

A Mobile Augmented Reality System to Enjoy the Sagrada Familia[†]

J. Gimeno¹, F. Pardo², P. Morillo¹ and M. Fernández¹

¹Instituto de Robótica, Universidad de Valencia, Spain

²Departamento de Informática, Universidad de Valencia, Spain

Abstract

SAFAR is a mobile Augmented Reality application that makes possible to see in real-time the ‘Sagrada Familia’ as it will look like when it be finished. The application successfully integrates outdoor tracking and 3D rendering processes to provide an augmented view of this under construction monument to city travellers, equipped with Nokia mobile phones. This outdoor tracking process includes a custom skyline detection algorithm, which provides real-time tracking and avoids not only external calibrations, but also additional sensors connected to the mobile phone. Moreover, the occlusion problem of the resulting images are improved by means of predefined drawings sequences, which control the combination between computer-generated and displayed video images. The performance evaluation results show that both the outdoor tracking and the 3D rendering techniques combined in SAFAR requires relatively low computational load, while providing a proper accuracy.

Categories and Subject Descriptors (according to ACM CCS): Information Interfaces and Presentation [H.5.1]: Multimedia Information Systems—Artificial, augmented and virtual realities; Image Processing and Computer Vision [I.4.8]: Scene Analysis—Tracking

1. Introduction

Augmented Reality (AR) has already been used, among other purposes [Azu95], to add 3D contextual information for outdoor archaeological sites and to reproduce ancient cities on their own current ruins [VVMT02]. In this sense, this technology can be used to show the final aspect of under construction buildings before they are totally finished. The motivation of this work has been to develop a mobile AR application, called SAFAR, oriented to improve the user experience in visiting one of the most visited tourist places in Europe.

SAFAR (which stands from ‘Sagrada Familia’ Augmented Reality) is a mobile application which offers 2D/3D contextual information of the ‘Sagrada Familia’ monument to visitors. This information is provided by means of a mo-

bile phone, which acts both as visualization system and as data acquisition system. The ‘Sagrada Familia’ is a monumental church located in Barcelona (Spain) and was designed by Antonio Gaudi. The architect started the construction of this world-wide symbol of Barcelona and Spain at the end of the XIX century. Since this large church is planned to be finished in 2025, the purpose of SAFAR is to allow city travellers to visualize the final aspect of this under construction monument before the building works are completed. In order to develop a final and useful application within the means of the average mobile phone user, we have selected a Nokia N95 mobile phone as hardware platform, without any type of additional hardware or software improvement.

Nowadays, a standard mobile phone could be considered as a cost-effective and lightweight hardware platform for AR purposes. These devices provide a well-known user interface and are usually fully equipped with, among other hardware elements, accelerometers or on-board cameras for providing video see-through visualization. Moreover, most of the devices offer a fast interface that is relatively easy to program.

[†] This work has been supported the Spanish Ministry of Education, the Science and European FEDER Funds (research project DPI2006-14928-C02-02) and Orange R&D Spain (Barcelona)

In this sense, a large number of recent developments have demonstrated that these devices can support mobile AR application providing sufficient computational capabilities for interactive image processing, computer vision or 3D camera pose estimation [GTx08, SW07, WLS08].

However, these limitations are more restrictive when the mobile AR applications have to operate in outdoor environments, where the weather, scenarios and lighting conditions are uncontrolled and could have a significant impact on the reliability of the system. For this reason, the software designed for mobile AR applications (especially tracking software subsystems) should be robust to variations in illumination, atmospheric conditions, user's viewpoints or the presence of moving objects within the scene.

This paper describes into detail how the real-time tracking and the 3D visualization problems have been addressed in the design of SAFAR. In this sense, SAFAR makes use of a tracking mechanism based on skyline recognition to locate the position of the main parts of the 'Sagrada Familia' monument within the scene. This mechanism only receives as input the images captured by the on-board camera and obtains an accurate estimation of the user's point of view in real-time. Besides, the paper introduces the occlusion problem in AR systems and how it has been solved in SAFAR using predefined drawings sequences. These sequences define a custom visualization order, which controls the combination between computer-generated and displayed video images as they are generated in each frame period. Next, a performance evaluation of SAFAR is presented including throughput, tracking and occlusion results for the presented mobile AR system. The results of this evaluation show acceptable results, in terms of real-time tracking and occlusion, and an average frame rate ranging from 12 to 16 frames per second when SAFAR is executed on a Nokia N95 mobile phone.

The rest of the paper is organized as follows: Section 2 analyzes the related work on interaction techniques for mobiles devices as well as the weaknesses of the existing proposals. Section 3 describes the application scenario for the presented approach and addresses the problem of developing a mobile Augmented Reality system for this type of still under construction monuments. Next, Section 4 shows the proposed tracking mechanism based on skyline recognition. Section 5 describes the prototype application to test our system, which is evaluated in Section 6. Finally, Section 7 presents some concluding remarks and the future work to be done.

2. Previous Work

The term Augmented Reality (AR) is often used to define some computer graphic procedures where the real-world view is superimposed by computer-generated objects in real-time [Azu95]. Unlike the Virtual Reality, where the user is provided with a completely natural experience in realistic

simulated worlds, the goal of the Augmented Reality is to realistically enhance the user's view of the real-world with visual information provided by virtual objects. Following this general definition, the main AR systems are classified according to the employed visualization systems: retinal displays, head-mounted displays (HMDs), hand-held devices and enhanced projection systems [BR05]. The next sections centers around the hand-held devices since they mainly support the AR systems based on mobile computing.

2.1. Mobile Augmented Reality

Mobile AR is an advanced and innovative technology where mobile devices equipping on-board cameras such as cell phones, PDAs or pocket-side computers become the hardware support for the development of AR applications [HFTR99]. The current challenges of mobile AR systems are to apply this concept in truly mobile scenarios; that is, away from the carefully conditioned environments of research laboratories and special-purpose work areas [KH04].

The first developments of mobile AR systems started by the middle of the 1990s when the mobile computers and the tracking devices became small and powerful enough to support AR applications in dynamic mobile settings. The Columbia Touring Machine [FMH97] is considered as one of the earliest developments of an outdoor mobile AR system. This prototypes includes an HMD, connected to a bulky backpack, to show visual tour guide information to campus visitors of the University of Columbia, merged with some buildings and monuments located within the campus.

Some years later, the launch of the current third generation mobile phones, which includes important hardware capabilities such as more powerful processor, high resolution displays or even 2D/3D graphics hardware acceleration, has lead the development of several mobile AR applications. The MARA project [KM06], developed by the Nokia Research Center, is a well-known example of this applications. In this project, the Nokia S60 mobile phone using the tracking information from the attached external sensors shows users 2D graphics and text information about the user's surroundings in real-time. Including MARA, a wide variety of these developments is reviewed in [KH04].

Nowadays, the mobile AR computing is considered as one of the main research areas in the field of Augmented Reality. Mobile systems are currently used in many different technological applications such as [KM06] or [WLS08]. In most of these applications, the developed mobile systems use external sensors or on-board GPS receivers to obtain the current user position. This process is called outdoor-tracking [YNA99] and becomes a key aspect for the development of high-performance mobile AR systems in the near future.

2.2. Outdoor Tracking

Traditionally, the registration problem is considered as one of the main challenges in the field of Augmented Reality [Azu95]. The registration problem consists in properly align real-world images and computer-generated graphics in real-time [YNA99]. Although several AR systems including indoor tracking mechanisms have been presented [MLB04, WLS08, HBO05], none of them has been easily migrated to work on outdoor environments properly. The reason of this limitation consists of outdoor environments use to be nondeterministic and do not usually allow to add artificial markers, to control the lighting conditions or, even, to restrict the user's movements to facilitate the location of the user's positions. This problem is more difficult to solve for AR systems based on mobile computing.

The purpose of most mobile AR applications oriented to outdoor environments is to guide the user through buildings and monuments and to become a tourist guide [KBB04, TKY03]. Since in most of the cases the contextual information is offered to the users as text messages, these applications do not require a high level of visual coherence (registration) between the real-world images and the computer-generated graphics. Moreover, the tracking requirements are fulfilled by the information provided from an on-board or a wireless GPS receiver [HFTR99].

When this tracking information is not accurate enough to locate the user position, some systems introduced advanced location techniques based on hybrid tracking [AHI*99]. Hybrid tracking consists of combining different acquisition mechanisms to obtain a more precise user's location results. An extended technique of hybrid tracking oriented to outdoor configurations corresponds to a combination of GPS and electronic compass signals [KM06]. In this combination, the GPS signal offers an estimation of the current user position whereas the electronic compass achieves accurate measurements of the user direction.

More recent contributions have presented new hybrid tracking techniques, based on computer vision, in order to improve the accuracy of the registration problem [SAYT01, KTO04]. These techniques avoid the use of external sensor devices connected to the mobile phone and obtain the users' point of view by means of analyzing the images captured by the on-board camera. However, they have a high computational cost since the object and pattern recognition, in computer vision, is a complex process that cannot be solved completely by analytical methods and does not admit a general solution [SS01]. The complexity of this problem is increased if the systems work in outdoor environments where not only the lighting conditions, climatology or the apparition of unexpected objects are uncontrollable, but also it is impossible to place external markers within the scene to simplify the tracking problem.

The most extended approach to the tracking problem in

outdoor environments utilizes some type of contextual information of interest to the application in order to simplify the problem [Low04, ZK05]. The AR-PDA project [GFM*03] takes advantage of the 3G network capabilities included in most of current mobile phones. In that project, the images of a camera mounted on the device are encoded and sent via a wireless network to a server. The server analyzes the image, detects objects and decides both the type and the position of the synthetic 3D graphics which mix with the real image. The augmented images are encoded and sent back over the network to the mobile phone which decodes and displays them.

Although the type of the included contextual information differs among different AR applications, the detection of the skyline has been widely validated as a proper tracking mechanism to locate the position of users and objects within the scene [Beh99, KTO04]. The skyline is the outline of a building, city or other things against the horizon. Since the skyline could be considered as the fingerprint of buildings, a unique point of this visual element could be detected and used to recognize and distinguish among facades of buildings. The skyline detection has been used in many different localization problems such as building recognition [ZK05], vehicle self driving [FCLS93] and outdoor-tracking in Mixed Reality systems [KTO04].

The skyline can be modelled as a simple line that can be considered as a mathematical function of one variable. Following this approach, the detection problem could be reduced to find some critical points in this function [Beh99]. This feature will be exploited in the hybrid tracking systems included in the presented mobile AR system.

3. Application Scenario

The 'Sagrada Familia' is a monumental church built by Antonio Gaudi, which is considered as one the worldwide best-known architects of the 20th century. This masterpiece of modernist architecture was declared a Patrimony to Humanity by UNESCO and it is located in the center of the high populated city of Barcelona (Spain), where it receives hundreds of visitors a day. Since Gaudi only completed the main facade of the building (called the Nativity Facade), this zone of the monument gathers the most amount of visitors. For this reason, the system considers a scenario where users have a mobile phone, or PDA, equipped with the most common configuration for mid-range phones such as a on-board camera and a general-purpose application processor. When a user approaches to the Nativity Facade of the 'Sagrada Familia', he sees the church as an under construction building full of cranes and construction workers. At this time, the user can launch SAFAR in this mobile phone and point at the monument. By means of the images captured by the on-board camera, the application recognizes the current point of view of the 'Sagrada Familia' and merges the real-world images with the unbuilt zones of the monument in real-time.

Therefore, the system allows visitors to visualize the final state of the finished ‘Sagrada Familia’ in the next twenty years as well as obtain more information (in text, audio and video format) about the most important parts of the monument.

Figure 1 shows the localization of the monumental church within the urban area of this city as well as the current state of the construction. This figure shows the ten most important places where the visitors visualize the Nativity Facade, which is considered the most important zone of the ‘Sagrada Familia’ and is totally finished on all sides. We have denoted these points as hotspots. Although a building detection process based on computer vision could be an unaffordable task in mobile computing, because of the multiples views of the building to be detected, the urban localization of the ‘Sagrada Familia’ building simplifies the problem. Since the visitors are usually located within the streets converging towards the monuments, the number of feasible views of the monument are limited depending on the hotspots where they are located. In this sense, Figure 1 shows how each hotspot is classified into four different views of the building to be detected and tracked in real-time.

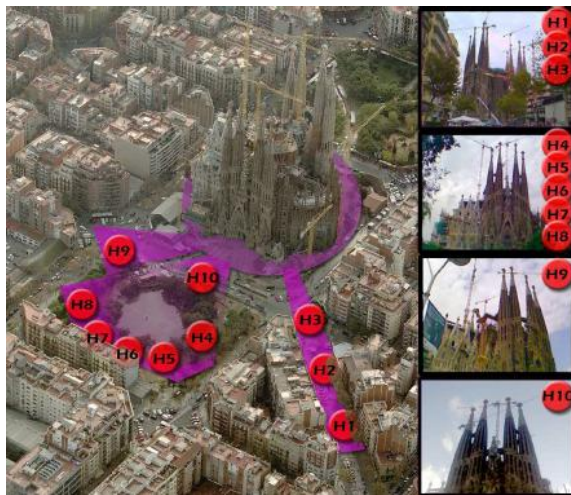


Figure 1: The special location of the ‘Sagrada Familia’ building within the city of Barcelona (Spain)

Since the environment conditions are uncontrollable in this scenario, the behavior of the system should be invariant to lighting conditions and to the apparition of unexpected objects within the scene such as construction workers, moving cranes or trees.

4. Building detection and Tracking

One of the main problems in Augmented Reality is the accurate detection and tracking of the object to be enhanced.

In real-time applications, as the one presented here, the detection and tracking has not only to be accurate but fast. Accuracy and speed are usually opposite and the enhancement of one use to be the worsening of the other. In this application, we have especially focused on the development of a fast, and easy to implement, algorithm due to the limitations of the computing hardware included in most of the current average mobile phones.

A full 2D image processing, using the high resolution images captured by the on-board cameras included in the cell phones, could be an unaffordable task to be performed in real time [SW07, WLS08]. For this reason, alternative strategies, such as linearization and/or binarization are often applied in some mobile phone applications to reduce the amount of data to be processed. One of these techniques, which could be especially suitable for building recognition, is the skyline detection where a 2D image processing problem can be transformed to a less computational intensive 1D image processing task [FCLS93]. Our skyline detection algorithm has specific features to aim our first objective which is the speed of the whole application. It works as follows:

1. Only the blue component of the RGB image is employed in the calculations. The blue component yields the most contrasted image, clearly differencing the sky from the buildings, even in cloudy days.
2. The skyline is calculated by the detection of the edge obtained by a simple threshold which separates the sky and the rest of the image. The first step consists on calculating this threshold. We make the assumption that most of the upmost line in the image is sky, so we calculate the sky value as the average level of this line, which has been previously subsampled to reduce the amount of data to process. We then fix the skyline threshold value as the sky value minus a *security* δ value. In our specific application $\delta = 30$ (illumination level ranges from 0 to 255).
3. The first point of the skyline is calculated looking for the first point below threshold in the first column from the top of the image to the bottom. This is the starting point of the algorithm and there is no need for much accuracy at this stage.
4. There is no need to look for the edge in the whole next column as other algorithms do, instead, we use the last skyline point as the starting point for looking for the edge. If this starting point is sky then the algorithm looks downwards for the edge, else it looks upwards. Since the skyline varies little from column to column, this algorithm runs really fast, having $O(n)$ complexity.
5. As stated before, the skyline detection may become complicated depending on the specific environment. In the ‘Sagrada Familia’ building case, there is a problem with the cranes that are currently in operation; they not only change the building shape, but they even move changing the skyline at any time with an unpredictable behavior. Cranes are removed by further checking edge consistency: if the edge belongs to a piece of building with a

width larger than a w factor, the edge is considered good, else, further checking for an edge is performed below this width w . For the ‘Sagrada Familia’ building application we have found $w = 6$ as a good value to remove crane noise as shown in Figure 2. Higher values for w may be more secure, but also more time consuming.

Once the skyline is obtained, the next step is to segment it in order to extract the building features included in this representation. Each building has its own specific features and may vary from one application to another. The ‘Sagrada Familia’ building has four towers that leave a unique pattern in the skyline. Common pattern recognition algorithms can be very intensive processing tasks even in a 1D line. Instead of using classical pattern recognition techniques, we have developed a fast algorithm oriented to tower recognition. The last step in the building recognition consists of calculating a confidence estimation of the towers positions. If this estimation is positive, the positions (and the zoom factor) of the towers are given along with the most probable point of view of the user as marked in Figure 1.

Although the skyline approximation could seem a reduction of a 2D image processing problem to a linear cost task, the skyline detection could become a complex task depending on the environment conditions, the scene variability or the expected accuracy. The ‘Sagrada Familia’ building can be easily recognized in the skyline by its four high towers. Each tower can be characterized by an increasing line, a maximum at the top and then a decreasing line. The skyline is processed by the algorithm trying to find this pattern. All the matches are stored and the four maxima are chosen as tower position candidates. The process consisting on skyline extraction, crane detection and tower recognition are executed simultaneously while processing the image from left to right. Figure 2 includes two images showing the effect performing a crane correction in order to extract the overall shape of the skyline.

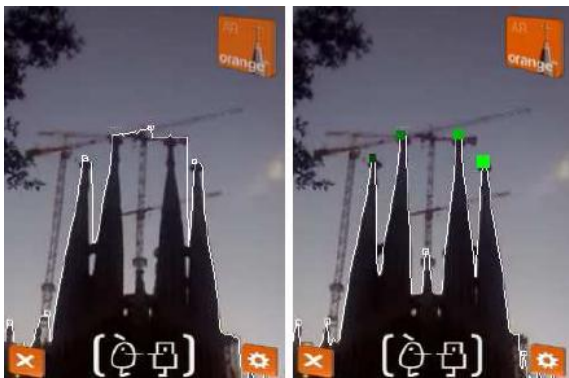


Figure 2: Skyline without crane correction (left) and with crane correction (right)

5. Gaudi’s ‘Sagrada Familia’ AR Application

SAFAR has a software design following an “event-driven object-oriented” architecture. These software designs describe synchronous event-driven architectures composed of a set of components (modules), which are based on a classic object-oriented approach. The modules exchange custom messages which model internal actions or external events captured in real-time. In the case of SAFAR, the software system is composed of six independent and interrelated sub-components, which work concurrently in a real-time manner.

Figure 3 shows the diagram view of the architecture in SAFAR. This figure shows how the proposed architecture relies on a centralized approach around an *AR engine*. This kernel is responsible for launching the application, controlling the user interface and receiving the input data from the *tracking module* and the *keypad module*. The development of the AR engine has been performed using the standard OpenGL ES (OpenGL for Embedded Systems). The *tracking module* implements our building recognition method based on the detection of the skyline from the image flow captured by the *camera module*. However, the main objective of the AR engine consists in merging the real-world images and the computer-generated graphics, which are visualized by the user by means of the *display module*. Finally, the *data module* loads the configuration of the application consisting in a set of XML files, which describes the 3D virtual objects to be blended with the real scene.

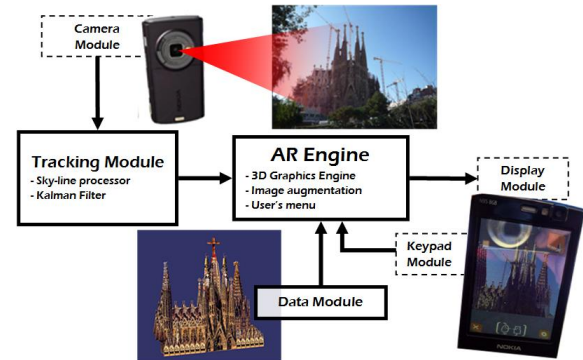


Figure 3: A diagram view of the software architecture in SAFAR

5.1. Tracking Module

The tracking module is in charge of implementing the capture process of the current position of the ‘Sagrada Familia’ building in SAFAR. This module implements the algorithms described in Section 4 as well as a filtering mechanism, which is activated when it is unable to estimate the current position of the building properly.

The execution of the proposed tracking process is as follows: First, the image is captured by the on-board camera of

the cell phone and converted from a native image representation (denoted as CFbsBitmap) to a RGB (red-green-blue) representation. Second, the skyline calculation process is executed and the initial tracking estimation of the 'Sagrada Família' is obtained. Next, these tracking values are updated running a Kalman filter approach [Sor85], where the user's point of view is updated. As a result, the tracking process obtains the updated user's point of view, with an associated trust factor, and a final converted image to be properly visualized in a mobile device.

The tracking module establishes a bidirectional communication to the AR Engine. Besides the tracking information, which is sent to the kernel, the tracking module receives the user configuration from the AR Engine. This configuration is selected by the user by means of the application menu installed within the mobile phone. Although this menu includes several options, the user can select the tracking security level of the application, denoted as δ . This value defines the threshold for the boundary detection process which differences the sky from the buildings in the captured image. By default, δ is equal to 30 and can range from 0 to 50. As this parameter decreases the system is more accurate with relatively low contrast scenes. Otherwise, high contrast scenes are fairly controlled when δ is increased.

5.2. AR Engine

The AR Engine module properly blends the real-world images and the computer-generated graphics from the current camera image and the user's point of view obtained by the tracking module, in real-time.

The execution of the AR Engine is as follows: First, the key events resulting from user's actions are processed. These actions contain visualization options selected by the user from the application menu. Second, the 3D graphic information is generated from the selected visualization mode and the current user's point of view. This process maintains a proper visual occlusion since the computer-generated graphics could be generated behind, or in front, of the real images. Next, a GUI is added to the augmented visual information by means of an overlay layer. Finally, the combined images are visualized in the display of the mobile phone.

The AR Engine has been developed using the standard OpenGL ES (OpenGL for Embedded Systems). Although others graphic libraries such as Performer, OpenSG or OSG provide high-level graphic data structures, called scene graphs [BC07], these libraries are currently not available to execute 3D real-time applications on mobile phone hardware. In order to improve the performance of the rendering process of the mobile application we have adapted to the Symbian operative system our own high-level library [SRC02], which includes a custom scene graph and a culling process. The 3D model corresponding to the 'Sagrada Família' Church, as well as the rest of the objects included

with the scene, have been designed using 3D Studio Max Autodesk tool. During the modelling process, the main parts of the monument such as facades, towers, etc. are labelled in order to allow the AR Engine an easy and real-time identification of the model within the scene graph.

5.2.1. Occlusion management

One of the main problems in the development of AR application consists in maintaining a proper occlusion between the real-world images and the computer-generated graphics. This problem is still more time consuming in mobile AR application because of the low performance hardware included in these systems. In most of mobile AR applications, the computer-generated information is usually located over the real-world images. In our case, we have developed an occlusion technique based on a previously calculated skyline detection. This technique divides the image captured by the on-board camera into two different sections denoted as background and real scene. The background corresponds to the parts of the image located above the skyline, while the real scene includes the zones of the image located below the skyline. When both sections are identified it is possible to perform a proper drawing sequence while maintaining a proper occlusion between real-world and computer-generated information.

In order to show the behavior of this technique the hotspot 5, corresponding to H5 in Figure 1, has been chosen. The proper drawing sequence for a user located in H5 corresponds drawing of background, 3D model representation and finally drawing of the real scene. This drawing sequence represents a 3D model located in front of the sky and properly hidden by the Nativity Facade. Figure 4 shows three images describing the result of the visualization of the 'Sagrada Família' monument from the hotspot 1 using SAFAR. The left image shows the result of the skyline obtained from the camera image. The center image shows the final result obtained in H1 if the total 3D model is managed by a single object. In this case, this object is represented between the background and the real scene. However, the right image shows the result of a proper occlusion of the H1 hotspot, where the 3D model is divided into completed zones and under construction zones (where the apse is located). In this case, the computer-generated representation of the apse is drawn over the real scene.

6. Tests and Results

Several experiments have been conducted to evaluate the system performance and usability of the proposed Mobile AR system. In these experiments, we have selected a set of ten people familiarized with AR applications. The experiments were carried out in three different days where all the participants were executing SAFAR (in all the ten different hotspots) for an average period of time corresponding to eight minutes per hotspot. When the experiments finished,

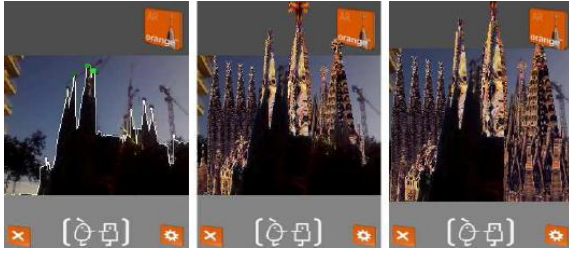


Figure 4: Skyline calculated (left); drawing order background - 3D - real scene (center); correct drawing order background - 3D (except ABSIS) - real scene - ABSIS (right)

they scored usability, interactivity and tracking in a scale from 0 (worst) to 10 (best) as well as providing additional comments and suggestions about the proposed system. The Mobile phone used in the test was a Nokia N95 8G, equipped with an ARM-11 processor, 160 MB of RAM and Symbian S60 3rd Edition Feature Pack as operating system. In order to register the actions of the users as they were performed, the application was modified to produce a log file with different parameters concerning the tracking algorithm as well as other performance parameters. Moreover, some subjective impressions from non expert users were required to evaluate the user experiences about the usability and interest of the application. Figure 5 shows a real execution of SAFAR in front of the main facade of the ‘Sagrada Familia’.



Figure 5: A real execution of the SAFAR application in front of the main facade of the ‘Sagrada Familia’ monument

One of the main system performance parameters in mobile AR systems corresponds to the system throughput, defined as the effective number of frames per second (denoted as FPS) displayed by the mobile device [Azu95, KH04, SW07]. When SAFAR is executed on a Nokia N95 mobile phone, the system maintains a stable frame rate ranging from 12 FPS to 16 FPS. This frame rate is obtained regardless

of the different locations of the hotspots since the 3D models visualized from each hot point are similar (in terms of polygon load) and the tracking algorithm is not affected by the variation of the point of view point of the users. This maintained frame rate is not low for a mobile AR application [HBO05, WLS08] and provides the user an adequate interactive interface. In this sense, the system obtained an average score of 8.5 on interactivity. The main reason for not achieving a higher throughput is related to the conversion of the captured image from the original format (FbsBitmap) to the internal texture format of the graphic engine (OpenGL ES) on a Nokia N95. This process requires an extra conversion process from a BGR to a RGB format for a 320x240 pixels color image.

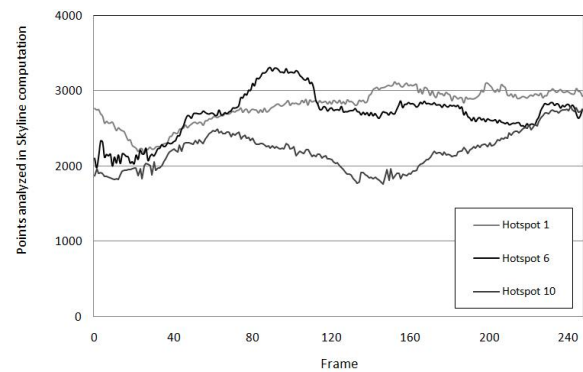


Figure 6: Number of points analyzed during a skyline evaluation

Regarding the performance of the tracking algorithm, the users scored the usability of the overall system with an average of 9.2. Moreover, they scored the quality of the tracking system with an average of 8.1 emphasizing the fast recognition process of the monument as they move the mobile phones. In this sense, it is noteworthy that the proposed algorithm only processes an average of 4% of the whole image in order to obtain a proper tracking of the building within the outdoor scene. Figure 6 shows the amount of analyzed points corresponding to a representative execution of SAFAR composed of 240 frames. The 320x240 pixels color image where obtained in H1, H6 and H10 and the average number of analyzed points ranged from 1.800 to 3.200, from a total of 76.800 pixels existing in the overall image.

7. Conclusions and Future Work

In this paper, we have presented SAFAR a mobile Augmented Reality application specifically designed to provide 2D/3D contextual information of the ‘Sagrada Familia’ monument to city visitors. We have described that it is possible that a mobile AR application can be successfully executed on an outdoor environment. In this sense, some specific features, detected within the selected scenario to be

augmented, have been selected to provide additional information to the tracking system achieving a more accurate tracking results. Concretely, the proposed tracking process includes a custom skyline detection algorithm. This algorithm provides real-time tracking and avoids not only external calibrations, but also additional sensors connected to the mobile phone. The performance evaluation results, obtained on a Nokia N95 cell phone, show that both the outdoor tracking and the 3D rendering techniques combined in SAFAR require low computational load, while providing a proper accuracy.

As a future work to be done, we plan to improve the outdoor tracking algorithm in order to avoid occlusion problems because of skyline estimation errors. Moreover, we expect to develop new versions of the proposed system adapted to other interest buildings and monuments in Europe as well as porting SAFAR to other mobile platforms such as Android, Blackberry, Windows Mobile and Iphone.

References

- [AHI*99] AZUMA R., HOFF B., III H. N., SARFATY R., DAILY M., BISHOP G., VICCI L., WELCH G., NEUMANN U., YOU S., NICHOLS R., CANNON J.: Making Augmented Reality Work Outdoors Requires Hybrid Tracking. In *Proceedings of the International Workshop on Augmented Reality (IWAR '98:)* (1999), A. K. Peters, pp. 219–224. 3
- [Azu95] AZUMA R.: A Survey of Augmented Reality. *Presence* 6 (1995), 355–385. 1, 2, 3, 7
- [BC07] BALE K., CHAPMAN P.: *Scenograph Technologies: A Review*. University of Hull, 2007. 6
- [Beh99] BEHRINGER R.: Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors. In *Proceedings of the IEEE Virtual Reality* (1999), IEEE Computer Society, pp. 244–251. 3
- [BR05] BIMBER O., RASKAR R.: *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Ltd., 2005. 2
- [FCLS93] FANG M., CHIU M., LIANG C., SINGH A.: Skyline for Video-based Virtual Rail for Vehicle Navigation. pp. 207–212. 3, 4
- [FMH97] FEINER S., MACINTYRE B., HOLLERER T.: A Touring Machine: Prototyping 3d Mobile Augmented Reality Systems for Exploring the Urban Environment. *Wearable Computers, IEEE International Symposium* (1997), 74. 2
- [GFM*03] GAUSEMEIER J., FRUEND J., MATYSZCZOK C., BRUEDERLIN B., BEIER D.: Development of a Real Time Image Based Object Recognition Method for Mobile AR-Devices. In *AFRIGRAPH '03: Proceedings of the 2nd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (2003), ACM, pp. 133–139. 3
- [GTX08] G. TAKACS V. CHANDRASEKHAR N. G., XIONG Y.: Girod: Outdoor Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization. In *Proceedings of the ACM International Conference on Multimedia and Information Retrieval (MIR'08)* (2008), pp. 1–13. 2
- [HBO05] HENRYSSON A., BILLINGHURST M., OLLILA M.: Face to face collaborative ar on mobile phones. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)* (October 2005), pp. 80–89. 3, 7
- [HFTR99] HÖLLERER T., FEINER S., TERAUCHI T., RASHID G.: Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. *Computers and Graphics* 23, 6 (December 1999), 779–785. 2, 3
- [KBB04] KRÖSCHE J., BALDZER J., BOLL S.: Mobidemobile Multimedia in Monument Conservation. *IEEE Multimedia* 11, 2 (April–June 2004), 72–77. 3
- [KH04] KARIMI H., HAMMAD A.: *Telegeoinformatics: Location-Based Computing and Services*. Taylor and Francis Books Ltd, 2004. 2, 7
- [KM06] KÄHÄRI M., MURPHY D.: Mara - Sensor Based Augmented Reality System for Mobile Imaging. In *Proceedings of the 5th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '06)* (September 2006). 2, 3
- [KTO04] KAMEDA Y., TAKEMASA T., OHTA Y.: Outdoor See-Through Vision Utilizing Surveillance Cameras. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)* (Nov. 2004), pp. 151–160. 3
- [Low04] LOWE D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (November 2004), 91–110. 3
- [MLB04] MOHRING M., LESSIG C., BIMBER O.: Video See-Through AR on Consumer Cell-Phones. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)* (November 2004), pp. 252–253. 3
- [SAYT01] SATOH K., ANABUKI M., YAMAMOTO H., TAMURA H.: A Hybrid Registration Method for Outdoor Augmented Reality. In *Proceedings of the IEEE/ACM International Symposium on Augmented Reality (ISAR'01)* (2001), IEEE Computer Society, pp. 67–76. 3
- [Sor85] SORENSON W.: *Kalman Filtering*. IEEE Computer Society, 1985. 6
- [SRC02] SERON F., RODRIGUEZ R., CEREZO E.: Adding support for high-level skeletal animation. *IEEE Transactions on Visualization and Computer Graphics* 8, 4 (2002), 360–372. 6
- [SS01] SHAPIRO L., STOCKMAN G.: *Computer Vision*. Prentice Hall, 2001. 3
- [SW07] SCHMALSTIEG D., WAGNER D.: Experiences with Handheld Augmented Reality. In *Proceedings of the 6th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)* (November 2007), pp. 1–13. 2, 4, 7
- [TKY03] TENMOKU R., KANBARA M., YOKOYA N.: A Wearable Augmented Reality System Using Positioning Infrastructures and a Pedometer. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC)* (2003), IEEE Computer Society, p. 110. 3
- [VVMTO2] V. VLAHAKIS N. IOANNIDIS J. K., M. TSOTROS I. S.: Archeoguide: An Augmented Reality Guide for Archaeological Sites. *IEEE Computer Graphics and Applications* 22 (September 2002), 52–60. 1
- [WLS08] WAGNER D., LANGLLOTZ T., SCHMALSTIEG D.: Robust and unobtrusive marker tracking on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'08)* (September 2008), pp. 121–124. 2, 3, 4, 7
- [YNA99] YOU S., NEUMANN U., AZUMA R.: Orientation Tracking for Outdoor Augmented Reality Registration. *IEEE Computer Graphics and Applications* 19 (1999), 36–42. 2, 3
- [ZK05] ZHANG W., KOSECKA J.: Localization Based on Building Recognition. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 3, IEEE Computer Society, p. 21. 3