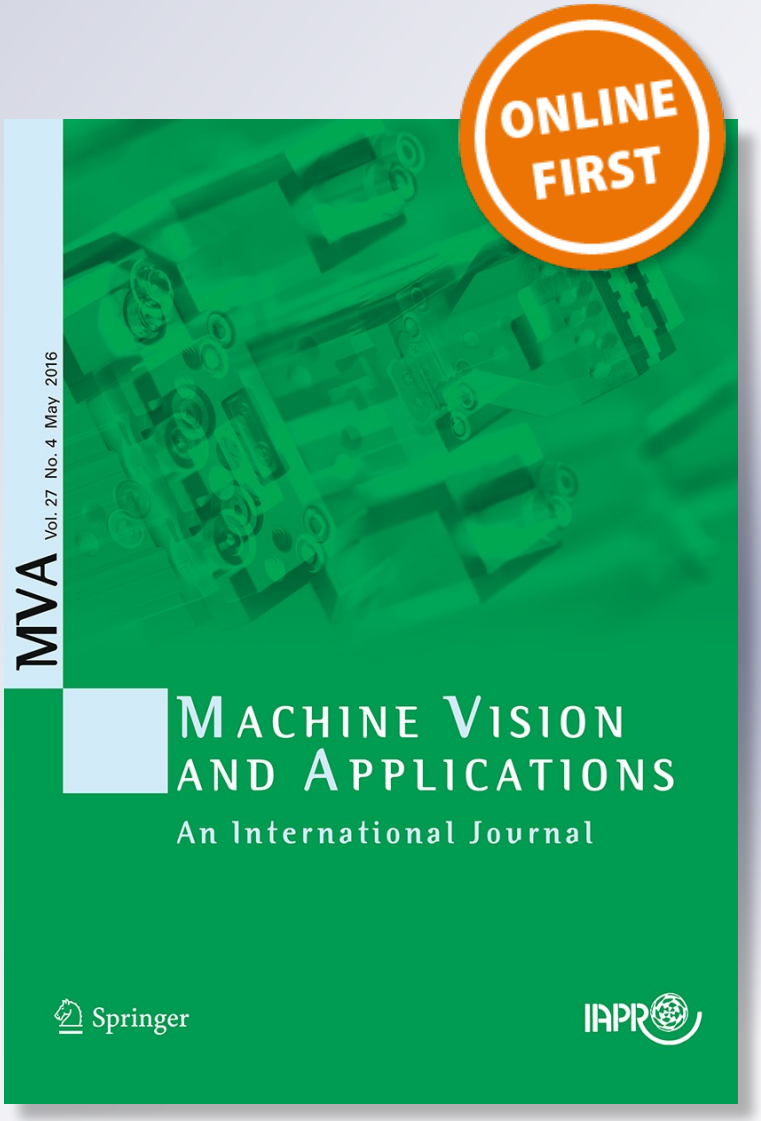
Discriminative local binary pattern

Takumi Kobayashi

Machine Vision and Applications

ISSN 0932-8092

Machine Vision and Applications
DOI 10.1007/s00138-016-0780-8



Springer

Machine Vision and Applications
DOI 10.1007/s00138-016-0780-8

SPECIAL ISSUE PAPER

Discriminative local binary pattern

Takumi Kobayashi[1]

Received: 30 November 2015 / Revised: 4 April 2016 / Accepted: 9 May 2016

Abstract Local binary pattern (LBP) is widely used to extract image features as well as motion features in various visual recognition tasks. LBP is formulated in quite a simple form and thus enables us to extract effective features with a low computational cost. There, however, are some limitations mainly regarding sensitivity to noise and loss of image contrast information. In this paper, we propose a novel LBP-based feature extraction method to remedy those drawbacks without degrading the simplicity of the original LBP formulation. LBP is built upon encoding local pixel intensities into binary patterns which can be regarded as separating them into two modes (clusters). We introduce Fisher discriminant criterion to optimize the LBP coding for exploiting binary patterns more stably and discriminatively with robustness to noise. Besides, image contrast information is incorporated in a unified way by leveraging the discriminant score as a weight on the binary pattern; therefore, the prominent patterns, such as around edges, are emphasized. The proposed method is applicable to extract not only image features but also motion features by both efficiently decomposing a XYT volume patch into 2-D patches and employing the effective thresholding strategy based on the volume patch. In the experiments on various visual recognition tasks, the proposed method exhibits superior performance compared to the ordinary LBP and the other methods.

Keywords Visual recognition · Image feature · Local binary pattern · Discriminant criterion

✉ Takumi Kobayashi
takumi.kobayashi@aist.go.jp

[1] National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Japan

1 Introduction

It is a primary process to extract features from images and videos for visual recognition, such as object classification and action recognition. While various types of image feature have been proposed so far [4,14,19,37] and extended for motion features [10,15,33], local binary pattern (LBP) [26,31] is one of the commonly used features due to its simple formulation and high performance. The LBP method has been mainly applied to measure texture characteristics [7,8,26–28], and in recent years it is shown to be favorably applicable to various kinds of visual recognition tasks besides texture classification, such as face recognition [1,34], face detection [9], pedestrian detection [37] and sound classification [16]. LBP is also known as census transform [40] and utilized for a holistic image descriptor [38]. As in the other image feature extraction methods, LBP is extended to 3-D volume features for classifying dynamic textures [42,43], lip movement [3] and human action [22,24].

The LBP method encodes pixel intensities of a local patch into binary patterns on the basis of the center pixel intensity. Although it is nicely formulated in the simple form, there are some limitations in LBP, mainly regarding sensitivity to noise and loss of local textual information, i.e., image contrast. In the last two decades, considerable research effort has been made to address those drawbacks of LBP leading to variants of LBP. In [28], the image contrast information is separately extracted by computing variance of local pixel intensities and joint distribution of the contrast feature and LBP is employed. The contrast information represented by the local variance is naturally incorporated into LBP formulation via weighting binary patterns in [7]. LBP can be combined with histogram of oriented gradient (HOG) features [4] to compensate the loss of contrast information [37]. The robustness to noise is improved by developing the binary

Published online: 07 June 2016

patterns to ternary patterns [34] which are further extended to quinary ones [25], though the number of patterns corresponding to the feature dimensionality is significantly increased; those methods [25,34] compress the patterns by considering the ternary/quinary values separately. It is also possible to build noise-robust LBP by simply considering local statistics, mean [9] and median [8], as a threshold instead of the center pixel intensity in coding. To further improve robustness, we have recently extended LBP to fully incorporate the statistical information, mean and variance, in the processes both of coding and weighting [16]. For more elaborated review of LBP, refer to [31].

In this paper, we propose a novel method to extract LBP-based features by remedying the limitations of LBP while retaining simplicity of the original LBP formulation. We first generalize the LBP formulation by focusing on the two fundamental processes of coding and weighting, and then along the line of [7–9,16], propose *discriminative LBP* by providing a discriminative approach to determine those two ingredients. In the discriminative approach, LBP coding is regarded as separating local pixel intensity distribution into two modes (clusters) and from that viewpoint, a threshold is optimized by maximizing the Fisher discriminant score which is further utilized in weighting. Thereby, the discriminative LBP stably encodes the local pixel intensities into binary patterns via the optimization with high robustness to noise, incorporating image contrast information in a unified manner. Due to its simplicity as in the ordinary LBP, the proposed method can be easily integrated with the sophisticated extension which has been applied to LBP, such as uniform pattern [27] and combination with the other image features [37]. In addition, it can be extended to 3-D volume feature extraction as in the other LBP variants [24,43] by efficiently decomposing a 3-D volume patch into an ensemble of 2-D patches as well as employing the effective thresholding strategy based on the volume patch.

The rest of this paper is organized as follows: in the next section, we present the general formulation of LBP with brief reviews of the LBP variant methods in that framework and subsequently detail the proposed discriminative LBP method. Section 3 describes the extension of the proposed method to volume feature extraction, and in Sect. 4 several techniques are presented to further improve the effectiveness of the proposed feature. The experimental results on image classification for pedestrian detection and face recognition as well as on action classification are shown in Sect. 5, and finally Sect. 6 contains our concluding remarks.

This paper is extended from the CAIP2015 conference paper [12], containing the substantial improvements mainly in that we present the extended method to volume feature extraction with thorough analysis on it in the experiments on action classification. We also improved and detailed the description of the proposed method by presenting qualitative comparison of the LBP codes and practical algorithms with computational analysis.

2 Discriminative local binary pattern

This section describes the detail of the proposed method, called *discriminative LBP*. We first give a general formulation for extracting local binary patterns (LBP) [26] with reviewing the previous LBP variants based on that formulation. Then, the discriminative perspective is introduced into the processes both of coding and weighting which are fundamental in the general formulation.

Although in this section we basically proceed to explain and discuss the method in the case of *image* feature extraction, the proposed method can be extended to extract features from a *volume*, such as motion images (spatio-temporal volume) [43], as described in Sect. 3.

2.1 General formulation for LBP

Let $\boldsymbol{r} = (x, y)$ be a spatial position in a two-dimensional image I and $I(\boldsymbol{r})$ indicates the pixel intensity at that position. LBP method [26] focuses on a local image patch and encodes local pixel intensities by binarizing them as follows;

$$\mathrm{F}(\mathcal{L}_{\boldsymbol{c}}; \tau_{\boldsymbol{c}}) = \sum_{i=1}^{N} 2^{i-1} [\![I(\boldsymbol{r}_i) > \tau_{\boldsymbol{c}}]\!] \in \{0, \ldots, 2^N - 1\}, \quad (1)$$

where $[\![\cdot]\!]$ indicates the Iverson bracket that equals to 1 if the condition in the brackets is satisfied and 0 otherwise. $\mathcal{L}_{\boldsymbol{c}} = \{\boldsymbol{r}_i\}_{i=1}^{N}$ denotes a local pixel configuration centered at $\boldsymbol{c} \in \mathbb{R}^2$, comprising N spatial positions $\boldsymbol{r}_i$ that surround $\boldsymbol{c}$. For example, the simplest and widely used configuration consists of $N = 8$ surrounding pixels in a 3×3 local patch, as shown in Fig. 1a, and it is further extended in a multi-scale setting [28]. Though the number of codes (binary patterns) is exponentially increased according to N, it is also possible to suppress the pattern variation by considering uniform patterns [27] as described in Sect. 4.

The local image pattern on $\mathcal{L}_{\boldsymbol{c}}$ is encoded into a N-bit code by means of binarization of pixel intensities with a threshold $\tau_{\boldsymbol{c}}$ via (1). Finally, the LBP codes are aggregated to LBP histogram $\boldsymbol{z} \in \mathbb{R}^{2^N}$ over a region of interest (ROI) $\mathbb{D}$,

$$z_j = \sum_{\boldsymbol{c} \in \mathbb{D}} w_{\boldsymbol{c}} [\![\mathrm{F}(\mathcal{L}_{\boldsymbol{c}}; \tau_{\boldsymbol{c}}) = j - 1]\!], \quad j \in \{1, \ldots, 2^N\}, \quad (2)$$

where $w_{\boldsymbol{c}}$ is a voting weight to represent significance of the local binary pattern (code). In the parts-based features, the weight $w_{\boldsymbol{c}}$ also works for representing the parts (ROI) as described in Sect. 4.

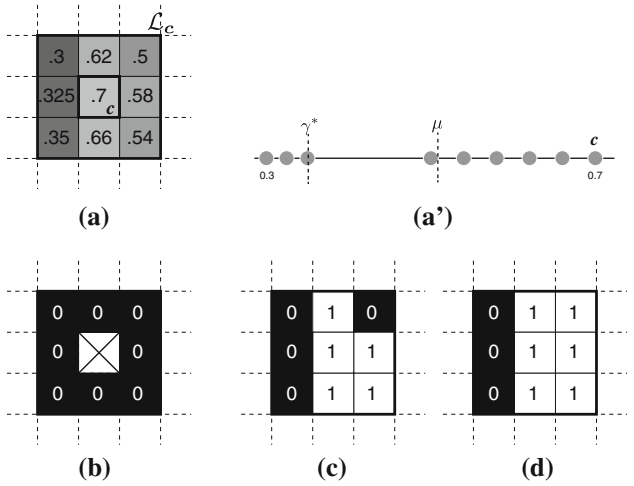


Fig. 1 Examples of LBP codes by various thresholds. A local patch (**a**) of pixel intensity distribution (**a**′) is encoded into binary codes by ordinary LBP $\tau = I(\boldsymbol{c})$ [26] (**b**), statistics-based LBP $\tau = \mu$ [16] (**c**) and the proposed method $\tau = \gamma^*$ (**d**). In **c**, **d**, $\mathcal{L}_{\boldsymbol{c}}$ includes the center pixel $\boldsymbol{c}$. The proposed method produces a stable code with a large margin which is hardly affected by noise

Table 1 Comparison in variants of LBP in the general formulation (1, 2) containing the threshold τ and voting weight w

Method	τ	w
Ordinary LBP [26]	$I(\boldsymbol{c})$	1
Median LBP [8]	$\mathtt{median}(I)$	1
Improved LBP [9]	μ	1
LBP variance [7]	$I(\boldsymbol{c})$	σ^2
Statistics-based LBP [16]	μ	σ
Discriminative LBP (proposed)	$\arg\max \sigma_B$	$\sqrt{\frac{\max \sigma_B^2}{\sigma^2+C}}$

As shown in (1) and (2), the general formulation of LBP contains two ingredients, the threshold $\tau_{\boldsymbol{c}}$ and the weight $w_{\boldsymbol{c}}$. From these perspectives, LBP variants can be placed in this general formulation as shown in Table 1. As to coding, an ordinary LBP [26] is established by setting the center pixel intensity $I(\boldsymbol{c})$ as the threshold, $\tau = I(\boldsymbol{c})$. The variant methods [8,9] modify it by employing local statistics, $\tau = \mu = \frac{1}{N}\sum_i I(\boldsymbol{r}_i)$ and $\tau = \mathtt{median}_i[I(\boldsymbol{r}_i)]$, respectively. On the other hand, for weighting, most methods simply employ *hard* voting weights, i.e., $w = 1$, as the original LBP [26] does, which means that all the LBP codes equally contribute to characterize an image, losing local image contrast information. To compensate it, the local variance, $\sigma^2 = \frac{1}{N}\sum_i (I(\boldsymbol{r}_i) - \mu)^2$ is separately employed as the local image contrast in [28], and it is incorporated as the weight w in [7]. Recently, we have proposed statistics-based LBP [16] by effectively applying those simple statistics, mean and standard deviation, to both coding and weighting as $\tau = \mu$ and $w = \sigma$.

Thus, we can say that the LBP variants are formulated by modifying two ingredients τ and w. Along this line, we propose a novel method by designing them in a discriminative manner for extracting effective image features of high robustness with exploiting image contrast.

2.2 Discriminative coding

We propose a novel coding method which optimizes the threshold τ and the voting weight w in (1, 2) based on a discriminative criterion.

The LBP coding (1) can be viewed as approximating local pixel intensity distribution in $\mathcal{L}_{\boldsymbol{c}}$ by two modes separated by the threshold τ. In the previous methods, the spatial center, $I(\boldsymbol{c})$, or statistical centers, μ and $\mathtt{median}_i[I(\boldsymbol{r}_i)]$, are simply employed a priori, but those are not regarded as the optimum from the viewpoint of the approximation. Therefore, we measure *quality* of the coding (approximation) in a least-square framework by introducing the following residual error,

$$\epsilon(\tau) = \frac{1}{N}\left\{ \sum_{i|I(\boldsymbol{r}_i)\le\tau} (I(\boldsymbol{r}_i) - \mu_0)^2 + \sum_{i|I(\boldsymbol{r}_i)>\tau} (I(\boldsymbol{r}_i) - \mu_1)^2 \right\}, \tag{3}$$

where

$$\mu_0 = \frac{1}{N_0}\sum_{i|I(\boldsymbol{r}_i)\le\tau} I(\boldsymbol{r}_i), \quad N_0 = \sum_i [\![I(\boldsymbol{r}_i) \le \tau]\!], \tag{4}$$

$$\mu_1 = \frac{1}{N_1}\sum_{i|I(\boldsymbol{r}_i)>\tau} I(\boldsymbol{r}_i), \quad N_1 = \sum_i [\![I(\boldsymbol{r}_i) > \tau]\!], \tag{5}$$

and obviously $N = N_0 + N_1$ and $\mu = \frac{N_0}{N}\mu_0 + \frac{N_1}{N}\mu_1$. Here, two modes are represented by the mean μ_0 and μ_1, respectively. This least-square formulation also means to fit two Gaussian models in the pixel intensity distribution from a probabilistic viewpoint. We determine the threshold τ so as to minimize this approximation error.

The residual error ϵ corresponds to within-class variance σ_W^2 for the two classes which are partitioned by the threshold τ. Thus, minimizing ϵ coincides with maximization of Fisher discriminant score [5], actually maximization of between-class variance σ_B^2, since $\sigma_B^2(\tau) = \sigma^2 - \sigma_W^2(\tau)$ where σ^2 is the constant total variance;

$$\sigma_B^2(\tau) = \frac{N_0}{N}(\mu_0 - \mu)^2 + \frac{N_1}{N}(\mu_1 - \mu)^2 \tag{6}$$

$$= \frac{N_0 N_1}{N^2}(\mu_1 - \mu_0)^2 = \frac{(N_0\mu - N_0\mu_0)^2}{N_0(N - N_0)}. \tag{7}$$

Thus, the threshold τ that minimizes the approximation error ϵ (3) is obtained by

$$\gamma^* = \arg \max_{\tau \in \{I(\boldsymbol{r}_i)\}_{i=1}^N} \sigma_B^2(\tau). \tag{8}$$

Thereby, the proposed discriminative coding with γ^* reduces the error ϵ in assigning binary codes (1) as well as enhances the discriminativity (σ_B) between two modes partitioned by the threshold. This procedure is performed in the same way as Otsu's auto-thresholding method [29] applied to pixel intensities $\{I(\boldsymbol{r}_i)\}_{i=1}^N$ as detailed in Sect. 2.4.

Next, we can accordingly determine the voting weight w as the (square root of) discriminant score;

$$w = \sqrt{\frac{\sigma_B^2(\gamma^*)}{\sigma^2 + C}}, \tag{9}$$

where C is a small constant to avoid numerical instability for smaller σ, especially in the case that local pixel intensities are close to uniform; in this study, we set $C = 0.01^2$ for pixel intensity scale [0, 1]. This weight reflects how far the two modes are separated by γ^* and therefore measures significance of the corresponding binary pattern.

Note that in the proposed method which frees the center pixel from being the threshold, we have a choice whether the local patch $\mathcal{L}_{\boldsymbol{c}}$ contains the center pixel ($N = 9$) or not ($N = 8$). It is basically determined according to the computational requirement such as memory limitation and computation speed.

The proposed method is built on the optimization (8), requiring extra computation cost compared to the other methods which employ hard coding [8,9,26] and soft coding with simple statistics [7,16] of lower computational burden. It, however, is negligible in the case of a smaller local patch size N and such computational issue is discussed in Sect. 2.4.

2.3 Characteristics of discriminative coding

2.3.1 Robustness

The ordinary LBP [26] of $\tau = I(\boldsymbol{c})$ and $w = 1$ always assigns a local image pattern with one of the LBP codes, no matter whether the image pattern is less significant, such as being close to uniform, i.e., less image contrast. This is because the LBP coding takes into account only magnitude relationships between the pixel intensities of a center, $I(\boldsymbol{c})$, and its neighborhoods, $I(\boldsymbol{r}_i)$, in disregard of the margin. Thus, even a small fluctuation on the pixels whose intensities are close to $I(\boldsymbol{c})$ easily degenerates the LBP code by breaking up the magnitude relationships, which results in totally different features. In other words, the binary codes on the pixel intensities of a small margin from $I(\boldsymbol{c})$ are vulnerable to noise, causing unstable LBP features.

On the other hand, the proposed coding extracts a discriminative structure of a local pixel intensity distribution,

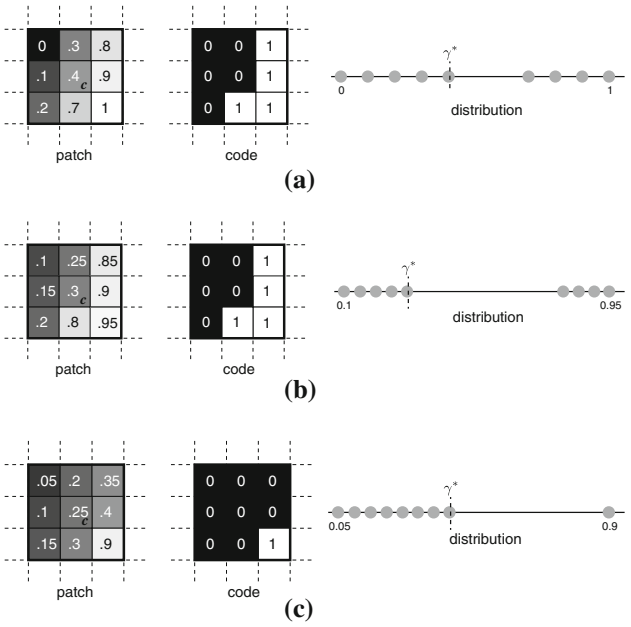


Fig. 2 Examples of weights in the proposed method. In each figure, the input local patch, the resultant binary pattern (code) and the pixel intensity distribution are shown from *left* to *right*. Details are in the text. **a** $w = 0.92$, **b** $w = 0.98$, **c** $w = 0.89$

exhibiting high robustness to noise. In the structure, two modes endowed by the threshold γ^* are discriminatively separated with a statistically large margin due to maximizing Fisher discriminant score in (8); it exhibits stable patterns as shown in Fig. 1. Those binary patterns of large margin are hardly affected by noise and contribute to robust features. Thus, we can stably exploit an essential (binary) pattern of a pixel intensity distribution even under perturbation on pixel intensities. Besides, in weighting, the significance of the local pattern is effectively measured by Fisher discriminant score (9) as shown in Fig. 2. Even for the similar image patches resulting in the same code, the patch of well-separated pixel intensities (Fig. 2b) gets the larger weight than that of blurred intensities (Fig. 2a). And, smaller weight is assigned to the patch of which distribution is highly biased (Fig. 2c), even though it is significantly separated. Such a biased distribution can be regarded as a noisy pattern containing an outlier and thus it is favorable that such code contributes less to the feature.

The LBP code maps are visualized in Fig. 3 using pseudo-color to represent the codes. The proposed method produces rather smooth and stable code maps (Fig. 3c) in comparison with that of the original LBP (Fig. 3b) which is noisy and unstable, especially on the vertical lines in the image, consistent codes are assigned by the proposed method, though the original LBP produces noisy ones. And, the proposed weight map (Fig. 3d) reveals the edge-like structure in the image; the weight is high around the edge region where the local pixel intensity distribution is well separated, while it is low on the

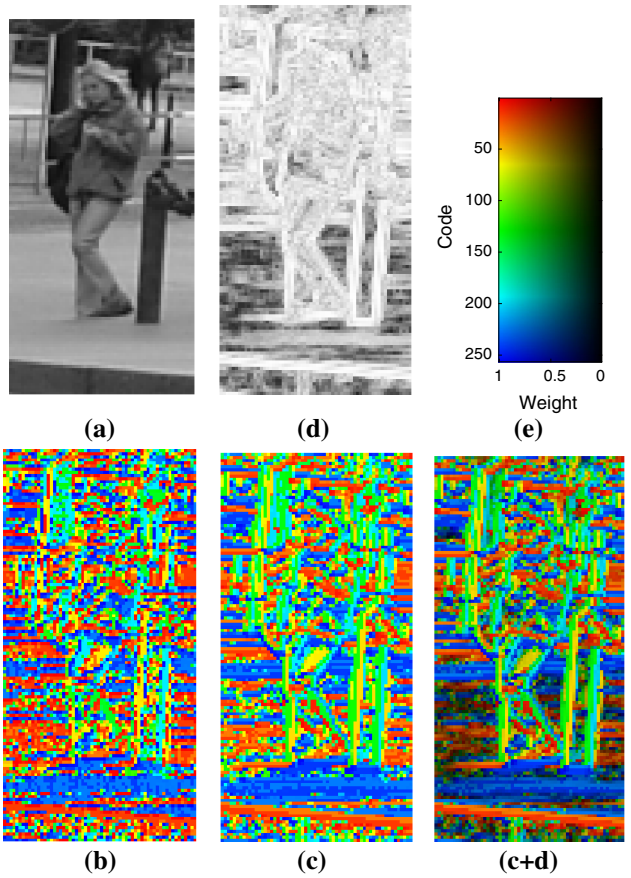


Fig. 3 Visualization for LBP codes. The input image (**a**) is converted to the original LBP code map (**b**) and ours (**c**) which are shown in *pseudo-color* (**e**). The proposed method also produces the weight map (**d**) shown in *gray scale* [0, 1], and our codes combined with the weights are visualized in (**c** + **d**). This figure is best viewed in color (color figure online)

homogeneous region. Those edge-based structure helps us to effectively characterize the image content [4]. Thereby, in contrast to the ordinary LBP, the proposed method composed of the codes (Fig. 3c) and weights (Fig. 3d) can extract effective features (Fig. 3c + d) with enhancing robustness to noise.

2.3.2 Invariance

It is noteworthy that the proposed LBP is invariant to affine transformation of pixel intensities, $aI(\boldsymbol{r}) + b$, in terms of coding and weighting as in the ordinary LBP, while the statistics-based LBP [16] is affected by scaling a in the weight $w = \sigma$.

2.3.3 Geometric feature

The proposed method effectively extracts the geometrical characteristics in an image, various patterns of gradients and

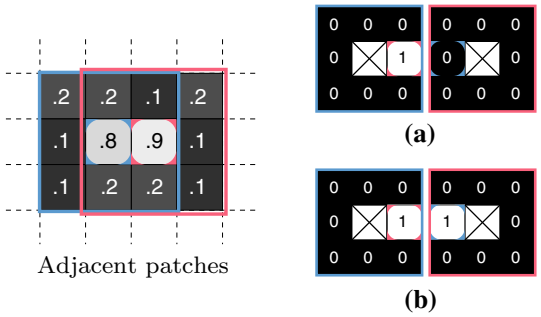


Fig. 4 LBP codes for adjacent patches. The adjacent patches share their center pixels of which codes are always flipped in the original LBP (**a**). The proposed method appropriately encodes them based on the local pixel patterns (**b**)

curvatures which are considered to be fundamental local geometries for describing an image structure. Those essential characteristics are represented by the local binary patterns which reflect discriminative structures of the pixel intensity distributions with high robustness to noise. Through weighting by Fisher discriminant scores, the patches of less texture are ignored, contributing less to the feature, while distinctive ones, such as around object edges, are highly focused on by large weights as shown in Fig. 3d.

There exist some constraints in the original LBP feature [38]. The LBP coding is based on pair-wise comparison between the center pixel and the neighboring one. Thus, in adjacent local patches sharing the pair, the codes that corresponds to the pair are *always flipped* in disregard of the other pixels in the patches (Fig. 4a). In contrast, the proposed method encodes the local patch based on the whole pixels (intensity distribution) in the patch and thus the shared pixel pair is appropriately (adaptively) encoded according to the pixel pattern in the local patch (Fig. 4b).

2.4 Computational issues

We finally mention the computational issue in the proposed method. The method requires the optimization (8) of rather high computational cost compared to the previous LBP-based methods which employs simple coding scheme. The optimization (8) can be practically performed in two ways of greedy (Algorithm 1) and efficient approaches (Algorithm 2). The greedy approach computes $\sigma_B^2(\tau)$ by checking all pixel pairs in a brute-force manner, $O(N^2)$, and this would work for smaller number of pixel N in a local patch. On the other hand, the efficient approach based on [29] incrementally updates σ_B^2 according to the *sorted* pixel intensities in $O(N \log(N))$ and thus is more efficient for larger N since the scoring procedure would be the bottle neck in the smaller N. Those two approaches are empirically compared in Fig. 5 where the number of pixels N in the local patch $\mathcal{L}_{\boldsymbol{c}}$ is varied from $N = 8, 9$ for a 2-D image patch to $N = 26, 27$ for a

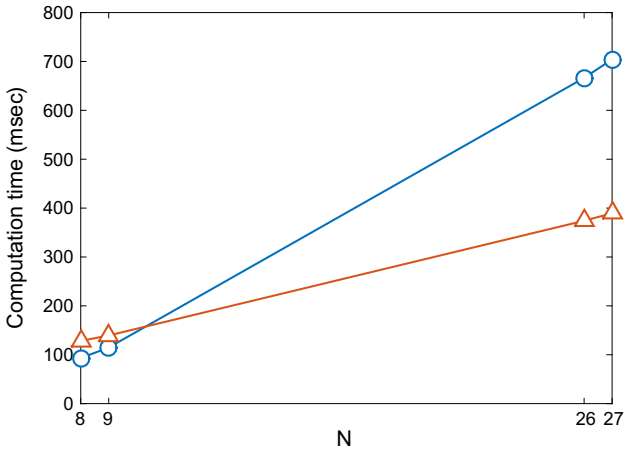


Fig. 5 Computation time in discriminative LBP coding. The optimization for discriminative codes (8) is performed on an image of 640×480 pixels with $N = 8, 9$, and a volume of $640 \times 480 \times 3$ voxels with $N = 26, 27$, both of which produce $304964 = 638 \cdot 478$ patches

Algorithm 1 : Greedy approach for discriminative thresholding

Require: $\{I_i\}_{i=1}^N$: N pixel intensities.
Total mean: $\mu = \frac{1}{N}\sum_{i=1}^N I_i$.
for $i = 1$ **to** N **do**
  $N_0 = 0,\ \xi_0 = 0$.
  for $j = 1$ **to** N **do**
    if $I_j \le I_i$ **then**
      $N_0 \leftarrow N_0 + 1$
      $\xi_0 \leftarrow \xi_0 + I_j$
    end if
  end for
  $\sigma_B(I_i) = \frac{(N_0\mu - \xi_0)^2}{N_0(N-N_0)}$.
end for
Ensure: $\gamma^* = \arg\max_{I_i} \sigma_B(I_i)$: optimum threshold.

Algorithm 2 : Efficient approach for discriminative thresholding

Require: $\{I_i\}_{i=1}^N$: N pixel intensities.
Total mean: $\mu = \frac{1}{N}\sum_{i=1}^N I_i$.
Sort $\{I_i\}$ into $\{\tilde{I}_i\}$ such that $\tilde{I}_i \le \tilde{I}_j$ $(i < j)$.
$N_0 = 0,\ \xi_0 = 0$.
for $i = 1$ **to** $N - 1$ **do**
  $N_0 \leftarrow N_0 + 1$
  $\xi_0 \leftarrow \xi_0 + I_i$
  $\sigma_B(\tilde{I}_i) = \frac{(N_0\mu - \xi_0)^2}{N_0(N-N_0)}$.
end for
Ensure: $\gamma^* = \arg\max_{I_i} \sigma_B(I_i)$: optimum threshold.

3-D volume patch (described in Sect. 3). As expected, in the case of smaller N of an *image* patch, the greedy approach is faster than the efficient one, but the result is inverted for larger N of a *volume* patch. Thus, we select the optimization approach according to the domain that the method is applied; the greedy approach (Algorithm 1) for 2-D image feature extraction and the efficient one (Algorithm 2) for 3-D volume feature extraction.

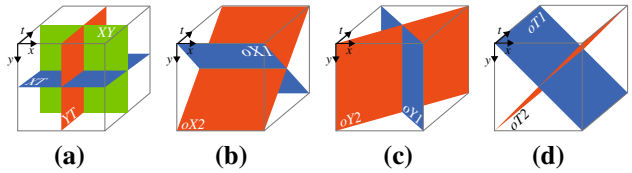


Fig. 6 Decomposition of a 3-D volume patch into nine types of 2-D patches. *XY*, *XT* and *YT* patches are ordinary used to represent *XYT* spatio-temporal volume of motion images (**a**). In this study, we also consider the other types of patches along *X*, *Y* and *T*-axis each of which provide two orthogonal patches (**b**, **c**, **d**)

3 Extension to volume feature extraction

The formulation described above for (2-D) image feature extraction is straightforwardly applied to 3-D volume feature extraction, such as in motion (XYT) recognition, by extending the local patch $\mathcal{L}_c$ of 3×3 pixels to 3-D patch of $3 \times 3 \times 3$ voxels. It, however, results in infeasibly large-dimensional features; $3 \times 3 \times 3$ patch leads to $N = 27$ and $2^{27} \sim 100$M codes. Therefore, the LBP-based volume feature extraction methods [3,22,42,43] suppress the dimensionality by decomposing the local volume patch into an ensemble of 2-D patches. In most cases, the 3-D (XYT) volume is represented in a marginal manner by using three types of 2-D patches on XY, XT and YT slices (Fig. 6a). In this study, we additionally consider the other types of patches along X, Y and T-axis as shown in Fig. 6bcd. Along each axis, two orthogonal patches are conceivable; for example, along X-axis, there are two patches of oX1 and oX2 which are orthogonal, other than XY and XT patches. In total, nine types of 2-D patches are sampled from the 3-D volume patch and they are empirically compared in Sect. 5.4.

In the ordinary LBP coding, the decomposed patches that share the center pixel are encoded by the identical threshold τ of the center pixel intensity. In contrast, for the discriminative coding, there are two ways to determine the threshold. One way, called *respective thresholding*, is that we regard those patches to be independently drawn from the volume and give the threshold individually for each patch; three patches are encoded by respective thresholds. The other one, called *joint thresholding*, is based on the joint representation of those patches. Namely, the single threshold is computed over the volume patch ($3 \times 3 \times 3$) from which the decomposed patches are drawn, and it is identically applied to encode those patches. From the computational perspective, the respective thresholding approach to perform the optimization (8) on a 2-D patch several times is comparable to the joint thresholding that performs it only once but on a 3-D volume patch; the computation on the 2-D patch is about three times faster than that on the 3-D patch as shown in Fig. 5. These two approaches are empirically compared in terms of performance in Sect. 5.4.

4 Techniques for effective feature

We mention some practically useful techniques for extracting effective features.

4.1 Normalization

The discriminative LBP produces features in a histogram form which is regarded as a discrete probability distribution over the LBP codes. The Hellinger (Bhattacharya) kernel can be effectively applied to measure the similarity between those probability distributions [2], and it is possible to embed the kernel in a (linear) dot product of the feature vectors by normalizing the features in the following form [32]; $\hat{z} = \sqrt{\frac{z}{\|z\|_1}}$. This normalization enhances the discriminative power of features by enhancing difference on smaller feature values while suppressing it on larger values via the square root function.

4.2 Cell-structured feature

In the case of object-related classification, it is demanded to extract features sensitive to *parts* which compose the target objects. Those part-based features are naively extracted by partitioning the object image into subregions, called *cells*, on which the features are computed [4,19]. In this setting, the voting weights in (2) are composed not only of the weight to represent pattern significance (9) but also of the closeness to surrounding cells via bilinear voting on 2-D spatial grids [19] and trilinear voting on 3-D spatio-temporal grids [33]. The final feature is built by simply concatenating all cell-wise features. Note that in this study, the above-mentioned normalization is applied to respective cell-wise feature vectors before concatenation.

4.3 Binary pattern reduction

The dimensionality of the LBP-based feature is exponentially increased according to the number of pixels N in the local patch $\mathcal{L}_{\boldsymbol{c}}$. If one wants to reduce the feature dimensionality such as due to memory limitation, binary patterns can be reduced by applying *uniform patterns* [27]. Uniform patterns are constructed by allowing only a few times 0/1 transitions on the neighborhood pixels surrounding the center $\boldsymbol{c}$ in the 2-D patch; 256-dimensional features of $N = 8$ are reduced to 58-dimensional ones by uniform patterns allowing only two times 0/1 transitions and 512-dimensional features of $N = 9$ including the center pixel become 114-dimensional ones as well.[1] In the case of volume data, the uniform pattern is applied to respective 2-D patches into which the 3-D volume patch is decomposed.

5 Experimental results

We first apply the proposed method to image classification tasks of pedestrian detection using the Daimler Chrysler pedestrian benchmark dataset [23] for evaluating the performance from various aspects (Sect. 5.1) and INRIA person dataset [4] (Sect. 5.2), and of face recognition using FERET dataset [30] (Sect. 5.3). Then, the extended method to volume features (Sect. 3) is applied to action classification on HMDB51 dataset [17] (Sect. 5.4).

In feature extraction, the local patch $\mathcal{L}_{\boldsymbol{c}}$ is restricted within 3×3 pixels ($3 \times 3 \times 3$ voxels) except for face recognition (Sect. 5.3) since the larger patch degrades performance as reported in [37], and we apply L_2-Hellinger normalization to LBP-based feature vectors.

5.1 Performance analysis on Daimler Chrysler dataset

The Daimler Chrysler pedestrian dataset [23] is composed of five disjoint sets, three for training and two for test. Each set has 4800 pedestrian and 5000 pedestrian-free images of 18×36 pixels. For constructing cell-structured features, we consider cells of 6×6 pixels, producing 3×6 cells over an image. We follow the standard evaluation protocol on this dataset, in which the linear classifier is trained on two out of three training sets by using liblinear [6] and is tested on each of the test sets, producing six evaluation results. We measure the average of accuracies at equal error rate across the six classification results.

In the following, we analyze in detail the proposed method in terms of coding by τ, weighting with w and feature dimensionality controlled by a local patch $\mathcal{L}_{\boldsymbol{c}}$ and pattern reduction (Sect. 4). Performance results in various settings are shown in Table 2.

5.1.1 Coding and weighting

Compared to the ordinary LBP (the first row in Table 2), the proposed method (the last row) significantly improves the performance with and without uniform patterns (Table 2ab). Under the condition of the same feature dimensionality, the method is still largely superior to the ordinary LBP as shown in lines 1 and 5 of Table 2, though only weighting and coding are modified to discriminative ones (Sect. 2.2). In addition, our method outperforms the statistics-based LBP [16] in all feature dimensionalities; see lines 3, 5, 7 and 9 in Table 2. We

[1] 58 patterns for $N = 8$ consist of 1 flat pattern for zero 0/1 transition, 56 moderate patterns for less than or equal to twice transitions and 1 messy pattern for greater than twice transitions. In $N = 9$, we consider 1 flat and 1 messy patterns no matter what the center pixel is, and $112 = 56 \times 2$ moderate patterns according to the center pixel state.

Table 2 Performance analysis on the Daimler Chrysler dataset for various settings in LBP formulation

	(a) Full binary pattern					(b) Uniform pattern				
	$\mathcal{L}_c$	τ	w	Dim.	Acc. (%)	$\mathcal{L}_c$	τ	w	Dim.	Acc. (%)
1.	$N=8$	$I(\boldsymbol{c})$	1	256	92.29	$N=8$	$I(\boldsymbol{c})$	1	58	91.32
2.	$N=8$	μ	1	256	94.04	$N=8$	μ	1	58	93.42
3.	$N=8$	μ	σ	256	94.32	$N=8$	μ	σ	58	93.64
4.	$N=8$	γ^*	1	256	95.02	$N=8$	γ^*	1	58	94.71
5.	$N=8$	γ^*	$\sqrt{\frac{\sigma_B^2}{\sigma^2+C}}$	256	95.11	$N=8$	γ^*	$\sqrt{\frac{\sigma_B^2}{\sigma^2+C}}$	58	94.77
6.	$N=9$	μ	1	512	94.62	$N=9$	μ	1	114	94.23
7.	$N=9$	μ	σ	512	94.87	$N=9$	μ	σ	114	94.40
8.	$N=9$	γ^*	1	512	95.12	$N=9$	γ^*	1	114	94.93
9.	$N=9$	γ^*	$\sqrt{\frac{\sigma_B^2}{\sigma^2+C}}$	512	95.25	$N=9$	γ^*	$\sqrt{\frac{\sigma_B^2}{\sigma^2+C}}$	114	95.16

The local patch $\mathcal{L}_c$ of $N=8$ excludes the center pixel. The number of dimensionality of cell-wise features is shown in the column of 'Dim.'. The performances of the proposed method are underlined

further set the weighting as $w=1$ in both statistics-based LBP and our method in order to give light on the effectiveness of the discriminative coding with threshold γ^*. A threshold in coding is crucial to encode the local pixel intensities into a binary pattern, while weighting works just for assigning significance to those patterns. Comparing the methods of $w=1$, the thresholds μ and γ^* are superior to the ordinary threshold $I(\boldsymbol{c})$ and in particular, our discriminative threshold γ^* significantly outperforms both of μ and $I(\boldsymbol{c})$. Thus, it is confirmed that the proposed method which discriminatively optimizes the threshold can effectively work in constructing local binary patterns for image features. By incorporating discriminative weights, the performance is further improved as shown in lines 4–5 and 8–9.

5.1.2 Dimensionality

By controlling a local patch $\mathcal{L}_c$ and applying the uniform pattern (Sect. 4), the feature dimensionality is halved, accordingly causing a little performance degeneration; compare (a) with (b), and lines 2–5 with 6–9 in Table 2. Note that in the case that a local patch $\mathcal{L}_c$ is of $N=8$, the proposed and statistics-based methods do not take into account the center pixel intensity $I(\boldsymbol{c})$ at all in coding and weighting. Figure 7 graphically summarizes the performance results from the viewpoint of the feature dimensionalities. The performance gain achieved by the proposed method is larger in the lower dimensional features. This is because the discriminative power per feature element (binary pattern) is higher

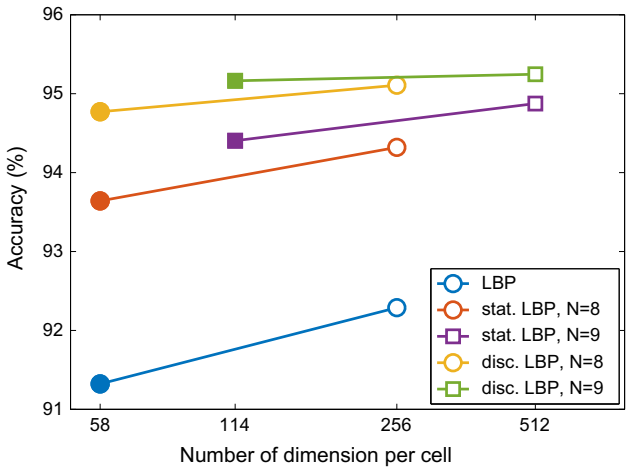


Fig. 7 Performance analysis on the Daimler Chrysler dataset in terms of feature dimensionality. *Empty* and *filled markers* indicate the performances of full binary patterns and uniform patterns, respectively. The *horizontal axis* shows dimensionality in log scale. This figure is best viewed in color (color figure online)

in the proposed method due to the discriminative coding and thus even lower dimensional features work well in classification. Thus, we can say that the proposed method is effective especially for lower dimensional LBP features such as by applying the uniform pattern, which is practically useful due to saving memory usage for features. Based on the trade-off between performance and dimensionality, we recommend to apply the proposed method with the uniform pattern and $N=9$ local patch including the center pixel.

Table 3 Performance comparison to the other methods on the Daimler Chrysler dataset

Method	Acc. (%)
Ours, $N=9$, full	**95.25**
Ours, $N=9$, uniform	**95.16**
HOG [4]	86.41
Maji and Berg[20]	89.25
Vedaldi and Zisserman[35]	91.10
Kobayashi [11]	94.32

The performances by our methods are highlighted by bold

5.1.3 Comparison to the other methods

The proposed method is compared to the other methods than LBP; HOG [4], additive kernel-based feature maps [20,35] and higher-order co-occurrence [11]. Although our method is quite simple, the performance is superior to those methods; note that even the method of $N = 9$ with the uniform pattern outperforms those state-of-the-arts (Table 3).

5.2 INRIA person dataset

Next, the proposed method is tested on the INRIA person dataset [4]. It contains 2416 person annotations and 1218 person-free images for training, and 1132 person annotations and 453 person-free images for test; the person annotations (bounding boxes) are scaled into a fixed size of 64 × 128 pixels. Cell-structured features are computed on cells of 8×8 or 16×16 pixels, producing 8×16 or 4×8 cells on a detection window of 64 × 128 pixels. In each cell, LBP-based features with uniform patterns of $N = 9$ are extracted to reduce the feature dimensionality. The performance is shown in Fig. 8 where for quantifying and comparing methods, we plotted detection error trade-off curves by calculating miss rate and false-positive rate per detection window.

As shown in Fig. 8a, the proposed method outperforms LBP-related methods [16,26] and HOG [4] in both cases of 8 × 8 and 16 × 16 px cells. Note that the method with cells of 16 × 16 pixels produces 3648-dimensional feature vector which is close to HOG dimensionality (3780 dimension). The larger cell of 16 × 16 pixels contains a substantial number of pixels, i.e., LBP codes, to construct features, which statistically contributes to increase robustness of noise-sensitive LBP features; the LBP method becomes even comparable to the statistics-based LBP method [16] as shown in Fig. 8a (comparing dashed lines for 16 × 16 px cells with solid ones for 8×8 px cells). In contrast, the proposed method is superior to the LBP method in any cases due to discriminative coding.

Finally, the LBP-based features are combined with HOG as proposed in [37]; Fig. 8b shows the performance results.

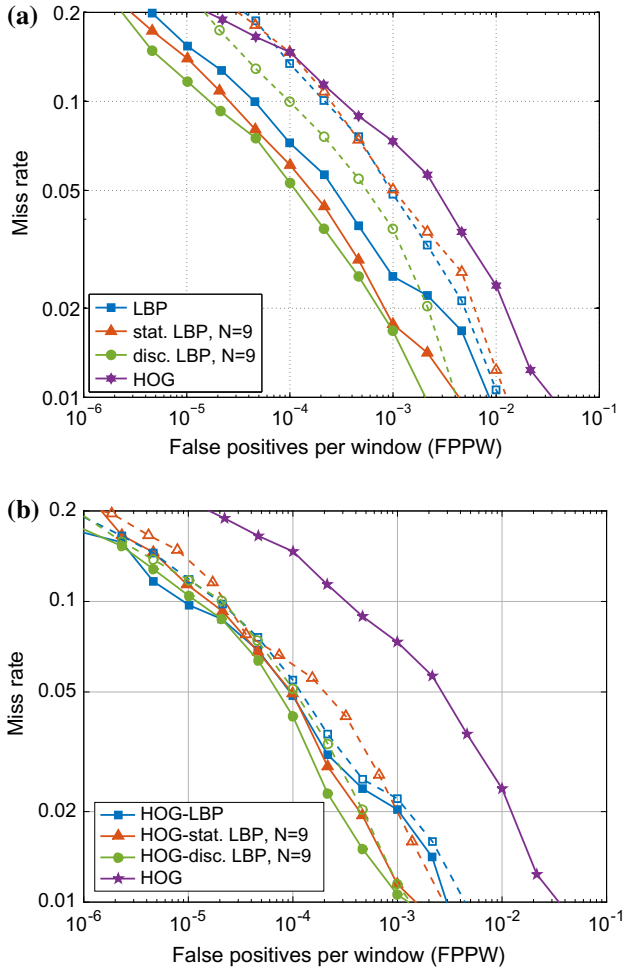


Fig. 8 Performance comparison on the INRIA dataset. The solid lines show the performance of LBP-based features with cells of 8 × 8 pixels, while the *dashed lines* are for cells of 16 × 16 pixels. Note that the uniform patterns are applied to LBP-based features. The performance of single type of feature is shown in (**a**), while that of combined features with HOG is in (**b**). The ordinary HOG-LBP method [37] is denoted by HOG-LBP

The performance is improved by the combination and the proposed method again outperforms the ordinary HOG-LBP [37].

5.3 Face recognition

We tested the method on a face recognition task using FERET dataset [30]. In this experiment, we used frontal face images which are partitioned into the following five sets; *fa* is a gallery set of 1196 persons, *fb* set (1195 images) is taken with different facial expression, *fc* (194 images) is captured under different lighting condition, *dup I* (722 images) is taken later in time, and *dup II* (234 images) is a subset of *dup I* containing images taken at least a year later. The sets other than *fa* are regarded as probe sets on which the classification is performed.

Table 4 Recognition rate (%) on the FERET dataset

Method	*fb*	*fc*	*dup I*	*dup II*
LBP	**98**	84	82	71
Stat. LBP	**98**	87	81	73
Ours	**98**	90	**83**	**76**
Ahonen et al. [1]	97	79	66	64
Zhang et al. [41]	**98**	**97**	74	71
Lei et al. [18]	97	90	71	67
Xie et al. [39]	97	**97**	75	71

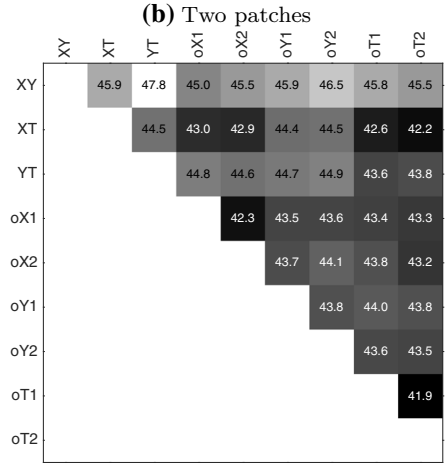
The best performances are highlighted by bold

Table 5 Performance results (%) of the proposed method on various types of patches in the HMDB51 dataset

(a) Single patch

Patch	XY	XT	YT	oX1	oX2	oY1	oY2	oT1	oT2
Acc.	44.1	40.8	43.5	41.0	41.0	43.0	43.4	41.1	40.5

(b) Two patches



(c) Three patches

Patches	XY+XT+YT	XY+YT
Acc.	47.2	**47.8**

An facial image of 130×150 pixels is spatially partitioned into 18×21 cells at each of which the LBP-based feature is extracted. The LBP feature is computed on 8 sampling points along the circle of 3 pixel radius. In classification, since each individual is represented by only one frontal face image in the *fa* gallery set, we apply the exemplar SVM method [13,21]. The method individually trains linear SVM classifiers for respective persons on the *fa* set and an (unknown) input face image is classified into the one of the maximum classification score. It is superior to 1-NN in that individual classifiers contain discriminative information endowed by SVM [13, 21].

The performance results are shown in Table 4. Classification on the *fb* set is the easiest and thus the performance of the LBP method is almost saturated, exhibiting no performance gain by the proposed method. In contrast, on the *fc*, *dup I* and *dup II* sets, the performance is successfully improved; especially, on the *dup I* and *dup II* sets, the method exhibits superior performance to the others.

5.4 Action classification

Lastly, we apply the method to action classification from videos. A video is represented as a sequence of image frames which form volume data in XYT domain, and thus the extended method described in Sect. 3 effectively works to extract local motion descriptors from the XYT volume patches in the framework using dense trajectory [36]. As in [36], we extract plenty of trajectories of 15 frame time length starting from dense spatio-temporal points in an input video sequence. Each trajectory is spatially extended to 32×32 pixels along a X-Y image plane (frame), resulting in $32 \times 32 \times 15$ tube, not "cuboid", which is partitioned into $2 \times 2 \times 2$ cells to extract cell-based volume descriptors followed by dimensionality reduction via PCA. The descriptor features are extracted by the proposed method of $N = 8$ with uniform patterns using joint thresholding (Sect. 3) on the decomposed 2-D patch in $3 \times 3 \times 3$ voxels. As a result, the input video sequence is represented by a bag of local volume descriptors and then the Fisher kernel encoding [32] is applied to them.

We tested the method on the HMDB51 dataset [17] which is collected from a variety of sources ranging from digitized movies to YouTube videos, containing 6766 video sequences of 51 action categories in total. The performance is measured by following the original protocol; we report the averaged classification accuracy over three training-test splits [17] in which there are 70 videos for training and 30 videos for test in each class, and note that in this experiment we use the original videos which are not stabilized.

As described in Sect. 3, it is infeasible to extract LBP-based features directly from the 3-D volume patch, and therefore, the volume patch is decomposed into various types of 2-D patch on which the features are actually computed. We compare those 2-D patches in Table 5. Table 5a shows the performance comparison of various types of 2-D patch. The XY patch performs best and the YT one is the second, which implies the patches related to Y-axis perform well; actually, oY1 and oY2 patches also work well. On the other hand, the patches related to X-axis are relatively inferior, such as in XT, oX1 and oX2. This is due to the difference in horizontal and vertical movement. The human motion is dominated by horizontal movement, e.g., translation, with large displacement along X-axis which is difficult to be properly characterized by the small LBP patch, in this case 3×3

Table 6 Comparison in thresholding strategies on the HMDB51 dataset

Patch	XY	XT	YT	$XY+YT$
Respective thresholding	40.0	39.9	43.3	47.6
Joint thresholding	**44.1**	**40.8**	**43.5**	**47.8**

The better performances are highlighted by bold

Table 7 Performance comparison on the HMDB51 dataset

(a) Comparison on LBP-based methods

Method	LBP	stat. LBP	Ours
Acc.	43.7	47.3	**47.8**

(b) Combination with dense trajectory feature [36]

Method	[17]	[36]	[36]+LBP	[36]+stat. LBP	[36]+Ours
Acc.	23.2	51.2	51.6	52.3	**52.7**

The best performances are highlighted by bold

pixels, on XT slice. In contrast, the vertical movement along Y-axis is relatively small and its discriminative features can be appropriately extracted by the LBP patch for discriminating actions. The combination of patches is also compared in Table 5b which shows that the best performance is obtained by combining XY and YT patches. While the patch combinations on the basis of Y-axis, such as $XY+$oY1 and $XY+$oY2, also improve performance, the $XY+YT$ patch combination is superior since it integrates orthogonal patches of XY and YT favorably compensating motion and shape information to each other; the oY1 and oY2 patches contain the information from X-axis which is already extracted by the XY patch. As shown in Table 5c, the $XY+YT$ combination outperforms the full set of $XY+XT+YT$ which degrades performance by adding the redundant and less-informative patch of XT.

Next, we compared the thresholding strategies described in Sect. 3, *respective* and *joint* thresholding, in Table 6. While in the respective thresholding each patch has its own discriminative threshold in disregard of the others, by which the XY patch results in completely shape (texture) feature, the joint thresholding naively introduces volume information in the threshold shared across all the patches, which leads to performance improvement especially on XY patch, as shown in Table 6. Based on this result, we employ the *joint* thresholding.

By using the patch combination of $XY+YT$, the LBP-based methods are compared in Table 7a, showing that the proposed method is superior to the original LBP and statistics-based LBP as is the case with image classification described in the previous sub-sections. The proposed method is then combined with the features [36] composed of HOG, histogram of optical flow (HOF) and motion boundary histogram (MDH), and favorably improve the performance as shown in Table 7b.

6 Conclusion

In this paper, we have proposed a novel LBP-based method to extract effective features from images and volume data, such as motion images. We generalize the LBP formulation by focusing on the two fundamental processes of coding and weighting, and the proposed method provides a discriminative approach to determine those two fundamentals. In the discriminative approach, LBP coding to binarize pixel intensities by a threshold is regarded as separating a local pixel intensity distribution into two modes, and from that viewpoint the threshold is optimized by maximizing the Fisher discriminant score which is subsequently employed in weighting. So discriminatively optimized thresholds significantly contribute to construct effective features of high robustness to noise and the weight of the discriminant scores efficiently exploit image contrast information to reveal the visual structure of the content (object). The proposed method retains the simple formulation of the original LBP and is also applicable to extract volume features via efficiently decomposing a 3-D volume patch into 2-D patches as well as employing the effective thresholding strategy based on the volume patch. The experimental results on various visual recognition tasks including image and action classification show that the proposed method exhibits favorable performance compared to the other methods.

References

1. Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2007)
3. Chan, C.H., Goswami, B., Kittler, J., Christmas, W.: Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication. IEEE Trans. Inf. Forensics Secur. **7**(2), 602–612 (2012)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Conf. Comput. Vis. Pattern Recognit. **1**, 886–893 (2005)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Hoboken (2001)
6. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
7. Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using lbp variance (LBPV) with global matching. Pattern Recognit. **43**(3), 706–719 (2010)
8. Hafiane, A., Seetharaman, G., Zavidovique, B.: Median binary pattern for texture classification. In: International Conference on Image Analysis and Recognition, pp. 387–398 (2007)

9. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved lbp under bayesian framework. In: International Conference on Image and Graphics, pp. 306–309 (2004)
10. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3D-gradients. In: British Machine Vision Conference, pp. 995–1004 (2008)
11. Kobayashi, T.: Higher-order co-occurrence features based on discriminative co-clusters for image classification. In: British Machine Vision Conference, pp. 64.1–64.11 (2012)
12. Kobayashi, T.: Discriminative local binary pattern for image feature extraction. In: International Conference on Computer Analysis of Images and Patterns, pp. 594–605 (2015)
13. Kobayashi, T.: Three viewpoints toward exemplar SVM. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2765–2773 (2015)
14. Kobayashi, T., Otsu, N.: Image feature extraction using gradient local auto-correlations. In: European Conference on Computer Vision, pp. 346–358 (2008)
15. Kobayashi, T., Otsu, N.: Motion recognition using local auto-correlation of space-time gradients. Pattern Recognit. Lett. **33**(9), 1188–1195 (2012)
16. Kobayashi, T., Ye, J.: Acoustic feature extraction by statistics based local binary pattern for environmental sound classification. In: International Conference on Acoustic, Speech and Signal Processing, pp. 3076–3080 (2014)
17. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: International Conference on Computer Vision, pp. 2556–2563 (2011)
18. Lei, Z., Li, S.Z., Chu, R., Zhu, X.: Face recognition with local gabor textons. In: International Conference on Biometrics, pp. 49–57 (2007)
19. Lowe, D.: Distinctive image features from scale invariant features. Int. J. Comput. Vis. **60**, 91–110 (2004)
20. Maji, S., Berg, A.: Max-margin additive classifiers for detection. In: International Conference on Computer Vision, pp. 40–47 (2009)
21. Malisiewicz, T., Gupta, A., Efros, A.: Ensemble of exemplar-svms for object detection and beyond. In: International Conference on Computer Vision, pp. 89–96 (2011)
22. Mattivi, R., Shao, L.: Human action recognition ising LBP-TOP as sparse spatio-temporal feature descriptor. In: International Conference on Computer Analysis of Images and Patterns, pp. 740–747 (2009)
23. Munder, S., Gavrila, D.M.: An experimental study on pedestrian classification. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1863–1868 (2006)
24. Nanni, L., Brahnam, S., Lumini, A.: Local ternary patterns from three orthogonal planes for human action classification. Expert Syst. Appl. **38**(5), 5125–5128 (2011)
25. Nanni, L., Lumini, A., Brahnam, S.: Local binary patterns variants as texture descriptors for medical image analysis. Artif. Intell. Med. **49**(2), 117–125 (2010)
26. Ojala, T., Pietikäinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: International Conference on Pattern Recognition, pp. 582–585 (1994)
27. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recognit. **29**(1), 51–59(1998)
28. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
29. Otsu, N.: Discriminant and least squares threshold selection. In: International Conference on Pattern Recognition, pp. 592–596 (1978)
30. Phillips, P., Wechsler, H., Huang, J., Rauss, P.: The feret database and evaluation procedure for face recognition algorithms. Image Vis. Comput. **16**(10), 295–306 (1998)
31. Pietikäinen, M., Zhao, G., Hadid, A., Ahonen, T.: Computer Vision Using Local Binary Pattern. Springer, New York (2011)
32. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
33. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: ACM Conference on Multimedia, pp. 357–360 (2007)
34. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Process. **19**(6), 1635–1650 (2010)
35. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
36. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vision **103**, 60–79 (2013)
37. Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: International Conference on Computer Vision, pp. 32–39 (2009)
38. Wu, J., Rehg, J.M.: Centrist: a visual descriptor for scene categorization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1489–1501 (2011)
39. Xie, S., Shan, S., Chen, X., Meng, X., Gao, W.: Learned local gabor patterns for face representation and recognition. Sig. Process. **89**(12), 2333–2344 (2009)
40. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: European Conference on Computer Vision, pp. 151–158 (1994)
41. Zhang, W., Shan, S., Gao, W., Zhang, H.: Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In: International Conference on Computer Vision, pp. 786–791 (2005)
42. Zhao, G., Ahonen, T., Matas, J., Pietikäinen, M.: Rotation-invariant image and video description with local binary pattern features. IEEE Trans. Image Process. **21**(4), 1465–1477 (2012)
43. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 915–928 (2007)

Takumi Kobayashi received Ms. Eng. from University of Tokyo in 2005 and Dr. Eng. from University of Tsukuba in 2009. He was a researcher at Toshiba Corporation in 2006 and then joined National Institute of Advanced Industrial Science and Technology (AIST), Japan, in 2007. His research interest includes pattern recognition.