A LOW COMPLEXITY APPROACH TO SEMANTIC CLASSIFICATION OF MOBILE MULTIMEDIA DATA

A Thesis

Submitted to the Faculty

of

Purdue University

by

Ashok Raj Kumaran Mariappan

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2006

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor Professor Edward J. Delp for his invaluable guidance, support. Thank you Prof. Delp for giving me the opportunities to work in several projects in the Video and Image Processing Laboratory (VIPER) Lab, a true world-class research lab. I believe I have learned a lot from you, academic and otherwise. I truly enjoyed working with you.

I would like to acknowledge the contribution of Michael Igarta who worked on “Indoor/ outdoor,” “Face/ not face” classification for mobile images.

I would also like to thank my graduate advisory committee members Professors Mark R. Bell and Yung-Hsiang Lu for supporting my work.

This work was sponsored by a grant from Multimedia Research Lab, Motorola Labs. I would like to thank Dr. Cuneyt Taskiran and Mr. Bhavan Gandhi for their support, suggestions and criticisms.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Mariappan, Ashok Raj Kumaran M.S.E.C.E., Purdue University, August, 2006. A Low Complexity Approach to Semantic Classification of Mobile Multimedia Data. Major Professor: Edward J. Delp.

With the proliferation of cameras in handheld devices that allow users to capture still images and video sequences, providing users with software tools to efficiently manage multimedia data has become essential. In many cases users desire to organize their personal multimedia data in a way that exploits the content of the data. In this dissertation we describe low-complex algorithms that can be used to derive semantic labels: "indoor/ outdoor," "face/ not face," for mobile images and mobile video sequences, and also "motion/ not motion" label for mobile video sequences.We also describe a method for summarizing mobile video sequences. These algorithms have been developed with the goal of being able to derive the semantic labels on the mobile terminal without any offline computation. We demonstrate the classification performance of the methods with a test image and video database and demonstrate their computational complexity using a typical processor used in many mobile terminals.

1. INTRODUCTION

The capabilities of handheld devices have grown tremendously with their popularity in the recent times. Most of the handheld devices today feature a digital camera capable of capturing still images at a resolution of 1 megapixels (MP) and video sequences of QCIF resolution or less. This combined with the increasing digital data storage capabilities in hand held devices, has resulted in users capturing and storing images and video sequences in handheld devices in the order of hundreds and it is a non-trivial issue organizing the multimedia data. In mobile devices with camera, the acquired images and video sequences are typically organized based on a combination of one or more of the following features:

- Time
- Location, i.e., GPS information
- User's input, i.e., any additional information that user provides

These features method may not be the best for most efficient organization or retrieval of organization of multimedia data. In many cases users desire to cluster the multimedia data in a way that exploits the actual content of the multimedia data. For example, users may want to view video sequences that were taken outside or video sequences that contain familiar faces. Such functionality requires the availability of features describing the content of the multimedia data, i.e., semantic features. A media browsing system that would support this functionality will have three components:

- Low-level feature extraction
- Classification to derive the semantic labels

- Presentation to user via a graphical interface

The first two components are the focus of this dissertation.

1.1 Pseudo-Semantic Labeling

High-level true semantic labels such as “young girl running” and “park scene” characterizes multimedia data based on its content. Ideally, such semantic labels might provide the most useful descriptions for indexing and searching visual content. Currently, however, automatic extraction of truly semantic features is a challenging task. Most approaches in content-based retrieval rely on either low-level models such as color and edges, or domain-specific models like anchor shot models in news video sequences. While low-level features are easy to derive, they do not yield adequate results for many applications. Pseudo-semantic labeling bridges the gap between low-level and true semantic labels.

The pseudo-semantic labels that we automatically generate for mobile images are:

- Indoor/ Outdoor
- Face/ Not Face

The pseudo-semantic labels that we automatically generate for mobile video sequences are:

- Indoor/ Outdoor
- Face/ Not Face
- Motion/ Not Motion

In addition to the pseudo-semantic labels for mobile video sequence we have developed an automatic video summarization system. This system automatically generates key frame summaries. This would enable users to skim through the video content without actually viewing the video sequence.

An overview of a pseudo-semantic media browsing system is shown in Figure 1.1.

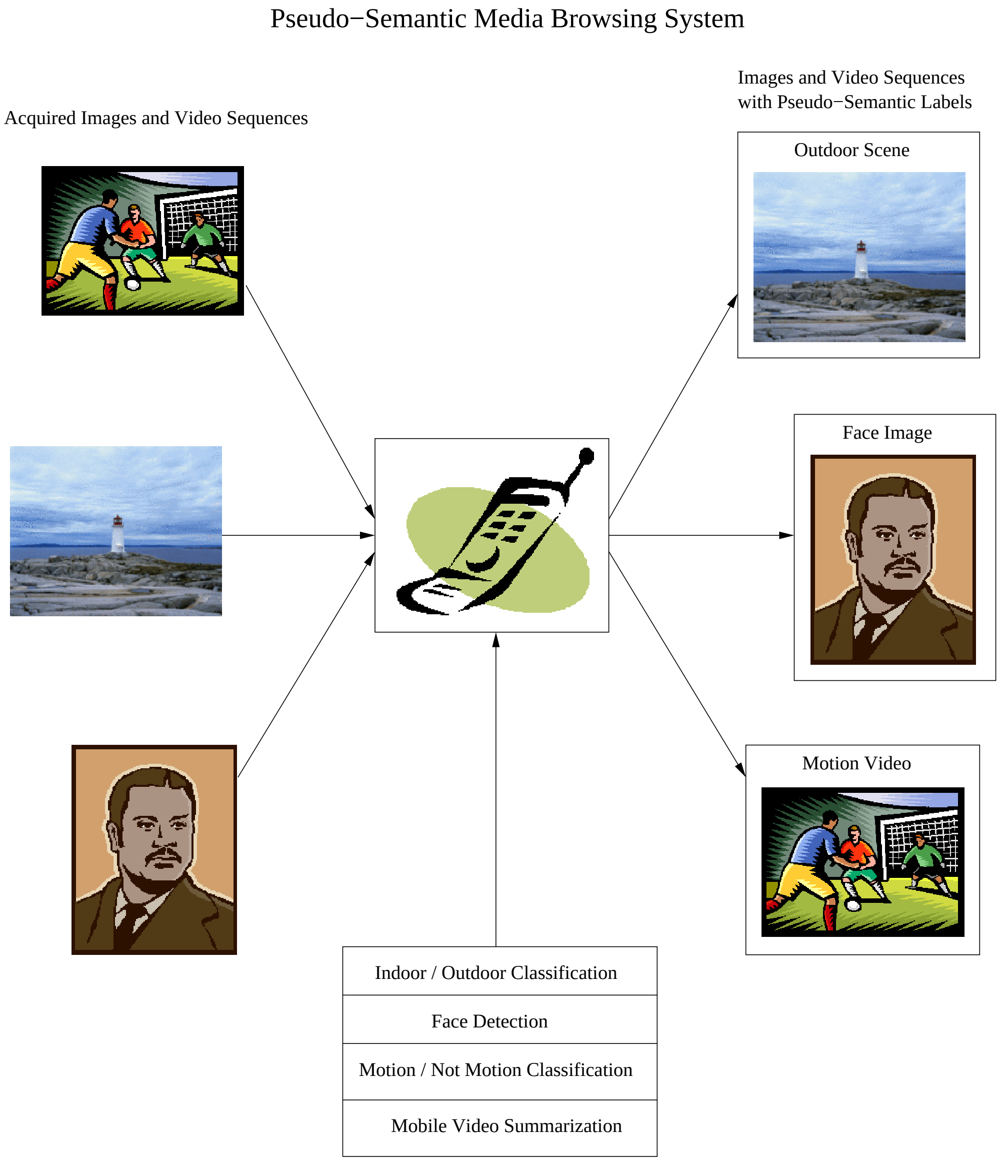

Pseudo−Semantic Media Browsing System
Acquired Images and Video Sequences
Images and Video Sequences
with Pseudo−Semantic Labels
Outdoor Scene
Face Image
Motion Video
Indoor / Outdoor Classification
Face Detection
Motion / Not Motion Classification
Mobile Video Summarization

Fig. 1.1. Overview of mobile device based pseudo-semantic media browsing system.

In Figure 1.1, acquired still images and snapshots of video sequences are shown in the left side of the image. Images and video sequences with automatically derived semantic labels are shown in the right side of the image. These pseudo-semantic labels can then be used to derive higher level semantic labels for the multimedia data. For example a video sequence with the labels “outdoor” and “motion” labels may indicate that the video sequence has a “sports” label.

1.2 The Need for Low Complex Algorithms

In order to derive such semantic classification it is not reasonable to expect the user to devote offline computation time for the multimedia data that are stored in a mobile terminal. Hence the labels have to be derived on the mobile terminal. This adds a restriction to the classification algorithms , which is that the algorithms should not be computationally intensive. The reasons for this are:

- mobile phones typically have slower processors in comparison to a personal computer
- the memory available to execute user programs is low
- the power on a mobile phone is limited, i.e., mobile phones operate on batteries
- the user would like to have the semantic label generated as fast as possible on his phone

Hence, the classification algorithms have to be of low complexity which would enable the user to derive these semantic labels on the mobile device without any offline computation. In [1], we examined semantic labels “face/ not face,” “indoor/ outdoor” that can be used for images. In [2], we examined semantic labels “face/ not face,” “indoor/ outdoor” and “motion/ not motion” that can be used for mobile video sequences. In this dissertation we describe our complete approach to low-complexity techniques for pseudo-semantic labeling for both mobile images and mobile video sequences based on contents.

In order to test the classification performance of the algorithms, we created a test database of images and video sequences. The database is described in detail in Chapter 6. All the video sequences in the test database are of 3gp video format.

3GP is a multimedia container format defined by 3rd Generation Partnership Project (3GPP) for use on Third Generation (3G) mobile phones [3]. It is a simplified version of MPEG-4 (Part 14). MPEG-4 is a standard to compress audio and video digital data developed by the Moving Pictures Experts Group (MPEG) [4], which is a working group of International Organizations of Standards (ISO) [5] / International Electrotechnical Commission (IEC) [6].

1.3 Contributions of this Dissertation

- **Low-Complex Mobile Based Algorithms**

  With most of the content that users create being stored on the mobile device, it is only natural that the semantic details be derived on the mobile device. We have explored the possibility of doing this. This required that the algorithms we develop be of low complexity.

- **"Indoor/ outdoor" Classification**

  We developed low complex algorithm for "Indoor/ outdoor" classification. We were able to achieve a classification accuracy of 85% for mobile images and a classification accuracy 75% for mobile video sequences. The execution time for this algorithm is less than 5 seconds for typical mobile image and mobile video sequence. This make the algorithm an ideal choice for semantic classifiers that can be used.

- **"Face/ not face" Classification**

  We developed low complex algorithm for "Face/ not face" classification. We were able to achieve a classification accuracy of 81% for mobile images and a classification accuracy of 71% for mobile video sequences respectively. The execution time of this algorithm is 3 seconds for a typical mobile image, and is around 20 seconds for a typical mobile video sequence. The execution time is high due to the fact that the entire frame of all examined frames are searched for faces of all possible size. This can be greatly improved by restricting the search region, face size, and using face tracking algorithms.

- **"Motion/ not motion" Classification**

  We developed low complex algorithm for "Motion/ not not" classification for mobile video sequences. We were able to achieve a classification accuracy of 87% for our test database. The execution time for this algorithm is 7 seconds for a typical mobile video sequence.

- **Mobile Video Summarization**

  We developed low complex algorithm for mobile video summarization based on two features standard deviation and histogram intersection of intensity of video frames. The execution time is around 8 seconds for a typical mobile video sequence. This can be used to generate story-board like summaries which would enable users to skim through the video sequences without actually viewing the video sequences.

This dissertation is organized as follows. In Chapter 2 we describe the algorithm for "indoor/ outdoor" classification. In Chapter 3 we describe the algorithm for face detection. In Chapter 4 we describe the algorithm for "motion/ not motion" classification. In Chapter 5 we describe the algorithm for automatic video summarization for mobile video sequences. In Chapter 6 we describe the test database used and the set of experiments performed on our test image and video database. Also, in Chapter 6 we describe the implementation of these techniques on a processor used in many mobile terminals.

2. INDOOR OUTDOOR CLASSIFICATION

2.1 Previous Work

Some of the earliest work in the area of “indoor/ outdoor” scene detection was performed by Picard [7]. In this scheme, color histograms and texture features were used to classify the images. Using 32-bin histograms in the Ohta color space [8] and the nearest neighbor classification rule, a 73.2% image classification performance was reported. Using a combination of multi-resolution simultaneous autoregressive model (MSAR) for texture features and the color histograms they achieve an overall classification rate of 90.3%. In [9] a Bayesian network is used to integrate low-level color and texture features and semantic sky and grass features.

Another technique reported in [10] uses a two-stage classifier using two features. Using a support vector machine (SVM) classifier [11], the algorithm independently classifies image sub-partitions according to color in the LST color space and texture features using wavelet coefficients. The classified sub-partitions are then used by the second stage SVM classifier to determine a final “indoor/ outdoor” decision. This algorithm achieves an overall classification rate of 90.2% on a database of 1200 images. In [12], metadata from camera such as exposure time, flash fired, and subject distance are used in a Bayesian Network for semantic classification. In [13], Binary Bayesian Classifiers to do a hierarchical classification into “indoor/ outdoor” class, “city/ landscape” class. The above techniques are computationally complex and in some cases require multiple passes through the image. These are not suitable for our goal of being able to do the classification on the mobile device.

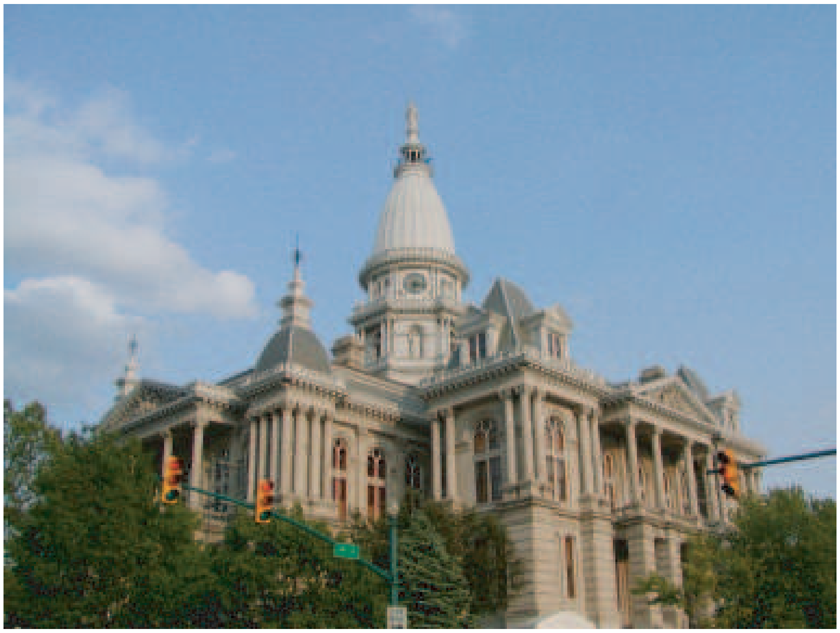


Fig. 2.1. A typical outdoor scene with sky.

2.2 Detection Method

Our approach in developing a lightweight algorithm was to exploit color characteristics that would be considered a “typical” indoor and outdoor image. A typical outdoor image is shown in Figure 2.1 with a large blue sky background occupying the upper portion of the image. Our “indoor/ outdoor” label attempts to detect the presence of blue sky in the upper portion of the image.

We examined the red, green and blue components (RGB) in images, and determined that they do not show any obvious separation making them unsuitable for “indoor/ outdoor” classification. We then examined the YC_rC_b space. A scatter plot of the mean values of C_r and C_b color components for the images in our database is shown in Figure 2.2. These color features will form the basis of our “indoor/ outdoor” label derivation. From the results in Figure 2.2, the problem is generalized to a two-dimensional linear classification problem as shown in Figure 2.3.

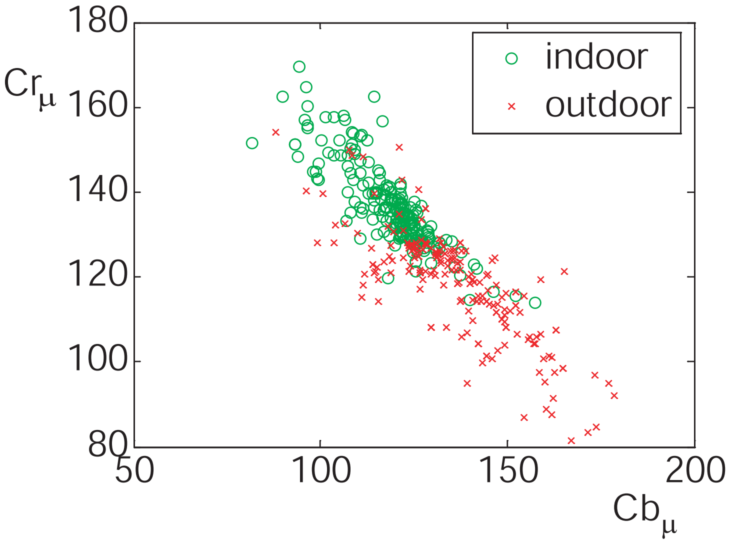


Fig. 2.2. Scatter plot of C_r and C_b color components.

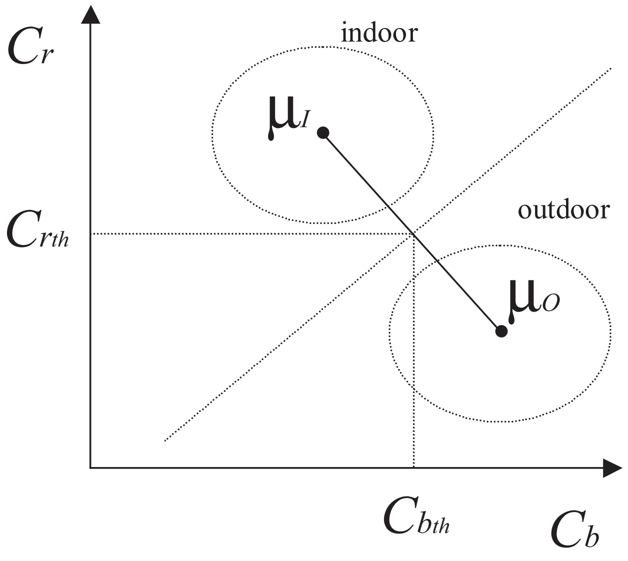


Fig. 2.3. “Indoor/ outdoor” classification based on scatter plot of C_r and C_b color components.

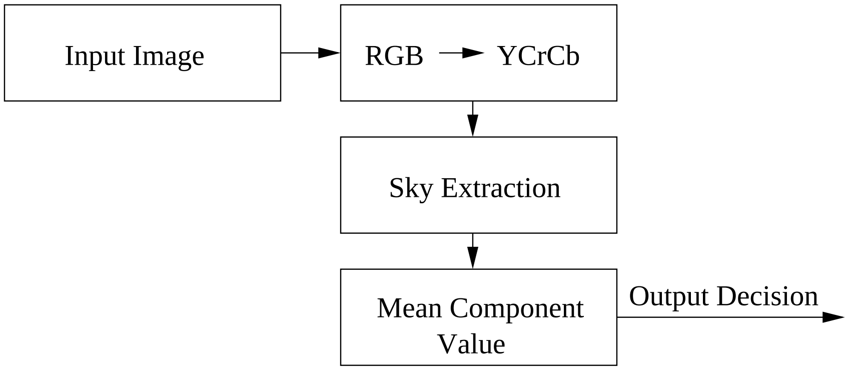


Fig. 2.4. “Indoor/ outdoor” classification algorithm for still images.

Given the clear separation between indoor and outdoor images in terms of C_RC_B values and with the motivation to develop low complex algorithm, the two-dimensional linear classification problem is reduced to a one-dimensional linear classification problem using a single chrominance component.

2.2.1 Image Classification

Given an image in the RGB color space, pixel values are converted to the YC_RC_B color space. Using a single chrominance component, i.e. C_R or C_B, the mean value of the top $X\%$ of the image, which corresponds to the sky region, is obtained (we describe the derivation of the value X below.) The mean is compared with a predetermined threshold to obtain the final “indoor/ outdoor” classification.

The schematic diagram for classification of a still image is given in Figure 2.4.

2.2.2 Video Classification

To classify a video sequence, frames are extracted from the 3gp video sequence by examining one frame per second. Each of these frames are processed for “indoor/outdoor” detection. The mean value of the top $X\%$ of C_r component of each frame, which corresponds to the sky region, is obtained. The mean is compared with

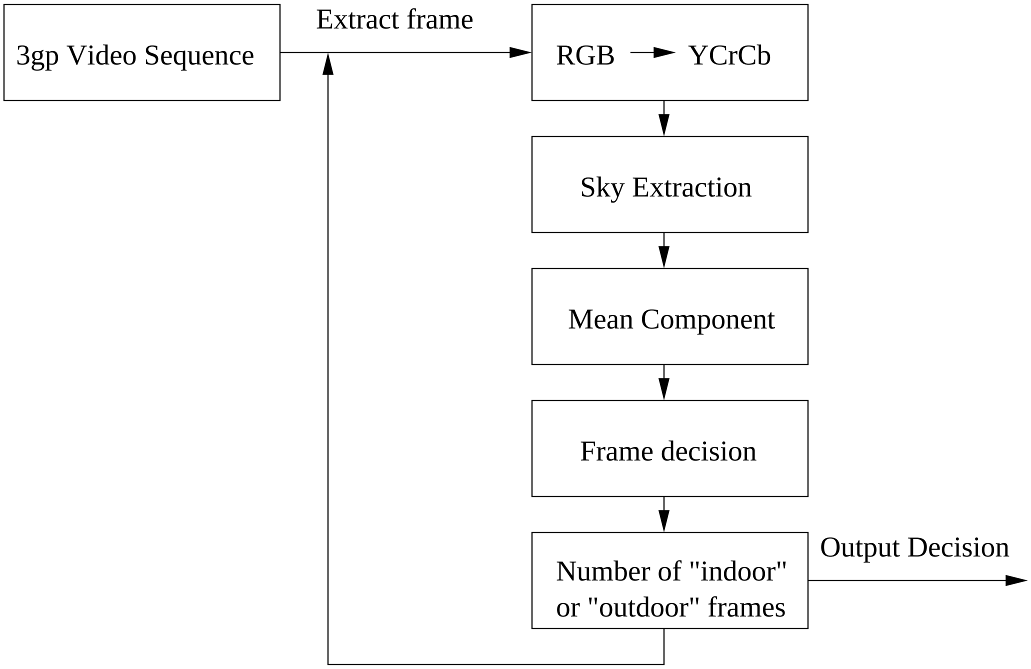


Fig. 2.5. “Indoor/ outdoor” classification algorithm for video sequences.

a pre-determined threshold of the chrominance C_r to determine if the frame is “indoor” or “outdoor” frame. If the number of “outdoor” frames is greater than the number of “indoor” frames, then the video sequence is classified as “outdoor” video, else it is classified as “indoor” video. The schematic diagram for classification of a 3gp video sequence is shown in Figure 2.5.

2.3 Training Method

A training database of 200 images, described in Section 6.1, was used to determine the optimum thresholds of the mean component and the optimum sky fraction $X\%$. The optimum thresholds that separate the “indoor” and “outdoor” images were obtained using leave-one-out cross-validation. The fraction of the image considered to be “sky” was varied from $10\% - 50\%$. The mean value of C_r component in the top 35% of the image was empirically determined to be the optimal for the training database. The test database used to evaluate the algorithm and the experiments performed are presented in Chapter 6.

3. FACE DETECTION

3.1 Previous Work

The survey work presented in [14] summarizes the existing methods for face detection as follows:

- Knowledge-based methods, which are rule based methods that encodes human knowledge of what constitutes a human face

- Template matching methods, where several standard patterns of face are stored, and decisions are made based on correlation between input image and the template

- Feature invariant approaches, that aim to find structural features that exist even when pose, viewpoint, lighting conditions vary

- Appearance-based methods, in which the template is learned from a set of training images

The work presented in [15, 16] uses a Gaussian mixture model to detect skin, perform unsupervised segmentation, and iteratively merges "skin-like" regions to detect faces. Recursive steps, involving histogram analysis, are used to extract skin-color distribution in [17]. Skin patches are detected based on color information, and face candidates are generated based on the spatial arrangement of the skin patches in [18]. In [19], a dynamic programming approach is taken to detect faces in video sequences based on template matching method. In [20], the authors use a distribution-based distance measure and support vector machine (SVM) to detect face based on 1-D Haar wavelet representation of input image. In [21, 22], the learning technique Adaptive Boosting [23, 24] is used to obtain a small number of critical features from a very

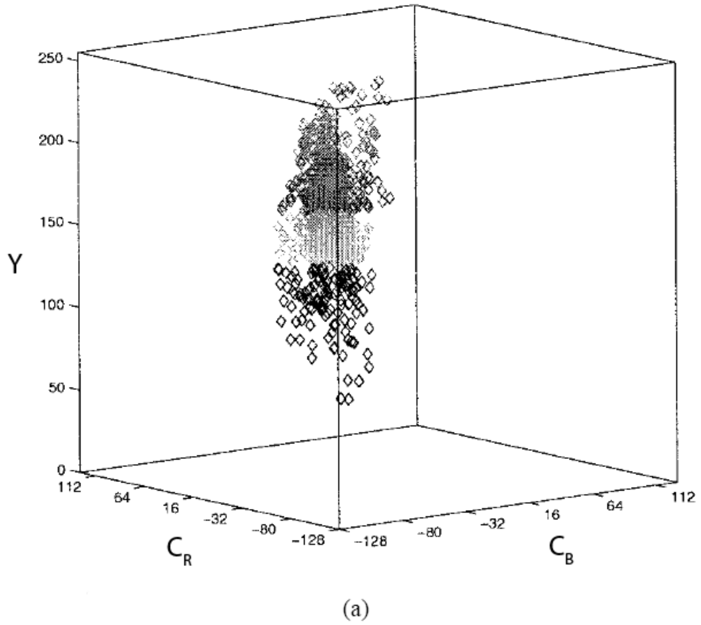


Fig. 3.1. "Skin-like" pixels are highly correlated in the Cr/Cb color space.

large set of potential features. This method requires training data for "face" and "not face" class in the order of ten thousands of images. The face detection work in [25] determined "skin-like" pixels to be highly correlated in the C_r and C_b components and less dependent upon the Y component. They proposed a skin color model which is not dependent upon illumination or relative lightness/darkness of skin-tone. This clustering in C_r and C_b components is illustrated in Figure 3.1.

3.2 Skin Detection

The first step in our face detection approach is to create a binary mask of the original image for "skin-like" pixels. Our method for detecting "skin-like" pixels is similar to the method described in [25], where thresholds are determined empirically for finding skin pixels in the HSV and YC_RC_B color spaces. Similar performance was reported for both color spaces, but our results show slightly better performance for the YC_RC_B color space.

In order to create the binary mask, first the original RGB image is converted to a YC_RC_B image. The C_RC_B components of the pixel values are then thresholded and a pixel is considered to be "skin-like" if it satisfies the following constraints:

$$\begin{aligned}
C_R &\geq -2(C_B + 24), \\
C_R &\geq -4(C_B + 32), \\
C_R &\geq -(C_B + 17), \\
C_R &\geq 2.5(C_B + \theta_1), \\
C_R &\geq \theta_3, \\
C_R &\geq 0.5(\theta_4 - C_B), \\
C_R &\leq \frac{220 - C_B}{6}, \\
C_R &\leq \frac{4}{3}(\theta_2 - C_B),
\end{aligned} \tag{3.1}$$

where the constants $\theta_1, \theta_2, \theta_3, \theta_4$ are given by

$$\begin{aligned}
&\text{for } Y > 128 \\
\theta_1 &= -2 + \frac{2 - Y}{16}, \\
\theta_2 &= 20 - \frac{256 - Y}{16}, \\
\theta_3 &= 6, \\
\theta_4 &= -8, \\
&\text{for } Y \leq 128 \\
\theta_1 &= 6, \\
\theta_2 &= 12, \\
\theta_3 &= 2 + \frac{Y}{32}, \\
\theta_4 &= -16\frac{Y}{16}.
\end{aligned} \tag{3.2}$$

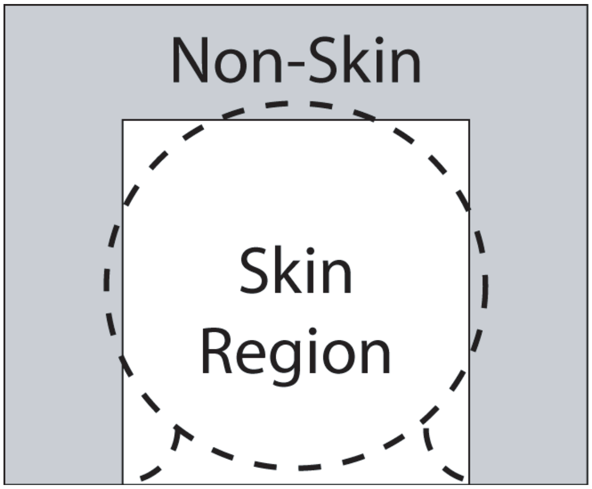


Fig. 3.2. Template used to find faces in images.

3.3 Block Level Processing

For faster processing, the binary mask after skin detection is sub-sampled into 16 × 16 blocks. Since we are interested in regions containing "skin-like" pixels with a 3 × 3 minimum block size, this downsampling of the binary image does not affect our overall results. A 3 × 3 median filter is used on the binary mask image to remove noise.

3.4 Face Template Matching

Our algorithm attempts to match a "typical face" using a pre-defined template, similar to the method described in [26]. We define a typical face as a region of skin pixels with the left, top, and right sides consisting of non-skin pixels. For example, a face in an image would be surrounded by non-skin pixels, such as hair or background. This concept is illustrated in Figure 3.2. In this figure the face, which consists of skin-pixels, is represented by the non-shaded area. The shaded area surrounding the face represents non-skin pixels.

If the fraction of pixels in the face area, that are skin, exceeds the threshold T_{FA} and if fraction of pixels in the face border area, that are non-skin, exceeds the threshold T_{FB} then the area is a candidate face region.

An aspect ratio of 1.75 is used for the rectangles of the face template similar to [26]. The smallest size face that we attempt to detect is 48×48 pixels (3×3) blocks and the maximum size is the size of the image. The face template is moved across the image, and if the thresholds T_{FA} and T_{FB} are satisfied, the region bounded by the template is a candidate face region. These thresholds can be easily determined by counting the number of ones in the binary mask image, and hence it is a low complexity procedure.

3.5 "Face/ Not Face" Decision

The final "face/ not face" classification decision is based on the number of candidate face regions present in the image. This approach is motivated by the observation that face images contained a large number of candidate face regions and non-face images contained few candidate face regions. If an image contains a face, many candidate face regions exist depending on the size and position of the candidate face region.

3.6 Detection Method

The outline of face detection method is illustrated in Figure 3.3 with an example "face" image.

3.6.1 Image Classification

Binary mask of the input image, based on skin detection, is generated. This binary mask is downsampled to a macroblock image. Any noise that may appear is removed by median filtering the macroblock image. The macroblock image is then matched with a predefined template. This algorithm is outlined in Figure 3.4.

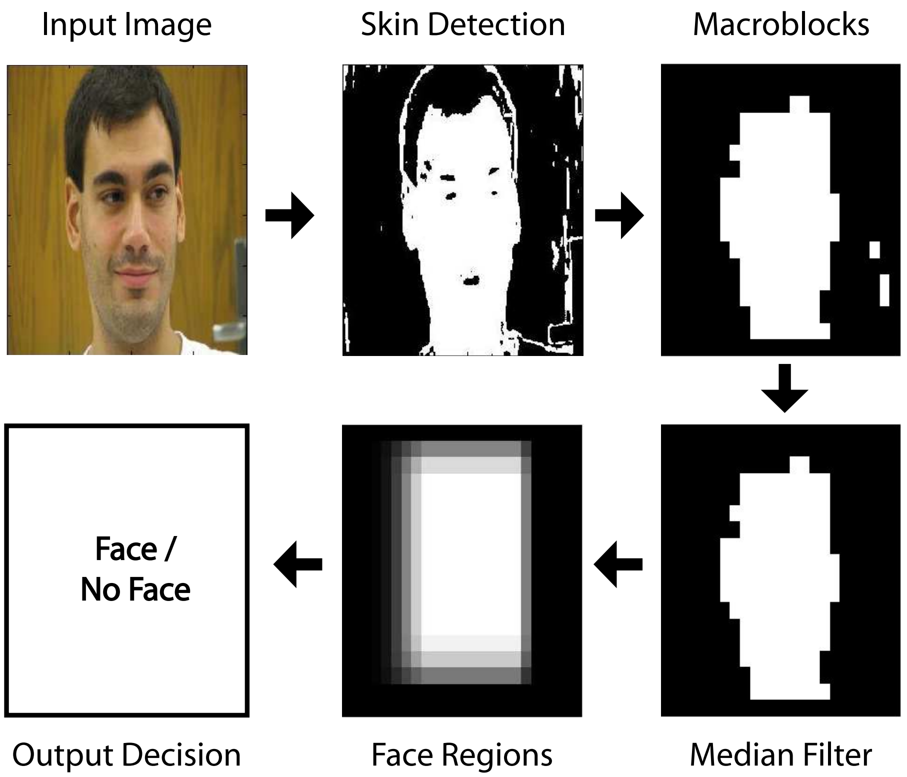


Fig. 3.3. Outline of face detection method.

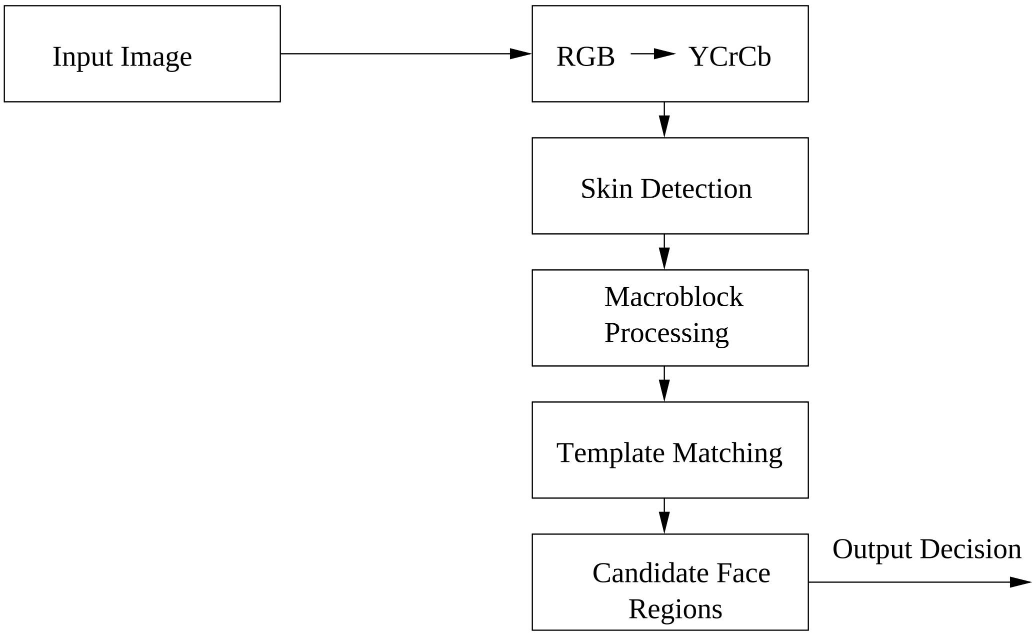


Fig. 3.4. “Face/ not face” classification algorithm for still images.

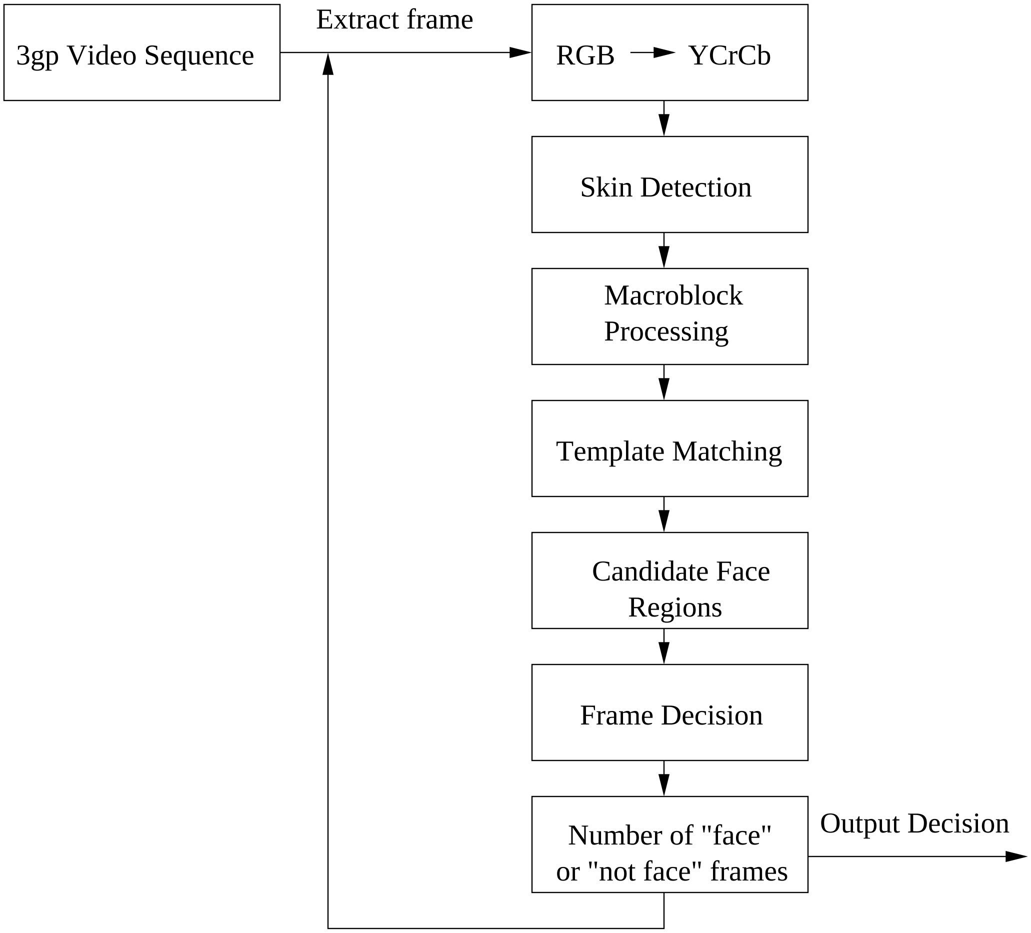


Fig. 3.5. “Face/ not face” classification algorithm for video sequences.

3.6.2 Video Classification

Similar to “indoor/ outdoor” classification, frames are extracted from the 3gp video sequence at a rate of one frame per second, and each frame is processed for face detection. Each frame is classified as “face” frame or “not face” frame. The final video classification decision is based on the number of “face/ not face” frames. If the number of “face” frames is greater than the number of “not face” frames, then video sequence is classified as “face” video else, it is classified as “not face” video. The schematic diagram for classification of a 3gp video sequence is shown in Figure 3.5.

3.7 Training Method

A training database of 200 images, described in Section 6.1, was used to determine the optimum thresholds of T_{FA}, T_{FB}. The optimum thresholds were obtained using leave-one-out cross-validation. The thresholds T_{FA}, T_{FB} were empirically determined to be 0.8 and 0.7 respectively. One macroblock wide face border was used for the face template. The test database used to evaluate the algorithm and the experiments performed are presented in Chapter 6.

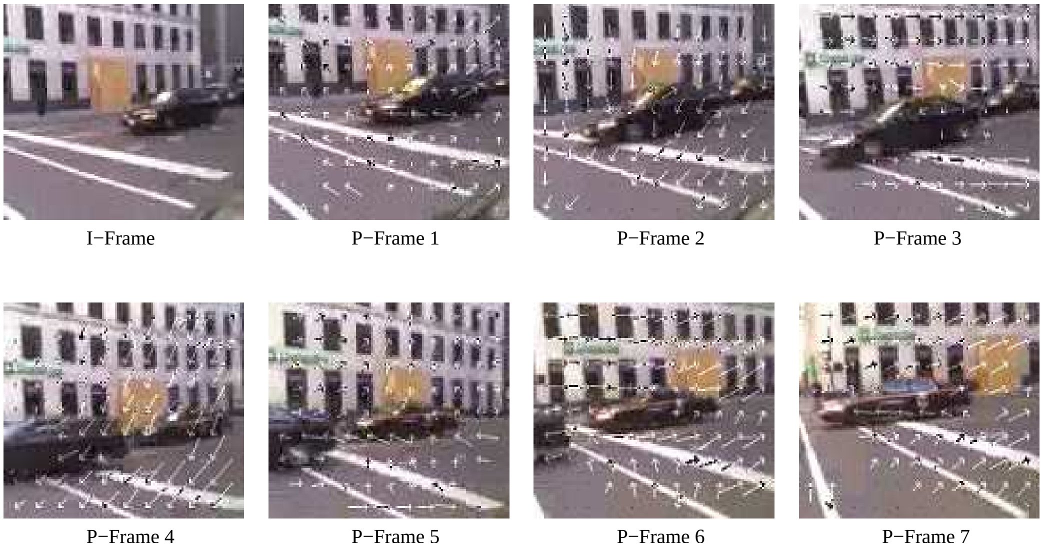
4. “MOTION/ NOT MOTION” CLASSIFICATION

Motion, an essential feature of video data, is an important feature to be considered for video classification. The goal here is to classify the given video sequence as “motion” or “not motion” based on the amount of motion or activity. These labels can be used to derive higher level labels such as “action” or “sports.” In [27] Deng uses motion vector histograms to represent motion, for content based search of video. Ma et. al., [28] propose a motion pattern descriptor that characterizes motion in a generic way, and use Support Vector Machines (SVM) as classifiers. In [29] Ardizzone splits a video sequence into sequence of shots, and extract representative frames and use the representative frames to derive motion information. The work in [30] first explored the use of motion vectors for deriving semantic information about video sequences.

4.1 Classification Method

Our algorithm uses the motion vectors in a 3gp video sequence. A 3gp video is encoded as a sequence of frames, with I (intra) frames and P (predicted) frames. I-frames are the reference frames, and each P-frame is predicted with reference to the I-frame. Each frame is subdivided in to 16 × 16 pixels known as macroblocks. During encoding, motion estimation is done for each macroblocks with respect to the I-frame, and the displacement of each macroblock is stored as a motion vector. In our approach we extract motion vector from each macroblock of each P frame.

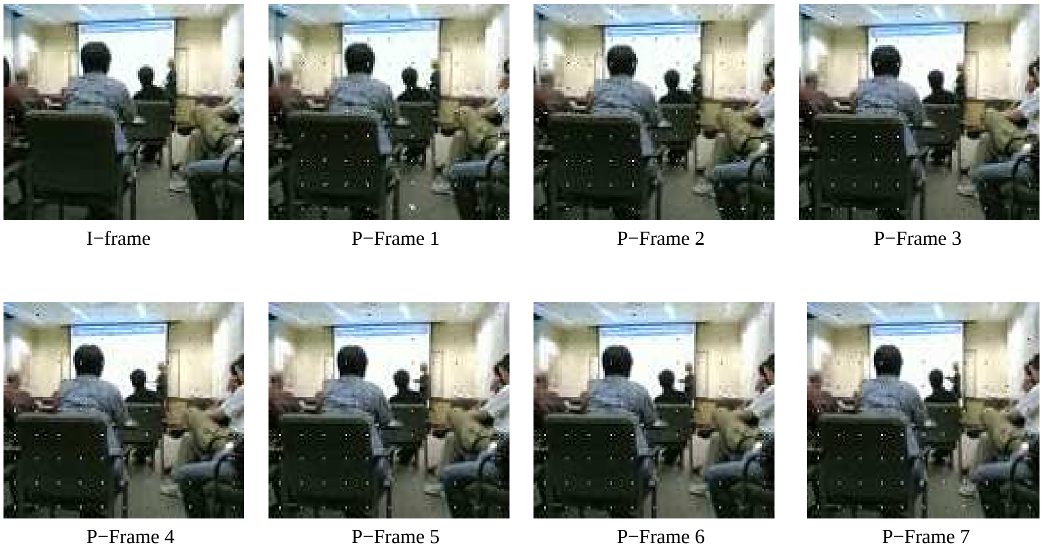
A sequence of frames of the pattern IPPPPPPP, of a video sequence with “motion” label is shown in Figure 4.1. The first frame is an I-frame, and the subsequent frames are P-frames. For each macroblock in the P-frame, its corresponding displacement with respect to the I-frame is shown by arrow whose length and direction represents the motion vector. A similar sequence of a typical video sequence with “not motion”



I−Frame P−Frame 1 P−Frame 2 P−Frame 3

P−Frame 4 P−Frame 5 P−Frame 6 P−Frame 7

Fig. 4.1. Motion vectors with reference to I-frame for an example "motion" video sequence.



I−frame P−Frame 1 P−Frame 2 P−Frame 3

P−Frame 4 P−Frame 5 P−Frame 6 P−Frame 7

Fig. 4.2. Motion vectors with reference to I-frame for an example "not motion" video sequence.

label is shown in Figure 4.2. Here, the macroblocks are not displaced with respect to the I-frame.

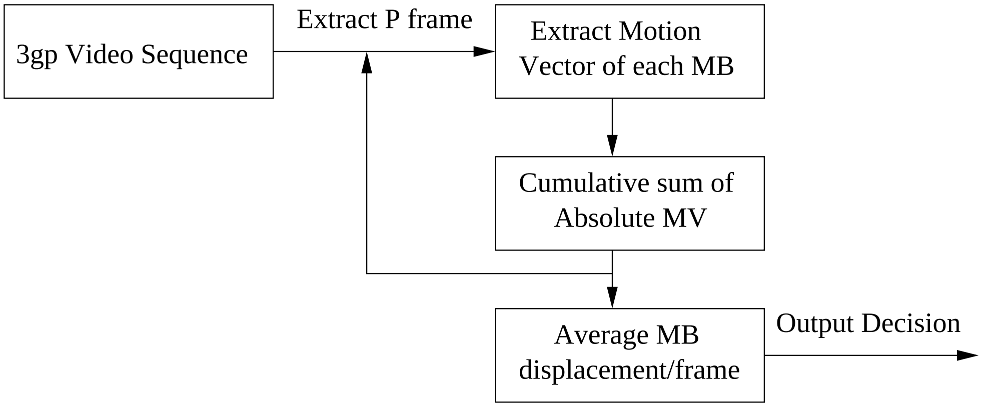

3gp Video Sequence
Extract P frame
Extract Motion Vector of each MB
Cumulative sum of Absolute MV
Average MB displacement/frame
Output Decision

Fig. 4.3. “Motion/ not motion” classification algorithm for video sequences.

For each video sequence we determine the average macroblock displacement per P-frame. We use the following notation:

N - Number of P-frames in a video sequence

P - Number of macroblocks in each P-frame

d_{ij} - Motion vector of macroblock j of frame i

Then the average macroblock motion vector D per P-frame is given by,

$$D = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |d_{ij}|. \tag{4.1}$$

D represents the average number of pixels each macroblock moves in a given video sequence. We used a video database that contained 51 video sequences to determine optimum threshold D_{TH} that separates “motion” and “not motion” sequences. The optimum threshold D_{TH} was determined to be 1.2 pixels/ macroblock/ P-frame. If D is greater than D_{TH} the video sequence is classified as “motion,” else it is classified as “not motion.” The outline of the “motion/ not motion” classification algorithm is show in Figure 4.3. The test database used to evaluate the algorithm and the experiments performed are presented in Chapter 6.

5. VIDEO SUMMARIZATION

5.1 Previous Work

Summaries in terms of “key frames” of the video sequence enable users to skim through the video content rapidly without actually viewing the video sequence. A large number of techniques [31] have been developed for detecting transitions (or shot boundaries) in structured videos, i.e., videos that were created and edited and have clear shot boundaries such as cuts, fades, zoom, etc. The different shot boundary detections methods can be categorized as:

- Methods based on pixel-wise frame difference
- Methods based on color histograms
- Methods based on edge detection
- Methods based on motion vector information
- Model-based techniques

The main problem with video sequences created using mobile telephones, is that the video sequences do not have clear shot boundaries and are highly unstructured. Lienhart in [32] proposed a method for video summarization for personal videos that users create using digital handycam. The method is based on segmenting the “time and date” feature from the video sequence, and using text recognition algorithms, to cluster shots. The clustered shots are shortened using the audio information. However, video sequences obtained using mobile telephones do not have “time and date” information displayed in each frame. In [33], Singular Value Decomposition (SVD) is applied to a three dimensional histograms in RGB color space. Tseng et.

al in [34], have developed a system for video summarization that first generates shot boundaries, and has a feature to add user annotation which is stored in MPEG-7 format. But in this system the shot boundary detection and all the query processing are done offline on a database server, and a video middleware. The work in [35, 36] presents a summarization method for video sequences created using Digital Video Recorders (DVR). This involves classifying the audio data into classes such as excited speech, applause, cheering, music, normal speech using Gaussian Mixture Models (GMM). The audio signal in mobile video is of low quality and is not reliable. Hence the only available information to summarize a video sequence is the visual content.

The simplest approach to summarize the video would be to select frames at regular time intervals. However, for a video in which the visual content does not change or in which the visual content changes too fast, frames selected at regular time intervals would not provide the best summary. The approach we take is to use simple low-level features to derive a dissimilarity metric between frames, and extract representative frames using the dissimilarity metric. This is explained in the following section. Our goal with respect to video summarization is to represent a given video sequence with a minimum number of frames.

5.2 Summarization Method

Histogram analysis and standard deviation based metrics for shot boundary detection have been used extensively for shot boundary detection [31]. We use the generalized trace based on two features: histogram and standard deviation similar to [37]. The choice of these features were motivated by the fact that the two features complement each other's weaknesses. The pixel-based techniques may give false alarm when there the video sequence contains significant camera movement, or moving objects. The histogram-based technique is fairly immune to these effects, but does not detect scene changes if the distribution of luminance does not change significantly.

Given a video sequence, V, composed of N frames represented by $\{f_i\}$. Let $\vec{x}_i = [x_{1i}\ x_{2i}]^T$, be a feature vector of length two extracted from the pair of frames $\{f_i, f_{i+1}\}$. The generalized trace, d, for V is defined as

$$d_i = \|\vec{x}_i - \vec{x}_{i+1}\|_2. \tag{5.1}$$

The first feature dissimilarity measure based on histogram intersection given by the following equation,

$$x_{1i} = \frac{1}{2T}\sum_{j=1}^{K} |h_i(j) - h_{i+1}(j)|, \tag{5.2}$$

where h_i and h_{i+1} are the luminance histograms for frame f_i and f_{i+1}, respectively, K is the number of bins used, T is the number of pixels in a frame.

The second feature used is the absolute value of the difference of standard deviations of the luminance component of the frames f_i and f_{i+1}. It is given as:

$$x_{2i} = |\sigma_i - \sigma_{i+1}|, \tag{5.3}$$

where,

$$\sigma_i^2 = \frac{1}{T-1}\sum_m \sum_n (Y_i(m,n) - \mu_i)^2, \tag{5.4}$$

$$\mu_i = \frac{1}{T}\sum_m \sum_n (Y_i(m,n)). \tag{5.5}$$

To detect scene changes using a dissimilarity metric, several approaches have been proposed based on sliding window and other techniques [31, 38]. In [37], Taskiran considers the shot boundary detection as an one dimensional edge detection problem. But for our goal of choosing representative frames for video sequence of duration less than 180 seconds, these methods are complicated. Hence we normalize the generalized trace and detect scene boundaries based on a global threshold. The global threshold was heuristically chosen to be 0.2. Hence, we declare a new scene s_j, starting at frame f_i, if the difference metric d_i is greater than 20% of the maximum of the difference metric. In order to reduce the false positives, new scenes detected that are less than

F frames apart are not taken in to account. We used a value of 15 for F for this work.

After determining the scene change boundaries we select one representative frame for each scene s_j. The representative frame f_r for the scene s_j is selected such that, d_i is minimum for $i \in s_j$. The test database used to evaluate the algorithm and the experiments performed are presented in Chapter 6.

6. EXPERIMENTAL RESULTS

6.1 Image Database

A subset of images taken from our laboratory database of images was used for our tests. Images that appeared to be naturally indoor or outdoor to the casual viewer were selected. Images that do not distinctly belong in either class were discarded from the test image database. Such images include but are not limited to: space images, computer generated images, hand drawn figures, and maps. The content of the images varies widely. Some images do not contain any visible sky regions. People appeared in both the indoor and outdoor images. The database consists of 400 images, with 200 images used for training and 200 images used for testing. Each image in the database is a 24-bit RGB color image, 8 bits per sample for each color component. Few images from the database are shown in Figure 6.1.

6.2 Video Database

A database of approximately 200 minutes of 3gp mobile video was created using the following devices:

- Motorola A780 mobile phone
- Nokia 6630 mobile phone
- Nokia 6681 mobile phone
- Sony digital handycam

The database consists of several short video sequences with a minimum duration of 15 seconds and a maximum duration of 180 seconds. There are a total of 324 video

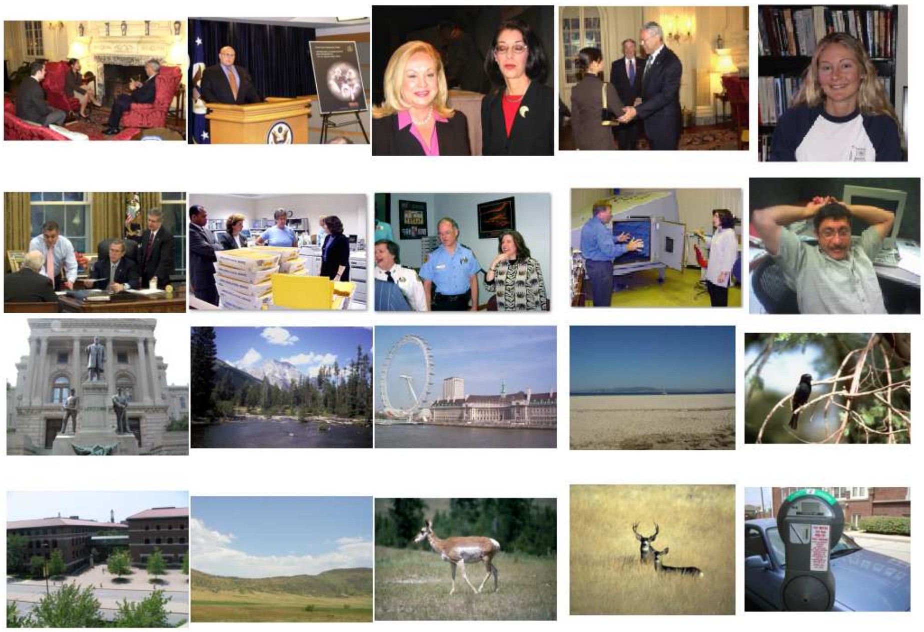


Fig. 6.1. Sample images from database.

Table 6.1
Specification of 3gp video sequences used.

Resolution	QCIF 176 × 144 or less
Frame rate	15 frames/second or less
Data rate	192 kilobytes/second or less

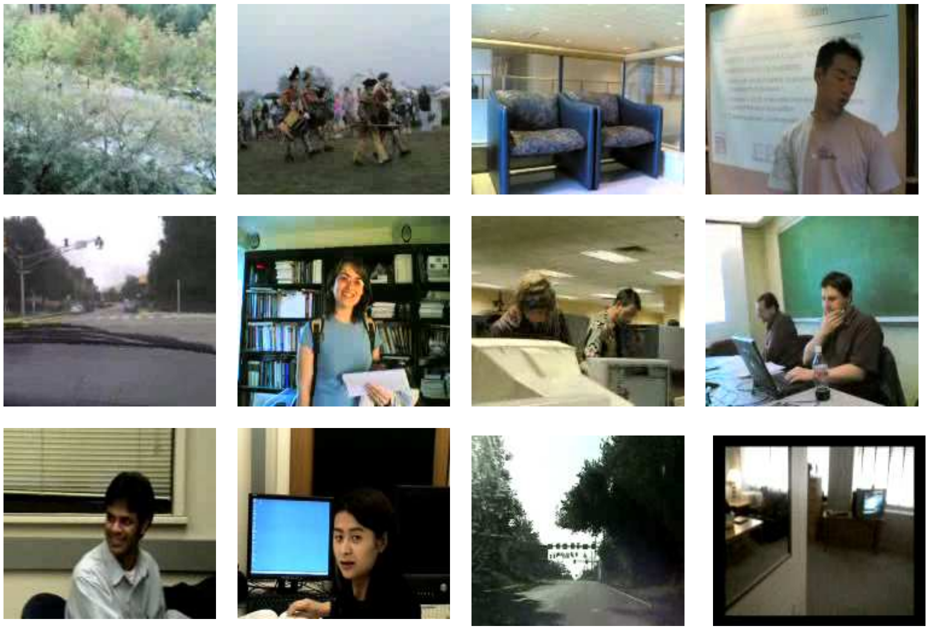


Fig. 6.2. Snapshot from sequences in video database.

sequences. The specification of the video sequences in the database is given in Table 6.1.

The video sequences obtained using the Sony digital handycam were converted to 3gp format with the above specifications using the FFmpeg Multimedia System [39]. Snapshots from a few of the video sequences are shown in Figure 6.2.

6.3 Indoor Outdoor Classification

6.3.1 Image Classification

The test image database of 200 images was classified as 100 “outdoor” images and 100 “indoor” images. Each image was manually labeled as “indoor” or “outdoor” by a human subject and the label was independently verified by another human subject. We were able to achieve a correct classification rate of 85%.

6.3.2 Video Classification

The test video database of 324 video sequences was classified as 174 “outdoor” video sequences and 150 “indoor” video sequences. Each video sequence was manually labeled as “indoor” or “outdoor” by a human subject and the label was independently verified by another human subject. The content of the video sequences varies widely. The presence of sky is mixed in the video sequences. For sequences with mixed content, i.e., part of the video sequence having “indoor” frames and part of the video sequence having “outdoor” frames, if the number of “indoor” frames is greater than the number of “outdoor” frames, then the video sequence is classified as “indoor” video, else it is classified as “outdoor” video.

We were able to achieve a classification rate of 75%. The incorrectly classified “outdoor” sequences were the ones that did not have any sky regions. The top portion of the frames of the incorrectly classified “indoor” sequences have the same color characteristics as an “outdoor” sky region.

6.4 Face Detection

6.4.1 Image Classification

We empirically determined the optimum thresholds T_{FA} as 0.8 and T_{FB} as 0.7 for our database. Using these thresholds we were able to achieve a classification rate of 81%.

A correct "face" classification is shown in Figure 6.3. "Skin-like" pixels are segmented from the background, and face template is used to determine the face regions. In Figure 6.4, the building color is similar to skin color, and skin detection step results in a binary mask with many "skin- like" pixels. But the thresholds for the face template are not satisfied and there are no candidate face regions, which results in a correct "not face" classification. In Figure 6.5, most of the regions in the image have "skin-like" pixels, and the actual face region is merged with the background. As a result, the face template is not able to find "non-skin" border resulting in an incorrect "not face" classification. Another example is shown in Figure 6.6. Here skin like regions that correspond to the hand in the image are detected correctly. But the thresholds for the face template are also satisfied, and this results in an incorrect "face" classification.

6.4.2 Video Classification

The test database was classified as 63 "face" sequences and 261 "not face" sequences. Only sequences with full frontal face, and of size greater than 24×24 pixels were considered for "face" sequences. Each of the video sequence was manually labeled by a human subject and verified by another human subject. For sequences with mixed content, i.e., part of the video sequence having "face" frames and part of the video sequence having "not face" frames, if the number of "face" frames is greater than the number of "not face" frames then the video sequence is classified as "face" video, else it is classified as "not face" video.

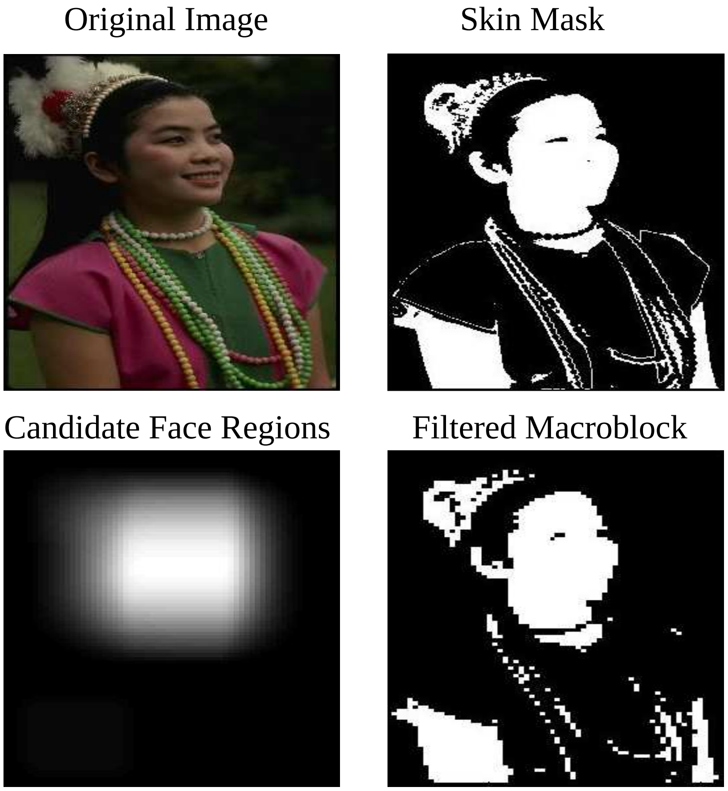


Fig. 6.3. Correct "face" classification example.

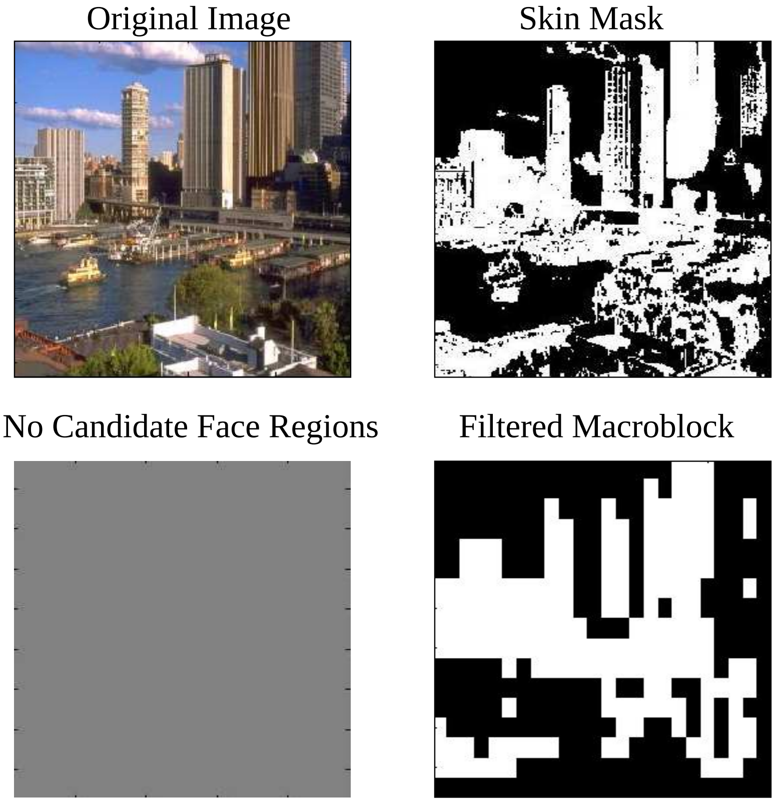
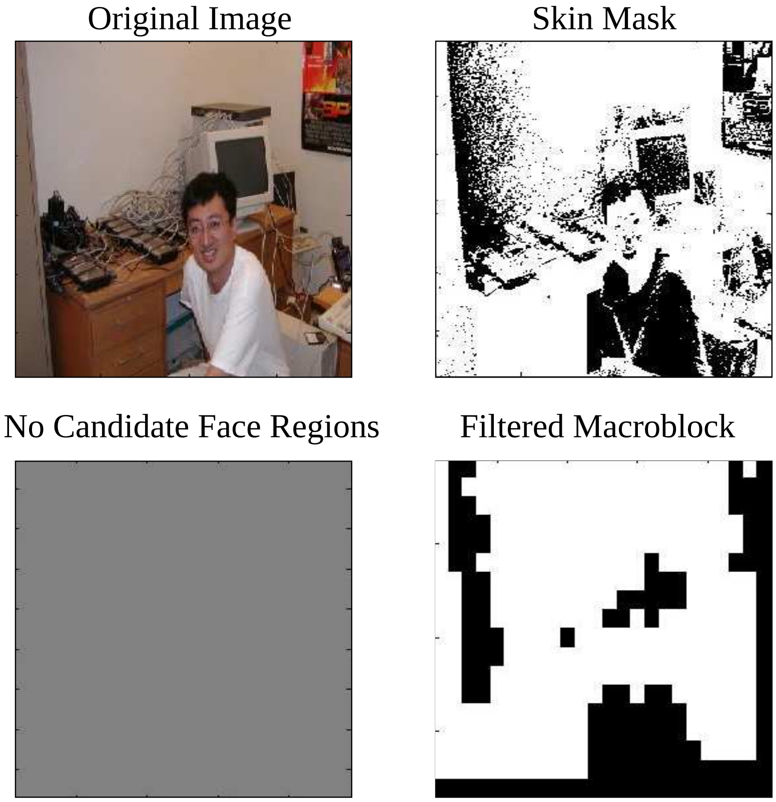


Fig. 6.4. Correct not “face” classification example.



Original Image

Skin Mask

No Candidate Face Regions

Filtered Macroblock

Fig. 6.5. Incorrect “not face” classification example.

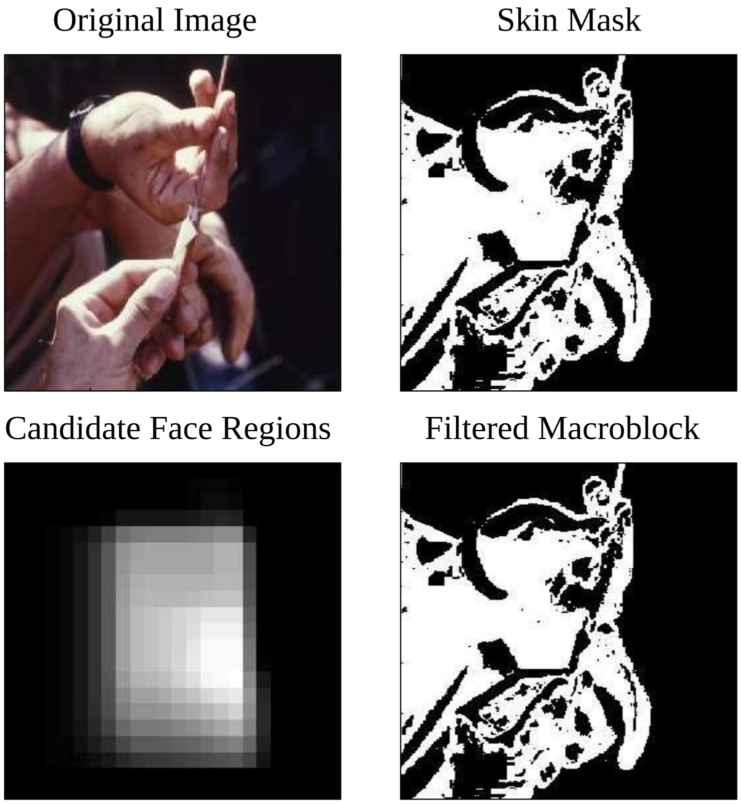


Fig. 6.6. Incorrect “face” classification example.

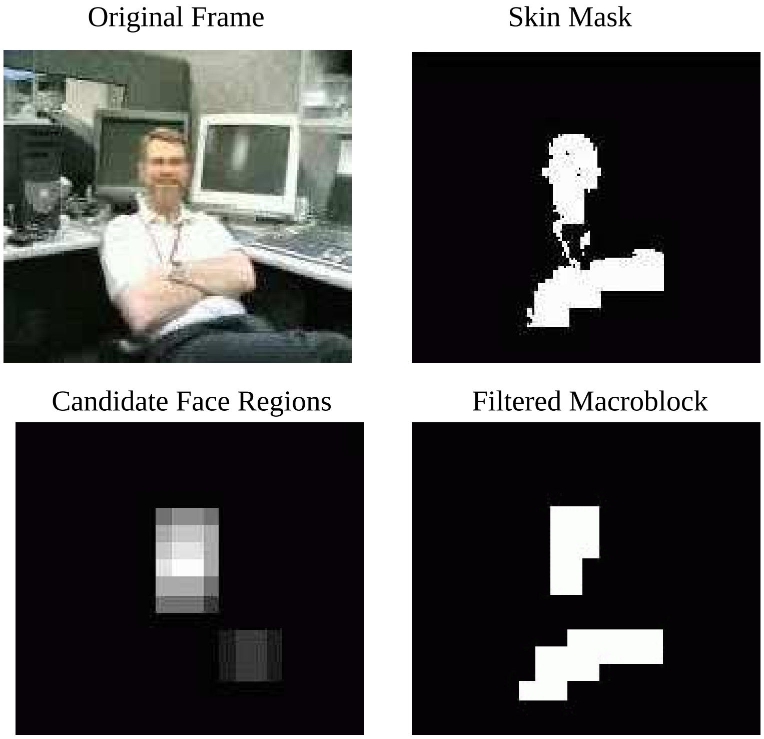


Fig. 6.7. Correct “face” video classification example.

We were able to achieve a classification rate of 71%. A correct “face” classification is shown in Figure 6.7. Here skin detection labels the “skin like” pixels. The face template labels the candidate face regions and the result is a correct face classification. A correct “not face” classification is shown in Figure 6.8. In this frame few false “skin like” pixels are detected. But the “face template” does not label any of the detected “skin like” regions as face and the result is a correct “not face” classification.

6.5 “Motion/ Not Motion” Classification

Only the video sequences that were obtained using the Motorola A780, Nokia 3360 and Nokia 6681 mobile telephones were considered. There were a total of 197 video sequences with 142 “motion” sequences, and 55 “not motion” sequences. Each video sequence was manually labeled as “motion” or “not motion” by a human subject and the label was independently verified by another human subject. For the manual clas-

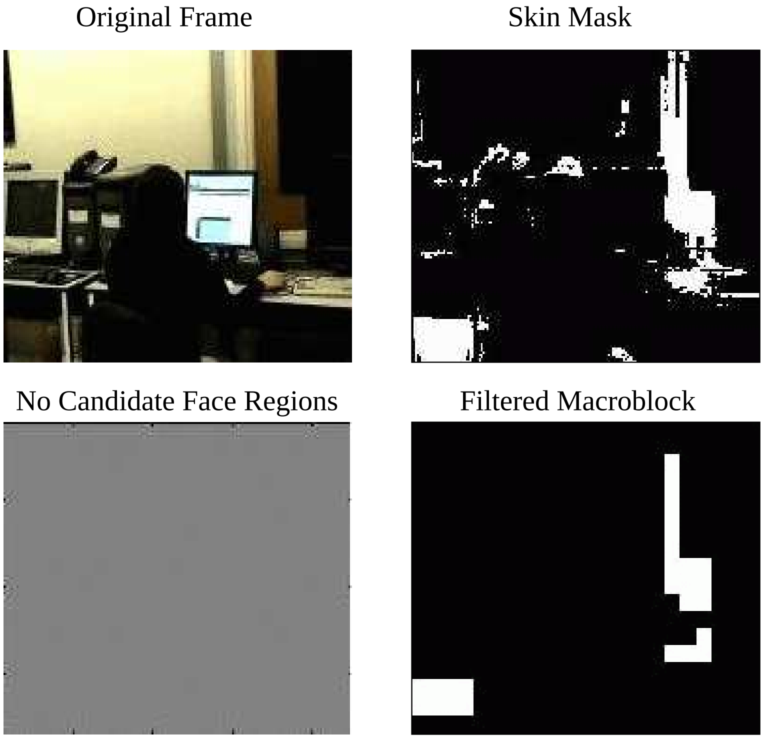


Fig. 6.8. Correct “not face” video classification example.

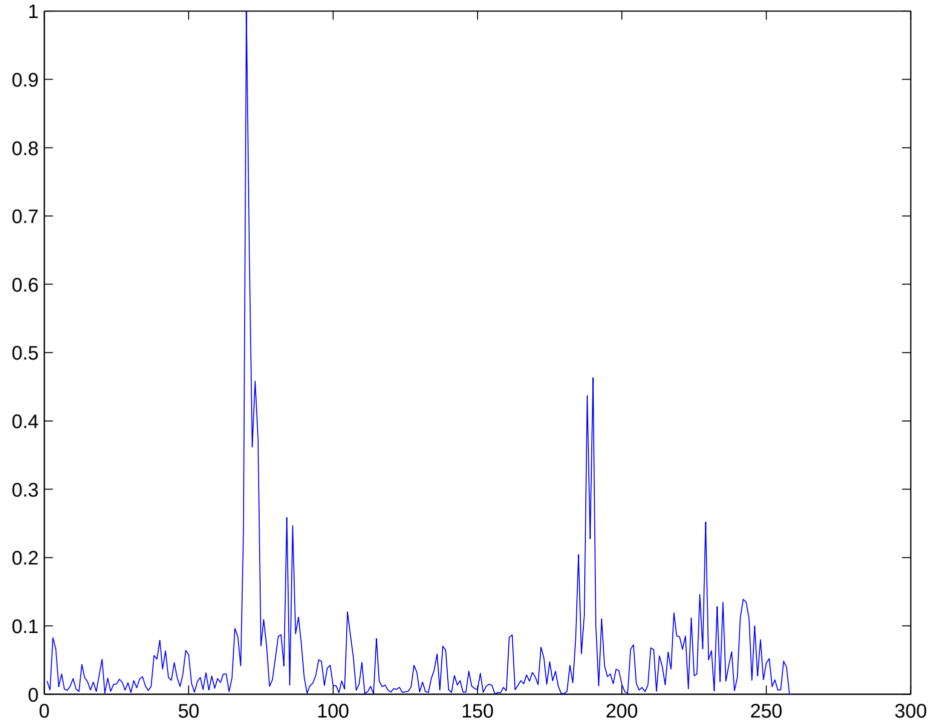


Fig. 6.9. The generalized trace for example video1.

sification, sequences with continuous camera movement for at least half the duration, were considered to be "motion" sequence, and the rest were classified as "not motion" sequence.

Considering the average macroblock displacement D per P-frame given by equation 4.1, the optimal threshold D_{TH} was determined to be 1.2 pixels/macroblock/P-frame for a training database of about 51 video sequences. We were able to achieve a classification result of 87%, using this method. Most of the incorrect "not motion" videos were because of high of camera shake.

6.6 Mobile Video Summarization

The generalized trace based on the histogram and standard deviation of luminance for two sequences in our database is shown in Figure 6.9 and 6.10.

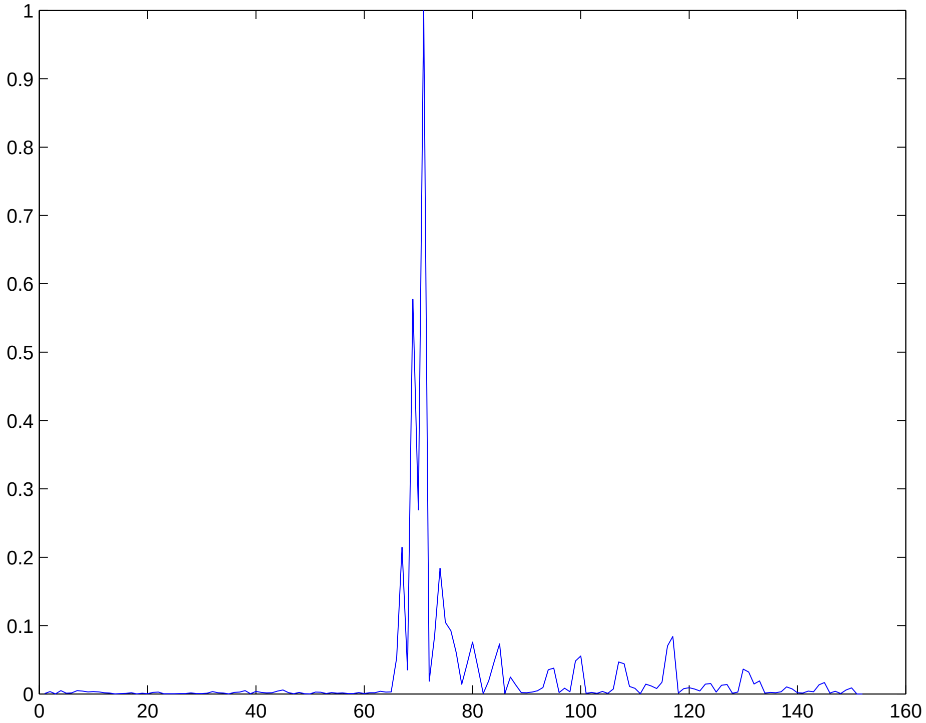
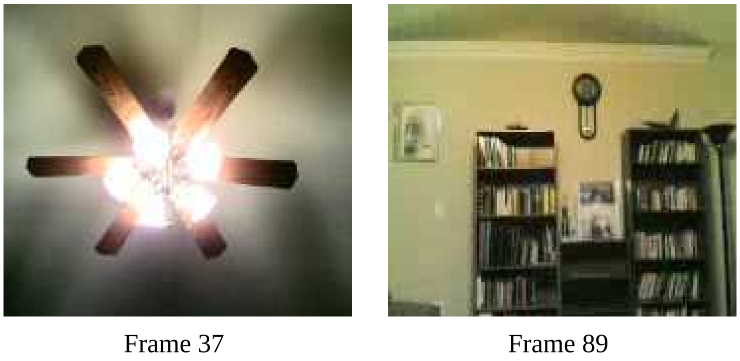


Fig. 6.10. The generalized trace for example video2.



Frame 37

Frame 89

Fig. 6.11. Representative frames determined for example video2.

Table 6.2
Classification results for "indoor/ outdoor," "face/ not face," "motion/not motion" labels.

Label	Image	Video
	Classification Result (%)	Classification Result(%)
Indoor/ Outdoor	85	75
Face/ Not Face	81	71
Motion/ Not Motion	-	87

There are two scenes in video2: scene one from frame 1 to frame 66: $s_1 = \{1, 2, \ldots, 66\}$, scene two from frame 68 to the last frame 152: $s_2 = \{68, 69, \ldots, 152\}$. Based on the method described in Chapter 5, the scene boundary was detected as 67. Two representative frames were determined: frame 38 from s_1 and frame 89 from s_2. They are shown in Figure 6.11.

The classification results for "indoor/ outdoor," "face/ not face," "motion/ not motion" labels are summarized in Table 6.2.

6.7 Target Platform

The goal of this work was to achieve a reasonable classification rate and also be able to label the video sequences on mobile devices without any offline computing.

Table 6.3
Execution time on Compaq iPAQ H3970 handheld PDA.

Label	Input	Execution Time (s)
Indoor/ Outdoor	24-bit RGB color image	1
	30 second 3gp video sequence	5
Face/ Not Face	24-bit RGB color image	3
	30 second 3gp video sequence	20
Motion/ Not Motion	30 second 3gp video sequence	7
Video Summarization	30 second 3gp video sequence	8

The target platform we used to test our algorithms was a Compaq iPAQ H3970 handheld PDA. The metric we chose to evaluate the complexity of our algorithms was execution time on a handheld PDA. This was motivated by the fact that execution time directly relates to the power consumption of the mobile device. The target handheld has an Intel XScale PXA250 processor, running at 400 megahertz, which is based on the ARM architecture [40], and lacks floating point hardware. For memory, it has 32 MB of flash-ROM and 64 MB of SDRAM. The original Microsoft PocketPC operating system was removed and Familiar Linux v0.72 [41] was installed. This is a Linux distribution targeted for the iPAQ series of PDAs. The FFmpeg Multimedia System [39] was used to decode the 3gp sequences.

The execution time for “indoor/ outdoor,” “face/ not face,” “motion/ not motion” labels and video summarization on the Compaq iPAQ H3970 is shown in Table 6.3. These include the time for decoding the 3gp video, extracting label information and making a classification decision. The high execution time for Face Detection for video sequences is due to the fact that the entire frame of the all the frames analyzed is searched for matching faces. This can be improved by restricting the search area for subsequent frames based on occurrence of previous frames.

6.8 Implementation on Mobile Telephone

The algorithms for “indoor/ outdoor” classification, “face/ not face” classification for mobile images were implemented in Java Platform, Micro Edition (J2ME) [42]. J2ME is a collection of Java APIs for the development of software for resource constrained devices such as mobile phones. These Java programs were installed on the mobile phones Motorola A780 and Nokia 6681 that were used to create the video database. The two phones are Mobile Information Device Profile (MIDP) [43] compliant. MIDP is a specification published for use of Java on embedded devices.

The Motorola A780 mobile phone is based on Linux Operating System and it features a Intel XScale PXA270 Processor, with 48 Megabytes of internal memory. The Nokia 6681 mobile phone is based on Symbian Operating System, and it features a 220 megahertz Processor based on ARM architecture. The execution time of “indoor/ outdoor” classification and “face/ not face” classification algorithms were consistent with the results obtained in Table 6.3.

7. CONCLUSION

The proliferation of cameras and low-cost storage mediums in mobile handheld devices have made it possible for users to acquire large number of images and video sequences and store it on the mobile devices. This immediately results in the need for efficient organization of the stored media data, which would enable efficient retrieval and indexing. Semantic labels that describe the contents of the multimedia data, provide the best way to organize the data. In this dissertation, we examined several semantic classification problems for mobile images and video sequences.

7.1 Contributions of this Dissertation

- **Low-Complex Mobile Based Algorithms**

  With most of the content that users create being stored on the mobile device, it is only natural that the semantic details be derived on the mobile device. We have explored the possibility of doing this. This required that the algorithms we develop be of low complexity.

- **"Indoor/ outdoor" Classification**

  We developed low complex algorithm for "Indoor/ outdoor" classification. We were able to achieve a classification accuracy of 85% for mobile images and a classification accuracy 75% for mobile video sequences. The execution time for this algorithm is less than 5 seconds for typical mobile image and mobile video sequence. This make the algorithm an ideal choice for semantic classifiers that can be used.

- **"Face/ not face" Classification**

We developed low complex algorithm for "Face/ not face" classification. We were able to achieve a classification accuracy of 81% for mobile images and a classification accuracy of 71% for mobile video sequences respectively. The execution time of this algorithm is 3 seconds for a typical mobile image, and is around 20 seconds for a typical mobile video sequence. The execution time is high due to the fact that the entire frame of all examined frames are searched for faces of all possible size. This can be greatly improved by restricting the search region, face size, and using face tracking algorithms.

- **"Motion/ not motion" Classification**

  We developed low complex algorithm for "Motion/ not not" classification for mobile video sequences. We were able to achieve a classification accuracy of 87% for our test database. The execution time for this algorithm is 7 seconds for a typical mobile video sequence.

- **Mobile Video Summarization**

  We developed low complex algorithm for mobile video summarization based on two features standard deviation and histogram intersection of intensity of video frames. The execution time is around 8 seconds for a typical mobile video sequence. This can be used to generate story-board like summaries which would enable users to skim through the video sequences without actually viewing the video sequences.

In conclusion, we have developed lightweight algorithms to perform the labeling with relatively good performance, both in terms of classification accuracy and execution time, on a database of approximately 400 images and 200 minutes of 3gp video sequences, that can be run on a handheld mobile device.

7.2 Suggestions for Future Work

- **Compressed Domain Analysis**

Minimal decoding of MPEG video stream can be used to obtain the Discrete Cosine Transform (DCT) coefficients, which is equivalent to a lower resolution version of the video stream. Performing analysis on the DCT co-efficients will result in a further reduction in the complexity of algorithms. However, since the mobile video sequences are not of very high resolution (typically QCIF resolution or less), performing analysis on compressed domain might not yield adequate results. Nevertheless it will be interesting to evaluate the performance in terms of classification accuracy and complexity.

- **Face Tracking**

  The current face detection algorithm for mobile video sequences attempts to search for candidate face regions of all size, throughout the frame, in all the frames. This can be improved by restricting the search area based on the occurrence of face in previous frames.

- **Camera metadata**

  Metadata that can be acquired from camera such as time, GPS location can be augmented to the semantic data. For example, time and GPS information can be combined to determine the weather condition that existed at the time the data (image or video sequence) was obtained. This information can add on significantly to the existing semantic information.

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Mariappan, M. Igarta, C. Taskiran, B. Gandhi, and E. J. Delp, "A low-level approach to semantic classification of mobile multimedia content," in *Proceedings of the 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, pp. 111–117, November/December 2005.

[2] A. Mariappan, M. Igarta, C. Taskiran, B. Gandhi, and E. J. Delp, "A study of low-complexity tools for semantic classification of mobile video," in *Proceedings of the SPIE International Conference on Multimedia on Mobile Devices II*, vol. 6074, pp. 71–82, January 2006.

[3] "Third generation partnership project." `http://www.3gpp.org/`.

[4] "Moving pictures expert group." `http://www.chiariglione.org/mpeg/`.

[5] "International organization of standards." `http://www.iso.org/`.

[6] "International electrotechnical commission." `http://www.iec.ch/`.

[7] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *IEEE International Workshop on Content-Based Access of Image and Video Databases*, pp. 42–51, 1998.

[8] Y. Ohta, T. Kanade, and T. Takai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222–241, 1980.

[9] J. Luo and A. Savakis, "Indoor vs outdoor classification of consumer photographs using low-level and semantic features," in *Proceedings. 2001 International Conference on Image Processing*, vol. 2, pp. 745–748, October 2001.

[10] N. Serrano, A. Savakis, and A. Luo, "A computationally efficient approach to indoor/outdoor scene classification," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 146–149, 2002.

[11] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[12] M. Boutell and J. Luo, "Bayesian fusion of camera metadata cues in semantic scene classification," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004*, vol. 2, pp. 623–630, June/July 2004.

[13] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang, "Content-based hierarchical classification of vacation images," in *IEEE International Conference on Multimedia Computing and Systems, 1999*, vol. 1, pp. 518–523, July 1999.

[14] M.-H. Yang and N. A. D.J. Kriegman, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, May 2002.

[15] A. Albiol, C. A. Bouman, and E. J. Delp, "Face detection for pseudo-semantic labeling in video databases," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 607–611, October 1999.

[16] C. Taskiran, J. Y. Chen, A. Albiol, L. Torres, C. A. Bouman, and E. J. Delp, "ViBE: A compressed video database structured for active browsing and search," *IEEE Transactions on Multimedia*, vol. 6, pp. 103–118, February 2004.

[17] S. Kawato and J. Ohya, "Automatic skin-color distribution extraction for face detection and tracking," in *Proceedings of the 5th IEEE International Conference on Signal Processing*, vol. 2, pp. 1415–1418, August 2000.

[18] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 696–706, May 2002.

[19] Z. Liu and Y. Wang, "Face detection and tracking in video using dynamic programming," in *Proceedings of International Conference on Image Processing*, vol. 1, pp. 53–56, 2000.

[20] P. Shih and C. Liu, "Face detection using distribution-based distance and support vector machine," in *Proceedings of Sixth International Conference on Computational Intelligence and Multimedia Applications*, pp. 327–332, 2005.

[21] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[22] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, no. 2, 2002.

[23] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory*, pp. 23–37, 1995.

[24] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods.," 1997.

[25] C. Garcia and G. Tzirtas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions on Multimedia*, vol. 1, pp. 264–277, September 1999.

[26] H. Wang and S. F. Chang, "A highly efficient system for automatic face region detection in mpeg video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 615–628, August 1997.

[27] Y. Deng and B. S. Manjunath, "Content-based search of video using color, texture and motion," in *Proceedings of the IEEE International on Conference Image Processing*, vol. 2, pp. 534–537, 1997.

[28] Y.-F. Ma and H.-J. Zhang, "Motion pattern-based video classification and retrieval," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 2, pp. 199–208, 2003.

[29] E. Ardizzone, M. L. Casia, and D. Molinelli, “Motion and color based video indexing and retrieval,” in *Proceedings of the International Conference on Pattern Recognition*, pp. 135–139, 1996.

[30] E. Ardizzone, M. L. Casia, A. Avanzato, and A. Bruna, “Video indexing using mpeg motion compensation vectors,” in *Proceedings of the IEEE International on Multimedia Computing and Systems*, vol. 2, pp. 725–729, July 1999.

[31] R. Lienhart, “Reliable transition detection in videos a survey and practitioner’s guide,” *International Journal of Image and Graphics*, vol. 1, no. 3, pp. 469–486, 2001.

[32] R. Lienhart, “Dynamic video summarization of home video,” in *Proceedings of SPIE Conference on Storage and Retrieval for Media Databases*, vol. 3972, pp. 378–389, January 2000.

[33] Y. Gong and X. Liu, “Generating optimal video summaries,” in *IEEE International Conference on Multimedia and Expo (III), ICME 2000*, vol. 3, pp. 1559–1562, 2000.

[34] B. L. Tseng, C.-Y. Lin, and J. R. Smith, “Video summarization and personalization for pervasive mobile devices,” in *Storage and Retrieval for Media Databases 2002* (M. M. Yeung, C.-S. Li, and R. W. Lienhart, eds.), vol. 4676, pp. 359–370, SPIE, 2002.

[35] I. Otsuka, R. Radhakrishnan, M. Siracusa, A. Divakaran, and H. Mishima, “An enhanced video summarization system using audio features for a personal video recorder,” *International Conference on Consumer Electronics, ICCE ’06, 2006 Digest of Technical Papers*, pp. 297–298, January 2006.

[36] I. Otsuka, R. Radhakrishnan, M. Siracusa, A. Divakaran, and H. Mishima, “An enhanced video summarization system using audio features for a personal video recorder,” *IEEE Transacations on Consumer Electronics*, vol. 52, pp. 168–172, February 2006.

[37] C. Taskiran and E. J. Delp, “Video scene change detection using the generalized sequence trace,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2961–2964, May 1998.

[38] B. Yeo and B. Liu, “Rapid scene analysis on compressed video,” *IEEE Transactions on Circuits and Systems for video Technology*, vol. 5, pp. 533–544, December 1995.

[39] “FFmpeg multimedia system.” `http://ffmpeg.sourceforge.net/`.

[40] “Arm homepage.” `http://www.arm.com/`.

[41] “The familiar project.” `http://familiar.handhelds.com/`.

[42] “Java platform, micro edition.” `http://java.sun.com/javame/index.jsp`.

[43] “Mobile information device profile.” `http://java.sun.com/products/midp/`.