Fast and Robust Face Finding via Local Context

Hannes Kruppa[1] Modesto Castrillon Santana[2] Bernt Schiele[1]

[1]Perceptual Computing and Computer Vision Group
ETH Zurich
CH-8092 Zurich, Switzerland
{kruppa, schiele}@inf.ethz.ch

[2]IUSIANI
University of Las Palmas de Gran Canaria
Campus de Tafira, 35017, Gran Canaria, Spain
modesto@dis.ulpgc.es

Abstract

In visual surveillance face detection can be an important cue for initializing tracking algorithms. Recent work in psychophics hints at the importance of the local context of a face for robust detection, such as head contours and torso. This paper describes a detector that actively utilizes the idea of local context. The promise is to gain robustness that goes beyond the capabilities of traditional face detection making it particularly interesting for surveillance. The performance of the proposed detector in terms of accuracy and speed is evaluated on data sets from PETS 2000 and PETS 2003 and compared to the object-centered approach. Particular attention is paid to the role of available image resolution.

1. Introduction and Related Work

Fast and robust target detection from a single image is a desirable capability for tracking and surveillance systems. First, it allows to *verify* the relevance of currently tracked targets and drop undesired hypotheses to make tracking more efficient. Second it allows to *recover* from tracking failure and to reinitialize on targets that have been missed so far. Third, it can effectively *complement* tracking in cases where the target exhibits very little motion or where the target movement is highly discontinuous (e.g. jumps caused by missing video frames). All these examples require that detection be fast and robust.

Often, detecting people in the scene is of particular interest. The classical cues for people detection by means of computer vision are the human face [1, 2], the head [3, 4], the entire body including legs [5] as well as the human skin [6]. Among these face detection in still images is probably the most popular. Recent algorithms use fast-to-compute features and a cascade structure to achieve real-time performance at high levels of accuracy [7]. A detector of this type has been successfully employed for eye localization in the FGNET video sequence (PETS 2003 workshop [8]).

However, it is surprising to see how easily face detectors can be fooled by situations where humans have no problem to reliably detect faces. Such cases have been systematically studied in psychophysical experiments by Sinha and Torralba [9, 10]. One of the common findings is that the human visual system can robustly discriminate real faces from face-like patterns at very low resolutions. Computational systems on the other hand not only require a much larger amount of facial detail for detecting faces in real scenes, but also yield false alarms that are correctly rejected by human observers.

Figure 1: Examples of different faces (inner rectangle), their local context as proposed in this paper (outer rectangle) and their global context (outside the outer rectangle). For illustration purposes only one face per image is examined here.

Torralba's experiments indicate that as the level of detail decreases humans make use of the *local context*, i.e. a local area surrounding the face. This contrasts the assumption behind the predominant object-centered approach that the only image features that are relevant for the detection of an object at one spatial location are the features that potentially

belong to the object and not to the background.

For illustration figure 1 shows some examples of faces within their local and global context. This paper describes a detector that actively utilizes local context as a predictive cue for computational face detection. The promise is to gain robustness that goes beyond the capabilities of traditional object-centered face detectors making it particularly relevant for surveillance.

Section 2 formalizes the idea of local context and analyses the differences in the resulting detectors when trained with or without local context, respectively. A boosted detector cascade is used as the underlying detector. The detection capabilities of the local context detector are then compared with a state-of-the art object-centered approach in section 3. The employed test sets are based on the FGNET video conference data (PETS 2003) and the parking lot sequence of PETS 2000. Section 4 is a concluding summary of this work.

2. Local Context vs. Object-centered Detection

Following the work of [11] a formal derivation for local context is as follows: first, the posterior probability for the presence of object O can be decomposed using Bayes Rule as

$$P(O \mid \mathbf{v}) \simeq P(O \mid \mathbf{v}_L) = \frac{P(\mathbf{v}_L \mid O)}{P(\mathbf{v}_L)} P(O) \qquad (1)$$

where the image measurements $\mathbf{v}$ are in this case local measurements, that is $\mathbf{v} = \mathbf{v_L}$. The object-centered object likelihood is denoted by $P(\mathbf{v}_L \mid O)$ and $P(O)$ is the object specific prior. However, in order to capture dependencies between an object and its context the measurement vector can be extended to include features outside the target object

$$\mathbf{v} = \{\mathbf{v}_L, \mathbf{v}_C\} \qquad (2)$$

where $\mathbf{v}_C$ are measurements of the object's context. Applying Bayes Rule now leads to an expression where all probabilities are conditioned on contextual information

$$P(O \mid \mathbf{v}) = \frac{P(O, \mathbf{v})}{P(\mathbf{v})} = \frac{P(\mathbf{v}_L \mid O, \mathbf{v}_C)}{P(\mathbf{v}_L \mid \mathbf{v}_C)} P(O, \mathbf{v}_C) \qquad (3)$$

To implement the local context approach we train a detector with instances that contain a person's entire head, neck and part of the upper body. Intuitively this choice promises to contain important cues for the presence of faces. The resulting training data is quite different from object-centered approaches where only the faces themselves are considered. Figure 2 shows training examples of both paradigms for comparison.

During detection the actual face location is inferred by assuming a fixed position within the detection window. The size and location of the face are directly computed from the width and height of the detected local context. This is illustrated in figure 3.

Figure 2: Examples of training instances used in the object-centered approach (top row) versus the proposed approach based on local context (bottom row).

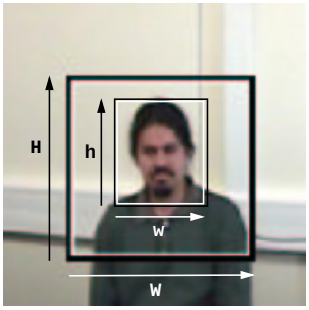

Figure 3: Whenever the local context of a face is detected the actual face location is inferred by assuming a fixed position within the detection frame. Here $w = W/2$, $h = H/2$ and the offset relative to the upper left corner is set to $(W/4, H/10)$.

The employed detector framework is a modified version of the Viola-Jones detector and available through the Open Computer Vision Library [12]. Details about the underlying algorithms the learning approach and the parameters are given in a separate subsection (section 2.1).

Basically, the features of this detector are weighted differences of integrals over rectangular subregions. Figure 4 visualizes the set of available feature types (figure taken from [13]) where black and white rectangles corresponds to positive and negative weights, respectively. The feature types consist of four different edge features, eight line features and two center-surround features.

The learning algorithm (which is in a way reminiscent of decision-tree learning) automatically selects the most discriminant features considering all possible feature types, sizes and locations. It is interesting to compare the selected features in case of the object-centered approach and

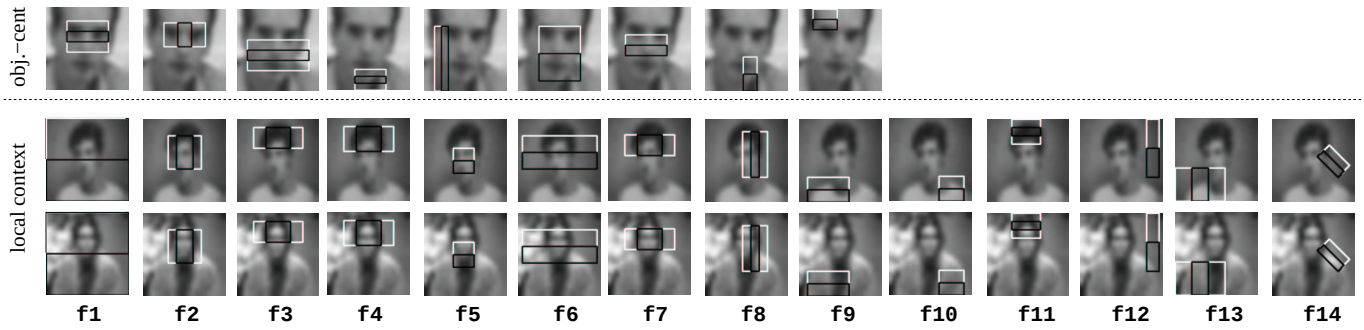

Figure 5: Automatically selected features (first classifier stage) for the object-centered face detector (9 features overlaid on a random training instance, top row) and the local context detector (14 features overlayed on two different training instances), bottom two rows – the number of features is automatically determined by the learning algorithm). In the local context case the first feature (f1) extends over the entire patch, thus making use of information that is not available in the object-centered case. In addition, features f9 and f10 as well as feature f14 capture the left and right shoulder to help in the face detection task.

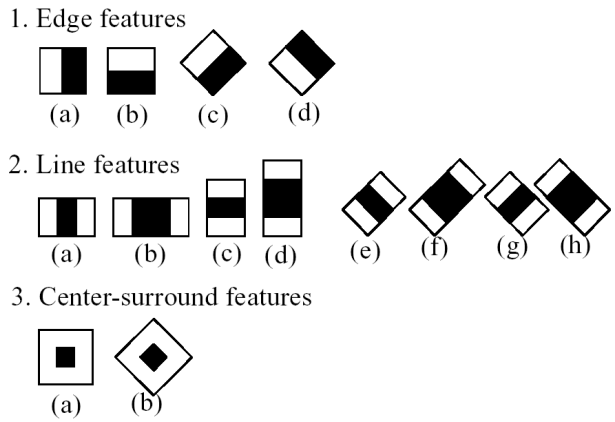

Figure 4: Extended integral feature set according to [13] including rotated features. These features are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses.

the local context approach. Figure 5 is a visualization of the selected rectangle features (top row: object-centered case, bottom two rows: local context case). Two different training instances (individuals) are shown in the local context case for illustration purposes. The features displayed here are evaluated in the first stage of the detector cascade and can therefore be regarded as the most important features. There are 9 features in the object-centered case and 14 features in the local context case (the learning algorithm determines the number of required features per stage automatically).

In the object-centered case the first three features capture inner-face regions, in particular around the eyes and around the mouth and nose by using horizontal and vertical line features. Additional features are mostly edge features to capture the contours of the head and chin (f5, f8 and f9 in the figure).

Contrastingly, for the local context case the very first feature extends over the entire patch, i.e. it actually makes use of the local context. The following features capture head and body contours (features f2-f5). Other features capture the left and right shoulder (features f9 and f10 in the upper example, feature f14 in the bottom example). Hence, this is quite different from traditional face detectors, which rely on facial parts alone.

2.1. Learning Approach and Implementation Details

The employed detector framework is based on the idea of a boosted classifier cascade (see [7]) but extends the original feature set and offers different boosting variants for learning [13]. This subsection summarizes the most essential implementation details regarding features, learning algorithm and training parameters.

The feature types as depicted in figure 4 are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses. Their main advantage is that they can be computed in constant time at any scale. Each feature is computed by summing up pixels within smaller rectangles

$$\text{feature}_\text{I} = \sum_{i \in I = \{1,\ldots,N\}} \omega_i * RecSum(r_i) \qquad (4)$$

with weights $\omega_i \in \mathbb{R}$, rectangles r_i and their number N. Only weighted combinations of pixel sums of two rectangles are considered, that is, $N = 2$. The weights have opposite signs (indicated as black and white in the figure), and

are used to compensate between differences in area. Efficient computation is achieved by using *summed area tables*. Rotated features and center-surround features have been added to the original feature set of Viola-Jones by Lienhart et al [13] using rotated summed area tables. The original set consists only of features (1a), (1b), (2a) and (2c) as well as one diagonal feature which is subsumed by the rotated features. The augmented feature set has been shown to extend the expressiveness and versatility of the original features leading to more accurate detectors. Note again that this feature representation does not require to compute an image pyramid to search over different scales.

The cascade learning algorithm is similar to decision-tree learning. Essentially, a classifier cascade can be seen as a degenerated decision tree. For each stage in the cascade a separate subclassifier is trained to detect almost all target objects while rejecting a certain fraction of the non-object patterns. The resulting detection rate and false positive rate of the cascade is then given by

$$F = \prod_{i=1}^{K} f_i \qquad\qquad D = \prod_{i=1}^{K} d_i \tag{5}$$

For example, if a 20 stage detector is trained such that at each stage 50% of the non-object patterns are eliminated (target false positive rate) while falsely eliminating only 0.1% of the object patterns (target detection rate) then the expected overall detection rate is $0.999^{20} \approx 0.98$ with a false positive rate of $0.5^{20} \approx 0.9 * 10^{-6}$. Ultimately, the desired number of stages, the target false positive rate and the target detection rate per stage allow to trade-off accuracy and speed of the resulting classifier. This also explains the different numbers of features in the first stage for the object-centered detector and for the local context detector as shown in figure 5.

Individual stages are trained using boosting which combines a set of "weak learners" into a powerful committee ("strong classifier"). In this case a weak learner is equivalent to one specific feature and a (automatically learned) binary threshold on its value. Each round of boosting selects the weak learner (i.e. feature type and threshold) that best classifies the weighted training set. The first boosting round assumes a uniform weighting of the training data while successive stages assign higher weights to misclassified training instances. This lets the algorithm focus on the "hard" cases in successive rounds. The different boosting variants Discrete, Real and Gentle AdaBoost offered by the OpenCV framework mainly differ in how they determine the new weights of the training data. For details the reader is referred to [14]. It has been empirically shown in [12] that the Gentle Adaboost variant outperforms Discrete and Real Adaboost for face detection tasks both in accuracy and speed. Thus Gentle Adaboost (GAB) has been adopted.

Parameter	Loc.Context	Obj-centered
Positive examples	960	5000
Negative examples	1232	3000
Stages	21	25
Min hit rate	0.995000	N/A
Max false alarm rate	0.500000	N/A
Width	20	24
Height	20	24

Table 1: Comparison of training parameters for the local context detector and the object-centered detector. An optimized and pre-built detector of Lienhart et al was used here for which not all parameters have been reported (N/A in the table).

Finally, we summarize the most important training parameters such as the type and number of training instances, the target rates and the detection window size. About 1000 local context examples were gathered from the world wide web and from private photo collections. In order to limit the amount of variation and hence increase discriminance only frontal views have been used for training and instances have been roughly aligned. Each positive example is scaled to 20×20 pixels.

For gathering negative examples a subset of the WuArchive[1] image database has been used that did not contain any people. These images are repeatedly scanned by the learning algorithm to search for object-like patterns. These specific "border-cases" allow to refine the decision boundary (this method is sometimes referred to as "bootstrap"). Training the local context detector took around 48 hours on a 1GHz Pentium III machine. Table 1 compares the training parameters of the local context detector to the object-centered detector which comes with the OpenCV Library and which is used in the experiments in the following section.

3. Performance Evaluation

To understand the relevance of local context several experiments have been carried out on PETS video sequences, namely the FGNET video conference data from PETS 2003 and the parking lot sequence from PETS 2000. Both data sets are disjoint from the data used for training.

3.1. Indoor Sequence

¿From the FGNET video conference data every 100th frame from sequences A, B, and D (cameras 1 and 2) is used in the following experiments[2]. This results in a total of 502 frames containing 1160 faces, 160 of which are profiles

[1] http://wuarchive.wustl.edu/

[2] a similar subset was used in [8]. However, in this paper we have included frames 21900-22300 as well

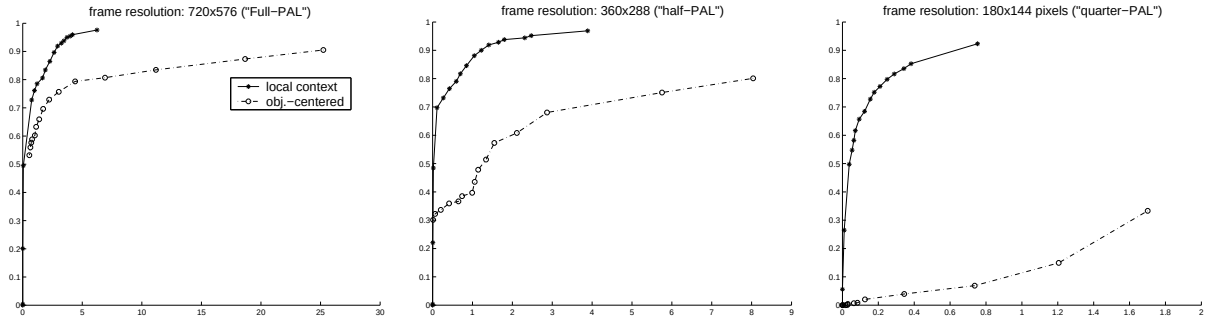

Figure 7: Detection accuracy on the FGNET data set. Each plot shows the percentage of detected faces (vertical) vs. the number of false positives per frame (horizontal). The ROC curves show the performance of the object-centered detector and the local context detector. Note that the frame resolution is decreased from the left plot to the right plot. At the original resolution shown on the right side, the local context detector dominates already because it is more robust to face pose changes. At lower frame resolutions (middle plot and right side plot) facial details deteriorate and the object-centered approach becomes unreliable. The local context on the other hand is not affected. Since the local context detector can operate robustly at very low frame resolutions it actually runs 15 times faster than the traditional object-centered approach at the same level of accuracy.

Figure 6: FGNET sequence, face pose changes: The left plot shows detections of the local context of faces while the right plot shows the output of the object-centered detector. The latter misses two faces because it is restricted to frontal view detection. While an additional specialized side-view detector could be trained (assuming the object-centered paradigm) this would at least require to gather a large amount of additional training data of profile views. Contrastingly, the local context approach does not require such specialized training and is robust to face pose changes.

(about 14%). Each frame has a resolution of 720×576 pixels and faces are about 48×64 pixels large. The sequences show a conference room across either side of a desk with people occasionally entering and leaving the room.

The left plot in figure 7 shows the face detection performance on the FGNET data set in terms of the ROC curve. The percentage of retrieved faces is given on the vertical axis and the number of false positives per frame is shown on the horizontal axis. Points on the curve are sampled by varying the detection threshold b of the final stage in the detector cascade:

$$H = sign\left[\sum_{i=1}^{K} h_i + b\right] \tag{6}$$

where H is the output of the final classifier, and h_i denotes the individual stages. The cascade is successively cropped in order to yield additional detections. Both the performance of the object-centered and the local context detector are shown. For the object-centered version the face detector by Lienhart et al. has been used[3]. The detector has been shown to yield excellent performance comparable to the state-of-the-art [12]. As can be seen in the left plot of figure 7 at 5 false alarms the object-centered detector retrieves 80% of the faces and the curve flattens from thereon. Contrastingly, the local context detector yields 95% of the faces at the same level of false alarms.

This can be explained by profile views of faces contained in the data which are not detected by the object-centered detector used here. The local context, however, is not affected by face pose changes and can thus detect both frontal and side-views. An example frame containing side-views is shown in figure 6 also showing the outputs of both detectors. In the object-centered approach one would have to separately collect profile instances and train a specialized detector in order to achieve a comparable performance level. However, it has also been found in [15, 16] that profile detection generally tends to be more error-prone because certain discriminant sub-patterns (such as the eye-to-eye symmetry in frontals) are missing. The local context detector

[3]this face detector is now part of the Open Computer Vision Library, www.sourceforge.net

overcomes these difficulties successfully.

Additional experiments were performed to examine the influence of available image resolution. To this end each frame is downsampled from the original resolution (720×576 which approximately corresponds to PAL resolution) to 360×288 ("half-PAL") and to 180×144 pixels ("quarter-PAL"), respectively. Accordingly, the available face resolution decreases from 48×64 to 24×32 and to 12×16 pixels approximately. Figure 8 shows an example frame where

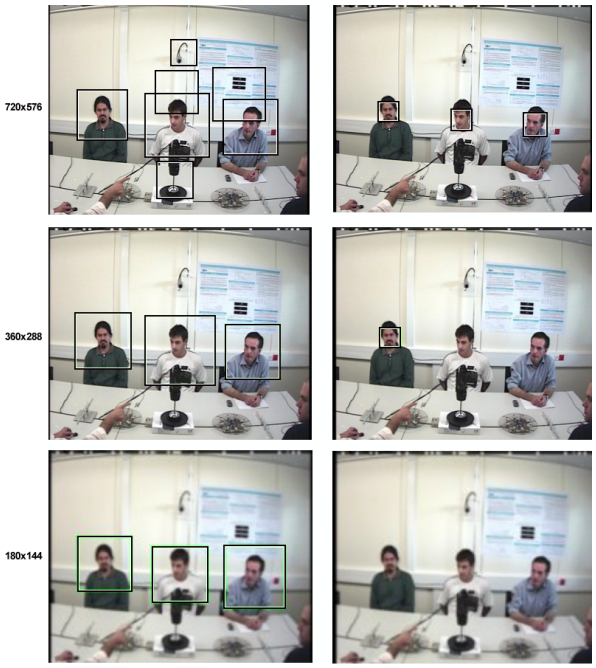

Figure 8: FGNET sequence, resolution changes: Each row of this figure corresponds to a different frame resolution (frame resolution decreases from the top to the bottom row). The images have been rescaled to the same size only for illustration purposes. The left column shows detections based on local context, the right column is from the object-centered approach. This example illustrates that as facial details degrade the object-centered approach misses actual faces. The local context cue is much less affected by resolution changes. It consistently retrieves all three faces at all tested resolutions, while the object-centered approach does so only for the highest frame resolution.

each row corresponds to a different resolution (all frames have then been resized for visualization purposes). The left column shows detections based on local context, the right column is from the object-centered approach. At half-PAL resolution (middle row) the object-centered detector apparently becomes less tolerant to variations in facial appearance, in this case caused by slight pose changes of the middle and right person. As a result it fails to detect these faces at lower resolutions. The local context detector on the other hand does not rely on facial details alone and can still locate all faces successfully. The situation aggravates for the object-centered approach as resolution is further decreased. At quarter-PAL resolution it does not return any detection in this example while the local context approach again detects all faces.

A quantitative account of this experiment is given by the plots in the middle and to the right of figure 7 showing the ROC curves of both detectors when applied to the downsampled data sets. At half-PAL resolution (middle plot) the object-centered detector yields about 70% of the faces at 5 false alarms which is a 15% drop compared to the full resolution. It is directly affected by the decrease in available facial detail. Contrastingly, the local context detector's performance remains stable at 95% given the same number of false alarms. This effect becomes even stronger at quarter-PAL resolution. The corresponding ROC is shown to the right in the same figure. In the quarter-PAL case the object centered approach detects only 10% at 1 false alarm per frame while the local context detector succeeds for more than 90% of the faces. Overall the local context detector provides improvements in detection rates by 15%, 25% and 80% at corresponding levels of false alarms.

The possibility to robustly operate at low resolutions can provide a significant speed-up in face search. For illustration, consider the case where we want to obtain 80% of the faces with less than 5 false positives per frame. Using the face detector this is only possible at the highest frame resolution (full-PAL), where processing of a single frame takes 1.2 seconds. Contrastingly, the local context detector achieves this accuracy already at the quarter-PAL resolution within 0.08 seconds per frame. This corresponds to a 15-fold speedup.

It must be emphasized here that the local context detector was not systematically optimized (e.g. by testing different training parameters) and thus the reported results could probably be further improved.

3.2. Outdoor Sequence

For investigating their suitability to surveillance both detectors were applied to the parking lot test sequence of PETS 2000. This video sequence shows a parking lot from a single static camera looking from above. The video shows cars and people entering and exiting the scene. Every 10th frame was used which results in 146 frames containing 210 faces. Each frame has a resolution of 756×576 pixels. Faces are as small as 12×14 pixels.

Figure 9 shows an example frame of the sequence with detections of the local context detector. The situation is much more difficult than in the video conference setting: the background is more complex which inevitably leads to more false alarms. Moreover, the perspective is difficult

Figure 9: Parking Lot Sequence: Inferring the presence of faces through local context. In this outdoor sequence at least 1'000'000 candidate subregions per frame had to be correctly rejected. The situation is much more demanding than the FGNET scenarios because of the perspective (downward looking camera) and the background clutter.

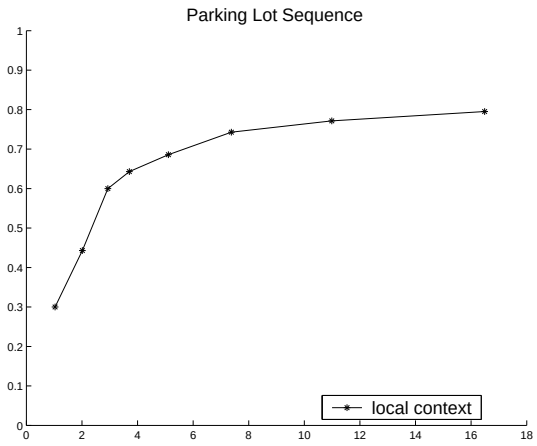

Figure 10: Parking Lot Sequence: The ROC curve shows the performance of the local context detector. The object-centered approach fails because faces are too small in this sequence, so its ROC is not shown. The local context detector retrieves about 75% of the actual faces at 7 false alarms per frame. This clearly indicates that the local context detector goes beyond the capabilities of object-centered face detectors.

as the camera is looking downward and people occasionally turn their upper bodies to the side. The maximum total area covered by faces is only about 0.1% per frame compared to about 2% in the indoor sequence. This means that at least 1'000'000 candidate subregions per frame have to be correctly rejected to yield an acceptable false alarm rate. Note that the legs of people in this sequence are sometimes occluded, for instance when they walk in between parking cars. This makes the scenario difficult for pedestrian detection approaches such as [5]. However, their upper bodies are permanently visible which effectively allows local context detection.

The ROC curve for the local context detector is shown in figure 10. The object-centered approach fails completely on this sequence: it would require about twice the available image resolution to detect any faces. Hence only the ROC of the local context detector is visible here. As can be seen, it retrieves about 75% of the actual faces at 7 false alarms per frame. This already shows that the local context detector goes beyond the capabilities of object-centered face detectors. Moreover, in a surveillance application one could further reduce the number of false alarms for example by using background subtraction ([8]).

4. Summary and Conclusions

This paper evaluated the performance of a local context detector as a means of finding human faces. The detector was implemented using a framework which is part of the Open Computer Vision Library and as such freely available. The approach was evaluated on two different PETS sequences: one indoor sequence (video conference) and one outdoor sequence (parking lot). It has been shown by ROC analysis that the local context cue outperforms the employed face detector. This is mainly due to its greater robustness to pose changes, to variation in individual appearance as well as to low image resolutions. Robust operation at low resolutions not only speeds up the search process but is also of particular interest for surveillance where close up shots of people are often not available. The analysis of the outdoor sequence shows that the local context detector goes beyond the capabilities of object-centered face detectors and correctly infers the locations of human faces as small as 9×12 pixels.

Acknowledgments

The second author is supported by Beleyma and Unelco through Fundación Canaria Universitaria de Las Palmas, and by research projects PI2000/042 of Gobierno de Canarias and UNI2002/16 of Universidad de Las Palmas de Gran Canaria.

References

[1] Erik Hjelmas and Boon Kee Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, no. 3, 2001.

[2] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Detecting faces in images: A survey," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[3] Marco La Cascia and Stan Sclaroff, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, April 2000.

[4] Stan Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proc. IEEE Conf. on CVPR*, 1998.

[5] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proceedings of the International Conference on Computer Vision*, 1998.

[6] Michael J. Jones and James M. Rehg, "Statistical color models with application to skin detection," Technical Report Series CRL 98/11, Cambridge Research Laboratory, December 1998.

[7] Paul Viola and Michael J. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, 2001.

[8] David Cristinacce and Tim Cootes, "A comparison of two real-time face detection systems," in *Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*. March 31 2003, IEEE.

[9] Pawan Sinha, "Qualitative representations for recognition," in *Biologically Motivated Computer Vision (BMCV)*, H.H. Buelthoff et al., Ed., 2002, pp. 249–262.

[10] Pawan Sinha Antonio Torralba, "Detecting faces in impoverished images," in *AI Memo 2001-028, CBCL Memo 208*, 2001.

[11] Antonio Torralba, "Contextual modulation of target saliency," in *Advances in Neural Information Processing Systems*, 2001.

[12] Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," in *DAGM'03*, Magdeburg, Germany, September 2003.

[13] Rainer Lienhart, Luhong Liang, and Alexander Kuranov, "An extended set of haar-like features for rapid object detection," Technical report, Intel Research, June 2002.

[14] Y. Freund and R.E. Schapire, "Experiments with a new boosting algorithm.," in *Proceedings of the 13th International Conference on Machine Learning*, 2001, pp. 148–156.

[15] Henry Schneiderman and Takeo Kanade, "A statistical method for 3d object detection applied to faces and cars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[16] S.Z. Li et al, "Statistical learning of multi-view face detection," in *ECCV*, 2002.