uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Gukhwan Kim

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical and Computer Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Designing Robust Fiduciary Markers for Indoor Localization

TITRE DE LA THÈSE / TITLE OF THESIS

Emil M-Petriu

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Gabriel Wainer

Abdulmotaleb Elsaddik

Azzedine Boukerche

Gary W. Slater

Le Doyen de la Faculte des etudes superieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Designing Robust Fiduciary Markers for Indoor Localization

by

Gukhwan Kim

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the Requirements for the Degree of

Master of Applied Science

In Electrical and Computer Engineering

June 2010

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering

(Electrical and Computer Engineering)

Your file Votre référence
ISBN: 978-0-494-73783-5
Our file Notre référence
ISBN: 978-0-494-73783-5

Acknowledgment

I wish to express my sincere appreciation to my thesis supervisor, Dr. Emil Petriu, for his support, encouragement, and continuous interest throughout the conduct of the research presented herein.

I wish to acknowledge the ROKA (Republic of Korea Army), University of Ottawa, and Dr. Emil Petriu for providing the financial support for this work.

I also wish all my gratitude to my friends and colleagues for the laughs, valuable discussions, supports. Thanks to Jongeun Cha, Jaewoo Jung, Seungwoo Jung, Martyn Johnson, Haeyoung Kim, Valentine Borsu, Fouad Khalil, and Altaf Hossain.

Finally, I wish to deeply thank my wife Soul Lee, my parents Jongdon Kim and Aeyoung Lim, my brother Junhwan Kim, and my parents in law Sangchul Lee and Sangrye Lee, for their unwavering moral support and for believing me.

Abstract

In this thesis, two fiduciary markers are newly designed as an alternate method for an indoor localization system. The main purpose of this work is to design more robust fiduciary markers than currently available ones in aspect of data restoration ability against data loss. In addition, achieving more accurate pose tracking performance is another goal of this work.

Currently available fiduciary markers may not be ideal for indoor localization due to their lack of robustness against occlusions. Thus, proposed fiduciary markers must provide better performance under occlusions with respect to a marker detection rate. By adopting digital encoding/decoding techniques combined with using multiple sub-markers or PRMVS pattern, proposed ones achieved a better marker detection rate under partial occlusions. In addition, the experimental results show that proposed fiduciary markers provide more accurate camera pose measurement than other fiduciary markers.

Table of Contents

List of Figures vi

List of Tables viii

Acronyms ix

1. Introduction 1

1.1. Scope and the Thesis outline 4

1.2. Organization of Thesis 4

2. Related Work 6

2.1. Various 2D Markers and Fiduciary Markers 6

2.2. Fiduciary Markers for Augmented Reality 14

2.3. Fiduciary Markers for Localization 18

3. Background 21

3.1. Principles of Vision System 21

3.1.1. Pin-hole Camera Model 22

3.1.2. Camera Calibration 26

3.2. Neural Network for Character Recognition 35

3.2.1. Neural Network and Perceptron 35

3.2.2. Backpropagation Neural Network 39

3.2.3. Character Recognition 42

3.3. Reed Solomon Codes for Error Correction 46

3.3.1. Classification of Reed Solomon codes 46

3.3.2. Galois Finite Field 47

3.3.3. Reed Solomon Encoding / Decoding, ... 51

3.4. Pseudo Random Multi-Valued Sequences and Fiduciary Marker 55

3.4.1. PRMVS for New Feature Points of Fiduciary Marker 55

3.4.2. PRMVS Pattern Codification 58

4. Designing Robust fiduciary Marker 61

4.1. Two Proposed Fiduciary Markers 62

4.2. Sub-Marker Detection in MTag 66

4.3. PRMVS feature points detection in CPRTag 77

4.4. Achieving Robustness against Data Loss/Error with RS/CRC coding..86

5. Experimental results 92

5.1. Experimental setup 92

5.2. Marker Detection Performance 94

5.2.1. False Positive Marker Detection 94

5.2.2. Inter-Marker Confusion 97

5.3. Marker Detection under Occlusion 103

5.4. Pose Estimation Performance 106

6. Conclusions 1144

6.1. Summary of the thesis 1144

6.2. Thesis contributions 1166

6.3. Suggestions for Future Research 1177

7. References 119

List of Figures

Figure 1. Various Bar-Code Markers ... 7
Figure 2. ARToolKit sample markers ... 10
Figure 3. Sample markers of the four types of fiduciary marker shown in [31] ... 12
Figure 4. ARTag marker samples ... 13
Figure 5. Video based AR technologies ... 14
Figure 6. Expedition Schatzseche with handheld device ... 17
Figure 7. Pinhole camera model ... 23
Figure 8. Perspective Transformation of an object in a scene ... 25
Figure 9. Checkerboard Pattern for Camera Calibration in OpenCV ... 34
Figure 10. Biological neurons and a perceptron ... 36
Figure 11. Differenct types of transfer function ... 37
Figure 12. Types of perceptron and the classification ability of each type ... 38
Figure 13. Feedforward Backpropagation Neural Network ... 39
Figure 14. Reed-Solomon code structural definition ... 46
Figure 15. PRMVS pattern generation sequence ... 59
Figure 16. An example of the generated 12 × 12 PRMVS pattern ... 60
Figure 17. Two proposed fiduciary marker systems ... 62
Figure 18. Block diagram of the MTag fiduciary marker system ... 63
Figure 19. Block diagram of the CPRTag fiduciary marker system ... 65
Figure 20. Four sub-markers used in the MTag shown in Fig. 19 ... 67
Figure 21. Block diagram of the sub-marker detection procedure ... 68
Figure 22. Some sample alphabet images for the ANN training ... 71
Figure 23. Marker detection and character extraction procedure ... 73
Figure 24. Alphabet recognition processing time of four different size images ... 75
Figure 25. Sub-marker configuration file content and the geometric relationships of Four sub-markers with respect to the MTag coordinate ... 76
Figure 26. 9 × 9 PRMVS pattern captured in a color image ... 79
Figure 27. Image processing procedure for the following code validation ... 82
Figure 28. An example of PRMVS code validation procedure ... 85
Figure 29. Structures of the two proposed fiduciary markers ... 88

Figure 30. Digital encoding process to create inner patterns 89
Figure 31. Digital decoding process to generate marker ID 91
Figure 32. Experimental setup with a laptop, a web camera, and a fiduciary marker 93
Figure 33. Hamming distance histogram obtained from the initially designed
Marker library 99
Figure 34. Hamming distance histograms of 4 different marker libraries obtained
with 4 pairs of RS and CRC polynomials shown in Table 6 101
Figure 35. Four marker sets of the 24 occluded marker images generated for the
marker detection experiment 104
Figure 36. Marker detection test with 24 occluded marker images 105
Figure 37. Incorrect marker detection results of the occluded ARToolKit markers 106
Figure 38. System errors of four marker systems obtained from the distance measurement
accuracy test with 9 distance variations and 6 angle variations 109
Figure 39-1. System error comparison on a specific angle between
four marker systems 110
Figure 39-2. System error comparison on a specific angle between
four marker systems 111
Figure 40. Distance measurement STD comparison of four fiduciary marker systems 112

List of Tables

Table 1. The filed elements for *GF(16)* with $p(x) = x^4 + x + 1$ 50

Table 2. Examples of the 12 numerical alphabet feature descriptions 72

Table 3. Alphabet recognition rate(%) with various image sizes 74

Table 4. 49 sub-pattern vectors with 9 color elements obtained by 3×3 windowing on 9×9 CPRTag..80

Table 5. Experimental results on false positive marker detection 96

Table 6. Polynomials used for RS / CRC checksum generations 100

Table 7. Number of marker pairs with a given hamming distance of 8 marker libraries...............102

Acronyms

Abbreviation	Description
GPS	Global Positioning System
SIFT	Scale Invariant Feature Transform
PRBA	Pseudo Random Binary Array
PRMVS	Pseudo Random Multi-Valued Sequences
ECC	Error Correction Code
RS	Reed-Solomon
DLT	Direct Linear Transformation
ANN	Artificial Neural Network
MLP	Multi-Layered Perceptron
CRC	Cyclic Redundancy Check

1. Introduction

Indoor navigation cannot use the widespread outdoor localization technique such as GPS (Global Positioning System) because of its obvious physical limitations, e.g. the satellites signals GPS is relying on are blocked or unreliable inside buildings [1].

Various types of indoor localization methods using ultra-sonic, infrared, and radio sensors have been reported in the literature [2]-[4]. The indoor positioning with these techniques requires an infrastructure to transmit and receive localization signals. In addition, the cost to deploy the transmitters to cover the most of the indoor environment could become high.

Computer vision techniques could provide, in many cases, a more cost effective and precise alternative for the indoor positioning system. These techniques could be categorized into two groups.

The first category of the vision-based indoor localization techniques analyzes environmental components and matches the result with pre-stored data [5]-[7], [40],[41]. The other uses various types of marker which is used for calculation of the position information [8]-[10],[17],[21],[22],[26],[30],[31],[33].

The SIFT(Scale Invariant Feature Transform) is the one of the famous algorithm belongs to the first category [11]-[12]. The SIFT feature points are invariant to scale changes, rotation, affine transformation, and illumination changes, making them independent of the camera viewpoints, and suitable landmarks for the localization.

ARToolKit is a frequently used fiduciary marker for the indoor localization system [22][26]. Although these markers are developed for camera pose tracking usage in Augmented Reality or Virtual Reality, some researchers adopted the fiduciary marker systems for an indoor localization purpose due to its pose tracking ability.

The vision-based positioning methods are generally lower in cost and have broader coverage than wireless systems, techniques using fiduciary markers seems more efficient. Since the techniques in the first group may not respond well to scene changes and require larger memory capacity and high computing cost in the state of art technology, they may not be proper to be used for the on-line navigation method. Beside, building a database which is used for analyzing the environment and matching results is time consuming and non-trivial tasks. For example, in order to employ the SIFT to the navigation system, the user needs to collect environmental images having enough feature points through the whole navigation route. In addition, each feature point is required to be assigned the location, scale, and orientation to form a database [13].

In [14]-[16], the authors discuss the marker-based optical indoor localization method with pseudorandom encoding for the absolute position measurement of automated guided vehicles. This method marks the full length of the path on the ground or wall. The vehicle travels along the pseudorandom encoded guide-path and reads the particular mark corresponding to the current position of the vehicle. By using pseudorandom encoding / decoding method, total number of bits to quantize the whole length of path could be effectively reduced [14]-[16]. However, since the vehicle should follow guide-path, the workspace of the vehicle is restricted.

To overcome this limitation, a vision-based two dimensional pseudorandom encoding method is also introduced later in [17]. In this method, the recovery of the absolute 2-D position index is based on PRBA (Pseudo Random Binary Array) window property. The PRBA pattern could be deployed onto the environments such as walls, ceilings or floors to recover the absolute position. In addition, the PRBA pattern can be used for model driven 3D pose estimation when the pattern attached to surface of any object. Pseudo Random Multi-Valued Sequences (PRMVS) is an improved version of PRBA. The PRMVS exploits the window property of the pseudo-random sequences generated by feedback shift-registers using primitive polynomials defined over Galois filed [18][19]. Since it costs a lot and harms the atheistic appeal to attach the PR patterns on the all the possible route of the navigation system, it might not a good approach. Thus, in this thesis, a new way to utilize the PRMVS for a fiduciary marker is proposed and it would be relatively easy to produce, manipulate, and deploy the PR pattern.

Fiduciary markers are 2D planar markers having special pattern which the computer vision algorithm can detect. ARToolKit, ARTag, and QR-Code [20] are some of currently available fiduciary markers. As the name of markers show, the fiduciary markers are widely used in Augmented Reality (AR) since the marker tracking system can provide the 6 DOF pose of a marker, which are critical parameters for the AR. Several researchers have been implemented the indoor localization system using the currently available fiduciary markers [21]-[23], however, since the enrolled markers in their system are originally designed for the AR system, the markers does not function well. As in the environments of indoor localization system there are more chances that markers are occluded or partially hidden by pedestrians or other objects, the markers for the navigation system should be more robust than the AR markers in some aspects.

Most of the AR markers consist of a rectangle outer black boarder and inner data region. The outer boarder provides the information of the marker existence to the computer vision algorithm, and inner data area provides the marker ID information. As the outer rectangle black boarder is used for the detection of the marker, when the boarder is occluded the marker detection fails, and consequently, the marker ID information cannot be recovered from the inner data region. According to the author's research, only ARTag has a prediction algorithm to overcome the situation. Thus, proposed fiduciary markers must have better data restoration and prediction algorithm for indoor localization purpose as well as more accurate camera pose measurements.

1.1 Scope and the Thesis outline

The objective of the research is to design robust fiduciary markers which can be used for an indoor localization system using a consumer grade web camera and to compare the performance of proposed markers with currently available markers. The tasks and challenges are as follows:

- Compare the features of the currently available markers
- Define the feature points of the new marker
- Design the new marker for the indoor localization system
- Implement the marker template detection and prediction algorithm
- Implement the error correction algorithm against the data loss or data corruption
- Compare performances of proposed markers with other fiduciary markers with respect to data restoration against data loss or data corruption and camera pose measurement accuracy

1.2 Organization of Thesis

Chapter 2 introduces various fiduciary markers and its applications.

Chapter 3 presents background knowledge associated with design of robust fiduciary marker systems for indoor localization.

Chapter 4 discusses how the two proposed fiduciary marker systems are designed and operated while achieving relatively reliable characteristics for the indoor localization comparing to the currently available fiduciary markers.

Chapter 5 presents results of fiduciary marker detection and localization experiments conducted with two proposed fiduciary marker systems.

Chapter 6 summaries the work describes in previous chapters. In addition, possible future work and suggestions are discussed in this chapter.

2. Related Work

This chapter introduces various fiduciary markers and its applications. The concept and need of the fiduciary marker are shown. Related works in the fiduciary marker design, AR applications, and localization applications are summarized.

2.1 Various 2D Markers and Fiduciary Markers

The 2D markers can be categorized into two groups, Bar-code type 2D markers and fiduciary markers. One dimensional bar-codes have been used to automate the identification of products since 1970s. However, its small data capacity limits their usefulness for the direct transmission of business data in machine readable form. This is obvious since the one dimensional bar-codes convey the encoded information only in the horizontal direction.

As new tagging methods to convey more information on their stock arose in many businesses or applications, the 2D bar-codes which utilize the vertical space as well as the horizontal space to encode more data have been developed. The conventional linear 1D bar-code is usually read by active laser scanner but the 2D bar-code is read by passive camera instead of the laser scanner. They are originally intended to carry information, not to localize whereas the fiduciary markers provide pose information which is essential in the AR applications [24]. Fig. 1 shows the common 2D bar codes.

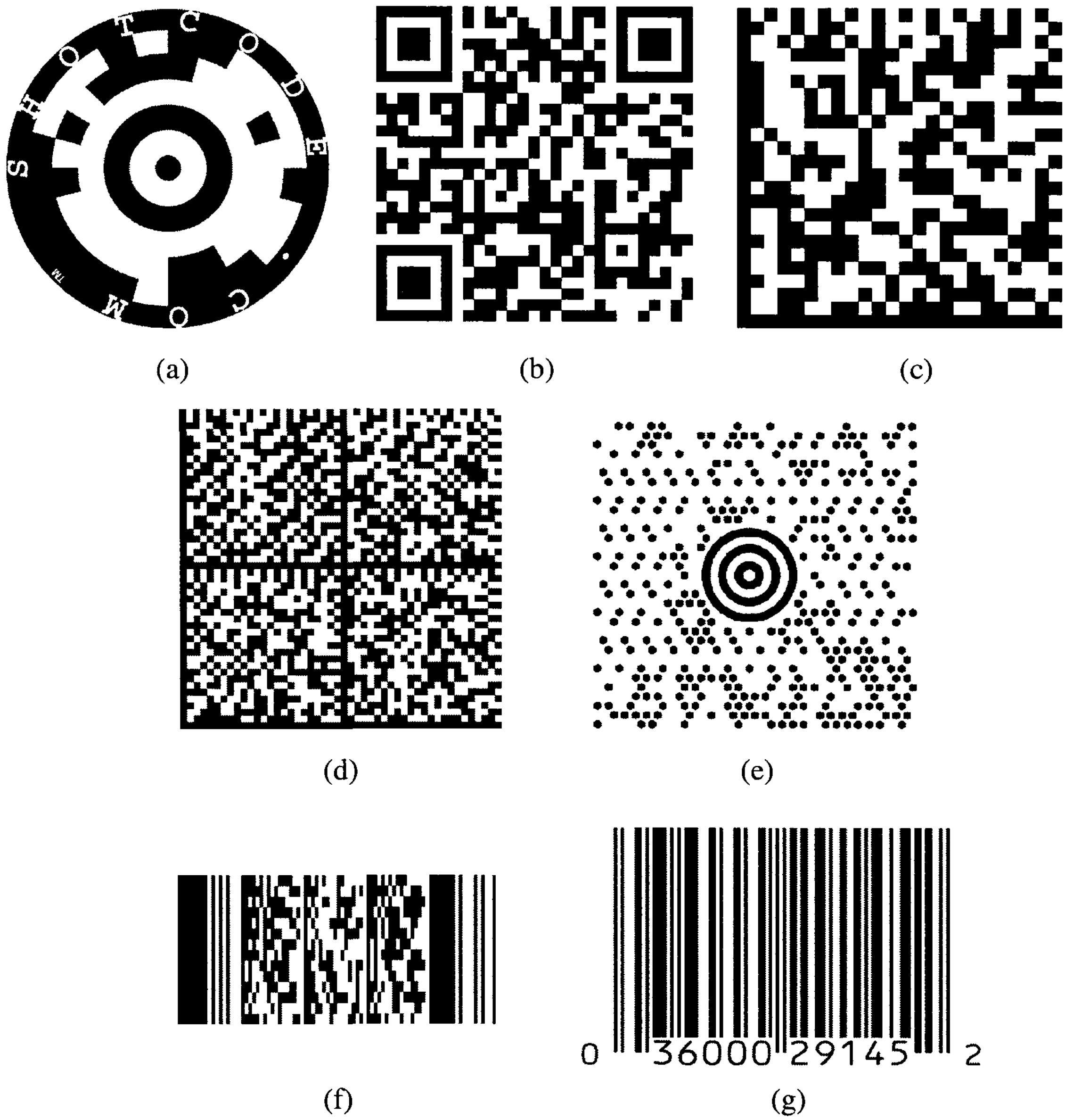


Figure 1. Various Bar-Code Markers: (a) ShotCode (b) QR-Code (c) SemaCode (d) 2D DataMatrix (e) Maxicode (f) PDF417 Code (g) Conventional 1D Bar-code

US Postal Service uses the Maxicode to convey shipping information. QR Code (Fig. 1(b)) and Data matrix (Fig. 1(d)) are two other examples designed to contain information on industrial settings for part labelling. The QR Code has been gaining popularity in Asian countries such as Japan and South Korea. Three types of 2D bar-code described above use provision for error correction codes (ECC) to recover its data when some of the bits are incorrectly read. Data matrix used ECC200 standard and both of the QR Code and the Maxicode use Reed-Solomon Code.

The 2D bar-codes are becoming widely used in industry for classification and identification of goods. Most of them share the data encoding procedure using binary values for reflectance. This is because the bi-tonal reduces the number of decision making steps per pixel, that is, there is no worry for gray shade of the data bits. This also reduces the lighting and camera sensitivity requirement and removes a need for linearization [24]-[26]. Some parts of the data bits in the bar-code patterns are used for an error detection and error correction so even in the occlusions of bar-codes or loss of data bits during dara reading, the information could be retrieved.

However, since the 2D bar-code, which is the first category of the 2D marker, is not originally intended to be used for the localization, therefore it might not be useful for the fiduciary marker systems. The fiduciary marker used for the localization should provide at least four distinct feature points easy to be detected but the 2D bar-codes usually do not have enough distinct feature points. In addition, in order to correctly decode the marker information from the pattern data bits, the size of the 2D bar-code is required to be large enough in the image. Otherwise, the range of the bar-code detection would be very limited [24].

The other category of the 2D marker is the fiduciary marker. The fiduciary markers are useful in many situations where object recognition or pose estimation is needed with a high reliability, where natural features to track are insufficient, and where it is hard to deploy other types of beacons [27]. The fiduciary markers are added onto the environments for a robust feature point detection in computer vision systems. They are deployed in the physical environments so that the 6 DOF pose of a moving camera can be computed or they may be attached on moving objects so that the 6 DOF pose relative to either a fixed or moving camera can be calculated [32]. Example applications using fiduciary marker systems include Augmented Reality (AR), position tracking, photo-modeling, and robot navigation. Recently, even a commercial internet shopping website (tobi.com) uses an augmented reality application which can superimpose fashion items onto an image of a customer captured by a web-cam. This AR application, 'Fashionista', is developed by an advertising company 'Zugara' and uses a fiduciary marker to obtain a user's pose in the captured image.

A fiduciary marker system consists of some unique patterns and feature points along with the localization and identification algorithms vital for marker detection and pattern analyzing in computer vision system. Such system should reliably report four or more points per marker when it is seen in the image. The patterns should be distinct enough so that the marker is not confused with the environments. Ideally the system should have a library of many unique markers that can be distinguished one from another. The image processing should be robust enough to find the markers under various lighting conditions, image noise and blurring, unknown scale and projection, and partial occlusion. Preferably, the markers should be passive planar patterns for a convenient printing and mounting, and should be detectable with required size pixels to maximize the range of usage [24]-[28].

The detected feature points are used to define an interested image region, that is, the information area of the marker. Various lighting conditions, cluttered scenes, objects similar with the shape of the marker and unfocused camera are the typical causes of the false detecting of the feature points and false recognition of the patterns. A non-robust marker detection system providing such a false reporting may lead a severe problem in a higher level procedure in the system. For example, in AR application, if the image feature points do not correspond well with the marker feature points, the rendered virtual objects in the scene would flicker or be unstable, or wrong object would appear.

The most common fiduciary marker system is ARToolKit. The ARToolKit markers are shown in Fig. 2. In recent years, many augmented reality (AR) and human computer interaction (HCI) projects have used ARToolKit markers which consist of a square outline with an internal pattern identified by correlation [24][28]. This trend is due to its available open source code and well documented references.

The bi-tonal markers are recognized by detecting the marker's black borders, that is, finding connected groups of pixels below a certain gray level threshold once the input image is converted from RGB or RGBA into GRAY scale image. Then the contour of each group is extracted, and finally those blobs surrounded by four straight lines are marked as potential markers. The four corners of every potential marker are used to calculate a homography in order to remove the perspective distortion. The internal pattern of a marker

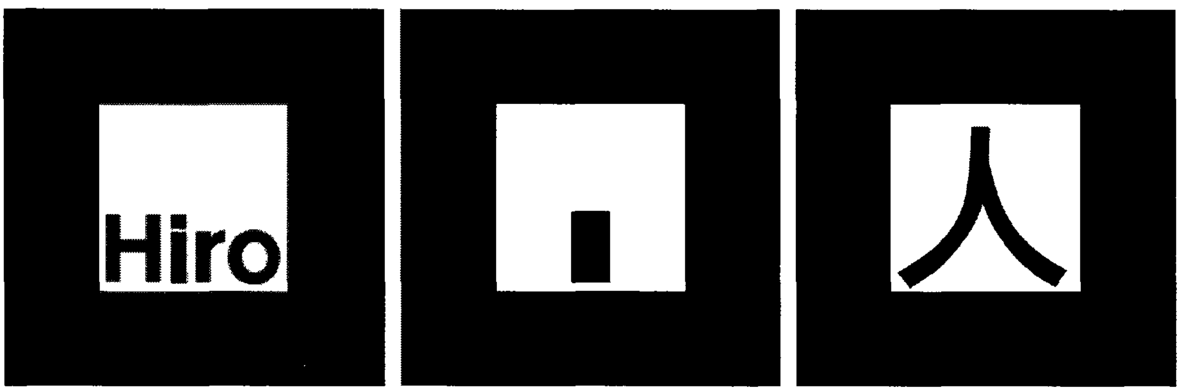


Figure 2. ARToolKit sample markers. *ARToolKit provides open source code so that users can register their own markers. However, if the total number of registered markers is increased, there are more chances for false detection.*

is then unwarped from the input image so that the sample grids of N by N gray values inside are extracted for pattern recognition. The gray values form a feature vector that is compared to a library of feature vectors of pre-registered markers by correlation method. The output of this template matching is a confidence factor. If this confidence factor is greater than a certain threshold, a marker has been found. Most of other fiduciary marker systems use similar detection method, which is detecting a black border with four straight lines, but the pattern recognition schemes are various in many systems. Since the ARToolKit detection is based on a highly simplified template matching algorithm that compares three of the geometry invariants of the marker region with those of patterns pre-registered in the system, the ARToolKit tends to falsely report marker IDs under the situations of existence of multi-markers [24]-[26],[29].

Many researchers have done experiments in purpose to improve the accuracy of ARToolKit marker detection algorithm [29]-[31], however, other fiduciary fiduciary marker system does not have much references on its accuracy test. This is in reliance on the usability (open source code and well documented references) of the ARToolKit. Zhang [31] and his companies have tested four types of fiduciary marker, ARToolKit (ATK), Hoffman (HOM), Institut Graphische Datenverarbeitung (IGD), and Siemens Corporate Research (SCR) marker systems (shown in Fig. 3). In this experiment, they evaluated the marker systems focused on the following four properties: usability, efficiency, accuracy, and reliability. These four major properties describe performance, advantages, and disadvantages of marker tracking systems. Usability describes whether the system is easily integrated in user's applications. Efficiency can be evaluated by computing the tracking time performance of the system. Accuracy is defined as how accurate the marker system estimates the location of the feature points of the marker. Although they tested different

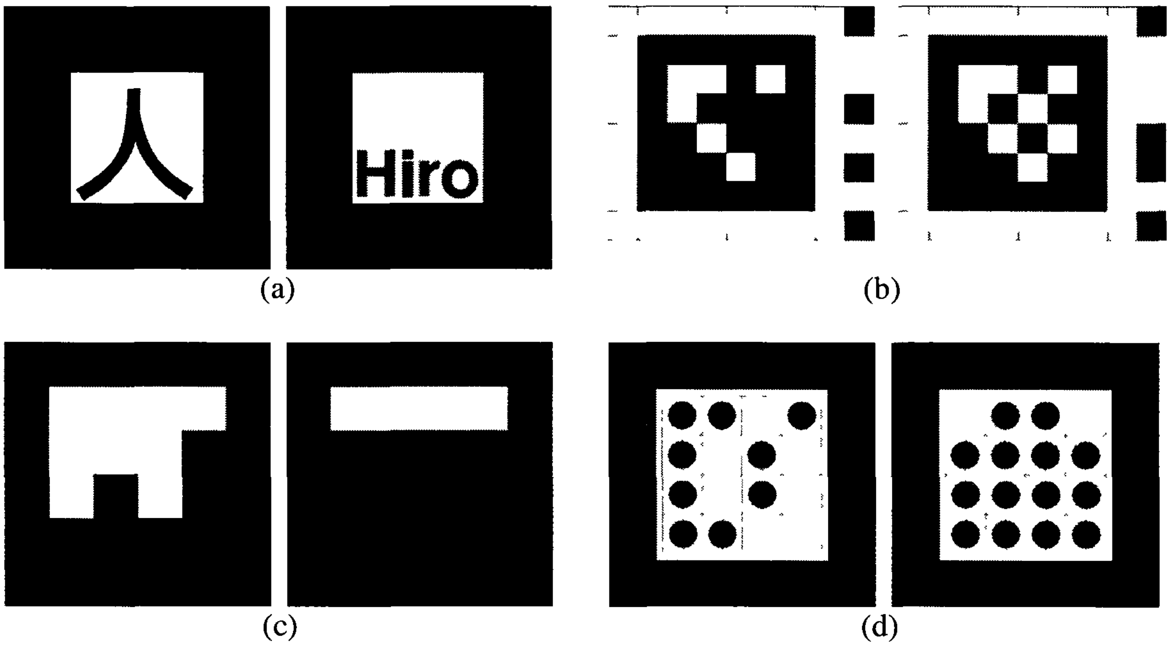


Figure 3. Sample markers of the four types of fiduciary marker shown in [31].
(a) ARToolKit (b) HOM (c) IGD (d) SCR

types of fiduciary markers, they mentioned that the work was not intended to rank the marker systems since the usability of the each marker system depends on what application the user utilizes it for. However, according to their work, the ATK showed the lowest performance while the HOM ranked the highest in the multi-marker situation and ATK performed well in badly focused image while the IGD did not.

When we consider the indoor environments, objects in sight may be confused with markers, and many pedestrians or obstacles may occlude markers. Thus the fiduciary marker for indoor navigation system should be robust against marker occlusion. Since the four types of fiduciary marker system above do not provide the immunity against the occlusion, new type of fiduciary marker system equipped with "Guessing" or "Predicting" function under the occlusion is needed for the indoor navigation system.

ARTag [20][24][25] is the one of the recently developed fiduciary marker system by Mark Fiala, who was a researcher in National Research Council of Canada, and it has

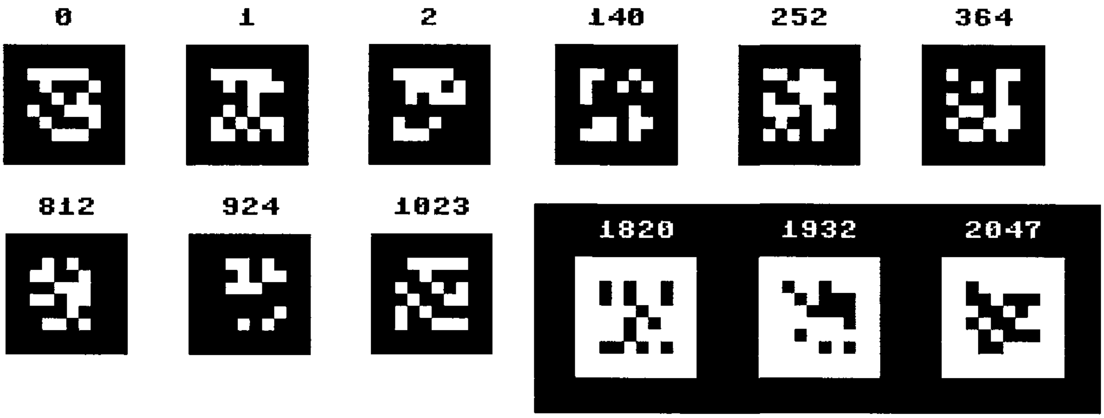


Figure 4. ARTag marker samples. *ARTag markers are bi-tonal marker patterns consisting of a square border and a 6 by 6 interior grid of cells representing logic '0' or '1'.*

gained popularity in the AR field since it accommodates some robust features above four types of system does not provide. Fig. 4 presents the ARTag markers. The inventor describes the strength of the ARTag as below [32]

- It runs in near real-time.
- It reliably finds markers in images.
- ARTag markers can be partially covered and will still be detected.
- The markers use reliable digital coding techniques, so the chance of falsely detecting an ARTag marker or confusing one for another is very low.
- ARTag has a large library of markers (2002), so one can use them on many objects.
- The system can handle varied lighting situations.
- ARTag can be used in arrays. The ARTag library combines the pose information from multiple markers.

Above listed features of the ARTag make it attractive to be used in the indoor localization system as well as AR applications. Especially, the robustness against the occlusion is essential property for the indoor navigation marker system as stated in previous discussion. However, the total number of available markers in the ARTag is 2002, and this might not be enough to cover large indoor area. Thus a new type of fiduciary marker system with a larger library than that of ARTag is still needed.

2.2 Fiduciary Markers for Augmented Reality

Augmented Reality has increasingly received an intense media attention for recent decades. Since Ivan Sutherland has created a working prototype of what is widely considered as the first Virtual Reality (VR) and AR systems at 1968 [32], various applications have been introduced to the world and some of those are commercially available now.

The AR technology that enhances a physical space with computer generated information and overlay virtual objects onto real world seems to have a wider range of applications than conventional VR [33]. Such applications utilized with AR include instructing repair or manufacturing, in-patient visualization of medical data, and annotation [34][35][36]. Besides adding objects to a real environment, Augmented Reality also has the potential to remove them. Current work has focused on adding virtual objects to a real environment. However, graphic overlays might also be used to remove or hide parts of real environment from a user.

A basic design concern in building an AR system is how to combine real and virtual world. Two basic choices are available: optical and video technologies. A see-through Head Mounted Display (HMD) is one device used to combine real and virtual scene, that is, it allows a user to see the real world, with virtual objects superimposed by optical or video technologies. The other type of technology, video based AR, combines the real world and the virtual world in the image processor and projects the augmented reality onto the display screen. In [32], the author describes two AR paradigms belong to the video based AR (See Fig. 5).

The first paradigm of the video based AR is named 'Magic Mirror.' The technique involves putting a computer monitor or a project screen behind the area captured by the AR video camera. The display is a mirror image which displays the augmentation combining the real world and the virtual world. The other is named 'Magic Lens' and it allows a user to see the real world augmented with virtual world through a lens such as handheld computer or mobile phone screen.

Fiduciary markers are playing an important role in the AR field described above One of the most basic problems currently limiting the performance of Augmented Reality applications is the registration problem. The objects in the real and virtual world must be properly aligned with respect to each other, otherwise the illusion that the two worlds coexist will not be compromised. For the accurate registration, it is required to measure

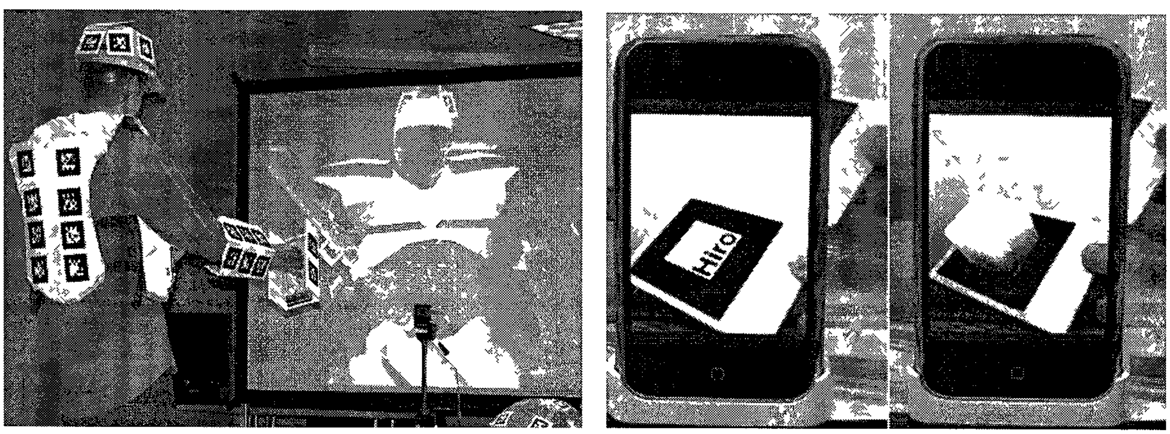


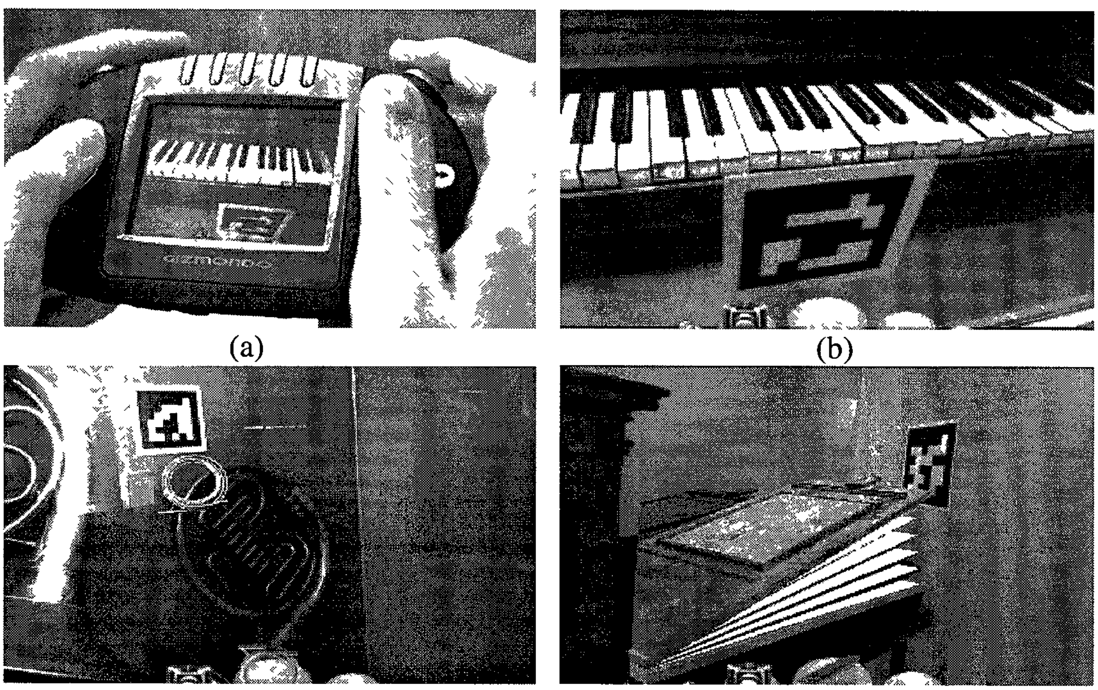
Figure 5. Video based AR technologies. *Left : Magic Mirror technology on handheld device with a ARTag marker, Right : Magic Lens technology with ARToolKit markers*

camera motions corresponding to user's view precisely.

The registration methods are divided into a sensor-based approach and vision-based approach. By using some sensors such as magnetic or gyro, the registration can be stable against a various illumination conditions and rapid motion of cameras [38],[39]. However, the rotations and translations of the camera obtained only from those sensors are not accurate enough to achieve complete geometric registration of virtual objects in AR. Furthermore, those sensors might limit the users working space or be affected by other environmental parameters.

On the other hand, the vision-based registration is cost effective approach only required to install one or multiple cameras [37]. This approach is divided into marker based registration and natural feature based registration. Natural feature based approach might be the ultimate destination of the AR registration method since it does not need to install any artificial features in the environments and to be equipped with special sensors except the camera. However, designing and building such an algorithm in a real time is not easy task, and in fact, the feature points distinct enough to track in nature are not common [40][41]. In contrast, the marker-based approach allows the user deploy the artificial markers in the environments as many as the application needs and a robust marker tracking algorithm can be developed. These advantages have brought the popularity of fiduciary markers such as the ARToolKit in recent years.

One good example of the AR applications with the fiduciary marker is well described in [42]. Handheld Devices (HD) seem to be a superior alternative for AR especially for untrained users in unsupervised environments. They are more convenient than the HMD because the users are comfortable to operate them due to the advent of the smart-phones. In addition, the wide spread of the smart-phone makes it possible to bring AR into our daily lives. The author of the [42], Dieter Schmalstieg, has created a software framework of AR, *Stdierstube ES*, for ultra light devices. In 2007, he implemented a AR museum application, named *Expedition Schatzseche*, for Carinthia State Museum and is now permanently available there for booking by high school visitors (See Fig. 6). This type of application can be categorized into 'Magic Lens' of the video-based AR. In this application, the fiduciary marker provides translation and rotation parameters of the camera with respect to the marker. The parameters can be computed from the pre-defined feature points of the marker, that is, the four corner points of the black marker border. Then, the pattern inside of the marker provides information of the later action which the AR application should do. Thus, how accurate the marker detection and tracking algorithm is will decide the performance of the AR application.



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

Figure 6. Expedition Schatzseche with handheld device. *(a)Captured scene via HD (b) Scene with a fiduciary marker. (c)(d) Rendered Scene : the fiduciary markers provide the rendering information to the image processing algorithm*

2.3 Fiduciary Markers for Localization

In the any kind of location-based system, the user or the automated system should be able to obtain accurate position data in real time. Furthermore, these systems should work well over large areas, be easy to setup, and be cost effective.

For outdoor environments, the Global Positioning Systems (GPSs) are common and have been used in a variety of regions. However, the satellite signals could be blocked or highly corrupted in indoor environments and even in outdoor environment with dense skyscrapers. Thus many alternative indoor tracking systems have been proposed but each of them comes with its own problems.

Many kind of indoor positioning systems use infrared, magnetic, or other wireless communications with transmitters and receivers capable of sending and detecting those unique signals. Although those systems can provide accurate position information, their range is limited due to the install cost or environmental constraints. In addition, the set up and calibration of such system is non-trivial.

In order to avoid these limitations of the systems, computer vision has been used widely as this system does not need heavy equipments such as signal transmitters, power lines for signal transmitters, and receivers. The computer vision localization only needs an image processing tool and one or multiple cameras.

The vision based localization method can be divided into a marker based and a natural feature based localization as the vision based AR registration method is. As described in Section 2.2, the natural feature based approach might be the final destination of vision based localization since it is the way human being track themselves. However, for now, it is a more challenging method than the marker based approach. On the other hand, detecting and decoding artificial markers is more robust and works better under uncontrolled lighting condition and with less computational efforts.

The marker based localization system can provide 6 DOF pose of the marker with respect to the camera coordinate or the pose of camera with respect the marker if the marker pose is already known. This is possible because the marker has distinct feature points with known position information and the marker tracking algorithm can compute the homography matrix contains the translation and rotation information between the marker and the camera by using those unique feature points.

In [9], [21], [22], and [26], examples of the indoor navigation application are presented and three of them use ARToolKit markers for the localization. The markers are deployed on any 'hot spot' of the indoor environments so that the user or automated system with a camera can move around while detecting the markers. In other case, the markers are attached on a moving user or an automated system while the fixed camera detecting the markers. Either way, the image processing algorithm computes the 6 DOF pose of the camera or the marker and let the system reliably maintain the location information unless the tracking fails.

As described before, the ARToolKit marker detecting algorithm uses simple template matching correlation method to find the pre-registered marker through the scene. If the total number of the registered marker is increased, the false negative or false positive marker detection would occur more frequently as the similarity among markers is increased. Thus the system needs to reuse the sets of markers within the given environments to decrease size of a marker library [9],[22].

This is the problem of not only the ARToolKit markers but also the other currently available fiduciary markers. The ARTag, another frequently used marker system, only has

2,002 markers and the ARToolKit Plus [21],[42] has 4,096 markers. Since the library of the marker may not big enough to cover the inside of a building, the limited number of markers of the currently available marker systems is another problem should be solved for the indoor localization.

The main difference between marker based AR application and Localization application is where the event happens. In the case of AR, the user can keep preventing the marker from occlusions by moving their camera or removing any objects hiding the marker. In the case of indoor localization, however, the user might not be able to remove the pedestrians or objects hiding the marker. For the fully automated indoor localization system, the situation is even worse. Thus the marker should be robust against the partial occlusion and the data loss.

Many scholars have invested their efforts on designing reliable marker systems. However, it is hard to find fiduciary marker system armed with robustness against the partial occlusion. Some of the currently available marker systems have error correction codes in their pattern to recover data loss or data corruption for pattern recognition. However, if the markers are not detected due to the occlusion on the markers, the error correction technique would be wasted. Therefore, the marker system for indoor localization should have immunity for the occlusion as well as data loss or data corruption.

3. Background

In this chapter, the background knowledge associated with the designing robust fiduciary marker for the indoor localization is presented.

Firstly, the principles of the vision system, especially basic of the camera calibration are summarized. Since the calibrated camera can provide more accurate geometric information of the captured objects in the image, this is a pre-requisite task.

Secondly, 'Back-Propagation Neural Network' scheme is presented since the proposed fiduciary marker uses the neural network as a marker pattern template matching algorithm.

Thirdly, Reed Solomon Error Correction Code (RS-ECC) is introduced. The RS-ECC is a digital error correction technique widely used for the digital communication. Here, the proposed marker uses RS-ECC encoding/decoding technique for achieving robustness against data loss or data corruption.

Finally, the Pseudo-Random Multi-Valued Sequences (PRMVSs) is briefly explained to show how the PRMVS works in the fiduciary marker system. Since the PRMVS provides unique feature points when the pattern is presented in a scene, it can be adopted in the fiduciary marker system.

3.1. Principles of Vision System

This part summarizes the camera properties and the camera calibration procedure essential for accurate image processing. Since most of the camera does not capture the real world as it is, that is, the image captured by the camera is distorted by the non-perfect lens or structural problems of the camera, the image should be corrected before further processing. The image correction can be achieved when the associated camera parameters are known. The camera calibration is all about how to compute those associated camera parameters. Before discussing the camera calibration procedure, the camera model should be defined to derive the required parameters with respect to the image correction. The OpenCV provides a set of functions of the camera calibration and the procedure to compute the camera parameters is described in end of this sub-chapter.

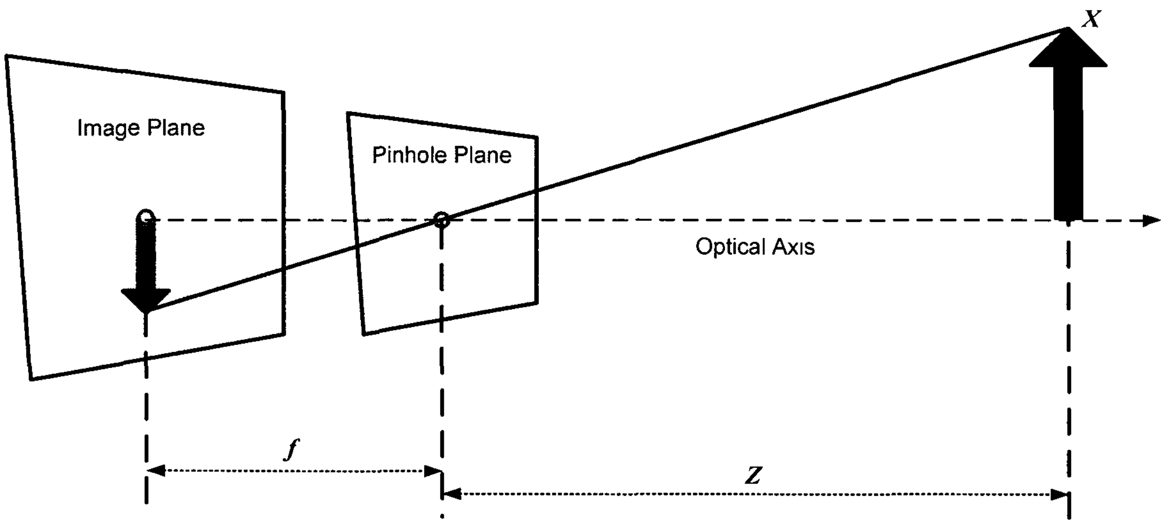
3.1.1. Pin-hole Camera Model

The way a camera sensor detects the light emitted from an object imitates how a human's eye works. The concept of how the eye or camera detects objects or a scene has been constantly and deeply researched for several centuries. The 'Pinhole' camera model is a relatively simple and useful model to examine the phenomenon or the concept. The 'pinhole' is a virtual plane having tiny (pin) hole on the middle of it, and the light only can get through the tiny hole.

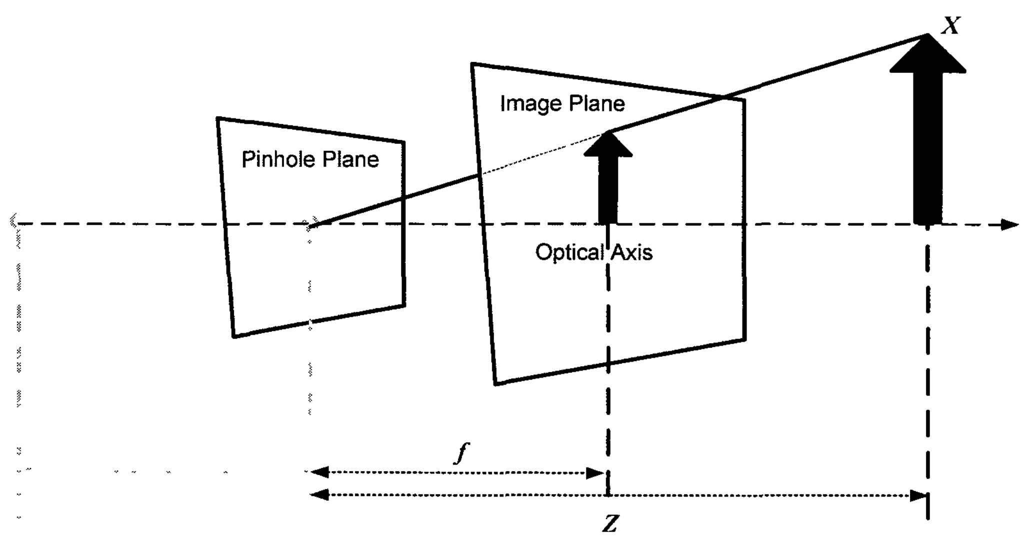
Most of the cameras do not use the pinhole model but they use lenses to absorb enough lights in a short moment. The camera with lens, however, has complex structure and distorts the original image due to its non-perfection. The pinhole camera model provides the basic concept of the camera and the more complex model contains other parameters associated with the camera lens can be derived from the pinhole model.

The pinhole camera model is the perspective transform which assumes that any point imaged in a three dimensional viewing volume projects through a camera focal point (pinhole) onto an imagine plane. The image plane is normal to the focal axis. Fig. 7(a) illustrates the pinhole camera model. The focal length is denoted by f and Z denotes the distance between the pinhole plane and the real object. The x represents the object size in the image plane and the X is the object size in the real world. The object size x in pixels can be computed by

$$-x = f\left(\frac{X}{Z}\right). \tag{1}$$



(a)



(b)

Figure 7. Pinhole camera model. *(a) Original pinhole camera model. (b) Modified pinhole camera model for mathematical convenience : the image plane is moved in front of the pinhole plane.*

This equation could be modified with the re-positioned image plane in the pinhole camera model (See Fig. 7(b).) This modified pinhole camera model, which is mathematically equivalent to the physical setup of the Fig. 7(a), provides the mathematical simplicity as well as better comprehension. By this modified model, the eq. (1) is changed as

$$x = f\left(\frac{X}{Z}\right). \tag{2}$$

In fact, the center of the image plane does not exactly correspond with the camera principal point, which means the object, the center of an image, and the focal point are not collinear, and making those three points collinear is almost impossible. Thus in order to denote the offsets of the image center point, two new parameters, c_x and c_y should be introduced. Then, the any 3D point in the real world would be projected on the image plane by

$$x_{screen} = f_x\left(\frac{X}{Z}\right) + c_x, \qquad y_{screen} = f_y\left(\frac{Y}{Z}\right) + c_y. \tag{3}$$

where

$$f_x = s_x \times f \quad \text{and} \quad f_y = s_y \times f \tag{4}$$

S_x : number of pixels in unit length in x direction of the image sensor

S_y : number of pixels in unit length in y direction of the image sensor.

Basic Projective Transform

The relation between a projected 2D point on the image plane, $q = [\ x\ \ y\ ]^T$, and a 3D point in the real, $Q = [\ X\ \ Y\ \ Z\ ]^T$, is called 'projective transform.' For the mathematical convenience, the homogeneous coordinate of the image point q which is the augmented vector by adding scalar w as the last element is used. Thus the basic projective transform matrix could be expressed as

$$\tilde{q} = MQ, \quad \tilde{q} = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, \quad M = \begin{bmatrix} f_x & 0 & c_y \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5}$$

where matrix M is camera intrinsic matrix and f_x, f_y, c_x, and c_y are the intrinsic parameters.

Rotation matrix and Translation vector

Any 3D point in the world coordinate can be expressed as a 3D point in the camera coordinate with a 'Rotation matrix' and a 'Translation vector.' A rotation of a 3D coordinate is equivalent to three sequential 2D rotation of the 3D coordinate as shown below

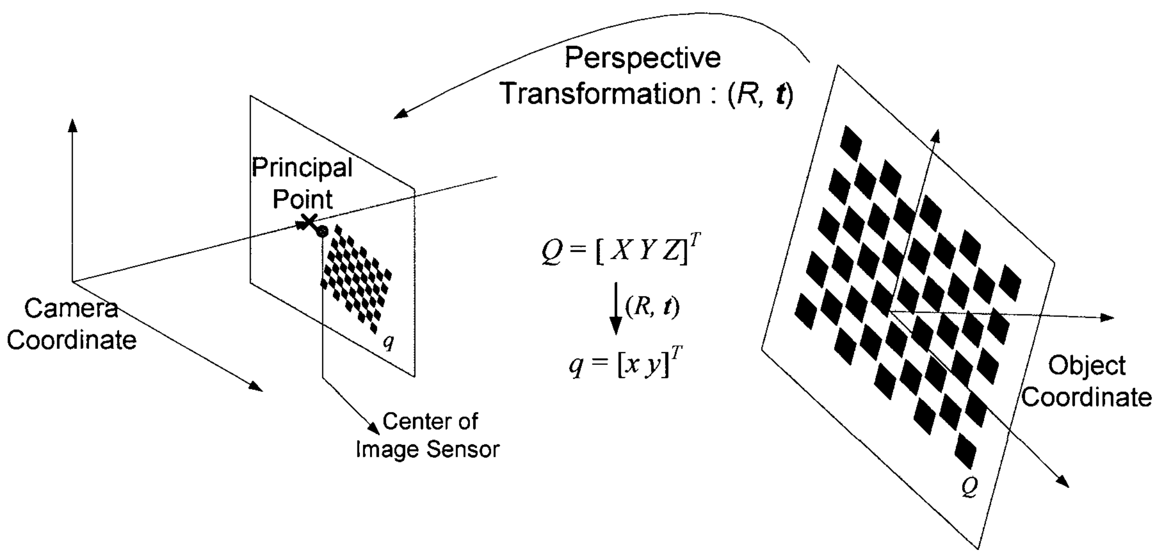


Figure 8. Perspective Transformation of an object in a scene. *The principal point which is the cross point of the image plane and the optical axis does not correspond with the center of the image sensor. The object consists of black squares called 'checkerboard' used for camera calibration.*

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_z(\theta) \times R_y(\varphi) \times R_x(\phi)$$
$$= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{6}$$

and the rotation matrix is a orthogonal matrix, that is, $R^T R = RR^T = I$.

The translation vector denoted the offset between the origin of the object coordinate and the camera coordinate. This can be computed as $\boldsymbol{t} = origin_{object}$ - $origin_{camera}$.

Finally, the pinhole camera perspective transform equations can be derived with above parameters. The relationship between the 3D point Q and its image projection q is given by

$$\tilde{q} = sM\begin{bmatrix} R & \vec{t} \end{bmatrix}\tilde{Q} \equiv sMW\tilde{Q}. \tag{7}$$

where s is an arbitrary scale factor, W is called the extrinsic parameters and the M is called the intrinsic parameters. The 3 by 4 matrix W is also called camera projection matrix, which mixes both rotation and translation parameters.

3.1.2. Camera Calibration

Camera calibration is an essential task in computer vision system in order to extract metric information from the 2D images [43]. The task determines parameters of the transformation between an object in 3D space and the 2D image observed by the camera from visual information (images). The transformation includes extrinsic parameters which are the rotation R (orientation) and the translation $\boldsymbol{t}$ (location) of the camera and intrinsic parameters which are f_x, f_y, c_x, and c_y. The rotation matrix, although consisting of nine elements, has only 3 DOF. The translation vector ***t*** obviously has 3 parameters. Therefore, there are six extrinsic parameters and four intrinsic parameters, totalling 10 parameters.

Lens Distortion Correction

The pinhole camera model presented above is a simple linear model that serves as the foundation of computer vision. However, it has been shown that the pinhole model is insufficient for most computer vision applications [44]. The pinhole model does not account for non-linearity comes from manufacturing tolerances, lens imperfections, assembly errors and intensity/spatial quantization effects.

Usually, the pinhole model is a basis that is extended with some corrections for the systematically distorted image coordinates. The most commonly used correction is radial lens distortion that causes the actual image point to be displaced radially in the image plane [44]. Center of curvature of lens surfaces are not always strictly collinear. This introduces another common distortion type called tangential distortion. The expressions for the both distortions are shown below:

Radial distortion

$$\begin{aligned} x_{corrected} &= x(1+k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1+k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned}, \tag{8}$$

Tangential distortion

$$\begin{aligned} x_{corrected} &= x + [2 p_1 xy + p_2 (r^2 + 2x^2)] \\ y_{corrected} &= y + [p_1 (r^2 + 2y^2) + 2 p_2 xy] \end{aligned}, \tag{9}$$

where $r^2 = x^2 + y^2$.

Homography between the model plane and its image

In computer vision, the 'planar homography' denotes the perspective transformation of the scene from one plane to another plane. The transformation of the points on the 3D planar object to the camera images is an example of planar homography. This transformation can be computed with a set of corresponding points on the 3D planar object and the 2D image when they are expressed in the homogenous coordinates.

$$\tilde{q} = sH\tilde{Q}, \qquad \text{where} \quad \tilde{Q} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad \tilde{q} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \text{and } s \text{ is scale factor.} \tag{10}$$

As stated in eq.(7), the homography transformation H can be divided into two physical matrices, the camera intrinsic matrix M and the camera extrinsic matrix W. Without loss of generality, it can be assumed that the 3D model plane is on $Z = 0$ of the world coordinate system. The rotation matrix is denoted with three column vectors r_1, r_2, and r_2. Then the eq. (10) is restated as

$$\tilde{q} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM\begin{bmatrix} r_1 & r_2 & r_3 & \vec{t} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = sM\begin{bmatrix} r_1 & r_2 & \vec{t} \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = sH\tilde{Q}', \tag{11}$$

where

$$H = sM\begin{bmatrix} r_1 & r_2 & \vec{t} \end{bmatrix} \quad \text{and} \quad \tilde{Q}' = \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{12}$$

By these simplified expressions of the camera extrinsic matrix and homogenous world coordinate points, the planar homography matrix can be estimated with a set of corresponding points on the 3D planar object in the world coordinate system and the 2D points in the image coordinated system. Let's denote it by $H = [h_1 \ \ h_2 \ \ h_3]$. From eq. (12), we have

$$[h_1 \ h_2 \ h_3] = s M \begin{bmatrix} r_1 & r_2 & \vec{t} \end{bmatrix}$$

and

$$h_1 = sMr_1 \qquad \text{or} \qquad r_1 = \lambda M^{-1} h_1$$

$$h2 = sMr_2 \qquad \text{or} \qquad r_2 = \lambda M^{-1} h_2$$

$$h_3 = sMr_3 \qquad \text{or} \qquad r_3 = \lambda M^{-1} h_3$$

where s = $1/\lambda$.

Using the knowledge that r_1 and r_2 are orthonormal which means that the norms of two vectors are equal and the cross product equals to zero, we have

$$h_1^T M^{-T} M^{-1} h_2 = 0 \tag{14}$$

and

$$h_1^T M^{-T} M^{-1} h_1 = h_2^T M^{-T} M^{-1} h_2. \tag{15}$$

These are the two basic constraints on the intrinsic and extrinsic parameters, given on homography transformation. Since a homography transformation matrix has 8 DOF and there are 6 extrinsic parameters, we can only obtain 2 constraints on the intrinsic parameters.

The pinhole camera model and its associated parameters can be computed with Direct Linear Transformation (DLT) [43],[44],[45]. Since no iterations are required, DLT is computationally fast, however, it has at least following two disadvantages. First, lens distortion cannot be considered and secondly, the DLT is more difficult to be fixed. Due to these difficulties the more accurate camera calibration methods are required.

In [46],[47], the author used 'Closed-form solution' and non-linear optimization technique based on the maximum likelihood criterion to solve the camera calibration problem.

If we let

$$B = M^{-T}M^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x^{\,2}} & 0 & \frac{-c_x}{f_x^{\,2}} \\ 0 & \frac{1}{f_y^{\,2}} & \frac{-c_y}{f_y^{\,2}} \\ \frac{-c_x}{f_x^{\,2}} & \frac{-c_y}{f_y^{\,2}} & \frac{c_x}{f_x^{\,2}} + \frac{c_y}{f_y^{\,2}} + 1 \end{bmatrix}. \tag{16}$$

Since the B is symmetric, it can be expressed as 6 dimensional vector

$$\boldsymbol{b} = [B_{11}\ B_{12}\ B_{22}\ B_{13}\ B_{23}\ B_{33}]^T \tag{17}$$

The term, $h_i^T M^{-T} M^{-1} h_j\,(h_i^T B h_j)$, can be re-phrased with this 6D vector $\boldsymbol{b}$,

$$h_i^T B h_j = v_{ij}^{\ T} b = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i3}h_{j3} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}, \tag{18}$$

with $v_{ij} = [h_{i1}h_{j1}\ \ h_{i1}h_{j2} + h_{i2}h_{j1}\ \ h_{i2}h_{j2}\ \ h_{i3}h_{j1} + h_{i1}h_{j3}\ \ h_{i3}h_{j2} + h_{i2}h_{j3}\ \ h_{i3}h_{j3}]$.

Therefore, the two fundamental constraints eq. (14) and eq. (15), from a given homography transformation, can be rewritten as

$$\begin{bmatrix} v_{12}^{\ T} \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0. \tag{19}$$

If n images of the model plane are observed (n homography transformations), by stacking n such equations as eq. (19) we get

$$Vb = 0, \tag{20}$$

where V is a $2n \times 6$ matrix. If the number of images, n, is larger than or equal to 2, the solution of the vector $\boldsymbol{b}$ can be computed.

The solution to eq. (20) is well known as the eigenvector of $V^T V$ associated with the smallest eigenvalue. Once the $\boldsymbol{b}$ is obtained, then the camera intrinsic parameters in matrix M can be solved as below

$$\begin{aligned} f_x &= \sqrt{\lambda / B_{11}} \\ f_y &= \sqrt{\lambda B_{11} / (B_{11} B_{22} - B_{12}{}^2)} \\ c_x &= -B_{13} f_x{}^2 / \lambda \\ c_y &= (B_{12} B_{13} - B_{11} B_{23}) / (B_{11} B_{22} - B_{12}{}^2) \end{aligned} \tag{21}$$

where $$\lambda = B_{33} - (B_{13}{}^2 + c_y (B_{12} B_{13} - B_{11} B_{23})) / B_{11}. \tag{22}$$

The extrinsic parameters can be calculated as

$$\begin{aligned} r_1 &= \lambda M^{-1} h_1 \\ r_2 &= \lambda M^{-1} h_2 \\ r_3 &= r_1 \times r_2 \\ \boldsymbol{t} &= \lambda M^{-1} h_3 \end{aligned} \tag{23}$$

Up to now, the lens distortion of a camera has not been considered and intrinsic and extrinsic parameters are derived with the lens distortion parameters set as 0. Since the lens distortion factor has significant effects on most cameras, the distortion should be corrected to derive the metric information about real objects in real world from an image.

Let $q_p = [x_p \;\; y_p]^T$ be the ideal (non-observable distortion free) image coordinates and $q = [x \;\; y]^T$ the corresponding real observed image coordinates. Then, the ideal point can be derived with the previously estimated intrinsic parameters as follow

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_x X^W / Z^W + c_x \\ f_y Y^W / Z^W + c_y \end{bmatrix} \tag{24}$$

A set of corresponding points, q_p and q, is used for calculating the five lens distortion parameters k_1, k_2, k_3, p_1, and p_2 by stacking n (number of observe points) equations as following equation

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ 2p_2 xy + p_1(r^2 + 2y^2) \end{bmatrix}. \tag{25}$$

After solving all of the intrinsic parameters and lens distortion parameters, then the whole procedure is repeated to refine those parameters. Since the intrinsic parameters are computed with idea pinhole camera model, they do not include the lens distortion effects. Thus by doing the whole procedure recursively until the 9 intrinsic parameters, f_x, f_y, c_x,, c_y, k_1, k_2, k_3, p_1, and p_2 are converged, the camera calibration can be done.

Camera Calibration Procedure

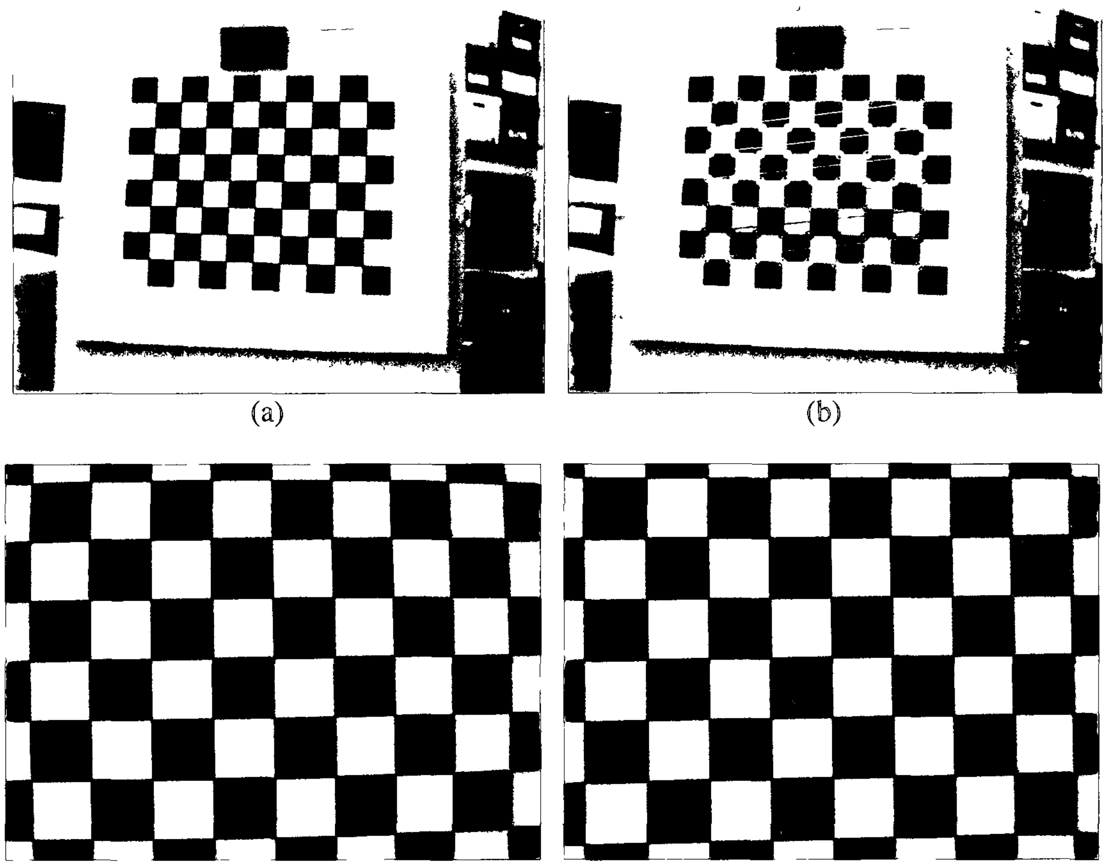
OpenCV is well known library of programming functions developed by Intel for the computer vision system and it provides a set of functions associated with the camera calibration. The procedure for computing camera parameters is based on Zhang's method described in [46],[47] and for the lens distortion model, it is following Brown's work [48].

Many scholars have been proposed and improve the camera calibration methods for more convenient and reliable way and the works in [46],[47],[49] was the first fully exploited camera calibration method with multiple views of planar pattern taken from unknown view points. These works describe how to obtain the 10 parameters of the camera from a minimum of two homographies between the planar pattern and image plane. Mainly due to the convenience of the plane based method and the well documented references of the OpenCV, the calibration function of the OpenCV is widely used. In this thesis, the camera calibration is performed by using OpenCV camera calibration functions.

The following steps are the procedure of the camera calibration.

1. Print a pattern and attach it to a planar surface.
2. Take a few images of the model plane under different orientations by moving either the plane or the camera.
3. Detect the feature points in the images.
4. Estimate the five intrinsic parameters and all extrinsic parameters of the each image by closed form solution.
5. Estimate the coefficients of the two lens distortion factors by the same manner.
6. Repeat the step 4 and 5 until the 9 intrinsic parameters are converged.

Fig. 9(a) presents a checkerboard pattern used for the camera calibration. The checkerboard pattern is s most widely used pattern in camera calibration because of the easy sub-pixel detection algorithm with high precision and the Fig.(b) shows the detected feature points in a sub-pixel accuracy [50]. After computing the 9 camera intrinsic parameters, the original image distorted by the lens and camera structural imperfection can be corrected as shown in Fig. 9(c) and Fig. 9(d).



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

Figure 9. Checkerboard Pattern for Camera Calibration in OpenCV. *(a) Raw image captured by a camera. (b) Feature detected image by corner detection function. (c) Distorted image by lens distortion. (d) Corrected image after camera calibration.*

3.2. Neural Network for Character Recognition

This sub-chapter gives a brief explanation about "Artificial Neural Network (ANN)" and how to solve the pattern recognition problem. Since the proposed fiduciary marker system uses alphabet characters for identifying four sub-markers, a reliable OCR (Optical Character Recognition) algorithm is required, and ANN is selected to satisfy this requirement.

The first two parts of this sub-chapter present basics of ANN and 'Backpropagation Neural Network', and the last part explains the OCR procedure achieved by the neural network pattern recognition scheme

.

3.2.1. Neural Network and Perceptron

ANNs are trainable and dynamic systems that can estimate input-output functions [51],[52],[53]. The ANN is a network structure consisting of a number of nodes connected through directional links. Each node represents a process unit and the links between nodes define the casual relationship between the connected nodes.

The ANNs have received a lot of attentions because it mimics the remarkable information processing characteristics of the biological system such as nonlinearity, high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information, and their capability to generalize [54]. Those characteristics of processing are necessary in real world because

- nonlinearity allows better fit to the data,
- noise-insensitivity provides accurate prediction in the presence of uncertain data and measurement errors,

- high parallelism implies fast processing and hardware failure-tolerance,
- learning and adaptivity allow the system to modify its internal structure in response to changing environment, and
- generalization enables application of the model to unlearned data.

Basic findings from the biological neuron operation enabled to model the operation of simple artificial neurons. An artificial neuron receives inputs as stimuli from the environment, combines them in a special way to form a 'net input' ξ, transmits it through a transfer function $\sigma(\xi)$, and passes the output y forward to another neuron or the final destination. Fig. 10 illustrates n biological neurons with various signals of intensity x and synaptic strength w feeding into neuron with transfer function $\sigma(\xi)$, and the system of

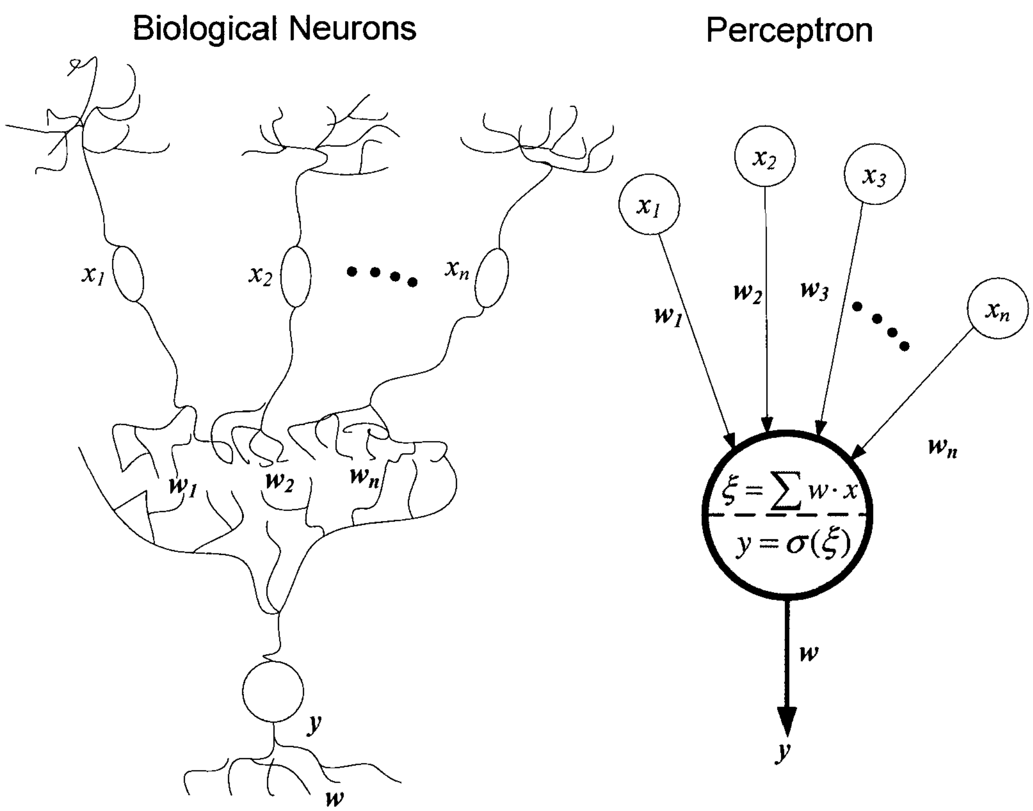


Figure 10. Biological neurons and a perceptron. *The perceptron consists of a set of neurons and the signals from multiple neurons are summed up via the single layer perceptron. Then, the transfer function decides the output of the perceptron.*

equivalent artificial neurons called 'Perceptron'.

The net input is computed as the inner product of the input signals $X = [x_1, x_2, ...x_n]$ and their connection strength(weight) $W = [w_1, w_2, ...w_n]$. For n signals, the perceptron operation is expressed as

$$y = \begin{Bmatrix} 1, & if \ \sum_{1}^{n} w_i \cdot x_i \geq threshold\ value \\ 0, & if \ \sum_{1}^{n} w_i \cdot x_i < threshold\ value \end{Bmatrix} \tag{26}$$

with 1 indicating 'on' and 0 indicating 'off'.

The transfer function $\sigma(\xi)$ is a output conditioning function which activates or deactivates the output of the neuron. In case of the original perceptron model, the unit step function is used for the transfer function but there are other types of transfer function such as piecewise linear, sigmoid, or Gaussian as shown in Fig. 11.

The perceptron can be trained on a set of examples, called 'training set', using a special learning rule. The perceptron weights are changed in proportion to the difference between the desired output and the perceptron output for each training example. The error is a function of all the weights and forms an irregular multi-demensional complex hyperplane with many peaks, saddle points, and minima. Using specialized search technique, the learning process tries to obtain the set of weights that corresponds to the

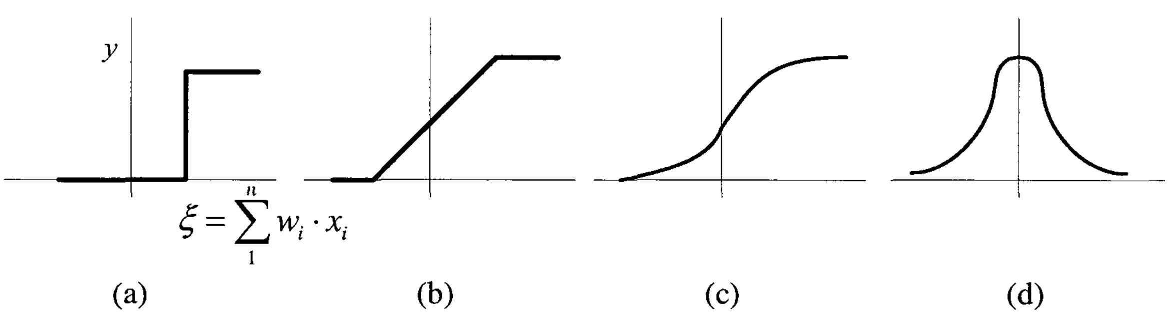


Figure 11. Differenct types of transfer function. *(a) unit step function, (b) piecewise linear function, (c) sigmoid function, (d) Gaussian function.*

global minimum.

In order to deal with nonlinearity separable problems, additional layers of neurons placed between the input layer and the output layer are needed. This ANN architecture is called 'Multi-Layered Perceptron' (MLP). Because these intermediate layers do not interact with the external environment, they are called hidden layers. The addition of hidden layer extends the ability of the ANN to solve the nonlinear classification problems (See Fig. 12) [54],[55].

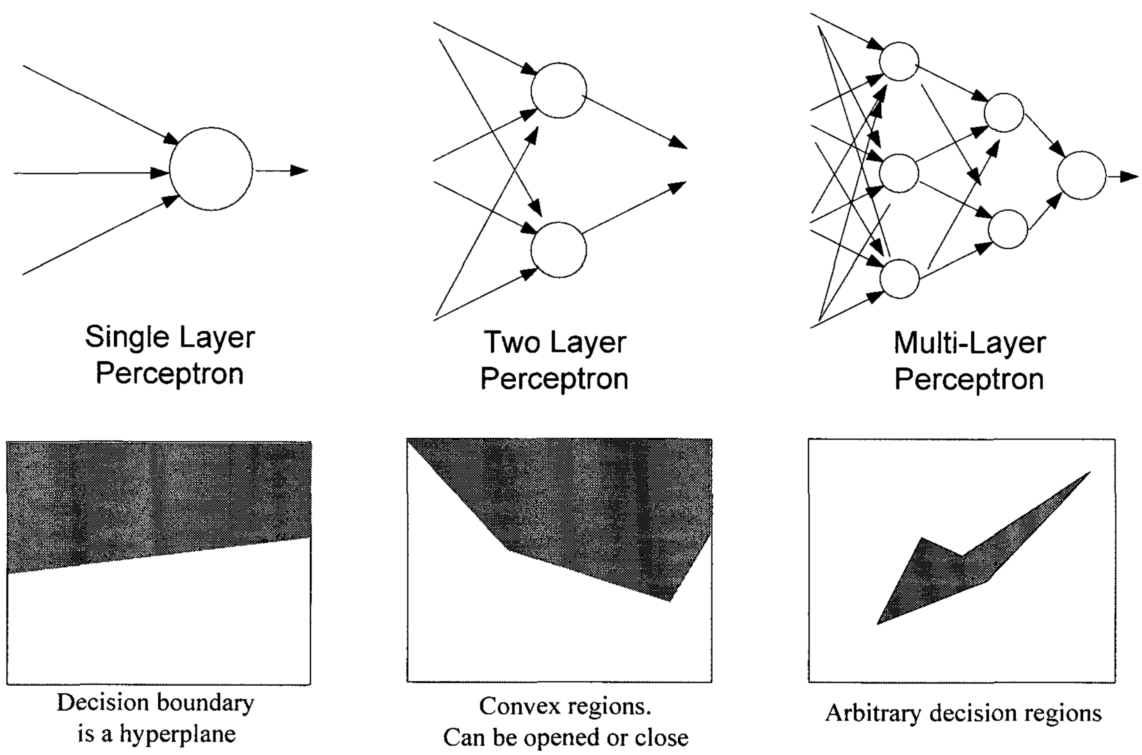


Figure 12. Types of perceptron and the classification ability of each type. *The non-linearity problem can be solved with the MLP.*

3.2.2. Backpropagation Neural Network

The feedforward error-backpropagation learning algorithm is the most famous procedure for training ANNs. Each iteration in backpropagation neural network consists of two phases: feedforward activation to produce a solution, and a backward propagation of the computed error to modify the connection weights. The backpropagation is based on searching an error surface using gradient descent for point with minimum error.

Fig. 13 shows a typical multi-layered perceptron (MLP) and it has three layers. The input layer is connected to the outside world and has no processing power in it. The hidden layer which cannot be seen from the outside of the network is the layer provides flexibility

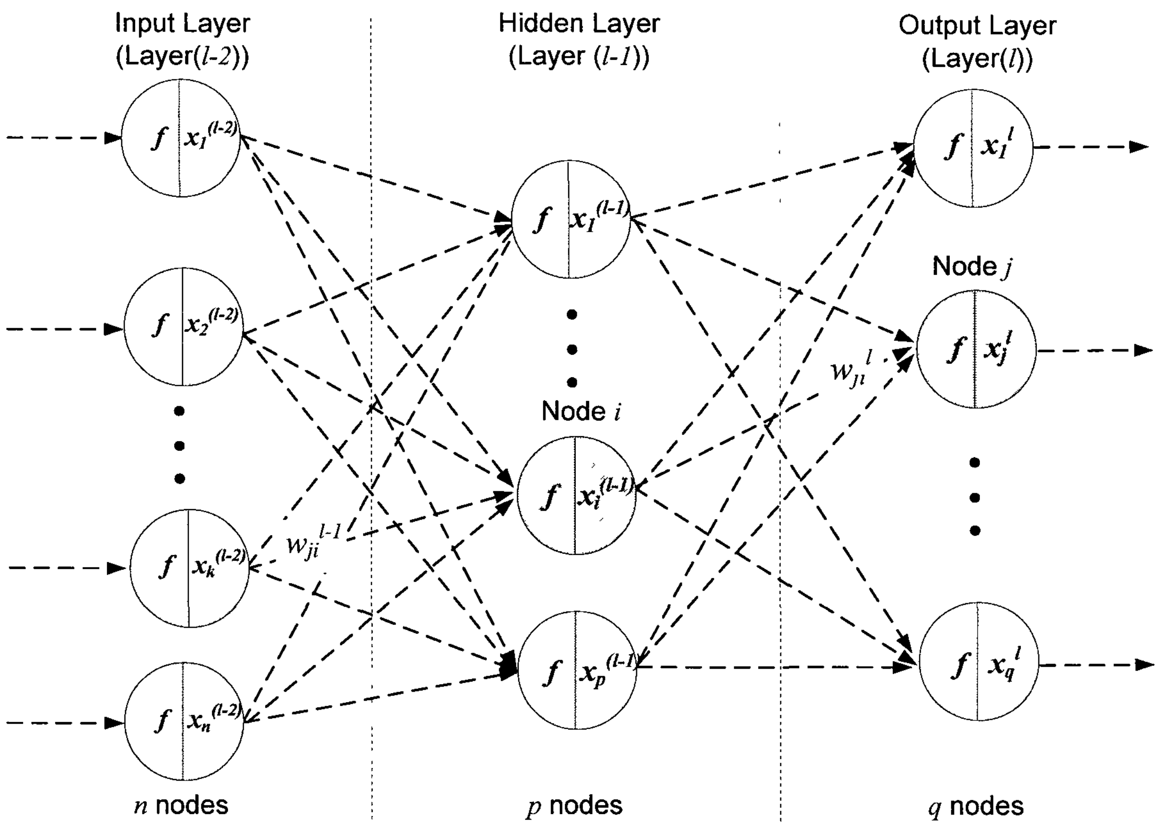


Figure 13. Feedforward Backpropagation Neural Network. *There are n nodes in input layer, p nodes in hidden layer, and q nodes in output layer.*

to the network. The output layer is connected to the outside world again.

Let's assume there are L interlayers in MLP network. For any interlayer l, there are N_l nodes and $N_l \times N_{l-1}$ connection links with weights $W \in R^{N_l \times N_{l-1}}$, where N_l and N_{l-1} are the numbers of nodes in interlayers l and l-1 respectively. A connection weight is denoted by w_{ji}^l if it connects a jth node in interlayer l and an ith node in interlayer l-1. In any interlayer l, a typical neuron j integrates signals from preceding interlayer l-1 conjunction with link weights and produces a net effect ξ and it can be denoted as:

$$\xi_j{}^l = \sum_{i=1}^{N_{l-1}} w_{ji}{}^l \cdot x_i{}^{l-1} . \tag{27}$$

The node ouput x_j^l is determined using a transfer function $\sigma(\ )$, that converts the total signal into a real number from a bounded interval by

$$x_j{}^l = \sigma(\xi_j{}^l) = \sigma\left(\sum_{i=1}^{N_{l-1}} w_{ji}{}^l \cdot x_i{}^{l-1}\right) . \tag{28}$$

In order to train a multi-layered perceptron, a new learning rule is needed. The original perceptron learning rule cannot be extended to the multi-layered perceptron since the output function of the original perceptron, the step function, is not differentiable. Thus a new transfer function is needed that is non-linear, otherwise non-linearly separable functions could not be implemented, but which is differentiable. The one that is most often used in MLP is the sigmoid function, shown in Fig. 11(c). The equation for this function is

$$\sigma(\xi) = \frac{1}{1+e^{-\xi}} . \tag{29}$$

where $-\infty < \xi < \infty$ and $0.0 \le \sigma(\xi) \le 1.0$.

In the backpropagation phase of each training iteration (t), an arbitrary weight w_{ji}^{l} will be updated from its previous state (t-1) value according to

$$w_{ji}^{\,l}(t) = w_{ji}^{\,l}(t-1) + \Delta w_{ji}^{\,l}(t) \tag{30}$$

where $\Delta w_{ji}^{\,l}$ is the incremental change in the weight. The weight change is determined via the modified delta rule which can be written as

$$\Delta w_{ji}^{\,l} = \eta \delta_j^{\,l} x_i^{\,l-1} + \mu \Delta w_{ji}^{\,l}(-) \tag{31}$$

where η is the learning rate controlling the update step size, μ is the momentum coefficient, $w_{ji}^{l}(-)$ is the previous weight, and x_i^{l-1} is the input from the (l-1)th interlayer. The first part of the right hand side of eq. (31) is the original delta rule.

The weight change in eq. (31) can also be expressed with gradient descent in generalized form for an interlayer l:

$$\Delta w_{ji}^{\,l} = -k\left(\frac{\partial e^{l}}{\partial w_{ji}^{\,l}}\right) \tag{32}$$

and therefore, in order to determine the incremental changes for the l th interlayer, the main task is to quantify the error gradient.

Importantly, in the required weight change, the definition of the $\delta_i^{\,l}$ changes depending on whether the considered neuron is in the output layer or in a hidden layer.

In the case of output layer, $\delta_i^{\,l}$ can be calculated by

$$\delta_j^{\,L} = (x_j^{\,L} - y_j)\,x_j^{\,L}(1 - x_j^{\,L}) \tag{33}$$

where y_j is the desired output.

If the neuron is in a hidden layer, it is computed as below

$$\delta_j^{\,l} = x_j^{\,L}(1 - x_j^{\,L})\left(\sum_{k=1}^{r} \delta_k^{\,l+1} w_{kj}^{\,l+1}\right), \tag{34}$$

where $\delta_k^{\,l+1}$ is calculated for a given non-output layer from the one level higher layer. These equations provide all necessary equations that are needed to apply backpropagation to a multi-layered perceptron.

3.2.3. Character Recognition

Optical Character Recognition (OCR) is a well known and profoundly researched field of automatic classification methods. In addition to its practical interest, it shows all the typical problems encountered with classification such as choice of data features, choice of classifier, and choice of learning rule. There are three different approaches to improve the performance of the OCR system [56]. First, development of the better classifier can improve the performance. The second approach is to combine the outputs of multiple classifiers. Lastly, exploring new features which possesses high discriminative power can enhance the performance of the OCR system.

In fact, the performance of an OCR system heavily depends on the features used in it [56],[57]. Many scholars have proposed various features of many kinds of a character system. The various features of character systems can be categorized into structural and numerical feature. A structural feature is derived from the geometrical shape of the character images and has intuitive semantic contents whereas a numerical feature based on the transformed image converted from the original image into some other forms. The numerical feature might not be understandable.

The structural features of character images are more suitable for machine printed or hand-printed characters since the approach using structural features normally has a limitation on discriminating the structural variations of handwritten characters [56]. Since in the proposed fiduciary marker, printed alphabet characters are used for the marker identification, the approach using structural features would provide satisfying recognition result.

In [68], the authors investigated the ability of several variations of Holland-Style adaptive classifier systems using the 16 structural features of alphabet character system and achieved a recognition performance rate of 82.7%. The authors of [69] recognize the English letters using modular adaptive RBF type neural networks with 91.85% accuracy using the 16 alphabet features proposed in [68] and they also achieved a recognition rate of 94.13% using combinative neural networks based classifiers in [70].

For the proposed marker system, the 12 primitive structural features among 16 features described in [68]–[70] are adopted to recognize the 26 alphabet characters. When the marker is shown in the captured image, the marker detection algorithm extracts the character image for the marker identification. The extracted character image is, then, fed into the feature extraction algorithm to form a structural feature vector of the character.

Following descriptions are the the 12 primitive structural features used in the proposed marker system among the 16 features proposed in [68]

1. The total number of "On" pixels in the alphabet image.
2. The mean horizontal position of all "On" pixels relative to the center of the box. This feature has a negative value if the image is "left heavy" as the letter L.
3. The mean vertical position of all "On" pixels relative to the center of the box.
4. The mean squared value of the horizontal pixel distances as measured in 6 above. This attribute will have a higher value for images whose pixels are more widely

separated in the horizontal direction as the letters W or M.

5. The mean squared value of the vertical pixel distances as measured in 7 above.
6. The mean product of the horizontal and vertical distances for each "On" pixel as measured in 6 and 7 above.
7. The mean value of the squared horizontal distance times the vertical distance for each "On" pixel. This measures the correlation of the horizontal variance with the vertical position.
8. The mean value of the squared vertical distance times the horizontal distance for each "On" pixel. This measures the correlation of the vertical variance with the horizontal position.
9. The mean number of edges encountered when making systematic scans from left to right at all vertical positions within the box. This measure distinguishes between letters like "W" or "M" and letters like 'T' or "L."
10. The sum of the vertical positions of edges encountered as measured in 13 above. This feature will give a higher value if there are more edges at the top of the box, as in the letter "Y."
11. The mean number of edges encountered when making systematic scans of the image from bottom to top over all horizontal positions within the box.
12. The sum of horizontal positions of edges encountered as measured in 15 above

By this definition of the feature space, each alphabet character image can have 12 numerical attributes such as the position and dimensions of bounding box and number of black pixels. Each character image is then scanned, pixel by pixel, to extract 12 numerical attributes. These attributes represent primitive statistical features of the pixel distribution and fully describe the geometrical structure of the each character.

After these 12 structural features of the training character set are computed by the feature extraction algorithm, the feature vector is fed into the MLP backpropagation neural network meaning that there are 12 input nodes in input layer. Then, all the connection weights are modified by the backpropagation learning rule described in the previous section. During training the error is measured between the desired output and the actual output, and the aim is to reduce this error by adjusting the weights.

Surprisingly, the network never performs well on the test set as it did on the training set. This phenomenon is known as 'over-training' and the network is said to over fit the training data [71]. To avoid this phenomenon, we need to split the whole data set into three sets rather than two sets. These sets are usually called the training set, test set, and validation set. During training, the testing set is also tested with the modified weights by the training set and shows the test error. If the test error is increasing, it means that the network is over trained and is needed to stop the training. This strategy makes it possible to avoid over fitting problem.

3.3. Reed Solomon Codes for Error Correction

Reed-Solomon error correcting codes (RS codes) are widely used in communication systems and data storages to recover data from possible errors that occur during transmission or reading failures [66]. This property of the RS codes allows the fiduciary marker to be robust against occlusions, data bit loss or data bit reading failures when the marker identification algorithm reads the marker data.

This sub-chapter summarizes principles of the Reed-Solomon codes and how to apply the RS codes to the proposed fiduciary marker system.

3.3.1. Classification of Reed Solomon codes

A Reed-Solomon code is a block code, meaning that the message to be transmitted is divided up to separate blocks of data. Each block has parity protection information added to it to form a self contained codeword. It is also systematic, meaning that the

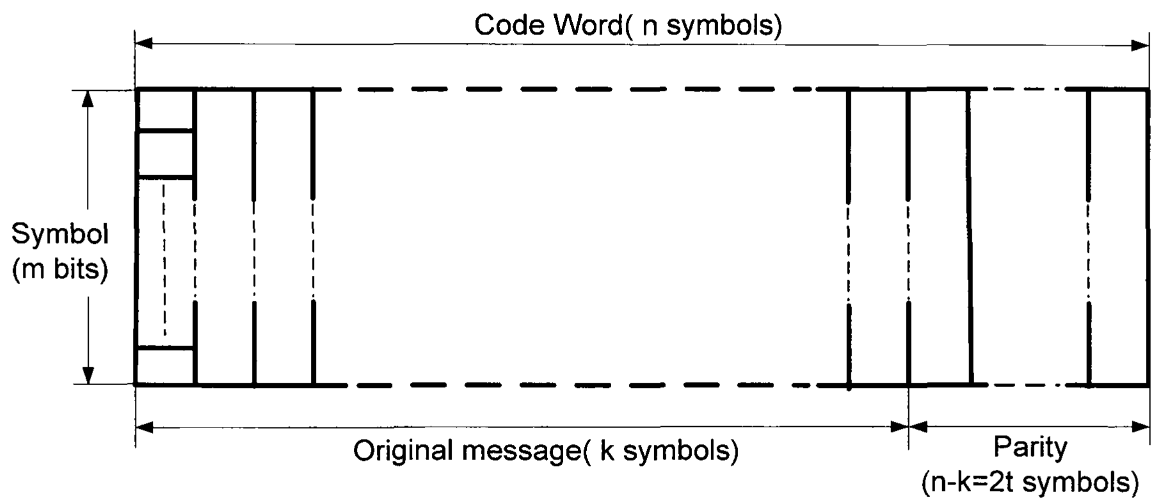


Figure 14. Reed-Solomon code structural definition. *The codeword consists of k data symbols and 2t parity symbols, totaling n of m-bits symbol.*

encoding process does not change the message symbols but the protection symbols are added as a separate part of the block. The structure of the RS code is shown in Fig. 14.

Choosing different parameters for a RS code provides different levels of protection and affects the complexity of implementation. Thus a RS code can be described as an *(n, k)* code, where n is the codeword length in symbols and k is the number of information symbols in the message. The constraint for the code word length is

$$n \leq 2^m - 1 \tag{35}$$

where m is the number of bits in a symbol. When eq. (35) is not an equality, this is referred to as a shortened form of the RS code. There are n-k paraty symbols and t symbol error can be corrected in a codeword, where

$$t = (n-k)\,/\,2\text{, for } n\text{-}k \text{ even} \qquad \text{or} \qquad t = (n-k-1)/2\text{, for } n\text{-}k \text{ odd.} \tag{36}$$

3.3.2. Galois Finite Field

A Galois field consists of a set of elements and the elements are based on a primitive element, usually denoted α, and take the values $0, \alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{N-1}$ to form a set of 2^m elements, where $N = 2^m - 1$. The field is then known as $GF(2^m)$.

The value of α is usually chosen to be 2, although other values can be used. Having chosen α, higher powers can then be obtained by multiplying by α at each step. However, it should be noted that the rules of multiplication in a Galois filed are not those that we might normally expect.

In addition to the powers of α form, each field element can also be represented by a polynomial expression of the form shown below

$$a_{m-1}x^{m-1} + a_{m-2}x^{m-2} \ldots + a_1 x + a_0 \tag{37}$$

where the coefficients a_{m-1} to a_0 take the value 0 or 1. Thus we can describe a field element using the binary number $a_{m-1}, a_{m-2},....,a_1,a_0$ and the 2^m field elements correspond to the 2^m combinations of the m-bit number.

For example, in the Galois field with 16 elements (known as $GF(2^4)$ with m=4), the polynomial representation of any symbol is

$$a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 \tag{38}$$

with $a_3a_2a_1a_0$ corresponding to the binary numbers 0000 to 1111. Alteratively we can refer to the field elements by the decimal equivalents 0 to 15 as a short-hand version of the binary numbers.

The arithmetic combination of Galois field elements always produces another field element. When we add two field elements, we add the two polynomials

$$(a_{m-1}x^{m-1} + ... + a_1x + a_0) + (b_{m-1}x^{m-1} + ... + b_1x + b_0) = c_{m-1}x^{m-1} + ... + c_1x + c_0 \tag{39}$$

where

$c_i = 0,$ for $a_i = b_i$ and $c_i = 1,$ for $a \neq b_i$

Thus two Galois field elements are added by modulo-2 addition of the coefficients, or in binary form, producing the bit-by-bit exclusive-OR function of the two binary numbers. For example, in $GF(2^4)$ we can add field elements $x^3 + x$ and $x^3 + x^2 + 1$ to produce $x^2 + x + 1$. In case of the binary numbers, this can be restated as 1010 + 1101 = 0111 ($c_3 = 0, c_2 = 1, c_1 = 1, c_0 = 1$) or as decimals 10 + 13 = 7.

Because of the exclusive-OR function, the addition of any element to itself produces zero. So the addition in Galois field such as 2 + 2 produces zero. Furthermore, the subtraction of two Galois field elements turns out to be exactly same as the addition since the subtraction and the addition defined in the XOR function produce the same results. It is useful to realise that a field element can be added or subtracted with exactly same effect, so minus sign can be replaced by plus sign in field element arithmetic.

"Primitive polynomial $p(x)$' is an important part of the defining the RS code. The primitive polynomial is a polynomial can be defined up to degree of m (m is from the GF(2^m)) which is irreducible. It forms part of the process of multiplying two field elements together. For a Galois field of a particular size, there is a choice of suitable polynomials. Using a different primitive polynomial with the $p(x)$ of specific RS code will produce incorrect results. Some examples of the primitive polynomials are shown below

$$m = 4: \qquad p(x) = x^4 + x + 1 \tag{40}$$

$$m = 8: \qquad p(x) = x^8 + x^4 + x^3 + x^2 + 1 \tag{41}$$

$$m = 16: \qquad p(x) = x^{16} + x^{12} + x^3 + x + 1 \tag{42}$$

$$m = 32: \qquad p(x) = x^{32} + x^{22} + x^2 + x + 1 \tag{43}$$

$$m = 64: \qquad p(x) = x^{64} + x^4 + x^3 + x + 1 \tag{44}$$

All the non-zero elements of the Galois field can be constructed by using the fact that the primitive element α is a root of the primitive polynomial, so that $p(\alpha) = 0$. Thus the primitive polynomial for $GF(2^4)$, that is, $p(x) = x^4 + x + 1$, can be written as

$$p(\alpha) = \alpha^4 + \alpha + 1 = 0 \qquad \text{or} \qquad \alpha^4 = \alpha + 1 \tag{45}$$

This primitive polynomial defines the Galois field elements of the $GF(2^4)$, and those are presented in Table 1. If the process shown in Table 1 is continued over α^{14}, it is found that $\alpha^{15} = \alpha^0$, $\alpha^{16} = \alpha^1$, ...so that the sequence repeats with all the values remaining valid field elements. The values of the message and parity symbols of a Reed-Solomon code are elements of a Galois field. Thus for a code based on m-bit symbols, the Galois filed has 2^m elements.

TABLE I
THE FILED ELEMENTS FOR $GF(16)$ WITH
$p(x) = x^4 + x + 1$

Field elements			
Index form	Polynomial Form	Binary form	Decimal form
0	0	0000	0
α^0	1	0001	1
α^1	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

An (n, k) RS code is constructed by forming the 'code generator polynomial $g(x)$', consisting of $n\text{-}k = 2t$ factors, the roots of which are consecutive elements of the Galois field. Choosing consecutive elements ensures that the distance properties of the code are maximised. Thus the code generator polynomial takes the form of

$$g(x) = (x + \alpha^b)(x + \alpha^{b+1})\ldots.(x+\alpha^{b+2t-1}). \tag{46}$$

In the proposed fiduciary marker system, the (15, 5) type RS code is used for the data encoding/decoding. For a (15,5) code, the block length is 15 symbols, 5 of which are information symbols and the remaining 10 symbols are parity words. Since $t = 5$, the code can correct errors up to 5 symbols in a block. Substituting 15 for n in eq. (35) gives the value of m as 4, so each symbol consists of a 4-bit word can the code is based on the $GF(2^4)$ with 16 elements.

The code generator polynomial for correcting up to 5 error words requires 10 consecutive elements of the field as roots, so the code generator polynomial can be written as

$$g(x)=\prod_{i=b}^{b+9}(x+a^{i}). \tag{47}$$

3.3.3. Reed Solomon Encoding / Decoding,

The k information symbols that form the message to be encoded as one block can be represented by a polynomial $m(x)$ of order $k-1$ as below

$$M(x) = M_{k-1}x^{k-1} + M_{k-2}x^{k-2} + \dots + M_1x^1 + M_0 \tag{48}$$

where each of the coefficients $M_{k-1}, M_{k-2}, \dots, M_1, M_0$ is an m-bit message symbols such as $M_{k-1} = s_{m-1}x^{m-1}+s_{m-2}x^{m-2}\dots+s_1x+s_0$, that is, an element of $GF(x^m)$.

The data block $M(x)$ (message) is systematically encoded and defined by the eq. (48) with the generator polynomial $g(x)$ and the number of parity symbols $2t$, resulting the encoded data block $C(x)$ (RS code).

$$C(x) = M(x)x^{2t} + M(x) \bmod g(x) \tag{49}$$

where $g(x)$ is the generator polynomial of degree $2t$ and given by the eq. (46).

After going through a noisy transmission channel, the encoded data can be represented as

$$R(x) = C(x) + E(x) \tag{50}$$

where $E(x)$ represents the error polynomial with the same degree as $C(x)$ and $R(x)$.

Once the decoder evaluates $E(x)$, the transmitted message, $C(x)$, is then recovered by adding the received message, $R(x)$, to the error polynomial, $E(x)$, as shown below

$$
\begin{aligned}
C(x) &= R(x) + E(x) \\
&= C(x) + E(x) + E(x) = C(x) \qquad (51)
\end{aligned}
$$

Note that $E(x) + E(x) = 0$ because addition in Galois field is equivalent to an exclusive-OR and $E(x)$ *XOR* $E(x)$ equals 0.

The encoded codeword is always divisible by the generator polynomial without remainder and this property extends to the individual factors of the generator polynomial. Therefore the first step in the decoding process is to divide the received polynomial by each of the factors $(x + \alpha^i)$ of the generator polynomial shown in eq. (46). This produces a quotient and a remainder as shown below

$$\frac{R(x)}{x+a^i} = Q_i(x) + \frac{S_i}{x+a^i} \qquad \text{for} \qquad b \le i \le b+2t-1. \qquad (52)$$

The remainders S_i resulting from these divisions are known as the 'Syndromes' and these can be written as $S_0, S_1, \ldots S_{2t-1}$ for $b = 0$.

Rearranging eq. (52) produces $S_i = Q_i(x) \times (x + \alpha^i) + R(x)$ so that when $x = \alpha^i$ this reduces to

$$
\begin{aligned}
S_i &= R(\alpha^i) \\
&= R_{n-1}(\alpha^i)^{n-1} + R_{n-2}(\alpha^i)^{n-2} + \ldots. + R_1\alpha^i + R_0 \qquad (53)
\end{aligned}
$$

where the coefficients $R_{n-1} \ldots. R_0$ are the symbols of the received code word.

The syndrome polynomial $S(x)$ can be written as eq. (54) and is used to detect the errors in the transmitted code word by the Key Equation functional block described later.

$$S(x) = \sum_{i=b}^{b+2t-1} S_i . \qquad (54)$$

When $S(x) = 0$, the received code word is error free otherwise, the Key Equation Solver will use $S(x)$ to generate the error locator polynomial $\Lambda(x)$ and the error evaluator polynomial $\Omega(x)$.

The Key Equation is an polynomial function that describes the relationship between the syndrome $S(x)$, error locator $\Lambda(x)$, error evaluator $\Omega(x)$ by the equation shown below

$$\Lambda(x)\, S(x) = \Omega(x) \bmod x^{2t} \tag{55}$$

where

$$\begin{aligned}\Lambda(x) &= \prod_{j=1}^{v}(1 - X_j x)\\ &= 1 + \lambda_1 x + \lambda_2 x^2 + \ldots + \lambda_v x^v\end{aligned} \tag{56}$$

$$\begin{aligned}\Omega(x) &= \sum_{i=1}^{v} Y_i X_i \prod_{j=1, j\neq i}^{v}(1 - X_j x)\\ &= w_0 + w_1 x + w_2 x^2 + \ldots + w_{v-1} x^{v-1}\end{aligned} \tag{57}$$

The error locator polynomial $\Lambda(x)$ has a degree of $v \leq t$ and the error evaluator polynomial $\Omega(x)$ has degree of at most v-1 to determine the magnitude of v number of errors. There are different algorithms that have been used to solve the key equation and two common ones are the Euclidean algorithm and the Berlekamp-Massey (BM) algorithm. The Euclidean algorithm has simpler structure than the BM does. However, it needs a significant amount of logic elements to implement the polynomial division function.

The locations of the errors are determined on the error locator polynomial $\Lambda(x)$. Each α^j for $m_0 \leq j \leq m_0 + 2t - 1$ is plugged into eq. (56). If $\Lambda(x) = 0$, the location of an error is c, when $\alpha^{-j} = \alpha^c$ in the Galois Field. This process is known as the Chien search algorithm. The magnitude of the errors are determined on the error evaluator polynomial $\Omega(x)$. For the ith symbol in the code word, the error magnitude can be computed by

$$E(x)_i = \alpha^i \frac{\Omega(\alpha^{-i})}{\Lambda'(\alpha^{-i})}. \tag{58}$$

The Reed-Solomon error correction codes are widely used in communication systems and data storages to recover data from possible errors that occur during transmission and from disc error respectively [66]. For example, RS codes are used in the digital video broadcasting (DVB) standard DVB-S and in the optical network G. 709, which has a fast transmission rate of 40 Gbps.

This RS code also can be used in the computer vision system for correcting false data reading or data loss. The QR code shown in Fig. 1(b) adopts the RS code and provides four different levels of error correction (7%, 15%, 25%, and 30% of the symbol area) [67]. The RS codes are arranged in the QR code data area so that the data can be read correctly even when they are occluded or damaged. The proposed fiduciary marker systems with RS codes are inspired by the QR code. The proposed fiduciary marker provides 33.3% data restoration (error correction) performance meaning that among 15 symbols (each symbol consists of 4bits) up to 5 symbols can be restored from data loss or false data reading.

3.4. Pseudo Random Multi-Valued Sequences and Fiduciary Marker

This sub-chapter describes the alternative way to assign new features on the proposed marker system by using Pseudo-Random patterns.

Most of the currently available fiduciary marker systems stick to use four corners of the black quadrilateral border as the feature points which are essential for tracking and identification procedure. This sub-chapter discusses the drawback of the quadrilateral border based approach and suggests new features of the proposed marker system.

3.4.1. PRMVS for New Feature Points of Fiduciary Marker

As described in the section 2.1, most of the currently available fiduciary marker systems consist of a quadrilateral black border and an inner data region. The black border is an essential element of the marker system since it provides at least four feature points, which are the four edges, to solve the correspondence problem [24]-[26]. This means that with the four image coordinates and corresponding marker, one can compute the homography matrix and correct the perspective distortion to extract the marker information in the inner data region. This approach, based on quadrilateral corner points, can locate the four corner points to sub-pixel accuracy, however, it shows poor marker detection performance under occlusions.

For many years, various marker systems have been introduced and improved the data capacity, robustness on uncontrolled lighting condition, and data error correction ability. After the advent of the ARToolkit, most of the newly developed marker systems have kept the structure consists of a quadrilateral black border and inner data region. Unfortunately, only few efforts have been invested to introduce new features being used to track the marker.

In order to track the marker and to compute the homography matrix, at least four corresponding feature points should be provided consistently to the marker detection system. The more points the better accuracy of the detection system. Therefore, if a fiduciary marker has more feature points, the system could be more reliable and accurate.

For more reliable fiduciary features, geometric projective invariants are explored in early time of AR [58],[59]. 'Nestor' is a recently developed planar shapes recognition and a 3D pose tracking system using geometric invariants features of natural planar shapes [57]. The system uses contour concavities of a detected shape to generate projective invariant signatures, which allow shape recognition across different viewpoints. According to the inventor of Nestor, the shapes allow identification and pose estimation in cases of partial occlusion and moderate projective distortion.

However, in spite of the robustness of the geometric invariants of shape, the natural feature tracking system is not eligible for the indoor localization system since in order to cover large area of the indoor environment, the number of registered natural shapes in the library is increased resulting increasing computation cost for matching and identification procedures. Thus, in order to keep the computational cost as low as possible, it would be better idea to use digital coding algorithm for pattern matching and identification as ARTag [24],[25] or Studierstube Tracker [42] does. In addition, new feature space still needs to be defined.

'Pseudo-Random Sequences (PRSs)' are certain binary sequences where each binary element has unique information together with predefined number of neighbor elements and have been widely used in AR or VR field. Due to its window property [60], each element in the sequence can be uniquely distinguished, and this property allows PRSs to be used in position recovery applications such as 2D absolute position recovery for Automated Guided Vehicle (AGV) [14]-[17],[61],[62] or 3D object model recovery with PRS encoded structured light [18],[19][63]-[65]. Due to this unique geometric information property of the PR pattern, PRSs can be used for new features of the fiduciary marker.

In [61],[62], the authors presented the 3D position recovery method with 2D array of PR patterns called Pseudo-Random Binary Array (PRBA). In case of [61] the PR pattern is engraved (fixed) onto objects so it is relatively easy to implement whereas in [62] the PR pattern is projected onto the environment by a LCD or slides projector. The later approach is more complex than former one since it is required to calibrate the relationship between the camera and the projector to recover 3D position of the PR pattern.

Structured Light (SL) is well known efficient technique that allows recover 3D object models from a single monocular image by projecting light patterns with known a priori spatial distribution onto the object's ROI. In [19],[64], different types of the pattern codification strategies are explained. The three main categories of the SL pattern codification are time-multiplexing, spatial neighboring, and direct coding.

Pseudo-Random arrays fall into the spatial neighboring codification category which encodes positions as spatially distributed codes. Since the PR pattern offers simplified way to encode spatially distributed maps that are uniquely defined [19], a variety of such patterns have been proposed. As stated above, PRBA is the one of the proposed patterns and one of the most interesting pattern codification techniques is Morano's Pseudo-Random Multi-Valued Sequence (PRMVS), referred to Perfect Sub-Maps (PSMs). [65].

The second type of proposed fiduciary marker design is inspired by the structured light pseudo-random codification. The PRMVS pattern is adopted for a fiduciary marker and engraved onto the marker instead of projecting the pattern so that enough feature points can be assigned to it and the system can achieve robustness against occlusions.

3.4.2. PRMVS Pattern Codification

A pseudo-random multi-valued sequence is a 2D matrix of size $k \times l$ in which many of the possible sub matrices of size $v \times w$ appear exactly once following the window property of the pseudo-random sequence [19]. The absolute position information of the each element of the $k \times l$ matrix can be recovered by sliding the $v \times w$ window on the given pattern.

The PSM pattern developed by Morano [65] is square matrix M of size $L \times L$ ($k = l = L$) and each element m_{ij} within M is assigned one value from a palette of A possible values. The elements of the matrix M is defined off-line following a pseudo-random iteration process. A position of the every matrix element m_{ij} is indexed by a nine-element vector $V_{(ij)}$ where,

$$
\begin{aligned}
V_{(ij)} &= \{m_{i-1,j-1}, m_{i-1,j}, m_{i-1,j+1}, m_{i,j-1}, m_{i,j}, m_{i,j+1}, m_{i+1,j-1}, m_{i+1,j}, m_{i+1,j+1}\} \\
&= \{V_{(ij)1}, V_{(ij)2}, V_{(ij)3}, V_{(ij)4}, V_{(ij)5}, V_{(ij)6}, V_{(ij)7}, V_{(ij)8}, V_{(ij)9}\} \qquad (59)
\end{aligned}
$$

Every $V_{(ij)}$ should equal or exceed a defined minimum Hamming distance, h, where the Hamming distance between $V_{(ij)}$ and $V_{(i'j')}$ is defined as

$$H(ij,i'j') = \sum_{r=1}^{w \times v} (\delta_r) \qquad \text{where} \qquad \delta_r = \begin{cases} 0 & \text{if } V_{(ij)_r} = V_{(i'j')_r} \\ 1 & \text{otherwise} \end{cases} \tag{60}$$

The algorithm used to generate an pseudo-random multi-valued sequence with the above properties is based on a brute-force [64] meaning that the codes are generated by iterating between adding one new codeword and checking in against all preceding ones. For example, for constructing an $L \times L$ matrix M based on three colors (values) with window size of 3 × 3 the following steps are taken: first, a sub-matrix of 3 × 3 is chosen randomly and is places in the north-west vertex of the M. Then consecutive random columns of 1 × 3 are added to the right of this initial sub-matrix, maintaining the integrity

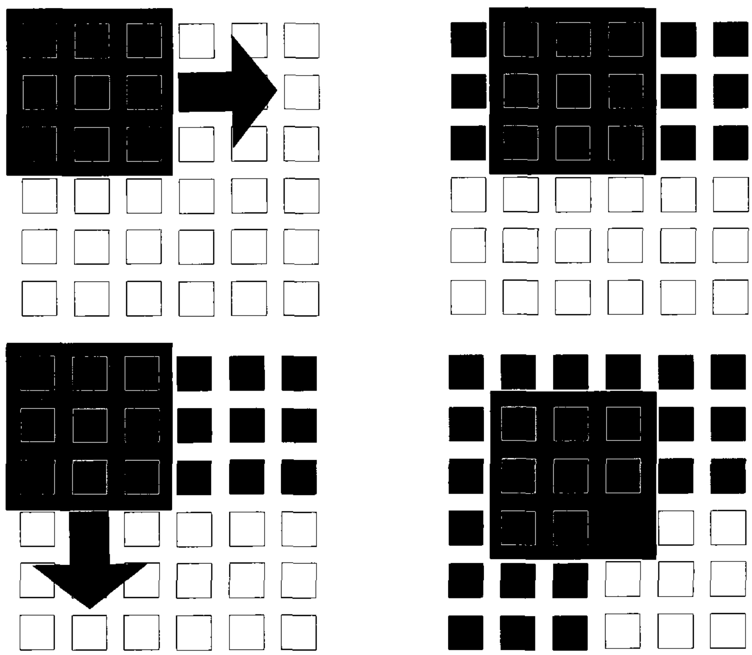


Figure 15. PRMVS pattern generation sequence. *1) Top left : Seeding the first 3×3 pattern. 2) Top right : Consecutive 3×1 columns are added to the right of initial pattern according to the generation rules. 3) Bottom left : Rows of 3 × 1 are added beneath of the initial sub-matrix. 4) Bottom right : Both horizontal and vertical processes are repeated by incrementing the starting coordinates by one*

of the window property of the array and the Hamming distance criteria. Afterwards, rows of 3 × 1 are added beneath of the initial sub-matrix in a similar way. Then, both horizontal and vertical processes are repeated by incrementing the starting coordinates by one, until the whole matrix is filled. The code generating procedure is briefly shown in Fig. 15 and the generated PRMVS example is shown in Fig. 16

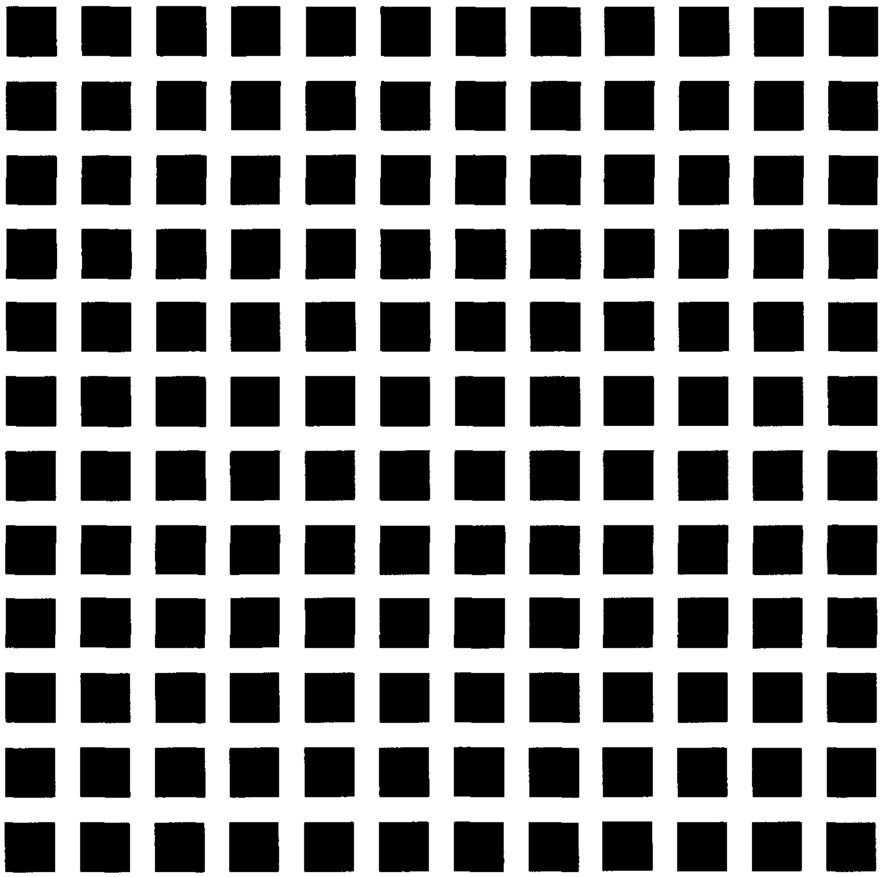


Figure 16. An example of the generated 12 × 12 PRMVS pattern. *A part of this pattern (9 x 9) could be used for a proposed marker system for assigning enough feature points.*

4. Designing Robust fiduciary Marker

This chapter presents how the two proposed fiduciary marker systems are designed and operated while achieving relatively reliable characteristics for the indoor localization comparing to the currently available fiduciary markers.

Through this thesis, two types of fiduciary marker system, named MTag and CPRTag, are proposed for indoor localization purpose.

The MTag fiduciary marker system locates four sub-markers to four corners of MTag marker which has marker ID information encoded with digital techniques. A sub-marker consists of a quadrilateral black border and an alphabet character. By locating four sub-markers on a marker, the marker detection algorithm could be more robust. The 60 bits data, which is marker ID information, is encoded between the four sub-markers while considering effective geometrical distribution to achieve robustness against the partial occlusion of the marker.

The other fiduciary marker system, CPRTag, utilizes the PRMVS pattern which provides abundant unique feature points that hold absolute position information. By engraving the PRVMS pattern on CPRTag instead of the quadrilateral border, the marker still provides enough feature points even when most of the marker area is hidden therefore it can be located precisely. The PRMVS pattern uses multi-values instead of a binary value. Three colors (Red, Green, and Blue) symbolize three values of PRMVS of the CPRTag.

The two fiduciary marker systems, MTag and CPRTag, adopt Reed-Solomon (RS) and Cyclic Redundancy Check (CRC) encoding / decoding techniques for the marker information data generation and transmission. By using RS / CRC coding techniques for marker ID information, the corrupted data bits can be restored up to 33.3% of the original pattern, which is pseudo-random multi-valued sequence (PRMVS), and inner data region located on the spare space among color dots.

The MTag marker system stores geometrical information of each sub-marker with respect to marker coordinate system and uses it to estimate its position of four vertices of the MTag with the detected sub-marker vertices information.

As presented in Section 3.1.2, a transformation from plane (or image) to another plane (or image) can be described by the homography matrix. The homography matrix can be computed if there are more than three corresponding points on two planes (or images). It provides important metric information which could be translated to geometrical information such as rotation and translation of an object.

MTag has 16 feature points (correspondences) from four vertices of each sub-marker. Thus, even under the situation where three sub-markers are hidden or occluded,

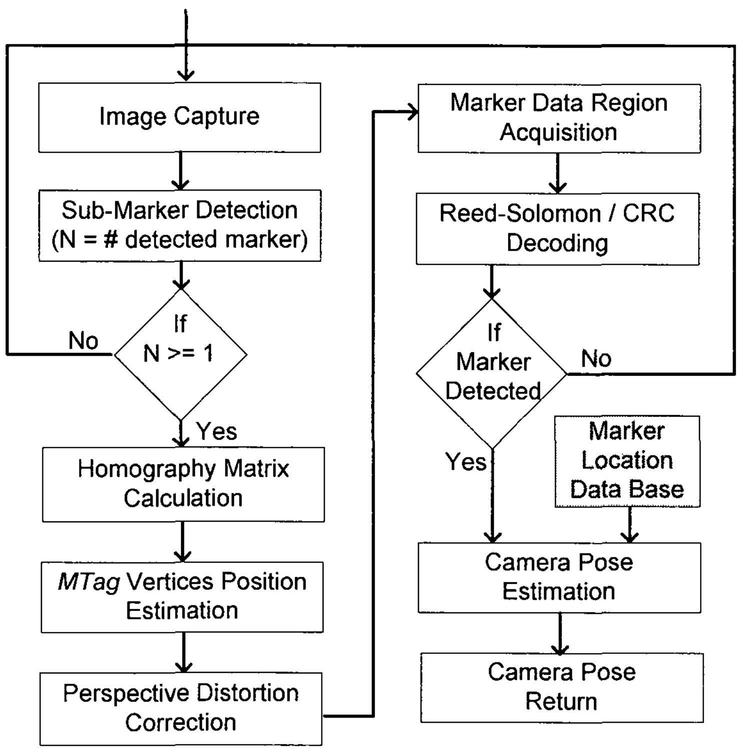
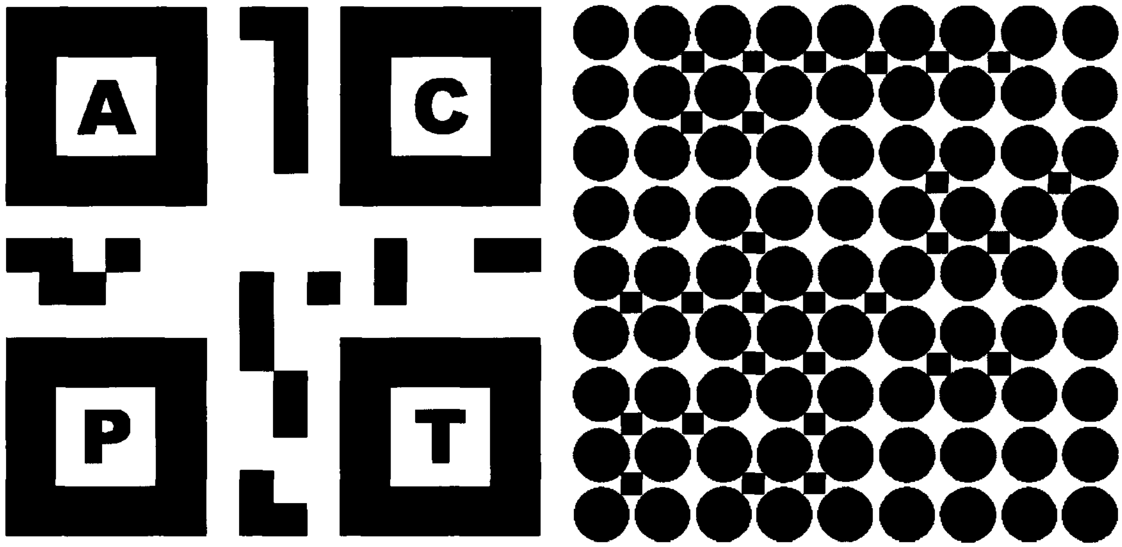


Figure 18. Block diagram of the MTag fiduciary marker system.

data. This property of the RS / CRC encoded data adds more robustness to each marker system.



(a) (b)

Figure 17. Two proposed fiduciary marker systems. *(a) 'MTag' using four sub-markers and RS-coded marker information. (b) 'CPRTag' using PRMVS pattern and RS-coded marker information.*

4.1. Two Proposed Fiduciary Markers

Fig. 17 illustrates the two types of proposed fiduciary marker systems. The fiduciary marker system shown in Fig. (a) is '*MTag*' (Multiple sub-marker Tag) and the marker shown in Fig. (b) is '*CPRTag*' (Color Pseudo-Random Tag). MTag consists of four sub-markers and cross shaped inner data region. Each sub-marker is formed with an English alphabet character and a black quadrilateral border. CPRTag has a RGB color

MTag can still provide four feature points to the marker system. The marker system can estimate the position of four vertices of MTag using geometrical information of the detected vertices and subsequently, a correction of the perspective distortion of the inner marker data area could be done with the estimated position of four vertices.

The data region consists of 60 small black and white blocks. The black small blocks represent binary symbol '1's and the white (or blank) blocks stand for '0's. The 60 bit data stream is retrieved from the data region according to the pre-defined bit positions and it is passed to the Reed-Solomon (RS) decoding / Cyclic Redundancy Check (CRC) procedure.

The RS code can recover the data bit loss or correct the data bit error up to 33.3% which means that MTag can be detected even when 5 data symbols (1 symbol = 4bits) are corrupted. Each data bit stream is composed of three code blocks; 3 symbols (12 bits) of Data block, 2 symbols (8bits) of CRC block, and 10 symbols (40 bits) of RS Parity block, totalling 15 symbols (60 bits).

Fig. 18 shows the block diagram of the MTag fiduciary marker system. The proposed marker is originally designed for the indoor localization purpose. Thus this marker system should be able to provide the camera pose information in an acceptable error range. The experimental results on localization performance of each proposed marker system is presented in next chapter. With the additional marker location database, the gray block in Fig. 18, containing the marker position with respect to the world coordinate system, the fiduciary marker system can provide the world coordinate position of the camera (or the user).

The other type of proposed fiduciary marker, called CPRTag, utilizes the pseudo-random multi-valued sequence to provide unique feature points instead of the quadrilateral black border used in most of the currently available fiduciary marker systems. One of the disadvantages of using the quadrilateral black border is the high marker detection failure rate even under the slight marker occlusions. Since this marker detection approach is based on the detecting the black quadrilateral blob in the gray scale image, if an object is superimposed on the marker, the quad is not shown as a quadrilateral in the gray scale image after binary thresholding procedure and the marker detection, consequently, fails.

To solve this problem, the pseudo-random pattern is used in the proposed marker system. In the PRMVS pattern described in Section 3.4.2, every possible 3 × 3 sub-pattern

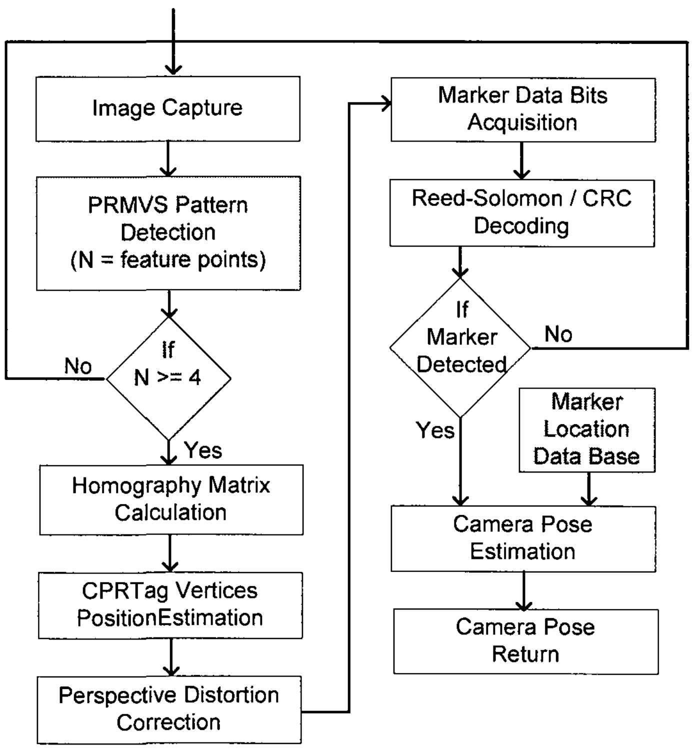


Figure 19. Block diagram of the CPRTag fiduciary marker system.

has an absolute position and this position information can be used to define the unique feature points of the marker instead of the four vertices of the quadrilateral border.

All 3 × 3 sub-patterns are stored as vectors taking 9 elements and each unique vector has the absolute position resides on the marker. In the case of the 9 × 9 CPRTag, the 49 3×3 sub-patterns can be defined which means that 49 feature points are present in the 9 × 9 CPRTag. Since the marker could be properly localized if at least 4 feature points are shown among 49, therefore, the CPRTag can achieve its robustness against the partial occlusions of the marker.

The data bits are located on the space among the color blobs, as shown in Fig. 17(b), to prevent the color pattern in the captured image being blurred by the black squares. The data block of CPRTag consists of 60 bits as MTag data block, in addition, its structure of the codeword is same with MTag.

Fig. 19 illustrates the block diagram of the CPRTag fiduciary marker system. The only difference between MTag system and the CPRTag system is the feature point detection algorithm in both block diagrams.

4.2. Sub-Marker Detection in MTag

As explained in previous Section, MTag consists of four sub-markers and RS / CRC coded data region. This MTag structure is designed in consideration to detect the marker in messy indoor environments therewith it allows overcoming the situations of partial occlusion which frequently occur in indoor environments.

The fiduciary markers are required to be detected reliably even under severely disordered environment. However, the ARToolKit shows a low performance in such environment as showed in [20][24][25][29] since the ARToolKit relies on simple template matching algorithm to detect the markers. This simple correlation method can reduce computational cost, however, it can result in a high marker detection failure rate.

Due to this disadvantage of the ARToolKit, the proposed marker system, MTag, adopted artificial neural network (ANN) as a sub-marker detection engine. Few extracted features of the inner pattern are fed into the ANN, and the pattern ID is produced as an output of the ANN. For distinctive inner patterns, four different English alphabets were used.

The ANNs are trainable and dynamic systems that can estimate input-output functions [51][52]. Due to the ANN's characteristic to mimic the human perception process, it can solve non-linear classification problems. The character recognition is a highly non-linear problem where multi-dimensional feature space exists so the ANN could be a good solution for the problems. As stated in Section 3.2.3, the 12 distinct features from character images can be derived from 16 features introduced in [68]. Since 26 English alphabets are classified with 12 dimensional features, the character recognition procedure turns out a highly complex and non-linear problem. The multi-layered perceptron (MLP) with a hidden layer can solve this arbitrary decision (classification) problem.

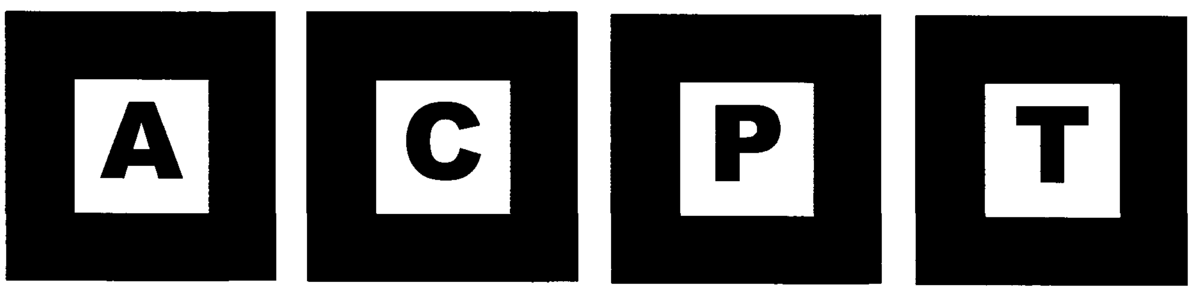


Figure 20. Four sub-markers used in the MTag shown in Fig. 19.

Four sub-markers consist of a black rectangle border and an alphabet letter as shown in Fig. 20. The black rectangle border of the marker, that is a marker template, is used as a sub-marker recognition beacon. The size of the outer rectangle can be defined by a user but for the indoor localization the marker size should be large enough so the marker can be detected in a further range. The edges of the outer rectangles of four sub-markers are 16 feature points and used for calculation of 6-DOF camera pose. In addition, the inner pattern area is extracted by using geometrical information of the 4 vertices of a marker template.

The sub-marker detection and letter recognition processing flow is illustrated in Fig.

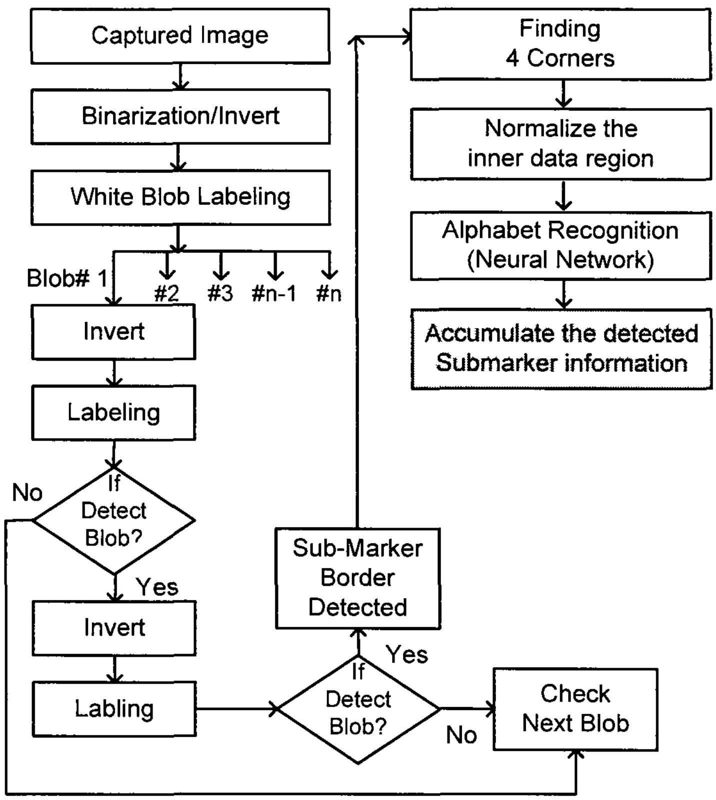


Figure 21. Block diagram of the sub-marker detection procedure.

21. First, an image captured by the vision sensor is binarized with adaptive thresholding method. Then, the binarized image is color-inverted. Further, if the newly converted white areas have bigger than the given pixel size and a proper ratio *r (height/width)*, then the areas are labeled. Since the makers are composed of three blobs as it is shown in Fig. 20; the outer black boarder, the inner white rectangle, and the alphabet letter, the marker could be detected in a unique way by doing blob labeling three times hierarchically even in the messy or disordered indoor environment (The probability of the presence of the similar structured objects are very low in reality). If the marker is detected, four corners of the marker would be found in this order.

Up to now, although the marker has been detected, it is not validated with the inner pattern. In order to confirm that the detected object is a marker, the inner pattern of the detected quadrilateral is required to be recognized as one of the 26 English alphabets. If the detected marker is proven to be one of the user-defined markers, then, the four feature points are accumulated in a stack to be used later as four correspondences of the homography transformation.

The Optical Character Recognition (OCR) of the fiduciary marker system is processed with the multi-layered perceptron artificial neural network that is trained by the backpropagation method. The ANN used in this OCR procedure consists of an input layer, a hidden layer, and an output layer. The input layer has 12 nodes (neurons) corresponding to a number of alphabet features and the output layer has 26 nodes corresponding to a number of English alphabets.

A useful guide to the maximum number of layers comes from the Kolmogorov Existence Theorem and the discussion in [72][73] where it is argued that one does not need more than two hidden layers for processing real valued input data, the accuracy of the approximation is controlled by the number of neurons per layer, and not the number of layers. However, this argument does not necessarily mean that the two hidden layered MLPs are the most efficient MLP. A research has shown that only three layers are enough to get a satisfactory result in complex approximation problems, and the accuracy of the approximation depends on the number of neurons in the hidden layer [74].

Based on above discussion, the number of hidden layer is set as one and the number of nodes on the hidden layer is empirically set as hundred for the OCR ANN. The character recognition rate in the testing phase shows that the ANN with one hidden layer consists of hundred nodes achieving non-linear decision ability in high accuracy.

In the training phase, four different alphabet fonts are used to generate the basis of the training set. Further training sets are produced by adding random noise, corrupting the shape, and tilting the basis. 25% of the training images are randomly tilted at angles between 20 deg. and -20 deg., another 25% of the training images are edge blended and tilted, another 25% of the images are partially deleted and tilted, and remained 25% of the images are tilted, edge blended, and partially deleted. 1,000 training images are generated for each character, totaling 26,000 images.

Then, 12 features of each alphabet character from an image are extracted with given 12 criterions described in Section 3.2.3 and stored for the ANN training. As a result of the ANN training with 26,000 images, the connection link weights are computed and stored as a XML format file. Fig. 22 shows some sample images of the training set and Table 2 describes the 12 features of the 4 sample images.

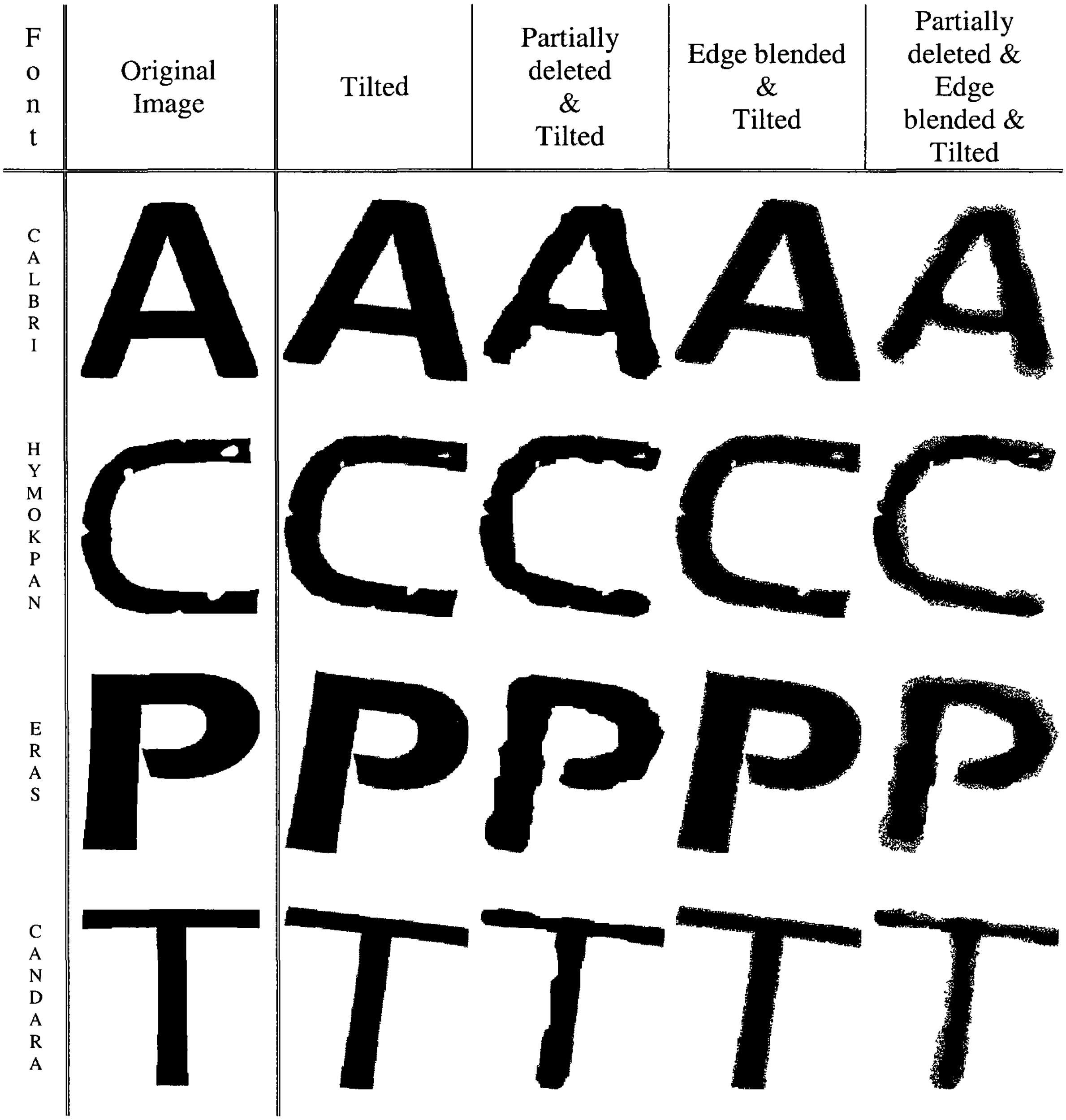


Figure 22. Some sample alphabet images for the ANN training

Fig. 23 illustrates the marker detection and pre-processing procedure for OCR. After the marker detection phase (Fig. 23(b)), the inner quadrilateral is unwrapped as a rectangle image through the affine transformation (Fig. 23(d)). Then, the alphabet blob is trimmed into a smallest box that the blob can fit into. The trimmed alphabet blob box is

TABLE II
EXAMPLES OF THE 12 NUMERICAL ALPHABET FEATURE DESCRIPTIONS

#				
1	15059	14637	16705	12239
2	114	82	85	93
3	104	97	87	81
4	36601690	39635128	53505413	17714550
5	38701938	58679187	41221524	40487698
6	-1281512	170699	-14626121	-3837776
7	1106188096	-76184265	414142503	-742628620
8	360515030	1757896939	-965940075	351229952
9	1685	1341	1499	1223
10	1470	1432	1465	1175
11	186601	124526	132846	109008
12	153140	127219	130348	83957

normalized as a user defined square size image for further processing (Fig. 23(e)). The normalized alphabet image produced by the image pre-processing algorithm is presented in Fig. 23(f). After the character image normalization, 12 features are extracted and fed into the ANN for the character recognition.

The performance of the ANN character recognition algorithm is evaluated in the perspective of the recognition rate and the processing time with four different size images (200 × 200, 100 × 100, 50 × 50, 25 × 25) of the all alphabets. Consequently, the average recognition rates are 99.34%, 98.81%, 97.57%, and 96.24% for each case of 200 × 200, 100 × 100, 50 × 50, and 25 × 25 pixel size images respectively. As shown in Table 3, except the cases of B, D, H, M, N, and Q, more than 95% recognition rates are achieved in overall cases. Besides, the recognition rates are more than 95% upon all alphabet images when it is evaluated with images equal or larger than 100 × 100 pixels size except the the case of character Q.

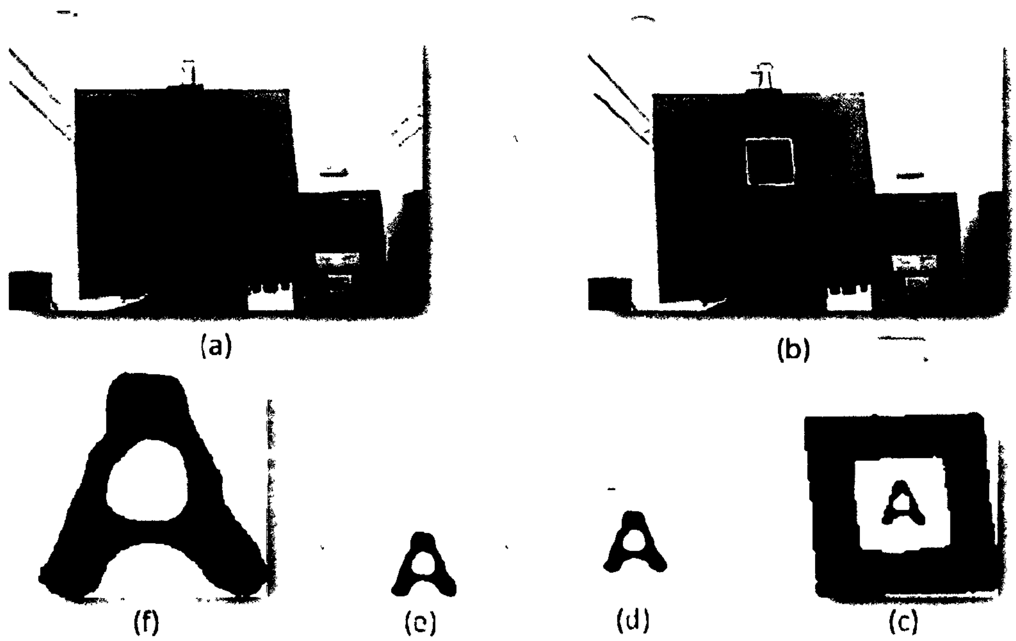


Figure 23. Marker detection and character extraction procedure *(a) Captured image. (b)Marker border detection. (c)Thresholded binary image. (d)Perspective distortion correction. (e)Character image extraction. (f) 200×200 pixels normalized character image.*

Fig. 24 shows the overall processing time of the four different sizes of all alphabet images. The processing rates are 15.8Hz, 17,2Hz, 18.2Hz, and 21.2Hz in the case of 200 × 200, 100 × 100, 50 × 50, and 25 × 25 pixel size images respectively. Since the size of an image affects the feature extraction time, the smaller image gives shorter processing time. However, for the reliable system the size of the normalized image is suggested to be larger than 25 × 25, because 25 × 25 pixel sized images do not provide good recognition rate as it is validated in the recognition rate evaluation. Therefore, for a better performance of the sub-marker detection algorithm, the size of the normalized character image is decided to be 200 × 200 pixels. In addition, four characters (A, C, P, and T) are selected among 20 characters showing acceptable recognition rate to generate the MTag structure with four sub-markers. However, any possible combinations of four characters among twenty can organize the MTag structure as well.

TABLE III
ALPHABET RECOGNITION RATE(%) WITH VARIOUS IMAGE SIZES

Alphabet	Alphabet Image size(Pixels)			
	200×200	100×100	50×50	25×25
A	100	100	98.2	97.5
B	98.6	97.4	96.3	93.2
C	100	99.1	98.2	96.8
D	98.4	97.5	96.1	93.6
E	100	100	98.9	97.1
F	99.8	99.2	98.0	96.9
G	99.4	98.9	98.9	96.8
H	98.3	97.1	95.9	92.7
I	100	99.9	99.5	99.1
J	100	99.7	98.2	97.9
K	100	99.9	98.3	98.0
L	100	99.5	98.7	98.1
M	97.1	96.8	93.1	91.1
N	97.5	96.5	94.9	92.1
O	99.8	99.3	98.0	97.6
P	100	99.0	98.6	97.2
Q	94.5	94.1	91.6	90.6
R	100	99.8	99.1	98.2
S	100	100	98.9	98.6
T	100	100	99.1	98.7
U	99.9	98.3	97.9	96.2
V	99.9	98.7	97.3	96.4
W	99.6	98.9	97.8	96.0
X	100	99.9	98.8	97.9
Y	100	99.7	98.4	97.1
Z	100	99.9	98.3	97.0
Average	99.34	98.81	97.57	96.24

The MTag arranges four sub-markers to the four corners of the rectangle marker template as shown in Fig. 17(a). This geometric placement of the sub-markers allows that the templates of MTag could be detected even under the circumstance where three of the four sub-markers are hidden or occluded. The geometric information of four sub-markers with respect to the origin of the MTag coordinate is stored in a configuration file and used to find correspondent marker coordinates of the detected sub-marker feature points. The example configuration file storing sub-marker information is described in Fig. 25;

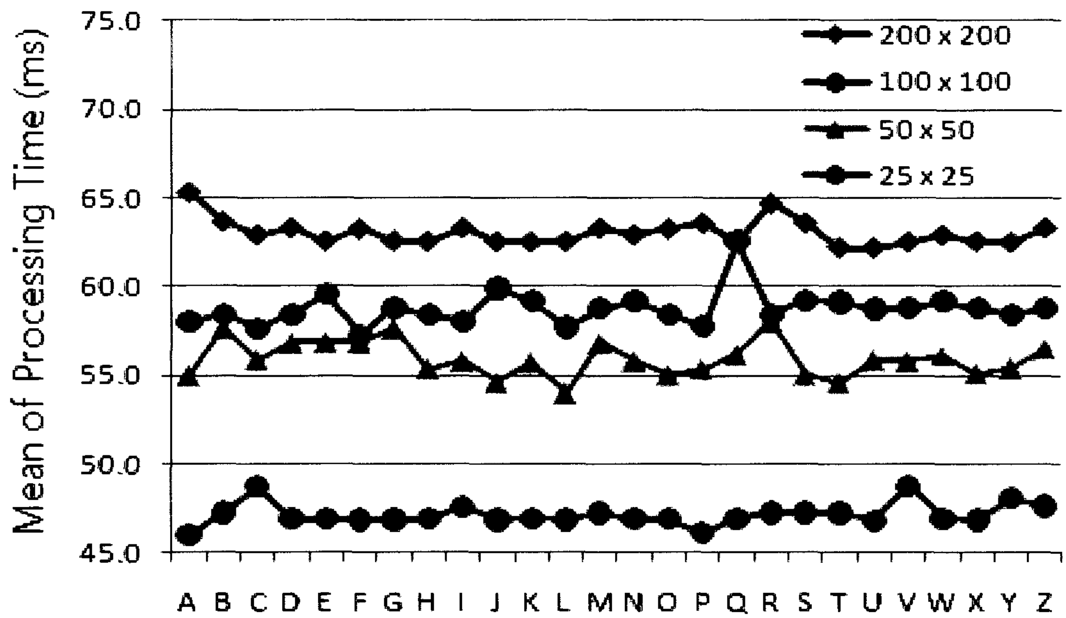


Figure 24. Alphabet recognition processing time of four different size images

marker 1 character : A
Marker width & height : 66.0
Coordinate origin : 0.0 0.0
Pattern transformation matrix :

1.0	0.0	0.0	-55.0
0.0	1.0	0.0	55.0
0.0	0.0	1.0	0.0

marker 2 character : C
Marker width & height : 66.0
Coordinate origin : 0.0 0.0
Pattern transformation matrix :

1.0	0.0	0.0	55.0
0.0	1.0	0.0	55.0
0.0	0.0	1.0	0.0

marker 3 character : T
Marker width & height : 66.0
Coordinate origin : 0.0 0.0
Pattern transformation matrix :

1.0	0.0	0.0	55.0
0.0	1.0	0.0	-55.0
0.0	0.0	1.0	0.0

marker 4 character : P
Marker width & height : 66.0
Coordinate origin : 0.0 0.0
Pattern transformation matrix :

1.0	0.0	0.0	-55.0
0.0	1.0	0.0	-55.0
0.0	0.0	1.0	0.0

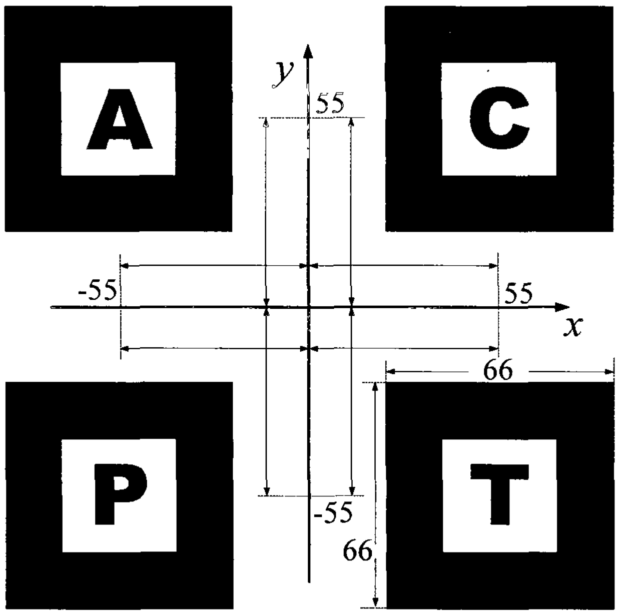


Figure 25. Sub-marker configuration file content and the geometric relationships of four sub-markers with respect to the MTag coordinate

4.3. PRMVS feature points detection in CPRTag

In order to assign at least four correspondences (feature points with known geometrical information) which are used for homography matrix calculation to a fiduciary marker, most of the currently available markers place a black quadrilateral as a marker border. By detecting four corners of the black quadrilateral, the requirement of minimum number of correspondences could be satisfied. However, this approach is vulnerable to the partial occlusion of black quadrilateral border, since the four feature points may not be found when the border detection is failed from an occlusion. Thus, new approach to assign correspondences to a fiduciary marker is developed and applied to CPRTag system.

CPRTag (Color Pseudo-Random pattern Tag) consists of the pseudo-random multi-valued sequence (PRMVS) and the RS/CRC coded data matrix. The PRMVS pattern could provide enough feature points with known geometrical information of feature points so that the 6 DOF pose of the CPRTag can be estimated with an acceptable accuracy.

As described in Section 3.4., the pseudo-random encoded patterns have been widely used in the field of machine vision or computer vision. Due to its unique window property, the PR pattern could quantize a real world by the pre-defined size of the PR pattern while the pattern being projected [18][19][61]-[65] or physically fixed [14]-[17] onto the real world.

The perfect sub-map (PSM) pattern developed by Morano [65] is a successful pseudo-random multi-valued sequence consists of square matrix M of size $L \times L$ ($k = l = L$). Each element m_{ij} within M is assigned one value from a palette of possible multiple values. In [19][65], this pattern projected onto the environment to assign numerous correspondences to the real world so that the captured 3D scene can be recovered as a 3D graphic scene. Each element of the square matrix M could be uniquely defined with 8 neighbour elements within a 3×3 window, and it holds its own geometrical information with respect to the pattern coordinate. The three dimensional information of the captured scene with the projected PSM pattern is then, calculated by using simple triangulation of three positions of the original pattern, the captured pattern, and the projected pattern.

The PSM pattern could be projected onto the real world as stated above but it could be fixed onto the real world as well. If the pattern is installed on the object surface, the homography transformation is used to retrieve the geometrical information from the captured scene instead of the triangulation. The PSM pattern on the object surface provides enough feature points with known geometrical information, thus, the captured object could be represented with the 3D information with respect to the world coordinate.

A 9×9 PSM pattern is used to construct 9×9 PRMVS pattern of the CPRTag but any other PSM patterns produced by the pattern generation rule could be used for the CPRTag. Fig. 26 shows a captured scene with a 9×9 PRMVS pattern used in CPRTag. The three multiple values used in the PRMVS pattern are represented as three different colors; red, green, and blue, and 81 dots consist of the 9×9 pattern. The radius of each dot is 1.91cm. An empty space created by 2×2 dots is reserved for a data bit, and total 60 bits are placed among 81 color dots. Except the 32 dots in outer layer, 49 dots could be contained in 3×3 window if the window slides over the PRMVS pattern so that each of 49 dots can be uniquely recognized with 8 neighbour dots after color filtering. The 9 elements of each 3×3 pseudo-random color pattern could be described as a vector $V_{(ij)}$ shown in eq. (52) and 49 unique sub-patterns of the CPRTag could be listed as shown in Table 4.

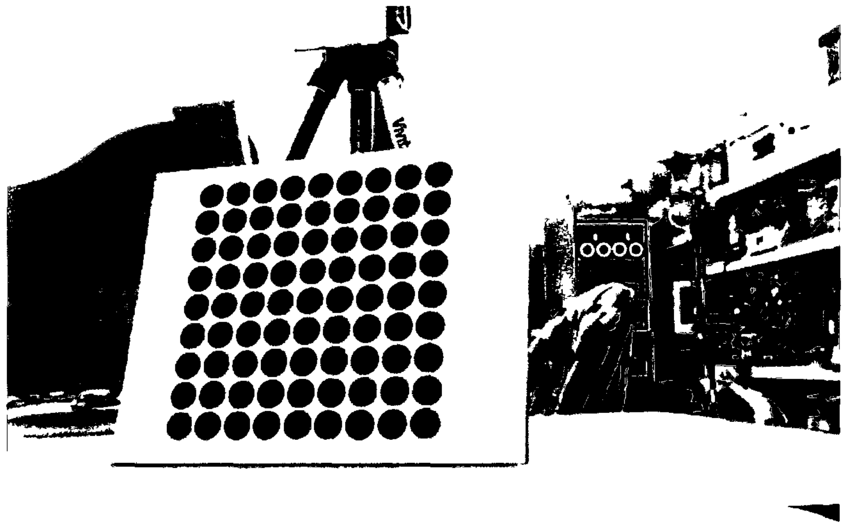


Figure 26. 9 × 9 PRMVS pattern captured in a color image. *Three colors; red, green, and blue, are used to construct PRMVS pattern*

In order to extract geometrical information of the PRMVS pattern, all three colors should be reliably filtered from the captured color image. There are various robust methods for the reliable color segmentation, however, most of them are not adequate for a real-time application. Since the indoor localization with CPRTag should be done in real-time, computationally efficient color segmentation algorithm is required. Therefore, to achieve a relatively reliable and fast color segmentation performance, here, a simple thresholding method is used on two different color spaces to isolate the colors of interest in an image.

In fact, color information is widely used in marker detection since the color is an easily acceptable and computationally inexpensive feature. However, since the color information modeling involves three main factors, an illumination, object surface characteristic, and imaging device, the process is not trivial.

According to the image formation, the RGB values are device and illuminant dependant so the color is not an intrinsic property of an object itself. Thus, whether the image is illuminated with unknown light source or illuminated un-uniformly, the color detection of an image is prone to the error.

Many researchers have sought methods that are insensitive to illumination for object detection and recognition. However, still there is no absolute solution for the robust color segmentation. Generally, the color segmentation based on color invariant shows a better performance in practical applications while the method with color constancy is a

TABLE IV
49 SUB-PATTERN VECTORS WITH 9 COLOR ELEMENTS OBTAINED BY 3×3 WINDOWING ON 9×9 CPRTAG

Sub-patterns		1	2	3	4	5	6	7	8	9	Sub-patterns		1	2	3	4	5	6	7	8	9
1	V_{22}	R	R	B	G	R	R	G	G	G	26	V_{56}	R	R	G	R	B	G	G	B	B
2	V_{23}	R	B	G	B	G	R	G	G	R	27	V_{57}	R	G	B	G	R	R	B	B	G
3	V_{24}	B	B	R	G	B	R	G	R	B	28	V_{58}	G	B	R	B	G	G	B	G	G
4	V_{25}	B	B	B	R	G	B	R	B	R	29	V_{62}	R	G	G	G	G	G	B	R	G
5	V_{26}	B	B	B	B	R	B	B	R	G	30	V_{63}	G	R	G	G	G	G	R	G	R
6	V_{27}	B	G	G	B	B	B	R	G	R	31	V_{64}	R	G	G	G	G	G	G	R	B
7	V_{28}	G	G	R	G	B	B	G	R	G	32	V_{65}	G	B	R	G	G	G	R	B	B
8	V_{32}	G	G	R	R	R	G	G	B	B	33	V_{66}	B	B	R	R	G	G	B	B	G
9	V_{33}	G	R	B	R	R	G	B	B	R	34	V_{67}	B	G	G	R	R	R	B	G	G
10	V_{34}	R	B	B	B	R	G	B	R	B	35	V_{68}	G	G	B	G	R	R	G	G	R
11	V_{35}	B	R	B	B	B	R	R	B	R	36	V_{72}	R	G	G	R	G	G	G	R	B
12	V_{36}	R	G	B	B	B	B	B	R	G	37	V_{73}	G	R	R	G	R	R	R	B	B
13	V_{37}	G	R	G	B	B	R	R	G	B	38	V_{74}	R	B	G	R	G	G	B	B	R
14	V_{38}	R	G	G	G	B	G	G	B	G	39	V_{75}	B	B	B	G	R	R	B	R	B
15	V_{42}	B	B	G	G	G	G	G	G	G	40	V_{76}	B	G	B	B	G	G	R	B	R
16	V_{43}	B	R	R	G	G	B	G	G	G	41	V_{77}	G	G	G	B	B	B	B	R	G
17	V_{44}	R	B	B	R	G	B	G	G	G	42	V_{78}	G	R	G	G	B	B	R	G	B
18	V_{45}	B	R	R	B	R	R	G	G	R	43	V_{82}	R	B	G	R	B	B	G	G	B
19	V_{46}	R	G	G	R	B	B	G	R	R	44	V_{83}	B	B	R	G	R	R	G	B	B
20	V_{47}	G	B	R	G	R	R	R	R	G	45	V_{84}	B	R	B	R	B	G	B	B	G
21	V_{48}	B	R	G	R	G	G	R	G	R	46	V_{85}	R	B	B	B	R	R	B	G	B
22	V_{52}	G	G	B	B	G	G	G	R	G	47	V_{86}	B	R	G	B	B	B	G	B	B
23	V_{53}	G	G	R	B	B	G	R	G	R	48	V_{87}	R	G	G	G	B	B	B	B	G
24	V_{54}	G	G	G	R	B	G	G	R	G	49	V_{88}	G	B	R	G	G	G	B	G	B
25	V_{55}	G	R	R	G	R	G	R	G	G											

more powerful solution in theory [74][75]. Thus, the color segmentation method is done with the color invariant approach in this thesis.

The HSV color space suggested by Smith is a cylindrical space and can be modeled as a haxcone turned upside down. The top plane of the hexcone corresponds to V (value) = 1, and contains a maximum intensity colors. The point at the apex is black and has the coordinate $V = 0$. A Hue is the angle around the vertical axis, with red at $H = 0$. Note that complementary colors are 180 deg opposite to one another as measured by hue. The value of S (saturation) is a ratio, ranging from 0 on the V axis to 1 on the triangular sides of the hexcone. The point $S = 0$, $V = 1$ is white. The V (value) is used only to define the brightness of a color. Since the color is described by only two variables, namely, Hue and Saturation, the HSV color space is relatively common and intuitive.

The normalized RGB (NRGB) color space is obtained from RGB using simple normalization as below

$$r=\frac{R}{R+G+G}, \qquad g=\frac{G}{R+G+G}, \qquad b=\frac{B}{R+G+G}. \tag{61}$$

The three normalized components, r, g, and b are called pure colors. The coordinates (r,g,b) are the simplest example of color invariance coordinates. They are invariant to changes in lighting geometry as well as to overall change in intensity of the incident light, as they contain no information about luminance. It can also be deduced from the preceding equations that a sum of the three components is always 1, so it is enough to use only two components r and g to completely describe the marker color space.

The color segmentation based on HSV color space can deal with different types of illumination to some extent but it often fails under great illumination changes. However, the NRGB color space is insensitive to the intensity of the illumination and can cope with

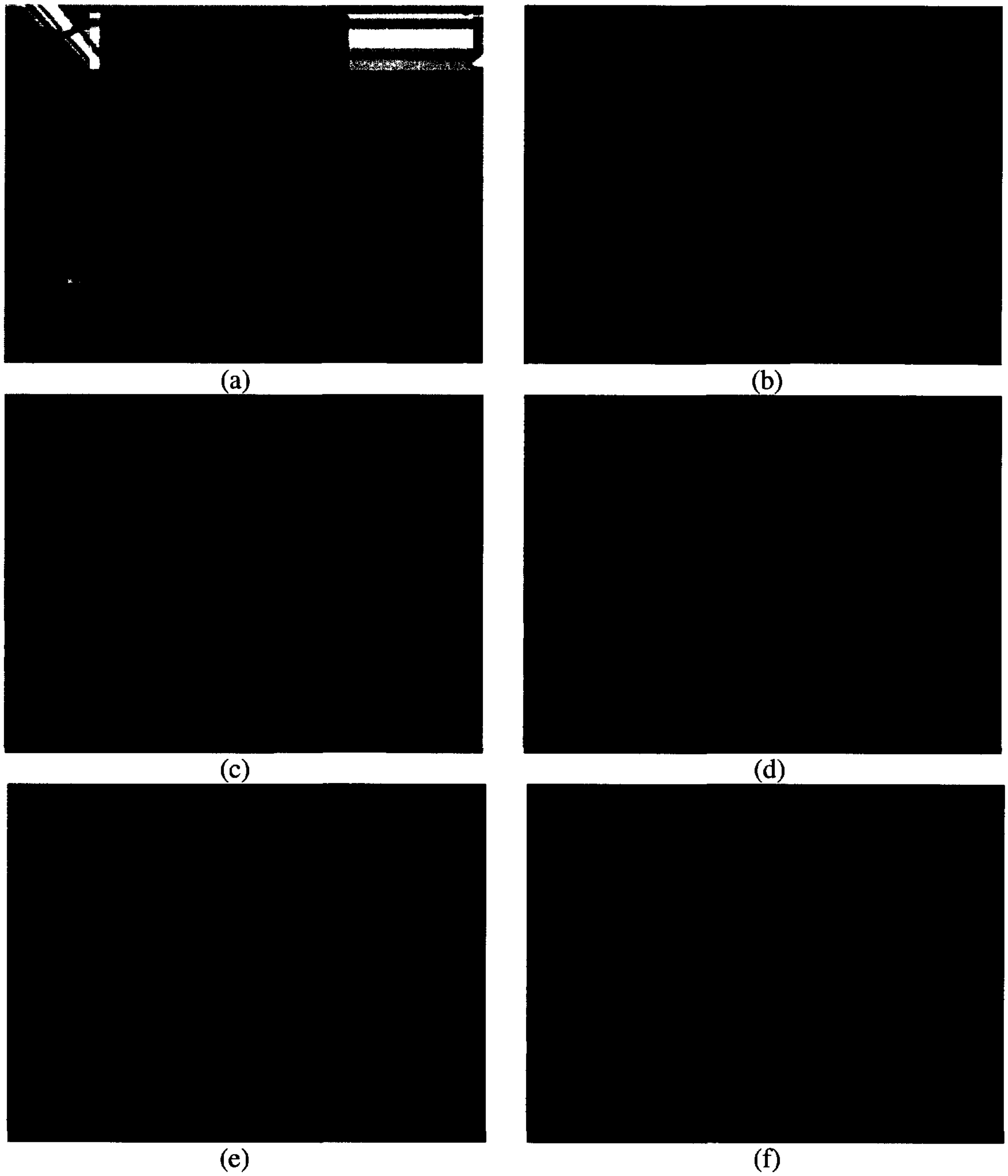


Figure 27. Image processing procedure for the following code validation. (a) Un-distorted captured Image. (b) Thresholded three color image. (c) After small blobs are discarded (d) After big blobs are discarded. (e) After unproper ratio blobs are discarded. (f) After the ‘opening’ operation(still has incorrectly segmented color blobs.)

the situation to some extent. Thus, above two color spaces are incorporated into the color segmentation algorithm of the CPRTag to solve the illumination variations of a scene.

This combination of the two color segmentation methods, the HSV and the NRGB color space, can provide a better result on the color segmentation than either HSV or NRGB method.

In spite of the effort to discriminate the color blobs properly, some color blobs, however, can be incorrectly segmented due to a strong non-uniform intensity distribution. In order to avoid this situation while ensuring a reliable PRMVS decoding performance, the segmented blobs are further examined with statistical criteria. Fig. 27(b), (c), (d), (e) shows the statistical analysis procedure of a color segmented image. First, the segmented color blobs are checked with size constraints. Color blobs bigger than certain pixel size are discarded. In addition, very small color blobs are regarded as noise and discarded as well. Then, a ratio of the blob width and height is checked, and if either width or height is much longer than the other, the blob is also discarded.

Fig. 17(e) represents an image still contains incorrectly segmented color blobs after the statistical analysis. Some blobs are merged and some blobs have non-circular shape due to a skew created by variable surface orientation. In order to split those connected blobs and generate proper shaped color blobs, two morphological operations are done on the color segmented image. First, the erosion is performed recursively to disconnect the incorrectly merged color blobs and then, the dilation is applied to the image to fill the pitted areas of color blobs. These two consecutive operations, the erosion followed by the dilation, are called 'opening' operation. By doing this operation few times undesired shape of blobs such as merged color regions, blobs with holes, or non-circular blobs could be corrected to some extent. The remained unwanted color blobs are, then, excluded from the set of candidate PRMVS color dots to ensure the accuracy of the following PRMVS decoding and pose estimation procedures. Fig. 27(f) illustrates the three colored image after the opening operation. In this image, two consecutive green color dots are still connected so those two blobs are excluded from the set of the candidate PRMVS color dots.

The image coordinates of the detected color blobs need to be retrieved from the color segmented image since the relative positions (distances and orientations) of one color blob with respect to the other remained color blobs are used to find 8 nearest color blobs which are necessary for generating the 3×3 sub-pattern vector shown in Table 4. For this, the center of each labeled feature points is calculated using the centroid method [76].

After all of the pre-processing procedure described above, the PRMVS decoding procedure can be launched to find the 49 feature points of the CPRTag. This procedure is called 'Pseudo-Random Code Validation'. As previously explained, the feature points can be uniquely defined with the 8 nearest neighboring dots. A created vector which consists of 9 elements, a feature point and 8 neighboring dots, in 3 × 3 window exists only once among various possible combinations. However, if we only lean on 8 nearest color blobs, some of the feature vectors cannot be extracted correctly since there might be missing or un-properly segmented color blobs among the obtained candidates. Therefore, the angular search method needs to be adopted to align the nearest 8 neighbor candidate color blobs. Since the 8 neighbor blobs should be distributed uniformly 360 deg. around the center color blob, any color vector with a non-uniform angular distribution of the 9 elements are regarded as an error. Only those 9 elements satisfying distance and angular distribution check is stored in the stack as a form of a vector. In addition, the angular positions of 8 nearest blobs are stored as another type of feature vector. These two vectors are used to validate the PR code in next step.

After generating the feature vectors based on their geometrical proximity to a given color blob, a code validation is performed by computing a hamming distance of the

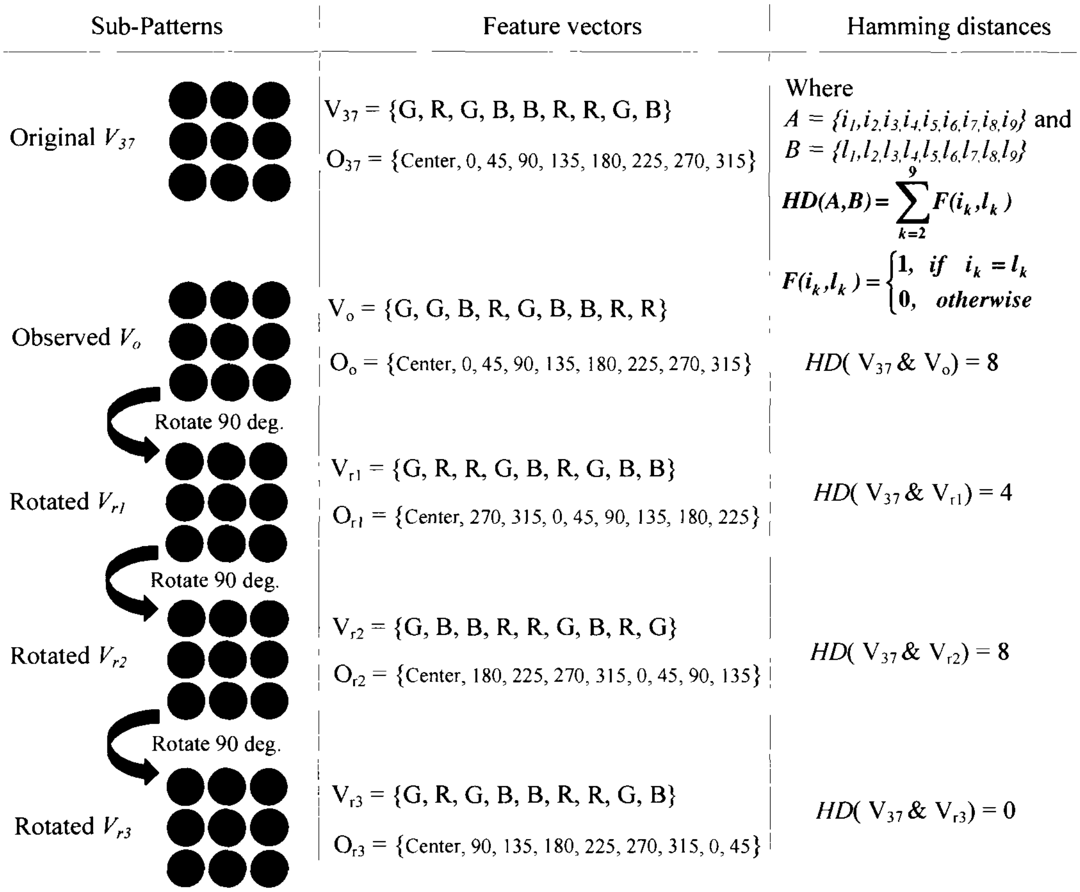


Figure 28. An example of PRMVS code validation procedure. *The pattern located on 3th row and 7th column is captured in an image and validated by the Hamming distance.*

obtained 9 element feature vectors by comparing its pattern with the original PRMVS pattern. Previously calculated orientation feature vectors are used to match the obtained feature vectors to its original feature vectors. If zero hamming distance is resulted, the desired sub-pattern number is returned with the direction of the obtained pattern. The Fig. 28 shows the code validation procedure. The obtained pattern V_o is compared with all sub-patterns in the database. If the pattern does not match to any of them with zero hamming distance, the obtained one is rotated by 90 deg. counter-clockwise and compared recursively until it finds its matched pattern. After rotating V_o 270 deg., the pattern is matched to the V_{37}, thus, it is said that the direction of the CPRTag is 270 deg. or -90 deg. equivalently.

If more than 3 codes are detected via the code validation procedure, the 6 DOF camera pose can be estimated with a known position of those detected sub-patterns. As explained in Section 3.1.2. and 4.2., the feature points obtained from an image with known position information with respect the marker coordinate system are used for calculation of the homography transformation matrix between image coordinate and marker plane coordinate. Then, with the known camera intrinsic parameters and camera distortion parameters, the rotation matrix and translation vector can be calculated.

By using PRMVS pattern, CPRTag can provide enough feature points even when half of the marker is occluded. Due to such strength of using PRMVS pattern, the CPRTag can overcome the marker detection failure problems of partial occlusion which most of the current marker systems suffer from.

4.4 Achieving Robustness against Data Loss/Error with RS/CRC coding

The famous marker system, ARToolKit, provides planar fiduciary markers that are feasible for printing and manipulating. The marker recognition is done by detecting quadrilateral black border followed by correlating an interior pattern with the known patterns. Since the correlation method in pattern recognition uses a 16×16 feature vector extracted from the inner pattern of the marker, if the shape of inner pattern is corrupted or partially occluded, a false marker detection rate such as false positive, false negative, and inter-marker confusion rate, is easily increased. Also, the false marker detection rate increases as the library size increases [20] since the uniqueness of each marker deteriorates with larger number of stored patterns.

There are few fiduciary marker systems using digital encoding techniques for more reliable marker recognition. ARTag shown in Fig. 4 is one of the most recently developed fiduciary marker system using digital technique. This adopted the Cyclic Redundancy Check (CRC) and Forward Error Correction (FEC) digital techniques to encode and decode marker information to and from the 36-bit code area of the marker. Among 36 bits, 10 bits are used for the marker ID and remained 26 bits are used for error detection and error correction of the data reading. Due to these techniques, the ARTag is capable for an error correction up to 2 bits error out of 36 bits and can detect most of the erroneous bit readings. This relatively robust marker identification method might be enough for most of the AR applications. However, the ARTag does not provide enough markers (ARTag library size = 2002) and error correction performance because the total number of bits can be recovered from noisy data reading is only two bits. Therefore, the two marker systems, MTag and CPRTag, expand the data encoding and decoding method to support indoor localization usage of those markers.

The digital techniques used by two proposed marker systems are inspired by the QR code [67] equipped with Reed-Solomon code (RS code) for the error correction. The RS code is originally designed for communication systems and data storages to recover data from possible errors [3]. The encoding and decoding of the marker information by using RS code allow the fiduciary marker to be robust against partial occlusions, possible data bit loss or data bit reading failures when the marker data is read.

As stated in Section 4.1., the interior data region consists of 60 bits, including 12 bits encoding marker ID, 8 bits for CRC check bits, and 40 bits for RS code check bits. As defined in the RS-code, consecutive 4bits compose a symbol so the marker ID data, the CRC data, and the RS code parity data compose 3 symbols, 2 symbols, and 10 symbols respectively, totalling 15 symbols. Fig. 29 presents the structures of the inner data region of MTag and CPRTag. As described in Fig. 29, the 60 bits are arranged in the position of each bit number.

Once the homography transformation between the image coordinates and the marker plane coordinates is computed using detected feature points on the marker, a quadrilateral template can be located (projected) on the captured image using homography matrix and position information of the marker template. The inner pattern region, then, can be sampled with pre-defined grid as shown in Fig. 29. If a grid is occupied by black pixels

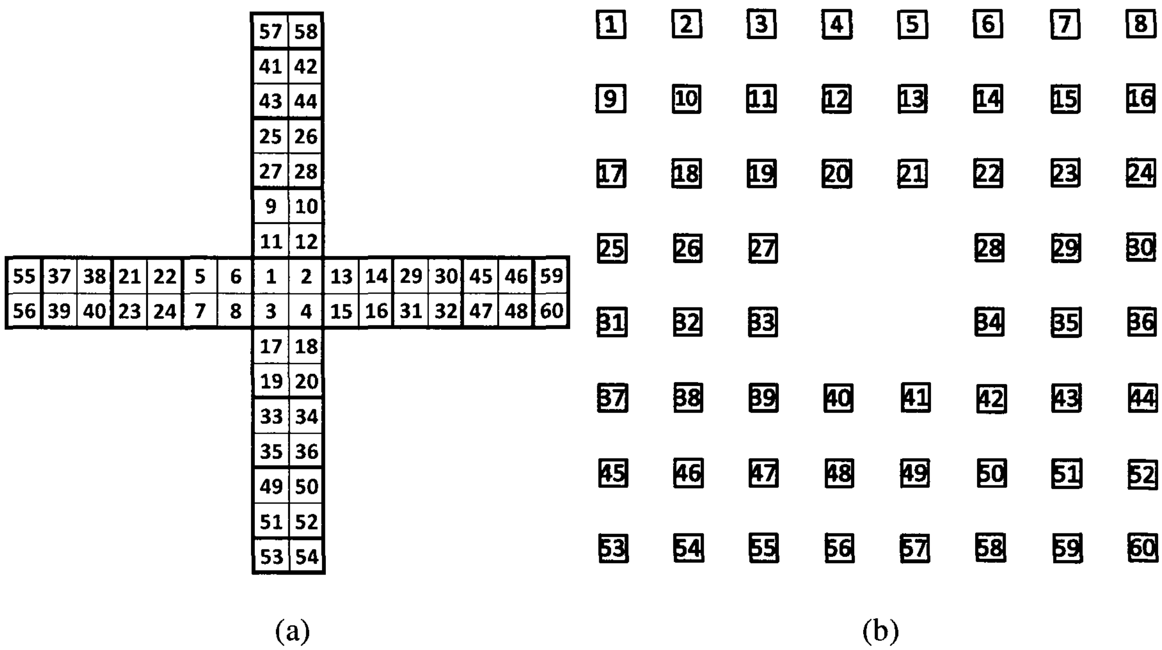


Figure 29. Structures of the two proposed fiduciary markers
(a) Data bit structure of MTag. (b) Data bit structure of CPRTag

over certain percentage, (normally 60% but if higher accuracy is required, the threshold value needs to be larger.) the grid is assigned with a digital symbol '1' otherwise symbol '0'. Through this grid occupancy test, 60bits are extracted from the inner data region of the marker image and validated in the decoding process.

The 12 bits-marker ID from marker 1 to marker 4096 is encapsulated into 60 bits using two digital techniques. These 60 bits are engraved into the inner data patterns illustrated in Fig. 29. This inner marker pattern generation is an encoding phase of the 12 bits marker ID information. The encoding phase is presented in Fig. 30. This process generates 4096 60bits-long pattern codes, and these binary pattern codes are engraved into the inner data region to create 4096 distinct fiduciary markers.

The CRC is originally developed by W. Perterson [77] to detect the error caused by a noisy or damaged transmission or storage media in a digital communication system. It calculates a short, fixed-length binary sequence, known as the CRC code, for each block of data and adds the sequence to the original data. CRC is named since the check code is a redundancy and the algorithm is based on cyclic codes. Its computation resembles a polynomial long division operation where the quotient is discarded and the remainder

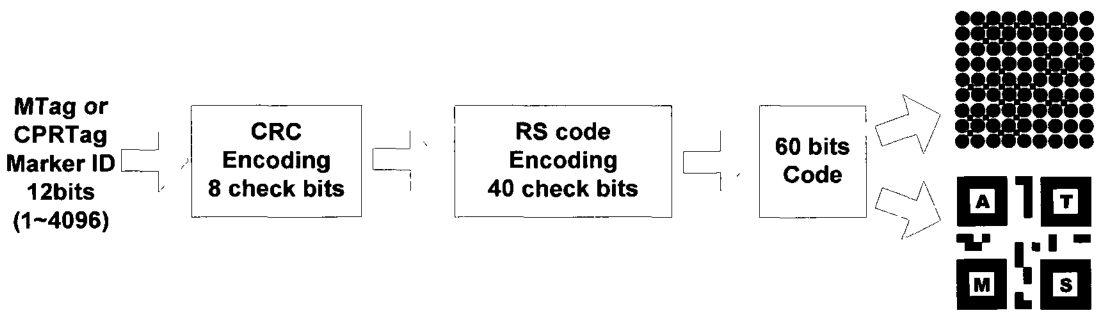


Figure 30. Digital encoding process to create inner patterns. *The 12bits marker ID information is encapsulated into 60bits pattern code with 8 CRC parity bits and 40 RS check bits. Then, the pattern code is located into the inner data region by the order described in Fig. 29.*

becomes the result. The polynomial coefficients are calculated based on Galois field GF(2) which is a field of two elements, 0 and 1.

A CRC-8 polynomial is applied to the 12 bits marker ID data to generate 20 bits data block. The 20 bits data block is, then, fed into the RS encoding algorithm to generate 60 bits RS / CRC encoded data block by adding 40 bits which are RS code check bits.

The CRC code is adopted in the fiduciary marker pattern encoding / decoding phase since encoding the marker ID only with RS code does not provide capability to alarm the false marker detection in the decoding phase if the error is over 33.3%. As it is mentioned, the RS code can correct up to 5 symbols error, however, if the error of the data reading is more than 5 symbols among 15 symbols, the RS decoding cannot recover the original data and results in incorrectly decoded data. To detect and inform that the decoded data is not restored correctly to the system, the CRC code needs to be added to the data block. In the decoding phase of the CRC the remainder of the division operation with CRC-8 polynomial must be 0, otherwise the code is not considered to be correctly decoded from the RS decoding algorithm or the read data from an image is not from a fiduciary marker but from a random object. The probability of a RS decoded 20 bits data block passes the CRC test procedure is only 0.39% ($\frac{1}{2^8}$) meaning that most of the erroneously decoded 20 bits data blocks could be discarded at the CRC test procedure.

The RS encoding / decoding procedure provides error detecting and correcting ability to the proposed fiduciary marker systems. In the RS decoding phase, the algorithm can recover the original code from the corrupted data up to 33.3% whereas the ARTag, which is using a forward error correction technique, can recover up to 8.3% (it can correct only 2 error bits among 36 bits). The RS encoded marker pattern solves the false negative

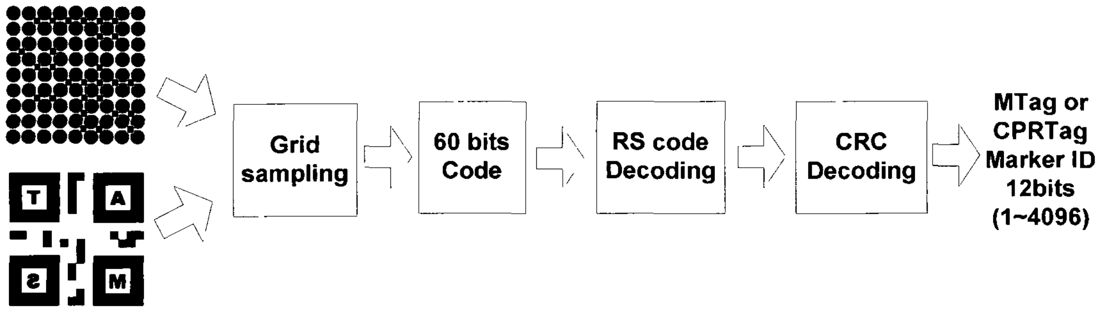


Figure 31. Digital decoding process to generate marker ID. *The interior pattern is extracted by the grid sampling procedure and re-arranged as a 60 bits code. Then, the code is decoded by RS / CRC decoding algorithms. This process allows detecting and correcting errors up to 33.3% of the 60 bits code.*

detection problem caused by the grid sampling error due to an imperfect threshold, misalignment of the detected marker template, specular reflections, partial occlusion, and image reading noise to some extent. Using RS code might increase the false positive rate but the presence of the CRC code in the marker data prevents the false positive detection caused by the RS code. Fig. 31 shows the marker data decoding procedure stated above.

5. Experimental results

This chapter presents results of fiduciary marker detection and localization experiments conducted with two proposed fiduciary marker systems. ARToolKit and ARTag marker systems are selected to be compared with the proposed ones since those two markers are currently the most famous fiduciary marker systems.

The marker detection rate can be evaluated with false positive marker detection rate, false negative marker detection rate, and inter marker confusion rate. For a reliable marker system, the marker system must show low rates in three detection criterion. The proposed marker systems are originally designed for the indoor localization so localization accuracy experiments were performed to validate its usefulness. The localization results must be acceptable and provide satisfactory accuracy with acceptable error for the indoor localization usage.

5.1. Experimental setup

The fiduciary marker detection system used in the experiment consists of a charger coupled device (CCD) digital camera, a laptop computer interfaces with the camera, and a fiduciary marker. The camera is Logitech C905 and firmly mounted on a tripod. The computer used in the experiments is equipped with Intel Core 2 Duo 997MHz processor, 2GB of RAM and Windows XP operating system. A fiduciary marker is selected from each fiduciary marker libraries among four different marker systems; ARToolKit, ARTag, MTag, and CPRTag thus the total number of markers used in the experiment is four. A marker is attached to a wall so the camera can capture images containing the markers. Fig. 32 shows the full experimental setup for testing several fiduciary marker systems.

The proposed marker systems are developed based on OpenCV 2.0 and ARToolKit and ARTag are based on the older versions of OpenCV and OpenGL. The software for the operation of two proposed fiduciary marker systems can be divided into 6 important parts as following; camera calibration, sub-marker detection using OCR with ANN, RGB color segmentation, PR pattern code validation, homography calculation, and pose calibration. For the experiment, the software is incorporated with ARToolKit software and ARTag software so that the experiment could be done on the same environments.

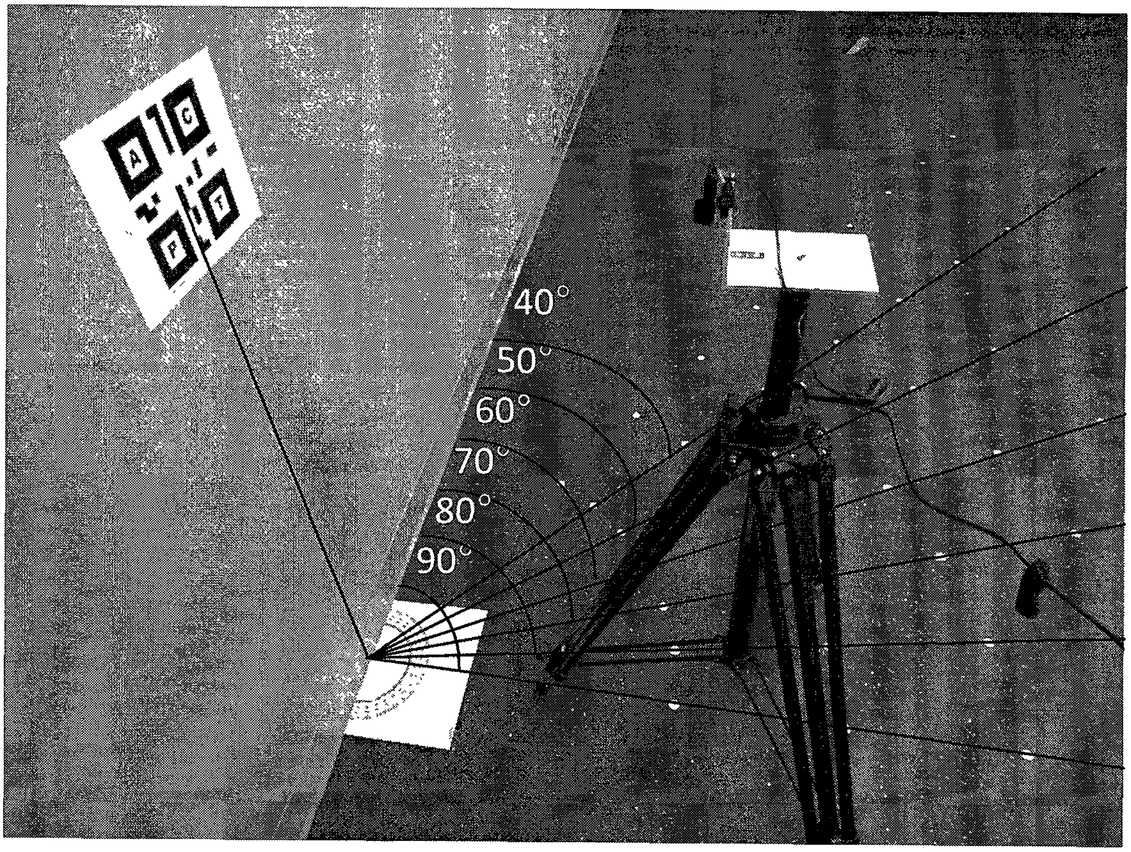


Figure 32. Experimental setup with a laptop, a web camera, and a fiduciary marker

5.2. Marker Detection Performance

5.2.1 False Positive Marker Detection

The two proposed fiduciary marker systems utilize the digital encoding and decoding techniques which are RS code and CRC code, and ARTag uses similar techniques with the proposed ones. ARTag encodes 36 bits in a inner marker data region and can correct erroneous 2 bits among 36 bits which equals to 5.5% error correction capacity and can be compared with the proposed marker systems which have 33.3% error correction capacity.

The false positive detection is one of the possible detection failures occurs once the existence of a marker is incorrectly reported when there is no marker in the captured scene. Due to the error correction ability of those marker systems, the false positive detection rate could be increased a bit since a random object which has similar frequency distribution of the interior region with the actual marker could fool the marker detection algorithm after the error correction method applied. However, based on the probabilistic study of the false positive marker detection, it is proved that a chance of erroneous marker detection report under non-existence of the marker is very low.

When the error correction capacity is 5 symbols among 15 symbols of a codeword (each symbol consist of 4 bits), each valid erroneous MTag and CPRTag with 15 symbols codeword could be mapped from $1 + 15\times15 + (15\times15 \times 14\times15) + (15\times15 \times 14\times15 \times 13\times15) + (15\times15 \times 14\times15 \times 13\times15 \times 12\times15) + (15\times15 \times 14\times15 \times 13\times15 \times 12\times15 \times 11\times 15) = 274{,}130{,}000{,}000$ codes since a erroneous symbol can have 15 error cases out of 16 possible codes produced by 4 bits. This can be restated as $274{,}130{,}000{,}000\times4096 =$ 112,280 billion 60 bit marker data which would force the system to report the existence of the marker. As a result, the probability of the false positive marker detection from a randomly decoded 60 bits marker data is 112,280 billion / 2^{60} = 0.00097%. Thus, the non-marker with a similar inner frequency distribution has 0.00097% chance of false positive marker detection and this means the false positive marker detection is statistically almost impossible. In the case of ARTag, the probability is 0.0078% and this is larger than the probability of MTag or CPRTag.

Based on this statistical result, an experiment with respect to the false marker detection was done with several image sequences which do not have any fiduciary markers. The ARToolKit and ARTag are compared with the proposed marker systems by running the marker detection algorithm on the same image sequences. ARTooKit provides confidence factor which defines a threshold level for the binarization of captured image and this factor severely affects the marker detection rate. Thus, the confidence factor of ARToolKit is also considered in the experiment. In addition, since the size of the marker library of ARToolKit can also affects the false positive detection rate, it is included in the experiment as well.

Four image sequences are recorded at different indoor environments and each video file is tested with ARToolKit, ARTag, and the proposed two marker systems. The image sequences include no markers so the detected marker is considered as false marker detection. As stated above, in the case of ARToolKit, the marker detection performance is compared under different sized libraries and different level of confidence factors to prove the imperfection of the marker detection algorithm. Table 5 presents the results of the false marker detection experiment. As size of the ARToolKit marker library increases, the false marker detection rate also increases since there are more chances that a random object is recognized as a ARToolKit marker. Therefore, the false positive detection rate increases.

On the other hand, ARTag, MTag, and CPRTag show zero false marker detection due to an advantage of using digital techniques for the marker detection.

One can suggest that using high confidence factor for ARToolKit would provide acceptable false marker detection rate, however, raising confidence factor could reject the actually existed markers and result in marker detection failure. Moreover, if one uses ARToolKit for the indoor localization application, the size of the marker library would be much larger than the experimental setup, so the false positive marker detection rate is increased and consequently the user may face a system breakdown.

The theoretical probability of the false marker detection for ARTag is 0.0078% and that of MTag and CPRTag is 0.00097% (still not zero) so the false marker detection is expected to be seen, however, since the marker templates and its spatial frequency could distinctively exist in the capture image, the false marker detection could be prevented after the marker template search procedure. As stated in previous chapters, ARTag collects the marker candidates by looking for black quadrilateral objects with unique frequency distribution of inner area, the MTag searches for black quadrilateral border and recognizes the inner data region with ANN OCR procedure, and the CPRTag filters the color image

TABLE V
EXPERIMENTAL RESULTS ON FALSE POSITIVE MARKER DETECTION

Image sequences		Number of false positive detections										
		ARToolKit								ARTag	MTag	CPRTag
Type	Frames	Marker library size = 10				Marker library size = 20						
		c.f.=0.6	c.f.=0.7	c.f.=0.8	c.f.=0.9	c.f.=0.6	c.f.=0.7	c.f.=0.8	c.f.=0.9			
A	1194	11	1	0	0	18	3	1	0	0	0	0
B	3291	5	0	0	0	7	0	0	0	0	0	0
C	3087	46	31	4	1	89	62	21	6	0	0	0
D	8521	32	5	0	0	72	13	2	0	0	0	0
Total	16093	94	37	4	1	186	78	24	6	0	0	0

and finds the PRMVS pattern. The features searched by those three systems are very unique in the real environments so the possible marker candidates could be largely narrowed down and zero false detection rates could be achieved.

5.2.2 Inter-Marker Confusion

The inter-marker confusion happens if a 60 bits marker ID codeword has very a low hamming distance with other marker ID codewords so the erroneous codeword fools the marker detection algorithm. The inter-marker confusion rate could be investigated by computing the probability of confusing one marker ID codeword with another in the marker library.

Since MTag and CPRTag use digital techniques for the sampling of the inner marker data region and decodes the marker ID information for a marker recognition, the inter-marker confusion rate of MTag and CPRTag could be achieved relatively lower than the rate of ARToolKit which uses image correlation method for the inner data region matching. Unlike ARToolKit, for MTag and CPRTag, the investigation of the inter-marker confusion rate of MTag and CPRTag could be performed through statistical analysis based on a property of digital encoding / decoding techniques.

The probability that a 60 bits codeword X containing marker ID information confused with other markers in the library is computed by a summation of the probabilities that a given marker is falsely recognized as one of the marker codewords in the library and can be stated as $P(\neq X)$. For instance, the probability that a 60 bits codeword X is detected as another codeword Y, when the hamming distance between X and Y is 15, could be calculated as $p^{15}(1-p)^{45}$ where p is the probability that a bit is flipped.

In order to calculate the probability $P(\neq X)$, one needs to know the number of markers with same length of hamming distances when the codeword X is compared with all remained codewords in the marker library. The number of codewords with hamming distance n is noted as function $HD(n)$. In addition, we assume that the bit errors are uncorrelated and independent events so that the probability of n bits flipped among 60 bits is $p(n) = p^n$ and the probability of n bits transmitted correctly is $q(n) = (1\text{-}p)^n$. With this hamming distance function $HD(n)$ and the assumption, the final probability $P(\neq X)$ could be calculated as follow;

$$P(\neq A) = \sum_{n=1}^{60} HD(n) \cdot p^n (1-p)^{60-n} \tag{62}$$

Since the probability p which describes the probability that one bit is read incorrectly is system dependent, it could be varied with system environment. However, the hamming distances of the possible pairs of two different markers in the library could be manipulated by a system developer. In the designing phase of the fiduciary marker system, if one uses a different primitive polynomial of the Reed-Solomon code and a devisor polynomial of the CRC code, the hamming distances between two markers are changed.

The probability $P(\neq X)$ is dominated by first few terms of the right side of the eq. (55), that is, the marker pares of a smaller hamming distance has more effects on the probability $P(\neq X)$ since the probability p is much larger than the probability q most of the time ($p << q$). Fig. 33 shows a hamming distance histogram of an initially designed marker set of MTag and CPRTag. The first bin represents the marker pairs with hamming distance 16 and the number of pairs is 4095. In order to reduce the inter-marker confusion rate, the first part of the histogram is required either to be pushed out to the right or the size of it needs to be lowered.

The CRC code and RS code generation use code generator polynomials to calculate the checksums which are added to the 12 bits original data. The primitive polynomial for the RS code must be irreducible and has degree of 4 which is equal to the number of bits in each symbol. The polynomial used in CRC code is called CRC-8 polynomial and is also irreducible and has degree of 8. The 60 bits codeword of a given 12 bits marker ID codeword is totally changed if one uses different polynomials for the CRC / RS code generation procedure. Few polynomials which could be used for the CRC code and RS code checksum generations are shown in Table 6.

By using these polynomials, 8 possible combinations of two polynomials could be paired. The 4096 codewords corresponding 4096 marker IDs are produced with those 8 pairs and the hamming distances between 4096 codewords are generated for 8 cases. Fig. 34 illustrates the hamming distance histograms obtained from the 8 combinations of the polynomials. The histograms are shown in logarithmic scale and some parts of the histogram with zero values are omitted for the simplicity of the figure. The values of the histogram are numerically restated in Table 6 for a better understanding. The total number of possible marker pairs is 4096 × 4095 and since the MTag and the CPRTag have 4096

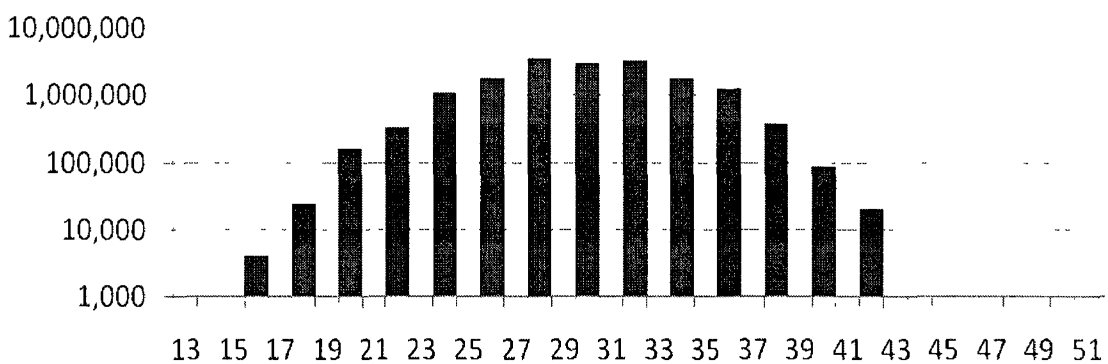


Figure 33. Hamming distance histogram obtained from the initially designed marker library. *The number of marker pairs with hamming distance 16 is 4094 and these pairs dominate the inter-marker confusion probability.*

markers, the 4096 codewords produced by the CRC / RS encoding technique with marker IDs from 0 to 4095 must hold hamming distances larger than zero with other marker codeword. As shown on the Fig. 33, the hamming distances of all possible marker pairs have even numbers due to a code generation characteristic of the RS code. Since the RS parity bits are added to the original data while making the number of bits holding '1' among 60 bits even, the hamming distance between any marker pair would be even.

Most of the hamming distances computed from the possible marker pairs are shown to be larger than 17 except the case of using RS#2 and CRC#4 polynomials (from now, the marker library generated with RS#2 and CRC#4 polynomials is called 'RS#2/CRC#4 library'). The number of marker pairs with hamming distance 16 in the RS#2/CRC#4 library have strongest effects on the probability of the inter-marker confusion. In addition, this library shows that the numbers of marker pairs of consequent hamming distances (hamming distance 17, 18, and 19) are also larger than few of other marker libraries. Thus, the RS#2/CRC#4 library is considered as a worst marker library among the 8 libraries when the inter-marker confusion rate is considered.

In this point of view, the RS#1/CRC#3 library could be regarded as the best library since the number of marker pairs with first three hamming distances are smaller than the others ($HD(18)$ = 12,282, $HD(20)$ = 114,632, and $HD(22)$ = 327, 520). Therefore, for the

TABLE VI
POLYNOMIALS USED FOR RS / CRC CHECKSUM GENERATIONS

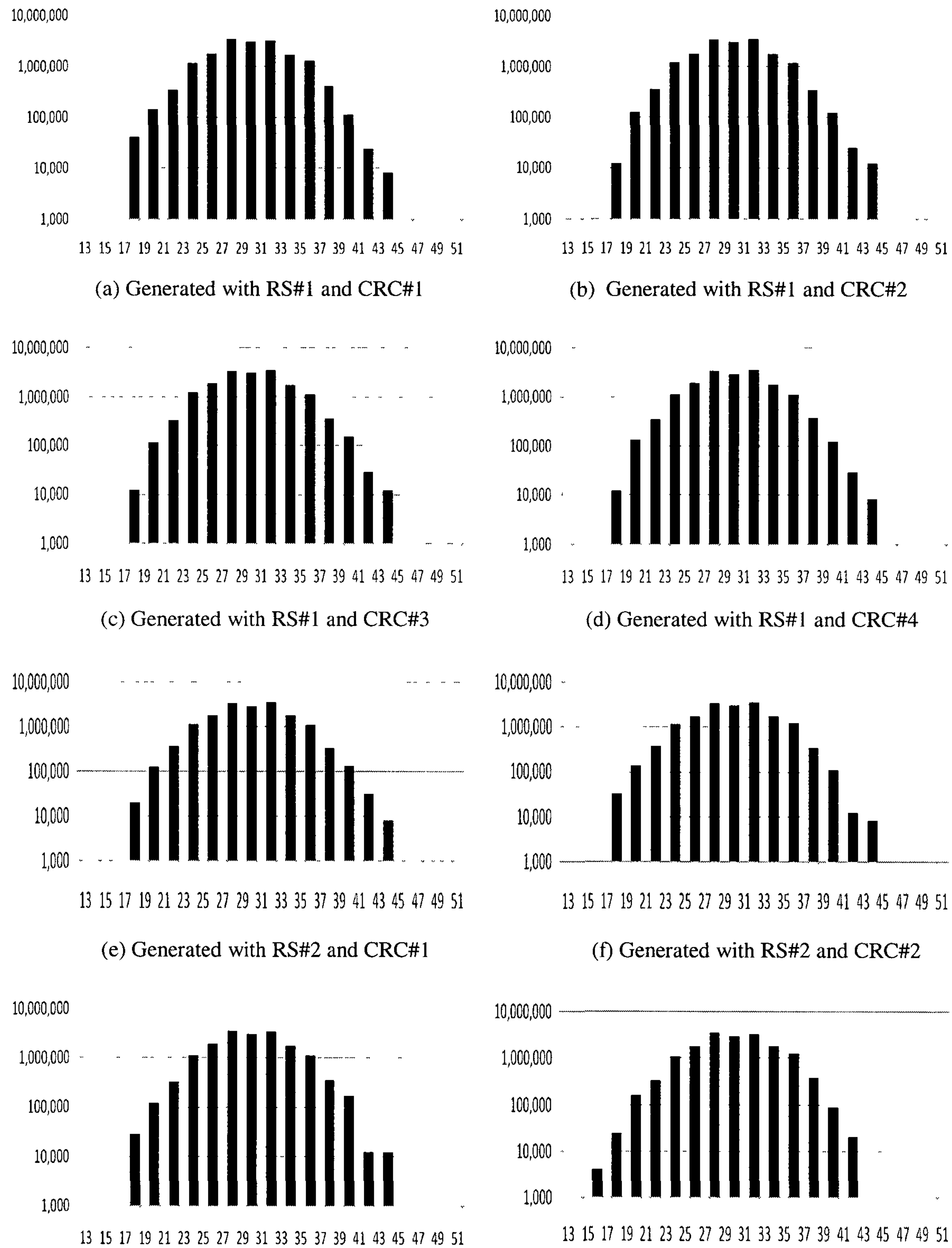
Types	Polynomials
RS #1	$x^4 + x + 1$
RS #2	$x^4 + x^3 + 1$
CRC #1	$x^8 + x^2 + x + 1$
CRC #2	$x^8 + x^3 + x^2 + x + 1$
CRC #3	$x^8 + x^4 + x^3 + x^2 + x + 1$
CRC #4	$x^8 + x^4 + x^3 + x + 1$

final marker library, the RS#1 and CRC#3 polynomials are chosen to generate the marker set to overcome the inter-marker confusion problem to some extent.

TABLE VII

NUMBER OF MARKER PAIRS WITH A GIVEN HAMMING DISTANCE OF 8 MARKER LIBRARIES

Hamming Distances	RS#1 CRC#1	RS#1 CRC#2	RS#1 CRC#3	RS#1 CRC#4	RS#2 CRC#1	RS#2 CRC#2	RS#2 CRC#3	RS#2 CRC#4
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	4,094
17	0	0	0	0	0	0	0	0
18	40,940	12,282	**12,282**	12,282	20,470	32,752	28,658	24,564
19	0	0	0	0	0	0	0	0
20	143,290	126,914	**114,632**	135,102	126,914	139,196	122,820	159,666
21	0	0	0	0	0	0	0	0
22	347,990	364,366	**327,520**	343,896	364,366	376,648	327,520	331,614
23	0	0	0	0	0	0	0	0
24	1,187,260	1,215,918	1,228,200	1,138,132	1,142,226	1,154,508	1,117,662	1,105,380
25	0	0	0	0	0	0	0	0
26	1,768,608	1,764,514	1,834,112	1,903,710	1,817,736	1,744,044	1,883,240	1,780,890
27	0	0	0	0	0	0	0	0
28	3,414,396	3,324,328	3,365,268	3,406,208	3,430,772	3,295,670	3,463,524	3,537,216
29	0	0	0	0	0	0	0	0
30	3,045,936	3,066,406	3,041,842	2,882,176	2,931,304	3,090,970	2,976,338	2,980,432
31	0	0	0	0	0	0	0	0
32	3,279,294	3,455,336	3,414,396	3,483,994	3,467,618	3,496,276	3,406,208	3,283,388
33	0	0	0	0	0	0	0	0
34	1,703,104	1,752,232	1,744,044	1,789,078	1,830,018	1,731,762	1,760,420	1,809,548
35	0	0	0	0	0	0	0	0
36	1,281,422	1,170,884	1,142,226	1,134,038	1,117,662	1,224,106	1,134,038	1,248,670
37	0	0	0	0	0	0	0	0
38	405,306	352,084	347,990	376,648	339,802	347,990	347,990	388,930
39	0	0	0	0	0	0	0	0
40	114,632	122,820	151,478	122,820	135,102	110,538	171,948	90,068
41	0	0	0	0	0	0	0	0
42	24,564	24,564	28,658	28,658	32,752	12,282	12,282	20,470
43	0	0	0	0	0	0	0	0
44	8,188	12,282	12,282	8,188	8,188	8,188	12,282	0
45	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0



(a) Generated with RS#1 and CRC#1

(b) Generated with RS#1 and CRC#2

(c) Generated with RS#1 and CRC#3

(d) Generated with RS#1 and CRC#4

(e) Generated with RS#2 and CRC#1

(f) Generated with RS#2 and CRC#2

(g) Generated with RS#2 and CRC#3

(h) Generated with RS#2 and CRC#4

Figure 34. Hamming distance histograms of 4 different marker libraries obtained with 4 pairs of RS and CRC polynomials shown in Table 6.

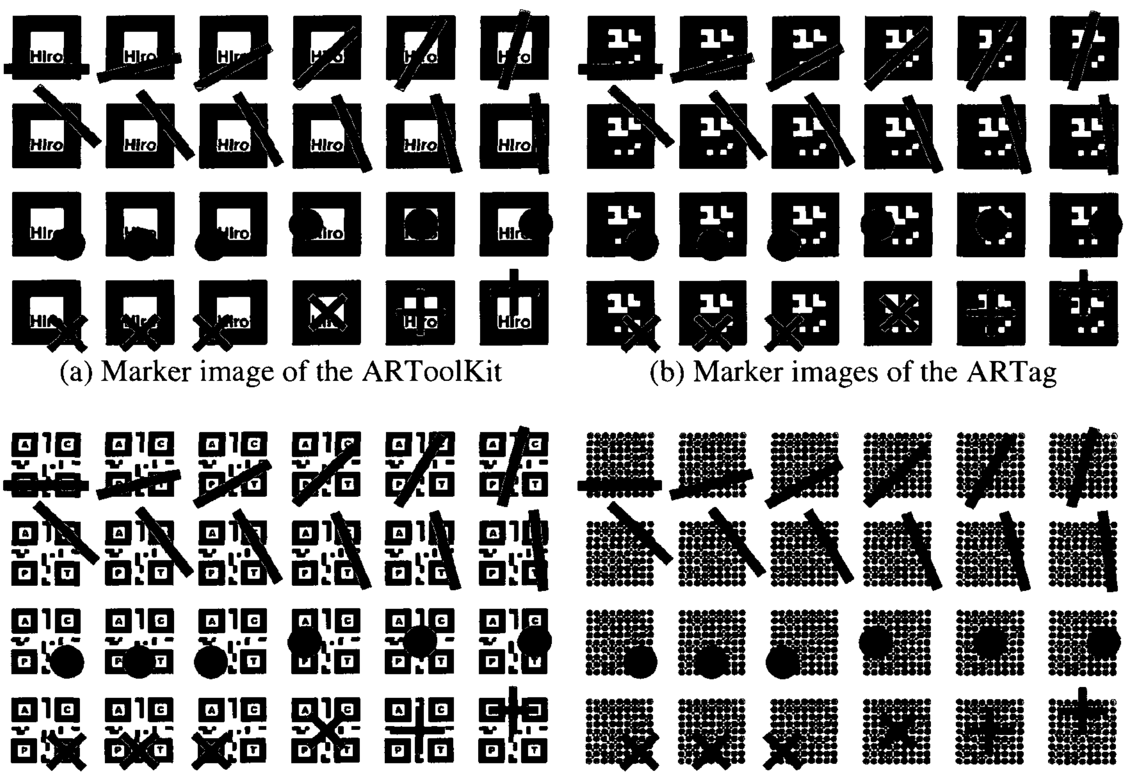
5.3. Marker Detection under Occlusion

The main advantage of two proposed marker systems is that these markers could overcome a marker detection failure under the partial occlusions to some extent due to new marker detection scheme with multiple sub-markers and the PRMVS pattern. Most of the currently available marker systems do not take account the occlusion problems in the marker designing phase so that the systems cannot provide a successful marker tracking performance even with very small occlusion. The inventor of the ARTag [24, 25], M. Fiala tried hard to overcome the occlusion problem by using the line detection and digital data encoding/decoding techniques. The ARTag marker system does not rely on binary thresholding method in finding a black quadrilateral border but it uses edge-based detection method to find a quadrilateral border. The binary thresholding method tends to fail in marker detection under various lighting conditions and partial occlusions whereas the edge-based detection method provides better results under those conditions. In addition, the internal pattern detection is performed with a digital approach instead of simple correlation method used in ARToolKit so the marker detection could easily overcome small portion occlusions.

However, even though the ARTag puts an effort to overcome the partial occlusion problem, the performance is still not satisfactory when we consider the environments of the indoor localization applications. Most of the fiduciary marker systems target the VR or AR applications. In addition, for those applications, the markers are used under user-controlable environment so that the severe occlusion problems do not occur frequently. Thus, ARTag which provide 5.5% data restoration ability could be enough for those VR or AR environments but not for the indoor localization environments.

Two proposed marker systems theoretically provide more reliable marker detection than ARTag and ARToolKit by using the multiple sub-markers and PRMVS pattern. In addition, they support 33.3% data restoration ability for the data loss or error read from the inner marker ID pattern. Therefore, they could be much more robust under partial occlusions. The following experiment compares the marker detection performances of the four marker system (ARToolKit, ARTag, MTag, and CPRTag) with partially occluded markers.

Fig. 35 illustrates the marker set used in this experiment. The four markers are occluded with three different objects in various position and angles and 24 images of the



(a) Marker image of the ARToolKit (b) Marker images of the ARTag

(c) Marker image of the MTag (d) Marker images of the CPRTag

Figure 35. Four marker sets of the 24 occluded marker images generated for the marker detection experiment. *The markers are occluded with three objects in different positions and angles.*

occluded marker are generated for each marker type. Then, those images are fed into the marker detection algorithm and the marker detection rates are checked.

Fig. 36 shows the marker detection results for each marker system. As it is shown, MTag and CPRTag provide a better marker detection performance than ARTag and ARToolKit. Interestingly, the marker detection rate of ARToolKit is differed by a confidence factor. If one sets the confidence factor as 0.9, the ARToolKit system cannot detect any of the occluded marker images but with confidence factor 0.5, the detection rate is increased up to 29%. However, this does not mean that the marker detection performance of the ARToolKit has been increased because if the confidence factor is low, the false positive rate is increased and consequently, any marker shaped objects can be regarded as a real marker. Furthermore, even if the ARToolKit detects the occluded marker images with lower a confidence factor, the contour of the detected marker is corrupted as shown in Fig. 37 so the pose estimation using four vertices of the marker contour would end up with larger error.

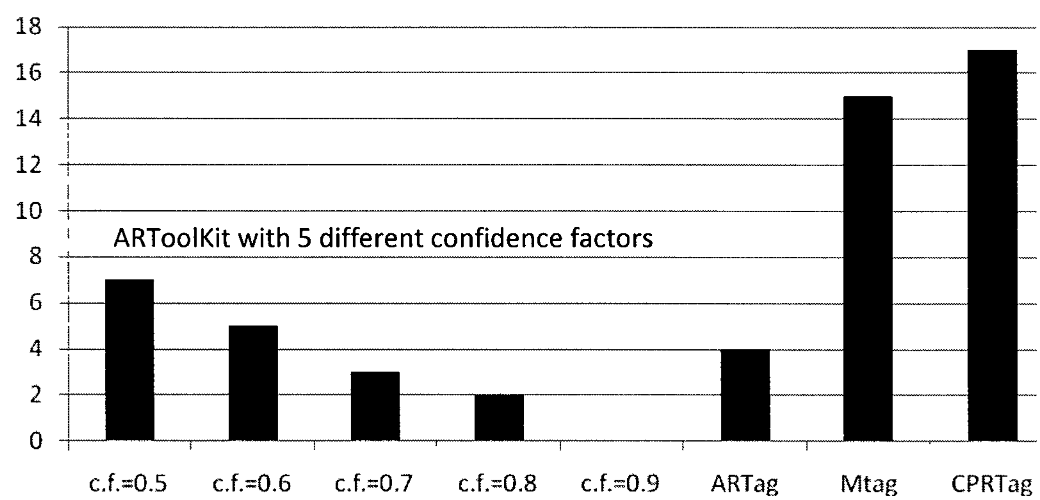


Figure 36. Marker detection test with 24 occluded marker images. *The CPRTag and MTag show best performance under partial occlusions of the marker.*

Figure 37. Incorrect marker detection results of the occluded ARToolKit markers. *The erroneous marker vertices tracking could result in camera pose estimation error.*

Another interesting result is that the ARTag shows a coarse marker detection rate (16%). Since the ARTag inventor does not provide the source code of it, it cannot be further examined but it is believed that the most of marker detection failures in this experiment might be caused by low data restoration ability of the digital techniques. In other words, although the marker system might be able to detect the marker templates in some occlusion cases but it might not be able to restore the marker ID from the occluded inner data.

On the other hands, MRag and CPRTag overcome this situation by adapting more reliable digital technique called Reed Solomon code and achieve better data restoration performance, and consequently result in better marker detection rates. This advantage of two proposed marker systems could allow them to be used in the uncontrolled environments such as hallways, corridors, or work places.

5.4. Pose Estimation Performance

The fiduciary markers are commonly used as a vision tracking method for the Augmented Reality or the Virtual reality systems. Since the fiduciary marker tracking system provides 6 pose of the marker with respect to the camera, an object could be superimposed on the captured image by using the geometrical information of the fiduciary marker.

The ARToolKit is a fiduciary marker system developed by the Human Interface Technology (HIT) Lab at the University of Washington. After the ARToolKit launched, many AR applications have used the system for the optical tracking. One of the problems of ARToolKit is that an instability problem can occur while tracking especially from partial occlusions. To solve this problem, the M. Fiala [24, 25] developed ARTag, and it could solve the instability problem to some extent due to its data restoration property. However, as stated in previous section, since MTag and CPRTag provide better data restoration performances than ARTag, the proposed fiduciary marker systems could be dedicated for the solution of the instability problem.

To compare the pose tracking performances of the four fiduciary marker systems; ARTag, ARToolKit, MTag, and CPRTag, a distance measurement accuracy test was done with four markers sampled from the each of the four marker libraries. The experimental environment is described in Fig. 32. A marker among four different markers is located in the center of the work place, and a camera attached to a tripod is placed away from the marker within a specific distance and an angle. 2,000 image sequences are captured with 9 distance variations (60cm, 90cm, 120cm, 150cm, 180cm, 210cm, 240cm, 270cm, and 300cm) and 6 angle variations (90 deg. 80 deg., 70 deg., 60 deg., 50 deg., and 40 deg.). The angle of the camera position is changed only around the y-axis. Then, the marker in the center of the work place is replaced with other marker. Thus, the total number of the image sequences is 9 distances $\times$ 6 angles $\times$ 4 markers $\times$ 2,000 images = 432,000 image sequences.

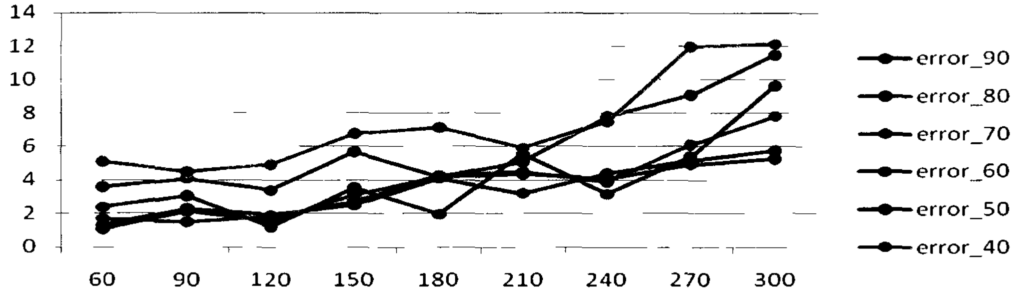
The camera pose is calculated from each image and stored in the data storage.

Although errors along with each of *x, y,* and *z* axis of one marker system are required to be compared with errors of other system, the distance errors are compared between markers for the simplicity of the experiment. Since there is no ground truth of the camera parameters and we use estimated camera parameters instead of the ground truth, it is hard to tell the experimental set-up is noise free. Thus, the evaluator needs to be aware of this property of the vision related experiments.

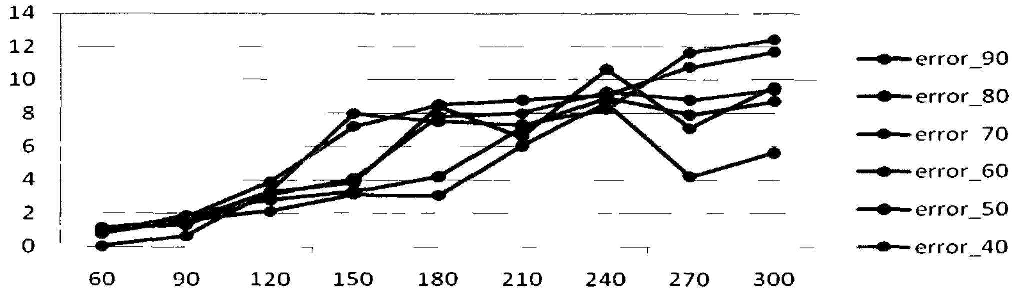
Fig. 38 shows the system errors of the four marker systems with 9 distance variations and 6 angle variations. The errors tends to get larger in accordance with the increase of the distance between the marker and the camera since the image coordinates used for the homography calculation become inaccurate due to the quantization step of the digital imaging (low resolution). In addition, as the angle between the marker and the camera decreased, the error of the distance measurement is increased since the shape of the marker is distorted and consequently, the image coordinates of the feature points are prone to more noise.

Fig. 39 illustrates the distance errors of four marker systems measured from distance 60cm to 300cm with fixed angles (90 deg. 80 deg., 70 deg., 60 deg., 50 deg., and 40 deg.). When we compare the distance errors between the marker systems, the errors are smaller in the cases of MTag and CPRTag than those of ARTag and ARToolKit. Since MTag and CPRTag have higher number in maximum feature points, 16 and 49 respectively, than ARTag and ARToolKit, which have only 4, the result from the pose calculation algorithm could be more accurate by using Least Square Error method.

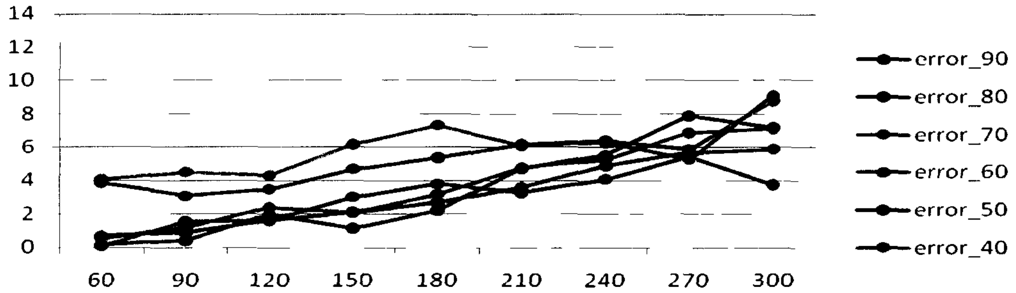
Fig. 40 shows distance error standard deviations of four marker systems. Based on the result, it is noted that the distance measurement performance of the proposed fiduciary markers are more precise than the others.



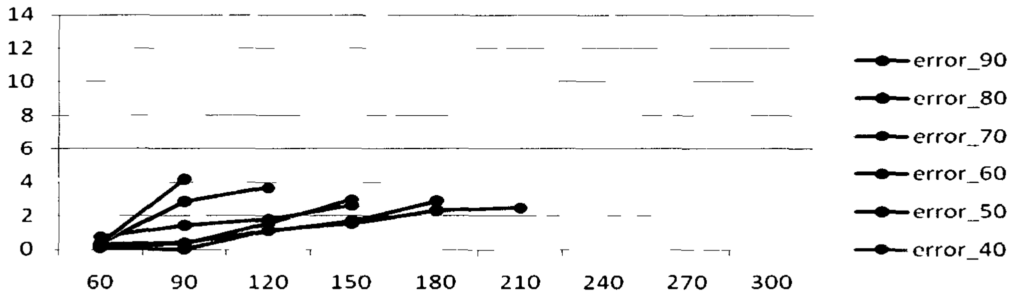
(a) Distance errors(cm) of the ARTag



(b) Distance errors(cm) of the ARToolKit

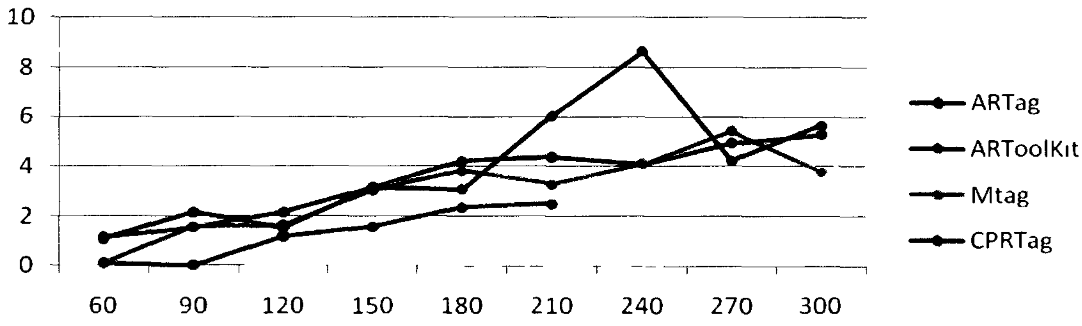


(c) Distance errors(cm) of the MTag

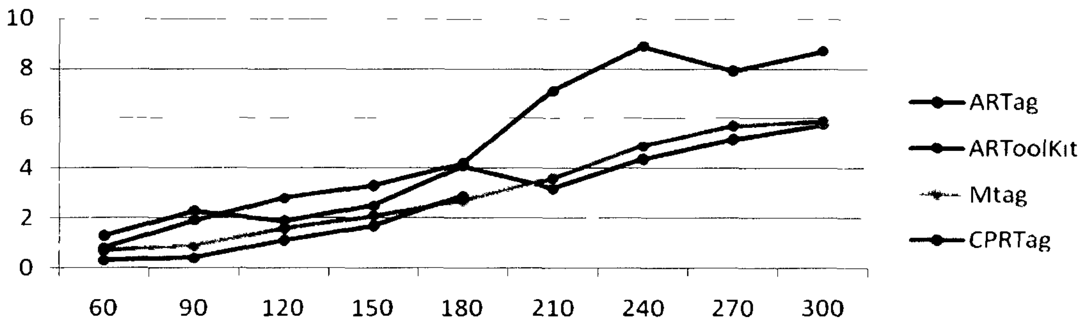


(d) Distance errors(cm) of the CPRTag

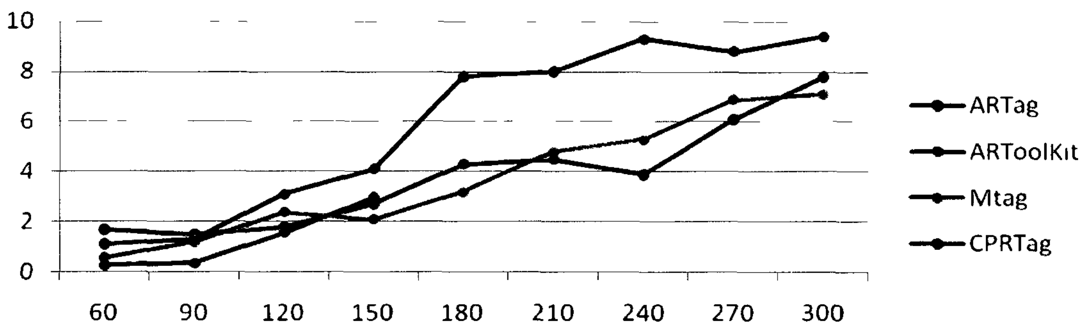
Figure 38. System errors of four marker systems obtained from the distance measurement accuracy test with 9 distance variations and 6 angle variations



(a) Distance errors(cm) at 90 deg.

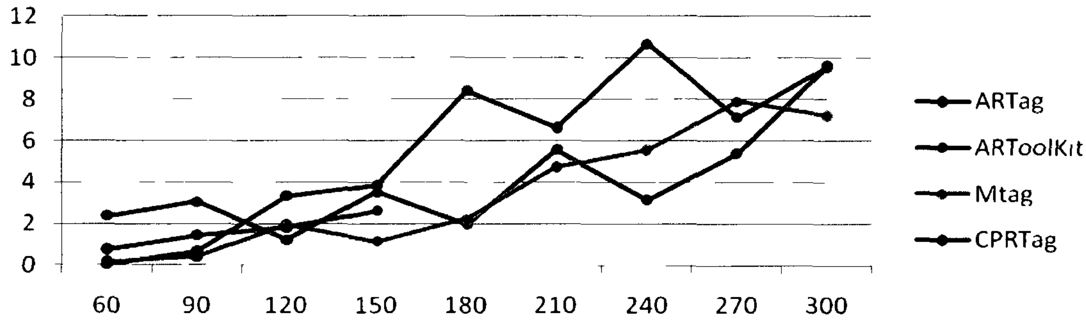


(b) Distance errors(cm) at 80 deg.

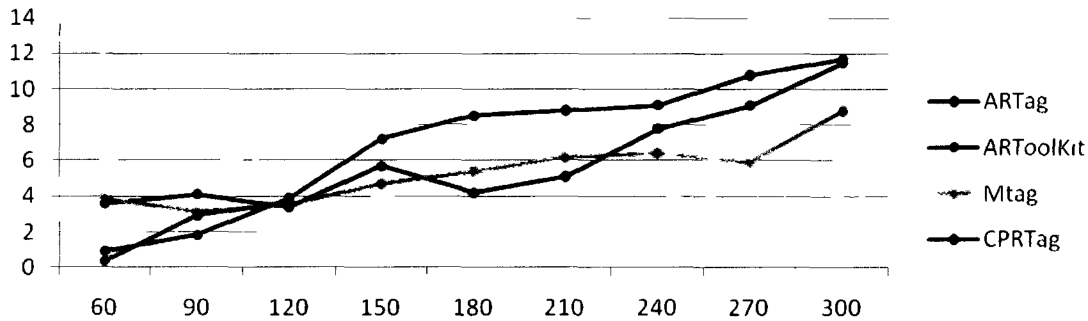


(c) Distance errors(cm) at 70 deg.

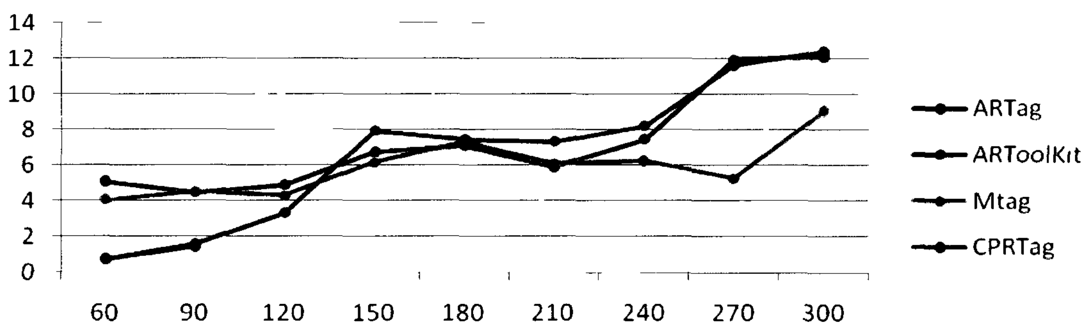
Figure 39-1. System error comparison on a specific angle between four marker systems



(a) Distance errors(cm) at 60 deg.

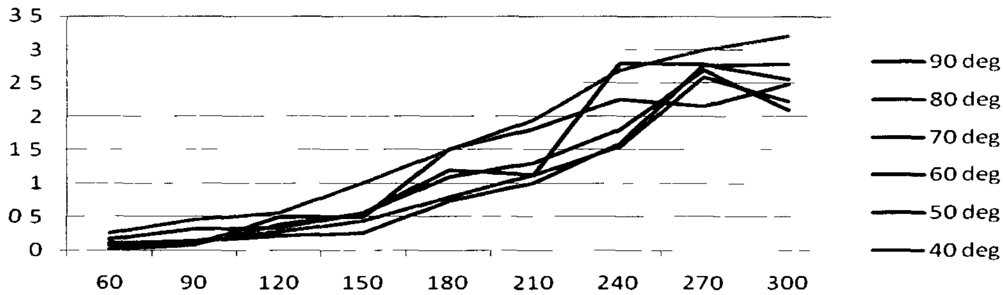


(b) Distance errors(cm) at 50 deg.

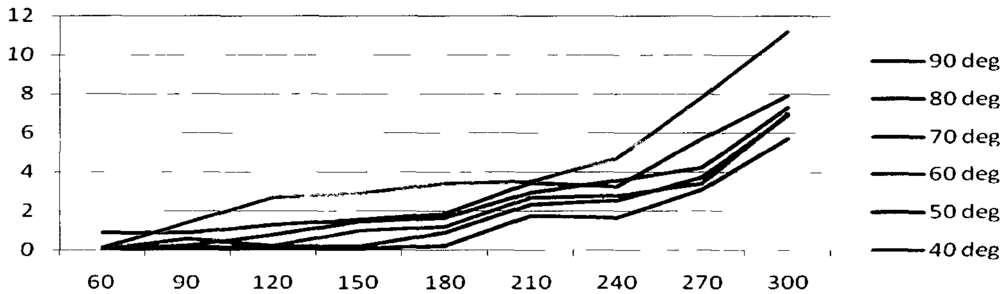


(c) Distance errors(cm) at 40 deg.

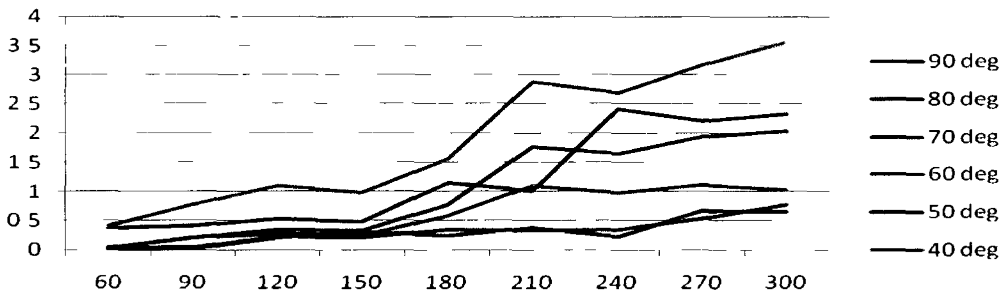
Figure 39-2. System error comparison on a specific angle between four marker systems



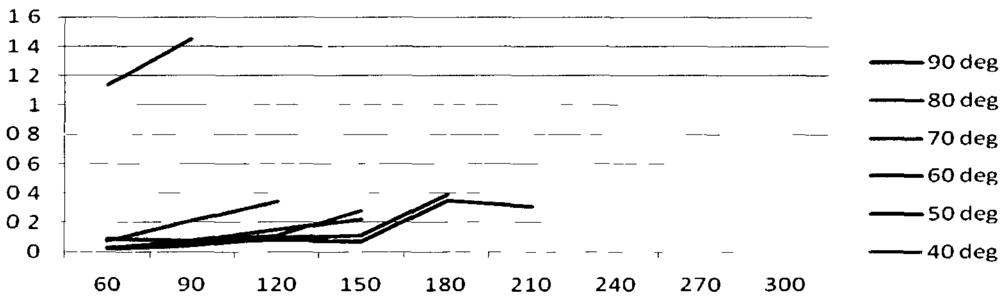
(a) Distance errors STD (cm) of the ARTag



(b) Distance errors STD (cm) of the ARToolKit



(c) Distance errors STD (cm) of the MTag



(d) Distance errors STD (cm) of the CPRTag

Figure 40. Distance measurement STD comparison of four fiduciary marker systems

The CPRTag provides most accurate pose estimation results among four marker systems, however, it shows low marker recognition rates as distance between the marker and the camera increases and as angle decreases. Even, the marker could not be detected over 90 cm at 40 deg. Since the marker detection algorithm is based on the color segmentation and detection, and uses simple thresholding method and fixed number of morphological operations, the marker detection algorithm is lack of adaptability and fails frequently with increased distance and decreased angle. The CPRTag has the most reliable property overcoming partial occlusions as shown in previous section but has lowest performance with respect to the marker detection rate due to lack of flexibility. Thus, the marker detection algorithm of the CPRTag is required to be improved for the indoor localization usage.

6. Conclusions

6.1. Summary of the thesis

In this thesis, two fiduciary markers which could be used for the indoor localization are proposed. The two proposed fiduciary markers are designed while considering the messy indoor environments to overcome some constraints exist in indoor localization applications. For recent years, various fiduciary marker systems have been introduced as AR field gets more attention from the researchers since the fiduciary marker system is an easy, affordable, and light solution for tracking applications. These advantages of the fiduciary marker in tracking method allows one to use them for the indoor localization instead of using an expensive, heavy, and complex beacon system such as RFID, Wi-Fi, or Pseudo-GPS.

Most of currently available fiduciary markers are designed for the AR or VR applications so the author put an effort to implement a fiduciary marker system to be adequate for the indoor localization usage. The most important features of the fiduciary marker required for the indoor localization are a robustness against partial occlusions and a better accuracy of the pose tracking.

To achieve those features, the author reviewed the currently available marker systems such as ARToolKit and ARTag and studied the advantages and disadvantages of them to compromise them and to accomplish the indoor localization-oriented fiduciary marker systems.

The ARToolKit is the most famous fiduciary marker system but its marker detection scheme based on inner pattern correlation method provides coarse results, that is relatively higher false marker detection rate. In addition, the marker pose estimation result is inaccurate. In the case of ARTag, the marker detection performance is better than that of ARToolKit but shows unsatisfactory results under partial occlusion situations.

In purpose to overcome those weak points of two fiduciary marker systems and to achieve desired performance, the author applied 'Artificial Neural Network' scheme for the pattern recognition, digital data transmission techniques called 'Reed-Solomon code' and 'Cyclic Redundancy Check' and 'Pseudo-Random Multi-Valued Sequences' to two proposed marker systems named 'MTag' and 'CPRTag'.

The successful marker detection depends on how effectively the system finds marker candidates and how accurately the system retrieves marker ID information from the marker candidates. In the case of the ARToolKit, the system finds marker candidates well but the marker ID information extraction is not reliable and fails frequently. In the case of the ARTag, the marker ID information decoding procedure performs well and finds markers accurately. However, both fiduciary marker systems show low marker detection rates under partial occlusion and this weakness should be improved for the indoor localization.

The pose estimation accuracy is affected by how accurately the system finds the known feature points of the fiduciary marker in a captured image since the feature points from image coordinates are used for the marker pose tracking. Since the calculation is performed with Least Square Error (LSE) method, more the data, more accurate pose estimation the system achieves. ARToolKit and ARTag provide four feature points which are four corner points of the marker template, and the number of feature points satisfies the minimum number of feature points (= 4) required for the marker pose calculation. For better accuracy of the pose estimation result, the proposed ones need to be designed to provide more feature points.

6.2. Thesis contributions

MTag, one of the proposed fiduciary marker systems, is originally inspired by the ARToolKit. Since the ARToolKit shows bad performance with respect to the marker ID recognition although it is very simple and light system, the MTag arranges four ARToolKit marker-shaped sub-markers and uses it for detecting marker candidates. For the marker ID information extraction, the MTag utilizes two digital techniques, Reed-Solomon code and Cyclic-Redundancy-Check so it achieves low false marker detection rate and high marker detection rate under partial occlusions. In designing phase, two polynomials used for the two digital techniques are chosen carefully to reduce the chance of false marker detection. By using RS-code and CRC, the marker could provide 33.3% data restoration ability and this might be regarded as the most significant contribution on the designing new fiduciary markers.

CPRTag, the other proposed fiduciary marker system, adopts the Pseudo-Random Multi-Valued Sequences for the marker template instead of simply using the black quadrilateral template used in most of the currently available fiduciary marker systems. The disadvantage of using a black quadrilateral marker template is that a marker cannot be detected when the template is slightly occluded or hidden. However, with the PR pattern engraved into the CPRTag, a marker could be detected even under occlusions over 3/4 of the marker area. In addition, CPRTag has 49 feature points with known geometrical information so the pose estimation could be more accurate than any other fiduciary marker systems.

6.3. Suggestions for Future Research

The two proposed fiduciary marker systems have few weak points although they achieved desired properties required for the indoor localization.

MTag and CPRTag show a lower marker detection rate as distance between the marker and the camera is increased. This is because sub-marker detection algorithm of the MTag could not recognize the characters in the four sub-markers correctly as distance increases. Moreover, since the PR pattern detection algorithm of the CPRTag analyzes the color information of the captured scene and finds the color PR pattern of the marker template, if the size of the captured image of the CPRTag is decreased, the marker system could not properly label the color PR pattern. Besides, the proposed markers occupy more area than the other fiduciary markers to achieve the same marker detection performance in general. Thus, for better marker detection rate under long distance range and with smaller sized markers, the sub-marker detection and PR pattern detection algorithm should be further improved for the flexibility with distance changes.

In order to achieve these strengths, new features, which are more unique, noticeable, and computationally inexpensive, could be suggested for the MTag or CPRTag instead of alphabet characters or the color PR patterns. In [61], two simple 2D objects composed of bars and circles are used for representing the Pseudo-Random Binary Array. Since those two objects are uniquely defined with the number of ends and number of joints, they could be easily differentiated. In addition, frequency distributions of the user-defined objects could be used for describing a PR pattern as well as the geometrical information of the objects. With these additional properties of new features, a fiduciary marker would be more unique and outstanding. Consequently, the marker detection computation time could be decreased, and the maximum marker detection range could be increased as well.

The proposed fiduciary marker systems are designed for indoor localization usage to provide easy, affordable, and light but reliable method. Most of the indoor localization or navigation systems rely on beacon systems using ultra-sonic, infrared, or radio sensors. Since these systems are costly for installation, maintenance, and operation, the proposed fiduciary marker systems are worth being used for the indoor localization.

One of possible applications using proposed fiduciary markers for indoor localization is a path-finder with a smart-phone. Recently, due to the advent of processing power and default camera installation of a smart-phone, various types of vision-based applications running on a smart-phone have been released to the public. If a smart-phone user, especially a blind, captures an indoor scene with a wall-attached fiduciary marker via on-board camera, the system could retrieve the current position with respect to the marker position. Then, the system looks for the marker position with respect to the building coordinate through marker position data-base. Finally, a user could be informed his/her current position.

Furthermore, the fiduciary marker system could form into a continuous indoor localization system by combining it with an inertial measurement unit or an odometery. This combination of two different sensors and its process are called 'Sensor Fusion [77][78][79]'. The combined system would be a possible solution for the indoor localization and thus, an examination of this navigation system would be necessary for future studies.

References

[1] M.S, Grewal, L.R. Weill, and A.P. Andrew, *Global positioning Systems, Inertial navigation and integration*, New Jersey: John Wiley & Sons,, 2nd ed., 2007.

[2] O. Wijk, H.I. Christensen, "Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data," *Robotics and Autonomous Systems*, vol.31, issues 1-2, pp.31-42, 2000.

[3] F. Raab, E. Blood, T. Steiner, and R. Jones, "Magnetic position and orientation tracking system," *IEEE Trans. Aerospace and Electronics Systems*, vol.AES-15, No.5, pp.709-718, 1979.

[4] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata. "Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/RFID/Self-contained Sensor System". in *Proc.16th International Conference on Automotive Technology(ICAT`06)*, pp.1310-1321, 2006.

[5] D. Sticker and T. Kettenbach, "Real-time and Markerless Vision-Based Tracking for Outdoor Augmented Reality Applications," in *Proc. 4th International Symposium on Augmented Reality (ISAR `01)*, pp. 189-190, 2001.

[6] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3d camera tracking," in *Proc. 3rd* International Symposium on Mixed and Augmented Reality (*ISMAR`04)*, pp. 48-57, 2004.

[7] A. J. Davison, "Real-Time Simultaneous Localization and Mapping with a Single Camera," in *Proc. ICCV'03*, Vol. 2, pp. 1403-1410, 2003

[8] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," in *Proc. 2nd IEEE Workshop on*

Augmented Reality(IWAR '99), pp. 85-94, 1999.

[9] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann, and D. Schmalstieg, "Structured Visual Markers for Indoor Pathfinding," in *Proc. ART02*, 2002.

[10] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Proc. International Symposium on Mixed and Augmented Reality (ISMAR'02)*, pp. 27-36, 2002.

[11] D.G. Lowe, "Object Recognition from Local Sclae-Invariant Features," in *Proc. of the International Conference on Computer Vision*, vol.2, pp. 1150-1157, 1999

[12] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, issue 2, pp. 92-110, 2004.

[13] S.Y. Park, S.C. Jung, Y.S. Song, and H.J. Kim, "Mobile robot localization in indoor environment using Scale-Invariant visual landmarks," in *Proc. the 18th IAPR International Conference on Pattern Recognition (ICPR'06)*, pp. 159-163, 2008.

[14] E.M. Petriu and J.S. Basran, "On the position measurement of automated guided vehicles using pseudorandom encoding," *IEEE Trans. Instrumentation and Measurement* , vol. 38, No. 3, pp. 799-803, 1989.

[15] E. M. Petriu, "Absolute-type position transducers using a pseudorandom encoding," *IEEE Trans. Instrumentation Measurements.,* vol. 1M-36, pp. 950-955, Dec. 1987

[16] E. M. Petriu, "Absolute position recovery for automated path-guided vehicles," *presented at ROBOTS 12 and VISION '88 Conf. Exibit.*, Detroit, MI, June 5-9, 1988.

[17] E.M. Petriu, W.S. McMath, S.K. Yeung, N. Trif, and T. Bieseman, "Two-dimensional position recovery for a free-ranging automatedguided vehicle," *IEEE Trans. Instrumentation and Measurement*, vol 42, issue 3, pp. 701-706, 1993.

[18] A. Kazantsev and E.M. Petriu, "Robust Pseudo-Random Coded Colored Structured Light Technique for 3D object Model Recovery", *IEEE Int'l Workshop on Robotic and Sensors Environments*, pp. 150-155, 2008

[19] P. Payeur and D. Desjardins, "Structured Light Stereoscopic Imaging with Dynamic Pseudo-Random Patterns," in *Proc. of the International Conference on Image Analysis and Recognition(ICIAR '09)*, pp. 687-696, 2009

[20] M. Fiala, "ARTag Revision 1. A Fiduciary Marker System Using Digital Techniques," *published as NRC/EPR-1117*, 2004

[21] A. Mullpni, D. Wagner, D. Schmalstieg, and I. Barakonyi, "Indoor positioning and navigation with camera phones," *IEEE Pervasive Computing*, vol 8. issue 2. pp. 22-31, 2009.

[22] S.S. Chawathe, "Marker-Based Localizing for Indoor Navigation," in *Proc. of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 885-890, 2007.

[23] T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, and S. Lieberknecht, "An augmented reality museum guide," in *Proc. of the International Symposium on Mixed and Augmented Reality*, pp. 103-106, 2008.

[24] M. Fiala, "Designing Highly Reliable Fiducial Markers," *IEEE Trans. Of Pattern Analysis and Machine Intelligence*, issue 99, pp. 1-8, 2009.

[25] M. Fiala, "ARTag, a fiducial marker system using digital techniques," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* vol. 2, pp. 590-596, 2005.

[26] M. Fiala, "Vision Guided Control of Multiple Robots," in *Proc. 1st Canadian*

Conference on Computer and Robot Vision,(CCCR'04), pp 241-246, 2004.

[27] C. B. Owen, F. Xiao, and P. Middlin, "What is the best fiducial?", *The 1st IEEE International Augmented Reality Toolkit Workshop,* pp. 1-8, 2002.

[28] M. Hirzer, "Marker Detection for Augmented Reality Application," 2008, Available at : http://studierstube.icg.tu-graz.ac.at.

[29] X. Zhang, S. Fronz, and N. Navab, "Visual Marker Detection and Decoding in AR System : A Comparative Study," in *Proc. of the International Symposium on Mixed and Augmented Reality, ISMAR `02*, pp. 97 - 106 , 2002.

[30] P. Malbezin, W. Piekarski, and B. H. Thomas, "Measuring ARToolkit Accuracy in Long Distance Tracking Experiments," in *ART`02, 1st, International Augmented Reality Toolkit Worshop*, Darmstadt, Germany, 2002.

[31] D. F. Abawi, J. Bienwald, and R. Dorner, "Accuracy in Optical Tracking with Fiducial Markers: An Accuracy Function for ARToolKit," in *Proc. of the Third IEEE and ACM International Symposium on Mixed Augmented Reality, ISMAR `04*, 2004

[32] S. Cawood and M. Fiala, *Augmented Reality : A Practical Guide*, Pragmatic Bookshelf, USA, 2007.

[33] T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura, "AR^{2}Hockey : A Case Study of Collaborative Augmented Reality," in *Proc. Of IEEE Virtual Reality Annual International Syposium, VRAIS `98*, pp. 268-275, USA, 1998

[34] S. Feiner, B. MacIntyre, and D. Seligmann, "Knowledge-based augmented reality," *Comm. ACM*, vol.36, No.7, pp.52-62, 1993.

[35] M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging virtual reality with the real world: Seeing ultrasound imagery within the patient," *Computer Graphics*, vol. 26, no.2,

pp. 203-210, 1992.

[36] E. Rose et al., "Annotating real world objects using augmented reality," in *Proc. of Computer Graphics International '95*, pp.357-370, 1995.

[37] S. Uchiyama, K. Takemoto, H. Yamamoto, and H. Tamura, "MR platform: A basic body on which mixed reality applications are built," in *Proc. of the International Symposium on Mixed Augmented Reality(ISMAR`02)*, pp. 246–253, 2002.

[38] Y. Uematsu and H. Saito, "AR Registration by Merging Multi Planar Markers at Arbitrary Positions and Poses via Projective Space," in *Proc. of the 2005 international conference on Augmented Tele-existence,* pp. 48-55. 2005

[39] R. Miguel, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz, "Hybrid tracking for outdoor augmented reality applications," *IEEE Computer Graphics and Applications*, pp. 54–63, 2002.

[40] U. Neumann and S. You, "Natural feature tracking for augmented reality," *IEEE Trans. on Multimadia*, vol. 1, pp 53–64, 1999.

[41] K. W. Chia, A. Cheok, and S. J. D. Prince, "Online 6 DOF augmented reality registration from natural features," in *Proc. of the International Symposium on Mixed Augmented Reality(ISMAR`02),* pp. 305–313, 2002.

[42] D. Schmalsteig and D. Wagner, "Experiences with Handheld Augmented Reality," *in Proc. of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1-13, 2007

[43] G. Mediono, S. B. Kang, *Emerging Engineering Topics in Computer vision*, USA : Prentice Hall, pp. 4-40, 2004.

[44] H. M. Karara, Ed., *Handbook of Non-Topographic Photogrammetry,* American Society for Photogrammetry, 2 ed., 1979.

[45] J. Heikkila and O. Silven, “A Four-Step Camera Calibration Procedure with Implicit Image Correction,” in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’97)*, pp. 1106–1112, 1997.

[46] Z. Zhang, “Flexible Camera Calibration By Viewing a Plane From Unknown Orientations,” in *Proc. of the IEEE International Conference on Computer Vision (ICCV'99)*, pp.666-673, 1999.

[47] Z. Zhang, “A Flexible New Technique for Camera Calibration”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.22 no.11, pp.1330-1334, 2000.

[48] D. C. Brown, “Close-Range Camera Calibration,” Photogrammetric Engineering 37, pp. 855-866, 1971.

[49] P. F. Sturm and S. J. Maybank, “On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications,” in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99),* vol. 1, pp. pp.1432 – 1440, 1999.

[50] D. Chen and G. Zhang, “A New Sub-Pixel Detector for X-Corners in Camera Calibration Targets,” in *Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (WSCG’05), pp. 97-100, 2005.

[51] H.N. Robert, *Neural networks for perception: computation, learning, architectures*, Orlando: Harcourt Brace & Co., 1992, pp.65-93.

[52] I. A. Basheer, M. Hajmeer, “Artificial neural networks: fundamentals, computing, design, and application,” *Journal of Microbiological Methods*, vol. 43, pp. 3-31. 2000.

[53] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical System Approach To*

Machine Intelligence, EngleWood Cliffs, NJ : Prentice Hall, 1991

[54] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *IEEE Computer*, vol. 29 no. 3 pp. 31-44, 1996.

[55] W. E. Snyder and H. Qi, *Machine Vision*, Cambridge, UK: Cambridge University Press, 2004.

[56] I. S. Oh, C. Y. Suen, "Distance features for neural network-based recognition of handwritten characters," *International Journal on Document Analysis and Recognition*, vol. 1, no. 2, pp.73-88, 1998.

[57] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst, "Shape Recognition and Pose Estimation for Mobile Augmented Reality", in *Proc. of the International Symposium on Mixed Augmented Reality*(ISMAR '09), pp. 65-71, 2009.

[58] C.A. Rothwell, A. Zisserman, D.A. Forsyth, and J.L. Mundy, "Canonical Frames for Planar Object Recognition", in *Proc. of the Second European Conference on Computer Vision*, pp. 757-772, 1992.

[59] A. Zisserman, D.A. Forsyth, J.L. Mundy, C. A. Rothwell, J. Liu, and N. Pillow, "3D object ecognition using invariance". *Artificial Intelligence*, pp. 239-288, 1995.

[60] F. J. McWilliams and N.J.A. Sloane, "Pseudo-Random Sequences and Arrays," in *Proc. of IEEE*, vol. 64, pp.1715-1729, 1976.

[61] E. M. Petriu, T. Bieseman, N. Trif, W. S. McMath, and S. K. Yeung, "Visual Object Recognition Using Pseudo-Random Grid Encoding," in *Proc. of the International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1617 – 1624, 1992.

[62] H. J. W. Spoelder, F.M. Vos, and E. M. Petriu, "A Study of the Robustness of Pseudorandom Binary-Array-Based Surface Characterization," *IEEE Trans. on*

Instrumentation and Measurement, vol. 47, no. 4, 1998.

[63] I. Kim, S. Kim, S. Yu, and S. Lee, "Robust 3D Face Data Acquisition Using a Sequential Color-Coded Pattern and Stereo Camera System," in *Proc. of the International Conference on Computational Science and its Applications*, vol. 3984, pp. 87-95, 2006.

[64] J. Salvi, J. Pages, and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern recognition*, vol. 37, no. 4, pp. 827-849. 2004.

[65] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, and S. Zietz, "Structured Light Using Pseudorandom codes," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, 1998.

[66] S. B. Wicker and V. K. Bhargava, *Reed Solomon Code and their applications*, Piscataway, NJ: IEEE Press, 1994

[67] T.G. Soon, "QR code," *The Synthesis Journal*, pp. 59-78. 2008.

[68] P. W. Frey and D. J. Slate, "Letter Recognition Using Holland-Style Adaptive Classifiers," *Machine Learning*, vol. 6, no. 2, pp 161-182, 1991.

[69] G. Daqi, L. Chengyin, and L. Changwu, "Modular Adaptive RBF-Type Neural Networks for Letter Recognition," in *Proc. of the International Joint Conference on Neural Network* , vol. 1, pp. 571 – 576, 2003

[70] G. Daqi, X. Chao, and N. Guiping, "Combinative Neural-Network-Based Classifiers for Optical Handwritten Character and Letter Recognition," in *Proc. of the International Joint Conference on Neural Networks*, vol. 3, pp.2232-2237, 2003.

[71] P. Picton, *Neural Networks,* New York, NY: Palgrave , pp. 37-47, 2000.

[72] R. P. Lippmann, "An introduction to computing with neural nets," *ACM SIGARCH Computer Architecture New*, vol. 16, issue 1, pp. 7-25, 1988.

[73] A. Lapedes and R. Farber, "How neural nets work," in *Proc. of the Neural information processing systems*, pp. 442-456, 1988.

[74] H. Zhang, F. Yang, Y. Wu, and M. Paindavoine, "Robust color circle-marker detection algorithm based on color information and Hough transformation," *Optical Engineering,* vol 48, issue 10, pp. 107202-1-07202-10 2009.

[75] G. D. Finlayson, S. Hordley, G. Schaefer, and G. Y. Tian, "Illuminant and device invariant color using hidtogram equalization," *Pattern Recognition*, vol. 38, pp. 179-190, 1998

[76] L. P. MacNamee, E. M. Petriu, and H. J. W. Spoelder, "Photogrammetric Calibration of a Mobile Robot Model," IEEE Instrumentation and Measurement Technology Conference, pp. 245-250.

[77] W. W. Peterson and D. T. Brown, "Cyclic Codes for Error Detection," in Proc. of the IRE, vol. 49, issue 1, pp. 228-235, 1961.

[78] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots", *IEEE Transactions on Robotics and Automation,* vol. 15, no. 2, pp. 219-229, 1999.

[79] Q.H. Meng, Y.C. Sun, and Z.L. Cao, "Adaptive estended Kalman filter based Mobile robot Localization", *Cambridge University Press, Robotica*, vol. 18, pp. 459-473, 2000.

[80] E. Ivanjko, I. Petrovic, and M. Vasak, "Sonar-based pose Tracking of Indoor Mobile Robots", *Journal for control, measurement, electronics, computing and communications*, vol. 45, no. 3-4, pp.145-154, 2004.