2010 11th. Int. Conf. Control, Automation, Robotics and Vision
Singapore, 5-8th December 2010

A Unified 2D-3D Video Scene Change Detection Framework for Mobile Camera Platforms

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

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

***Abstract*—In this paper, we present a novel scene change detection algorithm for mobile camera platforms. Our approach integrates sparse 3D scene background modelling and dense 2D image background modelling into a unified framework. The 3D scene background modelling identifies inconsistent clusters over time in a set of 3D cloud points as the scene changes. The 2D image background modelling further confirms the scene changes by finding inconsistent appearances in a set of aligned images using the classical MRF background subtraction technique. We evaluate the performance of our proposed system on a number of challenging video datasets obtained from a camera placed on a moving vehicle and the experiments show that our proposed method outperforms previous works in scene change detection, which suggested the feasibility of our approach.**

***Index Terms*—Scene change detection, moving images, 3D and 2D background modelling**

I. Introduction

Automatic scene change detection in moving images is challenging for three reasons: (a) the footage is collected at different times (different lighting conditions), (b) the camera is mobile (different camera poses), and (c) the foreground covers various types of object classes (people, cars, buildings etc). These conditions cause state-of-the-art methods in scene change detection [1]–[7] to become impractical. The different lighting conditions caused by footage taken at different times violates the basic assumption that the background needs to have static lighting conditions, causing most background subtraction algorithms [7]–[9] to fail. As the vehicle/camera moves, the scenes over time are not located at the same position, which causes the traditional methods of accumulating the background information for modelling to become infeasible. Finally, direct foreground detection algorithms cannot be employed since the foregrounds in this problem belongs to diverse object classes (people, cars, etc).

In this paper, we present a novel scene change detection algorithm that integrates sparse 3D scene background modelling and dense 2D image background modelling into a unified framework. The process consists of two major steps: data alignment followed by a change detection process. The first step involves transforming the input video that are time-ordered image sequences into: (a) time-ordered 3D scene data, (b) motion-field data, and (c) 2D aligned image data (view sites), as shown in Figure 1 (a). In these representations, images and features from different video are geometrically aligned together and hence can be used for background modelling purposes. This step addresses the problem of moving cameras.

The second step involves detecting changes on this data. To tackle the different lighting conditions problem, our scene change detection is motivated by the following three observations:

1) *Spatial consistency*. Background features should be consistent over time. This is measured by the frequency of feature occurrence across all video. Intuitively, areas with random feature occurrence generally translate to foregrounds, like passengers and moving cars. On the other hand, buildings, vegetation, and sky are usually background areas, with high feature spatial consistency.
2) *Speed*. Non-stationary objects, for example moving vehicles or pedestrians, have a higher probability of being foreground as compared to stationary objects, for example buildings, vegetation, and sky.
3) *Appearance consistency*. Background areas should have consistent structural (gradient) appearance over time. Areas with high appearance consistency normally belong to background objects (e.g. houses, sky, etc), while low appearance consistency areas imply changes.

Since our proposed background modelling uses robust features and gradients which are partially invariant to lighting changes, it addresses the different lighting condition problem. The significance of our proposed system is that it provides a basic framework for many higher-level applications such as autonomous environment-change monitoring system.

This paper is organised as follows. In the following section, an overview of the proposed system is presented. In Sections III–VI, we describe our proposed scene detection approach in detail. Section VII presents the experimental results of our proposed method compared with existing techniques in background modelling. Section VIII concludes the paper.

II. Overview of The Algorithm

Our approach is illustrated in Figure 1. The 3D scene of each video is reconstructed using the Structure-From-Motion method [10]. This provides sparse three dimensional points of the scene relative to the camera motion. The Iterative Closest Point method [11] is then applied to align multiple 3D scenes, forming the time-ordered 3D scene data. We formulate 3D

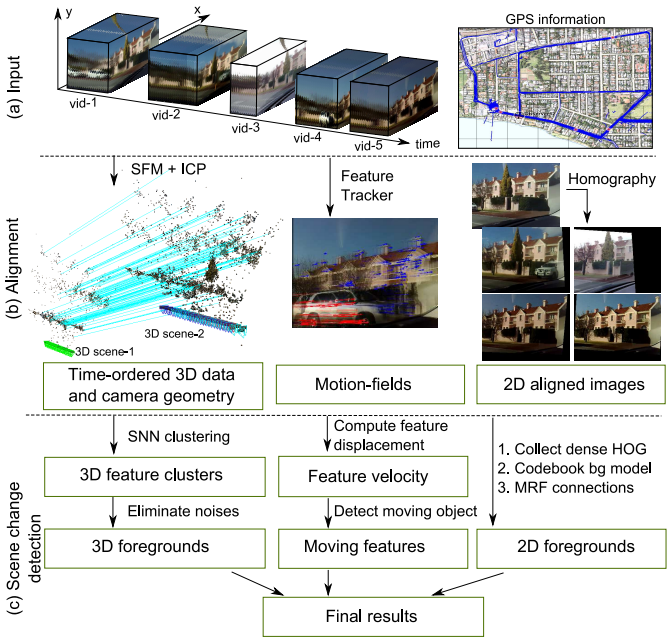

Fig. 1: Overview of the proposed algorithm.

scene change detection in a clustering framework [12], finding neighbourhood point correspondences or spatial consistency in the time-ordered 3D scene data. Intuitively, points that have high neighbourhood correspondence belong to background objects because it implies that the objects do not move, whilst points that appear inconsistently have a higher likelihood of being foreground objects.

Stable features are tracked over consecutively aligned images to form the motion-field data. Since moving objects have different velocities compared to static objects we are able to extract different class features according to their relative speed.

Using the recovered camera geometry, images from different video can be efficiently aligned to a fixed coordinate system using Homography method [13]. To model the background, we proposed the HOG [14] codebook modelling approach due to its robustness against illumination changes. In addition, we employ the Markov Random Field spatio-temporal framework [8] to obtain consistent scene changes in the video. Finally, we combine all outputs using morphological operations to obtain the final scene change detection results.

III. 3D Scene Change Detection

The 3D scene change detection involves constructing the time-ordered 3D scene data, determining the feature occurrence consistency and finding the final foreground clusters from the time-ordered 3D scene data.

A. Time-ordered 3D Scene Construction

The time-ordered 3D scene data is constructed from a set of video. The process involves iteratively aligning the 3D Structure-From-Motion (SFM) of a video to the existing time-ordered 3D scene data using Iterative Closest Point (ICP).

1) Structure From Motion (SFM) [10]: is a method to recover the camera geometry and three-dimensional structure of a scene by analysing motion of images over time. To do this, stable feature points, like SIFT [15], are extracted and tracked over a number of images and the three-dimensional structure information is obtained by minimising the sum of the squared projection error using bundle adjustment as:

$$\min_{T_j, X_i} \sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij}^2 \quad \text{with } e_{ij} = |\mathcal{P}(T_j, X_i) - x_{ij}| \qquad (1)$$

where X_i and x_{ij} denote the 3D and 2D coordinates of feature i in image j respectively. Function $\mathcal{P}$ is the projection of 3D object points to image coordinates and T_j denotes the transformation of the camera coordinate system of image j to an arbitrary world coordinate system. This error function can be minimised using non-linear least-square optimisation [16] or linear approaches based on projective geometry [17]. The results of SFM are sparse 3D cloud points relative to the estimated camera geometry (position and orientation).

2) Iterative Closest Point (ICP): Instead of recomputing the time-ordered 3D scene each time a new video is introduced, we employ the Iterative Closest Point approach [11] to merge the new SFM scene to the existing time-ordered 3D scene data. ICP [11] is a method to fit points in a target model to points in a source model. The goal of the algorithm is to find the 3D translation t vector and rotation R matrix that minimizes the sum of square errors with respect to the source points and their corresponding target points:

$$E(R, t) = \frac{1}{N} \sum_{i=1}^{N} ||X_i - RY_i - t||^2 \qquad (2)$$

where X_i and Y_i are corresponding points. The algorithm follows 4 major steps:

1) Correlate points using feature correspondence
2) Estimate the parameters using a mean square cost function as in Equation 2.
3) Transform the points using the estimated parameters.
4) Iterate until the error is below a certain threshold.

B. 3D Cloud Point Subtraction

Once we obtain the time-ordered 3D scene data, the task is to identify different spatial-occurrence patterns from a target scene to all other scenes. Intuitively, a feature is considered as "background" if it appears frequently across different video sequences, otherwise it is considered as "foreground". This problem can be viewed as a clustering problem of finding neighbourhood point correspondences across a set of 3D cloud points. In this work, we employ a similar idea as the shared nearest neighbour (SNN) clustering algorithm [12]. Given two points (a source and a target point) from two different scenes, we consider them as the same point if they satisfy the following three conditions: first, the corresponding target point has to be one of K-Nearest Neighbours from the nearest neighbours of the source point and vice versa. Secondly, their distance must

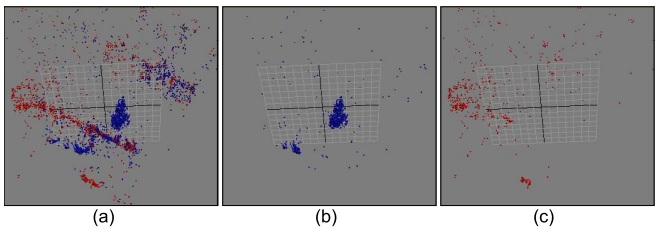

Fig. 2: 3D cloud point subtraction based on the spatial point similarity. (a) The blue and red scenes are created from two video data taken at different times. (b) The result of the 3D cloud point subtraction for the blue scene. (c) The result of the 3D cloud point subtraction for the red scene.

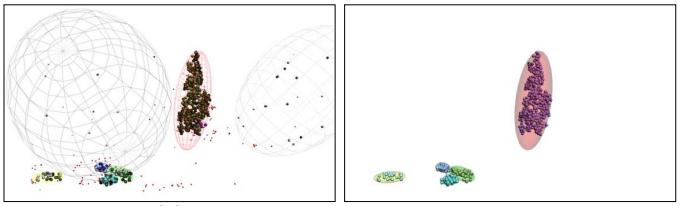

Fig. 3: An example of SNN clustering algorithm to detect foreground and eliminate noise. (a) A set of clusters are obtained after applying SNN clustering algorithm on the 3D cloud points subtraction results in Figure 2-(b). (b) The detected foreground clusters.

lie within a distance threshold Eps. Finally, both points have to share minimal of $MinPts$ to the similar nearest neighbours. The last condition enforces a tighter connection between the two points. Following this process, each 3D point now has information about its neighbours, in other words, its occurrence frequency. Let $L(X_i)$ be the frequency of a feature X_i over M number of video, where $L(X_i) \leq M$. By choosing a threshold th_L on $L(X_i)$, we are able to pick objects with different lifespan. If th_L is small, we are looking for short-duration foreground changes such as vehicles or pedestrians. As th_L increases, we obtain more background objects (high number of occurrences).

C. Spatial Clustering to Identify Foreground Objects and Eliminate Noise

Following the subtraction process, we obtain clusters that contain noise and foreground points (see Figure 2). To eliminate the noise, we apply the SNN clustering algorithm [12] on the remaining 3D points (after the subtraction process). Each point in a cluster is classified as one of the following classes: *core*, *border*, or *noise*. We define a point as

1) *Core* when its density is greater than Eps, where density of a point is computed by counting how many of its K nearest neighbours shares at least the same M number of nearest neighbours.
2) *Noise* for all non-core points which are outside the radius of Eps of a core point.
3) *Border* are the remaining points not classified as *core* or *noise*.

In other words, a *core* point should satisfy two conditions: (1) it must have a high number of points that share the same nearest neighbours and (2) between the point and its neighbours, they must also share a high number of similar neighbours. *Border* points are points that share just enough nearest neighbours with *core* points. *Noi*se points are all non *core* points where the number of shared nearest neighbours is less than the Eps (distance threshold). The results of this process are clusters with different shapes and densities. In this work, we assume that the sparse clusters are noise. To eliminate noise clusters, we compute the mean and standard deviations of each cluster. We then eliminate clusters that are three standard deviations from the mean. Figures 3 show an example of SNN clustering algorithm used to eliminate noises. In Figure 3 (a), we see that the SNN clustering algorithm is able to separate different density clusters based on their spatial characteristics, where the red points are noise and the rest are considered foreground clusters. The 3D scene change detection results are shown in Figure 3 (b).

The SNN clustering algorithm is suitable for our problem for two reasons. First, because our foreground objects include different types of objects such as pedestrians, people on bicycles, cars, trees, etc, all having different shapes and densities, and SNN being a density based algorithm can handle this problem. Secondly the SNN algorithm is more flexible since it only requires prior knowledge about the number of shared nearest neighbours, as compared to the K-means clustering algorithm [18] that requires the number of clusters to be known before hand or DBScan [19] that requires explicit knowledge about the distance threshold.

IV. Stationary/Non-stationary Object Probability

In video sequence, an object can be categorised as either stationary or non-stationary based on its velocity. While stationary objects can either be background or foreground, non-stationary objects generally belong to the foreground, e.g. moving objects like walking pedestrians and moving vehicles always belong to foreground. Based on this, we assume that the non-stationary features belong to foreground objects.

To estimate the velocity of an object, we need to track and compute the feature displacement over the consecutive images. In a static camera, the feature displacement is simply the spatial pixel differences of the same feature over two consecutive frames using optical flow or SIFT flow [20] feature tracking. The problem becomes more complicated when the camera itself is moving, as in our case. We also need to estimate the camera displacement over consecutive sequences. Intuitively, since we have recovered the camera geometry from SFM, we can compute the relative distance from one camera to the other, which is equivalent to stitching the consecutive image sequences using the view frustum concept [21].

In computer graphics, a view frustum [21] is the field-of-view of a camera that appears on the screen. Given the camera trajectory, our proposed solution is to project all images to the *far plane* of the camera frustum. At the end of the projection process we then flatten these far planes into a single plane. In

Fig. 4: Example of two scenes with objects of different velocity. (Left) A moving car has longer motion fields compared to other objects in the scene such as houses and trees. (Right) Static objects normally have similar average velocity.

this plane, all consecutive images are roughly stitched together. The average velocity of a feature can then be computed as the relative distance between the corresponding feature displacement in the motion-field image. Note that although the images do not stitch perfectly this strategy allows us to estimate the relative feature velocity, which is enough for our problem. Figure 4 shows motion-fields of moving and stationary objects in moving video sequences.

We then combine the motion-field and the SNN clustering results using morphological operation of 'OR' to form the final 3D scene change detection results. The 3D result is generally too sparse to provide full segmentation of the objects. Next, we present a Markov Random Field (MRF) framework based on robust feature background modelling to extract scene changes inside the 2D aligned image clusters.

V. 2D Change Detection

A. 2D Data Alignment

Similar to traditional methods for foreground-background modelling, it is required for images to be aligned before the background model can be created. In this work, since we have recovered the camera location and orientation from SFM, the best Homography matrix (image alignment) for each image can be searched efficiently in the order of its nearest neighbour cameras. This is motivated by the fact that the best alignment target image has no significant transformation changes (in terms of skew, rotation, scale, and translation), and thus cameras that are closest to the source camera will have a higher probability of being aligned compared to those that are far from the camera. Once the ordering is determined we employ the Homography method as in [13] to align images from different video.

B. Spatio-Temporal MRFs for Scene Change Detection

Since our problem of detecting scene changes is an image-labelling problem, it can be further viewed as a problem of inferring the Maximum A Posterior (MAP) solution of a MRF model. We employ a multi-resolution MRF background modelling approach similar to that in [8] to model the spatio-temporal connections between images in the video. The likelihood of the problem can be written as:

$$p(Z_i|X_i,\theta) = p(Z_i|X_i,\theta_i) \tag{3}$$

where X_i, Z_i , and θ_i are the state, observation, and background models for pixel i respectively.

The background model θ is one of the most important elements that significantly influence the segmentation results. Therefore, it should reflect the characteristics of the dataset. Due to different lighting conditions in most of the dataset, we employ a HOG codebook based background modelling similar to [22]. We present each pixel of the background using the Histogram of Oriented Gradient [14], built from a window region centered around the pixel. The codebook for each pixel is constructed by grouping similar background HOG features to form a set of codewords. The weight of each codeword is defined by the number of the feature occurrences. The advantage of using the codebook based approach is because it preserves the original structure of the histograms (a distinctive property of the feature), creating more robust background models when compared to treating each channel independently.

Given a target image, the observation likelihood of each pixel is computed using,

$$p(Z_i|\theta_i) = p(Z_i) = CB(Z_i,\theta_i) \tag{4}$$

where θ_i be the background model for each pixel and $CB(.)$ is a function that returns the probability of the current feature matching the best background codeword. The comparison between HOG features and the codebook background features are performed using the Bhattacharyya coefficient. Since there is no guarantee that all the images from different times will align perfectly, we search for maximum responses from around the neighbourhood and redefine the background probability of a pixel as:

$$P_{\mathcal{N}}(Z_i|\theta_i) = \max_{j\in\mathcal{N}(i)} P(Z_i|\theta_j) \tag{5}$$

where $\mathcal{N}$ is the neighbour pixel of i. Intuitively, this model searches for the closest candidate pixel around a set of neighbours, since we assume that even if misalignment exists, it lies around the neighbours. Although this approach addresses the misalignment problem, it introduces the problem of losing true detections, as mentioned by [23]. Hence we incorporate a further constraint on the neighbourhood conditions, i.e. if a pixel is truly a background pixel that happens to move to a new location, then some of its neighbourhood pixels should also move, since we assume that the misalignment happens not for one single pixel, but for a group of pixels. Therefore, we define the probability of displacement as:

$$P_{\mathcal{C}}(Z_i) = \prod_{j\in\{i\cup\mathcal{N}(i)\}} P_{\mathcal{N}}(Z_j) \tag{6}$$

where $P_{\mathcal{C}}(Z_i)$ is computed as the product over the connected components of the neighbourhood pixels. By substituting Equation 6 into the MRF framework proposed by [8], the conditional posterior probability of the problem can be computed.

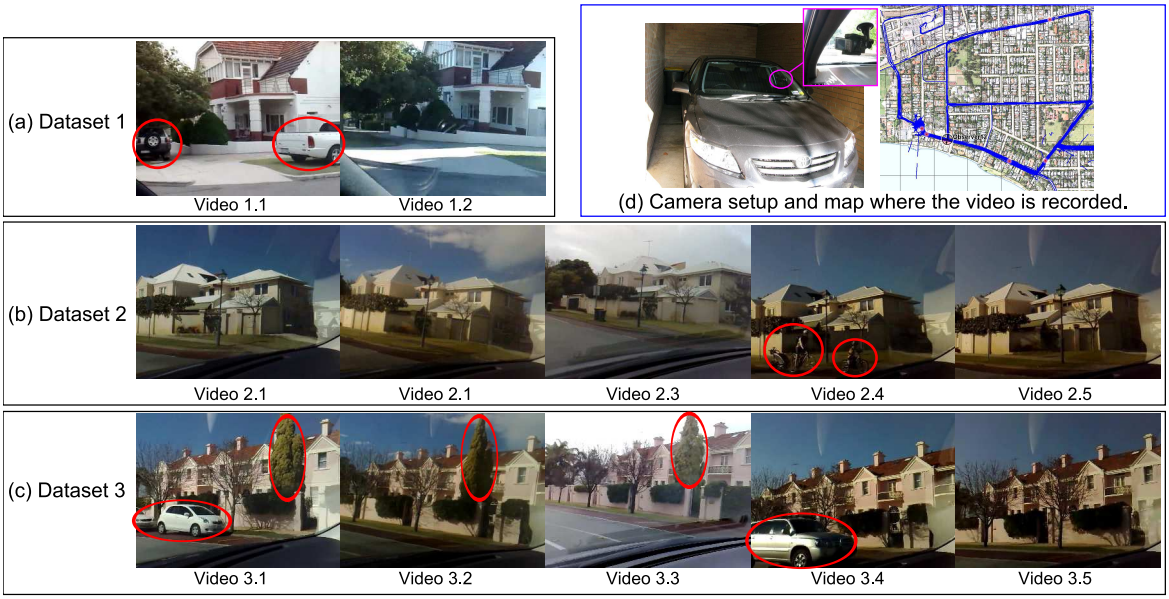

Fig. 5: The experimental datasets. Note that: for explanation purposes, the red ovals indicate the scene changes (ground truth) in each dataset. However, all experiments were performed on the original images (without the ground truth information).

VI. Combining Results from 3D and 2D Scene Change Detection

The final scene change detection results are obtained by merging the 3D with 2D results based on their connected components. We name the combined approach as *SFM-HOG* scene detection. To merge the two results, we first project the 3D scene change detection results back to the 2D images using the view frustum method. Next, we compute the connected components for both 3D and 2D results to identify blob clusters. Finally, we find the overlap connected component regions between the 3D and 2D results. If the number of overlap pixels between the two connected components is larger than a certain threshold, we assign it as a foreground object, otherwise it is assigned as background noise. In other words, this strategy chooses the final foreground results that agree with both the 3D and 2D results.

VII. Experimental Results

A. Scene Changes Dataset

The experimental datasets were collected using a single front-facing camera (Nokia N95[1]) placed on the dash board of a moving car. Figure 5 (d) shows the physical camera setup and the GPS locations of the scene where the video was taken. The camera was set up to record a video stream of size 640x480 at 30 frames per second (fps) and was located on the left side of the car looking to the right. During data collection, the vehicle travelled at 40–50 km/hour.

The experimental data consists of three different scene datasets. Table I summarises the descriptions of the video whilst Figure 5 shows snapshots of the different datasets. Each row represents the dataset and each column represents the different time the video was taken.

Experiment I: The first dataset consists of only two video sequences: video 1.1 and video 1.2. There are two *scene changes* between these two video. First, there is a white jeep parked besides the tree in video 1.1 but not in video 1.2. Similarly, there is a white pickup truck in video 1.1 that does not appear in video 1.2, as shown in Figure 5 (a). The *background objects* (*scenes that remain unchanged*) in this dataset include houses, vegetation, and shadows.

Experiment II: Five video are used in the second experiment as shown in Figure 5 (b). In this experiment, video 2.1, 2.2, 2.3, and 2.5 do not contain any changes, whilst there are *three scene changes* in video 2.4. The first two changes are a man and a child riding bicycles (between frames 86–92). Between frames 108 to 112 of video 2.4, there is a sedan driving on the opposite side of the road.

Experiment III: Figure 5 (c) shows five video that are used in the third experiment. In this experiment, the tree in front of the house was cut off after video 3.3; therefore, it does not appear in video 3.4 and 3.5. Other changes include (1) two white sedans appearing in video 3.1 but not in other video; (2) a jeep travels in the opposite direction of the traffic in video 3.4, as shown in Figure 5 (c).

B. Evaluation Criteria

To evaluate the performance of our proposed framework, we present a quantitative measurement in terms of the bounding box which is computed using the $TPR(\%) = \frac{TP}{TP+FN}$ and $FPR(\%) = \frac{FP}{FP+TN}$, where TP, TN, FP, FN are the true positives, true negatives, false positives, and false negatives respectively. We compute the *ratio of the overlap* to determine the class results, which is expressed as:

[1] http://www.nokia.co.uk/find-products/all-phones/nokia-n95/specifications

Dataset	Number of video	Number of frames	Video ID	Frame index
1	2	39	1.1	1–20
			1.2	21–39
2	5	176	2.1	1–36
			2.2	37–54
			2.3	55–85
			2.4	86–112
			2.5	113–176
3	5	136	3.1	1–28
			3.2	29–51
			3.3	52–73
			3.4	74–100
			3.5	101–136

TABLE I: Data description.

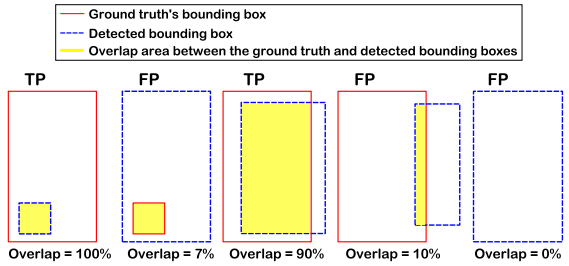

Fig. 6: Evaluation Criteria.

$$overlap = \frac{\text{area(detected} \cap \text{groundtruth)}}{\text{area(detected)}} \quad (7)$$

A detected candidate is considered to be a true positive when the *overlap ratio* between the detected and the ground truth bounding boxes is above 40%, otherwise it is considered as a false positive, as shown in Figure 6.

C. Comparison between our approach and other approaches

We perform a comparison study on three different approaches: 2D background modelling alone [8]; 3D scene change detection using SFM features; and the combined approach (SFM-HOG). Table II, Figures 7, 8, and 9 show the overall performance and snapshots of the detection results for experiment I, II, and III respectively.

Experiment I: Figure 7 shows some comparative results between 3D, 2D, and SFM-HOG approaches of dataset 1, where each column represents results from different approaches. The green bounding boxes and the red coloured pixels indicate the detected foregrounds. From Figure 7, we can see that the high number of false detections for using either 3D or 2D result alone is greatly reduced when the two approaches are combined. The false detections in Figure 7 (c) are mainly caused by the limited number of video used in this experiment and there is large difference in lighting between these video. The shadow in video 1.2 does not present as part of the 3D foreground because the 3D feature density in that area is small and thus is discarded during the thresholding process. However the 2D approach is not able to handle the shadow noise which results in false detections, as shown in Figure 7 (b). In this experiment we achieve 66.62% true positives detection (TPR) with 18.69% false positives (FPR) for the SFM-HOG approach, 35.71/64.29% (TPR/FPR) for 3D, and 42.52/77.35% (TPR/FPR) for 2D approach. This demonstrates the effectiveness of combining both the 2D and 3D scene detection results.

Bounding box		*Dataset 1*	Dataset 2	Dataset 3
TPR	3D	35.71%	57.72%	53.29%
	2D	42.52%	67.11%	56.28%
	SFM-HOG	66.62%	79.29%	77.06%
FPR	3D	64.29%	10.15%	24.62%
	2D	77.35%	82.75%	73.46%
	SFM-HOG	18.69%	11.23%	10.94%

TABLE II: Evaluation results.

Experiment II: Figure 8 shows some comparative results between 2D, 3D, and SFM-HOG approaches. Although the lighting differs between all video, they still have similar gradient structures, hence we obtain good image segmentation results as shown in Figure 8 (c). In this experiment, the SFM-HOG algorithm is able to correctly identify the foreground objects that belong to a man, a child, and a sedan respectively. In this experiment, we achieve 79.29/11.23% (TPR/FPR) for SFM-HOG approach, 57.72/10.15% (TPR/FPR) for 3D approach, and 67.11/82.75% (TPR/FPR) for 2D approach. In general, the SFM-HOG method is able to obtain the bounding boxes of the objects correctly, with a small number of false positives.

Experiment III: From Figures 9 (c), we can see that the SFM-HOG is able to detect the scene changes correctly, which consist of the tree, the sedans, and the jeep. From the 3D results (see Figure 9 (a)) we can see that it contains less foreground information as compared to the 2D results with a higher number of false positives, as shown in Figure 9 (b). By combining the results we show that the SFM-HOG is able to extract the changes correctly without large numbers of false positives, as shown in Figure 9. In this experiment, we achieve 77.06/10.94% (TPR/FPR) for the SFM-HOG approach, 53.29/24.62% (TPR/FPR) for 3D approach, and 56.28/73.46% (TPR/FPR) for 2D approach.

Overall discussion: The results from Table II show that by using the 2D scene change detection approach alone generally results in higher false positives as compared to SFM-HOG, while the 3D scene change detection approach alone gives lower true positives as compared to SFM-HOG. On the other hand, when they are combined together, the best performance is achieved. The intuitive reason for this is because the noisy results of 3D and 2D do not overlap with each other due to the different nature of the two techniques. The 2D results fill the sparse 3D results to give more information about the foreground objects. At the same time, since the 3D results do not contain noise, it reduces the false detections in the 2D results. Based on the results at Table II, we can also see that using less numbers of video such as experiment 1 result in low performance. As the dataset size increases, the noise of the dataset can be reduced by comparing multiple images and thus we obtain better overall performance.

VIII. Conclusion

In this paper we present a system for scene change detection. Our proposed framework is able to detect scene changes correctly despite the challenges in the datasets (variability in camera pose and lighting conditions). The main contribution lies in the use of 3D scene information to provide stable scene changes and 2D segmentation algorithm to obtain more detailed information about the object.

Acknowledgment

This work is based upon work funded in part by DTI and the Australian Research Council.

References

[1] C. Y. Fang, S. W. Chen, , and C. S. Fuh, "Automatic change detection of driving environments in a vision-based driver assistance system," *IEEE Trans. Neural Networks*, vol. 14, pp. 646–657, 2003.

[2] S. Berrabah, G. de Cubber, V. Enescu, and H. Sahli, "MRF-Based Foreground Detection in Image Sequences from a Moving Camera," in *ICIP06*, 2006, pp. 1125–1128.

[3] J. Landabaso and M. Pardas, "A Unified Framework for Consistent 2-D/3-D Foreground Object Detection," *CirSysVideo*, vol. 18, no. 8, pp. 1040–1051, August 2008.

[4] G. Qian and R. Chellappa, "Moving targets detection using sequential importance sampling," in *ICASSP*, 2001.

[5] M. Irani and P. Anandan, "A Unified Approach to Moving Object Detection in 2D and 3D Scenes," *PAMI*, vol. 20, no. 6, 1998.

[6] M. Lourakis, A. Argyros, and S. Orphanoudakis, "Independent 3D Motion Detection Using Residual Parallax Normal Flow Fields," in *ICCV*, 1998, pp. 1012–1017.

[7] E. Durucan and T. Ebrahimi, "Change detection and background extraction by linear algebra," *PIEEE*, vol. 89, no. 10, pp. 1368–1381, 2001.

[8] W. Xu, Y. Zhou, Y. Gong, and H. Tao, "Background Modeling Using Time Dependent Markov Random Field With Image Pyramid," in *MOTION*, 2005.

[9] C. Stauffer and W. E. L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *PAMI*, vol. 8, no. 22, pp. 747–757, 2000.

[10] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," *IJCV*, vol. 80, no. 2, pp. 189–210, 2008.

[11] P. J. Besl and N. D. Mckay, "A Method for Registration of 3-D Shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, February 1992.

[12] L. Ertöz, M. Steinbach, and V. Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data," in *SIAM*, San Francisco, CA, USA, 2003.

[13] W. Leoputra, S. Venkatesh, and T. Tan, "Pedestrian detection for mobile bus surveillance," in *ICARCV*, vol. 2, 2008.

[14] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, vol. 2, June 2005, pp. 886–893.

[15] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[16] B. K. Horn, *Robot Vision*. McGraw-Hill Higher Education, March 1986.

[17] L. Quan and Z. Lan, "Linear N-Point Camera Pose Determination," *PAMI*, vol. 21, no. 8, pp. 774–780, 1999.

[18] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[19] M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *KDD*, 1996, pp. 226–231.

[20] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "SIFT Flow: Dense Correspondence across Different Scenes," in *ECCV*, 2008, pp. 28–42.

[21] U. Assarsson and T. Möller, "Optimized View Frustum Culling Algorithms for Bounding Boxes," *Journal of Graphics Tools*, vol. 5, pp. 9–22, 2000.

[22] W. Leoputra, S. Venkatesh, and T. Tan, "Passenger monitoring in moving bus video," in *ICARCV*, vol. 2, December 2008, pp. 741–744.

[23] A. M. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric Model for Background Subtraction," in *ECCV*, 2000, pp. 751–767.

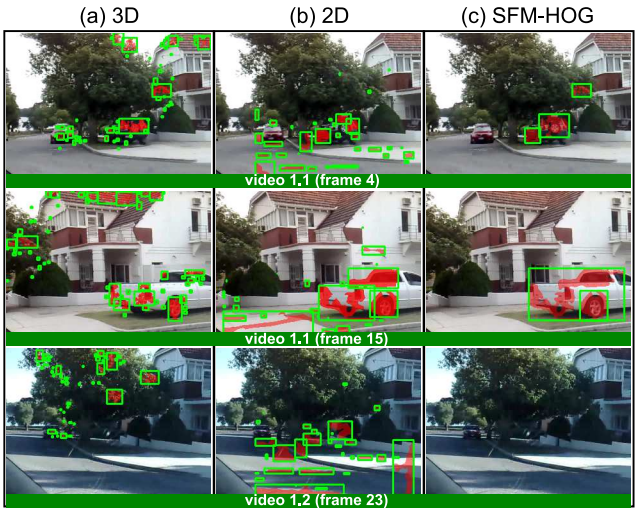

Fig. 7: The scene change detection results for dataset 1.

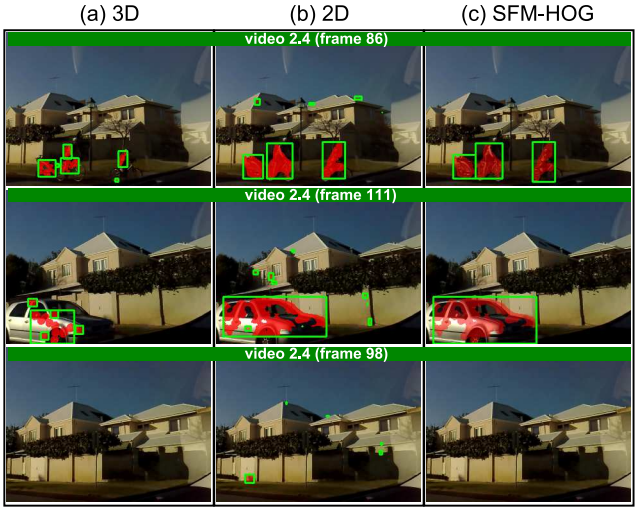

Fig. 8: The scene change detection results for dataset 2.

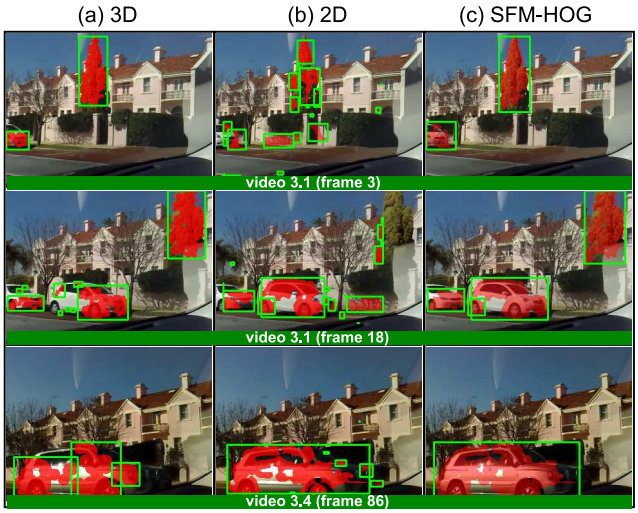

Fig. 9: The scene change detection results for dataset 3.