Enhancing Spatial Exploration of Outdoor Object Recognition and Tracking with ARToolKit NFT Markers

Blagoj Nenovski, Igor Nedelkovski

University St Kliment Ohridski, 1 Maj nn, 7000 Bitola, R. North Macedonia

blagoj.nenovski@uklo.edu.mk; igor.nedelkovski@uklo.edu.mk

Abstract:
ARToolKit is an open source augmented reality toolkit that supports the recognition of fiducial markers and NFT (natural feature tracking) markers. There is significant research on optimizing and improving fiducial markers but an evident research gap on NFT makers. In this paper we provide a continuation of our previous research on creating NFT markers of outdoor objects by: choosing a source for the marker photo, comparing a range against the entire objects façade, address the level of initialization features and the level of tracking features in the NFT marker creation process.

Keywords:
ARToolKit, NFT, AR, Augmented reality, marker creation

1. Introduction

In our previous work [1] we defined an augmented reality end-to-end platform for spatial exploration with the time as an added component. This defining process was part of our goal to enable the exploration of past and future or houses or buildings that can be recognized by their natural features extracted from a photo of their façades. Since ARToolKit did not support cloud recognition we created a platform consisting of a smartphone app for recognizing the objects and displaying multimedia and a server for storage and distribution of the markers. By setting the bar for the platform to be available to as many users as possible we had to focus and develop guidelines for the best practices for creating markers and app settings that would allow for fast recognition and stable tracking. This translated to an app that would run on Android smartphones with low specifications, but thanks to the rapid development of smartphone processors and the included components such as camera, GPS and 4G connectivity we could focus on recognizing the objects by their natural features instead of using fiducial markers. We addressed the generating markers with the available parameters and the process of adding markers. We also addressed the importance of the camera calibration when using ARToolKit for marker recognition.

When recognizing outdoor objects there are factors in the environment such lighting, reflective surfaces and occlusion that can drastically affect the user experience. In a setup where the before mentioned conditions are similar, the methodology of marker creation as well as the app parameters are of key importance. We have worked on resizing the photo before creating the marker, emphasized the importance of camera calibration as well as the camera resolution on the recognition speed and quality tracking of outdoor objects [2].

2. Related work

There is a comparative study of planar fiducial markers [3] that analyzes the literature, describes the differences and limitations and conducts detailed experiments to compare the sensitivity, specificity, accuracy, computational cost and performance under occlusion.

Research has been done on comparing systems like ARTag, AprilTag and CALTag on the reliability and detection rate when occlusion of various types and intensity is present [4]. ARToolKit markers have been compared with similar systems like ARTag on the reliability, detection rates, and immunity to lightning and occlusion [5].

Fiducial marker optimizer is presented [6] in order to optimize the design attributes of ARToolKit markers, including black to white ratio, edge sharpness, information complexity and to reduce inter-marker confusion. There are multiple factors [7] that are important when designing and tracking ARToolKit fiducial markers. Each of the factors can affect the accuracy, detection speed and inter-marker confusion. The specific distribution of tracking accuracy and its dependency on the distance and the angle between the camera and the fiducial marker is addressed in [8].

In regard to ARToolKit specifically, there is research on the effect of edge sharpness, noise and markers distinction on markers reliability with a developed specialized algorithm for designing sharp-edged, de-noised and distinct markers [9]. Fiducial markers in ARToolKit have been explored in terms of marker sizes, marker distance from the camera, marker speed, the brightness in environment, the contrast level of lighting, as well as the correlation between marker size and distance [10].

Work has been done on solving the tracking failure problem on partially occluded marker in multi marker environment with addition of codebook based foreground detection model for detecting hand region in unexpected background environment [11]. There is a path generation algorithm [12] that automatically identifies fiducial markers in a building in order to create a path for user navigation. The algorithm has been implemented in an android application and internal mechanism for database creation and guidance system has been discussed.

Even relatively new research on the development of a network camera system for long distance use of augmented reality function using ARToolKit [13] focuses on using fiducial markers.

Another research focuses on tailoring paper media markers, improving recognition accuracy via integrated single-response matrix and optimized image matching for real-time tracking. Enhancing ARToolKit SDK's image segmentation by simulating scene changes with a 45° marker card rotation relative to the camera is addressed in [14].

3. Methodology

To determine the recognition speed we used ARToolKit’s feedback on the state of a marker being loaded and a marker being recognized. We achieved this by subtracting the marker load time from the marker recognition time. Our focus was to create better markers that could easily be recognized and tracked by entry level devices so we used Samsung J3(2017) and Samsung J4+ (2018) as validation devices. To create the markers we used different devices to eliminate the advantage in a scenario where the marker creation device is the same with our validation smartphones. While testing the recognition speed we put both smartphones in a fixed position and run each test for 5 times. For quantification of the marker tracking we simulated the use of the app in a perspective of a user that is using the app for the first time: holding the smartphone in a natural position, pointing it at the object until the marker is recognized after which we simulated various intensity phone movements. All of the tests were done with sampleRate set to 30 and cutoffFreq parameter set to 15.

To remove the parts that are not needed for object recognition and to enable quality display of multimedia over the recognized object, prior the marker creation we extracted the objects façades from the photos. Based on the conclusion of our previous work the images were resized to 1000 pixels before the marker creation. Visualized results of the recognition speed are the representation of subtracting the marker recognition time form a baseline of 5000ms as our defined limit for good user experience.

4. Choosing a source for the marker photo

First we wanted to determine the difference in recognition and tracking of markers created with different devices. For this we took a photo of the object for recognition, with a mobile phone (different from the validation devices) and with a digital camera (Nikon D3300). We initially scaled both images down to 1000 pixels and created markers from them with DPI values set to 96, 48 and 24. The created markers resulted in the following files (expressed in size) and the following number of initialization and tracking features.

Table 1:
File size and number of features

		File size in KB				features	
DPI	Source	iset	fset	fset3	Sum (KB)	fset	fset3
96	Smartphone	94	3	76	173	142	596
96	Digital camera	77	3	77	157	136	588
48	Smartphone	28	1	67	96	39	523
48	Digital camera	26	1	68	95	44	517
24	Smartphone	8	1	40	49	10	294
24	Digital camera	8	1	38	47	11	307

From table 1 we can see that the size of the files, as well as the number of initialization and tracking features have close values when using a smartphone and a digital camera for the photo from which the markers were created.

We performed additional analysis of the extracted features for each of the markers and found that for the most part the features are repeated among the markers with the same DPI value. However, markers created from a mobile phone and a digital camera photo are not identical and that they contain non-repeating features or features that are located in different positions.

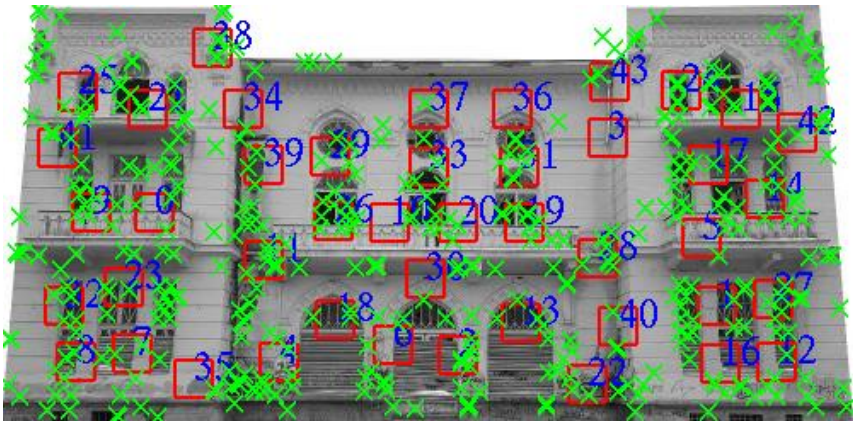

Figure 1: Extracted features, source: smartphone, 48 DPI

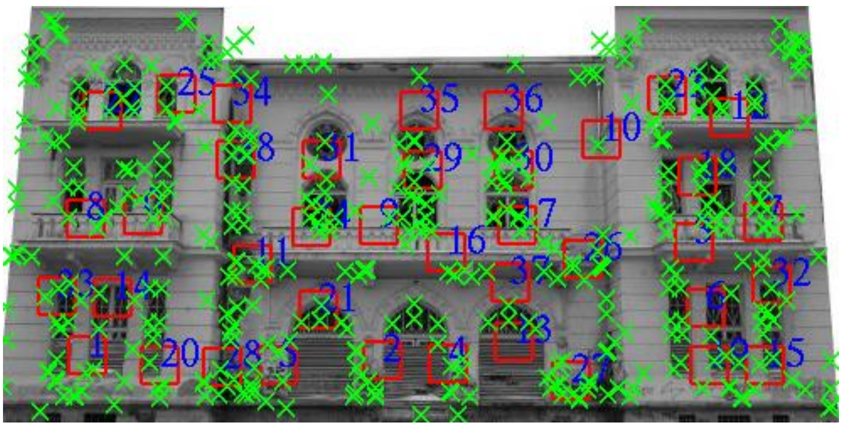

Figure 2: Extracted features, source: digital camera, 48 DPI

From the obtained results, we can see that although the original photo taken with a digital device has a higher resolution and a greater number of details, by reducing the photo to 1,000 pixels, regardless of the source, the recognition speeds in both cases are almost identical.

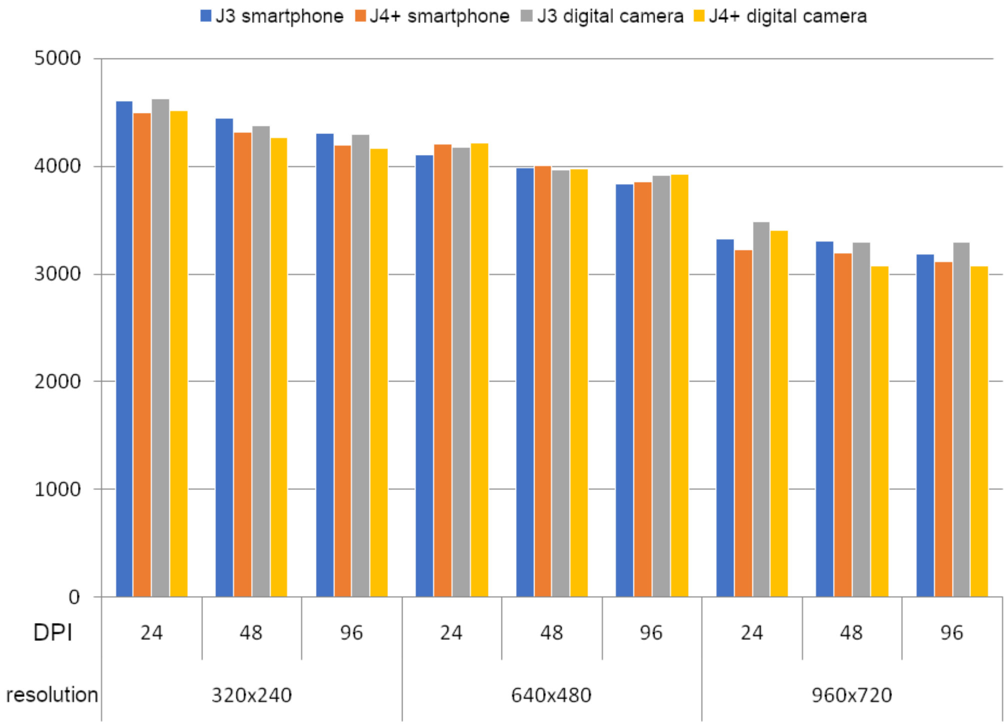

Figure 3. Recognition speed of markers created from smartphone and a digital camera photo

By further checking the quality of the tracking, we concluded that using each of the resolution values and each of the DPI values of the created markers, we obtained quality tracking, regardless of whether the photo to create the markers was taken with a mobile phone or a digital camera.

5. Range of the façade to create the markers

Here we focused on the range of the façade photo from which the markers are created. The reason we implemented this step is that the conditions do not always allow for a complete view of the objects. These conditions can be: the interference from additional objects or nature; many reflective surfaces; insufficient space to move away from the object.

In this step we tested the recognition speed as well as the tracking quality. In doing so, we compared the marker from the previous step (size: 1,000px, source: phone) with a marker with a different range from the same photo (in this case of the middle part of the façade). Our platform automates a large part of the steps, so for this marker we extracted the middle part of the façade from the original photo from the same photo as the previous marker. Then we reduced the image to a size of 1000 pixels and created markers with a DPI value of 96, 48 and 24. As a result we got the following markers and number of features.

Table 2:
File size and number of features

		File size in KB				features	
DPI	Range	iset	fset	fset3	Sum (KB)	fset	fset3
96	Wide	94	3	76	173	142	596
96	Middle	142	5	79	226	225	605
48	Wide	28	1	67	96	39	523
48	Middle	48	2	64	114	58	491
24	Wide	8	1	40	49	10	294
24	Middle	14	1	46	61	21	353

From table 2 we can see that the number of initialization features is almost the same for 96 DPI (wide 596, middle 605), higher for wide (523) than middle (491) for 48 DPI, and higher for middle (353) compared to wide (294) for 24 DPI. However, the difference in these initialization feature numbers is not large as seen from the recognition speed results. Here we should state that the features from the middle marker are located in one part of the camera view when exploring the object. We can see the extracted features of this marker in the following figure.

Figure 4. Extracted features, 48 DPI

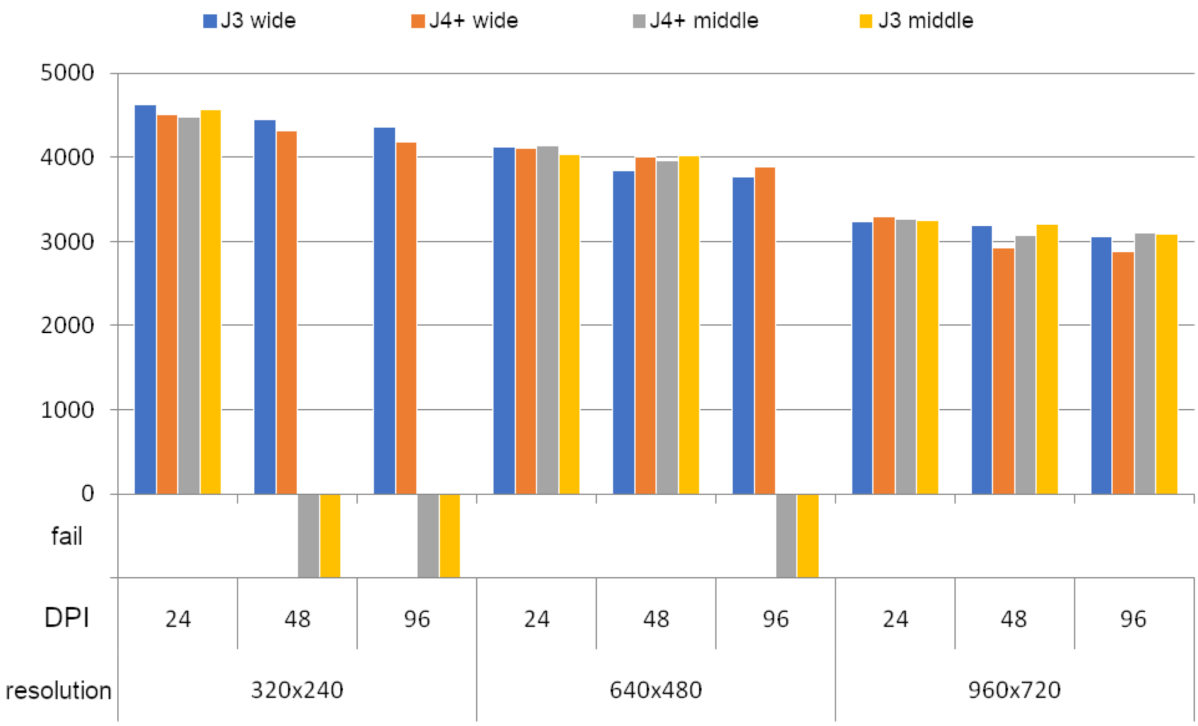

Figure 5. Recognition speed of markers created with different ranges of the photo

From the obtained results, we realized that in the cases where we had successful recognition with both markers, the speed of recognition was almost identical. On the other hand, since the testing was done with a camera setting that had a view of the entire object, we got 3 cases in the two phones: 320 x 240 camera resolution (48 DPI and 96 DPI) and 640 x 480 camera resolution (96 DPI) where we had failed recognition. With these cases, we confirmed the relationship between the DPI value and the resolution value of the camera.

Since with the marker with a DPI value of 24 we had a smaller number of features, we got a successful recognition with the camera resolution at 320 x 240. At the same resolution we had a failed recognition for the markers created with DPI values of 48 and 96. The same situation repeated for the 640 x 480 camera resolution for the marker with a value of 96 DPI. Already at a resolution of 960 x 720, enough details were obtained from the camera to recognize all three values for the middle marker. Of course, if the phone was placed closer to the middle of the object, then we would get different results, but in such a case the display image would cover the entire screen, so the object could not be explored by changing the position of the photo in relation to the position of the marker in the camera view.

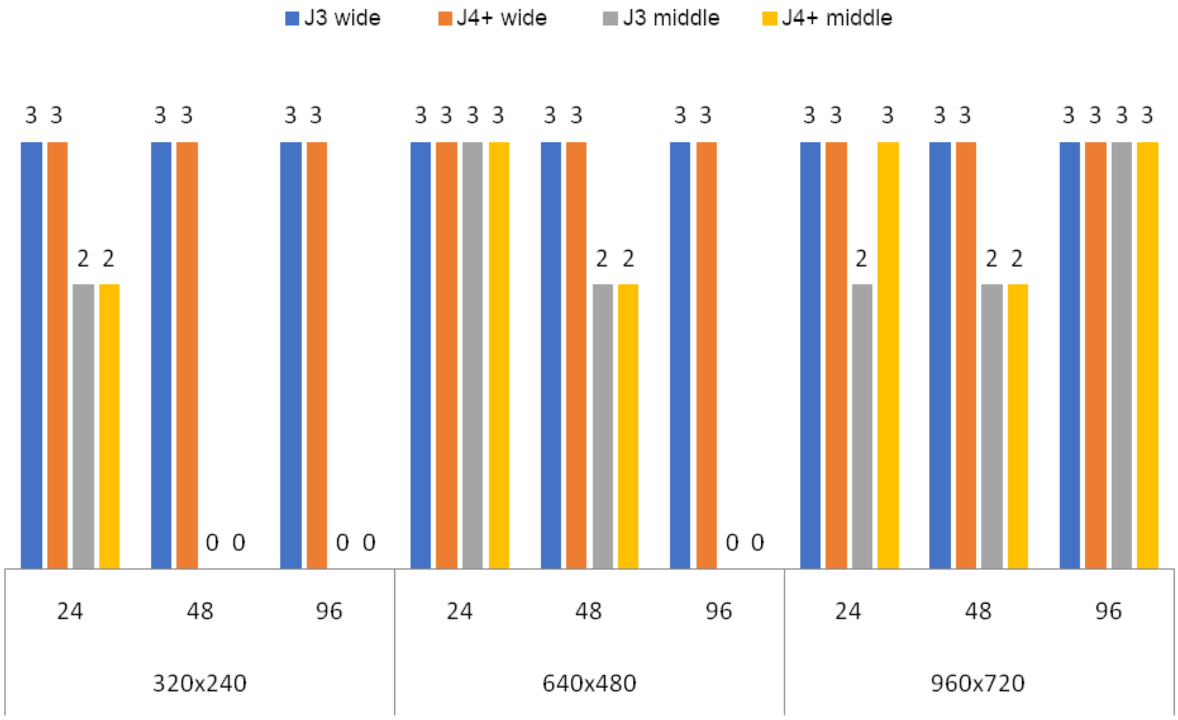

Figure 6. Tracking quality of the markers created with different ranges of the photo

In the tracking quality check we again failed to achieve recognition in the same cases that got the failed results in the recognition speed validation. In two cases (resolution 640 x 480, 48 DPI and resolution 960 x 720, 96 DPI) we had high tracking quality with both markers and both phones. In all other cases, either one or both phones consistently had lower tracking quality when the middle marker was used. As we previously emphasized, this condition is due to the distance from which the object is being explored. Since in the case of the middle marker, only the middle part of the building's façade is used and the same photo (of the entire building) is used for display, we had a lower tracking quality with this marker. This happened both when the phone is held steady and during similar movement, as for the wide marker.

6. Level of the initialization features

In the fourth step we created individual markers with a DPI value of 48, a default value of 2 for level of the tracking features, but a different value for the initialization features level from 0 for the lowest level, up to 3 for the highest level of initialization features.

For each of the four individual markers we got files of equal size, i.e. iset 28 KB, fset 1 KB, and fset3 67 KB. For the default level (2) of tracking features, we obtained the same number of features (44) for each of the markers. On the other hand, despite the fact that for each of the four markers we had set a different level of initialization features, for each of the markers we obtained the same number of features (517).

In the five repetitions of individual tests for the marker, we got results that were close to each other, which we can see in the graphical representation of the average value from the two phones.

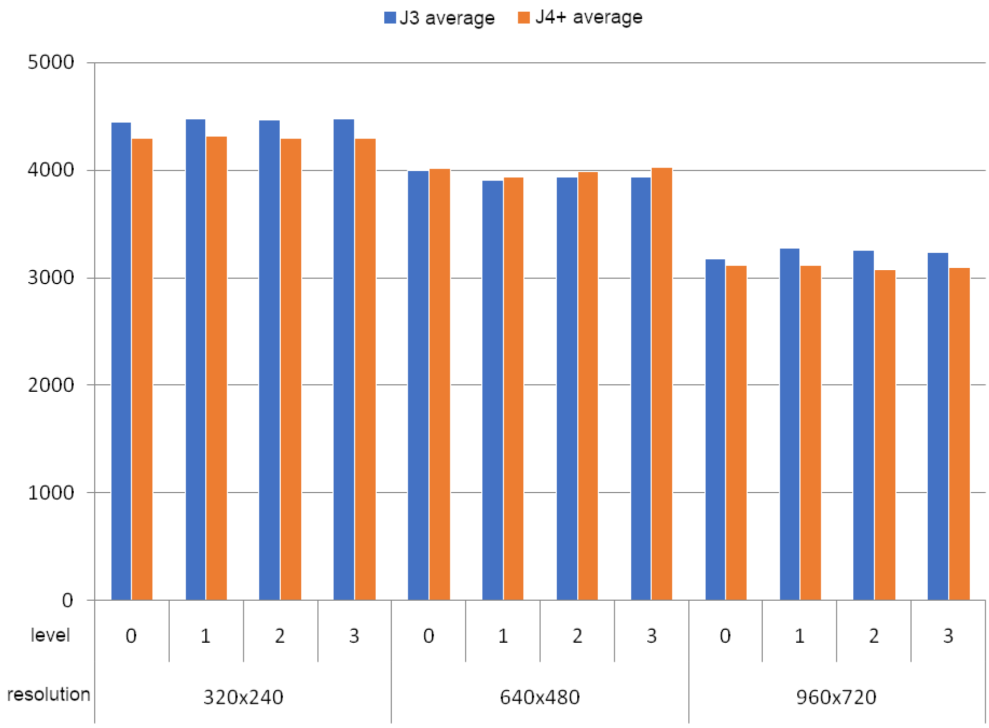

Figure 7. Recognition speed at different levels of initialization features

As a result of the identical markers, we got identical results, regardless of the set level of the initialization features. From the display in the graph, we can see the difference in the recognition speed with respect to the change in the camera resolution of the phones. Again we can see the trend of decreasing recognition speed when increasing camera resolution. This difference is smaller between values 320 x 240 and 640 x 480, where the difference is less than half a second, and larger between values 640 x 480 and 960 x 720, where the difference in recognition speed is about one second.

Getting the same result for the initialization features regardless of the setting is due to the setting not being supported by ARToolKit 5.3. Customization was part of the SURF extractor used in V5.2, but switching to the FREAK detector in 5.3 regardless of the settings gives the same result.

In the tracking quality test with each of the markers and with each resolution value, we obtained a high level of tracking quality, noting that in real world scenarios, at the time of not recognizing an already recognized marker, the speed of re-recognition effects the user experience. This means that at 320 x 240 and 640 x 480 resolution values, the photo display is significantly faster and provides a more realistic experience compared to the 960 x 720 camera resolution.

7. Level of tracking features

In this step, we tested the speed of marker recognition and its tracking with different levels of tracking features. In the previous step, we observed that regardless of the level set for initialization features, we consistently got the same files and the same number of features, resulting in equal recognition speed. Since there are a total of 5 levels of tracking features, in this step we created a total of 5 markers and set a variable level for the tracking features while keeping the same level for the initialization features. The different numbers of tracking features are shown in the following table.

Table 3
Number of features at different set level of tracking features

level	fset3 features	fset features
0	17	517
1	22	517
2	44	517
3	44	517
4	68	517

.

As expected the recognition speed is the same as in the previous step. When analyzing the tracking quality, we obtained stable tracking for each of the markers in each of the resolution values. Only, when using the 320 x 240 resolution and tracking features at level 0, we got a lower tracking quality, but such a difference is only noticeable when directly comparing the different markers.

8. Conclusion

We can conclude that when following the principle of reducing the photo size prior creating the NFT marker, using a digital camera for taking the photo does not differ from taking the photo with a smartphone in regards to both the recognition speed and the tacking quality.

With successful recognition of NFT markers when using a part of an objects façade for marker creation we can expect similar speed to when using the entire façade. But there are instances when using part of the façade that result with failed recognition as well as the problem with overlaying multimedia while tracking the marker.

We noticed that although there is a setting for selecting the level of initialization features, created markers with variable setting for initialization features bring identical file sizes and the same number of initialization features. Choosing the level of tracking features plays a small role in the tracking quality, except for when using the lowest level while coupled with 320 x 240 for the camera resolution.

References:

[1] B. Nenovski and I. Nedelkovski, "Defining a feature-rich end-to-end augmented reality platform for spatial exploration," in Proceedings/8th International Conference on Applied Internet and Information Technologies, vol. 8, no. 1, pp. 103-108, "St Kliment Ohridski" University-Bitola, Faculty of Information and Communication Technologies-Bitola, Republic of Macedonia, 2018.
[2] B. Nenovski and I. Nedelkovski, "Recognizing and tracking outdoor objects by using ARToolKit markers," International Journal of Computer Science & Information Technology (IJCSIT), vol. 11, no. 6, pp. 21-28, 2019.
[3] D. Jurado-Rodriguez, R. Muñoz-Salinas, S. Garrido-Jurado, and R. Medina-Carnicer, "Planar fiducial markers: a comparative study," Virtual Reality, pp. 1-17, 2023.
[4] Sagitov, K. Shabalina, R. Lavrenov, and E. Magid, "Comparing fiducial marker systems in the presence of occlusion," 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), St. Petersburg, Russia, 2017, pp. 377-382, doi: 10.1109/ICMSC.2017.7959505.
[5] M. Fiala, "Comparing ARTag and ARToolkit Plus fiducial marker systems," in IEEE International Workshop on Haptic Audio Visual Environments and their Applications, pp. 6-pp, IEEE, 2005.
[6] D. Khan, S. Ullah, D.-M. Yan, I. Rabbi, P. Richard, T. Hoang, M. Billinghurst, and X. Zhang, "Robust tracking through the design of high quality fiducial markers: an optimization tool for ARToolKit," IEEE Access, vol. 6, pp. 22421-22433, 2018.
[7] D. Khan, S. Ullah, and I. Rabbi, "Factors affecting the design and tracking of ARToolKit markers," Computer Standards & Interfaces, vol. 41, pp. 56-66, 2015.

[8] D. F. Abawi, J. Bienwald, and R. Dorner, "Accuracy in optical tracking with fiducial markers: an accuracy function for ARToolKit," in Third IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 260-261, IEEE, 2004.

[9] D. Khan, S. Ullah, and I. Rabbi, "Sharp-Edged, De-Noised, and Distinct (SDD) Marker Creation for ARToolKit," in Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings 20, pp. 396-407, Springer International Publishing, 2014.

[10] I. Rabbi, S. Ullah, M. Javed, and K. Zen, "Analysis of ARToolKit fiducial markers attributes for robust tracking," in International Conference of Recent Trends in Information and Communication Technologies (IRICT'14), University Technology Malaysia, pp. 281-290, 2014.

[11] H. Wang, J. Qin, and F. Zhang, "A new interaction method for augmented reality based on ARToolKit," in 2015 8th International Congress on Image and Signal Processing (CISP), pp. 578-583, IEEE, 2015.

[12] Ullah, S., Rahman, I. U., & Rahman, S. U. "Systematic augmentation of ARToolKit markers for indoor navigation and guidance." Proc. Pakistan Academy of Sciences: A. Physical and Computational Sciences, vol. 56, no. 1, pp. 1-8, 2019.

[13] K. Ogata, S. Nakatani, and S. Yamaguchi, "A network camera system enabling long-distance use of augmented reality functionality using ARToolKit," International Journal of Innovative Computing, Information and Control, vol. 17, no. 2, pp. 655-669, 2021.

[14] W. Zhang, W. Zhang, M. Xu, Y. Zheng, and H. Wang, "Design and implementation of augmented reality system for paper media based on ARToolKit," in Applications of Digital Image Processing XLIV, vol. 11842, pp. 423-430, SPIE, 2021.