View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DCU Online Research Access Service

Intelligent Middleware for Adaptive Sensing of Tennis Coaching Sessions

Richard Tynan[1] Anthony Schoofs[1] Conor Muldoon[1] G.M.P. O'Hare[1]
[1]Clarity: The Centre for SensorWeb Technologies
University College Dublin, Dublin 4, Ireland
{richard.tynan, anthony.schoofs, conor.muldoon, gregory.ohare}@ucd.ie

Ciaran O'Conaire[2] Philip Kelly[2] Noel E. O'Connor[2]
[2]Clarity: The Centre for SensorWeb Technologies
Dublin City University, Dublin 9, Ireland
oconaire@eeng.dcu.ie, kellyp@eeng.dcu.ie, noel.oconnor@dcu.ie

***Abstract*—In professional tennis training matches, the coach needs to be able to view play from the most appropriate angle in order to monitor players activities. In this paper, we present a system which can adapt the operation of a series of cameras in order to maintain optimal system performance based on a set of wireless sensors. This setup is used as a testbed for an agent based intelligent middleware that can correlate data from many different wired and wireless sensors and provide effective in-situ decision making. The proposed solution is flexible enough to allow the addition of new sensors and actuators. Within this setup we also provide details of a case study for the embedded control of cameras through the use of Ubisense data.**

I. Introduction

To be able to teach effectively, a tennis coach must have in-depth understanding of the sport from the fundamental skills to advanced tactics and strategy. In addition, he/she must be able to reveal performance and tactical issues through efficient post-training or post-game analysis and relay this information effectively to the athlete. Video analysis can be an extremely useful medium within the area of sports coaching. This technology provides a means for a coach to effectively identify areas within a players technique requiring improvement to maximise their technical advancement, and to minimise injury potential. In addition, the medium provides the facilities through which a coach can highlight tactical awareness components of a students game to increase their cognitive understanding of game playing [12].

In collaboration with Tennis Ireland [19], the national governing body for the sport of tennis in Ireland, the TennisSense system [20] has been developed at their coaching headquarters. This is a technology platform which aims to provide their coaches with technological solutions that allow them to more effectively develop the next generation of elite tennis athletes. The TennisSense system comprises a wide range of sensory modalities to aid in the performance analysis of the athlete. A closer look at the components of the TennisSense system will be presented in the next section.

Given the wide range of components and the possible tasks they can perform, an adaptive middleware is required to manage both wired and wireless sensors in order to deliver the best Quality of Service (QoS) to the coach. This paper will present a novel agent-based middleware that is used to integrate multiple data sources from wired and wireless sensor networks. In addition to the integration of data, in-situ decision making will also be demonstrated to ensure that overall system goals are achieved.

In the next section, some related work is presented, following this, the TennisSense system is explored in greater detail. Agent Factory Micro Edition is then detailed as this is the underlying technology on which our adaptive middleware is built. We then examine the architecture of the system from the middleware perspective followed by a case study of the middleware in action to control the operation of the cameras. Finally we end with our proposed future work and the conclusions drawn from this initial work.

II. Related Work

There is much work in sports video analysis focusing on the detection of important events in the video [17]. However, the majority of approaches use broadcast sports video and can exploit the editing style of the content, such as the close-up of a player after a soccer goal, for example. Player-tracking and ball-tracking are also common methods for extracting semantic knowledge and the majority of approaches employ background subtraction or frame differencing to obtain candidate blobs which are then classified, tracked or discarded [8], [5], [14]. The main difficulty in such approaches is the problem of occlusion, but this can be overcome by using multiple cameras [14] or using an overhead camera, as we do in this work. Candidate detections can be filtered if they do not conform to certain constraints, such as knowledge of the ball's colour [5]. Image processing of this magnitude is notoriously resource consuming making it an undesirable option particularly when a more accurate localisation system is available. Lee et al. [7] explore camera selection in a surveillance context and propose a fast sub-optimal algorithm for selecting a subset of cameras that give the best view of the detected faces in the scene. Since

their cameras are calibrated,they can make inferences as to the overlap in camera views. For our middleware, we wish to actuate changes in the operation of the cameras based on sensed data to accurately capture the athlete's performance. The camera feed that offers the best view, which will be included in the report for the coach and player, will depend on the runtime performance of the cameras in terms of their tracking accuracy.

III. Background

In order to be become a successful tennis player one needs to possess a highly proficient and varied skill-set that covers such diverse areas as physical fitness, agility, mental toughness, tactical awareness, and technical ability. It has been shown, that the amount of strength, speed, agility and flexibility conditioning a player is prepared to undertake has been linked to the standard they play at [16]. However, in younger players, technical stroke production appears to influence rankings more than physical ability [6], and as such their training should concentrate on effective and efficient stroke mechanics, improving technique and ball placement, with less emphasis on physical conditioning. Other work has emphasised that a tactical approach to training (emphasising the role of strategy, tactics and decision making) for younger players leads to better game performance [21].

A. TennisSense Sensor Components

The first set of sensory data is from nine time-synchronised IP cameras positioned around the tennis court as indicated in Figure 1. The cameras are strategically located around a tennis court to capture a tennis match from a variety of coach-defined angles.

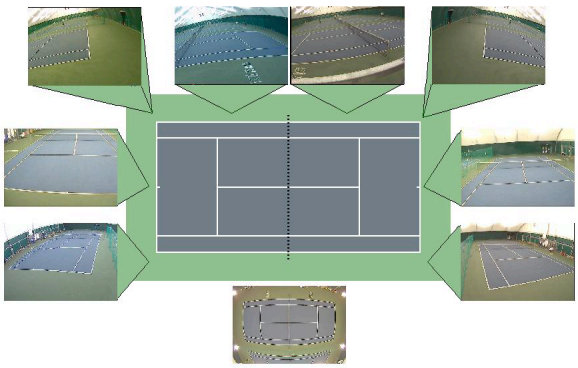

Fig. 1. Camera locations around the court.

Each camera allows the control of its pan, tilt and zoom (PTZ) functionality. During specification of the requirements of the system, the coaches requested the installation of an overhead camera with a wide-field of view. While this viewpoint provides a means to visualise tactical shots and movement during matches, it provides little information for the analysis of technical ability. Conversely, the other camera feeds can provide good viewpoints for technical assessment, but these viewpoints provide much less information about decision making and player strategy.

The second set of sensory data in the system is based on a Ubisense infrastructure [22]. The Ubisense system consists of a fixed set of receivers and mobile tags. The 3D position of the tags can be calculated by the fixed infrastructure along with the tag id and the timestamp of the reading. In this work, the player wears the tag which allows us to gain access to the players location even when they are out of view of the camera infrastructure.

B. TennisSense Actuator Components

As described previously, the cameras function as both sensors and actuators. In their sensory role they provide content, from which features of the match can be extracted. They also provide the ability to control their pan, tilt and zoom functionality. From the coaches perspective it is important that the cameras are pointed at the appropriate place at all times. With this in mind, our agent based middleware can control the PZT functionality of the cameras according to the Ubisense data it receives, and this will be described in the next section.

IV. System Architecture

Our proposed architecture is based on the wide range of scenarios that may need to be catered for within such a system, Figure 2. For instance simple logging of data from many distinct data sources may be required. Data markup may also be required and this may need to be performed live or in an offline fashion.

Other requirements on the middleware might be to provide some rudimentary image analysis for control of another component of the distributed application or feedback to a user of the system. The image analysis may also occur offline on logged data with the middleware not taking any decisions and merely acting as a data transport.

The middleware will also have to cater for the addition of new sensory modalities, for instance Foster-Miller vests [23] or interial sensors for fine grained observations of the individual limbs of the athelete. The system must also be able to control some operations of itself, as illustrated by the PTZ camera control in Figure 2.

This highlights an important role of the agents within the system - namely negotiation to achieve multiple goals. To see this, consider the cameras as both sensors and actuators. Based on the image analysis, the cameras may need to be retasked a certain way and a conflicting configuration may be suggested by the Ubisense and physiological data. It will be up to the agents at run time to decide how to optimise the configuration to satisfy as many goals as possible. Managing this process centrally may be impossible due to data volumes and bandwidth constraints.

A. Advantages of Using an Agent-based Middleware

The use of an agent based middleware such as the one proposed here enhances interoperability between systems. Since the Agent Communication Language (ACL) used by

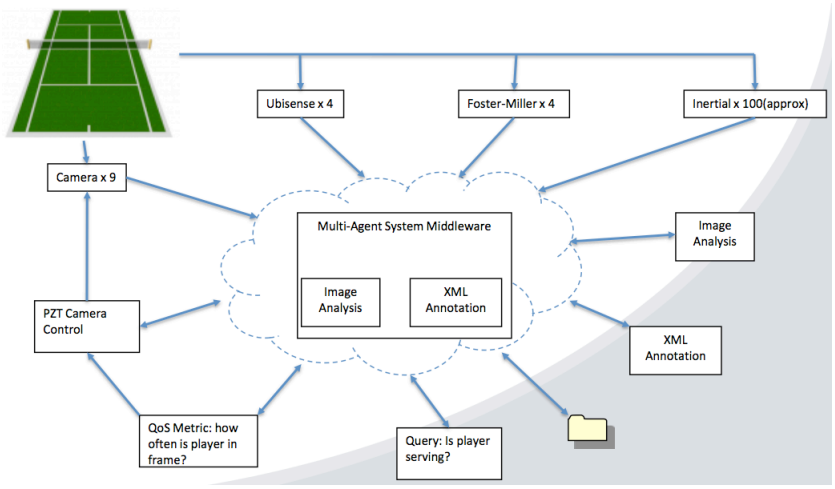

Fig. 2. Middleware architecture for the TennisSense system.

our agents is FIPA compliant, the agents can interact with other agents which also conform to this international standard. Another advantage is that it facilitates data management within the network. For example, data can be fused and/or correlated in the network thus limiting transmissions and overall data volumes. Such correlation is a vital component as displaying the raw data is not an option due to information overload on the coach and/or player.

Our approach also promotes flexibility in that new components can be added with ease and in some cases the system may be able to automatically discover how to use the new component. The agents can advertise services they offer and existing agents can make use of their services. Such a mechanism is referred to as Yellow Pages discovery [11] and can considerably reduce the burden of deploying additional components into the overall distributed application. Another related advantage is the reuse of components in subsequent applications and as with the addition of new components it may be possible for the new system to automatically discover how to reuse the component based on the appropriate advertisement of the services provided by the agent.

One serious disadvantage with systems of this type is the storage and treatment of data. Often, when a large amount of data is generated and in a short space of time, the purpose of the data is forgotten. With our middleware approach, the semantics of the sensory data is encoded with the data and therefore can be stored along with the data itself. While this adds to the overall volume, it will enable future harvesting of the data along with its semantics. Whether this is performed in the middleware or externally will be a performance related choice and may be decided dynamically by the agents in the system.

The use of a middleware such as this also enables a consistent and homogenous programming model to be used across multiple heterogenous sensors and actuators. This means that a common abstraction to data sources, controllers and actuators can be provided, which reduces the complexity of the application development. This is useful for the addition of components that cannot be automatically be incorporated into the application.

Many aspects of the operation of the system will need to be controlled at run time. The proposed solution enables the real time, in-situ decision making required of some applications that cannot have decisions made centrally due to latency issues for example. Finally, agent migration within the middleware can effectively balance load and ensure that a computation or deliberation is conducted on appropriate nodes within the distributed application.

V. AFME

In this paper we advocate the use of intelligent agents as the technology on which our middleware is built. An agent is an autonomous system. That is, a system that is capable of deciding at run time what is required to achieve its goals or objectives without human intervention. A multi agent system is a form of computationally reflective system that is concerned with the interaction, collaboration, and competition of multiple autonomous agents. Though endowed with particular responsibilities, each individual agent interacts with other agents in the framework to fulfil the objectives of the system. In artificial intelligence, the notion of an agent merges the concepts of behavioural systems with logical approaches. Agents are situated in the environment, but they are autonomous and do not simply react to it. Computational reflection is one of the most important areas of artificial intelligence; it is a technique that enables a system to maintain meta information about itself and use this information to change its behaviour or, in other words, to adapt. In the agent community parlance, this meta information is commonly referred to as an agent's belief set or an agent's model or the world.

Intelligent agents symbolically model their environment and manipulate these symbols in order to act. To model intentional agency, an abstraction level is chosen for the symbols such that they represent mental attitudes. Agent oriented programming is a paradigm for directly programming the behaviour of agents

using languages whose semantics capture a theory of rational agency [18]. Agents often have a set of beliefs, desires, and intentions and an interpreter that determines how the agents should achieve their goals within the environment [15].

Central to the coordination and collaboration of multiple agents is the existence of an Agent Communication Language (ACL) that is shared and understood by all agents[1]. The necessity to support inter-agent communication has led to the development of an international ACL standard, which has been ratified by the Foundation for Intelligent Physical Agents (FIPA). More recently, FIPA has been subsumed into the IEEE computer society and forms an autonomous standards committee with the objective of facilitating interoperability between agents and other non-agent technologies.

In this paper, we are concerned with Agent Factory Micro Edition (AFME) [10], [11], [9], which is a minimized footprint intelligent agent platform for resource constrained devices. AFME is based on Agent Factory [3], [13], a pre-existing agent platform for desktop environments. In AFME, as with many other intelligent agent platforms, system functionality is delivered through a combination of imperative and declarative code.

In AFME, the imperative functionality is in the form of a set of perceptors and actuators[2]. The declarative functionality is in the form of commitment rules, which define the conditions under which agents should adopt commitments to perform primitive actions or plans. In short, perceptors generate meta information (beliefs) about the system state along with information related to the environment, potentially coming from hardware sensors or video cameras. Using this information, the agent will decide, using its internal declarative rule set, on the actions that need to be performed. Actuators provide the functionality for primitive or atomic actions. In AFME, the functionality for the perceptors and actuators is implemented within Java (the CLDC J2ME configuration). It should be noted that when an agent adopts a plan or a primitive action fails, a commitment management process is invoked. A discussion of commitments and the commitment management process is beyond the scope of this paper[3]. The following is an example of a simple AFME rule:

```
a(?var),b(?var)>doSomething(?var);
```

In the above example, if the agent adopts the beliefs *a(?var)* and *b(?var)*, the action *doSomething(?var)* will be adopted. That is, the belief sentence represents the predicates or conditions under which actions are performed[4]. In AFME, the ? symbol represents a variable. In AFME, the truth of the belief sentence is evaluated using resolution based reasoning, which is the goal based querying mechanism used within Prolog.

AFME is documented extensively elsewhere. The remainder of this section will briefly discuss some new enhancements/changes in the latest release of the platform. In the current version of the system, it is possible to encode rudimentary mathematical predicates (equalities and inequalities) into the agent's declarative rules. In certain circumstances, this improves the efficiency of the platform by reducing the number of control algorithm cycles required to evaluate a set of conditions within a commitment rule. Previously such functionality would have to be encoded imperatively within Java. The message transport and migration services have been restructured to enable the development of consistent code for Sun SPOTs (WSN motes) and the CDC peer to peer services (Gateway and PDA type devices). These services enable agents to communicate and move between devices respectively. Support classes have been created to aid the direct construction of agents within Java. These classes are useful for integrating agents with pre-existing applications and for quick testing solutions.

VI. Case Study: Adaptive Camera Control

In order to realise our intelligent agent-based middleware, we have selected an exemplar scenario which demonstrates many of the advantages of our approach while, more importantly, providing functionality that is crucial in assisting a coach with the evaluation of the athletes performance. This demonstrator differs from existing work such as [4] in that rather than the Ubisense data merely being used to select the optimal camera feed to provide for the coaching staff, we use the data to dynamically retask the operation of the sensors to achieve optimal results.

A. Process outline

This section gives a general overview of the adaptive camera control process we applied to the TennisSense infrastructure. In essence it consists of having the camera follow the tennis player in real-time, by setting correct pan, tilt and zoom camera parameters according to the player's location received from the Ubisense system. The agent based approach fits such a process by providing a goal-oriented programming model, with optimal duration the player is in shot as the goal to achieve.

Fig. 3. Ubisense tag centered in the camera's field of view. The movement of the tag worn by the athlete is detected by the Ubisense infrastructure and it's position calculated.

[1]Game Theory approaches can be used to enable agents to coordinate actions without communication through the assumption of mutual rationality, but in the majority of frameworks agents do communicate.

[2]It should be noted that the word perceptor is used rather than sensor to distinguish the software component from the hardware. Additionally, there are perceptors, such as perceptors that monitor system state, that do not have a hardware analogue.

[3]The interested reader is directed to [2].

[4]For clarity, we have simplified the description of the process here. A more accurate description would be the belief sentence specifies the conditions under which a commitment management process commences.

The system comprises the Ubisense tag and sensors to locate the player Figure 3, the monitoring station to process the camera settings based on the Ubisense input, and the networked video camera that is actuated by the monitoring station. The tennis player is given a Ubisense tag that will be responsible for emitting Ultra Wide Band (UWB) pulses. Ubisense sensors placed in the tennis court infrastructure will sense the pulses and determine the player's location by correlating the various angles of arrival on each sensor. Ubisense's UWB radio systems can deliver up to about 15cm location accuracy indoors [22].

Data containing identity, timestamp and (x, y, z) coordinates of the player is subsequently sent to the monitoring station, where the agent based middleware is implemented. Our software agent implements a perceptor that creates beliefs from this live data feed e.g. BEL(playerLocation(x, y, z)). A Belief represents the informational state that an agent has at a certain time about the surrounding environment as discussed in the previous section. In our case study, the informational state is the player's location. Based on these beliefs, the software agent will take actions that will be triggered according to the agent's desires. For example, we consider in this case study the control of PTZ camera values to ensure the player is in shot for the optimal amount of time.

The control of the camera is realized via an agent actuator. The actuator bases its actions on the integrated beliefs. The main action is to send the right PTZ configuration parameters to the networked camera. AXIS 212 PTZ cameras [1] allow full overview and instant zoom with no moving parts. Such a camera provides a large-size picture in perfect quality and instantly enlarges an area when a pan/tilt/zoom command is issued. This is depicted in Figure 4. (1) represents the image displayed by the camera with the current settings, corresponding to a set of PTZ values that configured the camera to focus on the player at the previous location. Figure 4. (2) corresponds to the picture we are aiming at, so that we have the player (i.e. the Ubisense tag) in spot. Other cameras within the TennisSense system physically move according to the set parameters but we have not considered these in our initial scenario.

The agent actuator then needs to first translate the Ubisense coordinates of the player into the camera's frame of reference in order to calculate the right PTZ value settings. Then in a second step the actuator initiates a command to the networked camera to have it focus on the player's location.

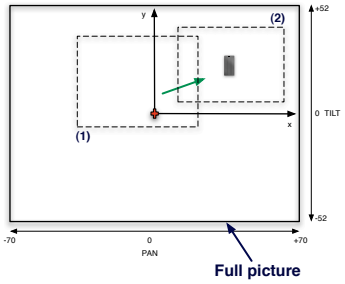

Fig. 4. Networked camera image control with no moving parts.

B. Implementation

We implemented the aforementioned process using AFME, presented in Section 4. Agent perceptors and actuators are Java objects. The perceptor *run* method connects to the Ubisense server, grabs live readings and creates beliefs in the agent's mental state. The actuator's first duty is to retrieve the location information from the beliefs, and translate the Ubisense coordinates into the camera's frame of reference. The camera is fixed to the infrastructure and, as shown in Figure 4, provides a full overview picture centered on a cross. The cross's location is static and is the origin of the frame of reference, to which any pan, tilt and zoom coordinates will refer to. Assuming that the cross' location is (0,0,0), we can calculate the (x0,y0,z0) location of the camera as shown in Figure 5.

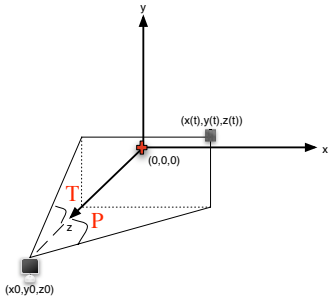

Fig. 5. PTZ calculation in the camera's frame of reference.

With respect to the Ubisense tag's location, we initially calibrate the system by placing the Ubisense tag at the cross' location. This allows us to translate the coordinates given by the Ubisense system in the camera's frame of reference. Further Ubisense coordinates are translated via this initial calibration. Therefore at any time *t*, we are able to have the (x,y,z) coordinates of the player. Calculating the pan, tilt and zoom to correctly set the camera to the new position is achieved by calculating the horizontal and vertical angles obtained from projecting the (x,y,z) coordinates on the horizontal and vertical 2D plans. This is shown in Figure 5 with *T* (Tilt) and *P* (Pan) indicating the absolute angle values that need to be sent to the camera. Zoom control is either constant or calculated from the absolute distance between the camera and the player.

AXIS networked cameras have an HTTP-based application programming interface. This allows for setting and retrieving camera parameters. In our case study, this permits a remote control of the camera from our Java agent Actuator. For example, the following http command:

http://cameraIP/axis-cgi/com/ptz.cgi?pan=20&tilt=-10&zoom=900

will request the camera to focus on the specific portion of the overview image defined by the pan, tilt and zoom passed in the http command. We are then able to keep track of the player in real-time. In our future work, we aim to push our intelligent middleware as close to the leaf nodes of the sensor and actuator network as possible. With this in mind we aim to

host our AFME agents on the cameras themselves, to provide in situ decision making as closer to the physical actuator as possible.

VII. Future Work

In the future, we aim to utilise our intelligent middleware to improve the quality of the system to the tennis coaching in a number of ways. First and foremost, we are in the process of conducting an experimental evaluation of our work to see if we can improve the coverage of the athlete as they move around the court. We also wish to include multiple cameras in the middleware and explore further improvements that can be achieved through the negotiation of agents resident of the nodes themselves, to decide which of them should be tasked to optimse the coverage of the player.

As part of the TennisSense system, physiological data will also be incorporated. By bringing this data into the middleware via our agents, the interporability and flexibility of our proposed approach will be tested. Future decisions about camera positioning may be taken based in respiratory data instead of just location information. For instance if a player's forehand deteriorates when under pressure, the system may choose at runtime to concentrate on this aspect of the players game based on the data feeds.

Inertial sensing is a way to get fine grained performance data of individual limbs of the player and this will also be included in our future work. It is envisaged that these sensors would be able to host one of our intentional agents on the device, allowing our middleware to permeate the environment as much as possible. Finally we would like to include multiple tags per player, for instance to allow cameras to focus on the footwork or shoulder action of the player. Our agent base approach will be vital here as depending on the orientation of the player, the cameras may have to negotiate the hand-off of responsibility to one of its peers.

VIII. Conclusions

Tennis coaching is proving to be one application domain with promising potential for applied research in the areas of intelligent middleware for sensor networks. The combination of mobile, static and wearable sensors, including visual, location, physiological and inertial sensing poses many problems in terms of data management, system reaction time and flexibility.

In this paper, we detailed a system that can bring together many distinct sensing modalities and utilise the data in an intelligent manner to provide in-situ decision making. Such decisions are taken opportunistically by agents at run time in order to optimise system performance from a coaching perspective. This occurs without control from a central entity and allows many factors to be taken into consideration when deciding how and when to act.

The proposed system is by no means limited to sensing tennis athletes from a training perspective. Many sensor network applications will require the integration of many heterogenous platforms, intelligent data management and intelligent in situ decision making with appropriate actions being taken without the need for human intervention. In this regard, we aim to demonstrate in future application scenarios the utility of using our approach in terms of flexibility and improvement to quality of service for the end user.

IX. Acknowledgements

The authors gratefully acknowledge the support of Science Foundation Ireland (SFI) under Grant No. 07/CE/11147.

References

[1] Axis Website. www.axis.com.
[2] R. Collier. *A Framework for the Engineering of Agent-Oriented Applications*. PhD thesis, University College Dublin, 2001.
[3] R. Collier, G. O'Hare, T. Lowen, and C. Rooney. Beyond prototyping in the factory of the agents. In *CEEMAS*, 2003.
[4] D. Connaghan, S. Hughes, G. May, P. Kelly, C. O'Conaire, N. E. O'Connor, D. O'Gorman, A. F. Smeaton, and N. Moyna. A sensing platform for physiological and contextual feedback to tennis athletes. In *Body Sensor Networks*, 2009.
[5] J. K. Fan Ye, William J. Christmas. A tennis ball tracking algorithm for automatic annotation of tennis match. In *British Machine Vision Conference*, pages 619–628, 2005.
[6] M. Kovacs. Applied physiology of tennis performance. *British Journal of Sports Medicine*, 40:381–386, 2006.
[7] H. Lee, L. Tessens, M. Morbée, H. K. Aghajan, and W. Philips. Suboptimal camera selection in practical vision networks through shape approximation. In *ACIVS*, pages 266–277, 2008.
[8] H. Miyamori and S.-i. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *FG '00: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, page 320, Washington, DC, USA, 2000. IEEE Computer Society.
[9] C. Muldoon. *An Agent Framework for Ubiquitous Services*. PhD thesis, School of Computer Science and Informatics, Dublin, Ireland, 2007.
[10] C. Muldoon, G. O Hare, R. Collier, and M. O Grady. *Multi-Agent Programming: Languages, Platforms and Applications*, chapter Towards Pervasive Intelligence: Reflections on the Evolution of the Agent Factory Framework, pages 187–210. Springer-Verlag Publishers, 2009.
[11] C. Muldoon, G. O'Hare, R. Collier, and M. O'Grady. Agent Factory Micro Edition: A Framework for Ambient Applications. In *Intelligent Agents in Computing Systems*, volume 3993 of *Lecture Notes in Computer Science*, pages 727–734, Reading, UK, 28-31 May 2006. Springer.
[12] N. E. O'Connor, P. Kelly, C. O'Conaire, D. Connaghan, A. F. Smeaton, B. Caulfield, D. Diamond, and N. Moyna. Tennissense: A multi-modal sensing platform for sport. *EICIM News, Special Edition on the Sensor Web*, 1:1, 2009.
[13] G. O'Hare. Agent Factory: An Environment for the Fabrication of Distributed Artificial Systems. *Foundations of Distributed Artificial Intelligence, Sixth Generation Computer Series, Wiley Interscience Pubs*, 1:449–484, 1996.
[14] G. S. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *ICPR*, pages 4152–4156, 2000.
[15] A. S. Rao and M. P. Georgeff. Bdi agents: From theory to practice. In *First Intl. Conference on Multiagent Systems*, 1995.
[16] P. Roetert, S. Brown, P. Piorkowskil, and R. Woods. Fitness comparisons among three different levels of elite tennis players. *Journal of Strength and Conditioning Research*, 10(3):139–143, 1996.
[17] D. A. Sadlier and N. E. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Trans. Circuits Syst. Video Techn.*, 15(10):1225–1233, 2005.
[18] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.
[19] Tennis Ireland. http://www.tennisireland.ie/.
[20] TennisSense Website. http://www.cdvp.dcu.ie/tennisireland.
[21] A. Turner. A comparative analisis of two approaches for teaching tennis: Game based approach versus technique approach. In *2nd ITF Tennis science and technology Congress*, 2003.
[22] Ubisense. http://www.ubisense.net/.
[23] F. M. Website. http://www.foster-miller.com/.