MERL – A MITSUBISHI ELECTRIC RESEARCH LABORATORY
http://www.merl.com

Multi-Camera Surveillance: Object-Based Summarization Approach

Fatih Porikli

TR-2003-145 March 2004

Abstract

An automatic object tracking and video summarization method for multicamera systems with a large number of non-overlapping field-of-view cameras is explained. In this system, video sequences are stored for each object as opposed to storing a sequence for each camera. Object-based representation enables annotation of video segments, and extraction of content semantics for further analysis and summarization. Objects are tracked at each camera by background subtraction and mean-shift analysis. Then the correspondence of objects between different cameras is established by using a Bayesian Belief Network. This framework empowers the user to get a concise response to queries such as "which locations did an object visited on Monday and what did he do there?"

Publication History:

1. First printing, TR-2003-145, March 2004

Chapter #

MULTI-CAMERA SURVEILLANCE

Objec-Based Summarization Approach

Fatih Porikli
Mitsubishi Electric Research Laboratories

Abstract: An automatic object tracking and video summarization method for multi-camera systems with a large number of non-overlapping field-of-view cameras is explained. In this system, video sequences are stored for each object as opposed to storing a sequence for each camera. Object-based representation enables annotation of video segments, and extraction of content semantics for further analysis and summarization. Objects are tracked at each camera by background subtraction and mean-shift analysis. Then the correspondence of objects between different cameras is established by using a Bayesian Belief Network. This framework empowers the user to get a concise response to queries such as ``which locations did an object visited on Monday and what did he do there?"

Key words: Multi-camera surveillance, summarization, belief-nets

Most of the current indoor surveillance applications have single-camera single-room architecture where the cameras are stationary. Typically, each camera is assigned to a dedicated video recorder that can store the streaming video in either time-lapsed or event-based mode. These events are often limited to simple motion detection mechanisms. Considering the huge amount of the video data a multi-camera system may produces over a short time period, more sophisticated tools for control, representation, and content analysis became an urgent need. The nature of surveillance applications demands automatic and accurate detection of object of interest, intra-camera tracking, fusion of multiple modalities to solve inter-camera correspondence problem, easy access and retrieving video data, capability to make semantic query, and abstraction of video content.

Yet another challenge is the extraction of content semantics. In last decade, the coding standards have allowed efficient storage, compression, and communication by handling video information as signals. More recently,

object-based encoding and content-based retrieval become possible by extracting and analyzing features of pixels. However, the challenge remains for automatically extracting semantic labels for video content, including labeling of objects, events, places, people, and so forth. By labeling video content at the semantic level, the content will be easier to search, filter, index, summarize, and personalize. The process of video technologies involves transition from dealing with pixels to features to semantics to knowledge as illustrated in Fig.1. In bridging these gaps from pixels to knowledge, objects and models have an important role. The MPEG-7 standard embodies content description models but it does not specify how to extract them. Here, we developed an object-based video content labeling method to restructure the camera-oriented videos into object-oriented results. We propose a summarization technique using the motion activity characteristics of the encoded video segments to provide a solution to the storage and presentation of the immense video data.

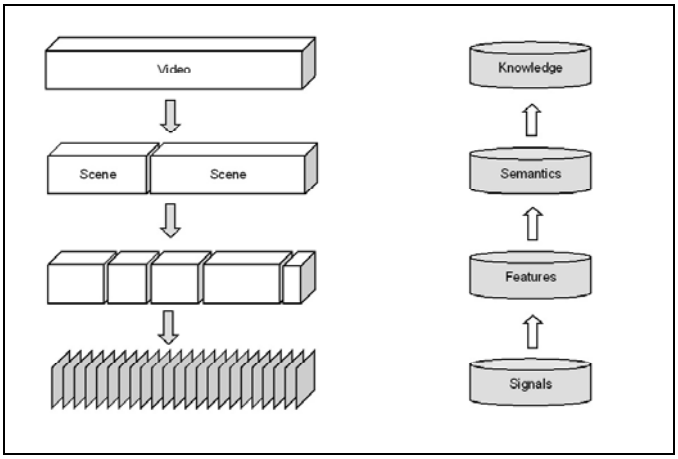

Figure #-1. Formative and informative representation of a video sequence

Although several multi-camera setups have been adapted for 3D vision problems, the non-overlapping camera systems are not investigated thoroughly. A multi-camera system is proposed in [1]. This system is based on a Gaussian mixture model background subtraction and Kalman filtering to find people in an indoor environment. A Bayesian network is used to combine multiple modalities. Among these modalities, the epipolar, homography, and landmark information assume any pair of cameras in the system has an overlapping field-of-view. Therefore, it is not applicable to the single-camera/single-room architectures.

In this paper, we designed a framework where we can extract the object-wise semantics from a non-overlapping field-of-view multi-camera system.

This framework has four main components: automatic tracking, inter-camera data fusion, query generation, and summarization. A flow diagram of the system is shown in Fig. 2. In Section 2, we present a single-camera tracking method. Section 3 explains a Bayesian Belief Network for inter-camera correspondence by using object properties and system modalities such as camera location information. In Section 4, we present query generation and summarization.

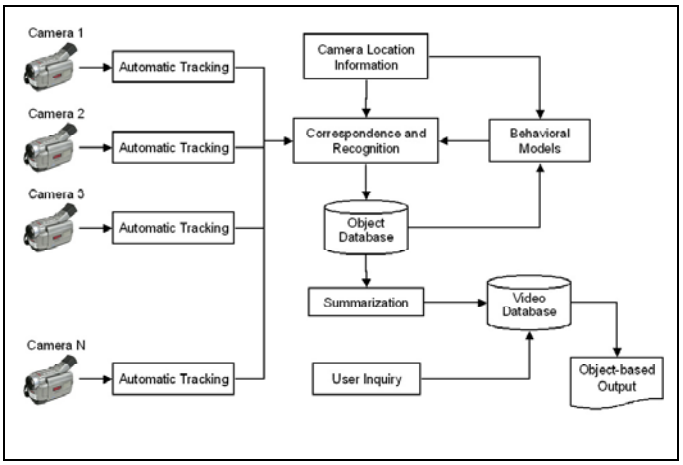

Figure #-2. Architecture of the multi-camera surveillance system

1. AUTOMATIC TRACKING

A common approach for detecting a moving object for a stationary camera setup is background subtraction. The main idea is to subtract the current image from a reference image that is constructed from the static image pixels during a period of time. Background detection approaches can be classified as non-adaptive and adaptive methods. Manual selection, pixel-wise voting, and mean value search algorithms are among the non-adaptive methods. Adaptive methods include averaging consecutive frames over time, Gaussian mixture models [5,6], alpha blending [7], Kalman filtering [8], and other statistical models [9].

Although averaging and alpha blending are simple and fast, they are not effective for scenes with many moving objects particularly if they move slowly. Besides, they cannot handle multi-modal backgrounds. They may not recover when an object occupies the scene at the initialization phase. Pixel-wise voting among the accumulated images may handle some of the

recovery problems, however it becomes computationally very expensive with the increasing number of images.

The Kalman filtering approach may only provide some partial solution. Since lighting conditions may change in most applications, the reference image should be adaptive as well. The Gaussian model based approaches have capability of dealing with illumination changes. Also, it can learn the repetitive variations. However, objects that stop moving may become a part of the background in case the object boundaries are not exact. For the high number of models (>3), this method becomes too slow to be practical.

The tracking of objects can be done either by backtracking or by forward tracking. The backtracking based approach segments foreground regions in the current image and then establishes the correspondence between the current and previous images.

The forward-tracking approach estimates the positions of the regions in the current frame using the segmentation result obtained for the previous image. The limitation of the backtracking approaches is that fixed templates may not be sufficient for all possible objects. A well-known forward-tracking technique is mean-shift analysis, which is a nonparametric density gradient estimator [3]. It is employed to derive the object candidate that is the most similar to a given model while predicting the next object location. This method provides accurate localization, and it is computationally fast. However, it is not automatic since it requires initial models.

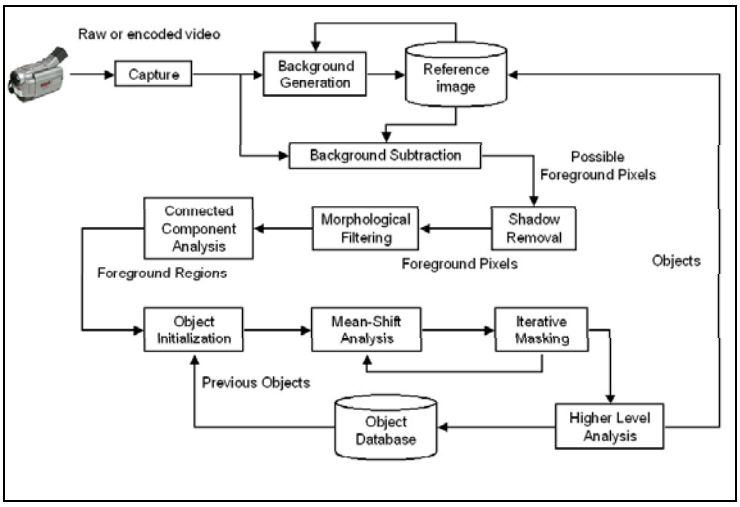

Figure #-3. Single-camera tracking algorithm.

As shown in Fig. 3, our method constructs a reference image using pixel-wise mixture models, finds changed part of image by background subtraction, removes shadows by analyzing color and spatial properties of pixels, determines objects, and tracks them in the consecutive frames.

In background subtraction, the current image is compared to a reference image to detect the changed pixels. The reference image is constructed by utilizing pixel-wise mixture of models as in [5]. We model the history of each pixel by a mixture of Gaussian distributions as

$$P(p,t) = \sum_{n}^{N} w_n(t) g\left(p, \mu_n(t), \sigma_n^2(t)\right)$$

where N is the number of the distributions, $w_n(t)$ is the weight, $\mu_n(t)$ and $\sigma_n^2(t)$ are the mean and the variance of the n^{th} Gaussian model at frame t respectively, and g is a density function as

$$g\left(p, \mu_n(t), \sigma_n^2(t)\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[I(p)-\mu]^2}{\sigma^2}}$$

The reference image is updated by comparing the current pixel with the existing Gaussian distributions. In case of the current pixel's color value is within a certain distance of the mean value of a distribution, it is assigned as a match. This distance threshold is set to 2.5σ to include 95% of the distribution. If none of the K distributions ($K<N$) match the current pixel value, a new distribution is initialized. In case of $K=N$, the distribution with the highest variance is replaced with a distribution with the current pixels value as its mean value, and an initial variance. The initial variance is chosen a large value for all distributions. The mean and variance of the matched distributions are updated using a learning coefficient as

$$\mu_n(t) = [1-\alpha]\mu_n(t-1) + \alpha I(p)$$
$$\sigma_n^2(t) = [1-\alpha]\sigma_n^2(t-1) + \alpha[\mu_n(t) - I(p)]^2$$

The weights of the distributions at a frame are adjusted by alpha blending as

$$w_n(t) = \begin{cases} (1-\alpha)w_n(t-1) + \alpha & |\mu_n(t) - I(p)| < 2.5\sigma \\ (1-\alpha)w_n(t-1) & |\mu_n(t) - I(p)| \geq 2.5\sigma \end{cases}$$

The learning coefficient α serves as a parameter that controls the rate of the adaptation of the reference image to the current frame. For this purpose, we measure the illumination change δ for a small subset Q of the pixels as

$$\delta = \left| 1 - \sum_{q \in Q} \frac{\langle B(q), I(q) \rangle}{|B(q)|} \right|$$

where *B(q)* represents the background color vector at the pixel *q*. In case the value of the illumination change is relatively large, the learning parameter is adjusted linearly by $\alpha = c_1 + c_2\delta$. Unlike the traditional background update mechanisms that refresh the current background model at certain time intervals, we adapt the frequency of the update mechanism Δt_m by using the illumination change as

$$\Delta t_m = \begin{cases} \Delta t_{\max} & \delta < \tau_{\min} \\ \Delta t_{m-1} & \tau_{\min} \le \delta < \tau_{\max} \\ 1 & \tau_{\max} \le \delta \end{cases}$$

where $\tau_{\min} << \tau_{\max}$, and $\Delta t_{\max}$ is the number of frames that background model should be updated even if there is no significant illumination change.

After background subtraction, we detect and remove shadow pixels from the set of the foreground pixels. Likelihood of being a shadow pixel is evaluated iteratively by observing the color space attributes and local spatial properties. Shadow removal has two stages. At the first stage pixel-wise color change is evaluated to determine the possible shadow pixels. At the second stage, an iterative classification based on the local information within a local window around a pixel is done. After shadow removal, we have the binary image of foreground pixels that corresponds to the objects. The next task is to find the separate objects. To accomplish this, we first remove speckle noise, then determine connected regions, and group regions into separate objects. To speed up the filtering, we map each 32 horizontal pixels of the binary foreground-background map into a 4-byte integer number. By shifting right and left, and applying logical inclusion with the upper and lower rows, we actually do a morphological dilation operation. In the second pass, logical exclusion is applied similarly to erode the binary image.

While the connected component analysis, we compute the total number of pixels of a connected region, its center of mass, ant its inner/outer boxes coordinates. The inner box contains 90% of the pixels by starting from the pixels close to the center of mass. A rule-based decision mechanism initializes an object by evaluating the connected components. We use box closeness to merge the connected components. For each group of merged components, an object such that its status is set to "possible" is initialized, and a singe outer box is fitted. We track objects by computing the highest gradient direction of color histogram, which is implemented as a maximization process. Using the histogram $h_1(n)$ extracted at the previous frame, this process is iterated as

1. Compute the histogram $h_2(y_0)$ in the current frame, calculate

$$\rho[h_1, h_2(y_0)] = \sum_n^N \sqrt{h_1 . h_2(y_0)}$$

2. Compute the weights β_i *i=1,...,R*
3. Derive the new location y_1 by mean-shift,

$$y_1 = \frac{\sum_i^R I(p_i)\beta_i k(p_i)}{\sum_i^R \beta_i k(p_i)}$$

4. Update the target histogram $h_2(y_1)$, and calculate $\rho[h_1, h_2(y_1)]$ as above
5. Stop if $\left|\rho[h_1, h_2(y_1)]\right| < \varepsilon$, else $y_1 \to y_0$, go to step 1.

where the distance between two histograms are defined as $d(y) = \sqrt{1 - \rho[h_1, h_2(y)]}$. A sample tracking result is shown in Fig.4.

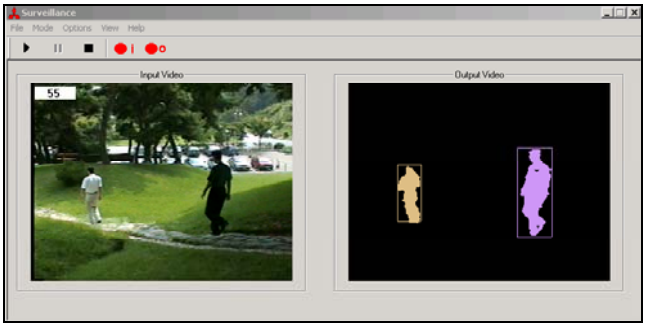

Figure #-4. Single-camera tracking example from MPEG-7 ETRI dataset

2. CORRESPONDENCE AND FUSION

Another issue of the multi-camera system is the problem of integrating the tracking results of multiple cameras. To find the corresponding objects in different cameras and in a central database that keeps records of the previous appearances of all objects, the system evaluates the likelihood of possible pair-wise object matches. This evaluation is done by fusing the general object features such as color, shape, texture features, and other application specific modalities, i.e. camera layout information, behavioral statistics, human-face features, etc.

Color feature is the most common feature that widely accepted by object recognition systems since it is relatively robust towards the size and orientation changes. Possible color features are color templates, histograms, moments, signatures (dominant colors), and partitive color layouts. In a

multi-camera setting, illumination, camera distortion, and object resolution differences are most likely to happen. Thus, the color feature should be able to compensate inter-camera distortions as well as illumination changes. Since pixel-wise object template representations are very sensitive to the scale deviations, they are not suitable in our setting. Color signature is defined as a selection of the colors from the quantized color space. A disadvantage of color signature is that they are computational complex. Color layout features have the ability of representing the spatial and color distribution properties at the same time. To extend the global color histogram to a local one, a natural approach is to divide the whole object into sub-blocks and extract color features from each of the sub-blocks. However, they require careful application since they depend on the shape of the object. Thus, we preferred to use color histogram to represent color properties of objects.

Statistically, a color histogram denotes the joint probability of the intensities of the color channels. By modeling the inter-camera distortion and illumination changes as functions of histograms, the matching performance can be improved. We use a cross-correlation based histogram similarity metric to compensate the illumination and inter-camera distortions. This metric uses a cross-correlation matrix H where $h_{mn} = 1 - |h_1(m) - h_2(n)|$ using the normalized color histograms of the corresponding objects at different cameras. We find the maximum gain path by dynamic programming. By comparing this path with an inter-camera characteristic path for the current camera pair, we compensate for the camera distortion. The inter-camera characteristics are obtained by training. This metric evaluates the illumination differences as well.

Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity. Although, it contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment, such level of detail is usually not available in low-resolution surveillance video.

Shape provides another clue for object matching. In general, Fourier descriptor and moment invariants are the most common shape representations. The main idea of Fourier descriptor is to use the Fourier transformed boundary as the shape feature. The main idea of moment invariants is to use region-based moments, which are invariant to transformations, as the shape feature. The biggest drawback of shape features is the sensitivity to scale changes and boundary inaccuracies. Using height descriptor will only help if we have the ground plane. However, since the existing multi-camera systems are difficult to calibrate, a precise ground plane is difficult to obtain. Therefore, the height is not an effective feature.

Using faces to match object between cameras remains the only solution for certain cases, i.e. at a military complex that everyone dresses in identical clothes. However, in typical surveillance applications cameras are usually located far away from the object routes, which result in low-resolution face images. Another concern is the face orientation. Most face-based methods work only for frontal images. The accuracy of identification quickly decreases even with the slight orientation differences. Acquiring a high resolution and frontal picture of an object is not possible always to facilitate facial identification methods. Still, image resolution enhancement techniques may render face features for matching problems in the future.

There is a strong correlation between camera system geometry and likelihood of the objects appearing in a certain camera after they exit from another one. As illustrated in Fig.5, we formulate the camera system as a Bayesian Belief Network, which is a graphical representation of a joint probability distribution over a set of random variables. A BBN is a directed graph in which each set of random variable is represented by a node, and directed edges between nodes represent conditional dependencies. The dependencies can represent the casual inferences among variables. The transition probabilities, which correspond to the likelihood of a person moving from one camera to another linked camera, are learned by observing the system. Note that, each direction on a link may have different probability, however the total incoming and outgoing probability values are equal to one. To satisfy the second constrain, some slack nodes that correspond to the unmonitored entrance/exit regions are added to the graph.

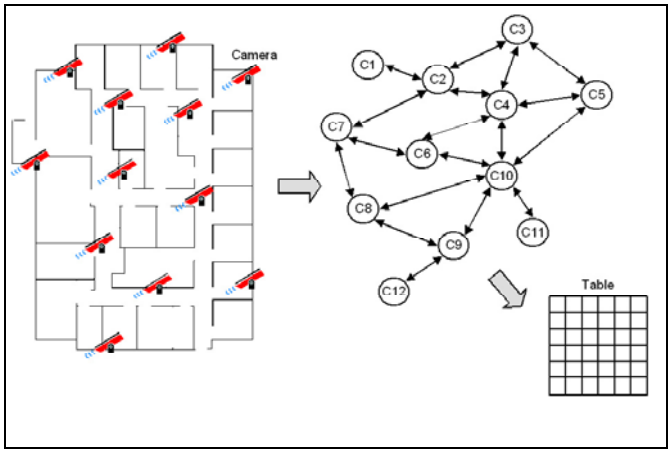

Figure #-5. Each camera corresponds to a node in the directed graph. The links show the possible physical routes between the cameras.

Initially, there is no object assigned to any node of the BNN, the number of objects in the cameras and objects in the database are equal to zero. The database keeps track of individual objects. Let *Ci* the camera object *O* is detected. For each detected new object, a database entry is made using its color histogram features. If the object *O* exits from the camera *Cj*, then the conditional probability $P_O(Cj|Ci)$$ of the same object will be seen on a camera C_j is computed by $P_O(C_j|C_i) = P(Cj|C_{s1})P(C_{s1}|C_{s2})...P(C_{sk}|Ci)$ where *{s1,s2,..,sk}* is the highest probability path from *Ci* to *Cj* on the graph. Due to the dynamic nature of the surveillance system, these conditional probabilities should change with time; $P_O(Cj,t|Ci) = P(Cj,t|C_{s1})P(C_{s1},t|C_{s2}...P(C_{sk},t|Ci)$. The conditional probabilities are eroded by time as $P_O(Cj,t|Ci) = k.P_O(Cj,t-1|Ci)$ where $k<1$ since the object may exit from the system completely. Here, we do not think a multi-camera system should be a closed graph. However, the conditional probabilities do not become less than a threshold $P_O(Cj,\infty|C_i) = 1/(M+1)$, which corresponds to the identical and independent nodes. Here, *M* is the number of cameras, and the addition is due to we treat the database as another node.

As a new object is detected, it is compared with the objects in the database and with the objects disappeared from a camera but still is not matched. The comparison is based on the color histogram similarity. For more than one object correspondence, we normalize each similarity score with the total of similarity scores of all possible pairs. By scaling these scores with the conditional probabilities, we select the correct match as the pair *(Om,On)* that maximizes $g*_{mn} P(Cm,t|Ci) P(Cn,t|Cj)$. To match objects between two cameras *Ci* and *Cj*, we evaluate the matching for all objects simultaneously instead of matching each single subset independently to minimize the matching conflicts.

3. OBJECT-BASED QUERY AND SUMMARIZATION

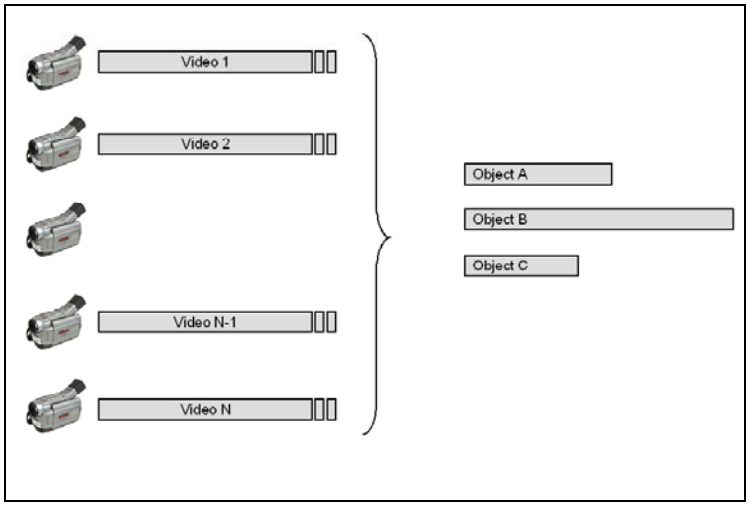

Figure #-6. Instead of camera-based representation, multiple videos can be restructured as object-based sequences

After matching the objects between the cameras, we label each video frame according to the object appearances. This enables us to include content semantics in the subsequent processes. A semantic scene is defined as a collection of shots that are consistent with respect to a certain semantic, in our case, the identities of objects. Since we have the position information of the cameras and we extracted which objects appeared in a which video at what time, we can, for instance, query for what an object did in a certain time period at a certain location. Thus, we are able to represent the video sequences with respect to the detected objects as illustrated in Fig.6.

To obtain concise and very low-bit representation of query results, we generate an abstract of the query result video sequence. Video abstraction can be done either by providing a ``preview'', which consists of a concatenation of key-video segments, or a set of key frames chosen from the frames that comprise the video sequence. The key-frame-based summary is a collection of frames that aims to capture all of the visual essence of the video except, of course, the motion. Making it ideal for rapid browsing of stored video, its constituent key frames can serve as pointers to the desired portions of the content.

A key frame generation technique based on a measure of the fidelity of a set of key frames is introduced in [2]. The fidelity measure is defined as the

Semi-Hausdorff distance between the set of key frames *S* and the set of frames *R* in the video sequences. A practical definition of the Semi-Hausdorff distance is as follows: Let the key frame set consist of S_{max} frames, and let the set of frames *R* contain R_{max} frames. Let the dissimilarity between two frames *Si* and *Ri* be *d(Si,Ri)*. We define *fi* for a frame *Ri* as $fi=min[d(Sk,Ri)],\ k=1..S_{max}$. Then the Semi-Hausdorff distance between *S* and *R* is given by $max[fi],\ i=1..R_{max}$. This way we end up finding out how well the key frame set *S* represents *R*, because the better the representation the lower the Semi-Hausdorff distance between *S* and *R*. For example, in the trivial case, if the *S* and *R* are identical, the Semi-Hausdorff distance is zero. On the other hand, a high Semi-Hausdorff distance indicates that at least one of the frames in *R* was not well represented by any of the frames in the key frame set *S*. As a frame dissimilarity metric, we proposed to use motion activity descriptor. The motion activity score of a frame is defined as the standard deviation of motion vector magnitude. By treating the motion activity scores of a video segment as a distribution function, we obtain a cumulative motion activity function. We quantize standard deviation of motion vectors of MPEG-1 video to classify segments into five classes ranging from very low to very high intensity. In [4], it is showed that the frame at which the cumulative motion activity is half the maximum value is also the halfway point for the cumulative increment in information. This implies that it would be the best choice for the first key frame since it would have the minimum Semi-Hausdorff distance. Being forced to pick the first frame as a key frame is disadvantageous, i.e. not all of the target object is visible, or it is very small in the first frame in comparison to rest of the segment. This motivates us to find a key frame based on motion activity that would be better than the first frame. Thus, we select the key frames according to the cumulative function as shown in Fig. 7.

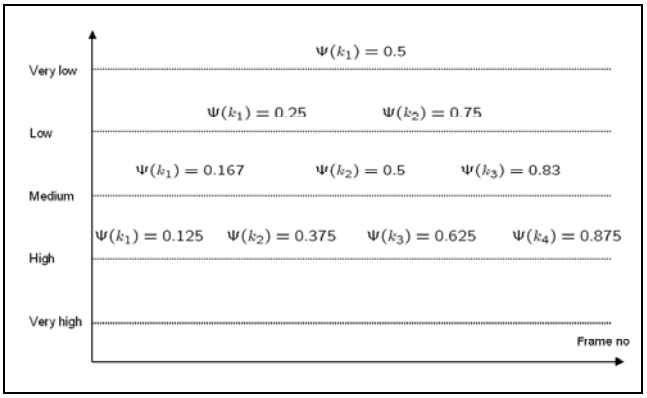

Figure #-7. Motion activity based key-frame extraction optimal strategy: Note that the middle of each segment is picked

For a single camera, tracking takes less than 28 milliseconds/frame on average for color video at 320x240 spatial resolution on a Pentium4, 1.8Ghz computer. A base station controls the fusion of the multi camera information, and its computational load is negligible. We presented the object-based inquiry user interface in Fig.8. The tracking method can follow several objects at the same time even some of the objects are totally occluded for a long time. Furthermore, it provides accurate object boundaries. We initialized the Bayesian Network with identical conditional probabilities. These probabilities may also be adapted by observing the long-term motion behavior of the objects. Sample query results are shown in Fig.9. We are able to extract all appearances of the same person in a specified time period accurately. We can also count the number of different people. The background generation method is computationally more feasible than the existing mixture models, and it can achieve real-time performance even for full resolution video owing to the new illumination change detection and reference image refresh mechanisms. The shadow removal method effectively filters most shadow pixels without breaking object regions apart. This method is robust towards the perturbations of the filter parameters, and it adapts easily for different lighting conditions. The performance of the background adaptation and mean-shift analysis based object tracking method is comparable with the state-of-art, and it is fully automatic. It does not have the intrinsic shortcomings of the template-matching approaches such as resolution, pose, and illumination dependencies. The object-based representation enables us to associate content semantics, thus we can generate query-based summaries. This is an important functionality to retrieve the target video segments from a large database of surveillance video. The motion activity based summarization is numerically and visually comparable with the existing techniques and relies on computationally simple motion feature extraction in the compressed domain, and is thus much simpler than other techniques. Using additional biometrics such as face, gait, and speech, may be necessary for long-time tracking scenarios where the color features of an object change. Face recognition is a possible solution for this problem. Currently, we are investigating robust and computationally feasible ways of integrating face features.

One storage space related challenge of the object-based representation arises when the object number is much greater than the number of cameras. In this case, instead of storing object-based video sequences, a conversion table that keeps the pointers from the camera-based video segments to the object-based video segments may be a better solution. The current tracking system is designed for stationary cameras. In the future, we consider improving the object tracking method so that it can handle pan-tilt-zoom cameras as well.

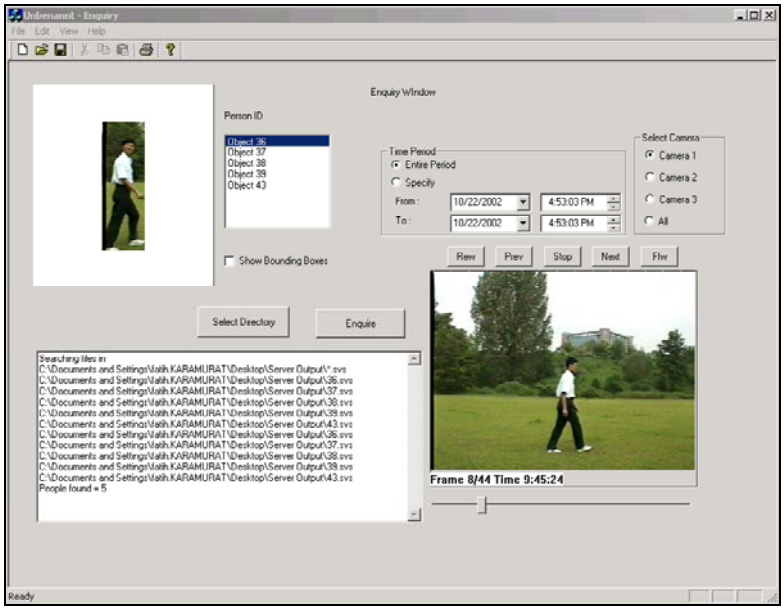

Figure #-5. Inquiry system that a user can specify the camera, time, and object to generate summary

Figure #-6. The retrieved instances of two objects in one camera; the person with white shirt at frames 10, 20, 24 (first row), 215, 224, 238 (second row), and another person with red shirt 142, 154, 392 (last row) on the same camera.

REFERENCES

1. T. H. Chang and S. Gong. ``Bayesian modality fusion for tracking multiple people with a multi-camera system". In Proc. European Workshop on Advanced Video-based Surveillance Systems, Kingston, UK, 2001

2. H. S. Chang, S. Sull, and S. U. Lee, ‘‘Efficient video indexing scheme for content-based retrieval,’’ IEEE Trans. Circuits Syst. Video Technol., 9(8), 1999.

3. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects sing mean shift, IEEE Conf. Computer Vision and Pattern Recognition, SouthCarolina, Vol. 2, 142-149, 2000.

4. A. Divakaran ``Video summarization using descriptors of motion activity: A motion activity based approach to key-frame extraction from video shots" Journal of Electronic Imaging, Vol. 10(4), 2001.

5. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," In Proc. of the Thirteenth Conf. on Uncertainty in Artificial Intelligence, 1997.

6. C. Stauffer and W.Grimson, "Adaptive background mixture models for real-time tracking", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, 1999.

7. T.Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection", Proc. of IEEE ICCV FRAME-RATE Workshop, 1999.

8. C. Ridder, O. Munkelt, and H. Kirchner, ``Adaptive background estimation and foreground detection using Kalman-filtering". Proc. of Int'l Conf. on recent Advances in Mechatronics, 193-199, 1995.

9. C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body". IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 780–785, July 1998.