Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks

LUKAS ESTERLE, Alpen-Adria Universität Klagenfurt and Lakeside Labs, AT
PETER R. LEWIS, University of Birmingham, UK
XIN YAO, University of Birmingham, UK
BERNHARD RINNER, Alpen-Adria Universität Klagenfurt and Lakeside Labs, AT

In this paper we present an approach to object tracking handover in a network of smart cameras, based on self-interested autonomous agents, 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 learn the vision graph, i.e., the camera neighbourhood relations, 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, enabling efficient deployment in unknown scenarios and camera network topologies, 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. We have evaluated our approach both in a simulation study and in network of real distributed smart cameras.

1. INTRODUCTION

There is currently a trend in camera networks towards increasing resource-awareness, adaptability and autonomy. Such visual sensor networks [Soro and Heinzelman 2009; Bhanu et al. 2011] avoid resource-intensive infrastructures for data and energy distribution, are able to modify their functionality during runtime and perform in-network processing of the captured data. 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., [Yilmaz et al. 2006]), 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., [Li and Bhanu 2009; Song et al. 2010]). 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. We use the term socio-economic to describe our approach, since it is inspired by both social and economic principles. The approach employs self-interested, autonomous cameras exchanging responsibility for tracking objects in a market mechanism in order to maximise their own utility. When a handover is required, an auction is initiated and cameras that have received the auction initiation try to detect the object within their field of view (FOV). This means, in case an object does move into the FOV of another camera and no auction has been initiated, the object will pass unnoticed. Having the object detected within its FOV, the camera can try to get the responsibility to track the object by bidding for it. This allows each camera in the system to track only those objects returning a high utility and keep resource expensive tasks such as (re-)detection of objects at a minimum. 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 significant 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, the system 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.

This paper extends our preliminary work [Esterle et al. 2011] by reporting on an implementation of the technique in a network of real smart cameras, by an extensive experimental evaluation and by a more detailed discussion of related work. 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 to efficiently manage this trade-off. Our novel approach has been evaluated by a simulation study and experiments on a real smart camera network.

The remainder of this paper is structured as follows. Section 2 provides a background to object tracking with distributed smart cameras and market-based control. Section 3 formally introduces the problem of handing over the tracking responsibility and the identification of neighbourhood relations in multi-camera networks, i.e., the generation of the vision graph. Section 4 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. Sections 5 and 6 describe our experimental study based on simulations and measurements in a network of distributed smart cameras; they also summarise key results. Finally, Section 7 discusses the implications of this work and identifies areas for further study.

2. RELATED WORK

2.1. 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 [Erdem and Sclaroff 2005]. 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 [Li and Bhanu 2009].

In smart camera networks much effort is put on distributed and resource-aware handover mechanisms due to increased scalability and robustness [Rinner and Wolf 2008]. One of the first autonomous handover approaches on smart cameras was presented by Quaritsch et al. [Quaritsch et al. 2007]. 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 FOVs of the cameras. Whenever a target object enters a migration region, a tracker is then started on the neighbouring camera(s). Möller et al. [Möller et al. 2008] present a similar approach which is also based on a static and *a priori* known vision graph to identify the next camera for the handover.

Li and Bhanu [Li and Bhanu 2011] present a game-theoretic approach to 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. Additionally, the algorithm has been implemented in a centralised way, which does not provide good scalability or robustness. Morioka et al. [Morioka et al. 2010] describe a fuzzy-based camera selection method for multi-camera tracking. Although the authors aim for a distributed and efficient solution, their approach requires the tracking states and camera states of all cameras for every selection decision. Qureshi and Terzopoulos [Qureshi and Terzopoulos 2008] 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. Chen et al. [Chen et al. 2010] introduce a handover mechanism which considers the trackability of the object as well as the available resources of the camera for selecting the new camera. The trackability of the object of interest is computed by the resolution, the distance to the border of the FOV and occlusion. However, this handover mechanism requires calibrated cameras with overlapping FOVs.

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 [Detmold et al. 2007]. There are two main cases to distinguish for topology estimation: overlapping and non-overlapping FOVs [Radke 2010]. In the overlapping case, cameras observe parts of the same environment from different perspectives, i.e., the vision graph contains an edge between two cameras if they share some FOV. For example, [Cheng et al. 2007] use SIFT features to identify common areas and create the vision graph in a distributed camera network. In the non-overlapping case, cameras do not observe the same part of the environment. Neighbourhood relationships between the cameras—and hence edges in the vision graph—are induced by the likelihood that an object in one camera appears in another after some time.

Observing moving objects within the network is often used to learn the topology over time (e.g., [Ellis et al. 2003; Mandel et al. 2007; Kim et al. 2009]). The estimation approaches vary in the topology assumptions (e.g., overlapping or non-overlapping FOVs), topology modelling and the extraction of relevant information from individual camera views. Marinakis and Dudek [Marinakis and Dudek 2006] use a Monte Carlo Expectation-Maximization algorithm to learn the neighbourhood relations in a network of sensors. The approach uses only detection events from the deployed sensors and results in a probabilistic model of the neighbourhood relations in the sensor network. Makris et al. [Makris et al. 2004] exploit temporal correlations in observations of object movements, i.e., the entry and the exit points of the object in each cam-

era, through the camera network. They also use Expectation-Maximization to learn a Gaussian mixture model that links the entry/exit zones of neighbouring cameras.

In our approach, the trading of an object provides an implicit snapshot on the network topology and is used as an event for generating the vision graph, i.e., the "selling" and the "buying" by cameras represent a neighbourhood relationship.

2.2. Market-based Control

Fundamentally, the task of deciding which camera should track which object at a given time can be viewed as a resource allocation problem: how to allocate the known objects amongst the cameras, given their local knowledge, available resources and objectives [Rinner et al. 2012]. Due to common ground between the characteristics of networks of autonomous nodes and those of economic systems, one approach is to view the camera network as an economy. Indeed, economics has long been successfully used as inspiration for techniques to perform resource allocation in engineered systems. This has resulted in a family of techniques, known as market-based control, concerned with the application of economic principles in tackling resource allocation problems in distributed systems [Clearwater 1996].

The typical approach taken is for the actions and decisions of autonomous nodes in a market-based system to be automated by the use of software agents, which interact through some defined market mechanism. Buyer agents attempt to purchase resources from the available sellers, to maximise their utility function, which is derived from the task or requirements of the user. Sellers charge an amount of either real or artificial money for the resource, at a price determined by their strategy and dependent on factors such as the quantity or quality of the resource being provided. Since self-interested buyers can be expected to pay more for resources which they desire more, and self-interested sellers will charge what they can get away with in order to maximise their payoff, resources will tend to go to those who value them the most. The fundamental idea is that competition between agents for the same resources leads to an efficient allocation, reflective of the agents' objectives and preferences, as represented in their local utility functions.

Since the concept of market-based control was first proposed by Clearwater et al. [Clearwater 1996], it has been applied to a wide range of application domains, with significant benefits. In load management in giant-scale web services, Brewer [Brewer 2001] argues that by incorporating into a request for a resource, a notion of its value or cost, smart agents allow for responsive adaptation and graceful degradation in the presence of changes to the network. Similarly, Gupta et al. [Gupta et al. 1999] argued that mechanisms involving pricing and user self-selection can be preferable, since they can remove the need for centrally managed and enforced resource usage limits. The mechanisms used in market-based control systems typically fall into one of four groups: centralised auctions, distributed auctions, bilateral bargaining and posted offer markets. Reviews of the varying assumptions and benefits of these techniques are provided by Cliff [Cliff 1997] and Lewis et al. [Lewis et al. 2010].

Common examples of centralised auctions include English, Dutch and Vickrey auctions, in which a central auctioneer facilitates the bidding and determines the allocation of resources for all participating agents. Where scarcity exists on both the seller and buyer sides, double auctions such as the Continuous Double Auction and Clearing House provide an alternative approach [Friedman and Rust 1993]. Research in the field of automated mechanism design also suggests that other less obvious auction mechanisms may lead to more efficient outcomes in certain circumstances [Phelps et al. 2009]. Wolski et al.'s [Wolski et al. 2001] G-Commerce model is an example of an application of a centralised market to resource allocation in computational grids.

However, Cliff and Bruten [Cliff and Bruten 1998] argue that the presence of such a centralised process or component removes the primary advantage of using a market-based system: its robust, decentralised, self-organising properties. Distributed auction mechanisms (e.g. [Hausheer and Stiller 2005]) attempt to mitigate this, reducing the fragility associated with reliance upon a single point, providing more scalability and allowing for dynamic composition of auctions. Typically, either the central auctioneer is replaced by a number of local ones, which may communicate through some secure means, or else the auctioneer role is fulfilled by a spare, disinterested node. Double auctions for example, though relying on a specialist to match bids and asks [Phelps et al. 2004], may be decentralised by the presence of multiple specialists between which the participants may choose [Niu et al. 2008; Robinson et al. 2010].

One simple approach to distributed auctions, and that taken in this paper, is that individual nodes themselves host independent auctions for their own resources. One early example of this approach in computational resource allocation is *Spawn* [Waldspurger et al. 1992], in which agents attempting to acquire resources bid in sealed-bid auctions hosted by agents of the providers, for their resources. Depending on the utility functions of the buyers and the auction mechanism used by the resource owner, this can require a high level of strategic ability on the part of buyers, as they must decide in which auctions to participate. Since they may win multiple auctions, questions arise of how to handle these situations. Literature exists which explores the dilemma faced by buying agents bidding in multiple auctions, such as that by Gerding et al. [Gerding et al. 2006; 2007]. However, this is only an issue if the buyer requires a limited supply of equivalent commodities, which are available from a range of sellers. A buyer which does not consider items as substitutes for each other need not be concerned with this eventuality.

The third family of market-based control approaches depends not on an auctioneer or other centralised entity, but instead makes use of the ability of intelligent agents to bargain between themselves in order to arrive at acceptable prices. This approach is realised in the AVALANCHE [Eymann 2001], and CATNET [Ardaiz et al. 2006] systems, and takes inspiration from Agent-based Computational Economics (ACE) [Tesfatsion and Judd 2006], an agent-based modelling technique which attempts to replicate the dynamics of human markets with complex cognitive agents. Bargaining agents are the subject of continuing research (e.g., [Gerding and La Poutré 2006; Lopes et al. 2008; Chandra et al. 2010]).

The final family of approaches takes inspiration from modern retail markets, using a mechanism usually referred to as the *posted price* or *posted offer* model [Plott and Smith 1978; Ketcham et al. 1984], though in online content delivery it is sometimes referred to as the *quoted price* model [Jagannathan and Almeroth 2002]. This is a fully decentralised approach to the determination of price without the need for complex bilateral negotiation, and provides a potentially simpler alternative. The application of posted offer markets to computational resource allocation is the topic of ongoing research [Lewis et al. 2010], and the technique has been shown to demonstrate high levels of scalability and robustness. However one assumption of the current state of the art is the ability of nodes to broadcast information globally about the system with negligible cost.

In this paper, we make use of distributed owner-hosted Vickrey auctions [Vickrey 1961], as will be described in Section 4. Since we use Vickrey auctions, which have truth-telling as their dominant strategy for single item auctions, we need not be concerned with complex bidding strategies for buying agents. Additionally, since each object to be tracked is unique and therefore our resources are not substitutable, our buyers need not be concerned with the consequences of winning multiple auctions for

the same item. However, with no centralised auctioneers, we retain the scalability and robustness properties, which are inherent to the approach.

3. 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. The decision of camera i 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 FOV of the camera. 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.

The handover of the tracking responsibility is a local mechanism in a camera network, i.e., only the (small) set of neighbouring cameras can contribute to the handover decision. The vision graph ($G_v = (V, E)$) expresses such neighbourhood relations in the camera network. Two cameras i and k (both members of V) are connected in G_v by an directed edge $e_{i,k} \in E$, if they have an overlapping FOV. We extend this basic definition of G_v to non-overlapping cameras as well by introducing an edge if a moving object in camera i can appear in camera k within some time. Weights of the edges can be used to express the likelihood and rates of the object re-appearance (cp. Section 4.2). Thus, to reduce communication for deriving the handover decision at camera i, it is sufficient to exchange information only among i and its adjacent cameras $N_G(i)$. As a side effect, besides reducing the communication, our approach reduces the overall resource consumption within the network when only a small set of neighbouring cameras contribute to the handover decision process. This is because only those cameras have to (re-)detect the object upon request while other cameras are unaffected.

4. 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. First, the allocation of objects to cameras makes use of a market-based approach, similarly to those described in Section 2.2. Second, 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 [Dorigo and Stützle 2004], where artificial pheromones are used to find good (i.e., short) paths in a network. 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.

4.1. 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

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

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

where $\phi_i : O_i \to \{0, 1\}$ 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 [Vickrey 1961] 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. [Lewis et al. 2010]), 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 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. In order to determine the likely value, each camera receiving an auction invitation checks whether an object within its current FOV matches with the received object description. 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 utility of the camera (see equation 2).

4.2. 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 relative utility of the cameras 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 is 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 properties of the problem. 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, over time the strength of the links also decreases, allowing the system to overcome changes in topology or fields of view of the cameras over time. The pheromone update rule is shown in equation 3.

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

As in ant colony optimisation, ρ is the evaporation rate parameter, which can be understood as a forgetting factor; 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: first proportionally to the strength of the links, as given in equation 4 and second 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.

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

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

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

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

where $\epsilon = 0.1$. Only in the case of having no links in its vision graph ($\tau_{im} = 0$), a camera will use broadcast instead of STEP communication to build up its own vision graph.

4.3. Autonomous Camera Control

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

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 utility of the camera 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 utility of the camera.
(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.

As indicated in step 4, the handover algorithm should be repeated at regular intervals 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.

As commonly used in distributed tracking applications, the object or person of interest has to be defined by an operator. In our application, the operator therefore has to connect to a remote camera and select the object or person to be tracked in a user interface. This user interface is only required to initiate the tracking process and does not act as a central component or is not needed in any way besides initialisation, to support our approach.

5. EXPERIMENTAL STUDY IN SIMULATION

To test our approach, we created a software framework for simulating multiple object tracking within a network of static cameras. Object movement and camera coverage was simply modelled in 2D space. The simulation software was implemented in Java and was able to simulate the movement of objects, the execution of the camera control and the data exchange between cameras based on a fixed simulation time interval.

5.1. Simulation Environment

In the simulation, the fields of view of the cameras 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 single camera tracking (i.e., every object within the FOV is properly detected and identified) and calculate the visibility of an object based on the inverse 2D Euclidean distance between the camera and the simulated position of the object. To keep a constant number of objects in the simulation, objects cannot leave the simulation but once they reach the boundary of the environment, change their direction randomly and continue in that direction until another boundary is reached. Since objects do not leave our simulation environment, objects will never get lost for the entire network forever. In a real environment this might not be realistic. If a camera has not seen one of its owned objects and did not receive a bid for the initiated auction for an extended amount of time, it may choose to stop attempting to detect or track it, in order to save resources. Exactly how this is handled is largely an implementation detail handled at the camera level, and will depend on the priorities associated with the scenario.

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.

5.2. Test Scenarios

For the simulation framework, six 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 path of the object is not always covered by cameras. This is used to illustrate that our approach can deal with non-overlapping FOVs. Scenario 3 is more complex, simulating a heavily covered corridor. Scenario 4 is similar, but with more irregular overlaps. Scenario 5 again is a rather simple scenario but this time using five cameras with staggered positions. And finally, scenario 6 illustrates an example network with 36 randomly placed cameras.

Furthermore, we tested the approach with different numbers of objects. As already mentioned, the number of objects is kept constant for the duration of each simulation run. Initially each object moves in a straight line in a certain direction, which is defined such that it moves through the FOVs of the cameras, until it reaches a boundary where it changes its direction randomly.

5.3. Broadcast Approaches

Initially, two simple broadcast approaches, which we refer to as *active* and *passive*, were tested in the simulation environment. In both approaches, each advertisement message is broadcasted 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 always has ownership of it, but on the other hand the communication between the cameras is significantly higher. Contrary to this, the *passive* approach minimises the communication by sending advertisement messages only when an object is about to leave the FOV of its current owner. Furthermore cameras are only required to attempt to gain ownership of objects when receiving an auction initiation and hence have lower resource consumption. Though this reduces
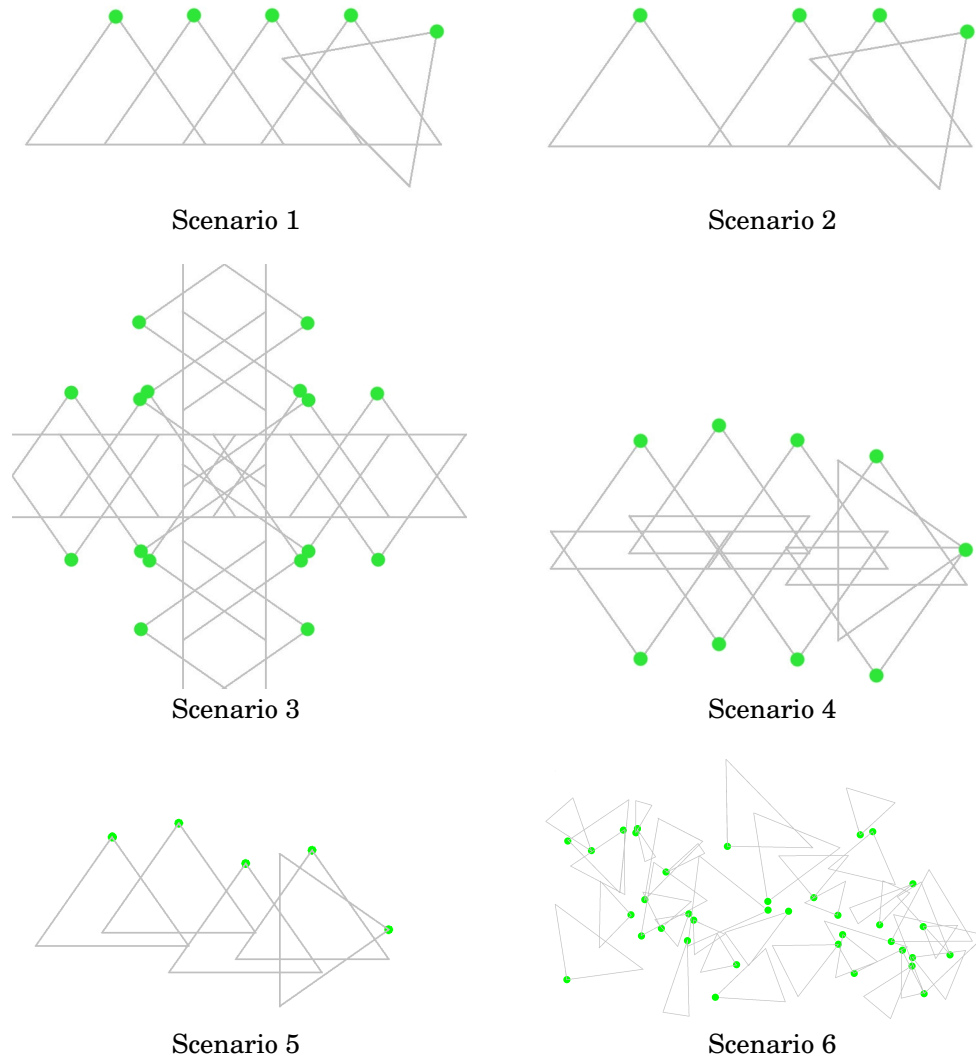
Fig. 1. Illustrations of the scenarios tested. Each camera is represented by a circle, with its field of view indicated by the associated triangle.

communication, it requires that the utility of the camera from the object is almost zero before handing over, even though another camera might have had a better view earlier. This means, our *active* approach refers to auctions being initiated at regular intervals while our *passive* approach initiates auctions only on-demand, dependent on the physical environment.

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 visibility of the objects to the
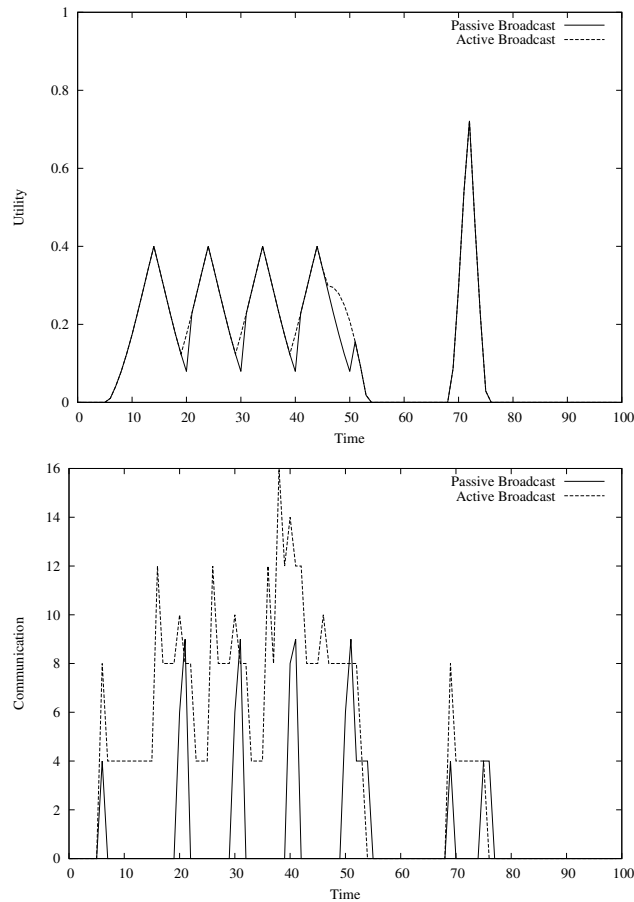
Fig. 2.   System utility (above) and communication usage (below) over time, during a typical run of scenario 1 with one object. Active and passive broadcast algorithms are compared.

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 from the *active* broadcast approach.

5.4. 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 more efficiently the trade-off between communication and tracking performance. By scheduling the communication intelligently, as described in Section 4, 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 4, 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.
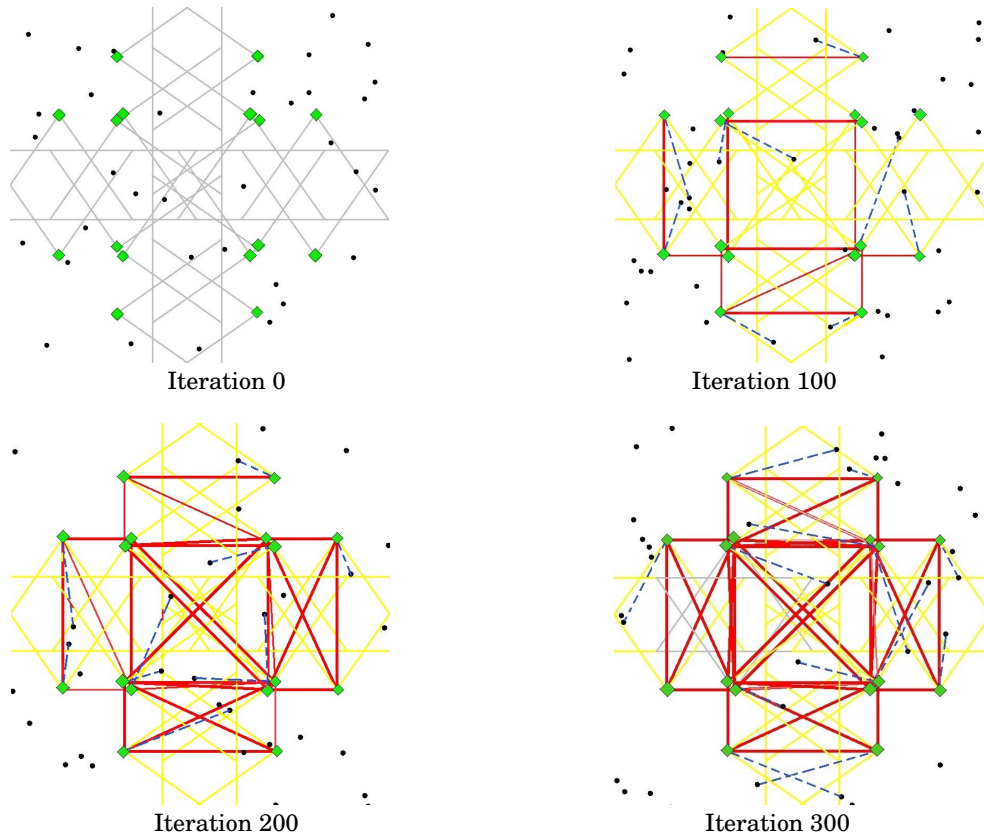
Fig. 3. The vision graph is built up during runtime through trading interactions. Red lines indicate links in the vision graph; thickness indicates strength. Cameras are indicated as small squares. Tracked objects are assigned to the respective camera via a dashed line.

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 on scenario 1 (upper left), scenario 3 (upper right), scenario 5 (lower left) with one object in the environment, and also our random scenario - scenario 6 (lower right) - with 31 objects in the environment. Due to the stochastic nature of the trajectory of the object and the 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 scenarios is obtained when switching between *active* and *passive* approaches. However, in the more complex scenarios (e.g., scenario 3 or 4), 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. As one can see, the results for scenarios with a lower number of cameras make it important
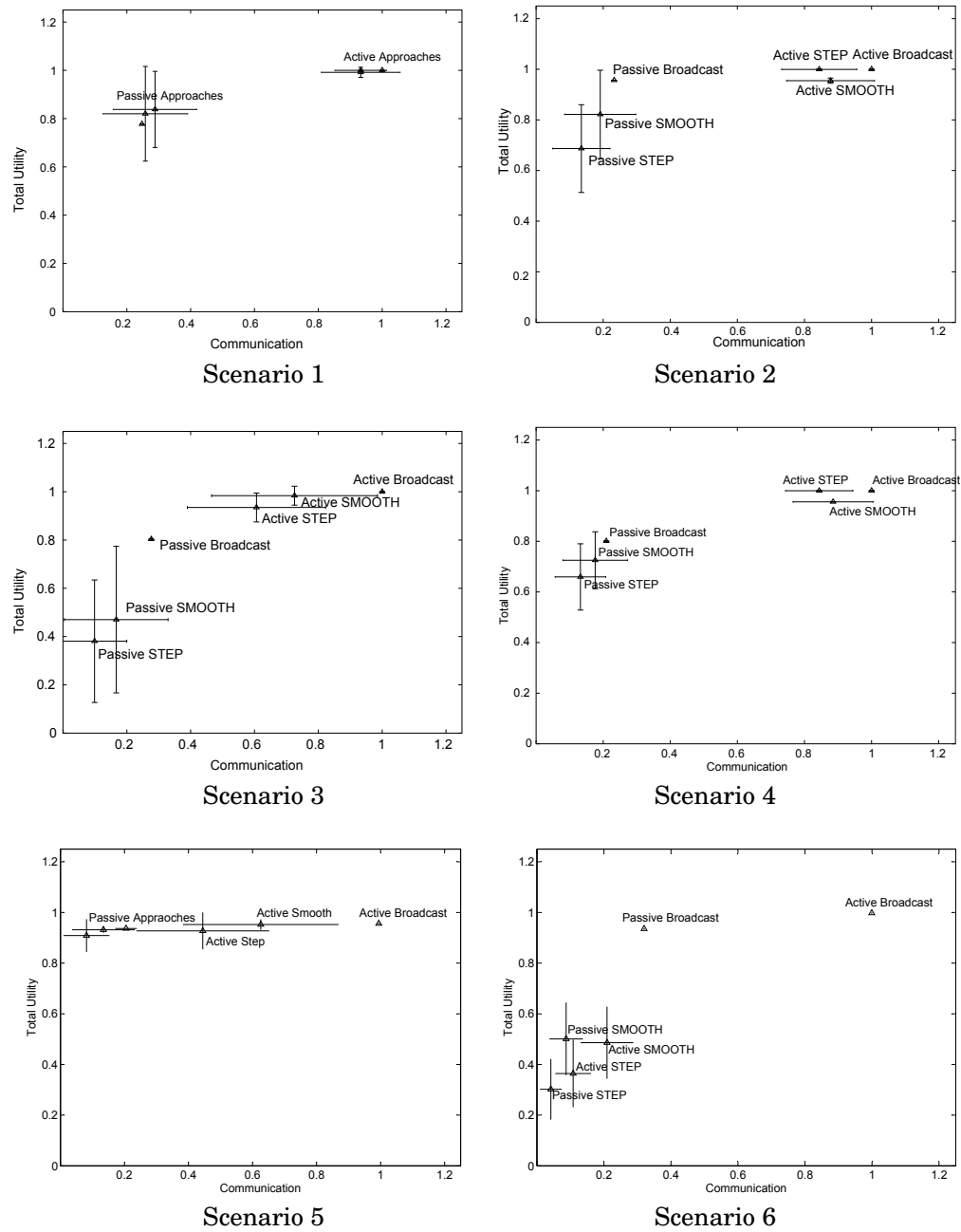
Fig. 4.   Performance (overall utility calculated across 1000 time steps) for all six scenarios using our differ-
ent algorithms. Both utility and communication values are normalised by those from the active broadcast
algorithm. The trade-off between performance and communication is apparent. Due to the stochastic na-
ture of the object paths and algorithms, the mean and standard deviation are shown, calculated over 30
independent runs of the simulation.

to select the right algorithm variant, while scenarios with a higher number of cameras have to consider the correct communication strategy as well.

For example, it could be imagined that for a camera network, with a complexity similar to scenario 3 or 4, 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. In completely random scenarios, though, switching between *active* and *passive* approaches has less impact on the performance than switching between communication schedules. This is mainly due to the fact that a high number of objects may create very strong links between certain cameras at an early stage of the simulation. Hence other cameras are not considered anymore as communication partners at a later stage and therefore utility is lost. Again an operator could switch from broadcast communication to a less communication intensive approach as soon as the vision graph is developed.

In [Esterle et al. 2012], we relaxed some of our previous assumptions such as instantaneous handover, instantaneous detection, instantaneous auctions and instantaneous communication in our simulation environment which were not reflective of the real world. Furthermore, we introduced events such as failures of cameras to our simulator to show the robustness of our algorithm. Doing so, we were able to show how to improve our previous approach substantially in that paper, especially in terms of robustness and adaptivity.

The cameras still learn neighbourhood relationships online, and cluster into groups to reduce communication within the network. The paper presents our fully decentralised approach increasing the adaptivity and robustness of the camera network when compared to a static approach based on *a priori* known vision graph, while not requiring a centralised component. The approach improves the robustness of the network by enabling it to relearn the vision graph in case of camera failure. Furthermore, it improves adaptivity by incorporating newly added cameras into the network during runtime.

In [Esterle et al. 2012], we showed that different variants of our approach are able to retain or even increase their utility after events occur which changes the topology of the network. For example, when comparing cumulative network utility with overall communication overhead, the technique maintains a steady performance despite the presence of uncertainties. Depending on the scenario and the injected event (camera failure or camera adding) the overall utility exploiting dynamic vision graph information was 10%-20% better than exploiting only the static vision graph information.

6. EXPERIMENTAL STUDY WITH REAL CAMERAS

To evaluate our approach under realistic conditions, we implemented our novel multi-camera tracking approach in a network of five distributed and autonomous smart cameras. In this real network study, a feature-based tracker is responsible for tracking the object of interest within the FOV of the camera. Each camera runs the autonomous camera control algorithm (algorithm 1) to perform the tracking handover and to update the locally stored vision graph information.

6.1. Experimental Setup

Our camera network is composed of custom-built smart cameras which are equipped with an Intel Atom processor running at 1.6 GHz and an 100 MBit Ethernet interface

Fig. 5. The smart camera platform. A custom-built camera running Linux on a 1.6GHz Intel Atom processor with 2 GB memory and 100 MBit Ethernet interface. The image sensor is a CCD colour image sensor with a native resolution of 1280×1024 pixels.

(cp. Figure 5). The smart camera includes a CCD colour image sensor with a native resolution of 1280×1024 pixels. The processor runs a standard Linux distribution that provides flexible integration of additional hardware modules and external software libraries.

We placed the cameras in a laboratory room setting similar to the first two scenarios of the simulation. In our experiments we performed tracking of a single person within this network of five cameras. To achieve reproducible results, we recorded videos of two different scenarios on each camera. For the evaluation, these synchronised videos served then as input for the cameras. Test scenario 1 was recorded with overlapping cameras and lasts about 70 seconds. Hence, the person was visible by at least one camera during the entire test sequence. Test scenario 2 represents a non-overlapping camera setup. Here the person moved differently; as a result, the person was not visible for some period by any camera at all. Scenario 2 lasts about 130 seconds. Figure 6 shows selected captured images of all five cameras.

6.2. Implementation Details

Clearly, there are a number of differences between the simulation environment and the real camera system. This required some refinements to the approach used, when implementing the techniques designed on the simulation platform on the real network. These implementation details are concerned with (i) the tracking algorithm and the computation of the confidence, (ii) the required time for deriving the handover decision and (iii) the triggering of handover for the passive algorithm. We describe these implementation aspects in the following paragraphs.

We use a simple frame-to-frame feature-based matching as single camera tracking algorithm which exploits SIFT features [Lowe 2004] to model the object of interest. To perform tracking in a single camera, the object must be first detected in an entire frame using the SIFT feature model and, if successful, be re-identified within some search window on a frame-by-frame basis. Thus, to advertise an object to a neighbouring camera (cp. step 1(a) in algorithm 1) we need to transfer the object model. Again, we do not use the SIFT features to build our vision graph. We rather use SIFT fea-

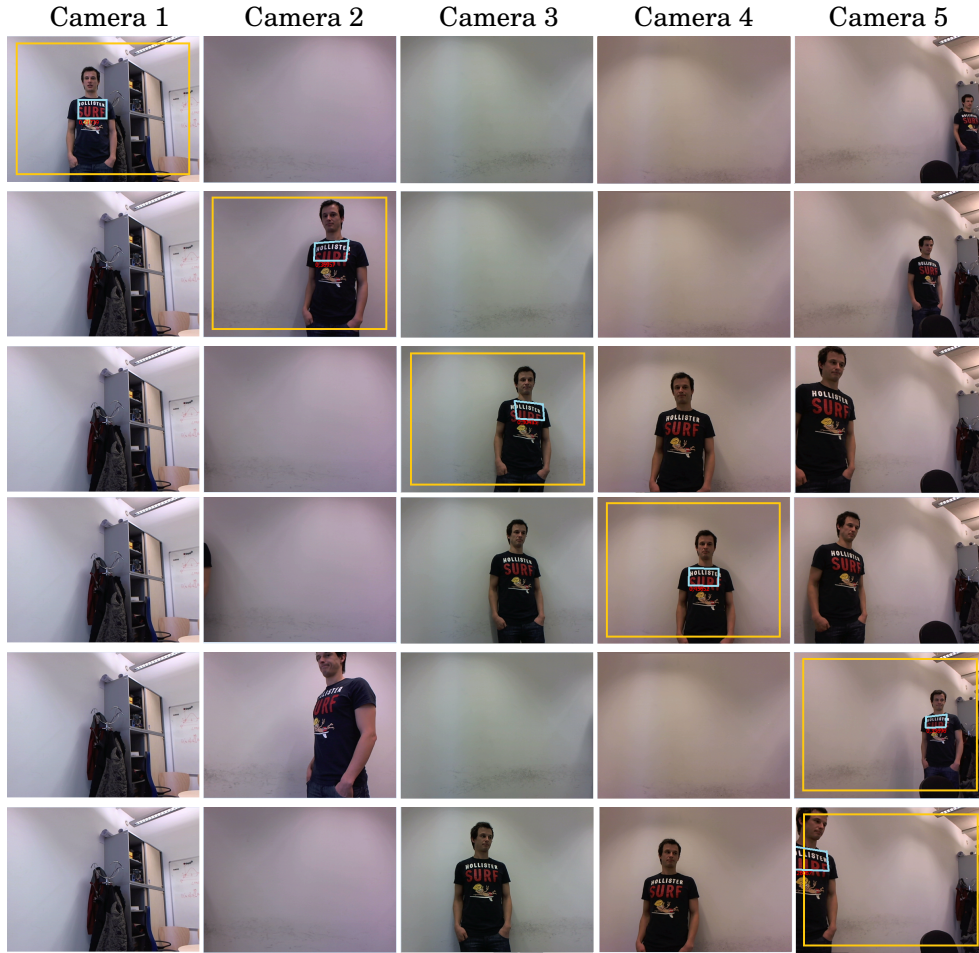| Camera 1 | Camera 2 | Camera 3 | Camera 4 | Camera 5 |

Fig. 6. Captures images of test scenario 2. Each column corresponds to one camera, and each row corresponds to a specific time point. The time difference between individual rows is about 15 seconds. The blue box represents the currently tracked object, and the orange box represents the boundary margin of the FOV of the camera.

tures to track and re-identify objects and persons in other cameras and therefore do not require overlapping FOVs.

In our implementation we define the matching rate of the SIFT features as confidence for the object detection and tracking. The model of object j is given by the set of SIFT features F_{m_j} of a specified region of interest[1]. To identify an object within the FOV, the SIFT features of the current frame img are compared to F_{m_j} using the SIFT feature matching algorithm. Thus, we define the confidence c_j of detection or tracking object j by

$$c_j = \frac{|F_{m_j}|}{|Match(F_{img}, F_{m_j})|} \qquad (6)$$

---

[1]Currently, this region of interest is defined by the user.

where $Match(F_{img}, F_{m_j})$ represents the set of matched features. Note that in the current implementation we set the visibility parameter of an object j as $v_j = 1$. Thus, we do not currently consider distance and orientation information for the utility.

Due to the communication delay of the network, the object advertisement and the subsequent object detection procedure at neighbouring cameras, the handover process is no longer instantaneous. For example, the SIFT-based model of an object has a typically size of 100 kB. Thus, the transfer of such model during the advertisement and the re-detection requires approximately 300 ms. To allow every camera enough time to participate in an auction, every auction has a certain duration. This so called auction window, starts with the arrival of the first bid. As soon as the time of the auction window has elapsed, the winner of the auction is being determined and newly arriving bids are discarded. For the update of the vision graph we defined the "sampling time" as 100 ms. Thus, the strengths τ_i of all edges are decreased by $(1 - \rho)$ every 100 ms. As defined in equation 3 a strength τ_{ix} is increased by Δ only if a trade between the two cameras has occurred. In our implementation, we set $\Delta = 1$ and $\rho = 0.005$.

For the passive approach, we defined a margin in the FOV to trigger the advertisement of an object (cp. Figure 6). Thus, when an object is detected three consecutive times within the margin we consider the object to be about to leave the FOV of the camera and start with handover.

6.3. Results

When implementing the technique on a real camera system, the significant effect of two key differences became apparent. Firstly, the simulator operates using discrete time, with one auction corresponding to one time-step, while the real system operates in real-time. This becomes even more important since the tracker is not able to process every single frame because it is computationally too intensive. The deviation for the acquired utility emerges with every handover between the cameras. Table 6.3 compares the number of handovers for the different scenarios using our active and passive approaches with different auction windows. Therefore, an appropriate auction window must be chosen for auctions to complete. Again, we ran our scenarios 30 times and, similar to our simulation, we calculated the average of the accumulated utility over all cameras in every second. We studied the cases when we used no auction window as well as when the auction window was 500 milliseconds and one second for both our active and our passive approach; figure 7 shows the results of this for scenario 1 and scenario 2.

Table I. Comparison of handovers using different auction windows within the active and passive approach in scenario 1 and scenario 2.

	0 ms	500 ms	1000 ms
Scenario 1 - Passive	4	4	4
Scenario 1 - Active	9	6	6
Scenario 2 - Passive	8	9	7
Scenario 2 - Active	18	14	14

As described in step 4, our algorithm has to be repeated at a regular interval. We found that the choice of interval to repeat our algorithm as well as the choice of auction window may affect the utility obtained by the system, and hence the underlying tracking performance of the network. Altering the auction window has the same effect as altering the interval at which the algorithm is repeated, since the cameras do not re-advertise objects while waiting for existing bids to arrive.

This is also due to the fact that handing over tracking responsibility from one camera to another does not work seamlessly. As soon as the auction ends and the winning camera has been determined, the tracker at the *selling* camera is stopped and the tracking responsibility is handed over to the winning camera. During this time no utility is gathered and hence utility is *lost*. In case the tracker gains only little utility by switching to another camera, this lost utility is not compensated. The illustrations in Figure 7 show this effect in certain situations (e.g. Scenario 2) where the active approach does not perform as well as the passive approach. Figure 8 clearly shows the deviation in overall performance for small auction windows where our active approach without an auction window does not lie on the Pareto front.

The second key difference of smart cameras is the processing power which is available to the operating trackers. Such resources can quickly become consumed when too many trackers are operating concurrently, for example due to a large number of objects being tracked, or the need to respond to a large number of auction calls. When this occurs, every single tracker will process fewer frames per second. This enhances the effect of creating less utility for a certain camera and the respective trackers within a discrete time window.

As in our simulation environment, we generated the vision graph during runtime. We employed the vision graph for our communication schedules SMOOTH and STEP. Figure 9 illustrates the generation of the vision graph during runtime for scenario 2. One can see how the vision graph is built up over time and how it evaporates, due to a lack of handovers between the cameras on the left.

Figure 10 shows the overall performance of our active and passive approaches using broadcast, SMOOTH and STEP communication schedules for scenario 2 with one object in the environment. Again there is a clear trade-off between the achieved utility and communication. It is also apparent that the results of the deployed system are very similar to the equivalent scenario studied in the simulation environment (see Figure 4, scenario 1).

7. 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 as well as in a real camera network, this virtual market for objects to track achieves scalable and robust 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. In our real network implementation, we demonstrated the autonomous handover and vision graph generation in overlapping and non-overlapping scenarios. We demonstrated the generation of a vision graph using a simple frame-to-frame feature based matching algorithm exploiting SIFT features. To compensate delays introduced by communication (e.g., transfer of the object model) and processing (e.g., object detection and tracking) we evaluated different auction window sizes.

In our experiments we have explored the trade-off between communication effort and tracking performance. 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 the tracking scenarios. Hence, a network operator can choose among different performance/communication settings.

In simple simulation scenarios, *passive* approaches achieved a communication reduction of around 75% for a 20% penalty in tracking performance. Using our implemen-
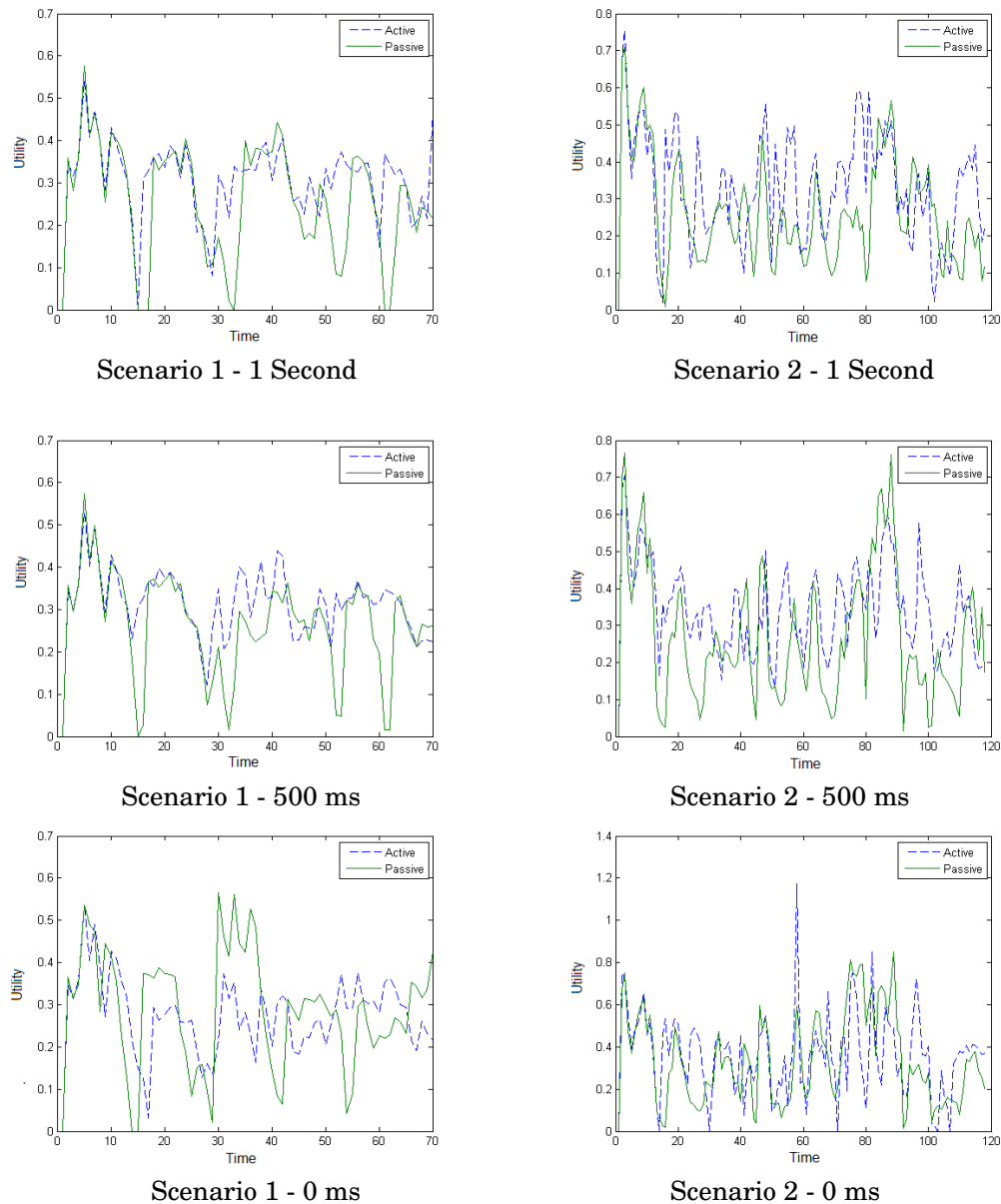
Fig. 7. Illustrations of the accumulated utility over time using the 1 second auction window for auctions used in the active and passive approach in scenario 1 as well as for scenario 2 (top row), the 500 millisecond auction window for auctions used in active and passive approach in the first and second scenario (middle row) and bot approaches without an auction window (bottom row).

tation in a real camera network, similar simple scenarios achieved a communication reduction around $40-45\%$ for only $10-15\%$ penalty in tracking performance. The more complex the scenarios got, the higher was the trade-off between *active* and *passive* approaches and allowed reductions in communication by as much as 90%. Interestingly, in completely random scenarios the broadcast approaches showed superiority over our
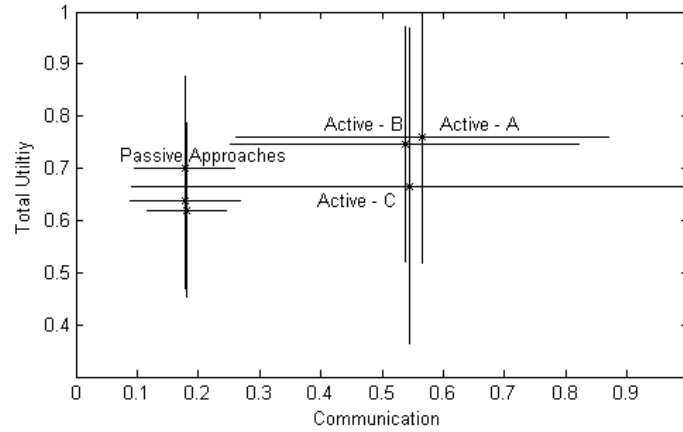
Fig. 8. Direct comparison of the different durations of auctions in our active approach versus the different auction windows in our passive approach. Again, *Active - A*, *Active - B*, and *Active - C* represent the active approach initiating an auction every third processed frame with timing windows of 1 second, 500 ms, and no timing window, respectively. It is apparent that the selection of an appropriate auction window may have a high impact to the overall performance.

0 Seconds   40 Seconds

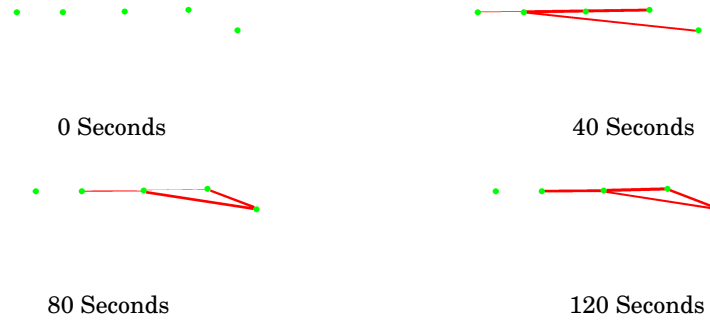80 Seconds   120 Seconds

Fig. 9. The vision graph is built up during runtime through trading interactions for scenario 2 in our real camera network. Dots indicate cameras, lines indicate links in the vision graph; thickness indicates strength.

SMOOTH and STEP approaches where *passive* broadcast had about 65% more overall utility than *passive* STEP but only about 25% more communication effort.

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 resource consumption of cameras. Especially, introducing a more constraining resource model for each camera might be very interesting. This would extend our current utility function for objects, introduce a cost for tracking objects and make the owning camera even more special. Another direction includes the modelling of the utility function over (future) time periods and the elaboration of more advanced trading mechanisms. Yet another direction is the experimental evaluation of this novel approach on larger camera networks and to explore the trade-off among performance, communication and resources in more detail.
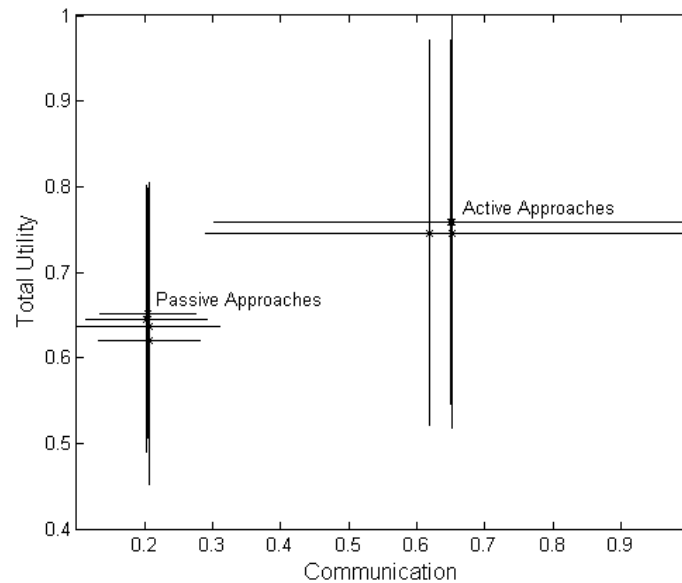
Fig. 10.   Performance (overall utility calculated over duration of scenarios) of each of the six algorithms in scenario 2. Both utility and communication values are normalised by those of the active broadcast. The trade-off between the two approaches in a real system is apparent.

ACKNOWLEDGMENTS

REFERENCES

ARDAIZ, O., ARTIGAS, P., EYMANN, T., FREITAG, F., NAVARRO, L., AND REINICKE, M. 2006. The catallaxy approach for decentralized economic-based allocation in grid resource and service markets. *Applied Intelligence 25,* 2, 131–145.

BHANU, B., RAVISHANKAR, C. V., ROY-CHOWDHURY, A. K., AGHAJAN, H., AND TERZOPOULOS, D., Eds. 2011. *Distributed Video Sensor Networks*. Springer.

BREWER, E. A. 2001. Lessons from giant-scale services. *IEEE Internet Computing 5,* 4, 46–55.

CHANDRA, A., OLIVETO, P. S., AND YAO, X. 2010. Co-evolution of optimal agents for the alternating offers bargaining game. In *Proceedings of the European Conference on the Applications of Evolutionary Computation (EvoApplications) 2010), Lecture Notes in Computer Science*. Vol. 6024. Springer, 61–70.

CHEN, C.-H., YAO, Y., PAGE, D., ABIDI, B., KOSCHAN, A., AND ABIDI, M. 2010. Camera handoff with adaptive resource management for multi-camera multi-object tracking. *Image and Vision Computing 28*, 851–864.

CHENG, Z., DEVARAJAN, D., AND RADKE, R. J. 2007. Determining Vision Graphs for Distributed Camera Networks using Feature Digests. *EURASIP J. Appl. Signal Process. 2007,* 1, 220–220.

CLEARWATER, S. H., Ed. 1996. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, Singapore.

CLIFF, D. 1997. Minimal-intelligence agents for bargaining behaviors in market-based environments. Tech. Rep. HPL-97-91, HP Labs, Bristol, UK.

CLIFF, D. AND BRUTEN, J. 1998. Simple bargaining agents for decentralized market-based control. Tech. Rep. HPL-98-17, HP Laboratories, Bristol, UK.

DETMOLD, H., VAN DEN HENGEL, A., DICK, A., CICHOWSKI, A., HILL, R., KOCADAG, E., FALKNER, K., AND MUNRO, D. S. 2007. Topology Estimation for Thousand-Camera Surveillance Networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Vienna, Austria, 195–202.

DORIGO, M. AND STÜTZLE, T. 2004. *Ant Colony Optimization*. The MIT Press.

ELLIS, T. J., MAKRIS, D., AND BLACK, J. K. 2003. Learning a multi-camera topology. In *Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 165–171.

ERDEM, U. M. AND SCLAROFF, S. 2005. 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, 105–110.

ESTERLE, L., LEWIS, P., BOGDANSKI, M., RINNER, B., AND YAO, X. 2011. A Socio-Economic Approach to Online Vision Graph Generation and Handover in Distributed Smart Camera Networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. 1 –6.

ESTERLE, L., LEWIS, P. R., RINNER, B., AND YAO, X. 2012. Improved Adaptivity and Robustness in Decentralised Multi-Camera Networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Hong Kong.

EYMANN, T. 2001. Co-evolution of bargaining strategies in a decentralized multi-agent system. In *AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems*. Falmouth, MA, 126–134.

FRIEDMAN, D. P. AND RUST, J. 1993. *The Double Auction Market: Institutions, Theories, and Evidence*. Westview Press, Boulder, Colorado, USA.

GERDING, E. H., DASH, R. K., YUEN, D. C. K., AND JENNINGS, N. R. 2006. Optimal bidding strategies for simultaneous vickrey auctions with perfect substitutes. In *Proceedings of the 8th Workshop on Game Theoretic and Decision Theoretic Agents*. Hakodate, Japan, 1017.

GERDING, E. H., DASH, R. K., YUEN, D. C. K., AND JENNINGS, N. R. 2007. Bidding optimally in concurrent second-price auctions of perfectly substitutable goods. In *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-07)*. Honolulu, Hawaii, USA, 267–274.

GERDING, E. H. AND LA POUTRÉ, J. A. 2006. Bilateral bargaining with multiple opportunities: Knowing your opponent's bargaining position. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 36,* 1, 45–55.

GUPTA, A., STAHL, D. O., AND WHINSTON, A. B. 1999. The economics of network management. *Communications of the ACM 42,* 9, 57–63.

HAUSHEER, D. AND STILLER, B. 2005. Peermart: The technology for a distributed auction-based market for peer-to-peer services. In *Proceedings of the IEEE International Conference on Communications*. Vol. 3. 1583–1587.

JAGANNATHAN, S. AND ALMEROTH, K. C. 2002. Price issues in delivering e-content on-demand. *ACM SIGecom Exchanges 3,* 2, 18–27.

KETCHAM, J., SMITH, V. L., AND WILLIAMS, A. W. 1984. A comparison of posted-offer and double-auction pricing institutions. *The Review of Economic Studies 51,* 4, 595–614.

KIM, H., ROMBERG, J., AND WOLF, W. 2009. 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, 1 –8.

LEWIS, P. R., MARROW, P., AND YAO, X. 2010. Resource Allocation in Decentralised Computational Systems: An Evolutionary Market Based Approach. *Journal of Autonomic Agents and Multi-Agent Systems 21,* 2, 143–171.

LI, Y. AND BHANU, B. 2009. 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, 1–8.

LI, Y. AND BHANU, B. 2011. Utility-based Camera Assignment in a Video Network: A Game Theoretic Framework. *IEEE Sensors Journal 11,* 3, 676–687.

LOPES, F., WOOLDRIDGE, M., AND NOVAIS, A. Q. 2008. Negotiation among autonomous computational agents: principles , analysis and challenges. *Artificial Intelligence Review 2008,* 29, 1–44.

LOWE, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60,* 2, 91–110.

MAKRIS, D., ELLIS, T., AND BLACK, J. 2004. Bridging the Gaps between Cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 205–210.

MANDEL, Z., SHIMSHONI, I., AND KEREN, D. 2007. Multi-Camera Topology Recovery from Coherent Motion. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*. Vienna, Austria, 243–250.

MARINAKIS, D. AND DUDEK, G. 2006. A Practical Algorithm for Network Topology Inference. In *Proceedings of IEEE International Conference on Robotics and Automation*. 3108–3115.

MÖLLER, B., PLÖTZ, T., , AND FINK, G. A. 2008. Calibration-free Camera Hand-Over for Fast and Reliable Person Tracking in Multi-Camera Setups. In *Proceedings of the 19th International Conference on Pattern Recognition*. Tampla, FL, USA, 1–4.

MORIOKA, K., KOVACS, S., LEE, J.-H., AND KORONDI, P. 2010. A Cooperative Object Tracking System with Fuzzy-Based Adaptive Camera Selection. *International Journal on Smart Sensing and Intelligent Control 3,* 3, 338–358.

NIU, J., CAI, K., PARSONS, S., GERDING, E., MCBURNEY, P., MOYAUX, T., PHELPS, S., AND SHIELD, D. 2008. JCAT: a platform for the TAC market design competition. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: Demo papers*. International Foundation for Autonomous Agents and Multiagent Systems, 1649–1650.

PHELPS, S., MCBURNEY, P., AND PARSONS, S. 2009. Evolutionary mechanism design: a review. *Autonomous Agents and Multi-Agent Systems 21,* 2, 237–264.

PHELPS, S., PARSONS, S. D., AND MCBURNEY, P. 2004. An evolutionary game-theoretic comparision of two double-auction market designs. In *Proceedings of the 6th Workshop on Agent Mediated Electronic Commerce*. New York, NY, USA, 101–114.

PLOTT, C. R. AND SMITH, V. L. 1978. An experimental examination of two exchange institutions. *The Review of Economic Studies*, 133–153.

QUARITSCH, M., KREUZTHALER, M., RINNER, B., BISCHOF, H., AND STROBL, B. 2007. Autonomous Multicamera Tracking on Embedded Smart Cameras. *EURASIP Journal on Embedded Systems Volume 2007*, 10.

QURESHI, F. AND TERZOPOULOS, D. 2008. Multi-camera Control through Constraint Satisfaction for Persistent Surveillance. In *IEEE Conference on Vision and Signal-based Surveillance*. Santa Fe, USA, 211–218.

RADKE, R. J. 2010. A survey of distributed computer vision algorithms. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, 35–55.

RINNER, B., DIEBER, B., ESTERLE, L., LEWIS, P. R., AND YAO, X. 2012. Resource-aware configuration in smart camera networks. In *Proc. Int. IEEE Workshop on Camera Networks and Wide Area Scene Analysis (CVPRW)*. IEEE, 1–8.

RINNER, B. AND WOLF, W. 2008. Introduction to Distributed Smart Cameras. *Proceedings of the IEEE 96,* 10, 1565–1575.

ROBINSON, E., MCBURNEY, P., AND YAO, X. 2010. How Specialised Are Specialists? Generalisation Properties of Entries from the 2008 and 2009 TAC Market Design Competitions. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, E. David, E. Gerding, D. Sarne, and O. Shehory, Eds. Lecture Notes in Business Information Processing Series, vol. 59. Springer Berlin Heidelberg, 178–194.

SONG, B., KAMAL, A. T., SOTO, C., ROY-CHOWDHURY, A. K., AND FARRELL, J. A. 2010. Tracking and Activity Recognition Through Consensus in Distributed Camera Networks. *IEEE Transactions on Image Processing 19,* 10, 2564–2579.

SORO, S. AND HEINZELMAN, W. B. 2009. A Survey of Visual Sensor Networks. *Advances in Multimedia 2009*, 21.

TESFATSION, L. AND JUDD, K. 2006. *Handbook of Computational Economics*. Elsevier.

VICKREY, W. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance 16,* 1, 8–37.

WALDSPURGER, C. A., HOGG, T., HUBERMAN, B. A., KEPHART, J. O., AND STORNETTA, S. 1992. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering 18,* 2, 103–117.

WOLSKI, R., PLANK, J., BREVIK, J., AND BRYAN, T. 2001. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications 15,* 3, 258.

YILMAZ, A., JAVED, O., AND SHAH, M. 2006. Object Tracking: A Survey. *ACM Computing Surveys 38,* 4, 1–45.