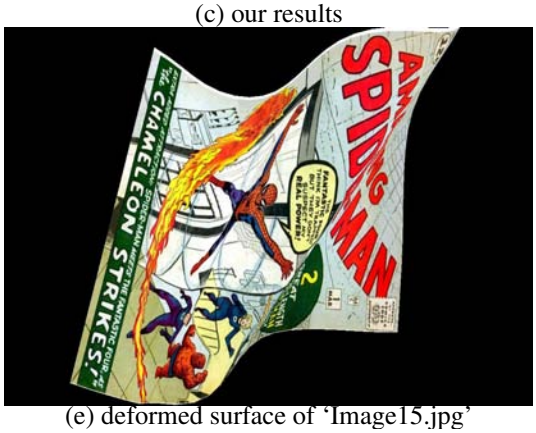
(b) visualizes the error



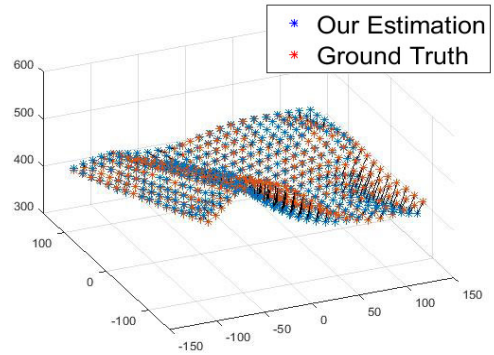
(c) our results



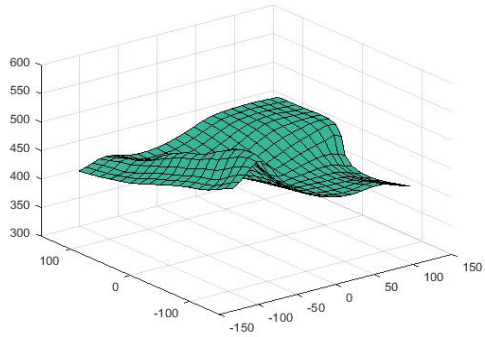
(d) the ground truth



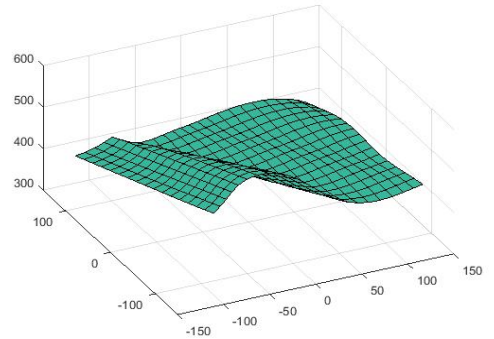
(e) deformed surface of 'Image15.jpg'



(f) visualizes the error



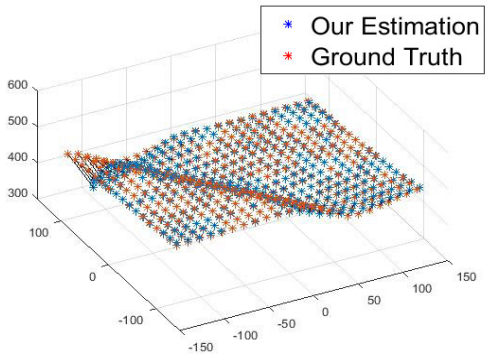
(g) our results



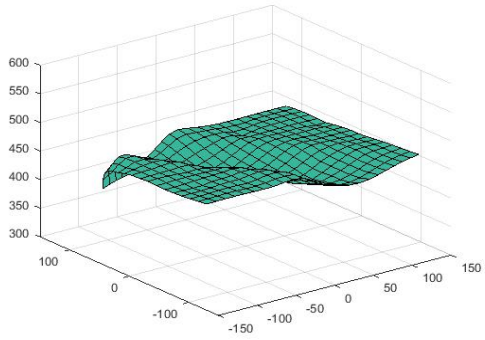
(h) the ground truth

Figure 13: The results of proposed method on artificially deformed Spiderman datasets 'Image14.jpg' and 'Image15.jpg'.

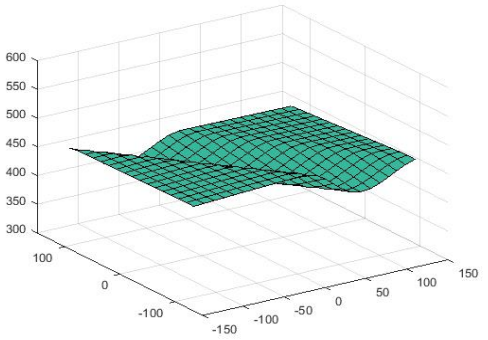
(a) deformed surface of 'Image19.jpg'



(b) visualizes the error



(c) our results



(d) the ground truth

Figure 14: The results of proposed method on artificially deformed Spiderman dataset 'Image19.jpg'.

Table 12: Comparing the accuracy of the proposed method and the one in [23] on Kinect_paper dataset

Method	Average Recunstruction Time (ms)	Average Error (mm)
Our Method	84	6.58
Ngo *et al.* [23]	1662	5.20

is used to evaluate accuracy. It is noticeable that the proposed algorithm is implemented in Matlab and is run on an Intel Core i7 2.4 GHz CPU with 8 GB RAM.

5.1. Real Dataset

To evaluate the result of our method, different datasets are used. Kinect_paper dataset is provided by Varol *et al.*. [24] It includes 193 frames, which are captured using Kinect sensor. The first frame is template image and for all the other frames, the ground truth and the matched pairs with template frame are available.

The template image is planar and there is no occlusion in the frames. For this dataset, since the corresponding points are given, we did not use our matching method to find the new corresponding keypoints.

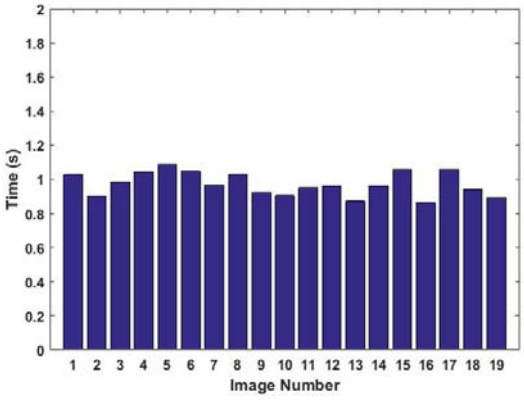


Figure 15: Reconstruction time for artificially deformed Spiderman dataset. In this dataset, 408 control points are considered to reconstruct each surface. Someone can accelerate the algorithm by reducing the number of control points.

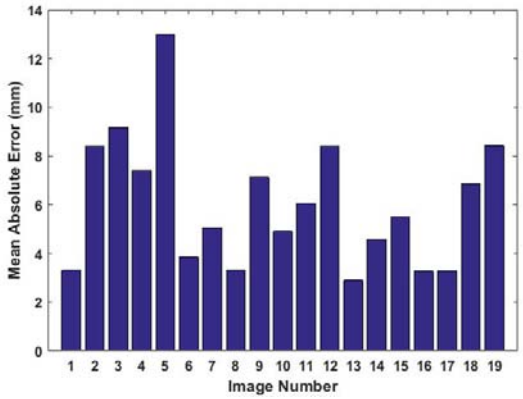


Figure 16: The mean absolute error of proposed method on the artificially deformed Spiderman dataset.

Therefore, nearest keypoints to the control points are selected and their depths are estimated using the rest of corresponding points. Of course, there are some outliers among the provided corresponding points.Figures 9 and 10 show the results with respect to mean absolute error and efficiency for all the sequences and Table 12 shows the average of all. Also, Figure 11 shows the recovered surfaces for 'frame_041', 'frame_071', 'frame_081' and 'frame_101'.

The proposed method can process 14 FPS that means it can be used for online purposes. It should be notified that the time to find the corresponding keypoints are not considered here, since, these points have already been provided in this dataset.

5.2. Artificial Dataset

In addition to the real datasets which have been introduced, we evaluate our proposed method on artificially deformed Spiderman datasets, too. The template image which is used in the results is planar and it is assumed to be perpendicular with the camera principal ray. The distance between virtual camera and the template image is 450mm. Figures 12, 13 and 14 show the results of our proposed method on five artificially deformed Spiderman datasets. Figure 15 presents the required time to extract the deformed surface for Spiderman dataset. For each deformed image of this dataset, 408 control points are used. Of course, to accelerate our algorithm, we can reduce the number of control points. Also, Figure 16 shows the mean absolute error of the deformed images.

6. Conclusion and Future Work

In this paper, a new approach was presented to recover the shape of a deformable surface. The proposed method directly estimates the depth of each point using its local texture. Indeed, to find the Jacobian of the point under deformation, the local texture of control points were used to extract new corresponding points. Then these corresponding points were used to estimate the affine transformation between the original corresponding pair. Finally, the depth of each keypoint was estimated using its location and first-order differential statistics. This method not only is super-fast to reconstruct the surface, but also has good accuracy, comparable with state-of-the-art methods.

Moreover, two new algorithms called Robust Gaussian Model for Geometrical Outlier Removal (RBM_GOR) and Bayesian Robust Mixture Model for Geometrical Outlier Removal (BRMM_GOR) were presented to remove the outliers between the corresponding pairs. Both algorithms are based on Geometrical features. These methods not only detect and remove the outliers, but also find new corresponding points which play a very important role for reconstructing surfaces.

7. Acknowledgements

We want to express our gratitude to the authors of [23] for the Kinect_paper dataset and implementation of their reconstruction method. Also, we would thank Giuseppe Marchioro for his help during this work. This research has received funding from the Ministry of Science, Research and Technology of Islamic Republic of Iran and Image Science for Interventional Techniques laboratory of the Auvergne University of France. This research has received funding from the EUs FP7 through the ERC research grant 307483 FLEXABLE.

References

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

[2] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, "The Geometry of Multiple Images", MIT Press, 2001.

[3] Y. Li, Y. Wang, M. Case, S.F. Chang and P. K. Allen, "Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach", International Conference on Intelligent Robots and Systems, 2014.

[4] B. Frank, C. Stachniss and R. Schmedding, "Matthias Teschner and Wolfram Burgard, Learning object deformation models for robot motion planning", Robotics and Autonomous Systems, Vol. 62, No. 8, pp. 1153-1174, 2014.

[5] A. Malti, A. Bartoli and T. Collins, "Template-Based Conformal Shape-from-Motion from Registered Laparoscopic Images", Medical Image Understanding and Analysis Conference, 2011.

[6] M. Hu, P. Penney, D. Rueckert, P. Edwards, F. Bello, R. Casula, M. Figl and D. Hawkes, "Non-rigid reconstruction of the beating heart surface for minimally invasive cardiac surgery", Medical Image Computing and Computer-Assisted Intervention, Vol. 5761, pp. 34-42, 2009.

[7] T. Collins, B. Compte, A. Bartoli, "Deformable Shape-From-Motion in Laparoscopy using a Rigid Sliding Window", Medical Image Understanding and Analysis Conference, 2011.

[8] M. Salzmann and P. Fua, "Deformable Surface 3D Reconstruction from Monocular Images", Morgan & Claypool Publishers, 2010.

[9] R. Szeliski, "Computer vision: algorithms and applications, Springer-Verlag London", 2011.

[10] F. Moreno-noguer, M. Salzmann, V. Lepetit and P. Fua, "Capturing 3D Stretchable Surfaces from Single Images in Closed Form", Conference on Computer Vision and Pattern Recognition, pp. 1842-1849, 2009.

[11] M. Salzmann, R. Hartley and P. Fua, "Convex optimization for deformable surface 3-D tracking", International Conference on Computer Vision, pp. 1-8, 2007.

[12] A. Bartoli, Y. Gerard, F. Chadebecq, T. Collins and D. Pizarro, "Shape-from-Template", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, Issue 10, pp. 2099-2118, 2015.

[13] A. Bartoli, Y. Gerard, F. Chadebecq and T. Collins, "On Template-Based Reconstruction from a Single View: Analytical Solutions and Proofs of Well-Posedness for Developable, Isometric and Conformal Surfaces", Computer Vision and Pattern Recognition, pp. 2026-2033, 2012.

[14] A. Ecker, A. Jepson, and K. Kutulakos, "Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities", in European Conference on Computer Vision, Vol. 5302, pp. 127-140, 2008.

[15] S. Shen, W. Shi and Y. Liu, "Monocular 3D Tracking of Inextensible Deformable Surfaces Under L2-Norm", Asian Conference on Computer Vision, Vol. 19, Issue 2, pp. 512-521, 2009.

[16] F. Brunet, R. Hartley, A. Bartoli, N. Navab and R. Malgouyres, "Monocular Template-Based Reconstruction of Smooth and Inextensible Surfaces", Asian Conference on Computer Vision, Vol. 6494, pp. 52-66, 2010.

[17] N. Gumerov, A. Zandifar, R. Duraiswami and L. Davis, "Structure of Applicable Surfaces from Single Views", European Conference on Computer Vision, Vol. 3023, pp 482-496, 2004.

[18] J. Liang, D. Dementhon and D. Doermann, "Flattening Curved Documents in Images", Computer Vision and Pattern Recognition, pp. 338345, 2005.

[19] M. Perriollat and A. Bartoli, "A Computational Model of Bounded Developable Surfaces with Application to Image-Based Three-Dimensional Reconstruction", Computer Animation and Virtual Worlds, vol. 24, no. 5, 2013.

[20] I. A. Bachelder and S. Ullman, "Contour matching using local affine transformations", Computer Vision and Pattern Recognition, IL, pp. 798-801, 1992.

[21] M. Salzmann, Toyota Technol. Inst. at Chicago, Chicago, IL, USA, P. Fua, "Linear Local Models for Monocular Reconstruction of Deformable Surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, Issue 5, pp. 931-944, 2011.

[22] J. Ostlund, A. Varol, D. Ngo and P. Fua, "Laplacian Meshes for Monocular 3D Shape Recovery", European Conference on Computer Vision, pp 412-425, 2012.

[23] D. Ngo, J. Ostlund and P. Fua, "Template-based Monocular 3D Shape Recovery using Laplacian Meshes", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015.

[24] A. Varol, A. Shaji, M. Salzmann and P. Fua, "Monocular 3D Reconstruction of Locally Textured Surfaces.", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, Issue 6, pp. 1118-1130, 2012.

[25] A. Locher, L. Elsen, X. Boix and L. V. Gool, "Reconstruction of Inextensible Surfaces on a Budget via Bootstrapping", International Conference on 3D Vision, Vol. 1, pp. 240-247, 2014.

[26] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool and B K. U. Leuven, "Speeded-Up Robust Features (SURF)", Journal Computer Vision and Image Understanding, Vol. 110, Issue 3, pp. 346-359, 2008.

[27] M. Ozuysal, M. Calonder, V. Lepetit and P. Fua, "Fast Keypoint Recognition Using Random Ferns", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, Issue 3, pp. 448-461, 2010.

[28] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF", International Conference on Computer Vision, pp. 2564-2571, 2011.

[29] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, pp. 91-110, 2004.

[30] K. M. Yi, E. Trulls, V. Lepetit and P. Fua, "LIFT: Learned Invariant Feature Transform", European Conference on Computer Vision, Vol. 9910 , pp. 467-483, 2016.

[31] E. Rosten and T. "Drummond, Machine Learning for High-Speed Corner Detection", European Conference on Computer Vision, Vol. 3951, pp. 430-443, 2006.

[32] T. Collins, A. Bartoli, "Using Isometry to Classify Correct/Incorrect 3D-2D Correspondences", European Conference on Computer Vision, Vol. 8692, pp. 325-340, 2014.

[33] A. Bartoli, M. Perriollat and S. Chambon, "Generalized Thin-Plate Spline Warps", International Journal of Computer Vision, Vol. 88, No. 1, pp. 85-110, May 2010.

[34] T. Collins and A. Bartoli, "Realtime Shape-from-Template: System and Applications", IEEE International Symposium on Mixed and Augmented Reality, pp. 116-119. 2015.

[35] M. Perriollat and A. Bartoli, "A Computational Model of Bounded Developable Surfaces with Application to Image-Based 3D Reconstruction", Computer Animation and Virtual Worlds, Vol. 24, No. 5, pp. 459-476, 2013.

[36] Q. H. Tran, T. J. Chin, G. Carneiro, M. S. Brown and D. Suter, "In Defence of RANSAC for Outlier Rejection in Deformable Registration", European Conference on Computer Vision, Vol. 7575, pp. 274-287, 2012.

[37] D. Pizarro and A. Bartoli, "Feature-based deformable surface detection with self-occlusion reasoning", International Journal of Computer Vision , Vol. 97, Issue 1, pp. 54-70, 2012.

[38] G. Agamennoni, "Bayesian Clustering with Outliers and Missing Values", Technical Report ACFR-TR-2013-001. Australian Centre for Field Robotics, University of Sydney, NSW Australia, 2013.