Colour Markers Tracking for Human Locomotion Modeling

Jerome Liew Qing Yin\(^1\)*, Prof Koo Voon Chet\(^1\) and Lim Chot Hun\(^1\)

\(^1\)Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia
Tel: +606-2523060; E-mail: jeromeliew89@gmail.com

Abstract - This paper describes a tracking method to detect the location of colour markers wore on the joints of human moving parts. A simple Graphical User Interface is used to obtain the colour profile of the desired target. Similarity map is then calculated to extract the pixels that resemble the targets. Colour seed and flood fill is used to label and obtain properties of the different targets detected. Nearest distance is lastly used to maintain the identity of one target and distinguish from the rest.

Index Terms – difference intensity, colour seed, markers, objects, blobs.

I. INTRODUCTION

In order to model or represent a human motion, say, for the use in the research area of humanoids, for gaming purposes, for medical evaluation etc., a way to record a human motion has to be developed. There are a few options that exist to perform this task and vary in terms of accuracy, equipment, suitable environment etc. Probably the option with the most equipment is to use a suit equipped with Inertial Measurement Unit (IMU). These sensors could be used to derive the acceleration, velocity and the degree of movement of human joints. However, the downside of this option is that it would be a tedious process to get the correct suit size for different individuals. Electromyography is one of the options where electric signals generated during muscle activity can be amplified and further analyzed for medical purpose and classified to control device such as prosthetic body parts, mobile robots etc. [3].

Options that use a number of cameras installed in a controlled environment does not burden the human operator with suits and equipment compared to the two options above. Human motion can be reconstructed from a multiple view camera setup, similar to the effect of a stereo-vision used to derive the 3D image of an object. This method does not require the user to wear anything. The downside of this method is the extensive processing and calculation needed and also the errors in position accuracy. To achieve high accuracy and data rates, a well-known method is used whereby highly reflective markers which emit distinct infrared signature were attached to several points on the experiment subject. Multiple cameras with infrared filters are then used to track the movement of the markers.

Colour markers could be used as potential replacement for highly reflective markers, thus eliminate the need to use infrared pass filter for the cameras. However, colour tracking does have disadvantage compare to infrared point sensing. If the scene of observation of the cameras has pixels or areas that has the same colour profile as the markers, those pixels would be regarded as the markers as well. Under such condition, either a controlled environment is needed or algorithm is to be developed to provide a remedy to this issue. Nevertheless, the advantage of using colour markers remains appealing as it ease the effort to classify the points and differentiate body parts, thus provide good analysis and little processing requirement.

II. IMAGE DATA

A. OpenCV Library and Iplimage Structure
“OpenCV (Open Source Computer Vision) is a library released under a BSD license; it is free for both academic and commercial use. It has C++, C, Python and Java interfaces running on Windows, Linux, Android and Mac. The library has >2500 optimized algorithms. It is used around the world, has >2.5M downloads and >40K people in the user group. Uses range from interactive art, to mine inspection, stitching maps on the web on through advanced robotics.” [1] In C programming, OpenCV save image in the structure of Iplimage (inherited from Intel Image Processing Library).

B. Image Type and Colour Space

Images can come in different types which vary in the number of Channels and the representation of the pixels. For example, a gray-scale image is a single layer image where each pixel has value range from 0 to 255. 0 represents the colour black, 255 represents the colour white, while the values between 0 and 255 represent different intensity of gray.

An RGB image is a 3 layer colour image, where each pixel has 3 channels. Each channel has value range from 0 to 255. The 3 different values in each channel represent the intensity of Blue, Green and Red of a pixel. Different objects of different colour appear differently because of the different combinations of RGB values. We can then establish a colour profile for the sample object of interest. This colour profile can then be compared with other pixels in the image to generate a 2D map of difference intensity which later can be processed to extract the locations of the objects which has nearer colour profile to that of the sample.

III. PROCESSING METHOD

Taking a blank white colour image and minus the resulting map generated, we can obtain the inverse of the resulting map as plotted in a 3D space as shown below, with the z-axis represents the magnitude of difference.

![Fig.1. (a) Colour profile selected: green ball, (b) Difference Intensity map](image)

Fig.1. (a) Colour profile selected: green ball, (b) Difference Intensity map

![Fig.2. Map of difference intensity plotted in MATLAB using the function 'surf' (Top View)](image)

Fig.2. Map of difference intensity plotted in MATLAB using the function 'surf' (Top View)

![Fig.3. Map of difference intensity plotted in MATLAB using the function 'surf' (angled View). Note the part that is red in colour is the highest which is the location where the ball is.](image)

Fig.3. Map of difference intensity plotted in MATLAB using the function 'surf' (angled View). Note the part that is red in colour is the highest which is the location where the ball is.

A. Adaptive Threshold

As the result shown, the pixels with the nearest similarity will have the highest value (near white). However, due to inconsistent lighting, even a single colour object will not have consistent colour across its whole body. Thus, a threshold is needed to set a range of colour values which we would classify as a part of the object of interest. However, fixing a stagnant threshold would result in the sudden appearance and

IJMOT-2014-1-554 © 2014 IAMOT
disappearance of the object of interest as presented in a simple illustration below.

![Fig.4](image)

Fig.4. Objects that are detected above the green colour line threshold could appear for a while in (a) but disappear in the next frame (b).

Thus, an adaptive threshold is needed to maintain the tracking consistency. The approach here is to set the threshold at a value away from the maximum point (values of the nearest pixels to the sample colour profile) of the result map. Thus, as the maximum value changes, the threshold would be adjusted accordingly as well. We will consider pixels that hold value between the maximum and the threshold be considered as part of the object. The result can be shown in a binary image (black and white image) as shown below.

![Fig.5](image)

Fig.5. (a) Original image (b) Difference intensity map

**B. Erosion and Dilation**

The result obtained after the previous procedure could still ends up having lots of pepper noises which could appear as false input. Probably the simplest method to filter these out is to use the method erosion, where a square kernel (smallest 3x3) with its anchor at the middle point, is used to sweep across the whole image and the pixels under that kernel is computed for the minimal pixel value which will then replace the pixel at the anchor point. By repeating this operation for several times, pepper size white pixels in the resulting binary image would be eliminated while the big ones will remain. Dilation is the reverse of the erosion operation where it results in increasing the size of the white pixels, thus helps in recovering the rough size of the detected objects.

![Fig.6](image)

Fig.6. After adaptive threshold

**C. Blob Colourization and Size Filter**

The next task is to differentiate all these blobs from one another and give identity to each of them. This can be done by putting in a colour seed in one of the white blobs, and then colour the whole blobs according to the colour of that seed. We can then increment the value of seed and colour another white blobs with a different intensity of gray. However, there is only a limited amount of colour that we can use here if we were to follow the range of the data type ‘char’ where only the values from 1 to 254 can be used as the colour seed. We can use the area size of the blobs as deciding factor to eliminate the small blobs and reserve the colours for the next seed. A windowed filter could be used to define two thresholds where blobs with size within the window stays while those objects below and above the windows be eliminated. This process would produce a result below:
Fig. 7. After erosion, dilation, 4 huge blobs remained and are colour coded with different intensity of grey.

D. Derive Properties of Blob

From the result above, we could obtain properties such as median, height and width for further processing. In order to acquire the middle point of a blob, the method that is used here is to look for a transition from an empty row to a row which has at least 1 pixel containing the seed’s colour. Next is to look for the opposite of that transition. This will provide a top and bottom border which we can used to compute the height and y-component of the mid-point. The same procedure can be applied to obtain the width and x-component of the mid-point. The area of the blob could be done by summing up all the pixels which has the same colour. The image below shows the result.

Fig. 8. Using different colour to show different objects.

E. Maintain the Identity of each blob

Due to the way images pixels are scanned, the blob that is the nearest to the starting point will always be labeled as the first detected blob. Thus, even though the objects scanned in the scene changes their position, the object nearest to the starting point carry the identity of 1. As an early approach to solve this issue, a method which uses nearest distance as a decision engine is used and the result is shown as below:

Fig. 9: initial detection

Fig. 10: The label of each item still remains after positions of the object are rotated anti-clockwise.

The identity of a detected object could be maintained even though it swaps its position with another object of similar colour.

IV. CONCLUSION

By defining a certain colour profile of our selection, we could device a tracking application that tracks of objects with similar colour. However, the application is yet to run fully on its own mind, GUIs are still needed for an operator to tweak the parameters and settings for optimal result. A few more rules and algorithm needs to be developed to account for occlusion, objects movement out of scene etc. With the current progress, it is possible to track simple movement of a human equipped with the colour markers, for example, walking in a direction perpendicular to the camera view.
ACKNOWLEDGMENT

This work is partially funded by the Centre for Remote Sensing and Surveillance Technology (CRSST), Multimedia University.

REFERENCES