

Hybrid Approach Using Sensors, GPS and Vision Based Tracking to Improve the Registration in Mobile Augmented Reality Applications

Carlos Santos, Tiago Araújo, Jefferson Morais and Bianchi Meiguins

Computer Science Post-Graduate Program, Universidade Federal do Pará, Brazil
{carlosresque, tiagoaraujo, jmorais, bianchi}@ufpa.br

Abstract

One of the main challenges to creating rich, seamless, and adaptive Augmented Reality (AR) browsers is the accurate registration of the virtual contents in the real world. Usually, the AR browsers offer augmented navigation functionality by GPS and sensors, such as magnetometer and accelerometer. However, the position of virtual markers suffers some errors when the user is near to the desired location, due to many factors such as sensors failures and bad internet connections, among others. Additionally, to identify the correct marker, when there are many, is a challenging task to users. Therefore, to mitigate these problems, this paper proposes a hybrid approach of location and vision based tracking for AR applications, since the image recognition can be very helpful to identify near locations, avoiding misplaced markers and at the same time giving emphasis to that marker. Furthermore, to avoid bottlenecks in the AR browser applications the combination of the quality of vision-based tracking and the speed of the sensors is proposed. The designed system gets the information about the Points Of Interests (POIs), recommend places to explore around the user via GPS and sensors (as already done by current AR browsers) and run the recognition process only for the nearest POI to improve its registration. Aiming to choose the best recognition algorithm for this scenario, precision and time tests are performed using three algorithms (ORB, BRISK, and AKAZE) to detect keypoints and compute their features, and two algorithms (RANSAC and LMEDS) to estimate camera pose. The test pointed that the combination of AKAZE and RANSAC has the best accuracy, but an impractical time to use in real time application. Hence, the usage of vision techniques in an interval of time (skipping some frames) and the usage of inertial sensors movements to update the skipped frames are proposed, in order to use this solution on a mobile platform. Finally, the system solution was implemented in a tourism mobile AR application and some results are presented.

Keywords: Mobile Augmented Reality, Registration, Location Based Application, Vision Based Application

1. Introduction

Augmented Reality (AR) is being widely used in applications such as games, architecture, military training and professional qualification in several areas [1-2]. Besides, the evolution and popularization of mobile devices generate opportunities to bring this technology to the daily life of people.

Langlotz *et al.* [3] discuss the challenges to developing the next generation of AR browsers with rich, seamless, and adaptive content. And one of the main challenges is accurate and global registration.

Registration is the process to integrate the virtual content in the right place of the real scene (captured by the camera device), after that, this new integrated augmented scene is showed for the user on device's screen in real time.

AR browsers that rely only on location and sensors generally have inaccurate registration (mainly for short distances). Moreover, according to studies presented by Mulloni *et al.* [4] the tracking technology directly impacts the user experience and adoption. Hence, the correct and intuitive registration is of great importance.

Despite the majority of AR browser applications use GPS (Global Positioning System) and sensors for tracking, this approach is not accurate for short distances, causing a slight unalignment between the virtual marker and the building facade, as shown in Figure 1. Additionally, it is not widespread AR browsers applications offer image recognition tracking to mitigate the problem. Furthermore, they have problems when showing too many locations in the same direction, losing the emphasis on the location seeing by the user in that moment, making the task of finding the correct marker harder.

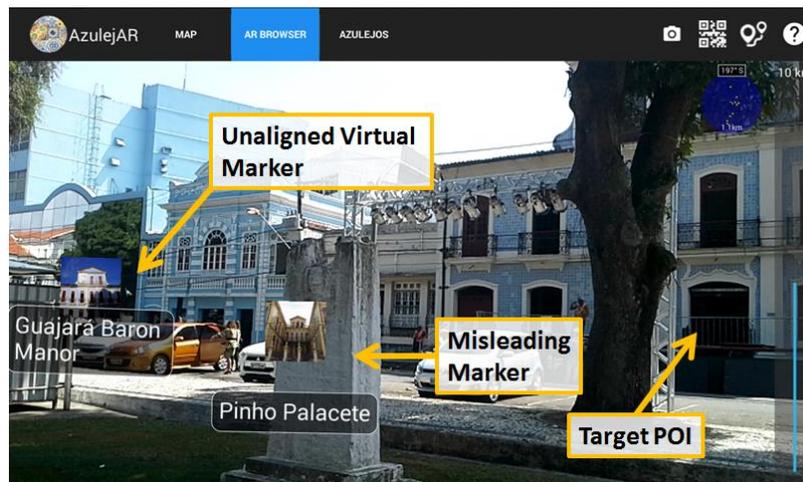


Figure 1. This Image Shows a Failure in Virtual Markers Position Based Only in GPS and Sensors

In order to improve the user experience in a mobile AR application, this paper proposes a hybrid form to identify Points of Interest (POIs) by combining image recognition with sensors and GPS, to overcome the obstacle of inaccurate positioning of POIs near the user. At the same time, the application can give a better highlight in the recognized POI and indicate more intuitively the real location of the POI to the user.

Aiming to select the best approaches for image recognition, tests are performed with some algorithms of image recognition present in the OPENCV library [5]. Since the majority of vision based AR applications rely on the recognition of planar surfaces, the proposed use case is based on this type of recognition. Therefore, the algorithms used in this work have the following workflow: keypoints detection in two images (one from camera's device and other from image dataset); features calculation of these keypoints; matching between the keypoints of both images using the calculated features and a metric of distance; and estimation of camera position based on the matches found, as showed in Figure 2.

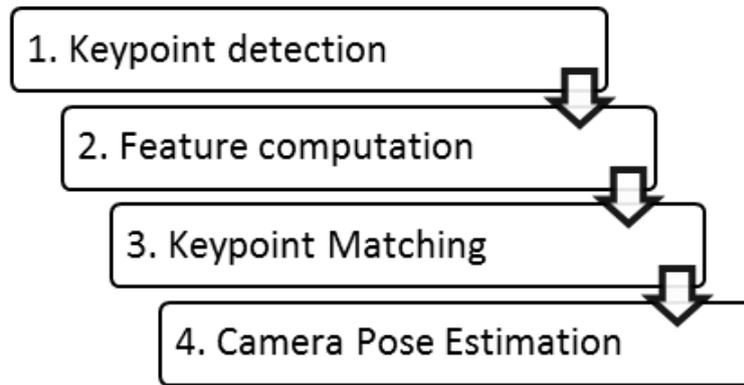


Figure 2. The Summary of Image Recognition Workflow

The tests for choosing the best algorithms were performed based on its processing time and precision. The purpose is to find a good balance between precision and processing time since it will run on mobile devices, which can have different setups and not always a high processing power.

The implemented application has the tourism scenario, aiming to facilitate the access and discovery of historical places. The user's experience can be improved by using the recognition of buildings' façade combined with tradition location based AR browser. For example, Figure 3 presents a use case of what can be done using facade recognition in a mobile application. In this case, the facade of an abandoned historical building (with restoration process planned) (a) was recognized using a reference image (b), hence it was possible to display how this building will look after its restoration (c).

The following sections of this paper are organized in four sections. Section 2 gives a brief description of the theory applied in this work, focusing on the algorithms and techniques used along with this work. Section 3 presents the preparation and organization of the performance tests done in the OpenCV's algorithms. Section 4 discusses the obtained preliminary results. Section 5 describes the proposed approach and lastly, Section 6 concludes the work and draws future works.

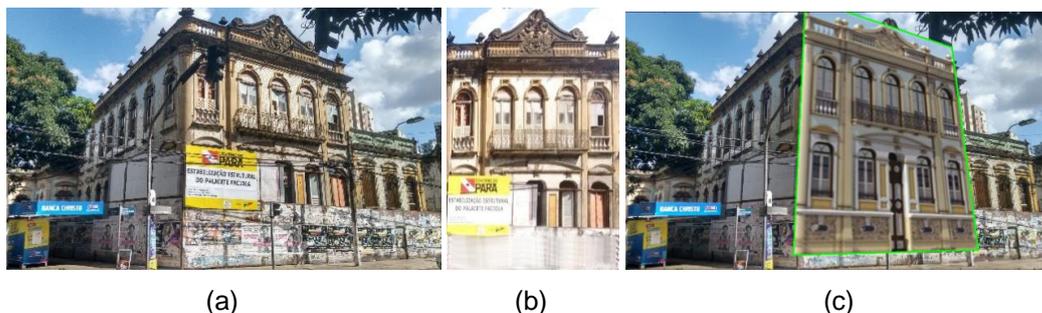


Figure 3. Illustrative Resume of a Feature for a Mobile AR Application Geared Towards Tourism. The Image (a) Shows a Picture of an Abandoned Historical Building. The Image (b) Presents the Reference Image Used by the Recognition Algorithms. After Recognizing the Façade and Extracted the Positioning Parameters the Image (c) Shows the Reconstructed Building Façade (Image Overlays the Scene Image)

2. Background

This section describes the theories related to this work. Also, the concepts and algorithms used in this article are presented.

2.1. Mobile Augmented Reality

Augmented Reality (AR) is a technique used to augment the human senses, be it improving or completing the real world with digital content, using computers [6]. Azuma *et al.* [7] define AR as a system that combines virtual elements with the real scene, it is interactive and in real time, and registers or aligns real and virtual objects in a single scene, giving the impression that the virtual objects are, in fact, part of the real scene.

A common scenario for AR applications is presented in Figure 4. The tracking system extracts information from the real world image (or from sensors), making it possible to align the virtual world camera with the real world camera. Thus, the graphic system can draw (rasterize) virtual objects with the same perspective of the real world. The phase of the registry is the moment that the real world information is combined with virtual information. After that, the resulting combination is presented to the user.

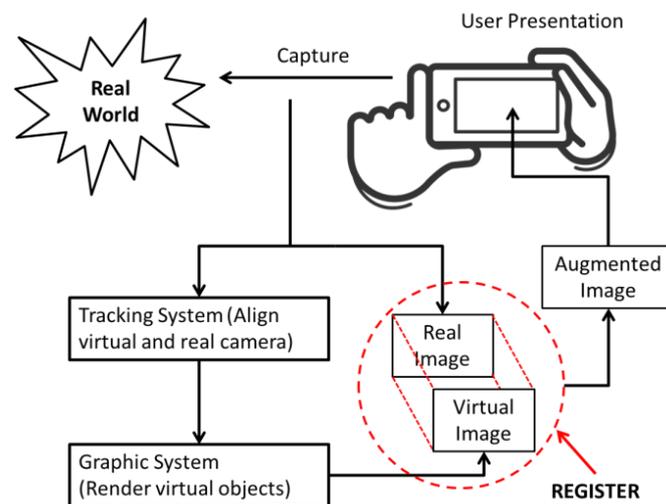


Figure 4. A Typical Scenario of Mobile AR Applications, Adapted from [6]

Currently, the mobile AR application used in this work has a tracking system that works by reading sensors, such as GPS, magnetometer, and accelerometer. The aim is to add the image recognition (historical buildings' facades) in this system.

Tests are performed to decide which algorithms are best suited to each step of the system. Algorithms of detection and matching of keypoints between a reference image and the scene image are tested, as well as algorithms that calculate the homography matrix from the matched keypoints. The homography matrix is the parameters of the real camera, thereby it allows to calculate the position of virtual elements in the real scene.

2.2. Keypoints Detection and Description

Some algorithms can be used to find keypoints on two images (in general, query and reference images), such that it is possible to find a set of matching keypoints pairs between the two images, thus performing the recognition of the reference image in the query image.

There are an extensive number of algorithms that performs the feature detection and compute an array of features for those keypoints (this array of features is generally named as the description). However, this significant amount of algorithms makes difficult the

task of finding the best one for each situation. Thereby, in this article, a test with three algorithms that is already implemented in the OpenCV is performed.

The algorithms are ORB [8], AKAZE [9], and BRISK [10]. The three algorithms use binary descriptors to perform the matching of keypoints. The option for binary descriptors is taken from the fact that the calculation of features and their correspondences are faster with binary descriptors, thus more feasible for use in mobile augmented reality applications.

ORB is a combination of two other techniques: FAST [11] a fast keypoints detector and the BRIEF [12] for computing descriptors of keypoints in the image through simple binary tests. ORB performs tests to make BRIEF invariant to rotation. Rublee *et al.* [8] have presented comparisons of ORB with other known techniques (such as SIFT [13] and SURF [14]) and as a result, ORB is twice faster than SIFT, as SIFT demand high computational cost, especially in real-time systems. It is also an order of magnitude faster than the SURF. SIFT and SURF are the two classical algorithms used for keypoint detection and feature computing, but both demand high computational requirements to run in real time.

BRISK is another technique based on FAST for the detection and makes a rotation and scale invariant description of keypoints in images. The algorithm uses a handcrafted pattern for sampling neighbors of the keypoints and makes binary comparisons to generate the keypoint description. It is also an order of magnitude faster than the SURF and came from the same need to reduce the computational time of SIFT and SURF, but trying to keep the same level of precision.

The AKAZE algorithm, as the previously mentioned algorithms, has faster computing than the established SURF and SIFT and is also faster than its predecessor KAZE [15], and maintains a good precision in variations of viewpoint and noise in the image. Using the FED – Fast Explicit Diffusion – it improves the detection characteristics of a nonlinear scale space.

2.3. Homography Transformation

The homography transformation is a type of geometric transformation that describes the movement of a plan projected by a fixed camera, or similarly, can describe the camera movement if the plan is fixed. The homography can be used to calculate the projected movement of a 3D plan, in such way that it can describe the movement of each projected point that belongs to the plan with only one 2D homogeneous matrix [16].

This homography matrix is generally named \mathbf{H} , and to know the position of each projected point that belongs to a plan after an arbitrary movement, the matrix \mathbf{H} have to be multiplied with the matrix of points \mathbf{P} (2D and projected) that belongs to the arbitrary plan. The result of this matrix multiplication is the set of points \mathbf{P}' that represent each 2D projected point after the arbitrary movement.

The matrix \mathbf{H} has eight degrees of freedom which represents the numbers of parameters which can vary independently, giving to this transformation the possibility of performing perspective transformations.

In image recognition tasks, one wants to know what is the movement of the plan given two images, or generally what is the camera pose given that the plan is fixed. In this case, the homography matrix has to be calculated given a set of pairs of points that belongs to the plan in the two images. The calculation of the homography matrix would be easy if the matches were always correct, but this usually does not occur in real situations. Therefore, one has to use robust methods to calculate this matrix.

Currently, the OpenCV library has two robust methods to calculate the homography matrix given a set of matches between two images. The methods are RANSAC [17] and Least Medians (LMEDS) [18]. Usually, the reference image is well aligned (facing the camera) and has no occlusions, and the task is to find what the movement this reference image has performed to achieve the position observed in a query image.

RANSAC [17] is a robust algorithm that can estimate a model (in this case, the camera pose) based on data (in this case, a set of matched keypoints) with a high degree of precision, even if the data contains outliers. This algorithm randomly selects a minimum set of correspondences and checks the number of matches that agrees (the consensus). The random sample and the consensus check are repeated a given number of times and after RANSAC selects the model with bigger consensus. It requires two parameters to be configured: a distance threshold to classify if a match belongs or not to the consensus of a sample and the number of iterations the algorithm should run. Depending on the number of iterations and the proportion of wrong matches the final solution may not be the optimal solution.

LMEDS [18] is an algorithm that has emerged from the attempt to improve the existing classical least squares technique, which aims to minimize the errors discarding some outliers. It works well when there are less than 50% of outliers (wrong matches). One advantage of LMEDS is that users do not need to configure parameter, such as thresholds or number of iterations, and the downside is the disability to deal with situations where the wrong matches account more than a half of total matches [16].

3. Preparation and Organization of Tests

This section presents the setup and workflow of the preliminary test.

3.1. Offline Test Environment

The laptop used in tests was an Inspiron 14' 2640, Intel Core i5, 6GB DDR3 RAM with Ubuntu 14.04.4 LTS OS. The language used was Python 2.7.6, since its syntax is high level and easy to understand, implying in simple comprehension and execution of generated scripts, easing the tests elaboration.

The libraries used were Matplotlib 1.5.1, Scipy 0.13.3, NumPy 1.11.0 and OpenCV 3.1.0. These libraries are used to run the algorithms of computational vision related to the tests.

3.2. Obtaining the Images Dataset

To obtain the images used in the experiments, the following mobile devices were used: MOTO G second-generation, Android version 6.0, rear camera of eight (8) megapixels, aperture of f2.0 and maximum resolution of 3264x2448 pixels; MOTO E second-generation, Android version 5.1, rear camera of five (5) megapixels, maximum resolution of 2592x1944 pixels for images; and a Samsung Galaxy Tab third-generation, Android version 4.4.2, rear camera of 3.15 megapixels with maximum resolution of 2048x1536 pixels. The last one device was used to run AR application with image recognition algorithms.

The images obtained for the experiments are called scene images, due to their purpose for use in mobile AR applications. These images have facades of historical buildings in the city of Belém of Pará (Brazil), and for tests, the following POIs were chosen: Sé's Church, Faciola's Palace and IHGP (Historical and Geographical Institute of Pará). The pictures were captured in the morning, afternoon and late afternoon.

The test used 124 images of three facades under varied lighting conditions (very sunny, normal lighting and dim lighting) and different capture positions, aiming to measure the processing time and precision of the algorithms present on OpenCV.

3.3. Test Workflow

Figure 5 shows an overview of the strategy approached in this paper to perform the overlay of images in the place of the historical buildings' facades. Since this study is geared towards use in a mobile AR system, the building which has to be detected in the image is chosen based on the user's geographic position and in the direction where the mobile device's camera is being pointed to. The available sensors in mobile devices, such as GPS, magnetometer, accelerometer and gyroscope, are used for this purpose.

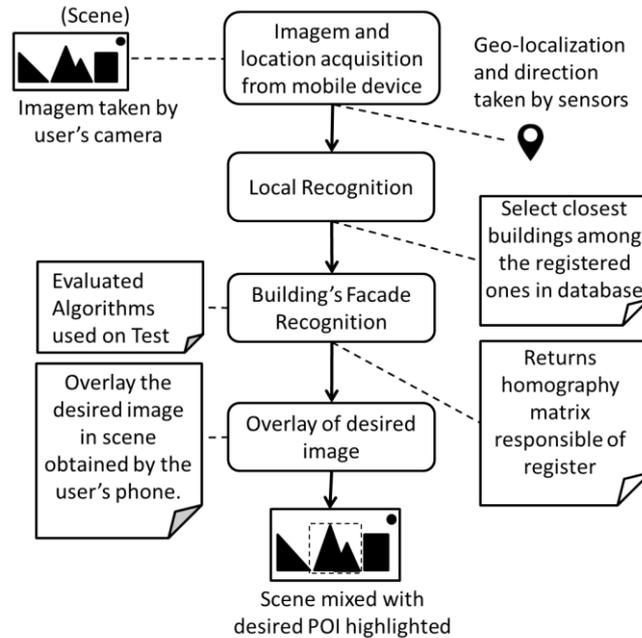


Figure 5. The Flow of the Proposed Mobile AR Application (Based on Location) to Overlay the Scene Image with Augmented Information Using the Facade Detection as the Tracking System

The location recognition is performed starting from the user position (latitude and longitude), verifying the direction where the user is pointing the camera (using position and movement sensors), and choosing the nearest location in the direction of the user's camera. The locations must be previously registered with their respective geographic coordinates.

After the system identifies which building must be found in the image, the step of recognition starts, using the image captured by the mobile device's camera (scene image).

To achieve this, algorithms of detection and matching of keypoints were used. In this paper, three algorithms with binary descriptors were evaluated. They are ORB, BRISK, and AKAZE. A brute force algorithm performs the correspondence of keypoints, using the Hamming distance [19] among the binary descriptors.

After finding a group of pairs of keypoints, two algorithms were tested to calculate the homography matrix: RANSAC and LMEDS.

Figure 6 presents the sequence of operations to perform the tests. First, the scene and reference images are scaled considering their larger size (width or height) to predefinition sizes (600, 800, 1000 and 1200), the other side is calculated proportionally, preserving the aspect ratio. After that, the algorithms to detect the reference image in the scene and to obtain the homography matrix are applied. The traced arrows point to the variations done for comparison in tests.

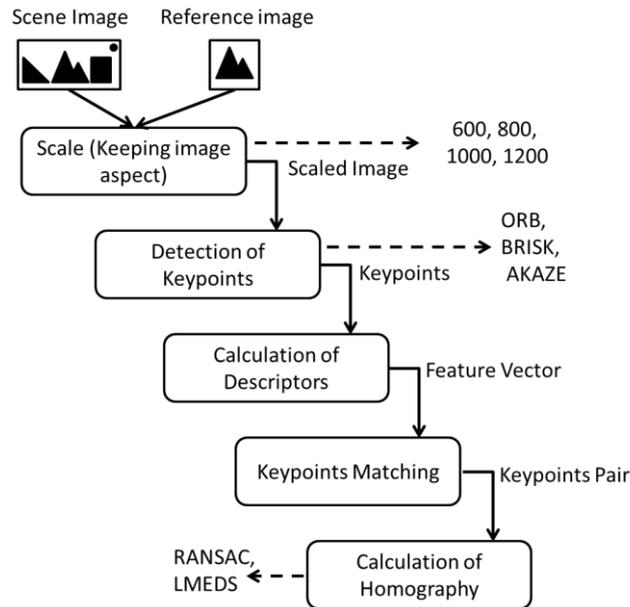


Figure 6. The Sequence of Operations Performed in the Facade Detection. the Traced Arrows Point to the Values Evaluated in Tests

Tests were performed measuring the time and precision (measured according to the register result) of each technique. Figure 7 presents an example of failure in the calculation of homography matrix, and consequently failure in register on the scene.

This first test is off-line and static using pictures of building facades with various positions and lighting conditions.

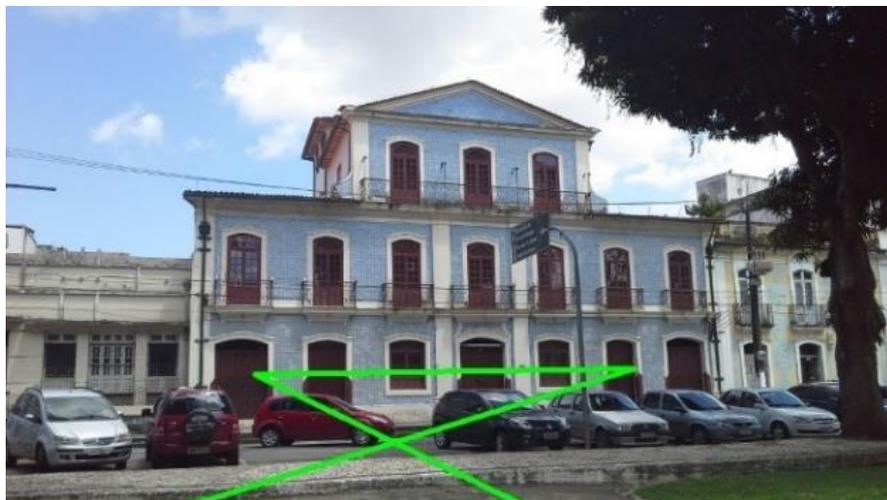


Figure 7. An Example of Failure in the Process of Obtaining Parameters from the User's Camera

4. Preliminary Test Results

This section presents the results of tests based on the images sizes (600, 800, 1000 and 1200), on the algorithms of keypoints detection (ORB, BRISK, and AKAZE) and on the homography calculation (RANSAC and LMEDS).

Figure 8 displays the comparison graphs for time and precision between ORB, BRISK and AKAZE algorithms, using RANSAC or LMEDS and varying the image size. It is worth mentioning that the image was scaled during the algorithm execution and the scale time was considered in tests. The images have an average size of 2500 pixels of width by 1400 pixels of height before the scale operation.

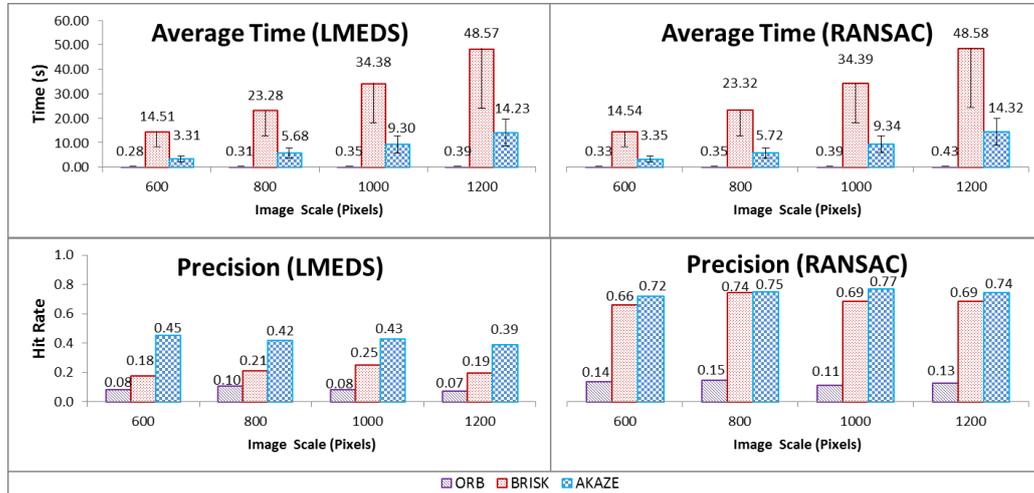


Figure 8. Charts of Time and Precision Tests Showing all the Values and Algorithms Used

The AKAZE algorithm combined with RANSAC had an average precision better than BRISK and ORB. ORB had an execution time lower than the other algorithms, however, its precision was significantly lower. BRISK had a similar precision to AKAZE, but its execution time was much higher.

In this way, the combination of AKAZE and RANSAC algorithms had the best cost/benefit relation considering time and precision. Although, the time of the AKAZE algorithm is impractical to use frame by frame in a mobile AR application which depends on the real-time characteristic.

5. Proposed Approach

This section presents the proposed approach working on the mobile AR application which has the tourism context. The application's name is AzulejAR.

The purpose of AzulejAR is to invite users to explore POIs about azulejos (historical tiles present in buildings' facades or interior) and to know more about its history and technical details. It is possible to find these collections of azulejos in churches, palaces, and particular houses.

The application will assist the public as a guide with a virtual circuit in the city, using GPS and inertial sensors, enabling the user to discover where the POIs are. Once the POI is located the application shows information about the POI.

In this article was suggested that, when the user arrives near to the POI location, the localization system is not entirely accurate, hence the application use the already known location and direction of the nearest POIs to run image recognition algorithms and display virtual markers.

Some problems can be resolved with the image recognition, for example, highlight just the facade of the building that the user is pointing at (Figure 9). Figure 9 (a) shows the previous version of AzulejAR pointing to the Sé's Church. However, two markers are displayed that could confuse the users.

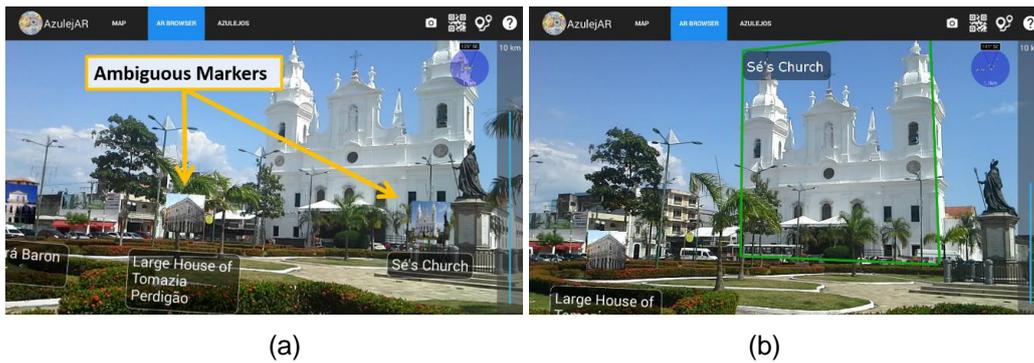


Figure 9. Markers Close to Each Other can confuse the User (a), while the POI is Clearly Highlighted in the Image Recognition Approach (b)

Figure 10 points another problem that can occur when the user is very near to the desired location. In Figure 10 (a) the previous version of AzulejAR shows the wrong place to the user because of GPS's precision failure. In this case, the user is very close to the place and the application considers that the user has passed the place and the desired POI is in the backward direction. Figure 10 (b) shows the problem solved with the image recognition functionality.

In that manner, the system gives more emphasis on the POI that is near to the user and at the same time the user has more space to interact with that POI. The user can touch in the entire area of the building's facade to obtain more information about that POI. Figure 11 shows the content displayed after the user interacts with the POI by touching in the marker.

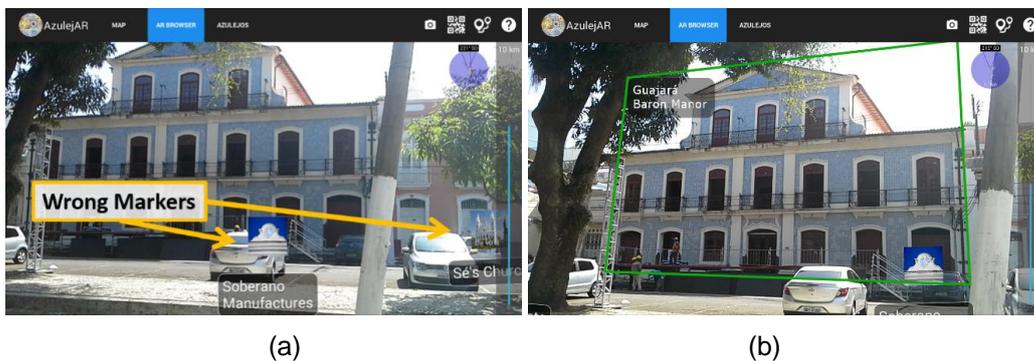


Figure 10. The Accuracy of GPS Causes Wrong Positions of Virtual Markers (a). This Problem is mitigated with Image Recognition Functionality (B)

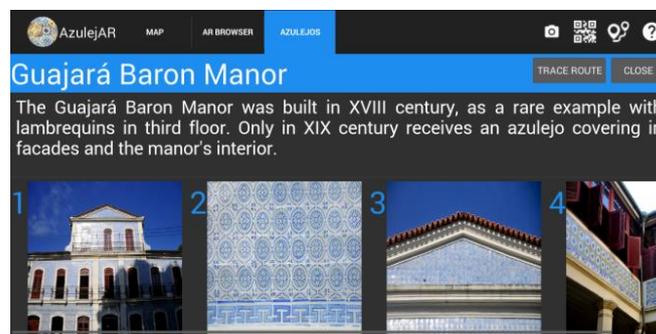


Figure 11. After the User Touches in the Virtual Marker, the Content of the POI is Shown

Although the problem has been solved for intuitiveness problems, the overload of these algorithms is still a problem, once the application has other functionalities running in parallel. The frames per seconds' ratio decreases drastically using this solution frame by frame. Thus, the proposed approach is to use the image recognition method skipping some frames and completing the movement of the virtual marker with movements captured by inertial sensors, as it is already used for the POIs that are not near to the user.

Algorithm 1 presents in summarized way the procedure of jumping frames and synchronizes the jumped frames with sensor movements.

Attempting to figure out the impact of skipping frames it was measured the average time of the entire process of registering the virtual world in the real one, varying the number of skipped frames. The smartphone used to measure the time is the first smartphone (MOTO G) described in Section 3.2.

Figure 12 presents the average time of registration in relation to the number of skipped frames. Analyzing the chart it can be concluded that 10 skipped frames is a good trade-off because it greatly decreases the average time and at the same time maintains the quality of visual recognition for each 10 frames. Furthermore, it does not have an expressive decrease in time when skipping more than 10 frames, while the system takes longer to use the vision tracking, decreasing the quality of tracking.

Algorithm 1: Pseudo-code of the skipping frames logic.

```
int cont;
Matrix camera_pose;
PointOfInterest poi;
int n_skip = 10; //can be defined by the app's FPS

CameraCaptureLoop (frame){
    if(cont > int n_skip){
        poi = getNearestPOI();
        camera_pose = runImageRecognition(poi);
        cont = 0;
    }else{
        updatePositionWithSensors(camera_pose);
        cont++;
    }
}
```

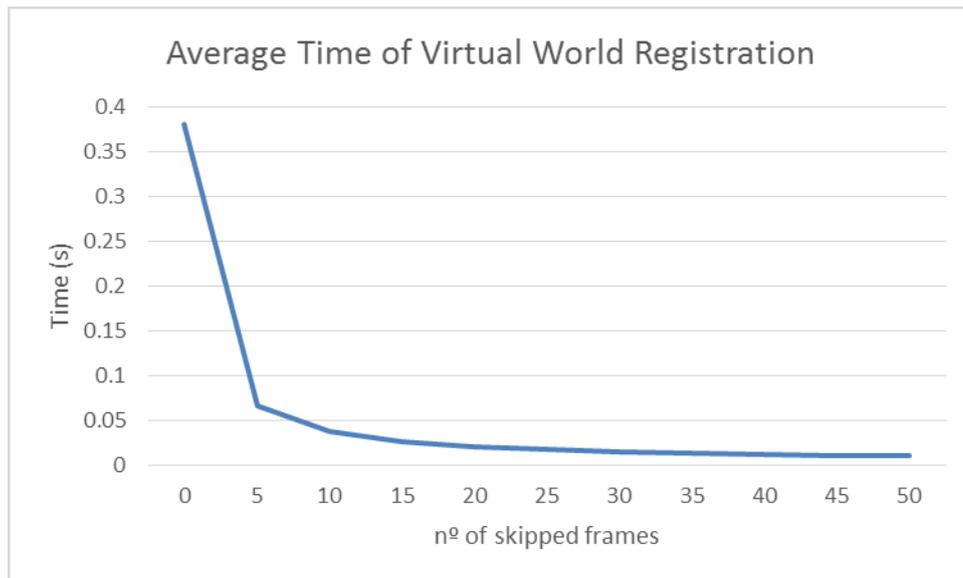


Figure 12. Average Time of the Register Process According to the Number of Skipped Frames (Updated by Sensors Instead of Image Recognition)

6. Conclusion

This paper has proposed of combining GPS and inertial sensors with image recognition for mobile AR browser application, in such manner the virtual markers of outdoor applications could be drawn in more intuitive and perceptive way. Furthermore, this approach can prevent slight GPS fails causing misplaced markers that can compromise the user experience.

Some offline tests have been performed with image recognition algorithms to study the cost/benefit relation between processing time and precision. Thanks to the presented study case, the tests have been performed for detection of historic buildings facades. The evaluated algorithms are ORB, BRISK, and AKAZE for detection and description of keypoints and RANSAC and LMEDS for estimating camera pose. The tests aimed to choose a set of techniques and parameters for use in a mobile augmented reality application.

The tests results show that combining AKAZE technique with RANSAC had a better cost/benefit gain in relation to time and recognition precision, and using an image with a smaller scale (between 800 and 600 pixels for the bigger image side) does not significantly compromise the precision of the recognition process while decreasing the overall processing time. Although the AKAZE algorithm has a better cost/benefit, its time is still unpractical to a real-time mobile application using it in every frame of camera stream. Hence, the present article proposes to use it in a reduced number of frames, skipping some of them. Moreover, to avoid lagged virtual markers, the markers positions are updated with movements captured by sensors when the image recognition is not running.

The summarized contributions are the following:

- The combination of GPS and inertial sensors with image recognition features for mobile AR browsers.
- The suggestion of the best combination of image recognition algorithms based on the preliminary test.
- A more intuitive way of showing virtual markers to the users.
- The strategy of skipping frames of image recognition and updating them with sensors movements, avoiding system bottlenecks.

As future works, ways to improve the skipping frames strategy will be analyzed, for example using the ORB algorithm (that has better time performance, according to presented test) to image recognition using the inertial sensors to improve its accuracy. Moreover, usability tests will be performed to verify the user experience in mobile AR application using the proposed hybrid approach. Finally, a comparative analysis of the frames per second rate and the quality of marker position will be performed to select the best number of frames to be skipped.

References

- [1] C. D. Scott and R. E. Smalley, "Diagnostic Ultrasound: Principles and Instruments", *Journal of Nanosci. Nanotechnology.*, vol. 3, no. 2, (2003), pp. 75-80.
- [2] D. Krevelen, R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations", *The International Journal of Visual Reality.*, vol. 9, no. 2, (2010), pp. 1-20.
- [3] M. Billinghurst, A. Clark, G. Lee, "A Survey of Augmented Reality. Foundations and Trends in Human-Computer Interaction", Now Publishers Inc. Hanover, MA, USA, vol. 8, no. 2-3, (2015), pp. 73-272.
- [4] T. Langlotz, T. Nguyen, D. Schmalstieg, R. Grasset, "Next-Generation Augmented Reality Browsers: Rich, Seamless, and Adaptive", *Proceedings of the IEEE*, vol. 102, no. 2, (2014), pp. 155-169.
- [5] R. Mulloni *et al.* "Experiences with the Impact of Tracking Technology in Mobile Augmented Reality Evaluations". *Proceedings of the MobileHCI 2012 Workshop MobiVis*, (2012).
- [6] G. Bradski. "The opencv library". In *Doctor Dobbs Journal*, (2000), pp 120-126.
- [7] O. Bimber, R. Raskar. "Spatial Augmented Reality: Merging Real and Virtual Worlds". Taylor & Francis Group, (2005).
- [8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. Macintyre, "Recent Advances in Augmented Reality", *Computer Graphics and Applications*, vol. 21, no. 6, (2001), pp. 34-47.
- [9] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT and SURF", *Proceedings of the International Conference on Computer Vision (ICCV)*, IEEE, (2011), pp. 2564-2571.
- [10] P. Alcantarilla, J. Nuevo, A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces". *Proceedings of the British Machine Vision Conference (BMVC)*, (2013).
- [11] S. Leutenegger, M. Chli, R. Siegwart, 2011. "BRISK: Binary robust invariant scalable keypoints", *Proceedings of the Conference on Computer Vision (ICCV)*, IEEE, (2011).
- [12] E. Rosten, R. Porter, T. Drummond, "Faster and better: a machine learning approach to corner detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32 no. 1, (2010), pp. 105-119.
- [13] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, "BRIEF: Computing a Local Binary Descriptor Very Fast". *Pattern Analysis and Machine Intelligence*. vol. 34, no. 7, (2012), pp. 1281-1298.
- [14] D. Lowe, "Object Recognition from Local Scale-Invariant Features", *Proceedings of the International Conference on Computer Vision*, (1999), pp. 1-8.
- [15] H. Bay, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features". *Proceedings of the European Conference on Computer Vision*, (2006).
- [16] P. Alcantarilla, A. Bartoli, A. Davison, "KAZE Features". *Proceedings of the European Conference on Computer Vision*, (2012).
- [17] E. Marchand, H. Uchiyama, F. Spindler, "Pose estimation for augmented reality: a hands-on survey", *IEEE Transactions on Visualization and Computer Graphics*. vol. PP, no. 99, (2015).
- [18] S. Choi, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Communications of the ACM*, vol. 24, no. 6, (1981), pp. 381-395.
- [19] P. Rousseeuw, "Least Median of Squares Regression". *Journal of the American Statistical Association*, vol 79, no. 388, (1984), pp. 871-880.
- [20] R. Hamming, "Error Detecting and Error Correcting Codes". *Bell System Technical Journal*, vol. 29, no. 2, (1950), pp. 147-160.

