Jurnal
Teknologi

Full Paper

HISTOGRAM OF TRANSITION FOR HUMAN HEAD RECOGNITION

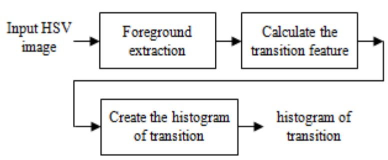
Panca Mudjirahardjo, M. Fauzan Edy Purnomo, Rini Nur Hasanah, Hadi Suyono

Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Jl. MT. Haryono 167,Malang, Indonesia (65145)

Article history
Received
12 August 2015
Received in revised form
18 December 2015
Accepted
5 January 2016

*Corresponding author
panca@ub.ac.id

Graphical abstract



Abstract

The main component for head recognition is a feature extraction. One of them as our novel method is histogram of transition. This feature is relied on foreground extraction. In this paper we evaluate some pre-processing to get foreground extraction before we calculate the histogram of transition.
We evaluate the performance of recognition rate in related with preprocessing of input image, such as color, size and orientation. We evaluate for Red-Green-Blue (RGB) and Hue-saturation-Value (HSV) color image; multi scale of 10×15 pixels, 20×30 pixels and 40×60 pixels; and multi orientation angle of 315°, 330°, 345°, 15°, 30°, and 45°.
For comparison, we compare the recognition rate with the existing method of feature extraction, i.e. Histogram of Oriented Gradients (HOG) and Linear Binary Pattern (LBP). The experimental results show Histogram of Transition robust for changing of color, size and orientation angle.

Keywords: Histogram of transition; head recognition; HOG; LBP.

1.0 INTRODUCTION

Head detection and recognition have been an important research in the last few years. Many applications use this research, such as robotics, automated room monitoring, people counting, person tracking, etc. Many new methods are introduced in this field, to improve the computation time and the recognition rate. One of them is the method based on feature extraction.

Feature extraction plays an important role in head recognition. It transforms an original image into a specific vector to be fed into a classifier. An original image cannot be further processed directly. Raw information in an original image does not represent a specific pattern and a machine cannot understand that information.

In an image, there are foreground and background patterns. In a simple image, foreground and background pattern can be separated clearly. In a complex image, however, foreground and background pattern cannot be separated clearly. There are many texture patterns both on foreground and background. Sometimes, both foreground and background contain similar texture and color on them. This is a difficult task in a head detection and recognition system. The system has to recognize a foreground pattern as a head or a non-head. Correct choice of a foreground extraction method will increase the recognition rate.

A feature is assumed to be able to distinguish a foreground and a background pattern. All of features distinguish a foreground pattern over the background from the edge pattern of the foreground, since a foreground has a specific edge pattern over the background.

Currently the most commonly used feature extraction methods are Histogram of Oriented Gradients (HOG)[1][2] and Linear Binary Pattern

(LBP)[3]. The new feature extraction is a histogram of transition as our novel method [5][6]. This feature is relied on a background extraction. The simple method to extract a foreground is by using a difference function. Where we define some pixels as foreground pixels, then we calculate all pixel intensity with respect to the foreground pixels. If the difference result is less than or equal to the threshold, then the pixel is consider as foreground, otherwise is as background.

The overview of this experiment is shown in Figure 1.The structure of this paper is as follows. Section 2 explains the preprocessing. Histogram of transition method is overviewed in section 3. Experimental results are shown in section 4. Finally the paper is concluded in section 5.

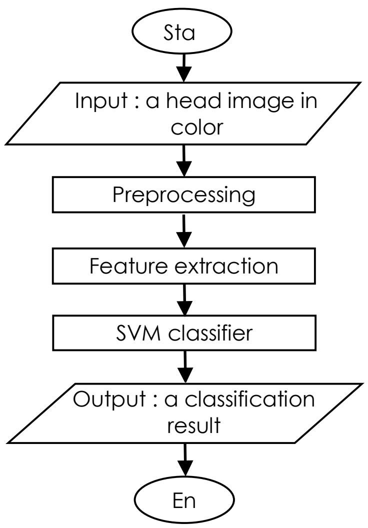


Figure 1 Overview of the experiment.

2.0 PREPROCESSING

In this section, we normalize the image size into the normal size 20×30 pixels. Algorithm to increase the image size 10×15 pixels into 20×30 pixels (twice scaling) is depicted in Figure 2, as follows [12]:

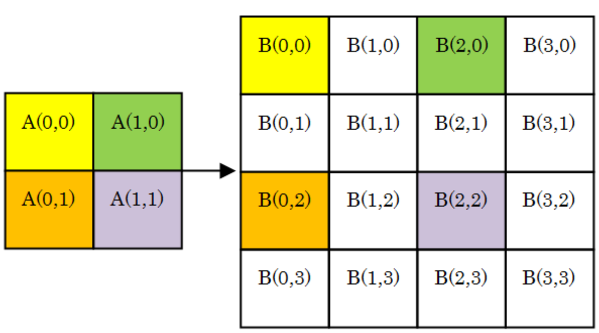


Figure 2 Illustration to increase the image size

Initially for all (colB = even and rowB = even), we set :

$$I_{B(\mathrm{colB},\mathrm{rowB})} = I_{A(\mathrm{colB}/2,\mathrm{rowB}/2)} \qquad (1)$$

if (colB = odd and rowB = even), then

$$I_{B(colB,rowB)} = \frac{I_{B(colB-1,rowB)} + I_{B(colB+1,rowB)}}{2} \qquad (2)$$

if (colB = even and rowB = odd), then

$$I_{B(colB,rowB)} = \frac{I_{B(colB,rowB-1)} + I_{B(colB,rowB+1)}}{2} \qquad (3)$$

if (colB = odd and rowB = odd), then

$$I_{B(colB,rowB)} = \frac{I_{B(colB-1,rowB-1)} + I_{B(colB+1,rowB+1)}}{2} \qquad (4)$$

if (colB = end of colB) , then $I_{B(\mathrm{colB},\mathrm{rowB})} = I_{B(\mathrm{colB-1},\mathrm{rowB})}$
if (rowB = end of rowB) , then $I_{B(\mathrm{colB},\mathrm{rowB})} = I_{B(\mathrm{colB},\mathrm{rowB-1})}$

where $I_{B(x,y)}$: pixel intensity of image B at pixel location (x,y)

The algorithm to reduce the image size 40×60 pixels into 20×30 pixels (half scaling) is depicted in Figure 3, as follows:

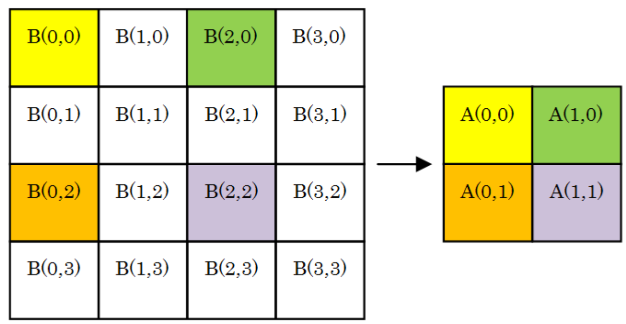


Figure 3 Illustration to reduce the image size

$$I_{A(n,m)} = \max(I_{B(n,m)}, I_{B(n+1,m)}, I_{B(n,m+1)}, I_{B(n+1,m+1)},). (2)$$

In Figure 2 and Figure 3, the same color means the same pixel intensity.

3.0 HISTOGRAM OF TRANSITION

In this section, we explain our novel method, a histogram of transition [5][6]. As the first step to create a histogram of transition, we calculate a transition feature. A transition feature is to compute the location and the number of transitions from background to foreground along horizontal and vertical lines. So, this transition feature relies on foreground extraction. Figure4 shows the overview of creating a histogram of transition.

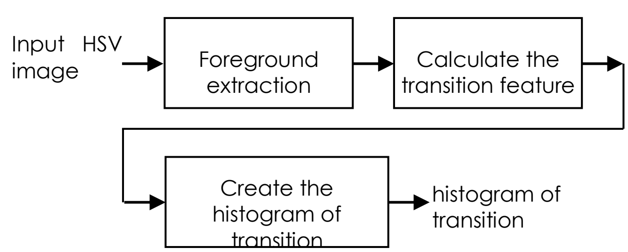


Figure 4 The overview of histogram of transition feature extraction

In this paper, we consider to use a simple foreground extraction. A simple algorithm to extract foreground is that we determine some reference pixel coordinates as foreground [5][6]. Then we compare another pixel's intensity (I_x) to the reference pixel's intensity (I_R). Since we extract foreground in an RGB image, we have pixel's intensity in red, green and blue. We determine a pixel at coordinate (x,y) as foreground or background by equation (3),

$$I(x,y)=\begin{cases}\text{foreground} & \delta(I_x,I_R)<th\\ \text{background} & \text{otherwise}\end{cases} \qquad (3)$$

where $\delta(.)$ is a distance function. In this paper, we use Euclidean distance.

After we perform (3), we perform opening morphology by using 3×3 structuring elements to get the foregrounds. Then we extract the feature of these foregrounds. Our feature refers to [7][8]. Transition feature has been used successfully in handwritten characters recognition, but it hasn't been used in head detection yet. Due to a simple calculation to create a feature vector, we apply the idea to head recognition. We do some modifications on it to be able to be used for head recognition.

The idea is to compute the location and the number of transitions from background to foreground along horizontal and vertical lines. This transition calculation is performed from right to left, left to right, top to bottom, and bottom to top. Since a constant dimensional feature is required as input to the SVM classifier, an encoding scheme is developed.

In the first stage of feature extraction, the transition in each direction is calculated. Each transition is represented as a fraction of the distance across the image in the direction under consideration. These fractions are computed in the increasing order, differed from [8] in decreasing order. For example, when calculating the location of transitions from left-to-right, a transition close to the left edge would have a low value and a transition far from the left edge would have a high value as illustrated in Figure 5.

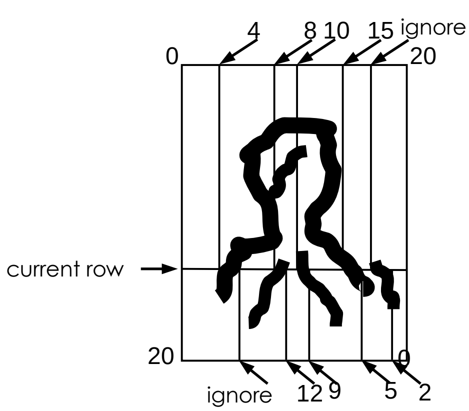


Figure 5 The first stage of transition feature extraction shown for transitions from the left and from the right on one row of the image, with M = 4.

The maximum number of transitions, M, are counted on each line. If there are more than M transitions in a line, only the first M are counted and the rest are ignored. M is set to 4 in the experiment. If there are less than M transitions on a line, then the "nonexistent" transitions are assigned as a value of 0.

More precisely, by a line we mean a row or a column of the head image. Let h be the height of the image and w be the width of the image. We assign exactly M values to each line, say $t_1,t_2,\ldots,t_M$. We assume that there are n transitions on a line located at (x_i,y_i) for i = 1,2, … n. The algorithm for calculating the transition feature can be represented as follows: It doesn't require normalization as in [7][8]:

```
for i = 1 to min (n,M)
        if the line is a row then
              ti = xi;
        else
              ti = yi;
        end if;
end for;
if n <M then
        For i = n+1 to M
             ti = 0;
        end for;
end if;
```

The transitions are resampled to a 4-point sequence for each direction and assembled into a feature vector. The four transitions for each row (column) are represented as two-dimensional (2-D) array, t = [t_{ij}] for i = 1,…,h(w) and j = 1,…,4.

The second stage is generating a histogram of transition. It is different from [8] where they calculated local averaging on the columns of t. Histogram of transition shows how often the location of the transition occurs at each transition. An example of generating a histogram of transition for transition left-to-right is shown in Figure 6.

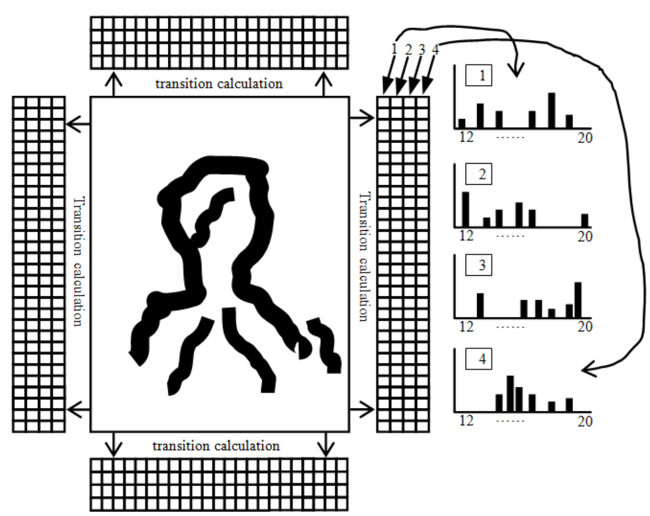


Figure 6 The second stage of transition feature calculation consisting of generating a histogram of transition

This histogram of transition creates a feature vector to be fed into the input of a SVM classifier [9].

The HOG feature contains gradient information of a pixel among its neighbor. Thus they give a high magnitude at the edge. On the other hand, the LBP feature gives a binary pattern with a pixel among its neighbor. The histogram of transition feature looks like the HOG feature: It gives the edge position from right, left, top and bottom side. In contrast to the HOG feature, the calculation of the histogram of transition feature is simpler.

4.0 EXPERIMENTAL RESULTS

The experimental environment is as follows: Operating system is Windows 7 professional; the processor is Intel® core™ i7 CPU 870 @2.93GHz and the used software is Microsoft Visual Studio 2010.

For robust detection, we use backgrounds and negative samples at outdoor scenery. We use INRIA data [1][10] for training and testing images. For training, we use image size 20×30 pixels, positive sample of 2,000 images, negative sample of 4,500 images. For testing, we use image size of 10×15 pixels, 20×30 pixels and 40×60 pixels. Positive sample of 100 images, negative sample of 300 images are employed. The image orientations are 315°, 330°, 345°, 15°, 30° and 45°. Figure 7 shows some positive and negative samples of normal orientation, and Figure 8 shows the input image with multi orientation.

Figure 7 Samples for SVM training with normal orientation: (a) positive samples, (b) negative samples

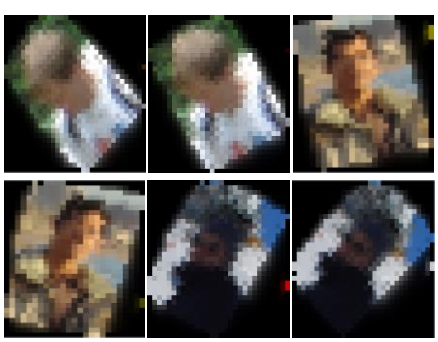


Figure 8 Input image with multi orientation, from top-left to the bottom-right, 315°, 330 °, 345 °, 15 °, 30 ° and 45 ° respectively

In this research, we use 9 bins to cluster the gradient image [11]. One is for value I_x and I_y equals zero, the other 8 bins for representing the orientation from 0° to 360°. For image size 20×30, cell size 5×5, and block size 15×15 with overlapped pixels of 5 pixels, the number of feature dimensions equals

the_number_of_bins × *the_number_of_cells_in_a_block* × *the_number_of_blocks*, which amounts to 648 dimensions.

In using the LBP operator, we divide an image into four non-overlapping regions [11]. The number of feature dimensions equals

the_number_of_region × *max_number_of_decimal_value*,

which amounts to 1020 dimensions.

For the image preprocessing to extract transition feature, we extract the foreground using a difference function Eq. (9). First, we determine 5 and 32 reference pixel coordinates as foreground. These coordinates are fixed for all the training and the test data. The coordinates should represent position of head and shoulder. For 5 reference pixel coordinates are (10,8), (10,15), (10,22), (5,22) and (15,22) as in [5][6]. For 32 reference pixel coordinates are (10,*i*) and (*j*,26), where *i* = 6,7,…,26 and *j* = 15,16,…,25. Then, we check all pixels' intensity to the five reference pixel's intensity with Euclidean distance, by Eq. (9). If a pixel's intensity distance to the one or more of five reference pixel's intensity is less than a threshold, the pixel should be a foreground pixel, otherwise a background pixel [11].

In this research, the maximum number of transition, *M*, is 4. The number of feature dimension,

2×((*M*×*width_of_image*)+ (*M*×*height_of_image*)), are 400 dimensions.

The recognition rate of positive and negative sample for Histogram of Transition is shown in Figure 9. From that chart, the HSV color image yields the recognition rate better than the RGB color image, both for positive and negative image. So, for the next experiment, we use the HSV color image for the input image and 32 reference pixel coordinates.

The result of multi scale performance is summarized in Table 1[12].The result of head recognitionfor normal orientation is summarized in Table 2, and for multi orientation is summarized in Table 3 and Table 4.

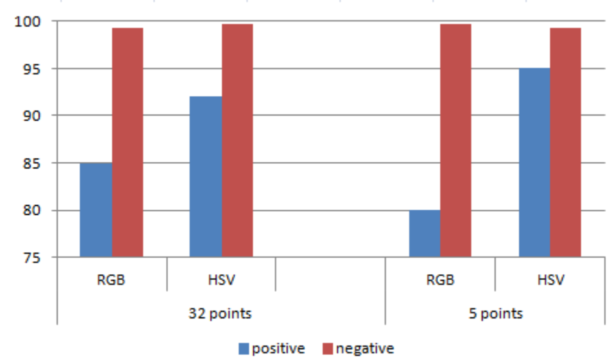


Figure 9 Recognition rate of histogram of transition based on image color. Image size is 20×30 pixels

Table 1 The result of multi scale performance

Feature	The number of array	Recognition rate (%)						Execution time (ms)
		Small size (10×15)		Normal size (20×30)		Big size (40×60)		
		positive	Negative	Positive	Negative	Positive	negative	
HOG	648	83	98.00	84	98.30	84	98.67	0.353
LBP	1020	92	61.00	**95**	80.00	**96**	61.33	0.261
Histogram of transition	400	**92**	**99.70**	92	**99.70**	92	**99.67**	0.087

Table 2 The result of head recognition for normal orientation

Feature	The number of array	Recognition rate (%)		Execution time (ms)
		positive	Negative	
HOG	648	84	98.3	0.353
LBP	1020	95	80.0	0.261
Histogram of transition (32)	**400**	**92**	**99.7**	**0.087**

Table 3 The result of head recognition with multi orientation

orientasi	FHOG			LBP			Histogram of transition (5)		
	positive	negative	total	positive	negative	total	positive	negative	total
315◦	45	92.33	80.5	43	98	84.25	77	94.33	**90**
330◦	56	98.33	87.75	59	94.67	85.75	79	95.67	**91.5**
345◦	57	99.33	88.75	69	94.67	88.25	88	98	**95.5**
15◦	63	99.67	90.5	88	87	87.25	91	99.33	**97.25**
30◦	49	99.67	87	48	95.67	83.75	76	95.67	90.75
45◦	44	98	84.5	56	94	84.5	80	95.67	**91.75**

Table 4 The result of head recognition with multi orientation

orientasi	Histogram of transition (32)		
	positive	negative	total
315◦	42	100	85.5
330◦	54	100	88.5
345◦	84	99	95.25
15◦	86	99.33	96
30◦	76	98.67	**93**
45◦	55	98.67	87.75

5.0 CONCLUSION

In this paper, we evaluate the performance of preprocessing of input image, then we extract the feature and perform a comparison of the existing image features extraction methods using a static image. The existing features are HOG and LBP, and the proposed feature is a histogram of transition.

In design, the proposed feature is robust for multiscale image and multi orientation angle. It has the acceptable recognition rate compared to the existing features.

The proposed feature has the number of array less than the existing features, and the computation of feature transition is simpler than the existing features. These conditions give the computation of the proposed feature faster than the computation of existing features.This performance shows that the proposed feature can be used for real time application.

As future work, we are going to conduct experiments to improve foreground extraction.

Acknowledgement

We are grateful for the Power System Engineering and Energy Management Research Group (PSeeMRG) of Universitas Brawijaya, Indonesia, for financing this publication.

References

[1] Dalal N., Triggs, B. 2005. Histograms of oriented gradients for human detection, *Proceeding of Computer Vision and Pattern Recognition*. 1: 886-893.

[2] Perdersoli M.,Gonzalez J., Chakraborty B., VillanuevaJ. 2007. Boosting Histograms Of Oriented Gradients For Human Detection, In: Proc. 2nd Computer Vision: *Advances in Research and Development (CVCRD)*. 1–6.

[3] Heusch G., Rodriguez Y., MarcelS. 2006. Local Binary Patterns As An Image Preprocessing For Face Authentication, *Proceeding of Automatic face and Gesture Recognition (FGR 2006)*. 9-14.

[4] OjalaT., Pietik¨ainen M., Harwood D.1996. A Comparative Study Of Texture measures with classification based on

featured distributions, *Pattern Recognition*, 29(1): 51–59.

[5] Mudjirahardjo P., Tan J.K., Kim H., Ishikawa S. 2014. Head Detection And Tracking For An Intelligent Room, Proc. *SICE Annual Conference 2014*. 353-358.

[6] Mudjirahardjo P., Tan J.K., Ishikawa S. 2015. A Study On Human Motion Detection - Toward Abnormal Motion Identification. *Ph.D. Thesis, Kyushu Institute of Technology, Japan*, 35-55.

[7] Mudjirahardjo P. Widodo T.S., Susanto A. 2001. Penerapan Jaringan Perambatan Balik Untuk Pengenalan Kode Pos Tulisan Tangan (The Implementation Of Back-Propagation Network For Recognition Of Handwritten Post Code). *Master Thesis, Universitas Gadjah Mada*,1-100.

[8] GaderP.D., MohamedM., ChiangJ.H. 1997. Handwritten Word Recognition With Character And Inter-Character Neural Networks, *IEEE Trans. On Systems, Man, and Cybernetics – Part B: Cybernetics*. 27(1): 158-164.

[9] Hsu C. W., Chang C. C. Lin, C. J. 2010. A Practical Guide To Support Vector Classification. *Proceeding of IEEE Intelligent Vehicles Symposium*. 1:1-6.

[10] Keneddy, J. 2013. http://pascal.inrialpes.fr/data/human/

[11] MudjirahardjoP., TanJ.K., KimH., Ishikawa,S.2015. Comparison Of Feature Extraction Methods For Head Recognition. *Proceeding 17th International Electronics Symposium (IES)*.

[12] MudjirahardjoP., Purnomo M. F. E., Hasanah R. N., Suyono H. 2015. Multi Scale Performance Of Feature Extraction For Human Head Recognition. *International Journal of Applied Engineering Research*. 10(17): 37731-37736.