САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ**

**Селезнева Ольга Владимировна**

**Выпускная квалификационная работа аспиранта**

**Сопровождение малоразмерных маневрирующих судов в радиолокационной системе**

05.13.01

Системный анализ, управление и обработка информации

Научный руководитель,

доктор физ.-мат. наук,

профессор

Веремей Е. И.

Санкт-Петербург

2016

Оглавление

[Введение 3](#_Toc453664280)

[Глава 1. Характеристика алгоритма сопровождения целей в радиолокационной системе 6](#_Toc453664281)

[Предварительная обработка радиолокационного сигнала 6](#_Toc453664282)

[Выделение объектов и вычисление их параметров 7](#_Toc453664283)

[Прогнозирование положения отметки 8](#_Toc453664284)

[Выбор отметки в стробе сопровождения 10](#_Toc453664285)

[Сложные случаи сопровождения целей 11](#_Toc453664286)

[Глава 2. Выделение объектов и вычисление их параметров 14](#_Toc453664287)

[Общая информация 14](#_Toc453664288)

[Алгоритм выделения связных областей 16](#_Toc453664289)

[Усовершенствованный алгоритм выделения связных областей 18](#_Toc453664290)

[Вычисление параметров отметок 20](#_Toc453664291)

[Глава 3. Прогнозирование положения отметки, определение манёвра 24](#_Toc453664292)

[Прогнозирование положения отметки 24](#_Toc453664293)

[Сравнение алгоритмов с использованием линейной и квадратичной модели движения 27](#_Toc453664294)

[Строб сопровождения отметки 29](#_Toc453664295)

[Определение маневра 29](#_Toc453664296)

[Глава 4. Выбор отметки в области сопровождения 31](#_Toc453664297)

[Первый этап обработки 31](#_Toc453664298)

[Второй этап обработки 31](#_Toc453664299)

[Третий этап обработки 33](#_Toc453664300)

[Заключение 35](#_Toc453664301)

[Список литературы: 38](#_Toc453664302)

# Введение

Радиолокационная станция (РЛС) — система для обнаружения и определения параметров (положения, размеров, скорости) морских, воздушных, наземных объектов. Возможность обнаружения с помощью РЛС надводных объектов позволяет использовать её не только как средство обеспечения навигационной безопасности, но и как средство расхождения с другими объектами.

Проектирование алгоритма разрешения сложных ситуаций движения судов используя данные, полученные от РЛС, оказалось нетривиальной задачей. Частые случаи столкновения морских судов в частности и аварийных ситуаций в целом потребовали разработку средства автоматизированной радиолокационной прокладки (САРП), тем самым обеспечив безопасное движение судов в зоне контроля.

Модуль САРП обеспечивает возможность безопасного разрешения ситуаций, вызванных сложным движением судов в зоне контроля. Наблюдение за движением объекта позволяет получить траекторию судна, оценить скорость и расстояние между объектами на экране и использовать предупредительный сигнал с указаниями действий необходимых для предотвращения конфликтных ситуаций.

Также данный модуль может использоваться для охраны территории –автоматически локализовать объекты и предоставлять их координаты оператору.

Для своевременного предотвращения аварийных ситуаций и обеспечения контроля, необходимо, чтобы система сопровождения удовлетворяла ряду критериев:

1. Точность расчёта параметров сопровождаемого судна – географические координаты, курс, скорость, оценка размеров;
2. Сопровождение в реальном времени – изменения параметров движения объекта должны предоставляться на лету, то есть разница во времени между изменениями параметров и их расчётом должна быть минимальна;
3. Возможность своевременной индикации манёвра судна –сообщение о резком изменении курса или скорости судна должно поступать вовремя;
4. Возможность сопровождения большого числа объектов – в зоне действия РЛС должны быть локализованы все суда;
5. Корректность работы модуля в случаях сильных морских волнений;
6. Возможность дифференциации объектов с пересекающимися траекториями.

Необходимо, чтобы все перечисленные параметры были реализованы при разработке алгоритма сопровождения, что является нетривиальной задачей, ввиду ограничения по вычислительным мощностям. В первой главе данной работы описывается общая концепция работы модуля, на примере алгоритма, реализованного в индикаторе «Нева-ОПВ», его плюсы и минусы. Весь алгоритм разбит на несколько модулей, каждый из которых подробно описан и оценен. В главах 2-4 приведён ряд усовершенствований данных модулей, которые обеспечивают более быструю и точную оценку данных, получаемых с радиолокационной системы.

Так, во второй главе рассматривается алгоритм разбора радиолокационной информации с использованием вычислений на видеопроцессоре. Данные технологии позволяют значительно ускорить обработку и получить геометрические характеристики всех объектов в зоне действия РЛС за конечное время, удовлетворяющее условиям, описанным выше.

В третьей главе работы описывается метод прогнозирования положения судна, его сравнение с методом, использующим в своей основе линейную модель движения на основе графиков, полученных в результате применения этих методов. Также здесь описан алгоритм выявления манёвра.

В четвёртой главе представлен трёхступенчатый метод нахождения подходящих отметок в области поиска (стробе). Этот алгоритм учитывает данные, полученные на предыдущих шагах, что позволяет выбрать нужную отметку даже в сильно зашумлённой увеличенной из-за наличия манёвра области.

# Глава 1. Характеристика алгоритма сопровождения целей в радиолокационной системе

Алгоритм сопровождения по радиолокационной развёртке можно разделить на четыре этапа:

1. Предварительная обработка радиолокационного сигнала;
2. Обработка радиолокационного сигнала – выделение объектов и вычисление их параметров;
3. Прогнозирование положения отметки с учётом предыдущих значений;
4. Выбор отметки в стробе сопровождения;
5. Разбор сложных случаев сопровождения нескольких объектов с пересекающимися или близко лежащими траекториями.

Ввиду того что исследования на тему САРП не теряют актуальности, было разработано множество алгоритмов сопровождения. В этой работе будут рассмотрены подробнее вышеописанные пункты и оценены, на примере алгоритма сопровождения целей в индикаторе «Нева-ОПВ».

## Предварительная обработка радиолокационного сигнала

Из полученных от радиолокационных станций (РЛС) данных формируются изображения, каждый пиксель которых кодируется числом от 0 до 255, отвечающим за определённый цвет градации серого. Далее изображения подвергаются обработке, целью которой служит удаление помех. Затем отфильтрованное изображение преобразуется в 2-х битный массив, то есть каждый пиксель принимает значение от 0 до 3, где 0 означает, что объект на изображении отсутствует, а 3 – яркость объекта максимальна.

Методы фильтрации могут быть разными и зависят от различных факторов – погодные условия, параметры целей, которые надо сопровождать. В данной работе больше не будет уделено внимания этому пункту.

На рисунке 1.1 показана работа одного из элементарных наборов фильтров. Серым цветом изображены входные данные, цветами жёлтых оттенков – отфильтрованные данные, представленные, как описано выше пикселями, кодируемыми значениями от 0 до 3. На рисунке 1.а. применён только фильтр преобразующий изображение в 2-х битный массив; на 1.б. – набор фильтров, выделяющих подвижные объекты: удалены статичные объекты и уменьшено количество отметок, появившихся из-за волнения поверхности воды вблизи берега.

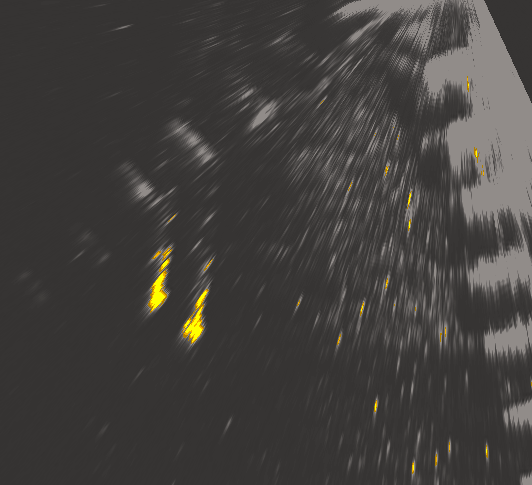
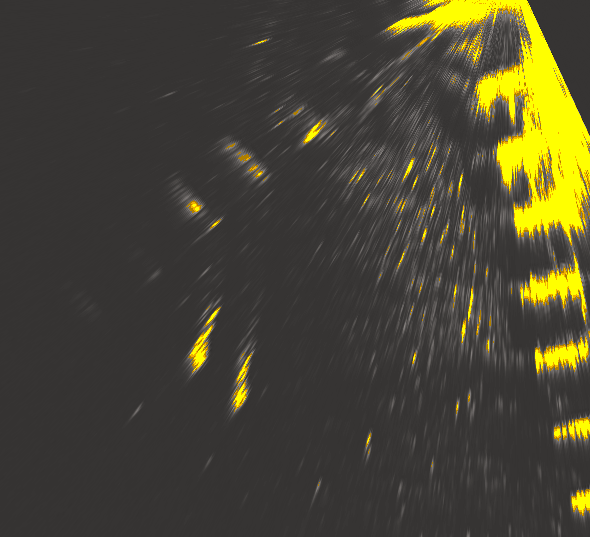


Рисунок 1.1. Пример применения фильтров

## Выделение объектов и вычисление их параметров

После обработки изображений происходит их анализ, выделение связных областей, которые в дальнейшем можно принять за возможные положения целей, вычисление их параметров – геометрических позиций, размеров.

Проблема, поставленная в данной работе, заключается в необходимости сопровождения небольших маневрирующих объектов, движущихся в условиях значительных волнений морской поверхности. Решение данной задачи заключается в доработке алгоритмов, позволяющей использовать высокие вычислительные нагрузки, оптимизации их параметров и адаптации вычислений для современных процессоров. В алгоритме, реализованном в индикаторе «Нева-ОПВ» этой проблеме не уделяется должного внимания, изображение усредняется для уменьшения объёмов данных, поступающих на обработку из-за чего теряется точность и появляется угроза объединения небольших объектов с шумами, образованными из-за волнения, другими суднами или неподвижными объектами, что ведёт к невозможности сопровождения. В некоторых случаях обработка изображений происходит не в реальном времени, а с некоторой задержкой или обрабатываются не все данные – вычисления происходят или на небольшой области изображения, что приводит к проблемам с сопровождением целей, выходящих за её пределы, или обрабатываются не все изображения.

В данной работе будут рассматриваться новые алгоритмы обработки, реализованные с применением параллельных вычислений на видеопроцессоре (CUDA). Будут показаны преимущества такого подхода, который гарантирует быструю обработку больших данных без потери точности расчётов.

## Прогнозирование положения отметки

В данном пункте будут рассмотрены основные концепции САРП. Они с течением времени не были подвержены каким-либо изменениям, поэтому будут описаны кратко.

Дан набор координат, полученный извне (выбор пользователя), где приблизительно находится отметка, характеризующая выбранное для сопровождения судно. Вокруг этой точки строится зона S1 - строб первичного захвата (рисунок 1.2), в котором ищется максимально близкая к точке отметка, если такой отметки не нашлось, посылается об этом сигнал.

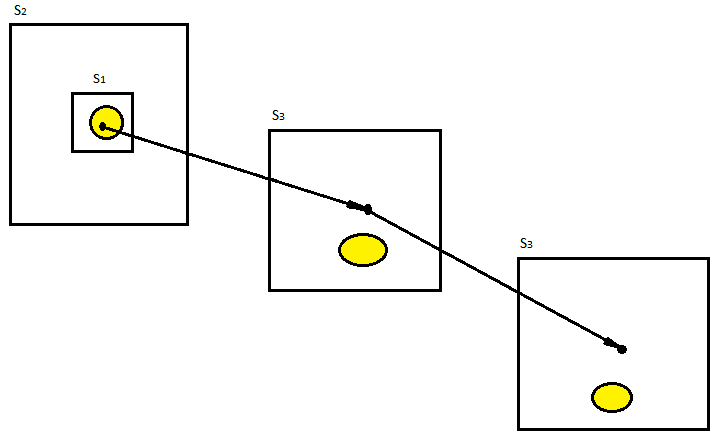


Рисунок 1.2. Схема сопровождения

Далее строится зона S2, - строб вторичного захвата, размер которого выбирается в зависимости от возможного манёвра судна. В этой зоне по некоторым критериям, которые будут описаны далее, ищется вторая отметка. После нахождения второй отметки можно примерно представить, как движется судно – его моментальные курс и скорость.

; ;

; .

Тут и – координаты отметки в декартовых координатах.

Получив значения скорости и курса можно примерно оценить, где в следующий момент окажется объект. В данной точке будет строиться новая область S3 - строб сопровождения, в котором будет искаться отметка, характеризующая новое положение судна. Таким же образом строятся стробы S4, S5, S6,.. для последующих положений судна.

Очевидно, что чем точнее алгоритм, лежащий в основе вычисления нового возможного положения отметки, тем больше вероятность успешного сопровождения судна. Обычно для этих целей используется метод наименьших квадратов в основе которого лежит линейная модель. Иногда используется упрощённый фильтр Калмана, как например в алгоритме, реализованном в индикаторе «Нева-ОПВ». В основной части работы в главе 3 будет проведено сравнение данных методов и метода, используемого в разработанном алгоритме.

Так как рассматриваются быстрые маневрирующие объекты, то важно опередить начала манёвра и вовремя принять меры, этот пункт также будет рассмотрен в главе 3.

## Выбор отметки в стробе сопровождения

В данной работе рассматривается случай сопровождения быстрых маневрирующих судов, в связи с этим сколь малой бы ни была ошибка алгоритма прогнозирования траектории, он не может абсолютно точно поймать момент манёвра и проследить подобного рода траекторию. Чтобы избежать срыва целей (цель ушла за пределы зоны поиска), необходимо увеличить область, в которой выбираются отметки – строб сопровождения, так что её размеры будут включать в себя поправку на возможность сложного манёвра.

Увеличение строба сопровождения приведет к увеличению количества ложных отметок, попавших в него. Также необходимо учитывать, что из-за движения таких целей образуются сильные волнения – остаточный шлейф от движения. На рисунке 1.3 представлен пример – скутер оставил за собой «хвост» – отражение от кильватерного следа, который после фильтрации отобразился четырьмя отметками (жёлтый цвет). Подобные артефакты могут стать сильной проблемой для нахождения истинной отметки.

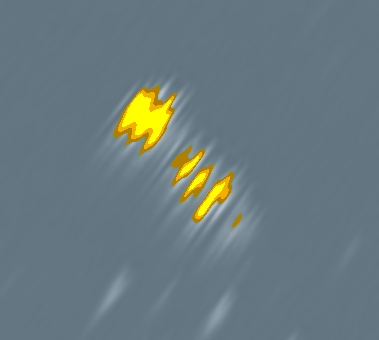


Рисунок 1.3. «Хвост» от отметки

Данная проблема решается тем, что используются набор критериев, по которым из попавших в строб отметок выбирается одна (или несколько), далее принимаемая за истинную и по которой строятся дальнейшие расчёты. Обычно критерии включают в себя оценку расстояния отметок до построенной точки, являющейся центром зоны поиска и примитивные оценки геометрических характеристик (размеры отметки). В главе 4 будет рассматриваться трёхэтапный выбор отметок.

## Сложные случаи сопровождения целей

Помимо проблем с отслеживанием манёвров и зашумлённостью изображений существует проблема сопровождения объектов, движущихся по сложным пересекающимся траекториям.

На рисунке 1.4 показан пример такого движения – два скутера на большой скорости движутся сильно маневрируя, и в трёх местах их пути пересекаются. Скутеры показаны отметками жёлтого цвета, их след представлен зелёным цветом. Проблема здесь заключается в том, что в какие-то моменты времени стробы сопровождения обоих объектов имеют общие области. Объекты довольно похожи по характеристикам, и поэтому вероятно, что критерии, описанные выше, не подойдут для выбора отметок. Необходимо учесть и этот фактор.

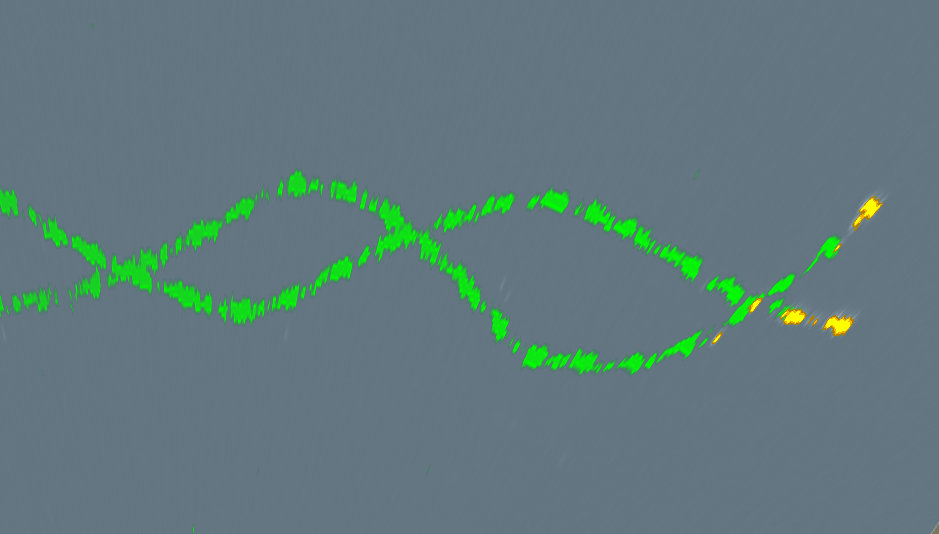


Рисунок 1.4. Сложное движение двух объектов

Рисунок 1.5 иллюстрирует описанную проблему – сопровождаются две цели, представленные двумя отметками (обозначены жёлтым цветом, сопровождение показано белым кружочком). Стробы сопровождения обозначены белыми четырёхугольниками. Обе отметки лежат на пересечении этих стробов, они расположены достаточно близко друг другу и похожи по форме.

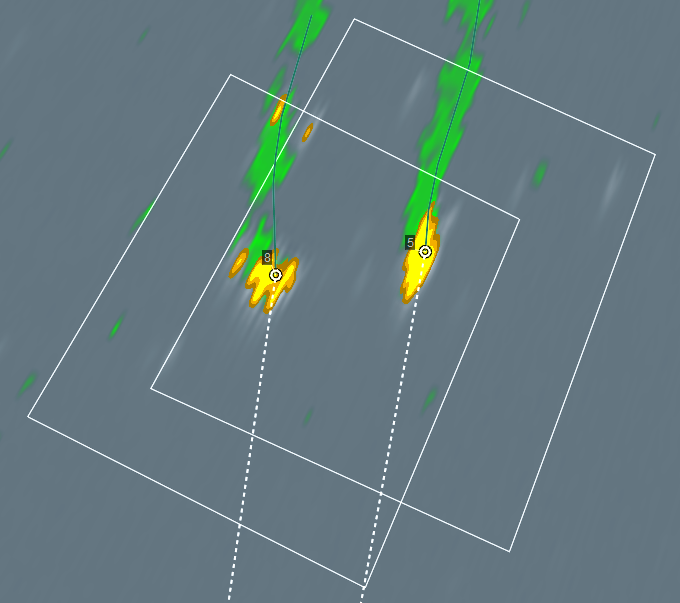


Рисунок 1.5. Пересечение стробов сопровождения

# Глава 2. Выделение объектов и вычисление их параметров

## Общая информация

После того, как радиолокационная информация была отфильтрована и преобразована в 2-х битный массив, где каждый пиксель кодируется значениями от 0 до 3 (0 – отсутствие сигнала, 1 – слабый сигнал, 2 – средний, 3 – сильный), ее необходимо обработать – выделить отметки, которые потенциально могут быть в дальнейшем положениями сопровождаемого объекта.

Далее в этом пункте будет введено понятие связных областей. Связная область – набор пикселей, у каждого из которых есть хотя бы один соседний пиксель, принадлежащий данному множеству.

Виды связности:

* 4-связность – соседями для пикселя являются четыре пикселя, расположенных сверху, слева, справа, снизу от него, то есть все пиксели, которые имеют общее с ним ребро.
* 8-связность – соседями для пикселя являются восемь пикселей, которые имеют или общее с ним ребро или общую вершину.

На рисунке 2.1 показаны примеры связности – жёлтым цветом показан пиксель, относительно которого ищутся связные с ним соседи, которые в свою очередь отображены зелёным цветом.

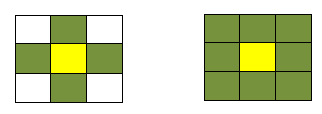


Рисунок 2.1. Примеры связных областей. а) 4-связность, б) 8-связность

В данной работе будут рассматриваться только 4-связные области. Самостоятельными отметками будут являться те, которые отвечают критериям 4-связных областей.

На рисунке 2.2 показана небольшая область изображения, которое поступает после фильтрации. Серым цветом обозначено входное изображение, оттенками жёлтого – обработанное. Как видно, даже на небольшом участке возможно наличие огромного числа связанных отметок. Обычно на зашумлённом изображении присутствуют порядка нескольких десятков тысяч отметок.

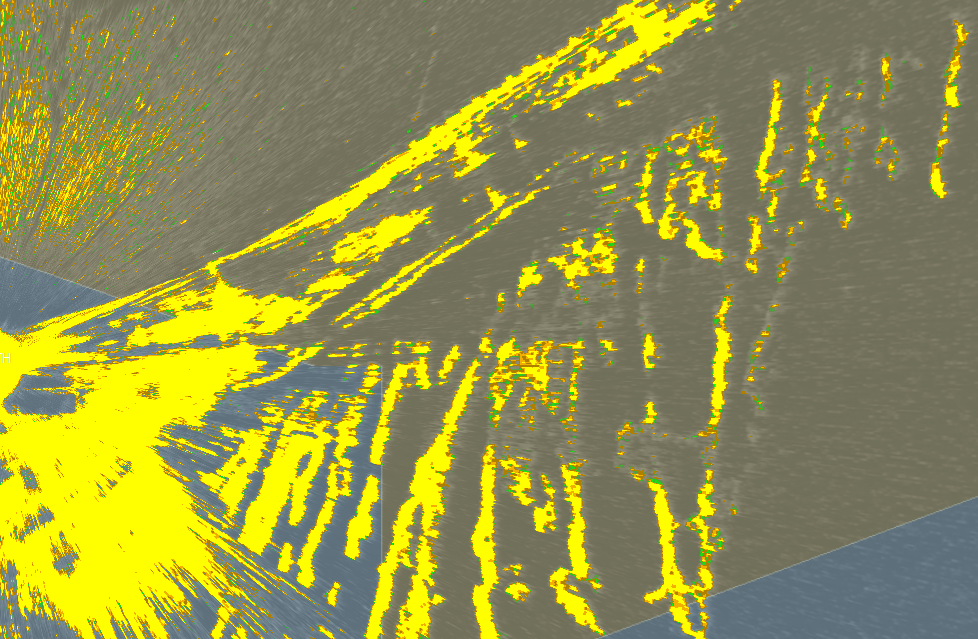


Рисунок 2.2. Пример входного изображения

Каждую отметку необходимо выделить, присвоить уникальный номер и рассчитать все геометрические характеристики. Все эти вычисления должны происходить достаточно быстро – на изображение размерами 4000 \* 4000 пикселей, отводится несколько секунд. Очевидно, с такими ограничениями подобную задачу обычными средствами не получится решить. Для этого будем использовать технологию параллельных вычислений на видеопроцессоре, в частности будем применять программно-аппаратную архитектуру CUDA, разработанную компанией NVIDIA.

## Алгоритм выделения связных областей

Сравнение нескольких алгоритмов разработанных под расчёты на видеопроцессоре представлены в статье [3]. В данной работе опишем один из методов, представленных в статье, модифицированный для произведения более быстрых расчётов.

Задачу выделения связных областей можно рассмотреть с позиции задачи обхода графа. В качестве звеньев графа выступают точки массива, а соединяющее звенья ребро соответствует ребру между точками, принятыми за соседние. Первоначально все элементы входного массива, значения которых больше либо равно 1, должны быть перенумерованы в соответствии с занимаемым положением в массиве. Если число в массиве рано 0, то оно не меняет своего значения. Далее необходимо, пройдя по соседним точкам, например, используя правило обхода жадным алгоритмом – идти в направлении большего соседа, дойти до точки с максимальным значением, которое потом будет распространено на все точки связной области.

На рисунке 2.3 представлен пример такого обхода. За начальную точку, была принята точка с индексом 2, конечной оказалась точка с индексом 34.

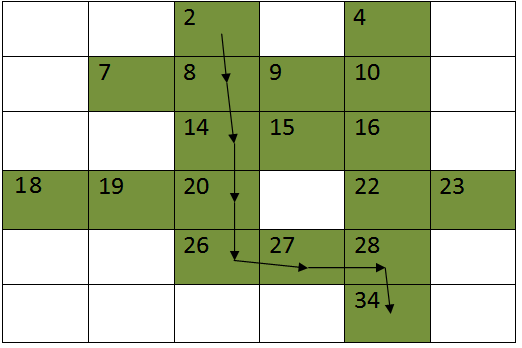
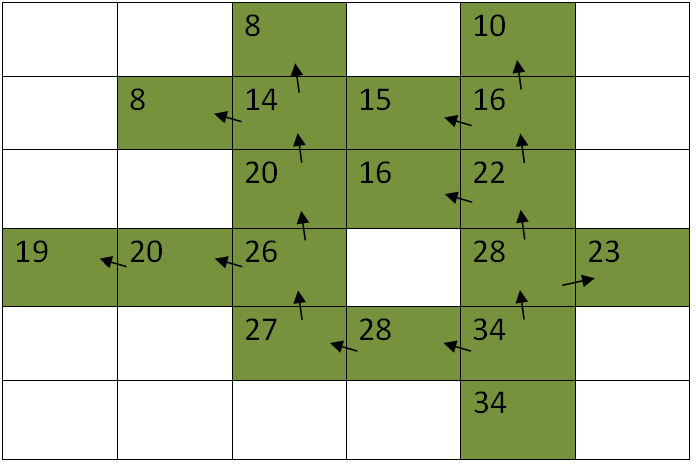
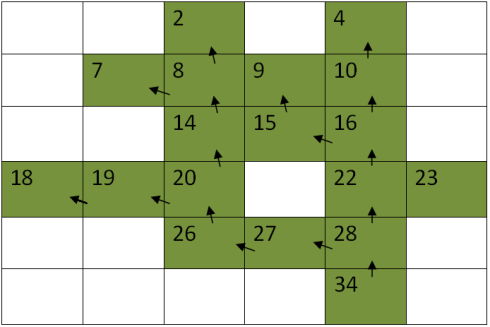
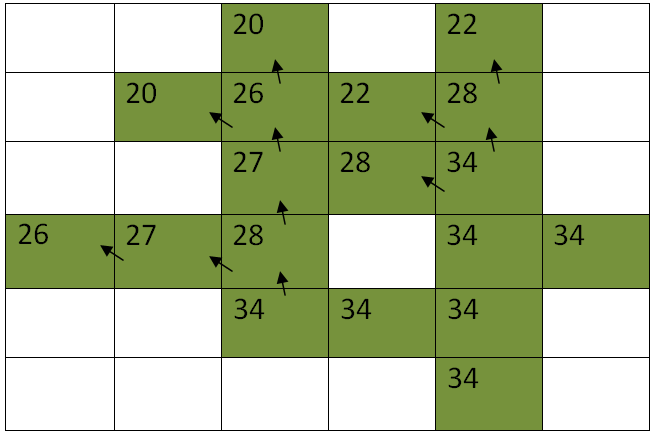
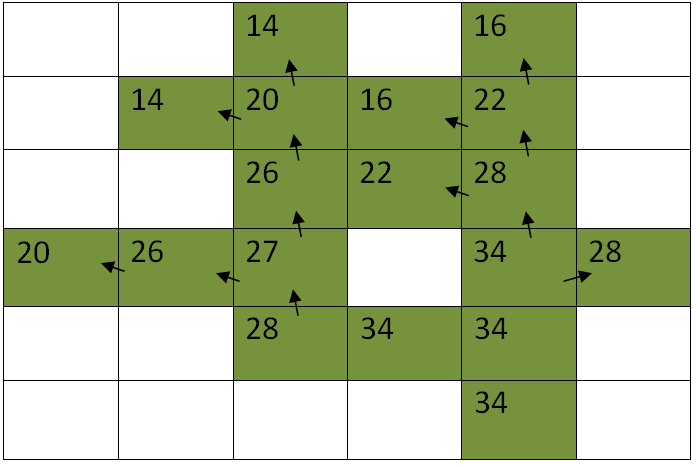
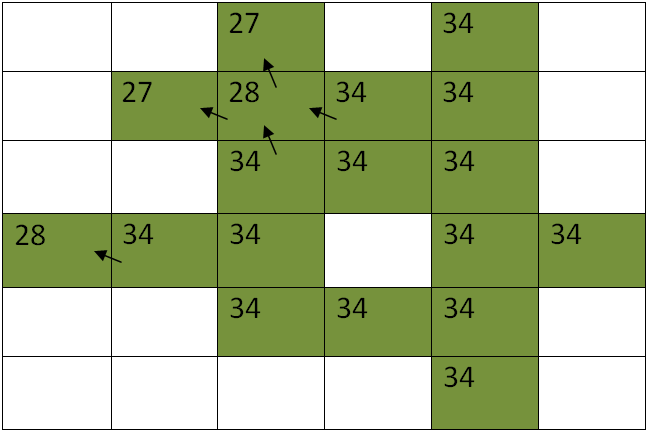
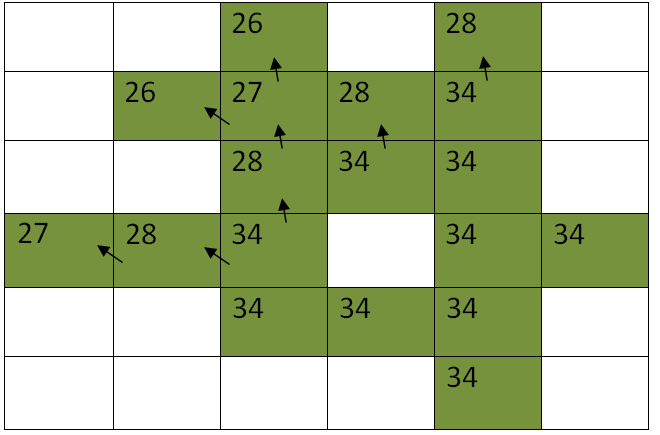


Рисунок 2.3. Пример обхода жадным алгоритмом

Используя эту концепцию сформулируем алгоритм для параллельных вычислений. Пусть в каждом потоке выполняются операции сравнения и замены для каждого пикселя изображения, то есть – каждый пиксель, не равный 0 сравнивается с соседями, выбирается тот, чей номер будет больше и это число будет записываться наместо того, которое было у рассматриваемого пикселя. Очевидно, что после окончания работы всех потоков все пиксели отметки не будут содержать один номер, поэтому процедура должна повторяться, пока у всех пикселей отметки не найдётся ни одного соседа с большим номером. Пример работы проиллюстрирован на рисунке 2.4.







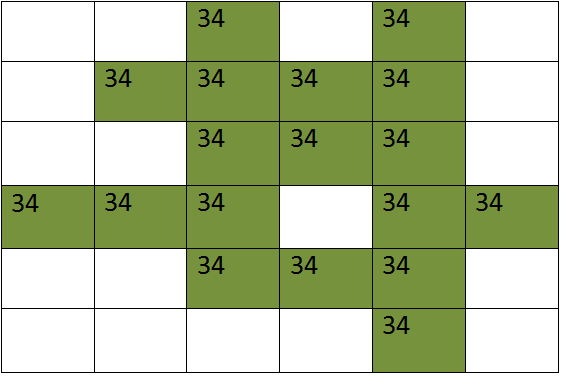
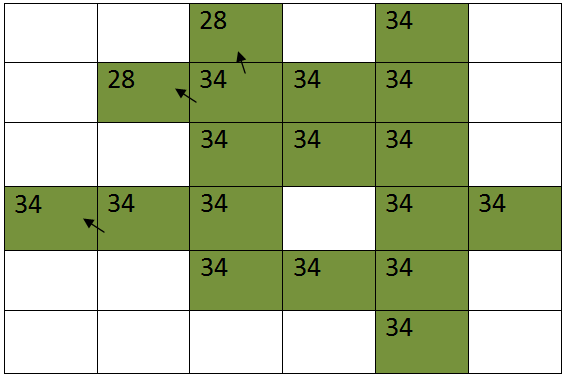


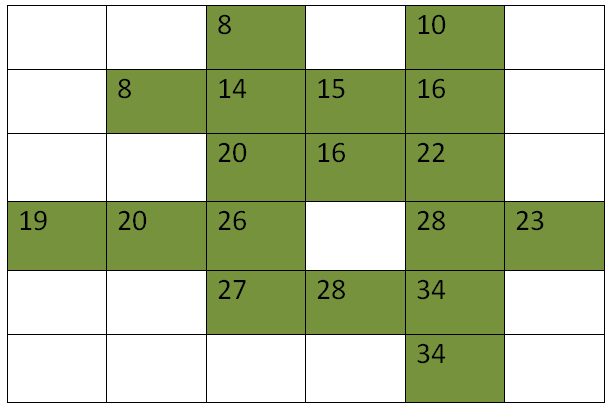
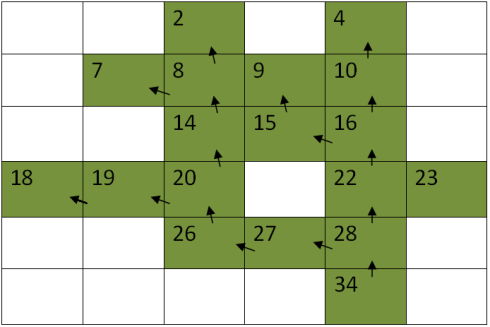
Рисунок 2.4. Иллюстрация работы алгоритма

Как видно на иллюстрации, для небольшой несложной отметки достаточно конечного количества итераций – в данном случае 8. В последней итерации происходит проверка на то, что у всех пикселей отметки соседей с большими индексами не осталось.

## Усовершенствованный алгоритм выделения связных областей

Этот алгоритм прост, легко реализуем, и даёт хорошие результаты по скорости. Его можно усовершенствовать, если вместе с вышеописанных функций на каждой итерации вызывать ещё одну. В этой добавочной функции в цикле для каждого потока у пикселя *c[i]* проверяется его порядковый номер и сравнивается с записанным в пиксель индексом. Если они не совпадают, то рассматривается пиксель, расположенный под номером равным индексу первого пикселя, аналогично для нового пикселя сравниваются порядковый номер и присвоенный пиксель – происходит вторая итерация цикла. Цикл заканчивается тогда, когда у очередного пикселя номер позиции и индекс совпадут. Индекс, записанный в последнем пикселе, записывается в первый. Алгоритм можно записать формулами:

На рисунке 2.5 показана работа обоих функций вместе.



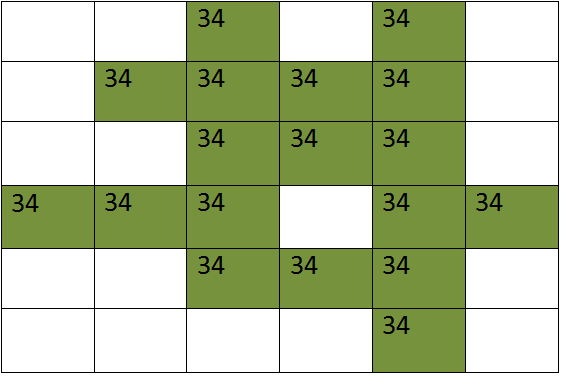


Рисунок 2.5. Иллюстрация работы алгоритма

Далее приведён пример работы алгоритма второй функции по шагам для нескольких точек.

Для точки, находящейся в ячейке 2:

После прохода первой функции вместо 2 стало 8 – 2 не равно 8, идём на точку 8, на месте 8 записан индекс 14 – 8 не равно 14, далее проходим точки 20, 26, 27, 28 и в конечном итоге доходим до 34. Таким образом, в пиксель, расположенный на месте с номером 2 записываем индекс 34.

Цепочка для точки, находящейся в ячейке 4:

4<-10<-16<-22<-28<-34.

Цепочка для точки, находящейся в ячейке 7:

7<-8<-14<-20<-26<-27<-28<-34.

Цепочка для точки, находящейся в ячейке 8:

8<-14<-20<-26<-27<-28<-34.

Аналогичные цепочки можно построить для всех остальных точек. В итоге получится, что все индексы заменятся на значение равное 34 – максимальное значение в массиве. Из этих рассуждений видно, что вместо 8 итераций цикла, используются только две, в первой из них производится изменения индексов, а во второй конечное сравнение с соседними элементами. Таким образом, время, затраченное на обработку изображения, сокращается в *m* раз в зависимости от сложности обрабатываемых отметок, где где *n* - количество точек, принадлежащих отметке.

## Вычисление параметров отметок

После того как были выделены связные области, каждый пиксель отметки идентифицируется номером, который в дальнейшем считается индексом отметки. При этом по описанному выше алгоритму следует, что, после обработки любая точка, у которой существует связная с ней точка, имеет индекс равный индексу соседа, и нету ни одной пары отметок с совпадающими индексами.

Таким образом, можно сказать, что на выходе алгоритма имеем массив, в каждой точке которого записан номер, показывающий принадлежность точки к отметке. Имея это можно составить функцию, вызывающееся для каждого пикселя всей картинки отдельно (каждый поток вызывается для отдельной точки изображения) и содержащее в себе следующий набор действий:

1. Читается значение пикселя *c[i]*;
2. Проверяется значение *c[i]* на неравенство 0;
3. При прохождении условия в пункте 2, в структуре, соответствующей отметке с индексом, равным индексу *c[i]* изменяются параметры.

Ниже приведены параметры, которые рассчитываются для отметки:

* Сумма «цвета» (сумма значений, записанных в изображении, полученном после фильтрации всех пикселей отметки):

;

* Центр отметки по дистанции (отстояние отметки от центра радарной развёртки):
* Центр отметки по углу (значение угла между отметкой и осью, проведённой от центра в направлении севера):
* Максимальное значение по дистанции (максимум значений отстояния всех пикселей отметки от центра радарной развёртки):
* Максимальное значение по углу (максимум значений угла между пикселями отметки и осью, проведённой от центра в направлении севера):
* Минимальное значение по дистанции (минимум значений отстояния всех пикселей отметки от центра радарной развёртки):
* Минимальное значение по углу (минимум значений угла между пикселями отметки и осью, проведённой от центра в направлении севера):

;

Так как операции между потоками происходят асинхронно, то нельзя из разных потоков использовать операции преобразования одного и того же элемента. Например, при вычислении суммы цвета, из каждого потока, принадлежащих пикселю отметки, добавляется число, равное значению пикселя, записанному во входном массиве – из нескольких потоков происходит чтение и запись значений в одну точку графической памяти. В какой-то момент времени запись в одном потоке может произойти в интервал времени между чтением и записью в другом потоке, что приведёт к тому, что действия в первом потоке в конечном итоге не произведут ни каких изменений в вычисляемом параметре.

Чтобы избежать подобных проблем, в функции используются атомарные операции сложения и сравнения – операции, в которых происходит чтение и перезапись в одну ячейку графической памяти одновременно, тесть, пока происходит работа с параметром из одного потока, другие потоки, не могут обратиться к этому участку памяти и переводятся в режим ожидания. Минус использования данных операция заключается в том, что увеличивается время выполнения функции, но это время значительно меньше того, которое выигрываем, используя данные алгоритмы, что на это недостаток можно не обращать внимания.

На рисунке 2.5 показан пример работы алгоритма. Цветами, жёлтого оттенка представлены отметки. Вокруг отметок, по значениям, полученным при расчёте параметров, построены белые прямоугольники, очерчивающие границы отметок. Белый кружок представляют собой индикатор сопровождаемой цели и расположен в центре тяжести отметки.



Рисунок 2.5. Пример работы

# Глава 3. Прогнозирование положения отметки, определение манёвра

## Прогнозирование положения отметки

Рассматривается упорядоченный ряд радиолокационных развёрток, полученных через фиксированный интервал времени. На одном из них выбирается некоторый объект для сопровождения (отметка). Необходимо найти данный объект на следующих за выбранным изображениях и построить траекторию его движения. Считается, что радиолокационная станция неподвижна (изображения располагаются в одних координатах), количество подвижных объектов на изображении ограничено (морские суда).

На рисунке 3.1 выше проиллюстрирована траектория, построенная по координатам, соответствующим центрам отметки. Как видно по рисунку, траектория движения дополнительно зашумлена.



Рисунок 3.1. Пример траектории, построенной по зашумлённым данным

Также мы не имеем точных сведений относительно параметров ошибки в получаемых геометрических позициях объектов. Они могут возникать как из-за неточности аппаратных средств (погрешность аппаратуры, неточности калибровки РЛС, шумы на линии передачи данных), так и из-за погодных условий, общей обстановки и первичной обработки картинок (цветокоррекция, удаления шума). Поэтому использовать более алгоритмы, учитывающие факторы влияния шумов, довольно проблематично.

Дополнительно, в силу ограничений производительности процессора и объёму используемой оперативной памяти, был наложен ряд ограничений при выборе методов. Сравнение нескольких алгоритмов прогнозирования положения представлены в статье [10]. В этой работе будет описан метод наименьших квадратов (МНК) для квадратичной зависимости.

Метод наименьших квадратов является простым и эффективным решением для поиска кривой, максимально близко проходящей возле заданных точек. В качестве функции в описываемом методе будет использоваться полином:

Параметры будут вычисляться таким образом, чтобы минимизировать функционал:

где *n* – размерность пространства, в котором происходят вычисления, в нашем случае *n = 2*. *m* – степень полинома, тут *m = 2*.

Для квадратичной зависимости полином, описывающий искомую функцию, будет выглядеть:

где

Параметры будут вычисляться таким образом, чтобы минимизировать функционал:

где *n* – количество измерений, участвующих в поиске параметров.

Количество измерений, необходимых для расчёта параметров должно удовлетворять следующим критериям:

1. Оно должно быть достаточным, чтобы позиция отметки, координаты которой были искажены шумом, не оказывала сильного влияния на рассчитываемую траекторию;
2. Одновременно с вышеуказанным пунктом, оно должно быть не сильно большим, чтобы успеть зафиксировать манёвр объекта.

Экспериментально было получено, что лучшим значением для количества измерений, участвующих при вычислении параметров является значение *n = 20*.

После решения системы уравнений и нахождения параметров, описывающих кривую траектории, можно найти координаты, в которых будет находиться строб сопровождения, в котором будет искаться новая отметка.

На рисунке 3.2 сиреневым и зелёным цветом изображены кривые, построенные по точкам, соответствующим центрам отметки и точкам, построенным по спрогнозированным координатам соответственно. Как видно из рисунка, кривые достаточно близко идут друг к другу, при том полученная кривая (показана зелёным цветом) получилась более гладкой, что показывает эффективность работы алгоритма.

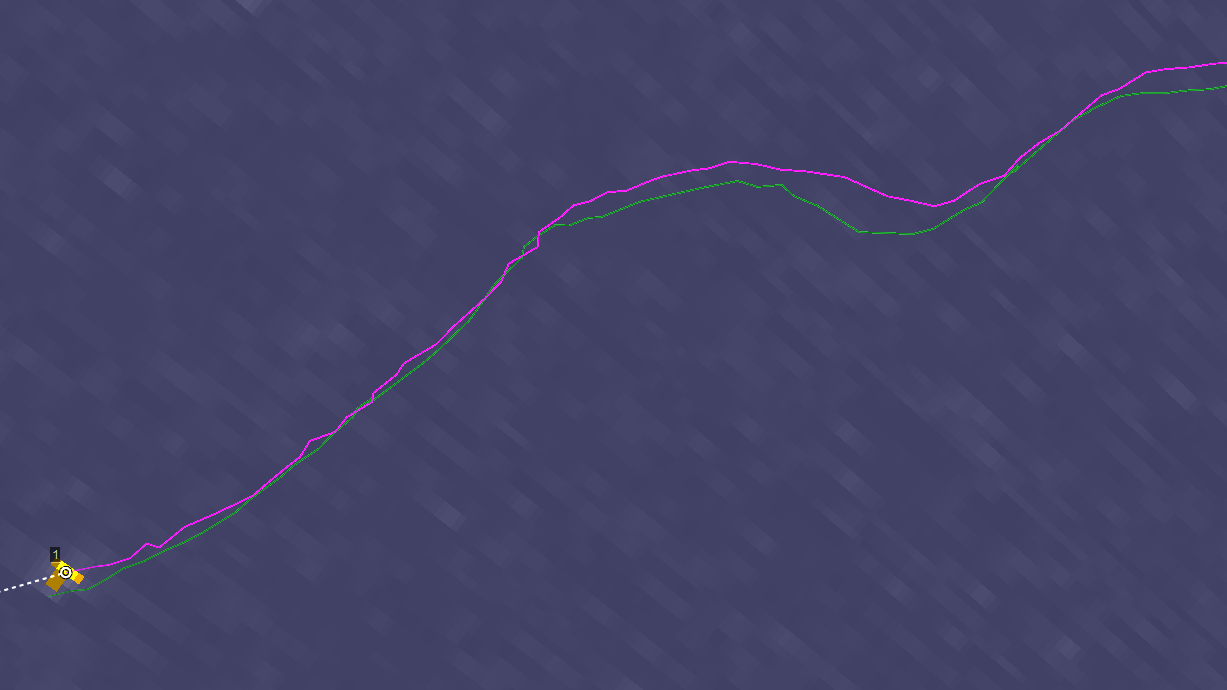


Рисунок 3.2. Пример траектории, построенной по полученным зашумлённым данным

## Сравнение алгоритмов с использованием линейной и квадратичной модели движения

Чтобы показать эффективность описанного выше метода, в данном пункте приводится сравнение использования метода наименьших квадратов для линейной и квадратичной моделей движения. Для этого на каждом шаге сопровождения, при добавлении новой отметки, рассчитывается ошибка аппроксимации и ошибка прогноза.

Ошибка аппроксимации:

Ошибка, описанная формулой, является суммой от значений координат, полученных при использовании описанной модели и зашумлённых значений координат отметки на интервале *n*.

На рисунке 3.3 изображены две кривые, синим цветом показаны значения ошибки для линейной модели, красным – ошибки квадратичной модели. Как видно из рисунка, использованная тут квадратичная модель даёт значительно лучшие результаты.

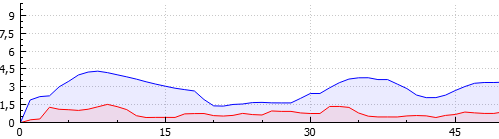


Рисунок 3.3. Графики ошибок аппроксимации

Ошибка прогноза:

Приведённая ошибка является отклонением найденного значения от предсказанного на предыдущем шаге. Данная ошибка показывает, насколько неточный результат дала использованная модель.

На рисунке 3.4 показаны графики ошибок. Аналогично предыдущему, синим цветом показаны значения ошибки для линейной модели, красным – ошибки квадратичной модели. В отличии от графика ошибок аппроксимации, график ошибок прогнозирования не показывает сильного превосходства одной модели над ругой. Но если рассмотреть время, когда был произведён манёвр (промежуток возле 30 и 45 шага), то видно, что квадратичная модель справилась лучше.

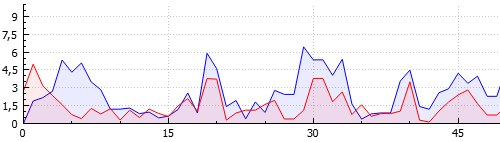


Рисунок 3.4. Графики ошибок прогнозирования

## Строб сопровождения отметки

Из графика ошибок прогнозирования, очевидно, невозможно добиться точного определения позиции объекта, на это влияют как шумы, появляющиеся на исследуемом изображении из-за волнение морской поверхности, ошибок аппаратуры и прочего, так и маневрирования сопровождаемого объекта. Очевидно также, что нельзя ожидать от сопровождаемого судна, прямолинейности и равномерности движения.

Из вышеописанного следует, что при обнаружении следующей отметки, необходимо построить область её поиска – строб сопровождения. Далее необходимо оценить размеры данного строба. При этом должно учитываться, что чем больше строб, тем больше вероятность выбрать неверную отметку, как продолжение траектории цели.

В данной работе размеры строба вычисляются как сумма трёх параметров:

* Размер самого объекта, значение которого получаем при обработке поступающей радиолокационной информации. Вычисление этих параметров описано в предыдущей главе;
* Задаваемое фиксированное значение, вычисляемое с учётом физически возможного значения перемещения объекта за время между приходом двух изображений;
* Значение ошибки прогнозирования.

## Определение маневра

В этой работе манёвром судна будет считаться резкое изменение курса и скорости. Если не иметь алгоритма определение начала манёвра, то можно допустить значительную ошибку экстраполяции центра строба сопровождения, потому что используемые алгоритмы довольно инертны и какое-либо резкое приращение значений координат на одном шаге, не приведут к сильным изменениям параметров модели. Поэтому важно иметь возможность вовремя определить момент, когда следует предпринимать действия по определению маневра, чтобы не потерять объект.

На рисунке с ошибкой аппроксимации по графику линейной модели можно увидеть, что есть два манёвра, один начинается ориентировочно на 30м шаге, второй – на 45м. Оценка графика аппроксимации для выявления манёвра не подходит по причинам пологости графика. График ошибок прогнозирования в данном случае будет более показателен

Область поиска (строб сопровождения) строится с учётом возможного манёвра сопровождаемого объекта. Сравнивание ошибки прогнозирования с размером области поиска отметки даёт возможность определения манёвра: если ошибка оказалась больше половины размера строба сопровождения, можно считать, что произошел маневр.

После определения момента маневрирования судна, необходимо предпринять меры, которые обеспечат возможность не потерять цель – необходимо длину выборки сопровождения для построения траектории, чтобы увеличить влияние вновь пришедших и увеличить размер области поиска следующей отметки.

# Глава 4. Выбор отметки в области сопровождения

В случаях резкого маневрирования довольно сложно, а иногда невозможно предсказать точно положение отметки в следующий момент времени, в связи с этим необходимо расширять строб сопровождения отметки, что приводит к увеличению сторонних объектов, которые могут попасть в него. Помимо этого, здесь рассматривается ситуация сильного морского волнения, что также усложняет задачу поиска. Для решения проблемы здесь будет предложен метод трёхступенчатой проверки отметок.

## Первый этап обработки

Для каждой отметки вычисляется параметр, являющийся суммой «цвета» отметки, подробнее это описывается в главе 2. На каждом шаге сопровождения при добавлении новой отметки, рассчитывается математическое ожидание данного параметра на некотором промежутке – *n* точек, где *n* выбирается из тех же соображений, что и количество точек для вычисления параметров модели прогнозирования в главе. Отсюда следует, что *n = 20*.

Очевидно, как бы ни двигалась цель, отметка от неё не сможет увеличиться в *m* раз, где *m* – некоторое конечное число. После ряда экспериментов для разных целей в различных ситуациях, в качества *m* было взято число 10.

Таким образом, на первичном этапе селекции отметок, выбираются те, размеры которых не превышают средних размеров отметки цели по последним 20 позициям в 10 раз. Аналогично размеры отметки не должны быть меньше в 10 раз, чем среднее значение.

## Второй этап обработки

Помимо среднего значения суммы «цвета», для целей можно оценить статистические параметры – математическое ожидание и дисперсию для размеров по углу и дистанции. Имея данные параметры для пришедших отметок можно оценить величины:

где:

, , – математическое ожидание для суммы цвета, размеров по углу и по дистанции соответственно.

,, – дисперсии для суммы цвета, размеров по углу и по дистанции соответственно.

,, – значения суммы цвета, размеров по углу и по дистанции для фильтруемой отметки соответственно.

Данные величины оценивают отклонение параметров отметки от значений, накопленных по мере сопровождения цели, чем больше это значение, тем меньше вероятность соответствия отметки и цели. В качестве конечного функционала, по которому будет производиться выборка отметок можно взять сумму из представленных трёх параметров:

;

Описанные выше величины также включают в себя характер движения цели – если на всём протяжении пути отметка была по какому-либо из параметров нестабильна (большое значение дисперсии), то вычисляемое значение будет малым, и поэтому будет не сильно влиять на конечное значение функционала .

Зачастую получается, что в строб сопровождения может попасть несколько отметок, мало отличающихся по форме, и истинная отметка может дать несколько худший результат. На рисунке 4.1 показан пример данного поведения. Как видно по рисунку – параллельным курсом на небольшом расстоянии идут два небольших объекта – скутера, отметки которых мало отличаются друг от друга.

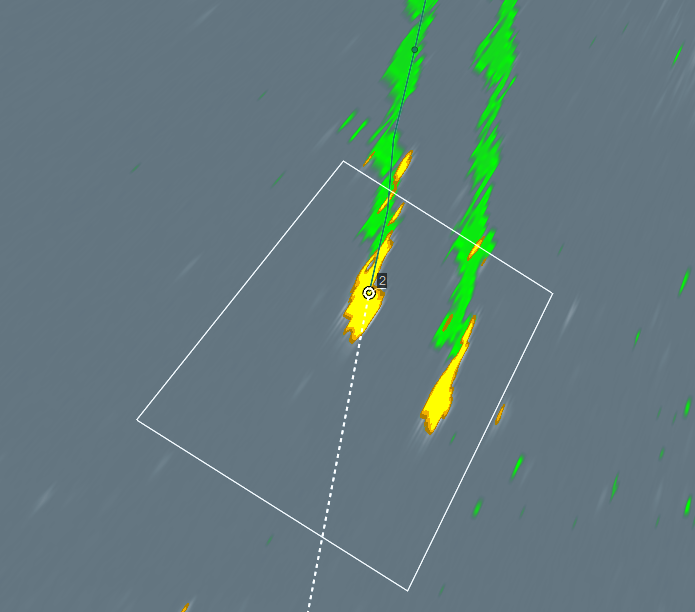


Рисунок 4.1. Пример сложной ситуации

Чтобы не пропустить истинную отметку строгим отбором, необходимо пропускать все отметки, незначительно отличающиеся от максимально похожей. Таким образом, необходимо найти значение функционала, имеющее наименьшее значение - и сравнивая для всех отметок, найденных в стробе сопровождения выбрать те, которые удовлетворяют условию:

где *k* подбирается в зависимости от зашумлённости картинки и маневрировании цели – чем больше шумов и меньше скорость, тем больше должно быть значение *k* (лучше всего оказалось значение *k = 3*).

## Третий этап обработки

Из пришедших второй этап отметок необходимо выбрать одну. Третий этап обработки включает в себя сравнение отклонений отметок в стробе сопровождения от координат, вычисленных на стадии поиска спрогнозированного положения цели (центра строба). Таким образом, для каждой отметки вычисляются значения:

.

Затем выбирается отметка с отклонением равным:

.

# Заключение

В данной работе было представлено средство автоматизированной радиолокационной прокладки успешно реализующее обнаружение и сопровождение морских судов. Одним из модулей такого типа является алгоритм, реализованный в индикаторе «Нева-ОПВ». В первой главе была описана основная концепция этого алгоритма, оценены его достоинства и недостатки. Далее были приведены усовершенствования для данного алгоритма, улучшающие сопровождение целей в случаях:

* Малых маневрирующих судов;
* Сильных помех от морского волнения.

Время выполнения алгоритма было сокращено за счёт использования вычислений на видеопроцессоре; это привело к возможности расширения охватываемой РЛС области обработки и увеличения количества единовременно сопровождаемых объектов.

Также, в связи с тем, что были использованы метод наименьших квадратов для квадратичной модели движения и трёхэтапный алгоритм селекции отметок в стробе сопровождения, появилась возможность сопровождения быстрых маневрирующих судов.

На рисунке показана работа алгоритма, реализованного в индикаторе «Нева-ОПВ». На рисунке 1 изображены дискретные траектории (зелёные пятна) двух небольших быстроходных объектов (жёлтые пятна), идущих по пересекающимся траекториям. Зелёная и тёмно-синяя кривые построены по координатам расположения отметки в момент сопровождения – непрерывная траектория сопровождения. По траекториям сопровождения видно, что во время второго пересечения траекторий произошёл перескок – в качестве истинной была принята неправильная отметка, что привело к сопровождению ложного объекта. Серыми линиями показаны рассчитанные направления движения: одна из отметок была потеряна – ее линия начинается с места, где жёлтой отметки уже нет.

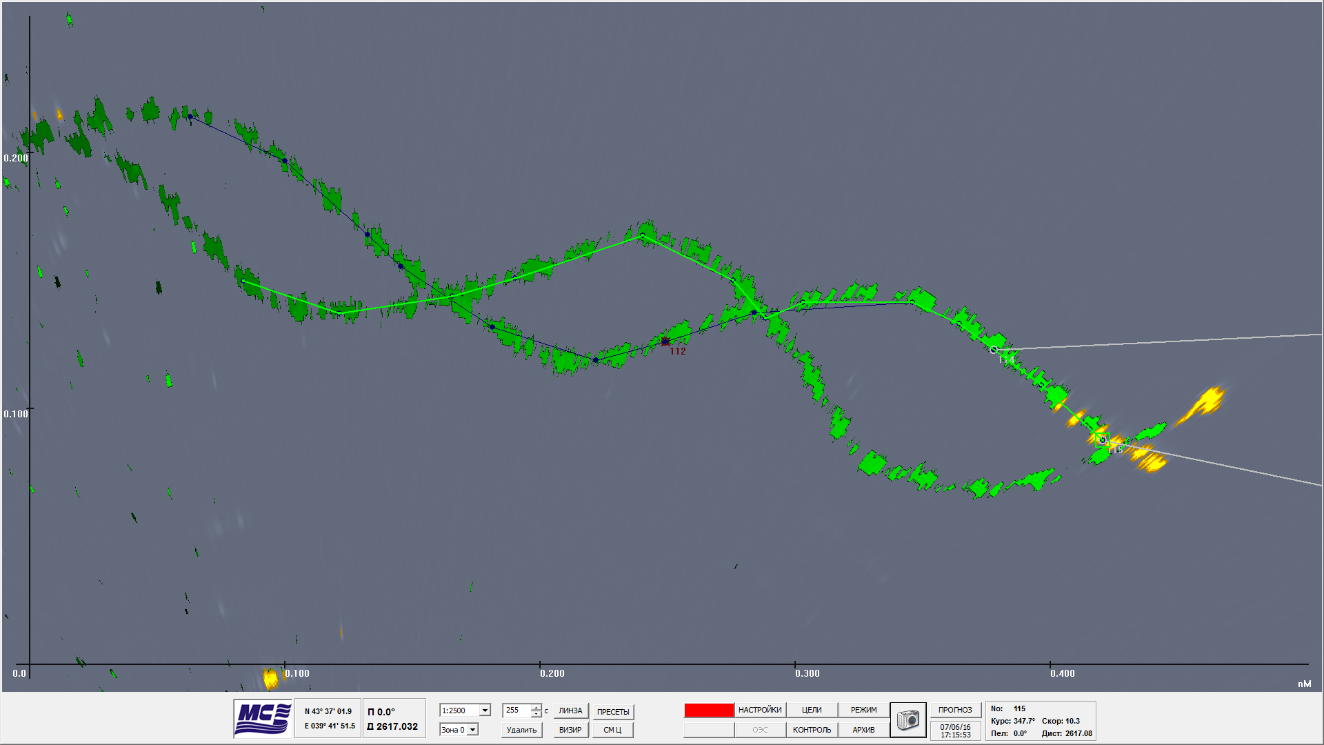


Рисунок 1. Индикатор «Нева-ОПВ»

На рисунке 2 проиллюстрирован описанный в данной работе алгоритм. Дискретные траектории движения и непрерывные траектории сопровождения аналогичны предыдущему примеру. В данном случае сопровождение произошло успешно с поправкой на краткосрочную неопределённость в последней точке пересечения траекторий.



Рисунок 2. Пример работы алгоритма

Описанный в данной работе алгоритм был успешно внедрён в индикатор «Смарт-Арм» и в данный момент находится в стадии тестирования.

# Список литературы:

1. Джейсон Сандерс, Эдвард Кэндрот - Технология CUDA в примерах. Введение в программирование графических процессоров / Джейсон Сандерс, Эдвард Кэндрот. — М.: ДМК Пресс, 2011.
2. Томас Х. Кормен и др. Глава 16. Жадные алгоритмы // Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS. — 1-е изд. — М.: Московского центра непрерывного математического образования, 2001. — С. 889-892. — ISBN 5-900916-37-5.
3. Сравнение алгоритмов выделения связных областей, реализованных с использованием платформы CUDA. - [Современные информационные технологии и ИТ-образование. -  2014. - Т. 1. - № 1(9). - 957 с.](http://elibrary.ru/contents.asp?titleid=52785)
4. Bedrich Benes, Ondrej Stava, Radomir Mech, and Gavin Miller, Guided Procedural Modeling, in Computer Graphics Forum (Eurographics), 2011. pp:325-334
5. Линник Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений. — 2-е изд. — М., 1962. (математическая теория)
6. Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering 82: 35, 1960.
7. Кузьмин С.З. Основы проектирования систем цифровой обработки радиолокационной информации. - М.: "Радио и связь", 1986 - 352 с.
8. Peter, Matisko. Optimality Tests and Adaptive Kalman Filter. 16th IFAC Symposium on System Identification. 16th IFAC Symposium on System Identification. p. 1523, 2012.
9. Fuller, W. A. Measurement Error Models. John Wiley & Sons, 1987.
10. Сравнение методов прогнозирования траектории морских судов. - Современные информационные технологии и ИТ-образование. -  2015. - Том. 2. - № 11. - 541 с.
11. Real time prediction of track while scan data. - 6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation