Multi-view space object recognition and pose estimation based on kernel regression

Zhang Haopeng a,b,*, Jiang Zhiguo a,b

a Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China
b Beijing Key Laboratory of Digital Media, Beijing 100191, China

Received 12 September 2013; revised 10 December 2013; accepted 7 January 2014
Available online 15 March 2014

KEYWORDS
Kernel regression;
Object recognition;
Pose estimation;
Space objects;
Vision-based

Abstract The application of high-performance imaging sensors in space-based space surveillance systems makes it possible to recognize space objects and estimate their poses using vision-based methods. In this paper, we proposed a kernel regression-based method for joint multi-view space object recognition and pose estimation. We built a new simulated satellite image dataset named BUAA-SID 1.5 to test our method using different image representations. We evaluated our method for recognition-only tasks, pose estimation-only tasks, and joint recognition and pose estimation tasks. Experimental results show that our method outperforms the state-of-the-arts in space object recognition, and can recognize space objects and estimate their poses effectively and robustly against noise and lighting conditions.

© 2014 Production and hosting by Elsevier Ltd. on behalf of CSAA & BUAA.
Open access under CC BY-NC-ND license.

1. Introduction

Space surveillance systems are very important for space situational awareness. More and more states are pursuing national space surveillance systems to improve space surveillance ability. To detect, track, catalog, and identify man-made objects orbiting Earth is one of the fundamental requirements of space surveillance. Space-based space surveillance systems can improve the capability of space surveillance by using space-based visible sensors, as they can avoid the impacts of atmosphere and clouds, which are suffered by ground-based systems. Space objects considered by space surveillance systems include active/inactive satellites, spent rocket bodies, fragmentation debris, etc., among which satellites are usually paid more attention to due to their importance for military and civilian uses. Space objects mentioned in this paper mainly refer to satellites.

With the application and development of high-performance imaging sensors in space-based systems, such as visible, infrared, laser, radar, etc., high-quality image data can be obtained for vision systems used in the aerospace. On one hand, vision-based pose estimation of space objects has been widely used for autonomous rendezvous and docking, satellite navigation, on-orbit self-serving, etc. Most pose estimation methods used in binocular or monocular vision systems need camera calibration, or require optical markers on a target satellite. Besides, due to the principle of binocular vision, binocular vision systems cannot provide useful range information...
if the distances from the target satellite to the cameras are far greater than that between the two cameras. Zhang et al.\textsuperscript{12} introduced a monocular vision-based method without the requirement of camera calibration or optical markers, in which a generative model was learned only by training images using computer vision technology. On the other hand, recently, there has been an increasing interest in space object recognition from multi-view images. Fan et al.\textsuperscript{13} extracted combined moment invariants as features and used a neural network to recognize space objects. Wang et al.\textsuperscript{14} proposed an improved kernel clustering algorithm based on Voronoi distance for space object recognition. Meng et al.\textsuperscript{15} introduced kernel locality preserving projections to reduce the dimensionality of the combined features describing multi-view space objects, and achieved effective recognition results on a satellite image dataset. Shi et al.\textsuperscript{16} also tested their elastic net sparse coding-based space object recognition method on the same dataset as Meng et al.\textsuperscript{11} Ding et al.\textsuperscript{17} proposed an approach for autonomous space object identification based on normalized affine moment invariants and illumination invariant multi-scale autoconvolution. These space object recognition methods only concern the recognition problem and formulate the viewpoint (pose) variance as a factor that affects the robustness of the methods, not attempting to estimate the poses of space objects simultaneously when recognizing them. To the best of our knowledge, there are few works formulating vision-based space object recognition and pose estimation in one framework.

In this paper, we propose a kernel regression-based method for joint multi-view space object recognition and pose estimation. Kernel regression\textsuperscript{18} is a nonparametric regression method that can avoid solving for the unknown parameters in a regression model, and learn the nonlinear map between the input variable and the output variable only using training data. Therefore, it is widely applied in the areas of signal processing,\textsuperscript{19} image processing,\textsuperscript{20} computer vision,\textsuperscript{21} etc. In this paper, we introduce a kernel regression model for multi-view space object recognition, and constrain the output of the model to better estimate the poses of space objects. We can achieve recognition and pose estimation in one kernel regression-based framework. Our method can be regarded as a monocular vision-based method, and does not need camera calibration or optical markers when estimating poses. Compared to the generative model in Ref.\textsuperscript{6}, our method is discriminative, so that it can get solutions more rapidly by avoiding the searching approach when solving the model. Experimental results validate the effectiveness and robustness of our method.

The rest of the paper is organized as follows: Section 2 describes the details of the kernel regression-based method for multi-view space object recognition and pose estimation. Experimental results are shown in Section 3. The last section comes to conclusions.

2. Methods

2.1. Kernel regression framework

Regression analysis is a group of statistical methods for estimating relationships among variables, including linear regression, nonlinear regression, logistic regression, nonparametric regression, etc. Regression analysis is regarded as a supervised learning method\textsuperscript{22} in the area of pattern recognition and machine learning, which is used to predict the continuous values of target variables when input variables are given. The general form of a regression model can be written as

\[
y \approx f(x, \beta)
\]  

where \(f\) is the regression function describing the relationship between the input variable \(x\) and the output variable \(y\). The input variable \(x\) can be a scalar or a vector, while the output variable \(y\) is usually a scalar. The unknown parameter \(\beta\) is a scalar if there is only one parameter or a vector if there are multiple parameters.

Kernel regression is a nonparametric regression method used to discover nonlinear relationships among variables. Given the training data \(\{(x_i, y_i) \in \mathbb{R}^2, y_i \in \mathbb{R}, i = 1, 2, \ldots, M\}\), where \(\mathbb{R}\) stands for Euclid space, \(D\) is the dimensionality of the input variable, and \(M\) is the number of training data points, according to the representation theorem,\textsuperscript{23} the regression function can be represented as a linear combination of kernels around the training data points (or a subset of them). Thus, we can get the regression function in the following form

\[
y = f(x) = \sum_{i=1}^{M} \alpha_i k(x, x_i)
\]  

where the kernel function \(k(\cdot, \cdot)\) can measure the similarity between the inputs, and the coefficients \(\alpha_i\) can be learned from the given training data via solving a system of linear equations.

Let \(\mathbf{z} = [z_1, z_2, \ldots, z_M]^T \in \mathbb{R}^M\) and \(\mathbf{k}(x) = [k(x, x_1), k(x, x_2), \ldots, k(x, x_M)]\), then Eq. (2) can be rewritten as

\[
y = f(x) = \mathbf{k}(x)\mathbf{z}
\]  

The coefficient vector \(\mathbf{z}\) can be obtained by solving the following equation

\[
y = \mathbf{K}\mathbf{z}
\]  

where \(y = [y_1, y_2, \ldots, y_M]^T \in \mathbb{R}^M\) and the kernel matrix

\[
\mathbf{K} = \begin{bmatrix}
k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_M) \\
k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_M) \\
\vdots & \vdots & \ddots & \vdots \\
k(x_M, x_1) & k(x_M, x_2) & \cdots & k(x_M, x_M)
\end{bmatrix} \in \mathbb{R}^{M \times M}
\]  

If the \(\mathbf{K}^{-1}\) exists, i.e., Eq. (4) is well-posed, \(\mathbf{z}\) can be easily solved as

\[
\mathbf{z} = \mathbf{K}^{-1}y
\]  

Otherwise, if \(\mathbf{K}^{-1}\) does not exist, i.e., the kernel matrix \(\mathbf{K}\) is not full-ranked, Eq. (4) is ill-posed. Then, we can obtain a regularized solution using Tikhonov regularization\textsuperscript{24} as

\[
\mathbf{z} = (\mathbf{K}^T\mathbf{K} + \lambda I)^{-1}\mathbf{K}^T y
\]  

where \(I\) is an \(M\)-order identity matrix, and \(\lambda > 0\) is the regularization parameter. When \(\lambda \rightarrow 0\),

\[
\mathbf{K}^T \mathbf{K} + \lambda I \rightarrow \mathbf{K}^T \rightarrow \mathbf{K}^T
\]  

where \(\mathbf{K}^T\) denotes the Moore-Penrose pseudo-inverse of \(\mathbf{K}\).

In this paper, we particularly use Gaussian kernel to measure the similarity between the inputs, i.e.,

\[
k(x, x_i) = \exp[-\|x - x_i\|^2/(2\sigma^2)]
\]
where \(|\cdot|\) denotes the second norm in \(\mathbf{R}^p\), and the scale parameter \(\sigma^2\) is set as

\[
\sigma^2 = \max_{i,j \in \{1,2,\cdots, M\}} \|x_i - x_j\|^2 / 10
\]  

Notice that other kernels may also work for the kernel regression model. Experimental analysis of how to choose the kernel is described in Subsection 3.3.

2.2. Kernel regression for object recognition

When kernel regression is used for a recognition task, the output should be class labels. For a \(C\)-class problem, we define the output as a label vector \(y_i = [0, 0, \cdots, 0, 1, 0, 0, \cdots, 0] (i = 1, 2, \cdots, M)\), where \(y_i \in \mathbf{R}^C\) and the value “1” locates at the \(c\)-th dimension of the vector for class \(c (c = 1, 2, \cdots, C)\), so as to improve the discriminability of the outputs. Then, the training data for the recognition task can be written as \(\{(x_i, y_i) | x_i \in \mathbf{R}^p, y_i \in \mathbf{R}^C, i = 1, 2, \cdots, M\}\). For the case of a vector output, i.e., the dimensionality of the output variable is more than one, we can learn kernel regression models for each dimension. Therefore, the coefficient vector in Eq. (3) is extended to a coefficient matrix, of which the columns are the coefficient vectors of kernel regression models for each dimension of the outputs. Thus, the kernel regression model for object recognition can be

\[
y = f(x) = \kappa(x) A
\]  

where the coefficient matrix \(A \in \mathbf{R}^{M \times C}\) can be obtained by solving the following equation

\[
Y = KA
\]  

where

\[
Y = \begin{bmatrix} y_1 \newline y_2 \newline \vdots \newline y_M \end{bmatrix} \in \mathbf{R}^{M \times C}
\]  

and the kernel matrix \(K\) follows the same formulation as Eq. (5).

Considering the existence of \(K^{-1}\), i.e., the rank of \(K\), we can get the coefficient matrix as

\[
A = \begin{cases} K^{-1} Y & \text{K is full-ranked} \\ (K^T K + \lambda I)^{-1} K^T Y & \text{K is not full-ranked} \end{cases}
\]  

Given a new input \(x^*\), the class label can be obtained by applying a nearest neighbor classifier to the output label vector \(y^*\) of the learned kernel regression model.

2.3. Kernel regression for pose estimation

The pose of a space object is usually denoted by Euler angles, including yaw angle, pitch angle, and roll angle. It is clear that images of one object with different poses lie on a low-dimensional manifold (view manifold) in the visual input space. For example, in the case of 1D pose variation, e.g., pose changes in the direction of yaw from 0° to 360°, the images will lie on a 1D closed manifold as shown in 3D Euclid space in Fig. 1.

This embedding result was obtained by locally linear embedding (LLE)\(^\text{LLE}\) using images of one satellite in the 1D subset of BUAA-SID 1.5 introduced in Subsection 3.1. It is obvious that all such 1D closed manifolds of different objects are topologically equivalent to a unit circle in 2D Euclid space, i.e., a normalized 1-sphere. Similar things happen in the cases of 2D and 3D pose variations. Therefore, we can separately use a normalized 1-sphere in 2D Euclid space to represent 1D pose variation of a space object, a normalized 2-sphere in 3D Euclid space to represent 2D pose variation, and a normalized 3-sphere in 4D Euclid space to represent 3D pose variation. By this way, the output of the kernel regression model for a pose estimation task will be constrained to the conceptual geometry of the pose.

Let the pose of input \(x_i\) be

\[
p = \begin{bmatrix} \cos \theta_i, \sin \theta_i \\
\cos \phi_i, \sin \phi_i, \sin \psi_i \\
\sin \phi_i, \cos \psi_i \end{bmatrix}^T
\]  

where \(\theta_i, \phi_i, \text{ and } \psi_i\) are the yaw angle, pitch angle, and roll angle, respectively. For clarity and without loss of generality, we consider the case of 1D pose variation in this section. Then, the kernel regression model for pose estimation can be

\[
p = g(x) = \kappa(x) B
\]  

where the coefficient matrix \(B \in \mathbf{R}^{M \times 2}\) is

\[
B = \begin{cases} K^{-1} P & \text{K is full-ranked} \\ (K^T K + \lambda I)^{-1} K^T P & \text{K is not full-ranked} \end{cases}
\]  

where

\[
P = [p_1, p_2, \cdots, p_M]^T \in \mathbf{R}^{M \times 2}
\]  

Given a new input \(x^*\), the estimated pose \(p^*\), i.e., the output of the kernel regression model in Eq. (16), is

\[
p^* = g(x^*) = \kappa(x^*) B = [p_1^*, p_2^*]
\]
where \( p_1 \) and \( p_2 \) are the first and second dimensions of \( p^* \). Then, the estimated yaw angle \( \theta' \) can be calculated by the inverse tangent function

\[
\theta' = \arctan \frac{p_2}{p_1}
\]

Similar approaches can be achieved in the cases of 2D and 3D pose variations.

### 2.4. Joint framework for object recognition and pose estimation

To jointly recognize space objects and estimate their poses using a kernel regression model in one framework, we can use the combined formulation as the following according to Eqs. (11) and (16), i.e.,

\[
[y, p] = \kappa(x) [A, B]
\]

where the combined coefficient matrix \([A, B] \in \mathbb{R}^{M \times (C + 2)}\) can be solved as

\[
[A, B] = \begin{cases} 
K^{-1} [Y, P] & \text{if } K \text{ is full-ranked} \\
(K^T K + \lambda I)^{-1} K^T [Y, P] & \text{if } K \text{ is not full-ranked}
\end{cases} 
\]

where \( Y \) and \( P \) are defined in Eqs. (13) and (18), respectively. When the combined output \([y', p']\) of a given input \( x^* \) is obtained from the learned combined model in Eq. (21), we can solve for class label and Euler angles based on \( y' \) and \( p' \) using the methods mentioned above.

### 3. Experiments

#### 3.1. BUAA-SID dataset

BUAA-SID 1.0\(^{21}\) is a publicly available satellite image dataset, containing 4600 gray images of 20 satellites from 230 viewpoints sampled on a viewing sphere, together with 4600 corresponding binary images. This dataset is suitable for testing the performance of multi-view space object recognition in the case of 2D pose variation, and has been used in Refs.\(^{11,12}\). However, there is no change of lighting conditions in BUAA-SID 1.0, and the 230 viewpoints are not sampled densely enough to better test continuous pose estimation performance. Although there have been other datasets which contain dense view samples, such as Multi-View Cars dataset\(^{22}\) and RGB-D Object dataset\(^{23}\), they can only be used for pose estimation of general objects, not space objects. Therefore, we selected the 3D models of the first ten satellites in BUAA-SID 1.0, simulated images using the method in Ref.\(^{24}\), and built a new simulated satellite image dataset named BUAA-SID 1.5. BUAA-SID 1.5 contains four subsets. The first subset, the 1D subset, includes 3600 grayscale images of ten satellites captured from 360 viewpoints uniformly sampled on a circle with the pitch angle \( \varphi = 0 \) and the yaw angle \( \theta \in [0, 2\pi] \).

The second subset is the 2D subset, which includes 2042 grayscale images of one satellite captured from 2042 viewpoints on a view sphere with the pitch angle \( \varphi = [-0.5\pi, 0.5\pi] \) and the yaw angle \( \theta \in [0, 2\pi] \). The third subset is the lighting subset. This subset contains 10080 gray images of one satellite from the same viewpoints as the 1D subset, which were simulated in different lighting conditions, i.e., the phase angle of the light ranging from 0° to 90° in steps of 10° while the altitude angle of the light in the range of 0°, 90°, and 180° separately. Images in the last subset, i.e., the noise subset, were obtained by adding Gaussian white noise to images in the 1D subset, with variance from 0.001 to 0.01 in steps of 0.001. We can perform a better validation for our methods on BUAA-SID 1.5 for both multi-view space object recognition and pose estimation. Fig. 2 shows some sample images in BUAA-SID 1.5, one gray image for each satellite together with its corresponding binary image.

#### 3.2. Representation of space objects

How to represent space objects using their multi-view images is very important for space object recognition and pose estimation. The representation should be discriminative and pose-sensitive, so as to achieve good recognition and pose estimation performance. Considering the characteristics of space objects, we use two kinds of features for representation. One kind is shape representations, including moment invariants (MI),\(^{25}\) Fourier descriptors (FD),\(^{26}\) and distance transform (DT).\(^{8}\) They are computed based on binary images to capture the geometries of space objects. Binary images (BI) themselves are also used as a shape representation. The other kind is appearance representations, including gray images (GI), and a recently popular local descriptor histogram of oriented gradients (HOG)\(^{32}\), to capture the appearance variations of space objects when their viewpoints or poses change. In this paper, we do not describe the details of the implementation.

![Fig. 2 Sample images in the BUAA-SID 1.5 dataset.](image)

(a) BI  (b) DT  (c) GI  (d) HOG

![Fig. 3 Visualization of image representations.](image)
of MI, FD, DT, or HOG, as they can be easily found in the references. We only give visualization results of BI, DT, GI, and HOG in Fig. 3. MI and FD cannot be shown as an image like HOG in Fig. 3. Because HOG was arranged by following the location of the region in the image where it was computed, it can recover the 2D location information to be shown in 2D space, but MI and FD are just 1D vectors without any 2D information.

3.3. Choice of kernels

There are different kinds of kernels that can be chosen for the kernel regression model, typically including Gaussian kernel (Eq. (9)), linear kernel, sigmoid kernel, etc. The linear kernel is

$$k(x_i, x_j) = x_i^T x_j$$

and the sigmoid kernel is

$$k(x_i, x_j) = \tanh(s x_i^T x_j + c)$$

(23)

(24)

where $\tanh(\cdot)$ is the hyperbolic tangent function, and $s$ and $c$ are parameters. We tested these three kernels in a joint space object recognition and pose estimation task using GI representation, with the same experimental setting as in Subsection 3.6.1. Results are shown in Table 1, where MAE is the mean absolute error (see Subsection 3.5.1 for details). The parameter of the Gaussian kernel is set as Eq. (10), and the parameters of the sigmoid kernel are set as

$$s = \frac{0.01}{\max_{i,j \in \{1, 2, \ldots, M\}} x_i^T x_j}, \quad c = 0$$

(25)

From Table 1, we can see that all the three kernels can achieve a 100% recognition accuracy, but the Gaussian kernel performs the best in pose estimation. Therefore, we choose the Gaussian kernel in the following experiments.

3.4. Space object recognition

To test the space object recognition performance of our proposed method, we did experiments on BUAA-SID 1.0, and followed the same data splitting as that in Refs.11,12 for a fair comparison. Results are shown in Tables 2 and 3. For image representation, we separately used 12-dimensional moment invariants, 20-dimensional Fourier descriptors, and a 324-dimensional histogram of oriented gradients feature computed in 6-by-6 windows per bounding box with 9 histogram bins.

In Table 2, we used half images of the whole dataset for training and the rest for testing, the same as in Ref.11. Comparing with the results in Ref.11, we can improve the recognition accuracy from 95.87% to 98.00%, and reach a very high level of recognition performance on the multi-view dataset with 2D pose variation. In Table 3, we randomly selected 80, 90, and 100 images from each satellite for training, as in Ref.12, and tested the rest. We reported the average recognition accuracy of 10 rounds of experiments. From Table 3, we can see that our method significantly outperforms the space object recognition method proposed in Ref.12, regardless of the number of training images.

It can also be seen that appearance representation HOG outperforms shape representations MI and FD. This is because appearance representation contains more information of space objects than shape representations which only cover the geometric information of space objects. Actually, the methods described in Refs.11,12 for space object recognition are feature projection or feature extraction approaches, representing the images of space objects from one feature space to another so as to improve the recognition performance of traditional classifiers, such as k-nearest neighbors or support vector machines (SVM). However, our kernel regression-based method is a recognition model, and can accept any kind of image representations as inputs, so the recognition accuracy may be further improved if we use the output of the feature projection or feature extraction approaches, such as KLPP in Ref.11 and elastic net sparse coding in Ref.12, as the input of our model.

3.5. Space object pose estimation

3.5.1. 1D and 2D pose estimation on BUAA-SID 1.5

We performed pose estimation experiments on the 1D and 2D subsets of BUAA-SID 1.5 using all the six image representations mentioned in Subsection 3.2. Half of the images in the 1D and 2D subsets were used to train the kernel regression model, and the rest were for testing, i.e., we trained our model using one of every two images in the dataset and tested the rest. Ten different models were trained, one for each satellite in the 1D subset. The results in Table 4 are the average performance of ten satellites in the 1D subset, and 2D pose estimation results on the 2D subset are shown in Table 5.
For quantitative evaluation, we define absolute error (AE) as the absolute value of the difference between the estimated pose angle and the ground truth angle. We report the percentage of test images of which the pose angle is correctly estimated with an AE less than a threshold (1°, 2°, or 5°) to describe the distribution of angle errors, and we also report the MAE on the entire testing set to indicate the average performance of pose estimation.

From the results in Tables 4 and 5, we can see that gray image outperforms other image representations as it is more sensitive to pose variation. The accuracy of 2D pose estimation is lower than that of 1D case. This is reasonable, considering the increasing difficulty of the pose estimation problem in 2D case than that in 1D. In terms of AE < 5°, our method can achieve a more than 80% accuracy using 4 image representations in 1D case, about 80% using HOG in 2D case, and about 98% using GI in 2D case as well. It means that we can provide an estimated pose angle with an error less than 5° for 80% testing images, and such results are useful to accurately estimate the pose of space objects in practical application of space systems.

3.5.2. Robustness against noise

We tested our method on the noise subset of BUAA-SID 1.5 to evaluate the robustness against noise. Practically, we trained our model using the training set in Subsection 3.5.1, and tested it using the images in the noise subset corresponding to the testing set. Results are shown in Fig. 4. As the image represen-
tations are sensitive to noise, the pose estimation performance decreases when the variance of Gaussian noise increases. Gray image performs the best among the six image representations, and can provide acceptable results (about 80% in terms of AE < 5°, with an MAE less than 5°) even if the image has strong noise (with a variance of 0.01).

### 3.5.3. Robustness against lighting

As image representations based on gray images, such as GI and HOG, are sensitive to lighting conditions, we tested our method on the lighting subset of BUAA-SID 1.5, in order to validate the robustness against lighting conditions. Fig. 5 shows the experimental results. It is shown that the mean absolute error of GI is getting worse rapidly when the lighting phase angle is more than 30°, while the performance of HOG changes slowly when the lighting phase angle is less than 70°. HOG is more robust than GI when the lighting condition changes, although GI performs better when the lighting phase angle is less than 10°, i.e., in nearly ideal lighting condition. Our model trained using HOG representation can achieve an MAE less than 5° and a more than 70% accuracy in terms of AE < 5° even if the light phase angle is large (50°).

### 3.6. Space object recognition and pose estimation

#### 3.6.1. Joint recognition and pose estimation for space objects on BUAA-SID 1.5

We did experiments on the 1D subset of BUAA-SID 1.5 to evaluate the performance of our method for jointly solving space object recognition and pose estimation. Dataset splitting was the same as in Subsection 3.5.1, i.e., half of all the images for training and the rest for testing. Results are shown in Table 6. It should be noted that the MAE in Table 6 is calculated only on correctly recognized testing images, so the pose estimation results are worse than those in Table 4. Another reason is that in Table 4, the kernel regression model for pose estimation was trained separately for each satellite, while in Table 6, only one model was trained for all satellites in the dataset. The model for pose estimation may be under-fitted, because there may be images of different satellites with the same pose angle. From Table 6, we can see that our model can obtain outstanding recognition accuracy together with acceptable pose estimation performance. The image representations of MI and FD can also achieve successful recognition, with accuracies more than 94%, although the pose estimation results are terrible.

#### 3.6.2. Robustness against noise

To evaluate the robustness against noise, we experimented on the noise subset of BUAA-SID 1.5, by testing the trained kernel regression model in Subsection 3.6.1 using the images in the noise subset corresponding to the testing set. Fig. 6 shows the experimental results. We can see that the performance of pose estimation is affected significantly by noise, while the recognition performances of our model trained using BI, DT, and GI as image representations are more robust against noise than those using MI, FD, and HOG.

#### 3.6.3. Robustness against lighting

Experimental results on the lighting subset of BUAA-SID 1.5 are illustrated in Fig. 7. We recognized the satellite in the lighting subset and estimated its poses using the model trained in Subsection 3.6.1 for ten satellites. Fig. 7 shows that pose estimation performance is acceptable when the lighting phase angle is less than 30°, with an MAE less than 10°. The robustness of our model for space object recognition is very strong even if the lighting condition is extremely bad, i.e., the lighting phase angle is more than 80°. HOG can even perform nearly

---

**Table 6** Space object recognition and pose estimation results on the 1D subset of BUAA-SID 1.5.

<table>
<thead>
<tr>
<th>Image representation</th>
<th>Recognition accuracy (%)</th>
<th>MAE (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BI</td>
<td>100</td>
<td>6.80</td>
</tr>
<tr>
<td>DT</td>
<td>100</td>
<td>10.47</td>
</tr>
<tr>
<td>MI</td>
<td>94.00</td>
<td>84.65</td>
</tr>
<tr>
<td>FD</td>
<td>94.56</td>
<td>58.34</td>
</tr>
<tr>
<td>GI</td>
<td>100</td>
<td>0.43</td>
</tr>
<tr>
<td>HOG</td>
<td>100</td>
<td>6.47</td>
</tr>
</tbody>
</table>

---

Fig. 5 Pose estimation results on the lighting subset of BUAA-SID 1.5.
100% when the lighting phase angle is 80° and about 70% when the lighting phase angle is 90°.

4. Conclusions

In this paper, we proposed a kernel regression-based method for jointly recognizing multi-view space objects and estimating their poses. We performed space object recognition experiments on BUAA-SID 1.0, and obtained better recognition results than the state-of-the-arts. Besides, we built a new simulated satellite image dataset based on BUAA-SID 1.0, named BUAA-SID 1.5, to evaluate the proposed method. Experimental results on BUAA-SID 1.5 validated the effectiveness and robustness of our method. In future work, the more challenging problem of space object recognition and pose estimation in the case of 3D pose variation should be considered, as well as a better combined or projected image representation.

Acknowledgements

The authors thank the anonymous reviewers for their critical and constructive review of the manuscript. This study was co-supported by the National Natural Science Foundation of China (Grant Nos. 61371134, 61071137) and the National Basic Research Program of China (No. 2010CB327900).

References

Multi-view space object recognition and pose estimation based on kernel regression