ijecs open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 6 June, 2014 Page No. 6610-6614

Tracking Algorithm for Augmented Reality System

Mayank Pandey, Dr. Manoj Wadhwa , Ms. Prabha Nair
Research Scholar
M.Tech(CSE), Galgotias University, Greater Noida
Professor(HOD)
CSE & IT, Echelon Institute of Technology,Faridabad
Asst. Professor
CSE,Galgotias University,Greater Noida

Abstract: Augmented Reality(AR) aims at merging the real and the virtual in order to enrich a real environment with virtual information. Augmentations range from simple text annotations accompanying real objects to virtual environment of real life objects inserted into a real environment. In the latter case the ultimate goal is to make it impossible to differentiate between real and virtual objects. Several problems need to be overcome before realizing this goal. Amongst them are the rigid registration of virtual objects into the real environment. The key technology of Augmented reality includes displaying, registration and tracking, interactive, etc, among them , registration and tracking is most important technology. In this paper, we will demonstrate the algorithm for tracking a object from a real world and augmenting it on a virtual environment.

Keywords: Augmented Reality, Tracking, Hue Saturation Value, Binary Motion Map, Mean shift.

INTRODUCTION

Accurate registration of virtual objects into a real environment is an outspoken problem in Augmented Reality(AR). This problem needs to be solved regardless of the complexity of the virtual objects one wishes to enhance the real environment with. Both simple text annotations and complex virtual mimics of real-life objects need to be placed rigidly into the real environment. Augmented Reality Systems that lack this requirement will demonstrate serious 'jittering' of virtual objects in the real environment and will therefore fail to give the user a real-life impression of the augmented outcome.

BACKGROUND AND PREVIOUS WORKS

In recent years, object tracking has become one of the most active research areas in computer vision field, and a lot of different object tracking algorithms and their related
applications have been proposed.

Mean shift algorithm with quickly matching mode is widely used in object tracking on the basis of the estimation theory of nonparametric kernel probability density [1].

In 1975, the mean shift algorithm was first proposed by Fukunaga et al [2] and applied to pattern recognition.

In 1995, Cheng [3] introduced it to the field of computer vision.

In 1999, the mean shift algorithm was applied by Comaniciu [4] to object tracking. But the mean shift algorithm is not able to update the object model in the process of tracking, which will result in inaccurate scale locating and even object losing while object's scale varies obviously.

Camshift (continuous adaptive of mean shift) algorithm is proposed by Bradski [5] in order to solve such problems.

This algorithm can automatically adjust the window size to fit the size of object changes in the image. It is effective to resolve the problem of inaccurate object tracking due to the deformation of moving object.

In [6], Sum and Liu solve the object losing problem due to the acceleration of object by using Camshift algorithm and adaptive expanding the search window. However, the track process of Camshift algorithm is easily affected by the object of similar color because it establishes the object histogram model based on the color information. So the effect of algorithm is not satisfactory when the color of object is similar to the background.

FLOW OF AUGMENTED REALITY

Flow of Augmented reality is divided in following four steps[7]:

- A real world information is obtained through input devices.
- Use image recognition technology to analyze the real world and camera position information.
- Generate a virtual model with graph system.
- Finally integrate virtual model into video display on terminal display.

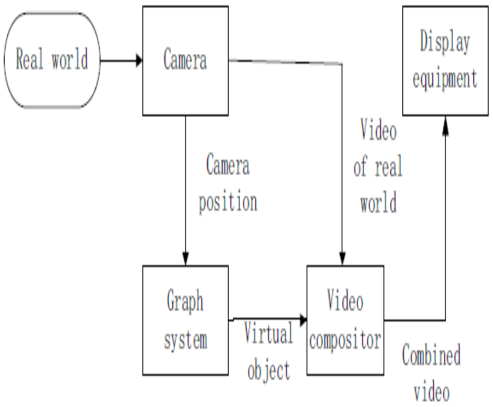


Figure 1: Workflow of Augmented Reality

I. LAYOUT OF PROPOSED ALGORITHM

The layout of the proposed solution is as shown in figure2. In this firstly we input a camera image as an input image then feature detection technique is used to find out the interest point in an image and then extracted part is mapped as a viewing coordinate of a user with addition of a virtual environment features .Now the output as an augmented image is found a output frame.

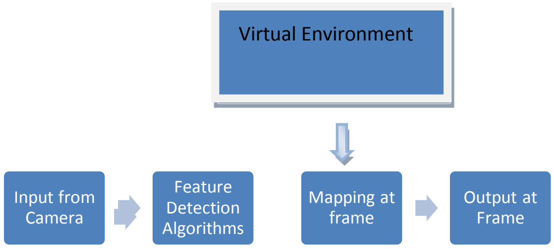


Figure 2: Augmented Reality Layout

PROPOSED ALGORITHM

The principle of proposed algorithm is described as follows. Firstly,we apply several feature detection techniques to an image obtained from a camera now the clear or noiseless image is used to convert the image into color probability distribution according to the color histogram of object. Now, we initialize the size and the position of the search window.Then, the window is adjusted with the obtained result from previous frame adaptively. Finally, we can locate the center of the object in the current image by the result of search.

Algorithm is as follows:

1) **Input**: RGB Image, Threshold T
   Output: Augmented Image
2) **Apply RGB to HSV Conversion():** Color vision can be processed using RGB color space or HSV color space. RGB color space describes colors in terms of the amount of red, green, and blue present. HSV color space describes colors in terms of the Hue, Saturation, and Value. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The HSV model describes colors similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors, where as, HSV describes color using more familiar comparisons such as color, vibrancy and brightness.

The R,G,B values are divided by 255 to change the range from 0..255 to 0..1:

$R' = R/255$

$G' = G/255$

$B' = B/255$

$Cmax = \max(R', G', B')$

$Cmin = \min(R', G', B')$

$\Delta = Cmax - Cmin$

Hue calculation:

$$H = \begin{cases} 60^\circ \times \left(\frac{G'-B'}{\Delta} mod6\right) & , Cmax = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , Cmax = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , Cmax = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , \Delta = 0 \\ \frac{\Delta}{Cmax} & , \Delta <> 0 \end{cases}$$

Value calculation:

V = Cmax

3) Calculate change in image when motion accomplished i.e. I to I'

Abs(D) = | I(x,y,t) – I(x,y,t-1)|

4) If(abs(D)>T)

Binary motion map=0;

Else 1;

5) If(Binary motion map=0)
Probability Image=0;
Else D(x,y,t) \\ Calculated in Step 3

6) Is Image clear i.e Noiseless??

If no apply the filter to clear noise

To remove salt and pepper noise from the corrupted image the below described algorithm is used.

Step 1: A two dimensional window (denoted by 3×3 W) of size 3x3 is selected and centered around the processed pixel p(x, y) in the corrupted image.

Step 2: Sort the pixels in the selected window according to the ascending order and find the median pixel value denoted by Pmed), maximum pixel value (Pmax) and minimum pixel value (Pmin) of the sorted vector V0. Now the first and last elements of the vector V0 is the Pmin and Pmax respectively and the middle element of the vector is the Pmed.

Step 3: If the processed pixel is within the range Pmin < P(x, y) < Pmax , Pmin > 0 and Pmax < 255,it is classified as uncorrupted pixel and it is left unchanged. Otherwise p(x, y)is classified as corrupted pixel.

Step 4: If p(x, y) is corrupted pixel, then we have the following two cases:

Case 1: If Pmin < Pmed < Pmax and 0 < Pmed < 255,replace the corrupted pixel p(x, y) with Pmed

Case 2: If the condition in case 1 is not satisfied then Pmed is a noisy pixel. In this case compute the difference between each pair of adjacent pixel across the sorted vector V0 and obtain the difference vector VD .Then find the maximum difference in the VD and mark its corresponding pixel in the V0 to the processed pixel.

Step 5: Step 1 to step 4 are repeated until the processing is completed for the entire image.

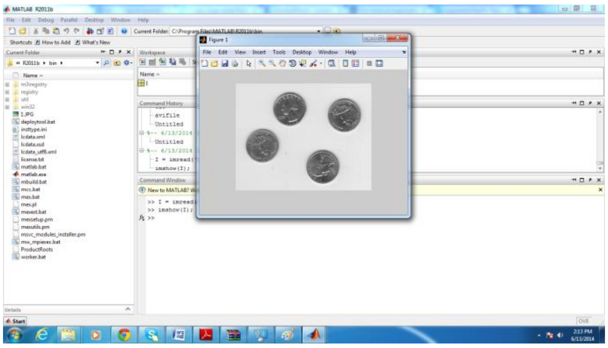
7) Final Probability Image= 0 if Binary motion map=0
Else α* Probabity_Image +(1-α)Color_Probability_Image

8) Combine Probability Image and the virtual environment using the feature detection technique.

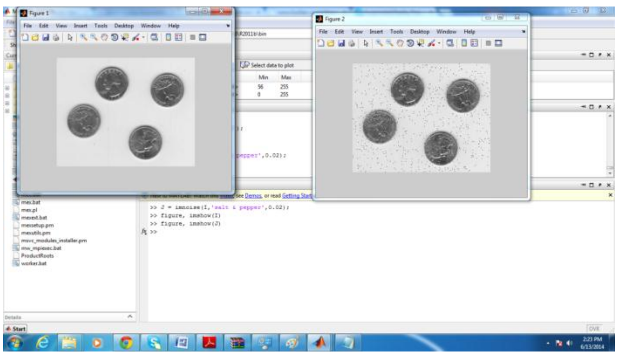
Thus, an augmented image is obtained.

EXPERIMENTAL RESULTS

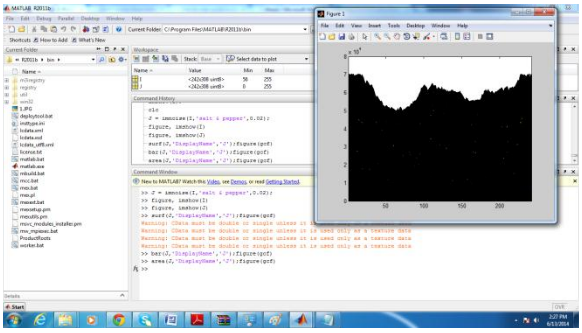
Input an image I to the the MATLAB software on which we have to perform an operations:



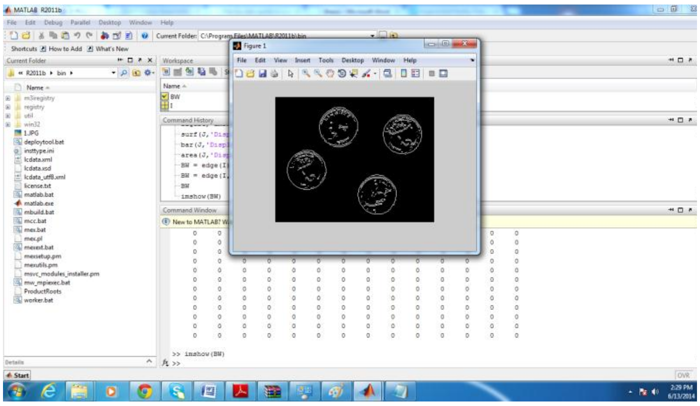
Now after adding the salt and pepper noise to a image we get a output as:



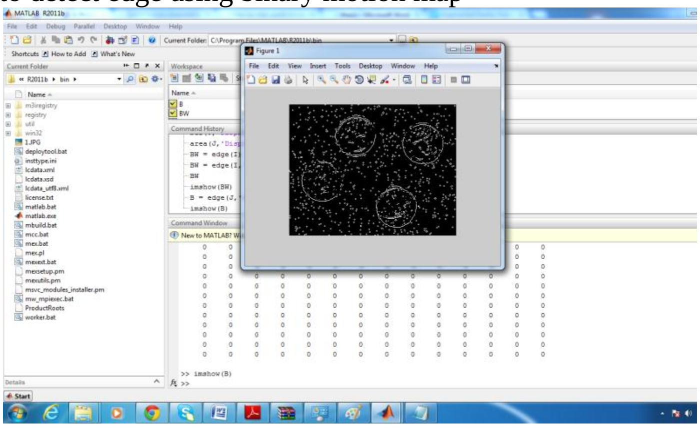
Now using surf, bar and area operation on a noisy image



Now using a sobel operator on input image I to detect the edge on a binary motion map

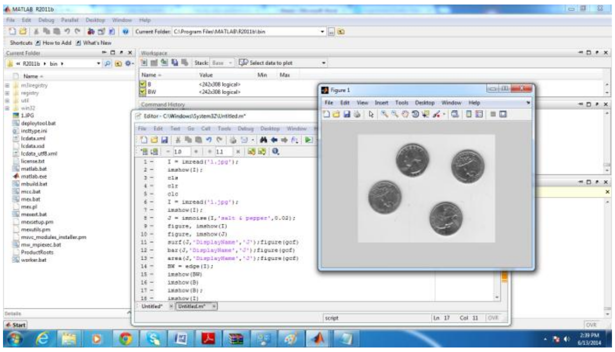


Now using sobel operator on a noisy image J to detect edge using binary motion map

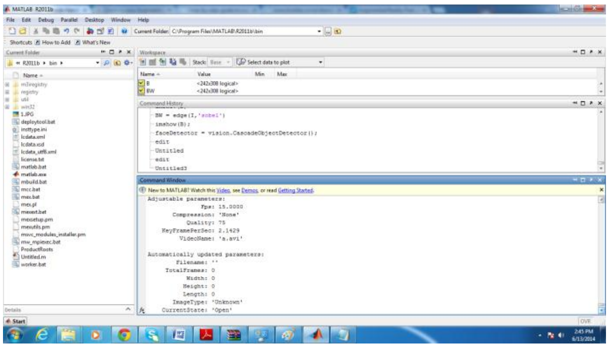


Now after all feature detection technique we

obtain a RGB image



Now compiling and running the algorithm to find the augmented result we obtain a augmented image and and the details of the video used in an experiment



CONCLUSION

A new approach is proposed to resolve several problems of augmentation technique of real image to virtual environment. Problem is having low robustness against images and having low robustness against distortions of images. An intelligent algorithm will try to create accurate alignment of an image, noiseless image is obtained. Using the proposed genetic algorithm, real image could be embedded into a virtual environment , and several issues such as environment issue, color, environment color, image resolution and filtering color correctness and contract were resolved. In this paper, we have demonstrated the algorithm for tracking a object from a real world and augmenting it on a virtual environment.

REFERENCES

[1] Comaniciu D, Ramesh V, Meer P, “Real-time tracking of nongrid objects using mean shift” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition[C], 2000, PP:142-149.

[2] Fukunaga K; Hostetler L D, “The estimation of the gradient of a density function, with applications in pattern recognition”, IEEE Transaction on Information Theory[J], 1975, PP:32–40

[3] Cheng Yi-zong, “Mean shift, mode seeking, and clustering”, IEEE Transaction Pattern Analysis and Machine Intelligence[J], 1995,
PP:790-799.

[4] Comaniciu D, Meer P., “Mean shift analysis and applications”, IEEE International Conference on Computer Vision[C], 1999, PP:1197-1203.

[5] Bradski R, “Computer vision face tracking as a component of a perceptual user interface.” ,Proceedings of IEEE Workshop Applications of Computer Vision[J],1998, PP:214-219.

[6] Sum Kai, Liu Shi-rong, “Combined algorithm with modified Camshift and Kalman filter for multi object tracking”, Information and
Control[J], 2009, PP:9-14.

[7] Ruobing Yang, “The Study and Improvement of Augmented Reality Based on Feature Matching”, Media School, Linyi University,China,vol. 11, pp.586-589, 2011