Unsupervised Methods for Camera Pose Estimation and People Counting in Crowded Scenes

Nada Elasal

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

August 2016

Abstract

Most visual crowd counting methods rely on training with labeled data to learn a mapping between features in the image and the number of people in the scene. However, the exact nature of this mapping may change as a function of different scene and viewing conditions, limiting the ability of such supervised systems to generalize to novel conditions, and thus preventing broad deployment. Here I propose an alternative, unsupervised strategy anchored on a 3D simulation that automatically learns how groups of people appear in the image and adapts to the signal processing parameters of the current viewing scenario. To implement this 3D strategy, knowledge of the camera parameters is required. Most methods for automatic camera calibration make assumptions about regularities in scene structure or motion patterns, which do not always apply. I propose a novel motion-based approach for recovering camera tilt that does not require tracking. Having an automatic camera calibration method allows for the implementation of an accurate crowd counting algorithm that reasons in 3D. The system is evaluated on various datasets and compared against state-of-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is a great amount of research in the field of computer vision devoted to studying images and videos of isolated objects or humans. Although this is a critical first step for many computer vision tasks, such as object recognition, tracking and segmentation, it is important to note that this case is not representative of the real world. In real world scenarios, humans and objects usually appear in groups. Hence, studying and implementing models that are able to handle the challenges of crowds is becoming increasingly essential.

Recently, there has been a growing interest in the computer vision community in crowd behaviour analysis. The field has a wide range of application domains, including crowd counting, pedestrian tracking, abnormal activity detection, crowd management, design of public spaces and graphical crowd simulation. However, the problem is a highly complex one. With a large number of people in a scene, the number of simultaneous interactions between those individuals increases. Additionally, a number of typical computer vision tasks, such as detection, segmentation and tracking, become hard to perform reliably in a crowded scene due to the high level of occlusion and low resolution per individual. With this level of complexity, methods for automated crowd analytics should exploit new techniques that scale well and take into consideration the unique challenges of crowd scenes.

The ultimate goal of automated crowd analytics is to extract all possible properties and characteristics of the crowd, e.g. member count, major directions of motion, degree of coherency among its members, social groups, among others. Computing each of these properties represents a complex computer vision task on its own. For my thesis work, I focus on one specific task: crowd counting. The count of crowd members is a basic and essential piece of information to characterize any given crowd regardless of the application domain. Systems that perform other crowd analysis tasks such as social group identification or anomaly detection could utilize such knowledge to give a more informative description of a crowd.

Crowd counting is a challenging task mainly due to effects of perspective distortion, high degree of occlusion and clustering of people. Prior methods proposed the use of labeled training data to learn the mapping between clusters of people observed as foreground image blobs and the count in a certain scene. While this approach might lead to highly accurate results for a given scenario, it is not expected to generalize well to new scenes with different conditions. To handle further complications introduced by occlusion and perspective distortion 3D scene geometry has to be taken into account. Most prior work uses ’perspective maps’ to account for distance scaling over the image. However, this technique does not fully account for the occlusion process. Clusters high in the image tend to occlude each other, complicating the mapping between cluster size in the image and the true count. Here I propose an efficient, fully unsupervised method for crowd counting that handles the aforementioned issues by applying a simple simulation technique in 3D. Evaluation experiments show that the method achieves state-of-the-art results, has remarkably low bias and scales easily to denser crowds while not requiring any supervised training.

Reasoning in 3D requires knowledge about camera pose and parameters. In a typical scenario, camera roll is small and focal length is known or can be estimated, so that the critical remaining unknown is camera tilt. Existing methods in the literature for camera pose estimation typically rely on the following to recover camera parameters: 1) prior knowledge of static features such as families of straight parallel lines, curves or orthogonal structure in the scene or 2) detecting and tracking local features or active agents such as pedestrians and vehicles in the scene to extract straight line motion trajectories or 3) a combination of both. Here I show that in the case of crowded scenes relying on such features proves to be unreliable because 1) structural cues are often not present, confounded by irregularities or not easily detected due to occlusions and shadows, 2) the assumption of straight line motion can only be found in a very restricted set of scenarios such as highway scenes, and 3) successful detection and tracking of objects is often impossible due to the high degree of occlusion and low resolution per individual in crowded scenes.

This motivated the first part of my thesis work: developing a novel method for recovery of camera tilt. The method relies upon simple analysis of low level motion features, avoiding all dependencies mentioned above. Additionally, it is completely unsupervised and can therefore be applied without modification to a wide range of scene conditions. Results show that, where state-of-the-art camera calibration methods fail, the proposed method accurately recovers camera pose. I further show that the estimated camera tilt can be used to support an unsupervised method for crowd counting, the second contribution of my thesis.

1.1 Document Layout

This thesis is organized as follows. In Chapter 2, I present a novel method for estimating camera pose from motion. An analysis of the camera calibration literature is given in Section 2.2, highlighting gaps and limitations. I summarize the contributions of my work in Section 2.3. In Section 2.4, I discuss the camera model and the geometry assumed throughout the algorithm. A detailed description of the algorithm is given in Section 2.5. In Section 2.6 I present the datasets used for the experimental evaluation of the proposed method. Performance evaluation results are described in Section 2.7. In Section 2.8, evaluation results, possible limitations and future work are discussed.

In Chapter 3 of this thesis I present my work on people count estimation in crowded scenes. I introduce the problem of crowd counting in Section 3.1 and put my work in context by analyzing related work in literature in Section 3.2. A summary of contributions is given in Section 3.3. Section 3.4 gives a detailed description of the method proposed. I present the datasets used for the experimental evaluation of the proposed method in Section 3.5. Performance evaluation results are described in Section 3.6. Limitations and future work of the proposed method are discussed in Section 3.7.

Finally, I discuss implications of my thesis work as well as identify areas for future work in Chapter 4.

Chapter 2

Estimating Camera Pose from Motion without Tracking

2.1 Introduction

A major challenge for automatic video analytics is the geometric distortion induced by projection to the image, which complicates almost all tasks, including object detection, velocity estimation and crowd analysis. For many systems, camera roll is small and focal length is known or can be estimated, so that the critical remaining unknown is camera tilt. If camera tilt can be estimated, imagery can be rectified to remove the effects of projection, allowing unbiased analysis.

2.2 Prior Work

2.2.1 Mapping the 2D Image Plane to the 3D World

To perform successful camera calibration, a transformation between 2D coordinates in the image plane and the corresponding 3D real world coordinates is required. Such transformation is only possible with knowledge of a set of intrinsic (focal length, scale factors, distortion coefficients, and coordinates of the principal point) and extrinsic parameters (camera height, pan angle, tilt angle, and roll angle) of the camera system. Typically, intrinsic parameters remain constant over the period of camera operation, and can therefore be calibrated prior to installation, with the exception of the focal length, which can be varied to provide different views of the scene. On the other hand, extrinsic parameters may be changed with the use of Pan-tilt-zoom (PTZ) cameras. To solve for those parameters, existing auto-calibration methods compute vanishing points (i.e. points, where parallel lines in the world meet in the image), which makes the system of equations solvable. Those methods differ in the number of vanishing points computed: In theory, one vanishing point is sufficient, assuming a priori knowledge of the other parameters. On the other extreme, some methods compute three vanishing points, to avoid dependence on known parameters. For the purpose of my thesis, I will categorize existing auto-calibration algorithms, based on the type of scene features used to compute vanishing points.

2.2.2 Pose Estimation from Structure in Static Images

Auto-calibration and rectification algorithms typically rely upon prior knowledge of static features such as families of straight parallel lines, curves [11] or orthogonal structure [21, 26, 45, 46, 13] in the scene from which vanishing points can be computed. For those algorithms, successful detection of those structural features is key to recovering camera pose accurately. However in many situations these static features are not present, are confounded by irregularities or are not easily detected due to occlusions and shadows (e.g., 2.1(a)), which results in failure of this type of methods.

2.2.3 Pose Estimation from Motion Information

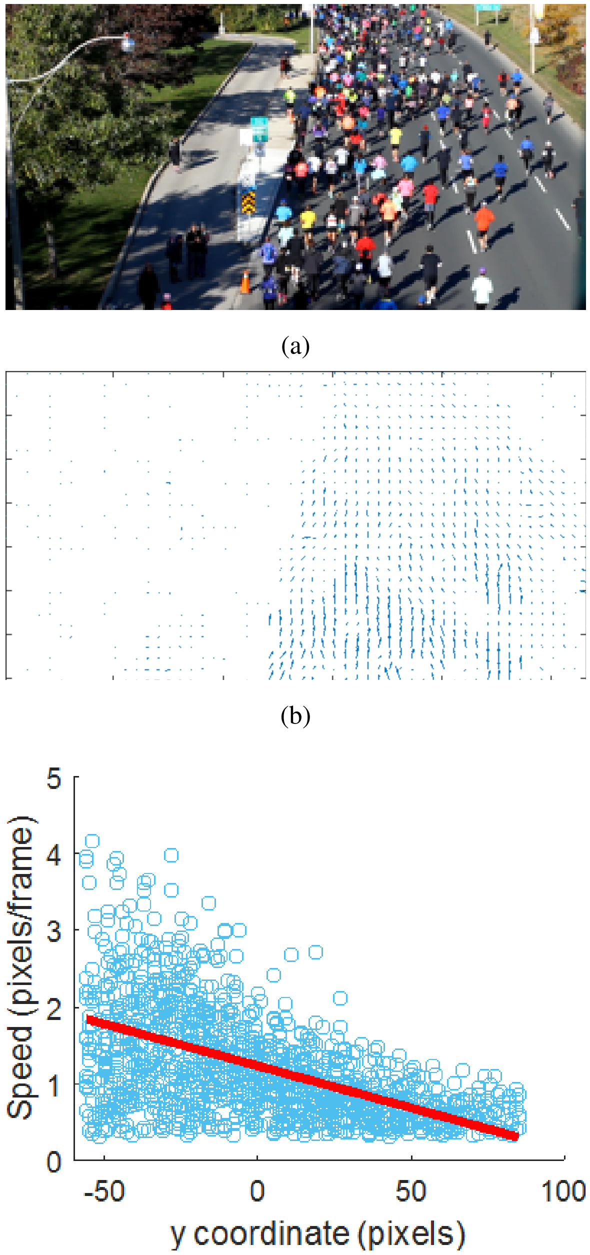
An alternative or complementary approach is to use motion information from the active agents such as pedestrians and vehicles in the scene. This approach has been applied to traffic analysis. For example, Zhang [49] estimated two vanishing points from vehicle shape and then used motion information to distinguish between the two corresponding principal axes. Zhang *et al* [50] detected and classified moving objects to estimate two horizontal vanishing points from vehicles and a vertical vanishing point from the orientation of pedestrians in the scene. Dubska *et al* [16] tracked local feature points to obtain straight motion trajectories that could be used to estimate one ground plane vanishing point.

Pedestrian motion has also been used for auto-calibration. For example, Kuo *et al* [30] estimated camera parameters from sequences of co-planar key-point features projecting from the main joints of a walking human. However, this approach has only been demonstrated using motion-capture data; reliable automatic detection of these key points from surveillance video in crowded scenes would be challenging. The human tracking approach of Lv *et al* [33] is potentially more feasible for real-world scenes as it does not require joint structure to be identified. However it does still require accurate segmentation of individual human forms and identification of their principal axes to estimate vertical, as well as accurate detection of feet and head positions and tracking over time to estimate the horizon line. Bose *et al* [3] reported a more general approach to auto-calibration based on tracking multiple objects (pedestrians or cars). The method does not require any form of shape analysis, and hence does not depend upon an exact segmentation of the moving objects. However, it does require that these objects be tracked over time, and assumes that the tracked objects are moving at constant speed along straight trajectories in the scene. Limitations of this category of methods are:

1. The assumption of straight line motion of active agents, which restricts the application domain to very specific scenarios like highway scenes.
2. Dependence on detection and tracking of objects or local features, which is a computationally expensive task, but most importantly it can rarely be performed reliably in crowded scenes, due to occlusions and low resolution per individuals.

2.3 Summary of Contributions

In this part of my thesis work, I present a novel method for recovering camera pose, specifically tilt, based on a very simple analysis of the motion field. The proposed method does not involve explicit computation of vanishing points. To the best of my knowledge, there has been no prior work done on using low level motion features to recover camera pose without explicitly searching for orthogonal directions to compute vanishing points. The method is anchored on a relatively general assumption of zero correlation of speed with position in the direction pro-



(a)

(b)

(c)

Figure 2.1: a) Sample frame from marathon dataset, b) Corresponding optical flow field, c) Scatter plot of speed (magnitude of optical flow vectors) vs y coordinate. (y increases upward.)

jecting to the vertical axis of the image. Advantages of the proposed approach include:

1. No dependence on the visibility of regular static structures in the scene.
2. No requirements that moving objects be segmented.
3. No dependence on object shape analysis.
4. No requirement that objects or features be tracked over time.
5. No assumption that objects move in the same direction.
6. No assumption that individual motions are linear or constant speed.
7. No dependence on learned parameters, meaning that the approach can be applied to a broad range of situations without retraining.

Fig. 2.1 illustrates the idea behind the approach. Despite stationary motion statistics over the right portion of the ground plane (top), the oblique angle of the camera induces a projective distortion on the optic flow (center), resulting in a decline in image speed with height in the image. This statistical relationship can be captured with a simple affine model (bottom). The strength of this affine relationship generally increases with the camera tilt angle ϕ relative to the ground surface normal (Fig. 2.2). Given an estimate $\hat{\phi}$ of the tilt angle, the optic flow field can be re-rendered in rectified coordinates and this should result in a reduced correlation between image speed and height in the image. Thus the tilt angle can be estimated by gradient descent on the variance in the rectified optic flow explained by the affine model.

I stress that this algorithm makes no assumption about the azimuthal angle of the camera relative to the motion in the scene. To verify this, I evaluate performance for a variety of motion scenarios: In the first and third scenes of Fig. 2.6, image motion is primarily along the y-axis of the camera frame, in the second scene it is in diverse directions, and in the final scene the dominant motion is at roughly 20 deg counter-clockwise from the x-axis, i.e., much closer to the x-axis than the y-axis.

2.4 Geometry

I assume a camera with known focal length, a central principal point, square pixels and zero skew. (Other values for these parameters could easily be incorporated or calibrated out if measured in the lab.) I also assume negligible camera roll, which is reasonable for many installations. In principal, this method could be generalized to estimate camera roll by searching for the direction in the image that maximizes the correlation with image speed, but I have not yet explored this possibility.

I assume a planar horizontal ground surface and adopt a right-hand world coordinate system $[X, Y, Z]$ where the Z-axis is in the upward normal direction (Fig. 2.2 a). Without loss of generality, I locate the origin of the world coordinate system at the intersection of the optical axis of the camera with the ground plane, and assume that the x-axis of the image coordinate system aligns with the X axis of the world coordinate system, so that the y-axis of the image is the projection of the Y-axis of the world frame.

Under these conditions, a point $[X, Y]^T$ on the ground plane projects to a point $[x, y]^T$ on the image plane according to [20]

$$\lambda[x, y, 1]^T = H[X, Y, 1]^T, \tag{2.1}$$

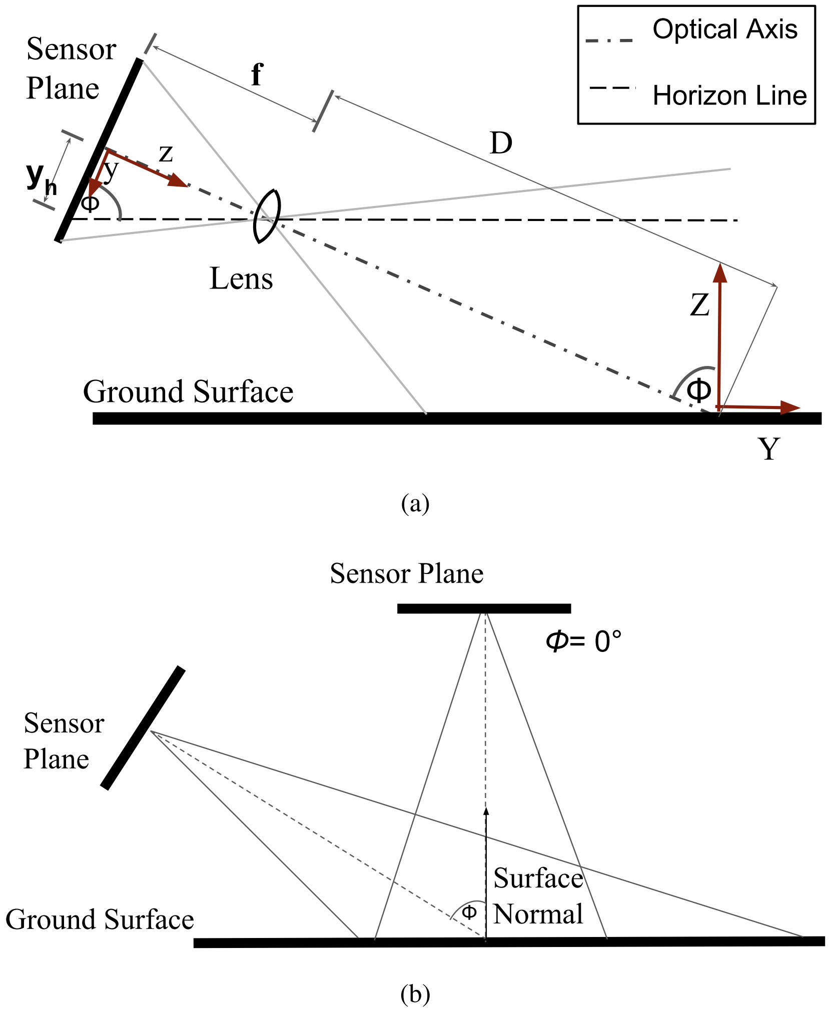


Figure 2.2: a) I seek to estimate the camera tilt angle ϕ. The light grey lines demarcate the field of view of the camera. The X axis of the world coordinate system and the x axis in the image plane are aligned and pointing in the direction perpendicular to the paper surface. b) Rectified plan view of the ground surface ($\phi = 0$) seen from a height D.

where λ is a scaling factor and the homography H is given by

$$H = \begin{bmatrix} f & 0 & 0 \\ 0 & f\cos\phi & 0 \\ 0 & \sin\phi & D \end{bmatrix} \tag{2.2}$$

Here f is the focal length in pixels, D is the distance of the optical centre of the camera from the ground plane along the optic axis, and ϕ is the tilt angle of the camera relative to the ground plane: $\phi = 0$ when the camera points straight down at the ground surface and increases to $\pi/2$ as the camera tilts up toward the horizon.

Conversely, points in the image can be back projected to the ground plane using the inverse of this homography, $[X, Y, 1]^T = \lambda H^{-1}[x, y, 1]^T$, where

$$H^{-1} = \begin{bmatrix} f^{-1} & 0 & 0 \\ 0 & (f\cos\phi)^{-1} & 0 \\ 0 & -(fD)^{-1}\tan\phi & \frac{1}{D} \end{bmatrix} \tag{2.3}$$

In Euclidean coordinates this backprojection can be written as:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{f/D}{1 - y/y_h} \begin{bmatrix} x \\ y/\cos\phi \end{bmatrix}, \tag{2.4}$$

where $y_h = f\cot\phi$ is the image projection of the horizon.

As a final step, I can apply the homography H of Eqn. (2.2) with a tilt angle of $\phi = 0$ to the scene points $[X, Y]^T$ computed using (2.4), transferring these scene points to image points $[x_r, y_r]^T$ taken by a “bird’s eye” virtual camera, yielding a rectified plan view of the ground surface seen from a height D (Figure 2.2 b):

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \frac{1}{1 - y/y_h} \begin{bmatrix} x \\ y/\cos\phi \end{bmatrix} \tag{2.5}$$

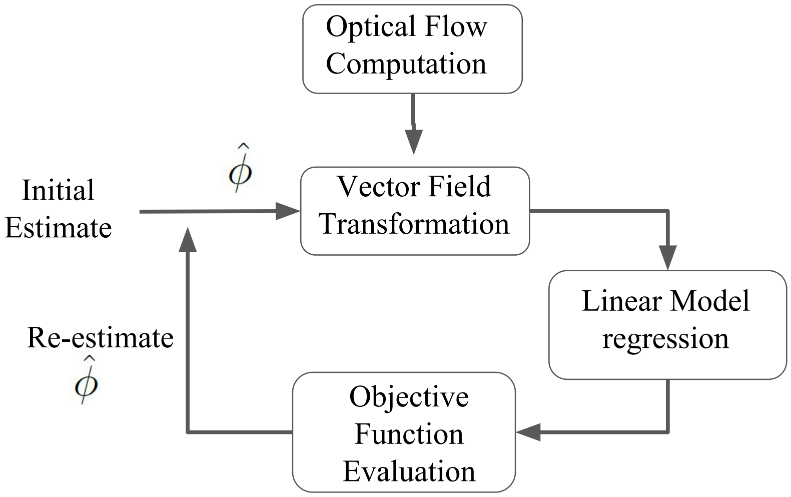


Figure 2.3: Algorithm overview

Taking the time derivative, I can compute the rectified optic flow field:

$$\mathbf{v}_r = \begin{bmatrix} x'_r \\ y'_r \end{bmatrix} = \frac{1}{\left(1 - y/y_h\right)^2} \begin{bmatrix} \left(1 - y/y_h\right) x' + (x/y_h) y' \\ y' / \cos\phi \end{bmatrix} \tag{2.6}$$

The key assumption is that the average rectified speed $v_r = |\mathbf{v}_r|$ is invariant with the vertical image coordinate. Thus an estimate of the tilt angle ϕ can be evaluated by measuring the correlation of $v_r\left(x_r, y_r|\phi\right)$ with y_r.

2.5 Algorithm

Figure 2.3 provides an overview of the algorithm. The system estimates the tilt angle value using an iterative approach. The input to the system are the motion fields of a sequence of frames. At each iteration, the motion vectors are projected to the hypothesized rectification plane using the current tilt estimate. Next, the correlation of the rectified speeds with the height in the rectified image plane is measured using a linear regression model. The proportion of variance explained by the current model is then evaluated as the objective function. Finally a new estimate for the tilt angle value ϕ is computed in a gradient descent fashion to minimize the objective function.

2.5.1 Objective Function

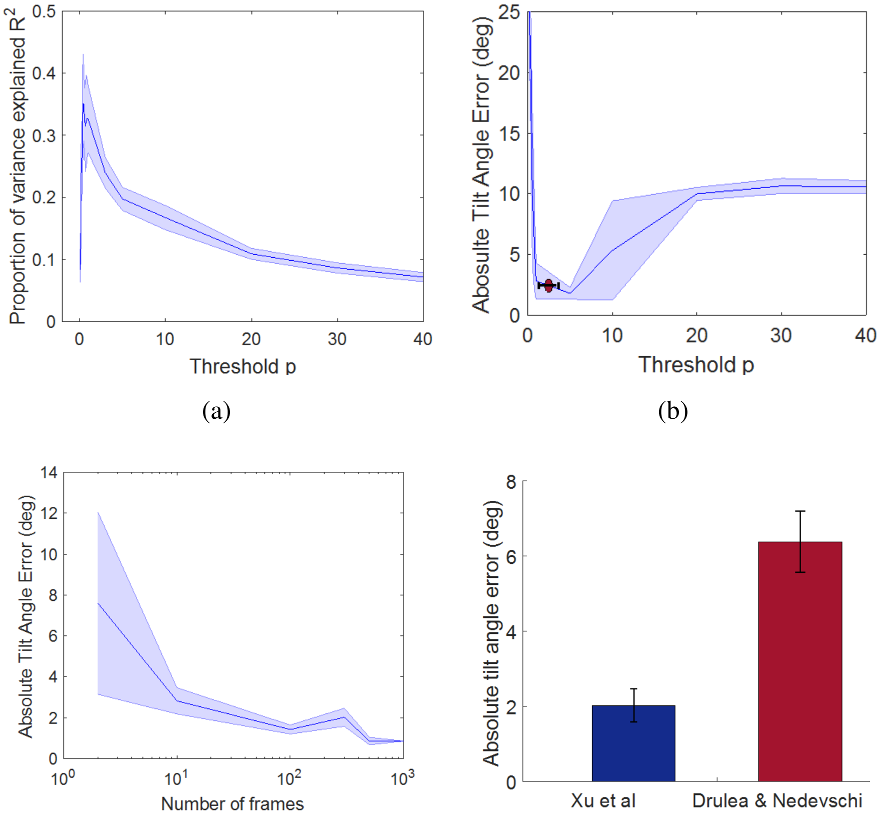
Given an estimate $\hat{\phi}$ of the tilt angle, I can compute the rectified speeds $v_r\left(x_r, y_r|\hat{\phi}\right)$. To assess correlation with y_r I use the simple affine model $\hat{v}_r = ay_r + b$ and determine maximum likelihood estimates of the parameters (a, b) by linear regression. The strength of this correlation is measured by the proportion of variance $R^2\left(\hat{\phi}\right)$ explained by the model, given the estimated tilt $\hat{\phi}$:

$$R^2\left(\hat{\phi}\right) = 1 - \frac{\mathbb{E}\left[(v_r - \hat{v}_r)^2\right]}{\mathbb{E}\left[(v_r - \bar{v}_r)^2\right]}, \tag{2.7}$$

where $\bar{v}_r$ is the average rectified speed over the rectified image. I seek the tilt angle ϕ^* that minimizes $R^2\left(\hat{\phi}\right)$.

2.5.2 Optimization

The tilt angle ϕ^* is estimated by iterative minimization of Eqn. 2.7 using MATLAB's **fminsearch** (Nelder-Mead simplex method). In my experiments I repeat the search from a coarse regular sampling of initial estimates $0 \leq \hat{\phi} \leq \pi/2$, selecting the ϕ^* that yields the minimum R^2. However in practice I find that given sufficient input data ($>$ 100 frames) the error function is convex and a single search initiated at $\hat{\phi} = \pi/4$ suffices.



(a)

(b)

(c)

(d)

Figure 2.4: Analysis of algorithm dependence on key variables, for the highway dataset. Shading and error bars indicate standard error of the mean. (a-c) show results using the optic flow algorithm of Xu et al [48]. (a), (b) and (d) show results for 300-frame sequences. a) Average proportion of variance R^2 explained by the affine model as a function of the optic flow threshold p. b) Average tilt error of trackless algorithm as a function of the optic flow threshold p. The red dot indicates the threshold p chosen automatically by the algorithm. c) Average tilt error of trackless algorithm as a function of the number of video frames analyzed. d) Average tilt error of trackless algorithm based on the optical flow algorithms of Xu et al [48] and Drulea & Nedevschi[15].

2.5.3 Optical Flow Computation

I employ the optical flow algorithm of Xu et al [48]: Fig. 2.1(b) shows an example for the marathon dataset. Since I am only concerned with motion on the ground plane, motion vectors above the current estimate of the horizon $\hat{y}_h = f \cot \hat{\phi}$ are ignored.

Ground plane motion will generally be sparse and spatially interleaved with noise due to small environmental motions, camera vibration etc. that does not correlate with y_r and thus could reduce accuracy. This problem can be mitigated by filtering out all but the largest $p\%$ of motion vectors from each image frame: Fig. 2.5 shows an example for the marathon data set. However, this leaves the problem of estimating the optimal threshold p.

To avoid supervised learning of this parameter, which may not generalize to novel scenarios, I employ an adaptive method to select p individually for each video sequence. In particular, I select the p ($0 < p \leq 100$) that maximizes $R^2(0)$, the variance in unrectified image speed $v\left(x, y\right)$ explained by correlation with the vertical image coordinate y (Fig. 2.1(c)). Fig. 2.4(a) shows that for the highway dataset, the proportion of variance explained peaks when a relatively small fraction (around 4%) of the motion vectors are employed . Fig. 2.4(b) shows that by this threshold yields nearly minimal error in the tilt angle estimate. Thus by selecting the threshold that maximizes variance explained in the unrectified image, I adaptively optimize the accuracy of the algorithm.

Dependence on the optic flow method employed is also evaluated (Fig 2.4(d)). I evaluate the methods of Xu et al [48] and Drulea & Nedevschi[15], both highly ranked on the Middlebury dataset [2]. While both work reasonably well, I find the algorithm of Xu et al [48] more accurate for this application and dataset.

Fig. 2.4(c) shows how accuracy varies as a function of integration time. For the highway dataset, performance is very good for durations of 50 frames (1.7 sec) or more.

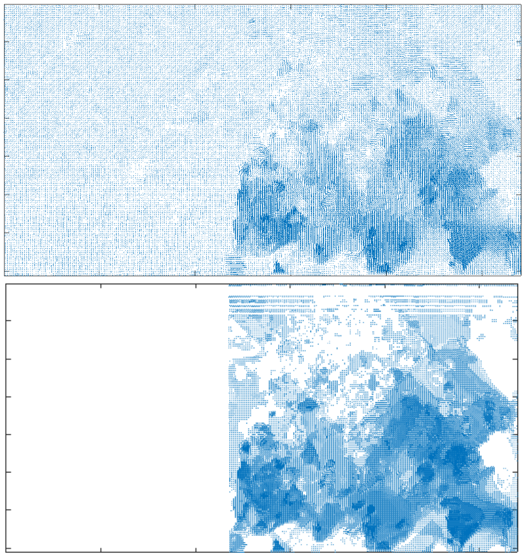


Figure 2.5: Example optical flow vector field from the marathon dataset, before (left) and after (right) noise removal. The automatically selected speed threshold is p=28%.

2.6 Datasets

I evaluate the proposed method on 4 diverse datasets (Fig. 2.6) recorded with 3 different camera/lens systems to assess the generality of the approach: 1) a highway scene where the moving agents are vehicles, 2) an outdoor campus scene where the moving agents are pedestrians, 3) an urban marathon scene where the moving agents are runners and 4) an indoor scene where the moving agents are

Dataset	Focal length (pixels)	True tilt angle (deg)
Highway	174	87.8
Outdoor Pedestrian	953	81.0
Marathon	700	76.4
Indoor	584	60.7

Table 2.1: Parameters of the four datasets.

pedestrians. All camera/lens systems were calibrated in the lab using a standard calibration procedure to determine focal length f (Table 2.2). Each dataset was partitioned into 5 clips of 300 frames each.

The **highway dataset** was recorded with a Point Grey Cricket camera equipped with a wide-angle lens at 30 fps and down-sampled to 275×155 pixels. Ground truth camera tilt angle $\phi = 87.8$ deg was estimated manually from the horizon image height y_h using the relation $y_h = f \cot \phi$ (Eqn. 2.4). The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 2% for this dataset. Average run time per sequence was 32 sec (excluding optical flow computation).

The **outdoor pedestrian dataset** was recorded with a Point Grey Cricket camera equipped with a 16 mm lens at 30 fps. The frames were down-sampled to 320×165 pixels. A digital inclinometer was used to measure ground truth tilt angle: $\phi = 81.0$ degrees. The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 10% for this dataset. Average run time per sequence was 115 sec (excluding optical flow computation).

The **marathon dataset** was recorded with a Canon EOS Rebel T3i camera equipped with a 40 mm lens at 30 fps. The frames were cropped and down-sampled to 324×156 pixels. A digital inclinometer was used to measure ground truth tilt angle: $\phi = 76.4$ degrees. The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 28% for this dataset. Average run time per sequence was 125 sec (excluding optical flow computation).

The **indoor dataset** was recorded with a Canon EOS Rebel T3i camera equipped with a 40 mm lens at 30 fps. The frames were down-sampled to 320×182 pixels. A digital inclinometer was used to measure ground truth tilt angle: $\phi = 60.7$ degrees. The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 5% for this dataset. Average run time per sequence was 105 sec (excluding optical flow computation).

2.7 Evaluation

2.7.1 Implementation

I employed the optical flow method of Xu et al [48] (code downloaded from `www.cse.cuhk.edu.hk/leojia/projects/flow`), with parameters matching those used by the authors for evaluation on the Middlebury Benchmark: regularization strength: 6, occlusion handling: 1 and large motion: 0. The average run time for optical flow computation is 45 sec per frame. I implemented the proposed algorithm in MATLAB and have not yet optimized the code for speed (run times listed below). All experiments were conducted on a 4-core desktop computer.

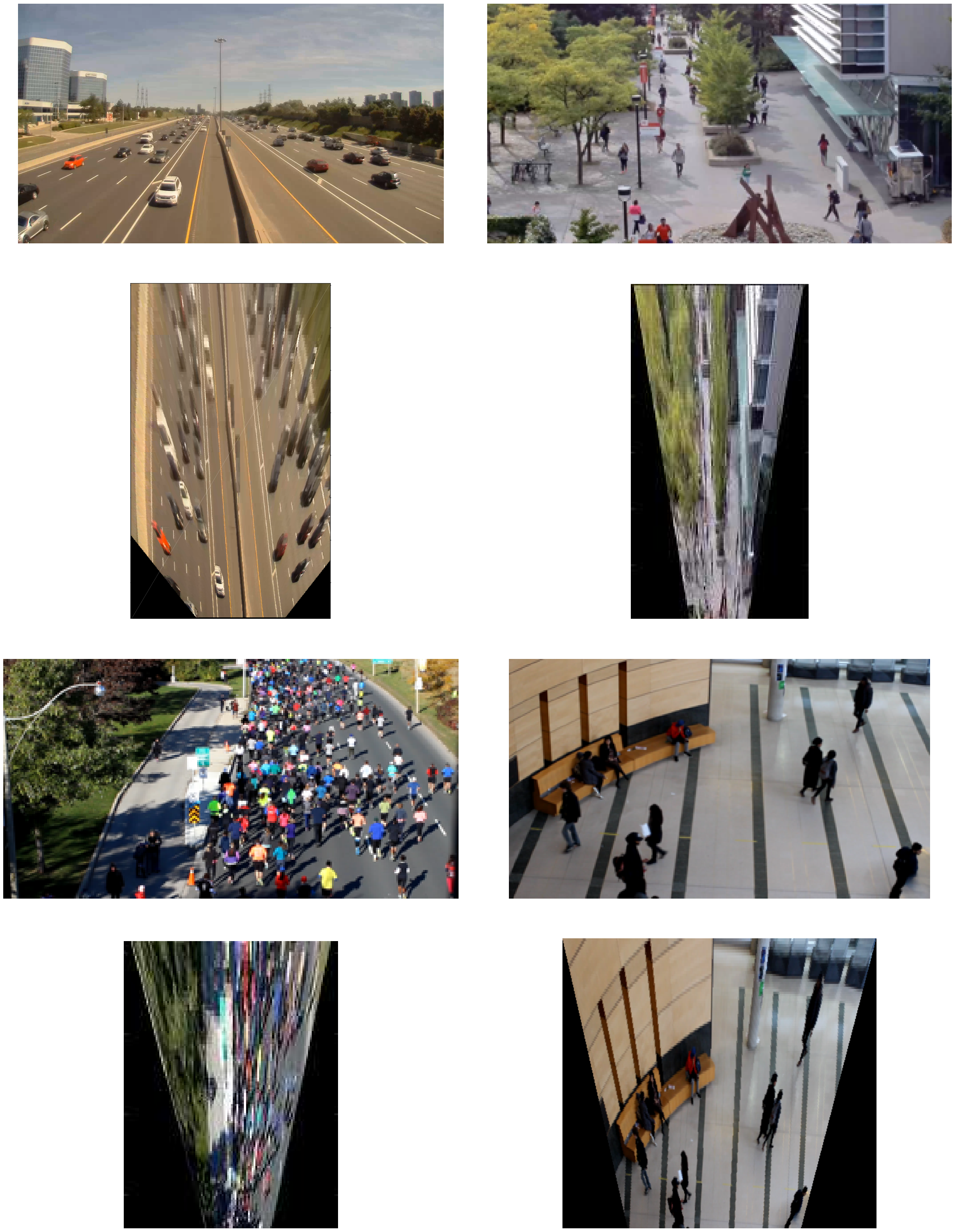


Figure 2.6: Sample frames and example rectifications computed using the proposed trackless method for the highway, outdoor pedestrian, marathon and indoor datasets.

2.7.2 Evaluation & Comparison with Static Methods

Fig. 2.7 shows the quantitative performance of the proposed method on the four datasets and Fig. 2.6 shows example rectifications based on these estimated tilt angles. I was able to secure code directly from the authors of three prior methods that use static features (lines or curves) to estimate vanishing points and tilt angle [11, 45, 47]; Fig. 2.7 shows the performance of these prior static methods alongside mine.

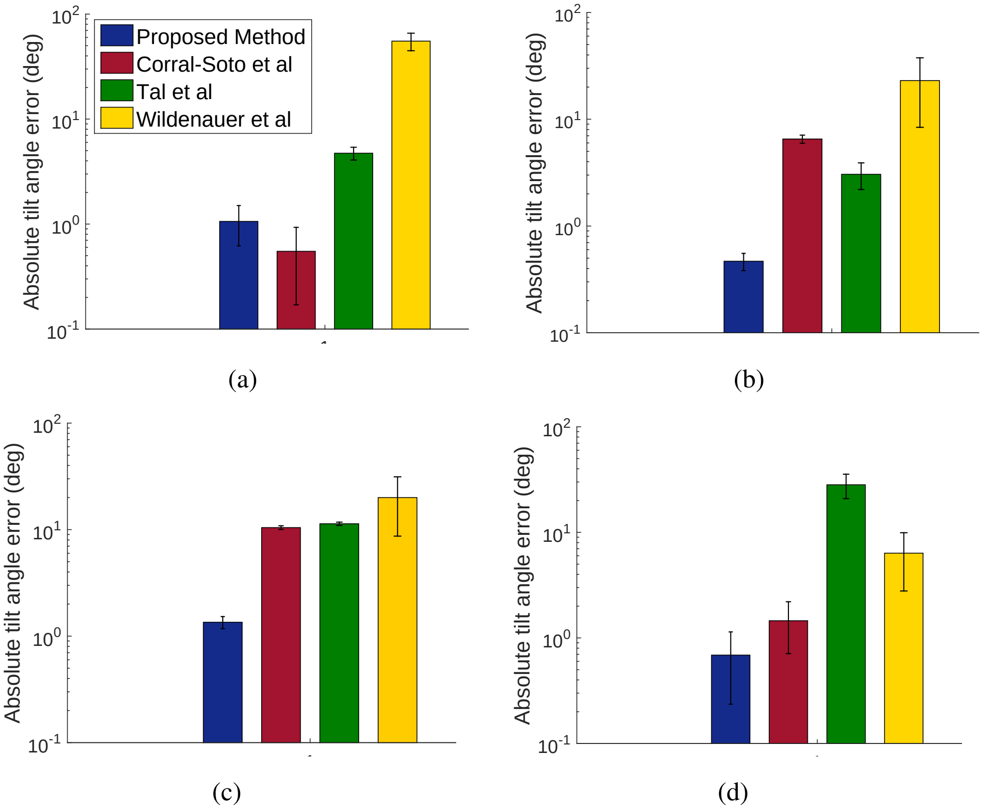
Highway Dataset

Mean absolute error for my method was 1.06 deg, much better than the methods of Tal & Elder and Wildenauer & Hanbury (4.73 deg and 55.28 deg, respectively), but somewhat worse than the method of Corral-Soto & Elder (0.55 deg), which I note was designed specifically for highway applications.

The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 2% for this dataset. Average run time per sequence was 32 sec (excluding optical flow computation).

Outdoor Pedestrian Dataset

Mean absolute error for my method was only 0.46 deg. I note that the proposed method is highly accurate here despite the substantial variations in the directions and speeds of motion of the pedestrians in this video, highlighting the fact that my method does not require that motions of individual agents be similar. The relatively large errors produced by the prior static methods evaluated (6.53, 3.05, 23.41 deg) presumably reflect the relative sparseness of static regularities in the scene.



(a)

(b)

(c)

(d)

Figure 2.7: Performance of the proposed method compared with three state-of-the-art static methods from Corral-Soto & Elder [11], Tal & Elder [45] and Wildenauer & Hanbury [47] on (a) the highway dataset, (b) the outdoor pedestrian dataset, (c) the marathon dataset and d) the indoor dataset.

The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 10% for this dataset. Average run time per sequence was 115 sec (excluding optical flow computation).

Marathon Dataset

Mean absolute error for my method was 1.48 deg, substantially better than competing static methods (10.45, 11.35 and 19.99 deg). Again, this superiority reflects the relatively strong motion signals and sparseness of clear static cues such as parallel lines and curves.

The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 28% for this dataset. Average run time per sequence was 125 sec (excluding optical flow computation).

Indoor Dataset

Mean absolute error for my method was 0.68 deg, substantially outperforming competing static methods (1.45, 28.22 and 6.35 deg). Note that in this dataset the dominant direction of motion is roughly 20 degrees counterclockwise from the x-axis of the image. The excellent performance of the proposed algorithm for this example illustrates the invariance of the method to the dominant direction of motion.

The average speed threshold p over the five independent experiments, selected automatically (Section 2.5.3), was 5% for this dataset. Average run time per sequence was 105 sec (excluding optical flow computation).

2.7.3 Comparison with Motion-Based Methods

It would be ideal to compare the proposed method with the prior motion-based methods from Dubska *et al* [16] and Zhang *et al* [50] directly on a common dataset as well. Unfortunately, despite contacting authors I was unable to obtain code or

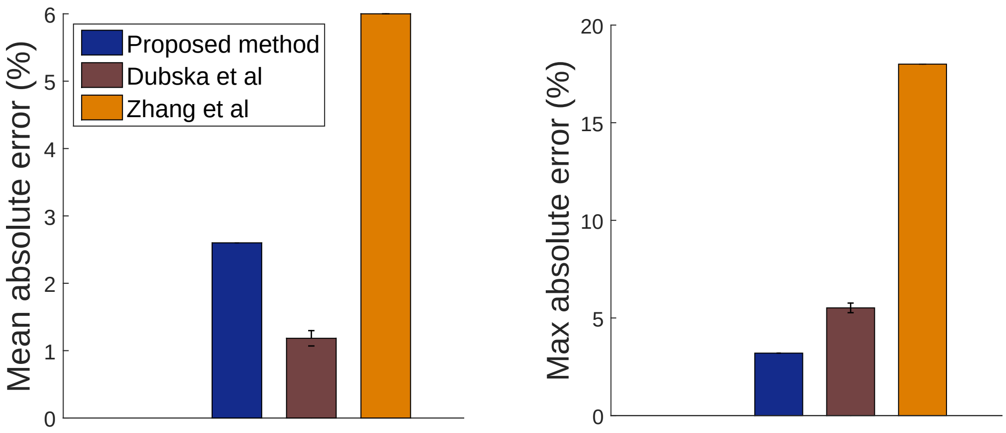
Table 2.2: Experimental results of the proposed method on four datasets.

Dataset	True tilt angle (deg)	p (%)	Mean error (deg)
Highway	87.8	2	1.06
Outdoor Pedestrian	81.0	10	0.46
Marathon	76.4	28	1.48
Indoor	60.7	5	0.68

datasets used for either method. The best I can do at this stage is to compare performance of these prior algorithms on their proprietary datasets, as reported by the authors, with the performance of the proposed algorithm on my dataset. (I caution that since there could be systematic differences in the difficulty of the datasets, this comparison should not be used to formally rank the algorithms.)

Comparison with these prior motion-based methods is further complicated by the fact that in this prior work the authors did not have ground truth tilt estimates. Instead, they manually identified point pairs in the image known to lie in a horizontal plane and to be equidistant in the scene, and then computed the average absolute percentage deviation of the distances in the rectified imagery from their mean: a more accurate homography estimate should lead to lower average deviation.

Since equidistant horizontal point pairs are not easily identifiable in my pedestrian and marathon datasets, attention was restricted to the highway dataset. I identified 8 point pairs, each pair projecting from fixed points at the same height on the same vehicle, over 10 consecutive frames. I then projected these points



(a)

(b)

Figure 2.8: Performance of the proposed method on the highway dataset compared with two state-of-the-art motion-based methods from Dubska *et al* [16] and Zhang *et al* [50] **on different datasets**. a) Mean absolute deviation of point pairs (%), b) Maximum absolute deviation of point pairs (%).

to the rectified image using the estimated homography matrix and measured the mean absolute deviation of their separation over the 10 frames, as in Dubska *et al* [16].

Fig. 2.8 shows mean absolute deviation of distance between point pairs in rectified imagery for my algorithm on the highway dataset, compared with the errors reported by Dubska *et al* [16] and Zhang *et al* [50] on their respective datasets. I find that by this measure the proposed method has a mean error of 2.6%, lying between the performance reported by Dubska *et al* [16] (1.18%) and that reported by Zhang *et al* [50] (6%) on their respective datasets. However, my method appears to have higher reliability, with a maximum error rate of only 3.2%, compared to 5.5% for the method of Dubska *et al* [16] and 18% for the method of Zhang *et al* [50]. I emphasize that the proposed method also does not require explicit tracking or vanishing point estimation, as required by these prior methods, and thus has potentially lower computational requirements and greater generality.

2.8 Discussion

The proposed trackless motion-based method for camera tilt estimation was found to work reliably over four very different camera/lens systems, scenes, active agents and patterns of motion, with average absolute tilt errors ranging from 0.46 to 1.48 deg. For the highway dataset the static method of Corral-Soto & Elder [11] based on curve parallelism was found to be slightly more accurate. This dataset represents an ideal case scenario for the method of Corral-Soto & Elder, where the family of parallel lane markings are clearly visible. My motion-based method performs almost as well (0.55 deg vs 1.06 deg error) and much better than the other two state-of-the-art static methods (4.73 deg and 55.28 deg error) I assessed[45, 47] .

For the other three datasets (outdoor pedestrian, marathon and indoor) where parallel families of lines and curves are either not as clearly visible or occluded by moving agents, the proposed method outperformed all three methods based on static structural features [45, 11, 47] by a large margin, highlighting the relative generality of the proposed approach.

Unfortunately direct comparison of the proposed trackless motion-based method against prior motion-based methods from Dubska *et al* [16] and Zhang *et al* [50] was not possible due to lack of a common dataset and/or shared code. However, informal comparison on different but similar datasets (Fig. 2.8) suggests that for traffic data, my approach may be comparable to the method of Dubska *et al* (higher mean error, lower max error) and much better than the method of Zhang *et al*. I also note that these two competing motion-based approaches depend upon explicit appearance modeling, feature tracking and vanishing point estimation, and have been tailored specifically to traffic applications. The proposed approach, on the other hand, works for general dynamic scenes on a ground plane and does not require explicit tracking or vanishing point estimation.

Just as static methods do poorly when regular static features are sparse, motion-based methods such as the trackless method proposed here will do less well when motion is sparse. Characterizing exactly how performance varies with sparsity of motion remains a topic for future work, however note that the automatic method for denoising the optic flow field selects as little as 2% of optic flow vectors with good results, suggesting that dense motion is not necessarily required. Furthermore, in a motion-based method, the quality of the estimate can be improved continuously over time as additional independent motion vectors are observed, something that is not possible for static methods. However, since there will always be some scenes where static methods work best and others where motion-based methods work best, the ultimate system would likely employ both approaches and arbitrate between them on a case-by-case basis.

In future work, I intend to characterize the accuracy of the method as a function of motion density and to extend the method to recover roll angle by searching for the direction in the image that maximizes the correlation with image speed. Another goal is to develop compatible methods for estimating focal length, important for camera installations employing zoom lenses. However this will, I believe, entail stronger assumptions on the pattern of motions, for example, that the average flow field is parallel in the scene.

Chapter 3

Unsupervised Crowd Counting

3.1 Introduction

Automatic systems for estimating the number of people in a visual scene have applications in urban planning, transportation, event management, retail, security, emergency response and disaster management. The problem can be challenging for a number of reasons: 1) the projected size of people in the scene varies over the image depending upon camera parameters, not all of which may be known accurately, 2) inevitable errors in signal processing may lead to partial detection (Fig. 3.1), 3) people in the scene will often overlap on projection and thus be detected as a single cluster rather than individuals, 4) due to variations in pose and distance, detailed features of the human body may not be discriminable, making it difficult to accurately parse these clusters(Fig. 3.1), and 5) the computational budget is typically limited by the need to run at frame rate.

Here I propose to meet these challenges by embracing two key principles: 1) Reasoning in 3D and 2) Unsupervised adaptive processing.

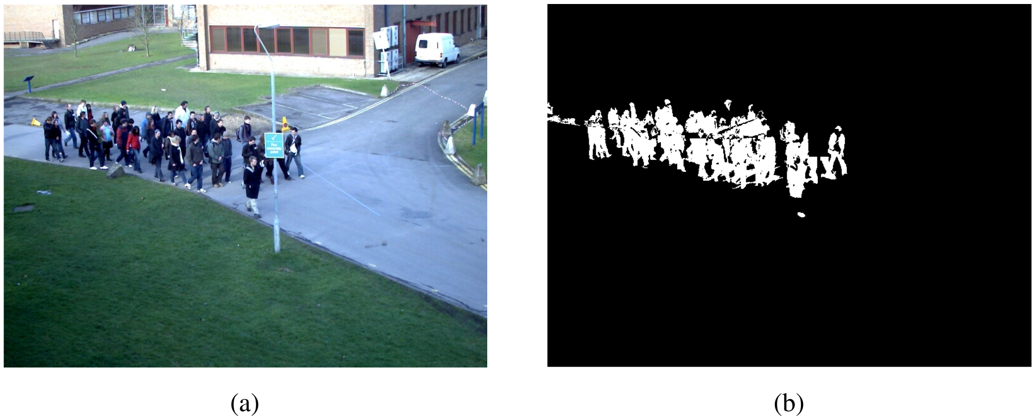


Figure 3.1: (a) High density frame from PETS 2009 dataset. (b) Background subtraction

3.1.1 Reasoning in 3D

Reasoning in 3D is crucial for accurate crowd counting, for two reasons. First, to produce reliable estimates, stationarity assumptions must be made, and while such assumptions can be reasonable in scene (e.g., ground plane) coordinates, they are generally *not* reasonable in the image, due to the effects of projection. For example, for a scene where people are uniformly distributed over the ground plane, their projected size falls and their density typically rises higher in the image due to perspective projection. Second, the occlusions caused by projection are a central complication of crowd estimation, and these can only be properly modeled in 3D.

While prior work has incorporated elements of 3D reasoning (see below), in most cases this has amounted only to inverse scaling with estimated distance, which does not adequately handle occlusion. Attempts to incorporate more complete 3D modeling, on the other hand, have suffered from the combinatorial complexity of considering all possible configurations of people in the scene, ruling out frame rate deployment. One of my key contributions is to show that the computationally intensive component of the problem can be solved offline using a simple sampling technique, limiting run time computations to simple one-shot feature estimation and linear estimation.

3.1.2 Unsupervised Adaptive Processing

A second key problem is that, due to inevitable errors in assumptions and parameter estimates as well as signal processing noise, theoretical predictions of how people should appear in the image will tend to be systematically biased. For example, errors in background subtraction may tend to miss the head and/or feet of people in the scene, making image segments smaller than predicted.

It is tempting to try to reduce this bias through standard supervised methods. Indeed, most prior methods use training videos annotated with the number and usually the location of the people in the scene to learn a regressor that predicts the number of people in each frame from a collection of image features. The difficulty is that the statistical relationships learned may be quite specific to the particular viewing geometry (distance, camera tilt, focal length) and photometric conditions (illumination, background colours) as well as the signal processing parameters (camera resolution, noise) of the training data set, and thus may not generalize well to new scenes.

To avoid this problem, I propose a novel unsupervised adaptive method that fine-tunes a model for the mapping between people in the scene and observations in the image. Combining this adaptive method with a novel 3D simulation approach leads to a fast and accurate crowd estimation algorithm with remarkably low bias.

3.2 Prior Work

It is important to be clear on the goal. Some prior work (e.g., [37, 39]), aims only to report some relative measure of crowd density. Here, I focus on the more challenging problem of crowd *counting*, i.e., estimating the absolute number of people in each frame of the scene. To place my work in context, I focus on two key issues: reasoning in 3D and unsupervised adaptive processing.

3.2.1 Reasoning in 3D

The importance of accounting for the effects of 3D perspective projection has long been recognized, however mainly this has consisted of weighting pixels by 'perspective maps' that take into account distance scaling over the image. Early attempts used manually [37, 34] or automatically [32] labelled features to compute the perspective map, while later methods [41, 4, 10, 9, 19] have tended to use camera calibration data.

Unfortunately, perspective scaling does not fully account for the complexities of occlusion: clusters of people, especially when seen higher in the image, tend to occlude each other (Fig. 3.1), and as a consequence the number of people in an image cluster can vary in a complex way with the size of the image cluster. Failing to account for this effect will lead to systematic bias.

Ryan et al. [42] addressed this problem by explicitly incorporating the angle of the view vector with respect to the ground plane as a predictor of the size of the crowd within a standard supervised regression framework. However, their approach requires annotation of each individual within each frame of the training dataset, and is subject to the limitations of supervised approaches (see below).

There are also several prior studies that attempt to address the occlusion problem through a more complete 3D modeling of the scene. Zhao [51] employed 3D human models consisting of four ellipsoids (accounting for head, torso and legs), matching to detected foreground regions. Because individuals are explicitly modeled as 3D objects in the scene, the effects of occlusion can be factored in. However, due to the combinatorial nature of the problem, the method is computationally intensive, making it infeasible for real-time applications for large crowds. Moreover, as crowd density increases, the discriminability between different possible configurations is expected to decline.

Kilambi et al. [27, 28] (also see Fehr et al. [18]) avoided this computational complexity by modeling image segments as aggregates, back-projecting to head and ground planes (Fig. 3.2b) to identify the ground plane footprint representing the 3D extent of the group, and thus properly accounting for occlusion. However, the method requires labelled training data and supervision to compute the back-projected area of individual people, and this computation assumes all people to be of exactly the same height. Estimation of the number of people in a group further relies on the assumption that individuals in a group are separated by a gap of exactly one foot. Neither of these assumptions will be correct in practice.

One way of overcoming these limitations is through direct simulation. Dong et al. [14] used 3D motion capture data of walking humans and rendered the models to the image to compute an average 2D human shape. They then simulated the image appearance of groups of people in the scene. Aside from the motion capture, this approach does not require supervision. However, this lack of supervision also

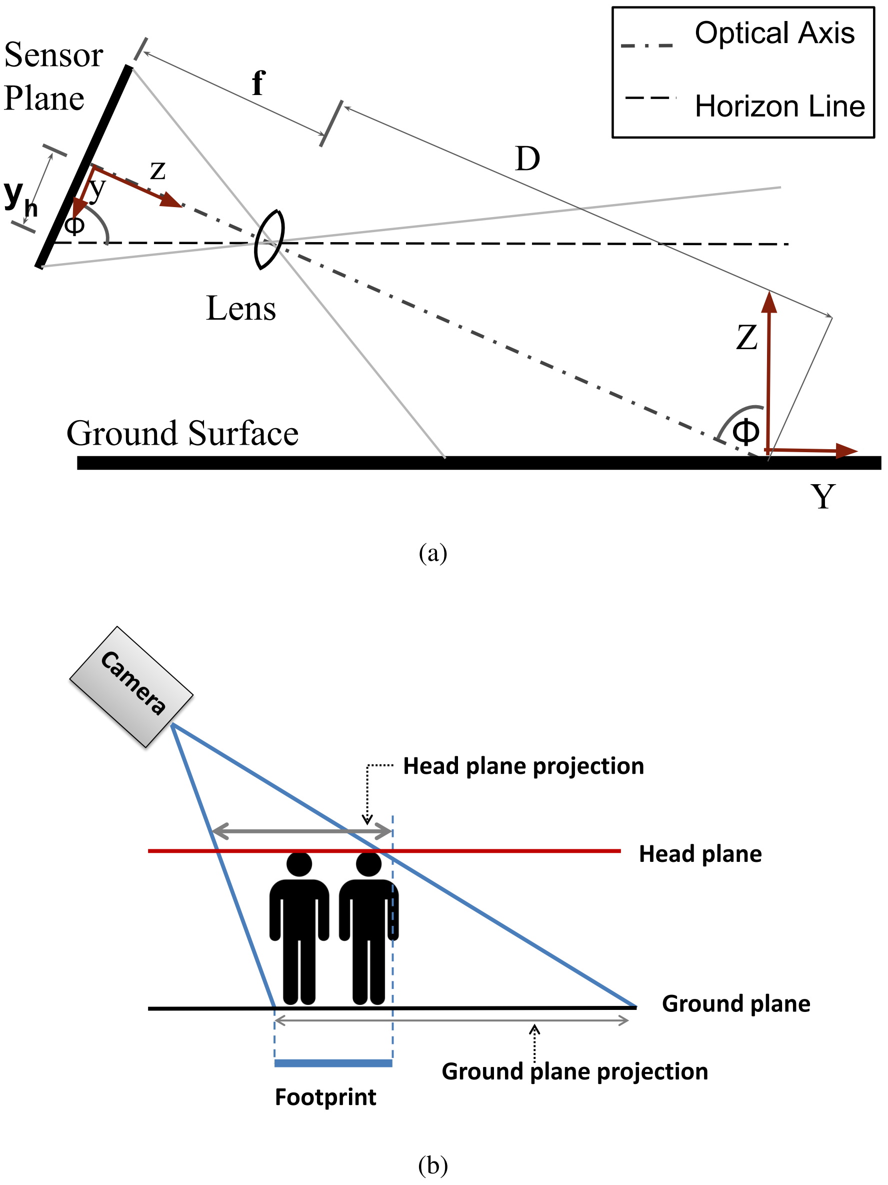


Figure 3.2: (a) Viewing geometry. The light grey lines demarcate the field of view of the camera. The X axis of the world coordinate frame and the x axis of the image coordinate frame point out of the page. (b) Back-projection using head and ground planes.

means that there is no mechanism to account for variations in the appearance of image segments due to partial failures in background subtraction and noise. Also, group size was limited to six individuals: it is unclear whether the method could generalize to larger crowds.

3.2.2 Unsupervised Adaptive Processing

Most prior methods for crowd estimation are supervised [35, 7, 8, 22, 31, 27, 29, 38, 14, 5, 25, 28, 1, 6, 10, 9, 42, 19, 44, 24, 40]. A training dataset is hand-labelled, at least with the number of people in each frame and usually with their locations within the frame. Features of the image (e.g., keypoints, blob descriptors) are then used as predictors and a regressor is learned that can predict the number of people in a frame. As has been noted previously [42] this approach can be problematic, as the regressor may learn an appropriate mapping for the training dataset, but this mapping may not generalize to new scenes, new cameras and new viewing geometries (e.g., tilt angles).

Acknowledging this problem, a few groups have proposed methods that do not involve explicit training [36, 41, 4]. The key problem here is clustering and occlusion in densely crowded scenes. Without supervision, some general principle must be identified that allows features in a connected cluster or blob in the image to be mapped to an estimate of the number of people in the cluster.

Celik et al. [4] fit segments in the image with perspective-scaled rectangular models of fixed size. However this approach is likely to lead to bias, as there is no way to adjust the model based on biases in the observed data (e.g., missing heads or feet), and occlusions and clustering are not handled explicitly.

If the trajectories of individuals can be assumed to be independent, small clusters of features can potentially be tracked over time while each individual is in the field of view [36]. In this way, temporary clusters formed in the image can be disambiguated by integration over time. However this approach does not generalize to dense crowds or social groups, where individuals may walk in close proximity with very similar trajectories.

Rittscher et al. [41] have attempted to improve individuation within clusters by using an adaptive mixture model over rectangular shapes approximating individuals within a cluster. The height and width of the shapes are governed by a Gaussian prior, allowing some adaptability. However, the parameters of the prior must still be selected empirically in advance, reducing generality. Moreover, while the proposed system was never evaluated on (and likely was not intended for) crowd counting per se, as for other individuation approaches [36, 51, 31, 14, 44], it is likely to break down for larger and more dense crowds where occlusions obscure individual features, and may also become computationally prohibitive for large crowds.

3.3 Summary of Contributions

In this chapter, I propose an efficient, fully unsupervised method for crowd counting that handles issues of clustering and occlusion by reasoning in 3D. Following Kilambi et al [27, 28], both perspective scaling and occlusions are accounted for by projecting to the ground plane. However, rather than using training data to learn a fixed model of the appearance of an individual in the image and fixing the spacing between individuals, an adaptive method is used to learn a distribution over the appearance of individuals back-projected to the ground plane. As in Dong et al [14], I use a 3D simulation approach to learn, in unsupervised fashion, how to relate image observations to numbers of people. However, to simplify the analysis the effects of perspective projection are factored out by learning the mapping from detected segments to numbers of people in ground plane coordinates, and simpler features of these segments are used that will generalize to dense crowds. Finally, an adaptive unsupervised method for learning the distribution of individual appearance will account for signal processing noise and errors, avoiding the bias that would otherwise result from hardwiring the human model. The method is fast because the simulation need only be done periodically, as an unsupervised recalibration, while inference amounts to detection of connected segments (background subtraction), back-projection of these to form ground plane footprints, extraction of simple features of these footprints, and linear prediction of the number of people in each group.

To summarize, my primary contributions are:

1. A novel, adaptive, unsupervised method for learning the back-projected appearance of individuals in the scene.
2. A novel unsupervised 3D simulation method that learns how to map back-projected segments to numbers of people.

The advantages of the resulting system are:

1. Parameter-free and unsupervised, allowing generalization to arbitrary scenes, activities, cameras and viewing geometries.
2. Full accounting of the effects of perspective projection and occlusion.
3. Efficient and scalable to arbitrary crowd sizes.

3.4 Algorithm

3.4.1 Overview

Detection of people in the image is based on background subtraction, as is common in crowd estimation systems [43]. Analysis of the resulting foreground segments then unfolds in two stages: an unsupervised learning stage and an inference stage (Fig. 3.3). For present purposes I assume that internal and external camera parameters have been fully identified, as is the norm [43]. In practice, many pan/tilt cameras do not provide motor encoder feedback, however there are a number of good auto-calibration methods for estimating tilt angle online (e.g., [47, 45, 11]) that can be incorporated into the unsupervised learning stage. In the evaluations (Section 3.6) I compare crowd estimation accuracy based on full camera pre-calibration with accuracy using the method proposed in Chapter 2 for camera pose estimation to assess the impact of online calibration errors.

3.4.2 Foreground Segment Detection

I employ the background subtraction algorithm of Elder et al. [17], which is based on a pixel-wise two-component Gaussian mixture model estimated online using an approximation of the EM algorithm. The algorithm operates in the 2D colour subspace spanned by the second and third principal components of the colour distribution, and thus achieves a degree of insensitivity to shadows. Pixels with foreground probability greater than 0.5 are labelled as foreground, and segments are identified as eight-connected foreground regions. Fig. 3.4 shows an example frame on the indoor pedestrian dataset.

Partial inactivity or colour similarities between foreground and background

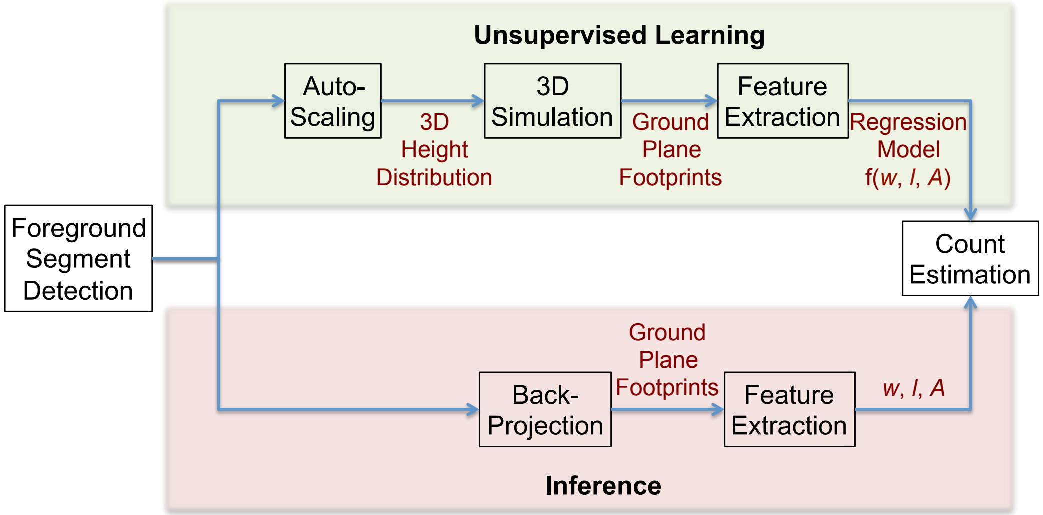
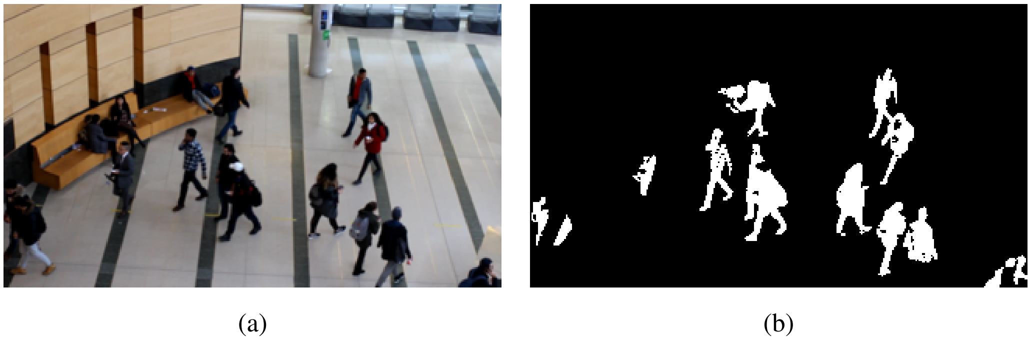


Figure 3.3: Algorithm overview

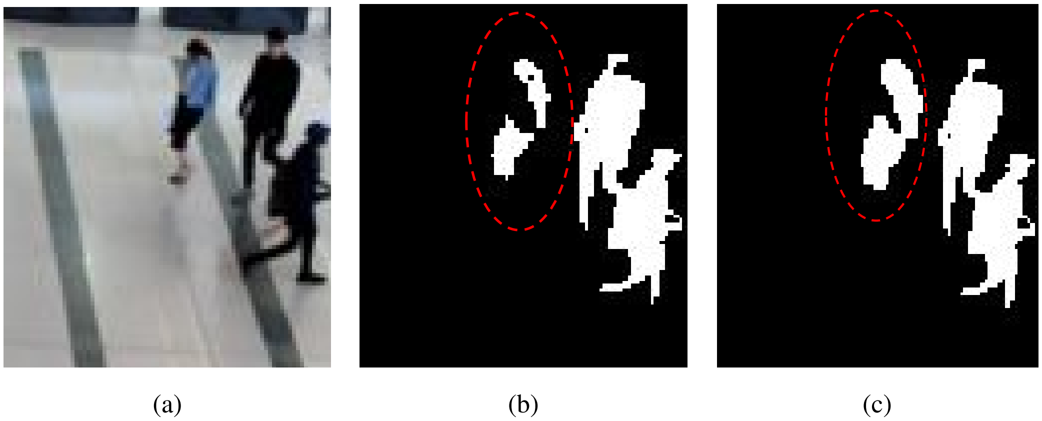
can lead to fragmentation of single individuals into disconnected image segments (Fig. 3.5(b)). To correct for this problem, I take advantage of domain constraints and camera calibration. Specifically all small segments are identified that, when back-projected to the scene, have vertical subtense less than half a normative human height of 1.7m. These small segments are considered candidate fragments and are therefore iteratively dilated until reaching half height. If through dilation a segment has merged with one or more other segments, these segments are assume to project from the same individual/group and the new merged and dilated segment is retained (Fig. 3.5(c)). If no merger has occurred, the segment is assumed to have no partner fragments and is restored to its original state. Note that 1) isolated segments and larger segments are unaffected by this selective dilation (Fig. 3.5(c)) and 2) nominal height is only used here as a rough bound to improve segmentation - heights are more accurately modeled in the auto scaling phase described below.



(a)

(b)

Figure 3.4: (a) Sample frame from indoor pedestrian dataset. (b) Background subtraction result.



(a)

(b)

(c)

Figure 3.5: Selective dilation to correct for fragmentation.

3.4.3 Unsupervised Learning

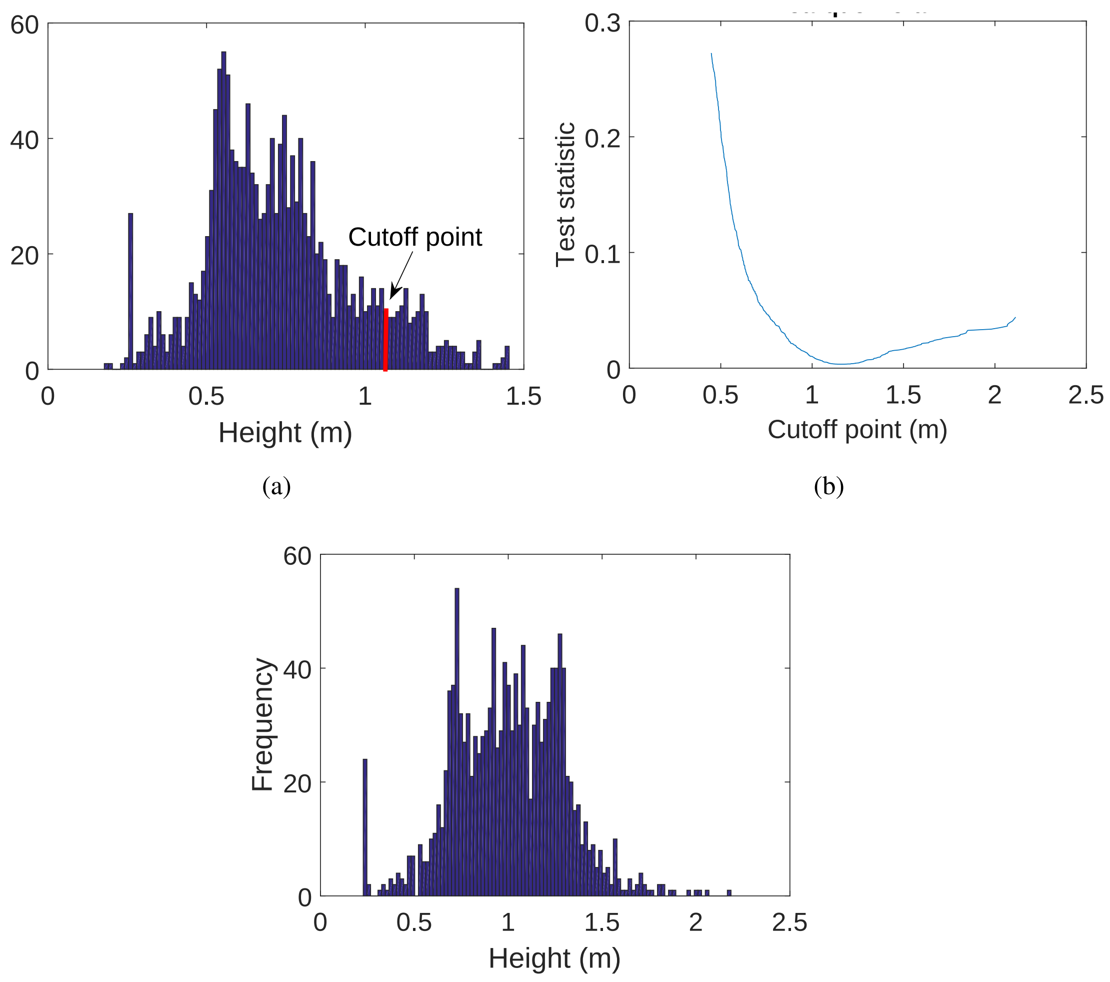
The goal of the unsupervised learning stage of the algorithm is to learn, without labelled training data, how to relate foreground segments in the image to the number of people in the scene. In my experiments, the unsupervised learning and inference stages of the algorithm operate on the same sequence of video frames. (As the method is unsupervised, there is no risk of over-fitting.) In a deployed system running continuously, the unsupervised learning stage would only be invoked periodically to recalibrate the system. The unsupervised learning stage consists of three sequenced computations: auto-scaling, 3D simulation and feature extraction.

Auto-Scaling

Assuming that individuals detected in the image can be mapped directly to a single normative human height in the scene is risky for several reasons. First, human height varies broadly, especially when children are considered. Second, even the best background subtraction algorithm will miss some extremal pixels, leading to segments smaller than predicted, and at other times may include false positive pixels projecting from shadows, leading to segments larger than predicted. In prior work this problem is handled by supervised learning, but this is also risky, as results may not generalize across datasets.

Here I take an unsupervised, adaptive approach based upon the image segments that have been thus far observed. The strategy of the approach is to identify segments that are likely to contain only one individual, and then to use these segments to fine-tune the scaling of the system.

I first compute an upper bound on the 3D height h and width w of each



(a)

(b)

(c)

Figure 3.6: Example of auto-scaling, using the indoor pedestrian dataset. (a) Histogram of square root of the segment areas. (b) Plot of the normalized Jarque-Bera test statistic. (c) Height histogram of image segments below the cutoff point.

back-projected segment. For segments greater than a normative back-projected height of 1.7m, I assume that the bottom pixel projects from the ground plane. For smaller segments, the middle of the segment is assumed to project from the normative mid-body plane (0.85m height). Then, assuming that all pixels of the segment project from the same horizontal distance allows us to estimate back-projected height and width of the segment. Note that the height estimate will be a loose bound for larger segments projecting from groups of individuals, but will tend to be a tight bound for small segments projecting from an individual or part of an individual, and it is these that are most important here (see below). I then use the square root of the product of height and width $\sqrt{hw}$ to represent the scale of the back-projected segment. The resulting distribution of back-projected scales (Fig. 3.6(a)) will generally be composed of a mixture of components from groups of different sizes (number of individuals), but clearly the left part of the distribution will be dominated by groups of size 1 (singletons). The objective is to estimate this component of the mixture, in order to scale the whole distribution.

To do this, I appeal to the central limit theorem, and assume that this component will be close to normal. Thus the Jarque-Bera test statistic [23] is computed for subsets of the distribution on a series of intervals $[0, s]$ as the maximum scale s is varied from 0 to the maximum scale observed. The Jarque-Bera statistic is essentially a weighted sum of skewness and kurtosis, and thus tends to 0 as the distribution approaches normality (Fig. 3.6(b)). It can thus be used to find an appropriate upper cutoff point for the singleton distribution. Selecting all segments below this cutoff allows to compute estimates of the heights for all singletons (Fig. 3.6(c)).

This method for identifying singleton segments will not always be correct:

there may be some group segments that are only partially detected and fall under the threshold, while some singleton segments may cast long shadows and exceed the threshold. However, here I rely only upon the approximate correctness of the *statistics* of the singleton density, which will serve as a generative distribution to sample from in the simulation phase.

3D Simulation

Fig. 3.7 illustrates the 3D simulation process. The fine-tuned distribution of heights estimated by auto-scaling (Step 1) reflects the portion of the human body that was successfully detected. In many cases, this will only reflect part of the full extent of the body, as smaller parts (e.g., head and feet) are often not fully detected. I roughly model this detected portion of the body as a 3D prolate spheroid (ellipsoid with circular symmetry about the major axis), with a 3:1 ratio of the vertical major axis to horizontal minor axes. Sampling from the height distribution thus yields a distribution of ellipsoids of various sizes. In order to simulate crowds, I can now sample fairly from this distribution, placing each ellipsoid randomly and uniformly over the ground plane (Step 2). Sampled ellipsoids that intersect with existing ellipsoids are discarded. Since it is not known which portion of the body was successfully detected, all ellipsoids taller than a normative height of 1.7m are placed on the ground plane, and all ellipsoids less than this height centred at a mid-body height of $1.7/2 = 0.85$m above the ground plane. For the experiments reported here crowds from $n = 1...200$ people are simulated, repeating the simulation 20 times.

In order to relate this to image measurements, these ellipsoids are projected to the image (Step 3). To simplify this step slightly, the projection is approximated

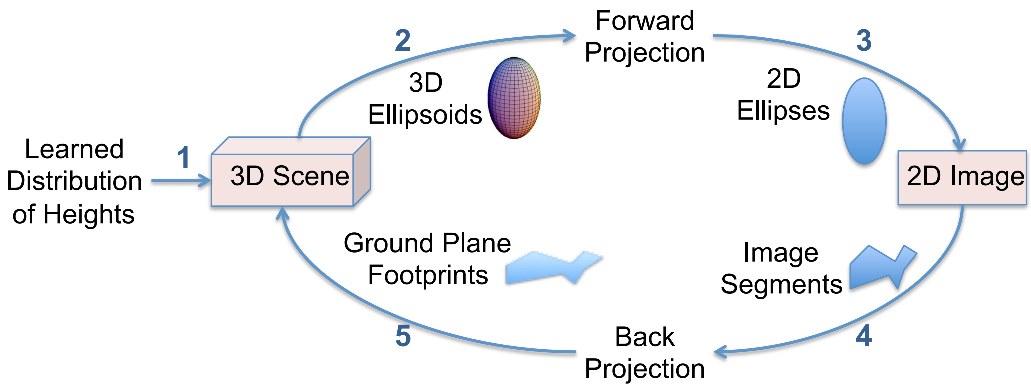
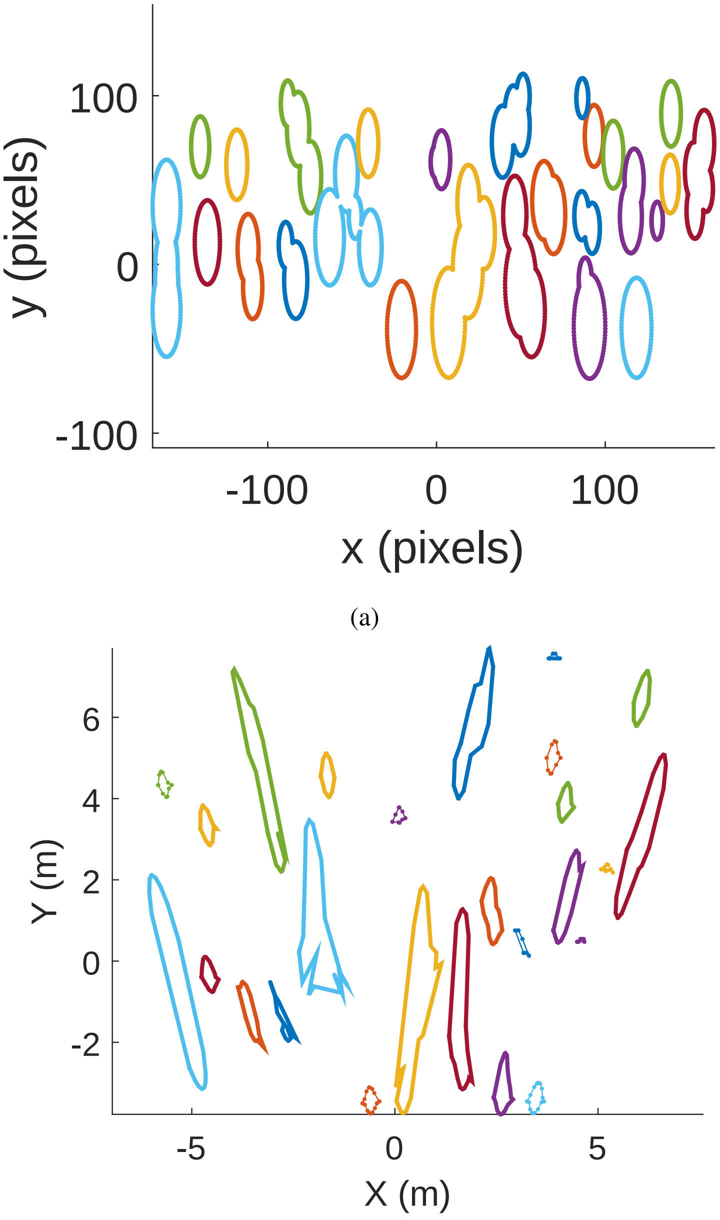


Figure 3.7: Simulation.

here as orthographic, so that each ellipsoid projects as a vertical ellipse. (Prior work suggests that a simple ellipse model can work as well as more complicated body models [14].) Note that this forward projection stage is crucial to modeling the occlusion process: ellipsoids representing distinct individuals in the scene may project to intersecting ellipses in the image, forming larger group segments (Step 4, Fig. 3.8(a)). Both the shape and size of these segments will tend to vary with their radial position on the ground plane and hence their height in the image. To factor this variation out, these segments are back-projected to the ground plane, using an adaptation of the method of Kilambi et al [27, 28] (Fig. 3.2 b).

Specifically, each image segment was partitioned into 10 vertical slices of equal width. Slices that back-project to less than a normative height of 1.7m are localized to a point on the ground plane given by horizontal coordinates of the back-projection of the midpoint of the slice to the mid-body plane (0.85). For slices that back-project to a height exceeding 1.7m, the top pixel is back-projected to the head plane at height 1.7m and the bottom pixel is back-projected to the ground plane. Projecting the top pixel vertically from the head plane to



(a)

(b)

Figure 3.8: (a) Image segments formed by occlusions of ellipsoidal models of human bodies in the simulated scene. (b) Back-projected ground plane footprints. Width and length of a footprint are calculated relative to the ground plane projection of the view vector.

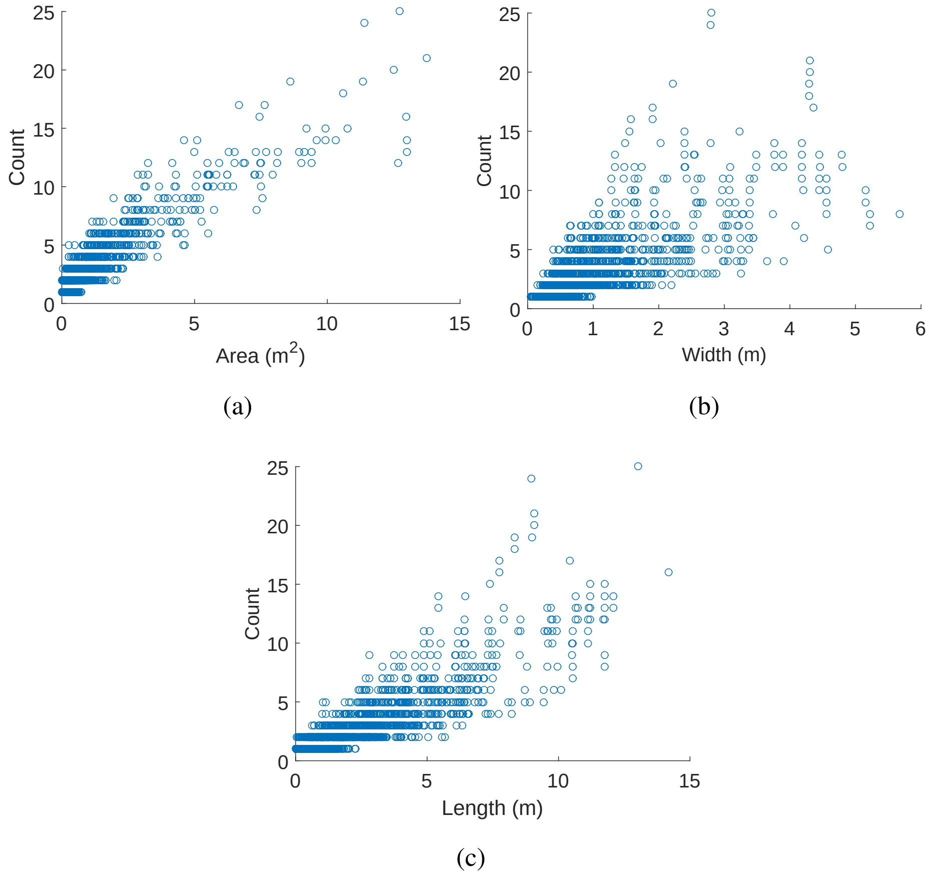
the ground plane thus delimits a radial slice on the ground plane, and sweeping through all 10 slices of the image segment traces a sequence of contiguous slices on the ground plane which together form a polygonal ground plane *footprint* (Step 5, Fig. 3.8(b)). These footprints are expected to be roughly invariant to the location of the group within the scene, and thus variation in the size of the footprint can be largely attributed to the number of individuals within the group.

Feature Extraction

This 3D simulation is used to learn a simple model relating the size of the ground plane footprint to the number of people generating it. Here three simple size features are used : the width w of the footprint in the direction normal to the ground plane projection of the view vector, the length l of the footprint in the direction of the ground plane projection of the view vector, and the area A of the footprint (Fig. 3.8b). The number of individuals in the segment is then predicted as a linear regression on these three variables. Fig. 3.9 shows the projection of this regression on the three variables: from these plots it is clear that the footprint area carries the most information regarding the number of people in the segment.

3.4.4 Inference

Once unsupervised learning is complete, inference is relatively straightforward and fast. Detected segments are back-projected to the ground plane as in simulation, and width, length and area features of the ground plane footprints are computed. These features are then entered into the regression model to compute the estimated number of people in the segment. Summing over all segments in the image yields an estimate of the number of people in the frame.



(a)

(b)

(c)

Figure 3.9: Simulated training data for three variables: (a) area, (b)width and (c)length.

3.5 Datasets

Two datasets are used to evaluate the proposed method. The first dataset is an indoor pedestrian dataset of 170 annotated frames (sample frame shown in Fig. 3.4) recorded with a Canon EOS Rebel T3i camera equipped with a 40mm lens at 30 fps. The frames were down-sampled to 320×182 pixels. The camera/lens system was calibrated in the lab using a standard calibration procedure. The tripod level was used to zero the camera roll and a digital inclinometer was used to accurately measure tilt angle at the scene: $\phi = 60.7$ deg. Camera height was measured with a tape measure and found to be 10.3m. The number of people per frame in this dataset ranges from $n = 5 - 16$.

The second dataset is the public PETS 2009 dataset, available at `www.cvg.rdg.ac.uk/PETS2009`. I selected this dataset as it is the most commonly used to evaluate recent crowd estimation algorithms. Specifically, I evaluated on two sequences of the dataset: 1) View 1 of the S1.L1.13-59 sequence, reporting results for region of interest R0 defined by the original PETS 2009 challenge. 2) View 1 of the challenging S1.L2.14-06 sequence, which involves a very dense crowd with high occlusion levels. Results are reported on the R1 region of interest, defined by the original PETS 2009 challenge. I used the available camera calibration data. The number of people in these sequences ranges from $n = 0 - 38$ per frame.

3.6 Evaluation

The system was implemented in unoptimized MATLAB code. I am confident the method will run in real-time when ported to C. All experiments were conducted on a 4-core desktop computer (3.40 GHz CPU).

Table 3.1: Summary of performance of the proposed method on three datasets. $\bar{n}$ denotes the mean ground truth count per frame.

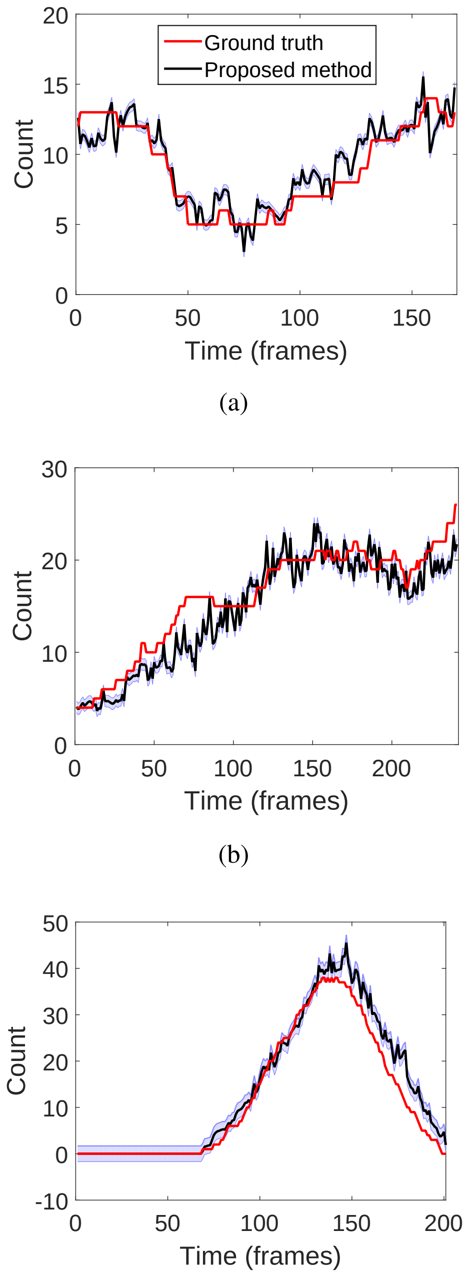
Dataset	$\bar{n}$	MAE	MAE (%)	Bias	Bias (%)
Indoor Pedestrian dataset	9.02	1.04	11.50	0.39	4.30
PETS S1.L1.13-59	15.95	1.97	12.35	-1.2	-7.90
PETS S1.L2.14-06	11.2	2.13	12.2	1.86	9.8

3.6.1 Indoor Pedestrian dataset

Figure 3.10(a) shows the estimated number of people in each frame over time, compared to ground truth. The method performs well, with a mean absolute error (MAE) of 1.04 people per frame (11.50%), and a remarkably low bias (mean signed error) of 0.39 people per frame (4.30%) (Table 3.1). Average runtime for count estimation was 0.04sec per frame.

I used this dataset to assess the contributions of the three features (width, length, area) of the ground plane footprints (Table 3.2). Not surprisingly, the footprint area appears to be most important, but the length and width features also contribute, at least for this dataset.

I also analyzed the feasibility of using the system in combination with an auto camera calibration algorithm. The automatic method for estimating tilt angle from Chapter 2 is employed. The algorithm yielded a tilt estimate for this dataset of 60.02 deg, representing an error of 0.68 deg. Using this biased tilt estimate increased the MAE from 1.04 to 1.19 people per frame, a fairly graceful degradation. This suggests that the method may be reasonably robust to such errors.



(a)

(b)

(c)

Figure 3.10: Performance of the proposed algorithm over time. (a) Indoor pedestrians dataset. (b) PETS S1.L1.13-59 sequence. (c) PETS S1.L2.14-06 sequence

Table 3.2: Performance on indoor pedestrian dataset for different combinations of ground plane footprint features.

Feature	MAE (per frame)
Width	1.96
Length	1.83
Area	1.44
Width + Length	1.65
Area + Width	1.4
Area+ Length	1.13
All	**1.04**

3.6.2 PETS 2009

Figure 3.10(b-c) shows the estimated number of people in each frame over time, compared to ground truth, and Table 3.1 summarizes performance in terms of MAE and bias. MAE and bias for PETS S1.L1.13-59 are comparable to the indoor pedestrian dataset in percentage terms, but somewhat higher for the S1.L2.14-06 dataset. Average runtime for count estimation was 0.31 sec per frame.

Table 3.3 compares the proposed method against state-of-art methods that have reported accuracy on these PETS datasets. The proposed algorithm outperforms all prior methods except for Jeong [24] on the first sequence and the supervised regression method proposed by Conte [9] on both sequences. I note, however, that Conte's results involve temporal smoothing of the count. Median temporal smoothing of my results with a window size of 36 frames also leads to improve-

Table 3.3: Comparison (MAE per frame) with previous algorithms on two PETS 2009 datasets.

Method	S1.L1.13-59	S1.L2.14-06
Albiol [1]	3.86	5.14
Fradi [19]	3.16	2.89
Conte [10]	2.24	4.66
Subburaman [44]	2.08	2.4
Jeong [24]	1.88	–
Conte [9]	1.59	1.99
Proposed method	**1.97**	**2.13**
Proposed method + temporal smoothing	**1.7**	**1.9**

ment (Table 3.3). More importantly, given its unsupervised nature, I expect that the proposed method will generalize more readily to a broad range of conditions (different cameras, tilt angles, illumination etc.). This is already suggested by its strong performance on both indoor and outdoor datasets.

3.7 Discussion

For accurate crowd understanding, the effects of perspective projection must be accurately accounted for. Most prior work handles scaling but not the effects of occlusion. Systems that attempt to model occlusion tend to break down for larger, denser crowds, require extensive supervised training, and make unreasonable assumptions about the people in the scene. In this chapter I have shown that through a periodic 3D recalibrating simulation of the scene, the effects of perspective projection and occlusion can be accurately accounted for. A key to accuracy is to exploit the appearance of singletons in the dataset that allow the simulation to be properly scaled. The result is a highly efficient inference method that does not require training, has remarkably low bias and scales easily to denser crowds.

One potential limitation of this work is that it requires the observation of one or more singleton (image segment projecting from a single person). In very dense crowds, this may not occur. However, I stress that this is only necessary for the periodic unsupervised learning phase. In extended surveillance scenarios, recalibration can be timed to coincide with sparser crowds, automatically detected by analyzing the distribution of back-projected segment heights.

I see many opportunities to improve the method. In the 3D simulation all smaller ellipsoids are centered at mid-body height. A more accurate regression model might be learned by randomizing the vertical offset of these smaller ellipsoids uniformly between ground plane and head plane contact. I also intend to relax the prolate spheroid model to allow more general triaxial ellipsoids, learning (without supervision) a distribution over the three distinct axis dimensions that best explains the observed singleton image segments. The proposed system uses width, length and area features of the ground plane footprints as predictors of the number of people in the segment. I have not systematically explored other features - there may be additional information in the shape of the footprint that would improve performance.

Finally, tracking image segments over time would allow counts to be smoothed independently for each segment and this might yield greater accuracy.

Chapter 4

Conclusion

There is a growing interest from many application domains such as surveillance, urban planning, transportation and public management to migrate from laser technology and LIDAR systems to using cameras equipped with computer vision software for accomplishing various surveillance and data collection tasks. This highlights the importance of developing computer vision methods that function successfully and reliably in real world applications. Specifically, methods that generalize well to different scenarios with minimal assumptions on regularities in structure, motion patterns and appearance are desirable in that context. While most surveillance cameras are pan-tilt-zoom cameras the majority are not equipped with encoders that can inform computer vision algorithms of the current pose of the camera. Thus auto calibration methods are critical in real world applications. Those requirements outline the key elements of the proposed work: An accurate auto calibration algorithm to provide updated camera pose information coupled with an efficient unsupervised crowd counting method, that generalizes well to various scenes and conditions. Accordingly, a crucial part of the evaluation in Chapter 3 is measuring the effect of employing the proposed auto calibration method on the counting accuracy. Such systems can be deployed in various public places like shopping malls, public transportation stations and sport stadiums to provide accurate crowd counts. This information is critical for a wide range of tasks such as enforcing maximum capacity regulations or crowd and disaster management. Integration with a more general crowd behavior analysis system will facilitate further tasks such as social analysis on crowds and detecting major motion patterns.

In this thesis I presented a method for efficient crowd counting that is fully unsupervised and relies upon occasional 3D simulation of the scene to learn the mapping between simple features of foreground regions and the count of people. While most prior methods use perspective maps that correct for distance scaling but do not fully account for occlusion effects, I propose a framework for modeling those effects in 3D by projecting the simulated agents to the image plane and then back-projecting to a footprint region in the ground plane. An algorithm for automatically adapting to signal processing errors in foreground detection allows the system to generalize well to different scenarios. Finally, by avoiding detection and tracking of individual human features the algorithm can easily scale to large numbers of people.

Utilizing 3D geometry requires knowledge about camera parameters. While most existing methods for automatic camera calibration rely on structural scene information, specific motion patterns or object tracking to solve this problem, those features are not necessarily present in unstructured crowded scenes or cannot be reliably detected. Here, I present a novel and very general method for recovering camera tilt from image motion in an unsupervised manner. Unlike prior methods, the proposed algorithm does not depend upon the visibility of regular static structures in the scene and does not require segmentation, shape analysis or feature tracking, thus reducing the required computation. This method does not require that objects move in the same direction or at constant velocities. Rather, it rests on the much more general assumption of zero correlation of speed with position in the direction projecting to the vertical axis of the image. A novel method for automatically and adaptively selecting the optimal subset of motion vectors generated by the objects moving in the scene means that the algorithm does not require training and is completely parameter free. This allows the algorithm to be applied to diverse scenarios without reconfiguration.

4.1 Future Work

In Chapters 2 and 3 I discussed opportunities for improvements to the proposed methods for camera pose estimation and crowd counting, respectively. Future work for the former includes extending the method to recover roll angle and developing complementary methods for focal length estimation. Further improvements on the crowd counting method include incorporating a more complex spheroid model, exploring further ground plane footprint features and tracking image segments over time.

More generally, I see various directions for building up on this work. Currently, I assume a uniform distribution of people over the ground plane. In reality, density usually varies across the scene depending upon different factors like the environment and the social groups within the crowd. Thus the system could be extended to learn the distribution of people in the scene. This can be used as an output of the system to facilitate further crowd analysis tasks such as social group identification and abnormality detection. Moreover, the learned distribution over the ground plane could be incorporated into the 3D simulation stage to simulate a crowd that is more similar to the crowd in the input images.

The auto-calibration method has been evaluated on datasets with different crowd densities ranging from sparse crowds, e.g. Indoor Pedestrian dataset, to extremely dense crowds, e.g. Marathon dataset. However, the crowd counting method has been evaluated on sparse and moderate density crowds. Evaluation on extremely dense crowd datasets such as the Marathon dataset was difficult due to the need for ground truth data in the form of people count per frame, which is a laborious task for images with such large number of people. Additionally, publicly available large scale crowd datasets do not provide focal length and camera height parameters, which are input to the proposed system. Future work could include performance evaluation of the proposed system on the Marathon dataset after obtaining the ground truth data. This will determine the applicability of the system for massive crowd management tasks.

Finally, integrating the output of the camera pose estimation and the crowd analysis methods with a more general system for 3D simulation of urban scenes, e.g. Corral-Soto et al [12], would allow for improved interpretation and visualization of scene dynamics.

Bibliography

[1] Antonio Albiol, Maria Julia Silla, Alberto Albiol, and Jos'e Manuel Mossi. Video analysis using corner motion statistics. *Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 31–38, 2009.

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[3] Biswajit Bose and Eric Grimson. Ground plane rectification by tracking moving objects. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 94–101, 2003.

[4] Hasan Celik, Alan Hanjalic, and Emile A Hendriks. Towards a robust solution to people counting. In *Image Processing, 2006 IEEE International Conference on*, pages 2401–2404. IEEE, 2006.

[5] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or

tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[6] Antoni B Chan, Mulloy Morrow, and Nuno Vasconcelos. Analysis of crowded scenes using holistic properties. *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, pages 101–108, 2009.

[7] Siu-Yeung Cho and Tommy WS Chow. A fast neural learning vision system for crowd estimation at underground stations platform. *Neural Processing Letters*, 10(2):111–120, 1999.

[8] Siu-Yeung Cho, Tommy WS Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(4):535–541, 1999.

[9] D. Conte, P. Foggia, G. Percannella, and M. Vento. A method based on the indirect approach for counting people in crowded scenes. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 111–118, Aug 2010.

[10] Donatello Conte, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento. A method for counting people in crowded scenes. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 225–232. IEEE, 2010.

[11] Eduardo R Corral-Soto and James H Elder. Automatic single-view calibration and rectification from parallel planar curves. In *Computer Vision–ECCV 2014*, pages 813–827. Springer, 2014.

[12] Eduardo R Corral-Soto, Ron Tal, Langyue Wang, Ravi Persad, Luo Chao, Chan Solomon, Bob Hou, Gunho Sohn, and James H Elder. 3d town: the automatic urban awareness project. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 433–440. IEEE, 2012.

[13] Jonathan Deutscher, Michael Isard, and John MacCormick. Automatic camera calibration from a single manhattan image. In *Computer Vision—ECCV 2002*, pages 175–188. Springer, 2002.

[14] Lan Dong, Vasu Parameswaran, Visvanathan Ramesh, and Imad Zoghlami. Fast crowd segmentation using shape indexing. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[15] Marius Drulea and Sergiu Nedevschi. Motion estimation using the correlation transform. *Image Processing, IEEE Transactions on*, 22(8):3260–3270, 2013.

[16] Marketa Dubska, Adam Herout, Roman Juranek, and Jakub Sochor. Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1162–1171, 2015.

[17] J. H. Elder, S.J.D. Prince, Y. Hou, M. Sizintsev, and E. Olevskiy. Pre-attentive and attentive detection of humans in wide-field scenes. *International Journal of Computer Vision*, 72(1):47–66, 2007.

[18] Duc Fehr, Ravishankar Sivalingam, Vassilios Morellas, Nikolaos Papanikolopoulos, Osama Lotfallah, and Youngchoon Park. Counting people in groups. In *Advanced Video and Signal Based Surveillance, 2009.*

AVSS'09. Sixth IEEE International Conference on, pages 152–157. IEEE, 2009.

[19] Hajer Fradi and Jean-Luc Dugelay. Low level crowd analysis using framewise normalized feature for people counting. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, pages 246–251. IEEE, 2012.

[20] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[21] Michael Hodlmoser, Branislav Micusik, and Martin Kampel. Camera autocalibration using pedestrians and zebra-crossings. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1697–1704. IEEE, 2011.

[22] D Huang, Tommy WS Chow, and WN Chau. Neural network based system for counting people. In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, volume 3, pages 2197–2201. IEEE, 2002.

[23] C.M. Jarque and A.K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3):255–259, 1980.

[24] Chi Yoon Jeong, SuGil Choi, and Seung Wan Han. A method for counting moving and stationary people by interest point classification. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4545–4548. IEEE, 2013.

[25] Michael J Jones and Daniel Snow. Pedestrian detection using boosted features over many frames. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[26] Neeraj K Kanhere and Stanley T Birchfield. A taxonomy and analysis of camera calibration methods for traffic monitoring applications. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):441–452, 2010.

[27] Prahlad Kilambi, Osama Masoud, and Nikolaos Papanikolopoulos. Crowd analysis at mass transit sites. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 753–758. IEEE, 2006.

[28] Prahlad Kilambi, Evan Ribnick, Ajay J Joshi, Osama Masoud, and Nikolaos Papanikolopoulos. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding*, 110(1):43–59, 2008.

[29] Dan Kong, Doug Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187–1190. IEEE, 2006.

[30] Paul Kuo, Jean-Christophe Nebel, and Dimitrios Makris. Camera autocalibration from articulated motion. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 135–140. IEEE, 2007.

[31] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885. IEEE, 2005.

[32] S.F. Lin, J.Y. Chen, and H.X. Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 31(6):645–654, November 2001.

[33] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 562–567. IEEE Comput. Soc, 2002.

[34] Ruihua Ma, Liyuan Li, Weimin Huang, and Qi Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170–173. IEEE, 2004.

[35] Aparecido Nilceu Marana, Luciano da Fontoura Costa, RA Lotufo, and Sergio A Velastin. Estimating crowd density with minkowski fractal dimension. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3521–3524. IEEE, 1999.

[36] Osama Masoud and Nikolaos P Papanikolopoulos. A novel method for tracking and counting pedestrians in real-time using a single camera. *Vehicular Technology, IEEE Transactions on*, 50(5):1267–1278, 2001.

[37] N. Paragios and V. Ramesh. A MRF-based approach for real-time subway monitoring. *Proceedings of the IEEE Computer Society Conferene on Computer Vision and Pattern Recognition*, 1:1034–1040, 2001.

[38] Vincent Rabaud and Serge Belongie. Counting crowded moving objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 705–711. IEEE, 2006.

[39] Hidayah Rahmalan, Mark S Nixon, and John N Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540–545. IET, 2006.

[40] Shirine Riachi, Walid Karam, and Hanna Greige. An improved real-time method for counting people in crowded scenes based on a statistical approach. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 2, pages 203–212. IEEE, 2014.

[41] Jens Rittscher, Peter H Tu, and Nils Krahnstoever. Simultaneous estimation of segmentation and shape. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 486–493. IEEE, 2005.

[42] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. Scene invariant crowd counting. In *International Conference on Digital Image Computing: Techniques and Applications*, pages 237–242, 2011.

[43] David Ryan, Simon Denman, Sridha Sridharan, and Clinton Fookes. An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding*, 130:1–17, 2015.

[44] Venkatesh Bala Subburaman, Adrien Descamps, and Cyril Carincotte. Counting people in the crowd using a generic head detector. In *Advanced*

Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on, pages 470–475. IEEE, 2012.

[45] Ron Tal and James H Elder. An accurate method for line detection and manhattan frame estimation. In *Computer Vision-ACCV 2012 Workshops*, pages 580–593. Springer, 2013.

[46] Elena Tretyak, Olga Barinova, Pushmeet Kohli, and Victor Lempitsky. Geometric image parsing in man-made environments. *International Journal of Computer Vision*, 97(3):305–321, 2012.

[47] Horst Wildenauer and Allan Hanbury. Robust camera self-calibration from monocular images of Manhattan worlds. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2831–2838. IEEE, 2012.

[48] Li Xu, Jiaya Jia, and Yasuyuki Matsushita. Motion detail preserving optical flow estimation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1293–1300. IEEE, 2010.

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

[50] Z. Zhang, M. Li, K. Huang, and T. Tan. Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[51] Tao Zhao and Ram Nevatia. Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–459. IEEE, 2003.