REAL-TIME FACE AND HAND DETECTION FOR VIDEOCONFERENCING ON A MOBILE DEVICE

Frank M. Ciaramello and Sheila S. Hemami

Visual Communication Laboratory
School of Electrical and Computer Engineering, Cornell University
Ithaca, NY, 14853
fmc3@cornell.edu, hemami@ece.cornell.edu

ABSTRACT

The increase in processing power on modern mobile devices allows for the implementation of more advanced image and video processing algorithms, such as real-time videoconferencing. In a videoconferencing setting, region of interest encoding techniques can be applied to improve the quality of the user's face. In this work, three face detection techniques are implemented on a mobile device and evaluated in terms of accuracy and speed. A shape-based detection algorithm achieves the fastest detection times of 165 msec, but fails to accurately detect the face in all cases. Local binary patterns and the Viola-Jones algorithm are both capable of accurately detection the face, but are significantly slower. Several methods for increasing the speed of these feature-based approaches are discussed. Finally, the results of the face detection are applied to an H.264 video encoder operating on the mobile device.

1. INTRODUCTION

Videoconferencing on mobile devices is becoming a possibility as cellular network bandwidths are rapidly increasing. Two-way video communication in this setting requires real-time processing on a cellular device. While these devices are more powerful than in the past, they still offer little computational power when compared to modern desktop computers. Slow processors constrain the complexity of the algorithms that can be implemented in real-time on a mobile device. Furthermore, the bandwidths available on a cellular network are significantly smaller than those available on a wired network. Consequently, advanced compression techniques are required to generate video sequences that are useful to the end users.

In traditional videoconferencing, enhancing the quality of the face regions is an effective method of improving the overall perceptual quality of the video [1, 2, 3]. Videoconferencing systems can also be applied to the specific task of transmitting American Sign Language (ASL) video. Such systems allow members of the Deaf community to communicate in their native language. Within this context, the information itself is contained in the signer's facial expressions and hand gestures. Encoding the face and hands with higher fidelity is essential to preserving the information in the sign language conversation [4, 5]. In both of these cases, identifying and encoding only the important portions of the video can result in a significant bit rate savings.

Many algorithms have been proposed to identify faces in images or to identify and track hands in a video sequence (see [6], [7] for surveys). Unfortunately, a large number of these algorithms are not appropriate for low-complexity devices. This work aims to present and analyze low complexity face and hand detection algorithms that can be implemented on a mobile phone. Section 2 describes the algorithms that are implemented on the mobile device. Section 3 compares the detection accuracy and speed of each of the algorithms. Finally, in Section 4, the results of the detection algorithms are combined with an H.264 video encoder in order to encode relevant portions of the video (e.g. the face and hands) with higher fidelity.

2. DETECTION ALGORITHMS

In both videoconferencing and ASL video telephony, encoding only the relevant portions of the sequence at a high quality can yield significant gains in compression. This improved compression is essential for meeting the bandwidth constraints of cellular networks, but requires additional computational complexity for identifying those relevant regions. In this section, the face and hands of an individual are identified through the use of skin segmentation and face detection algorithms. Based on the detected locations of the face and hands, the 16x16 macroblocks in the video are labeled as either face, hand, or background.

2.1. Color and shape based face detection

Face detection can be performed using shape and color information extracted from the image [1]. Skin pixels have a color distribution that is distinct from non-skin pixels [8]. Skin detection is performed in the YUV color space. Because the H.264 encoder also operates within this color space, no color conversion is required to perform the skin detection. The chrominance values (U and V) of skin pixels are modeled as a bivariate Gaussian distribution. The mean μ and covariance matrix Σ of the distribution are generated from a sample set of skin pixels. Skin-color segmentation is implemented by thresholding the Mahalanobis distance, $D_M^2(x)$, between a given pixel's chrominance values x and the skin pixel distribution.

$$D_M^2(x) = (x-\mu)^T \Sigma^{-1} (x-\mu) < \alpha \qquad (1)$$

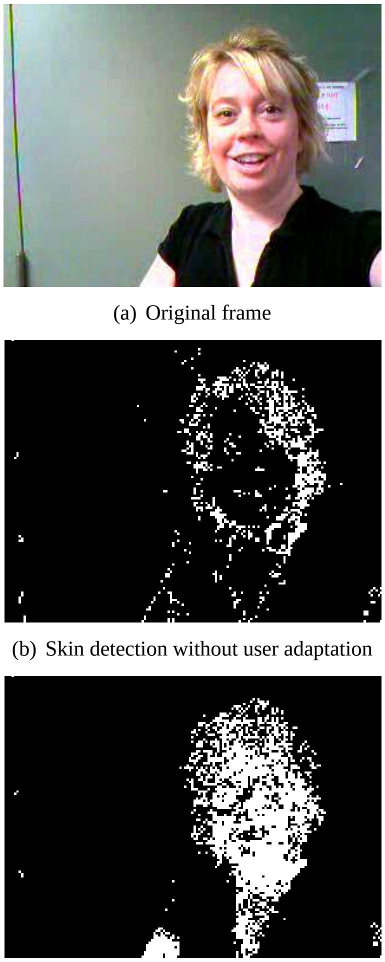
The skin segmentation can be improved by incorporating a user-adaptive skin model. During a video call, the skin color statistics are updated to more accurately model those of the current user. Figure 1 illustrates the improvement in the skin detection by performing this update. In this case, the skin pixels were manually selected and added to the model. In future work, this process can be automated. This update can be done while the call is being connected, by asking the user to hold her hand in a specific location. It can also be done automatically, by first applying face detection then extracting skin pixels, as in [9].

Provided that the skin segmentation is very accurate, a shape-based approach can be used to differentiate between the users face and hands. Given a binary skin map, a connected component analysis is used to identify the size and location of each cluster of skin pixels. Clusters of skin pixels smaller than a fixed threshold are discarded as noise. The remaining skin components are filtered with the morphological erode operator. This shape-based approach erodes the binary skin map using a vertically-oriented elliptical structuring element. Because the human head can be roughly modeled as an ellipse, the face is identified as the largest connected component remaining after the erosion [6].

In the presence of noisy backgrounds or poor lighting conditions, the skin detection can yield a non-trivial amount of false alarms, especially if the background contains skin-colored objects. Because of this, the morphological shape-based face detection fails and feature based techniques are required to identify the face region. Two feature-based detection algorithms are considered: local binary patterns and the Viola-Jones algorithm.

2.2. Local binary patterns

The first approach generates features based on local binary patterns (LBP) of luminance pixels [10]. The LBP is calculated from a neighborhood of L pixels surrounding each



(a) Original frame

(b) Skin detection without user adaptation

(c) Skin detection with user adaptation

Fig. 1. Comparison of skin detection algorithm with and without user adaptation.

pixel by thresholding each neighbor based on the center pixel's value and mapping this to a binary number. For example, using a 3x3 neighborhood ($L = 8$), a pixel whose neighbors are all greater than itself will have a LBP of 11111111. As a consequence of this binary representation, there are only 2^8, or 256, possible binary patterns for an individual pixel.

In order to perform the face detection, a set of LBPs are mapped to an appropriate feature as follows. Given a candidate window, the classification feature is the distribution of all of the local binary patterns in that window, e.g. the 256 bin histogram of possible binary patterns. The classifier is trained on a set of 19x19 face images taken from the FERET database [11]. The average of all the face histograms is used in the classification task. Face detection is performed by searching candidate 19x19 windows in the input image. For each window, the histogram of LBP values is computed and compared against the average face histogram using the Chi square distance, as in Equation 2. H_C and H_T correspond to the candidate and trained histograms, respectively.

$$\chi^2(H_C, H_T) = \sum_{i=1}^{256} \frac{(H_{Ci} - H_{Ti})^2}{H_{Ci} + H_{Ti}} \qquad (2)$$

If $\chi^2(H_C, H_T) < \beta$, the candidate window is identified as a face. In order to identify faces at multiple scales, the classification algorithm is run on downsampled versions of the original image. Overlapping face regions are identified as a single face with a bounding box corresponding to the average of the overlapping regions.

Since each pixel in a window is compared to each of its neighbors, the LBP classifier requires $O(W_W W_H L)$ operations, where W_W and W_H are the width and height of the window and L is the number of neighbors. To search the entire image, the total number of operations is $O(NW_W W_H L)$, where N is the number of candidate windows and is a function of the image size and number of image scales included in the search.

One of the main computational benefits of the LBP-based classifier is that the features themselves can be computed using only fixed-point operations. This is especially important on a mobile device in which floating-point operations must be emulated, which can be prohibitively slow. The most computationally costly part of the LBP-based classifier is the image scaling, since a pyramid of downsampled images must be generated for each scale that is to be searched.

2.3. Viola-Jones classifier cascade

The Viola-Jones face detection algorithm [12] can also be applied to identify the face region. This detection algorithm uses a series of classifier stages. At each stage, simple Haar-like rectangular features are computed in the candidate window. If the window is classified as a face, it continues to the next stage. Each stage is increasingly complex in terms of the number of features, in order to eliminate more non-face windows. Only a candidate window containing a face passes through all the stages in the classifier.

This paper uses the OpenCV implementation of the Viola-Jones algorithm [13], which has been ported for use on the mobile device. The OpenCV package provides a classifier cascade which has been trained for frontal face views. The Viola-Jones classifier has several computational benefits. First, the classifier cascade is organized such that simple classifiers using only a few features can quickly eliminate non-face windows. Second, the use of the integral image representation and simple rectangular features enables the algorithm to detect faces at a range of sizes without rescaling the entire image. The features themselves are scaled to search over larger windows in the image, without having to downsample the original image.

For an individual window, the Viola-Jones classifier requires $O(FS_F)$ operations, where F is the number of features being computed and S_F is the size of the feature (i.e., the number of pixels contained within the rectangular feature). By design, the value of F can vary tremendously. For windows containing a face, the candidate window passes through each stage of the classifier and, in the classifier used here, 2135 features in total are computed. However, a majority of the candidate windows are rejected by the first stage of the classifier, which computes only 3 features. To search the entire image, the total number of operations is $O(NFS_F)$, where N is the number of candidate windows and is a function of the image size and number of image scales included in the search.

The major drawback for implementation on a mobile device is the number of floating point operations. There are 21 classifier stages with between 3 and 200 features per stage. At each stage, the features are computed and compared against a floating point threshold, which results in a very large number of floating point operations, especially for windows which pass through multiple classification stages.

2.4. Hand Detection

While simply identifying the face region may be sufficient for generic videoconferencing, further processing must be done for American Sign Language (ASL) video. In ASL, information is conveyed through both facial expressions and hand gestures. In order to optimally encode ASL videos, the hands must also be identified. Following both skin segmentation and face detection, the signer's hands are identified as the large skin clusters not corresponding to the signer's face.

3. ACCURACY AND COMPUTATIONAL RESULTS

The algorithms described in Section 2 are implemented on an HTC Apache PocketPC with an Intel PXA270 processor running at 416 MHz, with 64 MB RAM, and a 240x320 LCD display. The device runs the Windows Mobile operating system. Three test videos of American Sign Language are used for the evaluation. Two of the videos were recorded using professional video equipment and downsampled to QCIF resolution (176x144) at 10 frames per second. One of these videos was recorded indoor in a studio, the other was recorded outdoors. The third video was captured using the camera on the PocketPC while being held by the signer. It was downsampled from QVGA to a resolution of 160x120 at 15 frames per second.

One of the primary factors controlling the speed of the feature based face detection algorithms are the number of image scales included in the search space. A large number of scales ensures that faces of any size will be found, but each scale adds a significant amount of computation time. The number of scales is limited by controlling the scaling factor and the minimum/maximum expected face size in the image. In this implementation, the scaling factor was set to 1.25, the maximum face size was set to 60% of the image width, and the minimum face size was set to 15% of the image width. Also, at each image scale, the search is performed for every other pixel.

The fastest face detection method is the shape-based approach, which runs at an average of 165 msec per frame. This method is very successful when the skin detection is very accurate. For the indoor scene, the average face detection rate was 93%. However, if the skin detection yields a non-trivial amount of false alarms, the shape-based approach completely breaks down, as is the case in the outdoor scene, as illustrated in Figures 2(a) and 2(d).

The LBP-based classifier achieves an average detection rate of 91%, but has a very large number of false positives, as illustrated in Figures 2(b) and 2(e). Out of 477 frames, the LBP classifier yielded 162 false alarms. The LBP classifier was also the slowest of the three methods, running at an average of 1841 msec per frame. Finally, the Viola-Jones classifier achieves an average detection rate of 90% with only 27 false alarms and runs at 1508 msec per frame.

Of the three face detection techniques, the Viola-Jones classifier achieves the optimal trade-off between positive detections and false alarms. However, in its default implementation, it runs at fewer than 1 frame per second. The search speed can be improved by decreasing the number of image scales (i.e., increasing the scaling factor) or limiting the search space at each scale. The search space can be reduced by only evaluating candidate windows if they contain skin pixels. It can also be reduced by limiting the search to windows which were within one macroblock of a face block in the previous frame. Table 1 demonstrates the speed improvements for each of these cases. At best, the Viola-Jones algorithm runs at approximately 2.8 frames per second.

4. ENCODING PLATFORM

The frame segmentation maps are used by an H.264 video encoding algorithm to achieve increased compression while maintaining the quality in the region-of-interest. In order to capture and encode video sequences in real-time, the x264 video encoder was ported to the mobile phone. x264 is an open-source implementation of H.264 which has been shown to be 50 times faster than the JM reference software with little reduction in performance [14]. As demonstrated in previous work, appropriately applying face and hand segmentation maps to sign language videos results in rate reductions as large as 60%, without sacrificing the overall intelligibility of the video [15]. The mobile phone can encode such videos by executing the face and hand detection algorithms prior to invoking the encoder. Figure 3 presents a frame encoded with this region-of-interest adjustment, using the shape-based detection. The quantization parameter of the face and hand macroblocks is reduced (i.e., the quality is increased) at the expense of the rest of the frame.

5. CONCLUSION

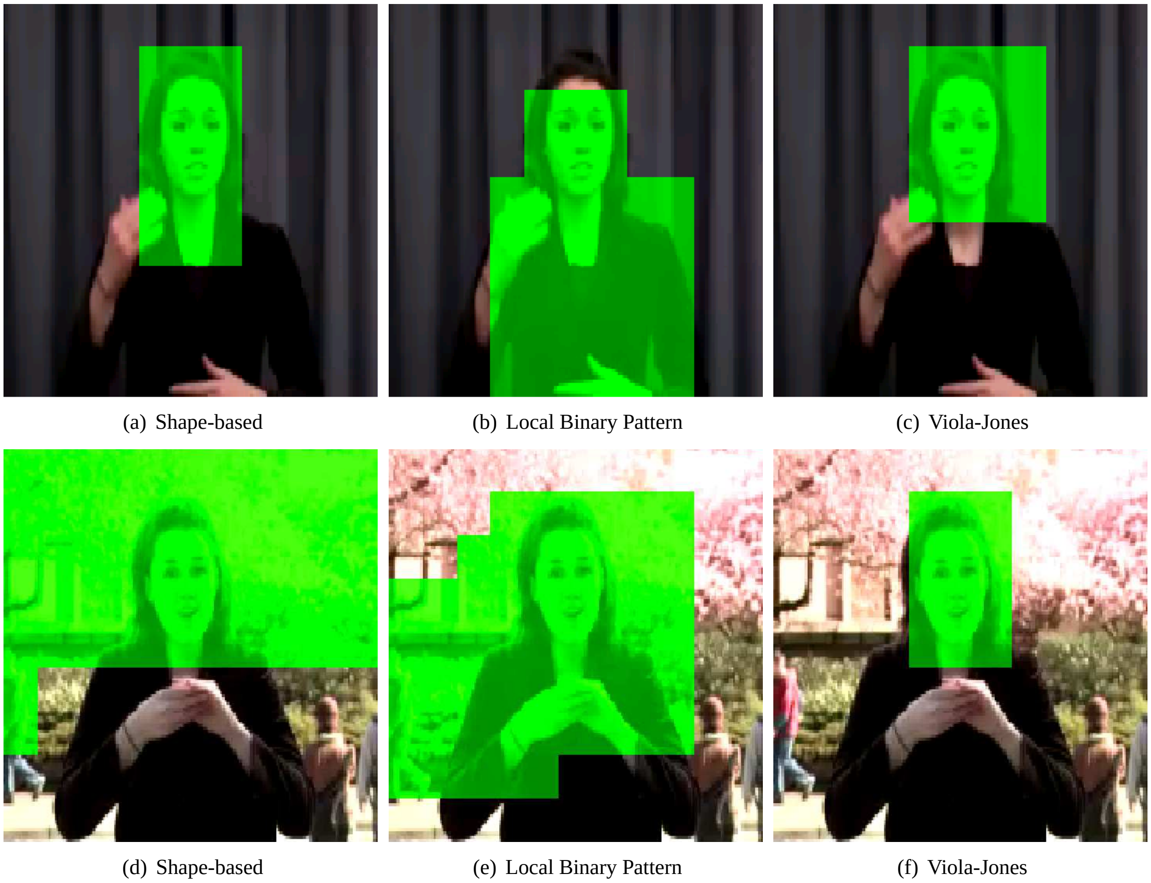
This work analyzes low complexity methods for identifying face and hand regions in a mobile video telephony setting. Shape-based processing is the most computationally efficient method for identifying the face and hands, but cannot adequately identify these regions in the presence of skin-colored backgrounds. In these noisy environments, feature-based face detection techniques are applied to the segmentation task. The Viola-Jones algorithm achieves 90% detection rates with almost no false positives. The feature-based techniques are further optimized by restricting the search space based on the location of skin pixels in the current frame or the face in previous frames. The detection algorithms provide an H.264 encoder with a macroblock-level map of the face and hands, allowing for the use of region-of-interest encoding techniques.

6. ACKNOWLEDGEMENTS

The authors would like to thank Jaehong Chon for his work in porting the x264 encoding software to the PocketPC and in developing the GUI for the MobileASL software.

7. REFERENCES

[1] D. Chai and K. N. Ngan, “Face segmentation using skin color map in videophone applications,” in *IEEE*



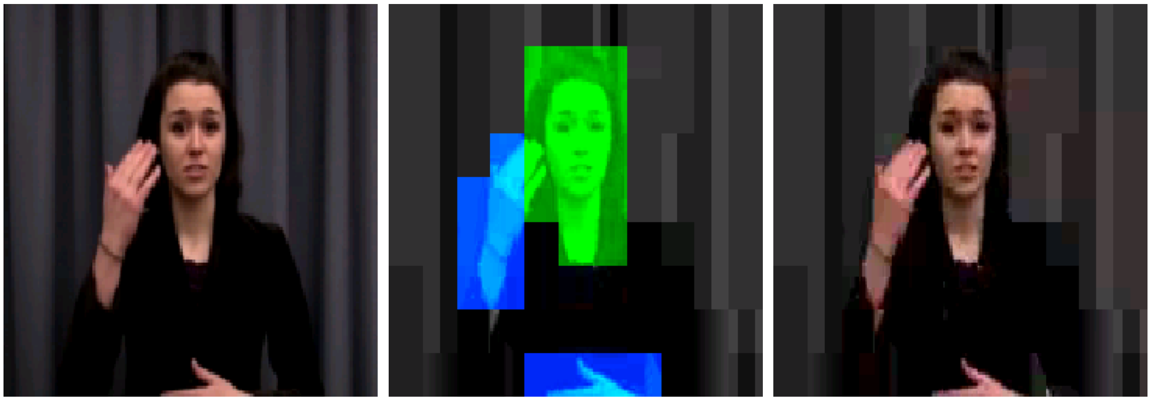
(a) Shape-based (b) Local Binary Pattern (c) Viola-Jones

(d) Shape-based (e) Local Binary Pattern (f) Viola-Jones

Fig. 2. Typical face detection results of the three detection algorithms. Green blocks indicate the macroblock contains part of the face. The shape-based approach works very well on the indoor sequence but fails when the skin detector yields inaccurate results, as in the cherry blossoms in the background. The LBP classifier achieves high detection rates but also has many false positives. The Viola-Jones classifier accurately detects the face with the fewest false positives.

Table 1. Improvements in speed of the Viola-Jones classifier by increasing the image scaling factor and by reducing the search space. The results are presented for the indoor video at QCIF resolution and are consistent for the other videos.

Image Scale	Search Restriction	Positive Detections	False Positives	Average Detection Time
Scale 1.25	No Restriction	87%	0	1257 msec
Scale 1.5	No Restriction	92%	0	806 msec
Scale 2.0	No Restriction	99%	0	423 msec
Scale 1.25	Face in Previous Frame	87%	0	1115 msec
Scale 1.5	Face in Previous Frame	92%	0	671 msec
Scale 2.0	Face in Previous Frame	99%	0	353 msec
Scale 1.25	Skin in Window	94%	0	975 msec
Scale 1.5	Skin in Window	95%	0	636 msec
Scale 2.0	Skin in Window	98%	0	389 msec



(a) Original frame

(b) Face and hand labels

(c) ROI encoded frame

Fig. 3. Illustration of varying region-of-interest quality. Note that the face and hands of the signer are maintained while the background is heavily distorted.

Trans. Circuits and Systems for Video Technology, vol. 9, no. 4, 1999, pp. 551–564.

[2] S. Daly, K. Matthews, and J. Ribas-Corbera, "Face-based visually-optimized image sequence coding," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, 1998, pp. 443–447 vol.3.

[3] C.-W. Lin, Y.-J. Chang, and Y.-C. Chen, "Low-complexity face-assisted video coding," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2, 2000, pp. 207–210 vol.2.

[4] F. Ciaramello and S. Hemami, "'Can you see me now?' An objective metric for predicting intelligibility of compressed American Sign Language video," in *Proc. SPIE Vol. 6492, Human Vision and Electronic Imaging '07*, B. E. Rogowitz, T. N. Pappas, and S. J. Daly, Eds., vol. 6492, 2007.

[5] D. Agrafiotis, N. Canagarajah, D. R. Bull, J. Kyle, H. Seers, and M. Dye, "A perceptually optimised video coding system for sign language communication at low bit rates," in *Signal Processing: Image Communication*, no. 21, 2006, pp. 531–549.

[6] M.-H. Yang, N. Ahuja, and M. Tabb, "Extraction of 2d motion trajectories and its application to hand gesture recognition," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, August 2002, pp. 1061–1074.

[7] S. C. Ong and S. Ranganath, "Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning," vol. 27, no. 6, June 2005, pp. 873–891.

[8] S. Phung, S. Bouzerdoum, A., and S. Chai, D., "Skin segmentation using color pixel classification: analysis and comparison," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 1, pp. 148–154, Jan 2005.

[9] J. Fritsch, S. Lang, A. Kleinehagenbrock, G. Fink, and G. Sagerer, "Improving adaptive skin color segmentation by incorporating results from face detection," in *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, 2002, pp. 337–343.

[10] A. Hadid, M. Pietikainen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," vol. 2, 2004, pp. II–797–II–804 Vol.2.

[11] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," vol. 22, no. 10, 2000, pp. 1090–1104.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," 2001.

[13] "Open source computer vision library." [Online]. Available: http://opencvlibrary.sourceforge.net/

[14] L. Merritt and R. Vanam, "Improved rate control and motion estimation for H.264 encoder," *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5, pp. V –309–V –312, Sept. 16 2007-Oct. 19 2007.

[15] F. Ciaramello and S. Hemami, "Complexity constrained rate-distortion optimization of sign language video using an objective intelligibility metric," vol. 6822, 2008.