International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2016, pp. 520-527
e-ISSN:2278-621X

MARKER BASED AR FOR A BETTER UNDERSTANDING OF VISUAL LEARNING

Jonathan Fernandes[1], Joseph Narasisa[2], Kevin Fernandes[3] and Sreenivasa B.L[4]

Abstract: This paper that enables Marker Based Application in Augmented Reality (AR) on the Android platform. Augment Reality combines the real world with virtual reality, 3D objects in real time. The proposed application is an android mobile based application to allow the user to view the virtual object in the real world using a marker based AR system. The images of an object's front, back, top, bottom, left and right side will be placed onto a 3D cube which will make up the complete virtual object. Then an environment is created through the amalgamation of the real world and the generated object will appear as has though it exists in the real environment. The advantage of this application is that it would display the object in 3D and it can be rotated virtually.
Keywords –: Marker Based AR, Marker-less AR, Image Capturing

I. INTRODUCTION

Augmented Reality (AR) a new technology that blends virtual objects into the real world. Augmented reality (AR) is alive direct or indirect view of a physical and real-world environment, One of the best overviews of the technology is that defined the field, described many problems, and summarized the developments up to that point. That paper provides a starting point for anyone interested in researching or using Augmented Reality. The different types of AR include Recognition, Location, Sensor, Outline and Projection.

Recognition: In this type of augmented reality it uses shapes for recognition, faces or real time items to provide virtual information to the user in real world.
Location: This type uses the GPS technology to provide us with the relevant directional information.

Sensor: A **sensor** is an object whose purpose is to detect events or changes in its environment. The position andorientation are determined by fusing information from a camera and an (IMU) Inertial Measurement Unit.

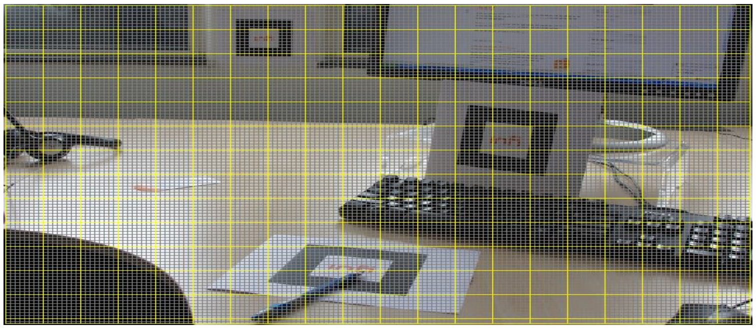
Outline: In this type of AR the outline of the human body or a part of the body are merged with the virtual materials.

[1] *Aloysius Institute of Management and Information Technology Department of Information Technology, Mangalore, Karnataka, India*
[2] *Aloysius Institute of Management and Information Technology Department of Information Technology, Mangalore, Karnataka, India*
[3] *Aloysius Institute of Management and Information Technology Department of Information Technology, Mangalore, Karnataka, India*
[4] *Aloysius Institute of Management and Information Technology Department of Information Technology, Mangalore, Karnataka, India*

II.PREFERRED ALGORITHM

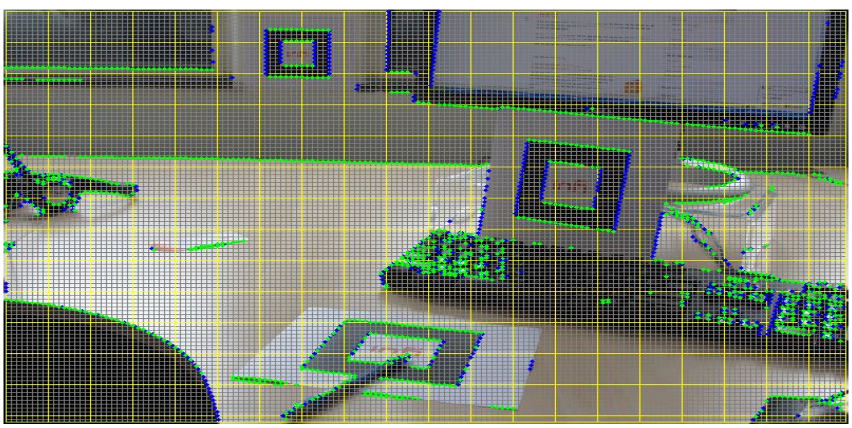
A. Infi Algorithm

Step 1: Divide image in regions



First the image is divided in small regions of 40x40 pixels and each region is divided into horizontal and vertical scan lines 5 pixels apart.
The next 3 steps are executed apart inside these regions which boosts the performance dramatically.

Step 2: Identify the edges in the region
Then a derivative of Gaussian is convolved on each scan line to estimate the component of the intensity gradient along the scaleline.



The edges in the image can been identified using the Gaussian derivative. This derivative is relatively cheap in terms of computations and is based on degrading the image with a small, separable, and integer valued filter in vertical and horizontal direction.
The below derivative is the Gaussian derivative kernel that is being used
[-3 -5 0 5 3] * A

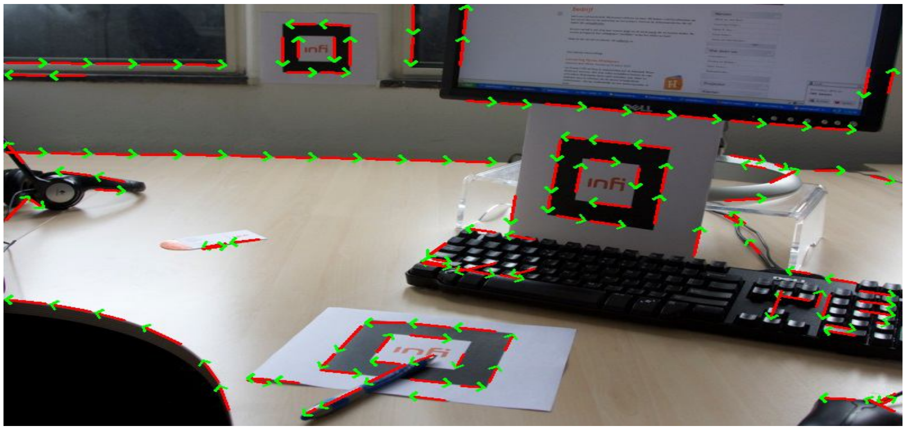
If an edgel is identified, the Sobel operator is used to resolve the orientation of the edgel.The sobel operator is given by:

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

The orientation found is then saved with the edgel. In the above figure, the blue ones are vertical orientated edgels and the green dots are horizontal orientated edgels.

Step 3: Find segments in regions

After the detection of the edgels, a RANSAC-grouper algorithm is used to construct line segments in each region .



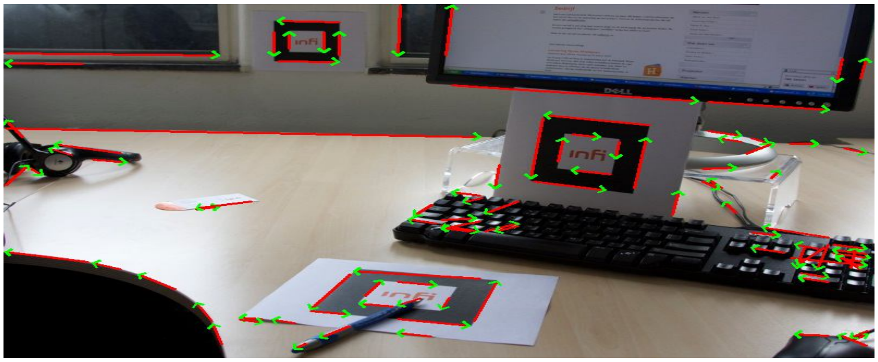
RANSAC is an abbreviation for “RANdomSAmple Consensus”. It’s an algorithm to find groups of “inliers”, i.e. points which can be fitted into a line.

The RANSAC algorithm in short:

1. Randomly 2 points are to be choosen from the same region, whose orientations are suitable with the line joining the two points, to hypothesize a line.
2. The number of points supporting each line is counted. To be considered part of a line a point must lie close to it and have a compatible orientation with the line.
3. Steps 1 and 2 are repeated upto25 times before a segment is found.
4. Lines with enough support ie at least 4 segments are considered as detected lines. Steps 1 to 3 are repeated until all such lines have been found.

As we know the orientation of the edgels from step 2, we can give the line segments an orientation. In the above image, the constructed segments are displayed as green and red arrows.

Step 4: Merge segments to lines
The segments are merged to lines in this step



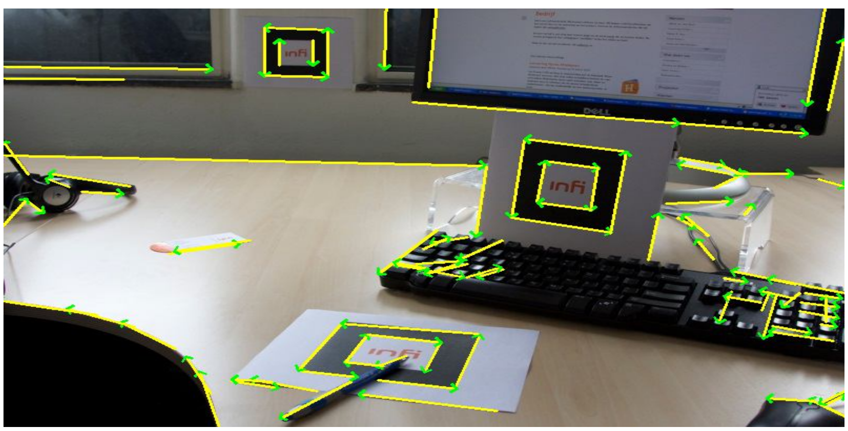
Two segments are merged if:

- The orientations are compatible.
- The segments are close to each other.
- If the pixels along the line between the two segments are on an edge, using the same edgel detection criteria.

In this algorithm all the segments are tested in their own region. When all possible segments are merged, the same operation is repeated for all segments in the whole image.

In the image above the merged lines are displayed as green/red arrows.

Step 5: Extend lines along edges
In this step the lines are extended along the edges.

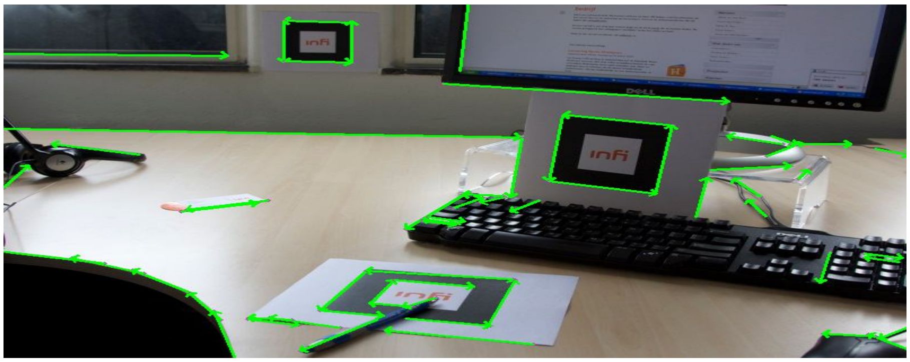


Because we scan only pixels on scan lines 5 pixels apart, it's inevitably that the merged lines don't fit on the entire length of the edge of a marker. So in this step we extend the detected lines pixel by pixel until we detect a corner:

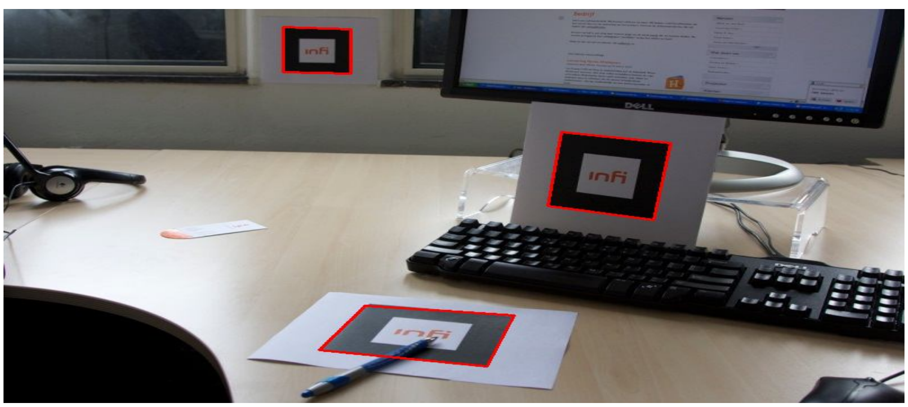
1. Extend each end of a line and check if each pixel is on an edge, using the same edgel detection criteria as used in step 2. Do this until we hit a pixel that's not an edge.

2. Take a pixel a couple of pixels further. If this pixel is 'white', we might have a found a corner of a black on white marker.

The extended lines are displayed as yellow arrows in the image above. In step 6 and 7 only lines with at least one corner are used.
Step 6: Keep lines with corners
Keep the lines with corners.



Step 7: Find markers
Finally, we try to find all markers in the image.



To detect the markers in the image, we try to find chains of 3 to 4 lines. A chain is a list of lines where the end of one of the lines hits the start of another.

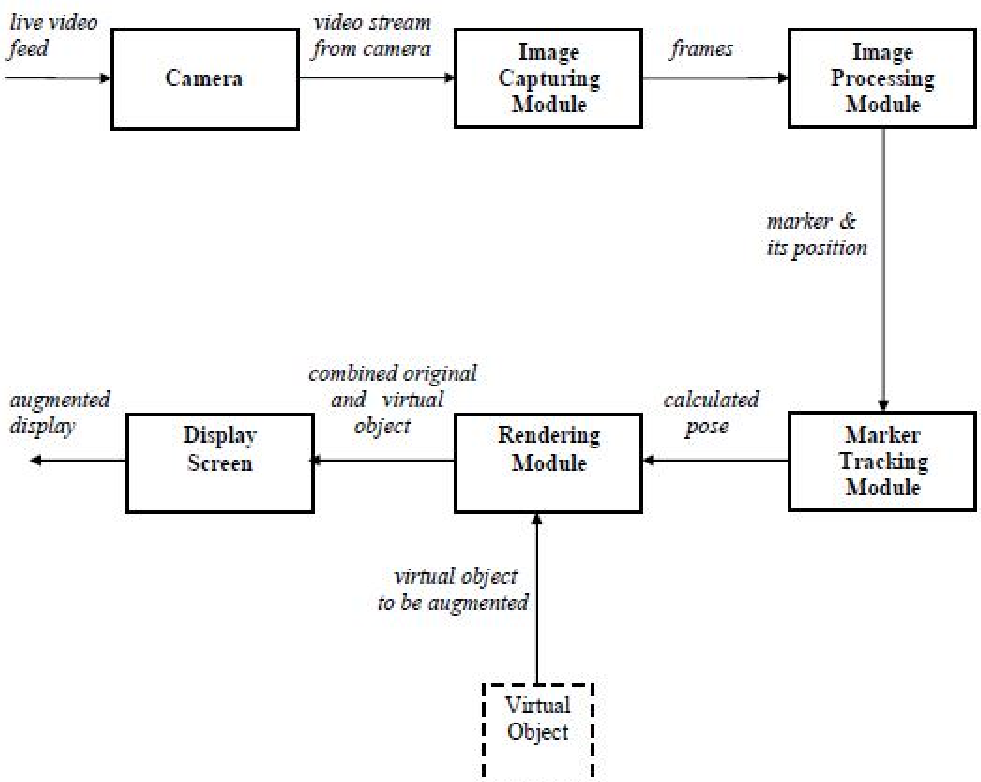
After finding such chains, we only want to keep chains which form a rectangle counter clock wise. These chains will have black 'inside' and are probably markers. Knowing the orientation of the lines, a cross product for each two successive line segments in a chain is used to check this condition.

After finding all markers, for each marker the 4 intersections of the lines in the chain are considered to be the corners. Calculating the positions of corners by these line-intersections gives a robust algorithm. Even if only 3 lines are detected and/or a corner is occluded, the marker will be correctly detected most of the time.

All detected markers are displayed as red rectangles.

III. PROPOSED MARKER BASED SYSTEM ARCHITECTURE

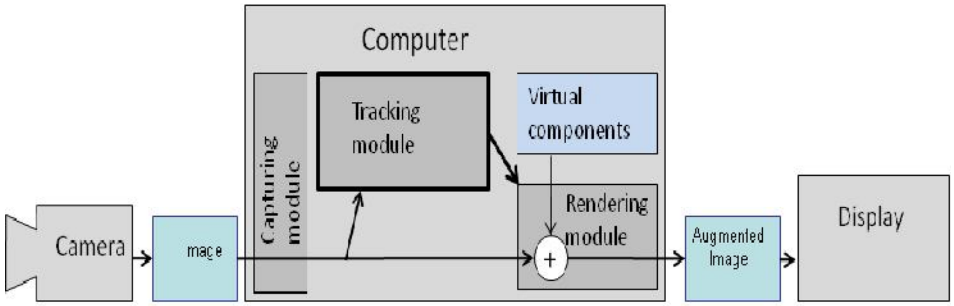
The system proposed in this paper is a marker based system and its architecture is shown in the figure below that contains the following modules. - Camera Module - Capturing Image Module - Processing Image Module - Rendering Module - Tracking Marker Module - Display the screen Flowchart for Simple AR System



Architecture Block diagram

The proposed system is a marker based system and its architecture as shown in figure 6 contains following modules.

a. Camera
b. Image Capturing Module
c. Image Processing Module
d. Rendering Module
e. Display Screen

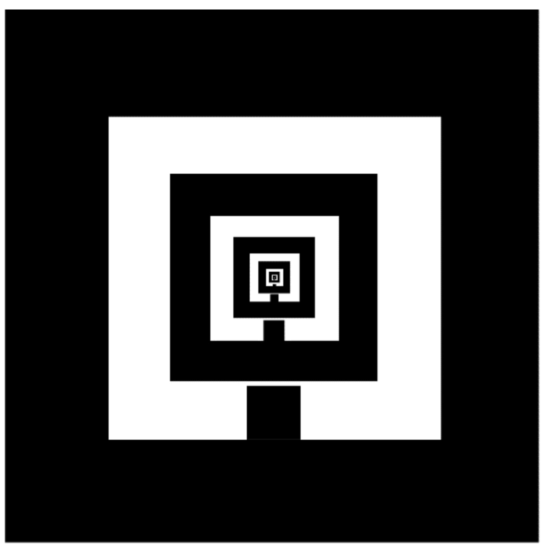


Flow Chart for System Architecture

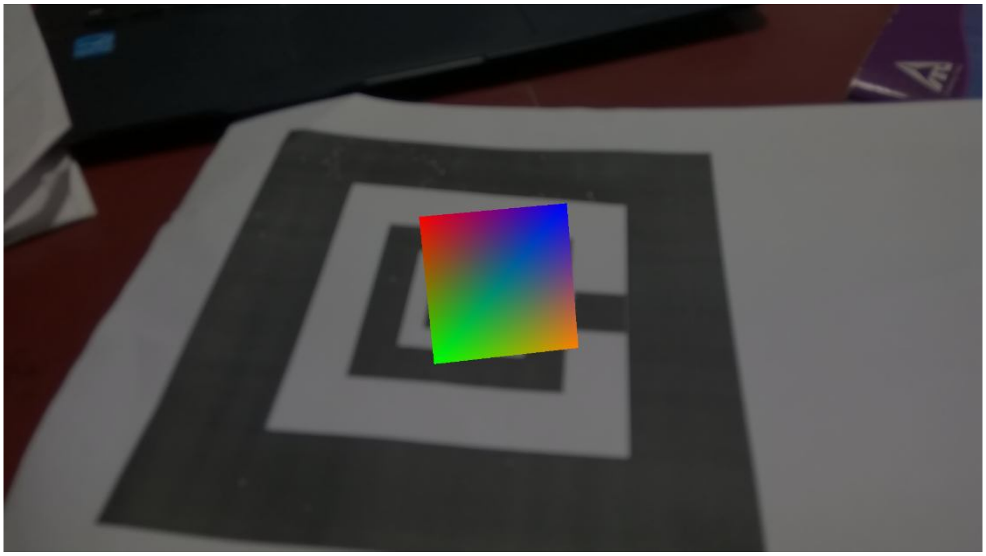
IV. EXPERIMENT AND RESULT

The test set for this evaluation experiment augmented reality is randomly selected from the internet. AR Toolkit and Android Studio software platform is use to perform the experiment. The PC for experiment is equipped with an Intel P4 2.4GHz Personal laptop and 2GB memory.

The proposed scheme is tested using a marker image. From this simulation of the experiment results, we can draw to the conclusion that via the help of the camera the marker image is scanned and the image is displayed on it.



The Cube Marker



3D Image of The Cube

V.CONCLUSION

This paper proposes a marker based augmented reality method through which will aide to combine virtual objects with the real world environment as proposed in this paper. This project will tend to be very much useful in visually understanding certain learning concepts visually. The proposed goal is the use of low cost devices as compared to the advanced AR devices.

REFERENCES

[1] B. Corona, M. Nakano, H. Pérez, “Adaptive Watermarking Algorithm for Binary Image Watermarks”, Lecture Notes in Computer Science,Springer, pp. 207-215, 2004.

[2] A. A. Reddy and B. N. Chatterji, "A new wavelet based logo-watermarking scheme," Pattern Recognition Letters, vol. 26, pp. 1019-1027, 2005.

[3] P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, "Robust spatial watermarking technique for colour images via direct saturation adjustment," Vision, Image and Signal Processing, IEE Proceedings, vol. 152, pp. 561-574, 2005.

[4] F. Gonzalez and J. Hernandez, " A tutorial on Digital Watermarking ", In IEEE annual Carnahan conference on security technology, Spain, 1999.

[5] Cornie and Reinder : Marker Dectection for AR applications.

[6] http://mobile.tutsplus.com/tutorials/augmented-reality

[7] http://mashable.com/follow/topics/augmented-reality