A Robust Visual Object Tracking Approach on a Mobile Device

Abdulmalik Danlami Mohammed and Tim Morris

School of Computer Science, University of Manchester,
Oxford Road, Manchester, UK
Abdulmalik.mohammed@postgrad.manchester.ac.uk,
tim.morris@manchester.ac.uk

Abstract. In this paper, we present an approach for tracking an object in video captured on a mobile device. We use a colour-based approach. The performance of many of these approaches degrades due to lighting changes and occlusion. To address the issue of lightning changes, our approach makes use of colour histogram that is generated by accumulating histograms derived from target objects imaged under different conditions. A CAMShift tracking algorithm is applied to the back-projected image to track the target object.

We have tested our approach by tracking an Emergency Exit sign and the results obtained show that the tracking is robust against lightning changes.

Keywords: CAMShift Algorithm, Histogram Backprojection, Colour Space.

1 Introduction

Object tracking is an important task that is required in many high-level computer vision applications. It is concerned with estimating the trajectory of an object in a given scene [1]. Recent advances in mobile phones technology, in particular, the low-cost, high-resolution camera, has opened a new research direction in object tracking technique.

Object Tracking algorithm have been use in applications such as face and head tracking, video surveillance system, human-computer-interaction, traffic monitoring system, document retrieval system[1].

Many tracking approaches utilise features such as colours, edges and textures to model the object to track. The choice of feature to use for tracking depends largely on how robust it is against challenges such as lighting changes, blurring due to camera motion and occlusion; they are robust to different challenges. Many implementations will therefore use a combination of features that collectively address most, or all of these issues. However, this will involve additional computation to the tracking system which may prevent real-time tracking from being achieved in devices with low processing power.

Most Visual objects contain colour combinations that make them distinct from the surrounding environment. For example, a standard exit sign in the UK is a green

Linawati et al. (Eds.): ICT-EurAsia 2014, LNCS 8407, pp. 190–198, 2014.

rectangular object with some or all of a schematic running man, an open door, an arrow and possibly the word "EXIT". The arrow and running man give an indication of the direction to be taken in the event of emergency. Hence, colour information should be helpful to detect and track sign.

In this paper, we utilise the colour information and the CAMShift tracking algorithm to detect and track sign using a camera phone. We aimed to implement a robust, light weight tracking algorithm on devices with low processing capability.

The contribution of this paper is in the application of CAMShift tracking algorithm based on robust histogram back projection on a mobile device.

The remainder of this paper is organised as follows: In Section 2 we discuss in detail works related to tracking of objects and their suitability for mobile devices. Section 3 provides an overview of a CAMShift tracking algorithm and its limitation in a controlled environment. In Section 4 we present an introduction to histogram backprojection, which is one of the useful steps required in CAMShift tracking algorithm. Section 5 discusses in detail our proposed approach utilising colour, we emphasise the methods we adopted to mitigate the effect of lighting changes. The experimental result of implementing our proposed tracking approach on mobile device is presented in Section 6. Finally, Section 7 concludes.

2 Related Work

In this section, we will discuss in details some of the several tracking approaches that have been proposed in the literature and their suitability for implementing on mobile devices.

Li and Zhi-Chun [3] enhance the CAMShift tracking algorithm by utilising skin-colour and contour information to respectively detect and track human ear. The proposed approach is divided into two stages. The first stage used CAMShift based on skin-color to track a face in the video frame, the second stage used contour information to track and locate the ear.

Tsen Min Chen et al [12] proposed an object tracking method that is based on probability distribution. To deal with issues of lighting changes, the method adjusts the model parameter during run-time.

Alper et al. [4] combine colour and texture features to track object and a shape feature to improve object tracking with CAMShift, especially in occlusion. The fused features (colour and textures) enable tracking in different appearances, even partial.

Yingyin et al [7] proposed a frame-differencing approach for moving object detection and further utilised the hue and edge orientation histogram of the target object to improve the CAMShift tracking algorithm.

Lixin Fan et al [13] proposed a tracking approach on mobile device using a haar-like feature matching to track target objects. They made use of an online update schema to improve tracking.

Jianwei Gong et al [14] present colour-based segmentation method with CAMShift tracking algorithm for detection and tracking of traffic lights using a camera that is mounted on a moving vehicle.

Donghe and Jinson [5] integrate a filtering prediction with a CAMShift tracking algorithm to improve object tracking against occlusion. The algorithm works by locating the face in the first three frames of the video sequence and using the position to initialise the parameters of the filter prediction for finding the location of the face in the following frame.

Gary Bradski [2] presents a face tracking approach in perceptual user interface using hue in the HSV colour system for tracking face and head.

In contrast to [2], David et al [6] make use of accumulated histograms to model same object in different appearances and for monitoring the object identities during tracking. The reference histogram is computed offline. However, the large memory requirement of [3,4,5], [7] makes those approaches unsuitable to be implemented on devices with limited processing capability.

Our approach is motivated by [2],[6] because of their low computational demand. A comprehensive review of object tracking approaches based on different object representation and features selection can be found in [1].

3 CAMShift Tracking Algorithm

In this Section, we provide an overview of the CAMShift tracking algorithm and its shortcomings, in particular when object's appearance changes and when an object is occluded.

Continuously Adaptive Mean Shift (CAMShift) is a tracking algorithm that uses colour features for tracking an object in a scene. This algorithm is an extension of the mean shift algorithm. It finds the mode of a probability distribution using the mean shift algorithm and then iteratively adjusts the parameter of the distribution until convergence. The basic principle of CAMShift is well documented in [2],[8]. Figure 1 shows the object tracking workflow.

The following steps describe the CAMShift algorithm:

(1) Set the initial location and size of the search window.
(2) Calculate the colour probability distribution within the search window.
(3) Find the new location and size of the search window using the mean shift algorithm.
(4) Compute the orientation and scale of the target.
(5) Capture the next frame and repeat from step 3.

The zeroth, first and second order image moments as shown below are used to find the mean location of the target object.

$$\boldsymbol{M_{00}} = \sum_x \sum_y \boldsymbol{I(x,y)}. \tag{1}$$

$$\boldsymbol{M_{10}} = \sum_x \sum_y \boldsymbol{xI(x,y)}; \quad \boldsymbol{M_{01}} = \sum_x \sum_y \boldsymbol{yI(x,y)}; \tag{2}$$

$$M_{20} = \sum_x \sum_y x^2 I(x,y); \quad M_{02} = \sum_x \sum_y y^2 I(x,y); \tag{3}$$

Where $I(x,y)$ is the pixel (probability) value at position (x,y) in the image, and x and y range over the search window. Then the mean location (centroid) in the search window is computed as:

$$x_c = \frac{M_{10}}{M_{00}}; \quad y_c = \frac{M_{01}}{M_{00}}; \tag{4}$$

To update the search window, the target object aspect ratio given below:

$$ratio = \frac{M_{20}}{x_c^2} \Big/ \frac{M_{02}}{y_c^2} \tag{5}$$

is used with:

$$width = 2M_{00}\, .ratio; \quad height = {}^{2M_{00}}/_{ratio} \tag{6}$$

CAMShift performs better given a simple background where lighting changes and object occlusion are absent. However, the algorithm can perform poorly in cases of occlusion by other objects or the presence of objects of similar colouration.

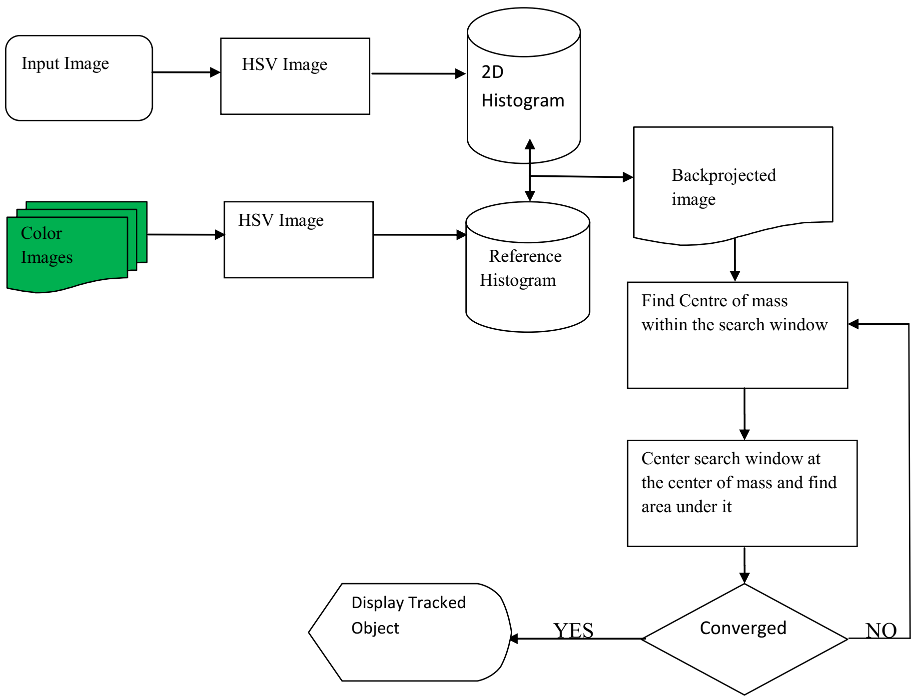

Fig. 1. Object Tracking schema

4 Histogram Backprojection

Histogram back projection is a simple technique proposed by [10] to find object correspondence in a colour image. The technique utilises the colour histogram of the captured image and histogram of the target object as a look-up table to generate a probability distribution image. Initially, a histogram $\boldsymbol{M_j}$ of the target object in some colour space is pre-computed followed by the histogram $\boldsymbol{I_j}$ of the scene where the search for the object is to take place. At this point, a ratio histogram $\boldsymbol{R_j}$ is computed by dividing the $\boldsymbol{M_j}$and $\boldsymbol{I_j}$ and the backprojected image $\boldsymbol{B_{x,y}}$ is generated according to equation (8).

$$\boldsymbol{R_j} = \boldsymbol{min}[\boldsymbol{M_j}/\boldsymbol{I_j},\ \boldsymbol{1}] \tag{7}$$

Where $\boldsymbol{j}$ represents the index of a bin

$$\boldsymbol{B_{x,y}} = \boldsymbol{R_{h(C_{x,y})}} \tag{8}$$

Where $\boldsymbol{C_{x,y}}$ is the colour value at location $\boldsymbol{(x,\ y)}$of the searched image and $\boldsymbol{h}(\boldsymbol{C_{x,y}})$ is the bin corresponding to $\boldsymbol{C_{x,y}}$ [11].

At this point, a blurring mask is applied to the back projected image and the location of the peak value in the image corresponds to the location of the object in the image. In particular, the technique is useful where a large database of the model histogram is required.

5 Propose Approach

We utilised the colour content of an object for tracking since many objects can be identified based on their colour contents. However, to address the issues associated with most colour-based tracking system as explained in section 3, we accumulated histograms of the objects captured under different lighting conditions to model the target object in different views. Our approach is motivated by the work of [6], but in contrast to [6], we implement the tracking algorithm on a mobile device rather than on device with GPU. The approach has low computational requirement since the accumulation of histograms is offline and thus suitable for mobile devices. Illumination invariance is achieved by retaining only the chrominance data. Whilst there are many ways to achieve this, we have selected to capture images in the HSV colour space and discard the V component.

In the case an Exit sign, its background is a standard green hence, we prebuilt a 2D histogram using the Hue and Saturation in the HSV colour space for individual green colour image captured under different lighting conditions. The advantage of HSV colour space for colour image processing is well researched and reported in the literatures [9]. The required data was manually cropped, to exclude white elements of the sign. Whilst this type of sign is extremely simple, having a one-colour background, the method is clearly applicable to any coloured sign.

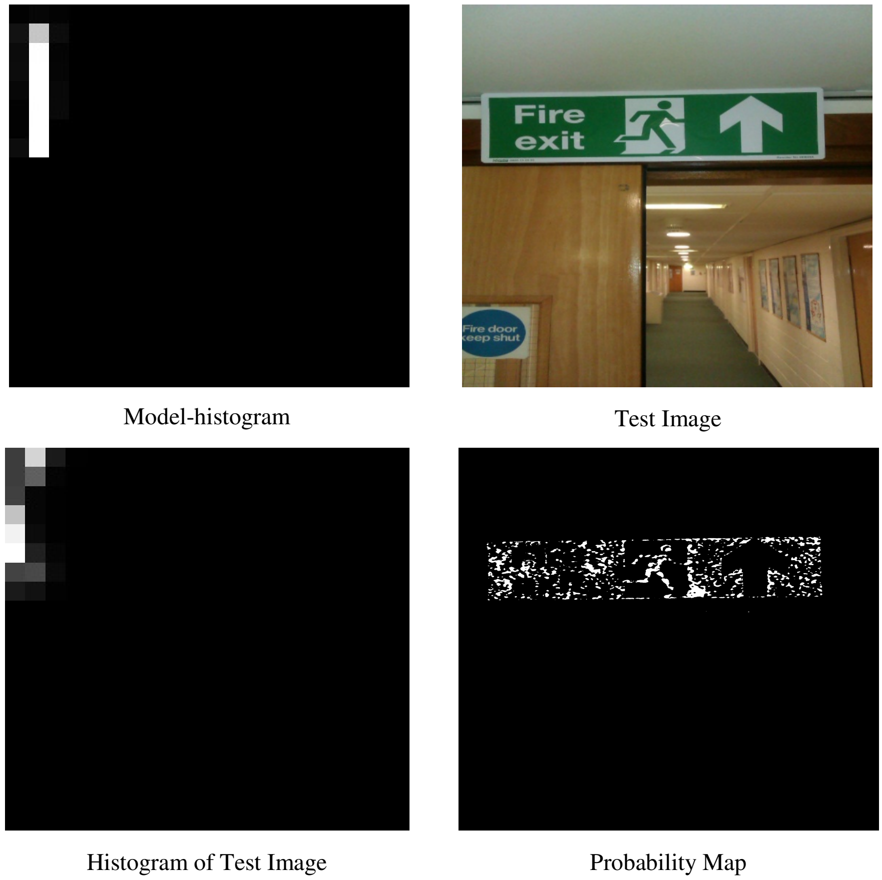

Fig. 2. (L-R) Model histogram, Input image, Histogram of input image, Backprojected image

In our experiment, we use a data set of 100 images to model the signs' colour distribution. The large number of images is to compensate for changes in the object appearance occasioned by camera motion, and, more importantly, lighting changes. The histogram's bin count is reduced to a manageable number to improve the histogram's accuracy and to reduce the computational requirement [8]. We tested our model histogram using different numbers of bins for the hue and saturation channels respectively and observed that a dense rather than a sparse histogram produced a better detection and tracking result. We performed all the steps prior to the back projection of the input image and saved the model histogram offline. This is to enhance tracking during run-time.

As explained in section 3, the model histogram served as a look-up table to generate a probability distribution image from the captured image. The result of histogram backprojection of the input image using the model histogram is shown in Figure 2.

As explained in Section 2, a search window is initialized with its size and position as the input image. CAMShift adjusts the position and size of the window in the back-projected image to locate the centre and size of the target object. In a sequence of video frames, CAMShift can predict the centre and size of the target object in the following frame using data from the previous frame.

6 Experimental Result

The first row of Fig. 3 shows images of emergency exit signs captured with an Android camera phone. As explained in Section 4, the model histogram that was saved offline is loaded at run-time to perform histogram backprojection on the input images. The result of this operation can be seen in row 2 of Fig. 3. Following this, CAMShift uses the probability distribution image to search for the peak location which corresponds to the location of the searched object and thus return the tracked region. The result of the tracking operation using CAMShift is shown in the last row of Fig. 3.

Fig. 3. Row 1: Original Exit sign Image; Row 2: Backprojected Image; Row 3: Tracked region of the exit image

7 Conclusion

In this paper, we have presented an approach to tracking an object on a mobile device. It makes use of reference colour histograms generated by accumulating and normalising histograms of target objects captured under various conditions that replicate the expected illuminations when the system is live. The reference histogram is saved offline to reduce computational time. This is a useful step towards tracking using CAMShift.

Our approach is simple and robust against lighting changes as shown in the results obtained, but fails to track the whole region of one of the signs because of the extreme camera viewpoint. This is evident in the tracked image shown in the second column of Fig. 3. This can be solved by integrating a shape feature to the detector. Furthermore, shape features can help improve tracking due to occlusion by object of similar colour.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A Survey. ACM Computing Survey 38, 1–45 (2006)
2. Bradski, G.R.: Computer Vision Face Tracking for Use in a Perceptual User Interface. Intel Technology Journal (1998)
3. Yuan, L., Mu, Z.-C.: Ear Detection Based on Skin-Color and Contour Information. In: 6th International Conference on Machine Learning and Cybernetics, vol. 4, pp. 2213–2217. IEEE, Hong Kong (2007)
4. Yilmaz, A., Li, X., Shah, M.: Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras. In: IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 26, pp. 1531–1536. IEEE Computer Society (2004)
5. Yang, D., Xia, J.: Face Tracking Based on Camshift Algorithm and Motion Prediction. In: International Workshop on Intelligent Systems and Applications, pp. 1–4. IEEE, Wuhan (2009)
6. Exner, D., Bruns, E., Kurz, D., Grundhofer, A., Bimber, O.: Fast and Robust CAMShift Tracking. In: Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–16. IEEE, San Francisco (2010)
7. Yue, Y., Gao, Y., Zhang, X.: An Improved Camshift Algorithm Based on Dynamic Background. In: 1st International Conference on Information Science and Engineering, pp. 1141–1144. IEEE, Nanjing (2009)
8. Allen, J.G., Xu, R.Y.D., Jin, J.S.: Object Tracking Using Camshift Algorithm and Multiple Quantized Feature Spaces. In: 5th Pan-Sydney Area Workshop on Visual Information Processing, pp. 3–7. ACM, Australia (2004)
9. Sural, S., Qian, G., Pramanik, S.: Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval. In: International Conference on Image Processing, vol. 2, pp. 589–592. IEEE (2002)
10. Swain, M.J., Ballard, D.H.: Indexing via Colour Histogram. In: Active Perception and Robotic Vision, vol. 83, pp. 261–273. Springer (1992)
11. Yoo, T.-W., Oh, I.-S.: A Fast Algorithm for Tracking Human Faces Based on Chromatic Histograms. In: Patter Recognition Letters, vol. 20, pp. 967–968. Elsevier (1999)

12. Chen, T.M., Luo, R.C., Hsiaso, T.H., Chia-Yi: Visual Tracking Using Adaptive Colour Histogram Model. In: 25th Annual Conference of IEEE, vol. 3, pp. 1336–1341. IEEE, San Jose (1999)
13. Fan, L., Riihimaki, M.: A Feature-Based Object Tracking Approach for Real Time Image Processing on Mobile Devices. In: 17th International Conference on Image Processing. IEEE, Hong Kong (2010)
14. Gong, J., Jiang, Y., Xiong, G., Guan, C., Tao, G., Chen, H.: The Recognition and Tracking of Traffic Lights Based on Colour Segmentation and CAMShift for Intelligent Vehicles. In: Intelligent Vehicles Symposium (IV), pp. 431–435. IEEE, San Diego (2010)