Multimed Tools Appl (2018) 77:9753–9777
https://doi.org/10.1007/s11042-018-5822-y

A next best view method based on self-occlusion information in depth images for moving object

Shihui Zhang[1,2] · **Xin Li**[1] · **Huan He**[1] · **Yuxia Miao**[1]

Received: 12 January 2017 / Revised: 5 December 2017 / Accepted: 21 February 2018 /
Published online: 8 March 2018

Abstract The determination of next best view of a camera for moving object has wide application in dynamic object scenario, such as unmanned aerial vehicle and automatic recognition. The major challenge of this problem is how to determine the next best view while the visual object is moving. In this work, a novel next best view method based on self-occlusion information in depth images of moving object is proposed. Firstly, a depth image of moving object is acquired and self-occlusion detection is utilized in the acquired image. On this basis, the self-occlusion regions are modeled by utilizing space quadrilateral subdivision. Secondly, according to the modeling result, a method based on the idea of mean shift is proposed to calculate the result of self-occlusion avoidance corresponding to the current object. Thirdly, the second depth image of moving object is acquired, and the feature points in two images are detected and matched, then the 3D motion estimation is accomplished by the 3D coordinates of feature points which are matched. Finally, the next best view is determined by combining the result of self-occlusion avoidance and 3D motion estimation. Experimental results validate that the proposed method is feasible and has relatively high real-time performance.

✉ Shihui Zhang
sshhzz@ysu.edu.cn

Xin Li
woai508305@163.com

Huan He
hih8928@stumail.ysu.edu.cn

Yuxia Miao
yuxia_miao@163.com

[1] School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

[2] The Key Laboratory for Computer Virtual Technology and System Integration of HeBei Province, Qinhuangdao 066004, China

Keywords Moving object · Depth image · Self-occlusion avoidance · 3D motion estimation · Next best view

1 Introduction

The determination of next best view is to find a new observation view based on the information of visual object in current view to achieve the goal that the maximal information of unknown regions of visual object can be obtained by the camera.

Nowadays, scholars have gained some achievements on the next best view. Connolly [7], as one of the earlier scholars studying the next best view, used partial octree model to describe visual object, and made different marks to the nodes to determine the next best view. Roy et al. [32] used search tree nodes to determine the next best view. Low et al. [21] proposed a next best view method by using an adaptive hierarchical algorithm. Blaer et al. [4] used a voxel-based occupancy method to plan the next best view. By combining GKLT feature tracking, Trummer et al. [34] explicitly used the knowledge about the current 3D estimation of the tracked point to determine the next best view. Based on the former work, they also proposed a next best view method by combining online method in literature [35]. Based on the model's covariance structure and appearance, Dunn et al. [12] determined the next best view by deploying a hierarchical uncertainty driven model refinement process. Jia et al. [17] determined the next best view by using information on the image sequences and their relative 3D positions. Haner et al. [15] proposed a method by using covariance propagation to determine the next best view. Based on the field-of-view constraint of stereo vision, Freundlich et al. [13] iteratively minimized the fused uncertainty to determine the next best view. Li et al. [20] extracted different views' features by unsupervised feature learning, and then trained classifiers to evaluate each view's discrimination ability to determine the next best view. Adler et al. [1] sorted the candidate views by achievable information gain to determine the next best view. Mauro et al. [24] proposed a next best view method based on the concept of view importance. Yiakoumettis et al. [40] introduced a relevance feedback on-line learning strategy to learn the user's preference to determine the next best view.

However, because of never considering occlusion factor in these methods, the more serious the occlusion phenomenon is, the lower the accuracy of these methods would be. Therefore, scholars proposed the next best view methods taking occlusion into account. Based on the Positional Space algorithm, Pito [30] determined the next best view from plenty of candidate views. Banta et al. [3] proposed a method based on the overall observation strategy to determine next best view. Li et al. [19] proposed a viewpoint planning method by calculating information entropy, and regarded the view corresponding to maximal information entropy as next best view. Vázquez et al. [36] proposed an automatic view selection using viewpoint entropy. Combining layered ray tracing and octree, Vasquez-Gomez et al. [37] constructed the object model and generated candidate views based on sorting of the utility function to determine the next best view. Wenhardt et al. [16] used a Kalman filter to obtain the best estimate of the object's geometry, and determined the next best view by choosing a suitable optimization criterion. Potthast et al. [31] utilized a belief model of the unobserved space to estimate the expected information gain of each possible viewpoint to determine the next best view. Kriegel et al. [18] proposed a surface-based next best view approach by creating a triangle surface and determining viewpoints similar to human intuition. Maver et al. [25] approximated the occluded region by polygons and used the occluded region information to

determine the next best view. Wu et al. [38] determined the next best view by utilizing layered contour fitting(LCF) with a density-based clustering algorithm. Giorgi et al. [14] determined the next best view according to semantic criteria. Munkelt et al. [26] proposed a next best view method based on voxel space. Based on a retrainable neural network architecture, Papaoulakis et al. [29] proposed a next best view method for detecting athletes in large-scale Olympic events. Delannay et al. [9] selected the next best view based on the contextual features. Chen et al. [5] extracted foreground likelihood and projected it to define a ground occupancy map to determine the next best view. Daniyal et al. [8] used a multivariate Gaussian distribution to determine the next best view. Chen et al. [6] used ray tracing to determine how much new information a given sensor perspective would reveal, and the next best view was determined by new information. Although these methods consider the factor of occlusion, there are limitations in time complexity [3, 19, 30, 36], specific equipment [16, 18, 31, 37], priori knowledge [14, 25, 26, 38], multi cameras [5, 6, 8, 9, 29] etc., and what's more, all the research objects in literature [1, 3–9, 12–21, 24–26, 29–32, 34–38, 40] are stationary. Furthermore, in many scientific research fields such as 3D reconstruction of moving object, automatic tracking, recognition of moving object, operation of robot in hazardous regions, spacecraft docking etc., the visual objects are moving and have self-occlusion, and these tasks have high demand for real-time. Due to the limitation in literature [1, 3–9, 12–21, 24–26, 29–32, 34–38, 40], they can't solve these issues.

Aiming to the moving visual object, depth images of object need to be matched for motion estimation. The ORB(Oriented Fast and Rotated BRIEF) algorithm proposed by Rublee et al. [33] has fast speed and high efficiency, which is widely applied in image-based matching. Makantasis et al. [22] utilized ORB to deal with image filtering from removing outlines as to perform a 3D image retrieval from the wild. Based on ORB algorithm, Mur-Artal et al. [27] proposed a feature-based monocular SLAM system operated in real-time, in small and large, indoor and outdoor environments. Mason et al. [23] developed an approach to object perception based on the principle of object discovery by using ORB. In this paper, based on ORB algorithm, a method to pre-match two images is proposed to estimate the motion of visual object.

The Kinect sensor shows promise in many computer vision applications, such as data acquisition and 3D modeling. Alexiadis et al. [2] described a novel system that automatically evaluated dance performances and provided the visual feedback to the performer in a 3D virtual environment, and the motion of a performer was acquired and modeled via Kinect-based human skeleton tracking. Dimitropoulos et al. [10] used the Kinect sensor to track the volume of a performer and produce skeletal data, so that the intangible treasures can be learned in an interactive 3D environment. Doulamis et al. [11] utilized the Kinect sensor to build Digital Heritage Libraries to protect the tangible and intangible cultural heritage. Yang et al. [39] proposed a real-time synthetic aperture imaging algorithm based on Kinect sensor. In the process of real experiments in this paper, the Kinect sensor is used to acquire depth images of moving objects.

In order to determine the next best view when the visual object is moving, this paper, by using the self-occlusion information in depth image, proposes a method through combining the self-occlusion avoidance and 3D motion estimation to determine the next best view. And the proposed method is different from the traditional next best view methods for reconstruction or recognition. The main purpose of proposed method is to observe the occluded region, which contains much useful information of the visual object. If the information of occluded region can be obtained, both the reconstruction and recognition results are greatly improved, so that

the visual system combining the proposed method can perform these tasks better, such as 3D reconstruction of moving object, automatic tracking, recognition of moving object, operation of robot in hazardous regions, spacecraft docking etc.. Experimental results in our work demonstrate that the proposed method is feasible and has relatively high real-time performance.

2 Problem formulation and method overview

The determination of next best view based on the self-occlusion information in depth image of moving object can be defined as that taking self-occlusion regions as the unknown regions and taking two depth images of moving object as the research object, the result of self-occlusion avoidance is calculated by using the self-occlusion information in the first depth image, then the result of 3D motion estimation is calculated by using the two depth images of moving object. Finally the next best view is determined by combining the result of self-occlusion avoidance and 3D motion estimation to achieve the goal that the maximal information of self-occlusion regions of the moving object can be obtained by the camera.

The definition of self-occlusion avoidance is that when the visual object is stationary, the next best view is calculated based on the self-occlusion information in depth image to achieve the goal that the maximal information of self-occlusion regions can be obtained by the camera. But in this paper, the problem of next best view specially refers to the fact that when the visual object is moving, the next best view is calculated by combining the result of self-occlusion avoidance and 3D motion estimation to achieve the goal that the maximal information of self-occlusion regions of the moving object can be obtained by the camera.

Fig. 1 shows the position relation between the depth camera and the moving object. Fig. 1a is the position relation in the initial view. The region $ABEA'ACDA'$ is the self-occlusion region. Fig. 1b is the position relation in the next best view which is only calculated by the method of self-occlusion avoidance. Fig. 1c is the position relation in the next best view which is calculated by our proposed method for moving object.

It can be seen from Fig. 1 that the camera can't reach the next best view if only use the method of self-occlusion avoidance, because the visual object is moving. But in our work, the motion of visual object is estimated, which compensates the visual object's motion, so the camera can achieve the next best view. It illustrates that the proposed method can solve the problem of next best view of a camera for moving object.

Based on the analysis above, the overall idea of proposed method is as follows. Firstly, the first depth image of moving object is acquired and the self-occlusion detection is utilized in the acquired image. On this basis, the self-occlusion regions are modeled by utilizing space quadrilateral subdivision and the area, center and normal vector of each space quadrilateral are calculated. Secondly, the result of self-occlusion avoidance corresponding to current object is calculated based on the idea of mean shift by using the space quadrilateral information. Thirdly, the second depth image of moving object is acquired and the mean curvature of each pixel in the two acquired images is calculated as the local invariant feature. The features of the two acquired images are pre-matched first, and a method to remove the wrong matching points by using the constraint of rigid invariance is utilized to get the accurate matching results. Then according to the accurate matching results, the 3D motion can be estimated by using the 3D coordinates of these accurate matched points. Finally, the next best view is determined by combining the results of self-occlusion avoidance and 3D motion estimation.

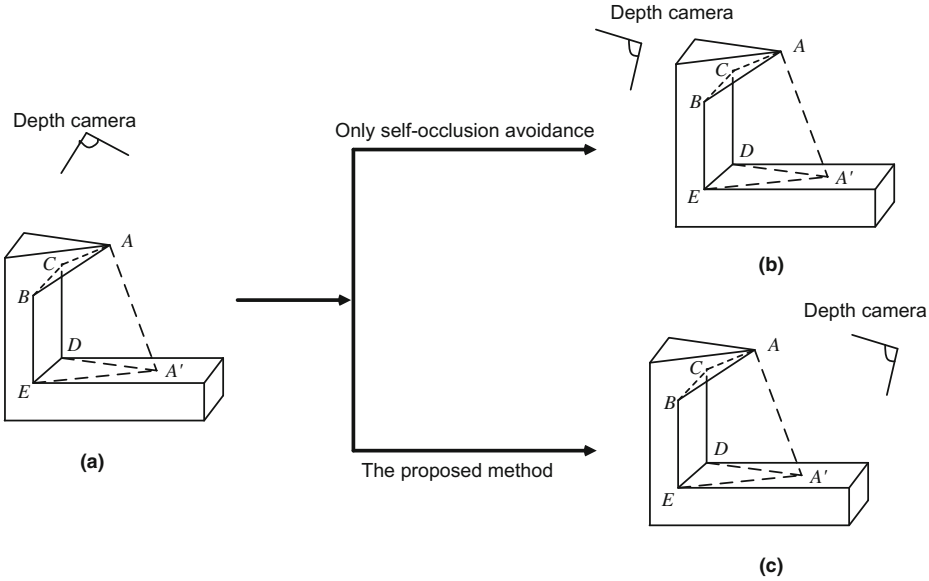

Fig. 1 The position relation between the depth camera and the moving object **(a)** The position relation in the initial view **(b)** The position relation in the next best view which is only calculated by the method of self-occlusion avoidance **(c)** The position relation in the next best view which is calculated by our proposed method for moving object

3 Self-occlusion avoidance

3.1 Modeling the self-occlusion regions

3.1.1 Obtaining the self-occlusion information of visual object

In order to model the self-occlusion regions, first of all, the self-occlusion information is obtained from the depth image of visual object. Self-occlusion information refers to the self-occlusion boundaries and its corresponding nether adjacent boundaries obtained from the depth image in the current view, and each obtained self-occlusion boundary and its corresponding adjacent boundary compose a self-occlusion region in 3D space. The self-occlusion boundaries and its corresponding nether adjacent boundaries are obtained by utilizing the method in literature [41], and then all the points on the self-occlusion boundaries compose the self-occlusion boundary set O and all the points on the nether adjacent boundaries compose the nether adjacent boundary set $O^{'}$. Fig. 2 shows the depth image of Bunny and its self-occlusion boundaries and nether adjacent boundaries in current view. The red points are the self-occlusion boundary points and the green points are the nether adjacent boundary points in Fig. 2b.

3.1.2 Modeling the self-occlusion regions based on the self-occlusion information

Based on the obtained self-occlusion information, the self-occlusion regions are modeled to provide the basis for self-occlusion avoidance. Because the internal information of self-occlusion regions is unknown, one self-occlusion region is subdivided to describe itself by

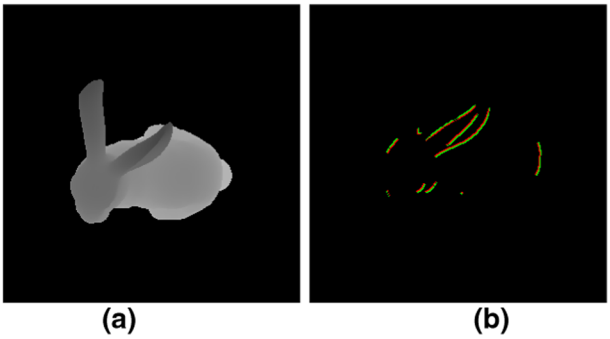

Fig. 2 The depth image of Bunny and its self-occlusion boundaries and nether adjacent boundaries **(a)** The depth image of Bunny **(b)** The self-occlusion boundaries and nether adjacent boundaries in the depth image of Bunny

the following method. Two adjacent self-occlusion points o_i, o_{i+1} on the same self-occlusion boundary are taken out from the self-occlusion boundary set O, meanwhile their corresponding adjacent points $o_i^{'}$, $o_{i+1}^{'}$ are taken out from the nether adjacent boundary set $O^{'}$. Then a space quadrilateral $o_i o_{i+1} o_{i+1}^{'} o_i^{'}$ is formed by the four points in 3D space and denoted by $patch_i$, where i is an integer from 1 to $N-1$, N is the number of points on the self-occlusion boundary. At last, all self-occlusion regions are modeled by the above space quadrilateral subdivision method. Fig. 3 shows the sketch map of self-occlusion region subdivision.

3.1.3 Calculating the area, center and normal vector of each patch

After modeling self-occlusion regions, the area, center and normal vector of each patch are calculated to solve the problem of next best view.

Firstly, the area of each patch is calculated. In order to describe the $patch_i$ as far as possible, the area S_i of $patch_i$ is defined as half of the sum area of 4 triangles which compose $patch_i$, namely

$$S_i = \frac{1}{2}\left(S_{\Delta o_i o_i^{'} o_{i+1}^{'}} + S_{\Delta o_i o_{i+1}^{'} o_{i+1}} + S_{\Delta o_i o_i^{'} o_{i+1}} + S_{\Delta o_{i+1} o_i^{'} o_{i+1}^{'}}\right) \quad \text{s.t.} \quad 1 \le i \le N-1 \tag{1}$$

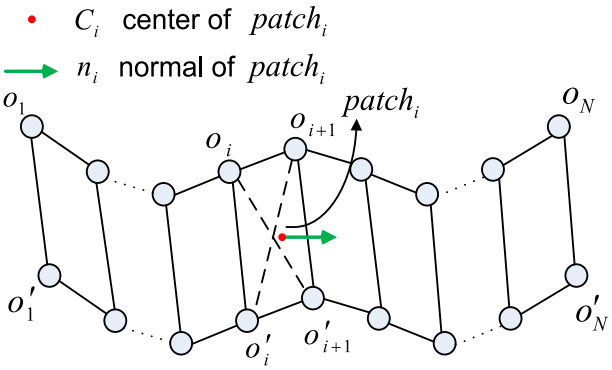

Fig. 3 The sketch map of self-occlusion region subdivision

where $S_{\Delta o_i o_i' o_{i+1}'}$, $S_{\Delta o_i o_{i+1}' o_{i+1}}$, $S_{\Delta o_i o_i' o_{i+1}}$, $S_{\Delta o_{i+1} o_i' o_{i+1}'}$ are the area of triangle $o_i o_i' o_{i+1}'$, triangle $o_i o_{i+1}' o_{i+1}$, triangle $o_i o_i' o_{i+1}$, triangle $o_{i+1} o_i' o_{i+1}'$ respectively.

Then, the center of each patch is calculated. The center C_i of $patch_i$ is defined as the average of coordinates of the four space quadrilateral points which compose $patch_i$, namely

$$C_i = \frac{1}{4}\left(o_i + o_{i+1} + o_i' + o_{i+1}'\right) \tag{2}$$

At last, the normal vector of each patch is calculated. The normal vector of $patch_i$ is defined as the vector which starts from C_i and parallels to the common perpendicular of $o_i o_{i+1}'$ and $o_i' o_{i+1}$. The direction of normal vector is toward outside of the visual object. The concrete method is as follows. Take o_i as the start point and o_{i+1}' as the end point constructs the vector $\boldsymbol{\mu}_i$, meanwhile take o_{i+1} as the start point and o_i' as the end point constructs the vector $\boldsymbol{\gamma}_i$. Then the normal vector $\mathbf{n}_i$ of $patch_i$ is defined as

$$\mathbf{n}_i = \boldsymbol{\mu}_i \times \boldsymbol{\gamma}_i \ \text{ or } \ \mathbf{n}_i = \boldsymbol{\gamma}_i \times \boldsymbol{\mu}_i \tag{3}$$

Through analysis of the self-occlusion boundary and its corresponding nether adjacent boundary, it can be seen that the depth values of self-occlusion points are less than the depth values of its corresponding nether adjacent points, so the direction of normal vector can be determined by the following method to ensure that the direction is toward outside of the visual object. In the depth image, $\mathbf{o_i o_{i+1}}$ is the vector from the point o_i to o_{i+1}, $\mathbf{o_i o_i'}$ is the vector from the point o_i to o_i' and $\mathbf{o_i o_{i+1}'}$ is the vector from the point o_i to o_{i+1}'. Take o_i as the circle center to rotate $\mathbf{o_i o_{i+1}}$ in a clockwise direction, if the rotation angle which is from $\mathbf{o_i o_{i+1}}$ to $\mathbf{o_i o_i'}$ is greater than 0° and less than or equal to 180°, meanwhile the rotation angle which is from $\mathbf{o_i o_{i+1}}$ to $\mathbf{o_i o_{i+1}'}$ is greater than or equal to 0° and less than 180°, the normal vector of $patch_i$ is defined as $\boldsymbol{\mu}_i \times \boldsymbol{\gamma}_i$ to ensure that the direction is toward outside of the visual object, namely

$$\mathbf{n}_i = \boldsymbol{\mu}_i \times \boldsymbol{\gamma}_i \tag{4}$$

if the rotation angle which is from $\mathbf{o_i o_{i+1}}$ to $\mathbf{o_i o_i'}$ is greater than or equal to 180° and less than 360°, meanwhile the rotation angle which is from $\mathbf{o_i o_{i+1}}$ to $\mathbf{o_i o_{i+1}'}$ is greater than 180° and less than or equal to 360°, the normal vector of $patch_i$ is defined as $\boldsymbol{\gamma}_i \times \boldsymbol{\mu}_i$ to ensure that the direction is toward outside of the visual object, namely

$$\mathbf{n}_i = \boldsymbol{\gamma}_i \times \boldsymbol{\mu}_i \tag{5}$$

3.2 The method of self-occlusion avoidance

After modeling the self-occlusion regions, a self-occlusion avoidance method is proposed based on the idea of mean shift by using the information of area and normal vector of each patch. The main process is as follows. Firstly, the best observation position of each patch is determined by using its information of area and normal vector, and all the best observation positions form a set S_p. Secondly, starting from the current camera position P_{begin}, based on the idea of mean shift, the center of mass of all the elements in S_p is calculated by using the information of area and normal vector, then the best observation position P_e of the self-occlusion avoidance result is determined by using the constraint of camera observation

distance to the center of mass. The best observation direction V_e of the self-occlusion avoidance result is the direction from P_e to the midpoint of all visible patch centers when the camera is in P_e. At last, by combining the best observation position and the best observation direction, the result of self-occlusion avoidance is (P_e, V_e). To make our proposed method clear, the concrete process is discussed as follows.

In order to calculate the best observation position of each patch, the normal vector of each patch needs to be handled. Firstly, the length of each normal vector is normalized. The normalized length is equal to the length of vector which is from P_{begin} to the center of visual object. Then the end point p_i of normal vector $\mathbf{n}_i$ which is from C_i is defined as

$$\left(x_{p_i}, y_{p_i}, z_{p_i}\right) = \mathbf{n}_i + \left(x_{C_i}, y_{C_i}, z_{C_i}\right) \tag{6}$$

where $\left(x_{p_i}, y_{p_i}, z_{p_i}\right)$ is the coordinate of p_i, $\left(x_{C_i}, y_{C_i}, z_{C_i}\right)$ is the coordinate of C_i.

Then p_i is defined as the best observation position of $patch_i$, and all the best observation position of patches form the set S_p.

After that, the mean shift vector $F(P_k)$ in P_k is defined as:

$$F(P_k) = \frac{1}{k} \sum_{p_i \in S_p} g_{P_k}(p_i)\omega(p_i)(p_i - P_k) \tag{7}$$

where $\omega(p_i)$ is the weight corresponding to the point p_i, k is the number of elements in S_p, $g_{P_k}(p_i)$ is defined as a sigmoid function to judge whether p_i has effect on iteration or not when the camera in P_k.

The weight $\omega(p_i)$ of point p_i in Eq. (7) is defined as the ratio of the area of $patch_i$ to the total area of all patches, namely

$$\omega(p_i) = \frac{S_i}{\sum_{i=1}^{N-1} S_i} \tag{8}$$

where S_i is the area of $patch_i$.

The equation of $g_{P_k}(p_i)$ is defined as

$$g_{P_k}(p_i) = \frac{1}{1 + e^{-\alpha \cos\theta_i}} \tag{9}$$

where $\theta_i \in [0, 180]$ is the angle between the normal vector of $patch_i$ and the vector from the center of $patch_i$ to P_k. α is a positive constant, the accuracy of result is proportional to its size. Considering the accuracy of result and consumption, we set $\alpha = 400$ in this paper. Analyzing Eq. (9), if θ_i is less than 90°, $\cos\theta_i$ is a positive, so $g_{P_k}(p_i)$ is approximately equal to 1. In this case, p_i has effect on iteration. If θ_i is greater than or equal to 90°, $\cos\theta_i$ is a negative or zero, so $g_{P_k}(p_i)$ is approximately equal to 0. In this case, p_i has no effect on iteration.

Afterwards, based on the mean shift vector and the constraint of camera observation distance, the best observation position of current visual object can be calculated by Eq. (10):

$$P_e = \underset{p_k}{\operatorname{argmin}} \| F(P_k) \| \tag{10}$$

where P_k is the kth iterative position.

The constraint condition minimizing $\|F(P_k)\|$ refers to the fact that the distance from the initial observation position P_0 to the center of visual object is equal to the distance from the best observation position P_e to the center of visual object.

In this paper, the initial iteration position is $P_0 = P_{begin}$, and the allowable error is $\varepsilon = 0.1$. When$\|F(P_k)\| > \varepsilon$, set $P_{k+1} = F(P_k) + P_k$ and continue iterating according to Eq. (10). While when $\|F(P_k)\| < \varepsilon$, the best observation position of self-occlusion avoidance result is $P_e = P_k$.

Then, the best observation direction of self-occlusion avoidance result is calculated. Firstly, the midpoint C_m of all visible patch centers when the camera is in P_e is calculated by

$$C_m = \frac{\sum_{p_i \in S_p}^{i \in [1, N-1]} g_{P_e}(p_i) C_i}{\sum_{p_i \in S_p} g_{P_e}(p_i)} \tag{11}$$

where C_i is the center of $patch_i$.

After calculating C_m, the best observation direction $\mathbf{V}_e$ of self-occlusion avoidance result is defined as the direction from P_e to C_m, namely

$$\mathbf{V}_e = C_m - P_e \tag{12}$$

Finally, the result of self-occlusion avoidance is $(P_e, \mathbf{V}_e)$.

4 3D motion estimation

4.1 Matching two depth images by utilizing ORB algorithm

In order to estimate the 3D motion of visual object, first of all, two acquired depth images should be matched. Because the ORB(Oriented Fast and Rotated BRIEF) algorithm in literature [33] has fast speed and high efficiency, it is utilized to pre-match the two depth images. The concrete process is as follows. Firstly, the mean curvature of each pixel in the two acquired depth images is calculated to be the feature of the pixel. Then the matching points are obtained by utilizing the ORB algorithm in the two depth images acquired before and after visual object motion respectively. Fig. 4 shows the matching results of two depth images acquired before and after the visual object Bunny motion respectively. The blue points in Fig. 4 are feature points, and the two feature points which are connected by the green line are a pair of matching points.

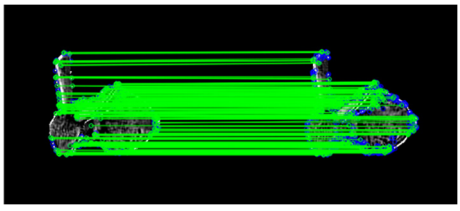

Fig. 4 The matching results of two depth images acquired before and after the visual object Bunny motion respectively

4.2 Filtering matching results to get accurate matching results

Because the error may cause mismatching, a method is proposed to filter matching results by using the constraint of rigid invariance to get the accurate matching results. The idea of proposed method is as follows. Based on the constraint of rigid invariance, the relative position of each matching point in the visual object is invariant in the process of visual object moving. Therefore, the triangle which is constructed by any three accurate matching points in the first image, and the triangle which is constructed by their corresponding points in the second image, should be congruent, and the inaccurate match points generally can not satisfy this condition, so the inaccurate match points can be removed by using this characteristic. Based on this characteristic, a triangular-based inaccurate matching point filter algorithm is presented in this paper. The main steps of the algorithm are as follows.

Firstly, all matching points in the first image are used to form the set M_1, and their corresponding points in the second image are used to form the set M_2. Secondly, triangle t_1 is constructed by the three points in M_1. Meanwhile, triangle t_2 is constructed by the three points corresponding to the points which are constructed triangle t_1, and each edge length of triangle t_1 and t_2 is calculated. Thirdly, through comparing the corresponding edge length of triangle t_1 and t_2, the matching points are filtered by the following rules.

(1) If all the corresponding edges length of triangle t_1 and t_2 are equal to each other, namely $t_1 \cong t_2$, the reason for this situation may be that the relative position of three pairs of matching points are invariant, so the three pairs of points are the accurate matching points. Then, the three points of triangle t_1 are deleted from M_1 and put into the set $M_1^{'}$, and their corresponding points of triangle t_2 are deleted from M_2 and put into the set $M_2^{'}$. Finally, three pairs of matching points from M_1 and M_2 are taken to construct the triangle t_1 and t_2 for further judgment continually.

(2) If two pairs of corresponding edges length of triangle t_1 and t_2 are equal to each other, one pair of corresponding edges length is not equal, the reason for this situation may be that one or two of the points which construct the unequal edge are mismatching points. In order to reduce the time complexity of the algorithm, we would consider that the two points which construct the unequal edge are mismatching points, so the two points which construct the unequal edge in triangle t_1 are deleted from M_1 and the two points which construct the unequal edge in triangle t_2 are deleted from M_2. Then, with the rest pair of matching points, two pairs of matching points from M_1 and M_2 are taken to construct the triangle t_1 and t_2 for further judgment.

(3) If only one pair of corresponding edges length of triangle t_1 and t_2 is equal each other, two pairs of corresponding edges length are not equal, the reason for this situation may be that the two points which construct the equal edge are accurate matching points while the another point which construct the triangle is mismatching point, so the point which is the common point of two unequal edges in triangle t_1 is deleted from M_1 and the point which is the common point of two unequal edges in triangle t_2 is deleted from M_2. Then, with the rest two pairs of matching points, one pair of matching points from M_1 and M_2 is taken to construct the triangle t_1 and t_2 for further judgment.

(4) If all the corresponding edge length of triangle t_1 and t_2 are unequal each other, the reason for this situation may be that the three pairs of points are the mismatching points, so the three points which constructed triangle t_1 are deleted from M_1 and the three points which

constructed triangle t_2 are deleted from M_2. Then, three pairs of matching points from M_1 and M_2 are taken to construct the triangle t_1 and t_2 for further judgment continually.

The process of filtering is repeated until that the number of points in M_1 and M_2 is all less than three. Then the points in $M_1^{'}$ and $M_2^{'}$ are the accurate matching points. Fig. 5 is the accurate matching results after filtering the matching results in Fig. 4.

4.3 The 3D motion estimation

According to the accurate matching results, the 3D motion can be estimated. The relation between the points in the depth images acquired before and after visual object motion respectively is

$$m_{2_i} = \boldsymbol{R} m_{1_i} + \boldsymbol{T} \tag{13}$$

Where m_{1_i} is the point in $M_1^{'}$, m_{2_i} is the corresponding point of m_{1_i} in $M_2^{'}$, $\boldsymbol{R}$ is unit orthogonal rotation matrix, **T** is the translation vector.

As can be seen from Eq. (13), the purpose of 3D motion estimation is to determine the $\boldsymbol{R}$ and **T** which let all of the m_{1_i} and m_{2_i} satisfy Eq. (13). Because the point-to-plane ICP(Iterative Closest Point) algorithm in literature [28] is faster than the traditional ICP algorithm, the $\boldsymbol{R}$ and **T** are calculated by utilizing the method in literature [28].

After solving $\boldsymbol{R}$ and **T**, the result of 3D motion estimation can be expressed as

$$d_2 = \boldsymbol{R} d_1 + \boldsymbol{T} \tag{14}$$

Where d_1 is the 3D point corresponding to the pixel in depth image acquired in the current view, d_2 is the 3D point corresponding to the pixel in depth image acquired in the next view.

5 The determination of next best view

The next best view is determined by combining the result of self-occlusion avoidance and 3D motion estimation. Because the position of moving object is constantly changing, the best view should be changed along with the moving object. In this paper, the self-occlusion avoidance result (P_e, V_e) is the best view when the visual object is not moving. When the visual object is moving, the position relation between the best view and visual object should be constant. Therefore, the self-occlusion avoidance result (P_e, V_e) should be changed based on the Eq. (14). The self-occlusion avoidance result (P_e, V_e) is calculated based on the first depth image, and the motion of visual object is estimated by two adjacent depth images (the first depth image and the second depth image) to obtain the motion information of visual object. Since the

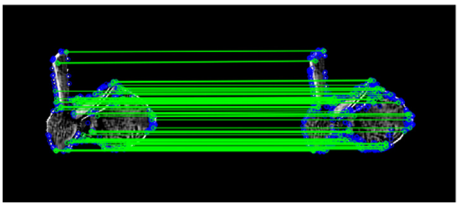

Fig. 5 The accurate matching results after filtering the matching results in Fig. 4

purpose of acquiring second depth image is to obtain the motion information of visual object, the effect of our next best view method is verified by the third depth image. Moreover, the camera is in the observation position of the first depth image initially, so the next best view (P_{nbv}, V_{nbv}) is calculated by using Eq. (14) twice. Namely

$$\begin{cases} P_{nbv} = \boldsymbol{R}^{*}\boldsymbol{R}P_e + (\boldsymbol{R} + \mathbf{I})\boldsymbol{T} \\ V_{nbv} = \boldsymbol{R}^{*}\boldsymbol{R}V_e + (\boldsymbol{R} + \mathbf{I})\boldsymbol{T} \end{cases} \tag{15}$$

Where $\boldsymbol{R}$, $\boldsymbol{T}$ are the unit orthogonal rotation matrix and the translation vector which are calculated by 3D motion estimation, $\mathbf{I}$ is the identity matrix. (P_e, V_e) is the result of self-occlusion avoidance.

6 Experiments and analysis

6.1 Experimental environment

In order to validate the effectiveness of proposed method, the experiments based on 3D object models in Stuttgart Range Image Database are conducted. The experimental hardware environment is the Intel (R) Pentium (R) CPU G2020 @ 2.90GHz, the memory is 4.00GB. The proposed method is implemented by combining C++ and OpenGL. In the process of simulation experiments, the parameter of projection matrix in OpenGL is (60, 1, 200, 600), the window size is 400×400, the initial observation position is $(0, -1, 300)$ and the initial observation direction is $(0, 1, -300)$. In the process of real experiments, depth images are acquired by using Kinect, the horizontal viewing angle is 57°, the distance from the camera to the center of the object is 1200 mm, and the window size is 640×480.

6.2 Experimental results and analysis

To validate the feasibility and real-time performance of proposed method, Section 6.2.1 gives the experimental results and analysis of self-occlusion avoidance. Section 6.2.2 gives the experimental results and analysis of 3D motion estimation. Section 6.2.3 gives the experimental results and analysis of the next best view method for moving object.

6.2.1 Experiments of self-occlusion avoidance

Fig. 6 shows the experimental results based on the self-occlusion avoidance method proposed in this paper. Fig. 6a is the name of visual object. Fig. 6b is the depth image acquired in the initial view. Fig. 6c is the self-occlusion boundaries and nether adjacent boundaries, where the red lines are self-occlusion boundaries and the green lines are nether adjacent boundaries. Fig. 6d is the normal vector of each patch. Fig. 6e is the visible patch observed from the result of self-occlusion avoidance. Fig. 6f is the depth image acquired from the result of self-occlusion avoidance.

As can be seen from Fig. 6, for the visual object Duck, as the self-occlusion phenomenon is not obvious, the visible patch from the result of self-occlusion avoidance is less, namely, the red region in Fig. 6e is smaller. While for the visual object Bunny, Mole, Rocker and Dragon, as the self-occlusion phenomenon is obvious, the visible patch from the result of self-occlusion

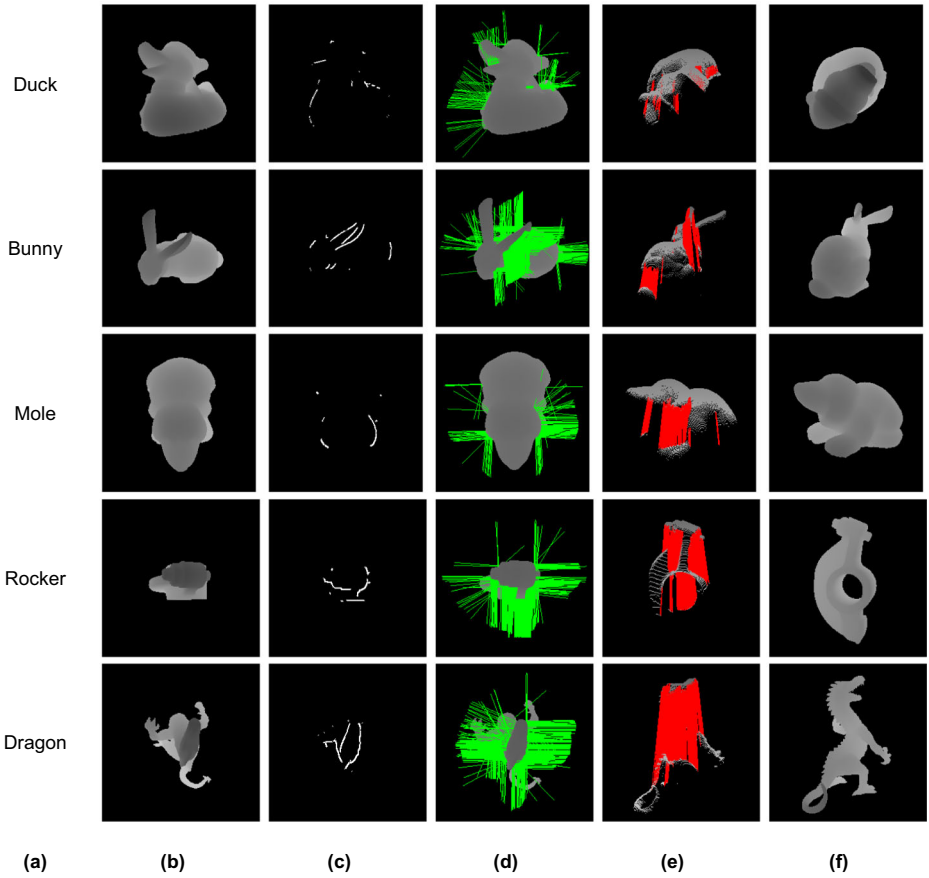

Fig. 6 The experimental results of self-occlusion avoidance **(a)** Visual object **(b)** Depth image acquired in initial view **(c)** Self-occlusion boundaries and nether adjacent boundaries **(d)** Normal vector of each patch **(e)** Visible patch from the result of self-occlusion avoidance **(f)** Depth image acquired from the result of self-occlusion avoidance

avoidance is more, namely, the red region in Fig. 6e is larger. Therefore, the more obvious the self-occlusion phenomenon is, the more effective the proposed method is. Meanwhile comparing the depth images in Fig. 6b and Fig. 6f, it can be seen that the results of self-occlusion avoidance which are calculated by the proposed method align with the observing habit of human vision.

In order to better evaluate the effect of the self-occlusion avoidance method proposed in this paper, the proposed self-occlusion avoidance method is compared with the methods in [15, 17] which are both based on the depth image and consider the occlusion. Fig. 7 shows the experimental results of different methods. Fig. 7a is the name of visual object. Fig. 7b is the depth image acquired in the initial view. Fig. 7c is the depth image acquired from the result calculated by the method in [17]. Fig. 7d is the depth image acquired from the result calculated by the method in [15]. Fig. 7e is the depth image acquired from the result calculated by the proposed self-occlusion avoidance method.

Analyzed from Fig. 7, the results calculated by the method in [17] focus on observing the back of visual object, and the results calculated by the method in [15] focus on observing the

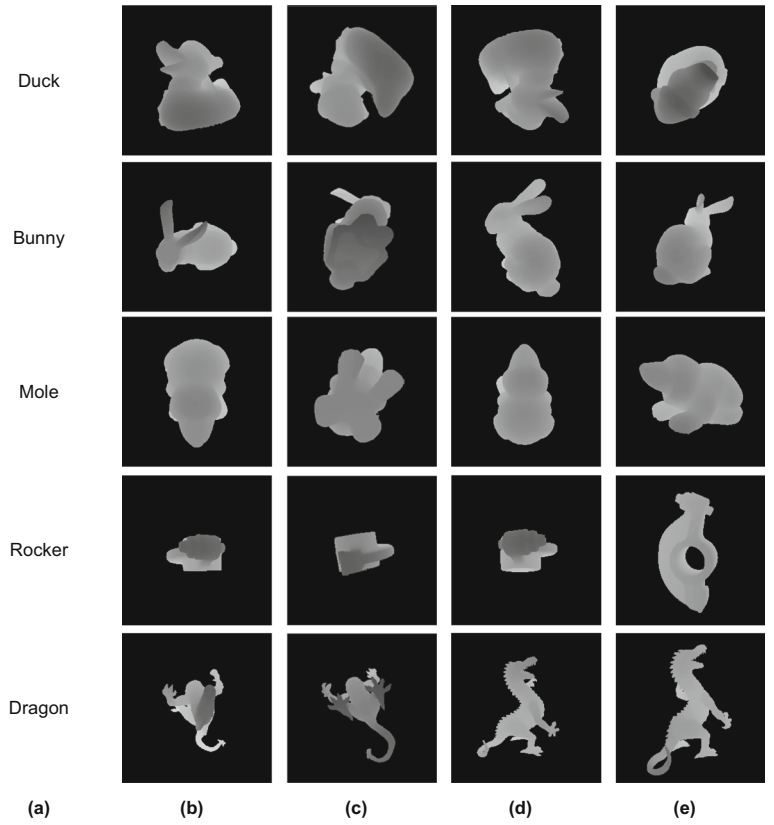

Fig. 7 The experimental results of different methods **(a)** Visual object **(b)** Depth image acquired in initial view **(c)** Depth image acquired from the result calculated by the method in [17] **(d)** Depth image acquired from the result calculated by the method in [15] **(e)** Depth image acquired from the result calculated by the proposed self-occlusion avoidance method

adjoining unknown region of the largest information gain point in initial view. While in this paper, based on the self-occlusion information in depth images acquired in initial view, the results calculated by the proposed self-occlusion avoidance method focus on observing self-occlusion region, which align with the observing habit of human vision.

In order to further examine the effect of proposed method, Table 1 shows the quantitative evaluation of different methods. In Table 1, N_n is the number of surface points, N_o is the number of overlap points, $N_{new} = N_n - N_o$ is the number of new added points, R_o is the overlap rate and R_{new} is the new added rate.

Analyzing Table 1, it shows that compared with the method in [17], for the visual objects where the region of back is larger than the region of self-occlusion, such as Duck, Bunny and Mole, the number of new added points in depth images acquired in the result of proposed method is relatively less. But for the visual objects where the region of back is smaller than the region of self-occlusion, such as Rocker and Dragon, the numbers of new added points in depth images acquired in the result of proposed method are relatively more(although the new rate is slightly lower). The reason is that the method in [17] focuses on considering the back of visual object, when the region of back is smaller than the region of self-occlusion, the method in [17] can't achieve good results. Therefore, the method in [17] has a relatively great

Table 1 The quantitative evaluation of different methods

Visual object	Method in [17]					Method in [15]					Proposed method				
	N_n	N_o	N_{new}	$R_o(\%)$	$R_{new}(\%)$	N_n	N_o	N_{new}	$R_o(\%)$	$R_{new}(\%)$	N_n	N_o	N_{new}	$R_o(\%)$	$R_{new}(\%)$
Duck	21,580	25	21,555	0.12	99.88	21,838	9769	12,069	44.73	55.27	16,111	3844	12,267	23.86	76.14
Bunny	18,839	1	18,838	0.01	99.99	17,601	5053	12,548	28.71	71.29	16,044	1826	14,218	11.38	88.62
Mole	17,612	0	17,612	0	100	14,767	6391	8376	43.28	56.72	15,728	1688	14,040	10.73	89.27
Rocker	4523	0	4523	0	100	4383	1157	3226	26.40	73.60	10,244	329	9915	3.21	96.79
Dragon	8017	12	8005	0.15	99.85	9557	585	8972	6.12	93.88	9203	331	8872	3.60	96.40

limitation. Compared with the method in [15], for the visual objects where the surface is not complex, such as Duck, Bunny, Mole and Rocker, the number of new added points in depth images acquired in the result of proposed method is relatively more. But for the visual objects where the surface is complex, such as Dragon, the number of new added points in depth images acquired in the result of proposed method is slightly less, but the new added rate is higher. The reason is that the method in [15] focuses on considering the adjoining unknown region of the largest information gain point, when the surface is not complex, the method in [15] can't achieve good results. Therefore, the method in [15] has a relatively great limitation. As can be seen from the experimental results of proposed self-occlusion avoidance method, overcoming the limitations of the method in [15, 17], the proposed method has a better applicability to different visual objects.

Because the research object is moving in this paper, the requirement of real-time performance is high. Table 2 shows the comparison of time consumption between the method in [17], the method in [15] and the proposed method.

As can be seen from Table 2, the time consumption of the proposed method is far less than the time consumption of the method in [15, 17]. The average time of obtaining self-occlusion information by the method in [24] is 47.43 ms. Even though considering that time, the average time consumption is 49.57 ms, which is also far less than the time consumption of the method in [15, 17]. Therefore, the proposed self-occlusion avoidance method has relatively high real-time performance.

6.2.2 Experiments of 3D motion estimation

In order to validate the feasibility and real-time performance of proposed 3D motion estimation method, this paper adopts several different methods to estimate the various motions of Bunny, and then the unit orthogonal rotation matrices and the translation vectors calculated by different 3D motion estimation methods are utilized to move the vector $(0, -1, 300)$ which is from the origin of the world coordinate system to the initial observation position. The results and time consumption of different methods are obtained for comparison. Table 3 shows the results and time consumption of different methods. In Table 3, the ideal results are calculated by multiplying the modelview matrix which is extracted from OpenGL directly and the vector $(0, -1, 300)$. Method 1 is only utilizing ICP algorithm in literature [30] to estimate the 3D motion. Method 2 is combining the ORB algorithm and the ICP algorithm to estimate the 3D motion, but in the process of 3D motion estimation, it doesn't filter the matching results. The proposed method not only combines the ORB algorithm and the ICP algorithm, but also filters the matching results in the process of 3D motion estimation. Motion modes include translation along the vector $[1,0,0]^T$ at the speed of 6cm/s, rotation around the vector $[4,1,2]^T$ at the speed of 60°/s, rotation around the vector $[2,5,1]^T$ at the speed of 20°/s and translation along the

Table 2 The comparison of time consumption between different methods

Method	Time consumption/ms					Average time consumption/ms
	Duck	Bunny	Mole	Rocker	Dragon	
Method in [17]	12,826.32	12,930.74	12,920.18	12,775.14	13,113.72	12,913.22
Method in [15]	8288.35	2517.48	7103.26	1086.23	1205.45	4040.15
Proposed method	1.78	3.79	1.25	1.35	2.53	2.14

vector $[1,0,0]^T$ at the speed of 10cm/s, rotation around the vector$[2,1,6]^T$ at the speed of 60°/s and translation along the vector$[1,0,0]^T$ at the speed of 4cm/s.

As can be seen from Table 3, the time consumption of method 1 is 7 to 9 times more than that of method 2 and the proposed method. The reason is that method 1 only adopts ICP algorithm to estimate the 3D motion, all the points in the two acquired depth images are iterated, so method 1 is limited in efficiency. Method 2 matches points by utilizing ORB algorithm first, and then iterates the matching points by utilizing ICP algorithm. It greatly reduces the number of iteration points. So compared with method 1, the time consumption is greatly reduced. But due to the influence of matching error and other factors, the results of method 2 differ greatly from the ideal results. So method 2 is limited in accuracy. On the basis of method 2, the proposed method uses the constraint of rigid invariance to filter the matching points. The proposed method reduces the number of mismatching points and iteration points in method 2, so the proposed method has higher accuracy than method 2, and the time consumption is less than method 2. Overall, through combining the ORB algorithm and the ICP algorithm, the proposed method reduces the time consumption of ICP algorithm, and puts forward the constraint of rigid invariance to improve the accuracy of 3D motion estimation. So the proposed method overcomes the limitations of method 1 and method 2, and it has higher real-time performance and accuracy than method 1 and method 2.

6.3 Experiments of next best view

To validate the feasibility of next best view method proposed in this paper, Fig. 8 shows the depth images acquired in the next best views which are calculated by the proposed method when visual objects are in different motions. The visual object Duck Bunny, Mole, Rocker and Dragon are 3D object models, and the visual object Kettle and Printer are the real objects. Fig.

Table 3 Results and time consumption of different methods

Motion mode	Method	Result	Time consumption/ ms
Translation along the vector $[1,0,0]^T$ at the speed of 6cm/s	Ideal result	(3, −1, 300)	–
	Method 1	(3.002, −1.027, 300.003)	414.14
	Method 2	(3.984, 1.158, 300.114)	60.28
	Proposed method	(3.160, −0.500, 300.009)	49.23
Rotation around the vector $[4,1,2]^T$ at the speed of 60°/s	Ideal result	(3.605, −14.664, 299.621)	–
	Method 1	(3.643, −14.647, 299.606)	475.09
	Method 2	(4.643, −5.690, 300.155)	53.18
	Proposed method	(4.768, −11.273, 300.014)	52.43
Rotation around the vector $[2,5,1]^T$ at the speed of 20°/s and translation along the vector $[1,0,0]^T$ at the speed of 10cm/s	Ideal result	(9.785, −2.888, 299.870)	–
	Method 1	(9.784, −2.909, 299.865)	423.14
	Method 2	(10.347, 1.354, 300.031)	55.66
	Proposed method	(9.625, −1.313, 299.554)	51.23
Rotation around the vector$[2,1,6]^T$ at the speed of 60°/s and translation along the vector$[1,0,0]^T$ at the speed of 4cm/s	Ideal result	(4.619, −5.744, 299.918)	–
	Method 1	(4.630, −5.757, 299.911)	375.51
	Method 2	(5.596, −3.684, 300.119)	53.56
	Proposed method	(4.530, −4.459, 299.946)	46.83

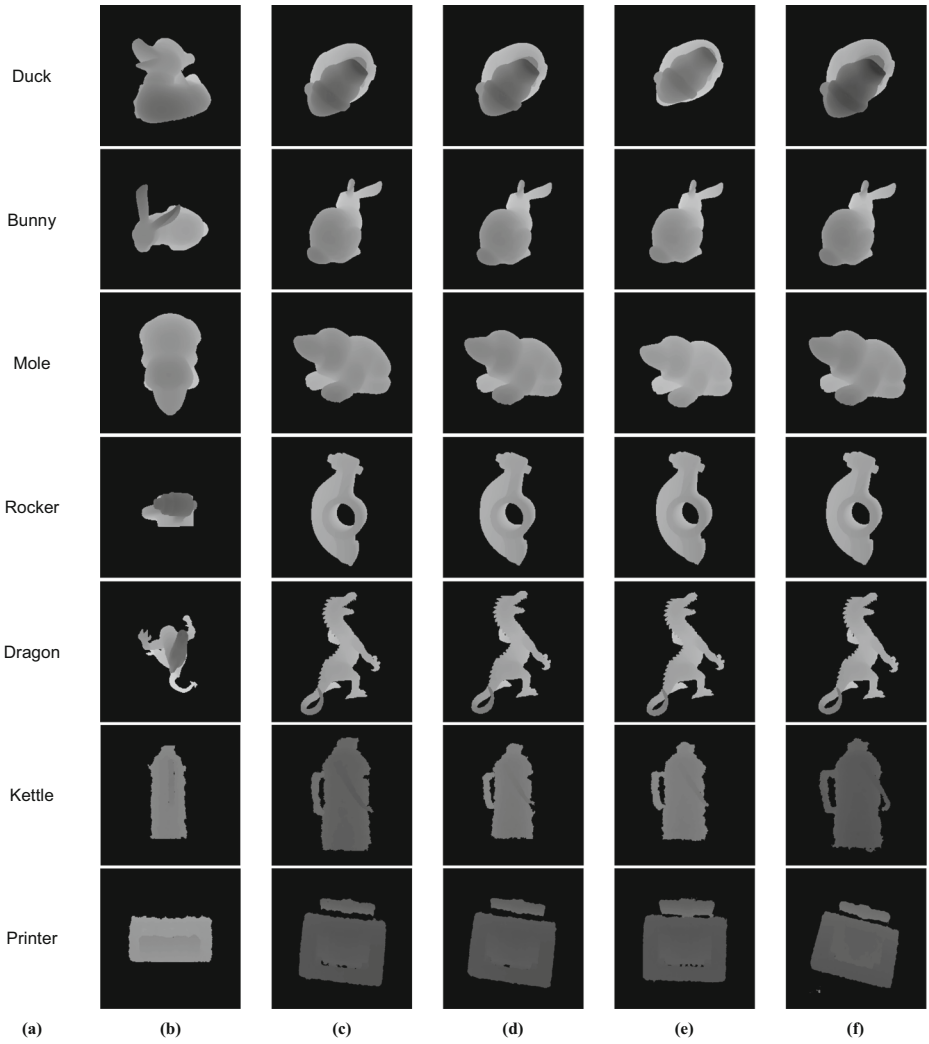

Fig. 8 Depth images acquired in the next best view when visual objects are in different motions **(a)** Visual object **(b)** Depth image acquired in initial view **(c)** Depth image acquired from the result of self-occlusion avoidance **(d)** Depth image acquired in the next best view when visual object does translation along the vector $[1,-1,-1]^T$ at the speed of $2\sqrt{3}$cm/s **(e)** Depth image acquired in the next best view when visual object does rotation around the vector $[2,1,1]^T$ at the speed of 30°/s **(f)** Depth image acquired in the next best view when visual object does rotation around the vector $[-4,1,2]^T$ at the speed of20°/sand translation along the vector $[2,0,1]^T$ at the speed of $2\sqrt{5}$cm/s.

8a is the name of visual object. Fig. 8b is the depth image acquired in the initial view. Fig. 8c is the depth image acquired from the result of self-occlusion avoidance. Fig. 8d is the depth image acquired in the next best view when visual object does translation along the vector $[1,-1,-1]^T$ at the speed of $2\sqrt{3}$cm/s. Fig. 8e is the depth image acquired in the next best view when visual object does rotation around the vector $[2,1,1]^T$ at the speed of 30°/s. Fig. 8f is the depth image acquired in the next best view when visual object does rotation around the vector

Table 4 The quantitative evaluation of the result of self-occlusion avoidance and different next best views

Visual object	Result of self-occlusion avoidance			Motion modes								
				Motion mode 1			Motion mode 2			Motion mode 3		
	N_n	N_{new}	$R_{new}(\%)$	N_n	N_{new}	$R_{new}(\%)$	N_n	N_{new}	$R_{new}(\%)$	N_n	N_{new}	$R_{new}(\%)$
Duck	16,111	12,267	76.14	16,089	12,187	75.75	16,698	12,605	75.48	17,818	13,123	73.65
Bunny	16,044	14,218	88.62	16,051	14,197	88.45	16,021	14,153	88.34	16,273	14,392	88.44
Mole	15,728	14,040	89.27	15,747	14,072	89.36	16,245	14,355	88.37	15,862	14,030	88.45
Rocker	10,244	9915	96.79	10,228	9873	96.53	10,311	9954	96.54	10,242	9881	96.48
Dragon	9203	8872	96.40	9173	8834	96.30	9383	9017	96.09	9119	8777	96.25
Kettle	22,188	17,557	79.13	17,155	12,524	73.01	16,898	12,267	72.59	25,058	19,427	81.52
printer	41,311	37,302	90.29	41,535	37,526	90.35	42,200	38,191	90.50	37,197	33,188	89.22

$[-4,1,2]^T$ at the speed of 20°/s and translation along the vector $[2,0,1]^T$ at the speed of $2\sqrt{5}$cm/s.

As can be seen from Fig. 8, for the 3D object models, the depth images are ideal (low noise and smooth boundary). However, for the real visual objects, there are noise pixels and depth data loss in the depth images acquired by using Kinect. The next best view is determined based on the self-occlusion information of moving object in this paper, so the depth images acquired in the next best views when the visual object is in different motions should be same as the depth image acquired in the result of self-occlusion avoidance. Analyzing the depth images in Fig. 8c, Fig. 8d, Fig. 8e and Fig. 8f, for the ideal 3D models, we can see that the depth images of 3D object models acquired in the next best views when visual object moving are almost same as the depth image acquired in the result of self-occlusion avoidance. For the real visual objects, the difference between the results of real visual objects is slightly larger than the results of ideal 3D object models. Through the analysis of the whole experimental process, the process of 3D motion estimation is the major cause for the difference. Compared with ideal 3D models, in the process of 3D motion estimation, the existence of noise pixels and depth data loss in the depth images acquired by using Kinect bring trouble in pre-matching the two depth images. The noise pixels make the mean curvature feature value of each pixel incorrect, and the depth data loss decreases the number of matching points. These cause the result of 3D motion estimation inaccurate. Therefore, the difference between the results of real visual objects is slightly larger but not obvious. This shows that the proposed method has a good applicability to the visual object in different motions.

In order to validate the effect of the next best view method in this paper, Table 4 shows the quantitative evaluation of the result of self-occlusion avoidance and the next best views when the visual object is in different motions. In Table 4, N_n is the number of surface points, N_{new} is the number of new added points, R_{new} is the new added rate. The motion mode 1 is that the visual object does translation along the vector $[1,-1,-1]^T$ at the speed of $2\sqrt{3}$cm/s, the motion mode 2 is that the visual object does rotation around the vector $[2,1,1]^T$ at the speed of 30°/s, the motion mode 3 is that the visual object does rotation around the vector $[-4,1,2]^T$ at the speed of 20°/s and translation along the vector $[2,0,1]^T$ at the speed of $2\sqrt{5}$cm/s.

It can be seen that when the visual object is in different motions, the number of surface points N_n, the number of new added points N_{new} and the new added rates R_{new} in the depth images acquired in different next best views which are calculated by the proposed method are

Table 5 The time consumption of the proposed method when visual objects are in different motions

Motion modes	Time consumption/ms							Average time consumption of different visual objects/ms
	Duck	Bunny	Mole	Rocker	Dragon	Kettle	Printer	
Motion mode 1	99.78	101.25	100.07	98.48	94.94	98.47	102.64	99.38
Motion mode 2	100.45	97.88	100.18	100.82	96.67	99.54	101.91	99.64
Motion mode 3	98.23	99.82	102.91	98.49	98.59	105.68	99.88	100.51
Motion mode 4	99.29	99.69	110.68	98.65	131.21	95.05	98.84	104.77
Motion mode 5	98.55	100.19	98.99	95.1	101.08	94.46	99.54	98.27
Average time consumption of different motion modes/ms	99.26	99.77	102.57	98.31	104.44	98.64	100.56	100.51

almost same as these in the depth image acquired in the result of self-occlusion avoidance. Even though considering the influence of 3D motion estimation on the result of next best view, the proposed next best view method has a good effect.

It can be analyzed from Table 4 that, when visual object is in different motions, the number of surface points N_n, the number of new added points N_{new} and the new added rates R_{new} in the depth images acquired in the next best views which are calculated by the proposed method are almost same. Moreover, there is no significant difference between the quantitative results of ideal 3D object models and real visual objects by the proposed method, which suggests that the proposed method has a good applicability to different motion modes.

Based on the comprehensive analysis of Fig. 8 and Table 4, it is obvious that for the 3D object models, an ideal next best view can be determined by the proposed method. For the real visual objects, the noise pixels and depth data loss would affect the 3D motion estimation of visual object, which leads to some errors of the experimental results of visual object, but the difference between the experimental results of visual object and the ideal 3D object models is not obvious. It can be seen that the noise pixels and depth data loss have a slight impact on the 3D motion estimation indeed, but a good next best view can be determined by the proposed method, which shows that not only for the ideal 3D object models, but also for the real visual objects including the noise pixels and depth data loss, the proposed method is very robust.

Table 5 shows the time consumption of proposed method when visual objects are in different motions. The motion mode 1 is that the visual object does translation along the vector $[1,-1,-1]^T$ at the speed of $2\sqrt{3}$cm/s, the motion mode 2 is that the visual object does rotation around the vector $[2,1,1]^T$ at the speed of 30°/s, the motion mode 3 is that the visual object does rotation around the vector $[-4,1,2]^T$ at the speed of 20°/s and translation along the vector $[2,0,1]^T$ at the speed of $2\sqrt{5}$cm/s, the motion mode 4 is that the visual object does rotation around the vector $[3,3,6]^T$ at the speed of 40°/s and translation along the vector $[0,2,1]^T$ at the speed of $2\sqrt{5}$cm/s, and the motion mode 5 is that the visual object does rotation around the vector $[1,-2,2]^T$ at the speed of 20°/s and translation along the vector $[1,2,0]^T$ at the speed of $2\sqrt{5}$cm/s.

Table 5 shows that the average time consumption of proposed method is 100.51 ms. Compared with the average time consumption of methods in [15, 17], it can be seen that the average time consumption of proposed method is much less than the average time consumption of methods which don't consider motion in [15, 17]. Therefore, the proposed method has a relatively high real-time performance. Moreover, there is no significant difference between the time consumption of ideal 3D object models and real visual objects by the proposed method. This shows that noise pixels and depth data loss in depth images acquired by using Kinect have few impacts on the time consumption, which illustrates that the proposed method has a relatively high real-time performance and applicability for real visual objects.

7 Conclusions

In this paper, a next best view method based on self-occlusion information in depth images for moving object is proposed. Based on this method, the next best view of a moving object can be effectively determined in real-time. We validate the proposed method by simulation experiments and real experiments.

The major contribution of this paper is that a next best view method for moving object is proposed. The proposed method determines the next best view of a moving object by combining the self-occlusion avoidance and 3D motion estimation, which overcomes the limitation that the traditional next best view methods only apply to the static visual objects. And it provides a means for solving the problem that self-occlusion avoidance methods don't work for moving object.

Another important contribution is that a self-occlusion avoidance method based on the idea of mean shift is proposed. Firstly, based on the self-occlusion information, this method models the self-occlusion regions by utilizing space quadrilateral subdivision. And then based on the idea of mean shift, the result of self-occlusion avoidance is calculated by using the quadrilateral information. This method provides a new means for solving the self-occlusion avoidance and significantly reduces the time consumption of the traditional self-occlusion avoidance methods.

Finally, a 3D motion estimation method through combining the ORB algorithm and the ICP algorithm is proposed. The proposed 3D motion estimation method significantly reduces the time consumption. And in the process of 3D motion estimation, a method to filter the matching results based on the constraint of rigid invariance is proposed to improve the precision of 3D motion estimation.

The method proposed in this paper describes a new idea of determining next best view. Future work may follow two directions. Because the existing next best view evaluation criteria are all for the static visual objects, we will describe a good evaluation criterion for the next best view of moving objects. Moreover, we also intend to determine the next best view for moving object in a complex environment.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No. 61379065 and the Natural Science Foundation of Hebei province in China under Grant No. F2014203119.

References

1. Adler B, Xiao J, Zhang J (2013) Finding next best views for autonomous UAV mapping through GPU-accelerated particle simulation. Ieee/rsj Int Conf Intel Robot Syst 1056–1061
2. Alexiadis D S, Kelly P, Daras P, et al (2011) Evaluating a dancer's performance using kinect-based skeleton tracking. Proc 19th ACM Int Conf Multimed 659–662
3. Banta JE, Wong LM, Dumont C et al (2000) A next-best-view system for autonomous 3-D object reconstruction. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 30(5):589–598
4. Blaer PS, Allen PK (2007) Data acquisition and view planning for 3-D modeling tasks. Ieee/rsj Int Conf Intel Robot Syst 417–422
5. Chen F, Delannay D, Vleeschouwer CD (2012) Multi-sensored Vision for Autonomous Production of Personalized Video Summaries. User Centric Media. Springer, Berlin Heidelberg, pp 113–122
6. Chen F, Vleeschouwer CD, Cavallaro A (2014) Resource allocation for personalized video summarization. IEEE Transactions on Multimedia 16(2):455–469
7. Connolly CI (1985) The determination of next best views. IEEE Int Conf Robot Auto 432–435
8. Daniyal F, Cavallaro A (2011) Multi-camera Scheduling for Video Production. Vis Med Prod 11–20

9. Delannay D, Danhier N, De Vleeschouwer C (2009) Detection and recognition of sports (wo)men from multiple views. ACM/IEEE Int Conf Distrib Smart Cam 1–7
10. Dimitropoulos K, Manitsaris S, Tsalakanidou F, et al (2015) Capturing the intangible an introduction to the i-Treasures project. Int Conf Comput Vis Theor Appl 773–781
11. Doulamis N, Doulamis A, Ioannidis C, et al (2017) Modelling of Static and Moving Objects: Digitizing Tangible and Intangible Cultural Heritage. Mixed Reality and Gamification for Cultural Heritage. Springer Int Publ 567–589
12. Dunn E, Frahm JM (2009) Next Best View Planning for Active Model Improvement. British Machine Vision Conference, BMVC 2009, London, UK, September 7-10, 2009. Proceedings. DBLP
13. Freundlich C, Mordohai P, Zavlanos MM (2013) A hybrid control approach to the next-best-view problem using stereo vision. IEEE International Conference on Robotics and Automation (ICRA) 2013:4493–4498
14. Giorgi D, Mortara M, Spagnuolo M (2010) 3D shape retrieval based on best view selection. ACM Workshop on 3d Object Retrieval 9–14
15. Haner S, Heyden A (2012) Covariance propagation and next best view planning for 3d reconstruction. Computer Vision–ECCV 2012. Springer Berlin Heidelberg
16. Irving Vasquez-Gomez J, Sucar L E, Murrieta-Cid R (2013) Hierarchical ray tracing for fast volumetric next-best-view planning. IEEE 2013 Int Conf Comput Robot Vis (CRV) 181–187
17. Jia Z, Chang YJ, Chen T (2009) Active view selection for object and pose recognition. IEEE, Int Conf Comput Vis Workshops 641–648
18. Kriegel S, Bodenmüller T, Suppa M, et al (2011) A surface-based Next-Best-View approach for automated 3D model completion of unknown objects. IEEE Int Conf Robot Autom 4869–4874
19. Li YF, Liu ZG (2005) Information entropy-based viewpoint planning for 3-D object reconstruction. IEEE Trans Robot 21(3):324–337
20. Li C, Sun Z, Song M, et al (2013) Best view selection of 3D models based on unsupervised feature learning and discrimination ability. Proc 6th Int Symp Vis Info Comm Int ACM 107–108
21. Low KL, Lastra A (2006) Efficient Constraint Evaluation Algorithms for Hierarchical Next-Best-View Planning. Int Symp 3d Data Proc, Visual Trans 830–837
22. Makantasis K, Doulamis A, Doulamis N et al (2016) In the wild image retrieval and clustering for 3d cultural heritage landmarks reconstruction. Multimedia Tools and Applications 75(7):3593–3629
23. Mason J, Marthi B, Parr R (2012) Object disappearance for object discovery. Ieee/rsj Int Conf Intel Robot Syst 2836–2843
24. Mauro M, Riemenschneider H, Signoroni A, et al (2014), A unified framework for content-aware view selection and planning through view importance, BMVC
25. Maver J, Bajcsy R (2002) Occlusions as a guide for planning the next view. IEEE Trans Pattern Anal Mach Intell 15(5):417–433
26. Munkelt C, Kühmstedt P, Denzler J (2014) Incorporation of a-priori information in planning the next best view. PLoS One 9(3):1–11
27. Mur-Artal R, Montiel JMM, Tardós JD (2015) Orb-slam: a versatile and accurate monocular slam system. IEEE Trans Robot 31(5):1147–1163
28. Nießner M, Dai A, Fisher M (2014) Combining Inertial Navigation and ICP for Real-time 3D Surface Reconstruction. Eurographics (Short Papers) 13–16
29. Papaoulakis N, Doulamis N, Patrikakis C, et al (2008) Real-time video analysis and personalized media streaming environments for large scale athletic events. ACM Workshop Anal Retri Events/Act Workflows Video Streams 105–112
30. Pito R (1999) A solution to the next best view problem for automated surface acquisition. IEEE Trans Patt Anal Mach Intel 1016–1030
31. Potthast C, Sukhatme GS (2014) A probabilistic framework for next best view estimation in a cluttered environment. J Vis Commun Image Represent 25(1):148–164
32. Roy SD, Chaudhury S, Banerjee S (2001) Recognizing Large 3-D Objects through Next View Planning using an Uncalibrated Camera. Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE Int Conf 276–281
33. Rublee E, Rabaud V, Konolige K, et al (2011) ORB: an efficient alternative to SIFT or SURF. 2011 I.E. Int Conf Comput Vis (ICCV) 2564–2571
34. Trummer M, Munkelt C, Denzler J (2009) Combined GKLT Feature Tracking and Reconstruction for Next Best View Planning. Pattern Recognition, Dagm Symposium, Jena, Germany, September 9-11(2009):161–170
35. Trummer M, Munkelt C, Denzler J (2010) Online next-best-view planning for accuracy optimization using an extended e-criterion. IEEE 20th Int Conf Patt Recog (ICPR) 1642–1645
36. Vázquez PP, Feixas M, Sbert M et al (2003) Automatic view selection using viewpoint entropy and its application to image-based modelling. Computer Graphics Forum 22(4):689–700

37. Wenhardt S, Deutsch B, Hornegger J, et al (2006) An Information Theoretic Approach for Next Best View Planning in 3-D Reconstruction. Int Conf Patt Recog 103–106
38. Wu B, Sun X, Wu Q et al (2015) Building Reconstruction From High-Resolution Multiview Aerial Imagery. IEEE Geosci Remote Sens Lett 12(4):855–859
39. Yang T, Ma W, Wang S et al (2016) Kinect based real-time synthetic aperture imaging through occlusion. Multimedia Tools and Applications 75(12):6925–6943
40. Yiakoumettis C, Doulamis N, Miaoulis G et al (2014) Active learning of user's preferences estimation towards a personalized 3d navigation of geo-referenced scenes. GeoInformatica 18(1):27–62
41. Zhang S, Liu J, Kong L (2012) A self-occlusion detection approach based on depth image using SVM. International Journal of Advanced Robotic Systems

Shihui Zhang received his Ph.D. degree from Yanshan University, Qinhuangdao, China, in 2005. Currently he is a professor and a Ph.D. supervisor at the School of Information Science and Engineering, Yanshan University. His research interests include visual information processing, pattern recognition and human-computer interaction.

Xin Li received his B.S. from Hebei Normal University, Shijiazhuang, China, in 2010. He is currently pursuing his M.S. degree at the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. His research interests include computer vision, image processing and pattern recognition.

Huan He received his M.S. degree from Yanshan University, Qinhuangdao, China, in 2015. Currently he is pursuing his Ph.D. degree at the School of Information Science and Engineering, Yanshan University, Qinhuangdao. His research interests include computer vision, machine learning, depth imaging and modeling.

Yuxia Miao received her B.S. from North College Of Beijing University Of Chemical Technology, Hebei, China, in 2010. She is currently pursuing her M.S. degree at the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China. Her research interests include computer vision and image processing.