This document is published in:

"Molina, J. M. et al. (eds.) (2011) *User-Centric Technologies and Applications*: *Proceedings of the CONTEXTS 2011 Workshop*, (Advances in Intelligent and Soft Computing, 94) Springer, pp. 1- 9. Doi: http://dx.doi.org/10.1007/978-3-642-19908-0_1"

Integrating Multicamera Surveillance Systems into Multiagent Location Systems

José Luis Guerrero, Antonio Berlanga and José Manuel Molina

Group of Applied Artificial Intelligence (GIAA), Computer Science Department
University Carlos III of Madrid
Colmenarejo, Spain
{joseluis.guerrero, antonio.berlanga, josemanuel.molina}@uc3m.es

Abstract. Users are increasingly demanding personalized services based on their context, being one of the key features of that context the user's position. There are a wide number of possible solutions to deal with the positioning issue, which, for different situations, may have different accuracy requirements. This paper presents this issue from the point of view of an existing multicamera surveillance system which requires to be integrated into a multiagent positioning system, including a tracking example with the presented architecture.

Keywords: Multicamera systems, PTZ cameras, multiagent systems, tracking

1 Introduction

A well known trend in nowadays computing is the requirement of users to obtain different ranges of services according to their position, preferences, past actions, etc, which is usually known as context aware computing. Pioneer works in this field in the early nineties [1], [2] introduced and used concepts like ubiquous or pervasive computing [3], dealing with the automatic availability of different computers in an invisible way to the user.

The definition of context can be divided into different categories, regarding the information processed. In [4] the three main categories considered were computing context (accessible services and communication issues), user context (user location and profile) and physical context (external conditions). In [5] an additional important category is included, the time context (current date and time). The definition included at the beginning of this introduction deals with the most sensible category for the final user, its own context.

Among the user context responsibilities, one of the key processes is to be able to successfully determine the user location. Location systems [6] have been designed for this task, both for indoor and outdoor locations. Probably the best known location system for outdoor positioning is GPS [7], whereas indoor location systems may range from radio frequency locating systems [8] to active bats approaches [9]. An important feature to compare the different available systems is the infrastructure requirements which the different locating systems require in order to perform their location procedures. In surveillance scenarios the most common infrastructure found is a set of cameras (currently commonly used for human guided surveillance) leading to the possibility of positioning using that already installed set of cameras in order to automate and improve the surveillance task [10].

Along with the positioning system, context aware systems may require a tracking procedure to determine the user's current position, according to the current information provided by the sensing device and the previous positions obtained. This need leads to the use of methods which can handle the inaccuracies of the positioning system and predict the user position according to certain mathematical models. Kalman filter [11] is one of the most extended techniques used for this task, even though (at least in its basic version) it is only suited for linear movement tracking.

As previously overviewed, different locating systems may provide the system with different information sources with different accuracies and different characteristics. When choosing a single one of those systems does not provide us the location quality required by our system, we may resort to a joint use of different systems, trying to keep their different benefits while minimizing their handicaps. This is performed by information fusion systems [12], which combine the information of these different systems in a variety of ways depending on the requirements of the final locating system. These systems frequently use some tracking function as an intermediate step in their fusion cycle.

The operation of the context aware system will require different services (which may or may not have dependencies among them) to be run at the same time, which, along with the ubiquous computing statements, requires to set up a distributed computing architecture. For this particular task, multi-agent systems [13] are particularly well designed, since they allow an easier automatic adaptation to the different environment situations which systems may develop. In [14], the benefits of developing multi agent systems as an information fusion system to guide Unmanned Aerial Vehicles (UAV) [15] are discussed, while in [16] a multi agent based system for location is presented.

The objective of this paper is to present the required architecture and functions to include multi-camera based surveillance system into a context-aware architecture. This architecture will deal with the control requirements for the different cameras, the access interfaces, both locally and remotely, and finally present an example regarding a tracking system for an object based on its color print.

The structure of the paper will be divided in the following sections: initially a system overview will be presented, detailing the components of the built system and detailing the architecture presented. The general architecture will be followed by a section containing the detailed proposal for the automated handling of the different cameras, leading to a final example showing the overall function of the presented system and the conclusions which the previous sections lead to.

2 System overview

The concrete system used may add or remove certain requirements regarding the inclusion of a surveillance based system into a multi agent context aware system. This section will detail the concrete components used in the built system, along with their introduced restrictions and the general architecture design.

The cameras used are a set of Pan-Tilt-Zoom (PTZ) cameras which are Sony's VISCA protocol [17] compliant, such as the one showed in figure 1. The concrete models used where not able to provide the system with digital captures of their images, so digitalizer cards where used for this process, in particular Matrox Morthis frame grabbers. This introduces a handicap in the system, since the image provided by those cards may be required both by processes being run in the camera control and by agents external to it, and these cards can provide the image only to a single process. To enable a good scalability of the system regarding the images handling, a provider agent for each of the cards is built, being the images obtained by this intermediate provider, and thus preventing the direct access to the card's library.

Figure 1. VISCA protocol compliant PTZ camera

The access to the camera's library was distributed into three different levels, regarding the processes performed at each level: the first level basically transcribes the function information to sets of bytes and sends it through the connected port. The second level provides an easy interface to access those low level functions, while the third is responsible of higher level control functions (such as zone limits and management).

The access to the controlling library may be performed on a local or remote way. To allow this, we have included a server wrapper which allows any process to send the required command and receive its response remotely, while, at the same time, the server itself operates as any other local process in its calls to the library. This implies the definition of a communication protocol between the server and the clients. The defined protocol is a simple alternative consisting in the command performed, the result (three digits) and the possibly required data (according to different commands). The different parts of this message are separated with the proper tokens to allow the parsing in the client. The message structure is overviewed in figure 2.

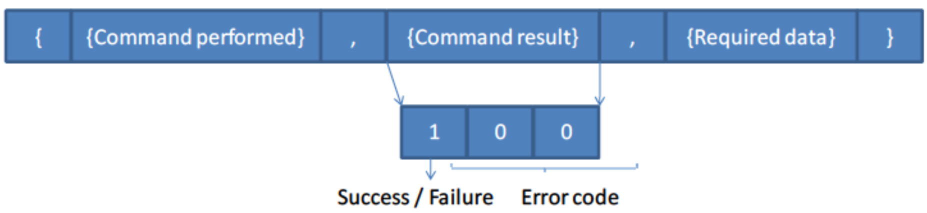

Figure 2. Remote communication protocol

An important feature is that the calls performed to the library are, by definition of the protocol, blocking, which means that no additional command may be performed by the camera until its previous one has been completed. The architecture presented is summarized in figure 3

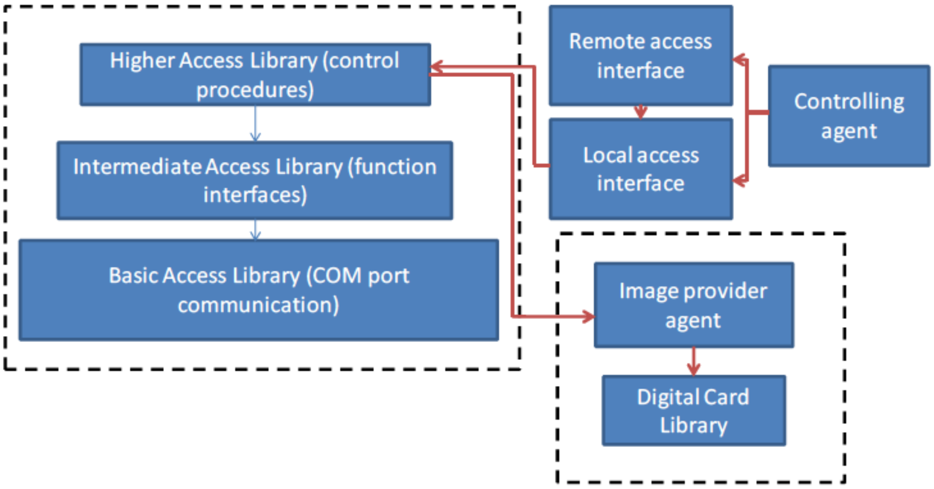

Figure 3. Architecture overview

3 PTZ Cameras Control Library

Building a library for the control of the different cameras implies both general and particular requirements, which at the same time may be closer to the actual functions which the camera may perform directly or require more intermediate computations apart from the actual device. To account for these different characteristics, the designed library has been divided into three different levels: basic, intermediate and higher access levels.

The basic level offers to the intermediate one the functions which the device can perform directly (such as moving to a specific position), converts them to its equivalent byte string, sends them over the port, receives the response from the device and converts it to a set of possible responses which are interpreted by the intermediate level library. The actual packets sent may have a length varying from three to sixteen bytes, with the structure shown in figure 4. The actual implementation of this basic library level has used the open source library libvisca[1].

The intermediate library level offers friendly interfaces to the low level functions, handling of the responses from the device and the required actions according to them and some basic device specific control. An example of this control may be to

[1] Available online at http://damien.douxchamps.net/libvisca/

determine the real movement bounds of the controlled device (something which cannot be performed by the basic library). If an order to move beyond the camera's boundaries in either of its axis is commanded, this intermediate level will detect it, move the camera as far as possible and response its caller with the proper information (indicating the wrong movement command and the action taken).

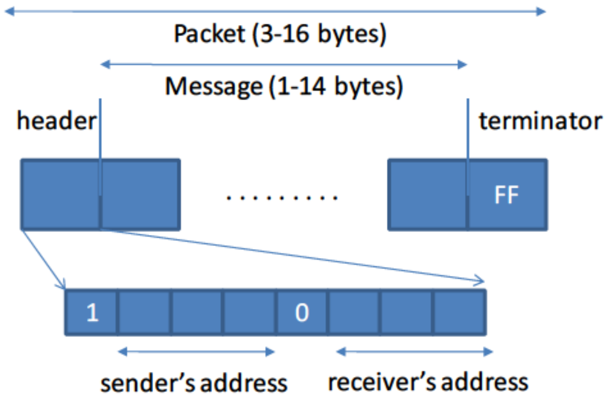

Figure 4. VISCA packet format

The higher access library introduces some not strictly device related actions into the library, allowing useful functions such as zone processing. Zone processing makes the system able to switch to previously established zones without requiring the caller to remember their coordinates. This also helps to develop surveillance routines moving through different zones of interest by a set of waypoints. These procedures may be very domain specific, according to the purpose of the system in which the library will be included, so they have been implemented at a different level to promote the reusability of the previous levels not including possibly unnecessary code. Another example of these higher routines may be the one performed by the agent in the example included in the following section.

4 A color based object tracker

Previous sections presented the architecture design and library details of the proposed system. In this section we will apply the explained to the task of following an object according to its color print. The color print of an object is the color components which the object has when it enters into our system. This approach can be useful in environments where the background of the image is prone to change (such as non-static cameras as the ones used in this paper). Depending on the application, it can be based on pre-established color prints or register the object color print and determine whether it is relevant or not once it enters the vision range of the given camera.

This function, even though here is presented as an agent procedure, can be considered as a higher level function of the camera library, since it can be useful in a multi agent surveillance system to coordinate the coverage of different cameras while one of them is performing the tracking of a certain object of interest.

The first level of this function performs the required image processing [18], obtaining the image to analyze and determining whether the interest color print is present in the image. The open computer vision library (OpenCV) [19] has been used to simplify this task. A color based filtering is performed, according to the histogram obtained from the color print, and then a linear filtering is performed according to the zone of interest's position, obtaining both the expected position of the object and the zone where the function musk look for it in the next iteration. Figure 5 shows an example of the color print filtering performed

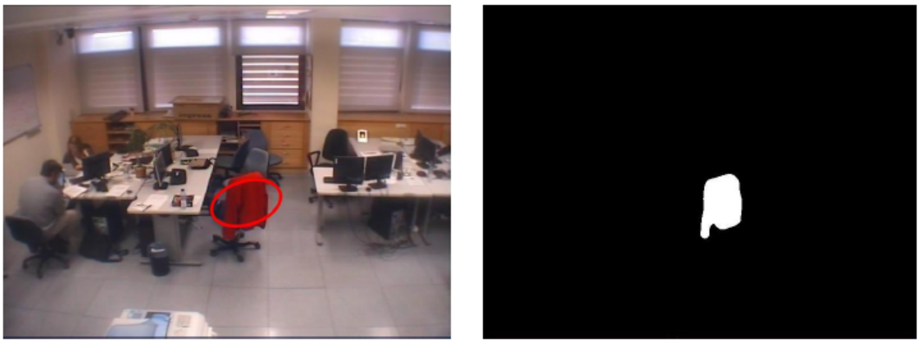

Figure 5. Example of the color print image filtering

The next action involves determining the amount of movement required by the camera in order to follow the object. Alternatives based on fuzzy logic [20] were tested, allowing a smooth control over the camera movement when the object was static, but they did not achieve good results tracking movement over a certain speed (probably due to the introduced delay which their calculations required). The final implemented solution performs a simpler and faster approach based only on the boundaries of the object, its center and the center of the image (moving the camera the required amount so that the boundaries of the object would include the center of the image).

The linear filtering is suitable for the task due to its low complexity, but it leads to what is usually called "*disengagement*": sudden movement changes make the filter predicted position wrong and thus it cannot find the object in its delimited zone of interest. To deal with this problem, a response based on two steps has been implemented: once the object of interest has not been found in the delimited zone, the whole image is searched for it, resetting the tracker state. If it cannot be found in the whole image, the camera resets its zoom status and goes back to its normal procedure (which might be to start a surveillance through different zones or stay in that position until another object matching the introduced color print is found).

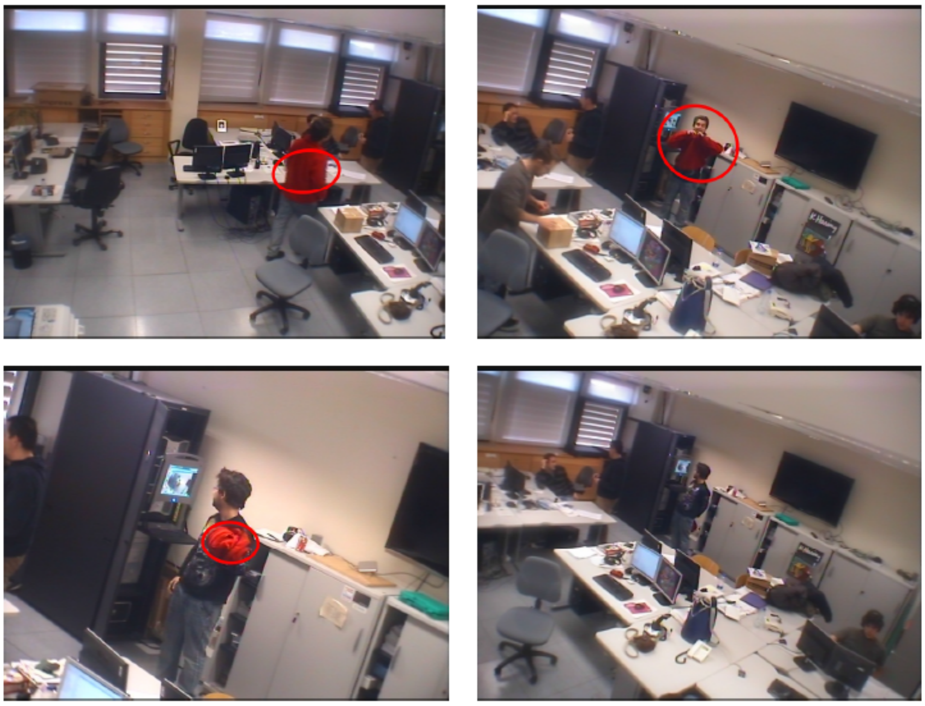

Figure 6. Example of a tracking situation. From the top image on the left, the object of interest starts to be tracker, on the following one it has been centered, followed by a zoom on it when its size is reduced and a final zoom out when it is lost

5 Conclusions

Reusing available resources is always a complex situation, especially when that reuse involves the inclusion of an automation process. Even so, this is the common trend for many currently human controlled systems. In this paper we have highlighted the versatility requirements for a surveillance system in order to integrate it into a multiagent positioning system, proposed an architecture based on three different library levels to cope with those requirements and shown an example of the results by means of a color print tracking system. Future lines cover the analysis of the whole positioning system performance, including quantitative measurements of the benefits of including the different information sources, in particular the ones provided by the integrated multicamera system.

Acknowledgments. This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Want, R., Hopper, A., Falcão, V. and Gibbons, J. "*The Active Badge location system*". ACM Transactions on Information Systems, 10(1):91-102, January 1992
2. Schilit, B., Theimer B. and Welch, B. "*Customizing mobile applications*". In Proceedings of USENIX Mobile & Location-Independent Computing Symposium, pp 129-138, USENIX Association. August 1993.
3. Weiser, M. "*The computer for the 21st century*". Scientific American, 94-104, September 1991.
4. Schilit, B., Adams, N. and Want, R. "*Context-aware computing applications*". In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 85-90, Santa Cruz, California, IEEE Computer Society Press. December 1994.
5. Chen, G. and Kotz, D. "*A Survey of Context-Aware Mobile Computing Research*" Dartmouth Computer Science Technical Report TR2000-381. 2000.
6. Hightower, J. and Borriello, G. "*Location systems for ubiquitous computing*" IEEE Computer Journal, 34 (8): 57-66. 2002.
7. Groves, P. D. "*Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*". Artech House, 2008
8. Bahl, P. and Padmanabhan, V.N. "*RADAR: An in-building RF-based user location and tracking system*". Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 2002
9. Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P. "*The Anatomy of a Context-Aware Application*" Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking, pp. 59-68. 1999
10. López de Ipiña, D., Mendonça, P. and Hopper, A. "*TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing*". Personal and Ubiquitous Computing, 6(3): 206-219. 2002.
11. Kalman, R.E. "*A new approach to linear filtering and prediction problems*". Transactions of the ASME- Journal of Basic Engineering, n. 82, pp 35-45. 1960.
12. Waltz, E. and Llinas, J. "*Multisensor data fusion*". Artech House Boston. 1990
13. Koski, A., Juhola, M., and Meriste, M. "*Syntactic Recognition of ECG Signals by Attributed Finite Automata*". Pattern Recognition, vol 28, no. 12, pp. 1927-1940, 1995
14. Guerrero, J.L., García, J. and Molina, J.M. "*Multi-agent Data Fusion Architecture Proposal for Obtaining an Integrated Navigated Solution on UAV's*". Proceedings of the tenth International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, pp 13-20. 2009.
15. Valavanis, K.P. "*Advances in Unmanned Aerial Vehicles. State of the Art and the Road to Autonomy*". International Series on Intelligent Systems, Control and Automation: Science and Engineering. Volume 33. Springer. 2007
16. Luis, A., Molina, J. M. and Patricio, M. A. "*Multi-Camera and Multi-Modal Sensor Fusion, an Architecture Overview*" International Symposium on Distributed Computing and Artificial Intelligence 2010, CONTEXTS Wokshop, September 2010.
17. Sony Corporation "*Sony EVI-D70/D70P Technical Manual*". 2003.
18. Fisher, R., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., and Trucco, E. "*Dictionary of Computer Vision and Image Processing*". John Willey. 2005.
19. Bradski, G. and Kaehler, A. "*Learning OpenCV: Computer Vision with the OpenCV Library*". O'Reilly Media. 2008
20. Klir, G.J. and Yuan, B. "*Fuzzy sets and fuzzy logic: theory and applications*". Prentice Hall. 1995.