Full-viewpoint 3D Space Object Recognition Based on Kernel Locality Preserving Projections

Meng Gang, Jiang Zhiguo*, Liu Zhengyi, Zhang Haopeng, Zhao Danpei

School of Astronautics, Beijing University of Aeronautics and Astronautics, Beijing 100191, China

Received 2 February 2010; accepted 3 June 2010

Abstract

Space object recognition plays an important role in spatial exploitation and surveillance, followed by two main problems: lacking of data and drastic changes in viewpoints. In this article, firstly, we build a three-dimensional (3D) satellites dataset named BUAA Satellite Image Dataset (BUAA-SID 1.0) to supply data for 3D space object research. Then, based on the dataset, we propose to recognize full-viewpoint 3D space objects based on kernel locality preserving projections (KLPP). To obtain more accurate and separable description of the objects, firstly, we build feature vectors employing moment invariants, Fourier descriptors, region covariance and histogram of oriented gradients. Then, we map the features into kernel space followed by dimensionality reduction using KLPP to obtain the submanifold of the features. At last, k-nearest neighbor (kNN) is used to accomplish the classification. Experimental results show that the proposed approach is more appropriate for space object recognition mainly considering changes of viewpoints. Encouraging recognition rate could be obtained based on images in BUAA-SID 1.0, and the highest recognition result could achieve 95.87%.

Keywords: satellites; object recognition; three-dimensional; image dataset; full-viewpoint; kernel locality preserving projections

1. Introduction

“Space object” can be understood as including “component parts of a space object as well as its launch vehicle and parts thereof”[1]. The number of space objects has been increasing rapidly in recent years. In 2000, the resident space object catalog contains more than 8 000 objects, consisting of active and inactive satellites, rocket bodies, and debris, with an active subset of over 800 objects[2]. Up to January 4, 2010, the number of objects in orbit has already reached 15 297, including 3 484 payloads[3]. Owing to military and security requirements, accurate and rapid detection, classification and recognition of space objects are more and more urgently needed in the fields of spatial exploitation and surveillance. In this article, we focus on the issue of recognition of three-dimensional (3D) space objects. To simplify the problem, we limit the space objects to manmade satellites.

There are two main differences between full-viewpoint 3D space object recognition and common object recognition such as human faces or cars: limitation on features of the objects and the uncertainty of viewpoint.

Usually, study of space objects is based on gray images. Thus, color information-based features could not be used. On the other hand, compared with common object recognition, images of satellites taken in deep space are usually small and their texture information is not rich. Furthermore, it does not sounds reasonable to describe the satellites using local descriptors considering the variance of the pose of the objects. Therefore, choosing appropriate descriptions for the objects is the first problem which will be confronted in full-viewpoint 3D space object recognition.

Another difference lies in full-viewpoint. Fig.1 demonstrates a comparison between images of space objects and traditional multi-viewpoint objects. For traditional “multi-viewpoint” or “full-viewpoint” object recognition, there is an implication assumption that the viewpoint changes in a certain range. For example, for human face recognition, the viewpoint only changes in the front part of the head; and for the chair, the viewpoints are limited to the above space of the object. However, image of a space object could be taken at any point in the sphere centered at the object, and the appearance of the same satellite changes greatly in images taken from different viewpoints, as

*Corresponding author. Tel.: +86-10-82338061.
E-mail address: jiangzg@buaa.edu.cn
Foundation items: National Natural Science Foundation of China (60776793, 60802043); National Basic Research Program of China (2010CB327900)
shown in Figs.1(a)-1(b). In the extreme it is even difficult for human eyes to accomplish the recognition, such as the last images in Figs.1(a)-1(b). Thus, common algorithms used for 3D object recognition such as matching⁴ and modeling⁵ could not suit our destination very well.

Fig.1 Comparison of images of full-viewpoint 3D space objects and traditional multi-viewpoint objects.

(a) An example of full-viewpoint images of a satellite

(b) Another example of full-viewpoint images of a satellite

(c) Multi-viewpoint images of a human face

(d) Multi-viewpoint images of a chair

Fig.1 Comparison of images of full-viewpoint 3D space objects and traditional multi-viewpoint objects.

With regard to full-viewpoint 3D space object recognition, there are two main difficulties.

The most important one is lacking of data. Since it is impossible to directly take real images of manmade satellites in the space, most of the studies on space objects are based on simulation. However, to the authors’ best knowledge, there are no image datasets based on space objects containing abundant information. Especially, there are no datasets containing full-viewpoint images of space objects. Thus, construction of 3D space objects dataset is necessary and urgent. In this article, we build an image dataset named BUAA Satellite Image Dataset (BUAA-SID 1.0) to simulate the full-viewpoint images of satellites. So far as we know, it is the first public 3D satellite full-viewpoint dataset.

Another difficulty is caused by full-viewpoint in addition to variance of the appearance of different satellites. As shown in Figs.1(a)-1(b), the appearances of satellites differ from each other obviously. They are different in size, number of wings, shape of the body, etc. And for the same kind of satellite, the appearance also changes apparently because of variance of viewpoint. Although a number of factors, such as changes of scale, illumination, pose, etc., make the recognition of 3D space objects more difficult, most of them could be overcome by means of detection, registration and segmentation. However, full-viewpoint could be overcome by none of the existing technologies. And drastic changes in viewpoint usually lead to failure in recognition. To simplify the problem, we do not consider the changes of scale and illumination in the BUAA-SID 1.0 and just highlight the changes of viewpoint.

Though the realization of full-viewpoint 3D space object recognition is important and urgent, there are few articles in this field. Thus, we have to refer to other similar problems such as multi-viewpoint or full-viewpoint object recognition. To handle the variation of viewpoint, researchers have developed several techniques. Multi-viewpoint human face and facial expression recognition may be the most well studied field. J. Lee, et al.⁶ employed an optimization search to find the optimal views in multi-viewpoint 3D face modeling. Y. Tong, et al.⁷ introduced a unified probabilistic facial action model based on the dynamic Bayesian network to accomplish facial action recognition. D. Smeets, et al.⁵ proposed an isometric deformation model to deal with variations of expression in face recognition. Modeling is another widely used technique. For example, M. Sun, et al.⁸ proposed a probabilistic framework for learning visual models of 3D object categories. However, just as mentioned above, as the kind and viewpoint of the satellite change, neither obtaining a general optimal viewpoint nor building a unified model of different satellites is easy to accomplish. In the presence of affine distortion, K. Yawichai, et al.⁹ proposed a neural network method to accelerate the speed of multi-viewpoint shape recognition using 1-dimensional triangle area representation. However, it is just effective for affine transformation, and does not consider the problems of great changes in viewpoint. Another common and intuitive thought for multi-viewpoint is based on invariant feature points. For example, Y. Z. Wang, et al.¹⁰ employed matching algorithms by tracking scale invariant feature transform (SIFT)¹¹ feature points in the plenoptic domain to enhance the robustness of multi-viewpoint object recognition. However, experiments show that, stable SIFT feature points, influenced by
the characteristic of images of satellites, are hardly to be extracted. Thus, description of space objects should be based on region features, not invariant feature points.

In most cases, it is not enough to describe space objects using just one kind of feature. On the other hand, just as in Ref.[12], multi-feature description is widely used in multi-viewpoint recognition, yet, features should be elaborately chosen. In this article, we combine shape and region features for description. Specifically, we use moment invariants[13], Fourier descriptors[14], region covariance[15] and histogram of oriented gradients[16]. After that, features are mapped into kernel space to get a better representation, based on invariant criterion function for clustering[17].

Though viewpoint changes greatly in full-viewpoint space object recognition and the dimensionality of feature vector is high, there might be reason to suspect that the “intrinsic dimensionality” of the feature is lower. This leads us to consider methods of dimensionality reduction or “graph embedding”[18]. We employ kernel locality preserving projections (KLPP)[19-20] to obtain the lower dimensional manifold of features, which performs better than traditional algorithms of dimensionality reduction such as principal component analysis (PCA)[21] or kernel PCA (KPCA).

As mentioned above, there are two contributions of this article. First, we build a 3D satellite full-viewpoint image dataset named BUAA-SID 1.0. Second, we propose to describe full-viewpoint 3D space objects with moment invariants, Fourier descriptor, region covariance and histogram of oriented gradients, and employ KLPP to achieve graph embedding. To our best knowledge, we are the first to implement 3D space object recognition based on this framework.

2. Satellite Image Dataset

A major problem that handicaps the study of space object recognition is lacking of data. Thus, we build a satellite image dataset named BUAA-SID 1.0 to collect the information of space objects (in this dataset, only manmade satellite models are contained). Since to a certain extent, position, rotation and scale changes of satellites could be ignored by means of detection and registration, in the dataset, we only emphasize viewpoint changes of the objects.

BUAA-SID 1.0 contains two sub-datasets: model dataset and image dataset. Model dataset contains 56 kinds of satellite models created by 3ds Max. For each model in model dataset, we define 230 viewpoints. And for each viewpoint, two images are generated: one is gray image and the other is binary image corresponding to the gray image. Thus, there are totally 460 images for each kind of satellite and 25 760 images in image dataset with the same size 160×120. Currently, only 20 kinds of satellite images in image dataset are released, the number of which is 9 200. In this article, we talk about space object recognition based on these released images.

To partition the viewpoints scattered on the sphere centered in the models, a simple way is to use interior polyhedron. However, there are some problems with this kind of partition. First, it is difficult to precisely express the latitude and longitude of every vertex. Second, it is not convenient for pose estimation of the satellites. Besides, it is not easy to generate images of the satellites based on this kind of partition in 3ds Max. Thus, in BUAA-SID 1.0, viewpoints are defined as follows.

Each viewpoint is defined by its latitude and longitude on the sphere. After the origin is defined, we divide the sphere every 15° in latitude. To assure the viewpoints scatter as uniformly as possible, for each latitude, we change the degree of longitude in the way shown in Table 1.

Table 1 Definition of viewpoints in BUAA-SID 1.0

<table>
<thead>
<tr>
<th>Image number</th>
<th>Latitude(°)</th>
<th>Interval of degrees of longitude(°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-35</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>36-71</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>72-95</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>96-113</td>
<td>45</td>
<td>20</td>
</tr>
<tr>
<td>114-125</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>126-131</td>
<td>75</td>
<td>60</td>
</tr>
<tr>
<td>132</td>
<td>90</td>
<td>1</td>
</tr>
<tr>
<td>133-168</td>
<td>−15</td>
<td>10</td>
</tr>
<tr>
<td>169-192</td>
<td>−30</td>
<td>15</td>
</tr>
<tr>
<td>193-210</td>
<td>−45</td>
<td>20</td>
</tr>
<tr>
<td>211-222</td>
<td>−60</td>
<td>30</td>
</tr>
<tr>
<td>223-228</td>
<td>−75</td>
<td>60</td>
</tr>
<tr>
<td>229</td>
<td>−90</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig.2 demonstrates the definition of viewpoints and Fig.3 shows examples of images in BUAA-SID 1.0. From Fig.3, it can be seen clearly that as the viewpoint changes, the appearance of the satellites changes greatly. The released version of BUAA-SID 1.0 will be found on http://imageprocessing.buaa.edu.cn.
3. Feature Description of 3D Space Objects

As mentioned in Section 1, the first problem which will be encountered in full-viewpoint 3D space object recognition is to describe the objects with proper features. In this article, the chosen features are moment invariants, Fourier descriptor, region covariance, and histograms of oriented gradient (HOG). In order to depress the influence of noise, before extracting these features, we smooth the image using Gaussian low pass filter.

3.1. Moment invariant and Fourier descriptor

Considering the characteristic of images of the satellites, an intuitive feature of the objects is based on shape. M. K. Hu’s seven moment invariants\(^{[13]}\) and Fourier descriptor\(^{[14]}\) are popular shape-based image invariants that are widely used in image recognition and indexing. Fourier descriptors are boundary-based image features which only compute the pixels along the image contours. M. K. Hu’s seven moment invariants are region-based image features which take all of the pixels of the image into account. Both Fourier descriptors and M. K. Hu’s seven moment invariants have the invariance property against affine transformations including scale change, translation and rotation.

Binary images in BUAA-SID 1.0 are used to generate the 7-dimensional moment invariants and 20-dimensional normalized Fourier descriptors.

Fig.4 shows the recognition rate of different dimensionality of Fourier descriptor. It is clear that using only Fourier descriptor to describe the object, the recognition rate increases obviously as the dimensionality increases. However, the recognition rate increases slowly after the number of the dimensionality reaches 20. Considering the complexity of computation, the dimensionality of Fourier descriptor is chosen to be 20.

Fig.3 Examples of images in BUAA-SID 1.0.
vector to obtain more accurate description.

3.2. Region covariance

Region covariance [15] is a region descriptor that attracts much attention in the fields of detection and classification [22-23], recently. It does not have any information regarding the ordering and the number of points. This implies a certain scale and rotation invariance over the regions in different images. For each input image, we extract the following 8-dimensional set of features for each pixel \((x, y)\):

\[
F_{RC} = \begin{bmatrix}
    x & y & |I_x| & |I_y| & \sqrt{I_x^2 + I_y^2} \\
    |I_{xx}| & |I_{yy}| & \arctan(I_y/I_x)
\end{bmatrix}^T
\]

where \(I_x, I_y\) and \(I_{xx}, I_{yy}\) are respectively the first and the second-order derivatives of the image. Given the rectangular region of each satellite, we can compute the covariance matrix of the feature set \(F_{RC}\) inside the window and obtain a 64-dimensional feature vector for each gray image.

3.3. Histogram of oriented gradients based on integral histogram

HOG [18] is a widely used gradient-based feature. To accelerate the calculation of HOG, we employ integral histogram in this article.

Integral histogram [24] is the expansion of integral image [25]. After the calculation of integral histogram of each pixel, the histogram of a rectangle could be obtained by simple add and subtraction operations.

Step 1 Calculate the magnitude \(G(x,y)\) and orientation \(\theta(x,y)\) of each pixel \((x,y)\) in the following way:

\[
\theta(x,y) = \begin{cases} 
    \theta(x,y) + \pi & \text{if } \theta(x,y) < 0 \\
    \theta(x,y) & \text{otherwise} 
\end{cases}
\]

The histogram of each pixel could be obtained by:

\[
H(x,y,k) = H(x-1,y,k) + H(x,y-1,k) - H(x-1,y-1,k) + Q(\tilde{\theta}(x,y)) \quad (k = 1,2,\ldots,b)
\]

where \(b\) is the number of bins, \(H(x,y,k)\) the summation of the orientation in the \(k\)th bin in integral histogram, \(Q(\tilde{\theta}(x,y))\) the weight of the \(k\)th bin that is obtained according to orientation \(\tilde{\theta}(x,y)\). \(Q(\tilde{\theta}(x,y))\) is decided by the magnitude \(G(x,y)\) corresponding to \((x,y)\).

As shown in Fig.5, the histogram of a rectangle in the image could be obtained as follows:

\[
V(k) = H(x_2,y_2,k) + H(x_1,y_1,k) - H(x_2,y_1,k) - H(x_1,y_2,k) \\
     (k = 1,2,\ldots,b)
\]

Fig.5  Calculation of histograms with integral histogram.

Step 2 Get features of the “cells” in the object. We divide the region of the object into \(M\times N\) cells (as shown in Fig.6) and calculate the histogram of each cell \(H_{ij}\) \((i = 1,2,\ldots,M; j = 1,2,\ldots,N)\) separately.

Step 3 Concatenate features of the cells to describe the object.

\[
F_{HOG} = [H_{11}, H_{12}, \ldots, H_{1N}, H_{21}, H_{22}, \ldots, H_{2N}, \ldots, H_{MN}] 
\]

To simplify the calculation of the HOG feature, the number of bins is 36 and we take the whole image as one cell, which means \(M = 1\) and \(N = 1\). Based on descriptions mentioned above, feature descriptors of each satellite are built to be a 127-dimensional vector as follows:

\[
F = [F_M, F_F, F_{RC}, F_{HOG}]
\]

where \(F_M, F_F\) and \(F_{HOG}\) are feature vectors generated by moment invariants, Fourier descriptor and HOG separately. Table 2 shows details of the dimensionality of feature vectors.

Table 2  Features used in proposed approach

<table>
<thead>
<tr>
<th>Moment invariant</th>
<th>Fourier descriptor</th>
<th>Region covariance</th>
<th>HOG</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>20</td>
<td>64</td>
<td>36</td>
<td>127</td>
</tr>
</tbody>
</table>

4. 3D Space Object Recognition Based on BUAAA-SID 1.0 with KLPP

4.1. Features in kernel space

As mentioned in Ref.[17], eigenvalues \(\lambda_1,\lambda_2,\ldots,\lambda_d\) of \(S_w^{-1}S_B\) are invariant under nonsingular transformations of the data, where \(d\) is the dimensionality of the feature, \(S_w\) is within-cluster matrix and \(S_B\) is be-
tween-cluster matrix. Thus, if we define:

- mean vector for the $i$th cluster $\mathbf{m}_i = \frac{1}{n_i} \sum x$ 
- total mean vector $\mathbf{m} = \frac{1}{n} \sum x = \frac{1}{n} \sum n_i \mathbf{m}_i$
- scatter matrix for the $i$th cluster $S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T$ (6)

where $x$ is a feature vector, $D$ the set of all samples, $D_i$ the set of samples of the $i$th class, $n_i$ the size of $D_i$, there are $n$ clusters and $c$ kinds of classes. Then,

$$S_W = \sum_{i=1}^c S_i$$ (7)

$$S_B = \sum_{i=1}^c n_i(m_i - m)(m_i - m)^T$$ (8)

The criterion function for clustering is as follows:\[^1\] \[^7\] \[^17\] :

$$J_x = \text{tr}(S_W^{-1}S_B) = \sum_{i=1}^d \lambda_i$$ (9)

In order to maximize the criterion function, we refer to the famous kernel trick\[^{[26]}\]. The basic idea of kernel method is to map original features into the kernel space. Suppose that the Euclidian space $\mathbb{R}^n$ is mapped to a Hilbert space $\mathcal{H}$ through a nonlinear mapping function $\phi: \mathbb{R}^n \rightarrow \mathcal{H}$ and let $\phi(x)$ denote the data matrix in the Hilbert space $\phi(x)=[\phi(x_1), \phi(x_2), \cdots, \phi(x_n)]$, the kernel function could be written as follows:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) = (\phi(x_i))^T \phi(x_j)$$ (10)

We employ the invariant criteria mentioned above to compare features in Euclidian space and kernel space. Results are shown in Table 3. The item “none” means we calculate the criterion function $J_x$ without any transformation. It is clear that features in kernel space are better than in Euclidian space. To the view of the criterion function, features transformed by LPP and KLPP are better than transformed by PCA and KPCA. And among the four algorithms, KLPP is the best to extract the features. Based on the criteria, features extracted by KLPP are better than others. This implies the argument that features in kernel space are more appropriate than in Euclidian space for space object recognition mainly considering changes of viewpoints.

| Table 3 Results of invariant criteria function $J_x$ of different algorithms |
|-----------------|---------|---------|---------|---------|
| None            | PCA     | KPCA    | LPP     | KLPP    |
| 13.082 6        | 10.018 5 | 11.779 6 | 25.814 9 | 613.646 8 |

4.2. Dimensionality reduction based on kernel locality preserving projections

As mentioned above, the dimensionality of the descriptor for each kind of satellite is 127, which is large. However, there might be reasons to suspect that the “intrinsic dimensionality” of the data is much lower. This leads to the consideration of methods of dimensionality reduction that allow one to represent the data in a lower dimensionality space.

PCA is a widely used traditional dimensionality reduction method. And KPCA is an extension of PCA in kernel space. In Ref.[27], the authors declare to apply KPCA to appearance-based 3D object recognition and pose estimation for the first time and obtained excellent recognition rates. However, neither PCA nor KPCA could improve the recognition ratio of full-viewpoint 3D space objects greatly, which will be talked about in Section 5.4.

In Ref.[19], an alternative to PCA called locality preserving projections (LPP) is proposed to obtain the low dimensional manifold embedded in the ambient space. The projections are obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. KLPP is the implementation of LPP in kernel space. In this article, we use features in kernel space and follow the steps listed in Ref.[19] to get the low dimensional embedding.

Step 1 Construct the adjacency graph. We employ the “supervised” mode to construct the graph, which means two nodes will be connected by an edge only if they belong to the same class.

Step 2 Heat kernel $\tilde{W}$ is used to weight the edges. If node $i$ and $j$ are connected, the weight of the edge is calculated as Eq.(11). The determination of the value of $t$ will be talked about in Section 5.3.

$$\tilde{w}_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{t^2}\right)$$ (11)

Step 3 Eigenmaps: compute the eigenvectors and eigenvalues based on the formula

$$\tilde{KL}K\tilde{a} = \lambda \tilde{KD}\tilde{K}^\dagger \tilde{a}$$ (12)

where $\tilde{D}$ is a diagonal matrix whose entries are column sums of $\tilde{W}$, the $j$th entry of $\tilde{D}$, $\tilde{D}_{jj} = \sum_i \tilde{w}_{ij}, \tilde{L} = \tilde{D} - \tilde{W}$ is the Laplacian matrix. The $i$th column of matrix $\tilde{K}$ is $\phi(x_i)$.

Let the column vectors $a_0, a_1, \cdots, a_{r-1}$ be the solutions of Eq.(12), ordered according to their eigenvalues, $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{r-1}$. Then, the embedding is

$$\phi(x_i) \rightarrow y_i = A^T \phi(x_i)$$

$$A = [a_0^T, a_1^T, \cdots, a_{r-1}^T]$$ (13)

5. Experimental Results

For images in BUAA-SID 1.0, half of the images are used for training and others are used for recognition. The numbers of images in these two sets are both 4,600, including 2,300 gray images and 2,300 binary images of 20 kinds of satellites. Since the results are decided by viewpoints of training, training images
could not be generated randomly. For two images corresponding to two adjacent viewpoints, one is put into training set and the other is put into testing set. As to the number of the images, training set is composed of the ones with odd numbers that is No.1, No.3, ..., No.229, totally 115 viewpoints, 230 images; and the others constitute the testing set.

Before talking about the experiments, we will define some symbols. \( F^i_j \) is the feature of the \( j \)th viewpoint of the \( i \)th kind of satellite extracted using Eq.(5); \( F^m \) is the feature matrix combining features of all the 115 training viewpoints for the \( i \)th kind of satellite; \( F_m \) is the feature of the \( m \)th image in the testing set; \( \phi(F^j_i) \), \( \phi(F^m) \) are the maps of \( F^j_i \), \( F^m \) in kernel space separately; \( y^j_i \) is the feature in submanifold corresponding to \( F^j_i \); and \( y_m \) is the feature in submanifold corresponding to \( F_m \).

Before training and testing, we have to extract \( F^j_i(i=1,2,\cdots,20; j=1,2,\cdots,115) \). To accelerate the process, we extract features in the dataset all together. It takes 146.80 s to get features for training set and 146.73 s for testing set.

The process of training is as follows. For \( F^j_i \), after the calculation of \( \phi(F^j_i) \), we get the optimal transformation matrix \( A_i \) using \( \phi(F^i) \) and \( y^j_i(i=1,2,\cdots,20; j=1,2,\cdots,115) \) using \( A_i \).

In the process of testing, we obtain \( \phi(F_m) \) using \( F^m \) and \( y_m \) using \( A_i \); then, \( k \)-nearest neighbor (\( k \)NN) is employed to accomplished the classification based on \( y^j_i \) and \( y_m \).

\( k \)NN can be viewed as an attempt to estimate the posterior probabilities \( P(o|x) \) from samples \( x \). We want to use a large value of \( k \) to obtain a reliable estimation. On the other hand, we want all of the \( k \) nearest neighbors \( x' \) to be very near \( x \) to be sure that \( P(o|x') \) is approximately the same as \( P(o|x) \). This forces us to choose a tradeoff value of \( k \) that is a small fraction of the number of samples. Details of comparison of recognition results using different \( k \) neighbors will be talked about in Section 5.2. And Table 4 is the comparison of processing time for each kind of algorithm.

### Table 4 Comparison of time consuming of different algorithms (20 kinds of satellites)

<table>
<thead>
<tr>
<th>Processing time</th>
<th>PCA</th>
<th>KPCA</th>
<th>LPP</th>
<th>KLPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time/s</td>
<td>0.149</td>
<td>2.565</td>
<td>2.626</td>
<td>0.446</td>
</tr>
<tr>
<td></td>
<td>0.776</td>
<td>1.292</td>
<td>1.547</td>
<td>0.811</td>
</tr>
<tr>
<td></td>
<td>0.806</td>
<td>0.897</td>
<td>0.806</td>
<td>0.610</td>
</tr>
</tbody>
</table>

### 5.1. Feature extraction experiment

Fig.7 shows the results of recognition rates using different dimensionalities of features. The 27-dimensional features are composed of moment invariant and Fourier descriptors, and the 127-dimensional features are composed of \( F_M \), \( F_B \), \( F_RC \) and \( F_HOG \) as mentioned above. The parameter \( k \) in \( k \)NN is chosen to be 30.

![Fig.7 Results of recognition rates using different dimensionalities of features.](image)

As the reduced dimensionality increases, the recognition rate increases slowly. However, after the reduced dimensionality of the features achieves 20 or more, the recognition rates keep stable. This is caused by the approximation of eigenvalues. When it is near zero, the eigenvalue and the corresponding eigenvector will be ignored. As shown in Fig.7, features of 127-dimensionality are more descriptive than the ones of 27-dimensionality, and KLPP with 127-dimensional features is the most descriptive.

### 5.2. Comparison of dimensionality reduction algorithms

This experiment is designed to show the influence of the parameter \( k \) in \( k \)NN to different dimensionality reduction algorithms. The feature \( F \) is extracted beforehand.

Fig.8 demonstrates comparison of recognition results using different algorithms and different \( k \) neighbors. As shown in Fig.8, it is clear that recognition rates increase as the dimensionality of the features increase. Results of PCA and KPCA decline obviously when \( k \) increases; however, results of LPP and KLPP are hardly influenced by the value of \( k \), which means that features extracted by LPP and KLPP are more stable than extracted by PCA and KPCA. And recognition rates of KLPP are higher than LPP, which has been mentioned above. Comparing results of different values of \( k \), the recognition rate keeps stable and acceptable when \( k=30 \).

### 5.3. Recognition results with different \( t \) in KLPP

The results of KLPP are greatly influenced by its parameters. This experiment is designed to determine the parameter \( t \) in Eq.(11) and the parameter \( t' \) in the kernel.

Fig.9 is the results of KLPP with different values of
the weight $t$. We employ “Gaussian kernel”. It can be seen from the figure that given a value of $t$, there are local maximums of the recognition rate. Different values of $t$ correspond to different local maximums.

When $t \geq 1$, the recognition results are almost the same. And the global maximum achieves when $t = 0.1$. Thus we choose $t = 1$ and $t' = 0.1$. The maximum recognition rate is 95.87%.

![Recognition results using different dimensionality reduction methods and different $k$ neighbors.](image)

**Fig.8** Recognition results using different dimensionality reduction methods and different $k$ neighbors.

**5.4. Comparison of recognition rates**

Fig.10 is the comparison of recognition rates. The parameters in KLPP are $t = 1$ and $t' = 0.1$. The $k$ in $k$NN is chosen to be 30.

Fig.10(a) is the comparison of the recognition rates of each kind of satellites, Fig.10(b) shows the comparison of total recognition rates, and Fig.10(c) shows the satellites corresponding to the numbers in Fig.10(a). For most kinds of satellites, KLPP performs better than other algorithms, except No.10 and No.19, where recognition rates of KLPP are lower than LPP. Thus, we can get the conclusion that the proposed approach works well in space object recognition and is better than other algorithms listed in this article in most cases.

Table 5 is the maximum recognition rates of different algorithms. The recognition rate with the proposed approach could achieve 95.87%.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recognition Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>69.57</td>
</tr>
<tr>
<td>PCA</td>
<td>68.57</td>
</tr>
<tr>
<td>KPCA</td>
<td>69.70</td>
</tr>
<tr>
<td>LPP</td>
<td>88.22</td>
</tr>
<tr>
<td>KLPP</td>
<td>95.87</td>
</tr>
</tbody>
</table>

**5.5. Recognition results of images with noise**

A big problem that will be encountered in space object research is the existence of noise. This experiment is designed to demonstrate the effectiveness of the
proposed method when noise exists.

The most common kind of noise in images of space object is Gaussian white noise. Thus, in the testing set, we add Gaussian white noise into each image with different parameters. Fig.11 is the experimental results. For the Gaussian white noise, the mean is 0 and we use different variance.

From the results, we can see that as the variance increases the recognition rate decreases. And the results of the proposed algorithm perform the best in them. However, it is clear that noise does influence the recognition results obviously.

6. Conclusions

We build a 3D satellites dataset named BUAA-SID 1.0 and propose to recognize full-viewpoint 3D space object based on KLPP. Experiments show that the proposed approach could provide an effective solution to space object recognition and performs better than other algorithms listed even when noise exists. The recognition rate could achieve 95.87%.

The shortage of the proposed approach, however, is obvious: a great number of full-viewpoint images of the objects are needed beforehand. Thus, we will focus on how to correctly recognize 3D objects with limited training examples. And we will test the effectiveness of the proposed approach on images of other objects models.

References


Biographies:

Meng Gang Born in 1982, he received his B.S. and M.S. degrees from South China University of Technology, in 2004 and 2007, respectively. He is now a Ph.D. candidate in Beijing University of Aeronautics and Astronautics, China. His research interests include image processing, pattern recognition, machine learning and embedded system.
E-mail: menggangmark@126.com

Jiang Zhiguo Born in 1965, he is a professor at Beijing University of Aeronautics and Astronautics, China. He currently serves as a standing member of the Executive Council of China Society of Image and Graphics and also a member of the Executive Council of Chinese Society of Astronautics. He is an Editor of Chinese Journal of Stereology and Image Analysis.
E-mail: jiangzg@buaa.edu.cn

Liu Zhengyi Born in 1986, he received his B.S. degree from the School of Electronic and Information Engineering, Xi’an Jiaotong University, Xi’an, China in 2008. He is currently working towards his M.S. degree at School of Astronautics, Beijing University of Aeronautics and Astronautics, Beijing, China. His research interests include image processing and pattern recognition.
E-mail: lsticklli@gmail.com