

Long-Term Online Multiface Tracking using Kalman Filter

Deepthi. S

Electronics and Communication Department
LMCST, University of Kerala
Trivandrum, India

Dr. Dinakar Das C. N.

Asso. Professor,
Electronics and Communication Department
LMCST, University of Kerala
Trivandrum, India

Abstract— The primary objective of multi-face tracking is to detect and track multiple faces in the current frame in the video sequence under various environmental conditions. Tracking of the face movement in the input frame of the video is the key process for various real time applications such as video-conferencing, human robotics or human computer interface or in the analysis of social interaction. The important step is to determine the path of the face. The various techniques to track the multiple moving faces in an input frame of video have been proposed. This paper gives a brief analysis of recent long-term online multi-face tracking algorithms based on Markov model and Kalman filtering.

Keywords—*Multi-Face Tracking; Long-Term Multiface Tracking; HMM (Hidden Markov Model); MCMC (Monte Carlo Markov Chain); PF (Particle Filter); KF (Kalman Filter)*

I. INTRODUCTION

The moving object tracking in video sequences has gained a great deal of interest in computer vision. Detection of moving objects in video streams is the first relevant step of information extraction in many computer vision applications, including traffic monitoring, automated remote video surveillance and people tracking. Conventional approach to object tracking is based on the difference between the current image and the background image. However, algorithms based on the difference image cannot simultaneously detect still objects. Furthermore, they cannot be applied to the case of a moving camera. Algorithms including the camera motion information have been proposed previously, but, they still contain problems in separating the information from the background. Object tracking has significance in real time environment because it enables several important applications such as security and surveillance to recognize people, to provide better sense of security.

Face detection is a computer technology that identifies human faces in digital images. It detects human faces which might then be used for recognizing a particular face. Face detection is used in biometrics, frequency as a part of a facial recognition system. Face detection is acquiring the interest of marketers. Face detection and recognition are challenging tasks due to variation in illumination, variability in scale, locate, orientation and pose. Facial expression, occlusion and lighting conditions also change the overall appearance of face.

Initially face-detection algorithms were concentrated on the detection of frontal human faces, however latest algorithms effort to solve the more common and difficult problem of multi-view face detection. That is, the recognition of faces that are either rotated along the axis from the face to the observer (in-plane rotation), or rotated beside the vertical or left-right axis (out-of-plane rotation), or both. The latest algorithms take into relative changes in the image or video by factors such as face appearance, lighting, and pose. Face detection using artificial neural networks was done by Rowley [1]. It is robust but computationally complex as the whole image has to be scanned at different scales and orientations. Feature-based (eyes, nose, and mouth) face detection is done by Yow *et al.* [2]. Statistical model of mutual distance between facial features are used to locate face in the image. Markov Random Fields have been used to model the spatial distribution of the grey level intensities of face images. Some of the eye location technique use infrared lighting to detect eye pupil. Eye location using genetic algorithm has been proposed by Wechsler [3]. Skin color is used extensively to segment the image, and localize the search for face. The detection of face using skin color fails when the source of lighting is not natural.

Face tracking is different from face detection in that face tracking uses temporal correlation to locate human faces in a video sequence, instead of detecting them in each frame independently. With temporal information, we can narrow down the search range significantly and thus make real-time tracking possible.

Recently there have been a lot of research efforts in face tracking. Yang and Waibel [4] built a real-time face tracking system based on normalized color. Bradski [5] proposed the continuously adaptive mean shift algorithm. Colmenarez, *et al.* [6], DeCarlo and Metaxas [7] used a 3D face model in the tracking process. Malsburg [8] tracked specific feature points on the face to track the face. However, none of these algorithms deal with multiple faces, especially occlusion between faces, effectively. In this paper, we propose a multiple face tracking algorithm based on constraining the speed and size changes of the faces.

Images containing faces are important to intelligent vision-based human computer interaction, and research struggles in face handling include face recognition, face tracking, pose estimation, and expression recognition, etc. However, many described approaches imagine that the faces in an image or an image sequence have been recognized and limited. To figure fully automated systems that examine the information enclosed in face images, robust and efficient face detection algorithms are necessary. Given an image, the objective of face detection is to recognize all image sections which comprise a face regardless of its three-dimensional location, positioning, and lighting situations. Such a problem is rebellious because faces are not harsh and have a high degree of irregularity in size, shape, color, and texture. Target tracking has a number of appliances such as human computer interface, security, Surveillance and video conferencing [9]. The human face attitudes even more problems than other matters since the human face is a vibrant object that comes in many shapes and colors. However, facial detection and tracking delivers many gains. Facial recognition is not achievable if the face is not inaccessible from the background. Human Computer Interaction (HCI) could greatly be enhanced by using reaction, pose, and signal recognition, all of which need face and facial feature detection and tracking. While many different algorithms occur to accomplish face detection, each has its own faults and powers. Some practice flesh tones, some use forms, and other are even more complex concerning templates, neural networks, or filters. These algorithms experience from the same problem; they are computationally pricey. An image is only a group of color and/or light intensity ideals. Observing these pixels for face detection is time utilizing and difficult to achieve because of the wide dissimilarities of shape and pigmentation within a human face. Pixels often need reanalysis for scaling and precision, Haar Classifiers, to quickly sense any object, containing human faces, using AdaBoost classifier cascades that are centered on Haar-like features and not pixels.

II. MULTIFACE TRACKING

Face tracking generally involves two stages:

- 1) Face Detection, where a photo is searched to find any face (shown here as a green rectangle), then image processing cleans up the facial image for easier recognition.
- 2) Face Tracking where that detected and processed face.

The previous papers use the algorithms as given below

A. Markov chain Monte Carlo(MCMC)

Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a large number of steps is then used as a sample of the desired distribution. The quality of the sample improves as a function of the number of steps. The most common

application of these algorithms is numerically calculating multi-dimensional integrals. In these methods, an ensemble of "walkers" moves around randomly. At each point where the walker steps, the integrand value at that point is counted towards the integral. The walker then may make a number of tentative steps around the area, looking for a place with reasonably high contribution to the integral to move into next. Random walk methods are a kind of random simulation or Monte Carlo method. However, whereas the random samples of the integrand used in a conventional Monte Carlo integration are statistically independent, those used in MCMC are correlated. A Markov chain is constructed in such a way as to have the integrand as its equilibrium distribution.

B. Hidden Markov Model(HMM)

A hidden Markov model (HMM) is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, recognition, part, musical score following, partial discharges and bioinformatics. A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Hidden Markov models can model complex Markov processes where the states emit the observations according to some probability distribution. One such example of distribution is Gaussian distribution, in such a Hidden Markov Model the states output is represented by a Gaussian distribution.

Moreover it could represent even more complex behaviour when the output of the states is represented as mixture of two or more Gaussians, in which case the probability of generating an observation is the product of the probability of first selecting one of the Gaussians and the probability of generating that observation from that Gaussian. In the hidden Markov models considered above, the state space of the hidden variables is discrete, while the observations themselves can either be discrete (typically generated from a categorical distribution) or continuous (typically from a Gaussian distribution). Hidden Markov models can also be generalized to allow continuous state spaces. Examples of such models are those where the Markov process over hidden variables is a linear dynamical system, with a linear relationship among related variables and where all hidden and observed variables follow a Gaussian distribution. In simple cases, such as the linear dynamical system just mentioned, exact inference is tractable (in this case, using the Kalman filter); however, in general, exact inference in HMMs with continuous latent variables is infeasible, and approximate methods must be used, such as the extended Kalman filter or the particle filter.

The parameter learning task in HMMs is to find, given an output sequence or a set of such sequences, the best set of state transition and output probabilities. The task is usually to derive the maximum likelihood estimate of the parameters of the HMM given the set of output sequences.

C. Reversible-jump Markov Chain Monte Carlo

Reversible-jump Markov chain Monte Carlo [10] is a generalization to standard Markov chain Monte Carlo (MCMC) procedure that permits replication of the posterior distribution on spaces of changing dimensions. Thus, the replication is possible even if the number of parameters in the models is not known.

D. Bayesian Filter

A Bayes filter [11] is an algorithm consumed in computer science for computing the probabilities of multiple beliefs to permit a robot to gather its location and direction. Basically, Bayes filters approve robots to always update their most likely location within a coordinate system, based on the most recently acquired sensor data. This is a recursive algorithm. It comprises of two parts: prediction and innovation. If the variables are straight and normally scattered the Bayes filter becomes equal to the Kalman filter. In a simple example, a robot traveling throughout a grid may have several different sensors that provide it with information about its surroundings. The robot may lead out with inevitability that it is farther from its original position, the robot has always less certainty about its position; using a Bayes filter, a probability can be assigned to the robot's belief about its current position, and that probability can be continuously updated from additional sensor information.

III. TARGET CREATION AND REMOVAL

The way objects are added and removed from the tracker is a key feature of the proposed algorithm. In previous work [12], [13], target creation and removal are directly integrated into the probabilistic tracking framework. However, this requires global scene likelihood models which are difficult to obtain in this type of application. Our goal is to achieve a high precision during tracking, *i.e.* we would like to avoid as much as possible false alarms. This means that the tracker should be able to detect as quickly as possible if there is a tracking failure; simultaneously, it should not stop tracking when there is no failure, since the algorithm may have to wait for a long time before the face is detected again. Surprisingly, this problem has received little attention in the past.

We propose to use two different Hidden Markov Models (HMM) for that purpose, as described in the following sections. One is used for object creation and the other for object removal, and they receive different types of observations.

A face detector (for both frontal and profile views) is called every 10 frames (*i.e.* roughly once per second, as our algorithm is able to process around 10 frames/s in real time). The HMMs are updated only at these instants, but rely on observations computed on all frames since the last update. According to our experiments, applying the detector to every frame did not greatly improve the tracking performance and considerably slowed down the algorithm. Detection gets associated with a target if their distance is smaller than two times their average width. Naturally, only un-associated detections are considered for the initialisation of a new target.

Before the creation and removal step, each detection is associated to a track provided the following conditions hold:

- 1) the detection is not associated with any other target,
- 2) it has the smallest distance to the tracked target,
- 3) the distance between detection and target is smaller than two times the average width of their bounding boxes,
- 4) the two bounding boxes overlap.

Although a more generic way would be to use training data to learn the association rules and parameters as done in [14], for instance, the above conditions work well for our data in the large majority of cases.

In the following, we describe the HMMs for target creation and removal. Note that naturally, only un-associated detections are considered for the initialisation of a new target.

A. Creation

When initialising a new target we have two objectives: first, minimise erroneous initialisations due to false detections, and second, initialise correct targets as early as possible.

For deciding when to add new targets to the face tracker, we propose a simple HMM that estimates the probability of a hidden, discrete variable c_t (i, j) indicating at each image position (i, j) if there is a face or not at this position. Fig.1.1 illustrates the model. In the following, we drop the (i, j) indices for clarity. Let us denote by $\mathbf{O}_t^c = [o_{t,1}^c, \dots, o_{t,N_c}^c]$ the set of N_c observations at each time step t , and by $\mathbf{O}_{1:t}^c = [\mathbf{O}_1^c, \dots, \mathbf{O}_t^c]$ the sequence of observations from time 1 to time t . Assuming the transition matrix is defined as: $p(c_t | c_{t-1}) = 1$ iff $c_t = c_{t-1}$ and 0 otherwise, the posterior probability of the state c_t can be recursively estimated as:

$$p(c_t = s | \mathbf{O}_{1:t}^c) = \frac{p(\mathbf{O}_t^c | c_t = s) p(c_{t-1} = s | \mathbf{O}_{1:t-1}^c)}{\sum_{s'} p(\mathbf{O}_t^c | c_t = s') p(c_{t-1} = s' | \mathbf{O}_{1:t-1}^c)} \quad (1)$$

Where

$$p(\mathbf{O}_t^c | c_t) = \prod_{i=1}^{N_c} p(o_{t,i}^c | c_t) \quad (2)$$

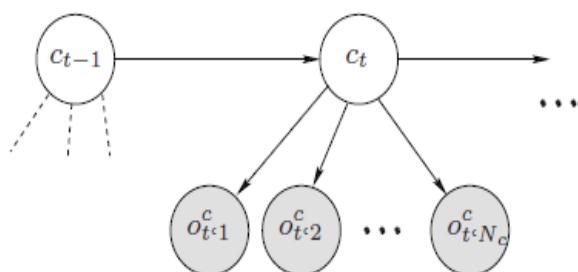


Fig. 1.1. HMM used at each image position for tracker target creation. The variable c_t indicates a face centred at a particular image position. The probability of c_t is estimated recursively using the observations $o_{t,1}^c, \dots, o_{t,N_c}^c$

B. Removal

During tracking, we want to assess at each point in time if the algorithm is still correctly following a face or if it has lost track. The algorithm can lose track, for example, when it gets distracted by a similar background region or when a person leaves the scene. More concretely, the objective is to interrupt the tracking as soon as possible if a failure occurs, and to continue tracking otherwise, even when a face has not been detected and associated with the track for a long time.

In a way similar to target initialization, we propose to use for each tracked face i an HMM estimating at each time step t the hidden status variable $k_{i,t}$ indicating correct tracking ($k_{i,t} = 1$) or tracking failure ($k_{i,t} = 0$). Fig. 1.2 will illustrate the model.

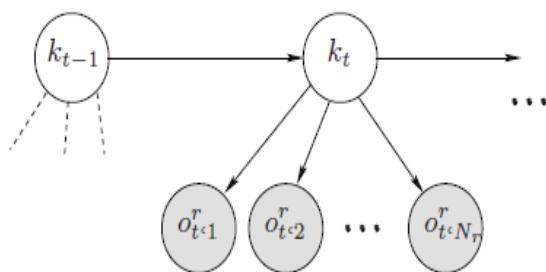


Fig. 1.2. HMM for target removal, used for each tracked face. The variable k_t indicates if a given face is still tracked correctly or if a failure occurred. The probability of k_t is estimated recursively using the observations $o_{t,1}^r, \dots, o_{t,N_r}^r$

IV. TRACKING USING KALMAN FILTER

The particle filter which is used for multiface tracking is having several advantages and is as follows:

- No restrictions in model – can be applied to non-Gaussian models, hierarchical models etc.
- Global approximation.
- Approaches the exact solution, when the number of samples goes to infinity.
- In its basic form, very easy to implement.
- Superset of other filtering methods – Kalman filter is a Rao-Blackwellized particle filter with one particle.

Although they have advantages, there are some disadvantages for the existing particle filters and are as follows:

- Computational requirements much higher than of the Kalman filters.
- Problems with nearly noise-free models, especially with accurate dynamic models.
- Very hard to find programming errors (i.e., to debug).
- The most efficient number of particles cannot be calculated.
- High computational complexity
- It is difficult to determine optimal number of particles

- Number of particles increase with increasing model dimension
- Potential problems: degeneracy and loss of diversity
- The choice of importance density is crucial

The Kalman filter permits one to adjust a model of some physical process. The purpose is to determine the parameters of an a priori model. The algorithm of the Kalman filter has several advantages. This is a statistical technique that adequately describes the random structure of experimental measurements. This filter is able to take into account quantities that are partially or completely neglected in other techniques (such as the variance of the initial estimate of the state and the variance of the model error). It provides information about the quality of the estimation by providing, in addition to the best estimate, the variance of the estimation error. The Kalman filter is well suited to the online digital processing. Its recursive structure allows its real-time execution without storing observations or past estimates. The main advantage of the Kalman filter is its ability to provide the quality of the estimate (i.e., the variance), and its relatively low complexity. However, its main disadvantage is that it provides accurate results only for Gaussian and linear models.

Table 1.1 Accuracy Comparsion

METHOD I		METHOD II	
Tracking : Particle Filter		Tracking : Kalman Filter	
Data set	Accuracy	Data set	Accuracy
1	80.3	1	94.4
2	67.6	2	69
3	74.6	3	78.9
4	87.3	4	88.7

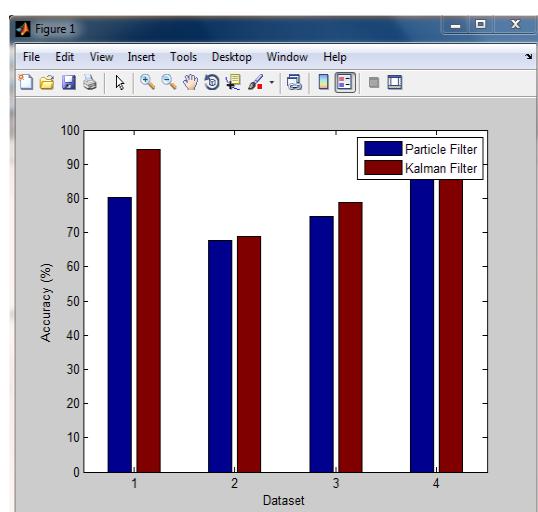


Fig. 1.3. Accuracy Plot of Particle and Kalman Filter

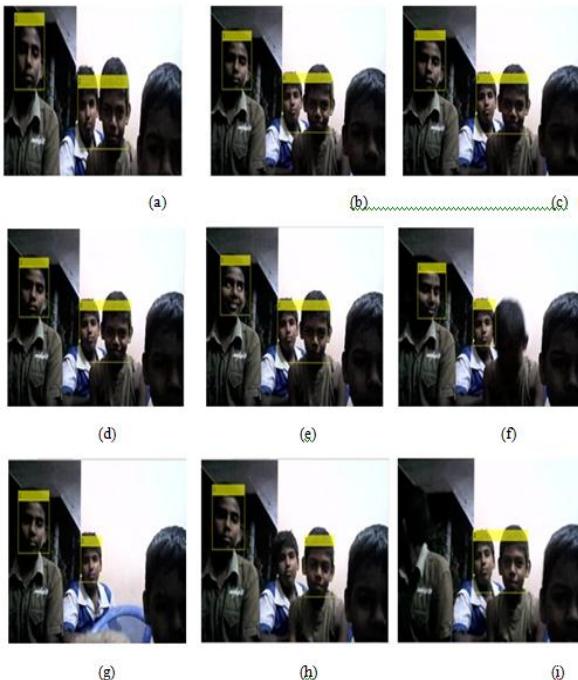


Fig.1.4: Detection and Tracking Results of Kalman Filter: (a) Frame No.99 (b) Frame No.152 (c) Frame No.174 (d) Frame No.200 (e) Frame No. 247(f) Frame No. 288(g) Frame No. 306(h) Frame No.352 (i) Frame No.398

V. CONCLUSION AND FUTURE WORK

In many visual multi-objects tracking applications, the question when to add or remove a target is not trivial due to, for example, and erroneous outputs of object detectors or observation models that cannot describe the full variability of the objects to track. This decision process is difficult due to object detector deficiencies or observation models that are insufficient to describe the full variability of tracked objects and deliver reliable likelihood (tracking) information. The proposed algorithm addresses the track management issue and presents a real-time online multiface tracking algorithm that effectively deals with the above difficulties.

An on-line multi-face tracking algorithm that effectively deals with situations where detections are rare or uncertain is presented. To achieve this, long-term observations from the image and the tracker itself are collected and processed in a principled way using two separate HMMs, deciding on when to add and remove a target to the tracker.

We can present a real-time, online multi-face tracking algorithm that effectively deals with missing or uncertain detections in a principled way. Tracking is formulated in a multi-face kalman filter framework. The size, top-left coordinate and velocity of motion of the detected face being

the parameters of the Kalman vector; the predicted values are used to locate faces in the next frame. Faces are redetected and the templates are updated at discrete time intervals when the similarity measures, between the faces detected and respective face templates, are less than a preset threshold.

This method is applied on real-time videos and shows a significant performance increase, compared to a traditional approach relying on head detection and likelihood models only.

REFERENCES

- [1] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.
- [2] K. C. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing*, 15(9):713–735, 1997.
- [3] J. Huang and H. Wechsler. Eye location using genetic algorithm. In *Proceedings, Second International Conference on Audio and Video-Based Biometric Person Authentication*, pages 130–135, March 1999.
- [4] J. Yang and A. Waibel, “A Real-Time Face Tracker,” *Proceedings of WACV’96*, pp. 142-147.
- [5] G. Bradski, “Computer Vision Face Tracking for Use in a Perceptual UserInterface,” http://developer.intel.com/technology/itj/q21998/articles/art_2.htm
- [6] A. Colmenarez, R. Lopez and T. Huang, “3D Model-Based Head Tracking,” *Visual Communication and Image Processing*, San Jose, CA, 1997
- [7] D. DeCarlo and D. Metaxas, “Deformable Model-Based Face Shape and Motion Estimation,” *IEEE Proc. of ICFG*, 1996.
- [8] T. Maurer and C. Malsburg, “Tracking and Learning Graphs and Pose on Image Sequences of Faces,” *IEEE Proc. of ICFG*, pp. 176-181, 1995.
- [9] Stefan Duffner and Jean-Marc Odobez “Track Creation and Deletion Framework for Long-Term Online Multiface Tracking” *IEEE Transactions on image processing*, vol. 22, no. 1, january 2013.
- [10] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, “Integrated person tracking using stereo, color, and pattern detection,” *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 175–185, 2000.
- [11] M. Yang, Y. Wu, and G. Hua, “Context-aware visual tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, Jul.2009.
- [12] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. *IEEE Trans. on PAMI*, 27(11):1805–1918, 11 2005.
- [13] Jian Yao and Jean-Marc Odobez. Multi-camera multi-person 3D space tracking with MCMC in surveillance scenarios. In *ECCV, workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*, Marseille, France, October 2008.
- [14] M. Richardson and P. Domingos, “Markov logic networks,” *Mach. Learn.*, vol. 62, nos. 1–2, pp. 107–136, 2006.
- [15] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, “Integrated person tracking using stereo, color, and pattern detection,” *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 175–185, 2000.