CHAPTER 6

6. HIGHLIGHTS-BASED ANNOTATION AND RETRIEVAL

6.1 Introduction

Automatic recognition of highlights from videos is a fundamental and challenging problem for automatic annotation and content-based retrieval applications. Generally speaking, a video contains frames in series and can thus be considered as to have a linear structure. On the other hand, videos can also be represented using a hierarchical structure in which video shots and video scenes are two commonly used higher levels than frames [4, 11, 103, 105]. Consequently, there are at least four levels in the hierarchical representation of a video, including key frames, shots, scenes and the whole video. Although this hierarchical structure provides a practical approach to video representation, it lacks the semantic content required by general users. Therefore, the aim here in this thesis is to extract semantics and highlights from videos and annotate the video shots for further content-based indexing and retrieval.

In many video applications, these highlights are emphasized by involving human objects in some pattern of motion, such as zoom-in and zoom-out camera motion or

walking, running object motion. Such highlights are used in TV news, sports games, and films, etc. [1, 8-10, 29]. As a result, it is intended to detect these kinds of scenarios for intelligent indexing and retrieval of video. Regarding the detection of human objects, they are determined via skin pixels classification. In fact, the identification of skin pixels in videos plays important roles for many significant image and vision applications, such as face detection, facial expression recognition, gesture recognition, human-computer interaction and even naked people detection [111-115]. With human objects detected, object-oriented indexing and access of the whole video can then be achieved.

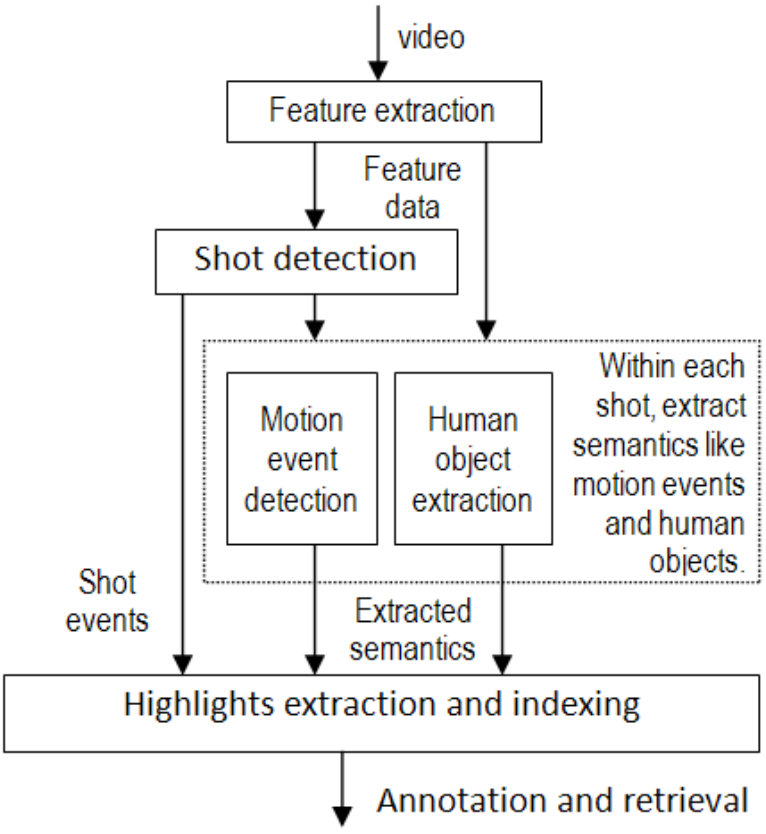


Figure 6.1. An overall system diagram for video highlights extraction for automatic annotation and content-based retrieval.

To illustrate how the proposed system works for highlights based video annotation and retrieval, an overall diagram is given in Figure 6.1 which contains five main blocks, i.e. feature extraction, shot detection, motion event detection, human object extraction, and extraction and indexing of highlights. It is interesting to note that high-level semantics as video highlights are automatically extracted via low-level feature analysis, including the detection of human objects and camera motion events for shot-level effective video annotation and content-based indexing and retrieval.

In this chapter, techniques are proposed to solve this problem using knowledge supported extraction of semantic contents, and the compressed-domain processing is employed for efficiency. Firstly, knowledge-supported rules are utilized for shot detection on the extracted DC-images, and statistical skin detection is applied for human object detection. Secondly, through filtering outliers in motion vectors, improved detection of camera motions like zooming, panning and tilting are achieved. High-level semantics like video highlights are then automatically extracted via low-level analysis in the detection of human objects and camera motion events, and finally these highlights are used for video annotation and retrieval. Results from large data of test videos have demonstrated the accuracy and robustness of the proposed techniques.

The rest of the chapter is organized as follows. Section 6.2 discusses the details of compressed-domain feature extraction and video segmentation. In Section 6.3, the

extraction of human objects, an improved scheme for determining camera motion patterns and an overall workflow for video highlights indexing and retrieval are presented. Comprehensive experimental results and discussions are presented in Section 6.4 with some brief conclusions summarised in Section 6.5.

6.2 Feature Extraction and Video Segmentation

6.2.1 Feature Extraction from MPEG Videos

For the i^{th} input frame f_i, using the similar approach as presented in Chapter 3 (see in Section 3.2.1), its DC parts of the Y, Cb and Cr component images can be extracted and respectively denoted as $Y_{dc}(i), U_{dc}(i)$ and $V_{dc}(i)$, where the number of macroblocks in the horizontal and vertical directions are denoted as N_h and N_v. Denote N_y, N_{cb} and N_{cr} respectively as the numbers of elements in $Y_{dc}^{(i)}, U_{dc}^{(i)}$ and $V_{dc}^{(i)}$, for the case of the 4:2:0 chromatic format, we have

$$N_y = 4N_h N_v, N_{cb} = N_{cr} = N_h N_v \tag{6-1}$$

For frames i and j, the normalized luminance difference is then defined as follows:

$$D_y(i, j) = \frac{1}{N_y \cdot 255} \sum_{n=1}^{N_y} \left| Y_{dc}^{(i)}(n) - Y_{dc}^{(j)}(n) \right| \tag{6-2}$$

Similarly, the normalized differences of the two chromatic components are defined below as $D_u(i, j)$ and $D_v(i, j)$.

$$D_u(i,j) = \frac{1}{N_{cb} \cdot 255} \sum_{n=1}^{N_{cb}} \left| U_{dc}^{(i)}(n) - U_{dc}^{(j)}(n) \right| \tag{6-3}$$

$$D_v(i,j) = \frac{1}{N_{cr} \cdot 255} \sum_{n=1}^{N_{cr}} \left| V_{dc}^{(i)}(n) - V_{dc}^{(j)}(n) \right| \tag{6-4}$$

In addition, M_i is defined as the motion magnitude measurement in frame i, on the basis of extracted motion vectors $V_x(i,n)$ and $V_y(i,n)$ from the n^{th} macroblock.

$$M_i = \frac{1}{\alpha_i} \sum_{n=1}^{N_h N_v} \beta_n \left(\left| V_x(i,n) \right| + \left| V_y(i,n) \right| \right) \tag{6-5}$$

$$\beta_n = \begin{cases} 1 & if \quad \left| V_x(i,n) \right| + \left| V_y(i,n) \right| \geq 2\overline{M}_i / 3 \\ 0 & otherwise \end{cases} \tag{6-6}$$

$$\begin{aligned} \overline{M}_i &= \frac{1}{N_h N_v} \sum_{n=1}^{N_h N_v} [\left| V_x(i,n) \right| + \left| V_y(i,n) \right|, \\ \lambda_i &= \frac{\alpha_i}{N_h N_v} \end{aligned} \tag{6-7}$$

where $\overline{M}_i$ is the average value of the sum of absolute velocities in horizontal and vertical directions of all the macroblocks in f_i, and α_i is the total number of valid motion vectors with their sum of absolute velocity values in horizontal and vertical directions larger than an adaptive threshold of value $2\overline{M}_i / 3$. For B-frames, $V_x(i,n)$ and $V_y(i,n)$ are defined as the absolute value of the forward or backward predicted motion vector components which is the largest. Moreover, λ_i indicates a ratio of verified motion vectors in the whole frame. For intra-coded macroblocks, the parameters simply satisfy $V_x(i,n) = V_y(i,n) = 0$.

6.2.2 Knowledge-Based Shot Change Detection

From the observations of over 25,000 frames taken from sequences containing more than 150 shots, it is found that shot boundary, including both cuts and gradual transitions (GT), can usually be characterized by the characteristics listed below:

- Luminance difference: When a shot change occurs, there is an apparent discontinuity in luminance intensity between the neighbouring frames;
- Chromatic difference: A discontinuity in chromatic signals between neighbouring frames is found when a shot change occurs.

However, it is also found some false alarms of two types:

- Motion: both camera motion and object motion will cause inconsistent measurement of frame difference in terms of $D_y(i, j)$, $D_u(i, j)$ and $D_v(i, j)$;
- Change of luminance and/or chrominance in frames may be caused by flicker or flashing light. Hence these two false alarms should also be removed.

Although both gradual transitions and cuts share some common appearances, such as apparent luminance and chromatic differences, they are different in two ways: i) the difference between neighbouring frames of a GT is smaller in comparison with neighbouring frames of a cut, although the boundary frames of a GT is as different as the boundary frames of a cut; ii) during the shot change a GT has more frames than a cut does. To this end, a cut can be considered as a coarse-sampled GT containing only two frames hence they can be detected simultaneously as presented below.

i) Firstly, each frame f_i is compared with its previous one, f_{i-d}, in terms of D_y, D_u and D_v against three thresholds y_{th}, u_{th}, v_{th}. If any of them exceeds the corresponding threshold, a candidate shot change is detected and go to ii); otherwise set $i = i + d$ and repeat i) until the whole sequence is examined;

ii) For the candidate shot change $[i-d, i]$, it needs to be determined as a GT or a cut. The shot change is a cut if $i_0 \in [i-d, i)$ can be found and one of the difference values $D_y(i_0, i_0+1)$, $D_u(i_0, i_0+1)$, $D_v(i_0, i_0+1)$ exceeds y_{th}, u_{th}, v_{th}, respectively. If the shot change is found to be a cut, then go to iv) otherwise go to iii);

iii) For the candidate $[i-d, i]$, its accurate boundaries i_-, i_+ need to be determined where $i_- \le i-d$, $i_+ \ge i$ and for all $i \in [i_-, i_+ - d]$ one of the difference values $D_y(i, i+d)$, $D_u(i, i+d)$, $D_v(i, i+d)$ exceeding y_{th}, u_{th}, v_{th}, respectively. Afterwards, set $i = i_+ + d$ and go to i) to process other frames;

iv) To verify each of the candidate cut, a sliding window of three frames is introduced, in which f_{i-1} and f_{i+1} are compared. Only if the corresponding difference D_y, D_u or D_v exceeds y_{th}, u_{th} or v_{th}, respectively is the shot change verified and recorded. Otherwise, it is a false alarm caused by either flicker or flashing light. After verification, set $i = i + d$ and go to i) to process other frames.

To cope with motion effects, the thresholds y_{th}, u_{th} and v_{th} are not fixed. Instead, their values are adapted to the image contents of the videos as discussed with reference to Eq. (6-8) below.

$$y_{th} = y_0 \lambda_i ;\ u_{th} = u_0 \lambda_i ;\ v_{th} = v_0 \lambda_i . \tag{6-8}$$

In Eq. (6-8), y_0, u_0 and v_0 are constant parameters which are further used to decide y_{th}, u_{th} and v_{th} by considering λ_i, the ratio of verified motion vectors in f_i as defined in Eq. (6-7). Apparently, a higher value of λ_i means more macroblocks in the frame contain obvious motion (more than M_{th}). The extreme case, $\lambda_i = 1$, refers to motion in all the macroblocks, which very likely corresponds to camera-induced global motion; and the extreme case, $\lambda_i = 0$, refers to motion-free smooth transition between frames. It is easy to see that a larger value of λ_i may cause larger frame difference disparity, and this is the basic motivation to adjust, for robustness, the thresholds y_{th}, u_{th} and v_{th} according to the change of λ_i with the motion magnitude in f_i.

6.3 Skin Detection

After video segmentation, video objects and events are extracted within each video shot. The events here refer to camera motions which will be extracted in Section 6.4 and in this Section techniques of skin detection are discussed for the identification of human entities from videos. Firstly, histogram-based approach is utilized to model colour models of skin and non-skin pixels, in which manual ground truth data of skin and non-skin masks are extracted for this purpose. The main difference between the work in this thesis and others is training in compressed domain, thus the probability from pixel level needs to be mapped to block level to cope with the requirements of

MPEG. With the obtained skin and non-skin models, Bayesian maximum a posteriori decision rule is employed for skin colour classification. To determine an optimal threshold, a likelihood ratio map of skin and non-skin colours is extracted, and the threshold is decided by using minimum probability error strategy. Further details of the proposed model and approach are described below.

6.3.1 Modelling Skin and Non-skin Colours in Compressed Domain

YCbCr colour space is adopted here as it is easily extracted from MPEG compressed videos. Then, for each colour entry $e_c = (y, c_b, c_r)$, its associated probabilities as skin and non-skin, $p(e_c / skin)$ and $p(e_c / nonskin)$, are extracted as follows.

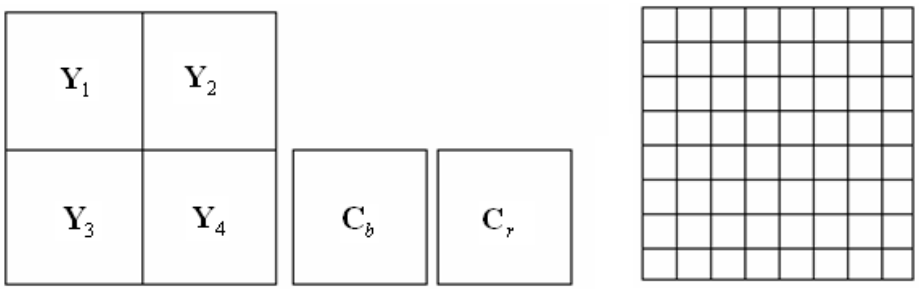
$$p(e_c / skin) = sum(e_c / skin) / V_s \tag{6-9}$$

$$p(e_c / nonskin) = sum(e_c / nonskin) / V_{\bar{s}} \tag{6-10}$$

where $sum(e_c / skin)$ and $sum(e_c / nonskin)$ denote number of occurrence in training data when the colour entry e_c appears as skin and non-skin, respectively. V_s and $V_{\bar{s}}$ indicate volumes of skin and non-skin data, i.e., total number of occurrences in each model.

In the uncompressed pixel domain, $sum(.)$ can be easily attained by counting pixels of same colour entry. However, it becomes complex to count in the compressed domain, as only blocks can be accessed, rather than pixels, to avoid expensive inverse DCT. In

fact, the proposed training in compressed domain is defined on the basis of DCT coefficients after simple entropy decoding. As a result, these DCT coefficients are extracted from each macroblock of 16*16 pixels. In 4:2:0 chrominance format, one macroblock contains four luminance sub-blocks and two chrominance sub-blocks, and each sub-block has a size of 8*8 pixels (see Fig. 6.2).



(a) (b)

Figure 6.2. One macroblock in 4:2:0 chrominance format contains four luminance subblocks and two chrominance subblocks (a) and each subblock has 8*8 pixels (b).

For simplicity, only the DC components in each sub-block are extracted. Therefore, there are totally 6 DC components of which four from Y sub-blocks, one from Cb and one from Cr sub-block, respectively. A combined colour entry of the macroblock, e_b, is then extracted by using the average luminance of four Y components as its luminance and Cb, Cr its chrominance components.

With the extracted block-based colour entry, its probability of skin and non-skin can also be decided in a similar way as defined in (6-9) and (6-10). However, new definition of the *sum*(.) function is defined in (6-11) and (6-12), where $N_s(b)$ and

$N_{\bar{s}}(b)$ indicate number of skin and non-skin pixels in the macroblock b, and $N = 256$ is the total number of pixels in b. Please note that $N_s(b) + N_{\bar{s}}(b) \neq N$ when masks of skin and non-skin are defined separately, especially when there are the third class of pixels introduced, although only two-classes training is utilized [112].

$$sum(e_b / skin) = N_s(b) / N \tag{6-11}$$

$$sum(e_b / nonskin) = N_{\bar{s}}(b) / N \tag{6-12}$$

6.3.2 Bayesian Classification

Please note that the probabilities extracted above are conditional probability of skin and non-skin, respectively. Given a colour entry e_b, the posterior probability of skin and non-skin are determined below based on the well-known Bayesian theorem in the inference process.

$$p(skin/e_b) = \frac{p(e_b / skin)\,p(skin)}{p(e_b / skin)\,p(skin) + p(e_b / nonskin)\,p(nonskin)} \tag{6-13}$$

$$p(nonskin/e_b) = \frac{p(e_b / nonskin)\,p(nonskin)}{p(e_b / skin)\,p(skin) + p(e_b / nonskin)\,p(nonskin)} \tag{6-14}$$

where $p(skin)$ and $p(nonskin)$ are the prior probability.

According to maximum a posteriori decision rule, e_b refers more likely to skin colour if its associated posterior probability of skin is more than that of non-skin, i.e.

$p(skin/e_b) > p(nonskin/e_b)$. In other words, it means the posterior probability of skin and non-skin satisfies (6-15), where $\theta \geq 1$ is a constant.

$$\frac{p(skin/e_b)}{p(nonskin/e_b)} = \frac{p(e_b/skin)p(skin)}{p(e_b/nonskin)p(nonskin)} > \theta \tag{6-15}$$

Since the prior probabilities of skin and non-skin are strongly dependent on the training data and seems neither reliable nor objective, they are omitted in classification by introducing a new term λ, where $\lambda = p(skin)/p(nonskin)$. Then, the decision rules in (6-15) becomes (6-16), which indicates thresholding of the likelihood ratio of skin and non-skin for classification, and $\eta = \theta/\lambda$ is a chosen threshold.

$$\frac{p(e_b/skin)}{p(e_b/nonskin)} > \eta \rightarrow skin \tag{6-16}$$

6.3.3 Optimal Thresholding

Obviously, the performance of detection and classification depends on a suitable parameter of η. There are several ways to choose this threshold, including global optimization on ROC analysis [113], minimum probability error [115], and even empirically [123]. In this chapter, a similar probability error analysis is adopted as used in [115], but the threshold is obtained by analyzing the effectiveness of extracted skin and non-skin models as below.

Firstly, a logarithmic likelihood map (LLM), $g(e_b)$, is derived as

$$g(e_b) = \rho \ln(1 + \frac{p(e_b / skin)}{p(e_b / nonskin)}) \qquad (6\text{-}17)$$

where $\rho > 0$ is a constant to scale LLM value within a given range, say [0, 255].

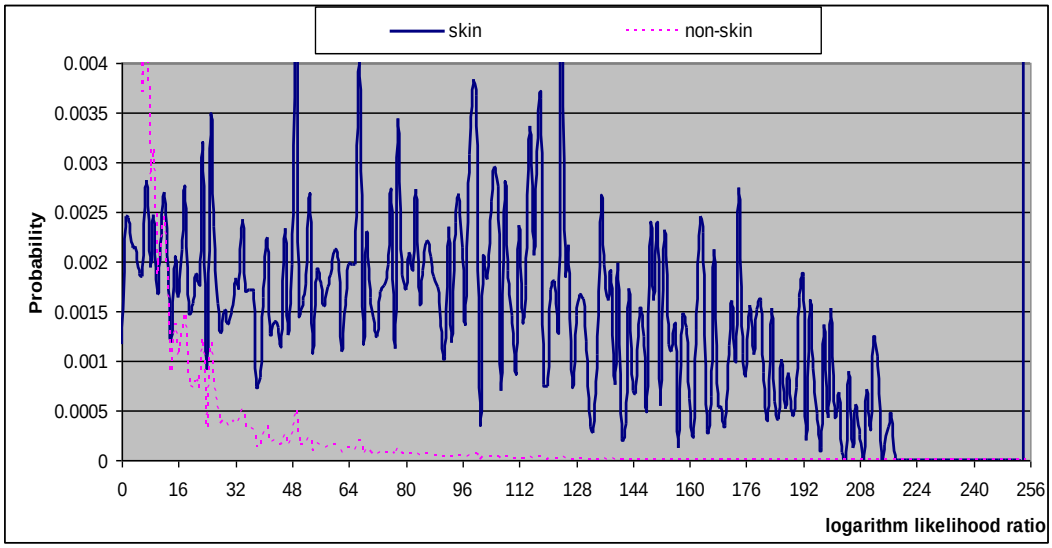


Figure 6.3. Histograms of logarithm likelihood ratio of skin and non-skin colours.

As a result, the classification process becomes thresholding on this LLM. There are two reasons to employ the logarithmic operator to likelihood ratio of skin and non-skin here: one is to enhance the details when the likelihood ratio is small, and the other is helps to constrain the large range of likelihood ratio into a relatively small range.

According to skin and non-skin pixels, two histograms of this LLM, H_s and $H_{\bar{s}}$ are extracted separately from both skin and non-skin masks in the training data. In Fig. 6.3, H_s and $H_{\bar{s}}$ show distributions of this logarithm likelihood map over sample colours of skin and non-skin, respectively. The accumulated probability of H_s and $H_{\bar{s}}$ are

extracted as A_s and $A_{\bar{s}}$, respectively. Curves of A_s and $A_{\bar{s}}$ against logarithm likelihood ratio are plotted in Fig. 6.4.

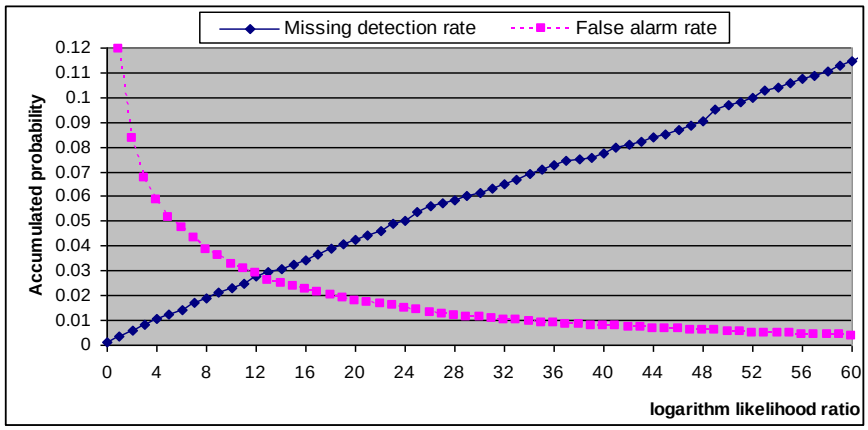


Figure 6.4. Curves of A_s and $A_{\bar{s}}$ against logarithm likelihood ratio indicates potential missing detection rate and false alarm rate.

Taking g as a threshold for classification, apparently, $A_s(g)$ denotes percentage of training data of skin colour has a logarithm likelihood ratio no more than g, i.e. the missing detection rate; and $A_{\bar{s}}(g)$ denotes percentage of training data of non-skin colour has a logarithm likelihood ratio greater than g, i.e. the false alarm rate. Then, the overall probability of error classification can be derived as

$$P_{error}(g) = A_s(g)\,p(skin) + A_{\bar{s}}(g)\,p(nonskin) \qquad (6\text{-}18)$$

One solution to obtain a suitable threshold g is to minimize $P_{error}(g)$ by taking $p(skin)$ and $p(nonskin)$ from training data as two weights in (6-18). An alternative solution is to choose the threshold which yields same false alarm rate and missing detection rate, i.e. $A_s(g) = A_{\bar{s}}(g)$, and the corresponding probability of error

classification becomes $A_s(g)$, too. As normally $p(skin) < p(nonskin)$, the threshold obtained in the second solution appears less than the one from the first solution. As a result, higher detection rate and more false alarms are intended to be detected. According to the training results showed in Fig 6.4, the threshold in the first solution is found as 49.25 with $P_{error} = 1.38\%$. While the threshold obtained from the second solution is 12.22 and $P_{error} = 2.82\%$. Since $\rho = 30$, the corresponding thresholds in (6-16) satisfies $\eta = 4.164$ and $\eta = 0.5028$, respectively. Please note the probability errors above are results from the training data only.

6.3.4 Post-processing

To fill small holes and also remove spurs in the detected mask, morphological filtering is applied to the detected masks. Let M_0 and M_s denote detected skin masks (both in binary) before and after this filtering, they satisfy

$$M_s = M_0 \oplus B - B \tag{6-19}$$

where B is a 3×3 structure element, $\oplus$ and – denote morphological dilation and erosion operators, respectively.

Besides, small areas with their sizes less than a given threshold, s_0, are also removed from M_s. Due to the fact that each pixel in the detected mask image represents one macroblock, i.e. 16*16 pixels in original frame image, a relative small s_0 no more than 3 should be chosen in the proposed system.

6.3.5 Experimental Results on Skin Detection

All the test data in the experiments is from Boston University which contains 21 sequences and can be accessed from (http://csr.bu.edu/colourtracking/pami/Data/) [112]. In each of the sequences, there is dynamic changing of illuminations which leads to some different between these frames. For convenience, the width of each frame is cropped from 641 to 640 and encoded each sequence as separate MPEG stream. Besides, two groups of ground truth maps are manually defined as skin and non-skin masks, respectively, which enables a third category of "don't care" pixels included in a non-skin mask which belongs to neither skin nor non-skin background. Four examples of test frames and their corresponding masks are illustrated in Fig. 6.5, and white pixels in (b-d) respectively refer to skin, non-skin and don't care masks.

(a) (b) (c) (d)

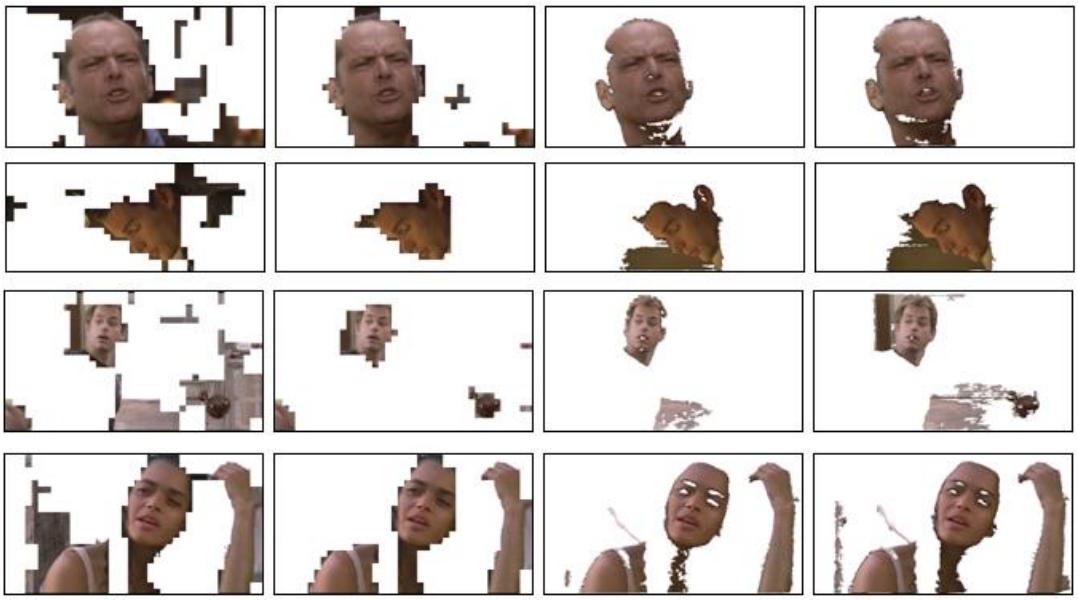
Figure 6.5. Examples of four test frames (a) and their associated masks of skin (b), non-skin (c) and don't care pixels (d).

In both training and test process, only the I-frame is chosen as its DCT coefficients can be directly extracted from compressed streams. In order to fully utilize the ground truth maps, the sequences of MPEG streams are coded with I-frame only, i.e. there is only one frame in each group of picture. Further information about these sequences can be found in details in [112].

Firstly, detected skin masks are compared with those from Sigal etc. in [112], in total four groups of results are compared. Two of them are ours using threshold of 12.2 and 49.25, respectively. The other two groups are results from static and dynamic models proposed in [112]. According to the source images in Fig. 6.5, detected skin masks are shown in Fig. 6.6, and from which several facts can be found as follows.

1) Results from threshold of 12.2 have more false alarms than those from threshold of 49.25, which indicates threshold derived from minimum probability of error classification more suitable in this context;
2) Although the dynamic model may help to fill the holes in detection by adapting the varying illumination, it also has the potential to cause more false alarms;
3) Pixel-based model in [112] can successfully exclude small non-skin areas like eyes, mouth and accurately locate non-skin boundaries owing to its finer resolution than the proposed approach, which has a minimum resolution of one macroblock, i.e. 16*16 pixels! However, in comparison with Sigal's approach, the results of proposed method from threshold 49.25 still yield better results in

the first two test images (need to remove small areas of noise) and comparable result in the third test image.



(a) (b) (c) (d)

Figure 6.6. Four results of detected skin from images in Fig. 6.5. (a) and (b) are results from proposed method using thresholds 12.2 and 49.25, respectively; (c) and (d) are results from Signal etc[112] using their static and dynamic models, respectively.

By calculating the correct detection rate of both skin and non-skin background, quantitative comparisons with those from Sigal's are given in Table 6.1, in which the results from Sigal are directly duplicated from [112]. Please note that due to the fact that the minimum resolution of the proposed approach in detection is a macroblock, hence accurate boundary of skin and non-skin areas cannot be yielded in the proposed approach, which certainly leads to inaccuracy quantitative measurements in such a performance analysis. Nevertheless the results achieved with threshold of 49.25 yield better or comparable results to Sigal's models in 11 sequences (#1, #3, #6, #10, #12,

#13, #14, #15, #19, #20 and #21). Considering its efficiency in compressed domain and inaccuracy in such a measurement, it is found that the results are shown very promising in spite of varying luminance in those test sequences.

Table 6.1. Performance comparisons of the proposed approach and Sigal etc. in [112]

Sequences		Our approach				Sigal' approach			
		Threshold = 12.2		Threshold = 49.25		Static model		Dynamic model	
#	#frames	**skin**	bk	**skin**	bk	**skin**	bk	**skin**	bk
1	100	**94.64**	97.39	**75.79**	99.20	**49.08**	99.96	**65.74**	99.35
2	72	**99.10**	90.60	**97.66**	97.35	**96.46**	99.99	**98.19**	99.73
3	72	**98.48**	93.53	**94.76**	99.82	**77.21**	91.62	**88.92**	86.43
4	110	**98.09**	95.22	**90.30**	98.47	**92.67**	99.73	**97.63**	99.13
5	75	**98.34**	98.94	**84.55**	99.84	**96.87**	99.86	**98.30**	99.66
6	72	**99.55**	98.73	**97.51**	99.68	**88.32**	99.23	**94.27**	99.14
7	76	**91.00**	99.648	**81.27**	99.60	**77.67**	100.0	**91.30**	100.0
8	73	**99.50**	92.84	**99.02**	95.87	**99.99**	98.72	**99.98**	97.17
9	72	**85.06**	99.68	**79.89**	99.91	**81.30**	99.62	**92.81**	100.0
10	73	**100.0**	45.50	**99.74**	52.61	**96.00**	36.56	**99.96**	15.72
11	233	**60.23**	99.14	**55.13**	99.46	**87.47**	99.93	**93.99**	99.59
12	72	**92.69**	97.44	**81.91**	99.29	**70.51**	97.49	**62.36**	95.94
13	350	**91.40**	99.01	**79.21**	99.85	**67.73**	99.96	**82.21**	99.71
14	72	**99.51**	98.90	**97.34**	99.78	**91.79**	99.98	**98.90**	97.73
15	75	**99.09**	89.74	**96.19**	97.31	**91.03**	95.37	**94.10**	90.18
16	50	**76.02**	98.91	**40.00**	99.97	**52.05**	100.0	**88.95**	99.82
17	75	**95.88**	99.90	**89.63**	99.97	**97.89**	99.98	**99.43**	99.66
18	91	**92.91**	99.82	**82.59**	99.97	**92.07**	99.99	**98.60**	99.94
19	73	**43.68**	99.67	**32.17**	99.89	**11.02**	99.91	**24.29**	99.48
20	120	**75.67**	99.70	**41.58**	99.96	**18.75**	100.0	**55.79**	90.95
21	53	**98.55**	92.20	**93.59**	98.52	**92.96**	98.42	**97.94**	95.15

6.4 Annotation and Retrieval of Video Highlights

With detected skin regions, human objects will be indexed with further extracted motion events which refer to several camera motions such as zoom in/out, pan and tilt. In addition, video highlights are then indexed and retrieved as certain objects involved in relevant events as discussed below.

6.4.1 Determining Camera Motion Patterns

According to the 6-parameter projective camera model considering only rotation and zoom between frames in (6-20) and (6-21), it is indicated in [105] that p_1 is camera zoom factor ($p_1 > 1$ represents zoom in and $p_1 > 1$ represents zoom out) and (x_i, y_i) and (x_{i-1}, y_{i-1}) are the image coordinates of corresponding points in f_i and f_{i-1}, respectively. In addition, p_5 and p_6 refer to perspective distortion effects, and p_2 represents rotation about the axis of the camera lens.

$$x_i = \frac{p_1 x_{i-1} + p_2 y_{i-1} + p_3}{p_5 x_{i-1} + p_6 y_{i-1} + 1}. \tag{6-20}$$

$$y_i = \frac{-p_2 x_{i-1} + p_1 y_{i-1} + p_4}{p_5 x_{i-1} + p_6 y_{i-1} + 1}. \tag{6-21}$$

If p_2, p_5 and p_6 all are set as 0, the above equations become

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} p_1 & 0 \\ 0 & p_1 \end{pmatrix} \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} p_3 \\ p_4 \end{pmatrix} \tag{6-22}$$

In [105], corresponding pair of points are obtained automatically by checking the two macroblocks in f_i and f_{i-1} connected by the motion vectors in P-frames of MPEG video. Finally, p_1 is determined below, where N refers to the number of inter-coded macroblocks:

$$p_1 = \frac{\sum_{k=1}^{N} (w_{i(k)} - \overline{w}_i)^{\mathrm{T}} (w_{i-1(k)} - \overline{w}_{i-1})}{\sum_{k=1}^{N} \| w_{i-1(k)} - \overline{w}_{i-1} \|^2} \tag{6-23}$$

$$\begin{pmatrix} p_3 \\ p_4 \end{pmatrix} = \frac{\overline{w}_i}{p_1} - \overline{w}_{i-1} = \frac{1}{p_1} \begin{pmatrix} \overline{x}_i \\ \overline{y}_i \end{pmatrix} - \begin{pmatrix} \overline{x}_{i-1} \\ \overline{y}_{i-1} \end{pmatrix} \tag{6-24}$$

where $w_{i(k)} = (x_{i(k)}, y_{i(k)})^{\mathrm{T}}$, $w_{i-1(k)} = (x_{i-1(k)}, y_{i-1(k)})^{\mathrm{T}}$, $\overline{w}_i = \sum_{k=1}^{N} \frac{w_{i(k)}}{N} = \begin{pmatrix} \overline{x}_i \\ \overline{y}_i \end{pmatrix}$, and $\overline{w}_{i-1} = \sum_{k=1}^{N} \frac{w_{i-1(k)}}{N} = \begin{pmatrix} \overline{x}_{i-1} \\ \overline{y}_{i-1} \end{pmatrix}$.

As having pointed out in the review chapter in Section 2.5, the above solution suffers from false alarms caused by object motion. In the improved approach, the method introduced in [118] is employed to remove outlier motion vectors using neighborhood and smoothness constraints. Hence not all the inter-coded macroblocks are used in estimating of camera motion. Instead, only macroblocks with motion vectors satisfying smoothness conditions in [118] are considered to estimate p_1 , p_3 and p_4 in (6-23) and (6-24).

In the following, motion patterns are determined according to estimated values of p_1 , p_3 and p_4, in which two separate decisions as pat_{zoom} and pat_{shift}, defined as follows. A small positive number $\delta \in (0, 0.05)$ is used for robustness.

$$pat_{zoom} = \begin{cases} zoom\_in, & if \quad p_1 > 1+\delta \\ zoom\_out, & if \quad p_1 < 1-\delta \\ none, & otherwise. \end{cases} \tag{6-25}$$

$$pat_{shift} = \begin{cases} pan, & if \quad |p_3| > 10\delta \wedge |p_4| < 10\delta \\ tilt, & if \quad |p_3| < 10\delta \wedge |p_4| > 10\delta \\ mix, & if \quad |p_3| > 10\delta \wedge |p_4| > 10\delta \\ none, & otherwise. \end{cases} \tag{6-26}$$

In addition, temporal median filtering is applied to these detected motion patterns for robustness, using a window size of selected as three frames. Afterwards, the number of

sequential frames with the same motion pattern is counted. If the number is found more than 5, corresponding to an interval of 0.2s at 25fps, the motion pattern is considered as valid due to the fact that our human vision system is less sensitive to events in short clips.

6.4.2 Highlights-Based Annotation and Retrieval

With extracted video events and objects, video highlights are then obtained as certain (human) objects undergoing motion events. These highlights are associated with corresponding video shots for shot level indexing and retrieval. Consequently, the overall workflow for the extraction of highlights is summarized below.

i) For each input video, shot boundaries are detected for video segmentation;

ii) Within each detected video shot, video events of camera motion patterns and video objects of human entities are extracted;

iii) Video highlights are determined as video objects in conjunction with certain events, such as panning or zooming-in human objects;

iv) Finally, these extracted video highlights are further used as semantic concepts in shot-level automatic video annotation, content indexing and retrieval.

6.5 Results and Discussions

In the experiments, 8 test sequences from four categories are utilised including film, sports, news and education programmes as summarized in Table 6.2. In total there are

104800 frames segmented into 278 shots (77 cuts and 201 GTs), also 27 zoomed-in human objects and 75 other camera events are manually extracted from these videos and labelled as highlights for further tests.

Table 6.2. Detail information about the test sequences

Video	**Category**	**Frames**	**Shots**		**Events**	
			cut	GT	Zoom	Pan/Tilt
(a)	Film	*16300*	*11*	*37*	*3*	*8*
(b)	Film	*8250*	*6*	*25*	*2*	*5*
(c)	Sports	*9900*	*9*	*22*	*3*	*11*
(d)	Sports	*24000*	*17*	*61*	*5*	*19*
(e)	News	*7700*	*9*	*15*	*2*	*9*
(f)	News	*8750*	*7*	*16*	*3*	*7*
(g)	Education	*11300*	*8*	*12*	*4*	*7*
(h)	Education	*18600*	*10*	*13*	*5*	*9*
Sum	N/A	*104800*	*77*	*201*	*27*	*75*

In the following, the results are evaluated in terms of video segmentation, object detection, event detection and content-based retrieval. For quantitative evaluation, precision and recall measures are calculated below in comparison with manual ground truth over the obtained results.

$$\text{Precision} = \frac{tp}{tp + fp} = \frac{|B_{\text{result}} \cap B_{gt}|}{|B_{\text{result}}|}. \tag{6-27}$$

$$\text{Recall} = \frac{tp}{tp + fn} = \frac{|B_{\text{result}} \cap B_{gt}|}{|B_{gt}|} = R_c. \tag{6-28}$$

Here B_{result} and B_{gt} denote detected (extracted) result and the ground truth respectively; tp and fp refer to true positive (correct detected) and false positive (false alarm) samples respectively, and fn denotes false negative (missing detected) samples.

In addition, a combined measurement of both precision and recall, F_1, can be used to evaluate the overall performance of shot change detection for video segmentation. This measurement is defined in (3-18) as $F_1 = \frac{2recall \cdot precision}{recall + precision}$.

6.5.1 Results in Video Segmentation

For video segmentation, precision and recall were measured over each test sequence as reported in Table 6.3. All the measures are presented in respect to detections of cuts, GTs and overall performance. Evaluations of F_1 measures over the eight test sequences are plotted in Fig. 6.7 for comparisons.

Table 6.3. Performance in terms of precision and recall measures for shot detection.

Video	**Cut**		**GT**		**Overall**	
	Precision	Recall	Precision	Recall	Precision	Recall
(a)	*91.7%*	*100.0%*	*80.5%*	*89.2%*	*83.0%*	*91.7%*
(b)	*100.0%*	*100.0%*	*81.5%*	*88.0%*	*84.8%*	*90.3%*
(c)	*100.0%*	*100.0%*	*83.3%*	*90.9%*	*85.3%*	*93.6%*
(d)	*88.9%*	*94.1%*	*86.2%*	*91.8%*	*86.7%*	*92.3%*
(e)	*100.0%*	*100.0%*	*76.5%*	*86.7%*	*84.6%*	*91.7%*
(f)	*100.0%*	*100.0%*	*73.7%*	*87.5%*	*80.8%*	*91.3%*
(g)	*100.0%*	*100.0%*	*78.6%*	*91.7%*	*86.4%*	*95.0%*
(h)	*90.9%*	*100.0%*	*73.3%*	*84.6%*	*80.8%*	*91.3%*
Mean	**95.0%**	**98.7%**	**81.1%**	**89.6%**	**84.5%**	**92.1%**

As can be seen from both Table 6.3 and Fig. 6.7, firstly the shot detection approach yields promising results in terms of good precision rates and recall rates. Secondly, much better results have been achieved in the detection of cuts in comparison with the

results for GT detection. Owing to the fact that the number of GTs is more than that of cuts in each sequence, the overall performance is slightly higher than that of GT detection but much lower than that of cut detection. Improvements in the detection of GTs need to be further investigated, especially in reducing false alarms to obtain higher precision measures.

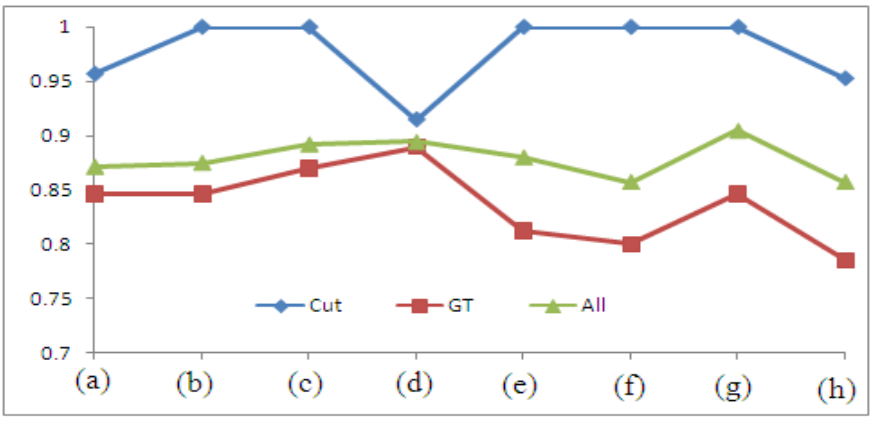


Figure 6.7. Plots of F_1 measures on the detection of cuts, GTs and all shots in the eight test sequences (a) to (h).

6.5.2 Results in Skin and Human Object Detection

After shot detection, human objects are extracted within each shot via the detection of skin pixels. A frame is labelled as containing human objects if there is at least one large skin block detected, satisfying two conditions: i) its area exceeds a given threshold and ii) the ratio of the height to width of its bounding box lies within a certain range. Figure 6.8 shows three examples to illustrate the detected skin blocks and labelled human objects and also shows the original images for comparisons. As

seen, the proposed algorithm has successfully detected skin blocks from different colour images. Applying the constraints of minimum area and valid range of height-width ratio, false alarms are removed.

(a) (b) (c)

Figure 6.8. Three groups of skin detection results (a-c), where the top row shows original image frames and corresponding skin results are shown at the bottom. Extracted human objects are labelled within red boxes.

6.5.3 Results in Determination of Motion Patterns

For camera event detection, the results extracted from each sequence are given in Table 6.4, again using the precision and recall measures. As can be seen, the recall rate is fairly good (greater than 85%) but precision needs to be improved. One reason is that the number of events contained in each video sequence is quite limited, especially for camera zooming events, hence even one false alarm may cause poor precision rate in quantitative evaluations. In addition, the result for detecting camera shifts including pan, tilt and mixing are better than for detecting zooming effects.

Table 6.4. Precision and recall measures for camera events detection.

Video	**Zooming**		**Shift**		**Overall**	
	Precision	Recall	Precision	Recall	Precision	Recall
(a)	*66.7%*	*66.7%*	*77.8%*	*87.5%*	*75.0%*	*81.8%*
(b)	*66.7%*	*100.0%*	*83.3%*	*100.0%*	*77.8%*	*100.0%*
(c)	*75.0%*	*100.0%*	*69.2%*	*81.8%*	*70.6%*	*85.7%*
(d)	*80.0%*	*80.0%*	*72.7%*	*84.2%*	*74.1%*	*83.3%*
(e)	*100.0%*	*100.0%*	*80.0%*	*88.9%*	*83.3%*	*90.9%*
(f)	*75.0%*	*100.0%*	*75.0%*	*85.7%*	*75.0%*	*90.0%*
(g)	*60.0%*	*75.0%*	*85.7%*	*85.7%*	*75.0%*	*81.8%*
(h)	*66.7%*	*80.0%*	*70.0%*	*77.8%*	*68.8%*	*78.6%*
Mean	**71.9%**	**85.2%**	**75.3%**	**85.3%**	**74.4%**	**85.3%**

6.5.4 Results in Video Highlights Annotation and Retrieval

With extracted human objects and detected camera events, the original videos are then annotated and indexed in each shot for content-based retrieval: i) Indexing of extracted human objects; ii) Indexing of camera events, including zooming in/out, panning, tilting, etc.; and iii) Indexing of human objects under camera events.

Consequently, three kind of content retrieval tests can be achieved by access to these indexes. For the first test, extracted human objects are shown as a list of main characters for both content-based browsing and example-based query. This is illustrated in Figure 6.9 within the main video retrieval interface where the image within the bounding box of associated skin block is clipped to show in the list. For the second test, different camera motion patterns are illustrated for sketch-based query as shown in Figure 6.10. The third test is a combination of the previous two tests.

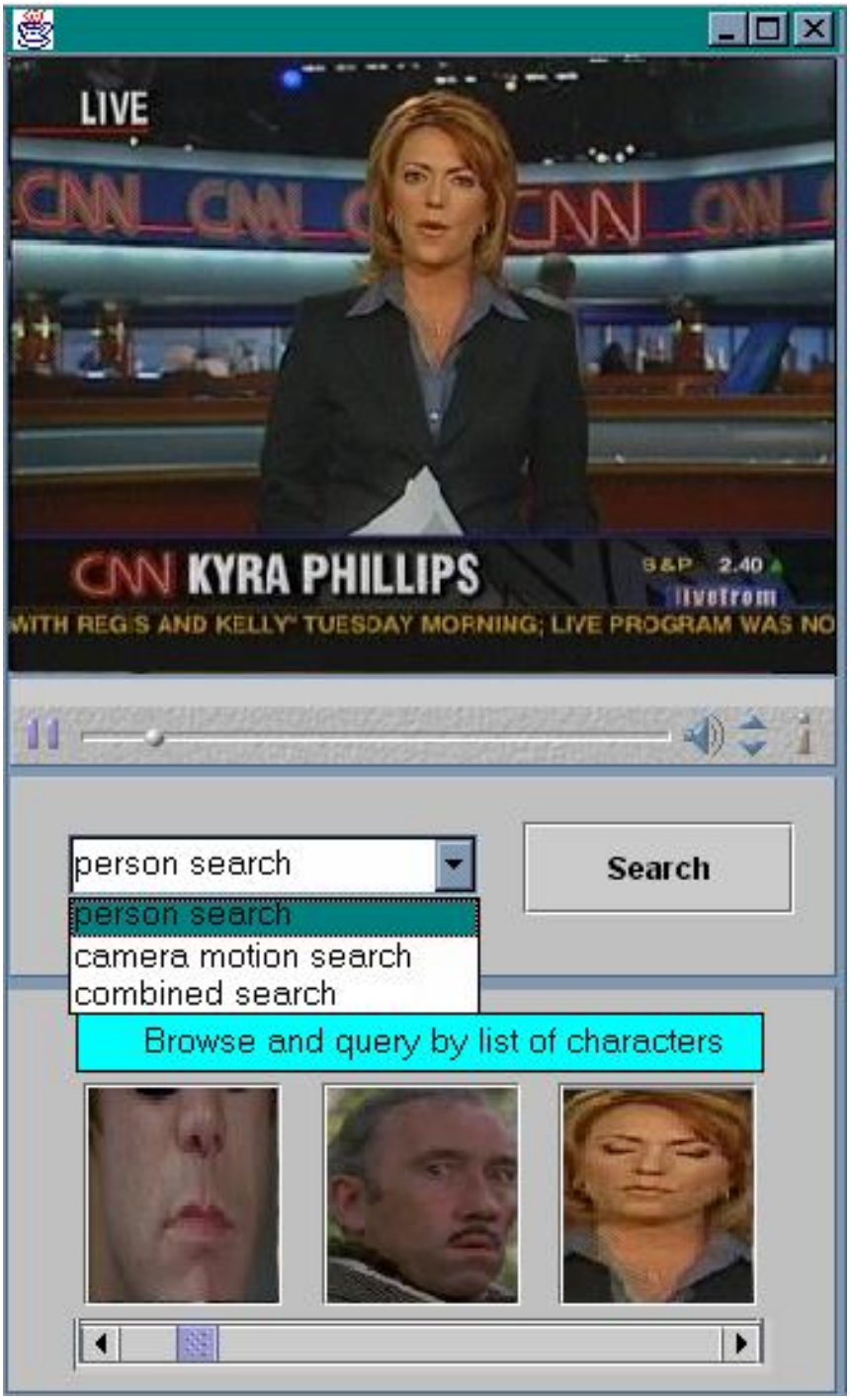


Figure 6.9. Main query interface with list of main characters automatically extracted via human object detection.

The three tests above have been found to be user-friendly in terms of effective video retrieval, especially in content-based retrieval at events level and objects level. The

corresponding results are analyzed in detail, again using precision and recall measures and presented in Figure 6.11. As can be seen, the best and the worst results are found in object retrieval and combined retrieval, respectively, whilst retrieval of events yields intermediate results. This is because human objects have been extracted more accurately than camera events. Since such inaccuracy in the extraction of objects and events will lead to inadequate content indexing, large errors will be generated in querying both objects and events due to accumulated inaccuracy in comparison with manual ground truth. However, the overall performance is still very promising considering the fact that the proposed algorithm is fully implemented in the compressed domain and the accuracy can be improved by introducing further processing in the pixel domain.

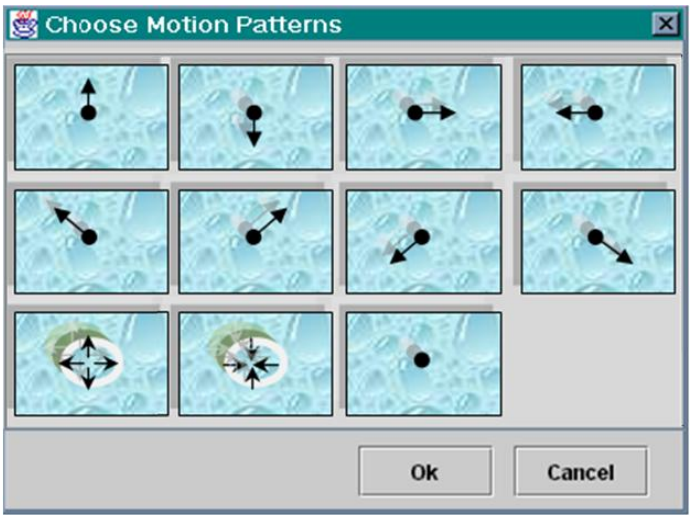


Figure 6.10. Sketches to illustrate various motion patterns: the first row is for pure pan/tilt; the second row is for mixed shift; and the last refers to zooming in/out and static cases.

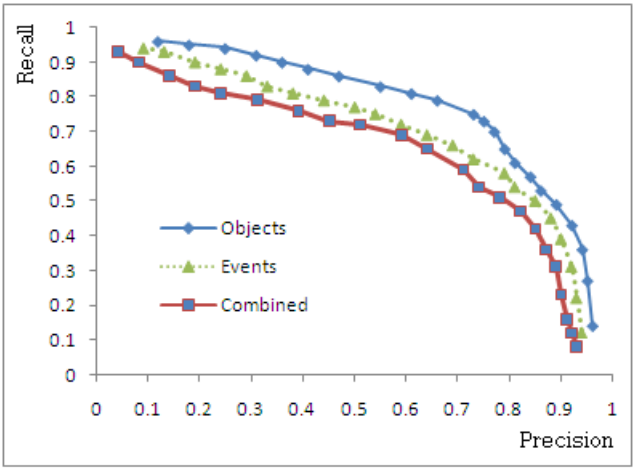


Figure 6.11. Retrieval performance measure by precision-recall curves in terms of objects, events and combined cases with objects under certain events.

6.6 Summary

In this chapter, extraction of video highlights in compressed videos using low level features is presented, which can be applied for automatic annotation and content based retrieval applications. It is found that features obtained from MPEG videos using the DCT domain and YUV colour spaces yielded quite good performance for shot detection, camera motion determination and statistical skin detection. Experiments with a variety of test sequences have successfully demonstrated the effectiveness of the proposed techniques. As the whole system is implemented in the compressed domain, real-time processing will be enabled for many potential applications. Further investigations will be undertaken for face detection and recognition from the detected skin candidates to improve highlight extraction and semantic video indexing, retrieval and annotation.