

Self-aware and Self-expressive Camera Networks

Bernhard Rinner, Lukas Esterle, Jennifer Simonjan, Georg Nebehay,
Roman Pflugfelder, Peter R. Lewis and Gustavo Fernández Domínguez

Abstract—Recent advances have enabled a dramatic change in camera networks. Smart cameras perform image analysis onboard, adapt their algorithms in response to changes in their environments and collaborate with other cameras in order to analyze the dynamic behavior of objects in partly unknown environments. A fixed configuration is infeasible to manage the trade-off among performance, flexibility, resources and reliability in such camera networks. We adopt the concepts of self-awareness and self-expression for autonomous monitoring of the state and progress of the camera network and adaptation of its behavior to changing conditions. We describe the building blocks for self-aware camera networks and demonstrate the key characteristics in simulation and a real camera network.

Index Terms—Self-awareness, self-expression, smart camera networks, real-time adaptability.

I. INTRODUCTION

Camera networks are now ubiquitous and have applications in security, disaster response, environmental monitoring and smart environments, among others. *Smart* camera networks have emerged recently by bringing together advances in computer vision, embedded computing, image sensors and networks. They are real-time, distributed, embedded systems that perform computer vision tasks using multiple cameras [1].

In order to provide in-network processing of captured data, smart camera networks have to deal with various challenges. Cameras analyze the highly dynamic behavior of objects, are deployed in partly unknown environments and cooperate with neighboring cameras on demand. Future camera networks should be able to achieve advanced levels of autonomous behavior to adapt themselves at runtime and learn behaviors appropriate to changing conditions. A particular challenge is to manage the trade-off of conflicting objectives such as high performance, low resource consumption and high reliability. A fixed configuration of the camera network is infeasible to overcome these challenges [2].

As a successful alternative we adopt the concepts of *self-awareness* (*SA*) and *self-expression* (*SE*), translating them to computational analogies and applying them to camera networks. Both concepts are new to the domains of computing and networking. Self-awareness refers to the ability of a system to obtain and maintain knowledge about its state, behavior and progress, enabling self-expression, the generation of autonomous behavior based on such self-awareness. Together, self-awareness and self-expression support effective and autonomous adaptation of behavior to changing conditions. Learning models online of the camera's state and context

B. Rinner, L. Esterle, and J. Simonjan are with the Alpen-Adria-Universität Klagenfurt, Austria.

G. Nebehay, R. Pflugfelder and G. Fernández Domínguez are with the Austrian Institute of Technology, Austria.

P.R. Lewis is with Aston University, U.K.

as well as decentralized decision making in the network are the fundamental techniques for achieving *SA* and *SE*. The building blocks and their interaction are explained in order to build a computationally self-aware and self-expressive camera network. Its features and capabilities are shown practically with a distributed multi-camera tracking application as an example.

AUTONOMOUS MULTI-CAMERA COORDINATION

Coordination is a fundamental problem in camera networks with the key objective to dynamically assign actions to cameras in order to improve the performance of the network tasks. These tasks may include coverage optimization, pan-tilt-zoom (PTZ) control and tracking and can be found in a myriad of different applications such as surveillance, transportation, security and activity recognition [a, b]. The assigned actions determine the placing and orientation of (mobile or PTZ) cameras, the sensing capabilities such as resolution and frame rate, the data processing and the data communication. Researchers have adopted different approaches to coordinate multiple active cameras such as control theory, game theory, state machines, multi-agent systems, probabilistic approaches, and many other ad-hoc approaches as well [c].

In *centralized* coordination the decision making is performed at a single node that receives data from each camera in the network. It introduces a significant computation and communication load but can achieve a high task performance due to the availability of the entire network information in a single node. In *decentralized* coordination the decision making and the required data from the cameras are distributed among the network. In *autonomous* coordination each camera individually decides what action to take based on its local assessment which may include information from other cameras.

The goal of multi-camera tracking is to detect, localize and track moving objects such as pedestrians or vehicles within the fields of views (FOV) of all cameras. Deciding which camera is responsible for tracking a specific object represents a typical coordination problem [d]. In centralized coordination the cameras send the traces of the objects within their FOV to a central node which then selects the best trace. In autonomous coordination each camera decides by its own when and to whom to handover the tracking responsibility. Deriving the handover decision based on incomplete information and with camera's limited resources is a fundamental challenge. *SA* and *SE*

helps to overcome some of the difficulties of coordinating the tracking responsibility and is able to achieve robust, flexible and scalable multi-camera tracking control with low computation and communication overhead.

References

- [a] R. J. Radke, "A Survey of Distributed Computer Vision Algorithms". in *Handbook of Ambient Intelligence and Smart Environments*. Springer, 2009, pp. 35–55.
- [b] B. Bhanu, C. V. Ravishankar, A. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds., *Distributed Video Sensor Networks*. Springer, 2011.
- [c] S. Chen, Y. Li, and N. M. Kwok, "Active Vision in Robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343-1377, 2011.
- [d] M. Taj and A. Cavallaro, "Distributed and decentralized multi-camera tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 46–58, May 2011.

II. SELF-AWARENESS AND SELF-EXPRESSION

In common with many other emerging computational systems, smart camera networks face the challenge of operating in a complex interconnected world, where they interact with people and each other in ways which are difficult to understand and predict. In order to meet this challenge, such systems must possess an increased level of awareness, both of the world around them and of themselves. But we must not only attempt to replicate the self-awareness capabilities of humans in computers; there will be important differences. Instead, the notion of *computational self-awareness* is being developed [3], inspired by concepts of human self-awareness. Concepts of self-awareness need to be translated from fields such as psychology and applied to the very different domain of computing. Systems possessing such computational self-awareness will be able to continuously learn and adapt during their lifetime. They will build up an awareness of themselves and of their own experience of the world which they inhabit. And they will also need to be aware of the way in which they themselves are aware of these things.

The study of self-awareness emerged as a field within psychology in the 1960's. Morin [4] defines self-awareness as "*the capacity to become the object of one's own attention*". A prerequisite for this is the capability to monitor or observe oneself. Our work is concerned with taking inspiration from self-awareness theories in order to better engineer computational systems. We therefore aim at bridging the gap between psychology [5] and engineering (e.g., [3]). As part of this, we developed an engineering methodology for self-aware and self-expressive computing systems, including a general reference architectural framework, able to be adapted to a wide range of applications. In Section III, we describe our work to apply this methodology, in particular concretizing the reference architectural framework for the autonomous coordination of nodes in a multi-camera network.

This framework embodies various concepts from self-awareness theory. One important one is that self-awareness can be an emergent property of collective systems, even when there is no single component with a global awareness of the whole system [6]. This is a key observation which can contribute to

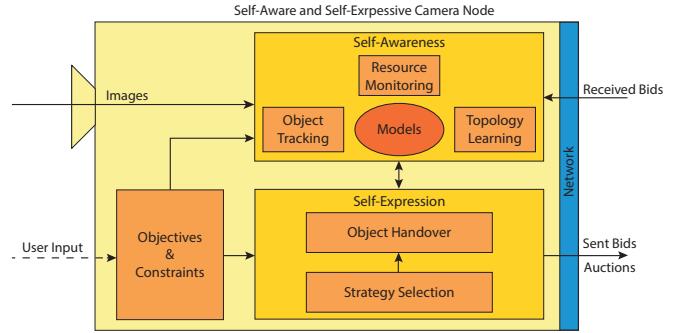


Fig. 1: The architecture of a self-aware and self-expressive camera node composed by six building blocks.

the design of self-aware systems: one need not require that a self-aware system has a node with global knowledge. Also in this framework, the term *self-expression* is used to refer to behavior based on self-awareness, and can also be considered both at the node level [3] and collective level [7].

There are several clusters of research in computer science and engineering which have used the term self-awareness explicitly [5]. However, our work represents the first holistic framework for describing and benchmarking the self-awareness properties of computational systems, and the benefits that computational self-awareness can bring. While we use a specific application context, as has been done in related work [8], [9], our work also generalizes this idea. Indeed, this article demonstrates the notion of computational self-awareness in our work on smart camera networks.

III. APPLICATION DESIGN WITH SA&SE BUILDING BLOCKS

We apply our generic architectural patterns [3] to develop a flexible SA&SE multi-camera application. The key approach here is to instantiate dedicated software components and their interactions. These components serve as distinct building blocks for the concrete multi-camera tracking application in our case. Fig. 1 depicts the overall architecture of a self-aware and self-expressive camera node contributing to the multi-camera tracking application. Each building block has specific objectives and interacts with other blocks in the network. Self-awareness is realized by individual blocks for *Object Tracking*, *Resource Monitoring*, and *Topology Learning*. Instead of relying on predefined knowledge and rules, these blocks utilize online learning and maintain models for the camera's state and context. These models serve as input to self-expression which is composed of the *Object Handover* and the *Strategy Selection* blocks. Finally, the *Objectives & Constraints* block represents the camera's objectives and resource constraints, where both have a strong influence on the other blocks.

We can compose a truly decentralized, self-aware and self-expressive camera network by aggregating our camera nodes. The building blocks of each node can be implemented by diverse algorithms from computer vision, online learning and decision making. However, resource awareness was one guiding principle for our design. Thus, all building blocks

must be able to execute in real-time on our resource limited embedded camera platforms.

A. Object Tracking

This building block employs a simple appearance-based approach for visual object tracking. The chosen method fulfills the cameras' computational resource and real-time requirements and achieves for the given test environment an acceptable level of robustness against dropped frames, occlusions and disappearance of objects.

In a first step, we identify foreground pixels in each camera by comparing the camera image to a background image that each camera learns individually about its field of view. This approach assumes in its simplest form static cameras. Foreground constitutes moving objects that comprises objects of interest that should be tracked. These foreground pixels are then grouped into candidate objects based on their connectedness. In a second step, we perform an association of these candidate objects to a template database that contains the objects of interest. To this end, we employ a measure of similarity according to [10] that we also interpret as the confidence in the validity of the association.

It is important to note that the approach does not aim to compete with the state of the art in visual multi-camera object tracking targeting general environmental conditions. Instead, the method serves as an exemplary implementation of a self-aware object tracking building block, sufficient to validate the SA&SE framework.

B. Object Handover

To coordinate the object tracking responsibilities in the camera network, we apply a novel market-based handover approach [11]. Here, the cameras treat object tracking responsibilities as goods, providing some utility over time. The cameras can decide in a self-expressive manner on their own when to "sell" tracking responsibilities to other cameras using virtual auctions. Whenever a camera decides to sell an object, it initiates an auction for this particular object by transferring an object description to the other cameras. The receiving cameras search their own FOV for the object and value the object based on detection confidence and visibility. These cameras return their valuation as a bid. The auctioneering camera selects the highest bidder and transfers the tracking responsibility. We use the Vickrey auction mechanism, which sells the good to the highest bidder for the second highest price, to make truthful bidding the dominant strategy among the participating cameras. Fig. 2 illustrates the key steps of the market-based handover which is a fully decentralized mechanism relying only on autonomous decisions of cameras.

An important question for the selling camera is to whom to send the auction invitations. Without any a priori knowledge about the network topology, invitations could be broadcasted to all cameras. By following this strategy the "best" camera for taking over the tracking responsibility will receive an invitation (and may respond with the highest bid). However, the broadcast strategy causes a significant communication and computation load, i.e., because each camera has to perform an object detection after receiving the invitation.

C. Topology Learning

If the auctioneering cameras are aware of the potentially "best" cameras in their neighborhood, this knowledge can be exploited to significantly reduce the overhead. Such topological information can be initially assigned to the cameras or computed by means of multi-camera calibration during the deployment of the camera network. However to improve self-awareness, we learn the topology by observing the bidding behavior of cameras over time. Each camera individually keeps track of their local neighbors and uses artificial pheromones to express the likelihood of a handover to that camera. Whenever a handover has taken place, the artificial pheromone to the succeeding camera is strengthened. If no trading occurs, pheromones evaporate over time. This mechanism enables each camera to deal with network uncertainties and to adapt to changes in their neighborhood topology, e.g., caused by adding, removing or failure of cameras or changes in the movement pattern of the objects.

We exploit the learned neighborhood topology by three different communication strategies for the handover: broadcast auctions to all cameras (BROADCAST), a smooth probabilistic multicast (SMOOTH) and a threshold-based probabilistic multicast (STEP) [11]. The SMOOTH strategy sends auction invitations to all neighbors with probability normalized to the current pheromone level. The STEP strategy sends invitations to all neighbors with pheromone level above a certain threshold and to neighbors below the threshold with some (low) probability. We further distinguish whether to send out invitations at regular intervals (ACTIVE) or only when the object is about to leave the FOV (PASSIVE). We then end up with six different self-expressive handover strategies by combining the approach on whom to invite and when to send out the invitations. Obviously, the selected strategy influences the achieved tracking utility as well as communication and computational overhead.

D. Strategy Selection

While the six handover strategies allow to trade off communication overhead against tracking utility and hence influence the behavior of the network, selecting a strategy is a difficult decision. The performance of each strategy strongly depends on factors such as the placement of the cameras, the movement patters of the objects and the object tracking algorithm. In principle, we can follow three approaches for strategy selection: (i) a homogeneous selection, where all cameras select the same strategy at deployment time, (ii) a heterogeneous selection, where the cameras can select their strategy individually at deployment time, and (iii) a dynamic selection, where each camera can select its strategy during runtime.

We use online learning algorithms, specifically multi-armed bandit problem solvers within each camera to learn the appropriate strategy for each node during runtime. Bandit solvers balance exploitation behavior, where a camera achieves high performance by using its currently known best strategy, with exploration, where the camera explores the effect of using other strategies to build up its knowledge [12]. We use standard bandit-solvers from the literature, namely Softmax,

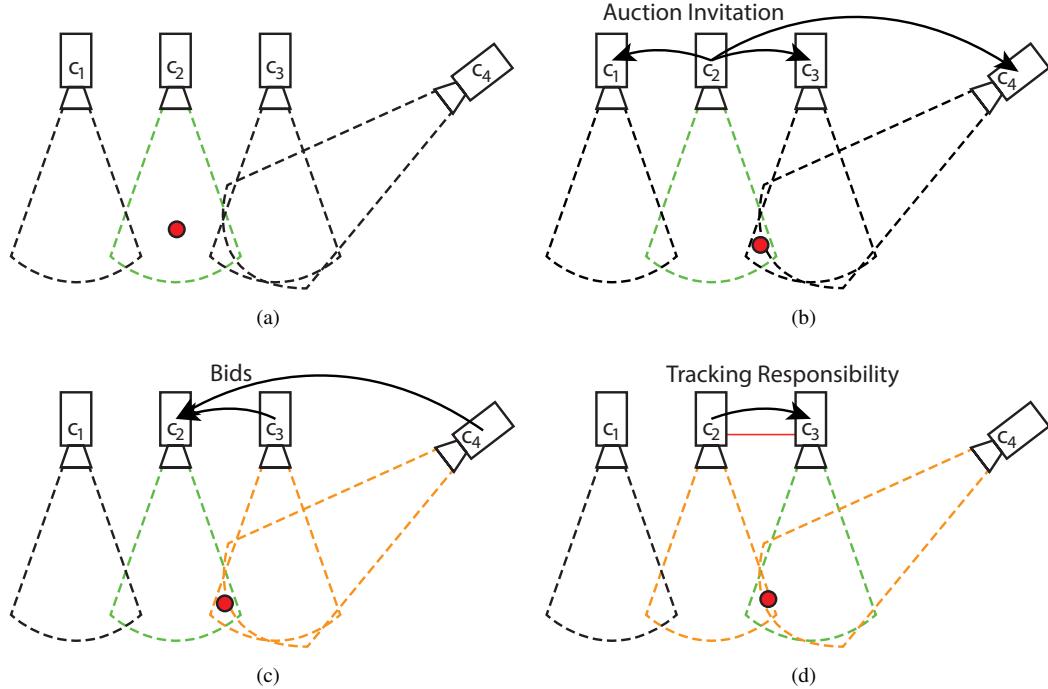


Fig. 2: Illustration of the market-based handover approach: An object of interest is tracked by camera c_2 indicated by the green FOV (a). c_2 initiates an auction (black arrows) for the object when it is about to leave the FOV (b). Cameras c_3 and c_4 detect the object in their FOVs (orange) and return bids (black arrows) based on their object valuation to c_2 (c). c_3 has offered the highest bid and wins the auction (d). The tracking responsibility is transferred from c_2 to c_3 (black arrow) and an artificial pheromone is added (indicated as red line between c_2 and c_3).

Epsilon-Greedy and UCB1. Dynamic strategy selection leads to another level of self-aware and self-expressive behavior of the camera network and is able to achieve a more Pareto efficient global performance than with any static selection.

E. Resource Monitoring

Resource monitoring is an important aspect of computational self-awareness, and its main objective is to observe the available resources on the camera nodes. The monitored data is further used to build up models of resource consumption for each task a camera is capable of performing. This information allows a self-expressive block (e.g., strategy selection) to reason not only about the performance of each task but also about its respective resource consumption. In our network, we currently monitor required processing power, available and allocated memory, and network traffic.

F. Constraints and Objectives

Each camera has some constraints and objectives which need to be considered for self-aware and self-expressive operation. In our system constraints specify some limitations of the available resources (processing, memory and networking) and help to decide on whether to bid for an object. On the other hand, objectives are related to the behavior of the cameras and specify, for example, some quality of service parameters or specific tasks the cameras should achieve.

IV. SMART CAMERA NETWORK

A. Experimental Study: Camera Network Setup

Fig. 3 depicts the setup of the smart camera network used for our experimental study. Four cameras (1–4) are mounted in a laboratory room with overlapping FOVs. Cameras 5 and 6 are placed in the corridor and lounge area, respectively. Our heterogeneous network is composed by different hardware platforms. Cameras 1 to 4 are equipped with Atom processors and connected via wired Ethernet. Cameras 5 and 6 are based on Pandaboard equipped with ARM processors and use WiFi for communication. All cameras run standard Linux and a distributed publish-subscribe middleware system [13] to provide a flexible software platform for software development. The building blocks have been implemented in C++ and C# using the middleware services for communication and control.

B. Tracking Results

For the evaluation of our object tracking block we use the smart camera network with various people walking through the indoor environment (as depicted in Fig. 3). We apply state-of-the-art metrics (e.g., [14]) for the evaluation. Fig. 4(a) summarizes the results of a typical scenario of concurrently tracking three selected persons independently walking around in the indoor environment for around 120 seconds. During this scenario, the persons were not continuously visible to all cameras; at some points, participants were not seen by any camera at all. The cameras mounted in the laboratory achieved

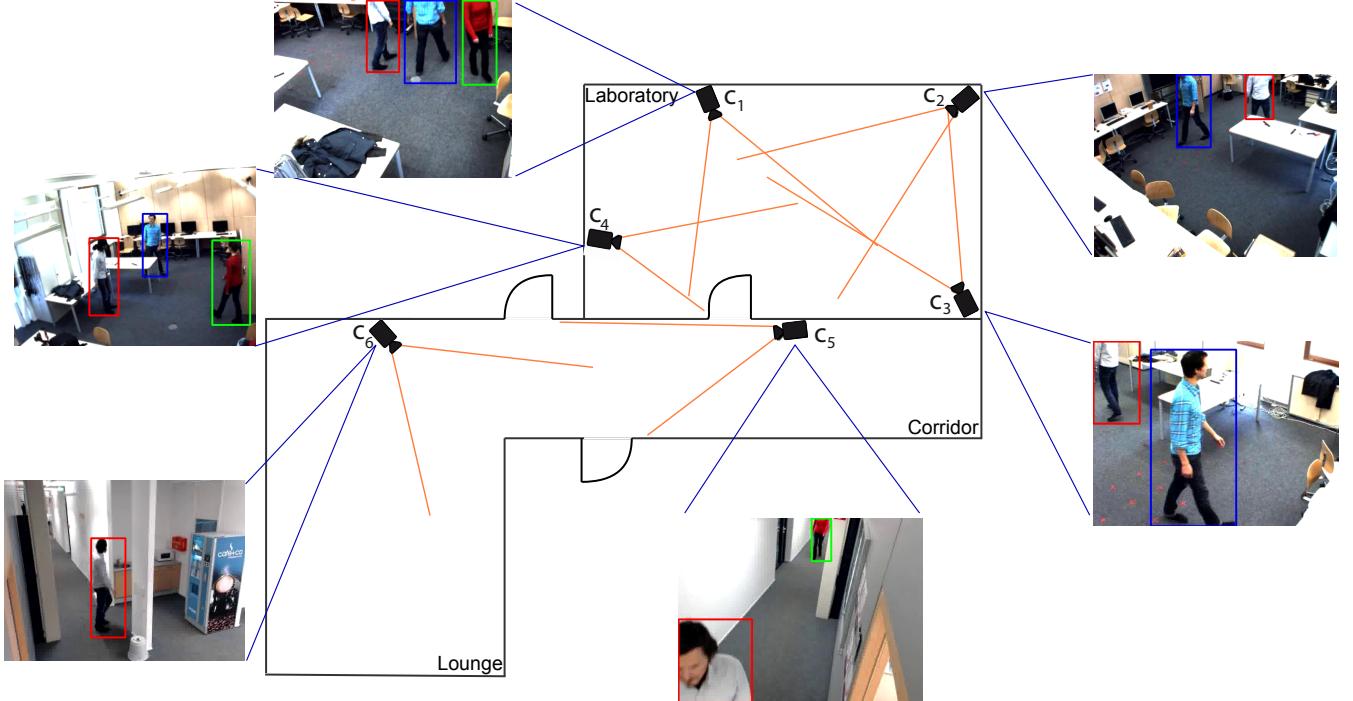


Fig. 3: The illustration above shows the smart camera network composed of six cameras deployed in an indoor environment. The cameras are depicted by a black camera symbol and their FOVs are indicated by orange lines. Snapshots of six cameras are indicated by blue lines. Those images show the status of object tracking over time; three people are tracked by the system marked by red, blue and green bounding boxes.

better detection and tracking results. The performance of cameras 5 and 6 degraded slightly due to moderate changes in lightning and object appearance. Overall, the experimental results show sufficient performance of the object tracker for correct object handover in the given test environment.

C. Topology Learning

Through our self-aware topology learning block each individual camera builds up a local neighborhood relationship graph. Fig. 4(b) shows the aggregated graph for the entire network after a test run in our smart camera network. The thickness of the red lines indicate the strength of the artificial pheromone deposit on this link which corresponds to the probability of an object transiting between the connected cameras. Initially, links are created between the camera in the lounge (camera 6) and those in the laboratory (cameras 1-4) due to misdetections of camera 5. Through the evaporation of the artificial pheromones, cameras can not only deal with errors induced by the tracking block but also with changes in the topology due to hardware errors or vandalism. Over time these links evaporate and a qualitatively correct neighborhood graph emerges.

D. Communication and Utility Trade-off

We evaluated the effect of the handover strategy on the overall tracking utility and communication overhead. Figure 5(a) depicts the trade-off of utility and communication for homogeneous strategy selection in our smart camera

network. The utility is defined as the aggregated tracking utility of all cameras, and the communication is defined as the number of all sent auction messages during the entire tracking operation. Both utility and communication values are normalized by those from the ACTIVE BROADCAST strategy. The six strategies result in six different trade-offs for utility and communication.

Figure 5(b) compares the achieved trade-off for homogeneous, heterogeneous and dynamic strategy selection in our CamSim¹ simulation tool [12]. Obviously, heterogeneous selection (black crosses) leads to many more outcomes in the objective space. The extension of the Pareto efficient frontier brought about by heterogeneity is also apparent. However, it is also clear that the outcomes of many heterogeneous strategies are dominated, and many are strictly worse than the original outcomes from the homogeneous strategies. We can clearly see the benefit of self-expressive behavior. Dynamic strategy selection, here implemented using reinforcement learning (colored symbols), is able to outperform the static (homogeneous and heterogeneous) strategies and to extend the Pareto front.

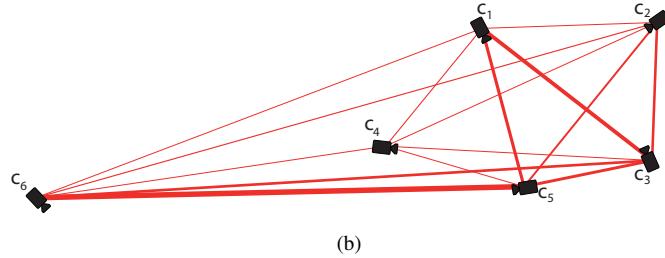
V. CONCLUSION

As demonstrated in this paper, self-awareness and self-expression are fundamental concepts for developing camera networks capable of learning and maintaining their topology, distributedly performing object detection and tracking handover as well as autonomously selecting strategies to achieve

¹<http://www.epics-project.eu/CamSim/>

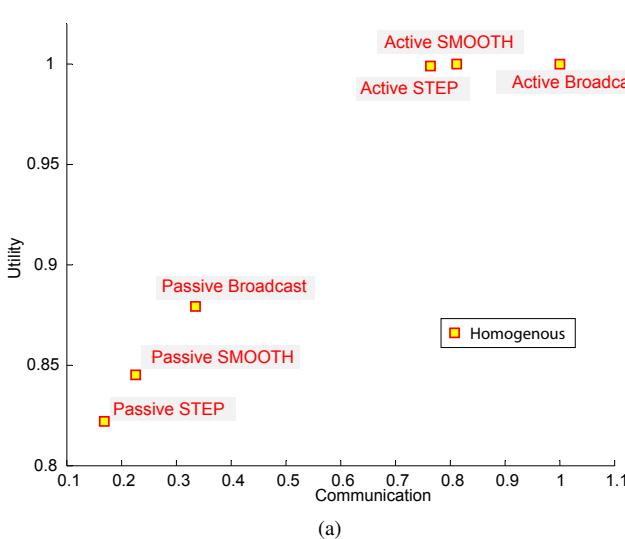
Camera number	Object detection: Sensitivity	Object tracking: CDT
1	0.78	3
2	0.73	4
3	0.87	3
4	0.76	3
5	0.48	2
6	0.54	2

(a)

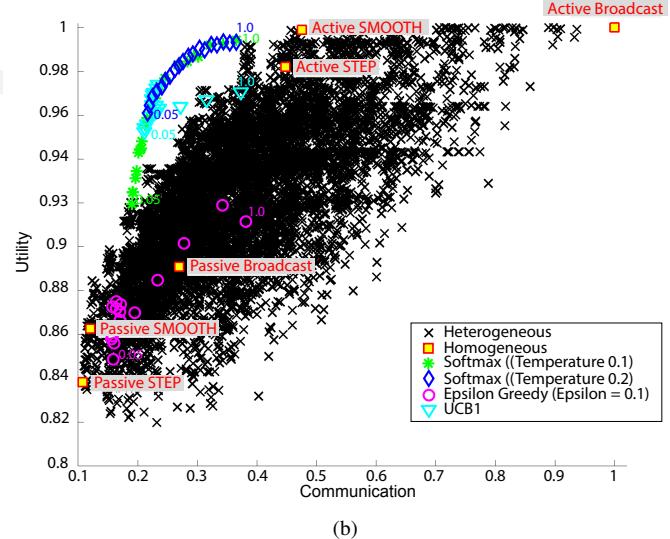


(b)

Fig. 4: Results for object tracking and topology learning for concurrently tracking three selected persons for around 120 seconds. The table (a) shows the sensitivity of object detection and the correct detected tracks (CDT) for each camera. The graph (b) shows the learned topology after 60 seconds by exploiting the trading behavior. The thickness of the red line indicates the pheromone level of the link.



(a)



(b)

Fig. 5: Performance for two exemplary scenarios from our smart camera network (a) and our simulation environment (b) showing homogeneous (red and yellow squares), heterogeneous (black crosses), and dynamically learned strategy assignments (colored symbols, representing different reinforcement learning strategies evaluated). The results have been normalized by the maximum value of the ACTIVE BROADCAST strategy and are averages over 30 runs with 1000 time steps each [12].

more Pareto efficient outcomes. The entire processing is encapsulated into six building blocks embedded onto resource-limited smart camera nodes and aggregated into a completely decentralized and thus scalable network. However, computational self-awareness and self-expression is not limited to camera networks. In fact, we are confident that SA and SE could serve as an enabling technology for future systems and networks meeting a multitude of requirements with respect to functionality, flexibility, performance, resource usage, costs, reliability and safety.

ACKNOWLEDGMENT

This work has been partially supported by the EU FP7 program under grant agreement no 257906 <http://www.epics-project.eu/>.

REFERENCES

- [1] M. Reisslein, B. Rinner, and A. Roy-Chowdhury, "Smart Camera Networks," *Computer*, vol. 47, no. 5, pp. 26–28, May 2014.

- [2] J. C. SanMiguel, K. Shoop, A. Cavallaro, C. Micheloni, and G. L. Foresti, "Self-Reconfigurable Smart Camera Networks," *Computer*, vol. 47, no. 5, pp. 67–73, May 2014.
- [3] F. Faniyi, P. R. Lewis, R. Bahsoon, and X. Yao, "Architecting self-aware software systems," in *Proc. Working IEEE/IFIP Conference on Software Architecture (WICSA) 2014*. IEEE, 2014, pp. 91–94.
- [4] A. Morin, "Levels of consciousness and self-awareness : A comparison and integration of various neurocognitive views," *Cons. and Cog.*, vol. 15, no. 2, pp. 358–71, 2006.
- [5] P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao, "A survey of self-awareness and its application in computing systems," in *Proc. Fifth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*. IEEE Computer Society Press, 2011, pp. 102–107.
- [6] M. Mitchell, "Self-awareness and control in decentralized systems," in *Working Papers of the AAAI 2005 Spring Symposium on Metacognition in Computation*. AAAI Press, 2005, pp. 80–85.
- [7] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, and M. Puviani, "On self-adaptation, self-expression, and self-awareness in autonomic service component ensembles," in *Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems, SASOW 2011, Ann Arbor, MI, USA, October 3-7, 2011, Workshops Proceedings*, 2011, pp. 108–113. [Online]. Available: <http://dx.doi.org/10.1109/SASOW.2011.24>

- [8] M. Cox, "Metacognition in computation: A selected research review," *Artificial Intelligence*, vol. 169, no. 2, pp. 104–141, 2005.
- [9] A. Agarwal, J. Miller, J. Eastep, D. Wentzlaaff, and H. Kasture, "Self-aware computing," MIT, Tech. Rep. AFRL-RI-RS-TR-2009-161, 2009.
- [10] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, 2003.
- [11] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, "Socio-economic vision graph generation and handover in distributed smart camera networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, pp. 20:1–20:24, 2014.
- [12] P. R. Lewis, L. Esterle, A. Chandra, B. Rinner, J. Torresen, and X. Yao, "Static, dynamic and adaptive heterogeneity in distributed smart camera networks," *ACM Transactions on Adaptive and Autonomous Systems*, p. 30, 2015, in print.
- [13] B. Dieber, J. Simonjan, L. Esterle, G. Nebelhay, R. Pflugfelder, G. Fernandez, and B. Rinner, "Ella: Middleware for Multi-camera Surveillance in Heterogeneous Visual Sensor Networks," in *Proceedings of the Seventh ACM/IEEE International Conference on Distributed Smart Cameras*, Palm Springs, USA, 2013.
- [14] A. Baumann, M. Boltz, J. Ebling, M. Koenig, H. S. Loos, M. Merkel, W. Niem, J. K. Warzelhan, and J. Yu, "A review and comparison of measures for automatic video surveillance systems," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 4, 2008.