DISTRIBUTED PARTICLE FILTER TRACKING WITH ONLINE MULTIPLE INSTANCE LEARNING IN A CAMERA SENSOR NETWORK

Zefeng Ni, Santhoshkumar Sunderrajan, Amir Rahimi, B.S. Manjunath

Department of Electrical and Computer Engineering
University of California Santa Barbara, CA, USA

ABSTRACT

This paper proposes a distributed algorithm for object tracking in a camera sensor network. At each camera node, an efficient online multiple instance learning algorithm is integrated with particle filter for camera's image plane tracking. To improve the tracking accuracy, each camera node shares its particle states with others and fuses multi-camera information locally. In particular, particle weights are updated according to the fused information. The effectiveness of the proposed algorithm is demonstrated on human tracking in challenging environments.

Index Terms— Camera sensor network, Distributed tracking, Particle filter, Multiple instance learning

1. INTRODUCTION

Object tracking using multiple cameras is a key step to many applications such as video surveillance. This paper proposes a consistent tracking algorithm using a distributed smart camera network where each camera node has its own processing power and it can communicate with each other. Conventionally, object tracking in a camera network is realized in two steps, 1) Visual tracking of the objects on the image plane at each camera node; and 2) Fusion of information on a global ground plane.

To reduce the system complexity, the above mentioned steps are often done in an open-loop sequential manner. Object tracking on the image plane of individual camera is often considered to be a solved problem when using a camera network. Unfortunately, this is not the case even with the state-of-art tracking algorithm. Robust visual tracking is still an open issue in real life tracking applications, e.g. rapid appearance changes in objects, lighting changes, occlusions, etc.

The tracking accuracy of a camera network can be improved through data fusion by exploiting the redundancy in multiple cameras with overlapping fields of views. The information can be fused either at a central/head node [1, 2, 3] or by a distributed consensus algorithm [4]. However, no matter how robust a fusion algorithm is, the entire object tracking process might still fail because of the inaccurate visual tracking on each camera's image plane.

The main bottleneck for robust object tracking in a camera network is the low level visual tracking at individual camera nodes. This paper proposes a closed-loop interaction between visual tracking on the image plane and data fusion on the global space. In other words, the fusion result is used as feedback to enhance the local tracking, as illustrated in Fig. 1 and Fig. 2. At each camera node, a learning-based tracking algorithm (i.e. a discriminative appearance model [5, 6] using multiple instance learning) and local particle filtering are used to track object's location on the image plane. Considering a synchronized and calibrated camera network, particles (estimated from the object's tracked blob) from individual camera nodes are then shared with each other over the network. At each camera node, a mixture of Gaussians are fit over all the particles and is used to drive a global ground plane Kalman filter. Then, the local particle's weight is re-adjusted based on Kalman filter posterior state density.

Compared with similar works in the literature, the main contributions of this paper can be summarized as follows

- For visual tracking at each node, multiple instance learning is used to learn a discriminative appearance model to deal with the appearance changes. This is combined with particle filter by generating the training examples based on particle distribution (section 3.1).
- A distributed fusion algorithm is proposed to fuse all the shared particles from multiple cameras and update the local particle weights based on the global ground plane Kalman filter(section 3.2).

2. RELATED WORK

Multi-sensor fusion and tracking have a long history in signal processing, control theory, and robotics [7]. There have been many efforts recently on tracking objects in a camera network setup [1, 2, 3, 4, 8]. However, most of these methods are not distributed, i.e. require a central server to collaborate all cameras. Most methods do not consider jointly the visual tracking on the image planes and the data fusion on the ground plane. The closest works to the proposed method in terms of the basic framework are [2] and [4].

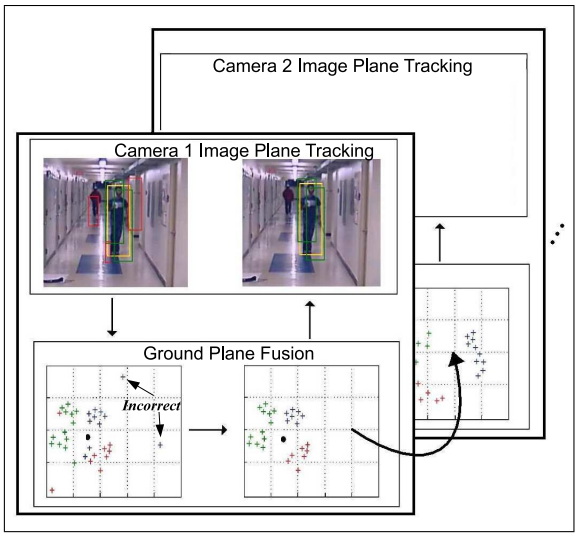

Fig. 1. Closed loop interaction between object's image plane tracking and ground plane fusion. As illustrated here, incorrect particles could be discarded because of the fusion.

In [4], a distributed Kalman consensus filter is used to fuse noisy measurements (object's ground location) from individual cameras to improve global tracking results and achieve a consensus at all the camera nodes. Although their proposed method is distributed, they assume that local object tracking on the camera's image plane is already solved. In particular, they model the ground plane measurements from each camera node with a simple Gaussian distribution. This might not be robust given the complex nature of vision-based tracking. Instead, in this paper, a set of particles are used to model the ground plane measurements from each camera node. In addition, the proposed method uses the fusion result to improve the visual tracking on the camera's image plane without any additional communication requirements. In [4], there is no such closed-loop interaction between the two modules.

In [2], a multi-camera people tracking algorithm based on collaborative particle filters is discussed. In particular, a target is tracked on both individual camera's image plane and on the ground plane by individual particle filters. The fusion results on the ground plane are incorporated by each camera as a boosted proposal function. This help to re-adjust particles for each camera's image plane tracking. This is similar to the proposed method. However, in the proposed method, re-adjusted local particles are not only used as a starting point for tracking in the next frame but also used to generate a more-representative set of training examples to update a discriminative classifier. This is due to the treatment of tracking problem as a classification problem so that both the object and the background are parts of the model. This method (often called "tracking by detection") achieves much more robust visual tracking compared to the classical color observation model used by [2]. In addition, in this paper, the fusion is achieved in a distributed manner without any central fusion module (needed in [2]). Therefore, there are no additional message exchanges in order to send the fusion results as feedback to camera node's local image plane tracking.

As described above, this paper uses a learning based tracking algorithm at each camera node to track a target on the image plane, i.e. use a discriminative classifier as the adaptive appearance model. By learning the appearance of both foreground and background, this kind of approaches that training a model to separate the object from the background via a discriminative classifier have been shown to achieve superior results [5, 6, 9]. To address the problem of noisy training examples when updating the discriminative classifier, a online multiple instance learning (MIL) algorithm similar to the MILBoost proposed by Babenko et al. [5] is used. In particular, this paper combine the MIL tracking in [5] with a particle filter. Details will be discussed in section 3.1.

3. PROPOSED TRACKING APPROACH

Fig. 2 shows an overview of distributed tracking system in a fixed camera network. Assuming that object has already been detected, the detected object serves as the input to the tracking system. In addition to this, cameras are assumed to be pre-calibrated with respect to the global ground plane.

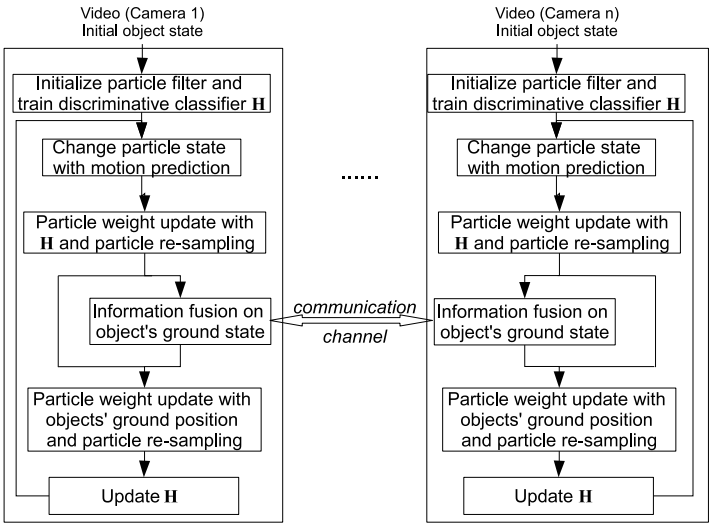

Fig. 2. Proposed Distributed Tracking Framework

3.1. Image Plane Tracking

Generally, any tracking system consists of three main components: image representation, appearance model and motion model. In this paper, image representation (computed for each image patch x) consists of a set of Haar-like features[10] and a special culture color histogram [11]. The discriminative classifier $\mathbf{H}$ forms the appearance model using a multiple

instance learning algorithm similar to [5]. Given an image patch x, and its binary label $y = 1$ indicating the presence of the tracked object, the instance probability $p(y = 1|x) = 1/(1 + exp(-\mathbf{H}(x)))$ ($p(y|x)$ is used as shorthand).

At frame t, tracker maintains the object state $\mathbf{O}_t = [row, col, scale]^{\mathrm{T}}$ with a particle set $\{\mathbf{O}_t^{(l)}, \pi_t^{(l)}\}$, where $[row, col]$ is the position on the image plane. Given a particle set from previous frame $t-1$, a basic sequential importance re-sampling particle filter is used to update the particles as follows:

1. Generate an updated particle set by sampling from the proposal distribution (assumed to be Gaussian here), $\mathbf{O}_t^{(l)} \sim p(\mathbf{O}_t^{(l)}|\mathbf{O}_{0:t-1}^{(l)}) = \mathcal{N}(\mathbf{O}_t^{(l)}; \mathbf{O}_{0:t-1}^{(l)}, \Psi)$, where Ψ is the covariance matrix of the state variables. I.e., state dynamics is modeled using a Brownian motion.

2. Re-weight each particle l according to $\mathbf{H}$: $\pi_t^{(l)} \propto \pi_{t-1}^{(l)} p(y|x(\mathbf{O}_t^{(l)}))$.

3. Re-sample particles from current particle set.

Once object's state is updated (after fusing the measurements from other cameras) at frame t, the tracker updates the appearance model $\mathbf{H}$ using the particle set. In particular, training examples for the classifier learning are generated based on the updated particle distribution. This is more robust compared with the greedy method of training example generation in [5].

3.2. Information Fusion across Multiple Camera Views

At the camera node i , every particle $\mathbf{O}_i^{(l)}(t)$ (i.e. an rectangular blob) corresponds to a possible location of the object's position $\mathbf{Z}_i^{(l)}(t) = [G_x(t), G_y(t)]$ on the ground plane. To avoid the complex task of detecting the intersection point of the visual object and the ground plane, the object's ground position is estimated by mapping the lower-middle image point of the blob to the ground plane with a pre-computed Homography. This simple method of ground position measurement is computationally efficient at the expense of noisy positions, as the particles are already noisy in nature. Therefore, a conventional Gaussian measurement noise assumption, as in [1, 4], would not be valid. In other words, the cameras have to share their particles, $\{\mathbf{Z}_i^{(l)}(t)\}$, directly with each other instead of a more compact distribution parameters.

To reach consensus among all the cameras and reduce the measurement noise, a mixture of Gaussians is fit over the $\{\mathbf{Z}_i^{(l)}(t)\}$ from all the cameras. Let $\mu_t^{(g)}$ and $\mathbf{P}_t^{(g)}$ be the mean and covariance of the fitted G Gaussians. Define measurement m_t and measurement noise covariance R_t as

$$m_t = \sum_{g=1}^{G} \alpha_t^{(g)} \mu_t^{(g)} \qquad (1)$$

$$R_t = \sum_{g=1}^{G} (\mathbf{P}_t^{(g)} + (\mu_t^{(g)} - m_t)(\mu_t^{(g)} - m_t)^T) \qquad (2)$$

where $\alpha_t^{(g)}$ is the weight of gth Gaussian component.

At each camera node, a Kalman filter is used to estimate the object's ground position. With the new measurement m_t and R_t, the state of the Kalman filter is updated.

From the posterior density of the Kalman filter, the distribution of object's ground plane position, $\mathbf{P}([G_x(t), G_y(t)])$, is obtained. The particle weights $\pi_t^{(l)}$ of $\mathbf{O}_i^{(l)}(t)$ can then be updated with $\mathbf{P}(\mathbf{Z}_i^{(l)}(t))$.

By re-weighting particles and re-sampling, incorrect particles (outliers) can be removed (See Fig. 1). This gives a more reliable estimation of the object's state in the image plane. These refined particles (image patches) are then used to generate a more representative set of training examples for updating the appearance model. This forms a closed loop between the image plane tracking and the global ground plane fusion (Particle weighting with model $\mathbf{H}$→Ground plan fusion →Particle re-weighting →Updating $\mathbf{H}$). Note that, each camera replicates this fusion operation at its local node. There is no need for a central controller. In addition, the sharing of particles across cameras does not put much burden on the communication channel compared with methods that requires sending image data across cameras.

4. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of the proposed method, it is tested with our own dataset with some complex scenarios consisting of five camera sensor network along a long corridor (See Fig. 3). In particular, we show how the proposed algorithm deal with the complex shapes and appearance changes in humans, and the unexpected illumination variations (e.g. shiny floor surface, shadows, sudden lighting changes caused by door-opening and closing etc.). During experiments, 200 particles are used at each camera node.

Fig. 3. Five Camera Sensor Network

Fig. 4 shows how the information fusion helps in cleaning up the particles. Before the fusion, object's ground plane position $\{\mathbf{Z}_i^{(l)}(t)\}$ is directly obtained from the particles (tracked blobs) of all the cameras and hence it is very noisy. With the fusion (section 3.2), object's position estimate on the ground plane becomes less noisy (Fig. 4(c)) by exploring the multiple camera's redundancy. With the updated global ground plane estimation, weights of the noisy particles on the image plane at each node are scaled down. As illustrated in Fig. 4(a,b) for

cameras 2 and 3, some of the non-conforming particles are discarded after particle re-sampling due to low weights.

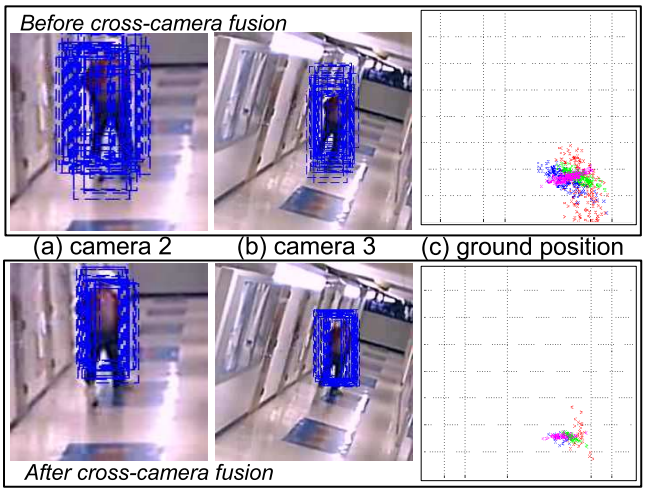

Fig. 4. Particle refinement due to information fusion across cameras. Each blue box represents a particle state. Different particle color indicates different cameras (Best viewed in color).

Fig. 5 shows how the proposed closed loop framework improves the local image plane tracking. In camera 1, when two objects get close to each other, some of the particles cover the wrong object. Since the learning algorithm uses these particles to generate the training examples, the appearance model $\mathbf{H}$ starts capturing some features from the wrong object and loses track of the correct object eventually. With the closed loop, wrong particles get discarded after the fusion and more representative particles are kept intact for the learning algorithm. As seen in the Fig. 5, even though the camera 1 has some difficulties in tracking the object, other cameras might have a clear view of the same object (such as camera 3 showed the figure).

5. CONCLUSION

In this paper, a distributed object tracking algorithm in a camera sensor network is proposed. At each camera node, multiple instance learning is effectively coupled with particle filter for local image plane tracking. Particles are shared across different cameras. Information from multiple cameras are fused on the ground plane and the fused information is used to re-weight the particles. The distributed nature of this algorithm keeps the communication channel free from frequent data exchanges (no image related data is shared). The proposed algorithm is fully distributed and it gives robust tracking results in-spite of noisy measurements at each node. For future work, we plan to extend the system for multiple object tracking where we need to associate objects across cameras.

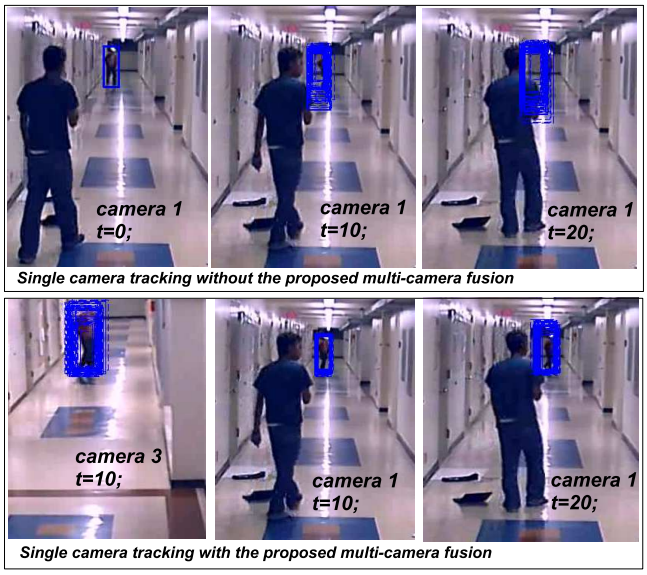

Fig. 5. Improvement in local image plane tracking with the proposed closed loop tracking algorithm.

6. REFERENCES

[1] C. Micheloni, G. L. Foresti, and L. Snidaro, "A network of co-operative cameras for visual surveillance," *IEE Proceedings Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 205– 212, Apr. 2005.

[2] Wei Du and Justus Piater, "Multi-camera people tracking by collaborative particle filters and principal axis-based integration," in *Asian Conference on Computer Vision (ACCV)*, 2007, pp. 365–374.

[3] Henry Medeiros, Johnny Park, and Avinash C. Kak, "Distributed object tracking using a cluster-based kalman filter in wireless camera networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 448–463, Aug. 2008.

[4] Cristian Soto, Bi Song, and Amit K. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2009, pp. 1486–1493.

[5] B. Babenko, M.-H Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 983–990.

[6] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *British Machine Vision Conference (BMVC)*, 2006, pp. 47–56.

[7] Reza Olfati-Saber and Nils F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in *American Control Conference*, June 2008, pp. 3157–3162.

[8] Saad M. Khan and Mubarak Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505 –519, Mar. 2009.

[9] S. Avidan, "Ensemble tracking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 494–501.

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

[11] Gang Wu, Amir Rahimi, Edward Y. Chang, Kingshy Goh, Tomy Tsai, Ankur Jain, and Yuan-Fang Wang, "Identifying color in motion in video sensors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 561–569.