TOWARDS AN AUTOMATIC ICE NAVIGATION SUPPORT SYSTEM IN THE ARCTIC AREA

by

Shahram Sattar
Bachelor of Science, Isfahan University, Iran, 2010

A thesis
presented to Ryerson University

in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Civil Engineering

Toronto, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Towards an Automatic Ice Navigation Support System in the Arctic Area

Master of Applied Science 2014
Shahram Sattar
Civil Engineering
Ryerson University

Abstract

Conventional ice navigation through sea ice is manually operated by well-trained navigators, whose experiences are heavily relied upon to guarantee the ship's safety. Despite increasingly available ice data and information, little has been done to develop automatic ice navigation systems to better guide ships in sea ice. In this study firstly navigable sea areas for different types of ships were identified according to the navigation codes in northern regions. Secondly, three algorithms of path planning were adopted to automatically compute the safest-and-shortest ship routes based on the concepts of the Voronoi diagram, Visibility graph, and Visibility-Voronoi diagram, respectively. These algorithms and results were compared and evaluated in terms of different application scenarios. Results show that the Visibility-Voronoi approach seems to be the best viable solution in terms of computing performance and navigation safety. The work will provide a basis for further development towards an automatic ice navigation support system.

Acknowledgements

Completing this thesis has been great milestone I would never have achieved without the support and encouragement of all the individuals mentioned in this section. I have started to note everything that I would like to put here since I started my study at Ryerson University. I would like to express my deepest appreciation to all those who made it possible for me to complete my studies at Ryerson University.

The idea for this research started about two year ago when I was studying as a MEng student and decided to switch to the MASc program. My supervisor, Prof. Songnian Li, discussed with me one of the potential fields of research he had. Prof. Songnian Li is an extremely nice and gentle person to have as a mentor. During my MEng studies, his encouragement and support led me to be interested in this topic and to transfer to the MASc program after one year. I would like to thank him for encouraging my research and for allowing me to grow as a research student. I deeply appreciate his financial and spiritual support during my Master's studies, since he is a person of excellent character. Furthermore, a special thank to Dr. Xintao liu, without his kindness and constant help this research would not have been possible.

I have been very fortunate to have Prof. Ahmed El-Rabbany as the graduate program director during my Master's studies at Ryerson. I am greatly indebted to Prof. Mike Chapman, whose lectures opened my eyes to see GIS in a different way. I am thankful to Mr. Christopher Daniel and Dr. Peter Kedron from the Spatial Analysis program for their teaching and guidance. Thanks also to the former and current graduate program administrators, Ms. Rachel Trozzolo and Ms. Rachel Harpley.

A special thanks to my family. Words cannot express how grateful I am to my father, mother, sister and brother. Without their moral support and motivation, I would not have been able to focus on my studies. I especially thank my father who encouraged me to become a Geomatics Engineer, which is not something I had previously considered. However, when I got into it, it stimulated my desire of study to become an expert in this field. Also, I am grateful to my uncle and his wife who provided lots of gatherings, lunches and dinners here.

I am very grateful for having to have a few close friends who really care about me. I am so glad to have Hossein Reshadi Nejad and Wei Huang, who have been my good friends since I started my Master's studies. Hossein came from my hometown, and we have many things in common to share with each other. When I met Hossein for the first time on campus, I was so glad to have such a good brother in Canada. Wei is a PhD student here and we spend most of our research time in the same lab. He always injects energy into the work. Some difficult decisions during my graduate study could not be made without his help. I also thank Dr. Wai Yeung Yan for his kind guidance and comments during my research. He is a person who has treated me like a friendly brother and I learned many valuable things from him. I also want to thank him for suggesting the LaTeX software to me, which is an outstanding open source document markup language for writing a professional thesis. I am glad to have known a diverse range of graduate students from different parts of the world, in particular Fredy Paya, Akram Afifi,Ida Ensafi, Hossein Matinmensesh, Kamil Fasial and Zheng Niu.

Finally, this work has been partially supported by the Beaufort Regional Environmental Assessment (BREA) project, entitled “CanICE” - a sea ice information database and web-based portal with novel, interactive knowledge discovery tools. The Beaufort Regional Environmental Assessment program is funded by Aboriginal Affairs and Northern Development Canada.

Shahram Sattar
June 5th, 2014

Dedication

To my lovely father & mother

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Motivation

Variation in sea ice area is an important factor in designing Arctic maritime transportation systems. Although the principal attraction of Arctic navigational options lies in the tremendous distance savings, maritime transportation in the Arctic area must be both safe and feasible. While traditional ship navigation has been based on the skill of captain and crew, it is not expected that all transiting vessels' crew have the same level of knowledge to guide ships in that area. Providing the required skills for mariners in that area is an important factor; therefore, other supports and services are necessary for safe and economical navigation. In the Arctic area, ice is the main concern for navigators because it not only restricts ships' movements but also sometimes forces them to change their route and speed (INCW, 2012).

The North West Passage (NWP), discovered in 1850s, is located in the middle of the Canadian Arctic, which connects the Atlantic and Pacific Oceans (Howell *et al.*, 2004). It provides a route between Europe and Asia that is 9000km shorter than the Panama Canal route and 1700 km shorter than traveling around Cape Horn in South America (Howell *et al.*, 2004). Although it has been relatively unusable due to the permanent sea ice, satellite observations indicate that the area of sea ice has decreased considerably over the last few years (Khon *et al.*, 2010). As a result, these changes provide greater access to marine navigation in the Arctic Ocean and higher potential for Arctic transportation systems. According to the Transportation Safety Board of Canada statistics, the number of shipping accidents that occurred over the entire geographical region of Canada decreased in 2012 compared to 2007-2011, excluding the Arctic area (See Fig. 1.1).

In several countries, there are systems to assist navigation in icy regions. For instance, in Canada, the Arctic Ice Regime Shipping Systems AIRSS (Transport Canada, 1996) has been implemented as a part of revised Arctic Shipping Pollution Prevention Regulations (ASPPR, 1989). ASPPR is responsible for the safety of shipping operations, setting requirements for hull strength, machinery strength and the limiting ice conditions in which various categories of vessels are permitted to operate (INCW, 2012).

According to Transport Canada's regulations, all polar vessels which navigate in Canadian Arctic water must have one "ice navigator" on-board for handling the planned route [1]. The ice navigator on-board a polar ship is an experienced person who can guarantee guiding the ship in the safest route among ice obsacles. The ice navigator should consider multiple factors, such as ice concentration, stage of development of the ice based on ice information obtained from the CIS, and the start and destination point of the ship, to find the safest and shortest route. As a result of the dynamic texture of ice in the Arctic area, the ice data should be updated constantly. Therefore, implementing an automatic ice navigation system can assist the ice navigator on-board a polar ship to select the proper route. The ice navigation support system to be developed in this thesis will help ice navigators when conducting two basic tasks: 1) identifying the unnavigable ice, and 2) finding the shortest path among all navigable paths between start and end points.

This thesis project is focused on using sea ice information to provide ice navigators with path planning support. In particular, it has investigated algorithms, methods and processes needed to process and analyze ice chart data to obtain optimal paths, which can be easily integrated into future automatic ice navigation support systems, either stand-alone or web-based. Although in this thesis, data from the Canadian Ice Service was employed, the results should be applicable to ice data from other organizations.

1.2 Research Objectives

The ultimate goal of this thesis research is to design a support system to assist ice navigators on-board polar ships to find, through the availability of ice information in different formats, the safest and shortest path among ice obstacles during a ship's voyage. Specifically, the objectives of the research are:

1. To develop a code for determining unnavigable areas in the Canadian Arctic area based on available ice information; and
2. To develop processes for generating virtual path networks in the navigable area based on three path planning algorithms.

---

[1] http://www.tc.gc.ca/eng/marinesafety/debs-Arctic-shipping-operations-ice-navigators-1708.htm

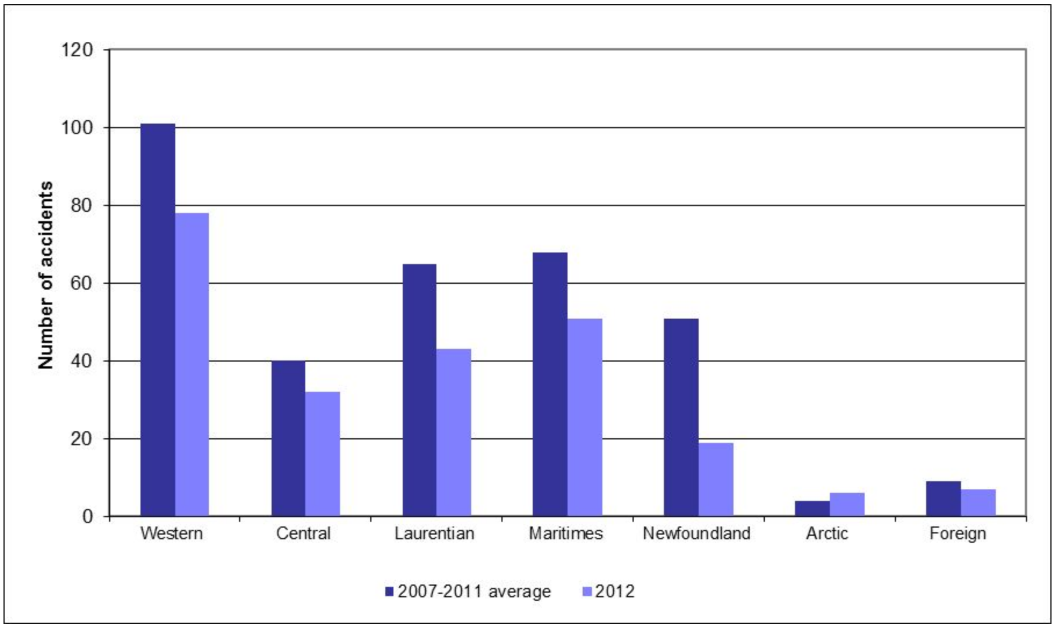


Figure 1.1: Shipping accidents by region, 2003-2012 [1]

To meet the first objective, a Python script was developed. The underlying principle of the script is the calculation of the ice numeral (IN). The IN for each ice polygon was calculated according to its ice information, and the ship type. Then, the unnavigable ice polygons were selected based on the value of the ice numeral.

The second objective was achieved by applying three different path planning algorithms borrowed from the field of robotics. These three algorithms include visibility graph, Voronoi diagram, and Visibility-Voronoi diagram. These algorithms were first applied to a set of synthetic obstacles for evaluation, and then to two different case study areas using real ice chart data.

1.3 Scope and Limitation

The scope of this thesis is designing the support system for assisting the ice navigator on-board a polar ship to find the shortest and safest path in the Canadian Arctic area. The available ice data, in vector format, is weekly ice charts. Also, in this research, the only optimization parameter used to find the best route is distance. Due to time and resource constraints faced by this thesis research, the results are subject to some limitations. First of all, for path planning

---

[1] `http://www.tsb.gc.ca/eng/stats/marine/2012/ss12.asp`

purposes, we ideally would have forecast and daily ice chart data since the ice always has unpredictable movements. Unfortunately, these data from the Canadian Arctic area have not been available in vector format. Secondly, to find the best route, there are plenty of factors that need to be considered, such as weather conditions, legal restrictions, water depth, topography, etc. Therefore, further consideration of these factors will enhance the related decision support.

1.4 Organization of the Thesis

The thesis is organized as follows: Chapter 2 presents a literature review of the sea ice information in the Canadian Arctic area, followed by a review of three appropriate path planning algorithms. Chapter 3 describes the methodology for identifying unnavigable ice areas, path planning, and finding the shortest path. Chapter 4 examines the ice selection and path planning algorithms on real ice data and the results from these algorithms are illustrated and compared. Conclusions of the research are drawn in Chapter 5, along with directions for future work.

Chapter 2

Literature Review

This chapter provides an overview of existing literature related to ice navigation in the Canadian Arctic. For ice navigation purposes, first, in Section 2.1, I review how sea ice forms, and the various types of ice available in the Canadian Arctic. Also, a brief history of Canadian ice information systems are reviewed. The sources for obtaining ice information data in the Canadian Arctic and the available formats which are accessible for current users are summarized in Section 2.2. Moreover, the procedure for ice navigators on-board polar ships applying current ice data to identify navigable areas is discussed. Section 2.3 provides information about all the parameters that should be considered during the voyage in the Arctic. These parameters may include the ship's function, or some environmental elements that help with safe ice navigation. To find the proper path in the Arctic area, three suitable path planning algorithms borrowed from the robotics field are reviewed in Section 2.4. Finally, Section 2.5 summarizes and points out the current need for an automatic support system for ice navigation, which justifies the need for this study.

2.1 Sea Ice

2.1.1 Sea Ice Formation

Sea ice covers approximately 11% of the earth's land surface and 7% of the world's oceans (Hnna, 1998). How is it formed in oceans? Bowditch *et al.* (2002) declared that the maximum density of fresh water occurs at a temperature of 4.0 degrees Celsius, and freezing occurs at zero degrees Celsius. The addition of salt will decrease both the maximum density temprature and the freezing temerature. If the surface seawater's temperature decreases, the surface seawater's density will increase. The convective density-driven currents move warmer water to the surface. If the rate of salinity were equal in the entire body of seawater, it would have to reach the freezing point before ice would begin to form. Since the vertical salinity distribution is different in various parts of the polar sea ice area, ice will begin forming at the water surface as water reaches the freezing point (Bowditch *et al.*, 2002).

2.1.2 Type of Ice

Several forms of floating ice may be met at sea. The most common one is sea ice which results from the freezing of surface sea water. The other forms are river ice and ice of land origin. River ice is met in harbours and bays and normally presents only a temporary interference to shipping navigation (INCW, 2012). Sea ice is typically classified by its stage of development and concentration. Sea ice is not always homogeneous. That is, an ice regime can consist of a mixture of ice types with various concentrations and thicknesses. The following categories illustrate types of floating ice (MSPORC, 2005):

Sea Ice

- The stages of development of sea ice is divided as follows: a) New Ice: Frazil Ice, Grease Ice, Slush and Shuga, b) Nilas: Dark Nilas, Light Nilas and Ice Rind, c) Yong Ice: Grey ice and Grey-white Ice, d) First-year Ice, and e) Old Ice (MSPORC, 2005)

Lake Ice

- Stages of development: New, Thin, Medium, Thick and Very thick according to the thickness: 5, 5-15,15-30, 30-70, 70 (Centimeters) (MSPORC, 2005).

River Ice

- A salinity of less than 24.7 parts per thousand during the ice formation process (MSPORC, 2005).

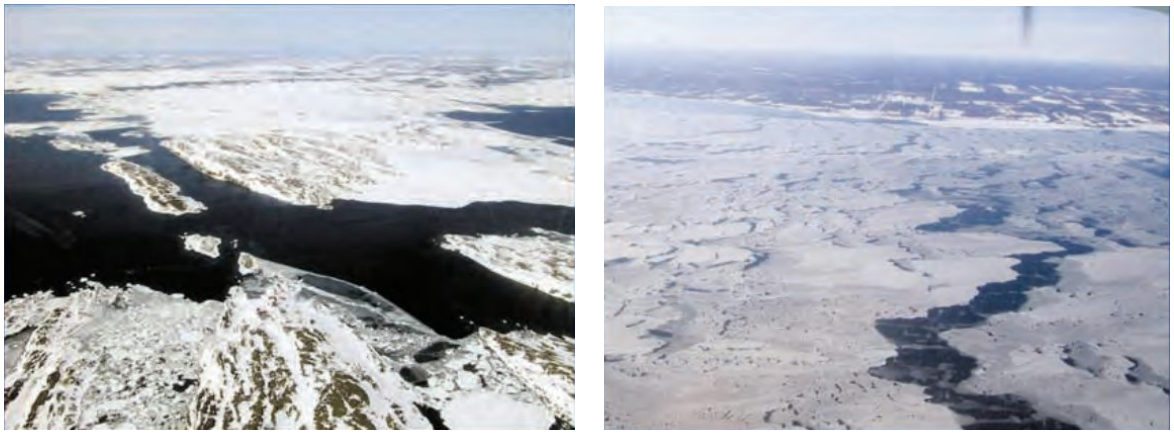
Ice of Land Origin

- Terminology: Firn (recrystallized dense particles), Glacier (massive snow from higher to lower, including inland ice sheets, ice shelves, ice streams, ice caps, ice piedmonts, and cirque glaciers), and various types of mountain (valley) glaciers (MSPORC, 2005).

According to INCW (2012), there are six different stages of development in the Canadian Arctic area based on their characteristics:

New Ice new ice has a soupy texture and matte appearance (INCW, 2012) (See Fig. 2.1(a)).

Nilas Nilas has a dark, matte appearance and its thickness may be up to 10 centimetres; so it does not pose any threat to ships (INCW, 2012) (See Fig. 2.1(b)).



(a) (b)

Figure 2.1: An example of ice data (a) fast ice forming in harbour [2] and (b) very close pack light nials and new ice[2]

Young Ice Young ice is grey coloured and is between 10 and 30 centimetres thick. As a result, it has sufficient hazard potential for non- strengthened ships (INCW, 2012).

First-Year Ice First-year ice is classified into three different stages such as thin, medium or thick, and its thickness typically being more than 30 centimetres and less than one year old. However, it is always hard to say how thick it is, because the colour and surface characteristics are quite constant all the over time (INCW, 2012) (See Fig. 2.2(a)).

Old Ice The swelling pattern of old ice is a permanent characteristic of its surface. Based on its growth history, multi-year ice glaciers vary substantially in size, thickness, and roughness. Old ice is divided into two different categories: second year ice which has survived only one summer's melt and multiyear ice which has survived at least two summers' melts (INCW, 2012) (See Fig. 2.2(b)).

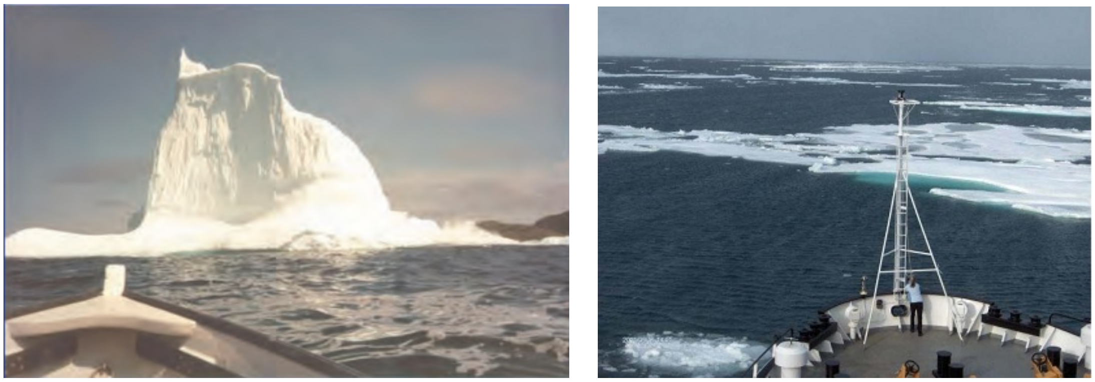
Icebergs, Bergy Bits and Growlers Icebergs, Bergy Bits and Growlers as shown in Fig. 2.3 pose the most dangerous hazards to ships in ice covered waters (INCW, 2012).

2.1.3 Canadian Ice Service

A research article by Howell *et al.* (2004), indicates that navigation through the ice was in the past governed empirically based on the Zone-date system. According to this system, access to Arctic locations was controlled on the basis of calendar dates. However, AIRSS which is

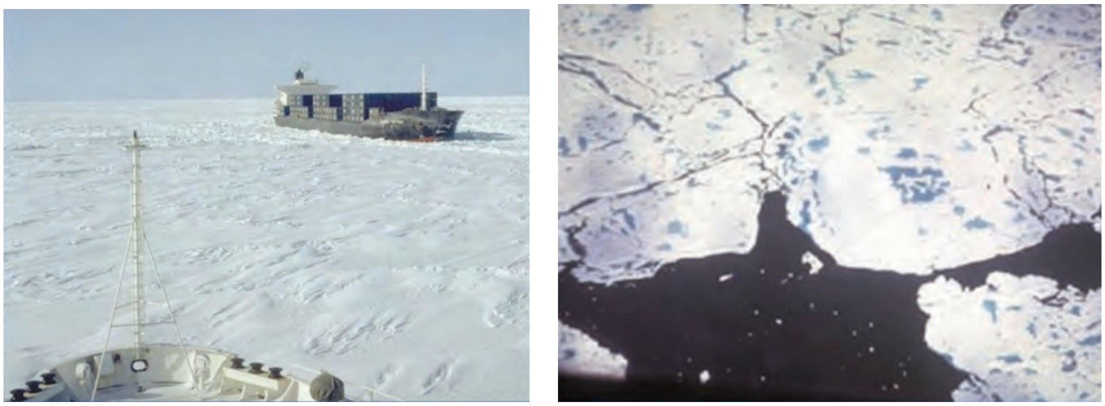
---

[2] `http://ec.gc.ca/glaces-ice/default.asp?lang=En&n=4FF82CBD-1`



(a) (b)

Figure 2.3: An example of ice data (a) pinnacled iceberg[3] (b) vessels encountering multi-year icel[3]



(a) (b)

Figure 2.2: An example of ice data (a) container ship tracking through a large pan of thin first year ice[3] (b) large pans of old ice showing secondary year of survival[3]

responsible for ice navigation in Canada, uses ice conditions provided by the Canadian Ice Service (CIS) and vessel class to provide guidance on safe navigation in ice and improved route efficiency in addiotion to the the zone-data system.

There are two different services responsible for vessel traffic in the Arctic area, "ECAREG CANADA" and "NORDREG CANADA" (INCW, 2012). The former is responsible for the Eastern Canadian Vessel Traffic Service System and the latter is responsible for Northern Canada Vessel Traffic Services (INCW, 2012). According to the ice navigation guidelines,

---

[3] `http://ec.gc.ca/glaces-ice/default.asp?lang=En&n=4FF82CBD-1`

the eastern traffic services system applies to every ship of 500 tonnes, or ships towing vessels with total tonnage of 500, or vessels carrying a pollutant or hazardous stuffs, as defined in law. The northern traffic services system applies to every ship of 300 tonnes, or ships towing vessels with total tonnage of 500 tonnes, or vessels carrying a pollutant or hazardous stuffs (INCW, 2012). The polar guidelines indicate that the most significant factor for vessels to operate in the Arctic area is that ships should be designed based on the Polar Class. Fig. 2.4 illustrates two ice breakers of the Canadian Ice Service on missions.

2.1.4 Level of Ice Information

Ice in the Arctic area has complex and dynamic textures. There are four available levels of ice information to support ice navigation in different periods of a vessel's voyage:

Background

This kind of information, used for strategic planning processes is basically historical information, and describes the variability in space and time of ice conditions for the region of navigation. Moreover, it depicts the relationship of ice conditions to other climatological factors such as wind, current, and tide (INCW, 2012).

Synoptic

This kind of information, used as a support tool for strategic planning, defines ice conditions for a specific area and period of time. This information includes both current and forecast ice information; however, it is not very detailed. This information is useful a couple of days before entering the ice (INCW, 2012).

Route Specific

This kind of information, used for the tactical planning stage, provides more detail thanthe synoptic information. It usually provides for a small area and extends to the identification of each floe and other information about the ice cover (INCW, 2012)..

Close Range

Close range information provides real time information for navigation. In fact, this information, used during the monitoring and implementation of a tactical passage plan, identifies individual hazards which mariners may encounter in the immediate path of the ship (INCW, 2012).

(a) (b)

Figure 2.4: Canadian Ice Service ice breaker[4]

2.2 Data Source for Ice Navigation

The main data source for ice navigation is radar imagery, provided by aircraft and helicopters. For example, RADARSAT, ENVISAT and SAR data all provide detailed coverage of ice information (INCW, 2012). CIS uses these images to cover the Arctic area annually. Consequently, data received in real-time by the two Canadian satellite receiving stations are processed and delivered to the CIS and transferred to mariners. To provide more accurate data for the ice analysis program, US polar orbiting satellites are also widely used. Moreover, microwave imagery is received daily to provide background information on the general ice distribution at low resolution.

2.2.1 Ice Chart

Ice charts published by CIS describe ice conditions at different locations. To publish ice charts, standard World Meteorological Organization (WMO) terms and symbols are used. The ice charts are the synoptic level information sources, and the averages for the area are depicted as ice conditions. Therefore, mariners should consider that there is always the possibility that local ice conditions may vary significantly from what is depicted on the charts. The ice charts are published daily at 1800 UTC during operating seasons (INCW, 2012).

According to Howell *et al.* (2004), since ice charts over the past 34 years lack information and coherence, CIS ice charts do not contain complete historical information about ice decay and ice ridging, and so the ice cannot be modeled accurately. Moreover, the CIS ice charts before 1983 were coded by ratio code instead of egg code. Both of them describe the sea

---

[4] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`

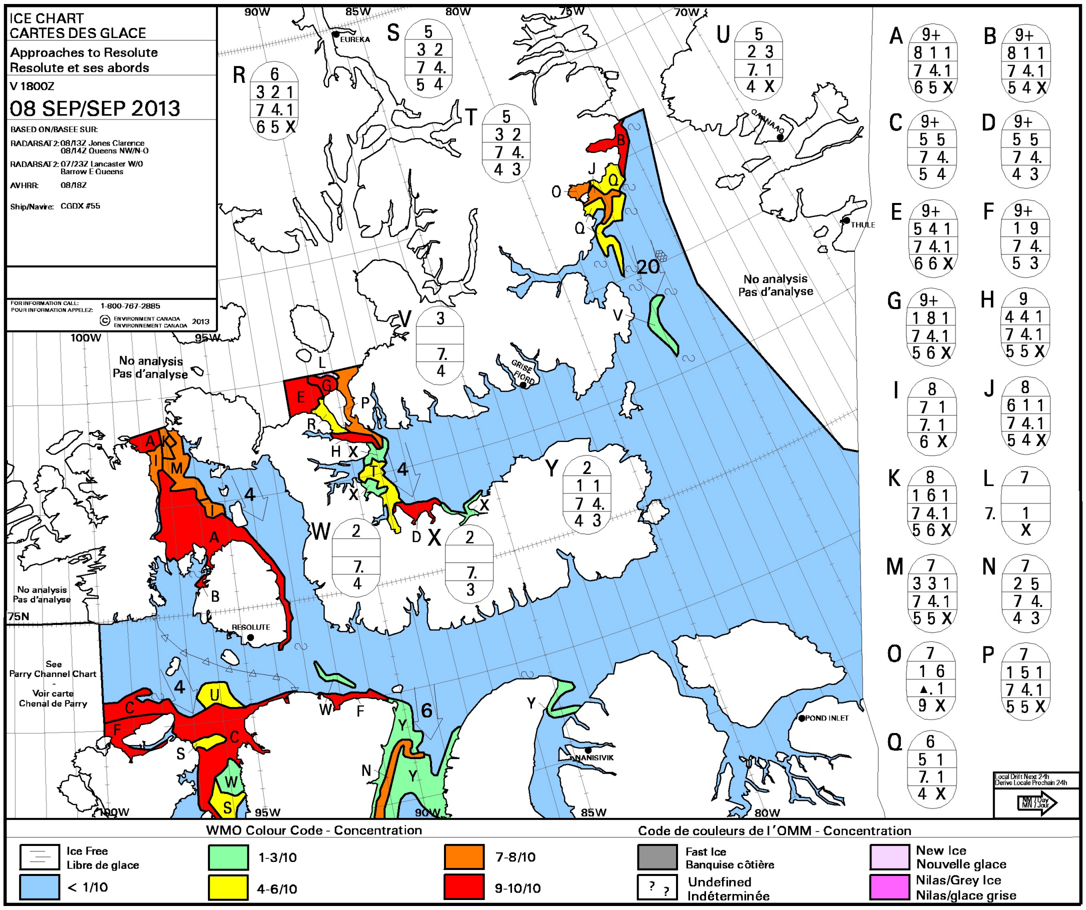


Figure 2.5: Ice chart sample from the Canadian Ice Service Archive[5]

ice regime; however, the egg code provides more information. In fact, egg code has distinct categories for old ice divided into multi-year sea ice and second year ice, but ratio code includes only old ice without any subdivision. Fig. 2.5 shows a sample of ice charts coded by egg code for the 8th of September 2013 in the Canadian Arctic.

2.2.2 Ice Egg

The "Egg Code" is a coding that contains data regarding stages of development or "age", form of ice or "floe size", and concentrations in an egg shaped description. This oval can contain a description of a maximum of three ice types. Fig. 2.6 depicts a summary diagram of the "Egg Code". This code follows the international agreement and shall be used in coding all sea ice and lake ice observations, where ranges are shown for thickness, floe size or other dimensions.

---

[5] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`

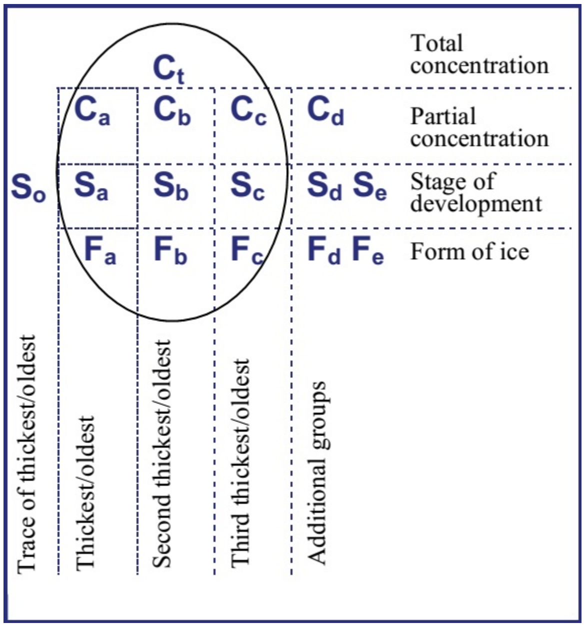


Figure 2.6: Egg code diagram[6]

2.2.3 Ice Multiplier

Each ice type including open water has a specific numerical value based on ice category and type of vessel. This number is called the ice multiplier (IM). In fact, ice multiplier is a unique value for the specific ice regime and type of ship. Fig. 2.7 illustrates an Ice multiplier table for different types of ship within different ice categories.

2.2.4 Ice Numeral

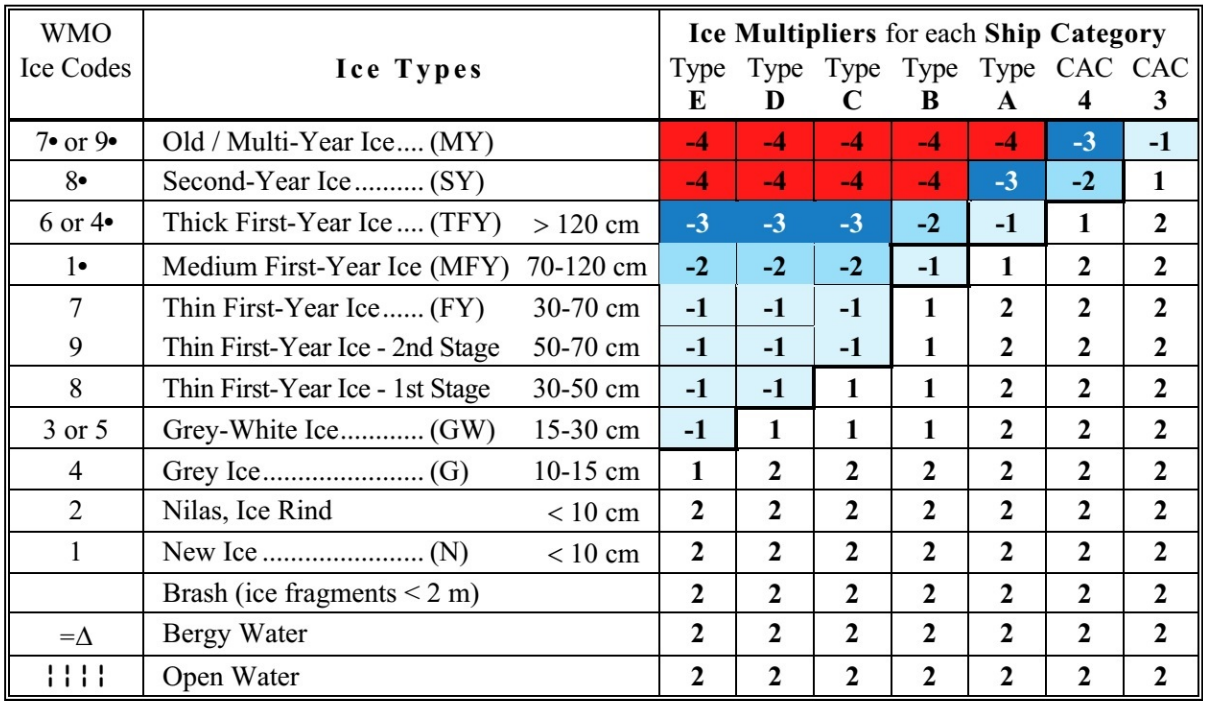
The ice numeral, which is a mathematical evaluation of an ice regime for determining whether a specific type of ship can enter that particular ice regime, is calculated by taking the sum of the products of the concentration of ice types present and their ice multiplier for any specific ice regime (INCW, 2012). The IN calculation presented in Eq. 2.1 takes into account ice concentration and the ice multiplier.

$$IN = (C_a \times IM_a) + (C_b \times IM_b) + ... + (C_n \times IM_n) \tag{2.1}$$

where C_a is the concentration in tenth of ice type “a” and IM_a is the ice multiplier for ice type “a”, and so on.

---

[6] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`



WMO Ice Codes	Ice Types		Ice Multipliers for each Ship Category Type E	Type D	Type C	Type B	Type A	CAC 4	CAC 3
7• or 9•	Old / Multi-Year Ice.... (MY)		-4	-4	-4	-4	-4	-3	-1
8•	Second-Year Ice.......... (SY)		-4	-4	-4	-4	-3	-2	1
6 or 4•	Thick First-Year Ice.... (TFY)	> 120 cm	-3	-3	-3	-2	-1	1	2
1•	Medium First-Year Ice (MFY)	70-120 cm	-2	-2	-2	-1	1	2	2
7	Thin First-Year Ice...... (FY)	30-70 cm	-1	-1	-1	1	2	2	2
9	Thin First-Year Ice - 2nd Stage	50-70 cm	-1	-1	-1	1	2	2	2
8	Thin First-Year Ice - 1st Stage	30-50 cm	-1	-1	1	1	2	2	2
3 or 5	Grey-White Ice............ (GW)	15-30 cm	-1	1	1	1	2	2	2
4	Grey Ice....................... (G)	10-15 cm	1	2	2	2	2	2	2
2	Nilas, Ice Rind	< 10 cm	2	2	2	2	2	2	2
1	New Ice .... (N)	< 10 cm	2	2	2	2	2	2	2
	Brash (ice fragments < 2 m)		2	2	2	2	2	2	2
=Δ	Bergy Water		2	2	2	2	2	2	2
¦ ¦ ¦ ¦	Open Water		2	2	2	2	2	2	2

Figure 2.7: Ice multiplier table[7]

As one can see, the ice numeral mainly depends on the particular ice regime and category of ship operating in the Arctic area. If the ice numeral calculated for a vessel is zero or positive,the vessel can proceed to that ice regime. However, if the ice numeral for any ice regime is negative, the vessel must use an alternate the plan such as selecting another route, waiting for the ice breaker to assist the vessel, or waiting for improved ice conditions (INCW, 2012).

2.2.5 Ice Forecast Information

CIS provides not only ice information at the synoptic level but also long-range forecasts to support marine activities. It also broadcasts to mariners valuable information for strategic planning such as warnings, short-range forecasts for ice, and iceberg conditions. According to Bowditch (2002), sea ice forecasts are mainly separated into four different categories: short-term (24-72 hour), weekly (5-7 day), monthly (15-30 day) and seasonal (60-90 day) forecasts. Short-term forecasts mainly concentrate on local scale analysis such as change in the ice cover area based on ice drift, ice formation, and erosion. On the other hand, weekly ice forecasts concentrate on the changing of ice edge location and its concentration. CIS uses many different empirical

---

[7] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`

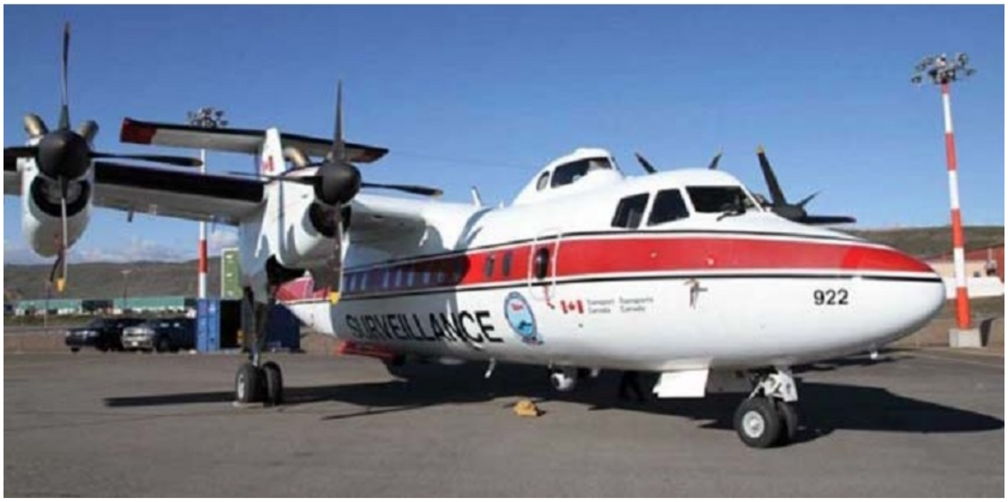


Figure 2.8: CIS ice surveillance and forecast[8]

prediction models to provide short-term and long-term forecasts. These predictive simulation models fundamentally relate ice drift with meteorological terms such as currents, wind and some thermodynamic models called the Polar Ice Prediction System (PIPS) (Bowditch, 2002). Fig. 2.8 illustrates the aircraft operated by Transport Canada and organized with CIS staff. The aircraft flight mission combined visual and radar investigation missions (INCW, 2012).

Ice conditions are difficult to predict because the drift caused by wind and tidal currents varies. Therefore, the results are always complex, and the information of oceanographic factors is not often accessible (INCW 2012). So, in most cases, the wind drift is determined and is forwarded to individual ships' captains to interpret in conjunction with their own positions and situations. New developments in computer processing and modeling for ice prediction are allowing more detailed data to be provided. Usually, ice forecasts are prepared once a day for a period of 24 to 144 hours because they are related to the frequency of the data involved. (INCW 2012).

2.2.6 Ice Drifting

Fig. 2.9 shows ice drift happening in the Arctic area. Even though surface currents have some effect on the drift of pack ice, the main factor is wind (Bowditch, 2002). Due to the Coriolis force, ice does not drift in the same direction of the wind depending on the force of the surface wind and the ice thickness. In the Northern Hemisphere, this drift is to the right side of the

---

[8] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`

Figure 2.9: Ice drifting[9]

wind direction, and in the Southern Hemisphere it is to the left. According to the equilibrium terms, the drift angle increases with increasing ice thickness and declining surface wind speed.

2.3 Ice Navigation Parameters

The ice navigator on-board polar ships should consider various parameters to finalize the path from the start point to the destination point. The actual visibility, weather conditions, maneuverability, condition of the ship, and sailing area are taken into account by the ice navigator as the safety conditions during the voyage.

Visibility

Although ice charts have an important role for vessels traveling in ice-covered regions, their significance is not comparable with real-time observations made from the bridge. Visibility of ice in the ocean depends on different parameters, including meteorological visibility, height of the iceberg, source and condition of light, and the observer. For instance, on a clear day with excellent visibility conditions, icebergs can be seen at a distance of 20 miles (INCW, 2012).

Speed

In all situations in the Arctic, it must be remembered that the force of impact of the ice varies as the square of the speed.Accordingly, even if the speed of a ship is less than

---

[9] `http://www.ccg-gcc.gc.ca/Ice_home/Ice_Publications/Ice-Navigation-in-Canadian-Waters`

doubled, the energy of an impact with any piece of ice may be more than doubled (INCW 2012). The safe speed in a specific ice regime is a result of visibility, ice type, and the maneuverability of the ship.

Maneuverability

The northern sea route is largely ice free for 8 weeks, from August to October, but ice-strengthened vessels can sail through these shipping lanes throughout the winter with the help of Icebreakers (INCW, 2012).

2.4 Path Planning

The concept of path planning occurs in frequently in technical applications such as path planning for dynamic objects (Yap, 1987), and safe vessel navigation in oceanic collision circumstances (Kolendo *et al.*, 2011). Kolendo *et al.* (2011) defined the concept of path planning in the following way: a given moving object; a given description of the environment; and the best path between a start and an end point in the given environment which passes through the open area, avoids all static and dynamic obstacles, and meets certain optimization criteria.

Generating non-collision paths for ships has become a challenging issue due to the increasing numbers of ships. To generate the safest path different models, such as static or dynamic, etc. (Lisowski., 1985; Satheros, 2008), and various methods (linear, non-linear, or evolutionary algorithm), are applied (Smierzchalski *et al.*, 2000). The compulsory trajectory for safe ice navigation is generated in two different phases: “off-line” and “on-line” phases. In the off-line phase, all ice navigation parameters (e.g., speed of ship, location of ice obstacles) are assumed constant. However, the ARPA (Automatic Radar Plotting Aids) system monitors all navigation parameters while a ship navigates through its trajectory. In the on-line phase, required corrections may be applied to the path based on the safety requirements.

In Section 2.4.1, three static path planning algorithms will be discussed which can help the ice navigator on-board a ship find the most feasible route through unnavigable ice. According to the INCW (2012) manual, when the decision has been made for a vessel to proceed into an ice covered area, a “Strategic Plan” is mandatory. First, a “Base Plan” of the route is selected based on safe open water navigation principles. Then, a map of the current ice conditions is overlaid on the ”Base Plan” chart, and selected routes are revised to avoid the ice completely if possible, or to minimize the time in serious ice conditions, which is called the “Strategic Plan”. The most important principle in the ice navigation is to avoid ice if one can. Even if moving in open water enlarges the distance between ports by a substantial amount, the time difference can compensate by increased speed in open water. As mentioned above, there are two steps for ships' operating planning in the Arctic area:

- Strategic Plan
  - In this phase mariners should consider a small-scale plan several days or weeks before encountering ice covered areas. This plan may be revised many times before the tactical phase. The strategic plan has four stages:
    1. **Appraisal**
       In this phase, ice navigators use all information sources, and obtain a complete picture of the ice conditions.
    2. **Planning**
       In this step the ice navigator will develop the route to the destination based on information in the previous phase.
    3. **Execution**
       When the planning of the passage has been finalized, the tactics for execution of the plan can be decided.
    4. **Monitoring**
       As the ship approaches the ice-covered area, not only the quality of ice information but also the accuracy of time of arrival will be updated and the route may be finalized.

According to the INCW manual (2012), while the vessel navigates towards the destination, ice conditions are changing continually, so the "Tactical Plan" should be considered. As the original strategic routes are modified to reflect the movement of ice, real time decisions for maneuvering are based on the analysis of what the ice navigator sees from the bridge. It is the final process which no electronic system has been able to achieve, because the experienced ice navigator is inadvertently processing all the clues in his mind. These clues consist of the different types of ice, the selection of dangerous floes from others which do not threaten damage, the thickness and age of the individual floes, the advance or retreat of the melting processes on the ice-field, the snow-cover covering dangerous ice features, the effects of wind and tide, and the other small hazards which remain undetected by any electronic sensing systems.

- Tactical phase
  - In this phase, ice navigators should consider a large-scale plan while they navigate towards ice covered areas. This plan may be revised many times during the voyage. the tactical plan has four stages:

1. **Appraisal**
   In this phase, ice navigators use all the data sources to acquire CIS observations and charts. In addition, the electronic navigation charts already provide resources for adjustment of ship-tracks. So, a “Tactical Plan” modification could be easily produced with an ice-map overlaid on the navigational data.
2. **Planning**
   In this step, the ice navigator will develop the route to the destination based on open water leads, first year ice leads, and avoidance of old ice areas and areas of pressure or potential pressure.
3. **Execution**
   Estimated times of arrival can be revised based on the finalized route.
4. **Monitoring**
   Progress of ice navigation should be monitored.

Consequently, there are many benefits to having support system available to apply frequent changes to the route. This platform will not only help the navigator make better navigation decisions through the ice, but also provide reliable results for frequent adjustments happening in ice navigation.

2.4.1 Path Planning Algorithms

There are different approaches to automatic path planning. Three feasible approaches were selected for ice navigation purposes including, Voronoi diagrams, visibility graphs and Voronoi-Visibility diagrams. The concept of each algorithm will be described in this section and the methodology of implementing each one will be explained in Chapter 3.

Path Planning Environment

Ships in the Canadian Arctic area navigate through some natural constraints such as land; ice bergs; shallow water; or other constraints caused by general regulations, for instance, traffic restricted areas defined by the Canadian Coast Guard. In this research, the natural constraints are assumed static, and defined by polygons. The degree of collision threat with those constraints depends on two distinct approaches: distance at closest approach point, and time of closest approach (Smierzchalski *et al.*, 2005). In this research, it is anticipated that all constraints which have a degree of threat for a ship appear in the path planning environment.

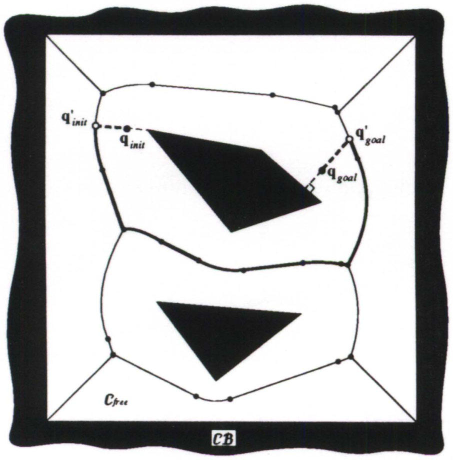


Figure 2.10: Voronoi diagram[10]

Voronoi Diagram

Acquiring a safe trajectory for ships is critical for ice navigation. The Titanic disaster happened in 1912 near Newfoundland and caused the death of approximately 1500 people, which was one of the deadliest tragedies in modern history. It is one of the best examples of the importance of safety in ice navigation. The Voronoi diagram is a well-known path planning algorithm for generating safety routes because it gives has the maximum clearance from all obstacles. However, this algorithm is demanding to generate. The concept of Generalized Voronoi Diagram (GVD) was first introduced in 1967 (Blum, 1967). Blum (1967) compared this symmetric method with the grass fire phenomenon where the fire in the boundaries of a grass field erupted to the center of the field. As the fire moved forward and was extinguished at some points, it shaped medial axes. Blum called these points the centers of Maxima Inscribed Disc (MID). Fig. 2.10 represents the concept of Voronoi diagram algorithm for the two convex obstacles in the specified environment. As Fig. 2.10 shows, the GVD consists of three elements, (1) connecting the start point to the GVD, (2) navigating through the Voronoi road map, (3) building a pathway to the neighborhood of the goal point. The Voronoi road map is formed by a set of points which are equidistant to three or more polygons. A well-known algorithm dealing with incremental construction of Hierarchical Generalized Voronoi Diagrams (HGVD) was developed by Choset and Burdick (2000) and Choset *et al.* (2000).

1. **Reached to the Voronoi Diagram**

---

[10] `http://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/basicmotion.html`

This component provides information about the start position in its environment. From the start point, numbers of rays emit toward the surrounding obstacles. Then, based on the angles and the magnitude of all rays, the point starts to move in the specific direction to become equidistant from the three obstacles closet to the start point. After this, the first vertex of the Voronoi Diagram is created (Masehian et al, 2010).

2. **Navigate through the medial axis**
   This step has four different components:

- *Projection Component*
  This component is the leading element of the motion planning section. In fact, this part moves the robot by small steps in the Voronoi trajectory with the proper vector size and direction from the current point (Masehian *et al.*, 2010).

- *Correction Component*
  This component keeps the tracking point on the Voronoi trajectory by checking the distance of neighbouring obstacles and their angles. Actually, this component revises the location of the tracking point and guides it into the stable state when it is realized that tracking point is off the road map (Masehian *et al.*, 2010).

- *Loop Handling Component*
  This component is used when the tracking point enters the previously-visited vertex of the Voronoi diagram. So, in this case, the tracking point tries to explore uninvestigated areas of its work space (Masehian *et al.*, 2010).

- *Backtracking Component*
  If the tracking point collides with any obstacles, or the environments' boundaries, during the Voronoi edge investigation, the backtracking components would be activated and turn the tracking point back to the last visited vertex point (Masehian *et al.*, 2010).

Visibility Graph

A visibility Graph is an approach taken from computational geometry and robot motion planning graphs. Actually, the visibility graph is a well-known approach for collision-free shortest path planning between a given start point and a given goal point (De Berg *et al.*, 2000). Fig. 2.11 represents the concept of Visibility graph algorithm for the three obstacles. As Fig. 2.11 demonstrates, the structure of the visibility graph suggested by De Berg *et al.* (2000) has two components: the set of polygons' vertices, and edges which are the connections of pairs of vertices that are visible to each other. In fact, it is necessary to consider only the edges that do not overlap the interior part of any obstacles. In general, the visibility graph is the tangent to the obstacles called valid visibility edges (Wein *et al.*, 2005).

Visibility-Voronoi Diagram

The Visibility-Voronoi (VV) approach is the hybrid method which evolved from the visibility graph with zero clearance, and the Voronoi diagram with infinity clearance (Wein *et al.*, 2005). Wein *et al.*, (2005) introduced a modifed version of the visibility graph which combines the visibility graph with the Voronoi diagram (De Berg *et al.*, 2000) (See Fig. 2.12). Fig. 2.12 depicts the VV approach for four convex obstacles in the specified environment. The cyan polygons are convex obstacles, the boundary of the expanded obstacles is drawn in a solid line, the relevant portion of the Voronoi diagram is dotted, and the visibility edges are drawn using a dashed line. The result from the VV algorithm is short, smooth, and dose not come too close to the obstacles. Actually, this method generates a natural-looking path which is very close to the possible route taken by human sense (Wein *et al.*, 2005).

- **Buffering the Obstacles**
  To have a proper clearnce from all obstacles and to avoid touching any obstacles' boundaries, all of them are buffered with the radius of specific clearance (C).

- **Constructing Visibility Graph**
  In this stage, the visibility graph algorithm is applied to the buffered obstacles. As a result, the visibility graph of the buffered obstacles contains all the possible routes with a clearance of at least C from all obstacles. Hence, this guarantees that the shortest path extracted from this method is both smooth and contains no sharp turns. Wein *et al.*, 2005 mentioned that VV with zero clearance is the Visibility graph approach and VV with infinity clearance is the Voronoi diagram.

Shortest Path Planning Algorithm

There are many approaches to find the shortest path among all available paths between a start and end point. One of the approach is Dijkstra's algorithm which is the single-source shortest path problem (SSSP) for a graph with non-negative edge path costs (Dijkstra, 1959). In this method, for a given start and end point in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path). The cost known as a weigh matrix can be the distances between graph's nodes or combination of distance with other criteria, such as risk of collisions, time-dependent networks, and etc.

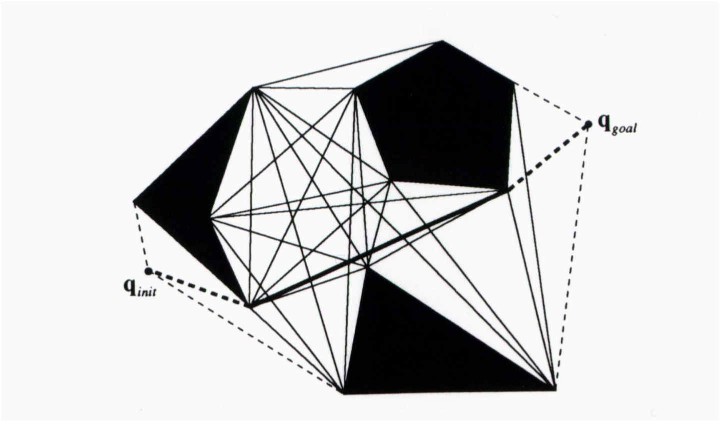


Figure 2.11: Visibility graph[11]

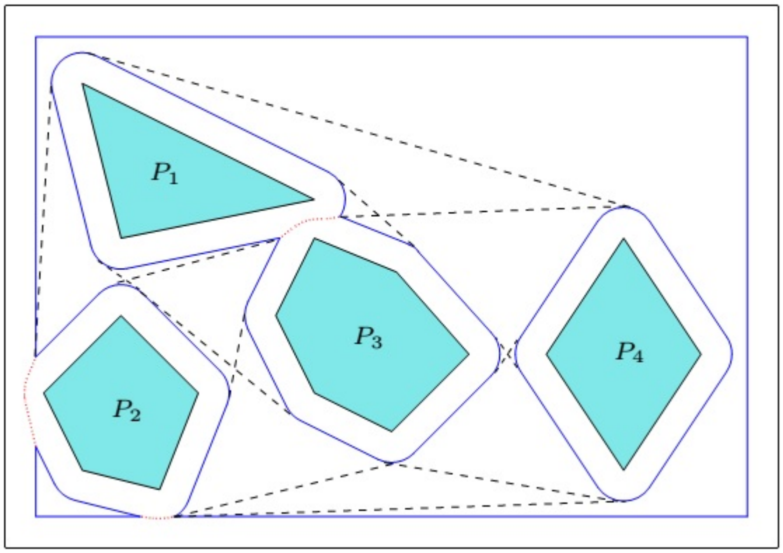


Figure 2.12: Visibility-Voronoi diagram[12]

---

[11] `http://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/basicmotion.html`

[12] `http://acg.cs.tau.ac.il/projects/internal-projects/the-visibility-voronoi-complex/project-page`

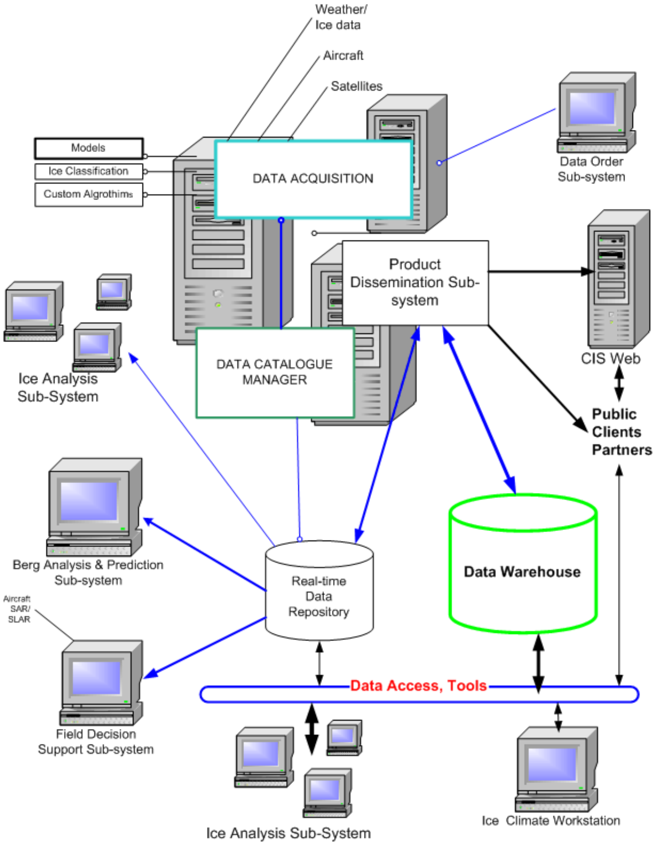


Figure 2.13: GIS Implementation of Ice Navigation [13]

2.5 GIS Implementation of Ice Navigation

Fig. 2.13 illustrates an Integrated Spatial Information System (ISIS) which can be used for implementing ice navigation system. An ice Navigation system must consist of three major components: (Koonar *et al.*, 2002):

- Sensor networks for obtaining the information including metrological and remotely sensed data of surface ;
- A system to transfer and distribute the ice information to/from different servers; and
- A digital platform for data integration, analysis and production.

Once the observational material from all sources has been combined into an ice chart which represents existing conditions, the ice center then has the task of distributing the chart to

---

[13] `http://proceedings.esri.com/library/userconf/proc02/pap1302/p1302.htm`

mariners while it is still usable. The ice data can also be converted to a predictive model by combining them with the meteorological and oceanographic parameters to provide further guidance to vessels in or near the ice.

2.6 Discussion

Navigation in ice-covered areas involves significant advanced planning and more safety measures than those taken with a typical ice free voyage (Bowditch *et al.*, 2002). The crews of polar ships should have sufficient fundamental knowledge for handling polar ships. Since a wide range of ice navigation parameters must be considered by the ice navigators with a board range of available ice data, implementing an automatic system to assist the ice navigator on-board polar ships for path planning is vital.

The concept of non-collision decision making in ship navigation appeared in an article of “the father of the differential game” by Isaacs, (1965). Although the concept of non-collision marine navigation seems obvious, uncertainty of information may result from external factors (sea state, weather conditions); in addition, incomplete knowledge about the other objects, such as ice obstacles, render the decision-making about ship trajectory complex (Lisowski, 2007). The problem of determining the safest trajectory for ice navigation is still a crucial concern since the number of polar ships navigating in the Canadian Arctic is increasing considerably. Computer supporting-system can certainly navigators’ duties. As a result of increasing demand for safe ice navigation, ships’ operators are required to install integrated, supported navigation systems on-board polar ships to guarantee that the ship’s trajectory observes all the requirements of the International Regulations for Preventing Collision at sea (COLREG) (Lisowski, 2007).

The previous literature reviews have addressed some difficulties for ice navigation path planning. Ice navigation can be categorized as a set of tasks with diverse time and multi-level control processes. The most complex part of the longest voyage is termed “Path Planning”(Smierzchalski *et al.*, 2000), decided on the basis of factors such as historical ice data, route length, weather forecast conditions, fuel consumption, and ship type. The most important part of ice navigation is a safe trajectory from the start to the destination point, which should consider existing navigational limitations including the existence of unnavigable ice-covered areas, and the restricted zone defined by the Canadian Coast Guard. In Chapter 3, the procedure for implementing the INSS is presented.

Chapter 3

Methodology

This chapter explains the methodology of implementing an ice navigation support system. The methodology of the INSS consists of the following sections: Section 3.1 summarizes the criteria to chose proper software to implement the INSS; Section 3.2 describes the overall workflow of the methodology employed in this research; and Section 3.3 describes the data set and study area of this research. In this section (Section 3.3), the ice data which is available for ice navigation in the Canadian Arctic and the coverage of ice data in this area will be discussed. At the end, a brief summary of this chapter is provided.

3.1 Software Selection Criteria and Process

Implementation of the ice navigation support system (INSS) is divided into two major parts: first, selection of unnavigable ice areas; second, path planning algorithms. In order to be able to determine what software or programming languages would be suitable for the INSS, a list of criteria was made in terms of functionality, accessibility, and computational skills. The following list of criteria were made to help with the choice of software for the selection of unnavigble ice areas:

1. A software package that is able to convert the “e00” file type to the shapefile.
2. A software that can overlay different layers for processing.
3. A software that has the ability to do calculations based on the values in the attribute tables of datasets.
4. A software package that has ability to select subsets of a dataset and combine different datasets together.

5. A software package that is able to combine different feature classes into one feature class.
6. A software package that has the ability to perform map generalizations of GIS datasets.
7. A software package that provides an Application Programming Interface (API) to evoke internal functions.

After much deliberation and consideration of the criteria, ArcGIS was chosen which satisfies all of these criteria. Esri's ArcGIS is a geospatial information system platform which has powerful abilities to generate and work with GIS datasets. Also, this software has the Python programming language as its API to automate all processes.

To implement the second part of the INSS's part part which is path planning algorithms, a list of criteria similar to the "determining unnavigable ice areas" part was made to determine what programming language would be suitable for implementation of path planning algorithms:

1. A programming language package that has the ability to implement algorithms.
2. A programming language that is able to import and export KML files.
3. A programming language that can do matrix manipulations and calculations.
4. A programming language that has the ability to plot functions and data.
5. A programming language that is able to create functions.
6. A programming language that has the ability to perform complex numerical computations.

Since MATLAB has been widely used in the research community, there exists a large number of functions and algorithms developed in the toolbox or by the research community that can be easily customized or built upon for the experimental work. Therefore, MATLAB was used as an integrated development platform (IDE) to implement the existing navigation algorithms as well as the proposed methods.

3.2 Workflow Overview

The overall workflow of the methodology employed in this thesis is represented in Fig. 3.1. The flowchart in Fig. 3.1 is divided into two main sections describing the steps taken in order to obtain the results in every part of the methodology. These two major steps summarize the tasks: 1) study area and determining unnavigable ice areas 2) path-planning algorithms.

As mentioned, the first step in this methodology deals with determining unnavigable ice areas. The overall workflow for determining unnavigable ice areas is represented in Fig. 3.2.

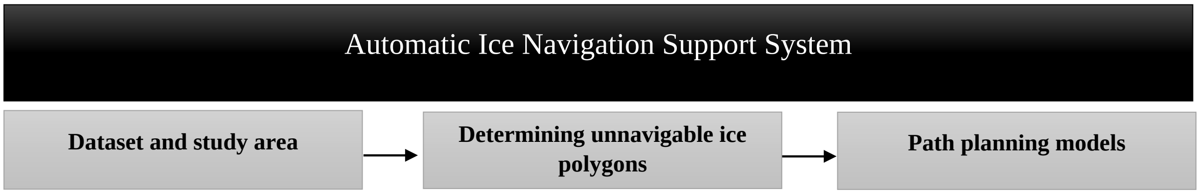

Automatic Ice Navigation Support System
Dataset and study area
Determining unnavigable ice polygons
Path planning models

Figure 3.1: Overall workflow

3.2.1 Determining Unnavigable Ice Polygons

Selection of unnavigable ice areas is the process of selecting the ice which is unnavigable for a specific type of ship in a specific period of time. A script was developed in Python 2.7 which is compatible with ArcGIS 10.2 or any other Integrated Development Environment (IDE) compiler such as pythonwin to process the selection of unnavigable ice polygons. The following steps were taken to select the unnavigable polygons from CIS weekly ice charts. The algorithm has two main parts. One part is importing, selecting, dissolving, clipping, and converting to a KML file; and the other part is calculating IN for each ice polygon. Fig. 3.2 shows the overall workflow for the selection of unnavigable ice polygons.

Importing Sea Ice Data

The first step is importing weekly ice chart data. As mentioned before, the weekly ice chart data is available on-line on the CISA website. In this part, firstly, the weekly ice chart data for the specific month was obtained from the CISA website and stored on the local hard drive. To import the data and convert it to the shapefiles, the "import from e00 file" script tool was applied. Then, the file name for each ice dataset was changed before it was converted to the shapefiles to make its file name smaller and more usable.

Calculating the Ice Numeral

As discussed in Chapter 2, based on the ice numeral concept, decisions can be made whether an ice polygon is navigable or not. Fig. 3.3 shows the overall workflow for calculating IN. In this step the IN for each polygon was computed and stored in the attribute table of the ice dataset, according to Eq. 2.1. All ice datasets have information about the ice concentration and stage of development of each ice polygon. Based on this information, IN can be calculated and stored into the attribute table of each ice datasets. So, a field was added into the attribute table of each ice dataset and the ice numeral was calculated and stored into the added field. To implement all these steps automatically, a Python script was developed as "CIceNum". This

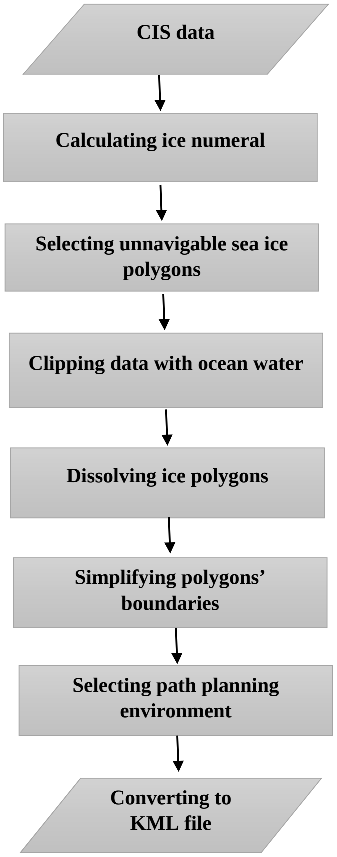

CIS data
Calculating ice numeral
Selecting unnavigable sea ice polygons
Clipping data with ocean water
Dissolving ice polygons
Simplifying polygons' boundaries
Selecting path planning environment
Converting to KML file

Figure 3.2: Overall workflow of selection of unnavigable ice

script calculates the ice numeral (IN) based on the total concentration and the first three partial concentrations of each ice polygon. All the ice multiplier values for calculating ice numerals were obtained from the INCW's ice multiplier table. The developed python script for selecting unnavigable ice area illustrates in algorithm 1.

Selecting Unnavigable Sea Ice Polygons

Then, the ice selection algorithm was applied to select the unnavigable icepolygons. According to the calculated IN, unnavigable ice polygons can be selected using feature selection tools available in most geographic information systems . Based on what was discussed in Chapter 2, if the ice numeral (IN) for each ice polygon is zero or negative, it will be considered as "unnavigable"; however, if the value is positive, it will be considered as "navigable". Consequently,

Algorithm 1 Determining Unnavigable Ice Polygons

```
function DetUnnavIcePolys()

    Input: CIS data
    function e00 to shapefile conversion(CIS data)    ▷ Data type conversion
        for $i$ = number of e00 files do e00 files are converted to the shapefiles (feature calsses)
            In this step the projection of ice data are defined
        end for
    end function
    for $i$ = number of feature classes do the following functions
        function Calculating ice numeral(shapefiles)    ▷ Calculating ice numeral
        end function
        function Selecting unnavigable ice polygons(shapefiles)    ▷ Identifying unnavigable ice polygons
        end function
        function Clipping ice data(shapefiles)    ▷ Clipping ice data with ocean water
        end function
        function Dissolving ice polygons(shapefiles)    ▷ Join multipart feature classes to one
        end function
        function Simplyfying polygons' boundaries(shapefiles)    ▷ Simplyfying the polygons' boundaries
        end function
    end for
    Path planning environment is selected
    The environment is converted to KML file
end function
```

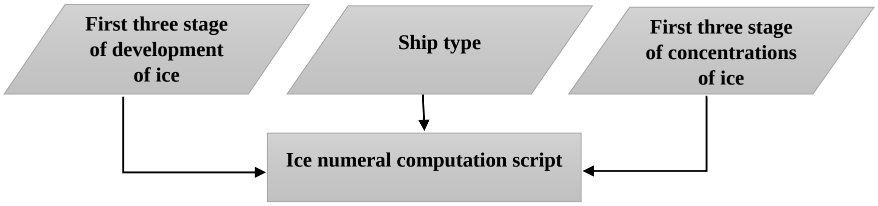

First three stage of development of ice
Ship type
First three stage of concentrations of ice
Ice numeral computation script

Figure 3.3: Overall workflow of ice numeral calculation

the unnavigable ice polygons were selected and stored in the new shapefile.

Clipping Data with Ocean Water

As mentioned in Section 3.1, the raw data covers the coast area in some regions. To remove the coastal area, the unnavigable ice data derived from the last step should be clipped with a Canadian water boundary map. The clipped data demonstrates the ice areas within the water body boundaries, which is usable for ice navigation.

Dissolving Ice Polygons

In this step, the dissolve tool was applied to join multipart feature classes of ice data into the unique feature class containing noncontiguous element. Thus, the entire selected obstacles from different feature classes combines into a single feature class. As a result, dealing with one shapefile is more convenient for further analysis.

Simplifying Polygon Boundaries

Dealing with complex obstacles for any path planning method are both time consuming and intensive. To simplify the polygons' boundaries and decrease the number of vertices, the simplifying polygon tool was applied to make the polygon boundaries simpler. In fact, this tool removes unnecessary vertices while preserving essential shape of the obstacles.

Selecting the Path Planning Environment

For any path planning purpose, a region should be selected consisting of a start and goal point and all the constraints with their degree of collision threat. In this research, ice polygons were focused on as the major threat in the Canadian Arctic. Therefore, both the unnavigable ice area and the coastal boundaries were selected as the "path planning environment".

Converting to KML File

Up to this point, the unnavigable obstacles were selected, transformed to the unique feature class, and simplified. In order to transfer the selected path planning environment to the Matlab environment for further path planning, it was converted to the KML file which is compatible with Matlab. KML stands for Keyhole Markup Language which is an XML notation for representing geographic annotation and visualization. "Layer to KML" is the tool which is used to transfer the shapefile to the KML file. The output from this tool is the KMZ file which is the zip format of KML. To convert it to the KML file, the KMZ file should be unzipped.

3.2.2 Path Navigation Algorithms

In this section, three different path planning algorithms were implemented for a ship. These algorithms always find paths without any quality considerations. That is, the path might make a large detour, or make lots of unnecessary turns. Whether the path is good or not depends on different parameters, such as the maneuverability of the different ships, weather conditions, and sea bed properties. In addition, as a result of the increasing number of commercial vessels, finding the shortest path is a priority for for the path planning in order to transport more goods or people in less time. In this research, quality control of the route was ignored, and only the Euclidean shortest path for the ship is illustrated by the three well-known path planning algorithms; however, in practical situations, not only the shortest path but also a safe one would be preferred. The following shortest path planning algorithms were implemented in the Matlab environment.

Voronoi Diagram

In the Voronoi diagram, the following steps were implemented to produce all candidate paths and define the shortest one based on Dijkstra's algorithm. As Fig. 3.4 describes, the Voronoi diagram function consists of different functions, such as importing the data into the Matlab workspace, converting concave obstacles to convex ones, creating the Voronoi diagram, generating the adjacency graph, and the Dijkstra algorithm. The pseudocode shown in algorithm 2 describe what functions are called within MATLAB environment, where these functions are customized based on the open source codes from MAthWorks file exchange[14].

The proposed method was implemented in the Matlab environment to process the Voronoi approach depicted in Fig. 3.4.

- **Importing Data**

---

[14] `http://www.mathworks.com/matlabcentral/fileexchange`

Algorithm 2 Voronoi diagram for ice obstacles

```
function VORONOI DIAGRAM()
    function KML TO SHAPEFILE(KML file)    ▷ Import polygons' boundaries
        for i = number of obstacles do Importing coordinate of vertices to a matrix
        end for
    end function
    function CONVEXHULL(Polygons' boundary)    ▷ Concave to convex conversion
        for i = number of obstacles do check polygons' shape
            if polygon is concave then convert it to the convex one
            else leave it
            end if
        end for
    end function
    function CREATEVORDIAGRAM(convex obstacles)    ▷ create voronoi diagram
        while not reached to Voronoi diagram do adjust tracking point position
        end while
        while on Voronoi diagram do continue on Voronoi diagram to discover uncovered area
        end while
    end function
    function PROGRAPH(Voronoi diagram)    ▷ Create adjacency graph
        P = Adjacency matrix
        for i = number of nodes do
            if the node is connected to the other nodes through the visibility edge then
                $P \leftarrow 1$
            else
                $P \leftarrow 0$
            end if
        end for
    end function
    function DIJKSTRA(Adjacency graph, Weigh matrix)    ▷ Finding the shortest path
    end function
end function
```

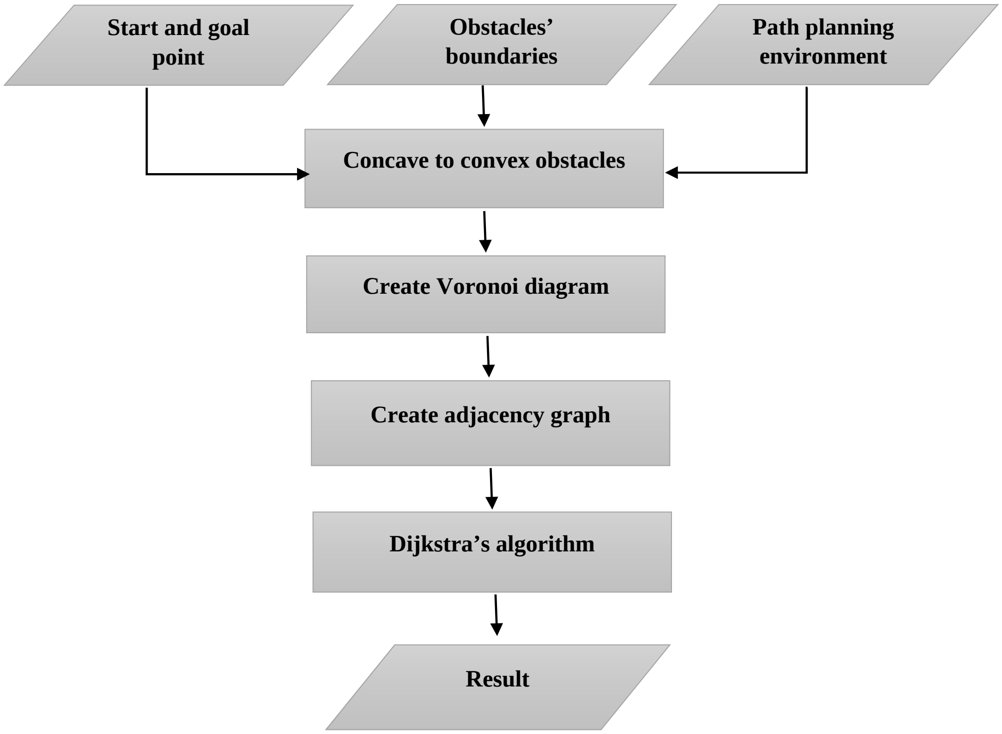


Figure 3.4: Voronoi diagram workflow

To import data, such as obstacle boundaries, start and end point coordinates and environment boundary, a function was implemented to process this step. This function read all data stored in KML format from the last step of "selecting unnavigable ice areas" methodology section.

- **Concave to Convex Conversion**
  To reduce the complexity of concave polygons, a function was implemented to convert concave polygons to convex ones. The results from this step prevent the creation of Voronoi trajectories in the areas where it is not necessary.

- **Creating the Voronoi Diagram**
  Based on what was mentioned in Chapter 2, a function which has various subfunctions was implemented to produce the Voronoi trajectory. This function consists of the following sub-funtions: 1) reached to the Voronoi diagram 2) navigate through the Voronoi trajectory 3) correction component 4) loop handling component 5) backtracking component

- **Creating the Adjacency Graph**
  After all available paths were generated in the previous step, a function was implemented to create an adjacency matrix. This matrix indicates which vertices of the Voronoi diagram are adjacent to which other vertices. Actually, this function creates an N by N matrix where N indicates the number of nodes. If a node is adjacent to another node, the function defines the adjacency by the value "1"; if there is no adjacency the function defines the non-adjacency by the value "0" value.

- **Finding the Shortest Path**
  When the Voronoi diagram is generated, the shortest path should be selected among all available paths between the start and end points. As mentioned before, Dijkstra's algorithm was applied for the shortest path optimization. This function finds the shortest path between start and end points based on the adjacency matrix and distances between adjacent nodes beginning from the start node to the end point.

Visibility Graph

In the Visibility diagram the following steps were implemented in the Matlab environment to generate all candidate paths from a start point to an end point and define the shortest one based on Dijkstra's algorithm. As Fig. 3.5 describes, the visibility graph function consists of different sub-functions, such as importing data into the matlab workspace, creating configuration spaces, creating visibility graphs, creating adjacency graphs and Dijkstra's algorithm. Algorithm 3 shows what functions are called within MATLAB environment, where these functions are customized based on the open source codes from MAthWorks file exchange.

The proposed method was implemented in the Matlab environment to process the Visibility graph depicted in Fig. 3.5.

- **Importing Data**
  In a manner similar to the Voronoi diagram, data, such as obstacle boundaries, start and end point coordinates, and environment boundary, should be imported. This function reads all data stored in KML format from the last step in "selecting unnavigable ice areas" in the Methodology part.

- **Creating the Configuration Space**
  In this research, the ship is assumed as a point. However, since a ship has dimensions, the size of the obstacles should be extended based on the ship size in order to accommodate the ship. As a result, the configuration space of the obstacles was applied. However, in this research no such changes to the obstacle were made.

Algorithm 3 Visibility graph for ice obstacles

```
function Visibility graph()
    function KML to Shapefile(KML file)                          ▷ Import polygons' boundary
        for i = number of obstacles do import coordinate of vertices to a matrix
        end for
    end function
    function convexHull(Ploygons' boundary)                      ▷ Concave to convex conversion
        for i = number of obstacles do check polygons' shape
            if polygon is concave then convert it to the convex one
            else leave it
            end if
        end for
    end function
    function configSpace(Convex obstacles)                       ▷ Create configuration space
        for i = number of obstacles do create configuration space for polygons
        end for
    end function
    function createVisGraph(configuration space)                 ▷ Create visibility graph
        for i = number of obstacles do
            for j = number of vertices do connect each vertex to the other nodes
            end for
        end for
    end function
    function reducedVisGraph(Visibility edges)     ▷ Remove redundant visibility edges
        for i = number of visibility edges do
            if visibility edge crosses inside the polygons then remove it from the list
            else leave it
            end if
        end for
    end function
    function ProGraph(Reduced visibility edges)                  ▷ Create adjacency graph
        P = Adjacency matrix
        for i = number of nodes do
            if the node is connected to the other nodes through the visibility edge then
                $P \leftarrow 1$
            else
                $P \leftarrow 0$
            end if
        end for
    end function
    function Dijkstra(Adjacency graph, Weigh matrix)       ▷ Finding the shortets path
    end function
end function
```

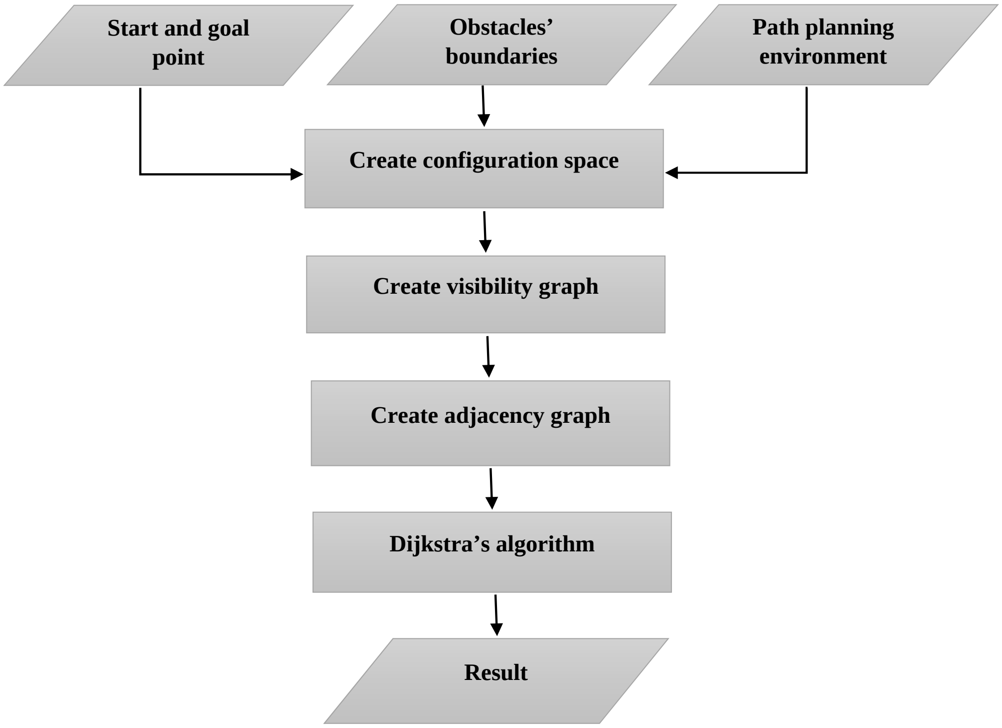


Figure 3.5: Visibility graph workflow

- **Constructing the Visibility Graph**
  Construction of the visibility graph and its reduced version was implemented based on a rotational sweep line algorithm. To build the visibility graph, a straight line called an edge would be connected from each vertex to the other vertices which are visible from each other. The vertices include the vertices of all obstacles and start and end points.

- **Creating the Adjacency Graph**
  In this step, an adjacency matrix was created. This matrix checks all nodes one by one and indicates which vertices of the visibility graph are adjacent to which other vertices and stores the adjacency information into a matrix.

- **Finding the Shortest Path**
  After constructing the visibility graph for all nodes, the shortest path should be selected based on Dijkstra's algorithm. This algorithm checks the distance from start to end points through all available routes and selects the shortest one.

Visibility-Voronoi

Visibility-Voronoi is another approach for generating paths in the collision-free environment. As Fig. 3.6 describes, the Visibility-Voronoi diagram function consists of different sub-functions, such as importing data into the Matlab workspace, buffering the obstacles, creating visibility graphs, creating adjacency graphs, and the Dijkstra algorithm.

The functions are employed within MATLAB environment for VV diagram represents in algorithm 4, where these functions are modified based on the open source codes from MAthWorks file exchange.

The proposed method was implemented in the Matlab environment to process the Visibility-Voronoi approach depicted in Fig. 3.6.

- **Import Data**
  Obstacle boundaries, start and end point coordinates, and environment boundary were imported by applying a developed function. This function reads all data stored in KML format from the last step in “selecting unnavigable ice areas” from the Methodology part.

- **Buffer obstacles**
  In order to have clearance from all obstacles, entire obstacles were buffered with proper distance. A function was developed to carry out this stage of the Visibility-Voronoi algorithm. The buffer distance may vary based on different circumstances such as ship type or ice type. However, in this research, a unique value was considered as the safety clearance.

- **Constructing the Visibility Graph**
  The Visibility graph function was implemented in a manner similar to constructing the visibility graph in the previous algorithm. To build visibility graphs, a straight line called an edge is connected from each vertex to the other vertices which are visible from each other. The vertices include the vertices of buffered obstacles and start and end points.

- **Creating the Adjacency Graph**
  In this function, an adjacency matrix is created. This matrix indicates which vertices of the Visibility-Voronoi diagram are adjacent to which other vertices. This function is the same as what was mentioned for the previous path planning algorithms.

- **Finding the Shortest Path**
  One of the goals of the INSS is finding the shortest path between all accessible routes between the start and end points. After constructing the Visibility-Voronoi diagram, Dijkstra’s algorithm can find the shortest path.

Algorithm 4 Visibility-Voronoi diagram for ice obstacles

```
function Visibility-Voronoi diagram()
    function KML to Shapefile(KML file)                          ▷ import polygons' boundary
        for i = number of obstacles do import coordinate of vertices to a matrix
        end for
    end function
    function Extend obstacles(Ploygons' boundary)                ▷ buffer the obstacles
        for i = number of obstacles do buffering the obstacles with defined buffer distance
        end for
    end function
    function createVisGraph(configuration space)                 ▷ create visibility graph
        for i = number of obstacles do
            for j = number of vertices do connect each vertex to the other nodes
            end for
        end for
    end function
    function reducedVisGraph(Visibility edges)       ▷ remove redundant visibility edges
        for i = number of visibility edges do
            if visibility edge crosses inside the polygons then remove it from the list
            else leave it
            end if
        end for
    end function
    function ProGraph(Reduced visibility edges)                  ▷ create adjacency graph
        P = Adjacency matrix
        for i = number of nodes do
            if the node is connected to the other nodes through the visibility edge then
                $P \leftarrow 1$
            else
                $P \leftarrow 0$
            end if
        end for
    end function
    function Dijkstra(Adjacency graph, Weigh matrix)        ▷ Finding the shortest path
    end function
end function
```

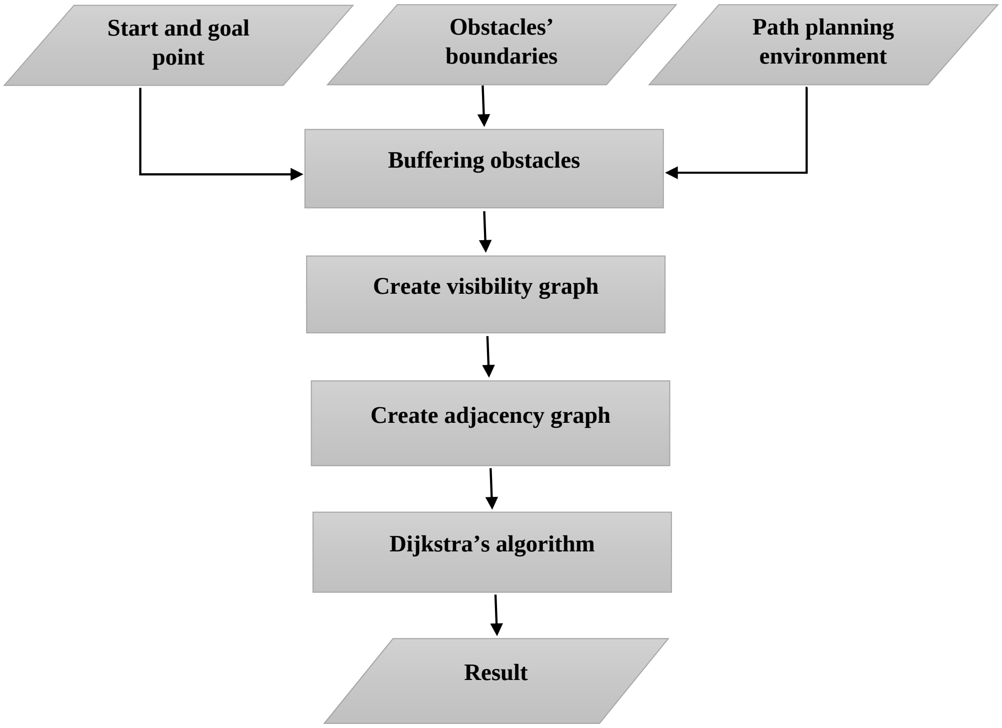


Figure 3.6: Visibility-Voronoi diagram workflow

All of the above path planning algorithms were implemented and tested on a desktop computer with the following specifications:

- **CPU** = Intel(R) Xeon(R) CPU E5-1650 3.2GHZ
- **RAM** = 16.0 GB.
- **Graphic Card** = NVIDIA Quadro 600 (4842 MB)
- **Operating system** = Windows 7 Enterprise (64-bit)

3.3 Dataset and Study Area

For the purpose of ice navigation in the Arctic, daily or hourly ice charts are necessary due to the dynamic texture of sea ice. However, the daily ice charts available in the Canadian Ice Service Archive (CISA) [15] are in image format which is not applicable in INSS. On the

---

[15] `http://iceweb1.cis.ec.gc.ca/Archive20/?lang=en`

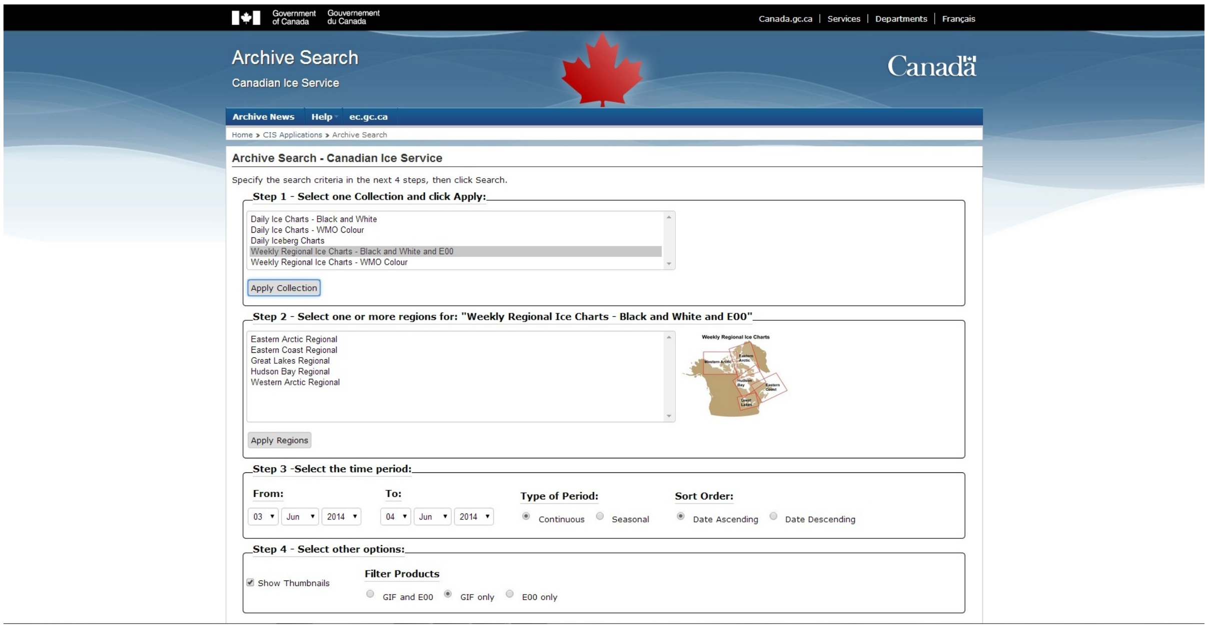
other hand, weekly regional ice charts available in CISA are in vector format which is the most compatible format with the designed INSS. As a result, the weekly ice charts from CISA were selected for the study. Fig. 3.7 demonstrates the screen shot from the CISA website where ice chart data may be downloaded.

The weekly ice chart data available on CISA is in "*.e00" format which is an "ArcInfo interchange file". The "ArcInfo Interchange file" is an exchange format used to transfer coverage, grid to TIN, and associated Info tables between different machines not linked with any type of file sharing network (ESRI, 2012). All types of ESRI software can import and convert this coverage format file into the shapefile format (*.shp), which is the proper format for further analysis. Shapefiles have some advantages in comparison with other data formats, such as "*.e00" files. They are faster to draw, edit, read, and write, and also they need less disk space to be stored (ESRI, 1998). The weekly ice chart in CISA covers five regions in the Canadian Arctic sea; the Western Arctic, Eastern Arctic, Hudson Bay, East Coast and Great Lake regions. Fig. 3.8 illustrates these regions covered by the weekly ice charts.

Since the weekly ice chart data covers the coastal area in some regions, a complete map containing all Canadian water bodies is required. The "Major Water Region" and the "Major Coast Region" shape files containing ocean boundaries and coast boundaries were obtained from the GeoScholar portal. The water body map can help us to make sure which parts of the weekly ice charts are within the oceans' boundaries.

3.4 Chapter Summary

In this chapter, the process of selecting proper software and programming language was mentioned (refer to Section 3.1). In addition, the proposed algorithm for determining unnavigable ice areas, calculating IN and three path planning algorithms are described (refer to Section 3.2). Afterward, the detailed description of datasets and study area was discussed. In Chapter 4, the proposed methodology of the INSS was conducted on both synthetic and real data in order to evaluate the methodology and the results.



Archive Search - Canadian Ice Service

Specify the search criteria in the next 4 steps, then click Search.

Step 1 - Select one Collection and click Apply:

Daily Ice Charts - Black and White
Daily Ice Charts - WMO Colour
Daily Iceberg Charts
Weekly Regional Ice Charts - Black and White and E00
Weekly Regional Ice Charts - WMO Colour

Step 2 - Select one or more regions for: "Weekly Regional Ice Charts - Black and White and E00"

Eastern Arctic Regional
Eastern Coast Regional
Great Lakes Regional
Hudson Bay Regional
Western Arctic Regional

Step 3 -Select the time period:

From: 03 Jun 2014
To: 04 Jun 2014
Type of Period: ● Continuous ○ Seasonal
Sort Order: ● Date Ascending ○ Date Descending

Step 4 - Select other options:

☑ Show Thumbnails
Filter Products: ○ GIF and E00 ● GIF only ○ E00 only

Figure 3.7: Canadian Ice Service Archive (CISA) website[9]

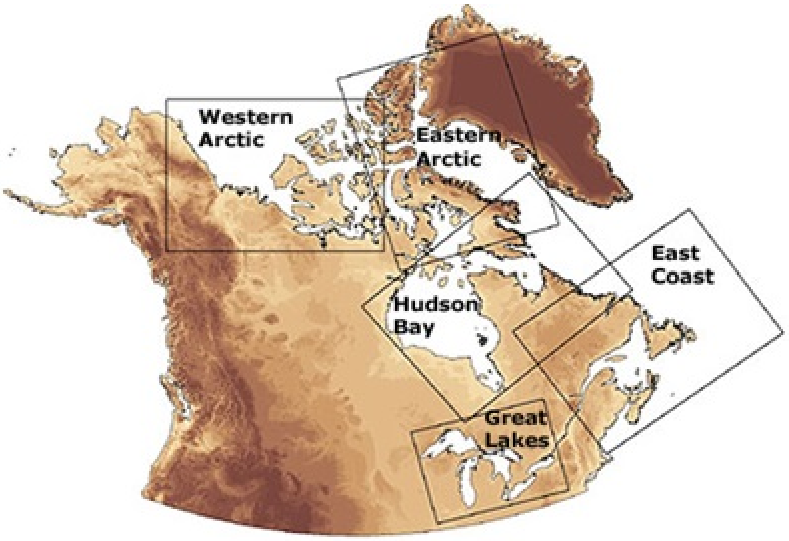


Figure 3.8: Ice data coverage in the Canadian Arctic area[10]

---

[16] http://iceweb1.cis.ec.gc.ca/Archive20/?lang=en

[17] http://ice-glaces.ec.gc.ca/content_contenu/images/regions.jpg

Chapter 4

Results and Discussions

In this chapter, the experimental results along with the outcomes from each step are presented. This chapter is divided into three main parts. Section 4.1 provides the preliminary results, illustrating the path planning results on synthetic ice data. The second part is about applying the ice navigation support system on real ice datasets (refer to Section 4.2). In Section 4.3, the outcomes from the ice navigation support system will be presented and discussed.

4.1 Synthetic Data

To determine whether the path planning algorithms were working properly for the ice data, eleven synthetic obstacles were created in a 20 by 20 kilometre environment. Both convex and concave obstacles were scattered within the environment boundary. First, the Voronoi diagram was applied to the obstacles and the result derived (see Fig. 4.1). Then, the final shortest path was defined based on Dijkstras algorithm.

As one can see from the result of path planning, the Voronoi trajectory has maximum clearance from all the obstacles. However, at some points it has sharp or unnecessary turns, which decreases the smoothness of the route. The total time to generate the Voronoi diagram and find the shortest path was 2089 seconds (about 35 minutes), and the total length of the shortest path indicated by green coloured lines was about 31 kilometres.

To generate a faster and smoother trajectory, a visibility graph was applied to the synthetic obstacles and the result derived (see Fig. 4.2). Since the visibility graph data is in general tangent to the obstacles, the paths generated usually contain semi-free configurations (the ship touches the exterior part of the obstacles); therefore, it does not have proper clearance from the obstacles, which is not acceptable for ice navigation.

Table 4.1: Comparison of CPU time and shortest path distance on the synthetical data

	Path Planning Model	**CPU Time (Seconds)**	**Shortest path distance (Kilometres)**
	Synthetic Data		
	Voronoi Diagram	2089	31
	Visibility Graph	13	28
	Visibility-Voronoi Diagram	100	28

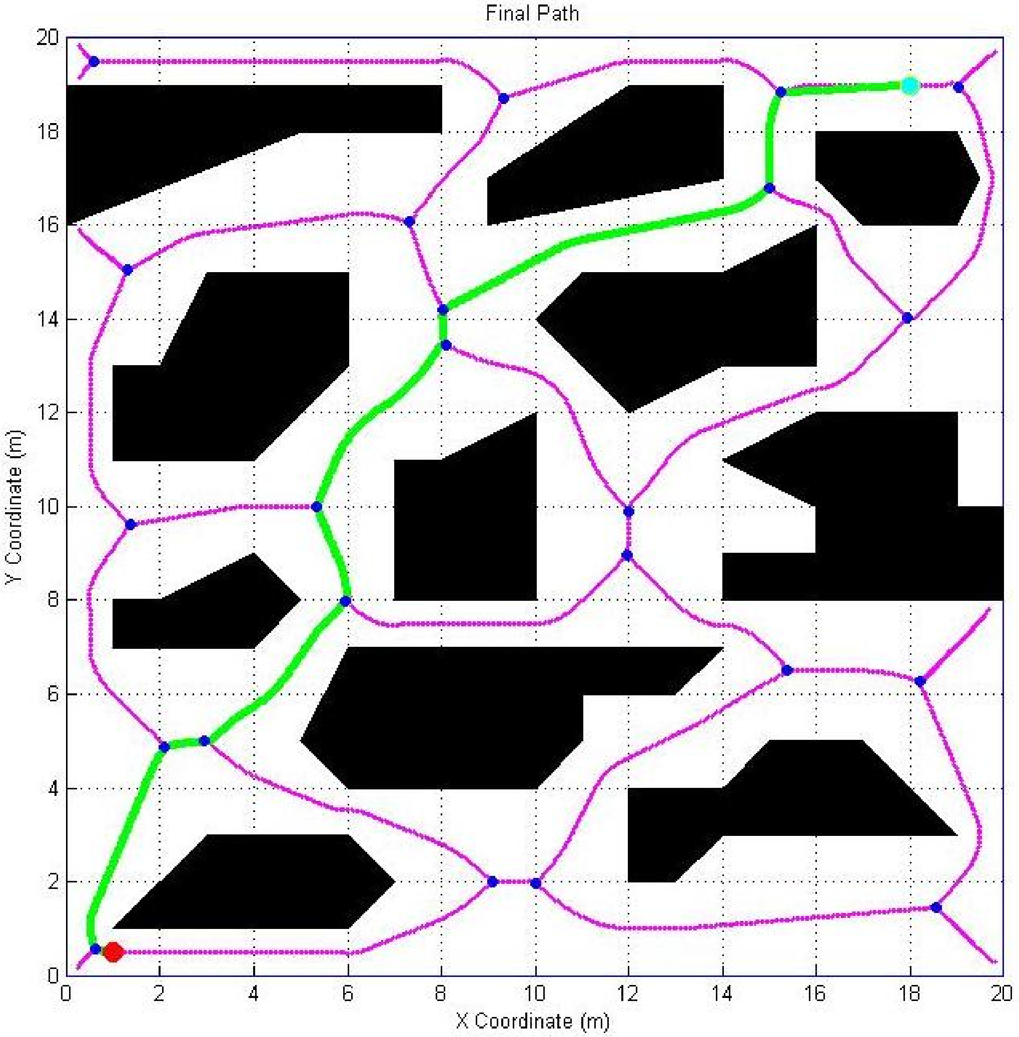


Figure 4.1: Voronoi diagram result on synthetic data

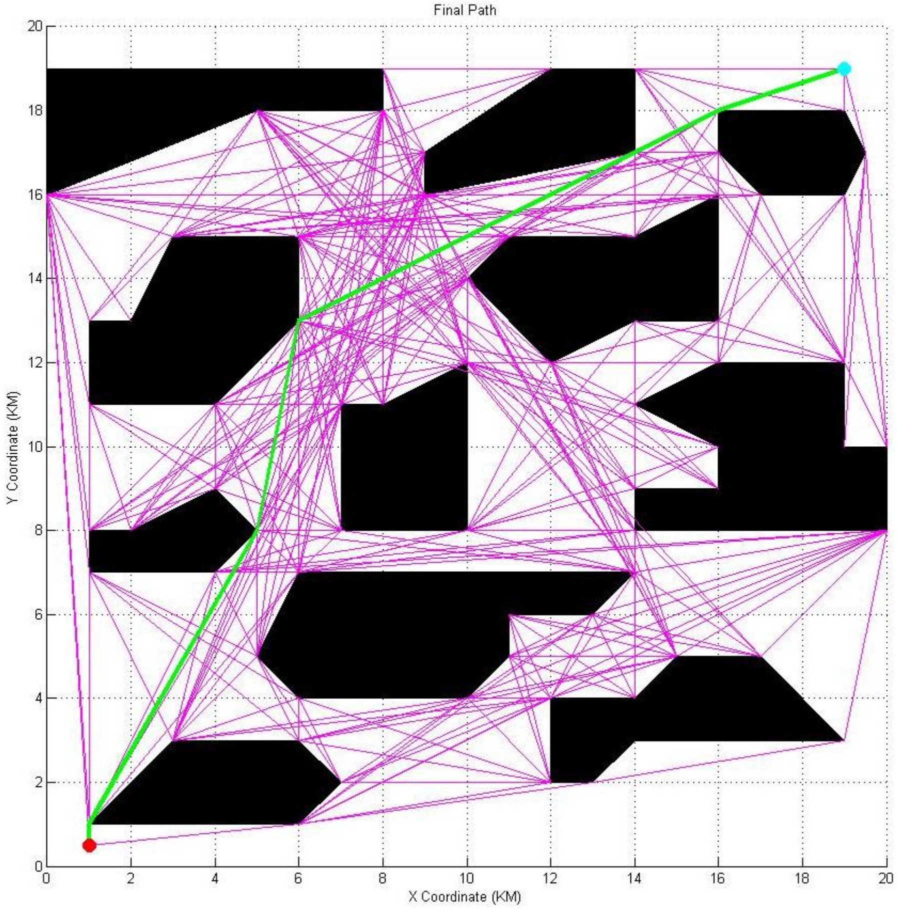


Figure 4.2: Visibility graph result on synthetic data

The total processing time for the visibility graph is 13 seconds and the total length for the shortest path is 28 kilometres. To have the advantages of both algorithms, they were combined together as the Visibility-Voronoi (VV) Diagram. In this algorithm, all the obstacles were buffered with 300 metres as their clearance distance, and then the visibility graph was applied to the buffered obstacles. Fig. 4.3 illustrates the outcome of the VV algorithm which is both smooth and collision free. The total processing time for the visibility graph was 114 seconds and the total length for the shortest path was 28 kilometres. Table 4.1 summarizes total distance and CPU time for each method on the synthetic data.

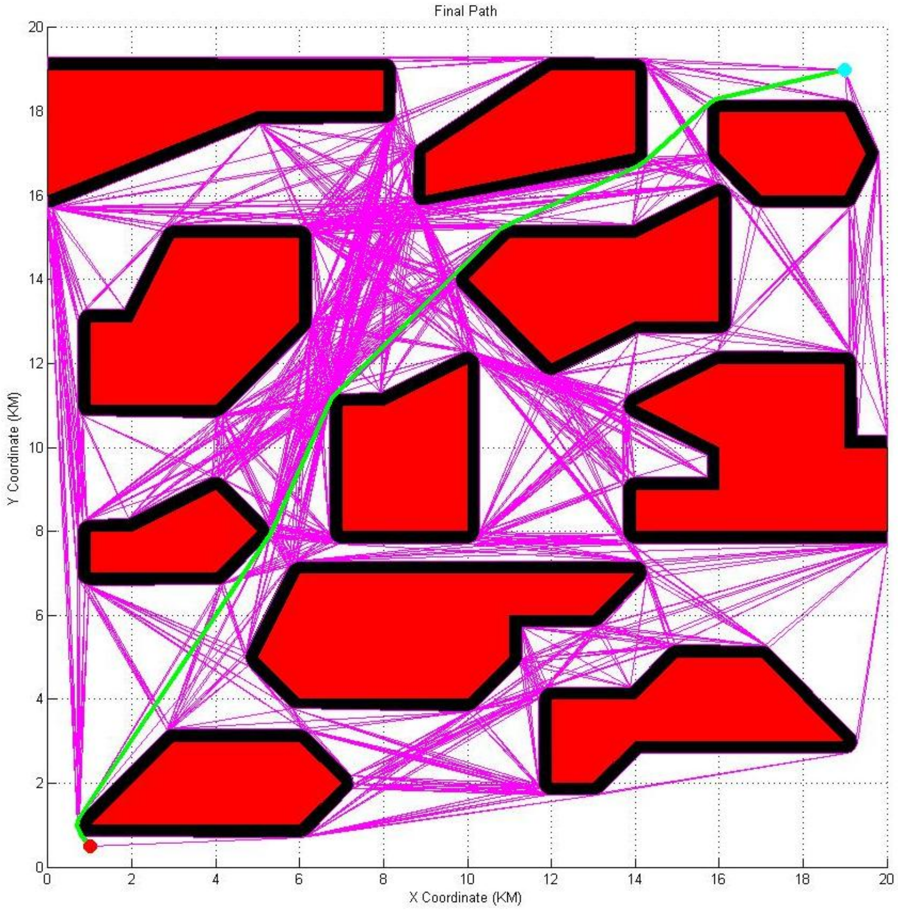


Figure 4.3: Visibility-Voronoi diagram result on the synthetic data

4.2 Real Data

After validating the path planning algorithms on synthetic obstacles, the INSS was tested on real datasets. The two datasets were selected from two different years and seasons to evaluate the support system (INSS). The first test was performed on ice data from September 2011. The second test was done on ice data from February 2014. The outcomes from each test are presented and compared in terms of total distance and CPU time.

4.2.1 Dataset and Study Area

The real ice data for the second week of September 2011 and February 2013 were obtained from the CISA website. Fig. 4.4 and Fig 4.5 demonstrate all the ice coverage data for both datasets. The red-coloured region represents the Western Arctic, the pink-coloured region represents the Eastern Arctic, the orange region represents the Hudson Bay area, and the purple region represents the East Coast. The beige region represents the coast, and the blue region represents the ocean data. As Fig. 4.5 illustrates, there is another part covered by ice during the winter season which was free of ice during September 2011. This part is the Great Lakes, marked by a green polygon. The Coordinate system of all datasets are WGS 84.

4.2.2 Applying Selection of Unnavigable Ice

In this research, the ship type for all testing was considered to be "CAC 3" which is the strongest according to the ice navigation manual (INCW, 2012). After calculating the IN, the algorithm automatically selected the ice with zero and negative values of IN (Unnavigable polygons) and clipped them with the ocean map. Finally, the clipped data boundaries were simplified and stored as a new feature class. Fig. 4.6 and Fig 4.7 represent the final output from the ice selection algorithm.

4.2.3 Selecting an Environment for the Path Planning

To select a proper environment for the path planning purposes, an area within the Canadian Arctic area was selected as an environment boundary. The environment should consist of a start point, an end point, and all the obstacles which have the potential of collision for the ship according to the start and end point. Fig. 4.8 represents the environment of path planning for the first experiment which has seven obstacles. This area is surrounded by an area of 26744.65 square kilometres. Also, Fig 4.9 illustrates the selected environment of path planning for the second experiment which has eleven obstacles and scattered in an area bigger than the first experiment. The second environment has an area of 248871.24 square kilometres. As Fig. 4.8 and Fig 4.9 demonstrates, the red part regions illustrate unnavigable areas. The blue part regions shows both navigable areas in the Canadian Arctic area and ocean's boundaries. Also, the beige part is the cost boundaries.

4.2.4 Implementing Path Planning Algorithms

In this study, due to the lack of information of ice movements, such as velocity of ice data, the path planning algorithms were considered for the static obstacles only. Also, the figures

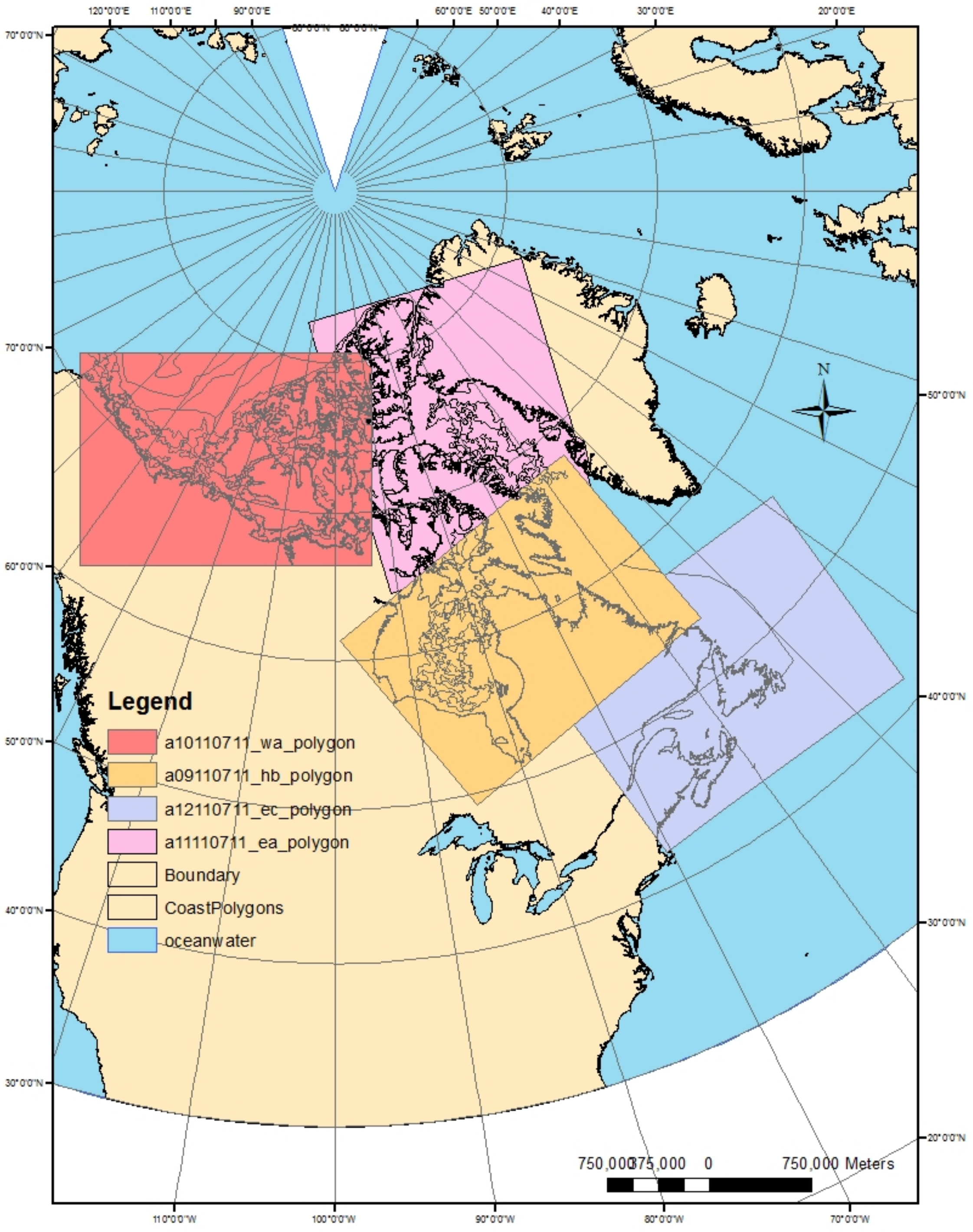


Figure 4.4: Weekly ice data coverage for the second week of September 2011

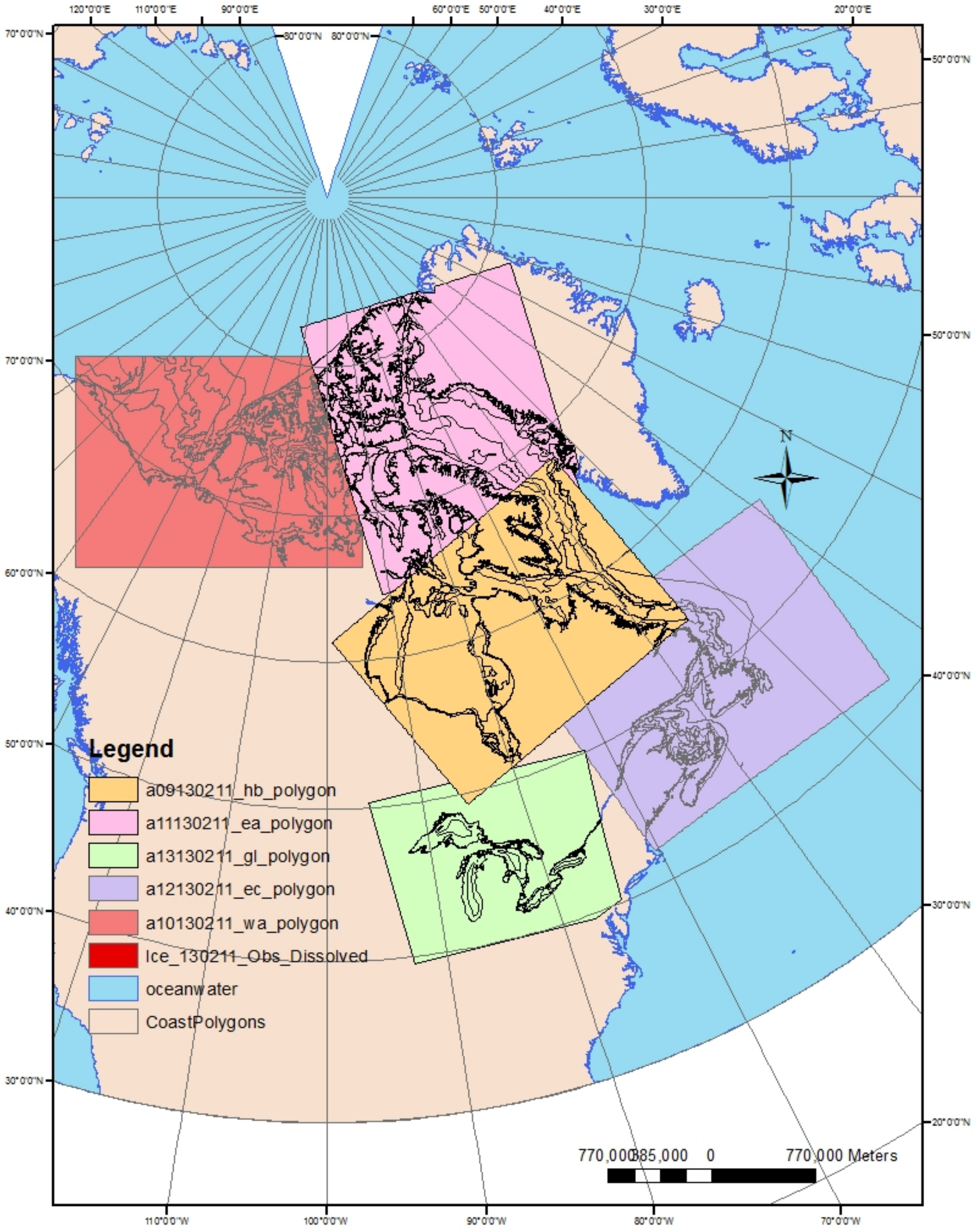


Figure 4.5: Weekly ice data coverage for the second week of February 2013

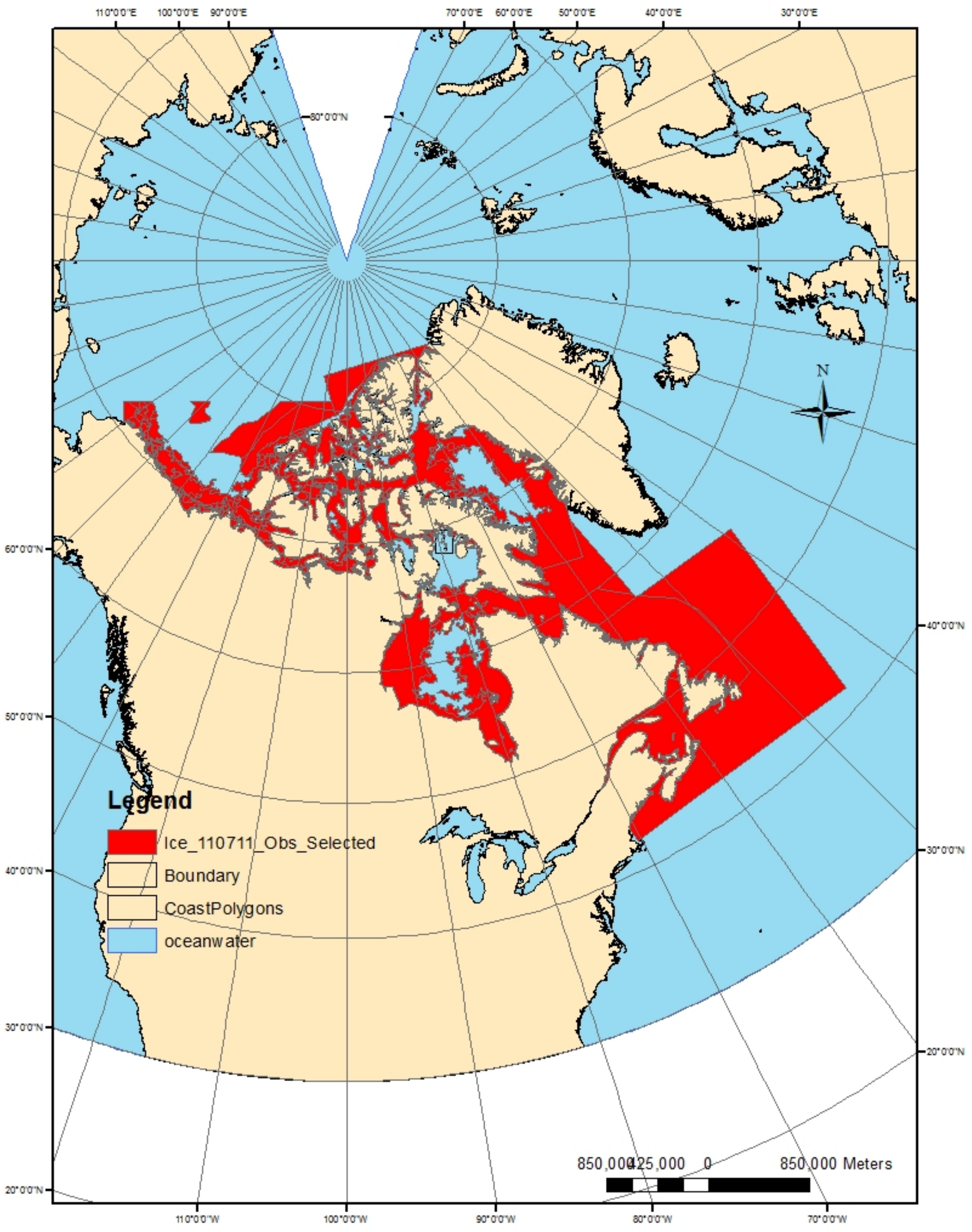


Figure 4.6: Ice selection result for the ice data of the second week of September 2011

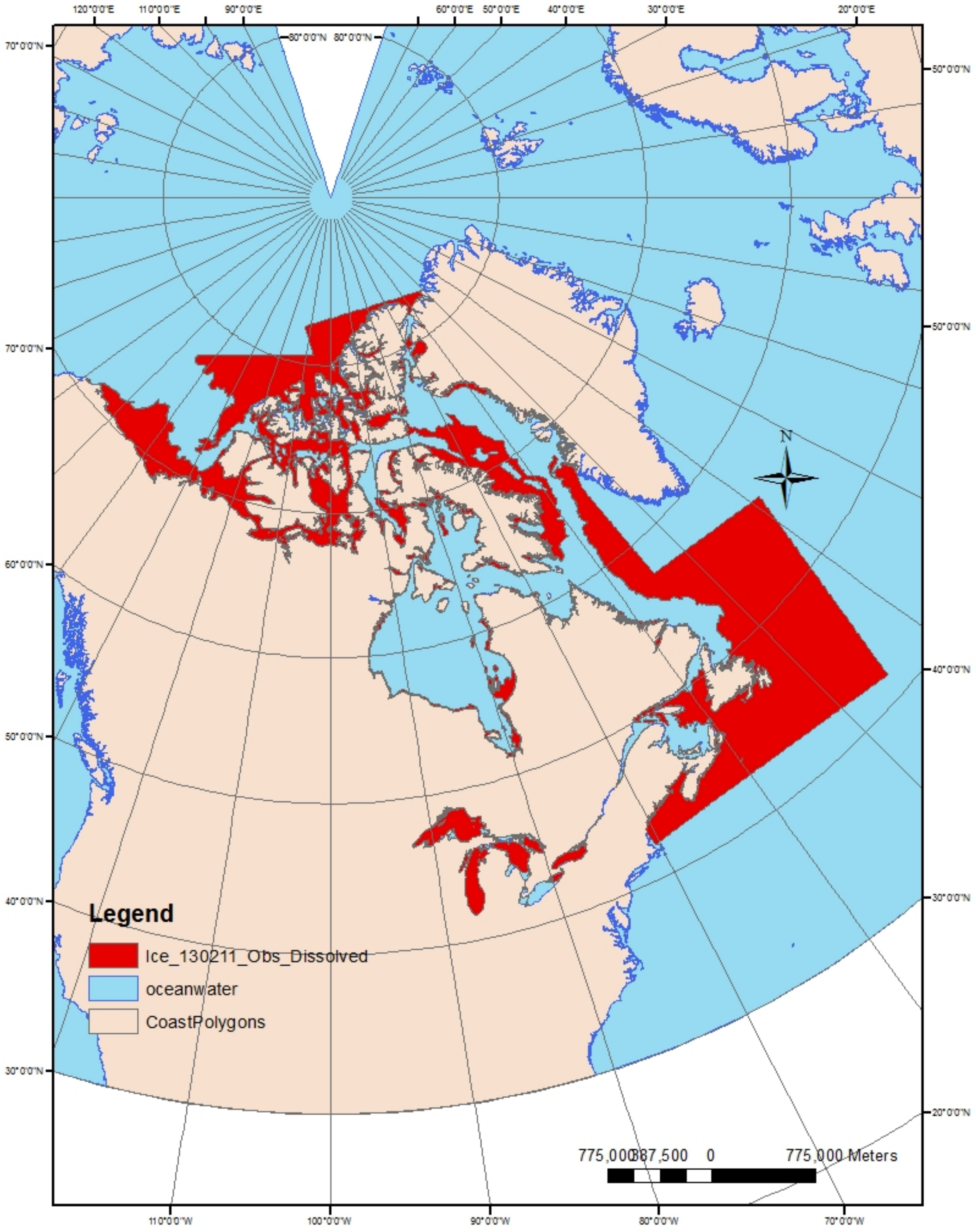


Figure 4.7: Ice selection result for the ice data of the second week of February 2013

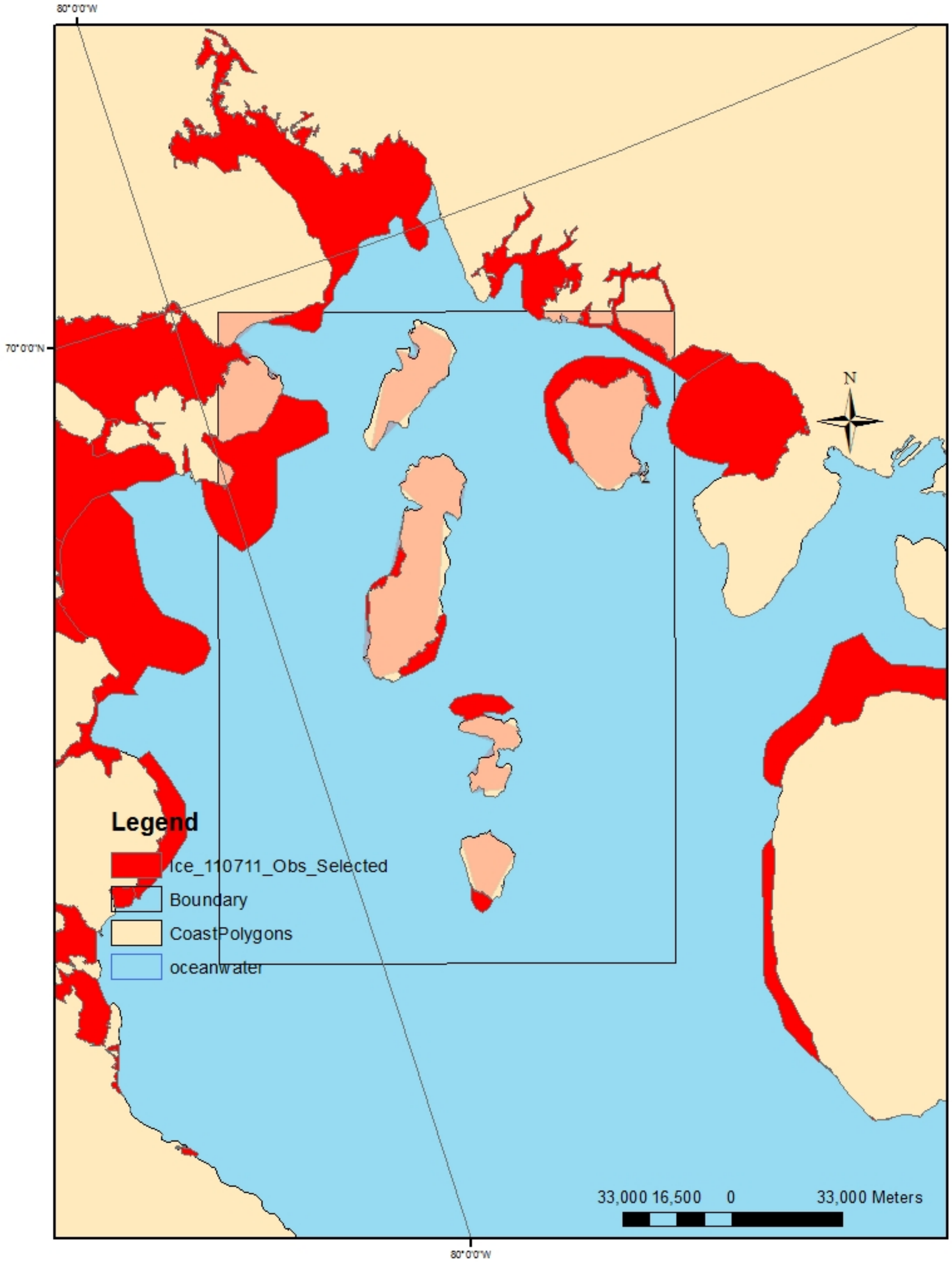


Figure 4.8: Study area (environment) for the path planning of the second week of September 2011

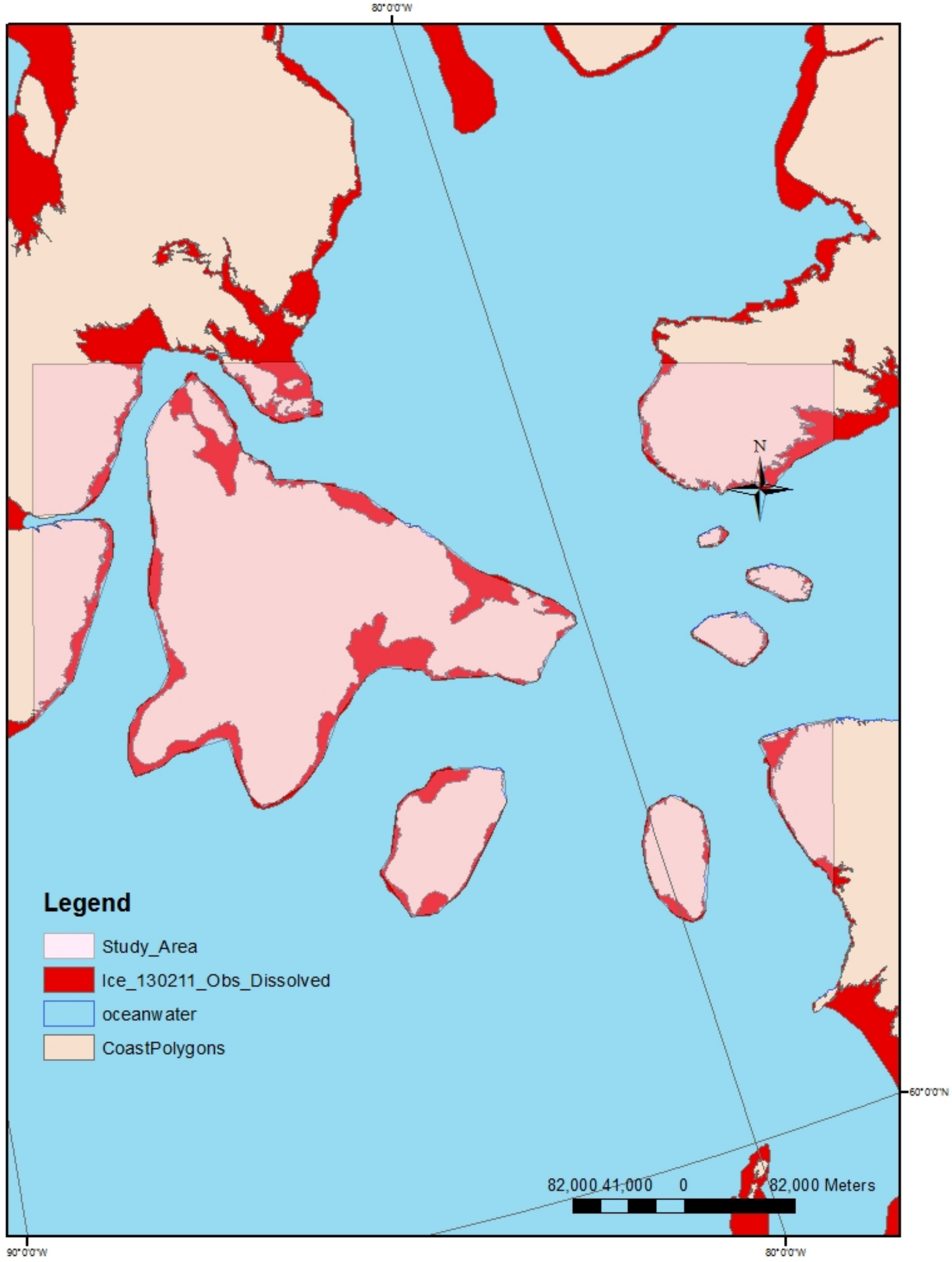


Figure 4.9: Study area (environment) for the path planning of the second week of February 2013

represented in this section are in different scales since the area of two study path planning environment are completely different.

Voronoi Diagram

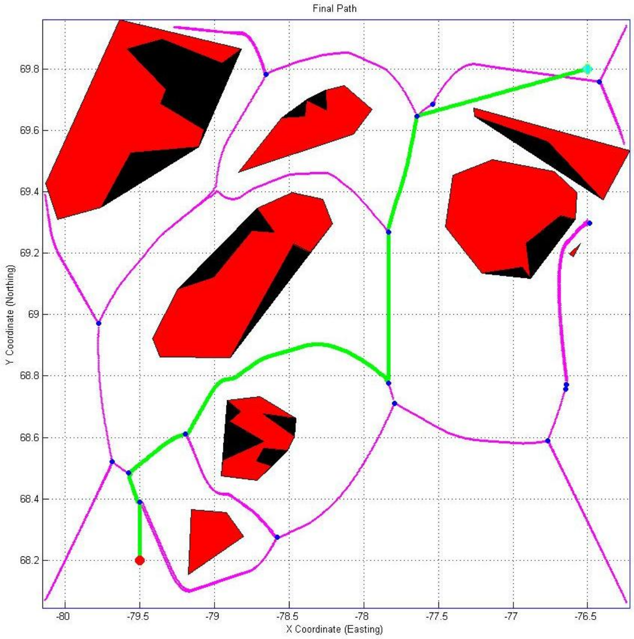
Fig 4.10 represents the results of the Voronoi diagram algorithm on both selected environments from the two experimental datasets. As Fig 4.10 shows, the red patches are the real forms of the obstacles, the black patches are the concave forms of the obstacles,the red point is the start point, and the cyan point is the end point. In addition, the purple curves are the Voronoi trajectories, the green curve shows the shortest path, and the blue points are nodes or turning points. As Fig 4.10 illustrates, the generated path has maximum clearance from all the obstacles; however it is not smooth in some turning points. Although the outputs from the Voronoi diagram are acceptable, there is one minor misconnection happens on right side of the Fig 4.10(a) during the generating of Voronoi diagram due to the complexity of algorithm and shape of obstacles.

Visibility Graph

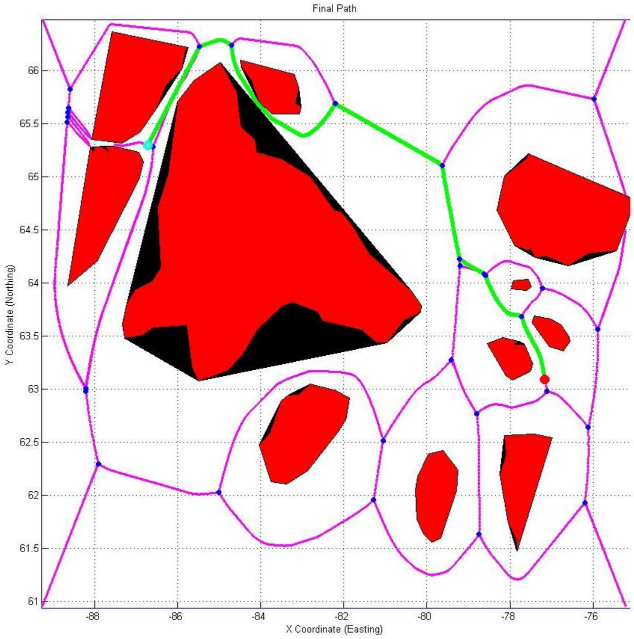
In this part, the visibility graph algorithm is tested on the selected environment. Fig. 4.11 illustrates the final result from the visibility graph algorithm for the two selected environments. The red patches are the real forms of the obstacles, the black patches are the concave forms of the obstacles, the red point is the start point, and the cyan point is the end point. In addition, the purple curves are the Voronoi diagram, the green curve shows the shortest path, and the blue points are nodes or turning points. Fig. 4.11(a) illustrates the first path planning environment covering smaller area than the second environment and has eight obstacles. Totally, 265 visibility edges were generated which has 57 nodes including start and end point. However, in Fig. 4.11(b) illustrating the second experiment of path planning environment which has eleven obstacles, the total generated visibility edges are 630 which has 174 nodes including the start and end point.

Visibility-Voronoi Diagram

Generating a safe and smooth path is the most important concern for ice navigation. Fig. 4.12 illustrates the result of the Visibility-Voronoi algorithm. The red part represents the real ice obstacles, and the black part is the clearance area from the ice, which is not safe to navigate. In this algorithm, first, the obstacles were buffered by 5000 metres (5 kilometres), which is assumed as the safety clearance distance from all the obstacles. Then, the visibility graph was applied on the buffered obstacles. Finally, Dijkstras algorithm found the path with the

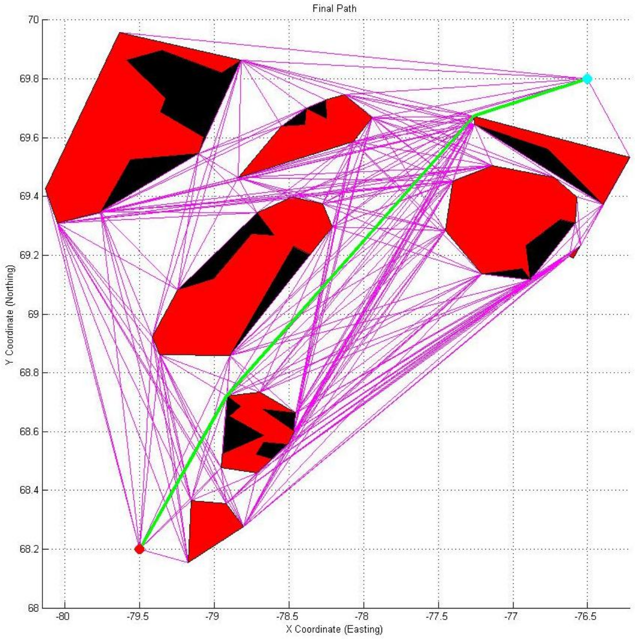


(a) Voronoi diagram's result on the selected environment of the second week of September 2011

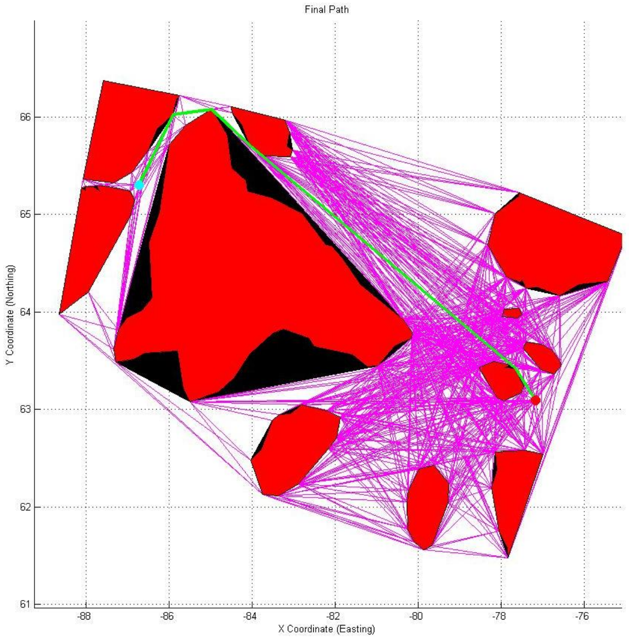


(b) Voronoi diagram's result on the selected environment of the second week of February 2013

Figure 4.10: Voronoi Diagram



(a) Visibility graph's result on the selected environment of the second week of September 2011



(b) Visibility graph's result on the selected environment of the second week of February 2013

Figure 4.11: Visibility graph

shortest distance between start and goal points in the environment. As one can see in Fig. 4.12, the final shortest path is smooth and is within the safe clearance distance from all obstacles. Fig. 4.12(a) represents the first environment for the path planning. Totally, 1274 visibility edges were generated over the buffered obstacles which has 206 nodes including start and point. However, in Fig. 4.12(b) illustrating the second experiment path planning environment, the total generated visibility edges over the buffered obstacles are 3853 edges which has 279 nodes including the start and end point.

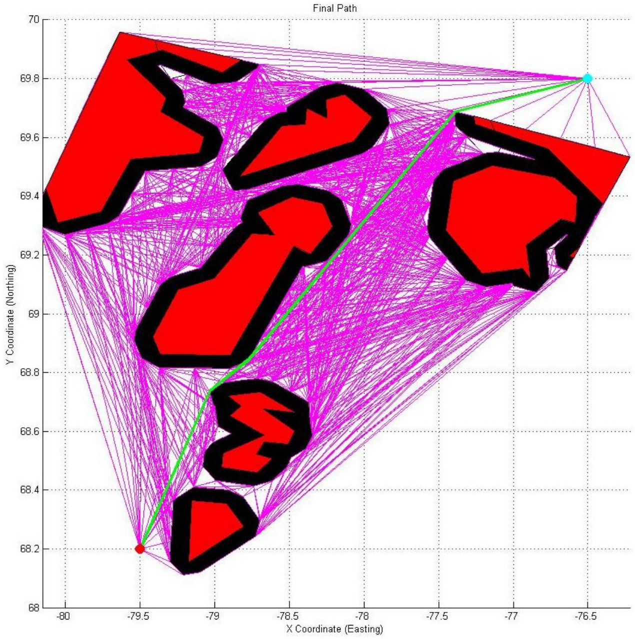
Different buffer sizes were tested on the both data experiments to evaluate the effect of buffer size on the computation time. The buffer size range from 500 metre to 6000 metre with 500 metre increment (See Fig. 4.13). Fig. 4.13(a) shows the time computation vs. buffer size for the first data experiment. As Fig. 4.13(a) illustrates, while the buffer size increases, the computation time decreases, generally. Fig. 4.13(b) illustrates the time computation vs. buffer size for the second experiment. As Fig. 4.13(b) illustrates, while the buffer distance increases, the time computation for the VV diagram decreases, as well. While changing the buffer distance didn't effect on time computation significantly, the same shortest path were generated, in all scenarios.

Table 4.2 summarizes the CPU time and the length of shortest distance in the three path planning algorithms.

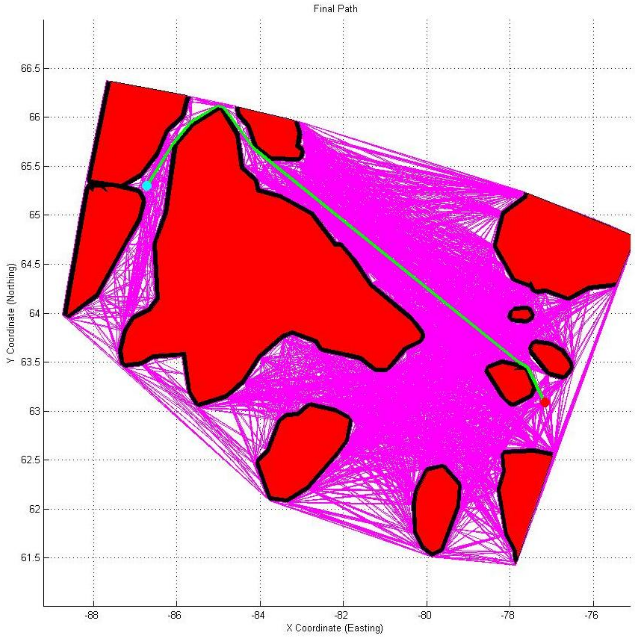
Table 4.2: Comparison of CPU time and shortest path distance in two experimental test

Path Planning Model	**CPU Time (Seconds)**	**Shortest path distance (Kilometers)**
First Experiment		
Voronoi Diagram	2220	420.40
Visibility Graph	14	342.20
Visibility-Voronoi Diagram	64.5	342.20
Second Experiment		
Voronoi Diagram	5545	852.50
Visibility Graph	60	792.10
Visibility-Voronoi Diagram	399	792.10

Table 4.2 represents CPU time and total distance for the three discussed shortest path planning algorithms for the two above-mentioned experiments. As one can see, because the second experiment's environment has a greater area and more obstacles than that of the first experiment, the CPU time and the total distance values in the second experiment are greater

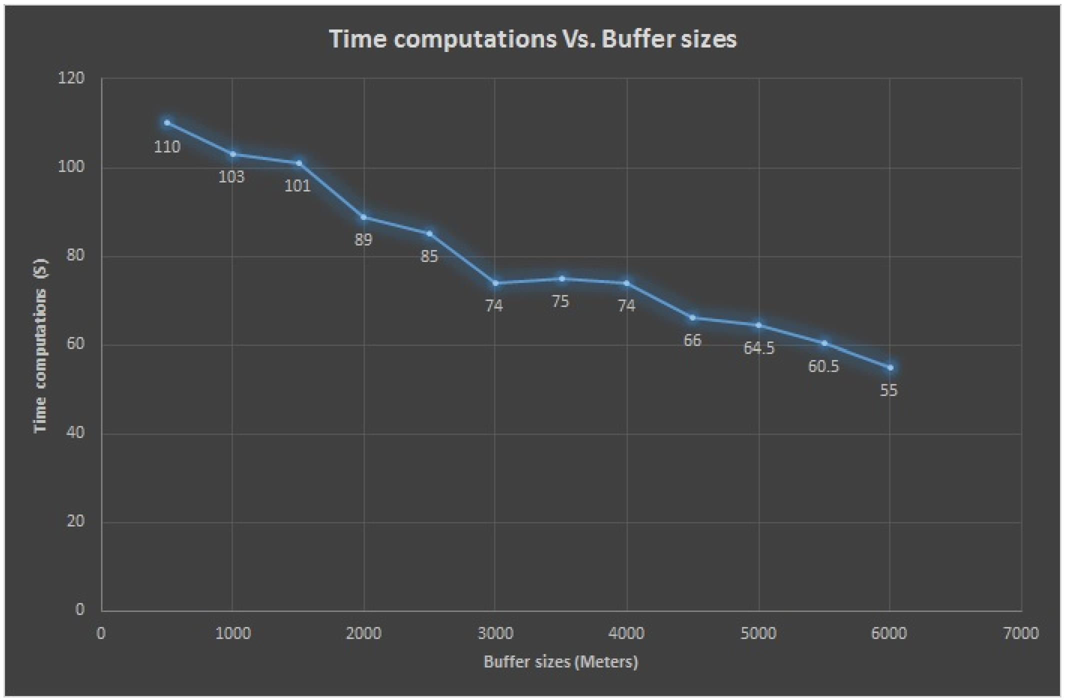


(a) Visibility-Voronoi Diagram's result on the selected environment of the second week of September 2011

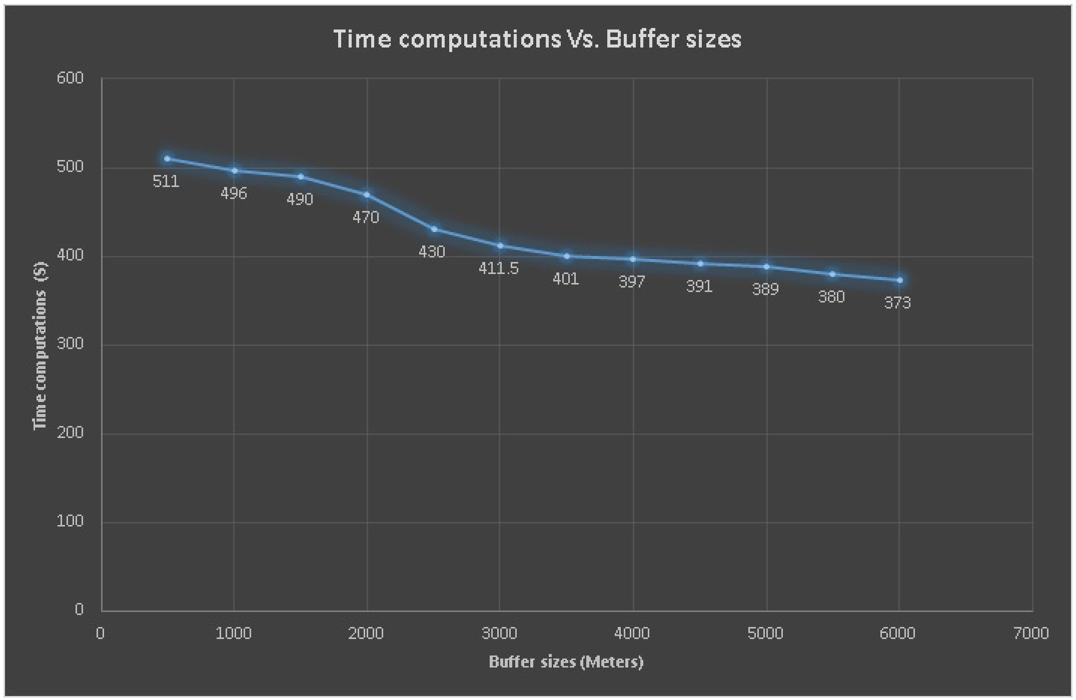


(b) Visibility-Voronoi Diagram's result on the selected environment of the second week of February 2013

Figure 4.12: Visibility-Voronoi Diagram



(a) Time computations vs. buffer sizes of the first data experiment



(b) Time computations vs. buffer sizes of the second data experiment

Figure 4.13: Time computations Vs. buffer sizes for VV diagram

than in the first one. Also, in each experiment, the Voronoi diagram has the maximum CPU time and maximum shortest path distance in comparison with the two other algorithms. The Visibility graph has the least CPU time between the three path planning algorithms. Although CPU time is greater for the Visibility-Voronoi diagram than for the Visibility graph, both the Visibility graph and the Visibility-Voronoi have the same shortest path distance.

4.3 Discussion

Due to the dynamic texture of sea ice, its location may vary during the ship's voyage. During the navigation through the ice, real time ice data is processed in the ice navigation support system's algorithm, and the trajectory which results from the INSS can be considered as an initial solution for the on-line and off-line phases along with the ARPA information. The updated trajectory would be passed along to the other part of the control process called "Adaptive Control". In this unit, additional navigation parameters including current, wind, tide, and hydrography maps are taken into account (Smierzchalski *et al.*, 2000).

The outcomes from these algorithms are interesting. The shortest paths found by the three algorithms demonstrate similar routes. However, each method has some advantages and disadvantages. The visibility graph outcome is smooth and fast to generate; nonetheless it is not a safe trajectory. The Voronoi diagram has the maximum clearance from the obstacles; nevertheless, it is very intensive to generate, and the generated path may contain sharp turns and may be remarkably longer than the feasible shortest path which share the same view as (Colm *et al.*, 1985). The Visibility-Voronoi which is both safe and fast is preferred for INSS path planning purposes if the safe clearance (C) is determined.

To balance the factor of safety as well as the traveling time, the Visibility-Voronoi deems to be a close-to-ideal approach for ice navigation. It is crucial to mention that the final path from any of these methods is not the final decision for the ship's trajectory, since there are plenty of other factors that should be considered for choosing the final path. Actually, the outcomes from all these methods can help the ice navigator find the ship's proper route through the Canadian Arctic.

Table 4.3: Comparison of three different path planning models

	Path Plannin Model	**Safety**	**Smoothness**	**Complexity**
	Voronoi Diagram	✓		✓
	Visibility Graph		✓	
	Visibility-Voronoi Diagram	✓	✓	

Chapter 5

Conclusion and Future Works

5.1 Conclusion

As explained in Chapter 1, the demand for a support system for ice navigation in the Canadian Arctic has risen dramatically due to the decreasing amount of ice coverage in the area caused by global warming. This thesis aims to implement a support system to help ice navigators on-board polar ships to find not only the shortest but also the safest path through the ice in the Canadian Arctic. Though some existing research has addressed how ice navigators decide ice navigation manually, there is no automatic support system for ice navigation. This research attempts to fill the gap in this regard by opening a path for implementing an integrated support system to help ice navigators find both the shortest and the safest path in the Arctic area. The contributions of this thesis research include the following:

1. **Developing a code to determine unnavigable ice areas.**A Python code was developed based on the IN concept to determine unnavigable ice obstacles. In this step the raw ice data was downloaded from the CIS archive database. The ice data includes information about the total ice concentration, partial concentrations, and stage of development within the attribute table used to calculate the ice numeral. Then, all ice obstacles with a negative ice numeral value were selected as unnavigable ice polygons. Finally, the ice obstacle polygons within the specific boundary called the Path Planning Environment were converted to the KML format which is the proper format for the next step, which is path planning.

2. **Developing a path planning system.** The KML file from the previous step, which includes the ice obstacle data, was imported to each path planning algorithm to find the shortest path. I used three different off-line path planning algorithms, the Visibility

Graph, Voronoi Diagram, and the Visibility-Voronoi Diagram. Although each algorithm provided the shortest path from the specified start point to the specified goal point, they provide different final results due to the different concepts they use. The advantages and disadvantages of each algorithm were discussed extensively in Chapter 4.

The results from all path planning algorithms can help the ice navigator on-board polar ships to make reasonable decisions about which path to take, or they can be used as the initial solution for the ARPA control system of polar ships during the “on-line” trajectory. In this control system, other navigation factors are taken into account to find the safest and shortest path between the start and goal points. Estimation of a collision free path for ships necessitates a challenging trade-off between necessary deviations from a given route and the safety issue of navigation (Kolendo *et al.*, 2011). Therefore, ice navigation is a multi-criteria path planning problem that considers both the safety and economic issues determining a ship’s trajectory.

5.2 Future Works

The research presented in this thesis provides a foundation for using the available historical ice data in the Canadian Arctic for polar ice path planning. The proposed approach has certain limitations that can be further investigated in future studies. Several essentials for future work are listed as follows:

Considering the dynamic texture of ice for path planning: In my research, I assumed that the ice obstacles were fixed in their location during the path planning; however, in reality, ice has a dynamic texture. The path planning used in this research is suitable for static obstacles. To consider dynamic obstacles, other path planning algorithms such as the Evolutionary Algorithm (EA) should be used. This optimization algorithm mimics the chromosome structure to find the best solution for dynamic environments.

Considering other criteria for route optimization: In this study, only distances between graph’s nodes were considered as the weigh matrix of Dijkstra’s algorithm since the other optimization variables were not accessible, such as risk of collision or time-dependent network. If these criteria are available, they can be combined into the weigh matrix of Dijkstra’s algorithm for more reliable route optimization.

Developing a web mapping prototype for representing the data and results: For instance, an additional decision support sub-system can be implemented to the ISIS (ref Section 2.5) for the ice navigation support system. A request from the ice navigator can be sent through the web to the ISIS, and the path planning request can be processed on the server side which is connected to the real time data repository to obtain the required ice data. Since the ice data should be updated continuously, having an integrated spatial information system to acquire and analyze the recent ice data and generate the safest path is critical.

References

Blum, H., *et al.*, 1967. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19 (5), 362–380.

Bowditch, N., Bowditch, J.I., and Office, U.S.H., 2002. The New American Practical Navigator. . New York: EM Blunt,[1802-18–], 453–479.

Canadian Ice Service Office - Environment Canada, 2005. *Manual of Standard Procedures for Observing and Reporting Ice Conditions(MSPORC)*. Meteorological Service of Canada.

Choset, H. and Burdick, J., 2000. Sensor-based exploration: The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19 (2), 96–125.

Choset, H., *et al.*, 2000. Sensor-based exploration: incremental construction of the hierarchical generalized Voronoi graph. *The International Journal of Robotics Research*, 19 (2), 126–148.

De Berg, M., *et al.*, 2000. *Computational geometry*. Springer.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1), 269–271.

Frederking, R., 1999. Adapting the Canadian Ice Regime System to Operational Ice Navigation.. *In*: *Proceedings 9th International Offshore and Polar Engineering Conference ISOPE'99*, Vol. 2, 659–664.

Gold, C., *et al.*, 2005. 3D geographic visualization: the Marine GIS. *Developments in Spatial Data Handling*. Springer, 17–28.

Guard, M.S.C.C., 2012. *Ice Navigation in Canadian Waters(INCW)*. Fisheries and Oceans Canada.

Hanna, E., 1998. How and why dose sea ice vary?. *Journal of Meteorology*, 23 (229), 153–158.

Howell, S.E. and Yackel, J., 2004. A vessel transit assessment of sea ice variability in the Western Arctic, 1969-2002: Implications for ship navigation. *Canadian Journal of Remote Sensing*, 30 (2), 205–215.

Khon, V.C. and Mokhov, I., 2010. Arctic climate changes and possible conditions of Arctic navigation in the 21st century. *Izvestiya, Atmospheric and Oceanic Physics*, 46 (1), 14–20.

Kolendo, P., Smierzchalski, R., and Jaworski, B., 2011. Experimental research on evolutionary path planning algorithm with fitness function scaling for collision scenarios. *In*: *9th International Symposium on Marine Navigation and Safety of Sea Transportation, TransNav.*

Koonar, A. and Scarlett, B., 2002. Integrated Spatial Information System. .

Lisowski, J., 1985. A Simulation Study of Various Approximate Models of Ships Dynamics in the Collision Avoidance Problem. *Foundation of Control Engineering*, 10 (2), 176–183.

Lisowski, J., 2007. The dynamic game models of safe navigation. *International Journal on Marine Navigation and Safety of Sea Transportation*, 1 (1), 11–18.

Masehian, E. and Naseri, A., 2010. Mobile Robot Online Motion Planning Using Generalized Voronoi Graphs. *Journal of Industrial Engineering*, 4 (5), 1–1.

Ó'Dúnlaing, C. and Yap, C.K., 1985. A retraction method for planning the motion of a disc. *Journal of Algorithms*, 6 (1), 104–111.

Smierzchalski, R. and Michalewicz, Z., 2000. Modeling of ship trajectory in collision situations by an evolutionary algorithm. *Evolutionary Computation, IEEE Transactions on*, 4 (3), 227–241.

Smierzchalski, R. and Michalewicz, Z., 2005. Path planning in dynamic environments. *Innovations in Robot Mobility and Control*. Springer, 135–153.

Somanathan, S., Flynn, P., and Szymanski, J., 2009. The northwest passage: a simulation. *Transportation Research Part A: Policy and Practice*, 43 (2), 127–135.

Statheros, T., Howells, G., and Maier, K.M., 2008. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation*, 61 (01), 129–142.

Wein, R., Van Den Berg, J.P., and Halperin, D., 2005. The visibility–voronoi complex and its applications. *In*: *Proceedings of the twenty-first annual symposium on Computational geometry*, 63–72.

Xiao, J., *et al.*, 1997. Adaptive evolutionary planner/navigator for mobile robots. *Evolutionary Computation, IEEE Transactions on*, 1 (1), 18–28.

Yap, C.K., 1987. 3 Algorithmic Motion Planning. 1, 95–143.

Glossary

INSS Automatic Ice Navigation Support System. 22, 41

AIRSS Arctic Ice Regime Shipping System. 1, 33, 34

ARPA Automatic Radar Plotting Aids. 22, 52, 55

ASPPR Arctic Shipping Pollution Prevention Regulations. 1

CIS Canadian Ice Service. 2, 9, 12, 54

CISA Canadian Ice Service Archive. ix, 25, 27, 34, 35, 41

COLREG International Regulations for Preventing Collision at Sea. 21

CPU central processing unit. viii, 39–41, 48, 52

ENVISAT Environmental Satellite. 9

ESRI Environmental Systems Research Institute. 34

GIS Geospatial Information System. iii

GVD Generalized Voronoi Diagram. 18

HGVD Hierarchical Generalized Voronoi Diagram. 18

IN Ice Numeral. 2, 11, 25–27, 34, 41, 54

INCW Ice Navigation in Canadian Arctic. 1, 5–9, 11, 12, 14–16, 41

KML Keyhole Markup Language. 24, 25, 28, 30–32, 54

KMZ Keyhole Markup Language files when compressed. 28

MSPORC Manula of Standard Procedures for Observing and Reporting Ice Conditions. 5

PIPS Polar Ice Prediction System. 12

SAR Synthetic aperture radar. 9

UTC Coordinated Universal Time. 9

VV Visibility-Voronoi Diagram. 19, 20, 39

WGS World Geodetic System. 41

XML Extensible Markup Language. 28