Title

People Tracking in Camera Networks: Three Open Questions

Journal

Computer, 48(3)

ISSN

0018-9162

Authors

Thakoor, Ninad S

An, Le

Bhanu, Bir

et al.

Publication Date

2015

DOI

10.1109/mc.2015.83

Peer reviewed

RESEARCH FEATURE

People Tracking in Camera Networks: Three Open Questions

Ninad S. Thakoor, Le An, and Bir Bhanu, University of California, Riverside
Santhoshkumar Sunderrajan and B.S. Manjunath, University of California, Santa Barbara

Camera networks provide opportunities for practical video surveillance and monitoring, but tracking people across the network presents many computational and modeling hurdles that researchers have yet to surmount.

In the aftermath of the 15 April 2013 Boston Marathon bombing, investigators found themselves with massive amounts of data from surveillance video cameras, but they struggled to gain useful information from the footage. The primary problem was the inability to track people across camera views, which is vital to making sense of footage from multiple cameras.

The Boston incident underlines the need for more in-depth research on how to keep tabs on the location and identity of dynamic objects in a scene, which is foundational to automatic video analysis for applications such as surveillance, monitoring, and behavioral analysis. Research into tracking people in a single-camera view has matured enough to produce reliable solutions,[1] and smart camera networks are sparking interest in tracking across multiple-camera views. However, tracking in this context has many more challenges than in a single view. When networked cameras have partially overlapping views, spatiotemporal constraints enable tracking, but in larger camera networks, overlap is often impractical, and appearance is the key tracking enabler.

For these nonoverlapping views, tracking across camera views becomes a reidentification problem governed by features such as color, shape, and soft biometrics. These appearance-based features need to be reliable even with arbitrary camera poses, illumination changes, and object occlusion. These requirements raise two important questions for computer vision research. The first is

> *With nonoverlappng views, how can we model appearance with robust features and match it accurately and efficiently with database subjects?*

Most methods that address overlapping views assume a common coordinate system in estimating object location. However, estimation errors and packet losses in neighboring camera nodes can seriously affect the fusion scheme that a particular camera has adopted. This raises the second question:

> *With overlapping views, how do we fuse object location estimates from neighboring camera nodes in a way that can resist outliers?*

With or without overlapping views, any camera network faces global bandwidth constraints, making it infeasible to transmit all the raw video data between every node pair. Also, individual nodes are ill equipped to perform large computations. Balancing the amount of

0018-9162/15/$31.00 © 2015 IEEE

(a) (b) (c)

FIGURE 1. Boston bombing suspects viewed in multiple cameras. Views of the same person differ markedly in three distinct camera views, highlighting the complexity of relying on biometrics for subject reidentification.

processing at the individual node with the amount of data to be transferred to neighboring nodes is a daunting problem and at the heart of the third question. Answering this question, which is more in the domain of networking and distributed computing, will become critical once computer vision researchers answer the first two questions.

> *What kind of information must we extract at individual camera nodes, and how should we share it with neighboring camera nodes for consistent distributed tracking in real time?*

Researchers are already attempting to answer the first two questions, applying multicamera tracking algorithms to applications such as monitoring indoor and outdoor scenes. We have explored this work to highlight basic principles that are relevant to both nonoverlapping and overlapping camera networks in broad areas.

We also looked briefly at relevant progress in distributed computing that addresses big data volume. Such efforts will become central to scaling camera networks.

NONOVERLAPPING VIEWS

Environments such as offices, buildings with multiple floors, schools, and airports are not conducive to overlapping camera views and thus must rely on person reidentification—a recognition task that essentially matches individuals across nonoverlapping camera views. Accurate person reidentification is required to track a specific subject throughout a building equipped with multiple nonoverlapping surveillance cameras.

Person reidentification is closely related to classical people tracking and individual recognition, but it has important differences. Classical people tracking involves estimating a person's trajectory from frame to frame as the person walks into a camera's view. Individual recognition involves determining the identity of a query subject by matching it with subjects in a database.

Person recognition is generally based on biometrics such as a person's face, iris, and gait—all of which are extremely difficult to capture reliably in a camera network, as Figure 1 illustrates.

Person reidentification must address the data association problems that arise when video or image capture is discontinuous in space and time—a daunting prospect because of the many factors that contribute to discontinuity:

- *Low resolution*. Most surveillance cameras have hardware limitations that prevent them from capturing high-resolution images.
- *Arbitrary poses*. Networked cameras have arbitrary orientations, which means the same person in each camera view can have a different orientation.
- *Changing illumination*. Because cameras capture images at different times and locations, the amount of light varies, which can dramatically change a person's appearance.
- *Occlusion*. Various dynamic or static objects in the scene can occlude a person, or the person might have accessories like a suitcase or backpack that partially occlude parts of the body critical to identification.

Reidentifying a person from a large database requires first extracting robust visual features (for example, color, shape, and texture) for both the query and database subjects and then matching the two sources relative to the extracted features. In other words, reidentification is *appearance based*.

Person-reidentification methods aim to extract feature representations with low variations for the same subject (intraclass) and high variations among different subjects (interclass). However, because a person's appearance can change significantly across cameras, the intraclass variation is often larger than the interclass variation, which makes accurate classification challenging.

To address this problem, researchers have devised approaches to learn the optimal distance metric for image pairs. The idea is to weight features according to their perceived importance in reidentification. The system applies machine learning techniques to learn a transformation of the original feature representation. The resulting feature representation is then the basis for minimizing intraclass distances and maximizing interclass ones. The disadvantage is that the learned transformation tends to overfit to the training data.

Research to refine metric learning approaches and improve the reliability

of appearance-based matching has produced a spectrum of reidentification schemes. Although some reidentification techniques such as color invariants concentrate on feature representation, the majority are metric learning approaches that apply a range of machine learning techniques.

Color invariants

Typically, color is a powerful reidentification cue, usually based on histograms of body parts or the entire body in color spaces, such as red-green-blue or hue-saturation-value. However, because cameras and imaging conditions vary, perceived color can be significantly different across scenes.

One proposal to overcome this problem[2] is based on invariant color description. Rather than depending on descriptions of exact color distributions, the approach relies on shape descriptors that encode the structure of those distributions as clouds, each with a distinct color. To provide the invariant description, the shape descriptor codes the cloud's shape and relative orientation. Thus, legs might have one cloud and the torso another. The authors report improved reidentification performance relative to traditional histogram description.[2]

Mining feature importance

The metric learning approach assumes that the assigned feature weights are universally accurate, but that might not always be the case. Clothing color, for example, might receive a high weight because of its importance in reidentification, but in a scene in which many people are wearing the same color jeans, clothing color is less important.

One research group[3] has proposed a method to determine person-specific feature importance without supervision. The method first clusters appearance features to find representative prototypes from the data and then determines the weights for each prototype's individual features by learning to classify the prototypes correctly on the basis of those features. For a reidentification query, the method computes query-specific weights by assigning the closest prototype weights or by combining the query's distances from all the prototypes. The method complements top-down supervised metric learning, offering a bottom-up, unsupervised approach to determining feature importance.

Relative distance comparison

Traditional metric learning approaches to reidentification must contend with large intraclass variations from changing image conditions. Each person can have a different degree of intraclass variation, and learning is required for a large number of undersampled classes. When learning's goal is to minimize intraclass distance while maximizing interclass distance, these limitations can lead to model overfit or intractability.

One proposed solution[4] addresses these limitations by formulating reidentification as a relative distance comparison (RDC) problem. Under this model, the distance for a true query match must be smaller than that for wrong matches. This constraint is more relaxed than trying to minimize intraclass distance for all the examples in some classes while maximizing the distance of all the examples in the others. Because the model is concerned only with the relative distances of query matches versus nonmatches, not their exact values, intraclass variations introduce less bias.

The authors report that the approach outperforms similar distance learning approaches,[4] such as Information Theoretic Metric Learning, adaptive boosting (AdaBoost), and RankSVM, a method for ranking based on support vector machines.

Reference-based reidentification

Reference-based reidentification[5] recognizes that appearance can change radically across camera views and addresses the difficulty of direct matching. Unlike current methods that directly compare the query and database, the proposed solution introduces a novel scheme that uses an independent reference set to indirectly match the query and database image. The method consists of two main steps: canonical correlation analysis (CCA) and the generation of learning and reference descriptors (RDs).

Canonical correlation. Using CCA to learn projection matrices maximizes the correlation between data from different views. CCA explores the relationship between two sets of random variables from different observations of the same data, such as subject images from different views. Its optimization stage finds projections with a maximum correlation between the two random variable sets—that is, it finds a transformation that more accurately couples features of the same subject from different views.

CCA, which is performed in an unsupervised manner, finds candidate projections by solving equivalent generalized eigenvalue problems. By choosing a certain number of basis

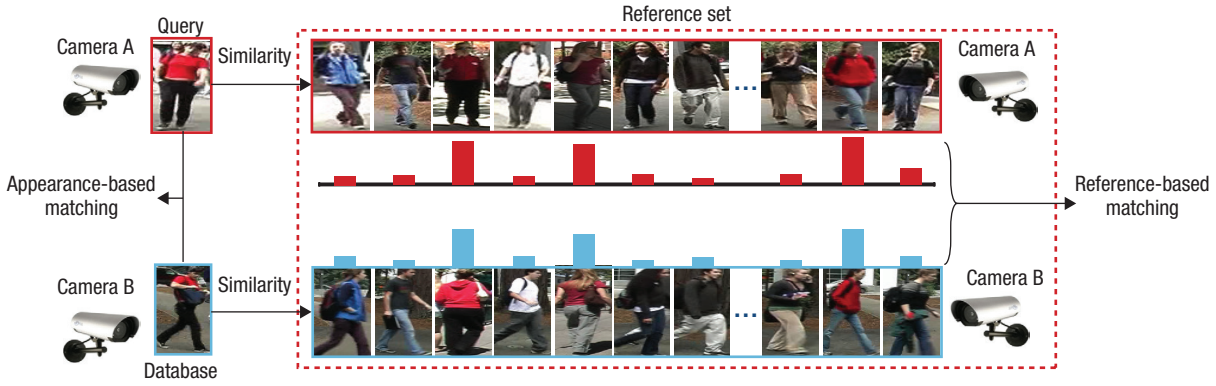

FIGURE 2. Generating a reference descriptor (RD). The system builds an RD from images in the reference set that most closely match the query image. For example, all the images in the red box are mostly frontal subject views because the query image has a frontal view. The database already holds the subject's profile view, which is represented by an RD of mostly profile views (images in the blue box). Bar heights indicate the degree of similarity between the images being described (query or database) and the reference set. The red and blue bars match closely for the query and database, which implies that RD-based reidentification is reliable.

vectors, it can simultaneously reduce feature dimensions.

Reference descriptor generation. Another novel part of reference-based reidentification is that neither complex feature representation design nor distance metric learning is necessary. Instead, a reference set becomes the basis for generating RDs for the query and database subjects. The reference set contains images of individuals that differ from the query and database subjects. For each individual in a reference set, the system stores images from different views—essentially acting as an intermediate matching mechanism.

As Figure 2 shows, computing the similarity between the query subject and each reference set individual provides the RD for a query subject. Each individual in reference set has multiple images, but to construct an RD, the system uses only the image most similar to the subject. In the figure, for example, the system chooses only frontal images for each individual in the reference set because the query subject is a frontal image. The subject's profile view is already in the database, stored as an RD that the system generates by comparing the subject's profile view to profile views of reference individuals.

Reidentification begins by extracting color and texture features for the query, database, and reference subjects. The system then applies CCA transformation to the query, database, and reference features and finally generates RDs for query and database subjects by comparing them to reference set individuals. Reidentification ends with a pair of matched RDs.

The reference-based method uses cosine similarity to find the database subject most similar to the query subject. Relative to similarity and distance measures, such as Euclidean and chi-square distance, cosine similarity is computationally more efficient, particularly for feature descriptors with many dimensions, and is scalable to large databases. Because RDs replace original features, there is no longer a need to compare image pairs directly to find the best match, which streamlines computation.

Another advantage of the reference-based method is that RDs are more distinct among different subjects and more consistent for the same subject despite the large appearance variation in the original images. This sharper distinction is a direct result of maximizing correlation through CCA.

Finally, because RD dimensions are independent of the original feature dimensions, it is possible to extract different features from different camera views for better discrimination.

All these advantages make the reference-based method a promising new approach to people tracking. Indeed, in an evaluation, it outperformed other state-of-the-art techniques including relative distance comparison.[5]

Context-aware learning for reidentification

Like most computer vision approaches, reidentification uses a fixed parameter or feature set that the system designer defines through experimentation or training. However, as environmental conditions change, this set might become invalid, which can degrade system performance. One novel proposal[6] avoids this problem by mapping environmental conditions or context to reidentification parameters or features.

The method is based on the idea that the importance of each reidentification feature—for example, scene illumination, a person's distance from the camera, subject size, and image-capture noise—can vary with context, which is time dependent. The method models feature importance as time-dependent weights, which are determined through reinforcement learning (RL),[7] a stochastic learning automaton that chooses actions according to some probability distribution.

In RL, the environment evaluates the actions and provides the machine with feedback or reinforcement. On the basis of this reinforcement, the machine updates its probability distribution so that the expected evaluation

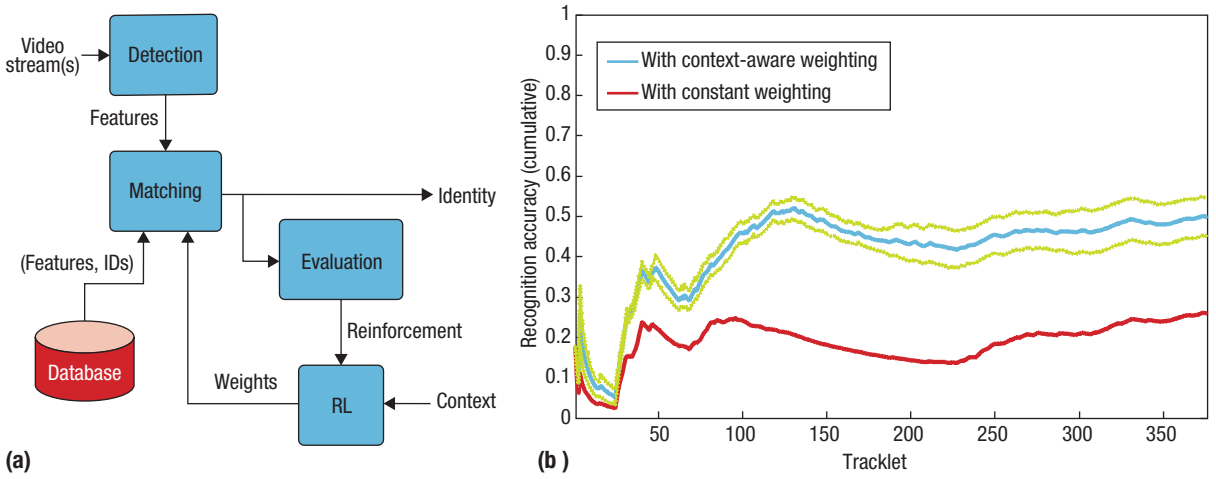

FIGURE 3. Context-aware learning. (a) Context-aware reidentification system and (b) comparison of cumulative accuracy with context-aware (blue) and constant (red) feature weighting with an average of 20 runs. Green lines represent the ±3 standard deviations of uncertainty for the 20 runs. Context-aware reidentification consistently performed higher than reidentification with fixed feature parameters. RL: reinforcement learning.

will be more favorable. *Reinforce*, a class of RL algorithms, can immediately evaluate the correctness of each identification decision. However, because these algorithms have binary output, the stochastic real-valued (SRV) algorithm is a better choice, since it allows real-valued outputs and associative learning. The real-valued outputs enable direct learning of feature importance, while the associative learning ability adds context awareness.

Figure 3a shows the structure of a reidentification system with context awareness. The system's database ($B = \{e_1, e_2, \ldots, e_L\}$) holds L examples of K different individuals from different cameras, where $L \geq K$, since the database can hold multiple examples of the same person. The lth example, e_l, is a pair of features and corresponding person (F_l, ID_l). When queried with Q_i, the database returns a ranked list of identities (output) $O_i = o_i^1, o_i^2, \ldots o_i^K$.

The list is ranked according to decreasing similarity or increasing distance. The distance D between query Q_i and example e_l is given by

$$D(Q_i, e_l) = \sum_{m=1}^{M} w^m(t) d(f_m, f'_m),$$

where $w^m(t)$ is the time-dependent weight for each feature, f_m is the feature from the query, f'_m is the feature from the example, and $d(\cdot)$ is the appropriate distance measure.

The context-aware method uses the SRV algorithm to learn the mapping between context and weights. As the user presents a query to the database, the method first estimates context and then uses the SRV algorithm to compute the weights, which it then supplies to the database. With updated weights, the database provides the ranked list O_i for the query, and the algorithm uses the true identity's rank O_{true} from the list to compute the reinforcement for the SRV units. A reinforcement of 1 signifies correct identification; 0 signifies incorrect identification. As Figure 3b shows, context-aware weighting significantly outperforms fixed weights.

OVERLAPPING VIEWS

Many researchers have investigated tracking across multiple cameras in the presence of ground planes, such as a road or floor—basically any plane in which human activity can take place. In most methods, a homographic transformation captures the relationship between the camera image plane and ground plane and represents the results as a 3 × 3 matrix ($\mathbf{H}_\pi$). Computing this matrix requires identifying parallel and perpendicular lines, which can be challenging. Urban scenes, for example, generally have roads, parking lots, and many buildings, which involve many such lines.

Multicamera person tracking requires estimating the *posterior*—a fusion of the individual's location estimates on the ground plane. One research group[8] has published a detailed comparison of decentralized and distributed tracking methods, focusing on the algorithms' energy and computational efficiencies.

Other work examines the problem from a computer vision perspective with the goal of building robust appearance and motion models. Research in this category uses probabilistic methods to promote collaboration between ground plane and image plane trackers. Most use either distributed Kalman filters or particle filters to estimate the posterior distribution of a subject's location on both the image and global ground planes, which yields the posterior density. Figure 4a illustrates how object location estimates from multiple cameras can be the basis for estimating posterior density on the global ground plane. (Posterior distribution is a probability distribution that represents updated beliefs about the parameter after seeing the data.)

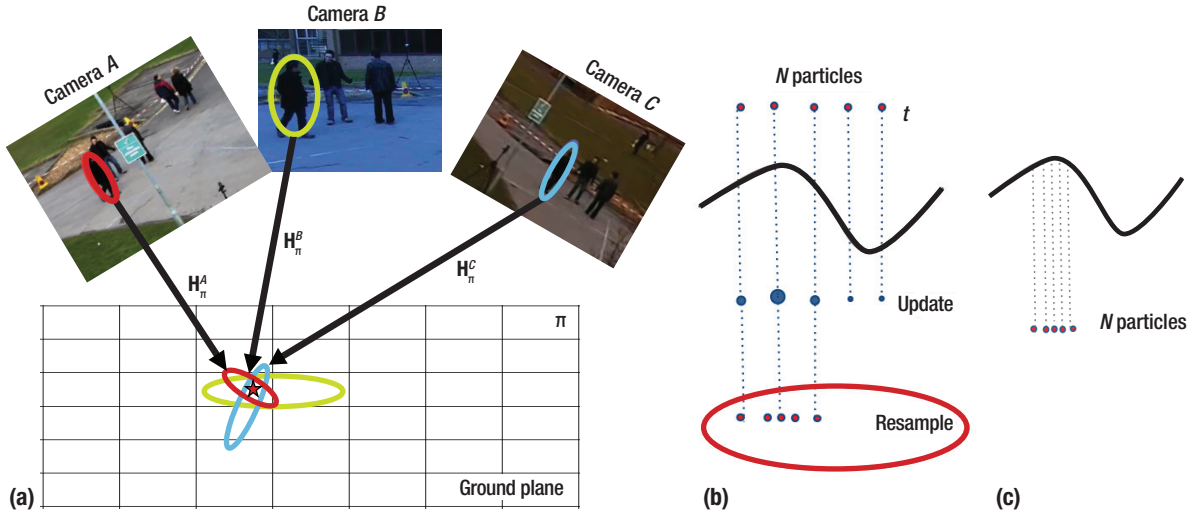

FIGURE 4. Estimating ground plane posterior density for object tracking. (a) Object location estimates obtained by homographic transformation from different views are fused to estimate the object's global ground plane location. $\mathbf{H}_\pi^A$, $\mathbf{H}_\pi^B$, and $\mathbf{H}_\pi^C$ represent the homography transformation between the ground plane π and Cameras *A*, *B*, and *C*, respectively. (b) Estimation with sequential importance resampling (SIR) particle filters, which add a resampling step; and (c) estimation with particle filters based on Markov Chain Monte Carlo (MCMC) sampling.

Kalman consensus filtering

Multicamera person tracking on the ground plane is essentially a consensus problem, since cameras use node-to-node communication to reach agreement about an individual's location. One solution[9] involves using Kalman consensus filtering based on distributed state estimation, assuming a linear dynamic system for state-space modeling. At every time step *t*, each camera finds its neighboring nodes and transmits the location estimate from time (*t* – 1) along with the corresponding information matrix. It also receives messages from neighboring cameras and fuses the information to compute the object's posterior state and error covariance matrix.

The approach assumes a Gaussian posterior, but in real-world scenarios, people exhibit complex motion patterns that are generally non-Gaussian. Others have shown that methods based on the Markov Chain Monte Carlo (MCMC) technique generalize well for arbitrary distributions in posterior estimation and can also model nonlinear object dynamics.[10]

Particle filtering

Methods based on particle filtering represent the posterior distribution as a set of weighted particles. Particle filters based on sequential importance sampling (SIS) evaluate how well each particle conforms to the assumed dynamic model and interpret the observations. Using this assessment, an algorithm generates a weighted particle approximation to the filtering distribution and computes posterior state estimates.

The goal of object tracking is to find the best object configuration $\mathbf{X}_t = [x_t, y_t]$ given observations up to time *t* where x_t and y_t are the coordinates of the object on the image or ground plane. The algorithm obtains the optimal object configuration using maximum a posteriori (MAP) estimation:

$$\hat{\mathbf{X}}_t = \arg\max_{\mathbf{X}_t} p(\mathbf{X}_t | \mathbf{Y}_{1:t})$$

A set of weighted particles in the sequential importance resampling (SIR) filters approximates the posterior at time *t* – 1:

$$p(\mathbf{X}_{t-1} | \mathbf{Y}_{t-1}) \approx \{\mathbf{X}_{t-1}^{(p)}, \pi_{t-1}^{(p)}\}_{p=1}^{P}$$

where *p* is the particle index and *P* is the number of particles. The weight of the *p*th particle is given by $\pi_t^{(p)} = p(\mathbf{Y}_t | \mathbf{X}_t^{(p)})$. Particle weights tend to accumulate in a few particles, a trend often referred to as particle degeneracy. SIR particle filters introduce a step to heuristically resample particles with larger weights before propagating state space. Figure 4b illustrates this process.

Distributed particle filtering approaches are computationally intensive relative to their parametric counterparts, and the information that a collaboration algorithm needs is not obvious. One approach to multicamera object tracking on the ground plane uses collaborative particle filters.[11] For every object, a pair of particle filters collaborate: one on the image and another on the ground plane. First, the image plane particle filters pass messages about the target location to the ground plane particle filter, which integrates information from multiple cameras on the basis of where the target's principal axis intersects. The ground plane particle filter then fuses multicamera information, and the image plane particle filter incorporates the fused results.

Compared to fusion frameworks that rely on precise feet positions, the use of the objects' principal axis during fusion produced significantly more accurate results. However, the proposed algorithm operates in an open-loop fashion; that is, it does not use ground plane tracking results to learn a better image plane tracking model.

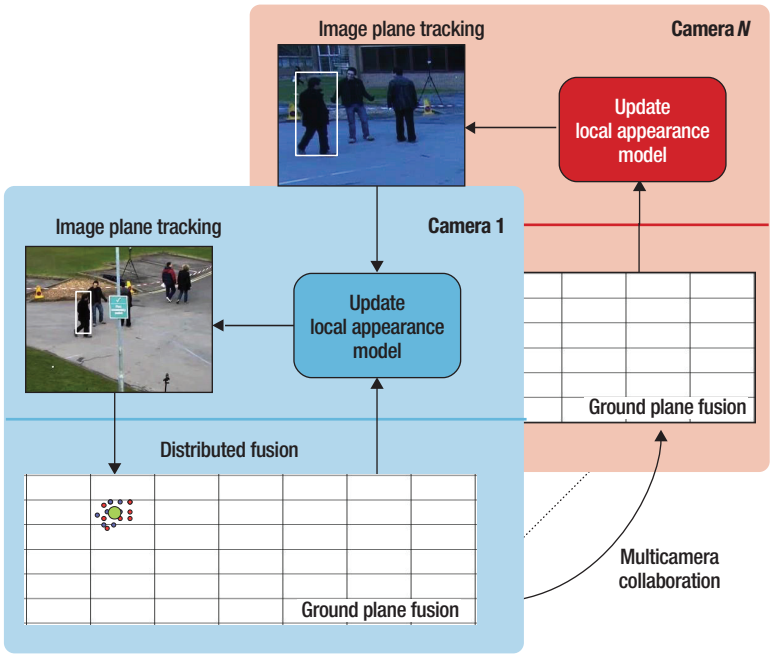

FIGURE 5. Active collaboration between image plane and ground plane trackers. The algorithm uses ground plane fusion results to learn a better appearance model for image plane tracking and generates training samples.

Closed-loop interaction with active collaboration

An alternative proposal introduces closed-loop interaction with an active collaboration mechanism based on distributed particle filter algorithms.[12] Figure 5 illustrates the active collaboration mechanism. The algorithm directly addresses the closed-loop interaction problem between image plane and ground plane tracking by using multiple instance learning to model object appearance in the image plane and generating training samples from particle locations.

Multiple-instance learning is a variation of supervised learning, in which the system provides labels for instance sets, or bags, instead of for individual instances and trains the classifier with a label ambiguity. At every time instance, t, the algorithm uses the appearance classifier to weight image plane particles and then shares particles with neighboring camera views. With the particles from neighboring camera views, each camera learns a Gaussian mixture model (GMM) to represent the posterior distribution on the ground plane. The method is based on the assumption that the entire network is time synchronized and that precomputed ground plane homography is available.

With the learned GMM, the system reweights image plane particles according to the ground plane posterior distribution. The reweighted particles define the positive samples for training the discriminative appearance model. The system randomly generates negative samples for training from the area outside of positive samples. By doing this, it effectively feeds back the ground plane fusion results to the image plane tracker and learns a robust appearance model from the samples computed using particle locations on the image plane.

The fusion mechanism is not resistant to outliers. For example, the overall tracking algorithm could still fail if most of the nodes fail. Although the method requires no raw image data transfer between camera views, transferring particles back and forth imposes a serious communication burden.

One solution to this problem uses prior knowledge about the scene to improve tracking.[13] The distributed tracking algorithm, which is based on MCMC sampling, uses a set of particle filters for every object. The local particle filter models object motion in the image plane. The global particle filter models object motion in the ground plane and takes prior scene knowledge into account to make results robust to outliers.

It is not straightforward to mix local and global particle filters. More important, the algorithm represents posterior distribution as a set of weighted particles, so it suffers from particle degeneracy.

The Metropolis-Hastings sampling algorithm[14] defines a Markov chain over configuration space $\mathbf{X}_t$, and the chain's stationary distribution equals the posterior distribution $p(\mathbf{X}|\mathbf{Y})$. As Figure 4c shows, the algorithm uses a set of unweighted samples

$$p(\mathbf{X}_t|\mathbf{Y}_t) \approx \{\mathbf{X}_t^{(p)}\}_{p=1}^{P}$$

to represent the posterior in MCMC-based particle filters and employs an interactive MCMC algorithm to combine local and global particle filters. MCMC method complexity varies linearly with object number, unlike the complexity of SIR particle filters, which varies exponentially.

An ensemble learner models local and global appearances. It explicitly models global appearance using Grassmann manifolds to account for viewpoint changes. The Grassmann manifold is the space of d dimensional subspace in R_n, and a point on the Grassmann manifold represents a subspace.

The sampling algorithm uses principal component analysis to project training samples from two views onto a lower dimensional subspace. It uses the

geodesic paths that are constant velocity curves on a manifold to obtain intermediate subspaces between two views. Finally, it uses intermediate subspaces that account for viewpoint changes to generate training samples for learning the global appearance model.

Serious issues remain in tracking people with multiple cameras. Researchers continue to confront computational and communication bottlenecks, as well as inaccurate appearance matching. Unfortunately, there are no ready answers to the open questions we pose, although work is in progress to address them.

For tracking with nonoverlapping camera views, reidentification is central, and most approaches try to tackle it as an image-matching problem without context awareness. However, recent work shows that context information can improve reidentification accuracy. Exploring a variety of contexts—both image and nonimage based—remains an active research direction.

For tracking with overlapping views, methods based on distributed filtering with Kalman and particle filters have weaknesses, but incorporating prior contextual information into the distributed particle filter framework shows promise. Algorithms can be used to leverage a variety of information on crowd flow, entry and exit points, and obstacles as input into filtering-based methods. Other algorithms are attempting to discriminate among people with a similar appearance who cross each other or form a group and then disperse either within a single view or across multiple views (co-occurence), with the goal of deriving more robust appearance models. Future research work should focus on exploiting these contextual priors to improve Bayesian MAP estimation.

ABOUT THE AUTHORS

NINAD S. THAKOOR is a project scientist at the Center for Research in Intelligent Systems at the University of California (UC), Riverside. His research interests include computer vision, pattern recognition, and image processing. Thakoor received a PhD in electrical engineering from the University of Texas at Arlington. He is a member of IEEE. Contact him at ninadt@ucr.edu.

SANTHOSHKUMAR SUNDERRAJAN is a senior member of the technical staff at Pinger, Inc. While performing the work reported in this article, he was a PhD candidate in electrical and computer engineering at UC Santa Barbara. His research interests include computer vision, camera networks, and large-scale machine learning. Sunderrajan received a PhD in electrical and computer engineering from UC Santa Barbara. Contact him at santhosh@ece.ucsb.edu.

LE AN is a postdoctoral research associate in the Biomedical Research Imaging Center at the University of North Carolina, at Chapel Hill. While performing the work reported in this article, he was a PhD candidate in the Department of Electrical Engineering at UC Riverside. His research interests include pattern recognition, machine learning, computer vision, and data mining. An received a PhD in electrical engineering from UC Riverside. He is a member of IEEE. Contact him at lan004@ucr.edu.

BIR BHANU is a Distinguished Professor of electrical and computer engineering and the director of the Center for Research in Intelligent Systems at UC Riverside. His research interests include image processing, computer vision, pattern recognition, machine learning, and databases. He is a Fellow of IEEE, the American Association for the Advancement of Science (AAAS), the International Society for Optics and Photonics (SPIE), and the International Association of Pattern Recognition (IAPR). Contact him at bhanu@cris.ucr.edu.

B.S. MANJUNATH is a professor of electrical and computer engineering at UC Santa Barbara. His research interests include bioimaging, informatics, large-scale image and video sensor networks, and multimedia databases. Manjunath received a PhD in electrical engineering from the University of Southern California. He is coeditor of *Introduction to MPEG-7* (Wiley, 2002) and a Fellow of IEEE. Contact him at manj@ece.ucsb.edu.

ACKNOWLEDGMENTS

Ninad S. Thakoor and Santhoshkumar Sunderrajan contributed equally to the original manuscript for this article. The work described in this article was supported in part by National Science Foundation grants 1330110 and 0905671 and by Office of Naval Research grants N00014-12-1-1026 and N00014-12-1-0503. The content herein does not reflect the position or policy of the US government.

REFERENCES

1. A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006; http://doi.acm.org/10.1145/1177352.1177355.
2. I. Kviatkovsky, A. Adam, and E. Rivlin, "Color Invariants for Person Reidentification," *IEEE Trans. Pattern*

Analysis and Machine Intelligence, vol. 35, no. 7, 2013, pp. 1622–1634.
3. C. Liu, S. Gong, and C.C. Loy, "On-the-Fly Feature Importance Mining for Person Reidentification," *Pattern Recognition*, vol. 47, no. 4, 2014, pp. 1602–1615.
4. W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by Relative Distance Comparison," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, 2013, pp. 653–668.
5. L. An et al., "Reference-Based Person Reidentification," *Proc. IEEE Int'l Conf. Advanced Video and Signal-Based Surveillance*, 2013, pp. 244–249.
6. N. Thakoor and B. Bhanu, "Context-Aware Reinforcement Learning for Reidentification in a Video Network," *Proc. ACM/IEEE Int'l Conf. Distributed Smart Cameras*, 2013; http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6778207.
7. R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
8. M. Taj and A. Cavallaro, "Distributed and Decentralized Multicamera Tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, 2011, pp. 46–58.
9. B. Song et al., "Tracking and Activity Recognition through Consensus in Distributed Camera Networks," *IEEE Trans. Image Processing*, vol. 19, no. 10, 2010, pp. 2564–2579.
10. J. Kwon and K-M. Lee, "Visual Tracking Decomposition," *Proc. 23rd IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 10), 2010, pp. 1269–1276
11. W. Du and J. Piater, "Multi-camera People Tracking by Collaborative Particle Filters and Principal Axis-Based Integration," *Proc. Asian Conf. Computer Vision*, 2007; pp. 365–374.
12. Z. Ni et al., "Distributed Particle Filter Tracking with Online Multiple Instance Learning in a Camera Sensor Network," *Proc. IEEE Int'l Conf. Image Processing*, 2010, pp. 37–40.
13. S. Sunderrajan and B.S. Manjunath, "Multiple View Discriminative Appearance Modeling with IMCMC for Distributed Tracking," *Proc. ACM/IEEE Int'l Conf. Distributed Smart Cameras*, 2013; http://dx.doi.org/10.1109/ICDSC.2013.6778203.
14. S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, vol. 49, no. 4., 1995, pp. 327–335.

Selected CS articles and columns are also available for free at **http://ComputingNow.computer.org**.