# CS-HOG : Color Similarity-based HOG

Yuhi Goto
Chubu University
Aichi, Japan
Email: yuhi@vision.cs.chubu.ac.jp

Yuji Yamauchi
Chubu University
Aichi, Japan
Email: yuu@vision.cs.chubu.ac.jp

Hironobu Fujiyoshi
Chubu University
Aichi, Japan
Email: hf@cs.chubu.ac.jp

***Abstract*—Conventional object detection methods often use local features based on object shape, of which the HOG feature is typical. In recent years, Color Self-Similarity (CSS) has been proposed as a local feature that uses color information. CSS involves computing color similarity as a basis for deciding the sameness of objects, and thus represent a feature that is effective for object detection. It has also been reported that detection performance can be improved by using the CSS feature together with the HOG feature or other shape-based feature. We propose a Color Similarity-based HOG (CS-HOG) feature that is based on color similarity for object detecting shapes. The CS-HOG feature enables clarification of the object shape by using color similarity to calculate the degree of object sameness, thus achieving highly-accurate object detection. Evaluation experiments show that the CS-HOG feature improves performance from $22.5\%$ and $27.2\%$ compared to the HOG feature and the CSS feature, and by $4.2\%$ compared to the HOG feature and the CSS feature used together.**

## I. Introduction

Object detection is expected to find applications in many fields, including Intelligent Transport Systems that use vehicle-mounted cameras and in the security and marketing fields using surveillance cameras. Many object detection methods extract local features from training samples and use statistical training methods to construct classifier[1]〜[6]. To achieve highly-accurate object detection, it is important to design local features that are captured the features of the object. Many conventional methods use gradients of brightness to understand object shapegroundid for designing local features that represent the features of objects. Such methods include Histograms of Oriented Gradients (HOG)[7], Edge Orientation Histograms (EOH)[8], and Edgelets[9]. Of those, the HOG feature uses the cumulative gradient magnitude in the gradient direction in a local region normalized in the local region, so it is robust to changes in illumination and geometry.

Because features that yield shape information are based on image brightness gradients, color information is not used. in area of object detection, color information is not useful, because the pixel colors are vary by human clothing or vehicle bodies. On the other hand, in areas of object tracking[10][11], person re-identificaton[12][13], Keypoint tracking[15][16] and image recognition[14] color image is effective information. because, the color image includes three times the information that the grayscale image contains. Therefore, one proposed feature that does use color information as a local feature is Color Self-Similarity (CSS)[17], Integral Channel Features[18], which confirms that color information can be effective in object detection. The CSS feature captures color similarity from the color histograms of two local regions as a measure of whether the objects seem to be the same or seem to be different. It thus represents an effective feature for object detection and is therefore an important key to designing local features that effectively represent object features.

We propose here the Color Similarity-based HOG (CS-HOG) as a local feature that effectively represents object features. CS-HOG feature uses color information to clearly capture object shape on the basis of color similarity, which is used to calculate a value that indicates object sameness. For example, when using HOG features to determine shape, processing is done to prevent the reversal of the gradient due to the light-dark relationship between the object region and the background region. With the CS-HOG feature, however, the shapes of the object and background can be obtained on the basis of color similarity, and a shape that represents the object region can be obtained. It is thus possible to obtain gradients that are consistent with the target object. By focusing on the relationship between the local regions for calculating the color similarity and the local region for calculating the shape, it is possible to suppress errors in detection due to shapes that are similar to the target object. In this way, the CS-HOG feature can flexibly capture the shape of the target object and enable diverse feature representation to realize highly accurate object detection.

## II. Conventional method

The general approach to object detection is to extract local features from a training sample and use them to train a recognizer. The HOG feature was proposed as a local feature for capturing object shape by Dalal $et.al.$[7] In conventional object detection methods, HOG features are a typical local feature for capturing object shape from grayscale images, and are the most often used. A different approach is the CSS feature proposed by Walk $et.al.$[17] which uses the color information in color images as a local feature. CSS features focus on color similarity as an effective feature representation for object detection. In this section, we describe the HOG shape feature and the CSS color similarity feature, which are local features that are closely related to our proposed CS-HOG feature.

978-1-4673-5621-3/13/$31.00 ©2013 IEEE

### A. Shape features : HOG

Histogram of Oriented Gradients (HOG) features[7] are histograms of oriented gradients that are created for each local region (cell) in a grayscale input image. Each histogram is normalized to the cell, and the feature is also normalized to multiple-cell block regions, making the feature robust against changes in lighting and geometry.

In the calculations, the input image is first normalized to a pre-defined size, and the gradient magnitude of pixel luminance and the gradient orientation are calculated. The calculated gradient orientation, is an angle from $0°$ to $360°$, but $180°$ is subtracted from calculated values greater than $360°$, so the result is an angle value from $0°$ to $180°$. In this way, we can obtain gradient orientations that are independent of the light-dark relationship of the luminance of the target and the background. Next, we use the calculated gradient magnitude m and gradient orientation to construct a histogram of oriented gradients for cell ($M \times M$ pixels). Creating histograms for each cell in this way yields a feature that is robust to minute geometrical changes within the local region. Finally, we use normalize the histogram of oriented gradients for each cell $c$ for each ($K \times K$cell) block region.

### B. Color similarity features : CSS

The Color Self-Similarity (CSS) feature[17] is a measure of the similarity of the color histograms of two cell regions . This feature captures the sameness or non-sameness with the detection target by measuring the color similarity of two local regions. The use of color similarity as a feature provides the advantage of independence of the color of detection target.

The computational procedure involves normalization of the input image to a pre-defined size and partitioning the image into cells, $c$, of $M \times M$ pixels. Then, color histograms for the cells represented in the HSV color system $\{H$ : hue, $S$ : saturation, $V$ : brightness$\}$ are created. Using the color histograms created from the two cells , the color similarity between the two cells. Calculation of the distance between histograms generally involves taking the intersection of histogram and the Bhattacharyya distance. Our preliminary experiments showed that the Euclidian distance was the most effective (as also reported in[17]), so we adopted that measure in our similarity calculations.

### C. Comparison of HOG features and CSS features

Here we compare the feature representation characteristics of HOG features and CSS features. Visualizations of the HOG and CSS features are compared in Fig. 1. The HOG feature represents object shape with gradients obtained from brightness values within a local region. However, as shown for the detection target in Fig. 1, the brightness of the object and background is sometimes similar in the grayscale image of a subject, but less so in color image of the same scene. The shape of the detection target cannot be clearly captured when calculating the HOG features from such images, but the CSS feature expresses the color similarity between local regions and can therefore use the additional information provided by color images to distinguish object and background. Thus, HOG features can capture object shape locally, but there is a problem with dependence on brightness information. The non-detection region example of Fig. 1 presents an example of when objects that are not the detection target are in the image. In that case, the CSS feature cannot represent the shape of the object in detail, so there may be errors in detection that result from capturing features that are similar to the detection target. Thus, CSS features can distinguish object and background regions by using color image information, but there is the problem of inability to represent object features in detail. We therefore combined these characteristics of HOG features and CSS features to design a local feature that has the advantages of both. In the following sections, we explain that feature, which we call the CS-HOG feature.

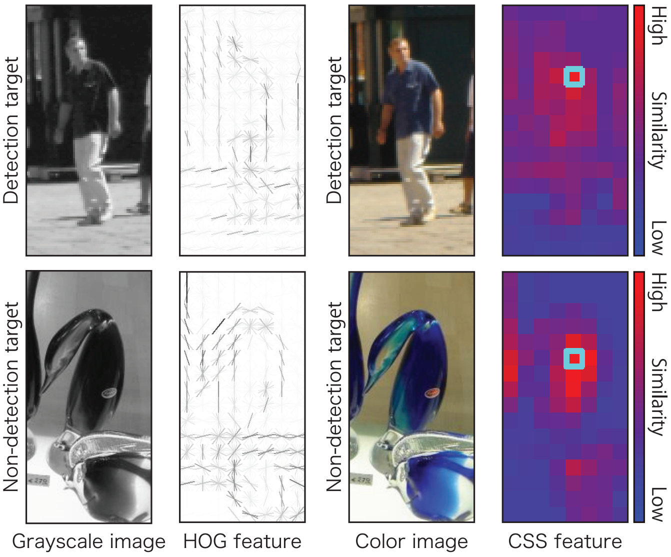


Fig. 1. Comparison of HOG features and CSS features

## III. Shape features based on color similarity : CS-HOG

We designed the Color Similarity-based HOG (CS- HOG) feature to capture shape by using color similarity to measure object sameness, thus clearly representing shape by a boundary between object and background. Because it can effectively represent object features when compared to HOG features, highly accurate object detection is expected. The process of calculating the CS-HOG feature involves first calculating the color similarity between local regions and pixels from the input image and then using the color similarity to calculate the shape feature (Fig. 2) . The following sections explain the computational method in detail.

### A. Calculation of color similarity between local regions and pixels

The procedure for calculating the CS-HOG features begins with normalizing the input image to a pre-defined size and calculating the color similarity for local regions (cells) and pixels. As shown in Fig. 3, Eq. (1) is used to calculate the typical color value for the cell of interest $c'$, $I(c', r)$, from the

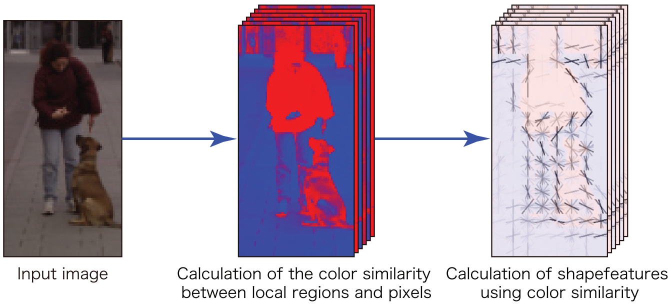

Input image
Calculation of the color similarity between local regions and pixels
Calculation of shapefeatures using color similarity

Fig. 2. Calculation of the CS-HOG feature

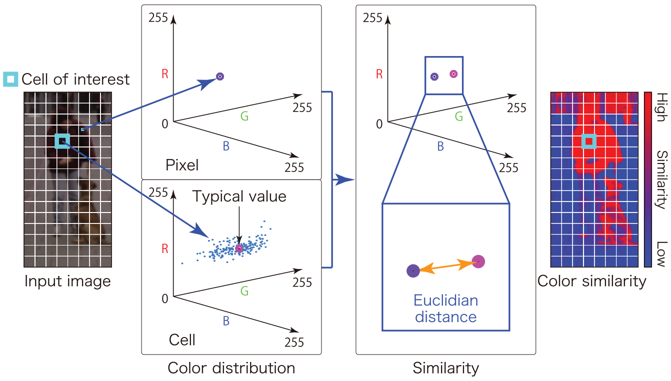


Fig. 3. Calculation of the color similarity between local regions and pixels

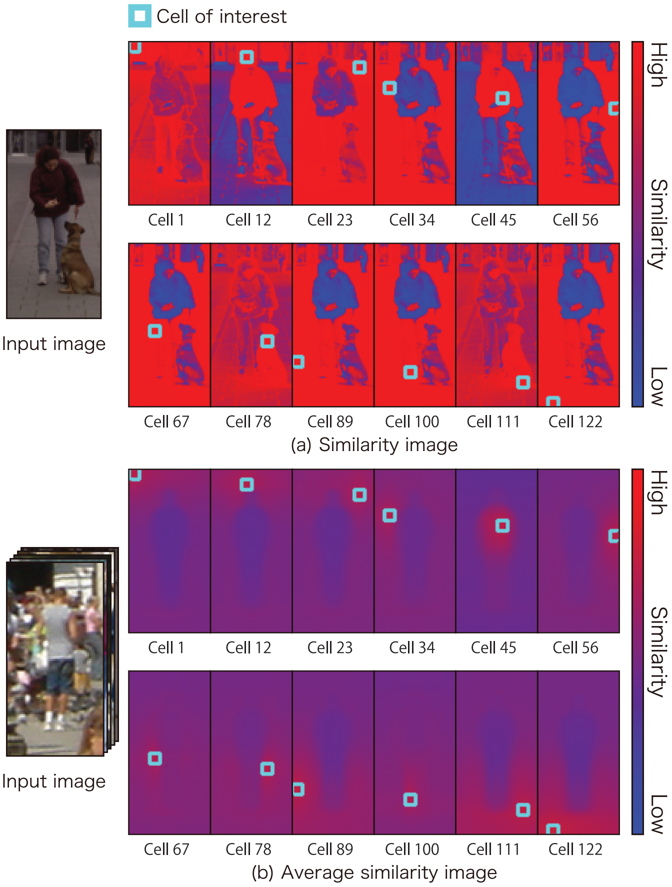
color of each pixel in $c'(M \times M$pixels) represented in the RGB color system $\{R$: red, $G$ : green, $B$ : blue$\}$ as $I_{c'}(x,y,R)$, $I_{c'}(x,y,G)$, $I_{c'}(x,y,B)$.

$$I(c',r) = \frac{1}{M \times M} \sum_{x}^{M} \sum_{y}^{M} I_{c'}(x,y,r) \tag{1}$$

Here, $r$ represents the elements of the RGB color system ($r$=$R,G,B$). The typical value $I(c',r)$ for cell of interest $c'$ and the color of each pixel $I(x,y,r)$ are used to calculate the similarity $s(x,y,c')$ based on the Euclidian distance of Eq. (2) in the same way as for the CSS feature.

$$s(x,y,c') = \sqrt{\sum_{r \in \{R,G,B\}} (I(c',r) - I(x,y,r))^2} \tag{2}$$

The similarity $s(x,y,c')$ is obtained for each cell of interest, $c'$. For a 64 by 128 pixel input image, the number of cells of interest is $(64/M) \times (128/M)$. Thus, when $M = 8$, 128 patterns of similarity are obtained for each pixel (referred to here as the similarity image for the cell of interest $c'$). An example of a similarity image created from a detection object image is shown in Fig. 4(a). As we can see in that figure, the color similarity can represent the quality of region sameness in the same way as does the CSS feature when the cell of interest $c'$ is selected in the object region (cell 45). Furthermore, the image confirms that the similarity with the object region is low when the cell of interest is in the background region (cell 89). As we can see from the average similarity image created from 2,416 detection target training images shown in Fig. 4(b), the same object region similarity is high for the entire detection target and low for the non-object region. The CS-HOG feature uses these similarity image features as a basis for detecting object shape.



(a) Similarity image

(b) Average similarity image

Fig. 4. Visualization of similarity images

### B. Using the HSV color system

The HSV color system has been reported as the most effective for use in calculating the color similarity[17]. The CSS feature calculates distance in the HSV color system by quantizing each element and creating histograms. In the calculation of color similarity between local regions and pixels for the CS-HOG feature, however, the HSV color system cannot be used to obtain the similarity values by the method described above. The HSV color system comprises the elements $H$ (hue), $S$ (saturation), and $V$ (brightness) , for which the coordinate system origins differ greatly, unlike in the RGB color system. For that reason, each element the distance cannot be calculated using each HSV element. To enable distance calculation using the HSV color system, we convert from each element$\{H,S,V\}$ to a common coordinate system $\{u,t,V\}$ with Eq. (3).

$$\begin{cases} u = S \times \cos H \\ t = S \times \sin H \\ V = V \end{cases} \tag{3}$$

In this way, we can represent the HSV values in a single coordinate system in the same way as using the RGB color system. The CS-HOG feature first converts the HSV color system to coordinate system $\{u,t,V\}$ and then performs the

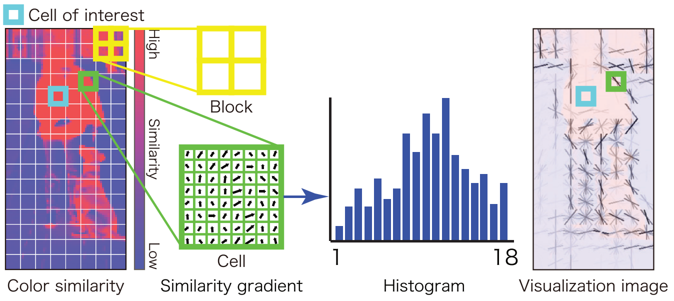


Fig. 5. Color Similarity-based HOG

distance calculation to obtain the color similarity values.

### C. Using color similarity to calculate shape features

The CS-HOG feature uses similarity images constructed from color similarity between local regions and pixels to obtain shape features (Fig. 5). The shape features are histograms of pixel gradients created for each cell $c$ using similarity images for each cell of interest $c'$ in the same way as for the HOG features. The computational procedure begins with using the similarity $s(x, y, c')$ calculated from the cell of interest $c'$ to calculate the gradient magnitude $m$ and gradient orientation $\theta$ for each pixel by using Eqs. (4) through (6).

$$m(x, y, c') = \sqrt{s_x(x, y, c')^2 + s_y(x, y, c')^2} \tag{4}$$

$$\theta(x, y, c') = \tan^{-1} \frac{s_x(x, y, c')}{s_y(x, y, c')} \tag{5}$$

$$\begin{cases} s_x(x, y, c') = s(x+1, y, c') - s(x-1, y, c') \\ s_y(x, y, c') = s(x, y+1, c') - s(x, y, -1, c') \end{cases} \tag{6}$$

To prevent the reversal of the gradient orientation $\theta'$ due to light-dark relationship of the brightness values, 180° is subtracted from gradient orientation $\theta$ when the value of $\theta$ exceeds 180° in calculating the HOG feature. In the proposed CS-HOG feature, the object-sameness of the cell of interest $c'$ is represented by color similarity, so it is possible to use the entire range of gradient orientations $\theta$ from 0° to 360°. Next, the gradient magnitude $m$ and gradient orientation $\theta$ calculated from the similarity image of the cell of interest $c'$ and Eq. (7) are used to create the histograms of oriented gradients $\boldsymbol{V_c}(c')$=$\{v_c(1, c'), v_c(2, c'), ...v_c(N, c')\}$ for cell $c$ ($M \times M$ pixels).

$$v_c(f(\theta), c') = \sum_{x}^{M} \sum_{y}^{M} \ m(x, y, c') \delta[\theta, f(\theta)] \tag{7}$$

Here, $f(\theta')$ represents the gradient orientation $\theta'$ quantized in the $N$ orientation. In the CS-HOG feature, a histogram of oriented gradients $\boldsymbol{V_c}(c')$ is created for each cell of interest $c'$, so for a 64 by 128 pixel input image with $M = 8$, 128 patterns of histograms of oriented gradients $\boldsymbol{V_c}(c')$ are created for one cell $c$. Finally, Eq. (8) is used to normalize the histograms of oriented gradients $\boldsymbol{V_c}(c')$ for each cell of interest $c'$ and for each $K \times K$ cell block region in the same way as for the HOG feature.

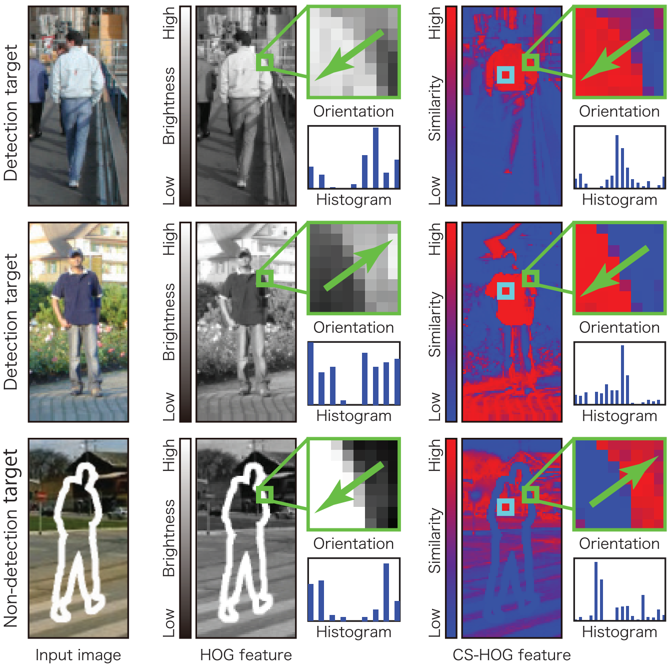


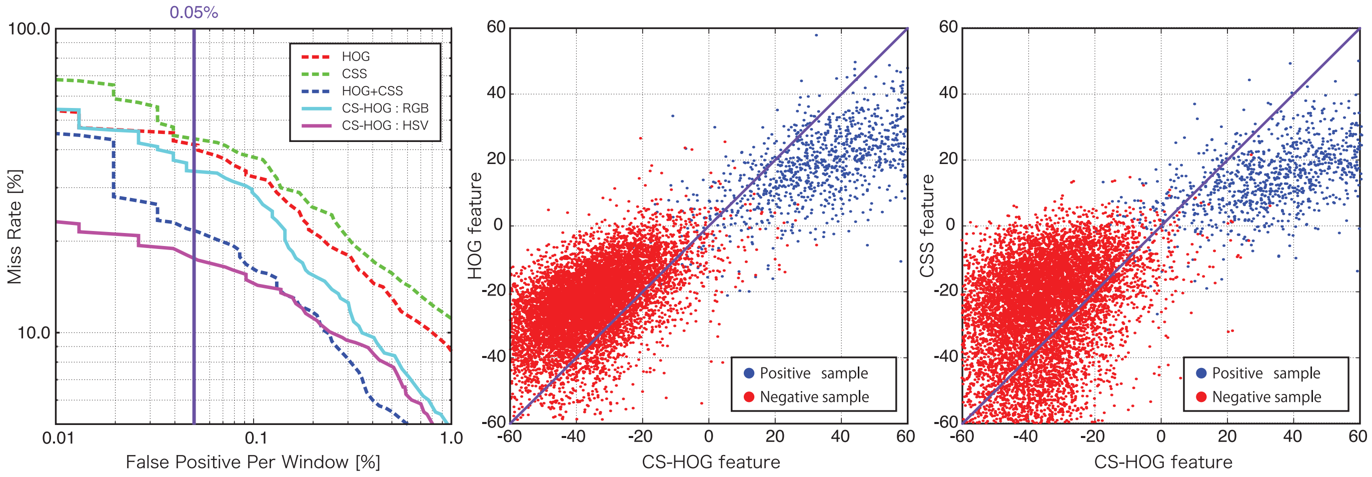
Fig. 6. Comparison of HOG features and CS-HOG features

$$v_c(n, c') = \frac{v_c(n, c')}{\sqrt{\sum_{k=1}^{b} v_c(k, c')^2 + \epsilon}} \qquad (\epsilon = 1) \tag{8}$$

Here, b is the number of gradient orientations in a block ($K \times K \times N$). The normalized histograms of oriented gradients $\boldsymbol{V_c}(c')$ are then $\boldsymbol{V_c}(c')$=$\{v_c(1, c'),$ $v_c(2, c'), ...v_c(b, c')\}$. The CS-HOG feature quantities are the values for the various quantization numbers obtained in the processing that uses Eqs. (4) to (8) to create the histograms of oriented gradients $\boldsymbol{V_c}(c')$ from the similarity images created from all of the cells of interest $c'$. Because the CS-HOG feature results from similarity images created for 128 patterns of cells of interest and 105 rounds of normalization for a 64 by 128 pixel input image with $M = 8, \ K = 2, \ N = 18$, the feature has $128 \times 2 \times 2 \times 18 \times 105 = 967,680$ dimensions.

### D. Comparison of HOG and CS-HOG

Here, we compare the feature representation characteristics of HOG features and CS-HOG features. Visualizations of HOG features and CS-HOG features are presented in Fig. 6(a). As we see from the two detection targets in the figure, the HOG features are limited to gradient orientations in the range from 0° to 180° to prevent reversal of the gradient orientation due to the light-dark relationship of brightness values of the object and background regions. As we can confirm from the histograms of oriented gradients (the histograms in the figure), the limit on gradient orientation makes it possible to capture features consistent with the detection target. On the other hand, the CS-HOG features use color similarity to represent object-sameness, so there is no need to limit the gradient orientation and all gradient orientations can be represented.



(a) DET curve

(b) Recognizer output

Fig. 7. Comparison of classification

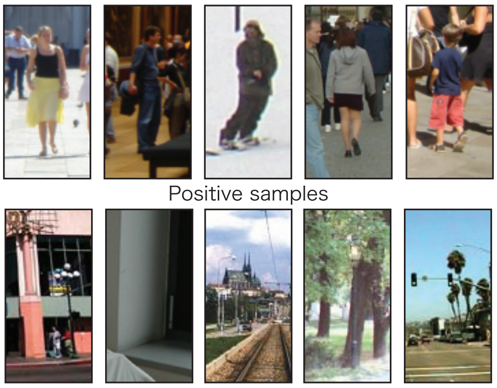


Fig. 8. Examples from the INRIA Person Dataset

Thus, feature representation is more diverse than with the HOG feature, and flexible capture of object shape should be possible. The non-detection target region in Fig. 6(b) is shown with a shape detection target artificially superimposed on the background. In that case, the HOG feature captures the similar shape because only local-region shape is detected, as we can see from the histogram of oriented gradients. That raises the possibility of misdetection on the basis of shape. The CS-HOG feature, however, uses local regions selected as cells of interest and captures the object region on the basis of an object sameness measure, which has the result of suppressing such effects. By obtaining shape features based on color similarity in this way, the CS-HOG feature is able to capture object shape flexibly with diverse feature representation and can therefore be expected to achieve highly accurate object detection.

## IV. Evaluation experiments

To test the effectiveness of the CS-HOG in representing features, we conducted experiments to evaluate the detection of humans in images.

### A. Data set

We used the INRIA Person Dataset[7], a data set in which the object regions have been determined in advance and which is widely-used as a benchmark for detection of humans. Examples from the INRIA Person Dataset for the detection target (positive samples) and non-detection target (negative samples) are shown in Fig. 8. The training sample contained 2,416 positive samples and 13,161 negative samples. The evaluation sample contained 1,126 positive samples. The negative samples were comprehensively raster-scanned from 453 background images. The images in both the training sample and the evaluation sample were normalized to a size of 64 by 128 pixels.

### B. Overview of the experiments

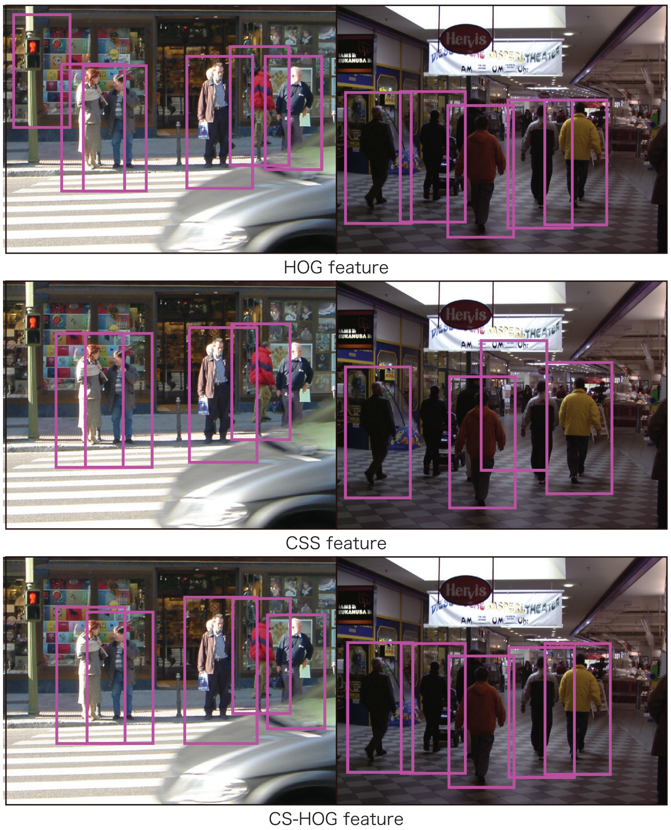
To demonstrate the effectiveness of the CS-HOG feature, we compared the discrimination accuracy using the following local features.

- HOG feature (HOG)
- CSS feature (CSS)
- Both HOG features and CSS features (HOG +CSS)
- CS-HOG feature : RGB color system (CS-HOG: RGB)
- CS-HOG feature : HSV color system (CS-HOG: HSV)

The CS-HOG feature can use either the RGB color system or the HSV color system to obtain color similarity values. The HOG feature plus CSS feature case uses a recognizer that was trained with both feature dimensions simultaneously. The feature parameters are cell size $M = 8$ pixels, block size $K = 2$ cells, and the quantization numbers $N = 18$. The method used to compare with the CS-HOG feature was to train a recognizer with Real AdaBoost[20]. In addition, the evaluation experiments were conducted with 500 rounds of Real AdaBoost training of a weak recognizer. The Detection Error Trade-off (DET) curve was used for the evaluation. The DET curve is plotted with the detection error rate on the horizontal axis and the detection failure rate on the vertical axis; values closer to the origin represent higher accuracy.

### C. Experimental results

The DET curves for the results of the experiments (Fig. 7(a)) show that the recognition accuracy is improved by from 5.7%



HOG feature

CSS feature

CS-HOG feature

Fig. 9. Example for human detection

and $8.1\%$ for CS-HOG:RGB compared to HOG and CSS for a detection error rate of $0.05\%$ Comparing CS-HOG:HSV with HOG and CSS, the improvement is from 22.5 to 27.2 and with CS-HOG:RGB the improvement is $16.8\%$. These results confirm that the CS-HOG feature that uses the HSV color system to calculate the similarity values is an effective representation. Furthermore, CS-HOG:HSV improved recognition accuracy by $4.2\%$ compared to HOG+CSS. These results show that color similarity and shape features work effectively together as a single feature. We checked how the output values of the recognizer varied for the CS-HOG feature the HOG feature, and the CSS feature. The distributions of the recognizer output values for the evaluation sample are shown in Fig. 7(b), where the horizontal axes are the recognizer output for the CS-HOG feature and the respective vertical axes are the output values for the HOG feature and the CSS feature. The diagonal line represents equal outputs. We can see that the CS-HOG feature output is higher than the output for the HOG feature and the CSS feature for from $86.7\%$ and $88.1\%$ of the positive samples and lower for from $67.2\%$ and $86.7\%$ of the negative samples. These results confirm that the CS-HOG feature output tends to be more correct than the output for the HOG feature or the CSS feature for all of the samples. Accordingly, the CS-HOG feature detects shape on the basis of color similarity and achieves highly accurate object detection by effectively capturing the features of the object, as we can also see from the human detection example of Fig. 9.

## V. Conclusion

We have proposed the CS-HOG feature, which uses a measure of color similarity to detect shape. This feature can capture shape with more flexibility than the HOG feature by calculating shape on the basis of the color similarity information in color images. In future work, we will investigate the effectiveness of the CS-HOG feature for detection of automobiles and people.

## References

[1] N. Dalal, B. Triggs, and C. Schmid “Human detection using oriented histograms of flow and appearance”, European Conference on Computer Vision, 2006.

[2] Y. D. Mu, S. C. Yan, Y. Liu, T. Hunag, and B. F. Zhou, “Discriminative local binary patterns for human detection in personal album”, Computer Vision and Pattern Recognition, pp.1-8, 2008.

[3] W.R. Schwartz A. Kembhavi D. Harwood and L.S. Davis, “Human Detection Using Partial Least Squares Analysis”, International Conference on Computer Vision, 2009.

[4] T. Watanabe, S. Ito, and K. Yokoi, “Co-occurrence histograms of oriented gradients for pedestrian detection”, Advances in Image and Video Technology, pp.37-47, 2009.

[5] X.Wang, T. X. Han, and S. Yan, “An HOG-LBP human detector with partial occlusion handling”, AInternational Conferenceon Computer Vision, pp.32-39, 2009.

[6] C.Wojek, S. Walk, S. Roth, and B. Schiele, “Monocular 3D scene understanding with explicit occlusion reasoning”, Computer Vision and Pattern Recognition, pp.1933-2000, 2011.

[7] N. Dalal, and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, Computer Vision and Pattern Recognition, vol.1, pp.886-893, 2005.

[8] K. Levi, and Y. Weiss, “Learning Object Detection from a Small Number of Examples: the Importance of Good Features”, Computer Vision and Pattern Recognition, vol.2, pp.53-60, 2004.

[9] B. Wu, and R. Nevatia, “Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors”, International Conferenceon Computer Vision, vol.1, pp.90-97, 2005.

[10] B. Wu, and R. Nevatia, “Condensation conditional density propagation for visual tracking”, International Journal of Computer Vision, vol.29, no.1, pp.5-28, 1998.

[11] D. Comanisiu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects using Mean Shift”, Computer Vision and Pattern Recognition, pp.142-149, 2000.

[12] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, “Shape and Appearance Context Modeling”, International Conferenceon Computer Vision, pp.1-8, 2007.

[13] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person Re-Identification by Symmetry-Driven Accumulation of Local Features”, Computer Vision and Pattern Recognition, pp.2360-2367, 2010.

[14] J. Shotton, M. Johnson and R. Cipolla, “Semantic Texton Forests for Image Categorization and Segmentation”, Computer Vision and Pattern Recognition, pp.1-8, 2008.

[15] V. Lepetit and P. Fua, “Keypoint recognition using randomized trees”, Pattern Analysis and Machine Intelligence, vol.28, no.9, pp.165-1479, 2006.

[16] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, “Fast Key- point Recognition using Random Ferns”, Pattern Analysis and Machine Intelligence, vol.32, no.3, pp.448-461, 2010.

[17] S. Walk, and N. Majer, “New Features and Insights for Pedestrian Detection”, Computer Vision and Pattern Recognition, pp.1030-1037, 2010.

[18] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral Channel Features”, British Machine Vision Conference, 2009.

[19] Y. Freund, and R. E. Schapire, “Experiments with a new boosting algorithm”, International Conference on Machine Learning, pp.148-156, 1996.

[20] R. E. Schapire, and Y. Singer, “Improved Boosting Algorithms Using Confidence-rated Predictions”, Machine Learning, vol.37, no.3, pp.297-336, 1999.