Acquisition and Registration of Human Point Cloud Based on Depth Camera

Huibai Wang¹, Shengyu Wang²

North China University of Technology, Beijing, China

Computer Department, North China University of Technology, Beijing, China

53112932@163.com, 448231847@qq.com

ABSTRACT

Registration of human cloud points is a key step in 3D human reconstruction. So, this paper proposes using depth camera to acquire human point cloud. Then, using feature matching and PnP (Perspective-n-Point) to solve the camera pose. Finally, use least square to optimize camera pose and implement point cloud registration. Experimental results show the effectiveness of this method.

Keywords: Feature Point Detection; Depth Camera; Point Cloud Registration.

1. INTRODUCTION

With the development of virtual reality and augmented reality, the real 3D human body model is applied to games, movies and animations. It has great commercial value. But the traditional 3D human model acquisition usually uses a large scanner or laser scanner to obtain multi view data; Then, by scanning the mark points in the body part, it achieves the registration of point cloud data and get the complete human point cloud; but these 3D scanners are expensive, complex operation. Therefore, the use of traditional scanners to obtain human point clouds has the disadvantages of high cost and difficult operation. With the development of hardware, a low cost depth camera is available on the market. These cameras can capture color and depth images. These cameras is easy to operate and cheap. So we use Microsoft's Kinect depth camera as a scanning device to capture human point clouds. But because of the limits of equipment, We can't get complete human cloud data at a time, so we need to repeatedly scan the body multi angle, and through the relevant registration algorithm achieve all point cloud transform into the same coordinate system.

2. OVERVIEW

In many applications of machine vision, such as stereo matching, image registration and shape recognition, etc. Registration of point clouds has always been a critical step. Point cloud registration is the registration of point clouds from different camera coordinate systems into the world coordinate system, achieve the object stitching, and get a complete object point cloud model. The accuracy of the point cloud registration is related to the complete model of the object. At present, the commonly used registration methods are genetic algorithm, least square matching method, three point pair method and ICP algorithm. ICP algorithm is a commonly used algorithm for point cloud registration, but the requirement of cloud point is very high, and the point cloud of registration should have most of the coincidence points, otherwise it will fail to register. Therefore, this paper proposes the use of depth camera to obtain human skeletal feature points pairs, and using SIFT algorithm for image feature extraction and matching^[1]. Finally, PnP (Perspective-n-Point) is used to solve the camera pose and make the optimization to realize the precise registration of point cloud with least square^[2].

3. CLOUD ACQUISITION AND REGISTRATION

Based on the depth camera, the human point cloud registration method is divided into three steps: (1) Use Kinect to get spatial color image, depth image and body color image, depth image, skeleton feature points, and filtering of these images to remove the noise^[3]. (2) Use SIFT feature extraction algorithm to extract and match feature points of two frames of color images, and select the wrong matching points. (3) Add the extracted skeleton pairs to the matched feature points, according to the depth information of the image, use PnP (Perspective-n-Point) to solve the camera pose of the 3D point to the 2D point. (4) Construction of minimum projective error to the pose obtained and solve. (5) The human image is transformed into human point cloud, and the human point cloud registration is implemented with the desired pose.

3.1 Image Capture

We use Microsoft Kinect cameras to capture spatial color images and depth images. Since the color and depth images of camera structure are not in the same coordinate system, it is necessary to align color images with depth images, we use the function MapDepthFrameToColorFrame of the Kinect SDK to map the depth image into color image space. Due to use the high 13 bits of depth image pixel value save a depth value, the lower three bits to save the user number information. The human body can be identified and the foreground of the human body is extracted from the image, get color and depth images of the human. While using the Kinect skeleton information to trace the 20 skeleton of human body. Since the storage order of skeleton points in each frame image is consistent, the skeleton points of each frame image can be matched into feature points, and the number of feature points is increasing.

3.2 Feature Point Extraction and Matching

In the field of computer vision, there are many stable local image features, such as the famous SIFT^[1], SURF^[4], ORB^[5] etc. They are repeatable, distinguishable, efficient, local. In this paper, we select the appropriate algorithm to extract features by comparing the number of extracted features, the time and the matching quality of different algorithms. The experimental results are shown in table 1:

Table 1. The connectivity detection result of video with large skin color area

Algorithm	Number of feature extraction	Take time /ms	Registration effect
ORB	500	14.257	commonly
SURF	89	62	commonly
SIFT	380	350	good

Through experimental comparison and analysis, considering the registration effect and time, we select SIFT (Scale-Invariant, Feature, Transform) for feature extraction. The algorithm takes full account of the changes of illumination, scale, rotation and so on in the process of image transformation, and the calculation is accurate. So we choose SIFT as the feature detector and descriptor extractors and extract the feature points from the two frames panoramic image captured above. Use SIFT to extract feature points x^m_t ($m=1,2,\dots,M$) in image I_t , and feature points x^n_{t+1} ($n=1,2,\dots,N$) in the image I_{t+1} . Using the Brute-Force Matcher measure the distances of descriptors between each feature point x^m_t and all x^n_{t+1} and sort them. Select the closest point as the matching point^[6].

The matched feature points have a large number of mismatches, so we need to do a filter, the algorithm follows:

(1) Filter the matched points and delete the feature points at distances less than K .

$$K = \frac{\min\_dist + \max\_dist}{2} \quad (1)$$

- (2) Subtract the coordinates of the matching points against F_t and F_{t+1} , obtaining the point P_i ($i=1,2,3,\dots$) and storing the P_i in the array M.
- (3) Because the camera usually moves horizontally, the point of $|p_i| > 90$ and the corresponding depth value $d=0$ of the feature points are deleted and storing the P_i in the array M2.
- (4) Count the number of points in the array M2 in which the PI is greater than zero, denoted as a (the number of forward moving points), and the feature point subscript i corresponding to the PI point is stored in the array Ma. The number of points less than zero is denoted as B (the number of negative translation points), and the feature point subscript i corresponding to the PI point is stored in the array Mb.
- (5) If the $a > b$ shows Image forward translation, negative translation for noise, so the feature points in the array Ma are correct matching points, whereas the point pairs in the Mb are taken. The result of the filter is shown in figure 1:

Figure 1. Feature point screening results

As you can see in figure 1:a, there is an incorrect matching point pair, and the results in figure 1:b are exactly filtered by the algorithm and the results clearly eliminate the wrong matching points. So, the accurate matching of image feature pairs is obtained.

3.3 Using PnP to Solve Camera Pose

PnP (Perspective-n-Point) is a method of solving 3D to 2D point pair motion. It describes how to estimate the pose of a camera when the n 3D space points and their projection positions are known. So we need to use depth image information to transform one point of feature points and the skeleton feature points into 3D points. The coordinate formula is as follows:

$$p(X, Y, Z)^T = \left(\frac{(i - cx)z}{fx}, \frac{(j - cy)z}{fy}, z \right)^T \quad (2)$$

The i, j coordinates of feature points, z for the corresponding depth information, f_x , f_y means focal length, c_x, c_y said photocenter. Now we have the 3D space coordinates and their projection positions with n feature points, and we use the direct linear transformation to solve the PnP problem. Now we know that the point in space is p, and the homogeneous coordinate is $p = (x, y, z, 1)^T$ in which the projection point is $x_1 = (u_1, v_1, 1)^T$ in the

image I_2 . We define an augmented matrix of $3 \times 4[R|t]$ and obtain the projection relation of P and x_1 , as shown in equation (3):

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3)$$

Eliminate s of the formula(3) and get the formula (4):

$$\begin{aligned} t_1^T P - t_3^T P u_1 &= 0, \\ t_2^T P - t_3^T P v_1 &= 0. \end{aligned} \quad (4)$$

At the formula (3) $t_1=(t_1,t_2,t_3,t_4)^T$, $t_2=(t_5,t_6,t_7,t_8)^T$, $t_3=(t_9,t_{10},t_{11},t_{12})^T$, The two linear constraints on t are derived by formula (4), and now there are N characteristic points, get the formula(5), Therefore, when the feature point pairs are greater than 6 pairs, we obtain the camera pose^[7].

$$\begin{pmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ P_N^T & 0 & -u_N P_N^T \\ 0 & P_N^T & -v_N P_N^T \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = 0 \quad (5)$$

3.4 Use Least Square to Optimize Camera Pose

Because of the camera noise and other reasons, the camera pose is only an initial value, so we need to optimize the camera pose. Therefore, We get the formula(6) according to the characteristics of $P_i=[X_i, Y_i, Z_i]^T$, and the pixel projection of the $u_i=[u_i, v_i]^T$, the camera pose, R, T (expressed as ξ Lie algebras).

$$s_i [u_i, v_i, 1]^T = K \exp(\xi^\wedge) [X_i, Y_i, Z_i, 1]^T \quad (6)$$

Due to the unknown camera pose and the noise of the observation point, the equation has errors. We construct the least squares problem (7) to minimize the sum of errors, and find the optimal solution of the camera pose:

$$\xi^* = \arg \min_{\xi} \frac{1}{2} \sum_{i=1}^n \left\| u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i \right\|_2^2 \quad (7)$$

In order to solve the equation, we need to know the derivative of each error term about the optimization variable, $P'=[X', Y', Z']^T$ is the space point coordinates of the P_i transform to the camera coordinate system, according to formula (6), get the formula (8):

$$u = f_x \frac{X'}{Y'} + c_x, v = f_y \frac{Y'}{Z'} + c_y \quad (8)$$

By the difference between u, v and observation, Get the error e . take the derivative of e with respect to feature points and camera pose, get the formula (9) (10):

$$\frac{\partial e}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{-f_x X'}{Z'^2} & \frac{-f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & \frac{-f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & \frac{-f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \quad (9)$$

$$\frac{\partial e}{\partial P} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{-f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & \frac{-f_y Y'}{Z'^2} \end{bmatrix} R \quad (10)$$

According to the formula (9) and the formula (10) we use Levin Berg Marquardt method for solving equation (7) to obtain the optimal solution of the camera pose. Finally, we transform the color image of human body into the point cloud according to formula (2), and multiply the point cloud and camera pose to realize the point cloud registration.

4. EXPERIMENTAL RESULTS

In this paper, we use depth camera to acquire human point cloud and implement point cloud registration. The experiment shows that the point cloud can be well registered, and the experimental results are shown in figure 2:

Figure 2. The result of Point cloud registration

Through the point cloud registration method, we can quickly use the low-cost depth camera to obtain the human point cloud for registration. At the same time, compared with the traditional point cloud registration method, it has the characteristics of high speed, small error, accurate registration, and the low correlation degree of point cloud. Compared with the traditional ICP point cloud registration method, the registration speed of this paper is very large, and the GPU acceleration point cloud registration is not needed, which reduces the demand for hardware.

Table 2. Registration time of different registration methods

Algorithm	Number of points cloud	Registration time /s
ICP	98323	14.257
Paper algorithm	98323	0.5

5. CONCLUSIONS

In this paper, the depth camera is used to separate the point cloud of human body from the object cloud. then, it used feature matching and PnP (Perspective-n-Point) to solve the camera pose. Finally, the least square is used to optimize the pose of the camera so as to achieve the registration of each cloud point. The accuracy of point cloud registration is improved, and this paper proposes a method to filter the false feature matching so as to improve the accuracy of the calculation. Compared with the traditional method of human point cloud registration, this paper reduces the hardware requirements and registration time, and improves the registration accuracy.

ACKNOWLEDGEMENTS

This work was supported by Fund as the Key Project of Science and Technology Plan of Beijing "The Research of Intelligent Retrieval and Data Mining Technology Oriented Video Promotion Service"(D16110100520000) and the dominant Discipline Project of North China University of Technology(XN044).

REFERENCES

- [1] D.G.Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*. Papers 60(2), 91-110 (2004).
- [2] L.Haomin, Z.Guofeng, and B.Hujun, "A survey of monocular simultaneous localization and mapping," *Journal of Computer-Aided Design and Computer Graphics*. Papers 28(6), 855-868 (2016).
- [3] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 127-136 (2011).
- [4] H.Bay, A.Ess, T.Tuytelaars, L.Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*. Papers 110(3), 346-359 (2008).
- [5] Rublee, Ethan; Rabaud, Vincent; Konolige, Kurt; Bradski, "ORB: an efficient alternative to SIFT or SURF," *IEEE International Conference on Computer Vision (ICCV)*, 2564-2571 (2011).
- [6] M.Muja and D.G.Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, 331-340 (2009).
- [7] V.Lepetit, F.Moreno-Noguer, and P.Fua, "Epnp: An accurate $O(n)$ solution to the pnp problem," *International Journal Computer Vision*. Papers 81(2), 155-166 (2008).
- [8] Guo Yu, M.Bennamoun, F.Shoel, "3D object recognition in cluttered scenes with local surface features: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(11), 2270-2287 (2014).