Change Detection in a 3-d World

Thomas Pollard and Joseph L. Mundy
Brown University
Providence, RI 02912

Abstract

This paper examines the problem of detecting changes in a 3-d scene from a sequence of images, taken by cameras with arbitrary but known pose. No prior knowledge of the state of normal appearance and geometry of object surfaces is assumed, and abnormal changes can occur in any image of the sequence. To the authors' knowledge, this paper is the first to address the change detection problem in such a general framework. Existing change detection algorithms that exploit multiple image viewpoints typically can detect only motion changes or assume a planar world geometry which cannot cope effectively with appearance changes due to occlusion and un-modeled 3-d scene geometry (egomotion parallax). The approach presented here can manage the complications of unknown and sometimes changing world surfaces by maintaining a 3-d voxel-based model, where probability distributions for surface occupancy and image appearance are stored in each voxel. The probability distributions at each voxel are continuously updated as new images are received. The key question of convergence of this joint estimation problem is answered by a formal proof based on realistic assumptions about the nature of real world scenes. A series of experiments are presented that evaluate change detection accuracy under laboratory-controlled conditions as well as aerial reconnaissance scenarios.

1. Introduction

The problem of detecting changes, typically moving objects, in image sequences taken from stationary cameras has been well studied and many algorithms exist that solve the problem with high accuracy. These algorithms are widely used in surveillance systems as well as in preprocessing video for a variety of computer vision applications. However there are many potential applications for change detection where the images are not taken from a stationary camera, such as in aerial surveillance, where images are collected from a range of different positions, orientations, and at different times of day (see figure 1). Some work has been done on detecting changes in aerial imagery, but it is limited by the range of camera motions allowed and the kinds of changes detected.

The ultimate objective of the research initiated in this paper is a solution to a large class of change detection problems where images are formed by cameras of arbitrary but known position, orientation, and resolution and under very different illumination conditions. The final solution should be a system which processes an image at a time, updates the world model from what has been learned, and discards the image, having incorporated all of the image's relevant information in the model. This paper is the first step toward that goal. The algorithm described here handles the restricted case where the images are of approximately the same resolution and are taken under similar lighting conditions, but may still be taken from arbitrary viewpoints with substantial foreground clutter. The objective of this paper is to demonstrate the algorithm's success at learning surface color and geometry probabilities under these restrictions.

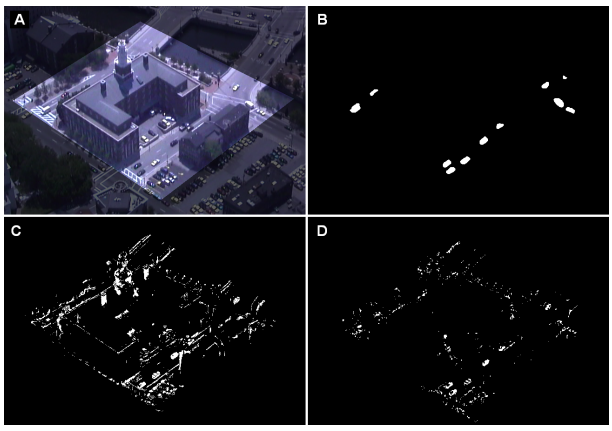

Figure 1. Change detection results for an urban scene after training on a sequence of 40 aerial images. A) an unseen image. B) hand-marked ground truth changes on this image. C) a planar change detection algorithm applied to ground-registered images. D) voxel algorithm change detection results. Voxel algorithm detects all changes with minimal errors, while the planar algorithm makes many large errors on the building.

1-4244-1180-7/07/$25.00 ©2007 IEEE

Under these general camera conditions the world can be viewed from any direction, so some kind of 3-d model for the world is needed. Doing change detection on some precomputed 3-d surface is an insufficient solution as there are many situations where the world surface may be changing over time(i.e. parked cars leaving, buildings being erected/demolished, etc.). The model presented in this paper uses voxels to represent the world, which has the advantage of being able to model the volumetric changes in the scene as well as the appearance changes on the surfaces.

2. Previous Work

A typical change detection algorithm takes an image as input, updates some appearance model of the scene from the information learned from the image, and identifies changes in that image. For stationary video the model of the scene can be as simple as the running average intensity at each pixel, or can be a more complicated probabilistic model possibly involving pixel neighborhoods, *e.g.* a Markov random field [5].

A widely-used technique for detecting change in stationary video is to maintain a Gaussian mixture model at each pixel. This was first presented in Grimson *etal* [8] and improved in many later papers, the most relevant to our work being KaewTraKulPong and Bowden [4]. Algorithms of this class can handle common appearance changes such as objects entering and leaving the scene, moving vegetation, and slow changes in illumination. A thorough survey of such algorithms is given in [7]. There has been limited research regarding change detection algorithms for non-stationary cameras. Mittal and Huttenlocher [6] apply stationary change detection techniques to image mosaics, but their work is restricted to cameras with no motion paralax, i.e. rotation about the camera's optical center. There is a substantial related literature on motion detection [3, 10, 9] but these techniques are restricted to detecting moving objects in a single continuous video sequence.

The change detection algorithm presented in this paper models the world scene with voxels, small units of volume analogous to pixels, used by the space-carving community to estimate surfaces from video or other image sets. Recent work has developed space-carving in a probabilistic framework, most notably [2, 11, 1]. Broadhurst et al.[2] provided some inspiration for the techniques in this paper in particular, most notably their choice of occlusion model, but while space carving is concerned with estimating exact surfaces from a fixed set of images, the change detection problem requires a flexible model of the world that can adapt as additional images are taken.

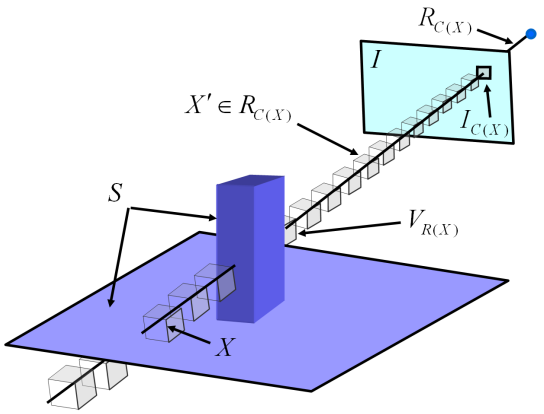

Figure 2. Voxel notation. X is voxel on the world surface $\mathcal{S}$ lying on the ray $R(X)$ and projecting into image I at pixel $I_{C(X)}$. However $V_{R(X)}$ is the voxel that actually produced color $I_{C(X)}$.

3. Algorithm

This section contains a description of the voxel model of the world, an algorithm for updating it as new images come in, and an algorithm for detecting changes in new images. The modeling process involves simultaneous estimation of both the surface of the world and a color model at each point on the surface.

To begin, a bounded volume containing all of the 3-d world of interest is partitioned into voxels, see figure 2. At any point in time, a voxel X has two possible states: a surface in the world; or in empty space or inside a solid object. The state of X being a surface voxel is denoted by $X \in \mathcal{S}$. It will prove beneficial to cast the surface occupancy problem into a Bayesian framework, so it makes sense to talk about probabilities like $P(X \in \mathcal{S})$ which measure one's belief that X is a surface voxel.

For any image I with known camera C the voxel volume can be partitioned into rays of voxels that project into common locations in the image. The ray to which voxel X belongs is denoted by $R(X)$ and the color intensity to which X maps in the image by $I_{C(X)}$. For any ray the voxels $X' \in R(X)$ can be ordered with the inequality $X'' > X'$, when X'' is further along the ray from the camera center than X'. Also for each ray there is exactly one voxel which produced the intensity seen in the image and this voxel is denoted by $V_{R(X)}$ or simply by V when the ray is understood.

In this paper only grey scale images are addressed, but the algorithm is easily generalized to color imagery, and hereafter pixel intensity will be referred to as color for convenience. The prediction of pixel colors observed in the images is based on the Gaussian mixture color models, however there exists the complication that for any pixel in the image there are many voxels in the world that can

potentially produce the observed color. The solution is to maintain a mixture of Gaussians model at each voxel to be used when it is time to consider the possibility that this voxel produced the observed color. The probability density $P(I_{C(X)}|V_{R(X)} = X)$ has a mixture of Gaussians distribution for voxel X.

Given a sequence of images $\{I^t\}$ and cameras $\{C^t\}$, for the first image the surface probability, $P^0(X \in \mathcal{S})$, for each voxel X is initialized to some common constant and the mixture of Gaussians model is initialized with the pixel color observed, $I^0_{C^0(X)}$. After observing image I^{t+1} the surface estimates $P^t(X \in \mathcal{S})$ and color distributions $P^t(I_{C(X)}|V_{R(X)} = X)$ for each voxel X must be updated. These two probabilities are updated separately as described in the sections 3.1 and 3.2. Section 3.3 discusses what can be said about convergence of the proposed model and section 3.4 gives equations for detecting change in a new image after the model has been trained on sufficiently many images.

3.1. Updating the Surface Probabilities

This section describes the procedure for updating the surface probability $P^t(X \in \mathcal{S})$ for voxel X with information from I^{t+1}. The assumption is made that $I^{t+1}_{C(X)}$ (from here on simply I_X) is the only relevant pixel in I^{t+1} for updating voxel X, as pixel neighborhoods are not used in this paper. With this assumption it is a consequence of Bayes rule that:

$$P^{t+1}(X \in \mathcal{S}) = P^t(X \in \mathcal{S}) \frac{P^t(I_X|X \in \mathcal{S})}{P^t(I_X)} \quad (1)$$

After expanding the color probability terms by voxels along $R(X)$ that actually produced the color, the multiplier for $P^t(X \in \mathcal{S})$ in equation 1 equals:

$$\frac{\sum_{X' \in R(X)} P^t(I_X|V = X')P^t(V = X'|X \in \mathcal{S})}{\sum_{X' \in R(X)} P^t(I_X|V = X')P^t(V = X')} \quad (2)$$

$P^t(I_X|V = X')$ is computed using the mixture of Gaussians color model stored in voxel X' after training on the first t images. The only question left is how to compute $P^t(V = X')$ and $P^t(V = X'|X \in \mathcal{S})$.

Consider $P^t(V = X')$, which is the probability that voxel X' produced the pixel color in the image considering only the voxel surface probabilities along the ray, and not any color information. From a geometric viewpoint, the voxel will produce the color if and only if the voxel is on the surface and it is unoccluded:

$$P^t(V = X') = P^t(X' \in \mathcal{S})P^t(X' \text{ is not occluded}) \quad (3)$$

Experiments were done with several occlusion models and similar results obtained each time, so the model with the most immediate physical interpretation is used here,

$$P^t(X' \text{ is not occluded }) = \prod_{X'' < X'} (1 - P^t(X'' \in \mathcal{S})), \quad (4)$$

which is the probability that all voxels between X' and the camera contain empty space. This completes the definition of $P^t(V = X')$.The equation for $P^t(V = X'|X \in \mathcal{S})$ is the same except that any instances of $P^t(X \in \mathcal{S})$ in the above have probability 1. Note that the update equations for $P^{t+1}(X \in \mathcal{S})$ have been derived using only the current surface and color probabilities for voxels lying in the same ray, and so are computable. To better understand the update multiplier in equation 2, rewrite it as follows:

$$\frac{preX^t + visX^t * P^t(I_X|V=X)}{preX^t + visX^t \left[P^t(X)P^t(I_X|V=X) + \left(1 - P^t(X)\right) postX^t \right]} \quad (5)$$

where:

$$\begin{aligned}
visX^t &= \prod_{X'' < X} \left(1 - P^t(X'')\right) = P^t(X' \text{ is not occluded}) \\
preX^t &= \sum_{X' < X} \left(P^t(I_X|V=X')P^t(X') \prod_{X'' < X'} \left(1 - P^t(X'')\right) \right) \\
postX^t &= \sum_{X' > X} \left(P^t(I_X|V=X')P^t(X') \prod_{X \leq X'' < X'} \left(1 - P^t(X'')\right) \right)
\end{aligned}$$

$preX^t$ and $postX^t$ are the probabilities of seeing the color when considering only the voxels on the ray above and below X respectively. It follows that the update multiplier is greater than 1 if and only if X explains the color better than the voxels below it.

3.2. Updating the Color Models

The update equations for the mixture of Gaussians color model used here are a modified version of the equations proposed in [4], which are an enhanced version of the Stauffer and Grimson change detection algorithm [8]. In this implementation a Gaussian mixture model is a combination of K Gaussian distributions with means μ_k, standard deviations σ_k, and weights w_k that are discussed later in this subsection. Each mode has distribution:

$$\eta_k(y) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(y-\mu_k)^2}{2\sigma_k^2}} \quad (6)$$

and the full mixture distribution is:

$$p(y) = \sum_{k=1}^{K} \frac{w_k}{W} \eta_k(y), \quad W = \sum_{k=1}^{K} w_k \quad (7)$$

As in [4] the modes are ranked by w_k/σ_k and to train the distribution on a new color c and weight dw the first mode

for which c lies within 2.5 standard deviations of the mean is updated with the following equations:

$$w_k^{n+1} = w_k^n + dw$$
$$\mu_k^{n+1} = \mu_k^n + \frac{dw}{dw + w_k^n}(c - \mu_k^n)$$
$$(\sigma_k^{n+1})^2 = (\sigma_k^n)^2 + \frac{dw}{dw + w_k^n}((c - \mu_k^n)^2 - (\sigma_k^n)^2) \quad (8)$$

These equations are the same as the EM-algorithm derived result in [4] in the case where the weights are constant. If c is not within 2.5 standard deviations of any mode, the least probable mode is destroyed and replaced with a high variance mode with mean c and weight dw.

The new weighting scheme is introduced here to handle the problem of many voxels projecting into the same pixel. Consider a ray R containing voxels $\{X'\}$. It is not known for certain which voxel produced the color in the image, but the probability of each voxel X' producing the color, $P^t(V = X')$, can be computed. To be fair, all of the voxels $\{X'\}$ along the ray must be trained on the color seen, but how little or how much a color effects a voxel's Gaussian mixture model can be controlled by choosing appropriate update weights, and the natural answer is to use the $P^t(V = X')$ as such weights, so that a voxel's color model is updated proportionately to how likely it is that the voxel produced the color seen. For a sometimes occluded surface voxel this scheme prevents occluding surface colors from seriously corrupting the voxel's color model.

3.3. Convergence

A key question is under what conditions this simultaneous modeling of geometry and appearance is guaranteed to converge, which shall be defined to occur when empty space voxels have their surface probabilities go to 0 and true surface voxels have their surface probabilities go to 1 and color models go to a single Gaussian mode with the minimum allowed variance. The convergence of a voxel is highly dependent on the range of viewing directions and local color variation in the world and in general cannot be met for every voxel in a scene. For example the exact surface of a uniformly colored roof viewed only from the air cannot be determined, as many voxels lying above and below the actual roof have color models identical to those of the true surface voxels. Still general conditions can be derived that guarantee a voxel's convergence. The case of convergence of a surface voxel X is discussed in detail below, and the argument for an empty space voxel is similar.

The first assumption made is that when it is unoccluded the surface at X projects the same color in each image, though in practice convergence can still occur when there are minimal variations in observed color. Suppose first that X is unoccluded in all views. Then X will see the same color in each image and eventually $P^t(I_X|V = X)$ is guaranteed to be maximal (as the mixture model variance is capped in practice). In particular it will be greater than $postX^t$, so the update multiplier in equation 5 will be greater than 1 and the surface probability for this voxel will be increasing with each new image. Increasing, but not necessarily to 1, and in regions of ambiguous projection in the world it will converge to something less than 1. The denominator of equation 5 can be rewritten as:

$$preX^t + visX^t * P^t(I_X|V = X) + ...$$
$$visX^t\big(1 - P^t(X)\big)\Big[postX^t - P^t(I_X|V = X)\Big] \quad (9)$$

and after substituting into equation 5 it is clear that the surface probabilities for X as a function of images trained on are a sequence of form:

$$p_{t+1} = p_t \frac{a_t}{a_t - b_t(1 - p_t)} \quad (10)$$

where:

$$a_t = preX^t + visX^t * P^t(I_X|V = X) \quad (11)$$
$$b_t = visX^t\Big[P^t(I_X|V = X) - postX^t\Big]$$

This sequence converges up to 1 provided that the $\{b_t\}$ are bounded from below (or contain a bounded subsequence). So there are two conditions for convergence of the voxel's surface probability to 1. Firstly $visX^t$ must be bounded from below, which will be automatically true for a voxel unoccluded in all views as the empty space voxels above it have decreasing surface probabilities via a modification of the above argument. Secondly $P^t(I_X|V = X) - postX^t$ must be bounded, requiring the voxels lying below the surface at X to be trained on some range of colors distinct from the color at X. If these conditions are met, X will converge.

If X is occluded in some views, by what has been shown above the occluding voxels will converge after enough images, preventing the occluding surfaces' colors from corrupting the color model at X. After sufficiently many images of the true surface color, the mixture distribution at X will be maximal, and we are in the same situation as before, and the surface probabilities here will again converge to 1. And so the following has been proven:

Convergence Theorem 1 *A voxel X lying on a world surface will converge provided that the surface images a constant color in all views, and that all voxels X' near X lying beneath the surface frequently project into some pixels of sufficiently different color from the true surface color at X.*

3.4. Detecting Change

Suppose that the model has been trained on an image sequence and it is desired to detect change in a new image

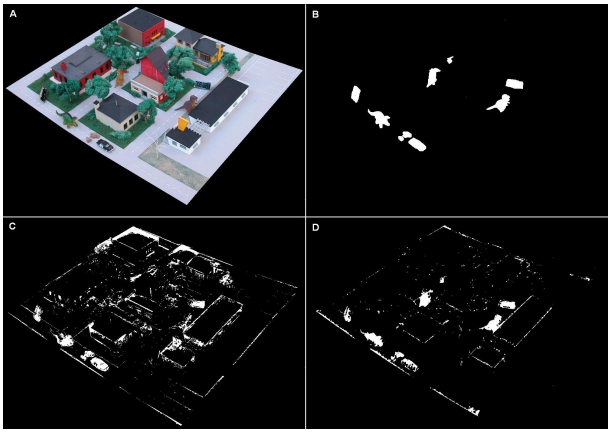

Figure 3. Change detection results for model town sequence. A) an unseen image. B) hand-marked ground truth changes on this image. C) a planar change detection algorithm applied to ground-registered images. D) voxel algorithm change detection results. Planar algorithm has significant errors do to unmodeled geometry.

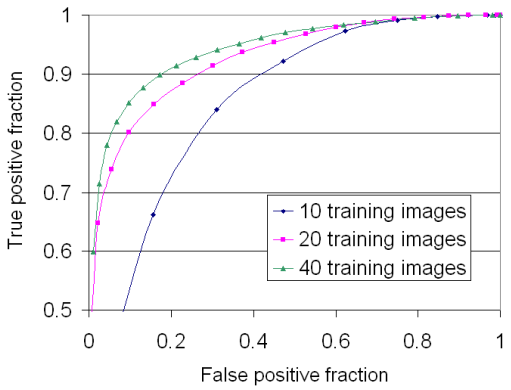

Figure 4. Change detection ROC curves for the voxel algorithm applied to the model town sequence with different numbers of training images. 20 images is sufficient to produce good change detection, with more images providing marginal improvement.

I with camera C. Consider a pixel I_X which backprojects into a ray R of voxels $\{X'\}$. The probability of seeing this color is then,

$$P(I_X) = \sum_{X' \in R} P^t(I_X|V = X')P^t(V = X') \qquad (12)$$

which are all computable quantities. These probabilities are then thresholded for all the pixels in the image to obtain a binary mask representing the detected change.

4. Results

This section presents change detection results on various image sequences. Camera matrices for all images are computed from manual correspondences to within 1 pixel error. In each experiment, changes are detected at different thresholds on 5-6 images for which ground truth change is hand marked for comparison. The false positive error rate is the fraction of mislabeled pixels in all images to the total number of pixels. Results are also given for a change detection algorithm where Gaussian mixture models are confined to a single world plane and images are registered to this plane,[4]. This planar algorithm performs better than one might expect in the presence of significant motion paralax, however many of the buildings viewed are uniformly textured and different positions on the surface can be represented by a single Gaussian mixture model. The error rates for both algorithms are presented as receiver operating characteristic (ROC) curves which plot the fraction of correctly detected changes against the fraction of falsely detected changes.

The first sequence is a collection of 40 images taken of a lifelike miniature town constructed from detailed model railroad kits and shot under a diffuse lighting condition. Each image was taken with a different arrangement of miniature cars and other clutter to simulate a dynamic environment. A sample image and detected changes are shown in figure 3, which displays the strengths and weaknesses of the algorithm presented here. The true changes in the scene are correctly marked and most errors occur at object boundaries in the image. These alignment errors occur because the color distributions at the true surface voxels producing these pixels are very sensitive to camera projection errors. For example, at a voxel on a black to white edge in the world the color model will be trained on pixel samples that are black, white, and everything in between, yielding a high variance mixture distribution that tends to mark anything as change. This problem will be addressed by taking voxel neighborhoods into account in future work.

In one experiment on this model town sequence, the proposed voxel algorithm is run three times on the image set with a different number of training images in each run. ROC curves are given in figure 4. As seen, 20 images is enough to produce highly accurate change detection results with additional images providing marginal improvement.

In another experiment with the model town, both voxel and planar change detection algorithms are trained on the 40 image sequence and changes detected on three different 5-image sets taken at varying viewing angles. ROC curves are given in figure 5. The voxel algorithm performs far better at all angles and though the accuracy of both algorithms degrades as the cameras move from a high angle to a more grazing angle, the planar algorithm performance degrades more rapidly.

Experiments are also done on two 40-image aerial sequences shot from helicopter. A sample frame from one sequence and detected changes are shown in figure 1. Many of

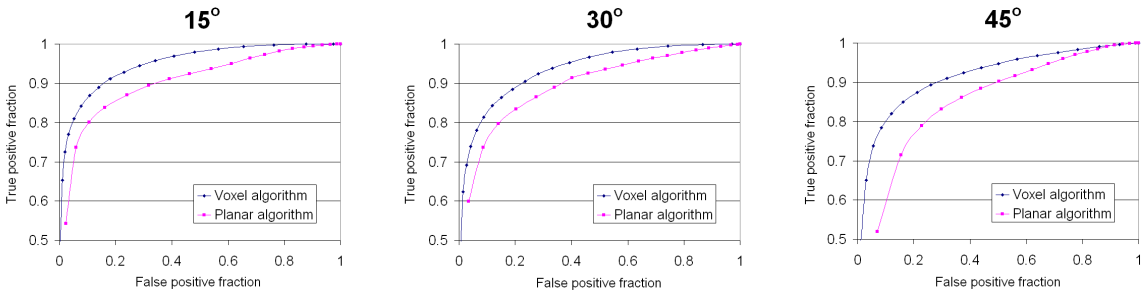

Figure 5. Change detection ROC curves for the model town sequence. Results are given for different viewing angles, measured from the normal to the ground plane. Voxel algorithm gives superior all around performance, with a greater difference at larger angles.

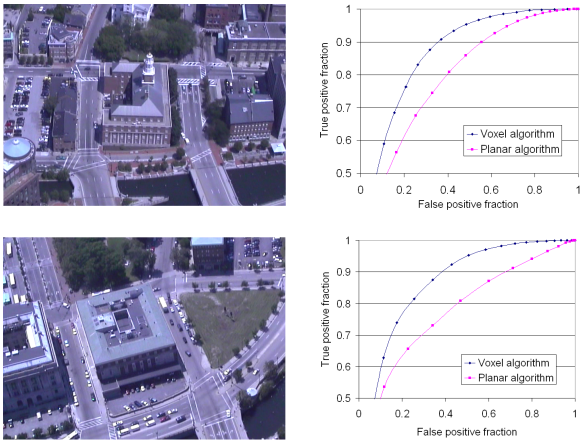

Figure 6. Sample images and change detection ROC curves for two aerial sequences. Violations of the constant color assumption cause more false positives in this sequence than for the model town, but the voxel algorithm still outperforms the planar algorithm significantly.

the false changes detected are specular highlights on parked cars and metal roofs, and this is to be expected as they are violations of the constant color intensity assumption. The voxel algorithm outperforms the planar algorithm by a wider margin in this real world scene as demonstrated by the ROC curves for both sequences shown in figure 6.

5. Conclusions and Further Work

This paper presented the first fully general model for change detection for images taken from arbitrary posed cameras. Conditions for the convergence of the update algorithm have been derived and convergence proved. The algorithm has been shown to produce excellent results on a number of real and miniature scenes. Quantitative comparisons with a planar based model demonstrate the power of using a volumetric description of the world. The limitations and shortcomings of the algorithm have been explored and found to be manageable. This algorithm presents a general framework so it will be easy to adapt new techniques as they are developed.

Future work will involve relaxing the constant resolution and illumination restrictions put on the images taken of the scene. Also some initial experiments have been performed to physically predict shadows from the learned geometry. This approach shows promise toward overcoming the need for a constant color intensity assumption.

References

[1] M. Agrawal and L. S. Davis. A probabilistic framework for surface reconstruction from multiple images. In *CVPR*, 2001.

[2] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *ICCV*, 2001.

[3] E. Hayman and J.-O. Eklundh. Statistical background subtraction for a mobile observer. In *ICCV*, 2003.

[4] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[5] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

[6] A. Mittal and D. Huttenlocher. Scene modeling for wide area surveillance and image synthesis. In *CVPR*, pages 160–167, 2000.

[7] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Trans on Image Processing, Vol. 14, No. 3*, 2005.

[8] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, pages 246–252, 1999.

[9] H. Tao, H. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *CVPR*, 2000.

[10] H. Yalcin, R. Collins, M. Hebert, and M. J. Black. A flow-based approach to vehicle detection and background mosaicking in airborne video. In *Video Proceedings in conjunction with CVPR'05*, June 2005.

[11] A. Yao and A. Calway. Dense 3-d structure from image sequences using probabilistic depth carving. In *ECCV*, 2004.