Relational HOG Feature with Wild-Card for Object Detection

Yuji Yamauchi[1], Chika Matsushima[1], Takayoshi Yamashita[2], Hironobu Fujiyoshi[1]
[1]Chubu University, Japan, [2]OMRON Corporation, Japan
{yuu, matsu}@vision.cs.chubu.ac.jp, takayosi@omm.ncl.omron.co.jp, hf@cs.chubu.ac.jp

Abstract

This paper proposes Relational HOG (R-HOG) features for object detection, and binary selection by using a wild-card "$*$" with Real AdaBoost. HOG features are effective for object detection, but their focus on local regions makes them high-dimensional features. To reduce the memory required for the HOG features, this paper proposes a new feature, R-HOG, which creates binary patterns from the HOG features extracted from two local regions. This approach enables the created binary patterns to reflect the relationships between local regions. Furthermore, we extend the R-HOG features by shifting the gradient orientations. These shifted Relational HOG (SR-HOG) features make it possible to clarify the size relationships of the HOG features. However, since R-HOG and SR-HOG features contain binary values not needed for classification, we have added a process to the Real AdaBoost learning algorithm in which "$*$" permits either of the two binary values "0" and "1", and so valid binary values can be selected. Evaluation experiment demonstrated that the SR-HOG features introducing "$*$" offers better detection performance than the conventional method (HOG feature) despite the reduced memory requirements.

1. Introduction

With the increasing use of digital cameras and vehicle-mounted cameras, the expectations for practical detection of humans in image for the purposes of improving image quality and assisting the drivers of vehicles are also rising. Research on the use of Field Programmable Gate Arrays (FPGA) or other such hardware implementations of that function has been done [12, 3, 6]. In hardware implementations, it is important that the detection method can operate with high accuracy, high-speed and low memory requirements.

Most detection methods proposed in recent years use combinations of local features of images and stochastic learning [2, 18, 7, 15, 17]. Local region gradients [5, 1], which are used as features in many proposals, can capture the shape of an object, but very many dimensions are required to obtain the features of each local region. The difficulty of accomplishing that with small-scale hardware that has limited memory is a major problem whose solution requires a reduction in the amount of feature data. Less data has two benefits. One is that less memory is needed and the other is that features can be categorical, each representing common properties.

Two approaches to reducing the amount of data can be considered: compressing the feature space to reduce the number of features and reducing the amount of data needed for each feature. The former approach includes methods such as vector quantization to reduce the number of features [9] and principle component analysis to compress the feature dimensions. These methods can retain the original amount of data while reducing the number of feature dimensions. Human detection, however, involves the processing of a huge number of detection windows, so these methods are very inefficient.

The latter approach involves quantizing features at a low bit rate. Scalar quantization, for example, can represent the feature data at a bit rate that fits the problem. Quantization is also an effective way to reduce the amount of data. In addition to representing the information with the minimum amount of data, it has the advantages of being robust against noise and easy to use. One method of quantization is threshold processing, which is simple and has the advantage of low computational cost. However, determining the optimum threshold for many samples is difficult. Another binarization method uses the size relationship. The Local Binary Pattern (LBP) proposed by Ojala $etal.$ [13, 16, 11] and a method that expands on that [4] have the advantage of not requiring a threshold, as binarization is based on comparison of the two values. Threshold binarization and size relationship binarization also differ in the data contained in one binary value. In threshold processing, the value represents only size, but when size relationship is used, the relation between two values is also included.

Our method focuses on binarization using the size relationship, which is one of the latter methods of reducing data quantity. To achieve highly accurate object detection

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00

while reducing the amount of feature data, we propose the Relational HOG (R-HOG) feature, a binarization method in which the size relationship is obtained by comparing HOG features from two local regions. Since R-HOG features uses the size relationship of two HOG features, they can represent the relatedness of local regions. However, they combine multiple binary values, and so contain values that are unnecessary for classification. We solved that problem by introducing a wild-card "$*$", which permits either of the two binary values of "0" and "1", when training. That makes it possible to select the binary values that are effective for classification by Real AdaBoost.

2. Related works

Local Binary Patterns (LBP) are a technique that is being applied in a variety of fields such as object detection, face recognition, and action recognition. The LBP features proposed by Ojala et al. [13] represent the magnitude relations of a pixel to adjacent pixels with a code, thus allowing representations of fixed shapes such as edges. Liao et al. proposed a method for application to face recognition in which LBP is extended to represent magnitude relations between the average luminance values in multi-resolution block areas [8]. Another technique applied in face recognition[14] and action recognition [20] is the Local Ternary Pattern (LTP), which extends LBP by using three values for the threshold. Although that method can capture the relations of nearby and adjacent pixels, it cannot represent other effective combinations that exist. Furthermore, the LTP approach requires optimum values for the three thresholds.

Methods that combine multiple feature quantities include the Joint Haar-like features proposed by Mita et al., which capture the relations of Haar-like features in different positions [10], and co-occurrence features that link the output values of weak classifiers with operators proposed by Yamauchi et al. [19]. These methods combine feature quantities on the basis of recognition results, so the combination of features may be negatively affected when the results are in error or when the target of detection is obscured.

In contrast to those methods, the features that we propose capture the relations of local gradients, and can thus capture the relationships with all areas rather than simply neighboring areas as in LBP and LTP. Furthermore, using the wild-card character "$*$" has a masking effect in which binaries that are not needed for distinguishing the target are not observed. It is therefore possible to use only the binaries that are effective for discrimination, and approach that can be expected to suppress degradation of recognition accuracy.

3. HOG features and binarization

This section describes HOG features and binarization as means of reducing the amount of HOG feature data.

3.1. HOG features

The Histograms of Oriented Gradients (HOG) feature proposed by Dalal $etal.$ [2] is a one-dimensional histogram of gradient orientations of intensity in local regions that can represent object shape. This feature is a histogram of adjacent pixel gradients for local regions, so it is not easily affected by local lighting conditions and is robust to changes in geometry.

To compute HOG features, first, the magnitude m and gradient orientation θ are calculated from the intensity I of the pixels. Next, using the calculated magnitude m and gradient orientation θ, the sum of the magnitudes of quantized gradient orientation θ' in cell region c ($p \times p$ pixels) are calculated. We represent the set of sums of magnitude in gradient orientation θ' as the N-orientation histogram $\boldsymbol{V}_c = \{v_c(1), v_c(2), \cdots, v_c(N)\}$. Finally, we use Eq. (1) to normalize the histogram by each block region ($q \times q$ cells) to extract the features.

$$v'_c(n) = \frac{v_c(n)}{\sqrt{\left(\sum_{k=1}^{q \times q \times N} v_c(k)^2\right) + \epsilon}} \quad (\epsilon = 1) \qquad (1)$$

After normalization, the histogram $\boldsymbol{V}'_c$ is $\boldsymbol{V}'_c = \{v'_c(1), v'_c(2), \cdots, v'_c(B \times N)\}$. Here, B is the number of cell regions that are contained in the block region.

3.2. Binarized HOG features (B-HOG)

Binarized HOG (B-HOG) features are obtained by thresholding. Those features can capture the relatedness of the gradient orientations within the cell region by observing the binary values for N orientations in the cell region as a single feature (binary pattern).

We use the eight-orientation histogram $\boldsymbol{V}'_c$ for the cell region subjected to threshold processing as shown in Eq. (2) to produce the B-HOG features $\boldsymbol{B}_c^{BH} = \{b_c^{BH}(1), b_c^{BH}(2), \cdots, b_c^{BH}(8)\}$. In reference [2], nine orientations are used, but in our work we choose eight orientations so that features can be represented as one byte.

$$b_c^{BH}(n) = \begin{cases} 1 & \text{if } v'_c(n) \geq t \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Here, the t represents the threshold. For example, when extracting HOG features for an input image such as Fig. 1 and binarizing the features, we get $\boldsymbol{B}_c^{BH} = (00001011)_2$.

3.3. Benefits and problems with B-HOG features

B-HOG features and HOG features vary with the amount of feature data. The HOG features obtained with Eq. (1)

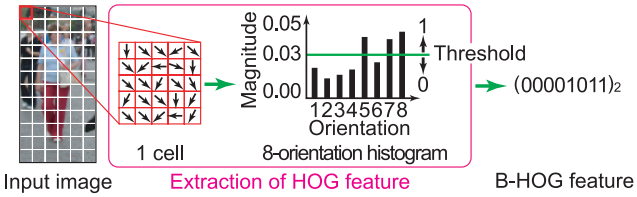


Figure 1. B-HOG feature calculation method.

must usually be represented by double precision real numbers (8 bytes), but the B-HOG features can be represented by one unsigned character (1 byte). Thus, B-HOG features can reduce memory use to $1/8$ that required by HOG features. However, the need to obtain the optimum binarization threshold values t for different human detection environments is a problem.

4. Proposed method

In this section, we first describe Relational HOG (R-HOG) features and Shifted Relational HOG (SR-HOG) features. To solve the problems described in section 3.3 while retaining the advantages of quantization, we first binarize by using the size relation of the HOG features extracted from two local regions. The R-HOG and SR-HOG features can capture the relatedness of local regions, but they contain binary values that are unnecessary for classification. Therefore, we introduce a wild-card "$*$", which permits either of the two binary values "0" and "1" in the training to allow the selection of the binary values that are effective in classification by Real AdaBoost.

4.1. Relational HOG features (R-HOG)

R-HOG features are binarized by comparing the two values of HOG features obtained from two local regions as shown in Fig. 2, thus reducing data quantity by eliminating use of a threshold. While B-HOG features can represent gradient magnitudes only as binary values, R-HOG features can also represent the relationship between two features. Furthermore, R-HOG feature binarizes the size relation of HOG features, so processing to normalize the HOG is not needed. Because the normalization processing has the highest computational cost of the HOG feature processing, the proposed method can greatly reduce the processing cost.

R-HOG features are the binarized feature quantities $\boldsymbol{B}^{RH}_{c_1c_2} = \{b^{RH}_{c_1c_2}(1), b^{RH}_{c_1c_2}(2), \cdots, b^{RH}_{c_1c_2}(8)\}$ that result from comparing the size relationship of the eight-orientation histograms $\boldsymbol{V}_{c_1}$ and $\boldsymbol{V}_{c_2}$ obtained from two cell regions c_1 and c_2, as shown in Eq. (3).

$$b^{RH}_{c_1c_2}(n) = \begin{cases} 1 & \text{if } v_{c_1}(n) \geq v_{c_2}(n) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Note that $v_c(n)$ does not require a normalization process. As we see in Fig. 2, we can create a binary pattern that

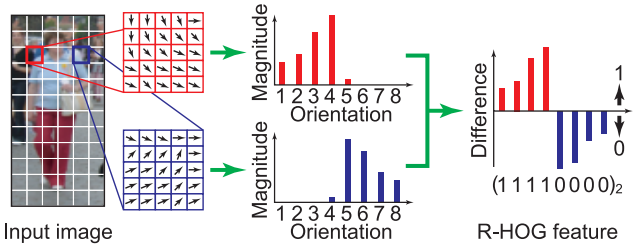


Figure 2. Binarization using HOG features of two cell regions.

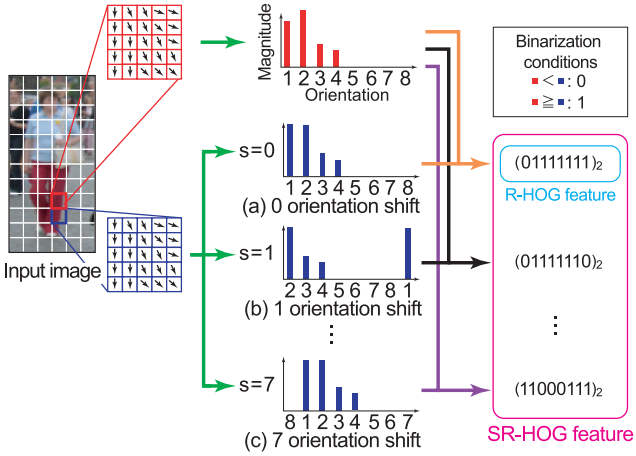


Figure 3. Introducing a shift in the orientation.

captures the relatedness of local regions by using the size relationship of features in two cell regions. In doing so, the R-HOG features are computed from all combinations of cell regions. However, as shown in Fig. 3, if the extracted features are similar, their size relation is not distinct, and so is difficult to represent clearly as binary values.

4.2. Shifted Relational HOG features (SR-HOG)

To solve the problems associated with R-HOG features, as shown in Fig. 3 (b) and (c), we shift the orientation of the eight-orientation histogram $\boldsymbol{V}_{c_2}$ extracted from one of the cell regions by $s(s = 0, 1, 2, \cdots, 7)$ to create the eight histograms $\boldsymbol{V}_{c_2 s}$. Then, we use Eq. (4) in the same way as Eq. (3) to obtain the size relationship and calculate the eight binarized features, $\boldsymbol{B}^{SRH}_{c_1c_2s}$.

$$b^{SRH}_{c_1c_2}(n, s) = \begin{cases} 1 & \text{if } v_{c_1}(n) \geq v_{c_2}((n+s)\%8) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Here, $\%$ is the modulo operator. By calculating the size relationship with the orientation-shifted histograms, the size relationship can be represented clearly even if the extracted features are similar. In this paper, we refer to the R-HOG features extracted with orientation shifting as Shifted Relational HOG (SR-HOG) features.

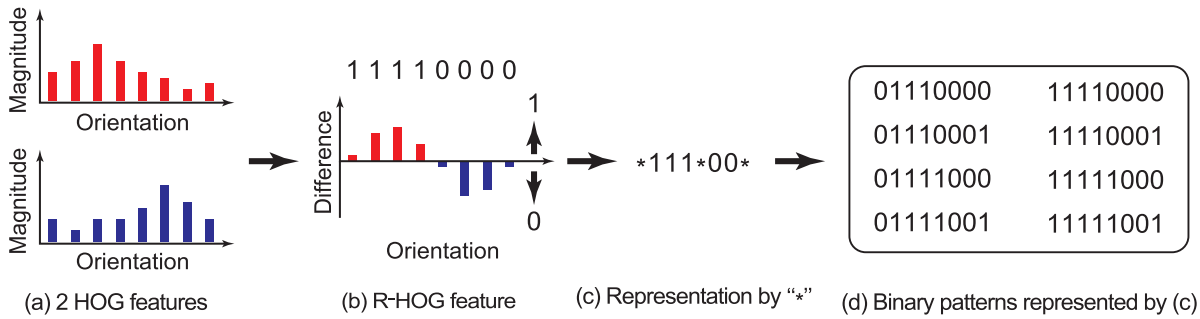


Figure 4. Example of a representation by binary patterns that include "$*$".

4.3. Binary selection by using a wild-card "$*$"

After extracting the R-HOG or SR-HOG features, training by the Real AdaBoost is applied. Introducing "$*$" into features that have been converted to binary patterns to select binary patterns that are effective in classification at the same time as selecting the position of cells and the binary patterns that are effective in discrimination is expected to increase the detection accuracy.

4.3.1 Introduction of a wild-card "$*$"

The proposed features are obtained as binary patterns (Fig. 4(b)) from two HOG features as shown in Fig. 4(a). However, when the difference in compared magnitude is small, the binary values may be easily reversed. Such binary values have low reliability, and so are a cause of error in classification. For that reason, we propose here a binary value selection method in which a wild-card "$*$" is introduced as shown in Fig. 4(c). The "$*$" permits either of the binary values "0" or "1". Doing so makes it possible to express multiple similar binary patterns simultaneously, as shown in Fig. 4(d). The number of "$*$" and the binary bits at which they are used are selected by Real AdaBoost. Then, for each combination of cell regions, 6,561 ($= 3^8$) binary patterns that include "$*$", are generated as weak classifier candidates, as shown in Fig. 5.

For example, if there are 8×16 cells in one detection window, then there are $8,128 (=_{128} C_2)$ cell combination patterns and 6,561 patterns of binary values plus "$*$" for each pair of cells, which gives about 50 million weak classifier candidates. These weak classifier candidates are used for training with Real AdaBoost.

4.3.2 Training method

The proposed training method involves using Real AdaBoost to select weak classifiers effective for classification from among the many weak classifier candidates described in section 4.3.1. The training process is shown in Fig. 6.

First, labeled training samples $(x_1, y_1), \cdots, (x_I, y_I)$ are prepared. The x_i are images and y_i are class labels. The

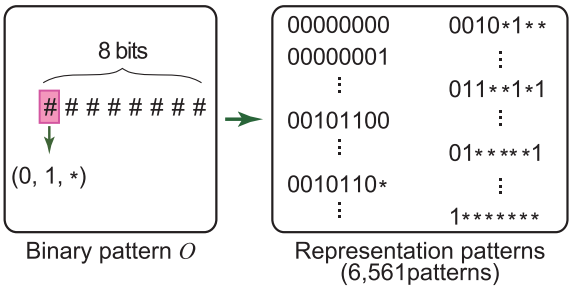


Figure 5. Patterns using wild-card "$*$" in a pair of cell regions.

class that is the target of detection is $y_i = +1$ and the class that is not the target of detection is $y_i = -1$. The sample weights are denoted as $D_l(i)$ and are initialized with Eq. (5).

Next, the processing up to the updating of the training sample weights from calculations of the sample weight frequencies is repeated for a certain number of weak classifiers, T, or until a certain detection rate is attained. First, the probability density functions W_+ and W_- that represent the frequency with which the binary patterns $F(x)$ extracted from all combinations of cell regions, and the patterns of "0", "1", and "$*$", o_r, are the same are calculated from Eq. (6) and Eq. (7) as shown in Fig. 7. $F(x)$ is a function of binary patterns from input image x; o_r are patterns composed of "0", "1", and "$*$" that exist in combinations of two cell regions, $r(\in R = \{c_i, c_j\}_{i=1,2,\cdots,127, j=i+1,i+2,\cdots,128})$, as shown in Fig. 5. The $+$ symbol represents the class of detection targets; the $-$ represents the class of non-targets. The δ is Kronecker's delta function, which returns a value of 1 when the two input binary patterns are the same. Here, if the features used in training are SR-HOG features, then in addition to the combination of two cell regions, r, the number of shifted orientations is also taken into account.

After calculating the probability density functions W_+ and W_-, Eq. (8) is used to calculate score Z, which represents the degree of separation. A larger Z value means a greater frequency difference between the positive sample and the negative sample. Thus, from among the binary patterns o_r, the weak classifier candidates for which the equiv-

1, Input: Labeled training samples I.
$\{x_i, y_i\}_{i=1\cdots I}, y_i \in \{-1, 1\}$

2, Initialization: Initialization of sample weights.

$$D_1(i) = 1/I \tag{5}$$

3, Training:

For $l = 1, \cdots, L$ //Number of weak classifiers.

For $r = 1, \cdots, R$ //Combination number of two cell regions.

For $o = 1, \cdots, O$ //Binary pattern with wild-card "$*$".

· Calculate the probability density functions.

$$W_+ = \sum_{i:y_i=+1} D_l(i)\delta[F(x_i), o_r] \tag{6}$$

$$W_- = \sum_{i:y_i=-1} D_l(i)\delta[F(x_i), o_r] \tag{7}$$

· Calculate the score Z.

$$Z_{o_r} = |W_+ - W_-| \tag{8}$$

End for

End for

· Select weak classifier h_l.

$$h_l = \arg\max_{o_r \in (O \times R)} Z_{o_r} \tag{9}$$

· Update sample weights.

$$D_{l+1}(i) = D_l(i) \exp\left[-y_i h_l(x_i)\right] \tag{10}$$

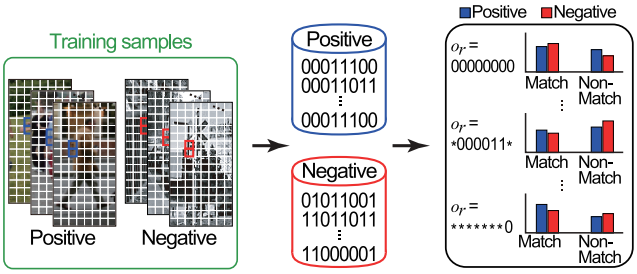
$$h(x_i) = \begin{cases} \frac{1}{2} \ln \frac{W_+ + \epsilon}{W_- + \epsilon} & \text{if } F(x_i) = o_r \\ \frac{1}{2} \ln \frac{(1-W_+) + \epsilon}{(1-W_-) + \epsilon} & \text{otherwise} \end{cases} \tag{11}$$

End for

4, Output: Strong classifier.

$$H(x) = \text{sign}\left[\sum_{l=1}^{L} h_l(x)\right] \tag{12}$$

Figure 6. Training algorithm.



(a) Cell regions: c_1, c_2 (b) Binarization: $F(x)$ (c) Probability density functions

Figure 7. Calculation of the probability density functions.

alent value Z from Eq. (9) is maximum are selected as the effective weak classifiers for round l, h_l.

After selection of the weak classifiers, Eq. (10) is used to update the training sample weights so that the training samples that failed to classify will classify correctly in the next round. Then, the probability density functions of the selected weak classifier sample, W_+ and W_-, are used to calculate the weak classifier output $h_l(x)$ from Eq. (9). Here, the W_+ and $(1 - W_+)$ and the W_- and $(1 - W_-)$ are normalized to the value 1, and ϵ is a coefficient for preventing the denominator from taking a value of 0 ($\epsilon = 1/I$).

Figure 8. Training datasets.

Finally, the processing up to this point is repeated for a certain number of weak classifiers or until a certain detection rate is obtained to construct the strong classifier $H(x)$ in Eq. (12).

5. Evaluation experiment

To evaluate the effectiveness of the proposed method, we conducted two experiments. First, we evaluate the effectiveness of R-HOG and SR-HOG features comparing with conventional HOG features. Next, we compared the accuracy before and after introduction of the wild-card "$*$" to SR-HOG features to evaluate the effectiveness of using "$*$".

5.1. Data sets

In the experiments, we used datasets for people, and vehicles. The people dataset is the INRIA Person Dataset [2]; the vehicle dataset is the one used in reference [19]. Each dataset is shown in Fig. 8. The training sample from the INRIA Person Dataset includes 2,416 positive samples and 12,180 negative samples; the test sample includes 1,126 positive samples and 453 negative samples. The negative test sample includes background images that contain no people. The vehicle dataset training sample included 710 positive samples and 8,800 negative samples; the test sample included 1,230 positive samples and 3,880 negative samples.

5.2. Overview of the evaluation experiment

The evaluation experiments compared the methods listed below.

- HOG features (HOG)

Dataset	Image size [pix.]	Cell size [pix.]	Block size [cell]	Orientation
INRIA[2]	64×128	8×8	2×2	8
Vehicle[19]	72×54	9×9	2×2	8

Table 1. Parameters used in the experiment for each dataset.

- Binarized HOG features (B-HOG)
- Relational HOG features (R-HOG)
- R-HOG features + shifted-orientation (SR-HOG)

The various dataset parameters are presented in Table 1. For the evaluation, we use the Detection Error Tradeoff (DET) curve. The DET curve has the False Positive Per Window (FPPW) values along the horizontal axis and the false negative rate on the vertical axis; points nearer to the origin in the lower left indicate higher detection accuracy. Also, the threshold t used when calculating the B-HOG features is the value determined in preliminary experiments to have the highest detection accuracy. The t value was 0.09 for the INRIA Person Dataset and 0.10 for the vehicle dataset.

5.3. Experiment 1: Effectiveness of R-HOG and SR-HOG features

In experiment 1, we tested the effectiveness of R-HOG and SR-HOG features. The DET curves that represent the results for the two datasets are presented in Fig. 9.

First, we compare the B-HOG features with the R-HOG features. Comparing the human detection rates for when the FPPW from Fig. 9(a) is 1.0×10^{-2}, the detection rate for the R-HOG features is about 8.5% higher than for the B-HOG features. Comparing the vehicle detection rates for when the FPPW from Fig. 9(b) is 1.0×10^{-2}, the detection rate for the R-HOG features is about 1.7% higher than for the B-HOG features.

Next, we compare R-HOG features and SR-HOG features. Comparing the people detection rates for when the FPPW from Fig. 9(a) is 1.0×10^{-2}, the detection rate for the SR-HOG features is about 1.7% higher than for the R-HOG features. Comparing the vehicle detection rates for when the FPPW from Fig. 9(b) is 1.0×10^{-2}, the detection rate for the SR-HOG features is about 2.1% higher than for the R-HOG features. From these results we know that R-HOG features, which are binary patterns obtained by size relationship, are superior to B-HOG features, which are binary patterns obtained by threshold processing, in capturing the relations between cell regions and thus provide a higher detection rate. Furthermore, shifting the gradient orientation of HOG features from one of the cell regions to obtain a binary pattern as is done for SR-HOG features clarifies the size relationship, so the performance is even higher than for R-HOG features.

Finally, we compare the R-HOG features and the SR-HOG features with the HOG features. For both human and vehicle detection, the detection rates for the R-HOG features are lower than for the HOG features. However, the detection rates for the SR-HOG features come closer to those for the HOG features, even though the feature data is reduced.

5.4. Experiment 2: Effectiveness of binary selection by a wilde-card "$*$"

In experiment 2, we tested the effectiveness of introducing "$*$" for selection of binary patterns that are effective for classification when using Real AdaBoost to train the classifiers. In this experiment, we focused on SR-HOG features, which have the best detection accuracy among the proposed methods as shown by the results of experiment 1, and determined the effects of introducing "$*$" in the training. The DET curves for the experimental results for the two datasets are shown in Fig. 10.

From Fig. 10, we can see that introduction of "$*$" to the training improved detection accuracy to a level higher than that for the HOG features. From Fig. 10(a), we see that comparing the human detection rates for when FPPW is 1.0×10^{-2} shows that introduction of "$*$" to SR-HOG features improves accuracy by about 4.1% relative to SR-HOG features without "$*$" and by about 1.5% relative to HOG features. From Fig. 10(b), we see that comparing the vehicle detection rates for when FPPW is 1.0×10^{-2} shows that introduction of "$*$" to SR-HOG features improves accuracy by about 1.2% relative to both SR-HOG features without "$*$" and HOG features. We consider this result to be due to suppression of the reversal of binary values to which "$*$" has been introduced, thus allowing selection of binary values that are even more effective in classification.

5.5. Discussion

Here, we show the proportions of features that contain various numbers of "$*$" that are selected as weak classifiers for each dataset to verify the effectiveness of introducing "$*$" and to ascertain the effectiveness of the SR-HOG features. The results are presented in Fig. 11.

From Fig. 11 (a) and (b), first, since features that include some of the "$*$" codes are selected as weak classifiers, it is clear that adding a process to introduce "$*$" when training is effective in classification. We also know that features which contain zero "$*$" are selected less frequently for both human and vehicle detection in these experiments. This means that introducing "$*$" greatly contributes to an improvement in detection accuracy.

Second, we see that the number of "$*$" for which the selection frequency is the highest is three for B-HOG features and one for SR-HOG features in the case of persons. In the case of vehicles, the numbers are four for B-HOG features and three for SR-HOG features. The fact that the features selected for the SR-HOG features contain fewer "$*$"

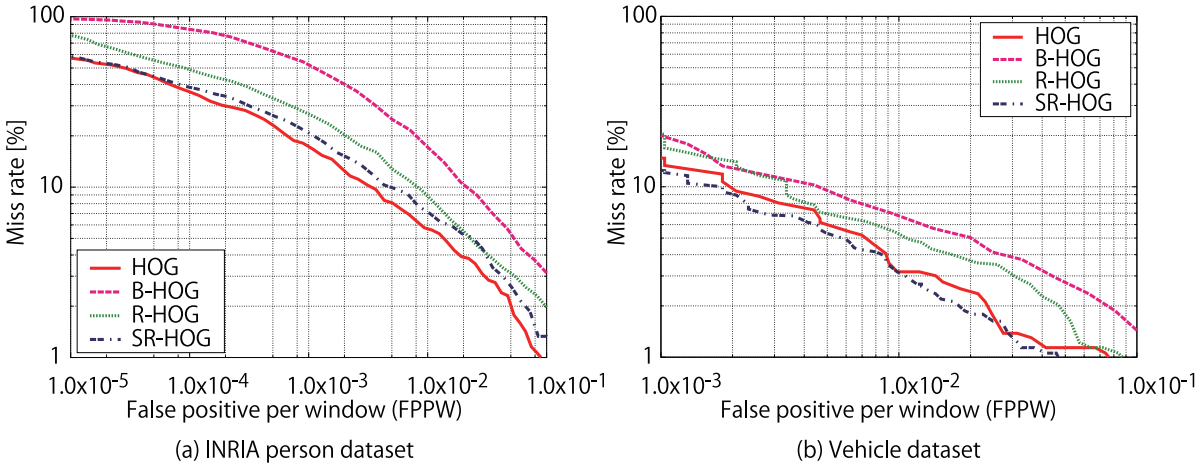


Figure 9. Effectiveness of R-HOG and SR-HOG features.

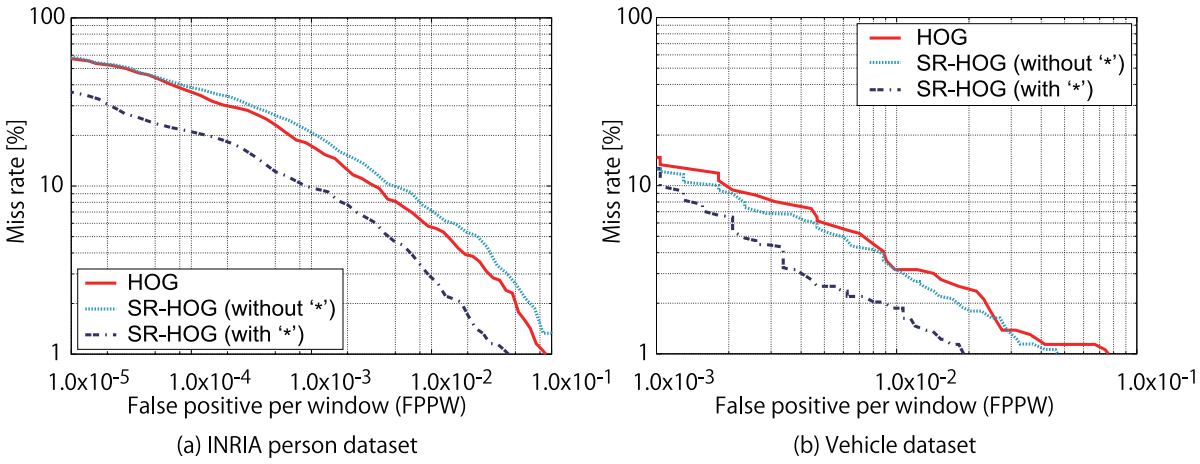


Figure 10. Effectiveness of the wild-card "∗".

for both people and vehicle detection means that the SR-HOG features include more binary values effective for classification than do B-HOG features. This is because the orientation shifting that is done to produce SR-HOG features considerably suppresses the ambiguity caused by the reversal of binary values, which can cause error in classification. Therefore, SR-HOG features can represent the characteristics of the target object more explicitly than B-HOG features.

5.6. Comparison of memory and computational cost

Table 2 shows the memory use and computational cost required for feature extraction and classification in one detection window (64×128 pixels) for HOG features, B-HOG features, R-HOG features, and SR-HOG features, assuming 500 weak classifiers.

The R-HOG features used about 87.5% less memory than the HOG features, and the SR-HOG features used 75.0% less memory than the HOG features. Both the R-HOG features and the SR-HOG features reduced the computational cost by about 50.0% compared with the HOG features. This is because the R-HOG features and the SR-HOG features do not require normalization processing.

When the ambiguous code "∗" is introduced, the amount of memory and the computational cost for online discrimination are almost the same as the values shown in Table 2, because the same number of weak classifiers are used in both processes. In general, binarization of features decreases the detection accuracy such as for B-HOG due to the reduction of information effective for classification. However, SR-HOG features introducing "∗" are able to achieve higher discrimination accuracy than HOG features even though binarization is applied.

6. Conclusion

We have proposed Relational HOG features with a wild-card "∗". This paper makes two contributions. The first is, Relational HOG features, which are binarized on the basis of the size relationship of HOG features extracted from two cell regions. These features can capture the relatedness of

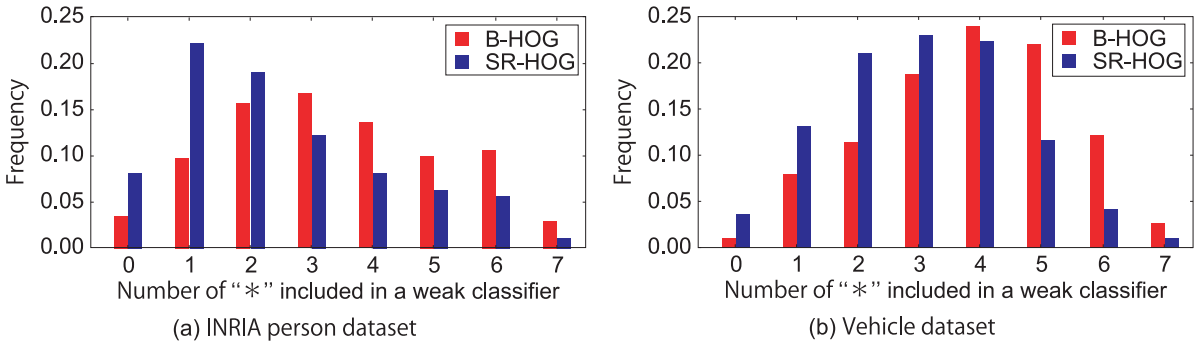


Figure 11. proportions of weak classifiers that contain various numbers of "∗".

Table 2. Comparison of memory use and computational cost.

Feature	HOG	B-HOG	R-HOG	SR-HOG
Memory[KB]	3.91	0.50	0.49	0.98
Computational cost[ms]	5.39×10^{-7}	5.40×10^{-7}	2.70×10^{-7}	2.70×10^{-7}

local regions in a single binary pattern.

The second contribution is the introduction of a wild-card "∗" to allow selection of binary values that are effective in classification with Real AdaBoost. When the difference in magnitude of HOG features is small, the binary value may reverse, which is a cause of lower detection accuracy. Introducing the wild-card "∗", which can take either of the two binary values "0" and "1", suppresses binary value reversal, and achieves highly accurate detection with less feature data.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proc. of ACM international conference on Image and video retrieval*, pages 401–408, 2007.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on CVPR*, volume 1, pages 886–893, 2005.

[3] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conf. on CVPR*, pages 1–8, 2008.

[4] A. Hadid, M. Pietikainen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *IEEE Conf. on CVPR*, volume 2, pages 797–804, 2004.

[5] C. Hou, H. Ai, and S. Lao. Multiview pedestrian detection based on vector boosting. In *Proc. of ACCV*, pages 210–219, 2007.

[6] K. Khattab, J. Dubois, and J. Miteran. Cascade boosting-based object detection from high-level description to hardware implementation. *EURASIP Journal on Embedded Systems*, 2009:1–12, 2009.

[7] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conf. on CVPR*, volume 1, pages 878–885, 2005.

[8] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li. Learning multi-scale block local binary patterns for face recognition. In *in Advances in Biometrics*, pages 828–837, 2007.

[9] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. In *IEEE Trans. on Communications*, volume 28, pages 84–95, 1980.

[10] T. Mita, T. Kaneko, and O. Hori. Joint haar-like features based on feature co-occurrence for face detection (in japanese). *IEICE Trans.*, J89-D(8):1791–1801, 2006.

[11] Y. Mu, S. Yan, Y. Liu, T. Huang, and B. Zhou. Discriminative local binary patterns for human detection in personal album. In *IEEE Conf. on CVPR*, pages 1–8, 2008.

[12] V. Nair, P. O. Laprise, and J. J. Clark. An FPGA-based people detection system. *EURASIP Journal on Applied Signal Processing*, 2005:1047–1061, 2005.

[13] T. Ojala, M. P. ainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. In *Pattern Recognition*, volume 29, pages 51–59, 1996.

[14] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19:1635–1650, 2010.

[15] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *IEEE Conf. on CVPR*, pages 1–8, 2007.

[16] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proc. of IEEE ICCV*, pages 1–8, 2009.

[17] T. Watanabe, S. Ito, and K. Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Proc. of PSIVT*, pages 37–47, 2008.

[18] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. of IEEE ICCV*, volume 1, pages 90–97, 2005.

[19] Y. Yamauchi, M. Takaki, T. Yamashita, and H. Fujiyoshi. Feature co-occurrence representation based on boosting for object detection. In *International Workshop on Socially Intelligent Surveillance and Monitoring(in conjunction with CVPR)*, pages 31–38, 2010.

[20] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Proc. of IEEE ICCV*, 2009.