15-13

15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

# Robust Markers for Visual Navigation using Reed-Solomon Codes

Jayavardhana Gubbi, Sandeep N. K, Pavan Kumar Reddy K, P. Balamuralidhar
TCS Research and Innovation,
Bangalore, India
j.gubbi@tcs.com, sandeep.nk@tcs.com, pavank.reddy@tcs.com, balamurali.p@tcs.com

## Abstract

*Indoor navigation of unmanned vehicles in GPS denied environment is challenging but a necessity in many real-world applications. Although fully autonomous indoor navigation has been shown to work using simultaneous localisation and mapping (SLAM), its accuracy and robustness are inadequate for commercial applications. A semi-autonomous approach is an option for indoor navigation can be achieved using visual markers such as ArUco. The errors caused by motion of robots, visual artifacts due to change in environmental conditions and other occlusion will impact the reliability of visual markers. In this paper, a new robust visual marker based on ArUco with error detection and correction capability is proposed using Reed-Solomon codes. A dictionary of 50 symbols is generated and tested under different conditions with good results in detection and identification.*

## 1 Introduction

Automation is the way forward in many fields of engineering that will have impact on the way we transact with the real world. Most automation in the previous decades were driven by stationary robots in industrial setting, which increased the throughput substantially bringing down the costs. The next wave of robotic automation will include mobile robots that will have seamless interaction with the real world.

In outdoor scenarios, drones make use of their onboard global positioning system (GPS) to achieve the localization and path-planning. In indoor GPS denied scenarios, on-board cameras can be used for localization and path-planning. ArUco markers have been developed recently for augmented reality application building on other methods that include bar-codes, QR codes and other visual markers [1]. The focus of this paper is to develop robust markers that can work under occlusion and other image errors. These errors could be due to motion of the mobile robot, marker wear and tear, low light setting, and improper key-frame detection. In order to achieve this, the existing ArUco marker has been extended by adding check and recovery information using Reed Solomon (RS) codes [2, 3]. We have proposed the use of RS code in altogether a new way making the existing ArUco markers more robust and easier to read during navigation. As far as the authors are aware, use of RS codes in the way presented is not available earlier.

## 2 Visual markers

In many applications, visual markers are used for identification of the object of interest. Business community was the first to use visual markers (also called fiducial markers) widely in the form of bar-code and QR-code for billing and retrieval. Several augmented reality application including hand held video games work based on such markers. There are many visual markers proposed literature with both error detection and correction capabilities. For a comprehensive list, the reader can refer Garrido *et al.* [1, 4]. They include square shaped markers such as Vuforia, ArUco, AprilTag, ARToolkit, ARTag, Matrix, BinARyID, CyberCode, VisualCode, IGD, SCR, HOM, Bokode. Intersense and ReacTIVision are circular and blob based markers respectively [1, 5, 6, 7]. The error correction capability is limited in these markers. In this work, the focus is on square markers and its error correction capabilities.

Localization is very important in industrial setting as the final outcome of any application developed has to be precise. For instance, an object in Aisle 2, shelf 201 has to be detected at the same location; otherwise, the system can never be deployed. A recent survey on the use of such visual markers for indoor navigation has been presented by La Delfa *et al.* [8]. They found that AprilTag works marginally better than ArUco markers for small marker sizes and farther distances. However, in the case of drone navigation, the camera cannot be taken very close to the markers and hence, reasonably sized markers will be required. We chose ArUco markers due to the ease of implementation but the proposed method is easily applicable on AprilTag as well with suitable modifications. The results presented here are based on vertical positioning in order to avoid damage and soiling of markers if laid on the ground. As far as we are aware, no other system has been proposed in literature with error correction capability for navigation.

ArUco markers were proposed in 2014 primarily targeting pose estimation in Augmented Reality (AR) applications. These are square markers with one pixel thick borders that will ensure easy visual identification. There are two main steps in marker creation: a) create a dictionary of symbols ensuring maximum inter class distance - this makes the system fault tolerant to an extent; and (b) a method to decode the marker in real-time. Many available image processing methods are intelligently combined in order to identify the marker accurately. In this work, no modification has been made to this work flow except for two changes: (a) for boundary detection, we use a separate marker; (b) an additional segmentation for reading the Reed-Solomon codes that surrounds the marker. Examples of ArUco markers from our dictionary are shown in Fig. 1.

ArUco markers have error correction and occlusion

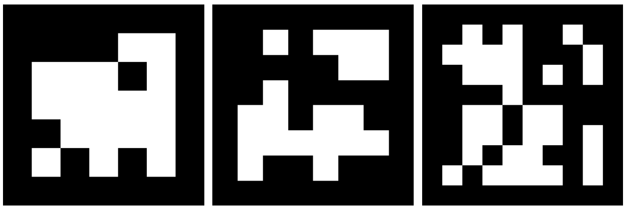

Figure 1: ArUco markers with $n = 5$, $n = 6$ and $n = 8$ from our dictionary for visual navigation.

handling capability. Error correction is performed by calculating the distance between the detected marker and the dictionary for all four possible rotation of the marker. Although this approach gives higher result compared to other methods such as ARToolkit (2 bit error correction), this is not technically similar to error correction performed in communication. In this paper, we propose a new method of *error correction in markers* using Reed-Solomon codes.

## 3 Robust visual markers

In many applications, visual markers should be robust to wear and tear, low illumination, motion blurring, reflections, and other common types of unnecessary effects. In communication theory, error detection and correction have been investigated in depth. Loss of packets in communication is often overcome using such error correction methods. Bose, Chaudhuri, and Hocquenghem (BCH) codes form a large class of powerful random error-correcting cyclic codes that is available in literature, which have abilities of multiple error correction. They are used in many real world application for transmission and storage such as CD, DVD, and satellite communication. They are binary error correcting codes and hence they were not ideally suited for error correction in the case of visual markers. Due to the fact that they are binary code, our attempt was to extend binary ArUco markers. The number of binary digits required for error correction are many and cannot be used for encoding purposes. ARToolKit and ARToolKitPlus use BCH codes in their markers. The focus of this work is non-binary error correction code. Reed-Solomon codes have been used in this study to make the ArUco codes robust, which is the first attempt to use non-binary error correction in visual markers.

### 3.1 Reed-Solomon codes

Reed-Solomon codes are algebraic error correcting codes that are widely used in communication and data storage. They are non-binary cyclic codes that are extremely easy to encode and decode in real-time. Let $p$ be the prime number and let $m \leq k \leq p$. The Reed-Solomon code over the field $\mathbb{F}$ where the message vector is of length $m$ and denoted by $[x_1, x_2, ...x_m]$. Let $P(t)$ be a polynomial of degree $m - 1$ given by:

$$P(t) = x_m t^{m-1} + x_{m-1} t^{m-2} + \cdots + x_2 t + x_1$$

Then the code vector $\mathbf{a}$ for the message vector is the list of first $n$ values of the polynomial $P(t)$:

$$\mathbf{a} = [a_1 \quad a_2 \quad ... \quad a_n] = [P(0) \quad P(1) \quad \cdots \quad P(n-1)]$$

The error correction capacity is given by the equation $n - m = 2t$ where, $n$ is the number of bits to be transmitted or stored, $m$ is the message length and $t$ is the number of errors that can be corrected. This means, in order to send $k$ digits of the message, RS will need $n = k + 2t$ digits. This will ensure $2t$ errors can be detected and $t$ errors can be corrected.

### 3.2 Modified ArUco marker with RS code for occlusion handling

The major contribution of this work is the use of RS code and the way RS code is employed in our marker. To maintain the simplicity of ArUco markers with additional robustness, we have modified the ArUco markers as shown in Fig. 2. The ArUco markers dictionary are generated using a standard procedure explained above. A $6 \times 6$ ArUco maker has a size of $8 \times 8$ including the border needed for marker detection in real-time. Each of the binary row is converted into a byte as shown. The 6 bytes are input into an RS encoder that generates two RS codes. The RS parity codes generated are converted back into binary and appended to the ArUco marker as shown in Fig. 2. The steps are repeated for all sides to generate RS encoded ArUco marker. For a $6 \times 6$ ArUco marker, a binary matrix of $13 \times 13$ is generated.

During the reconstruction phase, using the marker detector, we will end up getting two concentric squares that can be easily managed. However, when there is occlusion, the standard marker will not be able to decode as the boundary information is not available. Therefore, a unique way of representing the marker has been formulated. The encoding scheme ensures void areas are created in corners of the modified marker. In these positions, the system proposed in literature will yield a nested '$\sqcup$' representation in one of the directions due to the inner marker enclosure and the outer RS code enclosure. In order to improve the detection accuracy, a new representation is introduced at the corners as shown in Fig. 3a. It has relatively large circles on the corners and smaller circles right next to them. The circles at the corners act as border delimiters that improve the alignment of frames extracted when the marker is occluded. At least six circles are always visible so that the segmentation is accurate and reconstruction is feasible. In case the circles are invisible, the number of missing bits are huge enough for RS code to fail. A generalised Hough transform for circle detection is used on the binary image acquired for circle detection followed by marker detection. The steps to detect the occluded marker is summarised in Fig. 3b-d. After adding the new visual aid for detection, a binary matrix of $14 \times 14$ is generated.

If the entire square is visible, the standard decoding is applied and checked for errors. If part of the square is visible, it is due to occlusion. As the thickness of the marker units are known from the dictionary, missing or occluded information can be easily calculated. For two bytes of RS codes, one column of occlusion can be handled. As the information on the number of RS code is available *apriori*, it is easy to assess whether missing information can be recovered.

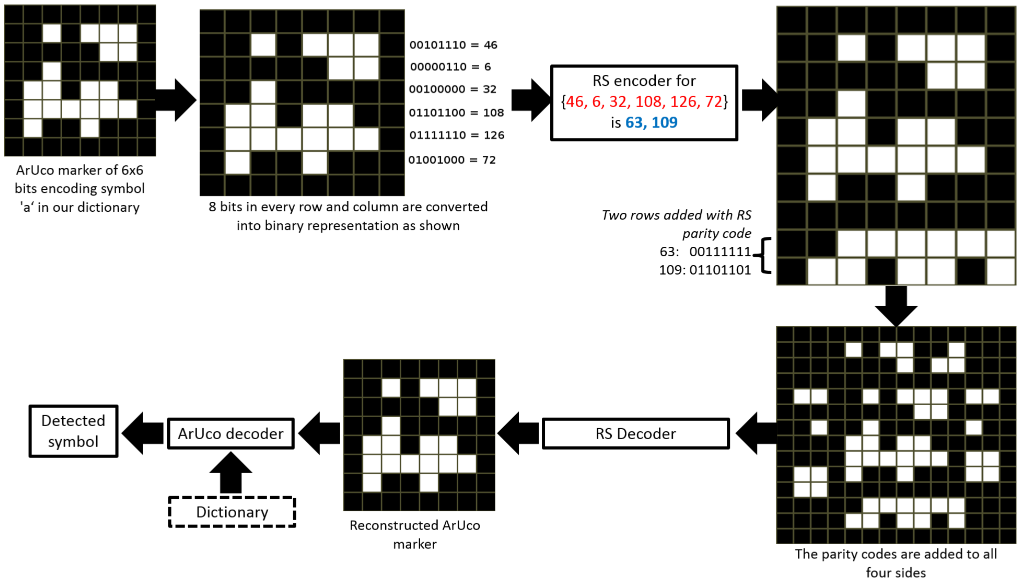

Figure 2: Flow chart of the proposed scheme

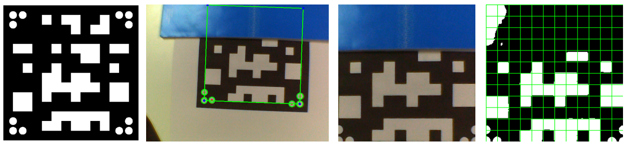
(a) (b) (c) (d)

Figure 3: Modified ArUco markers (a) Original marker in dictionary; (b) Occluded marker with circle detection; (c) segmented marker; and (d) overlaid grids for identification of marker

### 3.3 Experiments

The context of robust marker implementation is visual navigation using drones and other ground based robots. Consider a large setting where there is a lot of movement in products stacked. An example could be a warehouse or a retail store. Inventory management and stocktaking is a very tedious job carried out manually. Automatic stock taking using unmanned drones and robots during downtime of the facility will surely help the overall throughput. With the current trends in technology, a fully autonomous system is not feasible in realistic settings. A visual marker based navigation that can work in semi-autonomous mode within a GPS denied facility will aid in stock taking by unmanned systems. Key knowledge needed to update the centralised databases include localisation of the drone/robot, detection of articles and their placements. A marker based navigation will aid automatic localisation and article placement. Markers can aid in both these tasks very efficiently provided that challenges such as occlusion and wear and tear are addressed. This experiment is limited to building a marker for such purposes with added robustness.

Our dictionary consists of 50 symbols that include 26 alphabets (all capitals), 10 numerals and 14 commonly used symbols that is useful in asset identification and navigation. The size of the markers we considered was $6 \times 6$ and were printed as 3.5 *cm* markers. Two different configuration of RS codes were tested. The first one used two parity codes that could reconstruct one column (or bit) of error and the second one had four parity codes that could correct two columns (or bits) of error. It should be noted that the size of the resulting encoded matrix will increase substantially when we are adding more parity codes and the printing size of the marker should be increased accordingly. Examples with two parity codes and four parity codes are shown in Fig. 4

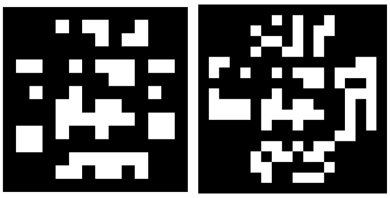

Figure 4: Modified ArUco markers with two parity codes (left) and four parity codes (right)

## 4 Results and Discussion

The first step in the process is accurate segmentation of the the markers in the presence or absence of occlusion. Fig. 5 shows the results of segmentation in the presence and absence of occlusion. As it can be seen, the results are good and sufficient for reconstruction and decoding of markers. Although the results were consistent for modified ArUco markers without occlusion, we found that there were some minor issues in achieving 100% segmentation for occluded scenarios.

Fig. 6 shows the results of reconstruction when the marker is occluded. The top row shows occluded input with varying illumination and the bottom row shows the detected marker. The markers were generated using two parity codes. It should be noted that ArUco markers themselves have some error correction capacity in them. However, due to occlusion, they will not be segmented using the scheme presented in Garrido-Jurado *et al.* [1].

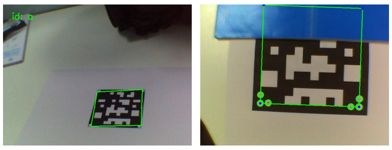

Figure 5: Results of segmentation of markers when there is no occlusion (left) and when there is an occlusion (right)

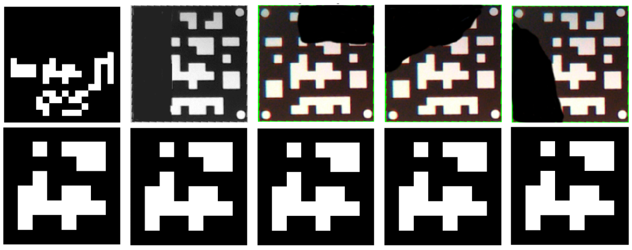

Figure 6: Results of reconstruction in the presence of occlusion

Another key aspect is in detection of the marker in real-time. In the original ArUco marker [1], the detection, identification and basic error correction time has been reported as 11.08 mSec. For the proposed markers the timing for five symbols randomly chosen are given in Table 1. Obviously, adding error detection and correction comes at a cost but the time taken for robust detection and identification is well within bounds for navigation application. The execution time was measured on a standard desktop system with Intel core i5-4590T processor and 4GB RAM. As it can be observed, the system performs better using circles as markers. This will not only perform faster but will also help in handling occlusion. The rectangular contour detection that is used in ArUco marker is the reason its lower performance in terms of time taken to decode [1].

The proposed marker can be used for real-time localization of the robot if the map of the indoor facility is provided. The approach is summarised in Fig. 7. As shown in the figure, the modified ArUco markers are stuck at strategic locations depending on the application, which are shown as black rectangles. The procedure will ensure accurate localization that is critical in indoor navigation.

## 5 Conclusion

A modified ArUco marker using Reed-Solomon code is proposed for visual navigation that has error detection and correction capability. A new detection scheme is proposed when a single marker is occluded improving the reliability and usability of the existing ArUco marker. The time taken to detect and identify the symbol is within tolerable limits for indoor navigation although it is higher than the standard ArUco markers. Finally, the usability of these markers in indoor localisation is discussed. The system is demonstrated by creating a limited dictionary of 50 symbols with and without occlusion. The results are highly promising and the versatility of the proposed technique to have higher error detection and correction capability is demonstrated using 6 × 6 markers.

Table 1: Processing times for five different symbols with and without occlusion

| Symbol | Using rectangular contour detection mSec | Using circle detection mSec |
|---|---|---|
| a | 92 | 28 |
| d | 94 | 27 |
| u | 96 | 65 |
| l | 99 | 27 |
| r | 97 | 28 |

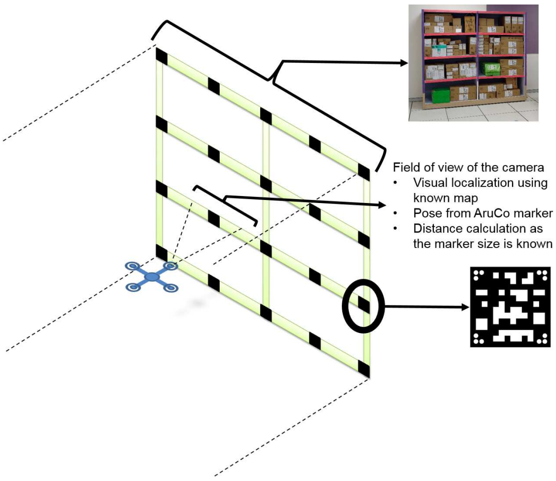

Figure 7: Schematic of localization approach that is achieved using a map of the facility

## References

[1] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and M. Marn-Jimnez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[2] S. Wicker, *Error Control Systems for Digital Communication and Storage.* Prentice Hall, 1995.

[3] X. Chen, *Error-Control Coding for Data Networks.* Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[4] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar, "Bokode: Imperceptible visual tags for camera based interaction from a distance," *ACM Trans. Graph.*, vol. 28, pp. 98:1–98:8, July 2009.

[5] A. Mutka, D. Miklic, I. Draganjac, and S. Bogdan, "A low cost vision based localization system using fiducial markers," {*IFAC*} *Proceedings Volumes*, vol. 41, no. 2, pp. 9528 – 9533, 2008. 17th {IFAC} World Congress.

[6] K. Tateno, I. Kitahara, and Y. Ohta, "A nested marker for augmented reality," in *IEEE Virtual Reality Conference, VR 2007, 10-14 March 2007, Charlotte, NC, USA, Proceedings*, pp. 259–262, 2007.

[7] T. Krajník, M. A. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent and Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.

[8] G. C. La Delfa, S. Monteleone, V. Catania, J. F. De Paz, and J. Bajo, "Performance analysis of visualmarkers for indoor navigation systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 8, pp. 730–740, 2016.