Using 3D Line Segments for Robust and Efficient Change Detection from Multiple Noisy Images

Ibrahim Eden and David B. Cooper

Division of Engineering
Brown University
Providence, RI, USA
{ieden,cooper}@lems.brown.edu

Abstract. In this paper, we propose a new approach to change detection that is based on the appearance or disappearance of 3D lines, which may be short, as seen in a new image. These 3D lines are estimated automatically and quickly from a set of previously-taken learning-images from arbitrary view points and under arbitrary lighting conditions. 3D change detection traditionally involves unsupervised estimation of scene geometry and the associated BRDF at each observable voxel in the scene, and the comparison of a new image with its prediction. If a significant number of pixels differ in the two aligned images, a change in the 3D scene is assumed to have occurred. The importance of our approach is that by comparing images of lines rather than of gray levels, we avoid the computationally intensive, and some-times impossible, tasks of estimating 3D surfaces and their associated BRDFs in the model-building stage. We estimate 3D lines instead where the lines are due to 3D ridges or BRDF ridges which are computationally much less costly and are more reliably detected. Our method is widely applicable as man-made structures consisting of 3D line segments are the main focus of most applications. The contributions of this paper are: change detection based on appropriate interpretation of line appearance and disappearance in a new image; unsupervised estimation of "short" 3D lines from multiple images such that the required computation is manageable and the estimation accuracy is high.

1 Introduction

The change detection problem consists of building an appearance model of a 3D scene using n images, and then based on an $n+1^{st}$ image, determining whether a "significant" change has taken place. A fundamental approach to this problem is to estimate a 3D model for the scene and the associated BRDF; then based on the knowledge of the $n+1^{st}$ image viewing position and scene illumination, a decision is made as to whether there is a significant difference between the $n+1^{st}$ image and its prediction by the n-image based 3D geometry and BRDF (bidirectional reflectance distribution function) estimates.

In its general form, all learning is done in the unsupervised mode, and the n-image based learning is not done for a static 3D scene but rather for a functioning scene where changes are often taking place. A complicating factor in the change detection problem

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 172–185, 2008.

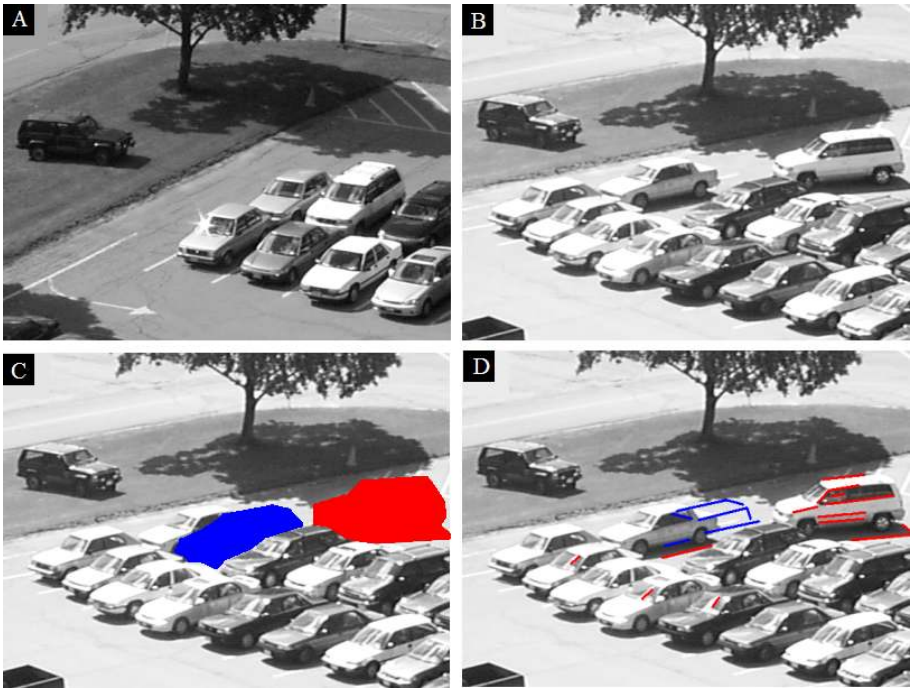

Fig. 1. Our line segment based change detection result after training on a sequence of 5 images. (A) A sample training image. (B) The test image. (C) Hand-marked ground truth for change where the new object is shown in "red" and the disappeared object is shown in "blue". (D) Result of our method. Lines associated with the new object are shown in "red" and lines associated with the disappeared object are shown in "blue". Two major change regions are detected with only a few false alarms due to specular highlights and object shadows. (This is a color image)

is that images can be taken at arbitrary time, under arbitrary lighting conditions and from arbitrary view points. Furthermore, they are usually single images and not video. For example, if they are taken from a flying aircraft, a 3D point in the scene is usually seen in one image and not in the immediately preceding or succeeding images, and is not seen again until the aircraft returns at some later time or until some other aircraft or satellite or moving camera on the ground sees the point at some later time.

In this paper, we assume n images are taken of a scene, and we then look for a change in the $n + 1^{st}$ image, and if one has occurred we try to explain its type (resulting from the arrival or from the departure of a 3D object). The learning is done in an unsupervised mode. We do not restrict ourselves to the case of buildings where the 3D lines are long, easy to detect, easy to estimate and are modest in number. Rather, we are interested in the case of many short lines where the lines can be portions of long curves or can be short straight line segments associated with complicated 3D objects, e.g., vehicles, scenes of damaged urban-scapes, natural structure, people, etc...

Why do we restrict this study to straight lines? We could deal with curves, but since curves can be decomposed into straight lines, and since straight lines – especially short

line segments - appear extensively in 3D scenes and in images, we decided to start with those. The important thing is that estimating 3D structure and the associated BRDF can often be done in theory, but this is usually difficult to do computationally. On the other hand, estimating 3D line segments is much more tractable and can be considered as a system in its own right or as contributing to applications that require efficient 3D structure estimation.

Our paper consists of the following. Given n images, we estimate all 3D lines that appear in three or more images. Our approach to 3D line estimation emphasizes computational speed and accuracy. For very short lines, accuracy is greatly improved by making use of incidence relations among the lines. For change detection we look for the appearance or disappearance of one or more line segments in the $n + 1^{st}$ image. This procedure depends on the camera position of the new image and the set of reconstructed 3D line segments in the learning period, and therefore an interpretation of whether a line is not seen because of self occlusion within the scene or because of a 3D change. Usually, but not always, if an existing 3D line should be visible in the $n + 1^{st}$ image and is not, the reason is because of occlusion by the arrival of a new object or departure of an existing object. If a new object arrives, there will usually be new lines that appear because of it, but it is possible that no new straight lines appear. Hence, detecting and interpreting change, if it occurs, based on straight line segments is not clear cut, and we deal with that problem in this paper.

2 Related Work

Some of the earlier work on change detection focuses on image sequences taken from stationary cameras. The main drawback of these methods is their likelihood to create false alarms in cases where pixel values are affected by viewpoint, illumination, seasonal and atmospheric changes. This is the reason why pixel (intensity) and block (histogram) based change detection algorithms such as image differencing [1,2] and background modeling methods [3] fail in some applications.

Meanwhile, there exist change detection methods designed for non-stationary image sequences. There has been a lot of work in the literature on methods based on detecting moving objects [4,5], but these methods assume one or more moving objects in a continuous video sequence. On the other hand, 3D voxel based methods [6] where distributions of surface occupancy and associated BRDF are stored in each voxel can manage complex and changing surfaces, but these methods suffer from sudden illumination changes, perform poorly around specular highlights and object boundaries.

To our knowledge, line segment based change detection methods have rarely been studied in computer vision literature. Rowe and Grewe [7] make use of 2D line segments in their algorithm, but their method is specifically designed for aerial images where the images can be registered using an affine transformation. Li et al. [8] provided a method of detecting urban changes from a pair of satellite images by identifying changed line segments over time. Their method does not estimate the 3D geometry associated with the line segments and takes a pair of satellite (aerial) images as input where line matching can be done by estimating the homography between the two images. The change detection method we propose in this work is more generic, it can

work on non-sequential image sequences where the viewpoint can change drastically between pairs of images and it is not based on any prior assumptions on the set of training images.

3 Multi-view Line Segment Matching and Reconstruction

Line segment matching over multiple images is known to be a difficult problem due to its exponential complexity requirement and challenging inputs. As a result of imperfections in edge detection and line fitting algorithms, lines are fragmented into small segments that diverge from the original line segments. When unreliable endpoints and topological relationships are given as inputs, exponential complexity search algorithms may fail to produce exact segment matching.

In this section, we present a generic, reliable and efficient method for multi-view line matching and reconstruction. Although our method is also suitable for small baseline problems (e.g. aerial images, continuous video sequences), such cases are not our primary focus as their line ordering along the epipolar direction does not change much and they can be solved efficiently by using planar homographies. In this paper, we focus on large baseline matching and reconstruction problems, where sudden illumination changes and specular highlights make it more difficult obtain consistent line segments in images of the same scene. These problems are more challenging as the line ordering in different images change due to differences in viewing angles. The following subsections describe three steps of our 3D line segment reconstruction method: an efficient line segment matching algorithm, reconstruction of single 3D lines segments and reconstruction of free form wire-frame structures.

3.1 An Efficient Line Segment Matching Algorithm

In general, the line matching problem is known to be exponential in the number of images. That is to say, given there are n images of the same scene and approximately m lines in each image, the total complexity of the line matching problem (the size of the search space) is $O(m^n)$. One way to reduce the combinatorial expansion of the matching problem is to use the epipolar beam [9,10]. Given the line $l = (x_1, x_2)$ in I, the corresponding line in I' should lie between $l'_1 = Fx_1$ and $l'_2 = Fx_2$ where F is the fundamental matrix between I and I' (see figure 2). While the epipolar beam reduces the combinatorial expansion of the matching algorithm, this reduction highly depends on the general alignment of line segments relative to epipolar lines. Plane sweep methods are also used to avoid the combinatorial expansion of the matching problem [11], but these methods do not perform well when the endpoints of 2D line segments are not consistent in different images of the same scene. Another way to increase the matching efficiency is to use color histogram based feature descriptors for 2D line segments [12], but these methods assume that colors only undergo slight changes and the data does not contain specular highlights. Our work focuses on more challenging real world problems where the above assumptions do not hold.

In this paper, we propose a new method that improves the multi-view line matching efficiency. Our method is based on the assumption that the 3D region of interest (ROI)

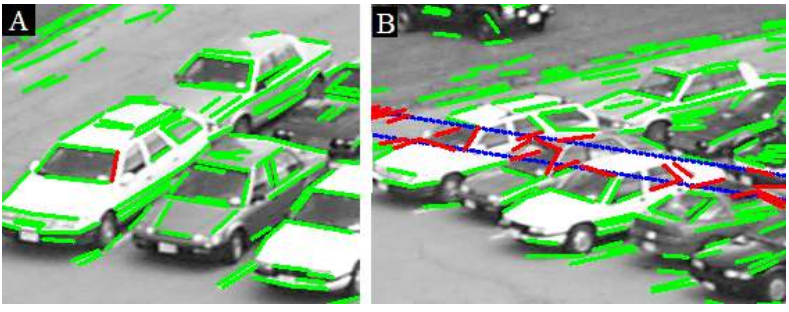

Fig. 2. An example of epipolar beam. (A) I_1: "selected line" for matching is shown in "red". (B) I_2: the epipolar beam associated with the selected line in I_1 is marked by "blue" and line segments that lie inside the epipolar beam (i.e., candidates for matching) are shown in "red". The epipolar beam in image (B) reduces the search space by 5.55. (This is a color image)

is approximately known, however this assumption is not a limitation for most multi-view applications, since the 3D ROI can be obtained by intersecting the viewing cones of the input images. The basic idea of our approach is to divide the 3D ROI into smaller cubes, and solve the matching problem for the line segments that lie inside each cube. The matching algorithm iteratively projects each cube into the set of training images, and extracts the set of 2D line segments in each image that lie (completely or partially) inside the convex polygon associated with the 3D cube.

Assuming that lines are distributed on the cubes homogeneously, the estimated number of lines inside each cube is $(\frac{m}{C})$, where C is the total number of cubes, and the algorithmic complexity of the introduced matching problem is $O(\frac{m^n}{C^n})$. It must be noted that under the assumption of homogenous distribution of lines segments in the 3D ROI the total matching complexity is reduced by a factor of $\frac{1}{C^n}$, where the efficiency gain is exponential. On the other hand, even in the existence of some dispersion over multiple cubes, our proposed algorithm substantially reduces the computational complexity of the matching algorithm. Figure 3 illustrates the quantitative comparison of different matching algorithms for 4 images of the same scene. The matching method we use in this work is a mixture of the algorithm described above and the epipolar beam method (EB+Cubes).

3.2 3D Line Segment Reconstruction

3D line segments are reconstructed by using sets of corresponding 2D line segments from different images obtained during the matching step. Each 3D line segment, $L = (X_1, X_2)$ is represented by 6 parameters in 3D space, where X_1 and X_2 are 3D points representing the end points. Here we assume that for each 3D line segment, corresponding set of 2D line segments are available in different images. We use the Nelder-Mead (Simplex) Method [13] to solve the minimization problem given in equation 1.

$$L^* = \underset{L \in \{R^3, R^3\}}{\arg\min} \sum_{i=1}^{n} d_l(l_i, l_i') + \beta \sum_{i=1}^{n} d_s(l_i, l_i'') \tag{1}$$

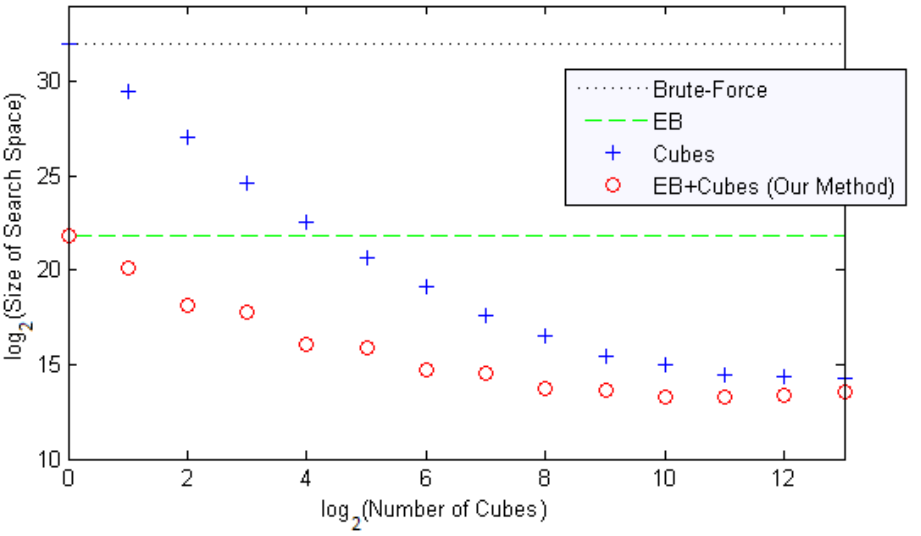

Fig. 3. Quantitative comparison of four different line segment matching algorithms using four images of the same scene. Brute-force: the simplest matching method, all combinations are checked over all images. EB: epipolar beam is used to reduce the search space. Cubes: the 3D space is splitted into smaller sections and matching is done for each section separately. EB+Cubes: (the method we use in this work) a combination of "Cubes" and "EB". It is shown that "Cubes+EB" method outperforms individual "EB" and "Cubes" methods and Brute-Force search. Notice that the size of the search space is given in logarithmic scale.

where $l'_i = (M_i X_1) \times (M_i X_2)$ is the projection of L to the i^{th} image as an infinite line, $l''_i = (M_i X_1, M_i X_2)$ is the projection as a line segment, M_i is the projection matrix for the i^{th} image, d_l is the distance metric between a line and a line segment and d_s is the distance metric between two line segments. The distance metrics d_l and d_s are defined as

$$d_l(l, l') = \sqrt{\frac{1}{|l|} \sum_{p \in l} d_p^2(p, l')}$$

$$d_s(l, l'') = \sqrt{\frac{1}{|l|} \sum_{p \in l} d_{ps}^2(p, l'')} + \sqrt{\frac{1}{|l''|} \sum_{p'' \in l''} d_{ps}^2(p'', l)}$$

where $d_p(p, l)$ is the perpendicular distance of a point (p) to an infinite 2D line and $d_{ps}(p, l)$ is the distance of a point to a line segment.

Note that, β in equation 1, is used to control the convergence of the local search algorithm. β is typically selected to be a number close to zero ($0 < \beta << 1$), so the first part of the objective function dominates the local search until the algorithm converges to the correct infinite line. Later, the second part (succeeded by β) of the objective function starts to dominate the local search algorithm in order to find the optimal end points for the 3D line segment.

3.3 Reconstruction of Wire-Frame Models

Next, we look into improving our line segment reconstruction results by exploiting the pairwise constraints between 3D line segments. Even though the single line segment reconstruction method explained in section 3.2 mostly provides good results, there exists degenerate cases that single line segment reconstruction fails to generate accurate estimations. This usually occurs when a 3D line segment and camera centers lie on the same plane. In this case, small errors in image space will lead to bigger errors in 3D space. One way to overcome this problem is to make use of incidence relations between two line segments (L and Y junctions) to improve the estimation of the 3D structure [14]. In the computer vision literature, models for special structures such as building roofs have been developed [15,16] using aerial images with relatively small baselines between pairs of images. These methods are not designed to work well for reconstructing free form wire-frame models for arbitrary 3D objects where the viewing angle between different images change drastically.

We use a pairwise checking algorithm between the reconstructed 3D line segments for forming wire-frame model without prior information on the structure of the 3D object. During the formation of the wire-frame model, we look at the closeness of the endpoints of reconstructed 3D line segments as well as the closeness of the endpoints of 2D line segments in images that are associated with the reconstructed 3D line segment. Two entities are joined if their endpoints are close enough both in the image space and the 3D space. The wireframe model keeps growing unless there are no line segments left that satisfy the closeness constraint.

Each wireframe model is formed as an undirected graph $G = (V, E)$ where the set of edges represent 3D line segments and vertices represent their endpoints. Instead of minimizing the objective function for each line segment separately, here we minimize an objective function for all lines (edges) in the wire-frame (graph) model. The criteria for wire-frame (graph) minimization problem is given in equation 2.

$$G^* = \underset{V \in R^{3N_V}}{\arg\min} \sum_{e \in E} \Big(\sum_{i=1}^{n} d_l(l_{i_e}, l'_{i_e}) + \beta \sum_{i=1}^{n} d_s(l_{i_e}, l''_{i_e}) \Big) \tag{2}$$

where N_V is the number of vertices in a wire-frame model and l_{i_e} is the line in i^{th} image associated with edge $e \in E$ in graph $G = (V, E)$.

Note that, the wireframe model introduces more constraints, therefore the sum of squares error of the individual line segment reconstruction is always smaller then the error of the wireframe model reconstruction. On the other hand, additional constraints provide better estimation of 3D line segments in terms of the actual 3D geometry. The result of 3D line segment reconstruction using wire-frame models is given in figure 4.

4 Change Detection

Our change detection method is based on the appearance and disappearance of line segments throughout an image sequence. We compare the geometry of lines rather than the gray levels of image pixels to avoid the computationally intensive, and some-times impossible, tasks of estimating 3D surfaces and their associated BRDFs in the model-building stage. Estimating 3D lines is computationally much less costly and is more

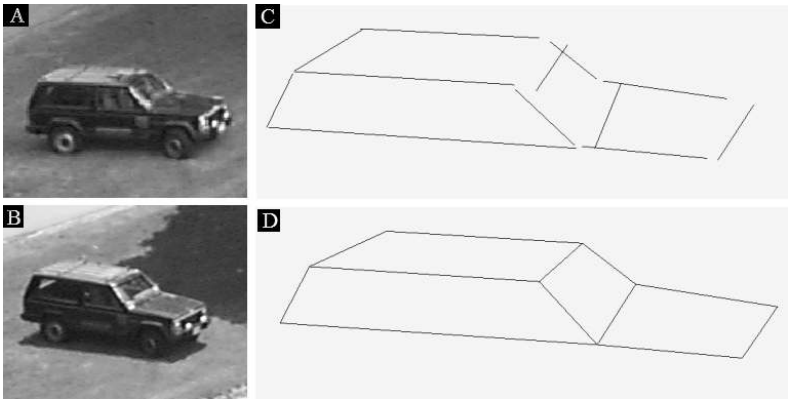

Fig. 4. Reconstruction results for 3D line segments of a Jeep. (A)-(B) The Jeep is shown from two different views. (C) Reconstruction results of single line segment reconstruction as explained in section 3.2. (D) Reconstruction results of the wire-frame model reconstruction as proposed in section 3.3.

reliable. Our method is widely applicable as man-made structures consisting of 3D line segments are the main focus of most applications.

4.1 Definition of the Problem

The definition of the general change detection problem is the following: A 3D surface model and BRDF are estimated for a region from a number of images taken by calibrated cameras in various positions at distinctly different times. This permits the prediction of the appearance of the entire region from a camera in an arbitrary new position. When a new image is taken by a calibrated camera from a new position, the computer must make a decision as whether a new object has appeared in the region or the image is of the same region [17]. In our change detection problem, which differs from the preceding, the algorithm must decide whether a new object has appeared in the region or whether an object in the region has left. We predict the 3D model which consists only of long and short straight lines, since estimating the complete 3D surface under varying illumination conditions and in the existence of specular highlights is often impractical. Moreover, image edges resulting from reflectance edges or from 3D surface ridges are less sensitive to image view direction and surface illumination, hence so are the 3D curves associated with these edges. For man-made objects and for general 3D curves, straight line approximations are usually appropriate and effective. Our method detects changes by interpreting reconstructed 3D line segments and 2D line segments detected in training and test images.

4.2 Decision Making Procedure

Our change detection method assigns a “state” to each 2D line segment in the test image and each reconstructed 3D line segment from the training images. These states are:

"not-changed", "changed (new)", "changed (removed)" and "occluded". The algorithm has two independent components. The first one determines the "state" of each 2D line segment $l \in V_{n+1}$ where V_{n+1} is the set of all 2D line segments detected in the test image. The second component estimates the "state" of each 3D line segment $L \in W_n$ where W_n is the set of reconstructed 3D line segments using the first n images. The decision of whether or not a change has occurred is based on the appearance (checked by tests T_1 and T_2) and disappearance (checked by tests T_3 and T_4) of a line segment in a given test image.

Change Detection for 2D Lines in the Test Image. We estimate the "state" of each 2D line segment in the new image using two statistical tests T_1 and T_2. The classification scheme for the 2D case is given in figure 5.

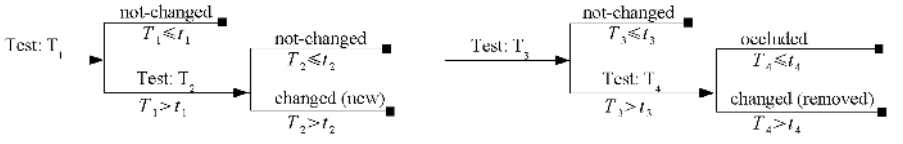

Fig. 5. General scheme for line segment based change detection. (Left) Change detection for 2D line segment in the test image. (Right) Change detection for reconstructed 3D line segments from training images. Threshold values t_1, t_2, t_3 and t_4 are selected to produce the desired change detection rate.

First we apply T_1 to test how well a 2D line segment in I_{n+1} fits to the 3D model W_n. Basically T_1 is the distance of the 2D line segment in the new image to the closest projection into I_{n+1} of the 3D line segment in the 3D model.

$$T_1 = \min_{L \in W_n} d_s(l, l''_{n+1})$$

The second step is, if necessary, to apply T_2 to test if there exists a 2D line segment in one of the past images that has taken from a similar viewing direction. Let's define the index of the image in the training set that has the closest camera projection matrix compared to the test image as c^*.

$$c^* = \arg\min_{i \in \{1,\ldots,n\}} \|M_{n+1} - M_i\|_F$$

where $\|\cdot\|_F$ is the Frobenius norm. T_2 is defined as:

$$T_2 = \min_{l \in V_{c^*}} d_s(l, l''_{c^*})$$

Here we assume that for large training sets, there exists a camera close to the camera that took the test image assuming that camera matrices are normalized.

Change Detection for Reconstructed 3D Line Segments. We estimate the "state" of each reconstructed 3D line segment using two statistical tests T_3 and T_4. The classification scheme for the 3D case is given in figure 5.

First we apply T_3 to test how well a 3D line segment in W_n fits to the 2D lines in the new image. Basically T_3 is the distance of a reconstructed 3D line segment to the closest 2D line segment in the new image.

$$T_3 = \min_{l \in V_{n+1}} d_s(l, l''_{n+1})$$

here l''_{n+1} is the projection of L to the $(n+1)^{st}$ image.

The second step is, if necessary, to apply T_4 to test if there is an existing 3D line segment in W_n that occludes the current 3D line L.

$$T_4 = \min_{G \in W_n \setminus \{L\}} d_s(g''_{n+1}, l''_{n+1}) Z(M_{n+1}, L, G)$$

Here g''_{n+1} is the projection of G into I_{n+1} as a line segment and $Z(M_{n+1}, L, G)$ returns 1 if both endpoints of G are closer to the camera center of M_{n+1} than both endpoints of L, otherwise it returns ∞.

5 Experimental Results

In this section, we present the results of our change detection algorithm in three different image sequences (experiments). It is assumed that the scene geometry does not change during the training period. The reconstruction of the 3D model and the change detection is done using the methods explained in section 3 and section 4. The aim of each experiment is to show that our change detection method successfully detects the changes and type of changes in test images in which the scene geometry is significantly different than the training images.

The first sequence is a collection of 5 training images and 1 test image, all of which are taken in a two hour interval. The result of this experiment is shown in figure 1. The test image is taken at a different time of the day and from a different camera position; hence the illumination and viewpoint direction is significantly different compared to the training images. There are two important changes that have taken place in this test image. The first is the disappearance of a vehicle that was previously parked in the training images. The second change is the appearance of a vehicle close to the empty parking spot. Both major changes (and their types) are detected accurately with a low level of false alarm rate and main regions of change have been successfully predicted. Notice that small "new lines" (shown in "red") in the front row of the parking place are due to the specular highlights that did not exist in the set of training images. There is significant illumination difference between the test and training images, since the test image is taken a few hours after the training images. The red line on the ground of the empty parking spot is due to the shadow of another car. Geometrically, that line was occluded by the car that has left the scene, so it can be thought as an existing line which did not show up in training images due to self occlusion. The result of this experiment shows that our method is robust to severe illumination and viewpoint changes.

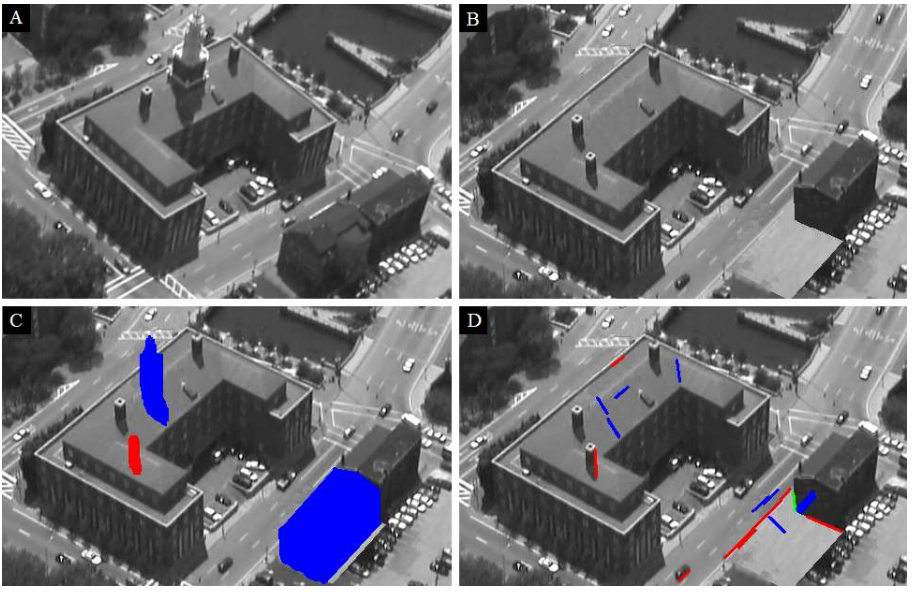

Fig. 6. Change detection results for an urban area after training on a sequence of 20 images. (A) A sample training image. (B) The test image. (C) Hand-marked ground truth for change where the new objects are labeled with "red" and the removed objects are labeled with "blue". (D) Line segment based change detection results in which "new" lines are shown in "red" and "removed" lines are shown in "blue". Our method detects the permanent change regions successfully and also recognizes a moving vehicle as an instance of temporal change. (This is a color image)

Unlike the first image sequence, the second and third image sequences do not have significant viewpoint and illumination changes between the training and test images. The result of the second experiment is shown in figure 6. To test our method, we manually created a few changes using image manipulation tools. We removed the town bell and added another chimney to a building. These change areas are marked with blue and red respectively in figure 6-C. Also the building at the bottom right corner of the test image is removed from the scene. This change is also marked with blue in figure 6-C. These major changes are successfully detected in this experiment. Also, despite the relatively small sizes of cars in this dataset, the change related to a moving vehicle (temporal change) is detected successfully.

The results of the third experiment is shown in figure 7. In this experiment, two new vehicles appear in the road and these changes are detected successfully. Similarly, we created manual changes in the scene geometry using image manipulation tools. We removed a few objects that existed on the terrace of the building in the bottom right part of the test image. These changes are also detected successfully by our change detection algorithm.

We also applied the Grimson change detection algorithm [3] to ground registered images for all sequences (see figure 8). Our 3D geometry based change detection method performs well for the first image sequence under significant viewpoint and illumination

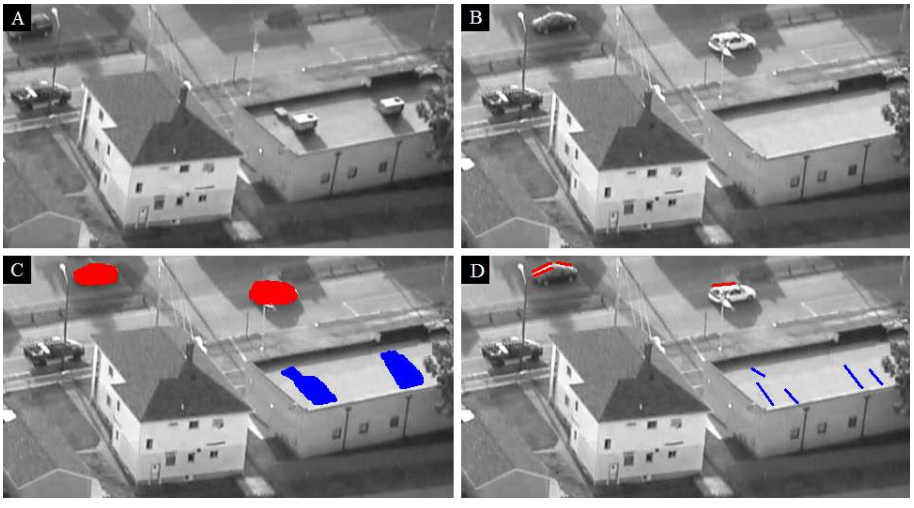

Fig. 7. Change detection results for an urban area after training on a sequence of 10 images. (A) A sample training image. (B) The test image. (C) Hand-marked ground truth for change where the new objects are labeled with "red" and the removed objects are labeled with "blue". (D) Line segment based change detection results in which "new" lines are shown in "red" and "removed" lines are shown in "blue". Our method detects the permanent change regions successfully and also recognizes two moving vehicles as instances of temporal change. (This is a color image)

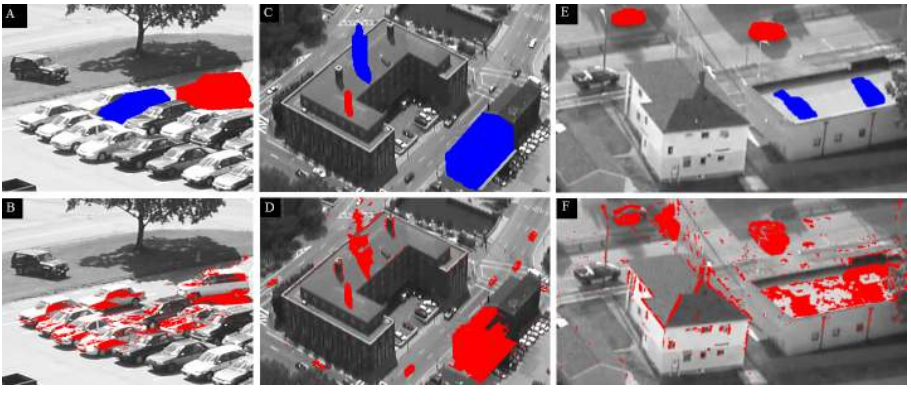

Fig. 8. Results of Grimson change detection algorithm applied to ground registered images. (A-B) Ground truth change and the result of the Grimson algorithm for the first sequence. (C-D) Ground truth change and the result of the Grimson algorithm for the second sequence. (E-F) Ground truth change and the result of the Grimson algorithm for the third sequence. (This is a color image)

differences between the training and test images. On the other hand, the Grimson method fails to detect changes reasonably due to viewpoint and illumination changes, and the existence of specular highlights. For second and third image sequences, the

Grimson method successfully detects changes except insignificant false alarms caused by the small viewpoint change between the training and test images.

To summarize our experimental results, we have shown that the significant change regions are successfully detected in all 3 experiments. The first experimental setup shows that our method can detect changes regardless of the changing viewpoint and illumination conditions. However, there are a few cases of insignificant false alarms possibly caused by shadows, specular highlights and lack of 3D geometry of newly exposed line segments in the test image. Also, it must be noted that unlike other change detection methods, our method detects the type of major changes successfully in almost all experiments.

6 Conclusion and Future Work

In this paper, we present the first generally applicable 3D line segment based change detection method for images taken from arbitrary viewing directions, at different times and under varying illumination conditions. Our method has been shown to detect significant changes with high accuracy on three different change detection experiments. Experiments indicate that our algorithm is capable of efficiently matching and accurately reconstructing small and large line segments, and successfully detecting changes (and their types) by interpreting 2D and 3D line segments. We show that our multi-view line segment matching algorithm works faster than other commonly used matching algorithms, and our 3D line segment reconstruction algorithm, exploring the connectivity of 2D and 3D line segments in order to constrain them in configurations, improve the accuracy of existing individual 3D line segment reconstruction techniques.

Future work will involve detection of shadow and specular highlight regions to improve the result of change detection by reducing the false alarm rate. Additionally, we are investigating ways of propagating the uncertainty of line segments during the 3D reconstruction process, and improving the change detection algorithm to work with uncertain projective geometry. Additionally, some initial experiments have already been performed to interpret the type of change associated with an existing pre-occluded line segment that appears in the test image. It is necessary to build statistical models using larger datasets to better interpret these changes related to newly appeared line segments in the test image.

Acknowledgments. This work was funded, in part, by the Lockheed Martin Corporation. We are grateful to Joseph L. Mundy for many insightful discussions on this work. We also thank Ece Kamar for her helpful comments on earlier drafts of this paper.

References

1. Bruzzone, L., Prieto, D.F.: Automatic analysis of the difference image for unsupervised change detection. IEEE Transactions on Geoscience and Remote Sensing 38(3), 1171–1182 (2000)
2. Bruzzone, L., Prieto, D.F.: An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. IEEE Transactions on Image Processing 11(4), 452–466 (2002)

3. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR, pp. 246–252 (1999)
4. Yalcin, H., Hebert, M., Collins, R.T., Black, M.J.: A flow-based approach to vehicle detection and background mosaicking in airborne video. In: CVPR, vol. II, p. 1202 (2005)
5. Broadhurst, A., Drummond, T., Cipolla, R.: A probabilistic framework for space carving. In: ICCV, pp. 388–393 (2001)
6. Pollard, T., Mundy, J.L.: Change detection in a 3-d world. In: CVPR, pp. 1–6 (2007)
7. Rowe, N.C., Grewe, L.L.: Change detection for linear features in aerial photographs using edge-finding. IEEE Transactions on Geoscience and Remote Sensing 39(7), 1608–1612 (2001)
8. Li, W., Li, X., Wu, Y., Hu, Z.: A novel framework for urban change detection using VHR satellite images. In: ICPR, pp. 312–315 (2006)
9. Schmid, C., Zisserman, A.: The geometry and matching of lines and curves over multiple views. International Journal of Computer Vision 40(3), 199–233 (2000)
10. Heuel, S., Förstner, W.: Matching, reconstructing and grouping 3D lines from multiple views using uncertain projective geometry. In: CVPR, pp. 517–524 (2001)
11. Taillandier, F., Deriche, R.: Reconstruction of 3D linear primitives from multiple views for urban areas modelisation. In: Photogrammetric Computer Vision, vol. B, p. 267 (2002)
12. Bay, H., Ferrari, V., Gool, L.J.V.: Wide-baseline stereo matching with line segments. In: CVPR, vol. I, pp. 329–336 (2005)
13. Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer Journal 7, 308–313 (1965)
14. Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(11), 1021–1032 (1995)
15. Moons, T., Frère, D., Vandekerckhove, J., Gool, L.V.: Automatic modelling and 3d reconstruction of urban house roofs from high resolution aerial imagery. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 410–425. Springer, Heidelberg (1998)
16. Baillard, C., Schmid, C., Zisserman, A., Fitzgibbon, A.W.: Automatic line matching and 3D reconstruction of buildings from multiple views. In: ISPRS Congress, pp. 69–80 (1999)
17. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing 14(3), 294–307 (2005)