FACE DETECTION AND TRACKING FOR ACTIVITY RECOGNITION

Ana Kuzmanic Skelin, Mirjana Bonkovic, Vladan Papic
Laboratory for Biomechanics and Automatic Control Systems
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
R. Boškovića b.b., 21 000 Split, Croatia
E-mail: akuzmani@fesb.hr

ABSTRACT
In this paper we present an approach to human activity recognition in an indoor environments. It combines Adaboost face detection with standard tracking methods – continuously adaptive mean-shift (CAMShift) and Kalman filtering. The output results are used to describe simple events of human activities in video using Finite State Machine. The system has been tested on several video sequences and we provide a summary of results.

KEY WORDS
Face detection, face tracking, activity recognition

1. Introduction
Human activity recognition research in computer vision evolved from the significant progress of detection and tracking approaches. Various approaches have been proposed in different scenarios, majority of which belong to surveillance type of applications (e.g. security application, meeting annotation, driver monitoring applications), which in general rely on human body motion analysis. Motion and related activities are considered at different levels of details: the analysis of whole body, body part, single subject motion and activity, and group interactions. Considering the activities in an indoor environments are occluded by surrounding objects, the visibility of human body is limited. Also, considering the wide field of view recordings provides low resolution, it is not possible to detect whole body nor fine information such as eye gaze. In such cases, a face as the most prominent part of human body is considered to be most suitable feature for localizing people in images. In this work, we presume that the face and the motion of the head provide sufficient information to describe activities and finally to interpret behavior.

Detection is well covered research area in literature. Face detection can be very accurate when face is in an upright position and the images are of high resolution. However, in an intended application faces can take up multiple views and can appear as a very small target. Among different techniques proposed for face detection, the Viola Jones detector [1] has the best results for low resolution images, uncontrolled illumination changes and small faces in arbitrary backgrounds.

Tracking is large research field as well. Kalman filters [2] and particle filters [3] are used for tracking when dynamic are modeled as linear and non-linear, respectively. Region tracking techniques, like mean-shift [4] and its continuously adaptive version [5], are suitable when there is enough information in color appearance. Lucas-Kanade tracker [6] is often applied for low-level feature tracking. For detailed overview of the three major fields: detection, tracking and activity recognition, the interested reader is referred to [7],[8],[9]. In the following sections a system that recognizes human activities based on combination of detection and tracking of faces will be described. Our algorithm uses basic face detectors, CAMShifts and Kalman implementations in OpenCV [10] are used, as they are publicly available.

2. Tracking faces using detections

2.1 Single-frame face detection

An algorithm proposed by Viola and Jones has been utilised to detect faces. This algorithm exploits the local contrast configurations driven by an Adaboost classifier to detect possible regions with human face. VJ detector provides scale invariant detections without the use of temporal information. Besides, detections can oscillate with small per-frame variations and can be inconsistent: a face can be detected in one frame, disappear in the next frame and reappear again. Although this detector is fast and robust to illumination conditions, it hardly works when face pose variation excides certain views. Figure 1. illustrates the head pose limits still detectable by V-J detector. Detector pose limits are defined at the ends of following angle intervals

\[ \phi = [-15^\circ, +15^\circ], \theta = [-10^\circ, +10^\circ] \text{ and } \varphi = [-20^\circ, +20^\circ] \]  

(\( \phi, \theta, \varphi \)– roll, pitch, yaw).

To ensure continuous face localization, to localize faces undetectable by the VJ detector, to maintain identity and eliminate false positives, we introduce a combination of CAMShift tracker and Kalman filtering. The algorithm flowchart is given in Figure 2.
Similar approaches that complement detection with trackers can be found in [11], [12].

Figure 1. Face detection using V-J detector

Figure 2. Overview of the proposed tracking algorithm

\section{2.2 Face tracking}

\textbf{CAMShift:} CAMShift algorithm is a non-parametric density gradient estimator. This adaptable version of mean-shift algorithm uses color histogram based target representation as a model. As the color distributions of the target changes over time, the algorithm has to be modified to adapt dynamically to the probability distribution it is tracking. Like in every region based tracking, the open question is the means and frequency of model update [13]. The model update aims to avoid track drifting when the appearance of face varies due to changes in pose and illumination. Usually, the CAMShift is updated at every frame. This makes it robust to detailed appearance changes caused by head pose variations. CAMShift tracks well even with the large displacement as long as the object appearance changes gradually between the consecutive frames. In reality, the appearance of the face can change considerably with time, or might get occluded. Due to CAMShift’s gradient ascend nature, it has a problem with occlusions making the tracker trapped by local maximum. To cope with this problem, target model is updated at every detection. In this work, CAMShift operates in such a way as to follow the detections whenever they are available or uses the last available detection either to initialize tracker or to update a target model. Additionally, target model is updated through Kalman filtering results in such a way as to stabilize the search region.

\textbf{Kalman filtering:} The Kalman filter algorithm belongs to the state-space approach class of tracking algorithms. It solves the tracking problem based on the state-space equation and the measurement equation. We used first-order linear dynamic model for position and zeroth order model for scale. Let \( \{ x_t, v_t, s_t \} \) be the position, displacement and scale for tracked region in the current frame, such that

\[
x_{t+1} = x_t + v_t + N(0; Q_x)
\]

\[
v_{t+1} = v_t + N(0; Q_v)
\]

\[
s_{t+1} = s_t + N(0; Q_s)
\]

We assume that \( \tilde{x}_t \) and \( \tilde{s}_t \) are the observations of \( x_t \) and \( s_t \) with additive Gaussian noises of covariance \( R_x \) and \( R_s \).

At time \( t \) we have estimate \( \{ \hat{x}_{t|t-1}, \hat{s}_{t|t-1} \} \) based on observations at previous instant \( t-1 \). The search for face is directed around the position \( \hat{x}_{t|t-1} \) and scale \( \hat{s}_{t|t-1} \).

Without Kalman filters region and scale estimates and VJ face detector update, the CAMShift fails within few iteration because the background is learned into the color distribution model of face. The face in our example moves across a region of background with which it shares a significant colour. Figure 3 shows the localization of face using proposed algorithm. Right portion of Fig. 3.(a) and Fig. 3.(b) shows the face region used to update model.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{Examples of face localization}
\end{figure}

\section{2.3 Initialization, association and termination}

Whenever the detected face in the current frame reaches a reliable threshold, indicating the times a new face is detected, and cannot be matched to any of the active tracks, it is used to start a new track. If there is no sufficient number of successive detections, it is considered to be a false detection and the track is discarded. Whenever a detected face is matched to any of the active tracks, it is used to update a tracking model.
Matching is done according to the spatial overlapping score $O$ defined by the ratio of the intersection and union of two regions $R_1$ and $R_2$:

$$O = \frac{(R_1 \cap R_2)}{(R_1 \cup R_2)} > 0.6$$

Matching can be done according to some of distance measures, as well. However, the overlap suffices for this scenario.

Termination is checked through the target history, i.e. age and life counters of accumulated detections $N_{detect}$ and tracking steps without target model updating $N_{track}$. The track is terminated when $N_{track}/N_{detect} > \alpha$ ($\alpha$ case dependent factor; we set $\alpha$ to 5) or no new detections have been associated with the track for $F$ frames. Also, the constraint on region size can terminate the track.

3. Activity recognition

During the tracking process the history of face tracks has been collected. Considering each activity produces specific head motion, a Finite State Machine (FSM) has been used to obtain the semantic description from analyzed sequences of movements. The states of the FSM are represented by nodes and transitions between two states are represented by edges. For each possible transition in the FSM there is a corresponding output symbol produced. Through the course of time a state to state path can be traced. Similar model has been used in [14],[15]. The input features to FSM consist of 2D vectors of face displacement coordinates ($\Delta x$, $\Delta y$). Coordinates of centroids of the tracked region are more stable than, for example, region corner coordinates. Basic face states are ‘still’, moving ‘left’, ‘right’, ‘up’, ‘down’ and ‘down’. By combining this states we can obtain the semantic description of following activities {standing}, {moving}, {shaking}, {nodding}, {moving left}, {moving right} and {sitting}.

The stationary {standing} and {sitting} states are indistinguishable because there are no changes in face displacements. This is resolved by the inclusion of transitional activities {sitting down} and {standing up} which are identified as continuous occurrences of ‘down’ and ‘up’ states. Different granularities of basic states enable estimation of different activities.

4. Results

The proposed algorithm has been tested on several sequences. Due to lack of ground truth data, the tracking algorithms performance is confirmed qualitatively by recognized activities. Representative sequence is presented in Figure 4. The images in Fig. 4. are annotated with the recognized activities estimated from tracked regions. Activities are successfully recognized as {walking}, {standing}, {sitting down} and {sitting}. For each activity there is a life score which indicates the duration of activity. This is used to prune false recognitions. For example, if the sequence is recognized as {walking} but has a small life score than it undergoes further checking or it is discarded.

In general, the activity recognizer performs well except in few following cases. Recognition of activities is dependent on the tracking stability. Activities such as {shaking} and {nodding} can appear due to jittering of detection or tracking. Also, in case a person moves slowly, his/hers relative displacement is lost in the states which represent stationary position.

The main advantage of the activity recognition module is that it is simple and can operate independently with any other detection and tracking module.

Figure 4. Sample video sequence with the annotated recognized activities
5. Conclusion

In this paper an approach to human activity recognition has been presented. It is based upon face detection combined with a tracker which enables smooth tracking of human faces across multiple views. Face detections are fed to CAMShift tracker which is additionally controlled by Kalman filtering based tracker. The tracking information is used as an input to Finite State Machine to annotate activity of a tracked person. The next stage of our research will be focused towards better classification of activities which would include uncertainties provided by the detection and tracking, such as stochastic FSM with Viterbi algorithm [16]. Finally, more quantitative analysis of system performance will be performed using annotated test sequences.

References
