

## Research Article

# The Study of Scene Classification in the Multisensor Remote Sensing Image Fusion

Ji Li<sup>1,2</sup> and Zhen Liu<sup>1,2</sup>

<sup>1</sup> College of Computer Science, Chongqing University, 400030 Shapingba, Chongqing, China

<sup>2</sup> Key Laboratory for Dependable Service Computing in Cyber Physics Society of Ministry of Education, China

Correspondence should be addressed to Ji Li; [leedge@cqu.edu.cn](mailto:leedge@cqu.edu.cn)

Received 24 March 2013; Accepted 28 April 2013

Academic Editor: Hua Li

Copyright © 2013 J. Li and Z. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a scene classification method for speeding up the multisensor remote sensing image fusion by using the singular value decomposition of quaternion matrix and the kernel principal component analysis (KPCA) to extract features. At first, images are segmented to patches by a regular grid, and for each patch, we extract color features by using quaternion singular value decomposition (QSVD) method, and the grey features are extracted by Gabor filter and then by using orientation histogram to describe the grey information. After that, we combine the color features and the orientation histogram together with the same weight to obtain the descriptor for each patch. All the patch descriptors are clustered to get visual words for each category. Then we apply KPCA to the visual words to get the subspaces of the category. The descriptors of a test image then are projected to the subspaces of all categories to get the projection length to all categories for the test image. Finally, support vector machine (SVM) with linear kernel function is used to get the scene classification performance. We experiment with three classification situations on OT8 dataset and compare our method with the typical scene classification method, probabilistic latent semantic analysis (pLSA), and the results confirm the feasibility of our method.

## 1. Introduction

Image fusion refers to fusing two or more images which are in the same scene into a composite image [1]. The composite image contains the significant information of these various source images. Then we can obtain a more comprehensive and clear description of the scene. Image fusion information is usually obtained from the multisensor remote sensing images. Information fusion of multisensor remote sensing images can collect a wealth of information, and remote sensing is able to improve the means of the global scope dynamic observation data. The categories of ground targets and the scene environment are different, and the remote sensing images obtained from each sensor are not necessarily the same object or the same scene. We need to classify these remote sensing images into various categories, such as city, forest, open country, mountain, and coast. By that way, it will be conducive to speed up the image fusion in the same scene, which is more conducive to improve the important influence of the multisensor remote sensing images in many

monitoring analytical applications, such as city planning, forest monitoring, land, and use monitoring [2], and flood monitoring.

Scene classification is an important part of the multi-sensor remote sensing image interpretation. Conventional scene classification methods mainly use the images' low level features (global color or texture histograms, the power spectrum, etc.) to classify only a small number of scene categories [3, 4]. Szummer and Picard proposed indoor and outdoor classification method. They used color, texture, and frequency features of local regions in their method and got the 90.3% correct classification [3]. Paek and Chang used the HSV color histogram and the edge direction histogram to represent each block of the image in the color and texture features and got the similar result with Szummer and Picard [4]. However, the classification accuracy of the method based on low-level features is not satisfactory.

In recent years, many authors proposed bag-of-visual words methods based on local patches without topological information for scene classification, which have been applied

to cases where there are a larger number of scene categories [5–9]. Fei-Fei and Perona proposed a scene classification method based on bag of keypoints to classify thirteen scene categories [5]. Bosch et al. also presented a method based on pLSA to classify a larger number of scene categories and got good classification performance [6]. These methods used local patches without topological information. However, they both [5, 6] reported that the classification result is better when local parts are selected from evenly sampled grid which were named dense patches in [6]. Therefore, in this paper we use dense patches from which we extract features for classification. Dense patches are constructed by segmenting images by regular grid. The visual words vocabulary can be obtained by clustering the features of patches with the  $k$ -means algorithm. Thus, an image can be viewed as a collection of “visual words.” The feature vector is developed from patches. In [6], they used color patches and extracted SIFT features in each color channel, respectively. But the dimension of the feature vector they got is very high, and the method spent a lot of time in feature extraction, clustering, and quantification. In order to reduce the time needed for classification, we should reduce the dimension of the feature vector.

To answer this demand, in this paper we use the quaternion singular value decomposition (QSVD) method to extract the color features of the image. Because the RGB channels are treated as an integrated whole in the quaternion model of a color image, the three colors do not have to be operated in each color channel independently. In this way, we take into account the direct interchannel effects in the three color channels and reduce the dimension of the feature vector. The singular value vector of a color image is robust to shift variations, scale variations, and rotation variations [10]. Therefore, the singular value decomposition of the quaternion model of a color image is effective for scene classification. We obtain a quaternion matrix from each patch of an input image.

Simultaneously, we also obtain the orientation histogram of multiscale Gabor feature from each patch. In recent years, it is reported that the orientation histogram of Gabor features with rough topological information which is developed from dense patches is effective for scene classification [7, 8]. We develop the orientation histogram from multiscale Gabor features because Gabor features which extract frequency and orientation information are better than the simple gradient [8]. Then we combine the color features obtained from the quaternion model and the orientation histogram to form the final feature vector computed from each patch. Although the most famous approach for scene classification is bag-of-visual words method, superior accuracy of KPCA of visual words to bag-of-visual words was reported in recent years [8, 11]. However, the computational cost of KPCA becomes high when the number of training samples increases. To reduce the computational cost, we must decrease the number of training samples. To solve the problem, we use the bag-of-visual words method to extract the representative feature vectors of each category. Then we applied KPCA to the set of visual words of each category. The uncorrelated mean projection length to the subspace of each category is used as input features

for classification. Finally, we use SVM to classify the input features. The mean projection length in subspace is used as features for support vector machine. The proposed method is evaluated using OT8 dataset [12]. We compare our method with the recent popular scene classification method pLSA in [6].

The rest of paper is organized as follows. In Section 2, the methods of context information extraction is presented. In Section 3 briefly introduces similarities with subspaces and explains how the information specialized for each category can be extracted. The SVM and devices on kernel are introduced in Section 4. Section 5 shows the experimental results and evaluates the results. Conclusion and future works are described in Section 6.

## 2. Context Information in Patch Space

For scene images, dense patches are obtained from a regular  $M \times M$  grid, and here we use  $16 \times 16$  grid. The visual words are developed by clustering descriptors of patches that are computed from the training images. Accordingly, all the patches are represented by the clustering centers called the visual words obtained by  $k$ -means algorithm. In this paper, for each patch we extract color features, and grey features, respectively, and use weights to integrate them as the entire features for classification.

*2.1. Singular Value Decomposition of Quaternion Matrices.* Color in images processing has been widely used recently. For color images, the color of each pixel is usually expressed in RGB color, and each element of the image matrix saves three values of RGB color. The traditional methods of color images processing usually separate the color images into RGB three channels images first and then use a variety of image processing methods to process images. But as a result of inherent relationship among RGB components, image separation leads to the distortion of color image. Recently, the effectiveness of QSVD in color images processing was reported [13–16]. In this paper, we develop the color features by using QSVD method. This is because the algorithm uses a quaternion to represent the RGB values of a pixel in color images. In this way we make the RGB color channels as an integrated whole, and take into account the direct interchannel effects in the three color channels.

First, we explain the quaternion briefly. The quaternion [13] is defined as

$$q = q_r + q_i i + q_j j + q_k k, \quad (1)$$

where  $q_r, q_i, q_j, q_k$  are all real and  $i, j, k$  are orthogonal imaginary operators, which satisfy the following identities:

$$\begin{aligned} i^2 = j^2 = k^2 = i \cdot j \cdot k = -1, \\ i \cdot j = k, \quad j \cdot k = i, \quad k \cdot i = j, \\ j \cdot i = -k, \quad k \cdot j = -i, \quad i \cdot k = -j. \end{aligned} \quad (2)$$

$q_r$  is the real part, and  $q_i i, q_j j, q_k k$  are the imaginary parts. A quaternion with no real part ( $q_r = 0$ ) is said to be pure quaternion.

If we set  $q_r$  in (1) as 0 and use  $q_i = f_r(x, y)$ ,  $q_j = f_g(x, y)$ , and  $q_k = f_b(x, y)$  to represent the RGB values of a pixel in color image, then an  $M \times N$  color image can be represented as a pure quaternion image by the following function  $f_{(q)}(x, y)$ :

$$f_{(q)}(x, y) = f_r(x, y)i + f_g(x, y)j + f_b(x, y)k, \quad (3)$$

where  $f_r(x, y)$ ,  $f_g(x, y)$ , and  $f_b(x, y)$ , respectively, are red, green and blue components of the pixel at position  $(x, y)$  in the image. Thus, the color image is a matrix of size  $M \times N$  over the quaternion field (restricted to pure quaternion). So a color image can be represented as a quaternion matrix.

In this paper, quaternion is represented as a matrix  $Q$ ,  $Q^{m \times n} = \{(a_{ij})_{m \times n} \mid a_{ij} \in Q, i = 1, 2, \dots, m, j = 1, 2, \dots, n\}$ . Consider an  $m \times n$  quaternion matrix  $X_{(q)}$ ,  $X_{(q)} \in Q^{m \times n}$  and  $X_{(q)} = X_r + X_i i + X_j j + X_k k$ , where  $X_l \in R^{m \times n}$  ( $l = r, i, j, k$ ), then the equivalent real matrix of  $X_{(q)}$  is

$$X_{e(r)} = \begin{pmatrix} X_r & X_i & X_j & X_k \\ -X_i & X_r & -X_k & X_j \\ -X_j & X_k & X_r & -X_i \\ -X_k & -X_j & X_i & X_r \end{pmatrix} \in R^{4m \times 4n}. \quad (4)$$

The definition of the equivalent real matrix of a quaternion matrix  $X_{(q)}$  denotes the relationship between the real matrix and the quaternion matrix. In addition, we can use the eigenvalues and eigenvectors of the equivalent real matrix to compute the eigenvalues and eigenvectors of a quaternion matrix [14]. The relationship between the singular value decomposition of a quaternion matrix and the singular value decomposition of the equivalent real matrix can be found in [14].

Next, we consider the QSVD of a color image. The proof of existence and meaning of the SVD of a quaternion matrix can be found in [14]. Any matrix  $X_{(q)} \in Q^{m \times n}$  admits a singular value decomposition given as

$$X_{(q)} = U_{(q)} \begin{pmatrix} \Lambda_r & 0 \\ 0 & 0 \end{pmatrix} V_{(q)}^H, \quad (5)$$

where  $H$  is the conjugate-transposition operator,  $U_{(q)} \in Q^{m \times m}$ , and  $V_{(q)} \in Q^{n \times n}$  are two unitary quaternion matrices. Besides,  $U_{(q)} U_{(q)}^H = I^{m \times m}$  and  $V_{(q)} V_{(q)}^H = I^{n \times n}$ , where  $I$  is the identify matrix. These matrices contain the left and right quaternion singular vectors of  $X_{(q)}$ .  $\Lambda_r$  is a real diagonal matrix, where  $r$  is the rank of  $X_{(q)}$ . The values on the diagonal of  $\Lambda_r$ ,  $\lambda_n$  (with  $1 \leq n \leq r$ ), are the singular values of the quaternion matrix, arranged in decreasing magnitude order along the diagonal. The method for obtaining the decomposition of a quaternion matrix can be found in [14], which is based on the equivalent real matrix.

Now, we explain how to extract color features from the QSVD results of a patch. In this paper, the color features are developed from the two quaternion unitary matrices of QSVD. The most important information is contained in the two quaternion unitary matrices of QSVD [14]. Like the

singular value decomposition (SVD) of a grey image, the SVD of a color image also can be represented as follows:

$$X_{(q)} = U_{(q)} \Lambda V_{(q)}^H = \sum_{i=1}^r \lambda_i u_{i(q)} v_{i(q)}^H. \quad (6)$$

Here,  $U_{(q)} = [u_{1(q)}, u_{2(q)}, \dots, u_{m(q)}]$ ,  $V_{(q)} = [v_{1(q)}, v_{2(q)}, \dots, v_{n(q)}]$ ,  $u_{i(q)}$  and  $v_{i(q)}$  are the  $i$ th column vectors of  $U_{(q)}$  and  $V_{(q)}$ , respectively.  $\lambda_i$  is the diagonal element of the real matrix  $\Lambda$ , and  $r$  is the rank of  $X_{(q)}$ . Each result of  $u_{i(q)} v_{i(q)}^H$  is referred as the  $i$ th weighted eigenimage [14]. From the original image and its eigenimages, we can find that most image information, that is, energy, color, and structure, is concentrated on the foreside of the eigenimages. Figure 1 illustrates an image and a part of its eigenimages. From the figure we can discover that the first three eigenimages possess most energy of the image [14]. Then we use the first three eigenimages to reconstruct the original image, and the reconstructed image is shown in Figure 1.

From Figure 1, we can observe that the first several eigenimages contain the average information about color and structure of the original image (low frequency information). So in this paper, we use the first eigenimage as the color feature for classification. We can get the average information of the original image, not the detail information. In this way, we can avoid the problem of over fitting and improve the generalization ability of the proposed method.

In the experiments, we use evenly sampled  $16 \times 16$  grid, and for each grid (each patch) we compute its equivalent real matrix, then obtain the result of QSVD of the real matrix, and then compute the first eigenimage as the color feature.

**2.2. Orientation Histogram Extraction Based on Gabor Features.** Recently, orientation histogram has been widely used in pattern recognition. Here, we develop the orientation histogram from Gabor features [8]. A distinct advantage of Gabor features is their optimality in time and frequency, or space and spatial frequency in two dimensions, providing the smallest possible pieces of information about time-frequency events.

In this paper, we define Gabor filters as

$$\Psi_k(x) = \frac{k_v^2}{\sigma^2} \exp\left(\frac{-k_v^2 x^T x}{2\sigma^2}\right) \times \left( \exp(ik^T x) - \exp\left(-\frac{\sigma^2}{2}\right) \right), \quad (7)$$

where  $x = (y, x)$ ,  $k = k_v \exp(i\phi) = (k_v \cos(\phi), k_v \sin(\phi))^T$ ,  $k_v = k_{\max}/f^v$ ,  $\phi = \mu \cdot \pi/8$ ,  $f = \sqrt{2}$ , and  $\sigma = \pi$  [7].  $f$  is the frequency of a sinusoidal plane wave,  $\phi$  is the anticlockwise rotation of the Gaussian envelope and the sinusoid, and  $\mu$  and  $v$  represent the orientation and the scale of the Gabor filter respectively. In the experiments of this essay, we use Gabor filters of eight different orientations ( $\mu = \{0, 1, \dots, 7\}$ ) with three frequency levels ( $v = \{0, 1, 2\}$ ). The size of Gabor filters of three different frequency levels is set to  $9 \times 9$ ,  $13 \times 13$ , and  $17 \times 17$  pixels, respectively [7].

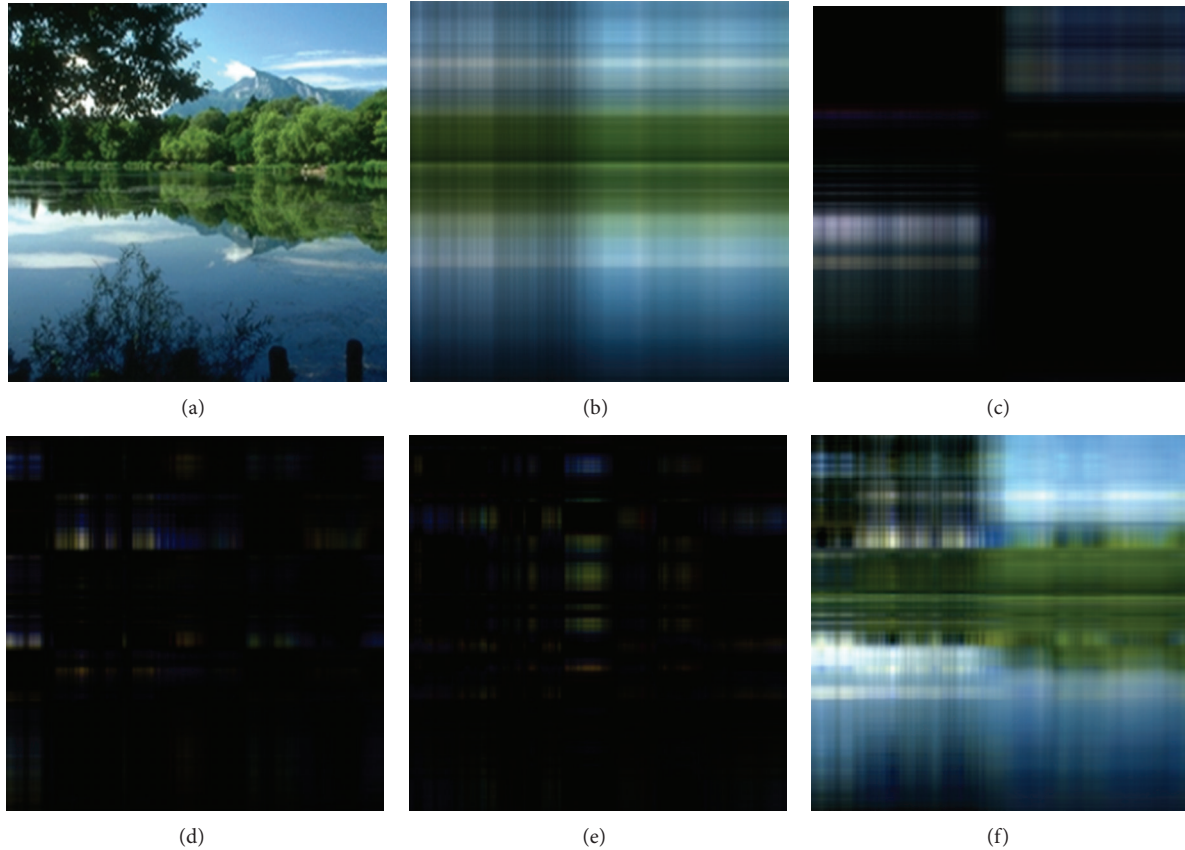


FIGURE 1: (a) Original image; (b) the first eigenimage; (c) the second eigenimage; (d) the third eigenimage; (e) the fourth eigenimage; (f) the reconstructed image of the first three eigenimages.

Figure 2 shows the output (norm of real and imaginary parts) of the Gabor filters with various  $\mu, \nu$ . Figure 2(a) shows the original image. Figure 2(b) is the Gabor output when Gabor filters with  $\mu = 4, \nu = 0$  are applied to Figure 2(a). Figure 2(c) is the output of the Gabor filters with  $\mu = 4, \nu = 1$ . Figure 2(d) is the output of the Gabor filters with  $\mu = 4, \nu = 2$ . From the figure, we can see that Gabor filter with small  $\nu$  being sensitive to fine edges, so the output image (b) is clearer than (c) and (d).

Now, we introduce how to develop the orientation histogram from the output of Gabor filters. In this paper, the orientation histogram is developed from a regular  $16 \times 16$  grid. Figure 3 lists an orientation histogram of the original image. First, for each scale, we extracted Gabor features (real and imaginary parts) of eight orientations from each grid of the input image. We compute the norm of real and imaginary parts for each pixel. Then, the orientation histogram with eight bins at each grid is developed by voting the output value of the maximum orientation at each pixel to the orientation bin [8]. In order to reduce the impact of light changes, we normalized the histogram of each patch. This process is repeated at each scale parameter independently. So we finally get an orientation histogram with 24 bins ( $24 = 3 \text{ scales} \times 8 \text{ orientation bins}$ ) from each grid.

We fuse the orientation histogram and the first eigenimage (the color feature vector) by the same weight for each patch to form the descriptor of each patch.

### 3. Subspaces Similarities

The proposed method uses KPCA to model the similarities with subspaces and extract features specialized for each category. KPCA can achieve the transformation from input space to feature space through the nonlinear mapping  $\Phi$  and then perform linear PCA on the mapped data, and so it has a strong nonlinear processing ability. However, the computational cost of KPCA becomes high when the number of training samples increases. To reduce the computational cost, we must decrease the number of training samples. To solve this problem, we use clustering algorithm. The cluster centers are selected from the descriptors of each category by using k-means method, and the cluster centers are used to construct the category-specific subspace by KPCA. The cluster centers are called visual words vividly, so KPCA was applied to the visual words of each category. Since the visual words of each category include various kinds of local parts, the distribution becomes nonlinear. Thus, the nonlinearity of KPCA is appropriate for modeling them.



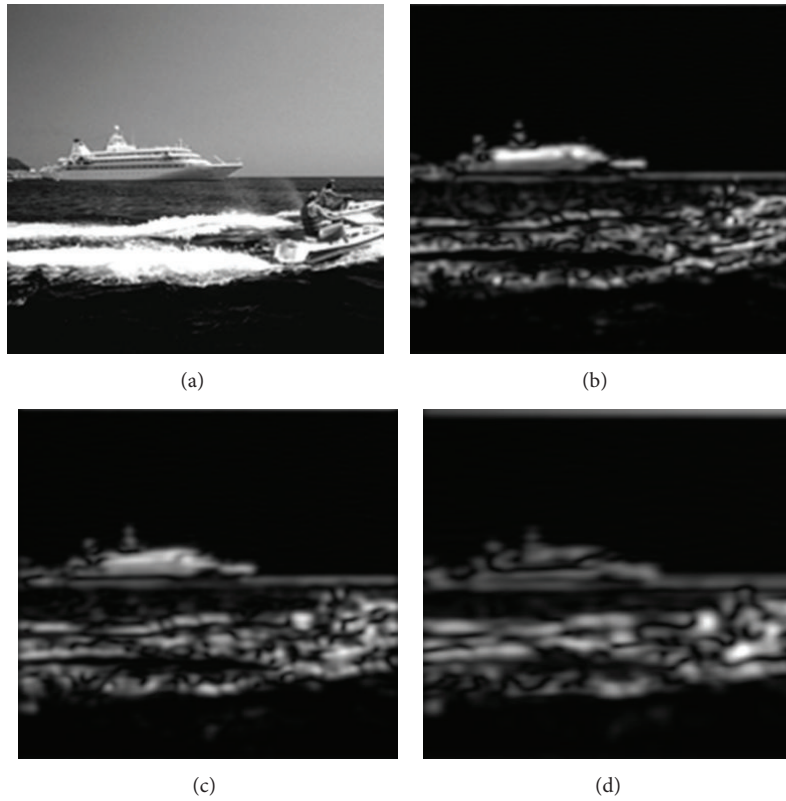


FIGURE 2: (a) Original image; (b) the output of Gabor filter with  $\mu = 4, \nu = 0$ ; (c) the output of Gabor filter with  $\mu = 4, \nu = 1$ ; (d) the output of Gabor filter with  $\mu = 4, \nu = 2$ .

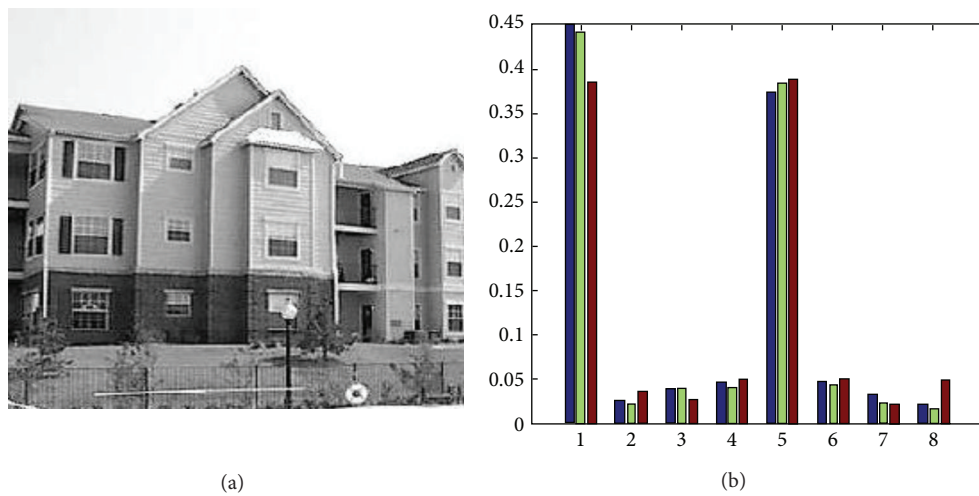


FIGURE 3: (a) Original image; (b) the orientation histogram with eight different orientations and three scales of the original image (the different colors represent different scales).

3.1. *Introduction of KPCA.* In the following, we introduce KPCA briefly [11]. Suppose we have a set of  $M$  samples in an  $N$ -dimensional space  $x_1, x_2, \dots, x_M \in R^N$ , and  $x$  is mapped into high dimensional space by nonlinear mapping  $\Phi(x)$ . By applying standard linear PCA in high dimensional space, nonlinear principal components are obtained. Nonlinear

mapping function  $\Phi$  is introduced to change the sample points  $x_1, x_2, \dots, x_M$  in the input space into sample points  $\Phi(x_1), \Phi(x_2), \dots, \Phi(x_M)$  in the feature space, and assume that

$$\sum_{i=1}^M \Phi(x_i) = 0. \tag{8}$$

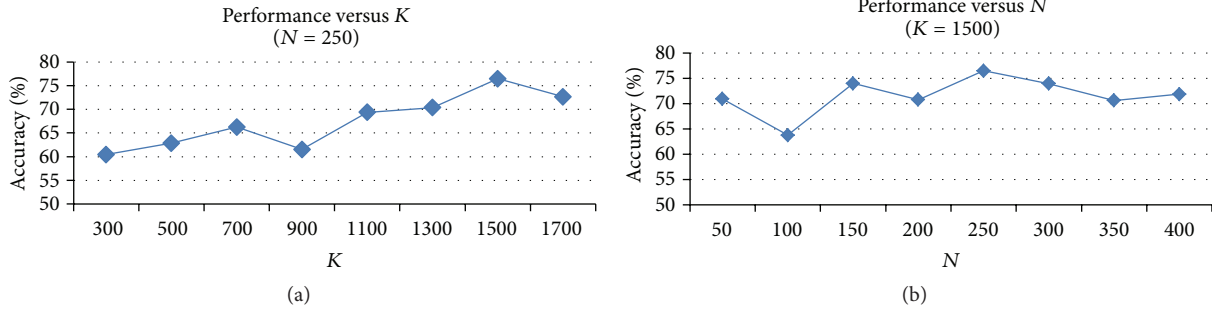


FIGURE 4: (a) Varying number of visual words  $K$  and fixing number of subspace axes  $N = 250$ ; (b) varying number of subspace axes  $N$  and fixing number of visual words  $K = 1500$ .

Thus, the covariance matrix in high dimensional space is computed by

$$C = \frac{1}{M} \sum_{i=1}^M \Phi(x_i) \Phi(x_i)^T. \quad (9)$$

Therefore, the PCA in feature space is to solve the eigenvalues  $\lambda$  and the eigenvectors  $V$  for KPCA in equation  $\lambda V = CV$ . Where  $V$  is defined as:

$$V = \sum_{i=1}^M \alpha_i \Phi(x_i), \quad (10)$$

where  $\alpha_i$  is the coefficient. Define the  $M \times M$  matrix  $K$  ( $K$  is the kernel function) as

$$K_{ij} = \Phi(x_i)^T \Phi(x_j). \quad (11)$$

Then we obtain the following eigenvalue problem:

$$M\lambda\alpha = K\alpha. \quad (12)$$

By solving the formula (10), we can obtain the required eigenvalues, eigenvectors, and  $\alpha$ .

In this paper, the  $q$ th descriptor  $y_q$  is projected to the subspace of each category, and the square of projection value to each axis is computed. The projection length represents the similarity with an axis in subspace of a certain category. The projection length  $z_{kq}^p$  of the descriptor  $y_q$  to  $p$ th principal component axis of the subspace of category  $k$  is computed as

$$z_{kq}^p = \left( \sum_{i=1}^M \alpha_{ki}^p K(x_{ki}, y_q) \right)^2, \quad (13)$$

where  $\alpha$  is the solution of KPCA,  $M$  is the number of visual words,  $x_{ki}$  is the  $i$ th visual word of the category  $k$ , and  $K$  is the kernel function. This equation shows that the projection length is computed by the weighted sum of similarities with the ensemble of visual words because kernel function measures the similarity with visual words [7, 11]. Here we use the normalized polynomial kernel as the kernel function in KPCA [11].

**3.2. Procedure of KPCA of Visual Words.** First, we get 256 descriptors from each training image of certain category. Then, by using  $k$ -means clustering method, we obtain the  $K$  visual words. Next, KPCA is applied to  $K$  visual words. Note that each category has one subspace. Finally, we compute the mean projection length for each subspace of the certain category. In this paper, the mean projection length is used to integrate all descriptors to be robust to the order of the descriptors [11]. In other words,  $p$ th projection length in the subspace of category  $k$  is computed as

$$z_k^p = \frac{1}{256} \sum_{i=1}^{256} z_{kq}^p. \quad (14)$$

The mean projection length to the subspace is used as the features for classification.

Let us consider the number of visual words ( $K$  in the  $k$ -means vector quantization) and the number of subspace axes ( $N$  the number of principal components in KPCA). We investigate the variation of classification performance with change in the number of visual words and subspace axes for the case of the 4 natural categories of OT8 dataset. We carry out experiments with the original images in OT8 dataset. In our experiments, datasets are split randomly into two separate sets of images, half for training and half for testing. We take 100 random images from the training set to form a validation set for finding the optimal parameters, and the rest of the training images are used to compute the visual words. The visual words are learnt from about 30 random training images of each category. This protocol is the same as the conventional method [6].

We find the optimal parameters ( $K$  and  $N$ ) over the validation set by using the cross-validation algorithm. Figure 4(a) shows the relationship between the classification performance and the number of visual words  $K$ . Figure 4(b) shows the relationship between the classification performance and the number of subspace axes  $N$ . To get the optimal parameters, we repeat the experiment ten times with varying random selection of the training and test sets and building the visual words vocabulary afresh each time. The mean classification rate of ten runs is used as a final result.

From Figure 4, we can see that the classification accuracy reaches the maximum value when the number of visual words  $K$  is 1500 and the number of subspace axes  $N$  is 250. Then the

optimal parameters are fixed and subsequent experiments use the optimal parameters.

#### 4. Classifications by SVM

We adopt SVM to the mean projection length to the subspaces of all categories. First, we explain the support vector machine (SVM) [8, 17] briefly. The SVM algorithm seeks the optimal hyperplane which maximizes the distance between hyperplane and nearest sample from it. SVM is basically a hyperplane classifier  $f(x) = \langle w, b \rangle + b$  aimed at solving the two-class problem. For nonlinearly separable data, a nonlinear mapping function  $\Phi: R^N \rightarrow F, x \rightarrow \Phi(x)$  is used to map them into a higher dimension feature space where the hyperplane classifier can be applied. When the training set (sample and its label) is denoted as  $S = ((x_i, y_i), \dots, (x_L, y_L))$ , the nonlinear SVM using kernel is defined to be

$$\begin{aligned} f(x) &= \sum_{i \in SV} \alpha_i y_i \Phi(x_i)^T \Phi(x) + b \\ &= \sum_{i \in SV} \alpha_i y_i K(x_i, x) + b, \end{aligned} \quad (15)$$

where SV shows the support vectors and  $y_i$  is the label of training sample  $x_i$ , and  $K(x_i, x)$  is a kernel function, for example, a polynomial kernel, a linear kernel, a radial basis function (RBF) kernel, and so forth. However, since category-specific nonlinear processing is done by KPCA, SVM with linear kernel is enough to categorize many scene categories. Moreover, the computational cost of linear SVM is low because the decision function is computed by inner product.

#### 5. Proposed Method for Scenes Classification

Most of the remote sensing images have a clear scene, such as desert, farmland, forest, and coast. So the multisensor remote sensing images are usually treated as scene images. This paper is trying to recognize the scenes of 8 different categories: coast, forest, highway, inside city, mountain, open country, street, and tall building, and the classifier is trained to identify an input image to the right scene category. The detailed work flow of the proposed method is shown in Figure 5.

The quaternion method and the orientation histogram of Gabor features are used to extract color features and grey features, respectively. Then they are integrated to form the descriptor of a patch. Next, we use the  $k$ -means clustering algorithm to obtain the visual words of a certain category. After that, KPCA is applied to the visual words to get the subspace of the category. The descriptors of a test image then project to the subspaces of all categories to get the projection length to all categories of the test image. Finally, mean projection length is computed by (14), and is inputted into SVM classifier.

#### 6. Evaluation Classification Results

In this paper, we consider three classification situations like the experiments in [6]: classification into eight categories,

and also classification within the two subsets of natural (four categories) and man-made (four categories) images. Then we compare our method with the typical scene classification method pLSA in [6].

We repeat the experiment six times with varying random selection of the training and test sets and building the visual words vocabulary afresh each time. The mean classification rate of six runs is used as a final result. All parameters are fixed with the number of visual words  $K = 1500$  and the number of subspace axes  $N = 250$ . All the experiments are carried by using Matlab on the same desktop computer with AMD Athlon(tm) II X2 3.0 G CPU.

Figure 6 illustrates the results of comparing our classification method with the typical scene classification methods in [6]. We compare our classification results for four natural categories with the pLSA methods [4]. Figure 6(a) shows the result of this comparison. We only compare with the performance when classifying the four natural categories using unnormalized images and with nonoverlapping  $11 \times 11$  patch in [6], since this situation in [6] is the closest to our experimental conditions in this paper. In our experiments, we use unnormalized images with nonoverlapping  $16 \times 16$  patch. From Figure 6(a), we can get that the classification rate of our method is about 1% better than the performance described in [6]. From [6], we can find that the size of patch is larger and that the classification accuracy is lower. In this paper, we use  $16 \times 16$  patch larger than the  $11 \times 11$  patch in [6], and so if we use  $16 \times 16$  patch to extract features by using the method in [6], we can get more poor results. In [6], they use color patches, and they consider the three color components HSV and obtain a  $112 \times 3 = 363$  dimensional vector. However, in our experiments, we have a  $3 \times 8 = 24$  dimensional grey feature vector and a  $16 \times 16 = 256$  dimensional color feature vector. So the total dimension of the descriptor for each patch is  $24 + 256 = 280$ . As we all know, the computational cost will increase when the feature dimensions increase. Our method is superior to pLSA in computational cost and computation speed. To sum up, our method outperforms the method proposed in [6] in this situation.

In the following, we investigate the impact of color information for scene classification. Figure 6(b) shows the results when classifying the images of scenes with color patches and grey patches. The performance when using color patches is nearly 5%-6% better than that when using grey patches. Color information is an important factor in scene classification. Here, the color patch means fusing QSVD features and Gabor features; the grey patch means only Gabor features.

The comparative results of our method and pLSA proposed in [6] are illustrated in Table 1, where we can get that our method outperforms the pLSA method in classification accuracy and feature dimensions by using grey patches. For color patches, in [6] Bosch et al. eventually adopted the Color SIFT four Concentric Circles (C4CC) patch. The feature dimension of the C4CC patch is  $3 \times 4 \times 128 = 1536$ , so the total dimension of features which are used to build the visual words vocabulary with the OT8 dataset is  $1536 \times 240 \times 961$  (1536 is the feature dimensions  $240 = 30 \times 8$ , where 30 is the number of images for building visual words for each category

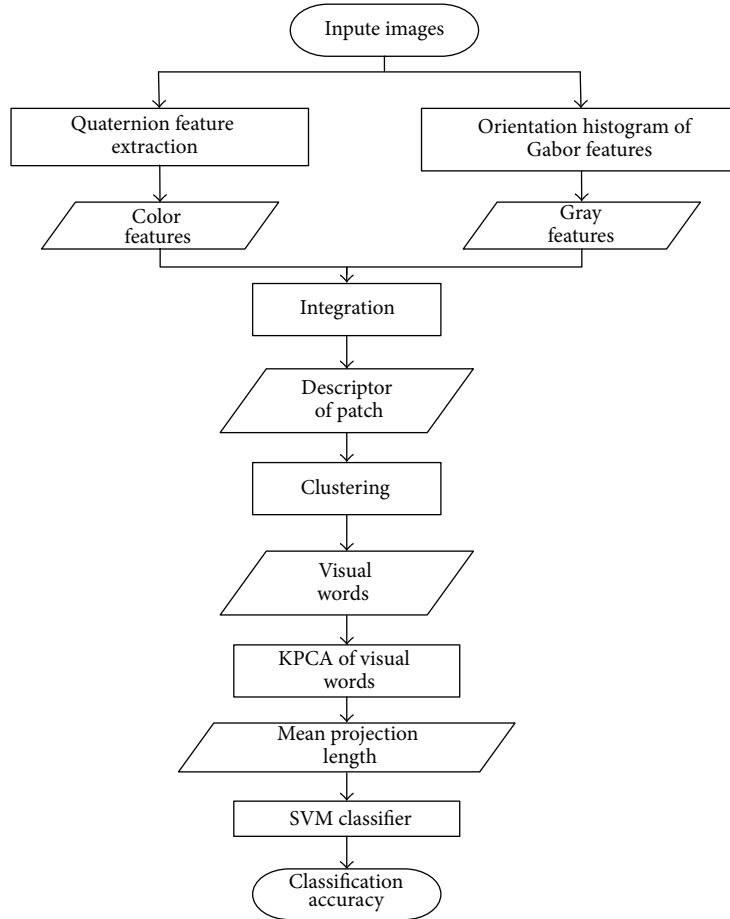


FIGURE 5: Detailed description of the proposed method for scenes classification.

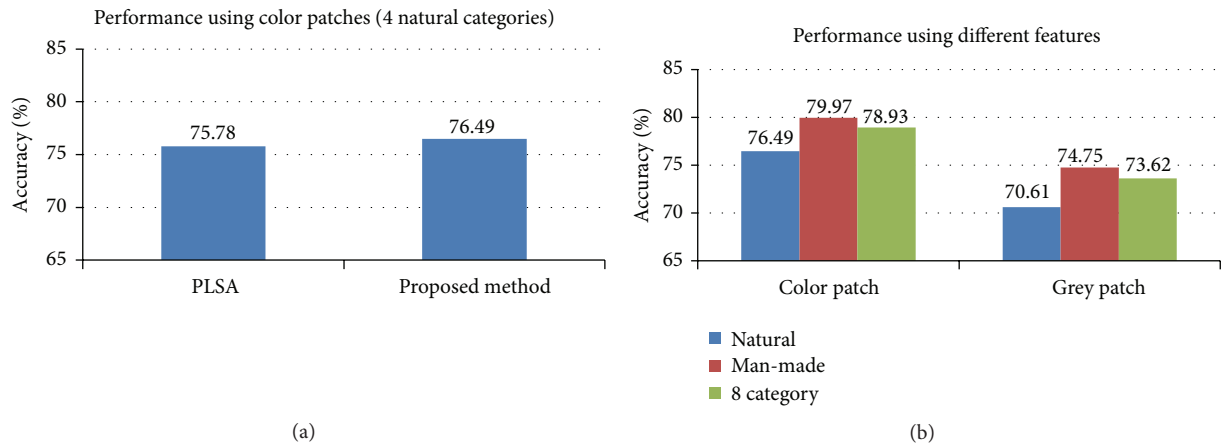


FIGURE 6: Results of proposed method in this paper. (a) Result of comparison to pLSA method; (b) performance when classifying natural, man-made, and all categories using different patches and features (color patch: QSVD features + gabor features; Grey patch: Gabor features).

TABLE 1: Comparative results of our method and pLSA.

Method	Gray patch		Classification accuracy (%)	Color patch		Average execution time for each image (in sec)
	Classification accuracy (%)	Feature dimension		Feature dimension		
Our method	73.62	24	<b>78.93</b>	280	<b>3.06</b>	
pLSA	71.51	121	86.65	1536	7.22	



and 8 is the number of categories; 961 is the number of patches in each image). When using single-precision floating-point integer to represent each feature dimension, it will totally take more than 2.5 G of memory to store the matrix used to build visual words. Therefore, this method will spend a lot of time in feature extraction, clustering, and quantification. The average processing time for each image is about 7.22 seconds. As a comparison, the feature dimension in our method is  $24 + 256 = 280$ , and the total dimension of features used to build the visual words vocabulary with the OT8 dataset is  $280 \times 240 \times 256$ . When using single-precision floating-point integer to represent each feature dimension, it only takes less than 0.13 G of memory to store the matrix used to build visual words. And the average processing time for each image in our method is about 3.06 seconds. Our method has an advantage over the method proposed in [6], regarding the run time, computation speed, and memory requirements.

Even if [6] has achieved higher classification accuracy than that of our method by using color patches, our method is superior to theirs in processing time, feature dimensions and hardware requirements. In some real-time emergency response systems, we need to consider the processing time firstly, then consider the hardware requirements, and finally consider the accuracy. In real-time emergency response systems about scene classification, accuracy is not the ultimate goal, quickly and accurately locating the target is the final demand. So the feature dimensions should not be too high. By our method, the feature dimension is lower than Bosch's method and processing time is also faster than [6]. Thus our method can be used in real-time emergency response systems about scene classification.

## 7. Conclusions

In this paper, we propose a scene classification method for color scene images. We use quaternion matrices to represent the color images and extract color feature by quaternion singular value decomposition. In this way, we take into account the direct interchannel effects in the RGB channels and reduce the dimension of the feature vector. To get the optimal texture information in time domain and frequency domain, we use Gabor filter to extract grey features. We also find the optimal parameters for getting the best classification accuracy by cross-validation experiments. From the evaluation results, we can observe that our method can classify the input image to the right category fastly and accurately. This work can be further extended to classify scene images in some real-time emergency response systems. In our method, for building the visual words vocabulary, we adopt the k-means algorithm, but the calculation of this algorithm is high. In the future work, we can consider the improvements in clustering and quantitative technologies to improve the classification performance. We also will look for better ways to improve the classification rate.

## Acknowledgments

This paper is supported by the Major Project of Chinese National Programs for Fundamental Research and Development (973 program, no. 2011CB302600), the Third Stage

Building of "211 Project" (Grant no. S-10218), the Fundamental Research Funds for the Central Universities in China (CDJZR1180004), and the National "12th Five-Year Technology Support Program" Medicinal Drugs and Herbs E-commerce Services Technology Research, Development and Application Demonstration (SQ2012BAJY3690).

## References

- [1] W. Huang and Z. Jing, "Evaluation of focus measures in multi-focus image fusion," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 493–500, 2007.
- [2] P. Xueqin, "Influence of image fusion methods on land cover/use classification," in *Proceedings of the Symposium from Cross-Strait Environment & Resources and 2nd Representative Conference of Chinese Environmental Resources & Ecological Conservation Society*, pp. 87–90, 2010.
- [3] M. Szummer and R. Picard, "Indoor-outdoor image classification," in *Proceedings of the IEEE International Workshop on Content-Based Access of Image and Video Database*, pp. 42–51, Bombay, India, 1998.
- [4] S. Paek and S. F. Chang, "A knowledge engineering approach for image classification based on probabilistic reasoning systems," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '00)*, pp. 1133–1136, August 2000.
- [5] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 524–531, June 2005.
- [6] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proceedings of the 9th European Conference on Computer Vision*, pp. 517–530, 2006.
- [7] K. Hotta, "Local autocorrelation of similarities with subspaces for shift invariant scene classification," *Pattern Recognition*, vol. 44, no. 4, pp. 794–799, 2011.
- [8] K. Hotta, "Scene classification based on multi-resolution orientation histogram of Gabor features," in *Proceedings of the International Conference on Computer Vision Systems*, vol. 5008 of *Lecture Notes in Computer Science*, pp. 291–301, 2008.
- [9] Q. Luo, "Research on E-learning system using speech emotion recognition," *Advanced Science Letters*, vol. 5, no. 1, pp. 363–366, 2012.
- [10] R. S. Ran and T. Z. Huang, "Color face recognition based on singular value feature extraction," *Application Research of Computers*, vol. 24, no. 7, pp. 298–299, 2007.
- [11] K. Hotta, "Object categorization based on kernel principal component analysis of visual words," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '08)*, pp. 1–8, January 2008.
- [12] [http://vision.stanford.edu/resources\\_links.html](http://vision.stanford.edu/resources_links.html).
- [13] R. S. Ran and T. Z. Huang, "The recognition of color images based on the singular value decompositions of quaternion matrices," *Scientific Journal of Computer Science*, vol. 33, no. 7, pp. 227–229, 2006.
- [14] Y. Xing and J. Q. Tan, "Color image decomposition based on the SVD of quaternion matrix," *Journal of Engineering Graphics*, vol. 32, no. 2, pp. 93–101, 2011.
- [15] A. P. Yefremov, "Splitting of 3D quaternion dimensions into 2D-sells and a, 'World Screen Technology,'" *Advanced Science Letters*, vol. 5, no. 1, pp. 288–293, 2012.

- [16] A. P. Yefremov, "Solutions of dynamic equations in quaternion model of relativity," *Advanced Science Letters*, vol. 3, no. 2, pp. 236–240, 2010.
- [17] S. D. M. Raja and A. Shanmugam, "ANN and SVM based war scene classification using wavelet features: a comparative study," *Journal of Computational Information Systems*, vol. 7, no. 5, pp. 1402–1411, 2011.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

