Scene Modeling for Wide Area Surveillance and Image Synthesis

Anurag Mittal, Dan Huttenlocher
Computer Science Department
Cornell University
Ithaca, NY 14850 USA
{anurag, dph}@cs.cornell.edu

Abstract

We present a method for modeling a scene that is observed by a moving camera, where only a portion of the scene is visible at any time. This method uses mixture models to represent pixels in a panoramic view, and to construct a "background image" that contains only static (non-moving) parts of the scene. The method can be used to reliably detect moving objects in a video sequence, detect patterns of activity over a wide field of view, and remove moving objects from a video or panoramic mosaic. The method also yields improved results in detecting moving objects and in constructing mosaics in the presence of moving objects, when compared with techniques that are not based on scene modeling. We present examples illustrating the results.

Keywords: Image Registration, Activity Monitoring, Moving Object Detection.

1 Introduction

In this paper we address the problem of monitoring activity over a wide area, using a moving camera. There has been considerable recent work on constructing wide area or panoramic views from multiple images (e.g., [11], [14]), however such work generally presumes that the scene is static (there are no moving objects in the field of view). There has also been recent work on activity monitoring and detection (e.g., [1], [2]), however such work generally assumes a non-moving camera. In contrast, we address the problem of activity monitoring and detection over a wide area, where a moving camera is used to "sweep" over the area of interest. Our approach is based on a combination of the image mosaic techniques that have been used for constructing panoramic views, and the mixture model techniques that have been used for activity monitoring. We create a model of the wide field of view that can be used to distinguish between the static (or stationary) *background* and the moving objects, or *foreground*.

Our method can be used for a number of applications, both in surveillance and monitoring and in image synthesis. For surveillance and monitoring, the method can be used to detect moving objects over a wide field of view area, to detect common patterns of activity over that field of view, and to detect activity that does not fit the common patterns. For image synthesis applications, our method can be used to create panoramic views that contain just the non-moving objects in a scene, to create synthetic panoramic views that place each moving object at just a single location, and to create synthetic videos that remove all or selected moving objects. In this paper we present examples illustrating these applications.

Background modeling in a wide field of view is not only useful for the above applications, it also enables improved accuracy in distinguishing between moving objects and the background. These improvements result from the fact that the wide field of view model has more information, and more stable information, than is present in individual image frames or adjacent frames. The same effect of increased accuracy is observed in standard image mosaic techniques for constructing panoramic views of static scenes. The most accurate image mosaic techniques are based on aligning each image frame with the overall mosaic rather than simply with the previous frame (or with a few frames nearby in time). We illustrate these improvements in accuracy by contrasting results using our wide area background modeling method with previous techniques. We show both that mosaic quality is improved and that very small moving objects can be detected.

2 Related Work

Many methods have been used for background modeling. Although most of these methods deal only with a fixed camera, they provide a good starting point for a moving camera scene. Simple methods include averaging the pixels at a particular location, taking

the median of all the values at a location, and calculating spatially weighted values in order to reduce the effect of outliers. Such techniques, which do not explicitly model background versus foreground, are of limited value in practice. Ridder et. al. [2] and others employ a Kalman filter-based background model. Each pixel is modeled using a Kalman filter and is updated in each frame differently depending on whether it is hypothesized to be part of the background or not. This approach, however, is not well suited to a changing background or a multi-modal background. Moreover even when an observed pixel value is part of the foreground, it has an effect on the background model.

Friedman and Russell [3] and Stauffer et. al. [1] take approaches based on using mixture models to represent the background. Friedman and Russell try to classify the pixels into three distributions, corresponding to the road color, the shadows and the car colors. This makes their work somewhat restricted to such scenarios, although the method can probably be applied to other ones. Stauffer et. al. use a more general scheme, which is the basis of the method used in our work also.

There is a large literature on image mosaic techniques for constructing panoramic views (e.g., [11], [12], [14]). Most approaches to this problem assume that there is not significant motion parallax, that is, depth variations in the scene are not apparent from the motion of the camera. This can be guaranteed by rotating the camera about its optical center (an approach taken by commercial systems such as QuicktimeVR), and also holds true for most cameras once objects are a few tens of feet away. We follow this assumption of no motion parallax, solving for a planar projective transformation that registers one image with another. The most accurate techniques for constructing panoramic views solve for such a transformation between each image frame and the panoramic view that has been constructed thus far. This helps avoid cascading errors that occur if each image frame is simply registered with the next one.

In general, image mosaic techniques assume a static scene. The presence of moving objects is problematic in two regards. First, and most important, such objects can throw off the image registration process because they provide incorrect information about how the images should be aligned. This results in a poor quality panoramic image. Registration errors can be addressed through the use of robust statistical techniques, and are also somewhat ameliorated by the use of pyramid-based registration methods ([5]). The second issue with moving objects is in the construction of the panoramic view. Simply taking the most recent pixel value, or the average pixel value, in constructing the panorama yields an overall image that contains bits and pieces of moving objects. The explicit background modeling of our approach addresses both of these problems.

3 General Overview of the Algorithm

Our method uses mixture models to form a model of the background. We represent each pixel location in the mosaic by a mixture of Gaussians. From these mixture models, we form an image representing the background by taking the mean of the highest weight Gaussian for each pixel. The highest weight Gaussian is the one that has the highest probability of occurrence. Without a lighting change, this will be the one that accounted for the largest number of observed pixel values, thus we assume that the background is observed more often than any foreground objects at each location. This background image can be used as a panoramic view of the "background scene", without any foreground objects.

For each new image obtained from the camera, we register it to this panoramic background image using any known good registration technique. The current image, having been registered with the background image, provides the input pixel values for updatng the mixture models at each pixel where new data is observed. Once these mixture models have been updated, a new background image is computed from these updated mixture models. This process is repeated for each new frame obtained.

As noted in the previous section, the presence of moving objects can lead to an erroneous registration results. By aligning each image frame with the background image, our technique avoids this problem. As the background image does not contain the moving objects, there is generally no good match for the portions of the image corresponding to moving objects, and thus they have little effect on the solution for the best registration transformation.

For the problem of detecting moving objects, a background image can also be very useful. Most techniques for detecting moving objects operate by registering successive image frames, and then subtracting the registered frames to see where there are differences. This only detects regions where the images are different, which is not necessarily the entire moving object (e.g., when part of object overlaps in the two frames). In contrast, when registering an image frame to a panoramic background image, the entire moving object can be readily detected based on the difference.

4 Background Modeling

We model the scene using a mixture model for each of the pixel locations in the mosaic. The probability of a pixel belonging to a particular Gaussian is proportional to the weight ascribed to that Gaussian. Within a particular model, the probability is distributed according to the Gaussian probability distribution scheme. More specifically, the probability of a pixel having an intensity value x, given that it belongs to a particular Gaussian j, is

$$Pr(X = x|J = j) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\mu_j)^2}{\sigma_j^2}} \quad (1)$$

where the random variable J denotes the Gaussian that a pixel belongs to, and μ_j and σ_j are the mean and standard deviation of the jth Gaussian. This scheme can easily be extended to color images by using multi-dimensional Gaussians, but here, we will deal only with 1-d Gaussians and gray images rather than color ones.

The probability that a pixel has intensity value x is then simply

$$Pr(X = x) = \sum_{j=1}^{K} w_j Pr(X = x|J = j) \quad (2)$$

where $w_j = Pr(J = j)$ is the weight of the jth model.

This scheme is useful in modeling multi-modal backgrounds and scene changes, as well as modeling transient moving objects. A multi-modal distribution can result from the swaying of trees, blinking of lights, or any other periodic change in image intensity. In a slow scene change, such as occurs with most outdoor changes in illumination, the weights of the different models would gradually shift to the new background. Often, the background change is only transient and is due to some moving object. In such a case, the weights would gradually revert back after the moving object is removed.

One can determine the best mixture model given the previous n values using what is called the Expectation Maximization (EM) algorithm (see Dempster et. al. [6]). We could run this algorithm for each time frame by taking a window of the previous n frames, or calculating the result for all the pixels so far. However, this algorithm is quite time-consuming and is impractical to run for each frame. Neal and Hinton [7] provide a scheme for incrementally updating the solution using the new input values which has been used successfully by Friedman and Russell [3] to get good mixture models. The incremental scheme does not yield the best solution, but only guarantees to converge to a local minimum. However, it has the drawback of not being able to handle changing background scenes properly since it does not have any scheme to reduce the effect of previous input values. The results when the background has more than the given number of Gaussians are also not clear.

Instead of using the expectation maximization scheme, we use a conceptually simpler scheme which is computationally less expensive and we have found also yields better results for our problem. We determine the mixture models by incrementally updating the models using the intensity values of the pixels that are registered at a particular location. First of all, we determine which model (Gaussian) the observed pixel value belongs to. A pixel can belong to a particular model if its distance from the mean is within a constant (a value between 2 and 3 is suitable) times the standard deviation. If a pixel belongs to more than one Gaussian, we take the one having the highest weight. This is done so as to avoid duplication of the gaussian models. The intensity value of the current pixel is then used to update the models. We can use a constant weight updating scheme and an exponential weighting scheme. However, since the different schemes are useful in different scenarios, we use both of them, switching between the two depending on some criteria which we will define shortly. Experimentally, this combined scheme works better than either of the schemes used alone. Below we discuss these two schemes and how to select between them.

The mean and variance of the matched distribution are updated as follows

$$\mu_{j,t} = (1 - \rho)\mu_{j,t-1} + \rho x_t \quad (3)$$

$$\sigma_{j,t}^2 = (1 - \rho)\sigma_{j,t-1}^2 + \rho(x_t - \mu_{j,t})^T(x_t - \mu_{j,t}) \quad (4)$$

where ρ is a constant which determines the rate of change of the mean and variance.

Note that the mean and variance are updated according to an exponential scheme where the recent pixels get exponentially higher weight. Although a constant weight scheme can also be used, the exponentially weighted scheme would be able to capture slowly changing scenes more easily and hence is used here.

If there is no model that the current pixel belongs to, a new Gaussian is added with a mean equal to that of the pixel value and a high variance. The weight is determined according to the model we are using. Due to a limited amount of memory space, we also require garbage collection when the number of models exceeds

some maximum value. At that time, we simply remove the model with the least weight.

The weights of the distributions are also updated, using either a constant or exponential model, as we now describe.

4.1 Constant Weight Updating Scheme

In this updating scheme, we update the weights of the models in a manner that awards equal weight to all the pixels registering to a particular location in the mosaic. To calculate the weights, we keep track of the number of pixels matching a particular mixture model and also the total number of pixels at that particular location. The weight for a given model is then simply the number of matching pixels divided by the total number of pixels. The new mean and variance are calculated using equations (3) and (4).

This scheme yields a result which works quite nicely in practice as long as the background does not change. The background image, corresponding to the means of the highest weight Gaussian models, would consist of those intensities that are visible for the longest amount of time. Even if the view is obstructed for considerable time by a transient object such as a person standing for a long time, or a car stopping at an intersection, the background would still not change for a long time (presuming many observations of the background intensity beforehand). Thus, it would yield the correct result even in the case of slow moving objects on a road, or a road obstructed by transient objects for a long time.

This model would encounter problems, however, if the background changes abruptly due to a large obstructing object or lighting changes. A large obstructing object cannot be modeled properly by any scheme, but it can be detected by our algorithm easily by a large mean square error and the large number of unmatched points in the image. We handle the lighting changes by using the exponential model when there is a lighting change as opposed to a registration error.

4.2 Exponential Weight Updating Scheme

In the exponential weighting scheme, we update the model parameters by awarding exponentially less weight to values that were observed earlier in time. The weights of the models are updated as follows. For each model j, the new weights are

$$w_{j,t} = (1-\alpha)w_{j,t-1} + \alpha(M_{j,t}) \quad (5)$$

where α is a constant determining the rate of change of the models, and $M_{j,t}$ is 1 if the current pixel is matched to the model j, and is equal to 0 if it is not the matched distribution. As with the constant weight updating scheme, the mean and variance are updated using equations (3) and (4).

The exponential scheme has been previously used by Stauffer et. al. [1]. However, used alone, this creates a tradeoff as far as the rate of change of the model is concerned. A slow rate of change would be unable to model the lighting or background changes properly and fast enough. In a fixed camera scene, this could mean a wrong result for quite some time, while in a moving camera scene, it could even throw off the background registration. On the other hand, a fast rate of change would mean that a foreground object would start to appear as background in a very small amount of time. Also there are problems due to the exponential nature of the updating, which are especially visible when the number of pixels at a particular location is small or when there are slowly moving foreground objects.

4.3 Choosing the Weight Update Scheme

In our algorithm, we use the constant weight updating scheme in the normal mode. In the case that a lighting change is detected, the weights are updated using the exponential scheme where the number of pixels belonging to the models changes exponentially, i.e.

$$n_{j,t} = (1-\alpha)n_{j,t-1} + \alpha(M_{j,t}) * (Totalno.ofpixels) \quad (6)$$

where the definitions are similar to equation (5). Note that this would require these variables to assume real values as opposed to integer values.

In order to distinguish between different scenarios, we calculate three quantities: (1) the normalized correlation between the background image and the current frame warped according to the calculated projective transformation values; (2) the mean square error between the background image and the warped current frame; and (3) the number of points in the current frame that do not match any of the Gaussian models with a large enough weight. These quantities are used to estimate whether there is a lighting change, registration error or other situation that warrants changing how the models are updated for the given frame.

A high normalized correlation and a high number of unmatched points is taken to indicate a lighting change, because the normalized correlation is not affected by a overall additive or multiplicative changes but most pixels will not find a good matching model in such a case. When this occurs, we use the exponential weight updating scheme for the models, so that the change in lighting is quickly adapted to by the background models.

A low normalized correlation, combined with a high number of unmatched points and a high mean square

error is taken to indicate either a complete change of background or a registration error. In this case, the models are not updated; the frame is simply discarded. A low number of unmatched points and a high mean square error are taken to indicate a transient object occupying part of the image. The program should continue to run in the normal mode (constant weight updating).

5 Image Registration and Mosaicing

As we have discussed in the introduction, our algorithm provides a method for image registration and mosaicing that is robust with respect to moving objects. In contrast, most of the methods currently used for image registration and mosaicing can easily be thrown off by the presence of objects which have image motion that differs substantially from that of the background. This can introduce errors in the registration, and these errors can accumulate over time. In such cases the resulting mosaic will not be very good. In the results section below we illustrate this difference using an aerial video sequence.

Our method for image registration and mosaicing is based on registering the current frame with the background image that has been derived from the mixture models. This background image contains at each pixel, the mean of the highest weighted Gaussian, and is an approximation to the highest probability value at each pixel. The current frame can be aligned to this background image using any good registation technique. In our current implementation we use a hybrid registration method that first solves for an affine transformation based on feature correspondences, and then uses that transformation to initialize a direct method of solving for a projective transformation. The first step uses the KLT feature tracker (described in [9] and [10]) to find corresponding features in the images and then solves for an affine transformation using a robust least squares fit. This affine transformation is used to initialize a direct method which yields a projective transformation between the images. The direct method operates in an iterative manner using the Levenberg-Marquardt method, a well known algorithm that is described in various papers (e.g., [5], [8]). It is also possible to use the Levenberg-Marquardt method directly, although use of the KLT tracker speeds up the registration.

Our algorithm provides a method for mosaicing that is robust to the presence of moving objects. First, as with several traditional mosaic techniques, we register each image to the entire mosaic rather than simply the previous frame. This limits cascading of registration errors. Second, when registering an image with the background mosaic, it is unlikely that there will be matches for moving objects. Thus the moving objects will not throw off the registration process. In general the background image does not contain any foreground objects (although when there are just a few images corresponding to a given pixel this may happen). Thus there is generally nothing in the background image that matches the foreground objects well, and the resulting registration is not affected by the moving objects.

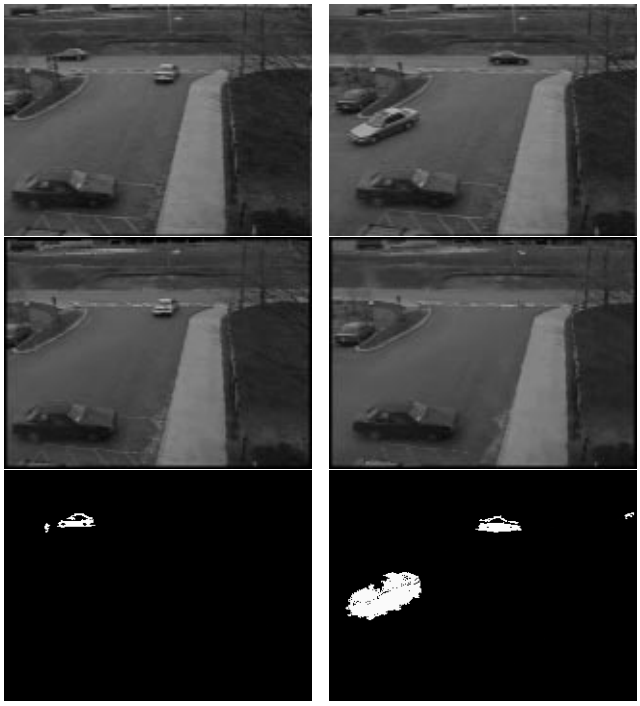

Figure 1: (a) Two views of a parking lot using a moving surveillance camera, (b) synthesized frames with foreground objects removed, and (c) the moving objects detected. Note that very small objects, such a person walking and a car entering behind trees were correctly detected.

This method also provides a good means of aligning successive frames, by constructing a background mosaic (although portions of the mosaic can be discarded over time if only successive frames are to be aligned).

6 Detecting Foreground Objects and Site Classification

To detect foreground (moving) objects in the current frame we first compute the registration of the frame to the panoramic background mosaic. Then we warp each pixel to the coordinate system of the mosaic, and search the nearby pixels for a matching Gaus-

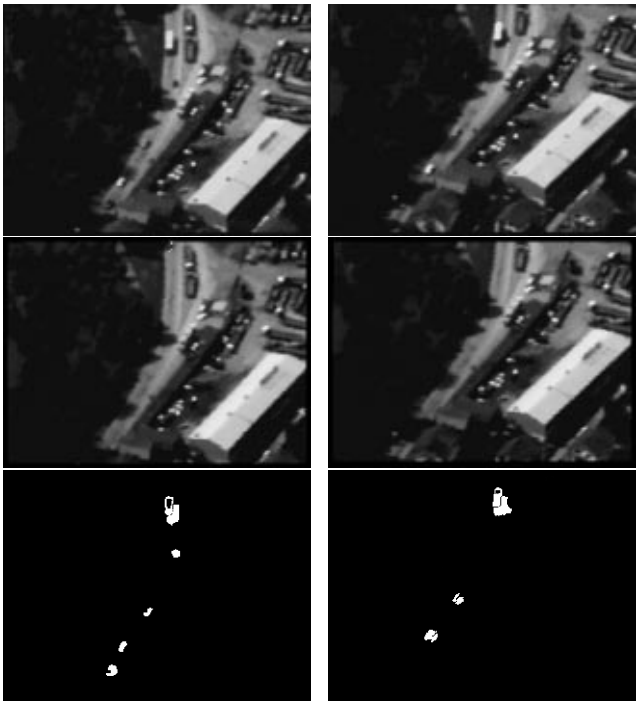

Figure 2: (a) Frame no. 42 and 57 from an aerial video, (b) synthesized frames with the moving objects removed, and (c) the corresponding moving objects detected.

Figure 3: The background mosaic from an aerial video of 150 frames using our algorithm

sian model. A small neighborhood is searched so as to allow for small alignment errors or small changes in background that may be caused by the swaying of trees etc. If the pixel does not match to any of the Gaussians in this neighborhood with a sufficiently large weight, we declare this point to be new, else it is part of the background. We finally run a connected components algorithm on this image to get rid of noise. Sufficiently large objects can then be taken as the foreground objects.

Note that slowly moving objects will not be detected as foreground objects since the weights of the Gaussians that such pixel values belong to will not be high enough to be counted as background. Thus slowly moving objects, which are often problematic in exponential forgetting algorithms, are not an issue here. The approach is able to quite accurately distinguish between moving objects and the background.

Based on the foreground objects that are detected in each frame, we can also perform site classification depicting the activity taking place at a particular location. This can be useful for surveillance applications of sites such as roads and parking lots, in which it is desirable to tell where in the image there is activity, or when that activity occurred. One such model can be obtained by simply keeping track of the amount of activity at each pixel in the panoramic mosaic of a scene. In the next section we show a gray image in which the individual pixels store the number of times that that a pixel of the mosaic registered a foreground object. Thus, areas in which more moving objects have been deteted will show up as brighter, and areas in which no moving objects were deteted will show up as black.

7 Results

In this section we present some of the results of our algorithm. The algorithm was found to work quite well over a range of scenarios including aerial video and ground-based surveillance of a region using a panning camera, for scenes containing multiple moving objects and a mix of roads, parking areas and walkways. Lighting changes were generally correctly detected and the algorithm was able to recover reasonably from these changes, yielding accurate models of the background within a few frames after the lighting change.

Figures 1 and 2 each show two frames from two video sequences (note these frames are not adjacent in

Figure 4: Figure where each individual pixel stores the number of times a foreground object was detected at that pixel. Note how the road is clearly distinguished from other areas due to objects moving on the road.

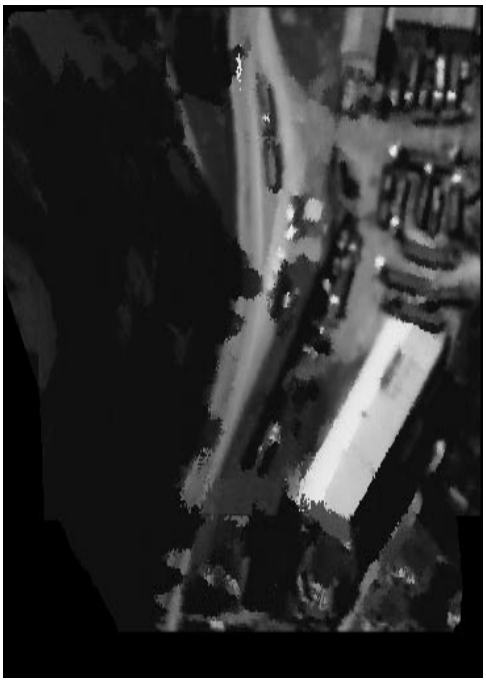

Figure 5: The background mosaic obtained when we register each frame with the mosaic where the mosaic is formed by taking the last pixel registered at that location.

time). The first row shows the original frames, the second row shows the synthesized frames in which all the foreground objects have been removed, and the third row shows the corresponding moving objects that were detected. The moving objects have clean boundaries and the entire motion area is detected, in contrast with methods based on the differene between registered frames, which detect just the part of the image that is different between frames. Note that in Figure 1 two very small objects are successfully detected. In the first column, the small object to the left of the car corresponds to a person walking on the sidewalk. In the second column, the small object on the very right is a car entering behind the trees. The synthesized frames are also quite clean, not containing evidence of the moving objects or parts of them.

Figure 3 shows the background mosaic obtained for the sequence from Figure 2. Note that at the upper left there is some evidence of the bus in this background mosaic, because that area was not observed for many frames without the bus present. Otherwise, however, there is no evidence of the moving objects in the mosaic. Figure 4 shows the corresponding "activity image" indicating the number of times that each pixel registered a moving object. The road is clearly visible in this image as a region of high activity (with the shadows cast by trees also visible by the fact that they break up a "stream" of activity).

In Figure 5 we present the background image obtained by registering the current frame with a more traditional image mosaic rather than the background image. This mosaic is formed by taking the intensity value of the last registered pixel at a given location, rather than using the mixture models to form a background image. This is a common technique, but as is visible from this mosaic from the first 75 frames of the sequence, the registration is not very robust. After 100 or so frames, the registraion is totally off. This example illustrates the utility of the background image for accurate image registration in the presence of moving objects.

Finally, in Figure 6, we present a background mosaic of a region using 1000 frames. The mosaic is quite clear, indicating accurate and stable alignment over 1000 frames in the presence of many moving objects in the form of moving people on the pathways. There is no evidence of these moving objects in the mosaic, which illustrates the usefulness of the algorithm in constructing panoramic views that contain just the non-moving objects in the scene.

Figure 6: The background mosaic from a sequence of 1000 frames captured from a surveillance camera moving in both vertical and horizontal directions

8 Summary and Conclusions

In this paper, we have presented an algorithm for wide area surveillance and monitoring. The method uses a mixture of Gaussian model to represent pixels in the scene. From these mixture models, a model of the background is construted using the mean of the highest weight Gaussian at each pixel. The background model provides a means of registering video frames that is robust in the presence of moving objects in the scene. The models can also be used to detect moving objects in a moving camera scene, and to create panoramic views and video sequences that do not contain any moving objects.

Acknowledgments

This research is supported by a grant from DARPA under contract DAAL01-97-K-0104.

References

[1] Chris Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. *CVPR99* Fort Collins, CO (June 1999).

[2] Christof Ridder, Olaf Munkelt, and harald kirchner. Adaptive background estimation and foreground detection using Kalman-Filtering. *Proceedings of the International Conference on recent Advances in Mechatronics*, ICRAM '95, UNESCO Chair on Mechatronics, 193-199, 1995.

[3] Nir Friedman and Stuart Russell. Image Segmentation in Video Sequences: A Probabilistic Approach. In *Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence(UAI)*, Aug. 1-3, 1997.

[4] W.E.L. Grimson, Chris Stauffer, Raquel Romano, and Lily Lee. Using adaptive tracking to classify and monitor activities in a site. In *CVPR1998*, Santa Barbara, CA. June 1998.

[5] J.R. Bergen, P. Anandan, K.J. Hanna and R. Hingorani. Hierarchical Model-Based Motion Estimation. *Proceedings ECCV-92*, Springer-Verlag, Italy, May 1992.

[6] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39 (Series B):1-38, 1977.

[7] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Unpublished manuscript, 1993.

[8] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *IJCV*, 2(1):51-76, June 1988.

[9] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Image Understanding Workshop*, pages 121-130, 1981.

[10] Jianbo Shi and Carlo Tomasi. Good Features to Track. *CVPR94*, pages 593-600.

[11] M. Irani, P. Anandan and S. Hsu. Mosiac based representations of video sequences and their applications. *Proceedings ICCV*, pages 605-611, 1995.

[12] H.S. Sawhney, S. Ayer and M. Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation, *Proceedings ICCV*, pages 583-590, 1995.

[13] M. Irani and P. Anandan. Video indexing based on mosaic representations. *Proceedings IEEE*, pages 905-921, May 1998.

[14] R. Szeliski. Image mosaicing for tele-reality applications. Technical Report CRL94/2, DEC-CRL, May 1994.