Geo-localization based on CNN feature matching*

TANG Jin[1,2], GONG Cheng[1], GUO Fan[2], YANG Zirong[2], and WU Zhihu[2]**

1. School of Computer Science and Engineering, Central South University, Changsha 410083, China

2. School of Automation, Central South University, Changsha 410083, China

(Received 16 September 2021; Revised 28 October 2021)

A geo-localization method is proposed for military and civilian applications, which is used when no global navigation satellite system (GNSS) information is available. The open graphics library (OpenGL) is used to build a three-dimensional geographic model of the test area using digital elevation model (DEM) data, and the skyline can thus be extracted with the model to form a database. Then, MultiSkip DeepLab (MS-DeepLab), a fully convolutional semantic segmentation network with multiple skip structures, is proposed to extract the skyline from the query image. Finally, a matching model based on convolutional neural network (CNN) feature is adopted to calculate the similarity between the skyline features of the query image and the DEM database to realize automatic geo-localization. The experiments are conducted at a 202.6 km^2 test site in north-eastern Changsha, China. 50 test points are selected to verify the effectiveness of the proposed method, and an excellent result with an average positioning error of 49.29 m is obtained.

Document code: A **Article ID:** 1673-1905(2022)05-0300-7

DOI https://doi.org/10.1007/s11801-022-1148-0

Recently, automatic geo-localization using a digital elevation model (DEM) and panoramic skyline has emerged as a powerful tool that can support a large range of military and civil applications, especially when no global navigation satellite system (GNSS) information is available. In this case, geo-localization using natural environment images is very useful. However, this is not an easy task because of complex and changeable conditions, such as lighting, vegetation, and seasons, in natural scenes.

A variety of image-based methods have been proposed to locate the position from captured images in recent years. These methods mainly determine the image location by constructing some features to compare the query image with a database image that has geographic markers. These methods can be divided into two categories, geo-localization in urban scenes and geo-localization in nonurban scenes. Compared with the former one, geo-localization in nonurban natural scenes is considered to be more challenging and has gained attention recently. For example, TALLURI et al[1] matched horizon lines extracted from a query image against those rendered from DEM to achieve the geo-localization. STEIN et al[2] also used horizon lines for localization. Localization using horizon line was further studied by NAVAL et al[3, 4]. WOO et al[5] studied navigation of unmanned aerial vehicle in mountain areas using DEM and infrared images with known altitude using altimeter. BAATZ et al[6] used horizon lines to construct local features (contourlets) and find the position. Geo-localization of untagged desert imagery was studied by TZENG et al[7], who proposed a novel skyline-based feature based on concavities. PORZI et al[8] proposed a fast method of automatic photo-to-terrain alignment for precise augmented reality on a mobile device. Smartphone sensors were used as an initial estimate for camera orientation, which was refined by silhouette matching algorithm similar to Ref.[9]. HAMMOUD et al[10] developed a geo-localization framework of street-level outdoor images using multiple sources of overhead reference imagery, including light detection and ranging (LIDAR), DEM and multi-spectral land cover/use imagery. An advanced approach based on horizon lines was presented by CHEN et al[11]. SAURER et al[12] proposed an automated approach for very large-scale visual localization that can efficiently exploit visual information (contours) and geometric constraints simultaneously. GRELSSON et al[13] proposed a position estimation method where the horizon line is extracted in a 360° panoramic image around the unmanned surface vessels. CHIODINI et al[14] performed a sensitivity analysis of the visual position estimator for rover algorithm using data and images provided by the National Aeronautics and Space Administration (NASA) mars exploration rover (MER) and the NASA mars reconnaissance orbiter. FUKUDA et al[15] used the skyline in a dune area to correct the position obtained by GNSS. They first used

* This work has been supported by the National Natural Science Foundation of China (No.61502537), and the Science & Technology Innovation System for National Defense of China (Nos.17-163-11-ZT-002-032-01 and 193-A11-103-11-04).

** E-mail: guofancsu@163.com

GNSS to obtain the current unknown area and the amplitude components were then extracted as skyline features to refine the position.

However, these existing methods mainly adopted hand-crafted features of skyline (e.g., contour words, concave-convex feature, amplitude components, etc) to obtain the final position. According to the published references, these methods seldom obtained accurate meter-level or even 10-meter-level localization in large-scale areas. This may be due to the complexity of the skyline positioning task (human disturbance and vegetation changes, poor skyline discrimination), and the complexity of designing features that describe the skyline more robustly. Compared to the traditional features used to describe the skyline in most studies, convolutional neural network (CNN) can save a lot of manual work and obtain more robust features in images by learning from a large amount of data[16]. Thus, a location algorithm is proposed to improve the accuracy of location estimation in a GNSS-denied environment. The main contributions of the letter are as follows.

The effective encoding of skyline is realized by using the pretrained VGG16 model and principal component analysis (PCA). The VGG16 convolution layer in our method is trained into an encoder to express the high-dimensional feature of the skyline, and we then use PCA to reduce the dimensionality of the CNN feature. The use of CNN makes it easier to extract robust features by training on large datasets than by designing traditional features. Using the learned representation of skyline from the encoder, the skyline feature of query image and DEM can be compared.

A new semantic segmentation model is constructed to extract the skyline from query image. The key idea is that the model combines the skip structure and the image detail obtained from the DeepLab V3+ to realize the combination of high-level semantics and low-level edge information. We refer to our proposed network as the MultiSkip DeepLab (MS-DeepLab). We show experimentally that the MS-DeepLab is better than other existing networks on the skyline extraction task.

Extensive experiments on the real testing points demonstrate that our method generally outperforms some existing methods for localization in GNSS-denied environment. Besides, our method can achieve geo-localization using only DEM data and panoramic images in a large outdoor area in China.

The main idea of this letter is to determine the geographic location of the query image by searching for the skyline image in the database that is most similar to the skyline in the query image. The proposed framework consists of two stages (see Fig.1).

The offline stage consists of constructing the skyline knowledge base, where we use open graphics library (OpenGL) to render panoramic renderings and encode their skylines. The online stage consists of segmenting skyline, encoding skyline image, matching feature to the reference knowledge base, generating probability map and output location.

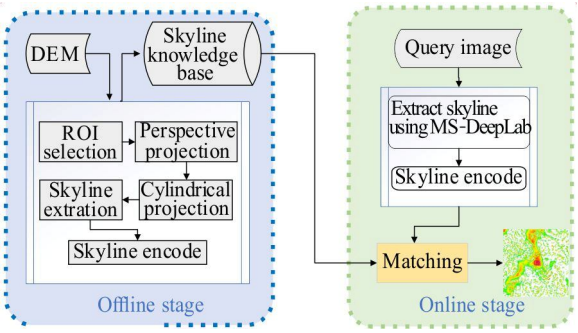


Fig.1 Flowchart of the proposed algorithm

Our query image is in fact a cylindrical projection. Thus, it is necessary to ensure that the images in the panoramic database are also cylindrical projections. Specifically, we developed an OpenGL camera roaming program. When rendering the model, the background is set to a specific pixel value, and the model color is replaced by the model depth. In the program, four images are obtained at each sampling point by controlling the position and viewing angle of the camera. The perspective projection rendering is used as an intermediate result (Fig.2(a)) to express the surrounding environment of the sampling point. The parameters of the rendered image are vertical field of view angle of 38°, horizontal field of view angle of 90°, pitch angle of 0°, and roll angle of 0°. This makes the rendered map also a depth map, and these four maps include all the environmental information around the 360° of the sampling points. Since what we need is a cylindrical projection panoramic image, we thus convert the above four perspective projection renderings to cylindrical projection (Fig.2(b)). Then the four renderings are stitched together to obtain a cylindrical projection panoramic rendering (Fig.2(c)). Therefore, the panoramic database is created and the edge of the skyline (Fig.2(d)) can be easily obtained using pixel segmentation.

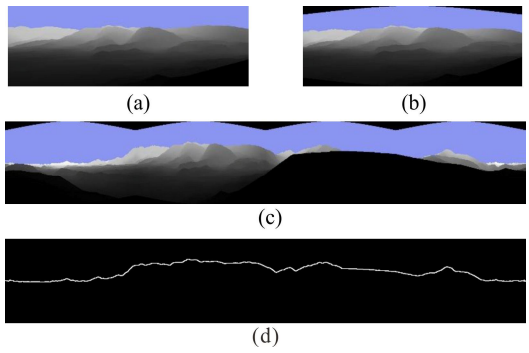
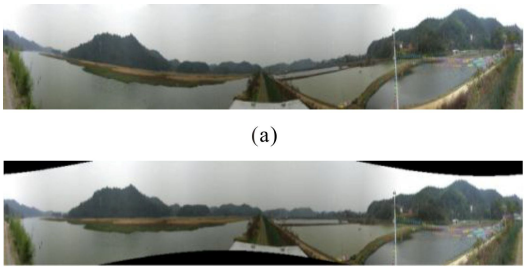


Fig.2 DEM rendering process: (a) Perspective projection rendering; (b) Cylindrical projection rendering; (c) Panoramic image obtained by cylindrical projection; (d) Extracted skyline from image (c)

Since our device generally has an inclination angle, distortion correction is necessary for our system. Fig.3 shows an example of the distortion correction result for panoramic image. One can see that the correction result is consistent with the cylindrical projection panoramic skyline in the database, whose pitch and roll angles are all 0°. In our experiments, the camera's pitch and roll angles are measured using sensors to compensate for the inclination of the natural panoramic image. The inclination of the skyline is corrected after skyline segmentation in our method. Besides, correcting the skyline inclination can also avoid the influence of the black area of the corrected image on the skyline extraction. The digital compass is used here to measure the absolute heading angle of the skyline, and the heading angle is then adopted to adjust the skyline of the query image to the same phase as the skyline in the database. This operation lays a good foundation for the following skyline matching step.



(a)

(b)

Fig.3 Distortion correction results: (a) Before correction; (b) After correction

For our skyline extraction task, the target we need to split is a boundary, not some regions, so we combine the skip structure used in U-Net[17] and the DeepLab V3+[18] model to construct a new semantic segmentation model of MS-DeepLab. On the one hand, DeepLab V3+[18] is proved to be effective in semantic image segmentation. On the other hand, the skip structure is adopted in many existing networks, such as U-Net[17] and fully convolutional network (FCN)[19] to combine both high-level semantics and low-level surface information, which makes it easier to reconstruct image segmentation details and makes the edge of the segmentation results clearer. Fig.4 shows the structure of the model. Different from the existing DeepLab V3+ model, we design some skip structures in the proposed model.

The encoder for DeepLab V3+ uses Resnet as the backbone. We adopt $\boldsymbol{X}^{l}_{\text{EN-out}}$ and $\boldsymbol{X}^{l}_{\text{DE-out}}$ to represent the output feature maps of the l-th layer encoder, $\boldsymbol{X}^{4}_{\text{EN-out}}$ goes through the atrous spatial pyramid pooling (ASPP)[20] module to further extract multi-scale information. We also adopt $\boldsymbol{X}_{\text{ASPP-out}}$ to represent the out feature map of ASPP layer. In DeepLab-decoder, $\boldsymbol{X}_{\text{ASPP-out}}$ is first up-sampled by a factor of 4 to concatenate with $\boldsymbol{X}^{2}_{\text{EN-out}}$, and finally up-sampled by a factor of 2 to obtain a segmentation result $\boldsymbol{X}^{1}_{\text{out}}$. The decode operation can be described as follows

$$\boldsymbol{X}^{1}_{\text{DE-out}} = \text{Conv}(\text{Concat}(\text{Conv}(\boldsymbol{X}^{2}_{\text{EN-out}}), \text{up}_4(\boldsymbol{X}_{\text{ASPP-out}}))),$$

$$\boldsymbol{X}_{\text{out}} = \text{up}(\boldsymbol{X}^{1}_{\text{DE}}, 4), \tag{1}$$

where Conv refers to convolution operation, Concat refers to concatenate operation, and up refers to upsampling operation.

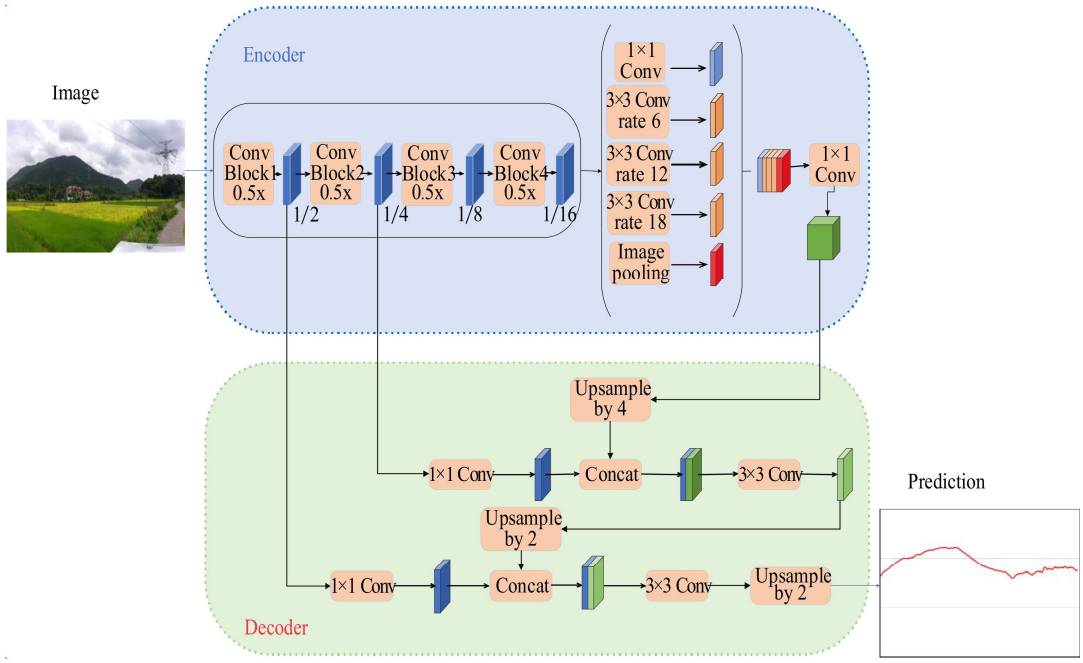


Fig.4 Structure of the proposed MS-DeepLab

The segmentation targets targeted by DeepLab V3+ are regional targets that require more attention to higher-level semantic features, but as a skyline segmentation task, our segmentation target can be seen as a boundary. In order to improve the segmentation accuracy of the boundary, the decoder section of MS-DeepLab pays special attention to the low-level feature maps that contribute to the boundary segmentation, as shown in Fig.4. In MS-decoder, $\boldsymbol{X}_{\text{ASPP-out}}$ is first up-sampled by a factor of 4 and concatenated with $\boldsymbol{X}^2_{\text{EN-out}}$ to obtain $\boldsymbol{X}^1_{\text{DE-out}}$. $\boldsymbol{X}^1_{\text{DE-out}}$ is up-sampled by a factor of 2 and then concatenated with $\boldsymbol{X}^1_{\text{EN-out}}$ to obtain $\boldsymbol{X}^2_{\text{DE-out}}$. Finally, $\boldsymbol{X}^2_{\text{DE-out}}$ is up-sampled by a factor of 2 to obtain the segmentation result $\boldsymbol{X}_{\text{MS-out}}$. The decode operation is described as

$$\begin{aligned}
\boldsymbol{X}^1_{\text{DE-out}} &= \text{Conv}(\text{Concat}(\text{Conv}(\boldsymbol{X}^2_{\text{EN-out}}), \text{up}_4(\boldsymbol{X}_{\text{ASPP-out}}))),\\
\boldsymbol{X}^2_{\text{DE-out}} &= \text{Conv}(\text{Concat}(\text{Conv}(\boldsymbol{X}^1_{\text{EN-out}}), \text{up}(\boldsymbol{X}^1_{\text{DE-out}}, 2))),\\
\boldsymbol{X}_{\text{MS-out}} &= \text{up}(\boldsymbol{X}^2_{\text{DE}}, 2). \qquad (2)
\end{aligned}$$

Due to MS-DeepLab's special attention to low-level features that are useful for improving boundary accuracy, the MS-DeepLab performs better than DeepLab V3+ on the skyline extraction task.

Generally, the extracted skyline has two characteristics. One is that the skyline may have some missing data and some noise caused by interference, and the other is that the skyline has local similarity, which means that the skylines in adjacent areas may have higher similarity. Due to factors such as vegetation coverage or artificial obstacles, there are slight differences between the skyline extracted from the query image and the skyline obtained from DEM rendering. We propose a matching model based CNN feature to solve the problem. We used the VGG16 model to extract the image features and the PCA algorithm to dimensionally reduce the features.

Here, the pretrained VGG16[21] is used to construct the feature encoder, and the feature map output is then extracted by the last convolutional layer of VGG16 as the feature expression of the image. Thanks to the effective encoding of skyline by using the pretrained VGG16, the high-dimensional feature can be better expressed. The VGG16 convolution layer pretrained on the ImageNet dataset is used to construct our matching model. Although the VGG model aims at the feature extraction of the general dataset, the experimental results prove the effectiveness of the VGG model for our task.

For offline knowledgebase establishment, the feature encoder is used to encode the skyline of the DEM panoramic rendering to obtain the offline database. In order to reduce the storage space of the features and improve the efficiency of the operation, we use this feature set to train the PCA model to downscale the CNN features, and the downscaled feature set is used as a knowledge base for online localization.

For online localization, a feature encoder is used to encode the skyline feature of the query image, and dimensionally reduce skyline features using a PCA model trained in the offline phase. Then, the Euclidean distance between the skyline feature of the query image and each DEM rendering in the offline skyline knowledge base is calculated to obtain the probability matrix of each point in the region of interest. To obtain the final location of the target, a Gaussian filter is used here to smooth the probability matrix. Finally, the point with the highest similarity (minimum distance) is selected as our final positioning point.

Fig.5 shows our experimental equipment, and the reconnaissance ball is used as our main data acquisition equipment. The ball contains a horizontal sensor to obtain the pitch angle and roll angle of the equipment. A digital compass is also used to collect the heading angle by using geomagnetic information. The sensor accuracy of the roll angle and pitch angle collected by the horizontal sensor is 0.1°, and the accuracy of the heading angle is 1° when no obvious interference exists. We installed the ball on the experimental vehicle, and the height from the ground was approximately 2.5 m when the ball was raised. Thus, it is convenient for us to collect data in mountain or hilly areas.

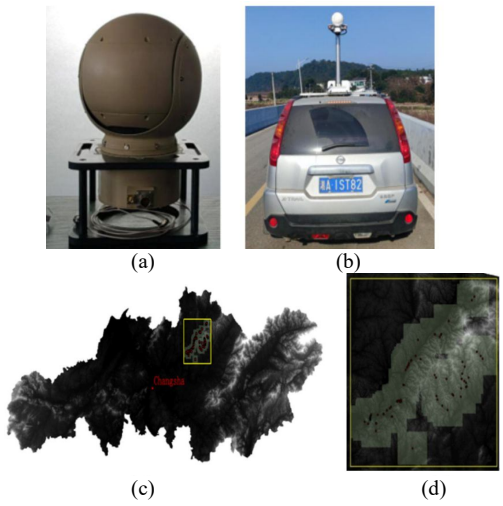


Fig.5 Experimental equipment and testing points: (a) Reconnaissance ball; (b) Our experimental vehicle; (c) Location of our test area; (d) Distribution of 50 test points

50 test points are selected to verify the effectiveness of the proposed method. For each testing point, we collect a panoramic query image that contains position, attitude and heading angle. The real position information was used as a label to obtain the positioning error, and the attitude and heading angle were used to correct the panoramic images. As shown in Fig.5(c), all the samples are distributed in a hilly area in north-eastern Changsha, China. The corresponding DEM data is obtained from the Hunan Remote Sensing Center of China. As can be seen in Fig.5(d), a total area of 202.6 km² is regarded as

our final test area. Besides, we also notice that the sampling density directly affects the positioning error of the final positioning system. To balance the calculation speed and the positioning accuracy, we take 10 m as our sampling interval to generate our panoramic database.

The dataset that used for training our MS-DeepLab to extract skyline has about 2 000 images, including hundreds of real-captured images and a part of Saurer's public dataset[12]. Our MS-DeepLab is trained on this dataset from scratch. In our experiments, we compare the inter section over union (IOU) index value of the proposed MS-DeepLab with those of U-Net and DeepLab V3+ by using the same training strategy. Experimental results show that the IOU values are 72.53, 71.21 and 76.04 for U-Net, DeepLab V3+ and MS-DeepLab, respectively. The improvement in skyline extraction accuracy is of great significance to the subsequent localization process. Although the improvement in IOU is only a few percent, it is still very important because these few percent are likely to be the pixel that is difficult to classify. Fig.6 shows the effect of partial skyline segmentation, where the scene was successfully localized due to the improvement in skyline accuracy.

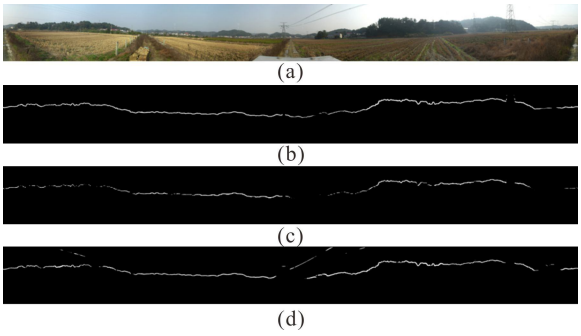


Fig.6 Comparison of segmentation results: (a) Original image; (b) Skyline obtained using MS-Deeplab; (c) Skyline obtained using DeepLab V3+; (d) Skyline obtained using U-Net

To visualize the location results, we plot our location probability. Fig.7(a) shows the positioning probability map, and the corresponding local amplified probability map is shown in Fig.7(b). In Fig.7(b), the center point of the red box is our location point, and the center point of the blue box is the labelled location. The closer to the red color, the higher the probability of the point is. As shown in Fig.7, the predicted location point is very close to the ground truth, which verifies the effectiveness of our method.

In our experiments, we also find that when the error is very small, the positioning point is located near the labelled point. In this case, the positioning is regarded as successful. Otherwise, when the error is large, the positioning point may randomly appear in the region of interest since the point with the highest possibility is bound to be output. This is regarded as a failure case.

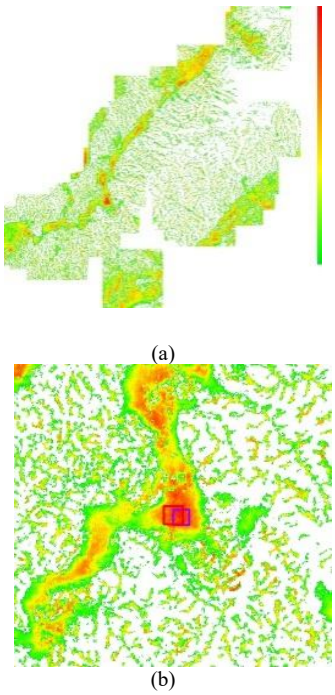


Fig.7 Positioning probability maps: (a) Global probability map; (b) Local amplified probability map

According to our principle, for the 50 test points, when directly using the extracted features for localization, the positioning success rate is 92%, and the average positioning error is 49.29 m. More specifically, 64% of the sample positioning errors are less than 50 m, 82% of the errors are less than meters. After using PCA to downscale the features, the localization accuracy did not decrease when the feature dimension was reduced to 2 048 dimensions. The decrease appeared when the feature dimension was reduced to 1 024 dimensions, as shown in Fig.8. After dimensionality reduction, the localization time was reduced to approximately 30 s.

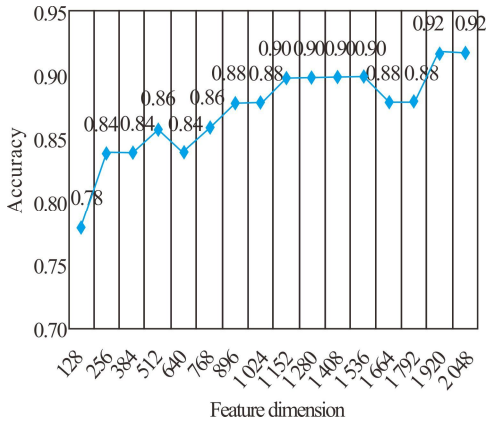


Fig.8 The effect of the number of dimensions of retained features on success rates

For the failure case, we find that these samples are generally captured in some extreme situations. For

example, a large part of the skyline is largely occluded by houses, towers or other men-made objects. The mountain peak is lost because of the limitation of the image view angle. All these factors make it very difficult to obtain a nearly complete skyline, and thus finding the most similar skyline in the DEM rendering database becomes impossible. Besides, we also compared our method with some influential localization methods. Tab.1 shows the comparison results. Test area defines the area on which the method has been tested in original publication; localization success rate (Local. succ.) denotes the best result achieved with given method; average error (Avg. err.) and maximum error (Max. err.) denote the average distance and maximum distance from the ground-truth position which is considered to be correct localization.

Tab.1 Performance comparison of different geo-localization methods

Method	Test area	Local. succ.	Avg. err.
Ref.[1]	148 km²	—	—
Ref.[2]	298 km²	—	—
Ref.[4]	—	—	—
Ref.[3]	900 km²	—	—
Ref.[5]	2.28 km²	—	393 m
Ref.[6]	40 000 km^2	88%	Max. err. 1 km
Ref.[7]	10 000 km^2	—	—
Ref.[8]	100 places in Alps	—	1.87°
Ref.[9]	28 photos in Alps, Rocky Mnts.	86%	—
Ref.[10]	20 000 km^2	49%	—
Ref.[11]	10 000 km^2 (America Asia)	60%	Max. err. 4.5 km
Ref.[12]	40 000 km^2	88% (<1 km)	—
Ref.[13]	0.006 4 km^2	—	2.72 m
Ref.[14]	1 km^2	—	51 m
Ref.[15]	0.25 km^2	—	1.81 m
Ours	**202.6 km^2**	**92% (<200 m)**	**43.13 m**

As can be seen in Tab.1, most existing methods are still not very precise. For example, in the results of SAURER et al[12], the distance under which the query is considered as correctly localized is 1 km, which is longer than our method, whose related distance is 200 m. In case of horizon-based localization proposed by SAURER et al[12], 40% of query images need user interaction for discovering horizon line, mainly due to tree occlusions which arise in real-world photos quite often[22]. Thanks to MS-DeepLab, our method can automatically extract skyline even in this complex situation. Several approaches for camera orientation estimation are also provided in Tab.1. The localization success rates of the two methods are 86% and 88%, respectively. The localization success rate of our method is the highest compared with other methods shown in Tab.1. Besides, our test area can be further extended to very large scale as long as certain conditions are met. It can be generally assumed that the larger the location area, the greater the probability of location failure, and the greater the probability of location error. Therefore, considering the location area size, location success rate and location accuracy, it can be assumed that our method has a greater advantage over some influential existing methods.

A geo-localization algorithm has been studied in this letter. First, we use OpenGL to render the DEM data into a three-dimensional model to establish a skyline knowledge base for retrieval and positioning. Second, MS-DeepLab is proposed to extract the image skyline. Third, the pretrained CNN model is used to extract the high-dimensional feature information as the feature expression of the skyline, which greatly improves the localization accuracy. A comparative study is proposed with a few representative methods, which demonstrates that similar or better results can be obtained by using the proposed geo-localization method.

Statements and Declarations

The authors declare that there are no conflicts of interest related to this article.

References

[1] TALLURI R, AGGARWAL J. Position estimation for an autonomous mobile robot in an outdoor environment[J]. IEEE transactions on robotics and automation, 1992, 8(5): 573-584.

[2] STEIN F, MEDIONI G. Map-based localization using the panoramic horizon[J]. IEEE transactions on robotics and automation, 1996, 11(6): 892-896.

[3] NAVAL P C. Camera pose estimation by alignment from a single mountain image[EB/OL]. (2010-07) [2021-10-11]. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=87E95F9B1E103A9679DA9009-C4CF1-F74?doi=10.1.1.102.9949&rep=rep1&type=pdf.

[4] NAVAL P C, MUKUNOKI M, MINOH M, et al. Estimating camera position and orientation from geographical map and mountain image[EB/OL]. (1997-04) [2021-10-11]. http://citeseerx.ist.psu.edu/viewdoc/Download?doi=10.1.1.14.3619&rep=rep1&type=pdf.

[5] WOO J, SON K, LI T, et al. Vision-based UAV navigation in mountain area[C]//Proceedings of the IAPR Conference on Machine Vision Applications (IAPR MVA), May 16-18, 2007, Tokyo, Japan. 2007: 236-239.

[6] BAATZ G, SAURER O, KOSER K, et al. Large scale visual geo-localization of images in mountainous terrain[C]//Proceedings of the 12th European Conference

on Computer Vision, October 7-13, 2012, Florence, Italy. Berlin, Heidelberg：Springer-Verlag, 2012：517-530.

[7] TZENG E, ZHAI A, CLEMENTS M, et al. User-driven geolocation of untagged desert imagery using digital elevation models[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 23-28, 2013, Portland, OR, USA. New York：IEEE, 2013：237-244.

[8] PORZI L, BULO S R, VALIGI P, et al. Learning contours for automatic annotations of mountains pictures on smartphone[C]//Proceedings of the International Conference on Distributed Smart Cameras, September 5, 2014, California, USA. New York：ACM, 2014：131-136.

[9] BABOUD L, CADIK M, EISEMANN E, et al. Automatic photo-to-terrain alignment for the annotation of mountain picture[C]//Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition, June 20-25, 2011, Colorado Springs, Colorado, USA. New York：IEEE, 2011：41-48.

[10] HAMMOUD R I, KUZDEBA S A, BERARD B, et al. Overhead-based image and video geo-localization framework[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 23-28, 2013, Portland, OR, USA. New York：IEEE, 2013：320-327.

[11] CHEN Y, QIAN G, GUNDA K, et al. Camera geolocation from mountain images[C]//Proceedings of the 18th International Conference on Information Fusion, July 6-9, 2015, Washington, DC, USA. New York：IEEE, 2015：1587-1596.

[12] SAURER O, BAATZ G, KOSER K, et al. Image based geo-localization in the Alps[J]. International journal of computer vision, 2016, 116(3)：213-225.

[13] GRELSSON B, ROBINSON A, FELSBERG M, et al. GNSS-level accurate camera localization with HorizonNet[J]. Journal of field robotics, 2020, 37(6)：951-971.

[14] CHIODINI S, PERTILE M, DEBEI S, et al. Mars rovers localization by matching local horizon to surface digital elevation models[C]//Proceedings of the IEEE International Workshop on Metrology for AeroSpace, June 21-23, 2017, Padua, Italy. New York：IEEE, 2017.

[15] FUKUDA S, NAKATANI S, NISHIYAMA M, et al. Geo-localization using ridgeline features extracted from 360-degree images of sand dunes[C]//Proceedings of the 15th International Conference on Computer Vision Theory and Applications, February 27-29, 2020, Valletta, Malta. New York：IEEE, 2020：621-627.

[16] DANIEL S T, LI M, MARGARET H. Face recognition：from traditional to deep learning methods[EB/OL]. (2019-11-15) [2021-10-11]. https：//arxiv.org/pdf/1811.00116.pdf.

[17] RONNEBERGER O, FISCHER P, BROX T. U-net：convolutional networks for biomedical image segmentation[C]//Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, October 5-9, 2015, Munich, Germany. Berlin, Heidelberg：Springer-Verlag, 2015：234-241.

[18] CHEN L C, ZHU Y, PAPANDREOU G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European Conference on Computer Vision, September 8-14, 2018, Munich, Germany. Berlin, Heidelberg：Springer-Verlag, 2018：833-851.

[19] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 7-12, 2015, Boston, MA, USA. New York：IEEE, 2015：3431-3440.

[20] CHEN L C, PAPANDEOU G, KOKKINOS I, et al. Deeplab：semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40：834-848.

[21] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2014-07-15) [2021-10-11]. https：//arxiv.org/pdf/1409.1556v6.pdf.

[22] BREJCHA J, CADIK M. State-of-the-art in visual geo-localization[J]. Pattern analysis and applications, 2017, 20：613-637.