S & M 3803

Automatic Merging Method for Sectional Map Based on Deep Learning

Shifan Liu,[1,2] Chen Xing,[1,2] Chengwei Dong,[1,2*] Yunhan Li,[1,2] and Peirun Cao[1,2]

[1]Beijing Institute of Surveying and Mapping, 15 Yangfangdian Road, Haidian District, Beijing 10038, China
[2]Beijing Key Laboratory of Urban Spatial Information Engineering,
15 Yangfangdian Road, Haidian District, Beijing 10038, China

(Received June 27, 2024; accepted October 11, 2024)

Keywords: deep learning, computer vision, sectional map, image registration, base map production

Owing to time and scene constraints, a significant number of sectional maps exist in paper form. These maps contain a vast amount of data and hold high information value. However, they often suffer from issues such as annotations, stains, deformation, and missing content during preservation. Traditional processing methods require a large amount of manual image registration, which is extremely inconvenient. In this study, a map image labeling program is designed using OpenCV to prepare a map image dataset, and the U2Net-p algorithm for map segmentation is trained on this dataset. Furthermore, a comprehensive method for automatically merging sectional maps is designed and implemented, which can repair and process sectional maps and seamlessly integrate them into target grids according to map sheet numbering rules. This method has been applied to the production of base maps for natural resource demarcation projects, achieving a stitching accuracy of 96.67% on marked anchor points and considerably improving processing speed. This indicates that our approach has broad application value in the field of automatic stitching and fusion of sectional map images.

1. Introduction

At the beginning of the 21st century, with the rapid development of digital technology, the surveying and mapping industry has gradually transitioned from the traditional manual or machine drawing method to advanced data acquisition technology. As information technology continues to progress, many paper maps have been converted into electronic image data by scanning.(1–3) However, for large paper maps, complete preservation is often cost-prohibitive owing to their oversized dimensions, leading to their storage in the form of rectangular sections.(4–6) These old maps contain valuable geographic information data that holds significant value in many key fields such as urban planning iteration, urban history research, cultural heritage protection, and natural resources right confirmation.(7,8) Thus, making full use of these old maps will have a far-reaching impact.

[*] Corresponding author: e-mail: donenei@126.com

https://doi.org/10.18494/SAM5206

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

Before utilizing sectional maps, it is typically necessary to concatenate and merge them to form a complete map. Currently, there are two main methods to merge sectional maps. The first involves vectorizing sectional maps and then creating a full map based on the vector map.(9) The second method directly merges raster images to form a full map.(10) However, when dealing with complex map elements in the first method, it is difficult to achieve automatic vector work and extra manual intervention cannot be avoided. On the other hand, although the second scheme currently has a relatively mature raster map image fusion technology, it necessitates that the segmented map be clear, standard, and non-defaced.

Unfortunately, owing to the constraints of the times, many old maps suffer from various issues such as the lack of geographical coordinates in the map outline, the lack of accurate geographical concepts, and excessive labeling and defacing during preservation and use. Additionally, limited by the scanning technology at that time, the scanning process may also produce shadows and deformities, which further complicate the utility of sectional maps. At present, there are few studies on utilizing these low-quality sectional map images, and the mainstream processing method involves manually marking feature points and carrying out image registration using ArcGIS or Photoshop to calibrate, cut, and assemble graphics into a usable map form.(11) This process is not only cumbersome and requires prior knowledge, but also results in significant repetitive work. As a result of these challenges, handling sectional maps has become burdensome under current strict map usage standards, making it difficult to utilize them effectively.

In recent years, there has been rapid development in computer vision technology, and the image segmentation algorithm has become more mature and continuously applied in various fields such as face recognition and medical imaging.(12,13) To effectively make use of old sectional maps, we aim to utilize computer vision technology to train a high-performance map image segmentation algorithm model in the field of mapping, use various computer graphics methods to repair and restore the distortion of segmented map images, and finally achieve automatic segmentation map fusion throughout the entire process.

In project implementation, we have also designed a method of matching marker anchor points to evaluate the effectiveness of the merged map image. The accuracy rate of the final assembly reached 96.67%, which was significantly higher than that achieved through manual processing. This demonstrates that the automatic merging scheme proposed in this paper not only considerably improves the efficiency but also enhances the precision of map fusion. It also proves that our proposed automatic merging scheme can effectively leverage historical data resources, injecting new vitality into their utilization while meeting actual production needs. Therefore, it holds significant promotional value.

2. Related Methods

In this section, the deep neural network model and computer graphics method used in this study are introduced.

2.1 U2Net and U2Net-p

U2Net is a salient object detection method that performs well in the foreground-background segmentation field.(14) This method adopts an encoder-decoder structure, with each encoder-decoder module consisting of a U-Net network. The algorithm architecture is shown in Fig. 1, where the encoder downsamples the image to extract its features, and the decoder then upsamples these features while integrating cross-layer connections from the encoder to achieve effective target detection capabilities. Through multi-scale feature fusion and skip connection mechanisms, U2Net can effectively identify salient objects in images and has been widely used in tasks such as image segmentation and object detection.

U2Net-p also uses an encoder-decoder structure but reduces network depth and computational load, serving as a lightweight version of U2Net. By lowering model complexity, U2Net-p enables the rapid processing of images with high detection accuracy even when computational resources are limited.

2.2 Affine transformation and projective transformation

Affine and projective transformations are two commonly used geometric transformation techniques in computer vision.(15) Affine transformation maintains the relative angles and proportions between images through operations such as translation, rotation, and scaling. In map image processing applications specifically, affine transformation is used to correct distortions

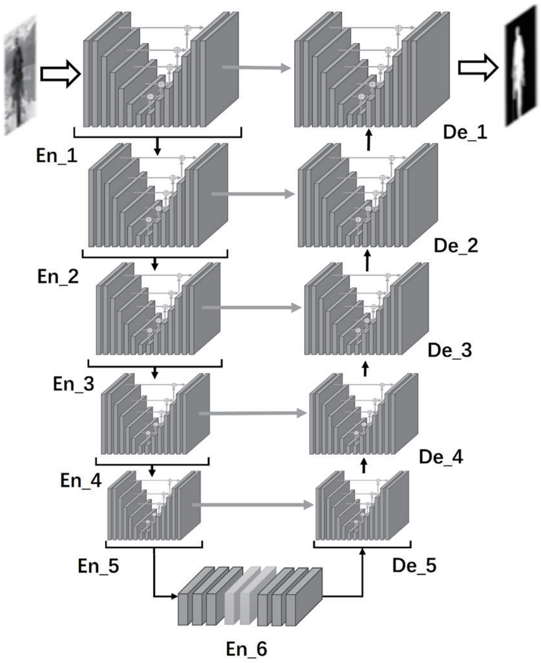


Fig. 1. Diagram of U2Net architecture.

and tilts in images. On the other hand, projective transformation changes the angular relationships between images and addresses distortions caused by the scanning perspective, enabling the accurate extraction of geographic information.

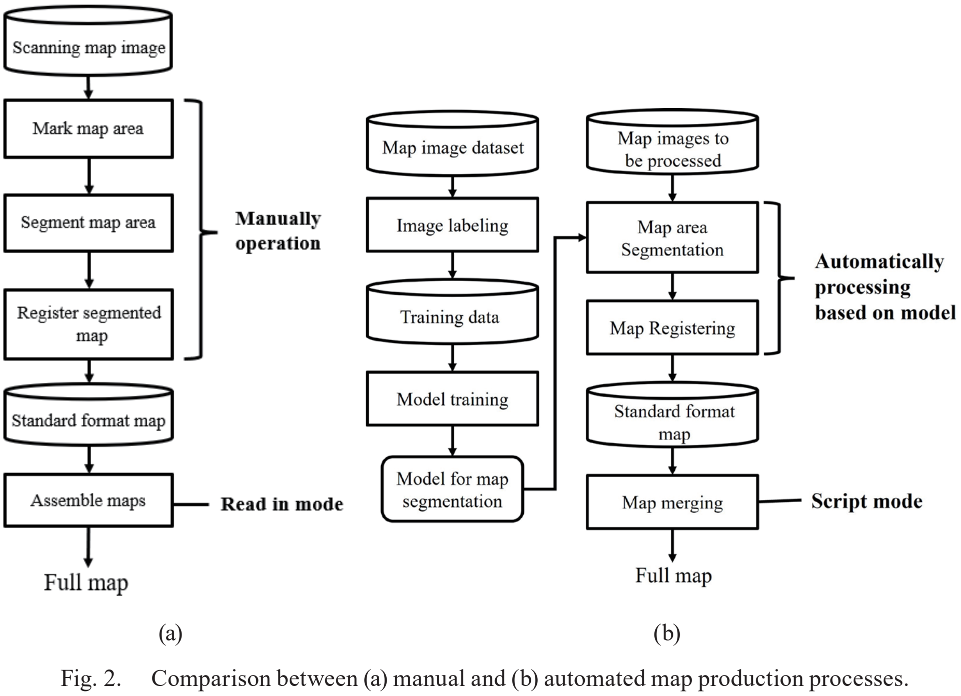
2.3 Image interpolation methods

Image interpolation is an important image processing technique in computer vision, used for enlarging or reducing images, improving image quality, or filling in gaps within images.(16) Common image interpolation methods include nearest-neighbor interpolation, bilinear interpolation, and bicubic interpolation. In map image processing, image interpolation is mainly used to fill missing areas in map images, enhancing the completeness and usability of the images.

3. Base Map Preparation Based on Sectional Maps: Content and Process

A large number of old map images do not conform to the standard map format and require certain processing before use. Figure 2(a) illustrates the general process of preparing a base map from sectional map images.

1. Mark and segment the map area from the original image.
2. Register the segmented image areas according to the standards for map usage, adjusting to make the map format uniform.
3. Assemble and merge multiple images to form a base map.



(a) (b)

Fig. 2. Comparison between (a) manual and (b) automated map production processes.

In traditional production processes, map area marking relies on tools such as Photoshop or ArcGIS, where individuals manually annotate the vertices of map edges, marking out map areas in closed sets of vertices for segmentation. The segmented map areas are typically irregular and require manual dragging and vertex adjustment for image registration to conform to the standardized map format. Finally, standard map images are stitched together with tools such as Photoshop or ArcGIS. The manual graphic processing workflow requires the handling of each map image individually, with workload linearly increasing with map size and image count, leading to insufficient processing accuracy and significant repetitive workload. Moreover, the parallel processing of a massive number of graphic files during multi-image stitching imposes a heavy burden on graphic processing software, resulting in poor merging performance and longer processing times as the number of sectional maps to be stitched increases.(17)

To address this issue, we propose an automated base map preparation method. This method, based on computer vision and deep learning technologies, automatically registers and merges old map images to form a base map. The preparation process is depicted in Fig. 2(b). First, data annotation is performed to create a map image dataset, and the U2Net-p for salient target segmentation is trained to realize the automatic segmentation of map areas. In image registration, various transformations including affine transformation, projective transformation, and several interpolation algorithms are implemented to fine-adjust images; thus, automated image registration is achieved, and the cut irregular graphics are mapped to the standard format. In the standard map assembling stage, scripting language is used to realize automatic image stitching, This approach avoids the challenge of loading a large number of images into computer memory and achieves the full-process automation of base map stitching, which greatly improves efficiency while reducing manual intervention and errors.

3.1 Dataset preparation

On the basis of the map storage and distribution platform of the Beijing Institute of Surveying and Mapping, we made a historical map image dataset using a total of 1,000 sectional maps of various types and scales in Beijing from 1990 to 2007. Using OpenCV, we employed a semi-automated method to prepare the labeled images for the training dataset. A labeled image is a binary image of two categories, where the map area contours are marked by closed shapes formed by vertex sets and the foreground (map area) is filled with 0, while the map background (other area) is filled with 1.

Figure 3(a) shows the flowchart of label production, and the specific process is as follows.

1. Perform morphological transformations. Use the cv2.morphologyEx() function to enhance the directional features of the image. Set the kernel to [0,0,0,1,0,0,0] for the horizontal direction and its transpose for the vertical direction. Apply both erosion and opening operations on the image to remove noise and reinforce the horizontal and vertical features, as shown in Step 1 of Fig. 3(b).
2. Detect map contours. Use the cv2.findContours() function to detect and filter the horizontal and vertical boundaries of the image. Then, calculate the intersection of these edges to approximate the boundary of the map area, as shown in Steps 2 and 3 of Fig. 3(b).

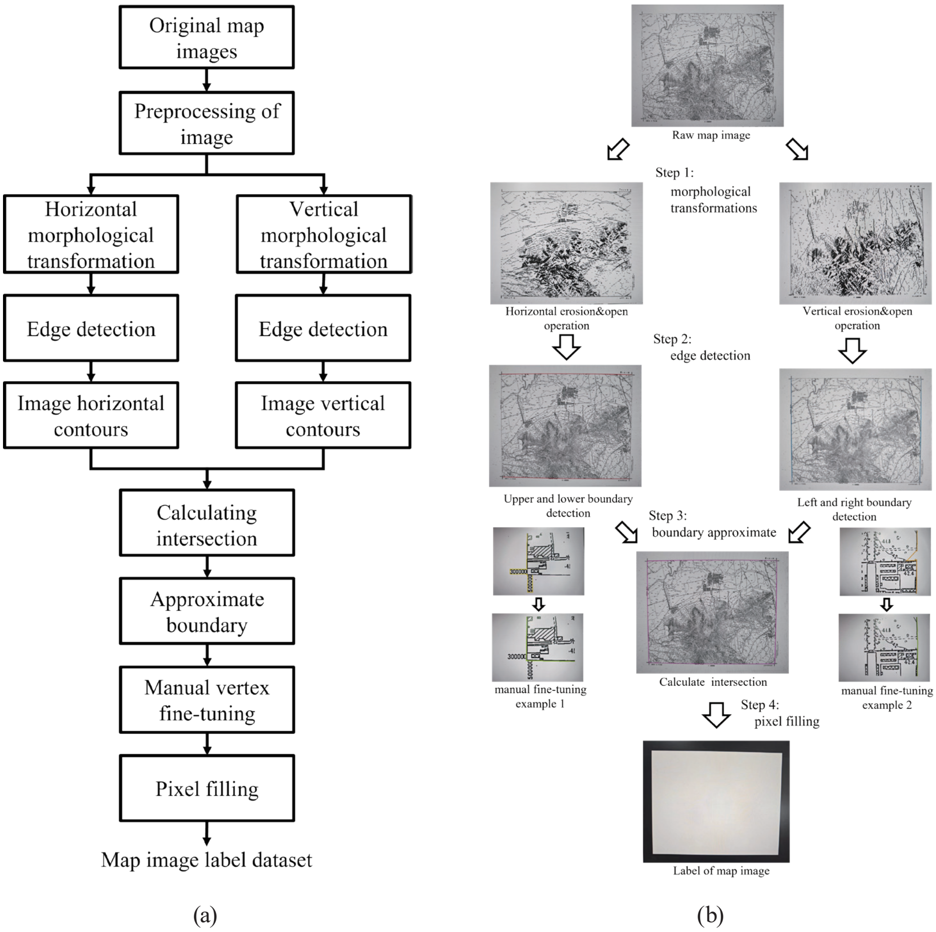

Original map images
Preprocessing of image
Horizontal morphological transformation
Vertical morphological transformation
Edge detection
Edge detection
Image horizontal contours
Image vertical contours
Calculating intersection
Approximate boundary
Manual vertex fine-tuning
Pixel filling
Map image label dataset
Raw map image
Step 1: morphological transformations
Horizontal erosion&open operation
Vertical erosion&open operation
Step 2: edge detection
Upper and lower boundary detection
Left and right boundary detection
Step 3: boundary approximate
manual fine-tuning example 1
Calculate intersection
manual fine-tuning example 2
Step 4: pixel filling
Label of map image
(a)
(b)

Fig. 3. (a) Flowchart and (b) diagram of label preparation.

3. Manual vertex fine-tuning. In cases where poor image quality or interference causes errors in boundary detection, manual adjustments are required. The map's boundary is represented by a series of vertices. In the visual annotation tool, such as LabelMe, manually adjust the number and positions of these vertices to better align with the actual map edges, as illustrated in the two examples in Step 3 of Fig. 3(b).
4. Execute pixel filling. Fill the map region inside the boundary (the foreground) with a pixel value of 255 and the area outside the boundary (the background) with a value of 0. This process creates a binary label image, where the map foreground is white and the background is black, as shown in Step 4 of Fig. 3(b).

To enhance the map image dataset and improve the model's generalization ability, various data augmentation methods are employed to expand the original dataset.

1. Geometric enhancement: utilizing operations such as rotation, flipping, and scaling to simulate deformations that maps might encounter in real-world usage.
2. Affine transformation: This is utilized to replicate distortions caused by angle changes during the scanning process.
3. Adding noise and random erasure: This is performed to mimic missing or blurred information on maps, helping the model better understand and handle uncertainties.

By applying a combination of various data augmentation methods, the original map dataset expands threefold, providing ample data samples for subsequent model training. To ensure the effectiveness of training, the dataset is divided into a training set and a test set in a ratio of 9:1.

3.2 Algorithm training

The task of map image segmentation can be classified as a binary classification problem of foreground–background separation. The distinction between the foreground and background in map images is relatively clear, making it possible for the U2Net-p algorithm to meet requirements with limited resources. The training process utilizes the cross-entropy loss function to calculate the corresponding loss for each decoder layer's output as well as the final output saliency map's loss, and a weighted sum of each layer's loss is used as an overall loss function:

$$L = \sum_{m=1}^{M} w^m l^m + w_{fuse} l_{fuse}, \tag{1}$$

where l^m is the loss of the m layer decoder, w^m is the corresponding weighting, and M = 6 is the number of decoder layers. w_{fuse} and l_{fuse} are the loss and weighting of the final fusion output saliency, respectively. In each training epoch, the original image is input into the algorithm, and the loss between the output and the label image is calculated. After each training batch, the loss is backpropagated and the model is updated. The training environment and hyperparameter configuration used in this study are shown in Table 1.

After 250 epochs of training with the prepared training dataset, the model basically converged. The loss convergence curve is shown in Fig. 4.

Table 1
Hyperparameter configuration of the experiment.

Option	Optimizer	Learning rate	Batch size	No. of epochs
Value	Adam	0.05	60	250

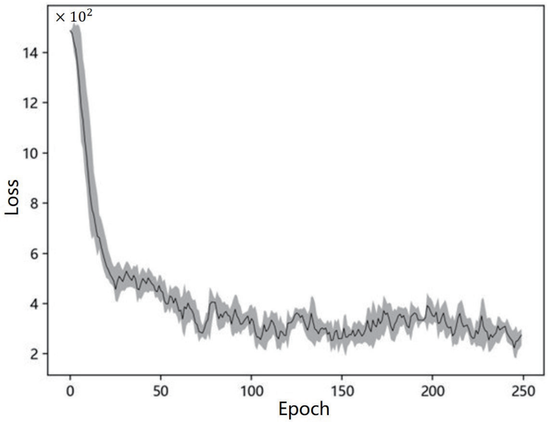


Fig. 4. Convergence curve of model training.

The loss convergence curve indicates that the model maintains an overall converging trend. In the first 50 iterations, the model's fitting ability is poor, resulting in a larger loss value and thus a higher loss reduction speed. In subsequent training, the model achieves stable convergence, and after approximately 200 iterations, the loss reduction trend approaches a stable state.

3.3 Model evaluation

To comprehensively evaluate the performance of the U2Net-p model in the map segmentation task, the mean Intersection over Union (mIoU) was proposed as an evaluation metric, which is defined as the average classification accuracy of each type of pixel.(16) Classification accuracy is further defined as the ratio of the number of correctly predicted pixels to the total number of pixels in that category.

On the test dataset, our trained U2Net-p model achieved an mIoU value of 98.63%, indicating exceptional performance in map area segmentation ability. Considering that the foreground part occupies the vast majority of the total area of the images, this high mIoU strongly demonstrates the effectiveness and accuracy of U2Net-p, and few minor segmentation errors at the edges of some maps are likely caused by annotation errors in the dataset or limitations in model training.

3.4 Fully automated map production process

On the basis of the accurate segmentation capabilities of the trained U2Net-p, we further designed and implemented an automated base map production process, as shown in Fig. 5. U2Net-p receives the raw map image and outputs a binary mask of the map area. Using OpenCV, we extracted the boundary of the mask and represented it as a closed set of vertices. With this set of boundary vertices, the map region is segmented and registered. The specific process is as follows.

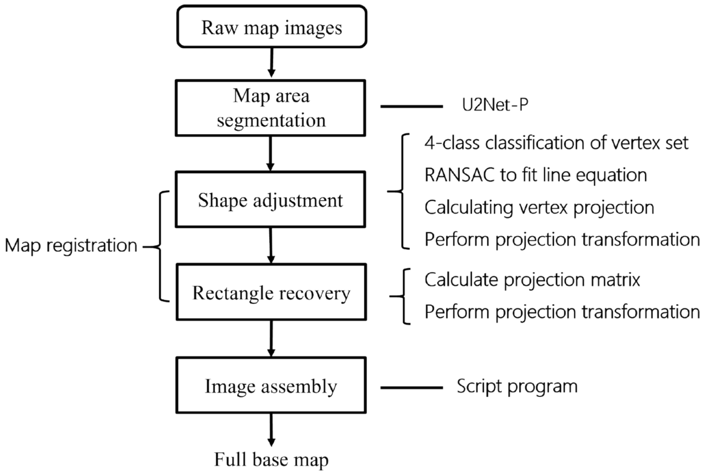

Raw map images
Map area segmentation
U2Net-P
4-class classification of vertex set
RANSAC to fit line equation
Calculating vertex projection
Perform projection transformation
Shape adjustment
Map registration
Rectangle recovery
Calculate projection matrix
Perform projection transformation
Image assembly
Script program
Full base map

Fig. 5. Automated base map production.

1. Take raw map images as input, and U2Net-p produces binary masks where the map area (foreground) is white and the background is black. The mask is used to segment the map area and compute the boundary vertices, denoted as $N = \{n_1, n_2, ..., n_i, ..., n_{|N|}\}$.
2. The vertex set N is classified into four groups on the basis of the edges they belong to, resulting in four sets, N_1, N_2, N_3, and N_4, each corresponding to one of the four sides of an approximate quadrilateral.
3. Apply the RANSAC algorithm(18) to four vertex sets and fit the four straight-line equations l_1, l_2, l_3, and l_4. Then, calculate the intersection points to obtain the four vertices d_1, d_2, d_3, and d_4 of the quadrilateral.
4. For each vertex n_i, its projection point n_i' on the corresponding fitted line is calculated. The vertex vectors $[n_1, n_2, ..., n_{|N|}]$ are projected onto the quadrilateral as $[n_1', n_2', \ldots, n_{|N|}']$. Then, perspective transformation is performed and the map region is mapped to the quadrilateral.
5. The height h and width w of a standard rectangular map frame are defined. The transformation matrix for projecting the quadrilateral's vertex matrix $[d_1, d_2, d_3, d_4]$ onto the standard rectangular frame $[[0,0],[w,0],[w,h],[0,h]]$ is calculated. Apply projection transformation, and map the map region to the standard rectangular map format, thereby restoring the original image to the standard map style.

The map image after registration becomes a standard rectangular map frame, as shown in Fig. 6. Finally, on the basis of the map sheet numbering of sectional maps, automated map stitching is executed in the script program, avoiding the substantial graphic computation requirements involved in image reading, display, and manual operations, thus completing the fully automated map sheet stitching process.

4. Implementation Verification

In the project of unified natural resource confirmation and registration, the automated base map preparation method proposed in this study has been implemented and verified. The scheme involved stitching together 928 historical sectional topographic maps of Beijing to form a base

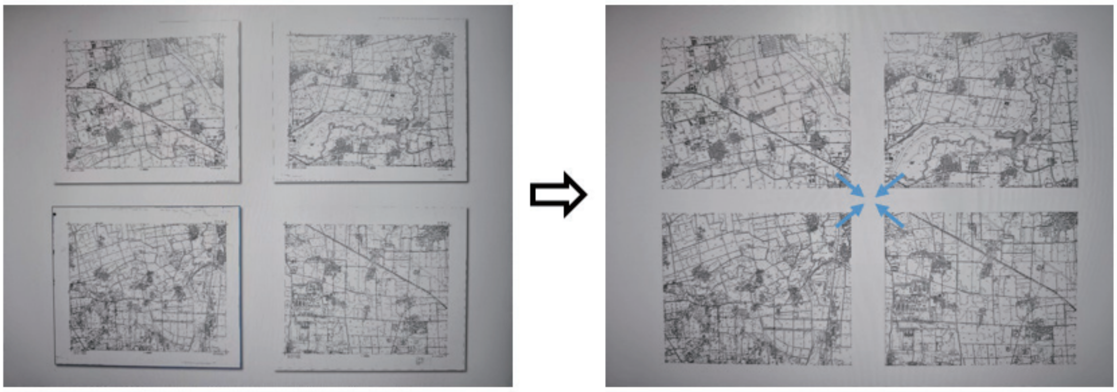


Fig. 6. (Color online) Map segmentation and registration.

map of the natural resource distribution in Beijing. On the basis of the pretrained U2Net-p model capability, 928 sectional maps were processed into standard map sheet formats.

In the base map stitching process, a five-digit coding rule was applied for sectional map numbering, structured as A–B–C, where A represents the region code, using a one-digit code, with values ranging from 1 to 4 to divide the Beijing map into four regions horizontally and vertically; B and C represent the row and column numbers in the respective regions, respectively, using two-digit codes with values ranging from 01 to 99. With this rule, the program is able to automatically identify and stitch sectional topographic maps. The effect of stitching is shown in Fig. 7; owing to resolution limitations, the full map cannot display the stitching effect in detail. Therefore, a portion of the full map has been extracted, and one specific area has been enlarged to clearly show the stitching result. The enlarged area corresponds to the sectional maps depicted in Fig. 6.

To better evaluate the quality of stitching, we adopted the method of marking anchor point pairs for evaluation. In adjacent sectional maps to be stitched, several prominent corresponding points were manually selected as anchor point pairs, as shown in Fig. 8. After automatic stitching was performed, the pixel displacement of the corresponding anchor points on both sides was detected. If the displacement fell within three pixels, it was considered as aligned; otherwise, as misaligned. Two indicators were defined to quantitatively evaluate the stitching effect of the maps: (1) stitching accuracy and (2) average pixel displacement variance. Stitching accuracy is defined as the ratio of the number of aligned anchor points to the total number of anchor points.

$$A = \frac{|P_a|}{|P|} \times 100\% \tag{2}$$

Here, P represents the set of anchor point pairs and P_a the set of aligned anchor point pairs. Stitching accuracy determines the quality of the stitching effect without any systematic bias. The higher the accuracy, the higher the usability of the stitched base map.

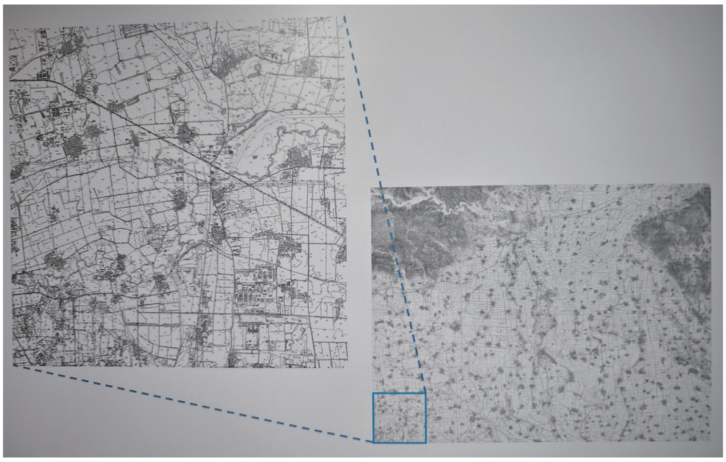


Fig. 7. (Color online) Map segmentation and registration.

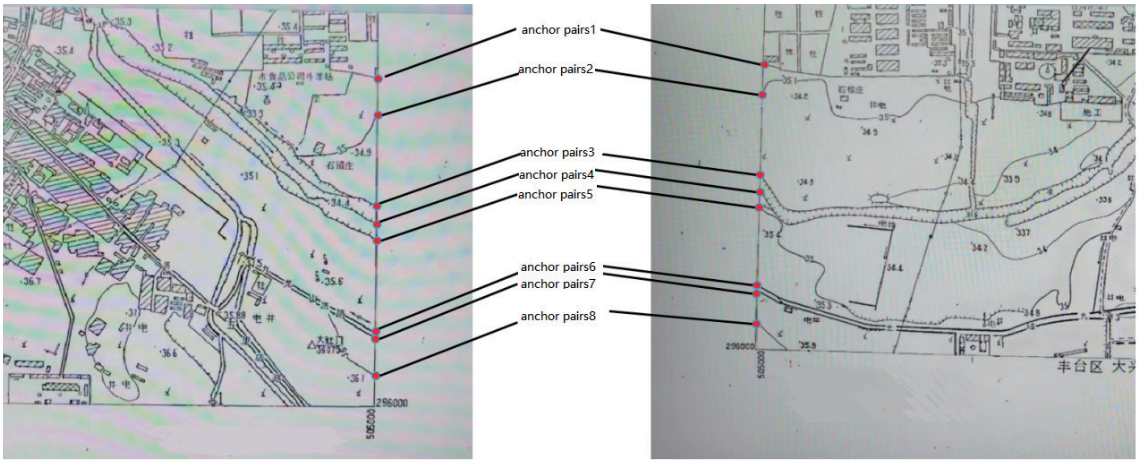


Fig. 8. (Color online) Marked anchor point pairs on adjacent sectional maps.

When measuring the pixel displacement of anchor points, the left and bottom edges of the images to be stitched are regarded as benchmarks, with upward and leftward displacements recorded as positive, while downward and rightward displacements recorded as negative. We recognize that the displacement of anchor point pairs may result from systematic errors during map image scanning processes or AI detection, leading to a consistent displacement direction and a similar displacement amount. Such systematic errors can be corrected through further pixel filling and translation transformation. Thus, we mainly focused on the chaotic displacement without significant characteristics. To compare the displacement degree caused by different interpolation methods, we used the variance of pixel displacement as a statistic to avoid influence from the mean value and reflect differences among pixel displacement data to the greatest extent. For all anchor points on the same edge, the variance is calculated, and the average pixel displacement variance is defined as the mean variance of pixel displacement on all stitched edges.

$$D_{avg} = \frac{1}{|E|} \sum_{e \in E, d \in e} \frac{1}{|d|} \sum_{i=1}^{|d|} \left(\delta_i - \overline{\delta} \right)^2 \tag{3}$$

Here, E is the set of edges to be stitched, $e \in E$ is an edge to be stitched, $d \in e$ is the set of all anchor points on edge e, δ_i is the displacement of the ith anchor point on edge e, and $\overline{\delta}$ is the average displacement of anchor points on edge e.

To compare the effect of the automated map stitching method with that of manual processing, 100 sectional maps from the same batch were manually stitched into a 10×10 map, and the stitching accuracy and average pixel displacement variance were calculated using the same method. The comparison results with the automated scheme are shown in Table 2. The stitching effect is shown in Fig. 9; the automated processing method avoids many operational errors during manual processing. Therefore, the values of stitching accuracy and average pixel displacement variance are both higher than the manual processing results, indicating that the stitching accuracy of the automated scheme is significantly superior to that of manual processing.

Table 2
Manual and automated scheme comparison.

	Dataset preparation and model training	Image segmentation and registration	Map stitching	Stitching accuracy	Average pixel displacement variance
Automatic processing	20 days	0.3 s/image	0.45 s	96.67%	106
Manual processing	—	30 min/image	7.3 h	73%	2186

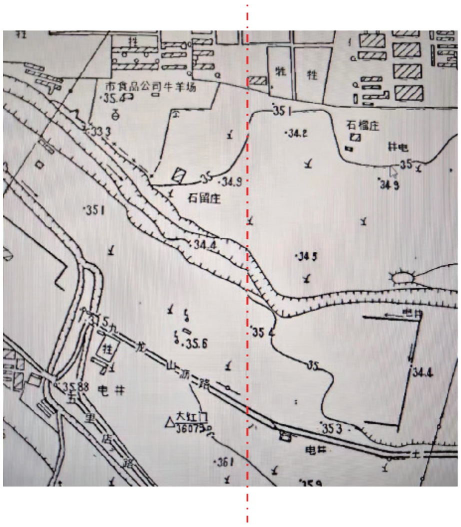


Fig. 9. (Color online) Part of stitched map.

5. Conclusions

On the basis of computer vision technology, we designed the automatic fusion scheme of the sectional map and carried out the implementation verification in the natural resources unified ownership registration project. On the basis of the semi-automatic map data annotation method, we prepared the map dataset and trained the U2Net-p model of map foreground segmentation. The mIoU of the model reached 98.37%. By using the model's accurate recognition ability, the automatic base map splicing process was further designed and realized, and the splicing performance was evaluated through the annotation of anchor points. The implementation result showed that the splicing accuracy reached 96.67%, compared with the 73% accuracy of manual processing. Compared with traditional manual processing, the automatic base map fusion method proposed in this paper significantly enhances efficiency and reduces manual intervention and errors, ensuring the accuracy and consistency of the processing results.

Acknowledgments

This work was supported by the Beijing Institute of Surveying and Mapping and the Beijing Key Laboratory of Urban Spatial Information Engineering.

References

1 ESRI: https://desktop.arcgis.com/zh-cn/arcmap/latest/manage-data/creating-new-features/about-preparing-to-digitize-a-paper-map.htm (accessed June 2024).
2 M. Dodge, R. Kitchin, and C. Perkins: The Map Reader: Theories of Mapping Practice and Cartographic Representation (Wiley, Chichester, 2011) p. 115.
3 Q. Sun, Y. Kan, and Q. Xiao: Manage. & Eng. **6** (2012) 27.
4 S. Xie and X. Feng: Acta Geod. Cartogr. Sin. **4** (2004) 356. https://doi.org/10.3321/j.issn:1001-1595.2004.04.015
5 J. H. Uhl and W. Duan: Handbook of Big Geospatial Data (2021). https://doi.org/10.1007/978-3-030-55462-0_20
6 P. Buonora: e-Perimetron **4** (2009) 192.
7 D. J. Stewart: Prof. Geogr. **53** (2001) 361. https://doi.org/10.1111/0033-0124.00290
8 L. Hong, S. Pang, and M. Geng: Chin. Agric. Sci. Bull. **37** (2021) 6. https://doi.org/10.11924/j.issn.1000-6850.casb2020-0195
9 S. X. Yuan: Sci. and Technol. Dyn. of Surv. and Mapp. **1** (1992).
10 Y. Chen, W. Liu, and S. Liu: Geomatics & Spatial Inf. Technol. **37** (2014) 97.
11 K. Zhao and L. Jiang: J. Geo-inf. Sci. **18** (2016) 21.
12 X. Liang, X. Lin, J. Quan, and K. Xiao: Acta Electron. Sin. **48** (2020) 11.
13 A. M. Hafiz and G. M. Bhat: Int. J. Multimedia Inf. Retr. **9** (2020) 171.
14 X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand: Pattern Recognit. **106** (2020) 107404. https://doi.org/10.1016/j.patcog.2020.107404
15 O. Ronneberger, P. Fischer, and T. Brox: Proc. Med. Image Comput. and Comput.-Assisted Intervention—MICCAI 2015: 18th Int. Conf., Munich, Germany, October 5–9, 2015, Part III 18 (Springer International Publishing, 2015) 234–241.
16 S. E. Umbaugh: Digital image process. anal.: Hum. and Comput. Vision Appl. with CVIPtools (CRC Press, 2010).
17 S. H. Hou and B. L. Guo: Comput. Eng. **31** (2005) 3. https://doi.org/10.3969/j.issn.1000-3428.2005.15.026
18 O. Chum, J. Matas, and J. Kittler: Lect. Notes Comput. Sci. **2781** (2003) 236. https://doi.org/10.1007/978-3-540-45243-0_31