

Analyzing AR Viewing Experience through Analytics Heat Maps for Augmented Content

Marcel Santos*, João Marcelo Teixeira†, Lucas Figueiredo‡, Veronica Teichrieb§, Cristiano Araújo¶
Voxar Labs, Computer Science Center, UFPE - Brazil

ABSTRACT

Analytics is a well-known form of capturing information about the user behavior of an application. Augmented reality applications deal with specific data such as the camera pose, not being supported by popular analytics frameworks. To fill such gap, this work proposes an analytics framework solution for augmented reality applications. It supports both marker-based and markerless augmented reality scenarios, collecting data related to camera pose and time spent by the user on each position. Besides the multiplatform capture tool, the framework provides a data analysis visualization tool capable of highlighting the most visited 3D positions, users main areas of interest over the marker plane, the 3D path performed by the camera and also a recovery of the content viewed by the user based on the collected camera pose information.

Index Terms: Augmented reality analytics; Framework; User behavior Analysis; User Experience; Augmented Content; Heat maps; Data Analysis

1 INTRODUCTION

According to Azuma [2], Augmented Reality (AR) “allows the user to see the real world, with virtual objects superimposed upon or composited with the real world”. Therefore, AR technologies can enhance the visual aspect of the real world, by providing additional contextualized information.

Because it can be applied to many application fields like entertainment, gaming, design, marketing, sales, and navigation, to name only a few, augmented reality is meant to become the technology of the future. And, in fact, in the last years, it has become more popular thanks to new companies investing in this field [8]. Microsoft, for example, is investing so much in its AR platform, the Microsoft HoloLens, Apple recently bought the Metaio company and Google, together with other companies, invested last year \$ 794 million in a startup, the MagicLeap. Moreover, the new devices which deliver more natural interfaces and new approaches to overcome previous AR limitations are leading Augmented Reality field to a rapid maturation.

The importance of AR is better described in the Goldman Sachs full market report [15]. This report reveals that in a worst case scenario AR will become a \$ 80B market. It means that, in the future, many people will be exposed to AR applications all the time in many different contexts, each one with its requirements. The amount of data generated from these users can be applied to get insights about how they use the AR technologies, having the potential to deliver more valuable applications.

As AR comes to maturity in terms of technology, it still lags behind in terms of having tools to help developers evaluate its impact

in the applications they develop. As is evident nowadays, the use of Business Intelligence and Analytics is critical to the success of an application [5]. Data Analytics examines a significant amount of data to uncover hidden patterns. It gives a competitive advantage to organizations which apply these analyses when making decisions about their product. The vast amount of data, sometimes called Big Data, is a more comprehensive resource for elevating business strategy. Content companies, for example, use many of the generated data from user clicking and watching to get insights about how to retain more user on its pages. Therefore, there are several tools developed with the purpose of data analytics, like Google Analytics [11], Optimizely [14], Yahoo Analytics [17] and FireBase [9].

None of the previously cited tools can deal with the specific and significant amount of data coming from Augmented Reality applications where applications are not restricted to traditional mouse and keyboard user interactions. It involves tracking and visualization of three-dimensional movements. An example of user interactions that need to be captured by analytics is the user attention focus and the user position in space. That means the evaluation of user experience with AR has to be done with small-scale experiments as in [13] where only a few dozens of people were participating, while with the use of analytics data from millions of user interactions can be utilized.

That means it is important to collect and analyze the use of additional features, such as camera pose, which demands specific approaches in order to have a proper Analytics tool that is effective for AR applications.

In this work we propose a *framework to provide analytics for AR-based applications*. The framework facilitates the process of collecting and analyzing data about the user focus and position in space from each session of use in the application. By consulting the analytics it is possible to better understand the user behavior and gather insights which were not accessible before for this type of applications. The main contribution of the proposed framework are described as follows:

- a *Data Extractor Module*, which is triggered by the running application to gather data from the AR experience in real-time and store it for later analysis;
- a tool for 2D visual data analysis (within our *Data Analysis Tool*), showing a *two dimensional heat map* of the user attention focus, which is obtained by the projection of the center of the camera on the augmented content;
- a tool for three-dimensional visual data analysis (within our *Data Analysis Tool*), providing a *three dimensional heat map* of the user position in space about the tracked pattern and the augmented model.
- a tool for visualizing the *spatial path* each user performed on their experience with the AR application;
- a tool for visualizing the virtual content from the same view-point each user viewed, also respecting the timeline of their session of use.

*e-mail: mss8@cin.ufpe.br

†e-mail: jmxnt@cin.ufpe.br

‡e-mail: lsf@cin.ufpe.br

§e-mail: vt@cin.ufpe.br

¶e-mail: cca2@cin.ufpe.br

1.1 Use Possibilities

There are some use possibilities for this AR analytics framework.

- We know the target point of interest by the user while using the 2D marker visualization feature from *Data Analysis Tool*. The view target of the user is important to understand to what parts of the template the user is paying more attention.
- Also, we can figure out parts of the virtual model that are more relevant to the user. Thus, if we want to put an advertisement in an AR application, for example, it is possible, using the proposed framework, to know if a specific place is effective to position an advertisement information because this place is a highly visited one by the user.
- The three-dimensional viewer gives information about how the user is interacting with the AR application. Thus, it gives ergonomics information and so ways to improve the user interaction experience with the application.
- This two information combined, plus the total time spent in the applications can deliver useful strategic information about user behavior, how to improve application quality and how to increase profits with this application by advertising. Such insights will be priceless when AR interfaces become widely used.

Figure 1 covers some examples of potential applications that can benefit from the proposed augmented reality analytics framework. In the 3D scenario, left image from figure 1, the user will surround the 3D object to have a complete visualization about the scenario and in the 2D, right image from the same figure, the user will focus on specific parts of the model, for example, the wheel of the car.



Figure 1: The left image [7] shows three dimensional information regarding user's 3D point of view gives useful insights about how to improve application user experience. Moreover, two dimensional information is very useful to figure out parts of model which are catching the user attention as showed in right image [10].

1.2 Structure

The article is organized as follows. Section 2 describes several general purpose analytics frameworks applicable to several kinds of applications, not necessarily AR ones. It also describes the works that try to analyze user behavior or get insights from users in AR applications. Section 3 describes the visual concepts of our analytics framework. Section 4 explains how we built our framework, its architecture, its features (data capture and data analysis) and how we validated the framework correctness and applicability in a real world context. Finally, the last section discusses the obtained results and our future approaches for improving the proposed framework.

2 RELATED WORK

We looked for several works related to analytics and user behavior analysis in a wide range of contexts (web, augmented reality and virtual reality applications) to understand how these tools work, and their positive and negative aspects. Major findings are presented in sequence.

2.1 Analytics on Web Content

We have found a lot of frameworks able to perform analytics on web content, such as Google Analytics [11], Optimizely [14], Flurry Analytics from Yahoo [17] and Firebase [9]. All of them are web based. Moreover, these frameworks have good data analysis tools with several charts and heat maps. However, they were developed thinking in usual kinds of applications like web content and so do not fit well to AR analytics. They do not, for example, provide ways to visualize three-dimensional data. These frameworks can be used to take advantage of their network implementation and server integration but must implement the tools to analyze and visualize the AR data such as the camera pose that indicates where the user is positioned and looking at in each frame.

2.2 Analytics on Virtual and Augmented Content

In order to understand how user behavior and interaction analysis has been done in three dimensional based applications, such as augmented and virtual reality ones, we looked for articles and applications in these fields and their analytics tools. Based on obtained results, it is evident to us that there is a limited range of tools for this purpose and the existing ones were developed with very limited functionalities, just to analyze a specific application, and, thus, it is not possible to integrate them into other applications.

Craftar [6] is a commercial AR solution that has analytics features available, which are limited only to providing detection rates and the most detected patterns as showed in figure 2. Contrarily from our proposed framework, it does not provide detailed information regarding user interaction and behavior.

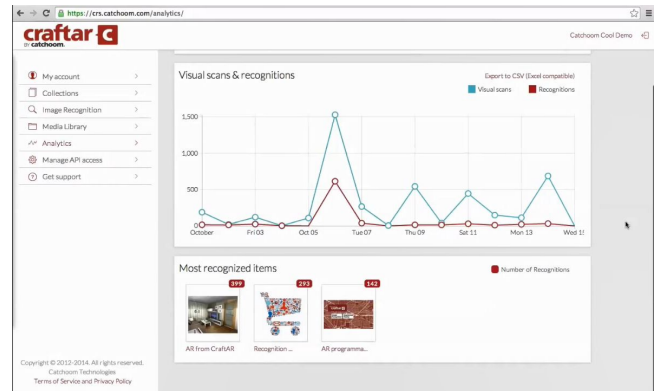


Figure 2: Analytics from a commercial AR solution. Only detection rates and most detected patterns are available.

In the game context, understanding the player behavior and making sense of gameplay actions is an important step of game design. G-player [4] is a tool which aims to deliver insights from players behavior. In order to do so, it uses several spatial visualizations, such as heat maps as showed in figure 3 and movement plots to get information from spatiotemporal data. This tool achieved good results in player behavior analysis. Moreover, despite the fact that it is being applied to games content, this work gave us helpful insights about how to use heat maps to evaluate a huge amount of temporal data.

Blippar [3] is a well known augmented reality mobile application and an advertising platform. This platform can transform everyday products into content-rich experiences using markerless image recognition approaches. Blippar, moreover, has a content creation platform which makes possible to create and manage AR resources. This platform is divided into front-end and back-end layers. The first one is related to customer experience. Through this layer, they will see three-dimensional models rendered on pre-inserted templates

using the Blippar dashboard. Moreover, back-end features include a data-metric dashboard to gauge how clients are engaging, as well as a heat map with granular data showing information like how many people are seeing the advertising, the user location and what they are clicking on. This back-end, however, does not show heat maps with information about how the user is interacting with the model, information which we believe is essential for content creation. Our inspiration comes from [4] that also uses heat maps to evaluate how the user is interacting with games to understand how to build better experience scenarios.

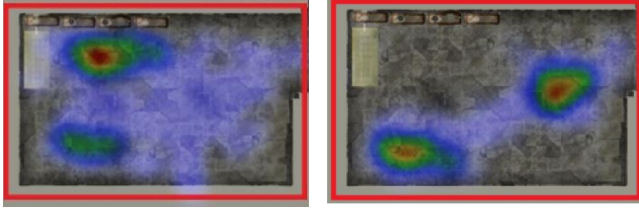


Figure 3: G-player heat maps [4].

3 VISUAL ANALYSIS CONCEPT

In order to define and illustrate a use of our framework, we draw the concept of our *Data Analysis Tool* considering a base application which uses a fiducial marker as target for tracking as shown in figure 4. We depict an application which augments the real world with both planar and three-dimensional objects.

As a sample application, consider that a promotional campaign from NASA uses an augmented reality application to provide an improved viewing experience of a 3D model of a telescope while displaying at the bottom the institution logo (figure 4). The user handles a tablet to observe the augmented content as shown on the top part of figure 4. Then we generate a 2D grid on the pattern (in this case a fiducial marker) plane.

The goal of the 2D visual data analysis is to provide a heat map of the regions within the pattern plane. This way, in order to understand the target region the user is using on each frame, we use the central image point and find its projection on the 2D grid as shown in the middle part of figure 4. As a result, by accumulating the time spent in each square of the 2D grid, it is possible to render the resultant 2D heat map as shown in the bottom part of figure 4.

The two-dimensional heat map is particularly useful for cases that provide planar augmentation such as the logo example shown in figure 4. By visualizing the heat map, it is possible to analyze which part of the logo is mostly focused, and understand if viewers are directing their attention as desired, on a publicity campaign for example. The 3D heat map, on the other side, may provide additional insight about the spatial behavior of users, describing how they positioned themselves in relation to the augmented content, and which angle of the 3D objects they looked at most. Figure 5 shows our concept for the 3D heat map.

The three-dimensional heat map also uses a discretization of the 3D environment, filling each voxel according to the time the user spent on that position. Note that the 3D grid has a larger base if compared to the virtual objects area. This occurs because that grid must comprise not only the virtual objects but also the space around them.

Figure 6 shows the concept of the path visualization, which is useful to understand the flow of the user experience. In this case, the user circled twice the tracked template. At last, the concept of the *tool for visualizing the content from user viewpoint* is inherently defined by the rendering of the 3D model and does not lack of a visual concept definition.

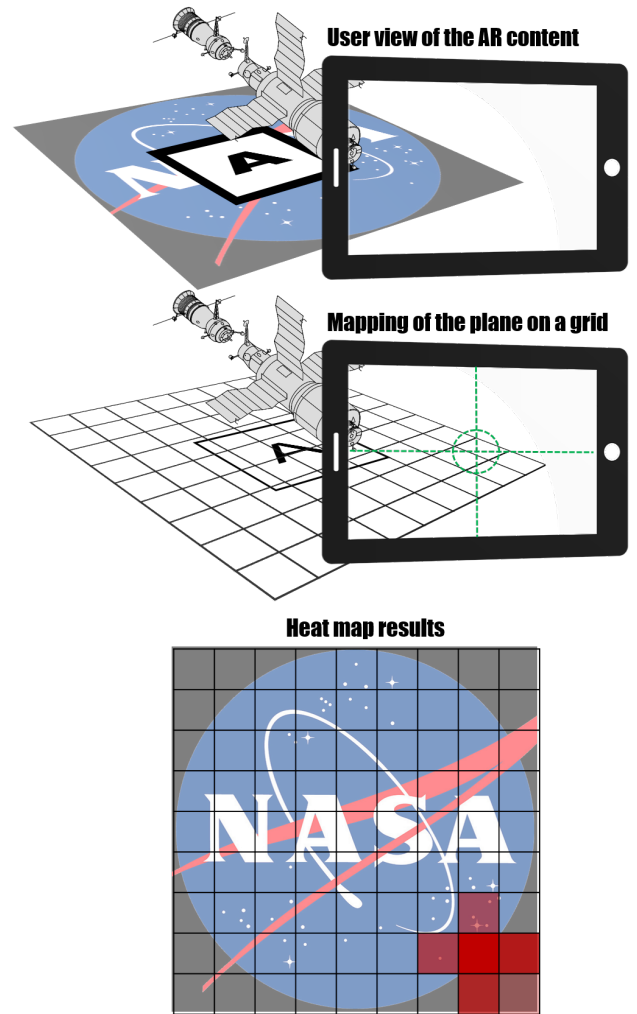


Figure 4: Concept of the 2D heat map on our *Data Analysis Tool*.

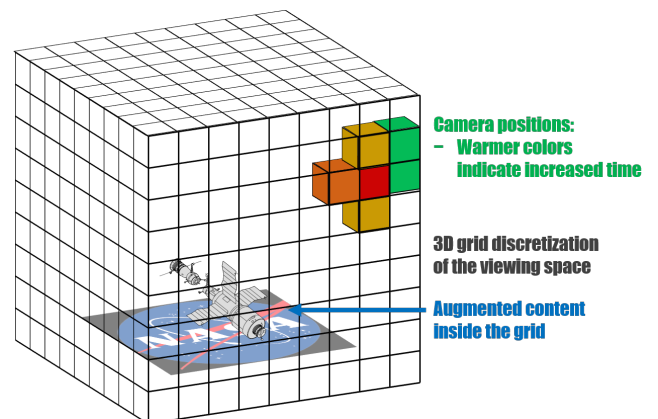


Figure 5: 3D analytics concept showing the 3D grid over the augmented objects, as well as the hotspots of the user position as colored voxels.

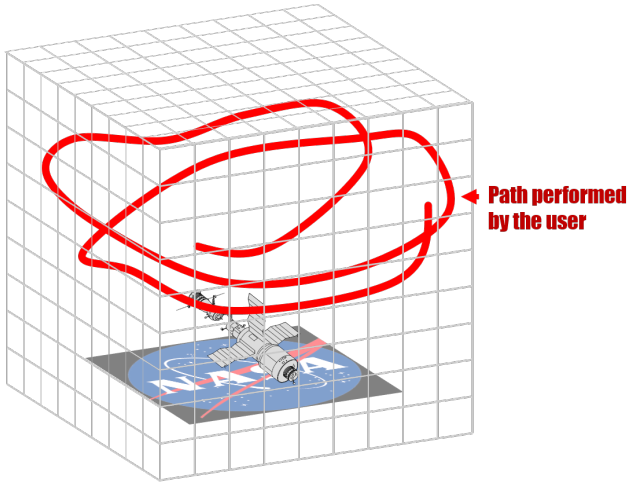


Figure 6: Path traced by the user during the experience with the AR application.

4 FRAMEWORK IMPLEMENTATION

We have developed a framework for providing AR applications analytics. This framework has two main modules: data capture and data analysis. The data capture module is a C++ library able to collect and store data from any AR application. To do so, developers have to add the provided library to their applications and simply use the library methods. To visualize and understand the collected data obtained from data capture, the data analysis module is used. Figure 7 provides an overview of the proposed augmented reality analytics framework.

The first framework module is responsible for data capture from the AR application (IOS ARToolkit [1] application in our example) by using the provided data capture C++ library. After this, a log file with data acquired from the application is generated. Listing 1 shows an example of a log file. The first three float numbers represent the camera center, while the following three numbers represent its view direction, the last integer number represents the time spent in this state in milliseconds. This file sizes 2MB to a 60s record and is used as input to the last step, the data viewer, a tool developed using THREE.JS library [16] which can run in modern web browsers.

Listing 1: Example of a generated log file

```
91.4096 -216.137 178.876 -0.1097 0.7909 -0.6019 43
51.4362 -143.164 294.833 -0.001456 0.5116 -0.8591 67
45.3716 -143.455 292.925 0.01267 0.51068 -0.8596 36
51.3777 -142.369 295.549 -0.006946 0.50333 -0.8640 26
44.5899 -147.26 289.178 0.0107257 0.521414 -0.853236 32
48.3618 -144.573 294.372 -0.001137 0.50860 -0.860997 33
43.5719 -144.864 291.856 0.0126232 0.513285 -0.858125
42.3215 -137.106 297.522 0.0155918 0.490897 -0.871078 33
48.131 -141.42 296.843 -0.00180715 0.503091 -0.864231 32
41.7779 -143.799 296.21 0.0142659 0.512103 -0.858806 33
40.7463 -148.416 291.417 0.0174645 0.530347 -0.847601 33
```

4.1 Data Capture

In order to collect a variety of data from AR applications, we developed a data collector that stores information about user position and orientation in three-dimensional space and the time spent on each position. The developers have to provide the 4x4 modelview matrix (extrinsic parameters matrix) to the library. This matrix is passed calling the *add.feature* method as showed in Listing 2 as a 16-dimensional vector, then the library can extract the needed data. In order to generate the log file, the developers have to call

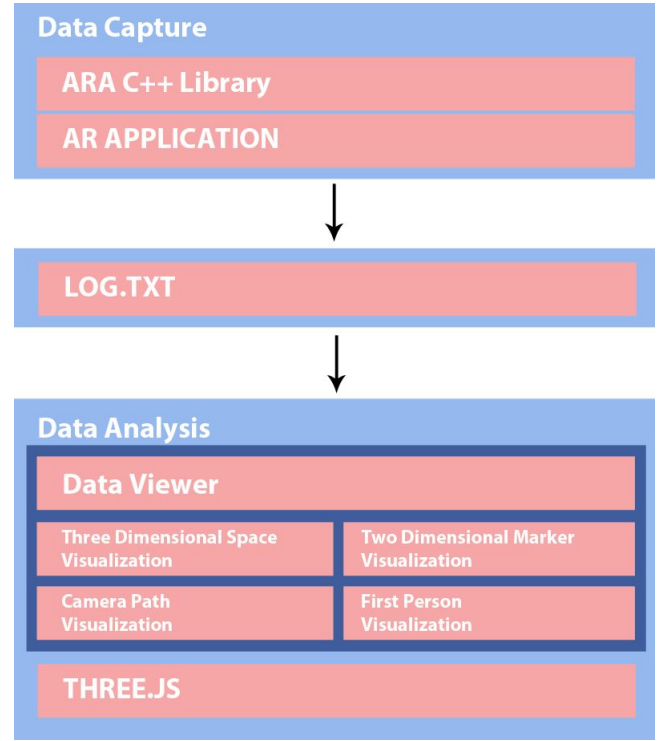


Figure 7: Framework overview.

save_log_file method as showed in Listing 3 when it detects the end of application.

The data collector is still not concerned with network integration and database storage. Therefore all the collected data is stored in a local text file and passed manually by the user to the data analyzer.

While user interacts with the AR application, the data collector stores the camera pose (its three-dimensional position and orientation) and the time spent on each pose. The three-dimensional position of the camera was obtained using equation 4, while its direction (the look at direction) is obtained from the first column of 3x3 rotation matrix (equation 2). With this information it is possible to evaluate the total amount of time spent on the application regarding its AR part, the parts of the model that the user spent more time viewing and the region of three-dimensional space where the user camera spent more time. This information are useful because sometimes it is necessary to know where to put an important information, for example, an advertising or information about a museum sculpture on a virtual model. Using these data it is possible to figure out the warmest regions of the model and thus put the important information there. They are important also because they can allow building strategic approaches to catch user's audience for a longer time.

$$MV = \begin{bmatrix} w1 & w2 & w3 & t_x \\ x1 & x2 & x3 & t_y \\ y1 & y2 & y3 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} w1 & w2 & w3 \\ x1 & x2 & x3 \\ y1 & y2 & y3 \end{bmatrix} \quad (2)$$

$$t = [t_x \quad t_y \quad t_z] \quad (3)$$

$$C = -R^t * t \quad (4)$$

Our data collector was developed as a library in C++ language and can be easily integrated with any AR application. To validate it we have put our data collector in an example application developed for the iOS platform using the ARToolKit library [1]. To obtain the generated log file in performed tests in iOS we had to use the iTunes application to recover it and so our save method was adapted to deal with this specific case. The code 4 shows how we save the log file in iOS.

Listing 2: Method used to add/store a new feature

```
static void add_feature(const float[] model_view_matrix);
```

Listing 3: Method used to save the log file

```
static void save_features(const std::string& dir);
```

Listing 4: iOS-specific code used to save the log file

```
void AnalyticsController::save_features(const std::string
& file_path)
{
    std::cout << "Saving file..." << std::endl;
    std::stringstream ss;

    // Converts the stored data into string
    for (auto &data : mem_data)
    {
        ss << data.center.x << " " << data.center.y << " "
            << data.center.z << " "
            << data.look.x << " " << data.look.y << " " <<
            data.look.z << " "
            << data.time << std::endl;
    }

    mem_data.clear();

#ifdef __APPLE__

    NSError *error = nil;
    NSString *plain_data = [NSString stringWithCString:ss
        .str().c_str() encoding:[NSString
        defaultCStringEncoding]];

    NSString *path = [NSString stringWithCString:
        file_path.c_str() encoding:[NSString
        defaultCStringEncoding]];

    NSURL *final_path = [NSURL URLWithString:[path
        stringByAddingPercentEscapesUsingEncoding:
       :NSUTF8StringEncoding]];

    [plain_data writeToFile:[final_path path] atomically:
        YES encoding:NSUTF8StringEncoding error:&error];

    if(error)
    {
        NSLog(@"Error: %@", error);
    }

#endif

    // ...
}
```

4.2 Data Analysis

The captured data using the data collector explained in the last section, despite the potential demonstrated there, is not so useful if visualization tools are not developed in order to easily understand its meaning. Thus, we developed a data analysis tool with four main features: three-dimensional space visualization, two-dimensional

marker visualization, camera path visualization and first person visualization. With these two features, we can take good insights about user interaction in an AR application.

Our data analysis tool was developed in JavaScript language using the THREE.JS library [16] and is supported by the most recent web browsers. This choice allows us to provide an easy to use web-based dashboard to anyone have access and analyze the collected data.

4.2.1 Three Dimensional Space Visualization

This feature can show a three-dimensional heat map using the points where users passed while they were executing the application. Figure 12 covers this feature working. Using data collected regarding three-dimensional position and time spent, we discretized the three-dimensional space in voxels and painted each one with a representative color: red means the users stayed there for a longer time while green means poorly visited regions. This three-dimensional color map becomes a three-dimensional heat map.

To discretize the three-dimensional space we normalize each three-dimensional point by dividing each coordinate by the maximum coordinate value of all points. Then we apply equation 5 to this point in order to get the voxel coordinate. The voxel resolution is a parameter which can be changed in run time and it impacts the number of voxels and thus the precision of the visualization tools.

$$voxel_coordinate = \lfloor discretized_point * RESOLUTION \rfloor + RESOLUTION \quad (5)$$

In order to obtain the color of each voxel as described above, we first obtain the maximum amount of time the user spent in all voxels. Using equation 6, we obtain a circular parametrization where each coordinate system represents a color component, as described in figure 8, and Θ represents the relation between the time spent in that voxel and the maximum time the user spent in all voxels. Then, using equations 7 and 8 we obtain each color component and then we define the final color using equation 9.

$$\Theta = \frac{voxel_time * \pi}{2 * max_time} \quad (6)$$

$$red = \lfloor 255 * \sin(\theta) \rfloor \quad (7)$$

$$green = \lfloor 255 * \cos(\theta) \rfloor \quad (8)$$

$$voxel_color = \begin{bmatrix} red & green & 0 \end{bmatrix} \quad (9)$$

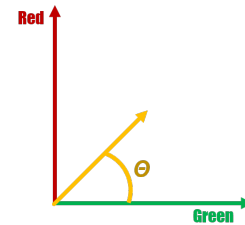


Figure 8: The representation in a two dimensional space of relation between the time spent int that voxel and the maximum time the user spent in all voxels.

4.2.2 Two Dimensional Marker Visualization

This feature shows, on top of a marker plane, a heat map with the regions the user looked at for a longer time. Using the camera center and its look at vector calculated for each point by which the user passed while interacting with the AR application, we trace straight lines, one for each point, and calculate the intersections of these lines with the marker plane, located at $z = 0$ as described in equation 10. With these intersection points, we can build a heat map based on the time the user stayed looking at that region and paint each region. We paint each region in red color, regions that were not seen have opacity zero, yet regions that user looked for more time have opacity one. This feature brings us useful insights of points of the marker that are catching more the user attention. Also, it is possible to infer what parts of the virtual model the user is viewing more. Figure 11 shows this feature working.

$$intersection = center.xy + direction.xy * \left(\frac{center.z}{direction.z} \right) \quad (10)$$

4.2.3 Camera Path Visualization

This feature is able to show the path traveled by the user camera during the user interaction. To do so a red line is drawn for each calculated camera center position as presented in figure 9.

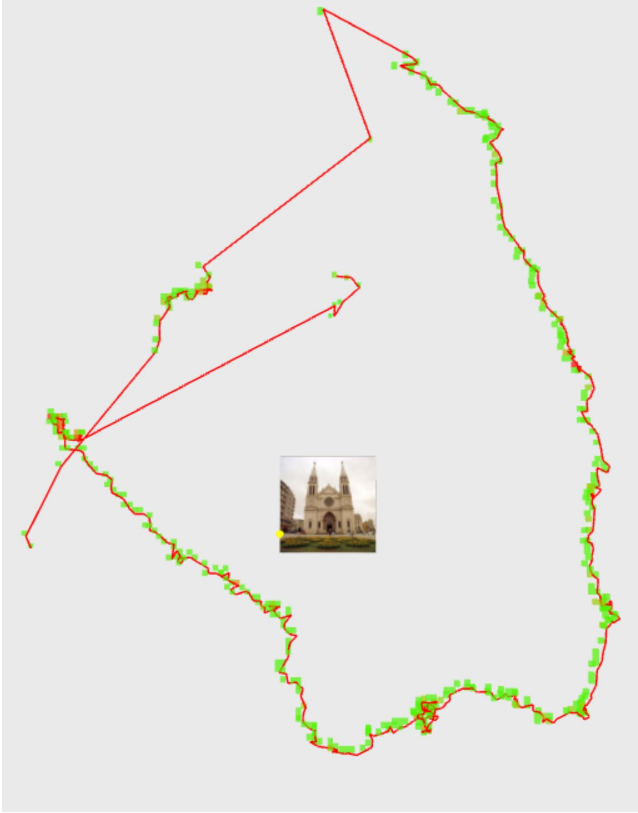


Figure 9: Top view of the camera path, performed by the user while interacting with application, around the marker (in red). The green cubes represent camera positions sampled by the proposed capture tool.

4.2.4 First Person Visualization

This feature presents the user viewpoints in an animated format in order to better understand how the user interacted in the three-dimensional space with the template. Figure 10 illustrates some of the frames captured.

4.3 Framework Validation

The proposed framework is able to deal with any AR application type, regardless of whether it is marker based or not. In our test scenario, described as follows, we choose a natural feature tracking (texture marker) application. In order to validate the developed framework we performed some tasks in our ARToolkit example application embedded with the proposed data collector library and then used the data visualizer to evaluate if the obtained data were coherent with the task performed.

In the first test, we put a target point at the center of an iPad (just to serve as a reference) and asked the user to follow the contour of the template (a church in this case) several times, as illustrated in the left image of Figure 11. The right image of Figure 11 is obtained from the data analyzer and shows the result of the task performed by the user. Despite some points out of church contour, which are probably result of user mismatch and tracking failures, the two-dimensional representation was accurately mapped by our data analysis tool. Figure 12 represents the 3D heat map regarding the same test performed.

In the second test of the framework, the template was put on a table and we asked users to surround it with the iPad several times while looking at it. Figure 9 shows the obtained points to this test case.

We conclude that in fact the behavior of the user in the real world was precisely mapped by our data analysis tool and so this tool can be used to understand how the user is interacting in regards to camera movement within the AR application.

5 CONCLUSION AND FUTURE WORKS

In this work we developed an AR analytics based framework, which was then discussed and evaluated regarding its potential. This framework provides a data capture library and a data analysis tool which can be used to evaluate the collected data from AR applications and can be useful to understand the user behavior in this kind of applications. We also applied it with real users in order to validate the framework applicability in real context.

Because this way of performing analytics in AR applications is still unexplored, it is necessary to invest in several future works. We intend to provide other ways to visualize the data. Although we believe that heat maps fit well to AR application data analysis, we still need to study different alternatives in order to figure out if there are more useful ones. We can also collect data from device sensors such as accelerometer and gyroscope in order to use when tracking fails, for example. Moreover, we intend to develop tools to deal with mouse and touch interaction (a click, for example), because we believe that it is an important way of interaction and can also deliver useful insights regarding AR applications use.

We also plan to test this tool with other AR platforms like Microsoft Hololens [12] and in other computer platforms such as Desktop and Android ones. Automatic posting of the tracking information to the web would be very useful and could reduce the framework's use complexity. In order to implement such feature, one may consider analyzing data compression, server optimization for large scale storage and other specificities related to network transmission.

We believe that our AR analytics framework has the potential to help people who are building AR applications to understand better their users in this kind of context and so get insights in advertising and selling applications and improving the quality of user interaction, thus delivering a better user experience.

ACKNOWLEDGMENTS

The authors wish to thank A Pró-Reitoria de Pesquisa e Pós-Graduado (PROPESq), project identification number 170112305, which supports the scientific initiation of the author Marcel S. Santos.



Figure 10: Frames captured by the First Person View mode. This visualization recovers the user view from camera pose.

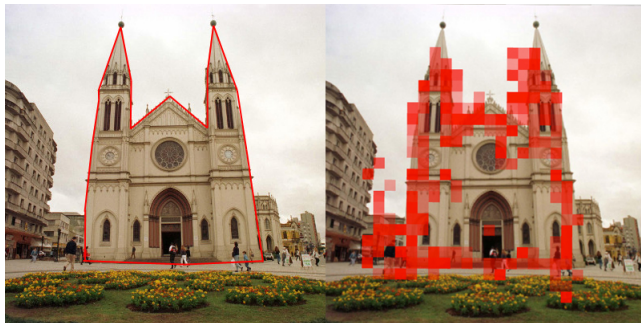


Figure 11: The left side image is a representation of 2D path requested to the user to follow while using the application. The right side image shows the points obtained by the collector and rendered in the data analysis tool.

REFERENCES

- [1] ARToolKIT. Artoolkit library. URL = <https://artoolkit.org/>.
- [2] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.
- [3] Blippar. Blippar provides business solutions that harness powerful augmented reality technologies., 2017. URL = <https://blippar.com/en/>.
- [4] A. Canossa, T.-H. D. Nguyen, and M. S. El-Nasr. G-player: Exploratory visual analytics for accessible knowledge discovery. 2016.
- [5] H. Chen, R. H. Chiang, and V. C. Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4):1165–1188, 2012.
- [6] Craftar. Craftar, 2017. URL = <https://catchoom.com/product/craftar/augmented-reality-and-image-recognition/>.
- [7] Dailymgenius. How augmented reality helps manufacturing enter a new phase, 2017. URL = goo.gl/KuVd6Z.
- [8] T. Economist. The promise of augmented reality., 2017. URL = <http://www.economist.com/news/science-and-technology/21716013-replacing-real-world-virtual-one-neat-trick-combining-two>.
- [9] Firebase. Firebase. URL = <https://firebase.google.com/?hl=pt-br>.
- [10] Goodworklabs. The future of ar technology., 2017. URL = goo.gl/tW847M.
- [11] Google. Google analytics. URL = <https://analytics.google.com/>.
- [12] Microsoft. Microsoft hololens, 2017. URL = <https://www.microsoft.com/en-us/hololens>.
- [13] T. Olsson, E. Lagerstam, T. Kärkkäinen, and K. Väänänen-Vainio-Mattila. Expected user experience of mobile augmented reality ser-

vices: a user study in the context of shopping centres. *Personal and Ubiquitous Computing*, 17(2):287–304, 2013. doi: 10.1007/s00779-011-0494-x

- [14] Optimizely. Optimizely. URL = <https://www.optimizely.com/>.
- [15] G. Sachs. Virtual and augmented reality. understanding the race for next computing platform. 2016. URL = <http://www.goldmansachs.com/our-thinking/pages/technology-driving-innovation-folder/virtual-and-augmented-reality/report.pdf>.
- [16] THREE.JS. Three.js library, 2017. URL = <https://threejs.org/>.
- [17] Yahoo! Flurry analytics. URL = <https://developer.yahoo.com/analytics/>.

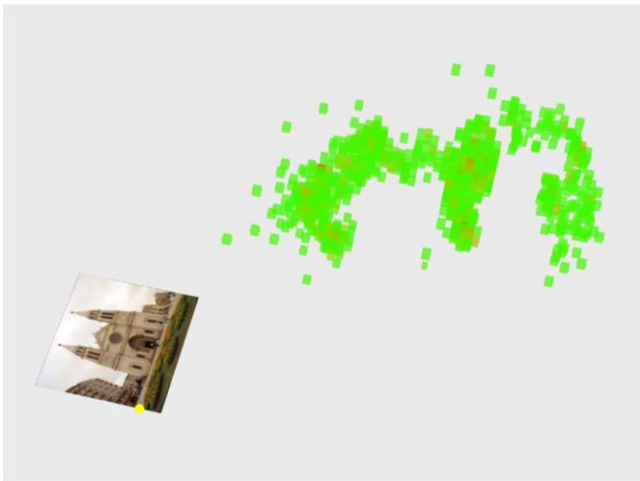
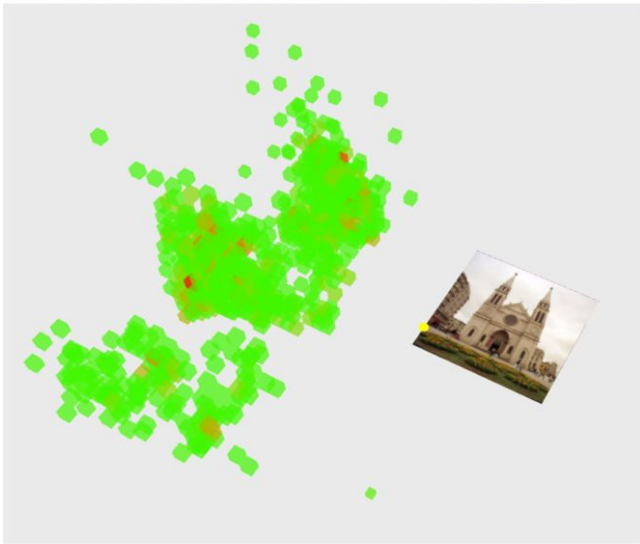
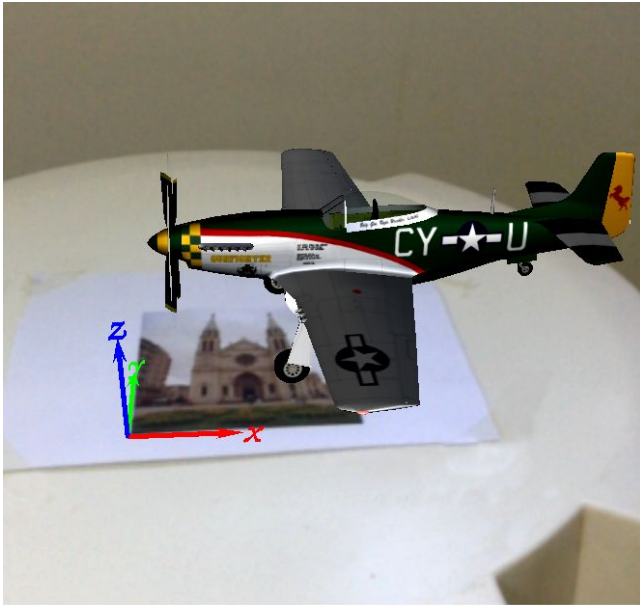


Figure 12: The top image was captured from an iPad, representing a user following the 2D path on the marker. The middle and bottom images show two different views of the 3D points captured by the tool.