Hindawi
International Journal of Aerospace Engineering
Volume 2023, Article ID 9631895, 13 pages
https://doi.org/10.1155/2023/9631895

Research Article

Flow-Based 6D Pose Tracking of Uncooperative Spacecrafts

Yu Su, Zexu Zhang, Mengmeng Yuan, and Yishi Wang

School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Zexu Zhang; zexuzhang@hit.edu.cn

Received 2 August 2022; Revised 29 June 2023; Accepted 10 October 2023; Published 22 November 2023

Academic Editor: Guillermo Valencia-Palomo

In this work, an optical-flow-based pose tracking method with long short-term memory for known uncooperative spacecraft is proposed. In combination with the segmentation network, we constrain the optical flow area of the target to cope with harsh lighting conditions and highly textured background. With the introduction of long short-term memory structure, the proposed method can maintain a robust and accurate tracking performance even in a long-term sequence of images. In our experiments, the pose tracking effects in the synthetic images as well as the SwissCube dataset images are tested, respectively. By comparing with the state-of-the-art pose tracking frameworks, we demonstrate the performance of our method and in particular the improvements under complex environments.

1. Introduction

Known uncooperative spacecraft 6D pose tracking is crucial in on-orbit operations, e.g., docking, rendezvous, servicing, and space debris removal [1]. These rely on precise and robust estimation of the relative pose under harsh lighting conditions and against highly textured background [2]. Considering the size, mass, and power, monocular sensors can ensure rapid pose determination for noncooperative target with lower power, lower hardware complexity and cost, and mass requirements.

In the method of traditional but state-of-the-art monocular pose determination for spaceborne applications, the features in images are first extracted (e.g., edges, corners, and lines). Through the matching between the features of the images, they establish the correspondences between the 2D pixels and the initial 3D model and then calculate the relative pose of the target. In order to maintain high overlap, matching is often performed on adjacent images, which could obtain more accurate correspondences. This process of pose tracking cannot avoid the accumulation of errors, which will even lead to the failure of the tracking mission.

As objects are more than just a collection of edges and geometric primitives, for some satellites whose models are known or have been roughly reconstructed, traditional vision methods cannot effectively use the information of the model, while convolutional neural networks (CNNs) can learn more complex and meaningful features to the task at hand while ignoring background features (e.g. clouds) based on context. Over the past decade, nearly all computer vision tasks have become increasingly dominated by CNNs [3–6]. Compared to prior techniques, CNNs have been shown to be more resilient to noise and better able to generalize to previously unseen scenarios. The 2019 Satellite Pose Estimation Challenge (SPEC) [7], hosted by Stanford University and the European Space Agency (ESA), saw all of its top-performing submissions employ CNN-based deep learning models.

Not only that, unlike imagery captured for terrestrial applications, space imagery is characterized by high contrast, low signal-to-noise ratio, and low sensor resolution. In the case of a known model or a relatively complete point cloud obtained in advance, using CNNs has been verified to have more accurate correspondences [8]. So far, after establishing correspondences between the object's 3D model and 2D pixel locations, the 6D pose is usually calculated by perspective-n-point (PnP) algorithm based on RANSAC [9, 10]. The method based on single-frame image extraction of correspondence has been proven to have a good estimation result [11–14].

For space target image sequence under harsh lighting conditions and serious background interference, some images are difficult to extract features correctly, which will

cause mistakes in pose estimation based on single frame image. At this point, we should effectively consider the time domain information of the image sequence and establish appropriate features [15]. Wen et al. [16] proposed a novel neural network architecture that could keep long-term track of an object's pose robustly and efficiently with RGB-D video sequence. To predict the relative pose between the current observation and the synthetic model rendering at the previous prediction, the method cleverly establishes the connection between interframes and makes the tracking effect more stable.

As the motion of the space target often has a certain regularity, long sequence images can achieve better analysis of target motion. The research in [17] introduced a keyframe memory pool to store the most informative historical observations. This not only greatly reduces the tracking drift problem but also effectively addressed the noisy segmentation and external occlusions from the interaction.

However, most space images do not have depth maps, and the depth range of the target is very large. Inspired by above methods, on the basis of utilizing interframe information, we could also input a set of image sequences to the network. Through a long-term memory cell, the motion law of the target could be analyzed, and the pose tracking will be more stable and accurate.

Therefore, in this article, a 6D pose tracking network for known uncooperative spacecraft image sequence is constructed, which takes the target 2D interframe transformation of pixels as input and 3D pose transformation as output. The main contributions this work makes are as follows:

(1) The target optical flow is used as the description of the 2D pixel transformation between frames. The segmentation network is introduced to constrain the target optical flow area and reduce the cumulative errors caused by the target pixel offset during the pose tracking process

(2) A long-term memory cell of LSTM network is used to learn the movement characteristics of the space target. By inputting a sequence of target optical flow, a robust and precise tracking network is established

2. Related Work

This paper focuses on the 6D pose tracking based on interframe relevance. Over the years, many methods have been proposed to describe the transformation between adjacent frames. The optical flow has been widely used in computer vision fields owing to its better tracking effect at small angle changes. Different from traditional feature point matching, optical flow can obtain more accurate interframe pixel changes to determine the correspondence, which could effectively utilize interframe information and obtain higher matching accuracy.

Among these flow-based tracking methods, sparse optical flow is widely used in many applications due to its high efficiency and accurate calculation method [18]. For space images, considering the complex lighting conditions and highly textured background, partial overexposure or overdarkness happens from time to time. In order not to rely too much on a small number of feature points, as an image registration method for image point-by-point matching, dense optical flow is introduced to calculate the offset of all points on the image to form a dense optical flow field.

Although dense optical flow is proven to be robust to rapid and irregular motion, since the optical flow of each pixel participates in the calculation, the computational burden obviously increases, and Brox et al. [19] proposed a combination of contour matching and optical flow. The purpose of this method is to use staggered contour matching to reduce the accumulation of tracking errors and optimize the tracking effect. This method proves that the integration of two constraints cannot only improve the calculation efficiency but also avoid the cumulative shift in the pixel tracking process.

In order to establish the constraints on the target area and focus on the target optical flow, we first segment the image independently. Encouraged by the successful application of deep learning technology in object classification and recognition [20–23], various researchers have made attempts to explore to solve the semantic segmentation problem of image pixel-level labels. Since the target model is known, the fully convolutional networks (FCN) [24] could be used to segment the image at the pixel level and obtain a higher-precision mask. The FCN network replaces the fully connected layer in the convolutional neural network with the convolutional layer, so that each pixel generates a prediction value to output the segmented image instead of the classification score and restores the feature map to the original image size through the deconvolution layer. Due to the unity of space background types, the method is more efficient and accurate, and the two-dimensional pixels of the independent segmentation can also prevent errors caused by accumulated offsets.

Although the combination of dense optical flow and segmentation network solves the problem of the transformation of two-dimensional pixels, we still need to make full use of the regularity of the target movement in the image sequence. Without introducing other sensors besides the camera, some optimization algorithms could be introduced to analyze the movement pattern of the target. Shantaiya et al. [25] combine optical flow and the Kalman filter to track the pose of the video dataset. This method can reduce the cumulative error in the tracking process when the target movement has certain regularity. In robotics, the time information in video data is also very important for the optimization of pose estimation and plays an important role in tasks such as route planning and active sensing [26–30].

With the development of deep learning, researchers have begun to consider the use of networks to solve continuous and multidimensional data prediction tasks. Wang et al. [31] proposed a method based on recurrent neural network (RNN) to estimate the depth and camera pose of a multiview monocular video sequence. By solving the gradient disappearance problem of RNN, the long short-term memory network (LSTM) has become one of the most advanced networks for processing time series-related data, which has been applied in fields of speech recognition [32], image

and caption generation [33, 34], and multidimensional image processing [35]. By combining input gate, forget gate, and output gate, LSTM cannot only analyze the characteristics of adjacent frames in a short period of time but also learn the motion laws of targets through long-term memory units, optimize tracking performance using time-domain information, and perform well in the long-term tracking process. Many technologies also introduce LSTM to pose-related tasks, such as motion tracking and motion recognition [36–40]. Through the long-term memory cell, target transformation in the entire image sequence can be used for estimation.

However, for image tracking, only LSTM is not enough. We also need to convert the image into features for storage and analysis to achieve better results. Therefore, in this article, the target optical flow area is first constrained based on the FCN target segmentation results, and the convolutional long short-term memory (ConvLSTM) architecture is introduced to combine with our pose estimation network, thus effectively utilizing the time domain information of the image sequence. On this basis, the cumulative errors generated in the tracking process can be also reduced.

3. Approach

Given a sequence of input images, our goal is to estimate their 3D rotation and translation. We assume that the target is rigid and that the 3D model is available. In this section, we use the image information of a sequence to calculate the dense optical flow between images and obtain the optical flow information of the target area using the mask obtained by the FCN network. Optical flow information describes the relative movement of the target between image frames. In order to more accurately estimate the 6D pose of the target, we need to add the target's two-dimensional and three-dimensional coordinates into the input of the network which needs to enrich the structural information.

Therefore, we use a two-dimensional matrix composed of the initial position of the target optical flow (that is, the 2D target object pixel points of the previous frame), the corresponding optical flow value, and the 3D points of the corresponding pose as the input of the network. Figure 1 depicts the overall architecture for target optical flow pose tracking. In the remainder of this section, we describe each stream in detail.

3.1. Dense Flow. Optical flow contains rich motion information, so it can be used to predict the direction and speed of object movement. When using dense sampling, it is necessary to calculate the dense optical flow for each pixel in the image. Dense optical flow can reduce the uncertainty caused by feature extraction compared with sparse optical flow.

As shown in Figure 2, given two images, small nonoverlapping patches of 4×4 pixels are extracted from the first image, and the patch in the first image is used to convolve with the patch in the second image to get the response map. After the calculation of this first-level response mapping, we use sparse convolution on the response map to calculate the response map of the larger patch. This procedure produces a pyramid of response maps.

By using a max pooling operator, the response map is guaranteed to be the same as the response map when the patch moves by one pixel. Since the response map changes slowly in space, a downsampling step is introduced to reduce the complexity of the following steps. In order to prevent the response map from converging too fast during sparse convolution, nonlinear filtering is introduced, and the position of each subpatch is optimized around 3×3 pixels.

The response mapping pyramid is constructed using a bottom-up method, while the top-down method is used to extract correspondence. The local maximum value in the different layers of the response mapping pyramid can generate the correspondences between local image patches via matching. To obtain dense correspondences between matched patches (i.e., at local maxima), it suffices to recover the path of response values that generate this maximum. For each local maximum, each layer in the pyramid is retrieved from the response map to generate a quasi-dense correspondence.

According to the optical flow, the corresponding relationship of the 2D points between the two frames can be obtained. For each 2D point $\mathbf{u}_i$ in the previous frame, there will be a 2D point in the new frame corresponding to it. We have

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta\mathbf{u}_i. \tag{1}$$

We projected the 3D points of the target with a known pose in the previous frame to the image plane and calculated the dense optical flow between the two frames. On this basis, 2D-3D point correspondences can be established based on 3D points' new positions in the current frame, and the new image 6D pose can be predicted.

Assume there is a 3×3 camera intrinsic parameter matrix $\mathbf{K}$, the 2D pixel points $\mathbf{u}_i$ of the target in the image, and its corresponding 3D points $\mathbf{p}_i (1 \le i \le n)$; then, we have

$$\lambda[\mathbf{u}_i, 1]^{\mathrm{T}} = \mathbf{K}(\mathbf{R}\mathbf{p}_i + \mathbf{t}), \tag{2}$$

where λ_i is a scale factor and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector that define the camera pose, respectively. Since $\mathbf{R}$ is a rotation, which has three degrees of freedom, and $\mathbf{t}$ likewise, so $\mathbf{R}$ and $\mathbf{t}$ have a total of 6 degrees of freedom.

3.2. Target Segmentation. The dense optical flow calculated for the interframe image contains all the pixels, but we mainly focused on the optical flow information of the target in the image and obtained the 6D Δpose estimation information of the target. The fully convolutional neural (FCN) network can segment the image at the pixel level to acquire the target mask and then determine the optical flow of the target area. FCN uses the deconvolution layer to upsample the feature map of the last convolution layer and restore it to the same size of the input image, so that a prediction can be generated for each pixel while retaining the original input image information in the space. The convolutional network architecture adopted in this article is shown in Figure 3.

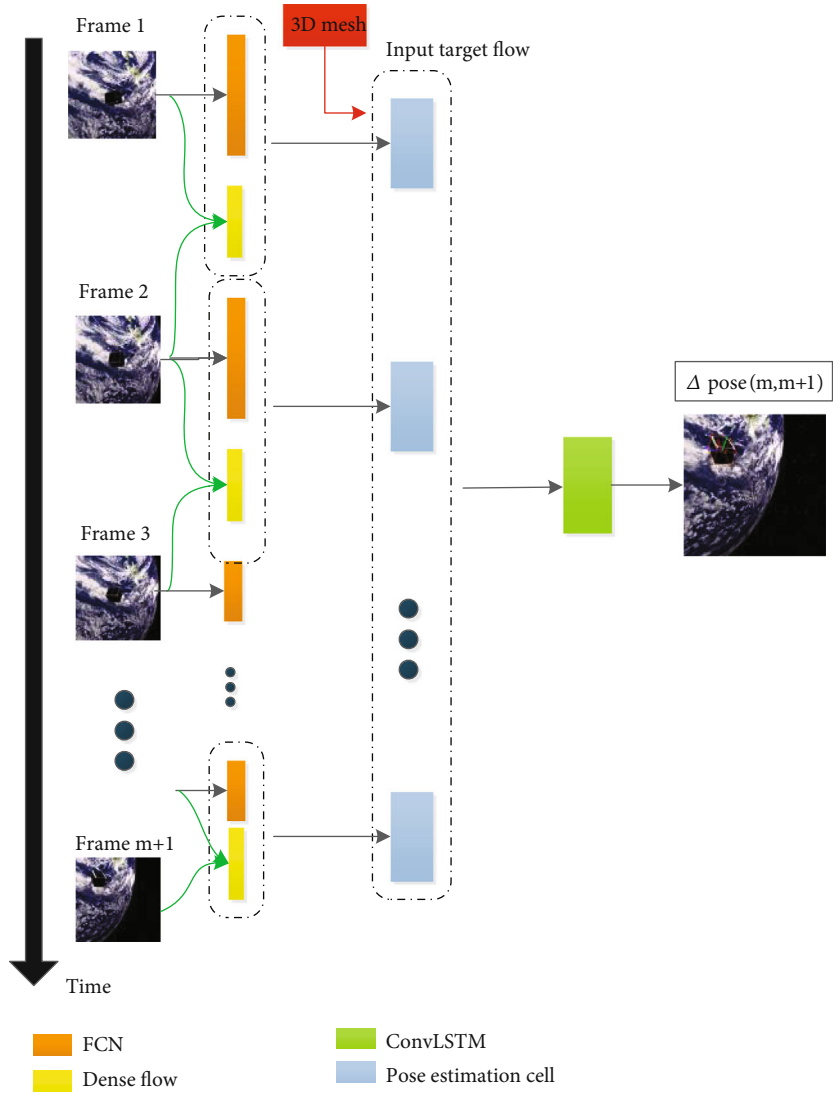

Figure 1: Overview of our framework.

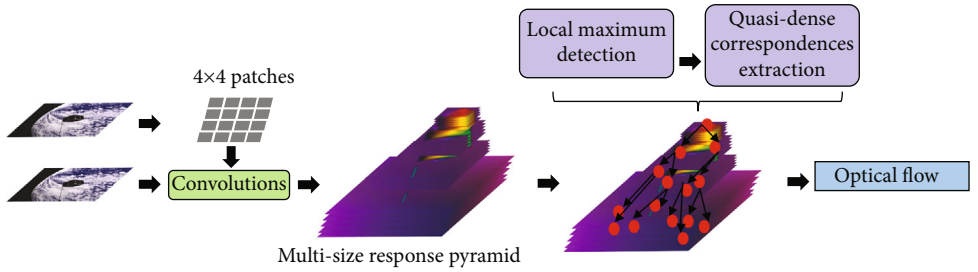

Figure 2: Outline of optical flow.

FCN uses the strip structure to superimpose the result of the deconvolution and the corresponding forward feature map, thereby obtaining more accurate pixel-level segmentation. By upsampling the low-resolution feature map with rich semantic information to the same size as the high-resolution feature map with rich edge information, and then adding them as the final semantic feature map, the robustness and accuracy can be ensured. We used FCN-8s to perform upsampling 3 times, fused the 3 deconvolution result images, and obtained the final prediction result.

By using the VGG model as the prenetwork for initialization, transfer learning and fine-tuning were conducted on the basis of pretraining. By overlaying the target mask with the image's dense flow field, we can obtain the dense optical flow of the target in the image.

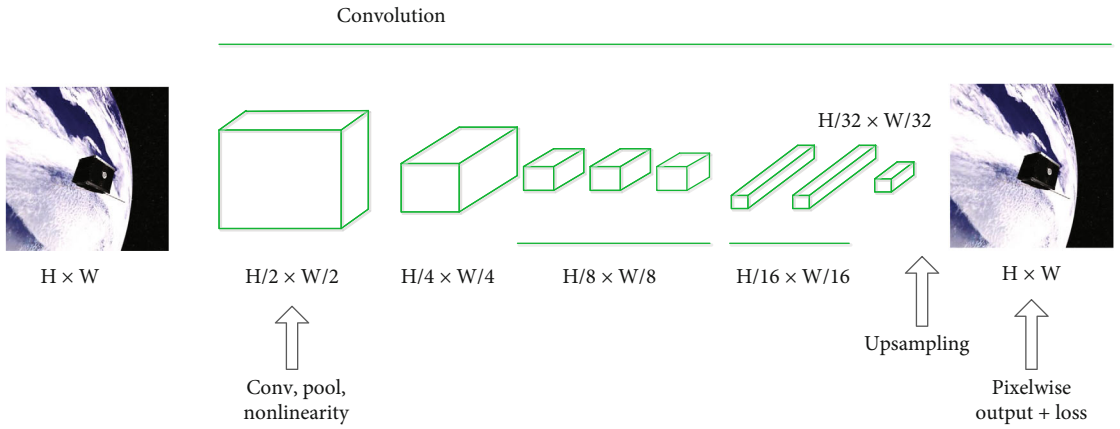

Figure 3: Segmentation network architecture.

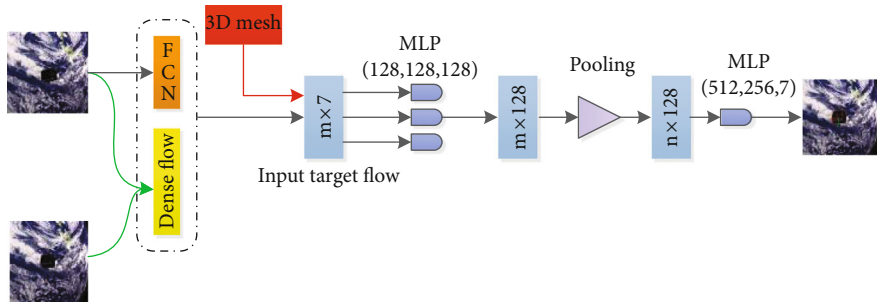

Figure 4: Pose estimation network architecture.

3.3. Flow-Based Pose Tracking

3.3.1. Pose Estimation Network. In addition to the optical flow, the initial two-dimensional points' position of the optical flow also contains the initial pose information of the target. In order to obtain the three-dimensional post change between frames of the target, we need to provide the initial pose information for the pose estimation network.

To this end, a simple network architecture was constructed to predict the pose from the target optical flow, as shown in Figure 4. It comprises three main modules [41], including a local feature extraction module with shared network parameters, a feature aggregation module, and a global inference module made of simple fully connected layers.

In the above network structure, an MLP with three layers was used to extract local features for the target optical flow, with weights shared across the target optical flow. A single max-pooling operation was carried out for aggregation so as to obtain a fixed dimension of the context representation and avoid bringing extra parameters. Finally, we aggregated the N-dimensional vector into the 6D pose output through another MLP. To this end, we adopted three fully connected layers and encoded the final pose as a quaternion and a translation.

The deep network described above gives us a differentiable approach to predict the 6D pose from target optical flow clusters for a given object. Given the target optical flow from FCN and dense flow, we need to establish the relationship between 2D and 3D transformation. To do so, another deep regressor f with parameters Θ was adopted.

$$(\Delta\mathbf{R}, \Delta\mathbf{t}) = g(f(\mathbf{p}_i, \mathbf{u}_{mi}, \Delta\mathbf{u}_{mi}), \Theta), 1 \le i \le n, \quad (3)$$

where $\Delta\mathbf{u}_{mi}$ is the target optical flow, $\mathbf{u}_{mi}$ is the 2D initial position of the target optical flow, $\mathbf{p}_i$ is the 3D point of the corresponding pose, $\Delta\mathbf{R}$ represents the posture change, and $\Delta\mathbf{t}$ represents the translation change.

3.3.2. Flow-Based Pose Tracking with ConvLSTM. As an extension of recurrent neural networks (RNN), LSTM introduces a cell to remember or forget information adaptively. The long-term memory unit of LSTM provides correlations of consecutive frames, and short-term memory is used to infer the current state. This can refine outputs to improve the current estimation and reduce accumulated errors over a long sequence. As shown in Figure 5, LSTM mainly includes input gate, forget gate, and output gate. After initializing the long-term memory cell and hidden cell, the input and hidden cell will enter the LSTM network together. Effective information will be processed through input gate and entered into long-term memory cell, while invalid information will be forgotten. Finally, a part of the output from the previous stage will become the input for the next stage.

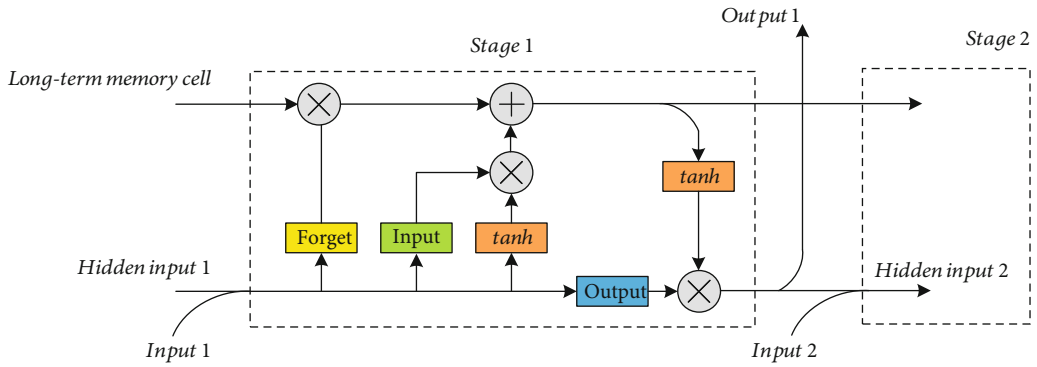

Figure 5: LSTM network architecture.

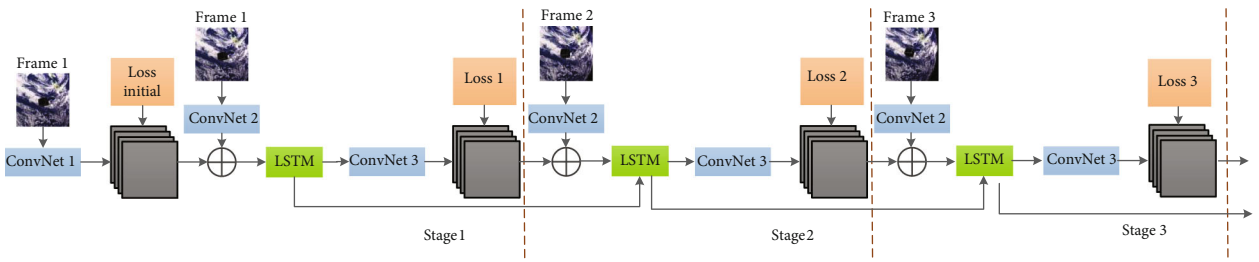

Figure 6: Target optical flow pose tracking with ConvLSTM.

ConvLSTM was developed based on LSTM, but a convolutional layer was added to better handle image spatial features. Its structure first convolves the input multidimensional matrix and then passes through the pose estimation unit constructed in the previous section. At the same time as outputting the results, a hidden unit is output. This hidden cell will form an input with the next frame of the image and enter the next pose estimation cell together. At the same time, each frame of the image will retain some information when entering the estimation cell, which is the long-term memory cell. This cell will be used to learn the motion laws of the target. Through continuous input of sequence data, the learning effect will gradually be optimized, ultimately affecting subsequent pose estimation.

As shown in Figure 6, assuming $\Delta\mathbf{T}$ represents the transformation of the pose, recurrent according to the stage, all the interframe data of 11 consecutive frames will be fed into the network to estimate the $\Delta\mathbf{T}$ from the 10th frame to the 11th frame. After obtaining the pose of the 11th frame, we used the images of frames 2-12 to estimate $\Delta\mathbf{T}$ of the 11th and 12th frames and so on.

To be more specific, a sequence of optical flow between frames will be used to predict the following 6D pose of a new frame. Given the input image, we took 11 images as a sequence to calculate the dense optical flow between frames. $\boldsymbol{I}_t$ represents the frame t.

$$\Delta\mathbf{u}_i^t = \mathbf{u}(\mathbf{F}(\mathbf{I}_{t-1}, \mathbf{I}_t)). \tag{4}$$

The dense optical flow $\mathbf{F}$ of the target area in the image was used as the input of the following network:

$$\Delta\mathbf{u}_{\text{mask}i}^{\ t} = \mathbf{u}(\mathbf{F}(\mathbf{I}_{t-1}, \mathbf{I}_t), \mathbf{M}_{t-1}). \tag{5}$$

The network model can be written as Equation (6).

$$\begin{aligned}(\Delta\mathbf{R}_t, \Delta\mathbf{t}_t) = g\big(&f_{t-10}\big(\mathbf{p}_i, \mathbf{u}_{\text{mask}i}^{\ t-10}, \Delta\mathbf{u}_{\text{mask}i}^{\ t-10}\big),\\ &\cdot f_{t-9}\big(\mathbf{p}_i, \mathbf{u}_{\text{mask}i}^{\ t-9}, \Delta\mathbf{u}_{\text{mask}i}^{\ t-9}\big), \cdots,\\ &\cdot f_t\big(\mathbf{p}_i, \mathbf{u}_{\text{mask}i}^{\ t}, \Delta\mathbf{u}_{\text{mask}i}^{\ t}\big), \Theta\big), 1 \le i \le n.\end{aligned} \tag{6}$$

To train it, we minimized the loss function as Equation (7).

$$L = \frac{1}{n}\sum_{i=1}^{n} \left\| \left(\bar{\mathbf{R}}\mathbf{p}_i + \bar{\mathbf{t}}\right) - \left(\mathbf{R}\mathbf{p}_i + \mathbf{t}\right) \right\|^2, \tag{7}$$

where $\bar{\mathbf{R}}$ and $\bar{\mathbf{t}}$ are the estimated rotation matrix and translation vector, respectively; $\mathbf{R}$ and $\mathbf{t}$ are the ground-truth ones. The rotations are calculated from the estimated and ground-truth quaternions, which can be carried out in a differentiable manner.

4. Experiments

Based on both synthetic data and real data from the dataset of SwissCube [42], the proposed flow-based pose tracking approach was compared with more traditional but state-of-the-art pose tracking frameworks and PnP-net [41].

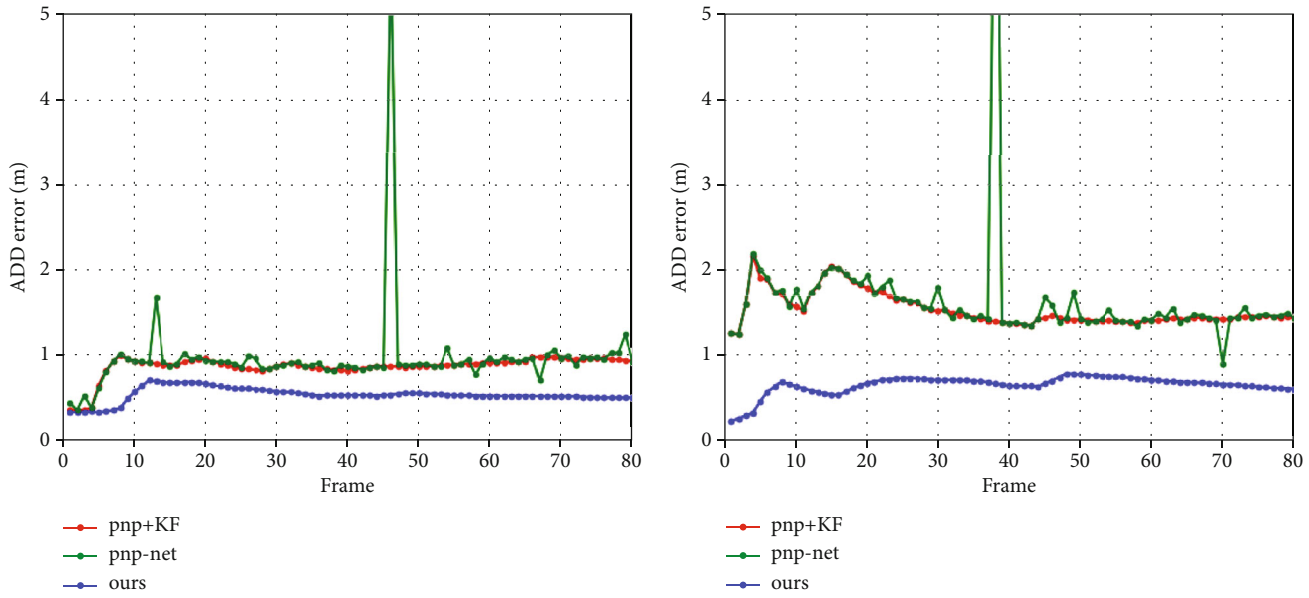

Figure 7: Comparison between proposed method with RANSAC PnP and PnP-net: (a, b) the add estimation error under 10% and 20% of outliers, respectively.

4.1. Metric. In terms of 3D error, the average distance error is converted by a 3D model based on the predicted pose and the real pose, respectively, which is called ADD error [43]. The pose accuracy in a sequence of image is calculated. In all test sets, the ADD-0.1d were reported, for which the predicted poses are considered to be correct if ADD error is smaller than 10% of the model diameter. In addition, the pose accuracy utilizes rotation and translation errors proposed in the literature [44].

$$\begin{cases} E_{\text{rot}}(\text{degrees}) = \max_{k=1}^{3} \left\{ \text{acos}\left(\mathbf{R}_{\text{true}}^{k} \times \mathbf{R}_{\text{est}}^{k}\right) \times \dfrac{180}{\pi} \right\}, \\ E_{\text{trans}}(m) = \dfrac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{t}_{\text{true}}^{i} - \mathbf{t}_{\text{est}}^{i} \right\|. \end{cases} \tag{8}$$

4.2. Synthetic Data. To simulate the sequence of optical flow, we first give the target an initial pose $\mathbf{R_0}$ and $\mathbf{t_0}$ and then made it transform regularly. With knowing the 3D mesh of the target and using a virtual calibrated camera (image size 1024×1024, focal length 90 mm), we projected the 3D mesh on the phase plane. After obtaining the 2D pixel coordinates, we simulated the optical flow by the transform of the target pixel coordinates. Then, the sequence of the optical flow will be given to LSTM as the input.

Recalling from Section 3.3, the network regresses pose from the target optical flow information and expects 2D inputs in the form $[x, y, z, u, v, \Delta u, \Delta v]$, where x, y, z represent the points cloud under the current frame image, u, v represent the starting position of the target optical flow, and $\Delta u, \Delta v$ represent the flow.

Table 1: Correct estimation of the proportion of synthetic test data: ADD $\leq 0.1d$, rotation error $\leq 3^{\circ}$, translation error $\leq 0.1d$; d represents the diameter of the model.

	ADD − 0.1d	R-3°	T − 0.1d
PnP+KF	77.2%	91.6%	80.5%
Ours	80.3%	94.7%	82.1%

Obviously, the target's two-dimensional pixels and their optical flow information are too much, so we took the resulting target optical flow from 1000 randomly sampled grid cells within the target mask. In order to simulate the solving error of optical flow, Gaussian error was added to Δu and Δv. At the same time, outliers were added on u, v (target 2D coordinates) in order to simulate the extraction error of FCN.

We trained our net for 350 epochs on 35 K synthetic training target optical flows with batch size of 32, and a learning rate of 0.0001 using the Adam optimizer. During training, we randomly added 2D noise with variance in the range of (0, 15) pixels and created 0% to 20% of outliers. We used 5 K sets of synthetic optical flow data and reported the mean pose accuracy in terms of the 3D space reconstruction error.

Combining a PnP algorithm with RANSAC is the most widespread approach to handle noisy correspondences [12]. The PnP-net pose estimation based on correspondence is the latest network alternative algorithm of the EPnP method. As shown in Figure 7, we compare the performance of these methods in a sequence of images. Moreover, we provide the accuracy of pose estimation on all test datasets, as shown in Table 1.

From the results of Figure 8, when the noise level is below 5%, the pixel coordinates of the target key points will

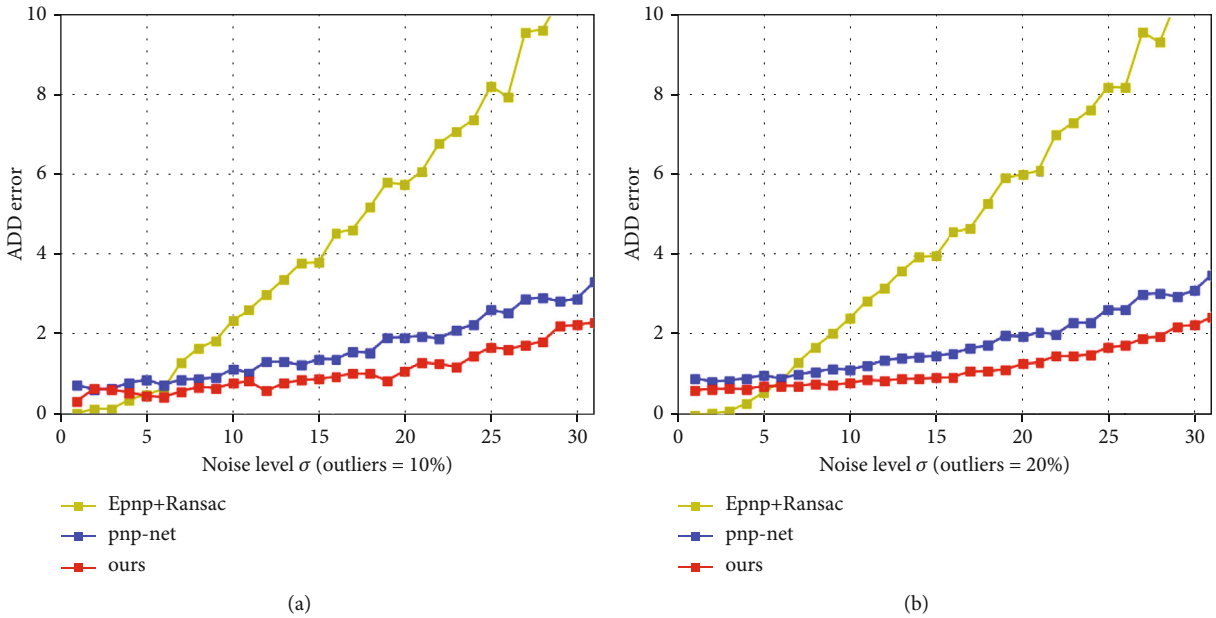

Figure 8: Comparison between proposed method with RANSAC PnP and PnP-net: (a, b) the add estimation error under 10% and 20% of outliers with the increase of noise level.

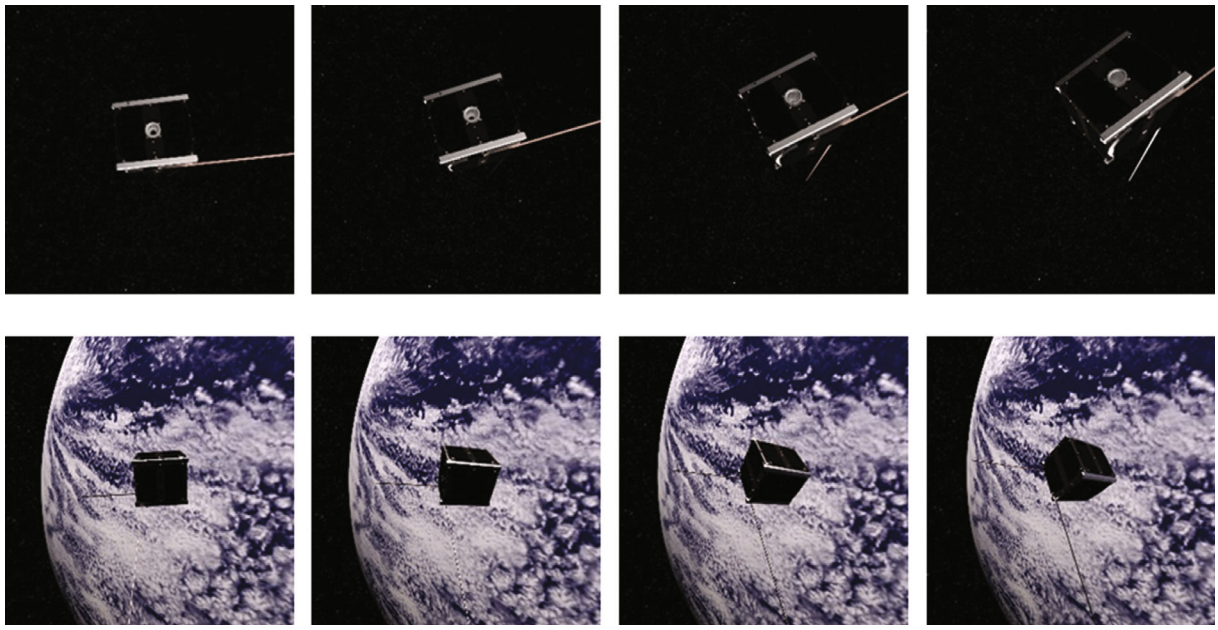

Figure 9: Original images: the first row is the original image sequence without background, and the second row is the original image sequence with the earth background.

be accurate. The precise pixel coordinates indicate that the transformation between the two sets of vectors is very consistent, and the matrix (rotation and translation) estimated by EPnP is more accurate. In fact, there is always a significant error in the pixel coordinates (i.e., 2D coordinates) of the extracted key points, and the level of positional noise interference is often higher than 10% or even 20%. At this point, PnP-net and the proposed method are much more accurate and robust to the increasing noise. But for different proportions of outliers, the network structure proposed in this article has better anti-interference ability. In a long-term sequence, the proposed method has a more stable tracking effect and better pose tracking accuracy.

4.3. Real Data. The proposed method was validated based on real data from a standard space target dataset. SwissCube dataset [42] is the newest dataset of space target, which comprise about 40 K real images from 100 video sequence, and parts of the original images are shown in Figure 9. The depth range is $1d$ to $20d$, where d indicates the diameter of the target without taking the antennas into account.

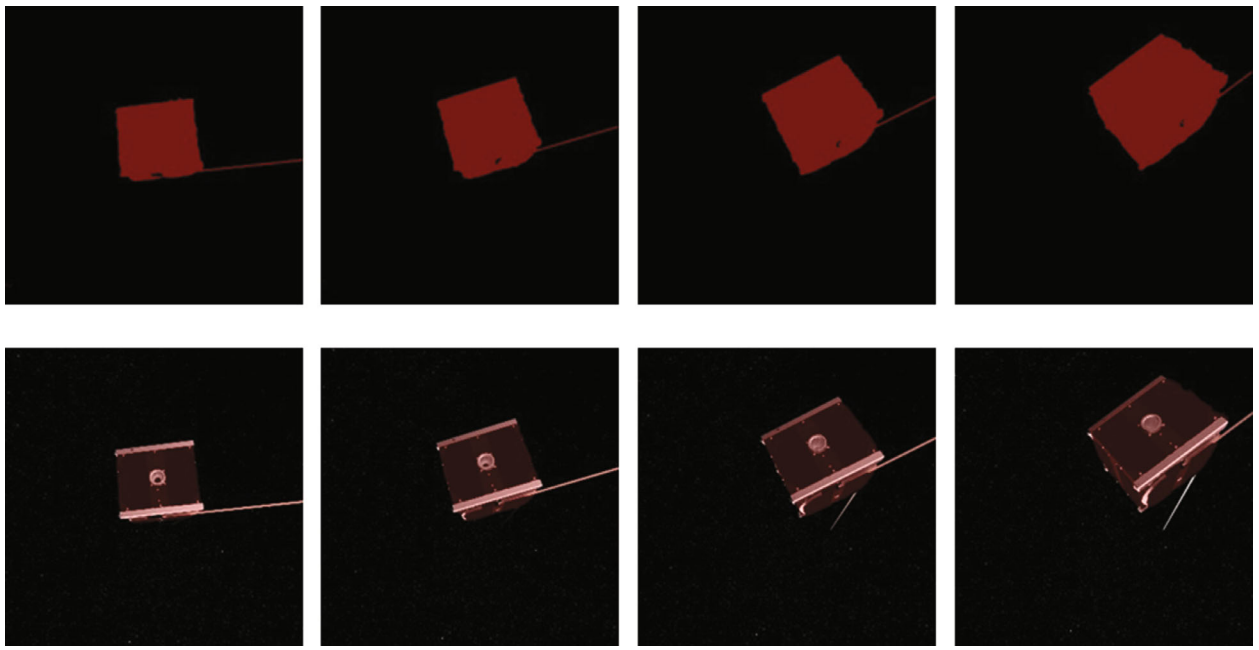

Figure 10: Target region segmentation (without earth background).

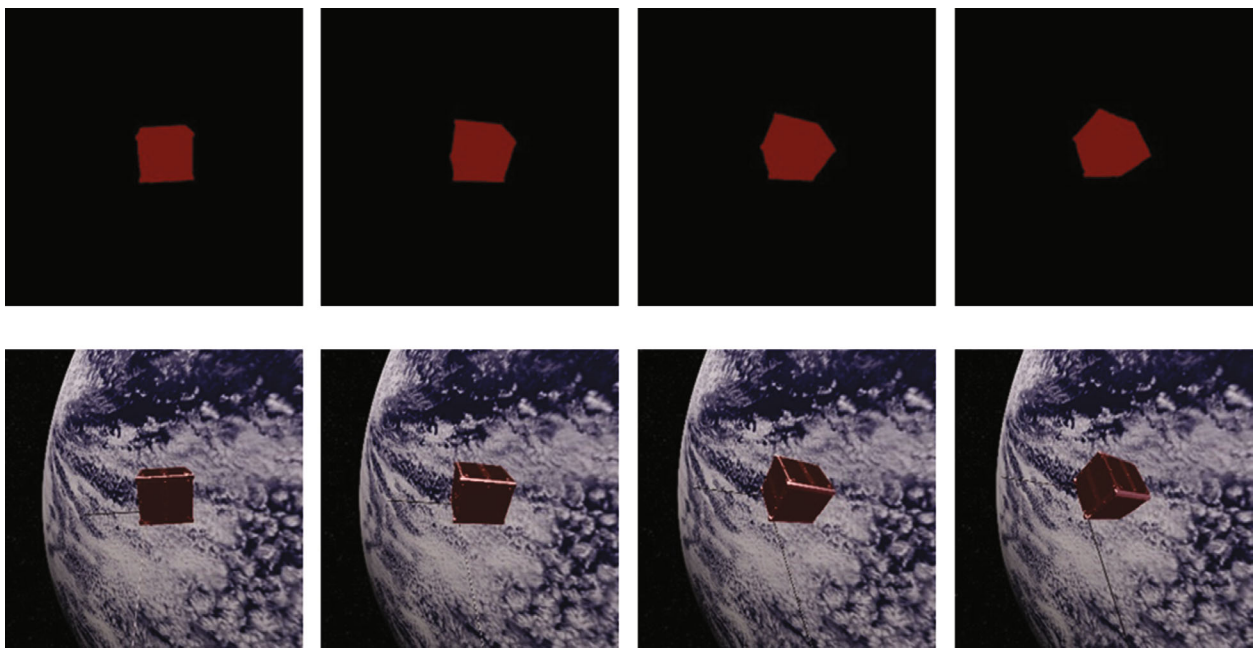

Figure 11: Target region segmentation (with earth background).

4.3.1. Training Procedure. By inputting all the images and training the FCN network with the known true value of the target mask image, a target mask segmentation network can be obtained. The training set contains 350 videos with a total of 350×100 frames of images, a total of 300 epochs are trained, and the number of samples in each epoch is 30,000.

In the first part of the network, we obtained the target area mask for the subsequent determination of the target optical flow. The extraction result for the SwissCube target area is shown in Figure 10 (no background) and Figure 11 (with background).

Assuming that we know the true poses of the initial 10 images, first, the mask map of the target in each image calculated by the network in part 1 was adopted to extract 1000 pixels on the target, and then the dense flow map was introduced to determine the target optical flow. The optical flow values of 1000 pixels were used to obtain the 3D mesh corresponding to these 1000 2D points based on the known real pose back-projection, forming a 1000×7 matrix, with 10 groups of optical flow including a total of $10 \times 1000 \times 7$ target optical flow sequences. As the input of the ConvLSTM network, the

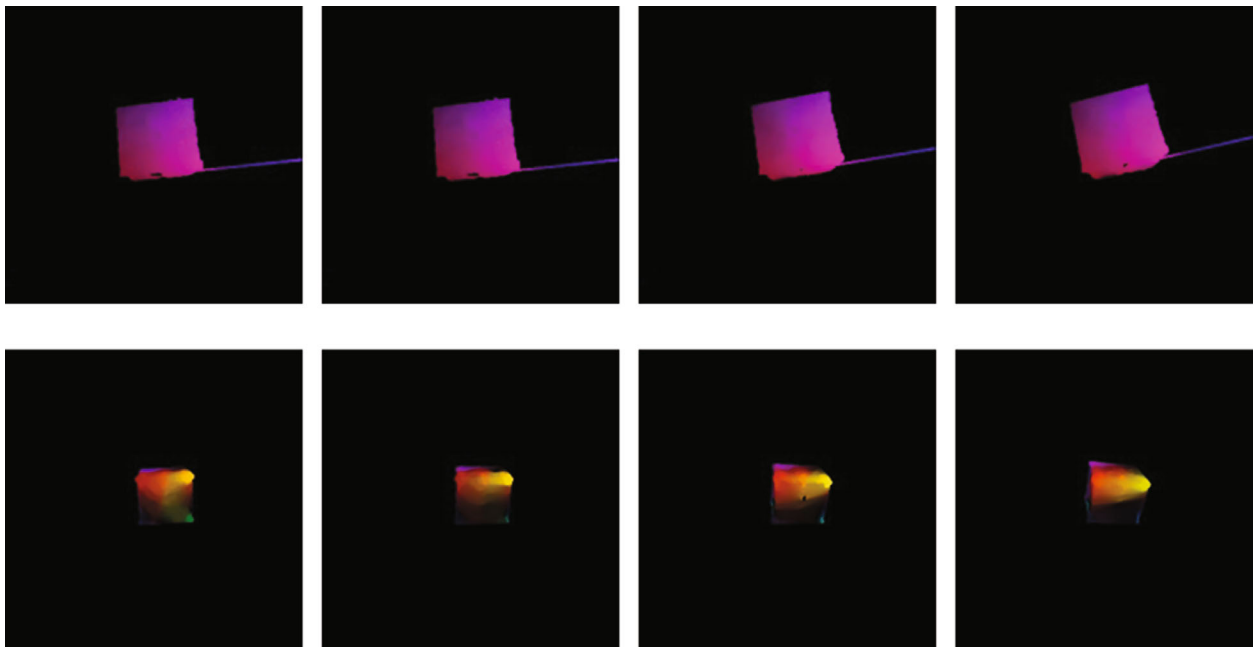

Figure 12: Target optical flow field.

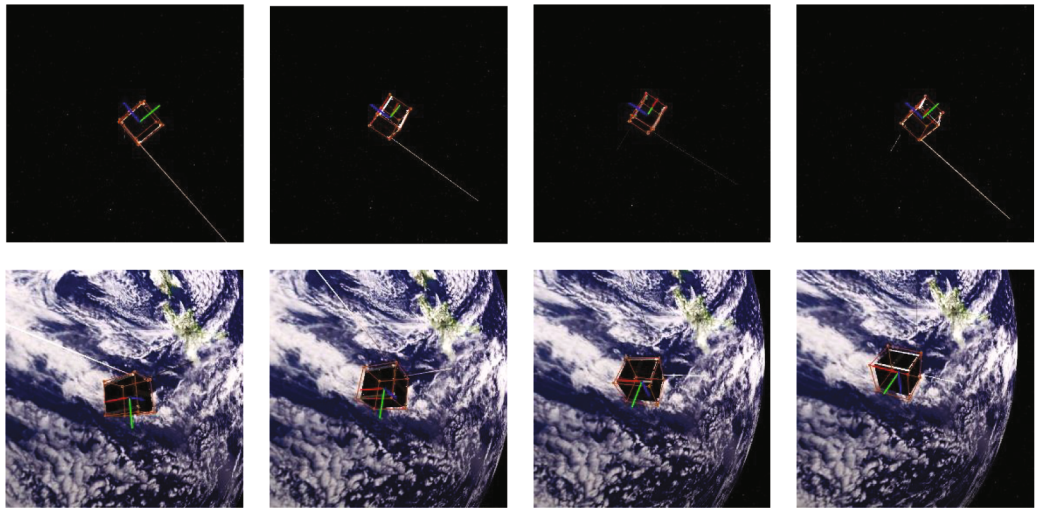

Figure 13: Pose tracking result.

$\Delta\mathbf{T}$ of the 11th frame image and the 10th frame image target was obtained, and supervised learning was performed according to the real value. The ConvLSTM network training set contains 350 videos with a total of 350×90 sequences, a total of 300 epochs are trained, and the number of sequence samples in each epoch is 30,000. The calculated target optical flow field is shown in Figure 12.

4.3.2. Pose Tracking Results. Using the abovementioned target optical flow composition sequence, the tracking results obtained are shown in Figure 13. We use PnP-net and PnP+KF (Kalman's filter) for comparative experiments. To this end, we calculated the pose tracking error in a sequence of images, as shown in Figure 14, and analyzed the percentage of correct estimates in all test sets, as shown in Table 2.

As shown in Table 2, traditional optical flow method has a relatively large error compared to other methods, which is mainly caused by cumulative errors. When using target optical flow in the network for end-to-end pose estimation, the error is significantly reduced, which is similar to the accuracy of the PnP-net algorithm. However, compared with other methods that introduce time domain optimization algorithm, the correctly estimated quantity is about 5% less. It can be summarized from the tracking error results in Figure 14 and Table 2 that using the interframe target optical flow as input can obtain a better pose

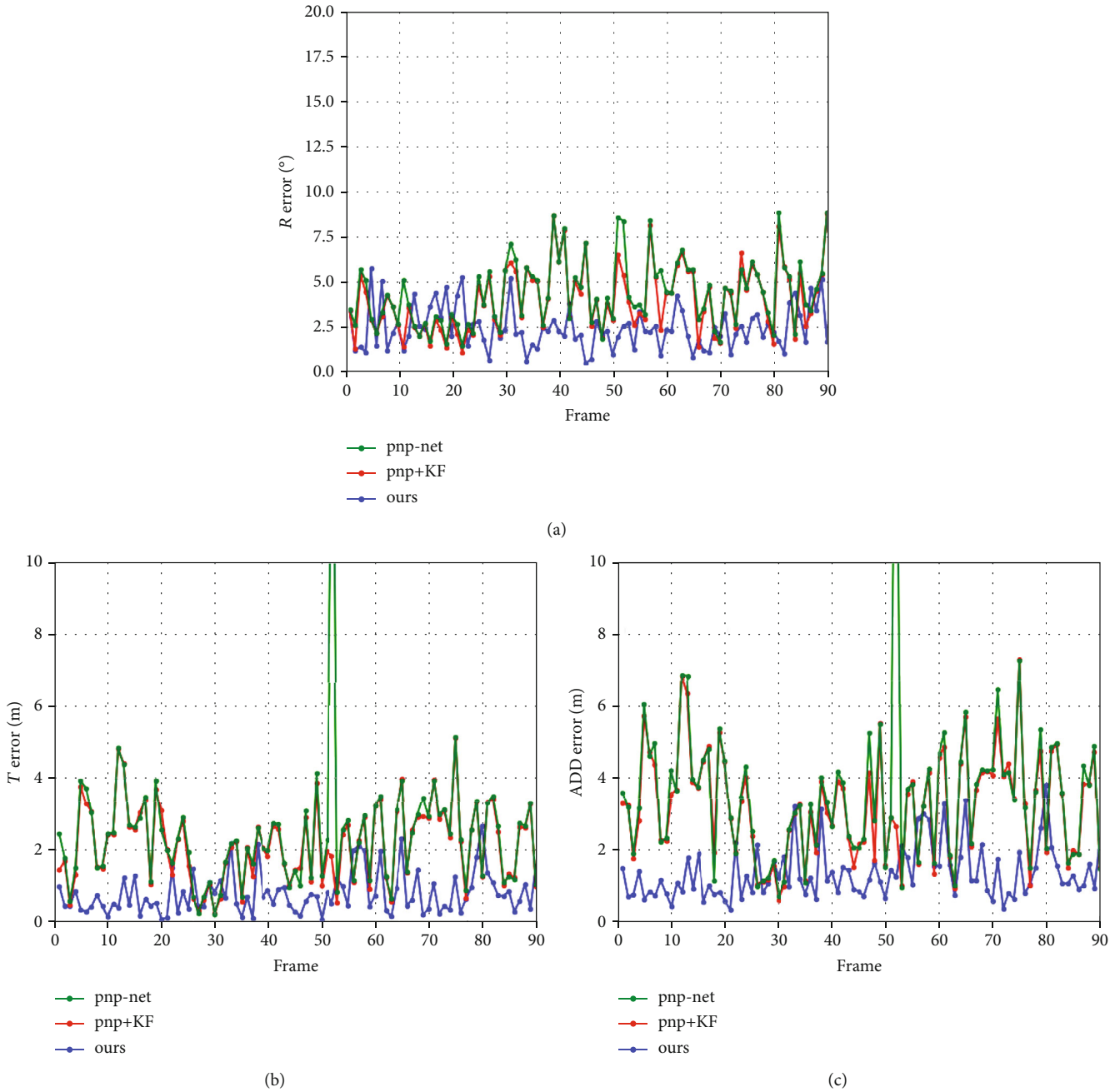

Figure 14: Comparison between the proposed method with PnP-net and PnP+KF: (a–c) the error of rotation and error of translation in one sequence with background influence.

Table 2: Correct estimation of all test sets: ADD ≤ 0.1d, rotation error ≤ 5°, translation error ≤ 0.1d; d represents the diameter of the model.

	ADD – 0.1d	R-5°	T – 0.1d
Traditional optical flow	42.09%	66.35%	53.93%
Target optical flow (without LSTM)	54.81%	74.29%	65.01%
PnP-net	55.22%	73.32%	63.17%
PnP+KF	60.90%	78.43%	65.55%
Wide-depth-range (RANSAC+PnP)	60.14%	76.28%	66.50%
Ours	63.69%	81.52%	70.36%

tracking effect, and the pose estimation result is more stable than that of PnP-net [39], wide-depth-range [40], and PnP+KF.

5. Conclusion

In this paper, a relative pose tracking method based on the target optical flow ConvLSTM framework is proposed, which not only uses the target optical flow to improve the accuracy of the interframe pose estimation but also realizes the analysis of the target motion law with the long-term sequence. The experimental data verifies the feasibility and effectiveness of the proposed method based on synthetic data and standard SwissCube dataset under highly textured background influence and harsh lighting conditions. The result shows that the proposed method is more stable and accurate in pose tracking for known uncooperative spacecraft image sequence compared with other methods. Future work will focus on designing an end-to-end network structure so that the target optical flow could be obtained in a direct and efficient way.

Data Availability

The main network structure and other codes used to support the findings of this study have been deposited in the link (https://github.com/guodong0909/Flow-based-LSTM).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] B. Taylor, G. Aglietti, S. Fellowes et al., "Remove debris mission, from concept to orbit," in *SmallSat 2018-32nd Annual AIAA/USU Conference on Small Satellites*, United States, 2018.

[2] P. F. Proença and G. Yang, "Deep learning for spacecraft pose estimation from photorealistic rendering," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6007–6013, Paris, France, 2020.

[3] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3385–3394, California, 2019.

[4] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[5] O. H. Jafari, S. K. Mustikovela, K. Pertsch, E. Brachmann, and C. Rother, "iPose: instance-aware 6D pose estimation of partly occluded objects," in *Asian Conference on Computer Vision*, pp. 477–492, Springer, 2018.

[6] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134, Germany, 2018.

[7] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and D. Simone, "Satellite pose estimation challenge: dataset, competition design, and results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 4083–4098, 2020.

[8] Y. Huo, Z. Li, and F. Zhang, "Fast and accurate spacecraft pose estimation from single shot space imagery using box reliability and keypoints existence judgments," *IEEE Access*, vol. 8, pp. 216283–216297, 2020.

[9] X.-S. Gao, X.-R. Hou, J. Tang, and H. F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[10] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: an accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.

[11] M. Rad and V. Lepetit, "Bb8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3836, Venice, Italy, 2017.

[12] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 292–301, Salt Lake City, 2018.

[13] C. R. Qi, S. Hao, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, Honolulu, 2017.

[14] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1941–1950, Korea, 2019.

[15] C. I. Patel, S. Garg, T. Zaveri, A. Banerjee, and R. Patel, "Human action recognition using fusion of features for unconstrained video sequences," *Computers and Electrical Engineering*, vol. 70, no. 2, pp. 284–301, 2018.

[16] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se(3)-TrackNet: data-driven 6D pose tracking by calibrating image residuals in synthetic domains," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10367–10373, Las Vegas, NV, USA, 2020.

[17] B. Wen, J. Tremblay, V. Blukis et al., "BundleSDF: neural 6-DoF tracking and 3D reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Canada, 2023.

[18] B. M. Nair, K. D. Kendricks, V. K. Asari, and R. F. Tuttle, "Optical flow based Kalman filter for body joint prediction and tracking using HOG-LBP matching," *Video Surveillance and Transportation Imaging Applications 2014*, vol. 9026, pp. 96–109, 2014.

[19] T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel, "High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9*, pp. 98–111, Berlin Heidelberg, 2006.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[21] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep fisher networks for large-scale image classification," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[22] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, pp. 1–9, Boston, Massachusetts, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, 2016.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, Massachusetts, 2015.

[25] S. Shantaiya, V. Kesari, and M. Kamal, "Multiple object tracking using Kalman filter and optical flow," *European Journal of Advances in Engineering and Technology*, vol. 2, no. 2, pp. 34–39, 2015.

[26] X. Deng, A. Mousavian, X. Yu, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: a Rao–Blackwellized particle filter for 6-D object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, 2021.

[27] Y. Oka, T. Kuroda, T. Migita, and T. Shakunaga, "Tracking 3D pose of rigid object by sparse template matching," in *2009 Fifth International Conference on Image and Graphics*, pp. 390–397, Xi'an, China, 2009.

[28] C. Choi and H. I. Christensen, "3D textureless object detection and tracking: an edge-based approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3877–3884, Vilamoura-Algarve, Portugal, 2012.

[29] A. Crivellaro, M. Rad, Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "A novel representation of parts for accurate 3D object detection and tracking in monocular images," in *Proceedings of the IEEE international conference on computer vision*, pp. 4391–4399, Santiago, Chile, 2015.

[30] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother, "6-DOF model based tracking via object coordinate regression," in *Computer Vision – ACCV 2014. ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M. H. Yang, Eds., vol. 9006 of Lecture Notes in Computer Science, Springer, Cham, 2015.

[31] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5555–5564, California, 2019.

[32] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, Vancouver, BC, Canada, 2013.

[33] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *Proceedings of The 33rd International Conference on Machine Learning, PMLR*, vol. 48, pp. 1747–1756, 2016.

[34] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, vol. 37, pp. 2048–2057, 2015.

[35] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," 2015, https://arxiv.org/abs/1507.01526.

[36] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4346–4354, Santiago, Chile, 2015.

[37] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5308–5317, Las Vegas, 2016.

[38] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Computer Vision – ECCV 2016. ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9907 of Lecture Notes in Computer Science, Springer, Cham, 2016.

[39] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2891–2900, Honolulu, 2017.

[40] Y. Luo, J. Ren, Z. Wang et al., "LSTM pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5207–5215, Salt Lake City, 2018.

[41] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6D object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2930–2939, 2020.

[42] Y. Hu, S. Speierer, W. Jakob, P. Fua, and M. Salzmann, "Wide-depth-range 6D object pose estimation in space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15870–15879, 2021.

[43] X. Yu, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: a convolutional neural network for 6d object pose estimation in cluttered scenes," 2017, https://arxiv.org/abs/1711.00199.

[44] S. Li, C. Xu, and M. Xie, "A robust O (n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.