Distributed Face Tracking and Recognition in Camera Networks

Miss M. M. Punse **Dr. Mrs. S. N. Kale** **Dr.V.M.Thakare**

Abstract **:- Networks of video cameras are being installed in many applications, e.g., surveillance and security, disaster response, environmental monitoring, etc. Currently, most of the data collected by such networks is analyzed manually, a task that is extremely tedious and reduces the potential of the installed networks. Tracking and activity recognition are two fundamental tasks in this regard. For multitarget tracking in a distributed camera network, the Kalman-Consensus algorithm can be adapted to take into account the directional nature of video sensors and the network topology. For the activity recognition problem, a probabilistic consensus scheme is derived which combines the similarity scores of neighboring cameras to come up with a probability for each action at the network level. This paper presents a distributed multicamera face tracking system suitable for large wired camera networks. Instead, an efficient camera clustering protocol is used to dynamically form groups of cameras for in-network tracking of individual faces. The clustering protocol includes cluster propagation mechanisms that allow the computational load of face tracking to be transferred to different cameras as the target move.**

Keywords **: Activity recognition, camera networks, consensus, distributed image processing, tracking.**

I. INTRODUCTION

Camera networks are being deployed for various applications like security and surveillance, disaster response and environmental modeling. However, there is little automated processing of the data. Moreover, most methods for multi camera analysis are centralized schemes that require the data to be present at a central server. In many applications, this is prohibitively expensive, both technically and economically. In this paper, we investigate distributed scene analysis algorithms by leveraging upon concepts of consensus that have been studied in the context of multi agent systems, but have had little applications in video analysis. Each camera estimates certain parameters based upon its Own sensed data which is then shared locally with the neighboring cameras in an iterative fashion, and a final estimate is arrived at in the network using consensus algorithms.

Due to broad establishment of surveillance video camera systems in recent years in both public and private venues, the recognition/verification of the subjects is often of interest and importance for purposes such as security monitoring, access control, etc. Some biometric traits such as gait can be used to recognize different subjects; however, it is preferred to use more distinct biometric clues such as face to identify a subject. The demand for robust face recognition in real-world surveillance cameras is increasing due to the needs of practical applications such as security and surveillance.

For many applications, for a number of reasons it is desirable that the video analysis tasks be decentralized. For example, there may be constraints of bandwidth, secure transmission, and difficulty in analyzing a huge amount of data centrally. In such situations, the cameras would have to act as autonomous agents making decisions in a decentralized manner. At the same time, however, the decisions of the cameras need to be coordinated so that there is a consensus on the state of the target even if each camera is an autonomous agent. Thus, the cameras, acting as autonomous agents, analyze the raw data locally, exchange only distilled information that is relevant to the collaboration, and reach a shared, global analysis of the scene.

II. BACKGROUND

Although there are a number of methods in video analysis that deal with multiple cameras, and even camera networks, distributed processing in camera networks has received very little attention. Methods have been developed for reaching consensus on a state observed independently by multiple sensors. However, there is very little study on the applicability of these methods in camera networks. In this paper, we show how to develop methods for tracking and activity recognition in a camera network where processing is distributed across the cameras. we focus on two problems. For activity recognition, we derive a new consensus algorithm based upon the recognized activity at each camera and the transition probabilities between various activities.

In a network of agents, consensus can be defined as reaching an agreement through cooperation regarding a certain quantity of interest that depends upon the information available to measurements from all agents. An Interaction rule that specifies the information exchange between an agent and all of its neighbours in the network and the method by which the information is used, is called a consensus algorithm. Cooperation means giving consent to providing one's state and following a common protocol that serves group objective.

Distributed computing has been a challenging field in computer science for the last few decades. A lot of work has been done on consensus algorithms which formed the baseline for distributed computing. Formally the study of consensus originated in management science and statistics in 1960s. The workings on asynchronous asymptotic agreement problems in distributed decision making systems and parallel computing were the initial works in systems and control theory on a distributed network. A theoretical framework for defining and solving consensus problems for networked dynamic systems was introduced in building on the earlier work.

The rest of the paper is organized as follows: **Section I** introduce the title of this paper. **Section II** discusses

background of this title**. Section III** discusses work done on various methodologies. **Section IV** describes existing methodologies. **Section V** discusses attributes and parameters and how they affect the result. **Section VI** describes the proposed methodology. **Section VII** discusses possible outcomes and results. Finally **section VIII** Conclude this paper.

III. PREVIOUS WORK DONE

There have been a few papers in the recent past that deal with networks of video sensors.. In a distributed target tracking approach using a cluster based Kalman filter was proposed. Here, a camera is selected as a cluster head which aggregates all the measurements of a target to estimate its position using a Kalman filter and sends that estimate to a central base station. Proposed tracking system differs from this method in that each camera has a consensus-based estimate of the target's state and thus, there is no need for additional computation and communication to select a cluster head. This approach used the distributed Kalman-Consensus filter which has been shown to be more effective than other distributed Kalman filter schemes [1].

Another problem that has received some attention in this context is the development of distributed embedded smart cameras. In the existing work on the use of several cameras simultaneously for estimating the pose of a face, a single computer pulls together either all of the images captured by the cameras or the features extracted from all the images [2]. These centralized approaches to pose estimation and tracking may involve extensive comparisons of the images, as in dense-stereo reconstruction or as in the construction of active appearance models. Such approaches are not easy to implement in a distributed computing environment composed of smart cameras.

There are two major shortcomings to all methods that use a single processor for the computation of the face pose: 1) The processor creates a single point of failure and a prominent point of vulnerability in the system and perhaps even more importantly2) the number of cameras that can be connected to the processor is determined by the capabilities of the processor [3]. For those reasons, focus here is on face pose estimation and tracking algorithms that are designed specifically for a distributed implementation. The clustering protocol includes cluster propagation mechanisms that allow the computational load of face tracking to be transferred to different cameras as the target move. However, it stands to reason that simultaneously using multiple images taken from different viewpoints can only lead to more robust estimation of the pose of a face and the enhanced ability to track the face/head of a person in motion [4].

IV. EXISTING METHODOLOGIES

Particular interest has been focused on learning a network topology i.e., configuring connections between cameras and entry/exit points in their view. There has also been recent work on tracking people in a multicamera setup. A lot of work has been done on consensus algorithms which formed the baseline for distributed computing. Formally the study of consensus originated in management science and statistics in 1960s. The working on asynchronous a asymptotic agreement problems in distributed decision making systems and parallel computing were the initial works in systems and control theory on a distributed network.

The ability of a computer system to detect and track people's faces in real time will open doors to a host of new applications ranging from human-computer interaction to surveillance. However, it stands to reason that simultaneously using multiple images taken from different viewpoints can only lead to more robust estimation of the pose of a face and the enhanced ability to track the face/head of a person in motion.

There are two commonly used graphs for representing a camera network: 1) A communication graph and 2) a vision graph. In a wired camera network, since each camera can communicate with all the other cameras in the network, the communication graph is fully connected. In a wireless camera network, each camera can only communicate with cameras within its radio range. The communication graph of wireless camera networks and the vision graph of wired camera networks share many similarities.

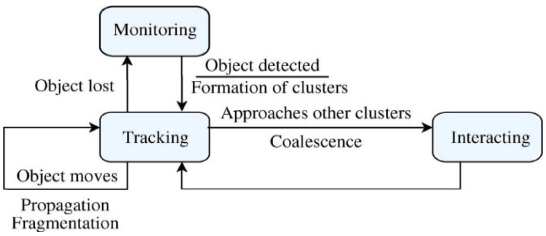

Fig.1. State transition diagram of a object tracking system using a camera network.

Fig. shows the state transition diagram of object tracking system using a camera network. Upon initialization, the network monitors the environment for any objects of interest. As objects are detected, for each object one or more clusters are formed to track it. These cluster propagate through the network to keep track of the objects in motion. Finally, if two or more clusters conclude that they are tracking the same object, they may coalesce into a larger cluster. Moreover, since the network is able to keep track of multiple objects simultaneously, each camera may belong to more than one cluster at the same time. This requires that each camera maintain a different state for each of the objects that it recognizes and that are currently being tracked by the network.

V. ANALYSIS AND DISCUSSION

To provide a quantitative evaluation of our distributed approach, we compare it to a centralized method which operates in a similar framework but does not include the clustering protocol. The centralized version has many of the features of the distributed approach. As in the distributed approach, we perform face detection locally in each camera.

The timing of both approaches is also similar. In Both systems, we synchronize the processing of images. Despite these similarities, there are fundamental differences between the centralized and distributed versions. In the centralized version, no leader is elected. Instead, collaborative processing takes place in two steps. In the first step, every camera sends the observations to a single node for central processing. In the second step, the central node processes all the observations it received for that frame in batch.

In order to provide a clear comparison of our results with ground truth, the observed sequence from each camera is temporally segmented based upon the ground truth. In practice, such a precise segmentation is not required; the observed sequence can be uniformly divided into short segments.

• Synchronization: The cameras in the network have been pre-synchronized, however, the frame synchronization may not be perfect due to slight frame rate difference between cameras. So the transmitted information between cameras includes a time stamp. In the distributed tracking framework, when a camera fuses the information from its neighboring cameras, it will do interpolation of the information vector (inAlgorithm1) as necessary. This will ensure that the information being fused is synchronized. While the activity recognition is done on each segment, unlike the frame based Kalman-consensus tracking, a precise synchronization of the cameras is not needed; precision of presynchronization is enough.

• Selection of Parameters: We can see that the consensus step in Algorithm 2 is a gradient descent algorithm that minimizes the cost function. The simplest way is to set a fixed small number, while some suggest using an adaptive step size. In our experiments, the step-size is fixed at 0.01.

VI. PROPOSED METHODOLOGY

This approach shows how to develop methods for tracking and activity recognition in a camera network where processing is distributed across the cameras. For this purpose, this approach shows, how consensus algorithms can be developed that are capable of converging to a solution, i.e., target state, based upon local decision making and exchange of these decisions among the cameras. Proposed method focused on two problems. For distributed tracking, this shows how the Kalman consensus algorithm can be adapted to camera networks taking into account issues like network topology, handoff and fault tolerance. For activity recognition, a new consensus algorithm based upon the recognized activity at each camera and the transition probabilities between various activities is derived. Experimental results and quantitative evaluation for both these methods are presented. This proposed work is a proof-of-concept study in using distributed processing algorithms for video analysis.

A special application of the Kalman-Consensus Filter is proposed to solve the problem of finding a consensus on the state vectors of multiple targets in a camera network. The situation where targets are moving on a ground plane and a homography between each camera's image plane and the ground plane is known is considered. Method shows how the state vector estimation for each target by each camera can be combined together through the consensus scheme. This method is independent of the tracking scheme employed in each camera, which may or may not be based upon the Kalman filter.

Algorithm1: Distributed Kalman-Consensus tracking algorithm performed by every at discrete time step k. The state estimate of T_l by C_i is represented with error covariance matrix.

Input: $\hat{\mathbf{x}}_i^l$ and $\mathbf{P}_i^l$ valid at k using measurements from time step $k-1$

for each T_l that is being viewed by $\{\mathcal{C}_i^n \cup C_i\}$ **do**

Obtain measurement $\mathbf{z}_i^l$ with covariance $\mathbf{R}_i^l$

Compute information vector and matrix

$$\mathbf{u}_i^l = \mathbf{F}_i^{l\,T} \left(\mathbf{R}_i^l\right)^{-1} \mathbf{z}_i^l$$
$$\mathbf{U}_i^l = \mathbf{F}_i^{l\,T} \left(\mathbf{R}_i^l\right)^{-1} \mathbf{F}_i^l$$

Send messages $\mathbf{m}_i^l = (\mathbf{u}_i^l, \mathbf{U}_i^l, \bar{\mathbf{x}}_i^l)$ to neighboring cameras $\mathcal{C}_i^n$

Receive messages $\mathbf{m}_j = (\mathbf{u}_j^l, \mathbf{U}_j^l, \bar{\mathbf{x}}_j^l)$ from all cameras $C_j \in \mathcal{C}_i^n$

Fuse information matrices and vectors

$$\mathbf{y}_i^l = \sum_{j \in (C_i \cup \mathcal{C}_i^n)} \mathbf{u}_j^l, \quad \mathbf{S}_i^l = \sum_{j \in (C_i \cup \mathcal{C}_i^n)} \mathbf{U}_j^l. \tag{5}$$

Compute the Kalman-Consensus state estimate

$$\mathbf{M}_i^l = \left((\mathbf{P}_i^l)^{-1} + \mathbf{S}_i^l\right)^{-1}$$
$$\hat{\mathbf{x}}_i^l = \bar{\mathbf{x}}_i^l + \mathbf{M}_i^l \left(\mathbf{y}_i^l - \mathbf{S}_i^l \bar{\mathbf{x}}_i^l\right) + \gamma \mathbf{M}_i^l \sum_{j \in \mathcal{C}_i^n} \left(\bar{\mathbf{x}}_j^l - \bar{\mathbf{x}}_i^l\right)$$
$$\gamma = 1 / \left(\|\mathbf{M}_i^l\| + 1\right), \quad \|\mathbf{X}\| = \left(tr(\mathbf{X}^T \mathbf{X})\right)^{\frac{1}{2}}. \tag{6}$$

Propagate the state and error covariance matrix from time step k to $k+1$

$$\mathbf{P}_i^l \leftarrow \mathbf{A}^l \mathbf{M}_i^l \mathbf{A}^{l\,T} + \mathbf{B}^l \mathbf{Q}^l \mathbf{B}^{l\,T}$$
$$\bar{\mathbf{x}}_i^l \leftarrow \mathbf{A}^l \hat{\mathbf{x}}_i^l. \tag{7}$$

end for

Algorithm2: Distributed Consensus based activity recognition algorithm performed by every C_i at step k.

Input: $\mathbf{w}_i(k-1)$

for each person that is being viewed by $\{\mathcal{C}_i^n \cup C_i\}$ **do**

Obtain observations $O_i(k)$

Compute local likelihood

$$\mathbf{v}_i(k) = \begin{bmatrix} v_i^1(k) \\ \vdots \\ v_i^Y(k) \end{bmatrix} = \begin{bmatrix} P\left(O_i(k) | y(k) = 1\right) \\ \vdots \\ P\left(O_i(k) | y(k) = Y\right) \end{bmatrix}$$

Send $\mathbf{v}_i(k)$ to neighboring cameras $\mathcal{C}_i^n$

Receive $\mathbf{v}_j(k)$ from all cameras $C_j \in \mathcal{C}_i^n$

Fuse information to estimate activity state see equation at the bottom of next page, where $\mathbf{M}$ is a $Y \times Y$ matrix with (i,j)th element to be $m(i,j)$

$$\Lambda\left(\mathbf{v}_j(k)\right) = \begin{bmatrix} v_j^1(k) & & \\ & \ddots & \\ & & v_j^Y(k) \end{bmatrix}$$

and $\mathbf{1}_Y = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ with Y elements.

repeat

Send $\mathbf{w}_i(k)$ to neighboring cameras $\mathcal{C}_i^n$

Receive $\mathbf{w}_j(k)$ from all cameras $C_j \in \mathcal{C}_i^n$

Compute the Consensus state estimate

$$\mathbf{w}_i(k) = \mathbf{w}_i(k) + \epsilon \sum_{j \in \mathcal{C}_i^n} \left(\mathbf{w}_j(k) - \mathbf{w}_i(k)\right)$$

until either a predefined iteration number is reached or $\sum_{j \in \mathcal{C}_i^n} \left(\mathbf{w}_j(k) - \mathbf{w}_i(k)\right)$ is smaller than a predefined small value

end for

VII. POSSIBLE OUTCOMES AND RESULTS

This approach is tested for tracking in a real camera network composed of 10 PTZ cameras looking over an outdoor area of approximately 10000 sq. feet. In the area under surveillance, there were eight targets in total that were to be tracked using this distributed Kalman-Consensus filtering approach. In this experiment, the measurements are obtained using histogram of

gradient (HOG) human detector. The association of measurements to targets is achieved based upon appearance (color) and motion information.. The results are shown on a nonstatic camera network. The cameras are controlled to always cover the entire area under surveillance through a game theoretic control framework. The change of camera settings does not affect the procedure of the Kalman-consensus filter. As the targets are observed in this area, the single-view tracking module in each camera determines the ground plane position of each target in its FOV and sends that information to the Kalman-Consensus filter which processes it together with the information received from the Kalman-Consensus filters of neighboring cameras.

Fig.2 (a) shows the distributed Kalman-Consensus tracks for the eight targets. The measurements of the different cameras are shown in a light gray color. As can be seen, the Kalman Consensus filter in each camera comes to a smooth estimate of the actual state for each target. Fig.2 (b) shows the distributed tracking results on the ground plane for one of the targets, T_5 . The dots correspond to the ground plane measurements from different cameras viewing the target while the solid line is the consensus-based estimate. As can be expected, the individual positions are different for each camera due to calibration and single-view tracking inaccuracies. As can be seen clearly, even though C_5^v is time varying, the Kalman-Consensus filter estimates the target's position seamlessly at all times.

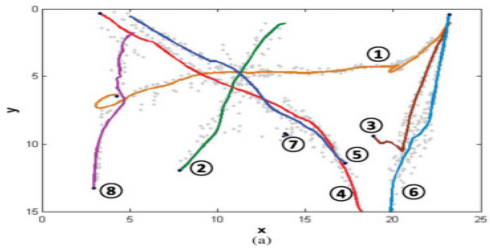

Fig. 2(a) Distributed Kalman-Consensus tracking trajectories for 8 targets

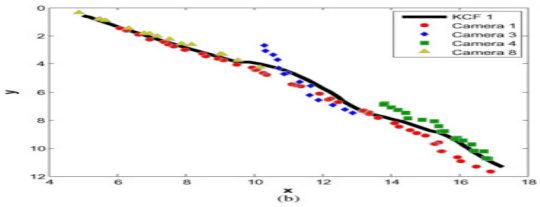

Fig. 2(b) Tracking results on the ground plane for one of the targets T_5.

VIII. CONCLUSION

In this distributed scene analysis algorithms are investigated by leveraging upon concepts of consensus. Two fundamental tasks are addressed—tracking and activity recognition in a distributed camera network. A robust approach to distributed multitarget tracking in a network of cameras is proposed. A distributed Kalman-Consensus filtering approach was used together with a dynamic network topology for persistently tracking multiple targets across several camera views. Each camera individually computes the world pose faces based upon their visual features. The observations of multiple cameras are integrated using a minimum variance estimator and tracked using a Kalman filter. A probabilistic consensus scheme for activity recognition was provided, which combines the similarity scores of neighboring cameras to come up with a probability for each action at the network level.

REFERENCES

[1] Josiah Yoder, Henry Medeiros, Johnny Park, and Avinash C. Kak, "Cluster-Based Distributed Face Tracking in Camera Networks", IEEE Transactions On Circuits And Systems, Vol. 19, no.10, pp 2551-2563, October 2010.

[2] Bi Song, Ahmed T. Kamal, Cristian Soto, Chong Ding, Jay A. Farrell and Amit K. Roy Chowdhury, "Tracking and Activity Recognition Through Consensus in Distributed Camera Networks", IEEE Transactions On Image Processing, vol.19, no. 10, pp 2564-2579, October 2010.

[3] Le An, Mehran Kafai, Student Member, IEEE, and Bir Bhanu, Fellow, IEEE, "Dynamic Bayesian Network for Unconstrained Face Recognition in Surveillance Camera Networks", IEEE Journal On Emerging And Selected Topics In Circuits And System, vol.3, no. 2, pp 155-164, June 2013.

[4] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in Proc. American Control Conf., pp. 3157–3162, Jun. 2008.