School of Electrical Engineering, Computing and Mathematical Sciences

A Fast Matching Algorithm for Images with Large Scale Disparity and its Application on UAV Autonomous Navigation

Shichu CHEN

This thesis is presented for the Degree of
Master of Philosophy
of
Curtin University

December 2020

Declaration

I hereby declare that, to the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

I have obtained permission from the copyright owners to use any third-party copyright material reproduced in the thesis, or to use any of my own published work in which the copyright is held by another party.

Signature: ________________

Date: ________________

Acknowledgements

I would like to thank those people who helped me in different ways during the period of the project. Without their help, the thesis could not have been possible. I want to express my sincere appreciation to my supervisor, Prof. Wanquan Liu for the constant guidance and encouragement. Thanks for offering me such an opportunity to be a research student. It is a truly indelible and valuable experience to work with you.

I am also grateful to my schoolmates, Huizhu Pan, Qilin Li, Antoni Liang, Yan Ren, Lu Tan, Ruhua Wang, Brad, and Nadith for many discussions and insightful suggestions.

Publications

The following works contribute to the author's Master Degree thesis, ordered chronologically:

- **Chen, S.**, Wang, Z., and Ren, Y. (2020). A fast matching algorithm for the images with large scale disparity. *Mathematical Foundations of Computing*, 3(3), 141-155.

- **Chen, S.**, Li, Q., and Ren, Y. (2021). A Novel UAV Autonomous Navigation Approach based on Fast Matching Algorithm for Images with Large Scale Disparity. *International Conference on Service Robotics Technologies.* (accepted)

Attribution Statement

Chapters 2, 3 and 4 of this thesis are based on works that have been published or accepted with joint-authorship. We hereby make an authorship attribution statement to clarify the contribution of the author Shichu Chen.

Chapter 2 and part of Chapter 4 are based on the publication:

- **Chen, S.**, Wang, Z., and Ren, Y. (2020). A fast matching algorithm for the images with large scale disparity. *Mathematical Foundations of Computing*, 3(3), 141-155.

  The design and implementation of the model and the numerical algorithm, experimental results, and the writing of the publication can be attributed to Shichu Chen.

Chapter 3 and part of Chapter 4 are based on the publication:

- **Chen, S.**, Li, Q., and Ren, Y. (2021). A Novel UAV Autonomous Navigation Approach based on Fast Matching Algorithm for Images with Large Scale Disparity. *International Conference on Service Robotics Technologies.* (accepted)

  The design and implementation of the model and the numerical algorithm, experimental results, and the writing of the publication can be attributed to Shichu Chen.

Signatures of co-authors:

Abstract

Computer vision is a quite popular field involving many disciplines. The knowledge and technologies in computer vision provide a basis for other more complex and more advanced computer applications such as automation and intelligence.
A basic branch in the computer vision field is the image matching domain relating to varieties of algorithms, which is used for seeking similar parts between images through the analysis of the correspondence, similarity, and consistency from features, structures, relationships, etc. Currently, one of the hottest directions is applying drones with computer vision to realize autonomous navigation.
An unmanned aerial vehicle (UAV) or known as drone, refers to a class of unmanned aircraft operated by radio remote control equipment or its program control device. Because of such advantages as small size, low cost, and convenient usability, UAV has acquired more and more applications in recent years. Moreover, with the development of computer vision technology, there is an increasing demand that requires drones to integrate computer vision knowledge in order to handle more difficult tasks independently without human intervention.
This thesis is mainly composed of two parts.
In part one, we analysed various computer vision algorithms and selected a suitable algorithm as the basic algorithm under the consideration of calculation speed and accuracy. In the meanwhile, we discussed the characteristic of UAV images: aerial images are of large scale disparity compared with satellite map and we also have examined the matching difficulties brought by the scale differences. Then we improved the basic algorithm and proposed a new method that could resolve

this type of matching problem. The new algorithm is coined as "A fast matching algorithm for the images with large scale disparity". Finally, we conducted extensive experiments to demonstrate the effectiveness of the proposed method. In part two, we applied the fast image matching algorithm on the autonomous navigation of UAV. Firstly, we briefly introduced some preliminary concepts of UAV navigation. Then we discussed the conditions of UAV navigation and determined the essential parameters and navigation rules. Thirdly, we designed two algorithms to verify our conceptions: the first algorithm is to test the effectiveness of the conceived navigation regulations and to search for the critical parameters; the second algorithm is to simulate the real navigation purely based on computer vision, especially, the fast image matching algorithm. Finally, we also compared the proposed method to another navigation algorithm to show a better performance of the proposed navigation method. Furthermore, these results and conclusions could be combined with other methods to provide further support for UAV's navigation applications and expansions.

In this thesis, the fast matching algorithm for images with large scale disparity is proposed. The new method is of better matching performance and faster matching speed, which provides a possible solution for UAV navigation by means of computer vision. Then the UAV autonomous navigation algorithm is designed and the effectiveness is proved in the following experiments. Therefore, this thesis provides a smart and feasible navigation method for broader applications of UAVs in the future.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Computer vision is an interdisciplinary science which studies how to make machines "see". More precisely, it refers to the utilization of cameras and computers instead of human eyes to identify, track, and measure targets (1; 2). Most of the information that humans receive about the world comes from the vision. Similarly, computer vision has become the basis of machine cognition to the world (3). And the ultimate goal is to make computers "understand the world" like humans (4). Computer vision does not have a strict definition and the similar and not often distinguished are machine vision and robot vision (5; 6). In a narrower sense, computer vision can be equivalent to image processing, image analysis, or pattern recognition (7). Figure 1.1 shows that computer vision interacts many disciplines, and it provides the basis for other domains.

The pioneers in the field of computer vision could date back to earlier times, but it did not gain much attention and development until the late 1970s when computers performance improved enough to handle large-scale data such as images (8). However, these developments often originate from the needs of solving problems in other fields, and what is meant by computer vision has not been formally defined. Often, there is no formula for how computer vision problems

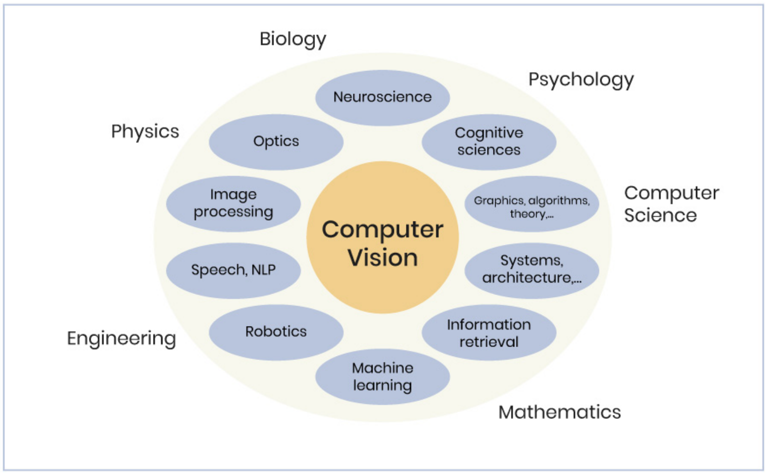


Figure 1.1: Computer vision is an overlapping field and it plays supportive role for other areas.

should be solved (9). Nevertheless, people have begun to master some methods for solving specific computer vision tasks.

With the performance improvement of computer hardware and the further development of brain science and neuroscience, recently, a kind of traditional method is renaissant, which is called Artificial Intelligence (AI) or Deep Learning (DL) (10; 11; 12). The basic idea behind AI and DL is utilizing computers to simulate the structure of the brain or the hierarchy of nerves, then using such a system to make decisions and judgments. Therefore, AI is widely applied to object recognition and pattern matching, which belongs to the central questions in computer vision (13). As one of the representative methods in deep learning, Convolutional Neural Networks (CNN), for example, is constructed by counterfeiting the visual perception mechanism of animals. The better performance is acquired by CNN in face recognition, fingerprints, text, and so on (14; 15).

The significance of computer vision is reflected in the problem it could solve. Computer vision enables the digital world to interact with the real world. It can equip many self-driving devices such as driverless cars, planes, and ships to make them conduct the senior operation without human assistance (16; 17). The autonomous equipment could greatly liberate human productivity. For instance, UAVs are increasingly applied in agriculture, infrastructure inspection, and portraying the three-dimensional map of cities (18). The Global Position System (GPS), which is often used as navigation guidance, however, may not be always available everywhere for the buildings' occlusion and the limited GPS receiving region (19; 20). Therefore, computer vision provides a possible solution for UAVs to address such trouble – GPS failure. Nowadays, a few explorations are developed in order to make UAV navigate and control themselves more smoothly and intelligently through computer vision technologies (21; 22).

Unfortunately, these computer vision methods including CNN are usually only applicable to resolve some specific problems. They cannot be widely applied in different situations. Normally, the application of these methods is a fundamental part of the large scale systems that solve more complex problems such as medical image processing, quality control, and measurement in industrial manufacturing (23). In most practical applications of computer vision, computers are pre-set to resolve specific tasks. Nevertheless, some algorithms based on machine learning are becoming popular. Only when the research on brain science is much more deeply developed and verified, the generic computer vision methods may come true in the future (24).

1.2 Related work

Image matching is a fundamental task of computer vision. Image matching refers to the procedure which identifies regions or structures with the same characteristic between two or more images through certain feature detection and description algorithms (25). For example, the matching between two grey images is achieved by comparing the correlation coefficient between the target area and the searching area within the same size window. By moving the searching window, the centre point of the window can be selected as the Eigen point when the correlation coefficient reaches the maximum compared to other places and this window could represent the most similar area (26). Its essence is the optimal searching problem based on similarity.

1.2.1 Classification of image matching

According to the characteristic of the image and the purpose of matching, image matching can be divided into different categories. Along with the different matching methods, many corresponding image processing algorithms also appeared.

Pixel value-based matching. This type of matching is commonly carried on grey images. The basic idea is that images are taken as two-dimensional signals, and then the statistics information is utilized to find the correlations between these signals. Grey matching often uses below similarity measurement functions such as correlation function and covariance function to evaluate the similarity and finally to determine the corresponding relationship between images (27).

A classic grey image matching algorithm is the normalization grey image matching method (28). And the core procedure is that we take the pixel values within a fixed window as a matrix so that the searching for matching is converted to measuring the similarity between two window matrices separately in the target image and searching image. The advantage of grey matching is that the principle is simple and easy to realize in coding, but the disadvantage is that the computation is complex and time-consuming. So, it is not suitable for applying in real-time tasks. Nowadays, however, some acceleration methods appeared such as FFT algorithm (29) and amplitude sorting algorithm (30).

Feature-based matching. This type of matching is commonly carried on a colour image. Image features mean some representative structures such as points, lines, areas, or senior features. The procedure of feature-based matching is that firstly extract some special features from two or more images separately; then describe these features and finally use those described parameters to establish the matching correspondence of features between images. Feature matching relates to many mathematical operations such as matrix operations, gradient solving, Fourier transforms, and Taylor expansions (31; 32; 33).

Accompanying different feature types, a lot of classic algorithms are proposed: Harris, SITF, ORB, and BRIEF, which are based on point feature (34; 35; 36; 37);

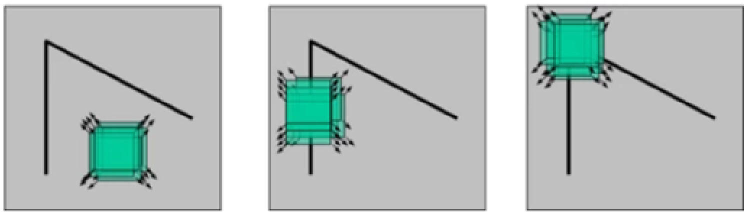


Figure 1.2: Harris corner detection.

edge feature algorithms include LoG operator and Sobel operator (38; 39). Regional feature algorithms include LC and HC (40; 41). Formula (1.1) displays the classic Harris corner detection algorithm, where $w(x, y)$ is the window function and represents the image gradient, [u, v] represents the offset, and [x, y] is each location within this window. Therefore, Harris equation could compute the changing of gradient and the place where prominent changing occurs is selected as a feature point. Figure 1-2 displays three conditions. In the first and second scenarios, there are no gradient changes. Only when the green window moves nearing the corner, the prominent gradient change will be detected in the third condition.

$$Harris(u, v) = \sum_{(x,y)} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \tag{1.1}$$

Relationship-based matching. Image matching based on a relationship is an exploring domain and it booms with the development of artificial intelligence on image processing. The normal solution is establishing a semantic network (42). It has made great progress but has not made a breakthrough.

1.2.2 Matching methods

Normally matching method is an independent part of computer vision. Different matching algorithms can be combined with different detection and description algorithms under specific demands (43; 44). For example, matching based on similarity is to calculate the distance among description vectors. Some classic matching algorithms include the sum of squared difference (SSD), the sum of absolute difference (SAD), and RANSAC (45; 46). Furthermore, some classification algorithms can also be used to a matching method such as SVM and KNN (47; 48).

1.2.3 Procedure of image matching

There is no fixed model of image matching procedure. The specific image matching process may vary according to the different characteristics of image features and different matching targets, but the basic procedure is as follows (49; 50).

Image obtaining. Digital image is generated by one or more image sensors. The sensors here can be various light-sensitive cameras including remote sensing equipment, X-ray tomography, radar, and ultrasonic receiver. Depending on sensors the generated picture can be a common two-dimensional image, a three-dimensional image group, or an image sequence. The pixel value of the picture often corresponds to the intensity of light in one or more spectral bands (grayscale or colour map).

Examination or division. Sometimes it is necessary to segment the image at first in order to separately extract more valuable parts for subsequent processing.

Pre-processing and features extracting. Some pre-processing are often adopted to make the images meet the requirements of features extracting methods and then certain feature or the feature combinations are extracted by implementing specific computer vision methods.

Matching. Take a comprehensive consideration of sample size, feature dimensions, matching efficiency and accuracy then select a suitable matching algorithm or choose a combination of multiple matching algorithms to do the matching.

1.3 Significance and contribution

With the development of computer vision and the emergence of civil unmanned aerial vehicles, a fresh demand is proposed - realizing the autonomous navigation of drones via CV technology. Computer vision could reduce the weight of a UAV and increase its battery life. Moreover, UAVs equipped with computer vision can cope with advanced tasks independently without human intervention. The ideal navigation reference image is the satellite map so the question of UAV navigation with the help of computer vision is converted to searching for correspondences between UAV aerial image and satellite map. However, there is a huge scale disparity for the same scenery between them for UAVs that normally fly at about five hundred meters while the satellite flies at several thousand meters above the ground.

In this thesis, firstly we propose a fast image matching algorithm, which is applicable to search for correspondences between images with large scale disparity. In other words, the fast matching method could bridge the huge scale gap for the same scenery separately exists on UAV aerial image and satellite map. In the meanwhile, the proposed method should be fast and accurate in order to be furtherly applied in real-time navigation.

Moreover, we extend the contents of the fast matching algorithm to make it possible to calculate the direction and the location of UAV which is the core concept in the navigation domain. We also propose the autonomous navigation rules for UAVs to simulate the navigation of drones based on the computer vision method. Finally, we apply the fast matching algorithm and the proposed navigation rules on the real UAV aerial images and google satellite map to realize the autonomous navigation of UAV.

1.4 Structure of this thesis

The remainder of this thesis is organized as follows:

In Chapter 2, firstly we discuss the pros and cons of several classic image matching algorithms then we select an algorithm as the basic method by balancing on time and accuracy in real-time processing. Secondly, we discuss the characteristics of UAV and satellite images and the matching difficulties due to the large scale disparity. Finally, we propose a novel fast image matching algorithm and present the pseudocode, and the flowchart. Furthermore, in order to utilize this method in navigation, we extend it with other functions to compute the location and orientation of the UAV.

In Chapter 3, we firstly review some basic concepts of UAV and explained several related and easily confusing terminologies in the UAV navigation domain. Then we describe the simulation environment and the navigation rules. Finally, we propose two simulation algorithms to imitate the UAV flying within a satellite map. In this process, we also apply the fast matching algorithm proposed in Chapter 2 to simulate autonomous navigation based on computer vision.

Chapter 4 is the experimental part. Firstly, the performance of the fast matching algorithm is verified and is compared with an alternative method. The comparison results show the proposed method is faster and more accurate in terms of image matching. Secondly, we verified the navigation algorithms and combined them with the fast matching method to simulate the autonomous navigation of UAVs. Again, we use an alternative method to acquire the results for comparison and the simulation experiments illustrate that the fast matching algorithm is of better performance.

Chapter 5 concludes the whole thesis, where we summarize the achievements and provide the direction of future research.

Chapter 2

A Fast Matching Algorithm for Images with Large Scale Disparity

2.1 Introduction

Object recognition and matching is a basic problem in the field of computer vision (25). Given two images with the same scenery but photographed under different circumstances such as illumination changes and camera variances, it is needed to find out the correspondences of the same objects separately existing in two images. This procedure is also applied in image retrieval, camera calibration, and image registration, etc.

In order to realize object recognition and matching, some classic algorithms are invented. D. Lowe proposed SIFT, which is famous for its accuracy and robustness (35). However, SIFT is time-consuming, so it cannot be applied in reality directly. Recently S. Katta and S. Pabboju etc. introduced PCA into SIFT which downgrades the dimension of SIFT descriptor to accelerate the matching speed (51; 52). However, the detector of SIFT remains low efficiency. Speeded-Up robust features (SURF) is originally proposed by H. Bay (53). Several different versions of SURF are also proposed (54). The principle of SURF is similar to SIFT but the former utilizes filtering to replace the convolution. Therefore, SURF acquires great computation efficiency by sacrificing a little accuracy. ORB algorithm is famous for the speed, which is even faster than SURF (36; 55). However, the scale tolerance of ORB is much worse, resulting in unstable matching performances. Salient detection is another classic object recognition algorithm, but normally it cannot achieve satisfactory results when the object is of complex structure (56).

AlexNet achieved great success on image identification in 2012, which leads to Convolutional Neural Networks' revive in the computer vision domain (57). Based on the thought indicated by CNN, a lot of improved frames and algorithms are proposed in recent years (58; 59). Generally, CNN needs huge data to train the model and to adjust many parameters, therefore, it is good at resolving problems with big datasets such as face recognition and semantic analysis. There are also a lot of comparison researches between CNN and traditional methods (60; 61). Furthermore, some mixed methods are proposed by integrating traditional methods with CNNs, such as CNN-SIFT and SurfCNN (62; 63; 64).

This thesis aims to seek a possible solution to realize autonomous navigation of small civil drones. Considering the cost, the lightweight of the drone, and the current trends about UAV, applying computer vision technologies is a desirable way to achieve that goal. Although, deep learning obtains more and more interest, however, it is far from mature and it has not become a common method (65). The algorithm based on image matching and adopted by UAV navigation needs comparative efficiency and accuracy on speed, better tolerance on the scale, and rotation invariance. therefore, the SURF algorithm is chosen as a preliminary method in this thesis. Nevertheless, the generic SURF cannot be directly applied in UAV navigation because there exists a characteristic difference between UAV aerial images and map. Compared with a satellite map, the aerial image taken by a flying UAV is of relatively narrow vision and the objects on drone image have a higher clarity. On the contrary, the map photographed by a satellite has a much broader vision, but the scenery is extremely small with worse clarity. The scale and quality disparity greatly exceed the scale tolerance of default SURF. Furthermore, the exhaustive searching for optimal scale ratio damages the efficiency, as the UAV navigation requires real-time processing (66). In the following, we firstly introduce the SURF for completeness.

2.2 Principles of SURF algorithm

Generally, SURF is a combination of two parts: detector and descriptor. For the detector part, SURF utilizes the Hessian determinant to find the place and the scale of blob-like structures; for the descriptor part, SURF utilizes Haar wavelet to preserve the neighbour information. This kind of feature makes SURF have a tolerance for brightness change and it can be further extended (67; 68).

2.2.1 Detection of interest point

SURF utilizes filtering operation applied on integral image to replace convolution operation carried on normal image to acquire the speed. Let $I(X)$ be a normal gray image and then its integral image $I_{\sum}(X)$ is defined as below. Given a location $X(x, y)$, the pixel value at this point X on $I(X)$ equals to the summation of all pixel values which are confined by a rectangle region with the original point on the left-up and this point X on $I(X)$ and the formula is as formula (2.1). So for a given area $\sum$, the area value can be easily calculated by three additions $\sum = A - B - C + D$ as shown in Figure 2.1.

$$I_{\sum}(X) = \sum_{i=1}^{x} \sum_{j=1}^{y} I(x, y) \tag{2.1}$$

The Hessian matrix is used to find blob-like structures and it can also be used to select the appropriate scale. Given a point $X(x, y)$ on a grey image I, the Hessian

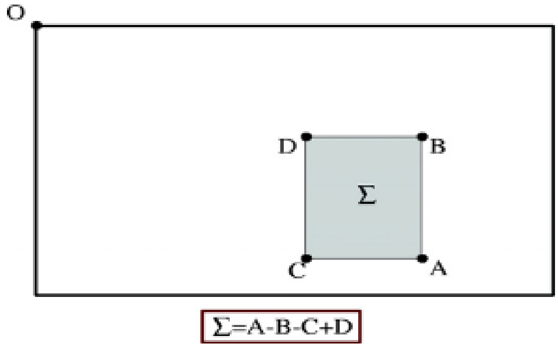


Figure 2.1: A region of intensities can be calculated in three additions on the integral image.

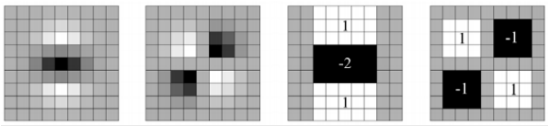


Figure 2.2: Left to right: templates of Gaussian second order partial derivative L_{yy} and L_{xy} separately; Approximations of corresponding box filters D_{yy} and D_{xy} respectively.

matrix formula with σ scale at X is defined as formula (2.2): where $L_{xx}(X, \sigma)$, is the convolution of Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I at point X. $L_{xy}(X, \sigma)$ and $L_{yy}(X, \sigma)$ have the similar meanings.

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \tag{2.2}$$

In order to simplify the computation, box filters can be applied on the integral image. Because box filter is the approximation of discretised and cropped the second order of Gaussian partial and mix-partial derivatives, thereby the convolution calculation could be reduced as filtering the integral image with box filters displayed in Figure 2.2. We denote D_{xx}, D_{xy} and D_{yy} as the filtering results which is separately corresponding to $L_{xx}(X, \sigma)$, $L_{xy}(X, \sigma)$ and $L_{yy}(X, \sigma)$, then the Hessian matrix (2.2) could be simplified to formula (2.3). The approximate determinant is acquired by formula (2.4).

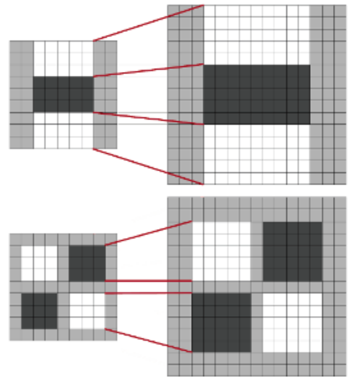


Figure 2.3: Filters D_{yy} (above) and D_{xy} (below) with two size: 9×9 templates (left) and 15×15 templates (right).

$$H_{approx} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.3}$$

$$Det(H_{approx}) = D_{xx}D_{yy} - w_2 D_{xy2} \tag{2.4}$$

It is worthy to point out that the initial size of box filters is 9×9 and the corresponding scale is $\sigma = 1.2$ as shown in Figure 2.2. In order to obey energy conservation and balance the approximation error, there is a weight w in formula (2.4) derived from formula (2.5). Furthermore, the filter responses need to be normalised to balance different sizes of box filters.

$$w = \frac{|L_{xy}(1.2)|_F |D_{yy}(9)|_F}{|L_{yy}(1.2)|_F |D_{xy}(9)|_F} \tag{2.5}$$

After above computations on integral image $I_{\sum}(X)$, at any point $X(x, y)$ with a given scale σ, we can figure out the approximation determination of point $X(x, y)$. By increasing the template size of box filters which is equivalent to increasing the scale σ as shown in Figure 2.3, for the same integral image $I_{\sum}(X)$ we can get different Hessian determination at point $X(x, y)$ on a higher scale. Now we can construct the scale pyramid. According to the size of filter template, normally SURF groups these filters into three octaves and each octave contains four layers. The permutation is shown in Figure 2.4. The gradually increased filter size is an

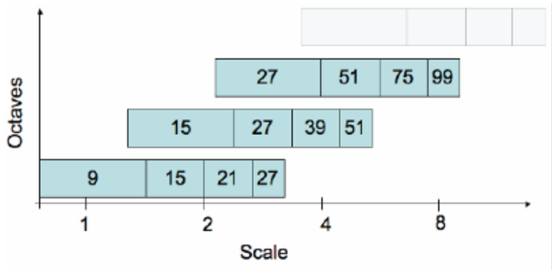


Figure 2.4: Filters and octaves permutation.

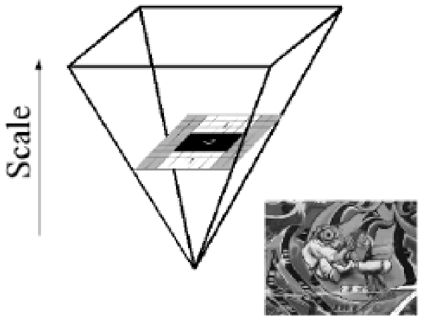


Figure 2.5: Scale space.

analogue to the increasing scale, and the image downgraded is as shown in Figure 2.5 (69). The first octave is composed of 9×9, 15×15, 21×21, and 27×27 filters, and the corresponding scale σ is 1.2, 2.0, 2.8, and 3.6. The second octave starts from 15×15 and the third starts from 27×27 which guarantees the continuity of scale change and accuracy.

When the filter's scale is comparable to the object's scale and the centre of filter is near to the interest point, the determinant achieves its extreme. As Figure 2.6 shown, we need to search for the extreme points in a $3 \times 3 \times 3$ neighbourhood in the scale pyramid then use non-maximum suppression (70) to select out these points whose determinants are extreme among its surrounding points as interest points then we use interpolation to get the accurate location and scale space information (71).

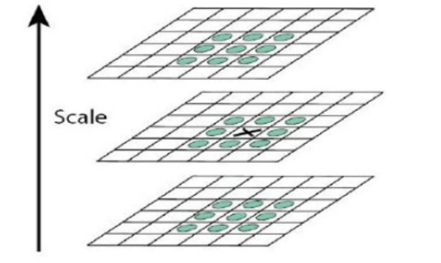


Figure 2.6: $3 \times 3 \times 3$ neighbourhood non-maximum suppression.

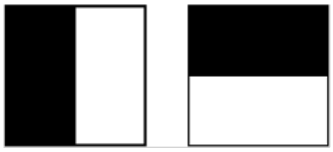


Figure 2.7: Haar wavelet templates in X and Y directions.

2.2.2 Description of interest point

In order to acquire the rotation invariance, a reproducible dominant orientation needs to compute for each interest. Firstly, we compute Haar wavelet response in horizontal and vertical directions within a circular region. Take the interest point as the circle centre and select $6s$ as the radius. The sampling step is s. The side length of Haar wavelet template is $4s$ as shown in Figure 2.7. Secondly, all these responses are weighted by Gaussian $\sigma = 2s$ centred at the interest point. A sector window with an angle of $\frac{\pi}{3}$ is gradually rotated around the interest point and within each sliding sector we sum up all the horizontal and vertical responses to get a local orientation vector and this procedure can be seen in Figure 2.8. Finally we choose the longest vector among all local vectors as the dominant orientation of current interest point. Now that the unique dominant orientation is defined, we could construct a rectangular with length $20s$, centred at the interest point and rotated to align with the dominant orientation. This rectangular is then divided into 4×4 sub-regions as shown in Figure 2.9. The sampling step is still s but the Haar wavelet template side length in the descriptor is chosen to $2s$. Again we calculate the horizontal and vertical responses (here the horizontal and vertical means align with the dominant orientation) and the responses are then weighted

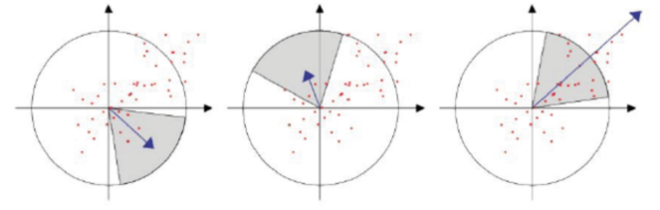


Figure 2.8: A sliding sector is used to find out dominant orientation.

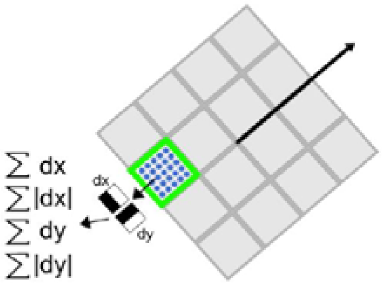


Figure 2.9: Descriptor and sub-region divisions.

with Gaussian $\sigma = 3.3s$ centred at the interest point. In each sub-region there is a four-dimensional descriptor vector $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|$ .The summation of the horizontal responses is represented as $\sum dx$, the summation of the vertical responses is represented as $\sum dy$, the summation of the horizontal and vertical absolute responses is separately represented as $\sum |dx|$ and $\sum |dy|$. Figure 2.10 displays that these responses represent different image local characteristic. Therefore, there are 4×4 sub-regions and each sub-region contains four parameters, totally $4 \times 4 \times 4 = 64$ dimensional vector is generated. Finally, this vector is divided by its length to get the unit descriptor vector for interest point.

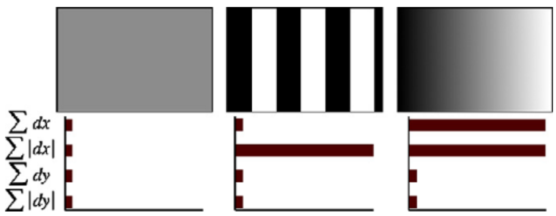


Figure 2.10: Different local characteristics.

2.3 Fast matching algorithm for images with large scale disparity

The flying height of a civil drone is commonly a few hundred meters, but the track height of the satellite is much higher from a few kilometres to thousands of kilometres. This makes a huge difference between the UAV aerial images and the satellite maps. Therefore, without any image pre-processing, it is not realistic to do matching among them by immediately using SURF (72). In order to simplify and clarify the discussion in the following, the aerial images captured by drones are taken at 500 meters above the ground. The satellite map which is composed of tiles is downloaded from Google satellite database level 15 which is equivalent to the vision height of 9600 meters. The website can be referred from (https://developers.google.com/maps/documentation/javascript/maptypes).

2.3.1 Optimal scaling ratio and matching criteria

After seriously analysing the UAV aerial images and satellite map, two critical differences are found. The first difference is that the perspective of the UAV is much bigger than the satellite's, which means the UAV image contains fewer objects, therefore the best matched objects on the aerial image only locate in a small and limited region on the satellite map. Indeed, the satellite map is composed of tiles so the seeking for optimal matching between UAV image and satellite map is converted into searching for best matching between UAV image and each tile consisting of satellite map (73). Hence, we need to define an evaluation formula to select out the best matched tile.

The second difference is there are a huge scale and resolution disparity between the UAV image and the satellite map or the tiles. The scale of the UAV image is much smaller, and the resolution is much higher. However, the tiles are of a much larger scale and terrible picture quality. Thus, before applying the SURF algorithm on the UAV aerial image and each tile, we need to balance both to a similar level.

On the one hand, in order to enhance the quality of tile and increase the number of interest points detected on tile, we double the size of each tile; on the other hand, we decrease the UAV image to make it compete with the double-sized tile to bridge the large scale disparity. For instance, the perimeter of a stadium takes up about 1172 pixels on UAV image, while the same perimeter only occupies about 141 pixels on the double-sized tile. That means the best scaling ratio maybe $141/1172 \approx 0.12$. More generally, the actual ground distances represented by each pixel on the UAV aerial image and tiles are different.

$$\alpha_{best} = \frac{D_{UAV}}{D_{map}} \times C \tag{2.6}$$

The formula (2.6) is defined to calculate the optimal scaling ratio, where D_{UAV} is the ground distance represented by each pixel on UAV image, and D_{map} is the ground distance represented by each pixel on the tile or satellite map. C is used to determine the quality of tile and in this thesis, we set it as two. If the drone's flight altitude is relatively stable, this ratio can be seen as a constant within a flying area. This fast matching method directly gives out the best scaling ratio thereby speeds up the whole algorithm.

Then the scaled UAV image I_{scaled} by the ratio α_{best} is matched to every tiles. The optimally matched tile is selected according to below formula (2.7) as the final matching criteria, where $Value$ is the evaluation index, i refers to the serial number of tiles, N_i refers to the number of matched pairs and $\bar{D_i}$ is the total average matching distances for matched pairs on each tile.

$$Value_i = argmin_i(\frac{\bar{D_i}}{N_i}) \tag{2.7}$$

Based on above two improvements, we could conceive the fast matching algorithm, and the pseudocode is listed in Algorithm 2.1 (74). Figure 2.11 displays the flowchart of this method.

Algorithm 2.1 Fast matching algorithm for images with large scale disparity

Require:

UAV aerial image I_{UAV}, satellite tiles $Tile_i$, i = 1,2,...,n and $C = 2$;

Ensure:

Best matching tile $Tile_b$ with I_{scaled};

$\alpha_{best} = \frac{D_{UAV}}{D_{Tile}} \times C$;

Reduce I_{UAV} with α_{best} to get I_{scaled};

Let $Value_i$ represent the corresponding matching performance between I_{scaled} and $Tile_b$;

for $i := 1$ to n do

Double the size of $Tile_i$;

Do the matching between the doubled $Tile_i$ and I_{scaled};

Matching performance is valued by $Value_i$;

end for

$b = argmax_i Value_i$ and $Tile_b$ is the best matching tile with I_{scaled};

return .

Besides, Ao used to propose an alternative method that uses a range of ratios to search for the optimal scaling ratio (66). In his algorithm, the initial value scale α_0 starts with the division of UAV flying height 500 meters to satellite vision height 9600 meters, that is α_0=500/9600≈0.05. And the range α_n is from 0.05 to 0.2 with step 0.01. In the process of gradually zooming out the UAV image - it is equivalent to technically elevating the height of the UAV up to the height of satellite - to find out the most suitable scaling ratio. Ao's method is exhaustive and it is time-consuming, so it is not suitable in navigation scenarios. However, as the recently only one method coping with a similar condition, we will use that method in comparison experiments.

2.3.2 Orientation and location

In order to apply the fast matching algorithm in the UAV navigation, acquiring the orientation and the location of the UAV is indispensable. Generally, the UAVs fly high so the aerial image can be assumed to be parallel to the satellite map. In other words, the rotation of the aerial image belongs to the in-plane rotation problem. We assume that the lens of the UAV is fixed on its body. Hence

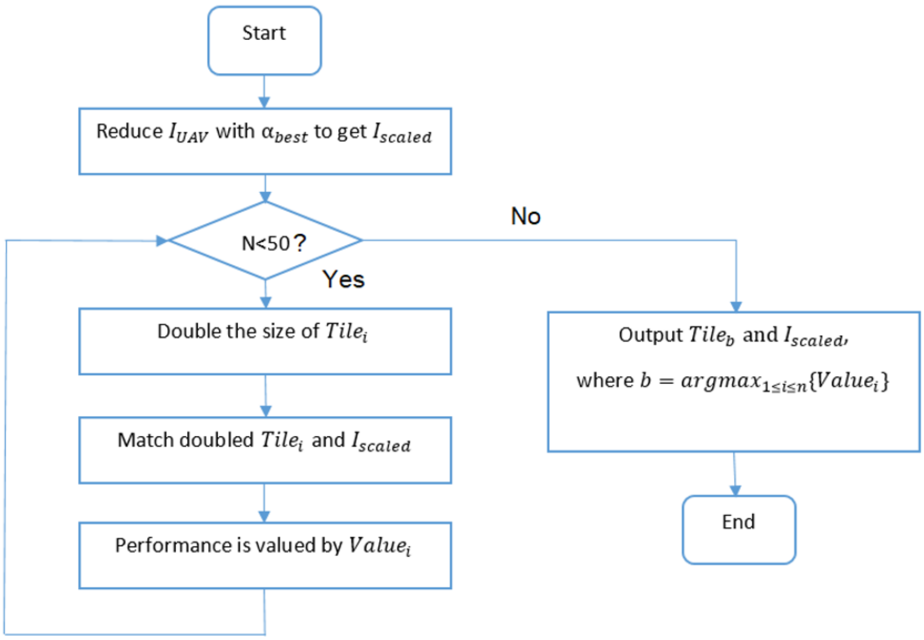


Figure 2.11: Flowchart of the fast matching algorithm

the rotation of the UAV is synchronized with the rotation of the aerial image photographed by the lens. Furthermore, we assume that the lens is perpendicular to the ground and the UAV is of little size. Thereby the centre of the aerial image can be seen as the location of the UAV.

The difference between α_{tile} and α_{UAV} separately existing in tile and UAV image can reflect the direction of UAV as shown in Figure 2.12, under the assumption that the satellite map occupies stable orientation as reference. The basic angle calculation is the arctangent function in formula (2.8) and the explanation is shown in Figure 2.12. For the image processing in computer vision, the top left corner is usually seen as the origin by convention. Points $A(a_1, a_2)$ and $B(b_1, b_2)$ define a straight line that conforms an *Angle* with the X-axis.

$$Angle = arcTan(\frac{b_2 - a_2}{b_1 - a_1}) \tag{2.8}$$

Given that the location of the matched interest points on a tile is known, the

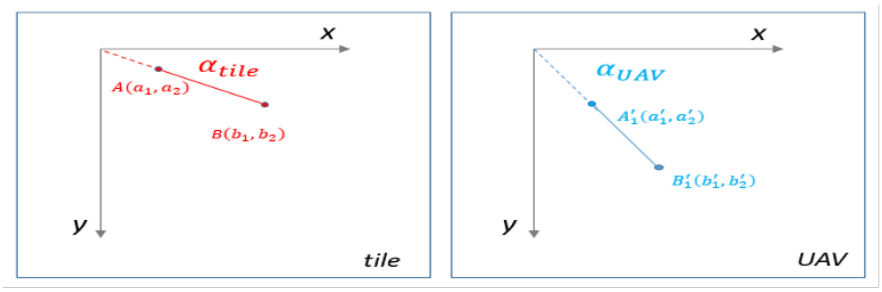


Figure 2.12: The angle can be calculated by arctangent value of two points coordinates.

orientation of the aerial image could be derived and the best scaling ratio can also be acquired by formula (2.6), then the corresponding location of each pixel on the aerial image can be computed. For convenience and representative, by assuming that the camera lens of the UAV is perpendicular to the aerial image, so the project location of the UAV is on the aerial image center. Then the location of the center can be calculated by the following formula (2.9), where Θ is the orientation of the UAV aerial image, x and y denote the coordinate on the rotated image, C_x and C_y denote the location point of rotation, $\begin{bmatrix} x' \\ y' \end{bmatrix}$ is the coordinates on the unrotated image, and the final location needs to be weighed by the optimal scaling ratio.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\Theta(x - C_x) - sin\Theta(y - C_y) + C_x \\ sin\Theta(x - C_x) + cos\Theta(y - C_y) + C_y \end{bmatrix} \times \alpha_{best} \tag{2.9}$$

Now that the orientation and the location of the UAV can be acquired, it is possible to implement UAV navigation by these measurements (75; 76).

2.4 Summary

In this chapter, we first reviewed the SURF algorithm because we take SURF as the base algorithm. Then we discussed the matching difficulties introduced by large scale disparity, especially in this scenario, the UAV aerial image and google map are taken for examples. Thirdly we proposed the corresponding methods to cope with each difficulty and finally we integrated these improvements to form a new algorithm: A fast matching algorithm for the images with large scale disparity. In order to apply the new method to UAV navigation, it is furtherly extended with another function to acquire the orientation and the location of the UAV.

Chapter 3

A Novel UAV Autonomous Navigation Approach

3.1 Introduction

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without a human pilot onboard and a type of unmanned vehicle. UAVs first appeared in the 1920s and they originated mostly in military applications (77; 78). Nowadays their use is rapidly expanding to civil, commercial, scientific, recreational, agricultural, and other applications, such as policing and surveillance, product deliveries, aerial photography, etc. Technically, UAVs can be divided into fixed-wing UAVs, vertical take-off and landing UAVs, unmanned airships, unmanned helicopters, and unmanned parachute aircraft, and so on (79; 80; 81). Figure 3.1 displays four types of UAVs. Compared to crewed aircraft, UAVs are more suitable for dull and dangerous missions. In the past few decades with the development of automatic control theory and communication technology the autonomous navigation of UAV by Global Positioning System (GPS) has achieved a satisfactory level, however, along with the boom of computer science, there is a higher level of demand on realizing the self-contained autonomous navigation by means of computer vision knowledge (82). The self-contained autonomous

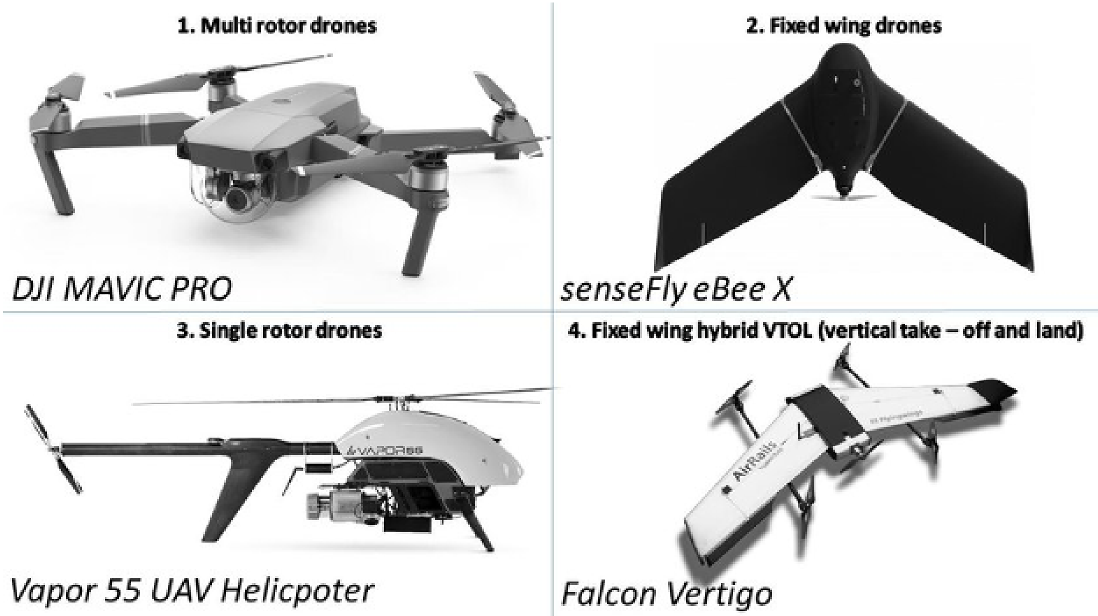


Figure 3.1: Different kinds of UAVs.*(The picture is from Journal of Civil Engineering: Utilizing drone technology in the civil engineering.)*

navigation is of anti-interference and more intelligence which could be applied in a more complex circumstance to finish a more difficult task.

3.2 Terminologies and concepts relating to UAV navigation

Speaking of UAV navigation, in order to have a better understanding, it is needed to introduce some basic ideas and sometimes easily confusing terminologies: guidance, navigation, and control, which has an abbreviation GNC (83). GNC is a branch of engineering dealing with the design of systems to control the movement of vehicles, especially, automobiles, ships, aircraft, and spacecraft. In the following, the relating concepts of GNC targeted in the drones domain are briefly introduced (84).

3.2.1 Guidance, navigation and control

Guidance: the main task of guidance is “knowing where the goal is and how to reach it”. UAV obtains the commands of position or speed required to reach the target, in the meanwhile the UAV needs to consider its current status and external environment. For example, when flying according to a planned route, UAV calculates the instructions for itself to go in straight or in a bent path along that route. What’s more, if there are obstacles or no-fly zones, UAVs need to calculate the feasible circumvention route or speed command, based on the drone’s flight performance (85; 86).

Navigation: the main task of navigation is ”knowing where you are and knowing your posture”. UAV obtains its current position, velocity, and other information (under a certain reference frame). If necessary, the drone also needs to acquire the current posture, attitude, and angular velocity. For example, a pure inertial navigation system can obtain the position, velocity, and acceleration of the drone in an inertial system, as well as the attitude angle and angular velocity relative to the inertial system. Figure 3.2 shows a common inertial navigation system with large volume and mass, which is not suitable for drones with small size. The GPS navigation system can provide information such as the speed, position, and heading angle of the drone in the World Geodetic System 1984 coordinate system (87). VICON and UWB belong to indoor positioning systems which could provide the speed and position information for reference (88; 89).

Control: the main task of control is “changing the flying posture and following the guidance instructions”. According to the current speed, posture, and other information, the UAV changes the following posture and speed parameters through specific manipulations, thereby the drone can achieve stable flying or track under the guidance instructions (90). For example, when a fixed-wing drone needs to climb at an altitude, UAV needs to issue a command to calculate the required pitch angle and pitch angular speed, at the same time the throttle command is required to keep the drone speed from decreasing significantly. Another example,

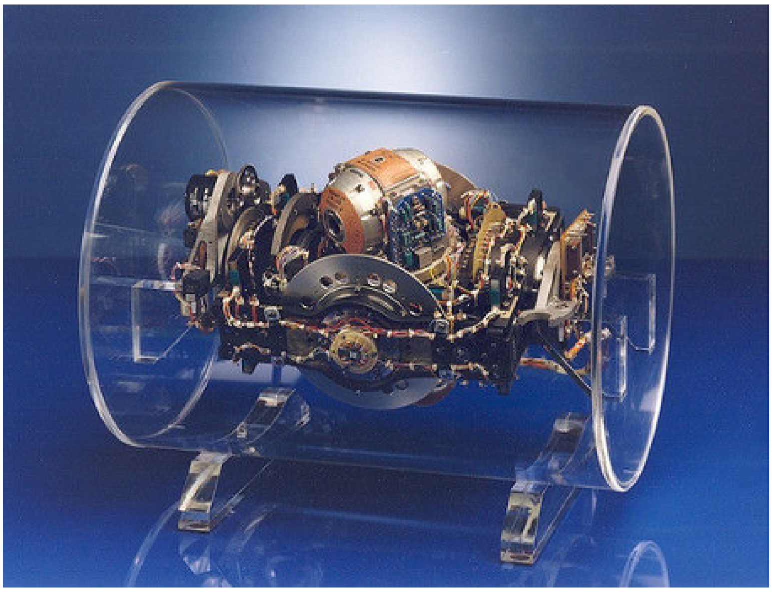


Figure 3.2: Common inertial navigation system with large volume and mass.*(The picture is from the restored Litton LN3-2A inertial navigator platform—Showcase by TJvV—1990.)*

if there is a crosswind, UAV needs to issue a command to calculate the required yaw angle and then use side slip to offset the effect of crosswind (91).

Theoretically, navigation, guidance, and control are dividable and responsible for different functions separately - they have an inheritance relationship in the calculation and execution of instructions. However, in actual systems, the three parts may have many overlapping areas and in the computer vision domain, the three parts are gradually melting together to make the UAV smarter and resolve the problem automatically (92; 93).

3.2.2 Categories of navigation

Although there are many different navigation principles, a UAV navigation system can be roughly divided into three classifications.

Navigation based on an absolute frame of reference. Inertial navigation and magnetic compass navigation are representative of this type. Inertial navigation utilizes the principles of Newtonian mechanics and constructs an inertial

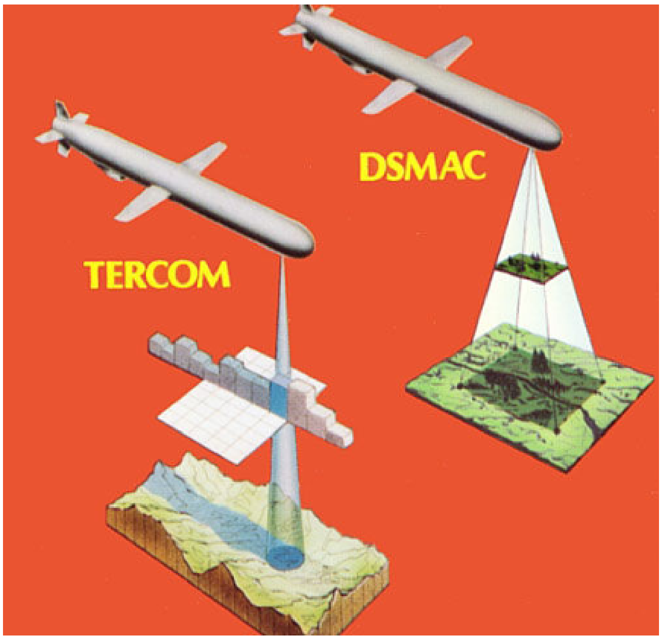


Figure 3.3: Navigation based on machine vision to capture and compare landform features.*(The picture is from General Dynamics - Cruise Missiles 1985.)*

platform that is stably connected to the aircraft (94). The speed, position, and posture angle information of an aircraft are derived from high-precision equipment. Although the accuracy of the platform inertial navigation is very high, it is not convenient to equip it on a small aircraft for the complex and bulky system. Besides, the error accumulation is hard to avoid, usually, inertial navigation needs other navigation information to amend the result (95).

Navigation based on distance measurement. Satellite navigation and indoor positioning belong to this category. This type of navigation method calculates the aircraft position by measuring the distance between the aircraft and a series of known precise locations. The popular WIFI positioning uses signal strength and the time difference between sending and receiving to compute the distance between the aircraft and each reference point, to calculate the real-time position of the aircraft (96).

Navigation based on features. Such as terrain matching, motion capture systems, and so on. This type of navigation method usually uses the real-time geomagnetic, geomorphological, and image features extracted by UAVs and then compares this information with the feature database or performs corresponding calculations to obtain the position and speed of aircraft and to realize the navigation function. For example, the terrain matching method used in cruise missiles and the currently popular Synthetic Aperture Radar (SAR) landform matching method, both extract one- or two-dimensional terrain and landform information on the flight path and then compare them with the digital elevation map library thereby know the current position, speed and other information, which is of great significance for long-term navigation when the satellite navigation signals may be lost (97). Figure 3.3 displays the cruise missiles navigate themselves according to the image or landform features. The methods of applying computer vision technology - identifying the features of markers at known locations - to estimate the position and velocity also fall into this category (98; 99).

Currently, most UAVs still use inertial navigation and satellite navigation combination as the basic navigation method, which can ensure stable navigation in most scenarios. However, more and more small-size civilian drones appear, and they are playing an increasingly important role in harsh environments such as earthquakes and disaster rescue. With the development of computer vision, more and more research concerns the intelligence and autonomous navigation of drones. This mode of navigation based on image features is our navigation method in this thesis.

3.3 An UAV visual autonomous navigation method

Traditionally, UAVs are big and heavy, as the navigation of them often relies on inertial equipment with larger volume and mass. Moreover, sometimes the navigation also needs the cooperation from humans and machines on the earth (100). In recent years, with the appearance of light and small drones and the development of computer technology, there is a rising demand to realize drone navigation in more intelligent and self-contained ways (101; 102). In the following, we attempt to utilize computer vision techniques to simulate the autonomous navigation of drones. Specifically, some landmarks are assumed as known locations on the map, and the drone is navigated by using these landmarks with the aid of captured images by the drones. The drone takes a photo of the scene underneath regularly and maps the objects to the known landmarks by the fast matching algorithm proposed previously. Ideally, once there are two pairs of correct matching, the orientation and location of the drone can be calculated so that the navigation of the drone towards the target landmark is feasible.

3.3.1 Simulation environment and navigation rules

The assumptions of this navigation simulation contain below five parts.

Reference map. There is a standard map as a UAV navigation reference map. In the lab, we place 10 toys in order as shown in Figure 3.4. The starting point is the left corner of the map (landmark 1), and the destination is the bottom right corner (landmark 10), with the requirement that the drone has to go through all landmarks one by one. It is assumed that the locations of all landmarks are known on the map, and the landmark is properly chosen so that all photos taken by the drone cover at least two neighbour landmarks. In this way, we can always navigate the drone using each photo taken. This is easy to achieve if the height of the drone and the distance of landmarks are known.

Speed and orientation correction angle of UAV. Assuming that the UAV is flying with a stable scaler speed and each time it only changes its direction with certain degrees – orientation correction step - to approach the right angle as the drone has some inertial so it can not adjust the right direction immediately. If the angle between the objective orientation and the orientation of the UAV is less than the orientation correction step, the UAV will adjust the orientation to align with the objective orientation. Normally the UAV needs to turn its direction several times and finally fly straight to the objective landmark.

Landmarks matching. Based on sampling rate, which is equivalent to the speed of UAV, in each step the UAV takes an aerial image and compares it with the standard map. The closest landmark except those already visited will be selected as the next flying target. Then we use the matching algorithm to calculate the distance and angle to the next target landmark. If the distance is smaller than a threshold, the UAV is assumed to have already passed in the current landmark then the next near landmark is selected as the future target landmark. If the distance is bigger than the threshold, the UAV will move one step and repeat the matching procedure. Here the threshold is 1.5 times the UAV speed.

Projective location of UAV on map. For convenience, we assume that the lens of the UAV is perpendicular to the map and the location of the UAV is simplified to the location of the centre of the lens. Therefore, the location of the UAV is located in the centre of its captured aerial image. Thanks to the extension part of the fast matching algorithm explained in the previous thesis, we could get the projective location and the orientation of the drone on the reference map as illustrated in Chapter 2.

Navigation trajectory. Finally, each projected location of the UAV is connected to get the entire route of the UAV. To compare the result, the method proposed by Ao is used to compute the projected location (66). A ground truth route is obtained by assuming all landmarks having been already perfectly matched.

3.3.2 Autonomous navigation algorithms of UAV

In order to realize simulations of UAV navigation based on the above assumptions and navigation rules, below two types of simulation algorithms are designed.

Algorithm 3.1: In the first algorithm, there are two purposes: one is to verify the effectiveness of our assumptions and rules – the UAV could arrive at each landmark with order sequence and the other one is to search for the suitable parameters on the UAV speed ($Velocity$) and orientation correction step ($Step_{angle}$) – the UAV would arrive landmark without detouring too much. In algorithm 3.1, the real image matching method is omitted, we assume that the matching is perfect therefore an ideal navigation route can be acquired, and then the route is

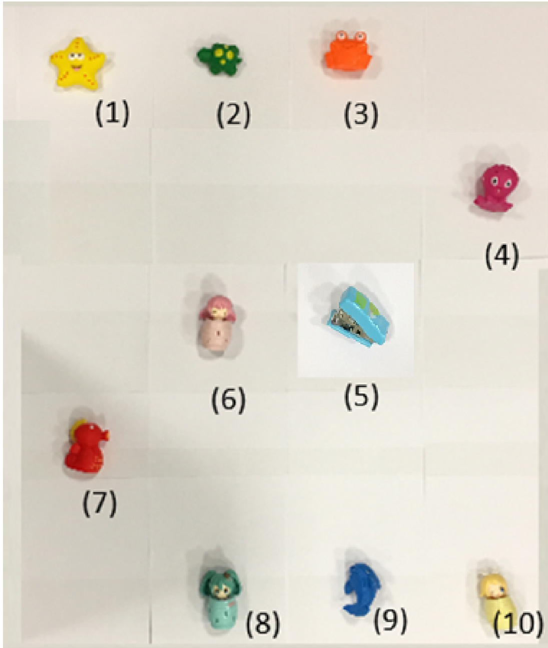


Figure 3.4: A sequence of landmarks as reference.

taken as the reference trajectory to evaluate the real navigation performance in the second algorithm. The first algorithm pseudocode is shown in Algorithm 3.1.

Algorithm 3.1 Fast matching algorithm for images with large scale disparity

Require:

Reference map, initial orientation and location, the sequence of $Landmark_i$, $i = 1, 2, ..., N$ with coordinates, $Velocity$, $Step_{angle}$ and $Dis - threshold$;

Ensure:

Trajectory of UAV flying on reference map;

Let $Place$ represent the final trajectory and $Location$ denote trajectory piece between each landmark;
Specify $Place_1$ with $Location_1$;
for $i = 1$ to N do
Input $Location_1$ with $Landmark_i$;
j is assigned by 1;
while distance($Location_j, Landmark_i$) $>$ $Dis - threshold$, do
j is added by 1;
$Location_j$ records new position after UAV forwards by $Velocity$ with orientation;
if difference(orientation, direction of $Landmark_i$) $>$ $Step_{angle}$, do
Adjust orientation closer to direction of $Landmark_i$ by $Step_{angle}$;
else
Adjust orientation align to direction of $Landmark_i$;
end if
Calculate distance($Location_j, Landmark_i$);
end while
Append $Place$ with $Location$;
$Location_1$ is initialized by $Location_j$;
end for
Outline the whole trajectory on map based on $Place$;
return .

Algorithm 3.2: The second algorithm will simulate the real navigation of the UAV. In this procedure we input the UAV aerial image which is taken at the initial place then the UAV would match this captured image with the objective landmark to acquire its projective location and orientation. If the orientation is not aligned with the landmark, UAV would also adjust its direction with orientation correction step nearer to the landmark, in the meanwhile, if the distance between the projective location and landmark location is larger than the distance threshold, the UAV would fly a stable distance along with the adjusted orientation. And then at the new place, the UAV would take another aerial image and match the newly captured image with a landmark to get the updates of the current location and orientation. If the orientation is in accord with the landmark's, the UAV direction would not change, besides, if the distance is smaller than the distance threshold, we assume that the UAV has already arrived at the current landmark and the next landmark is assigned as the objective landmark. The above procedure is processed automatically, and projective location series are recorded by the second algorithm therefore it can simulate the real UAV navigation route. The real navigation simulation algorithm pseudocode is displayed in Algorithm 3.2 and in order to highlight the simulation procedure by integrating the matching method, the matching part is written in function as shown in Algorithm 3.3.

Algorithm 3.2 UAV Navigation Algorithm in Real Matching Mode

Require:

Reference map, initial image, the sequence of $Landmark_i$, $i = 1, 2, ..., N$ with coordinates, $Velocity$, $Step_{angle}$ and $Dis - threshold$;

Ensure:

Trajectory of UAV flying on reference map;

1: Let $Place$ represent the final trajectory;
2: **for** $i = 1$ to N do
3: Call navigation function and deliver aerial image, $Landmark_i$, $Velocity$, $Step_{angle}$ and $Dis - threshold$;
4: Aerial image is updated which is taken near to $Landmark_i$;
5: Append $Place$ with $Location$;
6: **end for**
7: Outline the whole trajectory on map based on $Place$;
8: **return** .

Algorithm 3.3 Fast matching algorithm for images with large scale disparity

Require:
Image, $Landmark$, $Velocity$, $Step_{angle}$ and $Dis - threshold$;
Ensure:
Aerial image taken near $Landmark$ and $Location$;

Let $Location$ denote projective locaion;
$Location_1$ and orientation are acquired by matching aerial image with map;
j is assigned by 1;
while distance($Location_j, Landmark$) $> Dis - threshold$, do
j is added by 1;
$Location_j$ records new position after UAV forwards by $Velocity$ with orientation;
if difference(orientation, direction of $Landmark$) $> Step_{angle}$, do
Adjust orientation closer to direction of $Landmark$ by $Step_{angle}$;
else
Adjust orientation align to direction of $Landmark$;
end if
Take an aerial image and match it with map;
$Location_j$ and orientation are updated according to matching result;
if difference($Location_j$, $Landmark$) $< Step_{angle}$, do
Aerial image is updated with new aerial image;
end if
end while
return Aerial image and $Location$.

3.4 Summary

In this chapter, we first introduced the different types of UAVs and several similar and easily confusing terminologies to make the illustration of navigation clearer and more prominent. Secondly, we described navigation rules and simulation scenarios where the UAV is navigated only by computer vision techniques. Finally, we proposed two algorithms to verify the legitimacy of the navigation rules and to apply the fast matching algorithm in simulation. The experiment results are listed in the next chapter because these experiments utilize the fast matching algorithm and there are coherent links between each experiment.

Chapter 4

Experimental Results and Analysis

In chapter 2, we proposed the fast matching algorithm for the images with large scale disparity, which is suitable for seeking matching between images with the same scenes but without compatible scales. One of the actual applications to exploit this method is that matching the aerial images taken by the UAV with the satellite map. If the performance of this algorithm is acceptable, the UAV navigation based on the computer vision is expectable. Furthermore, we could achieve this goal: the drones can realize autonomous navigation without external input and assistance.

In chapter 3, we proposed the navigation rules of UAV and designed two navigation simulation algorithms. The first algorithm is conceived to verify the effectiveness of the navigation rules and search for suitable simulation parameters, and the second one is conceived to combine with the fast matching algorithm to achieve autonomous navigation only by utilizing computer vision technologies.

Therefore, this chapter can be divided into two phases. The first part presents the experimental results of the fast matching algorithm, which is shown to be better than an existing algorithm. The second part displays the comparison results of navigation simulation by two methods – the fast matching algorithm and Ao's method. The first phase illustrates the concrete matchings between UAV aerial images and satellite maps, which provides the basis for the whole navigation trajectory simulated in the second phase. In the following, the main algorithms are operated on the MATLAB platform.

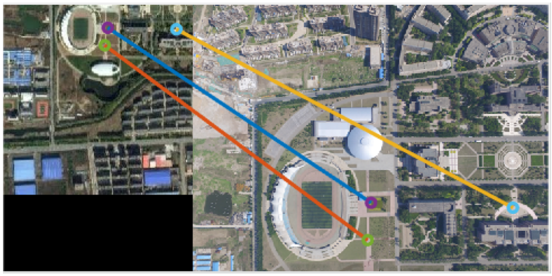
4.1 Experiments for the fast matching algorithm

In order to verify the matching accuracy and the speed improvement of the fast matching algorithm proposed in chapter 2, we design the following two experiments. For comparison, the experiment data are accord with Ao's data: the satellite map is Google map with 15th level and four aerial images are provided by Ao's paper.

Matching in unrotated condition: In this scenery, UAV images are parallel to the corresponding Google map. This condition is also verified by Ao's method. So, we expect to see that the correct tile could be selected out by the proposed method, i.e., the correct scenery is located. In the meanwhile, the computation time could be shortened dramatically compared with Ao's algorithm. The matching results, computation time, and the paired numbers of interest points will be recorded.

Matching in rotated condition: This scenario is that UAV images will be rotated with some angles before matching while the direction of the corresponding google map or tile is unchanged. Therefore, the effectiveness of the fast matching method can be tested on any angle and this is the common condition because actually, affected by the airflow or other factors, the direction of UAVs cannot be strictly parallel to the ground. In this experiment, the true direction and the orientation calculated by the fast matching algorithm will be recorded.

(a)

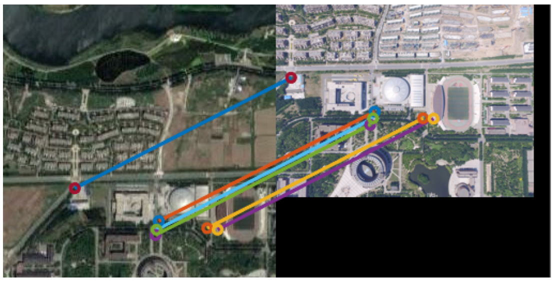


(b)

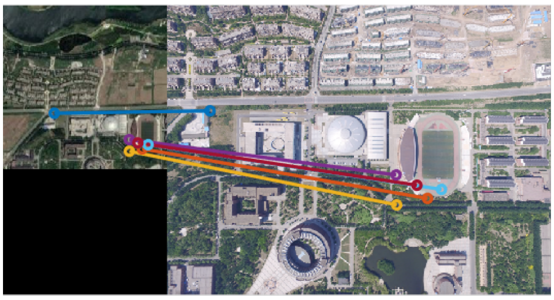
Figure 4.1: The same tile is selected and matched with the same aerial image. 6 matched pairs are found by the proposed fast algorithm as shown in Image (a), in the contrast, only 3 matched pairs are found by Ao's method as shown in Image (b).

4.1.1 Matching in unrotated condition

In this experiment, we would like to compare the performance between the fast matching algorithm and Ao's method applied in this scenario: the orientation of the aerial image taken by UAV is aligned with the direction of the satellite map. Figure 4.1 to Figure 4.4 displays four groups of image matching results: the matched tiles are on the left and the UAV aerial images are on the right. Each group contains two image matching results: the upper one is acquired by using the fast matching algorithm and the lower one is achieved by using Ao's method. It can be seen clearly that all UAV aerial images are respectively matched with correct small regions - tiles - on the satellite map.

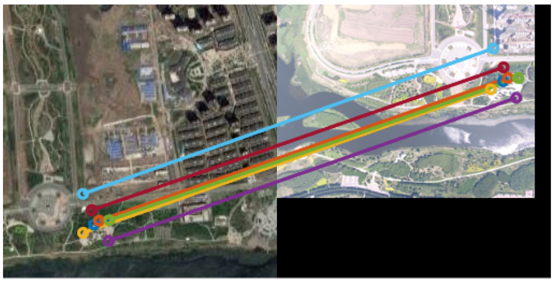


(a)



(b)

Figure 4.2: The same tile is selected and matched with the same aerial image. Both methods find 6 pairs of matched points. The scale ratio searched by Ao's method as shown in Image (b) is larger than the scale ratio adopted by the fast matching algorithm as shown in Image (a).

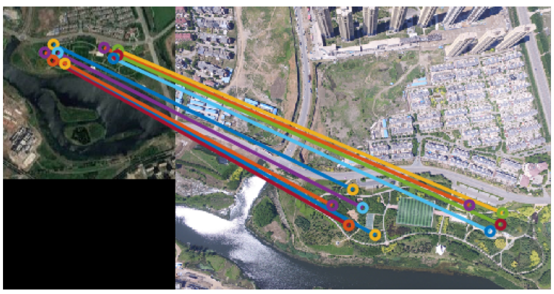


(a)

(b)

Figure 4.3: The different tile is selected and matched with the same aerial image. 7 matched pairs are found by the proposed fast algorithm as shown in Image (a), in the contrast, only 3 matched pairs are found by Ao's method as shown in Image (b).

(a)



(b)

Figure 4.4: The same tile is selected and matched with the same aerial image. The scale ratio searched by Ao's method as shown in Image (b) is larger than the scale ratio adopted by the fast matching algorithm as shown in Image (a).

The detailed comparison is summarised in Table 4.1. The first column is the serial number of images separately matched with tiles by using the fast method proposed and Ao's method. The second column is the time consuming with metric unit second, and the third column is the number of matched pairs with different methods.

As Table 4.1 shows, the average matching time for each UAV image is 23.4 seconds when applying the fast matching method. In the contrast, an average of 722.3 seconds is consumed by Ao's method. The proposed fast matching method is approximately thirty times faster than Ao's method, so it illustrates a huge improvement in the speed. Moreover, the overall numbers of the matched pairs applied with the fast method are better especially on image A and image C. As shown in Figure 4.1 and Figure 4.3, the numbers of the matched pairs are doubled. And these matched pairs are more centralized. The new fast algorithm not only maintains the accuracy of matching but also accelerates the matching speed greatly which means that it could be expected to be applied on practical application.

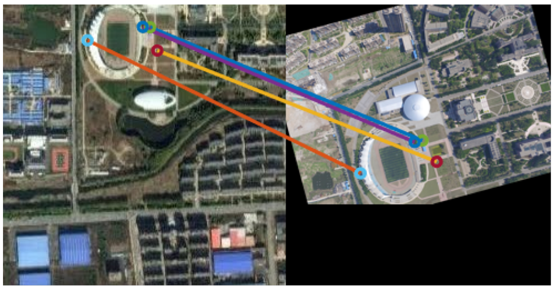
Table 4.1: Comparisons on time-consuming and numbers of matched pairs.

Image No.	Matching time (second)	Numbers of matched pairs
Image A using our method	**23.0**	**6**
Image A using Ao's method	703.7	3
Image B using our method	**24.1**	**6**
Image B using Ao's method	749.9	6
Image C using our method	**23.1**	**7**
Image C using Ao's method	738.1	3
Image D using our method	**23.3**	**7**
Image D using Ao's method	697.5	9

4.1.2 Matching in rotated condition

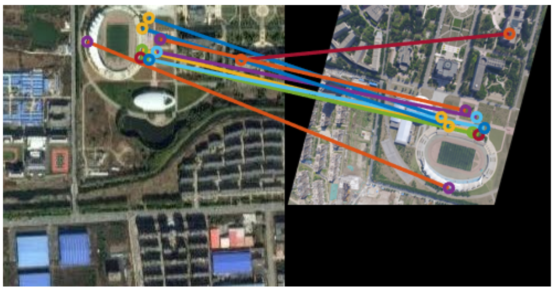
In this experiment, we would like to verify the capacity of the fast matching algorithm on the images with the same scene but with different directions. The first work is to verify whether the correct matched tile can also be selected out by using the new algorithm when the given UAV image is not parallel to the tile. The second work is to estimate the direction of the given UAV image according to google's satellite map. Figure 4.5 and Figure 4.6 display two groups of matching results: the matched tiles are on the left and the rotated UAV aerial images are on the right. Each group contains three image matching results. It can be seen clearly that all rotated UAV aerial images are respectively matched with correct small regions - tiles - on the satellite map.

Limited by space, only six rotation matching results are displayed in Figure 4.5 and Figure 4.6. The detailed orientation estimation in image rotation matching is summarised in Table 4.2. Here we test the first two UAV aerial images A and image B which are the aerial images displayed on the right parts in Figure 4.1 and Figure 4.2. The number of Image A and B is denoted in the first column in Table 4.2. The second column is the sequence of real rotation angles in degree. The last column is the corresponding average angle calculated by selecting three couples of matched pairs randomly. It can be seen clearly that the accuracy of the calculated rotation angle or direction is competitive close to the real image rotation angle: the average deviation in the image A rotation sequence is 1.768 degrees and the average deviation in the image B rotation sequence is 0.993 degree. The fast matching method can acquire the direction angle by the meaning of matching pairs, so it is further expected to be applied in UAV autonomous navigation.



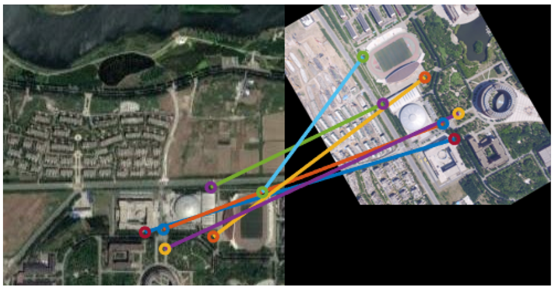
(a)

(b)

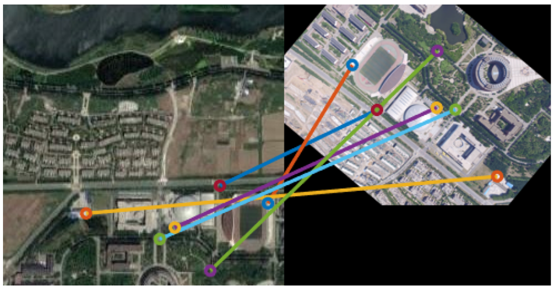


(c)

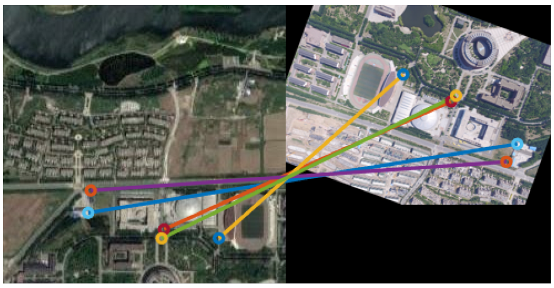
Figure 4.5: The correct tile is selected and matched with the rotated aerial image A on three angles separately, in anti-clockwise. Image (a) displays the matching result of rotation 15 degrees. Similarly, Image (b) and (c) demonstrate the matching results under 45 and 75 degrees rotation.



(a)



(b)



(c)

Figure 4.6: The correct tile is selected and matched with the rotated aerial image B on three angles separately, in anti-clockwise. Image (a) displays the matching result of rotation 120 degrees. Similarly, Image (b) and (c) demonstrate the matching results under 140 and 160 degrees rotation.

Table 4.2: Comparisons of real scene direction with calculated scene rotation direction.

Image No.	Real image direction (degree)	Calculated image direction (degree)
Image A	15	16.44
	30	29.57
	45	49.10
	60	59.82
	75	76.25
	90	93.21
Image B	110	109.07
	120	117.15
	130	130.09
	140	138.96
	150	149.41
	160	159.54

4.2 Experiments for UAV navigation simulation

In order to verify the effectiveness of the proposed navigation algorithms proposed in chapter 3, four experiments are conducted in this section as described below. As not having enough real UAV images with real captured images at each location along with the flying trajectory, the first three experiments are based on the simple simulated environments in our lab and only the last experiment is based on the real data from the authors.

1) Searching for appropriate speed and correction angle of the UAV: the first experiment aims to verify the effectiveness of our Algorithm 3.1 and to search for two key parameters: the line speed and rotation correction angle of the UAV. The simulated trajectory of the UAV should be effective which means the routine should connect each landmark effectively without unnecessary detours. Meanwhile, the navigation trajectory should display that the drone can correct its orientation several times before it flies straight to the next landmark as the UAV is of some inertial. The unacceptable simulation is that the UAV takes a huge detour or even flies out of the map. To this end, different settings of speed and rotation angle step are tested to find a suitable couple that gives the best route.

2) Comparing performance with two image matching algorithms in the simulated environment: in this experiment, the matching performance based on our new method – fast matching algorithm for images with large scale disparity will be compared with another matching method proposed by Ao. The matching result should be accurate enough to be applied in real navigation. The comparison is based on the same simulation trajectory acquired in previous experiment 1) - the ideal projective location sequence. At each projective spot, first, an aerial image is taken, and the direction of the image is aligned with orientation defined by current and next spots, and then the captured image is matched with the map to calculate the place of UAV in both methods separately; finally, the distance offset sequence between computed locations and the spot sequence is checked to get the comparison result.

3) Simulating real UAV navigation based on image matching algorithms: the purpose of this experiment is to verify the proposed navigation Algorithm 3.2 in the simulated environment. The UAV flies through each landmark from the top left landmark 1 to the bottom right landmark 10. In this simulation procedure, the UAV navigates purely based on the orientation and location obtained by an image matching algorithm. Also, Ao's method is taken as a comparison. The UAV needs to rectify its navigation only based on the initial aerial image taken at the starting region and the map. At the same time, the speed and correction angle are according to the ones having been approved in previous experiments.

4) Navigation simulation based on real aerial images and satellite map: in the last experiment, the navigation simulation of drones based on a real satellite map is carried on. The orientation is acquired by the matching result between the real aerial images taken by drones and the Google map. Among landmarks, the drones would follow the former assumptions having been applied and verified in the previous experiments – simulating that the drone is of stable speed and rotation correction angle. Each landmark region is seen as a calibration region that provides the references for the UAV to navigate with the order and direct the drone to arrive at the final destination without losing direction.

4.2.1 Determination of simulation parameters

The purpose of this experiment is to search for the appropriate speed and correction angle as the simulation parameters. These landmarks are at the centre of squares with the same length of 480 pixels, so the speed of the UAV could be reduced as the distance passed by the UAV in each step. Intuitively, the unit of correction for angles in degrees, not radians. Image (a) in Figure 4.7 shows the result of simulation with a larger speed of 120 pixels and a smaller rotation correction of 20 degrees in each step. It can be seen that the UAV takes many detours. Typically, from landmark 1 – starfish to landmark 2 – tortoise, the UAV flies over the tortoise three times, and finally flies into the threshold range of landmark 2, and then the objective landmark is set to landmark 3 – crab. Similarly, the UAV also takes a big detour to reach the landmark 4 – octopus, as it can only adjust its orientation at 20 degrees each time. Image (b) in Figure 4.7 displays the result of UAV simulation at speed 60 pixels and rotation speed 40 degrees. It shows that if the speed of the UAV is relatively small and the rotation correction angle is relatively big, the trajectory of the UAV is nearly a straight line. Even though the route looks well, but it is impractical because the drone cannot change the orientation so quickly during moving forward. Image (c) displays a preferable realistic trajectory with a speed of 100 pixels and rotation step 30 degrees. The trajectory is smooth with curves, which successfully simulates the inertial of the UAV and the detour is reasonable: if the orientation of the UAV has a larger difference in comparison to the orientation of the objective landmark, the UAV would turn more times. Therefore, in the following experiments, the below parameter couple are used: speed is 100 pixels, and rotation correction is 30 degrees.

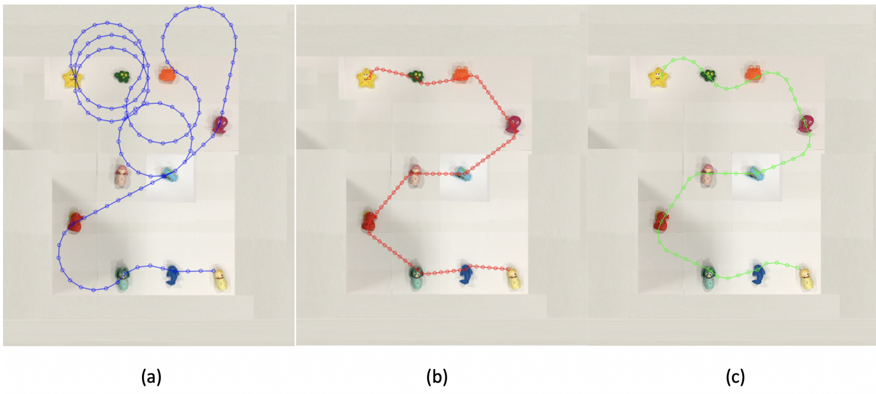


Figure 4.7: The blue line in Image (a) is the simulation trajectory of UAV with speed 120 pixels and correction orientation step 20 degrees; the red line in Image (b) displays the simulation trajectory with speed 60 pixels and the correction orientation step 40 degrees; Image (c) shows that the simulation result with speed 100 pixels and the correction angle step 30 degrees in green. These routes are the connection of adjacent locations and each small circle represents the projective locations of UAV at each step.

Moreover, the above experiments demonstrate that our Algorithm 3.1 is feasible, and our regulations of navigation simulation assumptions are reasonable: UAV could approach each landmark with the ordered sequence. And its orientation and speed changing step obey the rules prescribed previously. The following experiment can be carried on based on this experiment.

4.2.2 Comparison of image matching algorithms

In this experiment, the comparison results are acquired separately by the fast matching algorithm and Ao's method. Before applying the orientation and the location information to navigate UAV, it needs to select out suitable image matching algorithms to acquire both from numerous computer vision technologies. In comparison with similar research, Ao's method is the only recent approach to investigating a similar problem. Image (a) in Figure 4.8 demonstrates the ground-truth path generated by assuming the landmarks matching is perfect which is the same as Image (c) in Figure 4.7. It can be seen that the green line is smooth and perfectly connects each landmark. The green circles represent each location

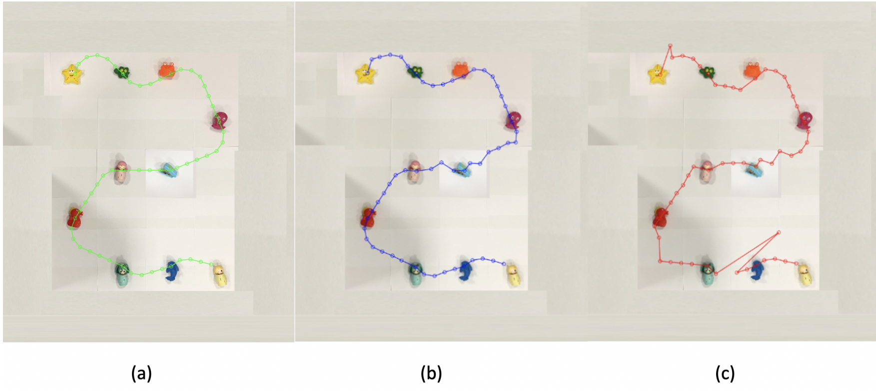


Figure 4.8: Image (a) displays the ideal path simulated by assuming perfect matching; Image (b) shows the trajectory generated by applying the fast matching method; the trajectory generated by Ao's method is displayed in Image (c). The blue trajectory is better without deviations compared to the red trajectory.

of the UAV. And at each green spot, a rectangular image is taken whose orientation is determined by the current spot and next spot. Therefore, a sequence of images is acquired to simulate the UAV aerial images taken along with this ideal path. The blue line on Image (b) is the route formed by connecting each calculated projective place by means of the fast matching algorithm to match with the above simulated aerial images. Compared with the ideal path, each computed projective place is relatively close to the spot where the aerial image is taken. In the contrast, the red line on Image (c) is obtained by the connection of each calculated projective place with Ao's method. The shape of the red trajectory is much worse than (b) based on the ground truth (a), the location deviation is larger, and it has a larger jump due to the matching error between the landmarks 1, 2 and landmarks 8, 9.

Table 4.3: The Euclidean distance difference of real location and estimated position with two methods.

$Landmark_i$	1	2	3	4	5	6	7	8	9	10	Average
$Dist(b, a)$	7.39	3.65	13.57	7.45	39.32	13.43	6.04	11.02	7.91	7.74	14.25
$Dist(c, a)$	9.03	3.57	24.41	6.61	67.85	14.39	4.93	30.23	7.14	10.07	24.41

The detailed comparison by using two image matching algorithms is summarised in Table 4.3, and the Euclidean distance is utilized to calculate the deviations. Table 4.3 shows the errors between these trajectory spots. Typically, here this table only displays the comparison result on 10 sample spots that are corresponding to 10 landmarks, as shown from 1 to 10 in the first row. The second row displays the distance between the ground truth location and the computed place obtained by the fast matching algorithm. The third row shows the distance between the ground truth location and the calculated place obtained by Ao's method. The average column is the average of all the spots' location distance differences along the simulation trajectory. In Table 4.3, it can be seen that at landmarks 3, 5, and 8, there are larger deviations computed by Ao's method. The speed of the ideal simulation is 100 pixels, so the overall error of our method is about fourteen percent deviation. In contrast, a twenty-four percent average deviation is generated by Ao's method. Therefore, our method is preferable in the following navigation simulation experiment.

4.2.3 Simulation based on image matching algorithms

In this experiment, real UAV navigation based on image matching algorithms are simulated and the Algorithm 3.2 will be verified. The parameters are the same as the settings in experiment 1) – the speed of the UAV is 100 pixels each time and the correction angle step is 30 degrees. Similarly, in the matching procedure, Ao's method is utilized to get another comparison trajectory and the ideal path is seen as the ground truth of the reference route displayed in Image (a) in Figure 4.9. Image (b) in Figure 4.9 shows the simulation trajectory of UAV navigation adopted by the fast matching algorithm. The blue circles represent the sequence of each matched and calculated projective location of the UAV. Compared with Image (c), the blue trajectory is relatively better for there is no big jump. The UAV could navigate itself from the starting landmark to the final landmark, in the meanwhile, it can adjust its orientation with the stable angle correction step. The path in Image (b) presents that "this drone flies smoothly." And the reason

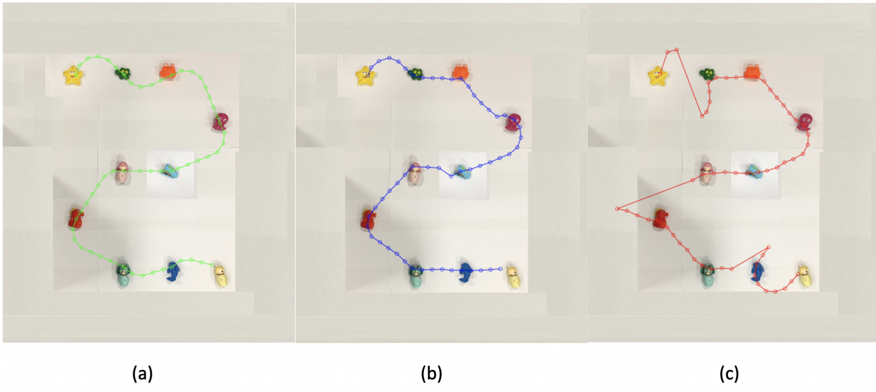


Figure 4.9: Image (a) is the ideal navigation path marked with a green line which is the same as previous experiments; Image (b) is the navigation route generated by the fast matching algorithm and the colour is marked with blue; Image (c) displays the simulation result obtained by Ao's method, the UAV arrives at the final landmark, however, the comparatively large error may lead navigation failure in reality.

is that the fast matching algorithm is suitable, and the calculated location is accurate.

Image (c) in Figure 4.9 displays the navigation result obtained by Ao's method. The red circles denote the computed locations after matching the UAV aerial image with the map. In Algorithm 3.1, the matching method is used to calculate the position of the UAV and by assuming that the UAV locates on the calculated positions, so there are jumps in the simulation trajectory. The jumps occur on Image (c) between landmarks 1, 2, landmarks 6, 7, and landmarks 8, 9. The reason is that it is indispensable to have two couples of interest points, in order to calculate the orientation as the previous illustration. In this experiment, two pairs of interest points are selected randomly, so this may lead to the location computation error. But this problem is built into the matching algorithm itself – bad matching may also be regarded as good matching. Such trouble is not discussed in this thesis. If the number of matching pairs is less than two, the orientation is set to 90 degrees for convenience in this thesis. In this experiment, our navigation rules are effective. The red trajectory in Image (c) shows that the drone could navigate to the final landmark successfully. This simulation result is

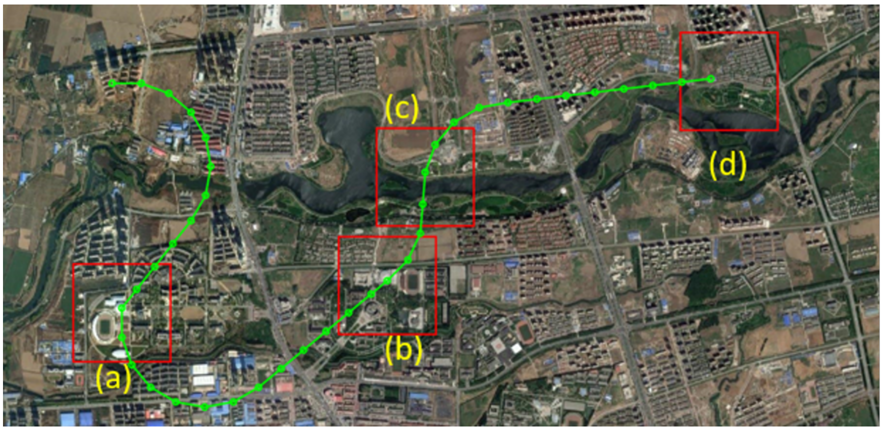


Figure 4.10: Simulation result with real Google map. The order of landmarks is as shown from the region (a) to region (b) and (c) and finally stop at region (d). The UAV starts flying from the top left corner with zero degrees to the horizontal line as default orientation.

achieved for these features of landmarks are prominent, in reality, image features may not be salient, and so the navigation could fail by using Ao's method.

4.2.4 Navigation simulation based on aerial images

Recently, the regulations targeted on UAVs are becoming more and more rigorous, and most regions of a city are forbidden to fly UAVs. So it is not realistic to test the civil drones' navigation on a larger and continuous region in a city. Moreover, the duration time of a civil UAV is comparatively short which makes it impossible for testing the navigation in a long and complete route. Therefore, in this experiment, the following procedure is designed to simulate the navigation of UAVs in the real environment.

Firstly, four regions are selected in the real city as landmarks (a), (b), (c), and (d) as shown in Figure 4.10. Secondly, the UAV is controlled to fly up to about 500 meters, then take some aerial images with different orientations. Thirdly, the matching result between aerial images and the satellite map is computed for the real orientation and location of the UAV. Fourthly, the navigation trajectory among landmarks is simulated according to the previous algorithm. Finally, the complete trajectory of UAV is formed by these simulated trajectories' connection.

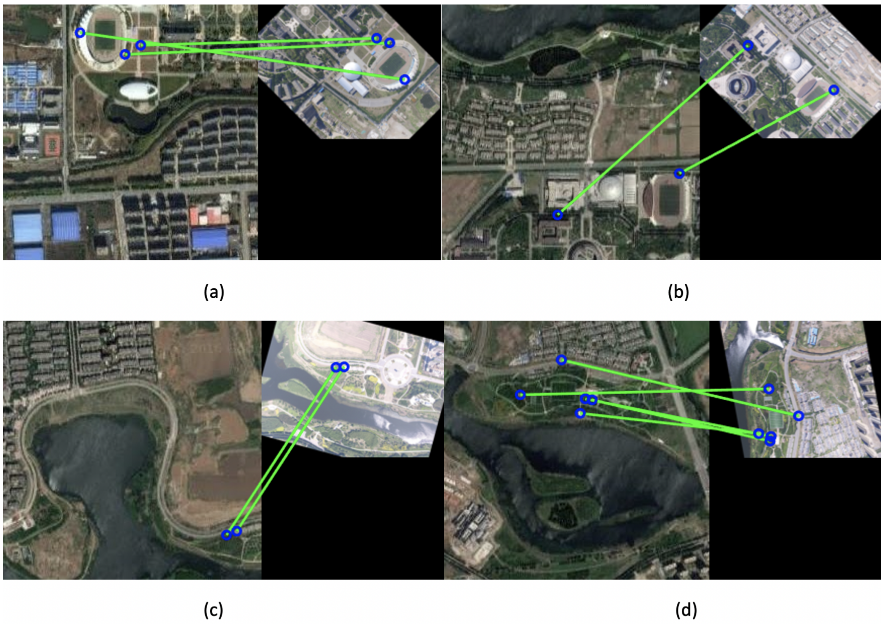


Figure 4.11: Image (a) (b) (c) and (d) display the matching results on four calibration regions (a) (b) (c) and (d) marked with red rectangular in Figure 4.10.

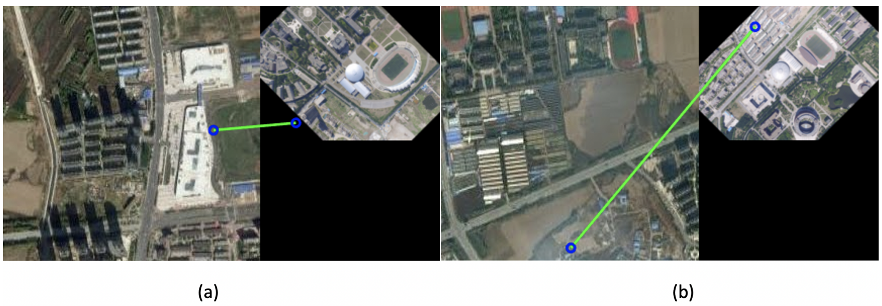


Figure 4.12: Image (a) displays the bad matching in Ao's method, and the correct matching is Image (a) in Figure 4.11 with fast matching algorithm; Image (b) displays the wrong matching in Ao's method, and the right matching is Image (b) in Figure 4.11 acquired by the fast-matching algorithm.

The regions where the real aerial images are taken can be seen as the calibration regions based on the above rules, which not only provide the calibration information as the ending for the previous simulated trajectory part but also offer the orientation and location as the starting for the following simulation trajectory part. The result of such navigation simulation is displayed in Figure 4.10. It can be seen that the proposed algorithm can smoothly navigate the UAV flying through each landmark, the whole trajectory is of high quality and the UAV acquires the right orientation at each region by applying the fast matching algorithm for images with large scale disparity.

Because only having four aerial images, this simulation can be called half-simulation – the real aerial images function as calibration images for the whole navigation trajectory. Nevertheless, the trajectory shown in Figure 4.10 illustrates that if the quantity of "calibration regions" is increased this simulated trajectory would approach the real navigation route. And this is the core meaning of navigation. The matching results at each landmark are displayed in Figure 4.11. It can be seen each landmark is successfully matched, and thus provides the correct location and orientation information. The Google satellite and four UAV aerial images are provided from Ao's paper. For comparison, the matching utilized by Ao's method on the four aerial images is also carried on. Image (a) and (b) obtain the wrong matching result and the other two images don't match any region in the satellite map displayed in Figure 4.12. Ao's method cannot find the right matching in the calibration regions, so the simulation fails. It is impossible to draw its trajectory on the satellite map.

4.3 Summary

In this chapter, we conducted two groups of experiments. The first group tests the fast matching algorithm which accelerates the matching speed and conserves the matching accuracy. At the same time, Ao's method which is the only recent algorithm coping with similar tasks is carried on in the same condition as a comparison to make these experimental results more convincing. The second group applies the proposed matching algorithm to the autonomous navigation of drones. The fast matching algorithm could compute the location and orientation of UAV by matching the aerial image with the satellite map which is the core element in “navigation”. Therefore, this experiment provides a new possible method for UAV navigation using only computer vision technology.

Chapter 5

Conclusions

Advancements in sensing systems and storage technology have created lots of high-resolution images with abundant information, which provides a steady flow of materials for processing and analysing by means of computer vision knowledge. Similarly, the developments of computer technology and chip technology have extended computer vision applications to a variety of aspects. In the meanwhile, more and more UAVs appear in various environments. Therefore, there is an increasing demand to combine UAVs with computer vision which could reduce the weight of UAVs and make them more intelligent to cope with more complex tasks without human intervention.

The image matching algorithms of computer vision in the past, either focus on the matching accuracy or pay attention to the computation overhead. In order to resolve one computer vision task, normally, the combination of several algorithms is required, and many methods need to be modified in advance then be utilized in specific problems. There are two challenges in applying computer vision to UAV navigation. One is the huge scale gap between an aerial image taken by drone and an image or map taken by satellite, which often leads to matching failure. The other one is computation efficiency as the UAV needs to adjust its orientation as soon as possible to avoid unnecessary detours and to conserve its energy.

Therefore, the fast matching algorithm for images with large scale disparity is proposed in this thesis, which could compute the orientation and location of UAV with a much faster speed. Furthermore, this method is applied in relatively simplified environment settings to verify its navigation performance in simulation mode and achieves expected results.

5.1 Fast matching algorithm

This thesis proposes a fast matching algorithm, which can be applied to the images with huge scale difference. In order to bridge the huge scale gap, the proposed method firstly introduces a scaling factor to downgrade UAV aerial image and at the same time to enhance the satellite map quality. Secondly, the proposed algorithm gives matching criteria used for the searching matching region on the satellite map for the vision of satellite is much wider which causes the satellite map is much spacious. According to the above two improvements, the fast matching method acquires efficiency on speed and accuracy on matching. Furthermore, the orientation evaluation and the location computation methods are presented, which makes this method can be applicable to navigation applications.

5.2 Autonomous navigation of UAV

Before applying the fast matching algorithm in the autonomous navigation simulation, at first, we conceive a set of rules and the simulation environment. Next, we design two algorithms: the first one is to verify the feasibility of the presumed navigation rules and the second one is to introduce the fast matching method to acquire the real-time orientation and location of UAV when it flies over each landmark region and arrives at final place. The simulation result is acceptable: by continuously matching, the UAV could navigate itself through these landmarks with priority order – the fast matching algorithm successfully provides the location and direction information. At last, we simulate navigation based on the real aerial and satellite images and achieve acceptable results. The navigation simulation of UAV conversely demonstrates the performance of the fast matching algorithm for image matching with large scale disparity.

5.3 Future work

The proposed algorithm could be further optimized in a specific task or a particular environment. For example, according to different topography, we could pre-process the aerial images to extract different and more salient features to increase the matching performance. Besides for a fixed satellite map which is used as the reference, we could store its features in a database. Therefore, the procedure of searching could be combined with other indexing methods and the matching efficiency is improved. Recently the artificial intelligence is hot, the ideas proposed in this thesis could integrate A.I. to enhance the capability and make the navigation acquire higher intelligence and resolve harder tasks.

Appendices

Appendix A

Attributions

1. **Chen, S.**, Wang, Z., and Ren, Y. (2020). A fast matching algorithm for the images with large scale disparity. *Mathematical Foundations of Computing*, 3(3), 141-155.

Name of Co-author	Acquisition of Data and Method	Data Conditioning and Manipulation	Analysis and Statistical Method	Interpretation and Discussion	Final Approval	Signature
Shichu Chen	✓	✓	✓	✓		
Zhiqiang Wang				✓	✓	
Yan Ren				✓	✓	

2. **Chen, S.**, Li, Q., and Ren, Y. (2021). A Novel UAV Autonomous Navigation Approach based on Fast Matching Algorithm for Images with Large Scale Disparity. *International Conference on Service Robotics Technologies.* (accepted)

Name of Co-author	Acquisition of Data and Method	Data Conditioning and Manipulation	Analysis and Statistical Method	Interpretation and Discussion	Final Approval	Signature
Shichu Chen	✓	✓	✓	✓		
Qilin Li			✓	✓	✓	
Yan Ren				✓	✓	

Bibliography

[1] P. J. Besl, Geometric modeling and computer vision, Proceedings of the IEEE 76 (8) (1988) 936–958.

[2] S. Yoshida, Computer vision, 2011.

[3] P. O. Haikonen, R. Brains, Circuits and systems for conscious machines, Wiley Online Library, 2007.

[4] P. R. Smart, Human-extended machine cognition, Cognitive Systems Research 49 (2018) 9–23.

[5] D. Vernon, Machine vision-automated visual inspection and robot vision, STIA 92 (1991) 40499.

[6] K. Chowdhary, Fundamentals of Artificial Intelligence, Springer, 2020.

[7] T. Huang, Computer vision: Evolution and promise.

[8] H. G. Barrow, J. M. Tenenbaum, Computational vision, Proceedings of the IEEE 69 (5) (1981) 572–595.

[9] L. Shapiro, G. Stockman, Computer vision prentice hall, Inc., New Jersey.

[10] K. H. Tan, B. P. Lim, The artificial intelligence renaissance: deep learning and the road to human-level machine intelligence, APSIPA Transactions on Signal and Information Processing 7.

[11] N. Rusk, Deep learning, Nature Methods 13 (1) (2016) 35–35.

[12] L. Deng, D. Yu, Deep learning: methods and applications, Foundations and trends in signal processing 7 (3–4) (2014) 197–387.

[13] G. Bradski, A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library, " O'Reilly Media, Inc.", 2008.

[14] L. C. Yan, B. Yoshua, H. Geoffrey, Deep learning, nature 521 (7553) (2015) 436–444.

[15] M. Jogin, M. Madhulika, G. Divya, R. Meghana, S. Apoorva, et al., Feature extraction using convolution neural networks (cnn) and deep learning, in: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, 2018, pp. 2319–2323.

[16] Y. H. Tseng, S. S. Jan, Combination of computer vision detection and segmentation for autonomous driving, in: 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), 2018, pp. 1047–1052.

[17] K. Amer, M. Samy, M. Shaker, M. ElHelw, Deep convolutional neural network based autonomous drone navigation, in: Thirteenth International Conference on Machine Vision, Vol. 11605, International Society for Optics and Photonics, 2021, p. 1160503.

[18] L. Klingbeil, M. Nieuwenhuisen, J. Schneider, C. Eling, D. Droeschel, D. Holz, T. Läbe, W. Förstner, S. Behnke, H. Kuhlmann, Towards autonomous navigation of an uav-based mobile mapping system, in: Proc. of Int. Conf. on Machine Control & Guidance (MCG), 2014.

[19] G. Balamurugan, J. Valarmathi, V. Naidu, Survey on uav navigation in gps denied environments, in: 2016 International conference on signal processing, communication, power and embedded system (SCOPES), IEEE, 2016, pp. 198–204.

[20] F. Vanegas, K. J. Gaston, J. Roberts, F. Gonzalez, A framework for uav navigation and exploration in gps-denied environments, in: 2019 ieee aerospace conference, IEEE, 2019, pp. 1–6.

[21] H. X. Pham, H. M. La, D. Feil-Seifer, L. V. Nguyen, Autonomous uav navigation using reinforcement learning, arXiv preprint arXiv:1801.05086.

[22] Y. Lu, Z. Xue, G.-S. Xia, L. Zhang, A survey on vision-based uav navigation, Geo-spatial information science 21 (1) (2018) 21–32.

[23] T. Morris, Computer vision and image processing, Palgrave Macmillan, 2004.

[24] N. Sebe, I. Cohen, A. Garg, T. S. Huang, Machine learning in computer vision, Vol. 29, Springer Science & Business Media, 2005.

[25] M. F. Bulbul, F. Badsha, R. Islam, Object detection by point feature matching using matlab, European Journal of Applied Sciences 6 (6) (2018) 22–22.

[26] D. Renza, D. M. Ballesteros, H. D. Ortiz, Text hiding in images based on qim and ovsf, IEEE Latin America Transactions 14 (3) (2016) 1206–1212.

[27] B. Potetz, T. S. Lee, Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes, JOSA A 20 (7) (2003) 1292–1303.

[28] W. J. MacLean, J. K. Tsotsos, Fast pattern recognition using normalized grey-scale correlation in a pyramid image representation, Machine Vision and Applications 19 (3) (2008) 163–179.

[29] B. S. Reddy, B. N. Chatterji, An fft-based technique for translation, rotation, and scale-invariant image registration, IEEE transactions on image processing 5 (8) (1996) 1266–1271.

[30] W. Lu, P. J. Parikh, J. P. Hubenschmidt, J. D. Bradley, D. A. Low, A comparison between amplitude sorting and phase-angle sorting using external respiratory measurement for 4d ct, Medical physics 33 (8) (2006) 2964–2974.

[31] C. May, The Fourier Series and Fourier Transform, 2020.

[32] H. Herrmann, H. Bucksch, Wörterbuch GeoTechnik/Dictionary Geotechnical Engineering: Deutsch–Englisch/German–English, Springer-Verlag, 2014.

[33] C. Arangala, Exploring Linear Algebra: Labs and Projects with MATLAB®, CRC Press, 2019.

[34] A. S. Eltanany, M. S. Elwan, A. Amein, Key point detection techniques, in: International Conference on Advanced Intelligent Systems and Informatics, Springer, 2019, pp. 901–911.

[35] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[36] B. Yang, Z. Wang, X. Yu, Extraction of orb features with an improved method, in: Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, 2019, pp. 1–6.

[37] Y. Liu, H. Zhang, H. Guo, N. N. Xiong, A fast-brisk feature detector with depth information, Sensors 18 (11) (2018) 3908.

[38] J. Zhang, W. Chang, L. Wu, Edge detection based on general grey correlation and log operator, in: 2010 International Conference on Artificial Intelligence and Computational Intelligence, Vol. 1, IEEE, 2010, pp. 480–483.

[39] Y. Zhang, X. Han, H. Zhang, L. Zhao, Edge detection algorithm of image fusion based on improved sobel operator, in: 2017 IEEE 3rd Infor-

mation Technology and Mechatronics Engineering Conference (ITOEC), IEEE, 2017, pp. 457–461.

[40] R. Tautenhahn, C. Boettcher, S. Neumann, Highly sensitive feature detection for high resolution lc/ms, BMC bioinformatics 9 (1) (2008) 504.

[41] S. Siggelkow, H. Burkhardt, Improvement of histogram-based image retrieval and classification, in: Object recognition supported by user interaction for service robots, Vol. 3, IEEE, 2002, pp. 367–370.

[42] J. Alamghtuf, F. Khelifi, Self geometric relationship-based matching for palmprint identification using sift, in: 2017 5th International Workshop on Biometrics and Forensics (IWBF), IEEE, 2017, pp. 1–5.

[43] J. Zhang, Q. Du, L. Xiongfei, R. Te, An effective image matching method, International Journal of Digital Content Technology and its Applications 7 (2013) 583–590.

[44] J. H. Suk, S. Yoon, L. Chun-Gi, I. J. Chun, T. M. Roh, Image matching devices and image matching methods thereof, uS Patent 8,693,785 (Apr. 8 2014).

[45] R. Khemiri, H. Kibeya, F. E. Sayadi, N. Bahri, M. Atri, N. Masmoudi, Optimisation of hevc motion estimation exploiting sad and ssd gpu-based implementation, IET Image Processing 12 (2) (2017) 243–253.

[46] S. Korman, R. Litman, Latent ransac, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6693–6702.

[47] S. Alam, M. Kang, J. Y. Pyun, G. R. Kwon, Performance of classification based on pca, linear svm, and multi-kernel svm, in: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2016, pp. 987–989.

[48] J. Guo, X. Wang, Image classification based on surf and knn, in: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), IEEE, 2019, pp. 356–359.

[49] A. Gruen, Development and status of image matching in photogrammetry, The Photogrammetric Record 27 (137) (2012) 36–57.

[50] N. Jayanthi, S. Indu, Comparison of image matching techniques, International journal of latest trends in engineering and technology 7 (3) (2016) 396–401.

[51] X. Li, J. Luo, C. Duan, P. Yin, Y. Zhi, Portrait matching based on sift features, in: IOP Conference Series: Materials Science and Engineering, Vol. 612, IOP Publishing, 2019, p. 032162.

[52] K. Sugamya, P. Suresh, A. V. Babu, R. Akhila, Cbir using sift with log, dog and pca, in: Data Engineering and Communication Technology, Springer, 2020, pp. 623–637.

[53] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: European conference on computer vision, Springer, 2006, pp. 404–417.

[54] L. Tong, X. Ying, 3d point cloud initial registration using surface curvature and surf matching, 3D Research 9 (3) (2018) 41.

[55] X. Wang, J. Zou, D. Shi, An improved orb image feature matching algorithm based on surf, in: 2018 3rd International Conference on Robotics and Automation Engineering (ICRAE), IEEE, 2018, pp. 218–222.

[56] J. Zhan, B. Hu, Salient object contour detection based on boundary similar region, in: 2012 Fourth International Conference on Digital Home, IEEE, 2012, pp. 335–339.

[57] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, V. K. Asari, The history began

from alexnet: A comprehensive survey on deep learning approaches, arXiv preprint arXiv:1803.01164.

[58] N. A. Muhammad, A. Nasir, Z. Ibrahim, N. Sabri, Evaluation of cnn, alexnet and googlenet for fruit recognition, Indonesian Journal of Electrical Engineering and Computer Science 12 (2) (2018) 468–475.

[59] J. E. Espinosa, S. A. Velastin, J. W. Branch, Vehicle detection using alex net and faster r-cnn deep learning models: a comparative study, in: International Visual Informatics Conference, Springer, 2017, pp. 3–15.

[60] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, Deep learning vs. traditional computer vision, in: Science and Information Conference, Springer, 2019, pp. 128–144.

[61] S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4353–4361.

[62] L. Zheng, Y. Yang, Q. Tian, Sift meets cnn: A decade survey of instance retrieval, IEEE transactions on pattern analysis and machine intelligence 40 (5) (2017) 1224–1244.

[63] A. M. Elmoogy, X. Dong, T. Lu, R. Westendorp, K. R. Tarimala, Surfcnn: A descriptor accelerated convolutional neural network for image-based indoor localization, IEEE Access 8 (2020) 59750–59759.

[64] Q. Zhao, B. Zhang, S. Lyu, H. Zhang, D. Sun, G. Li, W. Feng, A cnn-sift hybrid pedestrian navigation method based on first-person vision, Remote Sensing 10 (8) (2018) 1229.

[65] T. Serre, Deep learning: the good, the bad, and the ugly, Annual review of vision science 5 (2019) 399–426.

[66] T. Ao, X. Liu, Y. Ren, R. Luo, J. Xi, An approach to scene matching algorithm for uav autonomous navigation, in: 2018 Chinese Control And Decision Conference (CCDC), IEEE, 2018, pp. 996–1001.

[67] L. Zhu, Y. Wang, B. Zhao, X. Zhang, A fast image stitching algorithm based on improved surf, in: 2014 Tenth International Conference on Computational Intelligence and Security, IEEE, 2014, pp. 171–175.

[68] W. An, R. Yu, Y. Wu, An image registration algorithm based on fast, ransac and surf, in: Information Technology and Applications: Proceedings of the 2014 International Conference on Information technology and Applications (ITA 2014), Xian, China, 8-9 August 2014, CRC Press, 2015, p. 329.

[69] X. Li, X. Wang, C. Cheng, Application of scene recognition technology based on fast-er and surf algorithm in augmented reality.

[70] J. Hosang, R. Benenson, B. Schiele, Learning non-maximum suppression, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4507–4515.

[71] D. Xiang, B. Zhong, K. K. Ma, A location-aware scale-space method for salient object detection, in: 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 4195–4199.

[72] D. Zeng, L. Wu, B. Chen, W. Shen, Slope-restricted multi-scale feature matching for geostationary satellite remote sensing images, Remote Sensing 9 (6) (2017) 576.

[73] C. H. Tsai, Y. C. Lin, An accelerated image matching technique for uav orthoimage registration, ISPRS Journal of Photogrammetry and Remote Sensing 128 (2017) 130–145.

[74] S. Chen, Z. Wang, Y. Ren, A fast matching algorithm for the images with

large scale disparity, Mathematical Foundations of Computing 3 (3) (2020) 141.

[75] K. Nonami, Present state and future prospect of autonomous control technology for industrial drones, IEEJ Transactions on Electrical and Electronic Engineering 15 (1) (2020) 6–11.

[76] C. Chen, Y. Tian, L. Lin, S. Chen, H. Li, Y. Wang, K. Su, Obtaining world coordinate information of uav in gnss denied environments, Sensors 20 (8) (2020) 2241.

[77] R. Clarke, Understanding the drone epidemic, Computer Law & Security Review 30 (3) (2014) 230–246.

[78] J. D. Blom, Unmanned Aerial Systems: a historical perspective, Vol. 45, Citeseer, 2010.

[79] J. Zhang, Z. Guo, L. Wu, Research on control scheme of vertical take-off and landing fixed-wing uav, in: 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), IEEE, 2017, pp. 200–204.

[80] J. Dai, H. Nie, J. Ying, Y. Zhao, Y. Sun, Modeling and tracking control of unmanned helicopter, in: 2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE), IEEE, 2020, pp. 149–155.

[81] J. Zhao, J. Li, C. Liu, B. Li, Analysis for initial status parameters of parachute aircraft, in: Applied Mechanics and Materials, Vol. 496, Trans Tech Publ, 2014, pp. 736–743.

[82] X. Xue, Y. Lan, Z. Sun, C. Chang, W. C. Hoffmann, Develop an unmanned aerial vehicle based automatic aerial spraying system, Computers and electronics in agriculture 128 (2016) 58–66.

[83] O. Yakimenko, T. Jann, Guidance, navigation, and control.

[84] M. Crisci, W. Geri, G. Bertoni, Deis uav: Gnc design and avionics implementation issues, IFAC Proceedings Volumes 37 (6) (2004) 747–752.

[85] C. K. Sung, F. Segor, Adaptive pattern for autonomous uav guidance, in: Applications of Digital Image Processing XXXVI, Vol. 8856, International Society for Optics and Photonics, 2013, p. 88560P.

[86] S. Shaukat, U. Bhatti, Updraft modeling for uav guidance, Ph.D. thesis (2016).

[87] M. Macomber, world geodetic system 1984.

[88] O. Clarisse, K. Chang, Shi, Vicon, 1987.

[89] P. Das, M. Ghosh, A novel design of microstrip monopole antenna for uwb applications, in: 2020 4th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), IEEE, 2020, pp. 1–4.

[90] K. P. Valavanis, G. J. Vachtsevanos, Handbook of unmanned aerial vehicles, Vol. 1, Springer, 2015.

[91] B. Sanders, D. Vincenzi, Y. Shen, Investigation of gesture based uav control, in: International Conference on Applied Human Factors and Ergonomics, Springer, 2017, pp. 205–215.

[92] M. Azkarate, L. Gerdes, L. Joudrier, C. J. Pérez-del Pulgar, A gnc architecture for planetary rovers with autonomous navigation, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 3003–3009.

[93] A. Cadena, R. Ponguillo, D. Ochoa, Development of guidance, navigation and control system using fpga technology for an uav tricopter, in: Mechatronics and Robotics Engineering for Advanced and Intelligent Manufacturing, Springer, 2017, pp. 363–375.

[94] C. D. Wickens, M. Vincow, M. Yeh, Design Applications of Visual Spatial Thinking: The Importance of Frame of Reference., Cambridge University Press, 2005.

[95] B. Nelson, Understanding an inertial platform, NAVIGATION, Journal of the Institute of Navigation 12 (2) (1965) 145–152.

[96] I. Sharp, K. Yu, Wireless Positioning: Principles and Practice, Springer, 2019.

[97] O. Frey, Synthetic aperture radar, International Encyclopedia of Geography: People, the Earth, Environment and Technology: People, the Earth, Environment and Technology (2016) 1–15.

[98] C. Zhang, J. Chen, C. Song, J. Xu, An uav navigation aided with computer vision, in: The 26th Chinese Control and Decision Conference (2014 CCDC), IEEE, 2014, pp. 5297–5301.

[99] J. Sasiadek, M. Walker, Vision-based uav navigation, in: AIAA Guidance, Navigation and Control Conference and Exhibit, 2008, p. 6667.

[100] K. Tahar, S. Kamarudin, Uav onboard gps in positioning determination., International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 41.

[101] D. Giordan, M. S. Adams, I. Aicardi, M. Alicandro, P. Allasia, M. Baldo, P. De Berardinis, D. Dominici, D. Godone, P. Hobbs, et al., The use of unmanned aerial vehicles (uavs) for engineering geology applications, Bulletin of Engineering Geology and the Environment (2020) 1–45.

[102] K. Nonami, Present state and future prospect of autonomous control technology for industrial drones, IEEJ Transactions on Electrical and Electronic Engineering 15 (1) (2020) 6–11.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.