

RESEARCH

Open Access

Concave-convex local binary features for automatic target recognition in infrared imagery

Junding Sun¹, Guoliang Fan^{2*}, Liangjiang Yu² and Xiaosheng Wu¹

Abstract

This paper presents a novel feature extraction algorithm based on the local binary features for automatic target recognition (ATR) in infrared imagery. Since the inception of the local binary pattern (LBP) and local ternary pattern (LTP) features, many extensions have been proposed to improve their robustness and performance in a variety of applications. However, most attentions were paid to improve local feature extraction with little consideration on the incorporation of global or regional information. In this work, we propose a new concave-convex partition (CCP) strategy to improve LBP and LTP by dividing local features into two distinct groups, i.e., concave and convex, according to the contrast between local and global intensities. Then two separate histograms built from the two categories are concatenated together to form a new LBP/LTP code that is expected to better reflect both global and local information. Experimental results on standard texture images demonstrate the improved discriminability of the proposed features and those on infrared imagery further show that the proposed features can achieve competitive ATR results compared with state-of-the-art methods.

Keywords: Local binary pattern; Local ternary pattern; Automatic target recognition; Concave-convex partition

Introduction

Automatic target recognition (ATR) is an important and challenging problem for a wide range of military and civilian applications. Many ATR algorithms have been proposed for forward-looking infrared (FLIR) imagery which can be roughly classified into two groups, i.e., *learning-based* and *model-based* [1]. For the learning-based methods, a classifier or a subspace representation which is learned for a set of labeled training data is used for target recognition and classification [2,3]. On the other hand, the model-based approaches involve a set of target templates or feature maps created from CAD models or a model database and then match them with observed features to fulfill the ATR task [4-8]. Li et al. [1] gave an excellent survey of traditional ATR algorithms, including convolution neural network (CNN), principal component analysis (PCA), linear discriminant analysis

(LDA), learning vector quantization (LVQ), modular neural networks (MNN), and two model-based algorithms using the Hausdorff metric and geometric hashing. In [9], Patel et al. introduced an interesting ATR algorithm that was motivated by sparse representation-based classification (SRC) [10], which outperforms the traditional ones with promising results. Recently, there are also many hybrid vision approaches that combine learning-based and model-based ideas for object tracking and recognition in visible-band images [11-13].

As one of the learning approaches, the ATR task has also been cast as a texture analysis problem due to rich texture characteristics in most infrared imagery. Various texture-like features, including geometric, topological, and spectral features, were proposed for appearance-based ATR as reviewed in [14]. Moreover, wavelet-based PCA and independent component analysis (ICA) methods were developed in [15]. Both wavelet and Gabor features were used for target detection in [16]. A texture feature coding method (TFCM) was proposed for synthetic aperture radar (SAR)-based ATR in [17]. In this work, we are interested in the texture-based ATR approach. Particularly,

*Correspondence: guoliang.fan@okstate.edu

²School of Electrical and Computer Engineering, Oklahoma State University, Stillwater 74078, USA

Full list of author information is available at the end of the article

we are focused on how to extract effective yet simple local binary pattern (LBP) operators for infrared ATR. This research is motivated by the rapid growth of the development of various LBP features and their promising results in several ATR applications, such as maritime target detection and recognition in [18], infrared building recognition in [19], and ISAR-based ATR in [20].

The LBP operator was proposed by Ojala et al. in [21], which is based on the so-called texture unit (TU) introduced by Wang and He in [22]. LBP has been proved a robust and computationally simple approach to describe local structures and has been extensively exploited in many applications, such as texture analysis and classification [23-30], face recognition [31-36], motion analysis [37-39], and medical image analysis [40-42]. Since Ojala's original work, the LBP methodology has been developed with large number of extensions for improved performance. For example, Ojala et al. [43] extended the basic LBP to multiresolution gray scale, uniform, and rotation invariant patterns in order to achieve rotation invariance, optional neighborhoods, and strong discriminative capability. Tan and Triggs [32] proposed the local ternary pattern (LTP) to quantize the intensity difference between a pixel and its neighbors into three levels to further enhance the robustness to noise. LTP has been proven effectively in face recognition. Ahonen and Pietikäinen [44] introduced soft LBP (SLBP) to enhance the feature robustness in the sense that a small change in an input image causes only a small change in the feature extraction output by employing fuzzy theory. Guo et al. [45] proposed a complete LBP (CLBP), which fuses the signs (the same as the basic LBP) with the absolute value of the local intensity difference and the central gray level to improve the discriminative capability. Liao et al. [46] proposed a multi-block LBP (MB-LBP), which, instead of comparing pixels, compares local average intensities of neighboring sub-regions to capture not only the microstructures but also the macrostructures. Wolf et al. [47] proposed a similar scheme as [46], called three-patch LBP (TP-LBP) and four-patch LBP (FP-LBP), which involves a unique distance function to compare local blocks (patches), instead of a single pixel as in [21] or average intensity as in [46].

In this work, we are interested in the applicability of LBP/LTP for infrared ATR. We also take a different perspective to improve LBP/LTP features in this new context. Almost all the extensions of the basic LBP features mentioned above focus on local feature extraction in a small neighborhood. However, the local feature may not be sufficient for rotation invariant texture classification and other applications that require some information in a larger area. We will give an example shortly, where it is shown that the two local patches with entirely different gray levels may have the exact same LBP code. Therefore, we propose a concave-convex partition (CCP)

scheme to involve global information (i.e., the global mean) for LBP/LTP encoding. We also introduce a simple yet effective blocking technique to further improve the feature discriminability for infrared ATR. We evaluate the newly proposed CCP technique in two standard texture databases, and we compare also our CCP-based local binary features (LBP/LTP) with the latest sparsity-based ATR algorithm proposed in [9] where we have used four implementations from SparseLab [48] and SPGL1 [49].

The rest of the paper is organized as follows. In Section 'Brief review of LBP-based methods', we briefly discussed the basic LBP and LTP. Section 'Concave-convex local binary features' discusses the limitation of traditional LBP features and presents the new CCP scheme in detail. Section 'Experiments and discussions' reports the experimental results on two texture databases and a recent ATR database. Finally, we conclude our study in Section 'Conclusions'.

Brief review of LBP-based methods

Since its origination, there are many extensions of LBP. In this section, we only give a brief introduction of the basic LBP and one of its extensions, LTP.

Local binary pattern

The basic LBP operator is first introduced in [21] for texture analysis. It works by thresholding a neighborhood with the gray level of the central pixel. The LBP code is produced by multiplying the thresholded values by weights given by powers of 2 and adding the results in a clockwise way. It was extended to achieve rotation invariance, optional neighborhoods, and stronger discriminative capability in [43]. The basic LBP is commonly referred as $LBP_{P,R}$, which is written as

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(p_i - p_c) \times 2^i, \quad (1)$$

where $s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise,} \end{cases}$, P is the number of sampling pixels on the circle, R is the radius of the circle, p_c corresponds to the gray value of the central pixel, and p_i corresponds to the gray value of each sampling pixels on the circle. Figure 1 gives an example of calculating LBP code with $P = 8, R = 1$. In Figure 2, three different neighborhoods with a varying number of samples P and different neighborhood radii R are shown. It should be noted that the gray value of neighboring pixels which do not fall exactly on the grid is calculated by linear interpolation of the neighboring pixels.

The classical description of the rotation invariant LBP operator [43] can be defined as below:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{i=0}^{P-1} s(p_i - p_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases}, \quad (2)$$

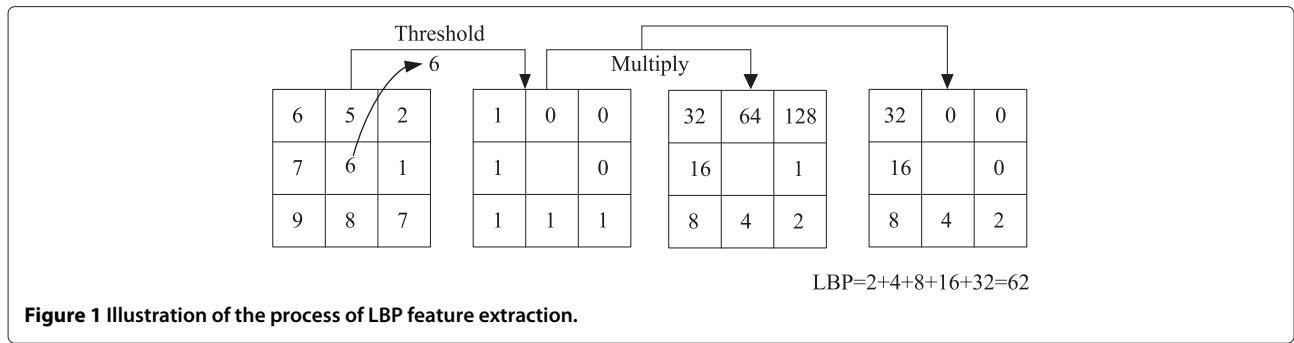


Figure 1 Illustration of the process of LBP feature extraction.

where the superscript *riu2* refers to the use of rotation invariant uniform patterns that have a *U* value ($U \leq 2$). The uniformity measure *U* corresponds to the number of transitions from 0 to 1 or 1 to 0 between successive bits in the circular representation of the binary code $LBP_{P,R}$, which is defined as

$$U(LBP_{P,R}) = |s(p_{P-1} - p_c) - s(p_0 - p_c)| + \sum_{i=1}^{P-1} |s(p_i - p_c) - s(p_{i-1} - p_c)| \quad (3)$$

All nonuniform patterns are classified as one pattern for $LBP_{P,R}^{riu2}$. The mapping from $LBP_{P,R}$ to $LBP_{P,R}^{riu2}$, which has $P + 2$ distinct output values, can be implemented with a lookup table.

Local ternary pattern

Because the gray value of the central pixel is directly used as a threshold, the LBP is sensitive to noise, especially in a smooth image region. To address this problem, Tan and Triggs [32] extended the original LBP to a version with three-value codes, which is called local ternary pattern. In LTP, the indicator $s(x)$ as in Equation 1 is given as

$$s(p_i - p_c - \tau) = \begin{cases} 1 & p_i - p_c \geq \tau \\ 0 & |p_i - p_c| < \tau \\ -1 & p_i - p_c < -\tau, \end{cases} \quad (4)$$

where τ is a user-set threshold. In order to reduce the dimension of LTP, a coding scheme is also represented

by Tan and Triggs [32] by splitting each ternary pattern into two parts: the positive part and the negative part, as illustrated in Figure 3. Though the LTP codes are more resistant to noise, it is no longer strictly invariant to gray-level transformations.

Concave-convex local binary features

In this section, we first discuss the limitation of the traditional LBP-based operators, and then we propose a new method called CCP to improve LBP/LTP features.

Limitation of traditional LBP-based methods

Until now, most LBP extensions concentrate on how to improve the discernibility of the feature extracted from the local neighborhoods. As a result, the emphasis of such LBP-based methods is on the local features. However, is local information sufficient for rotation invariant texture classification and other texture analysis applications? Figure 4 shed some light on this issue. Figure 4a shows two 8 neighborhoods in an texture image (160×160). The neighborhoods (a1) and (a2) in Figure 4a have the exact same LBP code (i.e., 219), positive LTP code (145), and negative LTP code (32), although they have entirely different visual perception. That is to say, the operators LBP and LTP mentioned above cannot draw any distinction between (a1) and (a2) in Figure 4a.

For further explanation, we refer to the average gray value of the neighborhood because it is clear that the

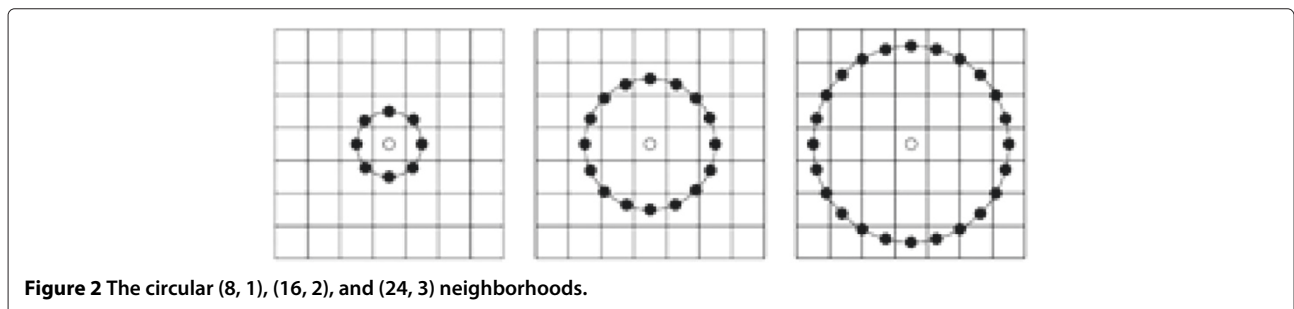


Figure 2 The circular (8, 1), (16, 2), and (24, 3) neighborhoods.

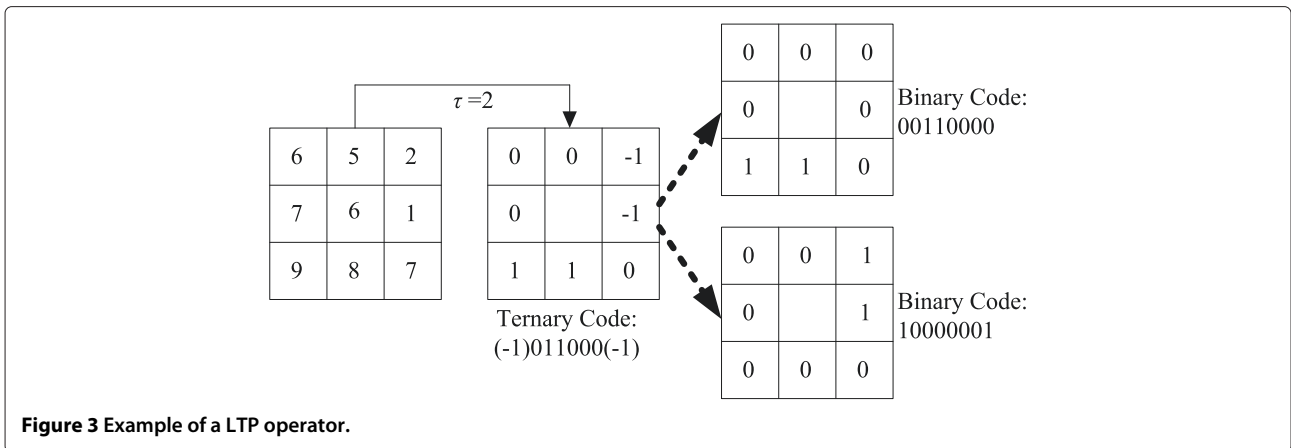


Figure 3 Example of a LTP operator.

average gray value of (a1) and (a2) in Figure 4a is very different. Figure 4b gives the distribution of the average gray value for the neighborhoods ($P = 8$ and $R = 1$) which have the same LBP code ($LBP_{8,1} = 4$). There are totally 649 neighborhoods and the maximum and minimum of the average gray value are 228.1 and 54.7, respectively. Obviously, many neighborhoods with the same LBP code have different visual perception because their average gray values differ greatly.

Most of other extensions of LBP also have the same limitation as LBP and LTP. The main reason is that such operators concentrate only on local features and the global feature of the image is neglected. Therefore, the basic LBP and most of its extensions cannot distinguish the patterns with different global intensities and with the same local features. There is little effort to address this problem directly so far. The recently proposed CLBP feature in [45] implicitly addresses this problem by involving an additional binary code that encodes the comparison between the central pixel and the global average. In this paper, we study this problem explicitly by dividing the local features into two groups to reflect the relative contrast between the local and global intensities.

Concave-convex partition

Since the neighborhoods with different visual perception may have the same binary code by the LBP-based operators, we can differentiate them from each other before computing the local binary codes. For example, we can use the bit-plane decomposition [50] to compute the local binary codes on each 1-b plane, and then integrate them together. In that case, the global information can be carried by the binary codes. The price paid by this method is the doubled feature dimensionality. Of course, there could be other methods which are able to achieve a similar effect. However, to keep the simplicity and effectiveness of the basic LBP, we propose a simple yet effective method in this paper to address the weakness. The new method is called concave-convex partition or CCP. It is known that the average gray level is a widely accepted statistical parameter for texture analysis. We also chose it to denote the gray-level variation of a neighborhood. Let $\mu_{i,j}$ denote the average gray level of a neighborhood (i, j), which is given as follows:

$$\mu_{i,j} = \frac{1}{P+1} \left(p_{ij} + \sum_{k=1}^P p_k \right). \quad (5)$$

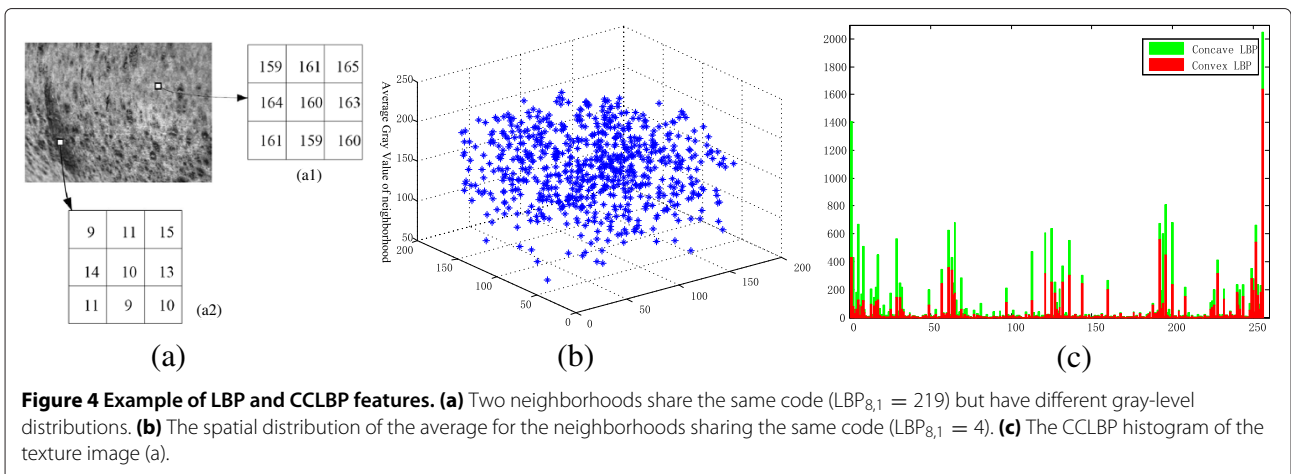


Figure 4 Example of LBP and CCLBP features. (a) Two neighborhoods share the same code ($LBP_{8,1} = 219$) but have different gray-level distributions. (b) The spatial distribution of the average for the neighborhoods sharing the same code ($LBP_{8,1} = 4$). (c) The CCLBP histogram of the texture image (a).

Here, we further manifest the limitations of traditional LBP-based methods on an infrared chip. Figure 5a shows an infrared chip with a target in the center (40×75), and Figure 5b presents the distribution of $\mu_{i,j}$ for all neighborhoods ($P = 8$ and $R = 1$) which have the same rotation invariant and uniform LBP code ($LBP_{8,1}^{riu2} = 3$). There are totally 347 neighborhoods in the image and the maximum and minimum of $\mu_{i,j}$ are 580.37 and 474.57, respectively. It is also obvious that there are many local neighborhoods with different visual perception but having the same code. In order to address this problem, we introduce the CCP technique based on $\mu_{i,j}$ as follows.

Definition of CCP

Let $\omega(i, j)$ be a neighborhood centered at pixel (i, j) in an image and $\mu_{i,j}$ be its average gray value. Given α as a CCP threshold, if $\mu_{i,j} < \alpha$, the central pixel $p_{i,j}$ is regarded as a concave pixel and the neighborhood $\omega(i, j)$ as a concave neighborhood. Otherwise, $p_{i,j}$ is regarded as a convex pixel and $\omega(i, j)$ is as a convex neighborhood. The set of all concave neighborhoods are denoted as $\Omega_1 = \{(i, j) | \mu_{i,j} < \alpha\}$, and all convex neighborhoods are denoted as $\Omega_2 = \{(i, j) | \mu_{i,j} \geq \alpha\}$. Now the key of CCP is to choose the optimal parameter α which should provide the least square error approximation to all local means:

$$\hat{\alpha} = \operatorname{argmin}_{\alpha} \sum_i \sum_j (\mu_{i,j} - \alpha)^2. \tag{6}$$

Hence, it is straightforward to get $\hat{\alpha}$ as the mean of all the local means $\mu_{i,j}$ as

$$\hat{\alpha} = \frac{1}{N_I \times N_J} \sum_i \sum_j \mu_{i,j}, \tag{7}$$

where $N_I \times N_J$ is the number of local neighborhoods in the image. To speed up the computation, we further use the global mean μ to approximate $\hat{\alpha}$.

The proposed CCP technique can be applied to nearly all LBP features and their extensions. The CCP-based

feature extraction has two steps. Firstly, the local binary code of each neighborhood is computed by a LBP operator. Secondly, the codes are divided into two categories according to CCP. After that, the original feature histogram of an image is decomposed into two parts, which are called the concave histogram (the statistics of the neighborhoods where $\mu_{i,j} < \mu$) and the convex histogram (the statistics of the neighborhoods where $\mu_{i,j} \geq \mu$).

It is expected that CCP-based LBP features will be enriched by incorporating the global mean. For example, Figure 4c gives the CCP-based LBP distribution of the texture image in Figure 4a, the LBP histogram ($P = 8, R = 1$) is decomposed into a concave one and a convex one. Figure 5c presents the CCP-based LBP distribution of the infrared chip shown in Figure 5a; the LBP histogram that is rotation invariant and uniform (represented by $LBP_{8,1}^{riu2}$) is decomposed into a concave one and a convex one. It is worth mentioning that we have tried CCP with more than two feature groups without further improvement, showing that the binary CCP is the most cost-effective one.

Dissimilarity measure

Various metrics have been presented to evaluate the dissimilarity between two histograms. As most LBP-based algorithms, we chose the chi square distance as the dissimilarity measure, which is defined as

$$d(H, B) = \sum_{i=1}^K \frac{(h_i - b_i)^2}{h_i + b_i}, \tag{8}$$

where $H = \{h_i\}$ and $B = \{b_i\}$ ($i = 1, 2, \dots, K$) denote the two feature histograms, and K is the number of bins in the histogram.

Experiments and discussions

In this section, we first evaluate and compare LBP [43], LTP [32], SLBP [44], and CLBP [45] along with their CCP-enhanced versions which are called CCP-based LBP

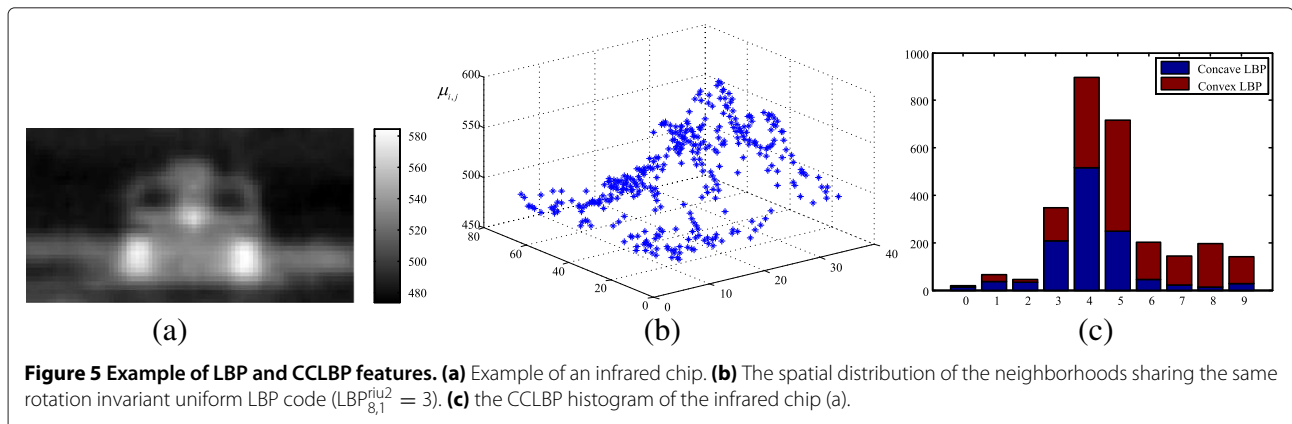


Table 1 Dimension comparison of each LBP feature

	$(P, R) = (8, 1)$	$(P, R) = (16, 2)$	$(P, R) = (24, 3)$
$LBP_{P,R}^{riu2}$	10	18	26
$CCLBP_{P,R}^{riu2}$	20	36	52
$LTP_{P,R}^{riu2}$	20	36	52
$CCLTP_{P,R}^{riu2}$	40	72	104
$SLBP_{P,R}^{riu2}$	10	18	26
$CCSLBP_{P,R}^{riu2}$	20	36	52
$CLBP_{P,R}^{riu2}$	100	324	676
$CCCLBP_{P,R}^{riu2}$	200	648	1,352

(CCLBP), CCP-based LTP (CCLTP), CCP-based SLBP (CCSLBP), and CCP-based CLBP (CCCLBP) for texture classification to demonstrate the usefulness of the proposed CCP technique. Then, we focus on LTP and CCLTP to examine their effectiveness for infrared ATR where four SRC-based methods are involved for performance evaluation. The experiments were conducted on a PC with the AMD Phenom(tm)II Processor (3.41GHz) (Advanced Micro Devices, Inc., Sunnyvale, CA, USA), 8 GB RAM, and the 64-b Win7 Operating System.

Three different multiresolutions are used for the eight features tested in this work, $P = 8$ and $R = 1$, $P = 16$ and $R = 2$, and $P = 24$ and $R = 3$. To simplify the feature computation, LTP is split into two LBPs as mentioned in [32], i.e., positive LBPs and negative LBPs. Then two histograms are built and concatenated into one histogram (shown in Figure 3) as the image feature, which is denoted as $LTP_{P,R}^{riu2}$. For CLBP, we choose $CLBP_{P,R}^{riu2}/M_{P,R}^{riu2}$ operator in the paper. We compare all eight LBP features in terms of their dimension in Table 1 where it is shown that both CLBP and CCCLBP have a significantly higher dimension than others.

Experiments on texture classification

For texture classification, we chose two commonly used textures databases, i.e., the Outex database [51] and

the Columbia-Utrecht Reflection and Texture (CURET) database [52].

Experimental results on Outex database

For the Outex database [51], we chose Outex_TC_0010 (TC10) and Outex_TC_0012 (TC12), where TC10 and TC12 contain the same 24 classes of textures collected under three different illuminants ('horizon,' 'inca,' and 't184') and nine different rotation angles (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, and 90°). There are 20 nonoverlapping 128×128 texture samples for each class under each situation. For TC10, samples of illuminant inca and angle 0° in each class were used for classifier training and the other eight rotation angles with the same illuminant were used for testing. Hence, there are 480 (24×20) models and 3,840 ($24 \times 8 \times 20$) validation samples. For TC12, all the $24 \times 20 \times 9$ samples captured under illumination t184 or horizon were used as the test data. Table 2 gives the experimental results of different LBP features, where t represents the test setup of illuminant t184 and h represents horizon. It is clear that CCP can improve the LBP, LTP, SLBP, and CLBP greatly. For $P = 8$ and $R = 1$, $P = 16$ and $R = 2$, and $P = 24$ and $R = 3$, $CCLBP_{P,R}^{riu2}$, $CCLTP_{P,R}^{riu2}$, $CCSLBP_{P,R}^{riu2}$, and $CCCLBP_{P,R}^{riu2}/M_{P,R}^{riu2}$ have an averaged accuracy improvement of 12.5%, 5.2%, 7.7%, and 2.9% over their original versions, respectively.

Experimental results on CURET database

The CURET database includes 61 classes of textures captured at different viewpoints and illumination orientations [52], and 92 images in each class are selected from the images shot from a viewing angle of less than 60°. To get statistically significant experimental results, N training images ($N = 46, 23, 12$, and 6) were randomly chosen from each class while the remaining ($92 - N$) images were used as the test set. The average accuracy over 23 random splits are listed in Table 3. Similar conclusions observed from TC10 and TC12 can also be drawn for the

Table 2 Classification accuracy (%) on TC10 and TC12 texture sets using different LBP operators

	$(P, R) = (8, 1)$			$(P, R) = (16, 2)$			$(P, R) = (24, 3)$					
	TC10	TC12		Average	TC10	TC12		Average	TC10	TC12		Average
		t	h			t	h			t	h	
$LBP_{P,R}^{riu2}$	84.81	65.46	63.68	71.31	89.40	82.26	75.20	82.28	95.07	85.04	80.78	86.96
$CCLBP_{P,R}^{riu2}$	95.96	90.57	90.97	92.50	94.81	90.74	89.46	91.67	97.81	93.42	90.87	94.03
$LTP_{P,R}^{riu2}$	94.14	75.87	73.95	81.32	96.95	90.16	86.94	91.35	98.20	93.58	89.42	93.73
$CCLTP_{P,R}^{riu2}$	96.87	86.96	88.10	90.64	98.20	94.53	94.46	95.73	98.75	95.67	92.91	95.77
$SLBP_{P,R}^{riu2}$	90.63	72.31	70.63	77.86	92.99	87.13	71.67	87.26	96.41	90.05	85.74	90.73
$CCSLBP_{P,R}^{riu2}$	95.94	87.41	87.64	90.33	97.50	92.73	92.59	94.27	97.86	94.26	91.20	94.44
$CLBP_{P,R}^{riu2}$	94.66	82.75	83.14	86.85	97.89	90.55	91.11	93.18	99.32	93.58	93.35	95.41
$CCCLBP_{P,R}^{riu2}$	97.29	90.20	92.93	93.47	98.80	92.50	93.49	94.93	98.72	94.81	94.28	95.93

Table 3 Classification accuracy (%) on CuRet database using different LBP operators

	$(P, R) = (8, 1)$				$(P, R) = (16, 2)$				$(P, R) = (24, 3)$			
	46	23	12	6	46	23	12	6	46	23	12	6
$LBP_{P,R}^{riu2}$	70.06	65.77	54.21	47.33	74.49	70.40	58.55	51.23	75.71	72.51	61.52	54.64
$CCLBP_{P,R}^{riu2}$	83.98	78.71	65.47	57.57	82.96	77.64	64.62	56.59	82.30	78.18	66.53	58.95
$LTP_{P,R}^{riu2}$	77.36	71.77	58.33	49.84	82.54	76.29	62.54	54.21	83.15	77.85	64.37	56.40
$CCLTP_{P,R}^{riu2}$	83.45	76.90	62.86	54.18	86.21	79.37	65.46	56.71	86.58	80.58	67.26	59.02
$SLBP_{P,R}^{riu2}$	71.45	62.75	53.61	39.55	75.55	66.64	58.26	44.11	77.26	67.24	58.67	47.16
$CCSLBP_{P,R}^{riu2}$	77.98	68.90	59.90	44.45	81.18	70.90	61.27	47.94	82.50	72.65	65.25	50.90
$CLBP_{P,R}^{riu2}$	85.78	79.69	66.84	58.61	86.28	80.72	67.39	59.02	86.25	81.47	68.54	60.36
$CCCLBP_{P,R}^{riu2}$	88.19	81.24	67.77	59.36	88.41	81.79	68.29	60.11	88.66	83.22	70.47	62.57

experimental results on the CURET dataset. $CCLBP_{P,R}^{riu2}$, $CCLTP_{P,R}^{riu2}$, $CCSLBP_{P,R}^{riu2}$ and $CCCLBP_{P,R}^{riu2}/M_{P,R}^{riu2}$ get an averaged accuracy improvement of 8.1%, 3.6%, 5.1%, and 1.6% over their original versions, respectively.

The results on the two texture databases show the effectiveness of the proposed CCP technique to improve various LBP features. It is worth mentioning that without CCP enhancement, CLBP shows the best performance and LTP is the second. However, the dimension of CLBP is more than 10 times of that LTP. With CCP enhancement,

CCCLBP is still moderately better than CCLTP. However, the high dimension of CCCLBP may limit its potential in real applications where multiple local features may be needed to deal with nonstationarity in an image. CCLTP has a more balanced trade-off between complexity and performance.

Experiments for infrared ATR

This section reports the evaluation of the eight LBP features in the context of infrared ATR. The LBP features of

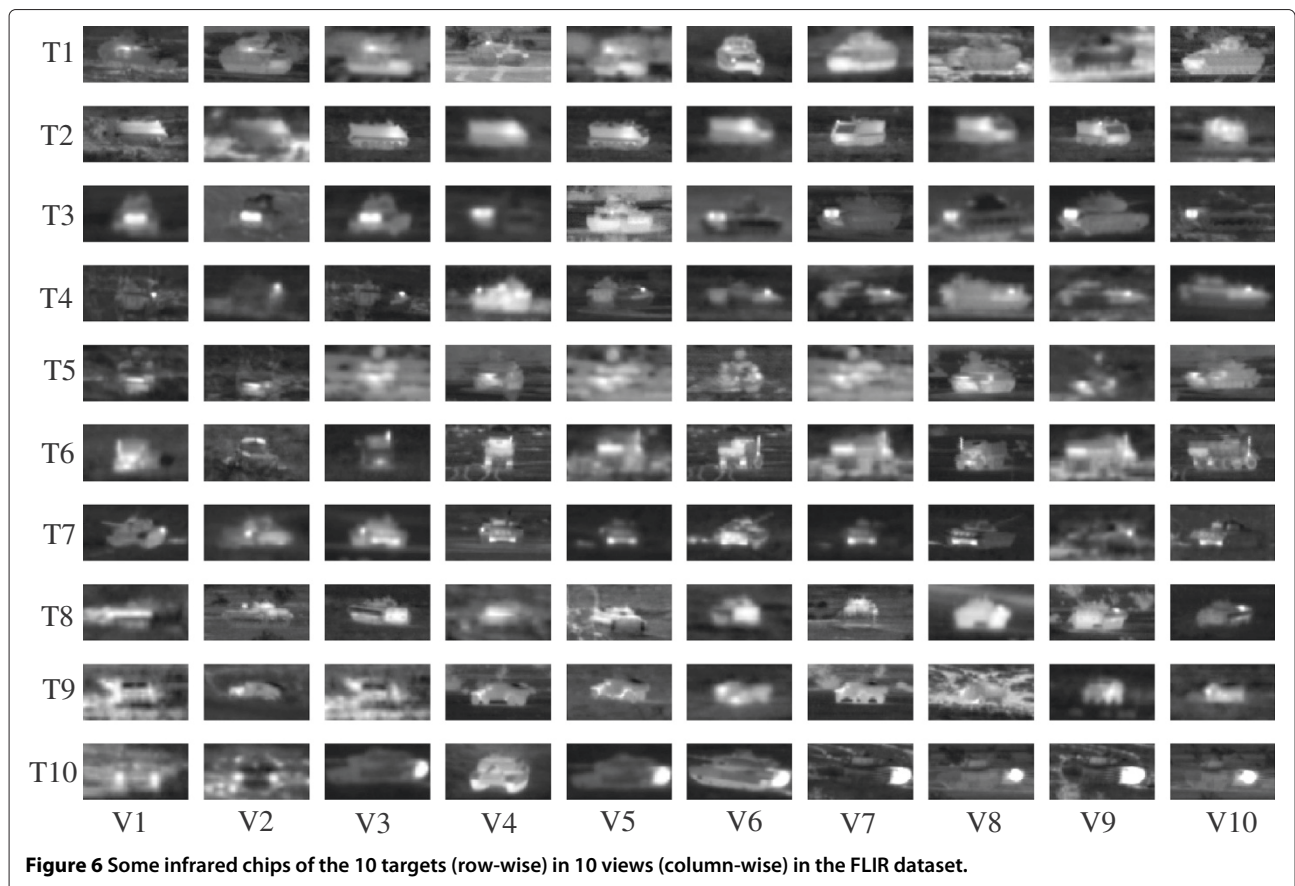


Figure 6 Some infrared chips of the 10 targets (row-wise) in 10 views (column-wise) in the FLIR dataset.

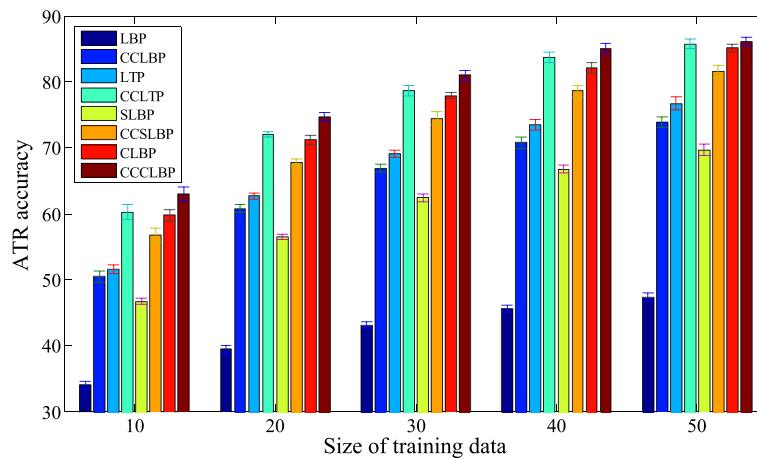


Figure 7 Recognition accuracy comparison for eight operators. LBP, CCLBP, LTP, CCLTP, SLBP, CCSLBP, CLBP, and CCCLBP.

three different scales ($P = 8$ and $R = 1$, $P = 16$ and $R = 2$, and $P = 24$ and $R = 3$) are concatenated together as an image feature which are denoted as $LBP_{8,1+16,2+24,3}^{riu2}$ (LBP), $CCLBP_{8,1+16,2+24,3}^{riu2}$ (CCLBP), $LTP_{8,1+16,2+24,3}^{riu2}$ (LTP), $CCLTP_{8,1+16,2+24,3}^{riu2}$ (CCLTP), $SLBP_{8,1+16,2+24,3}^{riu2}$ (SLBP), $CCSLBP_{8,1+16,2+24,3}^{riu2}$ (CCSLBP), $\{CLBP_S/M\}_{8,1+16,2+24,3}^{riu2}$ (CLBP), $\{CCCLBP_S/M\}_{8,1+16,2+24,3}^{riu2}$ (CCSLBP), respectively. The Comanche (Boeing-Sikorsky, USA) FLIR dataset is used here, similar to [9]. There are 10 different military targets denoted as T_1, T_2, \dots, T_{10} . For

each target, there are 72 orientations, corresponding to aspect angles of $0^\circ, 5^\circ, \dots, 355^\circ$. The dataset contains 437 to 759 images (40×75) for each target type, totally 6,930 infrared chips. In Figure 6, we show some infrared chips for 10 targets under 10 different views.

Comparison of LBP, CCLBP, LTP, CCLTP, SLBP, CCSLBP, CLBP, and CCCLBP

In this experiment, we randomly chose about 10% (718 chips), 20% (1,436 chips), 30% (2,154 chips), 40% (2872

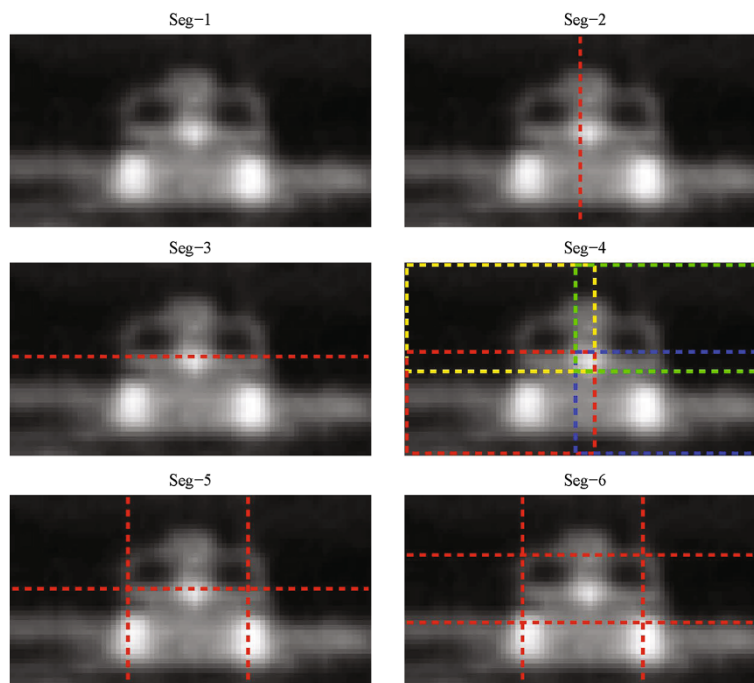


Figure 8 Six blocking methods to divide an infrared chip into multiple segments.

chips), and 50% (3590 chips) target chips of each target class as training data. The remaining 90%, 80%, 70%, 60%, and 50% images in the dataset are set as testing data, respectively. The mean and variance of the recognition accuracy averaged by 10 trials are given in Figure 7. It can be seen from the experimental results that

- With CCP enhancement, the accuracy improvements averaged over five training datasets of CCLBP, CCLTP, CCLBP, and CCCLBP are 23.25%, 9.22%, 11.43%, and 2.92%, respectively, compared with their original versions. A much less improvement of CCCLBP over CLBP is likely due to its high-dimension nature that may reduce the benefit of CCP.
- The experimental result also shows that eight LBP features, i.e., LBP, LTP, SLBP, CLBP, CCLBP, CCLTP, CCLSLBP, and CCCLBP, are robust for infrared ATR, because they are fairly stable in 10 random trials for each case.
- $\{CCCLBP_S/M\}_{8,1+16,2+24,3}^{riu2}$ gives the best performance among the eight operators, followed by $CCLTP_{8,1+16,2+24,3}^{riu2}$. However, the dimension of $\{CCCLBP_S/M\}_{8,1+16,2+24,3}^{riu2}$ is 2,200 which is over 10 times as the dimension of $CCLTP_{8,1+16,2+24,3}^{riu2}$ (216). On the other hand, the recognition accuracy of $CCLTP_{8,1+16,2+24,3}^{riu2}$ and $\{CCCLBP_S/M\}_{8,1+16,2+24,3}^{riu2}$ is getting closer as training data increase. Therefore, we will further make some detailed comparisons between LTP/CCLTP and other recent sparsity-based ATR methods in the following experiments.

Recognition rates of different blocking methods

The analysis reported in [53] shows that the holistic LBP histogram is appropriate for standard texture analysis but

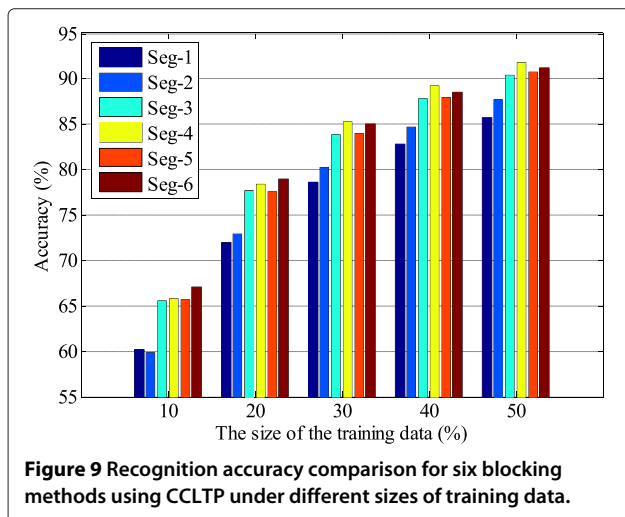


Figure 9 Recognition accuracy comparison for six blocking methods using CCLTP under different sizes of training data.

Table 4 Feature dimensionality and average feature extraction time per chip for different blocking schemes shown in Figure 8

	Seg-1	Seg-2	Seg-3	Seg-4	Seg-5	Seg-6
Dimension	216	432	432	864	1296	1944
Average feature extraction time (s)	0.033	0.036	0.036	0.043	0.049	0.059

not suitable for facial recognition. The reason is that the texture images have uniform or homogeneous property, however, the face images have large spatial variations due to distinct facial features. Therefore, a facial image is divided into some blocks. Then a LBP histogram is extracted from each block and all LBP histograms are concatenated together as a feature representation for face recognition. Similarly, the targets in infrared chips have strong spatial variations. A holistic LBP or LTP histogram is not suitable for ATR. Therefore, we divide an infrared chip into different blocks and extract a LTP/CCLTP histogram from each block.

We studied six different blocking methods, as illustrated in Figure 8. Seg-1 denotes the whole chip without blocking. Seg-2 divides a chip into the left-right blocks. Seg-3 segments a chip into the top-bottom blocks. Seg-4 divides a chip into four quadrants that are slightly overlapped. Seg-5 segments a chip into six blocks. Seg-6 partitions a chip into nine blocks. The operator, $CCLTP_{8,1+16,2+24,3}^{riu2}$ is chosen as the feature. A histogram $CCLTP_{8,1+16,2+24,3}^{riu2}$ is extracted from each block and all block-wise histograms are concatenated together as a target representation. The average recognition accuracy averaged by 10 trials is shown in Figure 9. The dimensionality and the average feature extraction time cost of each chip are given in Table 3. The average training and recognition time cost for each segmentation method is shown in Table 4. As seen from Figure 9, Tables 4 and 5, Seg-6 yields the best ATR recognition performance if we choose 10% and 20% chips as training data, and Seg-4 provides the best performance in other cases which is only slightly worse than ‘Seg-6’ in the 10% and 20% cases. However, the computational

Table 5 Time consumed for feature extraction, training, and recognition under different training datasets

	10%	20%	30%	40%	50%
Seg-1	0.0039	0.0076	0.0109	0.0122	0.0128
Seg-2	0.0086	0.0158	0.0208	0.0234	0.0274
Seg-3	0.0086	0.0159	0.0209	0.0235	0.0275
Seg-4	0.024	0.0422	0.0562	0.0594	0.0663
Seg-5	0.0398	0.0718	0.0947	0.1035	0.1103
Seg-6	0.0619	0.1111	0.1451	0.1633	0.1694

Table 6 Accuracy of infrared ATR for (%) six methods under different training datasets

	sCCLTP	sLTP	SPG_lasso	SparseLab_lasso	SparseLab_nlasso	SparseLab_OMP
10%	66.50	59.09	75.45	75.65	74.35	67.54
20%	79.05	72.72	84.43	83.95	82.46	76.61
30%	85.88	80.21	88.51	87.95	86.60	81.55
40%	89.80	84.66	91.10	90.24	89.12	84.87
50%	92.25	87.88	92.76	91.86	90.98	87.21
60%	93.55	89.63	93.87	93.04	92.34	88.61
70%	94.64	91.37	94.54	93.82	93.20	89.83
80%	95.29	92.29	95.23	94.43	94.08	90.95
Leave-one-out	97.63	95.28	96.87	95.84	95.53	93.15

complexity of Seg-6 is much heavier than that of ‘Seg-4’. Therefore, Seg-4 is considered the *optimal* blocking methods to apply the LTP/CCLTP for the infrared ATR task in this work.

Comparison of LTP/CCLTP and SRC methods

In this section, we compare the performance of LTP and CCLTP with SRC-based methods that achieved the state-of-the-art results in infrared ATR [9]. We have downloaded two SRC software packages which are SparseLab [48] and spectral projected gradient (SPGL1) [49]. These two packages support different sparse optimization algorithms. For SparseLab, we chose three sparse solutions: ‘lasso’, ‘nlasso’, and ‘OMP’, which are referred to as Sparselab-lasso, Sparselab-nlasso, and Sparselab-OMP in the following. For SPGL1, we chose one solution lasso, which is called SPG-lasso in this work. The reason we chose these four specific algorithms is mainly due to their relatively low computational load which is acceptable in the ATR experiments. To reduce the dimensionality of infrared chips, several dimensionality reduction techniques were studied in [9] where a simple 16×16 down-sampling was shown comparable with others, including Haar wavelet, PCA, and random projection. Thus, the dimensionality of each chip is downsampled from 40×75 to 16×16 as mentioned in [9].

In addition, we randomly selected 10% (718 chips), 20% (1436 chips), 30% (2154 chips), 40% (2,872 chips), 50% (3,590 chips), 60% (4,308 chips), 70% (4,958 chips), and

80% (5,607 chips) target chips of each target class as training data. The remaining 90%, 80%, 70%, 60%, 50%, 40%, 30%, and 20% images in the dataset are set as testing data, respectively. We augment the Seg-4 blocking method with LTP/CCLTP which leads to two new feature representations denoted by sLTP and sCCLTP. The recognition accuracy of sLTP, sCCLTP, and the four sparse-based methods that is averaged by 10 trials is given in Table 6 where we also include the leave-one-out result for each method. The average training and recognition time of six methods is shown in Table 7. As seen from Table 6, sCCLTP is showing improved performance as the training size grows. Especially for the leave-one-out experiment, sCCLTP provides the best ATR performance. The confusion matrices of six methods corresponding to the leave-one-out experiment are shown in Figure 10. It is shown that the sCCLTP result has only three non-diagonal entries greater than 1% (Figure 10a), while sLTP (Figure 10b), Sparselab-lasso (Figure 10c), Sparselab-nlasso (Figure 10d), Sparselab-OMP (Figure 10e), and SPG-lasso (Figure 10f) have 12, 9, 12, 23, and 5 nondiagonal entries greater than 1%, showing the best robustness of sCCLTP.

In order to further show the advantages of the proposed CCP methods, Figure 11 presents the curves of pose recognition accuracies vs. acceptable angle errors (up to 60°) for the leave-one-out experiment. As can be seen from Figure 11, sCCLTP again provides the best performance. Specifically, sCCLTP can get 66% pose recognition

Table 7 Averaged time consumed (s) for training and testing (sparsity-based methods do not have training stage)

	10%	20%	30%	40%	50%	60%	70%	80%
sCCLTP _{8,1+16,2+24,3} ^{riu2}	0.032	0.056	0.075	0.079	0.086	0.073	0.064	0.045
sLTP _{8,1+16,2+24,3} ^{riu2}	0.011	0.020	0.027	0.029	0.034	0.026	0.023	0.016
SPG_lasso	0.046	0.069	0.086	0.095	0.103	0.094	0.082	0.064
SparseLab_lasso	0.053	0.077	0.092	0.106	0.12	0.113	0.095	0.069
SparseLab_nlasso	0.031	0.051	0.064	0.075	0.085	0.08	0.068	0.049
SparseLab_OMP	0.022	0.032	0.037	0.04	0.044	0.041	0.032	0.022

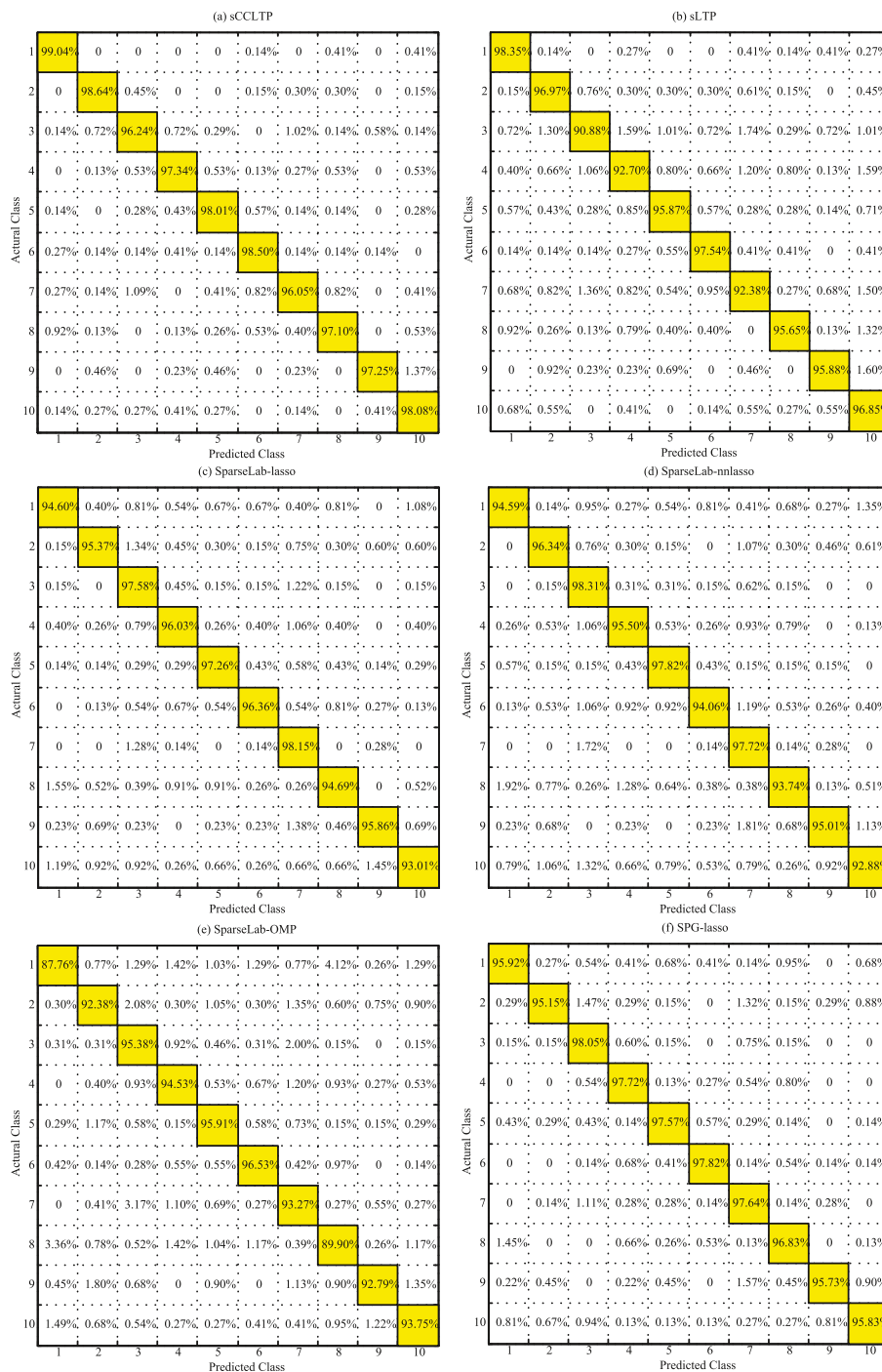


Figure 10 Confusion matrices of the six methods in the leave-one-out recognition experiment. **(a)** sCCLTP. **(b)** sLTP. **(c)** SparseLab-lasso. **(d)** SparseLab-nlasso. **(e)** SparseLab-OMP. **(f)** SPG-lasso.

accuracy with the angle error less than 5°, 85% with the angle error less than 10°, and more than 90% with the angle error less than 15°. On the other hand, sLTP has similar pose recognition performance as Sparselab-lasso and Sparselab-nlasso.

Conclusions

In this paper, a new LBP-based ATR algorithm is proposed using concave-convex partition local ternary pattern (CCLTP). Based on the analysis of the limitation of traditional LBP-based methods, CCP is presented to

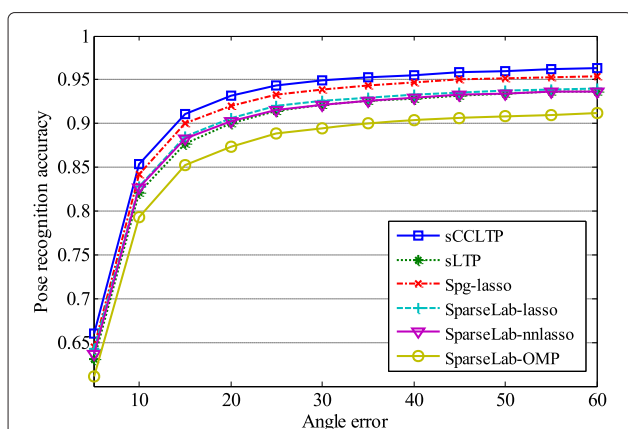


Figure 11 Curves of pose recognition accuracies versus threshold of acceptable angle error for leave-one-out experiment.

group local features into two categories in order to reflect the contrast between local and the global intensities. The improvement of the proposed CCLBP/CCLTP methods over LBP and LTP as well as two improved LBP features (SLBP and CLBP) is first demonstrated on two standard texture databases, and then CCLTP is further enhanced by augmenting a simple yet effective blocking scheme for infrared imagery, leading to sCCLTP which is evaluated against sLTP (block-based LTP) and four SRC-based methods on an infrared ATR database. Experimental results show that sCCLTP outperforms SRC-based methods and sLTP in terms of both recognition accuracy and pose estimation. It is worth mentioning that the proposed CCP technique can be applied to nearly all existing LBP features to improve feature discriminantability.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that improved this paper. This work was supported by the Backbone Teacher Grant of Henan Province (2010GGJS-059), the International Cooperation Project of Henan Province (134300510057), the Doctor and Backbone Teacher Grant of Henan Polytechnic University. The authors would like to thank MVG, Sparselab and SPGL1 for sharing the source codes of LBP and sparse-based methods.

Author details

¹School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China. ²School of Electrical and Computer Engineering, Oklahoma State University, Stillwater 74078, USA.

Received: 10 October 2013 Accepted: 27 March 2014

Published: 21 April 2014

References

1. B Li, R Chellappa, Q Zheng, S Der, N Nasrabadi, L Chan, L Wang, Experimental evaluation of forward-looking ir data set automatic target recognition approaches—a comparative study. *Comput. Vis. Image Understand.* **84**(1), 5–24 (2001)
2. LA Chan, NM Nasrabadi, V Mirelli, Multi-stage target recognition using modular vector quantizers and multilayer perceptrons, in *Proceedings of*

3. *IEEE Computer Society Conference on Computer Vision Pattern Recognition* (San Francisco, CA, 18–20 June 1996), pp. 114–119
4. LC Wang, SZ Der, NM Nasrabadi, A committee of networks classifier with multi-resolution feature extraction for automatic target recognition, in *Proceedings of IEEE International Conference on Neural Networks*, vol. 3 (Houston, TX, 9–12 June 1997), pp. 1596–1601
5. Y Lamdan, H Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, in *Proceedings of the 2nd International Conference on Computer Vision* (Tampa, FL, 5–8 December 1988), pp. 238–249
6. CF Olson, DP Huttenlocher, Automatic target recognition by matching oriented edge pixels. *IEEE Trans. Image Process.* **6**(1), 103–113 (1997)
7. U Grenander, MI Miller, A Srivastava, Hilbert-Schmidt lower bounds for estimators on matrix lie groups for ATR. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 790–802 (1998)
8. V Venkataraman, G Fan, L Yu, X Zhang, W Liu, JP Havlicek, Automated target tracking and recognition using coupled view and identity manifolds for shape representation. *EURASIP J. Adv. Signal Process.* **124**, 1–17 (2011)
9. J Gong, G Fan, L Yu, JP Havlicek, D Chen, N Fan, Joint view-identity manifold for infrared target tracking and recognition. *Comput. Vis. Image Understand.* **118**(1), 211–224 (2014)
10. VM Patel, NM Nasrabadi, R Chellappa, Sparsity-motivated automatic target recognition. *Appl. Opt.* **50**(10), 1425–1433 (2011)
11. J Wright, AY Yang, A Ganesh, SS Sastry, Y Ma, Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 210–227 (2009)
12. J Liebelt, C Schmid, K Schertler, Viewpoint-independent object class detection using 3D feature maps, in *Proceedings of IEEE Conference on CVPR* (Anchorage, AK, 23–28 June 2008), pp. 1–8
13. SM Khan, H Cheng, D Matthies, H Sawhney, 3D model based vehicle classification in aerial imagery, in *IEEE Conference on Computer Vision and Pattern Recognition* (San Francisco, CA, 13–18 June 2010), pp. 1681–1687
14. A Toshev, A Makadia, K Daniilidis, Shape-based object recognition in videos using 3D synthetic object models, in *IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL, 20–25 June 2009), pp. 288–295
15. B Bhanu, Automatic target recognition: state of the art survey. *IEEE Trans. Aero. Electron. Syst.* **AES-22**(4), 364–379 (1986)
16. K Messer, D de Ridder, J Kittler, Adaptive texture representation methods for automatic target recognition, in *Proceedings of British Machine Vision Conference BMVC99* (Nottingham, 13–16 September 1999), pp. 1–10
17. D Casasent, Y Smokelin, Wavelet and Gabor transforms for target detection. *Opt. Eng.* **31**(9), 1893–1898 (1992)
18. C Jeong, M Cha, H-M Kim, Texture feature coding method for SAR automatic target recognition with adaptive boosting, in *Proceedings of the 2nd Asian-Pacific Conference on Synthetic Aperture Radar* (Xian, 26–30 October 2009), pp. 473–476
19. N Rahmani, A Behrad, Automatic marine targets detection using features based on local Gabor binary pattern histogram sequence, in *Proceedings of the 1st International Conference on Computer and Knowledge Engineering* (Mashhad, 13–14 October 2011), pp. 195–201
20. Y Qin, Z Cao, Z Fang, A study on the difficulty prediction for infrared target recognition, in *Proceedings of SPIE*, vol. 8918 (Wuhan, 26 October 2013). doi:10.1117/12.2031100
21. F Wang, W Sheng, X Ma, H Wang, Target automatic recognition based on ISAR image with wavelet transform and MBLBP, in *2010 International Symposium on Signals Systems and Electronics*, vol. 2 (Nanjing, 17–20 September 2010), pp. 1–4
22. T Ojala, M Pietikäinen, D Harwood, A comparative study of texture measures with classification based on featured distributions. *Pattern Recogn.* **29**(1), 51–59 (1996)
23. L Wang, D-C He, Texture classification using texture spectrum. *Pattern Recogn.* **23**(8), 905–910 (1990)
24. M Pietikäinen, T Ojala, Z Xu, Rotation-invariant texture classification using feature distributions. *Pattern Recogn.* **33**(1), 43–52 (2000)
25. M Topi, P Matti, O Timo, Texture classification by multi-predicate local binary pattern operators, in *Proceedings of the International Conference on Pattern Recognition*, vol. 3 (Barcelona, 3–7 September 2000), pp. 939–942
26. T Ojala, K Valkealahti, E Oja, M Pietikäinen, Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recogn.* **34**(3), 727–739 (2001)

26. G Zhao, M Pietikäinen, Improving rotation invariance of the volume local binary pattern operator, in *Proceedings of IAPR Conference on Machine Vision Applications* (Tokyo, 16–18 May 2007), pp. 327–330
27. H Zhou, R Wang, C Wang, A novel extended local-binary-pattern operator for texture analysis. *Inform. Sci.* **178**(22), 4314–4325 (2008)
28. Z Guo, L Zhang, D Zhang, Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recogn.* **43**(3), 706–719 (2010)
29. Y Guo, G Zhao, M Pietikäinen, Discriminative features for texture description. *Pattern Recogn.* **45**(10), 3834–3843 (2012)
30. M Subrahmanyam, R Maheshwari, R Balasubramanian, Local maximum edge binary patterns: a new descriptor for image retrieval and object tracking. *Signal Process.* **92**(6), 1467–1479 (2012)
31. C Shan, S Gong, PW McOwan, Facial expression recognition based on local binary patterns: a comprehensive study. *Image Vis. Comput.* **27**(6), 803–816 (2009)
32. X Tan, B Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans. Image Process.* **19**(6), 1635–1650 (2010)
33. B Zhang, Y Gao, S Zhao, J Liu, Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. *IEEE Trans. Image Process.* **19**(2), 533–544 (2010)
34. D Huang, M Ardabilian, Y Wang, L Chen, A novel geometric facial representation based on multi-scale extended local binary patterns, in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition and Workshops* (Santa Barbara, 21–25 March 2011), pp. 1–7
35. D Huang, C Shan, M Ardabilian, Y Wang, L Chen, Local binary patterns and its application to facial image analysis: a survey. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **41**(6), 765–781 (2011)
36. L Nanni, A Lumini, S Brahmam, Survey on LBP based texture descriptors for image classification. *Exp. Syst. Appl.* **39**(3), 3634–3641 (2012)
37. M Heikkilä, M Pietikäinen, A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 657–662 (2006)
38. S Zhang, H Yao, S Liu, Dynamic background modeling and subtraction using spatio-temporal local binary patterns, in *Proceedings of the IEEE International Conference on Image Processing* (San Diego, CA, 12–15 October 2008), pp. 1556–1559
39. S Liao, G Zhao, V Kellokumpu, M Pietikäinen, SZ Li, Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (San Francisco, CA, 13–18 June 2010), pp. 1301–1306
40. L Nanni, A Lumini, S Brahmam, Local binary patterns variants as texture descriptors for medical image analysis. *Artif. Intell. Med.* **49**(2), 117–125 (2010)
41. L Srensen, SB Shaker, M De Bruijne, Quantitative analysis of pulmonary emphysema using local binary patterns. *IEEE Trans. Med. Imaging.* **29**(2), 559–569 (2010)
42. M Häfner, M Liedlgruber, A Uhl, A Vécsei, F Wrba, Color treatment in endoscopic image classification using multi-scale local color vector patterns. *Med. Image Anal.* **16**(1), 75–86 (2012)
43. T Ojala, M Pietikäinen, T Maenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
44. T Ahonen, M Pietikäinen, Soft histograms for local binary patterns, in *Proceedings of the Finnish Signal Processing Symposium* (Oulu, 30 October 2007), pp. 1–4
45. Z Guo, L Zhang, D Zhang, A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.* **19**(6), 1657–1663 (2010)
46. S Liao, X Zhu, Z Lei, L Zhang, S Li, Learning multi-scale block local binary patterns for face recognition. *Adv. Biometrics*, 828–837 (2007)
47. L Wolf, T Hassner, Y Taigman, Descriptor based methods in the wild, in *Faces in Real-Life Images Workshop at the European Conference on Computer Vision* (Marseille, 17–18 October 2008), pp. 1–14
48. SparseLab. <http://sparselab.stanford.edu>, website. Accessed 24 September 2013
49. SPGL1. <http://www.cs.ubc.ca/~mpf/spgl1>, website. Accessed 24 September 2013
50. S-J Ko, S-H Lee, K-H Lee, Digital image stabilizing algorithms based on bit-plane matching. *IEEE Trans. Consum. Electron.* **44**(3), 617–622 (1998)
51. T Ojala, T Maenpää, M Pietikäinen, J Viertola, J Kyllönen, S Huovinen, Outex-new framework for empirical evaluation of texture analysis algorithms, in *Proceeding of the IEEE International Conference on Pattern Recognition*, vol. 1 (Quebec, 11–15 August 2002), pp. 701–706
52. KJ Dana, B Van Ginneken, SK Nayar, JJ Koenderink, Reflectance and texture of real-world surfaces. *ACM Trans. Graph. (TOG)*. **18**(1), 1–34 (1999)
53. B Yang, S Chen, A comparative study on local binary pattern (LBP) based face recognition LBP histogram versus LBP image. *Neurocomputing*. **120**, 365–379 (2013)

doi:10.1186/1687-5281-2014-23

Cite this article as: Sun et al.: Concave-convex local binary features for automatic target recognition in infrared imagery. *EURASIP Journal on Image and Video Processing* 2014 **2014**:23.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
