Alignment and Super Pixel Segmentation of RGB-D Video Stream

Lianjun Liao[1, 2, 3], Yongbin Hao[2], Xiangyang Su[1], Shihong Xia[1]
[1]*Institute of Computing Technology, CAS, Beijing, 100190*
[2]*North China University of Technology, Beijing, 100044*
[3]*University of Chinese Academy of Sciences, Beijing, 100049*
E-mail:LiaoLianjun@ict.ac.cn

Abstract

In this work, the RGB-D video stream is captured from Microsoft's V2 Kinect, and is used for alignment and super pixel segmentation. We found an effective method to align the RGB-D video stream. Then the aligned depth video stream is optimized by the joint-bilateral filtering algorithm. The 3D scene can be reconstructed by the time and space alignment of the RGB-D video stream. Moreover, we proposed a new segmentation method for RGB-D video stream, which uses the K-means clustering method to produce super pixels. For the first time, we introduce the optical flow information of the video stream into super pixel segmentation. With the position, color, depth, and the optical flow information between the front and back frames of the color video, our new algorithm can make a more accurate super pixel segmentation. Finally, we do several experiment to demonstrate the effect of augment method.

Keywords: Alignment; Super pixel Segmentation; Joint bilateral filtering.

1. Introduction

Now the application of stereo vision is very wide, it can be used in aviation and remote sensing measurement, industrial automation and mobile robot automatic navigation, medical and health industry, and other fields. In 2010, Microsoft launched the Kinect camera. With it, the game players will play the somatosensory game. Why does the Kinect camera[4] has such a super power? The reason is that it integrates two kinds camera of: color camera and depth camera. Through the RGB-D video stream alignment, it can reconstruct the real scene to playing the game.

Image segmentation divides an image into regions which fulfill two criteria: intra-region similarity and inter-region dissimilarity. The super pixel segmentation focuses on intra-region similarity, possibly dividing an image into more segments than necessary. Such an over-segmentation greatly reduces scene complexity and can be used as the basis of advanced and expensive algorithms[1].

The alignment of the RGB-D video stream can be used for 3D scene reconstruction. Super pixel segmentation of video frame can be applied in pattern recognition, target identification and tracking applications.

1.1. Our work

In this paper, we captured the RGB-D video stream from the Microsoft's V2 Kinect binocular camera, and then align the depth video stream with the color video stream in time domain and space domain to reconstruct the 3D scene. In the process pipeline, we augment the DASP[1] method with the inter-frame information. Finally we show several experimental result to demonstrate the effect of our augment method, and it shows that our algorithm works well and can achieve perceptually better segmentation result.

1.2 Main contributions

We present an alternative effective solution for alignment and segmentation for RGB-D video stream. The contribution of this paper is:

(1) We found an effective alignment method for RGB-D video stream, and employ the joint bilateral filtering algorithm to optimize the depth video stream.

(2) On the basis of the DASP[1] algorithm, we proposed a new super pixel segmentation algorithm for the RGB-D video stream.

(3)The optical flow information between the video frames is first introduced into super pixel segmentation for moving objects in video.

2. Related work

The super pixel segmentation problem have been under intensive study, and now there are many super pixel

segmentation algorithm. Although these methods can achieve image segmentation in many scenario, there are some problems in the segmentation of complex images: edge loss, image blur and so on. In this paper, we give an effective alignment and segmentation method for RGB-D video stream. This section gives a brief introduction to the alignment of RGB-D video stream, the filtering of the image and the related work of image segmentation.

2.1. About the RGB−D image alignment

Since two camera in the Kinect displaced side by side, the color and depth videos stream capture different views of the same scene. In order to align the color image and the depth image, it is need to set up the geometric model of the camera, and need to obtain the camera parameters. In the geometric model of the camera, there exist four coordinate systems: world coordinate system, camera coordinate system, pixel coordinate system and image coordinate system. The relationship between these four coordinate systems is shown in figure 1.1[16]:

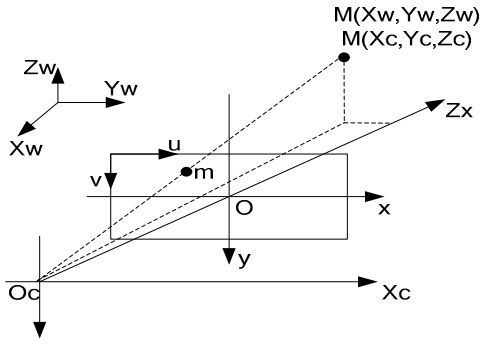


Figure 1.1 *The relationship of coordinate systems*

(1) The world coordinate system (X_w, Y_w, Z_w) is a user defined 3D space coordinate system, used to describe the coordinate position of objects in 3D space.

(2) The origin of the camera coordinate system (X_c, Y_c, Z_c) is the optical center of the camera, Z_c axis and the camera's optical axis coincide and perpendicular to the imaging plane, X_c, Y_c axis respectively parallel with the image coordinates of x, y axis, O_cO is f the focal length of the camera.

(3) The third one is the pixel coordinate system (u, v). Its origin is the upper left corner of the image, and its unit is pixel.

(4) The forth one is the image coordinate system. Its origin is the center of the image, and it is a physical coordinate system with unit of millimeter (mm).

Only establishing the geometric model of the camera, the color image and the depth image can still not be aligned. We do not know the relative position of these two cameras, and cannot convert the coordinate position from the pixel coordinate system to the world coordinate system. Therefore, it is need to obtain the internal and external parameters of the color camera and the depth camera. Zhengyou Zhang in 1998 proposed "a flexible new technique for camera calibration"[7], he used a calibration board to calibrate the color camera and the depth camera, and then estimated the intrinsic and extrinsic parameters of the camera. The basic principle is to establish the imaging model of the camera. The camera's internal and external parameters is calculated using the least squares method according to the relationship between the coordinates of the calibration plate in the three-dimensional coordinate system and the coordinates of the same calibration plate in the pixel coordinates system.

After the geometric model of camera is established and the internal and external parameters of the camera is acquired, the depth image can be aligned to the color image by means of coordinate transformation. The process of image alignment is as follows:

(1) Using the relationship between the pixel coordinate system of depth image and its image coordinate system, the depth image is converted from the depth pixel coordinate system to depth image coordinate system;

(2) Applying internal parameters of the depth camera, the pixels coordinates of the depth image is changed to the depth camera coordinate system;

(3)Then we convert the depth image from the depth camera coordinate system to the color camera coordinate system, by utilizing external parameters of the depth camera and the color camera;

(4) The internal parameters of the color camera was employed to transform the depth image from color camera coordinates to the color image coordinate system;

(5) Finally, the depth image is mapped to the pixel coordinate system of color image according to the relationship between the pixel coordinate system of color image and its image coordinate system;

With step (1)-(5), we can convert a depth image from the pixel coordinate system of a depth image to the pixel coordinate system of the color image, which can realize spatial alignment between the depth image and color image. Using the Microsoft V2 Kinect device to get the RGB-D video stream, aligned frame by frame, as a result, the whole RGB-D video stream can be aligned in space. The specific process of image alignment is shown in the 3.3 chapter.

2.2. Image filtering

The depth image captured from Microsoft's V2 Kinect contains a lot of noise. These noises will seriously affect the quality of the image, bring obstacles to image analysis.

Some filtering algorithms can be used to remove these noises. There are a lot of Image filtering (denoising) algorithm, Such as Gauss filter, mean filter, median filter, maximum uniformity filter, wavelet domain filtering, joint bilateral filtering, etc. Compared with these filtering algorithms, the joint-bilateral filtering algorithm[5] can

keep more of the original edge information of the image. In order to improve the performance of the program, we also make use of coarse-to-fine[15] technology.

2.3. Super pixel Segmentation

The super pixel segmentation divides the pixels into several of meaningful regions, which is used to replace the rigid structure of the pixel grid. Now there are a lot of super pixel segmentation algorithm, such as SLIC[3], DASP[1]. SLIC algorithm is proposed in 2011 by Radhakrishna Achanta et al. It is a super pixel segmentation algorithm using K-means with the spatial domain and color domain as coefficients. DASP algorithm is proposed in 2012 by David Weikersdorfer et al. This is also a super pixel segmentation algorithm using K-means. The difference between DASP and SLIC algorithm is the input data. The input of SLIC algorithm is only the color image, but the DASP's input is both the color image and the corresponding depth image. The above algorithms has the problem of image segmentation, image texture and information loss. In this paper, we adapt the DASP algorithm by making full use of the optical flow information in the video frames. With the improve algorithm, we can achieve more accurate segmentation of the moving object.

3. Alignment of RGB-D Video Stream

RGB-D video stream was captured using the Kinect and then carry out the time alignment on the video stream. After Invalid depth points were removed, the RGB-D video stream is spatially aligned. Then the depth video flow is optimized by applying the joint-bilateral filtering algorithm[5], that is, repaired the "holes" in the depth video frames. After the RGB-D video stream having been optimized, we can carry on the super pixel segmentation algorithm on it. Figure 3.1 shows the process steps of this paper.

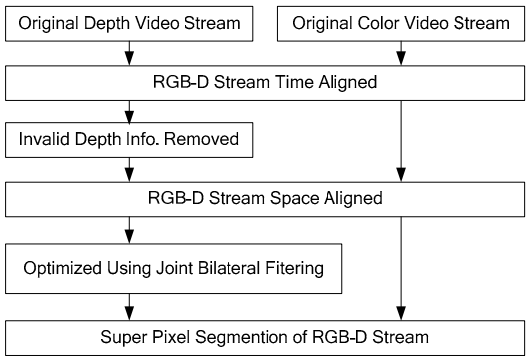


Figure 3.1 *The process steps of our algorithm*

The color video stream and the depth video stream captured from Microsoft V2 Kinect camera is not aligned. The alignment of video stream contains two aspects: time and space alignment. For the spatial alignment, we should choice which coordinate system as the target one: depth camera coordinate system or color camera coordinate system or other else[2]. Whatever camera coordinate system is choose as target one, seemly no means to avoid of the semi occluded region[12]. Since we hope the color image is more reliable, we select the color camera coordinate system as the target one, and then using projection methods, the depth frame is aligned to the color frame. The information of color image is enhanced by using the aligned depth image.

3.1. Camera Calibration

In order to project depth image to color image, firstly, we should calibrate camera to get the internal parameters and external parameters for the color camera and the depth camera. Since the difference between the color camera and depth camera is very small, the system deviation of Kinect depth camera can be ignore. Although there existing more advanced calibration methods, according to the experience of camera parameters calibration, it is reliable that the internal and external camera parameters are obtained by the basic calibration. We employ Zhengyou Zhang method[7] to obtain the internal and external camera parameters.

3.2. Remove Invalid Depth Pixels

In depth image from Kinect, the depth value of the pixel on an object edge is not reliable. In order to improve the accuracy of depth image, it is necessary to remove the invalid depth values. Firstly, image edge is detected using the Sobel operator. Secondly, the depth gradient along the detected edge is calculated through the calculation of the Sobel approximation of the 3*3 matrix. Then filter out the invalid depth value by a gradient threshold. Generally, the gradient threshold should choice from the range [100,200] (unit mm) .Note that the threshold is too small will filter out reliable depth value, and will distort the geometry of the object; the threshold is too general in the depth map leave too many invalid depth value. Empirically, this paper uses a threshold of 150 mm.

3.3. Time Alignment of RGB-D Video Stream

RGB-D video frames captured from the Microsoft V2 Kinect is not one-to-one corresponding in time. With the SDK for Microsoft Kinect, using the function Color Basic-D2D and Depth Basic-D2D we can only obtain the color image and depth image, not the RGB-D video stream. In order to get the RGB-D video stream, this paper modifies these two SDK functions to capture the RGB-D video stream. In order to ensure that the frames of RGB-D video stream is synchronized in time, the color video frame and the depth video frame are stored in the memory

alternatively. The dual channel DDR3 memory with bandwidth of 5.4GB/s is used in the experiment, the time interval for alternately write color video and depth video frames is within 0.2 milliseconds. The acquisition frame rate is 30fps, for both the Kinect V2 color camera and depth camera. The time to collecting a frame is roughly 33.33ms. Because of the time interval for video frame saved to memory is very small, the relative 33.33ms of 0.2ms is only 0.6%, we can assume that the color frame and depth frame is capture at the same time. This ensures that the RGB-D video stream is aligned in time. Because of the RGB-D video data size is very large, so the program runtime environment require large computer memory. In our experiment, the running memory used by the computer is 16G. If saving the RGB-D stream to hard disk, the program does not need a very large memory. However, saving data to disk is very slow; it will seriously reduce the RGB-D video frame rate. By using the improved acquisition of RGB-D video stream program, the color frame and depth frames of the captured RGB-D video stream is one-to-one corresponding in time.

3.4. Spatial Alignment of RGB-D Stream

Since the color camera and depth camera are not in the same position, so there is a parallax between them, which also led to that the color video frame and depth video frame's pixels is not one-to-one corresponding in space. In order to get the depth value of each pixel in the color frame to reconstruct the scene of RGB-D video, the RGB-D video stream is need to be aligned[6]. Frame by frame aligned the color and depth frame, we can achieve the spatial alignment of RGB-D video stream. The following is the steps for spatial alignment for the RGB-D video stream:

The corresponding relationship between the image coordinate system and the pixel coordinate system can be expressed by the following vector and matrix.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \Delta x & 0 & u_0 \\ 0 & \Delta y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.1)$$

Where$(u,\ v)$ represents a point in the pixel coordinate system. The point$(u_0, v_0)$is the original point of the pixel coordinate system. Δx,Δy represents a pixel's scale unit along x,y axis in the image coordinate system. The（x, y）represents point in the image coordinate system correspond to $(u,\ v)$.

The corresponding about the image coordinate system and the camera coordinate system is formulate by the following vector and matrix:

$$Z_c \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad (3.2)$$

Where f is the focal length of the camera, $(X_c$, Y_c , $Z_c)$ is the coordinate of a point in the camera system correspond to the point (x, y) in the image coordinate system, Z_c represents the depth value of the point $(u,\ v)$ the pixel coordinate system.

The formula(3.3) shows the relationship of the camera coordinate system and the world coordinate system.

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (3.3)$$

Where R is a 3*3 rotation matrix, T is 3*1 translation matrix. Using the above formulation, can convert a point in the camera coordinate system to the world coordinate system.

Combined the formulation（3.1）,（3.2）,（3.3）, we will obtain the following equation:

$$Z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \Delta x & 0 & u_0 \\ 0 & \Delta y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} x_y & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0^T & 0 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (3.4)$$

Where $\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$ represents the camera's internal parameters, R is a rotation matrix, T is a translation matrix, R, T represents the external parameters of the camera.

The rotation matrix R and the transfer matrix T can be obtained by camera calibration of the relative position of the color camera and the depth camera. Therefore, the point in the depth camera coordinate system can be transformed into the color camera coordinate system by applying R and T transform. United formula (3.1) and (3.2), we can draw a conclusion that color camera coordinates of points multiplied by the color camera internal parameters, we can get the color image pixel coordinates of the point.

According to the above steps, the depth of the RGB-D video stream can be aligned to the color frame, so as to realize the spatial alignment of the RGB-D video stream.

4. Filtering and Super Pixel Segmentation of RGB-D Video Stream

4.1. Joint-bilateral Filtering

After removing invalid depth pixels, the depth image will have some holes. Since the resolution of original depth image is smaller than that of color image, the aligned depth

image will have more holes. At the same time, the depth image from the V2 Kinect has many noise points and the quality of depth image is affected by these noises. In order to repair the holes on the aligned depth video stream frames and improve depth video stream quality, in this paper, we use the joint-bilateral filter algorithm[5].

Joint-bilateral filtering algorithm has two kernel functions, respectively in the space domain of the depth image and the color domain of the color image. Filtering by this algorithm, it can complement the holes and reduce noise in the depth image. The input of the joint-bilateral filter is the color image and the aligned depth. The calculation formula is as follows:

$$\mathrm{Depth}(X) = \frac{1}{w}\sum_{y\in N_X} w_c * w_s * d(y) \qquad (4.1)$$

$$\mathrm{w} = \sum_{y\in N_x} w_c * w_s \qquad (4.2)$$

N_X is a first order neighborhood of X, and y is a point in the neighborhood. w_s is linear correlation with pixel space, the greater the distance the less of correlation, the smaller weight w_s. w_c is related to the pixel color, the more similar the color is, the more of correlation and the greater the weight w_c. w is the product of w_c and w_s. Make joint use of w_s and w_c, it achieve the affections of smoothing image and edge preserving.

The calculation formulas for w_c and w_s are as follows:

$$w_c(i,j) = exp\left(-\frac{\left(C(i,j)-C(x,y)\right)^2}{2\sigma_c^2}\right) \qquad (4.3)$$

$$w_{s(i,j)} = exp\left(-\frac{\left((i,j)-(x,y)\right)^2}{2\sigma_s^2}\right) \qquad (4.4)$$

Where, $\mathrm{C}(x,y)$ is the color of neighborhood center point (x,y), $C(i,j)$ is the color of point (i,j) in that neighborhood, σ_c and σ_s is standard deviation of the Gaussian function.

Each pixel in the depth image needs to be filtered. When the neighborhood area is relatively large, the computational pressure of the program will rise up very soon. In order to raise the computational efficiency of the program, we employ the coarse-to-fine technology to improve the running efficiency of the program, by filtering the image at multilevel and multi-resolution.

Following will give a briefly review the each step of the method[2]. The method uses images of n level resolution, 0 to the n-1 level is from high to low resolution. Each level k has two inputs and one output, which has the same resolution. The input is color image C_k and the aligned depth image D_k, and the output is the depth F_k whose hole has been repaired.

In n-1 layer, having the low resolution, the input is the aligned depth image and color image. The output F_{n-1} is obtained by using joint bilateral filter algorithm, which is optimized depth image and its holes is repaired. Except the n-1 layer, all the other layers are calculated by the following steps:

By selecting a pixel sequentially in the X axis and the Y axis every g pixel, the color image C_k and depth image D_k is down sampled respectively into the C_{k+1} and D_{k+1},which is then used as the input of the k+1 layer for continuous down salmping. After al l of the lower layers having been optimized recursively, K + 1 layer will get the optimized depth image F_{k+1}.

In the $\mathrm{K}+1$ layer, repairing the holes of D_{k+1}, and obtain an optimized depth image F_{k+1}. Then the F_{k+1} is up sampled to complement the depth image U_k in the K layer. The up sampling is reverse progress of down sampling.

In the n-1 layer, also using the joint bilateral filter, complementing the depth image D_{n-1}, and generate the output F_{n-1}. Then the F_{n-1} is up sampled to complement the depth image U_{n-2} in the n-2 layer.

By make use of the joint-bilateral filtering and the use of coarse-to-fine technology to accelerate the progress, we can be more efficient to get a higher quality and well-aligned depth video stream.

4.2. Optical Flow

Optical flow is a 2D vector field, where each vector represents motion displacement vector of a pixel point from the current frame to the next frame. The arrow indicate the direction of moving objects, while the size of the arrow can also indicate the speed of physical movement.

In this paper, we use the optical flow information between the color video frames to enhance the image segmentation, which makes the image segmentation more accurate.

There are many kinds of methods to calculate the optical flow, for example, the Lucas-Kanade method[9] is to calculate the optical flow of some point sets. Horn-Schunck method[8]. Machieal Black method[10] and Farneback Gunnar method[11], these three methods can calculate the dense optical flow, that is, for each pixel points are calculated optical flow. By contrast with Horn-Schunck method and Farneback Gunnar method, the method of Black Machieal is higher in the accuracy of the optical flow and is more efficiency of calculation. Therefore, we choose the Black Machieal method to calculate the optical flow.

4.3. Super Pixel Segmentation Algorithm

The super pixel segmentation algorithm in this paper is an improvement of the DASP[1] algorithm. Similar to the DASP algorithm, the improved segmentation algorithm also uses the K-means method to cluster the pixels in color image. DASP algorithm uses a vector F to represent the property of pixels, the vector contains the location of the pixel point, color and depth. In our augmented algorithm, we add an element (pixel optical flow) to the vector in the vector F. After the optical flow being introduced, the segmentation of moving objects in depth video stream is more accurate. The in improved segmentation algorithm

consists of three steps, the first step is to select k super pixels, the second step is to calculate to determine each pixel of the image belongs to which super pixel, and the third step is to get the mean value of each super pixel, so generating a new super pixels.

In order to get the initial k super pixels, all of the first is to calculate the density of the depth image. Assuming that the super pixel is a circle disc with a radius of R, where the coordinates of center point is $(i,\ j)$, the depth value is $\mathrm{D}(i,j)$, f is the focal length of the depth camera. Using the pinhole model, this super pixel is projected onto the image plane. The radius r of the projected image plane can be computed by the formula (4.5):

$$\mathrm{r}(i,j) = \frac{f}{D_{(i,j)}}\mathrm{R} \qquad (4.5)$$

For points that are not parallel to the image plane, it need to be calculated with the slope of these points relative to the image plane. We can use the depth gradient$\nabla\mathrm{D}(i,j)$as an approximation slope. So the area of projection area in the image plane can be obtained by the formula (4-6):

$$\mathrm{A}(i,j) = \frac{r(i,j)^2\pi}{\sqrt{(\nabla D(i,j))^2+1}} \qquad (4.6)$$

The super pixel density can be defined as: $\rho(i,j) \propto \frac{1}{A(i,j)}$. According to the density of the depth image, the center of k super pixels can be found.

The initial segmentation does not guarantee that the super pixels are segmented along the edge of the texture. In order to respect the texture edge of the image, it is need to determine each pixel of the image belongs to which super pixel. Using formula (4.7) to calculate the weight:

$$Weight = W_p + W_c + W_d + W_{uv} \qquad (4.7)$$

$$W_p = k1 * (p1 - p0)^2 \qquad (4.8)$$

$$W_c = k2*(c1 - c0)^2 \qquad (4.9)$$

$$W_d = k3 * (d1 - d0)^2 \qquad (4.10)$$

$$W_{uv} = k4 * (uv1 - uv0)^2 \qquad (4.11)$$

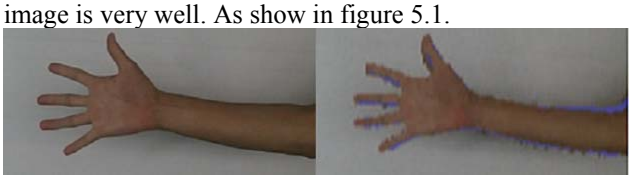
Where $k1, k2, k3, k4,$ respectively, represents the position, color, depth, optical flow coefficient. $p1$ is the coordinates of the current pixel, $p0$ is the coordinates of super pixel; $c1$ indicates the color of the current pixel, $c0$ indicates the color of super pixel; $d1$ is the depth of the current pixel, $d0$ is the depth of super pixel; $uv1$ represents optical flow of the current pixel. $uv0$ represents optical flow of super pixel. Note that the $Weight$ is to represent the weight of super pixels, which is calculated by (4.7).

When the pixel's $Weight$ is smaller than the super pixel's weight, the pixel belongs to the super pixel, otherwise, it does not belong to it. In order to determine all the pixels of a special super pixel, it is need to traverse the entire image. Obviously, if the number of super pixels is too many, and then the number of iterations would be very large, so that the performance of the program is relatively low. With the observation that only the pixels near the super pixel has relatively large in correlation, and it is very small in correlation for the far in distance pixels. Therefore, we only calculate the pixels within a certain range. This will greatly improve the performance of the program. When all the pixels have been determined belongs to which super pixels, it is necessary to calculate the mean value of the super pixel, and it is need to obtain new super pixel's the center, color, optical flow and normal direction. After several number iterations of the K-means algorithm, we can get the finally segmented image.

5. Experimental Results and Analysis

Using the above process pipeline in this paper, CPU is the Intel (R) Xeon (R) CPU E3-1240 V2, memory is 16GB, operating system is Windows7, the software environment is Open CV, Visual Studio 2012 and Matlab R2015a. In order to verify the correctness of the algorithm, we do some experiments with data set from the video from Kinect V2 and data set from DASP[1].

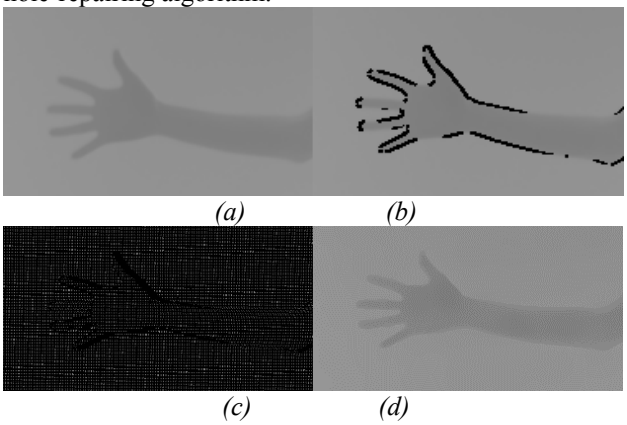
Experiment 1: Firstly, convert the coordinates of points in both the color image and the depth image into the color camera coordinate system. Thus, we got a point cloud in the color camera coordinate system. The aligned point is colored with the color from the color image, and we got a colored point cloud. By comparing the original color image (left) and aligned point cloud (right), we can found that the alignment effect of the color image and the depth image is very well. As show in figure 5.1.



(a) *(b)*

Figure 5.1 *Alignment Effect of Hand Point Cloud. (a)original color image,(b) Alignment Well with Depth Image. The point where absent depth information is assigned with blue color.*

Experiment 2: The experiment check the effective of hole-repairing algorithm.



(a) *(b)*

(c) *(d)*

Figure 5.2 *Repairing holes of deep video stream.(a) original deep image.(b)removed the invalid depth point.(c)the aligned deep image.(d)the optimized result. As we can see that the*

optimizing algorithm works well and image quality improved very much.

After the depth image aligned to the color image, some pixel of the color image will have no corresponding depth value. That is, there exist some "holes" in the depth image.

In Figure 5.2, using the human hand data set captured from Microsoft V2 Kinect, shown that our hole repairing methods works very well. The sub figure (b) shows there exist much holes after removing the invalid deep points, and from (c), we can see much more holes after alignment. The sub figure (d) shown that the image quality has improve much more after using the joint-bilateral filtering algorithm.

Experiment 3: In this experiment, we demonstrate our improved super pixel segmentation by comparing the segment result with DASP algorithm and SLIC algorithm.

In the figure 5.3, it is the segmentation result of using data set of human body captured by Microsoft's V2 Kinect. The result of SLIC algorithm, shown as sub figure (b), is lack of sense of hierarchy which may make the body and hand in the same block. As we can see from sub figure (c), DASP algorithm will lose some image information and the edge is relatively rough. Sub figure (d) is the result of our improved segmentation algorithm. In comparison, our segmentation is more accurate, and there is less loss of image.

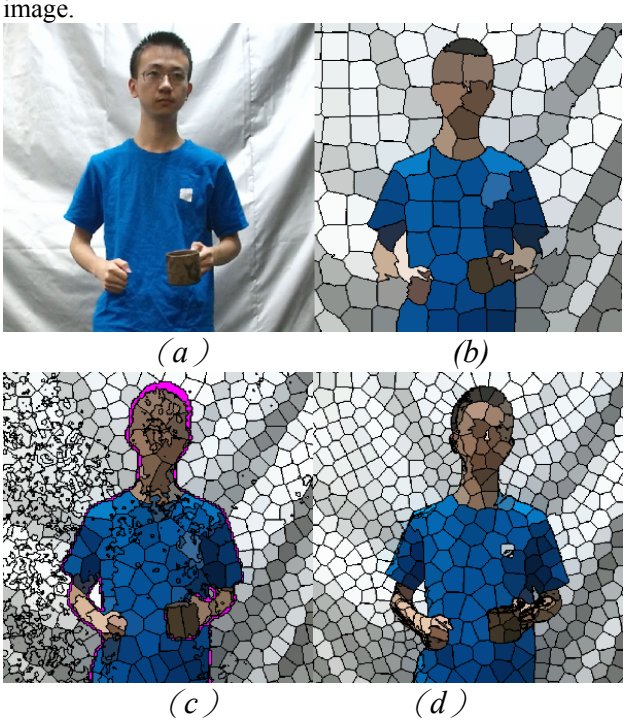


Figure 5.3 *Super pixel segmentation for human body. (a)original image, (b)result of SLIC algorithm, (c) result of DASP algorithm, (d) and result of our methods. Our segmentation is more accurate, and there is less loss of image.*

In following figure 5.4, is the result from the hand data from Microsoft's V2 Kinect, also shown that (b)the SLIC algorithm got a fuzzy edge of the hand and has destroyed the texture of the original image,(c)the result of DASP algorithm lose many image information. By contrast, our improved method (d) can keep more of the original image information and got a more accurate edge of the hand.

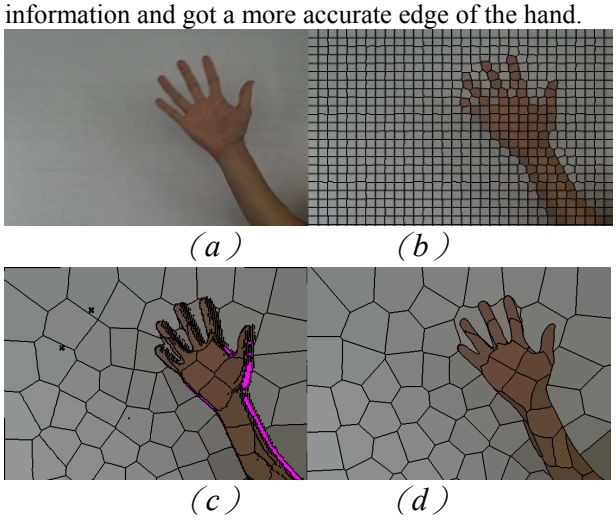


Figure 5.4 *Super pixel segmentation for human hand RGB-D video stream. (a)original image,(b) result of SLIC algorithm, (c)result of DASP algorithm,(d)and result of ours. Our method can keep more of the original image information and got an accurate edge of the hand.*

Experiment 4: In this experiment, we use the data set from DASP[1] to verify the segmentation effect.

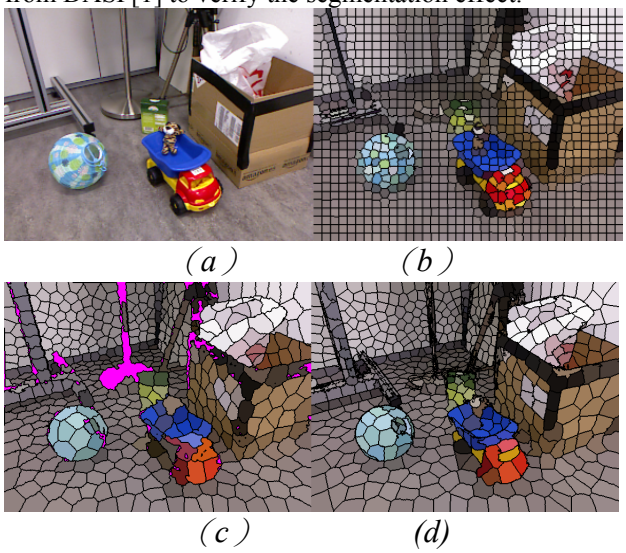


Figure 5.5 *Super pixel segmentation for lab scenes RGB-D video stream. (a)original image, (b)result of SLIC algorithm, (c)result of DASP algorithm, (d)and result of our new super pixel segmentation methods. Our method got an accurate edge and keep much more of the original texture information.*

In figure 5.5, it is obvious shown that (b)the SLIC algorithm is poorly to handle the object edge and has destroyed the texture of the original image,(c)the result of

DASP algorithm lose image information and the edge of object is not very clear. By contrast, our improved method (d) can keep much more of the original texture information and the edge is more accurate.

6. Conclusion and discussion

In this work, we established a pipeline to align the RGB-D video stream and perform super pixel segmentation algorithm on it. Based on DASP algorithm, we introduced the optical flow information of the video stream into super pixel segmentation. Through the experiment result analysis, our method works well and make a more accurate super pixel segmentation. Although the improved video stream super pixel segmentation algorithm is more accurate, but there are many smaller super pixels on the edge, so the super pixel segmentation algorithm can still be improved by makes larger super pixel block on the edges of objects.

Acknowledgment

This work was financially supported by "Training plan" for the cross training of high level talents in Beijing colleges and Universities.

References

[1] Weikersdorfer D, Gossow D, Beetz M. Depth-adaptive superpixels[C]//Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012: 2087-2090.

[2] Richardt C, Stoll C, Dodgson N A, et al. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2012, 31(2pt1): 247-256.

[3]Achanta R, Shaji A, Smith K, et al. SLIC superpixels compared to state-of-the-art superpixel methods[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2012, 34(11): 2274-2282.

[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finoc-chio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. IEEE Conference on Computer Vision and Pattern Recognition, 2011

[5] K OPF J., C OHEN M. F., L ISCHINSKI D., U YTTEN - DAELE M.: Joint bilateral upsampling. ACM Transactions on Graphics (Proc. SIGGRAPH) 26, 3 (2007), 96. 2, 4

[6] L INDNER M., K OLB A., H ARTMANN K.: Data-fusion of PMD-based distance-information and high-resolution RGB-images. In Proc. ISSCS (July 2007), pp. 121–124. 2, 3

[7] Zhang Z. A flexible new technique for camera calibration [J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000, 22(11): 1330-1334.

[8] Barron J L, Fleet D J, Beauchemin S S. Performance of optical flow techniques[J]. International journal of computer vision, 1994, 12(1): 43-77.

[9]Bruhn A, Weickert J, Schnörr C. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods[J]. International Journal of Computer Vision, 2005, 61(3): 211-231.

[10] Roth S, Black M J. On the spatial statistics of optical flow[J]. International Journal of Computer Vision, 2007, 74(1): 33-50.

[11] Farnebäck G. Two-frame motion estimation based on polynomial expansion[M]//Image analysis. Springer Berlin Heidelberg, 2003: 363-370.

[12] Lindner M, Kolb A, Hartmann K. Data-fusion of PMD-based distance-information and high-resolution RGB-images[C]//Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on. IEEE, 2007, 1: 1-4.

[13] Gao W, Zhang X, Yang L, et al. An improved Sobel edge detection[C]//Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. IEEE, 2010, 5: 67-71.

[14] Sharifi M, Fathy M, Mahmoudi M T. A classified and comparative study of edge detection algorithms[C]//Information Technology: Coding and Computing, 2002. Proceedings. International Conference on. IEEE, 2002: 117-120.

[15] Charniak E, Johnson M. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking[C]//Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005: 173-180.

[16] Camera calibration - the principle of camera calibration. http://www.aiuxian.com/article/p-162432.html. 2012,7,20