# Collaborative knowledge building by smart sensors

V M Bove Jr and J Mallett

*In this paper we explore decentralised approaches for gathering knowledge from sensing devices. We contrast these with centralised processes like data mining, which assume that sensors, devices, or even people contributing information to a pool, do not have a sense of the 'whole picture' or the goal of the data collection. Thus it is necessary for a centralised mining process to create value by sorting, co-ordinating, and distilling the raw information. We consider instead a situation in which the contributors are given a goal, and are given the ability to co-ordinate among themselves in such a way that each can maximise its contribution to the pool. We discuss advantages of this new approach such as scalability and communication efficiency, and explore how it may change the design of devices, communication infrastructures, and algorithms, using several projects from the Media Laboratory as illustrations.*

## 1. Introduction

Data mining [1—3] is the process of pattern or model discovery or verification in large collections of information, the latter typically having been collected by more than one observer over some range of time and space[1]. The roots and typical approaches of this field originate in an era when processing was much more expensive than is now the case, hence an emphasis on centralised computation and 'historical' analysis rather than understanding phenomena *in situ* and in real time; nevertheless the basic idea of answering questions that no one observer could manage in isolation not only continues to be of interest, but grows in importance as the number of things that can collect data grows.

In this paper, we consider situations in which information is being observed by a relatively dense (relative to the information-theoretical sampling requirements of the underlying phenomena) ensemble of devices that contain a substantial amount of local computational power. Rather than blindly routing raw, undifferentiated information to a central location which performs knowledge discovery, the devices themselves have a sense of the goal (or goals) of the overall process[2], and co-operate with one another in a collaborative knowledge building process. Our intent is to deploy knowledge-gathering and understanding devices in the world such that:

- the system answers questions no single device could answer,
- each contributor has a sense of its relationship with others and motivation to maximise its contribution (and a way of evaluating the value of its contribution),
- each additional contributor potentially adds detail/precision/speed to the answer.

the devices themselves have a sense of the goal of the overall process

This decentralised, locally intelligent 'ecosystem' of devices is intended to solve problems that are too finely grained or too local for efficient solution[3] by centralised or top-down architectures. The incremental, scalable nature of the system eliminates the need for large amounts of expensive, fixed infrastructure, and therefore enables getting useful answers with just a few units and a corresponding small cost of entry (though better answers might result from more units).

[1] Some authors treat the phrases 'data mining' and 'knowledge discovery in databases' (KDD) synonymously, while others regard the latter as the process of understanding of information uncovered by the former.

[2] Centralised processing is often applied in situations where there is no clear-cut goal at the time of collection, and non-localised patterns are discovered only through analysis of complete data sets. Even in these cases, collaboration among sensing devices can be valuable in that it can reduce redundancy or identify regions of possible interest for later analysis (thus making communication and later processing more efficient).

[3] Where efficiency might be measured in terms of, for example, time, hardware resources, power, or communications bandwidth.

## 2. Sensor networks

With the prices of networkable sensing devices dropping, the volume of information being collected by networked sensors has increased dramatically in recent years, and data mining is increasingly being applied to the resulting databases [4]. A common architectural approach to these problems has been a more or less direct replication of traditional data mining systems, in which the sensor nodes collect and route information to a central site, where database analysis is performed.

It is desirable when designing networked sensing systems to develop architectures that have certain general characteristics:

- power efficiency,
- communication bandwidth efficiency,
- minimal fixed infrastructure,
- incremental scalablility from a few units to a very large number,
- no potential single point of failure.

Clearly application specifics and budget constraints will adjust the relative importance of each of these as well as adding additional concerns.

When sensing devices contain more computation than needed just for data acquisition and communication, they can act as data analysers, not just sensors and routers. A strong motivation for doing the processing in the sensor network is that even if bandwidth were not an issue, long-range transmission of raw data would still cost several orders of magnitude more power than local processing of it [5] — bandwidth or communication-power concerns become particularly important for high-data-rate sensors such as video cameras. Ganesan et al [6] have argued recently that a centralised data-mining approach is inappropriate — particularly under communications bandwidth constraints — given that the sensor data is often spatially and temporally correlated such that transmission in its raw form unnecessarily consumes limited resources, and also given that many phenomena of interest are spatially and/or temporally local and could be uncovered efficiently by local analysis. The approach suggested by these authors involves distributed multiresolution wavelet analysis, and therefore is effective only for phenomena whose statistical properties are accessible to such analysis (mid- and high-level audio and video understanding problems, for instance, generally are not), but their optimisation strategies appear to be useful even beyond such situations. Work in distributed implementation of algorithms such as principal components analysis can be relevant here, too (see, for example, Kargupta et al [7]) though often the communication model assumed is that of a typical computing cluster rather than a smart sensor network.

Looking for local data correlations in a distributed sensor network in such a fashion is an example of collaborative signal and information processing (CSIP) [5], which is concerned with determining appropriate groups of smart sensors to co-operate on a particular information gathering task.

A variety of approaches have been proposed that keep the data at or near the sensors, to avoid overwhelming the network's communications bandwidth. In the IrisNet project [8], the outputs of sensors are regarded logically as a single XML database. Queries can be made using the standard XPATH query language, and the system routes queries to an appropriate level of a hierarchical data organisation. The TAG framework [9] supports SQL-like queries, and has been implemented in the TinyDB system on Berkeley motes; queries originate at an external 'base-station' and flood the network, whose sensors organise themselves into a routing tree which sends appropriate results back to the base-station.

> long-range transmission of raw data would cost orders of magnitude more power than local processing

In mobile-agent systems, the data likewise remains at the sensors, while queries and processing tasks are sent as software to the sensing and computing nodes. In order for this to make sense, the agent code (and the inter-agent traffic) must be significantly smaller than the raw sensor data. Qi et al [10] have described an application of mobile agents to multiresolution data integration problems. Mobile agents may be applicable to some of the problems we shall discuss in the next section of this paper, though how to employ them on more complex sorts of multiple-observer processing tasks still remains to be investigated in depth.

## 3. Smart cameras

An increasingly important category of networked sensors is the networked camera. Since a fair amount of hardware is needed just to control a camera's operation and conform to the requirements for membership on a network, it is a small step to the 'smart' camera, in which tasks of interest can be performed directly on the camera's processor [11—14]. While the basic ideas discussed in the preceding section apply to smart cameras as well as to other sorts of sensors, networks of cameras have several important characteristics that will affect a system architecture:

- the sensors are generally very directional (though there are exceptions, see Huang and Trivedi [12]),
- the sensors typically do not blanket an area as densely as simpler sensing devices would,
- as a result of the preceding two points, the sensors may have the ability to pan, tilt, zoom, or even relocate,
- the raw data is of potentially overwhelming size,
- the desired algorithms are complex, multi-dimensional, and nonlinear, and often consist of a significant amount of independent (i.e. one frame from one camera) processing followed by integrative processing.

Much of the initial interest in smart cameras involved surveillance, particularly for security, military applications, and monitoring of traffic or industrial processes; in each case the on-camera processing was intended to look for particular phenomena of interest, either increasing the productivity and accuracy of human observers of the resulting video or performing automated decision-making in the absence of human observers. Cameras with significant processing capabilities can also be used for applications such as scene modelling (inferring 3-D structure by integrating multiple spatial or temporal viewpoints), user interface (e.g. gesture interpretation or gaze tracking), or interpersonal communications (e.g. tracking people in a meeting room or segmenting them from known backgrounds for intelligent conferencing systems); these latter three categories of tasks are all of particular interest to the Object-Based Media Group at the MIT Media Laboratory.

> interest in smart cameras involved surveillance and monitoring of traffic or industrial processes

In a smart camera system, the local processing performed on a camera will typically result in a significant reduction of the amount of data transmitted among collaborating cameras, likewise in the amount of data transmitted out of the system. Examples of how this might happen include:

- event-of-interest detection (transmit only if the event is happening),
- region-of-interest identification (transmit only a spatial region in which a sought-for feature is visible, e.g. a face or a vehicle),
- feature or metadata extraction (e.g. an edge map for stereopsis algorithms, or a feature vector associated with a detected face for person identification),
- independent data compression,
- compression conditioned upon knowledge of other sensors, e.g. if a neighbouring camera has already transmitted, use that view as a predictor and send only an efficient description of the differences (Bove and Butera [15] give an example of how to compute such descriptions, though not necessarily an algorithm directly applicable to the anticipated architecture),
- high-level model building, where the model is more compact than the set of individual 2-D views.

How should the cameras in such a system be organised for effective collaboration? In the example IrisNet application presented in Deshpande et al [8] (fixed cameras reporting open parking spaces), there was a clear geographically based hierarchy that made sense for the problem at hand. But such a fixed organisation might not generalise to all situations; if the cameras were collaborating to track a moving vehicle, at certain periods of time, the data sharing and collaboration would cross ‘zones’, and, if the cameras themselves were mobile (as in our Eye Society system — see section 4 below), the organisation would have to be more fluid. Also, if cameras can participate in more than one task at a time, the different tasks could require different logical organisations, and some cameras might also not be able to participate in a task even if they were ideally located for doing so, if they were already fully computationally engaged[4]. Marcenaro et al discuss a fixed hierarchical organisation of intelligent cameras, hubs and control rooms, but also acknowledge the limitations of such systems and point to the need for more flexible architectures [13]. Collins et al describe an architecture in which tasks are dispatched from a central location [14]; however, their strategy for co-ordination and hand-off of dynamic tasks, such as moving-object tracking, would apply to a decentralised architecture as well.

Remagnino et al describe how high-level agents can be employed on smart cameras for analysis tasks [16], though not how appropriate groups of cameras can organise themselves; our research is looking to solve this problem through a group co-ordination protocol for autonomous smart cameras.

## 4. Research projects

A current research project in our group seeks to enable smart cameras and other smart sensing devices to organise themselves into groups in response to the needs of a particular task and the knowledge and availability of a particular device, rather than according to a fixed hierarchy based on hardware or geography.

Distributing tasks among multiple nodes has been extensively researched, both from the perspective of using some form of centralised manager or managers responsible for breaking the task up to subordinate nodes, and from the emergent results of co-ordination arising from sophisticated local decision making [17, 18]. Bidding schemes for deciding task allocation among participating nodes have also been studied, and were proposed as an early solution to sensor network problems by Smith [19] as part of the contract net framework.

In this sort of computational context, a group is typically defined as a set of processes that can communicate with one another using a one-to-many protocol. We have chosen also to allow direct one-to-one communication where the processes determine that it would be appropriate. The philosophy of the group-forming protocol we have developed is to provide a simple method for potential group members to establish their suitability for a particular task group, without extensive negotiations and most importantly without a need for any predefined hierarchical control structure such as task managers. Members may use the protocol to set up such structures if they wish, but it is not imposed upon them at the start.

Among autonomous elements, group creation becomes the problem of providing all the information required for individuals to decide whether or not they can usefully contribute to the group. For many tasks it is reasonable to assume that this information will necessarily be incomplete, since the desirability of group membership may be partially dependent on other individuals’ decisions to participate. These criteria make group creation a more dynamic and continuous process than typically seen under other

[4] IrisNet can perform dynamic load balancing for queries under a single task; here, we are considering multitasking.

assumptions. Consider as a simple example the problem of getting the *n* nearest cameras to some location to perform some task such as panning so that they are all looking at the same point. If one of the cameras is performing another task with which this requirement conflicts (like observing an object in another direction), it may choose not to participate in the group, thereby extending potential group membership to the $n + 1$th closest camera and beyond.

The group protocol includes in the group creation announcement information on the task to be performed, and a set of task-specific joining requirements. Example requirements might be a distance from a fixed location, having observed a phenomenon of interest, or perhaps proximity to another camera that has observed an object of interest (consider hand-off of a tracking task as a moving object passes from the view zone of one camera and towards that of another, or performing triangulation to determine 3-D position). An individual device can then make a decision on whether to join the group based on whether it meets the criteria and whether it is available to help (which might mean not performing a conflicting task, or still having memory or computation to spare).

The basic operations the protocol makes available to individuals include:

- creating a group and advertising its membership criteria,
- joining or leaving a group,
- sending and receiving messages within a group,
- modifying the membership criteria,
- re-evaluating whether to remain in the group (whenever another enters or leaves, or the criteria are modified, the group members are notified of the change and may need to re-evaluate their own membership),
- terminating a group.

These operations are available to an application through a relatively simple set of interface functions. A device may be a member of any number of groups simultaneously.

In principle, forming groups based on clear-cut criteria should be relatively straightforward; in practice, distributed and dynamic groups can exhibit unexpected behaviours, especially in the presence of large communication latencies or outright failures. Difficulties for group protocols include preventing circumstances where multiple separate groups are created where a single one was intended, avoiding oscillations or race conditions in group membership, and ensuring that groups shut down cleanly without leaving stray members (particularly troublesome if they attempt to relaunch the group when it is no longer needed). To test and verify group control and behaviour, we are developing a verification and simulation tool which can display the state of running groups, complete with task and membership information, and a log of group traffic. It is also capable of simulating group membership, and injecting group messages in order to verify group behaviour especially in the case of recovery from exceptions.

As an example of an application of the group-forming protocol, consider the case of a distributed face detection problem. A set of cameras looking into a room needs to determine the number of unique faces. Because of viewpoint differences, occlusions, and the like, none can see the whole room so this process requires integrating the observations of all cameras seeing a face. For simplicity, assume that the application is already installed in the cameras, and that they have already established some sense of where they are relative to other cameras; then the process can proceed essentially as follows:

- each camera runs a face detector independently, which for each face found returns a bounding box and feature vector,
- a camera seeing a face, and not already in a group matching it, creates a group with a membership criterion of seeing a face with a feature vector similar to its own — estimates of regions of the room viewable from a given camera can be added to the membership requirements to prune out impossible matches,
- a camera finding itself in two groups with similar criteria (meaning they are both discussing the same face) merges the groups into one.

The result will be one group per unique face. Within the groups, the member cameras can use messaging to determine which has the best view of each face (where ‘best’ might be defined as largest bounding box) and transmit that view out of the system to a user. Other current research includes the use of the protocol to allow mobile smart cameras to perform group calibration, a task whose multiple levels include identifying cameras with overlapping view zones, determining commonly visible scene features within the overlap regions, using the latter as input data for the actual calibration calculations, and possibly taking advantage of egomotion to verify or refine the solution obtained.

> distributed groups can exhibit unexpected behaviours in the presence of large communication latencies

We have built two hardware platforms for exploring such computational strategies for smart cameras; in one system the cameras are mobile, while the other combines smart cameras with other sensors and output devices.

The first platform on which we have implemented the group forming protocol is Eye Society (Fig 1) [20], a system of small wireless autonomous mobile cameras that operate as a group to solve machine vision tasks. Given sufficient on-board computing to do real-time scene analysis without external processing resources, these cameras enable us to investigate how scene understanding can be improved when each camera is independently capable of analysing its own sensor data, and

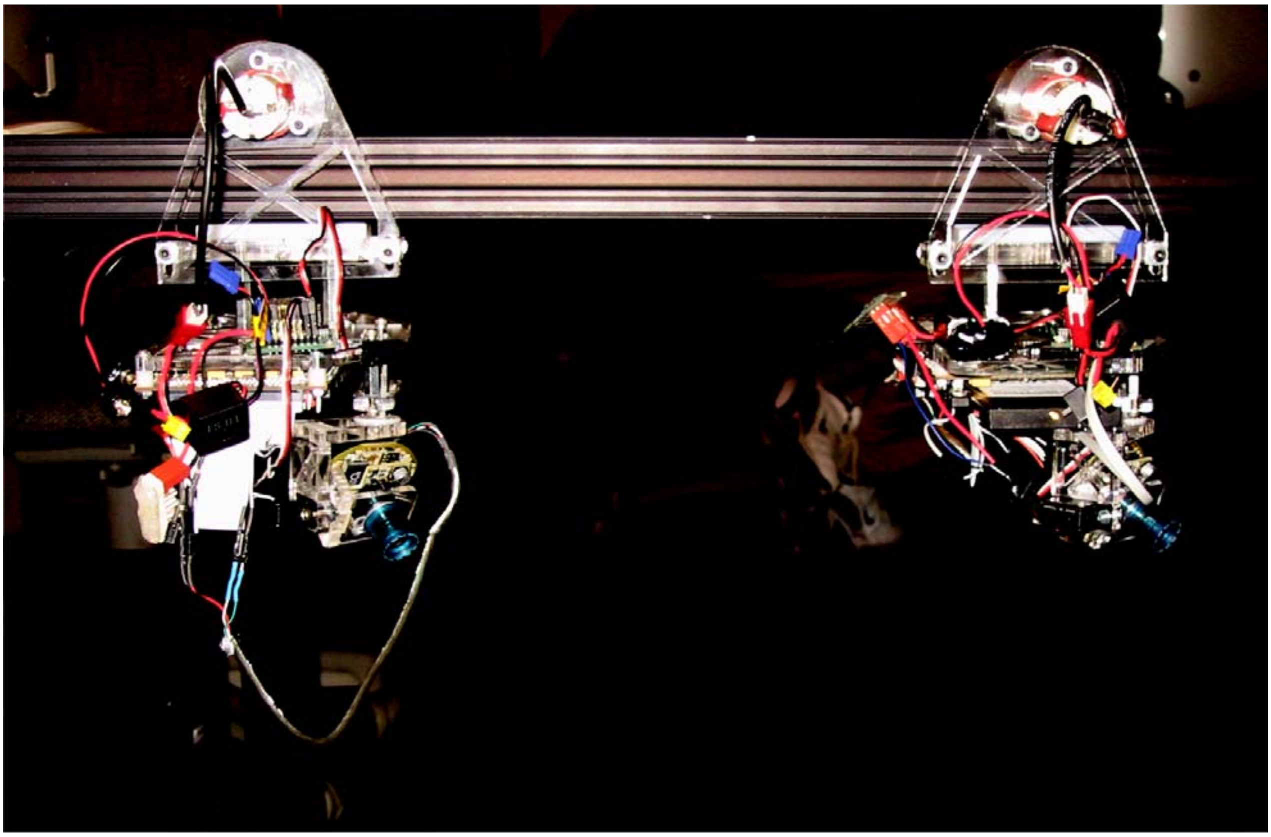

Fig 1 Two Eye Society cameras on an overhead track.

sharing information on what it sees with its fellow devices. The units travel on an overhead track, and the cameras are equipped with pan and tilt servos. Each camera in the project has IEEE802.11b wireless communication to other cameras and access to 'offshore' resources such as databases and file space as required. The current version of the controller centres on a commercial processor board containing a StrongARM SA-1110 processor, 32 MB of flash memory, an SA-1111 companion chip (which allows interfacing to USB devices), and additional analogue and digital inputs and outputs used to control locomotion and camera servos, and to accept input from sensors such as microphones. An add-on board of our design contains motor drivers.

Initial research investigates collaborative solutions to such problems as calibration and confirmation of observations. The ability of the cameras to move allows them to seek locations that minimise undesirable effects such as occlusion and specular reflections, maximise visibility of useful features such as edges and three-point perspective, verify hypotheses by such means as egomotion or stereopsis, and (particularly relevant to the topic of this paper) optimise their positions relative to those of other cameras.

Longer-term work includes increasing the processor power of the cameras to allow more sophisticated real-time processing, recasting traditional machine-vision algorithms for a distributed environment, incorporating additional sensors such as microphones, and enabling other sorts of devices (both stationary sensing devices of the sort described in the following discussion and unconstrained-motion floor robots) to become part of the acquisition-and-understanding ecosystem.

Another platform that shares software infrastructure with the preceding is the Smart Architectural Surfaces (SAS) project (Fig 2), a collaboration between the MIT Media Laboratory andb the Information and Communications University in Seoul, Korea, that examines the inclusion of sensory input/output and computation into building materials such that 'smart rooms' and other intelligent input/output spaces can be built from modular elements. SAS is a framework forexploring how intelligent devices that are part of the architectural fabric of a building can work together to understand and respond to people's activities, and to provide rich connections to people in other similarly equipped spaces. The communications and computational approaches are related to those of Eye Society, though with non-mobile devices that include more sensing modalities, and information output as well as input. Key assumptions include reconfigurability, incremental upgrade to a very large scale, and minimal centralised infrastructure.

Two conference room walls built from these units are shown in Fig 2. The basic unit is a tile, which snaps into a grid which provides support and power. Each tile contains similar hardware (though with a more up-to-date PXA255 processor) and communication to the Eye Society cameras, along with input devices such as a camera, microphone, ultrasonic

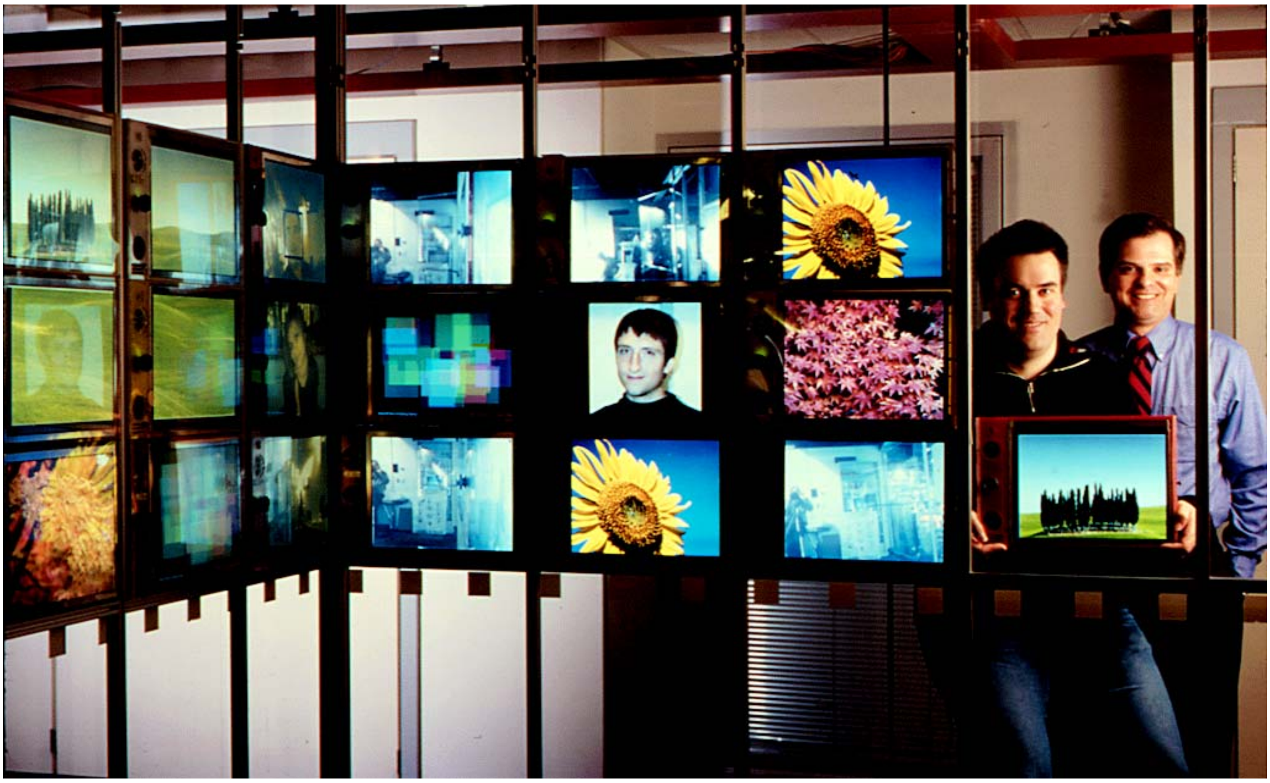

Fig 2 A set of Smart Architectural Surfaces tiles, each containing camera, display, speaker, microphone, and other sensors. [Photo credit: Webb Chappell]

proximity sensor, and a variety of other sensors. Output devices include a speaker and a display.

Collaborative knowledge-building applications for the SAS tiles that we are investigating include:

- the use of multiple cameras to infer position, activity, and gaze direction of one or more users for interface purposes,
- collaborative processing of microphone inputs for source localisation, source unmixing, and signal-to-noise ratio improvement,
- group camera calibration,
- correlation among multiple sensing modalities (e.g. associating objects sensed by the cameras with observations by the microphones or proximity sensors).

The group forming protocol is also appropriate for output applications such as finding a group of speakers or displays in a particular spatial relationship, and for allowing other devices such as personal digital assistants to join the group of tiles in order to provide additional input or output for the system.

## 5. Conclusions

Applying the same Viral Communications principles (such as incremental scalability and minimisation of centralised infrastructure) to the realm of networked sensing devices — in this particular case, cameras — yields the requirement for decentralised and self-organising architectures, the protocols to support them, and the redesign of processing algorithms for such an environment. The resulting collaborative approach not only offers advantages in scaling, cost-of-entry, and robustness, but also may make more efficient use of power and bandwidth.

## Acknowledgements

The authors thank our many colleagues and collaborators, in particular Arnaud Pilpre, Seongju Chang, and Applied Data Systems. We also thank the anonymous reviewers for their thoughtful suggestions and pointers to relevant literature. The work described in this paper has been supported by the Digital Life and Things that Think consortia at the MIT Media Laboratory, and by a research contract from ICU.

## References

1 Frawley W J, Piatetsky-Shapiro G and Matheus C: 'Knowledge discovery in databases: an overview', in Piatetsky-Shapiro G and Frawley W J (Eds): 'Knowledge Discovery In Databases', AAAI Press/MIT Press, Cambridge, MA, pp 1—30 (1991).

2 Agrawal R, Imielinski T and Swami A: 'Database mining: a performance perspective', IEEE Transactions on Knowledge and Data Engineering, 5, No 6, pp 914—925 (December 1993).

3 Mannila, H: 'Data mining: machine learning, statistics, and databases', Proc Eighth International Conference on Scientific and Statistical Database Systems, pp 2—9 (1996).

4 Chong C-Y and Kumar S P: 'Sensor networks: evolution, opportunities, and challenges', Proceedings of the IEEE, 91, No 8, pp 1247—1256 (August 2003).

5 Zhao F, Liu J, Liu J, Guibas L and Reich J: ‘Collaborative signal and information processing: an information-directed approach’, Proceedings of the IEEE, 91, No 8, pp 1199—1209 (August 2003).

6 Ganesan D, Estrin D and Heidemann J: ‘DIMENSIONS: Why do we need a new data handling architecture for sensor networks?’, ACM SIGCOMM Computer Communications Review, 33, No 1, pp 143—148 (January 2003).

7 Kargupta H, Huang W, Sivakumar K and Johnson E: ‘Distributed clustering using collective principal component analysis’, Knowledge and Information Systems Journal, 3, No 4, pp 422—448 (2000)

8 Deshpande A, Nath S, Gibbons P B and Seshan S: ‘Cache-and-query for wide area sensor databases’, Proc ACM SIGMOD2003, pp 503—514 (June 2003).

9 Hellerstein J M, Hong W, Madden S and Stanek K: ‘Beyond average: towards sophisticated sensing with queries’, Proc 2nd International Workshop on Information Processing in Sensor Networks (IPSN’03), pp 63—79 (April 2003).

10 Qi H, Iyengar S S, and Chakrabarty K: ‘Multi-resolution data integration using mobile agents in distributed sensor networks’, IEEE Trans Syst, Man, Cyber, 31, pp 383—391 (August 2001).

11 Wolf W, Ozer B and Lv T: ‘Smart cameras as embedded systems’, IEEE Computer, 35, No 9, pp 48—53 (September 2002).

12 Huang K S and Trivedi M M: ‘Distributed video arrays for tracking, human identification, and activity analysis’, Proc IEEE ICME2003, pp II-9—II-12 (2003).

13 Marcenaro L, Oberti F, Foresti G L and Regazzoni C S: ‘Distributed architectures and logical-task decomposition in multimedia surveillance systems’, Proceedings of the IEEE, 89, No 10, pp 1419—1440 (October 2001).

14 Collins R T, Lipton A J, Fujiyoshi H and Kanade T: ‘Algorithms for cooperative multisensor surveillance’, Proceedings of the IEEE, 89, No 10, pp 1456—1477 (October 2001).

15 Bove Jr V M and Butera W: ‘The coding ecology: image coding via competition among experts’, IEEE Trans Circuits and Systems for Video Technology, 10, pp 1049—1058 (October 2000).

16 Remagnino P, Orwell J, Greenhill D, Jones G A and Marchesotti L: ‘An agent society for scene interpretation’, in: ‘Multimedia Video-based Surveillance Systems: Requirements Issues and Solutions’, Kluwer, pp 108—117 (2000).

17 Durfee E H: ‘Partial global planning: a coordination framework for distributed hypothesis formation’, IEEE Trans on Systems, Man and Cyber, 21, No 5, pp 1167—1183 (September/October 1991).

18 Watlington J A and Bove Jr V M: ‘A system for parallel media processing’, Parallel Computing, 23, No 12, pp 1793—1809 (December 1997).

19 Smith R G and Davis R: ‘Applications of the contract net framework: distributed sensing’, Proc Workshop on Distributed Sensor Nets, pp 12—20 (1978).

20 Mallett J and Bove Jr V M: ‘Eye Society’, Proc IEEE ICME2003, pp II-17—II-20 (2003).

V Michael Bove Jr, who heads the Media Lab's Object-Based Media group, is the author or co-author of over 50 journal and conference papers on digital television systems, video processing hardware/ software design, multimedia, scene modelling, and optics. He holds patents on inventions relating to video recording, hardcopy, and medical imaging, was a member of several professional and government committees, and is on the board of editors of the Journal of the Society of Motion Picture and Television Engineers. In 2002 he was named a Fellow of the International Society for Optical Engineering. He is a founder of and technical advisor to WatchPoint Media Inc. He holds a BS in electrical engineering, an MS in visual studies, and a PhD in media technology, all from MIT.

Jacky Mallett is a PhD candidate in the MIT Lab’s Object-Based Media group.

She has a BSc in computer science from Loughborough University in England, and an MS in media arts and sciences from the MIT.

Her research interests are distributed autonomous computing, in particular working with groups of robotic cameras.