AUGMENTED REALITY: TIMETABLE DISPLAY ON STUDENT ID

Adam Finlayson
University of Canterbury
Department of Mechatronics Engineering, Postgrad
acf60@uclive.ac.nz

Richard Green
University of Canterbury
Department of Computer Science and Software Engineering, Supervisor
richard.green@canterbury.ac.nz

ABSTRACT

This paper proposes a method to provide University of Canterbury (UC) students an Augmented Reality (AR) timetable display above their student IDs, when viewed through a camera. The best AR systems integrate into the real-world environment seamlessly. Real time object tracking systems commonly use predetermined markers for tracking. This paper will review the Aruco marker system, and discuss a method to track and display data above a UC student ID. The results of this paper show that markerless object tracking is more suited for a consumer AR system, and better enhances the real world than traditional marked tracking systems.

1. INTRODUCTION

Recent advances in in mobile computing, computer graphics, wireless and sensor technologies allow for the fast development of Augmented Reality (AR) applications. A visual AR system enhances or augments the surroundings of the user with virtual information that is registered in 3D space and seems to coexist in the real world [1].

For virtual data to look convincingly integrated into an AR environment, feature detection is used to estimate the orientation of real-world objects. This allows an AR system to display virtual data relative to the real-world object, giving it the appearance of coexisting in the real-world.

This paper proposes a method of displaying student timetable data above a University of Canterbury (UC) student ID.

The first method is the use of Aruco markers [2] on the back of a student ID. The second is the use of markerless tracking [3] of the student ID.

2. BACKGROUND

2.1 AUGMENTED REALITY

- Brief background on applications of AR

2.2 ARUCO MARKER DETECTION

One of the most popular approaches to AR pose estimation is the use of binary square fiducial markers. An Aruco marker is a black and white square marker with wide back boarder with an inner binary matrix use as an identifier [4]. Aruco markers are rotation invariant and will be detected in any orientation.

An example Aruco marker can be seen in figure 1.

Figure 1: Example Aruco Marker

Detection of Aruco markers is achieved by the following steps [5]:

- *Image Segmentation:* Uses local adaptive thresholding to extract prominent contours in a grey scale image. (Figure 2).
- *Contour Extraction and Filtering:* Contour extraction is performed on the thresholded image using the Suzuki and Abe [6] algorithm. The Douglas-Peucker [7] algorithm is used to remove any contours that do not form a 4-vertex polygon. (Figure 3)
- *Marker Code Extraction and Identification:* A perspective transform is performed using the homography matrix. The image is binarized and divided into a grid. Each element in the grid is assigned 0 or 1 depending on the values of the pixels in the element. A rejection test is performed to show that there is a boarder of zeros. The inner grid is used to identify the Aruco ID by cross-checking the inner matrix against a dictionary. 4 matrices are checked, one for each rotation. If one of the matrices are within the dictionary, the Aruco marker is considered valid. (Figure 4).

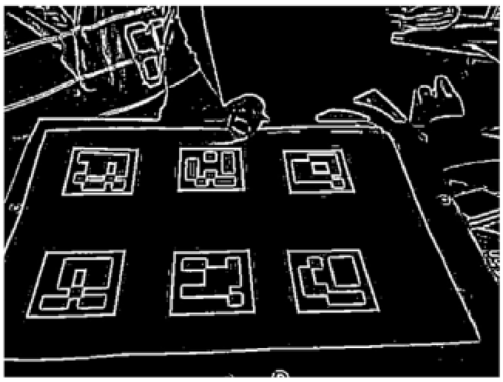

Figure 2: Local Thresholding [5]

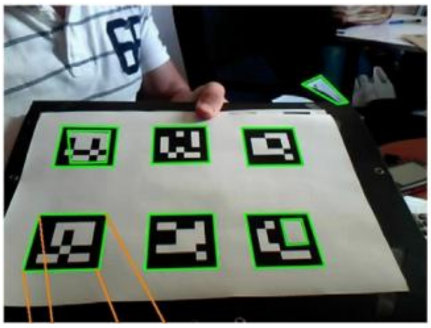

Figure 3: Contour Extraction and Filtering [5]

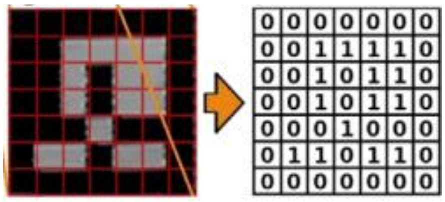

Figure 4: Marker Code Extraction and Identification [5]

As Aruco markers require 4 edges to be detected, if they are partially obstructed, they will not be registered by the detection process. Aruco markers must be square, black and white, and part of a specific Aruco dictionary. This makes them stand out when placed in the real-world.

2.3 MARKERLESS TRACKING

Markerless tracking allows for any object to be tracked by comparing onscreen features, such as edges, points, and colours, to a set of features from preloaded picture. The comparison allows for an object to be detected, as well as its pose estimated.

Markerless tracking allows for seamless detection and pose-estimation of real-world objects, without the need for fiducial markers [8]. This makes it an excellent option for AR systems.

Markerless tracking is achieved by the following steps:

- *Feature Detection:* Algorithms like SIFT [9] and SURF [10] are commonly used for feature detection. However, they are computationally demanding. AR systems are required to run in real time, requiring computationally efficient algorithms for feature detection. Oriented FAST and Rotated BREIF (ORB) [11] is a feature matching technique that is built on the FAST [12] key point detector and BREIF [13] descriptor. The ORB technique is limited by is lack of scale invariance.

- *Feature Matching:* This step matches precalculated features on an image of a single object, to features found in real time on a camera video. If there are enough matches between features, the object will be recognised as in the frame. Figure 5 shows the matching results using ORB on real-world images. Green lines are valid matches, red circles indicate unmatched points.

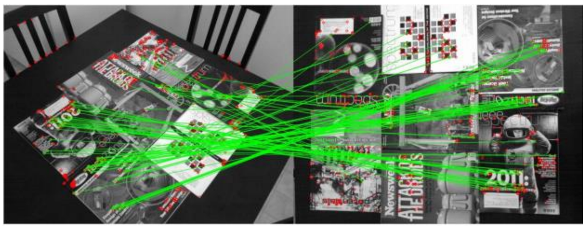

Figure 5: Feature matching using ORB [11]

3. METHOD

To show the limitations of using fiducial markers in an AR environment, an Aruco marker was attached to the back of a UC student ID. A basic Aruco marker tracking program was set up by implementing the steps described in [5]. Figure 6 shows the result of the application of these algorithms.

Figure 6: Aruco Bounding Box detection

By using the corners of this bounding box, an image containing timetable data could be perspective transformed onto the Aruco marker. Figure 7 show the result of this transform.

Figure 7: Student timetable data displayed above Aruco marker

To create a more immersive AR display, a markerless system was also implemented using the step described in 2.3. A still image of a student ID was used to find a set of features that would be used by the ORB feature detection. The image and its features can be seen in Figure 8.

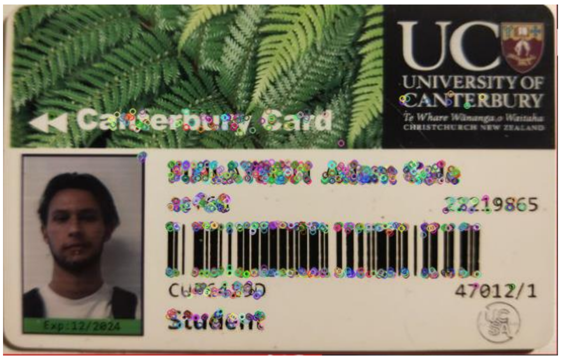

Figure 8: Student ID Features

Figure 9 shows the result of the brute force matching algorithm

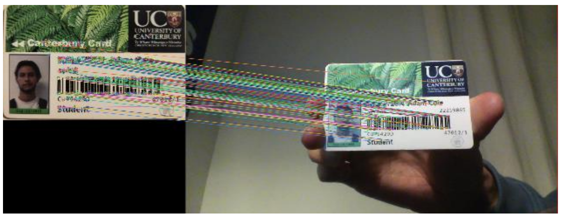

Figure 9: Feature Matching

Figure 10 shows the final markerless mapping.

Figure 10: Student timetable data displayed on top of UC student ID

4. RESULTS

4.1 TESTING ENVIRONMENT

OS: Windows 10 Home.
Processor: Intel Core i7 7700HQ
IDE: PyCharm Community
Language: Python 3.7
Device: Dell XPS 15 9560
Camera: Inbuilt Webcam, 720p, 30fps
OpenCV Version: opencv-contrib-python 4.5.2.54

4.2 IMMERSION

Markerless tracking shows a clear advantage in immersion over traditional fiducial markers. Displaying timetable data using only the features of the student ID provides a more immersive experience as there is no need to alter the ID to use it. This offers a greater degree of environmental enhancement over the fiducial markers.

4.3 COVERAGE

Aruco markers are prone to blockage. Markerless tracking reduces errors when blocked as it utilises individual features instead of contour detection. Figures 11 and 12 shows that Aruco markers fail with very little blockage, such as glare, or a finger. Figure 13 shows that the markerless tracking system can still detect a 50% covered ID.

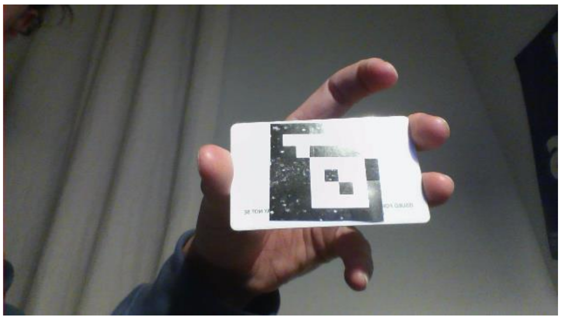

Figure 11: Effect of glare on Aruco marker

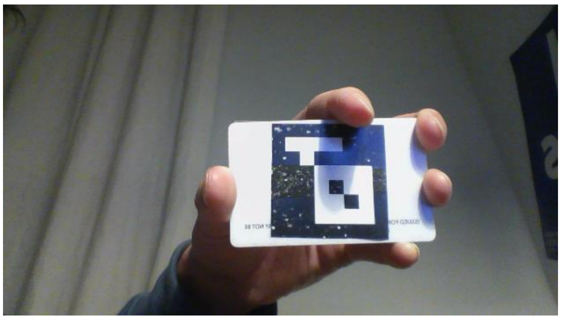

Figure 12: Effect of a small blockage of the Aruco marker

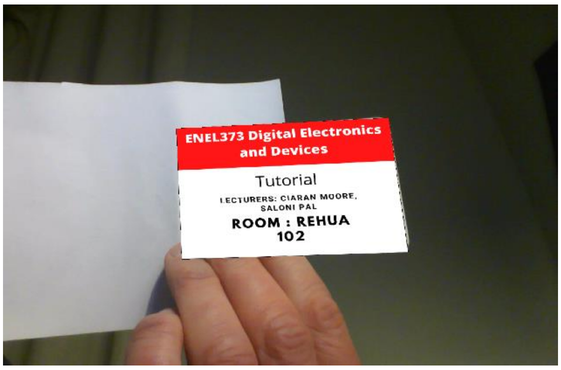

Figure 13: Markerless tracking registering ID with 50% coverage.

5. CONCLUSION

When creating a real time AR system, it is important that object detection and tracking is robust, effective, and immersive. Markerless AR is more immersive than traditional fiducial markers, and can still continue to track student IDs even when partially covered.

5.1 LIMITATIONS

Markerless tracking works well in a featureless environment, but when in a feature crowded area, the tracking often fails.

Markerless tracking also breaks down at distance, as cameras do not have enough resolution to see the details of the ID.

REFERENCES

[1] Z. Yovcheva, D. Buhalis and C. and Gatzidis, “Smartphone Augmented Reality Applications for Tourism,” Bournemouth University Business School, School of Design, Engineering and Computing, Bournemouth University, UK, 2012.

[2] Doxygen, “ArUco Marker Detection,” Open CV, [Online]. Available: https://docs.opencv.org/master/d9/d6a/group__aruco.html. [Accessed 30 Jun 2021].

[3] M. Workshop, “Augmented Reality,” CV Zone, [Online]. Available: https://www.computervision.zone/courses/augmented-reality/. [Accessed 30 Jun 2021].

[4] OpenCV, “Detection of ArUco Markers,” OpenCV, [Online]. Available: https://docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html. [Accessed 30 Jun 2021].

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers,” Elsevier, Computing and Numerical Analysis Department, Córdoba University, Edificio Einstein, 3a planta, Campus de Rabanales, 14071 Córdoba, Spain, 2014.

[6] K. A. Satoshi Suzuki, “Topological structural analysis of digitized binary images by border following,” Computer Vision, Graphics, and Image Processing, Volume 30, Issue 1, (https://www.sciencedirect.com/science/article/pii/0734189X85900167), 1985.

[7] D. Douglas and T. Peucker, “Algorithms for the reduction of the number of points required to represent a digitised line or its caricature.,” 1 Dec 1973. [Online].

Available: https://utpjournals.press/doi/10.3138/FM57-6770-U75U-7727. [Accessed 30 Jun 2021].

[8] I. Barandiaran, C. Paloc and M. Graña, “Real-time optical markerless tracking for augmented reality applications,” *Journal of Real-Time Image Processing,* vol. 5, p. 129–138, 2010.

[9] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” Computer Science Department, University of British Columbia, Vancouver, B.C., Canada, 2004.

[10] H. Bay, T. Tuytelaars and L. V. Gool, “SURF: Speeded Up Robust Features,” Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951., Springer, Berlin, Heidelberg, 2006.

[11] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” 2011. [Online]. Available: https://gaoyichao.com/Xiaotu/papers/2011%20-%20ORB%20an%20efficient%20alternative%20to%20SIFT%20or%20SURF%20-%20Rublee%20et%20al.pdf. [Accessed 30 Jun 2021].

[12] E. Rosten and T.Drummond., “Machine learning for highspeed corner detection,” *European Conference on Computer Vision,* vol. Volume I, 2006.

[13] M. Calonder, V. Lepetit, C. Strecha and P. Fua, “Brief: Binary robust independent elementary features,” *European Conference on Computer Vision,* vol. 1, 2010.