博　士　論　文

Research on stereo matching of sea surface images for long-distance 3D image measurement

知能情報システム工学専攻

楊　英

令和 04 年 3 月

福岡工業大学　大学院

遠距離三次元計測のための海面画像のステレオマッチングに関する研究

知能情報システム工学専攻　楊　英

津波は破壊的な自然災害である。現在の津波計測・到達予測システムには、限られた数の地震計や波浪計を使って地震波や海面の高さを測定し、津波の発生の有無、発生場所、発生時間、海岸に到達する規模などを予測するものがある。しかしながら、限られた数の計測機器を使用するため、大きな予測誤差が生じる可能性がある。この問題を解決するために、本論文ではステレオ視に基づく遠距離津波画像計測システムを提案している。

提案システムでは、海岸にカメラを設置し、4~20km の遠方海面をリアルタイムで撮影し、画像処理に津波発生の有無を判定し、津波発生の場合にはその到着時間と規模を予測する。本研究においては、4 ~ 20 km 遠距離の海面を 24 時間撮影すること、雨や雪などの悪天候下での画像計測、ステレオマッピング、海面高さの算出、津波発生の有無の判定方法、到達時間と規模の推定方法など、多くの課題がある。

本論文では、ステレオマッチングについて研究する。ステレオマッチングでは下記の 2 つの要件を満たす必要がある : 1) 精度要件 : 海面高度を正確に測定する能力、2) 速度要件 : 海面高度の測定を迅速に完了すること。しかしながら、遠距離の海面画像には「視差範囲が大きい」、「異なる波の特徴の相違が少ない」、「海面の非剛体変換」などの特徴があり、ステレオマッチングが困難である。

本論文では、上記のス問題を解決するために、下記の 3 つの側面からアプローチする：(1)特徴ベクトルと決定木による疎なマッチングを行い、高速処理を実現する。(2)リーニングコストボリュームとセミグローバル法による密なマッチングを行い、高精度を目指す。(3)ニューラルネットワークによる疎なマッチングを行う方法を提案し、マッチングの精度を確保する。

また、視差範囲が大きい問題を解決するために、視差 d と y 座標の関係を定式化する。異なる波の特徴が少ない問題を解決するために、ネットワーク構造を構築することで、海面画像の特徴マップを生成する。非剛体変換の問題を解決するために、海の波を区別するための特徴ベクトルを確立し、いくつかの決定論的特徴を敏感に判断し、いくつかの不確実な特徴を無視することで、波をマッチングするための決定木を構築する。

本論文は以下の通り 5 章より構成されている。

第 1 章では、津波計測の背景と本研究の目的を紹介する。

第 2 章では、波の特徴点の検出、記述、マッチング策略の選択という標準的なマッチングパイプラインに基づく、改良型ステレオマッチング手法を紹介する。各ステップは、遠距離の海面画像の特徴に適応するようにうまく設計されています。

第 3 章では、ニューラルネットワークを用いたステレオマッチングのための、学習データセットの作成、ネットワークの構築、ネットワークによる疎なマッチングの方法を紹介する。この章では、特徴ベースのステレオマッチング手法とニューラルネッ

トワークを用いた手法の比較実験を行っている。

第4章では、提案手法の実験結果を示す。実験では、2つの撮影地点から3つの期間内に撮影された海面画像を用いて行い、提案手法の有効性を検証する。

第5章では、この論文をまとめ、今後の課題について述べる。

キーワード：遠距離海波、海波のステレオマッチング、特徴ベクトル、リーニングコストボリューム、ニューラルネットワーク、訓練

西暦2022年03月01日

Research on stereo matching of sea surface images for long-distance 3D image measurement

Doctoral Course of Intelligent Information System Engineering

Yang Ying

Tsunamis are some of the most destructive natural disasters. Some proposed tsunami measurement and arrival prediction systems use a limited number of instruments, then judge the occurrence of the tsunami, forecast its arrival time, location and scale. Since there are a limited number of measurement instruments, there is a possibility that large prediction errors will occur. In order to solve this problem, a long-distance tsunami measurement system based on the binocular stereo vision principle is proposed in this paper.

The proposed system installs two cameras on the coast, takes long-distant image of sea surfaces about 4 ~ 20 km real-time, determines the tsunamis and predicts the arrival time and scale of a tsunami by image processing. In our project, there are many research subjects such as 24-hour image capture for long-distance sea surface of 4 ~ 20 km, image measurement in bad weather such as rain and snow, stereo mapping for binocular stereo vision, calculation of sea level height, method of determining the presence or absence of tsunami, and how to estimate the arrival time.

In this paper, we will focus on the stereo matching method. To achieve tsunami measurement, stereo matching needs to meet two requirements: 1) high precision to accurately perceive sea surface anomalies, 2) fast to achieve rapid perception of sea surface anomalies. However, long distance sea surface images have three main features: 1) large disparity range, 2) lacking feature points, 3) non-rigid transformation, they make stereo matching difficult.

To realize stereo matching of this system, it is accomplished from three aspects: (1) Sparse matching by feature vector and decision tree for high speed; (2) Dense matching using leaning cost volume and semi global method for high accuracy; (3) Feature map generation by neural network to secure the number of matching waves.

In addition, to solve the time and space consumption problem caused by large disparity range, we formulate the relationship between disparity d and the y coordinate. To solve the problem of non-rigid transformation, a feature vector is established to distinguish sea wave from each other and a decision tree is built to matching waves by making judgement sensitive to some deterministic features and ignoring some uncertain features. A non-end to end network structure is built to generate feature map of sea surface image to solve the problem of lacking feature points.

The outline of this thesis is as follows.

Chapter 1 presents the background of tsunami measurement and the purpose of this work.

Chapter 2 introduces the improved stereo matching method based on standard matching pipeline of feature point detection, description and matching strategy selection. Each step is well designed to adapt to long-distance sea surface images' characteristics.

Chapter 3 presents the stereo matching by neural network including the making of training data set, the establishment of non-end to end network as well as feature map generation by the network. The experiments to compare the traditional methods and the network are also conducted.

Chapter 4 presents the experiment results of the proposed method; the experiment was conducted on sea surface images taken within three periods from two sites to verify the efficiency of our method.

Chapter 5 summarizes this thesis.

Keyword *: Long distance sea waves, Stereo matching, Feature vector, Leaning cost volume, Neural network, Deep learning*

Mar. 01, 2022

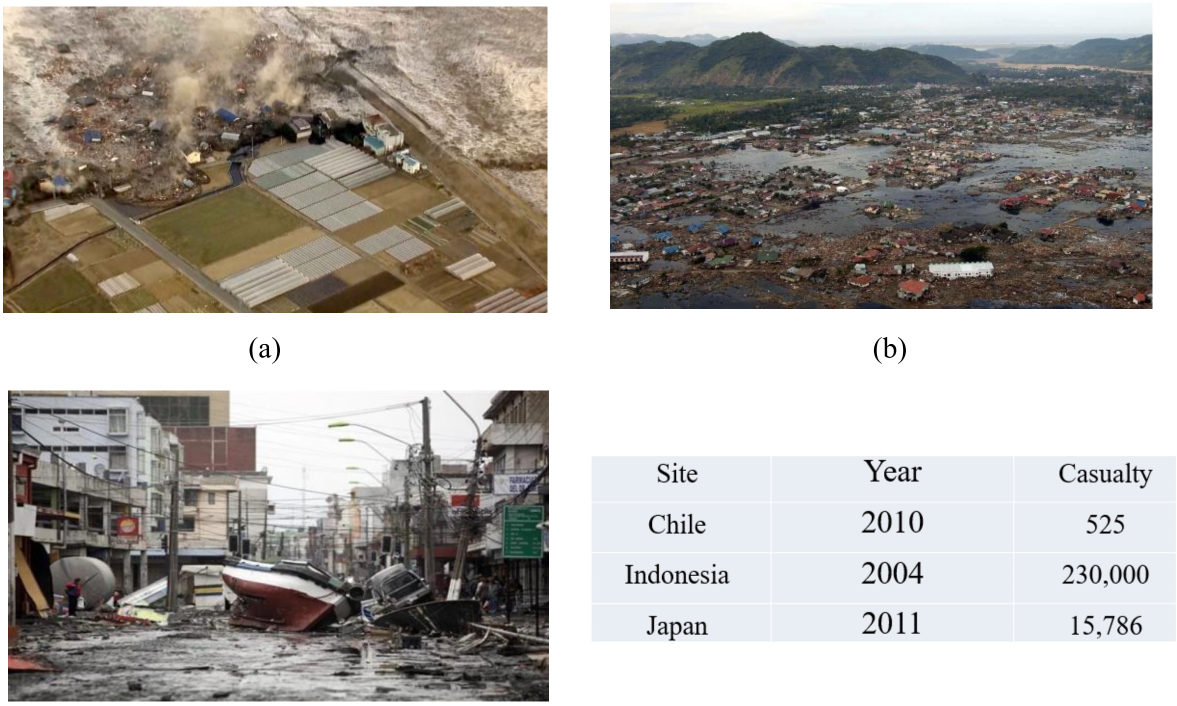
Index

Chapter 1 Introduction

Tsunamis are one of the most destructive nature disasters, it can cause huge casualties and property damage. If we can measure them when they happen and forecast their moving path as well as their onshore magnitudes, many victims' lives maybe saved and some property damage maybe recovered. In this section, we will give a specific introduction about the hazards of tsunami and the existing tsunami forecast system. The motivation and possibility of establishing a tsunami measurement system based on binocular vision principle is also discussed. We will also give a detailed description on the layout of this thesis.

1.1 Background of tsunami forecast

Looking back at human history, there are approximately 260 recorded tsunamis, with an average of one occurring every six to seven years. Fig.1.1 shows three destructive tsunamis happened in Japan, Indonesia and Chile [1-6], the largest one claimed nearly 230,000 lives.



(a)

(b)

(c)

(d)

Site	Year	Casualty
Chile	2010	525
Indonesia	2004	230,000
Japan	2011	15,786

Fig.1.1 Tsunamis and Tsunami Destruction, (a) the 2011 Tōhoku earthquake and tsunami, (b) 2004 Indian Ocean earthquake and tsunami, (c) The 2010 Chile earthquake and tsunami, and (d) the casualty of these three tsunamis.

In order to save lives and protect property, many researchers and organizations started tsunami measurement and forecast study from many years ago. The measurement system can be classified into two categories: on-shore measurement and pre-measurement. For the on-shore measurement, we use the tide gauge to measure the reaching land level height of tsunami. The shortage of this method is that it cannot measure the tsunami in the distance. And for the pre-shore measurement systems, we use the GPS buoys or submarine cables to measure the sea level height off-shore. Among them, the most reliable and have been put into use systems are the Deep-ocean Assessment and Reporting of Tsunamis (DART) system developed by Pacific Marine Environment Laboratory (PMEL) [7] and the Submarine Cable system developed by Japan Meteorological Agency (JMA) [8]. GPS buoys are expensive equipment, costing over 300 million yen per unit when they were first installed in 2005, and now it still cost hundreds of millions of yen, causing it can only be installed in specific spots, making it difficult to observe a wide area. The installation of submarine systems requires a huge amount of construction work and the agreement of local governments and fishery officials. Until now, the observation equipment has been installed at 150 locations along the coast of Japan, and the total cable length is about 5,700 km. In addition, the maintenance and management of the equipment and facilities require a great deal of money. As a result, it is difficult to spread the system in a wide range. Another way to forecast tsunami is that "point measurement but global forecast". These systems take a forecast mechanism of measurement and simulation. They take use of a limited number of seismometers, tide gauges and buoys to measure the information of earthquake and sea level height, then put this information into simulated data base to find out the most similar model and judge if there is a tsunami happening as well as its arrival time, location and scale. Due to the limited number of measurement equipment, there is a large possibility that a large prediction

Table 1. The tsunami warning of the Great March 11th Earthquake in Japan

Time Location	2011/3/11 14:49	2011/3/11 15:14	2011/3/11 15:30	2011/3/11 16:08
Eastern Hokkaido of Pacific Coast	0.5m	1m	3m	6m
Central Hokkaido of Pacific Coast	1m	2m	6m	8m
Western Hokkaido of Pacific Coast	0.5m	1m	4m	6m
Aomori Prefecture of Japan Sea Coast	0.5m	1m	2m	3m
Aomori Prefecture of Pacific Coast	1m	3m	8m	Over 10m
Iwate Prefecture	3m	6m	Over 10m	Over 10m
Miyagi Prefecture	6m	Over 10m	Over 10m	Over 10m
Fukushima Prefecture	3m	6m	Over 10m	Over 10m
Ibaraki Prefecture	2m	4m	Over 10m	Over 10m
Chiba Prefecture	2m	3m	Over 10m	Over 10m

error will occur. For example, Table 1 shows the tsunami warning released by JMA during the Great March 11th Earthquake in Japan, the forecasted sea level heights were revised several times after the earthquake happened in most coastal areas [9].

Tsunami is caused by the displacement of a substantial volume of water or perturbation of the sea [16]. The displacement of the water is usually caused by either the earthquakes, landslides under the sea, volcanic eruptions under the sea, glacier calving or more rarely by meteorites and nuclear tests [17,18]. Japan has built a completed earthquake and tsunami early warning system, it is made up of over 1600 seismometers, 4400 seismic intensity meters, 380 tsunami meters and 27 strain meters, the whole early warning system like the following Fig1.2 shows [19]. We gather earthquake information with seismometers and seismic intensity meters, achieve sea level information with tsunami meters, then put this information into the Data analysis and assembling information center to judge if there is an earthquake happened and if the residents need to prepare for evacuation.

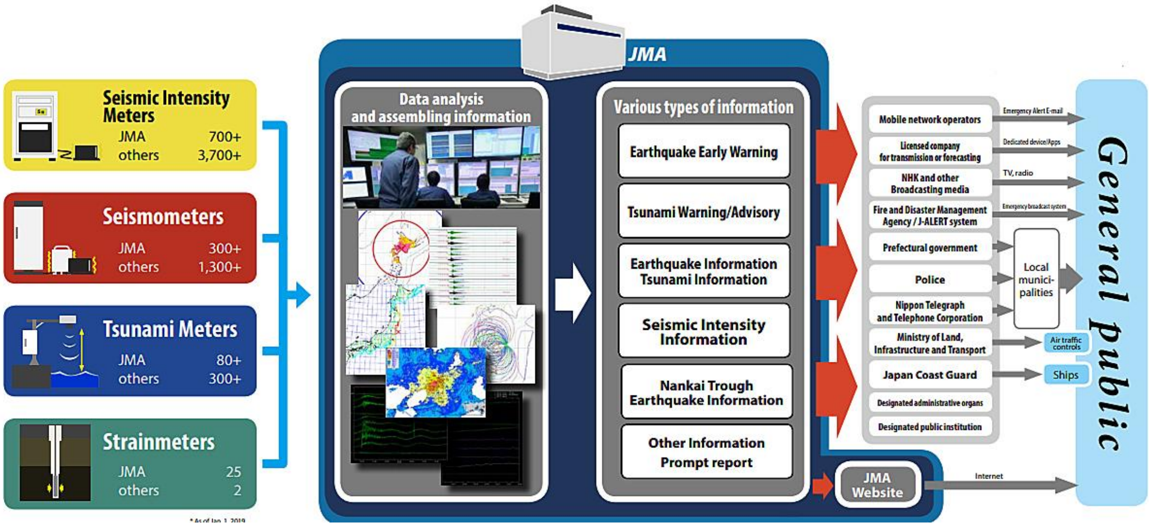

Seismic Intensity Meters
JMA 700+
others 3,700+
Seismometers
JMA 300+
others 1,300+
Tsunami Meters
JMA 80+
others 300+
Strainmeters
JMA 25
others 2
* As of Jan 1 2019
JMA
Data analysis and assembling information
Various types of information
Earthquake Early Warning
Tsunami Warning/Advisory
Earthquake Information Tsunami Information
Seismic Intensity Information
Nankai Trough Earthquake Information
Other Information Prompt report
JMA Website
Mobile network operators
Licensed company for transmission or forecasting
NHK and other Broadcasting media
Fire and Disaster Management Agency / J-ALERT system
Prefectural government
Police
Nippon Telegraph and Telephone Corporation
Ministry of Land, Infrastructure and Transport
Japan Coast Guard
Designated administrative organs
Designated public institution
Emergency Alert E-mail
Dedicated device/Apps
TV, radio
Emergency broadcast system
Local munici-palities
Air traffic controls
Ships
Internet
General public

Fig.1.2 The earthquake and tsunami early warning system.

Tsunami measurement systems such as The German–Indonesian Tsunami Early Warning System and the Tsunami Warning systems in India and Australia also take the similar mechanism [10-12]. But the sparseness and high maintenance cost of these system limited their capability. To complement this disadvantage, other tsunami measurement methods include rapid determination of sea level variations caused by tsunamis and tsunami parameter estimation from Global Navigation Satellite Systems and tsunami detection and forecasting by radar on unconventional airborne craft [13-15] are proposed.

They conducted airborne measurements of sea surface heights (SSH) using an airplane equipped with a nadir-pointing frequency-modulated continuous wave radar and a GNSS receiver. In the real implementation, an additional device is required for real-time data transmissions that are either ADS-B transponder or onboard modem. The SSH can be obtained by subtracting the distance between the airplane and ocean surface measured by the radar from the absolute altitude of the airplane relative to a reference of Earth ellipsoid retrieved by the

GNSS positioning method. However, it is impractical to drive an airborne during 24-hours to monitoring tsunami in the sea area where there is no sea and earthquake measurement equipment. It can only be used as a complementary approach of tsunami measurement.

In summary, for the on-shore measurement, it cannot measure in distance, for the pre-shore measurement, it is expensive to build the measurement equipment in all necessary locations and for the forecast by simulation, it is not precious enough.

1.2 Sea wave measurement based on stereo system

Recently, with the development of computer vision, many scholars have turned towards building 3D geometry of sea waves [20-23]. Wave reconstruction by point measurement of wave gauges cannot provide enough information for space-time wave dynamics. At the same time, Synthetic Aperture Radar (SAR) or Interferometric SAR (1NSAR) remote sensing provide sufficient resolution for measuring waves only at large spatial scales longer than 100 m [24,25]. To address this problem, measurement based on stereo vision attracts researchers' attention, which can provide both spatial and temporal data whose statistical content is richer than that of time series retrieved from wave gauges [26-29]. Wave research based on the stereo system started to become more common after a partially supervised 3D stereo system called Wave Acquisition Stereo System (WASS) was proposed [30]. It is a sea wave 3D reconstruction pipeline, it completely automates all the steps required to estimate dense point clouds from stereo images, including camera calibration, reliable stereo feature matching and mean sea-plane estimation etc., the following Fig.1.3 shows the whole flow chart of this pipeline.

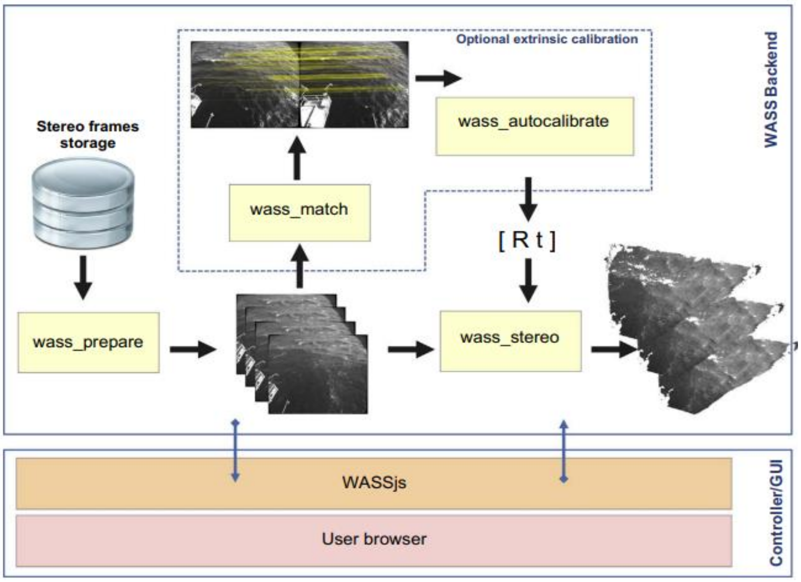


Fig.1.3 Components and workflow of the WASS reconstruction pipeline.

Other seminal reconstruction methods adopted local methods to compute the disparity map. In 2017, Bergamasco et al. proposed an open-source pipeline for the 3D stereo reconstruction of ocean waves. They specifically described all the steps required to estimate dense point clouds from stereo images. The system is mounted 12m above the mean sea level, covering an area of

85 × 65 m^2. Currently, mostly stereo systems for wave acquisition are used to compensate for the lost details of sea waves on small scale, and the 3D reconstruction scope is limited to the sea area near the system. However, the excellent advances in this field of research have given us the inspiration and confidence to build a novel long-distance stereo measurement system for tsunami measurement. With the use of telephoto lens, the measurement range of our proposed system is extended to 4-20km away from the system deployment site, which can meet the requirement of tsunami warning.

1.3 Tsunami measurement based on stereo system

In this paper, a much more small-scale and flexible tsunami stereo measurement system is proposed; it scans the sea surface to increase measurement coverage and aims to measure sea level height in real time within its coverage. Fig.1.4 (a) shows the proposed system configuration. Fig.1.4 (b) shows the data processing flowchart of the system. Firstly, we take sea surface images with the proposed stereo system which has one camera on the left and one on the right, then conduct stereo matching, calculate sea level height according to matching results, and lastly judge if there is a tsunami happening. Stereo matching is one of the key steps, and in this paper, we will focus on the stereo matching method of the proposed system. To realize accurate tsunami measurement, we must reduce stereo matching errors to smaller than eight pixels (we give the reasons for this accuracy requirement in the section of discussion) as well as processing time to less than 1/24s (real time measurement). Sea surface images lack texture and sea water is not still. Additionally, long distance measurement suffers from large disparities in search range. For these reasons, stereo matching in this system is difficult.

The existing stereo matching methods for wave reconstruction stereo systems can be classified into three categories: 1) local methods which suffer the delicate trade-off between the disparity window size (which influences the match localization accuracy) and the required surface smoothness [31-33]; 2) global methods which is so computationally intensive that is

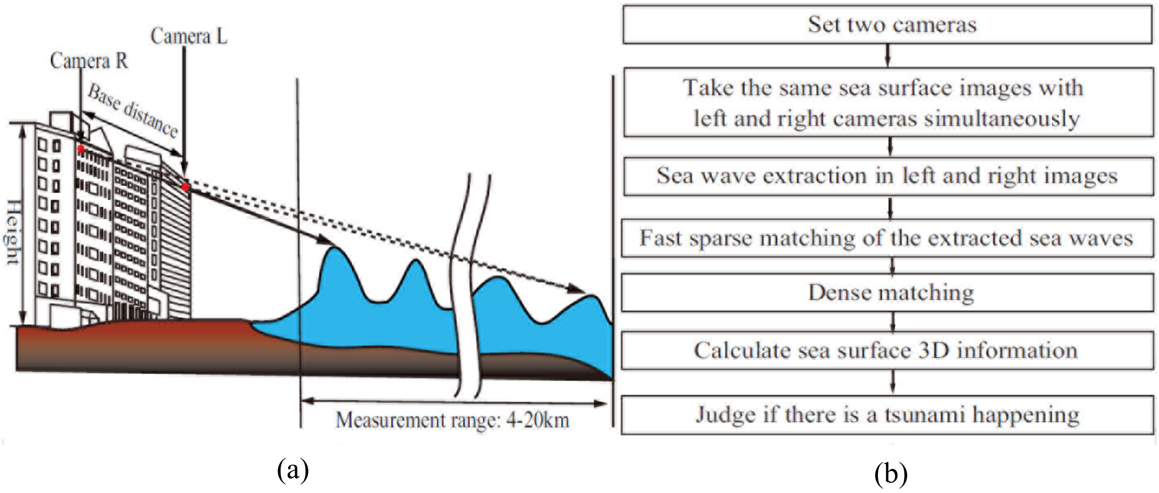

Camera L
Camera R
Base distance
Height
Measurement range: 4-20km
Set two cameras
Take the same sea surface images with left and right cameras simultaneously
Sea wave extraction in left and right images
Fast sparse matching of the extracted sea waves
Dense matching
Calculate sea surface 3D information
Judge if there is a tsunami happening
(a)
(b)

Fig.1.4 Proposed tsunami measurement system, (a)System configuration, (b)Data processing flow chart.

unlikely to be used in practice [34,35]; and 3) semi-global methods which is based on well packaged OpenCV library functions [38] and losses efficacy for long distance sea surface image pairs [36,37].

Recently, stereo matching utilizing deep neural networks has achieved significant advances [39-41]. We are now also working on stereo matching of sea surface images using deep learning network. But there are still some difficulties that remain unsolved, such as the lack of ground truth for supervised networks, low accuracy of unsupervised network results, time and memory consumption, etc. Therefore, the paper focuses on stereo matching using traditional methods. In [42], A Siamese network was built to complete sparse stereo matching of sea surface images. Fig.1.5(a) shows the matching result. However, it cannot be applied to tsunami measurement due to the time consumption, the running time of one pair of stereo images being more than 1 minute.

The key of sparse matching is the selection and description of feature points. Long distance sea surface images are low-texture images. Common feature point detectors, such as Harris, FAST, LoG and DOG can only detect feature points, and are not sensitive enough for low-texture images [43-46]. Thus, for sea surface images, we detect feature regions for sparse matching. An adaptive dynamic threshold method is used to detect sea waves as feature regions [47]. Common descriptors, such as SIFT, SURF, GLOH, DAISY, LIOP cannot be adaptive to the feature region’s size, which makes them difficult to describe the detected sea waves, because sea waves change randomly in shape, size and location [37,46,48-50]. Fig.1.5 (b) shows one of the results of these methods (RANSAC+SURF [37,51]), only well-characterized large waves being correctly matched. Therefore, we propose a new descriptor for sea waves to perform sparse matching. We will describe it in detail in the next subsection.

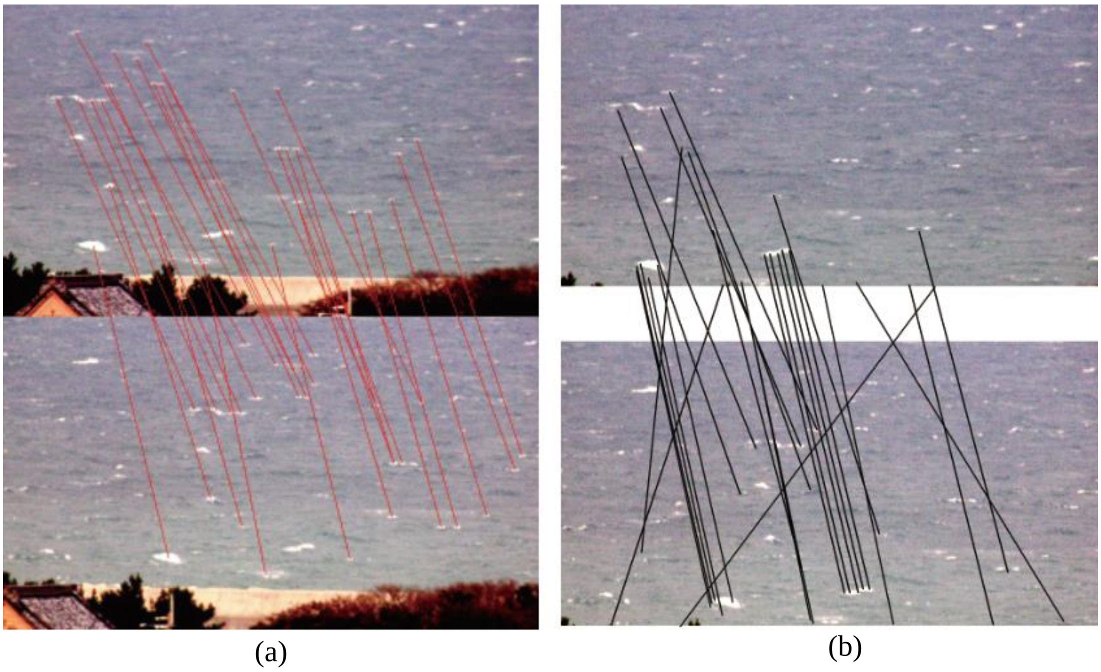


Fig.1.5 E.g. of Sparse matching, (a) result of Siamese network, (b) result of RANSAC+SURF.

The measurement of tsunami requires a smaller than 8 pixels matching error. It is difficult to ensure by sparse matching since it is a region to region matching method. So, the second step (dense matching) needs to be conducted. Dense matching contains: 1) cost computation, such as SAD, MI and NCC, etc. [52], and 2) Cost aggregation, such as SGM, graph cuts and BF [52-54]. Little research on dense matching has studied the establishment of cost volume. The previous algorithms assume that the disparity changes within a constant small range Ds so that the algorithm can find the best matching within limited memory space and computing time. However, for long distance sea surface images, the disparity varies in a range of over 600 pixels. The traditional cost volume will lead to greater than 4GB consumption of memory space. To solve this problem, this paper proposes a leaning cost volume based on the first step sparse matching result. We will give a detailed description in the next section.

1.4 Objective and main work of this research

This work mainly researched the stereo matching method for long distance sea surface images for tsunami measurement. Different from common stereo matching task, long distance sea surface images are lack of feature points, low texture and suffer from large disparity between left and right camera's view fields. These characteristics make stereo matching for them be very difficult. Four types of stereo matching methods are discussed in this thesis, experiment was conducted to compare the efficiency of these methods.

To solve the problem of lack distinguishable feature points, a feature vector is proposed to describe sea wave on the sea surface image and the sea wave is taken as feature region for sparse matching of sea surface images. During this process, another importance step is the matching strategy. Due to the different shooting angles and extraction thresholds, the same sea wave's feature vectors are not exactly alike from the left and right camera's field of view. A decision tree is established to match the sea waves which can make judgement based on analyzing some certain features and ignoring some uncertain features.

Region-to-region sparse matching only takes sea waves into consideration and it loses most of the information on the sea surface images. In order to take use of all the information on the sea surface images and realize 3D reconstruction of sea surface, the second step dense matching is necessary. As sea surface images suffer from large disparity problem. We cannot build cost volume directly for dense matching, because the large disparity will cause the cost volume consuming large space in computer memory and the aggregation of the cost volume will take long time than common task. Thus, we formulate the relationship between disparity d and the coordinate y to rapidly decrease the size of cost volume by proposing a leaning cost volume method. A dynamic penalty acquisition method is proposed to improve the precision of leaning cost volume and speed the aggregation progress.

With the development of neural network and its excellent performance in computer vision, semantic recognition etc., we also explored the stereo matching of sea surface images by neural network. A supervised network structure is established to complete sparse matching of sea surface images, a feature map generation module is proposed to replace the descriptor in traditional method. We also made a training set for this network based on the previous matching work. An end-to-end network is established to directly extract sea waves from sea surface image. The training data set is established based on our former extraction research. We train the matching module based on a single object tracking network. By the establishment of a alignment module, we can rapidly reduce the matching searching range. A refine module is also added to improve the final matching precision. As it is a challenge of network for long distance image stereo matching, it cannot be taken as a replacement of traditional method, but we hope the work of us can be a valuable exploration for other researchers.

The experiment is conducted on over 300 pairs of stereo images taken at two different sites: Fukuoka Institute of Technology and Fukuoka Kenritsu Suisan High School, with three monitoring distance: 4-10 km, 14-20 km and 8-14 km, during three period. Sea surface images were taken from sunrise to sunset, and experiment is only carried out on the images taken by the visible cameras.

1.5 Structure of this thesis

The arrangement of chapters is as follows:

Chapter 1 is the introduction, it introduced the background of tsunami forecast, including the forecast system in use and some methods research as a compensation of existing system. The configuration of our proposed system is also introduced in this chapter as well as the development of stereo system for sea wave reconstruction which gives us encouragement to build such a long-distance measurement stereo system. Finally, the chapter arrangement was introduced briefly.

Chapter 2 mainly introduced the improved stereo matching method based on standard matching pipeline. First of all, we give a detailed definition of stereo matching, then the designed elements used in our stereo matching process are introduced such as: feature points/regions extraction method, feature points/regions description method, cost computation method, cost aggregation method and the selected matching strategy. Then, the sparse matching and dense matching are conducted respectively.

Chapter 3 presents the stereo matching by neural network, including the making of training set and the construction of supervised network for sparse matching and sea wave extraction. The definition of loss function is also introduced in this chapter as well as an alignment module for decrease the cost volume.

Chapter 4 shows the experiment results. The specific configuration of one of our experiment systems is presented. Although there are several experiment results at the end of chapter 2 and 3, we give more general experiment results in this chapter to verify the efficiency of the proposed methods.

Chapter 5 concludes this thesis with the main contributions summarized.

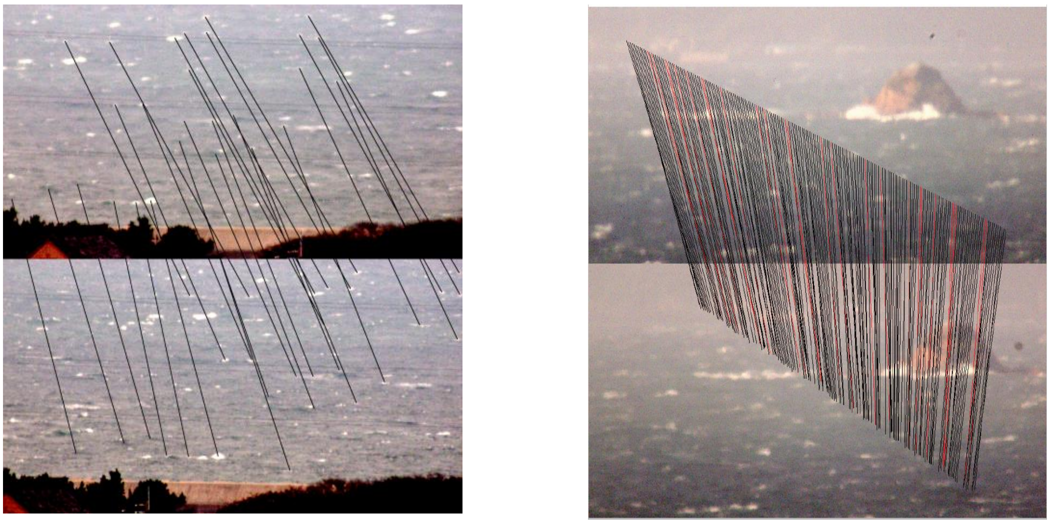
Chapter 2 Stereo matching by improvement of traditional method

In this section, we will introduce the traditional stereo matching method. It is one of the key steps for binocular vision system to realize 3D measurement by image processing. There are many well-designed algorithms used for stereo matching and with the increasement of storage and computing capability, deep learning has been greatly developed in the field of stereo matching. However due to the shortages of deep learning, it cannot replace the traditional methods yet, thus, in this section, we will still focus on the stereo matching based on traditional image proposing method.

We will start by the definition of stereo matching, then introduce the proposed feature description method as well as the selected matching strategy. The relationship between disparity and y coordinate is also formulated in this section, which is used to decrease the searching range of stereo matching. The experiment results are also partial displayed to validate the proposed method.

2.1 Introduction of stereo matching

As its name implies, the stereo matching is to find the pixels of a stereo system views that corresponding to the same 3D point in the scene. It is one of the core technologies in computer vision and recovers 3D structure from 2D images [61]. It has been widely used in areas such as autonomous driving, augmented reality and robotics navigation. In this paper, we expand the definition of stereo matching and defined that finding the same object in the views of a stereo system that corresponding to the same 3D object is also called stereo matching. We also defined that matching the objects is the sparse stereo matching and matching the pixels is the dense stereo matching. Fig. 2.1 illustrates the definition of sparse stereo matching and dense stereo matching. (a) is the sparse matching, it only finds the objects in stereo image pairs corresponding to the same sea waves and connects the them with black lines; (b) is part of the dense matching we only show the matching results of pixels on one line, it finds pixels



(a)

(b)

Fig.2.1 Illustration of stereo matching, (a) sparse stereo matching, (b) dense stereo matching.

corresponding to the same 3D pixels in the scene and connects them with lines, the black line is correct matching and the red line is the incorrect matching.

By observing the right and left images, it is easy for us to find out the objects/pixels corresponding to the same 3D objects/pixels. But as one of the core technologies of computer vision, we need to design algorithm to complete stereo matching. Simulating the process of manual matching, stereo matching algorithm need to realize three functions: 1) feature point/region extraction; 2) feature point/region description; 3) matching the same object/point by selected matching strategy.

For sparse and dense matching，the matching strategies are different. For sparse matching, the descriptor achieved is usually feature vector. Matching is conducted by searching the most similar feature vectors. Euclidean distance or Normalized Cross Correlation (NCC) are usually used to search the most similar feature vectors. In SIFT algorithm, the matching strategy should choose the most similar feature vectors with 128 dimensions, support vector machine (SVM) and k-nearest neighbors algorithm (KNN) are usually used. For dense matching, we usually take winner-take-all criteria as matching strategy, it means that we take the disparity which minimize the cost value as the final disparity value. Before we choose the matching strategy, we should give an introduction of cost value, it is calculated by the cost computation progress, and cost aggregation is conducted to remove the noise caused in this progress.

Cost is computed at each pixel by summing the intensity difference and gradient difference etc. for this pixel and its possible matched pixel point. Cost aggregation is usually conducted by some filters such as median filter, bilateral filter etc.

2.2 Feature point extraction

Feature point extraction is a connective term for feature point and region extraction, it is an importance step in sparse stereo matching process. Before matching the pixels or objects from one stereo image pairs of the same 3D pixel or object, the most beginning task is to find out the location of pixel point or object from the original images.

Common extraction methods can be divided into two categories: 1) corner detection by some well-designed operators to detect the corner, edge point etc. of the targets on the image as feature points, such as Harris, FAST, LoG and DOG operators [43-46]; 2) region detection by some dynamic threshold operator to detect all the points of the target on the image as feature regions, such as OTSU, gaussian adaptive threshold, mean adaptive threshold and block threshold [47,56,64,65]. In the next section, we will give a detailed introduction of these operators.

2.2.1 Corner detection

Corner detection utilizes well-designed operators to detect the corner as feature point. In this section, we will give a detailed introduction of Harris corner detector to help the reader understand feature point detection well.

A corner is a point that pixel intensity changes rapidly in two mutually perpendicular directions. Thus, to detect a corner we can sum the intensity changes around its neighborhood and then judge if it changes rapidly in two mutually perpendicular directions. The sum of intensity changes in the direction (u, v)can be calculated by the following equation:

$$E(u,v) = \sum_{x,y}^{M,N} w(x,y)\big(I(x+u,y+v) - I(x,y)\big)^2 \quad (2.1)$$

Here, the $I(x+u,y+v)$ is the pixel intensity of point $(x+u,y+v)$, $I(x,y)$ is the pixel intensity of point (x,y), M, N are the height and width of neighborhood size, $w(x,y)$ is the weight of different pixel point.

For the edge points, the gradient only has a large variation in the direction perpendicular to the edge; For the flat area points, the gradients do not have a variation; While for the corner points, the gradients have large variations in both mutually perpendicular directions. As shown in the Fig.2.2, the red circles indicate the corner locations, and the green circles indicate the corner detection results. We find that for graphs such as circles and large obtuse angles, the gradient variation in the two mutually perpendicular directions is small, so they are eliminated first in the corner detection process (the corner detection results depend on the parameter selection, if the parameter constraints are relaxed, circles and large obtuse angles can also be detected, but compared with line intersections or right angle points they are more unlike corner points).

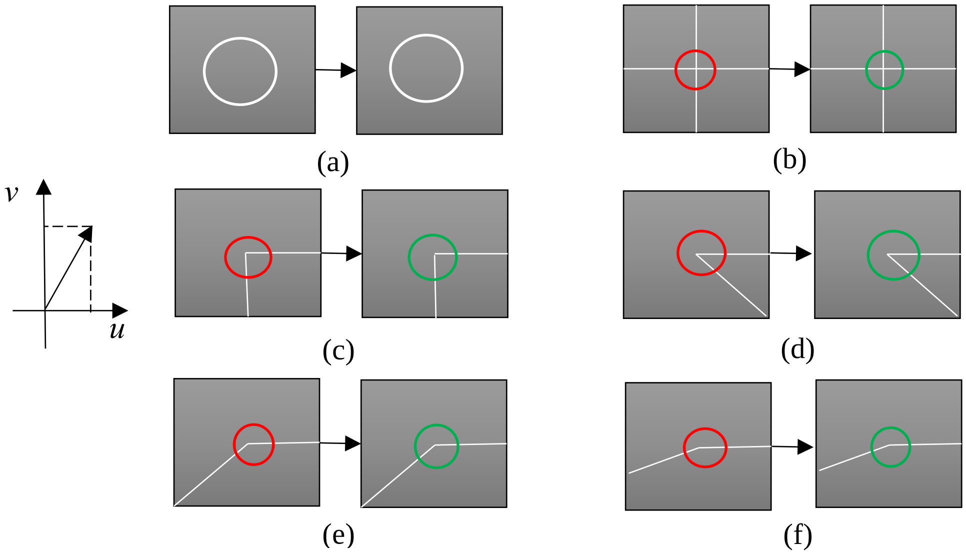


Fig.2.2 Illustration of corner detection, (a) (b) (c) (d) (e) (f) show different kinds of pixel points and their corner detection results, (a) is circle point, intensity changes small in two directions and no corner point detected, (b) is line intersection, intensity changes large in two directions and was detected as corner, (c) is right angle, intensity changes large in two directions and was detected as corner，(d) is sharp angle, intensity changes large in two directions and was detected as corner, (e) obtuse angle, intensity changes large in two directions and was detected as corner, (f) large obtuse angle, intensity changes small in two directions and no corner was detected.

To measure the intensity change, we use the Taylor series expansion, and can get the following equation:

$$E(u,v) = [u \quad v]\begin{bmatrix} \sum_{x,y}^{M,N} w(x,y)I_x^2 & \sum_{x,y}^{M,N} w(x,y)I_xI_y \\ \sum_{x,y}^{M,N} w(x,y)I_xI_y & \sum_{x,y}^{M,N} w(x,y)I_y^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} \quad (2.2)$$

We set $M = \begin{bmatrix} \sum_{x,y}^{M,N} w(x,y)I_x^2 & \sum_{x,y}^{M,N} w(x,y)I_xI_y \\ \sum_{x,y}^{M,N} w(x,y)I_xI_y & \sum_{x,y}^{M,N} w(x,y)I_y^2 \end{bmatrix}$,it is easy to find out that this is a real symmetric matrix, and it has two normalized eigen vectors α, β and eigen values λ_1, λ_2. Projecting $[u \quad v]$ onto the two normalized eigen vectors α, β, as $\begin{bmatrix} u \\ v \end{bmatrix} = u'\alpha + v'\beta$, we can get the following equation:

$$\begin{aligned} \mathrm{E(u,v)} &= (\mathrm{u}'\alpha + \mathrm{v}'\beta)^T M(\mathrm{u}'\alpha + \mathrm{v}'\beta) \\ &= (\mathrm{u}'\alpha + \mathrm{v}'\beta)^T \cdot (\lambda_1\mathrm{u}'\alpha + \lambda_2\mathrm{v}'\beta)^T \qquad (2.3) \\ &= \lambda_1\mathrm{u}'^2 + \lambda_2\mathrm{v}'^2 \end{aligned}$$

Thus, E(u, v) is a function of (u', v'), the change of (u', v') will influence the value of E(u, v). At the same time, there is a linear relationship between (u', v') and (u, v), it means that the change of (u, v) will cause the change of (u', v'), and finally influence the value of E(u, v). Observing equation (2.3), we can conclude three situations: 1) when λ_1, λ_2 are relatively large in absolute values, the small changes of (u, v) can cause E(u, v) be large; 2) when λ_1, λ_2 are relatively small in absolute values, the small changes of (u, v) cannot cause E(u, v) be large; 3)when there is one of λ_1, λ_2 being large, E(u, v) will be large only in the direction corresponding to the large eigen value.

To judge if a point is a corner or not, we must ensure that the two eigen values are relatively large at the same time. Therefore, we define a function to measure the large degree like the following equation shows:

$$\begin{aligned} corner &= det(M) - k * trace(M) \\ &= \lambda_1 * \lambda_2 - k(\lambda_1 + \lambda_2)^2 \qquad (2.4) \end{aligned}$$

The larger of λ_1, λ_2, the larger of $corner$. For each point in the image, we calculate its corresponding matrix M, then calculate the det and trace of M to figure out its *corner* value, the

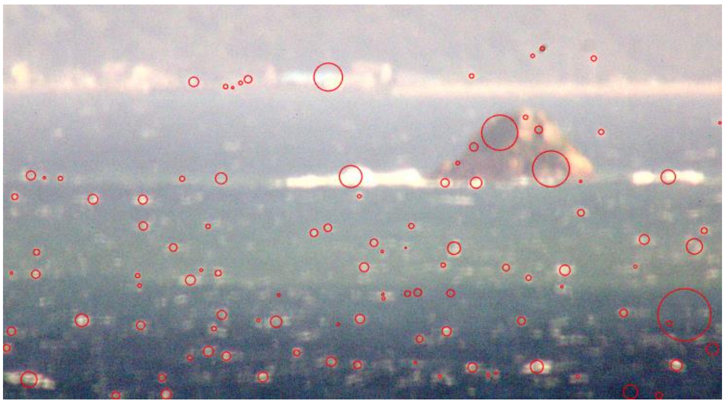


Fig.2.3 corner detection result of sea surface image.

corner points will output large *corner* values, with an appropriate threshold, we can detect all the corners in the image. Fig.2.3 shows one of the corner detection result of sea surface image.

2.2.2 Region detection

Region detection can find out all the pixel points on the object, compared with single point, region is much more stable. Common extraction methods include automatic threshold algorithm, dynamic threshold algorithm and reginal growth algorithm etc. In this section, we will give a detailed introduction of single threshold algorithm: OTSU and a dynamic threshold algorithm to help read have a good understanding of region detection.

Otsu algorithm [56] is an automatic threshold acquisition method proposed in 1979, it's a global single threshold binaryzation method, dividing the image into foreground and background according to maximum between-cluster variance.

Assume the gray level of image is S, the pixel number of *i* gray level is n_i, total pixel number is $Total = \sum_{i=0}^{S} n_i$, the probability of *i* gray level is $p_i = \frac{n_i}{Total}$, assume threshold is *t*, thus all pixels are divided into two class,

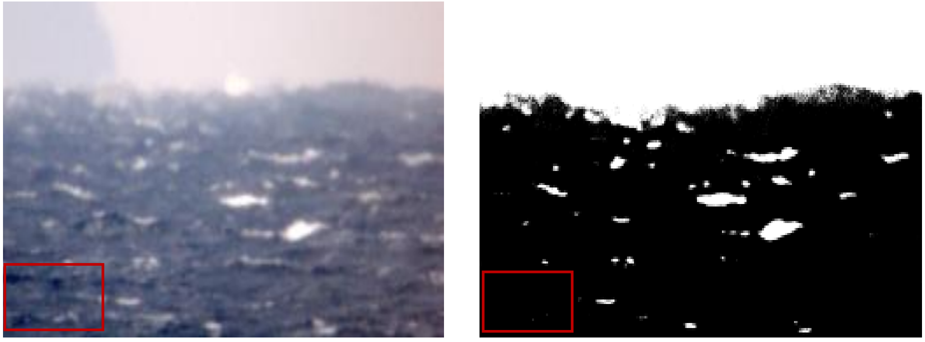
$$C_0 = \{1,2,\dots,t\}, C_1 = t+1, t+2, \dots, S \tag{2.5}$$

The average gray level is $mean = \sum_{i=0}^{S} i \times p_i$, average gray level of C_0 class is $mean_0 = mean_t / w_t$, and the average gray level of C_1 class is $mean_1 = (mean - mean_t)/(1 - w_t)$ in this formula, $mean_t = \sum_{i=0}^{t} i \times p_i$, $w_t = \sum_{i=0}^{t} p_i$.

Define between-cluster variance as:

$$\sigma^2(t) = (mean - mean_0)^2 \times w_t + (mean - mean_1)^2 \times (1 - w_t) \tag{2.6}$$

t ranges from 1 to S-1 to find the most appropriate value making $\sigma(t)^2$ maximum, that is the threshold dividing image into foreground and background, it's a method of automatic threshold acquisition and can meet the automation requirement of our system. But it cannot solve the problem of ununiformed illuminance problem of long-distance sea surface image. Fig.2.4 shows one of the extraction result of OTSU algorithm.



(a) (b)

Fig.2.4 Extraction result of OTSU algorithm, (a) is the original image and (b) is the extraction result of OTSU algorithm.

Red boxes show the extraction results when sea waves are darker than sea surface, few darker sea waves are extracted out. From the result, we conclude that Otsu algorithm can only extract several sea waves out when sea waves are obviously brighter than sea surface, and it loses efficacy when sea waves dark than sea surface.

A simple way to solve this problem is decrease threshold when sea waves are dark. It is easy to think of using dynamic threshold for sea wave detection. Dynamic threshold method [55] is automatic adaptive threshold algorithm, computing each pixel threshold by its surroundings. Like Fig.2.5 shows. Firstly, calculate the intersection thresholds by Otsu algorithm automatically, then calculate each pixel (x, y) threshold $t_{x,y}$ by formula (2.7):

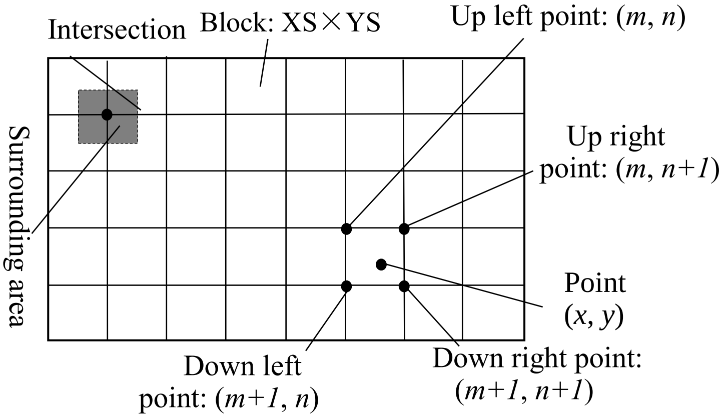


Fig.2.5 Threshold selection of dynamic threshold method

$$t_{x,y} = (1-p)*(1-q)*t_{m,n} + p*(1-q)*t_{m,n+1}$$
$$+(1-q)*p*t_{m+1,n} + p*q*t_{m+1,n+1} \tag{2.7}$$

In this formula, assume (XS, YS) is the size of block, thus $m = floor\left(\frac{x}{XS}\right)$, $n = floor\left(\frac{y}{YS}\right)$, $p = \frac{y}{YS} - floor\left(\frac{y}{YS}\right)$, $q = \frac{x}{XS} - floor\left(\frac{x}{XS}\right)$.

Fig.2.6 shows the extraction result of dynamic algorithm. The left one is extraction result, and right one is partial enlarged image within the red box. We find the rectangle noise is eliminate, and most sea waves are extracted out, but it causes other incorrect extraction like yellow boxes shows. It is because the luminance is not uniformed and intensity is complex with brighter and darker sea waves. To extract all the sea waves out by this method will cause the threshold selected for intersection in that area small, thus lead to threshold for each pixel smaller than required.

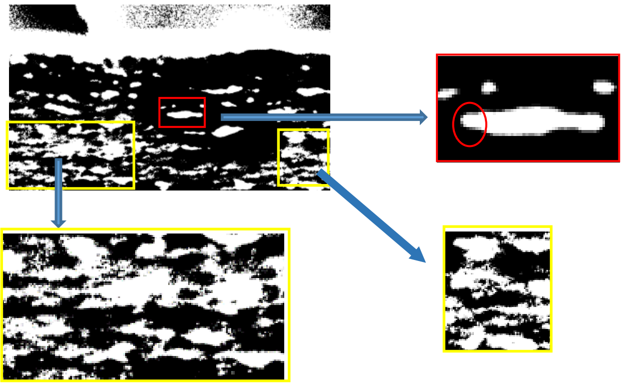


Fig.2.6 Extraction result of dynamic threshold

2.2.3 Sea wave detection

Sea wave extraction is the foundation of sea wave measurement, which plays a significant role in binocular stereo vision-based tsunami early warning system. However, long distance sea surface image possesses two typical features: ununiform illuminance and low signal to noise ratio that makes it difficult to extract sea waves out from sea surface image and lead to incorrect extraction. In this section, we improved Otsu algorithm to realize dynamic multi threshold algorithm for sea wave extraction.

Firstly, dividing the original image into small blocks and figuring out dynamic multi threshold according to local sea surface illuminance. Then defining an evaluation function to add constraint condition to adjacent block's threshold. Finally, extracting sea waves out using the block's threshold in all the blocks. By experimental results the positive extraction rate improves by 11.1%, compared with Otsu algorithm.

It's a multi-threshold method through improving Otsu algorithm. Because Otsu algorithm [56] is an automatic threshold acquisition method, which divide the image into foreground and background by one global threshold. It can meet the automation requirement of our system. But it cannot solve the problem of ununiformed illuminance problem of long-distance sea surface image, thus, we divide the original image to small blocks and define an evaluation function to add constraint condition to adjacent blocks' thresholds and modes.

Before introduction of our proposed method, we explain the effects of variables(t, φ), which is the basic of sea wave extraction. As Fig.2.6 showing: t represents the threshold and is easy to understand. f is the original image, it is a gray level image, with four gray levels: background is 255, three objects' gray value are 200, 140, 80 respectively. φ represents the mode during extraction process, when φ is equal to 0, the gray value larger than threshold is set to 0, smaller than threshold is set to 255 like the first row shows.

When φ is equal to 1, the gray value larger than threshold is set to 255, and smaller than threshold is set to 0, like the second row shows. $F(t, \varphi)$ represents its binarization result under threshold t and mode φ. Our proposed method adds constraint condition to blocks' t and φ parameters' selection process.

There are two kinds of sea waves on sea surface image: one brighter than sea surface, and the other one is darker than sea surface, like Fig.2.7 shows. (a) is the original image, (b), (c) are two different kinds sea waves. Obviously, the extraction modes for (b), (c) should be different to keep the background consistent: (b) is 0, (c) is 1. (d) and (e) are the extraction results. There is no method mentioned before taking mode choosing into account, thus, they cannot achieve ideal extraction result.

To remove incorrect extraction caused by independent threshold and mode selection process of each block, we define an evaluation function to add constraint condition to blocks' threshold and mode selection process, each small block's threshold and mode depend on its adjacent blocks' thresholds, modes and its own illuminance.

Fig.2.8 is the flowchart of sea wave extraction, the first, second and forth steps are the same as direct blocking method mentioned in subsection 2.2.2, and our proposed method focuses on third step: rearrange blocks' thresholds and modes by evaluation function. We will give definition of evaluation function next.

Fig.2.9 is one of image taken by our system, interest block $B(i,j)$: $XS \times YS$, and its adjacent blocks $B(i,j-1)$, $B(i-1,j)$ along X axis, Y axis are marked respectively, S_v and S_h are its overlap areas with vertical adjacent block $B(i-1,j)$ and horizontal adjacent block $B(i,j-1)$. Our goal is to rearrange the threshold and mode of the interest block $B(i,j)$ referring to the thresholds and modes of its adjacent blocks $B(i,j-1)$ and $B(i-1,j)$, based on the Otsu algorithm threshold of the interest block $B(i,j)$.

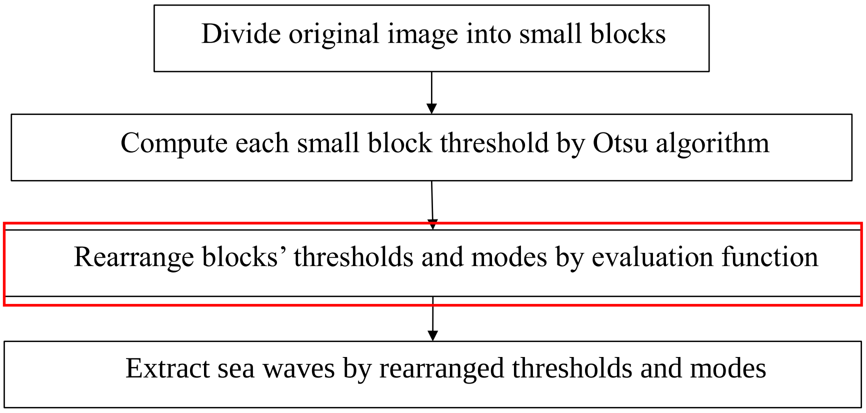

Divide original image into small blocks
Compute each small block threshold by Otsu algorithm
Rearrange blocks' thresholds and modes by evaluation function
Extract sea waves by rearranged thresholds and modes

Fig.2.8 Image processing flowchart.

Assume f as the original image, $F(k_t, k_\varphi)$ is its binaryzation result by threshold k_t, mode k_φ. $t_{i,j}$, and $\varphi_{i,j}$ are threshold and mode of interest block $B(i,j)$, computed by traditional Otsu algorithm independently, $\hat{t}_{i-1,j}, \hat{\varphi}_{i-1,j}$ represent rearranged threshold and mode of block $B(i,j)$. Formula (2.8) shows the definition of evaluation function.

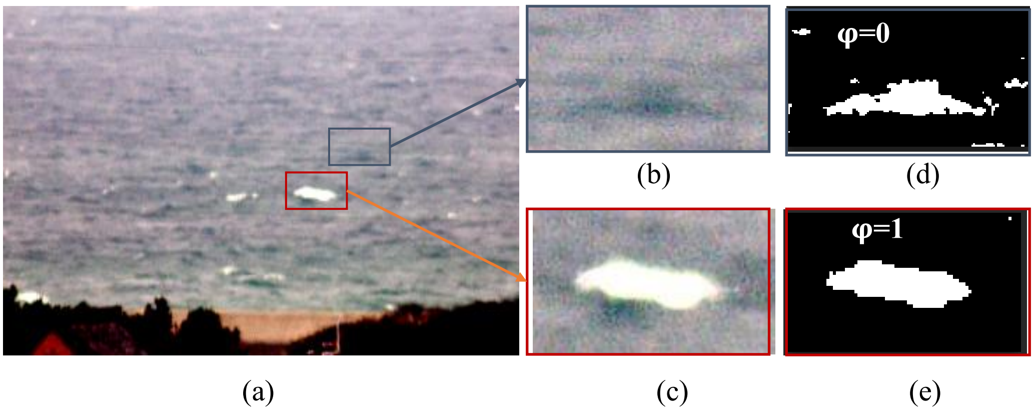


Fig.2.7 Two kinds of sea waves and their extraction results, (a) is the original image, (b) and (c) are two types of sea waves, (d) and (e) are their extraction results by proposed method.

$$g(k_t, k_\varphi) = w_s * S(k_t, k_\varphi) + w_d * D(k_t, t) \tag{2.8}$$

w_s, w_d are the weights of $S(k_t, k_\varphi)$ and $D(k_t, t)$. For interest block $B(i,j)$, the similarity item $S(k_t, k_\varphi)$ calculates similarity between overlap area S_h and S_v's two binarization results, one is binarized by variable threshold k_t mode k_φ and the other one is binarized by $\hat{t}_{i,j-1}, \hat{\varphi}_{i,j-1}$ and $\hat{t}_{i-1,j}, \hat{\varphi}_{i-1,j}$ respectively, the larger this item is, the better elimination of smooth rectangle edge. The similarity $S(k_t, k_\varphi)$ is computed by formula (2.9):

$$S(k_t, k_\varphi) = \frac{\sum_{x=1}^{c1} \sum_{y=1}^{d1} F_{x,y}(k_t, k_\varphi) \odot F_{x,y}(\hat{t}_{i,j-1}, \hat{\varphi}_{i,j-1})}{c \times d} + \frac{\sum_{x=1}^{c2} \sum_{y=1}^{d2} F_{x,y}(k_t, k_\varphi) \odot F_{x,y}(\hat{t}_{i-1,j}, \hat{\varphi}_{i-1,j})}{e \times f} \tag{2.9}$$

In this formula:

1) $c1 \times d1$, $c2 \times d2$ are the sizes of S_h and S_v respectively, as Fig.2.9 shows, $c1 = stepx$, $d1 = YS$, $c2 = XS$, $d2 = stepy$.

2) $F_{x,y}(t, \varphi)$ is S_h and S_v binarization result at pixel point (x, y), under threshold t, mode φ.

3)$a \odot b$ =1, when $a = b$, $a \odot b = 0$, when $a \neq b$.

4) k_t, k_φ are variables, $\hat{t}_{i,j-1}, \hat{\varphi}_{i,j-1}$, $\hat{t}_{i-1,j}, \hat{\varphi}_{i-1,j}$ are the thresholds and modes of adjacent block $B(i, j-1)$, $B(i-1, j)$ respectively, computed by our proposed method before $\hat{t}_{i,j}, \hat{\varphi}_{i,j}$. With the variety of k_t, k_φ, the value of $S(k_t, k_\varphi)$ changes.

The second item calculates the reciprocal of distance between $t_{i,j}$ and k_t, since Otsu algorithm tends to extract the most sea waves on sea surface image, the larger this item is, the more sea waves will be extracted out. The reciprocal of distance D is computed by formula (2.10):

$$D(t_{i,j}, k_t) = 1 - \frac{|k_t - t_{i,j}|}{t_{i,j} + \varepsilon} \tag{2.10}$$

In this formula, k_t is the variable, $t_{i,j}$ is the threshold computed by Otsu algorithm, ε is a value to avoid that denominator equals to 0.

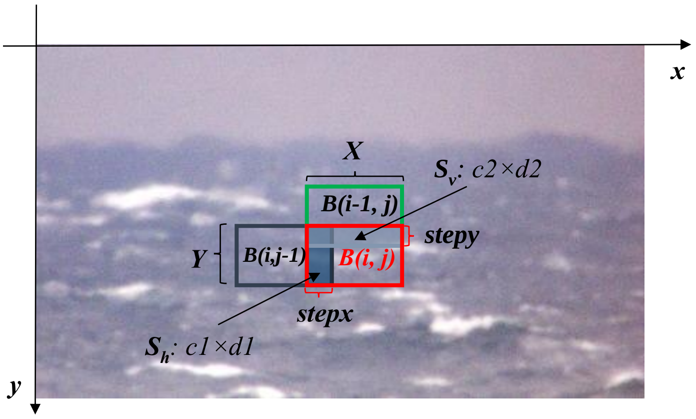


Fig.2.9 Explanation of adjacent blocks and overlap areas.

Our improved Otsu method aims to find out the most appropriate values of k_t, k_φ to maximize evaluation function, and take them as the rearranged threshold and mode $\hat{t}_{i,j}, \hat{\varphi}_{i,j}$ of block $B(i,j)$, that is:

$$\hat{t}_{i,j}, \hat{\varphi}_{i,j} = argmax\left(g(k_t, k_\varphi)\right) \tag{2.11}$$

To speed up processing of $\hat{t}_{i,j}, \hat{\varphi}_{i,j}$ achieving by solving formula (2.11), the variable k_t is limited within $[min\{t_{i,j}, \hat{t}_{i,j-1}, t_{i-1,j}\}\ ,\ max\{t_{i,j}, \hat{t}_{i,j-1}, t_{i-1,j}\}]$

φ can be 0 and 1. We use computer program to find most appropriate solution of formula (2.11).

Finally comes the experiment results of our proposed algorithm. Fig.2.10 shows the extraction results of our proposed method. (a), (c) are two types of sea surface images, taken at different time from different place, (b), (d) are their extraction results of our proposed method. Yellow boxes show the extraction results when sea wave is divided into different small blocks, by rearranging threshold and mode. Red boxes show the extraction result when sea waves are darker than sea surface, by rearrange threshold and mode, incorrect extraction (showing in Fig. 2.4) caused by dynamic threshold method can be eliminated. The green boxes show that darker and brighter sea waves can be extracted with different modes to keep the consistency of background by our proposed method. We also find there are few brighter sea waves in (c), if it is binarized by single threshold or only extract brighter sea waves, (we can only detect and match several sea waves to compute sea surface 3D coordinates and all the darker sea waves are removed) it is not enough for practical application. However, by our proposed method, we can extract most sea waves regardless of their luminance, like (d) shows.

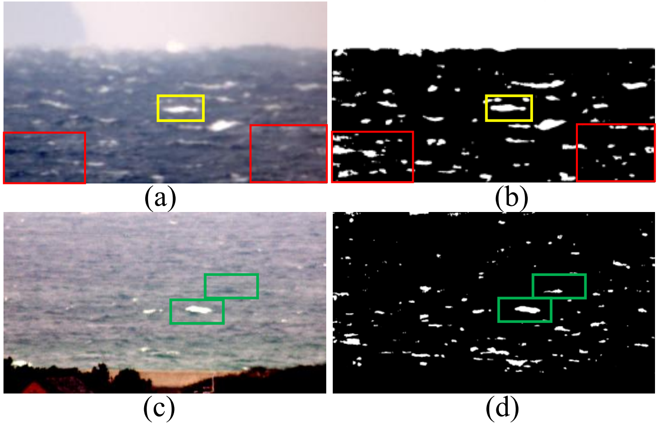


Fig.2.10 Extraction results, (a) and (c) are two types of sea surface images, (b) and (d) are their extraction results.

2.3 Descriptor of feature point and region

Feature descriptor is one of the most important steps for image tracking and classification. Good descriptor should be discriminative, robust and easy to compute. The raw pixel values such as color, gradient and filter responses are the simplest choice for image features, and were used for many years in computer vision. However, these features are not robust to illuminance change and nonrigid motion. There are many descriptors being designed to decrease the influence caused by illuminance change, nonrigid motion etc., such as SIFT, HOG, LBP etc. According to the region size where the descriptor achieves features from, we can divide descriptors into two categories: 1) local feature descriptor and 2) global feature descriptor.

In this section, we would like to give some simple introduction of sever feature descriptors to help reader have a good understanding of this concept and then give a detailed introduction of the feature vector proposed by us for sea wave description. We think that a well understanding of feature descriptor can also help reader understand the following feature map of network.

2.3.1 Local feature descriptor

Local feature descriptor encodes interesting information from the neighborhood of feature point into a series of numbers which is invariant under image transformation. As the region size is limited to the neighborhood of a feature point/region, we call it as local feature descriptor. This kind of descriptors can also be divided into two types based on its ability to self-adapt to object size: 1) fixed size descriptor and object size adaptive descriptor.

Some common descriptors such as Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG), DAISY etc. are fixed size descriptor, they generate features from a fixed size of neighborhood. We will give a detailed description of HOG descriptor to show a direct illustration of fixed size.

The essential thought behind the HOG descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions [66].

Now, I will show the key steps to achieve hog feature of sea waves, it includes four steps.

Firstly, it is the image normalization. To normalize the illuminance of original image, we transfer the original RGB image to grayscale image according to the following equation (2.12), and operate gamma correction on the achieved grayscale image according to formula (2.13).

$$gray = 0.3 * Channel_R + 0.59 * Channel_G + 0.11 * Channel_B \tag{2.12}$$

$$I'(x, y) = I(x, y)^{\gamma} \tag{2.13}$$

Gamma correction can solve the problem of ununiform illuminance by increasing or decreasing the whole brightness of the image, when $\gamma > 1$, the whole illuminance is increased, and when $\gamma < 1$, the whole illuminance is decreased.

Then, we need to calculate the gradient of the image. After illuminance normalization, we calculate vertical gradient map $G_y(x, y)$ and horizontal gradient map $G_x(x, y)$. To simplify computation process, we convolve the normalized image with vertical and horizontal gradient operators: $[-1 \quad 0 \quad 1]^T$, $[-1 \quad 0 \quad 1]$. Then calculate the magnitude and direction of each pixel point according to equation (2.14):

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

$$\theta(x, y) = \arctan \frac{G_y(x,y)}{G_x(x,y)} \tag{2.14}$$

Here, $G(x,y)$ is the magnitude of gradient at pixel point (x,y), $\theta(x,y)$ is the direction of gradient.

Thirdly, operate weighted voting orientation histogram. After generating gradient magnitude and its orientation at each point, we gather histogram of gradient orientation. Firstly, divide the image into small cells, no overlap with each other, then sum the gradient magnitudes of the same orientation bin within each small cell. The following image Fig.2.11 shows the definition of orientation bins. We divide the 360° orientation into 9 different bins, and take them as x axis of histogram, take the accumulated value of gradient magnitude within each bin as y axis.

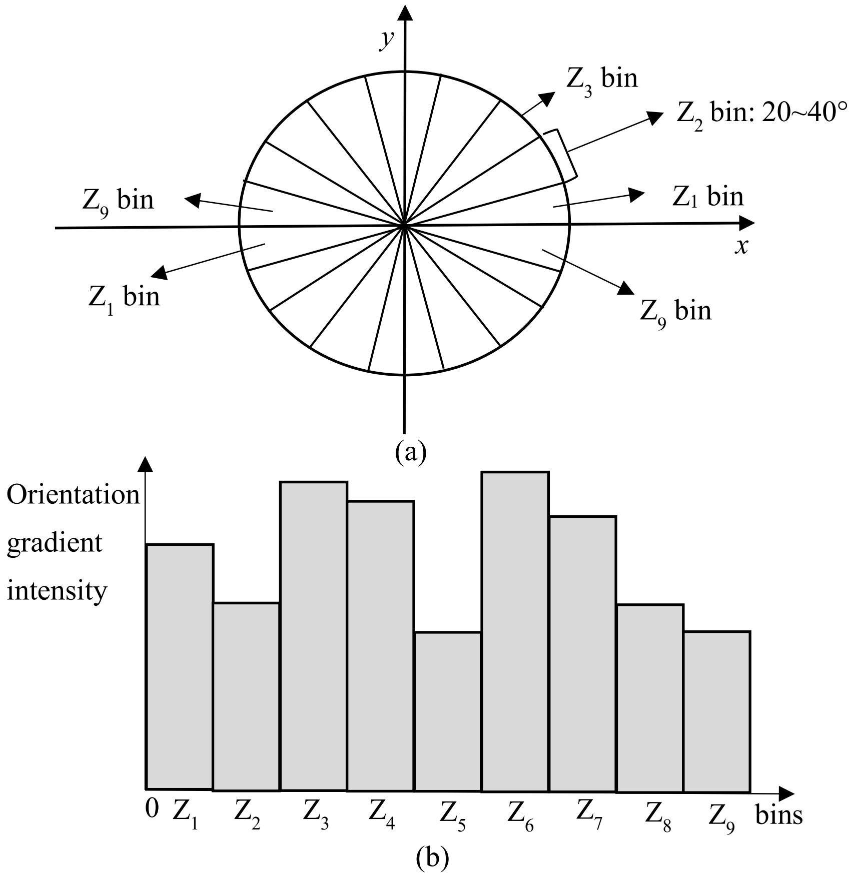


Fig.2.11 Set up histogram of orientation gradient within one cell, (a) is the illustration of bins, (b) is the orientation gradient histogram.

Finally, block histogram normalization. Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance. We evaluated a number of different normalization schemes. Most of them are based on grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window. In fact, we typically overlap the blocks so that each scalar cell response contributes several components to the final descriptor vector, each normalized with respect to a different block. This may seem redundant but good normalization is critical and including overlap significantly improves the performance.

In this thesis, we use square or rectangle blocks, in each block, there are four small cells, unite each cell's histogram to form the block's histogram, there are four different block normalization methods, (a) *L2-norm*, $v \to v/\sqrt{\|v\|_2^2 + \varepsilon^2}$v; (b) *L2-Hys*, *L2-norm* followed by clipping (limiting the maximum values of v to 0.2) and renormalizing; (c) *L1-norm*, $v \to v/(\|v\|_1 + \varepsilon)$; and (d)*L1-sqrt, L1-norm* followed by square root $v \to \sqrt{v/(\|v\|_1 + \varepsilon)}$, which amounts to treating the descriptor vectors as probability distributions and using the Bhattacharya distance between them. As *L2-norm* is simple and can generate relatively good result, we use *L2-norm* to normalize block histogram.

For an image generated by our system, Fig.2.12 shows the block select process. Each block is made up of four cells, each block has overlap area with its adjacent blocks. White rectangular boxes are the small cells, red, yellow, green and blue rectangular boxes are four adjacent blocks, the whole image is divided into 8*6 cells, four cells make up one block, there are 9 orientations in one cell, so combine four cells' orientation, we get the block's hog feature vector, its length is 4*9, from the former image, we find it is consisted of (8-1)*(6-1) blocks, therefore, the length of image hog feature is 7*5*9*4=1260.

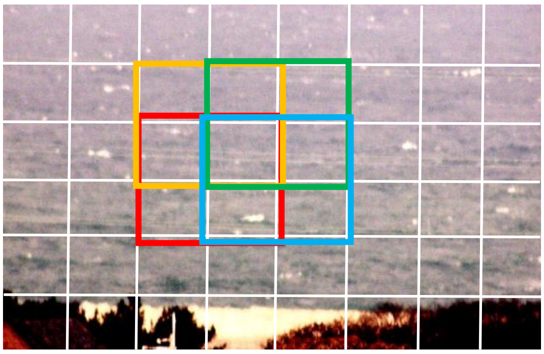


Fig.2.12 cells and blocks of hog feature.

During this process, the fixed size of neighborhood means that the sizes of small cell and block are fixed, it cannot be adaptive to the object's size. Thus, we believe that it is an excellent descriptor for feature points, but for different objects with different sizes, it will loss efficiency to describe the objects, which we think it is much more stable than single point. To make use of object size feature, some size adaptive operators are designed, such as Local Intensity Order Pattern (LIOP), Scale adaptive region covariance descriptor [67], Adaptive Local Descriptor Embedding Zernike Moments [68] etc. We will give a detailed introduction of LIOP to explain the meaning of size self-adaptability.

The basic principle of LIOP is that the relative order of pixel intensities remains unchanged when the whole intensity of image changes. The size of the neighborhood is achieved by scale inspection. The following Fig.2.13 shows the whole workflow of LIOP descriptor generation, it includes: 1) defining the detection region, 2) normalizing region, 3) region division and 4) counting feature histogram.

Firstly, the original image is smoothed by a gaussian filter to remove noise, then a feature point detector is used to localize the feature position and scale. Then we divided the region into several subregions based on the method proposed by Fan [69]. Specifically, all the pixels in the local patch are first sorted by their intensities in a non-descending order. Then, the local patch is equally quantized into B ordinal bins according to their orders. Fig.2.13(c) gives an illustration of such an intensity order-based region division where each ordinal bin is marked with a different color. Note that it is not only invariant to monotonic intensity changes and image rotation, but also contains much more spatial information than the ring-shaped region division.

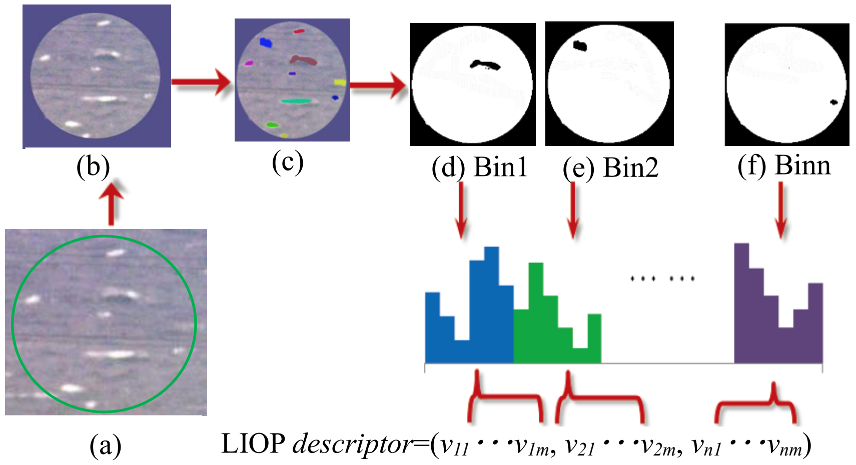

(b)
(c)
(d) Bin1
(e) Bin2
(f) Binn
…… …
(a)
LIOP $descriptor=(v_{11}\cdots v_{1m}, v_{21}\cdots v_{2m}, v_{n1}\cdots v_{nm})$

Fig.2.13 the workflow of LIOP, (a) is the detect region, (b) is the normalized region, (c) is the region division result, (d), (e) and (f) are the extracted sea waves.

The size self-adaptability means that the scale of object is inspected before we describe the feature point/region, the neighbor where descriptor collects information from is adaptive to the scale. For sea wave description, we also designed a size adaptive descriptor, a detailed introduction of it is in subsection 2.3.3.

2.3.2 Global feature descriptor

Local feature descriptor collects feature information from a limited size neighborhood of feature point/region, it loses the information outside the neighborhood, but global features can compensate for this deficiency. It encodes an image in a way that allows it to be compared and matched to other images. A global descriptor describes the whole image. They are generally not very robust as a change in part of the image may cause it to fail as it will affect the resulting descriptor. Mutual information (MI), Shape Matrices, Invariant Moments etc. are some common global feature descriptors. We will give a detailed introduction of MI to help reader have a good understanding of this kind descriptor.

Firstly, we give the definition of joint entropy:

$$MI_{I1,I2} = H_{I1} + H_{I2} - H_{I1,I2} \tag{2.15}$$

Here, $H_{I1,I2} = \sum_p h_{I1,I2}(I_{1p}, I_{2p})$,thus, the joint entropy is calculated as a sum of data terms that depend on corresponding intensities of a pixel p. $h_{I1,I2}(i,k) = -\frac{1}{n} log(P_{I1,I2}(i,k) \otimes g(i,k)) \otimes g(i,k)$, it is calculated from the join probability distribution $P_{I1,I2}$ of corresponding intensities. The number of corresponding pixels is n, and $P_{I1,I2}(i,k) = \frac{1}{n}\sum_p T[(i,k) = (I_{1p}, I_{2p})]$, The probability distribution of corresponding intensities is defined with the operator $T[\,]$, which is 1 if its argument is true and 0, otherwise.

After the definition of joint entropy, we explain how the MI can be used as global descriptor and realize the function of stereo matching. We assume that the base image is $I1$, the image needs to be matched is $Imatch$, the relationship between base image $I1$ and $Imatch$ can be formulated as the following:

$$q = e_{bm}(p) \tag{2.16}$$

Here, p is the pixel point on base image and q is its corresponding point on the matching image, the function $e_{bm}(p)$ symbolizes the disparity in the match image for the base image pixel p. Thus, $I2$ can be achieved by wrapping $Imatch$ according to $e_{bm}(p)$.

We know that the entropy H_{I1} is constant and H_{I2} is almost constant as the wrapping process merely redistributes the intensities of $I2$. Thus, $h_{I1,I2}(I_{1p}, I_{2p})$ serves as cost for matching two intensities. For well registered images, the joint entropy $H_{I1,I2}$ is low because one image can be predicted by the other, which corresponding to low information. Therefore, the stereo matching process is to find an appropriate wrapping function which minimizes the joint entropy.

As the global descriptor achieve information of pixel p from the whole image, it is sensitive with the change of image. For sea surface image matching, we do not think it is a good choice because the long-distance sea surface stereo images are not exactly alike and the illuminance of them are uneven. In the next subsection, we will introduce the method proposed for sea surface images.

2.3.3 Feature vector for sea wave description

As sea surface image suffers from low-quality, repetitive structure, it's difficult to construct highly discriminative descriptor for feature point, thus, we extract sea waves on sea surface as feature region and construct feature vector for each sea wave. Next, I will show specific contents of this feature vector.

Epipolar constraint is usually used to reduce searching area of matching process. Thus, sea wave location can be used as one of discriminative features. In this paper, sea wave is a feature region, and we take its gravity center to represent its location, it is determined by the nature of extraction algorithm: the same sea wave in one image pair is extracted by different thresholds, causing the extracted sizes, shapes and edges different within one image pair, thus, compared with center point, barycenter is much more stable.

As we have introduced before, sea water moves all the time, leading left and right sea surface images dissimilar due to different shooting angles and time. Meanwhile, almost all the sea waves are circle resembling, the problem now converts to matching circle resembling sea waves, at the same times, the positive matched sea wave pairs are dissimilar. Thus, we need to seek descriptor which is robust to positive matched sea wave pairs' dissimilarity and can distinguish circle resembling sea waves. Common descriptors like SIFT, DAISY, HOG are not adaptive to sea wave size because they generate descriptors within fixed region. Single feature is hard to distinguish sea waves from each other, one way to solve this problem is tracking and learning sea wave's feature during its existing time, but this method cannot realize real-time measurement. Another way is to integrate multi features to form matching feature vector.

For a specific sea wave, size, height and width will not change a lot simultaneously within one image pair, though the sea wave is extracted by different thresholds. Next, with a relatively accurate barycentric coordinates, diagonal lengths(45° and 135°) can be used to reflect sea wave shape. Circularity can be used to measure the similarity between sea wave and a circle but its effect is small as most sea waves are circle resembling. Brightness distribution can also be used as a discriminative feature, but it is also not a decisive discriminant feature as shooting from different angle, brightness distribution will change. In this paper, for a specific sea wave, we form sea waves distinctive descriptor by integrating its barycenter, size, circularity, width, height, brightness, diagonal lengths, and use this descriptor to distinguish and match sea waves. Fig. 2.14 shows the description of different features.

Within one image pair, We extract two groups of sea wave feature vectors, left: $\boldsymbol{F^L} = \{\boldsymbol{F_1^L}, \boldsymbol{F_2^L}, \dots, \boldsymbol{F_n^L}\}$, right: $\boldsymbol{F^R} = \{\boldsymbol{F_1^R}, \boldsymbol{F_2^R}, \dots, \boldsymbol{F_m^R}\}$, *n* and *m* are sea wave numbers in left and right images respectively, $\boldsymbol{F_i^L}$ is the feature vector of the i^{th} sea wave on the left image. Each feature vector *i* contains 8 features defined as follows:

$$[f_{i1} \quad f_{i2} \quad f_{i3} \quad f_{i4} \quad f_{i5} \quad f_{i6} \quad f_{i7} \quad f_{i8}] \tag{2.17}$$

where:

f_{i1} is the location of barycenter, displayed in x and y coordinates of the pixel;

f_{i2} is the size of the sea wave, displayed in total number of pixels;

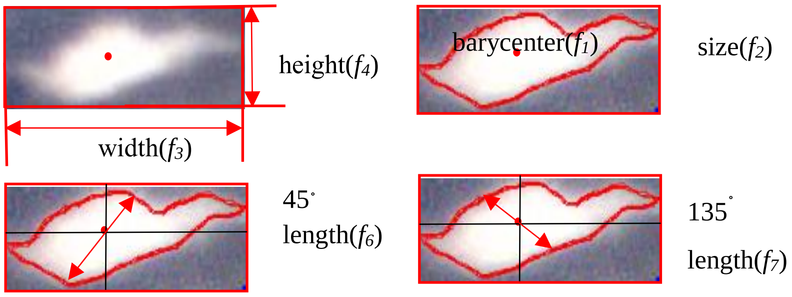


Fig. 2.14 illustration of each feature.

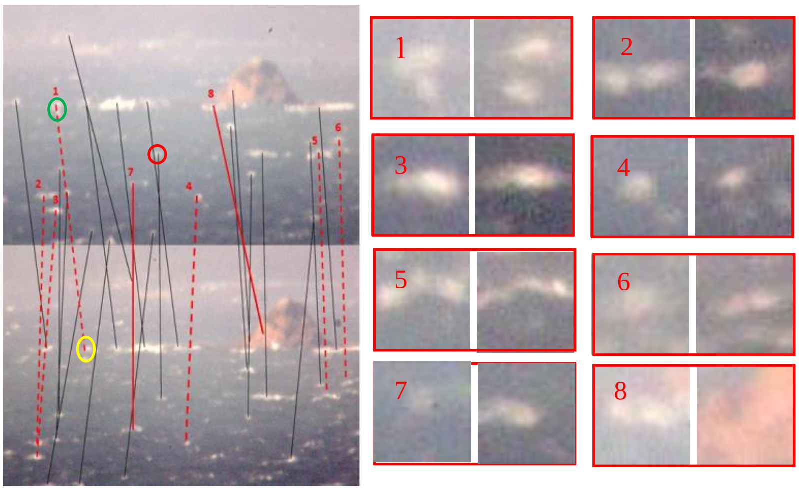
f_{i3}, f_{i4} are the width and height of the sea wave, displayed in largest numbers of wave pixel in vertical and horizontal orientations;

f_{i5} is the circularity of the sea wave, displayed in a number between 0 and 1. The larger it is the more similar it is to a circle;

f_{i6}, f_{i7} are the lengths of the sea wave on the two diagonals' orientations of 45° and 135°;

f_{i8} is the brightness of the sea wave, displayed in the sum of grayscale of all pixel points of sea wave i.

In the following content, the k^{th} feature of the calculated sea wave is easily expressed in f_k. As a result, for the proposed stereo system, the problem of sea wave matching can be converted to the problem of matching between two feature vector sets $\boldsymbol{F^L}$, $\boldsymbol{F^R}$. Fig.2.15 shows the matching result by Euclidean distance between feature vectors.



(a) (b)

Fig.2.15 Matching result of Euclidean distance between feature vectors, (a) is the matching result, (b) shows the enlarged images of incorrect matching.

The left image shows the matching result, black solid lines represent true positive matching (correct matching and detected by matching method), red solid lines represent false positive matching (incorrect matching but detected by matching method), red dotted lines represent false negative matching (correct matching but haven't been detected by matching method) . The right 8 images are the enlarged images of sea wave pairs which belongs to false positive and false negative matching.

By observing these enlarged images, we find that when false negative matching happens (NO.1,2,3,4,5,6 enlarged image pairs), two feature vectors at least have one component changing a lot, take NO.1 enlarged image pair as an example, in the left image, the sea wave can be taken as a whole sea wave, but in the right image, it splits into two waves due to different extraction threshold, shooting angle and time, obviously, it will make the two feature vectors different in height component(f_4) and cause false negative matching. When false positive matching happens (NO.7,8), the two matched feature vectors are much more similar with each other than other sea wave's feature vector according to Euclidean distance. Taken NO.7 enlarged image pair as an example, in the left image, it is a small sea wave, but in the right image, it is connected with its right-side sea wave and forms a large sea wave, there is no other sea wave around NO.7 sea wave, according to Euclidean distance, the two sea waves are matched and lead to false positive matching. To avoid these two types of incorrect matching, we need a matching strategy that allows one or more than one component of two feature vectors have large difference, as well as avoid low similarity false positive matching.

2.4 Sparse matching

In the previous subsection, we have showed some sparse matching results. Feature point/region detection and description are all sparse matching elements, thus in the subsection, we will give a detailed introduction of sparse matching.

Sparse matching algorithms are used to establish a set of robust matches between a stereo image pair. The sparse matches are usually used to compute the epipolar geometry, using techniques such as the RANSAC (random sampling) method. It plays an important role in computer vision applications, and only be performed on parts of stereo images. It is applied to address a variety of problems, such as feature-based techniques of simultaneous localization and mapping (SLAM) [69], indirect dense SLAM approaches [70], tracking task [71] and real-time mosaicking of aerial images [72].

Specifically, for sea surface image, the sparse matching means that we match the sea waves on the sea surface image instead of matching all the pixel points. The sea waves can be considered as feature region, we calculate its barycenter as its specific location, matching sea waves is matching the barycenter of the sea wave.

2.4.1 Matching strategy

After the feature point/region being detected and the features of the point/region being extracted, the next step is to matching the most similar points/regions. There are many functions can be used to measure the similarity of each features, such as Euclidean distance, Manhattan distance, Normalized Cross Correlation (NCC), Cosine, Hamming distance etc. In this subsection, we will give a detailed introduction of these similarity calculation function.

First of all, it is the Euclidean distance. It is the length of line segment between the two points in Euclidean space. It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem, therefore it can also be called Pythagorean distance. For two points p and q given by Cartesian coordinates in n-dimensional Euclidean space, the Euclidean distance between then can be calculated by the following formula:

$$d(p,q)=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\cdots+(p_i-q_i)^2\cdots+(p_n-q_n)^2} \quad (2.18)$$

Not limited to points, it also can be used to measure the similarity between two high dimensional vectors, and in subsection 2.3.3, we choose the Euclidean distance as the matching strategy to match the sea wave feature vectors. The larger this value is, the less similar they are. Although it can match most of sea waves correctly, there are still some errors existing because the extracted feature vectors for the same sea wave are not exactly alike due to the different shooting angles and extraction thresholds.

NCC is also a measure of similarity of two series in signal processing field, and also usually chosen as image matching strategy, as it will return large value when the searching template find the most similar part in original image. The cross correlation can be calculated by the following formula:

$$C_{I1,I2}(u,v) = \sum_{i,j} I1(u+i,v+j)I2(i,j) \tag{2.19}$$

$I1$ is the searching image and $I2$ is the matching template, when $C_{I1,I2}(u,v)$ responses the largest value, the corresponding location (u,v) is the final matching result. However, for sea wave matching, as we have calculated the feature vectors, the matching process is performed between different feature vectors, thus $(u,v) = (0,0)$. NCC is an exhaustive searching over all image patches in the searching image, it is slow. Recently, many speed NCC methods are proposed [73,74]. As its efficiency, we also do not take it for sea wave matching and in the subsection 2.4.2, we will introduce the matching strategy used in our proposed tsunami measurement system.

2.4.2 Simple introduction of decision tree

Decision tree [57] classifier (DTC) is capable to break down a complex decision-making process into a collection of simpler decisions, then providing a solution easier to interpret. It is chosen as matching strategy of our sea wave matching problem. DTC makes judgments based on global analysis of feature vectors' difference, selectively robust to some uncertainty features, and sensitive to some decisive features, it can reduce computational load of matching process as well as the influence caused by different extraction threshold, shooting angle and time.

It is commonly used in operations research and operations management. This algorithm has many advantages such as simple to understand and interpret, a white box model and can make judgement even with little hard data. There are many algorithms used to build decision tree, such as Iterative Dichotomiser 3 (ID3), successor of ID3 (C4.5), Classification and Regression Tree (CART) etc. Here, we give a simple introduction of ID3 to show the whole construction process of decision tree.

In ID3, the decision tree is built based on the concept of Information gain. Before this, we give the definition of entropy:

$$H(T) = -\sum_{i=1}^{J} p_i \, log_2 \, p_i \tag{2.20}$$

Where p_i is fraction that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree. The IG (Information Gain) is the decrease of the whole entropy after one-time decision.

$$IG(T,a) = -\sum_{i=1}^{J} p_i \, log_2 \, p_i - \sum_a p(a) \sum_{i=1}^{J} P_r(i|a) \, log_2 \, P_r(i|a) \tag{2.21}$$

The decision node chooses a if its corresponding $IG(T,a)$ is the largest. For each node of the tree, the information value "represents the expected amount of information that would be needed to specify whether a new instance should be classified yes or no, given that the example reached that node". However, this algorithm is applicable for the case of discrete features, the sea wave features are continuous. Thus, in the next subsection, we will introduce the method of building decision tree for continuous sea wave features.

2.4.3 Establish decision tree for sea wave matching

We extract sea waves within multi image pairs, get two sea wave feature vector sets $\boldsymbol{F^L} = \{\boldsymbol{F_1^L}, \boldsymbol{F_2^L}, \dots, \boldsymbol{F_n^L}\}$ (superscript L represents that the feature vector comes from left images), $\boldsymbol{F^R} = \{\boldsymbol{F_1^R}, \boldsymbol{F_2^R}, \dots, \boldsymbol{F_m^R}\}$ (superscript R represents that the feature vector comes from right images). After randomly combining between the elements within set $\boldsymbol{F^L}$ and $\boldsymbol{F^R}$, we get *N*=*m***n* pairs of combinations $\{(\boldsymbol{F_1^L}, \boldsymbol{F_1^R}), (\boldsymbol{F_1^L}, \boldsymbol{F_2^R}), \dots, (\boldsymbol{F_n^L}, \boldsymbol{F_m^R})\}$. Their matching results are stored in set $P = \{p1, p2, \dots, pN\}$, where pi equals 0 or 1, representing incorrect and correct matching labels respectively. We choose to use 70% of the pair combination results to form the training set, and the rest of the pairs to form the test set., we utilize C4.5 system [58] to construct decision tree. $(\boldsymbol{F_i^L}, \boldsymbol{F_j^R})$ represents two feature vectors, the superscripts of them indicate the source of them, for example, if $\boldsymbol{F_i^X} = \boldsymbol{F_1^R}$, it means the feature comes from the NO.1 sea wave on right image. The subscript of *f* indicate the specific feature, such as size, height, width etc.

To simplify, we generate new feature vector based on $(\boldsymbol{F_i^L}, \boldsymbol{F_j^R})$ as the inputs of decision tree system. For the barycenter *f*f_1, it obeys to epipolar constraint, and for correct matched pairs:

$$f_1^{i*m+j} = f_1^{\boldsymbol{F_i^L}^T} F f_1^{\boldsymbol{F_j^R}} \rightarrow 0 \tag{2.22}$$

F is the fundamental matrix. f_1^{i*m+j} is the first element of new feature vector generated from $(\boldsymbol{F_i^L}, \boldsymbol{F_j^R})$.

f_2, f_3, f_4, f_6, f_7 and f_8 reflect the sea wave's size, shape and brightness. The same sea wave in different images should share the similar values, so we utilize the ratio between smaller (superscript *s*) and larger (superscript *l*) values to measure similarity:

$$f_k^{i*m+j} = \frac{f_k^s}{f_k^l} \quad , k \in \{2,3,4,6,7,8\} \tag{2.23}$$

Superscript *s* represents the smaller one between $f_k^{\boldsymbol{F_i^L}}$ and $f_k^{\boldsymbol{F_j^R}}$, *l* represents the larger one, thus, f_k^{i*n+j} ranges within (0,1].

f_5 reflects the sea wave's curvature, and ranges within (0,1). If the sea wave's shape approaches being circular, f_5 will approach to 1. We utilize absolute error to measure the difference:

$$f_5^{i*m+j} = abs\left(f_5^{\boldsymbol{F_i^L}} - f_5^{\boldsymbol{F_j^R}}\right) \tag{2.24}$$

Thus, we generate a new feature vector like equation (2.25) shows, which can be directly used to construct the decision tree:

$$f^{i*m+j} = \left[f_1^{i*m+j}, f_2^{i*m+j}, f_3^{i*m+j}, f_4^{i*m+j}, f_5^{i*m+j}, f_6^{i*m+j}, f_7^{i*m+j}, f_8^{i*m+j}\right] \tag{2.25}$$

Table 2. shows parts of the extracted sea wave features generated by our wave extraction program, which is used to train decision tree.

Table 2. Extracted sea wave features(train data)

No	*l* (f_1)	*s* (f_2)	*h* (f_3)	*w* (f_4)	*c* (f_5)	*d45* (f_6)	*d135* (f_7)	*b* (f_8)	M
1	0.040	0.994	0.954	1.000	0.016	0.936	0.966	0.831	1
2	1.343	0.831	0.958	1.000	0.106	0.862	0.964	0.271	0
3	0.019	0.811	0.810	0.936	0.029	0.793	0.963	0.032	1
4	2.031	0.580	0.882	0.628	0.165	0.880	0.958	0.990	0
5	0.046	0.843	0.931	0.741	0.165	1.000	0.952	0.034	1
6	0.149	0.974	0.839	0.848	0.106	0.758	0.951	0.282	0
7	0.366	0.939	0.900	0.853	0.102	0.914	0.950	0.250	0
8	0.010	0.692	1.000	0.711	0.103	0.886	0.947	0.164	1
9	0.013	0.578	0.964	0.578	0.121	0.956	0.946	0.065	1
10	2.889	0.944	0.526	0.829	0.040	0.750	0.944	0.715	0
11	0.040	0.797	0.867	0.689	0.164	0.727	0.941	0.151	1
12	0.429	0.483	0.547	0.253	0.126	0.765	0.939	0.352	0

C4.5 system also takes information gain ration equation (2.26) to choose split property, it can resist to the uneven distribution of different categories train samples. *P* represents the output set of input training samples, and f_k is the feature, $k \in 1,2, \ldots ,8$. Please refer to [58] for details. Pessimistic Error Pruning (PEP) is used to avoid overfitting.

$$InfoGainRation(P, f_k) = \frac{InfoGain(P,f_k)}{SplitInfo_{f_k}(P)} \tag{2.26}$$

C4.5 proposes a standard continuous data discretization method, however, for our feature vector with real physical meaning, we can add some constraints to speed up split threshold choosing process:

a) f_1 represents location correlation degree, for true positive matching, it tends to 0. Thus, we do not need to find spilt threshold within the range (δ, +∞), δ≫0. In this paper, we choose $\delta = \frac{1}{2}(\max_j \left| f_1^{\boldsymbol{F_i^L}} F f_1^{\boldsymbol{F_j^R}} \right| + \min_j \left| f_1^{\boldsymbol{F_i^L}} F f_1^{\boldsymbol{F_j^R}} \right|)$.

b) f_2, f_3, f_4, f_6 and f_7 range within (0,1], they reflect the similarity of sea wave's size and shape, when similarity is smaller than 0.5, obviously, it is incorrect matching. Thus, the split threshold lies within [0.5,1].

Fig.2.16 shows the final established decision tree. It is established by training on 853 pairs of sea wave data (336 pairs of true positive matching+517 pairs of false positive matching and true negative matching), the circle nodes are decision nodes, one decision node represents a test on one kind of attribute. The variable ${f_k}_B^A$ in the circle represents that the attribute used to classified is f_k, to be classified training sea wave number is A, among the total training sea waves, true positive matching number is B. The number under decision node is split threshold, when sea wave's feature f_k is greater than this number, it is classified into the decision node's left sub-tree or sub node, otherwise right sub-tree or sub-node. The square nodes are leaf nodes, one leaf node represents a classification result. The variable No_B^A/Yes_B^A in leaf node represents that the sea wave is incorrect matched(false positive + true negative)/true positive matched, A and B have the same meanings like decision node.

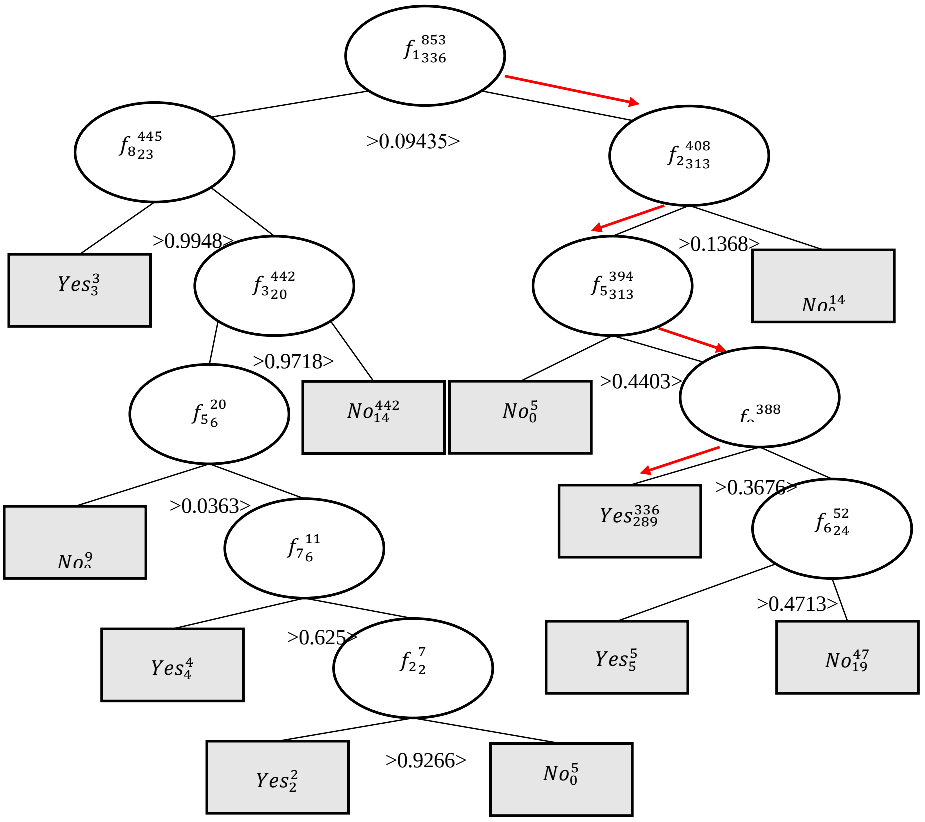


Fig.2.16 Final established decision tree.

The red arrow path represent one of the classification path, it means that when the feature vector [f_1,f_2,f_3,f_4,f_5,f_6,f_7,f_8] of a pair of sea waves satisfies f_1 <0.09435, f_2 >0.1268, f_5 <0.4403, f_8 >0.3676, they are true positive matching.

2.4.4 Matching results

Now, we give a specific example to show how the decision tree we built matches the sea waves which haven't been matched by the former method described in subsection 2.3.3. Take the former false negative matched sea wave 1(in Fig.2.15, marked by green circle) for example, we extract its feature vector, at the same time, we also extract its corrected matched sea wave's (in Fig.2.10, marked by yellow circle) feature vector out, as well as the sea wave's in red circle, and record them in Table 3.:

Table 3. Feature vectors of waves in green, yellow, red circles

Color	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
Green	(293,462)	1808	49	60	0.49	38	46	3.8
Yellow	(455,463)	1209	52	33	0.71	39	42	2.4
Red	(837,672)	864	41	27	0.70	32	34	1.6

As left and right images of sea wave 1 are different (see Fig.2.15), from Table 3, we find that the extracted circularities(f_5) and heights(f_4) (Table 3, recorded in red blank) of true matching sea waves(Fig.2.15, sea waves in green and yellow circles) are much more different than false matching sea waves(Fig.2.15, sea wave in red and yellow circles). Matching by Euclidean distance will take the false matching as final matching candidate. But by the decision tree we built, the true matching sea waves will pass red arrow path (see Fig.2.15)and be one of final matching candidates. Fig.2.17 shows the matching result of decision tree method.

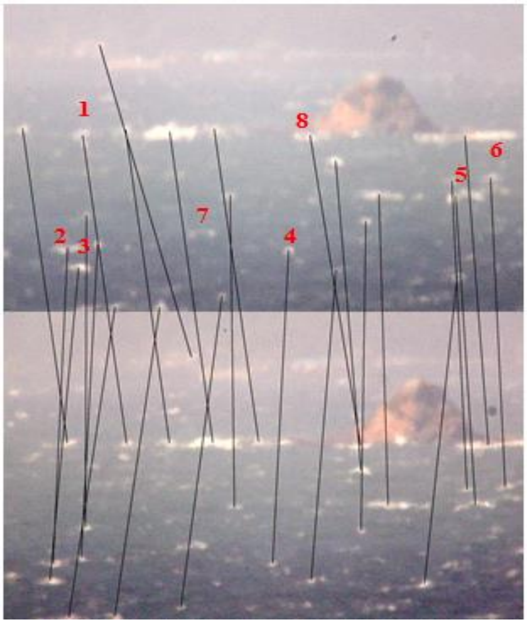


Fig.2.17 Matching result of decision tree method.

Comparing Fig.2.17 with Fig.2.15, we find that the false negative matching (NO. 1,2,3,4,5,6, they have true matching waves in right image but haven't been detected by the matching method described in 2.3.3) is eliminated and correctly matched by decision tree method. Sea wave 7,8 are false positive matching by the matching method described in 2.3.3,1) as sea wave 7 doesn't have true matching sea wave in right image, it is removed from the final matching result by decision tree method, 2) as sea wave 8 has true matching sea wave in right image, it is correctly matched by decision tree method. From the matching result, we can conclude that most false negative matching and false positive matching can be removed by decision tree method.

Decision tree method can eliminate false negative matching, but at the same time, it will also lead to false positive matching. Thus, the next stage is to remove the false positive matching from the matching result of decision tree method. Inspired by geometric and spatial coherence, we know that the calculated 3D coordinates based on decision tree matching results must be coherent. Please refer to [59] for detailed calibration process. In this paper, we only discuss the calculation of 3D coordinates, it is based on image stereo calibration result.

There are two constraint conditions ensuring spatial coherence: 1)The shooting distance *Z* of our system ranges from 4 km to 20 km; 2)the distance *Z* of sea waves close to sea surface image bottom must smaller than the sea waves close to sea surface image top, for a 2D point (*x*,*y*), the calculated distance *Z* is related to its y. Assume sea surface as a flat plane, all the points share the same *Y* coordinate, take the simplest case into consideration, we have: $Y_1 = \frac{by_1}{d_1} = Y_2 = \frac{by_2}{d_2}$.Therefore, $d \propto y$, at the same time, $Z \propto 1/d$, we can conclude

$Z \propto 1/y$.

In fact, real sea surface is not a flat plane, Y value always fluctuates within a range, but the overall relationship between y and *Z* is stable, it can be used to verify matching results. At the same time, we do not need to give a precious estimation of parameter matrix, because parameter matrix does not influence the relationship between y and *Z*.

2.5 Dense matching

Recently, advances in the field of computer vision have allowed for the generation of detailed and reliable point clouds from images. In stereo system, 3D geometry is obtained by creating images of the same object from different positions. This makes a single point on the object visible as a pixel in multiple images. For each image, a straight line can be drawn from the camera center through the pixel in the image. These lines will intersect at one point, which is the 3D location of the object point. The algorithm which matches all the points within camera scene is so called dense matching. It generates the point clouds for 3D reconstruction from stereo camera system instead of traditional Lidar system.

Dense matching is an approach to obtain corresponding point for every pixel within in an image. Rather than detecting and describing feature point/region like sparse matching, it compares two overlapping images point by point to generate disparity map. Commonly, it requires an image rectification step before the matching starts. The stereo images need to be wrapped in such a way that each row of pixels in one image corresponds exactly to one row in the other image. However, non-rigid motion or nonlinear movements, as unavoidable in sea surface image or aerial image, causes epipolar lines to be general curves and images that cannot be rectified [75]. Thus, for sea surface images we eliminate the image rectification process and do not wrap images before matching starting.

Sparse matching is a region to region matching method. In fact, when we measure sea level height at 20 km away from our camera, a 20-pixel disparity error will result in a sea level height measurement error of approximately 1m within our proposed system. Thus, the sparse matching result is not precise enough to be used to compute sea level height. It is necessary to perform precise dense matching to rectify the first step sparse matching result. Before introducing the specific content of dense matching, we firstly introduce some primary concepts which makes up the necessary elements of dense matching: 1) cost calculation, 2) cost aggregation, 3) cost volume. Before establishing the cost volume, we formulate the relationship between the disparity *d* and image coordinate *y*, and use this relationship to reduce the matching computation load.

2.5.1 Cost calculation and aggregation

As mentioned before, dense matching compares two overlapping images point by point to generate disparity map. How to compare the two overlapping images? It is the task of cost calculation. In some local algorithms, the cost calculation at a given image pixel depends only on the pixel intensity, gradient, color or some other visibly direct features. In some global algorithms, such as MI mentioned before, they may iteratively update the whole disparity map to decrease cost of the whole image. Take MI for example, it sets the joint entropy as matching cost and randomly starts with a random disparity map, update the disparity map within each iteration to decrease the joint entropy.

The matching cost of MI can be defined as the following formula:

$$C_{MI}(\boldsymbol{p}, d) = -mi_{I_b, f_D(I_m)}(I_{b\boldsymbol{p}}, I_{m\boldsymbol{q}}) \tag{2.27}$$

$$mi_{I_1,I_2}(i,k) = h_{I_1}(i) + h_{I_2}(k) - h_{I_1,I_2}(i,k) \tag{2.28}$$

Disparity map is required for warping I_m before calculating *mi*. The definition of h_{I_1}, h_{I_2} and h_{I_1,I_2} can be find in the formal section of global feature descriptor. It can be used as cost because an appropriate disparity map can warp I_m and make the warped image I_2 be similar to I_1 as much as possible, which will decrease the value of joint entropy.

The difference of intensity can also be used of cost, the cost can be defined as the following:

$$C_I = \|I_1 - I_2\| \tag{2.29}$$

Like the definition of MI cost, the variable I_1 is the base image and the variable I_2 is the warped image of I_m. The disparity map D control the warping progress of I_m, the more similar I_1 and I_2 are, the smaller the intensity difference will be, and the more correct the disparity map will be.

Another cost can be calculated by the difference of gradients of I_1 and I_2, it can be defined as the following equation (2.30):

$$C_G = \|\nabla_x I_1 - \nabla_x I_2\| \tag{2.30}$$

Here, ∇_x calculates the gradient in *x* axis direction, also we can define the gradient difference in *y* axis direction by replacing ∇_x with ∇_y. It is also easy to understand the meaning of this cost and we do not repeat it.

In some cases, the stereo images are taken with different shooting angles, which will make the whole intensities of left and right image different. Directly difference of intensity can not describe the similarity and measure the matching cost. Some descriptors which is robust to intensity change are proposed to form a much more appropriate cost for dense matching. Take descriptor census as an example. It is an image operator that associates to each pixel of a grayscale image a binary string, encoding whether the pixel has smaller intensity than each of its neighbors, one for each bit. It is a non-parametric transform that depends only on relative ordering of intensities, and not on the actual values of intensity, making it invariant with respect to monotonic variations of illumination, and it behaves well in presence of multimodal distributions of intensity, e.g. along object boundaries. Commonly, the neighborhood size of census transform is 3×3, comparing pixel p with all its neighbors by a function defined as the following:

$$\xi(p,p') = \begin{cases} 0 & if\ p > p' \\ 1 & if\ p \leq p' \end{cases} \tag{2.31}$$

The results of these comparisons are concatenated and the value of the transform is an 8-bit value, that can be easily encoded in a byte. The similarity of census transform results is calculated by hamming distance, it is a metric for comparing two binary data strings, it is the number of bit positions where the two binary bits are different. It is performed by XOR operation, followed by counting the total number of one in the resultant string. The whole workflow of calculating similarity by hamming distance is like the following Fig.2.18 shows.

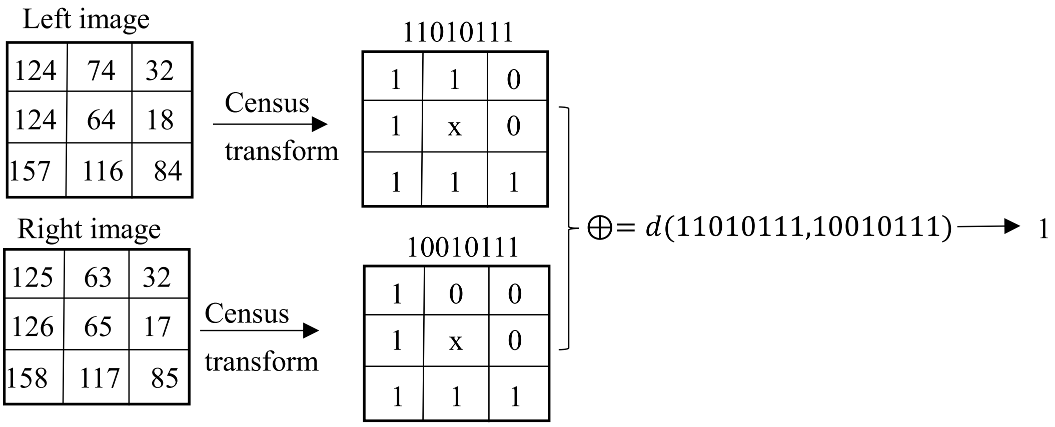


Fig.2.18 Workflow of calculating hamming distance.

As cost is calculated in each pixel point within image, it is easy to be disturbed by noise, thus it is necessary to operate the second step of cost aggregation. There are two kinds of cost aggregation algorithms: 1) local aggregation by filters, 2) global aggregation by minimizing the energy function. Local cost aggregation methods are traditionally performed locally by summing/averaging matching cost over windows with constant disparity. The most efficient local cost aggregation method is unnormalized box filtering which runs in linear time, such as median filtering, mean filtering as well as bilateral filter etc. A lot of approaches have been proposed to speed up these operators like guide filter.

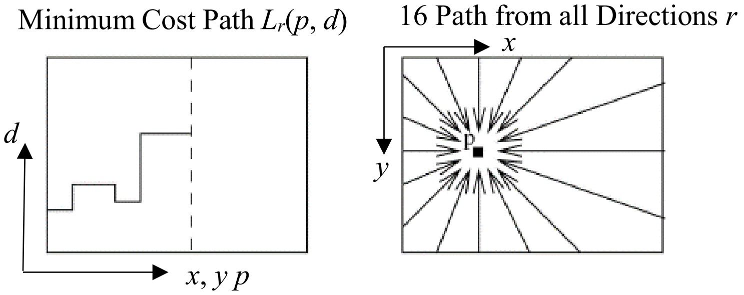
Global aggregation methods usually seek a disparity map which can minimize the energy function. It is conducted on the assumption that the disparity within an object region is continuous, discontinuity should be punished by a constant penalty or dynamic penalties. We add a discontinuous penalty at the end of cost function to form the energy function. A lot of methods are proposed to seek the minimum value of this function as it is a np-hard problem. Among them, the most popular methods include belief propagation, dynamic programming, semi-global matching and graph cut. Unlike local algorithms, a global algorithm estimates the disparity at one pixel using the disparity estimates at all the other pixels.

To maintain the integrity of the thesis, we would like to give a simple introduction of semi-global matching method (SGM) because we use an improvement of SGM to search the dense matching result of sea surface images. Firstly, we give the definition of energy function:

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 \, T\left[\left|D_p - D_q\right| = 1\right] + \sum_{q \in N_p} P_2 \, T\left[\left|D_p - D_q\right| > 1\right] \qquad (2.32)$$

The first term is the sum of all pixel matching costs for the disparities of *D*. The second term adds a constant penalty P_1 for all pixels q in the neighborhood N_p of p, for which the disparity changes a little bit (that is, 1 pixel). The third term adds a larger constant penalty P_2, for all larger disparity changes. We must ensure that $P_2 > P_1$.

The SGM aggregates matching costs in 1D from all directions equally, the smoothed cost $S(p, d)$ for a pixel p and disparity d is calculated by summing the costs of all 1D minimum cost paths that end in pixel p at disparity *d*, as Fig.2.19 shows.


Minimum Cost Path $L_r(p, d)$

16 Path from all Directions r

Fig.2.19 Aggregation of costs in disparity space

These paths through disparity space are projected as straight lines into the base image but as non-straight lines into the matching image, according to disparity changes along the paths. It is noteworthy that only the cost of the path is required and not the path itself. The cost $L_r(p, d)$ along a path traversed in the direction *r* of the pixel p at disparity *d* is defined recursively as:

$$\begin{aligned} L_r(p, d) = C(p, d) + min(L_r(p - r, d), L_r(p - r, d - 1) + P_1, \\ L_r(p - r, d + 1) + P_1, \min_i L_r(p - r, i) + P_2) \\ - \min_k L_r(p - r, k) \end{aligned} \qquad (2.33)$$

The remainder of the equation adds the lowest cost of the previous pixel $p - r$ of the path, including the appropriate penalty for discontinuities. This implements along an arbitrary 1D path. The costs L_r are summed over paths in all directions *r*. The number of paths must be at least eight and should be 16 for providing a good coverage of the 2D image. The final aggregated cost is calculated by the following equation (2.34):

$$S(p,d) = \sum_r L_r(p,d) \qquad (2.34)$$

Until now, we have introduced all the basic elements necessary for dense matching, and in the following section, we will introduce the proposed method for sea surface image matching. Before this, we formulate the relationship between disparity and coordinate to reduce searching region of dense matching.

2.5.2 Form the relationship between disparity and y coordinate

To set up our user coordinate system (UCS), we set the camera location at coordinate origin (0, 0, 0), with axis X, Y and Z as shown in Fig 7. We define the angle between the Z axis and the shooting camera's normal direction as θ, and earth surface as E. We assume the earth is a sphere. Its center is C = (0, H, 0), and its radius is R. The coordinate of target T in the UCS is (X, Y, Z), and the coordinate of its projected point *t* in the image plane is (*x, y*). According to Fig.2.20 and the pinhole camera model, we know that the relationship between the target T and its image projection point t is as follows:

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boldsymbol{A}[\boldsymbol{R} \quad \boldsymbol{T}]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (2.35)$$

Our UCS shares the same origin with the pinhole camera coordinate and rotates θ around the X axis. We assume this pinhole camera is the left camera. The rotation and translation of it is $[\boldsymbol{R} \quad \boldsymbol{T}]$. To simplify we assume for two well calibrated and rectified cameras that the rotation and translation between left and right cameras are *I* and $[b \quad 0 \quad 0 \quad 0]^T$. Accordingly, we can formulate the relationship between disparity *d* and image coordinate *y*, as the following (2.36) shows:

$$\mathrm{y} = \frac{fy \times d}{fx \times b}\left(H\cos^2\theta - \sqrt{R^2 - X^2 - H^2\sin^2\theta - 2H\sin\theta\frac{fx \times b}{d} - (\frac{fy \times b}{d})^2}\right) + h \qquad (2.36)$$

In this equation, *y* is the *y* coordinate of the target in the image plane, *h* (pixel) is the length of half the image plane's height, *fx* and *fy* (pixel) are the focal lengths of the camera, *d* (pixel) represents the disparity between left and right images, *b* represents the baseline length between

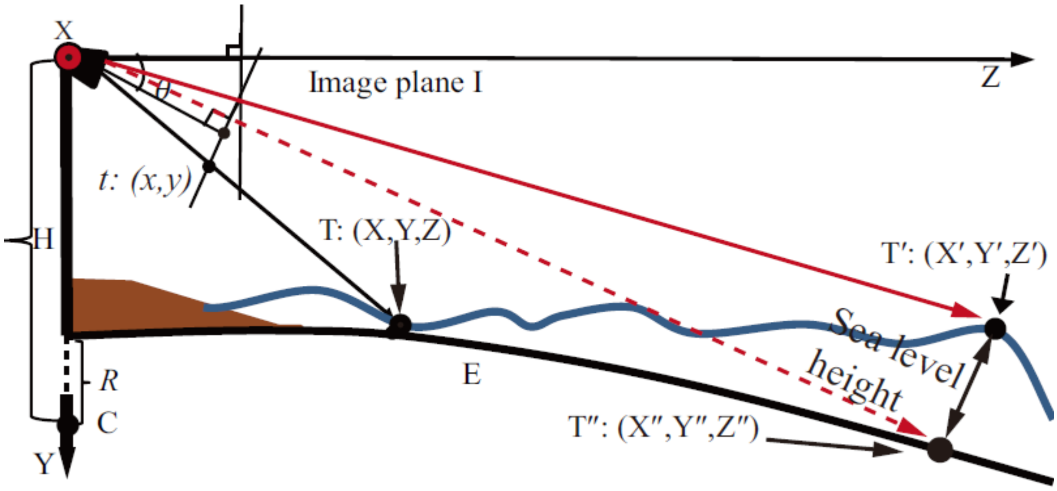


Fig.2.20 Set up user coordinate system (UCS).

left and right cameras, *R* is the radius of earth, *H* is the length between the earth center and our UCS origin, and *X* is the coordinate in the *X* axis of our UCS. From this equation, we know the relationship between *y* and *d*, as Fig.2.21(a) illustrates, the bold line representing the relationship between *d* and *y* within the image plane.

Given a sea surface image taken by our system, the disparity *d* increases when *y* increases. For target $\mathrm{T}': (\mathrm{X}', \mathrm{Y}', \mathrm{Z}')$, the sea surface height is not 0, and the coordinate *y* of the projected target point will be smaller than the value calculated according to Equation (2.36), which will cause a bright region in the disparity map, as Fig.2.21(b) illustrates (it is a schematic of a disparity simulation for the case where the sea surface height changes).

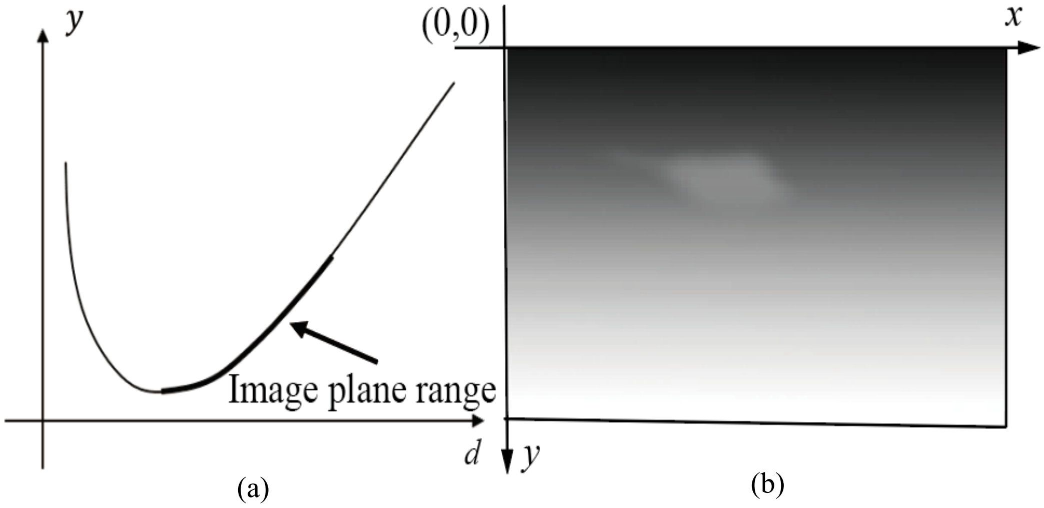


Fig.2.21 Illustration of the relationship between *y* and *d*, (a) is the image of Equation (2.36), (b) illustrates the disparity map in the case where sea wave height is not 0 (like the target T' shown in Fig.2.20).

According to the relationship between *d* and *y*, we can drastically reduce the size of cost volume. A much smaller leaning cost volume is built to accomplish dense matching of long-distance sea surface images, although the disparity d varies in a wide range for this class of images. We will introduce its construction in the subsection 2.5.3.

For simplicity, we can also simply assume that the relationship between y and d is proportional. It still yields a relatively accurate result, but the disparity range of the matching search will be extended and the accuracy will be slightly lower due to the low-texture nature of the sea surface images. In this paper, we still recommend one-time sparse matching for the same class of sea surface images to obtain an exact leaning cost volume first.

2.5.3 Building leaning cost volume

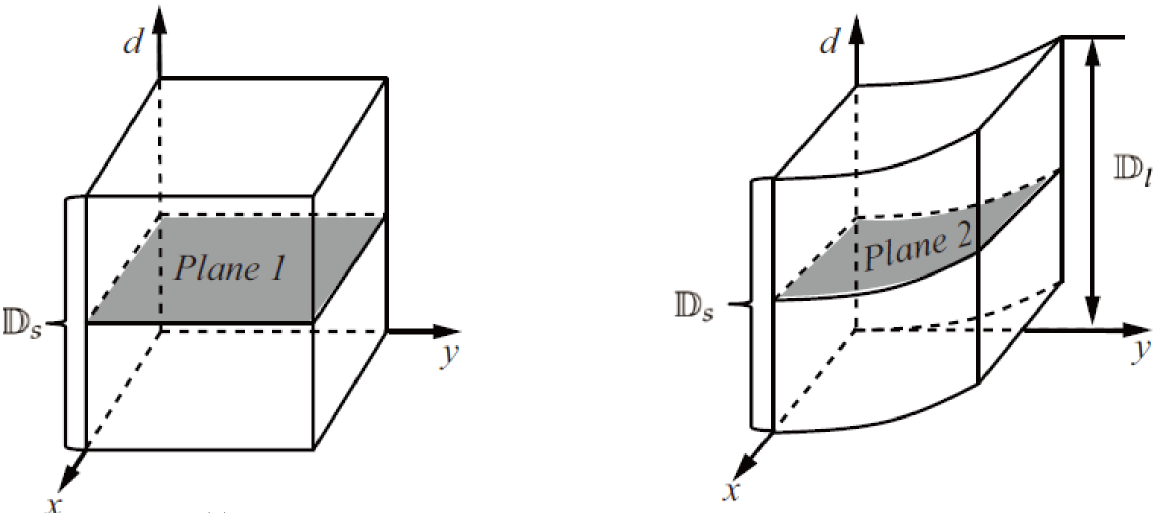
Leaning cost volume can be built based on two facts: 1) the relationship between disparity d and y coordinate obeying Equation (2.36), and 2) sparse matching has been accomplished, which can offer initial data for fitting the Equation (2.36) function.

This paper utilizes the least square method [60] to fit the function. To resist the illuminance change of left and right images caused by different shooting angles, the descriptor, which is used to compute the matching cost volume, is the combination of intensity, gradient and census [63] features. The matching cost of each pixel p (on the left image) is calculated from the difference between its feature descriptor and the feature descriptor of its suspected correspondence (on the right image), the suspected location being $q = f(p) + d$. The function $f(p)$ calculates the suspected location of the matching point on the right image according to Equation (2.36), and d is the disparity between left and right pixel points caused by the unevenness of the sea surface. Equation (2.37) shows the calculation of cost $C(p, d)$ at point p:

$$C(p,d) = 2 - exp\left(-\frac{(1-\alpha)\times min(\|I_p^L - I_q^R\|,\tau 1)}{\sigma_1}\right) \\ \times exp\left(\frac{\alpha\times min(\|\nabla_x I_p^L - \nabla_x I_q^R\|,\tau 2)}{\sigma_1}\right) - exp\left(-\frac{Hamming(census(p)^L, census(q)^R)}{\sigma_2}\right) \quad (2.37)$$

Here, I_p^L denotes the intensity of pixel point p. ∇_x is the grayscale gradient in the x axis direction. I_q^R is the intensity of the corresponding pixel of p on the right image. The disparity between them is $f(p) + d$. α balances the color and gradient terms' contributions, and $\tau 1$, $\tau 2$ are the truncation values. σ_1, σ_2 balance the census, intensity and gradient terms' contributions. $Hamming(census(p)^L, census(q)^R)$ represents calculating the hamming distance between vector $census(p)^L$ and $census(q)^R$. $census(p)^L$ denotes the census vector of pixel p on the left image, $census(q)^R$ is the census vector of the corresponding pixel of p on the right image. Census transformation is robust when the overall luminance of the image changes. We can choose the appropriate size of the neighborhood window to generate census vectors of different lengths.

Fig.2.22 demonstrates the cost volume. (a) is the common cost volume. All the points in the d = 1 plane share the same disparity. It is usually adapted when disparity varies in a small range Ds. (b) is the leaning cost volume proposed in this paper, the disparity plane is plane 2, and we can get it according to Equation (2.36). Although the disparity varies in a large range D_l, we can still search the best matching in the small range Ds. By the leaning cost volume, large disparity change can be calculated in advance according to Equation (2.36), and we just need to search the disparity $d \in Ds$ (= [0,20] interval for the best matching results. It greatly reduces the memory and time consumption of dense matching for the tsunami measurement system.



(a)

(b)

Fig.2.22 Illustration of cost volume, (a) common cost volume, (b) the leaning cost volume proposed in this paper for long distance matching.

2.5.4 Matching by the built cost volume

Pixel-wise cost calculation is generally ambiguous and wrong matches can easily have a lower cost value than correct ones, due to the low-texture area, noise and so forth. Therefore, we need to add other constraints to remove wrong matches. Observing the sea surface, we find that sea level height change is continuous, thus, we add a constraint that adjacent pixels must have similar disparities, and define the following global energy equation [37]:

$$E(D) = E_{data}(D) + E_{smooth}(D) = \sum_p (C(p, d_p) + \sum_{p' \in N_p} u(p, p') T[|d_p - d_{p'}| \neq 0]) \quad (2.38)$$

$E_{data}(D)$ is the data term of disparity map D, representing the sum of all pixels' cost $\sum_p (C(p, d_p)$. $E_{smooth}(D)$ is the smooth term, d_p is the disparity between pixel p on the left image and its suspected matching point on the right image, p' is the adjacent pixel point of p and $T[\cdot]$ equals 1 if the argument is true and 0 otherwise, Np represents the set of neighboring points of p. The $\mu(p, p')$ multiplier can be interpreted as the penalty of a discontinuity between p and p', and in this paper, it is the combination of the intensity difference and space distance, like (2.39) shows:

$$u(p,p') = P_1 exp(-\frac{\|p-p'\|}{\sigma_{sp}} - \frac{|I_p - I_{p'}|}{\sigma_I} \tag{2.39}$$

It is inspired by the bilateral filter [53], where P_1 is the maximal penalty for a discontinuous pixel, and $u(p,p')$ decreases when the intensity difference or space distance of pixel p and p' increases, which can preserve the discontinuity of the edge. σ_{sp} and σ_I balance the contributions of intensity and space distance to the discontinuity penalty. The problem of dense matching now converts to find a disparity map D that minimizes the global energy E(D). According to [54], it is an NP-hard problem. Graph cut and SGM are proposed in [52] and [54] respectively, and they can approximately minimize the energy function in polynomial time.

In this paper, we utilize the Semi-global matching algorithm to minimize the energy function; it is similar to [54]. However, different from [54], which adds a constant penalty P_1 for all pixels in the neighborhood of p when the disparity changes a little bit (that is, one pixel) and adds a larger constant penalty P_2 for all larger disparity changes, we calculate a dynamic penalty for each pixel in the neighborhood of p based on space and intensity differences, as (2.39) shows. Thus, we first need to calculate the penalty volume for the image.

The dimension of the penalty volume is $N \times H \times W$. *N* depends on the neighborhood window size, and *W*, *H* are the width and height of the image. Adjacent pixels share the same penalty, which can be used to reduce the size of the penalty volume in the *N* dimension, like Fig.2.23 shows. It is the 8-connection situation, where the window size is 3.

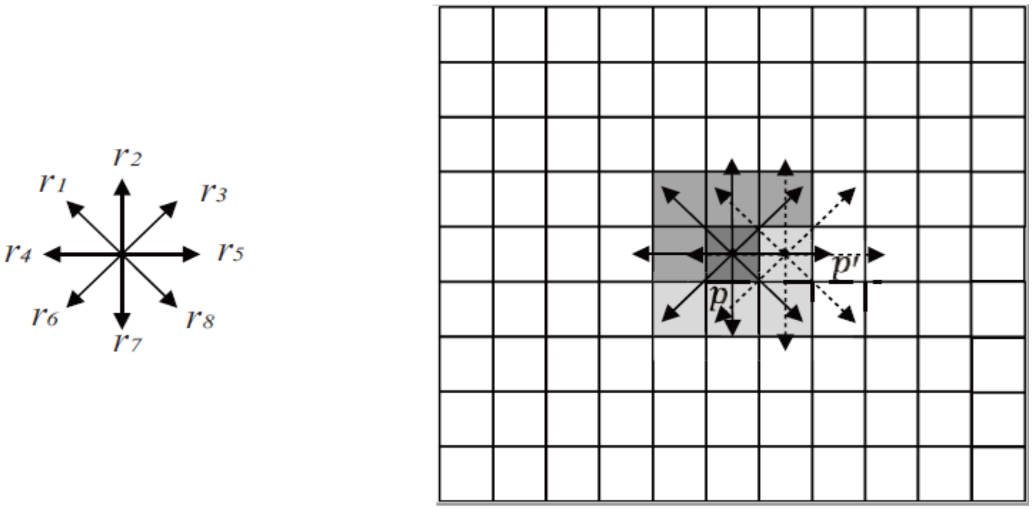


Fig.2.23 Demonstration of an 8-connection situation.

p': $(x+u, y+v)$is one adjacent pixel of p: (x,y) in the direction (u,v), where u, *v* are the horizontal and vertical displacements of the neighboring point with respect to the point p. We know $u(p,p') = u(p',p)$, thus the penalty of p in the *(u, v)* direction equals the penalty of q in the $(-u,-v)$ direction.

For point p, we have marked all its neighborhood points in 8 directions (in gray). If we calculate the penalty volume from the top left point of image, the penalty of adjacent points in the top left corner of p (in dark gray) has already been calculated. In other words, the penalty in r_1, r_2, r_3, r_4 directions have already been calculated; therefore, we just need to calculate and store the penalty of the remaining r_5, r_6, r_7, r_8 directions. For point p: (x, y), its penalty in the (u, v) direction equals the penalty of point p': $(x + u, y + v)$ in the $(-u, -v)$ direction, which can be used to search the penalty in r_1, r_2, r_3, r_4 directions from the former pixels' penalty vectors. We know r_i and (u, v) are two different forms of adjacent directions. If $r_i = (u, v)$, we can formulate the relationship between *i* and *u*, *v* as the following shows:

$$i = g(u, v) = \begin{cases} (v + \frac{wds}{2}) \times wds + u + \frac{wds}{2} + 1, v < 0||(v = 0, u < 0) \\ (v + \frac{wds}{2}) \times wds + u + \frac{wds}{2}), v > 0||(v = 0, u > 0) \end{cases} \quad (2.40)$$

i is the index of direction r_i, which can be calculated by function $g(u, v)$. wds is the window size of the neighborhood region. For any pixel in the image (except the first row and first column), we only calculate its penalty in the last half of the directions. The penalty $P(x, y)$ in the first half of directions r_i can be searched from the former pixels' penalty vectors, for which we have:

$$\begin{cases} P(x, y)_{ri} = P(x + u, y + v)_{rj} \\ i = g(u, v), j = g(-u, -v) \end{cases} \quad (2.41)$$

After we have established the penalty volume, the minimization method is similar to [54]. Fig.2.24 shows the summary of all the processing steps of our proposed method, including sparse matching and dense matching.

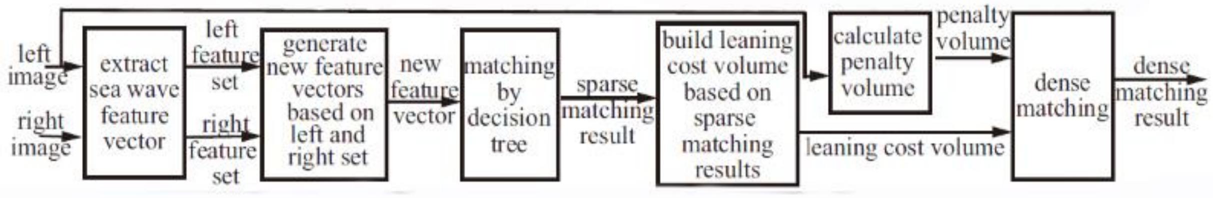


Fig.2.24 Workflow of the proposed method.

To complete dense matching, first, a leaning cost volume (size $W \times H \times Ds$), and a penalty volume $(N \times H \times W)$ need to be built in advance. Each element in the two volumes needs to be calculated, thus, the complexity is $O(WHDs + WHN)$. Then, we conduct cost aggregation similar to [54]. The calculation of smooth cost (cost aggregation) in one direction requires $O(Ds)$ steps at each pixel. Each pixel is visited exactly K (aggregation path number) times, which results in a total complexity of $O(KWHDs)$.

With the leaning cost volume, the complexity of the building cost volume decreases from $O(WHD_l)$ to $O(WHDs)$, For real long-distance sea surface images, *Ds* is approximately 20, *D_l* is approximately 600, and nearly 97% of running time and memory consumption are saved. In the cost aggregation process, by using the penalty volume, we can reduce the aggregation path number by half (compared to what the author of [54] used), which can save 50% in running time.

2.5.5 Dense matching result

Fig.2.25 shows one of the dense matching results of sea surface images by leaning cost volume. The image was taken at 17:00, with a monitoring distance of 8-14km. The energy minimization method used in the cost aggregation process is similar to the SGM method, except using dynamic penalties. (a) (b) are the left and right images and (c) is the disparity map of (a) and (b). The intensity of each pixel represents the disparity of each pixel between the left and right images. Higher intensity represents larger disparity, and the stripe-like areas in the bottom left and top right corners are disparity missing areas caused by the occlusion between the left and right images. Comparing the original image (a) and (b) with the disparity map (c), we can find that, consistent with the conclusion drawn in subsection 2.5.2, the intensity where there is a sea wave is larger than the intensity where there is the background. Since there is no planar structure on sea surface images, there is rarely area with the same disparity, it is consistent with figure (c). To validate the correctness of dense matching results, we choose a line on the left image (a). For each pixel on the line, we compute its matching pixel on the right image according to the disparity map (c).

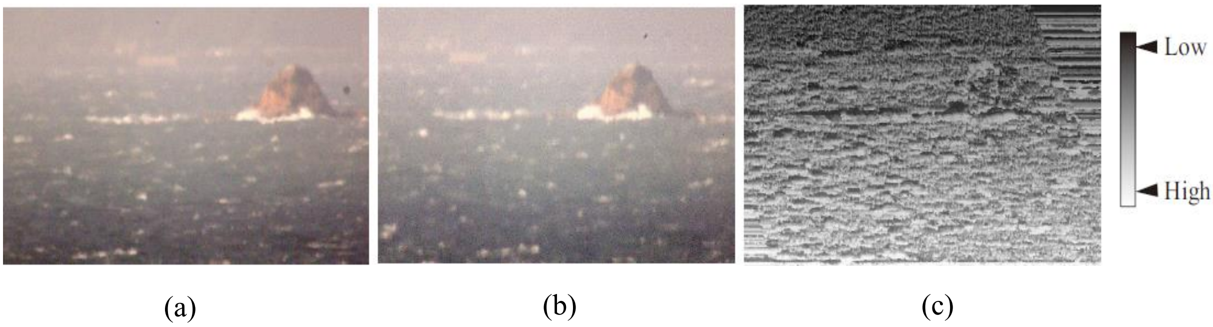


Fig.2.25 Dense matching results, (a), (b) are the left and right images and (c) is the disparity map of (a) and (b).

Fig.2.26 shows the matching results, and the two matched pixels are connected by a line. The matching accuracy is checked by manual operation. The black line represents correct matching. The red line represents incorrect matching, when the two linked points break the disparity continuity criterion. The matching accuracy of Fig.2.26 is 87.0%. In section 3, the relationship between *y* and *d* is shown in Fig.2.21 (a), and it is consistent to the Fig.2.26 result, in which the change tendency of the lines' slopes is consistent with the change of *y*.

2.6 Summary and Conclusion

In this section, the stereo matching by traditional matching methods are discussed. we proposed a matching method for long-distance sea surface image pair which lacks feature points and feature points in left, right images are not exactly alike. Our method combines the feature vector and decision tree algorithms to solve the problem of lacking feature points and allows

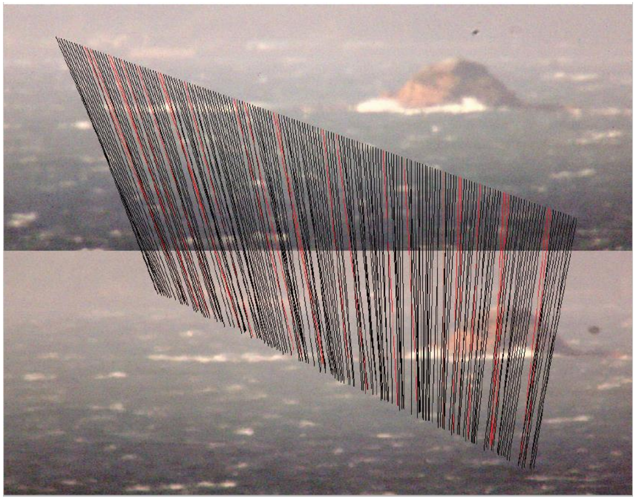


Fig.2.26 Demonstration of the matching result of pixel points on an oblique line, the top is the image taken by the left camera, and the bottom is the image taken by the right camera.

the existing of difference of correctly matched feature points as well as speeds up matching computation process. We also proposed a fast 3D verification idea according to the geometrical and spatial coherence of sea surface image to remove mismatch from primary matching results, experiment results indicate it can remove 100% mismatch fast.

A second step dense matching method is proposed based on the leaning cost volume and the dynamic penalty volume to decrease the searching region of dense matching and speed up the aggregation speed by minimization the established energy function. In order to reduce the influence of different shooting angles, the cost function combines the difference of intensity, gradient and census signal of the stereo image pairs. The experiment result shows that the proposed method can reach an accuracy of 87.0%, and it obeys the relationship formulated in this thesis.

Tsunami forecast requires us measure unusual sea level height changes as soon as possible, our proposed method decreases final feature vector dimension into 8D, as well as average comparison times into 4.3, which can rapidly improve the computation efficiency.

Chapter 3 Stereo matching by deep learning

With the development of artificial intelligence (AI), many scholars turned to complete stereo matching with neural network. Most of the traditional methods are very low accuracy compared to the AI-based approach. This can be observed in the standard benchmarking dataset, such as from the KITTI and the Middlebury, where AI methods rank at the top of the accuracy list. Additionally, the trend for solving computer vision problems uses AI or machine learning tools that become more apparent in recent years. Among this works, many of them focus on deep learning frameworks, which is one of the machine learning tools related to the convolutional neural network. Several mixed approaches between CNN based method and traditional handcraft method. In this thesis, we also applied these two kinds of network for our sea surface image stereo matching.

3.1 Introduction

Currently the rapid growth in computer vision provides various implementation of our day-to-day activities. It changed every aspects of our daily life. The decision-making process in our daily life is also affected by the development in computing technology. Currently, the researchers in the AI field are attempting to create the intelligent machine. In recent years, so many articles published in the area of AI make it become a concentrated topic in research area. Benefit from the huge development in computer hardware and software, the possibility of machine learning algorithms has been greatly exploited.

Deep learning has been developed more than thirty years. It becomes one of the focused branches of machine learning in recent years. The common definition of deep learning is a neural network containing more than two layers. This hierarchical learning requires neural networks with multiple layers to learn raw input data and transform it into something meaningful based on how we want to define the conclusion. Most of the implementation of a deep learning network is based on an artificial neural network that contains a hidden layer in addition to the input and output layer. There are four types of network architectures: unsupervised networks, convolutional neural networks (CNN), recurrent neural networks (RNN), and recursive neural networks.

Deep learning became popular in the area of computer vision after the propose of CNN in the computer vision area. The running CNN on the graphics processing unit (GPU) can improve the recognition rates in many vision benchmark databases such as MNIST, NIST SD 19, CIFAR10, and NORB.

Neural networks for stereo matching can be classified into supervised networks and self-supervised networks. Supervised network requires input ground truth to supervise the learning of the network. Zagoruyko and N. Komodakis et al. presented at CVPR 2015 an algorithm to extract features directly from binocular images using neural networks, which does not require the use of any artificially designed features and can combat various transformations between images, such as uneven illumination, different photographic views, etc., and concludes that neural networks have advantages in feature description that better than other algorithms. Experimental results also confirm that this network structure can achieve better results than human-designed features for some problems in the field of computer vision, and even better than some learned features [76]. Zbontar and LeCun proposed the use of convolutional neural networks on the computation of cost values for binocular stereo matching in 2015 [77]. The algorithm uses neural networks to compute the matching cost and uses cross-based cost aggregation and semi-global matching methods to correct the final disparity, as well as the left-right continuity principle to verify the occluded regions in the image, with The final experimental result has an error rate of 2.61% on the dataset KITTI, which is better than all the traditional algorithms at that time. Zhuoyuan Chen et al. proposed a data-driven neural network model [78], which is different from the model proposed by Zbontar et al. Abandoning the method of using DNN to obtain similarity in the network proposed by Zbontar, and improving the operation speed of the whole network by two orders of magnitude. The network is designed with a multi-scale feature extraction structure, capable of fusing multi-scale image information, and ranking third in terms of operational accuracy on the KITTI dataset. To solve the problem that deep learning relays excessively on the use of Siamese neural networks and other processing layers to achieve stereo matching resulting in excessive time consumption, Wenjie Luo et al. proposed a matching network capable of generating accurate matching results quickly, with a computation time of less than one second using GPUs [79], which generates a dot product layer for simply compute the features of the Siamese neural network output. The training process of the network transforms the matching task into a multi-classification problem, where each disparity corresponds to a classification, the network also outputs a calibration score, resulting in better matching results than the existing algorithms. Amit Shaked and Lior Wolf proposed an improved three-step matching mechanism for stereo matching [80]. A new highway network architecture for computing the matching cost at each possible disparity, based on multilevel weighted residual shortcuts, trained with a hybrid loss that supports multilevel comparison of image patches was proposed. A novel post-processing step was also proposed, which employed a second-deep convolutional neural network for pooling global information from multiple disparities. This network outputs both the image disparity map and a confidence in the prediction. The confidence score was achieved by training the network with a new technique called the reflective loss. The proposed pipeline achieves state of the art accuracy on the largest and most competitive stereo benchmarks, and the learned confidence is shown to outperform all existing alternatives.

Traditional stereo matching is usually divided into four major steps: cost calculation, cost aggregation, cost calculation, and disparity optimization. Non-end-to-end network structures apply neural networks to one or several of these four steps, which greatly reduces the training efficiency of neural networks and difficult to apply them to stereo matching. For this reason, researchers have proposed end-to-end neural networks that integrate the entire four steps into the one network, and researchers simply feed the network with data and let the network learn the corresponding parameters on its own. Dosovitskiy et al. first proposed the FlowNetCorr network structure to extend the application to optical flow prediction [81], which takes advantage of the powerful learning capability of deep neural networks and simplifies the whole prediction process to that directly input two frames and output the optical flow prediction results. Mayer et al. improved the FlowNetCorr network architecture and proposed the DisNet network architecture [82], which extended the application of FlowNetCorr to stereo matching. A scene flow prediction database called SceneFlow is also built, which laid the foundation for applying the end-to-end network architecture to stereo matching. Gidaris and Komodakis proposed a three-stage network structure to replace the traditional detection, prediction, and disparity correction processes [83], which improved the accuracy of disparity estimation compared with the traditional algorithm; Cheng et al. proposed a 3D CSPN network structure for stereo depth estimation based on the traditional CSPN network structure. S. Kim and D. B. Min et al. propose a new approach that imposes spatial consistency on the confidence estimation [84]. Specifically, a set of robust confidence features is extracted from each superpixel decomposed using the Gaussian mixture model, and then these features are concatenated with pixel-level confidence features. The features are then enhanced through adaptive filtering in the feature domain. In addition, the resulting confidence map, estimated using the confidence features with a random regression forest, is further improved through K-nearest neighbor-based aggregation scheme on both pixel and super-pixel level. To validate the proposed confidence estimation scheme, cost modulation or ground control points-based optimization in stereo matching is employed. Experimental results demonstrate that the proposed method outperforms state-of-the-art approaches on various benchmarks including challenging outdoor scenes. For supervised network, the most important step is the establishment of ground truth. Then we can design different network structure to extract feature map and different loss function to improve the final matching precision, as well as cost volume building methods to speed up the matching process.

Supervised neural networks require a large amount of well labelled ground truth to supervise the learning of the network, while many real-world applications, such as our sea surface image stereo matching, we are unable or difficult to obtain enough ground truth to meet the network training requirements, unsupervised networks do not need to input ground truth, and directly use the left view to generate the right view. The right image is used to control the learning process of the network, omitting the process of making the ground truth (a major challenge in neural networks). In the 2016 ECCV, Ravi Garg et al. proposed an unsupervised network structure for acquiring single-view depth map by reconstruction error and the principle of disparity continuity. Some experiments of this algorithm on the KITTI dataset achieve results that are not inferior to those of supervised networks [85]. C. Godard, and O. Mac Aodha et al. proposed a novel training method that allows the neural network to perform depth estimation using a single image without using ground truth [86]. They obtained the training loss of the network by using the image reconstruction error, but the accuracy of the obtained depth map would be low. To solve this problem, a new training loss that addes the disparity continuity information of left and right images to the definition of the loss to improve the accuracy and robustness of the algorithm is also proposed, which outperforms state of art algorithms on the KITTI dataset and even better than some supervised algorithms. Smolyanskiy, and A. Kamenev et al. proposed a semi-supervised stereo matching algorithm based on deep learning neural network, which is an improvement of GCNet. The activation function is Exponential Linear Unit (ELU) instead of the commonly used Rectified Linear Unit (RELU), which makes the training and computing speed of the network much faster, and finally the network defines a new argmax function instead of using the traditional soft argmax function. The network was validated on the KITTI2015 dataset [87]. Zhong et al. proposed a self-supervised network structure, but the results of network were lower than those of the supervised network due to problems such as occlusion between the left and right images and uneven illumination. Y. Luo and J. Ren et al. propose a novel network structure for the widely used monocular depth estimation problem that focuses only on direct regression from monocular images to obtain the desired results and leads to unsatisfactory final results [88], which reformulates the monocular depth estimation problem into two subproblems, a monocular synthetic image and a stereo matching problem. Adding geometric constraints to the inference process can reduce the reliance on ground truth. The entire network can still be trained in an end-to-end manner, and can extremely improve the final accuracy of the algorithm, which achieves the best experimental results on the KITTI database for all monocular depth predictions, and is better than the block-based stereo matching results with only a small training data set.

3.2 Stereo matching with supervised network

Convolutional neural networks have been shown to perform very well on high-level vision tasks such as image classification, object detection and semantic segmentation. According to the former subsection, we find CNNs have been applied to low-level vision tasks such as optical flow prediction recently. In the context of stereo estimation, they utilize CNN to compute the matching cost between two image patches. In particular, they used a Siamese network which takes the same sized left and right image patches with a few fully-connected layers on top to predict the matching cost. They trained the model to minimize a binary cross-entropy loss. Some other research works also investigated different CNN based architectures for comparing image patches. They concatenated on the left and right image patches as different channels works best, at the cost of being very slow. In our work, we also utilize the Siamese network to generate the feature map of left and right sea waves and built a supervised network architecture to match the waves.

For the stereo matching with supervised network, network structure is usually applied to some parts of stereo matching pipeline, such as feature extraction module, cost volume aggregation module etc. or all the necessary modules. And for stereo matching of sea surface images, we apply the network to the sea wave extraction module, feature map generation module and sparse matching module. Before we establish the specific network structure, we must make two types of training sets for sea wave extraction and sparse matching respectively.

The sea wave extraction module is established based on Region Proposal Network (RPN), the training data set is made by the extraction results of traditional sea wave extraction method. Feature map generation module is based on the Residual Networks (ResNets), convolution operation between target template and matching search template is used to generate correlation feature volume. Two different convolution network modules are designed to generate matching confidence and the final matching result. The training set of this matching module is made by our previously introduced sparse matching result. In the following subsection, we will give detailed introduction of these modules.

3.2.1 Labeling the training sets

Ground truth is one of the keys of supervised network. It is known as the true and real information provided by directly observation and measurement, opposed to the information achieved by inference. For supervised network, it acts as the supervisor to control the learning direction of the network. We usually build training sets from the ground truth to training established network. There are two types of training sets in this paper, one is for sea wave extraction and the other is for sparse matching. As network learning needs large amounts of training samples, it is impossible to manually measure all the ground truth, thus we made some of the training samples based on our previous work.

Firstly, we will introduce the training set by manually labelling. It is the sparse matching training data set, sea waves on the sea surface images are selected by the member in our lab, then the matching pairs and non-matching pairs are also manually checked, for matching waves, the matching labels are set to 1, and for the non-matching waves, the matching labels are set to 0. Fig.3.1 shows the data set structure of manual labeling results, it is combined of 532 pairs of non-matching waves and 532 matching waves. The test data set is combined of 50 pairs of non-matching waves and 50 matching waves.

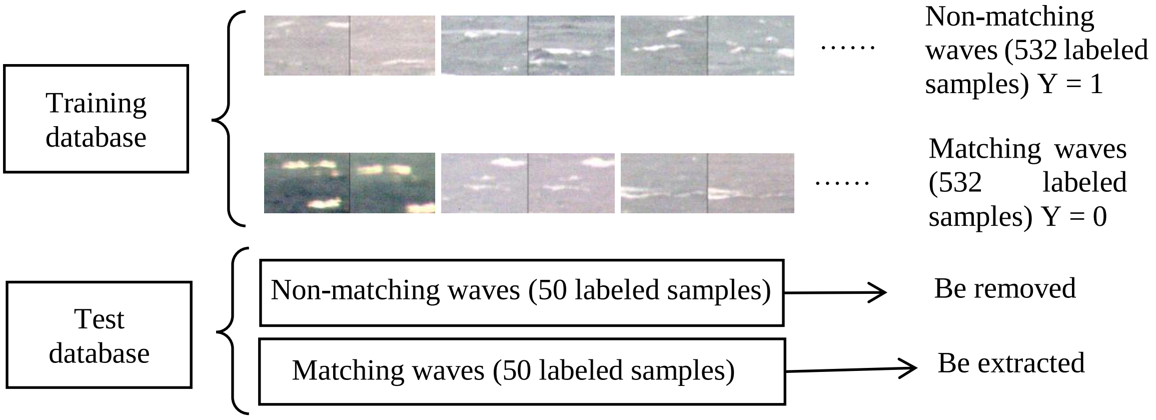
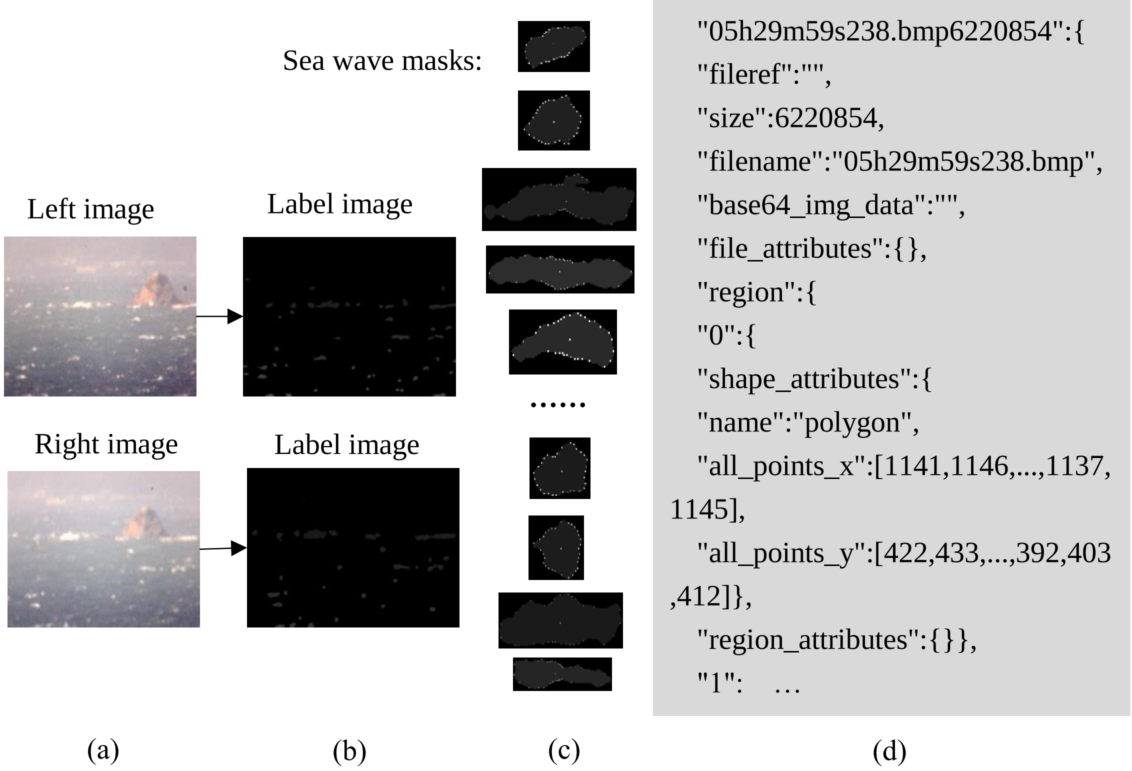

Training database
……
Non-matching waves (532 labeled samples) Y = 1
……
Matching waves (532 labeled samples) Y = 0
Test database
Non-matching waves (50 labeled samples)
Be removed
Matching waves (50 labeled samples)
Be extracted

Fig.3.1 The manual labelled training set.

Next, we will introduce the training data set made of manually inspection and traditional sea wave extraction method. It is used to supervise the learning of network that used to extract the sea waves from the sea surface images. As manually selected enough sea waves from sea surface images for network training is a labor-intensive and time-consuming task, we divide this work into two parts: 1) rough extraction of waves using traditional extraction algorithm; 2) manually check the extraction results and revise the wrongly extracted results.

Fig.3.2 shows one of the training sample made by this method, the sea waves are extracted by the extraction method proposed in our previous work. Each of the sea wave is labeled by a mask of polygon with 32 vertices, the labelled sea wave masks are saved in a json file with the information of sea wave size and mask vertices. 2556 sea wave masks are extracted for network training. In fig.3.2, there are four columns, from the left to right are original images, sea wave extraction results by extraction algorithm and manually inspection, marked sea wave masks and the content of labelled json file.



"05h29m59s238.bmp6220854":{
"fileref":"",
"size":6220854,
"filename":"05h29m59s238.bmp",
"base64_img_data":"",
"file_attributes":{},
"region":{
"0":{
"shape_attributes":{
"name":"polygon",
"all_points_x":[1141,1146,...,1137,
1145],
"all_points_y":[422,433,...,392,403
,412]},
"region_attributes":{}},
"1":   …

Fig.3.2 The labelled training set, (a) is the original stereo image pair, (b) is their extraction result, (c) is the cropped and marked result of sea waves, and (d) is the final output file

Finally, we will introduce the training data set used for sparse matching network learning. It is also made by traditional sparse matching method and manually inspection. Firstly, the rough matching results are produced by traditional matching method proposed in this thesis, then the matching results are manually inspected, incorrect matches are removed from the primary results, finally we selected the correctly matched wave pairs by a cutting programing, saving the left and right sea waves with same file names. Fig. 3.3 shows the training set labelling process, (a) are the original images from left and right camera views, the red boxes marked our interested sea waves, which will eventually be made into a training matching sample, (b) is the sparse matching result of these two images by traditional matching result and manual inspection, and our interested two sea waves are correctly matching, (c) is the mask of correctly extracted and matched waves, our interested two sea waves are marked with the same label, (d) is the final matched sample made by our interested two sea waves, the correctly matched sea waves from left and right images are named with same figure number. Need to note is that we do not

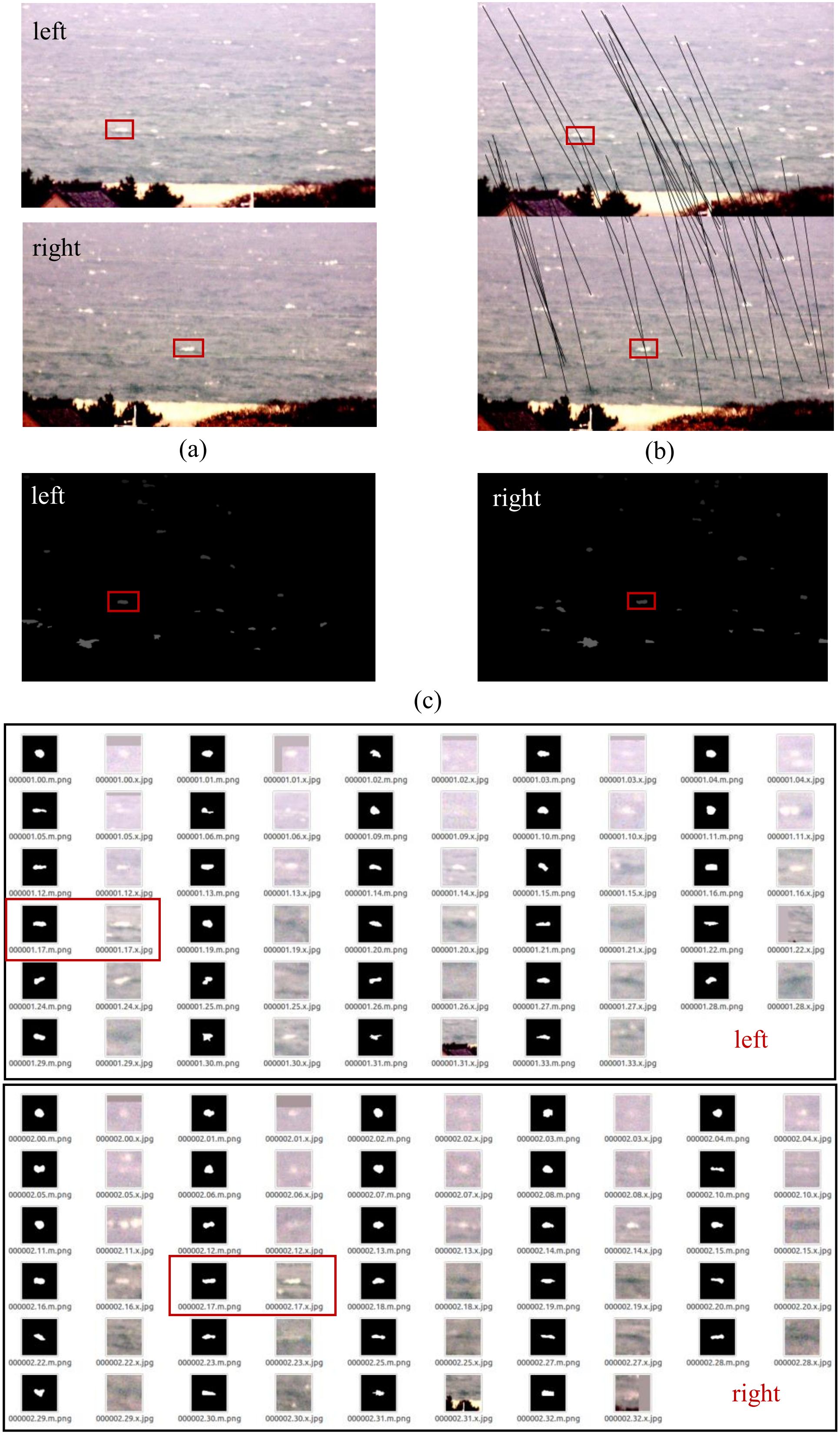


Fig.3.3 The process of making the training set, (a) is the original image pair, (b) is their matching result, (c) is the mask generated based on the matching result, and (d) is the final training sample.

make the negative samples in this training set because we will made negative samples during the learning process by randomly choose left and right sea waves from two correctly made samples.

These are the main training data sets made for sea surface image sparse matching. They are made by manual selection and traditional method results. The first and the third data sets are used for sparse matching from two aspects: one is transforming the matching problem to classification problem and the other is transforming the matching problem to object tracking problem. The second data set is used for sea wave extraction because it is time consumption if we directly generate bounding box from original image, then judge if it is sea wave bounding box and if it has corresponding on another image.

3.2.2 Basic network block

Convolutional neural networks (CNNs) are inherently limited to model geometric transformations due to the fixed geometric structures in their building modules. Thus, deformable convolution network is proposed to enhance transformation modeling capability of CNNs. They are built based on the idea of augmenting the spatial sampling locations in the modules with additional offsets and learning the offsets from the target tasks, without additional supervision. They can readily replace their plain counterparts in existing CNNs and can be easily trained end-to-end by standard back-propagation, giving rise to deformable convolutional networks. Extensive experiments validate the performance of our approach.

The 2D convolution consists of two steps: 1) sampling using a regular grid $\mathcal{R}$ over the input feature map *x*; 2) summation of sampled values weighted by *w*. The grid $\mathcal{R}$ defines the receptive field size and dilation. For example, $\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$ defines a 3×3 kernel with dilation 1. For each location p_0 on the output feature map y, we have:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \qquad (3.1)$$

Where p_n enumerates the location in $\mathcal{R}$. In deformable convolution, the regular grid $\mathcal{R}$ is augmented with offsets $\{\Delta p_n | n = 1,\ldots,N\}$, where $\mathrm{N} = |\mathcal{R}|$, equation (3.1) becomes:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \qquad (3.2)$$

Now, the sampling is on the irregular and offset locations $p_n + \Delta p_n$. As the offset Δp_n is typically fractional, Eq. (3.2) is implemented via bilinear interpolation as:

$$x(p) = \sum_q G(q,p) \cdot x(q) \qquad (3.3)$$

where p denotes an arbitrary (fractional) location ($p = p_0 + p_n + \Delta p_n$ for Eq. (3.2)), q enumerates all integral spatial locations in the feature map *x*, and $G(q,p)$ is the bilinear interpolation kernel. Note that *G* is two dimensional. It is separated into two one dimensional kernels as:

$$G(q,p) = g(q_x, p_x) \cdot g(q_y, p_y) \qquad (3.4)$$

where $g(a,b) = max(0,1 - |a - b|)$. Eq. (3.3) is fast to compute as $G(q,p)$ is non-zero only for a few qs. Fig.3.4 shows the illustration of deformable convolution.

As illustrated in Fig3.4, the offsets are obtained by applying a convolutional layer over the same input feature map. The convolution kernel is of the same spatial resolution and dilation as those of the current convolutional layer. The output offset fields have the same spatial resolution with the input feature map. The channel dimension 2N corresponds to N 2D offsets. During training, both the convolutional kernels for generating the output features and the offsets are learned simultaneously. To learn the offsets, the gradients are backpropagated through the bilinear operations in Eq. (3.3) and Eq. (3.4).

We also involved the deformable convolutional network for the establishment of feature map extraction module to increase the precision of mask because most of the sea waves' shapes are nearly circular.

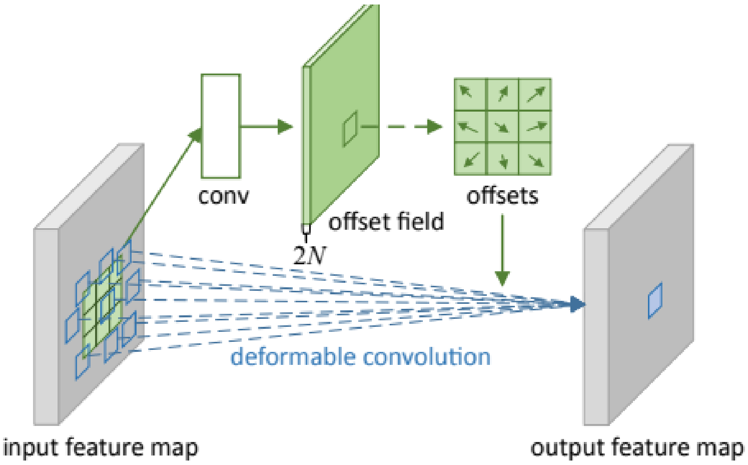


Fig.3.4 The illustration of 3×3 deformable convolution.

Deformable convolutional network is used as the basic network unit to form the basic network block: Residual Networks (ResNets). The motivation of building the ResNets is that with the development of Deep Convolution Neural Network (DCNN), researchers find that Deeper Networks fail to perform better than their Shallow counterparts due to that with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly, verified by experiments that such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error. To address the degradation problem, ResNets structure is introduced. A building block is defined as the following equation:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{3.5}$$

Here x and y are the input and output vectors of the layers considered. The function $\mathcal{F}(x, \{W_i\})$ represents the residual mapping to be learned. Fig.3.5 shows the structure of a building block, it has two layers, $\mathcal{F} = W_2\sigma(W_1x)$ in which σ denotes activation function of ReLU, the operation $\mathcal{F} + x$ is performed by a shortcut connection and element-wise addition, another nonlinear transformation is adopted after addition by using ReLU function. This kind of network does not have any parameters and is just there to add the output from the previous layer to the layer ahead, thus we take it as the basic block for establishing different modules of

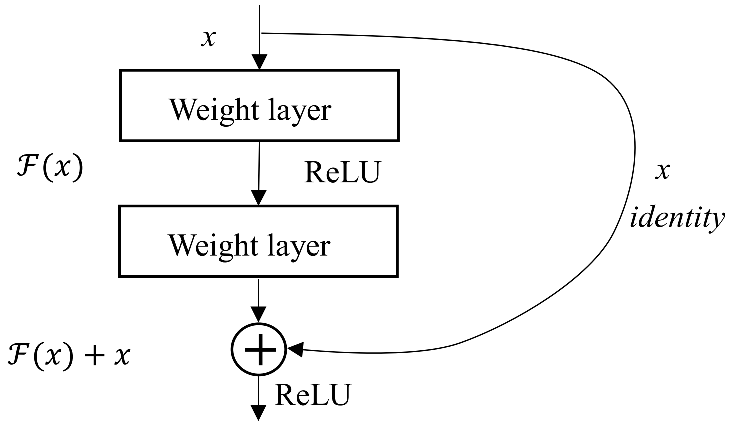


Fig.3.5 illumination of Residual block.

our sea wave extraction and sparse matching network.

Based on the ResNets blocks, we can build different feature map generation modules. The most commonly used structures are: 18-layer ResNets, 34-layer ResNets, 50-layer ResNets, 101-layer ResNets and 152-layer ResNets. The difference between them is the number of ResNets blocks, the more ResNets blocks there are, the more layers there are in the network. The following Table 4. shows the specific structures of these network.

Table 4. Structures of each type ResNets

Layer name	Output size	18-layer	34-layer	50-layer	101-layer	152-layer
Conv1	112×112	7×7, 64, stride 2				
		3×3, maxpool, stride 2				
Conv2_x	56×56	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$
Conv3_x	28×28	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times8$
Conv4_x	14×14	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times$ 36
Conv5_x	7×7	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$
FC	1×1	Average pool 1000-d fc, softmax				

For the feature map generation of sea surface images, we take the 50-layer ResNets as feature extraction module because we think the depth of the network is suitable for us to output a relatively precious result, neither too shallow to weaken the capability of the network nor too deep to cause the training process to converge very slowly. Due to the lack of training sample, we performed the network porting to refine the parameters for specific problems directly on the training results of the Coco dataset. Specifically, the network weight parameters in the second and third layers are frozen, and only the parameters in the first and fourth layers are adjusted during the training process.

3.2.3 Sea wave matching with Siamese Network

As we have introduced the basic unit and detailed structure of feature map generation module, now, we would like to introduce the method of sea wave extraction based on Siamese network, it is one of the important steps for sparse matching of sea surface images. A Siamese neural network (sometimes called a twin neural network) is an artificial neural network that uses the same weights while working in tandem on two different input vectors to compute comparable output vectors. The key of establishing this network is that we need to perform same operation on different inputs. We establish a Siamese network framework consisting of two ResNets for the left and right image feature map generation. The convolution layers in these two ResNets share the same weights.

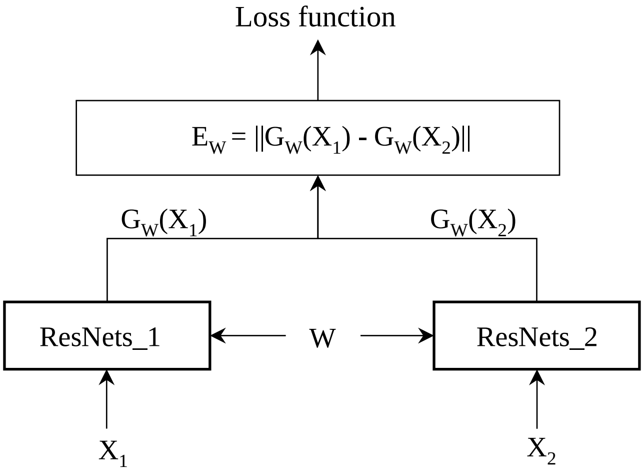


Fig.3.6 A common Siamese network structure.

One way two training this network is to design a contrastive loss function parameterized by outputs of the two ResNets, using the Adam optimizer to adjust the parameter learning process and minimize the loss function. A common Siamese network structure can be seen in the following Fig.3.6.

As the outputs of two networks, $G_W(X)$ is a function only respect to parameter W. With the similarity metric Ew between two inputs, the contrastive loss function can be defined. Siamese Network makes use of an optimizer to drive the learning process and minimize the loss function so that the similarity metric of the two inputs can be polarized. More precisely, the similarity metric becomes small for pairs of similar inputs and large for pairs of different inputs. ResNets can map the input to points in a low dimensional space and the function is necessary during the quantization process of similarity metric. Besides, the main advantage of CNN is that it can learn optimal shift-invariant local feature detectors and build representations that are robust to geometric distortions of the input image. So, we choose to form the Siamese network by CNN.

Next step is to form a Siamese network system. We make use of the open-source software library Tensorflow to establish the Siamese network framework whose structure can be seen in the following Fig.3.7. The Siamese network we formed is composed of two ResNets. The type of the ResNets is 50-layer ResNets with deformable convolutional unit. You can find the specific structure of it in Table 4. It contains four main residual blocks, transforming the original image into 1000-d 1×1 feature maps. The fully connected layer 1 contains 1000 neurons and the fully connected layer 2 has 128 neurons as the output of feature extraction module. We set the similarity metric *Ew* as follows:

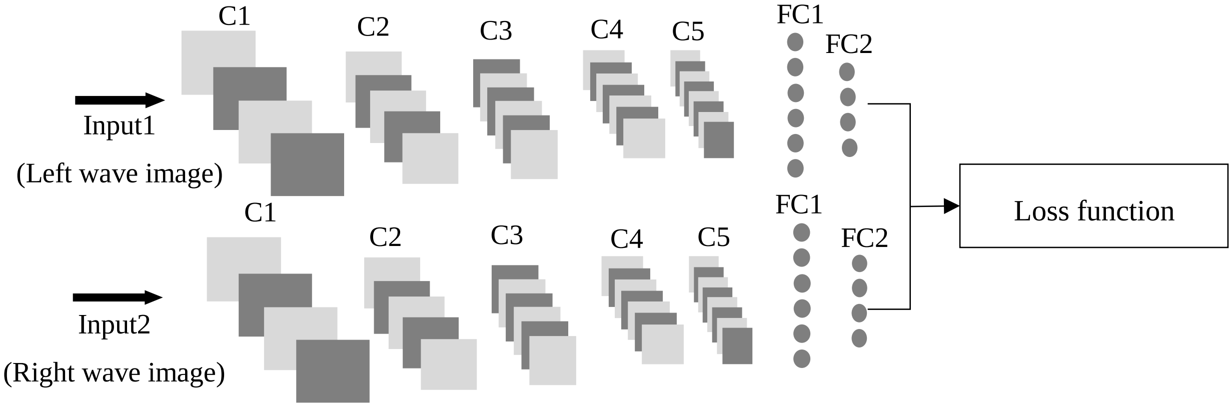


Fig.3.7 The Siamese network framework.

$$E_W = \sqrt{\left(G_W(X_1) - G_W(X_2)\right)^2 + e^{-6}} \tag{3.6}$$

$G_W(X_1)$ and *$G_W(X_2)$* represent the outputs of two ResNets. Then we can define the contrastive loss function as follows:

$$L_W = (1 - Y) \times E_W + \mathrm{Y} \times (max\{(5.0 - E_W), 0\})^2 \tag{3.7}$$

where *Y* means the label of samples. *Y*=0 represents matching samples while *Y*=1 represents non-matching samples. We input batches of 32 samples into the Siamese network system and train it with the Adam optimizer for 10000 times. After that, we can make use of the test database to verify the effect of sea wave matching.

This kind of Siamese Network usually used when the left and right sea waves has been exactly selected from the original image. We have transformed the matching problem to a classification problem. But if we only have the template of left sea wave and the template of right sea wave is not precious or lacked, how to match them by Siamese network is a problem. In the next content, we will discuss the solution of this problem.

Instead of inputting the outputs of conv5_x into an average pool layer to generate a 1000-d feature vectors, we design a convolutional production layer to generate a correlational feature volume, then use this volume to generate the matching confidence of each locations on the right images, an accurate mask is also generated by taking use of this volume. Fig.3.8 shows the structure of this network. The feature extraction module is the same as the previous network,

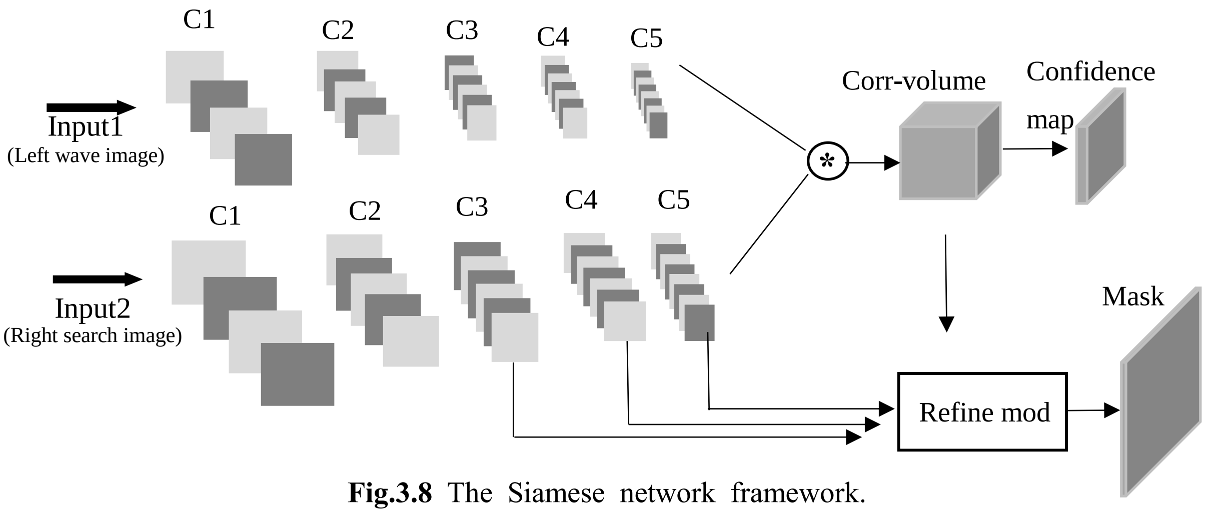


Fig.3.8 The Siamese network framework.

and the convolutional production is conducted between the left template feature map and the much larger right feature map. The convolutional production is defined as the following:

$$corr\_volume(x,y) = \sum_{dx=-kx}^{kx} \sum_{dy=-ky}^{ky} k(dx,dy) f(x+dx, y+dy) \tag{3.8}$$

Here, $k(dx,dy)$ is the element in left feature map, kx, ky are half of the width and height sizes of left feature map, $f(x,y)$ is the element in the right search image feature map. In this case, left feature map acts as kernel of convolution operation, and based on the knowledge of signal processing, the matched location will generate the largest response value. After a simple convolutional layer, we transform the corr-volume to classification confidence.

We also add a refinement module to produce the matching mask on the right image, it is also constructed by convolutional layers. It is easy to understand that the corr-volume contains the final matched information and the feature maps generated by the ResNets structure contains the target information. Combining this information can achieve the location and shape information of the final matched mask. The output sizes of feature maps from different ResNets parts decrease with the depth of layers. Large size feature maps are computed by primary layers' parameters, each element is generated from its adjacent pixels on the original image, the feeling field of the feature map is relatively small. Small size feature maps are computed from large size feature maps by convolutional operation, thus their feeling fields are larger than large size feature maps. Therefore, for the feature maps of ResNets, large size feature maps contain more detail information of sea wave while small size feature maps contain a large range of profile information. The refinement module is established based on this characteristic of feature maps and corr-volume.

Fig.3.9 shows the configuration of the refinement module, f_1, f_2, f_3 represent the output feature maps of Conv3_x, Conv4_x and Conv5_x in ResNets. We take corr-volume as a convolutional kernel and perform convolutional operation between this volume and feature maps from deep to shallow to generate the final predicted mask. The up-sampling function is realized by the pytorch library function. After convolutional operation with low resolution feature maps, the output results are up-sampled to higher resolution and low channels. Then the output results are convolved with the higher resolution feature maps again. The convolutional operation is conducted by additive method without multiplication.

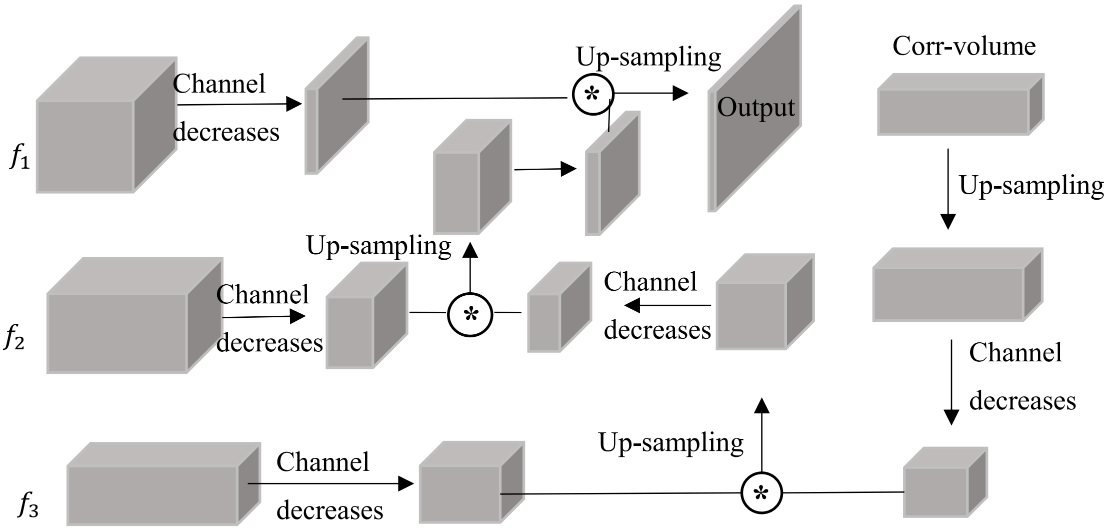


Fig.3.9 The refinement network framework.

Until now, we have introduced all the necessary modules of the sparse matching network, there are two outputs, one is the location of the matched mask and the other is the shape of the mask. To drive the network, we must define the loss function to adjust the parameters of this network. We define the loss function by combining the location loss and the mask loss. The following equation is the definition of loss function:

$$Loss = -w_l \frac{1}{N}\sum_i^N [y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] + w_m \frac{area\ of\ overlap}{area\ of\ union} \tag{3.9}$$

Here, w_l and w_m are the weights of location loss and mask loss, N is the size of the confidence map, y_i is the ground truth of each mask location, $\hat{y}_i$ is the predicted confidence of this location, easy to find that the location loss is the cross-entropy loss. We have labelled the ground truth for each sea wave and the network also outputs a predicted mask of each sea wave, the overlap of these two elements is $area\ of\ overlap$, the union of them is $area\ of\ union$, thus the ratio of them can measure the similarity between predicted mask and ground truth.

Training this network with the third dataset introduced in subsection 3.2.1, selecting the SGD as the optimizer, we can adjust the parameters in the network. Lacking the ground truth causes us to fix some parameters and only adjusting the parameters in refinement module and confidence generation module.

3.3 Sea wave extraction module

Both the networks introduced before have a prerequisite that the sea waves have been selected from the original images with precious or rough bounding boxes. In practice, it is impossible to control the inputs being appropriate images with only single sea wave in them. Therefore, a sea wave extraction module before we matching them is necessary. In this subsection, we will introduce the sea wave extraction module based on the improvement of Mask R-CNN structure [89]. It is made up of an bounding box generation module and a classification module.

The Region-based CNN (R-CNN) approach to bounding-box object detection is to attend to a manageable number of candidate object regions and evaluate convolutional networks independently on each RoI, RPN advanced this stream by learning the attention mechanism. Its purpose is to propose multiple objects that are identifiable within a particular image. In the next content, we will give a detailed introduction of bounding box generation module based on RPN. It completes three steps to find out the object bounding box: 1) generate anchor boxes with different scales and ratios; 2) classify each anchor box to foreground and background; 3)learn the shape offsets for anchor boxes to fit them for objects.

Every point in the feature map generated by the ResNets is an anchor point. For every anchor point, we generate its corresponding anchor boxes with two parameters: scales and aspect ratios. During generating the anchor boxes, another parameter also needs to be considered: the stride. Usually, we define this parameter to be the value of image decreasing ratio between the original image and feature map. For an original image of size n*n, the number of anchor boxes generated by the RPN can be calculated by the following equation:

$$an\_num = \frac{n}{stride} \times num\_scale \times num\_ratio \qquad (3.10)$$

The Fig.3.10 shows the generation of anchor boxes, the red points on the original image are the locations of each anchor and the distance between the adjacent point is the stride, at each anchor point, we generate anchor boxes with different scales and ratios.

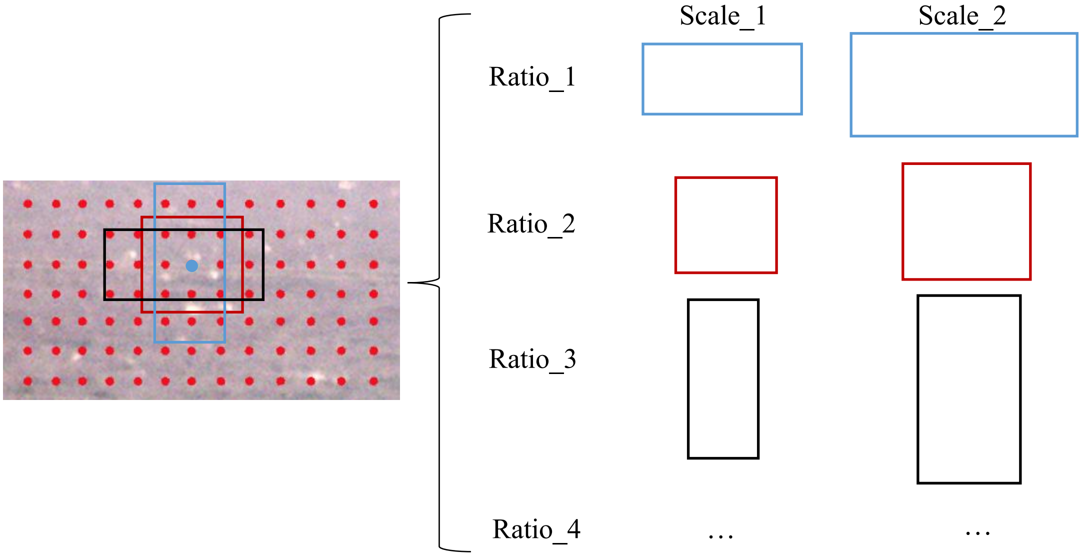


Fig.3.10 The generation of anchor boxes.

After we have generated the anchor boxes, we need to judge if the anchor box belongs to foreground or background, at the same time we need to learn the offsets for the foreground boxes to adjust for fitting the objects. Therefore, we add two convolution layers to transform the anchor boxes volume to class score map and offset value map. Those two layers are called rpn_cls_score layer and rpn_bbox_pred layer and the architecture is shown in Fig.3.11. The

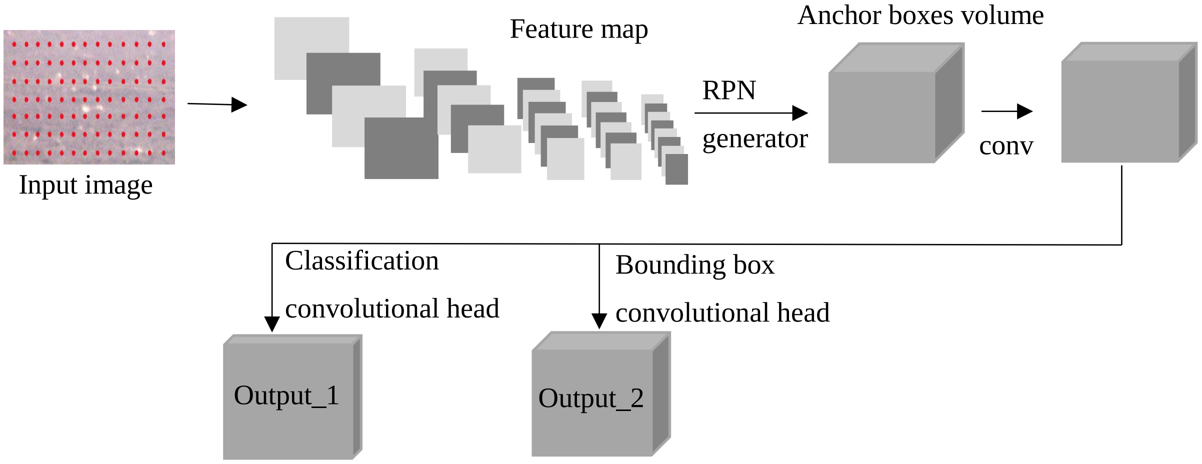


Fig.3.11 The configuration of object detection network.

offsets for each anchor contain four values: x, y, w, h, where (x, y) is the center of the box, w and h are width and height. To adjust the parameters in the convolutional layers in classification head and bounding box head, we need the ground truth training set and we use the second data set described in subsection 3.2.1. Anchor targets are generated by comparing the anchor boxes with ground truth boxes. This process is called anchor target generation. In anchor target generation, we calculate the overlap ratio of ground truth boxes with anchor boxes to check if it is foreground or background and then the difference in the coordinates are calculated as targets to be learned the bounding box head. The loss function for classification head and bounding box head are cross-entropy loss and smooth l1 loss, same as the loss function in subsection 3.2.3.

As the second data set described before has the mask of sea waves, it is a much finer spatial layout of a sea wave. Thus, we can add an extension branch for predicting an object mask (Region of Interest) in parallel with the existing branch for bounding box recognition and offset prediction.

The key element of mask prediction is the pixel-to-pixel alignment, which is the main missing piece with region proposal on feature map. it adopts a two-stage procedure by adding an extend branch to predict the mask for each region of interest. This is in contrast to most recent systems, where classification depends on mask predictions. Furthermore, it is simple to implement and train given the former framework. Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation. Fig.3.12 shows the whole structure of the network. The output after the anchor box volume is feed into three head branches to generate the classification, bounding box and mask prediction. Note that during the mask prediction parts, RoI pool is replaced by RoIAlign which helps to preserve spatial information which gets misaligned in case of RoI pool. RoIAlign uses binary interpolation to

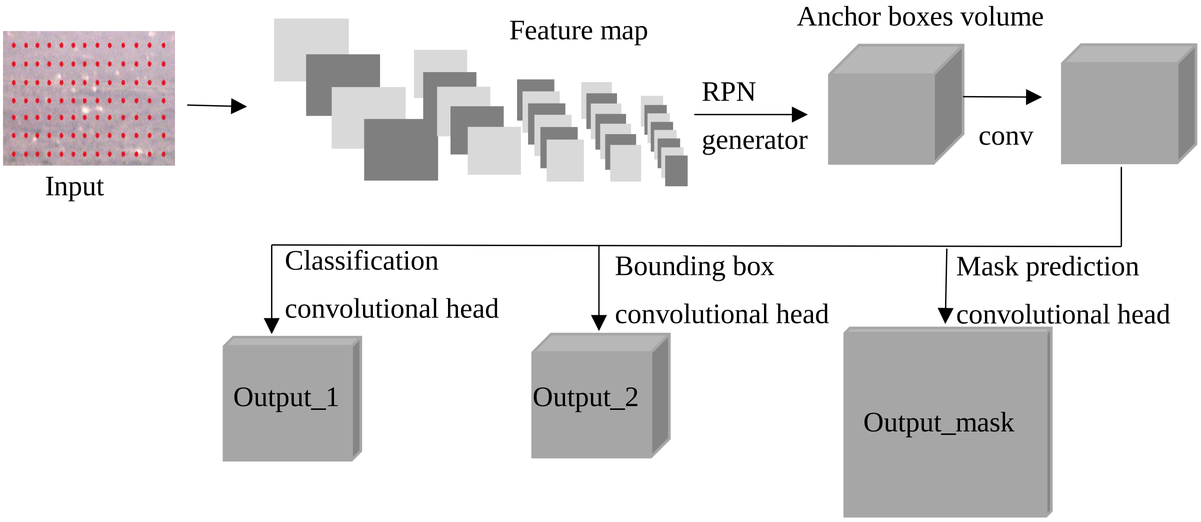


Fig.3.12 The configuration of sea wave extraction network.

create a feature map that is of fixed size. The output from RoIAlign layer is then fed into mask head, which consists of two convolution layers to generate mask for each RoI and then segment an image in pixel-to-pixel manner.

Until now, we have introduced all the modules of the sea wave extraction. In practice usage, we firstly use this network to select all the sea waves from the sea surface image. Feed the outputs of this network to the sparse matching network to complete the final matching. In the next subsection, we will give the sea wave extraction and matching results as well as its comparison with traditional method.

3.4 Stereo matching with self-supervised network

Sparse matching with network requires two conditions: 1) ground truth to supervise the learning process and 2) sea wave extraction from original image. To break these two limitations and generate a dense matching result attracts many researchers' enthusiasm, many types of self-supervised networks are proposed. This kind of network can be trained without ground truth, but detailed designed loss functions are used to control the parameter adjusting direction. Need to note that, we have built a self-supervised network for sea surface image dense matching, but due to many uncertain reasons, the output result is not ideal.

The dense matching for sea surface images is similar to other binocular systems' stereo matching problem, but it has a characteristic that the disparity within this kind of image pairs is much larger than others. In the next content, we will introduce a common pipeline for dense matching by deep learning of stereo images, an alignment module inspirited by RoIAlign in Mask R-CNN is built to make a rough alignment before the cost volume establishment.

3.4.1 Dense matching pipeline

Standard pipeline of the unsupervised stereo dense matching contains four modules: feature map generation module, cross feature volume building module, feature matching module and image warping module. Image reconstruction loss and left and right consistency loss are generated as the loss function to control the learning process. Fig.3.13 shows the whole pipeline.

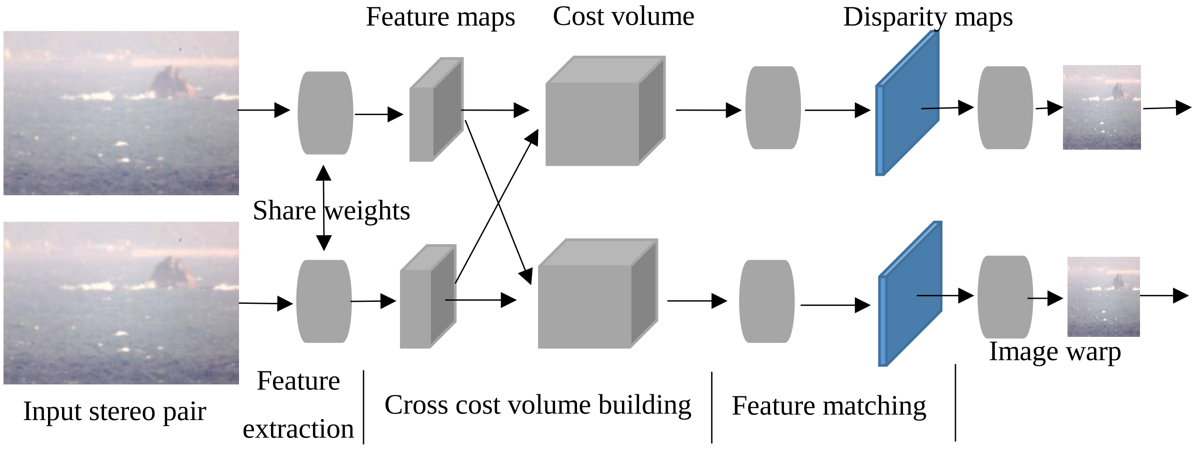


Fig.3.13 unsupervised stereo matching pipeline.

The feature extraction module can be selected as the same mentioned before, a ResNets based Siamese network. Different from sparse matching, we do not conduct the selection of RoI on the extracted feature maps. Instead, a cross cost volume is built based on the left and right feature maps. There are many methods can be used to build these cost volumes, such as absolute value error, normalized cross correlation and directly concatenate etc. This module along with the feature matching process are the focal points of researching stereo matching with network. Many algorithms are proposed to complete feature matching and generate a relatively accurate disparity map. Here, we introduce an embedding method for stereo matching, which makes a detailed design of building the cost volume and completing feature matching.

Given a pair of stereo images $\{I^L, I^R\}$, a typical cost volume is computed by the matching cost: denoting patches $I^L(p)$ and $I^R(p-d)$ centered at $p^L = (x, y)$ and ${p_d}^R = (x-d, y)$, we generate the cost $S(p,d) = f(I^L(p), I^R(p-d))$ for each p with different disparity $d = (d, 0)$. The cost serves as an important initialization, generally it needs to be processed by a non-local refinement such as cost aggregation or filtering. In this paper, the cost is computed by a 4-layer neural network and an inner-product layer. The inner-product layer is proposed to learn a large response in the case of correct matching or a small response for incorrect matching, it is different from traditional absolute value or normalized cross correlation. We also have an embedding module, which can fuse the coarse scale cost with fine scale cost. Denoting $I^L_\downarrow(p), I^R_\downarrow(p-d)$ as patches at the coarse scale, we have:

$$S(p,d) = w_1 < f(I^L(p), I^R(p-d)) > + w_2 < f(I^L_\downarrow(p), I^R_\downarrow(p-d)) > \qquad (3.11)$$

As we know, the choice of patch scale and size is very tricky: large patches with richer information are less ambiguous, but more risky of containing multiple objects and producing blurred boundaries; small patches have merits in motion details, but are very noisy. Accordingly, we use a weighted ensemble of two scales to combine the advantages of two scales.

Fig.3.14 shows the fusing process and the establishment of cost volume. The inputs are two scales feature maps from the former ResNets output, centered at $p^L = (x, y)$ and $p_d{}^R = (x - d, y)$. The cost value $< f(I^L(p), I^R(p - d)) >$ is computed by an inner-product layer, the vote module completes the function of equation 3.11 by a 1 × 1 × 2 convolutional layer for a weighted ensemble. Deep embedding produces high-quality initialization, which can be refined with an MRF-based stereo algorithm to obtain the state-of-the-art dense disparity estimates.

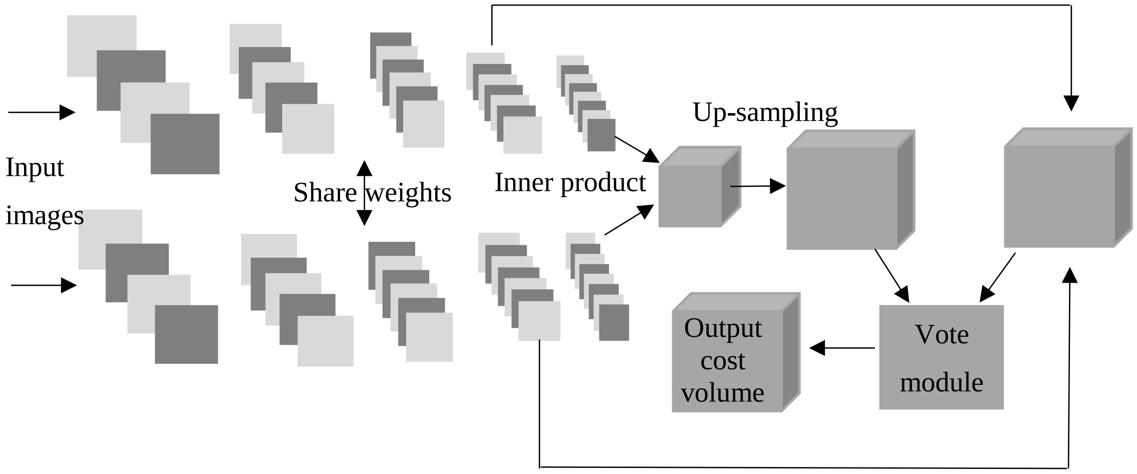


Fig.3.14 Building cost volume by fusing two scale inner product results.

Another important step of self-supervised matching is the definition of loss function. Different from supervised network, we do not have the ground truth to control the whole learning process, but inspirited by traditional matching method, we take use of image reconstruction error to control the learning process. In this thesis, we define a loss combine three main terms,

$$C = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r) \tag{3.12}$$

Where C_{ap} encourages the reconstructed image according to the learned disparity map to appear similar to the corresponding training input, C_{ds} enforces smooth disparities that the adjacent pixels' disparity values must be continued, and C_{lr} requires the predicted left and right disparities to be consistent. Each of the main terms contains both a left and a right image variant. Next, we will define each element of our loss in terms of the left image. The right image versions require to swap left for right and to sample in the opposite direction.

During the training process, the network reconstructed the left/right image by warping the original right/left image (the opposite stereo image) based on the generated disparity map. We use bilinear sampling where the output pixel is the weighted sum of four input pixels to warp image which is locally fully differentiable and integrates seamlessly into our fully convolutional architecture. This means that we do not require any simplification or approximation of the cost function.

The reconstruction error is computed by a combination of $L1$ loss and SSIM term between the input original image I_{ij}^l and its reconstruction $\hat{I}_{ij}^l$ like the following equation shows, where N is the total number of pixels, α balance the weight of absolute appearance difference and SSIM value, the detailed definition of SSIM can be find in [90].

$$C_{ap}^l = \frac{1}{N}\sum\nolimits_{i,j}\alpha\frac{1-SSIM(I_{ij}^l,\hat{I}_{ij}^l)}{2} + (1-\alpha)\left\|I_{ij}^l - \hat{I}_{ij}^l\right\| \tag{3.13}$$

The disparity smoothness loss is also added to require that the generated disparities of adjacent pixels must be continuous, because the distance around a target will not change sharply, this is consistent with the energy function defined in section 2. We sum all the gradients in *x* and *y* direction to compute the smoothness loss. As depth discontinuities often occur at image gradients, we weight this loss with an edge-aware term using the image gradients ∂I,

$$C_{ds}^l = \frac{1}{N}\sum\nolimits_{i,j}(\left|\partial_x d_{ij}^l\right|e^{-\left\|\partial_x I_{ij}^l\right\|} + \left|\partial_y d_{ij}^l\right|e^{-\left\|\partial_y I_{ij}^l\right\|}) \tag{3.14}$$

Here, d_{ij}^l is the disparity of pixel (*i*, *j*) on the left disparity map, ∂_x computes the gradient in *x* direction, $e^{-\left\|\partial_x I_{ij}^l\right\|}$ is the weight term to preserve edge. As the Fig. 3.13 shows, to produce more accurate disparity maps, we train our network to predict both the left and right image disparity maps simultaneously. It is easy to understand that the ground truths of left and right disparity maps are interchangeable. To measure the interchangeable ability of the generated disparity maps, we introduce the left-right disparity consistency penalty as part of the loss. This loss attempts to make the left-view disparity map be equal to the projected right-view disparity map, it is defined like the following:

$$C_{lr}^l = \frac{1}{N}\sum\nolimits_{i,j}\left|d_{ij}^l - d_{ij+d_{ij}^l}^r\right| \tag{3.15}$$

Here, $d_{ij+d_{ij}^l}^r$ is the disparity value on right disparity map at pixel $(i + d_{ij}^l, j)$, and this is the left consistence cost, we can also compute the right consistence cost by warping the left disparity map according to right map and calculate the absolute difference.

Until now, we have introduced all the modules necessary for self-supervised stereo matching. Need to note is that the matching result cannot compare with the supervised network, even if we add such a complicate loss function to control the whole learning process, because the opposite view image cannot provide a strong supervisor during the learning process. We think that it is a valuable try for some practical application that it is impossible to achieve ground truth for it, such as dense matching for sea surface images.

3.4.2 Alignment module for long distance sea surface image

According to the previous work, cross cost value is calculated at each possible disparity. However, for the long-distance sea surface images, the disparity varies in a wide range larger than 600 pixels, it is time and space consumption to calculate the cost value at each pixel. To solve this problem, we propose an alignment module that can shift the left and right feature maps before establishing the cost volume, it is inspired by the RoI Align module in Mask R-CNN. Fig.3.15 shows the structure after we have added this alignment module. We add this module between the feature extraction and cross cost volume building modules.

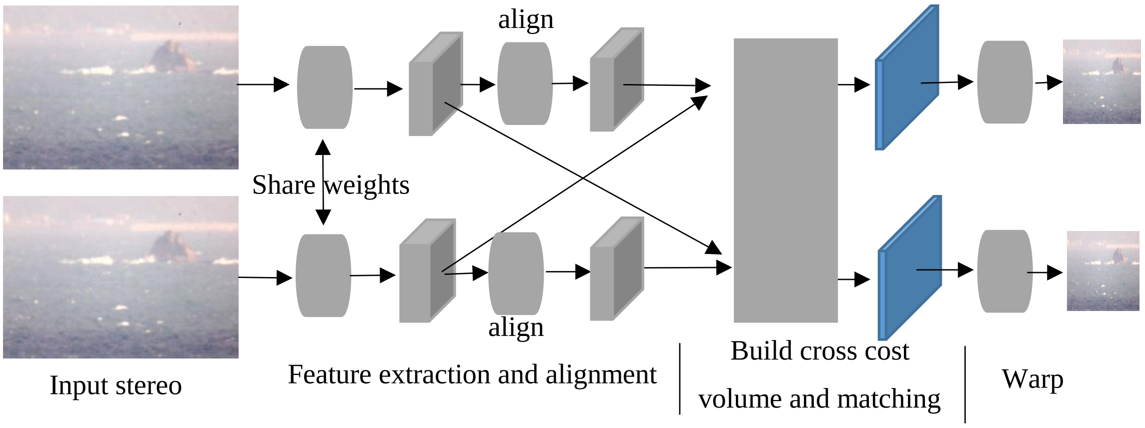


Fig.3.15 Align the feature maps before computing cost.

Observing the sea surface images, it is easy to find that the disparity between left and right image varies when y coordinate changes, and it obeys the relationship we have built in the front section. Now, we consider to use convolutional network to learning this relationship and complete the alignment. Common convolution operation uses a fixed kernel to traverse the entire input, which is no difference between the up and down parts of the image and it is not reasonable in this case. Thus, we add a transform operation before convolution layers. Fig.3.16 shows the transformation and the configuration of alignment module.

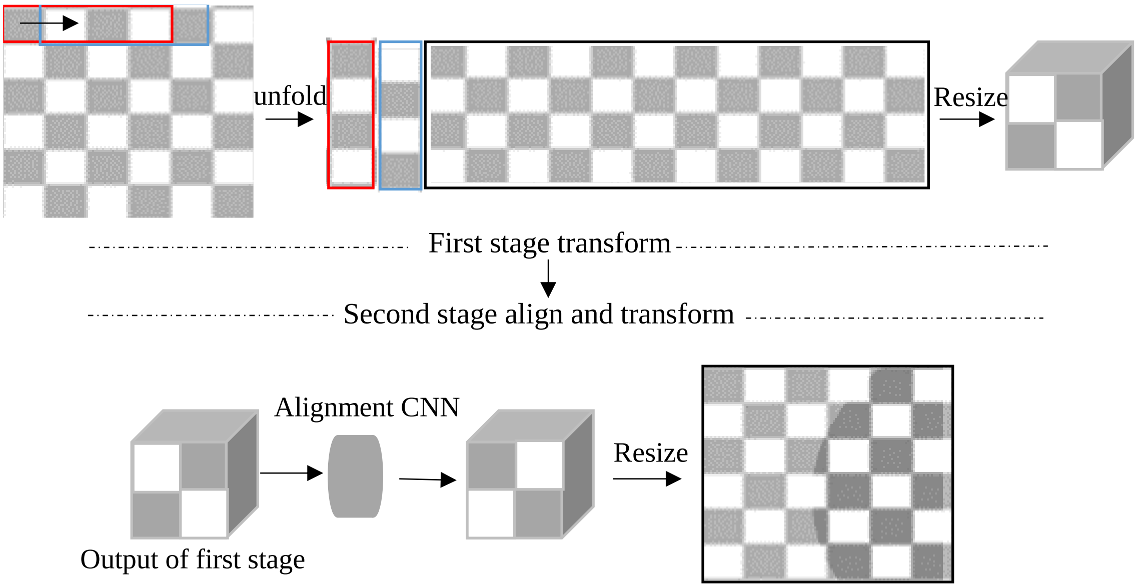
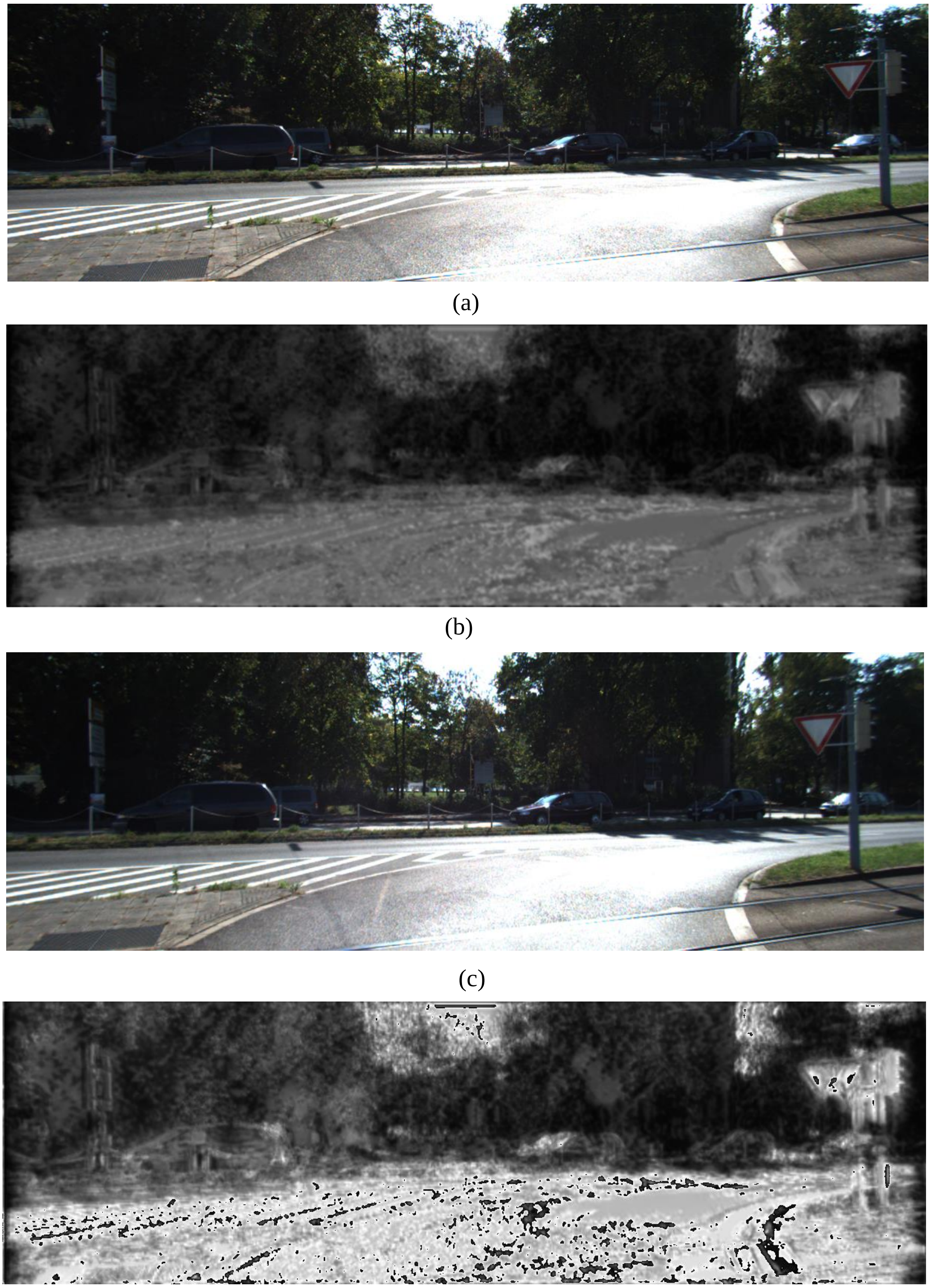


Fig.3.16 Alignment module workflow.



(a)

(b)

(c)

(d)

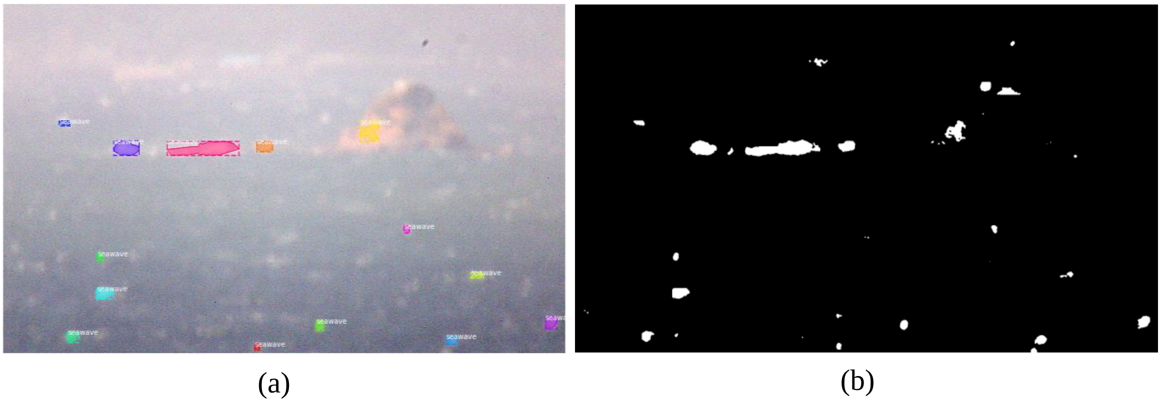
Fig.3.17 Disparity generation results, (a) and (c) are the left and right original images, (c) and (d) are their corresponding disparity maps.

Firstly, the feature map generated by the extraction module is unfolded along y coordinate like the Fig.3.16 shows, then resize the unfolded feature map and make the pixels on the same line to be in the same channel, feed this resized feature map to 2-layer convolutional modules and get the aligned feature map, resize it to the original size.

We have introduced the whole network established for our long-distance sea surface image stereo matching now. But due to the weak supervising strength, the stereo matching result is still not good. We think this network can also be used to other stereo matching network, as the training set of stereo image pairs of sea surface image has not been finished, we test our network on the other well-built data set, the result is not ideal. Fig.3.17 shows the disparity results of this network on the dataset of KITTI. From the results we find that the disparity map of the left image is continuous and smooth while the right disparity map has many black holes. The results are learned with Adam optimizer, learning rate is 0.002 and learning epoch number is 192. They are the best results we have achieved by this network. Many other learning parameters are also test, but the results are not ideal, the reason of hole occurrence and blur disparity map is still unknown now. In the future, we will try to adjust the network structure to improve the final results and finish the establish of sea surface image data set to apply this work to sea wave measurement.

3.5 Experiment results and discussion

In this section, we will give the experiment results of sea wave extraction and sparse matching by neural network. Firstly, we show the sea wave extraction results. Due to the large size of sea surface images and limited computational capability of our AI station, we set the training batch size to 8, fixed the parameter of convolutional layer in feature map generation module, only adjust the parameters of convolutional layer in three head modules. Adam optimizer is used to adjust the parameters. Fig.3.18 shows the comparison of sea wave extraction results by the network and traditional method. The left is the extraction result by network and the right is the extraction result of tradition method. We find that some part of the mountain is incorrectly extracted by both methods, however the network generates a much better result than tradition method in recognition the mountain as sea waves. Both two methods cannot recognize sea waves with low brightness.



(a) (b)

Fig.3.18 The comparison of sea wave extraction results, (a) is the extraction result of neural network, (b) is the extraction result of traditional method.

Fig.3.19 shows other four extraction results, we find that the problem of incorrectly recognizing mountain as sea waves is a common problem and the masks generated by the network is not precious in many times which maybe cause by the lack of training samples and the inaccurate of the ground truth, because the ground truth is made by traditional method, manual inspection only remove the incorrect matching but do not adjust the shape of mask. In the future, we will focus on enlarging the training set and making more precious mask to improve the final extraction result.

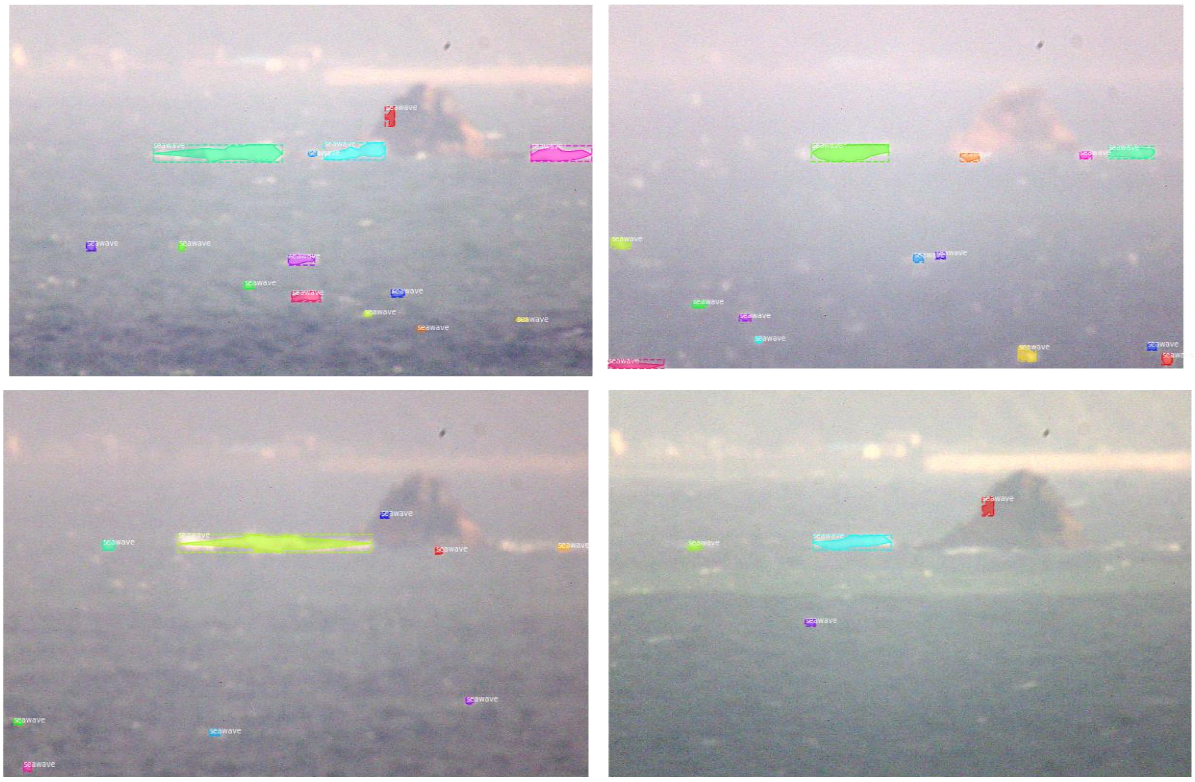
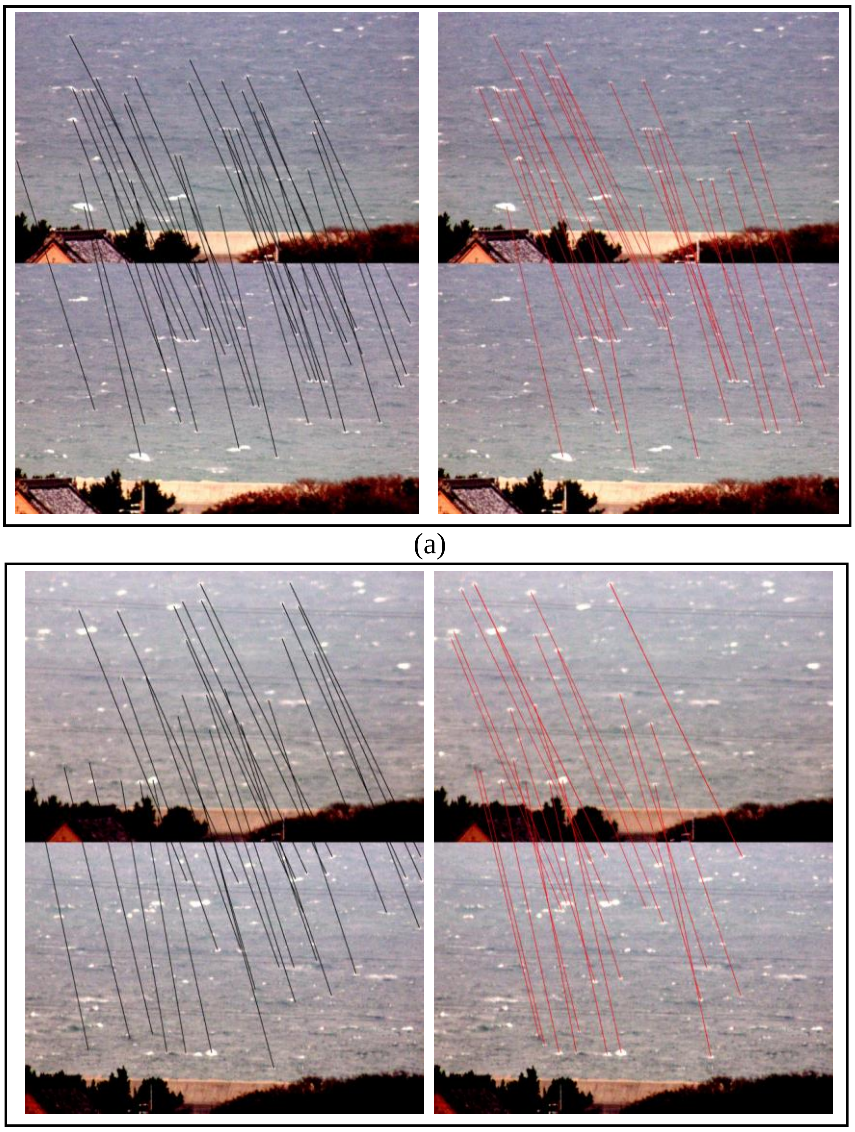


Fig.3.19 The sea wave extraction results.

Siamese network works on small blocks, it is not necessary to compute precious and recall of single sea surface image. We use the extraction network to cut the original image into small blocks which has one sea wave in the center position. We train the network for 10000 times. As the loss decreases, the similarity metric will be smaller for pairs of non-matching waves and larger for pairs of matching waves. In order to separate the two classifications, it is essential to find a threshold that can distinguish one from another. The inputs will be recognized as matching waves when their similarity metric is less than the threshold. Therefore, we make use of the test database to test the trained network with thresholds from 0.08 to 0.15. During the experiment, it will be a wrong case if the recognition of system varies from what we labeled. So, the test accuracy can be computed. Experimental results show that Siamese network reaches the highest 78% accuracy with the threshold of 0.11. Fig.3.20 shows the comparison of spares matching results of traditional method and Siamese network.



(a)

(b)

Fig.3.20 Matching results of traditional method (black) and CNN (red) method, (a) and (b) are two groups of comparison at different shooting time.

Aiming to carry out tsunami measurement, our laboratory proposes a method based on the binocular stereo vision. In the key step of sea wave matching, we compare traditional matching method with deep learning method to find out if we can improve the matching accuracy by one order magnitude through deep learning. Primary experiment result shows that with only with two convolution layers, two pooling layers and two fully connected layers, the network can reach 95% accuracy, 1.6% greater than the average accuracy of feature vector method and 5.8% greater than RANSAC+SURF method. In the future, we will focus on improving the final matching result of CNN method through adjusting the network parameter and training data.

Chapter 4 Experiments

In this section, we apply our method to sea surface images taken at different times with different illuminance conditions and shooting locations. We show two kinds of experiment results: sparse matching results and dense matching results. The sea surface images are captured by our tsunami measurement system in three periods: Feb 29-Mar 6, 2016; Mar 8-15, 2017; Aug 18-23, 2018. There were two sites, Fukuoka Kenritsu Suisan High School and Fukuoka Institute of Technology, with 3 monitoring distances: 14-20km, 4-10km and 8-14km. The experiments were performed on a desktop with a 3.4GHz Intel core CPU and 6GB of memory with C++ code.

4.1 Experiment configuration

The stereo system consists of two telephoto cameras to take sea surface images, two PTZ (Pan/Tilt/Zoom) heads that can be rotated in the pan and tilt directions to adjust the sight of each camera, one console panel to manually input control signal to PTZ heads, and two client computers to control the photography by adjusting the parameters of the cameras and receive the images captured by the cameras. There is also one total server to communicate with client computers, and control and send the necessary commands for photography to the client computers. Fig.4.1 is one of the deployed experiments of the proposed system. The two cameras were deployed 27m apart on the two ends of Fukuoka Institute of Technology's teaching building A. The measurement area was approximately 8km (closest point) to 14km (furthest point) away from the system at a height of approximately 30m above the sea level. (a) is the system's monitoring area, the red point depicts the position of system, and the area within the yellow lines is the cameras' field of view. (b) is the specific configuration of the proposed system. The top figure of (b) shows the components of our system, the middle figure demonstrates the actual deployment location of two telephoto cameras (the red points), and the bottom figure is the captured images of our system.

4.2 Sparse matching results

In order to evaluate the performance of the sparse matching method, we compare the matching results of the proposed method with the RANSAC+SURF method and Euclidean distance method. Precision, recall and runtimes are the three main evaluation terms, with precision and recall defined as the following, respectively:

$$precision = \frac{nTP}{(nTP+nFP)} \times 100 \tag{4.1}$$

$$recall = \frac{nTP}{nTP+nFN} \times 100 \tag{4.2}$$

where *nTP* and *nFP* are the numbers of correctly and wrongly detected correspondences in the matching method, respectively. *nFN* is the number of correct correspondences that are not detected.

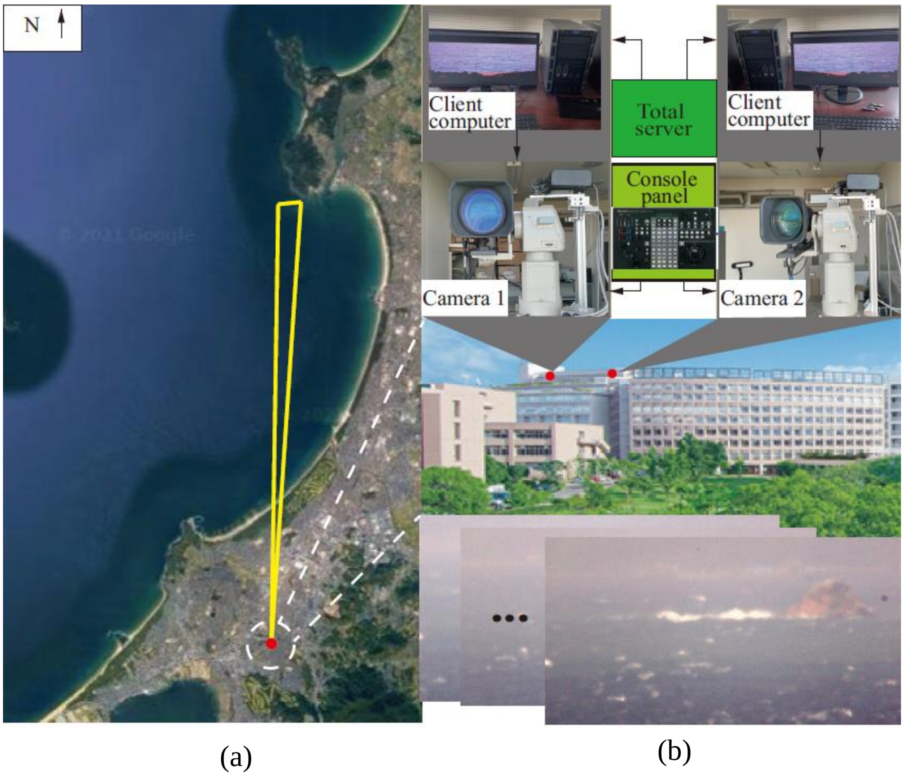


Fig.4.1 Experiment deployment of the proposed system. (a) the monitoring area, (b) actual components and deployment.

4.2.1 Sparse matching results of proposed traditional method

To illustrate the comparative result intuitively, we show 6 representative images' matching results taken at different times. In Fig.4.2, ①, ②, ③, ④, ⑤ and ⑥ are the 6 groups of comparison results of the representative image pairs. The (a) column is the RANSAC+SURF matching results, the (b) column is the Euclidean distance matching results and the (c) column is the results of our proposed method. ①, ②were taken Aug 18- 23, 2018 from Fukuoka Institute of Technology, with a monitoring distance of 8-14km. ③, ④ were taken at Mar 8-15, 2017 from Fukuoka Institute of Technology, the monitoring distance is 4-10km. ⑤ and ⑥ were taken Feb 29-Mar 6, 2016 from Fukuoka Kenritsu Suisan High School, with a monitoring distance of 14-20km.

From the matching results, we find that the RANSAC+SURF algorithm can generate stable and correct matching results only within the region where features are obvious and distinctive, such as mountains or large size sea waves. Our proposed method can match more than 90% of the sea waves correctly and stably. Among the three methods, our proposed method correctly matches the largest number of sea waves.

Table 4. Comparison of sparse matching results.

NO	RANSAC+SURF		Euclidean Method		Our Method	
	Precision	Recall	Precision	Recall	Precision	Recall
(a)	100.0	36.6	86.7	63.4	88.0	90.2
(b)	100.0	42.0	88.2	60.0	92.5	98.0
(c)	83.3	9.6	100.0	73.1	99.0	100.0
(d)	81.8	9.7	95.2	43.0	96.7	95.7
(e)	72.7	28.6	92.3	42.6	100.0	92.6
(f)	92.3	48.0	100.0	44.0	95.7	88.0
Average	88.4	29.1	93.7	54.4	95.3	94.1

We also conducted a quantitative comparison on different image pairs, the performance of each method is shown in Table 4. The average precision of the RANSAC+SURF algorithm was 88.4%, showing that sea waves can be correctly matched. However, the average recall was 29.1%, meaning that many sea waves are missed by the algorithm, and it will influence the final precision. The average precision of our proposed method is 95.3%; it is sufficient for the second step dense matching. The ground-truth is established by manual checking. From the comparison

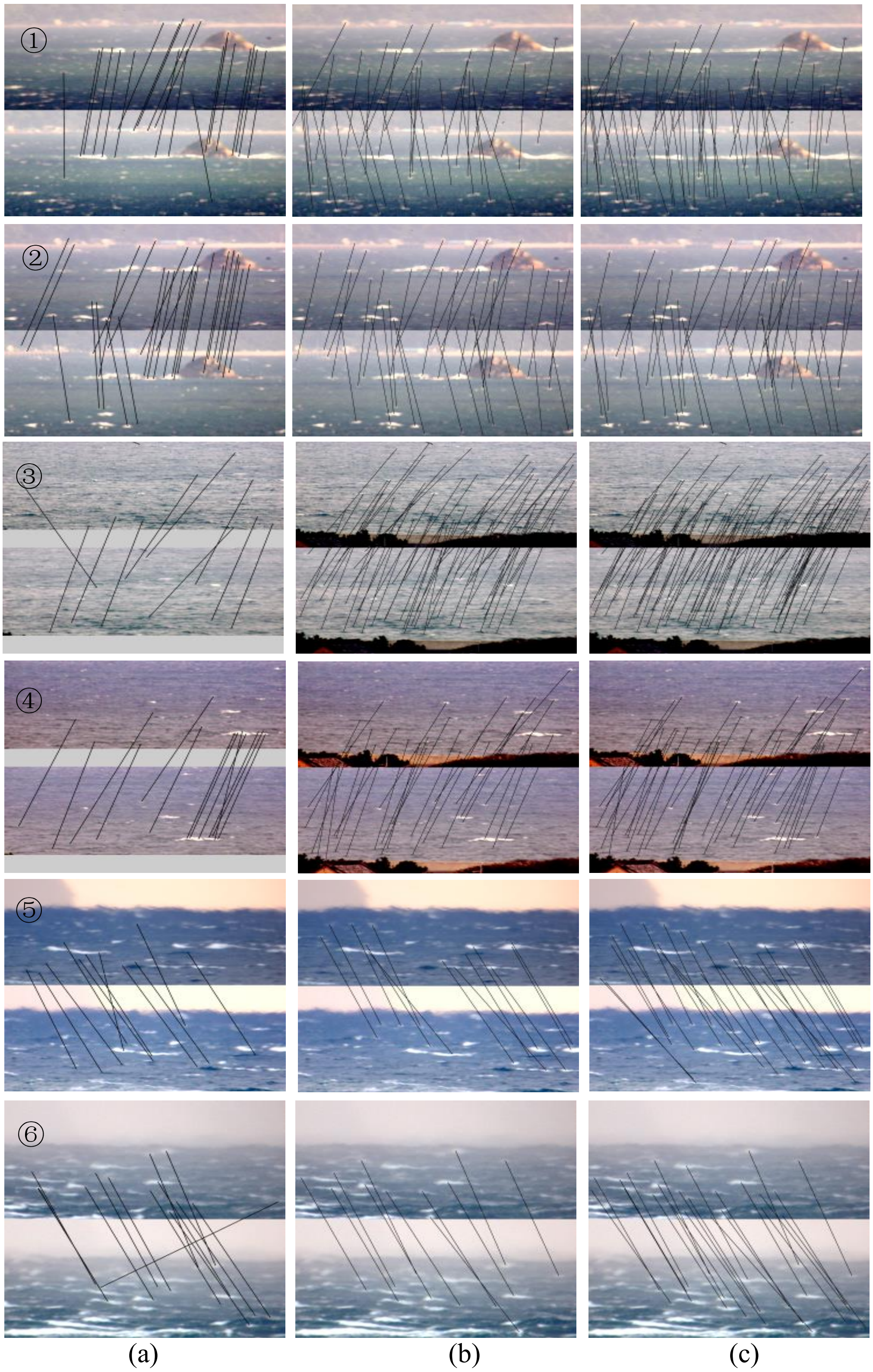


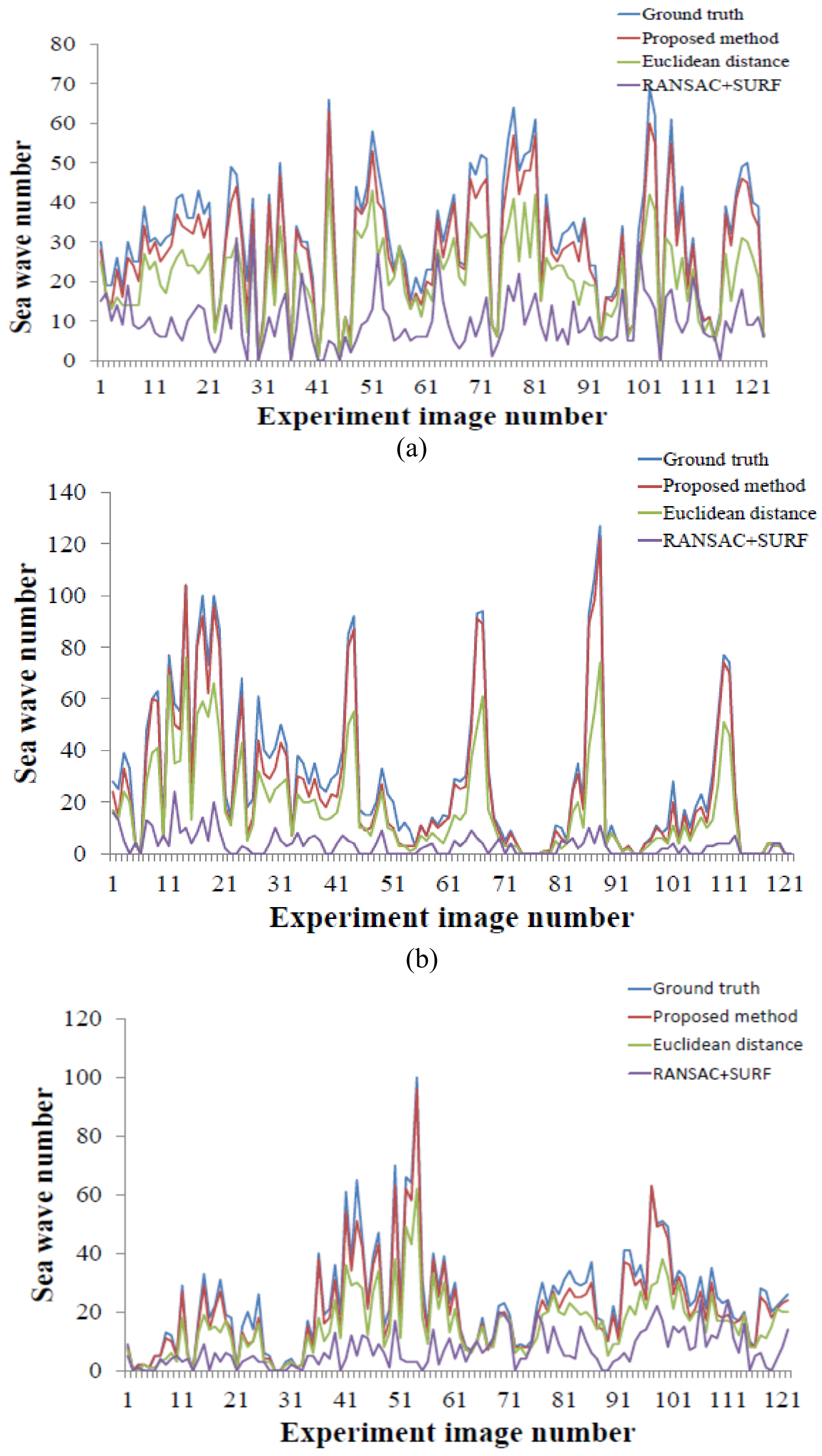
Fig.4.2 Comparison of RANSAC+SURF, Euclidean distance and the proposed method: Rows ① and② are the images taken at 18–23 August 2018 (8–14 km); Rows③ and ④ are the images taken at 8–15 March 2017 (4–10 km); Rows ⑤ and⑥ are the images taken at 29 February–6 March 2016 (14–20 km). Column (a) shows the results of RANSAC+SURF; Column (b) shows the results of Euclidean distance; and column (c) shows the results of the proposed method.

results, we can conclude that more than 90% of sea waves can be matched stably and correctly by feature vectors, regardless of different illumination conditions. It can solve the problem of no obvious feature points on the sea surface image during the sparse matching process.

The precision of Euclidean distance matching is 5.3% greater than the RANSAC+SURF method's. It means that the feature vector defined in this paper is effective. Meanwhile, the recall of our method is 39.7% greater than Euclidean distance method's. Thus, we can conclude that decision tree we built is more suitable for sea surface image matching. We also conducted a much more general comparison on sea surface image pairs with the RANSAC+SURF and Euclidean distance method. Fig.4.3 shows the comparison of the matching results of RANSAC+SURF, Euclidean distance and the proposed methods. The comparison is conducted on 367 pairs of sea surface images taken in 2016, 2017 and 2018.

The left column is the correctly matched sea wave numbers of the three matching methods. The horizontal axis is the image number ordering by image shooting time, the vertical axis is the correctly matched sea wave number, and each point (t, n) on the line represents that there are *n* correctly matched sea waves on the t^{th} sea surface image. The blue line represents the ground-truth sea wave number; it is counted by manual check. The purple line represents RANSAC+SURF matching results, the green line represents Euclidean distance matching results and the red line represents the number of correctly matched waves by our proposed method. We can find that most of the time, the red line is higher than the green and purple lines, meaning that our proposed method can match most sea waves at the most times. The right column is the matching precision of these three methods. The precision of our proposed method and the Euclidean distance matching is close in many cases, and in some cases, our proposed method is a little bit better than the Euclidean distance matching. Additionally, in many cases, our proposed method is better than RANSAC+SURF method.

To judge if two sea waves can be matched, we need to traverse from the root to the leaf of the decision tree. The maximum comparison time is the depth of the decision tree, six times, and the minimum comparison time is two times. There are nine comparison paths in the decision tree, the average comparison time is 4.25. It is nearly half the length of the feature vector. The Euclidean distance method and normalized cross correlation (NCC) used in SURF/SIFT need to traverse the whole feature vector. Thus, compared with these methods, the decision tree built in this paper can save half the time. Fig.4.4 shows the comparison results. The red line represents our proposed method's runtime, the green line is the runtime of the Euclidean distance matching and the purple line represents the runtime of the RANSAC+SURF (NCC is used to calculate similarity) method. We find the runtime of our proposed method and the Euclidean distance matching are very close. Observing the partial original running time data (see the small table in the upper right corner), we find our proposed method is 1–2 *ms* faster than Euclidean distance matching method in some cases. Our method is surprisingly more effective than the RANSAC+SURF algorithm. As RANSAC+SURF algorithm is point to point matching, the number of extracted feature points is much greater than our method's sea wave number. At the same time, the feature vector length is 128, much longer than our proposed feature vector length.

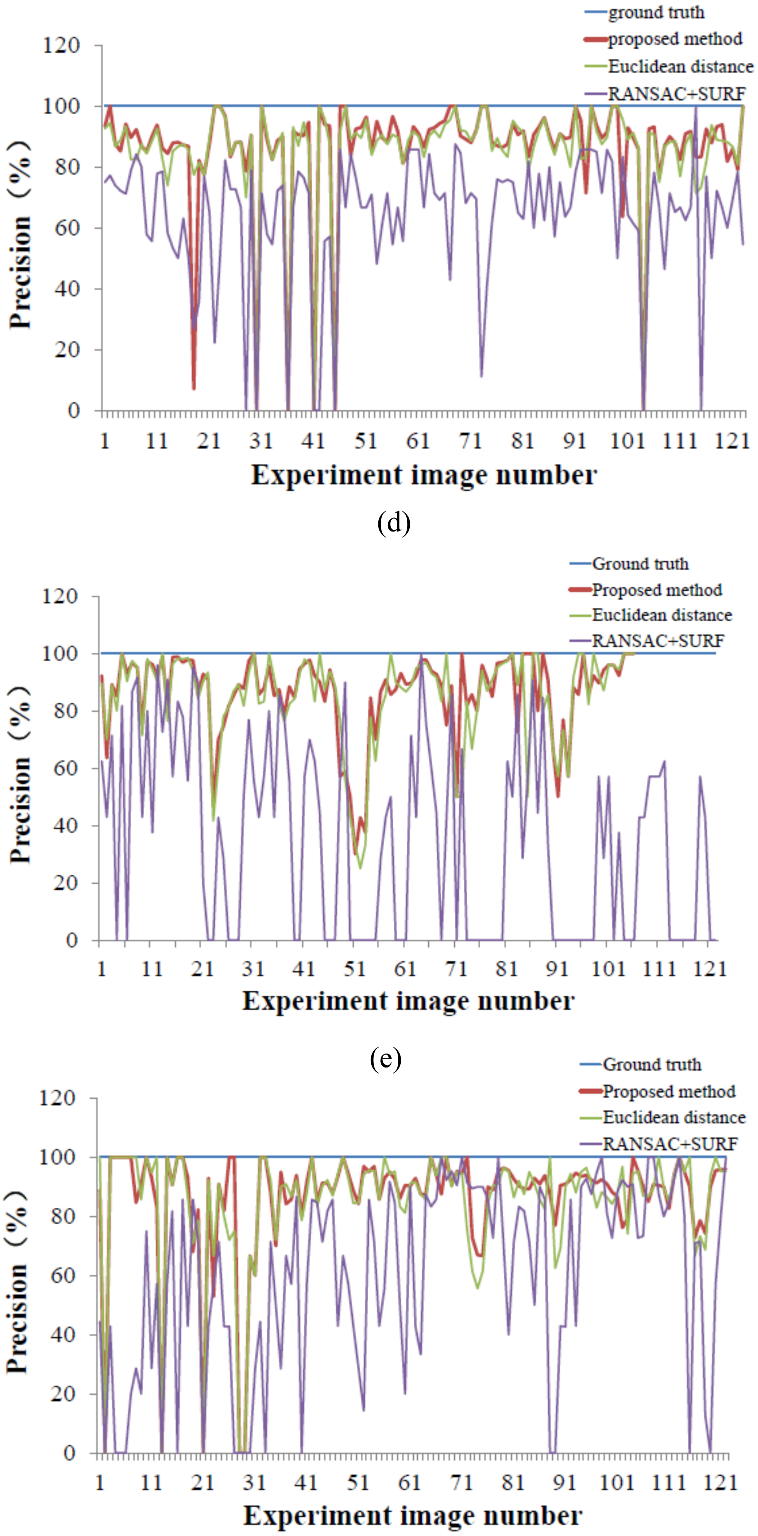


(a)

(b)

(c)

Fig.4.3 Comparison of three matching methods on correctly matched numbers and precision: (a,b) 29 February–6 March 2016 (14–20 km), (c,d) 8–15 March 2017 (4–10 km) and (e,f) 18–23 August 2018 (8–14 km).

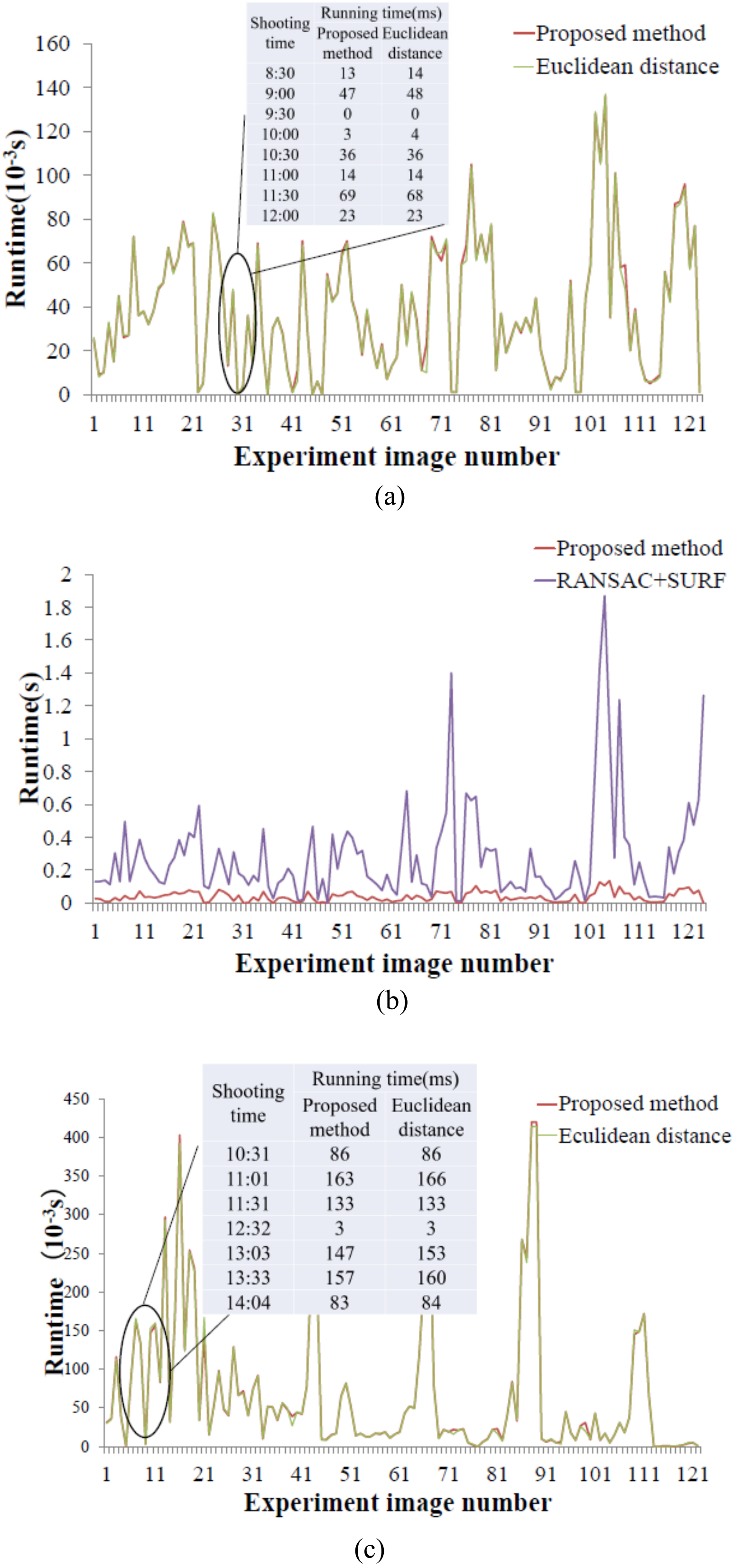


(d)

(e)

(f)

Fig.4.3 Comparison of three matching methods on correctly matched numbers and precision: (a,b) 29 February–6 March 2016 (14–20 km), (c,d) 8–15 March 2017 (4–10 km) and (e,f) 18–23 August 2018 (8–14 km).



Shooting time	Running time(ms)	
	Proposed method	Euclidean distance
8:30	13	14
9:00	47	48
9:30	0	0
10:00	3	4
10:30	36	36
11:00	14	14
11:30	69	68
12:00	23	23

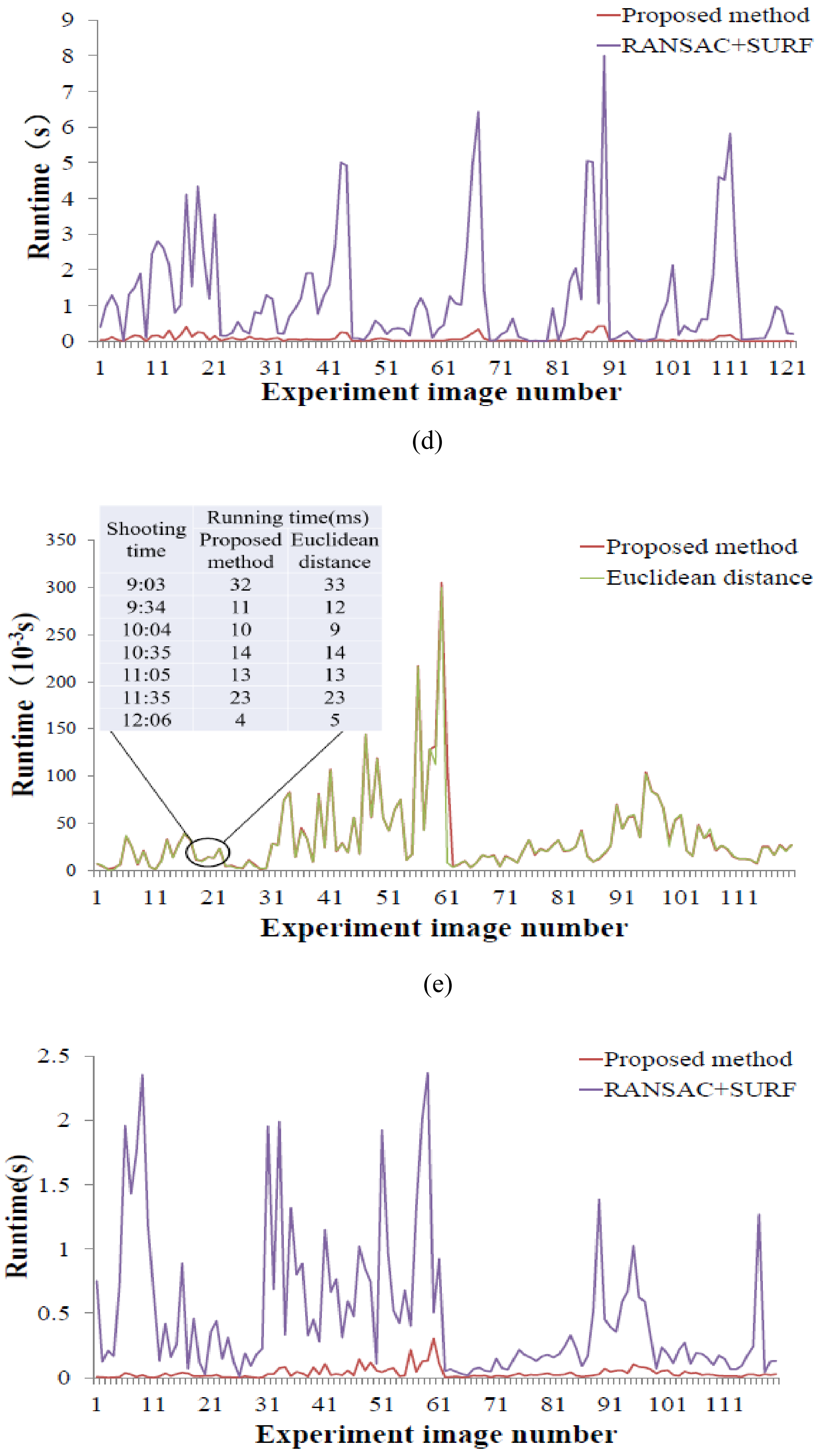
(a)

(b)

Shooting time	Running time(ms)	
	Proposed method	Euclidean distance
10:31	86	86
11:01	163	166
11:31	133	133
12:32	3	3
13:03	147	153
13:33	157	160
14:04	83	84

(c)

Fig.4.4 Comparison of three methods on runtime: (a,b) 29 February–6 March 2016 (14–20 km), (c,d) 8–15 March 2017 (4–10 km) and (e,f) 18–23 August 2018 (8–14 km).



(d)

Shooting time	Running time(ms)	
	Proposed method	Euclidean distance
9:03	32	33
9:34	11	12
10:04	10	9
10:35	14	14
11:05	13	13
11:35	23	23
12:06	4	5

(e)

(f)

Fig.4.4 Comparison of three methods on runtime: (a,b) 29 February–6 March 2016 (14–20 km), (c,d) 8–15 March 2017 (4–10 km) and (e,f) 18–23 August 2018 (8–14 km).

4.2.2 Dense matching results of proposed traditional method

We also conduct dense matching experiments on sea surface images taken at other times and locations. Fig.4.5 is the dense matching results of sea surface images, the (a) column is taken by the left camera, the (b) column is taken by the right camera and the (c) column is the disparity map. The irregular striped areas in the disparity map are non-overlapping areas in the left and right camera fields of view. ①and ② were taken at 18–23 August 2018 from Fukuoka Institute of Technology, with a monitoring distance of 8–14 km. ③, ④ were taken at 8–15 March, 2017 from the same site, with a monitoring distance of 4-10 km. ⑤ and ⑥ were taken at 29 February–6 March 2016 from Fukuoka Kenritsu Suisan High School, with a monitoring distance of 14–20 km. Due to the lack of ground truth, it is difficult for us to evaluate the dense matching accuracy and manual point by point inspection would be costly, thus, we can only perform a general accuracy check such as Fig.2.21. Limited by this paper's length, in this paragraph, we only make a general evaluation of the final results. According to the conclusion of Section 2.5.2, the disparity will increase in the area where there is a sea wave, coast or mountains, causing a whitish area on the disparity map. This phenomenon is consistent with our experimentally derived disparity maps.

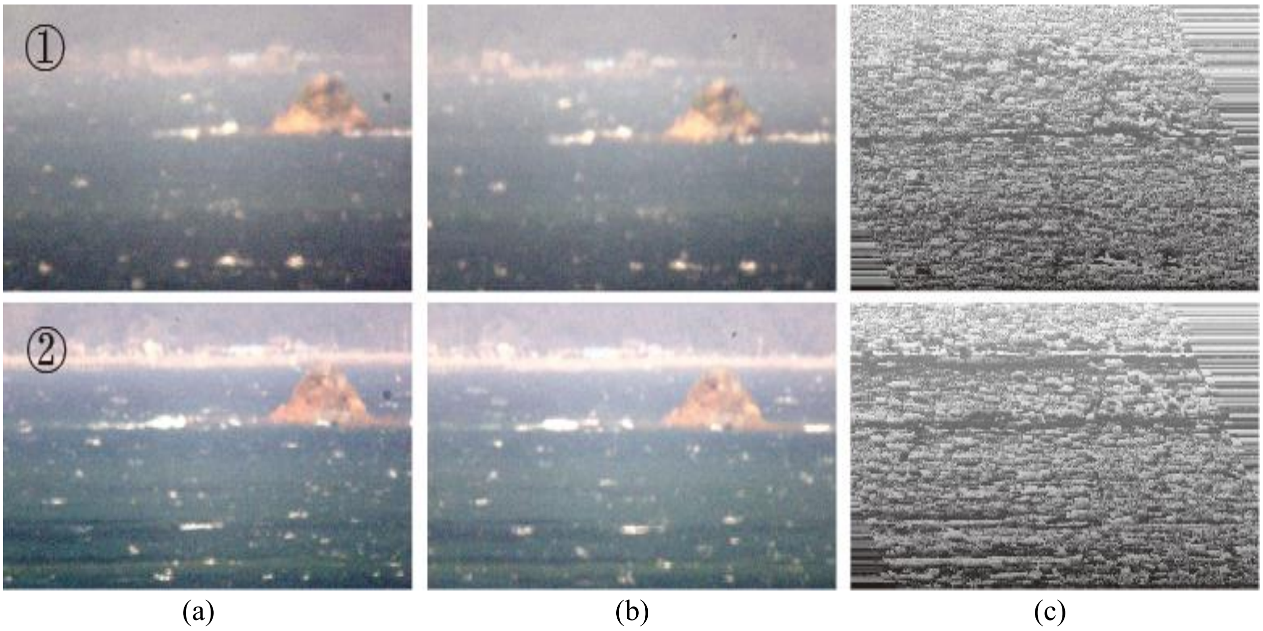
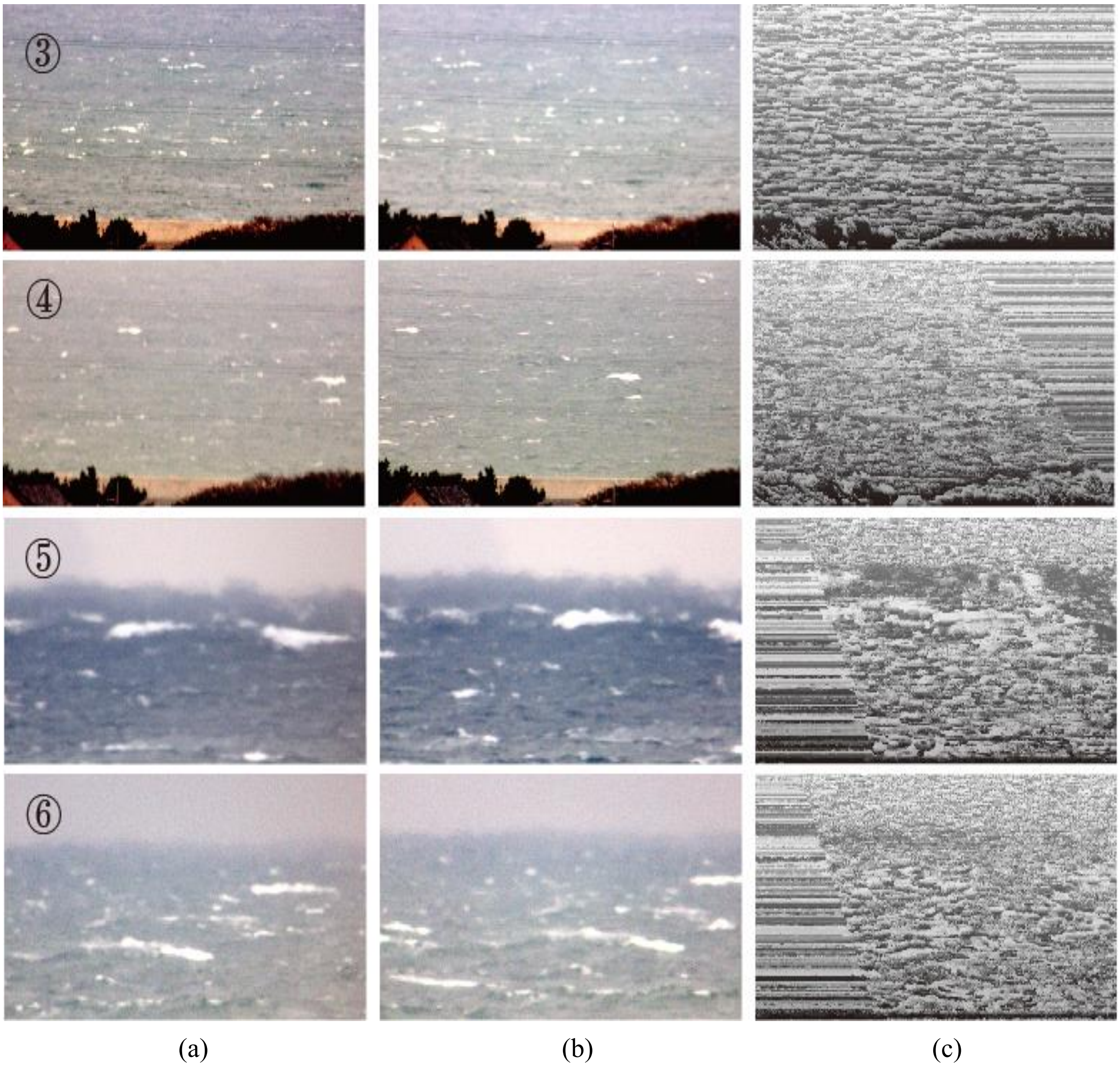


Fig.4.5 Disparity maps generated by dense matching: ①and ② are images taken at 18–23 August 2018 (8–14 km); Rows ③, ④are images taken at 8–15 March 2017 (4-10 km); and rows ⑤ and ⑥ are images taken at 29 February–6March 2016 (14–20 km). Column (a) shows the images taken by the left camera; Column (b) shows the images taken by the right camera; and column (c) shows the disparity maps.



(a) (b) (c)

Fig.4.5 Disparity maps generated by dense matching: ①and ② are images taken at 18–23 August 2018 (8–14 km); Rows ③, ④are images taken at 8–15 March 2017 (4-10 km); and rows ⑤ and ⑥ are images taken at 29 February–6March 2016 (14–20 km). Column (a) shows the images taken by the left camera; Column (b) shows the images taken by the right camera; and column (c) shows the disparity maps.

Chapter 5 Conclusion

We know that the wavelength of a tsunami wave is very long and usually the increase of sea surface height is small in the place where the tsunami occurs. According to its speed calculation equation $c = \sqrt{gh}$, where g is the acceleration of gravity and h is the water depth, the tsunami moves fast in deep sea areas and slows down near the coast. Thus, even a small increase in sea surface height of only 20 cm could be caused by a tsunami [62] and that has the potential to cause a large sea surface height increase near the coast. As the tsunami waves slow down near the coast, the wavelength becomes shorter while the wave energy is the same, resulting in a significant increase in sea surface height near the coast.

For different subsea topography, a tsunami 20 km away takes about 15–30 min to reach the coast. If our proposed stereo system can detect an abnormal sea surface rise of 20 cm and higher, we will have the opportunity to be able to provide coastal people with 15–20 min of escape time before the tsunami comes ashore. The proposed system has two long focal length cameras of 1140 mm, an acquisition rate of 30 fps, resolution of 1920 × 1080 pixels and two cameras are deployed 27 m apart from each other. The uncertainties in the calibration processing are controlled [63]. Thus, to realize real-time measurement for tsunami warning, the stereo matching error must be smaller than eight pixels and the running time of stereo matching must be shorter than 24^{-1} s. Furthermore, the stereo matching method must be capable of processing stereo images under different lighting conditions.

For the proposed method without deep learning, to verify the effectiveness on sea surface images under different lighting conditions, we conducted three groups of experiments, acquiring sea surface images from three different locations over three weeks, and performed sparse matching and dense matching on them. The experiment results show that our proposed method can correctly match greater than 90% of sea waves on the sea surface images, regardless of the distance from the surface, shooting angles and sea state conditions, it can meet the accuracy requirement of tsunami measurement. The running time of sparse matching ranges from 0–400ms when the correctly matched number ranges from 0 to 140 for stereo images under different conditions, for most stereo images, the running time is less than 40ms. Without considering the time consumption of dense matching, it can meet the requirement for real-time monitoring. We did not record the running time of dense matching, because it is much longer than sparse matching as we need to traverse the leaning cost volume for K (the searching path number) times (time complexity of $O(KWHDs)$). It is the major time consumption process, and causes our method to not be able to output 3D reconstruction of sea surface in real time. In the future, we will focus on increasing the speed of the dense matching by adding parallel operations and matrix operations. By manually checking one of the dense matching results, the dense matching accuracy is 87.0%, thus making it able to meet the accuracy requirement of sea surface reconstruction.

To resist the computation load caused by a large disparity range, we formulated the relationship between the disparity d and y coordinate. To reduce the computation load of dense matching, we constructed leaning cost volume based on this relationship. During this process, a dynamic penalty method is taken and penalty volume is calculated before we minimize the energy function. The final dense matching results validate our conclusions. In addition to time consumption, another limitation is the high requirement on image quality, the sparse matching can be conducted only in the case, where there are sea waves on the sea surface and the sea waves are captured by the stereo system. To address this insufficiency, we are considering fusing point-to-point sparse matching.

Differently from traditional sea surface stereo matching methods, we firstly perform the stereo matching on long distance sea surface images the monitoring distance ranges from 4 to 20km. Although the method here has many shortcomings, our expectation is that this attempt can open up new and exciting possibilities in terms of wave measurements for tsunami warnings.

We also study stereo matching with deep learning method to find out if we can improve the matching accuracy by one order magnitude through deep learning. For the proposed method with deep learning, we established three training sets to extract sea waves, realizing sparse and dense matching. A mask generation module based on CNN is built to extract sea waves from original images and an alignment module is built to reduce the space consumption of cost volume. However, due to the lack of training data, the stereo matching result is not well now. Thus, we did not conduct a large-scale comparison experiment of it. In the future, we will focus on improving the result of deep learning by increasing the training data and adjust the net structure.

Indeed, the tsunami problem is very serious and many elements can affect the system. For example, there are many elements affecting the sea surface elevations such as tidal waves, typhoon waves, tsunami waves, etc. Sea surface changes alone are not sufficient to indicate a tsunami. In our project, we believe that the wavelength of a tsunami wave is much longer than the wavelength of typhoon-induced waves, so when the sea surface height changes in a small area, we consider it to be caused by typhoons or sea breezes, etc. The change in sea surface height caused by tides has a certain time pattern, so when the sea surface height is abnormal in a wide area and the tidal factor is excluded, we will consider the occurrence of a tsunami. Furthermore, in our project, there are also many other research subjects such as 24h image capture for long-distance sea surface of 4–20km, image measurement in bad weather such as rain and snow, stereo mapping for binocular stereo vision, calculation of sea level height, method of determining the presence or absence of tsunami and how to estimate the arrival time. As we are limited by the paper length, in this paper, we only introduce the stereo matching algorithm. To achieve a high precision and fast stereo matching of proposed tsunami measurement system, we proposed a sea surface image matching method based on feature vectors and leaning cost volume, as well as two network structures to extract and match sea surface images.

Acknowledgements

I would first like to thank my tutor Prof. Lu of Graduate school of Intelligent Information System Engineering at Fukuoka Institute of Technology. Five years ago, he leaded me to the field of tsunami measurement by stereo system and has been my tutor since then. He helped me a lot in all aspects of life and studies during my studies in Japan, including but not limited to teaching me how to write a paper, how to do experiment, helping me revise my paper, and answering questions that arise in my research. He is also the supervisor of this thesis, without his help, this thesis could not have been successfully completed as now.

I would also thank the professors on the doctoral review committee, Prof. Song, Prof. Kogi, and Prof. Eguchiken. Their insightful comments and useful advice greatly helped me to refine my PhD thesis.

I would like to thank other students at the same laboratory for their kind help in my studies and life. Most of the time spent in the laboratory is enjoyable due to their kindness and help. All of the experiment data were obtained by their help, this research work could not be completed without their help.

Finally, I must give my faithful gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout these three and a half years and through the process of researching and writing my PhD thesis. This accomplishment would not have been possible without them. Thank you.

References

[1] US Navy 050102-N-9593M-040 A village near the coast of Sumatra lays in ruin after the Tsunami that struck South East Asia. https://commons.wikimedia.org/wiki/File:US_Navy_050102-N-9593M-040_South_East_Asia.jpg. (accessed on 19 Nov. 2021)

[2] NHK World-Japan. "The most powerful earthquake to hit Japan experienced by the survivor." https://www3.nhk.or.jp/nhkworld/en/ondemand/video/3016087/. (accessed on 19 Nov. 2021)

[3] Michael Warren, Roberto Candia. "Tsunami sweeps away entire towns on Chilean coast." https://www.sandiegouniontribune.com/sdut-tsunami-sweeps-away-entire-towns-on-chilean-coast-2010mar01-story.html. (accessed on 19 Nov. 2021)

[4] Pletcher, Kenneth and Rafferty, John P. "Japan earthquake and tsunami of 2011". Encyclopedia Britannica, 30 Nov. 2021, https://www.britannica.com/event/Japan-earthquake-and-tsunami-of-2011. (Accessed 25 January 2022).

[5] BBC News, "Indonesia quake toll jumps again." http://news.bbc.co.uk/2/hi/asia-pacific/4204385.stm. (accessed on 19 Nov. 2021).

[6] John P. Rafferty, Richard Pallardy. "Chile earthquake of 2010." https://www.britannica.com/event/Chile-earthquake-of-2010/. (accessed on 19 Nov. 2021)

[7] Meinig, C.; Stalin, S.E.; Nakamura, A.I.; González, F.; Milburn, H.B. Technology developments in real-time tsunami measuring, monitoring and forecasting. In Proceedings of OCEANS 2005 MTS/IEEE, Washington, DC, USA, 17–23 September 2005; pp. 1673–1679.

[8] Tatehata, H. The new tsunami warning system of the Japan Meteorological Agency. In Perspectives on Tsunami Hazard Reduction; Springer: Berlin/Heidelberg, Germany, 1997; pp. 175–188.

[9] 気象庁, 災害時自然現象報告書 2011 年第 1 号, 2011.

[10] Lauterjung, J.; Letz, H. 10 Years Indonesian Tsunami EarlyWarning System: Experiences, Lessons Learned and Outlook. 2017.

[11] Nayak, S.; Kumar, T.S. Indian tsunami warning system. Int. Arch. Photogramm Remote Sens. Spat. Inf. Sci. Beijing 2008, 37, 1501–1506.

[12] Allen, S.; Greenslade, D. Developing tsunami warnings from numerical model output. Nat. Hazards 2008, 46, 35–52.

[13] Larson, K.M.; Lay, T.; Yamazaki, Y.; Cheung, K.F.; Ye, L.; Williams, S.D.; Davis, J.L. Dynamic sea level variation from GNSS: 2020 Shumagin earthquake tsunami resonance

and Hurricane Laura. Geophys. Res. Lett. 2021, 48, e2020GL091378.

[14] Yu, K. Tsunami-wave parameter estimation using GNSS-based sea surface height measurement. IEEE Trans. Geosci. Remote Sens. 2014, 53, 2603–2611.

[15] Mulia, I.E.; Hirobe, T.; Inazu, D.; Endoh, T.; Niwa, Y.; Gusman, A.R.; Tatehata, H.; Waseda, T.; Hibiya, T. Advanced tsunami detection and forecasting by radar on unconventional airborne observing platforms. Sci. Rep. 2020, 10, 1–10.

[16] Haugen K, Lovholt F, Harbitz C. "Fundamental mechanisms for tsunami generation by submarine mass flows in idealised geometries". Marine and Petroleum Geology. 2005, 22 (1–2): 209–217.

[17] Margaritondo G. “Explaining the physics of tsunamis to undergraduate and non-physics students”. European Journal of Physics. 2005, 26 (3): 401.

[18] Voit S S. “Tsunamis”. Annual Review of Fluid Mechanics. 1987, 19 (1): 217–236.

[19] Japan Meteorological Agency, “地震・津波の観測監視体制” http://www.data.jma.go.jp/svd/eqev/data/monitor/index.html (accessed on 19 Nov. 2021).

[20] Shemdin, O.H.; Tran, H.M.; Wu, S. Directional measurement of short ocean waves with stereo photography. J. Geophys. Res.Ocean. 1988, 93, 13891–13901.

[21] Wanek, J.M.; Wu, C.H. Automated trinocular stereo imaging system for three-dimensional surface wave measurements. Ocean. Eng. 2006, 33, 723–747.

[22] Bechle, A.J.;Wu, C.H. Virtual wave gauges based upon stereo imaging for measuring surface wave characteristics. Coast. Eng. 2011, 58, 305–316.

[23] Kosnik, M.V.; Dulov, V.A. Extraction of short wind wave spectra from stereo images of the sea surface. Meas. Sci. Technol. 2010, 22, 015504.

[24] Marom, M., Goldstein, R.M., Thornton, E.B., Shemer, L, 1990. Remote sensing of ocean wave spectra by interferometric synthetic aperture radar. Nature 345, 793-795.

[25] Dankert, H., Horstmann, J., Lehner, S., Rosenthal, W.G., 2003. Detection of wave groups in SAR images and radar image sequences. IEEE Transactionson Geoscience and Remote Sensing 41,1437-1446.

[26] Benetazzo, A., 2006. Measurements of short water waves using stereo matched image sequences. Coastal Engineering 53,1013-1032.

[27] Fedele, F., Benetazzo, A., Forristall, G.Z., 2011. Space-time waves and spectra in the Northern Adriatic Sea via a Wave Acquisition System. Proceedings of the ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering (OMAE 2011) in Rotterdam, The Netherlands.

[28] Fedele, F., Gallego, G., Yezzi, A., Benetazzo, A., Cavaleri, L, Sclavo, M., Bastianini, M., 2012. Euler characteristics of oceanic sea states. Mathematics and Computers in Simulation 82 (6), 1102-1111.

[29] Gallego, G., Yezzi, A., Fedele, F., Benetazzo, A., 2011. A variational stereo method for the 3-D reconstruction of ocean waves. IEEE Transactions of Geosciences and Remote Sensing 49 (11), 4445-4457.

[30] Benetazzo, A. Measurements of short water waves using stereo matched image sequences. Coast. Eng. 2006, 53, 1013–1032.

[31] Wanek, J.M.; Wu, C.H. Automated trinocular stereo imaging system for three-dimensional surface wave measurements. Ocean. Eng. 2006, 33, 723–747.

[32] Benetazzo, A. Measurements of short water waves using stereo matched image sequences. Coast. Eng. 2006, 53, 1013–1032.

[33] Brandt, A.; Mann, J.; Rennie, S.; Herzog, A.; Criss, T. Three-dimensional imaging of the high sea-state wave field encompassing ship slamming events. J. Atmos. Ocean. Technol. 2010, 27, 737–752.

[34] Gallego, G.; Yezzi, A.; Fedele, F.; Benetazzo, A. A variational stereo method for the three-dimensional reconstruction of ocean waves. IEEE Trans. Geosci. Remote Sens. 2011, 49, 4445–4457.

[35] Gallego, G.; Yezzi, A.; Fedele, F.; Benetazzo, A. Variational stereo imaging of oceanic waves with statistical constraints. IEEE Trans. Image Process. 2013, 22, 4211–4223.

[36] Bergamasco, F.; Torsello, A.; Sclavo, M.; Barbariol, F.; Benetazzo, A. WASS: An open-source pipeline for 3D stereo reconstruction of ocean waves. Comput. Geosci. 2017, 107, 28–36.

[37] Vieira, M.; Guimarães, P.V.; Violante-Carvalho, N.; Benetazzo, A.; Bergamasco, F.; Pereira, H. A Low-Cost Stereo Video System for Measuring Directional WindWaves. J. Mar. Sci. Eng. 2020, 8, 831.

[38] Marengoni, M.; Stringhini, D. High level computer vision using opencv. In Proceedings of the 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials, Alagoas, Brazil, 28–30 August 2011; pp. 11–24.

[39] Cheng, X.; Wang, P.; Yang, R. Depth estimation via affinity learned with convolutional spatial propagation network. InProceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 103–119.

[40] Zhong, Y.; Dai, Y.; Li, H. Self-supervised learning for stereo matching with self-improving ability. arXiv 2017, arXiv:1709.00930.

[41] Ren, H.; Raj, A.; El-Khamy, M.; Lee, J. SUW-Learn: Joint Supervised, Unsupervised,

Weakly Supervised Deep Learning for Monocular Depth Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle,WA, USA, 14–19 June 2020; pp. 750–751.

[42] Chen, C.h.; Lu, C.w.; ying, Y. Method of SeaWave Extraction and Matching from Images Based on Convolutional Neural Network. In Proceedings of the 5th International Conference on Engineering, Applied Sciences and Technology, Luang Prabang, Laos, 2–5 July 2019; pp. 1–4.

[43] Harris, C.G.; Stephens, M.; others. A combined corner and edge detector. In Proceedings of the Alvey vision conference, Manchester, UK, 31 August–2 September 1988; Volume 15, p. 10-5244.

[44] Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.

[45] Marr, D.; Hildreth, E. Theory of edge detection. Proc. R. Soc. Lond. Ser. B Biol. Sci. 1980, 207, 187–217.

[46] Lowe, D.G. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 2004, 60, 91–110.

[47] Yang, Y.; Lu, C. Long-distance sea wave extraction method based on improved Otsu algorithm. Artif. Life Robot. 2019, 24, 304–311.

[48] Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 1615–1630.

[49] Tola, E.; Lepetit, V.; Fua, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. 2009, 32, 815–830.

[50] Wang, Z.; Fan, B.;Wu, F. Local intensity order pattern for feature description. In Proceedings of the 2011 International Conference on Computer Vision, Colorado Springs, CO, USA, 20–25 June 2011; pp. 603–610.

[51] Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.

[52] Kim, J.; others. Visual correspondence using energy minimization and mutual information. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 1033–1040.

[53] Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth international conference on computer vision (IEEE Cat. No. 98CH36271), Bombay, India, 7 January 1998; pp. 839–846.

[54] Hirschmuller, H. Stereo processing by semiglobal matching and mutual information. IEEE

Trans. Pattern Anal. Mach. Intell. 2007, 30, 328–341.

[55] Guofu Yin, Image segmentation based on threshold [D]. Xi'an: Xi'an Electronic and Science University, 2007.

[56] Nobuyuki Otsu. "A threshold selection method from gray level histograms". IEEE Trans. Sys., Man., Cyber. 1979, 9 (1): 62–66.

[57] Safavian, S.R., Landgrebe, D.: 'A survey of decision tree classifier methodology', IEEE Trans. Syst. Man Cybern., 1991, 21, pp. 660–674.

[58] Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[59] Hao Yi, Kazuhiro Tsujino and Cunwei Lu, 3-D image measurement of the sea for disaster prevention, Artificial Life and Robotics, DOI: 10.1007/s10015-018-0427-0, ISSN: 1433-5298 (Print) 1614-7456 (Online), Vol. 22, Issues 72, pp.1-7, Feb 2018.

[60] Ledvij, M. Curve fitting made easy. Ind. Phys. 2003, 9, 24–27.

[61] Zhang Y, Chen Y, Bai X, et al. Adaptive unimodal cost volume filtering for deep stereo matching[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 12926-12934.

[62] Office, C. Information about Tsunami. Available online: http://www.bousai.go.jp/kohou/kouhoubousai/h22/05/special_01.html/ (accessed on 16 September 2021).

[63] Zabih, R.; Woodfill, J. Non-parametric local transforms for computing visual correspondence. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 2–4 March 1994; Springer: Berlin/Heidelberg, Germany, 1994; pp. 151–158.

[64] Rahman M A, Ahmed B, Hossian M A, et al. An adaptive background modeling based on modified running Gaussian average method[C]//2017 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2017: 524-527.

[65] Campbell R A, Lasky E Z. Adaptive threshold procedures: BUDTIF[J]. The Journal of the Acoustical Society of America, 1968, 44(2): 537-541.

[66] Dalal N, Triggs B. Object detection using histograms of oriented gradients[C]//Pascal VOC Workshop, ECCV. 2006.

[67] Abu Hassan, Mohd Fauzi & Pri, Azurahisham & Ahmad, Zakiah & Azahar, Tengku. (2020). Scale adaptive region covariance descriptor for visual tracking. IOP Conference Series: Materials Science and Engineering. 932. 012090. 10.1088/1757-899X/932/1/012090.

[68] B. Zhou, X. Duan, W. Wei, D. Ye, M. Woźniak and R. Damaševičius, "An Adaptive Local Descriptor Embedding Zernike Moments for Image Matching," in IEEE Access, vol. 7, pp. 183971-183984, 2019, doi: 10.1109/ACCESS.2019.2960203.

[69] Hartmann, J., Klussendorff, J. H. and Maehle, E., 2013. A comparison of feature descriptors

for visual SLAM. In: Proceedings of IEEE European Conference on Mobile Robots, Barcelona, pp. 56-61.

[70] Wöhler, C., 2013. 3D Computer Vision: Efficient Methods and Applications (2nd ed.). Springer Science & Business Media, pp. 55-73.

[71] Wendel, A., Maurer, M., Graber, G., Pock, T. and Bischof, H., 2012. Dense reconstruction on-the-fly. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, pp. 1450-1457.

[72] Kekec, T., Yildirim, A. and Unel, M., 2014. A new approach to real-time mosaicing of aerial images. Robot Auton Syst, 62(12), pp. 1755-1767.

[73] Briechle K, Hanebeck U D. Template matching using fast normalized cross correlation[C]//Optical Pattern Recognition XII. International Society for Optics and Photonics, 2001, 4387: 95-102.

[74] Yoo J C, Han T H. Fast normalized cross-correlation[J]. Circuits, systems and signal processing, 2009, 28(6): 819-843.

[75] H. Hirschmu¨ ller, F. Scholten, and G. Hirzinger, "Stereo Vision Based Reconstruction of Huge Urban Areas from an Airborne Pushbroom Camera (HRSC)," Proc. 27th Symp. German Assoc. for Pattern Recognition, pp. 58-66, Aug./Sept. 2005.

[76] S. Gidaris, N. Komodakis, Detect, replace, re_ne: Deep structured prediction for pixel wise labeling, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5248.

[77] Zbontar J, LeCun Y. Computing the stereo matching cost with a convolutional neural network[C] Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1592-1599.

[78] Z. Y. Chen, X. Sun, L. Wang, Y. A. Yu, and C. Huang, "A deep visual correspondence embedding model for stereo matching costs," in Proceedings of the International Conference on Computer Vision, pp. 972–980, IEEE, Santiago, Chile, December 2015.

[79] W. Luo, A. G. Schwing, and R. Urtasun, "Efficient deep learning for stereo matching," in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5695–5703, Las Vegas, NV, USA, June 2016.

[80] A. Shaked, L. Wolf, and Ieee, "Improved stereo matching with constant highway networks and reflective confidence learning,"in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6901–6910, IEEE, Honolulu, HI, USA, July 2017.

[81] A. Dosovitskiy, P. Fischer, E. Ilg et al., "FlowNet: learning optical flow with convolutional networks," in Proceedings of the 2015 IEEE International Conference on Computer Vision

(ICCV), pp. 2758–2766, IEEE, Santiago, Chile, December.

[82] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical ow, and scene ow estimation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4040-4048.

[83] S. Gidaris and N. Komodakis, "Detect, replace, refine: deep structured prediction for pixel wise labeling," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7187–7196, IEEE, Honolulu, HI, USA, July 2017.

[84] Kim, Sunok, et al. "Feature augmentation for learning confidence measure in stereo matching." IEEE Transactions on Image Processing 26.12 (2017): 6019-6033.

[85] R. Garg, B. G. VijayKumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: geometry to the rescue," in Computer Vision-Eccv 2016 Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9912, pp. 740–756, Springer, Berlin, Germany, 2016.

[86] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6602–6611, IEEE, Honolulu, HI, USA, July 2017.

[87] N. Smolyanskiy, A. Kamenev, and S. Birchfield, "On the importance of stereo for accurate depth estimation: an efficient semi-supervised deep neural network approach," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1120–1128, IEEE, Salt Lake City, UT, USA, June 2018.

[88] Y. Luo, J. Ren, M. Lin et al., "Single view stereo matching," in Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 2018.

[89] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[90] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. Transactions on Image Processing, 2004. 4.

Publication list

[1] Y. Ying, L. Cunwei, “A Stereo Matching Method for 3D Image Measurement of Long-Distance Sea Surface”, Journal of Marine Science and Engineering (JMSE), 1-21, 1281, Volume 9, Issue 11, (2021)

[2] Y. Ying, C. Chenhao, L. Cunwei, “Method of Sea Wave Matching Based on Convolutional Neural Network -- A comparison with feature vector matching method”, 25th International Symposium on Artificial Life and Robotics, GS3-5, Beppu, Japan, Jan. 2020.

[3] 陳 志鵬, 楊 英, 盧 存偉, “ブロック分割二値化手法に基づく遠距離撮影画像の波抽出”, 2020年度電気・情報関係学会九州支部連合大会, 04-2A-09, Sept. 2020.

[4] Y. Ying, L. Cunwei, “Long-distance sea wave extraction method based on improved Otsu algorithm”, Artificial Life Robotics 304–311, Volume 24, Issue 3, (2019).

[5] Y. Ying, L. Cunwei, C. Chenhao, “Long distance sea surface images fast sparse matching by integrated feature vector”, Proc. of the Seventh Asia International Symposium on Mechatronics, E21, vol 589, Springer, Singapore, 2019.

[6] Y. Ying, L. Cunwei, C. Chenhao, “Research on stereo matching methods for long distance sea surface image”, 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), ID 125, pp.101-104, Luang Prabang, Lao PDR, 2019.

[7] C. Chenhao, L. Cunwei, Y. Ying, “Method of Sea Wave Matching Based on Siamese Network”, Proc. of the Seventh Asia International Symposium on Mechatronics, E25, Vol 589, Springer, Singapore, 2019.

[8] C. Chenhao, L. Cunwei, Y. Ying, “Method of Sea Wave Extraction and Matching from Images Based on Convolutional Neural Network”, 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), ID 131, pp.105-108, Luang Prabang, Lao PDR, 2019.

[9] Y. Ying, L. Cunwei, “Long distance sea wave extraction method based on improved Otsu algorithm”, 23rd International Symposium on Artificial Life and Robotics, OS7-4. Beppu, Japan, Jan. 2018.

[10] Y. Ying, L. Cunwei, “Long distance sea wave extraction method by improved block Otsu algorithm”, The 70th Joint Conference of Electrical, Electronics and information Engineers in Kyushu, 13-1A-05, Okinawa, Japan, Sept. 2017.

[11] Y. Ying, L. Cunwei, Y. Lei, T. Kazuhiro. “Improved Mean Shift Algorithm for Long Distance Sea Wave Tracking”, Proceedings of the IEICE General Conference 2017, D-11-41, Nagoya, Japan, Mar. 2017.