

This item is likely protected under Title 17 of the U.S. Copyright Law. Unless on a Creative Commons license, for uses protected by Copyright Law, contact the copyright holder or the author.

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

DEMO

MultiCAMBA: A System to Assist in the Broadcasting of Sport Events^{*}

Roberto Yus¹, David Antón², Eduardo Mena¹, Sergio Ilarri¹, and Arantza Illarramendi²

¹ University of Zaragoza, María de Luna 1, Zaragoza, Spain,
{ryus, emena, silarri}@unizar.es,

² Basque Country University, Manuel de Lardizábal s/n, San Sebastián, Spain,
danton004@ikasle.ehu.es, a.illarramendi@ehu.es

Abstract. In this demo paper we present *MultiCAMBA* (*Multi-CAMera Broadcasting Assistant*), a context- and location-aware system that, using a 3D model updated continuously with real-time data retrieved from the scenario, helps technical directors (TDs) in the live broadcasting task. They can indicate in run-time their interest in certain moving objects or geographic areas, and the system is in charge of selecting the cameras that can provide the kind of view required. To achieve this task, the system continuously recreates the views of the cameras in a 3D scenario, considering possible occlusions among the objects.

Key words: Location-aware system, Multiple camera management

1 Introduction

Nowadays, the broadcasting industry is demanding new software systems that facilitate the broadcasting task. On the one hand, these systems should help increase the richness of live content production. On the other hand, they should contribute to decrease the production costs, mainly by reducing the need of highly specialized human resources to produce (create, edit, and distribute) audiovisual content. In that direction, we have developed the MultiCAMBA system, that enables TDs to quickly select, among many cameras (some of them can be static and others be attached to moving objects), the one that fulfills his/her requirements and whose view should be broadcasted.

The key features of MultiCAMBA, which supports the processing of location-dependent queries [3] about moving objects and cameras, are briefly highlighted as follows:

- It allows TDs to query about cameras viewing a target object/area and even obtaining a specific kind of view (front, top, rear and covering 50% of the target, etc.), through an *easy-to-use interface that recreates the real scenario using Google Earth technology*.

^{*} This work has been supported by the CICYT project TIN2010-21387-C02.

- *It ranks the cameras* obtained as a result according to different criteria: the time needed to view the target object/area, the distance to the target, the percentage of the object viewed, the percentage of the shot occupied by the target object, and/or the distance to the target.
- *It offers advanced functionalities.* For example, it estimates the time needed to view a target object by considering the cameras features. Besides, it analyzes the view of a camera taking into account occlusions among objects.

Other works also address the problem of multi-camera management (e.g., [1, 2]). However, they only consider static cameras (i.e., at fixed locations). On the contrary, the cameras considered in our proposal can be static or attached to moving objects. Moreover, we are not aware of any system that has applied an architecture for processing location-dependent queries in the context of live content production to retrieve a list of cameras providing views that fulfill certain requirements.

In the following we explain the two main components of MultiCAMBA: 1) the *query processor*, that executes the TD queries about interesting objects/areas to obtain the cameras fulfilling his/her requirements; and 2) the *Graphical User Interface*, that allows the TD to define these queries and visualize their results. Additional details can be obtained from the Web page of MultiCAMBA (<http://sid.cps.unizar.es/MultiCAMBA>).

2 Query Processor

The query processor is the component in charge of managing the queries submitted by the TD. These location-dependent queries are continuously evaluated (every second) in order to provide accurate and up-to-date results to him/her.

For example, the following interesting queries, among others, are supported by the query processor:

- Cameras that can view a certain object.
- Cameras that capture a side view of a certain object.
- Cameras that can view a certain percentage of an object in a defined area.

The query processor implements the functions defined in [4] and the 3D techniques in [6] to obtain the cameras that are fulfilling (or would be able to fulfill in the near future) the specified requirements of the TD. To compute a certain camera view (and for example to obtain the percentage of an object covered by it), a 3D engine (specifically, *JMonkeyEngine*) has been used. Due to space limitations, we refer the interested reader to the aforementioned papers for further information about how these tasks are performed.

Testing MultiCAMBA in a scenario involving real moving objects and cameras is very costly and it would be difficult to design repeatable experiments. So, we have developed a simulator that enables the accurate simulation of these scenarios by using stored *real* samples (obtained every second) of the location and direction of objects and cameras. So far, we have evaluated our system

in the context of the popular rowing races of San Sebastian, using real data corresponding to the competition of September 2010. Performance tests in this scenario show that the system can achieve an average refreshment period of 0.085 seconds (see [6]), so MultiCAMBA can be used for live broadcasting.

3 Graphical User Interface

The Graphical User Interface (GUI) has been developed as a Web page where the TD can define queries about interesting objects, visualize the results to his/her queries, define areas of interest, etc. It is clear and easy to use, as its appearance is similar to the usual workplace of a TD, where there are several input sources to select and a screen where the next source to be aired can be previewed.

The main components of the GUI (see Figure 1) are:

- The *query interface*, that facilitates the definition and submission of queries and shows the results in tables.
- The *overview map*, a Google Earth plugin that supports navigating through the scenario, defining interesting areas, and visualizing the results of the queries submitted.
- The *camera inputs*, two windows where the user can preview the video stream of the selected cameras before broadcasting them.

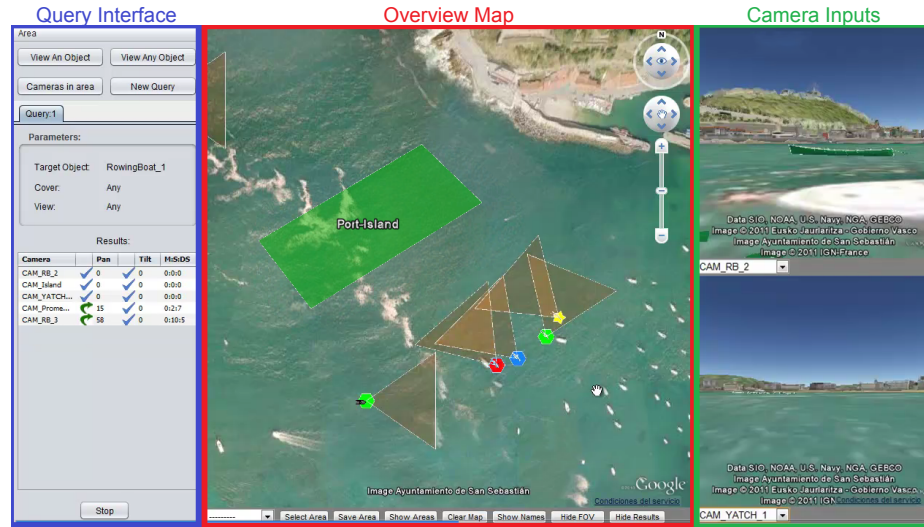


Fig. 1. Graphical User Interface using Google Earth plugins to simulate cameras

We use Google Earth to show to the TD a 3D up-to-date overview of the moving objects and cameras in the scenario and the results to his/her queries.

This is a user-friendly way of displaying the results and allows an easy interaction with the system. If no real video stream is available for the camera inputs, then the camera view is also recreated by using GE (as in [5]).

Different colors and shapes are used to clearly represent different important elements in the scenario: the target object (denoted by a yellow star), the cameras currently fulfilling the query constraints (green hexagons), the cameras that would be able to fulfill the requirements in a few seconds (blue hexagons), and the cameras that cannot fulfill the requirements (red hexagons). Figure 1 shows an example where the results of a query and the field-of-view (brown triangles) of the cameras are displayed on the overview map.

The GUI enables the TD to create his/her queries by selecting the target object or a specific camera (among the objects and cameras in the scenario), the area of interest where the target has to be located, the kind of view to obtain of the target (front, rear, side, etc.), the amount of the target that the camera has to cover (complete, any, or a certain percentage), etc. The TD can also define the ranking criterion to be applied to the answer set: 1) the time needed to obtain the required view (cameras that can provide the view right now could be more interesting); 2) the distance to the target (a shorter distance could lead to a more detailed shot); 3) the percentage of the target being viewed (TDs could want to view at least 80% or 100% of the target object); and/or 4) the amount of space that the target occupies in the shot (the more space occupied by the target the smaller the amount of space with other uninteresting objects). Once a query is defined, it can be submitted and/or stored to be loaded later.

4 Conclusions and Future Work

In summary, MultiCAMBA is a context- and location-aware system that gives a step forward in the development of software that allows to increase the efficiency in the broadcasting task by: 1) reducing the human effort required for live production (thanks to the friendly user interface provided) and 2) giving the opportunity to obtain richer content productions (due to the advanced features supported by the query processor).

As future work, we plan to consider more scenarios where the system can be helpful. For example, it can be used for other sport events such as cycling, where a distributed approach is needed due to the specific environment (long distances between objects, mountain landscape, etc.). Moreover, the features of MultiCAMBA could be also exploited in other non-sport scenarios (e.g., emergency management) involving moving objects and videocameras.

References

1. D. G. Aliaga, Y. Xu, and V. Popescu. Lag camera: A moving multi-camera array for scene-acquisition. *Journal of Virtual Reality and Broadcasting*, 3(10), December 2006.

2. W. Du, J.-B. Hayet, J. Piater, and J. Verly. Collaborative multi-camera tracking of athletes in team sports. In *Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE'06)*, May 2006.
3. S. Ilarri, E. Mena, and A. Illarramendi. Location-dependent query processing: Where we are and where we are heading. *ACM Computing Surveys*, ISSN 0360-0300, 42(3):12:1–12:73, March 2010.
4. S. Ilarri, E. Mena, A. Illarramendi, R. Yus, M. Laka, and G. Marcos. A friendly location-aware system to facilitate the work of technical directors when broadcasting sport events. *Mobile Information Systems*, 2012. To appear.
5. E. Michaelsen and K. Jaeger. A Google-Earth based test bed for structural image-based UAV navigation. In *12th International Conference on Information Fusion (FUSION'09)*, pages 340–346, July 2009.
6. R. Yus, E. Mena, J. Bernad, S. Ilarri, and A. Illarramendi. Location-aware system based on a dynamic 3D model to help in live broadcasting of sport events. In *19th ACM International Conference on Multimedia (MM'11)*. ACM Press, November 2011.