Article

Depth Estimation of a Deformable Object via a Monocular Camera

Guolai Jiang ^{1,2}, Shaokun Jin ^{1,2,*}, Yongsheng Ou ^{1,3,4,*} and Shoujun Zhou ¹

¹ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; gl.jiang@siat.ac.cn (G.J.); sj.zhou@siat.ac.cn (S.Z.)

² Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China

³ Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

⁴ Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

* Correspondence: sk.jin@siat.ac.cn (S.J.); ys.ou@siat.ac.cn (Y.O.)

Received: 28 January 2019; Accepted: 26 March 2019; Published: 1 April 2019

Abstract: The depth estimation of the 3D deformable object has become increasingly crucial to various intelligent applications. In this paper, we propose a feature-based approach for accurate depth estimation of a deformable 3D object with a single camera, which reduces the problem of depth estimation to a pose estimation problem. The proposed method needs to reconstruct the target object at the very beginning. With the 3D reconstruction as an a priori model, only one monocular image is required afterwards to estimate the target object's depth accurately, regardless of pose changes or deformability of the object. Experiments are taken on an NAO robot and a human to evaluate the depth estimation accuracy by the proposed method.

Keywords: depth estimation; deformable; 3D reconstruction; monocular; pose estimation

1. Introduction

In the case of human–robot cooperation, deep reinforcement learning (DRL) is used to train the robot to undertake the task. For an instance of the bolt screwing task, the human partner's arms might be obstacles for the robot during the working process (Figure 1). In order to train the capability of avoiding obstacles for the robot through DRL method, one has to prepare a huge number of samples, which are usually hard to generate. In this case, it can be done by reconstructing the 3D imaging [1–3] of a human worker executing the task. The reconstruction sequence of the human's arms can be used as moving obstacles to train the obstacle avoidance capability of the robot in a virtual environment (Figure 2).

In the scenarios above, a common prerequisite is the accurate pose information of a human or a robot. However, when an object is projected onto the camera plane, its depth information along the optical axis is lost, which possibly makes two actually far separated objects look close to each other [4]. This results in incorrect estimation of the pose without correct depth information. Although lasers are used to provide depth information, they are usually expensive. Low-cost equipment such as a Kinect are also prevailing alternatives, but their accuracy is not high enough (Kinect has an error of at least 4 mm and has a dead zone within 0.5 m. Additionally, the farther the distance is, the more inaccurate the measuring result is).

Many previous studies propose different approaches to produce depth estimation from images. Among them, a few researchers used optimization methods to handle the problem. Ranftl et al. propose a novel motion segmentation method to produce dense depth map from two consecutive frames

through a single monocular camera. They segment the optical flow field into a set of motion models, with which the scene is reconstructed by minimizing a convex program [5]. Smith et al. present a method which estimates depth from a single polarisation image by solving a large, sparse system of linear equations [6]. Karsch proposes a technique that automatically generates plausible depth maps from videos using non-parametric depth sampling [7]. Optimization methods need the manually defined constraints to guarantee the accuracy of the resulted depth estimation. Thus, these methods require human experiences to give effective constraints.

Figure 1. The tool used in a human–robot cooperation scenario for screwing a bolt. Since the human and the robot together screw the cross tool, the human arm might be obstacles to the robot during the bolt screwing process.

Figure 2. 3D reconstruction for training a cooperative robot, which provides training samples and allows the robot to learn to avoid obstacles in a virtual environment.

Additionally, approaches using learning-based methods to model manually extracted features are other promising alternatives as well. Saxena et al. propose a Markov Random Field (MRF) learning algorithm to handle monocular cues like texture gradients and variations, defocus, etc. They incorporate these cues into a stereo system to obtain depth estimation results [8]. Ma et al. improve the ResNet-50 network by transfer learning to tackle the depth estimation problem through a single image [9]. Haim et al. propose a phase-coded aperture camera for depth estimation. They equip the camera with an optical phase mask to produce unambiguous depth-related color characteristics for the captured image [10]. Gan et al. present a convolutional neural network architecture that pays more attention to the relationships of different image locations and incorporates the absolute and relative features [11]. These methods depend on the extracted features. In some scenarios, these features might not incorporate sufficient cues of the depth information.

Recently, approaches using deep learning to generate depth maps from images become prevailing. Fu et al. propose a space-increasing discretization (SID) strategy to discretize depth and recast depth network learning as an ordinal regression problem [12]. Jiao et al. propose an approach to handle the depth estimation and semantic labeling task simultaneously. They present a concept called the attention-driven loss for the network supervision, and a synergy network to learn the relevance between the two tasks [13]. Godard et al. trained an unsupervised deep neural network with binocular stereo data to address the ground truth depth data deficient problem in traditional depth estimation

methods. A novel training loss was proposed for the deep convolutional neural network to perform single image depth estimation with a high quality [14]. Inspired from the concept of autoencoders, Garg et al. trained the first convolutional neural network end-to-end from scratch for single view depth estimation in an unsupervised manner [15]. Deep learning methods are powerful tools for regressions. However, they usually need expensive platforms to run the algorithm and they usually need many training samples related to the target object for accurate depth estimation.

Different from the aforementioned previous approaches, the proposed method in this paper estimates depth information by means of 3D reconstruction. The first thing we need to do is to circle a camera around the target object and get its reconstruction at the very beginning. Afterwards, only a single monocular image is required to accurately estimate the depth information of the object no matter how it moves or becomes deformable in front of the camera. The proposed method is therefore more applicable in the scenarios where the target object is not rigid and where accurate depth information is necessary. With the point cloud of a target object (in a certain static pose) reconstructed beforehand, the proposed method can estimate the pose and reconstruct the point cloud of a target object (in its other poses) by a single input RGB image. This work can be used on not only humans and humanoid objects, but on other deformable objects as well.

The remainder of this paper is structured as follows. Sections 2–4 individually introduce the three modules of the proposed approach specifically. Section 5 introduces the whole flow chart of the proposed method. Experimental evaluations are provided on a NAO robot and a human in Section 6. Section 7 concludes the contributions of this paper.

2. 3D Labeled Reconstruction

This section introduces the priori model for the target object. The priori model is the 3D reconstruction (stored in the form of the point cloud [16–18]) of the target object in its stationary status, with a SIFT feature vector attached to each cloud point. Therefore, the priori model is built through two steps. First, we use a traditional 3D reconstruction approach (for static object) to reconstruct the target object by multiple images. Second, we use the SIFT algorithm [19] to extract feature vectors from the collected images, attaching them to the corresponding 3D points on the reconstructed point cloud.

2.1. 3D Reconstruction with Multiple Images

Given a static object, we can use a single camera (whose intrinsic parameter is f) to circle around the object, reconstructing a point cloud. In order to reconstruct the object, the total number of the images captured by the camera is N . Additionally, the total number of points on the object surface is M . The orientation and position of the camera with regard to the world frame at the i th instant can be respectively represented by a matrix R_i and a vector t_i . Denote $P_j = \begin{pmatrix} X_j & Y_j & Z_j \end{pmatrix}^T$ as the j th point on the object surface with regard to the world frame, $P_j^i = \begin{pmatrix} X_j^i & Y_j^i & Z_j^i \end{pmatrix}^T$ as the same j th point with regard to the camera frame at the i th instant, and $m_j^i = \begin{pmatrix} x_j^i & y_j^i \end{pmatrix}^T$ as the image coordinate of j th point at the i th instant (for denotation simplification, we define $m_j^i \in \emptyset$ if P_j is occluded under the observation by the camera at i th instant). The following can be given [20]:

$$\begin{bmatrix} P_j^i \\ 1 \end{bmatrix} = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_j \\ 1 \end{bmatrix}, \quad (1)$$

$$Z_j^i \begin{bmatrix} m_j^i \\ 1 \end{bmatrix} = \begin{bmatrix} f^i I_{2 \times 2} & O_{2 \times 1} \\ O_{1 \times 2} & 1 \end{bmatrix} P_j^i. \quad (2)$$

Let

$$v_j^i = Z_j^i m_j^i - \begin{bmatrix} f^i I_{2 \times 2} & O_{2 \times 1} \end{bmatrix} \begin{bmatrix} R_i & t_i \end{bmatrix} \begin{bmatrix} P_j^T & 1 \end{bmatrix}^T. \quad (3)$$

Subsequently, the desired result in this step is the 3D reconstruction $\{P_j^*\}_{j=1\dots M}$ in the following form:

$$\{P_j^*\}_{j=1\dots M} = \arg \min_{\{P_j\}_{j=1\dots M}} \sum_i^N \sum_j^M \|v_j^i\|. \quad (4)$$

2.2. SIFT Features to Label the 3D Reconstruction

Given a two-dimensional image $I(x, y)$, the SIFT algorithm [19] is able to extract effective key points through the LoG operator. By computing the gradients in the neighborhood of each key point, a corresponded descriptor vector can be obtained to distinguish the key point. Then, we can use the SIFT algorithm to find a set of feature points (denoted as $\{m_s^i\}_{s=1\dots S}$) and their corresponding descriptor vectors (denoted as $\{l_s^i\}_{s=1\dots S}$) for the image captured at the i^{th} instant, which jointly yield a two-tuple set denoted as $\{(m_s, l_s)\}_{s=1\dots S}$. Executing the same operation to all the images, we can finally get $\{(m_s^i, l_s^i)\}_{i=1\dots N, s=1\dots S^i}$ (where S^i indicates the total number of feature points derived from the SIFT algorithm for the image captured at the i th instant).

Subsequently, we need to attach the descriptor vectors to the corresponding 3D points on the surface of the reconstructed point cloud. It can be deduced from Equations (1) and (2) that

$$P_j = R_i^T \begin{bmatrix} \frac{d_j^i}{f^i} I_{2 \times 2} & O_{2 \times 1} \\ O_{1 \times 2} & d_j^i \end{bmatrix} \begin{bmatrix} m_j^i \\ 1 \end{bmatrix} - R_i^T t_i. \quad (5)$$

By Equation (5), we can determine the 3D point P_s^i on the reconstructed point cloud corresponding to the feature point m_s^i . Thus, we can acquire a two-tuple set $\{(P_s^i, l_s^i)\}_{i=1\dots N, s=1\dots S^i}$. For denotation simplification, we define that any P_j that is occluded from the camera view at the i th instant, or whose corresponding m_j^i is not a key point, still has a descriptor vector $l_j^i = 0$. Therefore, we can get $\{(P_j^i, l_j^i)\}_{i=1\dots N, j=1\dots M}$ after the N th instant. The required 3D labeled reconstruction is the following:

$$\{(P_j^*, l_j^*)\}_{j=1\dots M} = \{(P_j^*, \bar{l}_j)\}_{j=1\dots M'} \quad (6)$$

where $\bar{l}_j$ represents the average vector over all the non-zero descriptor vectors (i.e., $\{l_j^i \mid i \leq N, l_j^i \neq 0\}$) related to the 3D point P_j within the N th sampling instant.

3. Skeleton-Based Topological Segmentation

This section introduces how to provide the reconstructed point cloud with a robust topological segmentation, so as to deal with the case where the target object is not rigid. In this way, each sub-point-cloud derived from the topological segmentation is expected to be rigid. Topological segmentation based on the surface information of the target object is inclined to getting influenced by the surface noise, causing weak robustness. Therefore, the proposed topological segmentation is executed through two steps. First, we extract the skeleton of the reconstructed point cloud [21] and segment the skeleton based on its curvature. Second, we dilate the sub-skeletons [22] to yield the sub-point-clouds, which are the results of the topological segmentation.

Skeleton extraction and segmentation. Given an object denoted as $\Omega = \{P_j^*\}_{j=1\dots M'}$, we denote $CORE(\Omega)$ as the set of all the maximally inscribed spheres in Ω , none of which has common points of tangency with noisy surface. Then, the skeleton of Ω is denoted as $S(\Omega)$.

After extracting the skeleton of the reconstructed point cloud, we segment the skeleton according to its curvature. Supposing $C(\Omega) \subset S(\Omega)$, an equivalence relation $\sim_C$ implicated by $C(\Omega)$ is defined

as $\forall p_1, p_2 \in S(\Omega)$, $p_1 \sim_{C_1} p_2$ if and only if p_1, p_2 are on a curve segment whose ends are two points in $C(\Omega)$ and no other points in $C(\Omega)$ are on the same curve segment.

Thus, the curve segments determined by $C(\Omega)$ are the equivalence classes [23] $S_C(\Omega)$ defined as

$$S_C(\Omega) = S(\Omega) / \sim_C = \{\{p \in S(\Omega) : p \sim_C q\} : q \in S(\Omega)\}. \quad (7)$$

It is clear that the curve segments in $S_C(\Omega)$ are separated from each other. In this paper, we propose two categories of points, respectively, denoted as $C_1(\Omega)$ and $C_2(\Omega)$ in order that $C(\Omega) = C_1(\Omega) \cup C_2(\Omega)$ determines $\sim_C$ in Equation (7).

3.1. The First Category $C_1(\Omega)$

Supposing all the points on the skeleton $S(\Omega)$ constitute a set $\{p_k\}_{k=1\dots K}$ (K is the total number of elements in $S(\Omega)$). We use a set $\{e_{mn}\}_{\substack{0 < m \leq K \\ 0 < n \leq K}}$ to represent the connectivity of each two points from $S(\Omega)$. Specifically, $e_{mn} = 1$ if p_m and p_n are adjacent to each other. Otherwise, $e_{mn} = 0$. Subsequently, the first category of points $C_1(\Omega)$ is defined as

$$C_1(\Omega) = \left\{ p_m \in \{p_k\}_{k=1\dots K} \mid \sum_n^K e_{mn} > 2 \text{ or } \sum_n^K e_{mn} = 1 \right\}. \quad (8)$$

3.2. The Second Category $C_2(\Omega)$

Similar to $S_C(\Omega)$ (Equation (7)), $S(\Omega)$ segmented by $C_1(\Omega)$ is

$$S_{C_1}(\Omega) = S(\Omega) / \sim_{C_1}. \quad (9)$$

We define the function of the c th curve segment (suppose C curve segments totally) in $S_{C_1}(\Omega)$ as $r_c(u_c)$, where u_c is the arc length parameter of the function r_c and $u_c \in (0, L_c)$ (L_c is the length of the c th curve segment). Then, the Frenet formulas [24] of the c th curve is

$$\frac{d}{du_c} \begin{bmatrix} r_c \\ \alpha_c \\ \beta_c \\ \gamma_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \kappa_c & 0 \\ -\kappa_c & 0 & \tau_c \\ 0 & -\tau_c & 0 \end{bmatrix} \begin{bmatrix} \alpha_c \\ \beta_c \\ \gamma_c \end{bmatrix}, \quad (10)$$

where α_c, β_c and γ_c are, respectively, the unit vector tangent, normal unit vector and binormal unit vector of r_c ; κ_c and τ_c are the curvature and torsion.

We construct a quantity δ_c satisfying that $\delta_c \propto \kappa_c$ and $\delta_c \propto \tau_c$ as well as a threshold v . Subsequently, the second category of points $C_2(\Omega)$ is defined as

$$C_2(\Omega) = \{p = r_c(u_c) \mid \delta_c \geq v\}_{c=1\dots C}. \quad (11)$$

3.3. Skeleton Dilation with Constraints

Supposing B is a structuring element [20] in the form of a subset in R^3 , the dilation of $S(\Omega)$ by B after T iterations is defined as

$$B_T(S(\Omega)) = \bigoplus_{t=1\dots T} B \oplus S(\Omega). \quad (12)$$

The dilation operation is executed to segment the reconstructed point cloud by the sub-skeletons. Therefore, sub-point-clouds should be separated from each other. Additionally, the dilation should be stopped when reaching the surface of the reconstructed point cloud. Thus, in each iteration $t \in \{1\dots T\}$, we remove the points that violate the following two constraints:

$$B_t(S_{c_1}(\Omega)) \cap B_t(S_{c_2}(\Omega)) = \emptyset, \quad (13)$$

$$B_t(S_{c_1}(\Omega)) \cap \bar{\Omega} = \emptyset, \quad (14)$$

where $S_{c_1}(\Omega)$ and $S_{c_2}(\Omega)$ represent two distinct sub-skeletons in $S_C(\Omega)$, $\bar{\Omega}$ is the external space of Ω .

Moreover, for the dilation of each sub-skeleton $S_{c_1}(\Omega)$ under constraints Equations (13) and (14), the corresponding total iteration times T satisfy that

$$B_{T+1}(\Omega) = B_T(\Omega), \quad B_T(\Omega) - B_{T-1}(\Omega) \neq \emptyset. \quad (15)$$

Then, the dilation result of sub-skeletons in $S_C(\Omega)$ under constraints in the form of Equations (13)–(15) forms a equivalence relation for topological segmentations $(\Omega, \mathcal{T})$.

4. 3D Reconstruction at i th Time

This section introduces how to quickly reconstruct the dynamic object with a single RGB camera. For each frame by the camera, we first extract all the feature points. Through matching these feature points to those on the reconstructed point cloud, we are able to know to which sub-point-clouds these feature points individually correspond. Finally, the poses of the sub-point-clouds can be estimated by the correspondences between the feature points on the point cloud and the image feature points. This pose estimation problem can be easily handled by solving a nonlinear optimization. The reconstruction is therefore reduced to the reorganization of the sub-point-clouds with updated poses.

Denote the image captured at i th instant ($i > N$) as $\{m_i^j\}_{j=1 \dots M}$. Through the SIFT algorithm, we can extract the feature points and their corresponding descriptor vectors. Through Section 2.2, a labeled 3D point cloud is reconstructed, to which the descriptor vectors are attached. Thus, we can use the descriptors from the captured image and from the cloud to find their correspondence, which can be denoted as a two-tuple set $\{(m_i^{f_i}, P_{f_i}^*)\}_{f_i=j_1 \dots j_F}$. In this set, $m_i^{f_i}$ and $P_{f_i}^*$ respectively represent the feature point from the image and the cloud point, whose descriptor vectors are similar. In addition, $\{(m_i^{f_i}, P_{f_i}^*)\}_{f_i=j_1 \dots j_F}$ yields a bijective map $\varphi_i: \{m_i^{f_i}\}_{f_i=j_1 \dots j_F} \rightarrow \{P_{f_i}^*\}_{f_i=j_1 \dots j_F}$.

Suppose $\mathcal{T} = \{[p] : p \in \Omega\} = \{\{q \in \Omega : q \sim p\} : p \in \Omega\}$ as the basis for the topology of space $\Omega = \{P_j^*\}_{j=1 \dots M}$. We can get the basis for topology (denoted as $\mathcal{T}_i$) of the subset $\{P_{f_i}^*\}_{f_i=j_1 \dots j_F} \subset \{P_{f_i}^*\}_{j=1 \dots M}$ by

$$\mathcal{T}_i = \left\{ T_c \cap \{P_{f_i}^*\}_{f_i=j_1 \dots j_F} \mid T_c \in \mathcal{T} \right\}. \quad (16)$$

We can further acquire the basis for the topology (denoted as $\mathcal{T}_{m_i}$) of the set $\{m_i^{f_i}\}_{f_i=j_1 \dots j_F}$ by

$$\mathcal{T}_{m_i} = \left\{ \varphi_i^{-1}(T_{m_i}) \mid T_{m_i} \in \mathcal{T}_i \right\}. \quad (17)$$

Based on our design, each element in $\mathcal{T}$ is a rigid component of Ω . Thus, when the object represented by Ω moves stochastically, the elements in $T_c \in \mathcal{T}$ have the same rigid transformation, i.e., there exists a single pair (R_c^i, t_c^i) for T_c^i such that

$$T_c^i = R_c^i T_c + t_c^i, \quad (18)$$

where R_c^i is a three-dimensional rotation matrix, t_c^i is a three-dimensional translation vector and $R_c^i T_c^i + t_c^i$ is defined as

$$R_c^i T_c^i + t_c^i = \left\{ R_c^i p + t_c^i \mid p \in T_c^i, T_c^i \in \mathcal{T} \right\}, \quad (19)$$

where p is denoted in the form of a column vector.

Defining $T_{c_i} = T_c \cap \left\{ P_{f_i}^* \right\}_{f_i=j_1 \dots j_F}$, since $T_{c_i} \subset T_c$, based on Equation (18), we can get the actual coordinates of $T_{c_i}^i$ at i^{th} instant as

$$T_{c_i}^i = R_i^c T_{c_i} + t_i^c. \quad (20)$$

Then, we can get another expression of $\varphi^{-1}(T_{c_i})$ as

$$\varphi^{-1}(T_{c_i}) = \left\{ m_i(p) \mid p \in T_{c_i} \right\}, \quad (21)$$

where $m_i(\cdot)$ (based on Equation (2)) is an operator to transform a three-dimensional point to a two-dimensional coordinate related to the camera at i^{th} instant, which is defined as

$$\forall p \in R^3, \quad m_i(p) = \frac{f^i}{\begin{bmatrix} O_{1 \times 2} & 1 \end{bmatrix} p} \begin{bmatrix} I_{2 \times 2} & O_{2 \times 1} \end{bmatrix} p. \quad (22)$$

Thus, we can compute a (R_i^c, t_i^c) for $T_{c_i}^i$ by solving the following optimization problem:

$$(R_i^c, t_i^c) = \arg \min_{(R_i^c, t_i^c)} \sum_{p \in T_{c_i}} \left[\varphi^{-1}(p) - m_i(R_i^c p + t_i^c) \right]^2. \quad (23)$$

Finally, the raw result of the 3d reconstruction $\left\{ \tilde{P}_i^j \right\}_{j=1 \dots M}^{i \geq N}$ based on $\left\{ m_i^{f_i} \right\}_{f_i=j_1 \dots j_F}$ at i^{th} time is

$$\left\{ \tilde{P}_i^j \right\}_{j=1 \dots M}^{i \geq N} = \bigcup_{T_{c_i} \in \mathcal{T}} (R_i^c T_{c_i} + t_i^c). \quad (24)$$

5. Approach Overview

Specifically, we utilize a single camera to resolve the dynamic 3D object reconstruction. The problem can be formulated as: given the images $\left\{ m_i^j \right\}_{i=1 \dots N, j=1 \dots M}$ captured within time (the pose and position alters at each time $i \leq N$ in order that static 3D reconstruction of the object can be satisfactorily achieved) and images captured at time $i > N$, the expected result is the dense reconstruction of the object $\left\{ P_i^j \right\}_{i \geq N, j=1 \dots M}$ at the same time $i > N$.

Accordingly, we consider to fully utilize the 3D information acquired from the foregoing frames and then reduce the dynamic object reconstruction problem to a re-organizing problem. Thus, the proposed approach mainly incorporates three steps.

In the first step, we obtain the static 3D reconstruction of the target object in its stationary status. Specifically, the point cloud of the target object is acquired in the first phase through existing static 3D reconstruction methods. Meanwhile, we obtain the SIFT features on each image (among images utilized for 3D reconstruction) and attach the feature descriptors to the corresponding points on the reconstructed point cloud.

In the second step, we find an appropriate topological segmentation for the reconstructed point cloud such that each topological part moves rigidly during the object motion. Actually, the point cloud topological segmentation is also an open problem due to the fact that it is difficult to determine the standard for a satisfactory segmentation. In this paper, we transfer the point cloud segmentation problem into the skeleton segmentation problem, i.e., the segmentation of the point cloud results from the segmentation of its skeleton. This is based on the thought that the object skeleton is much more stable toward perturbation than the object surface. Thus, we can simply segment the skeleton into several sub-skeletons based on its curvature and torsion property before dilating each sub-skeleton to determine a corresponding topological part of the point cloud.

In the third step, when capturing a new image, we extract the SIFT features and match these features to those attached to the point cloud in the first phase. These matched features implicate the correspondence between the image and the point cloud. Based on the topological segmentation executed on the point cloud, the correspondence between the image and each topological part can also be computed. Then, the pose and position of each topological part can be deduced, and the reconstruction result can thus be obtained through a simple re-organization of these topological parts. The whole flow chart of the proposed approach is illustrated in Figure 3.

Figure 3. The framework of the proposed method.

Construction of rotation matrix. Since the analytic expression of rotation matrix R_i^c in Equations (18), (20), (23) and (24) is required for optimization program (following Equation (23)), we utilize the quaternion to construct the rotation matrix. Specifically, a quaternion is in the form of

$$q = w + x \cdot \vec{i} + y \cdot \vec{j} + z \cdot \vec{k}. \quad (25)$$

Then, the corresponding rotation R_i^c in the optimization problem of Equation (23) is

$$R_i^c = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(x^2 + z^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix}, \quad (26)$$

with a constraint as

$$w^2 + x^2 + y^2 + z^2 = 1, \quad (27)$$

which constitutes a simple constrained nonlinear optimization problem.

6. Experiments and Discussion

This section provides the experiment results from the proposed approach. We evaluate the proposed approach by a set of experiments on a NAO robot Figure 4 and a human being.

Figure 4. Experiments on a NAO robot. (a) shows the robot as the target object to evaluate the algorithms; (b) shows the priori model reconstructed at the beginning.

6.1. Experiments on a NAO Robot

NAO robot is an autonomous, programmable humanoid robot developed by Aldebaran Robotics (France), with a height of 58 centimetres and 25 degrees of freedom. We use a NAO robot as the target object for 3D reconstruction. During the experiment, the NAO robot continuously changes its poses such that it is deformable. We reconstruct the NAO robot to test the depth accuracy by the proposed algorithm.

This experiment is undertaken on a single monocular camera to reconstruct the NAO robot in its dynamic status. In order to guarantee the effectiveness of the proposed method, several specific processing procedures are listed in Appendices A–C.

Other procedures of the proposed approach in experiments follow the descriptions and formulas in Sections 2–4.

We compare the proposed method with approaches in [14,25]. The depth estimation result by the proposed method is shown in Figure 5. Note that the proposed method only estimates the depth information of the target object while the previous approaches estimate the depth of the whole image. Therefore, we only compare the depth estimation accuracies corresponding to the image regions where the target object appears. Four accuracy metrics are used [26] as shown in Table 1.

Table 1. Quantitative comparison on the aspects of accuracy and efficiency.

Metrics Names	Equations
Abs Relative Difference	$\frac{1}{T} \sum_{y \in T} y - y^* / y^*$
Squared Relative difference	$\frac{1}{T} \sum_{y \in T} y - y^* ^2 / y^*$
RMSE (linear)	$\sqrt{\frac{1}{T} \sum_{y \in T} y - y^* ^2}$
RMSE (log)	$\sqrt{\frac{1}{T} \sum_{y \in T} \log y - \log y^* ^2}$

Figure 5. The depth estimation results for the NAO robot by the approaches in [14,25] and the proposed method. We use the ground truth to build a mask so as to only display the depth information corresponding to the image region of the NAO robot.

The depth accuracies of the proposed method and other three approaches are shown in Table 2.

Table 2. Quantitative comparison on the aspects of accuracy and efficiency.

Approaches	Abs Relative Difference	Squared Relative Difference	RMSE (linear)	RMSE (log)
The approach in [25]	4.8693	510.8358	452.1486	3.3712
The approach in [14]	1.1972	53.7397	182.3361	1.2484
The proposed approach	0.9987	31.7892	152.0740	1.0743

6.2. Experiments on a Human

Experiments are also undertaken on a human to verify the proposed approach. We produce the 3D reconstruction of the human in his still pose at the very beginning. Subsequently, the human changes his pose and the camera captures the corresponding monocular image for each pose. We use these images to test the proposed approach and the previous algorithms [14,25]. The program to implement this experiment is the same as that for the NAO robot experiment.

The depth estimation results by the proposed method and other approaches are shown in Figure 6. Additionally, the depth accuracies of the proposed method and other three approaches are shown in Table 3.

Table 3. Quantitative comparison on the aspects of accuracy and efficiency.

Approaches	Abs Relative Difference	Squared Relative Difference	RMSE (linear)	RMSE (log)
The approach in [25]	5.9473	638.9376	367.0673	3.7256
The approach in [14]	2.0792	197.2335	216.7147	2.1638
The proposed approach	1.8274	83.5462	187.4539	2.0132

Figure 6. The depth estimation results for the human volunteer by the approaches in [14,25] and the proposed method.

Similar to the experiments taken on NAO robot, the experiments results can validate the effectiveness of the proposed method. The proposed method uses the 3D point cloud reconstructed ahead of time as the priori model. Subsequently, it depends on the image features to estimate the pose changes of the priori model, therefore being capable of estimating the depth information accurately.

7. Conclusions

In this paper, we propose a feature-based approach to estimate the depth of a deformable object accurately via a monocular camera. The proposed approach needs to reconstruct the target object in its initial pose as a priori model. Afterwards, only one monocular image is required to accurately estimate the depth of the target object no matter how it changes its pose. Experiments are undertaken on a NAO robot and a human to evaluate the accuracy of the proposed approach. In future work, we aim to accurately estimate depth of the same kind of deformable object, by only reconstructing a single instance of that kind as the priori model.

Author Contributions: The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. conceptualization, methodology and writing—original draft preparation: G.J.; software, validation and writing—review and editing: S.J.; investigation, resources, software and supervision: Y.O.; the investigation, resources and validation: S.Z.

Funding: This work was supported by the National Natural Science Foundation of China Grant No. U1613210, U1813208, the Shenzhen Fundamental Research Programs (JCYJ2016428154842603, JCYJ20170413165528221), and the Shenzhen Engineering Laboratory for Integration of Interventional Diagnosis and Treatment.

Conflicts of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Appendix A. SIFT Matching Accuracy

The SIFT algorithm, in spite of its effectiveness validated in [19], cannot actually match features from two images enough accurately, especially when a corresponding object in two images has deformation. Thus, in order to ameliorate the result of feature matching (crucial as a prerequisite of our approach), we propose the following engineering improvements:

For a certain feature point, we collect the 10 other feature points closest to it on the image. The 11 feature points forms a group. In this way, each feature point on the image can constitute a group, with itself as the center of the group. It is easy to consider that, if two feature points from two images match each other, their corresponding groups match each other as well.

When judging whether two feature points from different images are matched, we compare their groups. Specifically, we first look for two points respectively from two groups, which matched each other best (the matching degree is decided by the SIFT descriptor vectors of the feature points.). Recording the matching degree, we delete the two points from their groups. We continue looking for the next two points that are matched best in the same way. This is iteratively done until there are no more points in both groups. We accumulate the matching degrees of the feature point pairs as the similarity of the original two feature points.

Appendix B. Discrete Computation of Curvature and Torsion

In Equation (11), since δ_c should satisfy that $\delta_c \propto \kappa_c$, $\delta_c \propto \tau_c$, we compute δ_c in discrete situations when programming as: for any voxel (namely a 3D point restored in computers), we choose the 5th point individually from it along both sides. Construct two vectors respectively pointing from the voxel to the two chosen voxels. Then, the cosine value of the two vectors is selected as δ_c for simplicity.

Appendix C. Result Optimization for Smoothness

The result from Equation (23) can basically represent the pose of the target object, but the joint of two connected topological parts may be too coarse to be the final result. Thus, we propose a simple algorithm to smooth the joint of two connected topological parts. The procedures can be described as:

When the local point cloud is segmented into two topological parts, we know the correspondence between all the points on both borders. After solving the optimization problem as Equation (23), the relative pose of the two topological parts can also be acquired. Then, the new correspondence between the points on both borders can be deduced. We interpolate points equably along the line segment with them as ends. The interpolated points as well as the pairs of points constitutes several included angles by connecting neighbor points. Regard these included angles as variables, by solving an optimization problem to make the variance of these variables smallest, a smoothed reconstruction with smoothly connected topological parts can be obtained.

References

1. Xu, G.; Chen, J.Y.; Li, X.T. 3-D Reconstruction of Binocular Vision Using Distance Objective Generated From Two Pairs of Skew Projection Lines. *IEEE Access* **2017**, *5*, 27272–27280. [CrossRef]
2. Chu, P.M.; Cho, S.; Fong, S.; Park, Y.W.; Cho, K. 3D Reconstruction Framework for Multiple Remote Robots on Cloud System. *Symmetry* **2017**, *9*, 55. [CrossRef]
3. Xu, G.; Yuan, J.; Li, X.T.; Su, J. 3D reconstruction of laser projective point with projection invariant generated from five points on 2D target. *Sci. Rep.* **2017**, *7*, 7049. [CrossRef] [PubMed]
4. Xu, G.; Zhang, X.Y.; Li, X.T.; Su, J.; Hao, Z.B. Global Calibration Method of a Camera Using the Constraint of Line Features and 3D World Points. *Meas. Sci. Rev.* **2016**, *16*, 190–196. [CrossRef]
5. Ranftl, R.; Vineet, V.; Chen, Q.; Koltun, V. Dense Monocular Depth Estimation in Complex Dynamic Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
6. Smith, W.A.P.; Ramamoorthi, R.; Tozza, S. Linear Depth Estimation from an Uncalibrated, Monocular Polarisation Image. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
7. Karsch, K.; Liu, C.; Kang, S.B. Depth Transfer: Depth Extraction from Video Using Non-Parametric Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2144–2158. [CrossRef] [PubMed]
8. Saxena, A. Depth estimation using monocular and stereo cues. In Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007.
9. Depth Estimation from Single Image Using CNN-Residual Network. Available online: <http://cs231n.stanford.edu/reports/2017/pdfs/203.pdf> (accessed on 30 August 2017).
10. Haim, H.; Elmaleh, S.; Giryes, R.; Bronstein, A.M.; Marom, E. Depth Estimation From a Single Image Using Deep Learned Phase Coded Mask. *IEEE Trans. Comput. Imaging* **2018**, *4*, 298–310. [CrossRef]
11. Gan, Y.; Xu, X.; Sun, W.; Lin, L. Monocular Depth Estimation with Affinity, Vertical Pooling, and Label Enhancement. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
12. Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Tao, D. Deep Ordinal Regression Network for Monocular Depth Estimation. *arXiv* **2018**, arXiv:1806.02446.
13. Jiao, J.; Cao, Y.; Song, Y.; Lau, R.W.H. Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss. In Proceedings of the 15th European Conference, Munich, Germany, 8–14 September 2018.
14. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
15. Garg, R.; Bg, V.K.; Carneiro, G.; Reid, L. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. *arXiv* **2016**, arXiv:1603.04992.
16. Wang, G.H.; Chu, Y.B. A New Oren-Nayar Shape-from-Shading Approach for 3D Reconstruction Using High-Order Godunov-Based Scheme. *Algorithms* **2018**, *11*, 75. [CrossRef]

17. Zhu, W.; Chang, X.; Wang, Y.B.; Zhai, H.Y.; Yao, Z.X. Reconstruction of Hydraulic Fractures Using Passive Ultrasonic Travel-Time Tomography. *Energies* **2018**, *11*, 1321. [[CrossRef](#)]
18. Xu, G.; Yuan, J.; Li, X.T.; Su, J. Optimization reconstruction method of object profile using flexible laser plane and bi-planar references. *Sci. Rep.* **2018**, *8*, 1526. [[CrossRef](#)] [[PubMed](#)]
19. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
20. Stockman George, C. *Computer Vision*; Prentice Hall: Upper Saddle River, NJ, USA, 2001.
21. Jalba, A.; Sobiecki, A.; Telea, A. An Unified Multiscale Framework for Planar, Surface, and Curve Skeletonization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 30–45. [[CrossRef](#)] [[PubMed](#)]
22. Rodriguez, J.; Ayala, D. Erosion and Dilation on 2D and 3D Digital Images: A new size-independent approach. In Proceedings of the Vision Modeling & Visualization Conference, Stuttgart, Germany, 21–23 November 2001.
23. Munkres, J. *Introduction to Topology*; Saunders College Pub: Philadelphia, PA, USA, 1983.
24. Kreyszig, E. *Differential Geometry*; University of Toronto Press: Toronto, ON, Canada, 1959.
25. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. *arXiv* **2016**, arXiv:1606.00373.
26. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. Available online: <https://papers.nips.cc/paper/5539-depth-map-prediction-from-a-single-image-using-a-multi-scale-deep-network.pdf> (accessed on 10 September 2014).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).