1-1-2007

Detection and tracking of multiple moving objects in video.

Wei Huang
University of Windsor

DETECTION AND TRACKING OF MULTIPLE MOVING OBJECTS IN VIDEO

By

Wei Huang

A Thesis
Submitted to the Faculty of Graduate Studies through the
Department of Electrical and Computer Engineering in Partial Fulfillment of the
Requirements for the Degree of Master of Applied Science at the

University of Windsor

Windsor, Ontario, Canada

2007

Your file Votre référence
ISBN: 978-0-494-35050-8
Our file Notre référence
ISBN: 978-0-494-35050-8

ABSTRACT

In this thesis, we propose a new algorithm for detecting and tracking multiple moving objects in both outdoor and indoor environments. The proposed method measures the change of a combined color-texture feature vector in each image block to detect moving objects. The texture feature is extracted from DCT frequency domain. An attributed relational graph (ARG) is used to represent each object, in which vertices are associated to an object's sub-regions and edges represent spatial relations among the sub-regions. Multiple cues including color, texture, and spatial position are integrated to describe each object's sub-regions. Object tracking and identification are accomplished by graph matching. The inexact graph matching technique enables us to track partially occluded objects and to cope with object articulation. An ARG adaptation scheme is incorporated into the system to handle the changes in object scale and appearance. The experimental results prove the efficiency of the proposed method.

ACKNOWLEDGEMENTS

I would like to give my sincere thanks to my supervisor, Dr. Jonathan Wu for his guidance and great help which led to the successful completion of this thesis. I am also very grateful to Dr. Maher Sid-Ahmed for his insight and valuable suggestions. I wish to thank Dr. Xiaobu Yuan for his comments on this work. I would like to thank Ms. Andria Turner for her kind assistance during my study period.

I would like to thank my parents for their support, understanding and love over the years.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The efficient detection and tracking of moving objects is currently one of the most active research topics within the areas of computer vision and digital video processing. It has many applications in:

- Video surveillance
- Video indexing and retrieval
- Human-computer interaction
- Video communication
- Traffic control
- Vehicle navigation

There are two key steps in object tracking: detection of interesting moving objects and tracking of such objects from one frame to another in an image sequence. Motion detection aims at segmenting regions corresponding to moving objects from the rest of an image. Tracking moving objects over time typically involves matching objects in consecutive frames.

The major challenges encountered in object tracking are:

- Natural cluttered and dynamic background
- Illumination changes
- Cast shadows
- Partial and full occlusion
- Nonrigid or articulated nature of objects

✧ Changes in object scale and appearance

The complexity of the problem is increased if multiple moving objects need to be tracked, especially when objects have similar features. The goal of multiple objects tracking system is to track a variable number of objects and maintain the correct identities of the objects in successive video frames.

1.2 Research Objectives and Contributions

The objective of the thesis is to develop a novel approach to efficient detection and tracking of multiple moving objects in real-world scenarios, including both outdoor and indoor environments, using a single stationary video camera. The algorithm should be robust to partial static occlusion, clutter, cast shadows, and changes in object scale and appearance.

The main contributions of the thesis are: (1) A new motion detection method is introduced which measures the change of the combined color-texture feature vector in each image block within a time window and then directly obtains moving objects by statistically analyzing the change. The proposed motion detection algorithm has good performance in busy environments where a clean background is unavailable. (2) An attributed relational graph (ARG) is used to represent each moving object. The structure of the given object is modeled by the ARG in which the semantic information in vertices specifies the local properties, while edges represent spatial relations among them. Through ARG, the system has more power to distinguish different objects than many other systems that use global features, e.g. color histograms. Multiple cues including color, texture, and spatial position are combined to describe each object's sub-regions, which are associated to the vertices of the ARG. Inexact graph matching enables us to track partially occluded objects and to deal with object articulation. (3) An ARG adaptation scheme which associates a probability mask with every vertex and edge in the graph is incorporated into the system to handle the changes in object scale and appearance.

1.3 Thesis Overview

The thesis is organized in six chapters.

Chapter 1 presents an introduction to the multiple moving objects detection and tracking problem. The research objectives and contributions of the thesis are also provided.

Chapter 2 gives a brief review of the existing algorithms on both motion detection and object tracking, which are the two sub-problems addressed by the thesis.

Chapter 3 introduces the proposed motion detection algorithm. The combined color-texture feature vector is calculated in section 3.1. Then we explain the details of detecting moving objects using eigenspace decomposition and statistical analysis in section 3.2. The connected component analysis algorithm is provided in section 3.3. Cast shadow removal is discussed in section 3.4.

Chapter 4 describes the details of the proposed object tracking and identification algorithm. Section 4.1 describes how to construct the attributed relational graph to represent the detected object. Section 4.2 presents the scheme for attributed relational graph adaptation. Section 4.3 gives the details of identifying objects using inexact graph matching technique.

Chapter 5 shows the experimental results for real image sequences. Result comparison and analysis are also provided in this chapter.

Chapter 6 gives conclusions and future research work.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Related Work on Motion Detection

As for motion detection, the background subtraction technique which compares each new frame to the background image is a popular method. Any significant change between the incoming frame and the background model signifies a moving object. One of the simplest background models is a time-averaged image of the scene. This method requires the background to be visible during a significant portion of the time. It can not cope well with the scene in which many objects move slowly. Background subtraction using media filter was used in [1]. The background model was defined as the median of all the frames in the buffer at each pixel. Use of the median is based on the assumption that the background will be visible for more than fifty percent of the frames in the buffer. Compared with using the mean, the advantage of the median is that it avoids blending pixel values. Pfinder [2] used a single Gaussian distribution $N(\mu, \Sigma)$ to model the color at each pixel. The mean μ and covariance Σ of pixel values were recursively updated by an adaptive filter to adapt to slow changes in the scene. In outdoor environments with moving tree branches and bushes, different colors can be observed at a certain pixel, so the single Gaussian assumption will not hold. In [3], the pixel value was modeled by a mixture of weighted K (K is a small number from 3 to 5) Gaussian distributions to support multiple backgrounds. The weight was related to the frequency with which each Gaussian explains the background. This Gaussian mixture model is the underlying model for many background subtraction techniques [4][5][6]. However, this method has the problem of slow learning at the beginning, especially in such cases that a training period absent of foreground objects is not available. The Gaussian assumption does not always hold. To deal with the limitation of parametric methods, [7][8] used a nonparametric kernel density model by estimating the probability of pixel intensity directly from a set of most recent intensity values. Model adaptation can be achieved by removing older samples and adding new samples. This approach has high computational complexity as it needs to compute the intensity function for each pixel. In [9], eigenspace decomposition was used to detect objects. The background was represented by the first M significant eigenvectors. The current image was reconstructed by the M eigenvectors and the foreground pixels were detected by getting the difference between the actual image and its reconstruction. A training period was required to construct the background model. This approach is less sensitive to illumination. A foreground pixel may be absorbed into the background due to its intensity falling within the range of feature space. In [10], optical flow was calculated in the neighbourhood surrounding each pixel to evaluate the hypotheses of the background value at that pixel. The net flow will have large values in regions where moving objects are approaching or leaving and values near zero in regions where no motion is occurring. This method can handle busy environments, but requires the background to be approximately stationary. No reports were supplied on the success of this algorithm in outdoor scenes. [11] used a probabilistic background model based on Hidden Markov Model (HMM) to classify every pixel to be foreground, background, or shadow region. The different hidden states allow using a training sequence with foreground objects present in the scene to estimate the model parameters.

Besides pixel intensity, edge features have also been used for background modelling. The edge of images is less sensitive to illumination changes. In [1], background subtraction was performed in both color and edge density. A pixel was flagged as foreground if both the color and edge density support that classification. [12] classified pixels in the color and gradient domain separately at the pixel level, and then integrated the color and edge information at the region level. They removed spurious objects by claiming that the actual foreground object would have high values of gradient based background difference at its boundaries. In [6], a Bayesian network was introduced to describe the relationships among the background, intensity, and edge information.

2.2 Related Work on Object Tracking

The tasks of object detection and tracking can be performed jointly or separately. In the first case, objects are detected and tracked jointly by updating object information, such as position and velocity, obtained from previous frames. Then tracking becomes a *state estimation* problem. Kalman filters and particle filters are widely used [13][14][15][16]. The Kalman filter is only suitable for estimating the state of a linear system where the state variable is distributed by a Gaussian. The predicted position based on linear trajectory and constant velocity will fail to track the object when the object has a sudden movement and changes its velocity abruptly. The particle filtering technique has been proven to be a powerful tool for nonlinear and non-Gaussian systems. However, the large number of samples required to represent the posterior distribution prevents its use in high dimensional state space. Data association is required when tracking multiple objects using Kalman or particle filters. An incorrectly associated measurement can cause the filter to fail.

In the later case, objects are detected in each incoming frame, and then consistent labels are assigned to the detected objects across frames. This approach results in tracking as a *matching* problem. The similarity function which defines a distance between object model and candidates is calculated during the matching process. Using distinctive features to represent the object being tracked has great importance in this class of approach. The most widely used cues in object tracking are appearance, shape, spatial position, and motion. Color distributions have been effectively used for appearance modeling using both color histograms [15][17] and Gaussian mixture models [18][19]. [17] used a weighted color histogram to represent the object. The mean shift procedure was used to perform the maximization of the similarity measure. In [18] color distributions were modeled in the hue-saturation space using adaptive Gaussian mixture models. While the parameters of the model were updated over time, the number of components was fixed. [19] modeled color features at each pixel by a Gaussian mixture, where the number of Gaussians was automatically determined by a mean-shift algorithm. [20][21] utilized a color correlogram as the main model to represent object appearance. The color correlogram included the spatial correlation information of pairs of colors. [22] used an active shape model to track people. Person outline shapes were approximated by cubic B-splines. Principal component analysis (PCA) was applied to generate a subspace to explain the most significant shape modes. A training stage was needed in this method. In [23], an online shape model based on level set representation was used to model the object shape and its changes.

In cluttered environments, a single cue approach may lead to poor tracking performance. One solution is to combine two or more cues. [24] identified moving people by their color and spatial information. Discrete wavelet transform was used to remove fake motions. [25] integrated color and texture for object tracking by estimating their joint probability distribution function. Color was modeled by a 2D Gaussian distribution and texture was modeled by an autobinomial Gibbs Markov random field. [14] demonstrated that the combined color and texture cues provided a good tracking result that was more accurate than the two cues individually. [26] also showed that tracking with multiple weighted cues provided more reliable results. A framework for combining color, texture and edge cues was used for robust tracking. In [5], five significant features were used, including velocity, size, elliptic-fit aspect ratio, orientation, and dominant color. In [1], a hierarchical matching process was performed in the order of spatial position, shape and color.

Occlusion is a challenging problem when tracking objects in the real world. The occlusion can happen in two types: dynamic occlusions and static occlusions. The dynamic occlusion happens when objects in a group occlude each other. The static occlusion occurs when objects are occluded by the scene structures such as trees, buildings, or streetlights. In [27], depth ordering and position during occlusion were estimated by a visibility index which indicated the degree of occlusion of each object. [28] used the color distribution of the occluded person's clothing to recover that person after occlusion ended. In [29], a prior scene model was constructed, specifying the static occlusions in the scene by their location and dimension. An occlusion was inferred when the static occlusions overlapped the predicted bounding box of an object. [30] obtained depth ordering during occlusion from 3D information.

CHAPTER 3

DETECTION OF MOVING OBJECTS

3.1 Color-Texture Feature Approach

While color was widely used as a feature in background subtraction, texture information was neglected by many existing algorithms. [31] introduced an idea that the texture vectors are very likely to have a large spread when a moving object is passing through a fixed position. Although the texture vectors will not be constant when we observe a position corresponding to part of the background, its spread will be usually small. Motivated by this idea, we measure the change of a combined color-texture feature vector to detect moving objects. Combining color and texture as the feature vector can still extract foreground objects when the color distributions of the foreground and background are similar, in which case the Gaussian mixture model will fail. There are many approaches in texture feature extraction. [14][32] used wavelet coefficients to represent texture. In [33], Gabor filters were used to obtain object textures. [25] used Gibbs random fields (GRF) to model the texture information. Some researchers proposed to use discrete cosine transform (DCT) for texture representation [34]. The DCT transforms an image from the spatial domain to the frequency domain (Fig. 1). It separates the image into spectral sub-bands of differing importance.

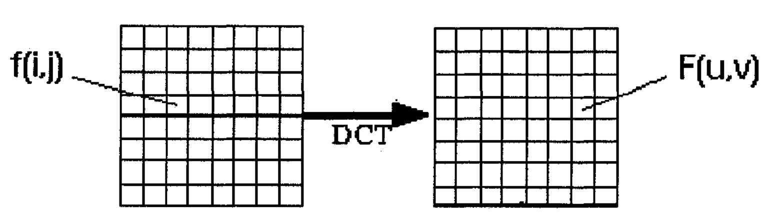


Fig. 1. 2D DCT applied to an 8 * 8 image block.

The 2D (N by M image) DCT can be written in terms of pixel values $f(i,j)$ and the frequency-domain coefficients $F(u,v)$:

$$F(u,v)=\left(\frac{2}{N}\right)^{\frac{1}{2}}\left(\frac{2}{M}\right)^{\frac{1}{2}}.A(u).A(v).\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\cos\left[\frac{(2i+1)u\pi}{2N}\right].\cos\left[\frac{(2j+1)v\pi}{2M}\right].f(i,j) \quad (1)$$

The corresponding inverse 2D DCT transform is:

$$f(i,j)=\left(\frac{2}{N}\right)^{\frac{1}{2}}\left(\frac{2}{M}\right)^{\frac{1}{2}}.\sum_{u=0}^{N-1}\sum_{v=0}^{M-1}A(u).A(v).\cos\left[\frac{(2i+1)u\pi}{2N}\right].\cos\left[\frac{(2j+1)v\pi}{2M}\right].F(u,v) \quad (2)$$

Where

$$A(y)=\begin{cases}\frac{1}{\sqrt{2}} & \text{for } y=0\\ 1 & \text{otherwise}\end{cases} \quad (3)$$

The frequency domain block contains one DC coefficient and other AC coefficients. The DC coefficient represents the average intensity of the block and the other AC coefficients represent some different pattern of image variation. For example, the coefficients of the most left region and those of the most upper region in a DCT transform domain represent some horizontal and vertical edge information, respectively.

We use some DCT coefficients as the texture feature. We partition every new frame into blocks with 8 * 8 pixels, where every two neighbouring blocks overlap each other by four pixels horizontally or vertically for improving the spatial resolutions of the detection results. A feature vector is extracted for each block. Eleven features are used for detection. Two of them are the average color components (A_{Cb}, A_{Cr}) in an 8 * 8 block. The other nine features are the AC coefficients in DCT transform domain as shown in Fig. 2. We use the well-known YC_bC_r color space, where Y encodes luminance, C_b and C_r encode color information (chrominance). To obtain the other nine features, the DCT is applied to the Y component of the image block. One color-texture feature vector for a block u is then expressed as:

$$\mathrm{f}_{u,t} = \left(A_{Cb}, A_{Cr}, A_{1,2}, A_{2,1}, A_{1,3}, A_{2,2}, A_{3,1}, A_{1,4}, A_{2,3}, A_{3,2}, A_{4,1}\right)^T \tag{4}$$

	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$				
$A_{2,1}$	$A_{2,2}$	$A_{2,3}$					
$A_{3,1}$	$A_{3,2}$						
$A_{4,1}$							

Fig. 2. DCT AC coefficients used as the texture feature.

3.2 Measuring the Change of the Color-Texture Feature Vector

By measuring the change of the color-texture feature vector over time, we are able to detect whether a particular block belongs to a background or to a moving object. If during a time interval there is no moving object in the block, the color-texture feature vectors will be close to each other. Thus the change of the feature vectors within the time window will be small. In contrast, if a moving object is passing through this block, the feature vectors will alter fast. Hence the change of the feature vectors during the time interval will be fairly large. For measuring the change of the color-texture feature vector, we compute the covariance matrix of the feature vectors in the same block location within a small number of consecutive frames. The eigenvalues of the covariance matrix refer to the variance of the data in the direction of the basis vectors. We use the largest eigenvalue as a local change measure. The larger the largest eigenvalue, the more likely is the presence of a moving object.

In practice, for each block u, we consider the color-texture feature vectors $\mathbf{f}_{u,\tau-S}, \mathbf{f}_{u,\tau-S+1}, \ldots, \mathbf{f}_{u,\tau}, \ldots, \mathbf{f}_{u,\tau+S-1}, \mathbf{f}_{u,\tau+S}$ for a symmetric window with size of $2S+1$ around the temporal instant τ. For these vectors, the covariance matrix $R_{u,\tau}$ is:

$$R_{u,\tau} = \frac{1}{2S+1} \sum_{t=\tau-S}^{\tau+S} \left(f_{u,t} - \bar{f}_u \right)\left(f_{u,t} - \bar{f}_u \right)^T \tag{5}$$

Then, the covariance matrix $R_{u,\tau}$ is decomposed into its eigenvectors $e_{u,\tau}(k)$ and eigenvalues $\lambda_{u,\tau}(k)$ $(k = 1,2,\ldots,11)$.

$$R_{u,\tau} . e_{u,\tau}(k) = \lambda_{u,\tau}(k) . e_{u,\tau}(k) \tag{6}$$

The largest eigenvalue $\lambda_{u,\tau}^{m}$ is the local change measure $C_{u,\tau}$.

$$C_{u,\tau} = \lambda_{u,\tau}^{m} \tag{7}$$

Finally, we mark each block as part of a moving object or background according as whether the change measure is larger than a threshold or not. We assume that the values of the local change measure $C_{u,\tau}$ in every video frame obey a Gaussian distribution. We compute the mean μ_τ and variance σ_τ^2 of all $C_{u,\tau}$ for $u = 1,2,\ldots,L$. L

is the total number of sub blocks of every video frame. A block will be labelled as moving if

$$\frac{\left(C_{u,\tau} - \mu_\tau\right)^2}{\sigma_\tau^2} > th1 \tag{8}$$

$$\mu_\tau = \frac{1}{L}\sum_{u=1}^{L} C_{u,\tau} \tag{9}$$

$$\sigma_\tau^2 = \frac{1}{L}\sum_{u=1}^{L}\left(C_{u,\tau} - \mu_\tau\right)^2 \tag{10}$$

Where $th1$ is a threshold. When the value of the pixel inside a moving object is zero, the moving object may be split into two different objects. We apply morphological operations on the binary image to restore some missing pixels of the moving object.

3.3 Connected Component Analysis [24][35]

The pixels belonging to an object are connected. A connected component analysis algorithm is used to find connected regions corresponding to the objects in the binary images that we obtained at the motion detection stage. We scan the binary image pixel by pixel, from left to right and from top to bottom. Let q denote the pixel at any step in the scanning process and we only need to consider the neighbors at directions 1, 2, 3 and 4 as shown in Fig. 3. To label 8-connected components, we use the following procedure. The nature of the scanning sequence ensures that these neighbors have already been processed by the time the procedure gets to q. If q is 0, move on to the next scanning position. If q is 1 and all four neighbors are 0, assign a new label to q. If only one of the neighbors is 1, assign its label to q. If two or more neighbors are 1, assign one of the labels to

q and make a note of the appropriate equivalences. After completing the scan of the image, we sort the equivalence label pairs into equivalence classes, assign a unique label to each class, and do a second scan through the image, replacing each label by the label assigned to its equivalence class. We use a size filter to remove the connected component whose area is below a threshold.

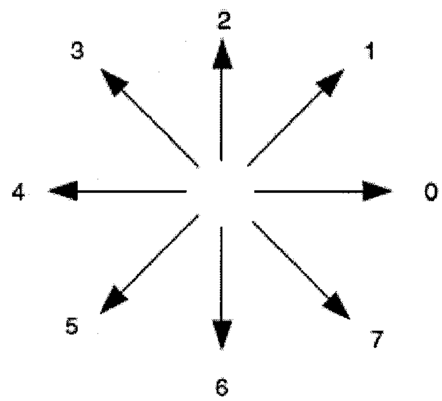


Fig. 3. Eight neighbors of pixel q.

3.4 Removing Cast Shadows and Adaptation to Varying Illumination

A shadow occurs when an object partially or totally occludes direct light from a source of illumination. A cast shadow is the area projected by the object in the direction of direct light. Moving cast shadows cause a frame difference between two consecutive images of video sequence. In this case, shadows cast by moving objects are often detected as a part of the moving objects because shadows move along with the movement of objects. When the detected objects contain shadows, the estimation of objects' locations and analysis of their shapes may go wrong. For this reason, the problem of shadow detection has been increasingly addressed in the last decade [36][37][38]. Two properties of cast shadow are used in our color-texture based motion detection method: a)

The chrominance of the cast shadow is identical or slightly shifted when compared with background. b) The same regions with or without cast shadows tend to have similar texture properties despite the illumination difference. Color information is useful for suppressing shadows from detection by separating color information from lightness information. The chrominance used in detection is more insensitive to small changes in illumination that are due to shadows. Combining color with texture as the feature vector and measuring the change of this feature vector for detecting moving objects can effectively remove shadows from moving objects.

In outdoor environments, the illumination conditions gradually change. The motion detection method should be adaptive to the lighting variations that inevitably occur. Texture is a feature that is not easily affected by varying illumination. In addition, varying lighting conditions do have effects on the intensity, but not on the two color information C_b and C_r. Therefore, the color-texture combined motion detection method is less sensitive to illumination changes.

CHAPTER 4

MULTIPLE MOVING OBJECTS TRACKING

4.1 Object Representation by Attributed Relational Graph

The success or failure of multiple objects tracking highly depends on how distinctive the object model is. Currently, color histograms are widely used to represent the detected object. The color histogram is robust against noise and orientation variance, but suffers from illumination changes or the presence of the confusing colors in background. More important, color histograms have limited discriminative power. The color features are represented globally and are not spatially localized. Therefore color histograms lose the spatial information about the color distributions. Two images producing identical color histograms may have totally different spatial organization of colors.

Attributed relational graph (ARG) is an essential data structure for representing relational information. It has been widely used in the areas of pattern recognition and computer vision [39][40][41][42][43]. In the thesis, we use the attributed relational graph to represent each object for tracking, in which vertices are associated to an object's sub-regions and edges represent spatial relations among the sub-regions. Through ARG, the system is capable of exploiting local rather than global information. This enables the system for instance to track a person with a red shirt and green pants in presence of a person wearing a green shirt and red pants, or vice versa. This would be impossible for many other systems that are evaluating global features, e.g. color histograms. Fig. 4 shows such a case.

Fig. 4. The two different images above have the same color histogram. We can distinguish them by using ARG.

To fragment an object into sub-regions, we first use a combined color-texture-position feature vector to describe the detected image blocks which belong to the object. The two color features are the average color components (A_{Cb}, A_{Cr}) in an 8*8 image block. The one texture feature is the square root of the average summation of the first nine squared AC coefficients along the zigzag scanning. The two position features are simply the coordinates of the image block. Including position features generally leads to smoother sub-regions. After obtaining the feature vectors for all the blocks, we perform normalization on the five features to eliminate the effects of different feature ranges. Then the k-means algorithm is used to cluster the feature vectors into several classes with every class in the feature space corresponding to one spatial sub-region of the detected object. The k-means algorithm does not specify the value of k. To compute the optimal value of k, we iterate it between a minimum $(k_{\min} = 2)$ and a maximum value $(k_{\max} = 5)$ until a stop constraint is satisfied. The maximum cluster number of 5 is sufficient in our experiments. The method of determining the number of clusters automatically may be added into this part in the future [44]. The mean-shift clustering procedure as a nonparametric density estimation technique can also be considered [45].

After the segmentation, we are ready to build the ARG for each detected object. An ARG is a graph in which attribute vectors are assigned to vertices and to edges. Formally, we define an ARG as $G = (N, E, \mu, \nu)$, where N represents the set of vertices of G and $E \subseteq N \times N$ the set of edges. Two vertices a, b of N are said to be adjacent if $(a,b) \in E$. Furthermore, $\mu : N \rightarrow L_N$ assigns an attribute vector to each vertex of G, while $\nu : E \rightarrow L_E$ assigns an attribute vector to each edge in G.

The structure of an object can be represented as a collection of sub-regions which are related by their relative positions within the object. The sub-regions are represented by vertices in a graph, while relations between them are represented by edges. Let us consider any two vertices a, b in N. The vertex attribute $\mu(a)$ is defined as follows:

$$\mu(a) = \left(C_{Cb}, C_{Cr}, T_{AC}, P_x, P_y\right)^T \tag{11}$$

The five terms correspond to the color component C_{Cb}, color component C_{Cr}, texture T_{AC}, spatial coordinate P_x and spatial coordinate P_y at the centroid location of a cluster, respectively. Each cluster obtained by k-means algorithm corresponds to a sub-region within the object.

The edge attribute $\nu(a,b)$, for a, b in E, is defined as the length value of the edge linking the two vertices a and b.

In practice, the object is represented by a signature, which is composed of two parts: the first one is the feature vectors of all sub-regions, which are called vertex attributes, and the second one is a representation of the topology of the sub-regions within the object. Spatial relationships between sub-regions are characterized by an adjacency matrix of sub-regions with a value of 1 if both sub-regions have at least one pixel in common, otherwise 0. For the pair of adjacent sub-regions, the length value of the corresponding edge is stored in a distance matrix.

In our approach, the ARG is extracted automatically; no prior model needs to be specified for an object. Thus the ARG can be generalized for modeling any object.

4.2 Attributed Relational Graph Adaptation

The tracking and identification of objects amounts to graph matching. Both input and model graphs are automatically extracted from video sequences. Usually, the graphs extracted in the first frame act as the model graphs. The appearance and shape of the object do not remain the same through the entire video, so the model graphs need to adapt. Once a track is created, a model graph of the object is initialized. This model graph is adapted every time the same object is tracked into the next frame. The model graph structure is altered by 1) adding new vertices and edges to the graph. 2) removing long-term unmatched vertices and edges in the model graph. 3) updating the matched vertices and edges by blending in new observations. We realize model graph adaptation by associating a probability mask with every vertex and edge in the graph. The probability mask records the likelihood of the vertices and edges being observed in the model graph. The model graph and probability mask are initialized at the first frame. The probabilities of the vertices and edges are initialized to1 at the first frame to add relatively great weight to the first model graph. On subsequent frames, the model graph and probability mask values are updated using the following formulae:

$$\mu(a,t)=\begin{cases}\alpha\,\mu(a,t-1)+(1-\alpha)\mu(a') & \text{if } \mu(a,t-1) \text{ is matched with } \mu(a') \\ \mu(a,t-1) & \text{if } \mu(a,t-1) \text{ does not have a match}\end{cases} \tag{12}$$

$$P_\mu(a,t)=\begin{cases}\alpha\,P_\mu(a,t-1)+(1-\alpha) & \text{if } \mu(a,t-1) \text{ has a match} \\ \alpha\,P_\mu(a,t-1) & \text{if } \mu(a,t-1) \text{ does not have a match}\end{cases} \tag{13}$$

$$P_\nu(a,b,t)=\begin{cases}\alpha\,P_\nu(a,b,t-1)+(1-\alpha) & \text{if } \nu(a,b,t-1) \text{ has a match} \\ \alpha\,P_\nu(a,b,t-1) & \text{if } \nu(a,b,t-1) \text{ does not have a match}\end{cases} \tag{14}$$

Where $\mu(a)$ and $\nu(a,b)$ are the vertex and edge attribute in the model graph, respectively. $\mu(a')$ is the vertex attribute in the input graph. P_μ and P_ν are the vertex and edge probability, respectively. α is an updating factor. We recalculate the edge attribute after the vertex attribute is updated. Newly observed vertices and edges are added to the model graph with probability initialized to 0.3. A new edge is added to an ARG during the adaptation process as shown in Fig. 5.

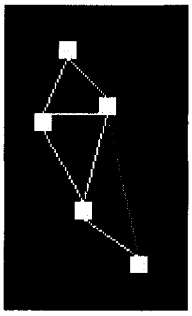


Fig. 5. Edge addition. The new edge, indicated by the darker line, is added to the model ARG.

4.3 Inexact Graph Matching for Object Tracking and Identification

When graphs are used to represent objects, the problem of objects tracking and identification can be seen as a problem of graph matching. The graph does not remain the same in different frames due to the nonrigid nature of the object. In real-world environments, objects may also be occluded by other moving objects or trees and buildings in the scene. So, the comparison among the input graph and model graph cannot be performed by exact graph matching procedures. The notion of inexact graph matching enables us to track partially occluded objects and deal with object articulation. Two matched graphs do not have to be identical but only similar in terms of vertex number, vertex attributes or edge number. Our implementation of the matching algorithm is given below:

In the following, a , b refer to vertices in the model graph M , and $a^{'}$, $b^{'}$ correspond to vertices in the input graph I .

1) For each vertex $a^{'}$ in the input graph, a search is conducted to find the best matching vertex a in the model graph, such that the Euclidian distance of the matching vertex attributes $d(\mu(a), \mu(a^{'}))$ is the minimum value. The vertex a in the model graph is matched with the vertex $a^{'}$ in the input graph if this minimum Euclidian distance is lower than a threshold. Otherwise, the vertex a in the model graph does not have a match in the input graph. The vertex similarity for this pair of vertices is computed as:

$$S^{\mu}{}_{aa^{'}} = e^{-\left|d\left(\mu(a), \mu\left(a^{'}\right)\right)\right|} \tag{15}$$

We have to satisfy two basic constraints during the matching process: a) A vertex in the input graph cannot match with two different vertices in the model graph. b) Two different vertices in the input graph cannot match with a single vertex in the model graph. It is possible that some vertices in the input graph do not have matching vertices in the model graph because the two graphs may have different vertex number.

2) After the vertices are matched, total similarity is computed by taking into account the topology of the matched graphs. Let the vertices a and b match a' and b' respectively. Then the topology similarity for this pair of vertices is computed as:

$$S^{\nu}{}_{aba'b'} = e^{-\left|\nu_{ab} - \nu_{a'b'}\right|} \tag{16}$$

Where ν_{ab} and $\nu_{a'b'}$ are the length values of the edges ab and $a'b'$.

3) The total similarity for matching the input graph to the model graph is given by

$$S_1\left(I, M\right) = \frac{\beta}{N_\mu} \sum_a P_\mu\left(a\right) S^{\mu}{}_{aa'} + \frac{1-\beta}{N_\nu} \sum_a \sum_b P_\nu\left(a, b\right) S^{\nu}{}_{aba'b'} \tag{17}$$

Where N_μ is the maximum value between the total vertex number of the input graph and the sum of vertex probability in the model graph. N_ν is the maximum value between the total edge number of the input graph and the sum of edge probability in the model graph. β is a scaling parameter which controls the relative importance of the two similarity functions. It is possible that edges in the input graph do not have corresponding edges in the model graph and vice-versa.

The total similarity is then scaled to reflect the difference in the position of the input and model objects:

$$S\left(I, M\right) = p * S_1\left(I, M\right) \tag{18}$$

$$p = \begin{cases} 1 & d < d_0 \\ e^{-1} & \text{otherwise} \end{cases} \qquad (19)$$

Where d is the Euclidian distance between the centroids of the input and model objects. d_0 is a constant. The motivation of adding the position scaling factor p into the similarity function is that an object will not move far from its last position. Therefore, the centroid presents us with a useful feature for tracking objects. To let this factor work properly, we update the model's position once we get an input object matched to that model.

The best candidate match M^* satisfies:

$$S(I, M^*) = \max_M S(I, M) \qquad (20)$$

When the value of $S(I, M^*)$ is more than a threshold, we say the input graph/object is identified with the model graph/object. Otherwise, we assume a new object enters the scene, it will be tracked and labelled, and the corresponding ARG of that object is constructed and stored in the model graph/object list.

CHAPTER 5

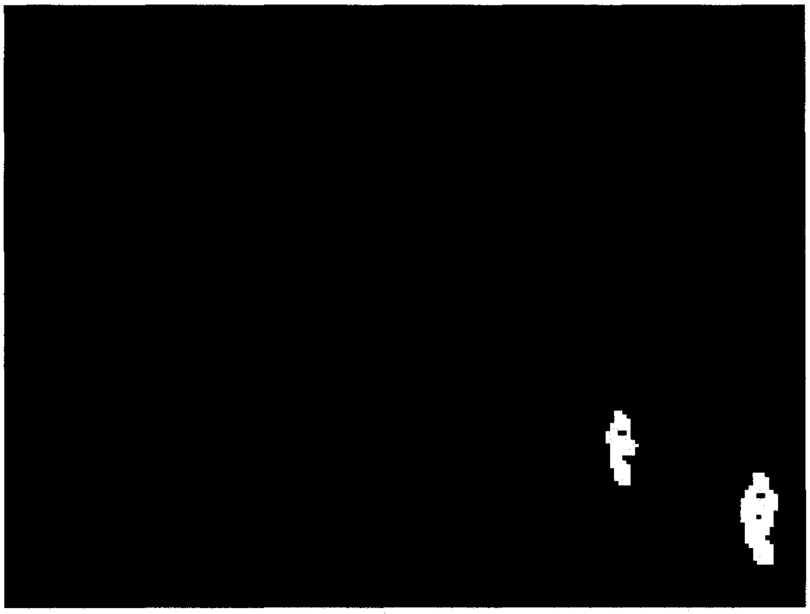
EXPERIMENTAL RESULTS AND DISCUSSION

Our proposed method is evaluated on both outdoor and indoor image sequences: PETS 2001 dataset 2 and PETS 2006 S3-T7-A dataset. Datasets include moving people and vehicles. We did experiments in several cases, such as a single object indoors, a single object in an outdoor environment, multiple objects indoors, and multiple objects outdoors with varied background. The image sizes of the PETS 2001 and PETS 2006 datasets are 768*576 and 720*576, respectively. The PETS 2001 datasets involve natural cluttered and dynamic background (swaying trees). The PETS 2006 dataset involves cast shadows.

5.1 Experimental Results of the Motion Detection Algorithm

Fig. 6 shows the result of our motion detection method. Morphological operations are used on the binary image to restore some missing pixels of a moving object.

(a)



(b)

(c)

Fig. 6. Motion detection result by the proposed motion detection algorithm. (a) An Original image from PETS 2001 dataset 2. (b) Motion detection result. (c) Motion detection result after morphological operations.

5.2 Comparison of the Motion Detection Algorithm with GMM approach

In this section, we compare our motion detection algorithm with the widely used background subtraction method: Gaussian mixture model (GMM). The GMM approach models each background pixel by a mixture of K Gaussian distributions. Different Gaussians are assumed to represent different colors. The weight parameters of the mixture represent the time portions that those colors stay in the scene. In this method, a pixel in the current frame is compared with every Gaussian in the model until a matching Gaussian is found. If a match is found, the mean and variance of the matched Gaussian is updated; otherwise the Gaussian with the least weight is replaced by a new Gaussian with mean equal to the current pixel color, an initially high variance and a small weight. All the K Gaussian distributions are ordered by their values of weight and variance. Higher-rank Gaussians have high weights and low variances. The first B Gaussian components are declared as background components. Each pixel is classified as a foreground pixel if it is more than 2.5 times the standard deviations away from all of the background components. The mixture of Gaussian technique faces the problem of slow learning at the beginning, especially in busy environments where a clean background is rare. If the first value of a given pixel is a foreground object then part of the background is occluded by moving objects. It will take several frames until the genuine background can be considered as a background. In addition, GMM can not distinguish between moving shadows and moving objects. Fig. 7-12 show some sample motion detection results obtained by GMM and our method. In the experiment, a person enters the scene at the first frame. Shadows are cast on the floor and move with the person. The GMM method has an unsatisfying detection performance because no clean images are available at the beginning. An artefact of the initial image lasts for several frames. Cast shadows are detected as part of the moving object. Better results can be seen from our algorithm.

(a)

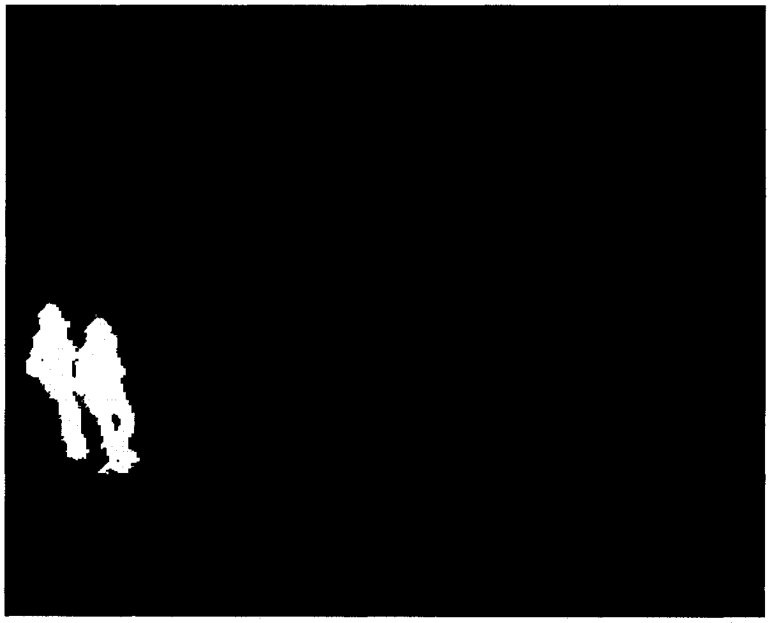
(b)

(c)

(d)

Fig. 7. (a)-(d) Original sample frames from PETS 2006 S3-T7-A dataset.

(a)



(b)

(c)

(d)

Fig. 8. (a)-(d) Motion detection results by GMM.

(a)

(b)

(c)

(d)

Fig. 9. (a)-(d) Motion detection results by our Algorithm.

Fig. 10. An original sample frame from PETS 2006 S3-T7-A dataset.

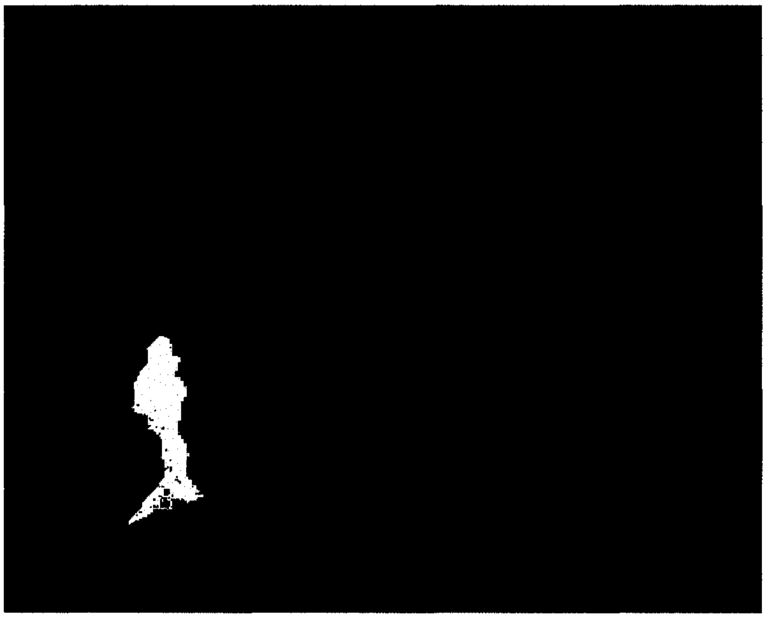


Fig. 11. Motion detection result by GMM.

Fig. 12. Motion detection result by our algorithm.

5.3 Experimental Results of the Tracking and Identification Algorithm

Fig. 13 illustrates some ARG examples for various objects, with squares representing vertices and lines representing edges. Fig. 14&15 show the evolution of two ARG models, indicating the progressive adaptation of the graph model to slow changes in appearance. The experimental results of tracking a single object indoors are shown in Fig. 16&17. Both persons enter the scene alone from the left side and leave from the lower side. One person put hands in pockets, and another pulls a luggage. Border Occlusion which is one type of static occlusions is well handled by our tracking algorithm. Fig. 18 shows the result of tracking a single car in an outdoor environment. Besides border occlusion, the scene involves another static occlusion, a streetlight. The scale of the car changes as it approaches the camera. Experimental results demonstrate the robustness of the proposed algorithm to partial static occlusion and object change in appearance and scale. Fig. 19 shows the result of tracking multiple objects indoors. At first, the man who wears blue clothes enters the scene from the lower side. His luggage enters into the view gradually. Then the man who wears black clothes enters the scene from the right side. They walk to the center of the scene. Finally, two men who merge into a single group enter the scene. Fig. 20 shows the result of tracking multiple objects in an outdoor environment with varied background. Two persons walk along the street. After a while, the third person emerges from behind the tree. In both outdoor and indoor environments, out algorithm tracks and identifies the objects correctly in successive video frames.

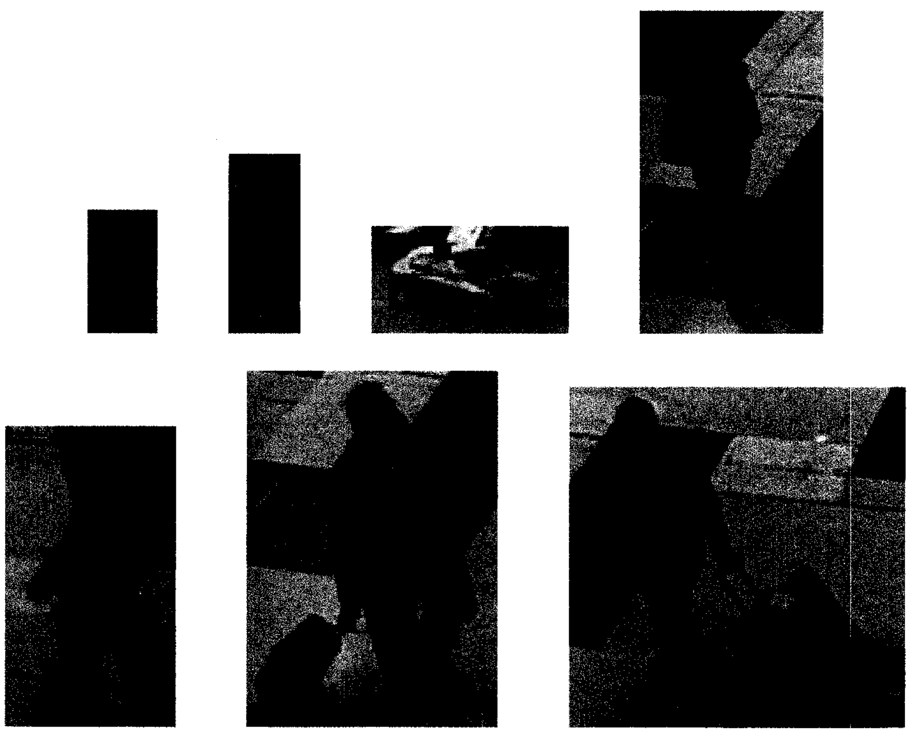


Fig. 13. The ARG is overlaid on various objects.

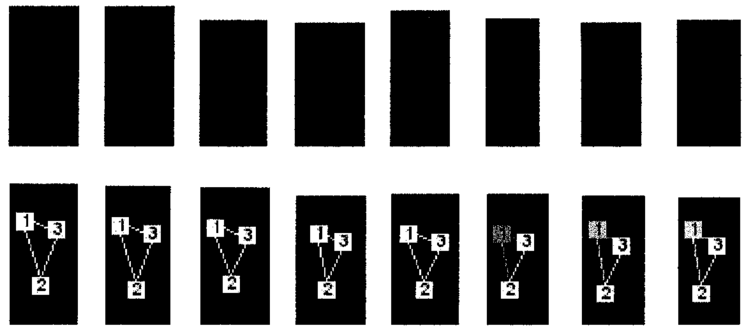


Fig. 14. The evolution of an ARG model. The upper row shows the ARG extracted from every incoming frame. The lower shows the model ARG, which is updated during the inexact graph matching process. The probability mask of vertices and edges is shows as grey levels, with white being 1. In the sixth column, the vertex 1 in the model graph does not have a match in the input graph, so its probability becomes lower, which is indicated by the darker square. The vertex 1 in the model graph gets matched in the following two images, making the square 1 become lighter again.

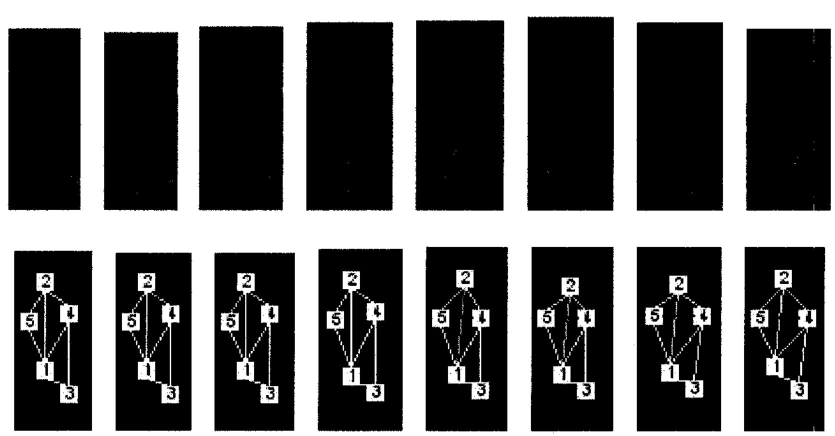


Fig. 15. The evolution of another ARG model. In the fifth column, the edge linking vertex 4 and 5 appears first in the input graph. It is initialized with the probability of 0.3 in the model graph. This edge disappears in the following three input images, so its grey level in the model graph becomes lower. In contrast, the edge linking vertex 1 and 2

disappears in the input graph in the fifth column. So, its probability in the model graph becomes lower, which is indicated by the darkening line. This edge in the model graph gets matched in the following three images, making the line become lighter again.

(a)

(b)

(c)

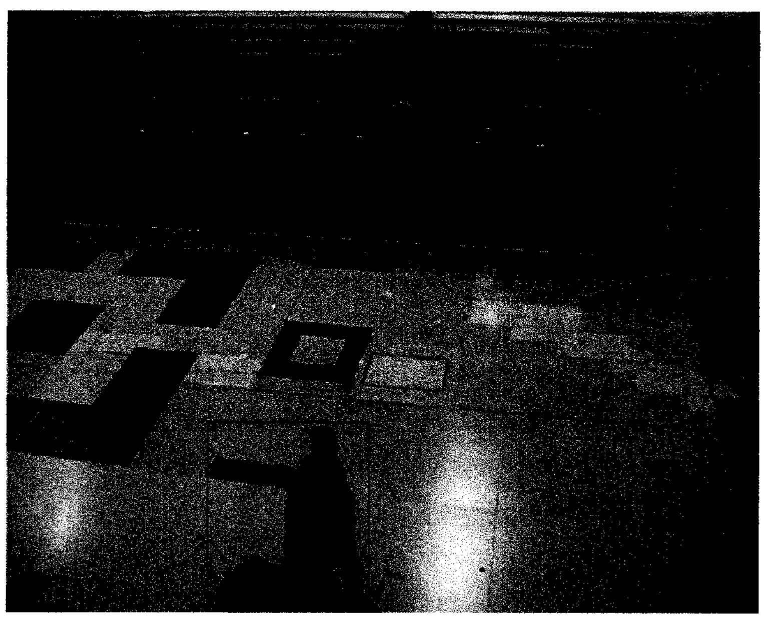
(d)

(e)

Fig. 16. (a)-(e) Tracking a single person indoors.

(a)

(b)

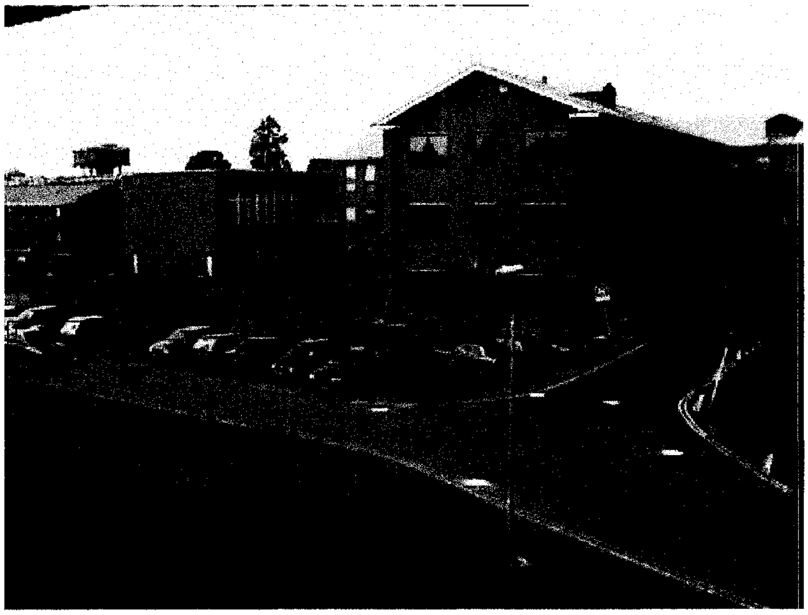


(c)

(d)

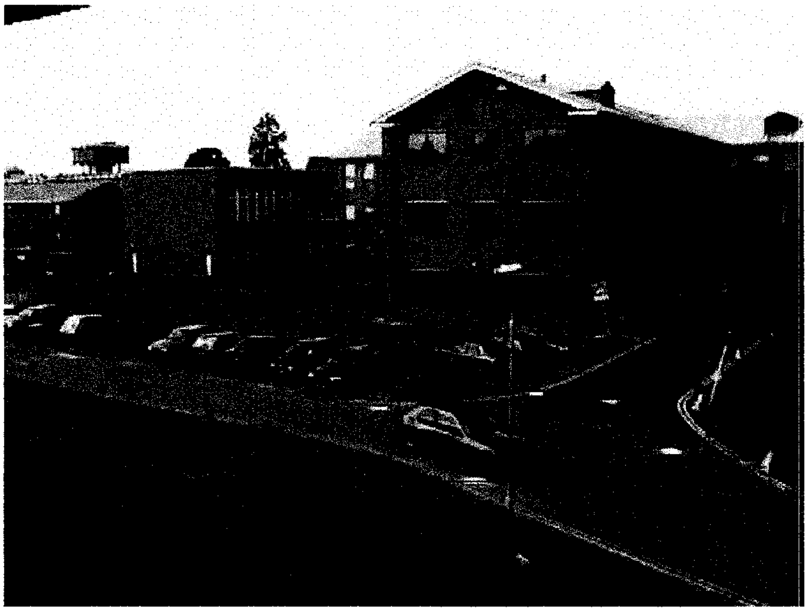
(e)

Fig. 17. (a)-(e) Tracking a single person with a luggage in an indoor environment.

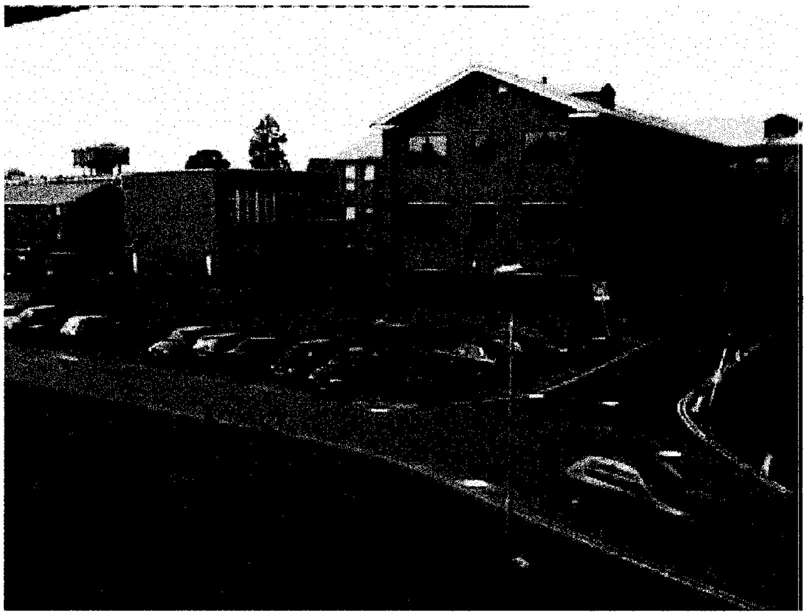


(a)

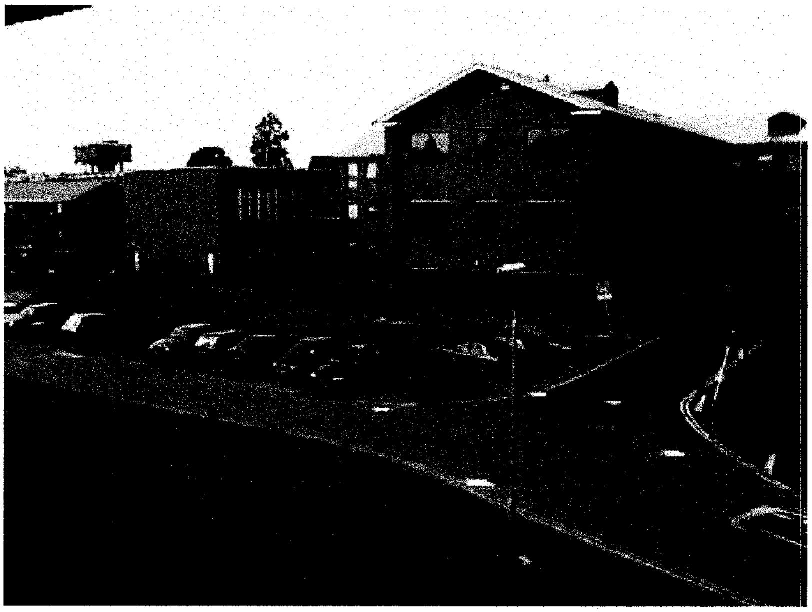
(b)



(c)



(d)



(e)

Fig. 18. (a)-(e) Tracking a single car outdoors.

(a)

(b)

(c)

(d)

(e)

(f)

Fig. 19. (a)-(f) Tracking multiples objects indoors.

(a)

(b)

(c)

(d)

(e)

Fig. 20. (a)-(e) Tracking multiple objects outdoors with varied background.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis, we propose a novel approach to efficient detection and tracking of multiple moving objects in both real-world outdoor and indoor environments using a single static video camera. The algorithm is robust to partial static occlusion, clutter, cast shadows, and changes in object scale and appearance. The experimental results prove the efficiency of the proposed method.

To detect the moving objects, we compute a combined color-texture feature vector for each image block and measure the change of the color-texture feature vector of the image block within a certain time interval. Combining color with texture as the feature vector and measuring the change of this feature vector for detecting moving objects can effectively remove shadows from moving objects. Our motion detection algorithm has good detection performance in busy environments where a clean background is not available. For tracking and identification of the detected multiple moving objects, we represent each object by an attributed relational graph, in which vertices are associated to an object's sub-regions and edges represent spatial relations among the sub-regions. Through ARG, the system has more power to distinguish different objects than many other systems that use global features, e.g. color histograms. Multiple cues are integrated to describe each object's sub-regions. Inexact graph matching enables us to track partially occluded objects and to handle object articulation. We incorporate an ARG adaptation scheme into the system to handle the changes in object scale and appearance.

6.2 Future Work

There are several aspects which need to be future explored.

- ✧ If several objects enter the scene together, the proposed motion detection algorithm will detect them as a single connected component. One of the future works is to segment the single blob into individual objects when an object is grouped with others. When people are the objects of tracking, recognizing the head top on the foreground boundary is one efficient method to locate overlapping persons[30].
- ✧ The algorithm deals with cast shadows satisfactorily in most cases. However, very long shadows may not always be completely removed. More powerful shadow removal module should be added into the system to improve the current method.
- ✧ The proposed tracking and identification algorithm can efficiently handle partial static occlusion, but the performance will degrade when an object is totally occluded. Kalman filter or particle filter may be use to predict the state of objects in case of complete occlusion. In addition, the system does not include a scheme to deal with dynamic occlusions where moving objects occlude each other. [21] supplies a solution to the situation where the models of each involved object are available before the dynamic occlusion occurs.

REFERENCES

[1] Q. Zhou and J.K. Aggarwal, "Object Tracking in an Outdoor Environment using Fusion of Features and Cameras," Image and Vision Computing, vol. 24, no. 11, pp. 1244-1255, November 2006.

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human body," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780-785, July 1997.

[3] C. Stauffer and W. Grimson, "Learning Patterns of Activity using Real-Time Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 747-757, Aug. 2000.

[4] M. Harville, G. Gordon, and J. Woodfill, "Foreground Segmentation using Adaptive Mixture Models in Color and Depth," Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video, Vancouver, Canada, pp. 3-11, 2001.

[5] L. Xu, J. Landabaso, and B. Lei, "Segmentation and Tracking of Multiple Moving Objects for Intelligent Video Analysis," BT Technology Journal, vol. 22, no. 3, pp. 140-150, 2004.

[6] Y. Wang, T. Tan, K. Loe, and J. Wu, "A Probabilistic Approach for Foreground and Shadow Segmentation in Monocular Image Sequences," Pattern Recognition, vol. 38, no. 11, pp. 1937-1946, November 2005.

[7] A. Elgammal, R. Duraiswami, D. Harwood and L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," Proceedings of the IEEE, vol. 90, no. 7, pp. 1151-1163, July 2002.

[8] A. Mittal and N. Parogios, "Motion-Based Background Subtraction using Adaptive Kernel Density Estimation," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 2, pp. 302-309, July 2004.

[9] N. Oliver, B. Rosari, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 831-843, Aug. 2000.

[10] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain, "A Background Model Initialization Algorithm for Video Surveillance," Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001), vol. 1, pp. 733-740, 2001.

[11] J. Rittscher, J. Kato, S. Joga, and A. Blake, "A Probabilistic Background Model for Tracking," Proceedings of the 6th European Conference on Computer Vision-Part II, pp. 336 – 350, 2000.

[12] O. Javed, K. Shafique, and M. Shah, "A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information," Proceedings of the Workshop on Motion and Video Computing (MOTION'02), pp. 22-27, 2002.

[13] N. Peterfreund, "Robust Tracking of Position and Velocity with Kalman Snakes," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 6, pp. 564-569, June 1999.

[14] P. Brasnett, L. Minhaylova, N. Canagarajah, and D. Bull, "Particle Filtering with Multiple Cues for Object Tracking in Video Sequences," Proc. of SPIE-IS&T Electronic Imaging, vol. 5685, pp. 430-441, 2005.

[15] P. Perez, C. Hue, J. Vermark, and M. Gangnet, "Color-based Probabilistic Tracking," ECCV 2002, LNCS 2350, pp. 661-675, 2002.

[16] R. Wang and T. Huang, "A Framework of Joint Object Tracking and Event Detection," Pattern Analysis & Applications, vol. 7, no. 4, pp. 343-355, December 2004.

[17] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003.

[18] S. Mckenna, Y. Raja, and S. Gong, "Tracking Colour Objects using Adaptive Mixture Models," Image and Vision Computing, vol. 17, no. 3-4, pp. 225-231, March 1999.

[19] B. Han and L. Davis, "On-Line Density-based Appearance Modeling for Object Tracking," Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 2, pp. 1492-1499, 2005.

[20] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Spatial Color Indexing and Applications," International Journal of Computer Vision, vol. 35, no. 3, pp. 245-268, December 1999.

[21] M. Balcells, D. Dementhon, and D. Doermann, "An Appearance-based Approach for Consistent Labeling of Humans and Objects in Video," Pattern Analysis & Applications, vol. 7, no. 4, pp. 373-385, December 2004.

[22] N. Siebel and S. Maybank, "Real-Time Tracking of Pedestrians and Vehicles," Proceedings of 2nd IEEE Int. Workshop on PETS (PETS'2001), Kauai, Hawaii, USA, December 2001.

[23] A. Yilmaz, X. Li, and M. Shah, "Contour-based Object Tracking with Occlusion Handling in Video Acquired using Mobile Cameras," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pp. 1531-1536, November 2004.

[24] F. Cheng and Y. Chen, "Real Time Multiple Objects Tracking and Identification based on Discrete Wavelet Transform," Pattern Recognition, vol. 39, no. 6, pp. 1126-1139, June 2006.

[25] E. Ozyildiz, N. Krahnstover, R. Sharma, "Adaptive Texture and Color Segmentation for Tracking Moving Objects," Pattern Recognition, vol. 35, no. 10, pp. 2013-2029, October 2002.

[26] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah, "Sequential Monte Carlo Tracking by Fusing Multiple Cues in Video Sequences," Image and Vision Computing, vol. 25, no. 8, pp. 1217-1227, August 2007.

[27] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, "Tracking Groups of People," Computer Vision and Image Understanding, vol. 80, no. 1, pp. 42-56, October 2000.

[28] C. Lerdsudwichai, M. Abdel-Mottaleb, and A. Ansari, "Tracking Multiple People with Recovery from Partial and Total Occlusion," Pattern Recognition, vol. 38, no. 7, pp. 1059-1070, July 2005.

[29] T. Ellis and M. Xu, "Object Detection and Tracking in an Open and Dynamic World," Proceedings of 2nd IEEE Int. Workshop on PETS (PETS'2001), Kauai, Hawaii, USA, December 2001.

[30] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Complex Situations," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1208-1221, September 2004.

[31] L. Latecki, R. Miezianko, and D. Pokrajac, "Motion Detection Based on Local Variation of Spatiotemporal Texture," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04), pp. 135-141, June 2004.

[32] J. Li, J. Wang, and G. Wiederhold, "Classification of Textured and Non-Textured Images using Region Segmentation," Proceedings of the 7th International Conference on Image Processing, Vancouver, pp. 754~757, 2000.

[33] H. Nguyen and A. Smeulders, "Robust Tracking using Foreground-Background Texture Discrimination," International Journal of Computer Vision, vol. 69, no. 3, pp. 277-293, September 2006.

[34] S. Lee, H. Bae, and S. Jung, "Efficient Content-based Image Retrieval Methods using Color and Texture," ETRI Journal, vol. 20, no. 3, pp. 272-283, 1998.

[35] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Assison Wesley, Reading, MA, 1992.

[36] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting Moving Shadows: Algorithms and Evaluation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 918-923, July 2003.

[37] A. Cavallaro, E. Salvador, and T. Ebrahimi, "Shadow-Aware Object-based Video Processing," IEE Proc.-Vis. Image Signal Process., vol. 152, no. 4, pp. 398-406, August 2005.

[38] F. Porikli and J. Thornton, "Shadow Flow: A Recursive Method to Learn Moving Cast Shadows," Proceedings of Tenth IEEE International Conference on Computer Vision, 2005 (ICCV 2005), vol. 1, pp. 891- 898, Oct. 2005.

[39] L. Consularo and R. Cesar-Jr., "Quadtree-Based Inexact Graph Matching for Image Analysis," Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI' 2005), pp. 205-212, 2005.

[40] R. Cesar Jr., E. Bengoetxea, I. Bloch, and P. Larranaga, "Inexact Graph Matching for Model-Based Recognition: Evaluation and Comparison of Optimization Algorithms," Pattern Recognition, vol. 38, no. 11, pp. 2099-2113, November 2005.

[41] T. Baba, R. Liu, S. Endo, S. Shiitani, Y. Uehara, D. Masumoto, and S. Nagata, "Similarity-based Partial Image Retrieval System for Engineering Drawings," Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05), pp. 303-310, 2005.

[42] P. Foggia, R. Genna, and M. Vento, "Prototype Learning with Attributed Relational Graphs," SSPR&SPR 2000, LNCS 1876, pp. 447-456, 2000.

[43] S. Aksoy, "Modeling of Remote Sensing Image Content using Attributed Relational Graphs," Proceedings of Joint IAPR International Workshops, SSPR/SPR 2006, Hong Kong, China, pp. 475-483, August 2006.

[44] D. Pelleg and A. Moore, "X-means: Extending K-means with Efficient Estimation of the Number of Clusters," Proceedings of the 17th International Conf. on Machine Learning, San Francisco, USA, pp.727-734, 2000.

[45] D. Comaniciu and P. Meer, "Mean Shift: a Robust Approach toward Feature Space Analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002.

VITA AUCTORIS

Wei Huang received her Bachelor's degree from Beijing Institute of Technology, Beijing, China. She is currently a candidate for the Master's degree in Electrical Engineering at the University of Windsor and hopes to graduate in Fall 2007.