School of Computing
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

A bio-inspired image segmentation method based on saccades

Jiaqi Liu

Submitted in accordance with the requirements for the degree of
<MSc Advanced Computer Science>

<2018/2019>

The candidate confirms that the following have been submitted*:*

<As an example>

Items	**Format**	**Recipient(s) and Date**
Deliverables 1	*Report*	*SSO (04/09/19)*
Deliverable 2	*Software codes or URL* https://github.com/fragileASH/Msc-project	*Supervisor, assessor (04/09/19)*

Type of Project: Empirical Investigation

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

Summary

This work is inspired by the fact that creatures rely on eye movements to obtain additional information and assist in the preprocessing of visual information. The project proposes an edge-based image segmentation method especially suitable for noise environment. This method USES simulated eye movements to augment the data, turning an image into a noisy video. Temporary gradients are obtained by recording intensity changes in each frame of video. Then these temporary gradients are transformed into frequency domain signals by discrete Fourier transform. The correlation coefficients of frequency domain signal and reference edge are calculated for each point. Then filter out the noise point and background point through the threshold value to get the edge graph. The segmentation results are obtained by performing morphological closure operation on the edge graph.

Under the condition of noise, the average intersection over union (IOU) score of this algorithm on the image of the data set is slightly larger than the segmentation based against the canny edge detector. On simple objects, if the parameters are properly configured, the score is about 10 times that of the latter.

The algorithm also has some drawbacks, such as insensitivity to complex materials and the need to manually set parameters.

I hope these limitations can be solved in future researches, and this work can bring inspiration and motivation to subsequent researchers.

Acknowledgements

Many thanks to my two supervisors, Anthony Cohn and Mohamed Abdellatif. They are talented and patient. Anthony is a busy man, but he is energetic in every meeting. He always points out the key points and takes details seriously. He also has a sense of humour that makes me feel relaxed. Mohamed is very approachable and provides the idea of this project. Whenever I feel worried, he always encourages me. He told me not to be afraid of failure, which is an important part of science. I feel lucky that I have two such supervisors.

A special thanks to my girlfriend, Yingxi Yi

I had a great time with her and it took a lot of pressure off me.

Without her, I would have finished it a long time ago.

Table of Contents

Summary ..... **iii**
Acknowledgements ..... **iv**
Table of Contents ..... **v**
Chapter 1 INTRODUCTION **1**
1.1 Problem definition 1
1.2 Objective of the project 2
1.3 Deliverables 2
1.4 Timeline and project management 3
1.5 Structure of the report 3
Chapter 2 Literature review and related works **5**
2.1 Eye movement research in the biological field 5
2.1.1 The introduction of saccade 5
2.1.2 The features of saccade 6
2.1.3 The benefit of saccades 9
2.2 Image noise 11
2.2.1 Introduction of image noise 11
2.2.2 Noise removal 12
2.3 Edge detection 13
2.4 Image segmentation 19
2.5 Related works 21
Chapter 3 Methodology **25**
3.1 Introduction 25
3.2 Data generation 26
3.3 Data extraction 29
3.4 Data processing 32
3.5 Edge detection 36
3.6 Segmentation 36
Chapter 4 Experiments **38**
4.1 Experiment environment 38
4.2 Data generation 39
4.3 Edge detection 41
4.4 Segmentation 42

Chapter 5 Results and Evaluation 44
5.1 The metrics 44
5.2 Results 44
5.3 Testing and evaluation 46
Chapter 6 Conclusions 51
6.1 Reviews 51
6.2 Limitation 51
6.3 Future work 51
6.3 self-reflection 52
List of References 54
Appendix A External Materials 59
Appendix B Ethical Issues Addressed 60

Chapter 1
INTRODUCTION

1.1 Problem definition

Before humans began to conduct large-scale, systematic scientific research, many people began to learn from and be inspired by nature and the biological world. Through the understanding and imitation of biological characteristics, we have obtained many promising research results and products that can be used to improve human life. The work to be described in this report is also based on biological inspiration, specifically, new ideas and improvements inspired by biological eye movements for image noise suppression, edge detection and image segmentation in computer vision.

Biological vision is a complex, sophisticated system that has not yet been fully understood. Part of what we know is that organic eyes don't stop, they move. The motion pattern of eye movement has been measured in few decades ago. In previous research, (Young and Sheena, 1975) identified the categories and pattern of different eye movement based on the features of motion. They found that some kinds of eye movement take fundamental responsibility for object recognition and information filtration. (Liversedge and Findlay, 2000) found that a movement pattern, called saccade, isn't a useless piece of redundancy, but one that can help creatures quickly sift through information and enhance their visual cognition.

Similarly, in image analysis, the application of saccadic motion on the camera head at the image collection end can provide us with more information. This information is not spatial but temporal. A reasonable assumption is that by analysing and utilizing this temporary information, we can achieve better results on some computer vision tasks.

Part of this discussion of related work focuses on the recognition of the saccade motion or shaking pattern of the camera. Most of these methods can be regarded as a time series analysis problem. The identified motion frequency, amplitude and other information can be used to describe a unique motion pattern. It is very natural to use machine learning to carry out such a task, because the information of motion patterns is easy to be discretized vectorized. In (Soran et al., 2012) ‘s work, they used support vector machines to classify motion patterns, achieving a high accuracy rate.

After understanding the overall movement pattern of the image, that is, the simulated eye movement pattern, we can get the information of the pattern of saccade. This information can be used for a variety of computer vision tasks. Tasks such as multi-target tracking and activity detection use mathematical models to represent the motion (Giefing et al., 1992) .

In this article, the focus is on high frequency, low amplitude eye movements. We believe that such movements can bring additional information about the intensity gradient of the image, known as the temporal gradient(Kratz and Nishino, 2009). By analysing such information, noise removal and edge detection can be achieved simultaneously, and better segmentation results based on edge detection can be obtained on noisy images.

When we get a series of images that move in a particular pattern, we can track the change of each pixel in the image. These differences can provide enough information to determine whether it is an edge point or a background point or a point inside an object. There will be no significant change in the intensity of the points inside the object or in the background. But edge points will have a relatively noticeable change in pixel intensity. Moreover, the changes will fit a pattern. If a change occurs at a point that does not conform to such a pattern, the point is considered as random noise.

The analysis method of this project is: record the changes of each point into an array and convert them into signals by a discrete Fourier transform. Then the similarity is calculated with a known reference point of following motion pattern, that is, the change records of real edge points. If it has a high similarity, it is considered as an edge point. Then the closed contour is obtained by combining edges, and the set of pixels inside the contour is taken as the image segmentation result. Experimental results show that this method can achieve satisfactory segmentation results on noisy images.

1.2 Objective of the project

This project aims to enhance image information by simulating eye saccadic movements, then analyse this information using appropriate new methods. This project provides a new method for edge detection and segmentation of noisy images. The analysis results can enhance the edge detection and segmentation results of noisy images. Moreover, a significant feature of this project is, this method can detect edges and remove noise at the same time.

1.3 Deliverables

The deliverables contains materials below:

- The simulated data generation method and example.
- The code of the experiment with a description.
- The result and evaluation
- The final report

1.4 Timeline and project management

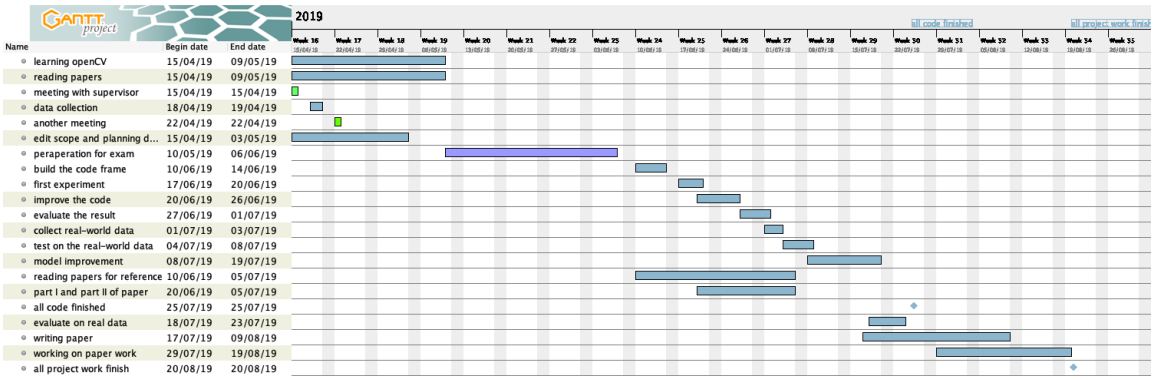


Figure 1.1 The Gantt chart of this project

At the beginning of the project, I made the project plan with the help of two supervisors, and proceeded according to the plan. The Gantt chart is used to manage the progress of the project. Purple occupies the space for the final exam, during which there is no project related work. The plan contains work and milestones for the major phases of the project. In practice, things didn't go exactly as planned, and changes were made to the content of the project itself.

The project is divided into five phases:

1. In the first stage, I read articles and learned related technical tools.
2. The second phase is to design a simulated generation method
3. The third stage is to design and implement the method on the simulated data.
4. The fourth stage is to test the method and evaluation.
5. The final phase is to write the final report and finish other document works.

1.5 Structure of the report

Chapter 2:

The second chapter is the literature review and the related work introduction, this chapter first reviews eye movement research in the biological vision domain. Secondly, it introduces the computer vision fields involved in this project, such as edge detection, image noise, image segmentation. Finally, this chapter introduces the work of other researchers on computer vision inspired by eye movement.

Chapter 3:

The third chapter is methodology, which will focus on the experimental ideas, experimental design and experimental process of this project. In this chapter, the idea of this method will be described in detail at first, the research hypothesis will be proposed, and the validity of this idea will be proved through relevant scientific theories. Then explain the theory used in the experiment.

Chapter 4:

The fourth chapter is the experimental description, which will include the experimental process and pseudo-code description. This chapter includes the platform used in the experiment, programming language, and expansion package. The reader will have a detailed understanding of how the experiment is implemented.

Chapter 5:

The fifth chapter is mainly about experimental results and evaluation. This section will show the image segmentation results and introduce the evaluation criteria. A comparison of the results with the baseline approach will occupy a large portion of this chapter.

Chapter 6:

Chapter 6 is a review and evaluation of the entire project, which will analyse the advantages and limitations of this approach — and looking forward to further work in the future.

Chapter 2
Literature review and related works

2.1 Eye movement research in the biological field

2.1.1 The introduction of saccade

Human vision is an extremely complex field. In biology, people have never stopped exploring the visual system. Most of the input of the biological visual system is from the eye. Studies have shown that the amount of information received by the retina of eyes is much less than that received by the visual centre of the brain. This fact implies the ability of biological eyeballs to compress information (Kuffler, 1953).The source of such information compression ability is the pre-processing of visual information in the form of biological electronic signals, such as feature abstraction and attention mechanism.

Biological eyes are capable of pre-processing visual information in a variety of ways, and a large part of this ability comes from eye movements. When it comes to eye movements, we usually divide eye movements into periods of eye movement during sleep and eye movements during visual tasks such as browsing and reading. Eye movements during sleep are more related to memory storage in the brain(Maquet et al., 1996). Figure 2.1 shows the eye movement path of the volunteers during food preparation. With prior knowledge of the required materials, the eye area selectively stayed in the relevant material area, ignoring information irrelevant to the current visual search task.

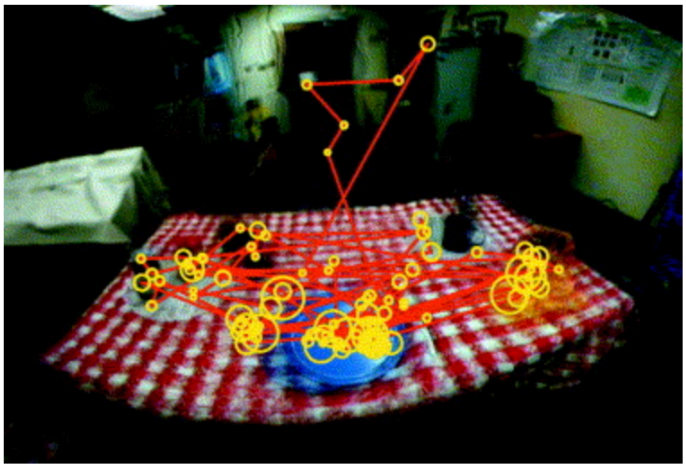


Figure 2.1 A saccade path diagram for a typical visual search task. The yellow circle represents the area where the centre of vision resides, and the red line represents the eye movement path (Hayhoe and Ballard, 2005).

This project focuses on eye movements during visual tasks. More specifically, we focused on a high-frequency, low-amplitude eye movement pattern. This eye movement pattern is called a saccade.

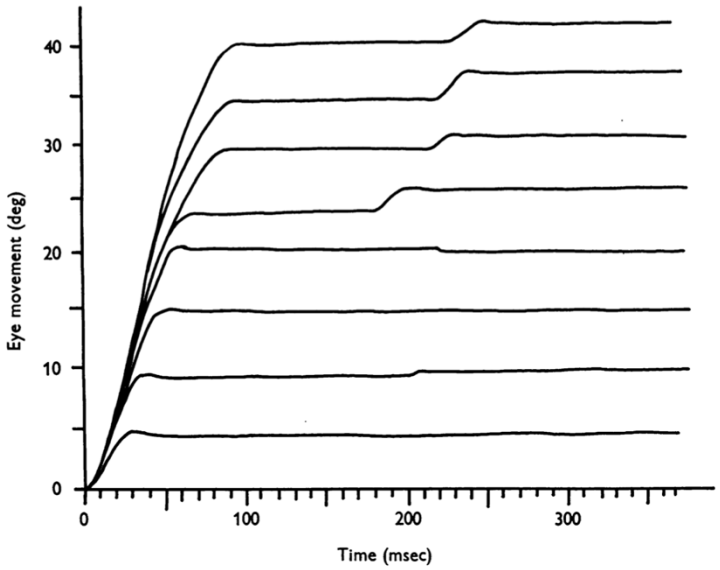


Figure 2.2 Typical saccade movement from 5 degree to 40 degree.(Robinson, 1964)

2.1.2 The features of saccade

Figure 2.2 shows a series of typical saccade motion angles changing over time.Besides, figure 2.3 illustrates the relationship between the magnitude and duration of typical saccade motion. Based on these data features and measurements of muscle activity during the saccade, (Robinson, 1964) divided saccades into four categories: normal, isometric, isotonic and high inertia.

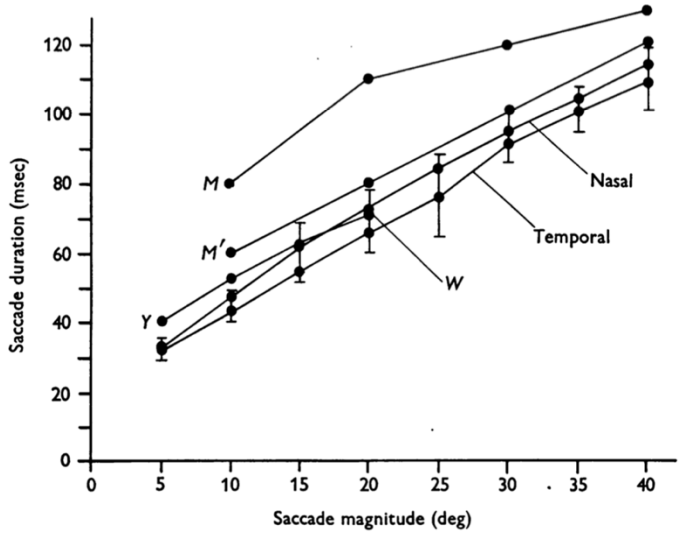


Figure 2.3 The duration and magnitude of saccade movement (Robinson, 1964)

In addition to the features mentioned above, another feature of saccade movement focused in this paper is unconsciousness. Eye movement is not consciously controlled, but an automatic mechanism. The saccade, associated with our attention mechanism, is a reflexive urge. This urge is triggered by the particular visual information we receive (color, shape) and the particular priori knowledge of the current visual task (the target in the visual search task). The saccade neurons in the superior colliculus (SC) and frontal eye fields (FEF) are activated by this visual information received by the eyeball. Specifically, the bioelectrical activity of neurons exceeds the threshold of starting saccade, thus triggering the saccade movement with a fixed pattern (Rayner et al., 1996).

In the field of vision, the researchers found that saccades are controlled by the cerebral cortex and are highly correlated with spatial memory, attention, and visual search (Pierrot-Deseilligny et al., 2004). Other studies have shown that the brain stem controls saccadic movement, and there is evidence that the biological signals that control saccade related muscles are generated in the brain stem and transmitted through the nervous system to the eye. Neurons in the brainstem not only produce the signals that make saccades happen, but also determine the direction and amplitude of saccades through the intensity of the signals (Sparks, 2002).

Saccades play an essential role in various visual tasks, which can be divided into visual search task and memory task. The pattern and function of saccade movement are different in these two tasks. The work of (Castelhano et al., 2009) visualized and quantified the saccade movement patterns in these two tasks.

(A) (B)

Figure2.4 Figure A shows the saccade trajectories in the memorization task, and figure B shows the saccade trajectories in the visual search task(Castelhano et al., 2009)

Figure 2.4 shows the different saccade trajectories in the two tasks, and we can see that in the memorization task, saccade trajectories are more extensive, and attention is more widely distributed. In the visual search task, saccades have a more compact

track and a significant allocation of attention to relevant content. Table 2.1 shows the difference of two kinds of tasks in the following indicators: percentage of scene area fixated, total scan path length, the total number of fixations, average fixation duration, average saccade amplitude, and the elapsed time to first saccade execution.

Table 2.1 The table illustrates the differences between memorization tasks and visual search in saccade mode(Castelhano et al., 2009).

	Memorization task		Visual search task		
	Mean	*SE*	Mean	*SE*	Difference
Percentage of scene area fixated	48%	0.08%	37%	0.09%	11%*
Total scan path length	82°	2.2°	74°	2.4°	8°**
Total number of fixations	27.6	0.09	24.2	0.08	3.4**
Average fixation duration (ms)	287	0.29	292	0.24	−5
Average saccade amplitude (deg)	3.0°	0.03	3.1°	0.03	−0.1
Elapsed time to first saccade execution (ms)	317	9.73	269	7.14	48**

Saccades can focus on relevant objects faster in conscious visual search and reduce the impact of irrelevant objects on visual search tasks. (Findlay, 1997)'s experiments show that colour is the most significant factor in saccades and path selection. In multi-objective object search, unconscious saccade movement will produce a surprisingly regular distribution as figure 2.5, which implies that the mechanism of saccade movement is composed of intricate patterns.

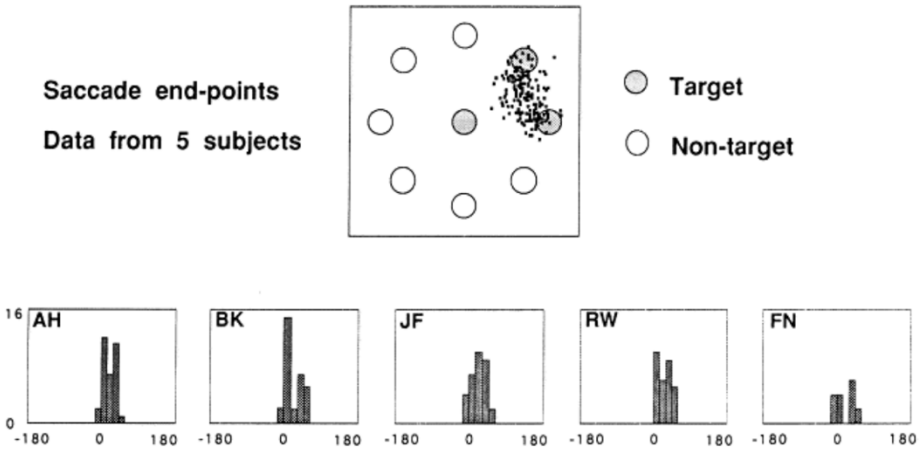


Figure 2.5 (Findlay, 1997) The saccade endpoint distribution of the five volunteers in the visual search showed apparent regularity.

2.1.3 The benefit of saccades

Saccades also play a significant role in information integration in human cognitive behaviour. The ability of saccades to extract information has been proved in an experiment(Irwin, 1991). Saccades integrate the visual information near the stopping point and improves the recognition ability of the visual system to the scene. In the human visual system, the saccade movement also undertakes the task of recognizing the edge and structure of objects. Saccade motion senses visual features such as lines and edges through a signal filtering method. The line and edge sensors respond to the local energy peaks of the odd-symmetric filter and even-symmetric filter respectively(Concetta Morrone and Burr, 1988). This mechanism enables the biological eye to respond and predict the structure and location of objects. These functions of saccades can bring many benefits to animals and humans. Mantises and pigeons, for example, can use their eye movements to get information about distance and depth in their field of vision(Bruckstein et al., 2005), which is of great help to predation and other behaviours.

Another contribution of the saccade to the visual system is the ability to perform visual tasks in the presence of interference, noise. (Zhao et al., 2012) studied the reasons why saccade movement enhanced perception ability. In an experiment they worked on, they compared the ability of five subjects (who were unaware of the purpose of the experiment and proved to have standard vision level through the optical test) to recognize the direction of a single target with noise interference and without noise. The results show that in the case of noise interference, rapid saccade in a fixed direction has a stronger resistance to noise and is less affected by interference, especially in the case of higher noise interference level.

The results of their work are shown in figure 2.6. In the image, a shorter duration means better cognitive performance. Five groups of images were compared with and without external noise by five different volunteers. Through maximum likelihood program analysis, rapid saccades in a fixed direction performed better and was less affected by noise. This also reveals the ability to ignore irrelevant details and eliminate distracting information of saccade.

The study of eye movement, especially saccades, in these biological fields explores the principles, formation, and characteristics of saccade movement. Although the relationship between eye movements and visual tasks is not fully understood, these studies reveal the importance of saccades for visual abilities such as cognition, feature extraction, spatial perception, attention, and noise exclusion. Moreover, it brings inspiration to the application of these abilities in the computer vision system.

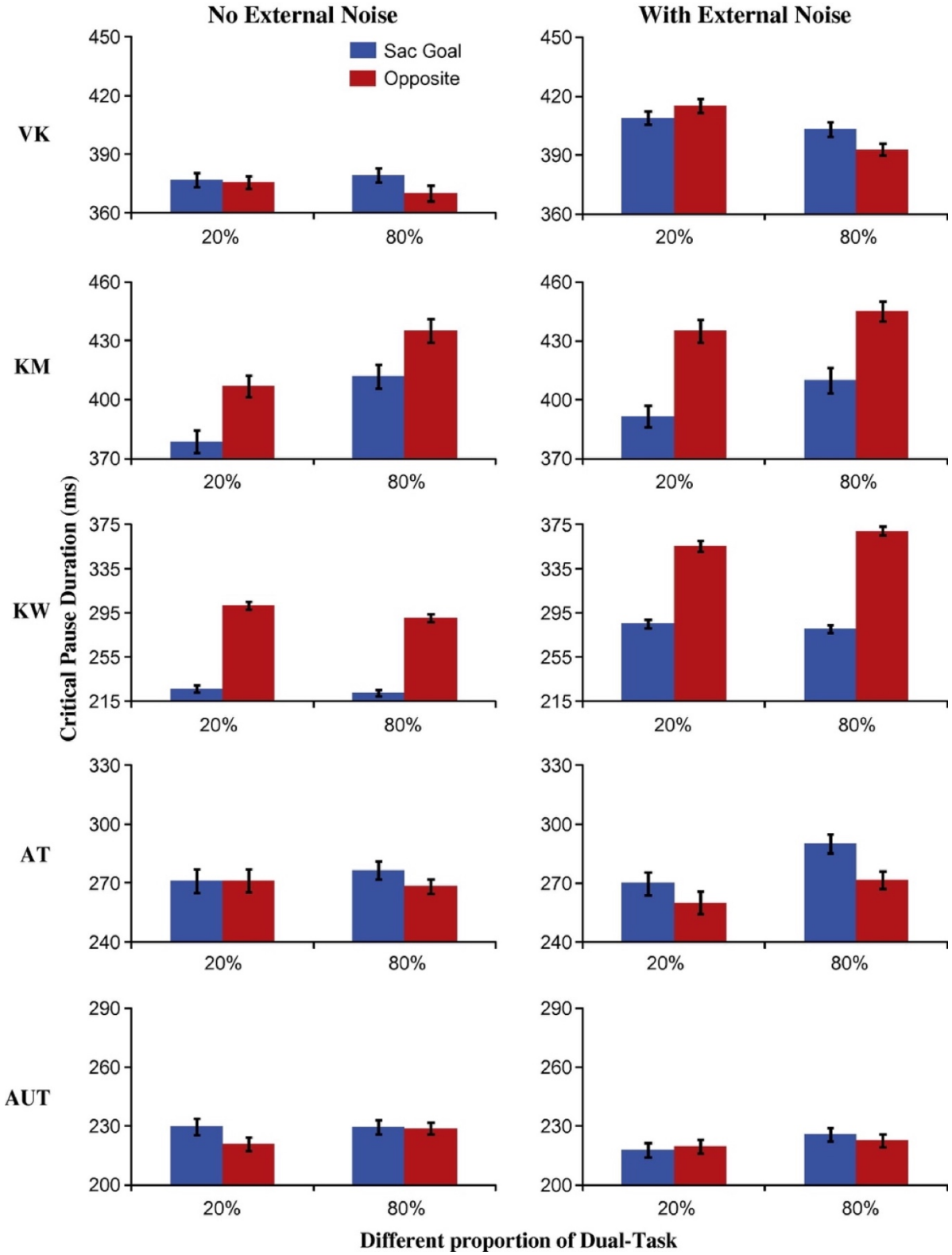


Figure 2.6 The performance comparison of saccade motion in single target direction recognition task with or without external noise(Zhao et al., 2012)

2.2 Image noise

2.2.1 Introduction of image noise

In image processing tasks, we expect well-designed algorithms and programs to give us satisfactory results. However, this expectation is often not satisfied due to factors other than algorithm design. Image noise is one such factor. Image noise is defined as "an unwanted component of the image" (Boncelet, 2009). The factors that produce noise are very diverse. All kinds of noise will be generated in the process of image transmission, production, preservation and modification.

The most common noise is the Gaussian noise; it occurs in almost all the signals. The occurrence of Gaussian noise follows a Gaussian distribution, and the probability density function is defined as the following formula:

$$p_a(a) = (2\pi)^{-n/2}|\Sigma|^{-1/2}e^{-(a-\mu)^T\Sigma^{-1}(a-\mu)/2}, \quad (2.1)$$

The two important parameters are standard deviation and expectation. In the standard case of a Gaussian distribution, we only need to consider the standard deviation. Figure 2.7 (A) shows the original image of an example image and (B) shows the image after adding Gaussian noise with mean value of 0 and standard deviation of 0.01.

(A) (B)

Figure 2.7 The image after adding Gussian noise is compared with the original image

Another common type of noise is salt and pepper noise, where only a few pixels are affected by noise, but their distribution is very random. The noise consists of white and black pixels, so it is called salt and pepper noise.

The formal description of salt and pepper noise is:

Let f(x,y) be the original image, and q(x,y) be the image processed by salt-pepper noise.

$$\Pr[q = f] = 1 - \alpha \tag{2.2}$$

$$\Pr[q = 255] = \alpha/2 \tag{2.3}$$

$$\Pr[q = 0] = \alpha/2, \tag{2.4}$$

The percentage of pixel values that do not change is 1 − α, where the α / 2 pixels turn black (intensity 0) and α / 2 of the pixels turn white (intensity 255).Figure 2.8 shows the results of Figure 2.7 (A) after adding salt and pepper noise with α value of 0.03.

Figure 2.8 The image after adding salt and pepper noise with α value of 0.03.

2.2.2 Noise removal

Noise level estimation is crucial. Many image processing algorithms can choose different parameters according to different noise levels to achieve the best performance.(Liu et al., 2006) used the Bayesian MAP inference to deduce the noise level of a single image.(Tai and Yang, 2008) proposed a method of noise level estimation using Laplace operator and adaptive edge detection, which achieved excellent performance.(Pyatykh et al., 2012)used principal component analysis to estimate noise level and found the similarity relationship between the minimum eigenvalue of image block covariance and image noise variance

Noise removal is an important field in computer vision and image processing. The two main image denoising methods are spatial feature-based method and signal filtering based method. The problem with denoising methods is that the details of the image are lost while the noise is removed. A series of adaptive filtering algorithms perform well in the denoising of digital images, such as median filtering, k-nearest neighbour

averaging, Lee additive and multiplicative filtering, and modified Wallis filtering(Mastin, 1985). The principle of these algorithms is to find the direct relationship between the grey value of the centre pixel and other adjacent pixels in a fixed window in the spatial domain (Lee, J.-S., 1981). Some algorithms can identify noise from the grayscale information of the whole image. The combination of frequency domain and space domain also provides a new idea for image denoising. The algorithm proposed by (Hirani and Totsuka, 1996) makes full use of the advantages of the two domains and avoids some limitations.

In recent years, with the rise of neural network research, researchers have shifted their attention to image denoising using neural networks. These methods exceed the performance of traditional methods in image denoising. (Burger et al., 2012)'s research proposed that multi-layer perceptron is used for image denoising, which not only achieves outstanding performance but also performs well in some uncommon noise types, with strong universality. (Xie et al., 2012)'s method combines sparse coding and pre-trained deep network with the denoising automatic encoder. This method applies an automatic encoder designed for unsupervised feature learning to image denoising. (Zhang et al., 2017)'s feedforward denoising convolutional neural network (DnCNNs) can deal with an unknown level of Gaussian noise and exceed the performance of Gaussian filtering.

Image noise is one of the disadvantageous factors that affect the performance of image processing tasks. Noise level estimation and noise removal are both vital tasks in the computer vision field. The latest research methods tend to use machine learning and deep learning to complete these two tasks.

2.3 Edge detection

Edge detection is an essential method in image pre-processing. The purpose of edge detection is to find the boundary of objects in the image and get the edge image. Edge image can be used for information extraction and image segmentation in computer vision. For a digital image, there is a high probability of an edge appearing where the intensity varies greatly, or where there is a discontinuity(Sonka et al., 2014). Most methods of edge detection are based on local discontinuity and sharp change of image.

The discontinuity of image intensity may be caused by the following factors(Barrow and Tenenbaum, 1981):

1. Change of object depth in the image;
2. Change of object surface orientation in the image.

3. Changes of object texture in the image.
4. Change of light brightness in the image

Our expectation for edge detection is that it returns a continuous set of curves that accurately outline the boundary of an object. These boundaries not only represent the structure of the object but also eliminate irrelevant information. This can simplify the representation of images and reduce many computations in image analysis. However, in reality, edge detection cannot get ideal results(Marr and Hildreth, 1980). Because of the complexity of image information, edges are often not connected. Another problem of edge detection is that the edge does not correspond to the object boundary in the image, that is, fake edge. These problems make it challenging to analyse the results of edge detection and reduce the performance of subsequent works.

In edge detection, a common method is to use an edge detector, a filter capable of extracting local intensity changes of the image to perform convolution operation on the image. Gradient operators become suitable edge detectors, which can calculate local gradients on two-dimensional data. The oldest gradient operator is the Robert edge operator(Roberts, 1963), which is very simple and consists of only the following two two-by-two size kernels.

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.6}$$

The Robert operator can highlight the intensity variation on the diagonal. However, its drawback is that it is susceptible to noise. The reason for this defect is that the size of Robert operator kernel is small and only a few pixels nearby are used to calculate the gradient. The gradient calculation method of Robert operator can be written as:

$$\nabla I(x, y) = G(x, y) = \sqrt{G_x^2 + G_y^2} \tag{2.7}$$

Figure 2.9 (B) shows the edge image detected by Robert operator, and figure 2.10 (B) shows the performance of Robert operator in the case of noise in the same picture. It can be seen that the edge detected by Robert operator is very sensitive to noise.

(A) (B)

Figure 2.9 (A) is the original image, (B) is the edge image detected by Robert operator

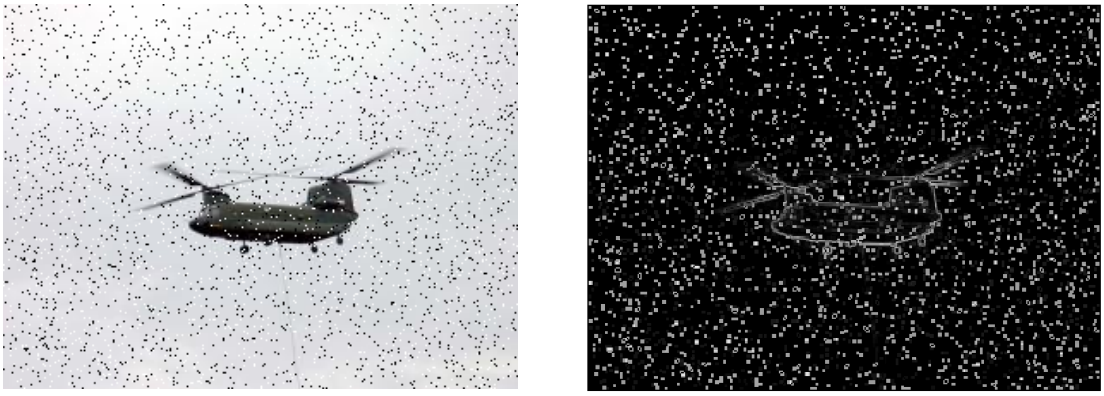


Figure 2.10 (A) is the original image with salt and pepper noise, (B) is the edge image detected by Robert operator

Figure 2.11 shows the intensity change curve of the edge pixel in the image. The principle of the gradient operator is to find the maximum value of the gradient, that is, the peak value of the first derivative of the intensity change curve shown in Figure 2.12. Sometimes finding peaks is difficult. When we observed the second derivative of the intensity curve, the zero-crossing point also represented the steepest slope of the intensity curve, as shown in Figure. 2.13.

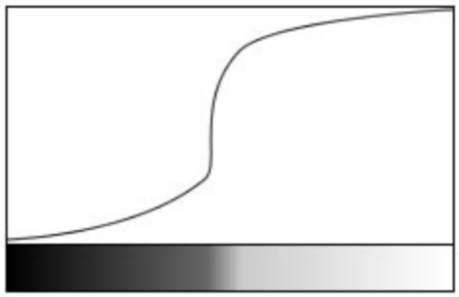


Figure 2.11 The intensity change curve of the edge pixel(The Sobel and Laplacian Edge Detectors)

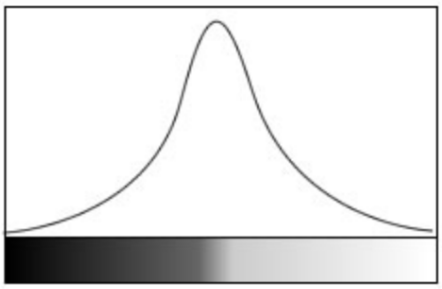


Figure 2.12 The first derivative of intensity change curve of the edge pixel(The Sobel and Laplacian Edge Detectors)

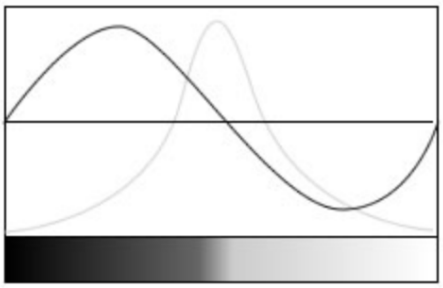


Figure 2.13 The second derivative of intensity change curve of the edge pixel(The Sobel and Laplacian Edge Detectors)

The Laplacian operator is a very popular gradient operator, which uses the second derivative to find the point where the gradient changes significantly(Mlsna and Rodriguez, 2009). The formula of Laplacian operator can be expressed as:

$$r^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.8}$$

The convolution kernel of Laplace operator for convolution is:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

(2.9)

Convolution of 8-connected neighbourhood pixels:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

(2.10)

Figure 2.14 is the result of edge extraction by Laplace operator in figure 2.9 (A). It can be observed that the edge extracted by Laplace operator is smoother. Figure 2.15 is the result of the same image with salt and pepper noise through the convolution of

Laplace operator. Laplacian operators are also very sensitive to noise due to the use of second order gradient.

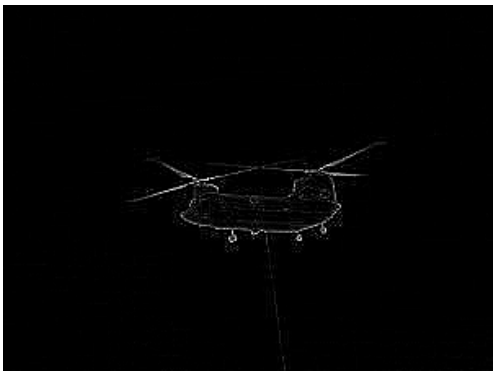


Figure 2.14 The edge image detected by Laplacian operator

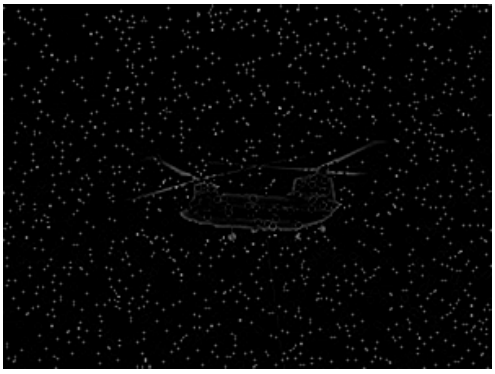


Figure 2.15 The edge image detected by Laplacian operator on an image with salt and pepper noise

Canny edge detector presented by (Canny, 1987) is one of the most popular edge detectors by far because it strictly meets the general standard of edge detection. Canny edge detection algorithm can detect as many edges as possible in the captured image, and the marked edge points are accurately positioned in the centre of the edge. Being sensitive to noise is also a problem of the Canny edge detector(Duan et al., 2005).

The Canny edge detector is a multi-stage algorithm. It consists of five stages as follow:

1. Use a Gaussian filter to filter noise and make the image smooth.

The formula of the Gaussian filter of size (2k+1)×(2k+1) is:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2+(j-(k+1))^2}{2\sigma^2}\right); 1 \le i,j \le (2k+1) \quad (2.11)$$

The size of the Gaussian filter core determines the performance of the Canny edge detector. The larger the size, the less sensitive the detector is to noise. Besides, with the increase of the kernel size of the Gaussian filter, the positioning error of edge detection will increase slightly. For most cases, 5 by 5 is a suitable size, but it varies from case to case.

2. Find the intensity gradient in the image

The Canny edge detector uses four different filters to find edges horizontally, vertically, and diagonally. The gradient is calculated as follows:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \tag{2.12}$$

3. Eliminate fake edges using a non-maximum suppression method

The strategy of the non-maximum suppression method is to set the intensity of the point with the highest local gradient to 255 and set all the other points to 0 so that the detector only responds to the point with the most active response

4. Use double thresholds to identify potential edges.

At this stage, the detector sets two thresholds, high and low. Points with intensity greater than the high threshold are marked as strong edge points. Points with intensity larger than the low threshold but smaller than the high threshold are marked as weak edges. The intensity of a point below the low threshold is reduced to 0 by non-maximum suppression.

5. Edge tracking by hysteresis

In the last stage, the strong edge point is connected to the weak edge point in its 8-connected neighbourhood pixels. This can filter out the random noise that appears far away from the strong edge points.

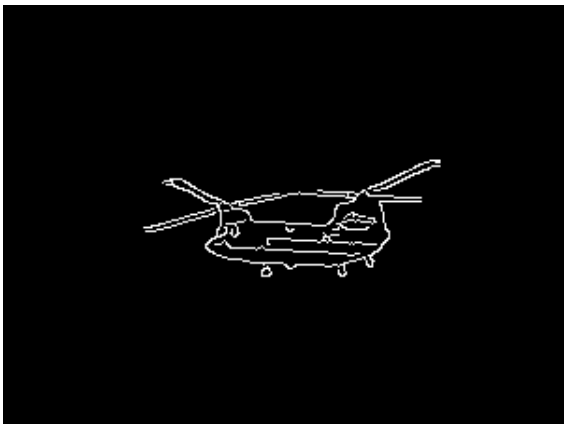


Figure 2.16 The edge image detected by Canny edge detector

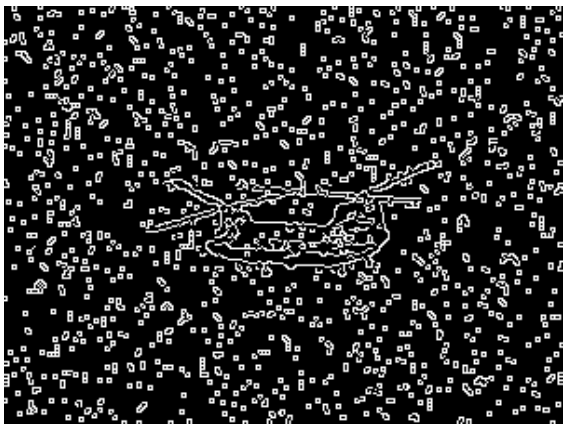


Figure 2.17 The edge image detected by Canny edge detector on noise image

Figure 2.16 shows the result of Canny edge detector for figure 2.9 (A), Figure 2.17 is the result of a canny detector for the same image with noise.

2.4 Image segmentation

Image segmentation is a process that can divide adjacent pixels in an image into several sub-sets according to certain rules. The goal of image segmentation is to make the representation of images more meaningful, easier to understand and analyse. The ideal segmentation result should segment pixels with similar features into the same group.(Fu and Mui, 1981) Moreover, the group after image segmentation should have a strong correspondence with objects and regions in the real world. Image segmentation has important applications in many fields, such as image information retrieval, medical imaging, object detection and various recognition tasks.

Image segmentation can be roughly divided into two categories. One is complete segmentation, which means that the result of segmentation corresponds to the unique objects in the real world. Another type is partial segmentation. The result of partial segmentation does not correspond to objects in the real world, but only reflects the similarity of pixel intensity distribution, variation and other information in adjacent regions (Sonka et al., 2014). Complete segmentation relies on semantic knowledge of the task and high-level image processing, and the content of this section will focus on partial segmentation.

The main methods of unsupervised image segmentation are:

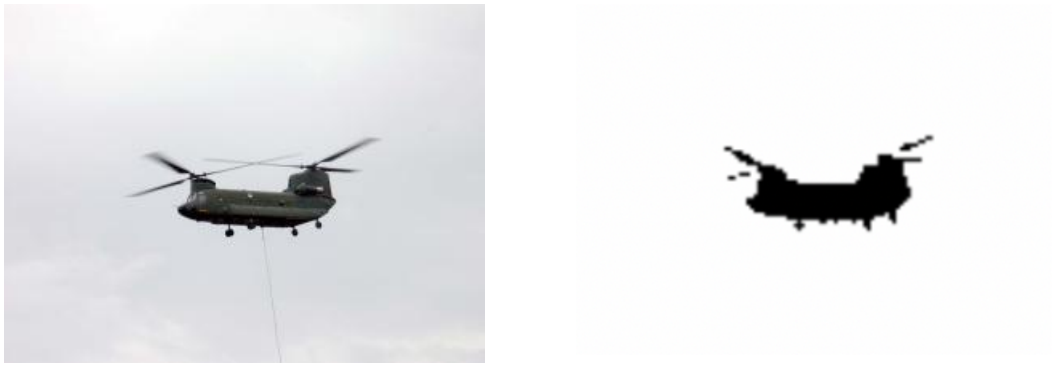
1. Thresholding based segmentation
2. Edge based segmentation
3. Region based segmentation
4. Clustering.

Segmentation based on the threshold is the most basic segmentation algorithm; the grey-level threshold method is the most simple threshold method. This method simply compares the intensity of each pixel with the threshold value. A pixel whose intensity is higher than the threshold value is set as a foreground object, and the pixels whose intensity is lower than the threshold value are marked as the background.

The key point of segmentation based on a threshold is the selection of the threshold. It is obviously not a wise decision to choose a fixed threshold for different images. An algorithm that automatically sets the threshold according to pixels distribution of the image is called an adaptive threshold algorithm. (Otsu, 1979)'s method is an excellent adaptive threshold algorithm. Otsu's method of threshold value by minimizing the intra-class diversities intensity variance to determine. The intra-class diversities variance is defined as：

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{2.13}$$

The results of the Otsu's method are shown in figure 2.18



（**A**） （**B**）

Figure 2.18 The result of Otsu's method

Edge-based segmentation is a vital segmentation algorithm which is also used in this project. Edge detection is the basis of edge segmentation(Muthukrishnan and Radha, 2011). Edge detection was covered in the previous section. Since edges are often discontinuous, edge integration is often required before segmentation(Kimmel, 2003).

Region-based segmentation is based on the assumption that adjacent pixels in a region will have similar intensity values. The basic method is to start growing with a seed pixel until there are no similar pixels around(Adams and Bischof, 1994). The key point of this method is finding an appropriate method to calculate the similarity between pixels and regions(Freixenet et al., 2002).

2.5 Related works

In computer vision, image processing and other research fields, the research inspired by saccade movement has been started very early. In computer vision, image processing and other research fields, the research inspired by saccade movement has been started very early. (Murray et al., 1995) developed a method to enhance motion detection and segmentation using the mechanism of saccade motion. The pattern of saccade movement is determined by the position and speed of the moving object. This method uses a finite state machine to transition from saccade motion to pursuit. Their conclusion is that saccade movement can move to a pursuit state more quickly. Their work found factors that affect success rates from saccades to pursuit state.

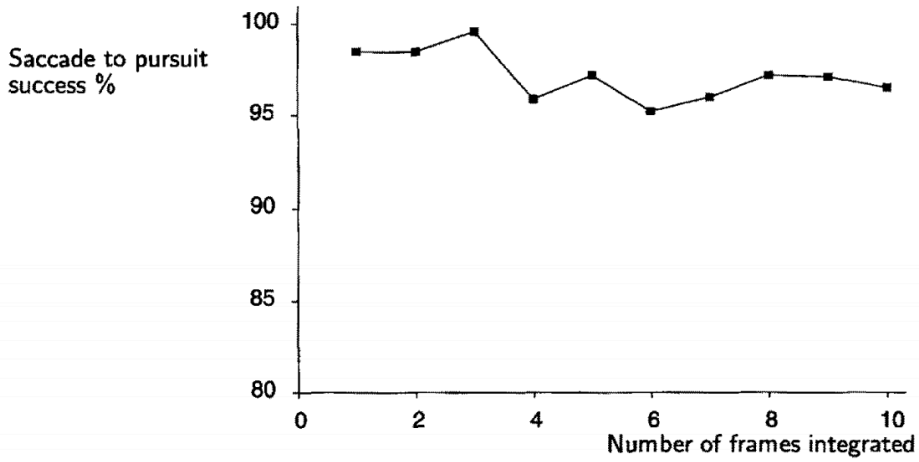


Figure 2.19 The number of frames before triggering saccade motion versus the success rate of state transformation

Figure 2.19 shows the relationship between the number of frames before triggering saccade motion and the success rate of state transformation.

(Neskovic et al., 2002) has designed a bio-inspired system that simulates saccade to identify cars in the video stream. Based on research into human vision, particularly attention mechanisms and fovea vision, the authors of this work defined objects as collections of features. Moreover, use a network structure to represent objects. Figure 2.20 shows a network structure. The first layer is the feature detector; the second layer takes features as input and outputs the probability of the occurrence of these features

to the next layer. This network structure explores the information in an image in a form similar to a human eye saccade.

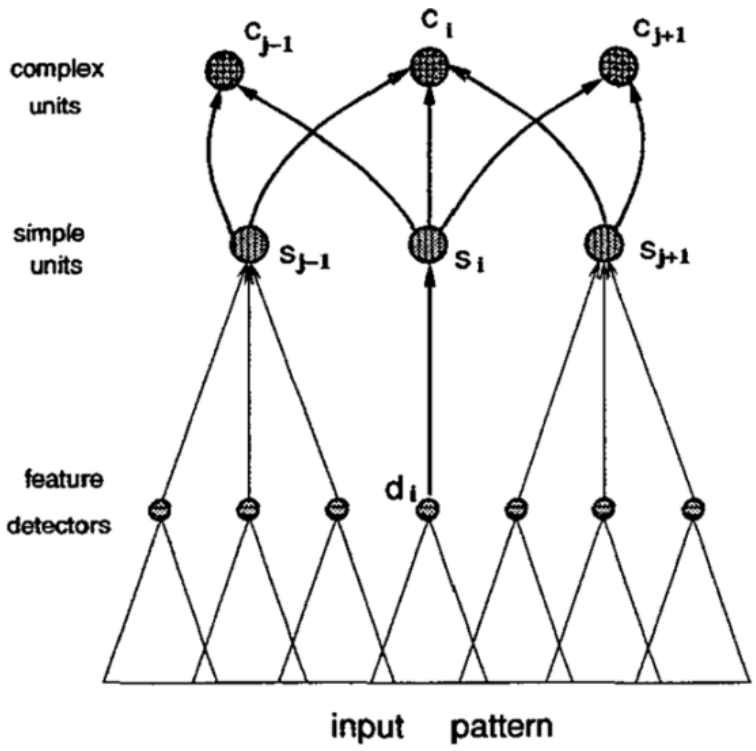


Figure 2.20 The network structure of object representation in (Neskovic et al., 2002)'s work

The feature detector extracts information through edge detection, which achieves an accuracy rate of nearly 100% when detecting stationary cars and more than 70% when detecting moving cars.

Inspired by the overlap of the retinal receptive field during eye movement, (Róka et al., 2007) developed a physical method to simulate eye movement. They investigated the structure of the retina in detail and used this simulated eye movement to enhance edge detection. Figure 2.21 shows the overlapping receptive fields during saccades.

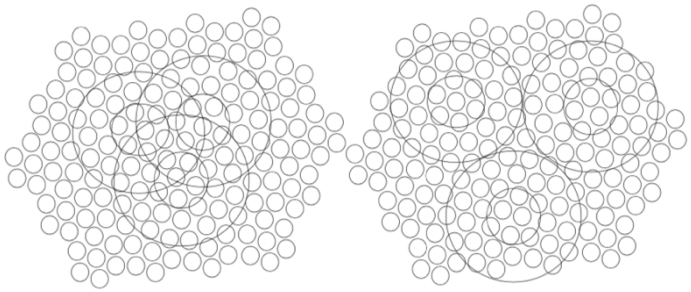


Figure 2.21 Overlapping receptive fields during saccades(Róka et al., 2007)

Their work simulates such overlap during the saccade and using an FPGA image receiver, as shown in figure 2.22, which can vibrate according to a particular pattern, to extract edge information with overlapping convolution.

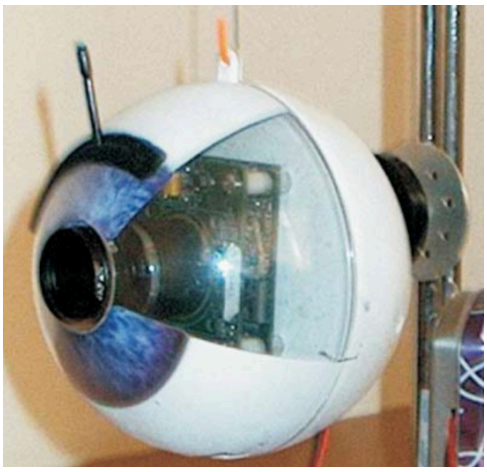


Figure 2.22 The FPGA image receiver (Róka et al., 2007)

Another edge detection method inspired by saccade movement was proposed by(Gao et al., 2008). They designed a good gradient operator with eye movement simulation ability to extract image edge information. The gradient operator they designed is time-dependent and generates a new convolution kernel at every moment according to the mechanism of the saccade. For input image I(x,y)of n-by-n-size, at time t, the output edge image S(x,y,t) is defined as:

$$S(x,y,t) \quad = \sum_{k=-N}^{N} \sum_{l=-N}^{N} f(k,l) I_t(x-k, y-l) \tag{2.14}$$

(Mignotte, 2017)has come up with another edge detector inspired by saccades. This edge detector simulates multiple directions of saccade motion and extracts a limited number of local features each time. Then these local features are combined according to morphological rules. The combined result becomes an edge image after edge thinning.

Figure 2.23 shows the framework of this method. Another feature of this method is to use short straight lines to approximate the fitting of smooth curves, and to fit all contours in polar coordinates, and then convert them into cartesian coordinates.

The image processing methods inspired by saccade focus on edge detection. As discussed in section 2.1, saccade motion plays an important role in feature abstraction, so it naturally becomes the methods of feature extraction in machine vision. In the

discussion of image segmentation in section 2.4, edge-based image segmentation relies on edge images. Saccade based method provides better edge images under the condition of noise, and these better edge images can improve the performance of edge-based image segmentation algorithm.

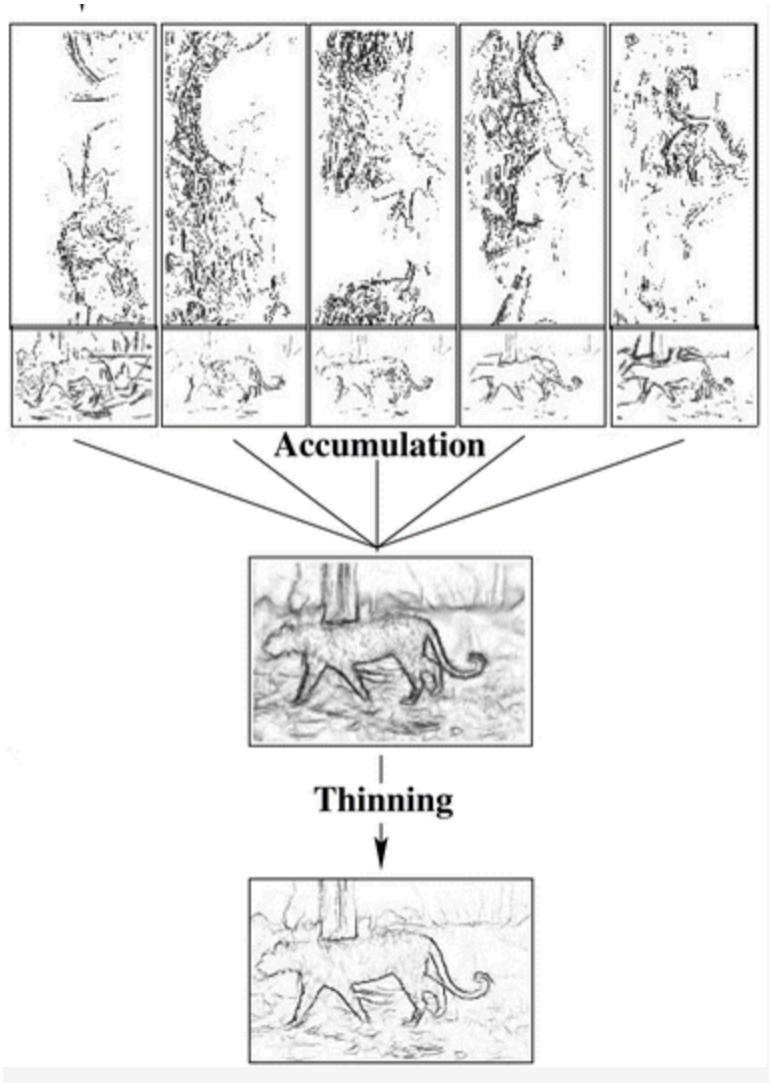


Figure 2.23 The framework of (Mignotte, 2017)'s method

Chapter 3
Methodology

3.1 Introduction

The focus of this chapter is to introduce the methodology used for the project and explain why it works in theory, especially from a computer vision perspective. Before introducing the project, it needs to be declared that the idea of the edge detection algorithm adopted in this project comes from my co-supervisor, Mohamed. He provides the initial code for the edge detection section. The experimental results were derived from the edited code, and essential parameters were modified to improve performance. We will discuss quantitatively how these parameters affect performance later.

The project can be divided into the following phases:

1. Data generation: the work in this stage is to carry out translation transformation and rotation of images in four directions with small amplitude. The transformation of the image should conform to a certain pattern. The operation of this stage can also be understood as data enhancement, generating a set of continuously changing images or video from a single image. In the process of data generation, random noise is added to verify the noise removal ability of the algorithm.

2. Data extraction: in this stage, the main work is to extract the data that the intensity of each pixel changes with time from continuous images. The data is represented as an array of vectors with the same length and number of images.

3. Data pre-processing: the data extracted in the previous stage is in the form of vector. Since motion patterns are estimated, a known edge vector can be used as a reference vector. At this stage, the vector storing the time change information of each pixel is converted into a signal by a discrete Fourier transform.

4. Edge detection and denoising: calculate the similarity between the intensity value of each point and the reference vector for the signals that show the change with time. The threshold method is used to mark the points whose pixel degree is higher than the threshold as edge points. This operation can filter out the background points (points with very small gradient value with the surrounding points) and noise (noise appearance does not conform to the pattern of edge points).

5. Image segmentation: the main task of this stage is to segment the edge image. As discussed in section 2.4, this project USES the edge-based segmentation method to segment the meaningful objects in the image. The segmentation results are binary images, and the segmentation objects are marked as white

The flow chart of this project is shown in figure 3.1

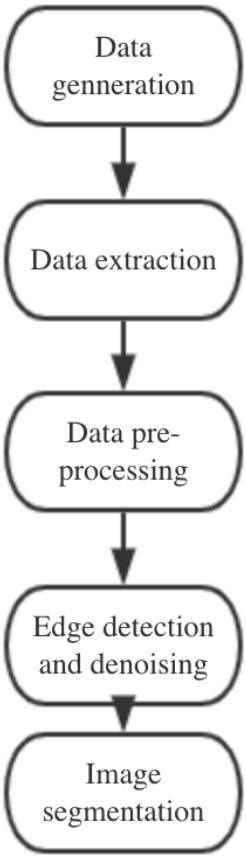


Figure 3.1 The flow chart of this project

3.2 Data generation

This phase of data generation is primarily inspired by the high-frequency saccades of the human eye. The purpose of data generation is also to generate continuous image stream with biological saccade feature. In the data generation stage, the input is an image, while the output is a noisy video generated according to the image, where the noise is random in each frame, and different noise levels can be adjusted.

The method of data generation is the affine transformation. This transformation does not change the content or relationship of the image (Weisstein, 2004). More specifically, this transformation preserves the original parallel lines and the original distance ratio of points on the same line. The affine transformation usually changes the distance between the points and the Angle between the lines (Stearns and Kannappan, 1995).

The formulaic expression of radiation transformation is as follows：

$$u = c_{11}x + c_{12}y + c_{13} \tag{3.1}$$

$$v = c_{21}x + c_{22}y + c_{23} \tag{3.2}$$

Affine transformations include translation, rotation, reflection, scaling, shear, and other operations (Oka and Kurauchi, 1990). The combination of these operations is also an affine transformation. In general, an affine transformation can be expressed as the product or sum of the original image and the affine matrix (Lee, M.-C. and Chen, 1999). In the data generation stage of this project, the affine transformation is mainly rotation and translation.

Figure 3.2 shows a translation operation. The transformation matrix of translation operation is:

$$\mathrm{T} = \begin{matrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{matrix} \tag{3.3}$$

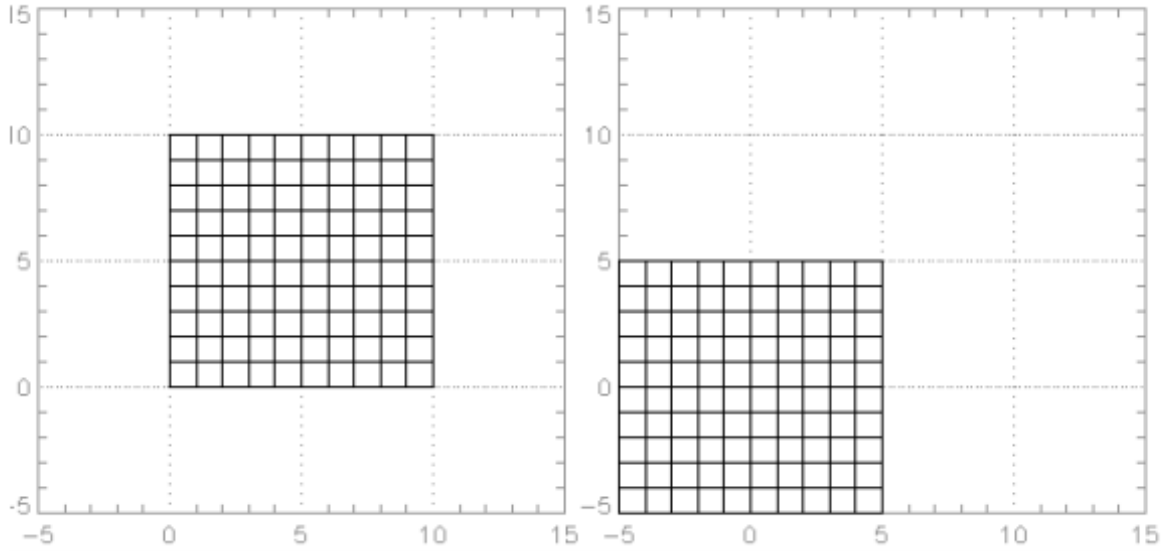


Figure 3.2 An example of translation.

Figure 3.3 shows a typical rotation. The transformation matrix of rotation transformation is:

$$\mathrm{T} = \begin{matrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{matrix} \tag{3.4}$$

The strategy of affine transformation in the data generation stage is:

1. Select small translation distance and rotation Angle when rotating and shifting. The aim is to detect gradient changes more accurately in later stages

2.Rotation and translation should be regular and follow a simple pattern. The significance of this is that the noise can be filtered out, and the edges that conform to the pattern can be detected.

In this project, the method of using affine transformation to generate data is that for each pixel (x, y), it is translated along the counterclockwise direction, and the amplitude of each translation is 1. That is, each pixel moves along (x+1, y), (x, y +1), (x -1, y), (x, y-1) cycles. While the translation is going on, it's also rotating at very small angles.

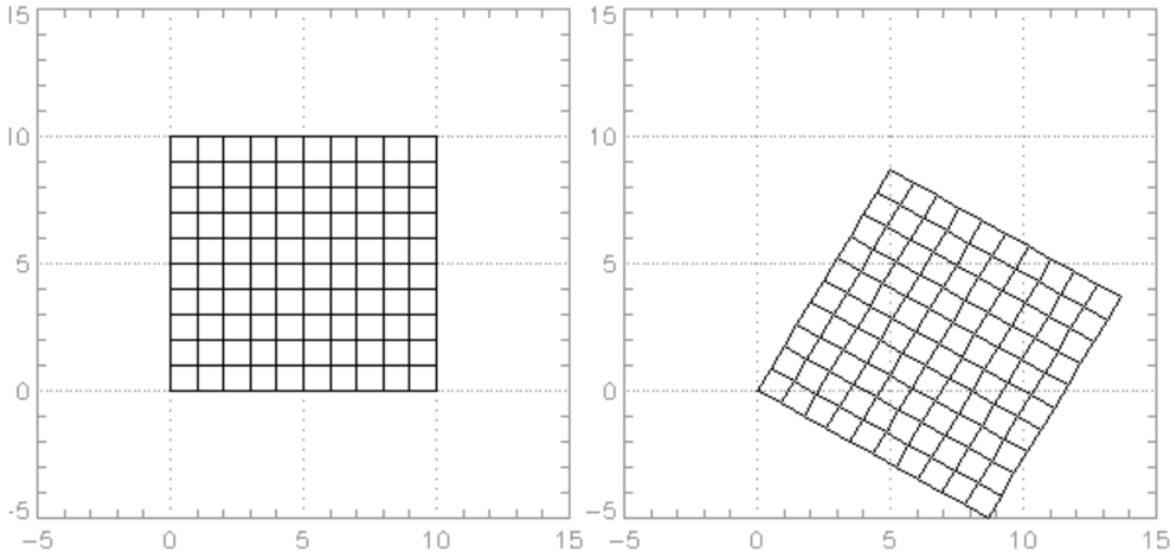


Figure 3.3 An example of rotation.

Figure 3.4 shows the image of the translation transformation. During the data generation phase, each pixel moves in this pattern to simulate human high-frequency eye movements.

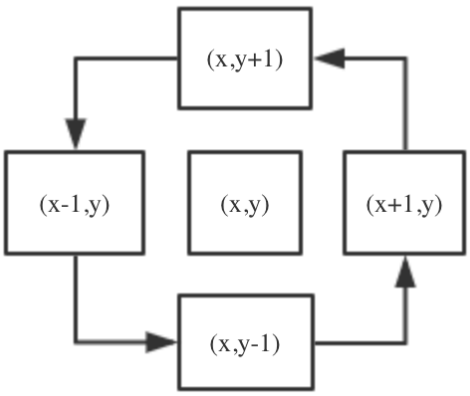


Figure 3.4 The pattern of simulated eye movement.

3.3 Data extraction

In the data extraction phase, the work is very simple, traverse the value of the coordinate in a series of images for each pixel coordinate and store it in a certain format. Arrays are a suitable storage format for this data. The purpose of extracting information is to analyse it, although the analysis takes place at the next stage. We can still find some patterns from the visualisation data at this stage.

The following series of images shows the intensity variations of three types of points under different image sequence lengths. Figure 3.5 and figure 3.6 respectively show the intensity change value of a background point in the sequence of 15 images and the sequence of 60 images. It can be seen that the intensity change feature of the background point hardly changes with time. The magnitude of the change in intensity is also tiny. Figure 3.7 and 3.8 show the intensity change value of an edge point in the sequence of 15 pictures and 60 pictures. It can be seen that the intensity change of the edge point shows a certain regularity, and the change range is large. Figure 3.9 and figure 3.10 show the changes of noise points in two cases. It can be seen that noise points have sharp changes, which can be distinguished from background points and edge points. With the extension of the length, the difference becomes more and more obvious.

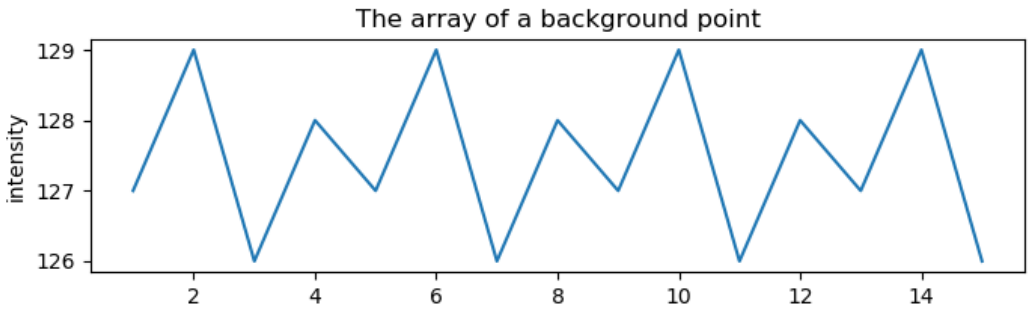


Figure 3.5 The intensity change of a background point of 15 images.

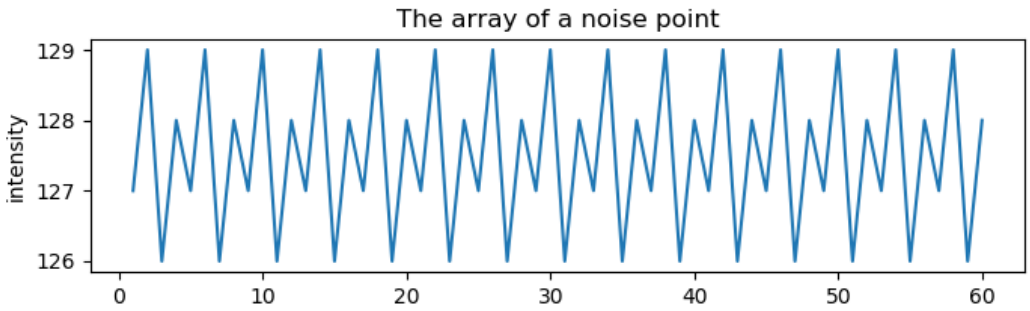


Figure 3.6 The intensity change of a background point of 60 images.

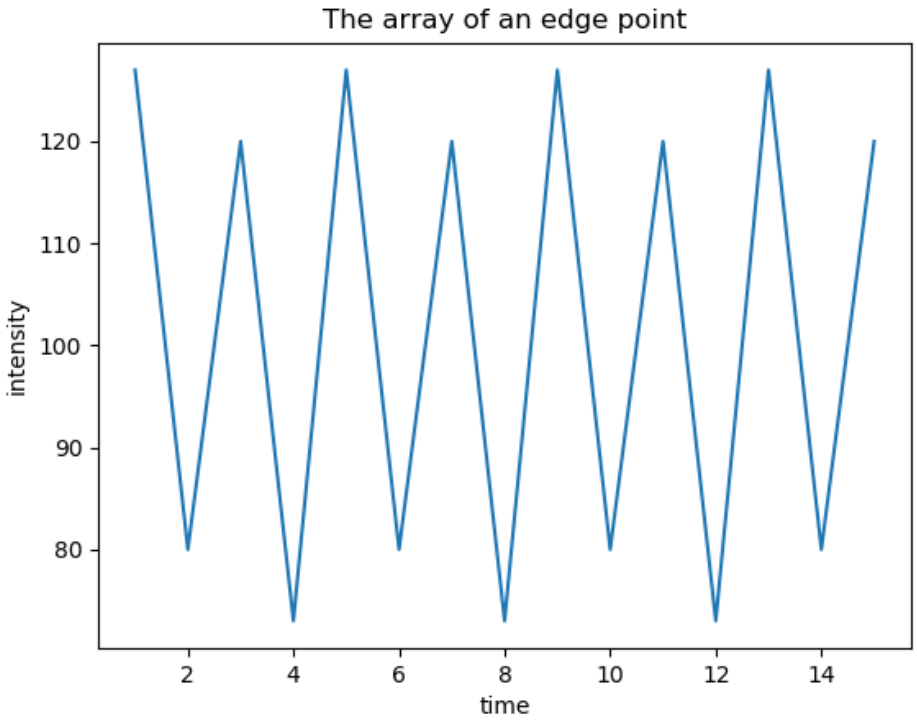


Figure 3.7 The intensity change of an edge point of 15 images.

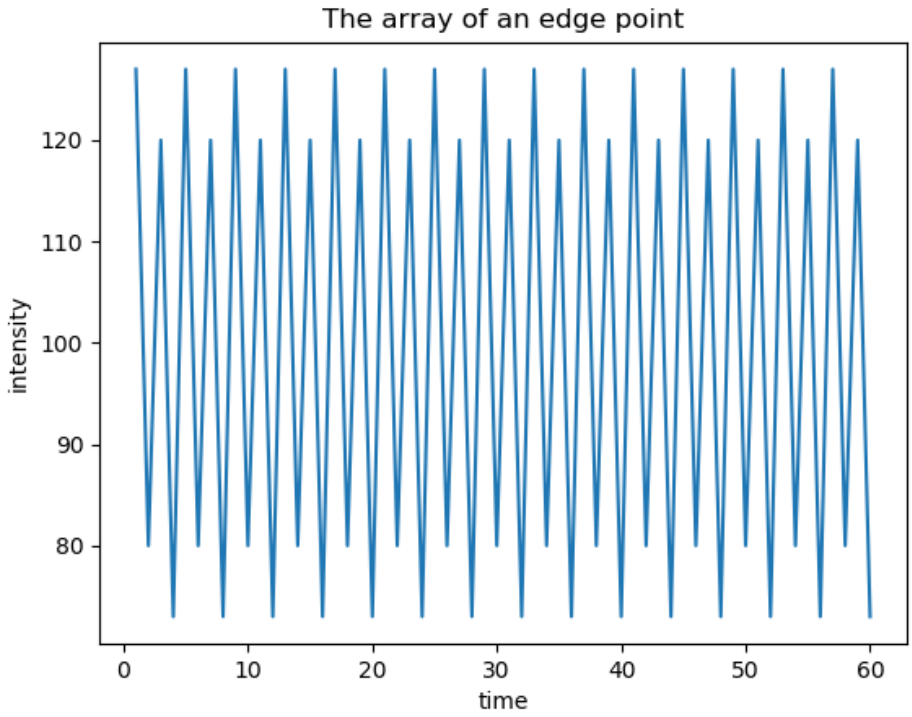


Figure 3.8 The intensity change of an edge point of 60 images.

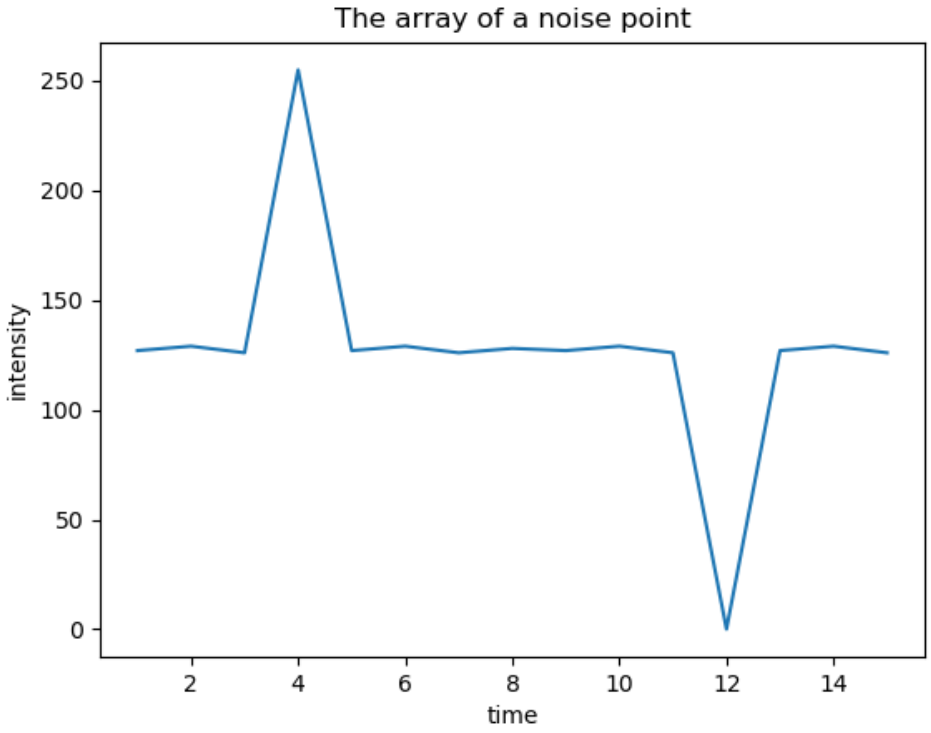


Figure 3.9 The intensity change of a noise point of 15 images.

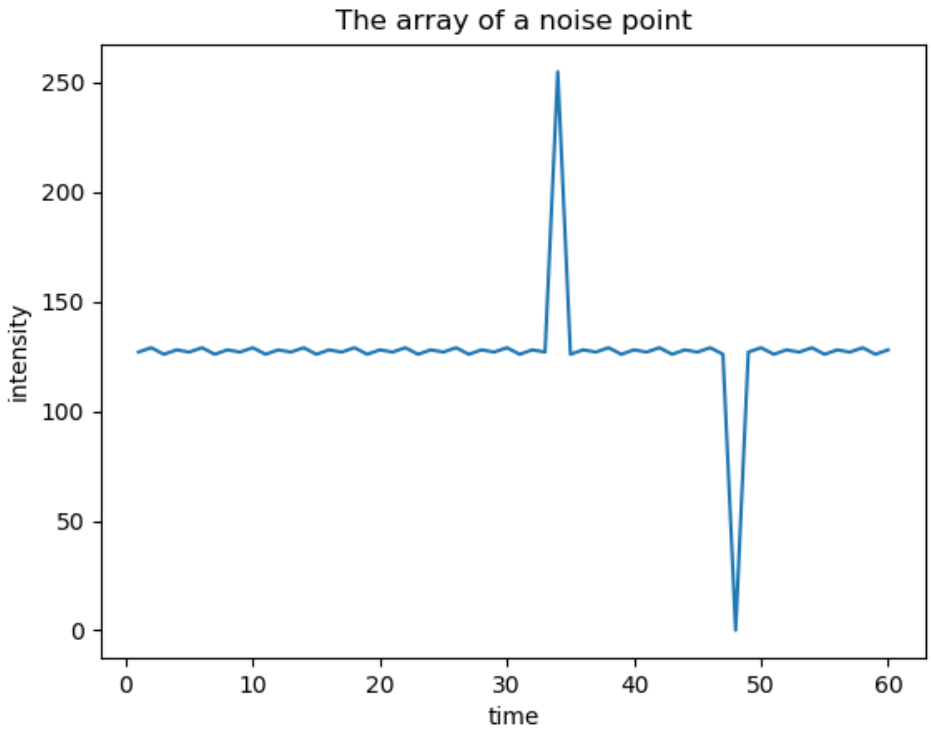


Figure 3.10 The intensity change of a noise point of 60 images.

3.4 Data processing

In the previous phase, we acquired data for analysis. In this phase, the goal is to convert the data into a form that is easier to analyse. Arrays can be thought of as vectors. For vector analysis, such as calculating similarity, the usual approach is to calculate Euclidean distance. However, due to the particularity of the problem dealt with in this project, these methods of calculating similarity do not work. For example, the edges of different colours have different intensity values, and the Euclidean distance will be very large.

This problem can be naturally understood as a time series analysis problem. These arrays can also be thought of as signals, and from signal processing, problems that are difficult to solve in the time domain can be well solved in the frequency domain. In this case, the tool we need is the discrete Fourier transform(DFT) (Bracewell and Bracewell, 1986) because the data are not continuous signals but discrete values. The formula of discrete Fourier transform is:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$
$$= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)] \qquad (3.5)$$

This calculation method is not very practical due to its complexity. The calculation method used in this project is the fast Fourier transform (FFT) (Cooley and Tukey, 1965). The result of the discrete Fourier transform is a complex number, where the real part represents the amplitude, and the imaginary part represents the phase. By such a transformation, the time-varying values in the time domain become a composite of discrete signals in the frequency domain.

The following series of images show a background point, edge point and noise point after discrete Fourier transform. Figures 3.11 and 3.12 are DFT results of a background point with 15 images and 60 images sequences and figures 3.13 and 3.14 are amplitude and phase distributions of an edge point with 15 images and 60 images sequences. Figures 3.15 and 3.16 are discrete Fourier transform results of a noise point in two cases.

It can be observed that the amplitude and phase of the background point are significantly different from the edge point and the noise point. It is easy to filter out the background points by calculating the similarity. Noise points and edge points are similar in amplitude and phase. An excellent way to distinguish them is to extend the length of the signal. After the extension, the difference between them became more apparent.

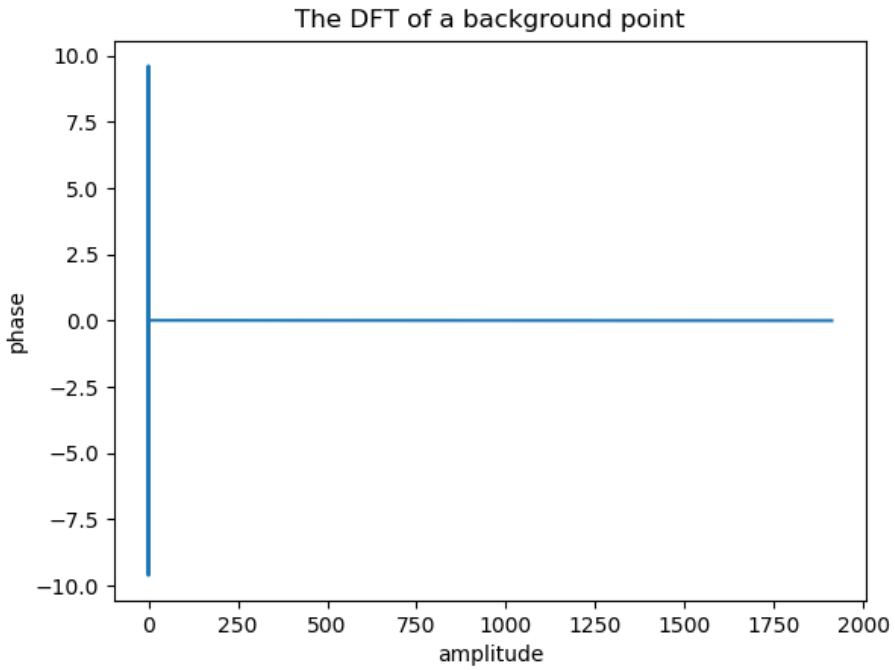


Figure 3.11 The amplitude and phase distribution of a background point of 15 images.

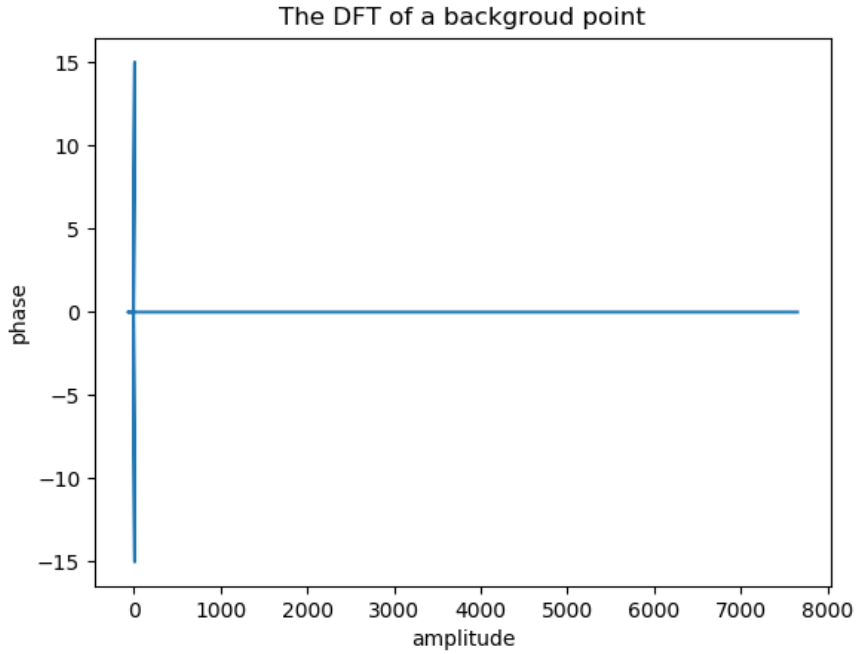


Figure 3.12 The amplitude and phase distribution of a background point of 60 images

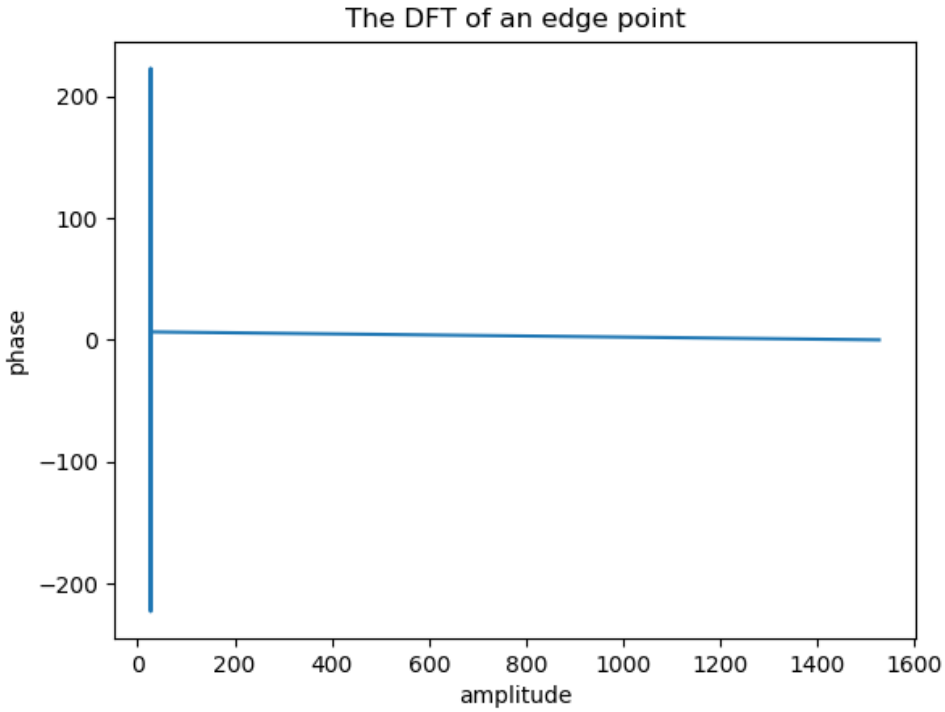


Figure 3.13 The amplitude and phase distribution of an edge point of 15 images

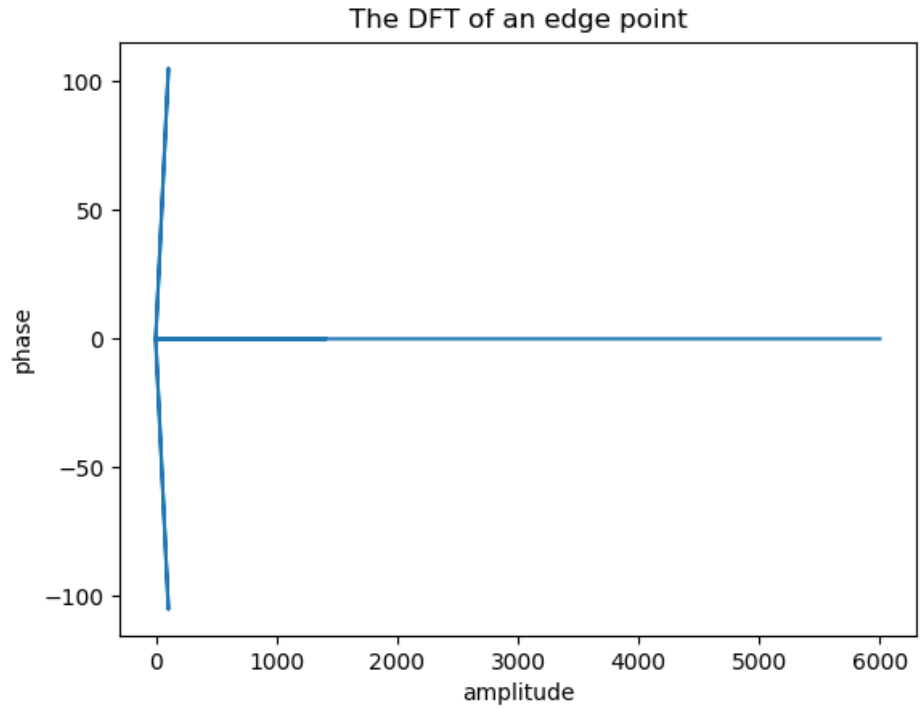


Figure 3.14 The amplitude and phase distribution of an edge point of 60 images

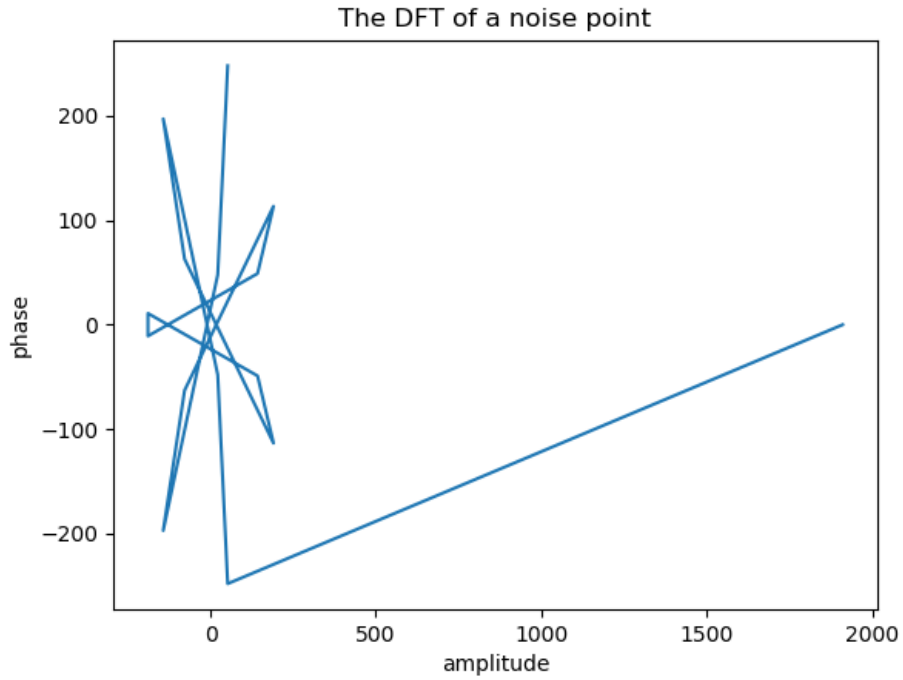


Figure 3.15 The amplitude and phase distribution of a noise point of 15 images

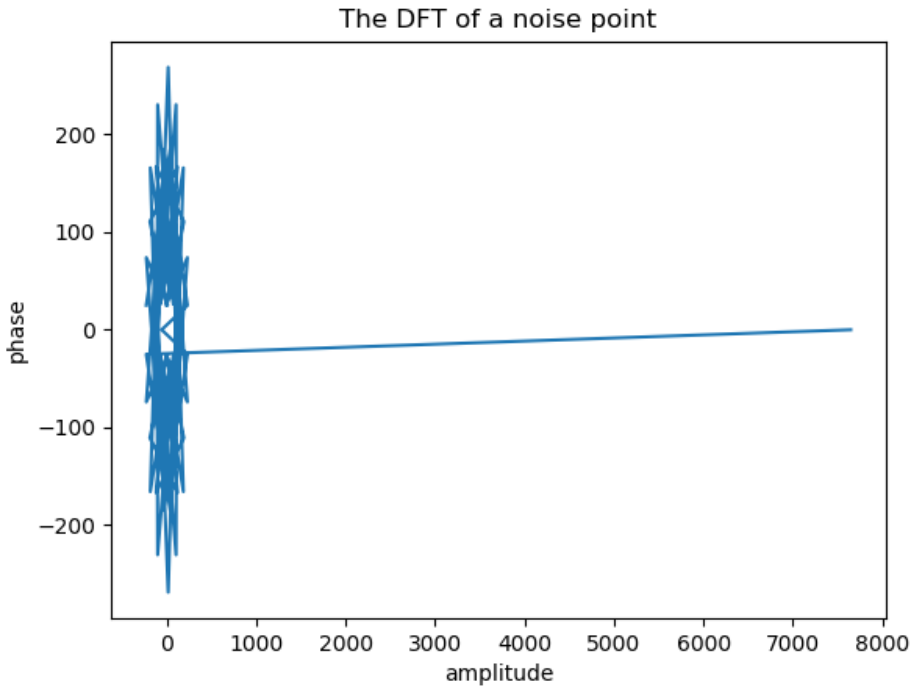


Figure 3.16 The amplitude and phase distribution of a noise point of 60 images

3.5 Edge detection

In the edge detection phase, there are two operations; one is to calculate the similarity; the other is the threshold. The process of this phase is to calculate the similarity of each signal (representing the change of a pixel in the time series) to the reference edge points. Then, by setting a reasonable threshold, the point similar to the reference edge point is retained as the edge point.

The Pearson correlation coefficient was used to calculate the similarity. The formula was:

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}} \tag{3.6}$$

Pearson correlation coefficient is a good indicator of the similarity of signals. After calculation, for each signal, there is a similarity of the reference edge. The next work is to set a reasonable threshold to filter out the low similarity points, that is, the background points and noise points in the image

The selection of threshold is one of the two important factors affecting the edge detection results; the other is the length of the signal. If the threshold value is set too large, some edge points will be filtered out, especially those whose intensity changes are not noticeable. If too small a threshold is selected, the noise will not be well filtered, and many fake edges will be generated. A relatively long signal is conducive to filtering noise out, because the longer the signal is, the smaller the proportion of noise in it. When calculating the similarity, the difference between noise points and edge points will become apparent. However, the long signal length will reduce the similarity between the edge point and the reference edge, this situation requires a lower threshold to avoid misclassification.

The edges obtained using this method are not optimal. The mark of edge points is not exactly located at edges of the image, but near edges. This is because the points near the edges show similar changes as they move. However, reasonable signal length and threshold can avoid some of these fake edge points.

3.6 Segmentation

After the edge image is detected, the next task is image segmentation. The task at this stage is to mark all the points inside the edges as white, representing an object in the real world. The segmentation task is made more challenging by some discontinuities in the edges detected in the previous step.

In this project, the segmentation method is closing operation in mathematical morphology. Mathematical morphology has been applied to image segmentation for a long time. The two basic operations of mathematical morphology are erosion and dilation. In binary morphology, structural elements (a preset binary graph) are used as detectors to detect whether certain areas contain matching shapes.

The formula of erosion is:

$$A \ominus B = \{z \in E | B_z \subseteq A\} \tag{3.6}$$

The formula of dilation is:

$$A \oplus B = \bigcup_{b \in B} A_b \tag{3.7}$$

The closing operation of morphology is dilation operation followed by erosion operation. This can fill holes in the image, connect adjacent objects, and smooth the image boundaries without significantly changing the area of the image.

The formula of closing operation is:

$$A \cdot B = (A \oplus B) \ominus B \tag{3.8}$$

The size of the structuring element is the main factor influencing the segmentation ability, which is proportional to the area within the image edge. For small areas of objects, small structural elements should be selected. If the size of structural elements is too large, the areas that should not be connected together will be considered as the same area, which adversely affects the segmentation accuracy. If the smaller structuring element size is chosen for a larger area, the interior of the object will exist some holes. This problem can be solved by using closing operations multiple times.

Chapter 4
Experiments

4.1 Experiment environment

In previous chapters, this paper reviews the progress of eye movement research and its applications in the field of computer vision. The theoretical basis of a new saccade - inspired image segmentation algorithm is presented. In this chapter, we will introduce how experiments verify the feasibility of this algorithm.

The programming language chosen for this experiment is python. The reasons for choosing python are as follows:

1. Python code is very easy to deploy and much cheaper to build and change.

2. The purpose of this experiment is to verify the hypothesis and test the performance of the algorithm. It has no high requirement on the running speed of the program.

3. Python has powerful extension packages such as Numpy and Opencv to support the functions required by this project, as well as Matplotlib for visualization and charting.

The IDE environment for this project is PyCharm, an easy-to-use and powerful python development environment.

All extension packages imported from outside the project are as follows：

- time.time
- cv2
- matplotlib.pyplot
- numpy
- scipy
- skimage
- scipy.signal.fftpack
- scipy.ndimage.fourier_shift
- scipy.signal.correlate
- skimage.data
- skimage.feature.register_translation
- skimage.feature.register_translation._upsampled_dft

The data used in this project is from the single object segmentation evaluation database(Alpert et al., 2011), which contains the image of a single object and ground truth. It can be used to evaluate the segmentation performance of different methods.

4.2 Data generation

In the data generation stage, the two main tasks are:

1. Regular affine transformation of still images

2. Add noise to each frame

The implementation of affine transformation depends on the cv2. warpAffine function of OpenCV.The parameter list and output of cv2.warpAffine are:

cv2.warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]]) → dst (OpenCV)

The first three parameters are necessary, and only the first three parameters are used in this project. The parameters are explained as follows:

src: input of the function, which in this project is the original image.

M: affine matrix determines the type of affine transformation. This project involves translation and rotation transformation matrices.

Desize: the size of the output image. In this project, the size of the output video is consistent with that of the original image.

For translation and rotation, there are different affine matrices. The affine matrix of translation is periodic, translation of one pixel in each of the four directions. The translation affine matrix is:

1. To the right:

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} \quad (4.1)$$

2. The upward:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{matrix} \quad (4.2)$$

3. To the left:

$$\begin{matrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{matrix} \quad (4.3)$$

4. Down:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & -1 \end{matrix} \quad (4.4)$$

For rotations, the affine matrix is obtained by the cv2.getRotationMatrix2D () function. The parameters of this function are listed as follows:

cv2.getRotationMatrix2D(center, angle, scale) → retval

The parameters are explained as follows:

center: centre of rotation, in this project this parameter is set to the image centre point.

angle: the angle of rotation

scale: isotropic scale factor

After the affine transformation, the image can be used as a frame of video. Before the video is inserted, noise needs to be added. The noise used in this project is salt and pepper noise. This noise works by randomly turning some pixels white or black with a certain probability.

The pseudo code of salt and pepper noise is:

```
---------------------------------------------------------------------------------------------------------
  Input:
  p : the probability of noise
  image: the original image
  output:
  out: the output image with salt and pepper noise
  th ← 1 - p
  for i in 0 to image.shape[0] do
      for j in 0 to image.shape[1] do
          rdn = random number in (0,1)
          if rdn < p
              out [i] [j] ← 0
          elif rdn > th
              out [i] [j]  ← 255
          else
              out [i] [j] ← image [i] [j]
      end
  end
-----------------------------------------------------------------
```

Figure 4.1 shows a frame in the generated video with a noise level of 0.01. The noise in the video is random and different for each frame. The noise level in the data generation stage affects the edge detection ability, which will be discussed in detail in later chapters.

Figure 4.1 A sample frame of the generated video.

4.3 Edge detection

After the data is generated, edge detection will be carried out. This part mainly uses the signal processing related functions in the Numpy expansion package. The discrete Fourier transform function is numpy.fft.fft(). Its input is a time domain signal, the output is a frequency domain signal. The function used to calculate signal similarity is numpy.corrcoef().It receives two signals as input and returns the value of their correlation coefficient.

The pseudo code of edge detection is：

```
---------------------------------------------------------------------------------------------------------
   Input : a video contains n frames, each frame with r rows and c columns
   Output : an edge image
   set a number m < n as the length of signal
   for k in 0 to m do
       image ← the k th frame of the video
   for i in 0 to r do
       for i in 0 to c do
           signal [i] [j] [k] = image [i] [j]
       end
   end
end
   for i in 0 to r
   for i in 0 to c
    calculate correlation coefficient of  fft(signal [i] [j]) and fft (reference edge)
    thresholding the correlation coefficient
   end
end
---------------------------------------------------------------------
```

The two key parameters in this stage are signal length and threshold. How these two parameters affect the results of edge detection will be discussed in the next chapter.

Figure 4.2 shows the edge detection results at the noise level of 0.02, signal length of 60 and the threshold value of 0.73. For comparison, figure 4.3 is the result of canny edge detector at the same noise level.

Figure 4.2 The edge image with the noise level of 0.02, signal length of 60 and the threshold value of 0.73.

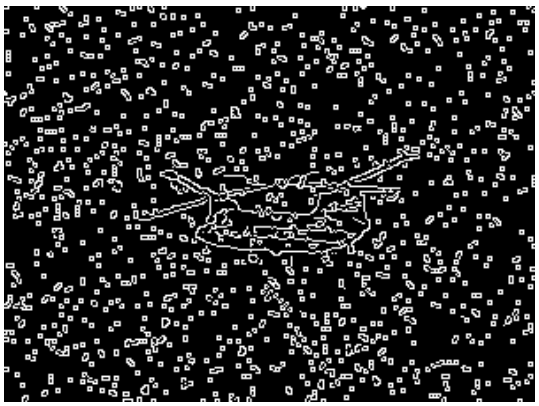


Figure 4.2 The edge image with the noise level of 0.02, signal length of 60 and the threshold value of 0.73.

It can be observed that the edge detection algorithm based on temporary gradient performs better than that based on spatial gradient under the condition of noise. This is because the distribution of noise in space is difficult to distinguish from the edge, but the distribution in time is very easy to distinguish.

4.4 Segmentation

After obtaining the edge image, the work of this step is to segment the edge image. The method of segmentation is morphological closing operation.

To deploy morphological closing operations, use the function cv2.morphologyEx of OpenCV

The parameter list of the function is:

cv2.morphologyEx(src, op, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]]]) → dst

In order to complete morphological closing operation:

src: the input image

op: set op value as MORPH_CLOSE to do the closing operation.

kernel : the structuring element, get by the function cv2.getStructuringElement().

For this method, it is essential to choose the appropriate kernel size. If the kernel size is too small, there will be a void inside the object. If the kernel size is too large, the segmentation result will be larger than the area inside the edge. In both cases, performance declines.

Figure 4.4 shows the segmentation result with the kernel size is 9 × 9. The choice of this parameter should depend on the area inside the edge. More specifically, the value of this parameter is proportional to the area inside the edge.

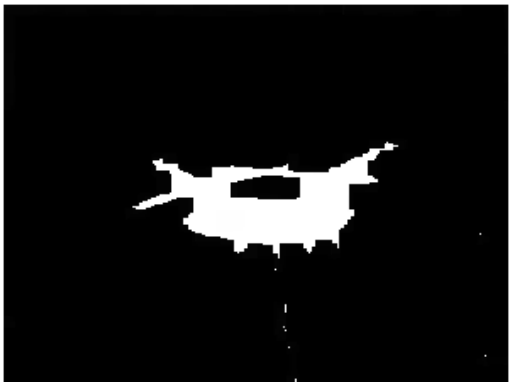


Figure 4.4 The segmentation result with a 9 × 9 kernel size.

All the code is in my github,URL: https://github.com/fragileASH/Msc-project

Chapter 5
Results and Evaluation

5.1 The metrics

To evaluate the results of this experiment, quantifiable metrics is required. The metric used in this project is the intersection over union (IOU). This metric can quantify the proportion of overlap between ground truth and prediction in their total area. Specifically, this metric calculates the total number of pixels that two images intersect divided by the total number of pixels that two images combine.

$$IoU = \frac{\text{ground truth } \cap \text{ prediction}}{\text{ground truth } \cup \text{ prediction}} \tag{5.1}$$

5.2 Results

The average IOU score of this algorithm on the data set is: 0.46081733106240763.

The average IOU score of segmentation based on canny edge detector is: 0.46081733106240763.

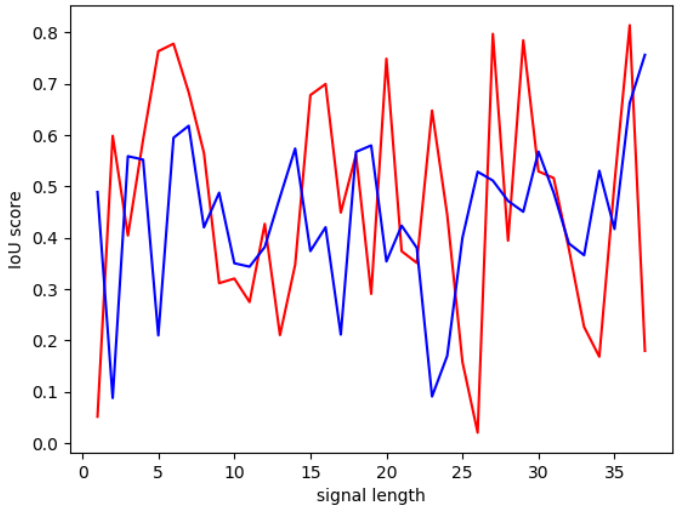


Figure 5.1 comparison of IOU score based on canny edge detection and sapping-based segmentation, the red segmentation is based on sapping-based segmentation, and the blue segmentation is based on canny edge detection

Figure 5.1 comparison of IOU scores based on sapping-based segmentation and canny edge detection, data were obtained from 37 randomly selected images in the data set. Other parameters remained fixed, as follows:

Signal length: 60

Threshold: 0.73

Noise level: 0.02

Kernel size: 9 by 9

Although the performance values of the two methods are close to that of the detector based on canny edge, in fact, the saccade-based segmentation algorithm performs far better than that based on canny edge detector under the condition of noise. This is because the segmentation is often a meaningless white area. In this case, the IOU metric gives a higher score. However, the segmentation results based on canny edge detector are almost all meaningful images, and its performance is poor only in cases where the texture is particularly dense and the intensity changes are not noticeable. Figures 5.2 shows some canny segmentation results with higher scores. Figure 5.3 shows the poor segmentation results based on saccade.

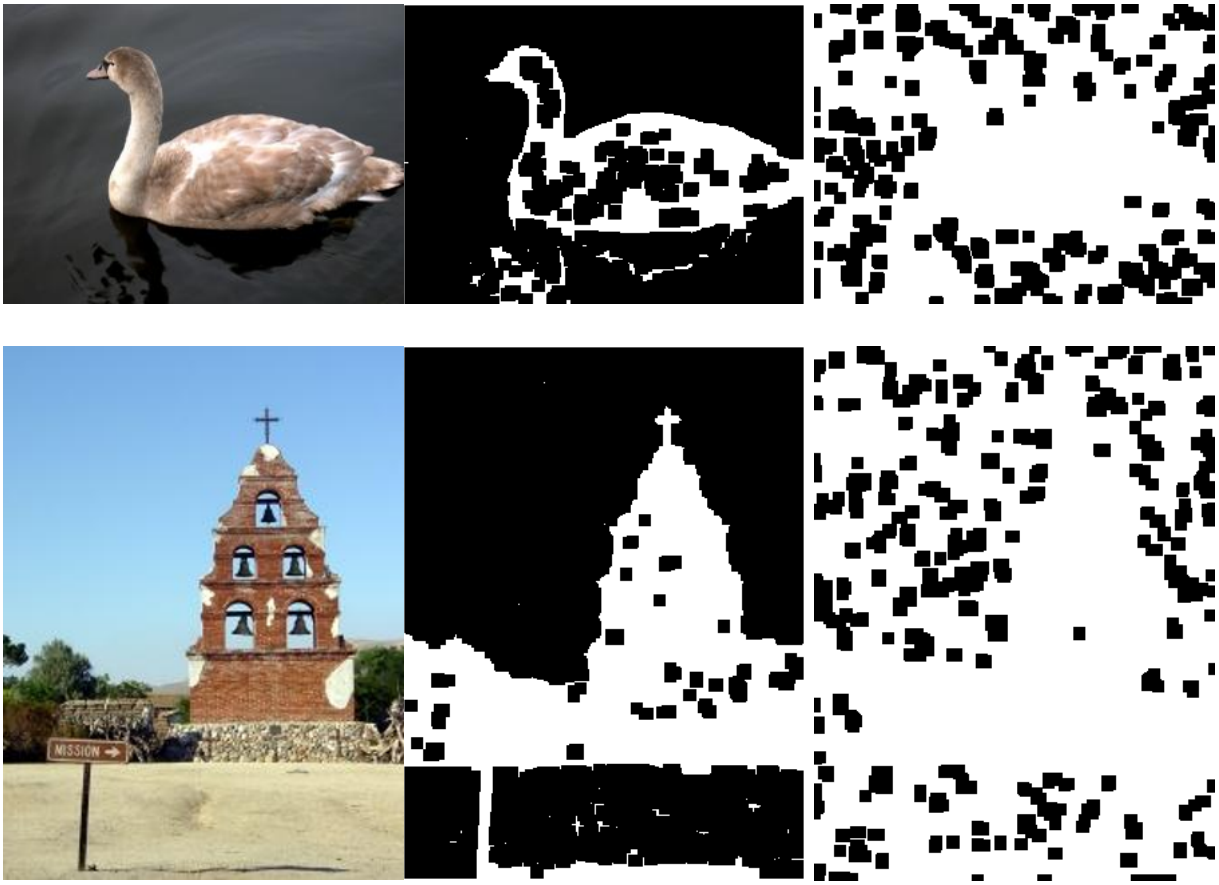


Figure 5.2 The image on the left is the original one, the centre is the saccade based segmentation, and the right is the segmentation result based on canny edge detection. The image on the right has a higher IOU score.

Figure 5.3 The right image is not an abstract painting, but a segmentation of a toad on the similar colour background. The algorithm is completely unable to distinguish colours that close together.

5.3 Testing and evaluation

Intersection over union was used as metric to study the influence of noise level on performance when signal length is 60, kernel size is 9 x 9, and threshold value is 0.75. And compared it with the segmentation results based on canny edge detector. Figure 5.4 shows the comparison results of the experiment.

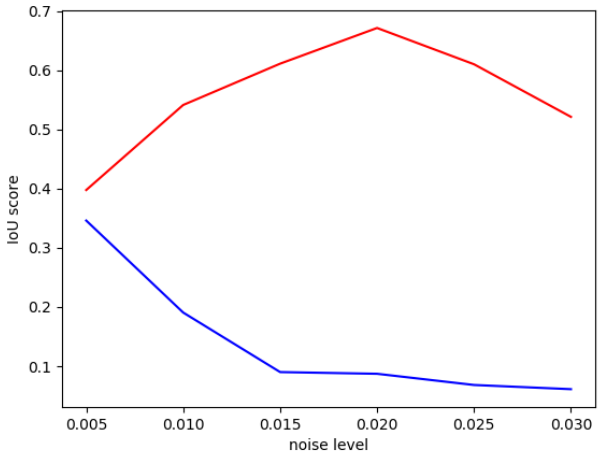


Figure 5.4 Comparison of IOU scores between saccade segmentation method and canny edge detection based segmentation at different noise levels.

Due to the randomness of noise distribution, each parameter configuration was repeated five times during the experiment, and the results were recorded and the mean value was taken as the standard.

By observing at the image, it can be seen that the performance of the two is very similar when the noise level is very low. The segmentation performance of canny edge detector decreases sharply when the noise level increases gradually because canny

edge detector is sensitive to noise. This kind of spatial gradient-based operators is very sensitive to image noise. However, the segmentation performance based on saccade increases. Because the threshold regards the slight intensity difference of background points as edge points under low noise. When the noise level rises, the probability of occasional noise at any location increases, these occasional noises can magnify the differences between the background points and the edge points, making the segmentation performance increase.

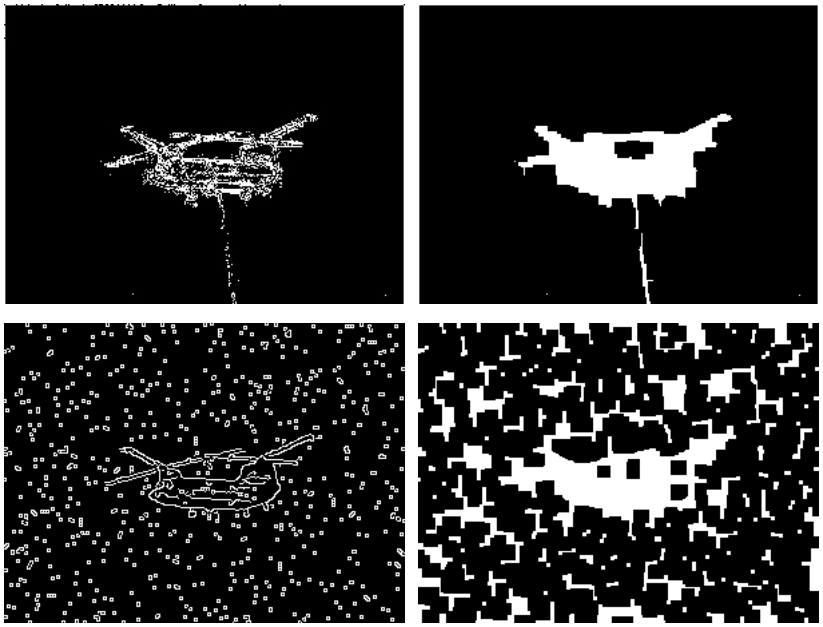


Figure 5.5 Edge image and segmentation results at noise level 0.01. The saccade segmentation is above, and the segmentation based on canny edge detector is below.

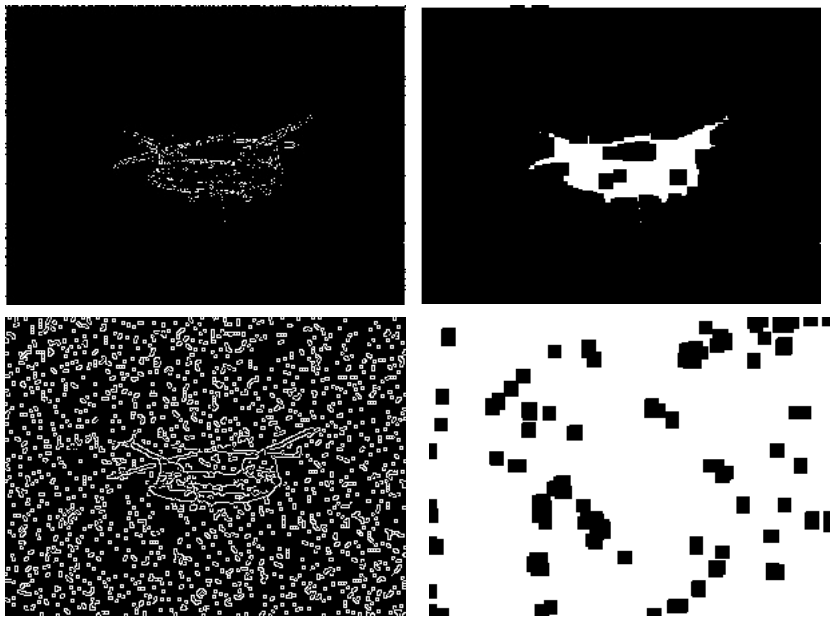


Figure 5.4 Edge image and segmentation results at noise level 0.03. The saccade segmentation is above, and the segmentation based on canny edge detector is below.

Figure 5.5 shows the edge detection and segmentation results of the two methods under noise level 0.01, and figure 5.4 shows the edge detection and segmentation results of the two methods under noise level 0.03

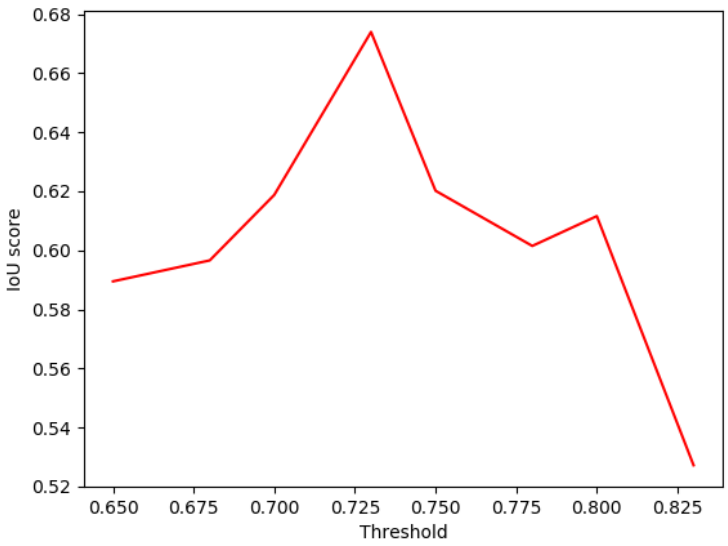


Figure 5.5 IOU scores of different threshold.

Another parameter that determines performance is the threshold. This threshold value refers to the threshold value of the correlation coefficient when judging whether a point is an edge point. Figure 5.5 shows the effect of thresholds on performance with other parameters constant. Signal length is 60, kernel size is 9 x 9, and noise is 0.02.

When the threshold value is low, the algorithm cannot filter out all the noise, so the performance is poor. When the threshold value is high, the algorithm will discard some weak edges, and there are more breakpoints in the edge image. It can also have a negative impact on performance. Figure 5.6 shows the results of edge detection and segmentation of three situation: too low threshold (0.65), suitable threshold (0.73) and too high threshold (0.83).

The length of the signal directly affects the calculation of signal similarity. Because in the case of short signal length, the probability of high similarity of any two signals is higher. However, the increase of signal length can ensure that the similarity expressed by the correlation coefficient is true. Figure 5.7 shows how changes in signal length affect performance when other parameters are the same.

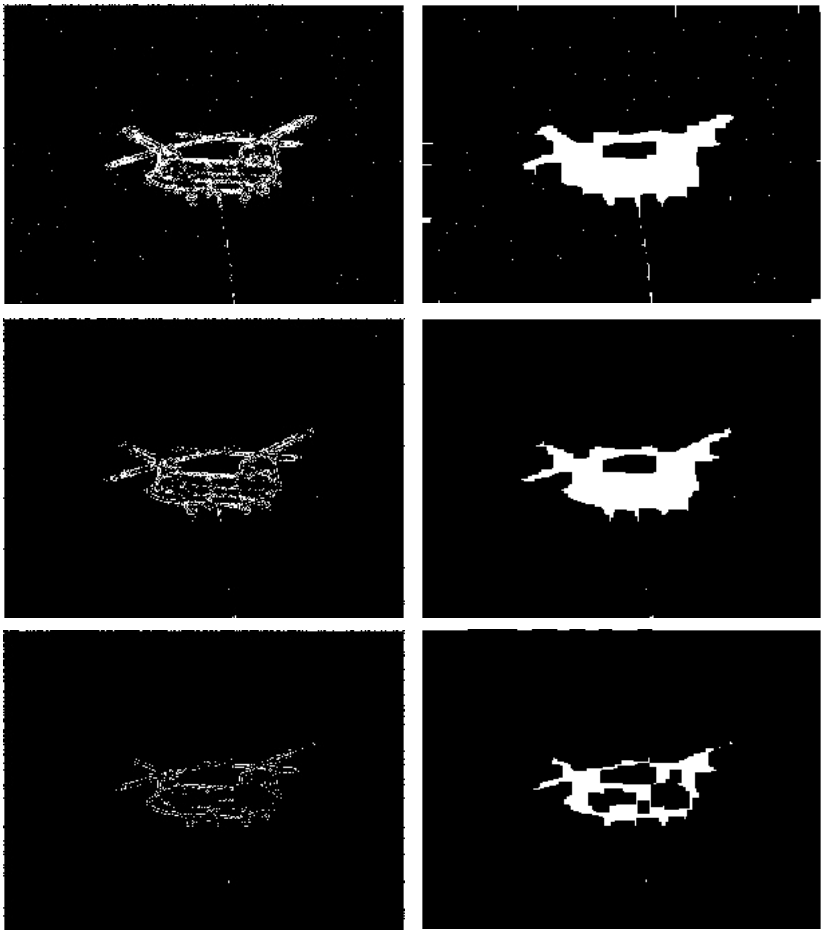


Figure 5.6 From top to bottom, it is the result of edge detection and segmentation in three cases: too low threshold (0.65), suitable for (0.73) and too high threshold (0.83).

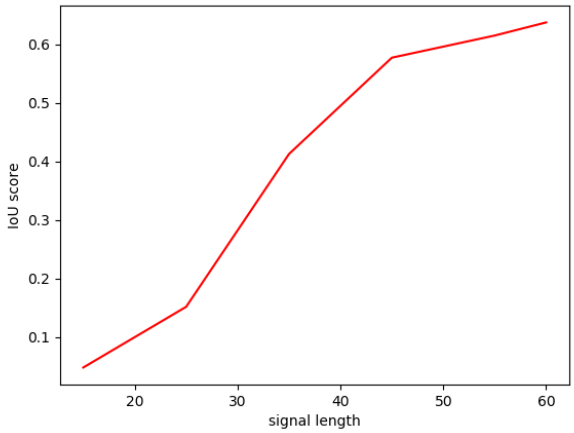


Figure 5.7 IOU scores of different signal length.

Performance is proportional to signal length, which indicates the importance of signal length. Figure 5.8 shows the results of edge detection and segmentation when the signal length is too short and the signal length is suitable.

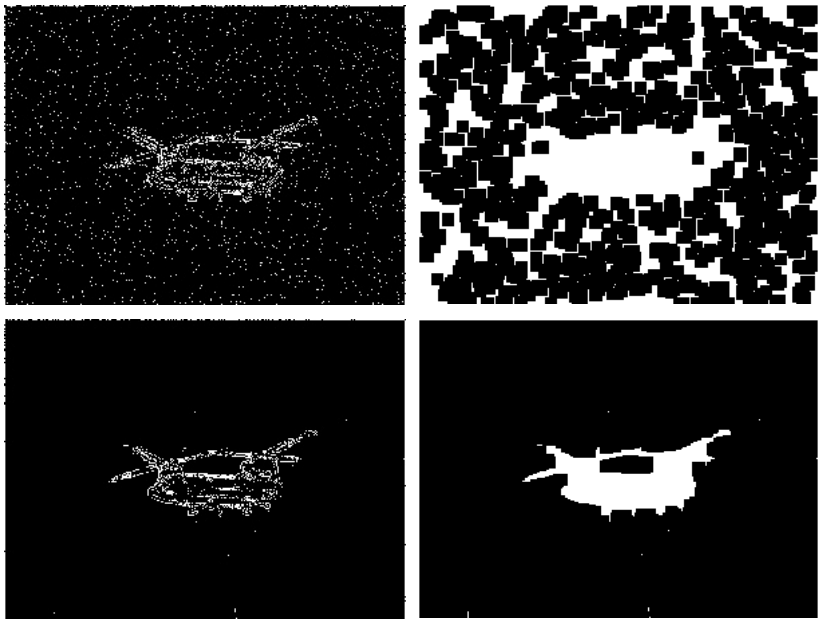


Figure 5.8 From top to bottom are the results of edge detection and segmentation in two cases: the signal length is too short (25) and the signal length is suitable (60)

Chapter 6
Conclusions

6.1 Reviews

This project proves that video data generated by simulated saccade motion can improve image segmentation performance. A reasonable method is proposed to process and analyse video data. Especially in the condition of noise, it has better performance than canny edge detection image segmentation algorithm. The project achieved its planned objectives and generated new ideas along the way. It is a rewarding project experience.

6.2 Limitation

Although the algorithm developed in this project has good segmentation performance under the condition of noise. However, there are limitations.

1. The edges obtained in the edge detection stage are not continuous lines, but edge points.

2. Only one simulated eye movement pattern was verified.

3. Reference edges are pre-set, not automatically acquired or generated.

4. The setting of many parameters, such as segmentation threshold and signal length, needs to be manually set, which is not fully automated.

5. In the image segmentation stage, the size of structuring element needs to be set manually, not automatically. This is a great inconvenience because for different images, this parameter should be set in direct proportion to the area within the edge. Using the same size makes the algorithm perform worse on a data set than it can.

6.3 Future work

If I have more time to devote to the follow-up work of this project, I will expand around the following aspects:

1. Simulate multiple saccade movement modes, which are not limited to four directions, but can be easily extended to eight directions. Alternatively, try more complex patterns, such as adding attention mechanisms that trigger the order of saccades.

2. Try to analyse the simulated eye movements with unknown patterns, which can be done by machine learning algorithms such as support vector machines.

3. In the edge detection stage, I will try to quantitatively analyse the relationship between the signal length of two parameters and the threshold, and analyse the relationship between the optimal choice of two parameters and the image itself. Automatic selection of optimal signal lengths and thresholds is a problem well worth the time.

4. The results of edge detection are discontinuous, which will have a negative impact on segmentation. This problem needs to be addressed.

5. The size selection of structuring element is also a problem to be solved in the segmentation stage, which is preferably proportional to the inner area of the edge.

6. Other edge-based segmentation algorithms should also be tried. Among these methods, there may be more robust segmentation methods or algorithms without setting parameters.

7. Other metrics should be used. The drawback of the IOU is that if you have a large white area, even a meaningless distribution score is very high. Reasonable metrics should avoid such situations, such as giving more weight to the points inside the matched object.

8. When the temporary gradient is obtained, clustering methods such as k-means can be used, which may be more helpful to segmentation.

9. Extend this method to the tracking of moving objects. Temporary gradients are more sensitive to change and are suitable for this type of task.

10. The computation time should be reduced. The current code cannot provide an immediate response. This method converts an image to a video, increasing the amount of information, so it takes extra time to run. However, there are ways to simplify or speed things up.

11. Find alternative ways to calculate the signal similarity, which may be more accurate than the correlation coefficient to express the similarity of two signals.

6.3 self-reflection

During this period of time, I completed this research project, and it is not easy to describe what this project brought to me.

1. In terms of professional knowledge, I have acquired much knowledge in image processing, especially in edge detection and image segmentation, as well as in signal processing. Also, I have read many articles about eye movement in biology. The knowledge is reflected in this graduation thesis and will help me in my future research and work.

2. At the same time, I learned the process of research work from putting forward hypotheses to verifying conclusions, as well as the methods of literature retrieval and reading. These are very helpful for future research.

3. It makes me realize the importance of time management. Excellent time management can make tasks more satisfying.

List of References

Adams, R. and Bischof, L. 1994. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence.* **16**(6), pp.641-647.

Alpert, S., Galun, M., Brandt, A. and Basri, R. 2011. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE transactions on pattern analysis and machine intelligence.* **34**(2), pp.315-327.

Barrow, H.G. and Tenenbaum, J.M. 1981. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence.* **17**(1-3), pp.75-116.

Boncelet, C. 2009. Image noise models. *The Essential Guide to Image Processing.* Elsevier, pp.143-167.

Bracewell, R.N. and Bracewell, R.N. 1986. *The Fourier transform and its applications.* McGraw-Hill New York.

Bruckstein, A., Holt, R.J., Katsman, I. and Rivlin, E. 2005. Head movements for depth perception: Praying mantis versus pigeon. *Autonomous Robots.* **18**(1), pp.21-42.

Burger, H.C., Schuler, C.J. and Harmeling, S. 2012. Image denoising: Can plain neural networks compete with BM3D? In: *2012 IEEE conference on computer vision and pattern recognition*: IEEE, pp.2392-2399.

Canny, J. 1987. A computational approach to edge detection. *Readings in computer vision.* Elsevier, pp.184-203.

Castelhano, M.S., Mack, M.L. and Henderson, J.M. 2009. Viewing task influences eye movement control during active scene perception. *Journal of Vision.* **9**(3), pp.6-6.

Concetta Morrone, M. and Burr, D. 1988. Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London. Series B. Biological Sciences.* **235**(1280), pp.221-245.

Cooley, J.W. and Tukey, J.W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation.* **19**(90), pp.297-301.

Duan, R.-l., Li, Q.-x. and Li, Y.-h. 2005. Summary of image edge detection [J]. *Optical Technique.* **3**(3), pp.415-419.

Findlay, J.M. 1997. Saccade Target Selection During Visual Search. *Vision Research.* **37**(5), pp.617-631.

Freixenet, J., Muñoz, X., Raba, D., Martí, J. and Cufí, X. 2002. Yet another survey on image segmentation: Region and boundary information integration. In: *European Conference on Computer Vision*: Springer, pp.408-422.

Fu, K.-S. and Mui, J. 1981. A survey on image segmentation. *Pattern recognition.* **13**(1), pp.3-16.

Gao, K., Yang, H., Chen, X. and Ni, G. 2008. A robust sub-pixel edge detection method of infrared image based on tremor-based retinal receptive field model. In: *Infrared Materials, Devices, and Applications*: International Society for Optics and Photonics, p.68351Y.

Giefing, G.-J., Janssen, H. and Mallot, H. 1992. Saccadic object recognition with an active vision system. In: *[1992] Proceedings. 11th IAPR International Conference on Pattern Recognition*: IEEE, pp.664-667.

Hayhoe, M. and Ballard, D. 2005. Eye movements in natural behavior. *Trends in cognitive sciences.* **9**(4), pp.188-194.

Hirani, A.N. and Totsuka, T. 1996. Combining frequency and spatial domain information for fast interactive image noise removal. In: *SIGGRAPH*, pp.269-276.

Irwin, D.E. 1991. Information integration across saccadic eye movements. *Cognitive psychology.* **23**(3), pp.420-456.

Kimmel, R. 2003. Fast edge integration. *Geometric Level Set Methods in Imaging, Vision, and Graphics.* Springer, pp.59-77.

Kratz, L. and Nishino, K. 2009. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*: IEEE, pp.1446-1453.

Kuffler, S.W. 1953. Discharge patterns and functional organization of mammalian retina. *Journal of neurophysiology.* **16**(1), pp.37-68.

Lee, J.-S. 1981. Refined filtering of image noise using local statistics. *Computer graphics and image processing.* **15**(4), pp.380-389.

Lee, M.-C. and Chen, W.-g. 1999. *Image compression and affine transformation for image motion compensation*. Google Patents.

Liu, C., Freeman, W.T., Szeliski, R. and Kang, S.B. 2006. Noise estimation from a single image. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*: IEEE, pp.901-908.

Liversedge, S.P. and Findlay, J.M. 2000. Saccadic eye movements and cognition. *Trends in cognitive sciences.* **4**(1), pp.6-14.

Maquet, P., Péters, J.-M., Aerts, J., Delfiore, G., Degueldre, C., Luxen, A. and Franck, G. 1996. Functional neuroanatomy of human rapid-eye-movement sleep and dreaming. *Nature.* **383**(6596), p163.

Marr, D. and Hildreth, E. 1980. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences.* **207**(1167), pp.187-217.

Mastin, G.A. 1985. Adaptive filters for digital image noise smoothing: An evaluation. *Computer Vision, Graphics, and Image Processing.* **31**(1), pp.103-121.

Mignotte, M. 2017. A biologically inspired framework for contour detection. *Pattern Analysis and Applications.* **20**(2), pp.365-381.

Mlsna, P.A. and Rodriguez, J.J. 2009. Gradient and laplacian edge detection. *The Essential Guide to Image Processing.* Elsevier, pp.495-524.

Murray, D.W., Bradshaw, K.J., McLauchlan, P.F., Reid, I.D. and Sharkey, P.M. 1995. Driving saccade to pursuit using image motion. *International Journal of Computer Vision.* **16**(3), pp.205-228.

Muthukrishnan, R. and Radha, M. 2011. Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology.* **3**(6), p259.

Neskovic, P., Cooper, L.N. and Schuster, D. 2002. A recognition system that uses saccades to detect cars from real-time video streams. In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*: IEEE, pp.2162-2166.

Oka, M. and Kurauchi, Y. 1990. *Method and system for image transformation*. Google Patents.

OpenCV. *Geometric Image Transformations.* [Online]. Available from: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html?highlight=warpaffine

Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics.* **9**(1), pp.62-66.

Pierrot-Deseilligny, C., Milea, D. and Müri, R.M. 2004. Eye movement control by the cerebral cortex. *Current opinion in neurology.* **17**(1), pp.17-25.

Pyatykh, S., Hesser, J. and Zheng, L. 2012. Image noise level estimation by principal component analysis. *IEEE transactions on image processing.* **22**(2), pp.687-699.

Rayner, K., Sereno, S.C. and Raney, G.E. 1996. Eye movement control in reading: a comparison of two types of models. *Journal of Experimental Psychology: Human Perception and Performance.* **22**(5), p1188.

Roberts, L.G. 1963. *Machine perception of three-dimensional solids*. thesis, Massachusetts Institute of Technology.

Robinson, D. 1964. The mechanics of human saccadic eye movement. *The Journal of physiology.* **174**(2), pp.245-264.

Róka, A., Csapó, Á., Reskó, B. and Baranyi, P. 2007. Edge detection model based on involuntary eye movements of the eye-retina system. *Acta Polytechnica Hungarica.* **4**(1), pp.31-46.

The Sobel and Laplacian Edge Detectors.

Sonka, M., Hlavac, V. and Boyle, R. 2014. *Image processing, analysis, and machine vision.* Cengage Learning.

Soran, B., Hwang, J.-N., Lee, S.-I. and Shapiro, L. 2012. Tremor detection using motion filtering and SVM. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*: IEEE, pp.178-181.

Sparks, D.L. 2002. The brainstem control of saccadic eye movements. *Nature Reviews Neuroscience.* **3**(12), p952.

Stearns, C.C. and Kannappan, K. 1995. *Method for 2-D affine transformation of images*. Google Patents.

Tai, S.-C. and Yang, S.-M. 2008. A fast method for image noise estimation using laplacian operator and adaptive edge detection. In: *2008 3rd International Symposium on Communications, Control and Signal Processing*: IEEE, pp.1077-1081.

Weisstein, E.W. 2004. Affine transformation.

Xie, J., Xu, L. and Chen, E. 2012. Image denoising and inpainting with deep neural networks. In: *Advances in neural information processing systems*, pp.341-349.

Young, L.R. and Sheena, D. 1975. Survey of eye movement recording methods. *Behavior research methods & instrumentation.* **7**(5), pp.397-429.

Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing.* **26**(7), pp.3142-3155.

Zhao, M., Gersch, T.M., Schnitzer, B.S., Dosher, B.A. and Kowler, E. 2012. Eye movements and attention: The role of pre-saccadic shifts of attention in perception, memory and the control of saccades. *Vision Research.* **74**, pp.40-60.

Appendix A
External Materials

All the extension package used in my project:

time.time
cv2
matplotlib.pyplot
numpy
scipy
skimage
scipy.signal.fftpack
scipy.ndimage.fourier_shift
scipy.signal.correlate
skimage.data
skimage.feature.register_translation
skimage.feature.register_translation._upsampled_dft

And the original version code of data generation and edge detection from Mohamed, see my project github URL: https://github.com/fragileASH/Msc-project

Appendix B
Ethical Issues Addressed

There are no ethical issues involved in this project.