Object Detection and Pose Tracking In Augmented Reality

Vipul P.Chauhan[1], Dr. Manish M. Kayasth[2]

[1]Veer Narmada South Gujarat University, Department of Computer Science, Udhna Magdalla Road, Surat-395007, Gujarat, India

[2] Udhna Academy College Of Computer Application And Information Technology, 214, Ranchhodnagar, Opp. Swaminarayan Temple, Surat-Navsari Road, Udhna, Surat - 394 210, Gujarat, India

Abstract: ***These papers introduce technique for object detection and pose tracking with a monocular camera for augmented reality applications. To visually merge a virtual object onto a real scene with geometrical consistency, a camera pose with respect to the scene needs to be computed. AR Toolkit is software that provides the facility to create AR systems. This paper is study a fiducially marker based Augmented Reality system, the algorithms it uses and how it can be implemented.***

Keywords: Augmented Reality (AR), Detection, Tracking, Pose Estimation, Marker detection algorithms.

1. Introduction

AR is an abbreviation of Augmented Reality, which in turn is the term for the concept of overlaying computer generated images on real-life images. Thus the images of reality are augmented with new information that can give the viewer a better sense of their contents.

Augmented reality (AR) is an increasingly recognized paradigm of information visualization that merges a computer-generated virtual object onto a real scene to visually augment reality [1]. Azuma has summarized three requirements in AR applications as follows [2]:
(1) combines real and virtual,
(2) Interactive in real time,
(3) Registered in 3-D.

To achieve these requirements, numerous technologies on computer vision and image processing have made immense contributions. With a monocular camera, 3D structure of a real scene is sensed to enable overlaying virtual objects with geometrical consistency in real-time

2. Types of Augmented Reality

Augmented reality can be implemented by using two types of technology: optical or video. Optical AR refers to a see through device (e.g. see through head-mounted display) that will allow the user to see the world but also see the superimposed graphics on the display. Video AR refers to video captured by a camera, which then is processed to add the superimposed graphics and displayed the resultant image to a monitor. Video based AR systems have certain advantages over the optical AR systems. Optical systems have trouble to obscure real world objects, because they cannot filter out the light from the real world object. On the other hand, video systems are more adaptable to handle this problem, because they can choose between the real and the virtual pixels since they have a digital image of the scene. Optical systems have issues with mismatching the real and virtual images due to the delay caused by the view of virtual images. In contrast, video systems can delay the real world images to match the delay on the virtual images.

Optical systems can only register information from the head tracker. However, video systems can utilize many more registration techniques because a digital image of the scene can be obtained.

3. Registration Techniques

One of the fundamental issues for augmented reality is the registration problem. The superimposed graphics are required to be aligned with the real world objects in order to preserve the harmony of the blending between the real and virtual environment. Some AR applications need accurate registration to function properly such as manufacturing a complex machine using 3-D annotated text as help.

Recently, the algorithms of object detection and pose Tracking have been incorporated in AR applications. In such systems, a number of objects are first registered on a database. When one of the registered objects is captured, it is recognized, tracked and localized to overlay its corresponding virtual object on it. For target objects, various types of size and shape can be used such as a small planar paper, a large room and so on. In the literature, many approaches of localization have been proposed. One of the techniques is to use fiducials (e.g. LEDs, markers, colored dots). The system already knows the fiducials locations in the real world and uses image processing to obtain the locations on the image. Using these two sets of locations, the system can apply the correct registration. However, the fiducial have to be always visible to prevent the registration from breaking.

Another registration technique is to use template matching. The system would take template images of the real world object used for the registration from different viewpoints. Then once the camera captures an image, the system will search for the object on the image using the previously taken templates.

These two approaches need camera calibration, which is finding the camera intrinsic values that would affect the image such as focal length, principal point, scale factors, lens distortion factors and others. They only provide location information that is relative to the camera.

4. Augmented Reality Toolkits

There are many toolkit programs are available on the internet that allows the user to create fiducial marker-based AR systems. The two most important ones are ARToolkit and ARTag.

4.1 ARToolkit

ARToolkit [3] is a widely used AR fiducial marker system which is known because it is simple, relatively robust, and freely available. It was created in 1999 by Hirokazu Kato in University of Washington HIT Lab. ARToolkit was one of the first programs used to develop AR systems. There are several others that have been based on ARToolkit, such as OSGART, FLARToolkit, NyARToolkit, AndAR and others. Most of these programs are ARToolkit versions ported to a different platform/language.

There are many steps involved in inserting computer generated images into real life images, but the theory behind most of these is not very complicated. The process begins with a camera taking an image and sending it into the pipeline. The image is then searched for special markers. If such a pattern is found, the program continues by trying to identify it by matching it to several patterns in a precompiled database. If a match is found the program knows what object to associate with that specific pattern. The information from the pattern in the camera image is used to determine its location and orientation relative to the camera. After that it is just a matter of rendering the object in the correct place. The main algorithm this program uses is shown in Figure 1.

4.2 ARTag

ARTag [4] was developed in 2004 by Mark Fiala in the Computational Video Group of the Institute for Information Technology of the National Research Council Canada. It improves on ARToolkit by using a more complex image Processing and takes advantage of more powerful computational machines. One of the main advantages are ARTag is able to register markers under uneven lighting, while ARToolkit cannot entirely. Another advantage is that ARTag can detect a marker even if it is occluded by a considerably large object. ARToolkit detection would break down if the marker is occluded by a few pixels.

5. Image Processing and Marker Detection Methods

It is crucial for AR applications to make the marker detection process as quick as possible. Therefore, a good marker is considered to be a marker that can be easily and reliably detected under different circumstances such as different lighting conditions, image colors and poses of the camera. Basically, the process of marker detection consists of two basic stages: 1) image processing and 2) identification of potential markers. Fig.2 illustrates marker processing algorithm and its steps related to image processing and detection of potential markers stages, which may be summarized as follows [5][6]:

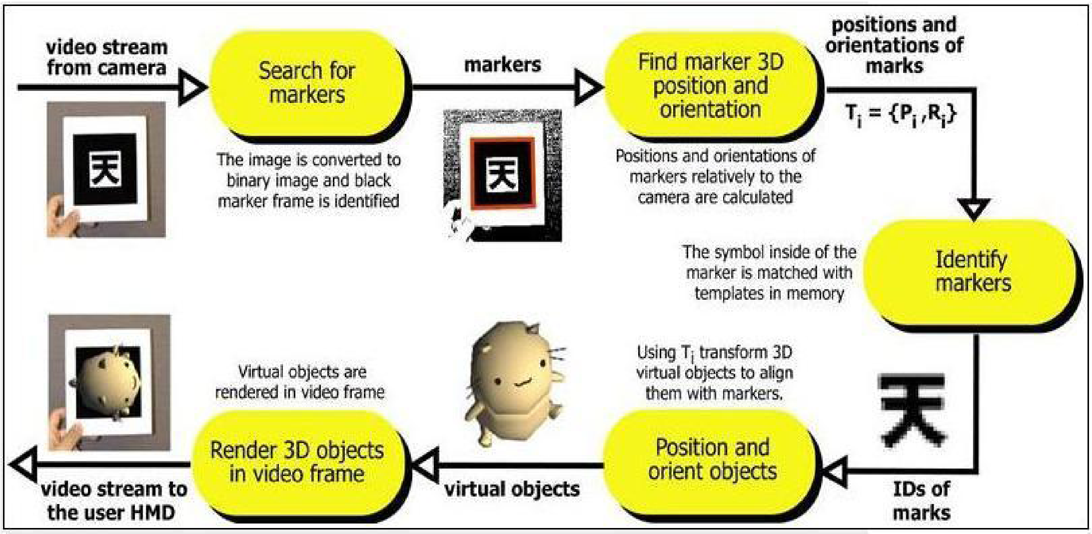


Figure 1: Algorithm to marker reorganization and pose Estimation process

1. Acquisition of a source image

2. Image processing:
- Gray scaling
- Thresholding

3. Detection of potential markers:
- Identification of connected components
- Extraction of contours of the objects
- Extraction of marker edges and corners
- Determination of marker square

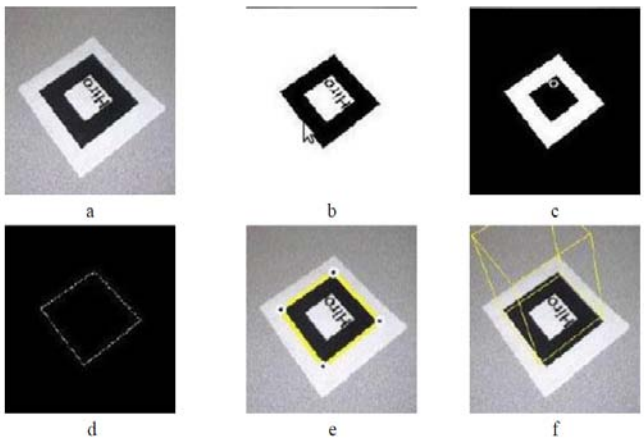


Figure 2: Illustration of marker processing algorithm steps: original image (a); thresholded image (b); connected components (c); contours (d); extracted marker edges and corners (e); fitted square (f) [5].

The first goal of a marker detection process is to find the outlines of potential markers, and then to deduce locations of marker's corners in the image. In addition, detection system needs to confirm that it really is a marker and decipher its identity. Finally, the system calculates the pose using the information from the detected marker location.

5.1 Image acquisition Stage

5.2 Image processing Stage

Before a potential marker can be identified, it is necessary to ensure that the system will be able to use the obtained original image for the marker detection process. It has been widely recognized that markers in black and white images are more easily detected than markers in color images. There are many methods are used to process the original image and prepared it for further use.

5.2.1 Colour-to-greyscale conversion methods

Greyscale images are often used in modern image detection and recognition systems rather than colour images, since the use of greyscale images simplifies significantly computations and algorithms. The most common colour-to-greyscale conversion algorithms are based on the use of red, green and blue colour channels denoted as R, G and B, respectively. The main principle of these conversion methods is converting colours in an image to shades of grey. The easiest and most straightforward grayscale algorithm is *Intensity* (also called Average) that represents the arithmetic mean of RGB channels as follows [7]:

Grayscale Intensity = 1/3 (R + G + B)

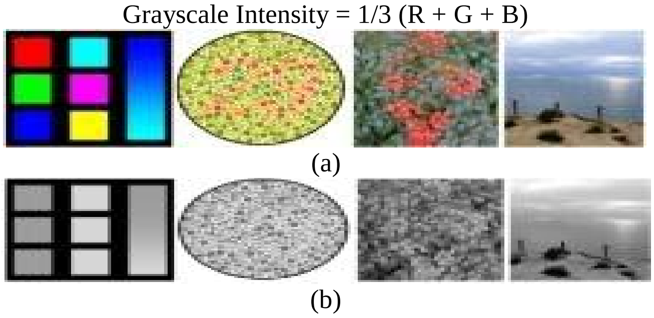


Figure 3: Comparison of the original color images (a) and greyscale images using Intensity algorithm (b)

Luminance algorithm is developed to match more accurately human eye's perception of brightness and represents the weighted mean of the RGB channels [6]:

Grayscale Luminance = 0.3R + 0.59G + 0.11B

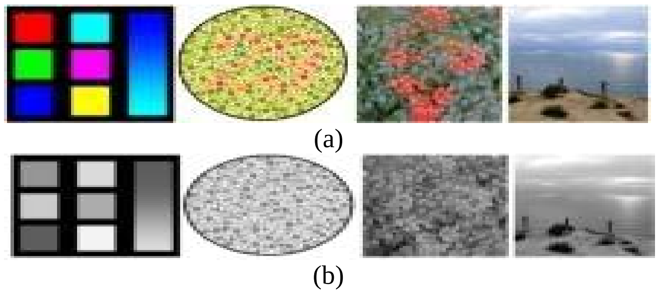


Figure 4: Comparison of the original color images (a) and greyscale images using Luminance algorithm (b) [7].

Value algorithm is the method that uses the maximum of the RGB values and, consequently, provides the brightest image colors [6]:

Grayscale Value = max (R, G, B)

Luster algorithm is based on the principle that all RGB colors must be fully saturated, which makes it less sensitive to variations in brightness. It is computed as the arithmetic mean of the maximum and minimum of RGB colors [7]:

Grayscale Luster = 1/2 (max (R, G, B) + min (R, G, B))

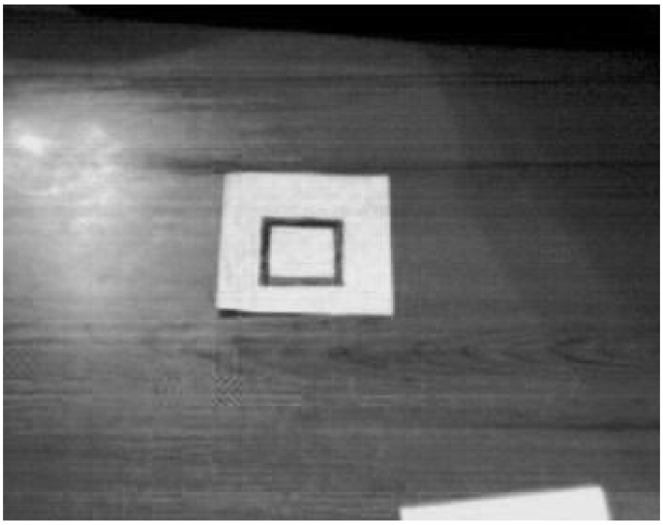
Decolorize is an alternative approach used to transfer colour images to greyscale that attempts to preserve and enhance colours. It is based on a complex linear algorithm that was found to be the best in preserving highest colour contrast of images as compared to other greyscale conversion methods [7].

5.2.2Thresholding methods

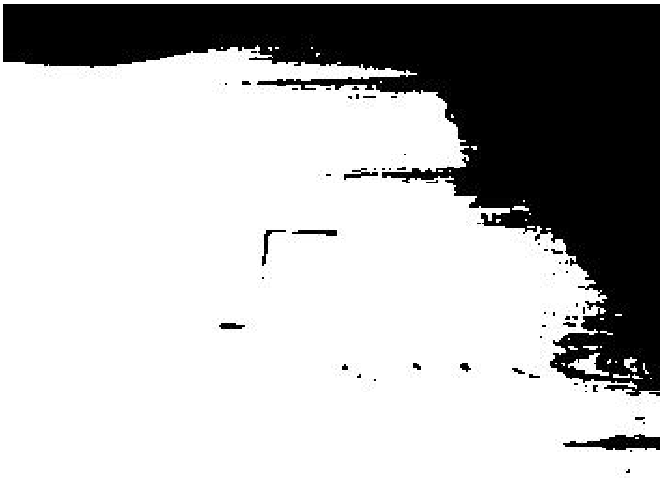
Thresholding is an effective technique used for image segmentation and creation of binary images from greyscale images. In the output image all pixels with the luminance level above a set threshold can be represented by gray-level 1, that is white, while the pixels below luminance values can be assigned with gray-level 0, that is black. Thresholding is very useful for separating gray pixels of objects from gray pixels of the background, so that the foreground and background can be easily identified as the result [6].

Thresholding algorithms:

Histogram shape-based algorithms are based on the use of histogram shape information. This class of algorithms involves analysis of the peaks, valleys and curvatures of the histogram. Some algorithms within this class focus on the calculating of the convex hull of the histogram and analysing its concavities points, where the deepest points are considered as potential thresholds. Other shape-based algorithms are based on curvature analysis, where the smoothed histogram is examined for the presence of peaks and valleys, and the valley point after the first peak becomes a candidate for a threshold [8].



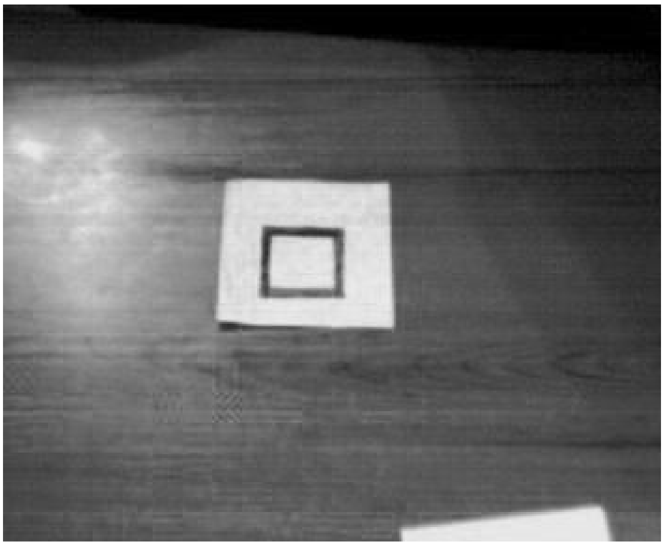
(a)



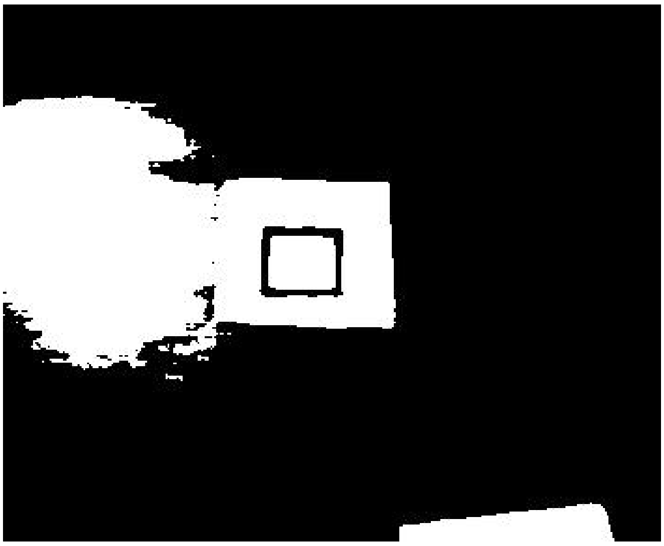
(b)

Figure 5: Comparison of the original greyscale image (a) and binarized image using strictly defined Thresholding (b) [9].

Clustering-based algorithms are based on the exploitation of Measurement space clustering. This category of algorithms focuses on clustering analysis, where gray-level pixels are examined and allocated to two clusters corresponding to two parts of a histogram, i.e. as the foreground (object) and background. Some algorithms within this category consider searching for a midpoint of the two histogram peaks, which are the highest and lowest gray values, so that a threshold is defined using the mean of the foreground and background. Other clustering-based algorithms assume that an appropriate threshold can be found by minimizing a weighted sum of the intra-class variances [8].



(a)



(b)

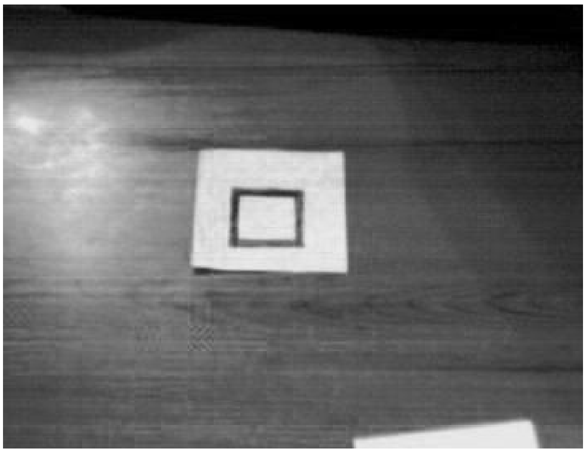
Figure 6: Comparison of the original greyscale image (a) and binaries image using Otsu method (b) [9].

Entropy-based algorithms are based on the use of histogram entropy information. In other words, this group of algorithms measures the entropy of the gray-level distribution in an image. Some algorithms within this group are based on the idea of the entropy maximization, so that, when the sum of the entropies of the image foreground and background achieves its maximum, the image is considered to be ideally threshold. Other entropy-based algorithms focus on minimizing the cross-entropy between the original gray-scale image and the final binary image, so that, the optimal threshold is selected, when there is a minimum data consistency between the images [8].

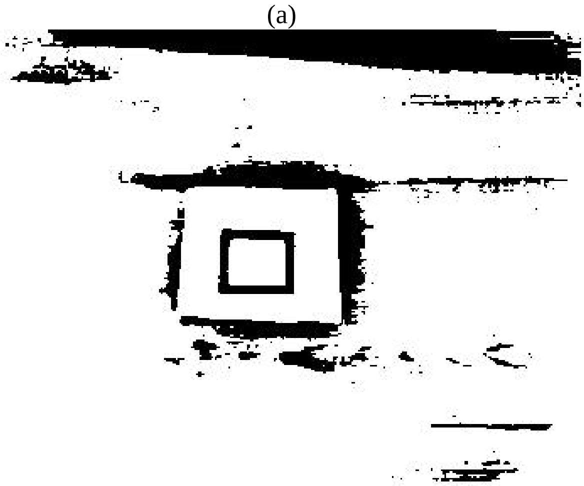
Object attribute-based algorithms achieve Thresholding based on the use of the image attribute information. This category of algorithms is based on identifying a measure of similarity of attributes between the original gray-scale image and the final binary image. Some algorithms within this category focus on searching coincidence in edge field attributes, where a threshold is determined by the maximum count of matching edges of the two edge fields. Other object attribute-based algorithms discuss moment preserving

Thresholding, where a threshold is found by matching the first three moments of the original greyscale image with first three moments of the final binary image [8].

The AR *Locally adaptive algorithms* focus on local characteristics information and local adaptation of a threshold for each image pixel. In this group of algorithms a threshold is found based on some local indicators such as range, variance or contrast. Some of the algorithms within this group are based on local variance method, where the threshold is calculated using the mean and standard deviation at each pixel. Other locally adaptive algorithms use local contrast method, where a gray value of each pixel is compared with the gray level in the surrounding area around the pixel, so that if the pixel is darker than the average it is classified as the foreground (object), otherwise it is classified as the background [8].



(a)



(b)

Figure 7: Comparison of the original greyscale image (a) and binarized image using adaptive Thresholding (b) [9].

5.3 Marker Detection Stage

5.3.1Identification of connected components

Connected components analysis is performed in respect of a binary image in order to fill in the gaps within a well-connected area and finally identify the closed areas in the image. The algorithm makes first pass scan of each pixel in the binary image, checks its connectivity to all its eight surrounding pixels and records equivalences between neighbouring labels. Then, during second pass algorithm divides elements into the closed areas and finally produces labelled maps of the connected components [10, 11].

5.3.2 Extraction of the contours of the objects

Contours of the objects on the image need to be extracted in order to determine the curve that represents a border of the marker. This curve will be used for future extraction of corners of the marker. The refined boundaries can be extracted by the set of edge detection algorithms.

The Marr-Hildreth edge detector is based on identification of the changes in intensity of the image at a given scale using the Laplacian (i.e. the second derivative) of the Gaussian, and the intensity changes detected are called zero-crossing segments [12]. So that, the basic steps of this algorithm involve smoothing the image by a Gaussian, application of the Laplacian and detection of the edges by identification of the slope across sign changes above a selected threshold [13].

The Canny edge detector achieves edge detection by the gradient method, where edges are detected by searching for the maximum pixels in the first derivative of the Gaussian [14]. The stages of this algorithm include smoothing the image by calculation of a two-dimensional Gaussian, determining the intensity gradient of the image, suppressing all non-maximum pixels by computing local maximums in the gradient magnitude and direction, and applying hysteresis Thresholding, where the edges pixels are found to be the pixels above a set threshold [13].

The Euclidian Distance and Vector Angle based edge detector employs two operators for edge detection – Euclidian Distance and Vector Angle. The Euclidian Distance allows determining edges on the basis of the pixels intensity, while the Vector Angle operator identifies edges on the basis of hue and saturation [13]. As compared to other edge detectors that use only one colour channel, this edge detector uses colour components of pixels. So that, the common stages of this algorithm include calculating the combination of both operators for the centre point and eight surrounding points of each pixel, assigning the maximum value to the centre point and, finally, applying a threshold for a new matrix image to remove false edges [15].

The depth edge detection using multi-flash imaging is based on detecting edges by taking multiple images with different light source positions. Based on the location of shadows the depth edge matrix is obtained. This algorithm steps include processing each additional image by removing the ambient component (i.e. image with ambient light) and calculating the light vector for each pixel. Then, the equation is applied to each light vector, which allows determining whether a pixel should be marked as an edge [13].

(a) (b) (c)

Figure 8: Illustration of edge detectors: Original image (a), Marr-Hildreth edge detector (b), canny edge detector (c).

5.3.3 Extraction of marker edges and corners

Recognition of marker square requires the extraction of edges and corners, which is performed by polygon approximation algorithm called *Douglas-Peucker algorithm* (also called Ramer-Douglas-Peucker algorithm, iterative closest point algorithm, the algorithm partitioning and merging). This algorithm is based on finding distance dimension for each point on the line and reconstruction of the simplified curve. Using this algorithm marker corners are found as the points of simplified square [16][6].

5.3.4 Determination of marker square

After identification of the coordinates of the marker corners, which ideally are to be perpendicular, but in reality are usually found to be at different angle. These corners, either being perpendicular or at different angle, represent two sides of the square and correspond to the axes. Therefore, the position of a camera with respect to the facility and the starting point of the origin can be determined. The idea behind this is that if camera's angle changes, then the projection size will change correspondingly [6].

Knowing the position of the camera and the reference point, it is possible to draw a projection of 3D model, in particular case, a cube. If a square with clear field is used as a marker and it is symmetric, then detection of rotation can be done only partially. Sometimes, it can be a satisfactory result, but if a superior result is required, then it possible to identify an additional marker inside the square and obtain a result with some angle of rotation, using *Hough transform algorithm* [17][18][6].

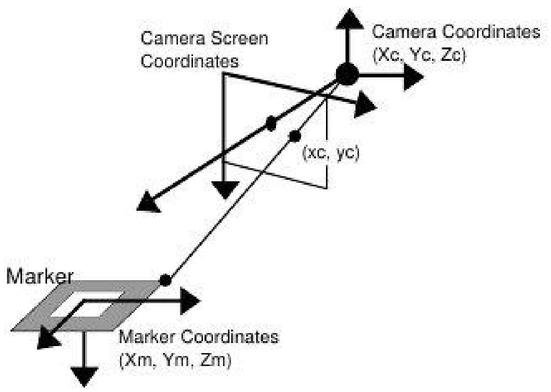


Figure 8: Illustration of calculation of marker coordinates [17]

6. Conclusion

In This paper we provide technique for object detection and pose tracking using ARtoolkit and ARtag for AR application. This paper provides algorithm to marker detection and pose tracking. In this paper we also provide information about different types of image Thresholding and Line detection algorithms which are used in marker detection Procedure.

References

[1] B. Furth, editor. *Handbook of Augmented Reality*. Springer.
[2] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
[3] ARToolkit http://www.hitl.washington.edu/artoolkit/
[4] ARTag http://www.artag.net/index.html
[5] ARToolKit: Computer Vision Algorithm. Internet WWW-page, URL:http://www.hitl.washington.edu/artoolkit/documentation/vision.htm (14.09.14) .
[6] Vadim Beglov ,"Object information based on marker recognition," University of Eastern Finland, July 2013
[7] Christopher Kanan , Garrison W. Cottrell: "Color-to-Grayscale: Does the Method Matter In Image Recognition? ," PLoS ONE, Volume 7, 2012.
[8] Mehmet Sezgin: "Survey over image Thresholding techniques and quantitative performance evaluation. ," Journal of Electronic Imaging 13 (1), 2004, 146–165.
[9] Andrew Babiy, "Augmented reality marker recognition.", Internet WWW-page, URL: http://habrahabr.ru/post/135659 .
[10] João Paulo Lima, Germano Guimarães, Guilherme Silva, João Marcelo,Teixeira Emanoel Xavier Veronica Teichrieb Judith Kelner: "ARCam: an FPGA-based Augmented Reality Framework," Universidade Federal de Pernambuco (UFPE).
[11] Wikipedia: Connected-component labeling. Internet WWW-page, URL: http://en.wikipedia.org/wiki/Connected component_labeling
[12] D. Marr, E. Hildreth,"Theory of Edge Detection," Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 207, No.1167, 1980, 187-217.
[13] Parvinder Singh Sandhu, Mamta Juneja, Ekta Walia, "Comparative Analysis of Edge Detection Techniques for extracting Refined Boundaries," International Conference on Machine Learning and Computing, IPCSIT vol.3, 2011
[14] CPSC 425 course material: Marr/Hildreth and Canny Edge Detection. Internet WWW-page, URL: http://www.cs.ubc.ca/~woodham/cpsc425/example /edge.html
[15] Slawo Wesolkowsky, Ed Jernigan,"Color edge detection in RGB using jointly Euclidean distance and Vector angle, Vision Interface, 1999.
[16] Wikipedia: Ramer–Douglas–Peucker algorithm. Internet WWW-page, URL: http://en.wikipedia.org/wiki/RamerDouglasPeucke algorithm
[17] Hirokazu Kato, Mark Billinghurst, "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System," IWAR '99 Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, 1999, 85.
[18] Xiang Zhang, Stephan Fronz, Nassir Navab,"Visual Marker Detection and Decoding in AR Systems: A Comparative Study," ISMAR '02 Proceedings of the 1st International Symposium on Mixed and Augmented Reality, IEEE Computer Society Washington, DC, USA, 2002, 97.
[19] Gianpierre Villagomez, "Augmented Reality," University of Kansas .
[20] Hideaki Uchiyama, Eric Marchand,, "Object Detection and Pose Tracking for Augmented Reality:Recent Approaches," published in irisa,2012

Author Profile

Chauhan Vipul received the B.Sc. degrees in Computer Science from VNSGU, Surat 2009 and M.Sc. degrees in Computer Science from Ganpat University, Mesana 2011. He is Student of M.phil in VNSGU, Surat. He is interested in Research area of OOP programming languages, Image Processing, Database and Augmented Reality.

Dr. Manish M. Kayasth is a Vice Principal Udhna Academy College Of Computer Application And Information Technology. He is awarded with degrees of B.Sc, L.L.b, M.C.A and PhD (computer science), His interested Research area is Image processing. He is also provides guidance for M.phil and PhD.