IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 2, Ver. III (Mar. - Apr. 2018), PP 31-37
www.iosrjournals.org

Color Feature Based Object Localization In Real Time Implementation

Panca Mudjirahardjo
Department Of Electrical Engineering, Faculty Of Engineering, Brawijaya University, Indonesia
Cortresponding Asuthor: Panca Mudjirahardjo

Abstract: *One of the important task in pattern detection and recognition is object localization. This important task is performed to reduce the searching time to the interest object. In this research we demonstrate our novel method of object localization based on color feature. Our novel method is a combination of histogram of s-RGB and histogram of Hue. In the training phase, we use these histograms to determine the color dominant in the initial Region of Interest (ROI). Then this information is used to label the interest object. We apply the row and column density function of pixels to reduce noise and localize the interest object. The comparison result with some processes, our system gives a best result and takes a short computation time of 48.37 ms, in the video rate of 15 frames per second (fps).*

Keywords *- color feature; histogram of s-RGBH; object localization; pattern recognition*

Date of Submission: 02-04-2018 Date of acceptance: 17-04-2018

I. Introduction

One of the task in pattern detection and recognition is object localization. This important task is required to reduce the searching time to the interest object. We don't require to run a window to search the object by scanning the whole of image frame. We only label the pixels in initial step to localize the interest object based on the feature. There are some features implementing for this task, such as color feature, texture feature, orientation feature and shape feature.

Some methods for object localization, such as matching method to match these objects with data stored in a database [1][2]. Clustering algorithm as in [3][4]. Clustering methods require the several clusters as the output of the system[5]. Segmentation system similar to [6]. The segmentation system or methods always need an origin point for the algorithm. Spatial Data Analysis involves illustration such relationships between objects based on their spatial properties i.e. positions, area coverage, density [7][8][9][10].

\Ozuysal et.al. [1] proposed a layered approach to object detection. They trained an estimator for the bounding box dimensions by using Support Vector Machine (SVM) classifier. Object in their database rotated at constant angular velocity and recovered its value by using the time of capture of a full rotation. This method had a high precision both at 0.5 overlap and 0.7 overlap. Li et.al [2] proposed a two-step domain adaptation for weakly supervised object localization: classification adaptation and detection adaptation. In the classification adaptation step, they trained a classification network using the given weak image-level labels. They trained the classification network to recognize the presence of a certain object category in an image. In the detection adaptation step, they used the classification network to collect class-specific object proposals and applied multiple instance learning to mine confident candidates.

Marszalek et.al. [3] used the Harris-Laplace or the Laplacian interest point detector to find a sparse set of salient image features. A featured shape mask similarity was used to measure similarity between two features. And to evaluate the shape masks, they used a bag-of-keypoints representation and a non-linear Support Vector Machine (SVM) with x^2 kernel. With this method, they were able to detect and localize object in vary position well. However, they required to train some positive and negative samples. Bostanci et.al. [4] proposed nethermost stretching trees which were often used for network routing so that minimize the number of hops for packages. The graph in their algorithm was built using the 2D positions of the point features acquired by the FAST algorithm. By this method, they were able to detect objects in stuck together. In this experiment, however, they used a one color background.

Self-Taught Localization (in brief STL) proposed by Bazzani et.al. [5] was to generate bounding boxes that are very likely to contain objects. Their proposed approach relied on the idea of covering out regions of an image provided as input to a deep network. The reduction in recognition score caused by the covering out is embedded into an agglomerative clustering method which merges regions for object localization.

A graph search algorithm was used by Sun et.al. [6] to split the graph of line fragments into smaller ones using certain node and edge weighting functions. They encoded how likely a line fragment or line fragment pair is to belong to an object. They used generic appearance and shape cues to score each cycle. These score are not constricted to linearly additive measures and can easily yield multiple hypotheses that share some edges.

Sigal [7] initialized object localization by performing object detection. In object detection, he performed object's part detection before detected the whole of object. He implemented graphical models and its inference method to detect the object's part and loose-limbed body model to localize the object. These method were able to detect and localize in multi-frame for single target and single frame for multi-target. However, it required a complex computation. Murphy et.al. [8] implemented feature dictionary with the bank of 13 filters and patch classifier of gentleBoost algorithm as local image feature. And gist of an image was as global image feature. With this method, they were able to detect and localize object well. However, they implemented the standard technique of object detection using sliding window classifiers applied to local features which required a large computation time. Leitner et.al. [9] investigated a novel approach to spatial perception. Their method combined the two calibration tasks, defining the camera parameters and precise kinematics, into one machine learning problem, removing the need for prior calibration. They applied machine learning approaches: Artificial Neural Networks (ANN) and Genetic Programming (GP). As the result, the accuracy achievement of ANN technique was better than GP.

To localize a head as object from background, Mudjirahardjo *et al* [10] used Euclidean distance function of the determined pixels as object and the surrounding pixels. The pixel was a part of object, when the distance was less than a threshold. They computed the distance in RGB and HSV color space. Mudjirahardjo *et al*. [11] extracted the interest object in dynamic background by using velocity histogram based on Harris corner detector, Lucas-Kanade tracker and shift histogram based on motion history image (MHI).

In this paper, we demonstrate and evaluate our novel method. Our novel method is segmentation method. It doesn't require artificial intelligence as learning algorithm.

II. Overview of the System

The overview of the proposed method is shown in Fig. 1 and 2. There are two phases to perform this method, i.e. training phase and running phase. In the training phase, we initialize the ROI manually then determine the dominant color in RGB space, based on histogram of s-RGB and in HSV space, based on histogram of HUE. The output of this phase are modus of s-RGB bin and modus of Hue. These two modus determine every pixel in the next subsequence frame belongs to the interest object or background. To evaluate the system performance, we compare the color feature in RGB space and combination of RGB-HSV space.

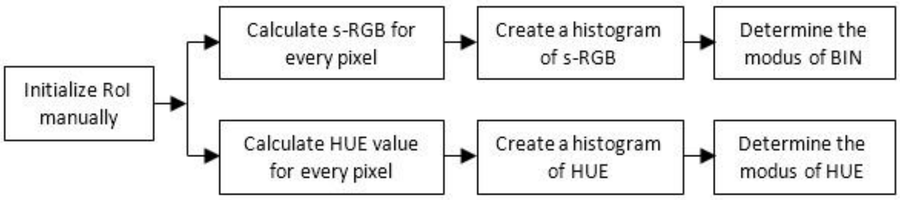

Figure 1. The overview of training phase

III. Method

This section explains our method to develop the object localization. First, we initialize the region of interest (ROI) of the object manually in the first frame. Second, we determine the dominant color based on histogram of s-RGB and histogram of Hue. Third, we calculate the density function of pixels to localize the object.

III.1 Determination the region of interest

To give flexibility an object which will be localized in the sequence frames, the first step is to initialize the region of interest (ROI) manually. This ROI is a part of the object. The aim is to capture the color feature of ROI. We calculate the histogram of s-RGB and histogram of Hue to get the dominant color of object. Then for the next sequence frames, we localize the object which has pixels with this dominant color in the determined s-RGB and Hue value.

III.2 Histogram of s-RGB

To provide the dominant color in a frame image, the process is explained in the following sub chapter. First, we calculate the sum of RGB intensity (s-RGB) at each pixel [12]. This calculation as in equation (1),

$$s-RGB(x,y)=I_R(x,y)+I_G(x,y)+I_B(x,y) \tag{1}$$

where *s-RGB*(*x*,*y*) is sum of RGB intensity at pixel coordinate (*x*,*y*), $I_R(x,y)$, $I_G(x,y)$, and $I_B(x,y)$ are red, green and blue intensity at pixel coordinate (*x*,*y*), respectively. When we use 8 bit to code a color intensity, then we can get the s-RGB value of 0-765. Second, we divide the s-RGB value into 16 bins, then create the histogram of s-RGB as shown in Fig. 3.

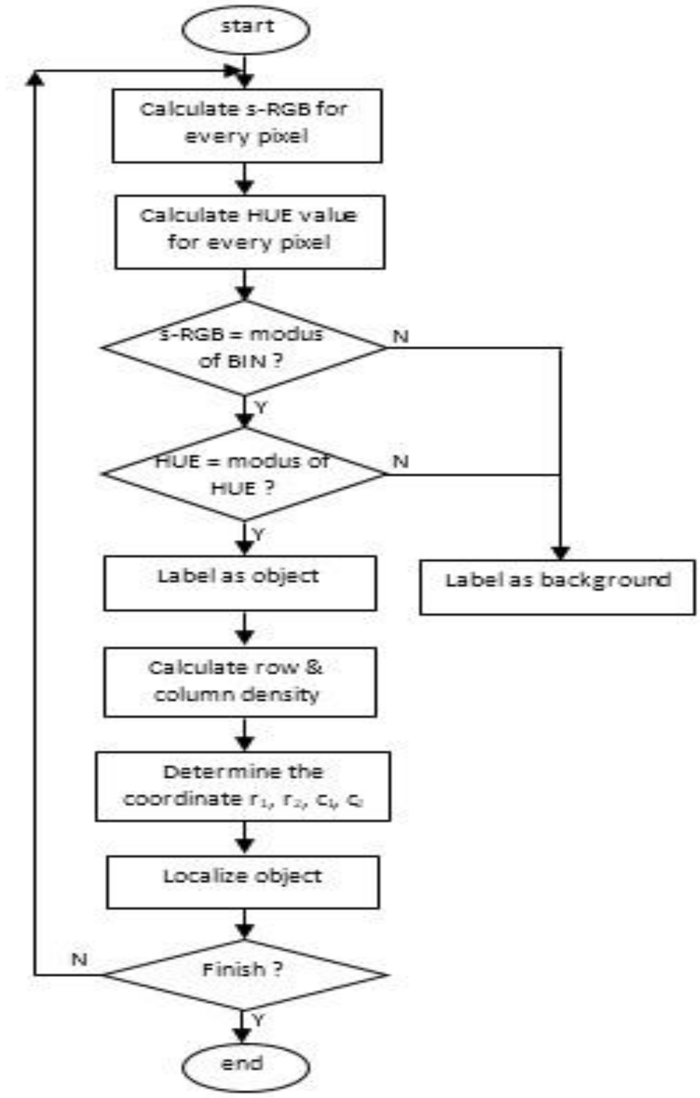

Figure 2. The overview of running phase

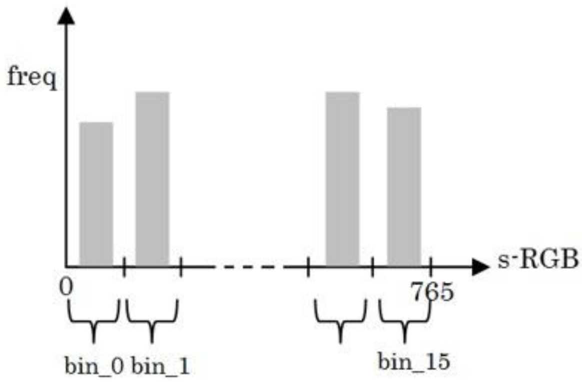

Figure 3. A histogram of s-RGB [9]

Third, from this histogram we determine modus bin of s-RGB by using equation (2),

$$\text{mod}_{\text{s-RGB}} = \arg\max_{bin}\left(histogram\_s - RGB\right) \quad (2)$$

This modus value restricts the pixels having bin of s-RGB for further processing.

III.3 Creating the histogram of Hue

Another color data information which we need in this method is histogram of Hue. It can be provided from HSV space (Hue, Saturation, Value). Conversion from RGB space into HSV space is as follow,

$$H = \begin{cases} \text{undefined} & \text{if } M - m = 0 \\ \left(60\dfrac{G-B}{M-m}\right)\text{mod}360 & \text{if } M = R \\ 60\dfrac{B-R}{M-m} + 120 & \text{if } M = G \\ 60\dfrac{R-G}{M-m} + 240 & \text{if } M = B \end{cases} \quad (3)$$

Where M = max(R,G,B) and m = (R,G,B).

$$S = \frac{M - m}{M} \quad (4)$$

And

$$V = M \quad (5)$$

Then we create histogram of Hue. From this histogram, we can provide modus of Hue as (6),

$$\text{mod}_{\text{Hue}} = \arg\max_{Hue}\left(histogram\_Hue\right) \quad (6)$$

From (6) we will see what Hue value is dominant in the frame image.
If we define the identity of pixel in image, I_P, as in (7),

$$I_P = \begin{cases} 1 & \textit{if } \text{s-RGB within mod}_{\text{s-RGB}} \text{ AND Hue within mod}_{\text{Hue}} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Then we decide a pixel in image as the interest object as in (8),

$$pixel = \begin{cases} I_P \times (\text{the interest object}) & \text{if mod}_{s-RGB} - th_1 < I_{s-RGB}(x,y) < \text{mod}_{s-RGB} + th_1 \\ & AND\ \text{mod}_{Hue} - th_2 < I_{Hue}(x,y) < \text{mod}_{Hue} + th_2 \\ I_P \times (\text{non-interest object}) & otherwise \end{cases} \quad (8)$$

where th_1 and th_2 are s-RGB threshold and Hue threshold value respectively; $I_{s\text{-}RGB}(x,y)$ and I_{Hue} are intensity of s-RGB and Hue at pixel coordinate (x,y) respectively.

III.4 Calculation the density function of interest pixels

This step is to reduce the noise. Due to we rely on color feature to localize an object, it should be another object with the same color will be localized. To determine the interest object, we apply the row and column density function of interest pixels. From binary image, result of object labeling, calculation the density function of interest pixels as follows

$$f_x = \sum_{n=1}^{r} I_{x,n} \quad (9)$$

$$f_y = \sum_{m=1}^{c} I_{y,m} \quad (10)$$

Where f_x and f_y are column and row density function respectively. x = {1,…, column size of image}. y = {1, …, row size of image}. $n = y$ and $m = x$.
To label a pixel as an interest object or the background, we apply an equation (11),

$$I_{a,b} = \begin{cases} 1 & \text{if pixel at } (a = x \text{ or } y) \text{ and } (b = n \text{ or } m) \text{ as object} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The row density function, f_y, and its properties is shown in Fig. 4. By applying a threshold, d, we can determine two coordinates of row. This calculation is also applied for column density function, f_x, to determine two coordinates of column. These two coordinates of row and two coordinates of column are used to localize the interest object.

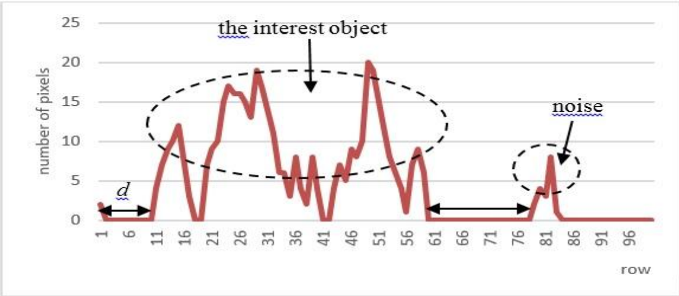

Figure 4. Row density function of pixels.

IV. Experimental Result And Discussion

To conduct the experiment, we set-up the experimental environment as follows: Operating system is Windows 8.1 Pro; the processor is Intel® core^{TM} i5-4210 U CPU @ 1.70 GHz 2.40 GHz; 4GB RAM; the built in web camera; and the used software is Microsoft Visual Studio 2010. We evaluate the image size of 640×480 pixels. Video rate is 15 fps. The interest object is my face. We set the initial ROI size of 100×100 pixels.

We perform a comparison of two color features. The first feature, we perform a framework as in Figure 1 and 2. The second process, we perform a framework based on histogram s-RGB only.

The experimental results are shown in Fig. 5,6 and Table 1. In Fig. 5 and 6, the first and second row, the RoI are my face. I wear a brown shirt. The third row, the RoI is a book. Fig. 5 show the s-RGBH feature can localize object and reduce noise better than by using the s-RGB feature (Fig. 6).

The computation time is shown in Table 1. It shows suitable for real time processing. For 15 fps, the period time is 66.67 ms. The computation time of our method is 48.37 ms.

Figure 5. The result based on histogram of s-RGBH.

Figure 6. The result based on histogram of s-RGB.

Table 1. The computation time for difference processes.

Process	Computation time (ms)
Histogram of s-RGBH (Figure 4)	48.37
Histogram of s-RGB (Figure 5)	41.24

V. Conclusion

In this paper we use s-RGBH feature as color feature to determine the color dominant in RGB and HSV space to label the interest object. Training phase is performed in the first frame in the initial ROI. Together with row and column density function of pixels, we develop an object localization. The row and column density function effectively to reduce noise. The comparison result with some processes, our system gives a best result and takes a short computation time of 48.37 ms, as suitable for real time application.

To improve the object detection and localization, our future work is to add another feature, such as texture feature and shape feature into our system.

Statement

In this research, I involved author's face as the object of this study, as mentioned it in section 4.

Acknowledgments

This work was supported by DRPM Ditjen Penguatan Riset dan Pengembangan Kemristekdikti Indonesia, Grant No. 063/SP2H/LT/DRPM/IV/2017.

References

[1]. M. Ozuysal, V. Lepetit, P. Fua. *Post Estimation for Category Specific Multiview Object Localization*. Proceeding of Computer Vision and Pattern Recognition (CVPR2009), (Florida, USA. 2009) pp. 1-8.

[2]. D. Li, J.B. Huang, Y. Li, S. Wang, M.H. Yang. *Weakly Supervised Object Localization with Progressive Domain Adaptation*. Proceeding of Computer Vision and Pattern Recognition (CVPR 2016), (Las Vegas, USA. 2016) pp. 3512-3520.

[3]. M. Marszalek, C. Schmid. *Accurate Object Localization with Shape Masks*. Proceeding of Computer Vision and Pattern Recognition (CVPR 2007), (Minnesota, USA. 2007) pp. 1-8.

[4]. E. Bostanci, B. Bostanci. *Object Localization and Spatial Analysis Using Computer Vision.* International Journal of Machine Learning and Computing, Vol. 1, No. 2, June 2011. Pp. 1-5.

[5]. L. Bazzani, A. Bergamo, D. Anguelov, L. Torresani. *Self-Taught Object Localization with Deep Networks.* Proceeding of Winter Conference on Applications of Computer Vision (WACV 2016), (New York, USA, 2016) pp. 1-9.
[6]. X. Sun, C.M. Christoudias, P. Fua. *Free-Shape Polygonal Object Localization.* Proceeding of European Conference on Computer Vision (ECCV 2014) (Zurich, Switzerland, 2014) pp. 317-332.
[7]. L. Sigal. *Continuous-state Graphical Models for Object Localization, Pose Estimation and Tracking.* Ph.D. Thesis. Brown University, Providence, Rhode Island, May 2008.
[8]. K. Murphy, A. Torralba, D. Eaton, W. Freeman. *Object Detection and Localization using Local and Global Features.* Proceeding of Computer Vision and Pattern Recognition (CVPR 2005) (San Diego, USA, 2005) pp. 1-20.
[9]. J. Leitner, S. Harding, M. Frank, A. Foster, J. Schmidhuber. *Learning Spatial Object Localization from Vision on a Humanoid Robot.* International Journal of Advanced Robotic Systems. Vol. 9, No. 243, 2012. Pp. 1-10.
[10]. P. Mudjirahardjo, M.F.E. Purnomo, R.N. Hasanah, H. Suyono. Histogram of Transition for Human Head Recognition. Jurnal Teknologi. 78(5-9): 53-58. (2016).
[11]. P. Mudjirahardjo, J.K. Tan, H. Kim, S. Ishikawa. Temporal Analysis for Fast Motion Detection in a Crowd. Journal Artificial Life and Robotics. 20: 56-63. (2015).
[12]. P. Mudjirahardjo, Nurussa'adah, P. Siwindarto. Soccer Field Detection Based on Histogram of s-RGB. ARPN Journal of Engineering and Applied Sciences. 11(21): 12405-12408. (2016).

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

* Panca Mudjirahardjo. "Color Feature Based Object Localization In Real Time Implementation." IOSR Journal of Computer Engineering (IOSR-JCE) 20.2 (2018): 31-37.