10-2001

On clustering and retrieval of video shots

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Ting-Chuen PONG

Hong-Jiang ZHANG

Citation

NGO, Chong-wah; PONG, Ting-Chuen; and ZHANG, Hong-Jiang. On clustering and retrieval of video shots. (2001). *Proceedings of the 9th ACM International Conference on Multimedia, Ottawa, Canada, 2001 September 30 - October 5*. 51-60.

Available at: https://ink.library.smu.edu.sg/sis_research/6482

On Clustering and Retrieval of Video Shots

Chong-Wah Ngo
Department of Computer Science
The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
cwngo@cs.ust.hk

Ting-Chuen Pong
Department of Computer Science
The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
tcpong@cs.ust.hk

Hong-Jiang Zhang
Microsoft Research China
5/F, Beijing Sigma Center,
Zhichun Road,
Haidian District, Beijing
100084, PRC
hjzhang @ microsoft.com

ABSTRACT

Clustering of video data is an important issue in video abstraction, browsing and retrieval. In this paper, we propose a two-level hierarchical clustering approach by aggregating shots with similar motion and color features. Motion features are computed directly from $2D$ tensor histograms, while color features are represented by $3D$ color histograms. Cluster validity analysis is further applied to automatically determine the number of clusters at each level. Video retrieval can then be done directly based on the result of clustering. The proposed approach is found to be useful particularly for sports games, where motion and color are important visual cues when searching and browsing the desired video shots. Since most games involve two teams, classification and retrieval of teams becomes an interesting topic. To achieve these goals, nevertheless, an initial as well as critical step is to isolate team players from background regions. Thus, we also introduce approach to segment foreground objects (players) prior to classification and retrieval.

Keywords

Hierarchical clustering, Motion and color retrieval, Team classification

1. INTRODUCTION

Clustering is a natural solution to abbreviate and organize the content of a video. A preview of the video content can simply be generated by showing a subset of clusters or the representative frames of each cluster. Similarly, retrieval can be performed in an efficient way since similar shots are indexed under the same cluster. Regardless of these advantages, general solutions for clustering video data is a hard problem. For certain applications, motion features dominate the clustering results; for others, visual cues such as color and texture are more important. Moreover, for certain types of applications, decoupling of camera and object motions ought to be done prior to clustering.

Clustering algorithms can be grouped into two categories: partitional and hierarchical [3]. Hanjalic & Zhang [1] have introduced a partitional clustering of video data by utilizing the color features of selected keyframes. Here, we introduce a two-level hierarchical clustering algorithm by making use of both color and motion features. The proposed clustering algorithm is unsupervised, the number of clusters is determined automatically by the cluster validity analysis [3]. Supervised version of clustering algorithms by Hidden Markov Models can be found in [2, 6], and by decision rules can be found in [8].

In this paper, we focus on organizing the content of video. Our approach is based on the pattern analysis and processing of temporal slices. We utilize the tensor histogram introduced by Ngo *et.* al. [4] for motion feature extraction and foreground object segmentation. By incorporating motion and color features, we further propose a two-level hierarchical clustering algorithm to organize and index the content of video. The constructed clustering structure inherently provides an indexing scheme for video retrieval. We hence study and investigate the fundamental difference and the effectiveness of retrieving with (cluster-based) and without (cluster-free) clustering structure.

Although the experiments are performed in the sport video domain, no specific domain knowledge is being utilized. We demonstrate that these videos are well represented as a two-level hierarchy. The top level is clustered by color features while the bottom level is clustered by motion features. The top level contains various clusters including wide-angle, medium-angle and close-up shots of players from different teams[1]. Each cluster can refer to a sport event. For instance, the wide-angle shots of a basketball video usually correspond to full court advances, while the wide-angle shots of a soccer video normally correspond to bird view scenes. The shots inside each cluster can be further partitioned according to their motion intensity. In this way, for example, the sub-cluster of a close-up shot can correspond either to “players running across the soccer field”, or “players standing on the field”. Such organization facilitates not only

[1] The classification of wide-angle, medium-angle and close-up shots are roughly based on the distance between the camera lens and the targeted scene.

MM'01, Sept. 30-Oct. 5, 2001, Ottawa, Canada.
Copyright2001 ACM 1-58113-394-4/01/0009...$5.00

video browsing and retrieval. but also some high-level video processing tasks. For instance, to perform player recognition. only those shots in the cluster that correspond to close-up shots of players are picked up for processing. To perform motion-based background reconstruction. the sub-cluster corresponds to “players running across the soccer field” is further selected for processing.

An advantage of working in the sport video domain is that thr camera motion is usually smooth and restricted to pan-tilt. stationary and zoom *This is because sport videos* are usually captured with several fixed cameras that are located in the stand. Since the camera motion is smooth. motion-based video segmentation can usually be performed in a robust way. In this paper we employ the foreground and background segmentation algorithm introduced by Ngo et. *al.* [4] to extract the players and subsequently perform team classification. We demonstrate that team classification with player segmentation can offer significant improvement over that of without player segmentation.

The paper is organized as follows. Section 2 gives a brief introduction on the motion pattern analysis and segmentation of video shots in temporal slices. These methods are further utilized in Section 3 for motion. color and object features extraction. Subsequently, Section 4 proposes a two-level hierachical clustering algorithm by motion and color features. Section 5 investigates the Fundamental difference of cluster-based and cluster-free retrieval. Section 6 presents our experimental results on clustering, retrieval and team classification of sport videos. Section 7 concludes our proposed works.

2. PROCESSING OF MOTION PATTERNS IN TEMPORAL SLICES

If we view a video as an image volume with (x, y) image dimension and t temporal dimension, the spatio-temporal slices are a set of $2D$ images in a volume with one dimension in t. and the other in x or y, for instance. One example is given in Figure 1; the horizontal axis is t. while the vertical axis is x. For our application, we process all slices. both horizontal and vertical. in a volume to analyze thr spatio-temporal patterns due to various motions. In addition. to reduce the computational and storage overhead. all processing are done directly in MPEG domain.

4 spatio-temporal slice, by first impression. is composed of color and texture components. On one hand, the discontinuity of color and texture infers the occurrence of a new event: on the other hand, the orientation of texture depicts camera and object motions. By processing the patterns in slices, motion analysis and segmentation can be done in a more effective and efficient wav. Figure 1 shows a spatio-temporal slice extracted from a video composed of three shots. As seen from the figure, camera motion can be inferred directlv from the texture pattern. For instance. thr slanted lines in shot B depict camera panning. and the lines which expand in a V-shape in shot, D depict zooming. In addition, multiple motions can also be perceived when two dissimilar texture patterns appeared in a shot. as shown in shot C. In this shot. the middle region describes a non-rigid object motion, while the background region indicates camera panning.

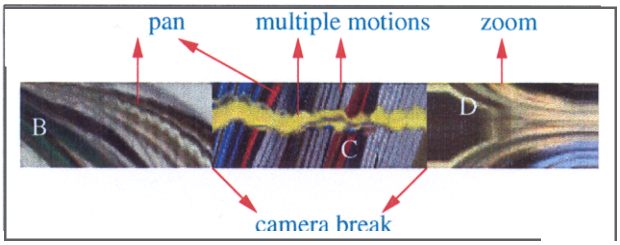
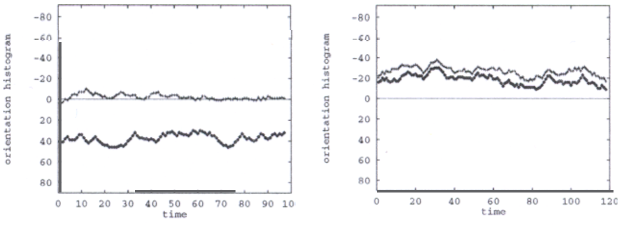


Figure 1: Patterns in a temporal slice.



(a) moving object (b) parallax panning

Figure 2: Motion trajectories in the tensor histograms.

2.1 Motion Analysis

We utilize the $2D$ tensor histogram approach [4] (see Appendix A) to extract the texture patterns from temporal slices for clustering and retrieval. A tensor histogram $M(\phi, t)$ with one dimension as an $1D$ orientation histogram and the other dimension as time, can be employed to model the distribution of texture orientations in slices. This distribution inherently reflects the motion trajectories in an image sequence. two examples are given in Figure 2. The trajectories in the figure are the histogram peaks tracked over *time.* In Figure 2(a) one trajectory indicates a non-stationary background, and the other indicates objects moving to the right: in Figure 2(b) two trajectories progress in a similar manner, they correspond to parallax motion (or camera panning).

2.2 Motion Segmentation

Motion segmentation can be achieved b y back-projecting the motion trajectories in a tensor histogram to temporal slices to form spatially separated motion layers. Each motion layer is directly associated with a layer of support. Figure 3 illustrates the major flow of this idea. Given a set of spatio-temporal slices. a $2D$ tensor histogram is computed. The $2D$ histogram is further non-uniformly quantized into a $1D$ normalized motion histogram to qualitatively represent the rigid camera and object motions. The peak of the histogram is then hack-projected onto the original image sequence. and is assumed as a background layer. As shown in the figure, the background panoramic image can be further reconstructed by aligning and pasting the projecting pixels. Thr foreground object, on the other hand. has a support layer which is the inverse of background support layer. The exact boundary of a foreground object can be located in a more precise way by the background subtraction technique and the color back-projection of the suspected foreground region (see Appendix B). For robustness, a segmented foreground object is represented by a probability map $\Pr(X = \text{Foreground}.\ t)$ which indicates the probability

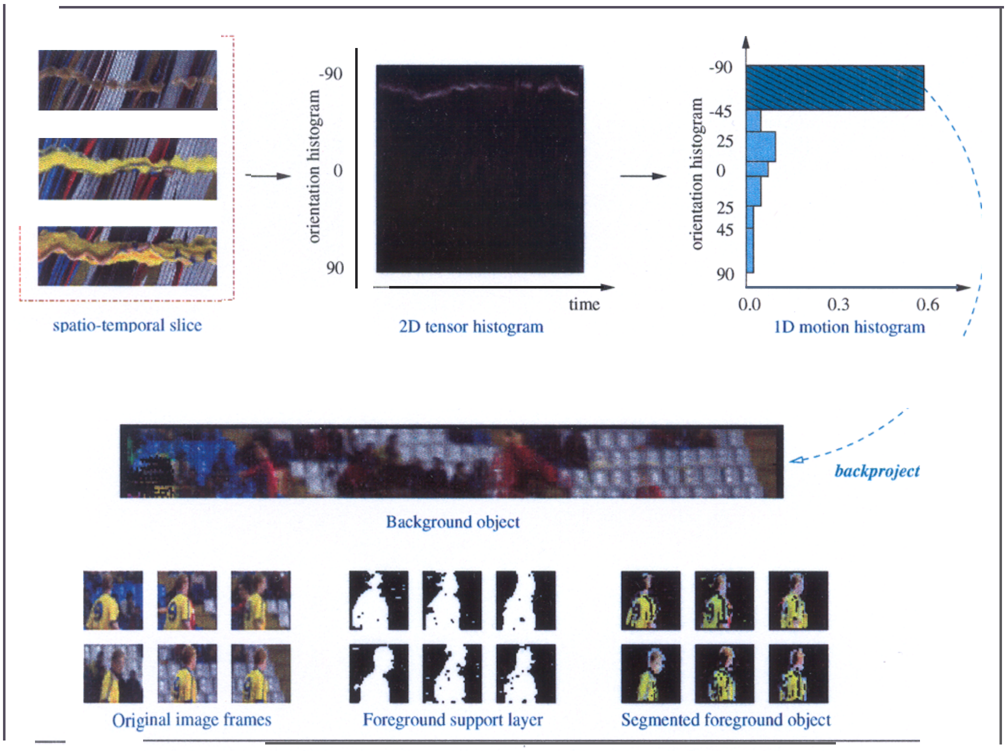


Figure 3: The scheme for foreground segmentation

of a pixel $\mathbf{I}(X,$ t) belongs to a foreground object.

3. FEATURE COMPUTATION

We present methods to extract motion and color features directly from shots. Roth motion and color features are represented as histograms, since this representation is effective and inexpensive for clustering and retrieval.

3.1 Motion Feature

Given a $2D$ tensor histogram $M(\phi, t)$ of a shot composed of n frames, the motion feature vector $\mathcal{M}$ which is composed of N feature points. is computed by

$$\mathcal{M}(k) = \frac{1}{n}\{\sum_{\phi'} \sum_{t=1}^{n} M(\phi', t)\} \quad \forall_{\phi'} \{\mathcal{Q}(\phi') = k\} \qquad (1)$$

where $\mathcal{Q}(\phi') =, \frac{|\phi|}{s}$ is a quantization function. $s = \frac{90^o}{N}$ is the step of quantization, and $k = \{1, 2, . , N$ I} represents a quantized level. Thr computed motion features describe the motion intensity of a shot. Since $|\phi \ \geq 0$, the motion feature is directionless. whether to encode the directional information to motion vector is tailored to applications. In our experiment. N is set to 10, and the tensor histograms of both horizontal and vertical slices are used for feature computation.

3.2 Color Feature of au Image Volume

The color feature of a shot, represented as a $3D$ color histogram. is computed directly in the YUV color space of its DC image sequence. A color histogram describes the global color distribution in a shot. It is easy to compute and is insensitive to small changes in viewing positions and partial occlusion. As a feature vector for clustering and retrieval, it is susceptible to false alarms. In our experiments, each color channel of YUV is uniformly quantized to four bins. results in a 64 dimensional color feature vector. Each color histogram will be further normalized by the number of frames in a shot.

3.3 Color Feature of a Segmented Object

Thr segmented object of an image frame I is associated with a support layer, Mask, and a probability map, Pr. The color histogram is computed as

$$\text{C(k)} = \frac{1}{T}\{\sum_{t=1}^{T} \sum_{\text{a}} \mathbf{Pr}(X, t)\} \qquad (2)$$

where $\Delta \in \{\mathcal{Q}(\mathbf{I}(X,t)) = k \ A \ Mask(X,t) = 1)$. and $\mathcal{Q}$ is a color quantization function. Meanwhile, *k is an index lo* a quantized value. X is an index to a pixel location, and t is an index to time. The histogram is computed directly in the YUV color space. and each color channel is uniformly quantized into four bins.

3.4 Distance Measure

The L_1 and L_2 norms are *two* of the most frequently used distance metrics for comparing two feature vectors. In practice. however, L_1 *norm* performs better than L_2 norm since it is more robust to outliers [5]. Furthermore. L_1 *norm is* more computationally efficient and robust. The distance between two non-zero length feature vectors 3 and 3' is computed as.

$$D(\mathcal{F}, \mathcal{F} \ ) = \frac{1}{Z(\mathcal{F}, \mathcal{F}')}\{\sum_{i=1}^{n} \ |\mathcal{F}(i) \ \ \mathcal{F}'(i)|^k\}^{\frac{1}{k}} \qquad (3)$$

where

$$Z(\mathcal{F},\mathcal{F}') = \sum_{i=1}^{n} \mathcal{F}(i) + \sum_{i=1}^{n} \mathcal{F}'(i) \quad (4)$$

is a normalizing function. In Eqn (3), k = 1 for L_1 norm and k = 2 for L_2 norm.

4. HIERARCHICAL CLUSTERING

We employ a two-level hierarchical clustering approach to group shots with similar color and motion. The algorithm is implemented in a top-down fashion, where color features are utilized at the top level, while motion features are used at the bottom level. At the top level, the color feature space is partitioned to k_c clusters. At the bottom level, each k_c cluster is further partitioned into k_m clusters.

4.1 K-Mean Algorithm

The k-mean algorithm is the most frequently used clustering algorithm due to its simplicity and efficiency. The algorithm is employed to cluster shots at each level of hierarchy independently. The k-mean algorithm is implemented as

- Step 1: Choose $\mu_1, \mu_2, \ldots, \mu_k$ as initial cluster centroids.
- Step 2: Classify each feature $\mathcal{F}$ to the cluster $\hat{p}$ with the smallest distance

  $$\hat{p} = \arg_{1\leq j\leq k} \min D(\mathcal{F}, \mu_j) \quad (5)$$

- *Step* 3: Based on the classification, update cluster centroids as

  $$\mu_j = \frac{1}{n_j}\sum_{i=1}^{n_j} \mathcal{F}_i^{(j)} \quad (6)$$

  where n_j is the number of shots in cluster j, and $\mathcal{F}_i^{(j)}$ is the i^{th} feature vector in cluster j.

- Step 4: If any cluster centroid changes value, goto Step 2; otherwise stop.

The algorithm is suboptimal. The initial guess of cluster centroids in *Step* 1, and the order in which feature vectors are classified can affect the clustering result. To diminish the sensitivity of *Step 1,* the initial centroids are selected in the following way:

- *Step* 1: Given n m-dimensional feature vectors, divide the m dimensions to p = $\frac{m}{k}$ subspaces. These subspaces are indexed by $[1, 2, \ldots, p], [p+1, p+2, \ldots, 2p]$, . . , $[(k-1)p, (k-1)p+1, \ldots, kp]$.
- Step 2: In each subspace j of $[(j-1)p, \ldots, jp]$, associate a value f_i^j for each feature vector $\mathcal{F}_i$ by

  $$f_i^j = \sum_{m=(j-1)p}^{jp} \mathcal{F}_i(m) \quad (7)$$

- *Step* 3: Choose the initial cluster centroids μ_1, μ_2, , μ_k by

  $$\mu_j = \arg_{\mathcal{F}_i} \max_{1<i<n} f_i^j \quad (8)$$

4.2 Cluster Validity

The number of clusters k needs to be explicitly specified for the k-mean algorithm. However, in most cases, k is not exactly known in advance. In order to find an optimal number of clusters, we have employed the cluster validity analysis [3]. The intuition is to find clusters that minimize intra-cluster distance while maximize inter-cluster distance. The cluster separation measure $p(k)$ is defined as

$$\rho(k) = \frac{1}{k}\sum_{i=1}^{k} \max_{1\leq n\leq k}\{\frac{\eta_i + \eta_j}{\xi_{ij}}\} \quad (9)$$

where

$$\eta_j = \frac{1}{n_j}\sum_{i=1}^{n_j} D(\mathcal{F}_i^{(j)}, \mu_j) \quad (10)$$

$$\xi_{ij} = D(\mu_i, \mu_j) \quad (11)$$

η_j is the intra-cluster distance of cluster j, while ξ_{ij} is the inter-cluster distance of clusters i and j. In the experiments, the optimal number of clusters $\hat{k}$ is selected as

$$\hat{k} = \min_{1\leq k\leq 10} \rho(k) \quad (12)$$

In other words, the k-mean algorithm is tested for $k = \{1, 2, \ldots, 10\}$, and the one which gives the lowest value of $p(k)$ is chosen.

4.3 Similarity Matrix

We can readily incorporate a similarity matrix Sim at each level of hierarchy to describe the similarity, i.e., the inverse of inter-class distance, between clusters i and j. For n clusters, we have

$$\mathcal{S}im = \begin{bmatrix} 1 & 1-\xi_{12} & \ldots & 1-\xi_{1n} \\ 1-\xi_{21} & 1 & \ldots & 1-\xi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 1-\xi_{n1} & 1-\xi_{12} & \ldots & 1 \end{bmatrix} \quad (13)$$

which is a symmetrical matrix. The matrix Sim can facilitate the retrieval process. To browse shots with a similar motion to cluster i, one can just sort the entries in the i^{th} row of Sim in descending order, and then traverse the links accordingly.

5. RETRIEVAL

Given a query represented by low-level features, motion and color in our case, a retrieval system returns a set of items sorting in ascending order according to their respective distances to the query. This is normally referred to as a k nearest neighbor (KNN) search problem, which is actively studied by both computational geometry and multimedia retrieval communities. By coupling clustering issues with retrieval problems, the clustering structure, on one hand, inherently provides an indexing scheme for retrieval, while on the other hand, intuitively speed up the retrieval time. We refer to this issue as a cluster-based retrieval problem. The fundamental difference between cluster-based and cluster-free retrieval are illustrated in Figure 4. Suppose a query is located at the boundary of two classes, cluster-free retrieval will include items from both classes, i.e., items inside the dotted circle in Figure 4, at the top of a ranked list. In contrast, clustered-based retrieval compares the distance between the query and each cluster centroid, and ranks the

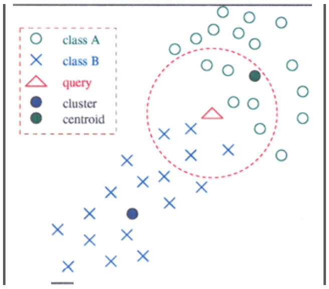


Figure 4: When a query locates at the boundary of two classes, the retrieval results of the cluster-based and cluster-free approach will be quite **different.**

items whose cluster centroid is nearer to the query at the top of a list,.

5.1 Cluster-based Retrieval

In a hierarchical clustering structure, a centroid at the top level represents the color characteristics of a cluster, while a centroid at the bottom level represents the motion characteristics of a cluster. During retrieval. cluster centroids at the top level of hierarchy are first compared with the color feature of a query. A cluster with the nearest centroid is first located. Then. its sub-clusters in the bottom level are further compared with the query. Thr itrms in one of these sub-clusters whose centroid is the nearest to the motion feature of the query. are sorted in ascending order of their distance to the query, and put accordingly at the top of a ranked list. The retrieval is processed in a depth-first-search-like manner, meaning that after all sub-clusters o f the most similar cluster are sorted, the next similar clustei is handled in the same way. ‘This process is repeated until the few most similar or all clusters are visited.

5.2 Cluster-Free Retrieval

If a clustering structure is not available, WC ran merge thr ranked lists given by both motion and color features. A straightforward way is to linearly weight the distance measures given by both features. Denote $< \mathcal{M},\mathcal{C} >$ and $< \mathcal{M}^{'}.\mathcal{C}^{'} >$ as two pairs of motion and color feature vectors, we have

$$\mathrm{D}(< \mathcal{M},\mathcal{C} >,< \mathcal{M}^{'}.\mathcal{C}^{'} >) = \alpha_{\mathcal{M}} D(\mathcal{M},\mathcal{M}^{'}) + \alpha_{\mathcal{C}} D(\mathcal{C},\mathcal{C}^{'})$$

where $\alpha_{\mathcal{M}}$ and $\alpha_{\mathcal{C}}$ are weights. and $\alpha_{\mathcal{M}} + \alpha_{\mathcal{C}} = 1.0$. ‘lb equally weight both features. we set $\alpha_{\mathcal{M}} = \alpha_{\mathcal{C}} = 0.5$.

6. EXPERIMENTS

To test thr effectiveness of the proposed clustering and retrieval approach. WC conduct experiments on both the hasketball and soccer video. The retrieval performance is evaluated in terms of recall and precision where

$$recall = \frac{\text{number of relevant shots retrieved}}{\text{total number of relevant shots in database}}$$

$$precision = \frac{\text{number of relevant shots retrieved}}{\text{total number of shots retrieved}}$$

Recall measures the ability to present all relevant items, while *precision* measures the ability to present only relevant items. *Recall* and *precision* are in the interval of $[0, 1]$. The recall-precision curve indicates a system's ability in ranking the relevant items. Ideally, precision values should bc equal to one across all recall values.

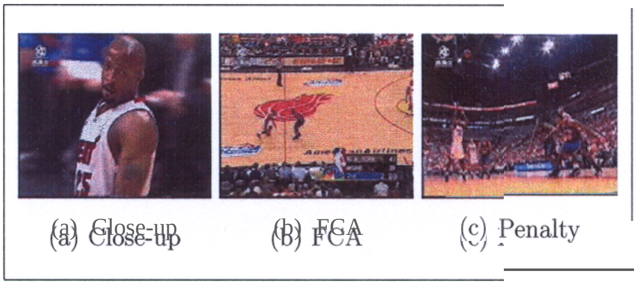

(a) Close-up (b) FCA (c) Penalty

Figure 5: Sample shots in the tested basketball video.

6.1 Basketball Video

The tested basketball video composed of 122 shots, approximately *24,000* frames. Most of the shots can be categorized under close up shots, full court advances (FCA) and penalty shots, as shown in Figure 3. Close up shots are normally of short duration, with the camera tracks a player from frame to frame. FCA are usually captured by a mid-court camera. The camera is panned towards the direction when the ball is being advanced from one end of the court to the other. Thr camera of penalty shots is usually tilted up at the moment when the hall is shot.

6.1. I *Clustering*

The tested video is first partitioned into shots and then a $2D$ tensor histogram is computed for each shot. For clustering, the motion and color features are extracted. respectively, from thr $2D$ *tensor* histograms and image volumes of shots. Figure 6 depicts thr clustering results by using L_1 norm as the distance measure. Shots that arc nearest to cluster centroids are shown in the figure to represent clusters. The top level has four clusters, while the bottom level consists of nine clusters.

By manual investigation of the clustering results. we summarize thr characteristics of each cluster as follows:

- Cluster *c.1* mostly consists of penalty shots, audience scene and few close up shots.
- Cluster *c.2* basically consists of players from both teams Thr players can not he classified according to their trams due to thr unsegmented cluttered background.
- Cluster *c.3* has mainly the FCA shots. These shots are grouped accordingly in the sub-cluster *m.3.1* by motion features.
- Cluster *c.4* consists of the logo sequence which is appeared prior to the replay of slow motion sequence.

6.1.2 Retrieval

‘The performance of both the cluster-based and the cluster-free retrieval approach is investigated. For the cluster-free retrieval. we examine also the retrieval by motion feature. retrival hy color feature, and retrieval by both color and motion features. In addition. for all the tested approaches, we

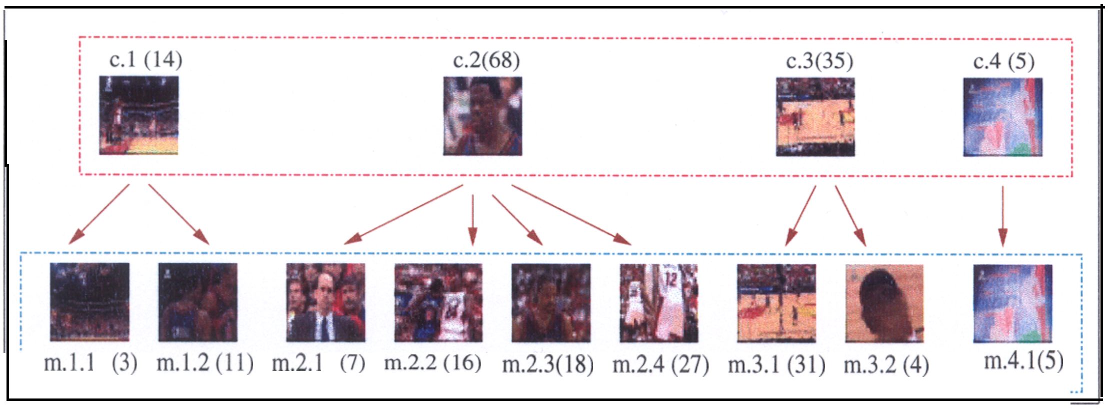


Figure 6: Clustering result of basketball video by using L_1 **norm** as distance measure. $X(Y)$: X is cluster label and Y is the number of shots in a cluster.

Table 1: Mean precision of **different** retrieval **approaches.**

Approach	Mean Precision	
	I,, norm	L_2 norm
Cluster-based Retrieval	0.53	0.50
Retrieval by motion and color	0.53	0.48
Retrieval by motion	0.53	0.53
Retrieval by color	0.43	0.42

investigate the retrieval performance of employing L_1 norm and L_2 norm as distance measure. For cluster-based retrieval, both clustering structures by L_1 norm and L_2 norm are constructed for retrieval.

Thr tested query set consists of 25 queries which are manually picked and checked to have good answers. They consist of close up of players from different teams, FCA, penalty and shooting shots. Players from different teams are placed in different classes. Similarly, players captured by different camera motions (e.g., track and zoom) are put in **different** classes

Figure 7 shows the recall-precision of the tested approaches hy using L_1 norm as distance measure, while Figure 8 shows the recall-precision hy using L_2 norm as distance measure. Table I summarizes the 11-point average precision values of various approaches. As indicated from thr experimental results. the retrieval performance based on L_1 norm distance measure is more robust and consistent than L_2 norm. Throughout the experiments. motion features alone give better retrieval performance compared with color features. For L_1 norm. by incorporating both color and motion features for retrieval. the retrieval precision does not show improvement compared to employing motion features alone. In addition. the performance of cluster-based and cluster-free retrieval based on L_1 *norm does not show* significantly different results. This may indicate that the feature space of basketball video is well separated among the categories of close up shots, FCA and penalty shots. This is not surprised since visually these categories have different color and motion content.

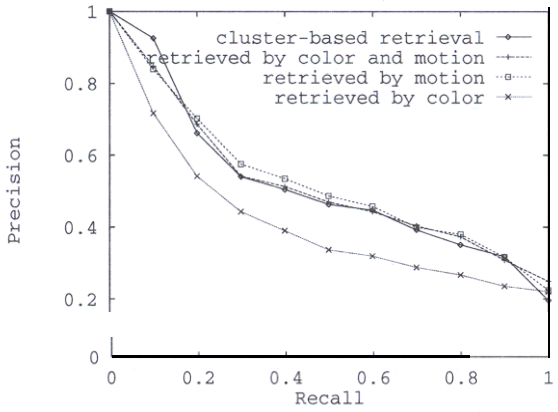


Figure 7: Recall and precision curves for basketball video (with L_1 norm as the distance measure).

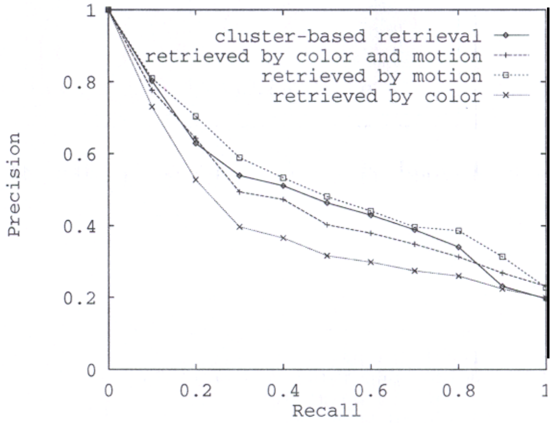


Figure 8: Recall and precision curves for basketball video (with L_2 norm as the distance measure).

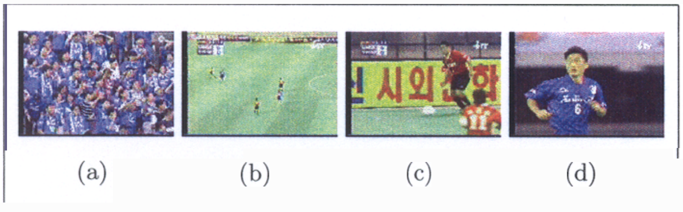


Figure 9: Sample shots in the tested soccer video: (a)audience; (b) bird view; (c) medium shot; (d) close-up.

6.2 Soccer Video

The tested video is composed of 404 shots, approximately 100.000 frames. Most of the shots can he categorized under audience. bird view. medium, and close up shots. as shown in Figure 9. For simplicity, we refer to thr two soccer teams as team A and team B. The color of thr audiences' clothing is same as thr players whom they support.

6.2. I Clustering

The tested video is first partitioned into shots and then a $2D$ tensor histogram is computed for each shot. For clustering, the motion and color features are extracted, respectively, from the $2D$ tensor histograms and image volumes of shots. Figure 10 depicts the clustering results hy using L_1 norm as thr distance measure. Shots that are the nearest to cluster centroids are shown in the figure to represent clusters. The top level has six clusters, while the bottom level consists of thirteen clusters.

By manual investigation of the clustering results, we summarize the characteristics of each cluster as follows:

- Cluster *c.1* mostly consists of players and audiences of team A, coaches of hoth teams, referees. and shots of players being hurt accidentallv. The audiences of team A are all clustered in the sub-cluster *m.1.2*. Meanwhile, the player tracking shots are inch&d in thr sub-cluster *m.1.1*.

- Cluster *c* 2 basically consists of players from both teams. with more player from team B. The audience of team B are all clustered in m2.1. The sub-cluster *m.2.2* consists of snapshots of players, while the sub-clustei *m.2* 3 comprises players being tracked in the soccer field.

- Cluster *c.3* has only two shots which are shown prior to the start of the soccer game.

- Cluster *c.4* has mainly the bird vlew of the soccer game. In *m 4.1*, the camera motion is stationary; in *m.4.2*, the camera pans to thr left and to the right when one tram attacks the other: m.4.3 includes the bird view of shooting shots are included.

- Cluster *c.* 5 has mostly the medium shots of players passing the ball around. The camera motion in *m.5.1* is stationary. while the camera motion in *m.5.2* tracks the players when ditching the ball and faring opponents.

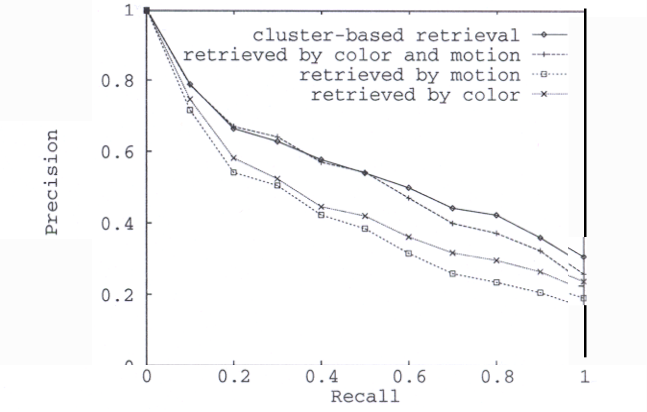


Figure 11: Recall and precision curves for soccer video (with L_1 norm as distance measure).

- Cluster *c.6* has one shot screening the sky. This shot is different from others in terms of color and content.

6.2.2 Refried

The performance of both the cluster-based and the cluster-free retrieval approach is investigated. For the cluster-free retrieval, we examine also the retrieval by motion feature. retrieval by color feature. and retrieval by both color and motion features. In addition, for all the tested approaches, we investigate thr retrieval performance of employing L_1 norm and L_2 norm as distance measure. For cluster-based retrieval. both clustering structures hy L_1 norm and L_2 norm are constructed for retrieval.

The tested query set consists of 53 queries. These queries have been manually picked and checked to have good answers. They consist, of close up of players from different teams. bird view and medium shots. audience from different teams. and shooting shots. Players and audience from different teams are placed in different classes. Similarly. shots with different camera motions are put in different classes. Thus, the effectiveness of discriminating shots hy color and motion can be experimented.

Figure 11 shows the recall-precision of the tested approaches by using L_1 norm as distance measure. while Figure 12 shows thr recall-precision by using L_2 norm as distance measure. Table 2 summarizes the ll-point average precision values of various approaches. Similar to the experiment on the basketball video, L_1 norm *is* superior to L_2 norm in term of mean precision. The main results based on L_1 norm are: retrieval by both color and motion features is constantly superior to retrieval by either one feature: the recall of cluster-based retrieval is better than that of cluster-free retrieval.

6.2.3 Clustering and Retrieval of Team Players

Clustering and retrieval of players by color features is an interesting topic: however, a perfect clustering requires a good segmentation tool to discriminate the players from the background.

In this experiment, we manually select 32 close up shots of players from both teams. each forms a class and has 16 shots.

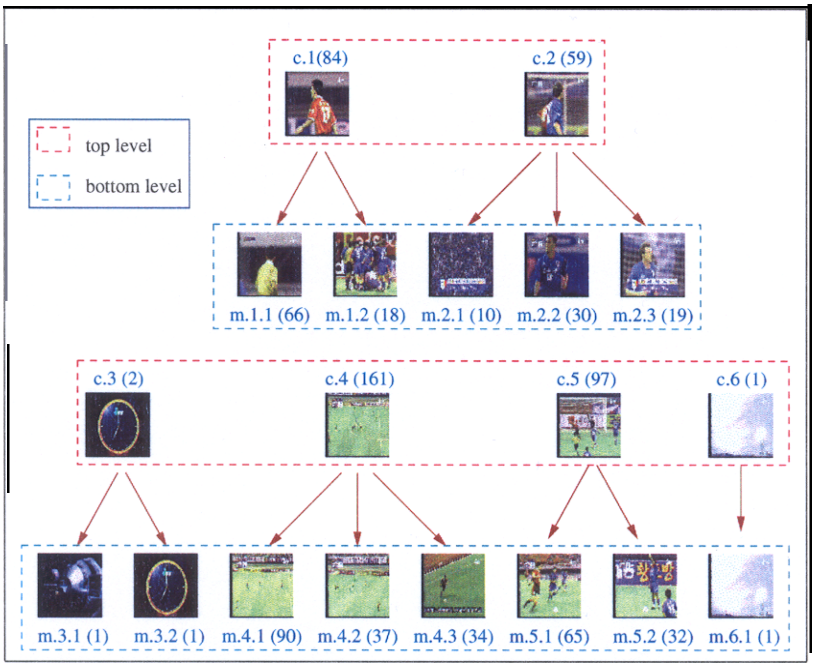


Figure 10: Clustering result of soccer video by using L_1 norm as distance measure. X(Y): X is cluster label and Y is the number of shots in a cluster.

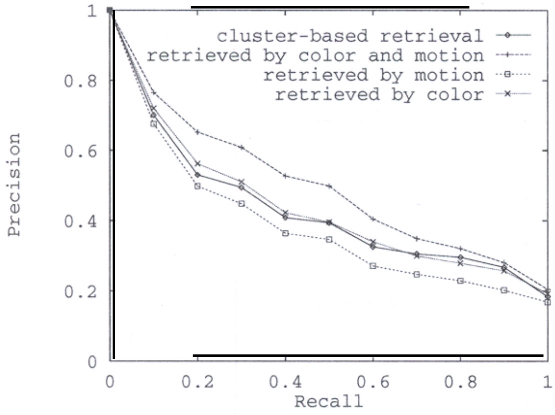


Figure 12: Recall and precision curves for soccer video (with L_2 norm as distance measure).

Table 2: Mean precision of different retrieval approaches.

Approach	Mean Precision	
	L_1 norm	L_2 norm
Cluster-based Retrieval	0.56	0.44
Retrieval by motion and color	0.53	0.51
Retrieval by motion	0.43	0.40
Retrieval by *color*	0.47	0.45

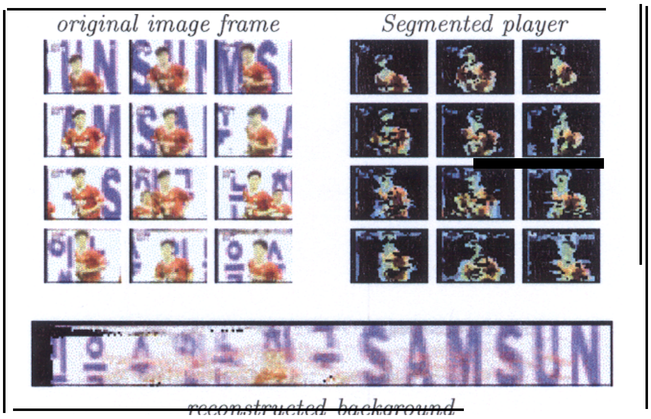


Figure 13: Result of player segmentation.

The foreground and background segmentation approach introduced in Section 2.2 is adopted to compute foreground images 3. before the k-mean algorithm ($k = 2$ in this case) is employed to cluster the players. One example of foreground (player) segmentation result is given in Figure 13. Notice that instead of employing L_1 or L_2 norm as feature distance measure, we use histogram intersection [7] For similarity measure since this measure can reduce the distraction of background noise after segmentation.

The classification rate[2], in term of confusion matrix, is shown in Table 3; meanwhile the classification performance of with

[2] No training is involved. The two clusters formed by the *k-mean* algorithm are checked to decide which team they belong to. For instance, if the number of team-A players in a cluster is more than that of team-B. the cluster belongs to class team-A. The classification rate is then computed accordingly.

with player segamentation

	Team-A	Team-B
Team-A	0.875%	0.125%
Team-B	0.125%	0.875%

without player segamentation

	Team-A	Team-B
Team-A	0.5625%	0.4375%
Team-B	0.25%	0.75%

Table 3: Confusion matrices of team classification.

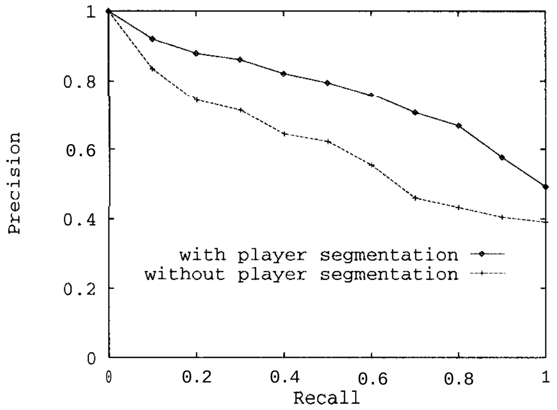


Figure 14: Recall and precision curves for player retrieval).

and without player segmentation is compared and contrast. In addition, the tested shots are also used as queries to investigate retrieval performance. The recall-precision of with and without player segmentation is given in Figure 14. The mean precision of the former approach is 0.77, while the latter is 0.61. As indicated by the experiment, foreground and background segmentation has offered significant improvement on the clustering and retrieval of players.

6.3 Speed Efficiency

Table 4 compares the motion and color features in term of the feature extraction time per image frame, and the feature vector length. The DC image size is 30 x 44. For the basketball video (122 shots), the clustering algorithm takes about 74 second to form a two-level hierarchical structure, while for the soccer video (404 shots), the algorithm takes approximately 861 sec (14.35 min). Table 5 further shows the average retrieval speed of 400 queries by the four tested approaches in the soccer video database. Cluster-based retrieval approach is about one time faster than of cluster-free approach (retrieval by motion and color features).

Table 4: Performance of motion and color features (on a Pentium III platform).

	Motion feature	Color feature
Feature extraction (sec)	0.072	0.0054
Feature vector length	18	64

Table 5: Retrieval speed on a database of 404 shots (on a Sun Spare Ultra-l machine).

Approach	speed (sec per shot)
Cluster-based Retrieval	0.29
Retrieval by motion and color	0.57
Retrieval by motion	0.30
Retrieval by color	0.31

7. CONCLUSIONS

We have described the issues of clustering and retrieval for video abstraction and browsing through the extraction of motion features from tensor histograms and color features from image volumes. We have experimented the proposed two-level hierarchical clustering algorithm with cluster validity analysis, and the cluster-based as well as cluster-free retrieval methods. In general, the proposed approaches are found to be suitable particularly for sport games where motion plays a critical roles in conveying the sport events. In the experiments, our clustering algorithm can successfully classify the content of basketball and soccer videos. Nevertheless, it is expected that player segmentation and hand-crafted domain specific knowledge will further improve the classification results. For retrieval, cluster-based approach in general gives slightly better results than that of cluster-free approach. Experimental results also indicate that player segmentation can offer significant improvement for team classification and retrieval.

8. REFERENCES

[1] A. Hanjalic and H. J. Zhang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE* Trans. on *Circuits* and *Systems* for *Video Technology,* 9(8):1280–1289, 1999.

[2] G. Iyengar and A. B. Lipman. Models for automatic classification of video sequences. In *Proc. SPIE* Storage *and Retrieval* for *Image and Video Databases VI,* pages 3312-3334, 1998.

[3] A. K. Jain. *Algorithms for Clustering* Data. Prentice Hall, 1988.

[4] C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin. Motion characterization by temporal slice analysis. In Computer *Vision and Pattern* Recognition, volume 2, pages 768-773, 2000.

[5] P. J. Rousseeuw and A. *M.* Leroy. *Robust* Regression *and Outlier Detection.* John Wiley and Sons, 1987.

[6] E. Sahouria and A. Zakhor. Content analysis of video using principle components. *IEEE* Trans. on *Circuits and Systems* for *Video* Technology, 9(8):1290–1298, Dec 1999.

[7] M. J. Swain and D. H. Ballard. Color indexing. *Int.* Journal *of Computer Vision,* 7(1):11–32, 1991.

[8] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE* Trans. on *Circuits* and Systems for *Video Technology,* 10(1):133–146, Feb 2000.

APPENDIX

A. TENSOR HISTOGRAM

Tensor histogram encodes the distribution of local orientation in temporal slices. It is computed based on the structure tensor to estimate the orientations of slices. The structural tensor Γ of slice H can be expressed as

$$\Gamma = \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix} = \begin{bmatrix} \sum_w \mathbf{H}_x^2 & \sum_w \mathbf{H}_x\mathbf{H}_t \\ \sum_w \mathbf{H}_x\mathbf{H}_t & \sum_w \mathbf{H}_t^2 \end{bmatrix} \quad (14)$$

where $\mathbf{H}_x$ and $\mathbf{H}_t$ are partial derivatives along the spatial and temporal dimensions respectively. The window of support w is set to 3 x 3 throughout the experiments. The rotation angle θ of I' indicates the direction of a gray level change in w. Rotating the principle axes of Γ by θ, we have

$$\mathbf{R}\begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix}\mathbf{R}^T = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_t \end{bmatrix} \quad (15)$$

where

$$\mathrm{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

From (15), since we have three equations with three unknowns, θ can be solved and expressed as

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mathbf{J}_{xt}}{\mathbf{J}_{xx} - \mathbf{Jtt}} \quad (16)$$

The local orientation ϕ of a w in slices is computed as

$$\phi = \begin{cases} \theta - \frac{\pi}{2} & \theta > 0 \\ \theta + \frac{\pi}{2} & \text{otherwise} \end{cases} \qquad \phi = [-\tfrac{\pi}{2}, \tfrac{\pi}{2}] \quad (17)$$

It is useful to add in a certainty measure to describe how well ϕ approximates the local orientation of w. The certainty c is estimated as

$$c = \frac{(\mathbf{J}_{xx} - \mathbf{J}_{tt})^2 + 4\mathbf{J}_{xt}^2}{(\mathbf{J}_{xx} + \mathbf{J}_{tt})^2} = (\frac{\lambda_x \quad \lambda_t}{\lambda_x + \lambda_t})^2 \quad (18)$$

and c = [0, 1]. For an ideal local orientation, c = 1 when either λ_x = 0 or λ_t = 0. For an isotropic structure i.e., $\lambda_x = \lambda_t$, c = 0.

The distribution of local orientations across time inherently reflects the motion trajectories in an image volume. A $2D$ tensor histogram $\mathbf{M}(\phi, t)$ with the dimensions as an 1D orientation histogram and time respectively, can be constructed to model the distribution. Mathematically, the histogram can be expressed as

$$\mathbf{M}(\phi, t) = \sum_{\Omega(\phi,t)} c(\Omega) \quad (19)$$

where $\Omega(\phi,t) = \{\mathbf{H}(x,t) | \Gamma(x,t) = \phi\}$ which means that each pixel in slices votes for the bin $(\phi$, t) with the certainty value c.

B. FOREGROUND OBJECT DETECTION

We introduce two different methods, namely background subtraction and color back-projection, to approximately segment the foreground objects. These two methods are finally combined to arrive at a better solution in locating the objects.

B.l Background Subtraction

The simplest approach to detect foreground objects is by subtracting image frames from a reconstructed background. Denote Bg as a reconstructed background and I as an image frame indexed by X = (x, y) and time t, we write

$$\mathbf{R}(X, t) = |\mathbf{Bg}(X + \mathbf{d}(t)) - \mathbf{I}(X, t)| \quad (20)$$

where $\mathbf{d(t)} = \sum_{i=0}^{k-1} \hat{d}(i)$ and R is a residue image. If Bg(X+ d(t)) is a hole, $\mathbf{R}(X, t)$ will be filled by the value 255.

B.2 Color Back-Projection

Suppose the approximate region of a foreground object is known, we can actually replace the color values of that region by its color distribution probabilities. In this case, the dominant color of a foreground object will have a high probability, while the sub-regions not belonging to the foreground object should ideally have values close to zero. Thus, we can automatically prune the approximate region, whilst effectively locating the foreground object.

In our scheme, the support layer of a foreground object Mask_f can be simply obtained by inverting the support layer of a background object Mask_b, i.e.,

$$\mathrm{Mask}_f(X, t) = \begin{cases} 1 & \text{if } \mathrm{Mask}_b(X, t) = 0 \\ 0 & \text{if } \mathrm{Mask}_b(X, t) = 1 \end{cases} \quad (21)$$

Our approach computes the $3D$ color histogram of the regions $\mathcal{R}$ supported by Mask_f throughout a sequence, and then projects the probability values [0, 1] back to $\mathcal{R}$. Let ‘1-l be a normalized histogram, and N_k be the k quantized color value, mathematically we have

$$\text{project: } \mathcal{H}(N_k) = \sum_t \sum_X \frac{1}{\mathcal{A}} \quad \forall_{X,t}\{\mathcal{Q}(\mathcal{R}(X, t)) = N_k\}$$

$$\text{back-project: } \hat{\mathcal{R}}(X, \mathrm{t}) = \mathcal{H}(\mathcal{Q}(\mathcal{R}(X, t)))$$

where $\mathcal{A}$ is the area of $\mathcal{R}$, while function $\mathcal{Q}$ is the color quantization.

B.3 Foreground Image Computation

Background reconstruction is always imperfect due to the ghosting effect, as a result, noise removal after background subtraction can be a dirty task. Likewise, the drawback of color back-projection is amplified when the foreground and background are somehow similar in color; the color histogram need to be finely quantized in order to distinguish the color of foreground and background objects. As a compromise, the two approaches can be linearly combined to trade-off their disadvantage. Denote $\hat{\mathrm{R}}$ as the normalized residue image of an image frame I, a foreground image 3 is computed by

$$\mathcal{F}(X, t) = \mathrm{Pr}(\mathrm{X} = \mathrm{Foreground}, t) \times \mathbf{I}(X, t) \quad (22)$$

where

$$\mathbf{Pr}(X = \mathrm{Foreground}, t) = \frac{1}{2}\{\hat{\mathbf{R}}(X, t) + \hat{\mathcal{R}}(X, t)\} \quad (23)$$

is the probability of a pixel $\mathbf{I}(X, t)$ belong to a foreground object. In (22), ideally, the background pixels of I should be set to zero, while the foreground pixels should be set to a value closed to the color value of I.