A Socio-Economic Approach to Online Vision Graph Generation and Handover in Distributed Smart Camera Networks

Lukas Esterle[†], Peter R. Lewis*, Marcin Bogdanski*, Bernhard Rinner[†] and Xin Yao*
* School of Computer Science
University of Birmingham, UK
{mxb039—p.r.lewis—x.yao}@cs.bham.ac.uk
[†] Institute of Networked and Embedded Systems
Klagenfurt University, Austria
{lukas.esterle—bernhard.rinner}@aau.at

Abstract—In this paper we propose an approach based on self-interested autonomous cameras, which exchange responsibility for tracking objects in a market mechanism, in order to maximise their own utility. A novel ant-colony inspired mechanism is used to grow the vision graph during runtime, which may then be used to optimise communication between cameras. The key benefits of our completely decentralised approach are on the one hand generating the vision graph online which permits the addition and removal cameras to the network during runtime and on the other hand relying only on local information, increasing the robustness of the system. Since our market-based approach does not rely on *a priori* topology information, the need for any multi-camera calibration can be avoided.

Index Terms—Smart camera networks; multi-camera tracking; market-based control; topology identification; ant algorithms.

I. INTRODUCTION

Object detection, tracking and activity recognition are important image analysis tasks in multi-camera networks. Many approaches have been proposed in the literature over the last few years (e.g., [1]), most of which rely on either some *a priori* knowledge about the network topology or centralised algorithms. Recently, tracking applications have been developed on smart camera networks where the processing is distributed among the camera nodes (e.g., [2], [3]). While these distributed approaches apply different control strategies for managing the tracking responsibilities, they rely on topology knowledge and/or require iterative information exchange among the cameras. Our novel approach overcomes these limitations and is able to achieve robust, flexible and scalable multi-camera control with low computation and communication overhead.

In this paper we present a socio-economic approach for online vision graph estimation and tracking handover in smart camera networks. Self-interested, autonomous cameras exchange responsibility for tracking objects in a market mechanism in order to maximise their own utility. When handover is required, an auction is initiated and cameras observing the object bid to track it. By observing the trading behaviour we learn the visual neighbourhood relations in the camera network and generate the vision graph of the network online.

We apply ant-colony inspired pheromones to grow this vision graph during runtime, which is then used to optimise the communication effort among the cameras.

Our approach offers several benefits: It is fully decentralised, requires only the exchange of local information, is computationally inexpensive, supports online processing and does not require any *a priori* knowledge about the camera network or objects of interest. As a result, our proposed approach is highly robust and works in dynamic environments where a camera can be added or removed from the network at any time without affecting any other parts of the network.

The main contributions of this research include the application of market-based principles to coordinate the handover in multi-object, multi-camera tracking, the online generation of the vision graph, the exploration of the trade-off between trading and communication effort and the use of ant inspired artificial pheromones to direct marketing effort in order efficiently manage this trade-off. Our novel approach has been evaluated by a simulation study.

The remainder of this paper is structured as follows. Section II provides a background to distributed smart cameras and the problems of handover and vision graph generation. Section III describes our approach, which makes use of local utility functions to aid the decision process, a market mechanism to allow cameras to hand over objects in order to maximise their utility, and pheromone-based rules for automatic vision graph generation. Section IV describes our experimental study and summarises key results. Finally, section V discusses the implications of this work and identifies areas for further study.

II. BACKGROUND AND PROBLEM DEFINITION

A. Object Tracking with Distributed Smart Cameras

In multi-camera tracking, the fundamental tasks of single camera object detection and tracking must be expanded by a *handover* mechanism which refers to finding the next camera to see the target object once it leaves the FOV of the current camera [4]. Various mechanisms have been proposed to solve the handover problem. These mechanisms vary in the required

assumptions of the camera network, the distribution of data and processing as well as the required resources [2].

In smart camera networks much effort is put on distributed and resource-aware handover mechanisms due to increased scalability and robustness [5]. One of the first autonomous handover approaches on smart cameras has been presented by Quaritsch et al. [6]. This approach relies on a static and *a priori* known vision graph. The neighbourhood structure is encoded in so-called migration regions which assign neighbouring cameras to specific areas in the cameras' FOVs. Whenever a target object enters a migration region, a tracker is then started on the neighbouring camera(s). Li and Bhanu [7] present a game-theoretic camera handover; as in our market-based approach, the next camera selection is based on a utility function which is computed by a bargaining approach among cameras "seeing" the tracking object. However, the bargaining requires several iterations among the involved cameras. The algorithm has been implemented in a centralised way, which does not provide good scalability or robustness. Qureshi and Terzopoulos [8] introduce a distributed camera coalition formation scheme for perceptive scene coverage and persistent surveillance by smart camera sensor networks. They demonstrate the camera selection and handover in a virtual environment.

The topology of a camera network is important for a number of higher-level functions such as multi-camera tracking, target following or camera placement optimization [9]. Observing moving objects within the network is often used to learn the topology over time (e.g., [10], [11], [12]). The estimation approaches vary in the topology assumptions (e.g., overlapping or non-overlapping FOVs), topology modelling and the extraction relevant information from individual camera views. In our approach, the trading of an object provides an implicit snapshot on the network topology, i.e., the "selling" and the "buying" by cameras represent a neighbourhood relationship.

B. Problem Formulation

Our primary objective is to track up to m distinct objects within the aggregated FOV of n fixed cameras in the network. Although an object might be "seen" by several cameras, a single camera is responsible for tracking this object. Thus, the network must distribute the tracking responsibility for at most m objects among the n cameras at any time. This tracking responsibility of camera i for object j can be expressed by j being a member of the set of objects "owned" by i, which we denote as O_i. When we say that camera i owns object j, we mean that it is responsible for tracking it, has the right to track it, and that it may sell it to other cameras. However, since our cameras are controlled by autonomous software agents, they make independent decisions about which object(s) in O_i to attempt to track. Camera i's decision to attempt to track object j is expressed as the binary function $\phi_i(j)$.

We assume that a camera can track up to k objects simultaneously without exceeding its resource limitations and hence without any degradation of the tracking performance. In our analysis we assume that the number of objects tracked by a camera is less than k. Thus, a conservative limit on the number

of objects would be $m \leq k$. When camera i attempts to track object j ($\phi_i(j) = 1$), a tracking module is initialized with a description of that object and is detecting and tracking the object within the camera's FOV. The tracking performance depends on various factors such as object descriptor, distance, orientation, partial occlusion and so on. In our study we simplify single camera tracking and subsume all these factors in a visibility parameter v_j which is determined by the distance and angle of the observed object to the observing camera. The tracking performance is estimated by a confidence value c_j. Both values c_j and v_j are between 0 and 1 as soon as the observed object is within the FOV of a camera, 0 otherwise.

III. A Socio-Economic Inspired Approach

The approach presented in this paper takes inspiration from both social and economic systems, and is based on two distinct concepts. Firstly, the allocation of objects to cameras makes use of a market-based approach, similarly to those described in [13]. Secondly, a pheromone-based mechanism inspired by social interactions in ant colonies is used to build the vision graph online, based on trading activity. This is then used to determine communication between cameras. The ant inspired approach is similar to ant colony optimisation [14], however our novel use of artificial pheromones to enable targeted marketing is a previously unexplored idea, which enables the efficient management of the trade-off between communication and utility. Additionally, the approach is robust to dynamics and inherently scalable. We therefore believe it has significant potential for a range of decentralised applications, of which distributed smart cameras are one example. The socio-economic algorithm is implemented locally in each camera.

A. Utility and Market Mechanism

For a given camera i and its set of owned objects O_i, we say that the instantaneous utility of camera i is given by

$$U_i(O_i, p, r) = \sum_{j \in O_i} u_i(j) - p + r \qquad (1)$$

$$= \sum_{j \in O_i} [c_j \cdot v_j \cdot \phi_i(j)] - p + r \qquad (2)$$

where $\phi_i : O_i \rightarrow \{0, 1\}$ and is 1 if camera i attempts to track object j and 0 otherwise. In addition to utility earned by tracking objects, a camera b may make a payment to another camera s in order to "buy" the right to track an object from that camera. This requires that the "selling" camera s already itself owns the object. If an exchange is agreed, then the object is removed from O_s and added to O_b. p denotes the sum of all payments made in trades in that iteration, and r conversely denotes the sum of all payments received.

To facilitate the exchange of objects, we propose the use of Vickrey auctions [15] hosted by the selling camera. The Vickrey auction, also known as the *second price sealed bid auction* is a single sided auction where bidders make one sealed bid for a single item. The auctioneer awards the item to the highest bidder, but at the price bid by the second highest bidder. The advantage of the Vickrey auction from an implementation

perspective is that it has a dominant strategy for bidders: to bid one's truthful valuation, regardless of the strategies of the other bidders. In contrast with other mechanisms, this removes the need for cameras to possess adaptive bidding strategies, or be required to learn a high performing context-dependent strategy. In common with other market-based control systems (e.g. [16]), currency is an artificial construct used as a tool for system management; no real money is used.

Therefore, in our model each camera, in the absence of any vision graph information, broadcasts information about the objects it is currently tracking at in order to solicit bids. Each camera i, upon observing such a broadcast, determines the likely value of having the right to track the object (i.e. having it in O_i) and if this value is positive, subsequently responds privately to the broadcasting camera with its bid. Since we use a Vickrey auction, each camera may place only one bid and the dominant strategy of each camera is to set this bid equal to its truthful valuation of the object in terms of its contribution to the camera's utility (see equation 2).

B. Pheromone-based Vision Graph Generation

One of the key advantages of our approach is that it does not require the vision graph to be known *a priori*, since cameras' relative utility is used to determine which camera the object should be handed over to. However, the broadcast method used to support this decision is inefficient in terms of communication overhead. For this reason, we use a pheromone-based method for building the vision graph online, from the trading activity occurring in the market. As the cameras learn the vision graph, they may scale down the amount of communication while still achieving high utility, by announcing their objects only to cameras which are their neighbours in the vision graph.

This use of artificial pheromones, built from previous trading activity to guide future marketing activity, is a novel and highly useful method to achieve efficient outcomes in the trade-off between communication and performance. Since the pheromones both are reinforced and evaporate over time, changes in the topology of the underlying vision graph during runtime can be adapted to in a robust manner, and the loss of individual cameras does not affect the wider system. Since marketing communication can be concentrated on only those small number of relevant camera nodes, our socio-economic approach allows significantly improved scalability.

In this model, vision graph information is distributed and local information stored in cameras. We therefore define for each camera i an adjacency list, E_i, the set of all links (or edges) local to that camera. Each element of E_i is the tuple (i, x, τ_{ix}), where x is another camera in the network and τ_{ix} is the strength of the link from camera i to camera x. Each camera is initialised with an adjacency list containing tuples from itself to all other cameras in the network, each tuple with a strength value $\tau_{ix} = 0$ for all x. Subsequently, each time camera i successfully sells an object to camera x, the corresponding strength value is increased by a value Δ. In Ant Colony Optimisation, the value of Δ is often determined by the

problem's properties. Although we have not yet investigated the effect of different Δ values in our model, we expect that the properties of the camera network and objects to be tracked will similarly affect optimal values for Δ.

However, following the analogy with pheromone evaporation in ant colonies [14], over time the strength of the links also decreases, allowing the system to overcome changes in topology or cameras' fields of view over time. The pheromone update rule is shown in equation 3.

$$\tau_{ix} = \begin{cases} (1-\rho) \cdot \tau_{ix} & \text{if no trade occurs on the edge} \\ (1-\rho) \cdot \tau_{ix} + \Delta & \text{if trade occurs on the edge} \end{cases}$$
(3)

As in ant colony optimisation, ρ is the evaporation rate parameter; higher values lead the pheromone to evaporate faster, enabling the system to adapt to changes quicker, but at a penalty of losing more historical vision graph information. However, our approach here is not ant colony optimisation, since pheromone information is not used to find optimal routes through the network, but instead to represent a social network of cameras with adjacent fields of view.

The initial broadcast behaviour of cameras can then be dialled down as the vision graph is built up. Specifically, when advertising an object that other cameras may wish to buy, a camera i sends a message to camera x with probability $P(i,x)$, otherwise it does not communicate with camera i at that time.

In this paper we consider two ways of determining these communication probabilities: firstly proportionally to the strength of the links, as given in equation 4 and secondly where the camera always advertises to those in its vision graph, and with some small probability every other camera in the network, as given in equation 5. We call these communication schedules SMOOTH and STEP respectively. This represents a novel use of ant inspired systems in the computing domain, as a method of managing communication schedules.

$$P_{\text{SMOOTH}}(i,x) = \frac{1 + \tau_{ix}}{1 + \tau_{im}}$$
(4)

where m is the camera with the highest strength value, e.g.

$$m = \underset{y}{\operatorname{argmax}} \, \tau_{iy}, \forall y$$

$$P_{\text{STEP}}(i,x) = \begin{cases} 1 & \text{if } \tau_{ix} > 0 \\ \eta & \text{otherwise} \end{cases}$$
(5)

C. Autonomous Camera Control

Putting together the aspects of the camera's utility function, decision process, trading behaviour and vision graph generation, we specify that each camera in the system behaves according to algorithm 1.

As indicated in step 4, the handover algorithm should be run regularly enough to ensure that objects are handed over as close as possible to the optimal time, but without spending unreasonable resources identifying objects in the scene purely for the purposes of determining optimal bids.

Algorithm 1 The camera handover algorithm

1) *Object trading of camera i*
    a) Advertise owned objects to each other camera x with probability $P(i, x)$.
    b) For each received advertised object j, respond with a bid at value $u_i(j)$ if this is greater than zero.
    c) Accept received bids for each object k for which $u_i(k)$ is less than the highest received bid. For each accepted bid:
        i) Remove k from O_i.
        ii) Respond to the camera making the highest bid, informing it of the required payment, the value of the second highest received bid.
        iii) Increment the camera's utility by the value of the second highest bid.
    d) For each object l for which the bid sent was accepted, add l to O_i and deduct the payment amount from the camera's utility.
2) *Vision graph update of camera i*: Update τ_{ix} for all x according to equation 3.
3) *Tracking decisions of camera i*: Select which objects in O_i to track in order to maximise $U_i(O_i)$.
4) Repeat at regular intervals.

IV. EXPERIMENTAL STUDY

To test our approach, we created a 2D simulation framework with static cameras; the cameras' fields of view are modelled as segments (however, visualised as triangles in figures 1 and 3). Each camera is controlled independently, by an autonomous software agent capable of communicating with other such agents via message passing. At this stage, we assume perfect tracking (i.e. every object within the FOV is properly detected and identified) and calculate the visibility of an object based the inverse Euclidean distance between the camera and the object and the simulated position of within the FOV.

In each simulation run, the total cumulative utility across all cameras was recorded (the social welfare) as a measure of tracking performance. The number of messages sent between cameras was also measured.

Four qualitatively different test scenarios were defined, and each of these was used to compare the performance of the six different variants of the approach presented. These are illustrated in figure 1. Scenario 1 is the simplest scenario, consisting of a row of cameras. Scenario 2 is similar though the object's path is not allways coverd by cameras. This is used to illustrate that our approach can deal with non-overlapping fields of view. Scenario 3 is more complex, simulating a heavily covered corridor. Scenario 4 is similar, but with more irregular overlaps.

Furthermore, we tested the approach with different numbers of objects. Each object moves in a straight line in a certain direction, which is initially defined such that it moves through the fields' of view of the cameras. To keep a constant number of objects in the simulation, objects cannot leave the simu-
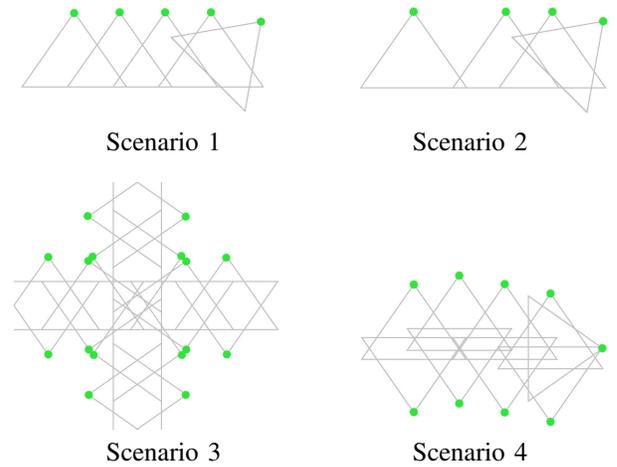

Fig. 1. Illustrations of the scenarios tested. Each camera is represented by a circle, with its field of view indicated by the associated triangle.

lation but once they reach the boundary of the environment, change their direction randomly and continue in that direction until another boundary is reached.

A. Broadcast Approaches

Initially, two simple broadcast approaches, which we refer to as *active* and *passive*, were tested in the simulation environment. In both broadcast approaches, each advertisement message is broadcast to all other cameras in the network. In the *active* approach, each camera advertises every object it owns to the entire network at each simulation time step. This means that other cameras attempt to gain ownership of objects as soon as they enter their FOV. On the one hand this results in a perfect tracking utility since the camera with the highest utility for an object has ownership of it, but on the other hand the communication between the cameras is significant. Contrary to this, the $passive$ approach minimizes the communication by sending advertisement messages only when an object is about to leave the FOV of its current owner. Though this reduces communication, it requires that the camera's utility from the object is almost zero before handing over, even though another camera might have had a better view earlier.

Figure 2 shows the overall system utility (i.e. the tracking performance of the network) and the communication overhead for the *active* and *passive* algorithms in scenario 1 with a single object moving from left to right. The spikes in utility occur when the object moves into the areas of high visibility in front of each of the cameras. Due to the particular set-up of this scenario, there is little difference in utility between the two approaches, other than between the final two cameras, where the *active* approach is able to hand over the object sooner, which increases the objects' visibility to the network, and hence system utility. However, it is clear that the *active* approach uses significantly more communication.

Since the *active* approach yields the highest possible levels of communication and utility, the subsequently presented results in this paper are normalised in each case by the results
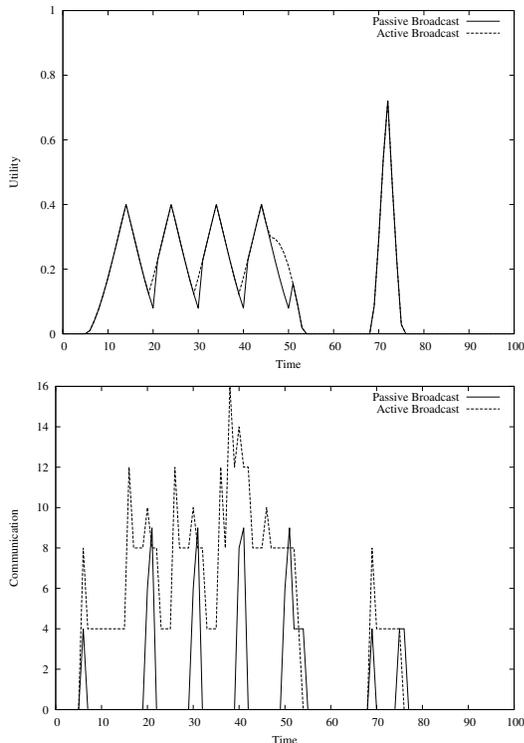
Fig. 2. System utility (above) and communication usage (below) over time, during a typical run of scenario 1 with one object. Active and passive broadcast algorithms are compared.

from the *active* broadcast approach.

B. Multicast Approaches

It is clear that the market-based approach presented does not require a vision graph in order to achieve effective object handover. However, by generating the vision graph during runtime, the camera network is able to achieve outcomes which balance the trade-off between communication and tracking performance. By scheduling the communication intelligently, as described in section III, the cameras may intelligently reduce communication, while minimising the associated performance penalty.

The following experiments illustrate the effect of the multicast approaches SMOOTH and STEP, as described in section III, when applied to both the *active* and *passive* schedules. In all cases, $\rho = 0.005$, $\Delta = 1.0$ and $c_j = 1$ for all cameras.

Figure 3 illustrates the pheromone-based approach to building the vision graph online during runtime. The state of the vision graph is shown at four points through the simulation, from initialisation where no adjacency information is known. As the objects are traded between cameras, the links (indicated by thicker red lines) are constructed. Over time, unused links reduce in strength.

Figure 4 shows the overall performance of each of the six variants of the approach firstly on scenario 1 and secondly on scenario 3, with one object in the environment. Due to the stochastic nature of the object's trajectory and the
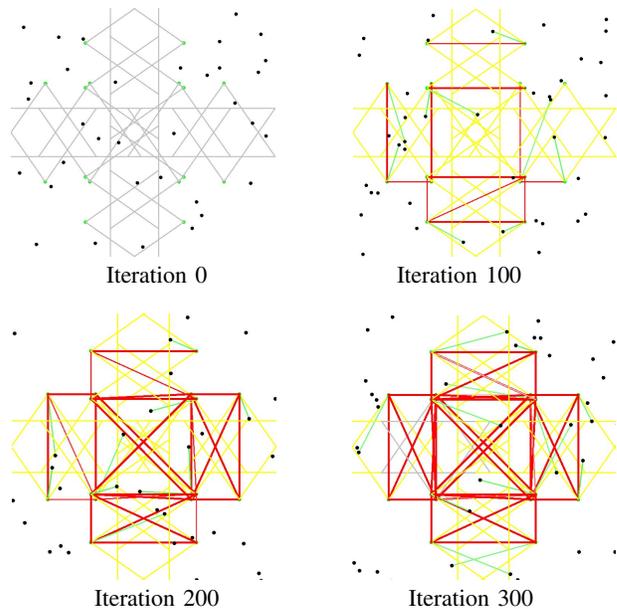

Fig. 3. The vision graph is built up during runtime through trading interactions. Red lines indicate links in the vision graph; thickness indicates strength. A cameras' FOV turning yellow indicates a detected object.

communication algorithms, mean and standard deviation are shown for each approach, calculated over 30 independent runs.

These results clearly show that the greatest difference between outcomes in the simpler scenario is obtained when switching between *active* and *passive* approaches, while the difference between broadcast and multicast communication schedules has little effect. However, in the more complex scenario, the different approaches yield different outcomes in the trade-off between communication and tracking performance. A Pareto front emerges, allowing the operator to select between different handover algorithms based on how performance and communication are valued.

For example, it could be imagined that for a camera network where high tracking performance is crucial, and cameras are connected with a high bandwidth connection, the *active* broadcast or *active* SMOOTH approaches would be most suitable. However, in a deployment where cameras have limited communication ability, some tracking performance can be traded off for communication efficiency by selecting perhaps *passive* broadcast or even *passive* STEP. Similar experiments with complex environments and larger numbers of objects yielded qualitatively similar Pareto fronts, indicating that this is a characteristic of complex tracking tasks.

V. CONCLUSIONS

In this paper, we have presented a socio-economic approach to identify spatial relations among FOVs in smart camera networks. This fully decentralised and computationally efficient approach relies on self-interested, autonomous cameras which trade tracking responsibilities for objects using Vickrey auctions. As demonstrated in our simulation study, this virtual market for objects to track achieves scalable and robust
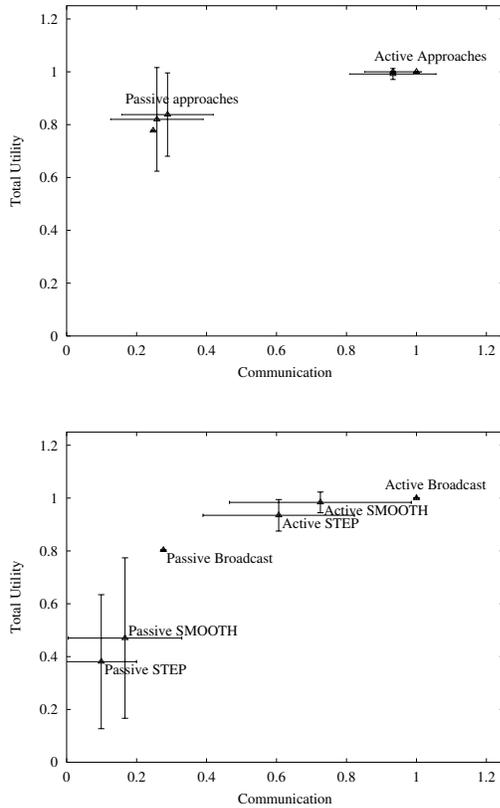
Fig. 4.   Performance (overall utility calculated across 1000 time steps) of each of the six algorithms in scenario 1 (above) and scenario 3 (below). Both utility and communication values are normalised by those from the active broadcast algorithm. The trade-off between performance and communication is apparent. Due to the stochastic nature of the object paths and algorithms, the mean and standard deviation are shown, calculated over 30 independent runs of the simulation.

tracking handover without relying on any a priori topology knowledge. By observing the trading behaviour we learn the visual neighbourhood relations in the camera network and generate the vision graph of the network.

In our experiments we have explored the trade-off between communication effort and tracking performance. Furthermore, we have presented a novel ant-inspired method for efficiently targeting marketing communication effort, such that the associated utility penalty in the trade-off is minimised. Our market-based approach results in a Pareto front for all tracking scenarios. Hence, a network operator can choose among different performance/communication settings.

In simple scenarios, *passive* approaches achieved a communication reduction of around 75% for a 20% penalty in tracking performance. Conversely, in more complex scenarios *passive* SMOOTH and *passive* STEP allowed reductions in communication by as much as 90%, but with tracking performance more than halved. However, *active* SMOOTH and *active* STEP were found to provide reasonable positions within the trade-off, allowing a 20-40% reduction in communication but with less than a 10% drop in performance.

We believe that socio-economic methods can fundamentally help to increase autonomy, robustness and flexibility in smart camera networks. However, there is still a lot of room for future work. One direction is to relax some of our assumptions on tracking performance, networking capabilities and spatial structure of the environment. Another direction includes the modelling of the utility function over (future) time periods and the elaboration of more advanced trading mechanisms. Finally, we are currently implementing our novel approach in a real smart camera network.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.
[2] Y. Li and B. Bhanu, "A Comparison of Techniques for Camera Selection and Handoff in a Video Network," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, Como, Italy, 2009, pp. 1–8.
[3] B. Song, A. T. Kamal, C. Soto, A. K. Roy-Chowdhury, and J. A. Farrell, "Tracking and Activity Recognition Through Consensus in Distributed Camera Networks," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2564–2579, 2010.
[4] U. M. Erdem and S. Sclaroff, "Look there! Predicting where to look for motion in an active camera network," in *Proceedings of the IEEE Conference on Vision and Signal-based Surveillance*, Como, Italy, 2005, pp. 105–110.
[5] B. Rinner and W. Wolf, "Introduction to Distributed Smart Cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1565–1575, October 2008.
[6] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous Multicamera Tracking on Embedded Smart Cameras," *EURASIP Journal on Embedded Systems Volume 2007*, vol. 2007, p. 10, 2007.
[7] Y. Li and B. Bhanu, "Utility-based Camera Assignment in a Video Network: A Game Theoretic Framework," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 676–687, 2011.
[8] F. Qureshi and D. Terzopoulos, "Multi-camera Control through Constraint Satisfaction for Persistent Surveillance," in *IEEE Conference on Vision and Signal-based Surveillance*, Santa Fe, USA, 2008, pp. 211–218.
[9] H. Detmold, A. van den Hengel, A. Dick, A. Cichowski, R. Hill, E. Kocadag, K. Falkner, and D. S. Munro, "Topology Estimation for Thousand-Camera Surveillance Networks," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, Austria, 2007, pp. 195–202.
[10] T. J. Ellis, D. Makris, and J. K. Black, "Learning a multi-camera topology," in *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003, pp. 165–171.
[11] Z. Mandel, I. Shimshoni, and D. Keren, "Multi-Camera Topology Recovery from Coherent Motion," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, Austria, 2007, pp. 243–250.
[12] H. Kim, J. Romberg, and W. Wolf, "Multi-Camera Tracking on a Graph Using Markov Chain Monte Carlo," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, Como, Italy, 2009.
[13] S. H. Clearwater, Ed., *Market-Based Control: A Paradigm for Distributed Resource Allocation*.   Singapore: World Scientific, 1996.
[14] M. Dorigo and T. Stützle, *Ant Colony Optimization*.   The MIT Press, 2004.
[15] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.
[16] P. R. Lewis, P. Marrow, and X. Yao, "Resource Allocation in Decentralised Computational Systems: An Evolutionary Market Based Approach," *Journal of Autonomic Agents and Multi-Agent Systems*, vol. 21, no. 2, pp. 143–171, 2010.