# Distributed Resource-aware Task Assignment for Complex Monitoring Scenarios in Visual Sensor Networks


Bernhard Dieber*†, Lukas Esterle*, Bernhard Rinner*

*Institute of Networked and Embedded Systems

Alpen-Adria Universität Klagenfurt, Austria

Email: {firstname.lastname@aau.at}

†Lakeside Labs

Klagenfurt, Austria



***Abstract*—Smart camera networks usually have limited resources. This includes not only processing power and memory but in many applications also the amount of available energy. To find a trade-off between available resources and the current requirements of the application, the configuration of the network is very important. Due to rapid changes in the visual sensor network's environment, reconfiguration must be dynamic and performed online. The reconfiguration must also be able to deal with heterogeneous tasks assigned within the network where the tasks require varying levels of dynamic actions by the cameras.**

**In this paper we present a fully distributed algorithm for a combined spatial coverage and object handover problem. In our approach we focus on (i) assigning heterogeneous tasks to cameras in order to cover a certain area of interest (ii) hand over moving objects between cameras to allow seamless tracking and (iii) perform (i) and (ii) in a combined fashion to maximize the overall quality of surveillance while minimizing the resources required during operation.**


## I. Introduction

Visual Sensor Networks (VSN) where cameras integrate sensing, image processing and communication into a single platform are adopted for an increasing number of applications. Along with the increasing pervasiveness comes the need for efficient management of the limited resources on the platform. This is especially true in cases where cameras are powered by batteries or unreliable energy sources like solar power. To tackle this problem, we need to dynamically allocate the available resources to achieve the monitoring tasks in the most efficient way.

In this paper, we extend our distributed algorithm for spatial coverage and sensor selection [1] with our market-based object handover mechanism [2] to form a hybrid, resource-aware approach to coverage and handover. With our approach, we are able to find resource-sparing solutions to complex but common tasks in visual monitoring. We provide coverage with different surveillance activities like motion detection or tracking and are able to handle the handover of tracked objects between cameras in a resource-aware way. We model the problem using separate representations for tasks that need object handover and tasks that perform spatial coverage (e.g. movement detection). We evaluate our approach by showing the basic mechanics of the algorithm and by comparing it to a benchmark algorithm in a larger scenario.

The rest of the paper is organized as follows. Section II discusses relevant related work. In Section III we describe our problem and its model in more detail and present our approach in Section IV. We evaluate our solution in Section V and conclude in Section VI.

## II. Related Work

Resource limitation is a typical problem many multi-camera networks have to face [3].

Maier et al. [4] aim at finding a tradeoff between energy consumption on nodes and the surveillance service delivered. Casares et al. [5] and Yu et al. [6] present methods for load management on visual sensor network nodes.

Chen et al. [7] propose an adaptive resource management mechanism for camera handoff incorporating the number of currently tracked objects on a node and the resources needed to track an additional one. Other strategies for resource management in camera networks are presented in [8], [9], [10].

Selecting the best camera for a particular task is crucial in a visual sensor network. Strategies for task assignment and object handover further influence the overall resource consumption in a visual sensor network. Cenedese et al. [11] formalize the problem of task assginment of multiagent-driven camera networks and present a decentralized algorithm to solve it. In [12] many approaches to this problem are compared and a game theoretic approach to its solution is shown in [13]. Further approaches for task assignment and sensor selection have been proposed in [14] [15].

Market-based control was first introduced by Clearwater et al. [16]. Since then a wide variety of applications in different domains have been proposed. Gupta et al. [17] propose to price internet traffic based on usage and user QoS requirments. Their pricing approach does not limit the quantities but lets the user self-select the quantities at a certain price. Cliff et al. [18] explains how to allocate and control resources in a mulit-agent system using marked-based control. The similarities between market-based control systems and real economies regarding decentralization, robustness and self-organiziation are pointed out. An evolutionary marked-based approach employing self-interested agents to allocate resources is presented by Lewis et al. [19]. Their agents have only private information available and do not assume cooperation between the agents nor have any centralized coordination.

In contrast to the approaches listed above which address single parts of the problem described in this paper, we present a holistic approach to spatial coverage and object handover.

## III. Problem Definition

This section presents a concise definition of our problem. A more detailed problem formulation can be found in [20], [1], [2].

### A. Overview and Assumptions

We consider the problem of simultaneous coverage and object tracking in sensor networks. In a VSN where cameras perform not one but multiple different tasks (like movement detection, object tracking or object identification) in different parts of the area of interest, a method for determining an optimal allocation of tasks to cameras must be found. In our case, we consider an optimal task allocation as fulfilling the requirements for coverage (i.e. performing a certain task at a certain level of quality) in different areas while minimizing the resource demands on the cameras. This includes not only assigning certain tasks to cameras but also configuring cameras according to required surveillance quality.

The properties of some tasks in this network change slowly. The task of performing motion detection may change its quality requirements or location at some time e.g. based on operator input or activity maps [21]. However, those changes are assumed to occur infrequently.

Other tasks like object tracking require highly dynamic action by the cameras. As the object moves, a camera must determine when to hand over the object to a neighboring camera (assuming a shared field of view) before losing it.

Figure 1 depicts our problem. A set $S$ of $n$ camera sensors is placed on a 2D space such that they have a partially shared FOV; the coverage area of each camera is represented by a segment. Within the field of views of the cameras, there are two types of targets:

*i) Observation points* are points in space where the VSN should pay attention to. They require certain surveillance activities (e.g. background subtraction or object detection) at a certain quality (expressed in pixels on target and frames per second). Observation points are used to model spatial coverage requirements with certain activity and quality requirements. Observation points may change over time (either their requirements or their location) but are assumed to change slowly. The set of all observation points $t_i$ is called $T$.

*ii) Objects to track* share the quality requirements of observation points but represent agile objects in the observed area. They have high dynamics, thus the sensor network must react quickly in order to not loose track of objects. If an object approaches the borders of the field of view of a camera, it must be handed over to a neighboring camera. In this handover procedure, the cameras must decide which of them is best suitable for tracking the object, i.e. it must deliver the required quality but must also try to save its resources (computational capacity, memory or energy reserves). The set of all objects to track is called $O$.

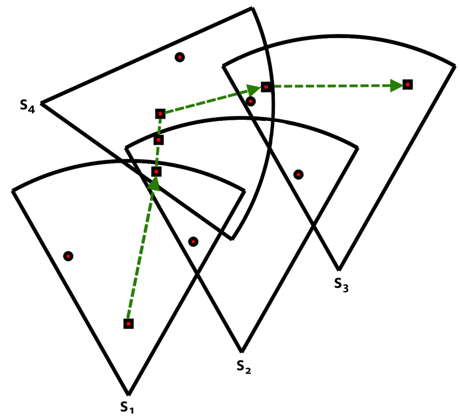



Fig. 1. A sample scenario with cameras $s_1 - s_4$ depicted as segments, five observation points represented as circles and one object with its trajectory.

In this combined coverage and handover problem we want to minimize the resource costs inflicted by static tasks (such as background subtraction) as wells as tasks with dynamic portions (such as tracking). However, resource consumption is not the only optimization criterion. We also include the surveillance quality expressed as minimum pixels on target and framerate for all objects and targets. Hence, we solve a multi-criterion optimization problem in two dimensions where we find the tradeoff between surveillance quality and resource consumption. To be able to perform the reconfiguration at runtime, a distributed approach is required.

## IV. A hybrid resource-aware task assignment and handover algorithm

In this Section we describe our combined task assignment and object handover algorithm. We tightly integrate the socio-economic object handover algorithm described in [2] with the distributed coverage and task assignment algorithm presented in [1]. The network performs a long-term reconfiguration for tasks with low dynamics (represented as observation points). For highly dynamic objects that are tracked by the cameras, we additionally perform object handover between cameras as necessary. However, situations might occur, where the resource allocation at a certain camera does not permit to track an additional object. In this case, the camera will try to perform a reconfiguration in order to free resources before entering the auctioning process.

Figure 2 shows a pseudo-code description of our algorithm. As described in [1], the distributed coverage and task assignment algorithm uses descriptors to exchange information between nodes. A descriptor is a small data packet containing only the identifier for the sender node, information about the

```
Algorithm distributed_task_assignment_and_handover()

On define new observation point t:
  if t can be covered:
    Calculate required (res, fps, activity)
    Calculate required resources for (res, fps, activity)
    Broadcast descriptor
  fi

On receive descriptor d for target t:
  if t is not in FOV
    end
  fi
  if d is already stored as best solution
    end
  fi
  if no queue for t exists:
    create queue q_t for t
    set timer for new queue
  else
    add d to q_t
    restart timer
  fi

On timer for q_t elapsed:
  take best descriptor d from q_t
  if better descriptor d_s available (local / stored solution)
    Broadcast d_s
  else
    Store d as remote best descriptor for t
    Broadcast d
  fi
  for each pending auction for o_i:
    Calculate required (res, fps, activity) to cover o_i
    Calculate required resources for (res, fps, activity)
    if required resources are within limits
      send bid for o_i
  end for

On handover necessary for object o_i:
  initiate auction for o_i

On receive auction initialization for o_i:
  if o_i is visible:
    Calculate required (res, fps, activity) to cover o_i
    Calculate required resources for (res, fps, activity)
    if required resources are within limits
      send bid for o_i
    else
      Find most expensive target t_j:
      Init reconfiguration for t_j
    fi
  fi

Do periodically:
  if uncovered target in range
    if it can be covered
    Calculate required (res, fps, activity)
    Calculate required resources (res, fps, activity)
    Broadcast descriptor
    fi
  fi

Do periodically:
  Select target t from targets covered by this node
  Send out descriptor for t
```

Fig. 2. Event-based pseudo code of the distributed algorithm. "On $x$" indicates the occurrence of event $x$ on the node. Events for new observation points or new descriptors are shown along with optional periodic activities for optimization.

observation point and the resources needed to cover it. During the operation of the VSN, nodes can detect hotspots that need additional surveillance activity (e.g. shared paths of tracked objects or areas with high movements). For those spots, they are able to define new observation points with the respective surveillance requirements. However, observation points could also be introduced by operators. For a new observation point, the defining node calculates and broadcasts an initial descriptor[1]. On receiving such a descriptor, a node will buffer it for a short period of time to be able to receive further descriptors for that point. After this period, it will evaluate all received descriptors and compare them to its own solution. The best descriptor from this operation is then accepted as the solution and is stored and broadcast. In addition, the algorithm will periodically improve the solution and try to re-arrange the observation points assignments.

Further, this algorithm is now extended with the object handover algorithm [2]. This is done to enhance long-term reconfiguration with the capability for handling moving objects in tracking algorithms. A node tracking an object can decide to perform a handover. If the handover requires a node to free resources, it will start a reconfiguration for its observation points.

### A. Handover via Auctions

In the handover algorithm we use the passive approach as described in [2] where an auction is initiated whenever the tracked object is about to leave the FOV of the camera responsible for tracking it. In this case, the camera will send a message including an object description to initiate the auction . After waiting the auction timeout interval, the auctioneering camera compares all received bids and hands the object over to the winning camera. We use Vickrey auctions [22] where the camera with the highest bid receives the object for the second highest price.

### B. Calculating the Bid

On receiving an auction initialization message, a camera determines if it is able to track this object $o$. If so, it will calculate the required sensor settings and resource requirements for performing this task. This is done using models of the hardware platform and of the surveillance activity (see [20] for details). In case the object can be covered within the resource constraints, the camera calculate the utility $u(o)$ for the object to express its valuation according to Equation (1). Thereafter, the camera will transmit a bid containing the utility as the offered amount of *money*. As usual in marked-based control systems, currency is an artifical construct and only used for

[1]Note, that we describe the coverage and task assignment algorithm to be using broadcast communication. However, if the handover algorithm has built up a suitable vision graph, we can use this vision graph to switch to multicast operation in order to reduce the number of messages required.

management purposes; no real money is exchanged.

$$u(o) = e_n(o) \cdot pot_n(o) \cdot fps_n(o) \tag{1a}$$
$$e_n(o) = (e_{cur}(o)/(e_{max} - e_{min}) \tag{1b}$$
$$pot_n(o) = (pot_{cur}(o)/pot(o)) \tag{1c}$$
$$fps_n(o) = (fps_{cur}(o)/fps(o)) \tag{1d}$$

In Equation (1a) we use normalized values for energy consumption $e_n(o)$, framerate $fps_n(o)$ and pixels on target $pot_n(o)$ (indicated by a subscript $_n$) to calculate a bid according to the resources required for tracking and the quality a camera can guarantee. The energy consumption in 1b is normalized between 0 and 1 using the minimum $e_{min}$ and maximum $e_{max}$ energy consumption on this node. $e_{cur}$ describes the required energy consumption for covering the targets assigned to that node plus the energy required for additionally tracking $o$.

The normalized quality indicators $pot_n$ and $fps_n$ in (1c) are calculated by using the required framerate and pixels on target of $o$ ($fps(o)$ and $pot(o)$) compared to the currently delivered quality ($pot_{cur}(o)$ and $fps_{cur}(o)$) that consider additionally its assigned tasks.

From Equation (1b) it can be seen that the value for $e_n(o)$ ranges from 0 to 1 while Equations (1c) and (1d) show that the values for $pot_n(o)$ and $fps_n(o)$ are relative to the quality required by $o$ (where a value of 1 means that the requirements are exactly met). This is done to enable networks of heterogeneous sensors. If the pixels on target and framerate were normalized between 0 and 1, lower utility values would be calculated for sensors with high hardware capabilities.

### C. Reconfiguration on Handover

If a camera receives an auction initialization but tracking the object would consume more resources than available, it initializes a reconfiguration for one of its assigned observation points. If it can free resources after the reconfiguration, it will participate in the auction.

To enable a timely finish of the reconfiguration, the buffering interval for descriptors should be significantly smaller than the duration of an auction. We propose at least a factor of 2 for this.

## V. Evaluation

We evaluate our approach by first showing the advantage in terms of energy demand of using our approach compared to always tracking with the highest possible quality. Second, we show that the reconfiguration can be used to free resources on cameras that would otherwise be unable to track an object. Finally, we compare our approach to a centralized approach in a larger scenario. To show the basic mechanics in our approach, we take the scenario shown in Figure 1 with four cameras with partially overlapping FOV. In the first scenario, we remove all observation points. The results shown below were obtained by means of simulation.

The moving objects in all scenarios require tracking at eight frames per second and with 14 pixels on target. These quality requirements must be met in order to achieve a feasible solution. Tracking at higher settings will result in improved quality but also higher resource demands. We take six snapshots (steps) of the scenario, calculate the benchmark results and compare them to the results of our approach (in-between steps we assume continuous tracking). Whenever an objects comes close to the edge of the FOV of a camera, the node will try to perform a handover.

We calculate the utility for the tracked objects according to Equation 1a and the predicted energy consumption in the total network (according to our platform and algorithm models) in each step.

Figure 3 shows that the reduction of consumed energy for our first scenario is apparent. While the total utility is lower in our approach, the object is still tracked at its required quality and the energy consumption is 45% lower.

In the second scenario, camera $S_3$ is covering two observation points with high quality requirements. If no dynamic reconfiguration is applied, the camera is unable to additionally track the object when it enters its field of view. Thus, the system looses track of the object which results in a zero utility. Figure 4 shows how the system adapts the task allocation from before and after step 5 in order to free resources on $S_3$.

The third scenario is more complex than the first two and defines ten sensors and eight fixed observation points. Over 10 steps, up to six objects move concurrently in the area of interest. The sensors are arranged to cover an L-shaped area that could e.g. be a large corridor in a building or a pathway between buildings. The setup is shown in Figure 5.

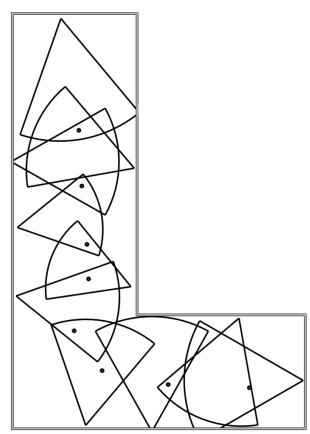


Fig. 5. Scenario 3 defines ten sensors and eight fixed observation points in an L-shaped area of interest.

We take the centralized algorithm presented in [20] as benchmark. The optimal assignment calculated by the centralized algorithm for each step is compared to the results of our approach to that. The central algorithm treats the moving objects as being static since it only considers static scenarios (it does not take care of object movement and handover but its input is $O \cup T$). However, the optimal solution for each step is

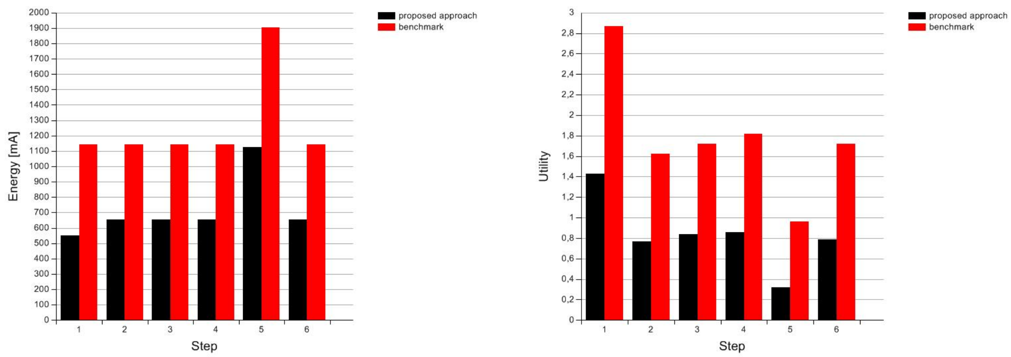
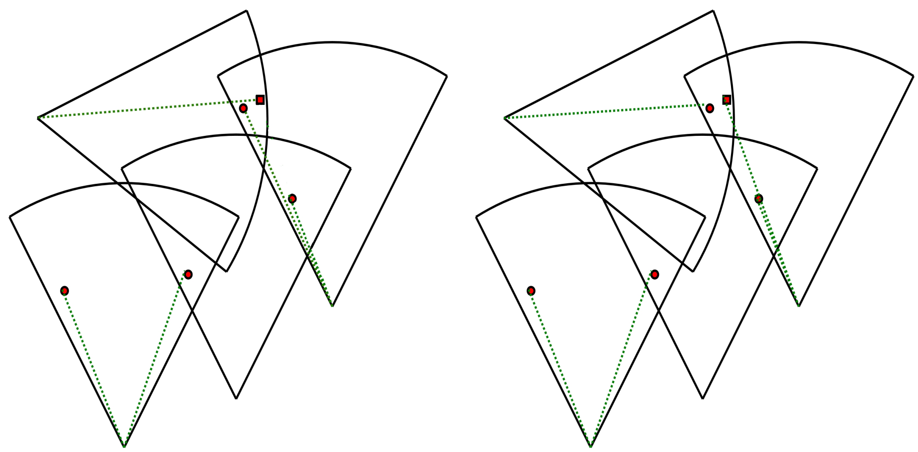



Fig. 3. The energy consumption and utility in scenario one compared to the benchmark.



(a) State before handover.

(b) State after handover and reconfiguration.

Fig. 4. The task allocation before and after the reconfiguration in step 5 in scenario 2.

a benchmark for the minimum energy consumption possible. The results of the proposed approach differ in several cases because an object has been handed over based on its movement direction.

In each step, we compare the total utility (the sum of the delivered utilities of all objects) and the total energy consumption in the network to the central algorithm. Further, we show how many times the network performs a handover or reconfiguration operation.

Scenario 3 results are shown in Figure 6. We always reach at least the benchmark utility but the handover causes increased energy demand is in some cases. The average total utility was 202% higher than in the benchmark solution but the average energy demand is increased by only 4.2%. Thus, the hybrid algorithm combines the ideas of reconfiguration for slow dynamics with handover for agile objects and still operates with nearly the same energy demand.

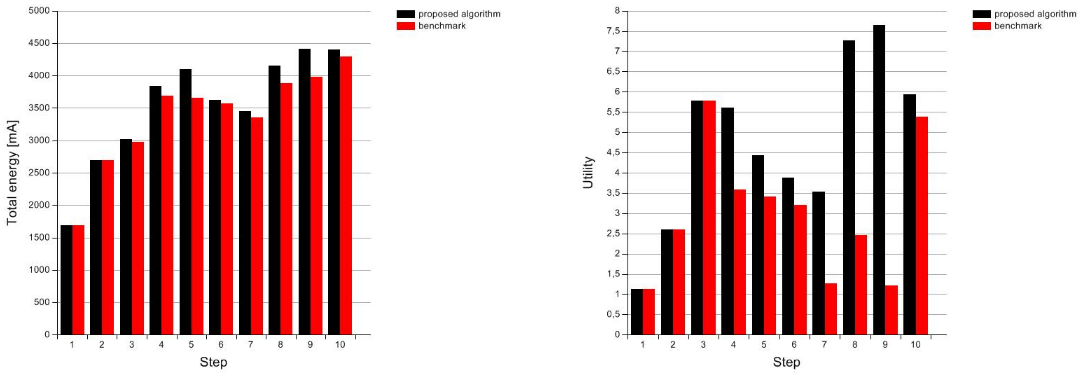
Table I shows the actions that have been taken in the single steps by our algorithm. It can be seen that several handover operations have been performed and in some cases also reconfiguration due to high object densities were necessary.

| Step | # Handovers | # Reconf. |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 1 | 0 |
| 4 | 2 | 0 |
| 5 | 3 | 1 |
| 6 | 3 | 1 |
| 7 | 2 | 0 |
| 8 | 3 | 1 |
| 9 | 3 | 0 |
| 10 | 3 | 0 |

TABLE I
THE NUMBER OF HANDOVERS AND RECONFIGURATIONS IN EACH STEP.

## VI. Conclusion

In this paper we have presented a combined approach for resource-aware task assignment for handling complex challenges in visual sensor networks. We have combined the idea of a distributed algorithm for combined sensor selection and task assignment for observation points with low agility with a socio-economic handover algorithm for dynamic objects. The




(a) The overall energy demand in the network in the proposed solution compared to the benchmark

(b) The total utility in the proposed approached compared to the benchmark solution.

Fig. 6. Comparison between the proposed solution and the benchmark for scenario 3.

algorithm determines for each handover operation the need to free resources on a target camera. We showed that the approach yields good results for continuously tracking an object and only requires some percent of additional energy to perform.

## Acknowledgment

This work is supported by Lakeside Labs GmbH, Klagenfurt, Austria and is funded in part by the European Union Seventh Framework Programme under grant agreement nº 257906, the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF 20214/18354/27107

## References

[1] B. Rinner, B. Dieber, L. Esterle, P. Lewis, and X. Yao, “Resource-aware configuration in smart camera networks,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, june 2012, pp. 58 –65.

[2] L. Esterle, P. Lewis, M. Bogdanski, B. Rinner, and X. Yao, “A socio-economic approach to online vision graph generation and handover in distributed smart camera networks,” in *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*, aug. 2011, pp. 1 –6.

[3] Hamid Aghajan and Andrea Cavallaro, Ed., *Multi-Camera Networks*. Elsevier, 2009.

[4] A. Maier, B. Rinner, W. Schriebl, and H. Schwabach, “Online Multi-Criterion Optimization for Dynamic Power-Aware Camera Configuration in Distributed Embedded Surveillance Clusters,” in *Proceedings of the IEEE 20th International Conference on Advanced Information Networking and Applications (AINA 06)*, Vienna, Austria, Apr. 2006, pp. 307–312.

[5] M. Casares and S. Velipasalar, “Adaptive Methodologies for Energy-Efficient Object Detection and Tracking With Battery-Powered Embedded Smart Cameras,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1438–1452, oct. 2011.

[6] C. Yu and G. Sharma, “Camera Scheduling and Energy Allocation for Lifetime Maximization in User-Centric Visual Sensor Networks,” *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2042–2055, 2010.

[7] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi, “Camera Handoff with Adaptive Resource Management for Multi-Camera Multi-Target Surveillance,” in *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 79 – 86.

[8] H.-P. Shiang and M. van der Schaar, “Information-Constrained Resource Allocation in Multicamera Wireless Surveillance Networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 505–517, 2010.

[9] Y. Durmus, A. Ozgovde, and C. Ersoy, “Distributed and Online Fair Resource Management in Video Surveillance Sensor Networks,” *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2011.

[10] D. R. Karuppiah, R. A. Grupen, Z. Zhu, and A. R. Hanson, “Automatic resource allocation in a distributed camera network,” *Machine Vision and Applications*, vol. 21, pp. 517–528, 2010.

[11] A. Cenedese, F. Cerruti, M. Fabbro, C. Masiero, and L. Schenato, “Decentralized task assignment in camera networks,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, dec. 2010, pp. 126 –131.

[12] Y. Li and B. Bhanu, “A comparison of techniques for camera selection and handoff in a video network,” in *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, 30 2009-sept. 2 2009, pp. 1 –8.

[13] ——, “Task-oriented camera assignment in a video network,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, nov. 2009, pp. 3473 –3476.

[14] L. Tessens, M. Morbee, H. Lee, W. Philips, and H. Aghajan, “Principal view determination for camera selection in distributed smart camera networks,” in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, sept. 2008, pp. 1 –10.

[15] E. Shen and R. Hornsey, “Camera selection using a local image quality metric for a distributed smart camera network,” in *Sensors, 2011 IEEE*, oct. 2011, pp. 1217 –1220.

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

[17] A. Gupta, D. O. Stahl, and A. B. Whinston, “The economics of network management,” *Communications of the ACM*, vol. 42, no. 9, pp. 57–63, September 1999.

[18] D. Cliff and J. Bruten, “Simple bargaining agents for decentralized market-based control,” HP Laboratories Bristol, Tech. Rep. HPL-97-91, 1998.

[19] 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.

[20] B. Dieber, C. Micheloni, and B. Rinner, “Resource-aware coverage and task assignment in visual sensor networks,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 10, pp. 1424 –1437, oct. 2011.

[21] C. Piciarelli, C. Micheloni, and G. L. Foresti, “Occlusion-aware multiple camera reconfiguration,” in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, 2010, pp. 88–94.

[22] W. Vickrey, “Counterspeculation, Auctions, and Competitive Sealed Tenders,” *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.