A Location-Aware System for Monitoring Sport Events

Sergio Ilarri
University of Zaragoza
Maria de Luna 1
Zaragoza, Spain
silarri@unizar.es

Eduardo Mena
University of Zaragoza
Maria de Luna 1
Zaragoza, Spain
emena@unizar.es

Arantza Illarramendi
Basque Country University
Manuel de Lardizabal s/n
San Sebastián, Spain
a.illarramendi@ehu.es

Gorka Marcos
Vicomtech Research Center
Paseo Mikeletegi 57
San Sebastián,Spain
gmarcos@vicomtech.org

ABSTRACT

Nowadays it is widely recognized the need of developing a new generation of professional equipment for broadcasting that allows a considerable reduction of productions costs in different scenarios. We consider mobile multimedia scenarios where the incorporation of last generation hardware and the development of new software should allow: 1) to decrease the needs of human resources to produce (create, edit, and distribute) audiovisual content; and 2) to increase the richness of the broadcasted product.

In this paper, we propose a software system that helps technical directors to deal with many motorized videocameras attached to moving or static objects in wireless environments. We use mobile agents to carry out the processing tasks wherever they are needed. Thus, agents are in charge of tracking the location of interesting static or moving objects and the (current and possible) focus range of all the cameras, as well as of refreshing the query answers efficiently and continuously. We present a traditional rowing race in the Basque Country as motivating scenario, where technical directors may indicate interesting objects or geographic areas in run-time and the system is in charge of selecting the best shots from the cameras in the scenario.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems – Query processing; H.2.8 [**Database Management**]: Database Applications; H.4.0 [**Information Systems Applications**]: General

Keywords

Mobile Multimedia Applications & Services, Enabling Infrastructures for Mobile Multimedia, Location-Dependent Query Processing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MoMM2010 8-10 November, 2010, Paris, France
Copyright 2010 ACM 978-1-4503-0440-5/10/11 ...$10.00.

1. INTRODUCTION

In the last years, different factors are provoking a deep revolution in the broadcast industry. First of all, there has been a strong decrease of the advertisement rates, mainly due to the economic crisis and the audience fragmentation caused by the digitalization and optimization of the spectrum (channel multiplication in satellite and terrestrial television) and the appearance of new communication platforms (e.g., video-blogs, Internet video platforms, mobile broadcasting, and so on). Secondly, continuous needs related with the upgrade of the technology (e.g., digitalization, development of media asset management systems, HD, 3D, etc.) are implying huge economic efforts. Finally, it is also evident that the expectations of the audience are increasing. The quality and richness of the content (e.g., amazing shots, last generation graphics, and so on) demanded by the audience and the advertisers is much bigger than in other distribution platforms such as Internet video platforms.

In such a context, organizations focus their efforts on the enrichment and the diversification of their offers but trying to reduce their production costs. On the one hand, the enrichment is tackled by the generation of very attractive and high-quality material, including in many cases technological support (e.g., the Obama hologram in the US presidential election in 2008). On the other hand, the reduction is considered by the acquisition of new professional products that, apart from begin able to deal with the last technology (e.g., IP interoperability, HD resolution, remote control mechanisms), either have a lower price due to the inclusion of hardware coming from the consumer electronic field or add new features that decrease the number of people that are required for content production. An example of this is the combination of professional cameras with low-cost remotely controlled cameras. This allows an enrichment of the content consuming experience without having a big impact on the total number of cameramen required. However, that solution has a serious impact in one of the most complex and critical tasks in an alive content production environment: the more cameras are employed, the more images are available, and therefore more complicated is for technical directors (people responsible for the content production of an event) to select the best video stream to broadcast.

The audiovisual production of rowing races in the Basque Country is a nice paradigm of this situation and it will serve us as sample scenario in this paper. The live broadcasting of such rowing races requires a very complex infrastructure: one helicopter, sailing boats with cameras and GPS transmitters, more cameras in the harbor, and a production headquarter (usually a trailer). In such a complex context, the technical director is responsible for the selection and coordination on the fly of the graphical material, video signals, and shots that are finally broadcasted to the TV audience. These tasks become especially difficult when different unexpected events happen at different geographic areas or different moving objects become interesting at the same time.

In order to help technical directors to obtain the best broadcasting results, we present in this paper a system that helps them to select the best candidate video signals coming from static or mobile cameras (i.e., cameras installed in rowing boats, in other sailing boats, in fixed locations, etc.). Our proposal, based on the LOQOMOTION system [9] extended with videocamera management, relies on mobile agent technology to bring the processing to the best place in the distributed wireless scenario, at any time. Thus, the camera selections provided by the system are updated continuously in an efficient manner, and it is possible to deal with different geographic areas of interest at the same time. The system can even alert about upcoming situations defined previously by the technical director, or when some event happens (e.g., focus a certain area as soon as any camera gets close enough).
The main contributions of our proposal are:

- We extend a general architecture for location query processing with videocamera management to help technical directors of sport events to deal with the multimedia information coming from different (static or moving) videocameras in order to cover different events.
- The system is able to focus on (previously or in runtime) defined geographic areas or (static or moving) objects, to provide technical directors with the best shots of such areas of interest.
- A flexible approach is followed, and therefore new functionalities can be added to the system in the near future without compromising the main architecture.

The rest of the paper is structured as follows. In Section 2 we detail the features of the sample sport event used as motivating context and some interesting multimedia location queries that we would like to process automatically. In Section 3 we introduce the concept of location-dependent queries, a general architecture for location-based query processing, and our proposal to extend it with the modeling and management of videocameras. In Section 4 we explain how queries in the sample scenario are processed by the proposed architecture. In Section 5 we review some related works. Finally, conclusions and future work are included in Section 6.

2. MOTIVATING CONTEXT: ROWING RACES IN SAN SEBASTIÁN

Rowing racing is a very popular sport in every seaside town of fishing tradition along the north of Spain (see Figure 1). In the Basque Country this sport is of a great importance, and almost every town has its own team, which always counts with the unconditional local support.

Figure 1: Boat from Castro Urdiales town

Although there are multiple races, the most important one is celebrated in San Sebastián bay, once a year, since 1879. The boats leave from the harbor where they come back after making a turn in the sea, outside the protection of the bay, covering a total distance of 3 miles. It is important to remark that there are four lanes but those lanes are only physically distinguished in the starting and turning points (see Figure 2).

Figure 2: Four boats right after the race starts

We have chosen this race due to the fact that it is a paradigmatic example of a potential live broadcasting event that can benefit from our work. In this race, celebrated during two consecutive weekends every September, the city of San Sebastián is *crowded* with many visitors, and the audience of the Basque TV broadcaster is very important. The technology and equipment involved in the event have been evolving during the last years and nowadays include multiple cameras (in sailing boats, in a helicopter, in the harbor, in an island nearby, etc.), a GPS transmitter on every boat, a software application for the panel of judges to help them to determine the distance between the boats and the location of a boat with respect to its lane, a software tool to show on TV the real positions of the boats in a 3D reconstruction of the bay, microphones to get the atmosphere sounds in different places, and so on.

In such a scenario, it is possible to attach a camera to each racing boat. However, it would be crucial to support the technical director in order to integrate the signal received from those cameras, considering that they should be remotely controlled and that are continuously moving. We would like to highlight the fact that in many cases these cameras will provide innovative and interesting shots from the content production perspective. For instance, these cameras may record the exciting moment of the turning in the sea, the view of the audience perceived by the rowers, overtaken manoeuvres, unusual ocean-side views of the island located in the middle of the bay, etc. Those shots or points of interest can be classified into predefined ones (e.g., the view of spectators in the harbor) or defined in live (e.g., the view of an overtaking manoeuvre between two boats captured by a third one). Figure 3 shows graphically some of the predefined points of interest overlapped in a real 3D reconstruction of the race scenario.

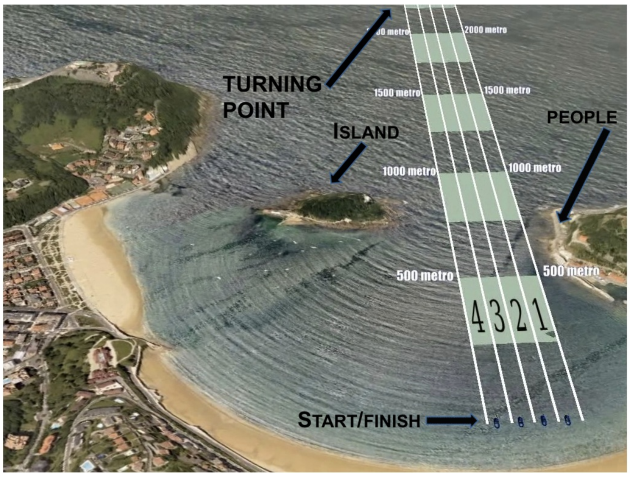


Figure 3: Predefined points of interest

It is important to emphasize that receiving the video signals is not the problem in this scenario, but the selection among multiple video signals. The system described in the following section helps the technical director in the identification and management of the best candidate videocameras in order to shot a concrete (static or moving) area or object of interest, whether it is predefined or not.

In the following, we enumerate some motivating queries that we would like to be answered and updated continuously:

1. *Query 1: View a certain boat.* The technical director could focus on a particular boat due to many reasons (it is the local team, it is leading the race, etc.), some of them caused by unexpected situations (a turn over, a broken oar, etc.) which obviously should be captured quickly from any camera.
2. *Query 2: Record a good side view of any boat.* The technical director could want to broadcast a close side view of some boat to show the big effort performed by its team of rowers during the race.
3. *Query 3: Record a wide view of the island from the ocean side.* This is interesting for technical directors because a view of the island from the ocean is usually very spectacular (it is a sheer cliff full of seagulls) and not as typical as the well-known view of the island from the bay. For this query, technical directors are interested in cameras located far from the island (to capture it completely) but within a certain range (to get also a detailed picture). So, for this query, the locations of the cameras play an important role.

Other areas that are very interesting for technical directors, and therefore could be the target of similar queries, are: the *ciaboga* area (where boats turn, which is a key part of the competition), the area of the harbor or the promenade at the seaside (usually crowded with people watching the race), etc. It would be interesting for technical directors to be able to predefine areas that are usually interesting at some time during the race. However, the system should also allow technical directors to select, in live, any area of interest by dragging the mouse on a map.

Recording Cameras

Nowadays, most of the light cameras employed in scenarios similar to the one described (e.g., F1 car races, sailing, etc.) offer a fixed view (e.g., a front view from the car driver). From the production perspective, this implies important difficulties for the generation of attractive and rich content, mainly due to the limitation or lack of control of the *zoom* (adjustment of the focal length of the lens to make the target appear close-up or far away), *pan* (horizontal movement), and *tilt* (vertical movement) of the camera.

However, the electronic consumer sector is providing new cameras with very competitive prices that provide an acceptable image quality while providing rich remote control functionalities. Figure 4 shows some examples of these cameras, that allow a remote control of zoom, tilt, and pan parameters.

(a) Axis 213 PTZ (b) Edimax IC-7000 PTn

Figure 4: Remote motorized cameras

3. MONITORING PROPOSAL

As the underlying architecture for a location-aware system for monitoring sport events, we advocate the adoption of a general-purpose location-dependent query processing system [9] that will be extended with the needed new functionalities. In order to get a better understanding of that system, in this section we present some basic concepts related with it. So, we first define and explain the concept of location-dependent query. Then, we summarize the general architecture proposed for location-dependent query processing. Finally, as one of the contributions of this paper, we indicate the new functionalities added to this system to manage references to videocameras in location queries.

3.1 Location-Dependent Queries

Location-dependent queries are queries whose answer depends on the locations of the objects involved. For example, a user with a PDA may want to locate available taxi cabs that are near him/her while he/she is walking home in a rainy day. These queries are usually considered as *continuous queries*, whose answer must be continuously refreshed.

For example, the answer to the previous query can change immediately due to the movements of people and taxi cabs. Moreover, even if the set of taxis satisfying the query condition does not change, their locations and distances to the user could change continuously, and therefore the answer to the query must be updated with the new location data.

To express location-dependent queries, we will use an SQL-like syntax with the following structure:

SELECT projections FROM sets-of-objects WHERE boolean-conditions

where *projections* is the list of attributes that we want to retrieve from the selected objects, *sets-of-objects* is a list of object classes that identify the kind of objects interesting for the query, and *boolean-conditions* is a boolean expression containing objects from *set-of-objects* that must satisfy the specified *location-dependent constraints*.

As example of a location-dependent query, the query in Figure 5 asks for rowing boats that are within 0.2 miles around *boat38*. This query includes an *inside* constraint expressed with the general syntax *inside(r, obj, target)*, which retrieves the objects of a certain *target* class (such objects are called *target objects*) within a specific distance *r* (which is called the *relevant radius*) of a certain moving or static object *obj* (that is called the *reference object*). Thus, in the sample query in Figure 5 the radius of the inside constraint is 0.2 miles, there is one reference object (*boat38*), and one target class (*RowingBoats*).

```
SELECT   B.id
  FROM   RowingBoats AS B
 WHERE   inside(0.2 miles, boat38, B)
```

Figure 5: Sample location-dependent query

Moving and static objects in a scenario are not single points but have a certain geographic extension, depending on their size. Thus, objects and areas are managed in the same way (an object is characterized by an area, which is its *extent* [16]). The techniques described in [8] are used to manage queries that take into account the extent of the objects involved.

Class *Objects* is the set of entities in the scenario, which could be equipped with a camera. *RowingBoats* is a subclass of *Objects*. An individual of *Objects* can be represented by the tuple <*id, name, extent, centroid, frontVector, camera*>, where *id* is a unique identifier of the object, *name* is the name of the object, *extent* is the area occupied by the object, *centroid* is the centroid of the extent of the object, *frontVector* is a vector pointing towards the frontal part of the object (see Section 3.3), and *camera* is the camera the object is equipped with (if any). For an example of the main elements characterizing a rowing boat, see Figure 6.

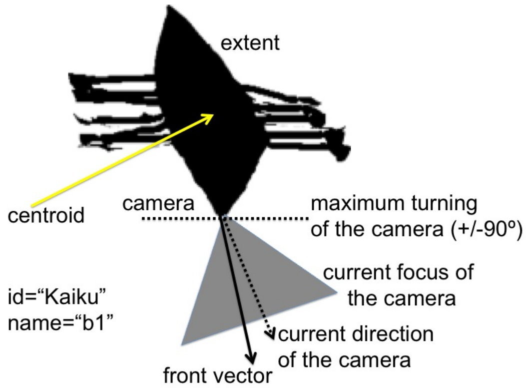


Figure 6: Main elements of a rowing boat

3.2 Processing Location-Dependent Queries

To process location-dependent queries in a mobile environment, we have proposed the system LOQOMOTION [9] (*LOcation-dependent Queries On Moving ObjecTs In mObile Networks*), a distributed location-dependent query processing system whose architecture is based on mobile agents. Mobile agents [2, 19] are programs that execute in contexts called *places*, hosted on computers or other devices, and can autonomously travel from *place* to *place* resuming their execution there. Thus, they are not bound to the computer where they were created; instead, they can move freely across different computers and devices. Mobile agents provide interesting features for distributed and wireless environments (e.g., see [17]), thanks to their autonomy, adaptability, and capability to move to remote computers.

LOQOMOTION deploys a network of agents to perform the query processing over a distributed set of objects which can detect objects moving within their range. Notice that a certain object could only detect a subset of objects in a scenario because of its limited range. The basic idea is that mobile agents move among objects in order to detect the target objects that are relevant for a query. As an example, let us assume that in the scenario shown in Figure 7 the object *Monitor* wants to retrieve the rowing boats within the area *S* centered on the black object *ref*. In the figure, we represent with rectangles the objects that have the capability to detect other objects within its range (*coverage area*). The query will by processed by an agent called *MonitorTracker*. To do that, it first needs to know the location of *ref*. Notice that object *ref* is beyond the coverage area of the *Monitor* object, and so the system will need to ask some other object that is able to detect the *ref* object. In particular, the location of *ref* is known by *Object2* (as *ref* is within its area). So, a mobile agent called *Tracker* travels to *Object2* to retrieve the current location of *ref*. Once the system knows the location of *ref*, it also knows exactly the circular area of interest *S*. Then, the rowing boats within that area are obtained, by using agents called *Updaters*. Thus, one Updater executing on each object whose range intersects with *S* (in the example, *Object2* and *Object3*) will keep track of the rowing boats entering *S*. Of course, as the interesting objects move, the network of agents will re-organize itself as needed (e.g., notice that when *ref* moves the area *S* moves as well). For more details, see [9, 10].

3.3 Extension to Monitor Multimedia Data: Access to Videocameras

In our context, one of the most interesting attributes of the objects in the scenario are obviously the videocameras, which play a key role for us. Thus, we do not only need to retrieve the objects that satisfy certain location-dependent constraints, but also filter out those objects whose cameras do not satisfy certain conditions/features needed for a suitable recording of the target (i.e., a static or moving object

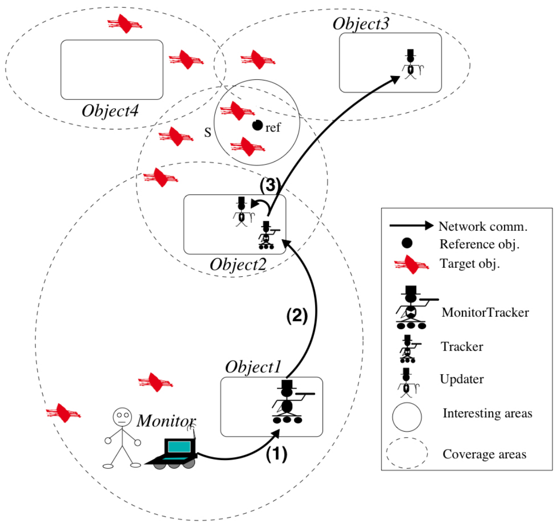
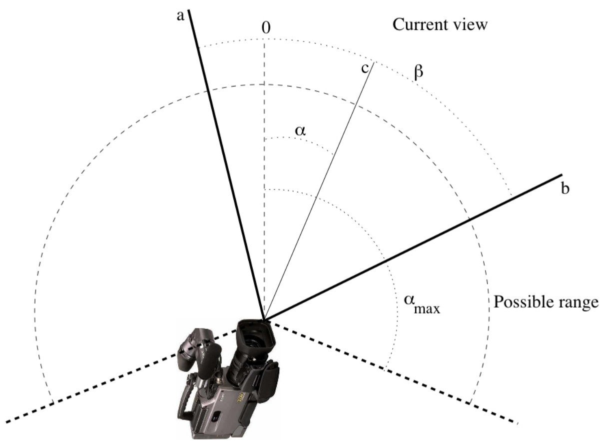


Figure 7: Architecture of LOQOMOTION

or area). For this purpose, we model a camera c as shown in Figure 8.



$$c = < id, \alpha, \alpha_{max}, \alpha_{speed}, \beta >$$

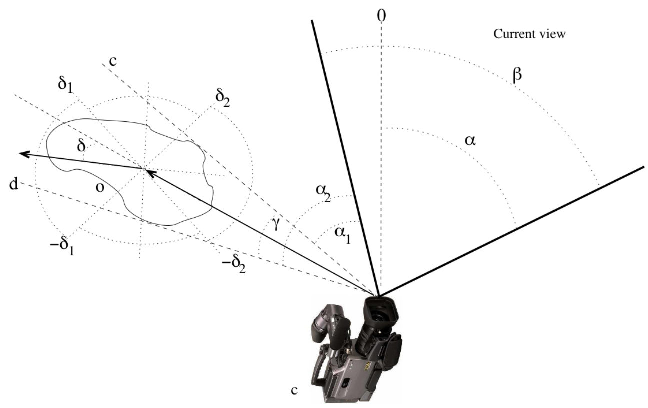
Figure 8: Modeling a videocamera

In the figure, *id* is a unique identifier; α, α_{max}, and α_{speed} are the current pan, the maximum pan possible (it can pan from $-\alpha_{max}$ to α_{max}), and the pan speed (degrees/second) of such a camera; finally, β is the wide of the focus (limited by lines a and b). Due to space limitations we do not consider here the vertical movement (tilt), which would be very similar to the pan model, or the zoom.

We define some functions that abstract us from the specific calculations needed, based on the features of the cameras, to verify certain conditions. Particularly, we define the function *panToView* (see Figure 9).

The function returns the signed[1] angle that the videocamera c should pan to view an object o from a certain view v (*front*, *rear*, *side*, or *any*) with the specified coverage *cov* (*full*, *incomplete*, *any*). Lines c and d are traced from the camera location to both sides of the geographic area associ-

[1]Positive values mean "right pan", and negative ones mean "left pan".



$$panToView(o,c,v,cov) = \begin{cases} \alpha_1 + \epsilon & cov \in \{incomplete, any\} \\ & \wedge(view(c,o) = v \vee v = any) \\ & \wedge - c.\alpha_{max} \leq \alpha + \alpha_1 \leq \\ & c.\alpha_{max} \\ \alpha_2 & cov = full \wedge \gamma \leq \beta \\ & \wedge(view(c,o) = v \vee v = any) \\ & \wedge - c.\alpha_{max} \leq \alpha + \alpha_2 \leq \\ & c.\alpha_{max} \\ -999 & otherwise \end{cases}$$

Figure 9: *panToView* function

ated to object o. It is important to consider that cameras usually have mechanical limitations about the possible turn angles $(-\alpha_{max}, \alpha_{max})$. So, when the camera cannot pan to view object o, satisfying the specified conditions, the above function returns a special value (in our prototype, -999). In the previous formula, ϵ is a small value greater than 0, which must be added to α_1 in order to start having the target object within the view of the camera.

Function $view(c,o)$ returns $v \in \{front, rear, side\}$ by considering in which of the four quadrants[2] defined by the angles $\pm\delta_1$, $\pm\delta_2$ falls the angle between 1) the line defined by the location of the camera c and the (centroid of the area of the) target object o, and 2) a predefined vector which indicates the *front* of the object o (such an angle is denoted by δ in Figure 9). The *view* function is defined as follows:

$$view(c,o) = \begin{cases} rear & |\delta| \leq \delta_1 \\ side & \delta_1 < |\delta| \leq \delta_2 \\ front & \delta_2 < |\delta| \end{cases} \quad 0^\circ \leq \delta, \delta_1, \delta_2 \leq 180^\circ$$

So, according to the location of the object and the camera in Figure 9, in that scenario *view(c, o)* returns "rear". Also, it should be noted that if a camera must pan to focus the target it will need some time to perform the turning. Therefore, we define *timeToView*, which returns the time (in seconds) needed to pan α degrees the camera c:

$$timeToView(o,c,v,cov) = \frac{panToView(o,c,v,cov)}{c.\alpha_{speed}}$$

These three functions will be used in the next section to take into account videocameras' features in location-dependent queries.

[2]The definition of front, rear, and side views could be defined by any other partitions.

4. USING LOQOMOTION IN THE ROWING BOATS SCENARIO

In this section, we first indicate how the different elements of the query processing architecture apply in the context of row races in San Sebastián:

- *Distributed Query Processing.* In the context of row races in San Sebastián there is a single object (a TV trailer) that is able to access all the objects (boats, people, etc.) in the scenario. Although LOQOMOTION is particularly adapted to perform well in a distributed infrastructure where there are different objects that can monitor different geographic areas, it can obviously also work when there is a single object covering the whole area of interest.

- *Inside Constraints.* As cameras that are very far from their target are usually of little interest, an *inside* constraint with an appropriate relevant radius can be used to retrieve the candidate cameras for recording the kind of shot (close, wide open, etc.)[3] we want.

- *Reference Objects.* In the context of row races, the reference object is the interesting object (e.g., a particular boat, the *island*, etc.) that must be recorded. As described in Section 3.1, the extent of the objects is considered when processing the queries.

- *Target Objects.* In the context of row races, the target objects are the objects (e.g., boats) that have cameras that may satisfy the conditions required to record the area of interest. For example, cameras located in boats or fixed cameras in the harbor are part of the answer to location queries.

As seen above, the proposed architecture fits the context of row races in San Sebastián. In the rest of this section, we will analyze how interesting sample queries described in Section 2 can be expressed (using an SQL-like syntax) in a way that allows their processing with the proposed architecture. For illustration, we will consider the scenario shown in Figure 10, where we assume that all the boats have a camera situated in their bow, which can turn +/-90°.

4.1 Query 1: View a Certain Boat

Let us suppose that we want to retrieve the cameras that can focus on a particular boat, for example the one named "Kaiku". The query would be expressed as follows:

```
    SELECT   O.camera.id, pan, time
      FROM   Objects AS O, RowingBoats AS B
     WHERE   pan=panToView("Kaiku", O.camera,
               any, full) AND pan<>-999
             AND time=timeToPan(O.camera,pan)
  ORDER BY   time ASC
```

Besides the identifiers of the cameras, it returns the number of degrees (*pan*[4]) that they should pan to focus the boat "Kaiku" (from any angle but providing a full view) as well as the time needed by that camera to do it (*time*). The cameras that currently have a full view of the "Kaiku" boat (whether it is a rear, side, or front view) appear first in the answer (*pan*=0, *time*=0), and the more time they need to

[3]In this work we do not deal with the possibility of zooming.
[4]Remember that it is a signed number; a negative value means "left pan" and a positive value means "right pan".

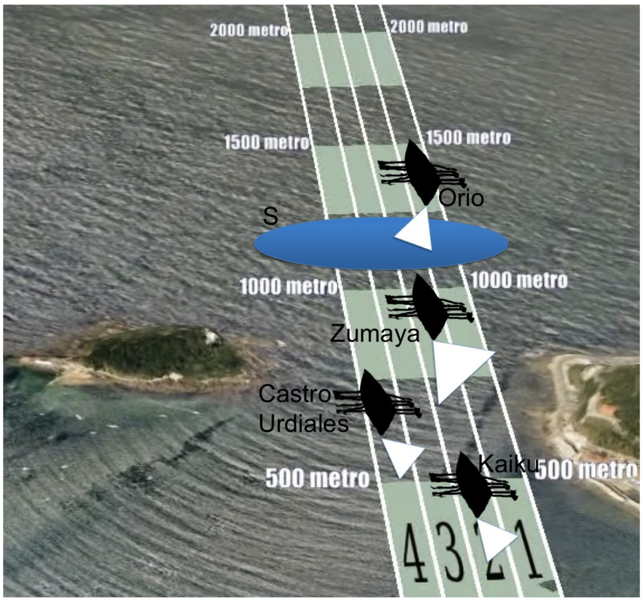


Figure 10: Sample scenario for the example queries

pan (to fully view the reference object "Kaiku") the later they appear in the answer because of the *order by* clause. For example, in Figure 10 the camera of the boat "Castro Urdiales" would be retrieved first (as it has a full view of the rear side of "Kaiku") and the one of "Zumaya" would be ranked in the second position (as it can turn slightly to have a full view of the rear side of "Kaiku"). Cameras that cannot view that object (e.g., the one of the boat "Orio" in Figure 10) are not included in the answer because of the *pan* <> −999 constraint (see Section 3.3). However, as this query is evaluated continuously, the answer is always up to date. Thus, the information retrieved by the continuous query will help technical directors to select the best shots.

4.2 Query 2: Record a Good Side View of Any Boat

Let us imagine that we want to retrieve only cameras which are currently streaming a good side view of any boat. This query would be expressed as follows:

```
    SELECT   O.camera.id
      FROM   Objects AS O, RowingBoats AS B
     WHERE   pan=panToView(B.id, O.camera, side, any)
             AND pan=0 AND inside(250 meters, B.id, O)
  ORDER BY   distance(B.id, O.id)
```

where we assume that a camera must be located no further than 250 meters in order to provide a good (full or incomplete) view of that boat. Moreover, we order the cameras according to the distance to the target, as closer cameras are more likely to be able to provide detailed views. Thus, for example, in the scenario of Figure 10 the camera of "Castro Urdiales" would be a good candidate to record the boat "Kaiku", as the query above does not require a full view but "any". If a full view was required instead and the constraint *pan=0* was replaced by *pan<>-999*, then "Zumaya" would be a better candidate to record the boat "Castro Urdiales", even though the distance between "Zumaya" and "Castro Urdiales" is higher than the distance between "Castro Urdiales" and "Kaiku" and besides "Zumaya" would need to turn its camera to record "Castro Urdiales".

4.3 Query 3: Record a wide view of the island from the ocean side

For this query, let us assume that the technical director previously defined an area S (see Figure 10), on the ocean side, which represents where a camera should be located in order to have the wanted wide view of the island from the ocean side. Thus, the query could be expressed as follows:

```
SELECT    O.camera.id, pan, time
FROM      Objects AS O
WHERE     inside(0 meters, S, O) AND
          pan=panToView(island, O.camera, any, full)
          AND pan<>-999
          AND time=timeToPan(O.camera,pan)
ORDER BY  time ASC
```

where with the constraint *inside(0 meters, S, O)* we require objects no further than 0 meters from S (that is, objects within S). In the scenario shown in Figure 10 no camera is currently within the area S. However, the *Orio* boat will enter soon S and so it will become part of the answer to the continuous query. It should be noted that the technical director may want to change at any time the size and location of the area S, in order to focus a different area (e.g., an accident that has happened). The system is able to adapt to changes in the monitoring requirements.

5. RELATED WORK

In this section, we analyze some related works in two different categories: works that focus on the use of multiple cameras to produce contents and works that are closer to the field of location-based services.

5.1 Works on Multi-Camera Management

Regarding the usage of several cameras for the production of professional content, there are multiple initiatives that tackle this issue with different goals and from different perspectives. For example, in [1] the authors use an array of cameras in order to acquire a 3D scene. In [3], the goal is to use multiple cameras to automatically infer what is happening in a tennis game. As a final example, in [6] several cameras are used to automatically monitor the movements of football players. In all these cases, the focus is on the operation and exploitation of the cameras, and not on the complexity of managing multiple video signals in real time.

Concerning the management of multiple cameras and screens, an interesting and representative work can be found in [5], regarding different approaches for concurrent access and management of multiple static cameras. It is also worth mentioning the proposal in [21], where the authors work on the optimal monitoring of multiples video signals. However, it is important to emphasize that these representative works on multi-camera management and monitoring consider only cameras that are static. On the contrary, the cameras considered in our proposal can move and pan.

Finally, we would like to mention the work presented in [13] regarding the integration in a 3D terrain model of the data coming from GPS receivers embedded in boats during the broadcasting of live rowing events. This work was performed to support the technical director for those situations when, due to bad weather conditions, a helicopter cannot be employed. This is a work complementary to the one presented in this paper, as it could provide a nice graphical interface where the answers obtained by our system could be shown.

5.2 Works on Location-Dependent Queries

As it is an important issue to build Location-Based Services (LBS), a major research effort is being invested in the development of systems that can effectively process location-dependent queries [11]. The existing proposals differ in different aspects, such as the types of queries that they can process, or the assumptions that they rely on and/or the underlying infrastructure required. For example, *MobiEyes* [7] requires the cooperation of the moving objects that are the targets of the queries to process the queries distributively, *LOQOMOTION* [9] performs a distributed processing on certain objects, *DOMINO* [22] exploits some knowledge about the trajectories of the objects, and *SEA-CNN* [23] focuses on the processing of continuous kNN queries.

However, we are not aware of any work that has applied an architecture for processing location-dependent queries in the context of sport events to retrieve relevant multimedia data. Although there are works that emphasize the importance of location-dependent query processing for multimedia computing (e.g., [12]), they do not consider that the queries themselves can have as a final goal to retrieve multimedia data that are relevant for the user. Works that propose location-dependent multimedia services for mobile users (e.g., [15], where a middleware based on mobile agents is presented) are not directly related either to the research presented in this paper.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown the usefulness of location-dependent query processing in a real-world sport event: the row races in San Sebastián. With that purpose, we have extended a general architecture for location-dependent query processing with the capabilities needed to detect suitable (static or moving) cameras to record a particular area. The main features of our proposal are:

- Videocameras' features are considered to help technical directors to access the multimedia information coming from different (static or moving) videocameras.
- Predefined or dynamically built geographic areas or static/moving objects can be set as targets for which suitable cameras can be found by the system.
- The proposal is flexible, and so new functionalities can be added to the system without important changes to the main architecture.

As future work, we plan to evaluate the proposal through simulations and perform some tests in real environments. Besides, we will extend the system to enable automatic recording of targets by videocameras, in order to further help the technical directors with the recording process. In some scenarios, some useful multimedia information could also be provided by mobile devices carried by people or tourists, and so exploiting the possibilities of a mobile peer-to-peer network would be interesting.

7. ACKNOWLEDGMENTS

This research work has been supported by the CICYT project TIN2007-68091-C02.

8. 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] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE*. Wiley, 2007.

[3] C. O. Conaire, P. Kelly, D. Connaghan, and N. E. O'Connor. TennisSense: A platform for extracting semantic information from multi-camera tennis data. In *16th International Conference on Digital Signal Processing (DSP'09), Santorini, Greece*, pages 1062–1067, 2009.

[4] A. Corral, Y. Manolopoulos, Y. Theodoridis, and M. Vassilakopoulos. Algorithms for processing K-closest-pair queries in spatial databases. *Data & Knowledge Engineering*, 49(1):67–104, 2004.

[5] G. W. Daniel and M. Chen. Interaction control protocols for distributed multi-user multi-camera environments. *Systemcis, Cybernetics and Informatics*, 1:29–38, 2003.

[6] W. Du, J. bernard 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), Graz, Austria*, pages 2–13, 2006.

[7] B. Gedik and L. Liu. MobiEyes: A distributed location monitoring service using moving location queries. *Transactions on Mobile Computing*, 5(10):1384–1402, October 2006.

[8] S. Ilarri, E. Mena, and C. Bobed. Processing location-dependent queries with location granules. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops (PerSys'07), Vilamoura, Algarve, Portugal*, volume 4806 of *Lecture Notes in Computer Science (LNCS)*, pages 856–866, November 2007.

[9] S. Ilarri, E. Mena, and A. Illarramendi. Location-dependent queries in mobile contexts: Distributed processing using mobile agents. *IEEE Transactions on Mobile Computing*, 5(8):1029–1043, August 2006.

[10] S. Ilarri, E. Mena, and A. Illarramendi. Using cooperative mobile agents to monitor distributed and dynamic environments. *Information Sciences*, 178(9):2105–2127, May 2008.

[11] S. Ilarri, E. Mena, and A. Illarramendi. Location-dependent query processing: Where we are and where we are heading. *ACM Computing Surveys*, 42(3):1–73, March 2010.

[12] A. Krikelis. Location-dependent multimedia computing. *IEEE Concurrency*, 7(2):13–15, April/June 1999.

[13] M. Laka, I. García, I. Macía, and A. Ugarte. TV sport broadcasts: Real time virtual representation in 3D terrain models. In *3DTV Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-Con'08), Istanbul, Turkey*, pages 405–408, 2008.

[14] S. Qiao, C. Tang, H. Jin, T. Long, S. Dai, Y. Ku, and M. Chau. PutMode: prediction of uncertain trajectories in moving objects databases. *Applied Intelligence*, page 17, 2009. Published online: March 13, 2009.

[15] M. H. Raza and M. A. Shibli. Mobile agent middleware for multimedia services. In *Ninth International Conference on Advanced Communication Technology (ICACT'07), Phoenix Park, Korea*, pages 1109–1114. IEEE Computer Society, 2007.

[16] P. Rigaux, M. Scholl, and A. Voisard. *Spatial databases with application to GIS*. Morgan Kaufmann Publishers Inc., 2002.

[17] C. Spyrou, G. Samaras, E. Pitoura, and P. Evripidou. Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies. *Mobile Networks and Applications*, 9(5):517–528, 2004.

[18] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *ACM SIGMOD International Conference on Management of Data (SIGMOD'04), Paris, France*, pages 611–622. ACM Press, 2004.

[19] R. Trillo, S. Ilarri, and E. Mena. Comparison and performance evaluation of mobile agent platforms. In *Third International Conference on Autonomic and Autonomous Systems (ICAS'07), Athens, Greece*, page 41 (6 pages). IEEE Computer Society, 2007.

[20] J. Veijalainen and M. Weske. Modeling static aspects of mobile electronic commerce environments. In *Advances in Mobile Commerce Technologies*, pages 137–170. IDEA Group Publishing, 2003.

[21] J. Wang. Experiential sampling for video surveillance. In *First ACM International Workshop on Video Surveillance (IWVS'03), Berkeley, California*, pages 77–86. ACM Press, 2003.

[22] O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases that track mobile units. *Distributed and Parallel Databases*, 7(3):257–287, July 1999.

[23] X. Xiong, M. F. Mokbel, and W. G. Aref. SEA-CNN: Scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases. In *21st International Conf. on Data Engineering (ICDE'05), Tokyo, Japan*, pages 643–654. IEEE Computer Society, 2005.