國立交通大學

資訊科學與工程研究所

博 士 論 文

以多攝影機進行人物定位

People Localization Using Multiple Cameras

研 究 生：羅國華

指導教授：莊仁輝　博士

陳華總　博士

中 華 民 國 一〇二 年 二 月

國立交通大學

資訊科學與工程研究所

博 士 論 文

以多攝影機進行人物定位

People Localization Using Multiple Cameras

研 究 生：羅國華

指導教授：莊仁輝　博士

陳華總　博士

中 華 民 國 一〇二 年 二 月

以多攝影機進行人物定位
People Localization Using Multiple Cameras

研 究 生：羅國華　　Student： Kuo-Hua Lo

指導教授：莊仁輝　　Advisor： Jen-Hui Chuang

陳華總　　Hua-Tsung Chen

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

February 1, 2013

Hsinchu, Taiwan, Republic of China

中華民國 一〇二 年二月

以多攝影機進行人物定位

研究生：羅國華　　指導教授: 莊仁輝
陳華總

國立交通大學資訊科學與工程研究所

摘要

在以視覺為基礎的人物定位與追蹤的研究中，人物遮掩是一個重要且具挑戰性的研究課題。為了處理這樣的問題在本博士論文中，我們提出數個以多攝影機進行人物定位的方法。先前被提出的方法是藉由將多個視角影像中的前景資訊投影至多參考平面來確認空間中不同高度的參考平面上是否有人物存在，因此比起僅使用單一參考平面，將能夠有效地處理人物遮掩之問題，然而這將使得計算量隨著參考平面與使用的畫面數量而大幅增加。為了減低上述投影所需之計算，我們提出了第一個方法：基於線段取樣式定位法。此方法可利用影像中垂直於地面直線的消失點，估計出人物的成樣本線段，如此一來，在各高度參考平面上的人物定位將僅需計算線段的交點來重建出人物的位置，而能夠大量地減少先前的作法中需將前景資訊投影於多重平面的計算量。接著我們對這些交點進行分析後，將不同平面的交點進行連線即可形成三維樣本線段。這些樣本線段經過品質的評估，並淘汰掉不合適的軸線後，依據分群的演算法被分為數群，再依照各群內的三維軸線整合的結果推算出人物的位置。

然而由於上述的方法在重建時仍需要較多的時間，為了更進一步地改善其效率，我們提出了第二種非重建型的人物定位方法。此方法不需要將所有的前景資訊投影到多重平面上，而是先初步地以足跡分析估計出人物的潛在位置，再產生三維樣本線段來確認人物所在的位置。這樣一來不僅改善了我們的計算速度，同時也可將人物的高度在計算的過程中估計出來。另外，我們也針對第一種方法進行改良，提出第三種人物定位方法。其主要的兩項改良為: (1)新的兩個垂直三角形的相交重建方式與微調步驟來找出人物可能的三維樣本線段，(2)新增兩項與頭部高度有關的幾何過濾規則，用來過濾這些三維樣本線段。兩者皆能夠改進定位正確性，包含了精確率與查全率(precision and recall) ，而(2)則能提升計算的效率。此外，我們還提出了一個具有視角不變之特性的線段對應性的測量方法，能以量化方式測量不同視角影像中任意線段之對應性。我們更進一步地將其應用於人物定位方法之上，不但改善了效率

而且並未減低其定位的正確性。最後我們探討了利用樣本線段之間的對應性以及兩個視角之間的角度，來進一步地降低人物定位誤差的可能性。

關鍵字: 消失點、二維/三維樣本線段取樣、多攝影機、人物定位、即時

People Localization Using Multiple Cameras

Student: Kuo-Hua Lo

Advisor: Jen-Hui Chung
Hua-Tsung Chen

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

Occlusion has been an important and challenging task in vision-based people localization and tracking. To handle this problem, we propose several people localization methods in this thesis, which are based on multiple cameras. Some existing methods have been proposed to check the existence of people at reference planes of different heights by projecting image foreground from multiple views to these planes; such approaches can deal with occlusions better than using only a single reference plane. In order to reduce the amount of calculation due to image projection, especially for a large number of reference planes and camera views, we first propose a sample line-based method. The method estimates 2D line samples, which are originated from the vanishing point of lines perpendicular to ground plane, for each person in different images and project these 2D line samples on reference planes to reconstruct people locations so that the computation of previous work can be greatly reduced. For the subsequent localization process, these intersection points are analyzed and integrated to form some 3D line samples, and these 3D line samples are then grouped and integrated to reconstruct the locations of people in the scene.

Because the above method still takes a lot computation during the reconstruction of 3D line sample, we propose the second method which is not based on reconstruction by projecting all foreground pixels to multiple reference planes. In particular, a footstep analysis is developed to find potential people locations, and 3D line samples are then generated to identify people locations. This method results in significant improvement in computational efficiency, with people heights being estimated as by-product. We proposed another method to improve the performance of the first method with (i) new reconstruction from the intersection of two vertical triangles and refinement procedures for possible 3D (vertical) line samples of human body and (ii) addition of two new geometric rules (associated with the head level of a person) for the screening of these samples. While (i) reconstructs a 3D line sample directly (and efficiently). Both of them offer valuable improvements in the localization performance, in terms of precision and recall, with (ii) also saving some computation time spent for invalid samples. In addition, we also propose a correspondence a view-invariant measure of 2D line segments in two different views. Such a quantitative measure can handle line segment of arbitrary configuration in the 3D scene. By

applying such a measure, efficiency of people localization is further improved without sacrificing the localization correctness. Finally, possibilities of using the correspondence of line samples and the difference between a pair of viewing angles to decrease the error of people localization as studied, with some promising results obtained.

Keywords: Vanishing point, 2D/3D line sampling, multi-camera, people localization, real-time

Contents

List of Figures

List of Tables

Chapter 1
Introduction

In recent years, visual surveillance using multiple cameras has attracted much attention in the computer vision community. Moreover, vision-based localization and tracking have shifted from monocular approaches to multi-camera approaches since the latter can often achieve better results. Especially when there are many people in the scene, serious occlusions may occur in multiple views and real-time people tracking and localization become a challenging problem. Thus, the previous works on visual surveillance are reviewed in the following in two categories: monocular approaches and multi-camera approaches.

1.1 Monocular approaches

In [1], [2], location and intensity of image foreground are extracted to allow construction of a human model, which allows us to match a subject image for tracking in successive grayscale images. In [3], color information is used to construct human models, wherein a person is modeled by several parts of similar color, and a Bayesian framework is employed to handle occlusion in the tracking process. In [4], an extension of particle filter using object contour is proposed to track the head of a person. In [5], a color-based tracking which integrates color distributions into particle filter is presented to describe people using ellipses and associated color histograms. The method is robust when dealing with partial occlusion, and is rotation and scale invariant. In [6] color, shape, and edge are integrated into particle filter to create a robust tracking method. Additionally, the authors propose an adaptive scheme to choose the most effective cues in different situations. However the performance of these methods might be seriously impaired when the human model of occluded persons is not updated in time that the appearance of a person may change significantly. To resolve such a problem, spatial/temporal features are used in [7] to train convolutional neural networks to achieve robust people tracking wherein the appearances of a target object of different views are adopted in the training stage.

Since single view tracking depends on inherently limited information from a single viewing angle, dealing with situations involving serious or full occlusions is quite difficult. Thus, many multi-view tracking approaches have been proposed. Unlike single view, multiple views can provide more visual information to cope with occlusions in human localization. For example, a stereo camera with small baseline can estimate depth information easily, whereas a set of wide-baseline cameras can decrease invisible regions. Finding feature correspondence is usually the most important step for many multi-camera approaches since only correct correspondences

between multiple cameras can ensure the correctness of subsequent processes, e.g., localization and tracking.

1.2 Multi-camera approaches

There are several types of multiple camera approaches for tracking people. The first type of approaches uses a stereo camera to obtain depth maps for tracking. The second type of approaches can be divided further into two sub-categories, region-based and point-based methods, both have to establish correspondence between different views for tacking. The third type of approaches seeks to find locations of persons directly without the correspondences of people in different views.

For the first type of approaches such as [8–10], a stereo camera is exploited to establish correspondence between two views to construct a depth map. By using such a map to avoid influences of moving shadows on foreground detection, better segmentation results can be obtained and object tracking becomes more robust. However, using a pair of cameras with a small baseline may suffer from total occlusions frequently. Without information of occluded regions (e.g., behind of a person closer to a stereo camera), the tracking performance is impaired.

Region-based methods of the second type generally regard people as regions and use region features to match people in multiple views. Most of these methods use color as the main feature to find correspondences of regions in different views. For instance, color and 3D position are utilized to match and track multiple objects by a tracking algorithm in [11]. In [12], the authors use Gaussian color models to segment foreground regions of people from each image. The results are then used to match regions from one view to another along epipolar lines to find correspondence across multiple views. After that, Kalman filters are used to track people on the ground plane. In [13], the authors use Bayesian networks for object tracking in individual views independently. After that, both geometry-based (epipolar geometry, homographies, and landmarks) and recognition-based (height and color of target appearance), are utilized to find correspondence across multiple views. However, one of the main disadvantages of these methods is that color information may degrade the performance of tracking since the appearance and color can change with scene illumination.

Point-based methods can be further divided into two additional sub-categories: 3D-based and 2D-based methods. 3D-based methods locate and find correspondence of target object in images based on 3D geometric constraints. These 3D-based methods often need a complete camera calibration. In [14], location of a person is described by a Gaussian distribution of its center of gravity (COG) in the scene. The distribution, which denotes the probability of the existence of a

COG point, is projected onto multiple views, respectively, and the correspondence of feature point can be found by maximizing the probability of the COG distribution in each view. In [15], people are modeled as vertical cylinders and tracked by optical flow. During the tracking process, the COG of human body in multiple views is used to estimate the people locations in the world coordinate. In [16], cameras are calibrated for the calculation of 3D positions of feet points of target people, and the correspondences can be established from these feet points. In [17], feature points are extracted from a (vertical) major line of the upper part of a human body. The correspondence of the human body is found by matching intensity and location through epipolar constraints. However, the extracted feature points from each view may not always correspond to the same point in the 3D space. In that case, the matching performance, the established correspondence, and tracking results may be impaired.

Different from the above 3D-based methods, some 2D-based methods has been presented to establish correspondences between multiple cameras by matching locations of feature points on a reference plane. In [18–20], homography constraint is used to match the locations of feet points in different views. However, these feature points may be occluded between objects. In [20], a method, which can detect whether the feet points of a person are occluded, is proposed to select a best view for each person appears in the scene. In contrast, authors in [21] propose a method using the axes of people to estimate the feet points in images. They segment a group of people into individual persons and estimate an axis for each of them. Then, the location of the feet point of a person is estimated as intersection point of his/her axis and the bottom of his/her bounding box. In [22], foregrounds of a person are perspectively projected from each view to the ground plane, with the corresponding camera being the projection center. For each camera, a line passing through (i) the projected foreground and (ii) the vertically projected camera center, both on the ground plane, is estimated. The person's location can then be estimated by calculating the intersection of these estimated lines on the ground plane based on the least square criterion. For most of the aforementioned point-based approaches, accurate detection/estimation of point/line features, and their correspondences in different views, are required; otherwise the correctness of a person's location will be seriously impaired.

In recent years, approaches of the third type are proposed. These methods, which do not need a complete camera calibration, can locate people directly without finding the correspondences of the people between views. In [23-24], the authors propose a method using cameras placed at high elevation to detect the heads of people. The method assumes the cameras are partially calibrated for homographic matrices for multiple planes with different heights. For each plane, intensity information of segmented foreground pixels is collected from all views, and head detection is

achieved through intensity correlation. In [25], the authors propose an interesting method to track people by locating them on similar reference planes. The foreground likelihood information of all image pixels captured from different views is projected and integrated on each reference plane to form an occupancy probability. Such probabilities from several frames are then processed by a graph cut algorithm to find trajectories of people. Although the correspondences of people between different views are not available[1], such an approach performs quite well under serious occlusions in a crowed scene. Due to the high complexity of pixel-based processing, the approach is implemented with CUDA (Nvidia GeForce 7300 GPU) to achieve real-time performance.

Unlike the above method that need to project all foreground pixels of all views to multiple reference planes via homography, we propose three efficient and effective people localization methods. The first one applies vanishing point-based line sampling to reduce the large amount of pixel processing so that computational efficiency can be greatly enhanced. The second one further improves efficiency and robustness of the first one by adopting a more accurate 3D reconstruction process, more effective geometric filtering rules, and a novel measure of line correspondence. Instead of 3D reconstruction, the third method uses a coarse-to-fine strategy to find people locations by 3D line sampling. Finally, error analysis is considered for further improvement of the accuracy of people localization for the second method.

1.3 Organization of the thesis

The remainder of this thesis is organized as follows. In Chapter 2, people localization via vanishing points of vertical lines and multiple homographic matrices is proposed. The vanishing points are used to generate 2D line samples of foreground regions in multiple views. Potential people locations are found by project each pair of 2D line samples from different views to the reference planes of different heights via homographic matrices. The intersection points are then connected to form 3D line samples. After that, the 3D line samples are checked against foreground regions of all views and grouped to locate people. Instead of reconstruction in the 3D space, we propose a grid-based approach to efficiently find potential people locations on the ground in Chapter 3. We then generate 3D sample lines for these potential people locations, refine their two ends, and remove those not covered by enough foreground pixels in all views. Additionally, people heights are estimated from the 3D line samples as by-products. In Chapter 4, a more efficient reconstruction method is proposed to improve the people localization approach described in Chapter 2, where reconstruction of 3D line samples takes a lot of computation time to project

[1] For example, no additional image processing procedures are performed to identify each individual from a crowd, e.g., through connected component analysis and principal axis analysis as adopted in [21].

2D line samples to multiple reference planes, a more efficient reconstruction approach which reconstructs a 3D line sample as the intersection of two vertical triangles is proposed. In addition, a pre-filtering procedure using a view-invariant measure of line correspondence is also introduced to further improve the efficiency. In Chapter 5, we first review an error analysis method for a pointing system. The idea is then extended and applied to our people localization method described in Chapter 3 to increase the accuracy of localization. Chapter 6 summarizes this thesis.

Chapter 2
Vanishing point-based line sampling for efficient people localization

In this chapter, vanishing point-based line sampling is introduced to increase computation speed of people localization. The vanishing points of vertical lines in the scene in images captured from different viewing angles are used to generate 2D line samples of foreground regions. Subsequently, 3D line samples of persons can be found efficiently via 3D reconstruction from stereo 2D line sample pairs to avoid pixel-based operations suggested in [23-25].

2.1 Construction of major axes for non-occluded persons from a pair of views

For a better understanding of the basic ideas of the proposed localization, we begin by illustrating how to localize people using the major axes (MA) of the foreground regions in 2D images. Assume the foreground of different persons do not overlap in a pair of views in which the major axis of each of them can be estimated correctly. By projecting these axes, instead of projecting all foreground pixels as in [25], onto multiple reference planes parallel to the ground plane, a 3D axis can be formed for each person by connecting corresponding intersection points of the projected 2D axes on these reference planes. Furthermore, a more efficient scheme is introduced to find the above 3D axis by calculating the intersection line segment of two triangles in the 3D space if the cameras centers can be estimated in advance.

2.1.1 Major axis estimation for a person in an image

In order to segment foreground regions of a person from an image, the Gaussian mixture model (GMM) [27], [28] can be applied. Assume region R obtained from foreground segmentation contains a great percentage of a person, we can estimate the major axis for the person by PCA. An example of an axis thus estimated is shown in Fig. 2.1. One can see that the estimated major axis can represent the elongated shape of a person very well.

2.1.2 Finding a 3D major axis of a person – two approaches

As shown in Fig. 2.2, Let L_1 and L_2 be the axes of a person obtained by PCA for View 1 and View 2, respectively. In addition, let P_{12}^{π} be the intersection point of the two lines containing the projections of L_1 and L_2, respectively, onto reference (ground) plane π from camera centers C_1 and

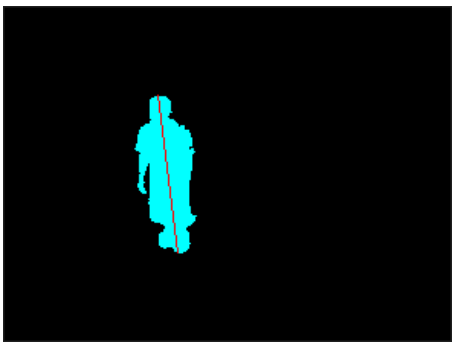


Fig. 2.1. Detected foreground regions and the estimated axis.

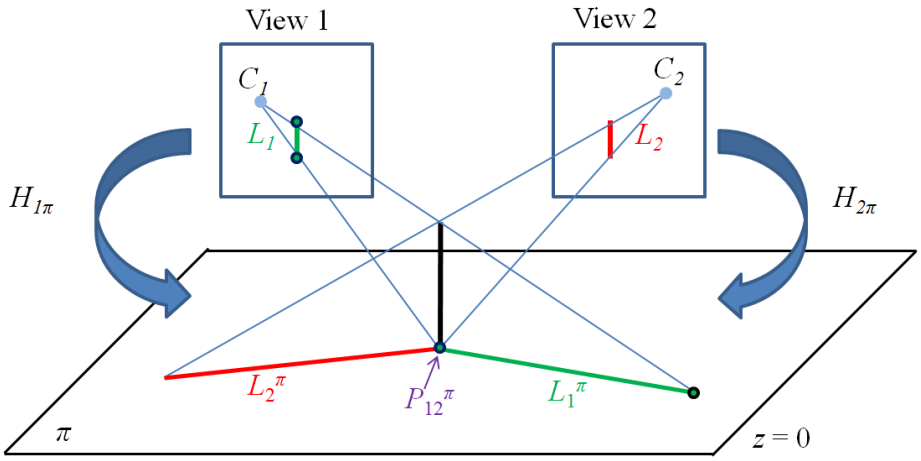


Fig. 2.2. Finding intersection points of two axes on a reference plane.

C_2. Ideally, for reference planes of different heights, such intersection points will either (i) belong to both the projected axes, or (ii) stay away from any of them if the corresponding heights are out of the range of the 3D axis. Fig. 2.3 shows samples of the 3D axis thus obtained for the person shown in Fig. 2.1. While intersection points satisfying (i) is colored in black, points not satisfying (i), including those contained in one but not both projected axes due to computation errors, are marked in red[2].

The above results provide us an important cue to the estimation of a person's height. Additionally, one can see that the 2D (horizontal) positions of these 3D points are quite consistent that a roughly vertical major axis (MA) of the person can be constructed by connecting the black points, i.e.,

$$Axis\_set_{1,2}^{h_b h_t} = \left\{ P_{1,2}^{h_b}, ..., P_{1,2}^{h_t} \right\} \tag{2.1}$$

with h_b and b_t being the heights of bottom and top end points of the axis, respectively.

[2] To find the above intersection points on reference planes of different heights, a method to produce multiple homographic matrices is introduced which can establish these matrices using only two marker points on each of the four calibrating pillars standing vertically on the ground plane. The detail can be found in Appendix A.

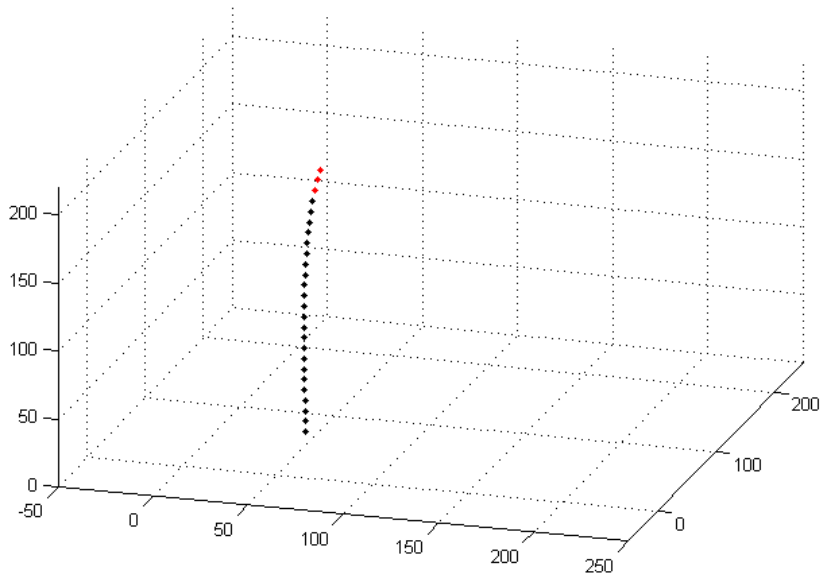


Fig. 2.3. The axis samples of the person shown in Fig. 2.1, which are reconstructed for reference (horizontal) planes with 4 cm spacing and up to 176cm in height.

2.1.3 Extension of finding 3D major axes for non-occluded multiple persons from a pair of views

The above method can be extended to estimate 3D MAs for multiple people if an axis can be found for each of them in two different views. Without knowing the correspondence of the axes in the two views, candidate 3D MAs can be constructed for all possible 2D MA pairs. For example, for *M* persons in View 1 and *N* persons in View 2, a total of *MN* candidate MAs can be constructed (minus those associated with triangle pairs which do not intersect, like the two blue triangles shown in Fig. 2.4).

For a candidate 3D MA obtained for person *i* in View 1 and person *j* in View 2, (1) can be rewritten as

$$Axis\_set_{1i,2j}^{h_b h_t} = \left\{ P_{1i,2j}^{h_b}, ..., P_{1i,2j}^{h_t} \right\} \tag{2.2}$$

Although we do not have correspondences of different people in these two views, it is possible to remove incorrect 3D MAs by checking the consistency in the foreground coverage, as will be explained in Subsection 2.2.1, with additional views. For example, while the two green axes in Fig. 2.4 are correct 3D MAs, the gray axis can be identified as an invalid axis from View 3[3].

[3] In general, incorrect MAs constructed from a pair of triangles can be removed by checking the consistency with an additional view point (in the 3D space) except for those view points which are coplanar (in a 2D subspace) with one of the two triangles mentioned above. Therefore, with the help of an additional camera, incorrect MAs will be removed completely, with zero probability for the above exceptions.

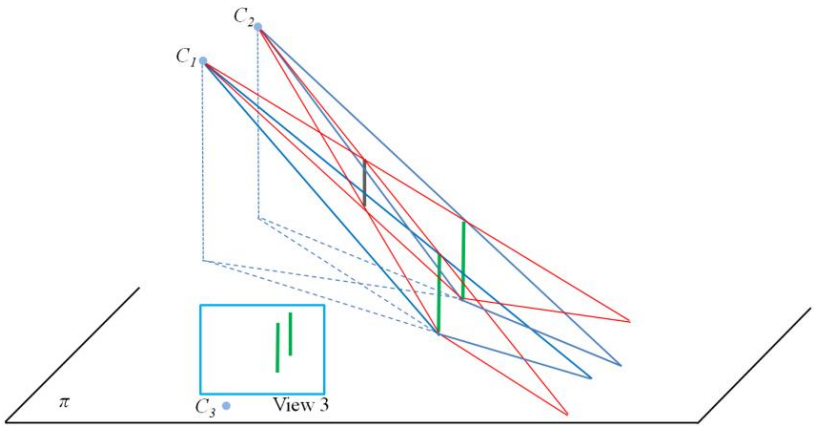
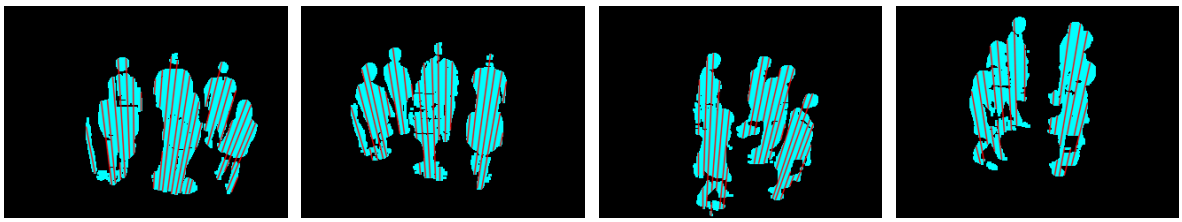


Fig. 2.4. Illustration of filtering out incorrect 3D MAs by using an extra view.

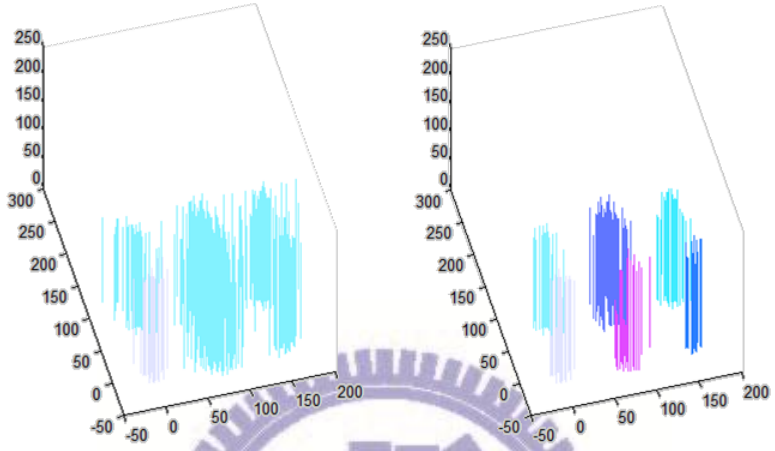
Fig. 2.5. An example of overlap foreground and the estimated axis.

2.2 Construction of major axes for multiple persons with occlusion

The above 2D PCA-based axis estimation can only cope with situations under which the foreground of a person is separable from others' in *all views*, and can be identified as one region by connected component analysis. However, in real applications, many people may appear in a monitored scene at the same time that each segmented foreground area may contain more than one person, as shown in Fig. 2.5, and the aforementioned axes detection approach will not work correctly. One possible solution proposed in [21] is to separate persons by projecting the foreground in the vertical direction to form a histogram, and then determining the boundaries between persons based on the location of peaks and valleys in the histogram, before each person can be represented by one axis for localization and tracking. However, the above approach may not work well when there is a very dense group of people appear in the scene, e.g., for the case shown in Fig. 2.6. For such more complicated situations, instead of estimating a 2D axis for each person, a 3D sampling scheme is proposed in this section wherein 2D line samples of the foreground regions from multiple views are used to generate some 3D line samples of the



(a) (b) (c) (d)



(e) (f)

Fig. 2.6. (a)-(d) 2D line samples in Views 1-4. (e) The unverified 3D line samples which survive Rules 1-2. (f) The results of filtering and grouping.

foreground "volume" based on the same idea described in Section 2.1. Then, with noises filtered out, these 3D line samples are verified with respect to different views by a back projection procedure. Finally, a grouping algorithm is applied to the remaining samples in the scene, before members of each group are integrated into a 3D MA.

2.2.1 Generating 3D line samples using vanishing points

Since the upper bodies of people are almost always perpendicular to the ground plane when they are standing and walking in a monitored scene, we first generate 2D line samples in each view which are originated from the vanishing point of vertical lines in the 3D scene (see Figs. 2.6(a)-(d))[4]. Thus, these 2D line samples correspond to a fan of vertical sampling slices in the 3D space originated from the vertical line containing the corresponding camera center. Note that generating 2D line samples is much faster than the axis estimation discussed in Section 2.1 since no additional image processing is required. The 2D sampling lines having very short lengths (less

[4] The vanishing point in each view can be estimated by calculating the intersection points of the four lines extended from the four upright pillars mentioned in Subsection 2.1.2.

than a threshold T_p) will be discarded since they are expected to be far away from a major axis and will have little contribution to the estimation of a 3D MA.

Next, for each pair of views, the remaining 2D line samples are used to reconstruct 3D line samples by the scheme described in Section 2.1. Since there may still be incorrect 3D line samples, such as the gray one shown in Fig. 2.4, two geometric rules can be used to filter out the 3D line samples that will not correctly represent a person in the 3D scene:

1) The length of a 3D line sample is shorter than T_{len},

2) The height of its bottom end point P^{hb} is higher than T_b.

Fig. 2.6(e) shows 3D line samples passed the two rules, each adjusted slightly so that it is perpendicular to ground plane.

After using the above two filtering rules, we further verify the 3D line samples against image foreground. To check the foreground coverage of a 3D line sample, we back-project its intersection points of different heights to all image views. For a person do appear in the monitored scene, these back-projected points should be covered by some foreground regions. For example, if all back-projected points in all views for a 3D MA are of foreground, its average foreground coverage rate (AFCR) is equal to 100%. A 3D line sample with AFCR lower than T_{fg} will be removed. Fig. 2.6(f) illustrates the filtering results for line samples shown in Fig. 2.6(e).

2.2.2 Integration of 3D line samples to form 3D major axes

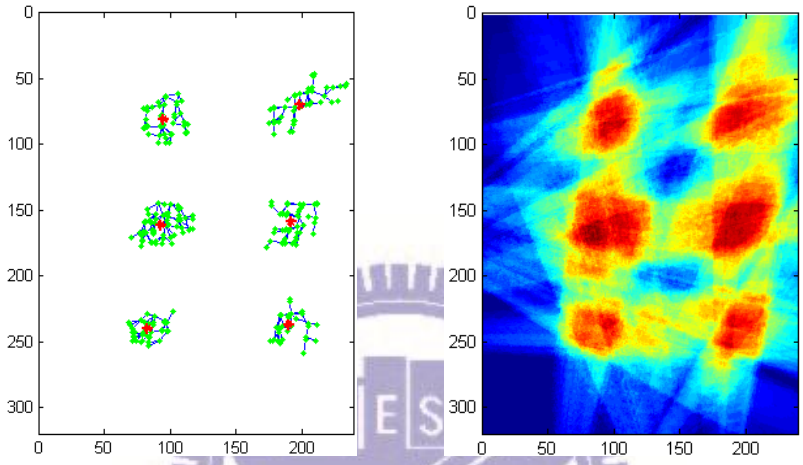
After the above verification procedure, the major axis of a person can be estimated from the remaining 3D line samples using a straightforward grouping algorithm[5]. Specifically, if the 2D horizontal distance between two 3D line samples is closer than a threshold T_c, an edge is established in an undirected graph. After that, we can easily find connecting component areas (3D line sample groups) in the graph. For example, Fig. 2.7(a) shows the input frame for Fig. 2.6(d), and Fig. 2.7(b) shows the undirected graph obtained by the above grouping algorithm, with green points representing the 3D line samples. To avoid some false positives in the grouping, a group containing a total number of 3D line samples less than threshold N_{line} will be removed.

To locate individual persons, the horizontal position of each of them can be estimated as the average, shown as red stars in Fig. 2.7(b), of the horizontal positions of the 3D line samples in the corresponding group[6]. In Fig. 2.7(c) we show the synergy map obtained with a method modified from [25]. Instead of considering the foreground probability of all image pixels, only those inside of foreground regions are taken into account. One can see the above distribution of each group

[5] Detail can be found in [46].

[6] The heights of the top and bottom ends of a 3D major axis are assigned as the heights of the highest and lowest end points in the corresponding group, respectively.

(a)



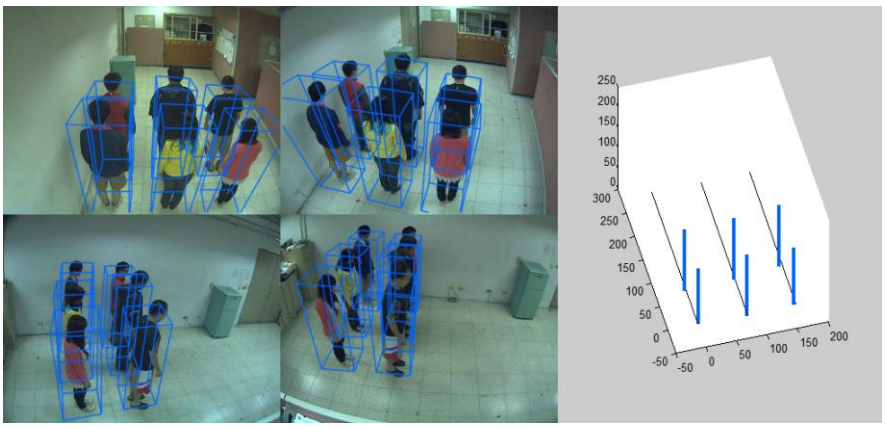
(b) (c)

Fig. 2.7. Grouping and localization results. (a) Input frame 532. (b) Grouping sets. (c) Accumulated synergy map of all reference planes.

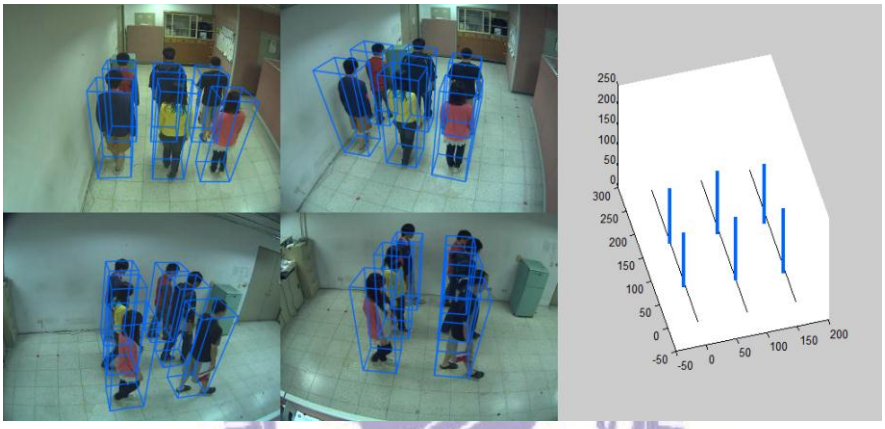
matches the corresponding occupied region (red color) in the map quite well, i.e., all red stars do fall inside of the occupied regions.

2.3 Experiments

In order to evaluate our method, we used an indoor video with a resolution of 320 × 240. The spacing between 51 adjacent reference planes was selected as 4cm. In the video, six people are walking along three edges of the tiles on the ground so we can easily evaluate the performance of localization. In Figs. 2.8(a) and (b), the bounding boxes with a fixed cross-section of 50cm x 50cm are back-projected to individual images with their height obtained from derived 3D MAs, shown on the right of the figures with bold lines. One can see that the six persons are well represented with these bounding boxes, and their locations having good matches with the specified tracks. For a comparison of computation time with [25], simulation is performed with an implementation based on C language on Windows 7 with, 4 GB RAM and a 2.4G Intel Core2 Duo CPU. Fig. 2.9(a) shows the processing speed, in frame rate per second (FPS), of our method for different portions of the video, with intervals A to F

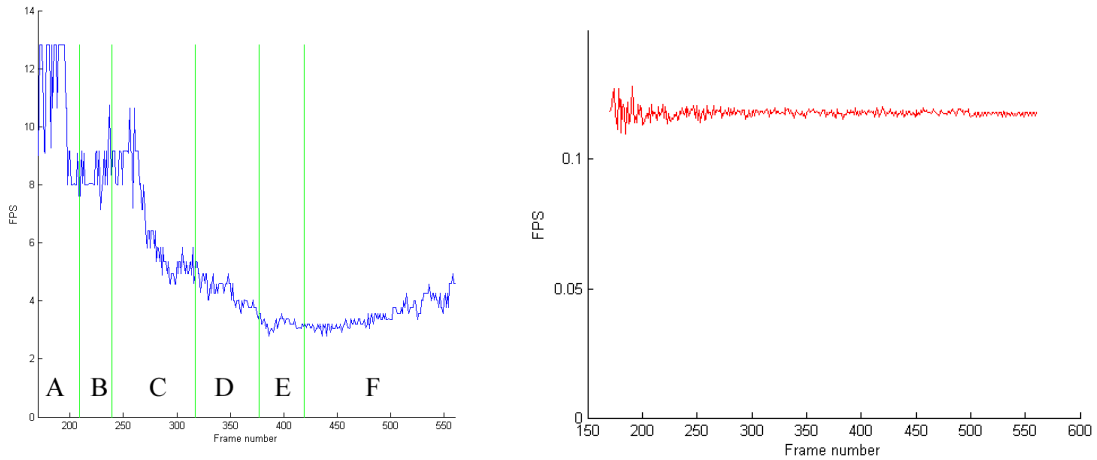


(a)



(b)

Fig. 2.8. Localization results for frame 475 and 540.



(a) (b)

Fig. 2.9. Processing speed (in frame rate per second) of (a) Our method. (b) The generation of accumulated synergy map from all reference planes.

corresponding to an increase from 1 to 6 persons in the scene, respectively. One can see that the processing speed varies with people count and more than 2.790 FPS can be achieved when there are six people in the scene. The average is 5.365 FPS. Fig. Fig. 2.9(b) shows the FPS required for the generation of synergy maps, as proposed in [25], which varies much less with time and has an average value of 0.118 FPS. (Note that CUDA adopted in [25] is not used here). This is because its time complexity mainly depends on the size of the whole image but not just the foreground.

2.4 Summary

We proposed a method for people localization which obtains 2D line samples, with each line originated from the vanishing point of vertical lines in the scene, of foreground regions in each view. Geometrically, a pair of line samples obtained from two different views corresponds to a vertical line in the scene. 3D point samples along such a vertical line can then be obtained by projecting the above 2D line samples and identifying their intersecting point on reference planes of different heights, using homographic matrices each associating an image to a reference plane. Finally, the 3D MA of each person is estimated by grouping 3D line segments derived from point samples satisfying some location and shape constraints. Since the most time-consuming process of homographic projections are performed for line samples instead of the whole image, the proposed approach can achieve near-real time performance for localization accuracies similar to that in [25].

Chapter 3
Acceleration of vanishing point-based line sampling scheme for people localization and height estimation via footstep analysis

In this chapter, the efficiency of the above line sample-based approach is further improved by considering only one reference (ground) plane and, without performing 3D reconstruction, adopting a 3D line sampling scheme. Fig. 3.1 illustrates the schematic diagram of the proposed framework. First, the preprocessing procedures of camera calibration and foreground segmentation are executed. Next, we generate lines originated from the vanishing point of vertical lines in the scene to sample the foreground objects (people) in each camera view, as in [26]. The line samples of foreground objects from all camera views are then projected onto the ground plane via homography, with regions crossed through by a large number of projected sample lines identified as candidate people regions. We then generate (vertical) 3D sample lines for these candidate people regions, refine their two ends, and remove those not covered by enough foreground pixels in all views. Finally, the remaining 3D sample lines are grouped into individual axes to indicate people locations. Additionally, the height of each person can also be estimated as by-product.

3.1 Finding candidate people regions (blocks)

According Fig. 3.1, we first generate 2D sample lines, originated from the vanishing point, of foreground regions in each camera view. The sample lines containing very few foreground pixels are discarded since they contribute little to the following localization process. Then, the remaining sample lines are projected onto the ground plane via homography. It is easy to see that the more a region is crossed through by the projected sample lines, the more likely the region contains a person. Thus, we discretize the ground plane into a grid of 50cm × 50cm blocks, each has about the area a standing person occupies, and count the number of crossing sample lines for each block.

However, the above line counts may distribute across neighboring blocks, as shown in Fig. 3.2(a). Thus, we add a second grid, which has an offset of 25cm in both X and Y directions (on the ground plane) from the first one. Note that the second grid can have higher counts in some grids for the above example, as shown Fig. 3.2(b). After merging the two layers of grids, we retain the higher count for each quarter block, as illustrated in Fig. 3.2(c). Finally, the quarter blocks whose counts are greater than a threshold T_{cn}[7] are identified as candidate people blocks (CPBs).

[7] We set T_{cn} =8, which means the block is crossed through by sample lines from at least two camera views.

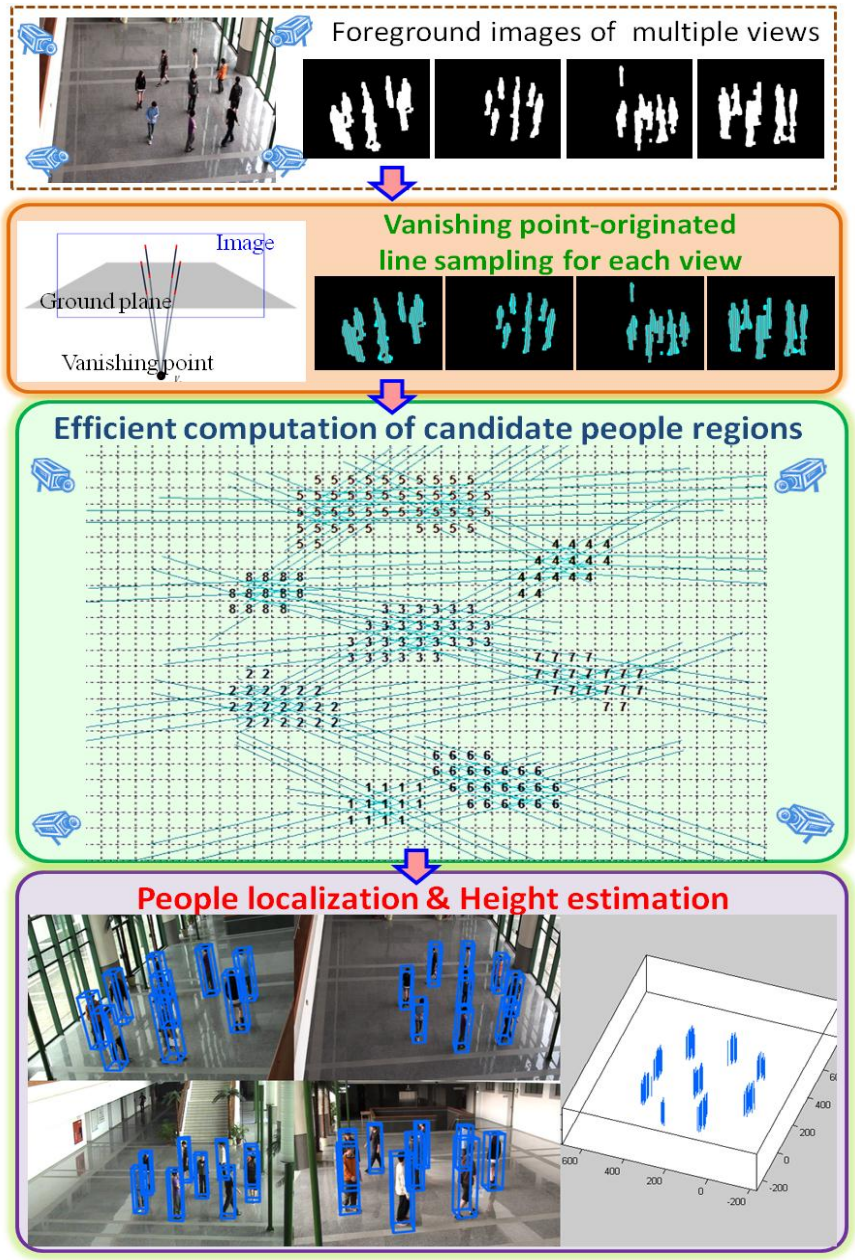
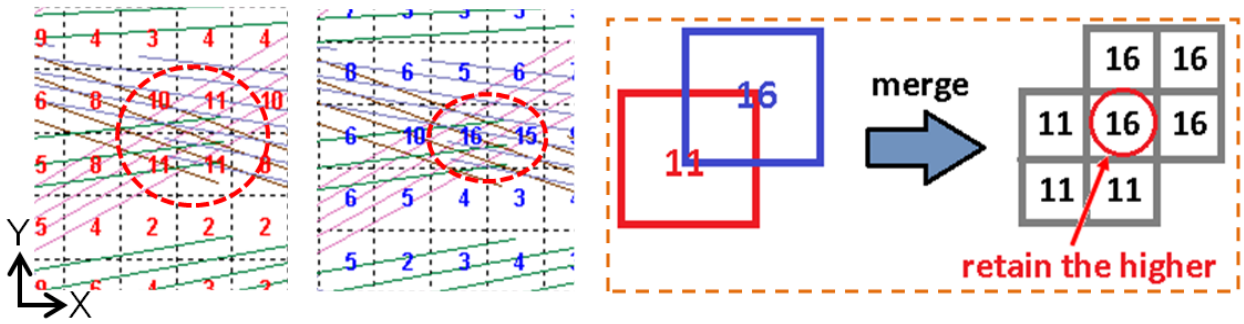


Fig. 3.1. Schematic diagram of the proposed people localization framework.



(a) (b) (c)

Fig. 3.2. Finding candidate people blocks (CPBs) by two-layered grids. (a) Layer 1 grid. (b) Layer 2 grid. (c) Merging the two-layered grids.

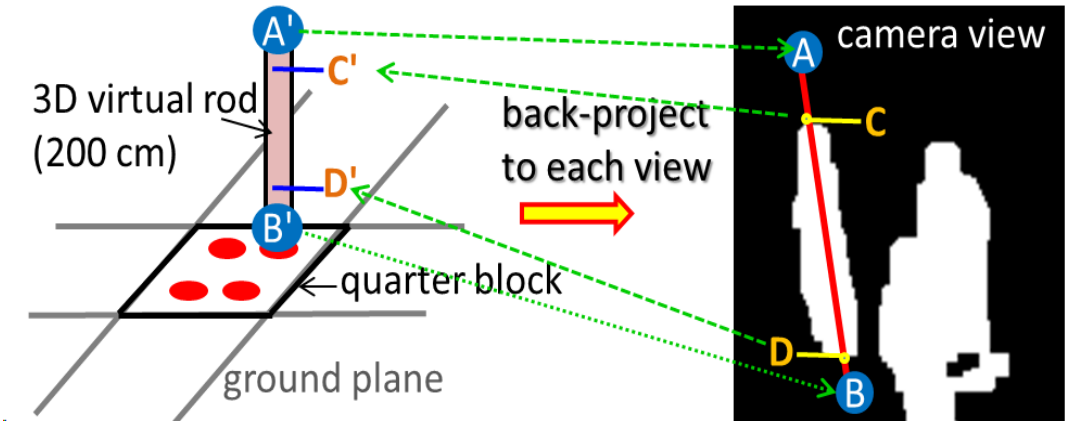


Fig. 3.3. Building and refining 3D virtual rods.

3.2 People localization and height estimation

In this section, to achieve the goal of people localization and height estimation, vertical line samples of human body are generated for the above CPBs. These line samples are then refined with respect to image foreground from different views, screened by some physical properties of human body, and grouped into axes of individual persons. In particular, four equally-spaced rods of 200cm in height are established on each CPB, as shown in Fig. 3.3. For each rod, we back-project it onto each camera view, and inwardly refine its top and bottom (*C* and *D* in Fig. 3.3, as well as *C* and *D'* calculated using view-invariant cross-ratio) until they are covered by a foreground region. For error tolerance, e.g., to cope with noises and occlusion, the intersection of all the refined 3D rods for each ground location from different camera views is adopted as the final line sample of possible human body.

Based on physical shape/size of a human body, we then apply the rules, as described in Subsection 2.2.1, to filter out incorrect 3D line samples obtained above. Also, the grouping procedure described in Subsection 2.2.2 is applied. Finally, for each group, the average location (maximum height) of the line samples is regarded as a person's location (height).

3.3 Experiments

To evaluate our methods under different degrees of occlusion, we captured several video sequences of indoor and outdoor scenes. For each scene, calibration pillars are placed vertically and then removed from the scene for the estimation of camera centers, vanishing points, and multiple homographic matrices (see Appendix A). These sequences are captured with different

numbers and trajectories of people. The computation is performed with a PC under Windows 7 with 4 GB RAM and a 2.4G Intel Core2 Duo CPU, without using any additional hardware.

Fig. 3.4 shows an instance of scenario S1 captured from four different viewing directions with a 360×240 image resolution. The average distance between the cameras and the monitored area is about 15m. One can see that the lighting conditions are quite complicated. The sun light may come through the windows directly and the reflections from the floor can be seen clearly. A total of 691 frames are captured for S1 wherein eight persons are walking around the ninth one standing near the center of the monitored area.

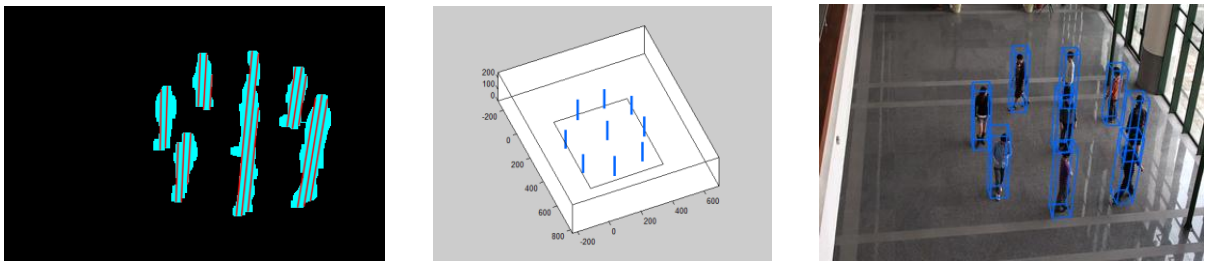
Figs. 3.5(a) and (b) show 2D line samples generated for Fig. 3.4(b) and the reconstructed 3D MAs, viewing from a slightly higher elevation angle, respectively. In addition, for a closer examination of the correctness of the proposed people localization and height estimation scheme, bounding boxes with a fixed cross-section, and with their height obtained from derived 3D MAs, are back-projected to the captured images, as shown in Fig. 3.5(c) for the image shown in Fig. 3.4(b). One can see that these bounding boxes do overlay nicely with the corresponding individuals. The recall and precision rates for the whole sequence are evaluated as 96.3% and 95.9%, respectively.

Fig. 3.6 shows similar localization results for scenario S2, which has the same people count as that for S1, but the nine people are walking randomly in the scene so that the occlusion among them becomes more serious. As a result, both the recall and precision rates are decreased slightly. To further examine the robustness of our method under serious occlusion, scenario S3 is evaluated, which is similar to S2 but having twelve persons randomly walking in the scene. Since the scene is becoming more crowded and serious occlusion may occur more frequently, foregrounds of different persons may easily merge into larger regions, as shown in Fig. 3.7(a). While satisfactory localization results are obtained in Figs. 3.7(b) and (c), the recall and precision rates for S3 are decreased to 91.9% and 90.0%, respectively.

The performance of the people localization approach described in this chapter is presented in Table 3.1. The precision and recall rates in all the three scenes are above 90%. Furthermore, the proposed approach achieves very high computational efficiency, even for the crowded scene S3, wherein 12 persons can be located quite accurately at a high processing speed of about 100 fps. For performance comparison, similar results of people localization obtained in [26] are listed in Table 3.2. One can see that the approach proposed in this chapter achieves similar precision and recall rates as in [26]. However, the processing speed is enhanced (about 2.6 times faster than [26]) due to the use of 3D line samples, instead of reconstructing 3D major axes via computing pairwise

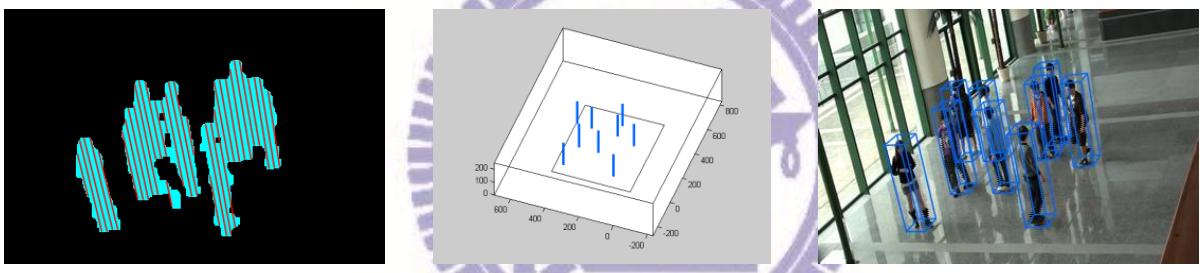
(a) (b) (c) (d)

Fig. 3.4. An instance of scenario S1, captured from four different viewing directions.



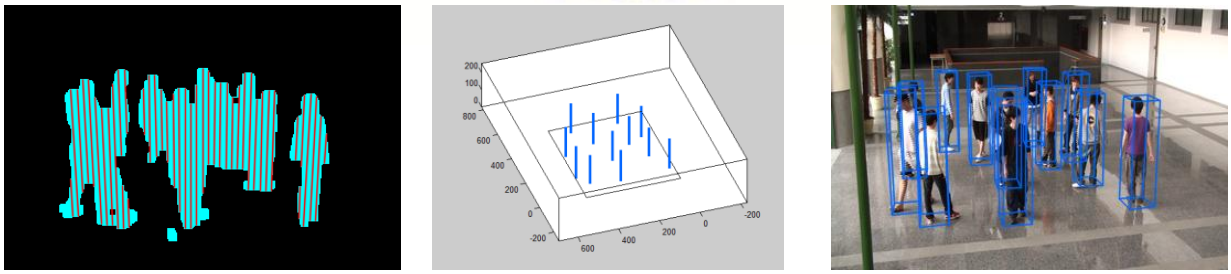
(a) (b) (c)

Fig. 3.5. Localization results for scenario S1. (a) Segmented foreground regions and 2D line samples for Fig. 3.6(b). (b) 3D major axes to represent different persons in the scene. (c) Localization results illustrated with bounding boxes.



(a) (b) (c)

Fig. 3.6. Localization results, similar to those shown in Fig. 3.7, for scenario S2.



(a) (b) (c)

Fig. 3.7. Localization results, similar to those shown in Fig. 3.7, for scenario S3.

Table 3.1. Performance of the proposed approach in this chapter.

Sequence	Recall	Precision	Avg. error	FPS
S1	96.3%	95.9%	12.16cm	30.74(0.47)
S2	95.2%	95.3%	10.94cm	32.06(0.52)
S3	91.9%	90.0%	11.32cm	23.78(0.41)

Table 3.2. Performance of people localization of [26].

Sequence	Recall	Precision	Avg. error	FPS
S1	92.0%	95.7%	11.60cm	11.62(1.008)
S2	94.9%	97.3%	10.00 cm	12.05(1.201)
S3	93.3%	94.3%	10.28 cm	8.34(1.025)

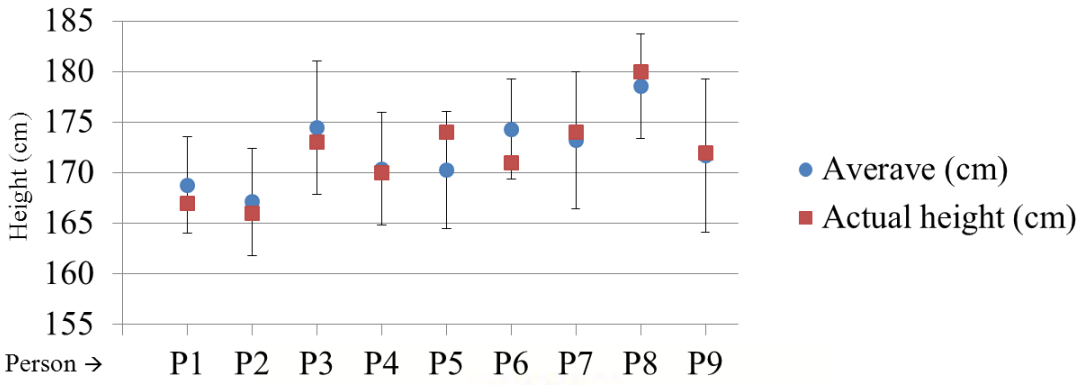


Fig. 3.8. Results of height estimation for S1.

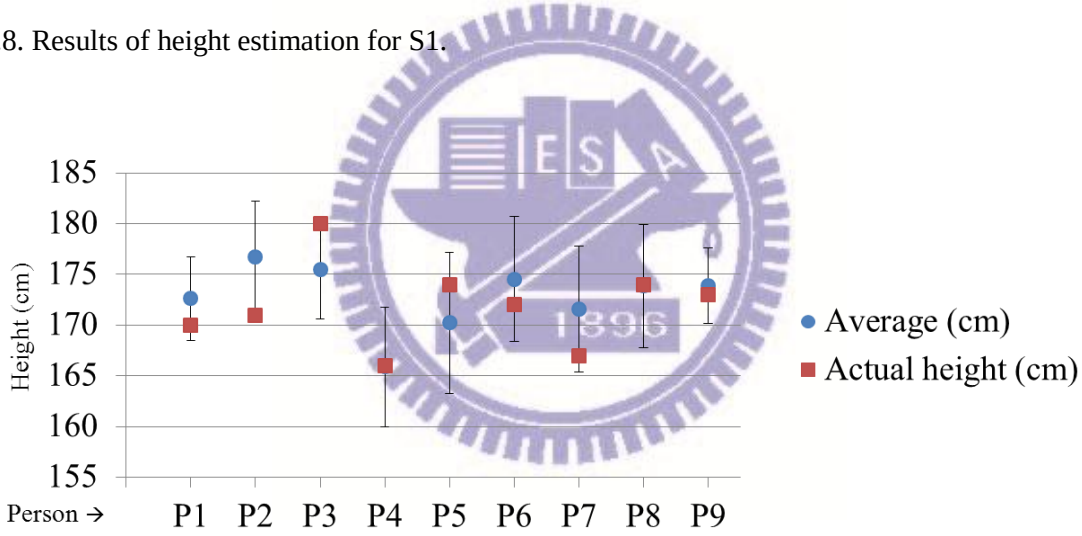


Fig. 3.9. Results of height estimation for S2.

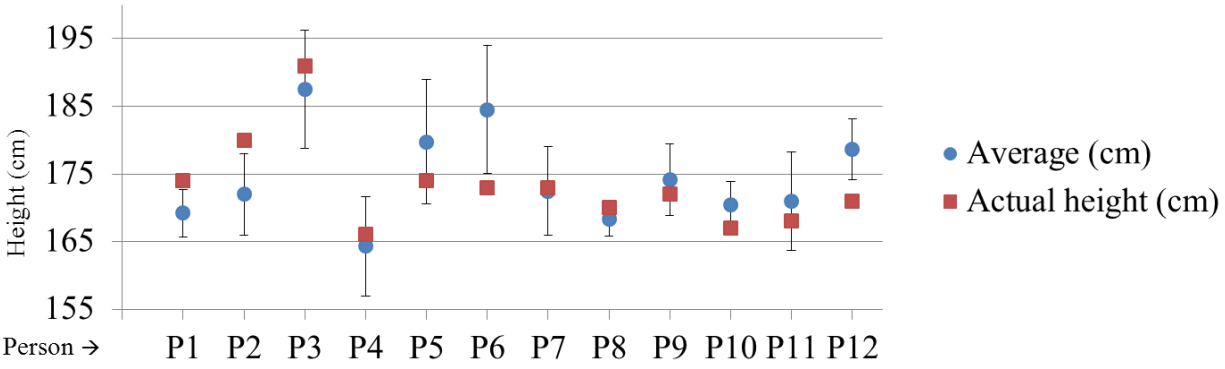


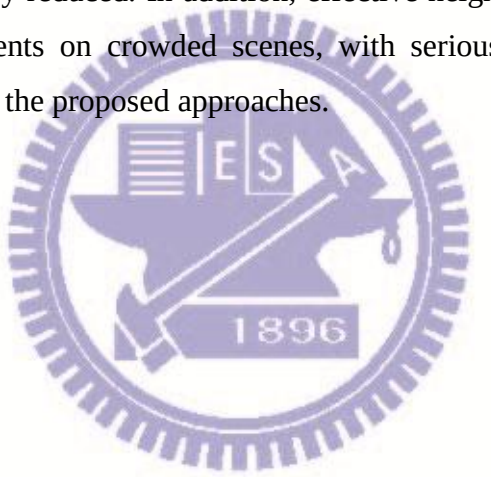
Fig. 3.10. Results of height estimation for S3.

intersections of sample lines of image foreground projected at different heights.

The results of person height estimation for S1 are presented in Fig. 3.8, where red squares indicate the actual heights and blue dots represent the estimated heights together with intervals of unit standard deviations. One can see the errors are less than 5cm. Similar estimation results for S2 can be observed in Fig. 3.9. However, in Fig. 3.10, the results of height estimation of a person (P6) has an error of more than 10cm, which may result from more serious occlusion.

3.4 Summary

We propose an efficient and effective approach for people localization using multiple cameras. Enhanced from [26], we retain the advantage of vanishing point-based line sampling, and develop a 3D line sampling scheme to estimate people locations, instead of reconstructing 3D major axes via computing pairwise intersections of the sample lines at different heights in [26]. The computation cost is greatly reduced. In addition, effective height estimation is also proposed in this chapter. The experiments on crowded scenes, with serious occlusions, also verify the effectiveness and efficiency of the proposed approaches.



Chapter 4
Enhancement of line-based people localization

In this chapter, enhancement of the efficiency of the people localization approach described in Chapter 2 (see also [26]) is considered. The three major improvements include (i) more efficient 3D reconstruction, (ii) more effective filtering of reconstructed 3D line samples, and (iii) the introduction of a view-invariant measure of line correspondence for early screen. While (i) and (ii) are direct improvements/enhancement of the approach presented in Chapter 2, (iii) introduces a new way of measuring the correspondence of two line samples obtained in different views.

4.1 Efficient 3D line construction from intersection of two triangles

While the approach described in Chapter 2 takes a lot computation time to calculate intersection points on multiple reference planes, as shown in Fig. 4.1 (left), an equivalent reconstruction of the 3D axis can actually be obtained by intersecting the two triangles[8], as shown in Fig. 4.1 (right). By adopting such a method, the computational time, which does not depend on the number of intersection points (reference planes), is expected to be decreased greatly. Axis points can then be estimated by a direct sampling along the 3D axis if necessary.

4.2 Refinement and verification of reconstructed 3D line samples

Although the rules of geometric filtering adopted in Chapter 2 are low-cost and effective, more filtering rules may be included to reduce miss detections. Since the two ends of a 3D line sample reconstructed above may be inaccurate, e.g., due to noise. We propose a refinement procedure to improve their precision. Additionally, two new rules are added, one before and the other after the refinement procedure, to increase the computation speed. Thus, the entire filtering procedure becomes more precise and effective. In particular the following new rule together with Rules 1-2, will be applied to a line sample right after the 3D reconstructoin,

3) The height of its top end point P^{ht} is lower than T_{tl}.

Fig. 4.2(a) shows line samples which survive Rules 1-3.

The main objective of the above three rules is to preserve two kinds of 3D line samples which correspond to (i) the full length of a standing/walking person or (ii) the head and torso of a

[8] The camera centers can be found in advance by at least two of the aforementioned four pillars.

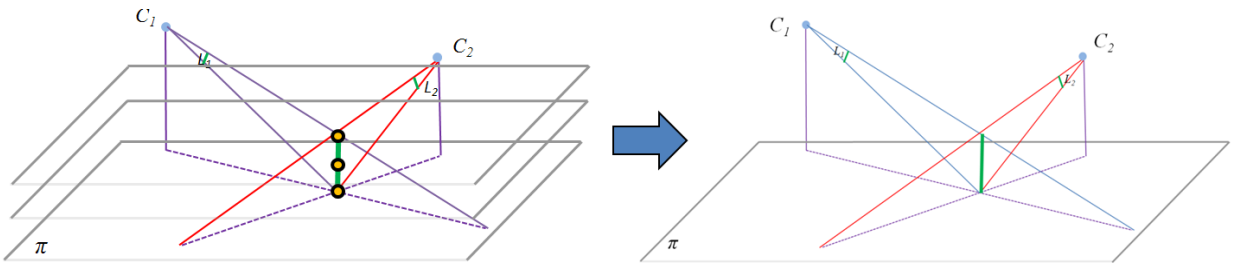


Fig. 4.1. Illustrations of the simplified 3D reconstruction.

person without his/her feet. By selecting appropriate thresholds, these three rules may also accommodate human activities such as jumping and squatting. In practice, these three rules can efficiently remove most of inappropriate 3D line samples, e.g., 84% of the originally reconstructed 3D line samples for the above example. However, since each 3D line sample is reconstructed by observations from two views only, the top and bottom ends of each 3D line sample may not be very accurate in position. To deal with such a problem, a refinement procedure using information from additional views, as described next, is adopted to find more accurate positions of the two end points before further verification of the 3D line samples are performed.

Conceptually, the refinement scheme is based on the fact that if a 3D line sample corresponds to a real person in the scene, its image in all views should be covered by foreground regions. In other words, its top and bottom end points will be covered by some foreground regions in *all* views. If that is not the case, the 3D line sample should be shortened until it falls within foreground regions in all views. Specifically, for each 3D line sample, we can use equally spaced sample points between its two ends P^{ht} and P^{hb} to form axis samples $\{P^{ht}, \ldots, P^{hb}\}$[9] (see (2.1) in Subsection 2.1.3). The refinement for the top end point corresponds to find the first sample point below P^{ht} such that it is covered by some foreground regions in all views. Similarly, the refinement of the bottom end point can be done by searching in the upward direction from P^{hb}.

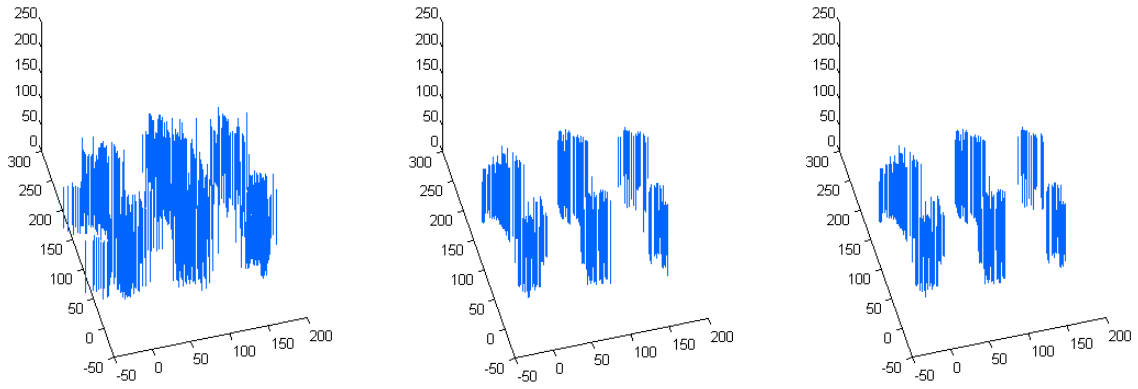
After such a refinement (shrinking) procedure, Rules 1-3 can be applied again, as well as using another new rule,

4) The height of top end point P^{ht} is higher than T_{th}.

to filter out inappropriate 3D line samples. One can see from Fig. 4.2(b) that rough people locations can be distinguished visually from the remaining 3D line samples. Finally, a threshold T_{fg} is used to filter out 3D line samples which do not have sufficient average foreground coverage rate (AFCR), as shown in Fig. 4.2(c)[10].

[9] The interpolation spacing between two adjacent sample points corresponds to a total number of N_{plane} equally spaced reference planes between the ground plane and the plane with 250cm in height.

[10] In our implementation, each sample point of a 3D line sample is projected to all views to check if it is covered by foreground for the computation of AFCR. For example, AFCR for each of the green axes shown in Fig. 2.4 is equal to 100% with respect to all (three) views.



(a) (b) (c)

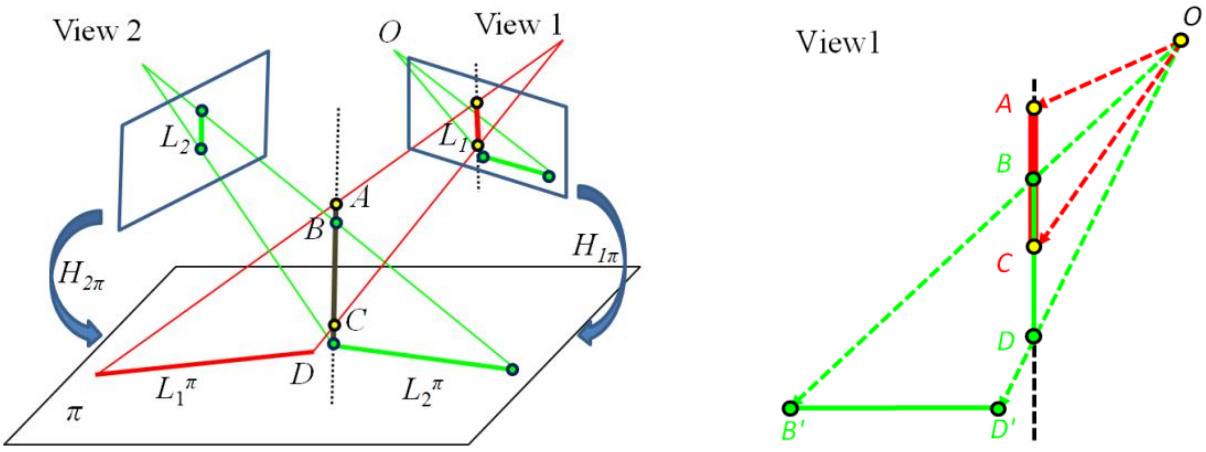
Fig. 4.2. Filtering results of input images shown in Figs. 2.6(a)-(d). (a) The unverified 3D line samples which survive Rules 1-3, (b) the refined line samples which survive Rules 1-4, (c) final line samples (see text).

4.3 Early screening for line correspondence

In this section, we propose a line correspondence measure of 2D line segments in two different views which is based on a formulation of cross ratio. Such a quantitative measure is view-invariant and can handle line segment of arbitrary configuration in the 3D scene and will be applied to the people localization methods described in Section 4.1 to filtered out non-corresponding line sample pairs before 3D reconstruction. Therefore, the computation speed of the proposed people localization can be further improved. We also convert the formulation to a more efficient form for computational efficiency. While such a measure is first illustrated via the concept of 3D reconstruction, as shown in Fig. 4.3(a), for a better understanding the basic idea, we will show that the measure can actually be computed in either one of the two views.

4.3.1 A view-invariant measure of line correspondence

Assume we have a pair of line samples in View 1 and View 2, respectively, and homographic matrices $H_{1\pi}$ and $H_{2\pi}$ between the two views and the ground plane π can be obtained from camera calibration. By projecting the line samples onto plane π, points A, B, C, and D can be obtained along a line in 3D space reconstructed by intersecting two planes each containing a camera center and the corresponding projected line sample. The lengths of $\overline{AB}$ and $\overline{CD}$ should be very small if the two line samples correspond to the same 3D line segment. If L_2^{π} is projected to View 1 (as $\overline{B'D'}$ in Fig. 4.3(b)) where A and C are end points of the line sample obtained in View 1, B and D can be calculated as intersection points of $\overrightarrow{OB'}$ and $\overrightarrow{OD'}$ and the line containing $\overline{AC}$,



(a) (b)

Fig. 4.3. (a) Illustration the basic idea of the proposed correspondence measure of two line features (samples). (b) Illustration of a general form of the view-invariant cross ratio.

respectively, with *O* being the camera center of View 2 which is found in advance.

Instead of using the above lengths, whose values will vary with view points, the view-invariant cross ratio, in one of several forms as discussed in [29], can be used to evaluate the degree of line correspondence as

$$CR = \frac{(\overrightarrow{OA} \times \overrightarrow{OB})(\overrightarrow{OC} \times \overrightarrow{OD})}{(\overrightarrow{OB} \times \overrightarrow{OC})(\overrightarrow{OA} \times \overrightarrow{OD})} \tag{3.1}$$

wherein each one of the four terms represents a signed triangular area in Fig. 4.3(b). If L_1 and L_2 correspond to a perfect match, points *A* and *B* (and points *C* and *D*) will coincide, and *CR* = 0. Moreover, since $\Delta OAB/\Delta OAB' = \Delta OBC/\Delta OB'C = \overline{OB}/\overline{OB'}$ and $\Delta OCD/\Delta OCD' = \Delta OAD/\Delta OAD' = \overline{OD}/\overline{OD'}$, we have

$$\frac{\Delta OAB \bullet \Delta OCD}{\Delta OBC \bullet \Delta OAD} = \frac{\Delta OAB' \bullet \Delta OCD'}{\Delta OB'C \bullet \Delta OAD'} \tag{3.2}$$

and (3.1) can be calculated more efficiently by

$$CR = \frac{(\overrightarrow{OA} \times \overrightarrow{OB'})(\overrightarrow{OC} \times \overrightarrow{OD'})}{(\overrightarrow{OB'} \times \overrightarrow{OC})(\overrightarrow{OA} \times \overrightarrow{OD'})} \tag{3.3}$$

since there is no need to compute *B* (*D*) from *B′* (*D′*). Thus, the proposed view-invariant measure of line correspondence, with a zero value representing a perfect match[11], can actually be evaluated in either one of the two views by first computing the homographic transform, e.g., $H_{1\pi}^{-1} H_{2\pi}$ for View 1 in Fig. 4.3(a), of two end points of a candidate line segment in another view.

[11] Values other than zero, as well as some special configurations of the above four points, will be considered in next subsection.

4.3.2 Applying the line correspondence measure to improve the efficiency of people localization

In this subsection, we will apply the proposed line correspondence measure to improve the efficiency of people localization Section 4.1[12]. Instead of finding correspondence of realistic line features in the scene, we will verify whether 2D line samples from different views belong to the same person. Thus, computations associated with a 3D line sample which are clearly resulted from two line samples of different persons can be avoided. Such computations include (i) 3D reconstruction of 3D line samples, as the mentioned in Section 4.1, (ii) 3D validations and (iii) 2D (foreground) consistency check of the 3D line sample. For example, physical properties of a human body can be used to validate the heights of *B* and *C*, and the length of $\overline{BC}$ in Fig. 4.3(a) for (ii). As for (iii), if a person does exist in the scene, the image of the person should be covered by some foreground regions in all views, so points on each 3D line samples are back projected to all views for further verification. While the complexity of (ii) is very low once (i) is done, (iii) is very expensive since each of the back projection requires a computation of homographic transformation.

Fig. 4.4 shows the procedure of determining whether two line samples obtained from two different views are likely to represent the same person using various parts of (3.3). First, if the denominator of (3.3) is not greater than zero, i.e.,

$$(\overrightarrow{OB'} \times \overrightarrow{OC})(\overrightarrow{OA} \times \overrightarrow{OD'}), \tag{3.4}$$

the reconstruction from the two line samples will have zero length. Thus, we can conclude that the samples belong to different persons. Except for the special cases, which seldom occur in practice, that one end or both ends of the two 2D line samples are reconstructed coincidentally that the numerator of (3.3) is equal to zero[13], (3.3) can be evaluated numerically to determine whether the reconstructed 3D line sample may result from the same person(s) that further refinements and verifications, e.g., (ii) and (iii), are needed.

Fig. 4.5 shows two numeric examples of the proposed line correspondence measure for some line samples shown in Figs. 2.6(b) and (d). While a small value (0.0034) is obtained for Fig. 4.5(a) where two line samples correspond to the same person, a larger value (0.0096) is obtained for Fig. 4.5(b) because of occlusion. A threshold of 0.01 is used in the experiments considered next to determine whether $|CR|$ is small enough.

[12] This is also true for the approach described in Chapter 2.

[13] It is easy to see that in either case, which hardly occurs in practice, additional views are still needed to refine and verify the reconstructed 3D line sample.

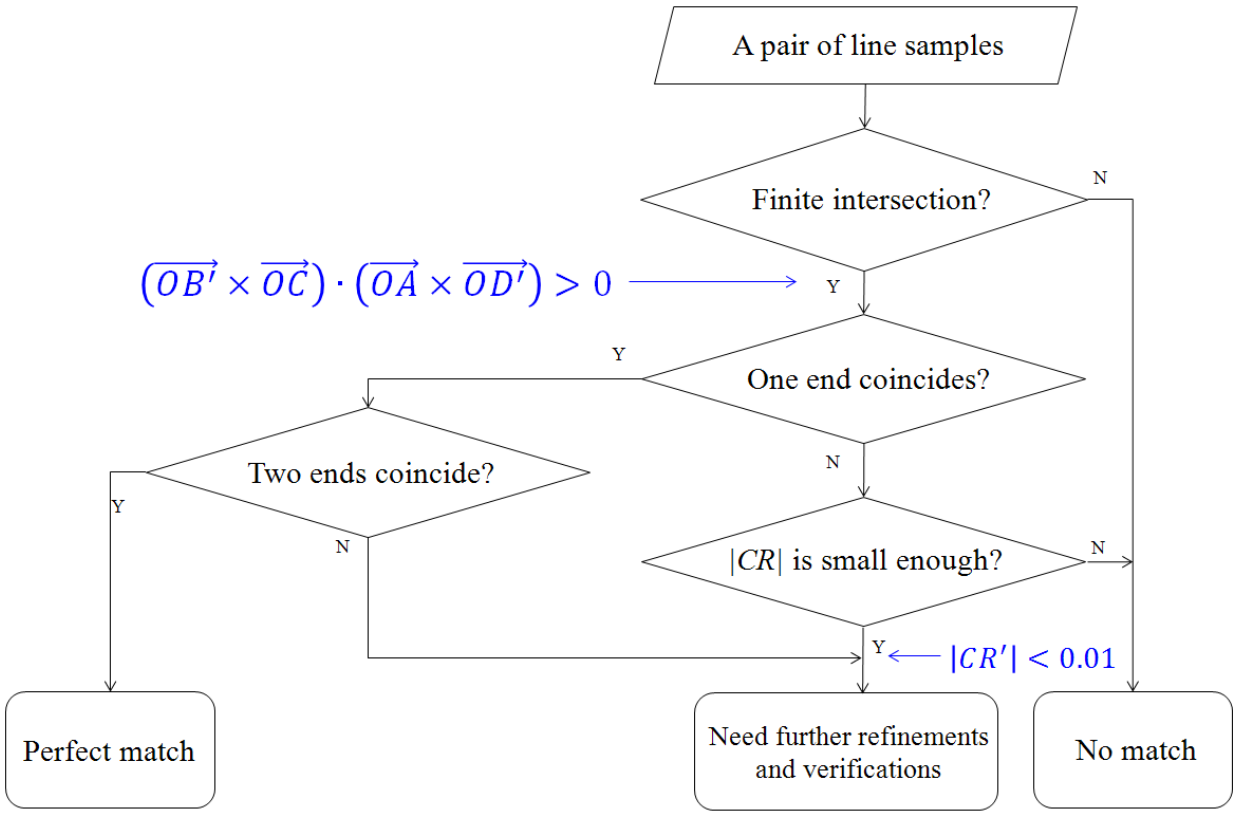
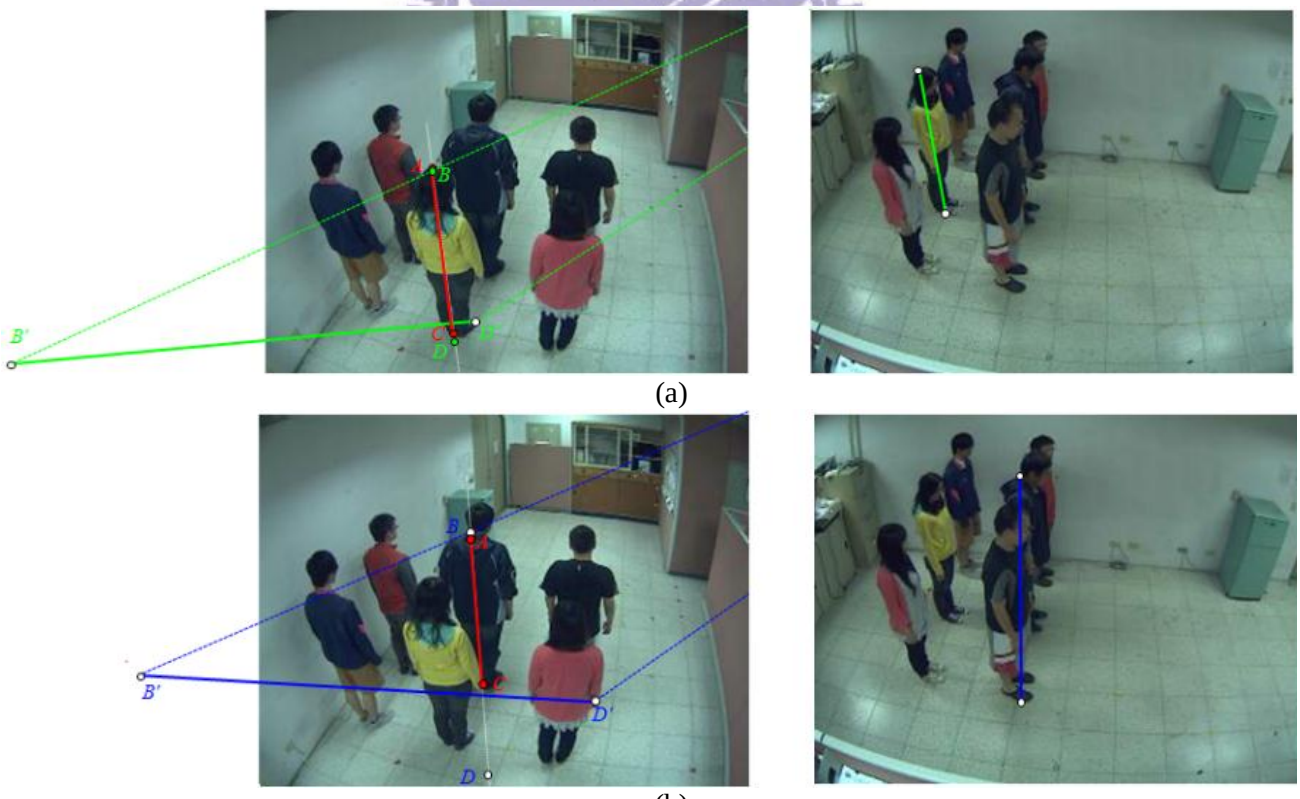


Fig. 4.4. Procedure to determine whether two line samples are likely to represent the same person.



(a)

(b)

Fig. 4.5. Illustration of numerical values of the proposed line correspondence measure (see text).

4.4 Experiments

In the following, we show the experiments of improvements described in Sections 4.1 and 4.2, respectively.

4.4.1 Applying the improvements described in Sections 4.1 and 4.2

In this subsection, improvements described in Section 4.1 are evaluated with several different videos taken from both indoor and outdoor scenes, with different degrees of occlusion. Comparisons with [25] and [26] are also included to show the proposed method can achieve comparable correctness/accuracy in localization but with much higher computation speed. Additionally, we investigate the performance of the proposed method with different numbers of cameras and densities of line samples in an image.

4.4.1.1 Experiments for different degrees of occlusion with indoor/outdoor sequences

The performance evaluation is implemented under Windows 7 with 4 GB RAM and a 2.4G Intel Core2 Duo CPU, without using any additional hardware. Table 4.1 summarizes detailed localization results of the proposed 3D line reconstruction method as well as three other methods. In addition to our previous work [26], a modified version[14] of the approach proposed in [25], is also implemented and tested. The proposed approach achieves the highest recall rates for S1-S3 while the other three methods achieve the highest precision rates for the three video sequences. Similarly, very small difference (within 0.65cm) among results obtained from these three methods can be found for the accuracy of derived people location except for the method proposed in Chapter 3. One can see the 3D line reconstruction method can achieve higher recall and precision rates (+3%) than the method described in Chapter 3 for S3. Overall, the mean value and standard deviation of (x-y) location errors of the proposed method for S1-S3, together, are equal to 10.70 cm and 5.90cm, respectively, which can hopefully be regarded as sufficient for many surveillance applications[15].

As for the computational speed, in frames per second (FPS), the values for different cases listed in Table 4.1 are evaluated without including the cost of foreground segmentation. One can

[14] In our implementation, which also does not perform people tracking, binary images of foregrounds are adopted as system input as the other two algorithms. A grid size of 100 × 100 is chosen for each of the twenty reference planes, with 10cm grid spacing. A grid point on the ground is regarded as occupied if more than T_{acc} = 11 grid points with the same horizontal coordinates (but on reference planes of different heights) correspond to image foreground in all (4) views. Then, connected component analysis is applied to identify connecting occupancy regions. The connected occupancy regions with very small areas, i.e., smaller than 22% of average area of such regions, are regarded as noise and are removed.

[15] The errors are only calculated for correctly detected people locations, which contribute to the precision rates listed in Table 3.1, i.e., with location errors less than 30cm.

Table 4.1. Localization results of sequences S1-S3.

Sequence	Number of frames/ persons	Method	Recall	Precision	Mean error (cm)	Frame per second
S1	691/9	Khan [25]	93.8%	95.7%	11.78(6.12)	0.46(0.003)
		Lo [26] (CH2)	92.0%	95.7%	11.60(5.91)	11.62(1.008)
		Lo [47] (CH3)	96.3%	**95.9%**	12.16(1.93)	30.74(0.47)
		Lo [48] (Secs. 4.1 & 4.2)	**96.5%**	95.6%	**11.42(5.89)**	**33.41(2.448)**
S2	776/9	Khan [25]	96.2%	**98.1%**	10.22(5.58)	0.46(0.003)
		Lo [26] (CH2)	94.9%	97.3%	**10.00(5.66)**	12.05(1.201)
		Lo [47] (CH3)	95.2%	95.3%	10.94(2.13)	32.06(0.52)
		Lo [48] (Secs. 4.1 & 4.2)	**96.8%**	97.0%	10.09(5.77)	**31.53(3.089)**
S3	271/12	Khan [25]	93.3%	94.2%	10.93(5.87)	0.46(0.003)
		Lo [26] (CH2)	93.3%	**94.3%**	**10.28(5.99)**	8.34(1.025)
		Lo [47] (CH3)	91.9%	90.0%	11.32(1.69)	23.78(0.41)
		Lo [48] (Secs. 4.1 & 4.2)	**95.2%**	93.6%	10.55(6.01)	**21.61(1.646)**

see that speed-up of more than an order of magnitude from the method in [25] can be achieved by the proposed approach, with as much as 70 times acceleration (near 2.7 times in speed improvement from our previous approach in [26]) in the process speed of S1. While real-time performance can be achieved for S1 and S2, the computation speed is down to a near real-time 21.61 FPS when the number of people increases to twelve[16]. Note that for [25] the computation times (in FPS) are about the same for different cases. This is because the time complexity in the generation of synergy maps is mainly depends on the size of each image frame and the total number of views. In addition, the computation speed of the 3D line construction method is quite similar to the method described in Chapter 3. However, the 3D line construction method can achieve higher recall and precision rates if the scene is more crowded, i.e., S2 and S3.

Although the above evaluations show that the proposed method can often provide reasonable good localization results, there are extreme cases of poor foreground segmentation which cannot be well handled with the proposed method. Figs. 4.6(a)-(h) show localization results and foreground regions for the 51th frame of S1. In Figs. 4.7(a) and (e), one can see the foreground segmentation of a person (in red circle) is very poor because of reflections as well as clustered

[16] This is because the computational time is dominated by the number of 2D line samples, which will grow with the area of foregrounds.

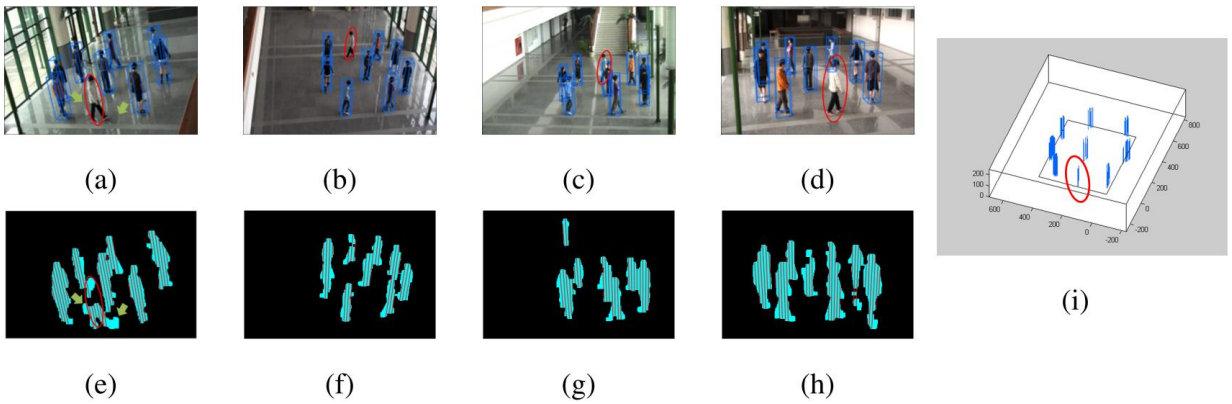


Fig 4.6. A failure example of the proposed method. (a)-(d) The localization results (illustrated with bounding boxes) of four views. (e)-(h) Corresponding foreground regions and 2D line samples. (i) 3D line samples to represent different persons in the scene.

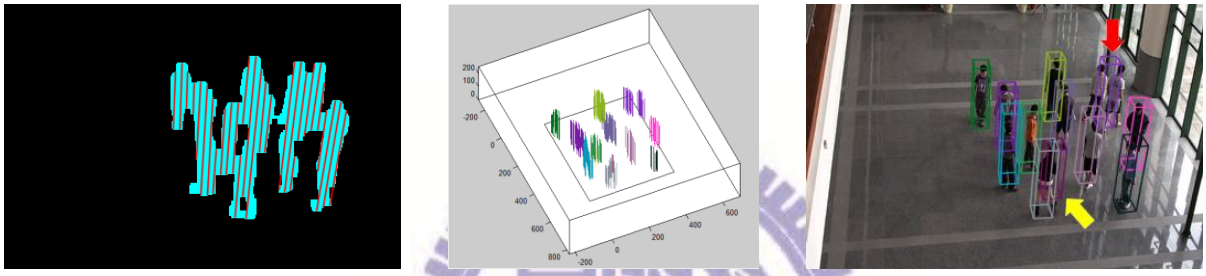


Fig. 4.7. An example of miss detections and false alarms of S3. (a) Segmented foreground regions and 2D line samples. (b) 3D line samples to represent different persons in the scene. (c) The localization results illustrated with bounding boxes. Note that corresponding colors are used in (b) and (c) for different groups/bounding boxes after grouping.

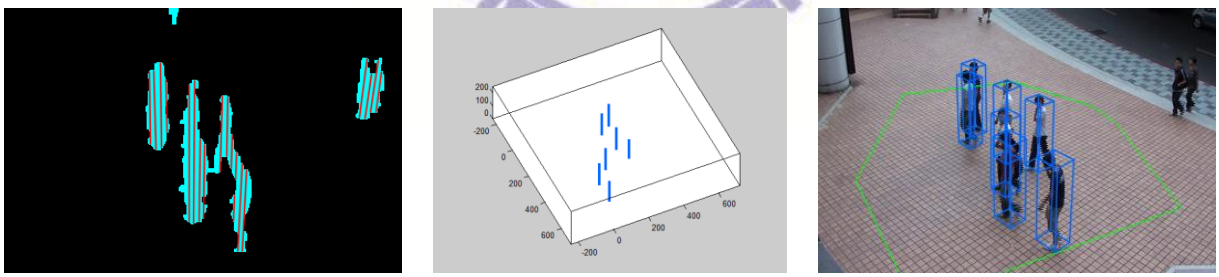


Fig. 4.8. Localization results for scenario S4.

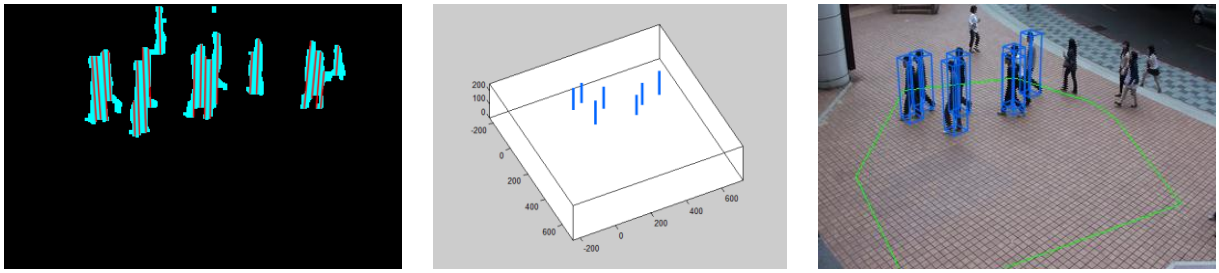


Fig. 4.9. Localization results for scenario S5.

background (see green arrows). Consequently, lesser 3D line samples are retained after the screening process, as show in Fig. 4.6(i), resulting a failure. Since some more 3D line samples can still be reconstructed correctly for that person as different time instances, erroneous results are generated for only 3 out of 20 frames (from frame 41 to frame 60), compared with 13 erroneous frames obtained from the method in [25].

On the other hand, problematic results may also be generated due to very serious occlusions. Firstly, as shown in Fig. 4.7, there may be ground region that is covered by foregrounds in all views. No matter a person does exist or not, a 3D MA will be generated. If such a 3D MA cannot be filtered out by the aforementioned geometric rules, a false alarm will occur (see the yellow arrow in Fig. 4.7(c))[17]. Secondly when the distances between people are too small (see the red arrow in Fig. 4.7(c)), their 3D line samples will be grouped into the same group (see Fig. 4.7(b)) resulting in two miss detections and one false alarm. This is because, for localization efficiency, the grouping scheme only determines whether the distance between two line samples is smaller than a threshold when grouping 3D line samples[18]. (More detailed discussion of the effect of the distance threshold can be founded in Appendix B.)

To further evaluate our method for outdoor environment, S4 and S5 are captured from a real scenario with image resolution of 360 × 240. In general, working in such an environment may be challenging for visual surveillance systems since there are more time varying factors such as illumination for object, speed of wind, and shadows of various strength. For the real scene under consideration, groups of people of different sizes are walking quickly through the monitored area[19] (green polygons in Figs. 4.8 and 4.9). Thus, less image frames are captured for S4 and S5 than those in S1-S3. Figs. 4.8 and 4.9 show snapshots of localization results for S4 and S5, respectively, with more statistics summarized in Table 4.2. One can see that the correctness/accuracy level similar to that shown in Table 4.1 can be achieved with the proposed approach except for larger differences between (i) recall and precision rates for S4, and (ii) mean localization errors for S4 and S5. Such differences may result from higher probability of the aforementioned occlusions for people walking together along a passage and/or complexities associated with an outdoor scene.

In practice, due to significant differences between the indoor and outdoor scenes where video sequences S1-S3 and S4-S5 are captured, respectively, different parameter values may need to be selected to achieve desirable localization results. In the next Subsection (4.4.1.2), effects of choosing different densities of 2D line samples in each image, as well as incorporating different

[17] Such a problem may be eliminated by adopting additional temporal information.

[18] To partially resolve this problem, a heuristic scheme is applied in our method. If a group containing a larger number of 3D line samples, it will be divided into two groups. Specifically, we calculate the average number of 3D line samples, N_C, in all groups, and divide a group into two groups if it contains more than more than $2N_C$ line samples.

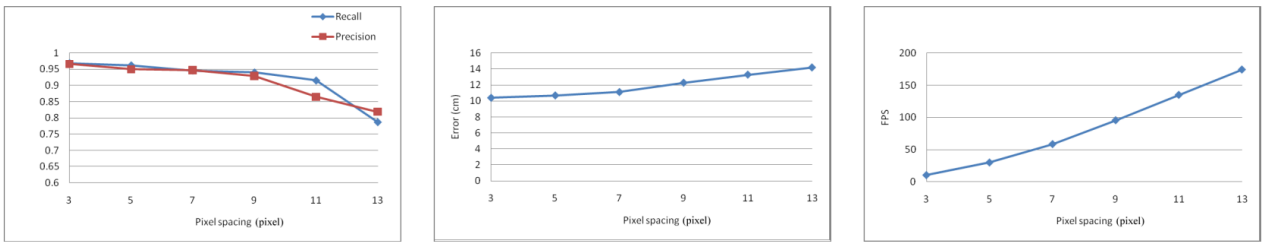
[19] It is assumed that the evaluation of people localization is only preformed for the monitored area.

Table 4.2. Localization results of sequences S4 and S5.

Sequence	Number of frames/ persons	Method	Recall	Precision	Mean error (cm)	Frame per second
S4	70/6-7	Khan [25]	**97.7%**	**91.1%**	9.08(5.30)	0.46(0.003)
		Lo [26] (CH2)	90.0%	75.4%	8.84(5.62)	12.05(1.201)
		Lo [47] (CH3)	97.0%	86.1%	10.11(1.97)	22.90(2.26)
		Lo [48] (Secs. 4.1 & 4.2)	97.5%	89.8%	**8.57**(5.05)	31.53(3.089)
S5	40/7	Khan [25]	97.1%	**97.8%**	11.48(6.25)	0.46(0.003)
		Lo [26] (CH2)	**97.5%**	91.0%	11.37(6.52)	8.34(1.025)
		Lo [47] (CH3)	94.3%	94.3%	**11.27**(2.61)	21.26(1.24)
		Lo [48] (Secs. 4.1 & 4.2)	95.0%	96.0%	11.70(6.02)	21.61(1.646)

Table 4.3. Results of using different numbers of cameras.

Number of cameras	3	4	5
Recall	95.4%	96.2%	98.3%
Precision	85.7%	95.0%	96.6%
Localization error (cm)	11.30	10.70	10.13
Frames per second	75.57	29.48	24.16



(a) (b) (c)

Fig. 4.10. Results of using different line densities (pixel-spacings, see text) with four cameras. (a) Recall and precision. (b) Localization error. (c) Computation speed.

numbers of camera in the proposed localization system, will be investigated (only for the indoor scene for brevity). While the two associated parameters will determine the initial amount of data to be processed by the proposed algorithm, other parameters will be used to tune the algorithm for

better performance under different environmental conditions, as will be discussed in Appendix B.

4.4.1.2 Experiments for different numbers of cameras and densities of sampling

To investigate the relationship between performance of localization and the numbers of cameras, the indoor scenarios S1-S3 are examined with an additional view captured from a different camera, and the results are presented in Table 4.3. One can see that while similar recall rates can be obtained by using different numbers of cameras, the precision rate of using three cameras is much lower than if four or five cameras are used. This implies that using only three cameras may not be sufficient when there are serious occlusions. In addition to above performance indices, adding more cameras also improves the system performance in terms of the localization accuracy. However, if slight degradations in these performance indices are acceptable, a set of four cameras may be used if hardware (cameras) cost is of major concern.

In order to investigate the influence of densities of sample lines in an image on the localization performance, a very simple sampling scheme is adopted in our method. In particular, the line samples are originated from the vanishing point to equally-spaced image pixels at the bottom row of the captured image. Fig. 4.10(a) shows the decreases of both the recall and precision rates with such pixel-spacing[20]. One can see that for spacing less than ten, similar recall and precision rates can be obtained, and a larger spacing seems to capture inadequate information for localization. Fig. 4.10(b) shows that the localization errors are growing slightly with pixel-spacing. Whether the localization errors due to different pixel-spacings are acceptable will depend on applications under consideration. Finally, Fig. 4.10(c) shows the growth of computation speed with pixel-spacing. Again, the choice among different pixel-spacing will depend on the requirement of system performance.

4.4.1.3 Exploring for more challenging scenes

As a preliminary investigation of possible extensions needed for the proposed approach to work for more challenging scenes, a busy street scene is considered in this subsection (4.4.1.3). Fig. 4.11 shows people localization results obtained by directly applying our algorithm, for the monitored area marked in green[21], for a time instance while six persons are crossing a street. Besides failure cases mentioned earlier (the red arrow indicates the merge of two persons, as in Fig. 4.7), additional interferences from non-human foreground objects (vehicles) include: (i)

[20] While a spacing of 5 pixels is selected for S1-S3, a spacing of 4.4 pixels is selected for S4-S5.

[21] Similar to the experiments conducted on S4 and S5, the evaluation of people localization is only preformed for the monitored area, and the image resolution is 360 × 240.

vehicle-people occlusion and (ii) presence of vehicles in the monitored area. While (i) can be seen in all four views but does not result in a problem in this case, (ii) does cause a false alarm (shown as a big (dark purple) group in Fig. 4.11(i)). Overall, the recall and precision rates for this challenging scene are evaluated as 80.9% and 80.2%, respectively, for a total of 108 image frames.

4.4.1.4 Summary

Instead of using all foreground pixels, line samples from multiple views are used to find possible 3D line samples of human body efficiently. While our earlier approach in [26] is a direct extension of the approach in [25] in that projection of pixels (lines in [26]) are computed for horizontal planes first, the algorithm presented in this Section 4.1 reconstructs the above samples in the 3D space directly. Additional efficiency of the proposed approach arises from effective screening of these 1D samples using new geometric constraints of the body. Such efficiency is crucial for certain surveillance applications which demand prompt attention (and high processing speed) with people localization being part of the complete process[22]. Experimental results demonstrate that the proposed method can handle serious occlusions in quite crowed scenes to provide localization results with correctness and accuracy, and localization accuracy, comparable to that attained with a modified version of [25], but with much higher processing speed. Additionally, because the proposed localization approach is based on 3D reconstruction/sampling, it is possible to extend the approach to track people in the 3D space[23].

4.4.2 Applying the improvements described in Section 4.3

For the screening procedure shown in Fig. 4.4, the filtering results for S1-S3 are shown in Table. 4.4. One can see the first step, by evaluating (3.4) only, can already filter out about 50% of total line sample pairs, and only about 7% line samples will be reconstructed for further processing. The evaluation is performed with a PC with 4 GB RAM and a 2.4G Intel i5 M520 CPU. As for the overall performance in people localization, one can see from Table. 4.5 that while recall, precision, and localization errors of the accelerated method are comparable with that in [26], the execution speed is more than three times of that in [26] for all sequences. We also applied the line correspondence measure to improve the efficiency of the method described in Section 4.1. Table 4.6 shows less acceleration compared to Table 4.5 because the computation cost of the rest processes of the method described in Section 4.1 is much lower than those in [26].

[22] For example, while localization-based people tracking is often needed in intruder detection and abnormal behavior detection, if such functions are to be implemented with no special hardware for acceleration, our approach will have better chance of fulfilling the requirement of real-time performance than that presented in [25]. As another example, effective people tracking based on the localization results may need to be developed for similar applications, which may be more sophisticated than that presented in [25] and implemented without any special hardware.

[23] To that end, constraints for human standing on the ground plane should be removed, which include Rules 2-4

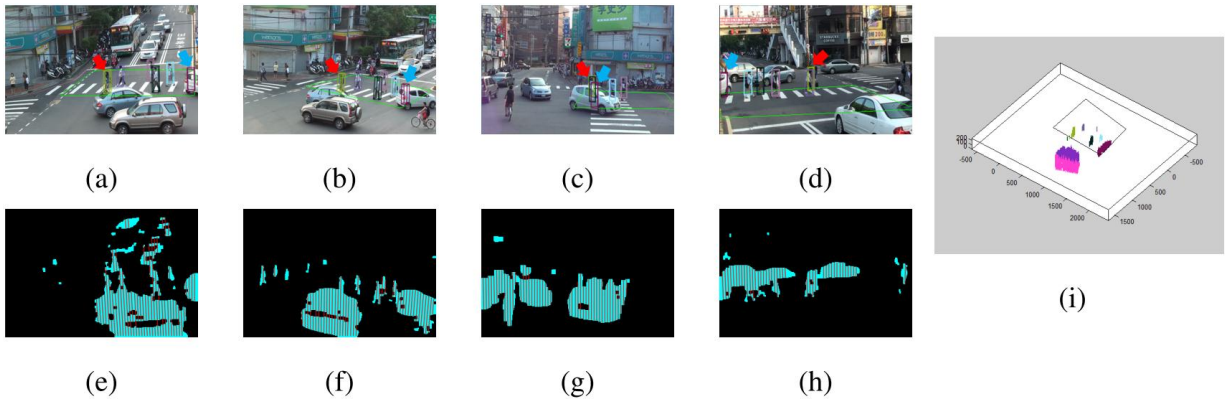


Fig. 4.11. A more challenging localization example for a busy street scene. (a)-(d) The localization results (illustrated with bounding boxes) of four views. (e)-(h) Corresponding foreground regions and 2D line samples. (i) 3D line samples to represent different persons in the scene.

Table 4.4. Filtering results of Fig. 4.4.

Sequence	Total pairs	No match pairs	Reconstructed pairs
S1	8282914	52.3%	6.3%
S2	8550765	48.9%	7.7%
S3	4298920	48.2%	6.7%

Table 4.5. Localization results of sequences S1-S3.

Sequence	Method	Recall	Precision	Mean error	FPS
S1	Lo [26]	93.7%	95.1%	11.07	26.69
	Lo [49] (Lo [26] + Sec. 4.3)	94.8%	95.1%	11.05	83.46 ← (3.12x)
S2	Lo [26]	94.6%	94.2%	9.57	26.33
	Lo [49] (Lo [26] + Sec. 4.3)	97.0%	93.1%	9.53	76.60 ← (2.90x)
S3	Lo [26]	92.3%	91.9%	9.57	18.09
	Lo [49] (Lo [26] + Sec. 4.3)	91.7%	95.6%	9.87	63.79 ← (3.50x)

Table 4.6. Localization results of the method proposed in Sections 4.1, 4.2, and 4.3.

Sequence	Method	Recall	Precision	Mean error	FPS
S1	Lo [48] (Secs. 4.1 & 4.2)	96.50%	95.60%	11.42(5.89)	127.80
	Lo [48, 49] (Secs. 4.1, 4.2, and 4.3)	93.83%	95.53%	11.25(5.93)	186.91←(1.46x)
S2	Lo [48] (Secs. 4.1 & 4.2)	96.80%	97.00%	10.09(5.77)	121.93
	Lo [48, 49] (Secs. 4.1, 4.2, and 4.3)	96.35%	96.69%	9.82 (5.57)	173.73←(1.42x)
S3	Lo [48] (Secs. 4.1 & 4.2)	95.20%	93.60%	10.55(6.01)	86.71
	Lo [48, 49] (Secs. 4.1, 4.2, and 4.3)	93.60%	94.53%	10.59(6.01)	140.12← 1.61x)

4.4.2.1 Summary of the experiments

A correspondence measure of 2D line segments in two different views is proposed. Such a measure can handle line segment of arbitrary configuration in the 3D scene and is view-invariant, i.e., same measurement can be obtained quantitatively from either one of a pair of views. Besides we also proposed a line-based people localization scheme by applying such a measure to improve the efficiency of the method described in [26] and Section 4.1. By verifying whether 2D line samples from different views belong to the same person, computations associated with invalid 3D line samples, which are resulted from different persons, can be avoided. Experiments are performed for videos of crowded scenes with various degrees of occlusion. Overall, people localization results, in terms of correctness and accuracy, comparable to the two localization methods can be obtained, but with more than 1.42 times increase in computation speed. Other applications of the proposed line correspondence measure, e.g., in robot SLAM, are currently under investigation.

4.5 Summary

In order to enhance the efficiency of [25], we propose a vanishing point-based line sampling technique in [26] (Chapter 2). While the main idea of the approach presented in [25] is to project dense 2D samples (image pixels) onto multiple (horizontal) planar surfaces in the 3D space (before these data are fused into 3D object distributions), it is simplified in [26] by projecting 1D image samples[24], i.e., lines passing through the vanishing point of vertical lines in the 3D space, instead (before their intersections are grouped into 3D line samples of the crowd through grouping). To further improve the efficiency of people localization, a novel approach is proposed in this chapter which projects the above line samples directly into the 3D space, i.e., along a fan of vertical planes originated from the vertical axis containing the camera center, to generate possible 1D (vertical line) samples of the 3D object[25]. Since realistic constraints of a human body can be adopted to refine and to verify these object samples, localization results compatible with those in [25] can be achieved, but with more than an order of magnitude in processing speed. Furthermore, we proposed a view-invariant correspondence measure of line segments in different images in Section 4.3 to improve the efficiency of the method proposed in [26] and Section 4.1.

[24] In the rest of this thesis, we will referred to these samples as 2D line samples.
[25] In the rest of this thesis, we will referred to these samples as 3D line samples.

Chapter 5
Error analysis of 3D line reconstruction from intersection of two triangles

In Chapter 3, we proposed a people localization method which is based on 3D line reconstruction from intersection of two triangles and can achieve high recall and precision rates. An empirical error analysis scheme for similar 3D line reconstruction is developed in this chapter for a simple pointing system. The related error analysis results are expected to further improve the accuracy for the people localization method mentioned above.

5.1 Motivation

For many HCI applications, pointing directions of a user can be transformed conveniently into instructions such as asking a robot to move to desired positions or controlling a computer by a virtual mouse. While real-time computations of the pointing direction (and its target) for a user are often needed, accuracy and stability of the computation are the most desirable attributes of such pointing systems.

In some pointing systems, human hands are exploited to give instruction via associated direction vectors. For example, the connected line from the finger root to the fingertip is recognized as a pointing direction in [30], while the pointing direction is connected from head to hand in [31]. Similarly, one eye and one fingertip are consider to form a direction vector in [32], while similar vector is established by connecting a line from shoulder to arm in [33]. Instead of using skin color to detect pointing direction of a human hand, as in [31, 32], motion analysis of feature points of user's hand is adopted to estimate the shoulder point and the direction vector in [34]. In [35], a vision-based method is proposed to find the pointing directions which are extended from head to hand. In addition, artificial neural networks are used to find head orientation to improve the accuracy of pointing results. In general, to locate the pointing position in a 3D environment, some forms of 3D reconstruction need to be carried out to determine the direction vector. In [32, 33, 35, 36], 3D voxels of a pair feature points used in the pointing are calculated before such a vector is formed.

In order to study the accuracy and stability of pointing, a real-time, vision-based system similar to that presented in [31] but with pointing direction specified by a pointer is implemented (see Fig. 5.1). By considering the intersection of planes in the 3D world, the system first calculates two planes each formed by two endpoints of the pointer and the center of one of the two cameras. The intersection of these two planes then forms the direction vector. Instead of explicitly deriving

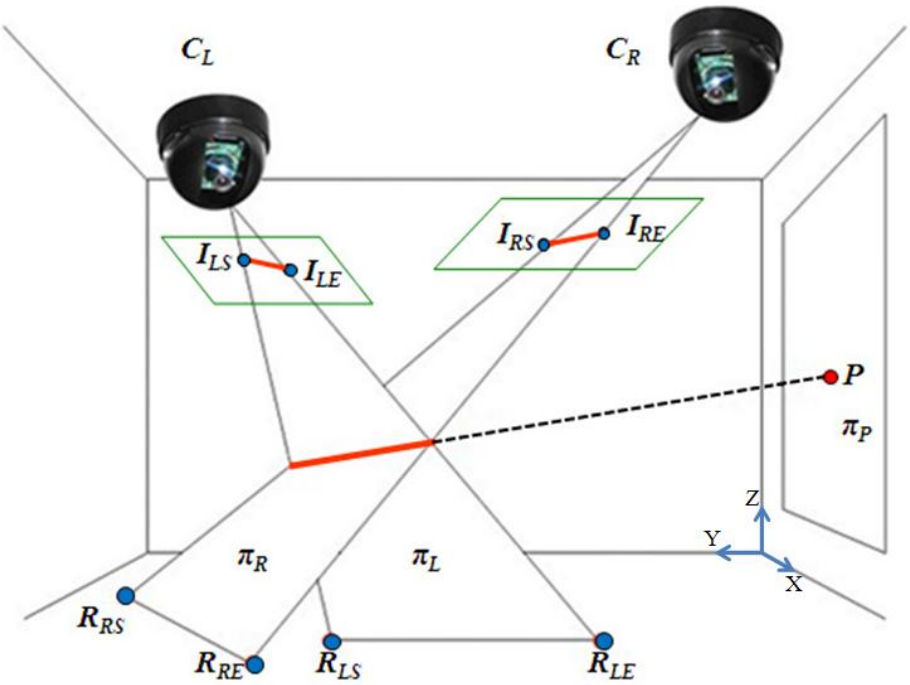


Fig. 5.1. Configuration of the pointing system and the reconstruction of a pointing point.

intrinsic and extrinsic camera parameters, the approach only needs the camera positions, and needs to calibrate the homographies, providing distortion in the camera from perspective projection is fixed.

For all pointing systems, different forms of measurement and computation errors can be generated during the reconstruction of the pointing line, which has five degrees of freedom, and a clear understanding of these errors may greatly improve the applicability of such systems. However, existing error analysis schemes are mainly concerned with planar localization, based on image data acquired by a single camera [37-41], as well as reconstruction of 3D point features using stereo cameras [42-45], which only have two/three degrees of freedom. In the following, an efficient error analysis scheme is established for an experimental pointing system by evaluating the error range of pointing results on a projection plane, e.g., a screen, with image data assumed to be corrupted by additive noises, as in some of the above approaches. Hopefully, with the help of such analysis, more robust pointing results can be achieved by selecting the most appropriate pointer positions, or pairs of cameras, that will result in minimal range of pointing error. In addition, more accurate people localization method can be achieved. While a pointer with bright color is used here to greatly reduce the influence of certain sources of error, e.g., those due to errors in image feature extraction, the error analysis results will provide an upper bound of pointing accuracy for systems using different pointers, e.g., those discussed in [30-33, 35, 36], if similar reconstruction process is adopted.

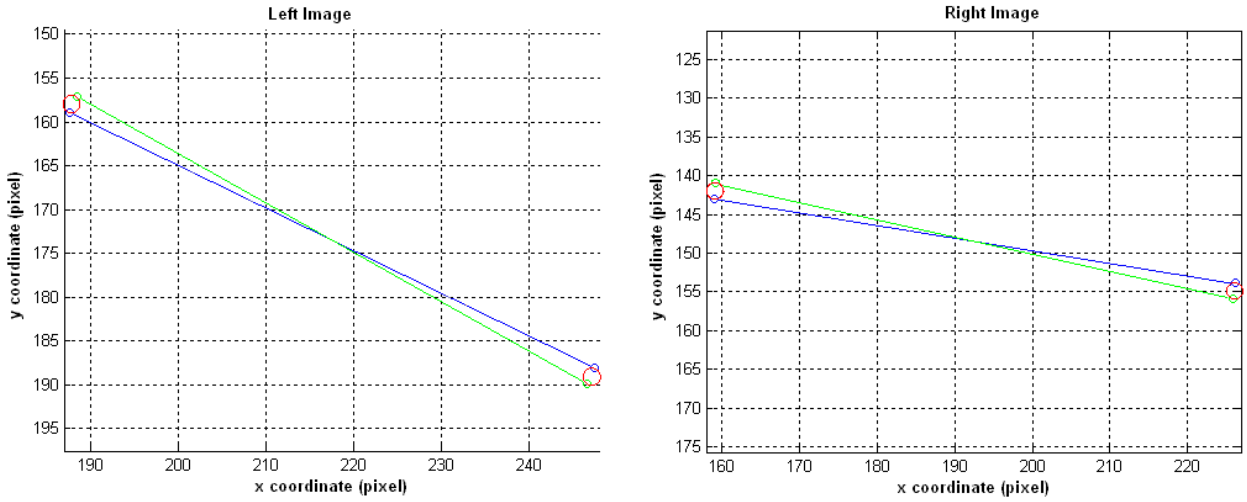


Fig. 5.2. Noise circles (simulated points) for the pointer endpoints located in stereo images shown in Fig. 5.1, and their CICTs (see text).

5.2 An experimental pointing system

In this section we describe the configuration of a simple experimental pointing system used in this thesis which is similar to [31] but using homographic transformations to derive the pointer direction. The system uses two cameras mounted on the ceiling, four reference points on the floor, and a projection plane perpendicular to the ground (see Fig. 5.1).[26] A two-fold simplification is associated with such a pointing system. First, unlike in [30], [32] which use color and brightness to find hand region, we use a pointer with bright color to reduce the complexity in feature extraction. Second, unlike in [31] and [33], the simple camera calibration similar to that used in [30] is adopted for 3D reconstruction based on homographic transformations. With such a simplified system configuration, the errors generated during the reconstruction process can be studied more easily and understood more clearly.

In the proposed approach, the left and right images are acquired simultaneously from the two cameras. For each of the stereo images, the image pixels of the pointer are obtained through a preprocessing step (see Appendix), and we calculate a best-fit line of these pixels via principal components analysis (PCA). The line intersects the bounding box of the above image pixels at two points, which are then regarded as (extended) endpoints of the pointer in the image. In this section, the two sets of pointer endpoints are denoted as $\{I_{LS}, I_{LE}\}$ and $\{I_{RS}, I_{RE}\}$ for the left and right images, respectively.

[26] The coordinates (in cm) of the two cameras C_L and C_R are (192, 365, 264) and (493, 122, 264), and the coordinate of the four corners of the projection plane are (115, 0, 243), (115, 0, 108), (295, 0, 108), and (295, 0, 243). In general, the pointing system can be used to identify a non-planar object at various locations in the 3D space. The projection plane is included here for demonstration purpose only.

Once the positions of the above endpoints are located in the left and right images, we use homographic transformation to find their projections, R_{LS}, R_{LE}, R_{RS}, and R_{RE} on the ground plane, as shown in Fig. 5.1.[27] Thus, plane π_L which contains R_{LS}, R_{LE}, and the center of the left camera C_L, and plane π_R which contains R_{RS}, R_{RE}, and C_R can be reconstructed. Planes π_L, π_R, and the projection plane π_P will then intersect at the pointing position P. Finally, we transform P into the 2D coordinate of the monitor screen through another homographic transformation, and display the reconstructed pointing position (RPP).

With the above simple reconstruction process (see Appendix), there is no need to find all camera parameters, as required in typical 3D reconstruction approaches, and the pointing system can operate efficiently in real-time. However, noises in the imaging process may result in reconstruction errors and thus unstable pointing position. To understand the influence of such undesirable effects, and hopefully to develop a scheme to reduce the influence accordingly, an efficient error analysis approach to the estimation of pointing errors is proposed and presented next.

5.3 Error analysis

For the real world implementation of the pointing system described above, the RPP and actual pointing position are not always the same. Such discrepancies can be categorized into (i) static and (ii) dynamic errors. Static errors such as digitization, lens distortion and measurement errors are almost unavoidable. For example, when we determine the positions of four reference points on the ground and image planes, for calculating the transformation matrix between the two planes, computation or measurement errors may occur. Such errors can be corrected by an additional homographic transformation, and may even be unnoticeable to a user in reality because of the simultaneous self-adjusting ability resulted from the visual feedback during the pointing operation. However, dynamic errors may cause obvious jitters in RPP, which are usually unacceptable. Thus, the error analysis discussed in this chapter will focus on (ii).

There are several sources of the dynamic errors, and a major one is due to noises associated with image acquisition. For example, pixels of the pointer region are identified in each of the stereo images before the PCA is performed; however, size and shape of the region may change with time because of illumination changes and influences of noise from the camera sensors[28]. In

[27] The corresponding transformations, H_L (for $I_{LX} \rightarrow R_{LX}$, $X = S, E$) and H_R, are found in advance by using positions of four reference points marked on the floor (not shown in Fig. 1), and their positions in the stereo images.

[28] Influences from more complex situations, e.g., when the pointer's color is close to the background, are not considered in this chapter since highly dynamic segmentation errors of the pointer due to pointer-background interaction may be so large that the error analysis of the RPPs will make no sense. (Similarly, extraction of reference points in the system calibration stage is also assumed to be free of such complex situations.) In general, more involved segmentation schemes will be needed to resolve such a problem, which is out of the scope of this chapter. One way of resolving such a problem is to employ special hardware in the system setup, e.g., attaching blinking LEDs [24] to the pointer.

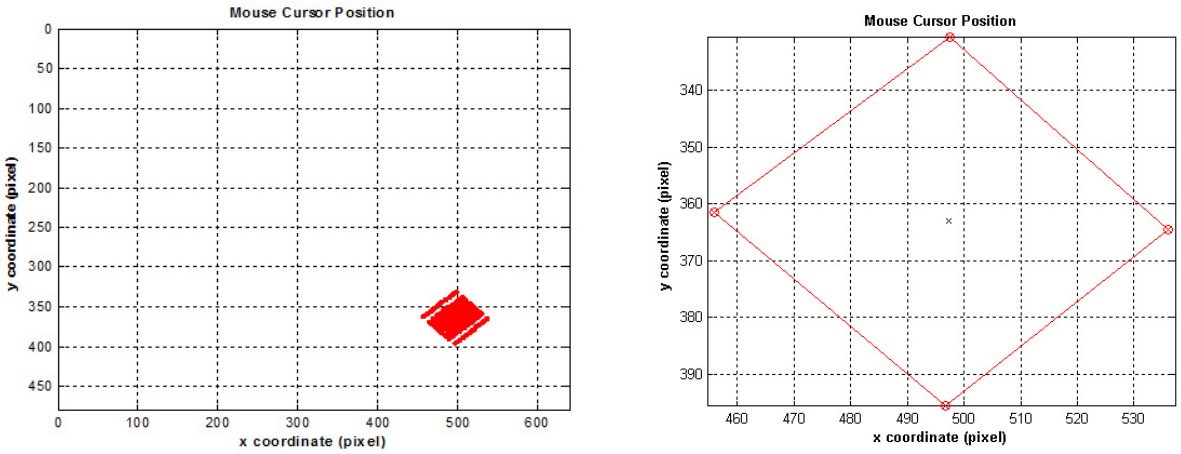
the following, some error analysis methods will be developed to investigate the influence of dynamic errors on the RPPs of the proposed systems. The goal is to correctly and efficiently identify the range of error in the position of RPP.

For the pointing system shown in Fig. 5.1, π_P, C_L and C_R are fixed in position; therefore, RPP is decided by the reconstructed planes π_L and π_R, and in turn decided by pointer endpoints I_{LS}, I_{LE}, I_{RS} and I_{RE}. The process of the extraction of these points from stereo images is often influenced by the imaging noises mentioned above. As a result, the obtained pointer endpoints are not stable, so is the calculated RPP. Thus, the deviation of the RPPs due to the variations of I_{LS}, I_{LE}, I_{RS} and I_{RE} will be the main focus of this chapter.

For a preliminary examination of the above deviation, simulated noises of unit magnitude are added to these pointer endpoints. In particular, 24 simulated points placed evenly (with 15° spacing) along "noise" circles with radius of 1 pixel are generated for I_{LS} = (188,158), I_{LE} = (247,189), I_{RS} = (159,142), and I_{RE} = (226,155) in Fig. 5.1, as shown in Fig. 5.2. In each run of the simulation, four points, each selected from one of the above four circles, are selected as endpoints of the pointer in the stereo images to reconstruct a RPP using aforementioned homographic transformations. Fig. 5.3(a) shows all 24^4 RPPs (in red), with the convex hull of them (the range of reconstruction errors) shown in Fig. 5.3(b), computed from the 24 × 4 simulated points.

In general, it is desirable to have such a range calculated more efficiently, e.g., with less simulated endpoints of the pointer. However, a direct reduction in the data size may underestimate range of reconstruction errors. For example, the blue region in Fig. 5.4 is obtained by using only 4 points (with 90° spacing) from each noise circle shown in Fig. 5.2.

From some close examinations of the relationship between the above reconstruction errors and the locations of the four simulated endpoints of the pointer obtained from Fig. 5.2, it is found that the error range is mainly due to (two) extreme values in the slopes of $\overline{I_{LS}'I_{LE}'}$ (and $\overline{I_{RS}'I_{RE}'}$). Based on such an observation, we then try to use only the contacts of the internal common tangents (CICTs) of the two noise circles in each of the stereo images (see Fig. 5.2 for such tangents). The range of reconstruction error thus obtained is also shown in Fig. 5.4 (as four points connected by black line segments). One can see that such results almost coincide with that obtained using all (24) points from each noise circle of simulated points shown in Fig. 5.2. A closer examination can be carried out by comparing the coordinates of the vertices shown in Fig. 5.4, as listed in Table 5.1. Thus, estimation of the error range from a larger number of the simulated points (24 × 4) can be replaced by using only the 8 (2 × 4) CICTs with negligible change in the estimation, and with the number of reconstructed RPPs reduced greatly (from 24^4 to 2^4).



(a) (b)

Fig. 5.3. (a) RPPs for simulated points shown in Fig. 5.2. (b) Range of reconstruction errors (with error-free reconstruction show by an ”x”).

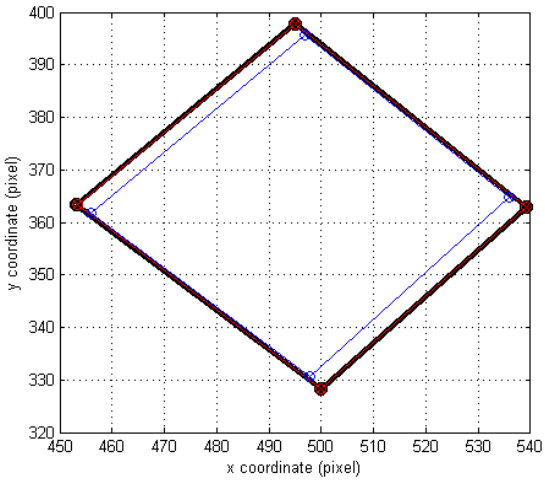


Fig. 5.4. Error range shown in Fig. 5.3(b) (red), similar range but obtained by using only 4 points (with 90◦ spacing) from each noise circle in Fig. 5.2 (blue), and error range based on internal common tangents (black, see text).

Table 5.1. Coordinates of the vertices shown in Fig. 5.4.

	x_{max}	x_{min}	y_{max}	y_{min}
(blue)	535.9857	455.9600	395.6483	330.5393
(red)	539.2251	453.4022	397.8248	328.1907
(black)	539.2422	453.2395	397.8823	328.1027

The above observations regarding CICTs of two noise circles, i.e., a RPP of a pointer from stereo images will be displaced much more when the pointer is rotated than if it’s translated with comparable amount of movements of its endpoints, can be explained with a simple example, as discussed in the following. Consider a pointing system with geometric configuration similar to that shown in Fig. 5.1, and assume the pointer is initially perpendicular to the projection plane. When the pointer is translated by k in a direction parallel to the projection plane, the RPP will be

translated by k too. However, if we fix the endpoint of the pointer which is far away from the projection plane as the center of rotation and rotate the pointer by θ degrees such that the other end of the pointer is displaced by $k = \theta r$, with r being the length of the pointer, the RPP will have a displacement of $k' > \theta d$ with d being the distance from the pointer to the projection plane. One can see that if $d >> r$, which is often the case in various pointing situations, the amount of movement of RPP with a rotated pointer is much larger than that due to a translated pointer, or $k' >> k$. Such an example reasonably explains why the estimated maximal error range (EMER) efficiently obtained using CICTs can represent the real error range with high accuracy, as the CICTs give the limits of the rotation angle of the pointer, with its end points confined to two noise circles in each of the stereo images.

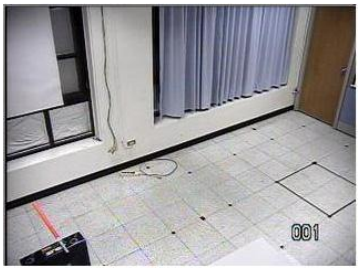
The use of unit circle for noise is only to provide a baseline for error estimation, which can in fact be adapted for specific applications. For pointing systems based on the estimation of two ends of an elongated pointer, the idea of CICTs can be generalized easily and applied to the spatial supports, regardless of their shapes[29], of the error distributions of the two points to estimate the EMER of the pointing position. Such supports can be obtained for a static pointer in each view by observing its two ends for some time.

5.4 Experiments

In order to clearly verify the validity of the EMERs with respect to actual error distributions, we focus on the static pointing situation in the experiments, i.e., we fix the pointer in space and measure the locus of RPPs. Thus, additional sources interferences, e.g., due to multi-camera synchronization and/or motion blur of a moving pointer, can be avoided. The error analysis results obtained here can be applied in the future to situations involving highly dynamic pointing situations if these interferences can be well controlled or even eliminated, e.g., via better imaging hardwares. We will first examine the proposed error estimation method by placing the pointer at a position, and pointing to a position on the projection plane. Then, pointing results obtained by selecting of a pair of cameras for each RPP according to the EMERs are compared with those obtained by using all cameras.

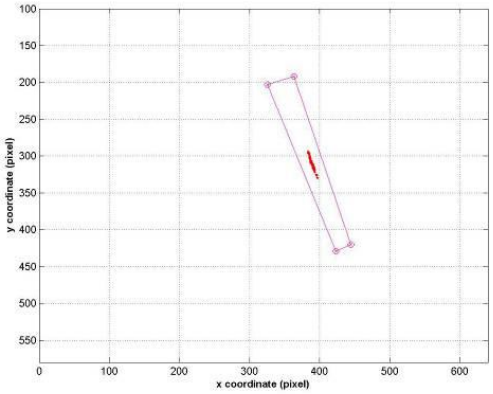
Figs. 5.5(a) and (b) show an orange stick which is fixed in the workspace and is used in the experiment as a pointer. In Fig. 5.5(c), the purple quadrilateral shows the EMER obtained for simulated 1-pixel error in point feature extraction, while the red dots are the actual positions of RPPs found by the pointing system during a period of 30 seconds. One can see that the latter is well bounded by the former.

[29] For example, error distributions can often be described by elliptical Gaussian blobs.



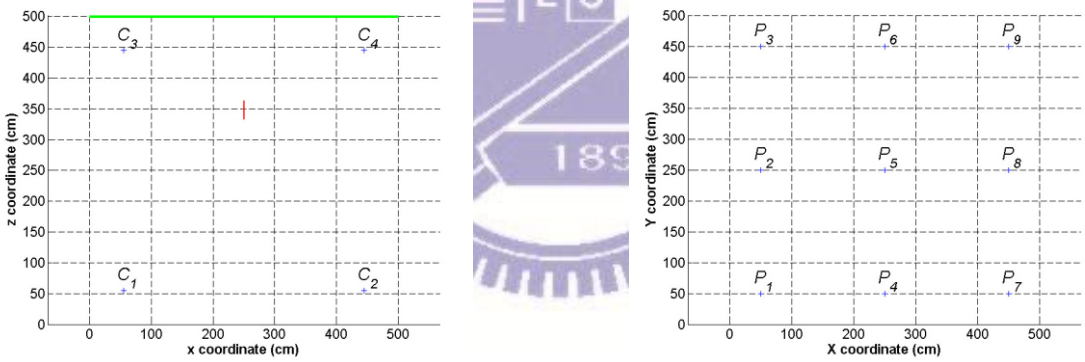
(a)

(b)



(c)

Fig. 5.5. (a) Left image. (b) Right image. (c) EMER and actual RPPs.



(a) (b)

Fig. 5.6. (a) Layout of the synthesized room. (b) Pointing positions on the projection plane.

The actual errors and estimated errors have a nice match in their distributions which are spatially highly directional. In particular, the locations of the RPPs are now distributed in a fairly narrow region, with its elongated direction well predicted by the EMER.

To further investigate the relationship between EMERs and different pointing positions, and with respect to different camera pairs, a synthesized room of size 500cm by 500cm is built. Fig. 5.6(a) shows the top view of the layout of the room. Cameras C_1, C_2, C_3 and C_4, marked as crosses, are mounted on a ceiling of height 250cm while pointing toward the center (250, 0, 250) of the room. The red line represents the pointer and the green line corresponds to the projection plane, on which a user will point to nine fixed pointing positions P_1, P_2, …, P_9 as shown in Fig. 5.6(b). The

Table 5.2. Suggestion of camera pairs.

Pointing positions	Camera pairs for smallest error range
P_1	C_3&C_4
P_2	C_3&C_4
P_3	C_1&C_3
P_4	C_3&C_4
P_5	C_1&C_4 or C_2&C_3
P_6	C_1&C_3 or C_2&C_4
P_7	C_3&C_4
P_8	C_3&C_4
P_9	C_2&C_4

resultant EMERs computed for different camera pairs are shown in Fig. 5.7. Because of the left-right symmetry of camera configuration with respect to the pointer (which is located 100cm above ground level), highly symmetrical patterns of EMERs can be observed.

The above EMERs can serve as good references for a user to select camera pairs that will achieve the highest stability in the pointing process. Table 5.2 shows such suggestion of camera pairs for each of the nine pointing positions[30], which correspond to the minimum areas of EMER. On the other hand, huge EMERs in Fig. 5.7 also indicate inappropriate camera pairs e.g., C_1&C_3 in Fig. 5.7(c) and C_2&C_4 in Fig. 5.7(d), that may result in highly unstable pointing and should be avoided. One of such EMERs of C_1&C_3 occurs while the pointer is pointed toward P_2. The problem is due to the very short pointer extracted in one of the pair of images (see Figs. 5.8(a) and (b)), which is highly sensitive to image noise and may cause huge reconstruction errors. Similar problem occurs when the pointer is pointed toward P_8 (see Figs. 5.8(c) and (d)). Note that some of EMERs shown in Fig. 5.7 are highly directional. Thus, suggestions other than those listed in Table 5.2 are possible if requirements of pointing accuracy for a particular application are not isotropic.

Fig. 5.9 shows similar experiment results obtained by moving the pointer left 150cm. The EMERs shown in Fig. 5.9 are not symmetrical due to the lack of symmetry in the geometry of system configuration. However, the huge EMER shown in Fig. 5.9(a), which does not correspond to a very short pointer in the image, as shown in Fig. 5.10, it is due to the fact that the reconstructed planes π_R and π_L are almost parallel to each other.

[30] These positions are mainly used to show that if arbitrary camera pairs are adopted for different locations on the projection plane, the resultant RPPs may be too unstable to be useful. For other locations, the trend of RPP stability may be estimated via interpolation, which is omitted for brevity. On the other hand, since the proposed CICT-based error analysis is extremely efficient, the EMER, as well as the preferred camera pair, may be estimated on the run, as the pointer ends are extracted, for arbitrary RPP and user (and pointer) locations. The above arguments can also be applied to the next set of experiments which use selected (200) pointer locations to show that using unit circles is as good as using more precise (often smaller) circles to simulate noises in terms of helping the user to avoid camera pair(s) of worst stability performance.

Table 5.3. Pointing errors of the two methods for the pointer placed at (250, 100, 350).

Pointing position	Mean error (standard deviation)cm	
	Our method	[31]
P_1	4.04(2.04)	4.16(2.32)
P_2	5.96(3.78)	6.35(3.66)
P_3	13.35(8.14)	24.48(11.73)
P_4	1.67(0.92)	1.58(0.87)
P_5	5.09(2.91)	4.33(2.21)
P_6	7.64(4.02)	7.10(3.50)
P_7	4.14(2.36)	3.79(2.31)
P_8	6.70(4.23)	7.40(4.09)
P_9	12.90(8.44)	24.74(10.09)

A comparison to using all cameras

While the goal of the proposed error analysis is to identify one of camera pairs that will result in best pointing performance in terms of pointing stability, the underlying assumption is that when highly unstable RPPs are reconstructed with data obtained from using all cameras, the problem can be alleviated by not using inappropriate cameras (or camera pairs) if possible. For example, if more than two cameras are used for the pointing system shown in Fig. 5.1, the proposed approach will choose two cameras to find the RPP, while a least square solution of RPP can be found by using all cameras, as in [31].[31] To verify the above assumption, additional experiments are conducted for the simulation environment described in Fig. 5.6, with additive noises.

Table 5.3 shows pointing errors generated by (i) the proposed method which selects a camera pair for each pointing position (from P_1 to P_9) according to Table 5.2[32] and (ii) the least square approach discussed in [31] which uses all cameras.[33] One can see that similar pointing accuracy (within 0.76cm) can be achieved by both (i) and (ii) for all pointing positions, except for P_3 and P_9 which correspond to the largest pointing error on the average for both methods. Intuitively, one would expect that most unstable pointing results will be generated for these two points, as shown in Fig. 5.11(a), since they correspond to the smallest angle between the pointer and the projection plane. Note that up to 47% reduction (from 24.74cm to 12.90cm) in mean pointing error can be

[31] For [31], the pointing direction is defined by the hand-head line, and the RPP is obtained as the least square solution of the intersection of projections of this line on the projection plane from all cameras. If there are only two cameras, as shown in Fig. 1, the two approaches will generate identical RPP.

[32] For P_5 (P_6), C_2&C_3 (C_2&C_4) are selected.

[33] To ensure a fair comparison, the two end points of the pointer adopted in our system for error analysis are used to define the pointing line in each camera view for both (i) and (ii).

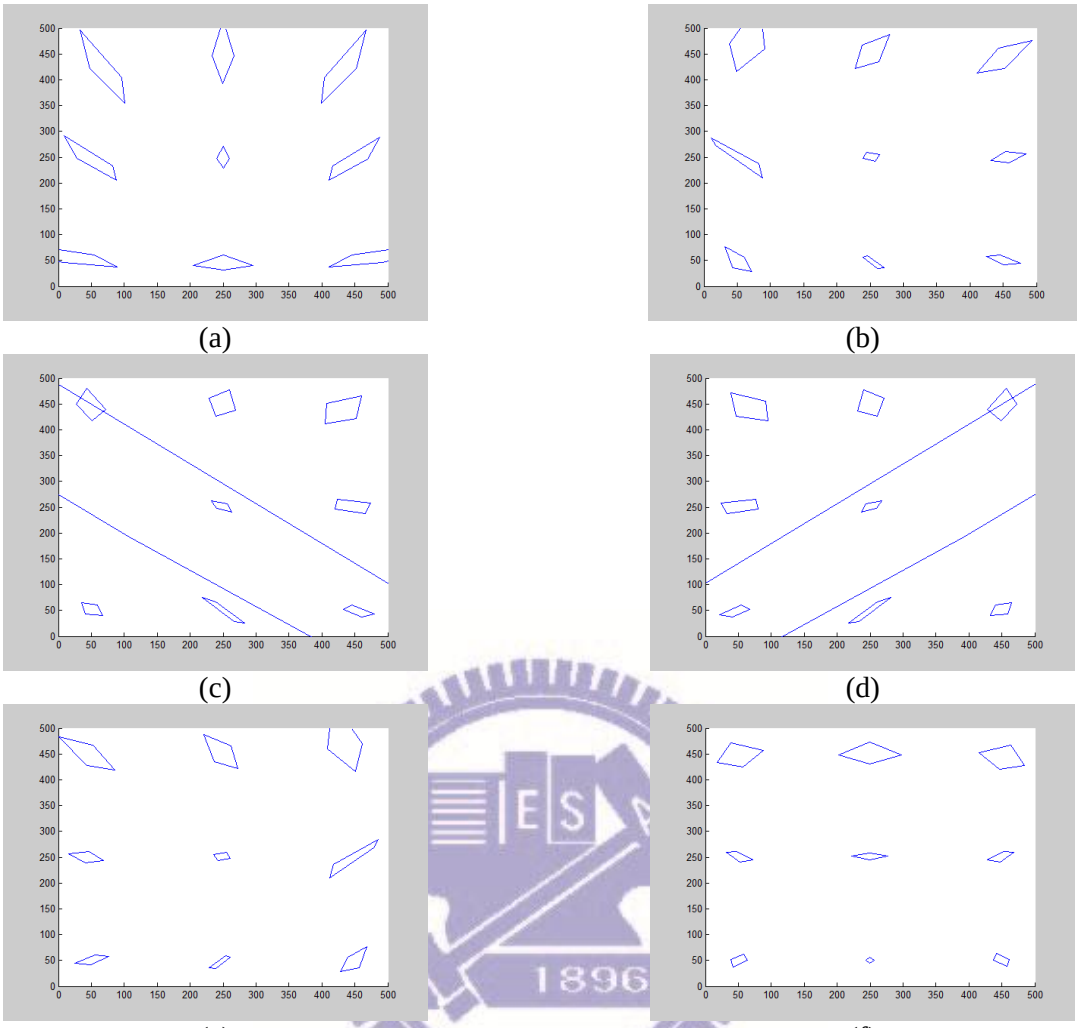


Fig. 5.7. Estimated maximal error ranges for different camera pairs: (a) C_1&C_2. (b) C_2&C_3. (c) C_1&C_3. (d) C_2&C_4. (e) C_1&C_4. (f) C_3&C_4.

Table 5.4. Pointing errors of the two methods for the pointer placed at (100, 100, 350).

Pointing position	Mean error (standard deviation)cm	
	Our method	[31]
P_1	2.12(1.11)	3.11(1.14)
P_2	4.28(2.80)	4.84(3.04)
P_3	9.43(6.82)	11.39(7.28)
P_4	2.67(1.82)	4.11(2.05)
P_5	3.40(1.71)	6.63(3.38)
P_6	5.99(3.04)	14.40(8.64)
P_7	7.02(5.10)	14.86(6.97)
P_8	7.23(4.75)	11.51(5.46)
P_9	10.87(5.07)	19.01(13.72)

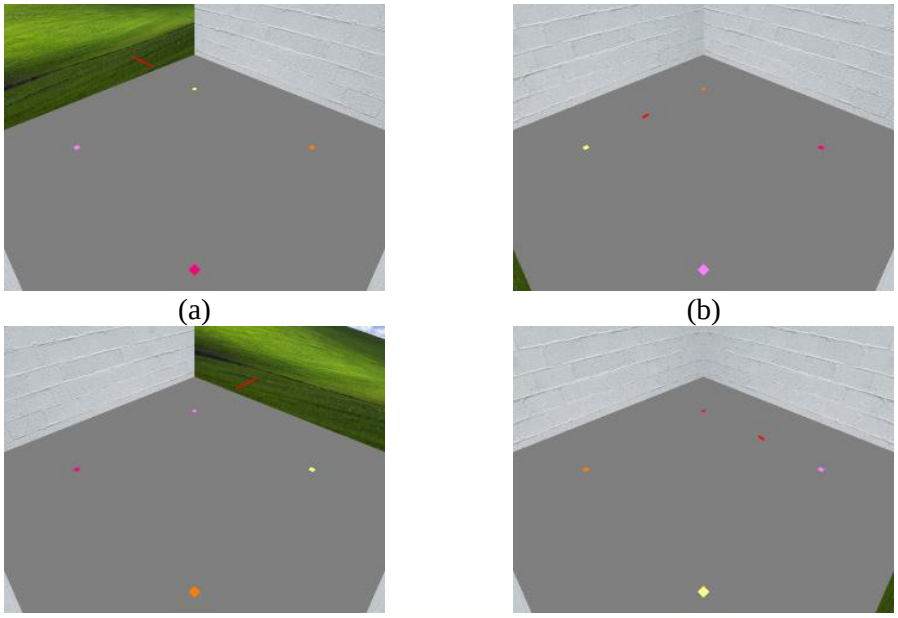


Fig. 5.8. (a) Image captured by C_1 when the pointer is pointing toward P_2. (b) Image captured by C_3 when the pointer is pointing toward P_2. (c) Image captured by C_2 when the pointer is pointing toward P_8. (d) Image captured by C_4 when the pointer is pointing toward P_8.

achieved with the proposed camera selection scheme for these worst case scenarios.

Additional observations can be made for more general system configurations wherein the pointer is moved left by 150cm from that specified above, as shown in Table 5.4. Unlike the nearly symmetric pattern shown in Fig. 5.11(a), the corresponding distributions of the RPPs shown in Fig. 5.11(b) are not symmetric since the camera locations are no longer symmetric with respect to the pointer position. Again, more than 40% reduction (from 19.01cm to 10.87cm) in mean pointing error can be achieved for the worst case situation with the proposed approach compared with the least square one. The above results suggest that the camera selection scheme based on the efficient error analysis proposed in this chapter can indeed help the pointing accuracy and stability.

5.5 Summary

In this chapter, a simple and real-time pointing system is implemented so that the pointing error can be examined closely. A pointer with bright color is used in the pointing process to reduce the complexity in extracting its direction in an image, and error ranges in the pointing position are estimated by synthetic image noises. To greatly increase the efficiency of the estimation, a fast analysis method is developed which only utilizes an extremely limited subset of noise data. With the help of such analysis, suitable operation positions may be suggested to a user of similar

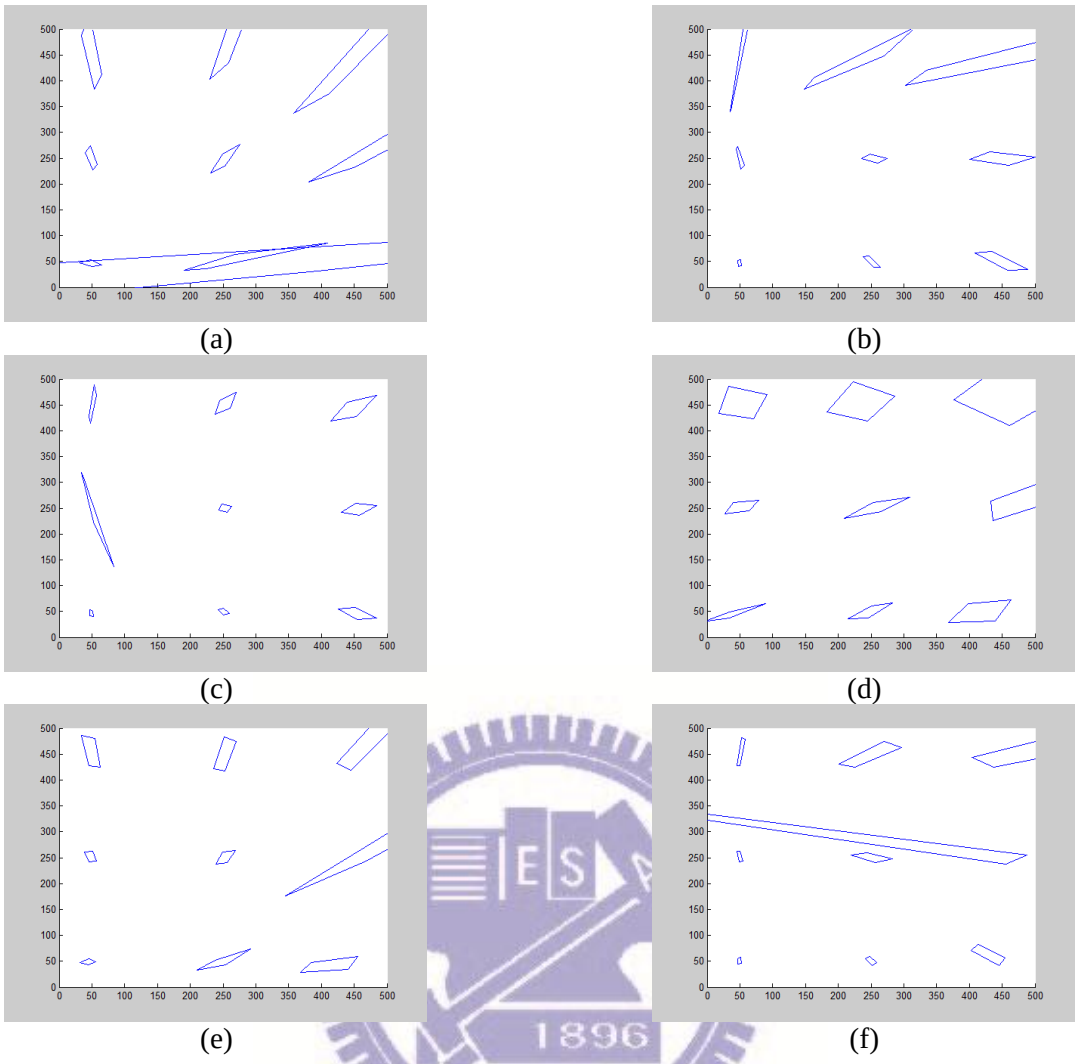
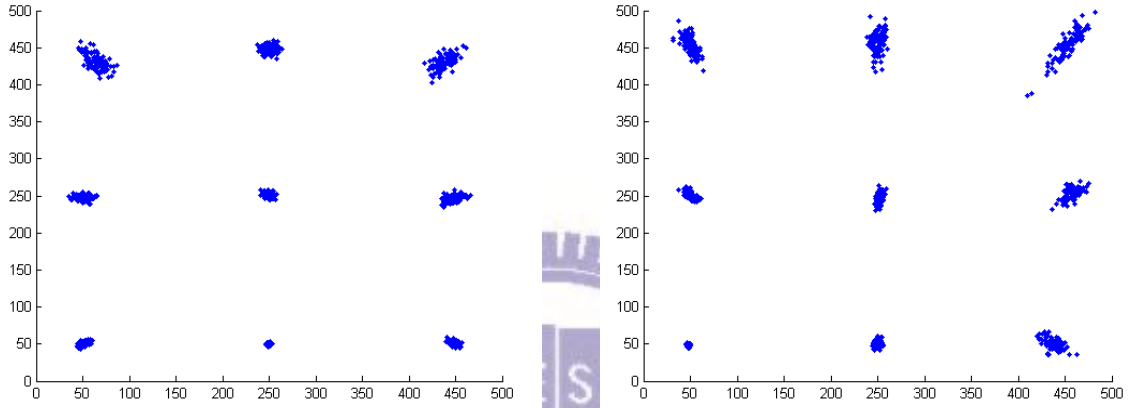


Fig. 5.9. Estimated maximal error ranges for the pointer moved left 150cm for different camera pairs: (a) C_1&C_2. (b) C_2&C_3. (c) C_1&C_3. (d) C_2&C_4. (e) C_1&C_4. (f) C_3&C_4.

pointing systems if the pointer can be used in different locations in a 3D workspace. Moreover, in a multi-camera environment, the overall pointing operation can achieve smallest error ranges, and most stable pointing results, by automatically selecting a pair of cameras based on the proposed error analysis scheme. While experiments are conducted and studied in this chapter for static pointing situations, the proposed approach is applicable to more dynamic situations, e.g., in applications wherein instructions are given via various trajectories of pointing positions. However, the error analysis method cannot be applied directly to our people localization system because the line correspondences between different views are unknown. Further investigation is needed to address such an issue.

(a) (b)

Fig. 5.10. Image captured by C_1 when the pointer is pointing toward P_7. (b) Image captured by C_2 when the pointer is pointing toward P_7.



(a) (b)

Fig. 5.11. Distribution of RPPs of the nine pointing positions for the pointer placed at (a) (250, 100, 350) and (b) (100, 100, 350).

Chapter 6
Conclusions and future works

In this thesis, three people localization methods using the vanishing points of vertical lines and binary foreground regions are proposed. The vanishing points are used to generate 2D line samples of foreground regions in multiple views efficiently. These 2D line samples can provide sufficient information (evidence) and enable the generation of 3D line samples for potential people locations. Additionally, a grid-based footstep analysis, followed by 3D line sampling, is proposed to find potential people locations. Thus, the costly 3D reconstruction can be avoided while the computation speed can be improved. Furthermore, to improve the efficiency of the first method, a refinement procedure of 3D line samples associated with geometric rules is proposed to filter out invalid 3D line samples very efficiently. Therefore, people localization can be achieved in real-time without using special hardware. However, a lot invalid 3D line samples are also reconstructed and processed further since the correspondence of line sample pairs between different views is unknown. To alleviate such a problem, a line correspondence measure of 2D line samples is proposed and applied to filter out non-corresponding 2D line sample pairs before the 3D reconstruction stage. Because more than 90% 2D line sample pairs can be filtered out, the computation efficiency is improve significantly. Finally, we propose an error analysis of 3D line reconstruction method to improve the accuracy of line-based pointing systems, which is expected to help the improvement of the accuracy of the proposed people localization methods in the future.

Appendix A
The derivation of multiple homographic matrices for planes of different heights

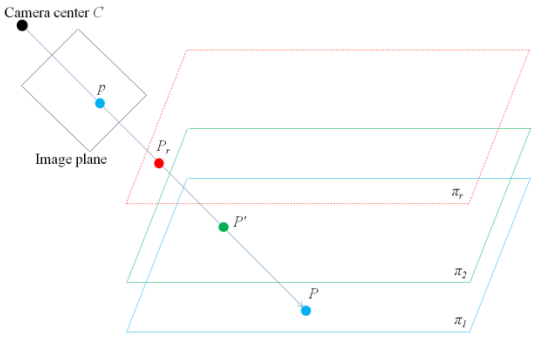


Fig. A.1. Illustration of calculation a reference point on π_r.

Homographic matrices are required for projecting 2D line samples onto the reference plane, as in Subsection 2.1.2. Also, the homographic matrices of multiple reference planes at different heights can be used to back-project points on a reference plane to different views for the computation of AFCR, as in Subsection 2.2.1.

In [23], [24], the authors use four vertical calibration pillars placed in the scene, with marker points at three known heights on each of them, to establish the homographies between image planes and reference planes at desired heights. Since a new reference point at any height along a pillar can be identified in the images of interest using the cross-ratio along that pillar, the above homographic relationship can actually be established for planes at arbitrary height. Thus, twelve (4 × 3) marker points are required for calculating all homographic matrices.

Instead of using twelve marker points, an approach for the derivation of multiple homographic matrices for planes of different heights, which only use eight (4 × 2) marker points on four pillars, is presented in the following. Assume each pillar has two marker points at planes π_1 and π_2 with heights h_1 and h_2, respectively. First, four marker points with height h_2 are used to calculate a homographic matrix H_{m2} between the image plane and the reference plane π_2 as shown in Fig. A.1. Then, we will produce four reference points on π_2 by projecting the four marker points with height h_1, respectively. More specifically, the image point p corresponding to the marker point P can be projected to π_2 by H_{m2} to obtain the world coordinate of P' as shown in Fig. A.1. After that, we can calculate a new reference point P_r on an arbitrary imaginary plane π_r with a specified height by calculating the intersection of PP' and π_r. Similarly, the rest three marker points with height h_1 can be used to produce another three new reference points on π_r. Finally, a homographic matrix H_{mr} can be found by using the four new reference points. By adopting such a method, we can produce a set of homographic matrices for reference planes of various heights using only eight marker points.

Appendix B
Setting the parameters

In Subsection 4.4.1, satisfactory results of people localization are obtained with the proposed approach for selected values of some parameters. We will show that it is not too hard to set these parameters properly in practice for different scenes. Table B.1 shows a list of such parameters together with the section (and subsections) in which each of them is used, and range of values tested for each of them. While the first three parameters are applied before and after the refinement process, in both 2.2.1 and 3.1.2, the rest are applied in Subsections 2.2.1, 2.2.2 and Section 4.2 As for their physical meanings, five of them are for measurements in the 3D scene (in cm), one of them is based on percentage values, two of them are for number counts, and the last one is for measurements in 2D image planes (in pixel).

In general, for satisfactory performance of the proposed localization approach, proper values should be assigned to the above parameters for each scene, or camera configuration. In Table B.1, appropriate value ranges, which yield reasonable localization results for S1-S3 taken from the indoor scene considered in Sec. 4.3, are listed for these parameters[34]. In particular, Figs. B.1-B.5 show such results, only for the most complicated S3 (with four views) for brevity, for the most important five parameters[35]. For each of the five figures, only one parameter is adjusted for easy observation of the trend of localization performance, which has fairly low sensitivity to the adjustment, with the parameter value used in Table I indicated by an arrow.

For recall and precision rates shown in these figures, significant changes (still within ±2.4% of that in Table B.1) mainly exist at one end of each plot except for Fig. B.1(a) and Fig. B.4(a). Besides, the plots of recall and precision rates are intersected at one point in each figure. For example, threshold T_C in Fig. B.4(a) specifies the maximum distance between two 3D line samples that can be grouped into the same group. If T_C is too small, a group corresponding to a person may be split into several groups, resulting in poor precision rate due to a lot of false positives. In contrast, if T_C is too larger, the recall rate tends to decrease due to miss detections resulted from incorrectly merged groups.

As for localization errors, variations caused by adjusting these parameters are fairly small, i.e., within ±0.50cm, except for Fig. B.1(b). Small variations in computation speed can also be found in these figures, except for Fig. B.1(c) and Fig. B.3(c). For Fig. B.3(c), it is easy to see that the computation time is directly related to the number of sample points of a 3D line sample which need to be verified against image foregrounds.

Overall, threshold T_{len}, which specifies the minimum length of a 3D line sample which should be covered by foreground regions in all views, seems to be most influential. While increasing its value to remove more (possibly incorrect) 3D line samples will always reduce the computation time, the precision/recall rates and localization accuracy will increase monotonically, up to 9% and 1.1cm in

[34] In all experiments in Sec. 4.3, T_P is arbitrarily chosen as 24 (pixels), i.e., 10% of the height of the input image. In practice, the T_P can be decide by human size in the image.

[35] The rest three parameters are associated Geometric Rules 2 to 4, respectively. For their ranges of values listed in Table B.1, recall and precision rates are basically the same as those listed in Table I. The screening with all the three rules, on the other hand, does increase the computation speed by 17%.

Table B.1. Recommended value ranges of parameters for S1-S3.

Subsection used	Parameter	Function/description	Value range
Subsection 2.2.1, 4.2	T_{len}	Minimum length of a 3D line sample. (Geometric Rule 1)	[100cm, 150cm]
Subsection 4.2	T_{tl}	Minimum height of a 3D line sample. (Geometric Rule 2)	[70cm, 130cm]
Subsection 2.2.1, 4.2	T_b	Minimum height of bottom of a 3D line sample. (Geometric Rule 3)	[70cm, 105cm]
Subsection 4.2	T_{th}	Maximum height of a 3D line sample. (Geometric Rule 4)	[190cm, 230cm]
Subsection 2.2.1	T_{fg}	Minimum AFCR of a 3D line sample.	[0.68, 0.97]
Subsection 4.2	N_{plane}	Number of reference planes.	[10, 45]
Subsection 2.2.2	T_c	Maximum distance between 3D line samples of a group.	[15cm, 40cm]
Subsection 2.2.2	N_{line}	Minimum number of 3D line samples of a group.	[1, 11]
Subsection 2.2.1	T_p	Minimum number of foreground pixels of a 2D line sample.	see text

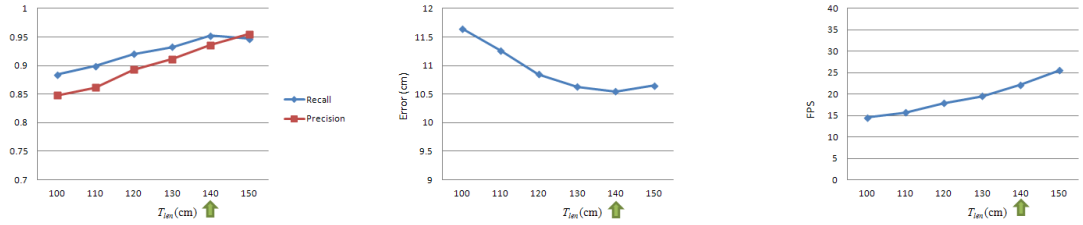
Table B.2. Parameter values selected for experiments presented in Sec. 4.3.

	T_{len}	T_{tl}	T_b	T_{th}	T_{fg}	N_{plane}	T_c	N_{line}
S1-S3	140	90	90	230	0.85	36	25	4
S4-S5	110	130	70	190	0.92	36	25	7

Fig. B.1, respectively, as its value is increased from 100cm to 140cm.

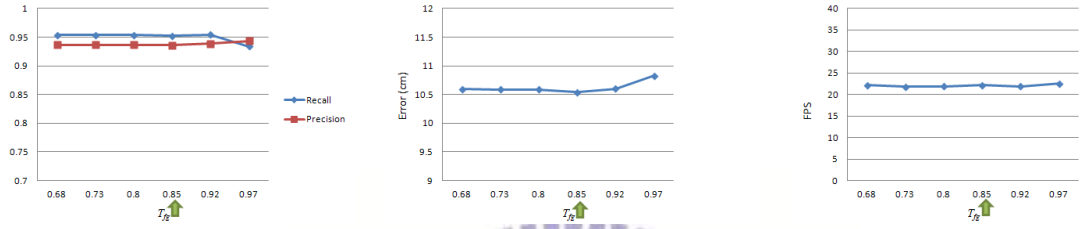
In practice, different values of all these parameters may need to be selected for different scenes and camera configurations. Table B.2 shows the two sets of (mostly different) parameter values selected for the indoor scene (for S1-S3) and the outdoor scene (for S4-S5) considered in Sec. 4.3. One can see that the values used for the latter are not far from the corresponding value ranges recommended in Table B.1 for the former. In general, once their values are determined, the algorithm will work consistently for the scene under consideration[36]. For example, Figs. B.6 and B.7 show testing results similar to Fig. B.1, but for sequences S1 and S2, respectively. One can see that good localization results can also be obtained with T_{len} = 140cm.

[36] As for automatic determination of appropriated parameter values, different approaches are currently under investigation. For example, by examining these three figures, it seems that it will be not necessary to consider larger values of T_{len} either (i) when the recall rate drops or (ii) when the mean localization error increases.



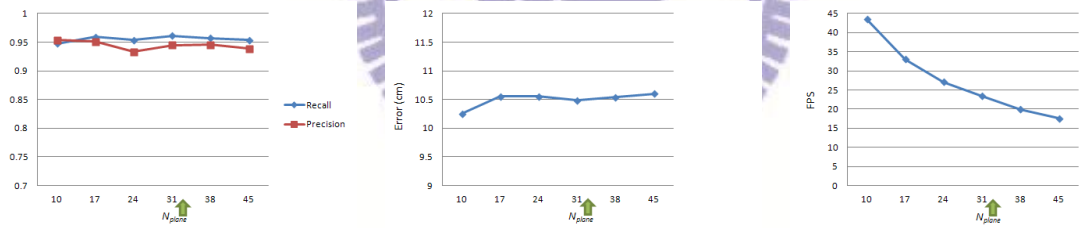
(a) (b) (c)

Fig. B.1. Results of using different values of T_{len}. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



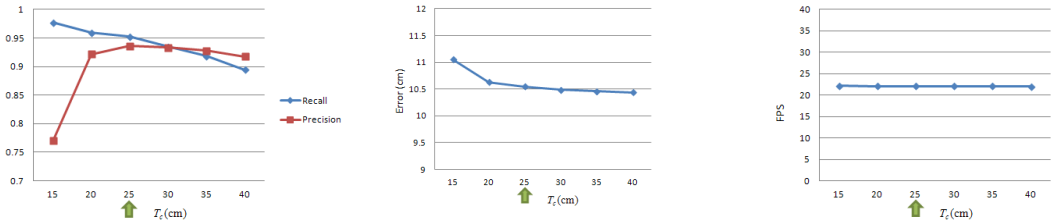
(a) (b) (c)

Fig. B.2. Results of using different values of T_{fg}. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



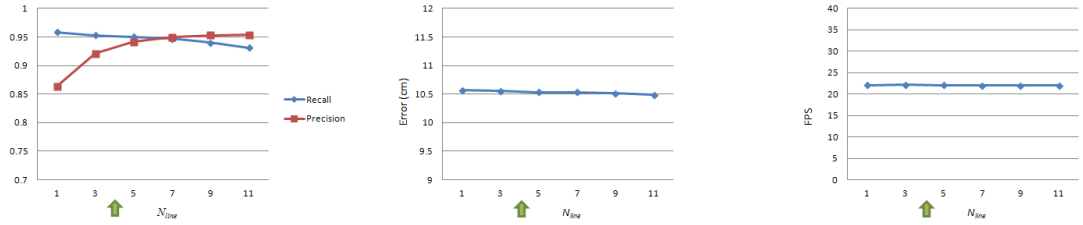
(a) (b) (c)

Fig. B.3. Results of using different values of N_{plane}. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



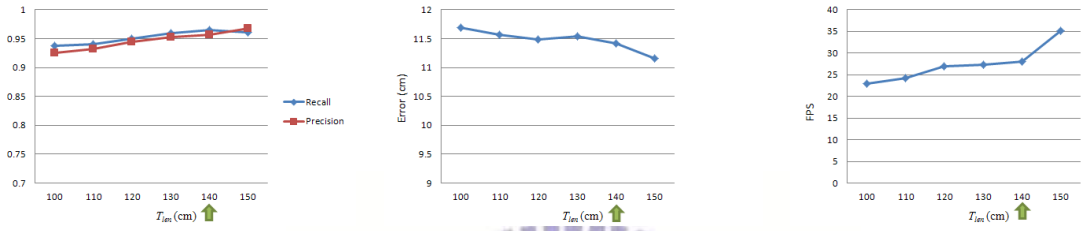
(a) (b) (c)

Fig. B.4. Results of using different values of T_c. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



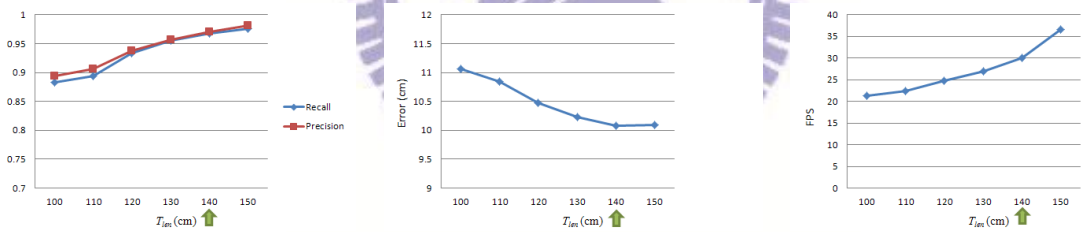
(a) (b) (c)

Fig. B.5. Results of using different values of N_{line}. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



(a) (b) (c)

Fig. B.6. Results of using different values of T_{len} for S1. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.



(a) (b) (c)

Fig. B.7. Results of using different values of T_{len} for S2. (a) Recall and precision. (b) Mean localization error. (c) Computation speed.

Appendix C
Two types of synergy maps

For better understanding of the effects of our implementation of [25], synergy maps created by (i) foreground likelihood maps used in [25] and (ii) the binary version (foreground regions used in this thesis) of (i) are both generated. Figs. C.1(a)-(d) show foreground likelihood maps obtained for Figs. 3.6(a)-(d), respectively. Even with pixels of lower likelihood filtered out, these foreground maps are still influenced greatly by the cluttered background with strong shadows and reflections. Figs. C.1(e) and (f) show synergy maps generated by (i) and (ii), respectively. One can see the positions with high occupancy likelihoods, which are also very close to the ground truth (marked as white crosses), are quite similar for these two types of synergy maps.

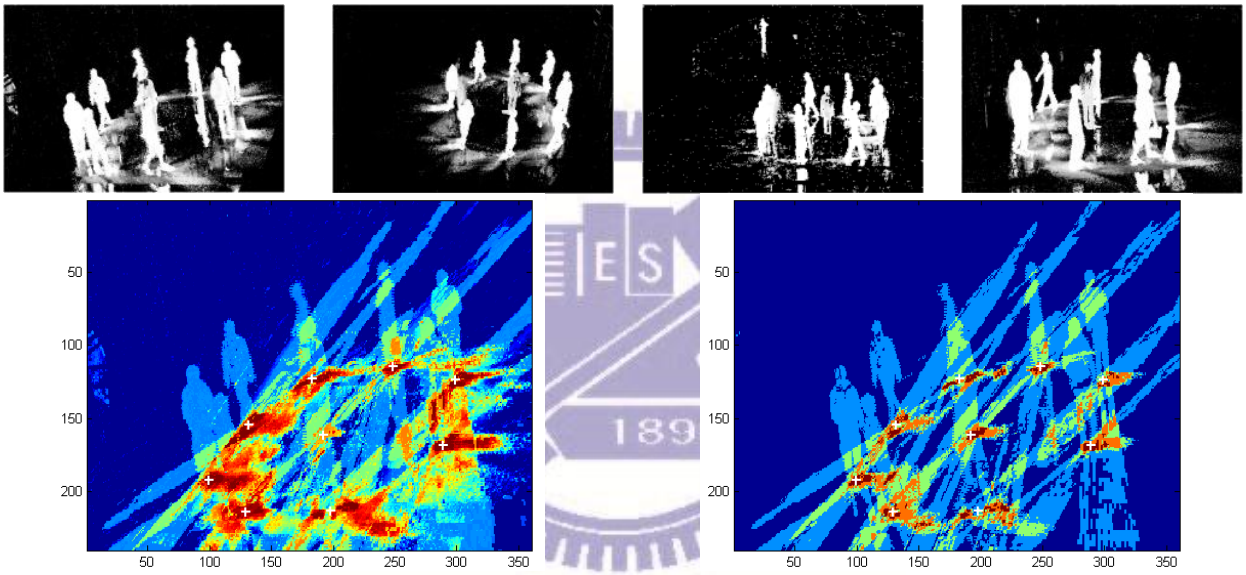


Fig. C.1. (a)-(d) Foreground likelihood maps. (e) The synergy map used in [25]. (f) The synergy map obtained by using binary foreground images.

Appendix D
The preprocessing step

The objective of this step is to extract the region of the pointer, analyze its orientation, and locate its two endpoints in an image. The pixels belonging to the region can be found by measuring similarities of the specified color distributions[37] which are obtained in advance. The measurement is achieved by thresholding in HSI color space to find out the pointer while avoiding the interference of the light changes. The pointer detection result of Fig. D.1(a) is shown in Fig. D.1(b). One can see that the pixels of the pointer do connect to each other and occupy a sufficient and elongated area. According to such observations, the connected component labeling is used to identify connected regions, and the region which has largest elongated area is selected as the region of the pointer. After that, principal components analysis is used to find its two axes. Assume a connected region which has n points is represented as $X = [x_1, x_2, \ldots, x_n]^T$. The mean value of the connected region is represented as $m = (\sum_{i=1}^{n} x_i)/n$. The covariance matrix S can be calculated by $\sum_{i=1}^{n}(X-m)(X-m)^T$. Next, the eigenvalues and eigenvectors can be found by eigen decomposition. The eigenvector corresponding to the largest eigenvalue can then be used to calculate a best fit line passing through the pointer. Finally, the two intersection points of (i) this line and (ii) the bounding box of the connected region will be defined as the two endpoints of the pointer.

(a) (b)

Fig. D.1. (a) An input image. (b) The detected pointer and its bounding box.

[37] The color distributions of a pointer are measured under several light sources. In our experiments, the measured color distributions are H: 340° ~ 20°, S: 0.5 ~ 0.9 and I: 0.35 ~ 0.7. In addition, in order to obtain a complete pointer region without many holes, we release the threshold as H: 300° ~ 40°, S: 0.2 ~ 1.0 and I: 0.3 ~ 1.0. In general, if the color is not changed suddenly and can be correctly detected by the assigned color distributions at an initial stage, the color distributions can be updated and utilized continuously.

Appendix E
Reconstruction of pointing points by homographic transformations

In order to find the pointing positions, 3D coordinates of R_{LS}, R_{LE}, R_{RS}, and R_{RE} are needed. These coordinates can be calculated from the above endpoints in the stereo images by using 3 × 3 homographic matrices, namely H_L and H_R, which can provide transformations of homogeneous coordinates between the image planes and the ground plane shown in Fig. 5.1. For example, given $I_{LS} = [u, v]^T$ and $I_{LE} = [u', v']^T$, we can obtain the 2D coordinates of $R_{LS} = [x, y]^T$ and $R_{LE} = [x', y']^T$ on the ground plane as

$$[x, y, 1]^T = H_L [u, v, 1]^T \tag{E.1}$$

and

$$[x', y', 1]^T = H_L [u', v', 1]^T, \tag{E.2}$$

respectively. Similarly, R_{RS} and R_{RE} can be found by H_R.

Next, we need to find the 3D plane equations of π_L and π_R, with the former being determined by C_L, R_{LS}, and R_{LE}, and the latter being determined by C_R, R_{RS}, and R_{RE}. Let π_L, π_R and π_P, be represented by equations

$$\alpha_L X + \beta_L Y + \gamma_L Z = \eta_L , \tag{E.3}$$

$$\alpha_R X + \beta_R Y + \gamma_R Z = \eta_R , \tag{E.4}$$

and

$$\alpha_P X + \beta_P Y + \gamma_P Z = \eta_P , \tag{E.5}$$

respectively, the pointing point P can then be calculated as

$$P = \begin{bmatrix} \alpha_L & \beta_L & \gamma_L \\ \alpha_R & \beta_R & \gamma_R \\ \alpha_P & \beta_P & \gamma_P \end{bmatrix}^{-1} \begin{bmatrix} \eta_L \\ \eta_R \\ \eta_P \end{bmatrix}. \tag{E.6}$$

Bibliography

[1] Q. Cai and J. K. Aggarwal, ”Automatic tracking of human motion in indoor scenes across multiple synchronized video streams,” in *Proc Int. Conf. on Computer Vision*, pp. 356–362, Jan. 1998.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, ”W4: A real time system for detecting and tracking people,” in *Proc IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 962, Jun. 1998.

[3] S. Khan and M. Shah, ”Tracking people in presence of occlusion,” in *Proc. Asian Conference on Computer Vision*, pp. 1132–1137, Jan. 2000.

[4] M. Isard and A. Blake, ”Condensation–conditional density propagation for visual tracking,” in *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[5] K. Nummiaro, E. Koller-Meier, and L. Van Gool, ”An adaptive colorbased particle filter,” in *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[6] H. Wang, D. Suter, K. Schindler, and C. Shen, ”Adaptive object tracking based on an effective appearance filter,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1661–1667, Sep. 2007.

[7] J. Fan, W. Xu, Y. Wu, and Y. Gong, ”Human tracking using convolutional neural networks,” *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.

[8] D. Beymer and K. Konolige, ”Real-time tracking of multiple people using stereo,” in *Proc. IEEE Frame Rate Workshop*, Sep. 1999.

[9] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb, ”Plan-view trajectory estimation with dense stereo background models,” in *Proc Int. Conf. on Computer Vision*, pp. 628–635, Jul. 2001.

[10] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, ”Integrated person tracking using stereo, color, and pattern detection,” *International Journal of Computer Vision*, vol. 37, no. 2, pp. 175–185, Jun. 2000.

[11] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones, ”A multi-agent framework for visual surveillance,” in *Proc. Int. Conf. on Image Analysis and Processing*, pp. 1104–1107, Sep. 1999.

[12] A. Mittal and L. Davis, ”M2 Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene,” in *International Journal of Computer Vision*, vol. 51, no. 3, pp. 189–203, Feb./Mar. 2003.

[13] T.-H. Chang and S. Gong, ”Tracking multiple people with a multicamera system,” in *Proc. IEEE Workshop Multi-Object Tracking*, pp. 19–26, Jul. 2001.

[14] A. Utsumi, H. Mori, J. Ohya, and M. Yachida, ”Multiple-human tracking using multiple cameras,” in *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 498–503, Apr. 1998.

[15]H. Tsutsui, J. Miura, and Y. Shirai, ”Optical flow-based person tracking by multiple cameras,” in *Proc. Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 91–96, Aug. 2001.

[16] P. Kelly, A. Katkere, D. Kuramura, S. Moezzi, and S. Chatterjee, ”An architecture for multiple perspective interactive video,” in *Proc ACM Int. Conf. on Multimedia. ACM*, pp. 201–212, Nov. 1995.

[17] Q. Cai and J. K. Aggarwal, ”Tracking human motion in structured environments using a distributed-camera system,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241–1247, Nov. 1999.

[18] S. Khan and M. Shah, ”Consistent labeling of tracked objects in multiple cameras with overlapping fields of view,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355–1360, Oct. 2003.

[19] S. Khan, O. Javed, and M. Shah, ”Tracking in uncalibrated cameras with overlapping field of view,” in *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, Dec. 2001.

[20] S. Sun, H. Lo, H. Lin, Y. Chen, F. Huang, and H. Liao, ”A multi-camera tracking system that can always select a better view to perform tracking,” in *Proc. APSIPA Annual Summit and Conference*, pp. 373–379, Oct. 2009.

[21] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, ”Principal axisbased correspondence between multiple cameras for people tracking,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 663–671, Apr. 2006.

[22] L. Sun, H. Di, L. Tao, and G. Xu, ”A robust approach for person localization in multi-camera environment,” in *Proc. Int. Conf. on Pattern Recognition*, pp. 4036–4039, Aug. 2010.

[23] R. Eshel and Y. Moses, ”Homography based multiple camera detection and tracking of people in a dense crowd,” in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.

[24] R. Eshel and Y. Moses, ”Tracking in a dense crowd using multiple cameras,” *International Journal of Computer Vision*, vol. 88, no. 1, pp. 129–143, May. 2010.

[25] S. M. Khan and M. Shah, ”Tracking multiple occluding people by localizing on multiple scene planes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, Mar. 2009.

[26] K.-H. Lo and J.-H. Chuang, ”Vanishing point-based line sampling for efficient axis-based people localization,” in *Proc. IEEE Int. Conf. on Image Processing*, pp. 529–532, Sep. 2011.

[27] C. Stauffer and W. E. L. Grimson, ”Adaptive background mixture models for real-time tracking,” in *Proc IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, Jun. 1999.

[28] H.-H. Lin, J.-H. Chuang, and T.-L. Liu, ”Regularized background adaptation: A novel learning rate control scheme for Gaussian mixture modeling,” IEEE Transactions on Image Processing, vol. 20, no. 3, pp. 822–836, Mar. 2010.

[29] J.-S. Liu and J.-H. Chuang, “A geometry-based error estimation for cross-ratios,” Pattern recognition, vol. 35, no. 1, pp. 155–167, 2002.

[30] S. Sato and S. Sakane, “Interactive hand pointer that projects a mark in the real work space,” *Transactions of the Institute of Electrical Engineers of Japan*, Vol. 121-C, 2001, pp. 1464-1470.

[31] C. Colombo, A. D. Bimbo, and A. Valli, “Visual capture and understanding of hand pointing actions in a 3-D environment,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 33, 2003, pp. 677-686.

[32] Y.-P. Hung, Y.-S. Yang, Y.-S. Chen, I.-B. Hsieh, and C.-S. Fuh, “Free-hand pointer by use of an active stereo vision system,” in *Proceedings of International Conference on Pattern Recognition*, 1998, pp. 1244-1246.

[33] E. Hosoya, H. Sato, M. Kitabata, H. Nojima, and A. Onozawa, “Arm-pointer: 3D pointing interface for real-world interaction,” in *Proceedings of the European Conference on Computer Vision Workshop on Human Computer Interaction*, 2004, pp. 72-82.

[34] P. Matikainen, P. Pillai, L. Mummert, R. Sukthankar, and M. Hebert, “Prop-free pointing detection in dynamic cluttered environments,” in *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2011, pp. 374-381.

[35] K. Nickel and R. Stiefelhagen, “Visual recognition of pointing gestures for human-robot interaction,” *Image and Vision Computing*, Vol. 25, 2007, pp. 1875-1884.

[36] Y. Guan and M. Zheng, “Real-time 3D pointing gesture recognition for natural HCI,” in *Proceedings of World Congress on Intelligent Control and Automation*, 2008, pp. 2433-2436.

[37] J.-H Chuang, J.-H. Kao, H.-H. Lin, and Y.-Ting Chiu, “Practical error analysis of cross-ratio-based planar localization,” in *Proceedings of Pacific Rim Symposium on Image Video and Technology*, 2007, pp. 727-736.

[38] N. X. Dao, B. J. You, and S. R. Oh, “Visual navigation for indoor mobile robots using a single camera,” in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2005, pp. 1992-1997.

[39] J.-S. Liu and J.-H. Chuang, “A geometry-based error estimation for cross-ratios,” *Pattern Recognition*, Vol. 35, 2002, pp. 155-167.

[40] E. Krotkov, “Mobile robot localization using a single image,” in *Proceedings of International Conference on Robotics and Automation*, 1989, pp. 978-983.

[41] S. Se, D. Lowe, and J. Little, “Local and global localization for mobile robots using visual landmarks,” in *Proceedings of International Conference on Intelligent Robots and Systems*, 2001, pp. 414-420.

[42] S. D. Blostein and T. S. Huang, “Error analysis in stereo determination of 3-D point positions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, 1987, pp. 752-765.

[43] L. Matthies and S. Shafer, “Error modeling in stereo navigation,” *IEEE Journal of Robotic and Automation*, Vol. 3, 1987, pp. 239-248.

[44] J. N. Sanders-Reed, "Error propagation in two-sensor 3D position estimation," *Optical Engineering*, Vol. 40, 2001, pp. 627-636.

[45] N. Georis, M. Petrou, and J. Kittler, "Error guided design of a 3D vision system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, 1998, pp. 366-379.

[46] J. Hopcroft and R. Tarjan, "Efficient algorithm for graph manipulation," *Communications of the ACM*, Vol. 16, No. 6, pp. 372-378, Jun. 1973.

[47] K.-H. Lo, C.-J. Wang, J.-H. Chuang, and H.-T. Chen, "Acceleration of vanishing point-based line sampling scheme for people localization and height estimation via 3D line sampling," in *Proc. IEEE Int. Conf. on Pattern Recognition*, pp. 2788-2791, Nov. 2012.

[48] K.-H. Lo and J.-H. Chuang, "Vanishing point-based line sampling for real-time people localization," *IEEE Transactions on Circuits and Systems for Video Technology*. (accepted)

[49] K.-H. Lo and J.-H. Chuang, "View-invariant measure of line correspondence and its application in people localization," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 1985-1988, Sep. 2012.

Appendix F. Publications

1. Kuo-Hua Lo and Jen-Hui Chuang, "Vanishing point-based line sampling for real-time people localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 10. (accepted)(SCI)
2. Kuo-Hua Lo, Jen-Hui Chuang, and Hua-Tsung Chen, "Efficient error analysis of a real-time vision-based pointing system", *Journal of Information Science and Engineering*, 2012, 09. (accepted)(SCI)
3. Kuo-Hua Lo, Chih-Jung Wang, Jen-Hui Chuang, and Hua-Tsung Chen, "Acceleration of vanishing point-based line sampling scheme for people localization and height estimation via 3D line sampling", *International Conference on Pattern Recognition*, Tsukuba, Japan, 2012,11.
4. Kuo-Hua Lo and Jen-Hui Chuang, "View-invariant measure of line correspondence and its application in people localization", *International Conference on Image Processing*, Florida, USA, 2012, 09.
5. Kuo-Hua Lo and Jen-Hui Chuang, "Vanishing point-based line sampling for efficient axis-based people localization", *International Conference on Image Processing*, pp. 529-532, Brussels, Belgium, 2011,09.
6. Kuo-Hua Lo and Jen-Hui Chuang, "Efficient people localization based on vanishing points and multiple homographies", *IPPR Conf. Computer Vision, Graphics, and Image Processing*, Taiwan, 2011,08. (Excellent paper award)
7. Kuo-Hua Lo, Jen-Hui Chuang, Yueh-Hsun Hsieh, and Hon-Yue Chou, "A point-based localization with error analysis", *International Computer Symposium*, Taiwan, 2010,11.
8. Sheng-Chung Huang, Han-Wei Kung, Kuo-Hua Lo, Hsing-Lu Huang, and Jen-Hui Chuang, "Human posture recognition system based on image scanning", *IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Taiwan, 2010,08.
9. Shang-Yi Lin, Ting-Chun Sun, Kuo-Hua Lo, Hsing-Lu Huang, and Jen-Hui Chuang, "Mobile localization system based on infrared cameras", *IPPR Conf. on Computer Vision, Graphics, and Image Processing, Taiwan*, 2010,08.
10. Jen-Hui Chuang, Chun-Wei Lee, and Kuo-Hua Lo, "Human activity analysis based on a torso-less representation", *International Conference on Pattern Recognition*, Tampa, USA, 2008,12.
11. Yi-Ta Tsai, Kuo-Hua Lo, Hsing-Lu Huang, and Jen-Hui Chuang, "Error analysis of a real-time vision- based pointing system", International Computer Symposium, Taiwan, 2008,11.
12. K.-H. Lo & M.T. Yang, "Shadow Detection by Integrating Multiple Features," *International Conference on Pattern Recognition*. August, 2006, Hong Kong.