Distributed Adaptive Task Allocation for Energy Conservation in Camera Sensor Networks

Christos Kyrkou, Theocaris Theocarides, Christos Panayiotou, Marios Polycarpou
Department of Electrical and Computer Engineering
KIOS Research Center for Intelligent Systems and Networks
University of Cyprus
{kkyrkou.christos, tttheocarides, christosp, mpolycar} @ ucy.ac.cy

ABSTRACT
Camera Sensor Networks (CSNs) have a large and diverse application spectrum ranging from security and safety-critical applications, to industrial monitoring, and augmented reality. Cameras in such networks are equipped with real-time multitasking processors and communication infrastructure, which enables them to perform various computer vision tasks in a distributed and collaborative manner. In many cases, the cameras in the network operate under limited or unreliable power sources. Therefore in order to extend the CSN lifetime it is important to manage the energy consumption of the cameras, which is related to the workload of the vision tasks they perform. Hence by managing and assigning vision tasks to cameras in an energy-aware manner it is possible to extend the network lifetime. In this paper we address this problem by proposing a distributed market-based solution where cameras bid for tasks using an energy-aware utility function. An additional novelty of the proposed solution is that as the cameras can adapt their bidding strategy based on their remaining energy levels. The results for different CSN configurations and setups show that the proposed methodology can increase network lifetime by 10%-30% while improving the number of dynamic and static tasks being monitored by 30-50%.

Categories and Subject Descriptors
• Networks~Network resources allocation • Networks~Sensor networks • Computing methodologies~Computer vision • Computing methodologies~Multi-agent systems

Keywords
Camera Sensor Networks; Distributed Task Allocation; Resource Management; Self-Reconfiguring Camera Networks

1. INTRODUCTION
Camera Sensor Networks (CSNs) have attracted considerable attention from academia as well as industry. Such systems consist of networked cameras that can communicate and perform multiple vision tasks while monitoring an area [1], [2]. The applications of CSNs encompass the domains of intelligent video surveillance systems, intelligent transportation systems, personalized medicine, and home entertainment [3], [4]. In such rapidly changing environments, smart cameras need to communicate in order to collaboratively assign areas and targets of interest amongst each other and execute higher-level tasks such as activity recognition, motion detection, or tracking and localization [5]. As an example, in public-space monitoring applications, cameras need to perform vision tasks such as target tracking for suspicious individuals and detecting motion in entrances/exits and other areas (Fig. 1). The vision tasks need to be performed with constraints regarding energy, computational resources, and real-time performance. These constraints may change over time depending on various factors; hence, the network needs to reconfigure over time and adapt to these changes [6].

Energy consumption, in particular, is of critical importance especially in the case of battery-operated embedded camera vision systems and in environments with little infrastructure, i.e., limited power supply. As such, the need arises to efficiently manage the available resources of each camera in a CSN to preserve energy and in turn maximize the lifetime of the network, without however, degrading the monitoring performance. Hence, any vision task that the network needs to carry out needs to be allocated amongst the cameras in a way that takes into consideration key constraints including energy, but at the same time also meet specific task requirements such as resolution and frame-rate. Attempts to address the problem of energy-based task allocation assume that all cameras execute the same task (usually target tracking) and hence the requirements of each task does not change [13]. In other cases however, where multiple tasks can be executed by the cameras, other works (e.g. [11]) do not consider energy as a key task assignment constraint and focus only on task-related performance metrics.

This paper addresses the problem of adaptive energy-aware task allocation in heterogeneous CSNs in two ways. First, the problem for multi-task allocation in CSNs is formulated, and a distributed adaptive market-based bidding solution is proposed that simultaneously considers the energy-level of a camera, its processing resources, and the requirements of the tasks in terms of frame-rate and resolution to determine which camera is more appropriate to handle each vision task. Since, we do not focus on a specific application, e.g. tracking, the methodology is generic and can be used for different types of tasks. Second, a mechanism is proposed that enables a camera to adapt its bidding strategy according to its current state in an attempt to further increase its lifetime. The proposed approach is evaluated using a fully simulated environment in MATLAB with different CSN configurations consisting of heterogeneous cameras and dynamic as well as static tasks. Using the proposed task allocation approach results indicate that the lifetime of CSNs can be improved between 10%-30% depending on the network configuration as well as amount of activity, while also being able to improve the number of monitored targets/areas by 30-50% for the different networks.
The rest of the paper is organized as follows. Section 2 discusses related work with regards to dynamic CSN reconfiguration and task allocation. Section 3 outlines the formulation and methodology of the proposed distributed market-based task allocation approach and adaptive bidding strategy. Section 4 discusses the evaluation process and presents the results. Finally, Section 5 provides the conclusions of this work and outlines plans for future research.

2. RELATED WORK

Emerging research in CSNs concerns the development of algorithms for dynamic reconfiguration and vision task allocation. A centralized evolutionary algorithm has been proposed in [8] that simultaneously addresses task assignment and area coverage. Essentially, it uses a two-step approach to first select cameras for area coverage and then assign tasks to cameras by considering only the frame-rate requirements of the tasks but not the energy resources of each camera. In [10] authors propose a decentralized approach for task assignment formulated as a stable marriage problem. The proposed approach is more suitable for one-to-one task assignment and cannot be used in cases where a single camera can execute multiple tasks. A distributed algorithm that simultaneously considers area coverage of the cameras and subsequently target handover for homogeneous cameras is presented in [11]. However, it only considers the tracking task and only for a small number of objects that can move within the area of observation. The centralized and distributed approaches outlined in [12] attempt to reallocate targets to cameras while managing their resources and also try to minimize the communication costs between them, rather than minimizing energy consumption to maximize their lifetime. Finally, the work in [13] attempts to perform energy-aware task assignment; however, it focuses only on target tracking with homogeneous cameras. In addition, the work does not directly consider energy or frame-rate in the utility function. Furthermore, the processing load is given only in terms of observed targets while in our work we provide a more general formulation based on camera processing resources. Overall, most existing works assume that cameras are homogeneous, execute only the tracking task, and do not consider energy levels for task allocation purposes. In contrast, in our work we incorporate energy levels and resources with resolution and frame-rate requirements in an adaptive distributed market-based task allocation algorithm in order to extend the lifetime of CSNs, which are comprised of heterogeneous cameras that can execute different vision tasks with varying requirements.

3. MODELLING AND PROBLEM FORMULATION

In the proposed formulation we consider the problem of energy-aware multi-task allocation in a camera network consisting of $N_c$ static camera nodes $C_j$ that belong in the set $C = \{C_1, C_2, ..., C_{N_c}\}$ in a 2D field (Fig. 2), and $N_T$ vision tasks $T_i$ in the set $T = \{T_1, T_2, ..., T_{N_T}\}$ that need to be executed and each task is related to either a target or area. We are interested in the assignment $T \rightarrow C$, that will lead to the output set: $\{ (C_j, T_i) \}$ for $1 \leq j \leq N_c$, which is the allocation of tasks to cameras. This assignment is dynamically updated every iteration step $t$ as the tasks that must be performed in the network, as well as the camera resource and energy levels, change over time. In this work we are interested in the assignment that maximizes the mean network lifetime defined as $\sum_{j=1}^{N_c} L_j / N_c$, where $L_j$ is the lifetime of each camera $C_j$ (i.e. the number of iteration steps that a camera was operational). We propose a solution based on a market-based bidding process where the cameras bid for each task based on their current state and suitability for each task. The assumptions and system model are described next followed by the outline of the market-based bidding process.

3.1 Camera Sensor Model

Each visual camera sensor $C_j$ in the network is modelled with a state $S_j(t) = (E_j(t), R_j)$ that describes its remaining battery energy levels, and available computational and memory resources respectively at iteration step $t$. In general, a smart camera combines a processing unit such as an embedded processor with multitasking capabilities [8], [9], or even more specialized hardware based on FPGAs [15], with an image sensor that allows it to process images on-board. The definition of resources can be different depending on the platform and desired granularity. For instance, it can represent CPU cycle slots, threads, or computing cores and memory. As such, we do not explicitly use a specific type of resource but rather resources are abstracted and modelled as units and so we assume that there are mechanisms in place that allow each camera to estimate the available resources. Therefore, even if the underlying architecture of each camera node may be different; each node will abstract it to the percentage of available resources. Hence, the same formulation can be followed for different types of camera platforms. We follow the same approach for the vision tasks which will be described in the following section. The resources that a camera can allocate depend on the activity in its FoV and the targets in the scene. Given the allocated resources for each task a camera $C_j$ can calculate the resulting frame-rate $F_j$ it can provide each task $T_i$ based on a priori
knowledge of the detection algorithm [7], and can calculate the resolution $D_{ji}$ at which it views a task-related area/target. The network can consist of heterogeneous cameras that have different features such as different Field-of-View (FoV), different energy levels, or different capabilities in terms of resources. Finally, we assume that a camera can consistently detect and identify each target/area in its FoV by employing appropriate mechanisms and algorithms as in [17].

### 3.2 Task Model

In this work we consider the vision task that can be executed by a CSN to be one of two different types, namely static and dynamic (Fig. 2). Static tasks (e.g. monitoring of restricted area) are associated with a specific area and are executed for a certain time after which they are considered obsolete. Information regarding the location and duration can be provided by the network operator for example. Dynamic tasks, such as target detection and tracking, are associated with a moving target that can change its position resulting in being viewed by different cameras. Each vision task $T_i$ is characterized by different required resources and frame-rate. The former depends on the method used to detect and track the objects (e.g. machine learning, background elimination), and the latter on the method as well as the requirements for detection performance and speed of the object. The resource requirements, complexities, and frame-rates requirements for computer vision tasks that can be executed in a CSN given specific parameters are well documented and thus can be configured a priori [7] in look-up-table fashion. Finally, it is assumed that each camera can determine the location in x,y coordinates of each task-related area/target based on the scale size and resolution $D_{ji}$ that it is detected as well as ground plane information [7]. Each camera uses the resolution and ground plane information to coordinate with other cameras regarding common targets and area views.

### 3.3 Power Model

The power consumption model of each camera in the network is based on the utilization of the computing and memory resources for running vision task(s) and for communicating with other cameras. This is also based on what other studies have shown [14]. Also, as in other wireless sensor models, we assume that the processing power is comparable to communication power [16]. Furthermore, it is assumed that a fixed amount of power is consumed at every time instance $t$ in order to perform necessary background tasks. Hence, we use the energy consumption model shown in (1) for each camera $C_j$ to find the energy consumed $E_j^c(t)$ by each camera at iteration step $t$.

$$E_j^c(t) = P_l + P_e(R_j) + P_e(m)$$  

(1)

In this model $P_e(R_j)$ is a look-up based function, which can differ between cameras, that provides the energy consumption estimate based on utilized resources which is proportional to the vision task workload, $P_e(m)$ is also a look-up based function that gives the communication power consumption for transmitting $m$ messages to other cameras, and $P_l$ is constant power used by the camera for basic background tasks. These energy consumption figures can be different for heterogeneous cameras. Power values for these parameters can be taken from standard smart camera datasheets [18] and case studies [19]. The framework supports different values that can be adjusted to meet the specifications of the targeted platform. Equation (2) thus iteratively gives the remaining battery level of a camera.

$$E_j(t) = E_j(t-1) - E_j^c(t), \quad E_j(0) = E_j^T$$  

(2)

In this equation $E_j^T$ is the total energy capacity of camera $C_j$, and the second term is the energy consumed at each iteration step. The subtraction gives as the remaining energy of each camera.

### 3.4 Distributed Market Based Multi-Task Bidding Process

In distributed market-based approaches, task allocation happens by means of auctions where agents (cameras) play the role of bidders to gain an item (tasks) [5], [20]. This process is facilitated using the concept of utility function, which measures the ability and willingness of a camera to execute a certain task. Utility is a function that assigns a number (priority) to each camera such that if the utility is higher ($U_a > U_b$) then that camera is preferred. The utility function $U_j(T_i, t)$, shown in (3), of camera $C_j$ for task $T_i$ is given by a weighted sum of four components, which encapsulate the state of the camera as well as its suitability to meet the requirements of a task. These are the remaining energy $E_j(t)$ of a camera, its available resources $R_j$; the frame-rate $F_{ji}$ it can provide to the task and the resolution $D_{ji}$ it views a task-related area/target. Each component is normalized between $[0,1]$. This utility function was chosen as it is computationally efficient and thus is suitable for resource-constraint systems.

$$U_j(T_i, t) = W_e E_j(t) + W_r R_j + W_f F_{ji} + W_d D_{ji}$$  

(3)

Cameras first detect areas/targets of interest and exchange image descriptors in order to establish image correspondences and common FoVs, prior to the actual task allocation the. Then the cameras can coordinate to execute area/target related tasks such as intrusion detection and target tracking. The bidding process for task allocation happens in an event driven manner where a bid for a task is initiated if there is an activity change in the FoV of a camera or if a camera is no longer operational. Once the task allocation process (Algorithm 1) is initiated each node that views a task-related area/target will have to evaluate its utility function $U_j(T_i, t)$ and also send a bid to the other nodes that view it. Each camera compares its own bid with the received bids. The highest bidder takes over the task and updates its state and parameters while informing the other cameras. At first, a camera assigns to itself all targets and observation areas viewed only by it after coordinating with other cameras, to guarantee that they will be monitored. After each camera updates their state, the assignment of the other tasks follows as described. A camera does not participate in the bidding process if it has no remaining available resources, no visibility of the task-related area/target, or no more remaining energy.

### 3.5 Adaptive Strategy Selection

To enhance the performance of the market-based event-driven multi-task bidding process we also propose an adaptive energy-oriented method for cameras to change their bidding strategy (Fig. 3). The approach is motivated by the fact that initially the cameras have full energy resources and thus can focus on satisfying the task demands, while only at the latter stages of their lifetime they can gradually give more importance to the energy factor in order to conserve energy. To do this we first introduce two energy thresholds used by all cameras ($\theta_1^T$ and $\theta_2^T$). While the energy levels of a camera remain above the first threshold the initial values for the weights ($W_e, W_r, W_f, W_d$) are used and the goal is to give more significance to the performance components of the utility function. Once, the energy drops below the first threshold the weights are adjusted so that the energy and resource factors in the utility function are given equal or higher importance than the performance factors in (3). Finally, once the energy drops below...
Algorithm 1: Task Allocation

At step \( t \)

For \( T \in T \) and Target entered/left FoV

If camera \( C_i \) can execute task \( T_i \) then

Multicast \( U(T_i,t) \) to cameras \( C_k \) \( \forall k \neq j \) that also see the area/target

Receive \( U_k(T_i,t) \) \( \forall k \neq f \) of cameras that also see the area/target

If \( U(T_i,t) > U_k(T_i,t) \forall k \neq f \) then

Camera \( C_i \) takes over Task \( T_i \)

Update \( R_j, E_j(t) \)

end

\[ \Theta_i^F < E_j(t) \leq \Theta_i^T \]

\[ \Theta_i^F < E_j(t) \leq \Theta_i^E \]

\[ E_j(t) \leq \Theta_i^E \]

\[ E_j(t) = \Theta_i^E \]

\[ E_j(t) = \Theta_i^T \]

\[ E_j(t) = \Theta_i^F \]

Figure 3. Algorithm 2: Camera strategy selection based on energy thresholds

the second threshold the impact of the last two terms in (3) is further reduced. This dynamic adaptation allows cameras to have different task assignment strategies depending on their operating environment, battery levels, and workload. Hence, it offers a balanced way of changing the objective of each camera, depending on its current state, in order to extend the overall network lifetime. Such techniques are also useful to prolong the operation of the network until an energy resource becomes available. Once this happens, a camera can then resume normal operation by reverting to a previous state, depending on its energy state, where it will favor performance metrics over resource metrics.

4. EVALUATION AND RESULTS

To evaluate our proposed distributed energy-aware task allocation approach we have developed a visual simulation environment in MATLAB encapsulating all the aforementioned models and algorithms. Within a 2D field, \( N_c \) smart cameras with different battery levels, consumption rates, resources, and FoVs, are generated either randomly, or under a specific pattern in an attempt to maximize camera overlap in a \( 200 \times 200m^2 \) area. In turn, targets with a random initial position that follow a random but structured path are generated. In addition, observation areas corresponding to static tasks that have a random start and end time and also have different requirements regarding resources and frame-rate are also generated within the field at a position where one or more cameras could monitor it. Consequently there are a total of \( N_v \) vision tasks that need to be performed within the network. In total 20 different camera network setups (Fig. 4) were generated based on predefined structures and also random camera placement with the configurations shown in Table I. The number of tasks also varied between scenarios, with an increasing number of tasks generated as the number of cameras increased. We compare the proposed method with the baseline approach (Row 1 in Table II) where tasks are allocated in a periodic fashion and only based on the resolution and performance factors of (3) and where the weights do not change over time. The different starting initial weight values used in the experiments are shown in Table II (Rows 2 − 3).

Various factors affect the performance of the energy-aware distributed task allocation method some of which are extrinsic such as the network structure, overlap between the FoV of the cameras, and the number and location of observation targets/areas, and some of which are intrinsic such as the weight values and energy threshold values. Networks with limited activity in the FoV of most cameras or limited overlap result in a small lifetime increase since the cameras have limited interaction and the opportunities to allocate tasks in different ways in order to conserve energy are also limited. However, in real-world environments the structure of CSNs is such that the camera FoVs will often overlap to cover an area from different angles and hence the energy gains are expected to be higher. In our experiments, we found that by weighing the energy and resource factor more at the initial stage resulted in reduced lifetime increase. We have also experimented with the different energy threshold values for the adaptive weight strategy outlined in Section 3.5. The best thresholds that maximized the lifetime while at the same time also consider the task requirements were between \( 40 \) − \( 50 \) for \( \theta_1^E \) and \( 10 \) − \( 20 \) for \( \theta_2^E \). Setting the thresholds higher was found to cause unnecessary task reassignments between nodes resulting in executing more tasks than was otherwise necessary. Setting lower thresholds resulted in shorter lifetime improvement since the energy saving strategy was not initiated in time to migrate tasks to other cameras. The thresholds were empirically defined from various runs; however, we will be exploring ways to automatically compute them and adjustment as part of our future work (i.e. Monte Carlo Simulation). The average lifetime of each network setup for the different task allocation approaches, after averaging for multiple runs with different task configurations such as target moving patterns and area locations, is shown in Fig. 5. For all setups, the average network lifetime is increased between 10% − 30%. In addition, Fig. 6 also illustrates the impact of each mechanism in the overall improvement which shows that the adaptive weight strategy is a significant factor in the overall lifetime increase. For network configurations where multiple cameras overlap with each other (e.g. Setsups 1 and 10) the improvements are larger, while for networks with a small number of cameras (e.g. Setup 6) which limits the choices the algorithm can make, there is a smaller improvement. The results in Fig.7 and Fig. 8 show the average

| Number of Cameras \( N_c \) | Number of vision tasks \( N_v \) |
|-----------------------------|-----------------------------|
| 2-16                        | 10-50                       |

| Vision Tasks Resource Requirements | Vision Tasks Performance Requirements |
|------------------------------------|---------------------------------------|
| 5-25 %                             | 6-30 FPS                               |

| Table 1. Parameter Values in Different Experiments |
|---------------------------------------------------|
| \( w_g \) | \( w_r \) | \( w_F \) | \( w_D \) |
|---------|---------|---------|---------|
| Baseline | 0       | 0       | 0.5     |
| Initial Set 1 | 0.15 | 0.15 | 0.35 | 0.35 |
| Initial Set 2 | 0.25 | 0.25 | 0.25 | 0.25 |

| Table 2. Different Initial Weight Values |
|------------------------------------------|
| \( w_g \) | \( w_r \) | \( w_F \) | \( w_D \) |
|---------|---------|---------|---------|
| Baseline | 0       | 0       | 0.5     |
| Initial Set 1 | 0.15 | 0.15 | 0.35 | 0.35 |
| Initial Set 2 | 0.25 | 0.25 | 0.25 | 0.25 |
The first 10 scenarios followed a predetermined structure while for the other 10 setups, camera placement and orientation was random. The different scenarios also varied with respect to the camera processing capabilities and FoV.

Figure 4. Examples of the different network camera setups used in simulations.

Figure 5. Average lifetime of network for different network setups and scenarios for proposed methodology and baseline model.

Figure 6. Percentage of average lifetime improvement of proposed approach over baseline. Breakdown of the impact of each component.

Figure 7. Number of dynamic tasks (moving targets) monitored on average throughout the lifetime of the network.

Figure 8. Number of static tasks (areas) monitored on average throughout the lifetime of the network.

Figure 9. Percentage of network-covered area that two or more cameras overlap.

Figure 10. Number of tasks (static and dynamic) that are present in the network.
that are monitored throughout the lifetime of the network. In all scenarios it is observed that the proposed algorithms do not negatively impact the performance of the network as the number of monitored targets/areas not only remains the same in most cases, but also when activity is present in the network, it increases as there are active cameras capable of capturing the events. In particular, the number of monitored targets has increased up to 30% for target monitoring and up to 50% for static area monitoring for some networks. We also present results, which show the overlapping factor of each setup in Fig. 9 and the number of targets/areas that are present in each network setup in Fig. 10. However, from the results there is no direct correlation between the overlapping factor and the lifetime improvement. Instead, what affects the lifetime improvement is the amount of activity in the network and where it happens in the network and which cameras observe it.

5. CONCLUSIONS
This paper presented preliminary results on distributed adaptive energy-centric multi-task-allocation in CSNs. The proposed methodology is generic and thus can be used for different applications besides target tracking and can also be used for monitoring of static areas. Furthermore, the dynamic strategy adaptation shows promising results and so it is worth exploring improvements in the future such as different threshold values and weights for each camera. As an immediate follow up we aim to further improve the adaptive strategy assignment by exploring how the different parameters can be optimally and dynamically adjusted during operation. We also plan to introduce uncertainties into the framework to consider cases where targets are occluded and not visible by the cameras and also perform experimental validation using a CSN based on IP cameras. Furthermore, we will also investigate an energy replenishment model to also incorporate into the different scenarios.

6. ACKNOWLEDGMENTS
This work was supported in part by the ERC Advanced Grant “Fault-Adaptive”, ERC grant agreement n° 291508.

7. REFERENCES
[1] Aghajan H. and Cavallaro A. 2009. Multi-Camera Networks: Principles and Applications. Academic Press.
[2] Bramberger M., Doblander A., Maier A., Rinner B., Schwabach H. 2003. Distributed embedded smart cameras for surveillance applications. Computer, 39, 2 (Feb. 2006), 68-75.
[3] Qureshi F. and Terzopoulos D. 2007. Camera Sensor Networks in Virtual Reality. First ACM/IEEE International Conference on Distributed Smart Cameras. (25-28 Sept. 2007). ICDSC 2007. 87-94.
[4] Rinner B. and Wolf B. W. 2008. An Introduction to Distributed Smart Cameras. Proceedings of the IEEE, 96, 10 (Oct. 2008), 1565-1575.
[5] Kamal AT., Farrell J.A., Roy-Chowdhury A.K. 2013. Information Weighted Consensus Filters and Their Application in Distributed Camera Networks. IEEE Transactions on Automatic Control. 58, 12 (Dec. 2013), 3112-3125.
[6] SanMiguel J.C., Micheloni C., Shoop K., Foresti G.L., Cavallaro A. 2014. Self-Reconfigurable Camera Sensor Networks. Computer, 47, 5 (May 2014), 67-73.
[7] Imran M., et al. 2014. Complexity Analysis of Vision Functions for Comparison of Wireless Smart Cameras. International Journal of Distributed Sensor Networks. 2014, 1 (January 2014). 15 pages.
[8] Dieber B., Micheloni C., Rinner B. 2011. Resource-Aware Coverage and Task Assignment in Camera Sensor Networks. IEEE Transactions on Circuits and Systems for Video Technology, 21, 10 (Oct. 2011). 1424-1437.
[9] Saipullah K.M., Anuar A., Ismail N.A., Soo, Y. 2012. OpenCV Based Real-Time Video Processing Using Android Smartphone. Int. J. Comput. Technol. Electron. Eng. 1, 3 (2012), 58-63.
[10] Cenedese A., Cerruti F., Fabbro M., Masiero C., Schenato L. 2010. Decentralized task assignment in camera networks. 49th IEEE Conference on Decision and Control. (15-17 Dec. 2010). CDC 2010., 126-131.
[11] Dieber B., Esterle L., Rinner B. 2012. Distributed resource-aware task assignment for complex monitoring scenarios in Camera Sensor Networks. Sixth International Conference on Distributed Smart Cameras. (30 Oct.-2 Nov. 2012). ICDSC 2012. 1-6.
[12] Rinner B., Dieber B., Esterle L., Lewis P.R., Yao X. 2012. Resource-aware configuration in Camera Sensor Networks. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. (16-21 June 2012). CVPRW 2012. 58-65.
[13] Ye C., Zheng Y., Velipasalar S., Gursoy M.C. 2013. Energy-aware and robust task (re)assignment in embedded Camera Sensor Networks. 10th IEEE International Conference on Advanced Video and Signal Based Surveillance. (27-30 Aug. 2013) AVSS 2013. 123-128.
[14] Casares M., Pinto A., Youlu Wang, Velipasalar S. 2009. Power consumption and performance analysis of object tracking and event detection with wireless embedded smart cameras. 3rd International Conference on Signal Processing and Communication Systems. (28-30 Sept. 2009). ICSPCS 2009. 1-8.
[15] Maggiani L., Salvadori C., Petracca M., Pagano P., Saletti R. Reconfigurable FPGA architecture for computer vision applications in Camera Sensor Networks International Conference on Distributed Smart Cameras.(29 Oct. - 1 Nov. 2013). ICDSC 2013. 1-6.
[16] Anastasi G., Conti M., Di Francesco M., Passarella A. 2009. Energy conservation in wireless sensor networks: A survey. Ad Hoc Netw. 7, 3 (May 2009), 537-568. DOI= http://dx.doi.org/10.1016/j.adhoc.2008.06.003
[17] Ttron R. and Vidal R. 2011. Distributed Computer Vision Algorithms. IEEE Signal Processing Magazine, 28, 3 (May 2011). 32-45.
[18] Rowe A G., Goode A., Goel D., Nourbakhsh I. 2007. CMUcam3: An Open Programmable Embedded Vision Sensor Technical Report Datasheet, Carnegie Mellon University.
[19] Saipullah K.M., Anuar A., Ismail N.A., Soo Y. 2012. Measuring power consumption for image processing on android smartphone. Am. J. Applied Sci. 9, 12 (2012) 2052-2057.
[20] Zimmerman, A. T., Lynch J. P., Ferrese F. T. 2013. Market-based resource allocation for distributed data processing in wireless sensor networks. ACM Trans. Embed. Comput. Syst. 12, 3 (April 2013), 28 pages. DOI= http://doi.acm.org/10.1145/2442116.2442134