GRAPH-CUT BASED MOTION SEGMENTATION WITH PHYSICAL SUPPORT RELATIONSHIPS IN CROWDED URBAN AREAS FROM A MOVING RGB-D CAMERA

Ching-Hsiang Hsu

Graduate Institute of Networking and Multimedia
National Taiwan University
Taipei, Taiwan

July 2014

Submitted in partial fulfilment of
the requirements for the degree of
Master of Science

Advisor: Chieh-Chih Wang

Thesis Committee:
Li-Chen Fu
Yung-Yu Chuang
Chu-song Chen
Sheng-Luen Chung (Department of Electrical Engineering, National Taiwan University of Science and Technology)

ABSTRACT

MOTION segmentation is an important and challenging problem in computer vision and robotics. Because of its complexity, this problem can be approached from respective angles depending on sensors, environment, and motions themselves. By utilizing RGB-D video captured by a structured light range sensor, we proposed a novel and robust algorithm to segment motions from consecutive frames. Based on a modified random sample consensus algorithm (RANSAC), we exploit the coherence of color and spatiality in the scene to estimate motion. It inherits the computational efficiency and probabilistic robustness from the RANSAC paradigm. After aligning two point clouds by the estimated transformation, we combine the output of inlier and outlier distribution with the prior knowledge of the RGB-D images to conduct segmentation by a graph-cut optimization scenario. Moreover, we introduce physical support relationships to better understand the motions in the environment. We provide a RGB-D dataset captured in a crowded urban environment to demonstrate our idea. Comparing to several motion segmentation methods in the same pipeline, we show that our approach performs well in the highly dynamic scene.

Keywords: Motion Segmentation, Moving Object Detection, RGB-D video, Physical Support Relationship

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In a number of applications using data from a moving camera in dynamic scenes, for instance, navigation, tracking, localization, simultaneous localization and mapping (SLAM) or structure form motion, it is inevitable to encounter multiple moving objects. Take SLAM for example. If the observed static and dynamic landmarks cannot be well separated, after updating the information of those landmarks, in the prediction step, new positions of the landmarks may become inaccurate. Furthermore, it has a big influence on mapping result and causes self-localization error. Therefore, in the work simultaneous localization, mapping, and moving object tracking (SLAMMOT) (Wang et al., 2007), it takes the moving objects into account to enhance the performance of SLAM. Generally, RGB-D motion segmentation aims to partition regions in one RGB-D image with same estimated motion in 6 degree of freedom (DoF). While, motion estimation and segmentation is a chicken-and-egg problem. In other words, the result of motion estimation and segmentation will influence each other.

In our approach, we argue that motion estimation has to be executed first. Since we are lack of shape pre-knowledge of diverse moving objects, it is arduous for us to segment motions. For example, in a traffic scenario, if we have the shape model of cars, by detecting one of components from a driving car, we have more evidences to segment the car accordingly. In addition, segmentation approaches applied to each frame independently often lead to unstable results. For instance, if the segmentation is based on color, it is difficult to fine-tune parameters to adjust to different lighting environment. Thence, we separate the problem of motion segmentation into two tasks. First, to get reliable transformations, a

modified RANSAC paradigm is conducted for motion estimation based on probabilistic assumption with over-segmented block units. It is capable of adapting to a certain percentage of outliers and return trusty transformations. Second, by observing the inlier and outlier distribution from the output of aligning two point clouds, we utilize this information to introduce likelihood relation in an undirected graph and encode the prior knowledge like color and position of the RGB-D image to the nodes in the graph. Since that within a block unit inside or on the boundary of a moving object, the color and depth values for each point are not all stable. If only the first task is carried out, it is hard to determine the parameters to group those block units. Thus, the second task is to deal with those uncertain cases. We intertwine the result of the first task with the second task to form an organic whole. By optimizing the graph, we can obtain a good quality motion segmentation with respect to motion, color and spatial relation. Finally, we propose the idea of physical support to visualize the motion intensity and help understand the semantic meaning between the motion segments.

More details about the two tasks and related pre-processing steps are revealed in Chapter 3. At first, we provide an approach overview of our framework. Then, we elaborate each aspect in the subsections. Mainly, our contribution is at section 3.3 and 3.4. In Chapter 2, we cover several related works on the problem of motion segmentation. We point out the characteristic and insufficiency of those works and highlight the merit of our approach. Our experimental results are demonstrated in Chapter 4. In Chapter 5, We make a conclusion and discuss the future work.

CHAPTER 2

Related Work

THERE are a number of approaches to tackle the problem of motion segmentation in a variety of scenes depending on the density of features, the characteristic of features, the optimization algorithm, the representation of motion segments and so on. From the viewpoint of sparse motion segmentation, (Costeira and Kanade, 1998) utilized sparse 2D trajectories as features to find consistent motion groups in an independent moving objects scenario based on the factorization method introduced by (Tomasi and Kanade, 1992). (Yan and Pollefeys, 2006) gave a summary of 2D trajectory approach and addressed the problem on many issues like articulated motion, non-rigid motion and etc. (Schindler, 2005) clustered corresponding 3D points to obtain a 3D motion by modelling the set of multi-view correspondences as an irregular Markov random field (MRF). (Perera and Barnes, 2013) used the variance of spatial distance between interest points to formulate the problem as solving a maximal clique enumeration problem of the similarity graph. Similarly, (Wang and Huang, 2013) classified motion by grouping sparse features according to a physical constraint in spatial distance. However, under the circumstance which it is not guaranteed that sparse interest points are stable and detectable, the accuracy of the segmentation result may drop accordingly.

Dense motion segmentation targets to group motion in pixel or voxel level. Moreover, with the accurate dense map from a RGB-D camera compared to one built by a stereo camera, more and more researchers try to solve the problem of motion segmentation by using dense features from a RGB-D camera for the purpose of enhancing the performance competed with sparse ones. Besides using sparse 2D correspondences, (Hadfield and Bowden, 2011) worked on extracting dense 3D correspondences to estimate scene flow which

is derived from optical flow approach based on particles. (Van den Bergh and Van Gool, 2012) proposed an over segmentation method by integrating a RGB-D image with optical flow estimation for motion segmentation. (Unger et al., 2012) improved the occlusion problem in optical flow by joint motion estimation and segmentation from the boundary information of segments in the RGB images. (Herbst et al., 2012) assumed that when objects are well separated in terms of the spatial distance, they can be segmented by motion cues based on a RANSAC inlier selection method and conditional random field (CRF) for post-processing. (Narayana et al., 2013) proposed a mixture model for segmentation based on optical flow orientation. Since it only models translation, it is prone to error when the rotation of camera occurs. (Stückler and Behnke, 2013) formulated the problem of motion segmentation into an expectation maximization (EM). In the E-step, it gives segments labels by using the estimated motion parameters in a graph-cut optimization with color and normal information; in the M-step, it estimates motion parameters by using iterative closest points method (ICP). However, the M-step the author speculates is unable to support the robustness of the algorithm. (Peter Ochs, 2013) exploited long-term motion cues to find trajectories in RGB images. Based on these semi-dense point trajectories, a dense segmentation is conducted through a color-based variational approach. However, to find reliable point trajectories, it relies on the performance of optical flow whcih is severely effected by lighting sources. (Hickson et al., 2014) constructed a graph by utilizing the information of depth, color, and optical flow estimation to do segmentation in a RGB-D video. Although it gives a good and efficient segmentation, when motion occurs, the result turns into unstable on the border of two moving objects. (Chiang, 2014) argued that the priority of motion estimation and segmentation will have a big influence on the result. In addition, he introduced a prioritized Gauss-Seidel method to accomplish motion segmentation. Yet, when the depth measurement becomes unstable, the performance may have a significant drop.

Regard to the mentioned insufficiency and assumptions, in this paper, we introduces a modified RANSAC paradigm to robustly estimate motion parameters in consecutive frames. Instead of applying ICP for motion estimation, by sampling in an over-segmented block with the coherence of color and spatiality with optical flow estimation, we use singular value decomposition (SVD) to calculate transformations. Theoretically, under the RANSAC paradigm, it can achieve 99% successful rate to pick inlier correspondences in the model. Then, we exploits the aligned result of motion estimation from point clouds to

form an energy function for segmentation optimization. This smoothness function is for handling unstable color and depth information of RGB-D images. Since we demonstrate our approach outdoors at night, some materials may influence the depth measurement and light sources may cause the performance degradation of optical flow estimation. In addition, inspired by (Silberman et al., 2012), we integrate physical support relationships with the result of motion segmentation to show the intensity of motion segments and help us figure out the semantic meaning between motion segments in the provided RGB-D data at crowded urban areas.

CHAPTER 3

Motion Segmentation

THIS chapter will discuss the detail of motion segmentation. In our approach, there are three main parts to accomplish motion segmentation: motion estimation, split process, and merge process. Before entering either main parts, there is a sequence of pre-processing procedure. After pre-processing, to estimate motion, first, we randomly sample adequate points in an over-segmented block to make sure that those selected points are all inlier and then use SVD to estimate corresponding transformation. We assumed that from a reasonable over-segmented block, probabilistically, it can sustain noise and result in a reliable transformation. Second, we exploit the transformation result to find inlier and outlier points of the segment. If the inlier ratio surpasses a threshold, this segment is considered to be a merge segment and will be put into a segment pool. If there are multiple merge segments in the segment pool, we will conduct a merge process. On the other hand, a split process will be carried on. In the split process, based on the result of inlier and outlier rate, we form the likelihood links in an undirected graph and utilize the color and spatial information of a RGB-D image as prior in the graph. We apply the min-cut/max-flow algorithm with pre-computed over-segmented blocks to optimize the graph in order to segment different motions; in the merge process, with the inlier ratio of the segment exceeding a threshold, we will have this segment pairwisely combine with other segments in the segment pool to test whether they can be merged together. If not, the segment will be put into the segment pool as well. Finally, we integrate the result of motion segmentation with the pre-processed physical support relationships to not only visualize the intensity of motion segments, but also try to grasp the semantic meaning between those motion segments.

3.1. Approach Overview

In this section, we introduce the framework of our system in Figure 3.1. At the beginning, we sequentially pre-processed the current RGB-D image in four steps: dense optical flow estimation, ground detection, RGB-D over-segmentation, and physical support relationship. Based on the over-segmented result, by conducting RANSAC method, each motion of the over-segmented block unit can be estimated accurately. In the first iteration of segmentation, we initialize the input RGB-D image as one segment. Subsequently, after using estimated transformations in the over-segmented block to align point clouds, we can get a highest inlier ratio. If the resulting inlier ratio is below a threshold, the process will jump into the split procedure. In the split process, the segment in the over-segmented block units may be divided into two smaller segments. On the contrary, if the inlier ratio is above a threshold, the process will run into the merge procedure. The segment may be merged with other segment in the segment pool. An integration will be applied to motion segments and physical support relationships until all segments are put into the segment pool. In the following sections, we will go through the details of each part.

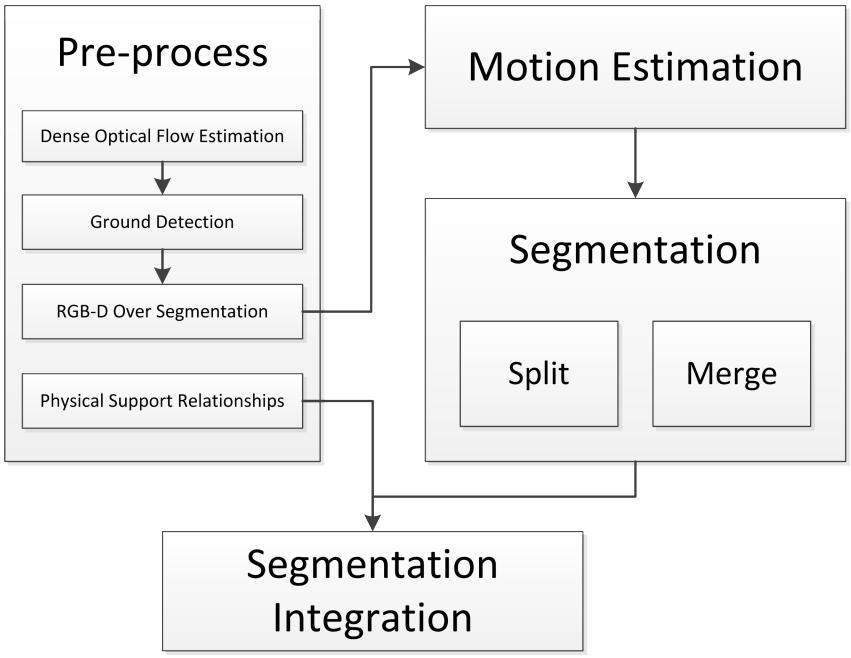


Figure 3.1. Workflow

3.2. Pre-process

3.2.1. Dense Optical Flow Estimation

In our framework, we use the GPU-based dense optical flow estimation (Werlberger et al., 2010) because it shows a great performance on the Middlebury optical flow benchmark and it can run closely in real-time. In Figure 3.2, it can give us a good initial 2D correspondence. But, in the outdoor scenario at night, the performance of the dense optical flow may be dramatically influenced by lighting sources.

(a) a good result

(b) a bad result

Figure 3.2. Pre-processed result of dense optical flow estimation. (a) shows a good result. Images from left to right are the previous frame, the current frame, and the optical flow result. (b) is a bad result since there is glare at the left top of the image. As you can see, there is an obvious abnormal part coloured in light green.

Furthermore, to calculate transformation matrix in 6 DoF, we make use of the interpolation method 3.1 to approximately get the 3D correspondence. Assumed that point p_a represents one of the pixel in the image and point p_b is p_a's optical flow result. Since the result of dense optical flow usually is a floating-point number, we utilize $floor$ and $ceil$ function to obtain integer indices. This way, by looking up the index table of the point cloud, we can get the corresponding 3D coordinates nearby. The Figure 3.3 illustrates the Equation 3.1, where x_i, y_i, z_i are the 3D coordinate at point p_i and px_i, py_i are the 2D coordinate at point p_i. i is from 1 to 8. Our target is to use 2D interpolation method to calculate

3D coordinate x_b, y_b, and z_b. One thing need to be careful of is that if the depth of the corresponding index in the point cloud is 0 or NaN, we will not take this invalid point into account.

$$
\begin{aligned}
x_5 &= x_1 + (x_4 - x_1) \times \frac{(px_5 - px_1)}{(px_4 - px_1)} \quad x_6 = x_2 + (x_3 - x_2) \times \frac{(px_6 - px_2)}{(px_3 - px_2)} \\
x_b &= x_6 + (x_5 - x_6) \times \frac{(py_b - py_3)}{(py_4 - py_3)} \quad y_7 = y_2 + (y_1 - y_2) \times \frac{(py_7 - py_2)}{(py_1 - py_2)} \\
y_8 &= y_3 + (y_4 - y_3) \times \frac{(py_8 - py_3)}{(py_4 - py_3)} \quad y_b = y_7 + (y_8 - y_7) \times \frac{(px_b - px_1)}{(px_4 - px_1)} \\
z_5 &= z_1 + (z_4 - z_1) \times \frac{(px_5 - px_1)}{(px_4 - px_1)} \quad z_6 = z_2 + (z_3 - z_2) \times \frac{(px_6 - px_2)}{(px_3 - px_2)} \\
z\prime &= z_6 + (z_5 - z_6) \times \frac{(py_b - py_3)}{(py_4 - py_3)} \quad z_7 = z_2 + (z_1 - z_2) \times \frac{(py_7 - py_2)}{(py_1 - py_2)} \\
z_8 &= z_3 + (z_4 - z_3) \times \frac{(py_8 - py_3)}{(py_4 - py_3)} \quad z\prime\prime = z_7 + (z_8 - z_7) \times \frac{(px_b - px_1)}{(px_4 - px_1)} \\
z_b &= \frac{z\prime + z\prime\prime}{2}
\end{aligned}
\tag{3.1}
$$

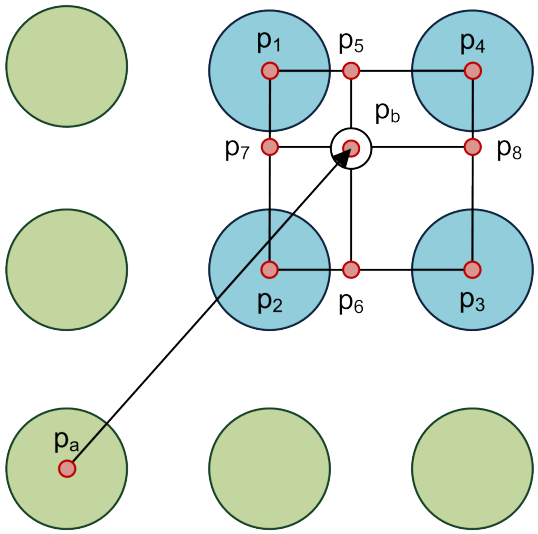
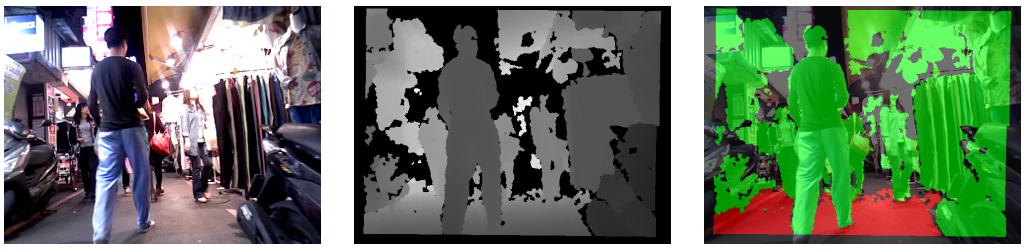


Figure 3.3. The green and blue dots are image pixels. The white dot p_b is the result position of dense optical flow. By interpolating the 3D coordinates of the red dots numbered from 1 to 8, the corresponding 3D coordinate of point p_b can be approximately calculated.

3.2.2. Ground Detection

To find the ground is an important step in our framework. Except for detecting moving object, we need to use the ground information to find physical support in subsection 3.2.4. If the ground is abnormally detected such as floating in the air, there may be weird conditions happened in finding physical support, for example, an object A which has farer distance from the real ground than object B can be topologically placed neared to the ground than object B. This way, in the following, if we have to determine those floating fragments, it is likely to have wrong association. Another problem is that if the detected ground includes motions, it will face a failure when estimating static segments because we assume that the ground is stationarily fixed. Basically, the algorithm we use to detect ground is the RANSAC based plane model segmentation from PCL [1]. But, we make several assumptions in order to detect reliable ground. As the ground location is approximately at the bottom to the middle of the RGB-D image, we only use half of the RGB-D image in the detection process. Besides, sometimes, when people are close to the camera, the detection may fail because compared to the plane on the ground, there is a bigger plane which is perpendicular to the ground caused by people. To solve the problem, we assume that the camera will not be held upside down or have a large difference of pitch angles. In other words, its moving direction is roughly parallel to the ground. This way, we can iteratively examine the detected planes until we find one that its plane parameter suits the assumption mentioned above. In Figure 3.4(c), the red part is the detected ground through the detection process.



(a) the color image of the frame k (b) the depth image of the frame k (c) a ground detection result of the frame k

Figure 3.4. Pre-processed result of ground detection.

[1]PCL: Point Cloud Library, http://pointclouds.org/documentation/tutorials/planar_segmentation.php

3.2.3. Over Segmentation

We apply over segmentation for two reasons. Inspired by (Yang et al., 2010), the first reason is that to get reliable transformation matrices, the RANSAC procedure can get rid of noise and outliers. Although RANSAC is effective for model fitting when there are outliers, if there are multiple motions in the scene, merely using trivial RANSAC is unable to obtain dependable transformation efficiently. We briefly review the foundation and the probabilistic formulation of RANSAC Equation 3.2 and 3.3

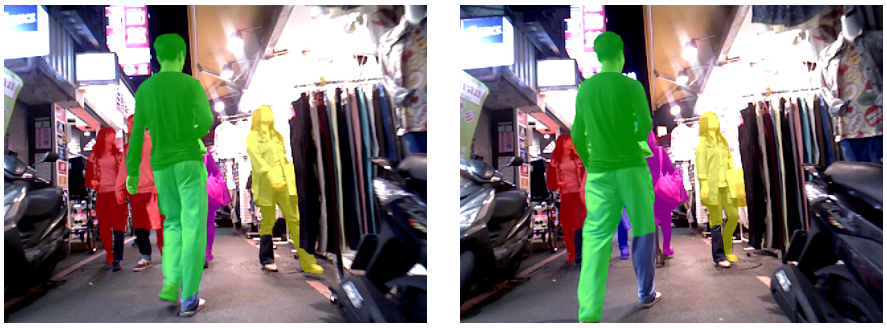
$$1 - p \geq (1 - w^n)^k \tag{3.2}$$

$$k \geq \frac{\log(1-p)}{\log(1-w^n)} \tag{3.3}$$

where w is the assumed probability of choosing s inlier points in each selection. Suppose that n points need to be used to estimate the model independently. w^n means that all n selected points are inlier. To maintain probability p which there will be at least one successful random selection, k must be equal to or greater than the value in Equation 3.3. However, this belief fails to consider the fact that to ensure with a static probability p regard to the environment, w is not tractable to the number of motion. For example, in Figure 3.5(a), if we want to sample two points to get the coloured motion group in four motion groups and assumed that the noise with probability 0.1 is uniformly distributed in the scene, from Equation 3.4

$$w = (1 - N) \times \prod_{\mathbf{c} \in \mathbf{C}} \frac{1}{M} \tag{3.4}$$

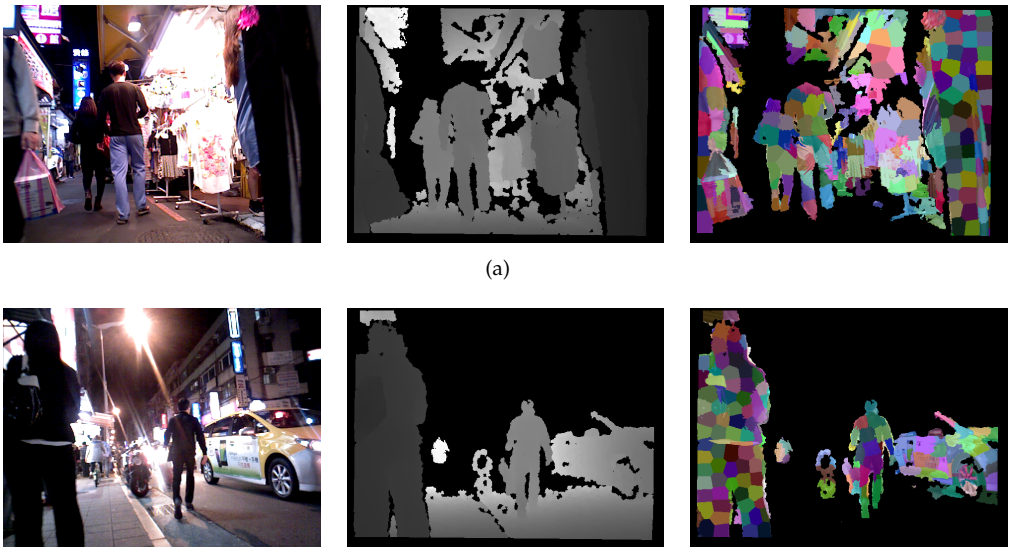
where N is the noise factor and c means the subset of points we want to sample in set C, the probability of sampling a motion group would be $0.9 \times 0.25 \times 0.25$. So as to solve the problem, the step of over-segmentation is to gather points with similar color and spatiality in the point cloud to form a block unit and sample points in the over-segmented block. Because of the coherent color and position of the block, the success rate is predictable and can be used to distinguish different motions by sampling sufficient points in the over-segmented block. The second reason is that instead of using every points to build the graph for optimization, we use over-segmented blocks as nodes in the graph. Each over-segmented block can describe a group of points with respect to their color and spatial information. Moreover, it



(a) four coloured motion groups (b) five coloured motion groups

Figure 3.5. The different color masks in (a) and (b) show the motion groups.

can be used for encoding other information like inlier rate. This way, time complexity can be improved significantly. After surveying RGB-D over segmentation methods, we adopt Simple Linear Iterative Clustering over segmentation (SLIC) (Achanta et al., 2012) as shown in Figure 3.6. It demonstrates a good quality on over segmentation, especially on the boundary of two objects. We projected the over segmentation result onto an image in Figure 3.6. In the figure, some over-segmented blocks are coloured black because most points inside the block are without depth information or on the detected ground.



(a)

(b)

Figure 3.6. Pre-processed over segmentation result. (a) and (b) are two over-segmentation results in different time stamps.

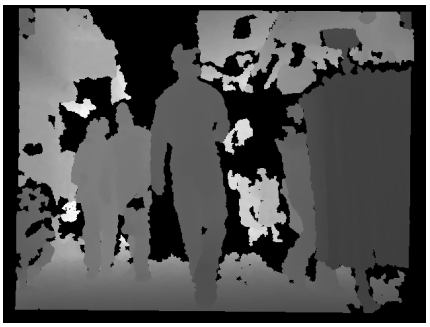
3.2.4. Physical Support Relationship

In this section, instead of showing motion segments with arbitrary colors as motion segmentation results, connecting our results with objects in real world is a better representation for understanding and visualization. Despite recognizing a variety of objects, to understand the physical meaning in the dynamic scene is a favourable alternative choice. Thus, to make the use of the physical constraints, the retrieved ground information in Subsection 3.2.2 can be used. Suppose all static and dynamic objects such as streetlights or pedestrians are supported by ground or other objects. Once, it is sure that on top of the ground there are voxels belong to an object, we do breadth first search to explore that object from those voxels. We assume that if voxels are part of the same object, without occlusion and greatly unstable measurement, all of them will be visited during one searching process. There are eight search orientation in the searching process. When the adjacent voxels are close enough in a threshold α, we group these voxels and the same physical support object together. Algorithm 1 shows the pseudo code for grouping physical support relationships. Admittedly, there is no denying that sometimes objects might be split by others or due to the limited perspective view only part of them could be seen. To deal with this issue, we explore those floating fragments without the prior of the ground by the same searching method mentioned above and we construct a kdtree based on the pre-computed physical support relationships. Then, we match every points in each floating fragment with the built kdtree to record which physical support relationship is the most closest one to the floating fragment. By clustering those floating fragments with their corresponding closest physical support relationship, the problem of unidentified floating fragments can be solved. In Figure 3.7(d), different physical support relationships are coloured distinctly. Some physical support relationships are mixed with multiple colors because the depth information is highly uncertain due to exceeding the suggested detected range distantly and special materials like glass.

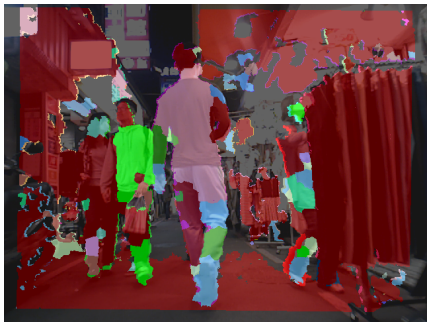
3.3. Motion Estimation

There are many motion estimation algorithms such as SVD, iterative closest point (ICP), Levenberg–Marquardt algorithm (LMA), and so on. The solutions of ICP and LM are iteratively calculated like Gauss-Newton method. They cannot give us a closed-form answer and may be resulted in a local minima solution. On the other hand, by using SVD

(a) the color image of the frame k



(b) the depth image of the frame k



(c) an integration result of the frame k

(d) physical support relationships of the frame k

Figure 3.7. Pre-processed physical support result. The ground in (d) is coloured black because more colors can be addressed to physical support relationships and it will not cause confusion.

motion estimation, we can sample least pairs of correspondences to achieve 99% successful rate based on the over-segmented blocks referred to subsection 3.2.3. Therefore, to pursue the efficiency and robustness in the RANSAC paradigm, we use SVD motion estimation to find reliable transformation matrices. Every over-segmented blocks are estimated one transformation in 6 DoF before next segmentation process. By sampling N 3D points in the current frame formed matrix A with their corresponding 3D points from optical flow estimation in subsection 3.2.1 formed matrix B, through SVD transformation estimation 3.6

$$\bar{A} = \begin{pmatrix} x_A^1 - \bar{x_A} & x_A^2 - \bar{x_A} & \cdots & x_A^N - \bar{x_A} \\ y_A^1 - \bar{y_A} & y_A^2 - \bar{y_A} & \cdots & y_A^N - \bar{y_A} \\ z_A^1 - \bar{z_A} & z_A^2 - \bar{z_A} & \cdots & z_A^N - \bar{z_A} \end{pmatrix} \quad \bar{B} = \begin{pmatrix} x_B^1 - \bar{x_B} & x_B^2 - \bar{x_B} & \cdots & x_B^N - \bar{x_B} \\ y_B^1 - \bar{y_B} & y_B^2 - \bar{y_B} & \cdots & y_B^N - \bar{y_B} \\ z_B^1 - \bar{z_B} & z_B^2 & \cdots & z_B^N - \bar{z_B} \end{pmatrix} \tag{3.5}$$

$$\bar{A}^T \bar{B} = U \Sigma V \tag{3.6}$$

Algorithm 1 Grouping Physical Support Relationships

Require: Given one RGB-D image, a label k, an integer table S for recording the label, a boolean table T for checking whether the location is visited or not, and a location $v(x, y)$ where either $(x-1,y-1)$, $(x,y-1)$ or $(x+1,y-1)$ is located at the ground.

Create a queue Q.
Set $T[x][y]$ is true.
Enqueue v onto Q.
Set $S[x][y]$ equals to k.
while Q is not empty. **do**
  $v\prime(x\prime, y\prime) \leftarrow Q$.dequeue()
  for all adjacent nodes of $v\prime$ from 8 directions: $(x\prime-1,y\prime-1)$, $(x\prime,y\prime-1)$, $(x\prime+1,y\prime-1)$, $(x\prime-1,y\prime)$, $(x\prime+1,y\prime)$, $(x\prime+1,y\prime-1)$, $(x\prime+1,y\prime)$, $(x\prime+1,y\prime+1)$ **do**
    $v\prime\prime(x\prime\prime, y\prime\prime) \leftarrow Q$.adjacentNode($v\prime$)
    if $T[x\prime\prime][y\prime\prime]$ is false **then**
      Set $T[x\prime\prime][y\prime\prime]$ is true.
      if the 3D Euclidean distance of $v\prime$ and $v\prime\prime$ is less than α **then**
        Enqueue $v\prime\prime$ onto Q.
        Set $S[x\prime\prime][y\prime\prime]$ equals to k.
      end if
    end if
  end for
end while
return table S

$$R = \begin{cases} V^T U^T, & \text{if } \left|V^T U^T\right| \geq 0 \\ V^{T\prime} U^T, & \text{if } \left|V^T U^T\right| < 0 \end{cases} \qquad V^{T\prime} = \begin{pmatrix} V_{11}^T & V_{12}^T & V_{13}^T \\ V_{21}^T & V_{22}^T & V_{23}^T \\ -V_{31}^T & -V_{32}^T & -V_{33}^T \end{pmatrix} \tag{3.7}$$

$$t = -R\bar{A} + \bar{B} \tag{3.8}$$

where matrix U, matrix Σ, and matrix V are 3×3 matrices, gives us a closed-form solution of rotation matrix R and translation matrix t. Suppose that by Equation 3.2 based on the coherence of one over-segmented block, we set $N = 4$ and $w = 0.5$, in order to acquire a 99% assurance of making at least one error-free selection, we repeat the above procedure to compute SVD transformation estimation at least 72 iterations to get a trusty motion estimation. After obtaining the reliable transformation, we align the segment with previous point cloud to calculate the inlier ratio. If the inlier ratio is below a hard threshold considering the measurement noise caused by the sensor, the process will run into a split process. On the contrary, if it is above the threshold, it will go into a merge process.

We determine an inlier point by a range threshold with respect to the precision of a structured light range sensor measurement. We model the range threshold parameter by using a simple extrapolation method 3.9

$$range(x) = \begin{cases} \varepsilon, & \text{if } x \leq 3.5 \; meter. \\ \varepsilon \times \dfrac{r}{3.5}, & \text{otherwise} \end{cases} \tag{3.9}$$

where ε is a given range threshold. r is the measured distance. In the equation, a linear model outputs the range threshold parameter which is out of suggesting range. Another issue needs to take care of is occlusion. For the purpose of enhancing the precision of the inlier ratio, we built a kdtree which contains all points transformed from the current point cloud to the previous point cloud. Assume that in the consecutive frames there is no drastic motion. By using the point in the previous frame to search in the built kdtree within a range, if at similar x and y coordinate the z value of that point is smaller than the z value of the finding point in the kdtree, we can eliminate the searched point in the current frame because the finding points are invisible in the previous frame. In other words, in the current frame, it is an outlier point. Therefore, we do not take it into account on calculating inlier ratio. To sum up, the time complexity of motion estimation is O($r \times n \times \log(m)$) where r is the number of RANSAC iteration, n is the number of used points in the motion segmentation, and m is the number of points in the previous frame. $\log(m)$ means the average time complexity in one kd-tree search process. The time complexity to determine the inlier ratio for next process is O($n \times \log(m)$), n is the number points in one segment, and m is the number of points in the previous frame.

3.4. Split Segment

Split procedure is one of the most important steps in the motion segmentation problem. In our approach, it tries to split one segment into two along the boundary of motions as much as possible. When there is a segment with unknown number of motions, we exploit the estimated motion result in section 3.3 from part of a moving object to separate it from other distinct motion. While we do not know the boundary of each motion segments, under the circumstance that we don't have the information of the shape of various moving objects and background, it is one of the most challenging parts in the motion segmentation. Despite the fact that we are not aware of the pre-knowledge, through observed geometry

and appearance information in current environment, we can efficiently solve the problem. We utilize the information to form a graph shown in Figure 3.8 and formulate an energy function 3.10.

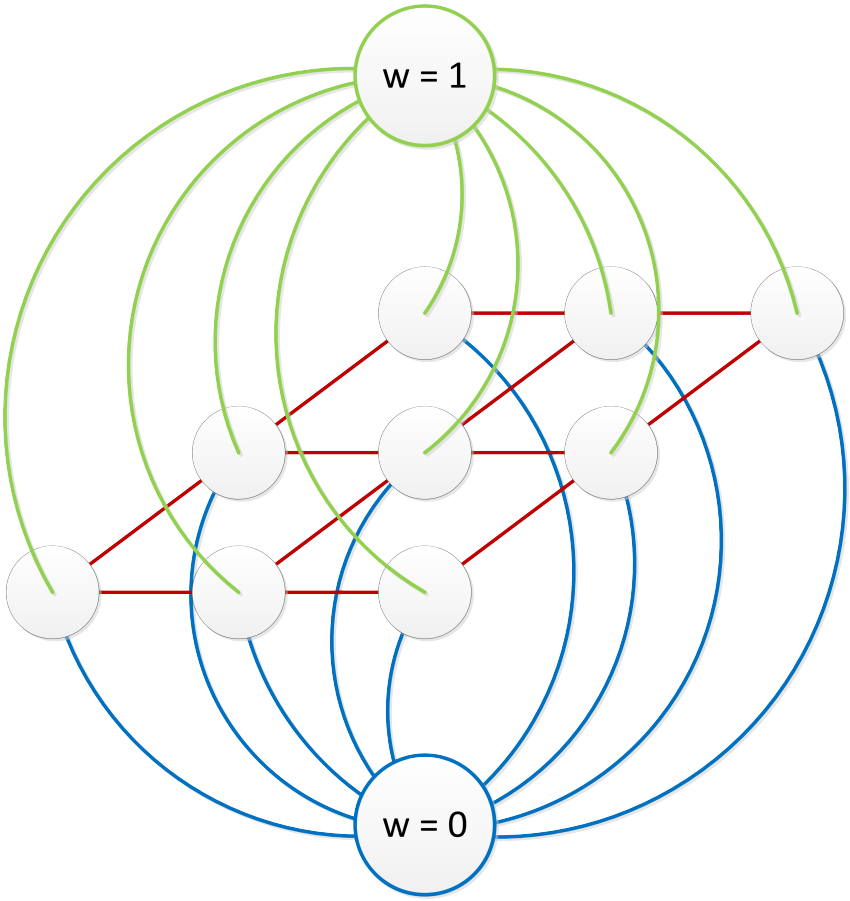


Figure 3.8. The graph represents the relation among each over-segmented block. The red link describes the coherence between two over-segmented blocks. The green link stands for the inlier rate of an over-segmented block. The blue link stands for the inlier rate of an over-segmented block.

$$\underset{x_i}{\operatorname{argmin}}\Big(\sum_{i\in\mathbf{V}} E_1(x_i) + \sum_{(i,j)\in\mathbf{E}} E_2(x_i,x_j)\Big) \tag{3.10}$$

Based on the over-segmented blocks in subsection 3.2.3, suppose that one segment is built as a graph $\mathscr{G} = <\mathscr{V}, \mathscr{E}>$ where $\mathscr{V}$ is the set of all nodes which also represent the over-segmented blocks and $\mathscr{E}$ is the set of all links connecting adjacent nodes in the graph $\mathscr{G}$. Combining with the obtained reliable transformation, after aligning two point cloud and

have inlier and outlier output as feedback, we encode the inlier and outlier rate as likelihood energy 3.11

$$E_1(x_i) = \begin{cases} r_i^{out}, & \text{if } w_i = 1 \\ r_i^{in}, & \text{if } w_i = 0 \end{cases} \tag{3.11}$$

where x_i is the block state to be determined, $w_i = 1$ means the block's tendency to be outlier and $w_i = 0$ means the block's tendency to be inlier. To describe the tendency mentioned above, we use the outlier rate, r_i^{out} and the inlier rate, r_i^{in} of the blocks. But, it is insufficient to determine the state of the nodes only by the likelihood energy. Therefore, we define a prior energy function 3.12 which indicate the relation of spatial and color between node i and j.

$$E2(x_i, x_j) = |x_i - x_j| \times \frac{1}{1 + \beta \times D_{ij} + \gamma \times C_{ij}} \tag{3.12}$$

where $|x_i - x_j|$ indicates the boundary relationship between different states. If it equals one, it allows us to record the coherence of two blocks. D_{ij} is the L1-norm of the distance between block i and j; C_{ij} is the L1-norm of the CIELAB color difference between block i and j. β and γ are weighting factors for deciding the influence of spatiality and color to the problem and the environment. The CIELAB color space is one of the most complete color spaces which describes all the colors visible to the human eye and was created to serve as a device-independent model. We utilize the prior energy function 3.12 to represent the spatial and color gradient along the object boundary. The prior energy 3.12 can be viewed as a constraint when assigning two different states to the blocks. If two uncertain blocks have coherent position and similar color, they get more chances to be segmented to the same state. On the other hand, if two uncertain blocks differ from each other a lot, the prior energy between them becomes large and forces them to be separated to different state. In Figure 3.9, it demonstrates three split iteration. In the middle column, the intensity of color yellow shows the inlier rate of the blocks during the split process. After each split process, one segment may be divided to two smaller segments, inlier and outlier segment. That is, it is possible that the segment will not be separated because of the similarity of color, spatiality, inlier rate, and outlier rate from the blocks. To minimize the energy function 3.10, we follow the standard optimization method in computer vision to adopt the min-cut/max-flow algorithm (Boykov et al., 2001; Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004).

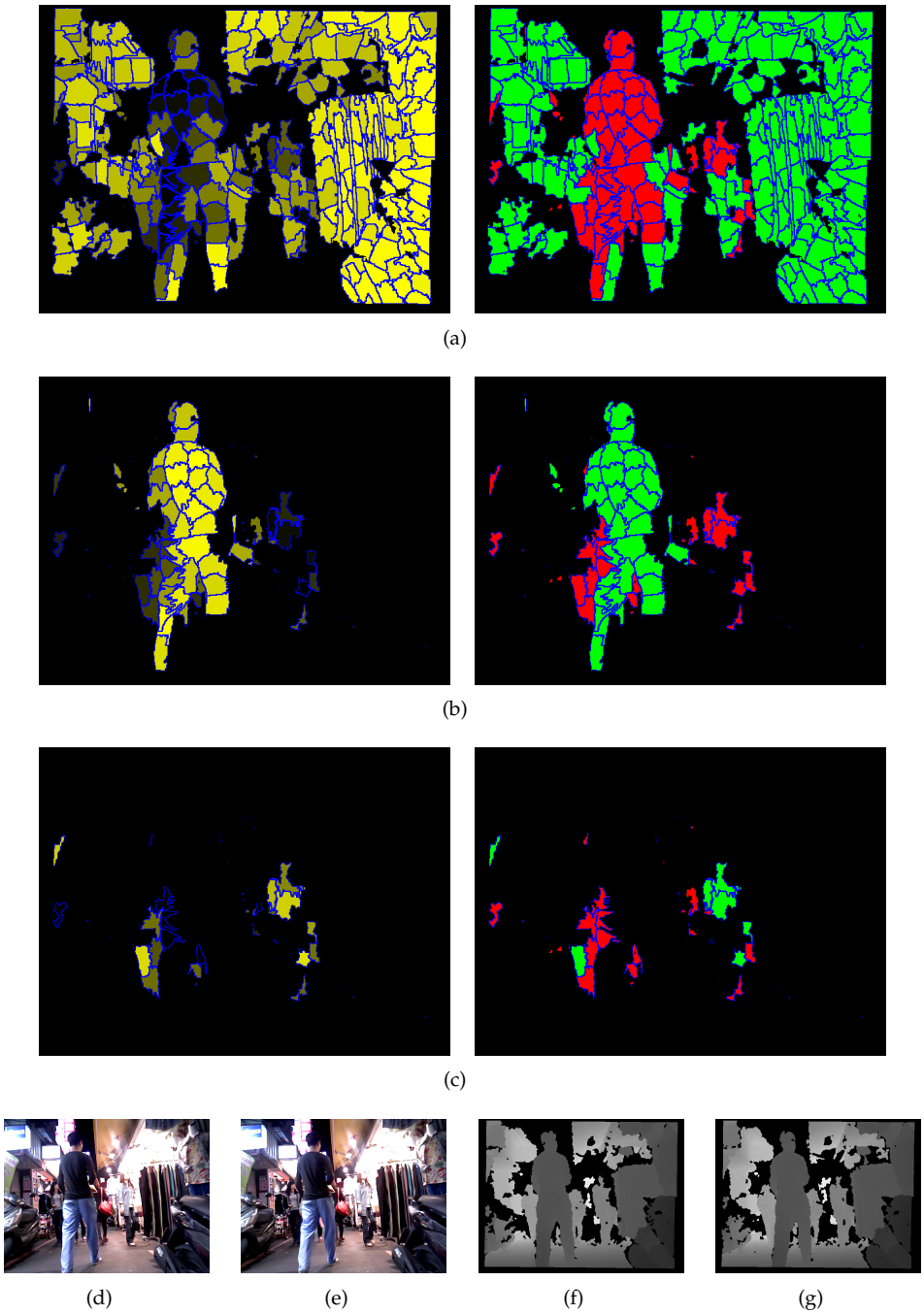


Figure 3.9. Three consecutive iterations in the split process. (a)-(c) illustrate the distribution map of inlier points and the final state after splitting process. (d)-(g) are the input of the consecutive color and depth frame.

3.5. Merge Segment

After getting a motion segment M with high inlier ratio regard to the measurement noise, if it is the first iteration of entering merge process, segment M will be put in a segment pool. In the subsequent merge process, 3D points in segment M will be combined with 3D points in one segment in the segment pool to test whether they should be merged. The testing rule for deciding whether merge or not is that the inlier ratio achieves the same threshold which is used to determine to run the split or merge process in section 3.3. By conducting a greedy method, if two segments align well, the one in the segment pool will be merged into segment M. Until all segments in the segment pool are tested with segment M, we will put segment M in the segment pool. In one merge iteration, the time complexity equals to O($p \times (s + s\prime) \times \log(m)$) where p is the number of segments in the segment pool, s is the number of points in segment M, $s\prime$ is the number of points in one segment in the segment pool, and m is the number of points in the previous frame. $p \times (s + s\prime)$ has a worst case $O(np)$ when s is larger than $s\prime$. Assume that there are total n points. And there are p segments in the segment pool.

If s is larger than $s\prime, \delta$ greater than $\frac{n}{p+1}$

$$p \times (s+s\prime) = p \times (\frac{n}{p+1} + \delta + \frac{n - \frac{n}{p+1} - \delta}{p}) = p \times [\frac{2np + \delta(p+1)(p-1)}{p(p+1)}] <$$

$$p \times [\frac{2np + n(p+1)(p-1)}{p(p+1)}] = \frac{np(2+p) - n}{p+1}) < \frac{np(2+p) - n}{p}) =$$

$$2n + np - \frac{n}{p} = O(np)$$

If s is the same as $s\prime$,

$$p \times (s+s\prime) = p \times (\frac{n}{p+1} + \frac{n - \frac{n}{p+1}}{p}) = p \times \frac{2n}{p+1} = 2n \times \frac{p}{p+1} < 2n \times \frac{p+1}{p+1} =$$

$$2n = O(n)$$

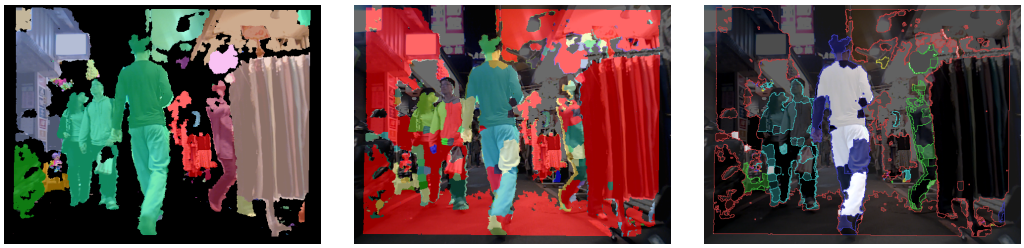
If s is smaller than $s\prime, \delta$ smaller than $\frac{n}{p+1}$

$$p \times (s+s\prime) = p \times (\frac{n}{p+1} + \delta + \frac{n - \frac{n}{p+1} - \delta}{p}) = p \times [\frac{2np + \delta(p+1)(p-1)}{p(p+1)}] <$$

$$p \times [\frac{2np + n(p-1)}{p(p+1)}] = \frac{3np - n}{p+1}) < \frac{3np - n}{p}) = 3n - \frac{n}{p} = O(n)$$

3.6. Result Integration

Since the result of the motion segmentation from sections 3.4 and 3.5 may contain too many kinds of colors, in order to let readers visualize the result more intuitively and comfortably, in this section, we integrate the result of motion segmentation with physical support relationships. Moreover, during the process, the intensity of each motion is recorded. By applying HSL color space, the intensity can be adjusted and displayed accordingly. HSL color space is one of the most common cylindrical-coordinate representations of points in a RGB color model. It represents the values of hue, saturation, and intensity in a cylinder color wheel. We demonstrate one of our result in Figure 3.10. In Figure 3.10(c), it is obvious that the upper body of the couple generate greater motion in contrast to their supporting legs. In addition, except showing joint motion segmentation result, we present three video



(a) physical support relationships (b) motion segmentation result (c) integration result

Figure 3.10. Integration of the segmentation result.

results with simple tracking. By sequentially matching one segment in the current frame and one in the previous frame, if they have the most points in common, they are determined correlated to each other between consecutive frames.

After segmenting motions in the last iteration, we combine each motion segment with the ground segment pairwisely. By using the same motion estimation in Section 3.3 to align the point cloud of the combined segment, we can obtain the inlier ratio of the grouped motion segment and ground segment. If the inlier ratio exceeds the same hard threshold mentioned in Section 3.3, we set the motion segment as part of ground segment. In addition, whether the integrated segments are belong to the ground segment is through the same testing processing.

CHAPTER 4

Experiments

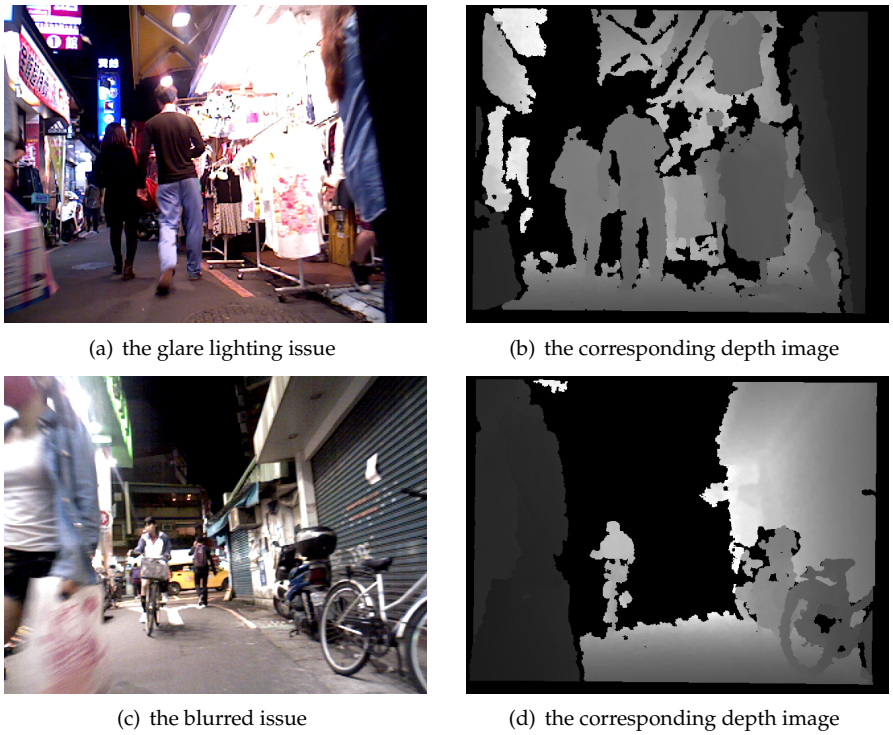
THE RGB-D video in the experiment was captured by a hand-held RGB-D camera at the night market near National Taiwan University. It is a challenging dataset with many pedestrians and arbitrary objects. Under this scenario, in the pre-processing part of the experiment, the number of blocks in the RGB-D over-segmentation is around 500. The distance criterion α which used for finding physical support relationships is set to 0.1 meter. In section 3.3, we pick four 3D correspondences for the SVD motion estimation and the range threshold, ε for determining one point is inlier or outlier is 0.015 meter. We set the hard threshold ratio, 90% to decide whether to conduct a merge procedure or a split process. In average, it takes about 0.05 second in one iteration to obtain the inlier ratio for comparison. In the split process, the parameter β and γ within the energy function 3.12 are 3.0 and 0.01 which means that given spatial coherence more influences than color can be adapted to this RGB-D dataset. Moreover, To compare our results with other's method, we implemented the work (Stückler and Behnke, 2013) with the same parameters like the number of blocks in the over-segmentation and the range threshold ε. Although we didn't adopt the over-segmentation data structure mentioned in (Stückler and Behnke, 2013), mainly it presented an efficient RGB-D motion segmentation by employing EM to infer motion assignment and estimation. Also, we had ran (Peter Ochs, 2013)'s and (Chiang, 2014)'s code test the same dataset. The comparison result is discussed in section 4.2. More details of the provided RGB-D dataset are shown in the subsequent section.

4.1. Setting of the RGB-D Camera and Crowded Urban Dataset

We utilize an ASUS Xtion Pro Live sensor because of its convenient and portable characteristic to collect data. The captured frame per second is around 25. The resolution of a RGB-D image is 640×480. The detected depth resolution is 100 um. The horizontal and vertical field of view for converting a RGB-D image to a point cloud are 1.02259993 and 0.79661566 respectively. Since the ASUS Xtion Pro Live is a structured light range sensor, to avoid natural light interference, we choose night time to collect outdoor data. We believe that in the near future, a RGB-D camera will be mounted on portable devices such as tablets or smart phones. Therefore, it motivates us to deal with an outdoor RGB-D video. Several issues need to be taken care during the night time in this crowded urban dataset. Due to the night time, in Figure 4.1(a), it shows that there is glare from signboards, fluorescent lamps or bulbs such as at the top right of the frame. It may have an influence to the performance of optical flow estimation. In spite of the degradation, the corresponding depth indicates that in the circled shining part there is a hanged T-shirt. Except for the lighting issue, a blurred image also will cause the performance drop of optical flow estimation while the depth of the blurred shopping bag is consistent. Despite the fact of lighting and blurred issue, in Figure 4.2, to investigate the depth accuracy in a more subtle way, as we can see, the depth of the helmet on the motorcycle is not continuous. At the beginning of the loop, the color illustrates the depth should be dark. However, the depth at the black cover of the helmet is grey which is contradict to the depth order. Though there exists depth inconsistency, not all parts of the motorcycle have the same drawback. On the contrary, from the proposed modified RANSAC paradigm and the smoothness energy function, the degradation mention in both color and depth issues will not influence our performance a lot and we can sustain persistent segmentation.

4.2. Results and Analysis

In this night market dataset, we clipped three videos for testing. The first test video consists of 146 frames with three major moving events. First, there is a woman with a shopping bag passing by shortly. The motion of the woman in the color image causes a blur effect. Then, behind the woman, there is a man moving in the same direction. Second, in front of us, a student rides her bicycle. After the bicycle event, a couple with similar motion is oncoming. The moving directions of the mentioned events are all relative to the



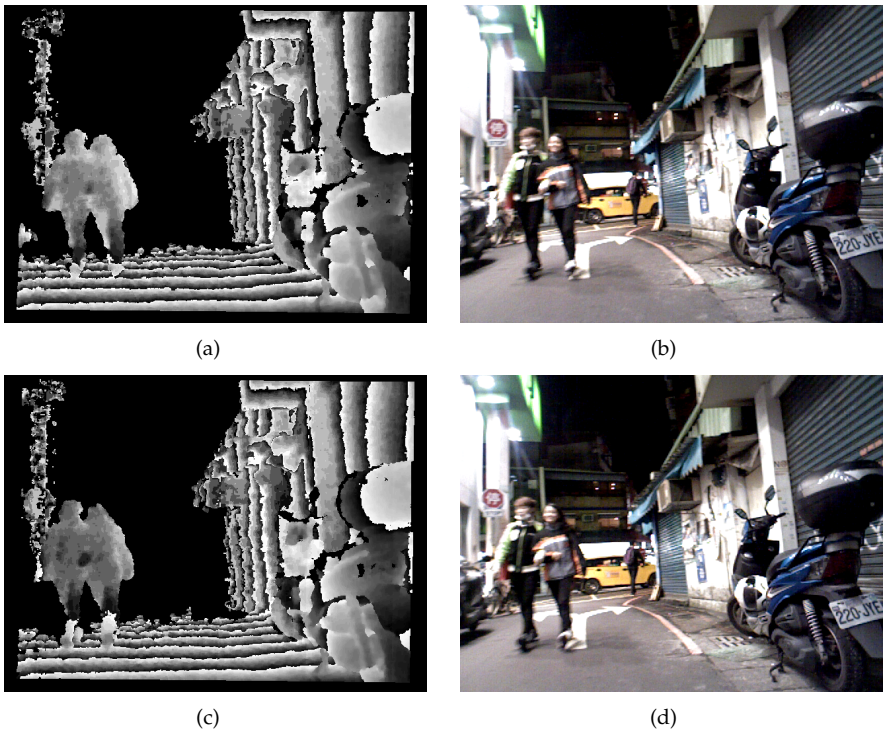
(a) the glare lighting issue (b) the corresponding depth image

(c) the blurred issue (d) the corresponding depth image

Figure 4.1. (a) and (c) shows the two issues of the color image in the dataset.

ego-motion. We select several motion segmentation and integration results of this video in Figure 4.3.

The second test video consists of 116 frames with one key event. There are five people involving in this video. A couple and a woman are moving in the relative direction on the left and right side of the scene. Then, the woman will be sheltered by the clothes stand. The key event is that the man in front of us will bypass a woman. When bypassing, the woman will stop and turn around to alter her moving path. In the first half of the video, the woman walks in the same direction as the camera motion. In the second half of the video, not only changing her heading direction, the woman gradually rotates her sagittal plane and faces about the camera. In Figure 4.4, we demonstrate some motion segmentation and integration results of this video. This video contains moving objects with rotation and translation with at most five moving objects. We choose this video for comparing our work with others.



(a) (b)

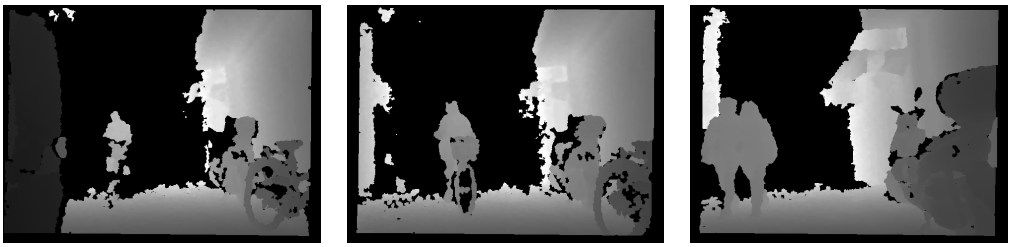
(c) (d)

Figure 4.2. (a) and (c) circularly display the depth measurement in 0.016 meter as one unit of gray scale from 0 to 255. (b) and (d) are the corresponding RGB image.

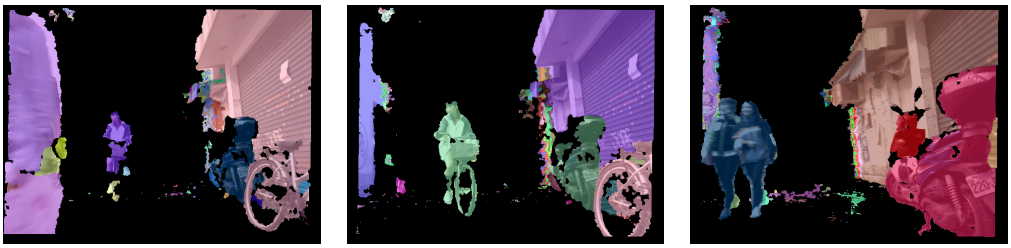
The third test video consists of 150 frames with one major event. In front of us, there is a man who hails a taxi and then the taxi slowly moves forward. When the man moves forward the taxi, he passes a parked scooter. Next to the man is a woman who stops for a short while and then turns around. A man who wears a helmet passes by the woman. About 6 meters away, there is a girl riding a bike far away from us. On the left side of the video, some people are waiting in the line. In Figure 4.5, we demonstrate some motion segmentation and integration results of this video.

For evaluating the accuracy, it is arduous to label different motion segments by hands only. Still, it is a research topic to create tools to help users annotate distinct motion segments. Therefore, we simplify the painful annotation work. For moving objects, we only label moving parts of those moving objects. In Figure 4.2, it shows the labelled ground truth under this policy. After carefully observing the whole video, the different color of segments represents different motion segments. The supporting legs of the pedestrians are considered to belong to background motion. To calculate the accuracy, we define a fraction

(a) the color images



(b) the depth images



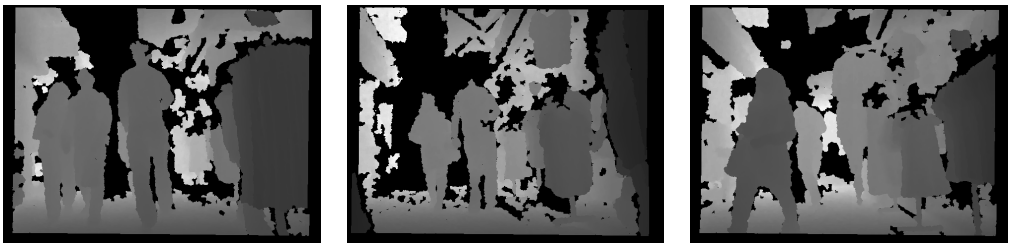
(c) the result of physical support relationships

(d) the motion segmentation result

(e) the integration result

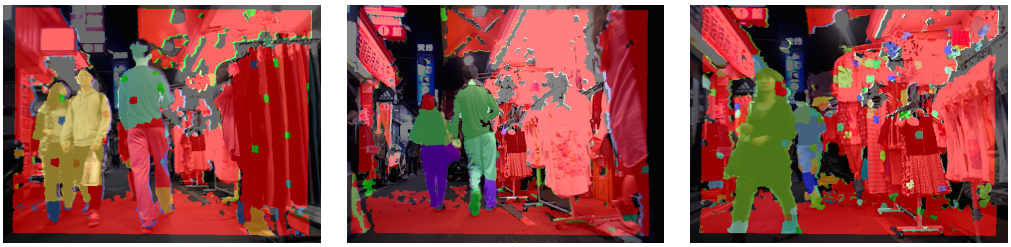
Figure 4.3. Three results of the first video. Some parts of the moving woman in the left side of the first column are segmented into background since the motion of those parts is too blurred to be deal with. The student rides a bike is well segmented. In the third column, the couple's legs are segmented in the background because they are supporting legs during the walking iteration.

(a) the color images

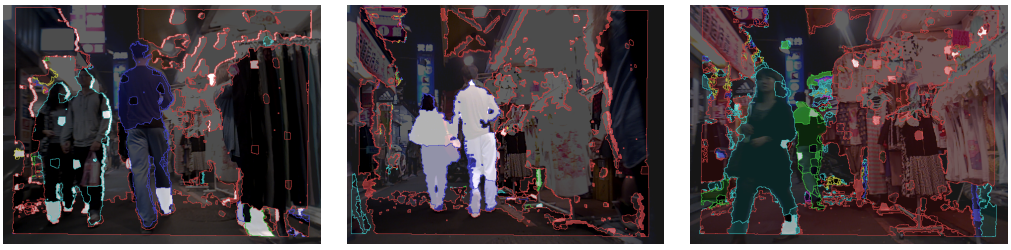


(b) the depth images

(c) the result of physical support relationships

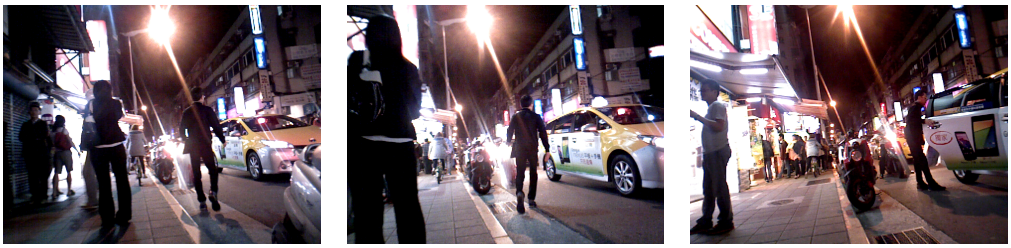


(d) the motion segmentation result

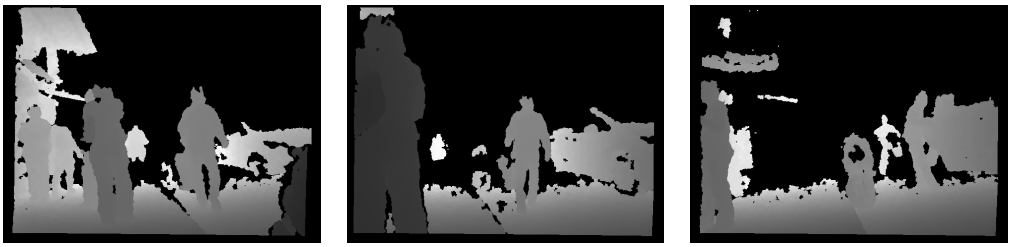


(e) the integration segmentation result

Figure 4.4. Three results of the second video. In the first column, some parts of the man in the middle are segmented into background because they are considered to be static. The left leg is the supporting leg; the result of the right leg is false positive. In the third column, the integration result has some small fragments because the depth of those fragments is not stable.



(a) the color images



(b) the depth images

(c) the result of physical support relationships

(d) the motion segmentation result

(e) the integration result

Figure 4.5. Three results of the third video. In the first column, the man who hails the taxi moves forward. The motion of the right leg of the man is obvious. A taxi which moves forward slowly is well segmented in the second column. In the third column, the motion segmentation result of the opening taxi's door is evident.

formula as

$$accuracy = \frac{max(C_i \cap L_j)}{M} \tag{4.1}$$

where C_i stands for the number of valid points in the computer motion segment, L_j stands for the number of valid points in the labelled ground truth segment and M means the number of valid points in the RGB-D image. $max(C_i, L_j)$ is the value of the number of valid points in the motion segment with the most overlapping points of the labelled ground truth segment. Since some parts in the RGB-D image are without depth information, points in these parts are invalid. Three cases elaborate the numerator term in Figure 4.7. In Figure 4.7(c), if there are multiple motion segments coloured in green and blue in a labelled ground truth segment, we assumed that they are belong to this ground truth segment because they do not share motion segments with other labelled ground truth segments. Another case in Figure 4.7(d) and 4.7(e) is that if there is a motion segment coloured in cyan belong to two different labelled ground truth, we only count the most overlapping points in one labelled ground truth segment. The other part of the motion segment is not counted. So the part in the cyan segment which overlaps to the red ground truth segment is added to the numerator term in Equation 4.1. While the other parts in the cyan segment which overlap to the green and yellow ground truth segment do not contribute to the numerator term. The evaluated accuracy is shown in Figure 4.9 with a description figure of several key frames and the average accuracy is at Table 4.1. To understand the evaluated accuracy figure, we list some insight of the events. There are four time slots between these key frames. As we can see in the evaluated accuracy figure, our approach outperforms than others in the first and second time slots. In these two time slots, there are at most five pedestrians. The motion of these pedestrians is complex with occlusion and dis-occlusion. Moreover, concerning to the environment, there is glare at the top left and right corner which will have some drawback to the color image. The woman in front of the man will be occluded and then appear in the left side of the man. In addition, a couple gradually dis-occludes by the man and moves toward us. On the other hand, there is a girl moving toward us who is gradually sheltered by a clothes stand. These effects caused by the girl and the couple lower the performance of (Chiang, 2014) and (Stückler and Behnke, 2013). For the performance of (Peter Ochs, 2013), mainly the drop is from the lighting issue. On the contrary, in the third and fourth time slot, the couple and the girl all disappear and the man tries to bypass the woman. When bypassing, the woman is not occluded. Then, the woman turns

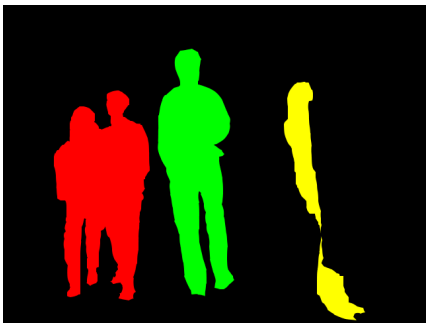
Figure 4.6. A demonstration of labelled ground truths

around and moves in a relative direction of both the man and us. These motions are much simpler than the previous ones with less occlusion and dis-occlusion. The glare in the top left corner is not as shining as in the first and second time slots. Therefore, the performance in average is good and the difference of the performance is not significant. While, at the end, there is a drop of (Peter Ochs, 2013) because the point trajectories of the man who is almost moving to the vanishing point of the image are difficult to track.

Table 4.1. Table of the average accuracy of the second video

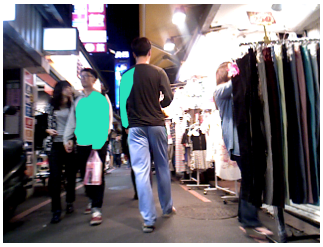
	Our	(Chiang, 2014)	(Peter Ochs, 2013)	(Stückler and Behnke, 2013)
accuracy	0.8726	0.8315	0.8181	0.8229

(a) the color image

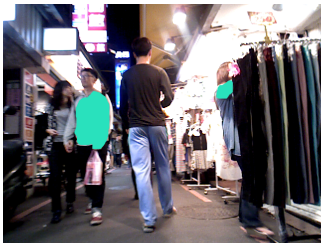


(b) the labelled ground truth

(c) one motion segment belongs to the same label

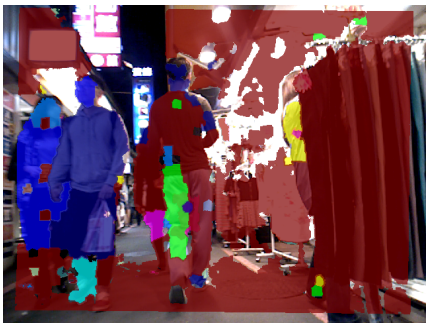


(d) one motion segment belongs to the different labels

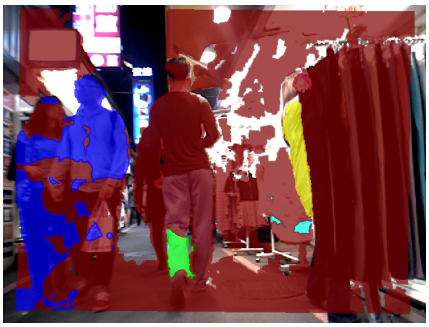


(e) one motion segment belongs to the different labels

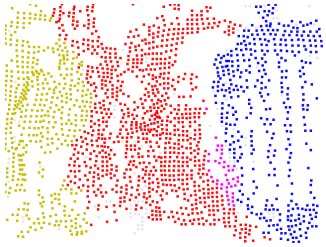
Figure 4.7. Three cases for elaboration of Equation 4.1



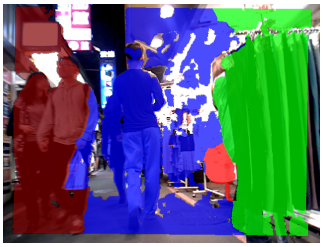
(a) our result at frame k



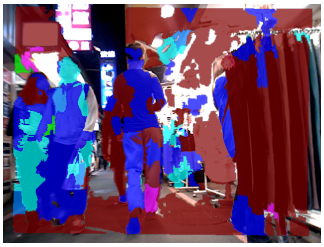
(b) one of results of (Chiang, 2014) at frame k



(c) one of semi-dense point trajectory results of (Peter Ochs, 2013) at frame k

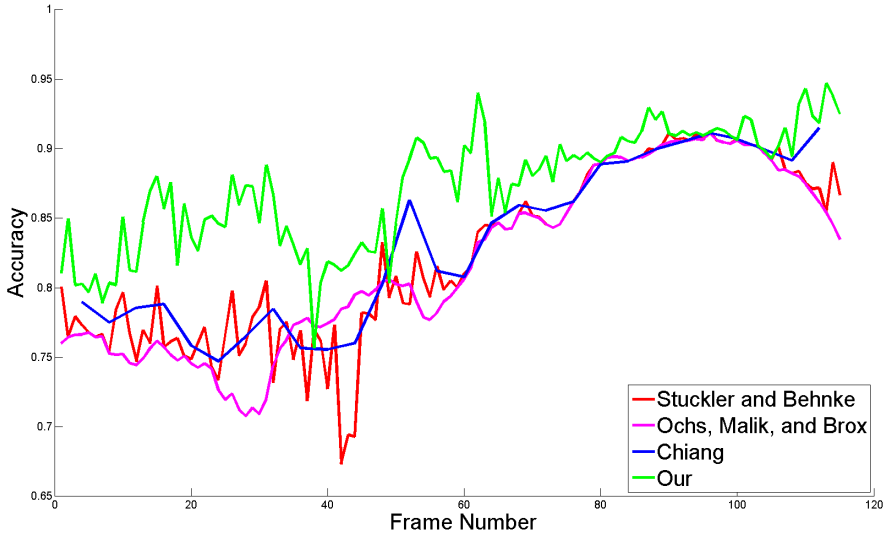


(d) one of results of (Peter Ochs, 2013) at frame k



(e) one of results of (Stückler and Behnke, 2013) at frame k

Figure 4.8. One of results in four approaches for calculating accuracy



(a)

(b)

Figure 4.9. (a) is the evaluated accuracy of the second video. (b) illustrates several key frames in the second video.

CHAPTER 5

Conclusion and Future Work

In conclusion, to solve the problem of motion segmentation with a RGB-D moving camera in a crowded urban scenario, we propose a modified RANSAC approach to robustly estimate motion parameters based on over-segmented blocks and conduct segmentation by the graph-cut optimization with an inlier and outlier distribution according to the estimated motion. In addition, an integration of the motion segmentation result and physical support relationships is introduced to represent the intensity of motion segments. At the end, we provide a challenging RGB-D dataset with labelled ground truth in a night market near National Taiwan University. We select part of the video for testing our approach. In the evaluation, we compare our approach with (Chiang, 2014), (Peter Ochs, 2013), and (Stückler and Behnke, 2013). We achieve better accuracy in average and discuss the performance result with the evaluated accuracy figure and a description figure with several key frames.

For the future work, we would like to use a traffic dataset captured by a HDL-32E 3D laser scanner. Since the density of the captured RGB-D image is much lower than one captured by an ASUS Xtion Pro sensor, we have to test our approach in this dataset and overcome this issue. Except for trying different dataset, we would like to label more ground truth and by learning annotating these motion segments, we would love to develop tools to ease the effort of annotation. For enhancing the performance of our approach, we would like to introduce more physical meaning like the variance of the displacement to the model for graph optimization to fully combine the RGB and 3D space with the motion space. Observing the alternative intensity of motion segments in one physical support, we can use

this information to predict possible motion and narrow the search space accordingly. Furthermore, by utilizing the physical support relationships, we want to understand and build semantic connections between motion segments to help solving other critical problems like SLAMMOT and Multi-agent localization.

BIBLIOGRAPHY

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Chiang, Y. (2014). A prioritized gauss-seidel method for dense correspondence estimation and motion segmentation in crowded urban areas with a moving depth camera. Master's thesis, National Taiwan University.

Costeira, J. P. and Kanade, T. (1998). A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179.

Hadfield, S. and Bowden, R. (2011). Kinecting the dots: Particle based scene flow from depth sensors. In *IEEE International Conference on Computer Vision*, pages 2290–2295. IEEE.

Herbst, E., Ren, X., and Fox, D. (2012). Object segmentation from motion with dense feature matching. In *IEEE International Conference on Robotics and Automation Workshop on Semantic Perception, Mapping and Exploration*.

Hickson, S., Birchfield, S., Essa, I., and Christensen, H. (2014). Efficient hierarchical graph-based segmentation of rgbd videos. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Kolmogorov, V. and Zabin, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Narayana, M., Hanson, A., and Learned-Miller, E. (2013). Coherent motion segmentation in moving camera videos using optical flow orientations. In *IEEE International Conference on Computer Vision*, pages 1577–1584. IEEE.

Perera, S. and Barnes, N. (2013). Maximal cliques based rigid body motion segmentation with a rgb-d camera. In *Asian Conference on Computer Vision*, pages 120–133. Springer.

Peter Ochs, Jitendra Malik, T. B. (2013). Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Schindler, K. (2005). Spatially consistent 3d motion segmentation. In *IEEE International Conference on Image Processing*, volume 3, pages III–409. IEEE.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.

Stückler, J. and Behnke, S. (2013). Efficient dense 3d rigid-body motion segmentation in rgb-d video. In *Proceedings of the British Machine Vision Conference*.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154.

Unger, M., Werlberger, M., Pock, T., and Bischof, H. (2012). Joint motion estimation and segmentation of complex scenes with label costs and occlusion modelling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1878–1885. IEEE.

Van den Bergh, M. and Van Gool, L. (2012). Real-time stereo and flow-based video segmentation with superpixels. In *IEEE Workshop on Applications of Computer Vision*, pages 89–96. IEEE.

Wang, C.-C., Thorpe, C., Thrun, S., Hebert, M., and Durrant-Whyte, H. (2007). Simultaneous localization, mapping and moving object tracking. *International Journal of Robotics Research*, 26(9):889–916.

Wang, Y. and Huang, S. (2013). An efficient motion segmentation algorithm for multibody rgb-d slam. In *Proceedings of Australasian Conference on Robotics and Automation*.

Werlberger, M., Pock, T., and Bischof, H. (2010). Motion estimation with non-local total variation regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA.

Yan, J. and Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, pages 94–106. Springer.

Yang, S.-W., Wang, C.-C., and Chang, C.-H. (2010). Ransac matching: Simultaneous registration and segmentation. In *IEEE International Conference on Robotics and Automation*, pages 1905–1912. IEEE.

Document Log:

Manuscript Version 1.0 — 16 June 2014

Typeset by $\mathcal{AMS}$-LaTeX — 14 November 2014

CHING-HSIANG HSU

THE ROBOT PERCEPTION AND LEARNING LAB., DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION ENGINEERING, NATIONAL TAIWAN UNIVERSITY, NO.1, SEC. 4, ROOSEVELT RD., DA-AN DISTRICT, TAIPEI CITY, 106, TAIWAN, *Tel.* : (+886) 2-3366-4888 EXT.407

E-mail address: r01944021@ntu.edu.tw