ArUco Markers

Project: Marker-based Localization

Niharika Jayanthi, Dheeraj Kamath
Mentor: Sanam Shakya

Goal

In this documentation, we shall learn-

- ArUco markers; encoding and decoding
- How to detect and decode markers from video feed

Prerequisites

- Hamming code

Theory

Introduction

ArUco marker is a 5x5 grid that is black and white in color. ArUco markers are based on Hamming code. In the grid, the first, third and fifth columns represent parity bits. The second and fourth columns represent the data bits. Hence, there are ten total data bits. So the maximum number of markers that can be encoded are-

2^10 = 1024

Encoding

Let us consider the number 650. It's binary representation is 1010001010. There are two data bits in each row.

Now, each row is encoded separately using slightly modified Hamming code. The first and third parity bits are calculated using even parity while the second parity bit uses odd parity. We get the following encoded values-

Data bit 2	Parity bit 3	Data bit 1	Parity bit 2	Parity bit 1
0	0	1	0	1
0	0	1	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	0	1

We rearrange the bits in each row and get the following result-

If the cell value is 0, color it black; if value is 1, color it white. This will give us the ArUco marker-

	1		0	
	1		0	
	0		0	
	1		0	
	1		0	

Figure 1: Data rows

Parity1	Data1	Parity3	Data2	Parity2
0	1	0	0	1
0	1	0	0	1
1	0	0	0	0
0	1	0	0	1
0	1	0	0	1

Figure 2: ArUco bits for 0b1010001010

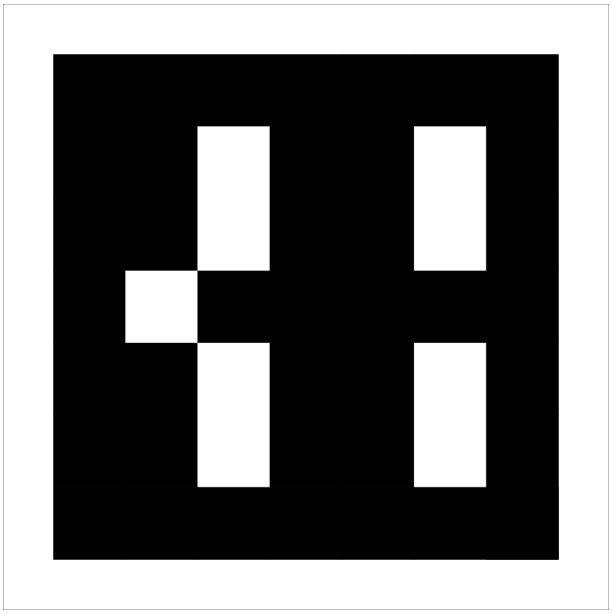

Figure 3: ArUco marker

The above marker is padded with one layer of black cells.

Decoding

After understanding the above section, decoding is an extremely simple process. The following steps are to be followed while decoding a perfect, computer-generated image of an ArUco marker.

Step 1 : Extract the ArUco from the image.

Step 2 : Remove the extra padding.

Step 3 : Divide the resulting image into a 5x5 grid and check the color in each cell of the second and fourth columns(in that order) in a top to bottom manner.

Step 4 : If the color is white, write 1; else, write it 0.

Step 5 : The resulting number will be in binary. Convert it into decimal

Applications

- These markers are utilized in augmented reality applications.
- They can be used in robot navigation algorithms.

Code

The main functions used are -

cv2.threshold()

cv2.approxPolyDP()

The code is written using the above-mentioned steps:

```
#Imports

import cv2

#Define globals

MAX_SIZE = 406

#Define helper functions

def extractAruco(img):

    #Inverse threshold to get the inner contour
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, th = cv2.threshold(gray, 127,255,cv2.THRESH_BINARY_INV)

    #Crop image
    contours, hierarchy = cv2.findContours(th, cv2.RETR_EXTERNAL,
                                           cv2.CHAIN_APPROX_SIMPLE)
    contours = contours[0]

    epsilon = 0.1*cv2.arcLength(contours,True)
    approx = cv2.approxPolyDP(contours,epsilon,True)

    image_crop=gray[approx[0,0,1]:approx[2,0,1],approx[0,0,0]:approx[2,0,0]]

    #Return extracted image
    return image_crop


def findArucoID(inp_img):

    #Extract image
    im = extractAruco(inp_img)

    #Resize it to smaller size for check
    im = cv2.resize(im, (MAX_SIZE,MAX_SIZE))
```

```
    #Remove padding
    width = MAX_SIZE / 7
    im = im[width:width*6,width:width*6]

    #Calculate ID
    ret_val = 0

    for y in range(5):
        _val1 = int(im[(y * width) +  (width / 2), width + width / 2]) #im[y, x] format
        _val2 = int(im[(y * width) +  (width / 2), 3 * width + width / 2]) #im[y, x] format

        if _val1 == 255:
            _val1 = 1

        if _val2 == 255:
            _val2 = 1


        ret_val = ret_val * 2 + _val1
        ret_val = ret_val * 2 + _val2

    return ret_val



image = cv2.imread("your_example.png")
ID = findArucoID(image)

print "Marker ID is", ID
```

Resources

- http://www.uco.es/investiga/grupos/ava/node/26
- http://iplimage.com/blog/create-markers-aruco/
- http://terpconnect.umd.edu/~jwelsh12/enes100/markergen.html

Exercises

- Write a code to decode markers present in an image taken by a camera.
- How will you detect a rotated marker?