BELIEF CONSENSUS FOR DISTRIBUTED ACTION RECOGNITION

Ahmed Tashrif Kamal, Bi Song and Amit K. Roy-Chowdhury

Department of Electrical Engineering, University of California, Riverside

ABSTRACT

In this work, we consider a camera network where processing is distributed across the cameras. Our goal is to recognize actions of multiple targets consistently observed over the entire network. To obtain consistent and better results we need to properly fuse the action scores from multiple cameras. There have been multiple works on distributed tracking and distributed data association for multiple targets in a camera network. We can use the data association results and tracking confidence scores to improve the action recognition results. We propose a consensus based framework for solving this problem in an integrated manner and with a completely distributed camera network architecture. We propose a novel method for weighting the action scores based on tracking confidences and show how the cameras can reach a consensus about the action of a target using *belief consensus*. We show real life experiments and performance metrics with multiple cameras and targets.

Index Terms— belief consensus, action recognition, distributed

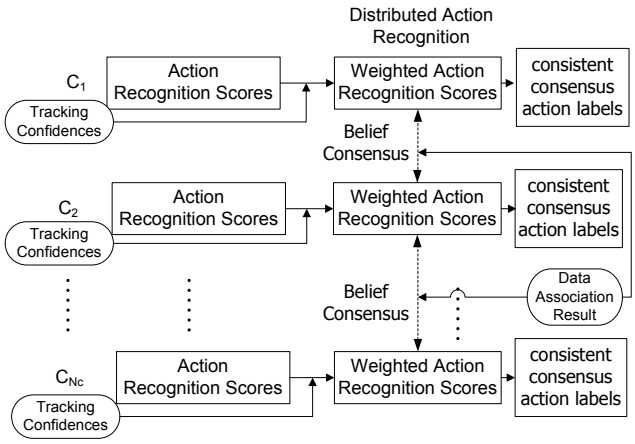

Fig. 1. Overall system diagram depicting our proposed framework for multi-target action recognition in distributed camera networks. The data in the ellipses are generated using the JPDA-KCF algorithm in [1] and utilized in our framework. The bidirectional arrows represent the information exchange among the cameras.

1. INTRODUCTION

Recently, as multi-camera installations are gaining popularity, the use of the information from multiple cameras in the decision making process can help us push the limits of automated video analysis further. However, this calls for the development of methods for analysis of the data coming from multiple video feeds. In many applications, a distributed network architecture is necessitated whereby video is analyzed in a distributed manner over the entire network rather than at a central server. An example could be a wireless network with limited bandwidth, but which is easy to install and can be mobile. In this paper, we consider such distributed camera networks and propose a consensus-based framework that is capable of performing action recognition.

In such multi-camera setups, there will be multiple targets and a single camera may only be capturing a portion of the area covered by the entire camera network, i.e., a camera only observes a subset of all the targets. In such scenarios, knowledge of the cross-camera target correspondence (data association) and tracks are essential for the correct target-specific analysis. Thus, our overall goal of distributed action recognition in the camera network requires feedback from the lower level tasks of distributed data association and tracking. However, our framework is generally applicable to other distributed estimation tasks as face/gait/pose estimation.

There has been recent work on distributed data association and tracking [1, 2]. In [1] a JPDA-KCF framework was proposed where the *Joint Probabilistic Data Association (JPDA)* step was used to solve the data association and a *Kalman Consensus Filter (KCF)* was used to track targets. In our framework, we use the tracking confidence scores obtained by the JPDA-KCF algorithm, to weight our single camera action recognition scores. Next, we incorporate a distributed information fusion approach called *Belief Consensus* [3] to fuse the weighted single camera action recognition scores. The data association results from the JPDA-KCF algorithm were used to fuse scores of appropriate targets together. The flowchart of our entire framework is given in Fig 1.

This work was partially supported under ONR grant N00014-09-1-0666 and NSF grant IIS-0905671.

Related Work

Several approaches are available for distributed data association in the multi-agent systems literature [4, 1, 2]. Some of the recent works on distributed tracking can be found in [1, 5, 6, 7]. In our work, we utilize the distributed tracking and data association scheme from [1].

In a distributed multi-agent system, different agents may propose different description of an observed target, introducing inconsistency in the network. To maintain consistency, there has to be a protocol running in each agent that makes all the agents in the network to reach a *consensus*. The *consensus* they try to reach is usually a function of their initial proposals. It may be the average [8] or the product [3] of the initial proposals. As our fusion schemes needs to take the product of the initial proposals (see Sec. 2.1 for details), we use the latter consensus algorithm which is also known as *Belief Consensus* [3]. A detailed review of consensus can be found in [9]. There have been a number of interesting papers in computer vision in the recent past that deal with consensus approaches [6, 10].

A review on multi-camera action recognition can be found in [11]. As can be seen from there, most of the methods on activity recognition using multiple cameras are centralized schemes. In [6], a distributed action recognition scheme was proposed, but unlike our approach, the final result depends strongly on the network topology. In [12, 13, 14], different frameworks were offered to cluster similar actions together. However, our goal is to make use of the data association and tracking results to improve action recognition results while maintaining consistency throughout the network. In [15], the authors proposed a method whereby after solving the data association, they output the best result among all the cameras. In real-life scenario, as one camera can only have a partial view of a target, the fusion of the single camera recognition scores may be better than the best result among the cameras, which is a motivation for proposing the consensus approach.

2. DETAILED METHODOLOGY

In a network of N_C cameras, let O_i^j be the observation from camera C_i which is associated to target T_j and $\mathcal{O}_j$ be the collection of all observations associated with T_j. The targets are tracked and associated using the JPDA-KCF scheme in [1]. The necessity of calibration depends absolutely on the tracking and data association scheme. For our work the JPDA-KCF scheme was implemented in a calibrated scenario. Let F_j be a vector of length N_C where $F_j(i)$ is the association strength of O_i^j with T_j (see Sec 2.2). Also, let y_j be the variable of action classes for target T_j. Each camera processes the tracked video of an observation for a few frames[1] and computes the probabilities for each action class which can be denoted as $P(O_i^j|y_j)$.

2.1. Information Fusion

Our goal is to compute $P(y_j|\mathcal{O}_j, F_j)$, the posterior probability of the action classes, given all the observations associated with T_j and the association strength vector F_j. Using Bayes' law we can write,

$$\begin{aligned} P(y_j|\mathcal{O}_j, F_j) &= \frac{P(\mathcal{O}_j|y_j, F_j)P(y_j|F_j)}{P(\mathcal{O}_j|F_j)} \\ &= \gamma P(\mathcal{O}_j|y_j, F_j) \\ &= \gamma P(O_1^j, O_2^j, ..., O_{N_C}^j|y_j, F_j) : \exists O_i^j \\ &= \gamma \prod_{\forall i:\exists O_i^j} P(O_i^j|y_j, F_j) \end{aligned} \quad (1)$$

The second step comes from the assumption of uniform[2] prior distribution $P(y_j|F_j)$, over the action classes, and also from the fact that $P(\mathcal{O}_j|F_j)$ is constant[3] for all action class. Thus, we can combine these two factors together into a normalizing constant γ. The fourth step comes from the independence condition [4] of O_i^j.

Having (1), our next step would be to weight the likelihoods using tracking confidence scores and then seek a distributed implementation for fusion.

2.2. Weighting Single Camera Recognition Scores with Tracking Confidences

In reality, there will be noise in image features, resulting in noisy tracks and data associations. The tracking module in [1] gives confidence scores in the tracks (the covariance matrix). Intuitively, the less the confidence of a track, the less certain the fusion result should be for an action label and should tend more towards uniform distribution over the action labels. In addition to that, the further an associated observation is from the mean of the track on the ground plane, the less it should contribute in the fusion. We can incorporate both these ideas by choosing the *Mahalanobis distance* as our distance metric. Let $\mathbf{x}_i^j$ be the position vector of an observation on the ground plane, $\mathbf{p}_j$ be the estimated position sub-vector and $\mathbf{E}_j$ be the position covariance sub-matrix for T_j acquired from the distributed tracking step in [1]. Thus, the Mahalanobis distance between O_i^j and T_j is:

[1] Finding the optimum size and placement of the time window for action recognition is still a very challenging problem. We do not focus on this issue; rather we divide the entire video into windows of the same predefined size.

[2] The prior does not necessarily have to be uniform, because the action recognition scores from the previous time window can be propagated through this prior using learnt transition probabilities between different action classes [16].

[3] As $P(\mathcal{O}_j|F_j)$ does not have y_j in its argument, it is constant for all action labels. Thus, for a specific F_j, $P(\mathcal{O}_j|F_j)$ is a constant scalar term.

[4] When action recognition features are extracted properly from raw observations, the effect of factors like appearance, shape, motion etc., other than action class, is eliminated. Thus, given the action class, the features (which we are calling observations in our context) become independent of each other.

$$D(O_i^j, T_j) = \sqrt{(\mathbf{x}_i^j - \mathbf{p}_j)^T \mathbf{E}_j^{-1} (\mathbf{x}_i^j - \mathbf{p}_j)} \quad (2)$$

Based on this distance measure, we can compute our association strength vector F_j using the following formula:

$$F_j(i) = \begin{cases} e^{-\alpha D(O_i^j, T_j)} & \text{if } \exists\ O_i^j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Here α is an empirically set parameter which controls the fall-off of the exponent. Next, we weight $P(O_i^j|y_j)$ with $F_j(i)$ to get $P(O_i^j|y_j, F_j)$ by using weighted mixture of distributions as,

$$P(O_i^j|y_j, F_j) = F_j(i) P(O_i^j|y_j) + (1 - F_j(i)) \frac{\overrightarrow{1}}{L} \quad (4)$$

Here L is the number of trained action classes, thus $\frac{\overrightarrow{1}}{L}$ is actually a uniform distribution over the action classes.

2.3. Distributed Implementation through Belief Consensus

Belief consensus states that, in a network of N agents, if each agent C_i has an initial proposal π_i, they can asymptotically reach the consensus of $(\prod_i \pi_i)^{\frac{1}{N}}$ (the geometric mean of the initial proposal), by iteratively updating their own proposals with the proposals of their neighbors using the formula:

$$\pi_i(k+1) = \pi_i(k) \left(\prod_{j \epsilon \mathcal{N}_{C_i}} \frac{\pi_j(k)}{\pi_i(k)} \right)^{\lambda} \quad (5)$$

Here, $\mathcal{N}_{C_i}$ is the set of the neighboring cameras of C_i and k is the iteration number. The proof of this formula and the analysis of the convergence speed parameter λ can be found in [3]. Thus, we have

$$\lim_{k \to \infty} \pi_i(k) = \left(\prod_i \pi_i(0) \right)^{\frac{1}{N}} \quad (6)$$

So, if all the nodes in the network know the number of cameras (N_{T_j})[5] observing target T_j, after some iterations, they can reach to a consensus about the action score of T_j. For each target T_j, each camera has to run a separate consensus scheme. We set the initial proposals of C_i for T_j as,

$$\pi_i^j(0) = \begin{cases} P(O_i^j|y_j, F_j) & \text{if } F_j(i) \neq 0 \\ \overrightarrow{1} & \text{otherwise} \end{cases} \quad (7)$$

As $F_j(i) = 0$ means there are no observations at C_i associated with T_j; for such a camera, setting the initial proposal to $\overrightarrow{1}$ will enable the product consensus to run unaffected by the fact of a camera not observing a target. After convergence, we have to take the ${N_{T_j}}^{th}$ power of $\pi_i^j(k)$ and normalize to get the actual consensus result for target T_j as necessitated by (6) and (1). Thus, all the cameras reach to a consensus on the action scores of each of the target present in the network.

[5] N_{T_j} can be computed along with the data association and tracking by enumeration. N_{T_j} will change only when the data association changes. Thus when a camera detects a change in its own data association, it can send an update message throughout the entire network of its new association and thus each camera will know N_{T_j} for all the targets.

	sit	walk	pick	hshk	hug	wave
Best Cam	0.49	0.29	0.53	0.49	0.64	0.77
Method in [6]	0.49	0.33	0.53	0.53	0.59	0.70
Our Method	0.67	0.42	0.73	0.64	0.84	0.87

Table 1. Each row shows the probability of correct match for different actions using a particular method. The first row holds the statistics for the case where the single camera with the highest probability of correct match was selected using ground truth.

3. EXPERIMENTAL EVALUATION

To validate our proposed approach for multi-camera action recognition, we carried out experiments on UCR multi-camera data-set [17]. UCR data-set is comprised of 2.5 hours of multi-view videos. 4-8 cameras are used to cover an area where 2-10 people are performing various real-life actions. We used 10 minutes of this data-set for training and another 30 minutes for testing.

The cameras were calibrated and the targets were detected using person detectors. Then they were tracked and associated using the JPDA-KCF method in [1]. For the simplicity of our experiment, only the video clips with actions performed according to our trained labels were used. We used the Bag-of-Words approach on top of Spatio-Temporal Interest Point (STIP) features and used Support Vector Machines for classification. Our single-view action recognition scheme is similar to [18]. We trained our system for six different action classes i.e. 1 - Sit, 2 - Walk, 3 - Pick, 4 - Handshake, 5 - Hug and 6 - Hand-wave. Next, we assume a loosely connected network topology, where a camera is connected to at most two other cameras. The more connected the network gets, the faster the consensus algorithm converges. After getting the single camera recognition scores, we weight them according to our scheme in Sec 2.2. Next, using our consensus scheme in Sec 2.3, we fused the action scores for each of the targets in each of the cameras.

The results achieved in different iterations of the consensus scheme is shown in Fig 2. We assumed convergence after 20 iterations. We show the statistical performance of our method and compare it with the average consensus scheme of [6] in Table 1. From the table, it is apparent that our method performed well. In the experiments, 88% of the time the most probable action label in the fusion result was the actual ground truth action, whereas for single cameras, it was only 30% of the time.

In the context of comparing our work against centralized multi-camera activity recognition schemes, we note that the belief consensus approach is guaranteed to converge to the posterior estimate of (1) as proved in [3], which is essentially computing the action label probabilities given the observations. Thus, given a feature set which leads to a certain performance in the centralized case, we show how to implement

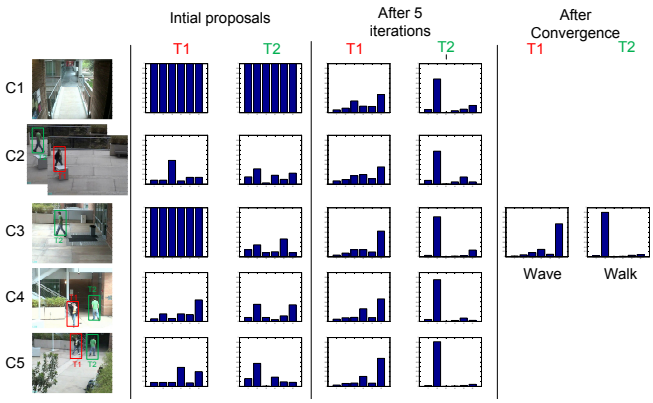

Fig. 2. In this figure, we show an example from our experiment. Each row in this figure represents each of the five cameras in the network and two people are tracked throughout the entire network. The first column represents the video feed from different cameras. The next two columns show the initial proposals of each cameras for each targets. These are the weighted single camera action recognition scores. As C_1 did not observe any of the targets, it started with a vectors of ones. The next two columns show the updated proposals after 5 iterations of belief consensus. Note how the cameras converge towards the same decision for each target. The last two columns shows the actual score computed by the fusion equation (1).

it in the decentralized situation without any degradation in performance.

4. CONCLUSIONS

In this paper, we investigated the utilization of multi-target tracking and data association to improve action recognition results in a distributed framework. While there has been recent work on consensus approaches for various computer vision problems, our work considered the integration of data association and tracking to achieve a higher level goal (action recognition) in a distributed framework using the belief consensus algorithm. We showed real life experiments and performance metrics with multiple cameras and targets. One of the research directions that remain to be explored is the integration of auto-calibration of the camera network in dynamic environments.

5. REFERENCES

[1] N.F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *Decision and Control, 47th IEEE Conference on*, 2008.

[2] L. Chen, M. Cetin, and A.S. Willsky, "Distributed data association for multi-target tracking in sensor networks," in *Proc. Int'l Conf. Information Fusion*, 2005.

[3] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," in *Network Embedded Sensing and Control*, 2006.

[4] S. Calderara, A. Prati, and R. Cucchiara, "Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance," *Computer Vision and Image Understanding*, 2008.

[5] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," *44th IEEE Conference on Decision and Control*, 2005.

[6] B. Song, A. Kamal, C. Soto, C. Ding, A. Roy Chowdhury, and J. Farrell, "Tracking and activity recognition through consensus in distributed camera networks," *Image Processing, IEEE Transactions on*, 2010.

[7] C. Soto, B. Song, and A. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[8] R. Olfati-Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference. Proceedings of the*, 2003.

[9] R. Olfati-saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE. 2007*.

[10] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on SE(3)," *IEEE/ACM Intl. Conf. on Distributed Smart Cameras*, 2008.

[11] A. Sankaranarayanan, R. Patro, P. Turaga, A. Varshney, and R. Chellappa, "Modeling and visualization of human activities for multicamera networks," *Journal on Image and Video Processing*, vol. 2009.

[12] X. Wang, K. Tieu, and E. Grimson, "Correspondence-free activity analysis and scene modeling in multiple camera views," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010.

[13] C. Change Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2009.

[14] E. Zelniker, S. Gong, and T. Xiang, "Global abnormal behaviour detection using a network of cctv cameras," in *IEEE International Workshop on Visual Surveillance*, 2008.

[15] S. Calderara, A. Prati, and R. Cucchiara, "A markerless approach for consistent action recognition in a multi-camera system," in *Distributed Smart Cameras. ACM/IEEE International Conference on*, 2008.

[16] J. Rittscher and A. Black, "Classification of human body motion," in *IEEE Intl. Conf. on Computer Vision*, 1999.

[17] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda, "Videoweb dataset for multi-camera activities and non-verbal communication," in *Distributed Video Sensor Networks, Springer 2010*.

[18] J. Liu and M. Shah, "Learning human actions via information maximization," in *Computer Vision and Pattern Recognition, 2008*.