Self-reconfigurable smart camera networks

Juan C. SanMiguel[1], Christian Micheloni[2], Karen Shoop[1], Gian Luca Foresti[2], Andrea Cavallaro[1]
[1]Queen Mary University of London, United Kingdom - [2]University of Udine, Italy

Recent hardware advances such as multi-core high-speed platforms allow cameras to perform multiple tasks whilst observing a scene: smart camera networks enable emerging applications that require adaptation to unforeseen conditions, changing tasks and constrained resources.

The advent of smart cameras will significantly improve video analytics capabilities with the use of networked embedded devices for sensing, processing and communicating data and metadata. Smart camera networks underpin the emergence of pervasive applications such as unmanned vehicles to explore environments for search and rescue missions, disaster management, wildlife conservation and routine monitoring. Other growing uses of smart camera networks include in-home assistive technologies, collaborative recognition of events using networked wearable cameras, security and surveillance. Predicting human behavior in public spaces is also key to the vision of smart cities.

Self-configuration of smart camera networks is necessary to yield efficient high-level understanding of and adaptation to dynamic scenes. In this article, we discuss recent approaches for autonomous smart camera network reconfiguration where camera cooperation plays a crucial role. We outline the requirements for self-reconfiguration, discuss existing solutions, challenges and open research issues.

I. Why Self-reconfiguration?

Smart camera networks operate in potentially unknown or poorly mapped dynamic environments, with limited or changing information about the relative camera location. Networks of cameras are composed of heterogeneous devices that may be handheld or on-board unmanned vehicles. Such devices need to adapt their position and their Field of View (FOV) in response to changes in tasks and in the environment. For example cooperation may lead to improved visual coverage and to increased accuracy in localization of the objects of interest by integrating multiple information sources (Figure 1).

Smart camera network self-reconfiguration is the autonomous and cooperative search for a network state that optimizes certain criteria such as a predefined level of task performance or use of resources. This optimal state depends on the network structure and on the state of individual cameras. The optimization involves modifying algorithmic and hardware parameters. Algorithmic parameters include capturing and processing frame rates, image resolution, compression level and task definition [1][2][3]. Hardware parameters include the number of cameras participating in a task, their position and their individual FOV [4][5][6][7]. Changing the FOV involves optimizing its extrinsic parameters, such as orientation and zoom, and intrinsic parameters, such as iris and focus. This optimization depends on quality criteria defined over the captured images. Examples of quality criteria include a minimum resolution of observed targets and their orientation relative to the camera.

Each smart camera coordinates with others via task-related and operational decisions. Task-related decisions include learning the appearance of moving targets and estimating their position, whereas operational decisions include information sharing across cameras to complete a specific task, such as scheduling video analytics procedures. Maximization of task performance in a distributed manner via autonomous and interacting cameras is preferred to the traditional centralized camera network paradigm. Compared to the centralized approach, distributed solutions may reduce the amount of bandwidth needed for communication, and improve reconfiguration efficiency and robustness to single points of failure.

Robustness to failure is an important feature for smart camera networks as these networks need to cope with malfunctions, bandwidth limitations and time-variable network characteristics. The network structure can also be modified when cameras move, are turned off to save energy, and added or removed during runtime. For these reasons reconfiguration capabilities are required to adapt to unforeseen conditions while simultaneously performing a range of tasks. Such tasks cannot always be predefined and may need to be updated in real time (e.g. when tracking firefighters and people fleeing a burning building while simultaneously monitoring the evolution of the fire itself).

Cameras might need to allocate subtasks to other cameras in order to accomplish a particular goal (e.g. to track a temporarily occluded object from a different viewpoint). Procedural similarities can be found in control theory problems and wireless sensor networks [4]. However, concurrent task allocation is necessary in smart camera networks as the number of tasks may outnumber the number of cameras. Moreover, smart camera networks present specific challenges related to the adaptability and directionality of the FOV.

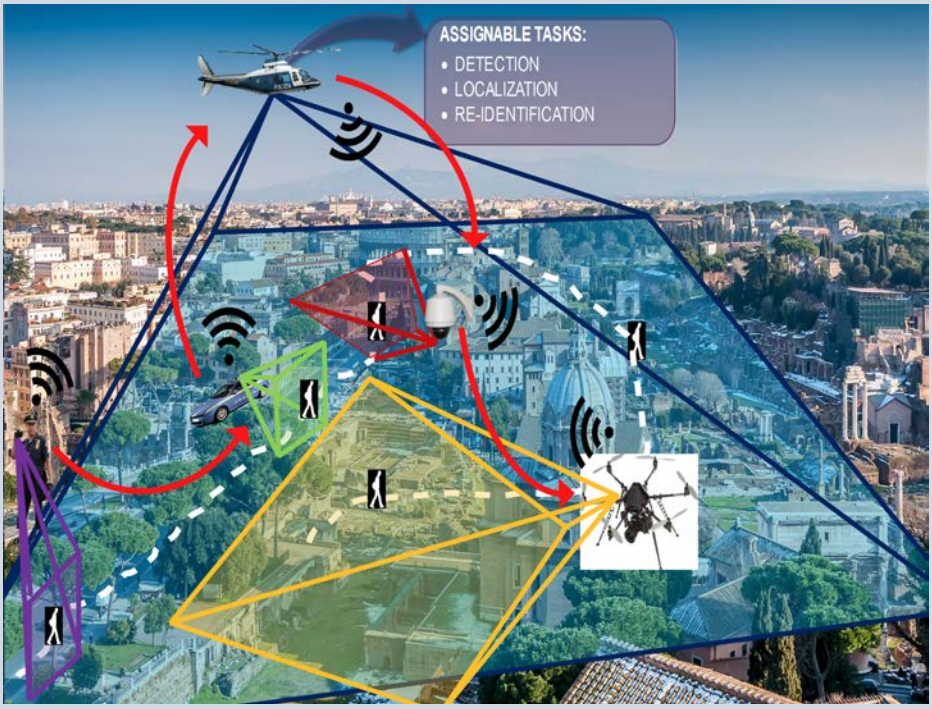

Figure 1 Sample scenario for a self-reconfigurable smart camera network. Heterogeneous cameras cooperate to distribute video analytics tasks. Cameras adapt their FOVs and the amount and type of information to be shared across the network to accomplish the tasks needed to achieve the goal of the network. Tasks may include detection, tracking, recognition and identification. Reconfiguration selects cameras [4], manages FOVs [6] and assigns tasks [2]. In the scenario depicted in this figure, cameras on unmanned aerial vehicles (UAVs) observe wider areas and predict desirable task-dependent network configurations. Cameras placed on unmanned ground vehicles (UGVs) can be redirected to inspect the scene from more convenient viewpoints to recognize specific objects, people or body gestures. Wearable cameras can be finally tasked to collect high-quality target close-ups to facilitate recognition and post-event investigation for law enforcement.

II. The five pillars of self-reconfiguration

Developing self-reconfiguration strategies for smart cameras requires modeling five interrelated elements, namely environment, camera, network, task and performance (Figure 2).

The *environment* describes the physical space being monitored [1], which includes observation locations (or control points) that must always be covered by the cameras, and static and moving obstacles that might lead to visual occlusions. A *camera* to be reconfigured is defined by its physical location (calibration) and resources related to power consumption (battery), sensing (sensor type and FOV), computation power (CPU and memory) and communication capabilities.

The *network* defines the connectivity among cameras, which can be described at vision and communication level [8]. Connectivity may change over time when considering camera mobility, time-varying numbers of cameras and communication bandwidth. A suitable protocol is necessary to manage communication and processing in the network. This protocol should consider many diverse aspects such as policies for sharing the physical medium, robustness against transmission failures, energy-efficient routing of information and support communication among heterogeneous devices.

Figure 2 Self-reconfiguring smart camera networks deal with five elements: environment, cameras, tasks, network and performance. Considering all these elements, smart cameras cooperate network-wide to find a configuration that maximizes situational awareness, which is achieved through the distributed and parallel execution of tasks such as detection, tracking and event analysis.

Task completion needs to consider resource constraints such as energy and computation costs [2]. Appropriate load balancing of tasks among camera nodes requires task decomposition according to complexity and required resources. For example, behavior recognition may be decomposed into object detection, feature extraction and spatio-temporal reasoning [9]. Each sub-task can be then allocated and performed by a different smart camera.

Finally, *performance* criteria quantify the success of a reconfiguration strategy. Typical performance measures consider the accuracy and the timeliness of task completion, the amount of energy used, the communication costs, and the lifetime of each camera and of the network.

In the next sections we outline how these intertwined elements are addressed to enable cooperative reconfiguration (Figure 3) while addressing the major smart camera network challenges, namely topology discovery and self-calibration, resource and task allocation, and active vision.

III. TOPOLOGY DISCOVERY AND SELF-CALIBRATION

The structural information of the smart camera network describes the spatial relationships among cameras. The primary goal of self-reconfiguration is to dynamically define the neighbors of each camera to enable cooperation. Adaptive self-calibration techniques describe the relative location of cameras and fuse data to satisfy tasks (e.g. to locate a target). Effective reconfiguration depends on the capability of deriving and updating the network structure via topology discovery and self-calibration.

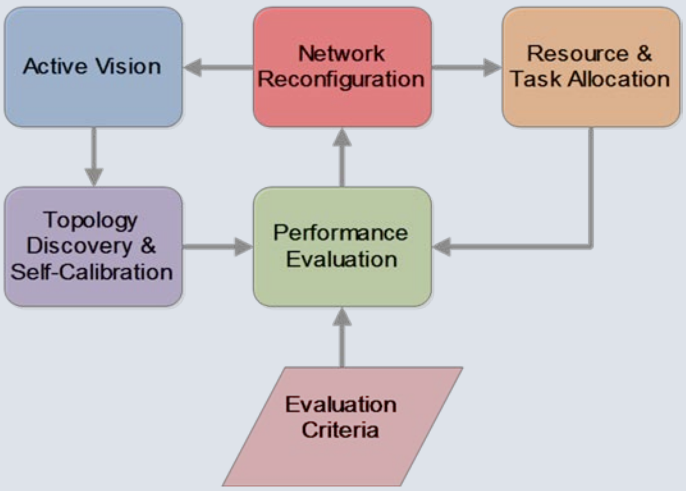

Figure 3. The self-reconfiguration of a smart camera network is a continuous cycle. Evaluation criteria describe the rules and the parameters to define the quality of the video analytics to be performed by the network and guides new reconfigurations. The network reconfiguration block selects cameras to be switched on/off and assigns tasks to operating cameras. All the decisions are taken considering the available resources.

Topology discovery identifies the neighbors of each camera. The concept of neighborhood is application and task dependent and, for a target tracking application, it is defined as the set of cameras which are more likely to capture a target that re-appears after leaving a FOV. The topology can be represented as an undirected weighted graph with links and nodes (cameras), where the links are weighted according to the spatial connectivity of the network. Directed graphs can be also considered if the relation between cameras is not symmetrical, such as in the case of a one-way traffic flow across FOVs. For overlapping FOVs, a pairwise cooperation scheme can be applied to determine neighbors by sharing and matching image features of different FOVs. Large-scale networks often present neighboring cameras with disjoint FOVs where camera neighbors do not share their FOV thus leading to unobserved areas. In this case, additional nodes can be defined [10] that represent the set of entry or exit regions of each FOV and their connectivity is described with additional edges in the graph.

The topology of a set of disjoint cameras can be found by analyzing motion patterns, by applying re-identification techniques to locate moving targets or by determining co-occurrences of events in the FOVs [11]. Target tracking in disjoint camera networks needs handover mechanisms that transfer the control of the tracking task across cameras and assign cameras to targets. When track-ownership is transferred, the corresponding weight in the topology graph is increased to reinforce connectivity (Figure 4). Auction mechanisms can be used to coordinate the handover: each time a target leaves the FOV of the camera assigned to track, its features are broadcast and cameras which can detect this target will then confirm their willingness to track [5]. The updated network topology is finally used to decrease the range of broadcast messages in future handover processes.

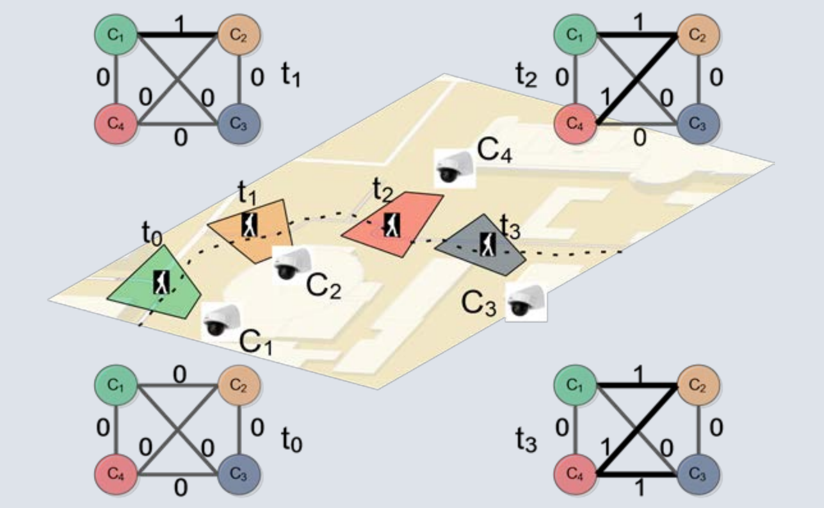

Figure 4. Target re-identification between disjoint FOVs helps to infer the spatial relationships of the cameras in the network. An iterative process executed for all the cameras and for all the targets discovers the network structure. Once estimated, the topology information helps the prediction of the next target appearance for the handover of the target ownership among cameras.

Self-Calibration is concerned with geometrical internal and external camera information. Internal camera information includes intrinsic parameters, which map the captured real world into the image plane, whereas external camera information comprises three-dimensional location and orientation within a global coordinate system. Calibration also allows computing the topology (i.e. the camera neighborhood) via geometrical analysis. Approaches for *self-calibration* vary depending on whether the FOVs are overlapping or disjoint. Generally, one camera is selected to define the coordinate system. A sequential calibration process starts from such initial camera and propagates throughout the network. Then cooperation strategies estimate the relative positions and orientations of the other cameras. For overlapping cameras, cooperative auto-calibration can be cast as a multi-view image matching problem where feature descriptors are extracted from each FOV and distributed across the network [12]. Cameras collaborate by comparing their locally computed descriptors with the descriptors received from the network in order to find the most suitable match. Pairwise correspondences are established between the FOVs to define potential matches. Next, calibration data is iteratively obtained and refined using optimization processes over the inter-camera correspondences. Cooperation can also be performed through local message exchange schemes, such as average consensus, to obtain estimations of the rotation and translation parameters for each camera [1]. For disjoint cameras, unobserved areas between FOVs make obtaining accurate calibration data a challenging task. As for topology, target motion can be employed to derive calibration data assuming a consistent motion structure across FOVs. In a network with disjoint FOVs, linear motion hypotheses are exploited to model the movement of multiple targets from one camera to another, and vice versa [13]. Such forward and backward models are employed to calibrate the cameras by pairwise cooperation via iterative refinement of the generated models.

IV. RESOURCE AND TASK ALLOCATION

Resource-constrained smart camera networks need to manage their battery lifetime, communication bandwidth and computation power. The importance of controlling the use of these resources increases with the scale of the network. Moreover, these resource-constrained settings require sharing the computational effort among cameras by allocating video analytics tasks to different cameras. Self-reconfiguration strategies can be applied for camera placement, dynamic resource management and task allocation.

Camera placement is concerned with deciding the number of cameras to be deployed, their absolute position and their relative configuration to fully cover an area of interest. This is an NP-hard problem for which a range of locally optimal solutions have been proposed considering different models of the FOVs and different objectives. For example, an activity density map that represents the locations in which targets frequently move can be matched with coverage models of the FOVs via centralized data fitting algorithms [6].

Dynamic resource management aims at changing the parameters of individual cameras and of the whole network to adapt to the scene dynamics and to the resource availability. Optimizing the use of resource to maximize network performance is an ongoing research area, sharing challenges with the control/robotics and the wireless sensor network communities [14]. These techniques manage camera power to meet the desired performance, to allocate resources for distributed tasks [2], or to optimize video data communication [3]. Performing such optimization in real-time is crucial for deploying sustainable networks and for extending their lifetime.

Task allocation or *scheduling* is a critical operation due to the complexity and volume of tasks for smart camera networks. This operation assigns different goals to each camera such that the primary network objective is accomplished. The solution has to consider different constraints imposed by resource availability, including the choice of which subset of cameras is most appropriate to satisfy a task [7]. Switching off unnecessary cameras to preserve power defines a new network topology that needs to be identified and tasks should be reallocated accordingly in real time. A distributed solution [2] must identify the different tasks to assign to cameras such that the fusion of their results achieves the expected level of description for the monitored area. Optimization methods can also be used for task allocation when each target is assigned to a specific quality of service (QoS) and different video analytic activities are activated [2].

Techniques such as evolutionary algorithms can help in defining metrics to evaluate constraints and objectives such as energy usage, QoS of video analytics procedures and amount of processing required [1]. Camera self-awareness can be increased via online estimation of task reliability. For example, cameras tracking moving targets might detect performance drops due to temporal occlusions of targets. One solution is re-configuring their cooperation strategies to improve task performance by handing tasks over to other cameras. Alternatively a camera could move to recapture a target in its FOV, provided this does not affect the tracking of other targets.

V. ACTIVE VISION

Active vision involves the continuous interaction between smart cameras and the environment to decide what to observe and how to prioritize observation tasks. Cameras can actively change intrinsic and extrinsic parameters (translation, rotation, pan, tilt and zooming) to adapt their FOVs to specific tasks. This adaptation can, for example, maximize object localization accuracy. Cameras can pan and tilt to improve the portion of the image covered by a selected target, zoom in on the target (a face or a number plate) for a closer look or tune internal parameters to enhance the quality of the captured image. These tasks can be accomplished in several ways, such as by increasing the iris parameter when a target enters a zone with low illumination conditions (e.g. areas in shade).

Cameras may compete or collaborate to maximize performance criteria. In the case of tracking, these performance criteria may include the target size and position. Selecting the most suitable camera to perform a task is frequently used for handover-based target tracking where cameras bargain to decide which tracks each target as it moves across different FOVs [15]. For example, collaboration via optimization methods ensures each target is covered by a certain number of active cameras reconfigured to achieve a specific FOV [2]. The optimization process computes a feasible configuration to satisfy the QoS for each target, which might also include distributing tasks among active cameras while meeting the network constraints.

Adapting a FOV based on the camera-network needs is subject to a trade-off between its capability to observe the whole scene and the capability to acquire important details about an object. For recognition tasks, a camera may need to narrow its FOV to zoom in on a particular feature, such as a face, thus neglecting information about the rest of the body and the wider scene. Other cameras must then cooperate to compensate for this lack of information by changing their parameters and modifying their FOVs accordingly. This task can be accomplished using game theory [4] by incorporating appropriate cost structures to quantify the performance of the tasks, the cameras and the overall network. For example, a camera may be providing close-up details of a target whereas a more distant camera can observe the whole target. Both cameras are rewarded with a higher preference (or lower cost, depending on the modeling) as they compensate each other for the risk of losing the global picture whilst capturing high-quality data. Provided individual task and camera costs are aligned to the global tasks of the network, cameras will adopt a cooperative strategy to maximize their preferences (or minimize their costs) and achieve an optimal solution. Scene understanding performance criteria such as tracking accuracy, pose, and image resolution can then be maximized [4]. A game-theoretic framework can be used jointly with a consensus approach [4] to implicitly coordinate the processing of neighboring cameras to estimate the location of moving targets. Explicit and implicit coordination strategies can be combined [9] to concurrently improve tracking accuracy and self-reconfiguration capabilities.

Cooperation in large-scale camera networks implies sharing information acquired from different viewpoints. Interaction mechanisms have to be developed for communicating among heterogeneous stationary and mobile devices. For example, person re-identification in camera networks can be aided by taking advantage of the estimation of future target movements across heterogeneous cameras [16]. Active cameras can self-configure for repositioning in order to increase re-identification performance. The integration of heterogeneous cameras as static and active sensors in the smart camera network is made challenging because of camera ego-motion, viewpoint differences and the need for image stabilization.

[Ending\conclusions]

Smart camera networks are at the frontier of work in cooperation and distributed decision-making for a broad variety of large-scale environments and applications. Networks of smart cameras can be employed to detect intruders in cities, to spot illegal logging in forests, to protect travel networks, and for search and rescue missions. Disaster recovery is receiving increasing interest as camera networks can support first responders after both man-made conflicts and natural disasters, as well as farming and wildlife conservation activities. Challenges to be addressed for facilitating the future development of cooperative reconfiguration strategies include adaptation to dynamic topology changes, coordination among heterogeneous devices, hierarchical data processing and the online evaluation of task performance. Envisioning large networks of heterogeneous smart cameras composed of static, wearable and mobile devices that autonomously interact and self-configure their capabilities to achieve the desired goals efficiently will depend on successfully addressing these challenges.

ACKNOWLEDGEMENTS

Juan C. SanMiguel was supported by the EU project Crowded Environments monitoring for Activity Understanding and Recognition (CENTAUR, FP7-PEOPLE-2012-IAPP) under GA number 324359. Christian Micheloni and Gian Luca Foresti were partially supported by the Italian Ministry of Defense under the grant "ADVISOR III: automatic deployment and self-reconfiguration of electro-optical sensors on board of UAVs for situational awareness. Andrea Cavallaro acknowledges the support of the Artemis JU and UK Technology Strategy Board as part of the Cognitive & Perceptive Cameras (COPCAMS) project under GA number 332913.

References

[1] Rinner, B.; Dieber, B.; Esterle, L.; Lewis, P. R. and Yao, X., "Resource-aware configuration in smart camera networks", IEEE Computer Vision and Pattern Recognition Workshops, pp. 58-65, Jun. 2012

[2] Dieber, B.; Micheloni, C. and Rinner, B., "Resource-aware coverage and task assignment in visual sensor networks", IEEE Trans. on Circuits and Systems for Video Technology, 21(10):1424-1437, Oct. 2011

[3] Aziz, S.M. and Pham, D. M., “Energy Efficient Image Transmission in Wireless Multimedia Sensor Networks”, IEEE Communications Letters, 17(6):1084-1087, Jun. 2013

[4] Ding, C.; Song, B.; Morye, A.; Farrell, J.A. and Roy-Chowdhury, A.K., “Collaborative sensing in a distributed PTZ camera network”, IEEE Trans. on Image Processing, 21 (7):3282-3295, Jul. 2012.

[5] Esterle, L.; Lewis, P.; Yao, X. and Rinner, B., “Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks”, ACM Trans. on Sensor Networks, 10(2): 1-24, Jan. 2014

[6] Micheloni, C.; Rinner, B. and Foresti, G.L., “Video analysis in pan-tilt-zoom camera networks”, IEEE Signal Processing Magazine, 27(5):78-90, Sept. 2010

[7] Tessens, L.; Morbée, M.; Aghajan, H. and Philips, W., “Camera selection for tracking in distributed smart camera networks”, ACM Trans. on Sensor Networks, 10(2): 1-33, Jan. 2014

[8] Tron, R. and Vidal, R., “Distributed Computer Vision Algorithms”, IEEE Signal Processing Magazine, 28(3):32-45, May 2011

[9] Roy-Chowdhury, A. K. and Song, B., “Camera networks: the acquisition and analysis of videos over wide areas”, Synthesis Lectures on Computer Vision, 3(1):1-133, 2012

[10] Wang, X., “Intelligent multi-camera video surveillance: A review”, Pattern Recognition Letters, 34(1):3-19, Jan. 2013

[11] Erdem, U. M. and Sclaroff, S., “Event prediction in a hybrid camera network”, ACM Trans. on Sensor Networks (TOSN), 8(2):1-27, Mar. 2012

[12] Devarajan, D.; Cheng, Z. and Radke, R., “Calibrating Distributed Camera Networks”, Proc. of the IEEE, 96 (10): 1625-1639, Oct. 2008

[13] Anjum, N. and Cavallaro, A., “Automated localization of a camera network”, IEEE Intelligent Systems, 27(5):10-18, Sep.-Oct 2012

[14] Li, W.; Delicato, F. C. and Zomaya, A., “Adaptive energy-efficient scheduling for hierarchical wireless sensor networks”, ACM Trans. on Sensor Networks, 9(3):1-33, Jun. 2013

[15] Li, Y. and Bhanu, B., “Utility-based camera assignment in a video network: A game theoretic framework”, IEEE Sensors Journal, 11 (3): 676-687, Mar. 2011

[16] Hua, G.; Fu, Y.; Turk, M.; Pollefeys, M. and Zhang, Z., "Introduction to the Special Issue on Mobile Vision", Int. Journal of Computer Vision, 96(3):277-279, Feb. 2012.

Juan C. SanMiguel is a postdoctoral researcher at Queen Mary University of London, UK. He received a Ph.D. in Electrical Engineering and Computer Science from the University Autonoma of Madrid, Spain. His research interests include multi-camera activity understanding, distributed target tracking and performance evaluation. Contact him at juan.carlos.sanmiguel@qmul.ac.uk

Christian Micheloni is Assistant Professor of image and video processing at the University of Udine, Italy. He received a Ph.D. in Computer Science from the University of Udine, Italy. His research interests include active vision for self-reconfiguring camera networks, machine learning and person re-identification. Contact him at christian.micheloni@uniud.it.

Karen Shoop is a faculty member at Queen Mary University of London, UK. She received a Ph.D. in Electronic Engineering from Queen Mary University of London. Her research focuses on distributed decision making. Contact her at karen.shoop@qmul.ac.uk

Gian Luca Foresti is Professor of Computer Science at University of Udine, where he is the Deputy-Director of the Department of Mathematics and Computer Science and Director of the Artificial Vision and Real Time Systems (AVIRES) Lab. He received a Ph.D. in Electrical Engineering from University of Genoa, Italy. His research interests involve active vision, audio-video processing, multi sensor data and information fusion, machine learning and pattern recognition. Contact him at gianluca.foresti@uniud.it.

Andrea Cavallaro is Professor of Multimedia Signal Processing and Director of the Centre for Intelligent Sensing at Queen Mary University of London, UK. He received a Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne. His research interests include smart camera networks, target detection and tracking, behaviour and identity recognition. Contact him at a.cavallaro@qmul.ac.uk.