



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа ядерных технологий

Направление подготовки – 01.03.02 Прикладная математика и информатика

ООП/ОПОП – Прикладная математика в инженерии

Отделение школы (НОЦ) – Отделение экспериментальной физики

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Алгоритмическое и программное обеспечение георадара на основе применения методов миграции

УДК 004.415:550.837.76:519.237

Обучающийся

Группа	ФИО	Подпись	Дата
0В91	Кравцов Константин Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегуров Александр Иванович	к.т.н., доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Кащук Ирина Вадимовна	к.т.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Антоневич Ольга Алексеевна	к.б.н., доцент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ИЯТШ	Крицкий Олег Леонидович	к.ф.-м.н., доцент		



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа ядерных технологий

Направление подготовки – 01.03.02 Прикладная математика и информатика

ООП/ОПОП – Прикладная математика в инженерии

Отделение школы (НОЦ) – Отделение экспериментальной физики

УТВЕРЖДАЮ:

Руководитель ООП/ОПОП

_____ Крицкий О. Л.
(Подпись) (Дата)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
0В91	Кравцов Константин Александрович

Тема работы:

Алгоритмическое и программное обеспечение георадара на основе применения методов миграции	
Утверждена приказом директора (дата, номер)	

Срок сдачи обучающимся выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Данные в виде временных рядов, отображающие распределение энергии электромагнитных волн в различных точках среды, полученные с помощью приемной антенны георадара. Модель среды в виде деревянной оболочки с арматурой и пластиковыми трубками внутри, залитой бетоном.
Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке	1. Разработка и реализация алгоритмов двумерной и трехмерной гиперболической миграции 2. Разработка и реализация алгоритмов двумерной и трехмерной миграции Столта 3. Разработка и реализация алгоритма интерполяции двумерных данных методом обратного взвешивания 4. Тестирование разработанных алгоритмов на искусственной модели среды
Перечень графического материала	1. Сфокусированные и несфокусированные изображения георадарных сканов

	2. Рисунки, демонстрирующие данные в трехмерном формате с помощью программ, способных представлять трехмерные данные в виде вокселей 3. Поясняющие рисунки, объясняющие виды георадарных данных и принципы работы алгоритмов миграции 4. Фотография модели среды
Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Кашук Ирина Вадимовна, Доцент ОСГН ШБИП
Социальная ответственность	Антоневич Ольга Алексеевна, Доцент ООД ШБИП

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Кочегуров Александр Иванович	к.т.н., доцент		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
0В91	Кравцов Константин Александрович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕДИНЕНИЕ И РЕСУРСОБЪЕДИНЕНИЕ»

Студенту:

Группа	ФИО
0B91	Кравцову Константину Александровичу

Школа	Инженерная школа ядерных технологий	Отделение Школа	Экспериментальной физики
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов инженерного проекта: материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г. Томска. Тарифные ставки исполнителей определены исходя из средних по рынку труда в г. Томске по версии hh.ru
2. Нормы и нормативы расходования ресурсов	Норма амортизационных отчислений на специальное оборудование
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Отчисления во внебюджетные фонды 30 %

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Анализ конкурентных технических решений	Расчет конкурентоспособности SWOT-анализ
2. Формирование плана и графика разработки и внедрения	Структура работ. Определение трудоемкости. Разработка графика проведения исследования
3. Составление бюджета инженерного проекта	Расчет бюджетной стоимости НИ
4. Оценка ресурсной, финансовой, бюджетной эффективности	Интегральный финансовый показатель. Интегральный показатель ресурсоэффективности. Интегральный показатель эффективности.

Перечень графического материала

1. Оценка конкурентоспособности ИП
2. Матрица SWOT
3. Диаграмма Ганта
4. Бюджет НИ
5. Основные показатели эффективности НИ

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Кашук Ирина Вадимовна	к.т.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0B91	Кравцов Константин Александрович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
0B91		Кравцову Константину Александровичу	
Школа	Инженерная школа ядерных технологий	Отделение (НОЦ)	Экспериментальной физики
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Тема ВКР:

Алгоритмическое и программное обеспечение георадара на основе применения методов миграции	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	<i>Объект исследования:</i> георадарное программное обеспечение <i>Область применения:</i> исследование фундаментов и несущих конструкций зданий <i>Рабочая зона:</i> офисное помещение <i>Размеры помещения:</i> 27,2 м ² <i>Количество и наименование оборудования рабочей зоны:</i> 1 персональный компьютер <i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> алгоритмическая и программная разработка с использованием персонального компьютера
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	–ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования» –ГОСТ 22269-76 Система "Человек-машина". Рабочее место оператора. Взаимное расположение элементов рабочего места – Трудовой кодекс Российской Федерации: федер. Закон от 30 дек. 2001 г. №197-ФЗ Раздел 10
2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов	Опасные факторы: – Опасность поражения электрическим током; Вредные факторы: – Отклонение показателей микроклимата; – Недостаточная освещённость рабочей зоны; – Пониженная световая и цветовая контрастность; – Повышенный уровень шума на рабочем месте; – Статические физические перегрузки; – Перегрузка зрительного аппарата;
3. Экологическая безопасность при разработке проектного решения:	Анализ воздействия на литосферу: – Утилизация компьютеров, оргтехники и бумаги; Анализ воздействия на гидросферу: – Производство компьютерной техники; Анализ воздействия на атмосферу:
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения:	– Наиболее типичная ЧС: Пожар;
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Антоневич Ольга Алексеевна	к.б.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
0B91	Кравцов Константин Александрович		

Реферат

Выпускная квалификационная работа содержит 114 страниц, 19 рисунков, 24 таблицы, 27 источников, 6 приложений.

Ключевые слова: алгоритмы миграции, гиперболическая миграция, миграция Столта, интерполяция методом обратного взвешивания, преобразование Фурье, временная область, частотная область

Объектом разработки являются алгоритмы интерполяции двумерных данных (методом обратного взвешивания) и миграции георадарных изображений (гиперболическая миграция и миграция Столта)

Цель работы – разработать алгоритмическое и программное обеспечение георадара, представляющее собой набор базового функционала, позволяющего преобразовывать несфокусированные георадарные изображения в сфокусированные

В результате работы разработаны и реализованы алгоритмы миграции георадарных изображений, а именно: алгоритм гиперболической миграции и алгоритм миграция Столта. Реализован алгоритм интерполяции двумерных методом обратного взвешивания, применяемый для преобразования набора двумерных несфокусированных изображений в целостное трехмерное изображение

Для реализации программного комплекса георадара выбран язык Python. Для подготовки данных для тестирования использовалась программа grgMax, симулирующая распространение электромагнитных волн в цифровых моделях сред.

Для итогового тестирования разработанных алгоритмов использовался макет бетонной среды, представляющий собой залитую бетоном деревянную оболочку с закрепленными внутри арматурными стержнями и пластиковыми трубками.

Содержание

Введение.....	11
1. Применение георадаров для исследования грунтов и несущих конструкций. Постановка задачи.	13
2. Анализ существующих методов миграции изображений	16
2.1. Задача миграции несфокусированного изображения и симуляция несфокусированных изображений	16
2.2. Применение модели «взрывающегося» отражателя для исследования методов миграции.....	18
2.3. 2D и 3D гиперболическая миграция.....	19
2.4. 2D FK и 3D FK миграция.....	21
2.5. Миграция на основе применения методов интерполяции	25
3. Разработка алгоритмов двумерной и трехмерной миграции.	
Программная реализация алгоритмов.....	30
3.1. Алгоритмы гиперболической миграции	30
3.2. Алгоритмы миграции Столта.....	32
3.3. Алгоритмы интерполяции методом обратного взвешивания.....	34
4. Миграция реальных данных на основе разработанных алгоритмов.....	37
4.1. Формирование реальных данных	37
4.2. Результаты применения миграции к реальным данным	39
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	41
5.1. Введение	41
5.2. Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения ..	41
5.2.1. Анализ конкурентных технических решений	41

5.2.2. SWOT-анализ.....	43
5.3. Планирование работ.....	47
5.3.1. Структура работ в рамках инженерного проекта	47
5.3.2. Определение трудоемкости выполнения работ и разработка графика проведения.....	48
5.4. Бюджет инженерного проекта	53
5.4.1. Расчет амортизации специального оборудования	54
5.4.2. Основная заработная плата исполнителей темы	56
5.4.3. Отчисления во внебюджетные фонды (страховые отчисления)	59
5.4.4. Накладные расходы	59
5.4.5. Бюджет инженерного проекта	60
5.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности проекта.....	61
5.6. Выводы по разделу	65
6. Социальная ответственность	66
6.1. Введение	66
6.2. Правовые и организационные вопросы обеспечения безопасности	66
6.3. Производственная безопасность.....	67
6.3.1. Опасность поражения электрическим током	68
6.3.2. Отклонение показателей микроклимата.....	69
6.3.3. Недостаточная освещенность рабочей зоны	71
6.3.4. Повышенная световая и цветовая контрастность.....	72
6.3.5. Повышенный уровень шума на рабочем месте	73
6.3.6. Статические физические перегрузки	74
6.3.7. Перегрузка зрительного аппарата	75

6.4. Экологическая безопасность.....	75
6.5. Безопасность в чрезвычайных ситуациях	77
6.5.1. Пожар	77
6.6. Вывод раздела.....	78
Заключение	80
Список использованных источников	81
Приложение А	86
Приложение Б.....	89
Приложение В	93
Приложение Г	97
Приложение Д	102
Приложение Е.....	107

Введение

В современном мире появляется все больше и больше технологий неразрушающего контроля, которые применяются во многих профессиональных сферах. Одними из таковых являются технологии, реализованные с использованием методов радиолокации, используемых в таких областях как медицина, строительство, охрана социально-значимых объектов и объектов транспортной инфраструктуры (вокзалов, метро, аэропортов и т.д.).

Каждая из сфер, в которых используются радиолокационные технологии, имеет свои особенности, а именно: различные объекты сканирования, различные устройства, используемые при сканировании и различные диапазоны излучаемых длин волн. Одной из основных областей применения радиолокационных технологий является георадарные обследования.

Георадарное обследование — это инструментальный метод диагностики, с помощью которого проводится исследование грунтов на строительной площадке, а также фундаментов и несущих конструкций различных объектов. Исследование с помощью георадара относится к неразрушающим методам и позволяет определить структуру грунта или конструкций без бурения шурфов и сверления отверстий. Кроме того, георадар позволяет обнаруживать пустоты и инженерные коммуникации под поверхностью грунта [1].

В качестве инструмента для георадарного обследования грунта или железобетонных конструкций фундамента используется георадар — геофизический прибор, который позволяет сканировать почву или строительные конструкции на большой глубине. Это устройство излучает волны на определенной частоте, которые отражаются от скрытых предметов, а затем воспринимаются приемником и анализируются микропроцессором.

После чего оператор получает на экране изображение грунта или строительной конструкции под поверхностью [1].

Целью работы является разработка алгоритмического и программного обеспечения георадара, включающего в себя базовые функции миграции георадарных изображений. Для её достижения необходимо решить ряд задач:

- Разработать и реализовать алгоритмы двумерной и трехмерной гиперболической миграции
- Разработать и реализовать алгоритмов двумерной и трехмерной миграции Столта
- Разработать и реализовать алгоритм интерполяции обратного взвешивания, для подготовки реальных трехмерных данных для миграции.
- Протестировать реализованные алгоритмы на искусственной модели бетонной среды. Проинтерпретировать результаты тестирования.

1. Применение георадаров для исследования грунтов и несущих конструкций. Постановка задачи.

В сфере строительства георадар часто представляет собой устройство, похожее на четырехколесную тележку, которую можно перемещать вдоль некоторой плоскости. Зачастую тележка обладает следующим набором основных компонент: аккумуляторный блок, энкодер (датчик угла поворота), векторный анализатор цепей, лазерные направляющие и две антенны, одна из которых излучает радиоволны, а другая принимает отраженный от расположенных в среде объектов сигнал. В комплекте с георадаром идет и специальное программное обеспечение, способное установить соединение с георадаром, принять с устройства все необходимые сигналы, обработать их в режиме реального времени и выдать графическое изображение на дисплей устройства (чаще всего, используемого вместе с георадаром планшета, поскольку его удобно переносить вместе с тележкой).

Таким образом, георадар возвращает на устройство представленный в цифровом виде радиосигнал, излученный и принятый в точке, где была расположена тележка георадара. Эти статические измеренные данные, собранные в одной точке, называются А-сканом [2].

Обычно исследование поверхности в одной точке является нецелесообразным и дающим наименьшее количество возможной полезной информации. Если собрать данные в некотором наборе точек, расположенных на прямой, получится В-скан. В ситуации измерения В-скана направленная вниз антенна георадара перемещается по прямой траектории на внешней поверхности, в то время как датчик GPR (Ground Penetrating Radar) собирает и записывает рассеянное поле в различных пространственных позициях [2].

Соответственно, собранные на площади А-сканы представляют собой набор В-сканов, по-другому называемый С-сканом. Все сканы представляют собой напряженность электрического поля на поверхности среды с течением

времени и отличаются лишь областью сканирования (в точке, по прямой, по площади). Обозначенные типы сканов изображены на рисунке 1.1.

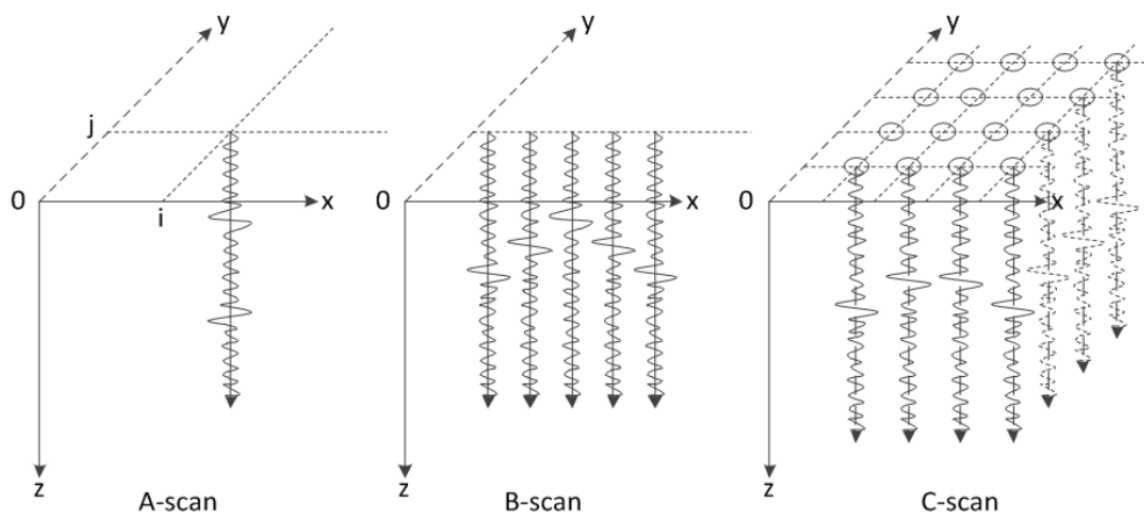


Рисунок 1.1 – Виды георадарных измерений [3]

На типичном изображении В-скана рассеиватели отображаются как гиперболические кривые из-за разного времени пробега электромагнитной волны во время движения антенны в направлении сканирования. Такая структура изображения может быть достаточной, когда задачей сканирования является только обнаружение целей. Однако в большинстве случаев требуется ориентировочная информация о расположении, форме и размерах целей [2]. Поскольку информация является неполной, то изображение, показывающее только распространение электромагнитных волн на поверхности среды с течением времени, называют нефокусированным. Таким образом, одной из наиболее часто решаемых задач с использованием георадарного изображения В-скана является преобразование (или миграция) нефокусированного пространственно-временного изображения (рисунок 1.2, слева) в сфокусированное изображение (рисунок 1.2, справа), показывающее истинное местоположение и размер отражателей с их некоторой отражательной способностью [2].

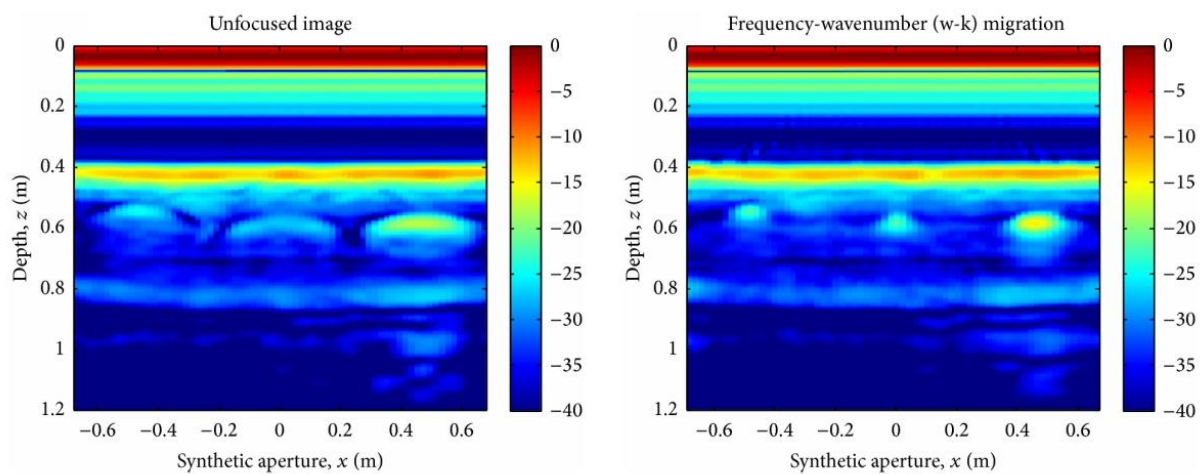


Рисунок 1.2 – Несфокусированное изображение (слева) и сфокусированное изображение (справа) [4]

2. Анализ существующих методов миграции изображений

2.1. Задача миграции несфокусированного изображения и симуляция несфокусированных изображений

При отсутствии георадара, способного предоставить исходные данные программисту, разрабатывающему реализацию алгоритмов миграции, можно прибегнуть к симуляции несфокусированных изображений. Для симуляции распространения волн в среде использовался проект gprMax.

gprMax — это программное обеспечение с открытым исходным кодом, которое моделирует распространение электромагнитных волн с использованием метода конечных разностей во временной области (FDTD) для численного моделирования георадара (GPR). gprMax изначально был разработан в 1996 году, когда численное моделирование методом FDTD и вообще численное моделирование георадара только зарождались. Современные вычислительные ресурсы дают возможность создавать подробные и сложные FDTD-модели георадара в такой степени, которая ранее была невозможна [5].

Для разработки алгоритмов миграции В-сканов была сгенерирована модель закопанного под среду (с диэлектрической проницаемостью, равной 6, характерной для бетона) цилиндра с идеальной отражающей способностью. Цифровая модель такого цилиндра изображена на рисунке 2.1. Для визуализации модели используется Open-Source проект ParaView [6].

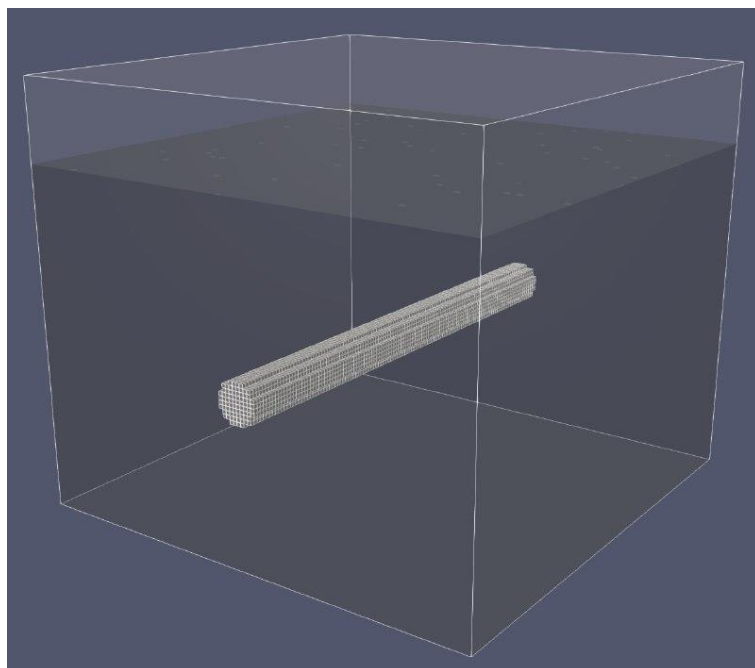


Рисунок 2.1 – Модель закопанного под среду цилиндра

С использованием gprMax на данной модели проведена симуляция распространения радиоволн в среде, так, как если бы георадар перемещался поперек цилиндра. В результате получен В-скан (рисунок 2.2).

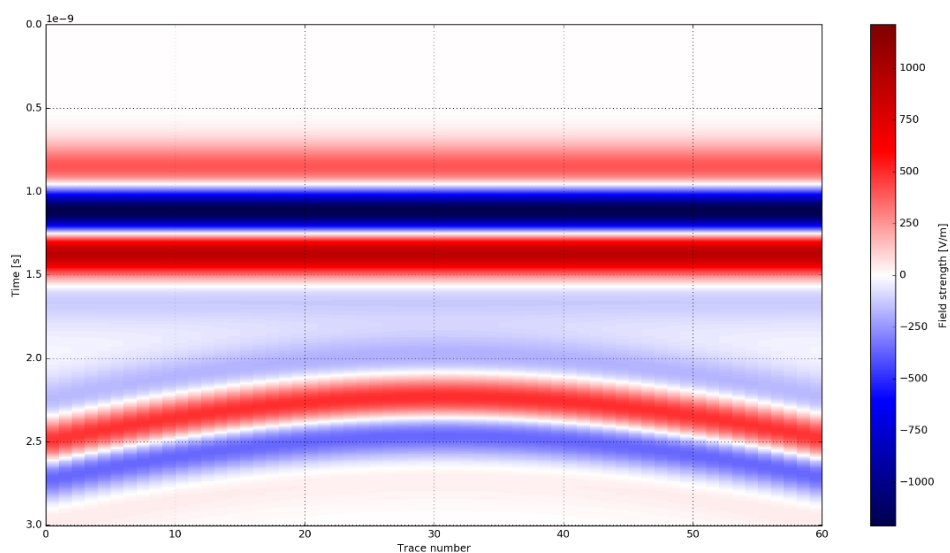


Рисунок 2.2 – измерение электромагнитного поля для заданной модели

Визуально понятно, что первая сине-красная полоса представляет собой отражение волн от плоскости, разделяющей диэлектрическую среду с воздушным пространством, а вторая полоса красно-синего цвета представляет собой отражение электромагнитных волн от цилиндрического тела.

2.2. Применение модели «взрывающегося» отражателя для исследования методов миграции

Большинство методов миграции основаны на концепции, называемой моделью «взрывающегося» отражателя (ESM, Exploding Source Model, часто ERM – Exploding Reflector Model). В 1985 г. ученый Клэрбаут пришел к умной идее думать об электромагнитном поле, приходящем в приемник радара, как если бы оно исходило от цели-отражателя. Таким образом, вместо того, чтобы рассматривать прохождение волны туда и обратно, можно предположить, что целевой отражатель как бы «взрывается» и вокруг своего реального местоположения посылает электромагнитную волну в приемник радара, как показано на рисунке 2.3 [4].

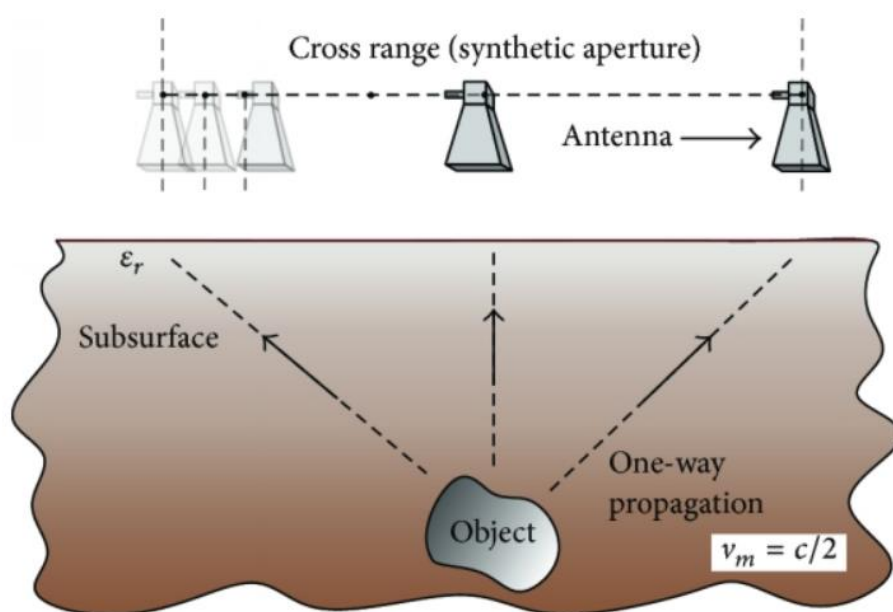


Рисунок 2.3 – модель «взрывающегося» отражателя [4]

В действительности, конечно, происходит двустороннее распространение волны между радаром и целью. Однако при использовании ESM предполагается, что собранные данные изначально исходят от цели. Следовательно, в ESM, изображенном на рисунке, моделируется одностороннее распространение. Поскольку время пробега электромагнитной волны становится равно половине действительного времени движения,

следует компенсировать скорость распространения электромагнитной волны, просто разделив ее на два: $v_m = c/2$ [4].

2.3. 2D и 3D гиперболическая миграция

В типичном применении георадара антенна радара улавливает рассеянную или отраженную назад электромагнитную волну от границы раздела воздух-земля и подземных объектов вместе со многими эффектами помех, в основном связанными с неоднородностями в пределах поверхности. В идеальном случае фаза рассеянного сигнала прямо пропорциональна времени пробега (или расстоянию), которое пройдет электромагнитная волна, в случае если среда распространения однородна [4]. В связи с этим, на изображениях В-сканов часто получается так, что волны, рассеянные от одного и того же объекта, интерферируя друг с другом, образуют четко выраженные параболические гиперболы (примеры гипербол можно рассмотреть на рисунке 2.4). Таким образом можно полагать, что целевой отражатель расположен в вершине, изображенной на В-скане параболической гиперболы.

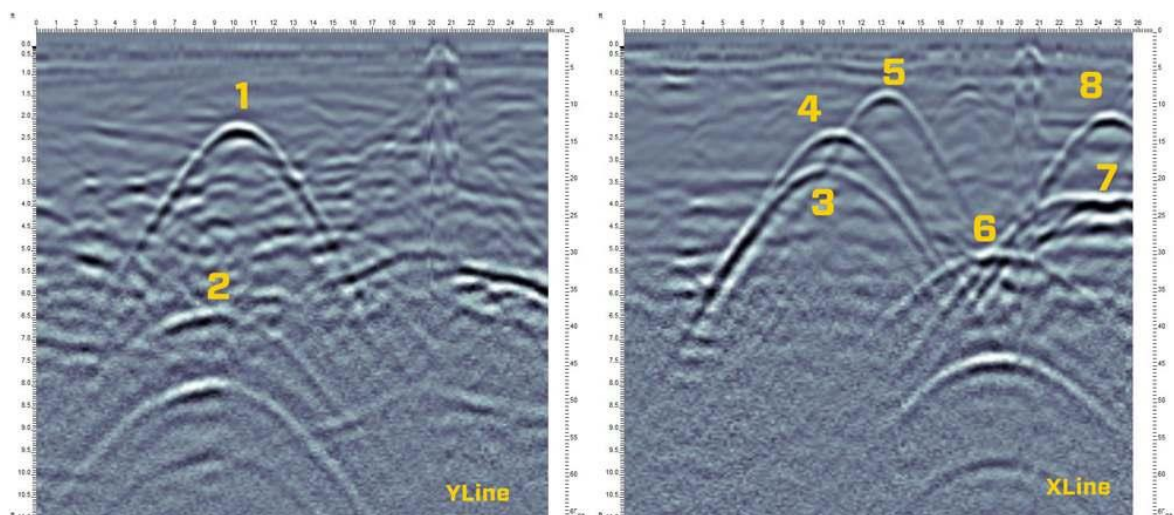


Рисунок 2.4 – пример характерных гипербол на изображении В-скана [7]

Если распространяющаяся среда однородна, параболическая гипербола на несфокусированном изображении может характеризоваться следующим

$$R = \sqrt{z_0^2 + (X - x_0)^2} \quad (2.1)$$

уравнением, во время движения радара по прямой траектории вдоль оси сканирования [4].

В приведенном выше уравнении X представляет собой вектор расположения радара вдоль В-скана и R представляет собой вектор длины пути от антенны до отражателя. Предполагая, что результирующее изображение можно рассматривать как вклад конечного числа гипербол, которые соответствуют различным точкам на отражателе (отражателях) под средой, можно применить следующую методологию для миграции несфокусированного изображения:

1. Для каждого пикселя в исходной двумерной матрице изображения В-скана получить набор пикселей, соответствующий шаблону параболической гиперболы (2.1).
2. Сохранить значения трассируемых пикселей по этому шаблону. На этом этапе алгоритма должен получиться одномерный массив данных, длина которого равна количеству пикселей по оси движения радара.
3. Затем вычислите среднеквадратичное значение энергии поля, для собранного по шаблону набора пикселей по формуле:

$$\{rms\ at\ (x_i, z_i)\} = \sqrt{\frac{(|E^s|^2 |(E^s)^*|^2)}{N}} = \frac{1}{\sqrt{N}} \sum_{i=1}^N |E_i^s|^2 \quad (2.2)$$

4. Вычисленное среднеквадратичное значение записывается в новое изображение. Этот процесс повторяется до тех пор, пока все пиксели исходного изображения не будут обработаны [4].

Трехмерная гиперболическая миграция по своему подобию похожа на двумерную гиперболическую миграцию. Основное отличие методов миграций заключается в размерности исходных данных. Нетрудно получить

соотношение для построения трехмерной гиперболы из соотношения (2.1) путем введения в рассмотрения дополнительной координаты y . Тогда формула для трехмерной фигуры, описывающей гиперболу в трехмерном пространстве, будет иметь вид:

$$R = \sqrt{z_0^2 + (X - x_0)^2 + (Y - y_0)^2} \quad (2.3)$$

Сам алгоритм расчета сфокусированного изображения не изменился. Стоит отметить, что временные затраты на реализацию алгоритма возросли с появлением дополнительной переменной, так как увеличилась сложность алгоритма до $O(N^3)$.

2.4. 2D FK и 3D FK миграция

Столт в 1978 г. предложил метод миграции, обычно называемый миграцией f - k (или иногда ω - k), позволяющая создавать мигрированные изображения в частотной области. Миграция Столта в настоящее время с большим отрывом является самым быстрым методом миграции [8]. В более общем виде данный метод будет рассматриваться в главе «3D FK миграция». Рассмотрим основные положения, характерные для миграции Столта в пространственно-временной плоскости.

Миграция f - k возникает из линейного волнового уравнения при применении преобразований Фурье.

Пусть $\psi(x, z, t)$ скалярное волновое поле согласно ERM в линейном двумерном волновом уравнении. Требуется найти волновое поле в момент «взрыва», т.е. $\psi(x, z, t = 0)$, зная волновое поле на поверхности среды $\psi(x, z = 0, t)$. Теперь обозначим $\phi(k_x, z, f)$ как Фурье преобразование от $\psi(x, z, t)$ по (x, t) . Тогда:

$$\psi(x, z, t) = \iint_{-\infty}^{+\infty} \phi(k_x, z, f) e^{2\pi i(k_x x - f t)} dk_x df \quad (2.4)$$

где k_x обозначает пространственное волновое число, связанное с x , и f – временную частоту. Для постоянной скорости распространения волны $\hat{c}$ (по модели ERM) применение преобразования Фурье к волновому уравнению приводит к уравнению Гельмгольца:

$$\frac{\partial^2 \phi}{\partial z^2} + 4\pi^2 \hat{k}_z^2 \phi = 0 \quad (2.5)$$

Здесь волновое число $\hat{k}_z$ представлено как:

$$\hat{k}_z^2 = \frac{f^2}{\hat{c}^2} - k_x^2 \quad (2.6)$$

Единственное граничное условие для уравнения (2.5) это $\phi(k_x, 0, f)$ представляющее преобразование Фурье от $\psi(x, z = 0, t)$. Чтобы устранить неточности, вызываемые использованием ERM модели, потребуем, чтобы $\psi(x, z, t)$ описывала только волны, движущиеся вверх (в сторону приемника). Таким образом, волновому полю позволено распространяться только в направлении $-z$ так, будто от целей исходят первичные отражения. Таким образом, можно решить волновое уравнение и получить мигрированное волновое поле.

$$\psi(x, z, 0) = \iint_{-\infty}^{+\infty} \phi(k_x, 0, f) e^{2\pi i(k_x x - f t)} dk_x df \quad (2.7)$$

Чтобы полноценно воспользоваться преобразованием Фурье, Столт предложил изменить переменную $\hat{k}_z^2$ введя:

$$f(\hat{k}_z) = \hat{c}^2 \operatorname{sign}(\hat{k}_z) \sqrt{k_x^2 + \hat{k}_z^2} \quad (2.8)$$

Это выражение описывает спектральное перестроение f-k-миграции Столта для PWI (Plane Wave Image – плоско-волновое изображение). С помощью замены переменных (2.8) получается окончательное решение для задачи миграции Столта:

$$\psi(x, z, 0) = \iint_{-\infty}^{+\infty} \frac{\hat{c}\hat{k}_z}{\sqrt{k_x^2 + \hat{k}_z^2}} \phi(k_x, 0, f(\hat{k}_z)) e^{2\pi i(k_x x - \hat{k}_z z)} dk_x d\hat{k}_z \quad (2.9)$$

Исходя из вида выражения (2.9), решение задачи миграции базируется на обратном преобразовании Фурье от

$$\frac{\hat{c}\hat{k}_z}{\sqrt{k_x^2 + \hat{k}_z^2}} \phi(k_x, 0, f(\hat{k}_z)) \quad (2.10)$$

Теперь у нас есть структура того, как выполнять f-k миграцию радиочастотных сигналов, представленных как PWI. Трехэтапный процесс преобразования нефокусированного изображения f-k методом можно описать следующим образом:

1. Сначала удаляются начальные нули радиочастотных сигналов из-за задержек излучения. Такое обрезание сигнала выравнивает гиперболы одинаковой кривизны, однако слегка искажает их, и больше при большом угле излучения.
2. Применяется алгоритм миграции Столта, описанный уравнениями (2.8) и (2.9) с использованием скорости распространения волны, соответствующей модели ERM. Отражатель расположенный в (x_s, z_s) будет перемещен в $(\hat{x}_s, \hat{z}_s)$ после миграции.
3. Выполняется координатное преобразование $(\hat{x}_s, \hat{z}_s) \rightarrow (x_s, z_s)$ чтобы вернуться к реальному положению отражателя [8].

В свободном пространстве волновое уравнение ограничивает распространение волн в пространстве и времени согласно

$$\left(\nabla^2 - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} \right) \psi = \nabla^2 \psi - \frac{1}{v^2} \frac{\partial^2 \psi}{\partial t^2} = 0 \quad (2.11)$$

где оператор Лапласа

$$\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.12)$$

Определяется пространственными переменными и скоростью света в среде, обычно задаваемой как $v = 3 \cdot 10^8$.

Предположим, для простоты, что эта поверхность сканирования плоская и находится в точке $z = 0$. Интерпретируем невидимое пространство под поверхностью в виде трехмерного объема, где каждая точка (или воксель в дискретном случае) излучает сферическую волну при $t = 0$. К сожалению, мы не можем напрямую получить это пространство, поэтому мы должны оценить его. Это порождает краевую задачу, требующую перейти от исходное поля с одним граничным условием к полю с другим граничным условием

$$\psi(x, y, z = 0, t) = \psi(x, y, z, t = 0) \quad (2.13)$$

Существует множество численных методов для решения волнового уравнения с некоторыми начальными условиями, включая метод конечных элементов и метод конечных разностей. Для однородной среды f-k миграция является самым быстрым из известных методов, обеспечивающих достаточно точное решение. Название метода предполагает, что f-k миграция связывает два граничных условия (см. уравнение (2.13)) в частотной области [9].

Чтобы понять этот подход, выразим зависящее от времени поле Ψ как суперпозицию плоских волн (выражаем ψ через спектр):

$$\psi(x, y, z, t) = \iiint \Phi(k_x, k_y, k_z) e^{2\pi i(k_x x + k_y y + k_z z - ft)} dk_x dk_y dk_z \quad (2.14)$$

Здесь $k = 2\pi \cdot (k_x, k_y, k_z)$ вектор, указывающий направление распространения каждой отдельной волны, а $|k| = 2\pi/\lambda$ для длины волны λ . Комплекснозначная функция Φ отображает амплитуду и фазу каждой плоской волны во времени $t = 0$. Волновой вектор k и частота f связаны дисперсионным соотношением:

$$f = v \sqrt{k_x^2 + k_y^2 + k_z^2} \quad (2.15)$$

Важное свойство уравнения (2.14) заключается в том, что функции Φ и ψ связаны преобразованием Фурье при $t = 0$. В качестве альтернативы можно выразить уравнение (2.14) как интеграл по k_x , k_y , и f . Это делается с помощью дисперсионного соотношения и его якобиана

$$\left| \frac{df}{dk_z} \right| = \frac{v|k_z|}{\sqrt{k_x^2 + k_y^2 + k_z^2}} \quad (2.16)$$

Для выполнения замены переменных

$$\psi(x, y, z, t) = \iiint \bar{\Phi}(k_x, k_y, f) e^{2\pi i(k_x x + k_y y + k_z z - ft)} dk_x dk_y f \quad (2.17)$$

Где

$$\Phi(k_x, k_y, k_z) = \frac{v|k_z|}{\sqrt{k_x^2 + k_y^2 + k_z^2}} \bar{\Phi}\left(k_x, k_y, v \sqrt{k_x^2 + k_y^2 + k_z^2}\right) \quad (2.18)$$

Это выражение для замены переменных представляет собой взвешенную одномерную интерполяцию, известную как интерполяция Столта [9].

Используя уравнения (2.14) – (2.18) приходим к выводу, что $f - k$ миграция представляет собой взятие функции $(x, y, z = 0, t)$ и следованию простым шагам для миграции исходного поля в поле $\psi(x, y, z, t = 0)$:

1. Трехмерное преобразование Фурье к измерениям
2. Применение взвешенной интерполяции к результирующему полю
3. Обратное трехмерное преобразование Фурье.

2.5. Миграция на основе применения методов интерполяции

Ранее было отмечено, что С-скан представляет собой набор А-сканов (измерений в точке), собранных в некоторой прямоугольной области на

поверхности среды. С-скан всего лишь содержит данные об электромагнитном поле в области сканирования с течением времени. В идеальном случае С-скан образуется при измерении электромагнитного поля в каждой точке области сканирования, однако зачастую это считается излишней и нецелесообразной работой по построению С-скана. В реальности, при использовании радара, в области сканирования получают два ортогональных друг другу набора В-сканов. В наборе каждый В-скан расположен на некотором расстоянии от соседних. Расстояние выбирают таким, чтобы между В-сканами могло разместиться несколько А-сканов. Для того, чтобы получить конечное изображение С-скана из получившихся В-сканов применяют различные методы интерполяции.

Если собранные данные представляют собой отсканированные в одном направлении В-сканы, то обычно применяется метод линейной интерполяции. Этот метод позволяет высчитать взвешенное арифметическое среднее из известных значений в точках прямоугольной проекции между двумя полосами сканирования. Данный метод полезен, в особенности при поиске линейных аномалий, расположенных перпендикулярно направлению сканирования [10].

Формула линейной интерполяции имеет вид [11]:

$$f(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} (x - x_i) \quad (2.19)$$

Для улучшения качества временных срезов, может быть выполнено двумерное сканирование, что даст более точные данные. В этом случае для каждой точки может быть применена билинейная интерполяция. Билинейная интерполяция представляет собой комбинацию двух независимых линейных интерполяций, полученных при перпендикулярных друг другу направлениях сканирования. Окончательное значение представляет собой среднее арифметическое, полученных в результате интерполяции в поперечном и продольном направлениях [10].

Рассмотрим случай билинейной интерполяции. Пусть известны значения в соседних точках Q_{11} , Q_{12} , Q_{21} , Q_{22} и требуется найти значение в некоторой точке P . При билинейной интерполяции первым делом интерполируется значение вспомогательных точек вдоль какой-либо из осей (например, вдоль оси абсцисс) [12]:

После чего проводится линейная интерполяция между вспомогательными точками R_1 и R_2 [12]:

$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad (2.21)$$

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (2.22)$$

После чего проводится линейная интерполяция между вспомогательными точками R_1 и R_2 [12]:

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (2.23)$$

Суть метода билинейной интерполяции отображена на рисунке 2.5.

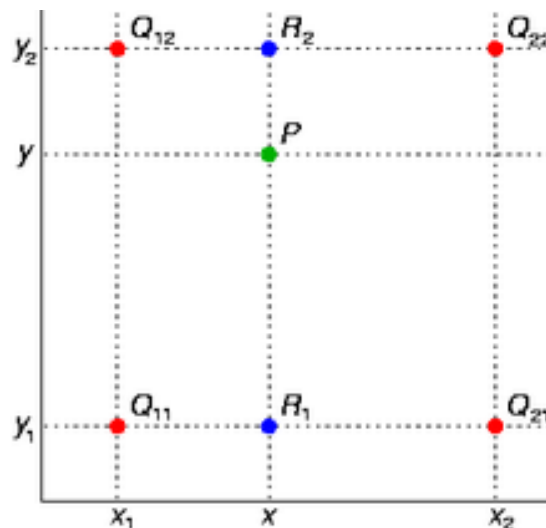


Рисунок 2.5 – схематическое изображение работы алгоритма билинейной интерполяции

Еще один простой подход из группы методов пространственной интерполяции это метод обратного взвешивания по расстоянию IDW (Inverse Distance Weighting), специально применяемый на рассеянных данных. Эта техника одна из наиболее распространенных для многомерной интерполяции во многих приложениях (например, в метеорологии, гидрологии, анализе изображении или георадарной томографии). Интерполируемое значение рассчитывается как среднее из набора значений v_i от конкретных точек используемых среднее арифметическое по следующей формуле:

$$v_{int} = \frac{\sum_{i=1}^N w_i v_i}{\sum_{i=1}^N w_i} \quad (2.24)$$

Где весовой коэффициент w_i рассчитывается в зависимости от расстояния d_i между интерполируемой точкой и каждой точек из рассматриваемого набора по следующей формуле:

$$w_i = \frac{1}{d_i^s} \quad (2.25)$$

Где s – обратный экспоненциальный множитель IEF (Inverse Exponential Factor), также называемый сглаживающим множителем. Если $s = 1$, весовой коэффициент обратно пропорционален расстоянию. При более высоких значениях s влияние соседних точек становится значительно более сильным в отличии от других. Коэффициент сглаживания обычно считается равным 2 [10].

Ключевым вопросом является правильный выбор точек для интерполяции. В общем случае, точки из всего набора данных могут участвовать в расчете каждого неизвестного значения; однако, такой подход неэффективен по времени. Решение этой проблемы заключается в исключении точек, достаточно сильно удаленных от каждой интерполируемой точки, поскольку будут иметь малые значения, согласно методу IDW. Один из методов предполагает поиск сканируемых точек в обозначенной области круга

определенного радиуса с центром в интерполированной точке. Для получения корректного изображения всей анализируемой области диаметр круга должен быть как минимум равен расстоянию между двумя соседними В-сканами. В противном случае несколько точек окажутся незатронутыми в результате интерполяции, что приведет к неполному и недостоверному изображению [10].

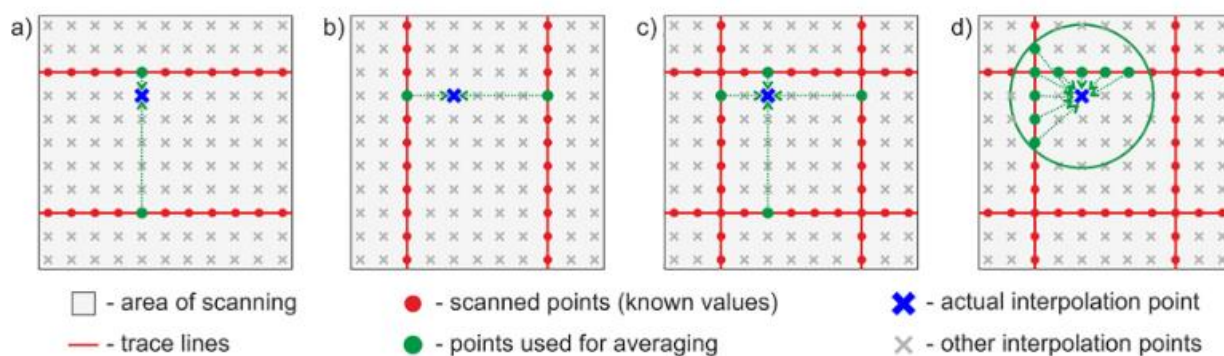


Рисунок 2.6 – различные способы интерполяции С-скана.

На рисунке 2.6 собраны все описанные выше методы интерполяции. Изображенные в пунктах а) и б) методы представляют собой применение алгоритма линейной интерполяции. Билинейная интерполяция отображена на изображении под пунктом с). Соответственно на рисунке d) изображено применение метода обратного взвешивания по расстоянию (IDW).

3. Разработка алгоритмов двумерной и трехмерной миграции.

Программная реализация алгоритмов

3.1. Алгоритмы гиперболической миграции

Для миграции двумерных несфокусированных изображений был разработан алгоритм (см. приложение А) на языке Python 3.0 и протестирован на исходных смоделированных данных для закопанного под среду цилиндрического отражателя (см. рис. 2.1 и 2.2). В результате миграции было получено изображение представленное на рисунке 3.1.

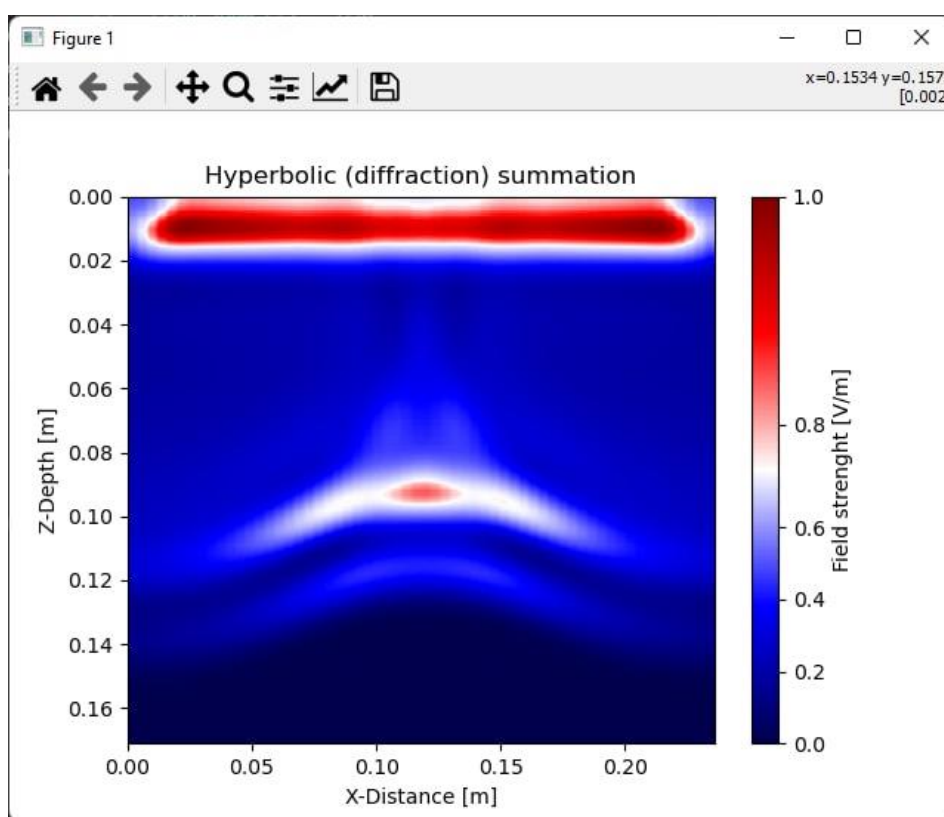


Рисунок 3.1 – результат двумерной гиперболической миграции

На изображении отчетливо видно красные объекты, соответственно представляющие собой разделяющую среды поверхность и закопанный под среду цилиндрический отражатель. Метод гиперболической суммации имеет два главных минуса. Первый заключается в том, что метод работает медленнее других с алгоритмической точки зрения, так как на каждый пиксель требуется найти и просуммировать множество других различных значений пикселей

изображения. Второй заключается в том, что на изображении возникают характерные эффекты, связанные с суммированием части гипербол от рассеянных волн в точках, не принадлежащих целевым отражателям, из-за чего на смигрированном изображении появляются аномалии похожие на различного рода кривые линии, с энергией поля близкой к энергии целевых отражателей.

Приведем реализацию алгоритма трехмерной гиперболической миграции в приложении Б. Рассмотрим визуальную интерпретацию гиперболической миграции для данных, полученных из модели закопанной под среду сферы (рисунок 3.2).

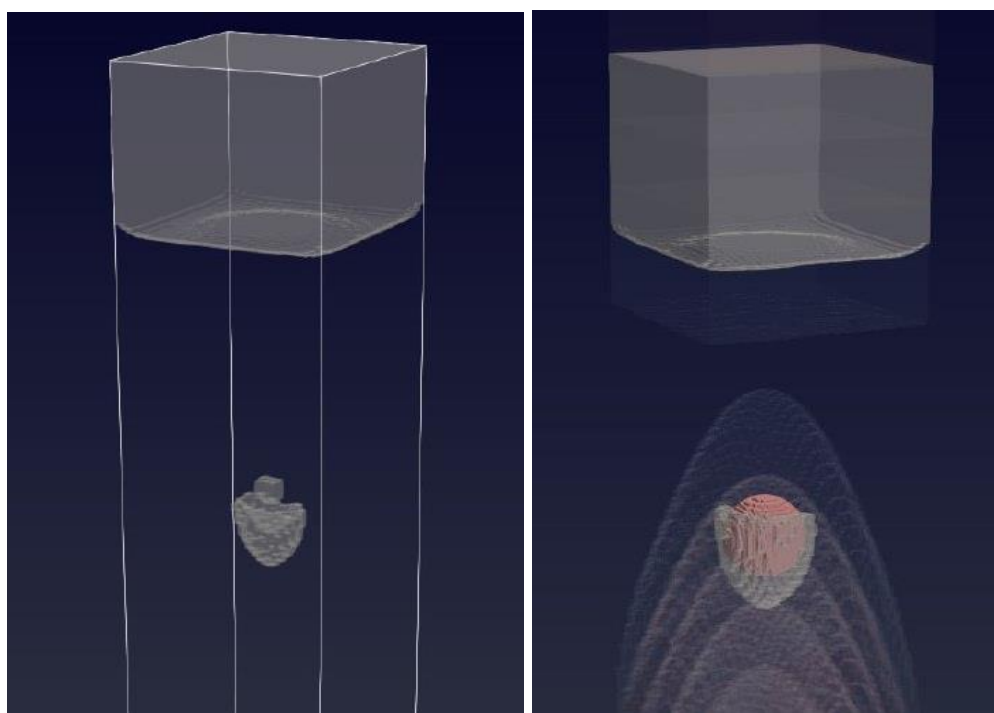


Рисунок 3.2 – результат работы метода трехмерной гиперболической миграции

На изображении слева можно видеть сфокусированное трехмерное изображение, состоящее из вокселей, интерпретирующих трехмерную действительность. На изображении слева отчетливо видно два отдельных объекта: поверхность среды (достаточно объемную из-за того, что основная часть энергии электромагнитных волн рассеивается именно в момент их

контакта с поверхностью среды) и некоторый объект закопанный под среду (исследуемая сфера). Объект не похож на сферу в верхней своей части, так как сумма гипербол в недостающих точках сферы достаточно мала (поскольку не захватывает отраженные от самой сферы волны). Такое изображение является корректным с точки зрения действия алгоритма. Изображение справа отображает собой отражение электромагнитных волн от сферической цели.

3.2. Алгоритмы миграции Столта

Основываясь на модели смоделированных данных для цилиндрического отражателя, закопанного под среду, реализуем алгоритм двумерной миграции Столта, использующий преобразования Фурье, чтобы получить сфокусированное изображение. Алгоритм миграции представлен в приложении В. В результате миграции было получено изображение, представленное на рисунке (3.3).

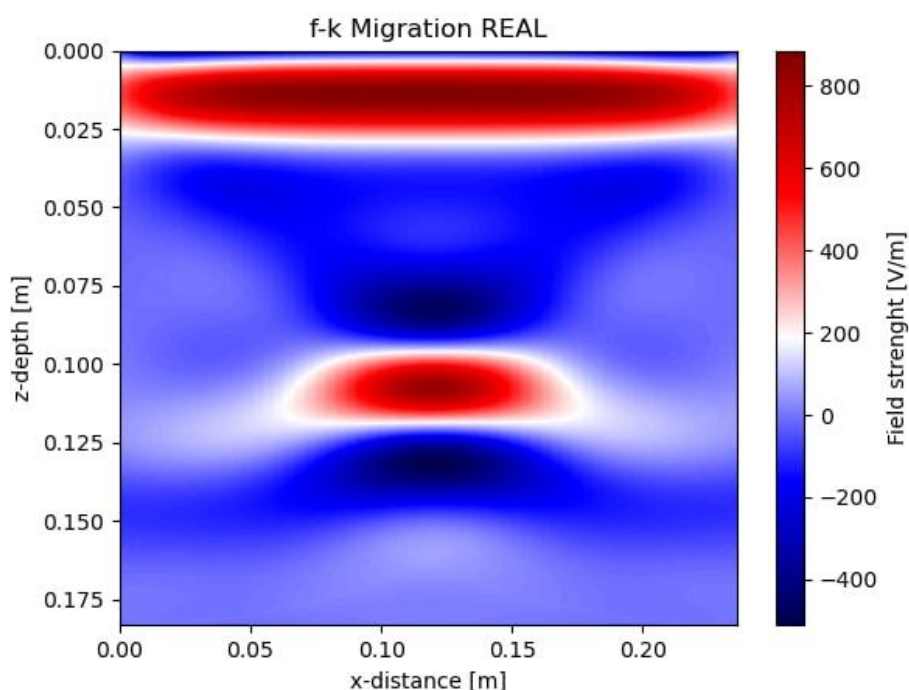


Рисунок 3.3 – результат двумерной миграции Столта

Реализуем алгоритм трехмерной миграции Столта (см. приложение Г) и приведем результат его работы для трехмерной модели закопанной под среду сферы (рисунок 3.4).

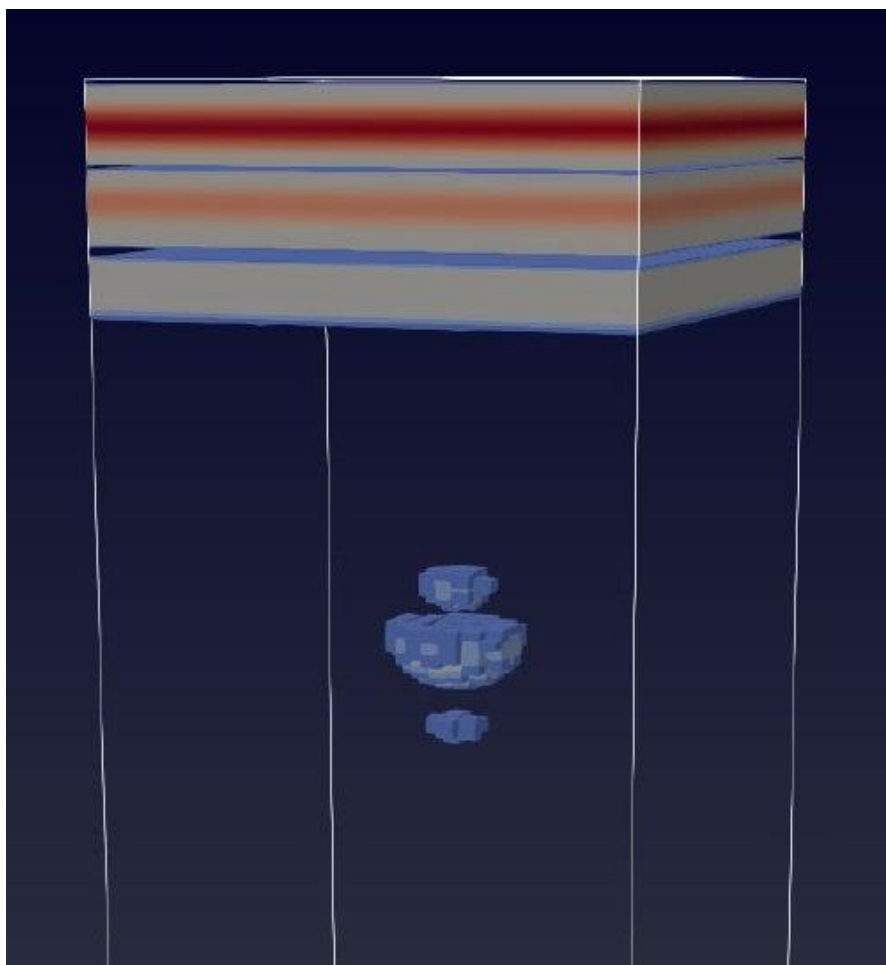


Рисунок 3.4 – результат работы метода f-k миграции

В отличие от гиперболической миграции, слои поверхности среды менее выражены, а целевой объект представлен в виде полусферы, от которой сверху и снизу расположены концентрированные пятна. Пятна представляют собой наложение переотраженных волн (точно так же происходит с поверхностью среды, из-за чего в самом верху отчетливо видно три плоских слоя, а не один цельный, причем каждый следующий слой по энергии волны меньше предыдущего), поэтому такое изображение с точки зрения работы алгоритма так же является корректным. Переотражения можно исправить, изменив параметры сканирующей антенны.

3.3. Алгоритмы интерполяции методом обратного взвешивания

Для демонстрации работы трехмерных методов миграции и построения интерполированного С-скана с помощью программы grgMax [5] были получены два набора В-сканов модели закопанной под среду сферы. Модель сферы изображена на рисунке (3.5)

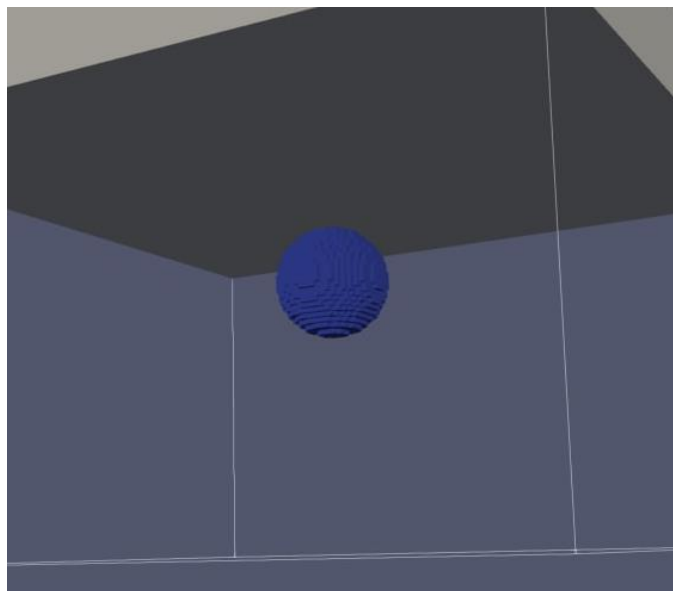


Рисунок 3.5 – изображение модели закопанной под среду сферы

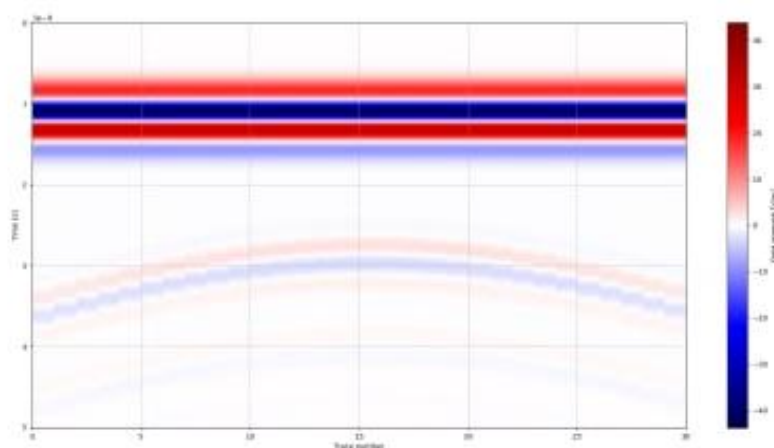


Рисунок 3.6 – один из В-сканов для модели сферы.

На В-скане закопанной сферы (рисунок 3.6) амплитуда отраженной от сферы волны менее ярко выраженная чем от поверхности среды. Это связано с глубиной, на которой расположена сфера.

Разработаем алгоритм линейной интерполяции по двум направлениям сканирования для данного набора В-сканов и представим результат интерполяции в виде нескольких срезов по глубине сканирования. Алгоритм интерполяции представлен в приложении Д.

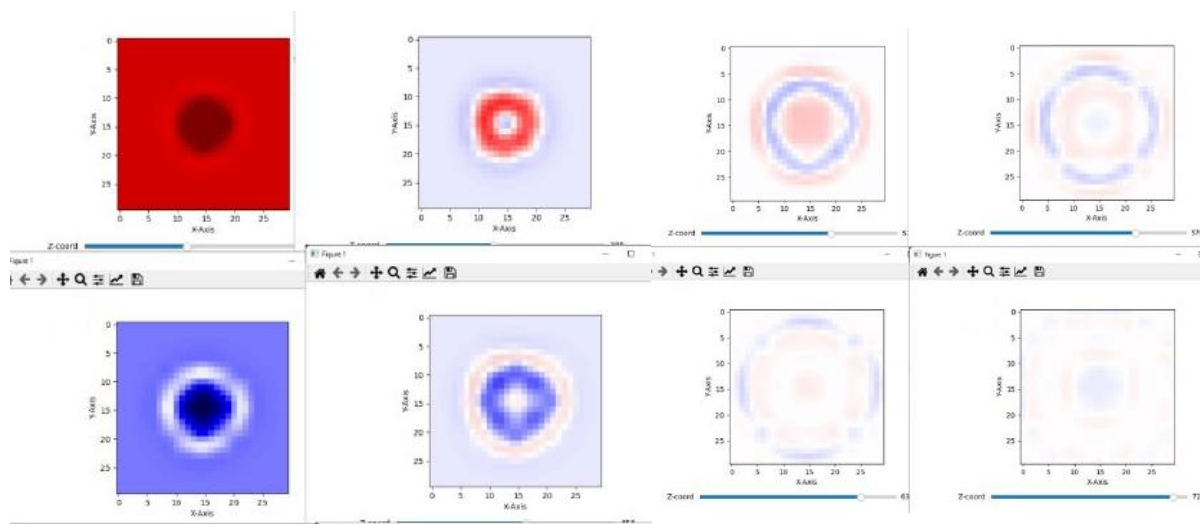


Рисунок 3.7 – Линейная интерполяция для построения С-скана.

На полученном изображении С-скана (рисунок 3.7) отчетливо видно, что на малой глубине сканирования, амплитуда отраженных волн принимает наибольшее значение (обозначается на графиках темным красным цветом). Именно на такой глубине сканирования находится раздел воздушной и исследуемой среды. Далее на более глубоких срезах можно видеть сферически рассеивающиеся волны, показывающие распространение электромагнитной волны, отраженной от цели, вплоть до полного затухания волн на некоторой глубине сканирования (вследствие потери всей энергии волны).

Итоговый массив трехмерных данных можно сохранить в любой удобный формат для использования его в трехмерных методах миграции.

Аналогично поступим и реализуем алгоритм обратного взвешивания по расстоянию (см. приложение Е.). Получились схожие изображения (рисунок 3.8) с изображениями, полученными с помощью линейной интерполяции.

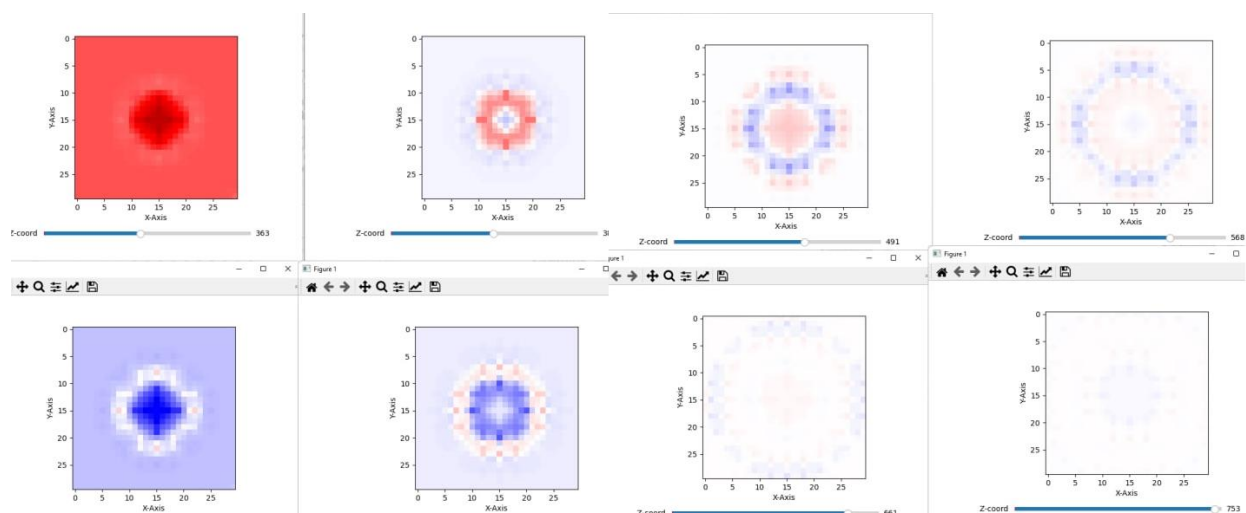


Рисунок 3.8 – Интерполяция методом IDW для получения C-скана.

4. Миграция реальных данных на основе разработанных алгоритмов

4.1. Формирование реальных данных

Рассмотрев методы гиперболической и FK миграции для двумерных и трехмерных несфокусированных изображений следует показать практическое применение этих методов. Для этого был подготовлен макет однородной среды, (см. рисунок 4.1) в которой друг за другом, в одном направлении расположены: стеклопластиковая трубка, пластиковая трубка, три арматуры, идущие друг за другом подряд, одиночная арматура и одиночная арматура большей толщины. Макет состоит из дерева, которое легко проводит электромагнитные волны. Так же на макете можно заметить мелкие детали, такие как крепежные элементы, с помощью которых крепятся трубки и арматуры, а также различной высоты болты, встроенные в дно макета.



Рисунок 4.1 – реальная модель для подготовки несфокусированного изображения.

Данный макет залили бетоном с диэлектрической проницаемостью приближенно равной шести. После чего с помощью георадара получили несколько В-сканов. Просканирован был не весь макет, а только часть макета от стеклопластиковой трубки до предпоследней арматуры. Шаг, с которым получали А-сканы составляет 1 мм, расстояние между В-сканами составляет 50 мм. Всего было сделано 11 В-сканов поперек расположению трубок и арматур и 10 В-сканов параллельно расположению трубок. Для одного из поперечных В-сканов было получено смигрированное изображение методом миграции Столта (см. рисунок 4.2).

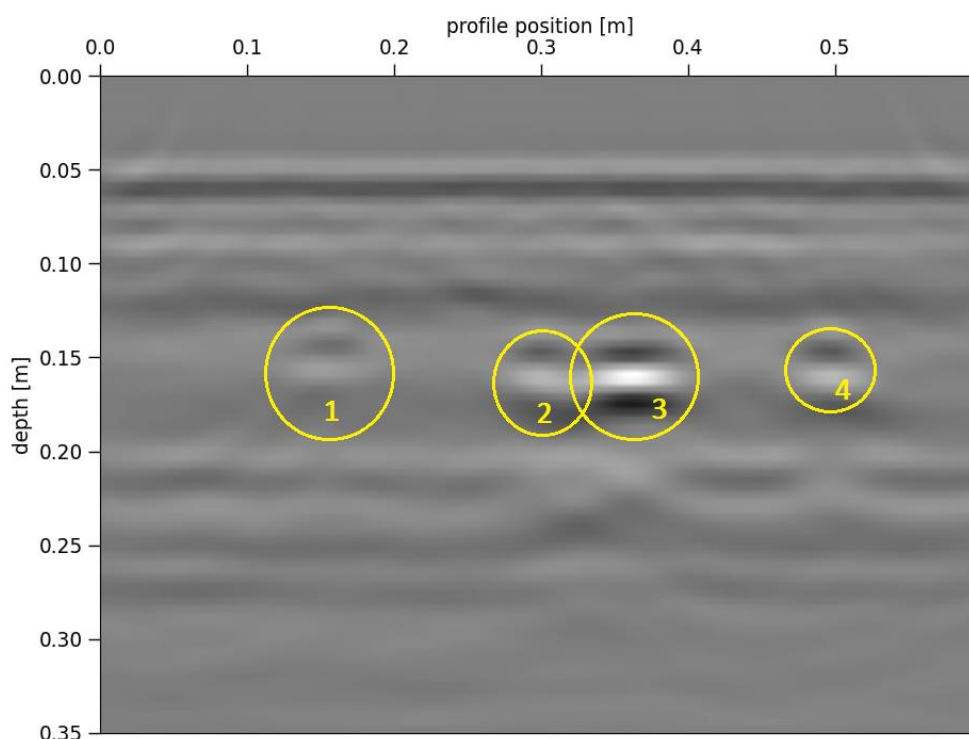


Рисунок 4.2 – миграция Столта на одном из В-сканов реального макета

На глубине 16-17 сантиметров отчетливо видны точки соответственно обозначающие отражатели макета (выделены желтым и обозначены числами. Здесь 1 – пластиковая трубка, 2 – одна из наиболее отдаленных арматур в группе из трех арматур, 3 – две арматуры, слившиеся на изображении в одну из-за низкой разрешающей способности радара, 4 – изображение одиночной арматуры.

4.2. Результаты применения миграции к реальным данным

Для выполнения трехмерной миграции Столта на исходных данных, данные были интерполированы с использованием линейной интерполяции по двум координатам (поскольку, интерполяция методом IDW очень затратна по времени). После чего для полученного С-скана был применен алгоритм миграции Столта. Полученное сфокусированное изображение изображено на рисунке 4.3.

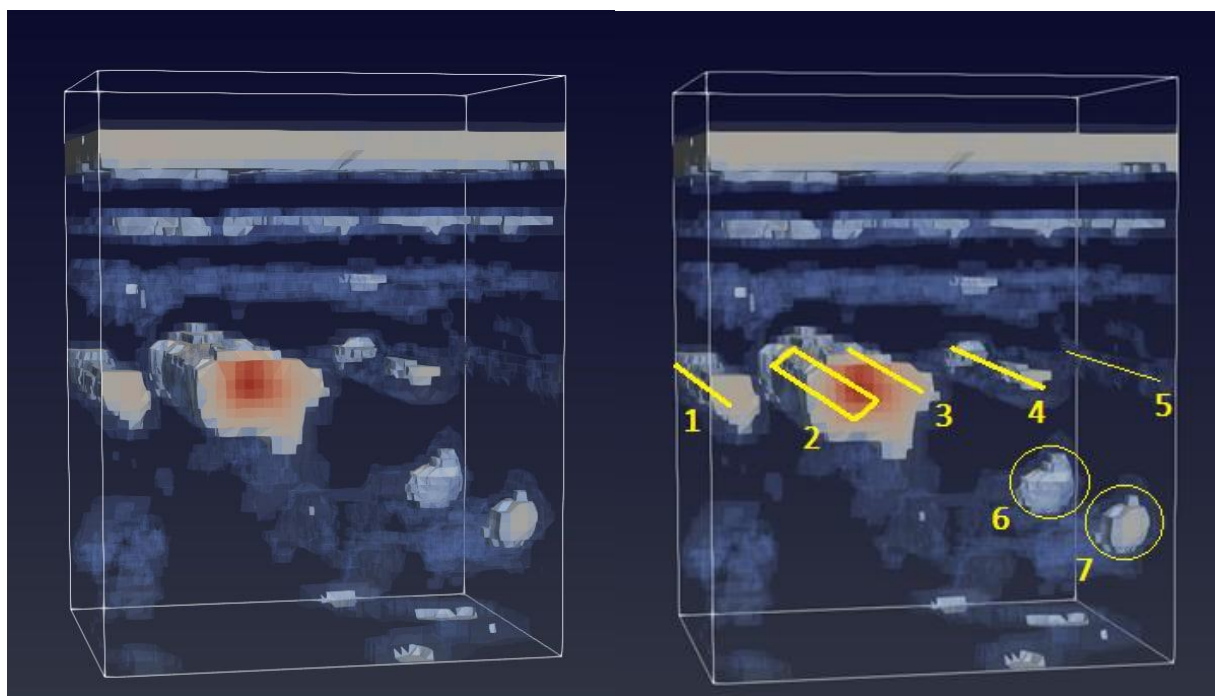


Рисунок 4.3 – сфокусированное изображение данных, представляющих реальный макет

Обозначим на рисунке 4.3 наиболее различимые объекты (справа). Цель номер 1 представляет собой пластиковую трубку на глубине 16-17 сантиметров. Цели номер два и три отображают сросшиеся вместе три арматуры. Однако из-за формы объекта, все еще можно увидитесь, что одна из арматур (цель 3) отделяется от общей группы. Цель под номером 4 представляет собой одиночную арматуру, а цель 5 представляет собой слабо заметную на изображении наиболее толстую арматуру. Кроме этих целей на изображении отчетливо видны цели 6 и 7 представляющие собой болты на дне

макета. К сожалению, определить количество и форму болтов по данному изображению не представляется возможным из-за низкой разрешающей способности радара. На изображении также видно поверхность среды и ее переотражения представляющие собой некоторые неоднородные слои с убывающей энергией сигнала. Неоднородность возникает из-за неоднородной и полостей среды. Кроме того под целями 1, 2, 3 видны переотражения волн. Таким образом можем убедиться в том, что полученное изображение является реальным представлением исходного макета.

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1. Введение

Раздел предполагает оценку конечного продукта в коммерческом и финансовом смысле, а также разработку плана по его развитию. Ценность продукта формируется за счет его превосходства над схожими товарами конкурентов и востребованности на рынке. Кроме того, раздел предполагает оценку возможных расходов и доходов, связанных с разработкой, продвижением и продажей продукта, оценку времени разработки продукта и срока его окупаемости.

Данный раздел, предусматривает рассмотрение следующих задач:

- Оценка коммерческого потенциала разработки.
- Составление плана разработки инженерного проекта;
- Расчет бюджета проекта;
- Оценка ресурсной, финансовой и бюджетной эффективности конечного продукта.

Цель данной ВКР – разработка алгоритмического и программного обеспечения георадара на основе применения методов миграции.

5.2. Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения

5.2.1. Анализ конкурентных технических решений

Технические решения прямо влияют на качественные характеристики конечного продукта. Поэтому перед разработкой продукта необходимо провести анализ конкретных технических решений и получить сравнительную эффективность разрабатываемого проекта. Кроме того, это поможет определить наиболее конкурентоспособную характеристику проекта, над

которой можно будет работать для повышения востребованности продукта на рынке.

В качестве конкурентов были выбраны:

1. «Proseq GPR» - ПО, поставляемое совместно с георадарным оборудованием компании Proseq. Популярно благодаря простоте интерфейса и эффективным внутренним алгоритмам, позволяющим формировать фокусированные изображения в реальном времени.
2. «GeoTechControl» - ПО, разработанное компанией GEOTECH, предназначенное для обработки и интерпретации данных, полученных сейсмоакустическим методом обследования строительных конструкций.

Рассмотрим оценочную карту конкурентоспособности для анализа технических и экономических критериев.

Таблица 5.1 – Оценочная карта конкурентоспособности.

Критерии оценки	Вес критерия	Баллы			Конкуренто-способность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
Технические критерии оценки ресурсоэффективности							
1. Работа в real-time режиме	0,20	5	5	0	1,00	1,00	0,00
2. Скорость миграции	0,15	5	5	4	0,75	0,75	0,60
3. Построение 3D-изображений	0,10	4	5	5	0,40	0,50	0,50
4. Скорость фильтрации	0,10	4	4	4	0,40	0,40	0,40
5. Скорость интерполяции	0,05	5	4	4	0,25	0,20	0,20
6. Построение 2D-изображений	0,05	5	5	3	0,25	0,25	0,15
Экономические критерии оценки эффективности							
1. Стоимость продукта	0,15	4	2	2	0,60	0,30	0,30
2. Срок эксплуатации	0,10	4	4	4	0,40	0,40	0,40
3. Необходимость в финансировании	0,10	1	2	2	0,10	0,20	0,20
Итого	1,00	39	36	28	4,35	4,00	2,75

В данной таблице использовались следующие обозначения: B_{ϕ} – баллы, начисляемые разрабатываемому проекту; B_{k1} – баллы, начисляемые Proseq GPR; B_{k2} – баллы, начисляемые GeoTechControl;

Конкурентоспособность вычисляется по формуле: $K_i = \sum B_i$, где B – вес критерия. Рассчитанные значения соответственно равны $K_{\phi} = 4,15$ ед., $K_{k1} = 4,00$ ед., $K_{k2} = 2,75$ ед.

Анализ конкурентных решений выявил, что некоторые конкурентные проекты упускают реализацию алгоритмов, позволяющих получать трехмерные сфокусированные изображения, а также не опираются на качество представляемой визуальной информации. Некоторые из реализуемых функций конкурентов требуют больше времени на выполнение, чем это допустимо для приложений, обрабатывающих информацию в реальном времени. Скорее всего это связано с использованием устаревших технических решений.

Главным конкурентом в разработке ПО георадарного оборудования оказался проект Proseq GPR. Он набрал почти максимум по всем допустимым критериям, что говорит о возможности разработки похожего на данный проект продукта, но с использованием новейших технологических решений и с оптимизацией бизнес-процессов для снижения стоимости и времени разработки проекта.

5.2.2. SWOT-анализ

Проведение такого типа анализа позволит выявить наиболее сильные и слабые стороны проекта, вычислить потенциальные возможности и угрозы, связанные с его разработкой.

Таблица 5.2 – Матрица SWOT-анализа

Сильные стороны	Слабые стороны
С1. Низкая стоимость разработки	Сл1. Узкий функционал

C2. Высокая скорость алгоритмов	Сл2. Низкая скорость обработки больших объемов данных
C3. Повышенный интерес на рынке	Сл3. Необходимость создания входных данных для тестирования
C4. Высокое качество выходных данных	Сл4. Нестабильность работы
C5. Обработка данных реальном времени	Сл5. Необходимость в доработке
Возможности	Угрозы
B1. Увеличение команды разработчиков	У1. Поломка георадарного оборудования с последующей невозможностью создания входных данных
B2. Применение современных технологий обработки информации	У2. Увеличение архитектуры и сложности проекта
B3. Использование научной литературы по реализации популярных миграционных методов	
B4. Использование совместно нескольких языков программирования во время разработки и их сильных и слабых сторон	
B5. Использование низкоуровневых языков программирования	

Разработка интерактивной матрицы проекта дает возможность определить какие сильные стороны или слабые проекта наиболее соответствуют возможностям и угрозам в реализации продукта. Соотношения параметров представлены в таблицах 5.3–5.6.

Таблица 5.3 – Интерактивная матрица проекта «Возможности проекта и сильные стороны»

Сильные стороны проекта						
Возможности проекта		C1	C2	C3	C4	C5
	B1	-	+	-	+	+
	B2	-	+	-	+	+
	B3	-	+	-	+	+
	B4	-	+	-	+	-
	B5	-	+	-	+	+

Таблица 5.4 – Интерактивная матрица проекта «Возможности проекта и слабые стороны»

Слабые стороны проекта						
Возможности проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	B1	-	-	-	+	+
	B2	-	+	-	+	+
	B3	-	-	-	-	-
	B4	-	+	-	+	+
	B5	-	+	-	-	+

Таблица 5.5 – Интерактивная матрица проекта «Угрозы проекта и сильные стороны»

Сильные стороны проекта						
Угрозы проекта		C1	C2	C3	C4	C5
	Y1	-	+	+	+	+
	Y2	-	-	+	-	-

Таблица 5.6 – Интерактивная матрица проекта «Угрозы проекта и слабые стороны»

Слабые стороны проекта						
Угрозы проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	Y1	-	-	+	-	-
	Y2	-	+	-	+	+

Результаты анализа представлены в итоговую таблицу 5.7.

Таблица 5.7 – Итоговая таблица SWOT-анализа

	Сильные стороны научно-исследовательского проекта	Слабые стороны научно-исследовательского проекта
	С1. Низкая стоимость разработки С2. Высокая скорость алгоритмов С3. Повышенный интерес на рынке	Сл1. Узкий функционал Сл2. Низкая скорость обработки больших объемов данных

	С4. Высокое качество выходных данных С5. Обработка данных реальном времени	Сл3. Необходимость создания входных данных для тестирования Сл4. Нестабильность работы Сл5. Необходимость в доработке
Возможности В1. Увеличение команды разработчиков В2. Применение современных технологий обработки информации В3. Использование научной литературы по реализации популярных миграционных методов В4. Использование совместно нескольких языков программирования во время разработки и их сильных и слабых сторон В5. Использование низкоуровневых языков программирования	Направления развития В1В2В3В4В5С2С4. Высокая скорость алгоритмов и высокое качество выходных данных достигается за счет большой и опытной команды разработчиков, использования новейших технологических решений, научной литературы, совместного использования языков программирования, включая низкоуровневые. В1В2В3В5С5. Для обработки данных в реальном времени не обязательно использование нескольких языков программирования, однако все остальные сильные стороны проекта используются для достижения этого результата.	Сдерживающие факторы В2В4В5Сл2. Применение новых методов обработки информации и совместное использование высокоуровневых и низкоуровневых языков программирования позволит разрабатывать обеспечение с максимально эффективной скоростью обработки информации. В1В2В4Сл4. Расширение команды разработчиков поможет быстрее выявлять узкие места в работе продукта, а применение новых методов и совместного использования языков уменьшит неявное поведение программы, т.к. современные технологии проектируются с учетом минимизации неявного поведения. В1В2В4В5Сл5. Расширение команды разработчиков и применение технологических сильных сторон проекта позволят быстрее решать проблемы связанные с появлением необходимости внесения нового функционала. Современные технологии разрабатываются максимально гибкими, чтобы как можно быстро вносить корректировки в программу. Использование низкоуровневых языков позволит дорабатывать низкопроизводительные части кода программы.
Угрозы У1. Поломка георадарного оборудования с последующей невозможностью	Угрозы развития У1С2С3С4С5. При условии поломки георадарного оборудования, это лишь замедлит время разработки продукта и повлечет за собой дополнительные финансовые	Уязвимости: У1Сл3. Поломка георадарного оборудования вызовет необходимость получения входных данных до его починки. Фактически это приведет к затратам на

создания входных данных. У2. Увеличение архитектуры и сложности проекта	расходы. На другие сильные стороны поломка оборудования никак не влияет. У2С3. Увеличение сложности архитектуры программы не только замедляет и повышает стоимость разработки программы, но и увеличивает количество возникающих ошибок, вследствие которых страдает качество кода и основных технических сильных сторон проекта, как скорость работы алгоритмов и качество выходных данных. На интерес на рынке данная угроза никак не влияет.	моделирование входных данных или заморозке разработки. У2Сл2Сл4Сл5. Увеличение архитектуры программы может существенно повлиять на качество кода в худшую сторону, а значит привести к уменьшению скорости работы программы и ошибкам. Кроме того, появится необходимость в исправлении всех возникших ошибок командой разработки.
--	--	---

SWOT-анализ показал, что у разрабатываемого проекта гораздо больше преимуществ, чем недостатков. Слабые стороны проекта нивелируются его возможностями. Единственной существенной угрозой во время разработки ПО является поломка георадарного оборудования, способная увеличить стоимость развития проекта.

5.3. Планирование работ

5.3.1. Структура работ в рамках инженерного проекта

Планирование предполагаемых работ производится в следующем порядке:

- определение структуры работ в рамках инженерного проекта;
- определение исполнителей по каждому виду работ;
- определение продолжительности работ;
- построение графика проведения работ.

Для разработки алгоритмического и программного комплекса георадара сформируем рабочую группу, которая будет состоять из разработчика, тестировщика и руководителя проекта. На таблице 5.8 представлен порядок выполнения работ разработки.

Таблица 5.8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб.	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания, утверждение плана-графика	Руководитель проекта
	2	Календарное планирование выполнения работ	Руководитель проекта
Обзор литературы по миграционным алгоритмам	3	Обзор литературы по миграционным алгоритмам	Разработчик
	4	Обзор технических решений в аналогичных проектах	Разработчик
Подготовительный этап	5	Подготовка входных данных для тестирования функционала	Тестировщик
Разработка алгоритма гиперболической миграции	6	Построение структурной схемы алгоритма гиперболической миграции	Разработчик
	7	Реализация алгоритма гиперболической миграции	Разработчик
	8	Отладка алгоритма гиперболической миграции	Тестировщик
Разработка алгоритма fk-миграции	9	Построение структурной схемы алгоритма fk-миграции	Разработчик
	10	Реализация алгоритма fk-миграции	Разработчик
	11	Отладка алгоритма fk-миграции	Тестировщик
Обобщение и оценка результатов	12	Оценка эффективности полученных алгоритмов, доработка выявленных недочетов	Разработчик, Тестировщик, Руководитель проекта
	13	Итоговое тестирование ПО	Тестировщик, Руководитель проекта
Подготовка документации по проекту	14	Составление технической документации по проекту	Разработчик, Руководитель проекта

5.3.2. Определение трудоемкости выполнения работ и разработка графика проведения

При разработке инженерных проектов определение трудоемкости проводимых работ позволит наиболее точно рассчитать временные и финансовые затраты проекта.

При определении ожидаемого (среднего) значения трудоемкости работ будем использовать формулу:

$$t_{\text{ож } i} = \frac{3t_{\text{min } i} + 2t_{\text{max } i}}{5}, \quad (5.1)$$

где $t_{\text{ож } i}$ – ожидаемая трудоемкость выполнения i -ой работы, человеко-дни.

$t_{\text{min } i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни.

$t_{\text{max } i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, человеко-дни.

Определить продолжительной каждой i -ой работы в рабочих днях, можно из ожидаемой трудоемкости с учетом параллельности выполнения работ разными исполнителями.

$$T_{p_i} = \frac{t_{\text{ож } i}}{\text{Ч}_i}, \quad (5.2)$$

где T_{p_i} – продолжительность одной работы, рабочие дни.

$t_{\text{ож } i}$ – ожидаемая трудоемкость выполнения одной работы, человеко-дни.

Ч_i – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Перевести длительность этапов из рабочих в календарные дни можно с помощью формулы:

$$T_{ki} = T_{p_i} \cdot k_{\text{кал}}, \quad (5.3)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – календарный коэффициент.

Календарный коэффициент определяется по формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = \frac{365}{365 - 104 - 14} = 1,48 \quad (5.4)$$

где $T_{\text{кал}}$ – общее количество календарных дней в году; $T_{\text{вых}}$ – общее количество выходных дней в году; $T_{\text{пр}}$ – общее количество праздничных дней в году (2023 год).

Расчеты временных показателей разработки инженерного проекта обобщены в таблице 5.9.

Таблица 5.9 – Временные показатели разработки инженерного проекта

Название работы	Трудоёмкость работ									Длительн ость работ в рабочих днях T_{pi}	Длительн ость работ в календарн ых днях T_{ki}
	t_{min} , чел-дни			t_{max} , чел-дни			$t_{ож}$, чел-дни				
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3		
1	2	3	4	5	6	7	8	9	10	11	12
1. Составление и утверждение технического задания, утверждение плана- графика	-	-	1	-	-	2	-	-	1,4	1,4	2
2. Календарное планирование выполнения работ	-	-	2	-	-	4	-	-	2,8	2,8	4

3. Обзор литературы по миграционным алгоритмам	5	-	-	12	-	-	7,8	-	-	7,8	12
4. Обзор технических решений в аналогичных проектах	5	-	-	12	-	-	7,8	-	-	7,8	12
5. Подготовка входных данных для тестирования функционала	-	2	-	-	5	-	-	3,2	-	3,2	5
6. Построение структурной схемы алгоритма гиперболической миграции	7	-	-	14	-	-	9,8	-	-	9,8	15
7. Реализация алгоритма гиперболической миграции	7	-	-	14	-	-	9,8	-	-	9,8	15
8. Отладка алгоритма гиперболической миграции	-	3	-	-	5	-	-	3,8	-	3,8	6
9. Построение структурной схемы алгоритма fk-миграции	7	-	-	14	-	-	9,8	-	-	9,8	15
10. Реализация алгоритма fk-миграции	7	-	-	14	-	-	9,8	-	-	9,8	15
11. Отладка алгоритма fk-миграции	-	3	-	-	5	-	-	3,8	-	3,8	6

12. Оценка эффективности полученных алгоритмов, доработка выявленных недочетов	7	7	7	14	14	14	9,8	9,8	9,8	9,8	15
13. Итоговое тестирование ПО	-	1	1	-	2	2	-	1,4	1,4	1,4	2
14. Составление технической документации по проекту	3	-	3	7	-	7	4,6	-	4,6	4,6	7
Итого:	48	16	14	101	31	29	69,2	22	20	85,6	126

Примечание: Исп. 1 – разработчик, Исп. 2 – тестировщик, Исп. 3 – руководитель проекта

На основе таблицы составлен календарный план-график выполнения проекта с использованием диаграммы Ганта (таблица 5.10).

Таблица 5.10 – Диаграмма Ганта

№	Вид работ	Исп	T_{ki} кал. Дн.	Продолжительность работ											
				февр			март			апр			май		
				1	2	3	1	2	3	1	2	3	1	2	3
1	Составление и утверждение технического задания, утверждение плана-графика	Исп3	2												
2	Календарное планирование выполнения работ	Исп3	4												
3	Обзор литературы по миграционным алгоритмам	Исп1	12												
4	Обзор технических решений в аналогичных проектах	Исп1	12												
5	Подготовка входных данных для тестирования функционала	Исп2	5												

№	Вид работ	Исп	T_{ki} , кал. Дн.	Продолжительность работ											
				февр			март			апр			май		
				1	2	3	1	2	3	1	2	3	1	2	3
6	Построение структурной схемы алгоритма гиперболической миграции	Исп1	15												
7	Реализация алгоритма гиперболической миграции	Исп1	15												
8	Отладка алгоритма гиперболической миграции	Исп2	6												
9	Построение структурной схемы алгоритма fk-миграции	Исп1	15												
10	Реализация алгоритма fk-миграции	Исп1	15												
11	Отладка алгоритма fk-миграции	Исп2	6												
12	Оценка эффективности полученных алгоритмов, доработка выявленных недочетов	Исп1 Исп2 Исп3	15												
13	Итоговое тестирование ПО	Исп2 Исп3	2												
14	Составление технической документации по проекту	Исп1 Исп3	7												

Примечание:

■ – Исп. 1 (разработчик), ▨ – Исп. 2 (тестировщик), ▩ – Исп. 3 (руководитель проекта)

5.4. Бюджет инженерного проекта

Все расходы, связанные с реализацией проекта, отражены в бюджете инженерного проекта. Сгруппируем затраты по следующим статьям:

- материальные затраты по инженерному проекту;
- затраты на специальное оборудование для выполнения работ;
- основная заработная плата исполнителей;

- дополнительная заработная плата исполнителей;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

5.4.1. Расчет амортизации специального оборудования

Расчет амортизации проводится следующим образом:

Норма амортизации рассчитывается по формуле:

$$H_A = \frac{1}{n}, \quad (5.5)$$

где n – срок полезного использования в количестве лет.

Амортизация оборудования рассчитывается по формуле:

$$A = \frac{H_A \cdot И}{12} \cdot t, \quad (5.6)$$

где $И$ – итоговая сумма, тыс. руб.; t – время использования, мес.

Инженерный проект подразумевает под собой разработку только продукта интеллектуального труда. Соответственно затраты на сырьё, комплектующие и различные материалы будут отсутствовать. В ходе разработки будет применяться только специальное оборудование.

Таблица 5.11 - Затраты на специальное оборудование

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, руб.	Общая стоимость оборудования, тыс.руб.
1.	Ноутбук Acer Aspire 3 A314- 23P-R3KH	1	43 000	43 000
2.	Ноутбук Acer Aspire 3 A315- 23	1	40 000	40 000

3.	Ноутбук Acer Aspire 3 A315-24P-R701	1	38 000	38 000
4.	Операционная система - Windows Pro	3	30 000	90 000

Поскольку рабочая группа уже обладает ноутбуками для проведения необходимых работ, в статье будут отражены только затраты на амортизационные отчисления. С учетом реального срока работы ноутбуков серии Acer (в среднем 8 лет), следует провести перерасчет амортизации на время разработки проекта (3,3 месяца).

Норма амортизационных отчислений на приобретение лицензии Windows Pro:

$$H_A = \frac{1}{8} \cdot 100 = 12,5\%$$

Из этого следует, что размер амортизации составит:

$$A_{\text{лиц}} = \frac{30\,000 \cdot 3}{12} \cdot 0,125 \cdot 3,3 = 3093,75 \text{ руб.}$$

Затраты на установку ОС на 3 ноутбука:

$$A_{\text{устан}} = 7500 \text{ рублей}$$

Используемые языки программирования являются открытыми и бесплатными для разработки коммерческих проектов.

Проведем амортизацию оборудования линейным способом. Гарантийный срок использования ноутбуков серии Acer составляет 3 года. Получим размер амортизационных отчислений за специальное оборудование (компьютеры) в ходе разработки инженерного проекта:

$$A_{\text{об}} = \frac{121\,000}{12} \cdot \frac{1}{3} \cdot 3,3 = 11091,67 \text{ руб.}$$

Итоговая сумма затрат, связанных с амортизацией оборудования:

$$З_{об} = A_{устан} + A_{лиц} + A_{об} = 7500 + 3093,75 + 11091,67 = 21685,42 \text{ руб.}$$

5.4.2. Основная заработная плата исполнителей темы

Раздел включает в себя расчет заработной платы исполнителей проекта, а именно программиста, тестировщика и руководителя. Расчет заработной платы в данной статье определяется исходя из трудоемкости проекта и рыночной цены труда по рассматриваемым профессиям.

Основная заработная плата $З_{осн}$ одного работника рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot T_p, \quad (5.7)$$

где $З_{дн}$ – среднедневная заработная плата, руб.; T_p – продолжительность работ, выполняемых работником, раб.дн. (таблица 5.9).

Месячный должностной оклад работника вычисляется по формуле:

$$З_m = З_{мс} \cdot (1 + k_{пр} + k_{\partial})k_p, \quad (5.8)$$

где $З_{мс}$ – средний оклад исполнителя по г. Томску, руб.; $k_{пр}$ – премиальный коэффициент, равен 0,3; k_{∂} – коэффициент доплат и надбавок, равен 0,2; k_p – районный коэффициент, равен 1,3 (для г. Томска).

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_m \cdot M}{F_{\partial}}, \quad (5.9)$$

где $З_m$ – месячный должностной оклад работника, руб.; F_{∂} – действительный годовой фонд рабочего времени научно-технического персонала, раб. дней; M – количество месяцев работы без отпуска в течение года:

– при отпуске в 28 раб. дня – $M = 11,2$ месяца;

– при отпуске в 56 раб. дней – М = 10,3 месяца неделя.

Произведем расчет месячных должностных окладов исполнителей опираясь на статистику портала hh.ru для г. Томска.

Месячный должностной оклад программиста:

$$З_{\text{м}} = 40\,000 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 78\,000 \text{ руб.}$$

Месячный должностной оклад тестировщика:

$$З_{\text{м}} = 27\,500 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 53\,625 \text{ руб.}$$

Месячный должностной оклад руководителя проекта:

$$З_{\text{м}} = 40\,500 \cdot (1 + 0,3 + 0,2) \cdot 1,3 = 78\,975 \text{ руб.}$$

Произведем расчет дневной заработной платы программиста (5-дневная рабочая неделя):

$$З_{\text{дн}} = \frac{x \cdot 11,2}{213} = 4101,41 \text{ руб.}$$

Произведем расчет дневной заработной платы тестировщика (5-дневная рабочая неделя):

$$З_{\text{дн}} = \frac{x \cdot 11,2}{213} = 2819,72 \text{ руб.}$$

Произведем расчет дневной заработной платы руководителя проекта (5-дневная рабочая неделя):

$$З_{\text{дн}} = \frac{x \cdot 11,2}{213} = 4152,68 \text{ руб.}$$

В таблице 5.12 обозначены показатели рабочего времени, в зависимости от вида рабочей недели. В таблице 5.13 представлен расчет основной заработной платы исполнителей проекта.

Таблица 5.12 – Баланс рабочего времени исполнителей

Показатели рабочего времени	Научный руководитель	Разработчик
Календарное число дней	365	365
Количество нерабочих дней - выходные дни - праздничные дни	52/14	104/14
Потери рабочего времени - отпуск - невыходы по болезни	48/5	24/10
Действительный годовой фонд рабочего времени	246	213

Таблица 5.13 – Расчет основной заработной платы исполнителей

Исполнители НИ	З _{мс} , руб	k _{пр}	k _д	k _р	З _м , руб	З _{дн} , руб	Т _р , раб. дн.	З _{осн} , руб
Программист	40 000	0,3	0,2	1,3	78 000	4101,41	69,2	283817,57
Тестирующий	27 500	0,3	0,2	1,3	53 625	2819,72	22,0	62033,84
Руководитель проекта	40 500	0,3	0,2	1,3	78 975	4152,68	20,0	83053,60
Итого:								428905,01

Дополнительная заработная плата определяется по формуле:

$$З_{\text{доп}} = k_{\text{доп}} \cdot З_{\text{осн}}, \quad (5.10)$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (принимается равным 0,15).

Произведем расчет дополнительной заработной платы для исполнителей:

– для программиста:

$$З_{\text{доп}} = 0,15 \cdot 283817,57 = 42572,64 \text{ руб.}$$

– для тестировщика:

$$З_{\text{доп}} = 0,15 \cdot 62033,84 = 9305,08 \text{ руб.}$$

– для руководителя проекта:

$$З_{\text{доп}} = 0,15 \cdot 83053,60 = 12458,04 \text{ руб.}$$

5.4.3. Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды определяется по формуле:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}), \quad (5.11)$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд ОМС и социальное страхование). Общая ставка взносов составляет в 2020 году – 30% (ст. 425, 426 НК РФ).

Определим отчисления во внебюджетные фонды для исполнителей по проекту:

– для программиста:

$$З_{\text{внеб}} = 0.3 \cdot 326390,20 = 97917,06 \text{ руб.}$$

– для тестировщика:

$$З_{\text{внеб}} = 0.3 \cdot 71338,92 = 21401,67 \text{ руб.}$$

– для руководителя проекта:

$$З_{\text{внеб}} = 0.3 \cdot 95511,64 = 28653,49 \text{ руб.}$$

5.4.4. Накладные расходы

Разработка инженерных проектов часто сопровождается накладными расходами в виде оплаты услуг связи, электроэнергии, почтовых и телеграфных расходов и т.д. Расчет накладных расходов производится по формуле:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}), \quad (5.12)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величина коэффициента накладных взята в размере 16%.

Произведем расчет начислений по накладным расходам:

$$З_{\text{накл}} = (147972,20 + 428905,01 + 21685,42) \cdot 0,16 = 95770,03 \text{ руб.}$$

5.4.5. Бюджет инженерного проекта

На рассмотренным статьям затрат составляется общая плановая себестоимость инженерного проекта, касающегося разработки алгоритмического и программного обеспечения георадара. Сгруппированные затраты отражены в форме, приведенной в таблице 5.14, вместе с оценкой затрат по аналогичным статьям для конкурентных проектов.

Таблица 5.14 – Группировка затрат по статьям

№	Наименование статьи	Сумма, руб.			Примечание
		Текущий Проект	Proseq GPR	Geo Tech Control	
1	Материальные затраты инженерного проекта	-	-	-	-
2	Затраты на специальное оборудование	21685,42	200000	150000	Пункт 5.3.1
3	Затраты по основной заработной плате исполнителей темы	428905,01	2432742	1843732	Пункт 5.3.2
4	Затраты по дополнительной заработной плате исполнителей темы	64335,76	364911	276560	-
5	Отчисления во внебюджетные фонды	147972,22	447624	339246	Пункт 5.3.3
6	Накладные расходы	95770,03	895249	678493	Пункт 5.3.4
Бюджет затрат НИР		758668,4	4340526	3288031	

5.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности проекта

Оценена эффективность проекта на основе интегрального показателя эффективности, полученного из интегральных показателей финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности инженерного проекта основан на информации о затратах трех конкурирующих проектов. За базу расчета берется наибольший интегральный показатель реализации технической задачи, с которым соотносятся финансовые значения по всем версиям исполнения.

В качестве аналогов данного инженерного проекта:

- 1) «Proseq GPR»;
- 2) «GeoTechControl».

Интегральный финансовый показатель разработки рассчитывается следующим образом:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (5.13)$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения.

$\Phi_{\text{текущ.проекта}} = 758668,4$ руб,

$\Phi_{\text{исп.1}} = 4340526$ руб,

$\Phi_{\text{исп.2}} = 3288031$ руб.

Произведем расчет финансовых показателей разработки:

$$I_{\text{финр}}^{\text{тек.пр.}} = \frac{\Phi_{\text{тек.пр.}}}{\Phi_{\text{max}}} = \frac{758668,4}{4340526} = 0.175$$

$$I_{\text{финр}}^{\text{исп.1}} = \frac{\Phi_{\text{исп.1}}}{\Phi_{\text{max}}} = \frac{4340526}{4340526} = 1.000$$

$$I_{\text{финр}}^{\text{исп.2}} = \frac{\Phi_{\text{исп.2}}}{\Phi_{\text{max}}} = \frac{3288031}{4340526} = 0.758$$

Исходя из полученных результатов, можно отметить, что текущий проект в несколько раз эффективнее своих конкурентов с позиции финансовой эффективности.

Интегральный показатель ресурсоэффективности вариантов реализации проекта (I_{pi}) исходит из сравнительной оценки взвешенных характеристик конкурентных проектов (таблица 5.15).

Таблица 5.15 – Сравнительная оценка характеристик вариантов проекта

Объект исследования Критерии	Весовой коэффициент параметра	Текущий проект	Исп.1	Исп.2
1. Работа в real-time режиме	0,35	10	10	0
2. Скорость миграции	0,25	10	10	8
3. Построение 3D-изображений	0,15	10	10	10
4. Скорость фильтрации	0,15	8	8	8
5. Скорость интерполяции	0,1	10	8	8
6. Построение 2D-изображений	0,1	10	10	6
ИТОГО	1	9,7	9,5	5,5

Расчет интегрального показателя для разрабатываемого проекта:

$$I_{p1} = 0,35 * 10 + 0,25 * 10 + 0,15 * 10 + 0,15 * 8 + 0,1 * 10 + 0,1 * 10 = 9,7$$

$$I_{p2} = 0,35 * 10 + 0,25 * 10 + 0,15 * 10 + 0,15 * 8 + 0,1 * 8 + 0,1 * 10 = 9,5$$

$$I_{p3} = 0,35 * 0 + 0,25 * 8 + 0,15 * 10 + 0,15 * 8 + 0,1 * 8 + 0,1 * 6 = 5,5$$

Интегральный показатель эффективности вариантов исполнения разработки вычисляется исходя из показателя ресурсоэффективности и интегрального финансового показателя согласно:

$$I_{\text{исп.}i} = \frac{I_{p-\text{исп.}i}}{I_{\text{финр}}}, \quad (5.14)$$

$$I_{\text{текущ.проекта}} = \frac{9,7}{0,175} = 55,4; I_{\text{исп.2}} = \frac{9,5}{1,000} = 9,5; I_{\text{исп.3}} = \frac{5,5}{0,758} = 7,3$$

Сравнительную эффективность проекта можно вычислить, используя интегральные показатель эффективности вариантов исполнения. С помощью показателя сравнительной эффективности (таблица 5.16) можно выбрать наиболее целесообразный вариант исполнения проекта.

Сравнительная эффективность проекта ($\mathcal{E}_{\text{ср}}$):

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп.1}}}{I_{\text{исп.2}}}, \quad (5.15)$$

Таблица 5.16 – Сравнительная эффективность разработки

№ п/п	Показатели	Текущий проект	Исп.1	Исп.2
1	Интегральный финансовый показатель разработки	0,175	1	0,758
2	Интегральный показатель ресурсоэффективности разработки	9,7	9,5	5,5
3	Интегральный показатель эффективности	55,4	9,5	7,3
4	Сравнительная эффективность вариантов исполнения	1	0,1715	0,1318

Из таблицы видно, что текущий проект эффективнее своих конкурентов. Это связано в первую очередь с тем, что для разработки решения не привлекается большая команда разработки и затраты на проект гораздо

меньше, чем у конкурентов, при этом качество проекта остается схожим за счет использования всех современных технологий разработки ПО.

5.6. Выводы по разделу

В результате выполнения целей раздела можно сделать следующие выводы:

1. Анализ конкурентных технических решений привел к тому, что рассматриваемый в данной работе проект является коммерчески выгодным и эффективным;
2. Проведено планирование разработки инженерного проекта командой из трех человек. По разработанному плану работ построена диаграмма Ганта.
3. Определена себестоимость технического решения размером в 759 т.р. Большинство расходов приходится на выплату заработной платы исполнителей проекта и на расходы во внебюджетные фонды;
4. Проведен интегральный анализ финансовой и ресурсной эффективности проекта, исходя из которого можно сделать следующие выводы:
 - Интегральный показатель финансовой эффективности равен 0.175, при этом проект оказался эффективнее своих конкурентов.
 - Значение интегрального показателя ресурсоэффективности составило 9,7, по сравнению с показателями конкурентов – 9,5 и 5,5;
 - Общий интегральный показатель эффективности составил 55,4, что как минимум в 6 раз выше, чем у конкурентов – 9,5 и 7,3.

6. Социальная ответственность

6.1. Введение

Объектом разработки данной ВКР является программное и алгоритмическое обеспечение георадара для исследования фундаментов и несущих конструкций зданий. Георадар — геофизический прибор, который позволяет сканировать почву или строительные конструкции на большой глубине. Это устройство излучает волны на определенной частоте, которые отражаются от скрытых предметов, а затем воспринимаются приемником и анализируются микропроцессором, после чего оператор получает на экране изображение грунта или строительной конструкции под поверхностью.

Работа выполняется с использованием персонального компьютера (ПК), поэтому в данном разделе анализируются опасные и вредные факторы, присутствующие при работе с ПК, их влияние на окружающую среду и возможные мероприятия по её защите.

Предметом исследования является рабочая зона программиста, состоящее из компьютерного стола, ПК, клавиатуры, компьютерной мыши, гарнитуры и кресла. Работы выполнялись в офисе компании Synecta, расположенной в г. Томске по адресу ул. Красноармейская, 128.

6.2. Правовые и организационные вопросы обеспечения безопасности

Процесс разработки программного обеспечения (ПО) осуществляется за компьютерным столом. Рабочее место должно быть устроено так, чтобы все органы управления находились в зоне досягаемости моторного поля оператора. Кроме того, органы управления следует располагать так, чтобы последовательность совершаемых оператором действий выполнялась слева направо и не вызывала необходимости перекрещивания рук. Все дисплеи следует располагать перпендикулярно нормальной линии взора оператора с допустимым отклонением от этой линии в 45°. Эти требования

устанавливаются согласно ГОСТ 12.2.032-78 «Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования» [13], а также ГОСТ 22269-76 Система "Человек-машина". «Рабочее место оператора. Взаимное расположение элементов рабочего места» [14].

Согласно требованиям ТК РФ «Рабочее время» к нормам труда (продолжительность рабочего дня, перерывы в течение рабочего дня, перерывы на обед) нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю, а также при условии работы пять дней в неделю составляет 8 часов в сутки [15]. Непрерывная работы с персональными электронно-вычислительными машинами (ПЭВМ) не должна превышать 2 часов, с обеспечением перерывов для поддержания оптимальной работоспособности сотрудника на протяжении рабочего времени [16].

Рабочее место, используемое при разработке ПО, было устроено так, чтобы удовлетворять всем требованиям по организации труда с ЭВМ, согласно характеру и времени работы, и обеспечивать комфортную среду для поддержания физического и психологического здоровья работника, использующего это рабочее место.

6.3. Производственная безопасность

В процессе работы с ПЭВМ во время разработки программного обеспечения работники находятся под воздействием различных вредных и опасных факторов, представленных в таблице 6.1. В таблице отражены наиболее вероятные факторы воздействия на разработчика во время различных стадий разработки ПО, а также соответствующие нормативные акты, описывающие допустимые нормы их воздействия.

Таблица 6.1 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Опасность поражения электрическим током	ГОСТ Р 58698-2019 «Защита от поражения электрическим током» [17]
Отклонение показателей микроклимата	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [16]
Недостаточная освещённость рабочей зоны	СП 52.13330.2016 «Естественное и искусственное освещение» [18]
Повышенная световая и цветовая контрастность	СП 52.13330.2016 «Естественное и искусственное освещение» [18]
Повышенный уровень шума на рабочем месте	ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности [19] СП 51.13330.2011 «Защита от шума. Актуализированная редакция СНиП 23-03-2003» [20]
Статические физические перегрузки	РД 153-34.0-03.298-2001 «Типовая инструкция по охране труда для пользователей ПЭВМ в электроэнергетике» [21]
Перегрузка зрительного аппарата	РД 153-34.0-03.298-2001 «Типовая инструкция по охране труда для пользователей ПЭВМ в электроэнергетике» [21]

6.3.1. Опасность поражения электрическим током

В процессе разработки ПО операторы взаимодействуют с ЭВМ. Нарушение целостности ЭВМ способно привести к накоплению электрического заряда внутри корпуса машины или на его стенках. Современные ЭВМ имеют защиту корпусов, предполагающую невозможность накопления электрического заряда на стенках корпуса. При этом высокую опасность представляют связанное с ЭВМ техническое оснащение, такое как удлинители, кабели питания. Кабели питания проводят ток напряжением 220В, который считается опасным для жизни. Нарушение изоляции может привести к производственному несчастному случаю.

Защита людей от электрического, статического токов и от воздействия электромагнитных полей входит в производственную электробезопасность. ГОСТ Р 58698-2019 регулирует нормы допустимых воздействий напряженности токов и полей на организм человека. Пороги воздействия отражены в таблице 6.2.

Таблица 6.2 – Пороги напряжения прикосновения для реагирования

Характер реагирования	Пороги напряжения, В
Реакция испуга	2 (переменный ток)
	8 (постоянный ток)
Мышечная реакция	20 (переменный ток)
	40 (постоянный ток)

Основными способами защиты от прямого поражения электрическим током являются: использование дополнительной изоляции проводки и объектов на электрически опасных участках (в том числе установка защитных ограждений), сокрытие опасных электрических участков в недосягаемых для работников зонах, использование автоматизированных приборов контроля характеристик электрической сети офисных помещений с возможностью отключения сети в случае короткого замыкания.

6.3.2. Отклонение показателей микроклимата

Превышение или преуменьшение допустимых показателей микроклимата на рабочем месте от нормативных влечет за собой негативные воздействия на здоровье работников. Изменения касаются не только показателей температуры воздушной среды помещения, но также влажности воздуха, скорости его движения. Ненормативные показатели микроклимата могут привести к переохлаждению или перегреву организма, пересыханию, к заражению болезнетворными бактериями. В процессе разработки ПО, персональные компьютеры выделяют небольшое количество тепла, способное изменить параметры микроклимата в помещении, уменьшая влажность воздуха и увеличивая его температуру.

СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» регламентирует санитарные нормы, относящиеся к микроклимату производственных помещений. Установленные предпочтительные и максимально допустимые значения показателей микроклимата на рабочих местах являются нормами для создания комфортных и безопасных условий труда, поддержания физиологических потребностей человека.

Разработка программного обеспечения по энергозатратам относится к категории Ia, поскольку производится сидя и характеризуется незначительными физическими усилиями. Для этой категории работ в таблицах 6.3 и 6.4 представлены оптимальные и допустимые значения показателей микроклимата на рабочих местах.

Таблица 6.3 – Оптимальные величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Тёплый	23-25	22-26	60-40	0,1

Таблица 6.4 – Допустимые величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
	диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более
Холодный	20,0 – 21,9	24,1 – 25,0	19,0 – 26,0	15 – 75	0,1	0,1
Тёплый	21,0 – 22,9	25,1 – 28,0	20,0 – 29,0	15 – 75	0,1	0,1

В случае, если поддерживать допустимые показатели микроклимата в производственном помещении не является возможным, проводятся защитные мероприятия для предотвращения возможного перегревания и переохлаждения работников, а именно: использование систем местного кондиционирования воздуха, точное определение периодов работы сотрудников, с установленным временем отдыха в других помещениях с нормативными показателями микроклимата, сокращение рабочей смены и другие различные мероприятия.

6.3.3. Недостаточная освещенность рабочей зоны

Недостаток освещенности рабочих пространств негативно влияет на сотрудников, поскольку требует от них большей концентрации во время рабочей деятельности. Это влечет за собой повышенную утомляемость, а вследствие нее низкую производительность специалистов. Кроме того, большое время работы в помещении с недостаточной освещенностью становится причиной ухудшения зрения.

СП 52.13330.2016 «Естественное и искусственное освещение» устанавливает нормы для естественного, искусственного и совместного освещения. Разработка программного обеспечения относится соответствует категории работ высокой точности – Б (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), с подразрядом 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%).

Требования к освещению рабочего помещения для разряда Б1 приведены в таблице 6.5.

Таблица 6.5 – Требования к освещению рабочего помещения

Искусственное освещение				Естественное освещение	
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, Кп, %, не более	Коэффициент естественной освещенности, %, при	
				верхнем или комбинированном	боковом
300	100	21	15	3	1

Яркий свет, оказавшийся в зоне периферийного зрения, влияет на глазные мышцы, сильно напрягая их. Чтобы уменьшить воздействие недостаточной освещенности на разработчиков ПО необходимо чтобы уровень естественного освещения рабочего пространства как можно ближе совпадал с яркостью монитора. Повысить уровень естественного освещения можно за счет расширения или добавления оконных проемов, а уровень искусственного освещения за счет добавления дополнительных осветительных приборов.

6.3.4. Повышенная световая и цветовая контрастность

Слишком низкий или слишком высокий световой и цветовой контраст на рабочем месте вызывает утомление и, как следствие, снижение уровня работоспособности специалиста. Кроме того, долгое время работы в условиях отсутствия нормальной световой и цветовой контрастности приводит к ухудшению зрения. СП 52.13330.2016 «Естественное и искусственное освещение» устанавливает нормы светового и цветового контраста. Для работы с ПЭВМ (категория работ Б1) регламентируемые нормы контрастности объекта с фоном представлены в таблице 6.6.

Таблица 6.6 – Требования к освещению рабочего помещения

Характеристика зрительной работы	Контраст объекта с фоном	Характеристика фона
Высокой точности	Малый	Средний
	Средний	Темный

Для нормализации светового и цветового контраста в помещении следует отрегулировать уровень естественной и искусственной освещенности рабочего пространства или заменить имеющиеся технические устройства для визуального воспроизведения информации (дисплеи) на более качественные, с возможностью детальной настройки яркости и контрастности изображения.

6.3.5. Повышенный уровень шума на рабочем месте

Шумы в производственном помещении приводят к психологическому стрессу во время рабочей деятельности, снижая концентрацию и внимание специалистов, вследствие чего падает общая производительность труда. Шумы в рабочем помещении возникают за счет активной работы различных устройств, таких как систем вентиляции и кондиционирования воздуха, шумов систем охлаждения персональных компьютеров.

ГОСТ 12.1.003-2014 ССБТ. «Шум. Общие требования безопасности» и СП 51.13330.2011 «Защита от шума. Актуализированная редакция СНиП 23-03-2003» утверждают нормативные показатели к уровню звука и звукового давления.

Характерные по отношению к разработке ПО показатели звукового давления и уровни звука представлены в таблице 6.7.

Таблица 6.7 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста

Назначение помещений или территорий	Время суток, ч	Для источников постоянного шума									Для источников непостоянного шума	
		Уровни звукового давления дБ, в октавных полосах частот со среднегеометрическими частотами, Гц									Эквивалентный корректурованный по А уровень звука $L_{Aэв}$, дБ	Максимальный корректурованный по А уровень звука L_{Amax} , дБ
		31,5	63	125	250	500	1000	2000	4000	8000		
											Уровень звука корректурованный по А, L_A , дБ	

Помещения офисов, рабочие помещения и кабинеты административных зданий, научно- исследовательских организаций	-
	86
	71
	61
	54
	49
	45
	42
	40
	38
	50
	50
	65

Снижение уровня шума в рабочем помещении достигается за счет уменьшения шумов, издаваемых техникой. Часто эти шумы исходят от того, что грязь и пыль во время работы вентиляторов цепляется за корпус и другие внутренние детали устройств, вызывая трение. Своевременное обслуживание техники на рабочих местах позволит не только повысить срок ее службы, но и уменьшить ощущаемый в помещении шум.

6.3.6. Статические физические перегрузки

Разработка программного обеспечения происходит за компьютером, часто в одной и той же позе долгое время работы, что вызывает напряжение мышц неподвижных частей тела: шеи, плечевого пояса и спины. Такие статические нагрузки в течение долго периода времени становятся причинами серьезных проблем со здоровьем, сопровождающихся возможными симптомами, такими как общее утомление организма, сонливость, боли в мышцах, онемение конечностей.

Дополнительные статические нагрузки на организм исходят из неправильной организации рабочего места. Сюда можно отнести низкую высоту сидений кресла относительно рабочего стола, из-за чего сотрудники прикладывают дополнительные усилия на поддержание рук во время работы, эти нагрузки распределяются на мышцы плеч и рук. В обратном случае, низкая высота рабочего стола относительно кресла напрягает мышцы спины и шеи, поскольку вынуждает сотрудника наклоняться к рабочей поверхности.

РД 153-34.0-03.298-2001 устанавливает рекомендуемое непрерывное время работы за компьютером не более 2 часов подряд без перерывов. ГОСТ 12.2.032-78 регламентирует случаи, когда невозможно регулировать высоту и угол наклона подставки для ног и размеры рабочей поверхности. В таком

случае разрешается изготавливать оборудование с постоянными параметрами высоты и наклона. При этом высота рабочей поверхности ограничена нижней границей в 725 мм, в то время как высота кресла должна регулироваться в пропорционально росту работника.

6.3.7. Перегрузка зрительного аппарата

РД 153-34.0-03.298-2001 описывает взаимные требования к расположению оператора и дисплея, а также возможные заболевания, вызванные нарушением регламентируемых требований. Работа за компьютером требует высокой зрительной сосредоточенности, что приводит к перенапряжению глазных мышц. Долгое непрерывное время работы за компьютером является следствием таких заболеваний, как спазм аккомодации, снижение остроты зрения, уменьшение запаса относительной аккомодации и близорукость. Тяжесть представленных заболеваний напрямую зависит от расстояния между монитором и зрительным аппаратом человека. Рекомендуется, чтобы это расстояние составляло 600-700 мм, и не было менее чем 500 мм. Кроме того, во время работы за компьютером рекомендуется делать перерывы. Перерывы следует делать не менее чем каждые 2 часа работы за компьютером.

6.4. Экологическая безопасность

Разработка программного обеспечения как процесс не оказывает вредного влияния на окружающую среду, однако используемое при этом оборудование оказывает небольшое отрицательное влияние за счет нагрева оборудования и малых выбросов негативных веществ, выделяющихся при этом на таких внутренних элементах, как материнская плата.

Внутренняя структура ПЭВМ представляет собой комплекс деталей с различными классами опасности от I до IV (материнские платы, аккумуляторы, ртутные лампы, трансформаторы и т.д.). В этот комплекс входят элементы содержащие такие вредные вещества как ртуть, свинец, литий, кадмий, олово, медь, железо, алюминий, магний, пластик. Классы

опасности и предельно допустимая концентрация (ПДК) этих веществ указаны в таблице 6.8 [22].

Таблица 6.8 – ПДК вредных веществ в воздухе рабочей зоны

Наименование вещества	Величина ПДК, мг/м ³	Класс опасности
Ртуть металлическая	0,01/0,005	I
Свинец и его неорганические соединения	0,01/0,005	I
Фториды лития	1/0.02	II
Кадмий и его неорганические соединения	0,05/0,01	I
Фториды олова	1/0.02	II
Медь	1/0,5	II
Соединения железа	1-6	III
Алюминий и его сплавы (в пересчете на алюминий)	6	IV
Магниты (алюминия, меди)	6	III

Утилизация компьютерной техники должна соответствовать требованиям ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» [23].

При неправильной утилизации комплектующих компьютера в литосферу попадают вредные вещества, содержащиеся в этих комплектующих. Процесс правильной утилизации компьютерной техники включает в себя разбор компьютерных компонент на составляющие, их сортировку и отправку на специально отведенные полигоны, на которых дальнейшей утилизацией компонент будет заниматься квалифицированный персонал.

В процессе производства компьютерных компонент происходят химические процессы с выделением сточных вод, попадающих в гидросферу планеты и загрязняющих ее.

Придерживаясь всех норм использования, производства и утилизации компьютерной техники можно добиться минимизации ее воздействия на окружающую среду.

6.5. Безопасность в чрезвычайных ситуациях

6.5.1. Пожар

Пожары в производственном помещении наиболее часто выражены как ЧС техногенного характера и возникают вследствие коротких замыканий или неисправностью контролирующих температуру элементов ПК. В режиме повышенной нагрузки неисправные компьютеры могут достигать пожароопасных температур, что может привести к пожару. Офисные помещения относятся к классу функциональной пожароопасности Ф4.3, поскольку к пожару наиболее часто приводит короткое замыкание техники, можно учитывать, что площадь пожарной нагрузки на каждом из участков пола не более 10м^2 , следовательно категория пожароопасности офисного помещения соответствует В4 с возможным классом пожара Е, характерного для горючих веществ (столов, кресел) и электроустановок, находящихся под напряжением [24].

Минимизировать риск пожара можно проводя своевременную диагностику всего рабочего оборудования, диагностику электрических цепей производственного здания. Всем сотрудникам следует соблюдать правила безопасности при работе с персональными компьютерами. Производственное здание должно удовлетворять требованиям пожарной безопасности, иметь систему охранно-пожарной сигнализации, план эвакуации, углекислотные огнетушители, эвакуационные указатели.

При пожаре сотрудники организации обязаны немедленно уведомить об этом рабочий персонал предприятия и сообщить в службу пожарной

охраны, указав адрес здания и конкретное место возгорания, если оно известно.

6.6. Вывод раздела

При работе над разделом «Социальная ответственность» рассмотрены важные нормативные документы, контролирующие обеспечение безопасности в производственных помещениях. В разделе рассмотрены и описаны опасные и вредные факторы, возникающие при разработке программного обеспечения для георадарного оборудования. Предъявлены способы снижения риска возникновения опасных и вредных факторов. Установлено влияние процесса разработки программного обеспечения на окружающую среду и описаны способы защиты окружающей среды от вредного воздействия в процессе производственной деятельности. Рассмотрены наиболее типичные для данного вида деятельности чрезвычайные ситуации и предложены меры для уменьшения риска их возникновения.

Определена категория персонала по электробезопасности. Персонал относится к I категории, поскольку не относится к электротехническому и электротехнологическому персоналу, но при этом выполняет работы, при которых может возникнуть опасность поражения электрическим током [25]. Относительно электробезопасности рабочее помещение имеет класс – без повышенной опасности, поскольку в помещении отсутствуют условия, создающие опасность (наличие пыли или высокие температуры и т.д.) [26].

Установлена категория тяжести выполняемых работ по разработке программного обеспечения – Ia (работы, производимые сидя, стоя или связанные с ходьбой и сопровождающиеся физическим напряжением).

Установлена категория зрительных работ по разработке программного обеспечения Б1 (наименьший или эквивалентный объект различения 0,30 – 0,50 мм; относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%).

Для рассматриваемого помещения установлена категория пожароопасности В4, с возможным классом пожара Е. Также установлена категория функциональной пожароопасности помещения для офиса Ф4.3.

Объект, на котором производятся работы по разработке ПО, оказывает незначительное вредное воздействие на окружающую среду и относится к объектам III категории [27].

Заключение

В результате выполнения выпускной квалификационной работы был разработан алгоритмический и программный комплекс георадара, состоящий из базовых миграционных алгоритмов.

На языке Python разработан и реализован алгоритм гиперболической миграции, позволяющий фокусировать миграционные изображения исходя из формы гиперболической кривой, описывающей распределение электромагнитной волны на определенной глубине скана.

Разработан и реализован алгоритм миграции Стола, представляющий собой разрешение волнового уравнения в частотной области. При реализации алгоритма для перехода между временной и частотной областями применялись многомерные Фурье-преобразования.

Разработан алгоритм интерполяции двумерных данных методом обратного взвешивания. Алгоритм применяется для преобразования набора двумерных несфокусированных изображений в сфокусированное трехмерное. Трехмерные изображения являются входными данными для алгоритмов 3D гиперболической миграции и миграции Столта.

Результат работы разработанного алгоритмического и программного комплекса продемонстрирован на искусственной модели бетонной среды. Результаты показали, что применение интерполяции для создания трехмерного несфокусированного скана и последующее применение к нему трехмерной миграции Столта позволяет различить скрытые внутри модели объекты.

В ходе работы разработанные алгоритмы миграции были протестированы на искусственно сгенерированных входных данных с использованием программы `gprMax`. Сравнены подходы к созданию трехмерных сканов с использованием линейной интерполяции и с использованием интерполяции методом обратного взвешивания. Алгоритмы интерполяции показали схожие результаты, поэтому предпочтительнее использовать линейную интерполяцию в пользу скорости работы алгоритма.

Список использованных источников

1. Георадарное обследование грунтов и фундаментов // Независимое экспертное партнерство [Электронный ресурс]. – Режим доступа: <https://www.nep.expert/obsledovanie/georadarnoe-obsledovanie/> (дата обращения: 27.05.2023).
2. Kirchhoff and F-K migration to focus ground penetrating radar images / N. Smitha, R,Ullas D, S. Abilash, N,Sridhara,&,Vipula S. // International Journal of Geo-Engineering. – 2016. – № 4. [Электронный ресурс]. – Режим доступа: <https://link.springer.com/content/pdf/10.1186/s40703-016-0019-6.pdf?pdf=button> (дата обращения: 27.05.2023).
3. Emanuele, Forte Ground Penetrating Radar GPR - principles / Forte Emanuele // Corso di Laurea Magistrale in GEOSCIENZE [Электронный ресурс]. – Режим доступа: https://moodle2.units.it/pluginfile.php/432592/mod_resource/content/2/UD4a_GPR_principles.pdf (дата обращения: 27.05.2023).
4. A Review on Migration Methods in B-Scan Ground Penetrating Radar Imaging / Özdemir Caner // Hindawi [Электронный ресурс]. – Режим доступа: <https://www.hindawi.com/journals/mpe/2014/280738/> (дата обращения: 27.05.2023).
5. Craig, Warren gprMax: Open source software to simulate electromagnetic wave propagation for Ground Penetrating Radar / Warren Craig, Giannopoulos Antonios, Giannakis Iraklis // Computer Physics Communications. – 2016. – С. 163-170 [Электронный ресурс]. – Режим доступа: <https://www.sciencedirect.com/science/article/pii/S0010465516302533?via%3Dihub> (дата обращения: 27.05.2023).
6. Unleash the Power of ParaView // ParaView [Электронный ресурс]. – Режим доступа: <https://www.paraview.org/> (дата обращения: 27.05.2023).

7. Отображение нескольких подземных коммуникаций // SENSORS & SOFTWARE from RADIODETECTION [Электронный ресурс]. – Режим доступа: <https://www.sensoft.ca/ru/case-studies/mapping-multiple-utilities/> (дата обращения: 27.05.2023).
8. Stolt's f-k migration for plane wave ultrasound imaging / Garcia Damien. // PDF [Электронный ресурс]. – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3970982/pdf/nihms3487.pdf> (дата обращения: 27.05.2023).
9. DAVID, B. L. Wave-Based Non-Line-of-Sight Imaging using Fast f-k Migration / B. L. DAVID. // STANFORD COMPUTATIONAL IMAGING LAB [Электронный ресурс]. – Режим доступа: http://www.computationalimaging.org/wp-content/uploads/2019/05/nlos_fk.pdf (дата обращения: 27.05.2023).
10. Magdalena, Rucka Interpolation methods in GPR tomographic imaging of linear and volume anomalies for cultural heritage diagnostics / Rucka Magdalena, Wojtczak Erwin, Zielin'ska Monika. // Measurement. – 2020. – № 154. [Электронный ресурс]. – Режим доступа: <https://reader.elsevier.com/reader/sd/pii/S0263224120300312> (дата обращения: 27.05.2023).
11. Линейная интерполяция // matematicus.ru [Электронный ресурс]. – Режим доступа: <https://www.matematicus.ru/vyshshaya-matematika/issledovanie-funktsii/linejnaya-interpolyatsiya> (дата обращения: 27.05.2023).
12. Компьютеры - Билинейная интерполяция // chinapads.ru [Электронный ресурс]. – Режим доступа: https://chinapads.ru/c/s/bilineynaya_interpolyatsiya (дата обращения: 27.05.2023).
13. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования // Электронный фонд правовых и нормативно-технических

- документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003913> (дата обращения: 27.05.2023).
- 14.ГОСТ 22269-76 Система "Человек-машина". Рабочее место оператора. Взаимное расположение элементов рабочего места // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200012834> (дата обращения: 27.05.2023).
- 15.Трудовой кодекс (ТК РФ) «Рабочее время»// Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/12125268/> (дата обращения: 27.05.2023).
- 16.СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573500115> (дата обращения: 27.05.2023).
- 17.ГОСТ Р 58698-2019 «Защита от поражения электрическим током» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200170001> (дата обращения: 20.05.2023).
- 18.СП 52.13330.2016 «Естественное и искусственное освещение» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 27.05.2023).
- 19.ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200118606> (дата обращения: 27.05.2023).

- 20.СП 51.13330.2011 «Защита от шума. Актуализированная редакция СНиП 23-03-2003» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200084097> (дата обращения: 27.05.2023).
- 21.РД 153-34.0-03.298-2001 «Типовая инструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200031404> (дата обращения: 27.05.2023).
- 22.ГОСТ 12.1.005-88 «Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200003608> (дата обращения: 27.05.2023).
- 23.ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» // Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200081740> (дата обращения: 27.05.2023).
- 24.Федеральный закон от 22.07.2008 N 123-ФЗ (ред. от 30.04.2021) "Технический регламент о требованиях пожарной безопасности // КонсультантПлюс [Электронный ресурс]. – Режим доступа: <https://mchs.gov.ru/uploads/document/2022-04-08/c907f456516c1f21009131cfdb944deb.pdf> (дата обращения: 27.05.2023).
- 25.Приказ «Об утверждении правил по охране труда при эксплуатации электроустановок» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573264184> (дата обращения: 27.05.2023);

- 26.«Правила устройства электроустановок» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200030216> (дата обращения: 27.05.2023);
- 27.Федеральный закон от 10.01.2002 N 7-ФЗ (ред. от 14.07.2022) "Об охране окружающей среды" (с изм. и доп., вступ. в силу с 01.03.2023) // КонсультантПлюс [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_34823/a646b4197509ac4f8583b190e3b018fc27f3ae35/ (дата обращения: 27.05.2023).

Приложение А

Реализация алгоритма двумерной гиперболической миграции

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
from matplotlib import colors
```

```
# 1. GLOBAL SETTINGS
```

```
IN_FILES_FOLDER = r'PATH TO FILE FOLDER'
```

```
IN_FILENAME = r'BSCAN FILE NAME'           // Should convert *.hdf5 file to *.txt
```

```
PLT_COLOMAP = 'seismic'
```

```
FLAG_SHOW_PLT = True
```

```
tw = 2.8e-9                                # time-window
```

```
dx = 0.004                                # step on X-axis
```

```
d_comp = 0.18                             # target distance
```

```
t_comp = 1.5e-9                           # wave travel time from emitter to target
```

```
er = 6                                    # medium permittivity
```

```
C = 3e8/np.sqrt(er)                       # wave propagation speed
```

```
# 2. LOADING DATA
```

```
try:
```

```
    IN_PATH = IN_FILES_FOLDER + IN_FILENAME
```

```
    in_data = np.loadtxt(IN_PATH)
```

```
except:
```

```
    print('Data loading is failed.')
```

```
# 3. PREPARE DATA
```

```
data_len = len(in_data)
```

```
X_data_len = len(in_data[0])
```

```
dt = tw/data_len
```

```
dz = C*dt/2
```

```
in_data_sqr = in_data**2
```

```
output_data = np.empty_like(in_data)
```

```
real_T = np.array([i*dt for i in range(0, data_len)])
```

```
real_Z = np.array([i*dz for i in range(0, data_len)])
```

```
real_X = np.arange(0, X_data_len*dx, dx)
```

```
Z_idx = np.arange(0, data_len)
```

```
X_idx = np.arange(0, X_data_len)
```

```
# 4. HYPERBOLIC MIGRATION
```

```
# Calculate parabola template for each pixel
```

```
for i in Z_idx:
```

```
    t = real_T[i]
```

```
    z = real_Z[i]
```

```
for j in X_idx:
```

```
    x = real_X[j]
```

```
# Finding trace-points for x0, z0 and calculate RMS
```

```
Z_trace = np.sqrt((real_X - x)**2 + z**2)
```

```
mask = np.ma.masked_array(Z_trace <= real_Z[-1])
```

```
Z_trace = Z_trace[mask]
```

```
# Convert Z_trace into Z-indices and get X-indices
```

```

X_trace_idx = X_idx[mask]
Z_trace_idx = (Z_trace/dz).astype(int)

```

```

# Calculating RMS

```

```

N = len(Z_trace_idx)
RMS = 1.0/np.sqrt(N)*np.sum(in_data_sqr[Z_trace_idx, X_trace_idx])
output_data[i, j] = RMS

```

```

# 5. DATA NORMALIZE

```

```

max_abs_value = np.max(np.abs(output_data))
output_data /= max_abs_value

```

```

# 6. SHOW MIGRATED IMAGE

```

```

if (FLAG_SHOW_PLT):

```

```

    fig, ax = plt.subplots()
    im = ax.imshow(output_data, cmap = PLT_COLOMAP, aspect='auto',
                    extent=[real_X[0], real_X[-1], real_Z[-1], real_Z[0]],
                    norm = colors.TwoSlopeNorm(vmin=0., vcenter=0.76, vmax=1))

```

```

    ax.set_ybound(real_Z[0], real_Z[-1])
    ax.set_title('Hyperbolic (diffraction) summation')
    ax.set_xlabel('X-Distance [m]')
    ax.set_ylabel('Z-Depth [m]')

```

```

    fig.colorbar(im, ax=ax, label='Field strenght [V/m]')
    plt.show()

```

Приложение Б

Реализация алгоритма трехмерной гиперболической миграции.

```
from matplotlib.widgets import Slider
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
from matplotlib import colors
```

```
# 1. Settings
```

```
IN_FILES_FOLDER = r'PATH TO FILE FOLDER'
```

```
IN_FILENAME = r'BSCAN FILE NAME in .npy format'
```

```
PLT_COLORMAP = 'seismic'
```

```
FLAG_SHOW_PLT = True
```

```
tw = 3.0e-9
```

```
dx = 0.008
```

```
dy = 0.008
```

```
rx_t = 244
```

```
er = 6
```

```
C = 3e8/np.sqrt(er)
```

```
# 2. Loading data
```

```
try:
```

```
    IN_PATH = IN_FILES_FOLDER + IN_FILENAME
```

```
    in_data = np.load(IN_PATH)
```

```
except:
```

```
    print('Data loading is failed.')
```

3. Prepare data

in_data = in_data[rx_t:]

Nt, Xn, Yn = in_data.shape

dt = tw/Nt

*dz = C*dt/2*

*in_data_sqr = in_data**2*

output_data = np.empty_like(in_data)

*real_T = np.array([i*dt for i in range(0, Nt)])*

*real_Z = np.array([i*dz for i in range(0, Nt)])*

*real_X = np.arange(0, Xn*dx, dx)*

*real_Y = np.arange(0, Yn*dy, dy)*

real_X_grid, real_Y_grid = np.meshgrid(real_X, real_Y)

Z_idx = np.arange(0, Nt)

X_idx = np.arange(0, Xn)

Y_idx = np.arange(0, Yn)

X_idx_grid, Y_idx_grid = np.meshgrid(X_idx, Y_idx)

3D Hyperbolic migration

for i in Z_idx:

print(i)

t = real_T[i]

z = real_Z[i]

for j in Y_idx:

```

for k in X_idx:
    y = real_Y[j]
    x = real_X[k]

    # Finding trace-points for x0, y0, z0 and calculate RMS
    Z_trace = np.sqrt((real_X_grid - x)**2 + (real_Y_grid - y)**2 + z**2)

    mask = np.ma.masked_array(Z_trace <= real_Z[-1])
    Z_trace = Z_trace[mask]

    # Convert Z_trace into Z-indices and get XY-indices
    X_trace_idx = X_idx_grid[mask]
    Y_trace_idx = Y_idx_grid[mask]
    Z_trace_idx = (Z_trace/dz).astype(int)

    # Calculating RMS
    N = Z_trace_idx.size
    RMS = 1.0/np.sqrt(N)*np.sum(in_data_sqr[Z_trace_idx, X_trace_idx,
Y_trace_idx])
    output_data[i, j, k] = RMS

# 4. Data normalize
max_abs_value = np.max(np.abs(output_data))
output_data /= max_abs_value

# 5. Show migrated image
if (FLAG_SHOW_PLT):

    x_axis = np.arange(Xn)
    y_axis = np.arange(Yn)

```

```

fig, ax = plt.subplots()
plt.subplots_adjust(left=0.25, bottom=0.25)

l = plt.imshow(output_data[0], cmap='seismic', vmin=0, vmax=1)

contour_axis = fig.gca()
contour_axis.set_xlabel("X-Axis")
contour_axis.set_ylabel("Y-Axis")

ax_z_color = 'lightgoldenrodyellow'
ax_z = plt.axes([0.25, 0.1, 0.6, 0.03], facecolor=ax_z_color)
slider_z = Slider(ax_z, 'Z-coord ', 0, Nt-1, valinit=0, valstep=1)

def update(val):
    contour_axis.clear()
    contour_axis.imshow(output_data[slider_z.val], cmap='seismic', vmin=0,
vmax=1)

    contour_axis.set_xlabel("X-Axis")
    contour_axis.set_ylabel("Y-Axis")
    plt.draw()

slider_z.on_changed(update)

# Save migrated image as .npy file
np.save("MIGRATION_3D.npy", output_data)
plt.show()

```

Приложение В

Реализация алгоритма двумерной миграции Столта

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
def fk_migration(E, Fs, dx, Um) -> np.ndarray:
```

```
    """
```

*Функция fktig выполняет миграцию несфокусированного изображения
путем*

*преобразования пространственно-временной области в
пространственно-частотную*

*Краткое описание преобразований: $E(t, x) \rightarrow E(f, kx) \rightarrow E(kz, kx) \rightarrow E(z,$
 $x)$*

```
    -----  
    * E - входные данные в формате  $E(t, x)$ 
```

```
    * Fs - частота дискретизации
```

```
    * dx - расстояние между отсчетами по оси X
```

```
    * Um - скорость распространения волны в среде
```

```
    """
```

```
    nt, nx = E.shape      # Размер изображения
```

```
    Fn = Fs/2             # Частота Найквиста
```

```
    Um = 2*Um/np.sqrt(2)  # Компонента скорости модифицируется в  
соответствии с ERM
```

```
    # ERM - Exploding Reflector Model
```

```
    # Задание осей
```

```
    f0 = np.arange(-nt/2, nt/2+1)*Fs/nt
```

```
    kx = np.arange(-nx/2, nx/2)/dx/nx
```

```

Kx, f = np.meshgrid(kx, f0)
fkz = Um * np.sign(f) * np.sqrt(Kx**2 + f**2/Um**2)

E = np.fft.rfft2(E, s=(nx, 2*nt), axes=(-1, -2))
E = np.fft.fftshift(E)

# Линейная интерполяция частотной области (FK-Mapping): E(kx, f) ->
E(kx, kz)
for i in range(nx):
    real_part = np.interp(fkz[:, i], f[:, i], np.real(E[:, i]))
    imag_part = np.interp(fkz[:, i], f[:, i], np.imag(E[:, i]))
    E[:, i] = real_part + 1j*imag_part

# Вычисление kz, умножение на масштабирующий множитель
kz = np.arange(-nt/2, nt/2+1)/Um/Fs/nt
_, kz = np.meshgrid(kx, kz)
E *= Um**2 * kz /(fkz+np.spacing(1))

E = np.fft.ifftshift(E)
E = np.fft.irfft2(E, axes=(-1, -2))[:nt]

x = np.arange(nx)*dx/2
z = np.arange(nt)*Um/(Fs/2)

return x, z, E

IN_FILES_FOLDER = r'PATH TO FILE FOLDER'
IN_FILENAME = r'BSCAN FILE NAME'

```

1. LOADING DATA

try:

```
IN_PATH = IN_FILES_FOLDER + IN_FILENAME
```

```
in_data = np.loadtxt(IN_PATH)
```

except:

```
print('Data loading is failed.')
```

```
er = 6
```

```
C = 3e8/np.sqrt(er)
```

```
f = 3e9
```

```
tw = 3e-9
```

```
dx = 0.004
```

```
PLT_COLORMAP = 'seismic'
```

```
FLAG_SHOW_PLT = True
```

```
ax_x, ax_z, out = fk_migration(in_data, 1/(tw/375), dx, Um=C)
```

```
ax_x += ax_x[-1]
```

2. SHOW MIGRATED IMAGE

if (FLAG_SHOW_PLT):

```
fig, ax = plt.subplots()
```

```
im = ax.imshow(np.real(out), cmap = PLT_COLORMAP, aspect='auto',  
               extent=[ax_x[0], ax_x[-1], ax_z[-1], ax_z[0]])
```

```
ax.set_title('f-k Migration REAL')
```

```
ax.set_xlabel('x-distance [m]')
```

```
ax.set_ylabel('z-depth [m]')
```

```
fig.colorbar(im, ax=ax, label='Field strenght [V/m]')  
plt.show()
```

Приложение Г

Реализация алгоритма трехмерной f-k миграции.

```
from matplotlib.widgets import Slider
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
def fk3d_migration(Data, dx, dy, dt, Um):
```

```
    """
```

```
        F-K Migration algorithm
```

```
        * Input Data -> E(t, y, x)
```

```
        * OUTput Data -> E(z, y, x)
```

```
        Additional parameters (required)
```

```
        * dx - step by X-spatial axis
```

```
        * dy - step by Y-spatial axis
```

```
        * dt - step by time axis
```

```
        * Um - plane wave speed in the medium
```

```
    """
```

```
# Pre-Processing
```

```
Nt, Nx, Ny = Data.shape
```

```
# Using only half of wave speed in medium (q.v. Exploding Reflector Model)
```

```
Um = Um/2
```

```
dz = Um*dt
```

```
# Zero-Padding Data to get correct data in result fourier transformation
```

```
Data = np.pad(Data, ((0, 1*Nt), (0, 0), (0, 0)))
```

```

# Fast Fourier Transform
Data = np.fft.rfftn(Data, axes=(-1, -2, -3))

# New axes (fftfreq()) realization
f_axis = np.fft.rfftfreq(2*Nt, dt)
kx_arr = np.fft.fftfreq(Ny, dy)
ky_arr = np.fft.fftfreq(Nx, dx)
kz_arr = np.fft.rfftfreq(2*Nt, dz)

# Stolt Interpolation
Resample_Data = Data

for x, kx in enumerate(kx_arr):
    for y, ky in enumerate(ky_arr):
        for z, kz in enumerate(kz_arr):

            M = Um * np.abs(kz) / (np.sqrt(kx**2 + ky**2 + kz**2) +
np.spacing(1))
            f = Um * np.sqrt(kx**2 + ky**2 + kz**2)

            # Interpolation between complex numbers
            Data[z, y, x] = M * np.interp(f, f_axis, Resample_Data[:, y, x])

# Inverse Fast Fourier Transform
Data = np.fft.irfftn(Data, axes=(-1, -2, -3))

# Unpadding data
Data = np.real(np.abs(Data[:Nt])**2)

# Normalizing data

```

```

    Data /= np.max(Data)

    # Return result array
    return Data

IN_FILES_FOLDER = r'PATH TO FILE FOLDER'
IN_FILENAME = r'BSCAN FILE NAME in .npy format'
t0 = 0

# 1. Loading data
try:
    IN_PATH = IN_FILES_FOLDER + IN_FILENAME
    in_data = np.load(IN_PATH)[t0:]
except:
    print('Data loading is failed.')

# 2. Settings
Nt, Nx, Ny = in_data.shape

er = 6
Um = 3e8/np.sqrt(er)

tw = 3e-9
dx = 0.008
dy = 0.008

dt = tw/Nt

PLT_COLORMAP = 'seismic'
FLAG_SHOW_PLT = True

```

3. Stolt Migration

```
OUT = fk3d_migration(in_data, dx, dy, dt, 0.11/1e-9/np.sqrt(2))
```

4. Show migrated image

```
if (FLAG_SHOW_PLT):
```

```
    min_value = np.min(OUT)
```

```
    max_value = np.max(OUT)
```

```
    x_axis = np.arange(Nx)
```

```
    y_axis = np.arange(Ny)
```

```
    fig, ax = plt.subplots()
```

```
    plt.subplots_adjust(left=0.25, bottom=0.25)
```

```
    l = plt.imshow(OUT[0], cmap=PLT_COLORMAP, vmin=min_value,  
vmax=max_value)
```

```
    contour_axis = fig.gca()
```

```
    contour_axis.set_xlabel("X-Axis")
```

```
    contour_axis.set_ylabel("Y-Axis")
```

```
    ax_z_color = 'lightgoldenrodyellow'
```

```
    ax_z = plt.axes([0.25, 0.1, 0.6, 0.03], facecolor=ax_z_color)
```

```
    slider_z = Slider(ax_z, 'Z-coord ', 0, Nt-1, valinit=0, valstep=1)
```

```
    def update(val):
```

```
        contour_axis.clear()
```

```
contour_axis.imshow(OUT[slider_z.val], cmap=PLT_COLORMAP,  
vmin=min_value, vmax=max_value)
```

```
contour_axis.set_xlabel("X-Axis")
```

```
contour_axis.set_ylabel("Y-Axis")
```

```
plt.draw()
```

```
slider_z.on_changed(update)
```

```
# Save migrated image to .npy file
```

```
np.save("MIGRATION_FK_2_0_3D.npy", OUT)
```

```
plt.show()
```

Приложение Д

Реализация алгоритма взвешенной линейной интерполяции по двум координатам.

```
from matplotlib.widgets import Slider
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
import os
```

```
def Cscan_linear_interp(Horizontal_scans, Vertical_scans):
```

```
    """
```

```
        Interpolate (3D) C-scan data each z-slice by half sum of two linear  
        interpolated C-scans on X-axis and Y-axis
```

```
        Requiered parameters
```

```
        -----
```

```
        `Horizontal_scans`: `3D-ndarray` like
```

```
            Array of B-Scans data in XZ plane with shape (t, y, x)
```

```
        `Vertical_scans`: `3D-ndarray` like
```

```
            Array of B-Scans data in YZ plane with shape (t, y, x)
```

```
        Return
```

```
        -----
```

```
        Interpolated C-scan data like `3D-ndarray` with shape (t, y, x)
```

```
    """
```

```
# 1. Shapes of data must be same sample size on time-axis
```

```
h_shape = Horizontal_scans.shape
```

```
v_shape = Vertical_scans.shape
```

```

if (h_shape[0] != v_shape[0]):
    raise ValueError('Shapes of input B-scans on time-axis must be same.')

# 2. Calculate variables
hor_scans = h_shape[2]
ver_scans = v_shape[2]

t_samples = h_shape[0]
y_samples = v_shape[1]
x_samples = h_shape[1]

# 3. C-scan interpolation
C_scan = np.zeros(shape=(t_samples, y_samples, x_samples))

for t in range(t_samples):

    for y in range(y_samples):
        C_scan[t, y, :] += np.interp(np.linspace(0, 1, num=x_samples),
np.linspace(0, 1, num=ver_scans), Vertical_scans[t, y, :])

        for x in range(x_samples):
            C_scan[t, :, x] += np.interp(np.linspace(0, 1, num=y_samples),
np.linspace(0, 1, num=hor_scans), Horizontal_scans[t, x, :])

# 4. Return C-scan
return C_scan

OUT_FILES_FOLDER = r'BSCANS FOLDER'
VERTICAL_BSCANS = r'vertical\\'

```

```
HORIZONTAL_BSCANS = r'horizontal\\'
```

```
C_SCAN_FILENAME = r'Real_Cscan'
```

```
PltColormap = 'seismic'
```

```
FlagShowPlt = True
```

```
FlagMakeFromData = True
```

```
z_start = 0
```

```
dx = 0.001
```

```
dy = 0.001
```

```
ver_Ascan_points = 30
```

```
hor_Ascan_points = 30
```

```
Vertical_Bscans = 11
```

```
Horizontal_Bscans = 10
```

```
Nt = 1050 - z_start
```

```
Vertical_scans_data = np.empty(shape=(Nt, ver_Ascan_points, Vertical_Bscans))
```

```
Horizontal_scans_data = np.empty(shape=(Nt, hor_Ascan_points,  
Horizontal_Bscans))
```

```
C_scan_data = np.zeros(shape=(Nt, ver_Ascan_points, hor_Ascan_points))
```

```
# Loading B-scans data from .npy files
```

```
for i in range(Vertical_Bscans):
```

```
    filename = OUT_FILES_FOLDER + VERTICAL_BSCANS + 'scan_' + str(i+1)  
    + '.npy'
```

```
    info, data = load_gpr(filename)
```

```
    Vertical_scans_data[:, :, i] += data[:, 0:-1:4]
```

```

for j in range(Horizontal_Bscans):
    filename = OUT_FILES_FOLDER + HORIZONTAL_BSCANS + 'scan_' +
    str(j+14) + '.npy'    info, data = load_gpr(filename)
    Horizontal_scans_data[:, :, j] += data[:, 0:-1:4]

if (FlagShowPlt):

    C_scan_data = Cscan_linear_interp(Horizontal_scans_data,
    Vertical_scans_data)

    max_abs_value = np.max(np.abs(C_scan_data))

    x_axis = np.arange(hor_Ascan_points)
    y_axis = np.arange(ver_Ascan_points)

    fig, ax = plt.subplots()
    plt.subplots_adjust(left=0.25, bottom=0.25)

    l = plt.imshow(C_scan_data[0], cmap='seismic', vmin=-max_abs_value,
    vmax=max_abs_value)

    contour_axis = fig.gca()
    contour_axis.set_xlabel("X-Axis")
    contour_axis.set_ylabel("Y-Axis")

    ax_z_color = 'lightgoldenrodyellow'
    ax_z = plt.axes([0.25, 0.1, 0.6, 0.03], facecolor=ax_z_color)
    #slider_z = Slider(ax_z, 'Z-coord ', 0, 29, valinit=0, valstep=1)
    slider_z = Slider(ax_z, 'Z-coord ', 0, Nt-1, valinit=0, valstep=1)

```

```
def update(val):  
    contour_axis.clear()  
    contour_axis.imshow(C_scan_data[slider_z.val], cmap='seismic', vmin=-  
max_abs_value, vmax=max_abs_value)
```

Приложение Е

Реализация алгоритма интерполяции методом обратного взвешивания по расстоянию.

```
from matplotlib.widgets import Slider
import matplotlib.pyplot as plt
import numpy as np
import os
```

```
def Cscan_IDW_interp(Horizontal_scans, Vertical_scans, dx = 1, dy = 1, radius = 3, ief = 2):
```

```
    """
```

Interpolate (3D) C-scan data each z-slice by weight sum of sample points in some specific radius from calculated point

Requiered parameters

`Horizontal\_scans`: `3D-ndarray like

Array of B-Scans data in XZ plane with shape (t, y, x)

`Vertical\_scans`: `3D-ndarray like

Array of B-Scans data in YZ plane with shape (t, y, x)

Optimal parameters

`dx`: `float like (default 1)

Distance between two samples in horizontal scans

`dy`: `float like (default 1)

Distance between two samples in vetrical scans

`radius:` float like (default 3)

Radius of interpolation

`ief:` int like (default 2)

*Inverse Exponential Factor. It influence on weigths, because
ief located in distance degree. $w = 1/d^{ief}$*

Return

*-----
Interpolated C-scan data like `3D-ndarray` with shape (t, y, x)*

"""

1. Shapes of data must be same sample size on time-axis

h_shape = Horizontal_scans.shape

v_shape = Vertical_scans.shape

if (h_shape[0] != v_shape[0]):

raise ValueError('Shapes of input B-scans on time-axis must be same.')

2. Calculate variables

hor_scans = h_shape[2]

ver_scans = v_shape[2]

t_samples = h_shape[0]

y_samples = v_shape[1]

x_samples = h_shape[1]

Distance between two similar B-scans in samples

y_sample_dist = y_samples // (hor_scans-1)

x_sample_dist = x_samples // (ver_scans-1)

```

# Points to interpolate
x_sample_arr = np.arange(x_samples)
y_sample_arr = np.arange(y_samples)
x_interp_arr = x_sample_arr[x_sample_arr % x_sample_dist != 0]
y_interp_arr = y_sample_arr[y_sample_arr % y_sample_dist != 0]

# This part calculated only once and return indexes and weights
# that will used next to interpolation grid values
radius_points = []

xr = int(radius/dx)
yr = int(radius/dy)

for x in range(-xr, xr+1):
    for y in range(-yr, yr+1):

        if (x == 0) and (y == 0): continue
        dist = distance(0, 0, x*dx, y*dy)

        if (dist <= radius):
            weight = 1/np.power(dist, ief)
            radius_points += [(x, y, weight)]

# 3. C-scan interpolation
C_scan = np.zeros(shape=(t_samples, y_samples, x_samples))

for t in range(t_samples):

    z_slice = np.zeros(shape=(y_samples+1, x_samples+1))

```

```

# z-slice has shape [y, x] like part of [t, y, x] array

# Horizontal component
for j in range(hor_scans):
    for i in range(x_samples):
        jx = j*y_sample_dist
        z_slice[jx, i] += Horizontal_scans[t, i, j]

# Vertical Component
for i in range(ver_scans):
    for j in range(y_samples):
        iy = i*x_sample_dist
        z_slice[j, iy] += Vertical_scans[t, j, i]

# Right-Bottom Corner Component
z_slice[-1, -1] = 0.5*z_slice[-2, -1] + 0.5*z_slice[-1, -2]

# Intersections
for i in range(0, y_samples, y_sample_dist):
    for j in range(0, x_samples, x_sample_dist):
        z_slice[i, j] *= 0.5

# Radial interpolation
for x0 in x_interp_arr:
    for y0 in y_interp_arr:
        total_sum = 0
        weight_sum = 0

        for x, y, weight in radius_points:
            xp = x0 + x

```

```

    yp = y0 + y
    if (xp)*(yp) < 0: continue
    if (xp > x_samples) or (yp > y_samples): continue
    if (xp % x_sample_dist == 0) or (yp % y_sample_dist == 0):
        total_sum += z_slice[yp, xp] * weight
        weight_sum += weight

    if (weight_sum != 0):
        z_slice[y0, x0] = total_sum / weight_sum
    else:
        z_slice[y0, x0] = 0

    # Set edited z_slice to the C_scan
    C_scan[t] = z_slice[:y_samples, :x_samples]

# 4. Return C-scan
return C_scan

OUT_FILES_FOLDER = r'BSCANS FOLDER'
VERTICAL_BSCANS = r'vertical\\'
HORIZONTAL_BSCANS = r'horizontal\\'

C_SCAN_FILENAME = r'Real_Cscan'

PltColormap = 'seismic'

FlagShowPlt      = True
FlagMakeFromData = True
z_start          = 0
dx               = 0.001

```

$dy = 0.001$

$ver_Ascan_points = 30$

$hor_Ascan_points = 30$

$Vertical_Bscans = 11$

$Horizontal_Bscans = 10$

$Nt = 1050 - z_start$

$Vertical_scans_data = np.empty(shape=(Nt, ver_Ascan_points, Vertical_Bscans))$

$Horizontal_scans_data = np.empty(shape=(Nt, hor_Ascan_points, Horizontal_Bscans))$

$C_scan_data = np.zeros(shape=(Nt, ver_Ascan_points, hor_Ascan_points))$

Loading B-scans data from .npy files

for i in range(Vertical_Bscans):

filename = OUT_FILES_FOLDER + VERTICAL_BSCANS + 'scan_' + str(i+1) + '.npy'

info, data = load_gpr(filename)

Vertical_scans_data[:, :, i] += data[:, 0:-1:4]

for j in range(Horizontal_Bscans):

filename = OUT_FILES_FOLDER + HORIZONTAL_BSCANS + 'scan_' + str(j+14) + '.npy' info, data = load_gpr(filename)

Horizontal_scans_data[:, :, j] += data[:, 0:-1:4]

if (FlagShowPlt):

$C_scan_data = Cscan_IDW_interp(Horizontal_scans_data, Vertical_scans_data, dx, dy, radius=16*dx, ief=2)$

```

max_abs_value = np.max(np.abs(C_scan_data))

x_axis = np.arange(hor_Ascan_points)
y_axis = np.arange(ver_Ascan_points)

fig, ax = plt.subplots()
plt.subplots_adjust(left=0.25, bottom=0.25)

l = plt.imshow(C_scan_data[0], cmap='seismic', vmin=-max_abs_value,
vmax=max_abs_value)

contour_axis = fig.gca()
contour_axis.set_xlabel("X-Axis")
contour_axis.set_ylabel("Y-Axis")

ax_z_color = 'lightgoldenrodyellow'
ax_z = plt.axes([0.25, 0.1, 0.6, 0.03], facecolor=ax_z_color)
#slider_z = Slider(ax_z, 'Z-coord ', 0, 29, valinit=0, valstep=1)
slider_z = Slider(ax_z, 'Z-coord ', 0, Nt-1, valinit=0, valstep=1)

def update(val):
    contour_axis.clear()
    contour_axis.imshow(C_scan_data[slider_z.val], cmap='seismic', vmin=-
max_abs_value, vmax=max_abs_value)

    contour_axis.set_xlabel("X-Axis")
    contour_axis.set_ylabel("Y-Axis")

plt.draw()

```

```
slider_z.on_changed(update)
```

```
plt.show()
```

```
# Generate .npy file
```

```
if (FlagMakeFromData):
```

```
    np.save(C_SCAN_FILENAME + '.npy', C_scan_data)
```