This is the published version:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30044574

Reproduced with the kind permissions of the copyright owner.

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright : 2008, IEEE
Pedestrian Detection for Mobile Bus Surveillance

Wilson S. Leoputra
Curtin University of Technology
Perth, Australia
w.leoputra@postgrad.curtin.edu.au

Svetha Venkatesh, Tele Tan
Curtin University of Technology
Perth, Australia
{s.venkatesh, t.tan}@curtin.edu.au

Abstract—In this paper, we present a system for pedestrian detection involving scenes captured by mobile bus surveillance cameras in busy city streets. Our approach integrates scene localization, foreground and background separation, and pedestrian detection modules into a unified detection framework. The scene localization module performs a two stage clustering of the video data. In the first stage, SIFT Homography is applied to cluster frames in terms of their structural similarities and second stage further clusters these aligned frames in terms of lighting. This produces clusters of images which are differential in viewpoint and lighting. A kernel density estimation (KDE) method for colour and gradient foreground-background separation are then used to construct background model for each image cluster which is subsequently used to detect all foreground pixels. Finally, using a hierarchical template matching approach, pedestrians can be identified. We have tested our system on a set of real bus video datasets and the experimental results verify that our system works well in practice.

Index Terms—Homography, Scene Localization, Non-Parametric Background Modeling, Hierarchical Template Matching.

I. INTRODUCTION

Large scale urban surveillance has traditionally relied on the deployment of many static cameras. However, it is expensive to populate large cities with such infrastructures. A recent surveillance system allows the use of front facing cameras mounted on bus fleets in cities to collect enormous amount of video data over time. The VirtualObserver [1] technology allows the retrieval of the footage for a given GPS location and time of interest. It returns “time-ordered” sequence of footage acquired by a number sensors, in a pseudo time-line, as shown in Figure 1. The footage at a given GPS location, returned by a number of buses can be different because: (a) GPS data may be noisy (b) The sensor in one bus may have different view point to that in another bus (c) Even if the footage for a sensor is amalgamated, it may view the environment from slightly different positions in the road (e.g. different lane). Further for each sensor, the data is taken under different lighting conditions, contains moving scenes, and is of low frame rate low resolution, imposing great challenges for foreground object detection from these video images.

Previous works on foreground extraction, such as [2], [3], [4], [5], cannot be applied on this data due to the strict assumptions that the images have to be captured by static cameras and have to have similar background lighting. Similarly, the direct pedestrian detection methods [6], [7] often encounter major problems, having high false positives.

In this paper, we present a new foreground detection system that tackles these challenges. Our approach integrates scene localization, foreground-background separation, and pedestrian detection modules into a unified detection framework. The scene localization module transforms sequences of video data into arrays of images grouped as “view sites”. A view site is a “unique region” storing images captured from different times, but having highly similar background structure and lighting (i.e. the images are aligned in static-camera-manner), which can then be used for foreground-background separation. To do this, we first use the VirtualObserver to retrieve all video images captured at a particular GPS location. Then, SIFT-Homography is applied to effectively estimate the scene-to-view-site mapping, measured by the scene structural similarities. For each view site, the images are further clustered according to their background intensities to produce clusters of images having similar background lighting. With the large amount of data, we are able to generate a rich profile of spatially aligned images for any view site along the bus route. This way, kernel density estimation (KDE) methods for colour and gradient foreground-background separation can be used to construct the background model for each intensity cluster and used to detect foreground objects. Finally, using a hierarchical template matching approach on the foregronds from KDE,
pedestrians can be identified.

The significance of our proposed system is that it provides a basic framework for automatic large-scale mobile surveillance applications and facilitates many higher-level applications such as crime investigation, crowd identification, query report generation, etc.

This paper is organised as follows. In the following section, related work is briefly summarized. In Sections III and IV, we describe our proposed approach in detail. Section V presents the experimental results of our proposed method compared with existing techniques. Section VI concludes the paper.

II. RELATED WORK

Foreground-background segmentation: This approach normally models the statistics of empty background and then assigns the pixels that change relatively “far” from the background model as the foreground. Examples include [2], [8], [9], [3]. Among them, parametric approaches for statistical modeling include Stauffer and Grimson [2] who use a k-Gaussian model to express the distribution of the background pixels, while Noriega et al. [9] use a local kernel histogram approach. On the other hand, Javed et al. [3] propose a hierarchical background model that combines both colour and gradient to detect foreground and Elgammal [8] propose a non-parametric approach to model the background distribution.

Hou et al. [4] assumethat the background can be observed as the highest frequency and hence propose a background reconstruction algorithm based on accepting cluster of frequent pixel values. Pan et al. [5] further extend this technique by incorporating two background models, one for backgrounds that appear for long durations, while the other acts as feedback information for motion detection (shorter term). The last two techniques belong to the background model reconstruction category.


Apart from the above literature, few works have addressed the same application as ours. One approach by Leibe et al. [13] who use stereo cameras on moving vehicles to detect foregrounds and recover 3D information.

III. SCENE LOCALIZATION

The video is captured using a VirtualObserver, and a two stage clustering is performed: Level 1, extracts for each sensor, frames that are aligned in terms of structure, and Level 2, further clusters these aligned frames in terms of lighting. This two stage approach caters for differential viewpoint (Level-1 clustering) and differential lighting (Level-2 clustering). Figure 2 shows the clustering stages. The clustering results at Level-2 are then used to learn the background model for each sensor.

A. Location based clustering (Level-1 Clustering)

The VirtualObserver [1] performs geometric queries, given a GPS location $l$ and radius precision $r$. It returns all geo-referenced tracks and associated video footage that satisfies the query. For a given GPS location, let the footage from sensor $i$ be amalgamated. Let the pseudo time-line block index, as shown in Figure 1 be $q_i$, with $n_q$ number of frames in each block. Then, any frame can be indexed as $f(i, q, u)$ or $f_{q_i, u}$, where $1 \leq u \leq n_q$.

We define a view site, $v_{\theta, \psi}$, as a cluster of images having both similar scene structure and lighting, where $\theta$ and $\psi$ are the corresponding indexes. Hence any frame in a view site can be indexed as $v(\theta, \psi, c)$, where $c$ is the frame index.

In Level-1 clustering, the task is to find the mapping of a set of frames $f_{q_i}^t$ to the “most appropriate” view site $v(\theta, -)$ from $V$, the entire view sites. The index $(\theta, -)$ indicates that all images in this stage are clustered only based on the scene structural similarities. This can be done by a process of incremental clustering. The measure of similarity between the cluster center in a view site $v_{\theta, \psi, c}$ and an incoming frame $f_{q_i, u}$ is calculated as

$$s(c_{v_{\theta, \psi, c}}, f(i, q, u)) = \homog(v(\theta, -), f_{q_i, u})$$

where $s(.)$ is the similarity function for the clustering algorithm, $\homog(.)$ is a function to compute a Homography matrix $H$, between a view-site $v(\theta, -)$ and an image $f_{q_i, u}$. This function is defined as follows: given two points $b$ and $b'$, which correspond to the same point in $v(\theta, -)$ and $f_{q_i, u}$ respectively, then there exists a 3x3 Homography matrix $H$ such that $b = Hb'$ [14]. The Homography matrix can be estimated using the least squares algorithm, given at least 4 or more corresponding points. Since one would expect the illumination conditions between the matching image pair to vary considerably, features that are resilient to illumination variations are used. In the implementation, point matching is defined as the correspondence between the stable SIFT features from two scenes [15], [14]. In order to achieve a high alignment accuracy, we impose a strict constraint on $H^*$, matrix with the least error in its transformation, to only accept small changes between the two scenes. In this implementation, we restrict the deviation of the scene transformation between
the scene and the view sites to 2%. Given baseline view sites, this alignment process can be adapted to register further images collected from different sensors (i.e., buses) taken at different times, generating a rich profile of spatially aligned images for any view site along the bus route.

B. Intensity based clustering (Level-2 Clustering)

After Level-1 clustering process, we obtain images that are grouped based on the background structural similarity but under different lighting conditions (for example: images that are aligned capturing a store which is closed during early mornings but opened during the afternoons). In Level-2 clustering, the aim is to further cluster images into sub-clusters of images having similar background intensities.

A typical approach for image intensity grouping is to compute the closeness or differences of the global intensities between two images, like the spatial envelope representations in [16]. If the global intensity differences are small, then the two images are considered to be similar and assigned in one group. However, this approach is limited to images that contain predominantly background information only. In our case, we want to avoid computing these differences when the foreground objects are present. We overcome this problem by computing only the differences of pixels, in which the SIFT features at those pixels location correspond to each background pixel. Using SIFT matching, we can control the fact that the corresponding pixels between two images always belong to the background, and only the differences between these background pixels are computed.

Formally, the task is to break down the frames in the first level view site \(v_{\theta,-} \) into a set of sub-clusters \(v_{\theta,\psi}\) having images with similar global intensity values. This again can be achieved by incremental clustering. Let \(\{(b_k, b'_k) : k = 1, \ldots, N\}\) be the same SIFT keypoints (background pixels) in the intensity cluster center \(c_{\theta,\psi}\) and the image \(v(\theta, -, e)\), i.e. the keypoint and \(b'\) correspond to each other. Let \(N\) be the total number of SIFT keypoints. Thus, we define the measure of similarity between \(c_{\theta,\psi}\) and \(v(\theta, -, e)\) as the sum of the normalized differences of the region surrounding SIFT features:

\[
s(c_{\theta,\psi}, v(\theta, -, e)) = \frac{1}{N} \sum_{k=1}^{N} |\text{hoi}(b_k) - \text{hoi}(b'_k)|
\]

where \(b\) and \(b'\) denote the same SIFT keypoints (background pixels) that appear in \(c_{\theta,\psi}\) and \(v(\theta, -, e)\) respectively, \(\text{hoi}(.)\) is a function that computes the Gaussian histogram-of-intensity (G-HOI) patches around each background pixel. In SIFT, each local keypoint descriptor is represented as a weighted Gaussian and interpolated histogram-of-gradient orientations (G-HOG) [15], [7]. We adopt the same idea, but apply it on the intensity instead (G-HOI). We divide the region into 8 patches organized by a \(2 \times 2\) grid, each with 16 bins. The histogram is then constructed by accumulating the weighted Gaussian kernel around the center of each keypoint. The resulting intensity patch descriptor is a 84 dimensional vector (\(2 \times 2 \times 16\)).

IV. COMBINED FOREGROUND BACKGROUND SEPARATION AND PEDESTRIAN DETECTION

Since our objective is to detect the foreground regions of a given scene, we will use the images taken from the intensity cluster from each view site, \(v_{\theta,\psi}\), to assist in segmentation. We formulate the problem of foreground-background separation using a non-parametric approach, similar to [8].

A. KDE on colour

Given a set of images \(v_{\theta,\psi} = \{f_1, \ldots, f_n\}\) in a specific view-site indexed by \(\theta\) and \(\psi\), then the background model can be estimated using the Parzen window technique [17]. We assume that the backgrounds correspond to the highest pixel occurrences and the KDE method with a Gaussian kernel is used to construct the background model. Let \(x_1, \ldots, x_n\) be a sample of intensity values for a particular pixel on the entire image set \(v_{\theta,\psi}\), in \(d\) dimensional space, from a multi-variate distribution \(P(x)\), then an estimate of \(\hat{P}(x)\) can be calculated as

\[
\hat{P}_H(x) = \frac{1}{n} |H|^{-\frac{d}{2}} \sum_{i=1}^{n} K(H^{-\frac{1}{2}}(x - x_i))
\]

where \(K\) is a kernel estimator function and \(H\) is a symmetric \(d \times d\) kernel bandwidth matrix. In the case where the kernel estimator is a Gaussian with \((0, \Sigma)\), i.e. \(H = \Sigma\). When dealing with colour (RGB) distributions, we can assume that each colour is independent of each other, thus \(\Sigma = \text{diag} \{\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2\}\). The final density estimation is written as

\[
\hat{P}_\sigma(x) = \frac{1}{n} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i - e_j)^2}{2\sigma_j^2}}
\]

Intuitively, Equation 3 can be interpreted as computing the pixel occurrence across \(n\) samples, weighted using a Gaussian kernel around the central pixel with a fixed standard deviation. We choose three \(x^*\) that have highest occurrences as the possible background pixels and the rest are set to foreground. We further check if the probability score, \(\hat{P}(x^*)\) is greater than a certain threshold, to assign them as background pixels, else they are considered as foreground pixels.
The results of this process is a separation of foreground and background pixels. However, false positives may be introduced as a result of image misalignment (from the scene localization process). Since these misalignments are due to camera motion, we can solve this problem by analyzing the neighborhood properties, as motivated by the work of [8]. If the correlation of a pixel to its neighborhood is high, then it is most likely to be a background pixel that has “moved” to a new location, otherwise, it is a foreground pixel that occludes the background. We define this probability as the neighborhood probability. Let $x_f$ be the observation of pixel $x$, that is detected as a foreground pixel. The neighborhood probability is defined as:

$$P_N(x_f) = \max_{y \in N(x)} P(x_f|B_y)$$

where $B_y$ is the background model at pixel location $y$ and $P(x_f|B_y)$ is computed using Equation 3. Although this approach addresses the misalignment problem, it introduces the problem of losing true detections, as mentioned by [8]. Hence we incorporate a further constraint on the neighborhood conditions, i.e. if a pixel is truly a background pixel that happens to move to a new location, then some of its neighborhood pixels should also move, since we assume that the misalignment happens not only for one single pixel, but for a group of pixels (as seen in Figure 3). Therefore, we define the probability of displacement as:

$$P_C(x_f) = \prod_{y \in \{x_f \cup N(x_f)\}} P_N(y) \quad (4)$$

where $P_C(x_f)$ is computed as the product over the connected components of the neighborhood pixels. Therefore, a pixel $x$ is considered as background only if both $P_N(x) > th1$ and $P_C(x) > th2$.

B. KDE on gradient

We build a gradient background model in a similar way to the colour based approach described in Section 4.2. First, we transform the colour image into an intensity image and then compute the edge (gradient) image using $\Delta(I) = \sqrt{d^2_x + d^2_y}$, where $\Delta$ is the magnitude gradient image, $d_x$ and $d_y$ are the horizontal and vertical rates of changes of $I$ respectively. Once we obtain a set of gradient images, we employ the same kernel density estimation method to search for the maximal occurrence of the edge pixels and mark them as background edges.

C. Combined KDE color and gradient

The results of the colour-based and gradient-based approaches are finally combined to determine the foreground regions. This approach can be outlined as follows: given a set of foreground regions in colour and gradient images respectively, we assume that a region is a foreground if and only if they are both strong in the edge and gradient foregrounds. A similar method has been used in [3]. Therefore we compute the combined score of a given foreground region as:

$$score = \frac{1}{|G|} \sum_{i \in \Delta R_a} \Delta I(i) \cdot G(i)$$

where $I$ is the original gradient image, $G$ is the foreground gradient image, $R_a$ is the foreground region. If the score is less than a threshold then the region is considered as a background.

D. Pedestrian detection

We assume that a pedestrian walks vertically on the ground plane. Under this assumption, we employ a similar approach for detecting pedestrians as described in [10], and apply it on the KDE foreground-background separation results. First, we construct a hierarchical set of templates for head, torso, and legs. During the detection, an optimized Bayesian formulation is used to find the best template configuration that matches the pedestrian. This hierarchical template matching can be performed efficiently for three reasons: First, using the hierarchical approach, we truncate many unnecessary matches. Secondly, the result of the foreground-background separation allows us to narrow down the search for pedestrian locations in the frame of interest. And finally, the pedestrian’s scale (height) can be estimated using a priori information like offline foot-to-head Homography calibration. An example of the calibration process is shown in Figure 4.

V. EXPERIMENTAL RESULTS

In order to verify the performance of our proposed system for foreground detection, we conduct a series of experiments using real world bus data. The experiments are comprised of two parts: (a) the comparison of our proposed approach with other approaches, in terms of foreground detection (b) the performance evaluation of our system on different bus datasets. The proposed algorithm is implemented in C++ and tested offline using a Pentium 3.00 GHz machine.

A. Dataset Description

We tested our system on 4 sets of video streams taken by the outward facing cameras mounted on buses. The selected buses were BlueCat buses plying the major routes within City of Perth, Western Australia. The onboard GPS records the location of the bus synchronized with video frames. We have chosen subsets of video footage and the detailed information is described in Figure 5.

Each dataset consists of video images recorded at 10 fps with a resolution of $384 \times 288$ pixels. Regions 1 and 2 have
a coverage radius of 250 metres as shown in Figure 5. Each region dataset is collected on different dates and times (one is on May 2007 and the other is taken at November 2007). We then evaluate the system performance separately based on the region. The first region dataset captures images on busy streets, including cluttered complex backgrounds (for example shops) and crowded pedestrians, while less frequent pedestrian traffic occurs in the second region dataset. For each region, we first perform the scene localization, construct the background model on 80% of the datasets (for both May and Nov), extract the foregrounds on the remaining 20%, and finally detect the pedestrians. Also, we process the video images specific to each sensor, avoiding the variability of the different intrinsic camera parameters.

### B. Scene localization results

From the above datasets, we have successfully registered 189 view-sites at Region 1 and 212 view-sites at Region 2 using the proposed method. At each view-site, the data is further registered based on intensity values, with the number of frames varying from 64 to 233. Figure 8 shows an example of the registered results from location to view-sites and then to intensity clusters. As can be seen, Figure 8-(left)(Level-2) shows significant image differences due to time of the day differences. One group has the store door closed in the mornings and the other group has the door opened in the afternoons. Similarly, Figure 8-(right)(Level-2) also shows the different lighting levels in the background. The first group has lights switched on, while the second group has the lights off. Figure 8-(middle)(Level-2) shows images taken from different time can be grouped inside a similar view-site (note that: the scenes contain large variability in terms of people and cars).

We measure the performance of the scene localization by using known ground truth. The accuracy of the proposed scene localization method is shown in Table I.

From Table I, we see that there exists a small percentage of false positives. These are largely caused by light reflecting off one of the side mirrors causing the SIFT-Homography alignment process to fail.

### C. Comparison between our approach and other approach

We perform comparison on two different approaches for background subtraction: Stauffer-Grimson-based background subtraction [2] and the proposed KDE-based method. Figure 7 shows the results from the different approaches. In this figure, the background reconstruction models are shown in the first column. Next, we see that Stauffer’s approach (third column) produces more false positives and under segmentation as compared to our KDE background subtraction approach (fourth column). The advantage of our method is evident as it reduces false positives by incorporating the neighborhood probability. On the other hand, the Stauffer-Grimson approach which deals only with pixel wise background models is unable to handle the pixel misalignment issues and results in several false positives.

### D. System performance

We further evaluate our system by testing our proposed algorithm on the challenging datasets described in Section V-A. Figure 9 shows the snapshots of the pedestrian detection results. The first three rows show the detection results in region 1, where cluttered complex backgrounds (for example shops) and occluded crowded pedestrians exist. The last row of this image displays results of the detected pedestrians in region 2. Instances where our system returns false foreground detections are generally due to cluttered backgrounds or the misclassification of images into view-sites during the scene localization, resulting in comparison with the incorrect background model. These mis-detections are normally caused by the lack of foreground edges, under-segmentation and relatively small foreground regions. Figure 6 shows the ROC curves of our system performance on selected 1000 frames for each dataset. It can be seen that the performance plots for datasets in region 2 are better compared to those on region 1. This is due to the complexity of the background and the degree of occlusion. Overall, we achieve an average detection rate of between 71% to 83%, with false alarms of an average of 1.8 to 5.1 pedestrians/frame by varying different parameters. These figures verify that our proposed system provides a valuable framework for performing automatic visual surveillance at a large scale, taking into consideration the low-resolution and low-frame-rate video quality.

### VI. Conclusion and future work

In this paper, we have presented an integrated system for foreground-detection involving mobile surveillance cameras.
We have proposed a new approach to localize moving scenes captured from the bus camera into the appropriate view-sites (location) and further cluster them based on their global-background-intensity similarities. Over a period of time, the resulting view-sites construct a stable background model. A KDE background model is then employed for building the background model and is used to detect the objects of interest. Finally, hierarchical templates for pedestrian are generated to detect people. We have evaluated the performance of system using the real world datasets, on city streets, demonstrating the feasibility of our method. Currently, our system does not incorporate any motion detection cues. When this information is available, it should increase the system performance.

ACKNOWLEDGMENT

This work is based upon work funded in part by DTI and The ARC.

REFERENCES

Fig. 8. Scene localization results (hierarchical tree representation of scene localization process). First, images at Level-0 represent original video sequences captured from moving bus cameras. These images are then clustered into the “appropriate” view-sites as shown in Level-1 Clustering ($L_1$) based on their scene-transformation similarities. The bottom layer Level-2 Clustering ($L_2$) are obtained by sub-clustering $L_1$ based on their global background intensity values. Images at $L_2$ exhibit properties of both aligned in a static manner and having similar background intensities, which can be used to construct background models to detect foregrounds-of-interest.

Fig. 9. Pedestrian detection results on different datasets. The hierarchical human template matching algorithm [10] is applied on the KDE foreground-background separation results (shown in Figure 7) to confirm the detection of pedestrians.