Research Article

Dynamic Self-Occlusion Avoidance Approach Based on the Depth Image Sequence of Moving Visual Object

Shihui Zhang, 1,2 Huan He, 1 Yucheng Zhang, 1 Xin Li, 1 and Yu Sang 1

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

Correspondence should be addressed to Shihui Zhang; sshhzz@ysu.edu.cn

Received 16 May 2016; Accepted 30 August 2016

Academic Editor: Alessandro Gasparetto

Copyright © 2016 Shihui Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to avoid the self-occlusion of a moving object is a challenging problem. An approach for dynamically avoiding self-occlusion is proposed based on the depth image sequence of moving visual object. Firstly, two adjacent depth images of a moving object are acquired and each pixel’s 3D coordinates in two adjacent depth images are calculated by utilizing antiprojection transformation. On this basis, the best view model is constructed according to the self-occlusion information in the second depth image. Secondly, the Gaussian curvature feature matrix corresponding to each depth image is calculated by using the pixels’ 3D coordinates. Thirdly, based on the characteristic that the Gaussian curvature is the intrinsic invariant of a surface, the object motion estimation is implemented by matching two Gaussian curvature feature matrices and using the coordinates’ changes of the matched 3D points. Finally, combining the best view model and the motion estimation result, the optimization theory is adopted for planning the camera behavior to accomplish dynamic self-occlusion avoidance process. Experimental results demonstrate the proposed approach is feasible and effective.

1. Introduction

Self-occlusion avoidance refers to the fact that when the self-occlusion of visual object occurs, the visual system changes the camera observation direction and position by using current detected self-occlusion cue to observe self-occlusion region further and obtains more information of the visual object so as to accomplish the related visual task better. Therefore, the main problem to be solved on self-occlusion avoidance is how to determine the best observation direction and position of the camera by utilizing the occlusion cue of current visual object.

According to spatiotemporal attribute difference of the visual object, the self-occlusion avoidance problem can be classified into static self-occlusion avoidance and dynamic self-occlusion avoidance. Static self-occlusion avoidance refers to the fact that when the visual object is static, the self-occlusion phenomenon is avoided by moving the camera to a next best view for observing the self-occlusion region detected in an image of the visual object. Dynamic self-occlusion avoidance refers to the fact that when the visual object is moving, the self-occlusion phenomenon is avoided by moving the camera step by step to the best view for observing the self-occlusion region detected in one frame from depth image sequence of the moving visual object. By the concept above, we can see the main difference between the two cases is that the result of static self-occlusion avoidance is a next best view under self-occlusion and the camera can accomplish the task of self-occlusion avoidance in the calculated view, which can be regarded as “one-step” type. While the result of dynamic self-occlusion avoidance is a series of next views, the camera needs multiple motions to avoid self-occlusion, which can be regarded as “step-by-step” type.

Compared with static self-occlusion avoidance, the visual object keeps moving in the process of dynamic self-occlusion avoidance, which makes it impossible to accomplish the dynamic self-occlusion avoidance through moving the camera to the calculated next best view directly. In order to accomplish the dynamic self-occlusion
avoidance, besides calculating the next views, the motion estimation about the moving visual object is necessary. Then the camera not only does synchronous motion with the visual object according to the motion estimation result, but also executes self-occlusion avoidance function simultaneously. In this way, the camera moves step by step until accomplishing the dynamic self-occlusion avoidance.

Before self-occlusion avoidance problem emerging, the similar problem called next best view has been widely studied. Currently, scholars have gained some achievements on the next best view. Connolly [1], 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 of different observation situations, so as to determine the next best view. By discretizing a fixed surface, Pito [2] determined the next best view from plenty of discretization candidate views. Based on the depth data, Whaithe and Ferrie [3] constructed parameter similarity model of object and planned the next best view of the camera according to the difference between the depth data and the current fitted model. Li and Liu [4] proposed a method which used B-spline to construct the model of object and determined the next best view by calculating information gain. Trummer et al. [5] proposed a next best view method by combining on-line theory to optimize the 3D reconstruction precision of any object. Unfortunately, because of never considering occlusion factor in these methods, the more serious the occlusion is, the more the accuracy of these methods would be affected.

Because ubiquitous occlusion would affect the result of the next best view, scholars further proposed the next best view methods taking occlusion into account. Banta et al. [6] proposed a combination method to determine the next best view under occlusion. Based on the integration of active and passive vision, Fang and He [7] determined the next best view through the shape of the shadow region and the concept of limit visible surface. Combining layered ray tracing and octree, Vasquez-Gomez et al. [8] constructed the object model and generated candidate views based on sorting of the utility function to determine the next best view. Potthast and Sukhatme [9] determined the next best view through the difference of information entropies under occlusion. Wu et al. [10] determined the next best view by using the algorithm of layered contour fitting (LCF) based on density. Although these methods consider the factor of occlusion (mutual occlusion or self-occlusion; this paper pays attention to self-occlusion), there are limitations in the camera position [6, 7], specific equipment [8, 9], a priori knowledge [7, 10], and so forth. And besides, the self-occlusion region is not modeled properly in these next best view methods. Moreover, there exist some differences (such as problem description and solving cue) between the next best view and self-occlusion avoidance, so the above methods have some reference significance but cannot be taken as the final solution to the problem of self-occlusion avoidance. Most importantly, all of the above methods aim at the study of static visual object, and they are not suitable for the moving visual object. That is to say, all of the above methods cannot be the solution to dynamic self-occlusion avoidance. However, in many scientific research fields such as real-time 3D reconstruction, object tracking, autonomous navigation, scene recognition of mobile robot, robot autonomous operation, and dynamic scene rendering, the self-occlusion of moving visual object is a universal phenomenon. Meanwhile, the visual system would be invalid even wrong if it cannot effectively detect and avoid self-occlusion of moving visual object; thus the visual system would lose its value, which makes how to avoid the self-occlusion of moving visual object become an inevitable and urgent problem.

According to investigation result, there is no related research on dynamic self-occlusion avoidance of moving visual object; that is to say, the research on dynamic self-occlusion avoidance is still in initial stage at present. Meanwhile, considering the objective fact that the 3D information of a scene can be better obtained from the depth image than the intensity image, an approach for avoiding the self-occlusion of the rigid motion object is proposed in this paper based on the depth image. In the process of designing the method, we mainly solve the following three problems: firstly how to design the solution to the dynamic self-occlusion avoidance problem; secondly how to estimate the motion of visual object based on depth image; thirdly how to measure the effect of dynamic self-occlusion avoidance method. Aiming at the first problem, the dynamic self-occlusion avoidance is posed as an optimization problem, and motion estimation is merged into the optimization process of best view model to solve the problem of dynamic self-occlusion avoidance. Aiming at the second problem, the two Gaussian curvature feature matrices are matched by SIFT algorithm and the efficient singular value decomposition (SVD) is used to estimate the motion of visual object. For the third problem, “effective avoidance rate” is proposed to measure the performance of the dynamic self-occlusion avoidance method. The rest of the paper is organized as follows. Section 2 is the method overview. Section 3 describes the dynamic self-occlusion avoidance method based on depth image. Section 4 presents our experimental results and Section 5 concludes the paper.

2. Method Overview

2.1. The Analysis of Dynamic Self-Occlusion Avoidance. The problem of dynamic self-occlusion avoidance can be described on the premise that the self-occlusion region in an image of visual object image sequence is taken as the research object; the reasonable next view sequence is planned for observing the self-occlusion region by moving the camera to achieve the goal that the most information about the self-occlusion region can be obtained from the camera final view. Because the visual object is moving in the process of self-occlusion avoidance, the next view of the camera should be adjusted dynamically to observe the vested self-occlusion region.
Figure 1 shows the sketch map of camera observing ideal object model. Figure 1(a) is the observing effect of camera in initial view, and the shadow region is the self-occlusion region to be avoided. In the case of object moving, if camera wants to obtain the information of the self-occlusion region, it must do the synchronous motion with the visual object and move to the best view for observing self-occlusion region at the same time. Figure 1(b) is the observing effect when the camera is avoiding self-occlusion. Figure 1(c) is the observing effect when the camera arrives at the final view, in which the camera can obtain the maximum information of self-occlusion region, so the process of dynamic self-occlusion avoidance is accomplished when the camera arrives at this view.

2.2. The Overall Idea of Dynamic Self-Occlusion Avoidance. Based on the analysis of dynamic self-occlusion avoidance above, an approach to dynamic self-occlusion avoidance is proposed. The overall idea of the approach is as follows. Firstly, two adjacent depth images of the moving object are acquired and each pixel’s 3D coordinates (all pixels’ 3D coordinates are in the same world coordinate system) in two depth images are calculated by utilizing antiprojection transformation, and the self-occlusion cue in the second depth image is detected. On this basis, the best view model is constructed according to the self-occlusion information in the second depth image. Secondly, according to the 3D coordinates calculated above, the Gaussian curvature feature matrices corresponding to the two adjacent depth images are calculated by using the pixels’ 3D coordinates. Then, the Gaussian curvature feature matrices corresponding to the two adjacent depth images are matched by SIFT algorithm, and the motion equation is estimated by using the 3D coordinates of the matched points. Finally, combining the best view model and the estimated motion equation of the visual object, a series of next views of the camera are planned to accomplish dynamic self-occlusion avoidance process.

3. The Approach to Dynamic Self-Occlusion Avoidance Based on Depth Image

3.1. Constructing the Self-Occlusion Region to Be Avoided and the Best View Model

3.1.1. Constructing the Self-Occlusion Region to Be Avoided. In order to solve the problem of dynamic self-occlusion avoidance, it is necessary to first construct the self-occlusion region to be avoided. As we know, the information of self-occlusion region in current view is unknown; that is to say, the geometry information of self-occlusion region could not be obtained directly according to the depth image acquired in current view, so the specific modeling method should be adopted to describe the self-occlusion region approximately. Figure 2 shows the local self-occlusion region of visual object and its approximate description, and in Figure 2(a), the red boundary is self-occlusion boundary and the blue boundary is the nether adjacent boundary corresponding to the red one. The method in literature [11] is used to detect the self-occlusion boundary in depth image, and all the points on self-occlusion boundary are organized to form self-occlusion boundary point set \( O \). As shown in Figure 2(b), the region between self-occlusion boundary and nether adjacent boundary in 3D space is the unknown self-occlusion region. Figure 2(c) shows the construction of self-occlusion region model by utilizing quadrilateral subdivision on unknown self-occlusion region.

The concrete steps of constructing self-occlusion region model are as follows. Firstly, take out the \( r \)th self-occlusion boundary point \( o_r \) from self-occlusion boundary point set \( O \) in turn. Mark the unmarked nonocclusion boundary point corresponding to the maximum of depth differences between the self-occlusion boundary point \( o_r \) and its eight neighbors as \( o_r' \), and add \( o_r' \) into nether adjacent boundary point set \( O' \). Secondly, use the neighboring self-occlusion boundary points \( o_r, o_{r+1} \) in \( O \) and their corresponding nether adjacent points \( o_r', o_{r+1}' \) in \( O' \) to form quadrilateral patch \( P_j \), as shown in Figure 2(c); \( j \) is the integer from 1 to \( N \) and the combination of all \( N \) quadrilaterals is approximate description of self-occlusion region. Thirdly, combining depth image and camera parameters, use antiprojection transformation to calculate each pixel’s 3D coordinates. At last, use the 3D coordinates of quadrilateral patch’s four vertices to calculate its normal and area, thus the modeling process of self-occlusion region can be accomplished.

On the premise of not considering the normal direction, the formula for calculating the normal \( \mathbf{v}_{\text{patch}_j} \) of quadrilateral patch \( j \) can be defined as

\[
\mathbf{v}_{\text{patch}_j} = \mathbf{u}_j \times \mathbf{w}_j
\]

or

\[
\mathbf{v}_{\text{patch}_j} = \mathbf{w}_j \times \mathbf{u}_j,
\]

where \( \mathbf{u}_j \) is the vector from point \( o_r \) to the midpoint of \( o_r' \) and \( o_{r+1}' \) in 3D space and \( \mathbf{w}_j \) is the vector from point \( o_r \) to point \( o_{r+1} \) in 3D space. Supposing the initial camera view is
In 3D Euclidean space, \((\mathbf{x}_{\text{begin}}, \mathbf{v}_{\text{begin}})\), in order to ensure that the direction of calculated normal points outside in the process of self-occlusion region modeling, the calculation formula of \(\mathbf{v}_{\text{patch}}\)’s normal is defined as

\[
\mathbf{v}_{\text{patch}} = \text{sign}\left(-\mathbf{v}'_{\text{patch}} \cdot \mathbf{v}_{\text{begin}}\right) \cdot \mathbf{v}'_{\text{patch}},
\]

where sign is the sign function; it means the function value is 1 when \(-\mathbf{v}'_{\text{patch}} \cdot \mathbf{v}_{\text{begin}} \geq 0\) and the function value is \(-1\) when \(-\mathbf{v}'_{\text{patch}} \cdot \mathbf{v}_{\text{begin}} < 0\). The area \(S_j\) of quadrilateral patch \(j\) is defined as

\[
S_j = \|\mathbf{v}_{\text{patch}}\|_2^2.
\]

In summary, the set \(P\) consisting of all quadrilaterals \(\mathbf{v}_{\text{patch}}\) is the modeling result of self-occlusion region, so the self-occlusion information to be avoided is described as

\[
P = \left\{ \left( \mathbf{o}_j, \mathbf{o}_{j+1} \right) \mid \mathbf{o}_{j}, \mathbf{o}_{j+1} \in O' \right\}.
\]

3.1.2. Constructing the Best View Model. The best view model should be constructed after obtaining the self-occlusion information to be avoided. This paper uses the area of visible self-occlusion region on next view to construct the objective function and meanwhile uses the size equality of real object corresponding to pixels in different depth images as constraint condition. So the best view model is defined as

\[
\arg \max_{\mathbf{x}_{\text{BV}}} \sum_{j=1}^{N} S_j \cos \left( \theta_j \left( \mathbf{x}_{\text{BV}} \right) \right)
\]

subject to

\[
\begin{align*}
\theta_j \left( \mathbf{x}_{\text{BV}} \right) &= \begin{cases} 
\theta_j \left( \mathbf{x}_{\text{BV}} \right) & \theta_j \left( \mathbf{x}_{\text{BV}} \right) < \frac{\pi}{2} \\
\pi & \frac{\pi}{2} \leq \theta_j \left( \mathbf{x}_{\text{BV}} \right) \leq \frac{\pi}{2} \\
\frac{\pi}{2} & \theta_j \left( \mathbf{x}_{\text{BV}} \right) \geq \frac{\pi}{2}
\end{cases} \\
\|\mathbf{x}_{\text{mid}} - \mathbf{x}_{\text{BV}}\|_2 &= \|\mathbf{x}_{\text{mid}} - \mathbf{x}_{\text{begin}}\|_2,
\end{align*}
\]

where \(\mathbf{x}_{\text{BV}}\) and \(\|\mathbf{x}_{\text{mid}} - \mathbf{x}_{\text{BV}}\|_2\) are, respectively, the position and direction of camera for observing the vested self-occlusion region at maximum level, \(N\) is the number of quadrilaterals, \(\theta_j(\mathbf{x}_{\text{BV}})\) is the angle between the normal of \(j\)th quadrilateral and the vector from the midpoint of \(j\)th quadrilateral to camera, \(\mathbf{x}_{\text{mid}}\) is the center of gravity of all visible points in initial camera view \((\mathbf{x}_{\text{begin}}, \mathbf{v}_{\text{begin}})\).

3.2. Estimating the Motion Equation of the Visual Object Based on Two Adjacent Depth Images. Different from the static self-occlusion avoidance, the dynamic self-occlusion avoidance is carried out on the premise that the visual object is moving, which makes the motion estimation of the visual object in 3D space become a necessary step in the process of dynamic self-occlusion avoidance. Because the existing motion estimation methods based on depth image all estimate the motion of the visual object in 2D images, this paper designs a feasible scheme based on the depth image to estimate the motion of the visual object in 3D space.

3.2.1. Calculating the Gaussian Curvature Feature Matrices of the Two Adjacent Depth Images. Considering that rigid body has rotation and translation invariant characteristics, the surface shape of visual object remains invariant in the motion process. Because the curvature feature reflects bending degree of the object surface, and Gaussian curvature only depends on Riemannian metric of the surface which is the intrinsic invariant of the surface (i.e., Gaussian curvature keeps invariant in rotation and translation), the motion of visual object is estimated based on Gaussian curvature feature matrices corresponding to the two adjacent depth images.

In 3D Euclidean space, \(k_1\) and \(k_2\) are two principal curvatures of a point on the differentiable surface; then the Gaussian curvature is defined as the product of two principal curvatures; namely \(k = k_1k_2\). The method of calculating the Gaussian curvature in literature [12] is implemented by optimizing convolution calculation, and it is high-efficient and suitable for different scales of curvature value, so the
method in literature [12] is used for calculating the Gaussian curvature feature matrix of depth image.

3.2.2. Estimating the Motion Equation of Visual Object Based on the Matched Feature Points. Analysis shows that the matched points can provide reliable information for the motion estimation if the Gaussian curvature feature matrices corresponding to two adjacent depth images of the visual object can be matched. SIFT algorithm in literature [13] is more efficient, and it is not only invariant to image translation, rotation, and scaling, but also robust to the change of visual angle, affine transformation, and noise, so the SIFT algorithm is adopted in this paper for matching the feature points.

SIFT algorithm mainly consists of two steps: the generation of SIFT features and the matching of feature vectors. In order to generate the SIFT features, firstly the scale space should be established. Because the Difference of Gaussians scale-space (DoG scale-space) has good stability in detection of key points, the DoG scale-space is used for detecting the key points in this paper. Secondly, the key points which are unstable and sensitive to noise are filtered. Then the direction of each feature point is calculated. Finally, the SIFT feature vector of each key point is generated. In the process of generating SIFT feature vector, 16 seed points are selected for each key point, and each seed point has 8 orientations. Therefore, the $16 \times 8 = 128$ dimensions of the SIFT feature vector can be obtained for each key point. After obtaining the SIFT feature vectors of the Gaussian curvature feature matrices corresponding to the two adjacent depth images, the two Gaussian curvature feature matrices can be matched, and the similarity of the vectors is measured by Euclidean distance in the process of matching.

After the key points of the Gaussian curvature feature matrices corresponding to the two adjacent depth images are matched, the motion equation of visual object can be estimated by using the pixels’ 3D coordinates which are obtained by antiprojection transformation. The motion equation is defined as

$$x'_j = Rx_j + T,$$  \hspace{1cm} (6)

where $x_j$ and $x'_j$ are, respectively, the 3D coordinates of points before and after visual object motion, $R$ is the unit orthogonal rotation matrix, and $T$ is the translation vector; then the result of motion estimation is the solution of minimizing the objective function $f(R, T)$. The objective function $f(R, T)$ is defined as

$$f(R, T) = \sum_{j=1}^{M} \| x'_j - (Rx_j + T) \|^2,$$  \hspace{1cm} (7)

where $M$ is the number of key points which are matched. In order to solve the $R$ and $T$ in formula (7), the matrix $H$ is defined as

$$H = \sum_{j=1}^{M} x'_j x_j^T.$$  \hspace{1cm} (8)

where $x'_j = x'_j - (1/M) \sum_{l=1}^{M} x'_l$, $x_j = x_j - (1/M) \sum_{l=1}^{M} x_l$. By carrying out singular value decomposition on $H$, $H$ can be expressed as $H=UAV^T$; then according to $U$ and $V$, $R$ in formula (7) can be deduced as

$$R = VU^T.$$  \hspace{1cm} (9)

After obtaining $R$, substitute $R$ into formula (10) to calculate $T$; namely,

$$T = x'_j - Rx_j.$$  \hspace{1cm} (10)

After obtaining $R$ and $T$, the motion equation shown in formula (6) can be obtained, and the motion estimation of the visual object is accomplished.

3.3. Planning the Next View Based on Motion Estimation and Best View Model. After the motion equation of visual object is estimated, the next view of the camera can be planned. Analysis can be known, if substituting the camera observation position into the motion equation of visual object, the camera can do the synchronous motion with the visual object. In addition, on the basis of the synchronous motion with the visual object, if the camera motion offset for avoiding self-occlusion is introduced, the whole process of dynamic self-occlusion avoidance can be achieved. By analyzing the best view model in formula (5), we can know it is the camera best view that can make formula (11) achieve maximum value. Formula (11) is defined as

$$f(x) = \sum_{j=1}^{N} S_j \cos(\theta_j(x)),$$  \hspace{1cm} (11)

where $x$ is the camera observation position. Since formula (11) is a nonconvex model, it is difficult to directly calculate its global optimal solution. Meanwhile, because the problem of dynamic self-occlusion avoidance needs to calculate a series of next views and the gradient information in formula (11) can provide basis for determining the camera motion offset of the self-occlusion avoidance process, in the process of dynamic self-occlusion avoidance, the gradient descent idea is used in this paper for defining the formula of camera observation position $x'_{NV}$ as

$$x'_{NV} = Rx_V + T + \delta \nabla f(x_V),$$  \hspace{1cm} (12)

where $x_V$ is the current observation position of the camera and $\delta$ is the offset coefficient. Based on the solution to formula (12), the formula of next observation direction $v_{NV}$ is defined as

$$v_{NV} = x_{mid} - x'_{NV}.$$  \hspace{1cm} (13)

According to the constraint condition in formula (5), we can search one point $x_{NV}$ along the opposite direction of $v_{NV}$ and make $x_{NV}$ meet

$$\| x_{mid} - x_{NV} \|_2 = \| x_{mid} - x_{begin} \|_2.$$  \hspace{1cm} (14)

By now, the obtained view $(x_{NV}, v_{NV})$ can be regarded as the new current view $(x_V, v_V)$. After acquiring the depth
image of visual object in the new view \((x_V, v_V)\), the process from formula (6) to formula (14) is repeated until the difference between two \(f(x)\) which are calculated in two adjacent views according to formula (11) is less than a given threshold. It means that a series of views \((x_{NV}, v_{NV})\) can be calculated during the visual object motion, so the process of dynamic self-occlusion avoidance can be accomplished.

### 3.4. The Algorithm of Dynamic Self-Occlusion Avoidance

**Algorithm 1** (dynamic self-occlusion avoidance).

**Input.** The camera internal and external parameters and two adjacent depth images acquired in initial camera view.

**Output.** The set of a series of camera views corresponding to the dynamic self-occlusion avoidance.

**Step 1.** Calculate the pixels’ 3D coordinates and the Gaussian curvature feature matrices corresponding to the two adjacent depth images.

**Step 2.** Detect the self-occlusion boundary in the second depth image and establish the self-occlusion region in the second depth image according to formula (1) to formula (4).

**Step 3.** Construct the best view model according to self-occlusion information (formula (5)).

**Step 4.** Match the key points based on Gaussian curvature feature matrices corresponding to the two adjacent depth images by using SIFT algorithm.

**Step 5.** Based on the 3D coordinates of matched key points, solve the objective function in formula (7) according to formula (8) to formula (10); then the visual object’s motion equation in formula (6) can be obtained.

**Step 6.** Plan the next view of camera according to formula (11) to formula (14).

**Step 7.** In the obtained next view, acquire a depth image and calculate \(f(x)\) in formula (11).

**Step 8.** If the difference between two adjacent \(f(x)\) is less than a given threshold, the process of dynamic self-occlusion avoidance can be terminated, and otherwise, jump to Step 4 to continue the process of dynamic self-occlusion avoidance.

### 4. Experiments and Analysis

#### 4.1. Experimental Environment.

In order to validate the effect of the method in this paper, the OpenGL is adopted to implement the process of dynamic self-occlusion avoidance during camera observing the moving object based on 3D object model in the Stuttgart Range Image Database. The experimental hardware environment is Intel CPU (R) Core (TM) i7-3770 3.40 GHz and the memory is 8.00 GB. The dynamic self-occlusion avoidance program is implemented with C++ language. In the process of self-occlusion avoidance, the parameter of the projection matrix in OpenGL is \((60, 1, 200, 600)\) and the window size is \(400 \times 400\).

#### 4.2. The Experimental Results and Analysis.

To verify the feasibility of the proposed method, we do experiments with different visual objects in different motion modes. During the experiments, the initial camera observation position is \(x_{\text{begin}} = [0.00,−1.00, 300.00]^T\) and the initial observation direction is \(v_{\text{begin}} = [0.00, 1.00, −300.00]^T\). The coordinate unit is millimeter (mm) and the offset coefficient is \(d = 4\). The process of dynamic self-occlusion avoidance will terminate if the calculated difference of \(f(x)\) between two adjacent views is less than the given threshold 30 mm\(^2\). The offset coefficient and threshold are chosen by empirical value. Due to the limited space, here we show 3 groups of experimental results corresponding to the visual object Bunny with relatively large self-occlusion region and Duck with relatively small self-occlusion region. The results are shown in Tables 1, 2, and 3, respectively. Table 1 shows the process of dynamic self-occlusion avoidance in which the visual object Bunny does translation along \([1,0,0]^T\) with the speed of 3 mm/s. Table 2 shows the process of dynamic self-occlusion avoidance in which the visual object Bunny does translation along \([2,1,0]^T\) with the speed of \(\sqrt{5}\) mm/s and rotation around \([−4,1,2.5]^T\) with the speed of \(1/\)s. Table 3 shows the process of dynamic self-occlusion avoidance in which the visual object Duck does translation along \([2,1,0]^T\) with the speed of \(\sqrt{5}\) mm/s and rotation around \([−4,1,2.5]^T\) with the speed of \(1/\)s. Because the camera needs multiple motions in the process of dynamic self-occlusion avoidance, only partial observing results the matched results of Gaussian curvature feature matrices and related camera information are shown in Tables 1, 2, and 3 after the results of first three images are shown.

The first row in Tables 1, 2, and 3 shows the depth image number corresponding to different views in the process of self-occlusion avoidance. At the end of the self-occlusion avoidance, the image number in Tables 1, 2, and 3 is 17, 16, and 13, respectively. Thus we can see that the camera movement times are different for different visual objects or the same visual object with different motions in the process of self-occlusion avoidance. The second row in Tables 1, 2, and 3 shows the acquired depth images of visual object in different camera views which are determined by proposed method. From the images in this row, we can see that the self-occlusion region is gradually observed more and more in the process of self-occlusion avoidance; for example, the self-occlusion region mainly caused by Bunny’s ear in the second image in Tables 1 and 2 and the self-occlusion region mainly caused by Duck’s mouth in the second image in Table 3 are better observed when the process of self-occlusion avoidance is accomplished. The third row in Tables 1, 2, and 3 shows the matched results of the Gaussian curvature feature matrices corresponding to the two adjacent depth images with SIFT algorithm. These matched results are the basis of motion estimation and self-occlusion avoidance, and good matched results are helpful to obtain accurate motion estimation; then the accurate camera motion trajectory can
be further calculated. Therefore, the matched results in the third row will influence the effect of self-occlusion avoidance to a certain extent. The fourth, fifth, and sixth rows in Tables 1, 2, and 3, respectively, show the total matched points, the error matched points, and the error rate in the process of matching with SIFT algorithm, where the error rate is equal to the error matched points divided by the total matched points. The seventh, eighth, and ninth rows in Tables 1, 2, and 3, respectively, show the observation position, observation direction, and the observed area of the vested self-occlusion region in related view when the visual object Bunny and Duck do different motions. It can be seen from these data and their change trends that the process of avoiding self-occlusion dynamically with proposed method accords with the observing habit of human vision.

In view of the fact that there is no method of dynamic self-occlusion avoidance to be compared and no Ground Truth about dynamic self-occlusion avoidance currently, it is very difficult to evaluate our method by comparing with other methods or Ground Truth. In order to make the further qualitative analysis of the feasibility and effectiveness of our method, after deeply analyzing the characteristics of dynamic self-occlusion avoidance, we propose an index named "effective avoidance rate" to measure the performance of dynamic self-occlusion avoidance method. Considering the effect of object surface, the same object will have different self-occlusion regions in different camera views, and meanwhile the distribution of the self-occlusion region is random; it is not realistic to find a camera view in which the camera can obtain all the information of the self-occlusion region, so it is inappropriate to take the ratio of the final observed self-occlusion region area to the total self-occlusion region area as evaluation criterion. In addition, because the goal of dynamic self-occlusion avoidance is that the camera not only can track the visual object but also can observe the vested self-occlusion region maximally, the camera will inevitably gradually be close to the best view in the process of dynamic self-occlusion avoidance, namely, the objective view described in formula (5). The area in the objective view $x_{BV}$ is also the self-occlusion region area really expected to be observed. Based on this, the proposed index of effective avoidance rate is defined as

$$
\eta = \frac{S_{\text{view}}/\sum_{j=1}^{N} S_j}{S_{\text{purpose}}/\sum_{j=1}^{N} S_j} \times 100\% = \frac{S_{\text{view}}}{S_{\text{purpose}}} \times 100\%,
$$

(15)

where $\eta$ is the effective avoidance rate, $S_{\text{view}}$ is the observed self-occlusion region area in the view, where the process of dynamic self-occlusion avoidance is accomplished, $S_{\text{purpose}}$ is the self-occlusion region area in the objective view, and $\sum_{j=1}^{N} S_j$ is the total self-occlusion region area to be avoided.

In order to make quantitative analysis of the performance of proposed method based on the effective avoidance rate, we have done multiple groups of experiments by adjusting the motion modes of different visual objects. Table 4

**Table 4:** The distribution of the self-occlusion region is random; it is possible to avoid self-occlusion regions in different camera views, and meanwhile the observed area of the vested self-occlusion region is random when the visual object Bunny and Duck do different motions.
### Table 2: The self-occlusion avoidance process of visual object Bunny with both translation and rotation.

<table>
<thead>
<tr>
<th>Information</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>…</th>
<th>5</th>
<th>…</th>
<th>9</th>
<th>…</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth image</td>
<td><img src="image1" alt="Depth image" /></td>
<td><img src="image2" alt="Depth image" /></td>
<td><img src="image3" alt="Depth image" /></td>
<td>…</td>
<td><img src="image5" alt="Depth image" /></td>
<td>…</td>
<td><img src="image9" alt="Depth image" /></td>
<td>…</td>
<td><img src="image15" alt="Depth image" /></td>
<td><img src="image16" alt="Depth image" /></td>
</tr>
<tr>
<td>Matching Gaussian curvature feature matrices with SIFT algorithm</td>
<td>—</td>
<td><img src="image1" alt="Matching image" /></td>
<td><img src="image2" alt="Matching image" /></td>
<td>…</td>
<td><img src="image5" alt="Matching image" /></td>
<td>…</td>
<td><img src="image9" alt="Matching image" /></td>
<td>…</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Total matched points</td>
<td>—</td>
<td>224</td>
<td>162</td>
<td>…</td>
<td>205</td>
<td>…</td>
<td>247</td>
<td>…</td>
<td>338</td>
<td>—</td>
</tr>
<tr>
<td>Error matched points</td>
<td>—</td>
<td>20</td>
<td>11</td>
<td>…</td>
<td>30</td>
<td>…</td>
<td>42</td>
<td>…</td>
<td>29</td>
<td>—</td>
</tr>
<tr>
<td>Error rate</td>
<td>—</td>
<td>8.9%</td>
<td>6.8%</td>
<td>…</td>
<td>14.6%</td>
<td>…</td>
<td>17.0%</td>
<td>…</td>
<td>8.6%</td>
<td>—</td>
</tr>
<tr>
<td>Observation position</td>
<td>0.00, −1.00, 300.00</td>
<td>0.00, −1.00, 300.00</td>
<td>56.68, −2.00, 295.07</td>
<td>…</td>
<td>127.16, −45.74, 180.11</td>
<td>…</td>
<td>211.41, −97.78, 180.11</td>
<td>…</td>
<td>253.84, −117.25, 92.19</td>
<td></td>
</tr>
<tr>
<td>Observation direction</td>
<td>0.00, 1.00, −300.00</td>
<td>0.00, 1.00, −300.00</td>
<td>−59.09, −4.85, −294.09</td>
<td>…</td>
<td>−133.20, −226.40, −148.33</td>
<td>…</td>
<td>−255.15, −145.29, −53.43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observed area</td>
<td>—</td>
<td>167.61</td>
<td>625.48</td>
<td>…</td>
<td>1453.64</td>
<td>…</td>
<td>2431.37</td>
<td>…</td>
<td>2648.57</td>
<td>2654.63</td>
</tr>
</tbody>
</table>

### Table 3: The self-occlusion avoidance process of visual object Duck with both translation and rotation.

<table>
<thead>
<tr>
<th>Information</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>…</th>
<th>6</th>
<th>…</th>
<th>9</th>
<th>…</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth image</td>
<td><img src="image1" alt="Depth image" /></td>
<td><img src="image2" alt="Depth image" /></td>
<td><img src="image3" alt="Depth image" /></td>
<td>…</td>
<td><img src="image6" alt="Depth image" /></td>
<td>…</td>
<td><img src="image9" alt="Depth image" /></td>
<td>…</td>
<td><img src="image12" alt="Depth image" /></td>
<td><img src="image13" alt="Depth image" /></td>
</tr>
<tr>
<td>Matching Gaussian curvature feature matrices with SIFT algorithm</td>
<td>—</td>
<td><img src="image1" alt="Matching image" /></td>
<td><img src="image2" alt="Matching image" /></td>
<td>…</td>
<td><img src="image6" alt="Matching image" /></td>
<td>…</td>
<td><img src="image9" alt="Matching image" /></td>
<td>…</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Total matched points</td>
<td>—</td>
<td>552</td>
<td>431</td>
<td>…</td>
<td>456</td>
<td>…</td>
<td>432</td>
<td>…</td>
<td>438</td>
<td>—</td>
</tr>
<tr>
<td>Error matched points</td>
<td>—</td>
<td>20</td>
<td>20</td>
<td>…</td>
<td>38</td>
<td>…</td>
<td>22</td>
<td>…</td>
<td>71</td>
<td>—</td>
</tr>
<tr>
<td>Error rate</td>
<td>—</td>
<td>3.6%</td>
<td>4.6%</td>
<td>…</td>
<td>8.3%</td>
<td>…</td>
<td>5.1%</td>
<td>…</td>
<td>16.2%</td>
<td>—</td>
</tr>
<tr>
<td>Observation position</td>
<td>0.00, −1.00, 300.00</td>
<td>0.00, −1.00, 300.00</td>
<td>−0.18, 26.13, 297.59</td>
<td>…</td>
<td>−9.32, 65.62, 292.07</td>
<td>…</td>
<td>−19.88, 106.03, 279.18</td>
<td>…</td>
<td>−49.74, 135.68, 265.74</td>
<td>105.82, 258.45</td>
</tr>
<tr>
<td>Observation direction</td>
<td>0.00, 1.00, −300.00</td>
<td>0.00, 1.00, −300.00</td>
<td>0.84, −25.99, −298.87</td>
<td>…</td>
<td>14.72, −68.07, −291.81</td>
<td>…</td>
<td>17.58, −92.39, −284.88</td>
<td>…</td>
<td>23.96, 273.91, −114.93</td>
<td></td>
</tr>
<tr>
<td>Observed area</td>
<td>—</td>
<td>83.04</td>
<td>222.32</td>
<td>…</td>
<td>317.65</td>
<td>…</td>
<td>462.84</td>
<td>…</td>
<td>705.77</td>
<td>733.58</td>
</tr>
<tr>
<td>Group number</td>
<td>Visual object</td>
<td>Initial camera view</td>
<td>Motion mode</td>
<td>( \sum_{j=1}^{N} S_j ) (mm²)</td>
<td>( S_{\text{purpose}} ) (mm²)</td>
<td>( S_{\text{view}} ) (mm²)</td>
<td>( \eta ) (%)</td>
<td>( T ) (s)</td>
<td>( \bar{T} ) (s)</td>
<td>( N )</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-------------</td>
<td>-----------</td>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Bunny</td>
<td>( x_{\text{begin}} = [0.00, 1.00, 0.00]^T ) ( v_{\text{begin}} = [0.00, 0.00, 2.00]^T )</td>
<td>Translation along [1.0, 0, 0]^T with the speed of 3 mm/s</td>
<td>7289.26</td>
<td>3189.12</td>
<td>3025.58</td>
<td>94.87</td>
<td>15.8</td>
<td>17</td>
<td>0.93</td>
</tr>
<tr>
<td>2</td>
<td>Bunny</td>
<td>( x_{\text{begin}} = [0.00, 1.00, 0.00]^T ) ( v_{\text{begin}} = [0.00, 0.00, 2.00]^T )</td>
<td>Translation along [2.0, 0, 0]^T with the speed of ( \sqrt{3} ) mm/s and rotation around [1.0, 0, 0]^T with the speed of 1°/s</td>
<td>7289.26</td>
<td>3189.12</td>
<td>2654.63</td>
<td>83.24</td>
<td>15.5</td>
<td>16</td>
<td>0.97</td>
</tr>
<tr>
<td>3</td>
<td>Duck</td>
<td>( x_{\text{begin}} = [0.00, 1.00, 0.00]^T ) ( v_{\text{begin}} = [0.00, 0.00, 2.00]^T )</td>
<td>Translation along [2.0, 0, 0]^T with the speed of ( \sqrt{3} ) mm/s and rotation around [1.0, 0, 0]^T with the speed of 1°/s</td>
<td>1911.10</td>
<td>1025.27</td>
<td>733.58</td>
<td>71.55</td>
<td>13.3</td>
<td>13</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>Bunny</td>
<td>( x_{\text{begin}} = [100.00, -100.00, 256.00]^T ) ( v_{\text{begin}} = [-100.00, 100.00, 256.00]^T )</td>
<td>Rotation around [7, -3, 4]^T with the speed of 2°/s</td>
<td>5028.92</td>
<td>3238.47</td>
<td>3017.48</td>
<td>93.18</td>
<td>13.6</td>
<td>15</td>
<td>0.91</td>
</tr>
<tr>
<td>5</td>
<td>Bunny</td>
<td>( x_{\text{begin}} = [150.00, -100.00, 196.00]^T ) ( v_{\text{begin}} = [-100.00, 150.00, 240.00]^T )</td>
<td>Translation along [1.0, 0, 0]^T with the speed of ( \sqrt{3} ) mm/s and rotation around [1.0, -1.0, -2.0]^T with the speed of 1°/s</td>
<td>4441.50</td>
<td>1758.18</td>
<td>1553.48</td>
<td>88.36</td>
<td>17.1</td>
<td>18</td>
<td>0.95</td>
</tr>
<tr>
<td>6</td>
<td>Duck</td>
<td>( x_{\text{begin}} = [100.00, -150.00, 240.00]^T ) ( v_{\text{begin}} = [-150.00, 100.00, 240.00]^T )</td>
<td>Translation along [3, 2, -1]^T with the speed of ( \sqrt{14} ) mm/s</td>
<td>3307.46</td>
<td>2788.37</td>
<td>2748.35</td>
<td>98.56</td>
<td>19.5</td>
<td>21</td>
<td>0.93</td>
</tr>
<tr>
<td>7</td>
<td>Mole</td>
<td>( x_{\text{begin}} = [-200.00, 50.00, 210.00]^T ) ( v_{\text{begin}} = [210.00, -50.00, -210.00]^T )</td>
<td>Translation along [2, 0, 2]^T with the speed of ( \sqrt{8} ) mm/s and rotation around [1, -1, -2]^T with the speed of 3°/s</td>
<td>320.01</td>
<td>221.68</td>
<td>133.09</td>
<td>60.04</td>
<td>11.2</td>
<td>11</td>
<td>1.02</td>
</tr>
<tr>
<td>8</td>
<td>Rocker</td>
<td>( x_{\text{begin}} = [-196.00, 170.00, 150.00]^T ) ( v_{\text{begin}} = [196.00, -170.00, -150.00]^T )</td>
<td>Translation along [1.0, 1.0, 1.0]^T with the speed of ( \sqrt{6} ) mm/s</td>
<td>1905.09</td>
<td>1440.92</td>
<td>1363.45</td>
<td>94.62</td>
<td>8.4</td>
<td>9</td>
<td>0.93</td>
</tr>
<tr>
<td>9</td>
<td>Rocker</td>
<td>( x_{\text{begin}} = [190.00, -200.00, 180.00]^T ) ( v_{\text{begin}} = [-190.00, 200.00, -180.00]^T )</td>
<td>Translation along [2, -1, -1]^T with the speed of ( \sqrt{6} ) mm/s and rotation around [1, -1, -2]^T with the speed of 3°/s</td>
<td>3433.54</td>
<td>2707.33</td>
<td>1947.98</td>
<td>71.95</td>
<td>9.8</td>
<td>10</td>
<td>0.98</td>
</tr>
<tr>
<td>10</td>
<td>Dragon</td>
<td>( x_{\text{begin}} = [190.00, 200.00, 180.00]^T ) ( v_{\text{begin}} = [-190.00, -200.00, -180.00]^T )</td>
<td>Translation along [2, 0, 0]^T with the speed of ( \sqrt{21} ) mm/s and rotation around [1, 2, -1]^T with the speed of 3°/s</td>
<td>5736.48</td>
<td>1948.61</td>
<td>1485.98</td>
<td>76.26</td>
<td>16.5</td>
<td>15</td>
<td>1.10</td>
</tr>
</tbody>
</table>

Average — — — 4063.56 2150.71 1866.36 86.78 14.1 14.5 0.97
shows the quantitative analysis results of total 10 groups of experiments which include the 3 groups of experiments described above and the other 7 groups. In addition, in the last column of each experiment, the average time, which is from the time that the camera arrives at a certain view to the time that the next view has been calculated, is shown.

From the 10 groups of experiments in Table 4 we can see that, affected by surface of visual object, the different self-occlusion region area $\sum_{j=1}^{N} S_j$ will be generated for the same (or different) visual object in different (or the same) camera views, and the distribution of the self-occlusion region is random, so the vested self-occlusion region area $\sum_{j=1}^{N} S_j$ will not be observed completely even if the camera is in the best view (the objective view). At the same time, the camera can only infinitely be close to but hardly reach the objective view accurately in the process of self-occlusion avoidance; that is to say, at the end of self-occlusion avoidance, the camera observed area $S_{view}$ can only be close to but not reach the area $S_{purpose}$ in the objective view, which also verifies the rationality of taking the effective avoidance rate $\eta$ as evaluation criterion. Besides comparing 10 groups of experiments in Table 4 we can know that the visual objects in groups 1, 4, 6, and 8 only do translation or rotation in different initial views, and the motion is relatively simple; then the average effective avoidance rate of these four groups is 95.29%. But in the groups 2, 3, 5, 7, 9, and 10 in Table 4, the visual objects do both translation and rotation in different initial views, the motion is relatively complex, and then the average effective avoidance rate of these six groups is 78.42%, which is obviously lower than that of the four preceding groups in which the visual objects only do translation or rotation. The main reason from our analysis is that many steps in the process of dynamic self-occlusion avoidance, such as antiprojection transformation, matching points based on SIFT and motion estimation, will make error and the error will accumulate when the camera changes its view. The most important thing is that error accumulation will become more obvious when the motion of the visual object is more complex, which will lead to the result that the effective avoidance rate will be significantly lower when the motion of the visual object is complex. Objectively speaking, it is inevitable that there are error and error accumulation in the process of self-occlusion avoidance, and it is not realistic to eliminate the effect of these errors completely; what different methods can do is to reduce or cut down the effect of these errors. The last row data in Table 4 shows that the average effective avoidance rate of the proposed method in this paper reaches 86.78%; therefore in the average sense, the better result of dynamic self-occlusion avoidance can be obtained based on the proposed method. Furthermore, from the last column data in Table 4 it can be seen that the average time consumption is 0.97s which is from the time that the camera arrives at a certain view to the time that the next view has been calculated. It shows that the proposed method has good real-time performance.

5. Conclusion

In this paper, an approach for avoiding self-occlusion dynamically is proposed based on the depth image sequence of moving visual object. The proposed approach does not need specific equipment and the a priori knowledge of visual object or limit the observation position of the camera on a fixed surface. This work is distinguished by three contributions. Firstly, the problem of dynamic self-occlusion avoidance which is still in initial stage at present is studied, and the solution to dynamic self-occlusion avoidance problem is proposed and implemented, which provides an idea for further studying the dynamic self-occlusion avoidance. Secondly, based on the depth images of moving visual object, the Gaussian curvature feature matrices are matched by SIFT algorithm, and the motion equation is efficiently estimated by using singular value decomposition. This provides a novel idea about motion estimation in 3D space based on the depth image of visual object. Thirdly, considering that there are no evaluation criterions about dynamic self-occlusion avoidance at present, an evaluation criterion named “effective avoidance rate” is proposed to measure the performance of dynamic self-occlusion avoidance reasonably, and it provides the referable evaluation basis for further studying the problem of dynamic self-occlusion avoidance. The experimental results show that the proposed approach achieves the goal that the camera avoids the self-occlusion automatically when the visual object is moving, and the avoidance behavior accords with the observing habit of human vision.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this article.

Acknowledgments

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


