A Tracking Framework for Accurate Face Localization

Ines Cherif, Vassilios Solachidis and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki 54124, Greece.
Tel: +30-2310996304
{ines,vasilis,pitas}@aiia.csd.auth.gr

Abstract. This paper proposes a complete framework for accurate face localization on video frames. Detection and forward tracking are first combined according to predefined rules to get a first set of face candidates. Backward tracking is then applied to provide another set of possible localizations. Finally a dynamic programming algorithm is used to select the candidates that minimize a specific cost function. This method was designed to handle different scale, pose and lighting conditions. The experiments show that it improves the face detection rate compared to a frame-based detector and provides a higher precision than a forward information-based tracker.

1 Introduction

Achieving a good localization of faces on video frames is of high importance for an application such as video indexing and thus, multiple approaches were proposed to increase the face detection rate. In this paper, we introduce a new method making full use of the information provided by a backward tracking process and merging the latter with the detection and forward tracking results using a Dynamic Programming (DP) algorithm. Detection and forward tracking were associated in several research works to improve the detection rate [1]. Combining forward and backward tracking, on the other hand is a rather new idea. It is suitable for analyzing movie or prerecorded content, since in such cases, we have access to the entire video. An extension to particle filtering is described in [2]. In this probabilistic framework, the preliminary detected faces are propagated by sequential forward tracking. A backward propagation is then performed to refine the previous results. As for Dynamic Programming techniques, they are widely used to tackle various issues, among them motion estimation [3], feature extraction and object segmentation [4]. They were also used to perform the face detection and tracking, searching for the best matching region for a given face template [5]. In [6], a multiple object tracking is presented, where the Viterbi Algorithm is used to find the best path between candidates selected according to skin color criteria.
In this paper, a new deterministic approach is presented. It applies face detection, forward tracking and backward tracking, using some predefined rules.

From all the possible extracted candidates, a Dynamic Programming algorithm selects those that minimize a cost function.
The paper is organized as follows: Section 2 presents the new framework for the extraction and labelling of the candidates for the face localizations. Section 3 describes how the trellis structure is applied to select the trajectory with the lowest cost. Section 4 provides the results obtained on several video sequences and section 5 concludes the paper.

2 Tracking Framework

In order to achieve a high detection rate on each frame of a video sequence, detection and tracking algorithms were combined and some rules were defined to form a complete tracking framework.

2.1 Detection

The implemented face detector is based on Haar-like features [7]. The algorithm provides good detection results in case the orientation of the face is almost frontal. But it also produces some false alarms. Therefore, a postprocessing step is added for rejecting detected faces, if the number of skin-like pixels present in the detected bounding box is below a threshold. The region of the image containing the detected face is converted into the HSV color space and two morphological operations, erosion and dilation are performed, in order to remove the sparse pixels. The detection bounding box is then replaced by the smallest bounding box containing all the skin-like pixels. This operation helps removing a part of the background and thus better defining the tracked region. The skin-like pixels are identified as those that fulfill the three following conditions:

$$0 < h < 0.1 \tag{1}$$

$$0.23 < s < 0.68 \tag{2}$$

$$0.27 < v \tag{3}$$

where h, s and v are the coordinates of the HSV color space. This approach is similar to the one used in [8].
The detection process is applied on the first and last frame of a shot and every five frames within the shot. This detection frequency appears to provide satisfactory results. Ideally, if a person is once correctly located in each shot, then the forthcoming processes will provide the missing localizations in the other frames.

2.2 Forward Tracking

To be able to localize faces on every video frame, a forward tracking process is performed on each frame, starting from frames where faces have been detected. The tracking algorithm used is the one described in [9], based on the so-called morphological elastic graph matching (EGM) algorithm. It is initialized by the output of the face detection algorithm and the faces can then be tracked until the next detection of the same face or until the end of the shot, if the faces are not detected again.
In fact, one face can be detected several times in a shot, this can lead to multiple tracking of a same actor, which is time consuming. To overcome this problem, a tracking rule is used in order to identify if newly detected faces correspond to previously tracked faces. This rule is based on the percentage of overlap P_{over} between the detected bounding boxes (D_i) and the ones resulting from the forward tracking (F) in the same frame. We define P_{over} as follows:

$$P_{over}(F) = \max_i \frac{A_{(F \bigcap D_i)}}{\min(A_{D_i}, A_F)} \tag{4}$$

where A_{D_i} is the area of the i^{th} detection bounding box and A_F is the area of the forward tracking bounding box. As for $A_{(F \bigcap D_i)}$, it corresponds to the area recovered by both bounding boxes. If P_{over} is higher than 70%, the two bounding boxes correspond to the same actor and the new detection is used to re-initialize the tracker.
This rule is illustrated on Fig 1. On the first frame of the shot, D_1 represents a detected face and is associated to a first actor. The forward tracking of the detected face is performed until the next detection frame and the bounding boxes are assigned the same label (Actor 1). On the next detection frame, D_2 and D_3 are compared to the tracking bounding box on the same frame. The face that fulfills the overlap condition (D_3) is assigned the same label (Actor 1) while the other (D_2) is associated to a new actor (Actor 2). This rule is applied to the other detections D_4 and D_5 as well.

2.3 Backward Tracking

In order to provide a new set of face candidates, a backward tracking process is performed on each frame. The tracker is initialized by the face detection results as shown in Fig 1. This backward process is very useful in case a face is not detected at the beginning but in the middle of a shot. The forward tracking provides the bounding box localizations from the detection frame to the end of the shot. As for the backward tracking, it will provide the missing results from the first frame of the shot to the frame where the last face detection has been performed.
A more interesting contribution of the backward tracking is obtained when the forward tracking or the detection process fails to accurately locate the face of

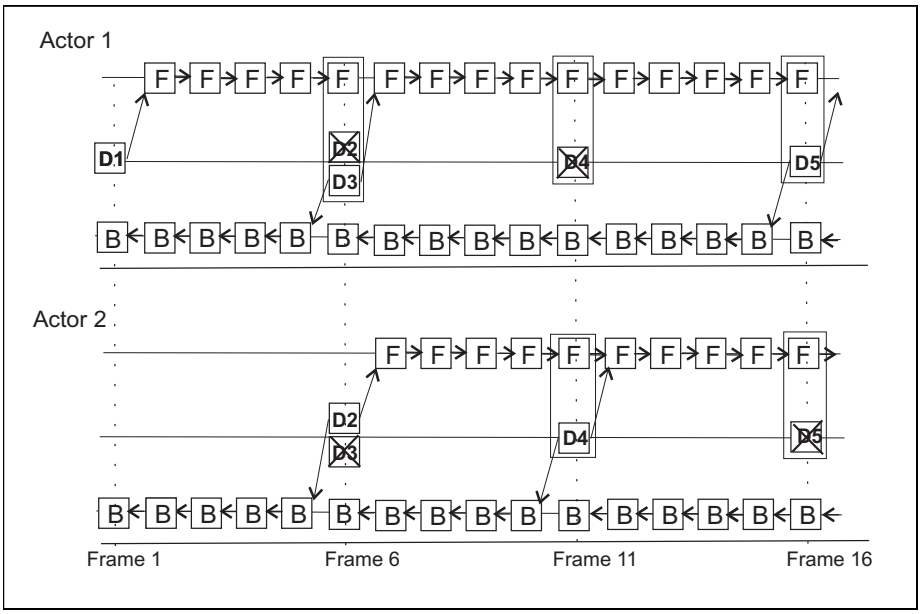

Fig. 1. Illustration of the tracking rule. (D): Detection bounding boxes, (F): Forward tracking bounding boxes and (B): Backward tracking bounding boxes

an actor on a frame i, due for instance to an occlusion, bad illumination or if the tracker sticks to the background. If the next detection of this same actor on the frame ($i + 5n$, $n \in \mathrm{N}^*$) is more precise, then this information will be propagated back and might generate, on i, a new face candidate with a higher accuracy.
Proceeding this way, we will get one, two or three candidates per frame for the face localization, corresponding to respectively the face detection, forward tracking and backward tracking results.

3 A trellis structure for optimal face detection

Now in order to improve face localization, Dynamic Programming is used as a postprocessing. In Section 2, each bounding box was assigned a label. Therefore a trellis can be defined for each actor as represented in Fig 2. The labels D, F and B define the states of the trellis diagram. The frames, where face detection took place can have states D, F and B, while the other frames can have states F and B only.
The complexity of the trellis is considerably reduced in comparison with other approaches that draw the trellis using all the bounding boxes provided by the detector or the tracker [6]. In fact, the number of possible paths in the trellis grows exponentially with the number of nodes. Therefore, limiting the number

of candidates to three is a major advantage of this method.

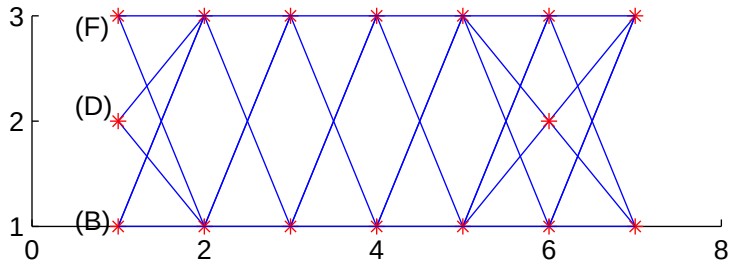

Fig. 2. Model of trellis with 7 frames ($N = 7$). (D): Detection results,(F): Forward tracking results and (B): Backward tracking results

3.1 Cost

Finding the optimal face detection/tracking is equivalent to a best path extraction from a trellis. For each frame of the video sequence we have one, two or three states representing the face candidates provided by the face detection/tracking framework. The cost of a path until the frame l can be expressed as follows:

$$C(l) = -\sum_{i=1}^{l} C(s_i) - \sum_{i=2}^{l} C(s_{i-1}, s_i) \tag{5}$$

For each edge connecting a state s_{i-1}(corresponding to a bounding box B_{i-1} in the previous frame) to another state s_i(corresponding to a bounding box B_i in the current frame) we define the transition cost $C(s_{i-1}, s_i)$ as a combination of two metrics $C_1(s_{i-1}, s_i)$ and $C_2(s_{i-1}, s_i)$:

1. The first cost C_1 takes into account the overlap between the bounding boxes referenced B_i and B_{i-1}.

$$\mathcal{O}(B_{i-1}, B_i) = \frac{A_{(B_{i-1} \bigcap B_i)}}{\min(A_{B_{i-1}}, A_{B_i})} \tag{6}$$

where A_{B_i} is the area of the bounding box B_i. $A_{(B_{i-1} \bigcap B_i)}$ represents the area of the intersection of the bounding boxes B_i and B_{i-1}. We will assume that the bounding boxes of two consecutive frames must have a non-zero overlap. C_1 will take a $-\infty$ value in order to forbid the transition between non-overlapping bounding boxes.

$$C_1(s_{i-1}, s_i) = \begin{cases} \mathcal{O}(B_i, B_{i+1}), \text{ if } \mathcal{O}(B_i, B_{i+1}) > 0 \\ -\infty, \text{ otherwise} \end{cases} \tag{7}$$

Practically, a very small negative value will suffice.

2. The cost C_2 is equal to the ratio between the areas of the bounding boxes as specified by Eq.8. This metric penalizes big changes of the bounding box area during tracking.

$$C_2(s_{i-1}, s_i) = \frac{min(A_{B_{i-1}}, A_{B_i})}{max(A_{B_{i-1}}, A_{B_i})} \tag{8}$$

The transition cost $C(s_{i-1}, s_i)$ is then deduced from $C_1(s_{i-1}, s_i)$ and $C_2(s_{i-1}, s_i)$ e.g. by simple multiplication.
To obtain now the node cost $C(s_i)$, we compute the distance between the center of the bounding box (x_{c_i}, y_{c_i}) and the centroid $(\overline{x}, \overline{y})$ of the skin-like pixels.

$$C(s_i) = \exp\left(-\frac{\sqrt{(\overline{x} - x_{c_i})^2 + (\overline{y} - y_{c_i})^2}}{\sqrt{H^2 + W^2}}\right) \tag{9}$$

with H and W being the height and width of the frame.
The position of the centroid is defined as follows:

$$\overline{x} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} jA(i,j) \tag{10}$$

$$\overline{y} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} iA(i,j) \tag{11}$$

where A is an $n \times m$ matrix, whose elements take the value 1 when the corresponding pixel in the bounding box B_i is skin-like and 0 otherwise.

Once both node and transition costs are defined, the optimal path will be extracted as follows. For each node on the frame l, the accumulate cost $C(l)$ from the first frame to l is calculated using the accumulate cost $C(l-1)$ to the different states in the frame $l-1$. The lowest cost provides the shortest path to the current node and the sequence of nodes leading to this cost are memorized. This process is iterated until the last frame. The shortest path is then retrieved by backtracking the path to the first frame. An example of optimal path is presented on Fig 3 for 30 video frames.

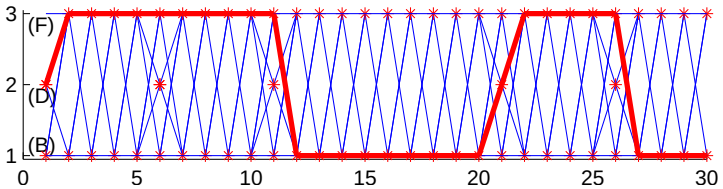

Fig. 3. Shortest path extracted from a 30-frame trellis.

4 Experiments and results

4.1 Metrics for performance evaluation

Three metrics are used to evaluate the performance of the algorithm described above:

- Detection Rate (DR)

$$DR = \frac{N_{GD}}{N_{GT}} \tag{12}$$

  where N_{GD} is the number of good detections within the set of detected bounding boxes. N_{GT} is the number of ground-truth bounding boxes. A detected bounding box is considered as good detection if $\frac{A_{(GT \bigcap D)}}{A_{GT}} > 0.3$, where $A_{(GT \bigcap D_i)}$ is the overlapping area between the ground-truth bounding box and the detected bounding box associated to it.
- False Alarm rate (FA)

$$FA = \frac{N_{FA}}{N_D} \tag{13}$$

  where N_{FA} refers to the number of false alarms within the set of detected bounding boxes. N_D is the number of bounding boxes detected. A bounding box is counted as false alarm if $\frac{A_{(GT \bigcap D)}}{A_{GT}} < 0.3$.
- Overlap precision measure (P)

$$P = \frac{1}{N_{GD}} \sum_{i=1}^{N_{GD}} \frac{A_{(GT \bigcap D_i)}}{\sqrt{A_{GT} A_{D_i}}} \tag{14}$$

  This metric evaluates the overlap between the ground-truth and the correctly detected bounding boxes [10]. This measure not only favors the bounding boxes presenting a high overlap with the ground-truth bounding boxes, but also penalizes those that contain a lot of non-ground-truth pixels

4.2 Results

Ground-truth has been generated manually for a series of video sequences in order to evaluate the performance of the algorithm.
The metrics were calculated for three sets of results. The first set (A) corresponds to the detections performed on each frame, the second set (B) contains the results of the detection (with a five-frame period) combined with a forward tracking process, while the third set (C) represents the detection, forward and backward results merged by the proposed algorithm as shown in the previous sections. The results obtained on three video sequences are presented in Table1.

In the three cases, we notice that the Detection Rate (DR) increases when forward tracking is used. In fact, the face detector fails to determine the position

	Sequence 1			Sequence 2			Sequence 3		
	setA	setB	setC	setA	setB	setC	setA	setB	setC
Detection Rate(DR)	0.6923	1	1	0.7345	1	1	0.6281	0.9587	1
False Alarm (FA)	0	0	0	0	0	0	0.4685	0.5105	0.5105
Precision (P)	0.7911	0.7183	0.7595	0.7971	0.7985	0.8044	0.8262	0.8242	0.8515

Table 1. Performance results.

of some faces due to the pose or the poor illumination. The missed faces can be recovered by the forward tracking process. The Detection Rate (DR) also further increases when both forward and backward tracking have been used, since the face was not detected at the beginning of the shot but after several frames. For each of these frames, the trellis contained only one candidate resulting from backward tracking.
Once candidates were provided by the detector, forward and backward trackers, the trellis performed a selection that always improved the overlap precision (P), i.e. the face localization on the video frame.
We can also notice that one drawback of the tracking approach is that when a face is erroneously detected, then it is tracked on the whole shot thus increasing the False Alarm rate (FA), as can be seen in the case of the sequence 3.

5 Conclusion

In this paper, we proposed a forward/backward tracking process providing an accurate face localization in digital videos. It can also be applied for tracking any object for which we process an object detector. The described process combines detection, forward and backward tracking algorithms in order to extract possible faces. These candidates are used as nodes in a trellis diagram. The extraction of the optimal path from this trellis provided us the optimal choice of the facial bounding boxes. Our approach was mainly oriented towards face localization improvement and we noticed in fact that the precision rate was increased, while realizing a good detection rate. In our future work we will go further into exploiting the trellis structure and work towards decreasing the false alarm rate by merging distinctive trajectories.

6 Acknowledgement

The work presented was developed within NM2 (New Media for a New Millenium), a European Integrated Project (http://www.ist-nm2.org), funded under the European Commission IST FP6 programme.

References

1. R.C. Verma, C. Schmid, and K. Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE Transactions PAMI*, vol. 25, no. 10, pp. 1215–1228, Oct. 2003.
2. Ji Tao and Yap-Peng Tan, "Accurate face localization in videos using effective information propagation," in *Proc. of the IEEE International Conference on Image Processing (ICIP 2005)*, Genoa, September 2005.
3. M. Gong, "Motion estimation using dynamic programming with selective path search," in *International Conference on Pattern Recognition*, Cambridge, United Kingdom, august 2004, vol. 4, pp. 203–206.
4. Changming Sun and Ben Appleton, "Multiple paths extraction in images using a constrained expanded trellis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1923–1933, 2005.
5. Z. Liu and Y. Wang, "Face detection and tracking in video using dynamic programming," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2000)*, 2000, pp. 53–56.
6. F. Pitié, S-A. Berrani, R. Dahyot, and A. Kokaram, "Off-line multiple object tracking using candidate selection and the viterbi algorithm," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2005)*, Genoa, September 2005.
7. Paul Viola and Michael J. Jones, "Robust real-time face detection," in *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001.
8. N. Nikolaidis G. Stamou, M. Krinidis and I. Pitas, "A monocular system for automatic face detection and tracking," in *Proc. of Visual Communications and Image Processing (VCIP 2005)*, Beijing, China, July 2005.
9. N. Nikolaidis G. Stamou and I. Pitas, "Object tracking based on morphological elastic graph matching," in *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2005)*, Genova, Italy, September 2005.
10. S. Huovinen B. Martinkauppi, M. Soriano and M. Laaksonen, "Face video database," in *Proc. First European Conference on Color in Graphics, Imaging and Vision (CGIV 2002)*, Poitiers, France, April 2002.