Visual art recognition using image processing

P. Mega[1], M. Nausathbanu[2], R. Sandhiya[3], Dr.C.S. Manikanda Babu[4]

[1]BE, Department of ECE, Sri Ramakrishna Engineering College, Coimbatore

[2]BE, Department of ECE, Sri Ramakrishna Engineering College, Coimbatore

[3]BE, Department of ECE, Sri Ramakrishna Engineering College, Coimbatore

[4]Associate Professor, Department of ECE, Sri Ramakrishna Engineering College, Coimbatore

Abstract: Gesture Recognition may be a technology that is employed to spot human gestures with the assistance of mathematical algorithms. Gesture recognition acknowledges the hand, tracks the hand movements & conjointly provides data concerning hand position orientation and flux of the fingers. the colour markers square measure placed at the tip of the user fingers. This helps the digital camera to spot the movement of hand and therefore the gesture recognition. The drawing application permits the user you to draw on any surface by following the tip movements of the user's finger. the photographs that square measure drawn by the user are often keep and replaced on the other surface. The user may also shuffle through numerous footage and drawing by victimization the hand gesture movements.

Keywords: Raspberry Pi, Python, Open CV.

I. OVERVIEW The project in the main focuses on the idea to implement the article detection and trailing supported its colour, that may be a visual primarily based project i.e., the input to the project are going to be the video/image information that is endlessly captured with the assistance of a digital camera that is interfaced to the Raspberry Pi. it'll discover the article and it tracks that object by moving the camera within the direction of the detected object. The visual information captured by the digital camera is processed within the Raspberry Pi and also the object is detected supported the colour or form and if the article is detected, the servo motor is turned in such how that where the article moves, the camera are going to be inform thereto object. Here, the servos are controlled by the assistance of a Microcontroller board known as Arduino board through its PWM pins. we will management the angle of servo rotations by the Arduino board i.e., by varied the heart beat widths. The objective is to discover associate degree object supported colour and also the create use of open supply hardware, thus Raspberry Pi processor board is that the most suitable choice for a private curious about low value Arm processor. it's several intrinsic options and lots of ports that makes the accustomed expertise the ability of employing a processor. The board comes with USB ports to that Camera, keyboard and mouse, Wi-Fi electronic device are often connected which supplies the sensation of performing on a system.

II. INTRODUCTION

With the rapid development of computer technology, contemporary human-computer interaction (HCI) devices/ techniques have become indispensable in individuals' daily lives. HCI devices/techniques have also dramatically altered our living habits with computers, consumer electronics, and mobile devices. The ease with which an HCI device or technique can be understood and operated by users has become one of the major considerations when selecting such a device. Therefore, it is necessary for researchers to develop advanced and user-friendly HCI technologies which are able to effortlessly translate users' intentions into corresponding commands without requiring users to learn or accommodate to commands without requiring users to learn or accommodate to the device. Technologies are being developed which are able to intuitively express users' intentions, such as handwriting, gestures, and human body language, to naturally control HCI devices. These technologies have many applications in the fields of remote control, virtual reality, sign language, signature authentication, sport science, health care, and medical rehabilitation.

Current trends in human machine interface have developed quite rapid in consumer devices such as multi touch of iPhone, motion sensing devices of Wi-Fi, and etc. However, the hand tracking systems as user input have been limited due to the technical constraints and prices. In this paper, we introduce a hand motion capture system using a single camera that enable to track 2D hand pose and interact with applications in real-time. Our goal is aimed to design the system that uses a consumer grade webcam as a low-cost approach while having to deal with the robustness, precision, and real-time constraints. There are many research try to solve the same problem. A webcam generally sits on top of the computer screen looking down towards the marker of user's hands. Our related work is presented in the following sections: system overview, design, evaluation, and future work as shown in Fig.1.

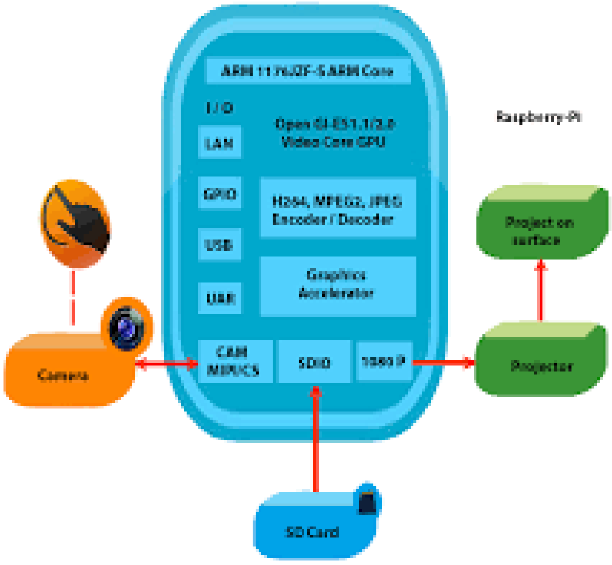


Fig.1. Block diagram.

III. RELATED WORK A. Trackers Using Colour Information

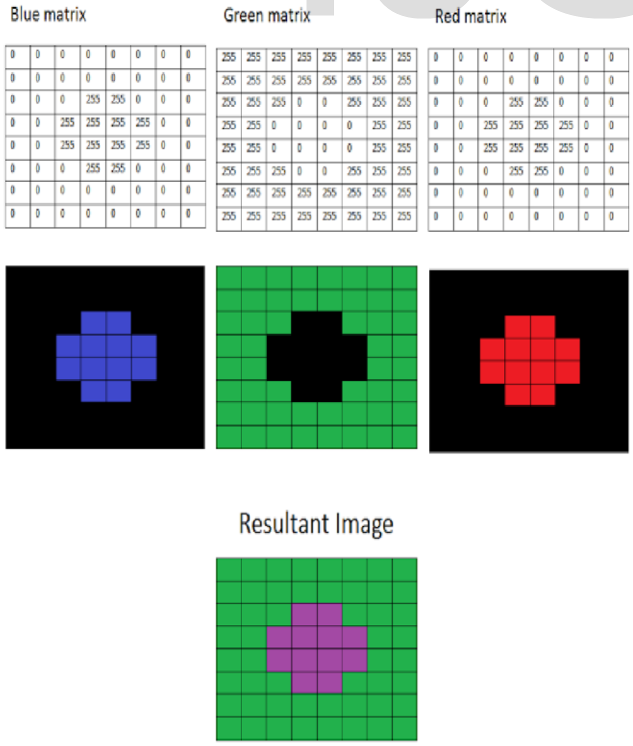
As a fundamental problem in vision, visual tracking has been drawing research attention for decades. A comprehensive review of the subject will be found. Since our focus is on integration colour data in pursuit, we have a tendency to review solely previous colour trackers because of house limitation. Table I lists the abbreviations of trackers mentioned during this paper. A notable early work on colour pursuit is that the colour particle filter introduced, that calculates the probability of every particle by scrutiny its color bar chart from the HSV colour house with the reference color model. within the target model and target candidates square measure described by smoothened color histograms measure from the RGB color house, and mean shift is employed to reduce the gap between the distinct distributions of the target model and target candidates. In RGB color distribution was accustomed describe the target model and candidates, and also the target object was set by minimizing the Kullback Leibler distance between the colour distributions of the target model and candidates with the assistance of a trust-region technique. VTD integrates basic trackers derived from the mixture of various basic observation and motion models, and 4 basic observation models, that use hue, saturation, intensity and edge templates as options severally, square measure adopted. ton measures the similarity between a candidate and also the target mistreatment domestically order less matching, and HSV color house is employed to explain the looks of every pixel.

MEEM uses options extracted within the research laboratory color house within the most up-to-date work, CSK is extended with color names, and to hurry up, the dimension of the initial color names is reduced with Associate in Nursing adaptation spatiality reduction technique. There are trackers that take color input, however don't expressly exploit the employment of color data. Despite previous arts, there's a scarcity of a scientific study and understanding of however color data will be accustomed improve visual pursuit. Our work aims to fill the gap by completely work the behaviour numerous of diverse state of the art visual trackers with various color representations.

B. Color Information in Other Vision Tasks

Not surprisingly, the discriminative power of color information has been systematically investigated for various vision topics, such as object recognition, human action recognition, object detection, etc. While being highly motivated by these pioneering works and borrowing some ideas from them, our work however focuses on visual tracking. To the best of our knowledge, this is the first comprehensive study on encoding color information for visual tracking. In fact, as shown in our experiments, many modern grayscale trackers, when augmented with color information, outperform previously proposed color trackers.

Working/ Design Theory: OpenCV usually captures images and videos in 8-bit, unsigned integer, BGR format. In other words, captured images can be considered as 3 matrices; BLUE, GREEN and RED (hence the name BGR) with integer values ranges from 0 to 255. The following image shows how a color image is represented using 3 matrices as shown in Fig.2.



Blue matrix

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	255	255	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	255	255	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Green matrix

255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	0	0	255	255	255
255	255	0	0	0	0	255	255
255	255	0	0	0	0	255	255
255	255	255	0	0	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Red matrix

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	255	255	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	255	255	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig.2.Color Image Represented in matrices.

In the higher than image, every tiny box represents a pixel of the image. In real pictures, these pixels square measure thus tiny that human eye cannot differentiate. Usually, one will assume that BGR color area is a lot of appropriate for color primarily based segmentation. however, HSV color area is that the most fitted color area for color primarily based image segmentation. So, within the higher than application, I even have reborn the colour area of original image of the video from BGR to HSV image. HSV color area is additionally consists of three matrices, HUE, SATURATION and worth. In OpenCV, worth vary for HUE, SATURATION and worth square measure severally 0-179, 0-255 and 0-255. HUE represents the colour, SATURATION represents quantity the number to that that various color is mixed with white and worth represents the amount to that that various color is mixed with black. In the higher than application, I even have thought of that the red object has HUE, SATURATION and worth in between 170-180, 160-255, 60-255 severally. Here the HUE is exclusive for that specific color distribution of that object however SATURATION and worth could also be vary per the lighting condition of that atmosphere. Hue values of basic colours:

- Orange 0-22
- Yellow 22- 38
- Green 38-75
- Blue 75-130
- Violet 130-160
- Red 160-179

These square measure approximate values. you have got to seek out the precise vary of HUE values consistent with the colour of the article. I found that the vary of 170-179 is ideal for the vary of hue values of my object. The SATURATION and price is depend upon the lighting condition of the surroundings yet because the surface of the article. After thresholding the image, you will see cabbage butterfly isolated objects here and there. it should be owing to noises within the image or the particular little objects that have a similar color as our main object. These unnecessary cabbage butterfly patches is eliminated by applying morphological gap. Morphological gap is achieved by a erosion, followed by the dilation with a similar structuring component. Threshold image might also have cabbage butterfly holes within the main objects here and there. it should be owing to noises within the image. These unnecessary little holes within the main object is eliminated by applying morphological closing. Morphological closing is achieved by a dilation, followed by the erosion with a similar structuring component.

void inRange(InputArray src, InputArray lowerb, InputArray upperb, OutputArray dst);

Checks that every component of 'src' lies between 'lowerb' and 'upperb'. If so, that several location of 'dst' is allotted '255' , otherwise '0'. (Pixels with price 255 is shown as white whereas pixels with price zero is shown as black) Arguments –

- **InputArray src** - Source image
- **InputArray lowerb** - inclusive lower boundary (If lowerb=Scalar(x, y, z), pixels that have values
- below x, y and z for HUE, SATURATION and price severally is taken into account as black pixels in dst image)
- **InputArray higherb** - Exclusive upper boundary (If it's upperb=Scalar(x, y, z), pixels that have values larger or equal than x, y and z for HUE, SATURATION and price severally is taken into account as black pixels in dst image)
- **OutputArray dst** - Destination image (should have identical size because the src image and may be 8-bit unsigned number, CV_8U)

void erode(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1,intborderType=BORDER_CONSTANT,constScalar&borderValue=morphologyDefaultBorderValue())

Arguments –

- **InputArray src** - supply image

This perform erode the provision image and stores the result in the destination image. In-place method is supported. (which suggests that you're going to be ready to use a similar variable for the provision and destination image). If the provision image is multi-channel, all channels square measure processed severally and so the result is hold on at intervals the destination image as separate channels. Arguments –

- **InputArray src** - supply image
- **OutputArray dst** - Destination image (should have an equivalent size and sort because the supply image)
- **InputArray kernel** - Structuring component that is employed to erode the supply image
- **Point anchor** - Position of the anchor among the kernel. If it's Point(-1, -1), the middle of the kernel is taken because the position of anchor
- **int iterations** - variety of times erosion is applied
- **int borderType** - picture element extrapolation methodology during a stipulation
- const Scalar& borderValue – Value of the pixels during a stipulation if borderType = BORDER_ CONSTANT

void dilate(InputArray src, OutputArray dst, InputArray kernel, Point anchor=Point(-1,-1), int iterations=1,intborderType=BORDER_CONSTANT,constScalar&borderValue=morphologyDefaultBorderValue()) This operate dilates the supply image and stores the lead to the destination image. In-place process is supported. (which means that you'll be able to use an equivalent variable for the supply and destination image). If the supply image is multi-channel, all channels area unit processed severally and also the result's keep within the destination image as separate channels.

- **OutputArray dst** - Destination image (should have an equivalent size and sort because the supply image)

- **InputArray kernel** - Structuring component that is employed to erode the supply image
- **Point anchor** - Position of the anchor among the kernel. If it's Point(-1, -1), the middle of the kernel is taken because the position of anchor
- **int iterations** - variety of times erosion is applied
- **int borderType** - picture element extrapolation methodology during a stipulation
- const Scalar& borderValue – Value of the pixels during a stipulation if borderType = BORDER_ CONSTANT

void cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0)

This perform convert a supply image from one color area to a different. In-place process is supported. (which suggests that you'll be able to use a similar variable for the supply and destination image)

- InputArray src - supply image
- OutputArray dst - Destination image (should have a similar size and therefore the depth because the supply image)
- int code - Color area conversion code (e.g - COLOR_BGR2HSV, COLOR_RGB2HSV, COLOR_ BGR2GRAY,COLOR_BGR2YCrCb,COLOR_BGR2BGRA, etc)
- int dstCn - variety of channels within the destination image. If it is 0, variety of channels comes mechanically from the supply image and therefore the color conversion code.

IV. RESULTS

Results of this paper is as shown in bellow Figs.3 to 6.

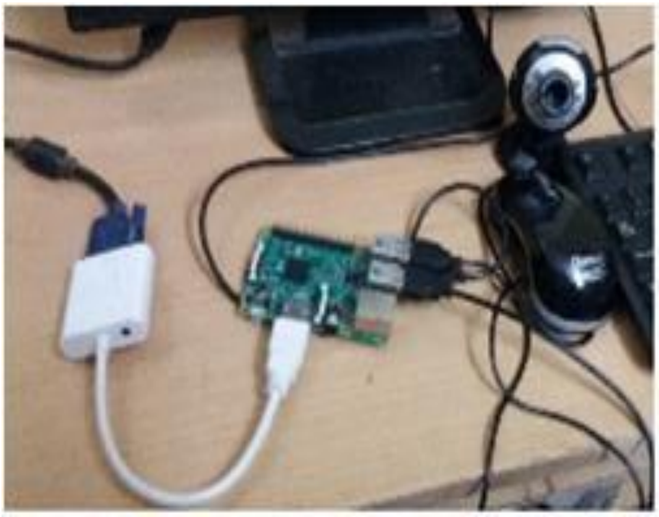


Fig.3. Interfacing PC to Raspberry Pi board and camera.

Fig.4. Raspberry Pi window editor.

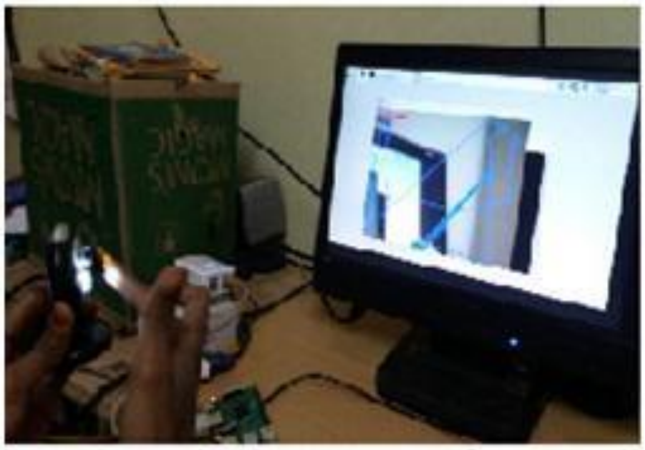


Fig.5. object captured through camera and selected color for drawing.

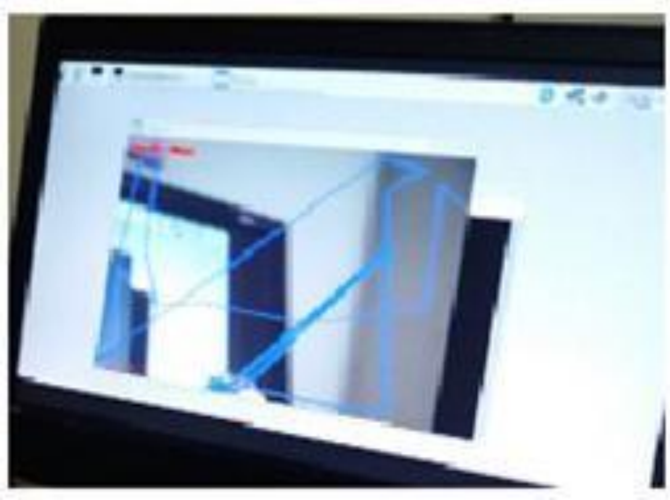


Fig.6. displayed output according to hand gesture drawing pattern.

V. CONCLUSION

We designed and build a model that may find the item of such as color which works on the idea of visual knowledge captured from a typical digital camera that encompasses a honest clarity. The rule is tested within the laboratory live and therefore the success rate is one hundred. The rule works well beneath all conditions and therefore the time taken to find and track.

VI. REFERENCES:

[1] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," Pattern Recognit., vol. 43, no. 10, pp. 3605–3620, 2010. [2] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," IEEE Trans. Signal Process., vol. 59, no. 12, pp. 6197–6205, Dec. 2011.

[3] W. C. Bang, W. Chang, K. H. Kang, E. S. Choi, A. Potanin, and D. Y. Kim, "Self-contained spatial input device for wearable computers," in Proc. IEEE Int. Conf. Wearable Comput., Oct. 2003, pp. 26–34. [4] G. Bailador, C. Sanchez-Avila, J. Guerra-Casanova, and A. de Santos Sierra, "Analysis of pattern recognition techniques for in-air signature biometrics," Pattern Recognit., vol. 44, nos. 10–11, pp. 2468–2478, 2011. [5] C. M. N. Brigante, N. Abbate, A. Basile, A. C. Faulisi, and S. Sessa, "Towards miniaturization of a MEMS-based wearable motion capture system," IEEE Trans. Ind. Electron., vol. 58, no. 8, pp. 3234–3241, Aug. 2011. [6] M. J. Caruso, "Application of magnetoresistive sensors in navigation systems," in Proc. SAE, 1997, pp. 15–21. [7] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecificuser hand gesture recognition," IEEE Sensors J., vol. 12, no. 5, pp. 1166–1173, May 2012. [8] Z. Dong, C. Wejinya, and W. J. Li, "An optical-tracking calibration method for MEMS-based digital writing instrument," IEEE Sensors J., vol. 10, no. 10, pp. 1543–1551, Oct. 2010. [9] M. H. Ko, G. West, S. Venkatesh, and M. Kumar, "Using dynamic time warping for online temporal fusion in multisensor systems," Inform. Fusion, vol. 9, no. 3, pp. 370–388, 2010.

[10] Y. S. Kim, B. S. Soh, and S. G. Lee, "A new wearable input device: SCURRY," IEEE Trans. Ind. Electron., vol. 52, no. 6, pp. 1490–1499, Dec. 2005. [11] S. Kallio, J. Kela, P. Korpipää, and J. Mäntyjärvi, "User independent gesture interaction for small handheld devices," Int. J. Pattern Recognit Artif. Intell., vol. 20, no. 4, pp. 505–524, 2006.