A Sparsity-Driven Approach to Multi-camera Tracking in Visual Sensor Networks

Serhan Coşar[a,b]
[a]INRIA Sophia Antipolis, STARS team
2004 R. des Lucioles, 06902 S. Antipolis, France
serhan.cosar@inria.fr

Müjdat Çetin[b]
[b]Faculty of Engineering and Natural Sciences
Sabancı University, 34956 Istanbul, TURKEY
{serhancosar,mcetin}@sabanciuniv.edu

Abstract

In this paper, a sparsity-driven approach is presented for multi-camera tracking in visual sensor networks (VSNs). VSNs consist of image sensors, embedded processors and wireless transceivers which are powered by batteries. Since the energy and bandwidth resources are limited, setting up a tracking system in VSNs is a challenging problem. Motivated by the goal of tracking in a bandwidth-constrained environment , we present a sparsity-driven method to compress the features extracted by the camera nodes, which are then transmitted across the network for distributed inference. We have designed special overcomplete dictionaries that match the structure of the features, leading to very parsimonious yet accurate representations. We have tested our method in indoor and outdoor people tracking scenarios. Our experimental results demonstrate how our approach leads to communication savings without significant loss in tracking performance.

1. Introduction

Over the past decade, large-scale camera networks have been a topic on increasing interest in security and surveillance. With the developments in wireless sensor networks and the availability of inexpensive imaging sensors, a new field has emerged: Visual sensor networks (VSNs), i.e., networks of wirelessly interconnected battery-operated devices that acquire video data. Using a camera in a wireless network leads to unique and challenging problems that are more complex than the traditional multi-camera video analysis systems and wireless sensor networks might have.

To minimize the amount of data to be communicated, in some methods simple features are used for communication. For instance, 2D trajectories are used in [10]. In [4], 3D trajectories together with color histograms are used. Hue histograms along with 2D position are used in [9]. These approaches may be capable of decreasing the communication, but they are not capable of maintaining robustness. Moreover, there are decentralized approaches in which cameras are grouped into clusters and tracking is performed by local cluster fusion nodes. For a nonoverlapping camera setup, tracking is performed by maximizing the similarity between the observed features from each camera and minimizing the long-term variation in appearance using graph matching at the fusion node [12]. For an overlapping camera setup, a cluster-based Kalman filter in a network of wireless cameras is proposed in [8, 15]. Local measurements of the target acquired by members of the cluster are sent to the fusion node. Then, the fusion node estimates the target position via an extended Kalman filter. Depending on the size of image features and the number of cameras in the network, even collecting features to the fusion node may become expensive for the network. In such cases, further approximations on features are necessary.

Compressed sensing and sparse representation have become important signal recovery techniques because of their success in various application areas [3, 7, 11]. In this paper, we propose a sparsity-driven tracking method that is suitable for energy and bandwidth constraints in VSNs. As in [2], our method is a decentralized tracking approach in which each camera node in the network performs feature extraction by itself and obtains image features (likelihood functions). Instead of directly sending likelihood functions to the fusion node, we find the sparse representation of the likelihoods. Rather than a block-based likelihood compression scheme as in [2], here we have designed special overcomplete dictionaries that are matched to the structure of the likelihood functions and used these dictionaries for sparse representation of likelihoods. The main contribution of this work is building a sparse representation framework and designing overcomplete dictionaries that are matched with the structure of likelihoods. In particular

our dictionaries are designed by exploiting the specific known geometry of the measurement scenario and by focusing on the problem of human tracking. Each element in the dictionary for each camera corresponds to the likelihood that would result from a single human at a particular location in the scene. Hence actual likelihoods extracted from real observations from scenes containing multiple individuals can be very sparsely represented in our approach. By using these dictionaries, we can represent likelihoods with few coefficients, and thereby decrease the communication between cameras and fusion nodes. To the best of our knowledge, a sparse representation based compression of likelihood functions computed in the context of tracking in a VSN has not been proposed in previous work. We have used our method within the context of a well-known multi-camera human tracking algorithm [5]. To this end, we have modified the method in [5] to obtain a decentralized tracking algorithm. We have tested the performance of our approach in indoor and outdoor tracking problems. Our experimental results show that our approach can achieve very good tracking performance by consuming very little communication bandwidth as compared to existing methods.

Section 2 presents the decentralized approach for multi-camera tracking. In Section 3, our sparse representation framework and specially-designed overcomplete dictionaries are presented. Experimental setup and the results are given in Section 4. Finally, we draw conclusion in Section 5.

2. Decentralized Tracking

2.1. Overview

In a traditional setup of camera networks, each camera acquires an image, sends this raw data to a central unit and in the central unit, relevant features are extracted, combined, finally, the positions of the humans are estimated. The presence of a single global fusion center leads to high data-transfer rates and the need for a computationally powerful machine, thereby, to a lack of scalability and energy efficiency. Compressing raw image data may decrease the communication in the network, but since the quality of images drops, it might also decrease the tracking performance. For this reason, centralized trackers are not very appropriate for use in VSN environments. In decentralized tracking, cameras are grouped into clusters and nodes communicate with their local cluster fusion nodes only [13]. Communication is reduced by limiting the cooperation within each cluster and among fusion nodes. After acquiring the images, each camera extracts useful features from the images it has observed and sends these features to the local fusion node. Using the multi-view image features, tracking is performed in the local fusion node.

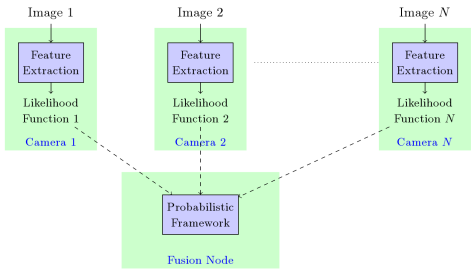


Figure 1. The flow diagram of a decentralized tracker using a probabilistic framework.

Modeling the dynamics of humans in a probabilistic framework is a common perspective of many multi-camera human tracking methods [5, 14, 6]. For each camera, a likelihood function is defined in terms of the observations obtained from its field of view. In centralized tracking, the likelihood functions are computed after collecting the image data at the central unit. For a decentralized approach, since each camera node extracts local features, these likelihood functions can be evaluated at the camera nodes and they can be sent to the fusion node. Then, in the fusion node the likelihoods can be combined and tracking can be performed in the probabilistic framework. A flow diagram of the decentralized approach is illustrated in Figure 1.

2.2. Multi-Camera Tracking Algorithm

In this section we describe the tracking method of [5] , as we apply our proposed approach within in the context of this method in this paper. In [5], the ground plane is discretized into a finite number G of regularly spaced 2D locations. Let $\mathbf{L}_t = (L_t^1, ..., L_t^{N^*})$ be the locations of individuals at time t, where N^* is the maximum number of individuals. Given T temporal frames from C cameras, $\mathbf{I}_t = (I_t^1, ..., I_t^C)$, the goal is to estimate the trajectory of person n, $\mathbf{L}^n = (L_1^n, ..., L_T^n)$, by seeking the maximum of the probability of both the observations and the trajectory ending up at location k at time t:

$$\Phi_t(k) = \max_{l_1^n, ..., l_{t-1}^n} P(\mathbf{I}_1, L_1^n = l_1^n, ..., \mathbf{I}_t, L_t^n = k) \quad (1)$$

Under a hidden Markov model, the above expression turns into the classical recursive expression:

$$\Phi_t(k) = \underbrace{P(\mathbf{I}_t | L_t^n = k)}_{Appearance\ model} \max_{\tau} \underbrace{P(L_t^n = k | L_{t-1}^n = \tau)}_{Motion\ model} \Phi_{t-1}(\tau) \quad (2)$$

The motion model is a distribution into a disc of limited radius and center τ, which corresponds to a loose bound on the maximum speed of a walking human.

From the input images $\mathbf{I}_t$, by using background subtraction, foreground binary masks, $\mathbf{B}_t$, are obtained. Let the colors of the pixels inside the blobs are denoted as $\mathbf{T}_t$ and X_k^t be a Boolean random variable denoting the presence of an individual at location k of the grid at time t. It is shown in [5] that the appearance model in Eq. 2 can be decomposed as:

$$P(\mathbf{I}_t|L_t^n = k) \propto \underbrace{P(L_t^n = k|X_k^t = 1, \mathbf{T}_t)}_{Color\ model} \quad \underbrace{P(X_k^t = 1|\mathbf{B}_t)}_{Ground\ plane\ occup.} \quad (3)$$

In [5], humans are represented as simple rectangles and these rectangles are used to create synthetic ideal images that would be observed if people were at given locations. Within this model, the ground plane occupancy is approximated by measuring the similarity between ideal images and foreground binary masks.

Let $T_t^c(k)$ denote the color of the pixels taken at the intersection of the foreground binary mask, B_t^c, from camera c at time t and the rectangle A_k^c corresponding to location k in that same field of view. Say we have the color distributions of the N^* individuals present in the scene, $\mu_1^c, ..., \mu_{N^*}^c$. The color model of person n in Eq. 3 can be expressed as:

$$P(L_t^n = k|X_k^t = 1, \mathbf{T}_t) \propto P(\mathbf{T}_t|L_t^n = k) = P(T_t^1(k), ..., T_t^C(k)|L_t^n = k) = \prod_{c=1}^{C} P(T_t^c(k)|L_t^n = k) \quad (4)$$

Different from [5], we represent $P(T_t^c(k)|L_t^n = k)$ by comparing the estimated color distribution (histogram) of the pixels in $T_t^c(k)$ and the color distribution μ_n^c with the Bhattacharya coefficient between two distributions. By performing a global search with dynamic programming using Eq. 2, the trajectory of each person can be estimated.

2.3. Decentralized Tracking Algorithm

From the above formulation, we can see that there are two different likelihood functions defined in the method. One is the ground plane occupancy map (GOM), $P(X_k^t = 1|\mathbf{B}_t)$, approximated using the foreground binary masks. The other is the ground plane color map (GCM), $P(L_t^n = k|X_k^t = 1, \mathbf{T}_t)$, which is a multi-view color likelihood function defined for each person individually. This map is obtained by combining the individual color maps, $P(T_t^c(k)|L_t^n = k)$, evaluated using the images each camera acquired. Since foreground binary masks are binary images that can be easily compressed by a lossless compression method, they can be directly sent to the fusion node without overloading the network. Therefore, as in the original method GOM is evaluated at the fusion node. In our framework, we evaluate GCM in a decentralized way (as presented in Figure 1): At each camera node ($c = 1, \cdots, C$), the local color likelihood function for the person of interest ($P(T_t^c(k)|L_t^n = k)$) is evaluated. Then, these likelihood functions are sent to the fusion node. At the fusion node, these likelihood functions are integrated to obtain the multi-view color likelihood function (GCM) (Eq. 4). By combining GCM and GOM with the motion model, the trajectory of the person of interest is estimated at the fusion node using dynamic programming (Eq. 2). The whole process is run for each person in the scene.

3. Sparse Representation of Likelihoods

3.1. Overview

The bandwidth required for sending likelihood functions depends on the size (i.e., the number of "pixels" in a 2D likelihood function) and the number of cameras in the network. To make the communication in the network feasible, in [2] a block-based compression scheme that use uses orthogonal transforms including the discrete cosine transform (DCT) and several wavelet transforms is followed. In that approach, likelihood functions are split into blocks and each block is transformed to DCT domain. Then, by taking only the significant coefficients, the likelihood functions are compressed and this new representation is sent to the fusion node. Here we propose using custom-designed overcomplete dictionaries for sparse representation of likelihood functions. At each camera node we propose to represent the likelihood functions sparsely in a proper basis and then send this representation instead of sending the function itself. In this way, we reduce the communication in the network. Mathematically, we have the following linear system:

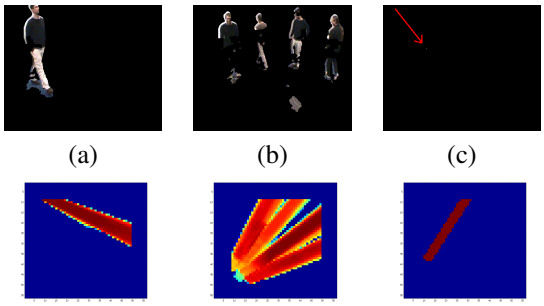
$$y_c = A_c \cdot x_c \quad (5)$$

where y_c and x_c represents the likelihood function of the camera c (for a person of interest in a particular time instant, $P(T_t^c(k)|L_t^n = k)$ in Eq. 4) and its sparse coefficients, respectively, and A_c is the overcomplete[1] dictionary matrix for camera c that represents the domain in which y_c has a sparse representation. To obtain the sparse representation of the likelihood function, at each camera we solve the optimization problem in Eq. 6.

$$\min_{x_c} ||y_c - A_c \cdot x_c||_2 + \lambda||x_c||_1 \quad (6)$$

Notice that in our sparse representation framework, we do not require the use of specific image features or likelihood functions. The only requirement is that the tracking method should be based on a probabilistic framework, which is a common approach for modeling the dynamics of humans. Hence, our method is a generic framework that can be used with many probabilistic tracking algorithms in a VSN environment. At the fusion node, likelihood functions of each

[1] The number of columns is bigger than the number of rows



(a) (b) (c) (d) (e) (f)

Figure 2. (a,b,c) Foreground images and (d,e,f) corresponding color model likelihood functions

camera can be reconstructed simply by multiplying the new representation with the matrix A_c.

3.2. Designing Overcomplete Dictionaries

The approach in [2] achieves likelihood compression through orthogonal transforms of blocks of the likelihood functions. Although such transforms provide some level of sparsity, they certainly do not fully exploit the structure of the likelihood functions. For computational reasons, these orthogonal transforms are applied to blocks of the likelihood functions, leading to blocking artifacts in the reconstructed likelihoods. Focusing on scenarios involving extreme bandwidth constraints, in this paper we propose designing overcomplete dictionaries that are matched to the structures of the full likelihood functions.

The likelihood functions we obtain from the color model in [5] have a special structure. Figure 2-a and Figure 2-b show foreground images captured from two different camera views when there is only one person in the scene and when the scene is crowded, respectively. The likelihood function obtained from these image are given in Figure 2-d and Figure 2-e, respectively. We observe that likelihood functions consist of quadrilateral-shaped components each of which is generated by a person in the scene. One of the important properties of these components is that their shapes do not depend on the value of the foreground pixels. These shapes depend only on the camera view and the position of the foreground pixels. For this reason, we can say that these quadrilateral-shaped components serve as building blocks of likelihood functions in a multi-person tracking scenario. By creating a dictionary from these building blocks, we can naturally and properly utilize the structure of the likelihood functions.

In order to find all the building blocks of likelihood functions, we need to create likelihood functions from all the possible foreground images. We start by a foreground image that is all-black except a single white pixel, i.e. a building block of foreground images (pointed with a red arrow in Figure 2-c), and obtain a likelihood function from this image (Figure 2-f). By changing the position of the white pixel and obtaining the likelihood function from that foreground image, we can create a pool composed of building blocks. For each camera, we create the dictionary matrix (A_c in Eq. 5) by arranging the building blocks of likelihoods as the columns of the matrix.

4. Experimental Results

4.1. Setup

In the experiments, we have simulated the VSN environment by using the indoor and outdoor multi-camera datasets in [5]. The indoor dataset includes four people sequentially entering a room and walking around. The sequence was shot by four synchronized cameras. In this sequence, the area of interest was discretized into $G = 56 \times 56 = 3136$ locations. The outdoor dataset was shot in a university campus and it includes up to four individuals appearing simultaneously. This sequence was shot by three synchronized cameras. The area of interest in this sequence was discretized into $G = 40 \times 40 = 1600$ locations. We have solved the optimization problem using the Homotopy algorithm [1] with λ set to 0.1 for the sparse representation problem at each camera.

4.2. Tracking in an Indoor Environment

In this subsection, we present the performance of our method used for indoor multi-person tracking. In the experiments, we have compared our method with the block-based compression scheme of [2]. Here we consider the version of [2] that uses DCT for feature compression with a block size of 8×8 and took only the 1, 2, 3, 4, 5, 10, and 25 most significant coefficient(s) per block. Consequently, with the likelihoods of 56×56 size, at each camera in total we end up with at most 49, 98, 147, 196, 245, 490 and 1225 coefficients per person. In our method, after sparse representation of color model likelihood of a person of interest is found, we only took 10, 15, 20, 25, 50 and 100 most significant coefficients.

A groundtruth for this sequence is obtained by manually marking the people in the ground plane. Tracking errors are evaluated via Euclidean distance between the tracking and manual marking results. Figure 3 presents the average of tracking errors over all people versus the total number of significant coefficients used in communication for our sparse representation framework. Since the total number of significant coefficients sent by a camera may change depending on the number of people at that moment, the maximum is shown in Figure 3. The performance of the block-based compression approach in [2] is also

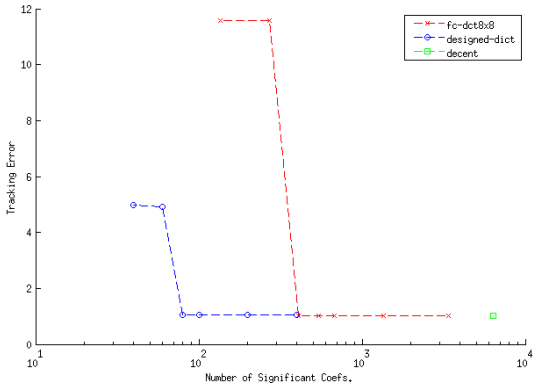


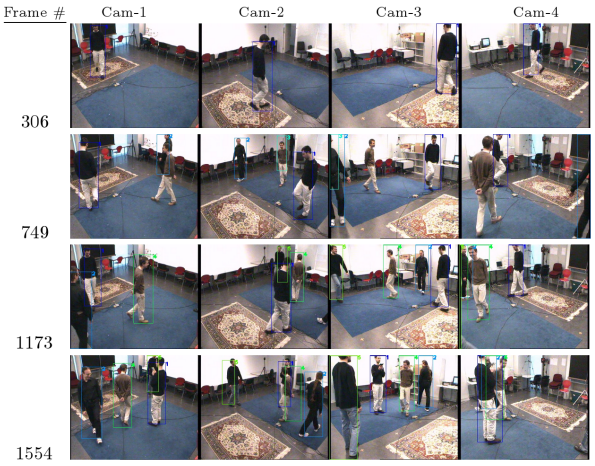
Figure 3. Indoor sequence: The average tracking errors vs. the number of coefficients for the approach in [2] (red), our framework (blue) and a decentralized method (green) that directly sends likelihood functions.

presented in Figure 3. It can be clearly seen that by using the custom-designed dictionaries, our sparse representation framework achieves much more bandwidth reduction than using the block-based compression approach. We note that the approach in [2] was shown to provide bandwidth savings over an image compression approach, which our approach improves further. To achieve an error of 1 pixel in the grid in average, our approach using specially-designed dictionaries needs at least 20 coefficients per person, whereas the approach in [2] needs at least 147 coefficients per person. The tracking results of the approach in [2] using 49 coefficients per person and our approach using 20 coefficients per person are given in Figure 4. It can be seen that, although the block-based compression approach can track the first and the second individuals very well, there is an identity association problem for the third and fourth individuals. Clearly, we can see that all people in the scene can be tracked very well by our approach.

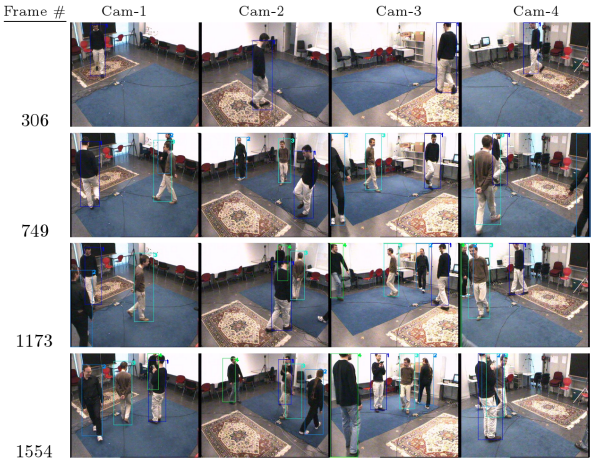
In the light of the results we obtained, for the same tracking performance, our method saves 80.39% of the bandwidth used by the block-based compression approach. Our method is also advantageous over an ordinary decentralized approach that directly sends likelihood functions to the fusion node. In such an approach, we send each data point in the likelihood function, resulting a need of sending 12544 values for tracking four people. The performance of this approach is also given in Figure 3. Our approach uses only 1.25% of the bandwidth needed by the decentralized approach.

4.3. Tracking in an Outdoor Environment

The performance of our method for outdoor multi-person tracking is presented in this subsection. Again, we have compared our method with the block-based



(a)



(b)

Figure 4. The tracking results for the indoor sequence: (a) the block-based compression approach in [2] using 49 coefficients per person, (b) our sparse representation framework using 20 coefficients per person in communication.

compression scheme in [2] using DCT domain with a block size of 8×8. For this approach, we took only the 4, 5, 10, 25, 30, 35 and 40 most significant coefficient(s) per block. Consequently, with the likelihoods of 40×40 size, at each camera in total we end up with at most 100, 125, 250, 625, 750, 875 and 1000 coefficients per person. In our method, after sparse representation of color model likelihood of a person of interest is found, we only took 5, 10, 15, 20, 25, 50 and 100 the most significant coefficients.

As in the indoor sequence, tracking errors are evaluated via the Euclidean distance between the tracking and manual marking results. Figure 5 presents the average of tracking errors over all people versus the total number of significant coefficients used in communication for our sparse representation framework. Again, the maximum number of coefficients is shown here. The performance

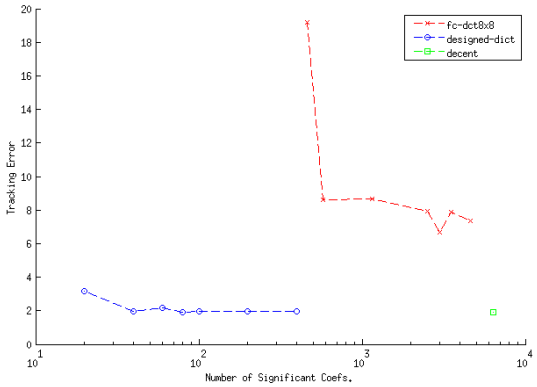


Figure 5. Outdoor sequence: The average tracking errors vs. the number of coefficients for the approach in [2] (red), our framework (blue) and a decentralized method (green) that directly sends likelihood functions.

of the block-based compression approach in [2] is also presented in Figure 5. It can be clearly seen that the block-based compression approach fails to maintain robust tracking while decreasing communication load in the network. Using less coefficients with this approach causes identity association problems. On the other hand, by using our framework we achieve an error of 1 pixel in the grid in average with 10 coefficients per person. The tracking results of the approach in [2] using 100 coefficients per person and our approach with specially-designed dictionaries using 10 coefficients per person are given in Figure 6. It can be seen that, the block-based compression fails to preserve identities. Especially, when a person leaves the scene and comes back, the person cannot be recognized and he or she is considered as a new person in the scene. Clearly, we can see that all people in the scene can be tracked very well by our approach with specially-designed dictionaries using 10 times less coefficients.

Based on these results, we can say that, our sparse representation framework successfully decreases communication load in the network without significantly degrading tracking performance. Our method is also advantageous over an ordinary decentralized approach that sends 6400 values for tracking four people (Figure 5). Our approach uses only 0.63% of the bandwidth needed by the decentralized approach.

5. Conclusion

Using a camera in a wireless network poses unique and challenging problems. This paper presents a novel method that can be used in VSNs for multi-camera person tracking applications. In our method, tracking is performed in a decentralized way: each camera extracts useful features

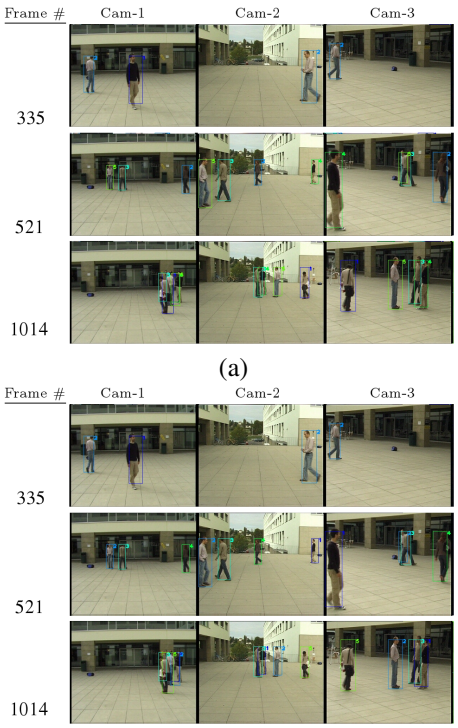


Figure 6. The tracking results for the outdoor sequence: (a) the block-based compression approach in [2] using 100 coefficients per person, (b) our sparse representation framework using 10 coefficients per person in communication.

from the images it has observed and sends them to a fusion node which performs tracking. In tracking, usually, extracting features results a likelihood function. Instead of sending the likelihood functions themselves to the fusion node, we have compressed the likelihoods via sparse representation methodology. In particular, we have designed special overcomplete dictionaries that are matched to the structure of likelihood functions and used these dictionaries for the sparse representation of likelihoods. To the best of our knowledge, this is the first method that uses sparse representation and specially designed dictionaries to compress likelihood functions and applies this idea for VSNs. Another advantage of this framework is that it does not require the use of a specific tracking method. In the light of the experimental results, we can say that our sparse representation framework is an effective approach that can be used together with any robust tracker in VSNs. By using overcomplete dictionaries that are matched to the structure of the likelihoods, for the same tracking performance, we achieve more bandwidth saving compared to existing methods.

References

[1] M. S. Asif and J. Romberg. Fast and accurate algorithms for re-weighted l1 norm minimization. *Submitted to IEEE Transactions on Signal Processing, July 2012*.

[2] S. Coşar and M. Çetin. Feature compression: A framework for multi-view multi-person tracking in visual sensor networks. *submitted to Journal of Visual Communication and Image Representation*.

[3] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83 –91, March 2008.

[4] S. Fleck, F. Busch, and W. Straß er. Adaptive probabilistic tracking embedded in smart cameras for distributed surveillance in a 3d model. *EURASIP J. Embedded Syst.*, 2007(1):24–24, 2007.

[5] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *PAMI*, 30(2), February 2008.

[6] A. Gupta, A. Mittal, and L. Davis. Constraint integration for efficient multiview pose estimation with self-occlusions. *PAMI*, 30(3), March 2008.

[7] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53 –69, Jan. 2008.

[8] H. Medeiros, J. Park, and A. Kak. Distributed object tracking using a cluster-based kalman filter in wireless camera networks. *Selected Topics in Signal Processing, IEEE Journal of*, 2(4):448 –463, Aug. 2008.

[9] E. Oto, F. Lau, and H. Aghajan. Color-based multiple agent tracking for wireless image sensor networks. In *ACIVS06*, pages 299–310, 2006.

[10] P. V. Pahalawatta and A. K. Katsaggelos. Optimal sensor selection for video-based target tracking in a wireless sensor network. In *in ICIP*, 2004.

[11] L. Potter, E. Ertin, J. Parker, and M. Cetin. Sparsity and compressed sensing in radar imaging. *Proceedings of the IEEE*, 98(6):1006 –1020, June 2010.

[12] B. Song and A. Roy-Chowdhury. Robust tracking in a camera network: A multi-objective optimization framework. *Selected Topics in Signal Processing, IEEE Journal of*, 2(4):582 –596, Aug. 2008.

[13] M. Taj and A. Cavallaro. Distributed and decentralized multicamera tracking. *Signal Processing Magazine, IEEE*, 28(3):46 –58, May 2011.

[14] J. Yao and J.-M. 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*, Oct. 2008.

[15] J. Yoder, H. Medeiros, J. Park, and A. Kak. Cluster-based distributed face tracking in camera networks. *Image Processing, IEEE Transactions on*, 19(10):2551 –2563, Oct. 2010.