# Distributed Task Allocation for Visual Sensor Networks: A Market-based Approach

Felix Pletzer
Pervasive Computing Group
Institute of Networked and Embedded Systems
Klagenfurt University
Email: felix.pletzer@uni-klu.ac.at

Bernhard Rinner
Pervasive Computing Group
Institute of Networked and Embedded Systems
Klagenfurt University
Email: bernhard.rinner@uni-klu.ac.at

***Abstract*—This work in progress presents a novel distributed task allocation method for visual sensor networks based on a computational market. Our proposed method automatically adapts the QoS levels of the individual tasks, depending on the resource requirements and the user-defined interest level of the service. Therefore, we define virtual commodity markets, where decentralized producer agents sell resource shares of nodes and communication links. Producer agents adapt their unit prices for resources depending on the demand for the individual resources. In this paper we discuss two pricing models: adjusted linear pricing and rate adaptive pricing. To allocate resources required for executing a task, a task allocation agent requests a number of offers from different producer agents. Task allocation agents use the received interest levels, which correspond to the virtual money, to lease resource shares for a specific time.**

## I. Introduction

Visual sensor networks [1] consist of a number of smart camera nodes and play an increasingly important role in various surveillance scenarios. In contrast to traditional surveillance systems that analyze the captured image data on centralized servers, smart cameras provide decentralized in-network analysis of the captured image data. Improved network scalability and reduced communication load are major advantages of smart camera networks. A typical smart camera node, such as described in [2], consists of an image sensor, an embedded processing unit and one or more communication interfaces. Each node provides a number of resources such as CPU time, memory and communication bandwidth. These resources are used for executing various image processing tasks such as person detection or video encoding.

Figure 1 illustrates the considered scenario. It shows a heterogeneous smart camera network [3] that consists of high performance camera nodes, low performance camera nodes, and data processing nodes—connected in a peer-to-peer fashion. The high performance camera nodes are equipped with an Intel Atom Z530 CPU (1.6 GHz, 1 GB RAM) connected to a 1280x1024 CCD sensor. The low performance cameras are based on an ARM Cortex-A8 (600 MHz 256 MB RAM) connected to a 800x600 CMOS sensor. Additionally, the network includes data processing nodes for in-network image analysis. However, these nodes have no on-board image sensor. All nodes are connected though 100 Mbit ethernet (high bandwidth data link) or IEEE-802.11 wireless network (low bandwidth data link).

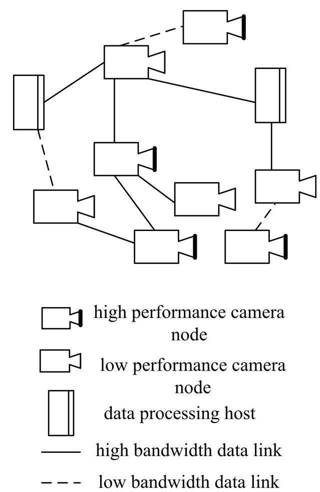


Fig. 1. Visual sensor network

This work deals with a decentralized, dynamic task allocation method for visual sensor networks based on a *computational market*. Each task processes data of a certain video source. Examples for tasks include person detection, motion analysis and video encoding. In general we assume the tasks to run in service mode, i.e., the execution times of the tasks are not known in advance. Furthermore, we consider the case of mobile tasks which can be migrated to a different node during run time. We further assume, that tasks are available at different quality levels (also called QoS levels). A QoS level is defined by a set of QoS parameters which influence the quality or accuracy of the result. Examples for QoS parameters include resolution, frame rate, or color depth. Moreover, QoS parameters, such as resolution of the captured image data, also have an impact on the resource requirements of the executed task.

Tasks can either be executed directly on the data source node (local execution) or on a different node in the network (remote execution). In case of remote execution, the captured raw data is streamed to the processing node, which requires additional communication resources. The goal of our allocation method is to dynamically distribute the tasks in a decentralized way using

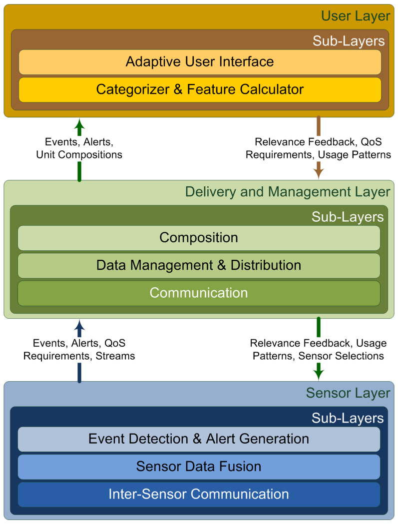


Fig. 2. Layer model of the Self-organizing Multimedia Architecture

a user-defined popularity measure, resource requirements of the individual QoS levels as well as resource constraints of the nodes.

This position paper presents a new market-based task allocation method for visual sensor networks. For modeling a computational economy, we adopt a multi-agent approach, where decentralized producer agents sell resource shares of nodes and communication links. Tasks act as consumers that purchase supplied resources. Therefore, producer agents use the demand for resources to update the underlying price model. Whenever a task is allocated, a task allocation agent requests a number of offers from different producer agents. Task allocation agents use the received interest levels, which correspond to the virtual money, to lease resource shares for a certain time.

The remaining paper is organized as follows: Section II briefly discusses related works, section III presents our proposed task allocation method and section IV discusses our approach and describes potential future work.

## II. Background and Related Work

This contribution is related to the *Self-organizing Multimedia Architecture(SOMA)* project [4]. The SOMA project aims to capture the whole life-cycle of multimedia content in a single architecture for large distributed multimedia systems. As illustrated in Figure 2, SOMA relies on a three-layer architecture consisting of a sensing layer, a distribution layer, and a presentation layer. Our work deals with the sensing layer of SOMA. The sensing layer implements a heterogeneous, multi-tier visual sensor network, that provides event detection and streaming services.

Market based resource allocation has been applied to different applications such as grid computing [5] or information service networks [6]. More recent work (e.g., [7]) deals with continuous auction protocols. Auction based methods provide a straightforward way to determine the market price of resources. However, concurrent bidding strategies are a major challenge of auction based methods. Although auction based methods are easy to implement, they usually perform worse than commodity markets [5]. Our work was also inspired by [8], that defines a virtual market for sensor networks. Sensor nodes perform reinforcement learning using received payments to adapt their operations.

## III. Market-based Resource Allocation Method

### A. Overview

The proposed method addresses the problem of automatic task allocation while balancing the network's resource utilization. Depending on the number of free resource units and the user-defined popularity, it adapts the QoS levels of the allocated tasks. Each task can be executed at a predefined number of QoS levels. QoS levels describe the provided quality in terms of QoS parameters such as image resolution or frame rate. Consequently, the choice of QoS level not only influences the quality of the output, but also impacts the amount of required resource units. In general, a task can be deployed on an arbitrary node and use an arbitrary data source, which delivers the input data for the service. However, executing a task on a different node than the data source node, requires allocation of additional network resources to deliver the sensed data, causing additional transportation costs.

Interest levels (called $I\$$) serve as user feedback, representing the user-defined popularity of a task. For our scenario we defined five interest levels ranging from 0 (no interest) to 4 (high interest). The idea of our market based solution is to use the user-defined interest levels as money equivalent, necessary for allocating resources. Tasks act as resource consumers with a specific resource demand. For a task running at a certain QoS level, we assume a constant resource demand, irrespective of the number of interested users. Therefore, it appears feasible to adapt the amount of allocated resources (and thus the provided QoS level) to the overall popularity of the task. Resource units can be leased at a certain unit price and have to be reallocated when the lease expires. In our first attempt, we included CPU load, memory and bandwidth resources to our model. Each task must provide a mapping for the resource requirements of their individual QoS levels.

For modeling the market based resource allocation method, we use a multi-agent approach. Producer agents act in charge of resource providers and sell free resource units to the task allocation agents. Every time, the amount of free resource units changes, the producer agent updates the unit price of the resource. Each producer agent is bound to a specific resource provider, such as camera node or communication link. Task allocation agents collect and accumulate the interest levels of specific tasks and decide about the deployment node. Therefore, a task allocation agent requests a number of allocation offers from the producer agent of the data source node and the local neighborhood. The task allocation agent choses the offer with the highest QoS level, which does not exceed its budget of accumulated interest levels. The behavior of the task allocation method highly depends on the chosen

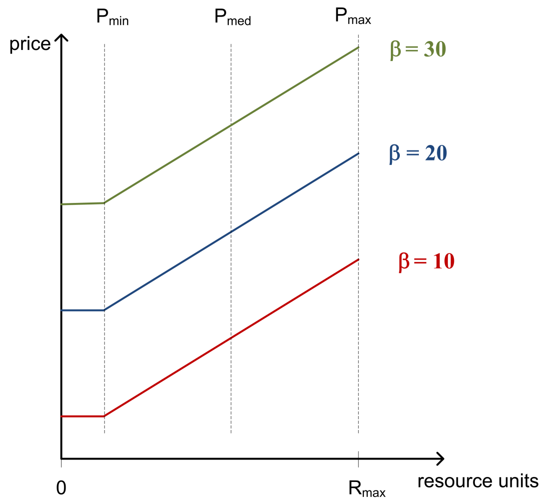


Fig. 3. Linear price lines for different volumes of $I\$$

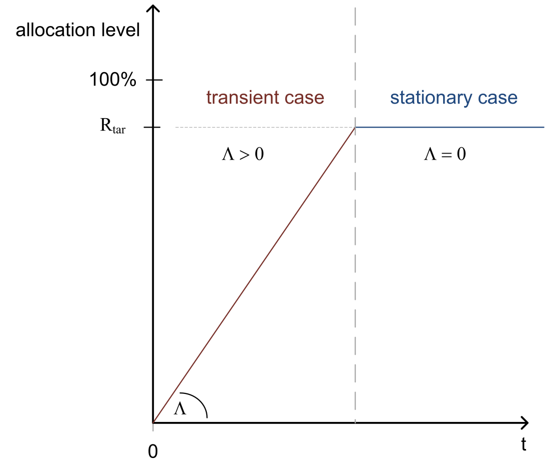


Fig. 4. Rate based pricing: Variation of target allocation rate $\Lambda$, depending on the resource allocation level

pricing model. We present two ideas for pricing models: adjusted linear pricing and rate adaptive pricing

## B. Adjusted Linear Pricing

This method was mainly inspired by yield management [9], frequently used for setting the price for airplane seats or hotel rooms, using demand responsive price discrimination. Instead of selling resource units at a constant price, the unit price increases proportional to the number of allocated units. Implicitly, the variation of the price also affects the QoS levels of the executed tasks. If the overall demand is high (scarce of resources), resources become less affordable for the executing tasks. However, since the volume of $I\$$ is not assumed to be constant in the system, a single linear price model is not sufficient. Instead we propose to adjust the linear price model to the average amount of $I\$$ in the neighborhood of the data source. Figure 3 shows an example for the linear price line for different $I\$$ volumes $\beta$ . For each $\beta$ the price for a resource unit ranges from a defined minimum price $P_{min}$ to a maximum price $P_{max}$.

Each time, the price of a resource unit is updated, the algorithm calculates $I_{avg}$ as the average $I\$$ volume of the neighborhood. Equation 1 describes the update of the $I_{avg}$ value. It updates $I_{avg}$ by averaging the volume of $I\$$ of tasks executed in the N-neighborhood $V$ of the data source (including the data source itself).

$$I_{avg} = \frac{\sum_{I\$ \in V} I\$}{N+1} \tag{1}$$

Using the current $I_{avg}$ value, the algorithm chooses the corresponding $\beta$ from the next higher price line and calculates $P_{med}$ and $P_{max}$ as shown by equation 2.

$$P_{med} = \frac{\beta}{R_{max}} \tag{2}$$

$$P_{max} = 2 \cdot P_{med} \tag{3}$$

As shown in equation 2 $P_{med}$ is calculated from the assumed $I\$$ volume $\beta$. We chose the average price $P_{med}$ in a way that the total price for all available resource units equals $\beta$. Hence, in the case of constant unit prices , the unit price $P_{med}$ would lead to resource partitions proportional to the amount of $I\$$, while fully utilizing the overall amount of resource units. Calculating $P_{max}$ by doubling $P_{med}$ preserves the expected overall revenue (calculated as integral over the price function) with respect to constant unit prices for a linear price line.

Using the obtained $P_{med}$, $P_{max}$, the number of allocated resource units $R_a$ and the number of requested resource units $R_r$ to calculate the updated unit price $P$ as follows:

$$\hat{P}(R_r) = P_{med} + 2 \cdot \frac{P_{max} - P_{med}}{R_{max}} \cdot (R_a + \frac{R_r - R_{max}}{2}) \tag{4}$$

$$P = max(\hat{P}, P_{min}) \tag{5}$$

Equation 4 describes the linear update of the unit price. It uses $P_{med}$ and $P_{max}$, which uniquely define the chosen price line. For $R_r$ resource units it determines the value at position $\frac{R_a+R_r}{2}$ on the price line. This value corresponds to the average unit price for the requested number resource units.

## C. Rate Adaptive Pricing

Our second approach for adjusting the price considers the allocation rate of the individual resources. Therefore we assume time synchronization of the camera nodes. Price updates only occur at discrete time intervals $t_i$. For each node, the method uses an exponential running average to estimate the allocation rates $\lambda$ of the individual resources. For updating the allocation rate, it uses $\hat{\lambda}_t$, which is the number of resources allocated in the previous time slot (negative value if more resources have been released than allocated).

$$\lambda_t = \alpha \cdot \hat{\lambda}_{t-1} + (1 - \alpha) \cdot \lambda_{t-1} \tag{6}$$

Equation 6 describes the update of the current allocation rate. It is only updated if at least one resource unit has been allocated or released in the previous time slot.

Using the current price $P$, allocation rate $\lambda_t$ and the target allocation rate $\Lambda$ and step size parameter $\gamma$ the price for a resource unit is updated as follows:

$$P_{t+1} = P_t + \gamma \cdot |\lambda_t - \Lambda| \quad (7)$$

To update the price, this method uses the absolute difference between the target allocation rate $\Lambda$ and the current estimate $\lambda_t$. The step size parameter $\gamma$ implicitly defines the relation between resource allocation rate and price and influences the adaptivity of the model. The model is first initialized with a first price estimate $P_0$ and gradually adapts to an equilibrium price.

For setting the target allocation rate $\Lambda$, we distinguish between two cases: the transient case and the stationary case (illustrated by figure 3). In transient case, the target allocation level $R_{tar}$ of the resource has not been reached and therefore $\Lambda$ must be greater than zero. In stationary case, when target allocation level of the resource has been reached, the target allocation rate is equal to zero, i.e., the amount of allocated resources equals the amount of released resources.

## IV. Discussion

The proposed task allocation method has several advantages. A major advantage evolves from the decentralized nature of the proposed method. In contrast to auction based methods, that usually rely on a central market place, producer agents only use local information to determine the price. Adjusting the price for resources depending on their supply/demand relation is a straightforward way for adapting the quality level of the executed tasks. Taking into account communication resources as well, our method also models costs of task migration. Since the task allocation agent always chooses the best offer, the method also supports substitution of resources. For instance, a task might be available with different CPU time/memory trade offs. Depending on the availability of resources (and user-defined interest levels), the method could find the best time/memory trade off for a specific task.

In general, however, determining the price is related to a prediction problem. Using the current demand, it tries to estimate the future demand level. Based on the estimated demand level, the method has to adapt the price accordingly. Adjusted linear pricing, for instance, adapts the price with respect to the number of allocated resources and the volume of interest levels in the neighborhood. Adapting the price according to the demand, improves balancing of the allocated tasks. Adjusting the money volume adapts the price line to improve the utilization of the resources.

Rate adaptive pricing is especially useful for highly dynamic scenarios where resources are frequently allocated and released. At the beginning, the target allocation rate $\Lambda$ is set to a specific value. Higher values of $\Lambda$ potentially lead to lower prices and therefore cause a higher resource utilization at startup. Moreover, our proposed task allocation method is well suited for heterogeneous environments. Different devices adapt their resource prices according to the device capabilities. For example, resource units of micro controllers usually have a higher price than resource units of a high-performance server.

First simulations indicate that the proposed market-based task allocation method provides a good solution for the discussed scenario. Resources are allocated in a balanced way, and the QoS level of the executed tasks is adapted according to the user-defined interest levels. We currently work on simulating a number of different scenarios with different interest level distributions to provide a detailed evaluation of the proposed method. We further plan to include additional parameters such as energy consumption to our model and investigate non-linear price functions. We finally integrate our task allocation method in a visual sensor network for autonomous event detection and deliver these events to multimedia distribution network. This network is used to deliver and present multimedia content of public events to a large set of users.

## Acknowledgment

This work was supported by the Lakeside Labs GmbH, Klagenfurt, Austria and funding from the European Regional Development Fund and the Carinthian Economic Promotion Fund (KWF) under grant 20214/17097/24774.

## References

[1] S. Soro and W. Heinzelman, "A Survey of Visual Sensor Networks," *Advances in Multimedia*, vol. 2009, p. 21 pages, 2009.

[2] W. Schriebl, T. Winkler, A. Starzacher, and B. Rinner, "A Pervasive Smart Camera Network Architecture applied for Multi-Camera Object Classification," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2009.

[3] B. Rinner, M. Quaritsch, W. Schriebl, T. Winkler, and W. Wolf, "The Evolution from Single to Pervasive Smart Cameras," in *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-08)*, Stanford, USA, 2008, p. 10.

[4] A. Sobe and L. Böszörmenyi, "Towards self-organizing multimedia delivery," *Reports of the Institute of Information Technology, Klagenfurt University, TR/ITEC/12/2.08, Tech. Rep*, 2008.

[5] R. Wolski, J. Plank, J. Brevik, and T. Bryan, "Analyzing market-based resource allocation strategies for the computational grid," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, p. 258, 2001.

[6] T. Mullen and M. Wellman, "A simple computational market for network information services," in *Proc. Int. Conference on Multi-Agent Systems*, 1995.

[7] B. Pourebrahimi, S. Ostadzadeh, and K. Bertels, "Resource Allocation in Market-based Grids Using a History-based Pricing Mechanism," *Advances in Computer and Information Sciences and Engineering*, pp. 97–100, 2008.

[8] G. Mainland, D. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, vol. 2, 2005.

[9] B. Smith, J. Leimkuhler, and R. Darrow, "Yield management at American airlines," *Interfaces*, vol. 22, no. 1, pp. 8–31, 1992.