Multiple Face Tracking with Appearance Modes and Reasoning

Prithwijit Guha, Mayank Jain, Nipun Pande, Tavleen Oberoi
TCS Innovation Labs, New Delhi, India
{prithwijit.guha,mayank10.j,nipun.pande,tavleen.oberoi}@tcs.com

Abstract— *Multiple face tracking plays a key role in applications related to security-surveillance, human-computer interactions, video indexing etc. Existing literature in face tracking has mainly focused on facial features from a detection/recognition viewpoint. On the other hand, we believe that reasoning with detected/tracked face regions has a strong role in multiple face tracking. We propose a reasoning scheme that binds face localization (motion and mean-shift) and detection (Ada-Boost with Haar features) for tracking multiple faces in image sequences. The reasoning procedure identifies the cases of face isolation (unoccluded), grouping/occlusions, detection/tracking failure, entry/exit and re-appearance of faces. Instantiation of these cases are used as cues in selective update of the facial features. Additionally, we maintain a Normalized Face Cluster Set (NFCS) to capture the appearance modes for varying facial poses. These cluster sets are further used in discriminating new faces from the existing ones while restoring the tracks of the later. Experimental validation on four video sequences has shown significant tracking performance under occlusions.*

Keywords: Multiple Face Tracking, Mean-Shift Tracker, Normalized Face Cluster Set, Reasoning, Appearance Model

1. Introduction

Videos with human participants are often referred by the actors' names who in turn are identified by their faces. It is often required to extract the trajectories of such faces from visual input (videos or image streams) for a wide range of applications like security-surveillance (face logging of visitors), face based video indexing (making a video browsable through face content), human-computer interactions (locating and tracking multiple users in front of camera) etc. and thus, multiple face tracking plays a central role in such applications. Note that this is a sub-case of the more general problem of multiple object tracking. Existing literature in (multiple) object tracking is huge and a nice overview of the same can be found in [1], [2].

Existing works in multiple face tracking have generally focused on appearance features for single face detection/tracking, the most common being the color distributions learned from skin pixels. Color features of moving skin patches have been used to identify and track facial regions using particle filters in [3]. A dominance weighing scheme is proposed in [4] where skin pixels are assigned larger weights for enhancing the performance of color based face trackers. Cues obtained from feature point correspondences (optical flow) and probabilistic face detection are combined in a Bayesian framework for tracking faces in [5]. Motion cues have also been used in [6] where a multi-scale elastic matching based optical flow algorithm is proposed for tracking faces. More complex approaches involving Relevance Vector Machines (RVM) and boosted learning [7]; dynamic Bayesian networks for occlusion handling [8] and hidden Markov models [9] have also been proposed for detecting and tracking multiple faces. Additionally, a post-processing method using Earth Mover's Distance based K-NN classifier [10] is proposed to discriminate true face tracks from the outliers.

The goal of this work is to develop a case based reasoning scheme for tracking multiple faces using appearance and motion features. The proposed approach is based on a non-evanescence assumption on the faces and that their deformations are continuous. The faces are initially detected by using Haar feature based classifiers [11] and are localized in subsequent frames (Sub-section 2.1) by motion prediction initialized mean-shift iterations [12]. The correspondences between the tracked and detected face regions are established by computing fractional overlaps which lead to the identification of a set of events (Sub-section 2.3). Our work aims at detecting and handling the following events while tracking multiple faces under occlusions.

- Isolated, unoccluded and well tracked faces
- Track loss and disappearance of faces
- Face grouping
- Entry/Exit and Reappearance of faces

The extracted face regions in a certain track are clustered to capture the appearance modes obtained for varying facial poses which are used for face track restorations (Sub-section 2.2). Our algorithm makes no constraining assumptions regarding the facial appearance and motion features, camera parameters, scene background model etc. and demonstrates that only reasoning applied on a set of easily computable event predicates leads to an efficient multiple face tracking system. Some salient strengths of the proposed scheme are the following.

(a) Ability to distinguish a variety of problem situations using computationally simple event predicates.
(b) Using the above information to decide on the advisability of feature updates.
(c) Inherent ability of recognizing failure situations, should

they occur.

(d) Ability to automatic track restoration at a later time

(e) No constraining assumptions related to face appearance/motion or camera/environment models.

The proposed approach for multiple face tracking (Section 2) is described next.

2. Multiple Face Tracking

We have used the Haar feature based face detectors [11] (available with OpenCV [13]) to segment the regions of left/right profile or frontal faces in the image sequence. However, these detectors are extremely sensitive to the facial pose. Thus, although they are very accurate in detecting faces in left/right profile or frontal faces, they fail when the facial pose changes. It is also not practical to use a lot of detectors, each tuned to different face orientations as that would lead to both high memory and processor usage. Thus, a detection reduced to a local neighborhood search guided by face features is advantageous to satisfy real-time constraints. Such a necessity is achieved by the procedure of tracking. We initialize the tracker with a face detection success, continue tracking where detection fails (due to facial pose variations) and update the target face features at times when the detectors succeed during the frame presence of the face.

Tracking algorithms constitute schemes for "*object representation*" and a procedure for "*inter-frame object region correspondence*" [1]. Additionally, a "*reasoning*" method is required for handling different trackers in cases involving multiple objects [2]. We next describe the proposed face region representation/localization schemes (Sub-section 2.1) and the adopted methodology of reasoning for tracking multiple faces (Sub-section 2.3).

2.1 Face Representation and Tracking

A set of facial features are initially computed from the face region detected by one of the profile/frontal face detectors and are updated whenever the face is detected next, isolated from other detected/tracked face regions. We use the face bounding rectangle, motion history, color distribution and a mixture model learned on normalized face appearances as features for representing the face.

The location of the face F in the image is identified by the face bounding rectangle $\mathbf{BR}(F)$ with sides parallel to image axes. We use a second order motion model (constant jerk), continuously updated from the 3 consecutive centroid positions of $\mathbf{BR}(F)$. Using this model, The centroidal position $\hat{\mathbf{C}}_t(F)$ at the t^{th} instant is predicted as $\hat{\mathbf{C}}_t(F) = 2.5\mathbf{C}_{t-1}(F) - 2\mathbf{C}_{t-2}(F) + 0.5\mathbf{C}_{t-3}(F)$. The color distribution $\mathbf{H}(F)$ of the face F is computed as a normalized color histogram, position weighted by the Epanechnikov kernel supported over the maximal elliptical region $\mathbf{BE}(F)$ (centered at $\mathbf{C}(F)$) inscribed in $\mathbf{BR}(F)$ [12]. Mean-shift iterations initialized from the motion model predicted position converge to localize the target face region in the current image. The mean-shift tracking algorithm maximizes the Bhattacharya co-efficient between the target color distribution $\mathbf{H}(F)$ and the color distribution computed from the localized region at each step of the iterations. The maximum Bhattacharya co-efficient obtained after the mean-shift tracker convergence is used as the tracking confidence $tc(F)$ of the face F [12]. We combine this color based representation with an appearance model to encode the structural information of the face. The RGB image region within $\mathbf{BR}(F)$ is first resized and then converted to a $q \times q$ monochrome image which is further normalized by its brightest pixel intensity to form the normalized face image nF of the face F. The normalization is performed to make the face image independent of illumination variations (Figure 1).

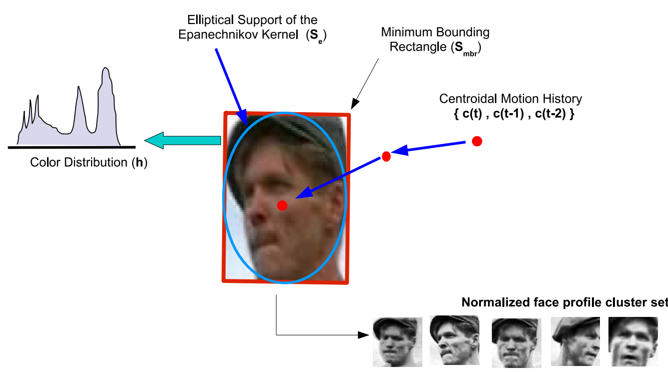


Fig. 1: Illustrating the face representation scheme. We use the bounding rectangle $\mathbf{BR}(F)$ of the face F, its second order motion model, the position weighted color distribution $\mathbf{H}(F)$ computed using the Epanechnikov kernel supported over the maximal ellipse in $\mathbf{BR}(F)$. The normalized face images obtained due to the facial pose variations are clustered to obtain the *Normalized Face Cluster Set* ($\mathbf{NFCS}(F)$).

2.2 Normalized Face Cluster Set

During the course of tracking, a person appears with various facial poses. We propose to cluster the normalized faces obtained from the different facial poses to learn the modes of his/her appearances thereby forming a *Normalized Face Cluster Set* ($\mathbf{NFCS}(F)$, henceforth). The normalized face image nF is re-arranged in a row-major format to generate the $d = q \times q$ dimensional feature vector $\mathbf{X}(nF)$. To achieve computational gain, we assume that the individual dimensions of the feature vector are un-correlated and hence, a diagonal co-variance matrix is sufficient to approximate the spread of the component Gaussians. A distribution over these feature vectors is approximated by learning a variant of the Gaussian mixture models where we construct a set of normalized face clusters.

The $\mathbf{NFCS}$ with K clusters is given by the set $\mathbf{NFCS} = \{(\mu_r, \sigma_r, \pi_r); r = 1, \ldots K\}$, where μ_r, σ_r are the respective mean and standard deviation vectors of the r^{th} cluster and the weighing parameter π_r is the fraction of the total number of normalized face vectors belonging to the r^{th} cluster. The $\mathbf{NFCS}$ initializes with $\mu_1 = \mathbf{X}(nF_1)$ and an initial standard deviation vector $\sigma_1 = \sigma_{init}$ and $\pi_1 = 1.0$.

Let there be K_{l-1} clusters in the $\mathbf{NFCS}$ until the processing of the vector $\mathbf{X}(nF_{l-1})$. We define the belongingness function $B_r(u)$ for the u^{th} dimension of the r^{th} cluster as $B_r(u)$ given by,

$$B_r(u) = \begin{cases} 1; & |\mathbf{X}(nF_l)[u] - \mu_r[u]| \leq \lambda\sigma_r[u] \\ 0; & \text{Otherwise} \end{cases} \tag{1}$$

λ being typically chosen between $2.5 - 5.0$ (Chebyshev's inequality [14]). The vector $\mathbf{X}(nF_l)$ is considered to belong to the r^{th} cluster if,

$$\sum_{u=1}^{d} B_r(u) \geq (1 - \eta_{mv})d \tag{2}$$

where $\eta_{mv} \in (0, 1)$ such that $\eta_{mv} \times d$ denotes the upper limit of tolerance on the number of membership violations in the normalized face vector. If $\mathbf{X}(nF_l)$ belongs to the r^{th} cluster, then its parameters are updated as,

$$\begin{aligned} \pi_r &\leftarrow (1 - \alpha_l)\pi_r + \alpha_l & (3) \\ \sigma_r^2[u] &\leftarrow (1 - \beta_r(l,u))[\sigma_r^2[u] + \beta_r(l,u)D_{lr}^2[u]] & (4) \\ \mu_r[u] &\leftarrow \mu_r[u] + \beta_r(l,u)D_{lr}[u] & (5) \end{aligned}$$

where $\alpha_l = \frac{1}{l}$, $\beta_r(l,u) = \frac{\alpha_l B_r(u)}{\pi_r}$ and $D_{lr}[u] = \mathbf{X}(nF_l)[u] - \mu_r[u]$. For all other clusters $r' \neq r$, the mean and standard deviation vectors remain unchanged while the cluster weight $\pi_{r'}$ is penalized as $\pi_{r'} \leftarrow (1 - \alpha_l)\pi_{r'}$. However, if $\mathbf{X}(nF_l)$ is not found to belong to any existing cluster, a new cluster is formed ($K_l = K_{l-1} + 1$) with its mean vector as $\mathbf{X}(nF_l)$, standard deviation vector as σ_{init} and weight $\frac{1}{l}$; the weights of the existing clusters are penalized as mentioned before.

It is worth noting that the parameter updates in equation 5 match the traditional Gaussian Mixture Model (GMM) learning [14]. In GMMs, all the dimensions of the mean vector are updated with the incoming data vector. However, here we update the mean and standard deviation vector dimensions selectively with membership checking to resist the fading out of the mean images. Hence, we call the $\mathbf{NFCS}$ as a variant of the mixture of Gaussians. Figure 1 shows a few mean images of the normalized face clusters learned from the tracked face sequences of the subject.

2.3 Handling Multiple Faces

Tracking multiple faces is not merely the implementation of multiple trackers but a reasoning scheme that binds the individual face trackers to act according to problem case based decisions. For example, consider the case of tracking a face which gets occluded by another object. A straight through tracking approach will try to establish correspondences even when the target face disappears in the image due to complete occlusion by some scene object leading to tracking failure. A reasoning scheme, on the other hand, will identify the problem situation of the disappearance due to the occlusion of the face and will accordingly wait for the face to reappear by freezing the concerned tracker. Our approach to multiple face tracking proposes a reasoning scheme to identify the cases of face grouping/isolation along with the scene entry/exit of new/existing faces.

The process of reasoning is performed over three sets, viz. the sets of *active*, *passive* and *detected* faces. The active set $\mathcal{F}_a(t)$ consists of the faces that are well tracked until the t^{th} instant. On the other hand, the passive set $\mathcal{F}_p(t)$ contains the objects for which either the system has lost track or are not visible in the scene. The set of detected faces $\mathcal{F}_d(t)$ contains the faces detected in the t^{th} frame. The system initializes itself with empty active/passive/detected face sets and the objects are added or removed accordingly as they enter or leave the field of view. During the process of reasoning, the objects are often switched between the active and passive sets as the track is lost or restored. We start the process of reasoning at the t^{th} frame based on the active/passive face sets available from the $(t-1)^{th}$ instant. The faces in the active set are first localized with motion prediction initialized mean-shift trackers (Sub-section 2.1. We compute the extent of overlap between the tracked face regions from the active set and the detected face regions to identify the isolation/grouping state of the faces. The reasoning scheme based on the tracked-detected region overlaps is described next.

Consider the case where m faces are detected ($\mathcal{F}_d = \{dF_j; j = 1 \ldots m\}$) while n faces were actively tracked till the last frame ($\mathcal{F}_a = \{aF_i; i = 1 \ldots n\}$). To analyze the correspondence between the tracked face region and the detected face area, we define the fractional overlap between the faces F_1 and F_2 as

$$\gamma(F_1, F_2) = \frac{|\mathbf{BR}(F_1) \cap \mathbf{BR}(F_2)|}{|\mathbf{BR}(F_1)|} \tag{6}$$

which signify the fraction of the bounding rectangle of F_1 overlapped with that of F_2. We consider the actively tracked face aF_i and the detected face dF_j to have significant overlap, if either of their fractional overlap with the other crosses a certain threshold η_{ad} (equation 8).

$$\begin{aligned} \mathbf{Overlaps}(aF_i, dF_j) \;\; \Rightarrow & \;\; [\gamma(aF_i, dF_j) \geq \eta_{ad}] & (7) \\ \vee & \;\; [\gamma(dF_j, aF_i) \geq \eta_{ad}] \end{aligned}$$

Let $\mathbf{S_{df}}(i)$ denote the set of detected faces which has significant overlap with the face aF_i in the active set and

$\mathbf{S_{af}}(j)$ represent the set of faces in the active set which has significant overlap with the detected face dF_j (equation 10).

$$\begin{aligned} \mathbf{S_{df}}(i) &= \{dF_k : [dF_k \in \mathcal{F}_d] \quad (8) \\ &\wedge \quad \mathbf{Overlaps}(aF_i, dF_k)\} \\ \mathbf{S_{af}}(j) &= \{aF_r : [aF_r \in \mathcal{F}_a] \quad (9) \\ &\wedge \quad \mathbf{Overlaps}(aF_r, dF_j)\} \end{aligned}$$

Based on the cardinalities of these sets associated with either of aF_i/dF_j and the tracking confidence $tc(aF_i)$, we identify the following situations during the process of tracking.

- **Isolation and Feature Update** – The face aF_i is considered to be isolated if it does not overlap with any other face in the active set, i.e. $\forall r \neq i \neg \mathbf{Overlaps}(aF_i, aF_r); aF_i, aF_r \in \mathcal{F}_a$. Under this condition of isolation of the tracked face, we update its features if there exists a pair (aF_i, dF_k) which significantly overlap only with each other and none else, i.e. $\exists k \mathbf{Overlaps}(aF_i, dF_k) \wedge [|\mathcal{S}_{df}(i)| = 1] \wedge [|\mathcal{S}_{af}(k)| = 1]$. In such a case, we are confident about the face boundaries on account of the face detection success and thus the color distribution and the motion features of aF_i are updated from the associated (detected) face dF_k.
- **Face Grouping** – The face is considered to be in a group (e.g. multiple persons with overlapping face regions) if the bounding rectangles of the tracked faces overlap. In such a case, even if a single detected face dF_k is associated to aF_i, we are not confident about the correspondence on account of multiple overlaps. Thus, in this case we only update the motion model of aF_i based on its currently tracked position.
- **Detection and/or Tracking Failure** – This is the case where due to facial pose variations, we fail to detect the presence of the face in the image. However, if the face aF_i is tracked well ($tc(aF_i) \geq \eta_{tc}$), we update only the motion model of aF_i. In absence of a confident face detection, we are not sure about the exact face boundaries and hence the color distribution is not updated. However, in case of both detection and tracking failure, aF_i is not associated with any detected face and the tracking confidence also drops below the threshold (η_{tc}). In this case, we consider aF_i to disappear from the scene (equation 11) and transfer it from $\mathcal{F}_a$ to $\mathcal{F}_p$.

$$\begin{aligned} \mathbf{Disappears}(aF_i) \quad &\Rightarrow \quad [|\mathcal{S}_{df}(i)| = 0] \quad (10) \\ &\wedge \quad [tc(aF_i) < \eta_{tc}] \end{aligned}$$

- **New Face Identification** – A new face in the scene does not overlap with any of the the bounding rectangles of the existing (tracked) faces. Thus, dF_j is considered a new face if $\mathbf{S_{af}}(j)$ is a null set.

$$\mathbf{NewFace}(dF_j) \Rightarrow |\mathbf{S_{af}}(j)| = 0 \quad (11)$$

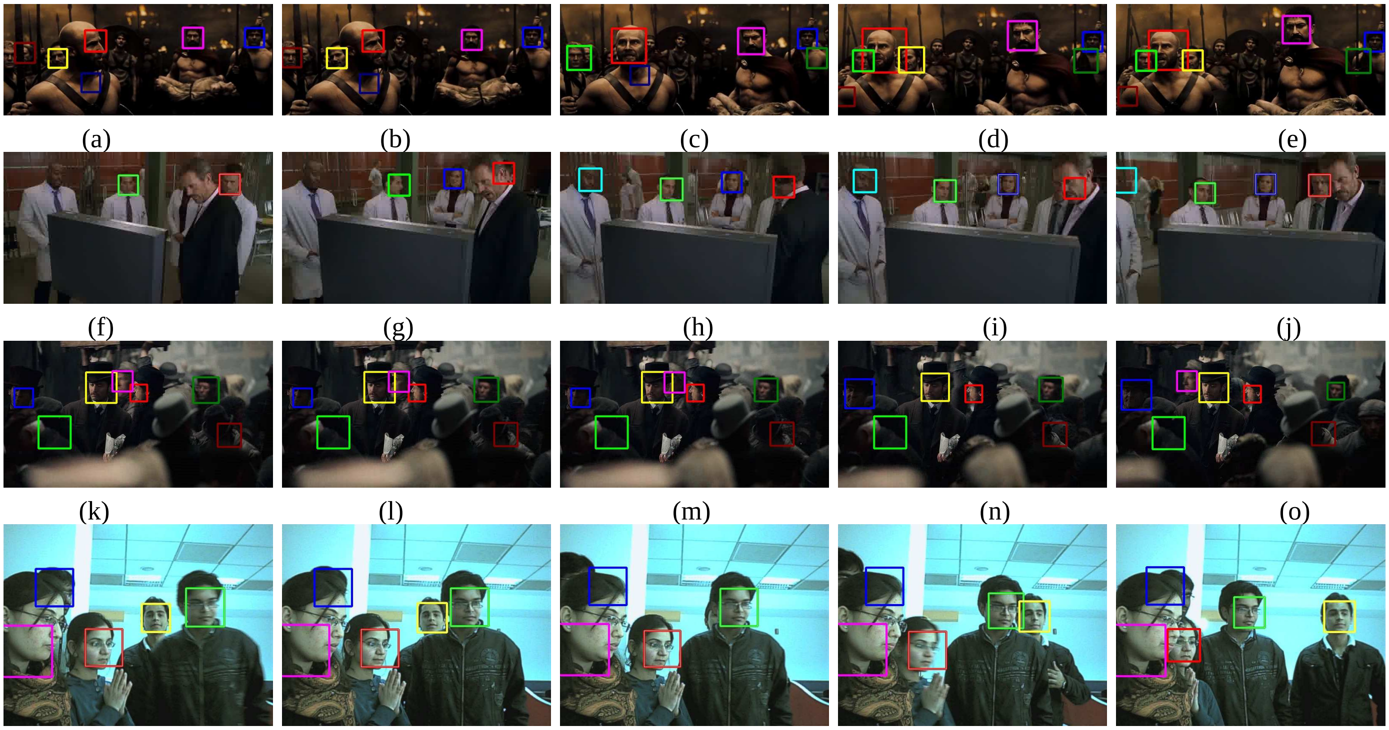
  Note that, the system might lose track of an existing face whose re-appearance is also detected as the occurrence of a new one. Hence, the newly detected face region is normalized first and checked against the $NFCS$ of the faces in $\mathcal{F}_p$ using the belongingness criterion outlined in Equation 2. If a match is found, the track of the corresponding face is restored by moving it from $\mathcal{F}_p$ to $\mathcal{F}_a$ and its color and motion features are re-initialized from the newly detected face region. However, if no matches are found, a new face is added to $\mathcal{F}_a$ whose color and motion features are learned from the newly detected face region.

During the course of multiple object tracking, the faces in the active set are identified in one of the above situations and the feature update or active to passive set transfer decisions are taken accordingly. By reasoning with these conditions, we initialize new trackers as new faces enter the scene and destroy them as the faces disappear.

3. Results

The proposed methodology is tested offline on the following 4 different sets of image sequences from the movies "*300*" (624 images) and "*Sherlock Holmes*" (840 images), "*House*" TV Series, Season 7 (495 images) and a sequence of 1116 images recorded in the lab with a webcam. We have empirically chosen the fractional overlap threshold as $\eta_{fo} = 0.1$, the tracking confidence threshold as $\eta_{tc} = 0.6$. The results of multiple object tracking in these videos are shown in figure 2. The proposed approach for multiple face tracking is implemented on a single core 1.6 GHz Intel Pentium-4 PC with semi-optimized coding and operates at 13.33 FPS (face detection stage included).

Performance Analysis – We present an object centric performance analysis by manually inspecting the surveillance log for computing the average rates of tracking precision and track switches. A tracker initialized over a certain face may eventually lose it on account of occlusions and may switch track to some other face(s) until an exit event. We consider the tracking to be successful if the localized region has a non-zero overlap with the actual face region in the image. Consider the case of a tracker with a life span of T frames, of which for the first T_{trk} frames, the tracker successfully tracks the same face over which it is initialized and then successively switches track to N_{switch} number of (different) faces(s) during the remaining $T - T_{trk}$ frames. The *tracking precision* of an individual object is then defined as $\frac{T_{trk}}{T}$ and the average tracking precision computed over the entire set of extracted faces is called the **Tracking Success Rate** for the entire video. In the same line, the **Tracker Failure Rate**



(a) (b) (c) (d) (e)
(f) (g) (h) (i) (j)
(k) (l) (m) (n) (o)
(p) (q) (r) (s) (t)

Fig. 2: Results of multiple face tracking under occlusions. (a)-(e) Movie *300* and (f)-(j) TV Series *House*, *Season* 7 – The faces are relatively unoccluded and tracked across the frames. (k)-(o) Movie *Sherlock Holmes* – The face marked with the pink bounding rectangle undergoes partial and full occlusion and is successfully tracked as it reappears. (p)-(t) Multiple faces are tracked under heavy occlusions on an image sequence recorded in the lab environment. Note the tracking performance for partial hiding to complete occlusion and re-appearance of the face marked with the yellow bounding box.

Table 1: Performance analysis of multiple face tracking.

	Tracking Success	**Tracker Failure**	**Tracker Re-initialization**
300 Figure 2(a)-(e)	85.10%	1.00	0.40
House Season 7 Figure 2(f)-(j)	87.80%	0.33	0.00
Sherlock Holmes Figure 2(k)-(o)	91.52%	0.40	0.40
Lab Sequence Figure 2(p)-(t)	81.20%	0.50	0.67

is evaluated as the average number of track switches over the entire set of extracted objects. After a track switch from the $T_{trk} + 1$ frame onwards, a different tracker may pick up the trail of this object through a track switch from some other face or through the initialization of a new tracker – let there be N_{reinit} number of tracker re-initializations on some face region. The **Tracker Re-initialization Rate** is defined as the average number of tracker re-initializations per face computed over the entire set of extracted faces. The performance of our tracking algorithm with respect to these measures are presented in table 1 .

4. Conclusion

We have proposed an algorithm for multiple face tracking through intelligent reasoning with a set of computationally simple event predicates. A face region is initially detected in one of left/right profile or frontal pose using Haar feature based boosted classifiers and is tracked in subsequent frames using motion prediction initialized mean-shift iterations. The tracked face regions belonging to a particular subject are normalized (to avoid illumination effects) and clustered to discover the appearance modes in various facial poses. The reasoning is performed by analyzing the fractional overlaps between tracked and detected face regions to identify a particular face in one of the states of isolation (no occlusion), grouping (with other faces), disappearance (near complete occlusions) or track loss and re-appearance/entry/exit. The proposed approach is not restricted by any camera or environment specific assumptions and thus performs satisfactorily in relatively unconstrained environments.

A natural extension to this work lies in improving the performance of the individual components – first, use of generalized face detectors (not restricted to full frontal or left/right profile pose); second, combining kernel based color chromaticity distribution trackers with facial appearance modes of $NFCS$ for improved performance under occlusions. This work has a number of applications in the fields of home entertainment, security surveillance and in indexing large video data sets. Multiple face tracking followed by time stamped face data logging has huge importance in the field of security surveillance for analyzing people traffic patterns in large buildings. Similarly, videos connected through similar faces (identified using matching $NFCS$) can be used as a cue for fast search in video databases, whether in the field of personal home video albums (small databases) or a storage of broadcast videos or collection of movies (very large databases). Integration of such indexing systems, coupled with language labels assigned to face tracks will lead to easily browsable videos where multiple face tracking will play a central role.

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Journal of Computing Surveys*, vol. 38, no. 4, 2006.

[2] P. Gabriel, J. Verly, J. Piater, and A. Genon, "The state of the art in multiple object tracking under occlusion in video sequences," in *Advanced Concepts for Intelligent Vision Systems*, 2003, pp. 166–173.

[3] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *European Conference On Computer Vision*, 2002, pp. 661–675.

[4] T. Yang, S. Z.Li, Q. Pan, J. Li, and C. Zhao, "Reliable and fast tracking of faces under varying pose," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 421–428.

[5] E. Loutas, I. Pitas, and C. Nikou, "Probabilistic multiple face detection and tracking using entropy measures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 128–135, 2004.

[6] S. M. Bhandarkar and X. Luo, "Integrated detection and tracking of multiple faces using particle filtering and optical flow-based elastic bunch graph," *Computer Vision and Image Understanding*, vol. 113, pp. 708–725, 2008.

[7] S. Shen and Y. Liu, "Efficient multiple faces tracking based on relevance vector machine and boosting learning," in *Congress on Image and Signal Processing*, 2008, pp. 161–165.

[8] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 789–795.

[9] S. Duffner and J.-M. Odobez, "Exploiting long-term observations for track creation and deletion in robust multi-face tracking," in *IEEE Conference on Automatic Face and Gesture Recognition*, 2011.

[10] X. Zhao, E. Delleandrea, and L. Chen, "A robust method for multiple face tracking using kalman filter," in *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009, pp. 67–72.

[11] P. Viola and M. Jones, "Robust real-time face detection," *International Journal on Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[12] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 142–149.

[13] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media Inc., 2008.

[14] R. Duda, P. Hart, and D. Stork, *Pattern Recognition*. 2nd Edition, John Wiley and Sons, 2003.