ROBUST HUMAN FACE HIDING ENSURING PRIVACY

Isabel Martínez-Ponte, Xavier Desurmont, Jerome Meessen and Jean-François Delaigle

{martinisa, desurmont, jerome.meessen, delaigle}@multitel.be
Multitel A.S.B.L, Avenue Copernic 1, B-7000 Mons, Belgium.

ABSTRACT

Nowadays, video surveillance of people must ensure privacy. In this paper, we propose a seamless solution to that problem by masking faces in video sequences, which keeps people anonymous. The system consists of two modules. First, an analysis module identifies and follows regions of interest (ROI's) where faces are detected. Second, the JPEG 2000 encoding module compresses the frames keeping the ROI's in a separate data layer, so that the correct rendering of human faces can be restricted.

The analysis module combines two complementary methods: face detection, to locate faces in the image, and tracking, to follow them seamlessly along the time. The fusion of these two methods increases robustness: once a face has already been detected in a frame, tracking may locate it in the consecutive frames, even when a face detection algorithm would not. In addition, detection of faces prevents tracking from loosing its targets.

The encoding module downshifts the JPEG 2000 data corresponding to the identified ROI's to the lowest quality layer of the codestream. When the transmission bandwidth is limited, the human faces will then be decoded with a lower visual quality, up to invisibility when required.

The proposed solution has been tested on different types of sequences. The results are presented in the paper.

1. INTRODUCTION

As the number of video surveillance systems increases, ensuring privacy gets more and more importance. The solution we present is based on face masking and consists of two steps: first, an analysis module locates ROI's where faces are detected; second, the JPEG 2000 encoding module isolates these ROI's at the end of the codestream, ensuring poor visual quality when the bandwidth is limited and allowing a restricted decoding of the human faces. A scheme of the system is shown in Figure 1.

Locating faces in images is a complex problem. Difficulty is due to several factors: occlusions, different poses of faces or facial expressions, presence of facial features such as beards or moustaches... An example of face detection used for video surveillance is described in [3]. However, using face detection only is not reliable enough. In order to ensure robustness along a video sequence, our analysis module combines two complementary methods: face detection and tracking.

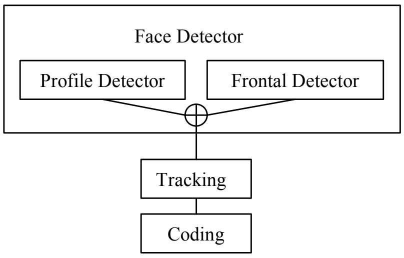

Figure 1. System configuration

JPEG 2000, the new standard for image compression, offers a new flexibility to the coded image [6], [13]. In particular, JPEG 2000 allows differentiating regions of interest from their background, by providing them with higher or lower visual quality. Different approaches have been proposed for coding JPEG 2000 ROI [16], [1], [15]. In this paper, we rely on a codestream organization driven by the ROI's identification. The ROI related codeblocks are isolated in the lowest quality layer, which has two consequences: first, when transmission bandwidth is limited, the ROI will be rendered with poor visual quality and, second, the full decoding of the ROI can be restricted to defined users with this particular right. Compared to the ROI protection used in [3], our system does not need any scrambling or encryption tool. Note that here, the ROI must be hidden rather than coded with higher quality as the in traditional definition of ROI.

The paper is organized as follows: in section 2 we explain the face detection and tracking algorithms used. The encoding strategy is described in section 3. Section 4 concludes our work.

2. ROBUST FACE DETECTION

2.1. Overview

Many techniques have been developed for face detection [20]: knowledge-based methods, based on the relationships between facial features [19], feature invariant approaches, which aim to find structural features [18], template matching methods, which are based on the correlation between an input image and several stored patterns of a face [5], and appearance-based methods, where the models are learned from a set of training images [14].

The detection method we used belongs to the last group. It is based on the OpenCV [12] implementation of the classifier proposed in [17] and improved in [10] and [11]. This classifier is actually a cascade of boosted classifiers working with Haar-like features, i.e. rectangular features which are reminiscents of Haar Basis functions and can be computed in any scale and location in constant time. This kind of classifiers shows very good performances, as we will see later, but is quite specific, that is, a single cascade classifier is overstrained to accommodate all in-class variability. So, classifiers trained for frontal faces can not be used to detect profile faces as the Haar features associated are not the same. Several solutions were proposed in the literature to extend the previous approach to multi-view faces. In [7] a decision tree is trained to determine the viewpoint class. The appropriate detector for that viewpoint is then run. In [9] a detector tree of boosted classifiers is introduced. At each node in the decision tree a clustering-and-splitting step is embedded into the training algorithm to construct branches in the classifier stages recursively. In [8], a FloatBoost learning method with pyramid architecture is used to deal with out-of-plane rotations, in-plane rotations and up-and-down rotations.

2.2. Algorithm evaluation

One profile face detector and four frontal face detectors are available in OpenCV. The frontal face detectors are listed here below:

1. stump-based 24x24 discrete adaboost,
2. stump-based 20x20 gentle adaboost,
3. tree-based 20x20 gentle adaboost,
4. stump-based 20x20 gentle adaboost (a tree of stage classifiers instead of a cascade).

So as to evaluate these detectors, we carried out several tests. A set of 160 images with 540 frontal faces was used. It consists of images obtained from [2] and images extracted from our video surveillance sequences. Two qualitative error rates (detection rate [4] and absolute false alarm rate) are used to compare the detectors.

cascade	detection rate	abs. false alarm rate
1	0.86	108
2	0.87	60
3	0.90	67
4	0.83	31

Table 1 Comparison of OpenCV frontal face detectors

As shown in Table 1, frontal faces were generally well located when the area was not too small and no in-plane rotations occurred. As expected, profile faces were not detected. Moreover, some faces were detected twice and several false alarms occurred.

The performance obtained with the first detector is a little lower than with the other detectors. We can see that, as shown in [10], gentle adaboost outperforms discrete adaboost and that we achieve better results with an input pattern size of 20x20. The results of the last three classifiers show a compromise between detection rate and false alarm rate. We can appreciate that, according to [10], tree slightly outperforms stumps as weak classifiers since stumps do not allow learning dependencies between features.

The profile face detector locates several faces not detected with the previous classifiers as well as some frontal faces. However, some profile faces we expected to be detected were not. This is due to the great variability of the profile faces in terms of in-plane and out-of-plane rotations.

2.2. Proposed Face Detection Approach

In our system (see Figure 1), we propose to use in parallel two of the cascades available in OpenCV, one for frontal faces (cascade 3) and one for profile faces, in order to handle multi-view. Combining a frontal and a profile face detector increases detection rate, but also false detection rate as expected. Tracking will allow us to reject most of the remaining false alarms and to follow the detected faces along the time even if they are not detected in all the frames.

2.4. Tracking

Regarding the tracking algorithm, two different approaches were tested: ‘bottom-up’, where the tracking algorithm tries to match along the time the faces located by the detection algorithm, and ‘top-down’, where tracks are initialised by the detection algorithm and the tracking tries to locate along the time certain features of the detected ROI's.

Results obtained with the bottom-up approach were not very satisfactory because the faces detected were too far in time to be matched, so we decided to follow a top-down approach which is illustrated in Figure 2. Once a face is detected, we calculate its features and estimate its position in the following frame. From the estimation, we search its new position and update the associated features.

2.4. Merging Face Detection and Tracking

The detection algorithm works along in time with the tracking algorithm in order to detect new faces and to gain certitude that the ones detected are real faces.

The same face is usually detected several times, but these detections should not lead to new tracks, they will be considered as new occurrences of the same object. The tracking algorithm accomplishes the fusion if the distance between the features associated is inferior to an established threshold.

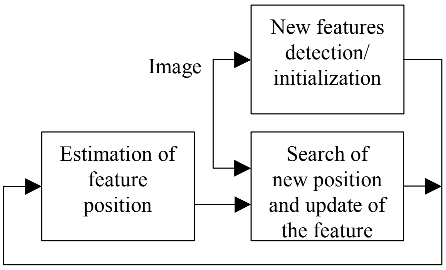

Figure 2. Top-down approach for face tracking

Detected objects whose position did not change for a fixed number of frames were discarded. Indeed, in the applications of interest, people do not remain immobile for a long time. This constraint reduces false detections.

When false detections occur in moving objects, these “false faces” are initially tracked by the algorithm. We have decided to give an initial belief ($b_{t0}=\beta$) to the detected face which decreases with time:

$$(1)\begin{cases} b_{t+1} = \max(b_t+\delta,\gamma_1) \text{ if face is detected}\\ b_{t+1}=\alpha b_t, \alpha<1 \text{ if face is not detected and } b_t<\gamma_1\end{cases}$$

If $b_{t+1}<\gamma_2$ the face is discarded. Figure 3 illustrates the behaviour of the algorithm. Two faces are detected in *t=t0* and *t=t1*; the first one is detected again in *t=t2, t=t3, t=t4, t=t5,* while the second is not redetected and so it is considered to be a false detection. Tracks became darker when their belief is smaller than a third threshold, γ_3, in order to show that the detection may be false. The values of γ_1 and γ_2 were set to 1 and 0.5 respectively and then the other parameters were set experimentally. Figure 4 shows a face being tracked by the analysis module.

The results obtained by combining face detection and tracking are significantly better than the ones obtained using face detection only. On the one hand, tracking locates faces in frames the detection algorithm would not. On the other hand, face detection finds new faces and prevents tracking from loosing its targets. As a consequence, detection rate increases while false detection rate decreases.

3. JPEG 2000 frame encoding

The JPEG 2000 image compression framework allows defining quality layers containing data from pre-defined spatial elements, which are corresponding to a entire number of JPEG 2000 codeblocks in the wavelet domain. So, in the following, we refer to codeblocks as the data chunks associated to spatial regions. Given a target bit rate, a quality layer consists of the codeblocks contributions maximizing the PSNR quality improvement of the decoded image when this layer is added to the preceding layers, see e.g. Figure 5. However, the codeblocks contribution selection can also be achieved based on the definition of regions of interest.

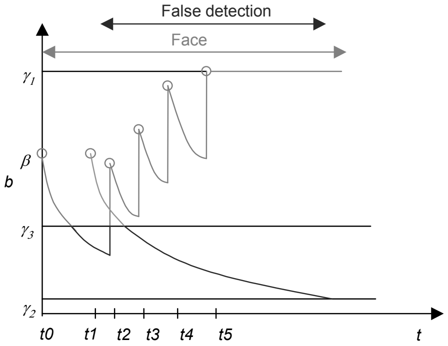

Figure 3. Behaviour of the detection and tracking algorithm when the object detected is a face and when it is a false detection.

Figure 4. Example of a result of the detection and tracking method.

In section 2, the analysis module has identified human faces that must be hidden through encoding. So, we decided to isolate the data from the ROI’s in a separate quality layer as shown on Figure 6. This particular layer is lower than every other quality layer in the coded frame. As a consequence, if this last layer is not decoded by the viewer, the human faces will be invisible. Privacy can then be restricted. Moreover, in case of limited bandwidth, the ROI’s will be rendered with a poorer quality compared to the background. Let’s note that this particular codestream ordering can be achieved online, through transcoding.

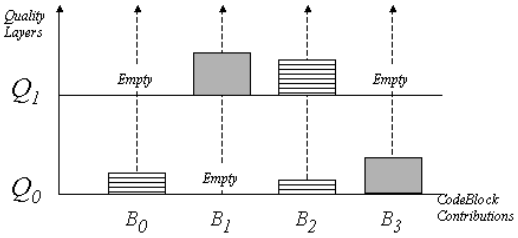

Figure 5. JPEG 2000 Quality layers

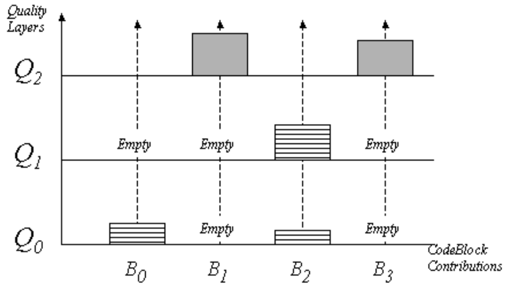

Figure 6. JPEG 2000 quality layers with isolated ROI

4. CONCLUSIONS

We have presented a robust system to mask faces in video surveillance sequences in order to ensure privacy.

First, the analysis module locates faces and follows them seamlessly along the time by the combination of face detection and tracking. These two methods complement each other to increase robustness: once a face has already been detected in a frame, tracking may locate it in frames where a face detection algorithm would not. In addition, the detection method prevents tracking from loosing its targets. The results obtained by the analysis module are satisfactory, especially in sequences were we can distinguish frontal up-right faces as the detection algorithm locates them accurately, which allows a good tracking afterwards.

Second, a JPEG 2000 encoding module isolates the data corresponding to the human faces in the lowest quality layer of the codestream. This ensures poor visual quality for these ROI in lossy compression, up to invisibility if required.

In the near future the analysis module will be improved to detect faces in a wider variety of poses.

9. ACKNOWLEDGEMENT

This work has been partially funded by the European Commission, within the scope of IST WCAM project (IST-507204.)

10. REFERENCES

[1] C. Christopoulos, J. Askelf, M. Larsson, “Efficient method for encoding regions of interest in the upcoming JPEG 2000 still image compression standard”, *IEEE Signal Processing Letters*, pp. 247-249, September 2000.

[2] CMU/VASC image database http://vasc.ri.cmu.edu/idb/html/face/index.html

[3] F. Dufaux, T. Ebrahimi, Emitall SA, “Smart video Surveillance System Preserving Privacy,” to appear in Proc. of SPIE Image and Video Communication and Processing, San Jose, USA, January 2005.

[4] T. Ellis, “Performance Metrics and Methods for Tracking in Surveillance”, *Proc. of the Third IEEE Workshop on PETS*, pp. 26-31, June 2002.

[5] V. Govindaraju, D. B. Sher, R. K. Srihari and S. N. Srihari, “Locating Human Faces in Newspaper Photographs”, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 549-554, 1989.

[6] ISO/IEC 15444-1/ IUT-T T.800, “JPEG2000 Image Coding System - Part 1: Core Coding System”, 2000.

[7] M. Jones and P. Viola, “Fast Multi-view Face Detection”, *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2003.

[8] S. Li, L. Zhu, Z. Zhang and H. Zhang, “FloatBoost Learning and Statistical Face Detection”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, n. 9, September 2004.

[9] R. Lienhart, L. Liang and A. Kuranov, “Detector Tree of Boosted Classifiers for Real-time Object Detection and Tracking”, *IEEE ICME2003,* July 2003.

[10] R. Lienhart, A. Kuranov and V. Pisarevsky, “Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection”, *Proc. of 25th German Pattern Recognition Symposium*, September 2003.

[11] R. Lienhart and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, *IEEE ICIP 2002*, vol. 1, pp. 900-903, September 2002.

[12] Open Computer Vision Library
http:/sourceforge.net/projects/opencvlibrary/

[13] M. Rabbani, R. Joshi, “An overview of the JPEG2000 still image compression standard”, *Signal Processing Image Communication*, Eurasip, vol. 17, n. 1, pp. 3-48, January 2002.

[14] H. Rowley, S. Baluja and T. Kanade, “Neural Network Based Face Detection”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, n. 1, pp. 23-38, January 1998.

[15] V. Sanchez, A. Basu and M. Mandal, “Prioritized Region of Interest Coding in JPEG2000”, *IEEE Trans. on CSVT*, vol. 14 (9), pp. 1149-1155, September 2004.

[16] D. Santa-Cruz, T. Ebrahimi, “An analytical study of JPEG2000 functionalities”, *ICIP2000*, Vancouver, September 2000.

[17] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features*”, Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, 2001.

[18] K. C. Yow and R. Cipolla, “Feature-Based Human Face Detection”, *Image and Vision Computing*, vol. 15, n, 9, pp. 713-735, 1997.

[19] G. Yang and T. S. Huang, “Human Face Detection in Complex Background”, *Pattern Recognition*, vol. 7, pp. 53-63, 1994.

[20] M. Yang, D. Kriegman and N. Ahuja, “Detecting faces in images: A survey”, *IEEE Transactions on Pattern analysis and Machine intelligence*, vol. 24, n. 1, pp. 34-58, January 2002.