A Study of Low-Complexity Tools for Semantic Classification of Mobile Video

Ashok Mariappan[a], Michael Igarta[a], Cuneyt Taskiran[b], Bhavan Gandhi[b], Edward J. Delp[a]

[a]Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana USA
[b]Multimedia Research Lab, Motorola Labs, Schaumburg, Illinois USA

ABSTRACT

With the proliferation of cameras in handheld devices that allows users to capture still images and videos, providing users with software tools to efficiently manage multimedia content has become essential. In many cases users desire to organize their personal media content using high-level semantic labels. In this paper we will describe low-complexity algorithms that can be used to derive semantic labels, such as "indoor/outdoor," "face/not face," and "motion/not motion" for mobile video sequences. We will also describe a method for summarizing mobile video sequences. We demonstrate the classification performance of the methods and 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 1MP and video sequences of QCIF resolution or less. It is common for users to store images and video sequences in handheld devices in the order of hundreds, if not thousands, and it is a non-trivial issue organizing the data. Personal multimedia content of users on mobile device is currently organized in a non-intuitive way using file name, time or maybe even location, if the device is equipped with a GPS sensor. But, in many cases users desire to cluster the multimedia data in a way that exploits the actual content of the 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 video sequences, 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, and presentation to user via a graphical interface. The first two components will be the focus of this paper.

High-level semantic labels such as "young girl running" and "park scene" characterizes a video 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 truly semantic labels.

In order to achieve such semantic classifications it is not reasonable to expect the user to devote offline computation or time to derive semantic labels for video sequences that are stored in a mobile terminal. Hence the labels have to be derived on the mobile terminal and it has to be a low computational task, since processing power is limited on the mobile terminal. In [1], we examined semantic labels "face/not face," "indoor/ outdoor" that can be used for images. In this paper we describe three low-complexity techniques for pseudo-semantic labeling for mobile 3gpp video sequences based on its contents and describe a set of experiments performed on our test video database. We will also describe the implementation of these techniques on a processor used in many mobile terminals.

This work was supported by a grant from Motorola Labs. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu.

Multimedia on Mobile Devices II, edited by Reiner Creutzburg, Jarmo H. Takala, Chang Wen Chen,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6074, 607408, © 2006 SPIE-IS&T · 0277-786X/06/$15

2. INDOOR/OUTDOOR CLASSIFICATION

2.1. Previous Work

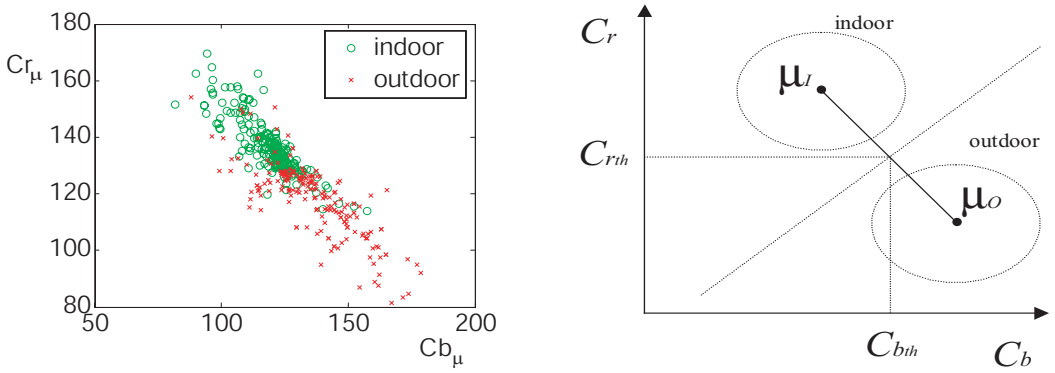
Some of the earliest work in the area of indoor/outdoor scene detection was performed by Picard [2]. In this scheme, color histograms and texture features were used to classify the images. Using 32-bin histograms in the Ohta color space [3] and the nearest neighbor classification rule, a 73.2% image 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%.

Another technique reported in [4] uses a two-stage classifier using two features. Using a support vector machine (SVM) classifier [5], 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. 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.

2.2. Detection Method

In [1], we derived labels "indoor/outdoor" for still images, considering an image with sky as "outdoor." Our "indoor/outdoor" label attempts to detect the presence of blue sky in upper portion of 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 scatterplot of the mean values of C_r and C_b color components for the images in our database is shown in Figure 1A. These color features will form the basis of our "indoor/outdoor" label derivation.

Based on this clustering, the problem of two-dimensional linear classification is reduced to a one-dimensional linear classification problem using a single chrominance component, as shown in Figure 1B.



A. Scatter plot of C_r and C_b color components. **B.** Indoor/Outdoor Classification

Figure 1.

The optimum thresholds that separate the "indoor" and "outdoor" images were obtained using leave-one-out cross-validation. The mean value of C_r component in the top 35% of the image was empirically determined to be the optimal for a training database of 400 still images. To classify a video sequence, frames are extracted from the 3gpp video sequence by examining one frame per second. Each of these frames are processed for "indoor/outdoor" detection. The mean value of the top 35% of each frame, which corresponds to the sky region, is obtained. The mean is compared with 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 "outdoor" video. The schematic diagram for the proposed algorithm is shown in Figure 2.

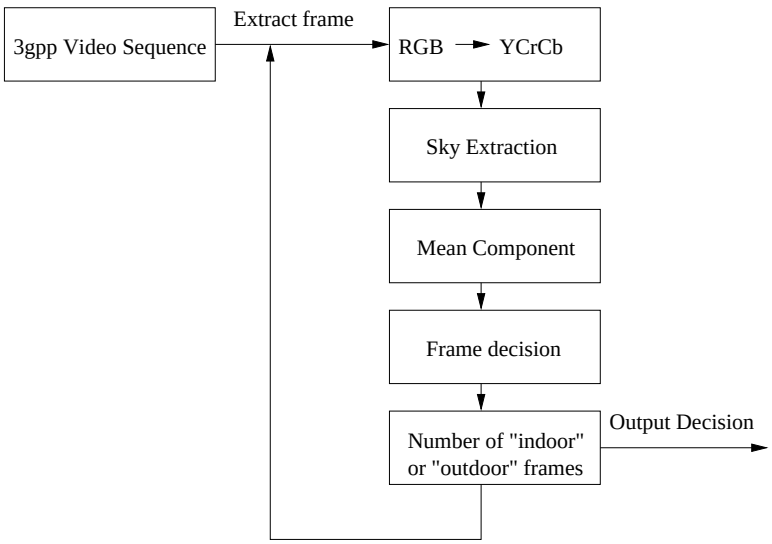


Figure 2. Proposed algorithm for "indoor/outdoor" classification

3. FACE DETECTION

3.1. Previous Work

The work presented in [6, 7] 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 [8]. Skin patches are detected based on color information, and face candidates are generated based on the spatial arrangement of the skin patches in [9]. The face detection work in [10] 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 the C_r and C_b components is illustrated in Figure 3.

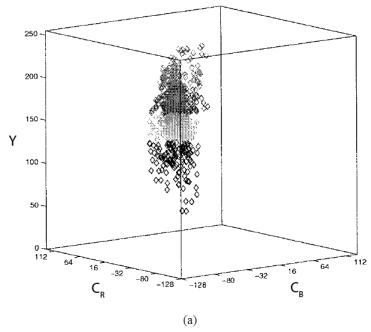


Figure 3. "Skin-like" pixels are highly correlated in the Cr/Cb color space.

3.2. Proposed Method

We use the framework presented in [1], and extend it to a 3gpp video sequence. Similar to "indoor/outdoor" classification, frames are extracted from the 3gpp video sequence at a rate of one frame per second, and each frames is processed for face detection. The final video classification decision is based on the number of frames of "face/not face." If the number of "face" frames are larger than the number of "not face" frames, then video sequence is classified as "face" video. We attempt to detect the presence of full frontal face with a minimum size of 24×24 pixels. The steps involved in classifying an individual frame are given below.

3.2.1. Skin Detection

The first step in our face detection approach is to create a binary mask of the current frame for "skin-like" pixels. Our method for detecting "skin-like" pixels is similar to the method described in [10], 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 frame in RGB color space is converted to the YC_rC_b color space. 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:

$$\begin{aligned} C_r &\geq -2(C_b + 24), \\ C_r &\geq -4(C_b + 32), \\ C_r &\geq -(C_b + 17), \\ C_r &\geq 2.5(C_b + \theta_1), \\ C_r &\geq \theta_3, \\ C_r &\geq 0.5(\theta_4 - C_b), \\ C_r &\leq \frac{220 - C_b}{6}, \\ C_r &\leq \frac{4}{3}(\theta_2 - C_b), \end{aligned} \tag{1}$$

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

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

3.2.2. Block Level Processing

For faster processing, the binary mask after skin detection is sub-sampled into 8×8 blocks. Since we are interested in regions containing "skin-like" pixels with a 3×3 minimum block size, this downscaling 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.

3.2.3. Face Template Matching

Our algorithm attempts to match a "typical face" using a pre-defined template, similar to the method described in [11]. 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 a frame would be surrounded by non-skin pixels, such as hair or background. This

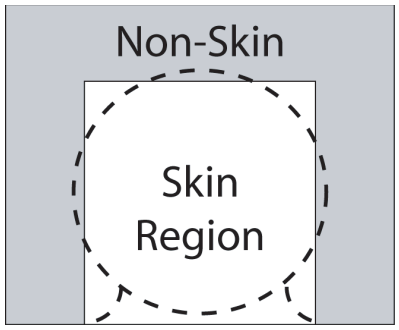


Figure 4. Template used to find faces in individual frames. A face is assumed to be a skin region surrounded by a non-skin region.

is illustrated in Figure 4. 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 [11]. The smallest size face that we attempt to detect is 24×24 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.

3.2.4. Face/Not Face Decision

The final "face/not face" classification decision for the current frame is based on the number of candidate face regions present in the image. This approach is motivated by the observation that "face" frames contained a large number of candidate face regions and "not face" frames contained few candidate face regions. If frame contains a face, many candidate face regions exist depending on the size and position of the candidate face region. The face detection algorithm is outlined in Figure 5 with an example face.

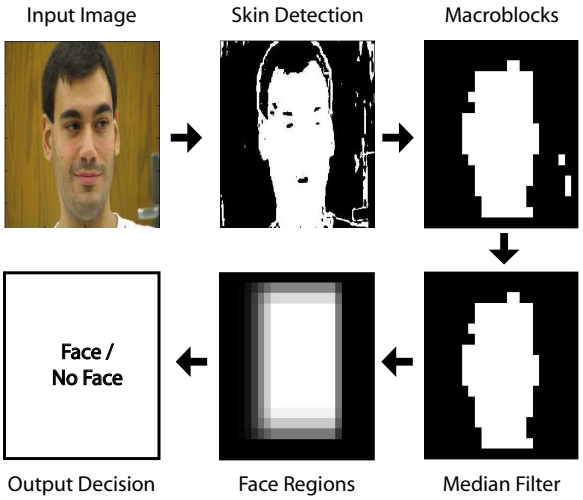


Figure 5. Outline of our proposed face detection method for a single frame.

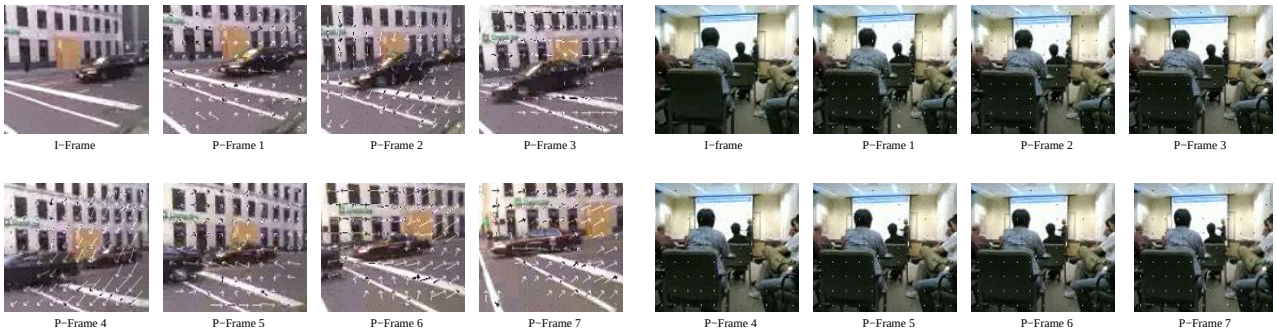
4. MOTION DETECTION

Motion is an essential feature of video data, and it 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 [12] Deng uses motion vector histograms to represent motion, for content based search of video. In [13] Ardizzone splits a video sequence into sequence of shots, and extract representative frames and use the representative frames to derive motion information.

4.1. Motion Detection

Our algorithm uses the motion vectors in a 3gpp video sequence. A 3gpp 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 6A. 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 the motion vector denoted by the arrow. A similar sequence of a typical video sequence with "not motion" label is shown in Figure 6B. Here, the macroblocks are not displaced with respect to the I-frame.



A. "Motion" video **B.** "Not Motion" video

Figure 6. Motion vectors with respect to I-frames for "motion" and "not motion" video sequences

For each video sequence we determine the average macroblock displacement per P-frame. If there are N P-frames in a video sequence, M macroblocks in each P-frame, and d_{ij} represents the motion vector of macroblock j of frame i, with respect to the current I-frame , the average macroblock motion vector D per P-frame is given by,

$$D = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |d_{ij}| \tag{3}$$

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 sequence is classified as "motion," else it is classified as "not motion."

5. VIDEO SUMMARIZATION

Summaries in terms of "key frames" of the video sequence enable users to skim through the video content rapidly without actually viewing the sequence. The main problem with video sequences created using mobile telephones, is that the sequences do not have clear shot boundaries. Lienhart in [14] proposed a method for video summarization. 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. The audio signal in mobile video is of low quality and is reliable. Hence the only available information to summarize a video sequence is the visual content. 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. Our goal with respect to video summarization is to represent a given video sequence with a minimum number of frames.

Histogram analysis and standard deviation based metrics for shot boundary detection have been used extensively for shot boundary detection [15]. We use the generalized trace based on two features: histogram and standard deviation similar to [16]. Given a video sequence, V, composed of N frames. Let $\{f_i\}$, $\vec{x}_i = [x_{1i}x_{2i}]^T$, 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

$$d_i = \|\vec{x}_i - \vec{x}_{i+1}\|_2. \tag{4}$$

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

$$x_{1i} = \frac{1}{2T}\sum_{j=1}^{K}|h_i(j) - h_{i+1}(j)| \tag{5}$$

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:

$$x_{2i} = |\sigma_i - \sigma_{i+1}|, \tag{6}$$

where,

$$\sigma_i^2 = \frac{1}{T-1}\sum_i\sum_j(Y_i(i,j) - \mu)^2. \tag{7}$$

To detect scene changes using a dissimilarity metric, several approaches have been proposed based on sliding window and other techniques [15, 17]. In [16], 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 too 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 $i \in s_j$.

6. EXPERIMENTAL RESULTS

6.1. Video Database

A database of approximately 200 minutes of 3gpp mobile video was created using Motorola A780, Nokia 6630, Nokia 6681 mobile telephones and 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 sequences. The specification of the video sequences in the database is given in Table 1.

Resolution	QCIF 176×144 or less
Frame rate	15 FPS or less
Data rate	192 kbps or less

Table 1. Specification of 3gpp video sequences used

The sequence obtained using the Sony digital handycam were converted to 3gpp format with the above specifications using the FFmpeg Multimedia System [18]. Frames from a few of the sequences are shown in Figure 7.

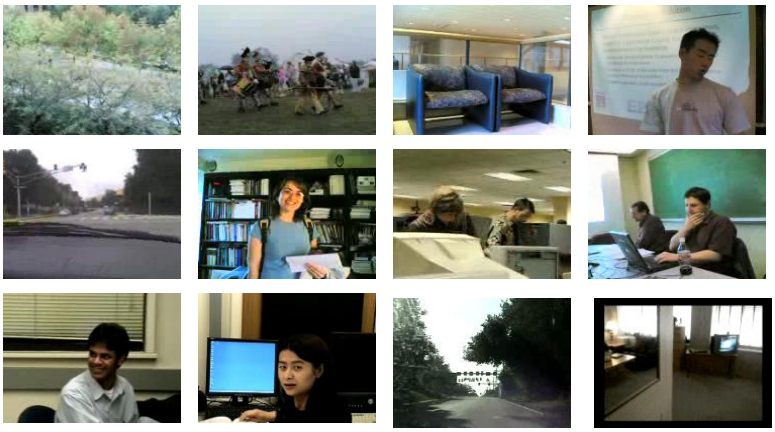


Figure 7. Snapshot from video sequences in database

6.2. Indoor/Outdoor Classification

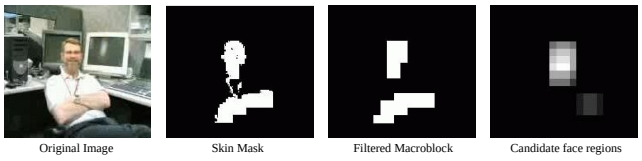
The test database of 324 video sequences was classified as 174 "outdoor" video and 150 "indoor" video. Each video sequence was manually labeled as "indoor" or "outdoor" by a human subject and the label was independently verified by another human subject. The content of the sequences varies widely. The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

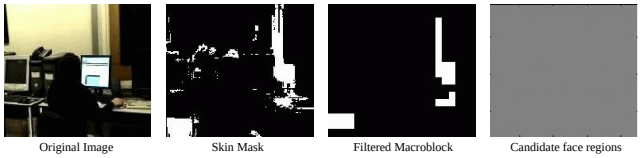
6.3. Face Detection

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 sequence was manually labeled by a human subject and verified by another human subject. For sequences with mixed content, i.e., part of the sequence having "face" frames and part of the 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.

We were able to achieve a classification rate of 71%. A correct "face" classification is shown in Figure 8A. 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 8B. 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.



A. Correct "face" classification



B. Correct "Not face" classification

Figure 8. Face detection examples.

6.4. Motion/Not motion

Only the 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 classification, 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 3, 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.

6.5. Video summarization

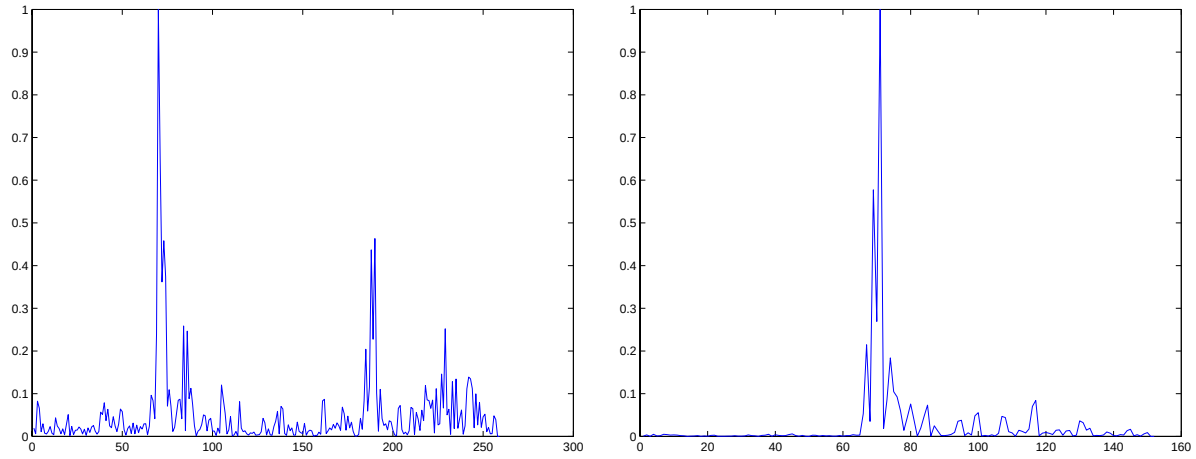
The generalized trace based on the histogram and standard deviation of luminance for two sequences in our database is shown in Figure 9A and 9B.

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 section 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 10. We are currently investigating additional features that can be used to determine the dissimilarity metric.

The classification results for "indoor/outdoor," "face/not face," "motion/not motion" labels are summarized in Table 2.

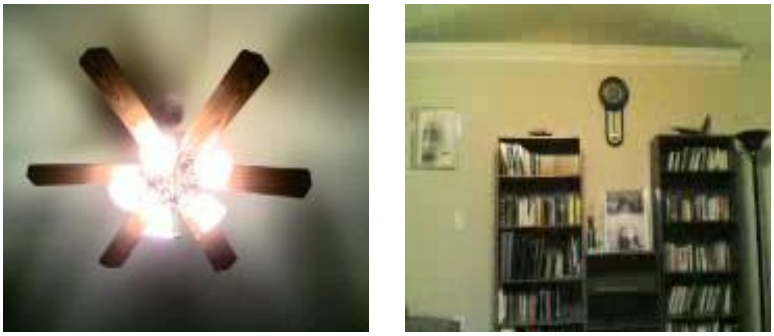
7. TARGET PLATFORM

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



A. Example video1 **B.** Example video2

Figure 9. The generalized trace examples



Frame 37 Frame 89

Figure 10. Representative frames determined for video2

Label	Individual Classification Result (%)	Overall classifcation(%)
Indoor	67	
Outdoor	83	75
Face	69	
Not Face	74	71
Motion	83	
Not Motion	91	87

Table 2. Classification results for "indoor/outdoor", "face/not face", "motion/not motion" labels.

MHz, which is based on the ARM architecture [19], 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 [20] was installed. This is a Linux distribution targeted for the iPAQ series of PDAs. The FFmpeg Multimedia System [18] was used to decode the 3gpp 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 3. These include the time for decoding the 3gpp video, extracting label information and making a classification decision.

Label	Input	Execution Time (s)
Indoor/Outdoor	30 second 3gpp video sequence	5
Face/Not Face	30 second 3gpp video sequence	20
Motion/Not Motion	30 second 3gpp video sequence	7
Video Summarization	30 second 3gpp video sequence	8

Table 3. Execution time on Compaq iPAQ H3970 handheld PDA.

8. CONCLUSIONS

In this paper, we examined three different semantic classification problems: "indoor/outdoor," "face/not face," and "motion/not motion" for 3gpp mobile video sequences. We developed lightweight algorithms to perform the labeling with relatively good performance on a database of approximately 200 minutes of 3gpp video. We presented a simple method for summarizing short video sequences. We are currently refining our methods with respect to classification performance and computational complexity.

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," *Proceedings of the 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, 2005, pp. 111–117.
2. M. Szummer and R. W. Picard, "Indoor-outdoor image classification," *IEEE International Workshop on Content-Based Access of Image and Video Databases*, 1998, pp. 42–51.
3. Y. Ohta, T. Kanade, and T. Takai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222–241, 1980.
4. N. Serrano, A. Savakis, and A. Luo, "A computationally efficient approach to indoor/outdoor scene classification," *Proceedings of the 16th International Conference on Pattern Recognition*, 2002, pp. 146–149.
5. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
6. A. Albiol, C. A. Bouman, and E. J. Delp, "Face detection for pseudo-semantic labeling in video databases," *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, October 1999, pp. 607–611.
7. 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, no. 1, pp. 103–118, February 2004.
8. S. Kawato and J. Ohya, "Automatic skin-color distribution extraction for face detection and tracking," *Proceedings of the 5th IEEE International Conference on Signal Processing*, vol. 2, August 2000, pp. 1415–1418.
9. 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, no. 5, pp. 696–706, May 2002.
10. C. Garcia and G. Tzirtas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 264–277, September 1999.
11. H. Wang and S. F. Chang, "A highly efficient system for automatic face region detection in mpeg video," vol. 7, no. 4, pp. 615–628, August 1997.
12. Y. Deng and B. S. Manjunath, "Content-based search of video using color, texture and motion," *Proceedings of the IEEE International on Conference Image Processing*, vol. 2, 1997, pp. 534–537.
13. E. Ardizzone, M. L. Casia, and D. Molinelli, "Motion and color based video indexing and retrieval," *Proceedings of the International Conference on Pattern Recognition*, 1996, pp. 135–139.
14. R. Lienhart, "Dynamic video summarization of home video," *Proceedings of SPIE Conference on Storage and Retrieval for Media Databases*, vol. 3972, January 2000, pp. 378–389.
15. ——, "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.

16. C. Taskiran and E. J. Delp, "Video scene change detection using the generalized sequence trace," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, May 1998, pp. 2961–2964.
17. B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Transactions on Circuits and Systems for video Technology*, vol. 5, no. 6, pp. 533–544, December 1995.
18. FFmpeg multimedia system. [Online]. Available: http://ffmpeg.sourceforge.net/
19. Arm homepage. [Online]. Available: http://www.arm.com/
20. The familiar project. [Online]. Available: http://familiar.handhelds.com/