The University of Tokyo

博士論文

Automatic Digital Object Model Reconstruction from Optical Flow Field Based Dense Aerial Image Matching

（オプティカルフロー場に基づく高密度空撮画像マッチング
を利用したデジタルオブジェクトモデルの自動生成）

袁巍

Abstract

Nowadays more and more high-tech applications need the specific 3D models and information. Traditional photogrammetric methods may meet the demand of accuracy, but the computational efficiency and outputs are hardly accepted. In many applications, such as autonomous driving, evacuation planning and so on, can be taking more use of the 3D model containing the object label attribute, which defines it as a digital object model (DOM) in this paper. In that case, the purpose of this study is to propose new methods to solve some key technologies for automatically generating such kind of 3D models. In detail, the following three issues have be mainly focused on:

(1) How to automatically find the seed points in poor texture remote sensing images?

(2) How to fast generate pixel-wised 3D dense point clouds?

(3) How to precisely annotate the 3D point clouds?

In order to efficiently solve these bottleneck problems mentioned above, a graph theory based seed point matching method for texture-poor images is firstly proposed. The high ordered graph matching algorithm is utilized to solve the correspondence point identification. It concerned both the radiometric and geometric constraint to make the final result denser and evenly distributed. Second, an aerial image dense matching method based on optical flow field is proposed. The optical flow field is calculated by using some seed points, so as to determine the similar region between images to reduce the redundant computation. A coarse to fine matching strategy is utilized to refine the generated 3D point clouds. The experimental results have shown that the proposed image dense matching method can achieve the matching accuracy of a sub-pixel level. Compared with the widely used dense matching methods, such as PMVS, the efficiency is higher, and fully meets the 3D object reconstruction and automatic generation of DSM requirements, etc. At last a multi-constraints fully convolutional network for the annotation of 3D point clouds is proposed. The images

and DSM are utilized as input to extract features to make the point cloud classification more accurate. The comparison of different deep neutral networks has demonstrated that the proposed residual fully convolutional network (Res-FCN) can achieve higher accuracy for 3D point cloud annotation. The conditional random field is utilized as the post-processing to make the object edges smoother and clearer.

In general, the digital object model (DOM) is automatically generated by seed correspondence searching, dense image matching and 3D point cloud annotation. The annotated labels are inserted into the DSM as the 4^{th} attribute of each point. In that case the interested objects can be easily extracted through the label information searching instead of point cloud segmentation procedures. The proposed methods in this paper can efficiently solve the technical bottleneck problems in each step mentioned above. In practical applications, the 3D object reconstruction can be quickly completed based on the DOM.

Acknowledgements

First and foremost I want to give my sincere and honored thank you to my advisor Professor Ryosuke Shibasaki. In the past four years, he gave me a lot of help, not only by tutorial my research parts, also help me to adjust the life in Japan. He is very kind hearted and easy-going to communicate. Although he is very busy, I can easily get his help by all kind of communication ways. Every time when I have difficulties in preparing presentation and facing the Scholarship interview, he gave me his advice and helped me pass the gate and get the best results. Again I want to say thanks to him, without his insightful advices this doctoral dissertation would hardly been completed.

Also I want to thanks to my co-advisor, Associate Professor Yoshihide Sekimoto. Different from Professor Shibasaki, Sekimoto sensei give me another sight to treat the research target. His strict attitude to research impressed me a lot. And his precious advices to my dissertation help me learn more and study more.

Otherwise I want give my greatest thanks to my family and my friends. No matter what kind of trouble I encountered and no matter what difficulties I have been though, you always gave me encourage and accompany with me. Without your help, I can achieve nothing. Because of you, I think I was the luckiest guy in the world.

In the end, I want to give my thanks to Professor Jianya Gong, the dean of school of remote sensing and information engineering, Wuhan University. Without his recommendation I will not come to the University of Tokyo, without his help I will not get this ability to doing research in photogrammetry and remote sensing area.

Thanks to those who had given me advices and supporting me continue my studies people. Gratefully thank you.

Contents

Chapter 1
Introduction

This doctoral dissertation is focusing on the 3D information acquire approach by the aerial sequence images. We use graph theory based image matching method to find the tie point in image sequence, and propose a new approach of dense image matching to get the density point clouds from the aerial images. In particularly, we provide a new strategy of matching method and experiment it on the dataset of different land surface to test its robustness and reliability.

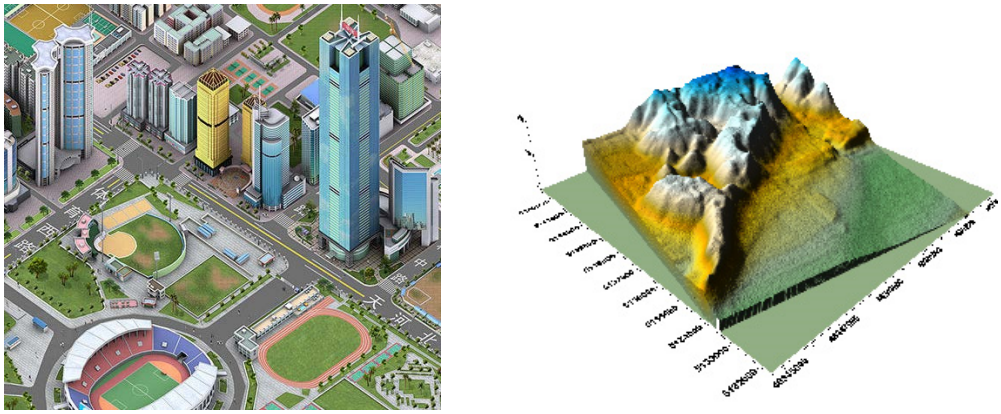
The following sections in this chapter is organized as follows: Section 1.1 introduces the background of this research, Section 1.2 briefly introduce our based idea and matching strategy, Section 1.3 summarizes the contribution of our research, At last, Section 1.4 outlines this dissertation.

1.1 Research background

On the research aspects, objective 3D reconstruction is a procedure that restoration the objects' 3 dimensional surface structure form 2 dimensional images. It has been a hot research topic in digital photogrammetry and computer vision field for many years (*Yuan and Ming, 2009*). It is mainly involved with dense matching approaches and three-dimensional reconstruction methods. Currently, many matured methods for reconstructions the objects' three-dimensional surface structure has been utilized in different kinds of applications. The key to improve all the art of work is to figure out how to acquire the 3D point clouds in a fast, reliable and effective way.

On the application aspects, nowadays more and more applications such as Google earth, car navigation system, 3D city maps, and DEM producing work, need the high density and high accurate 3D information. Compared with 2D information, 3D information is easier for people to identify and more familiar to the reality world. By using all kind of 3D maps just like Google map, people can easily find the destination

and understand the real location of their own, which brings a lot of convenience to people's ordinary life.



(a) (b)

Figure 1.1 Applications of 3D information. (a) 3D city map, (b) DEM (picture from www.baidu.com)

Meanwhile, 3D maps are more familiar with the real world (see Figure 1.1(a)). For normal people they do not need to understand the symbols in the old traditional maps. By using the 3D map they can easily find what they want in the searching area. On the other hand, DEM (see Figure 1.1(b)) shows a lot of land heigh informations to those who need this inforamtions. For example, in the management of recovery from the disaster and some urban planing field. They can use DEM to manage their budget and guide their constructions. Accurate 3D inforamtion is playing a important role in the real world. Confronting those demands, an effective 3D information acquire approach should be provided. And this is the motivation of our research.

1.2 Our approach

The approach in this doctoral dissertation aims at getting the 3D point clouds in a robust and effective way and identify it, meanwhile the economical consumption is also under our consideration. As we all know, traditional extraction methods of 3D point clouds are mainly devied in two kinds of approaches. The first one is using laser scanner to get the LiDAR point clouds, it is fast and accurate. But the device is expensive. Nowadays images are more and more easier to collected. And the price of

an camera is cheaper than a laser scanner. We can easily get a large area covered sequence images from one aerial photography. Which means the operation fee is lower. So we want to provide an effective and reliable matching approach to get the 3D information from the images. This is our based idea.

Traditional mataching approaches are mainly divied by two types, area based matching methods and feature based matching methods. Those kind of matching methods present a low efficiency and unreliable when dealing with the mission of per-pixel matching. Especially, when dealing with the aerial image the mismaching rate is very high. It is hardly utilized in the real applications (*Stefano et al, 2004*). Consider this situation, we provide a new strategy of matching approach. To meet the demand of per-pixel matching and the acuracy, the whole matching strategy is shown as Figuare 1.2.

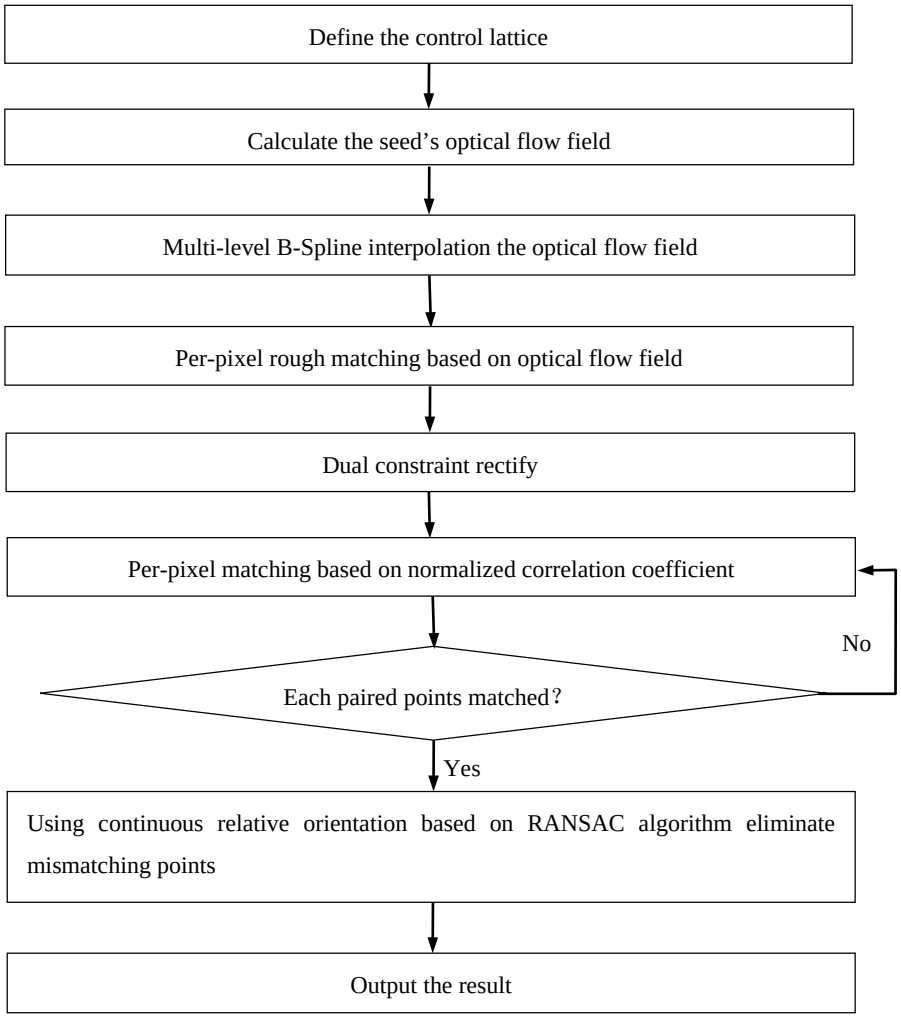


Figure 1.2 The matching straregy of our approach

Firstly, we use graph theory based matching method to match the raw stereo image pairs. Then using continuous relative orientation based on RANSAC to eliminate the mismaching points. Secondly we use the former result as the seed points to caculate the seed texure based optical flow field (OFF). Then using multi-level B-spline to simulation each pixel's relationship in the overlap area. Then build up the epipolar image between the image pairs, using correlation coefficients and texure coefficients to fine maching point pairs. In the end we will using continuous relative orientation based on RANSAC to eliminate the mismatching point pairs. After that we use the dense point clouds to generate the DSM for the classification step. The important algorithms and theories we will explain in chapters 3, 4 and 5.

1.3 Contributions

In this doctoral dissertation we provide 3 new approaches for solving the bottleneck problems of digital object model generation. Our main contributions can be summarized as follows.

- How to get the uniformed and accurate seed points in poor texture image area? We provide a graph theory based image matching method. It can use the structure information as a similarity measurement from different feature extractors. Therefore we can get a uniformed and accurate sparse matching result.
- How to get the 3D information from the real-world in a priority way? We provide an effective dense matching method to get the 3D point clouds from the aerial stereo image pairs. Therefore we can get the 3D information in a low consumption and fulfill the demand of applications.
- How to achieve an accurate point cloud identification results? We provide a new deep neural network, using both images and DSM as input to extract the features to segment the images. And using the segmentation results to label the generated 3D point clouds.

1.4 Outline

The following chapters are organized as follows:

- Chapter 2 gives a detailed introduction on the state of art researches on matching methods and annotation methods. In this chapter we focus on introducing the existed methods' advantages and shortages, and our research motivation.
- Chapter 3 describes the basic concepts and techniques using in extracting the image tie points. We will describe the theory and processing methods, and experimental results.
- Chapter 4 describes the basic concepts and techniques using in dense image matching. We will describe the theory and processing methods, and experimental results.
- Chapter 5 describes the basic concepts and algorithm using in 3D point clouds' annotation. And we describe the experimental result and analysis. Comparison experiments between our approach and some exist methods are taken to show advantages and disadvantages.
- Chapter 6 concludes this doctoral dissertation by highlight contributions and pointed out the shortage of our algorithm. Also future work will be described in this chapter.

Chapter 2
Literature review

The essence of image matching is an essential searching problem. So as to searching, a searching target library should be established first. Secondly the key characters and features of the target should be expressed in a priority issue. Then a searching procedure can be operated as the guidance by a searching strategy which obeyed by the certain similarity rules. If the searching target has some kind of relationship with the library established, we can define the similarity measures to find out the result.

For image matching, the interest point should be feature extraction firstly. It is the same procedure as establish the searching target library. Then we can search the matched point pairs in interest points by calculating the similarities among those interest points. Currently, the widely used matching methods are divided into two kind of types. One is area-based iamge matching method, the other is feature-based image matching method.

For point cloud annotation, traditional methods are always based on machine learning technology. In recent years, a lot of deep learning methods have shown great results in image classification competitions, which give us a motivation to utilize the new technology in our own research.

In this chapter we will introduce two types of matching methods and make a comparison to describe our motivation for providing the new approach. Also we will briefly introduce the widely used machine learning and deep learning methods for 3D point cloud annotation.

2.1 Image seed point extraction

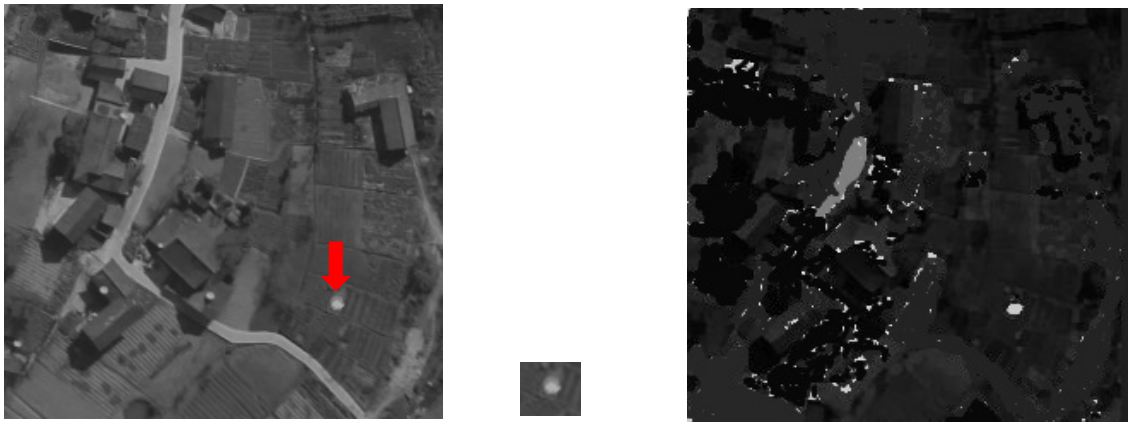
2.1.1 Area-based image matching

Area-based iamge matching algorithms are highly developed in the last decades. They

are the most theoretically sophisticated matching methods. Area-based iamge matching methods are always select the corresponding searching area between the stereo image pairs, then using relevant metrics to determine the similarity of the tow image block to locate the center of the image blocks as the matched point pairs. There are lots of matured measurements. For example, cross-correlation (*Welch, 1974*), sum of absolute differences, sum of squared difference (*Barnea and Silverman, 1972*) and Fourier correlation (*Bracewell, 1965*), those methods are widely used for image matching. In this section, we are focusing on introduce the cross-correlation method and least squares image method. These two methods are most essential and widely used algorithms in digital image processing.

2.1.1.1 Cross-correlation matching

Cross-correlation measurement is a widely used similarity measurement. By calculating the correlation index between the template window and target image and comparing the correlation index to the threshold to determine the center of the two image blocks are corresponding point pairs (*Gonzalez, 2004*), which shows in Figure 2.1.



(a) (b) (c)

Figure 2.1 The correlation between the template and the searching image. (a) searching image, (b) template, (c) the coefficient shows in image

The correlation index between the $m \times n$ template b and the searching image a can be described as follows.

$$c(x,y) = \sum_i \sum_j b(i,j)a(x+i, y+j) \tag{2.1}$$

where $c(x,y)$ is the correlation index; $b(i,j)$ is gray of the pixel (i,j) in template b; $a(i,j)$ is gray of the pixel (i,j) in the searching image a.

Because the template b and image a have always brightness changes and scale changes, normalized cross-correlation index are utilized to reduce the influence by those factors.

$$\sigma(x,y) = \frac{\sigma_{ab}}{\sigma_a \sigma_b} \tag{2.2}$$

where $\sigma(x,y)$ is the normalized cross correlation index; σ_{ab} is the covariance of image patches; σ_a is the standard deviation of searching window; σ_b is the standard deviation of template.

So we can determine the normalized correlation as follows:

$$\sigma(x,y) = \frac{\sum_i \sum_j [b(i,j) - \bar{b}][a(x+i, y+j) - \bar{a}]}{\sqrt{\sum_i \sum_j [b(i,j) - \bar{b}]^2 \sum_i \sum_j [a(x+i, y+j) - \bar{a}]^2}} \tag{2.3}$$

where $\bar{a}$ and $\bar{b}$ are stand for the average gray of the searching window and the template window, respectively.

When dealing with the aerial images, the image size are large. In order to reduce the calculation, we often use the under formula to calculate the correlation index.

$$\sigma(x,y) = \frac{\sum_i \sum_j [b(i,j)a(x+i, y+j)] - \frac{1}{mn} \sum_i \sum_j b(i,j) \sum_i \sum_j a(x+i, y+j)}{\sqrt{\sum_i \sum_j [b(i,j) - \bar{b}]^2 \sum_i \sum_j [a(x+i, y+j) - \bar{a}]^2}} \tag{2.4}$$

It is easy to find out that the normalized correlation index is invariant to the image template and the searching window's linear gray transformation. However, it is sensitive to the rotation and scaling. If the large rotation and scaling exits, correlation methods cannot acquire the accurate result. On the other hand, correlation coefficient

is related to the SNR of the image. It shows a great impact with the image noise, especially when the template and searching window has small gray contrast. The correlation coefficient is more susceptible to the noise. In order to acquire the better matching result, normally the interest points with large gray contrast should firstly be extracted. Then image de-nosing and distortion correction procedures should be applied. When all those image preprocessing has been done, the result quality will be improved.

2.1.1.2 Least square matching

Least square matching method is the most classic method in area-based image matching. It uses all aspect of information in the searching window and template. It also concerns about the geometric distortion, radiation distortion, and random error between the stereo image pairs, which makes the matched results achieve sub-pixel or even super-pixel accuracy. Consider about the cross correlation matching method, we can find it uses the similarity measurement to find the most similarity image blocks, and determines the center of the blocks as the corresponding point pairs. So its accuracy just achieves the pixel level. This is why we always use least square matching method to improve the matched result accuracy.

In this section, the basic theory of least square matching method will be introduced. One point least square matching method only consider about geometric distortion and radiation on the image coordinates. As the photographing position and the ground evaluation changes, the geometric distortion and the radiation distortion always exist between the template and the searching windows. Therefore the geometric distortion and radiation distortion correction should be taken before the correlation coefficient calculation. Equation (2.5) shows the basic equation of the least square image matching method.

$$g_1(x,y)+n_1(x,y)=h_0+h_1 g_2(a_0+a_1x+a_2y, b_0+b_1x+b_2y)+n_2(x,y) \qquad (2.5)$$

Where g_1, g_2 are the pixel gray of left and right image, respectively; n_1, n_2 are the noise of the left and right images, respectively; h_0, h_1 are the radiation distortion correction parameters; $a_0, a_1, a_2, b_0, b_1, b_2$ are the geometric distortion correction parameters. We linearization Equation (2.5) by Taylor's series expansion with the condition shown as follows:

$$h_0 = 0, h_1 = 0, a_0 = 0, a_1 = 0, a_2 = 1, b_0 = 0, b_1 = 0, b_2 = 1$$

The error equation of the least square matching method can be written as follows:

$$v = dh_0 + g_2 dh_1 + \frac{\partial g_2}{\partial x} da_0 + x\frac{\partial g_2}{\partial x} da_1 + y\frac{\partial g_2}{\partial x} da_2 + \frac{\partial g_2}{\partial y} db_0 + x\frac{\partial g_2}{\partial y} db_1 + y\frac{\partial g_2}{\partial y} db_2 - \Delta g \tag{2.6}$$

Where Δg is the corresponding pixels gray difference between the left image and the right image. In image processing each pixel gray is a discrete grid array. So we can use differential index instead of the partial derivative in Equation (2.6).

$$\begin{aligned} \frac{\partial g}{\partial x} &= \frac{1}{2}[g_2(i, j+1) - g_2(i, j-1)] \\ \frac{\partial g}{\partial y} &= \frac{1}{2}[g_2(i+1, j) - g_2(i-1, j)] \end{aligned} \tag{2.7}$$

Then we can build the matrix form of each pixel error equation as follows:

$$\mathbf{V} = \mathbf{C}\mathbf{x} - \mathbf{l} \tag{2.8}$$

where

$\mathbf{x} = \begin{bmatrix} dh_0 & dh_1 & da_0 & da_1 & da_2 & db_0 & db_1 & db_2 \end{bmatrix}^{\mathrm{T}}$;

$\mathbf{C} = \begin{bmatrix} 1 & g_2 & \frac{\partial g_2}{\partial x} & x\frac{\partial g_2}{\partial x} & y\frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} & x\frac{\partial g_2}{\partial y} & y\frac{\partial g_2}{\partial y} \end{bmatrix}$;

$\mathbf{l} = \Delta g$.

We can determine this equation with least square adjustment principle as follows:

$$\mathbf{x} = (\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{l} \tag{2.9}$$

We put the initial value into Equation (2.6), we can ultimately derive the distortion correction parameters by doing several iterations. The whole strategy of least square matching method is shown as Figure 2.2.

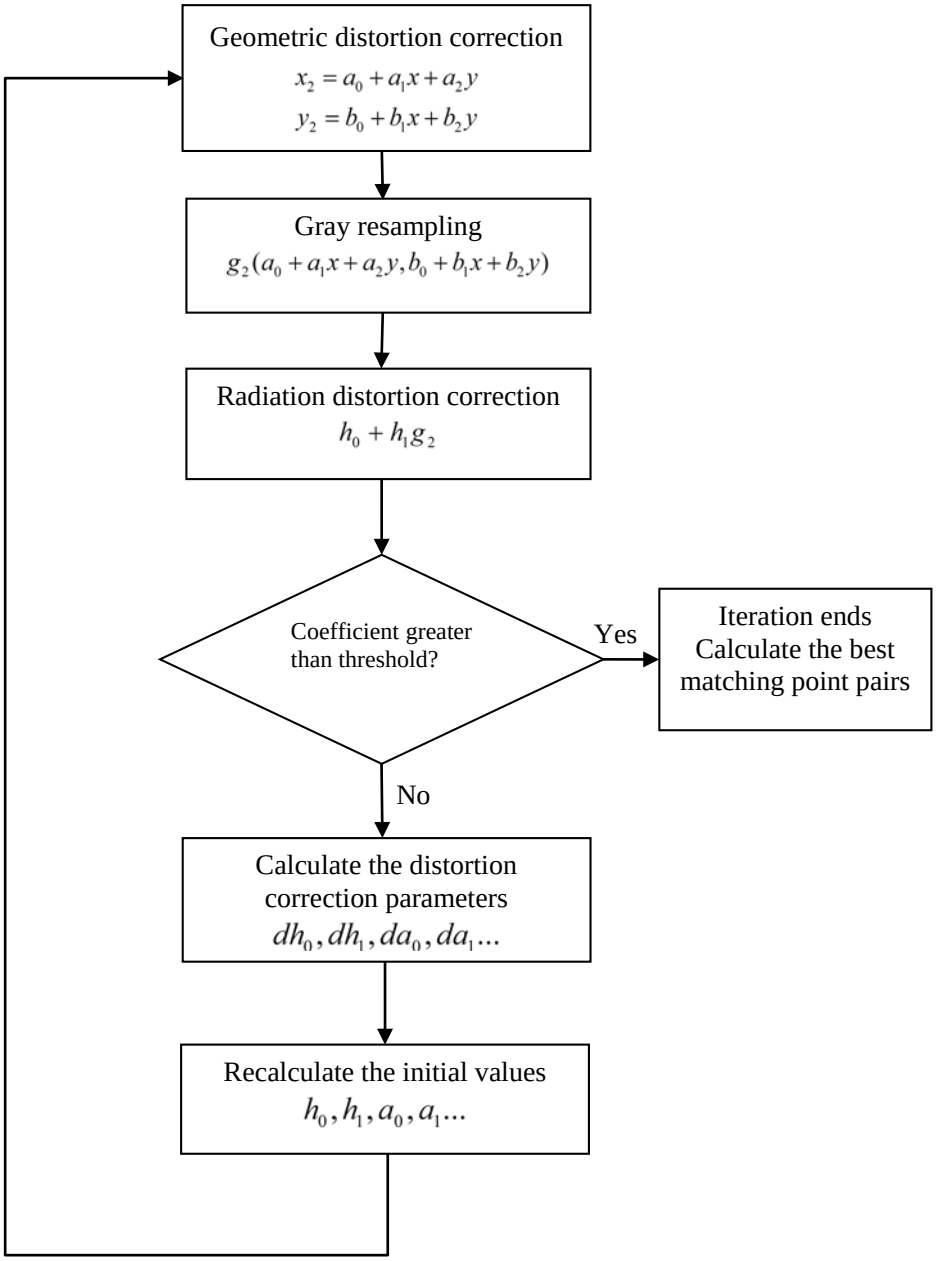


Figure 2.2 The strategy of least square matching method

2.1.2 Feature-based iamge matching

Feature-based matching always involves with feature extraction and feature matching. Different from area-based matching methods, feature-based matching methods concern about the local gray structure from one image, then extract the feature area or points presented by a group of parameters or rules. In the end, by comparing the feature parameters to determine the corresponding point pairs.

In this section, the feature-based matching methods will be introduced by two steps, first is feature point extraction, second is feature point matching. After that the widely used SIFT algorithm will be described.

2.1.2.1 Feature extraction

Feature extraction is a procedure to find out the interest point and area in one image. On image processing and digital photogrammetry field a good feature extraction operator may have those abilities: certainty, robustness, invariance, uniqueness and comprehensibility. Meanwhile, the good feature's distribution should not be too much concentrated.

Here we introduce some widely used point feature detectors, such as Moravec, Harris, and Förstner.

Moravec detector's basic assumption is each feature point has a great gray contrast on all directions. By calculating the searching window's gray gradient on vertical direction, horizontal direction and diagonal direction, determine the maximum and minimum gray variance points as the feature points. The feature detecting time is short, algorithm is simple. But it has low robustness when dealing with the image noise, rotation and contrast change (*Moravec, 1977*).

Harris detector is the improvement of Moravec detector, it calculate the local image block's differential by Gaussian template. The eigenvalues of the self-correlation matrix is the first order curvature of the self-correlation function. If one pixel's eigenvalues are large, then it determined to be the feature point. The advantage of the Harris detector is simple calculation and the distribution of the features is uniform. So engineers can extract the feature points in accordance with their need (*Harris and Stephens, 1988*).

Förstner detector is determined by calculating the 5×5 window's gray covariance matrix and each pixel's Robert gradient, the center pixel is (x, y), searching the nearly circular error ellipse in the image and select the minimum point as the feature point

(*Förstner and Gülch, 1987*). It has high self-adaptability and accuracy. When selecting the candidate points, the threshold is determined by calculating the mean weight of all the pixels. That means we cannot operate these two steps at the same time. This will influence the computation efficiency. Meanwhile, in practical applications we can hardly set the threshold based on empirical functions.

2.1.2.2 Feature matching

After the feature extraction, we may acquire a lot of feature points to be matched. Usually we extract features in both images on a stereo image pair. Then compare all the features to determine the corresponding point pairs. To reduce the pairing time, some data structure techniques are used like KD-tree (*Moore, 1991*).

2.1.2.3 SIFT matching

Scale invariant feature transform (SIFT) is first proposed by Lowe in 1999, then he developed it in 2004. The basic concept of SIFT is Lindeberg's space scale theory (*Lindeberg, 1993*). SIFT feature has several advantage, it shows a great robustness when dealing with image rotation, scaling, brightness change. Also show less affection with noise and affine distortion.

The completed SIFT feature matching approach involved with 5 steps: scale extreme detection, key point selection, generate the primary direction, feature describe and feature matching.

In this subsection, those five steps will be briefly introduced.

1. Scale extreme detection

Scale extreme detection involves with 3 steps, first one is Gauss pyramid images generate; second one is DOG (Difference of Gauss) producing; the last one is judgment of the extreme value of the neighborhoods.

As Lindeberg proved, scale normalized Gauss-Laplace kernel is the unique scale-invariant kernel. In order to reduce the calculation, he used the differential Gauss function to approximate the Gauss-Laplace function. It can be described as Equation (2.10).

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{2.10}$$

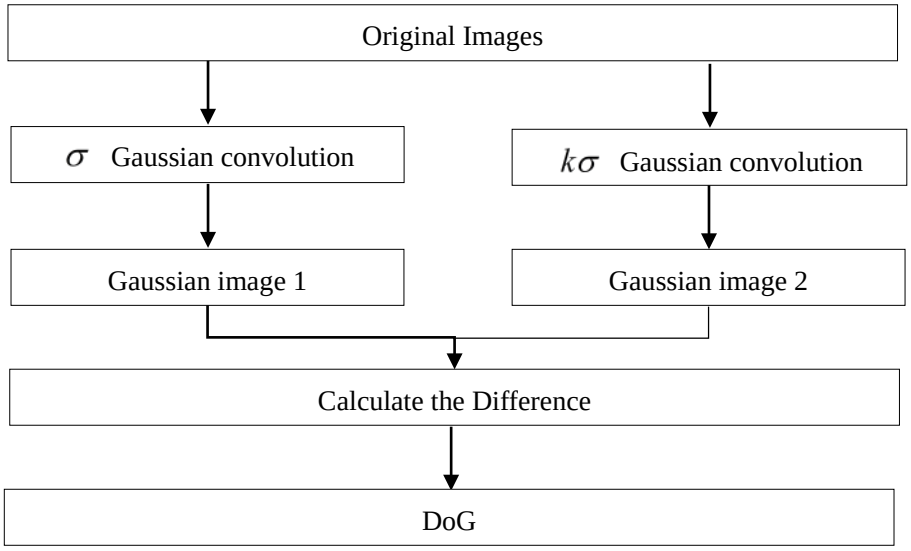
where *k* is the constant variable, when *k* equal to 1 we can change the approximate to equation. Normally $k = 2^{1/3}$; σ is the scale coefficient. It represents the window size of the Gaussian convolution. Meanwhile the window size of Gaussian convolution and the interest points' numbers are in positive correlation, which contrasts with the computation time. In Lowe's experiments he suggests to set the σ value to 1.6; G is the gray function of the image; ∇^2 is Laplace function. By deriving Equation (2.10) we can get Equation (2.11).

$$(k-1)\sigma^2\nabla^2 G \approx G(x, y, k\sigma) - G(x, y, \sigma) \tag{2.11}$$

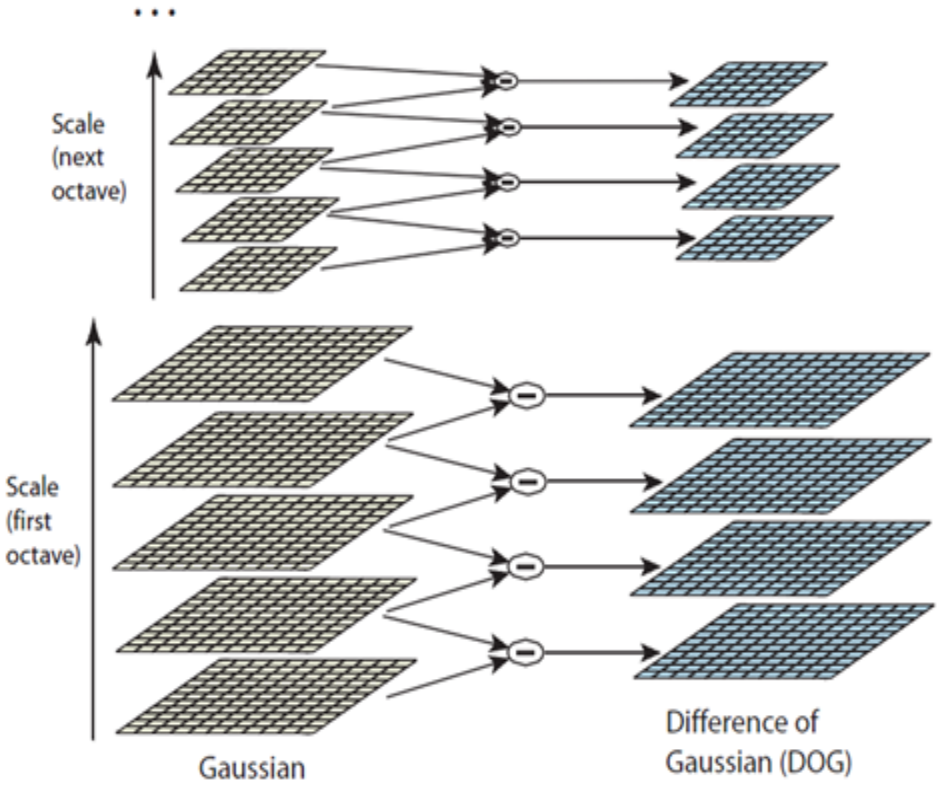
This is why Lindeberg used the differential Gauss function to approximate the Gauss-Laplace function. So we can obtain one image's represents by calculating its DOG in the adjacent scales (see Figure 2.3).

In theory, the numbers of the space scale are infinity, which means the number of the DoG is unlimited. Through a large number of experiments, Lowe proved that the small scaled image is the fine expression of the related large scaled image. But the number of the features is certainty. When scale change to a certain level, the feature number won't show a significant increase. Meanwhile σ value also affect to the features number. When σ equal to 1.6 the feature number will present in a stable level. Because of these, Lowe suggests to set the scale value as 3 and σ value as 1.6.

After DoG generating, we can easily detect the interest points as Figure 2.4 shows. By selecting the extreme value points between adjacent scales, the interest points are determined.



(a)



(b)

Figure 2.3 The DoG generation. (a) the strategy of DoG producing, (b) how to generate DoG (***Lowe et al, 2004***)

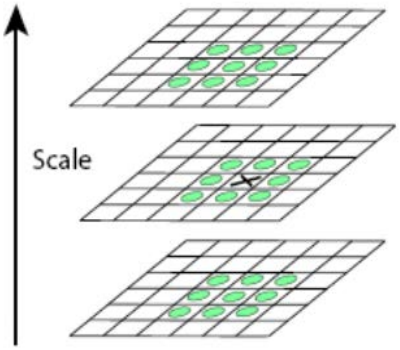


Figure 2.4 Interest point detection.

2. Key point selection

Key point selection is the procedure to reduce the unstable interest points. When the interest point detection are finished, a lot of interest points will be detected near the edges and line features in the image. Because of Gauss-Laplace function shows a strong line and edge response, most of these interest points are unstable. In order to improve the matching quality, those points should be removed.

Firstly, the accurate location of the interest points should be determined. By Taylor expansion the scale functions as follows:

$$D(x,y,\sigma) = D(x,y,\sigma) + \frac{\partial D^{\mathrm{T}}}{\partial x} + \frac{1}{2} X^{\mathrm{T}} \frac{\partial^2 D}{\partial x^2} X \tag{2.12}$$

By deriving the partial derivative of Equation (2.12) and calculating the extreme value on the condition (2.12) equal to 0, we can get the accurate location as $\hat{x}$:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{2.13}$$

By inserting Equation (2.13) to Equation (2.12) and only remain the first 2 sections, we can obtain Equation (2.14).

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X} \tag{2.14}$$

Equation (2.14) is very important. It represents the one point gray in the DoG. As Gaussian convolution has a strong response to the gray changes in the image. If image area has a large gray contrast, $D(\hat{X})$ will be a large absolute value. Lowe suggest to

set a threshold that $\left|D(\hat{X})\right|$ larger than 0.3 to obtain the stable feature points. Obviously, when the image area has low gray contrast and the texture are similar, we cannot obtain more stable feature points.

Otherwise, the line area and edge area are always the high gray contrast area in an image. They are sensitive to the image noise. So we should also remove these interest points.

As we all know Hessian matrix is an efficient method to detect the edge. The elements in Hessian matrix are the pixel gray in DoG image. It is easy to obtain. Hessian matrix described as follows:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{2.15}$$

Assume the α is the maximum eigenvalue of Hessian matrix, β is the minimum one then we can calculate as follows:

$$\begin{aligned} \mathrm{Tr}(\mathbf{H}) &= D_{xx} + D_{yy} = \alpha + \beta \\ \mathrm{Det}(\mathbf{H}) &= D_{xx} \cdot D_{yy} - (D_{xy})^2 = \alpha\beta \end{aligned} \tag{2.16}$$

Assume that $\alpha = \gamma\beta$ then

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{\alpha + \beta}{\alpha\beta} = \frac{(1+\gamma)^2}{\gamma} \tag{2.17}$$

We determine the edge points when the ratio of Hessian matrix's maximum eigenvalue and minimum value is larger than a threshold. Normally, if $\text{ratio} > \frac{(1+\gamma)^2}{\gamma}$ then we remove this point. Lowe suggests the threshold is 10.

After removing gray contrast and edge extreme value, the selected interest points will be the final candidates of feature points.

3. Primary direction calculation

After step 2, we can acquire a lot of feature points. In order to make each feature has the rotational invariance. Every point's primary direction should be calculated.

Each pixel's gradient can be calculated as follows:

$$grad(I(x,y)) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \tag{2.18}$$

The gradient magnitude can be calculated as follows:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{2.19}$$

The gradient direction can be calculated as follows:

$$\theta(x,y) = \arctan \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{2.20}$$

Where L is the Gaussian image which has the same scale as the feature points; (x, y) is the feature point's pixel coordinates in Gaussian images.

The localized gray gradient histogram in Gaussian image has 36 columns. Each column covered 10° of 360°. As the contribution of the gradient to feature point is decrease by the distance. We should weight each gradient by Gaussian function. And avoid the gradient histogram mutation.

4. Feature describe

Feature describe is the procedure to describe the feature points by unique parameter sequence. Described features can easily be identified, also it is unique. Which means the feature descriptor is the identification of each feature.

After each feature's primary direction has been calculated, we rotate the localized pixel to the primary direction to recalculate the feature descriptor. In this way we can ensure the invariance of the feature.

Firstly we select a window of 16×16 pixels as to calculate the feature descriptor. Secondly it is divided into 4 sub-rectangle, every sub-rectangle has 4×4 pixels. After that we calculate each pixel's gray gradient and direction in sub-rectangle, weighted by Gaussian distance. Then we uniform the gradient direction in 8 directions, each direction covered 45° area. In the end, we accumulate the gradient magnitude to the correction direction to obtain an 8 dimensional vector. According to this we can get the final SIFT feature descriptor as a 128 dimensional vector (see Figure 2.5).

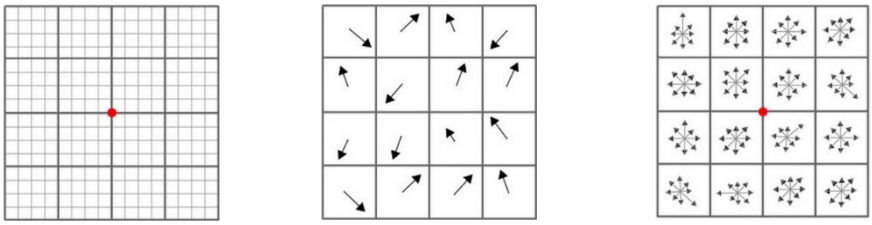
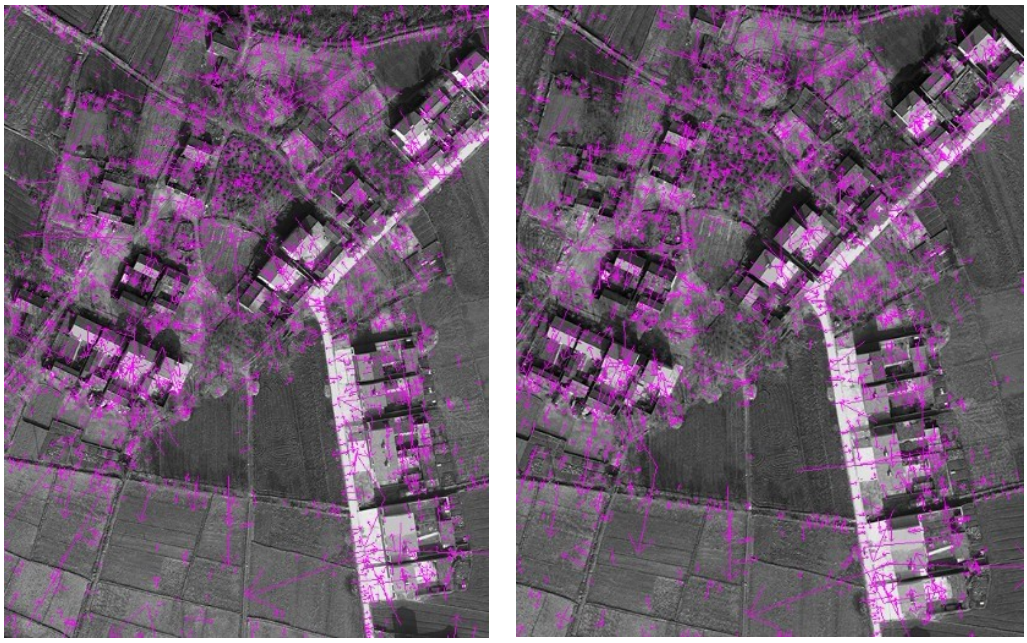


Figure 2.5 SIFT feature descriptor

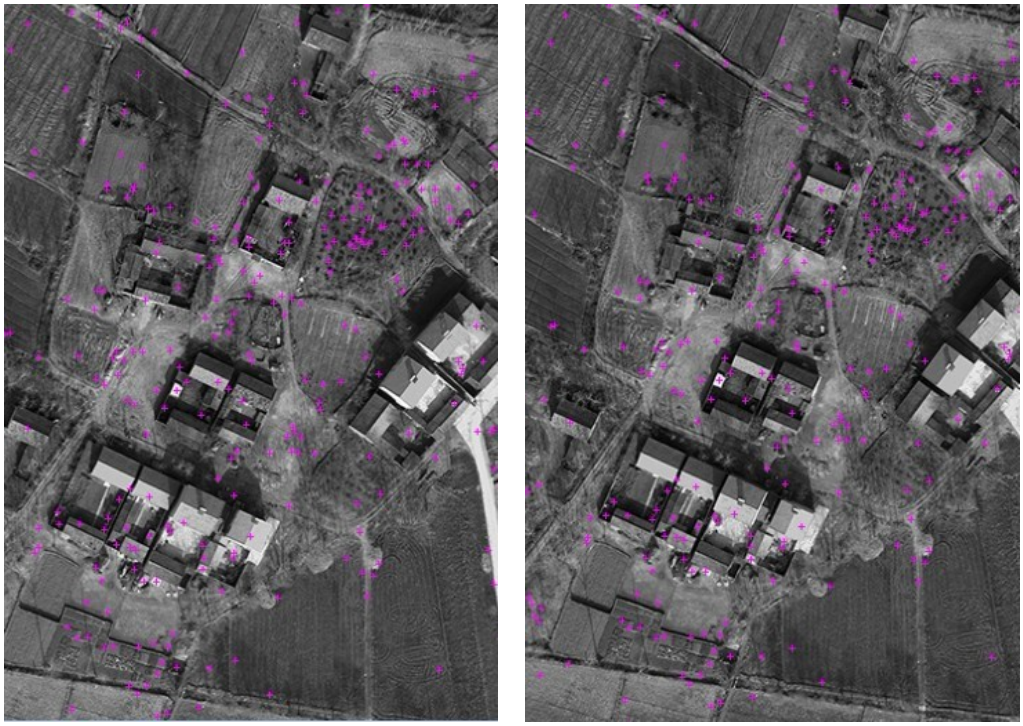
We can easily find out that the SIFT feature descriptor by a close relation with the local area's texture. If the local texture has strong repeatability, the descriptor's distinguishability is poor. It may cause large numbers of mismatching. We should use some effective error detection methods to remove these mismatching points.

5. Feature matching

The feature matching procedure is very simple as I have mentioned in section 2.1.2.2. SIFT feature matching is to find out the corresponding feature point pairs which has the smallest Euclidean distances between the stereo iamge pairs. Normally KD-tree is used to find the corresponding points. Figure 2.6 shows the SIFT feature and matching result.



(a)



(b)

Figure 2.6 The SIFT matching. (a) SIFT features between stereo image pairs, (b) The SIFT matching result

From Figure 2.6 we can obviously find out that SIFT features are concentrate at the rich texture region. At poor texture region, SIFT features are less. This is the shortage of SIFT matching.

2.1.3 Comprison of the widely used matching methods

In this sub-section, we make a simple comparison of the widely used matching methods on their ability of computation efficiency, rotation invariance, scale invariance, descriptor uniqueness.

1. Cross-correlation method

Obviously, correlation methods cannot deal with image rotation and scaling. Assume the template and the searching window is perfect matched. The correlation coefficient of the two image blocks is 1. When rotate the searching window in 45°, the coefficient

will be less than 1. As we introduced in section 2.1.1.1, the cross-correlation methods are sensitive to the image noise. When dealing with the lack of texture region, the correlation coefficient is inaccurate.

2. Least square matching method

Least square matching methods concerns about the geometric distortion, radiation distortion, and random error between the stereo image pairs, so different from correlation methods, the result of least square matching methods is in high accuracy. But it needs to give good initial values, which means the initial result's accuracy determine the least square adjustment convergence.

3. SIFT matching method

As we described in section 2.1.2.3, SIFT matching method shows a great robustness when dealing with image rotation, scaling, brightness change. However, SIFT features are concentrate at the texture-rich region. At texture-poor region, SIFT features are less.

We conclude our comparison in Table 2.1.

Table 2.1 Comparison of image matching methods

	Rotation invariance	Scaling invariance	Initial parameters	Descriptor uniqueness	Computation efficiency
Cross-correlation	×	×	×	×	√
Least square matching	√	√	○	×	○
SIFT	√	√	×	√	×

Ps: × stands for have not; √ stands for have and quality is high; ○ stands for have but quality is bare.

2.2 Three-dimensional point cloud extraction

2.2.1 LiDAR

Light detection and ranging (LiDAR) is a widely used 3D point clouds generation system in recent decades (*Wehr et al, 1999*). It can be divided into two categories: airborne and ground. The airborne laser radar is an airborne laser detection and ranging system installed on an aircraft, which can measure the 3D coordinates of ground objects. Airborne LiDAR is an active earth observation system. It integrates laser ranging technology, computer technology, inertial measurement unit (IMU)/DGPS differential positioning technology, which has made a major breakthrough in real-time acquisition of 3D spatial information. Compared with traditional photogrammetry measurements, it has the advistage of high automation, small weather impact, short data production cycle and high accuracy. However, using LiDAR system to generate 3D point clouds is much more expensive than traditional aerial photogrammetry system. In practical application and production, people usually select to use images for 3D point cloud generation.

2.2.2 Dense image matching

In past decades, dense image matching has drawn extensive attention from the fields of photogrammetry and computer vision, and has undergone substantial development (*Rothermel et al, 2011; Remondino et al, 2014*). Dense image matching algorithms can be divided into two kind of types, binocular-stereo matching and multi-view image matching. The binocular-stereo matching strategy is mostly used in the field of photogrammetry, in which the correspondences are determined by taking the geometric and radiometric constraints between stereo image pairs into consideration (*Torresani et al, 2013*). Binocular-stereo dense matching usually includes four steps (*Scharstein et al, 2001*): matching cost computation, cost aggregation, disparity optimization and disparity refinement. Based on the different cost aggregation

methods employed, dense image matching can also be divided into two categories, specifically: local algorithms (*Ke et al, 2004*) and global algorithms (*Tran et al, 2006; Issac et al, 2014*). The local algorithms, determine the correspondences by calculating the matching costs between the selected point and its surrounding local neighborhoods, and then use the winner takes all (WTA) strategy to select the point with the minimum matching cost as the final corresponding point (*Tola et al, 2008*). Since the local algorithms only use a part of the local neighbors for calculation, this leads to a low computational complexity and redundancy. However, they can easily become stuck in local optima, so that the matching result does not match the true topography. The global algorithms use pixel-based or object-based cost functions optimized by the energy function using graph cuts or Markov random field (MRF) methods in order to make the final matching result reach a global optimum (*Hirschmuller et al, 2008*). As these kinds of method take the whole image into consideration, the matching precision is higher than that of local algorithms, but it suffered from substantial numbers of redundant computations, resulting in low matching efficiency. Hirschmüller (*2008*) proposed a Semi-Global Matching (SGM) algorithm that improves the computational efficiency through multi-directional dynamic programming; compared with traditional global algorithms and local algorithms, only the non-occluded points are considered during the image matching process, and both the matching accuracy and efficiency are further improved (*Hirschmüller et al, 2009*). Although the binocular-stereo dense matching method has the advantages listed above, it only takes account of the information contained in two images, so its matching results display poor robustness to occlusion and noise (*Remondino et al, 2008*).

In the field of computer vision, multi-view matching method has always been a hot issue (*Seitz et al, 2006*). As the geometrical relationship and redundant information are considered during the matching process, the robustness of the matching result to occlusion and noise is obviously higher than that of binocular-stereo matching algorithms. Multi-view image matching methods can be categorized as: voxel-based matching algorithms, polygonal-mesh based matching algorithms, depth map based

matching algorithms and patch-based matching algorithms. For the voxel-based matching algorithms, as the grid size of the voxel should be considered in the matching process, the matching results are poor for large-scale scene images, which makes these algorithms inapplicable in photogrammetric applications (*Seitz et al, 1997; Sinha et al, 2007*). The polygonal-mesh based algorithms depend greatly on the prior input, which leads to inflexibility (*Yoon et al, 2006*). Although depth map based approaches are more flexible than others, when the obtained depth map are noisy and redundant, a series of post-processing measurements, such as fusion, de-noising and filtering of the depth map are needed so that the redundant computation of the whole algorithm will be greatly improved (*Hirschmüller, 2008; Bradley et al, 2008; Geiger A et al, 2010*). By finding sparse feature points in the image, patch-based matching method constructs several small feature patch sets and reaches dense image matching effects through matching propagation (*Habbecke et al, 2006; Shan Q et al, 2014; Schönberger et al, 2016*). The PMVS method proposed by Furukawa and Ponce (*2010*) is widely used among state-of-the-art algorithms. As PMVS does not require any prior knowledge or initial value and it is applicable to 3D reconstruction of large-scale images, it has been extensively applied to UAV-based low-altitude photogrammetric 3D measurements (*Furukawa et al, 2010*). Ai et al. (*2015*) fed the high-precision sparse matching points into PMVS software as seed points to obtain dense point clouds, which greatly improved the matching efficiency of PMVS (*Ai et al, 2015*). Shao et al. (*2016*) took the matching result of PMVS as initial values for constructing expanded patch sets, and the correspondences are adjusted using the least squares refinement and patch-based MPGC (*Shao et al, 2016*) methods; as a result, the obtained point clouds are much denser and more robust to occlusion (*Baltsavias et al, 1996*).

In the recent five years, a lot of researcher utilized deep learning method for image dense matching. Some of them represent the pixel-wised correspondence searching problem into a classification problem and train the deep neural network to find the optimized disparity map for the stereo image pairs (*Zbontar and Yan, 2015; Luo et al, 2016; Tulyakov et al, 2017*). It extremely improves the matching efficiency. But the

output result is just used as an initial input for real SGM method. Others trained deep neural network for initial parameter optimization for traditional dense matching method, such as cost function and cost aggregation directions (*Seki et al, 2017; Zhong et al, 2017*). It makes traditional methods have better performance with the train datasets. The deep learning based method shows a better matching efficiency and accuracy, but its results are quite depend on the train datasets. When the test dataset and training dataset have big difference, the results were poor. Which means they cannot used for practical applications.

2.3 Three dimensional point cloud annotation

2.3.1 Machine learning based annotation methods

In recent decades, machine learning methods are widely used in image segmentation and classification applications (*Kotsiantis, 2007*). And it has shown a great performance in many cases. In this section we will introduce some widely used machine learning methods which can be used for point cloud annotation.

2.3.1.1 K-means method

The theoretical basis of the K-means clustering algorithm is the error square sum criterion, assuming that m_i represents the sample mean, and its expression is:

$$m_i = \frac{1}{N_i}\sum_{y\in T_i} y \tag{2.21}$$

Where N_i represents the number of samples in the i-th cluster T_i and y represents the sample. Then the error square sum criterion can be expressed by J_e as follows:

$$J_e = \sum_{i=1}^{c}\sum_{y\in T_i}\|y - m_i\|^2 \tag{2.22}$$

The goal of the K-means clustering algorithm is to find the clustering combination that minimizes J_e. The main steps of the algorithm are summarized as follows:

(1) Randomly select k samples or select the first k samples of the sample set as

the initial cluster centers, respectively $z_1(m), z_2(m), \cdots, z_k(m)$, where m is the number of iterations, the initial value is 1.

(2) Calculating the distance between the sample to be classified and the cluster center one by one, and assigning the sample to be classified to the cluster center of the minimum distance to obtain K clusters;

(3) Calculate the cluster center of each cluster obtained in the previous step:

$$z_j(m+1) = \frac{1}{N_j}\sum_{y \in s_j(m)} y \tag{2.23}$$

Where N_j represents the number of samples in the j-th cluster domain s_j, $1 \leq j \leq m$. Repeat step (2) to reassign the sample to the new cluster center;

(4) Calculate the difference between the cluster centers obtained in steps (2) and (3). If the difference is less than the given threshold, the iteration is stopped. Otherwise, return to step (2).

K-means algorithms can get the results with the smallest squared error. When processing large datasets, it is relatively scalable and efficient, and the computational complexity is O(NKt). Where N represents the observation number of the dataset, and t represents the iteration numbers.

2.3.1.2 Maximum likelihood estimation method

The maximum likelihood method is also called Bayesian classification. It is a statistically based supervised classification method and a classical method for remote sensing image classification. It is based on the Bayes criterion and assumes sample data of various types of features. The attribution probability follows the Gaussian distribution and is then classified according to the Gaussian distribution law to obtain the final result.

When using the maximum likelihood method for classification, firstly, a suitable discriminant function is established according to the selected sample features, and then the data to be classified is input into the discriminant function to calculate the probability of the belonging category. Suppose the feature category is K, the pixel to be classified is X; $P(K)$ represents the prior probability of the category K, usually

given according to prior knowledge or assumed that the classes are equal; $P(X|K)$ represents the likelihood probability. The probability that pixel X appears in class K; $P(K|X)$ represents posterior probability, that is, the probability that pixel X belongs to feature K; $P(K)$ represents the probability of occurrence of class X, then $P(K|X)$ can be expressed as:

$$P(K|X) = \frac{P(X|K)P(K)}{P(X)} \tag{2.24}$$

In the formula, $P(X)$ is a constant and therefore can be ignored, so the expression of the discriminant function can be expressed as follows:

$$\mathrm{g}(x) = \ln\{P(x|k)P(k)\} = \ln P(x|k) + \ln P(k) \tag{2.25}$$

The advantage of this method is the model is simple, and the overall distribution of the dataset is unnecessary.

2.3.1.3 Support vector machine method

The Support Vector Machine (SVM) classification algorithm is currently the most popular supervised classification method. The main goal of the SVM algorithm is to find the optimal hyper plane that divides the feature space, so that the classification boundary is maximized. The basis of the algorithm is nonlinear mapping, in which the nonlinear mapping kernel function in high-dimensional space is replaced. Support vector machines can be generally divided into linear separable support vector machines, linear non-separable support vector machines and multi-class support vector machines:

(1) Linear separable support vector machine

The support vector machine classification algorithm is based on the principle of VC dimension theory and structural risk minimization, and based on this, finds an optimal hyper plane, so that the distance between the sample points to be classified is as far as possible from the hyper plane, so that this The classification plane optimally distinguishes the two types of training samples

(2) Linear inseparable support vector machine

In practical applications, linearly separable support vector machines are difficult to satisfy most classification requirements. Therefore, the nonlinear mapping φ(x) is used to transform x into high-dimensional sample space to establish the optimal classification hyper plane.

(3) Multi-classification support vector machine

Support vector machine multi-classification methods usually include one-to-one, one-to-many and two-way acyclic graphs. Among them, in the two-way acyclic graph classification, for the classification problems of n categories, it is first necessary to obtain training through training. n(n-1)/2 classifier models, then train the left or right child nodes based on the binary tree structure, and finally the leaf nodes determine the final class.

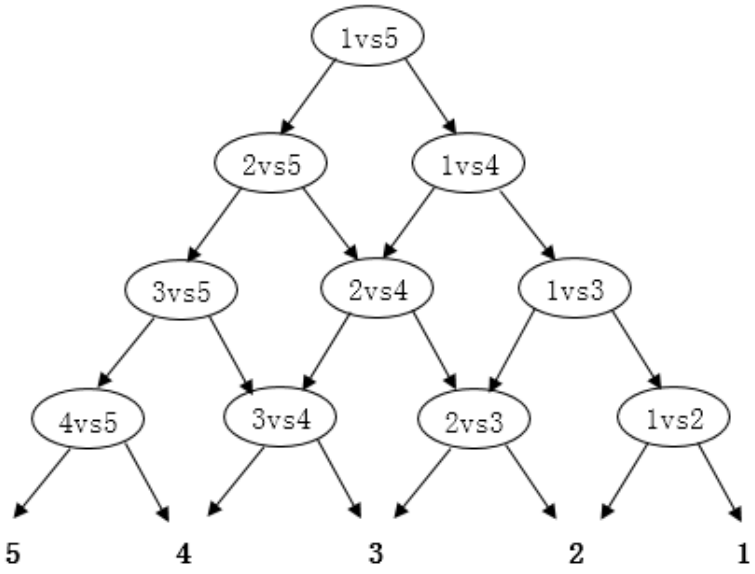


Fig 2.7 Binary tree structure training diagram (*Breiman, 2017*)

2.3.2 Deep learning based annotation methods

Deep learning has gradually played an important role in many research fields, including computer vision, natural language processing, handwritten character recognition, target detection, target tracking, information retrieval, pose estimation, image annotation, image classification, image generation, and features extraction and

coding, etc (*Lecun et al, 2015*). For example, in the field of image classification, the classification accuracy of convolutional neural networks introduced in the previous public datasets such as ImageNet and Cifar10 is increasing (*Krizhevsky et al, 2012*); in the field of target detection, R-CNN and Fast R-CNN (*Ren et al, 2015*) and many deep learning algorithms such as YOLO (*Redmon et al, 2016*) and SSD (*Liu et al, 2016*), the detection speed is become faster and faster. Until now the accuracy and speed of the SSD is 23 frames / sec and the accuracy achieved 75.1%. In the field of image segmentation, deep learning algorithm has been rapidly improved, the average accuracy of the original FCN (*Long et al, 2015*) model was 62.6%, and the average accuracy of DeepLab (*Chen et al, 2014*) increased to 72.7%. The accuracy of CRF-RNN (*Zheng et al, 2015*) is now 74.7%.

However, deep learning for point cloud annotation has been researched since 2015. Science the dataset is hardly acquired (*Hackel et al, 2017*).Considering point cloud only contained with 3D coordinates, most deep learning based methods represent the point cloud to 2D images for training and validation (*Hu et al, 2016; Gevaert et al, 2017*).There is still much room for development in this field. At this moment we describe the most widely used deep networks in image classification field.

2.3.2.1 AlexNet

Alex's AlexNet network structure won the 2012 ImageNet Image Recognition Competition and is a milestone in the field of computer vision. It proves the validity of convolutional neural networks in image classification and becomes the core algorithm for image classification. Breakthrough progress has been made in the field of image classification, which in turn promotes the development of subsequent deep learning. AlexNet consists of 5 convolutional layers, 3 maximum pooling layers and 3 fully connected layers. It takes the lead in using the ReLU function as an activation function, which solves the problem of gradient disappearance during training, and reduces the use of Dropout operations and data enhancement techniques. The network

was over-fitting and was first trained using 2 GPUs. Figure 2.8 shows the network structure of AlexNet:

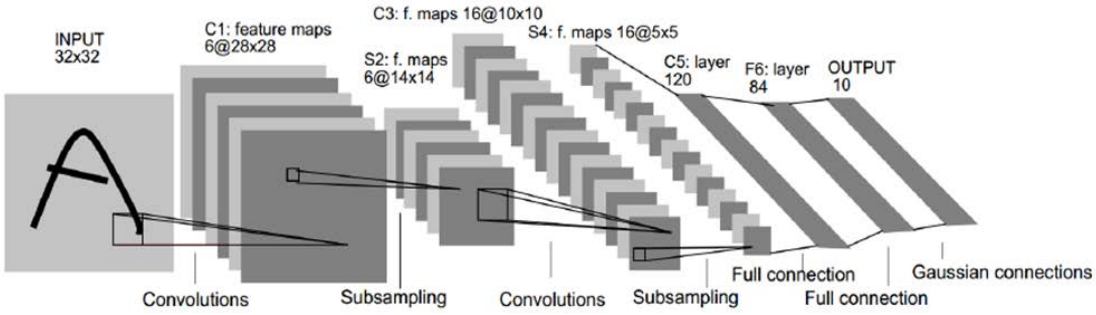


Fig 2.8 The structure of AlexNet (*Krizhevsky et al, 2012*)

Although AlexNet is really good at dealing with object recognition, it has shown a poor performance than other networks when deals with multiple label classification problem.

2.3.2.2 GoogleNet

GoogLeNet won the 2014 ImageNet Image Recognition Competition with a top-5 error rate of 6.67%. Its main feature is to increase the depth and width of the neural network and improve the classification effect on the basis of ensuring the constant computing resources. The most important improvement is the use of the Inception structure. The Inception structure uses dense components to approximate the optimal local sparse structure, expands the depth of the convolutional neural network to 22 layers and increases the width of the network while reducing the number of parameters. Figure 2.9 is a schematic diagram of the network structure of GoogLeNet.

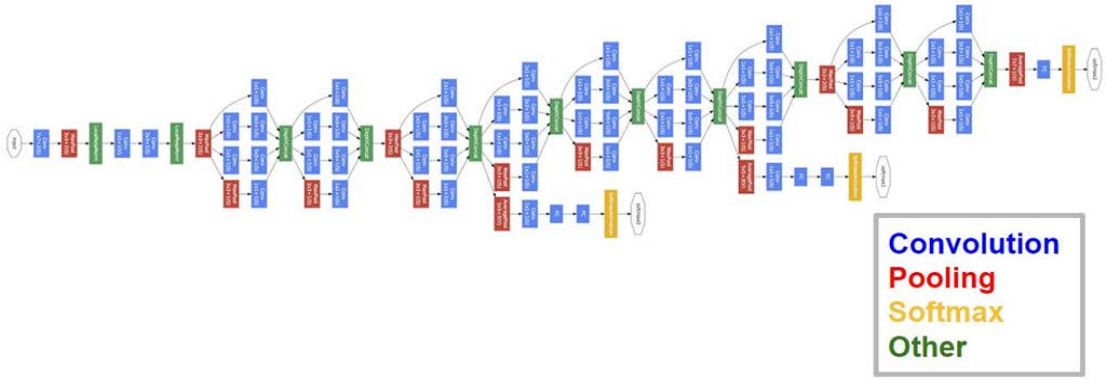


Fig 2.9 The structure of GoogLeNet (*Szegedy et al, 2015*)

GoogleNet shows a great performance on multiple label classification tasks, but the max pooling layer may cause the spatial information loss.

2.3.2.3 ResNet

ResNet, proposed by He Kaiming et al., is the champion of the 2015 ImageNet Image Recognition Competition. The 152-layer ResNet model solves the problem of the accuracy of the training set caused by the increase of the number of network layers by using the deep residual structure, and breaks the bottleneck that the number of network layers of the convolutional neural network cannot continue to be effectively deepened after reaching a certain number. Its network structure is shown in Figure 2.10.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig 2.10 The structure of ResNet (*He et al, 2015*)

The residual network has a simple structure and solves the problem of degraded performance of deep convolutional neural networks under extremely deep conditions. It has great performance on small datasets and easy implementation on other networks.

Chapter 3
Seed point extraction in poor texture remote sensing images

In this chapter, I will introduce the basic concept and algorithms used in our poor texture tie point matching approach. In the former chapter, I have introduced some widely used matching methods. They all have some advantages and limitations. In order to extract accurate and uniformed seed points in poor texture images, we proposed a graph theory based image matching approach. And I will detailed describe our approach by 4 sections.

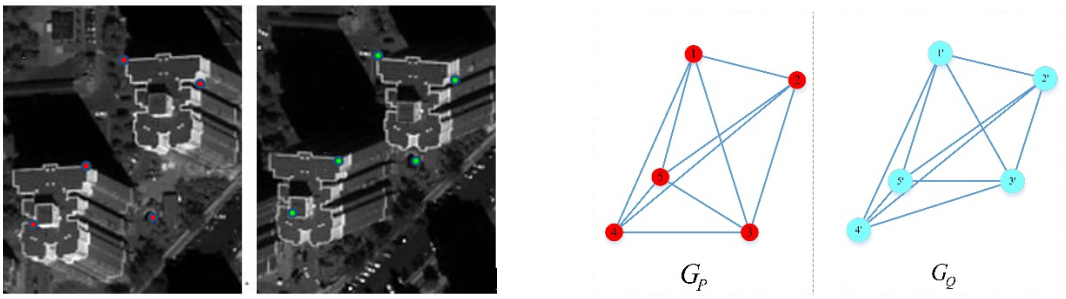
3.1 Methodology

This section proposes an EW-HOGM method to address the poor textural image matching problem. Section 3.1.1 presents the concepts of HOGM. Section 3.1.2 introduces the proposed EW-HOGM. Section 3.1.3 describes the experimental results of the synthetic data. Section 3.1.4 provides a theatrical analysis of the computational complexity.

3.1.1 HOGM

Image tie point matching finds correspondences between two sets of features. This process can be defined as a graph matching problem. As shown in Figure 3.1, five pairs of image feature points are extracted from the source and target images. Image feature points can be regarded as the nodes of graphs, and feature characters, such as locations and gray levels, are the attributes of the nodes. The relationship between two feature points, such as angles and distances, can be considered the edges of graphs. The matching problem of two image feature sets is a node matching problem of the two graphs. Hence, the tie point matching problem is transformed into a graph

matching problem.



(a) (b)

Figure 3.1 Two complete graphs constructed with image features. (a) Image features and (b) two graphs with their nodes representing image features.

Given two image feature sets *P* and *Q*, and their corresponding graphs G_P and G_Q. If nodes $V_i \in G_P$ and $V_{i'} \in G_Q$ are assignments, then $f_i \in P$ and $f_{i'} \in Q$ are tie points, and vice versa. The graph matching problem can be formulated as

$$C \underline{\underline{\Delta}} \{c_{ii'}\}_1^n = \{(V_i, V_{i'})\}_1^n, \quad n \le \min(n_P, n_Q) \tag{3.1}$$

where *C* is an assignment set; n_P and n_Q are the feature numbers of two feature sets; *i* and *i'* are the labels of nodes V_i and $V_{i'}$, which represent the node indices in this paper; and $c_{ii'}$ is an assignment element of *C*, and it indicates V_i corresponds to $V_{i'}$. The relationship (i.e., correspondences) of the graph nodes can also be depicted by assignment matrix $\mathbf{Z}^* \in \{0,1\}^{n_P \times n_Q}$, where $z_{ii'}^* = 1$ implies that V_i corresponds to $V_{i'}$ and $z_{ii'}^* = 0$ implies that V_i is not matched to any node in G_Q. As shown in Figure 3.2, the graph matching problem can be divided into three types based on their constraint forms: first-order, second-order, and high-order (third or higher) graph matching problems.

The first-order graph matching problem is established on a unitary affinity matrix $\mathbf{A} \in IR^{n_P \times n_Q}$ with

$$a_{ii'} = \Omega_1(c_{ii'}) = \exp\left(-\frac{1}{\varepsilon^2}\left(\left\|\mathbf{f}_i - \mathbf{f}_{i'}\right\|_2\right)^2\right) \tag{3.2}$$

where $a_{ii'}$ is the point-wise similarity of nodes V_i and $V_{i'}$; $\Omega_1(.)$ denotes the point-wise similarity measure (e.g., Euclidean distances of SIFT descriptors); $\|\cdot\|_2$ represents the length of a vector; $\mathbf{f}_i$ and $\mathbf{f}_{i'}$ are the attributes of nodes V_i and $V_{i'}$, respectively, i.e., they are the radiometric descriptors of f_i and $f_{i'}$.

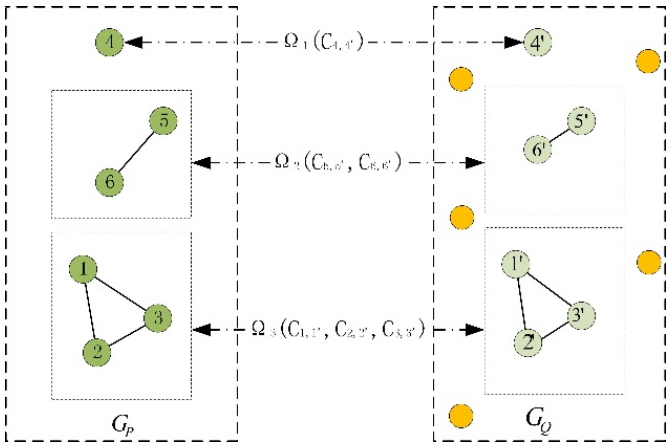


Figure 3.2 Multi-order graph matching problem (the yellow dots denote outliers)

The first-order graph matching problem finds the optimal solution for the following objective function:

$$\begin{aligned} &\mathbf{z}^* = \arg\max_{\mathbf{z}}(\mathbf{z}^{\mathrm{T}}\mathbf{a}) \in \{0,1\}^{n_P \times n_Q} \\ &s.t. \quad \mathbf{Z}\,1 \le 1, \mathbf{Z}^{\mathrm{T}}1 \le 1 \end{aligned} \tag{3.3}$$

where $\mathbf{a} \in IR^{n_P n_Q}$ is the row-wise vectorization of matrix **A**, 1 denotes a vector that all elements are one; **Z** is a soft assignment matrix located in the continuous vector space, and $\mathbf{Z}^* \in \{0,1\}^{n_P \times n_Q}$ is a hard assignment matrix. Thus, an additional process is required to discretize **Z** into a binary matrix, and the commonly selected method for discretization is the greedy algorithm (shown in Algorithm 1, *Leordeanu and Hebert, 2005*).

In Equation (3.3), $\mathbf{Z}\,1 \le 1, \mathbf{Z}^{\mathrm{T}}1 \le 1$ imposes restrictions on the matching correspondences. Every node of G_P has a maximum of one corresponding node in G_Q, and every node of G_Q has a maximum of one corresponding node in G_P. Equation (3.3)

can be solved using the Hungary algorithm (*Edmonds, 1965*) or approximated via dynamic programming (*Belongie et al., 2002*).

Algorithm 1: Greedy algorithm for discretization

Input: Soft assignment matrix $\mathbf{Z}^{m\times n}$

Output: Hard assignment matrix $\mathbf{Z}^{*m\times n}$

1 begin

2 for $i = 0...m$ do

3 find max($z_{i,i'}$) in row i of $\mathbf{Z}^{m\times n}$

4 if $\sum_{i'}^{n} z_{i,i'}^{*} = 0$, set $z_{i,i'}^{*} = 1$ and $z_{i,j'}^{*} = 0 (i' \neq j')$

5 else continue

6 end

7 end

The unitary similarity used in the first-order graph matching is only invariant to radiometric variation (e.g., image feature descriptors such as SIFT and SURF).

As shown in Figure 3.2, two edges $E_{ij} \in G_P$ and $E_{i'j'} \in G_Q$, as well as two node pairs $V_i, V_j \in E_{ij}$ and $V_{i'}, V_{j'} \in E_{i'j'}$, are given. Similar to the point-wise similarity measure, the pair-wise similarity measure can be denoted by $\Omega_2 = \Omega_2(c_{ii'}, c_{jj'})$. The affinity matrix defined under pair-wise constraints is given by

$$a_{ii'jj'} = \Omega_2(c_{ii'}, c_{jj'}) = \exp(-\frac{1}{\varepsilon^2}(\left\| \mathbf{f}_{ij} - \mathbf{f}_{i'j'} \right\|_2)^2) \tag{3.4}$$

where $\mathbf{f}_{ij}$ and $\mathbf{f}_{i'j'}$ are the descriptors of edges E_{ij} and $E_{i'j'}$, respectively.

The objective function of second-order graph matching problems is defined as follows:

$$\begin{aligned} &\mathbf{Z}^{*} = \arg\max_{\mathbf{z}} (\mathbf{Z}^{\mathrm{T}} \mathbf{A} \mathbf{Z}) \in \{0,1\}^{n_P \times n_Q} \\ &s.t. \quad \mathbf{Z}\, 1 \le 1, \mathbf{Z}^{\mathrm{T}} 1 \le 1 \end{aligned} \tag{3.5}$$

where $\mathbf{A} \in IR^{n_P n_Q \times n_P n_Q}$ is the affinity matrix of the two graphs. The elements of A are

constructed using Equation (3.4).

Equation (3.4) shows that the affinity matrix $\mathbf{A} \in IR^{n_P n_Q \times n_P n_Q}$ is non-negative and symmetric. The most effective means to solve Equation (3.5) is the relaxed spectral method (*Leordeanu and Hebert, 2005*):

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left(\frac{\mathbf{w}^{\mathrm{T}} \mathbf{A} \mathbf{w}}{\mathbf{w}^{\mathrm{T}} \mathbf{w}} \right), \tag{3.6}$$

where **w*** is the leading eigenvector of matrix **A**, and the final solution can be achieved using Algorithm 1.

Notably, the point-wise constraint is compatible with the pair-wise constraint. In Equation (3.4), the edge similarity $\Omega_2(c_{ii'}, c_{jj'})$ is equal to $\Omega_1(c_{ii'})$, given that *i=j* and *i'=j'*. Thus, point-wise similarity can be integrated into edge similarity in second-order graph matching. Apart from being invariant to radiometric variation, edge similarity is also invariant to rotation.

However, neither first-order nor second-order graph matching is invariant to scaling and small affine transformation. Thus, second-order graph matching should be extended to higher-order (third order or higher). For simplicity, third-order graph matching is presented as an example. Similar to point-wise and pair-wise similarities, triplet (triangle) similarity can be defined as

$$a_{ii'jj'kk'} = \Omega_3(c_{ii'}, c_{jj'}, c_{kk'}) = \exp\left(-\frac{1}{\varepsilon^2}\left(\left\|\mathbf{f}_{ijk} - \mathbf{f}_{i'j'k'}\right\|_2\right)^2\right) \tag{3.7}$$

where $a_{ii'jj'kk'}$ is the triplet similarity of triangles T_{ijk} and $T_{i'j'k'}$; and $\mathbf{f}_{ijk}$, $\mathbf{f}_{i'j'k'}$ are the descriptors of the two triangles. As shown in Figure 3.3, the two descriptors can be represented as $\mathbf{f}_{ijk} = (\cos\theta_i, \cos\theta_j, \cos\theta_k)$ and $\mathbf{f}_{i'j'k'} = (\cos\theta_{i'}, \cos\theta_{j'}, \cos\theta_{k'})$. Because the matched triangles are approximately similar, thus, triangle descriptors are invariant to scaling and rotation. The third-order graph matching is sufficient for remote sensing image matching, because the elevation variations of the Earth's surface is small compared with the altitudes of satellites. Thus, the ground can be regarded as flat and two conjugated regions can be approximated with similarity transform. The similarity transform accords with the triangle constraints.

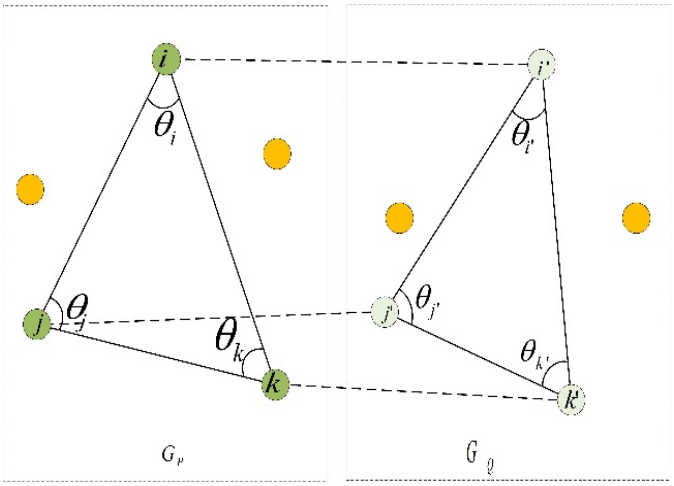


Figure 3.3 Diagram of triangle descriptors

The objective function of third-order graph matching is defined on an affinity tensor built using Equation (3.7):

$$\begin{aligned} &\mathbf{z}^{*} = \arg\max_{\mathbf{z}}(\mathbf{A}\otimes_3 \mathbf{z}\otimes_2 \mathbf{z}\otimes_1 \mathbf{z}) \in \{0,1\}^{n_P\times n_Q} \\ &s.t. \quad \mathbf{Z}\,1 \le 1, \mathbf{Z}^{\mathrm{T}}1 \le 1 \end{aligned} \tag{3.8}$$

where $\mathbf{A} \in IR^{n_P n_Q \times n_P n_Q \times n_P n_Q}$ is a third-order tensor; and $\otimes_t$ denotes the tensor product symbol. For the affinity tensor A, $a_{ii'jj'kk'}$ is a tensor element with a coordinate of (i×n+i', j×n+j',k×n+k') in the third-order tensor cube (shown in Figure 3.4). Equation (3.8) can be solved via tensor power iteration (*Duchenne et al., 2011; Lyzinski et al., 2016*).

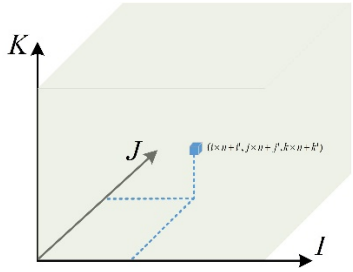


Figure 3.4 Illustration of $a_{ii'jj'kk'}$ in the third-order tensor cube

Higher-order graph matching can be directly extended using objective function (3.8). However, such matching can lead to an exponential growth of computational

complexity. Point-wise and pair-wise similarities can also be integrated into triangle similarities. In Equation (3.7), if $i = j = k, i' = j' = k'$, then triangle similarities degrade into point-wise similarities. Similarly, if $i = j \neq k, i' = j' \neq k'$; $i \neq j = k, i' \neq j' = k'$; or $j \neq i = k, j' \neq i' = k'$; then Equation (3.7) degrades into Equation (3.4). Thus, the third-order affinity tensor is the general form of the affinity matrix, and geometric and radiometric information can be easily integrated into the affinity tensor.

Given that $i = j = k, i' = j' = k'$, Equation (3.7) can be rewritten into the following form:

$$a_{ii'jj'kk'} = \exp(-\frac{1}{\varepsilon^2}(\left\|\mathbf{f}_i - \mathbf{f}_{i'}\right\|_2)^2). \tag{3.9}$$

Equation (3.9) integrates radiometric information into the third-order affinity tensor; thus, Equation (3.8) is globally optimal in geometry and radiometry.

However, point-wise similarities are related to the radiometric feature descriptor (e.g., 128 dimensions (D) SIFT descriptor or 36D SURF descriptor), whereas triplet similarities are related to geometric information (i.e., 3D triangle descriptor). Thus, these two similarity measures have different physical dimensions and should be normalized into a uniform measurement framework:

$$\tilde{\mathbf{f}} = \frac{\mathbf{f}}{\left\|\mathbf{f}\right\|_2} \tag{3.10}$$

where $\tilde{\mathbf{f}}$ denotes the normalized feature descriptor.

In addition, normalized descriptors are more distinctive than triangle descriptors because they have more dimensions. Thus, a balanced factor is required among the descriptors as follows:

$$w_f = \frac{n_2 \sum_{i=1}^{n_1} dof_i^1}{n_1 \sum_{i=1}^{n_2} dof_i^3} \tag{3.11}$$

where dof_i^1 is the point-wise similarity; dof_i^3 is the triplet similarity; and w_f is the balanced weighted factor; dof_i^1 and dof_i^3 can be estimated using a triangular

irregular network (TIN), which is constructed using two matched image feature point sets; n_1 and n_2 are the numbers of matched points and triangles, respectively, in TIN. Through normalization and balancing, geometric and radiometric information works in the same measurement framework and plays the same role in matching.

3.1.2 EW high-order affinity tensor

Real matching tasks arise from images captured by different sensors in various views or at different moments, thereby leading to the appearance of new features and the disappearance of old ones. These appearing and disappearing features constitute outliers. For example, in an image pair that consists of considerably high buildings, if one feature is detected in the source image, then its corresponding feature in the target image may be occluded by high buildings. Hence, the detected feature in the source image is an outlier. For a formal description, two graphs G_P and G_Q built using two feature sets are given. If $V_P \in G_P$, $V_Q \in G_Q$ and $\sum_i z^*_{i,p} \sum_j z^*_{q,j}$ (that is, *Vp* and *Vq* have no corresponding node), then *Vp* and *Vq* are outliers; otherwise, they are inliers. A major challenge in real-world graph matching problems is how to tolerate numerous outliers arising in typical visual tasks, such as image matching and object recognition. Generally, outliers are more than inliers, which results in difficulties in distinguishing inliers from outliers because of clustering. Outliers can also lead graph matching toward the local optima, and thus, produce erroneous image matching results. An EW-HOGM algorithm is proposed to address real-world image matching tasks.

As shown in Equations. (3.2), (3.4) and (3.7), if outliers are present in either feature point set, then the third-order affinity tensor **A** is equal to the correct affinity tensor $\tilde{\mathbf{A}}$, which is produced by point sets without outliers and with a turbulent tensor $\Delta\mathbf{A}$ created by the point sets without inlier and noise:

$$\mathbf{A} = \tilde{\mathbf{A}} + \Delta\mathbf{A} \tag{3.12}$$

Noise is neglected because its effect on graph matching is less than that of outliers.

The main idea of reining outliers is increasing $\tilde{\mathbf{A}}$ while decreasing $\Delta\mathbf{A}$.

The solution for Equation (3.8) is the following power iteration:

$$\forall I, \quad Z_I^{(n+1)} \leftarrow \sum_{J,K} \Omega_3(I,J,K) Z_J^{(n)} Z_K^{(n)} \tag{3.13}$$

where I, J, K are the shorthand of index pairs $(i,i'),(j,j'),(k,k')$, and n is the nth iteration.

Equation (3.13) is one step of tensor power iteration and is illustrated in Figure 3.5.

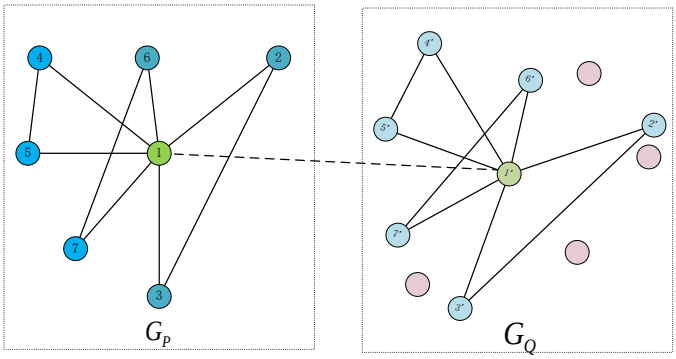


Figure 3.5 Depiction of node correspondences (the pink dots are outliers)

As shown in Figure 3.5, the relationship of V_1 and $V_{1'}$ is determined by triangles that contain vertices V_1 and $V_{1'}$. For example, the value of the soft assignment matrix element $Z_{1,1'}^{(n+1)}$ is determined by the sum of the weighted similarities of triangles $T_{1,2,3}$ and $T_{1',2',3'}$, $T_{1,4,5}$ and $T_{1',4',5'}$, $T_{1,3,7}$ and $T_{1',3'7'}$, and so on. That is, node similarity is determined by the edges of the triangles. Equation (3.13) can also be modified into another form as follows:

$$\forall I, J, \quad Z_I^{(n+1)} Z_J^{(n+1)} \leftarrow \sum_K \Omega_3^{(n)}(I,J,K) Z_K^{(n)}, \tag{3.14}$$

where $Z_I^{(n+1)} Z_J^{(n+1)}$ implicitly contains the edge assignment information. Equation (3.14) is illustrated in Figure 3.6.

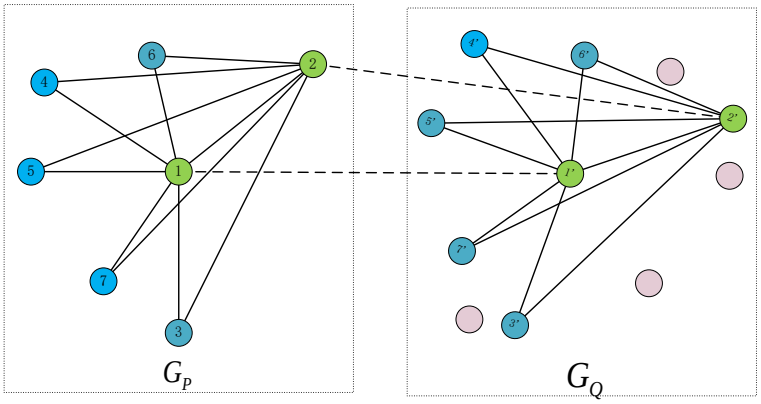


Figure 3.6 Depiction of edge correspondences (the pink dots are outliers).

Similar to that of node correspondences, the relationship of $E_{1,2}$ and $E_{1',2'}$ is determined by the triangles that contain edges $E_{1,2}$ and $E_{1',2'}$. For example, $Z_I^{(n+1)}Z_J^{(n+1)}$ is determined by the sum of the weighted similarities of $T_{1,2,3}$ and $T_{1',2',3'}$, $T_{1,2,4}$ and $T_{1',2',4'}$, $T_{1,2,5}$ and $T_{1',2',5'}$, and so on. Thus, the edge assignment probability can be defined as

$$s_{IJ}^{(n+1)} = \sum_K \Omega_3^{(n)}(c_I, c_J, c_K) Z_K^{(n)} \tag{3.15}$$

As shown in Equation (3.15), for the triangle pair T_{ijk} and $T_{i'j'k'}$, if edges $E_{i,j}$ and $E_{i',j'}$ are markedly different, then the probability that $T_{i,j,k}$ and $T_{i',j',k'}$ are matching triangles is significantly low. By contrast, if $T_{i,j,k}$ and $T_{i',j',k'}$ are matching triangles, then edges $E_{i,j}$ and $E_{i',j'}$ have a high matching probability. Thus, the edge assignment probability can be regarded as a weighted factor in constructing third-order tensors:

$$w_e(I,J) = \exp(-(s_{IJ} - 0.5N)^2), \tag{3.16}$$

where N is equal to $(n_P - 2) \times (n_Q - 2)$.

Substitute Equations (3.9), (3.11), and (3.16) into Equation (3.7), the EW tensor element can be written as

$$a_{ii'jj'kk'} = \begin{cases} \exp(-(w_f \left\| \mathbf{f}_{ijk} - \mathbf{f}_{i'j'k'} \right\|_2)^2 / \varepsilon^2), & if\ i = j = k\ and\ i' = j' = k' \\ w_e \times \exp(-(\left\| \mathbf{f}_{ijk} - \mathbf{f}_{i'j'k'} \right\|_2)^2 / \varepsilon^2), & if\ i \neq j \neq k\ and\ i' \neq j' \neq k' \\ 0 & others \end{cases} \quad (3.17)$$

For four nodes $V_i, V_j \in E_{ij}$ and $V_{i'}, V_{j'} \in E_{i'j'}$, if any of these nodes is an outlier, then the weighted factor computed using Equation (3.16) will be small, thereby leading to a smaller tensor element $a_{ii'jj'kk'}$. By contrast, if the four nodes are all inliers, then the tensor element $a_{ii'jj'kk'}$ will be augmented by a high weighted factor. In this manner, $\tilde{\mathbf{A}}$ is indirectly increased although $\Delta\mathbf{A}$ is nearly unchanged, and thus, robust to outliers.

3.2 Data descriptions and implementation details

Experimental data descriptions and implementation details are provided in this section. Section 3.2.1 presents the experiment design and data details. Section 3.2.2 provides the assessment criteria for comparing four tie point matching algorithms, namely, SIFT, SURF, FAST, and EW-HOGM. Section 3.2.3 describes the experiment results and discussions.

3.2.1. Experiment design

Four pairs of remote sensing images captured by different satellites are used to verify the effectiveness of the proposed algorithm. The details of the experimental data are presented in Table 3.1.

The first pair of images is captured by the camera of the Pléiades satellite with different angles in Beijing. The image content is composed of high buildings and bare lands. Large geometric and radiometric distortions occur between overlapping regions because of different camera angles and high buildings. The second pair of images is

from WorldView. These images mainly contain forests, which cause repetitive textures. In addition, considerable textural changes are found because acquisition time interval spans approximately 3 years. The third image pair comprises two SPOT 5 images of Baoji, which is located on the Loess Plateau in China. Massive homogeneous textures caused by deserts are present, as well as image distortions caused by topographic relief. The fourth pair of images covers low buildings and bare lands taken by SPOT 5. Substantial geometric distortions are observed because the images are captured by forward and backward cameras. All four pairs of images exhibit a considerable amount of poor texture, including occlusion, discontinuity, homogeneity, and repetitiveness. Geometric distortions are also found because of different camera angles and remarkable topographic relief.

Table 3.1 Details of experimental stereo image pairs

Pair No.	Satellite	Main contents	Image size (pixels)	GSD (m)	Acquisition time	Location	Check point number
1	Pléiades	High buildings and bare lands	9152 × 9720 9585 × 9793	0.5	2015 2015	China–Beijing	40
2	WorldView	Forests	10197 × 12011 10205 × 11545	0.5	2012 2015	China–Wuhan	45
3	SPOT 5	Deserts	8340 × 7960 7803 × 7803	2.5	2014 2014	China–Baoji	34
4	QuickBird	Low buildings and bare lands	9775 × 8903 10077 × 8935	0.61	2003 2003	USA–Spokane	32

The construction of a tensor on the complete images is nearly impossible because the experimental image sizes are large; therefore, a coarse-to-fine strategy is adopted. First, the source and target images are downsampled to 1/16 of their original size. Then, SIFT is applied to extract several tie points and estimate the homography matrix between image pairs. The source image is then divided into regular grid cells. Each grid cell is 400 × 400 pixels and marked by IA. The cell center of IA is projected

onto the target image via homography transformation. The corresponding region (marked by IB) of IA with 500 × 500 pixels is located at the center of the projected points. Lastly, the tie points of IA and IB are obtained using EW-HOGM, and the gross error detection procedure is applied using RANdom SAmple Consensus (RANSAC, *Fischler and Bolles, 1981*) with a projective model. The detailed procedure of EW-HOGM is shown in Figure 3.8.

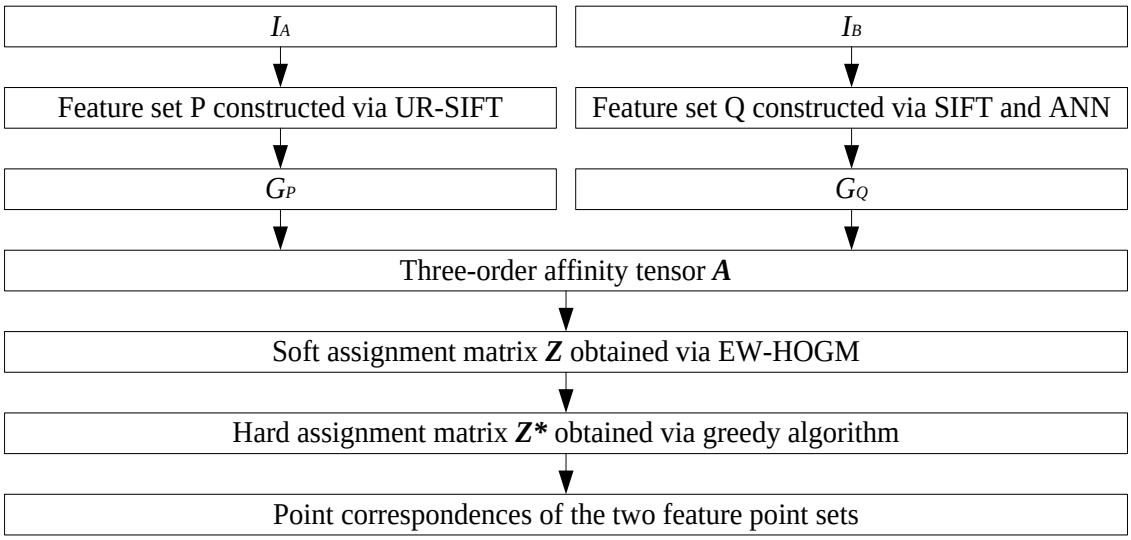


Figure 3.8 Workflow of tie point matching based on EW-HOGM

As shown in Figure 3.8, uniform robust SIFT (UR_SIFT, *Sedaghat et al., 2011*) is applied to obtain a uniformly distributed feature point set (donated by $\mathbf{S}_A$ with element size n_A). Moreover, to ensure the repetitiveness of features, SIFT with a low contrast threshold, which is set to 0.005 for normalized images, is used to extract the feature point set (donated by $\mathbf{S}'_B$) from I_B. For each feature point in $\mathbf{S}_A$, the approximate nearest neighbor (ANN) algorithm (*David and Sunil, 2010*) is applied to search k (k is 5 in these experiments) potential matches in $\mathbf{S}'_B$. Then, feature set $\mathbf{S}_B$ with an element size n_B is constructed using all potential matches that correspond to $\mathbf{S}_A$. Finally, G_A and G_B are built using $\mathbf{S}_A$ and $\mathbf{S}_B$, respectively. The initial tensor A of G_A and G_B can be established using Equation (3.17) with w_e = 1.0 and

$w_f = 0.01$. To make the third-order tensor against minor local deformation, ε is set to $\pi/15$.

As indicated in Section 3.2.4, the most time-consuming part of EW-HOGM is power iteration. Three strategies are adopted to make the tensor spare and thus, speed up iteration. First, a threshold is provided for triangle similarity. For two triangles T_{ijk} and $T_{i'j'k'}$, if their similarity is larger than $\pi/5$, then $a_{ii'jj'kk'}$ will be set to 0. Second, a sampling strategy is applied to build affinity **A** because forcing all triangles to be matched will be highly redundant. Thus, t (t is set to $(n_A-2)(n_A-1)n_A/27$ in these experiments) triangles in G_A are selected to compute tensor elements. Tensor elements related to the unselected triangles will be set to 0. Third, ANN-based searching is used to build the tensor. For triangle T_{ijk} in GA, the matched triangle in GB is most likely in its p (p is set to 20 in these experiments) nearest neighbors; thus, p nearest triangles are used to compute tensor elements related to T_{ijk}, and the other tensor elements related to T_{ijk} are set to 0. Hence, only t$\times$p nonzero elements remain, thereby making power iteration considerably faster. Besides, for a tradeoff between efficiency and accuracy, the number of power iteration is set to 5.

From the perspective of a probabilistic graph (*Egozi et al., 2013*), the elements of the soft assignment matrix **Z** express the assignment probabilities of graph nodes. For example, $z_{ii'}$ denotes the matching probability of nodes V_i and $V_{i'}$. Thus, the elements of **Z** can be ranked in descending order, and the first $\hat{n} < n_P$ elements can be selected as the final solution prior to discretization. This process can improve the robustness of the matching algorithms. Therefore a modified version of a greedy algorithm is used in the discretization process. One-to-one constraints are discarded (Step #4 in Algorithm 1) and the elements of Z are arranged in descending order. In addition, the first $n_A/2$ elements are selected to construct set $\{z^*_{ii'}\}$ and disregard others. Thus, the image feature pair set $\{(f_i, f_{i'})\}$ is the final matched feature set.

In the comparison experiments, three state-of-the-art radiometry-based tie point matching algorithms, namely, SIFT, SURF, and FAST, are used for matching. The matching process also applies the previously mentioned coarse-to-fine strategy. These three algorithms are matched through ANN with an NNDR threshold of 0.8.

3.2.2 Quality assessment criteria

Recall, dispersion, and positional accuracy are estimated in the comparison experiments to evaluate the feasibility of the proposed algorithm. Matching recall is estimated in a sub-image pair, whereas dispersion and positional accuracy are estimated in a complete image pair, as shown in Figure 3.9.

1. Recall

Four pairs of sub-images are artificially selected from the complete image pairs listed in Table 3.1. These sub-images are 400 × 400 pixels and have different types of poor textures (Figure 3.9(b)). A sub-image pair is regarded as I_A and I_B, which are matched with the workflow illustrated in Figure 3.8. The matching recall is defined as

$$recall = \frac{CM}{C} \tag{3.19}$$

where CM (correct matches) is the correct matched tie points, and C (correspondences) is the total number of corresponding features between I_A and I_B. C and CM are determined as follows. First, a skilled operator selects 8–10 evenly distributed tie points. Then, an accurate projective transformation model is computed using the selected tie points. Finally, the computed projective model with a threshold of 3.0 pixels is used to determine C and CM.

2. Dispersion

The dispersion of the tie points is calculated via TIN analysis because all the tie points can be regarded as a network, and triangles of TIN contain the dispersion information of tie points. The dispersion is estimated as follows. First, a TIN is constructed using Delaunay's algorithm (*Delaunay, 1934*). Subsequently, dispersion is estimated based

on Equation (18), which was first introduced by Zhu (*Zhu et al., 2006*):

$$D = D_S \times D_A = \sqrt{(\sum_i^n (\frac{A_i}{\overline{A}} - 1)^2)/(n-1)} \times \sqrt{(\sum_i^n (S_i - 1)^2)/(n-1)} \quad (3.18)$$

where D_A and D_S indicate the area and shape variations of triangles respectively; n denotes the number of triangles in TIN; A_i represents the area of triangle i; $\overline{A} = \sum_i A_i / n$ indicates the average area of all the triangles; $S_i = 3 \times \max(J_i)/\pi$, where $\max(J_i)$ represents the maximum angles in triangle i. If both D_A and D_S are small, then D should be also small. The smaller D is, the better the distribution of tie points is.

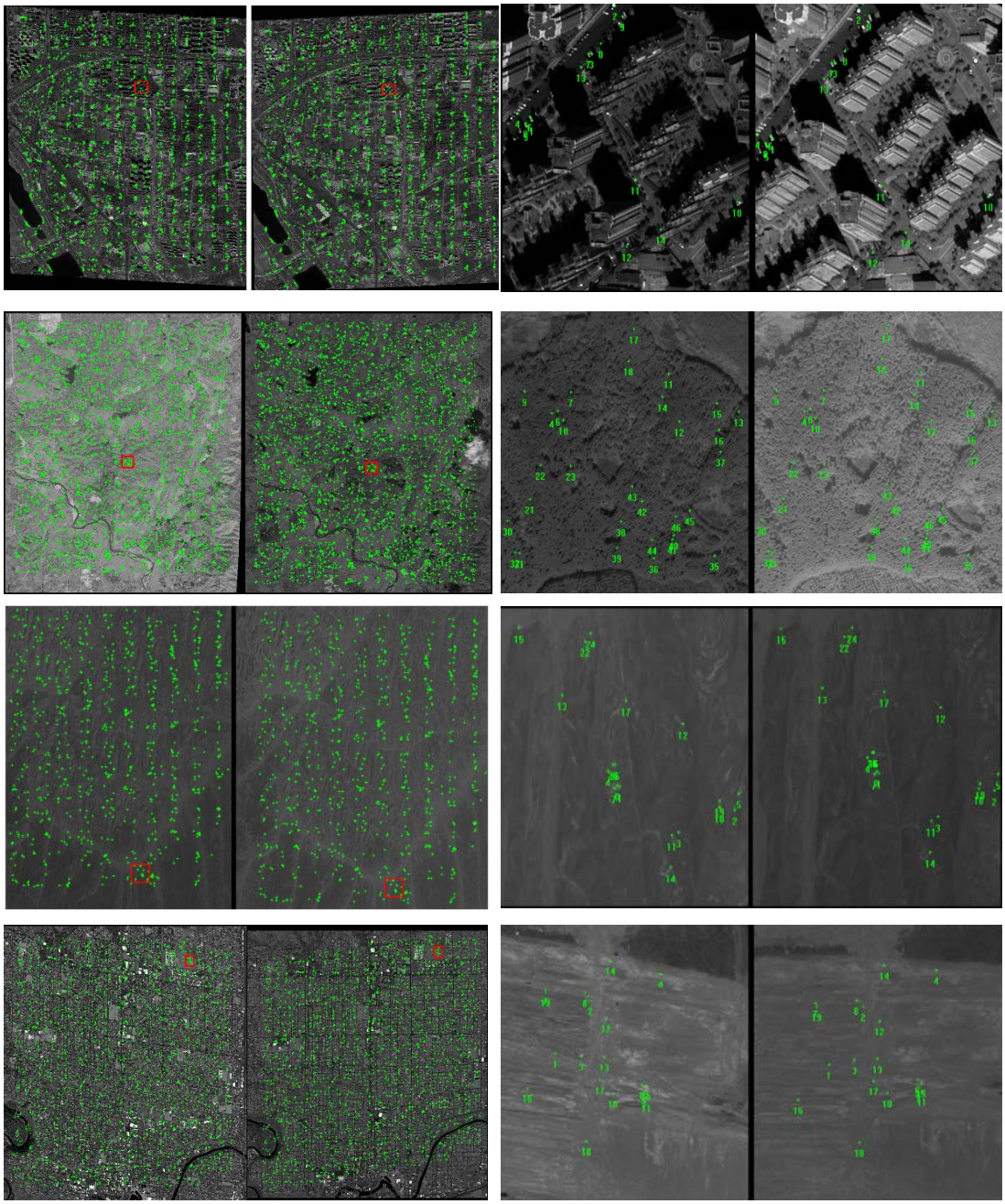
3. Positional accuracy

TIN integrated with piece-wise linear (PL) transform is reported to be a competitive method for image registrations (*Ye and Shan, 2014*); thus, TIN analysis and PL transform are applied in this study to estimate positional accuracy. TIN is constructed using the tie points of the complete image pair, and checkpoints are fed to the Delaunay triangulation. An affine transformation model is then fitted by the matched triangles. Lastly, positional accuracy is estimated via the root mean square error, which is computed through the affine transformation of checkpoints. The numbers of checkpoints in different image pairs are listed in Table 3.1.

However, different TIN construction algorithms may lead to various TIN structures, which generate varying dispersions and positional accuracy. By contrast, if tie points have a good distribution and abundant correct matches, then the variations of these values are negligible.

3.3 Experiment results and discussions

The experiment results of the tie point distributions are illustrated in Figure 3.9.



(a) (b)

Figure 3.9 Matching results based on EW-HOGM, and matching details of sub-image pairs. (a) Matching results of complete image pairs, (b) Matching details of sub-image pairs

As shown in Figure 3.9, the proposed EW-HOGM algorithm can obtain evenly distributed tie points in all four image pairs because of the adopted grid-matching strategy and the improved recall (Figure 3.10(a)). Moreover, the tie points in image pairs 1 and 3 are clustered (Figures. 3.9(a) and 3.9(c)). By contrast, the tie points in

image pairs 2 and 4 have more even distributions (Figures. 3.9(b) and 3.9(d)). This phenomenon is determined by both the underlying principle (i.e., matched triangles are similar) and the landscapes covered by the images. Many high buildings are seen in image pair 1, and a large topographic relief is evident in image pair 3. Thus, the matched triangles mostly have small areas. The smaller the areas of the matched triangles are, the more similar the triangles are. The topographic relief in image pairs 2 and 4 is relatively small; thus, the tie points are evenly distributed. The matching results of sub-image pairs intuitively show that EW-HOGM can obtain numerous tie points in poor textural images (Fig. 3.9, right column). These results are attributed to EW-HOGM seeking point correspondences that are similar in geometric structures and radiometric appearances, and the EW tensor makes the matching robust to outliers.

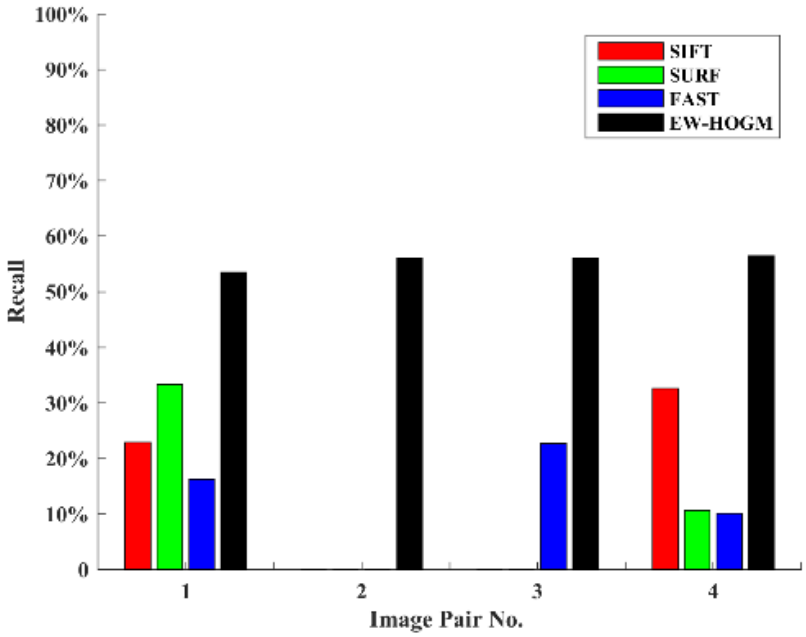
Numerous high buildings are found in sub-image pair 1. These buildings cause occlusions as well as discontinuous and shaded textures. The EW-HOGM algorithm obtains abundant tie points because the geometric similarities compensates for the radiometric distortions in the shaded regions. Moreover, the tie points in sub-image pair 1 are mostly located on the ground, which is also a reaction of the underlying principle that matched triangles are approximately similar. The satellite altitude is higher relative to the elevation variation of the Earth's surface. Thus, the ground surface can be regarded as a flat surface in small areas. Consequently, the local distortions in two conjugate patches are small and can be approximated via similarity transformation. However, the similarity transformation model cannot be used in the local images covering high buildings, and thus the tie points locate on the ground.

The image contents of sub-image pair 2 are forests. The self-similarities of trees causes repetitive textures in the images. Moreover, the two images are acquired at different moments. Thus, significant radiometric differences are observed. However, EW-HOGM obtains a certain number of tie points. The algorithm simultaneously uses geometric and radiometric constraints, and the geometric constraints play a role similar to that of radiometry when radiometric changes are relatively large.

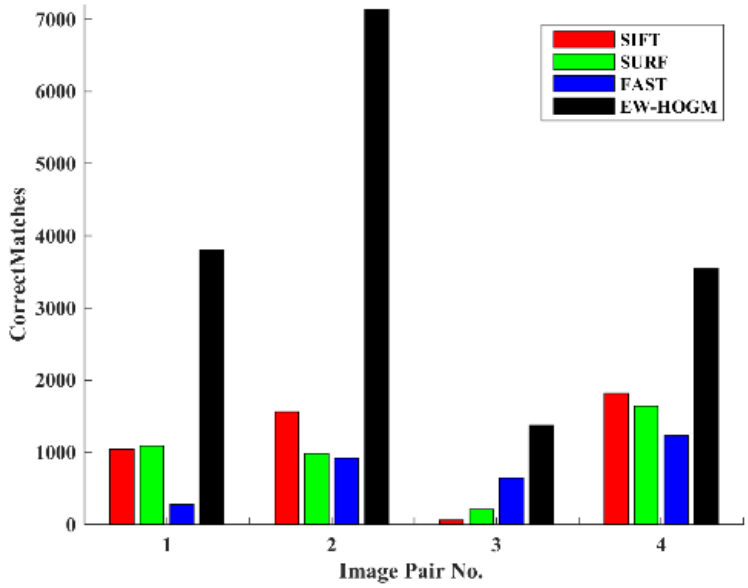
Sub-image pair 3 suffers from homogeneous textures caused by the desert, as well as

geometric distortions caused by a large topographic relief. EW-HOGM still works effectively. The similar result is obtained in sub-image pair 4, in which both homogeneous textures and large distortions exist.

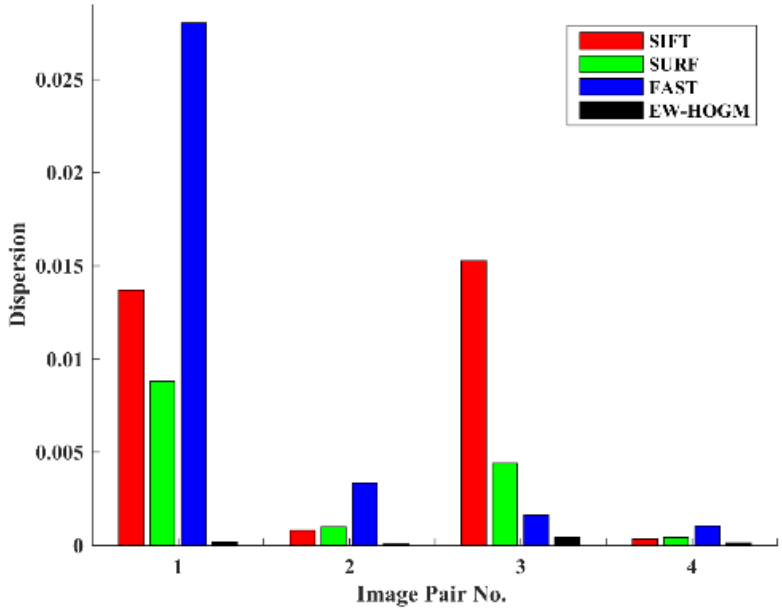
The quantitative experiment results are shown in Figure 3.10.



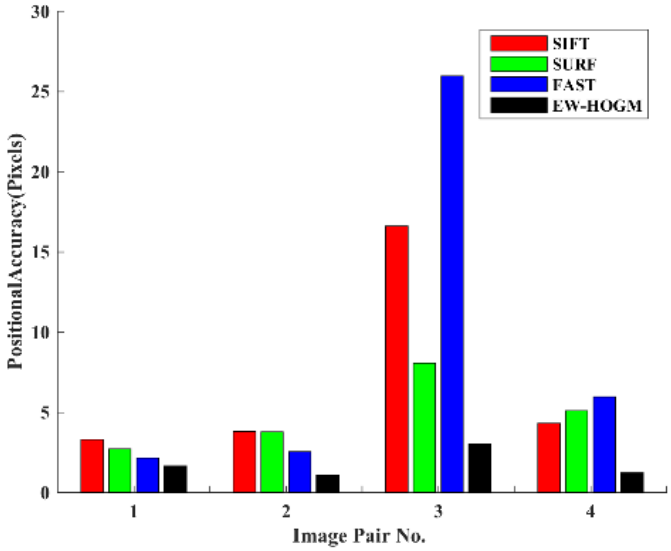
(a)



(b)



(c)



(d)

Figure 3.10 Comparison results of the four matching algorithms. (a) recall, (b) correct matches, (c) dispersion, (d) positional accuracy

As shown in Figure 10(a), traditional tie point matching algorithms, such as SIFT, FAST, and SURF, demonstrate different recall performances in poor textural image matching and the overall recall is lower than 35%. In general, tie points obtained

using radiometry-based algorithms are determined via the NNDR of radiometric descriptors. NNDR has two drawbacks in matching poor textural images. First, the descriptor distance of two matched features is insufficiently small when encountered with shaded, discontinuous or changed textures. The first and second sub-image pairs are the sample cases. The sub-image pair 1 consists of shaded and discontinuous textures caused by high buildings, as well as textural changes caused by different shooting times. The second sub-image pair suffers from repetitive and radiometry-changed textures. Thus the matching recall of the radiometry-based algorithms is unstable and varies from 0 to 35%. Second, descriptors are insufficiently distinctive to distinguish true matches from false ones in matching homogeneous and repetitive textures. The representative cases are last three image pairs, especially in sub-image pair 2, where all the radiometry-based algorithms are failed. NNDR-based matching algorithms, such as SIFT, SURF, and FAST, cannot overcome these two defects in matching poor textures, thereby leading to a low matching recall. However, the recall results of EW-HOGM are stable and are all higher than 50% although these four image pairs consist of different poor textural types. The higher matching recall benefits from the third-order affinity tensor and the EW strategy, the tensor encodes with geometric and radiometric information. Geometry describes the intrinsic relations of feature points, meanwhile radiometry represents local appearances, and the EW strategy makes matches robust to outliers. EW-HOGM avoids the NNDR rule and tends to find the best matches in geometry and radiometry, thereby leading to a high matching recall, particularly in the second and third sub-image pair, where SIFT, SURF, and FAST are almost all failed.

Moreover, EW-HOGM outperforms the other algorithms in correct matches because of high matching recall, especially in the third image pair (shown in Figure 3.10(b)). The third image pair consists of numerous homogenous textures caused by deserts, as well as considerable geometric distortions caused by different camera angles. Therefore, these poor textures result in less correct matches for SIFT, SURF, and FAST. However, homogeneity and distortions have fewer effects on EW-HOGM. The dispersion of EW-HOGM is stable and remains at approximately 0.001 (Figure

3.10(c)). The other three algorithms have various dispersions that are determined through image textures. The positional accuracy of EW-HOGM is more stable and higher compared with those of the others because of the improved distribution and the higher number of correct matches (Figure 3.10(d)). In addition, two instructive results are presented in Figure 3.10. First, matching homogeneous textural images (image pair 3) is the most challenging task among the four pairs of poor textural images. The matching recall, correct matches, dispersion, and positional accuracy of image pair 3 are relatively lower than those of the other image pairs. The essence of these results is the low signal-to-noise ratio of homogeneous textures, which causes a low repetitive rate of image features. Second, positional accuracy mainly depends on the number of correct matches and the dispersion of tie points. As shown in Figure 3.10, EW-HOGM obtains the most correct matches and the best dispersion in image pair 2, thereby leading to the highest positional accuracy among these image pairs. By contrast, SIFT obtains the least correct matches and the worst dispersion in image pair 3, thereby leading to a relatively lower positional accuracy among these image pairs.

3.4 Summary of this chapter

Image matching is a fundamental step in remote sensing image registration, aerial triangulation, and object detection. Although it has been well-addressed in rich textural images, it is remains a challenge in matching poor textural images because of homogeneous, repetitive, occluded, and discontinuous textures. Conventional algorithms are prone to failure because they use only radiometric information. This study presents a novel matching algorithm that integrates geometric and radiometric information into an affinity tensor and utilizes the EW strategy to address outliers. The proposed method involves four steps: feature detection, graph building, tensor construction, and high-order graph matching. In feature detection, UR_SIFT, SIFT, and ANN are applied, and the extracted image features are regarded as graph nodes. Then, the affinity tensor is built with its elements representing similarities of nodes and triangles. The EW strategy is embedded into power iteration to make matching

robust to outliers. The proposed method has been evaluated using four pairs of remote sensing images covered with four different poor texture types: high buildings, forests, deserts, and bare lands. Compared with traditionally used feature matching algorithms, such as SIFT, SURF, and FAST, the proposed EW-HOGM can achieve reliable matching results in terms of matching recall, correct matches, dispersion, and positional accuracy of the experimental data.

When the tolerance to outliers of EW-HOGM is considered, the algorithm can also be used in shape matching and 3D cloud registration, where the amount of outliers is massive. Moreover, EW-HOGM can also be applied to gross error detection if it is appropriately modified. However, a few problems should be further addressed. The computational complexity of EW-HOGM is high because the huge size of the affinity tensor increases computing operation in power iteration. Further research can introduce new strategies to make the tensor sparser and reduce computational complexity in power iteration. In addition, power iteration can also be implemented in a GPU-based parallel computing framework (*Silva, et al., 2016*) which can immensely speed up power iteration.

Chapter 4
Dense image matching

In this chapter, I will introduce the basic concept and algorithms used in our dense matching approach. In the first chapter, I have already introduced our matching strategy. In this chapter, I will detailed describe our approach by 4 sections, first one is preprocessing; second one is optical flow field based coarse matching; then dual constraint based fine matching; the last one is mismatching elimination.

4.1 Preprocessing

In this sub-section, I will introduce the preprocessing techniques used in our approach. Generally, we use feature based matching method to obtain the seed points for the coarse matching step, then using Homography to find the overlap between the stereo image pairs. As is known to us all, the corresponding point pairs can only exists in overlap area. By overlap detection, we can reduce the redundant calculations, improve the computation efficiency.

Homography is the corresponding relationship between the corresponding point pairs in different plane. In aerial image, the photography position is unstable. The simple affine cannot provide the reliable describe between the image pairs. We use the accurate Homography to calculate the rotation parameters between the aerial image pairs. Homography matrix can be described as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.1}$$

where (x, y) and (x', y') are presented the corresponding points homogeneous coordinates; **H** is the Homography matrix. It can be described as

$$\mathbf{H} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{4.2}$$

Obviously, the freedom of **H** is 8, which means we need at least 4 corresponding point pairs to solve the elements in **H**. After extracting image seed point, the corresponding point pairs are far more large than 4. We use least square adjustment method to improve the sloving accuracy of the Homography matrix.

Normally we use the down sampling image to calculate the Homography matrix. The scaling ratio is calculated by Equation (4.3).

$$\sigma = \frac{Max(w,l)}{p} \tag{4.3}$$

Where *w* and *l* are the width and the height of the image. *Max* is the logic function to determine the large value between the 2 parameters; *p* is the down sampling image's pixel numbers. By experiments we found that *p* can be set 256.

Down sampling may cause the noise on the image. So before the down sampling we use Gaussian filter to smooth the image.

4.2 Coarse matching

In this subsection, I will introduce the key concept of our approach. The most important thing in dense image matching is to find each pixel's corresponding pixel between the stereo image pairs. Like all the image matching methods the similarity measures are used to find the relationship between the features and areas. So I come up with an idea, why couldn't we use some methods to simulate each pixel's movement between the stereo image pairs, and use the movement field to guide the iamge matching. In computer vision field, optical flow is an efficient method to describe the pixel movement in video sequence.

4.2.1 Optical flow

Optical flow was first proposed by Gibson in 1950. It present the real object movement on the observation coordinates. It described as the instant speed of the

pixel movement. And the optical flow field is the gray scale movement on image surface. The essence of optical flow research is using the pixels' temporal intensity changes and correlation degrees to describe pixel's position change in sequence images. Nowadays optical flow field is widely used in super-resolution image reconstruction (*Elsd, 1996; Baker and Kanade, 1999*), image segmentation (*Advi, 1985; Galic and Loncaric, 2000*) and robot navigation area (*Revathi, and Hemalatha 2012*). All those works are involved with image matching.

Here we introduce our coarse matching step.

As introduce in subsection 4.1, we have already obtain the overlap area in the stereo image pairs also the high reliable seed points by preprocessing procedures. So we can use this information as the prior knowledge to calculate the overlap area's optical flow field.

There are 2 types of optical flow's calculation methods. The first one is using feature-based matching methods to find corresponding points and calculate the displacements between stereo image pairs. Using the displacement to describe the pixel movement. Second one is based on calculating the pixel's gray gradient find the searching area's small difference then calculates the optical flow.

In our approach, we have already obtained the corresponding feature point pairs and the overlap area. To each corresponding point pairs, we can calculate is optical flow as follows:

$$\vec{u}_i = [x_i - x_i' \quad y_i - y_i'] \tag{4.4}$$

Where $(x_i, y_i), (x_i', y_i')$ are the pixel coordinates of seed points on the left image and the right image, respectively.

4.2.2 Optical flow field based coarse matching

In section 4.2.1 we use the seed points get some discrete optical flows. In order to describe each pixel's movement the thin optical flow field is not enough. Traditional

optical flow measurements assume that in local region the brightness of the pixel will not change and the pixel's move range is small. When dealing with the aerial images. Those assumptions are not existed. Because of the photography position changes the image will have different kind of distortion. Also the occlusion may cause by shelter so brightness will also changes.

In this section, we use a Multi-level B-spline interpolation (Leeet al, 1997) to simulate the optical flow field.

Assume that $\Omega = \{(x, y) \mid 0 \le x < m, 0 \le y < n\}$ is the xy plane of the optical field. $\mathbf{P}(x, y, \Delta x)$ is the set of those discrete seed points. In order to approximate P, we can build a bi-cubic B-spline interpolation function f and a new interpolation grid Φ as shown in Figure 4.2, Φ is the $(m+3)\times(n+3)$ interpolation grid overlapped with the image overlap area.

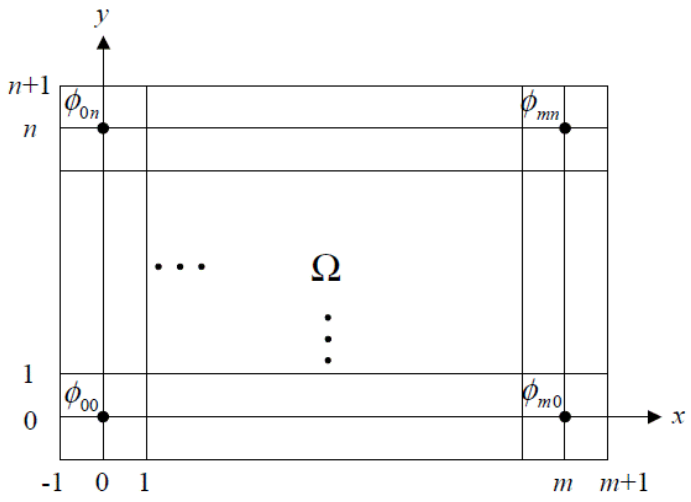


Figure 4.2 The relation schema of interpolation grid

The node value of the interpolation grid is the interpolation weight coefficient to the seed's points. So we represent our interpolation problem to figure out the optimized interpolation grid Φ.

As Figure 4.2 shown, the node value denote as follows:

$$\Phi_{ij} = \frac{\sum \Phi_c w_{ab}^2}{\sum w_c^2} \tag{4.5}$$

where,

$$\Phi_c = \frac{w_c z_c}{\sum_{a=0}^{3}\sum_{b=0}^{3} w_{ab}^2};$$

$$w_c = w_{ab} = B_a(s)B_b(t);$$

$$a = i+1-\lfloor x \rfloor, b = j+1-\lfloor y \rfloor, s = x-\lfloor x \rfloor, t = y-\lfloor y \rfloor;$$

(x_c, y_c, z_c) is the seed point coordinates in set P. And the uniform bi-cubic B-spline basis function denote as follows:

$$\begin{aligned} B_0(t) &= \frac{1}{6}(1-t)^3 \\ B_1(t) &= \frac{1}{6}(4-6t^2+3t^3) \\ B_2(t) &= \frac{1}{6}(1+3t+3t^2-3t^3) \\ B_3(t) &= \frac{1}{6}t^3 \end{aligned} \qquad 0 \le t < 1 \tag{4.6}$$

We use the above formulas to and Gaussian function to weight the coefficient by the distance between the interpolation grid nodes and the seed points. Then f can be denoted as:

$$f(x,y) = \sum_{k=0}^{3}\sum_{k=0}^{3} B_k(s)B_l(t)\Phi_{(i+k)(j+l)} \tag{4.7}$$

In order to reduce the interpolation error and make our result more reliable, we improve our model to multi-level. Assume the first interpolation grid is Φ_0, then we can easily calculate the interpolation function f_0 as mentioned above. Obviously, f_0 is the approximation to interpolation P. Hence we can calculate the difference $\Delta_c^1 = z_c - f_0(x_c, y_c)$ between them. Then we use the fine interpolation grid to approximate $P_1 = (x_c, y_c, \Delta_c^1)$ by calculating the interpolation function f_1. Then we can accumulate $f_0 + f_1$ and obtain a smaller difference to P. The difference denote as $\Delta_c^2 = z_c - f_0(x_c, y_c) - f_1(x_c, y_c)$. Through iteration we can denote that $f = \sum_{k=0}^{h} f_k$, here k is the iteration times.

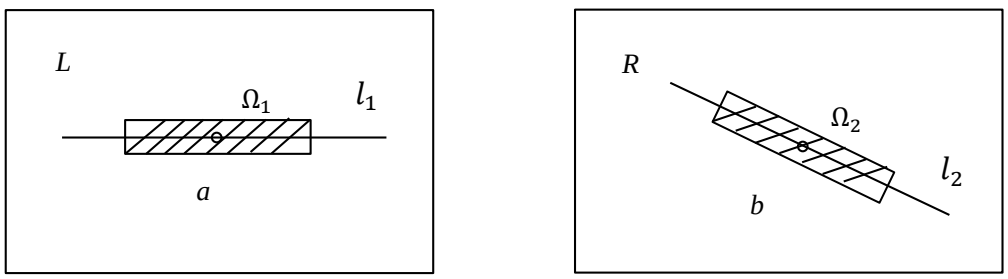
After optical flow field simulations, we can obtain the coarse matching result. Because B-spline is very smooth, so the coarse matching results' accuracy is ordered by the distance.

4.3 Fine matching

In this section I will introduce the concept and theories used in our approach. We use the coarse matching result as the guidance information. And use dual constraint to rectify the fine matching area. Reduce a lot of redundant computation to improve our approach's efficiency and accuracy.

4.3.1 Dual constraint rectify

In this subsection, I will introduce the first step of our fine matching. The concept of our dual constraint rectify is based on epilolar line constraint and affine transform. In the preprocessing step, we using PCA-SIFT matching combined with least square matching obtained some high accurate and reliable corresponding point pairs. By using these point pairs, we can easily calculate epilolar line of each point pairs in the left and right images. For each pixel in the left image, if it has not matched in the preprocessing step we use the optical flow field to find the approximate corresponding point in right image. Then using the epilolar line and expand it with the neighbor 2 pixels to build up the template and the searching window, shown as Figure 4.3



Left image Right image

Figure 4.3 Select the fine matching windows.

In Figure 4.3 the shelter area Ω_1 and Ω_2 is the selected matching area. l_1 is the epilolar line of point *a* in the left image, l_2 is the corresponding epilolar line of point *b* in right image. *a* and *b* are corresponding point pairs. Normally we can use 3 kind of information to describe a point: position, scale and angle. We denote the corresponding point pairs as $a(\mathrm{p}_a, s_a, \theta_a)$ and $b(\mathrm{p}_b, s_b, \theta_b)$, and denote the dual constraint rectify matrix as Equation (4.9).

$$A = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ \sin\vartheta & \cos\vartheta \end{bmatrix} = SR \tag{4.9}$$

Where S_x denotes the scaling coefficient on *x* direction; S_y denotes the scaling coefficient on *y* direction;$S = s_a/s_b$; ϑ denotes the corresponding windows relative rotation angle. The rectify procedure can be shown as Figure 4.4.

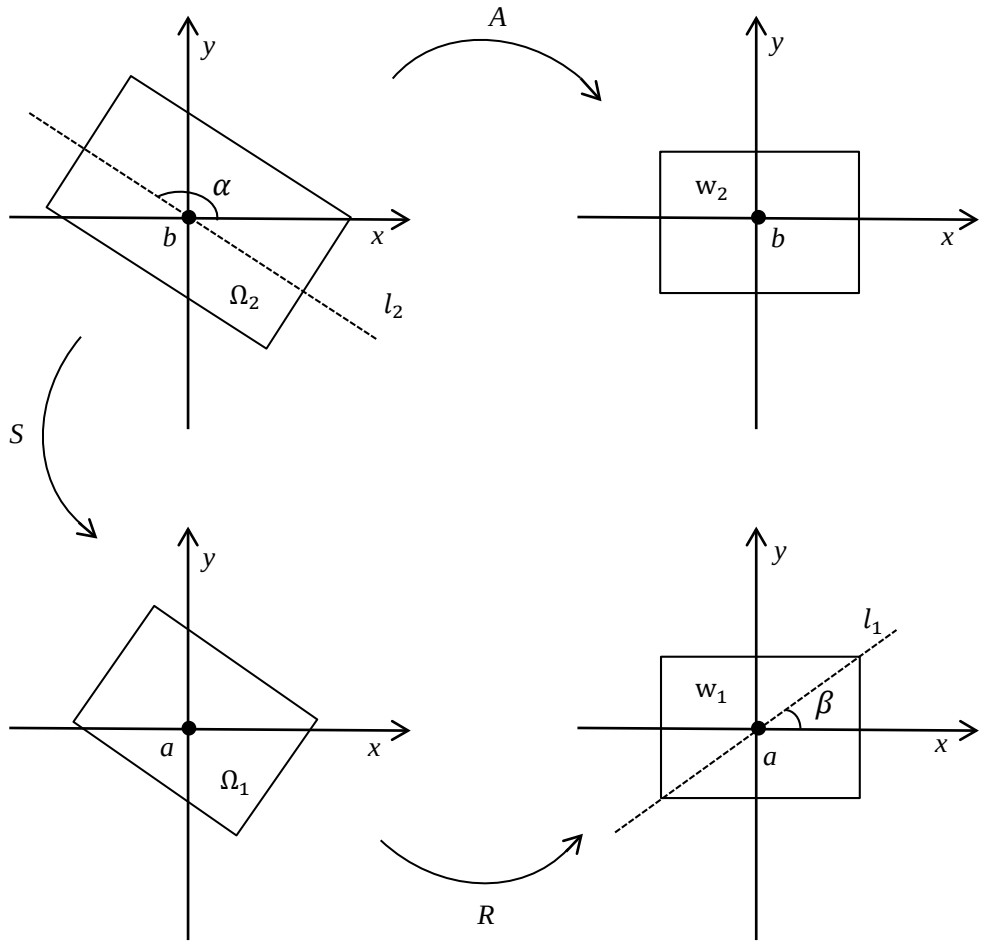


Figure 4.4 Dual constraint rectify

In calculation we use the distance between the selected points and the seed point to weight the scale coefficient. The weigh function is same as MBA we described in section 4.2.2; Assume the number of the seed point pairs is *n.* The *i*-th seed point's contribution to selected points is ρ_i, We calculate the scale coefficient as Equation (4.10).

$$s = \sum_{i=1}^{n} \rho_i s_i \tag{4.10}$$

We can use the dual constraint rectify matrix to affine the matching area. The final matching image blocks are shown as W_1 and W_2 in Figure 3.4. In this way we can reduce a lot of redundant computation.

4.3.2 Fitting position.

After dual constraint rectify, we use NCC (Normalized cross-correlation method) to do the fine matching. Then we select the 3×3 area centered by the matched point. Do polynomial fitting according to the NCC similarity. Determine the extreme point as the optimized matching point. The used polynomial is as follows.

$$f(x,y) = a_0x^2 + a_1y^2 + a_2xy + a_3x + a_4y + a_5 \tag{4.11}$$

In Equation (4.11), we denote $f(x,y) = 1/NCC(x,y)$ and use Gaussian function to weigh each pixel's contribution by distance. We obtain the result by using least square adjustment method.

$$\begin{aligned} x_{opt} &= (2a_1a_3 - a_2a_4)/(a_2^3 - 4a_1a_0) \\ y_{opt} &= (2a_0a_4 - a_2a_3)/(a_2^3 - 4a_1a_0) \end{aligned} \tag{4.12}$$

4.4 Mismatching point elimination

As is known to all, every matching method has mismatching in its result, a good matching approach needs to provide an efficient mismatching elimination procedure

to improve the result reliability. The classic mismatching elimination algorithms are data-snooping method and iteration method with variable weights, besides, least median of squares (*Massart, 1986*), MLESAC (*Torr and Zisserman, 2000*) and RANSAC methods are robustness to mismatching elimination. In this section, I will introduce the RANSAC based mismatching elimination algorithm used in our approach.

4.4.1 RANSAC.

RANSAC (Random Sample Consensus) was first proposed by Fischer and Bolles in 1981. The basic assumption of RANSC is the test sample involved with both inliers and outliers, the outliers are caused by noise, wrong assumption or miscalculations. If a correction test sample is given, the corresponding parameter model can be determined. The RANSAC theory can be described as follows:

(1) A mathematical model is selected. Assume that the model can be determined by at least n parameters. An observation set P is given, contained m observations, $m>n$. Randomly select a sub set S from P. S involved with n observations. Obviously, S can determine a mathematical model. Denote the mathematical model as M. Use all the observations in P to test model M. Set a test threshold. Passed observations can build a new set S_1. We denote S_1 is the model M's consistent set.

(2) If the observation number of S_1 is larger than the threshold t, we can use S_1 determine the new mathematical model M_1.

(3) If the observations number of S_1 is less than the threshold t, we can use first step to find the S_1 can meet the demands that the observations number of S_1 no less than t. When the iteration time achieves a certain number, still can't meet the condition we set, then RANSAC fails.

4.4.2 RANSAC based relative orientation

In our approach, the test sample has a very large dataset. Using the traditional

RANSAC may have numerous iterations. We assume the non-mismatching probability of our result is *p*, the probability of that we do k times random selection from our results get at least *n* non-mismatching point pairs is *Q*. Then $(1-p^n)^k$ denote the probability of the selected *k* subsets each one contained mismatching point pairs. We can denote *k* as follows:

$$k = \frac{\ln(1-Q)}{\ln(1-p^n)} \tag{4.13}$$

From Equation (4.13) we find that RANSAC iteration times are not affected by the number of observations according to our assumption. Normally the mismatching rate in PCA-SIFT is around 20%. As our image dense matching approach use the PCA-SIFT results as the initial input. The mismatching rate is related to PCA-SIFT. We assume the rate is 30%. Relative orientation needs at least 5 corresponding point pairs. If the correct probability is expected to reach 99%, the *k* value calculated by Equation (4.13) is 20.

In photogrammetry, the corresponding points in a stereo image pairs fulfills the coplanarity condition. So we use the coplanarity condition as the mathematical model to operate RANSAC. Coplanarity condition is shown as follows.

$$F = \begin{vmatrix} B_x & B_y & B_z \\ X & Y & Z \\ X' & Y' & Z' \end{vmatrix} = 0 \tag{4.14}$$

where $\mathbf{B} = \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}^{\mathrm{T}}$ denotes the photography baseline; $\mathbf{m} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x \\ y \\ -f \end{bmatrix} = \mathbf{p}$; $\mathbf{m}' = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \mathbf{R}\begin{bmatrix} x' \\ y' \\ -f' \end{bmatrix} = \mathbf{Rp}$; (x, y) and (x', y') denote the image coordinates of the corresponding points; f and f' denote the principle distance of left and image, respectively; **R** is the orthogonal transform matrix from right image to left image.

Linearizating Equation (4.14) by Taylor's series expansion can obtain an error function as follows:

$$v_Q = \frac{\partial F}{\partial \varphi} d\varphi + \frac{\partial F}{\partial \omega} d\omega + \frac{\partial F}{\partial \kappa} d\kappa + \frac{\partial F}{\partial B_x} dB_x + \frac{\partial F}{\partial B_y} dB_y - Q \tag{4.15}$$

If the observation number is larger than 5, least square adjustment method can be used to solve the error Equation (4.15). If one point's residual is larger than the 3 times of root mean square errors, this point can be accounted as the mismatching point.

4.5 Quality assessment

Comprehensive evaluation of the quality of the image dense matching point clouds is carried out in this paper, mainly from two aspects: specifically the subjective visual effect of the point clouds and quantitative analysis of the objective indicators. Subjective evaluation performs 3D visualization of the dense matching point clouds and compares it with the presently recognized SURF and PMVS matching effect. Objective evaluation employs quantitative analyses of quantitative indicators, such as the matching success rate, point cloud accuracy and matching reliability.

1. Matching success rate

In this paper a single stereo image pairs is taken as a statistical unit, and the ratio of the total number of image points to the number of obtained dense matching points in the stereo pair overlapping region is calculated according to Equation (4.16) as a measure of matching success rate (msr) of dense image matching. The higher the matching success rate, the better the dense matching effect.

$$msr = \frac{\text{Total number of matching points}}{\text{Total pixel number in overlap aera}} \times 100\% \tag{4.16}$$

2. Reliability

Vertical parallaxes of the dense matching point clouds of a stereo image pair are calculated one by one according to the relative orientation elements; when no mismatching point exists, the residual errors of the vertical parallaxes of all points should be in accordance with a normal distribution and almost close to 0. For one

point, it can be known from the reliability theory of Baarda, that, when the significance level α=0.1% is taken, the residual error of its vertical parallax should not be greater than $3.29\sigma_0$. Here, σ_0 is the unit weight root mean square error in relative orientation. With this criterion, the dense matching point clouds are transverse, the statistics of the overrun points should be carried out, and the ratio of the number of matching points in a stereo image pair to number of overrun points is calculated according to Equation (4.17) as a reliability measure of the dense image matching. The smaller the ratio is, the higher the dense matching reliability is.

$$r = \frac{\text{Number of mismatch points}}{\text{Total number of matching points}} \times 100\% \tag{4.17}$$

3. Accuracy in imagery

As well known, relative orientation is an analytic calculation process that generates pair-to-pair intersections of corresponding image rays on a stereo pair, and the goal of realizing a reasonable error distribution of the observed value of the image point coordinates by eliminating the parallax of the stereoscopic model to the minimum margin. As the basis for its calculation is only the observation value of image point coordinates, no non-photogrammetric measurement is involved. Hence, the error σ_0, in relative orientation can be regarded as a measure of matching point precision in imagery.

However, as the quantity of dense matching points in each stereo image pair is enormous, the relative orientation and gross error elimination are only conducted on the seed points set, through the continuous relative orientation model with variable weight and iteration methods. After the relative orientation elements are obtained, the residual errors of the vertical parallaxes of the dense matching point clouds are calculated one by one, all points greater than $3.29\sigma_0$ are removed, and the root mean square error σ of residuals of the vertical parallaxes of the other points is made. The smaller the σ is, the higher the dense matching accuracy is.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta q_i^2} \tag{4.18}$$

Where, n is the number of dense matching points; Δq_i is the residual error of the vertical parallax of the *i*-th dense matching point.

4. Accuracy in ground

For dense matching point clouds with the mismatching elimination, a point-wise calculation of their 3D ground coordinates is implemented according to the dual-image forward intersection principle, and discrete 3D point clouds, namely DSM can then be generated. Those pass points obtained by GPS-supported bundle block adjustment are taken as check points, and their neighborhood points in the DSM are searched according to their planimetry position. As OFFDIM realizes pixel-by-pixel dense matching effect, the nearest neighbor interpolation method is used in this paper to extract the elevation value of the point closest to check point in DSM, which can be used to obtain the elevation errors of check points. The accuracy m of the DSM can be calculated according to Equation (4.19) as the measure of the object accuracy of dense matching point clouds. The smaller the m value, the higher the dense image matching precision.

$$m = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta h_i^2} \tag{4.19}$$

Where, n is number of check points; Δh_i is height error of the i-th check point.

4.6 Experiment results and discussions

4.6.1 Experiment description

In this paper, the experiments are conducted on a data set with true color digital aerial images carrying GPS navigation data and photographed by UAV. Images were taken in May, 2016, and relevant technical parameters are listed in Table 4.1.

Table 4.1 Technical parameters of images in experimental projects

Items	Parameters
Aircraft	Unmanned Aerial Vehicle (UAV)
Aerial camera	PhaseOne IXU-1000

CCD size	4.6 μm
Ground sample distance (GSD)	7 cm
Focus length	51.21293 mm
Flight altitude	1290 m
Frame	11608 × 8708 pixels
Longitudinal overlap	60 %
Lateral overlap	30 %
Number of strips	8
Number of images	88
Ground control Point (GCP)	18
Pass points	55701
Block area	2.8 × 2.8 km2
Maximum topographic relief	54 m
GPS data update rate	1 second
GPS offset	0.1020 m, 0.0000 m, 0.3160 m

Our self-developed full-automatic digital photogrammetry system, named Imagination, was taken as data processing platform in the experiment in this paper. First, the automatic image measurement subsystem Imagination-AMS was used for the automatic turning point measurement of the pass points of all images, and artificial stereoscopic observation of image coordinates of all ground control points (GCPs) was carried out. Then the GNSS camera station positioning subsystem Imagination-GNSS was used to obtain the dynamic precise point positioning based on GPS carrier phase observations recorded during the aerial photography process to obtain the 3D coordinates of all camera stations. Next, the camera station coordinates were regarded as weighted observations, and the simultaneous combined adjustment subsystem Imagination-BBA was used to perform the GPS-supported bundle block adjustment. On this basis, for 4 strips (148~138, 122~132, 114~104 and 92~102), a total of 44 images were photographed in the west-east direction during the flying mission. The DEM automatic extraction subsystem Imagination-DEM was used to

implement the dense image matching based on optical flow field, and DSM discrete point clouds were automatically generated.

Figure 4.5 shows the distribution of 18 obvious surface feature points (such as traffic marker lines, wall corners, road intersections, house corners, etc.) observed in the experimental area that were used as GCPs. The GPS net static surveying method was used to accurately measure their ground coordinates, coordinate accuracies of 3 directions reached respectively ±10 cm in ground, and these points could be used as encrypted orientation points and check points for photogrammetric point determination.

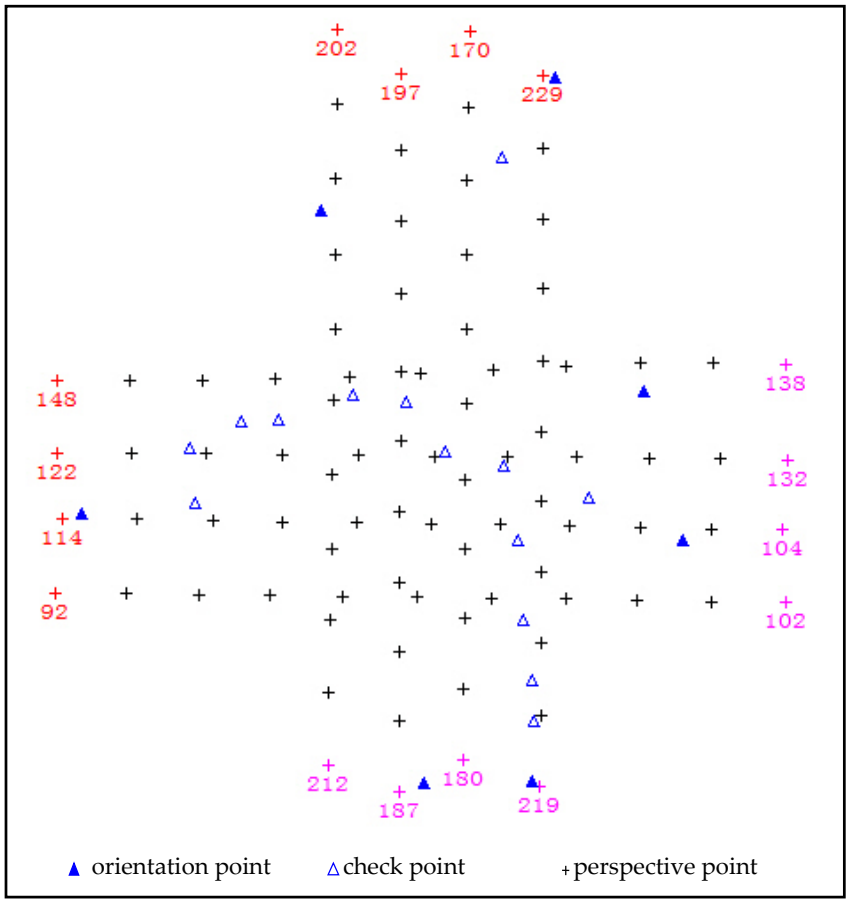


Figure 4.5 Distribution of ground control points in experimental area

For the measured image point coordinates, after Imagination-BBA was used to perform relative orientation and gross error elimination through the continuous relative orientation method with model connection conditions, the average σ_0 of 8 strips were calculated according to the residuals errors of the vertical parallaxes of relative orientation points, which are listed in Table 4.2.

Table 4.2 Accuracy of relative orientation by single image pairs

Strip no.	Number of stereo model	Number of orientation points in single stereo model	Number of joint points between stereo models	Average RMSE		Maximum RMSE		Minimum RMSE	
				(μm)	(pixel)	(μm)	(pixel)	(μm)	(pixel)
148-138	10	99-3683	42-316	0.97	0.22	1.29	0.30	0.75	0.16
122-132	10	135-3970	52-283	1.06	0.23	1.42	0.31	0.82	0.18
114-104	10	203-2663	91-257	0.88	0.20	1.01	0.22	0.74	0.17
92-102	10	205-4163	93-535	0.83	0.18	1.05	0.23	0.61	0.13
202-212	10	262-3374	90-302	0.83	0.18	1.05	0.23	0.72	0.16
197-187	10	224-2073	79-175	0.90	0.20	1.38	0.29	0.69	0.15
170-180	10	161-2522	50-359	0.96	0.21	1.15	0.25	0.78	0.17
229-219	10	266-3452	113-344	1.08	0.23	1.84	0.41	0.65	0.15

It can be seen from Table 4.2 that the relative orientation accuracy of single stereo pairs has no significant difference, and the minimum value, maximum value and mean value among the 80 numbers of σ_0 is ±0.61 μm, ±1.84 μm and ±0.95 μm, respectively. As the vertical parallax of image point is $q = y_1 - y_2$, according to the error propagation law, it is assumed that the measurements of the image point coordinates are mutually independent. In that case, $\sigma_y = \sigma_0 / \sqrt{2} = \pm 0.67$ μm, namely, ±0.15 pixels. That is, in the experimental images selected in this paper, and during the automatic process of identifying the corresponding image points, the measuring accuracy of the image point coordinates could reach the ±0.15 pixel level. In addition, as many orientation points were available in each stereo pair, the average redundant observation component is greater than 0.98. Hence the reliability of relative orientation was very favorable, which contributed to the elimination of the gross errors in the observations of the image point coordinates. The reserved orientation points are taken as seed points, and the dense image matching used in the optical flow field fitting was fully reliable.

A full GCP was respectively set in each of four corners in the test block as shown in Figure 4.5, GPS drift systematic error compensation model was introduced in strip-by-strip in the GPS camera station. Imagination-BBA subsystem was used for

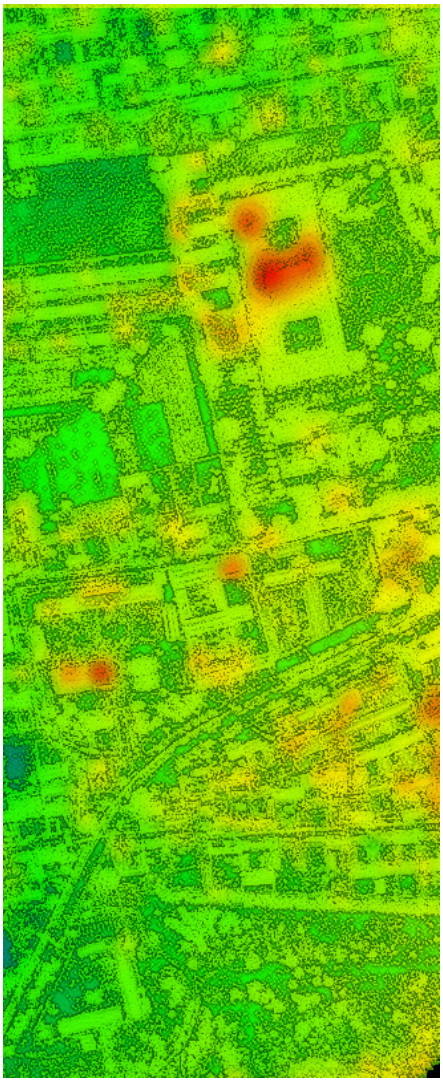
the GPS-supported bundle block adjustment on the experimental image sets (*Yuan, 2008*), and the root mean square error in the unit weights of the image point coordinate observations is ±0.7 μm, which was in close agreement with the ±0.67 μm measuring accuracy of the image point coordinates derived from the relative orientation. Moreover, the actual accuracies of pass points calculated from the 12 full GCPs were ±5.4 cm in planimetry and ±6.9 cm in elevation, that is, both the planimetry accuracy and the elevation accuracy is respectively superior to 1.0 GSD. Taking 55,701 pass points as ground check points, evaluating DTM the point cloud accuracy fully met the applicable requirements.

4.6.2 Dense matching effect

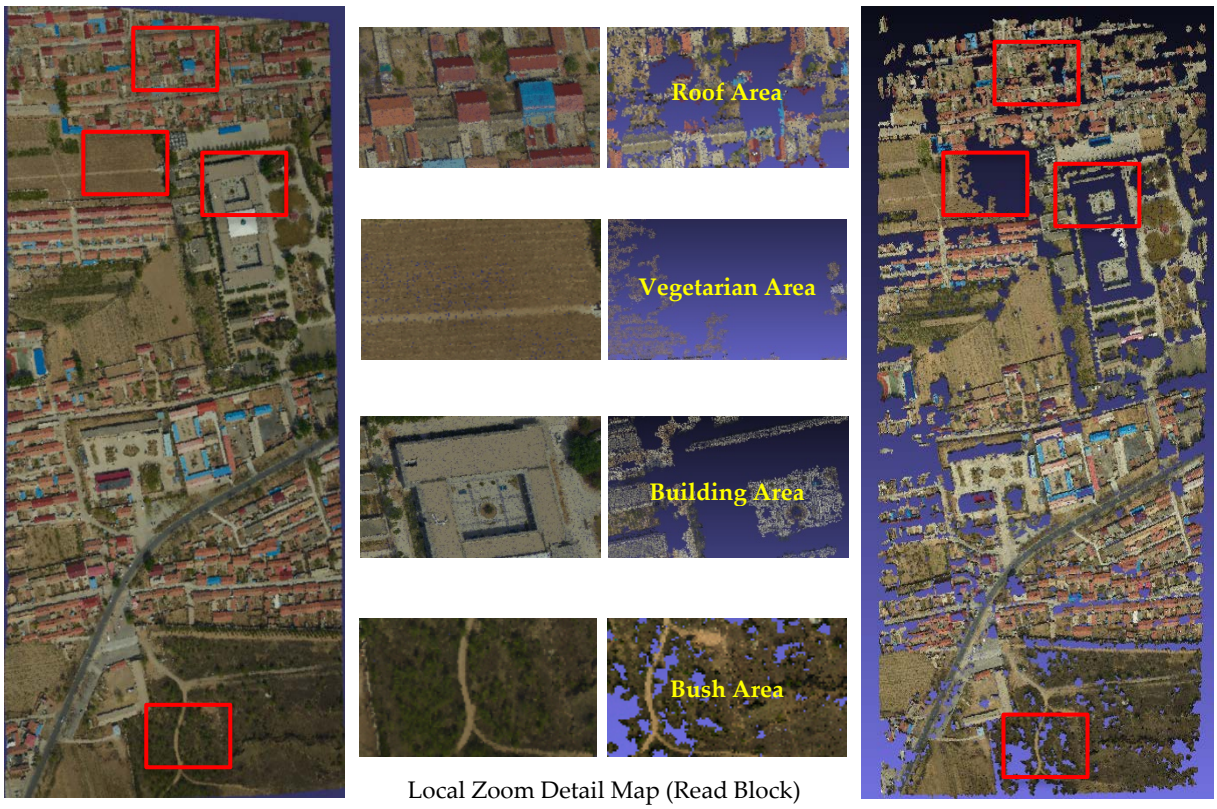
As the experimental images covered large areas, the repetition rate of ground features is high; here, only the 123-124 stereo image pair covering multiple texture feature was taken as an example (the dense matching effects of other stereo image pairs are consistent), and a comparison of the visual effects and subjective evaluation of the matching point clouds between the OFFDIM and PMVS were implemented. As shown in Figure 4.6(a), the stereo pair overlapping region included buildings with complex textures, farmland and bare land with limited texture, as well as bushes, independent trees and roads with repeated textures. Figure 4.6(b) shows the discrete DTM point clouds automatically generated through the OFFDIM dense matching results. This DTM shows the details of all of the kinds of ground features very clearly, and the edges of roads, houses and independent trees are clear and complete, which is in accord with the dense matching point clouds generated via the OFFDIM. The dense matching point clouds generated by PMVS in this region are shown in Figure 4.6(d). It can be clearly seen from Figure 4.6(c) and 4.6(d) that the point clouds generated through the OFFDIM are of high completeness and basically without voids, especially in farmland with a texture shortage and house regions of complex texture (as shown in red block in the figures). It is not hard to see that the OFFDIM could match roofs more completely and could obtain denser matching point clouds in farmland and

grassland regions with limited texture through the elaborate comparison of enlarged portions of these local regions. Even though, for regions such as roads and bare land, the results of the two matching methods are quite similar. Thus, it can be seen that the dense image matching method based on optical flow field are more robust to image textures, and the generated point clouds are much more denser.

(a) Original image



(b) Automatic generated DSM using OFFMID point clouds



Local Zoom Detail Map (Read Block)

(c) Point clouds by OFFDIM (d) Point clouds by PMVS

Local Zoom Detail Map (Read Block)

(c) Point clouds by OFFDIM (d) Point clouds by SURE

Figure 4.6 Dense matching point clouds in single stereo image pairs

4.6.3 Dense matching quality

First, for the dense matching results of 40 stereoscopic images in 4 strips, and the matching success rate of 3D point clouds in each stereo image pair overlapping region, their variation curves are shown in Figure 4.7, the overall matching success rate is higher than 98.5%, and the matching success rate of only 3 stereo image pairs is approximately 97.0%. This result indicated that the matching completeness of the 3D point clouds in the stereo image pair overlapping regions is very high. On the other hand, it can be seen from the matching reliability curves shown in Figure 4.8 that the overall mismatching rate of the 3D point clouds extracted via the OFFDIM is lower than 20%. As there are only 99 seed points in stereo image pair 140-141, and these seed points are concentrated in local region, the mismatching rate is as high as 65%. There are many buildings in the images that make up strip 148-138, and as textures are quite abundant in these areas, the overall mismatching rate is relatively high. The images in strip 114-102 covered farmlands of weak textures, and the overall mismatching rate is low. Thus, it could be seen that the OFFDIM method is robust.

Figure 4.9 reveals the accuracy change rules of the image space and object space of the pixel-by-pixel dense matching results in the stereo image pair overlapping region. It can be clearly seen from the figure that the dense matching point clouds obtained through the OFFDIM can reach the sub-pixel accuracy level in image; the dense matching accuracies of 40 stereo image pairs are all better than ±1.0 pixel, and the optimal accuracy reached ±0.55 pixel. The elevation accuracy in the object spaces is better than ±0.20 m on the whole, namely 3.0 GSD, but the elevation accuracies between different models differed to some degree and fluctuated within 2.0 GSD~3.5 GSD. It is not hard to find, through an elaborate comparison between the accuracy curves of the image space and the object space, that the variation rules of the two are totally consistent, which further verified the robustness of the OFFDIM method

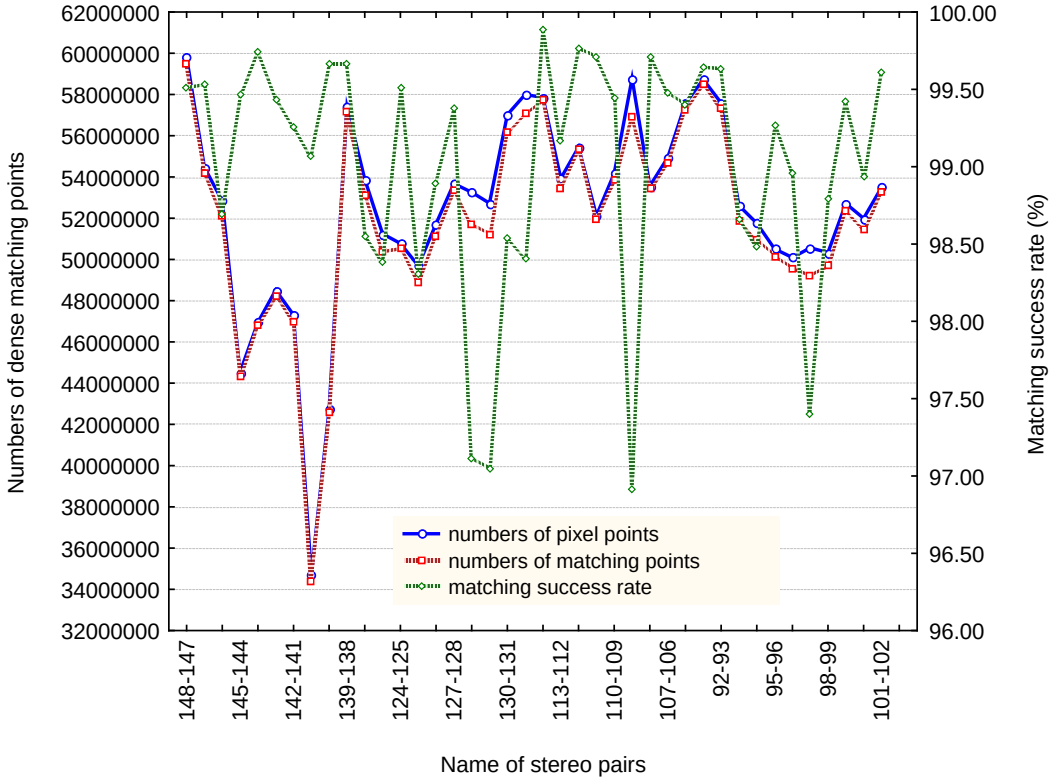


Figure 4.7 Matching success rate in single stereo image pairs

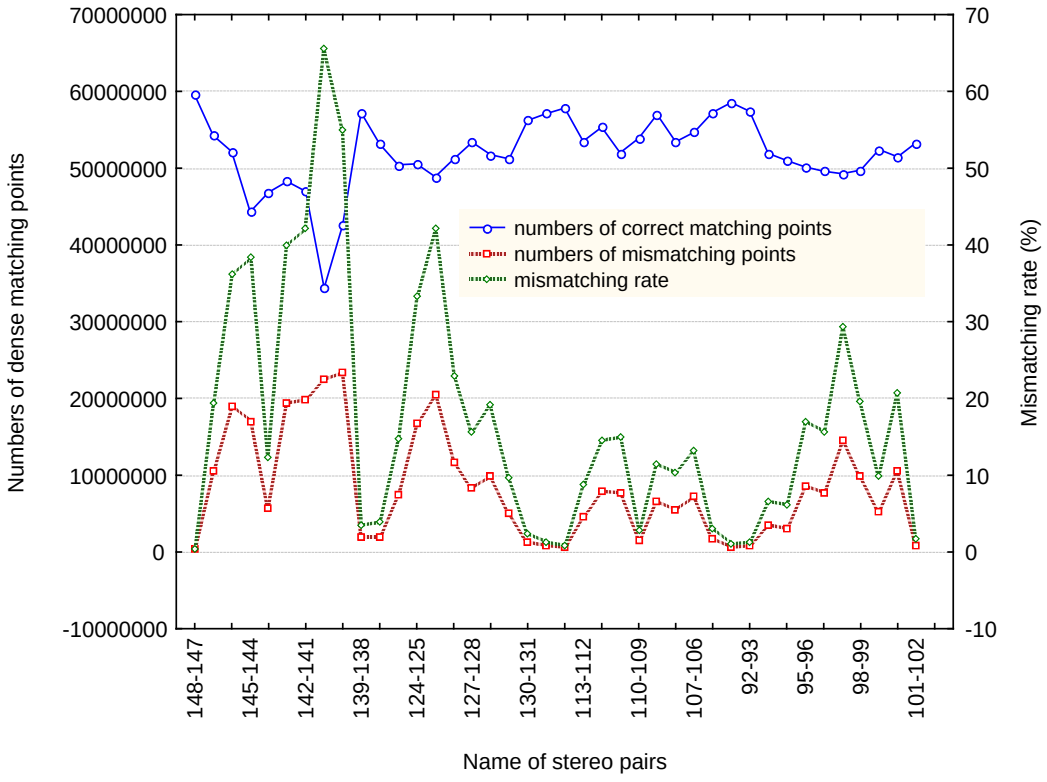


Figure 4.8 Mismatching rate in single stereo image pairs

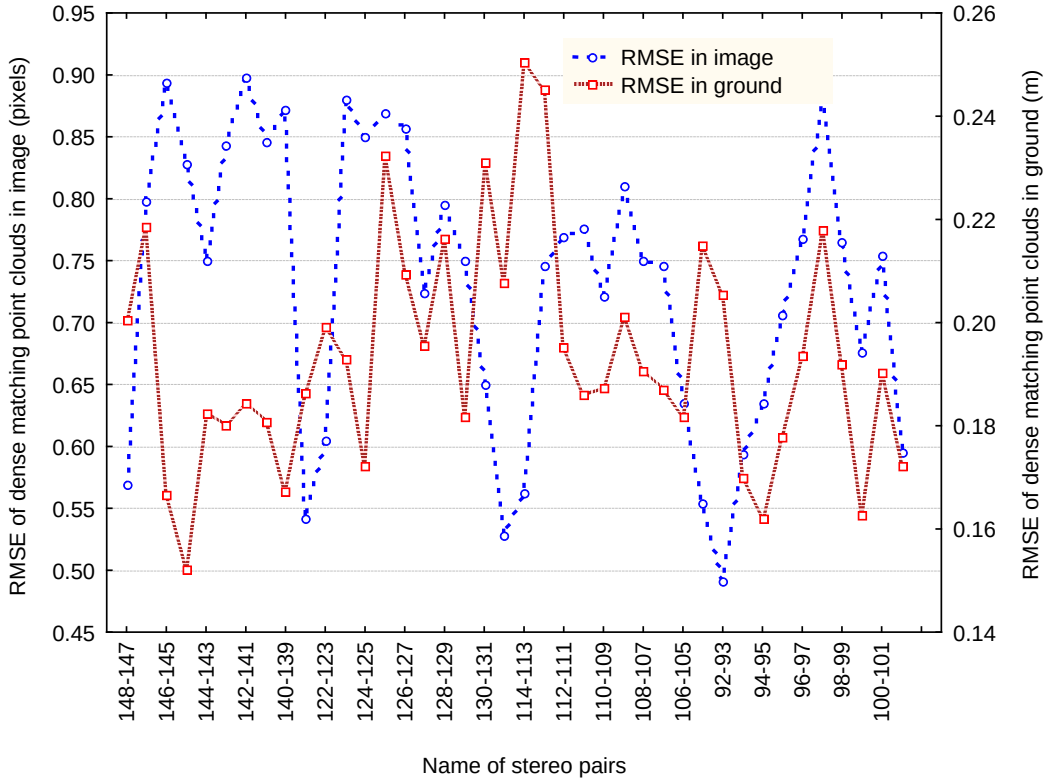


Figure 4.9 Dense matching accuracy in single stereo image pairs

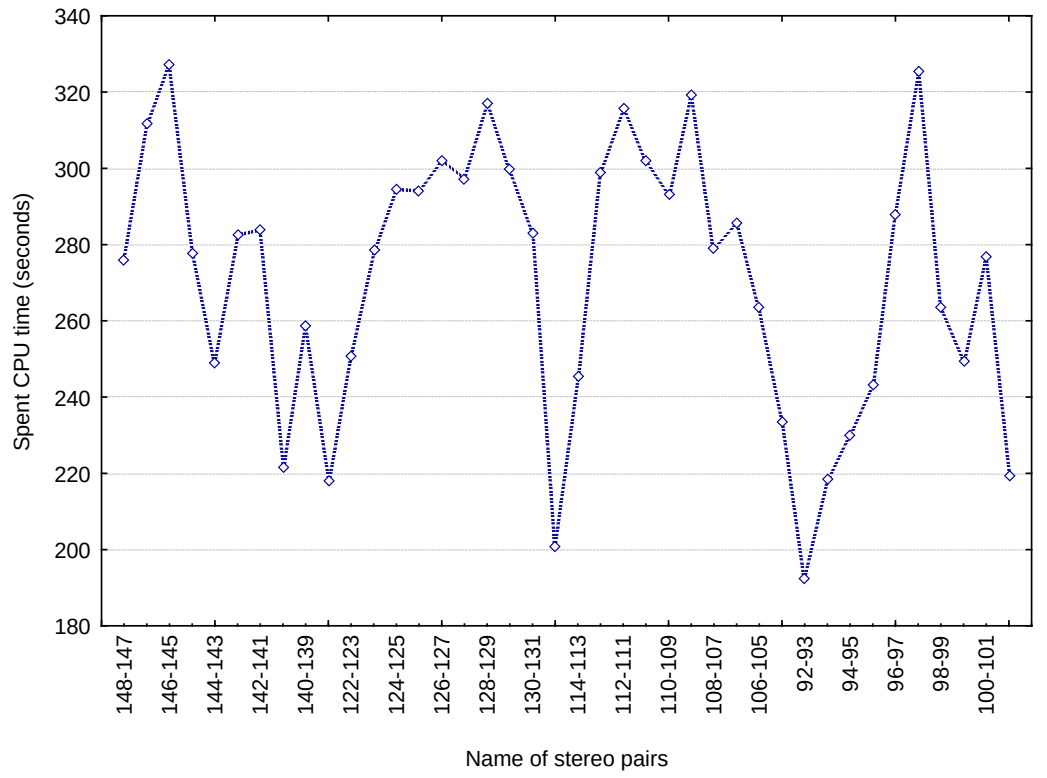


Figure 4.10 Consumed CPU time in single stereo image pair

4.6.4 Dense matching efficiency

All of the experiments in this paper were conducted on an Intel(R) Core™ i7-6700HQ CPU/2.6 GHz/16.0 GB RAM/x 64 portable computer running Windows10. Figure 4.10 shows the variation curve of the CPU time consumed by the dense image matching of 40 stereo image pairs. From a statistical perspective, the CPU time consumed, in processing low-altitude aerial UAV images with pixel sizes of 11608 × 8708 and 60% longitudinal overlap, the dense matching time of a single stereo image pair is within 192.593-327.246 seconds, and the efficiency is quite high. Table 3 compares the CPU time consumed by the OFFDIM and PMVS methods using images from 4 strips with strip being the unit, respectively. It can be seen from Table 4.3 that the matching operation speed of OFFDIM is approximately 6.2 times that of PMVS. It can also be found through the visual inspection of the 3D point cloud effect sketches of PMVS and the OFFDIM in the same region that the 3D point cloud density generated by the OFFDIM is far higher than that of PMVS; if the efficiency is calculated by the CPU consumed via single point, then the OFFDIM would have higher matching efficiency than PMVS.

Table 4.3 Consumed CPU time for dense image matching in single strip

Strip No.	Number of stereo model	CUP time (h : min : sec)		
		OFFDIM	PMVS	SURE
148-138	10	0:45:06	4:38:27	23:27
122-132	10	0:46:57	4:27:11	22:11
114-104	10	0:47:17	4:52:30	23:30
92-102	10	0:41:47	4:29:30	20:30

4.6.5 The effect of seed points on dense image matching based on optical flow field

As OFFDIM is a dense image matching method which extracts 3D point clouds in an image overlapping region based on seed points, a quantitative analysis of the correlation between the number of seed points and the matching effect was carried out in this paper. All seed points used in this paper came from pass points, and their distribution in the overlapping region was taken into consideration in the automatic image measurement; their distributions were uniform on the whole, and measurement accuracy was better than ±0.15 pixel. It can be clearly seen from Figure 8 that, as the number of seed points increased, the accuracy of the dense matching point clouds in image became higher and higher, but when the number of seed points reached 1,000 orders of magnitude, the matching accuracy tended to stablize. Figure 4.12 reveals that as the number of seed points increased, the dense matching success rate also improved, but when the number of seed points reached 1,000 orders of magnitude, the matching success rate almost did not change. As the estimation process that converts sparse optical flow fields to dense optical flow fields is a fitting and interpolation process, when the number of seed points increased, the estimation time of the dense optical flow fields increased correspondingly. It is not hard to obtain from the comprehensive analyses shown in Figure 4.11 and Figure 4.12 that a uniform distribution of 1,000 seed points in the overlapping region would be very good for the OFFDIM.

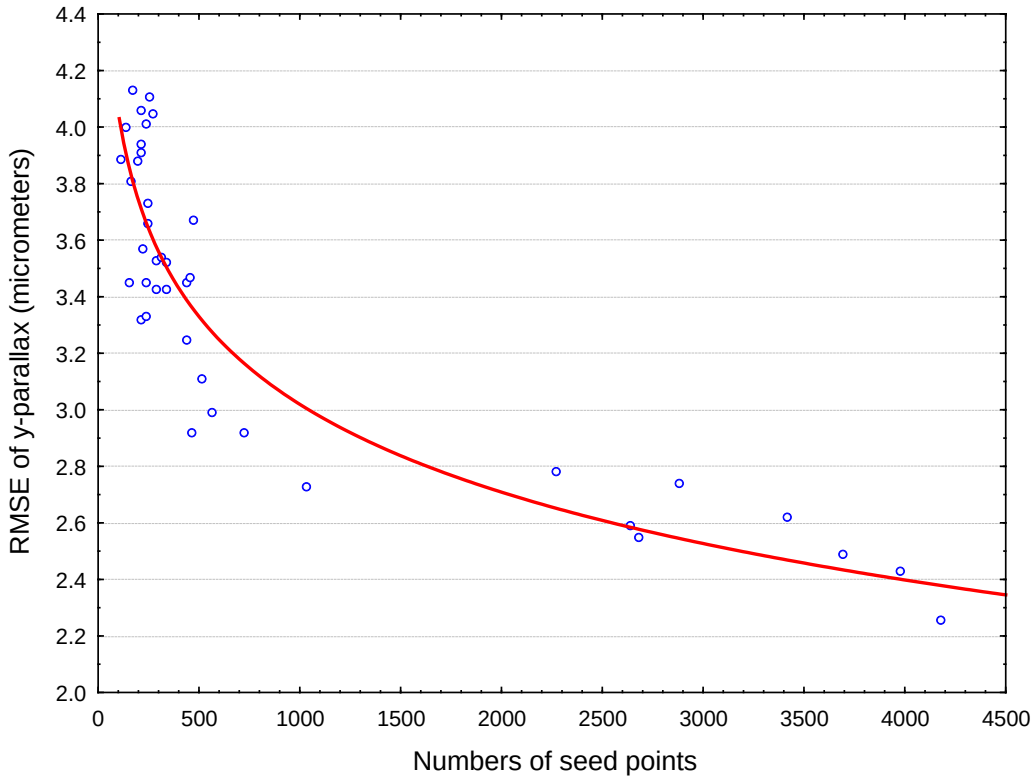


Figure 4.11 Dense matching accuracy curve with the number of seed points

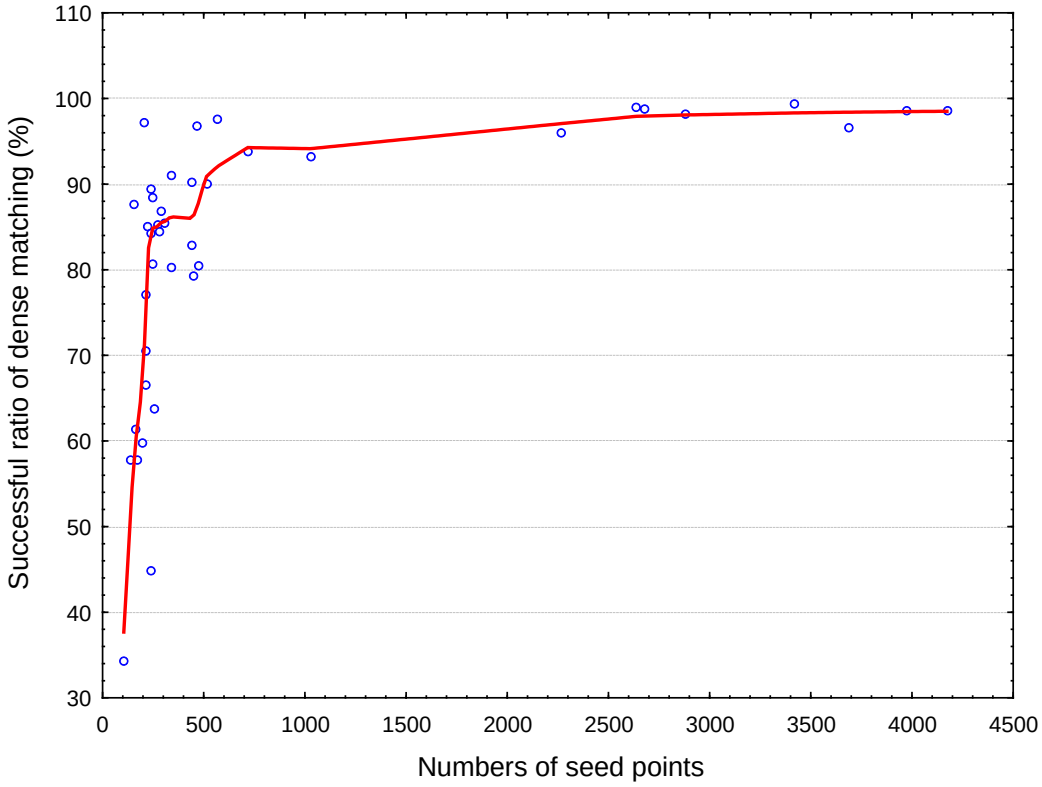


Figure 4.12 Dense matching reliability curve with the number of seed points

4.7 Summary of this chapter

An algorithm that uses accurate seed points to generate a dense optical flow field algorithm within a stereo image pair overlapping region and a dual-constraints refinement method were improved in this chapter. The proposed OFFDIM method can obtain a pixel-wised dense image matching results. The experimental results indicated that the matching success rate of OFFDIM is higher than 97%, and the matching accuracy reached the sub-pixel level; thereby, the automatically generated DSM elevation accuracy can be better than ±3 GSD. In addition, a comparison experiments with PMVS demonstrated that the matching efficiency of OFFDIM is improved by more than 5 times relative to that of PMVS, it has a higher matching success rate in some regions, like those containing houses and texture-poor regions, in aerial UAV images, and completeness of dense point clouds expressing ground features is better. However, the effect of the OFFDIM is closely related to the quantity, distribution and precision of seed points, and the dense image matching effect would be better in seed point regions with sufficient quantity and uniform distributed seed points.

At present, the proposed algorithm was only conducted on CPU, so the efficiency of the algorithm is expected to be further improved. How to adopt fragmental image processing technology, multi-threading computation or a GPU parallel algorithm will be the goal of future research and how to reduce the dependence of the algorithm on seed points and construct a practical seed point estimation model is worthy of concern.

Chapter 5
Deep Learning based 3D point cloud annotation

In this chapter, I will introduce the basic concept and algorithms used in our point cloud annotation approach. I will detailed describe our approach by 4 sections, first one is preprocessing; second one is modified deep neural network; then the experiment and analysis; the last one is the summary.

5.1 Proposed Res-FCN

5.1.1 U-Net

The U-Net network structure is proposed in the 2015 ISBI competition. It is an improve network from fully convolutional network. The structure forms a U-shaped structure through a shrinking network and an expansion network to extract features from the image, which won the championship of the 2015 ISBI competition.

The U-Net network consists of 23 convolutional layers. The structure is shown in Figure 5.1. The shrinking network is mainly responsible for the down sampling work, extracting high-dimensional feature information, and each down sampling contains two 3×3 convolution operations. A 2×2 pooling operation, with a rectified linear unit (ReLU) as the activation function, each time down sampling, the image size becomes 1/2 of the original size, and the number of features becomes twice the original. The expansion network is primarily responsible for the up sampling work, and each up sampling contains two 3×3 convolution operations by modifying the linear unit as the activation function. Each time up sampling, the image size becomes 2 times the original size, and the number of features becomes 1/2 of the original. In the up sampling operation, each output feature is merged with the features of the phased contraction network to complement the missing boundary information. Finally,

a 1×1 convolution operation is added to map the previously acquired features to the associated classification.

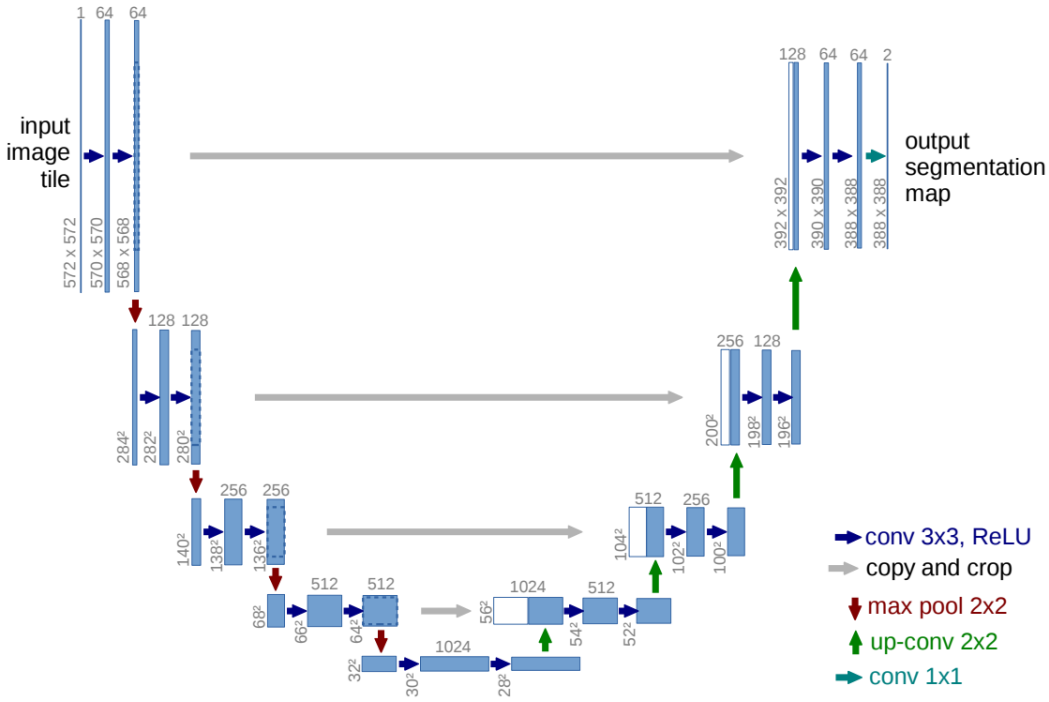


Figure 5.1 The structure of U-Net. White part is in the process of up sampling (*Ronneberger et al, 2015*).

Compared with other networks, U-Net has the advantages of simple structure, short training time, and few training parameters. However, compared with VGG, SegNet and other networks, the depth of U-Net is slightly insufficient.

5.1.2 Residual Network

In convolutional neural networks, the deeper the network hierarchy, the more errors are generated during training and the longer the training time. The emergence of the residual network has solved this problem to some extent. The residual network is the method proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun in the ILSVRC competition in 2015, and won the 2015 ILSVRC championship.

The residual network proposes a method of fitting the residual identity mapping, that

is, the convolution result is not directly used as an output, but the residual identity mapping is used to calculate, which is called a "shortcut". We assume that a hidden layer is F(x), which satisfies the mapping relationship F(x) = H(x) −x. If multiple nonlinear layers are combined, we can think of them as a complex network. We can also assume that implicit the residual of the layer is approximated by a complex function, for example H(x) = F(x) + x. The structure of the residual network is shown in Figure 5.2. As can be seen from the figure, the residual network reduces the training parameters by extracting the features of the convolutional layer cascaded output and input.

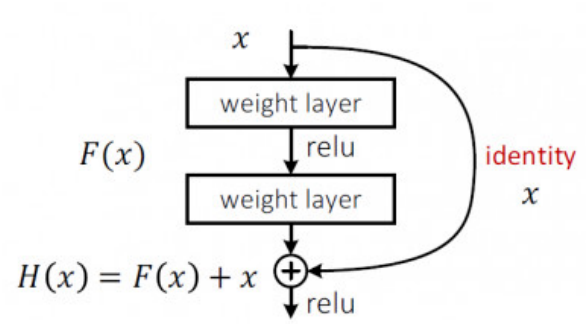


Figure 5.2 The structure of residual network

5.1.3 Residual fully convolutional network (Res-FCN)

The proposed Res-FCN is divided into two parts: the shrinking network and the expanding network. The shrinking network is similar to the shrinking network in U-Net. The difference is that the output of each layer is normalized first, followed by the activation function. Each up sampling step contains two 3×3 convolutional layers, a 1×1 "shortcut" and a 2×2 pooling layer. In every down sampling step, the picture size becomes 1/2 of the original, and the number of features acquired is doubled. The expansion network is similar to the expansion network in U-Net. Each up sampling contains two 3×3 convolutional layers, a 1×1 "shortcut" that needs to merge the results of the shrinking network before each up sampling. Similar to shrinking networks, each layer of output in an extended network requires advanced normalization and subsequent activation through an activation function. Finally, a 1×1

convolutional network is added to determine the result of the feature map. The network structure is shown in Figure 5.3.

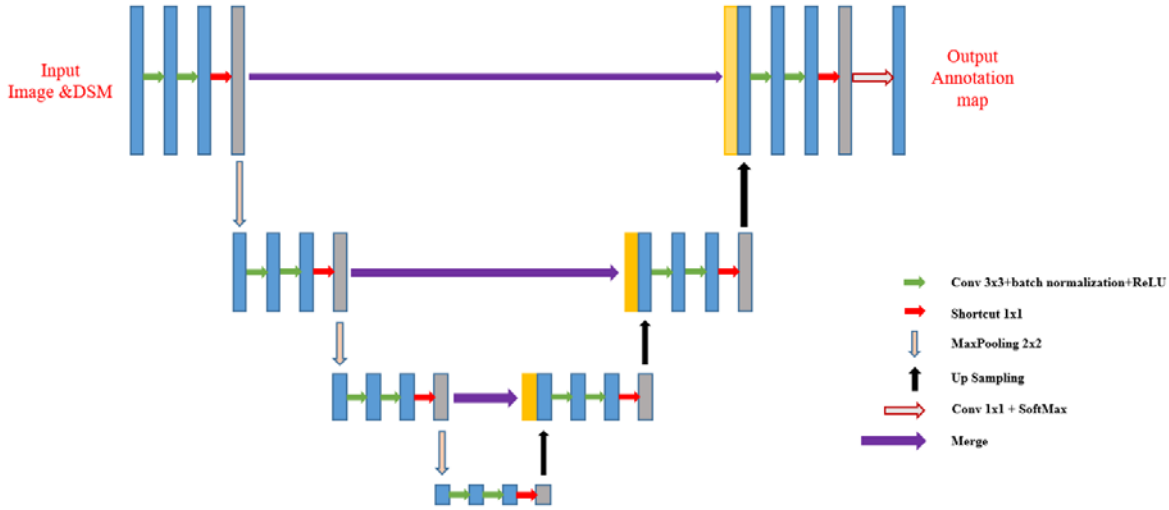


Figure 5.3 The structure of proposed Res-FCN

After joining the residual network, the proposed network has deeper levels and more training parameters than U-Net. To a certain extent, it makes up for the problem that the U-Net is not deep enough. At the same time, due to the characteristics of the residual network, it solves the problem of degraded performance of deep convolutional neural networks.

5.1.4 Conditional random fields

Conditional random fields (CRFs) are a discriminative probability undirected by Lafferty et al. based on Hidden Markov model (HMM) and Maximum entropy model (MEM). The graph model was originally used for labeling and segmentation of one-dimensional data. In 2003, Kumar extended the CRF model to a 2-dimensional structure and took the lead in applying it to image classification problems. Unlike the Markov random felds (MRF) model, which models the likelihood function, the CRF directly models the posterior distribution, so there is no need to satisfy the assumption of conditional independence between the observed data, so that any observation data can be represented. Therefore, the CRF model has been widely used and has been

introduced into many fields such as pattern recognition, image segmentation, target detection and remote sensing image classification. At the same time, more and more CRF improvement models have emerged, such as multi-scale CRF model, layered CRF model and hidden condition random field (HCRF) model.

In order to better understand the conditional random field, this section will first introduce the basic theory related to the probability map model, including the directed graph model, the undirected graph model, and the basic concepts of the generated model and the discriminant model, followed by the conditional random field， the main ideas and mathematical representations, and finally the application of conditional random fields in remote sensing image classification.

If an undirected probability graph has a Markov property, the undirected probability graph is called a Markov random field. Furthermore, if each node (random variable) of a Markov random field has an observation sample, and given the set of observation samples, the conditional distribution of the Markov random field is obtained, then the Markov random field has become a conditional random field.

The rigorous definition of the conditional random field is as follows:

Let $\mathrm{G}(V,E)$ denote an undirected graph, and the elements in $\mathrm{Y}=(Y_v)_{v\in V}$ correspond one-to-one with the vertices in the undirected graph G. When under the condition, the conditional probability distribution of the random variable Y_v obeys the Markov property of the graph: $\mathrm{p}(Y_v|X,Y_w,w\neq v)=\mathrm{p}(Y_v|X,Y_w,w\sim v)$, where $w\sim v$ means (w,v), that is, the edge of the undirected graph G. At this time we call (X,Y) a conditional random port. In the above definition, X and Y represent random variables with a joint distribution, where X represents an observation sequence that needs to be labeled or classified, and Y represents a marker sequence that marks or classifies, all $Y_i\in Y$ is assumed to be a finite number.

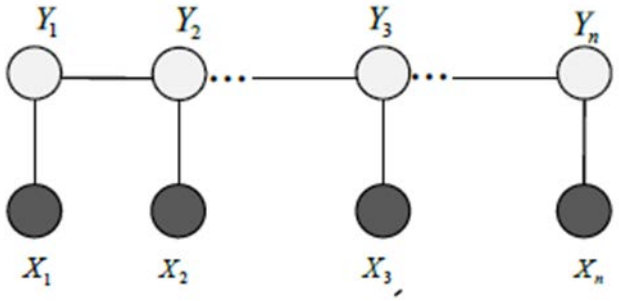


Figure 5.4 Linear chain conditional random field

According to the Hammersley-Clifford theorem, the posterior probability of the marker field x satisfies the Gibbs distribution:

$$p(x|y,\theta) = \frac{1}{Z(y,\theta)} \exp\{\sum_{c\epsilon C} \phi_C(x_C, y, \theta)\} \tag{5.1}$$

Where $Z(y,\theta) = \sum_x \exp\{\sum_{c\epsilon C} \phi_C(x_C, y, \theta)\}$ is a normalization function, ϕ_C represents a potential function, which is defined on the group C, and θ represents the parameters to be evaluated. In the application of the conditional random field model, defining a suitable potential function is a critical step.

In the conditional random field model, the potential function is usually represented by a linear combination of multiple features. A conditional random field model containing a one-dimensional potential function and a binary potential function can be expressed as follows:

$$\mathrm{p}(x|y,\theta) = \frac{1}{Z(y,\theta)} \exp\left\{\sum_{i\epsilon S}\sum_k \theta_{1k} f_k(x_i|y) + \sum_{i\epsilon S}\sum_{j\epsilon N}\sum_d \theta_{2d} g_d\left(x_i, x_j|y\right)\right\} \tag{5.2}$$

In the above formula, $f_k(x_i, y)$represents the k-th component of the D-dimensional unary eigenvector $f(x_i, y)$, and $g_d(x_i, x, y)$represents the M-dimensional binary eigenvector $g\left(x_i, x_j, y\right)$The d-th component of $\theta_1 = \{\theta_{1k}, k = 1,2,\dots,D\}$ is the parameter set of the one-potential function, $\theta_2 = \{\theta_{2d}, d = 1,2,\dots,M\}$ is the binary potential function The set of parameters, $Z(y,\theta)$ represents the normalization function.

1. Unary potential

The unary potential function $f_i(x_i|y)$ is a local constraint for a single node whose class is only related to the features of the current pixel and independent of the features of the surrounding nodes. The general definition is as follows:

(1) Logistic regression

Logistic Regression (LR) classifiers are suitable for two-category problems, and their expressions can be written as:

$$f_i(x_i|y,\omega) = \log\left(\frac{1}{1+\exp(-x_i\omega^T y_i)}\right) = \log\left(\sigma(x_i\omega^T y_i)\right) \tag{5.3}$$

Where $\sigma(x) = 1/(1+e^{-x})$ represents a Logistic function, and y_i represents a pixel value or a multi-dimensional feature vector, which usually represents the underlying features of the image such as color, grayscale, and texture, and n represents Vector dimension, $\omega = [\omega_1, \omega_2, \cdots, \omega_n]^T$represents the model parameter vector. For multi-classification problems, logistic regression can be extended to Multinomial Logistic Regression (MLR):

$$f_i(x_i|y,\omega) = \sum_{k=1}^{K} \delta(x_i = k) \log p(x_i = k|y,\omega) \tag{5.4}$$

Where $p(x_i = k|y,\omega)$ is a multiple logistic regression model whose expression is:

$$p(x_i = k|y,\omega) = \begin{cases} \frac{\exp(\omega_k^T y_i)}{1+\sum_{t=1}^{K-1}\exp(\omega_t^T y_i)}, & if \ \ k < K \\ \frac{1}{1+\sum_{t=1}^{K-1}\exp(\omega_t^T y_i)}, & if \ \ k = K \end{cases} \tag{5.5}$$

Where ω_k is the k-th parameter vector and n is the vector dimension.

(2) Support vector machines

For a two-category CRF model, the expression is:

$$f_i(x_i|y,\omega) = \log O(x_i|y,\omega) \tag{5.6}$$

In the formula,

$$O(x_i = 1|y,\omega) = \frac{1}{1+exp(a\times\Gamma(y_i)+b)} \tag{5.7}$$

Where, $\Gamma(\cdot)$ is the decision function of the support vector machine, and a and b are constants. For multi-class CRF models, multi-class support vector machines are used accordingly.

In addition to the above two forms, the one-dimensional potential function of the CRF model can also be established using various functions such as a kernel function, a Gaussian function, a regression tree, and a neural network.

2. Binary potential

The binary potential function is the core component of CRF, which enables CRF to describe the correlation and interaction between image pixels, and can reasonably model the spatial relationship between pixels, breaking the naive Bayesian classifier between pixels. Independent assumptions. In the MRF model, the pixel class is judged only by the feature vector of the pixel and the mark of its neighboring pixel. In the CRF model, the class judgment of the pixel also needs to consider the observation value of its neighboring pixel. If the pixels of the neighborhood have high image feature similarity, then the binary potential function $f_{ij}(x_i, x_j|y)$ is more likely to assign the same class mark to them if the pixels of the neighborhood have image features. The similarity is not high, then $f_{ij}(x_i, x_j|y)$ is more inclined to divide them into different categories, which will further improve the accuracy of the classification.

In the MRF model, the potential function is generally defined by the Potts model, but the Potts model that does not contain observation data cannot be directly used to define the potential function of the CRF model, so the potential function in the CRF is defined by the generalized Potts model.

For the two-category problem, the generalized Potts model has the following form:

$$f_{ij}(x_i, x_j|y) = x_i x_j v^T g_{ij}(y) \tag{5.8}$$

Where $g_{ij}(y)$ represents the eigenvector of the observation data corresponding to the coordinates (i, j), and $v = [v_1, v_2, \cdots, v_n]^T$ is the model parameter vector.

For multi-classification problems, the form is as follows:

$$f_{ij}(x_i, x_j|y) = \sum_{k,l\in\{l.\cdots,k\}} v_{kl}^T g_{ij}(y)\delta(x_i = k)\delta(x_j = l) \tag{5.9}$$

Where v_{kl} represents the $n \times K^2$ dimensional parameter vector, $g_{ij}(y)$ represents the binary eigenvector defined in the entire image, which describes the arbitrary correlation between the observed data, which is very important for the segmentation of the discontinuous region.

5.1.5 Conditional random fields in objects classification

This paper will use the conditional random field model as a post-processing method in the test phase to optimize the classification results. As shown in Figure 5.5, each node in the undirected graph corresponds to the category label and observation value of one pixel in the image, and the edge connection between the node and the node constitutes a conditional random field.

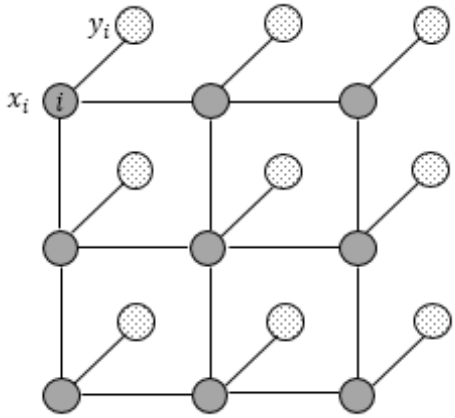


Figure 5.5 Conditional random field model

As an undirected graph model consistent with the Gibbs distribution, the Gibbs energy function of the conditional random field is as follows:

$$E(x) = \sum_i \psi_i (x_i) + \sum_{ij} \psi_{ij}(x_i, x_j) \tag{5.10}$$

Where x_i、x_j represent the category label values of the pixels i, j, respectively, and $\psi_i(x_i) = -\log P(x_i)$ represents a unitary potential number. Where $P(x_i)$ is the probability value of the output of the Softmax classifier The one-potential function indicates that the class is marked according to the characteristics of the node (pixel) itself in the model, and is only related to the local feature of the node itself, $\psi_{ij}(x_i, x_j)$ The binary potential function, which represents the probability that the node is marked as a class according to the similarity relationship between the neighboring nodes, and is related to the relationship between the nodes (pixels), that is, by describing the "coordinates between pixels" The "color" constraint relationship increases the probability that a pixel with a high similarity is classified into the same

category, and its expression is:

$$\psi_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \cdot k^m(f_i, f_j) \tag{5.11}$$

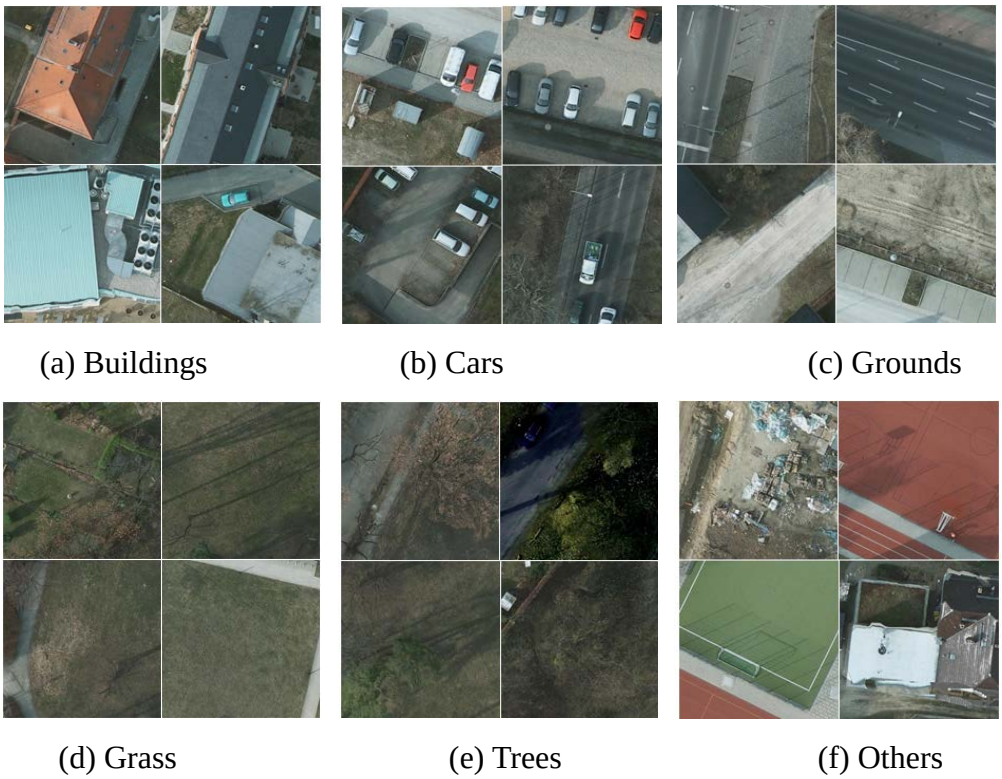
Where, $\mu(x_i, x_j)$ represents the indication function. When $x_i \neq x_j$, $\mu(x_i, x_j)$ has a value of 1, otherwise the value is 0, w^m represents a weight parameter, and $k^m(\cdot)$ represents a characteristic f. Gaussian kernel function. The kernel function used in this paper is as follows:

$$k(f_i, f_j) = w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \tag{5.12}$$

Where p represents the pixel coordinate value of the input image, I represents the pixel color value, σ_α、σ_β and σ_γ represent the scale of the Gaussian kernel; w_1 and w_2 represent the weight parameters of the two kernel functions, respectively.

5.2 Preprocessing

The experimental data selected in this paper is from the ISPRS WG II/4 public data set. The experimental area is taken in Potsdam, the capital of Brandenburg, Germany, as shown in Figure 5.6. The area is a densely populated urban area, mainly containing 6 kinds of land objects, such as houses, grounds, low vegetation, trees, vehicles and sundries. As can be seen from the figure, there are various types of houses in the experimental area. The houses with gray roofs are very similar to the road features, and it is difficult for human eyes to distinguish. The trees include green trees and dead trees. The dry tree canopy is intertwined with the shadows, which makes the classification more difficult. The vehicles include various types of cars and a small number of trucks, which are very similar to the shape of the temporary shacks. The sundries include the course. A typical object is such as garbage dumps, mounds, non-housing buildings and temporary shacks.



(a) Buildings (b) Cars (c) Grounds

(d) Grass (e) Trees (f) Others

Figure 5.6 The experimental data set

Experimental data includes images, digital surface models, and corresponding manual annotations. The ground resolution of the experimental data is 0.05 m, the image size is 6000×6000 pixels, the image is composed of three bands of RGB, the spectral resolution is 8 bit, and the DSM is a 32-bit single-band grayscale image. In this paper, 24 RGB images with category labels and corresponding DSMs were selected for experimentation, 22 of which were used as training and validation sets for the training phase, and 2 were used as test sets for the test phase. When performing model training, it is generally assumed that the data satisfies the independent and identical distribution, that is, the currently generated data can simulate the future data, so the generated data can be used to train the model, and then the trained model is used to fit the future data. However, in practical applications, the distribution of data usually changes, resulting in the assumption that independent and identical distribution does not hold. Moreover, the amount of data generated may not be sufficient to estimate the distribution of the entire data set, at which point the model is likely to have an over fitting. Over-fitting means that the output obtained by the training sample is basically the same as the

target output, but the output obtained by the test sample has a large difference from the target output, that is, the model generalization ability is too poor. In general, over fitting means that the model learns the training samples too thoroughly, so that the characteristics of the noise data are learned, which makes it impossible to correctly fit and classify the test samples expanding the training sample is a strategy to avoid over fitting often used in the data preparation stage. Since the sample size of the common data set is limited, it is necessary to use image processing to expand the training sample, that is, to translate, rotate, and change the brightness of the training data. A series of geometric transformations and radiation transformations are used for image expansion. The specific method of this paper is as follows: First, 22 images and corresponding DSMs are cropped into 4,950 image blocks of 400×400 pixels, of which 4000 are used as training sets. The rest is used as the validation set, and then the flip and transpose operations are simultaneously performed on the image and DSM of the training set, and random disturbances are added to the image contrast, saturation, brightness and hue, and then superimposed with the corresponding DSM to obtain the final training set data. The validation set and the training set data are obtained by directly superimposing the image and the DSM. So we use affine transformation and rotation to adding 1 training sample to 8 samples through data expansion. When remote sensing images are acquired, the uncertainty of factors such as weather, platform and time will cause the diversity of image brightness, color and sharpness. Through a series of geometric transformations and radiation transformations, the diversity of images can be simulated, and a large number of realities can be obtained. Distributed training samples, which greatly reduce the over-fitting problem and improve the robustness of the model.

5.3 Experiments and analysis

Figure 5.7(a) captures a typical area of 2000 × 2000 pixels in the test set image as an example, and Figure 5.7(b) shows the correct category of its manual labeling, Figure

5.7(c) is the result of classification by AlexNet. It can be seen from the figure that there is serious noise in the whole classification map, and the misidentification of houses, ground, vehicles and other places is obvious, but the overall classification result is good. Figure 5.7(d) shows the result by VGG16 method, the classification effect of the ground and the house is better than the AlexNet, but the phenomenon that the small area of the house edge is misclassified, and the noise is around the small object such as vehicles, sundries. Figure 5.7(f) shows the result by using the fully convolutional network (FCN) method for classification, it can be seen that the phenomenon of small points on the edge of the house is basically eliminated, and small objects such as vehicles are also very well classified. However, when only the image is used as the training data, the classification result of the FCN method is that the whole house is divided into the ground by the whole building, which greatly affects the overall classification effect, which is due to the difference between the house and an important feature of the ground. It is the elevation difference, and the elevation difference is difficult to reflect in the RGB image. When the image is used alone as the training data, there are some cases in which the FCN method is divided into the whole building and the large number of holes appear, which greatly affects the overall classification effect. This is because there are a large number of houses in the experimental data that are very similar to the ground features, and the elevation difference is an important feature that distinguishes the house from the ground. It is difficult to reflect in the images. For this reason, DSMs and images are used as inputs at the same time. The data is involved in training and classification, and the classification results are improved by the elevation information provided by the DSM.Figures 5.7(g) and (h) show the classification result of traditional ResNet and the proposed methods. It can be seen from the figure that the misclassified house and the ground are less. The noise has been significantly reduced, the edges of the objects are smooth and clear, and the classification result of the proposed method is very close to the real category.

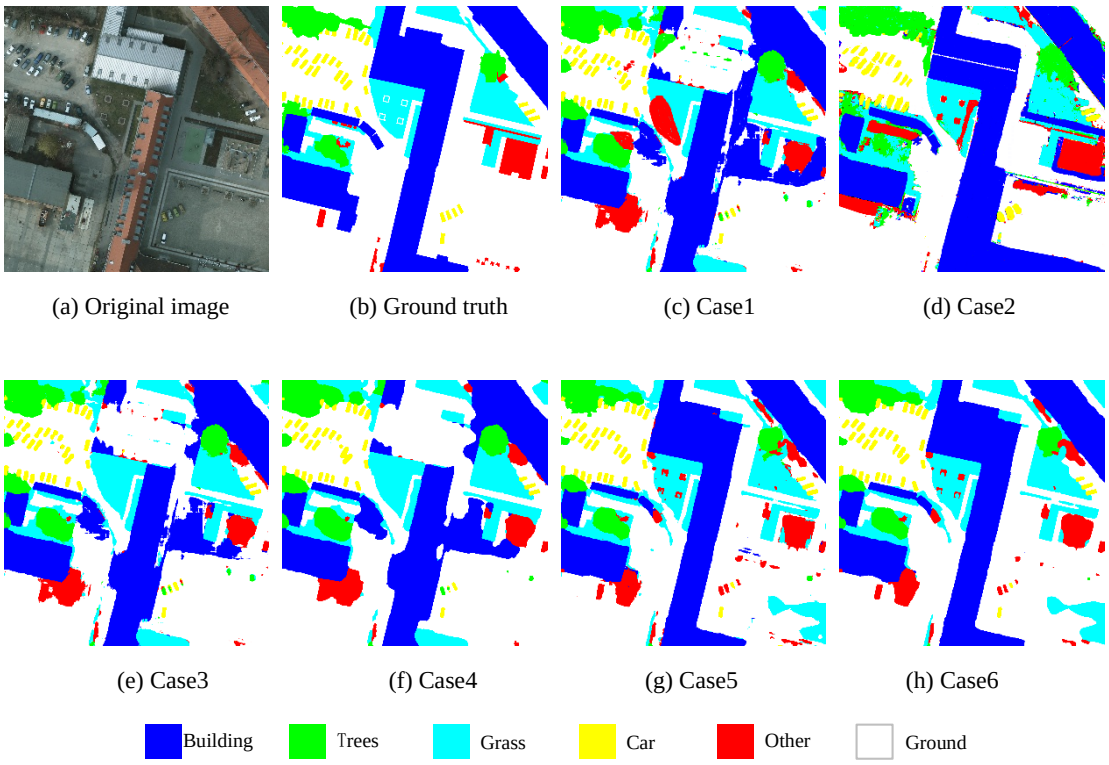


Figure 5.7 Classification results on ISPRS WG II/4 dataset

In this paper, the confusion matrix of each classification result is counted. And the classification results are quantitatively evaluated according to the statistical indicators based on the confusion matrix. The confusion matrix of the classification results is shown in Table 5.2. As shown in Table 5.2 to Table 5.4, the accuracy, recall rate and F1 measurement results of various types of features in the classification results of each method are respectively counted.

Table 5.1 Overview of different experimental schemes on ISPRS WG II/4 dataset

	Case1	Case2	Case3	Case4	Case5	Case6
Input	image+DSM	image+DSM	image+DSM	image+DSM	image+DSM	image+DSM
Methods	AlexNet	VGG16	FCN	FCN+CRF	ResNet	Res-FCN+CRF
Accuracy	80.56%	81.29%	84.76%	86.00%	87.25%	**88.80%**
Kappa	0.7446	0.7531	0.7707	0.7762	0.8242	**0.8303**

Table 5.2 Comparison of F1 score between different approaches

Methods	Building (%)	Ground (%)	Tree (%)	Grass (%)	Car (%)	Other (%)	Average (%)
AlexNet	87.50	82.27	70.54	68.53	87.21	30.35	83.08
VGG16	86.77	82.66	71.74	67.64	86.67	31.68	83.96
FCN	92.03	88.31	72.21	69.26	88.46	32.99	85.66
FCN+CRF	92.30	88.65	72.29	69.88	87.56	34.25	85.98
ResNet	96.27	90.94	75.57	72.83	**89.45**	41.79	88.96
Ours	**96.38**	**91.35**	**76.67**	**74.73**	88.80	**43.71**	**90.31**

Table 5.3 Comparison of classification recall between different approaches

Methods	Building (%)	Ground (%)	Tree (%)	Grass (%)	Car (%)	Other (%)	Average (%)
AlexNet	82.27	77.56	81.56	76.42	74.36	52.72	80.56
VGG16	86.83	78.16	79.72	78.65	82.46	55.36	81.29
FCN	89.17	82.86	**83.06**	78.49	87.68	58.70	84.76
FCN+CRF	89.32	84.60	83.98	77.45	84.72	58.61	86.00
ResNet	96.26	86.66	81.29	81.03	**89.57**	**66.51**	87.25
Ours	**96.38**	**88.35**	82.27	**82.38**	87.41	65.99	**88.80**

Tab.5.4 Comparison of classification accuracy between different approaches

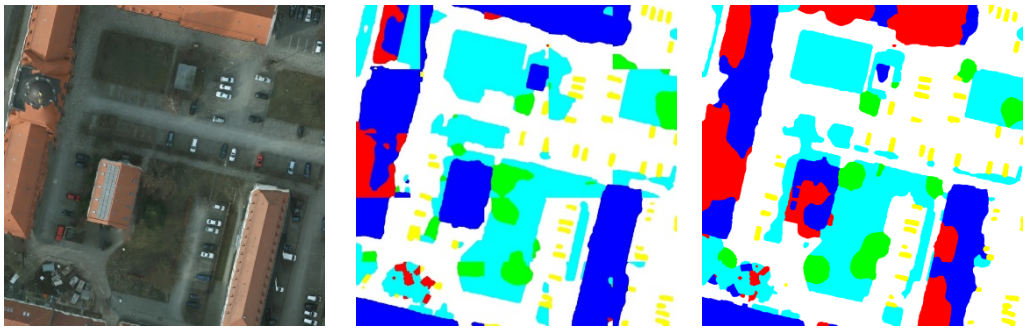
Methods	Building (%)	Ground (%)	Tree (%)	Grass (%)	Car (%)	Other (%)	Average (%)
AlexNet	93.83	88.65	63.37	60.67	88.29	25.12	83.22
VGG16	95.03	87.84	62.02	60.96	86.71	21.44	84.14
FCN	95.18	93.26	63.84	61.91	89.56	22.79	87.57
FCN+CRF	95.25	92.99	64.03	63.00	91.77	24.18	87.85
ResNet	96.37	**95.90**	70.45	66.34	89.42	30.26	90.27
Ours	**97.16**	95.61	**71.02**	**66.97**	**91.12**	**33.23**	**90.55**

The comparison of our proposed Res-FCN and traditional U-Net is shown as follows. We use the same dataset to training and validation the proposed Res-FCN and traditional U-Net. Both image and DSM are used as input and the CRF as the post

processing step. In Figure 5.8 and Table 5.5 we can obliviously see that our network get more accurate results.

Table 5.5 Comparison of classification accuracy between Res-FCN and U-Net

Methods	Building (%)	Ground (%)	Tree (%)	Grass (%)	Car (%)	Other (%)	Average (%)
U-Net	95.27	90.73	70.57	66.83	89.45	32.79	87.76
Ours	**97. 16**	**95.61**	**71.02**	**66.97**	**91.12**	**33.23**	**90.55**



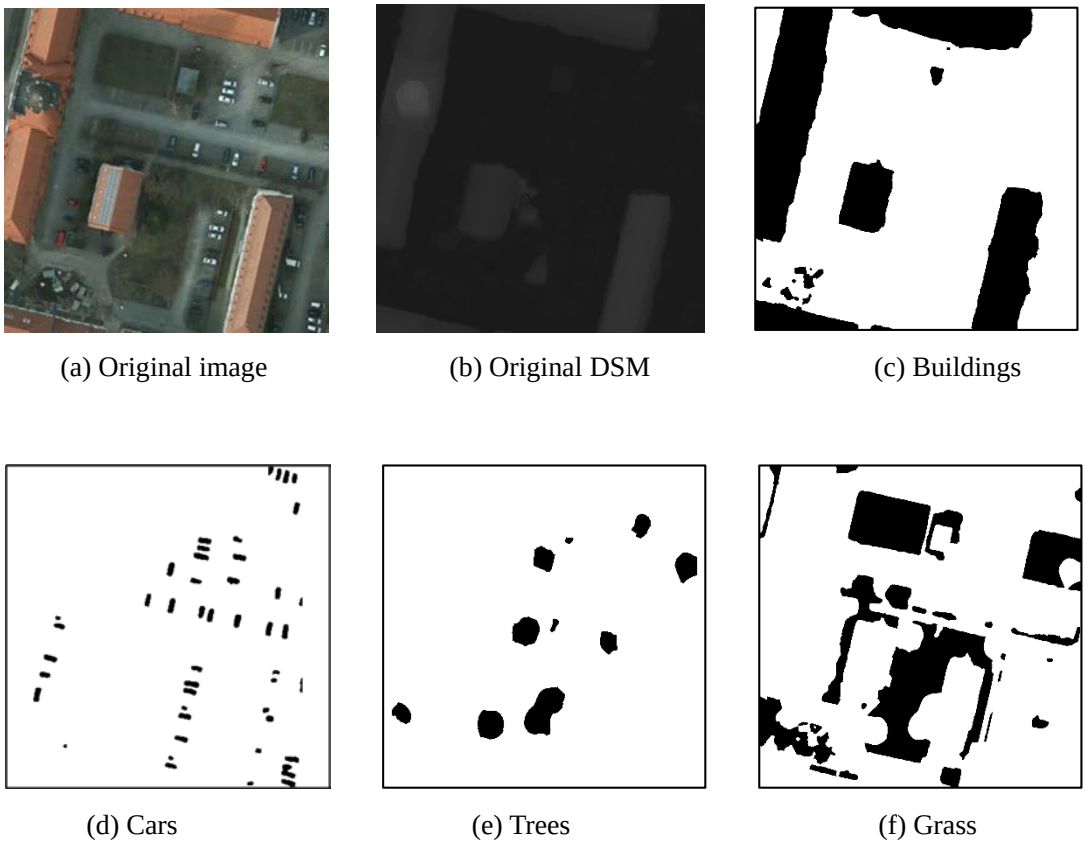
(a) Original image (b) Result by ResNet (c) Result by Res-FCN

Figure 5.8 The comparison of Res-FCN and U-Net

5.4 Digital object model gerenation

Digital object model (DOM) is generated by inserting the point cloud annotation results into the digital surface model (DSM), where every point in the DOM containing 2 attributes. One is the point's 3D coordinates and the other is the object annotation attribute. It has been formatted into a four dimensional Numpy array for data storage. Compared with 3D point clouds acquired by LiDAR system, DOM has its advantage in many specific applications. The LiDAR point clouds only have each point's 3D coordinates and the sensor's positioning information. This makes the specific object reconstruction on the raw point cloud data is impossible. The 3D point clouds should be segmented firstly. Considered that use only the point cloud to do the segmentation is harder than combined with corresponding images. If an engineer wants to build the 3D building model on this scene, DOM is easy-handling for them

to produce their work. Because we can easily select the interested object's point clouds in a scene for generating object directly, as shown in Figure 5.9.



(a) Original image (b) Original DSM (c) Buildings

(d) Cars (e) Trees (f) Grass

Figure 5.9 The slected DOM

We have stored the DOM as a 4 dimensional Numpy array. Its data structure is shown in Table 5.6. Each row data represents one point's attributes in DOM. In practical applications, if an engineer wants to reconstruct buildings in this scene, only search the 4th attribute which equals to "Building", then all the 3D point clouds in building area are automatically selected. In that case the production efficiency is improved. For photogrammetric production, the digital evaluation model (DEM) can be easily generated from DOM. Engineer only needs to search the object attribute which equals to "Ground" and the all ground point clouds are automatically extracted. The DEM can be easily generated by using interpolation of triangulated irregular network. In hazards loss assessment field, the collapsed buildings and damaged objects can be detected by setting the searching range of *X*, *Y* value and compared the *Z* value and

the object attribute value between the two DOMs before and after the hazards. For a building, it can be recognized as collapsed if the *Z* changes a lot.

Table 5.6 An example of data structure in DOM

X (°)	*Y* (°)	*Z* (m)	Object
116.41024449916938	39.91359571849836	102.7	Building
116.41024449917011	39.91359571849836	102.7	Building
116.52133315786237	39.82359567112345	95.61	Ground
116.52133315786241	39.82359567112322	96.71	Car

The proposed DOM is more useful for practical applications, such as urban planning, and evacuation planning, hazards loss assessment. Engineers can easily select the interest objects and use the 3D information to conduct their specific works.

5.5 Summary of this chapter

In this chapter we have described our method for points cloud annotation based on deep learning methods. The experimental results have demonstrated that our algorithm can improve the segmentation accuracy compared with some traditional deep learning methods and the non-modified ResNet. CRF is utilized as the post-processing step. It efficiently makes the results have smooth edges. However, as we only use one dataset to test our method. The robustness of our method cannot be tested. And the modified deep neural network has too much layers. The training is very time consuming. How to simplify the network is the future working directions.

Chapter 6
Conclusions and future works

6.1 Summary of the research works

In this doctoral dissertation, three new algorithms to solve 3 bottleneck problems of generating digital object model were proposed.

1. For poor textural remote sensing images, matching robustness is vulnerable to low contrast, repetitive patterns, occlusions and homogeneous textures. To address these problems, a novel feature matching algorithm is proposed in this paper which uses graph theory as a proxy: First, point features are respectively extracted in both source and target image to form feature set P and Q, which constructed graph G_P, G_Q subsequently. Then, an edge weighted strategy is adopted to build affinity tensor between G_P and G_Q. At last, the node correspondences between G_P and G_Q are acquired by using high order graph matching algorithm, and the feature matching process is finally completed by this proxy. In order to demonstrate the feasibility of our algorithm, several experiments are conducted, in which typical poor textural images that contain forest, desert, farmland and urban are used. And the comparison studies and experimental results proved that our algorithm has significantly improved on matching recall, number of correct matches and positional accuracy. Through this model, different feature extractors can be utilized to extract features and using the structure similarities to finding the correspondences. Compared with traditional and widely used tie point matching methods, our method can get more accurate and evenly distributed results. However, as the High ordered graph matching is very time consuming. The matching efficiency should be improved.
2. We proposed a reliable, efficient, robust image dense matching approach. The

result accuracy achieved sub-pixel level. The proposed method utilized optical flow fields as the instruction to reduce the redundant searching in fine matching step. And a multi-constraint fine matching is utilized to improve the 3D point cloud accuracy. The experimental results demonstrated that our method is 6 times faster than PMVS, and has better completeness than the commercial software named SURE. Image dense matching is a key technology in the fields of photogrammetry and computer vision that urgent requires solutions, and it is expected that it can be transformed from a research hotspot into practical utilization to accelerate automated progress of extracting 3D geospatial information from images for purposes including 3D object reconstruction, DEM extraction and oblique photogrammetry, and so on. An algorithm that uses accurate seed points to generate a dense optical flow field algorithm within the overlapping region of a stereo image pairs and a dual-constraints refinement method were improved in this paper. The proposed OFFDIM method can obtain a pixel-wised image dense matching results. The experimental results indicated that the matching success rate of OFFDIM is higher than 97%, and the matching accuracy reached the sub-pixel level; thereby, the automatically generated DSM accuracy can be better than ±2.5 GSD. In addition, a comparison experiments with PMVS demonstrate that the matching efficiency of OFFDIM is improved by more than 5 times relative to that of PMVS, it has a higher matching success rate in some regions, like those containing houses and texture-poor regions, in aerial UAV images, and completeness of dense point clouds expressing ground features is better. However, the effect of the OFFDIM is closely related to the quantity, distribution and precision of seed points, and the image dense matching effect would be better in seed point regions with sufficient quantity and uniform distributed seed points.

3. We combine the modified fully convolutional neural network and conditional random field to deal with point cloud annotation problem. The advantages of the two techniques are fully shown through the comparison experiments with

four deep learning based remote sensing image classification methods such as AlexNet, VGG16, fully convolutional network (FCN) and Residual Network (ResNet). At the same time, it is found that the spatial information can effectively improve the accuracy of image classification. The main research work and achievements of this study are as follows:

(1) In-depth research and summary of the development status of remote sensing image feature classification, we analyze the feasibility and advantages of deep learning methods.

(2) In order to avoid over-fitting results, on the basis of the existing training samples, the image and DSM of the training set are simultaneously flipped and transposed, and random disturbances are added to the contrast, saturation, brightness and hue of the image. Then, the final training set data is superimposed with the corresponding DSM, so that the training set scale is expanded by 8 times, which reduces the model over-fitting problem to some extent.

(3) The annotation results of Res-FCN, AlexNet, VGG16, FCN and maximum ResNet are compared, and the accuracy, recall rate, F1 measure and Kappa coefficient of each method are calculated. The evaluation index showed that the proposed residual fully convolutional neural network is not only superior to the other four deep learning classification methods in classification accuracy, and the other indicators are superior to the other four methods, which demonstrate its potential to practical applications.

(4) Aiming at the problem of rough annotation results of the fully convolutional neural network method, the conditional random field model is used to post-process the classification results. The results show that the edge of the classification result after the conditional random field model is processed becomes smoother and the noise is eliminated.

6.2 Future work

Nowadays UAV photogrammetric technologies have developed rapidly. UAV aerial images are not exactly the same as traditional aerial photogrammetric images. How to utilize our approach on UAV low attitude photogrammetry is worth researching on. On other aspects, it is necessary for me to improve our proposed methods to meet the demand of real-time applications in the next. At this moment, the generated DOM is stored as a Numpy array. For large-scale practical applications, how to use database to store and search DOM is a great challenge. Additionally, the annotated results are quite depended on the training dataset, how to use transfer learning methods to generate the annotation model is also necessary to study in the future.

Reference

Ackermann F, 1984. Digital image correlation: performance and potential application in photogrammetry. *The Photogrammetric Record*, 11(64): 429-439

Advi G, 1985. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384-401

Baarda W, 1968. A testing procedure for use in geodetic networks. Delft, Kanaalweg 4, Rijkscommissie voor Geodesie, 1

Bunge H J, Morris P R. 1982. Texture analysis in materials science: mathematical methods.

Barron J L, Fleet D J, et al, 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1): 43-77

Barnea D I, Silverman H F, 1972. A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, 100(2): 179-186

Beauchemin S S, Barron J L, 1995. The computation of optical flow. *ACM Computing Surveys (CSUR)*, 27(3):433-466

Baker S, Kanade T, 1999. *Super-Resolution Optical Flow*. Robotics Institute, Carnegie Mellon University, Pittsburgh.

Belongie S, Malik J, Puzicha J, 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 509-522

Bay H, Ferraris V, Van G L, 2005. Wide-baseline stereo matching with line segments. In: *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, pp. 329-336

Bay H, Tuytelaars T, Gool L V, 2006. SURF: speeded up robust features. *Computer Vision and Image Understanding*, 110(3): 404-417

Breiman L, 2017. *Classification and regression trees*. Routledge

Cho M, Sun J, Duchenne O, et al., 2014. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, pp. 2083-2090

Chung F R K, 1997. *Spectral Graph Theory*. American Mathematical Society

Cramer M, Stallmann D, 2001. On the use of GPS/inertial exterior orientation parameters in airborne photogrammetry. In: *Proceedings of OEEPE Work shop-Integrated Sensor Orientation*. Hannover.

Chen L C, Papandreou G, Kokkinos I, et al., 2014. Semantic image segmentation with deep convolutional nets and fully connected CRFS. *arXiv preprint arXiv:1412.7062*.

David M M, Sunil A, 2010. ANN: A Library for Approximate Nearest Neighbor Searching. http://www.cs.umd.edu/~mount/ANN/

Delaunay B, 1934. Sur la sphère vide. A la mémoire de Georges Voronoï. Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, 6, 793-800

Duchenne O, Bach F, Kweon I S, et al., 2011. A tensor-based algorithm for high-order graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12): 2383-2395

Edmonds, J., 1965. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3): 449-467

Egozi A, Keller Y, Guterman H, 2013. A probabilistic approach to spectral graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 18-27

Elsd M, 1996. *Super-resolution Reconstruction of Image Sequences-adaptive Filtering Approach*. Israel: The Techoion-Israel Institute of Technology

Fischler M A, Bolles R C, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381-395

Galic S, Loncaric S, 2000. Spatio-temporal image segmentation using optical flow and clustering algorithm. In: *IEEE Proceedings of the First International Workshop on Image and Signal Processing and Analysis*, Pula, 63-68.

Goldberg A V, 1997. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22(1):1–29.

Gold S, Rangarajan A, 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4): 377-388

Gonzalez R C, Woods R E, 1992. *Digital Image Processing*, Addison-Wesley, New Jersey

Gruen A W, Baltsavias E P, 1988. Geometrically constrained multiphoto matching. *Photogrammetric Engineering & Remote Sensing*, 54(5): 633-641.

Grigorescu L, Oproescu G, Diaconescu I, et al, 2011. Fourier transform and its applications. In: *Proceedings of the 13th IASME/WSEAS international conference on Mathematical Methods and Computational Techniques*. World Scientific and Engineering Academy and Society (WSEAS), pp. 140-146.

Gonalves H, Gonalves J A, Corte-Real L, 2009. Measures for an objective evaluation of the geometric correction process quality. *IEEE Geoscience and Remote Sensing Letters*, 6(2): 292-296

Goncalves H, Corte-Real L, Goncalves J A, 2011. Automatic image registration through image segmentation and SIFT. *IEEE Transactions on Geoscience and Remote Sensing*, 49(7): 2589-2600.

Gonzalez R C, Woods R E, 2002. *Digital Image Processing*. New Jersey: Prentice Hall.

Girshick R, Donahue J, Darrell T, et al, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH: 580-587.

Gevaert C M, Persello C, Nex F, et al, 2018. A deep learning approach to DTM extraction from imagery using rule-based training labels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 142: 106-123.

Harris C, Stephens M, 1988. A combined corner and edge detector. In: *Alvey Vision Conference*, 15: 50.

Hartley R, Zisserman A, 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hossam Isack, Yuri Boykov, 2014. Energy Based Multi-model Fitting & Matching for 3D Reconstruction. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio.

Hugenholtz C H, Whitehead K, Brown O W, et al, 2013. Geomorphological mapping with a small unmanned aircraft system (sUAS): feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model. *Geomorphology*, 194: 16-24

Huo C, Pan C, Huo L, et al, 2012. Multilevel SIFT matching for large-size VHR image registration. *IEEE Geoscience and Remote Sensing Letters*, 9(2): 171-175

Hartmann W, Havlena M, Schindler K, 2016. Recent developments in large-scale tie-point matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115: 47-62

Hinton G E, Osindero S, Teh Y W, 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18 (7): 1527-1554

He K, Zhang X, Ren S, et al, 2016. Deep Residual Learning for Image Recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, pp.770-778

Hu X, Yuan Y, 2016. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote sensing*, 8(9): 730

Hackel T, Savinov N, Ladicky L, et al, 2017. Semantic3D. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*

Kotsiantis S B, Zaharakis I, Pintelas P, 2007. Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160*: 3-24

Kim H, Lee S, 2010. A novel line matching method based on intersection context. In: *Proceedings 2010 IEEE International Conference on Robotics and Automation*, Washington, pp. 1014-1021

Ke Y, Sukthankar R, 2004. PCA-SIFT: A more distinctive representation for local image descriptors. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* pp. II-II

Krystian M, Cordelia S, 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10): 1615-1630

Krizhevsky A, Sutskever I, Hinton G E, 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp: 1097-1105

Lewis J P, 1995. Fast normalized cross-correlation. *Vision Interface*, 10(1): 120-123

Leordeanu M, Hebert M, 2005. A spectral technique for correspondence problems using pairwise constraints. In: *Proceedings of IEEE International Conference on Computer Vision*, Beijing, pp.1482-1489

Livi L, Rizzi A, 2013. The graph matching problem. *Pattern Analysis and Applications*, 16(3): 253-283

Lyzinski V, Fishkind D E, Fiori M, et al, 2016. Graph matching: relax at your own risk. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1): 60-73

Lee S, Chwa K Y, Hahn J, et al, 1995. Image metamorphosis using snakes and free-form deformations. In: *SIGARPH'95*, 3, pp. 439-448

Lee S, Wolberg G, Chwa K Y, et al, 1996. Image metamorphosis with scattered feature constraints. *IEEE Transaction on Visualization and Computer Graphics*, 2(4):337-354.

Lee S, Wolberg G, Shin S Y, et al, 1997. Scattered data interpolation with multilevel B-splines. *IEEE Transaction on Visualization and Computer Graphics*, 3(3):229-244

LOWE D G, 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91-110

Liu W, Anguelov D, Erhan D, et al, 2016. Ssd: Single shot multibox detector. In: *European conference on computer vision*. Springer, Cham, pp. 21-37

Long J, Shelhamer E, Darrell T, 2015. Fully convolutional networks for semantic segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA

Lecun Y, Bengio Y, Hinton G, 2015. Deep learning. *Nature*, 521(7553): 436-444

Mair E, Hager G D, Burschka D, et al, 2010. Adaptive and generic corner detection based on the accelerated segment test. In: *Proceedings of European Conference on Computer Vision*, Crete, pp. 183-196

Negahdaripour S, 1998. Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9): 961 979

Revathi R, Hemalatha M, 2012. Detecting objects in video frames using optical flow techniques. In: *IEEE 2012 International Conference on Emerging Trends in Science, Engineering and Technology*, India, pp. 329-333

Ren S, He K, Girshick R, et al. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 91-99

Redmon J, Divvala S, Girshick R, et al, 2016. You only look once: unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 779-788

Ronneberger O, Fischer P, Brox T, 2015. U-Net: Convolutional networks for biomedical image

segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*, Springer, Cham, pp. 234-241

Schölkopf B, Platt J, Hofmann T, 2006. Balanced graph matching. In: *Proceedings Conference on Neural Information Processing Systems*, Vancouver, pp. 313-320

Sedaghat A, Mokhtarzade M, Ebadi H, 2011. Uniform robust scale-invariant feature matching for optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11): 4516-4527

Serradell E, mustafa Ö, Lepetit V, et al, 2001. Combining geometric and appearance priors for robust homography estimation. *Lecture Notes in Computer Science*, 6313:58-72

Stefano L D, Marchionni M, Mattoccia S, 2004. A fast area-based stereo matching algorithm. *Image and Vision Computing*, 22(12):983-1005

Sedaghat A, Ebadi H, 2015. Distinctive order based self-similarity descriptor for multi-sensor remote sensing image matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 10(108): 62-71

Silva G R L, Medeiros R R, et al., 2016. GPIC-GPU power iteration cluster. arXiv:1604.02700

Sun Y, Zhao L, Huang S, et al, 2015. Line matching based on planar Homography for stereo aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104: 1-17

Silver, David, et al, 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529: 484-489

Simonyan K, Zisserman A, 2014. Very deep convolutional networks for large-scale image recognition. *Computer Science*, arXiv:1409.1556

Szegedy C, Liu W, Jia Y, et al, 2015. Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9

Torresani L, Kolmogorov V, Rother C, 2008. Feature correspondence via graph matching: Models and global optimization. In: *Proceedings of European Conference on Computer Vision*, Marseille, pp. 596-609

Torresani L, Kolmogorov V, Rother C, 2013. A dual decomposition approach to feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):259-271

Wang Jingxue, Zhu Qing, Wang Weixi, 2013. A dense matching algorithm of multi-view image

based on the integrated multiple matching primitives. *Acta Geodaetica et Cartographica Sinica*, 42(5):691-698

Wang Z, Wu F, Hu Z, 2009. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42(5): 941-953

Wu B, Zhang Y, Zhu Q, 2012. Integrated point and edge matching on poor textural images constrained by self-adaptive triangulations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 68(1): 40-55

Yang Huachao, Zhang Shubi, Zhang Qiuzhao, 2010. Least squares matching methods for wide base-line stereo images based on SIFT feature. *Acta Geodaetica et Cartographica Sinica*, 39(3):187-194

Yuan Xiuxiao, 1991. Some investigations on the accuracy of high-precision photogrammetric densification. *Acta Geodaetica et Cartographica Sinica*, 20(3):232-238

Yuan Xiuxiao, 2008. A Novel Method of Systematic Error Compensation for a Position and Orientation System. *Progress in Natural Science*, 18(8):953-963.

Yuan Xiuxiao, Ming Yang, 2009. A Novel Method of Multi-image Matching Using Image and Space Synthesis Information. *Acta Geodaetica et Cartographica Sinica*, 38(3): 216-222

Ye Y, Shan J, 2014. A local descriptor based registration method for multispectral remote sensing images with non-linear intensity differences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90(3): 83-95

Zhang L, Armin G, 2006. Multi-image matching for DSM generation from IKONOS imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(3): 195-211

Zhu Q, Wu B, Xu Z X, 2006. Seed point selection method for triangle constrained image matching propagation. *IEEE Geoscience and Remote Sensing Letters*, 3(2), 207-211

Zhu Qing, Wu Bo, Zhao Jie, 2007. Propagation strategies for stereo image matching based on the dynamic triangle constraint. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(4): 295–308.

Zickler S, Efros A, 2007. Detection of multiple deformable objects using PCA-SIFT. In: *Proceedings AAAI Conference on Artificial Intelligence*, Vancouver, pp.1127-1132

Zheng S, Jayasumana S, Romera-Paredes B, et al, 2015. Conditional random fields as recurrent neural networks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 1529-1537