Multiple Objects Tracking by Color-based Methods
Chun-Hung Chen and Cheng-Yan Kao
Department of Computer Science and Information Engineering
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei
Taiwan(R.O.C)

Abstract: Motion object tracking in real-time environment and video is a popular topic in multimedia processing. Various related researches proposed to handle particular cases in recent years. We proposed several modified methods about background modeling, foreground detection, moving object modeling and matching to achieve the goal of tracking multiple objects in indoor and outdoor scenarios. In our experimental settings, we can discriminate and track objects accurately as well as detect and deal with occlusions in all videos.

Key-Words: video surveillances, background modeling, foreground extraction, contrast histogram, object tracking, occlusion detection.

1 Introduction
Multimedia and image processing techniques have been developed for decades. Intelligent video surveillance system became a popular research topic along the improvements of hardware and software. We can record a high-resolution video by a camera and process every frame in real-time. The processes varied according to goals, recognized patterns of interest, tracked special objects, foreground only objects, and so on.

Video analysis has been popular research field in recent years. Application areas include hospitals, casinos, hotels, schools, and so on. Crime and emergency events may happen anywhere, from video records we can acquire clues, track suspects, and analyze activities.

In this section we discuss some robust systems proposed by others, but not all of them meet the requirement of real-time applications. Next two sections describe important processes in our system. We introduced a background model and two mechanisms to extract foreground objects in section 2. And in section 3 there are two filters which can be used to match object models and objects found in foregrounds. Section 4 and 5 are experiment results and conclusion, respectively.

1.1 Related Work
Object tracking can be divided into several steps, including foreground extraction (background subtraction), object recognition, model construction and update.

Chien et al. [1] proposed an approach for foreground extraction incorporating both consecutive frames subtraction and current frame and background subtraction. They described six cases to decide motion object, based on the results of background difference and frame difference.

Wallflower is a classic method to maintain background model which consisted of three levels (pixel, region and frame) [2]. In pixel level the foreground pixels would be discovered by history records, then the creation of each object is executed in terms of histograms of connected regions, adaptation of background models would take place in frame level.

Mixture of Gaussians is another efficient approach of foreground extraction [3]. Foreground pixels can be found from the comparison of several Gaussian models, which can be categorized as foreground and background models. These models have been built in terms of pixels observed previously.

Tracking moving objects demands appropriate mechanisms to construct, compare and update object models. Characteristics like texture, edge and corner or color distribution in local areas are common means to describe objects. Adaptive boosting is a widely used method mainly for face detection [4][5]. However, it needs sufficient training data before experiments. In real-time environment we created object models when new objects appeared first time and updated models every time when an object matching to a model.

A model of more information will bring better performance of tracking, Wang and Yagi [6] took three color spaces (RGB, SHV and normalized RG), shapes and textures as cues to model objects. Every color bands will be divided into various bins.
Techniques adapted including likelihood ratio to select features and mean-shift algorithm to track objects [7]. The later is a common approach of probability density functions of color histograms.

2 Preprocessing work
Before beginning tracking moving objects, we must do some preprocessing work to improve the efficiency of tracking. Two important tasks are building the background model and deciding the possible foregrounds.

2.1 Background Model Construction
The background source can be made from two cases, one is the frames without motion objects, we can call them “real backgrounds” because only stationary objects involved. The foreground will be extracted correctly if using efficient methods, but due to some effects like light, shadow and electro-mechanical problems, background model may deviate away from the reality. There are lots of solutions to this problem, such as building a nearly perfect model or updating model after we receive new frames.

The other source is taking previous frame as the current background. We do not consider the effects of lights if the time interval between two continuous frames is short. But after frame subtraction the foreground may not equal to moving object, because the overlay regions of motion object in the two frames won’t present obvious difference, this leads to that only parts of motion objects are regarded as foreground. This issue will cause difficulty for object tracking.

Considering the pros and cons of the two cases of background models, we choose the first as our mechanism and develop suitable updating method to maintain the integrity of the motion objects.

2.1.1 Background Mosaic
To save the computing time in the real-time environment, we divide background image into blocks with the same size. For every block we compute the mean and standard deviation of intensity of all pixels in it [8]. Then we call the contrast of a block as

\[
C(i, j) = \frac{sd(i, j)}{\text{mean}(i, j)},
\]

(1)

where \((i, j)\) is the position of the block, \(sd(i, j)\) the standard deviation, \(\text{mean}(i, j)\) the mean of intensity.

All information will be compared with current frame for foreground extraction and motion object identification later.

2.2 Foreground Extraction
We also compute the values of mean, standard deviation and contrast of all blocks in current frames. Afterward, we compare the contrasts of blocks in the same position between background model and the current frame. If the difference of contrasts between two blocks exceeds the threshold we set previously (the value will be assigned empirically), the block will be called “foreground block”.

The two methods will be combined and applied consecutively. A block will be a foreground block if it is identified by either method. On the other hand, a block will be background block if it does not be caught in both methods.
After identifying all foreground blocks in current frame, we merge these blocks to “foreground regions”. A foreground region is a connected component of foreground blocks. The definition of connection is that if two foreground blocks are 8-neighbors, they are connected, otherwise they are not connected. Because of the effect of noise, we set a threshold for foreground regions. If the size of a region did not exceed the threshold, the region would be treated as noise instead of foreground.

### 2.3 Background Model Updating

In previous section we discussed the background model issues. To match the current background we have to update the model. Something may join the background after we begin tracking, we can find that blocks corresponding to these things always recognized as foreground blocks, but in fact they have became parts of backgrounds.

We create a counter for all blocks [10]. When a block recognized as a foreground block, the counter of this block will increment, otherwise subtract. If the counter exceeds the threshold, we update the block in background model. That is, the contrast and mean intensity of block in the model will be replaced as the mean contrast and mean intensity of all foreground blocks with the same position in previous frames, respectively.

### 3 Motion Object Identification

From previous work we acquire foreground regions, which are potential motion objects. Here we propose 2 filters to create object models and identification objects.

First, we compare the similarity of color distribution of every pair, one selected from foreground regions and the other from motion object models. If a pair is similar by our definition, the comparison of detailed information of all foreground blocks and motion object model will be carried out in the next step.

#### 3.1 Similarity Comparison

In the beginning we check the similarity of foreground regions and motion object models by color histograms. The histogram is composed of all the color intensity of pixels in foreground regions instead of mean color intensity of every block.

The color intensity will be divided into bins, every intensity value belongs to only one bin. From [11], considering the spatial relation, we assign weights to a pixel by its distance to central pixel in the foreground region, the possibility of a intensity value assigned to a bin \( u \) is

\[
P_u = C \sum_{i=1}^{n} k(|x_i|) \delta [b(x_i) - u],
\]

where \( x_i \) is a pixel in foreground region, \( n \) is the total number of foreground blocks in the region, \( |x_i| \) is the distance of \( x_i \) to central point of the region, the function \( k \) assigns a weight to \( x_i \), \( \delta \) is Kronecker delta function, \( b(x_i) \) the bin \( x_i \) belonged to. Finally \( C \) normalizes \( P(u) \) in the range of 0 and 1,

\[
C = \frac{1}{\sum_{i=0}^{n} k(|x_i|)},
\]

Then, from experience we know that the scales of object changes frequently, the possibility would be changed to

\[
P_u = C \sum_{i=1}^{n} k(\frac{|x_i|}{h}) \delta [b(x_i) - u],
\]

where \( h \) is the scale of foreground region.

Finally the color similarity of foreground regions and motion object models can be evaluated by Bhattacharyya coefficient,

\[
\rho(y) = \rho[p(y)q] = \sum_{u=1}^{n} p_u(y)q_u,
\]

note \( y \) is the central point of foreground region, \( p \) the foreground region, \( q \) the object model.

When the similarity of a pair exceeds the threshold, it represents that the pair passed the first filter. In the next step, we will verify whether the region matches to the model in terms of detailed information. If not, there are three possible cases [12]:

**Case 1:** Occlusion happens, that is, more that one motion objects overlaid in the frame, foreground region contains these objects means the color histogram is the mixture of them. Because this filter can only analyze global information of objects, we just assume the situation (there are multiple objects in a region) happens but we can’t confirm it using this filter. The other filter will identify the situation because it compares local information.

**Case 2:** Motion objects changes their appearances or are occluded by background. For example, a man puts on a jacket, turns his head, or is sheltered by a desk. We can solve the problem by assigning an active position for
every motion object model. If the distance of a foreground region and an object model is short, it’s very possible that the region matched the model. Every time we meet the situation, the threshold of similarity (Bhattacharyya coefficient) will subtract to avoid this case. Note that tolerance degree increases may result in recognition failure.

Case 3: New object emerges, we can’t let the object match to any models through the two filters, we should create a new model for this object. Here a model contains the color distribution of the object and the last position it occurred. If an object hasn’t appeared for a period of time, the model corresponding to it would be deleted because we must save memory space and computing time. Like background model, we will update motion object models to become closer to reality when an object matches to a model,

\[ q_u = \beta p_u + (1 - \beta) q_u, \]  

(6)

The learning rate \( \beta \) is set to 0.1 in our experiments.

3.2 Detailed Information Comparison

When a pair of object and model passed first filter, this filter will check the information in all blocks among them. The contrast context histogram can be used as a tool to decide the similarity of two points in different images [13]. Here we use blocks instead to accelerate computation.

A foreground region composed of foreground blocks. For every neighbor block of every block, we compute the mean intensity of pixels in neighbor block higher than the block and lower than it. Then we get two values from each neighbors (Fig. 2), 16 values totally (8-neighbor). Considering object orientation, we “rotate” these 16 values about neighbor blocks in feature vector to meet the case that object changes its direction when comparing them. A neighbor block can rotate 7 times to locate in other place in Fig. 2, in the same meaning a feature vector can exchange its contents 7 times when comparing with other vectors.

The neighbor blocks of boundary blocks changed when objects moved. Some neighbors are composed of backgrounds. This means that the features we compute above are not adequate for boundary blocks. In case of errors, we compute features inside a block too. To keep the same form, we divided a block into nine sub-blocks of identical sizes as Fig. 2. The interior sub-block will be ignored. The same two values (mean intensity values of pixels greater and lower than interior sub-block) will be computed from every exterior sub-block, joining with mean red, green, blue values of the block, total 19 values as the contents of feature vector of the block.

The detailed model includes the information of neighbor blocks, mean color intensity, and the latest time when blocks occurred. A model will be deleted if it hasn’t appeared for a long time.

![Fig. 2. A varied form of contrast context histogram.](image)

In application 1, The central block represents a foreground block, other blocks are its neighbor blocks. We computed two values for each neighbor block. In application 2, a foreground block will be divided into 9 sub-blocks, we also computed 2 values for 8 responding sub-blocks.

3.3 Occlusion Detection

A tough problem of tracking is occlusion. We modified the two filters above to detect two cases of occlusion. One is that a foreground region contains more than one motion objects, and the other is a motion object covered by background. When a foreground region couldn’t match to any models by filter 1, we compare the foreground blocks with all models, according to the result we take different actions. If no models exist in the region, the region represents a new object, and we create a new model for it. If just one model is recognized, it means that the object was covered by background or it changed its appearance. Otherwise, if more than one models are recognized, occlusion of some models happened. Note that when occlusion took place, we didn’t update the detailed model of identified objects because we assumed these features are biased.

4 Experimental Results

Several videos have been adopted to evaluate our methods, for each we showed the foreground regions and/or recognition results of selected frames.

4.1 Occlusion of Indoor Environment
In the video (resolution is 352 x 288) two men appeared in two different sides and walked toward the other side. Then they shook hands and one of them left a case. In the end they went to opposite directions and disappeared. In Fig. 3, (a), (c), (e) are foreground regions of 3 frames. We can find the occlusion happens in (c). Blocks in different colors (b), (d), (f) represented unique object models.

4.2 Occlusion of Outdoor Environment
Another video (resolution is 640 x 480) presented that two men walked around in a square. In frame 620 we captured two objects (Fig. 4a) and they could match to two models in different colors (Fig. 4b), respectively. In frame 635 the man in left side covered the other partly (Fig. 4c), it is a case of occlusion. We can discriminate this mixture of objects successful in Fig. 4d. Although the man in the right side can not be recognized perfectly.

4.3 Vehicles Tracking
Test video was downloaded from PETS2001 dataset 1 and 2 [14], for general purpose we resized the resolution from 768x576 to 640x480 (Fig. 6, dataset 2). The background included buildings, cars, and grass, etc. Fig. 6a is not identical but similar to the background model. In following frames multiple people and vehicles appeared randomly. In Fig. 6b, a car is driven on the road in the left side. Next in Fig. 6c, it would leave the image. Several seconds later, the car came back (Fig. 6d), a man riding bicycle as another object in Fig. 6e and Fig. 6f.

Dataset 1 used the same view as Fig. 6a, but more people and vehicles showed (Fig. 7). We ignored all people and focused on vehicles. In Fig. 7a a car appeared on the bottom-right side, as this car pulled up, a white van appeared in Fig. 7b. A third car joined them in Fig. 7c. In the end, we still tracked them efficiently in Fig. 7d.

5 Conclusion
We proposed a motion object tracking system for indoor and outdoor environments, which occlusions randomly happened. We focus on tracking and occlusion control. Multiple methods about background modeling, foreground extraction, and object modeling and matching would be integrated to achieve the goal.

In experiments we tested our system using videos made by ourselves or downloaded from internet. The proposed system achieved satisfactory results of human and vehicle tracking through occlusions.
Fig. 6. (a) Frame 1 (b) Frame 320 (c) Frame 395 (d) Frame 1300 (e) Frame 1625 (f) Frame 1685

Fig. 7. (a) Frame 500 (b) Frame 0950 (c) Frame 2200 (d) Frame 2625

References: