Evaluation of Visible and Invisible Fiducial Markers for Clothing Tracking

Siyu E

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-153
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-153.html

June 1, 2012

Acknowledgement

This research project would not have been successful without the support of many people. We want to express our gratitude to the supervisor, Prof. Pieter Abbeel, who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude are also due to members of the Berkeley Robotics Lab, John Schulman without whose knowledge and assistance this study would not have been successful. Special thanks also to Prof. Jitendra Malik, Prof. Trevor Darrell for sharing the literature and invaluable assistance. Not forgetting to our friends who always been there. Our team wish to express our love and gratitude to beloved families and friends for their understanding and endless love, through the duration of this project.

CAPSTONE PROJECT PAPER

M.ENG SPRING 2012

UNIVERSITY OF CALIFORNIA, BERKELEY

Title: **Evaluation of Visible and Invisible Fiducial Markers for Clothing Tracking**

Author: **Siyu E**

Advisor: **Pieter Abbeel**

1 Abstract

In this paper, we summarize and evaluate the techniques for detecting and identifying the detection patterns in the context of clothing simulation for PR2 (and any robot using Robot Operating System as its OS). We illustrate in this context, why the marker processing algorithm introduced in ARToolkit - Based Tracking[7] is the best performer for minimal detectable pattern size and under warping configurations. Additionally, for our shirts to be similar enough with typical shirts, we also present a solution for PR2 to simulate the behavior of a unique marked-up clothing with invisible detection patterns. This article will describe in detail how we improve the algorithm to detect and identify UV ink markers using the Kinect system.

2 Introduction

In the near future, we want the PR2 (Personal Robot 2) to be able to replace humans in folding clothes. To realize this, the PR2 needs to be more reliable and efficient through a better Physics Based Simulation Engine - Bullet Physics, an open source simulation engine that is readily accessible in C++ - that can simulate clothing and its reaction when manipulated, since Current Algorithm has the unrealistic simulation problem of low success rate and slow process speed. In general, only twenty out of thirty attempts on clothing will succeed in folding with the current algorithm, and the PR2 is always repeating actions of grasping the clothing until it sees a desired configuration, which costs a long period of time. We improve it by creating a new simulation model to represent clothing manipulation as realistically as possible with Bullet Physics, which simulates the response of objects to manipulations using real physics. Additionally, for the simulation to have meaning to the real initial configuration of the clothing article, we need to compare and choose a suitable detection pattern among various kinds of marker patterns, which creates an algorithm to allow the PR2 to detect the important locations of a certain marked-up clothing through its cameras and then generate an accurate guess of the configuration of the clothing article.

After evaluating the detectable pattern size and their conditions under warping configurations, we choose the BCH marker, which is proven to be the most suitable one with our project, as the detection pattern and use ARToolkitPlus to figure out the configuration of the clothing article. ARToolkitPlus, an open source, is one of the most common frameworks for AR implementation, freely downloadable. It uses rectangular markers consisting of arbitrary black- and-white or color patterns, with a training step required to teach the system to recognize a particular pattern set. Through our algorithm, given an image of the marked up T-Shirt, we can extract tracking information of the markers of the shirt, and then determine the pose and location of each marker that was visible on the shirt with this tracking data information. We also create an Interface so that given a picture of a marked up shirt, it can be reproduced in simulation with the same configuration. From the returned locations and pose information of those markers from ARToolKitPlus, we are able to manipulate the mesh model in the simulation to the specified location so that it matches the real configuration of the shirt.

Furthermore, we purport to improve the product of clothing simulation to be more humane. Customers will prefer to buy a shirt without markers or with invisible markers; therefore we present a solution for PR2 to simulate the behavior of a unique marked-up clothing with invisible detection patterns. Infrared (IR), Ultraviolet (UV) and Multi-Frequency markers, using IR/UV cameras or cameras without IR/UV filters as their detection tools, have become the most common methods of invisible detection patterns. Kinect, a motion sensing input device, is also able to detect IR light and some UV light. After the comparison between IR and UV detection, we paint general BCH markers with UV ink and generate the algorithm to detect and identify the UV ink markers using the Kinect system.

This paper will discuss our approaches to the research on different detection patterns and the method to detect and identify UV markers. The first section will discuss the relevant technologies of detection patterns. The second section of this paper is about the methodology by which BCH markers are chosen as our detection pattern and how the detection and identification of UV markers are created. Lastly we will conclude our work for this project and discuss future possible extensions.

3 Literature Review

For a more comprehensive literature review, we refer to the readers several existing surveys on the broad domain of detection pattern research. We focus on the areas that are most closely related to our work: assessment of techniques for detecting and identifying invisible patterns, IR/UV patterns for invisible detection, and edge detection of gray-scale images.

There have been some encouraging results on detection patterns in the context of computer vision. The most common pattern used in daily life is the barcode. Examples include, but are not limited to the following cases [1][2][3]. Moreover, there are set of patterns that can be detected by a computer, which is equipped with a camera and an appropriate detection algorithm. Johannes Köhler et al. have presented an overview of existing marker systems [4]. A thorough comparative study on the design consequences of digital markers was done by Rice et al. [5]. They (Pervasive and Mobile Computing, 2006) used their machine vision framework for a theoretical evaluation of tag readability and a comparison of square and circular tag shapes. Furthermore they conclude that square tags carry a larger symbolic data payload than a circular tag of the same size, whereas circular tags offer better location and pose accuracy. The pose accuracy aspect can be confirmed for the algorithm proposed in Johannes Koehler et al.'s work [6].

A test application of ARToolkit is shown as an example in ARToolkit - Based Tracking [7], It is the most popular detection pattern using in computer vision. Daniel Wagner and Dieter Schmalstieg [8] presented ARToolKitPlus, a successor to the popular ARToolKit pose tracking library. They explained the need and specific requirements of pose tracking on mobile devices and how they met those requirements. To prove the applicability, they performed an extensive benchmark series on a broad range of off-the-shelf handhelds.

Jonathan Mooser et al. [9] provide a systematic way of printing and identifying a vast library of patterns by describing a new fiducials design called TriCodes, which is like a barcode. They compared TriCodes to the popular ARToolkit package, demonstrating its advantages in the presence of large numbers of fiducials.

Hideaki Uchiyama et al. [10] presented a novel approach to detect and track markers with randomly scattered dots for augmented reality applications. Eric Marchand extends planar fiducial markers using random dots for non-rigidly deformable markers [11].

As for invisible detection, Ian Davidson et al. [12] firstly gave a brief idea of UltraViolet (UV) detectors, Infrared (IR) detectors, UV/IR combination devices, Multi-Frequency detectors and Visual Flame Detectors, while comparing and valuing their functions. They also showed the Strengths and Limitations among these different flame detections.

Joseph Kostrzewa et al. [13] compared the Photon design to the Omega with particular focus on aspects which affect manufacturability and cost, therefore, concluded that FLIR Systems' next-generation miniature infrared camera has been specifically optimized for high-volume, low-cost applications for thermal imaging. Gary et al. [14] found that congruent auditory linguistic cues, but not visual cues, significantly improve perceptual sensitivity (as separate from decision bias) for detecting the presence of a visual stimulus. They also investigated the extent of these effects through follow-up studies.

4 Methodologies

4. 1 Patterns summarization

In order to find the most suitable detection pattern for our clothing simulation system, we have to go over several most frequently used patterns that can be detected with camera and an appropriate detection algorithm. Since the purpose of these markers is to return the important locations information of the shirt, we consider only algorithms and techniques that detect multiple markers.

We found square and circular tags to be the most often-used marker trackers, because these geometric primitives are well detectable in images. We look through the marker system and focus on researching into Barcodes, Maxicodes, Cybercodes, Tricodes, BCH markers, occluded circular markers, and random dot markers. See figure 1.

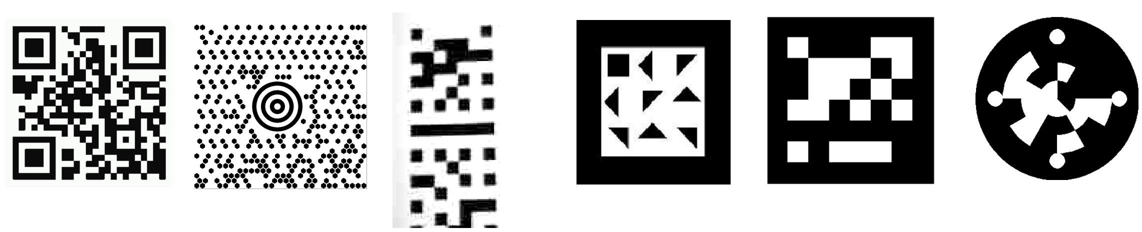
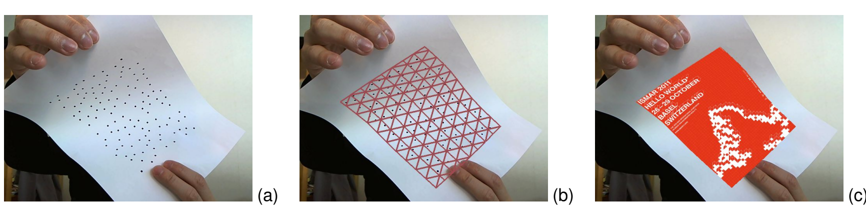


Figure 1. 1) Barcodes, 2) Maxicodes, 3) Cybercodes, 4) Tricodes, 5) BCH markers, 6) Occluded circular markers



7) Random dot markers

For our project, we need to enable the camera to detect and identify the markers within two meters independently. Therefore during our research, we consider more of the minimum detectable tag size in pixel and the amount of possible data part occlusion. Table 1, which is illustrated by Johannes Köhler et al. can be a very good reference to understand the problem and approach our following process. The markers should also be robust enough under warping configuration since the initial state of the shirt is very random. The marker is very difficult to be identified after being warped, because the length of each segment is inconsistent due to the warping. So we also consider if the certain marker can still work through the unwarping algorithm, which could detect the deformed marker image identified through the detected edges.

	# Tags	Red	Occl	FP	MC	MS
ARToolkit [8]	NP	0%	0%	0.5-4.4%+	N/A*	25-65
ARTag [4]	2002	70.2%	5%	0%	0.0039%	20
ARToolkit+ [23]	4096	66.6%	N/A*	0%	N/A	N/A
Intersense [13]	2^{15}	0%	0%	N/A*	N/A*	16*
[11]	255	75%	25%	0%	0.153%	17.36
TriCode [12]	2^{18}	25%	0%	N/A*	N/A	N/A*
reacTIVision [1]	89	0%	N/A	> 0%	N/A	N/A
Nishino [15]	17	0%	N/A	0%	N/A	40
Constanza [3]	N/A	0%	N/A*	0-16%	N/A*	N/A

Table 1 Tags: Tag amount in the library; Red: Redundancy in the identification pattern; Occl: Amount of possible data part occlusion; FP: False positive detection rate; MC: Marker confusion rate; MS: Minimum detectable tag size in pixel. “NP”: not possible for the marker type; “N/A”: no data available; “N/A*”: no precise data available, but facts that allow a comparative estimation. red: Correlation based-; blue: digital method based-; green: topology based identification.

4. 2 Method to detect and identify invisible markers

In order for the simulation to work on invisible markers, we first research on the properties of two common invisible lights: Infrared light (IR) and Ultraviolet light (UV), which are invisible to human eyes and normal cameras but can be detected by unique cameras (i.e. cameras without IR/UV filters). We find that IR is accessible to more cameras, including Kinect system. However UV is more common and more easily to realize in our project, and after testing, we notice that Kinect can also detect UV light. So we replace the previous markers with the invisible ones printed with UV ink.

The initial picture signature of UV marker in UV light detected by the Kinect is showed very clearly in computer. We convert video frames into images using 'cvCreateImage' for the following image processing. However, after gray scale processing and image binarization, the markers' edges are barely noticeable. Since the image signature detection is all done by the Kinect camera system, we rule out the possibility of the effect caused by the low resolution of Kinect camera. On the other hand, although the image of markers is not clear enough to be identified after the gray scale processing, we can still figure out the configuration of the markers. Therefore the results after gray scale processing should also be correct.

After excluding these factors, we focus on the factor of binarization and consider the problem to be result of the inappropriate assignment of binarization threshold value, that is, the problem on operating the function 'cvAdaptiveThreshold' which is contained by opencv. Actually the problem here lies on the parameter block_size (the sixth parameter of the function), which decides the local threshold value 'block'. If the value of 'block' is small enough (i.e. block_size = 3 or 5 or 7), the adaptive control will be very high, which means the pixel values of 'block' are all very similar. At this time the function fails to do the binarization and can only realize binarization at edges where the pixel values differences are very obvious. The result will be edge detection. On the other hand, if the value of 'block' is big enough (i.e. block_size = 21 or 31 or 41), 'cvAdaptiveThreshold' is a binarization function. Yet, no matter how we adjust the parameter, what we can only detect is the edge of the object rather than the marker itself.

So we reanalyze the problem from the beginning and try to find other possible factors that might affect the result. We redo the binarization on a gray scale image with obvious color differences and got very good results. So we suspect the problem to be with the gray scale image. The configuration of the markers is not obvious in gray scale image, which means the marker's gray values are similar with the surrounding gray values and it might be the reason that affects the following processing result. Through using Photoshop we detect the value of every pixel in the gray scale image and proved our previous assumption. The following work should be how to convert the RGB image in UV light into obvious gray scale image where the markers can be easily detected and identified.

Since the function 'cvCvtColor' contained by opencv is not good enough, we decide to generate an algorithm to convert the initial image into gray scale one by ourselves. We grab an RGB image in UV light using Kinect and do the pixel value detection. Of the UV image, the R components of pixels on the marker are almost 255, but the R components of pixels off the marker are just half of 255, although they are under UV light as well. And G/B components of all the pixels in the image are similar with each other. Based on the forming principle of gray scale image, we extract only R component to do the gray scale processing.

Our image generated from Kinect is of four channels, which is different from normal images of three channels. The RGB image detected with Kinect is also saved with four channels, and the R component is saved in the second channel. Through traversing the R-values of all the pixels in the RGB image we can convert the four-channel image into a single-channel image, that is, every Byte expresses a gray value. The algorithm is as followed:

```
Data1 = (BYTE) ColorImage->imageData [j*ColorImage->widthStep + i*4 + 2]
```

After this processing, the new gray scale image generated through opencv is very clear, however there's still a problem: the gray scale image we generate happens to be the opposite of the desired image, i.e. the pixel of gray scale 255 should be of 0. So we traverse the gray scale image again and traverse each point value, i.e. evaluate each point value with '255 – previous point value', and then get the desired gray scale image, with which can we identify the configuration of marker very easily.

Here we can use the previous function 'cvAdaptiveThreshold' or the other two methods: One is to use the function 'cvThreshold' directly to process the binarization; the other is to generate a function by ourselves by which to traverse each point value and then use the function contained by opencv to judge the binarization.

5 Results and Discussion

5. 1 Patterns Evaluation

Through analyzing, we know that the correlation based ARToolkit is probably the most widely used marker tracker due to its availability. And for many other reasons, we choose ARToolkit as our detection pattern.

Considering only the marker occlusion and minimal detectable pattern size, the occluded circular tags showed in figure1-6 should be the best choice. "Compared to square markers, occluded circular tags can be tracked in a more robust way, since the camera pose is in this case computed from the whole contour instead of only the four corners." However, the occluded circular tags are only robust when they are flat. When they are under warped configuration, they cannot be detected or identified successfully and efficiently, which doesn't fit the goal of our project. For the similar reason, barcode, Maxicode and Cybercode also fail to be the candidate of our detection pattern.

For the markers to be working under warped configuration, we consider Random Dot Marker, which is well used in meshing. It can be detected and identified even when they are bended, but it is more often used in generate an image rather than figure out the configuration of an object, which unfortunately didn't meet our requirement either. Moreover, only if one point missing, the detection is not that accurate any more.

How about Tricodes v.s. BCH markers? It has been proved that it is easier to detect triangles than diamond, and the results show that the identification accuracy of Tricodes is higher than that of BCH markers. The reason is that Tricodes use orientation information in detection rather than location information, and orientation information has been proved to be detected more easily. However, even though triangles can be detected through our previous algorithm, it's not as efficient as BCH markers, and it's hard to be processed through the unwarping algorithm, which aims to change the warped markers into their initial configurations (in this case, square). BCH markers can also generate more patterns than Tricodes, and the markers can be much smaller than Tricodes. That's why we choose BCH markers rather than Tricodes in our detetion and identification.

5. 2 UV ink markers' detection and identification

The normal method to detect UV ink image with the function 'cvCvtColor' contained by opencv does not work well. It can only detect the edge of the marker using differential cancellation. However the correct gray value of every pixel cannot be calculated. Through this method we can get images shown in Figure 2.

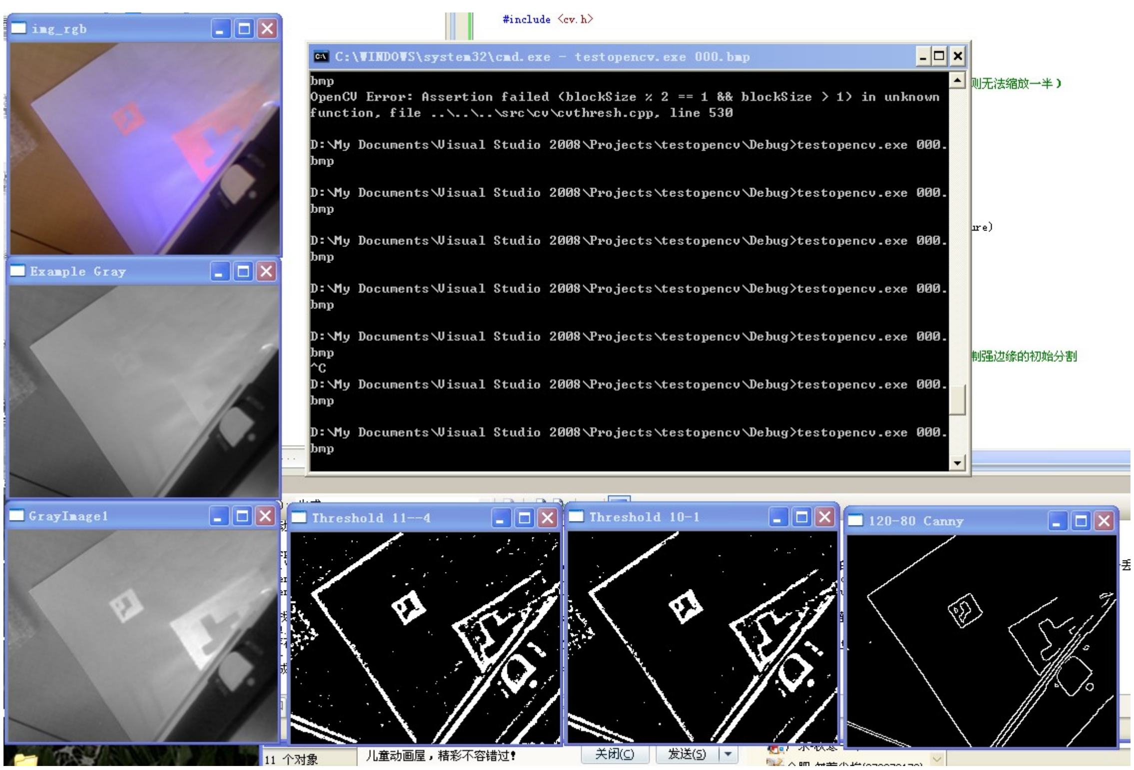


Figure 2
Img-rgb) Initial UV ink image
Example-Gray) Gray scale processing with previous algorithm
GrayImage1) Gray scale processing with the function 'cvCvtColor'
11-4) Adaptive function with previous algorithm
10-1) Adaptive function with the function 'cvCvtColor'
120-80 Canny) Adaptive function with new function Canny

Through the detection algorithm described in 4.2, we are able to convert the UV light image into gray scale image with obvious marker configuration, and accurately identify the UV ink markers, which are invisible to human eyes and normal cameras.

Figure 3 shows an example of the UV ink marker detection algorithm on marker #0.

'GrayImage2' and 'Example Gray' are the results of the function 'cvThreshold', which process the binarization directly:

cvThreshold(grey,adaptiveImg,155, 255, CV_THRESH_BINARY_INV);

'GrayImage1' and 'GrayImage4' are the results from the algorithm generated by our group, where we traverse each point value and then use the function contained by opencv to judge the binarization:

cvThreshold(grey,adaptiveImg,155, 255, CV_THRESH_BINARY);

From the figure we can conclude that the results of these two methods are similar, and we can use either way to reach our goal.

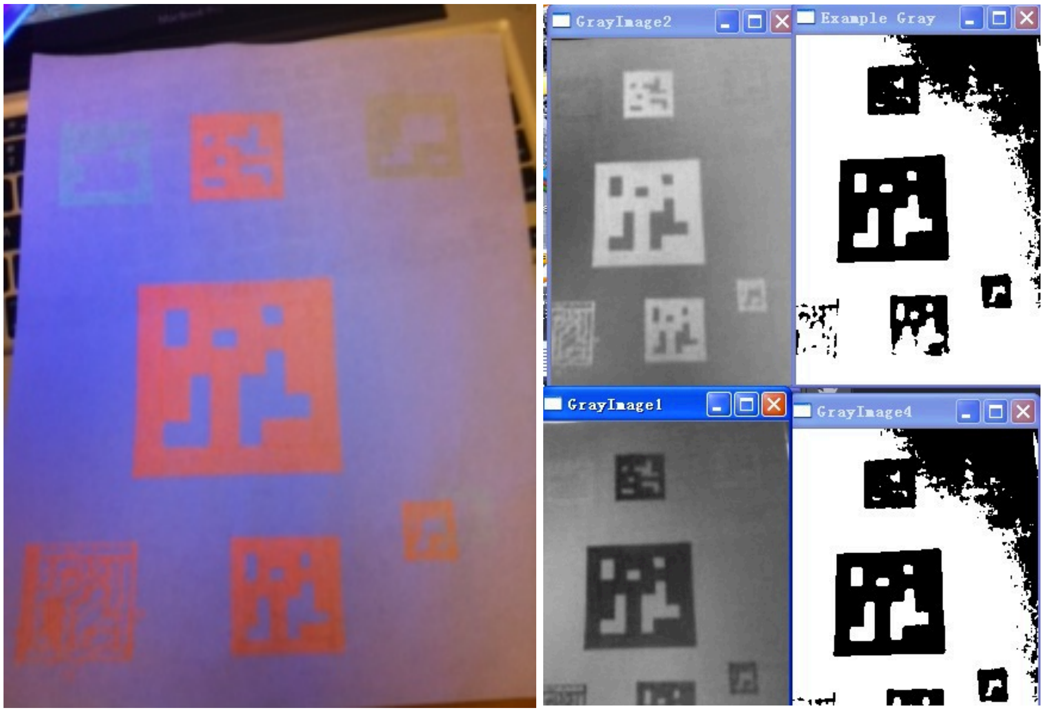


Figure 3. 1) Initial UV ink marker image (RGB), 2) Gray scale image through detection algorithm

Figure 4 shows the whole process of detecting and identifying the UV ink marker through our algorithm on marker #0. From the algorithm we can recognize marker #0 successfully. The same process and result applies to all other markers.

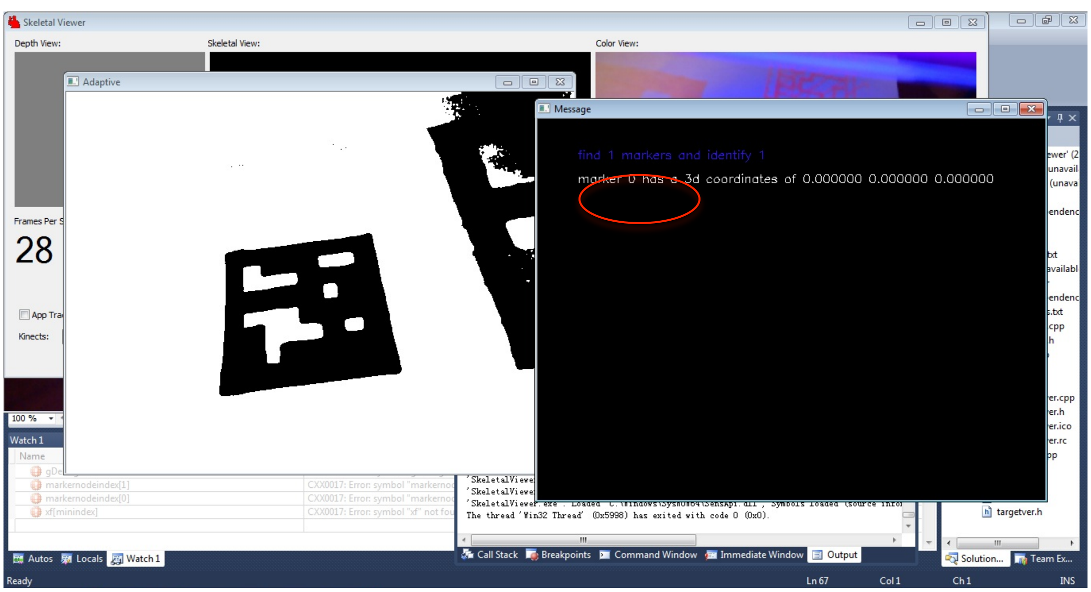


Figure 4. Identification to marker #0

6 Conclusions

From the project learned about the advantages and disadvantages of some common detection patterns, and come out with the best solution for our simulation project. On the other hand, we present a new solution for PR2 to work on UV invisible ink markers, and we obtained a deeper knowledge on the binarization processing, where the most important factor is to improve the converting quality from RGB image to gray scale image. We also learned about the storage pattern of RGB 3-channel-image, UV 4-channel-image and gray scale image. In general, an image with N channels is represented with N continuous BYTEs, and every pixel is expressed by the combination of these N BYTEs.

In future, we will keep thinking about replacing UV invisible markers with better solutions that can be accepted by customers more easily. We will also think about using the markers as ground truth data for computer vision algorithms. For the whole simulation project, we would like to come out with a better lead for the PR2 to follow and fold clothes, and is able to avoid useless movements automatically.

7 Acknowledgment

This research project would not have been successful without the support of many people. We want to express our gratitude to the supervisor, Prof. Pieter Abbeel, who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude are also due to members of the Berkeley Robotics Lab, John Schulman without whose knowledge and assistance this study would not have been successful.

Special thanks also to Prof. Jitendra Malik, Prof. Trevor Darrell for sharing the literature and invaluable assistance. Not forgetting to our friends who always been there. Our team wish to express our love and gratitude to beloved families and friends for their understanding and endless love, through the duration of this project.

References:

[1] Tat-Jun Chin, Hanlin Goh and Ngan-Meng Tan. Exact Integral Images at Generic Angles for 2D Barcode Detection. Computer Vision and Image Understanding Department. Institute for Infocomm Research, Singapore
[2] Patrick Martin and Jason Scott. Automatic 2D Barcode Location and Recognition
[3] Tat-Jun Chin, Hanlin Goh and Ngan-Meng Tan. Exact Integral Images at Generic Angles for 2D Barcode Detection. Institute for Infocomm Research, Singapore
[4] Johannes Köhler1, Alain Pagani1, and Didier Stricker1. Detection and Identification Techniques for Markers Used in Computer Vision. Department of Augmented Vision. German Research Center for Artificial Intelligence GmbH. Trippstadter Str. 122, 67663 Kaiserslautern, Germany
[5]Andrew C. Rice, Robert K. Harle, and Alastair R. Beresford. Analysing fundamental prop- erties of marker-based vision system designs. Pervasive and Mobile Computing, 2(4):453– 471, November 2006.
[6]Johannes Koehler. Robust detection and identification of circular markers for estimation of the camera pose in real time. Master's thesis, Technische Universität Kaiserslautern, 2009.
[7] ARToolkit - Based Tracking: A New Device in VRPN
[8] Daniel Wagner and Dieter Schmalstieg. ARToolKitPlus for Pose Tracking on Mobile Devices. Institute for Computer Graphics and Vision, Graz University of Technology
[9] Jonathan Mooser, Suya You, and Ulrich Neumann. TRICODES: A BARCODE-LIKE FIDUCIAL DESIGN FOR AUGMENTED REALITY MEDIA. CGIT Lab, University of Southern California
[10] Hideaki Uchiyama, Hideo Saito. Random Dot Markers. Keio University
[11] Hideaki Uchiyama, Eric Marchand. Deformable Random Dot Markers. INRIA Rennes Bretagne-Atlantique Universite ́ de Rennes 1, IRISA,
[12] Flame Detector Selection
[13] Beyond Omega: Next-Generation Miniature Infrared Camera
[14] Gary Lupyan1*, Michael J. Spivey2. Making the Invisible Visible: Verbal but Not Visual Cues Enhance Visual Detection. University of Pennsylvania, Philadelphia, Pennsylvania, United States of America, 2 University of California Merced, Merced, California, United States of America