3D Reconstruction of Specific Target Based on Kinectfusion

Q.L. Zhang and N. Fang

ABSTRACT

A new 3D reconstruction method of specific target which is mixed in complex scene is proposed by making full use of the point cloud model acquired by KinectFusion and other point cloud processing algorithm. The target needed to be reconstructed can be obtained by using point cloud segmentation based on RANSAC and Euclidean cluster extraction as well as using surface reconstruction based on concave hull. The purpose of segmentation is to break the point cloud down into distinct clusters which can then be processed independently. The surface reconstruction is to generate a surface of the target that is extracted by segmentation. Experimental results show that the method in this paper has an outstanding performance in 3D reconstruction of the specific target. 

KEYWORD: KinectFusion; point cloud; segmentation; surface reconstruction; 3D reconstruction

INTRODUCTION

3D reconstruction technology based on computer vision is widely used in virtual reality, cultural relic protection, biomedicine, 3D printing, mobile robot navigation, and other field. KinectFusion proposed by Microsoft is a complete real-time 3D reconstruction algorithm, which has a good performance in robustness and precision and, meanwhile, can deal with dynamically changing scenes [1]. Since the algorithm only uses the depth information and the RGB information is not used, the 3D reconstruction of the scene still can be carried out even under the condition of complete darkness.

However, KinectFusion cannot reconstruct a specific target which is mixed in the scene, as the scene with target will be reconstructed together. As shown in Fig. 1,
the aircraft is the target needed to be reconstructed. For this situation, a new 3D reconstruction method of specific target is proposed which can be a very good solution to solve this problem by using point cloud segmentation based on RANSAC and Euclidean cluster as well as surface reconstruction based on concave hull.

![Target within the scene](image1)
![Depth map](image2)
![Reconstruction](image3)

**Figure 1.** Target within the scene.  **Figure 2.** Depth map.  **Figure 3.** Reconstruction.

**KINECTFUSION**

**Kinect**

Kinect is a human-computer interaction device developed by Microsoft based on the interaction of the body, initially applied to the Xbox 360 game machine. Many researchers now use it for human face recognition, gesture recognition, speech recognition, bone tracking, gesture interaction and 3D reconstruction and other scientific research. Kinect can form a highly random diffraction spot in space, which will change with distance. As two speckle patterns are different in space, Kinect can obtain the depth of object by using the pattern of the surface.

**KinectFusion Algorithm**

KinectFusion is a real-time 3D reconstruction algorithm proposed by Microsoft in 2011. The algorithm has high robustness, and some details of the scene can be well reconstructed. The input of KinectFusion is depth map (Fig. 2) acquired by Kinect, and the output is a 3D reconstruction model of scene (Fig. 3).

The main steps of the KinectFusion algorithm are as follows:

1. A key step is to remove the noise in depth map, in the first instance, by using bilateral filter.

\[
D_i(u) = \frac{1}{W_p} \sum N_{f}(\| u - q \|_2) \times \exp^{-2 \sigma^2 \| e^2 \} \quad (1)
\]

Where \( u = (x, y)^T \) is image pixel; \( N_f(t) = \exp^{-2 \sigma^2 \| e^2 \} \) and \( W_p \) is a normalizing constant.

2. And then, the focus is to solve vertex and normal in order to gain the point cloud and related information.

\[
v_i(u) = D_i(u) K^{-1} \cdot [u, 1] \quad (2)
\]

\[
n_i(u) = (v_i(x + 1, y) - v_i(x, y)) \times (v_i(x, y + 1) - v_i(x, y)) \quad (3)
\]

Where \( D_i(u) \) is filtered depth map; \( K \) is calibration matrix; \( v_i(u) \) is vertex; \( n_i(u) \) is normal.
3. The most important step is the camera tracking, which is prepared for the integration of point cloud data that is in different perspectives. The first step of ICP is to find the corresponding points, and KinectFusion uses back projection method to do this.

   (1) Using the projection method to obtain preliminarily corresponding points;

   (2) Eliminating error corresponding points. Vertex and normal need to meet the following two conditions:

   \[
   \| \mathbf{v}^p - \mathbf{v}^p_0 \| < \varepsilon_d \tag{4}
   \]

   \[
   \text{abs}(\mathbf{n}^p \cdot \mathbf{n}^p_0) < \varepsilon_n \tag{5}
   \]

   Where \( \varepsilon_d \) is distance threshold; \( \varepsilon_n \) is normal threshold.

   (3) Solving the camera transformation matrix by using the corresponding points. According to the error mechanism of point to plane, the formula of total error between all the corresponding points is as follows:

   \[
   E(T_{g,\omega}) = \sum_{\omega \in \text{all frames}} \left\| (T_{g,\omega} \mathbf{v}_1(\omega) - \mathbf{v}_2(\omega))^T \mathbf{n}_2(\omega) \right\| \tag{6}
   \]

   Where \( \mathbf{d} = \pi (K_{\omega-1} \mathbf{v}_2(\omega)) \); \( \mathbf{v}_1(\omega) \) is vertex of the current frame; \( \mathbf{v}_2(\omega) \) is vertex of the predicted frame; \( \mathbf{n}_2(\omega) \) is normal of the predicted frame;

   \[
   T_{g,\omega} = \begin{bmatrix} R_{g,\omega} & t_{g,\omega} \\ 0 & 1 \end{bmatrix}
   \]

   represents the pose of camera.

   By optimizing the Equation (6), the camera pose is obtained.

4. Next, KinectFusion fuse the point cloud data by using the TSDF (truncated signed distance function) method on the basis of transformation matrix that represents the motion of the camera.

![Figure 4. Schematic diagram of TSDF.](image)

**PCL**

The Point Cloud Library (PCL) is an open source library which has a large number of generic algorithms and data structures. It contains the point cloud acquisition, filtering, segmentation, registration, retrieval, feature extraction, recognition, tracking, surface reconstruction and visualization algorithms. All these algorithms greatly facilitate the rapid acquisition and processing of 3D information.
KinFu

KinFu is an open source implementation of KinectFusion and is a part of PCL. It made some improvements, which can not only support more depth camera but also can save the reconstructed model as ply, vtk, or pcd.

Point Cloud Segmentation Based on RANSAC

RANSAC (short for random sample consensus) is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers. The algorithm was first published by Fischler and Bolles at SRI International in 1981 [5]. For point cloud processing, this algorithm is often used to extract the line, ball, cone, cylinder and other mathematical model.

In this paper, the point cloud segmentation based on RANSAC needs to be used. Before processing, the algorithm needs to set optimize coefficients, normal distance weight \( w \), model type \( (plane) \), method type \( (RANSAC) \), maximum iterations \( (i_{max}) \) and distance threshold \( (d) \). As shown in Fig. 5.

![Figure 5](image1.png)

Figure 5. Workflow of point cloud segmentation based on RANSAC.

![Figure 6](image2.png)

Figure 6. Workflow of point cloud segmentation based on Euclidean cluster extraction.

Point Cloud Segmentation Based on Clustering Analysis

Clustering segmentation divides the input point cloud into different groups according to some similarity measure. The classical clustering method has maximum-likelihood estimation (MLE), k-means clustering, fuzzy clustering and so on. After the completion of the cluster, the difference in the point cloud within the same group will be small.

Distance is often used as a measure of similarity, such as the Euclidean distance, Mahalanobis distance, Chebychev distance, Manhattan Distance and so on. The Euclidean distance is the "ordinary" distance between two points in Euclidean space [8], which can be used as evaluation index of similarity measure. The closer the distance is, the higher the similarity is. The Euclidean distance between points \( \mathbf{a} = (a_1, a_2, \ldots, a_n) \) and \( \mathbf{b} = (b_1, b_2, \ldots, b_n) \) is:
\[ D(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \]

\[ = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \quad (7) \]

In this paper, the point cloud segmentation based on Euclidean distance needs to be used before surface reconstruction so as to extract the target from scene. Before processing, the algorithm needs to set search method \((k-d\ tree)\), cluster tolerance \((t)\), minimum cluster size \((\text{size}_\text{min})\), and maximum cluster size \((\text{size}_\text{max})\). As shown in Fig. 6.

**Surface Reconstruction based on Concave Hull**

Concave hull and convex hull is a relative concept. The convex hull creates a polygon that can cover all the points, as shown in Fig. 7. However, the concave hull can get a polygon which describes the outline of the scattered points, as shown in Fig. 8. In the concave hull, there is a parameter \(\alpha\) which can control the similarity degree between the polygon and the shape of point cloud. Given a reasonable \(\alpha\), concave hull will create a good mesh fitted with the shape of the input point cloud.

![Figure 7. Convex hull.](image1)

![Figure 8. Concave hull.](image2)

**EXPERIMENTAL RESULTS AND ANALYSIS**

In this section, the validity of the 3D reconstruction method in this paper will be verified by comparative analysis.
Testing environment: WIN8 OS; VS2013; PCL 1.8.0; Kinect v2.

At the beginning of processing, the point cloud segmentation based on RANSAC is the first step so as to separate the plane and the rest. Next, the segmentation based on Euclidean cluster extraction is carried out in order to extract the target (aircraft) from the rest of the point cloud. Surface reconstruction based on concave hull is the last step. As shown in Fig. 9.

Figure 9. Workflow of 3D reconstruction.

Figure 10. Point cloud acquired by Kinfu (Fig. 10)

Input: Point cloud acquired by Kinfu (Fig. 10)

Point cloud segmentation based on Euclidean cluster extraction

Output: the rest without plane (Fig. 12)
Output: plane (Fig. 11)

Point cloud segmentation based on Euclidean cluster extraction

Output: target (Fig. 13)
Output: other clusters

Surface reconstruction based on concave hull

Output: mesh of target (Fig. 14)

Figure 12. Rest of the point cloud.

Figure 13. Point cloud of target.
CONCLUSIONS

For the situation of reconstructing specific target that is mixed in complex scene, point cloud segmentation based on RANSAC and Euclidean cluster extraction need to be executed before surface reconstruction based on concave hull.

Experimental results show that the 3D reconstruction method in this paper can reconstruct a fine 3D model which is mixed in complex scene.

REFERENCE

10. Information on http://pointclouds.org