

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**
Кафедра комплексной информационной безопасности
электронно-вычислительных систем (ЭСАУ)

К ЗАЩИТЕ ДОПУСТИТЬ
Заведующий кафедрой ЭСАУ
д.т.н., профессор
_____ О.И. Черепанов
" ____ " 2015г.

**РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ОЦЕНКИ ДЕФОРМАЦИИ ПОВЕРХНОСТИ ТВЁРДЫХ ТЕЛ ПО СЕРИИ
ОПТИЧЕСКИХ ИЗОБРАЖЕНИЙ**

Дипломная работа
по специальности 220301 - Автоматизация технологических процессов и
производств

СОГЛАСОВАНО

Консультант по экономике
Доцент кафедры Экономики
канд. экон. наук
_____ О.П. Полякова
" ____ " 2015г.

Студент гр. 530
кафедры ЭСАУ
_____ И.Ю. Поляков
" ____ " 2015г.

Консультант по безопасности
жизнедеятельности
Доцент кафедры РЭТЭМ
канд. хим. наук
_____ И.А. Екимова
" ____ " 2015г.

Руководитель
Мл. н. с. лаборатории полимерных
композитных материалов ИФПМ СО
РАН
канд. техн. наук
_____ П.С. Любутин
" ____ " 2015г.

Реферат

Выпускная квалификационная работа 76 с., 20 рис., 12 табл., 11 источников, 2 прил.

ПОЛЕ ВЕКТОРОВ СМЕЩЕНИЙ, КОМПОНЕНТЫ ДЕФОРМАЦИИ, СУБПИКСЕЛЬНАЯ ТОЧНОСТЬ

Целью настоящей работы является разработка программного обеспечения (ПО) для оценки деформаций поверхностей твёрдых тел, а также проведение исследований алгоритмов и методов как на модельных, так и на реальных оптических изображениях.

В работе исследовано влияние метода интерполяции изображений с субпиксельной точностью с использованием итеративного подхода на расчёт оптического потока(векторного поля).

Исследованы методы для поиска смещений больше чем 20px.

Проект выполнен с использованием следующих средств разработки: языка программирования C++(Qt), среды разработки QtCreator 3, Sublime 3. Система контроля версий git.

Программный продукт, разработанный в ходе данной работы, представлен в приложении.

Отчёт выполнен согласно ОС ТУСУР 01-2013 при помощи текстового процессора L^AT_EX.

The abstract

Diploma work contains 76 pages, 20 pictures, 7 tables, 11 sources, 2 appendix, 5 sheets of a graphic material.

FIELD DISPLACEMENT VECTOR, STRAIN COMPONENTS, SUB-PIXEL ACCURACY

The aim of this work is the development of software (software) to evaluate the deformation of surfaces of solids, as well as research methods and algorithms on model, and the real optical images.

The influence of the method of interpolation image with sub-pixel precision using an iterative approach to the calculation of the optical flow (vector field).

Investigate methods for finding the displacement of more than 20px.

The project was implemented using the following development tools: the programming language C ++ (Qt), development environment QtCreator 3, Sublime 3. A version control system git.

The software, developed in the course of this work, presented in the annex.

The report is made in accordance with the OS TUSUR 01-2013 using a word processor L^AT_EX.

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**
Кафедра электронных средств автоматизации и управления (ЭСАУ)

УТВЕРЖДАЮ
зав. каф. ЭСАУ
_____ О.И. Черепанов
"___" 2015г.

ЗАДАНИЕ

на дипломную работу студенту Полякову Игорю Юрьевичу группы 530,
факультета вычислительных систем.

1. Тема работы: Разработка и тестирование программного обеспечения
для оценки деформации поверхности твёрдых тел по серии оптических изоб-
ражений.

2. Срок сдачи студентом законченной работы: "___" 2015г.

3. Исходные данные к работе:

- Программное обеспечение для оценки деформации “Deformation Analysis”;
- Дифференциальный алгоритм Лукаса–Канаде.

4. Содержание расчёто-пояснительной записи (перечень подлежащих разработке вопросов):

- введение;
- описание алгоритмов;
- проектирование программного обеспечения;
- тестирование;
- заключение.

5. Перечень графического материала:

- презентация.

6. Консультанты по работе

- консультант по экономике: доцент кафедры экономики, кандидат экономических наук

О.П. Полякова

" ____ " 2015г.

- консультант по вопросам охраны труда: доцент кафедры РЭТЭМ, кандидат химических наук

И.А. Екимова

" ____ " 2015г.

Задание выдано:

Руководитель: младший научный сотрудник ИФПМ СО РАН, кандидат технических наук

П.С. Любутин

" ____ " 2015г.

Задание принято к исполнению:

студент И.Ю. Поляков

" ____ " 2015г.

Содержание

Введение		8
1 Описание алгоритмов и методов для решения задачи		9
1.1 Описание дифференциального алгоритма определения оптического потока Лукаса–Канаде		9
1.2 Оценка деформации		18
2 Конструкторско-технологическая часть		20
2.1 Программная платформа и язык программирования		20
2.2 Структурно функциональная схема программного обеспечения .		22
2.3 Сведения о платформе реализации с указанием основных функций операционной системы, необходимых для работы модуля . .		24
2.4 Выбор формата выходных файлов		25
2.5 Документирование ПО		26
2.6 Блок схему оригинальных, разработанных автором, алгоритмов работы основных, по мнению автора, программных модулей . . .		26
2.7 Интерфейс разрабатываемого программного обеспечения . . .		26
3 Тестирование программного обеспечения		32
3.1 Образцы изображений		32
3.2 Тестирование программного обеспечения		33
3.3 Исследование метода интерполяции		34
3.4 Исследование размера окна поиска		34
4 Безопасность жизнедеятельности		35
4.1 Анализ опасных и вредных производственных факторов на рабочем месте		35
4.2 Требования и защитные мероприятия в области БЖД		38
4.3 Требования эргономики и технической эстетики		47
4.4 Общие требования безопасности перед началом, вовремя, по окончании работы и в случае аварийных ситуаций		49
5 Технико-экономическое обоснование		52
5.1 Обоснование необходимости проводимого исследования . . .		52
5.2 Планирование комплекса работ по разработке программного обеспечения		52
5.3 Определение сметной стоимости проекта		54
5.4 Научно-технический эффект		58

5.5 1.5 Социальный эффект	59
6 Заключение	60
Список использованных источников	61
Приложение А Текст графического модуля	62
Приложение Б Документация программного обеспечения	64

Введение

Использование систем технического зрения (СТЗ) для определения параметров деформации представляется в настоящее время как наиболее перспективное, поскольку позволяет предсказывать места локализации деформации и концентрации критических напряжений задолго до возникновения микротрешин.

СТЗ оптико-телеизионных измерительных комплексов по принципу действия относятся к обзорно-сравнительным системам к классу корреляционных зрительных систем (КЗС) [1]. СТЗ данного типа находят широкое применение в различных системах промышленной автоматизации, в том числе и в робототехнических системах.

Экспериментальное изучение процессов пластической деформации, развивающихся в материалах и при различных условиях эксплуатации, позволит дать рекомендации по новым методам упрочнения, созданию оптимальных типов покрытий, соотношению механических свойств покрытия и матрицы (основы), составам композиционных материалов. Проведение таких экспериментальных исследований на мезоуровне требовало создания новых аппаратных и программных средств, в качестве которых могут выступать методы и средства технического зрения, включающие автоматизированные измерительные комплексы, способные оперативно выполнять обработку больших объемов видеинформации, производить высокоточные измерения, качественно и наглядно представлять полученные результаты.

Целью настоящей работы является разработка программного обеспечения (ПО), рассмотрение алгоритмов для оценки деформаций поверхностей твердых тел, т.е. для получения качественной и количественной оценки процессов, развивающихся в деформируемом твердом теле на мезоуровне, а также проведение исследований алгоритмов и методов на реальных оптических изображениях.

Область применения ПО: 1. экспериментальное исследование механизмов деформации и разрушения структурно-неоднородных материалов с целью последующего компьютерного моделирования их структуры и свойств; 2. аттестация режимов формирования защитных и упрочняющих покрытий с целью корректировки режимов их нанесения; 3. тарировка приборов неразрушающего контроля и дефектоскопии.

1 Описание алгоритмов и методов для решения задачи

Все современные дифференциальные алгоритмы слежения за особенностями опираются на работу 1981 году Лукаса и Канаде [2]. В 1991 году математическая формулировка этого алгоритма была изменена, и стала основой для всех последующих обобщений с учётом аффинных искажений окрестности и освещённости. Путём замены соответствующих переменных на константы любой из них превращается в обычный алгоритм Лукаса–Канаде [3].

1.1 Описание дифференциального алгоритма определения оптического потока Лукаса–Канаде

В основе всех дальнейших рассуждений лежит одно очень важное и не очень справедливое предположение: Предположим, что значения пикселей переходят из одного кадра в следующий без изменений. Таким образом, мы делаем допущение, что пиксели, относящиеся к одному и тому же объекту, могут смещаться в какую либо сторону, но их значение останется неизменным. Конечно же это предположение имеет мало общего с реальностью, потому что от кадра к кадру могут меняться глобальные условия освещения и освещённость самого движущегося объекта. Масса проблем связана с этим допущением, но, как ни странно, вопреки всему оно достаточно хорошо работает на практике. На математическом языке это допущение можно записать так:

$$I(x, y, t) = I(x + u_x, y + u_y, t + 1) \quad (1.1)$$

где I — это функция яркости пикселей от положения на кадре и времени. Другими словами x и y — это координаты пикселя в плоскости кадра, u — это смещение, а t — это номер кадра в последовательности. Условимся, что между двумя соседними кадрами проходит единичный отрезок времени.

1.1.1 Одномерный случай алгоритма

Для начала рассмотрим одномерный случай. Рассмотрим представленные на рисунке 1.1 два одномерных кадра размером 1x20 пикселей. На рисунке 1.1 кадр f_2 смешено вправо на 4 пикселя. Именно это смещение необходимо найти. Для этого представим эти же кадры в виде функций (рисунок 1.2). На входе позиция пикселя, на выходе — его интенсивность. В таком представле-

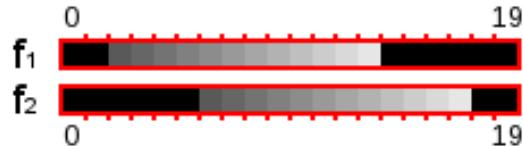


Рисунок 1.1 – Одномерный сдвиг яркостей

ние искомое смещение (d) видно еще более наглядно. В соответствии с нашим предположением, f_2 это просто смещённая f_1 , то есть можно сказать, что:

$$f_2(x) = f_1(x - d) \quad (1.2)$$

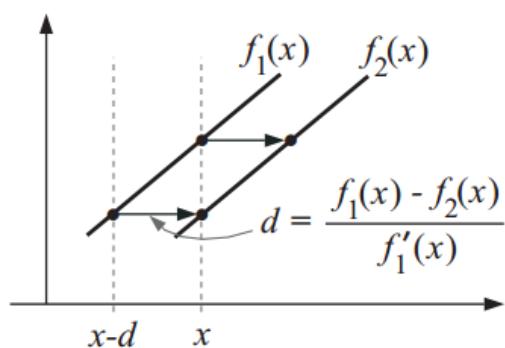


Рисунок 1.2 – Функция зависимости интенсивности от позиции пикселя

Обратите внимание, что f_1 и f_2 при желании можно записать и в общем виде: $f_1(x) = I(x, y, t)$ $f_2(x) = I(x, y, t)$; где y и t зафиксированы и равны нулю.

Для каждой координаты нам известны значения и в этой точке, кроме того мы можем вычислить их производные. Связем известные значения со смещением d . Для этого запишем разложение в ряд Тейлора для $f_1(x - d)$:

$$f_1(x - d) = f_1(x) - df_1'(x) + O(d^2 f_1'')$$

Сделаем второе важное предположение: Предположим, что достаточно хорошо аппроксимируется первой производной. Сделав это предположение, отбросим всё что после первой производной:

$$f_1(x - d) = f_1(x) - df_1'(x)$$

Насколько это корректно? В общем—то не очень, тут мы теряем в точности, если только наша функция/изображение не строго линейна, как в нашем

искусственном примере. Зато это существенно упрощает метод, а для достижения требуемой точности можно сделать последовательное приближение, которое мы рассмотрим позже.

Мы почти у цели. Смещение d — это наша искомая величина. Исходя из 1.2, сделаем замену: $f_2(x) = f_1(x) - df'_1(x)$ откуда выведем $d = \frac{f_1(x) - f_2}{f'_1(x)}$

1.1.2 Двумерный случай алгоритма

Теперь перейдём от одномерного случая к двумерному. Запишем разложение в ряд Тейлора для 1 и сразу отбросим все старшие производные. Вместо первой производной появляется градиент:

$$I(x + u_x, y + u_y, t + 1) = I(x, y, t) + \vec{u} \nabla I(x, y, t)$$

где $\vec{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$ — вектор смещения. В соответствии со сделанным допущением 1. Перепишем:

$$I(x, y, t) - I(x, y, t + 1) + \vec{u} \nabla I(x, y, t) = 0$$

Поскольку между двумя кадрами проходит единичный интервал времени, то можно сказать, что $I(x, y, t) - I(x, y, t + 1)$ есть не что иное, как производная по времени. Заменим:

$$\frac{\partial I(x, y, t)}{\partial t} + \vec{u} \nabla I(x, y, t) = 0$$

Перепишем ещё раз, раскрыв градиент:

$$\frac{\partial I(x, y, t)}{\partial t} + u_x \frac{\partial I(x, y, t)}{\partial x} + u_y \frac{\partial I(x, y, t)}{\partial y} = 0$$

Мы получили уравнение, которое говорит нам о том, что сумма частных производных должны быть равна нулю. Проблема только в том, что уравнение у нас одно, а неизвестных в нем два: u_x и u_y . На этом моменте начинается полет фантазии и разнообразие подходов.

Сделаем третье предположение: Предположим, что смещение пикселей между двумя кадрами невелико. Рассмотрим пиксель p , тогда, по алгоритму

Лукаса — Канаде, оптический поток должен быть одинаков для всех пикселей, находящихся в окне с центром в p . А именно, вектор оптического потока u_x и u_y в точке p должен быть решением системы уравнений. Очевидно, что в общем случае система не имеет решения, поэтому будем искать такие u_x и u_y , которые минимизируют ошибку:

$$\begin{cases} I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \\ \dots \\ I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \end{cases}$$

где q_1, q_2, \dots, q_n — пиксели внутри окна $I_x(q_i), I_y(q_i), I_t(q_i)$ — частные производные изображения I по координатам x, y и времени t , вычисленные в точке q_i . Это уравнение может быть записано в матричной форме:

$$Av = b$$

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

Полученную переопределенную систему решаем с помощью метода наименьших квадратов. Таким образом, получается система уравнений $2 \cdot 2$:

$$A^T A v = A^T b$$

откуда

$$v = (A^T A)^{-1} A^T b$$

где A^T — транспонированная матрица A . Получаем:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (1.3)$$

Вот собственно и все. Мы знаем приблизительное смещение пикселей между двумя соседними кадрами.

Поскольку в нахождении смещения каждого пикселя участвуют также соседние с ним пиксели, при реализации данного метода целесообразно предварительно посчитать производные кадра по горизонтали и вертикали.

1.1.3 Итеративный подход и субпиксельная точность

Описанный выше метод основан на трёх значительных допущениях, которые с одной стороны дают нам принципиальную возможность определить оптический поток, но с другой стороны вносят погрешность.

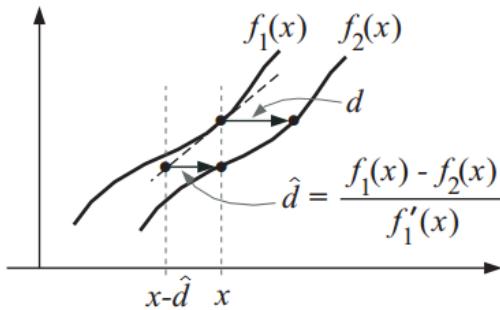


Рисунок 1.3 – Функция зависимости интенсивности от позиции пикселя

Одно из допущений нужно нам только для упрощения метода, и с его последствиями мы можем бороться. Мы предполагали, что для аппроксимации смещения нам будет достаточно первой производной. В общем случае это конечно же не так (рисунок 1.3).

Для достижение требуемой точности смещение для каждой пары кадров (назовём их F_i и F_{i+1}) можно вычислять итеративно. В литературе это называется искажением (warping). На практике это означает, что, вычислив смещения на первой итерации, мы перемещаем каждый пиксель кадра в противоположную сторону так, чтобы это смещение компенсировать.

На следующей итерации вместо исходного кадра F_{i+1} мы будем использовать его искажённый вариант F_{i+1}^1 . И так далее, пока на очередной итерации

все полученные смещения не окажутся меньше заданного порогового значения. Итоговое смещение для каждого конкретного пикселя мы получаем как сумму его смещений на всех итерациях.

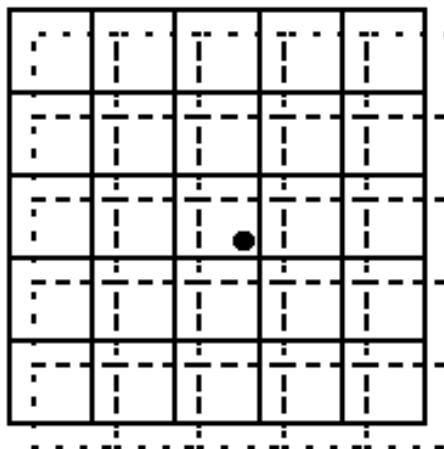


Рисунок 1.4 – Окрестность с субпиксельной точностью

Возникает одна проблема, искажённая область дискретна и состоит из пикселей. Мы не можем сместить область на значение x, y , если оно лежит в интервале $0 < |x, y| < 1$. Решением этой проблемы является — интерполяция или субпиксельная точность (рисунок 1.4).

В работе описывающей Пирамидальную версию алгоритма [4], которая подробнее описана в разделе 1.1.5, предлагалось использовать билинейную интерполяцию однако, хотелось бы попробовать другие способы интерполяции, и посмотреть на выходные результаты.

Методы интерполяции использованные для разработки и последующего тестирования:

- Билинейная
- Бикубическая
- Б-сплайн

<http://algolist.manual.ru/graphics/3dfa/q/articles/71.php>

По своей природе данный метод является локальным, то есть при определении смещения конкретного пикселя принимается во внимание только область вокруг этого пикселя — локальная окрестность. Как следствие, невозможно определить смещения внутри достаточно больших (больше размера локальной окрестности) равномерно окрашенных участков кадра. К счастью на реальных кадрах такие участки встречаются не часто, но эта особенность все же вносит

дополнительное отклонение от истинного смещения.

Если интервал времени между кадрами принять за 1, получается следующий алгоритм:

$$\begin{cases} d_k = 0 & \text{if } k = 0 \\ d_{k+1} = d_k + C^{-1} \sum_W [(I(x, t) - I(x + d_k, t + 1)) \nabla I(x, t)] & \text{if } k > 0 \end{cases}$$

Таким образом, этот алгоритм слежения фактически является поиском точки, в которой достигается минимум некоторой функции, методом градиентного спуска. Во время каждой итерации мы сдвигаемся вдоль направления градиента изображения в текущей точке.

1.1.4 Учёт аффинных преобразований

Перемещение объекта на входящем видео потоке, описанное в разделе 1.1, одно из простейших видов движения. Существуют более сложные: сжатие, увлечение, вращение и т.д. По сути это преобразование плоскостей, называемое аффинным. При условии:

- оно взаимно однозначно;
- образом любой прямой является прямая.

Алгоритм с учётом аффинных искажений хорошо описан в работе Ши Томаси канаде в 1994 году [5]. Движение пикселей окна описывается в виде $Ax + d$, где A — матрица $(2 \cdot 2)$, d — смещение $(2 \cdot 1)$.

Задача слежения за особенностью сводится к проблеме определения параметров движения и искажения окна особенности, при которой минимизируется разность:

$$r = \iint_W [J(\Delta x + d) - I(x)]^2$$

где W — окно особенности, а w — весовая функция (может использовать, а может и быть равна 1 во всем окне), $J(x)$ и $I(x)$ — два изображения.

Выражение дифференцируется относительно параметров движения, и производная приравнивается к 0. Затем система линеаризуется с помощью разложения функции изображения в ряд Тейлора:

$$J(\Delta x + d) = J(x) + g^T(u)$$

Это даст нам линейную $6 \cdot 6$ систему:

$$Tz = a$$

, где в векторе z объединены все искомые параметры:

$$z^T = [d_{xx} \ d_{yx} \ d_{xy} \ d_{yy} \ d_x \ d_y]$$

Вектор ошибки a записывается в виде:

$$a = \iint_W [I(x) - J(x)] \begin{bmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{bmatrix} \omega dx$$

А матрицу размерности $6 \cdot 6$ T можно представить следующим образом:

$$T = \iint_W [I(x) - J(x)] \begin{bmatrix} U & V \\ V^T & Z \end{bmatrix} \omega dx$$

$$U = \begin{bmatrix} x^3 g_x^3 & x^3 g_x g_y & x y g_x^3 & x y g_x g_y \\ x^3 g_x g_y & x^3 g_y^3 & x y g_x g_y & x y g_x^3 \\ x y g_x^3 & x y g_x g_y & y^3 g_x^3 & y^3 g_x g_y \\ x y g_x g_y & x y g_y^3 & y^3 g_x g_y & y^3 g_y^3 \end{bmatrix}$$

$$V^T = \begin{bmatrix} x g_x^3 & x g_x g_y & y g_x^3 & y g_x g_y \\ x g_x g_y & x g_y^3 & y g_x g_y & y g_y^3 \end{bmatrix}$$

$$Z = \begin{bmatrix} g_x^3 & g_x g_y \\ g_x g_y & g_y^3 \end{bmatrix}$$

Полученная система решается также итеративно по методу Ньютона—Рафсона.

Если движение считается не аффинным, а просто смещением, то первые четыре элемента искомого вектора z обращаются в 0, и значимыми остаются только последние два. Соответственно все сводится к алгоритму

Лукаса–Канаде.

1.1.5 Пирамидальная версия алгоритма

Пирамидальная версия или иерархический метод. В данном алгоритме важным является нахождение хорошего начального приближения для вектора скорости. Для этого обычно применяют пирамидальную версию алгоритма.

Ее идея заключается в том, что наряду с исходной парой изображений (F_i и F_{i+1}) рассматривают эти же изображения сжатые в два раза ($\frac{F_i}{2}$ и $\frac{F_{i+1}}{2}$), в четыре ($\frac{F_i}{4}$ и $\frac{F_{i+1}}{4}$) и т.д. (рисунок 1.5).

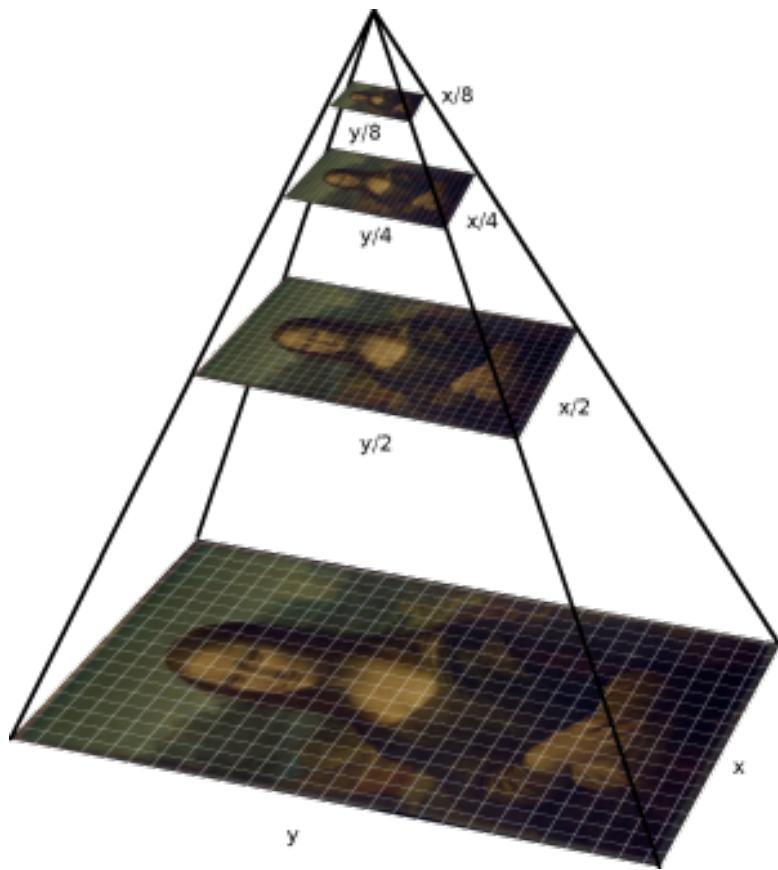


Рисунок 1.5 – Пример пирамиды Гаусса

Вектора скорости находят сначала на самом верхнем уровне пирамиды и затем спускаются вниз этаж за этажом. На самом верхнем уровне в качестве начального приближения берут нулевой вектор. На нижних уровнях за начальное приближение берут удвоенную скорость, полученную на предыдущем шаге. Все это вместе взятое обеспечивает хорошее сочетание скорости, точности и устойчивости алгоритма нахождения меж-кадрового движения в виде сдвигов [4]. Для самого первого уровня этот вектор принимается равным $(0, 0)$.

1.2 Оценка деформации

Деформация — изменение размеров и формы твёрдого тела под действием внешних сил (нагрузок) или каких-либо действий (например, температуры, электрических или магнитных полей).

При деформации точки твёрдого тела меняют своё положение. Точка с радиус-вектором r при деформации имеет новое положение r' , то есть осуществит перемещения $u = r' - r$.

Поле перемещений является одной из характеристик деформации, но оно неудобное для математического описания, поскольку, например, при удлинение стержня точки у его начала смещаются совсем мало, а в конце — довольно значительно. Гораздо важнее то, насколько точка тела сместились относительно соседней. Поэтому деформацию математически удобнее описывать производными от перемещения, которые образуют тензор, получивший название тензора деформации.

Виды деформаций

1.2.1 Линейная деформация Одноосный случай

Проявляется в растяжении—сжатии стержня вдоль его оси. Если выбрать в ненагруженном стержне два сечения, расположенных на определённом расстоянии и приложить к нему внешние силы, то расстояние между сечениями изменится.

Линейная деформация ε в произвольной точке тела является границей отношение прироста длины δL к исходной длине L , когда сама длина стремится к нулю.

$$\varepsilon = \lim_{L \rightarrow 0} \frac{\delta L}{L}$$

То есть при определении деформации в точке рассматриваются изменения в ее непосредственном окрестности. Общий случай

Для произвольного тела, испытывающего произвольного деформирования значение линейных деформаций может отличаться в зависимости от направления, в котором они рассматриваются. В этом случае линейные деформации рассматриваются в проекциях на оси декартовых координат. Тогда деформация отрезка АВ, лежит на оси x и точка В которая после деформации переместится

в т. В' запишется как:

$$\varepsilon_x = \lim_{B \rightarrow A} \frac{|AB'| - |AB|}{|AB|}$$

Проведя подобный анализ для оси y можно получить соответственно ε_y .

Имея это поле перемещений \vec{u} (компоненты вектора перемещений для всех точек тела) запишем компоненты тензоров деформации для декартовой системы координат.

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{bmatrix}$$

Компоненты рассчитываются путём численного дифференцирования полученного поля смещений. Выражения для продольной ε_{xx} , поперечной ε_{yy} , сдвиговой ε_{xy} и поворотной ω_z компонент тензора дисторсии:

$$\varepsilon_{xx} = \frac{dU_x}{dx} \quad (1.4)$$

$$\varepsilon_{yy} = \frac{dU_y}{dy} \quad (1.5)$$

$$\varepsilon_{xy} = \frac{1}{2} \left(\frac{dU_x}{dy} + \frac{dU_y}{dx} \right) \quad (1.6)$$

$$\omega_z = \frac{1}{2} \left(\frac{dU_y}{dx} - \frac{dU_x}{dy} \right) \quad (1.7)$$

Также выражение для интенсивности деформации сдвига γ_i

$$\gamma_i = \sqrt{\frac{2}{3}} \sqrt{(\varepsilon_{xx} - \varepsilon_{yy})^2 + \varepsilon_{yy}^2 + \varepsilon_{xy}^2 + \frac{3}{2} \varepsilon_{xy}^2} \quad (1.8)$$

2 Конструкторско-технологическая часть

2.1 Программная платформа и язык программирования

Для решения поставленной задачи необходимо использовать функциональную, эффективную и удобную платформу для разработки. В качестве такой платформы была выбрана среда разработки Qt Creator. Qt - это кроссплатформенная библиотека C++ классов для создания графических пользовательских интерфейсов (GUI) от фирмы Digia. Эта библиотека полностью объектно-ориентированная, что обеспечивает лёгкое расширение возможностей и создание новых компонентов. Ко всему прочему, она поддерживает огромнейшее количество платформ.

Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Список использованных классов фреймворка Qt:

- QDebug;
- QDir;
- QFile;
- QTextStream;
- QString;
- *QImage.

Класс QDebug обеспечивает выходной поток для отладочной информации.

Класс QDir обеспечивает доступ к структуре каталогов и их содержимого.

Класс QFile предоставляет интерфейс для чтения и записи файлов.

Класс QTextStream предоставляет удобный интерфейс для чтения и записи текста.

Класс QString обеспечивает строку символов Unicode.

Класс QImage предоставляет аппаратно-независимую работу с изображениями, даёт прямой доступ к данным каждого пикселя, и может быть использован в качестве устройства рисования. На ранних стадиях создание ПО это класс использовался в качестве основного инструмента для работы с изображениями, но впоследствии при переходе на основные модули библиотеки DV стал использовался объект Data2Db, и методы работы с ними.

Список стандартных библиотек используемых в проекте:

- getopt.h;
- iostream;
- math.h;
- stdlib.h;
- vector.

`getopt` библиотечная функция, специально разработанная для того чтобы облегчить обработку входных команд.

`iostream` заголовочный файл с классами, функциями и переменными для организации ввода-вывода в языке программирования C++. Он включён в стандартную библиотеку C++. Название образовано от Input/Output Stream («поток ввода-вывода»).

`math.h` — заголовочный файл стандартной библиотеки языка программирования C, разработанный для выполнения простых математических операций.

`stdlib.h` — заголовочный файл стандартной библиотеки языка Си, который содержит в себе функции, занимающиеся выделением памяти, контроль процесса выполнения программы, преобразования типов и другие.

`vector.h` — это замена стандартному динамическому массиву, память для которого выделяется вручную, с помощью оператора `new`.

Список классов и методов библиотеки DV используемых в проекте:

- Data2Db
- Matx22d
- Vec2d
- VF2d

2.1.1 Использование системы контроля версий

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возмож-

ность вернуться к определённым старым версиям этих файлов.

Для разработки программного комплекса решено использовать Git.

Git — распределённая система управления версиями файлов. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, как противоположность системе управления версиями Subversion (также известная как «SVN») [8].

Преимущества использования системы версий, очевидны:

- возможность возвращать к прежнему состоянию отдельные файлы или весь проект;
- возможность удалённой работы с текстами программ;
- доступ к последним изменениям в коде, т.к. всё хранится на сервере github.com;
- тексты программ открыты, доступ к ним можно получить доступ в интернет;
- просматривать происходящие со временем изменения.

Основные постулаты работы в системе Git:

- каждая задача решается в своей ветке;
- сохраняем изменения сразу, как что-то получили осмысленное;
- все сохранённые изменения должны быть осмысленно подписаны/прокомментированы.

Для работы над проектом был использован репозиторий на сервере github.com. Следок последних изменений из репозитория можно взять по адресу: `git clone git@github.com:IgorPolyakov/graduate-work.git`

2.2 Структурно функциональная схема программного обеспечения

Программное обеспечение состоит из нескольких блоков, представленных на рисунке 2.2.

- ПО для оценки деформации “Deformation Analysis”
- Форма для ввода входных параметров “strain_calculate.qml”
- ПО для расчёта деформации твердого тела “lucas_kanade”

Программа “Deformation Analysis” разрабатывалась сотрудниками ИФПМ и служит средством для запуска и отображения выходных результатов. Форма “strain_calculate.qml” является более удобным способом запуска основного приложения.

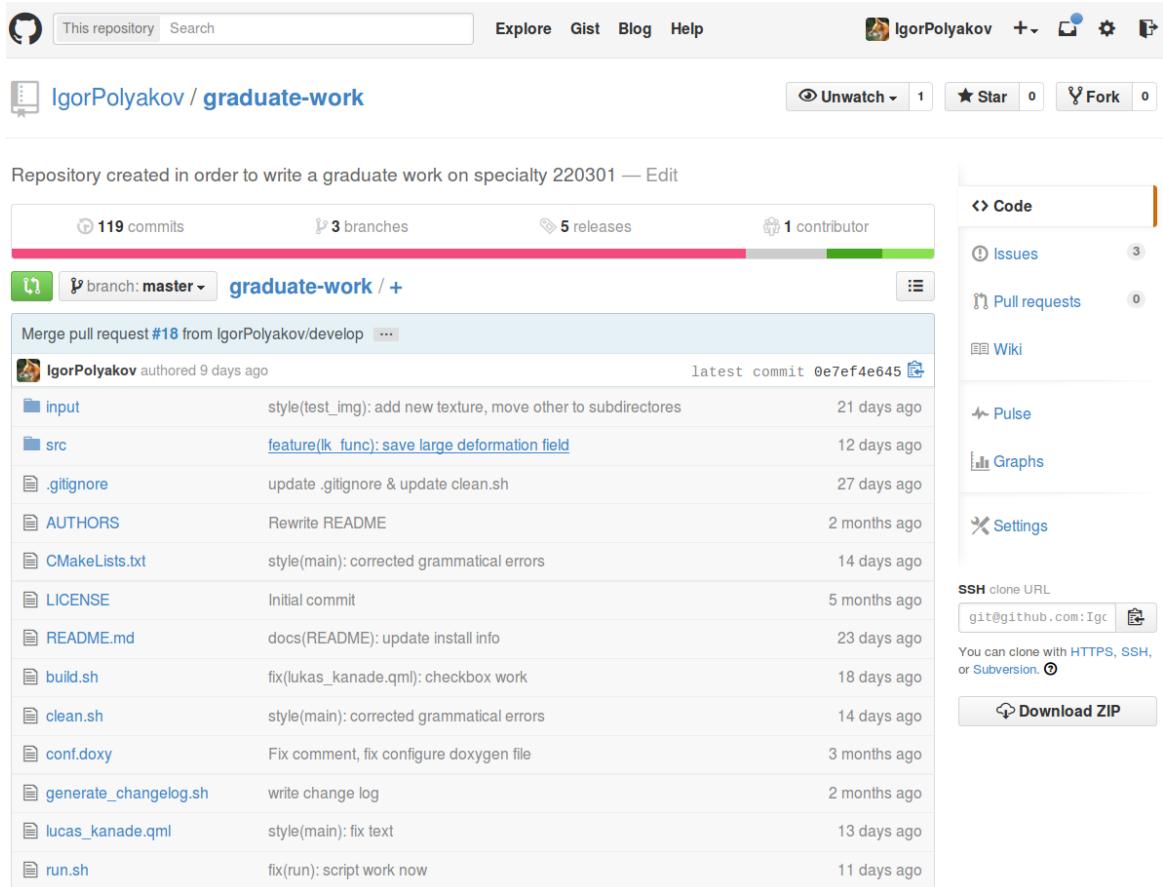


Рисунок 2.1 – Тексты программ в репозитории github

Рисунок 2.2 – Структура разработанного ПО

- 1) укрупненная структурно функциональная схема программного обеспечения, в составе которого работает модуль. Разрабатываемый модуль должен быть визуально выделен на общей схеме (обведен штриховой рамкой, обозначен другим цветом и т.д.);
- 2) структурно функциональная схема разрабатываемого программного модуля с обозначением входящих в него функциональных элементов и связей между ними. В связях надлежит доступными средствами выделить различные виды информационных потоков: символьные и кодовые массивы, бинарные сигналы индикации и управления, событийную информацию;
- 3) основные математические соотношения в виде формул и выражений (при разработке вычислительных программ не более, чем 1 плакат формата А1);
- 4) блок схема алгоритма работы модуля с достаточной степенью детализации (при наличии в разработке оригинальных и неочевидных алгоритмических решений);

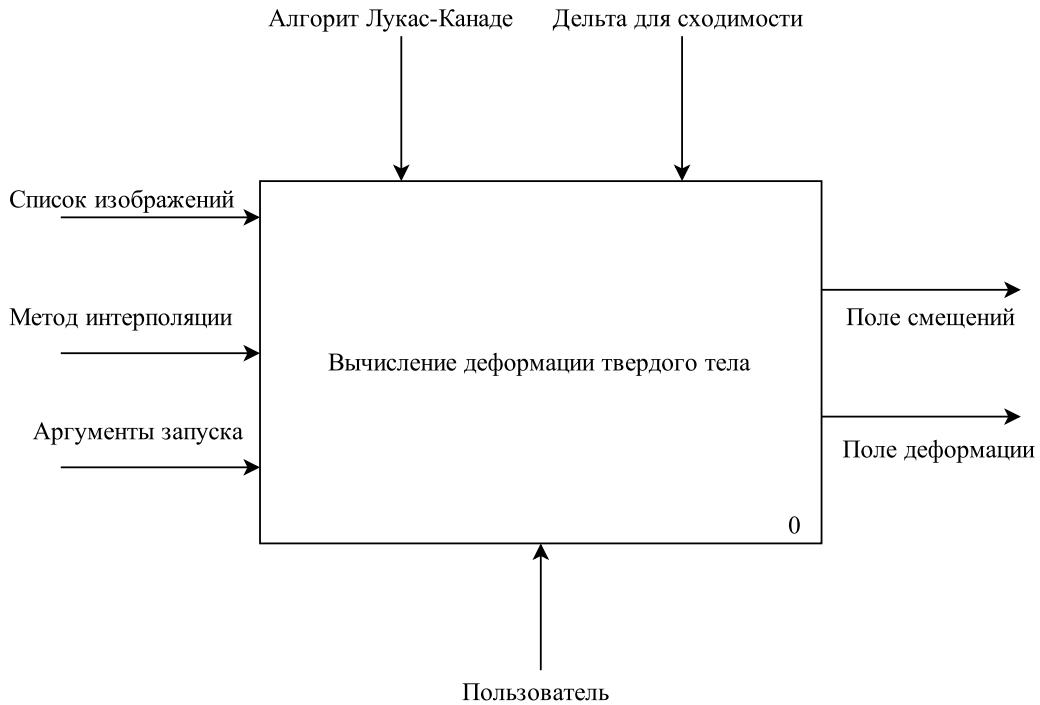


Рисунок 2.3 – Функциональная схема

- 5) изображения экранных форм в различных режимах работы программы(при разработке интерфейсных модулей);
- 6) материал, иллюстрирующий работу программы на тестовом или реальном примере, с использованием графиков, таблиц и пр.

2.3 Сведения о платформе реализации с указанием основных функций операционной системы, необходимых для работы модуля

2.3.1 Минимальные требования к аппаратному обеспечению

Минимальные системные требования [6]:

- процессор 1ГГц Pentium 4;
- оперативная память 512 Мб;
- место на жёстком диске – 9 ГБ.

2.3.2 Минимальные требования к программному обеспечению

Для корректной работы разрабатываемого программного комплекса на компьютере должна быть установлена операционная система Debian Squeeze или выше, данная система должна иметь набор библиотек Qt, CMake и DV.



Рисунок 2.4 – Структура HDF5 файла

2.4 Выбор формата выходных файлов

Для вывода результата был выбран формат файла HDF5.

Hierarchical Data Format, HDF (Иерархический формат данных) — название формата файлов, разработанного для хранения большого количества цифровой информации.

HDF5 — современная версия формата содержит иерархию из двух основных типов объектов: HDF-Structure-Example

- 1) Datasets — наборы данных, многомерные массивы объектов одного типа
- 2) Groups — группы, являются контейнерами для наборов данных и других групп

Содержимое файлов HDF5 организовано подобно иерархической файловой системе, и для доступа к данным применяются пути, сходные с POSIX-синтаксисом, например, /path/to/resource. Метаданные хранятся в виде набора именованных атрибутов объектов.[7]

Структура сохранения результатов работы представлена на рисунке 2.4

Под блоком исходные обрабатываемые изображения подразумевается входная пара изображений(левая, правая) в форматах png, bmp, jpg, tiff и др. В выходном файле они представлены двумя слоями: “left” и “right”. Они необходимы для удобства просмотра в программе Defomation-Analysis. Программа Defomation-Analysis умеет выводить отдельные слои и накладывать их друг на друга.

Блок векторное поле смещений является бинарным файлом, формата “vf”. Слой содержит информацию о размерах исходного изображения, координаты векторов смещений и векторы в формате double (формула 1.3).

Блок поля деформации также является бинарным файлом и содержит слои деформации:

- 1) ε_{xx} - продольная (формула 1.4);
- 2) ε_{yy} - поперечная (формула 1.5);
- 3) ε_{xy} - сдвиговая (формула 1.6);
- 4) w_i - поворотная (формула 1.7);
- 5) γ_z - интенсивность деформации сдвига (формула 1.8).

2.5 Документирование ПО

Для облегчения написания документации к текстам программ, можно воспользоваться генератором документации. Так как у автора работы есть опыт использования системы Doxygen, то она и будет использована.

На вход такого генератора поступает специальным образом комментированный текст программы, а иногда и другие компоненты программы, а на выходе создаётся готовая документация для распространения и использования.

Используемая система Doxygen как раз и выполняет эту задачу: она позволяет генерировать на основе исходного кода, содержащего комментарии специального вида, красивую и удобную документацию, содержащую в себе ссылки, диаграммы классов, вызовов и т.п. в различных форматах: HTML, LaTeX, CHM, RTF, PostScript, PDF, man-страницы.

Документация собранная системой Doxygen представления в приложении.

2.6 Блок схему оригинальных, разработанных автором, алгоритмов работы основных, по мнению автора, программных модулей

Основной алгоритм работы представлен на рисунке 2.5. Для удобства, крупное блоки вынесены в отдельные блок-схемы и представлены на рисунках 2.6 и т.д.

2.7 Интерфейс разрабатываемого программного обеспечения

2.7.1 Описание консольного интерфейса программы

Интерфейс программы представлен в двух реализациях. Первый — консольная программа в стиле классического Unix. Интерфейс командной строки более гибкий, позволяет выставить необходимые опции/флаги и запустить программу. Особенности в сравнении с графическим интерфейсом:

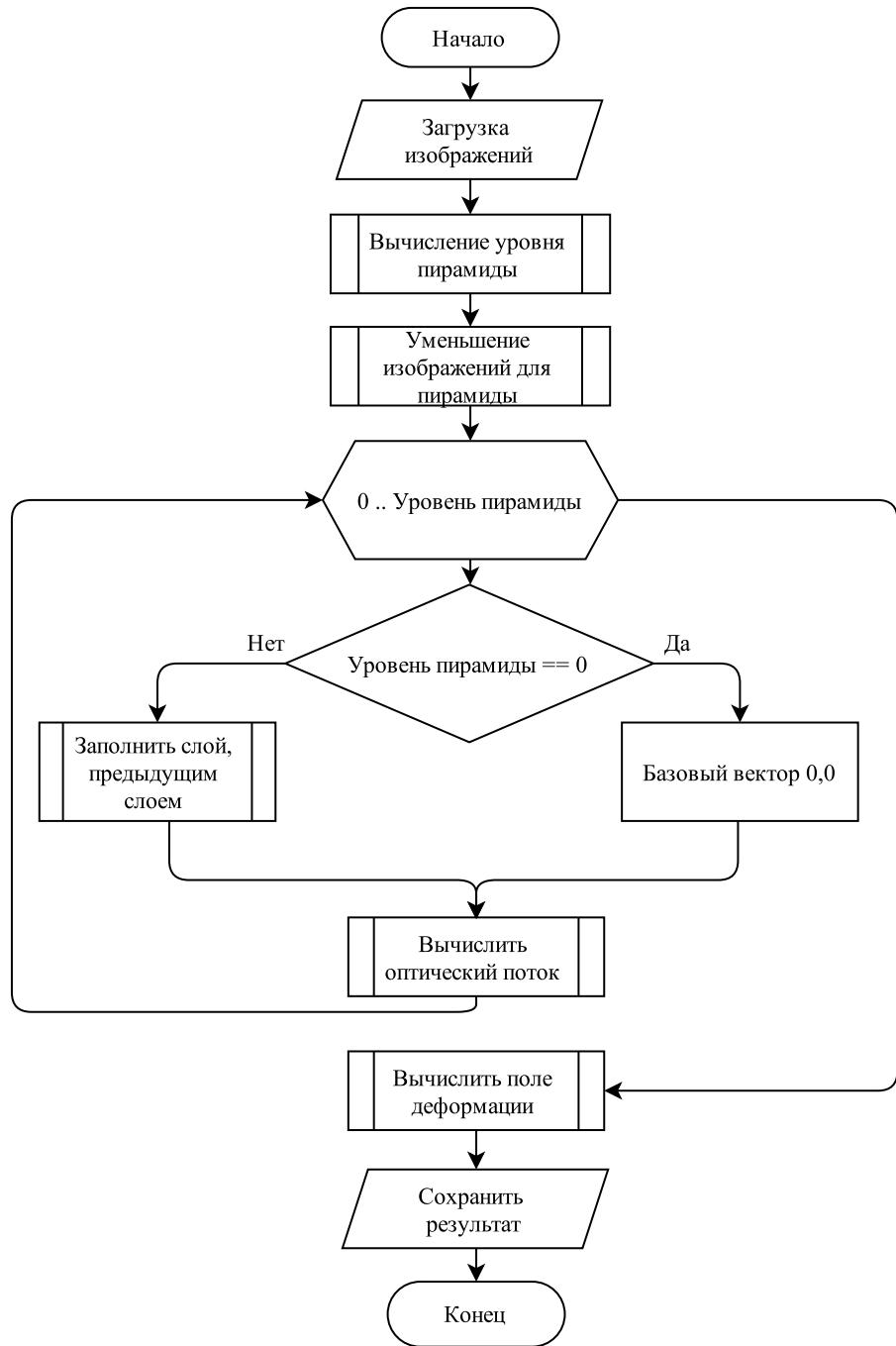


Рисунок 2.5 – Общий алгоритм работы

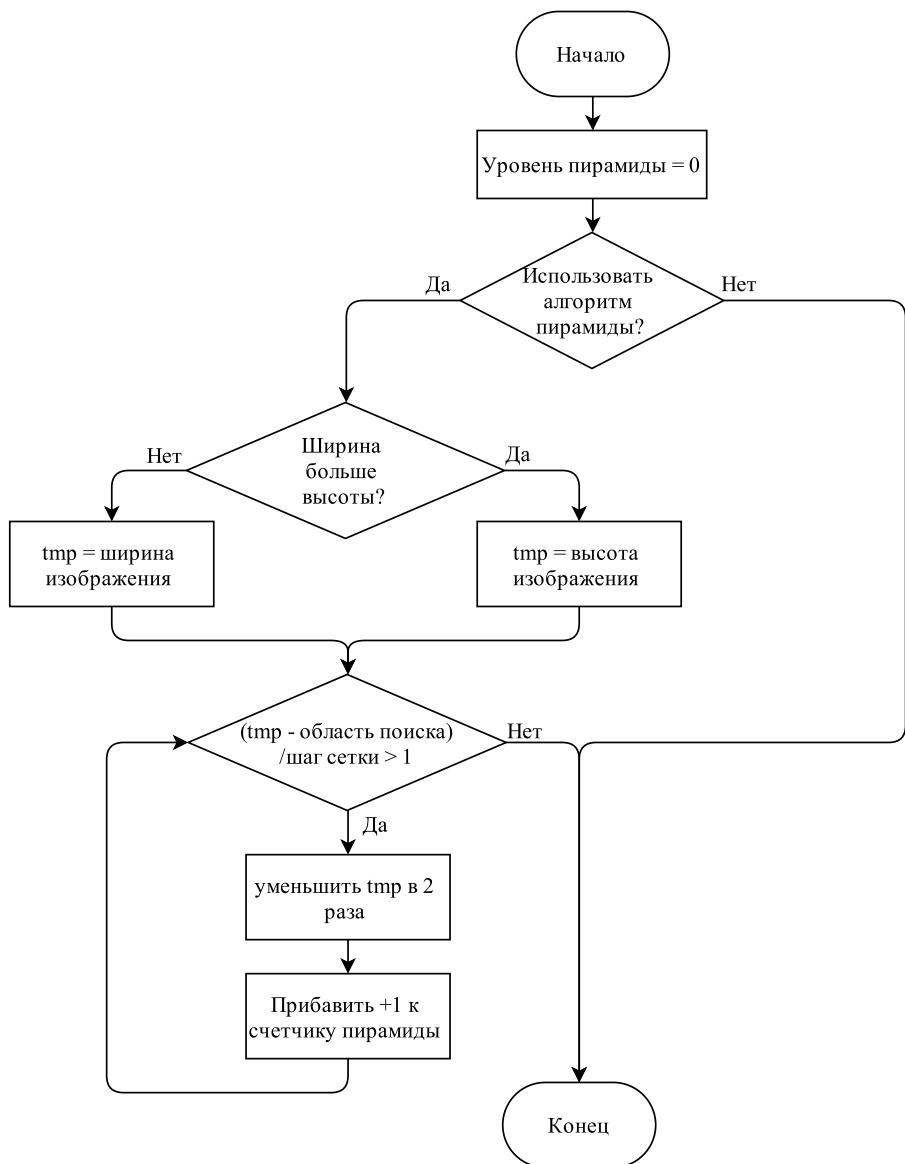


Рисунок 2.6 – Алгоритм вычисления уровня пирамиды

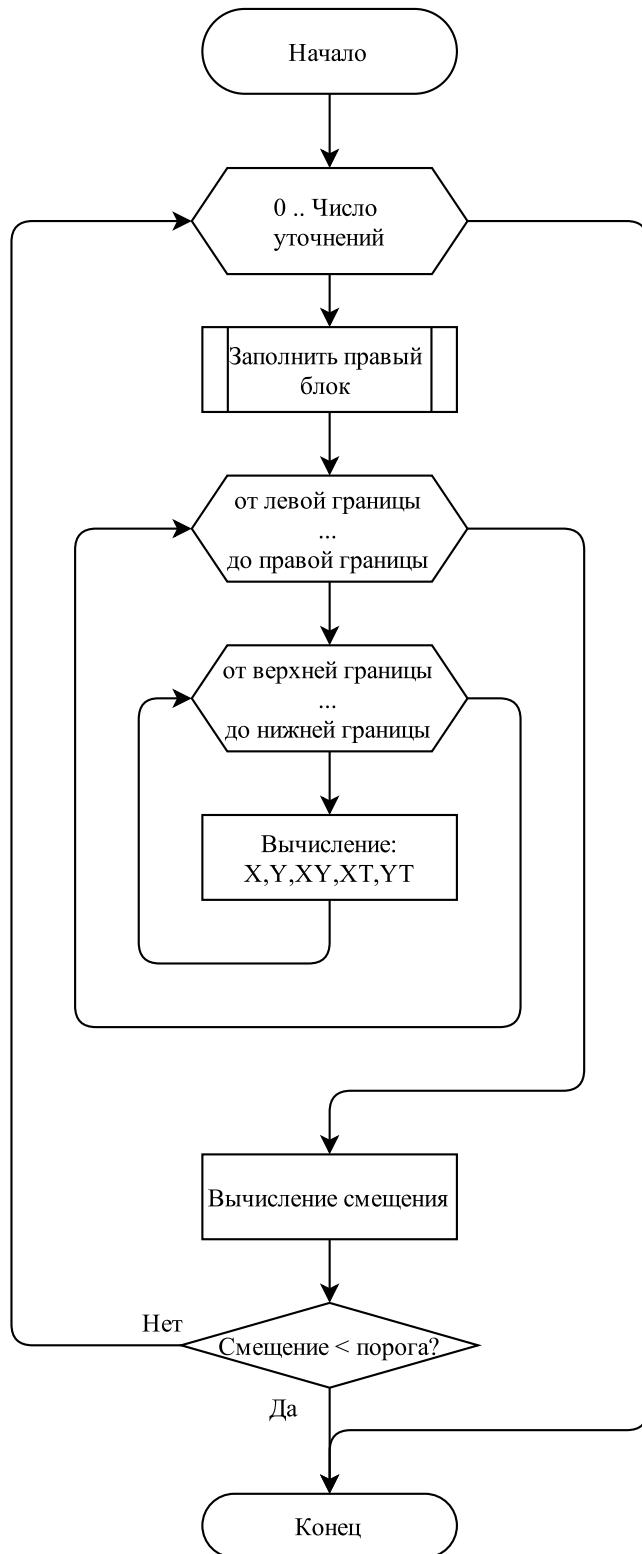
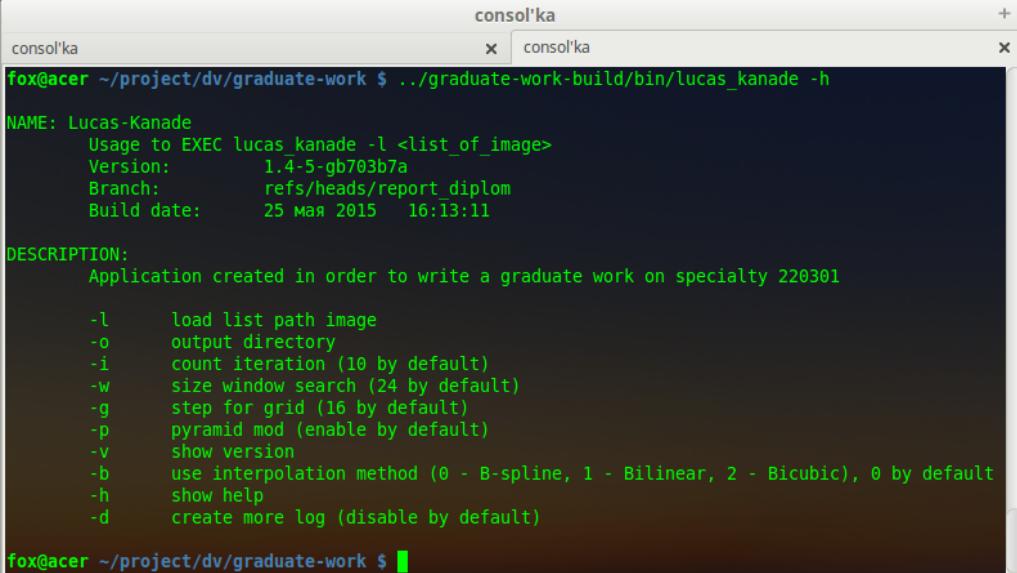


Рисунок 2.7 – Алгоритм вычисления оптического потока

- интерфейс командной строки позволяет писать скрипты для автоматизации запуска и тестирования с различными входными параметрами, что средствами графического интерфейса гораздо сложнее;
- большая функциональность;
- некая сложность при использовании неопытным пользователям;
- невозможность просмотра выходных результатов.



```

consol'ka
x consol'ka
fox@acer ~/project/dv/graduate-work $ ../../graduate-work-build/bin/lucas_kanade -h

NAME: Lucas-Kanade
      Usage to EXEC lucas_kanade -l <list_of_image>
      Version:      1.4-5-gb703b7a
      Branch:       refs/heads/report_diplom
      Build date:   25 мая 2015 16:13:11

DESCRIPTION:
      Application created in order to write a graduate work on specialty 220301

      -l      load list path image
      -o      output directory
      -i      count iteration (10 by default)
      -w      size window search (24 by default)
      -g      step for grid (16 by default)
      -p      pyramid mod (enable by default)
      -v      show version
      -b      use interpolation method (0 - B-spline, 1 - Bilinear, 2 - Bicubic), 0 by default
      -h      show help
      -d      create more log (disable by default)

fox@acer ~/project/dv/graduate-work $ █

```

Рисунок 2.8 – Пример консольного интерфейса в среде Linux

Перечень команд для запуска:

- l — список изображений для обработки;
- o — директория для выходных результатов;
- i — число уточнений при поиске смещённой части (по умолчанию равна 10);
- w — радиус окна поиска (по умолчанию равна 24);
- g — шаг между векторами оптического потока(по умолчанию равна 16);
- p — применить метод пирамиды(по умолчанию опция включена);
- v — показать версию программного обеспечения;
- b — использование метода интерполяции(0 — Б-сплайн, 1 — Билинейная, 2 — Бикубическая), по умолчанию 0;
- h — показать краткую справку;
- d — генерировать подробный лог файл(по умолчанию опция выключена).

2.7.2 Описание графического интерфейса программы

Второй графический — более удобный для неопытного пользователя. Общий вид ПО приведен на рисунке 2.9. Главное окно приложения включает в

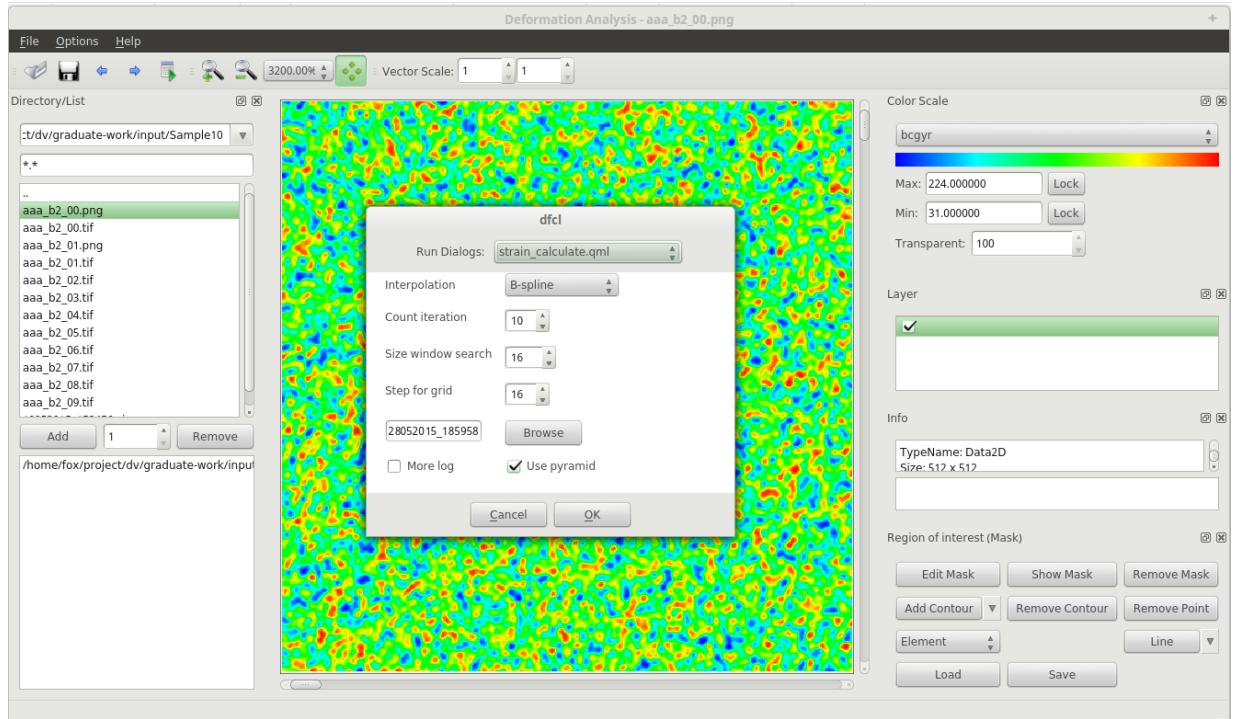


Рисунок 2.9 – Пример графического интерфейса

себя:

- окно просмотра документа;
- список файлов в директории;
- главное меню;
- панели инструментов;
- список файлов добавленных на обработку.

Форма запуска приложения, содержит большинство описанных ранее функций:

- метод интерполяции
- число уточнений при поиске смещённой части (задаётся в диапазоне от 10 до 100);
- радиус окна поиска (по умолчанию равна 24);
- шаг между векторами оптического потока(по умолчанию равна 16);
- директория для выходных результатов;
- режим логов;
- режим пирамиды.

3 Тестирование программного обеспечения

Тестирование на парах модельных изображений и на изображениях поверхностях реальных образцов.

3.1 Образцы изображений

3.1.1 Модельные изображения

В качестве образцов брались наборы изображений из интернет ресурса “Society for Experimental Mechanics (sem.org)” описание текстур находится в таблице 3.1, текстуры изображены на рисунке 3.3.

Таблица 3.1 – Описание модельных изображений

Серия	Диапазон яркостей	Уровень шума	Сдвиг (px)	Размеры
Модель многослойного изображения	0-188	Нет	1-20	510x510
Модель спекла	0-188	Нет	1-70	512x512
Модель высокий контраст	10-240	Низкий	0.1-1	487x325
Prosilica Bin	10-156	Низкий	0.1-1	487x325
Strain Gradient	20-185	Средний	0.01-1	512x512
Strain Gradient	30-225	Средний	0.01-1	512x512

3.1.2 Реальные отснятые изображения

Реально отснятые изображения предоставлены сотрудником ИФПМ СО РАН.

На первой серии (рисунок 3.1) изображён металлический образец из авиационного алюминиевого сплава Д16АТ нагружавшиеся на механической испытательной машине ИМАШ-2078 в условиях одноосного статического растяжения. Размеры предоставленных изображений 1920x1280 пикселей.

На второй серии (рисунок 3.2) изображён углерод-углеродный композиционный материал. Образец испытывали на одноосное статическое растяжение на электро-механической машине Instron 5582 со скоростью перемещения поглощаемого захвата 0,3 мм/мин.

Фотографирование поверхности осуществляли с помощью фотокамеры



Рисунок 3.1 – Раcтяжение пластины алюминия Д16АТ

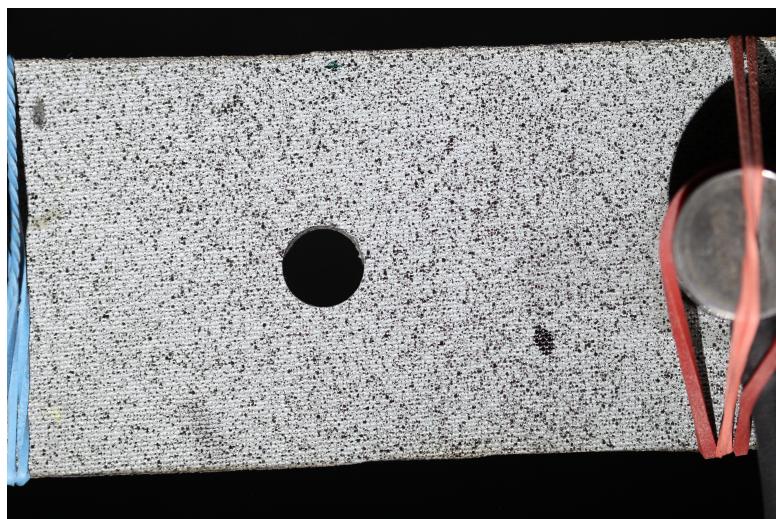


Рисунок 3.2 – Углерод-углеродный композиционный материал

Canon EOS 550D, оснащённой длиннофокусным объективом Canon EF-S 100-400mm 1/4-5.6 IS.

3.2 Тестирование программного обеспечения

3.2.1 Тестирование на модельных изображениях

Для первого раза используем изображения из серии Grey texture, представленные на рисунке 3.3 а). Они имеют сдвиг по оси x на один пиксель влево, по y сдвиг отсутствует.

Результатом работы ПО, как описано в первом разделе, является вектор-

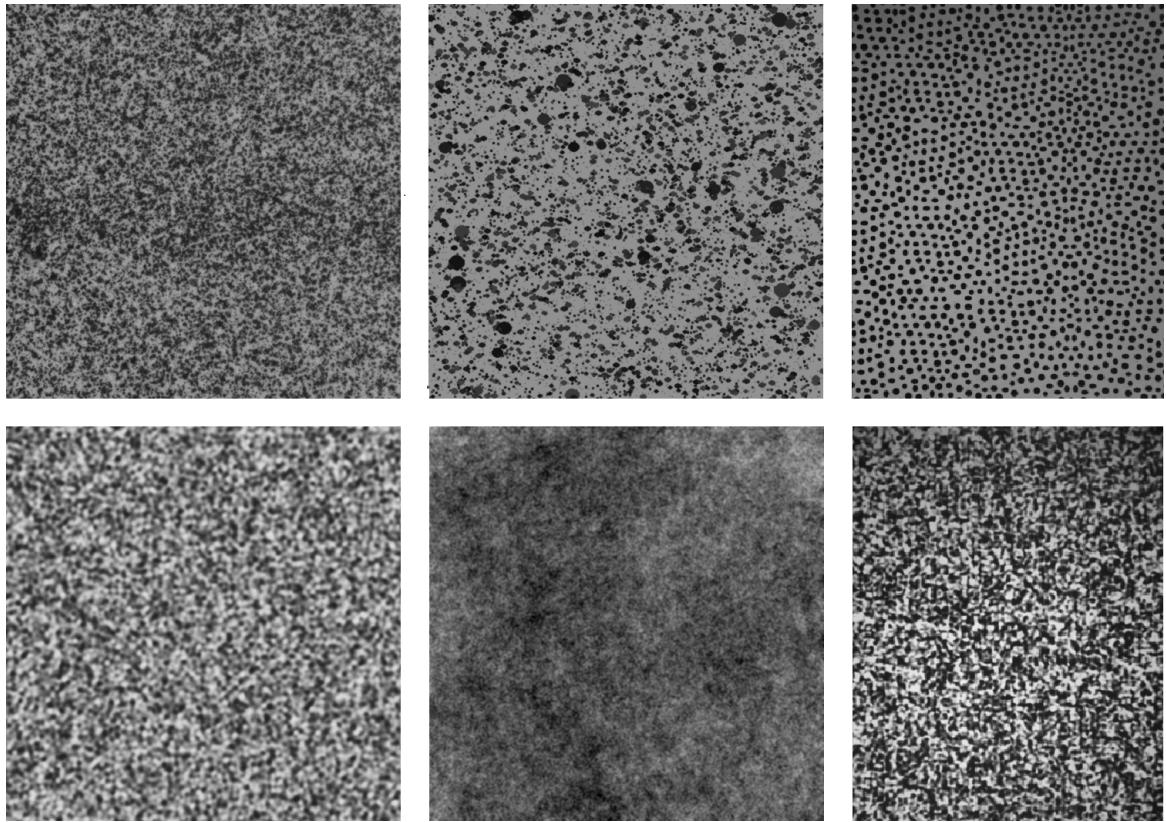


Рисунок 3.3 – Тестовая серия изображений: Grey set, High contrast, Sample6, Sample10, Sample11b, Grey set V2

ное поле и поля деформации твёрдого тела представленное на рисунке 3.4.

3.2.2 Тестирование на экспериментально полученных изображения

В данном разделе приведены результаты тестирования разрабатываемого программного обеспечения на изображениях поверхностей реальных образцов.

3.3 Исследование метода интерполяции

3.4 Исследование размера окна поиска

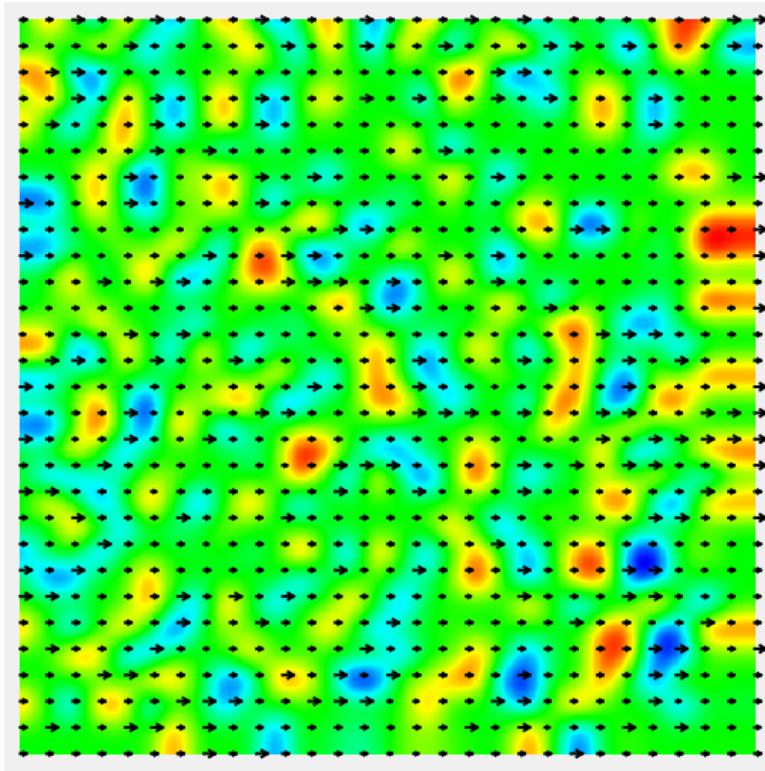


Рисунок 3.4 – Поле смещений и деформации твёрдого тела

4 Безопасность жизнедеятельности

4.1 Анализ опасных и вредных производственных факторов на рабочем месте

Производственный фактор, воздействие которого на работающего человека, в определённых условиях приводит к травме или другому внезапному резкому ухудшению здоровья, называется опасным. Если же производственный фактор приводит к заболеванию или снижению работоспособности, то его считают вредным (ГОСТ 12.0.002-80).

В зависимости от уровня и продолжительности воздействия вредный производственный фактор может стать опасным.

В целях предупреждения травматизма и профессиональных заболеваний при воздействии опасных и вредных производственных факторов (ОПФ и ВПФ) на предприятиях применяются меры по их предупреждению и устраниению, а также снижению степени воздействия на работников.

Согласно ГОСТ 12.0.003 – 74 “Опасные и вредные производственные факторы. Классификация”, все опасные и вредные производственные факторы делятся по природе воздействия на следующие группы:

- 1) физические;

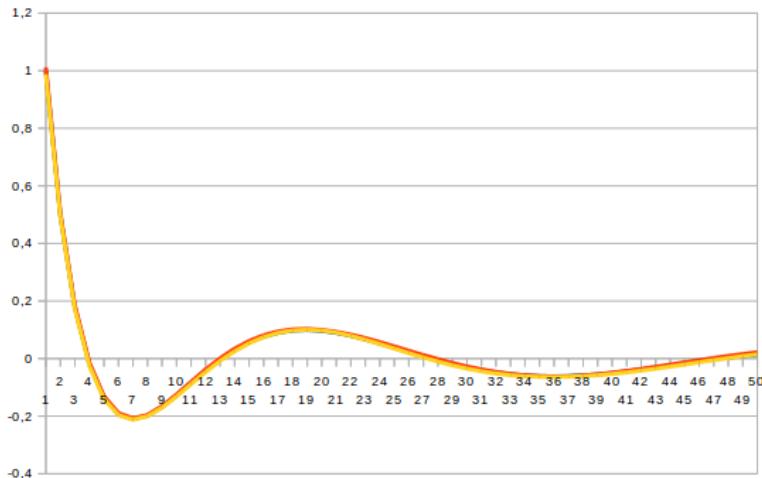


Рисунок 3.5 – Зависимость уровня ошибки, от числа итераций

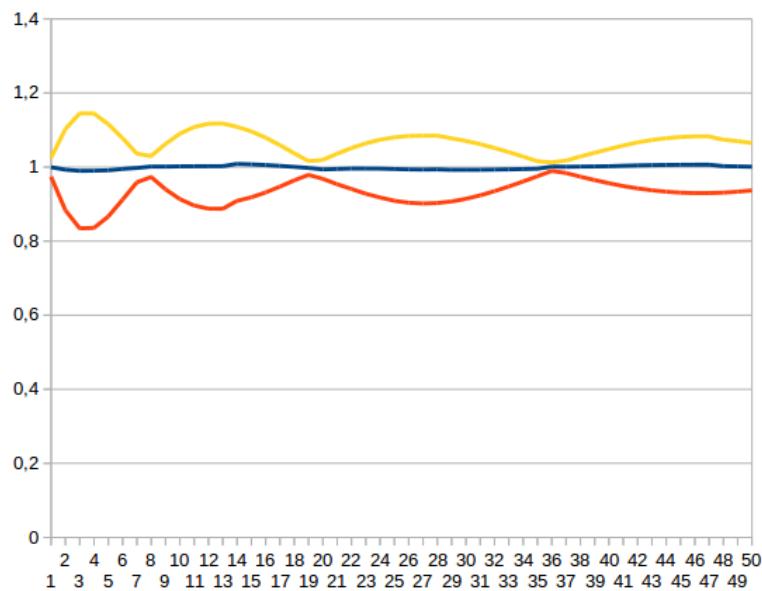


Рисунок 3.6 – Зависимость смещение по x, от числа итераций

- 2) химические;
- 3) биологические;
- 4) психофизиологические.

Из всех перечисленных факторов в наших условиях работы на организм действуют только физические и психофизиологические опасные и вредные производственные факторы.

К физическим опасным и вредным производственным факторам можно отнести:

- 1) недостаточная освещённость помещения. Качество поступающей информации во многом зависит от освещения. Недостаточное освещение не толь-

ко утомляет глаза, но и вызывает утомление организма в целом. При неудовлетворительном освещении снижается производительность труда и увеличивается брак;

2) повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека. Системный блок и монитор ЭВМ питаются от бытовой сети переменного тока напряжением 220В частотой 50 Гц;

3) повышенный уровень неионизирующих электромагнитных полей и излучений в рабочей зоне;

4) повышенный уровень шума. Одним из важнейших параметров, наносящим ущерб для здоровья и снижающим производительность труда, является шум. Шум может создаваться как работающим оборудованием, установками кондиционирования воздуха, работающими осветительными приборами дневного света, так и проникать извне. Действие шума различно: он затрудняет разборчивость речи, вызывает снижение работоспособности, повышает утомляемость, вызывает необратимые изменения в органах слуха человека. Шум воздействует не только на органы слуха, но и на весь организм человека через центральную нервную систему. Ослабляется внимание, ухудшается память, снижается реакция, увеличивается число ошибок при работе;

5) неудовлетворительное состояние параметров микроклимата. Количество теплоты, выделяемое человеком в окружающую среду, и охлаждающая способность среды должны быть адекватны, то есть человек должен себя чувствовать комфортно. В противном случае у человека возникают беспокоящие его температурные ощущения холода или перегрева, отрицательно сказывающиеся на его самочувствии и работоспособности;

6) возможность возникновения пожара. Во время работы с электрооборудованием в сети переменного тока может возникнуть перегрев и воспламенение самого оборудования или его обшивки.

К психофизиологическим факторам согласно ГОСТ 12.0.003-74 можно отнести такие факторы как:

- 1) физические (статические) перегрузки;
- 2) нервно-психические перегрузки.

В процессе работы инженер не подвержен воздействию химических и биологических производственных факторов.

4.2 Требования и защитные мероприятия в области БЖД

4.2.1 Освещение

Информация, которую человек получает из внешнего мира, поступает в основном через зрительный канал. Поэтому качество информации, получаемой посредством зрения, во многом зависит от освещения. Неудовлетворительное освещение может исказить информацию; кроме того, оно утомляет не только зрение, но вызывает утомление организма в целом. Неправильное освещение может также, являясь причиной травматизма: плохо освещенные опасные зоны, слепящие лампы и блики от них, резкие тени ухудшают или вызывают полную потерю ориентации работающих.

На практике пользуются двумя видами освещения — естественным и искусственным.

При освещении помещений, согласно СНиП 23-05-95 “Естественное и искусственное освещение”, необходимо соблюдать следующие требования:

1) естественное освещение должно осуществляться через светопроёмы, ориентированные преимущественно на север и северо-восток, и обеспечивать коэффициент естественной освещенности не ниже 1.2 %;

2) освещенность рабочего места пользователя ПЭВМ должно быть 200-400 лк (IV (в) разряд зрительных работ, “Естественное и искусственное освещение СНиП 23-05-95”);

3) для освещения зоны расположения документов допускается установка светильников местного освещения. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк;

4) в поле зрения оператора должно отсутствовать прямая и отражённая блескость;

5) на рабочем месте оператора должно быть ограничена пульсация освещённости от газоразрядных источников света;

6) в качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛБ. Допускается применение ламп накаливания в светильниках местного освещения.

Рассчитаем реальную освещенность на рабочем месте. Целью данного расчета является проверка соответствия освещенности в рабочем помещении норме

освещенности согласно СНиП 23-05-95.

Проведем проверочный расчет освещенности методом коэффициента использования светового потока.

Освещенность определяется по формуле:

$$E = \frac{F \cdot N \cdot \eta}{k \cdot S \cdot z} \quad (4.1)$$

где F – световой поток каждой из ламп, лм;

N – число источников света;

η – коэффициент использования светового потока;

k – коэффициент запаса, учитывающий запыление светильников и износ источников света;

S – площадь помещения, м²;

z – коэффициент неравномерности освещения.

Определим данные для расчета.

Коэффициент k для помещений, освещаемых лампами и при условии чистки светильников не реже двух раз в год берется равным 1,4 – 1,5.

При оптимальном расположении светильников коэффициент неравномерности z равен 1,1 – 1,2.

Коэффициент использования светового потока η зависит от типа светильника, коэффициента отражения светового потока от стен P_C , потолка P_P , пола $P_{ПОЛА}$, а также геометрических размеров помещения и высоты подвеса светильников, что учитывается одной комплексной характеристикой – индексом помещения:

$$I = \frac{S}{h \cdot (A + B)} \quad (4.2)$$

где h – высота подвеса светильников над рабочей поверхностью, м;

A – ширина помещения, м;

B – длина помещения, м.

Рассматриваемое помещение имеет следующие характеристики:

- ширина A – 5 м;
- длина B – 7 м;
- площадь помещения S – 35 м²;
- высота до осветительного прибора h – 3 м;

- количество ламп $N = 8$;
- поправочный коэффициент $Z = 1.1$;
- световой поток одной лампы $F = 1340$ лм;
- коэффициент запаса $k = 0.4$.

Тогда индекс помещения по формуле (4.2): Используя этот коэффициент по светотехнической таблице, находим, что для нашего помещения коэффициент η равен 53 %.

Теперь можно произвести расчет освещенности по формуле (4.1):

Согласно СНиП 23-05-95, освещенность рассматриваемого помещения находится в диапазоне оптимального освещения, т.к. по нормативам для разряда зрительной работы IV (в) норма освещенности находится в диапазоне от 200 лк до 400 лк. Это означает, что мощность и количество осветительных приборов для данного помещения выбраны правильно, и не требуется дополнительного освещения.

4.2.2 Микроклимат

Выбор типа *производственного помещения* определяется технологическим процессом, возможностью борьбы с шумом, вибрациями и загрязнением воздуха. Наличие больших оконных проемов и фонарей должно обеспечивать хорошую естественную освещенность. В помещении обязательно устройство вентиляции.

Объем и площадь производственного помещения, которые должны проходить на каждого работающего по санитарным нормам, должны быть не менее $20 m^3$ и $6 m^2$ соответственно. Высота производственных помещений не должна быть менее 4 м. Стены и потолки необходимо сооружать из малотеплопроводных материалов, не задерживающих осаждение пыли. Полы должны быть теплыми, эластичными, ровными и нескользкими.

Под оптимальными микроклиматическими условиями понимают такие сочетания параметров микроклимата, которые при детальном и систематическом воздействии на человека обеспечивают сохранение нормального функционального и теплового состояния организма.

На организм человека и работу вычислительной техники большое влия-

ние оказывает относительная влажность воздуха. При влажности воздуха до 40% становится хрупкой основа магнитной ленты, повышается износ магнитных головок, выходит из строя изоляция проводов, также возникает статическое электричество при движении носителей информации в ЭВМ. При относительной влажности воздуха более 75-80% снижается сопротивление изоляции, изменяются рабочие характеристики ЭВМ, возрастает интенсивность их отказов.

Скорость движения воздуха тоже оказывает влияние на функциональную деятельность человека, так как способствует испарению влаги с кожного покрова. А это, в свою очередь, приводит либо к высыханию кожи, либо к нарушению теплового равновесия организма, т.е. скорость движения воздуха, может иметь положительное значение с точки зрения физического охлаждения лишь до температуры воздуха $35 - 36^{\circ}\text{C}$. При дальнейшем повышении температуры окружающей среды единственным путем теплопередачи является испарение. Однако при повышении температуры выше 40°C движение даже относительно сухого воздуха может оказываться неблагоприятным фактором. Горячий воздух отдает теплоту телу, и подвижность воздуха в этом случае приводит не к охлаждению, а, наоборот, к нагреванию.

В машинном зале рекомендуется поддерживать температуру и влажность воздуха постоянными. Атмосферное давление должно быть в допустимых пределах, так как при пониженном, например, давлении ухудшается отвод тепла от элементов ЭВМ, снижаются изоляционные свойства узлов и устройств ЭВМ.

Воздух должен в значительной степени очищаться от пыли. ЭВМ, имеющие в своём составе устройства ввода-вывода на магнитных дисках, требуют этого, так как пылинки, попадающие на рабочую поверхность диска, могут привести к повреждению магнитной головки или поверхности диска. Пыль, оседающая на устройства и узлы ЭВМ, ухудшает теплоотдачу, может образовывать токопроводящие цепи, вызывает износ подвижных частей, нарушает контакты и приводит к засорению лёгких у работающего персонала.

Не менее важно и значение освещения. При неудовлетворительном освещении зрительная способность снижается, и могут появиться заболевания глаз. Правильно выполненная система освещения имеет большое значение в снижении травматизма, уменьшая потенциальную опасность многих производственных факторов; создаёт нормальные условия для работы органов зрения и повышает общую работоспособность организма.

С целью создания нормальных условий работы установлены нормы производственного микроклимата (ГОСТ 12.1.005-88). Эти нормы устанавливают оптимальные и допустимые значения температуры, относительной влажности и скорости движения воздуха для рабочей зоны помещений, оборудованных компьютерами с учётом тяжести выполняемой работы и сезона года.

Период года	Температура, С				Относительная влажность, проценты	Скорость движения воздуха, м\с			
	Оптимальная	Допустимая							
		Верхняя	Нижняя						
Пост.	Не пост.	Пост.	Не пост.	Пост.	Не пост.	Оптимальная	Допустимая		
Холодный	22-24	25	26	21	18	40-60	75	0,1	0,1
Теплый	23-25	28	30	22	20	40-60	70	0,1	0,1

Таблица 4.2 – Оптимальные и допустимые нормы температуры, относительной влажности и скорости движения воздуха в рабочей зоне производственного помещения

4.2.3 Электробезопасность

Электрические установки, к которым относится ЭВМ и стенд, представляют для человека потенциальную опасность. В процессе эксплуатации или при проведении профилактических работ человек может соприкасаться с частями, находящимися под током. Согласно классификации помещений по электробезопасности (ПУЭ – 7-е издание) рабочий кабинет относится к помещениям без повышенной опасности, характеризующимся наличием следующих условий:

- 1) напряжение питающей сети, 220 В;
- 2) напряжение питающей сети, 380 В;
- 3) относительная влажность воздуха, не более 75%;

4) средняя температура не более $25^{\circ}C$;

При нормальном режиме работы опасность поражения электрическим током невелика, однако, возможны режимы, называемые аварийными, когда происходит случайное электрическое соединение частей оборудования, находящихся под напряжением с заземлением конструкциями. Опасность поражения электрическим током существует всюду, где используются электроустановки, поэтому помещения без повышенной опасности нельзя назвать безопасными.

Любое из действий тока (термическое, электролитическое, механическое и биологическое) может привести к электрической травме, т. е. к повреждению организма, вызванному действием электрического тока или электрической дуги.

Основными техническими способами и средствами защиты от поражения током, согласно ГОСТ 12.1.019-79 “Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты”, являются: защитное зануление, выравнивание потенциалов, защитное заземление, электрическое разделение сети, изоляция токоведущих частей, ограждительные устройства и другое.

Согласно “Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов” ЭВМ и стенд, на которых производится работа, относятся к классу электробезопасности 01 (имеет рабочую изоляцию, элемент для заземления и провод без заземляющей шины для подключения питания).

Напряжения прикосновения и токи, протекающие через тело человека при нормальном (неаварийном) режиме электроустановки не должны превышать значений, указанных в таблице 4.3

Для исключения возможности поражения электрическим током проведены следующие меры:

- 1) токоведущие части, находящиеся под напряжением, изолированы;
- 2) корпуса приборов заземлены;
- 3) заземляющие проводники видны, а места их соединений скреплены резьбовыми соединениями.

Род тока	Нормируемая величина	Предельно допустимые значения, не более, при продолжительности воздействия тока t , с											
		0,01-0,08	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0	Свыше 1,0
Переменный 50 Гц	U, В	550	340	160	135	120	105	95	85	75	70	60	20
	I, мА	650	400	190	160	140	125	105	90	75	65	50	6
Переменный 400 Гц	U, В	650	500	500	330	250	200	170	140	130	110	100	36
	I, мА	650	500	500	330	250	200	170	140	130	200	100	8
Постоянный	U, В	650	500	400	350	300	250	240	230	220	210	200	40
	I, мА	650	500	400	350	300	250	240	230	220	210	200	15
Выпрямленный двух полупериодный	Um, В	650	500	400	300	270	230	220	210	200	190	180	—
	Im, мА	650	500	400	300	270	230	220	210	200	190	180	—
Выпрямленный одно полупериодный	Um, В	650	500	400	300	250	200	190	180	170	160	150	—
	Im, мА	650	500	400	300	250	200	190	180	170	160	150	—

Таблица 4.3 – Таблица предельно допустимых значений напряжений прикосновения и токов

4.2.4 Пожарная безопасность

Для выполнения требований по пожарной безопасности (ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность) необходимо удаление оборудования друг от друга на расстояние не менее 60-80 см.

Опорные части из изоляционных материалов, удерживающие детали, предназначенные для подключения к сети, а также изолирующие корпуса должны быть изготовлены из материалов, не представляющих опасность при возникновении коротких замыканий внутри блока или при нагреве, вызываемого плотным контактом внешних проводов.

Необходимо также наличие пожарного инвентаря и проведение инструктажей.

Приведем возможные причины возникновения пожаров:

- 1) наличие твердых горючих веществ;
- 2) опасная перегрузка сетей, которая ведет за собой сильный разогрев токопроводящих проводников и загорания изоляции;
- 3) короткие замыкания;
- 4) пуск оборудования после ремонта.

Для предупреждения пожаров от коротких замыканий, перегрузок необходим правильный выбор монтаж и соблюдение установленного режима эксплуатации

электрических сетей, дисплеев и других устройств.

4.2.5 Электромагнитные неионизирующие излучения

Длительное воздействие электромагнитных полей промышленной частоты (50 Гц) приводит к расстройствам в головном мозге и центральной нервной системе. В электрическом поле (ЭП) атомы и молекулы поляризуются. Полярные молекулы ориентируются по направлению распространения электромагнитного поля, что изменяет ориентацию клеток или цепей молекул, ослабляя биохимическую активность белковых молекул. В результате у человека наблюдаются головная боль в височной и затылочной областях, вялость, ухудшение памяти, боли в области сердца, угнетенное настроение, апатия, своеобразная депрессия с повышенной чувствительностью к яркому свету и интенсивному звуку, расстройство сна, сердечно-сосудистой системы (ССС), органов пищеварения, дыхания, повышенная раздражительность. Могут наблюдаться функциональные нарушения в ЦНС, а также изменения в составе крови.

Воздействие постоянного магнитного поля (ПМП) и с частотой 50 Гц на человека проявляется в индуцировании в теле человека вихревых токов.

При длительном систематическом воздействии могут возникнуть изменения функционального состояния нервной системы, иммунной системы и сердечно-сосудистой системы. Длительное воздействие ЭМП промышленной частоты может спровоцировать онкологические заболевания.

Предельно допустимые значения напряженности электрического и магнитного полей промышленной частоты в зависимости от времени их воздействия устанавливаются СанПиН 2.2.4.1191-03 “Электромагнитные поля в производственных условиях”. Согласно этому нормативному документу пребывание в ЭП промышленной частоты напряженностью до 5 кВ/м допускается в течение всего рабочего дня.

При работе на персональном компьютере расстояние от монитора до глаз пользователя должно быть не менее 50 см. Исследования показали, что с уменьшением расстояния на каждые 10 см уровень электромагнитного излучения возрастает в среднем в 1,5 раза, а с увеличением расстояния с 50 до 60 см уменьшение уровня электромагнитного поля идет в той же зависимости.

Таблица 7. - Допустимые значения параметров неионизирующих электромагнитных излучений(в соответствии с СанПиН 2.2.2.542-96)

Таблица 4.4 – Допустимые значения параметров неионизирующих электромагнитных излучений

Наименование параметра	Допустимые
Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	10В/м
Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	0,3А/м
Напряженность электростатического поля не должна превышать: для взрослых пользователей	20кВ/м
для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений	15кВ/м

4.2.6 Допустимые уровни звукового давления и звукового шума

Шум на исследовательском рабочем месте создаётся вентиляционной системой ПЭВМ и печатающим устройством. В процессе рабочего дня принтер включается по мере необходимости, поэтому шум следует квалифицировать как непостоянный, прерывистый.

Основными физическими величинами, характеризующими шум, являются интенсивность, звуковое давление и частота. Согласно ГОСТ 12.1.003-83 “Система стандартов безопасности труда. Шум. Общие требования безопасности” для оценки шума используют частотный спектр измеряемого уровня звукового давления, выраженного в дБ, в октавных полосах частот, который сравнивают с предельным спектром, приведены в таблице 4.5.

Таблица 4.5 – Значения предельно допустимых уровней шума на рабочих местах производительных предприятий

Рабочие места	Уровни звукового давления, дБ,,в октавной полосе частот, Гц								Уровни звука, дБ
	63	125	250	500	1000	2000	4000	8000	
(1)	71	61	54	49	45	45	40	38	50
(2)	79	70	63	58	55	55	50	49	60
(3)	83	74	68	63	60	60	55	54	65
(4)	94	87	82	78	75	75	71	70	80
(5)	99	92	86	83	80	78	76	74	85

(1) – помещение технологических бюро, лаборатории для теоретических работ;

(2) – помещения управления, рабочие комнаты;

4.3 Требования эргономики и технической эстетики

Для создания благоприятных условий труда в лаборатории необходимо учитывать психофизические особенности человека, а также общую гигиеническую обстановку. Большое значение в создании оптимальных условий труда имеют складывающиеся в трудовом коллективе взаимоотношения между студентами, которые принято называть социальным климатом. Человек, находящийся в состоянии нервного возбуждения допускает много ошибок при работе с технологической документацией.

С эстетической точки зрения помещение и оборудование должны быть окрашены в спокойные тона (синие, голубые, зелёные) успокаивающие и уменьшающие зрительное утомление. Окраска стен и дверей помещения должна иметь мягкие переходы без резких яркостных контрастов. В рабочем помещении стены покрашены бежевой краской, потолок покрыт белой плиткой, двери окрашены в синий цвет. Это даёт хорошее отражение и рассеяние света.

Важную роль играет планирование рабочего места, которое должно удовлетворять требованиям удобства выполнения работ и экономии энергии, времени инженера, удобства обслуживания устройств ЭВМ и соблюдения правил охраны труда. Рабочее место должно обеспечивать удобство выполнения работы в помещении сидя, стоя и соответствовать требованиям ГОСТ 12.2.032-78 и ГОСТ 12.2.033-78. Необходим учёт эргономических свойств человека, подбор вспомогательных предметов оборудования (столы, и т. п.), удобных для использования на рабочем месте. Для этого требуется рациональная расстановка оборудования, оптимальная организация рабочего места (правильный выбор основного технологического оборудования, удобство выполнения работ). Рабочее место при выполнении действий в положении сидя должно соответствовать нормам ГОСТ 12.2.032-78.

Определим требования к рабочему месту: обеспечение возможности удобного выполнения работ, учёт физической тяжести труда, учёт размеров рабочей зоны, учёт технологических особенностей процесса выполнения работ. Параметры рабочего места инженера приведены в таблице 4.6.

Таблица 4.6 – Параметры рабочего места инженера

Параметры	Рекомендуемые, мм	Фактические, мм
-----------	-------------------	-----------------

Высота сидения	450	440
Высота рабочей поверхности	720	700
Ширина сидения	500	490
Высота спинки сидения	800	740
Высота пространства для ног	600	700
Размеры рабочей поверхности	1600x900	1600x1000
Высота ПК	620x800	740
Расстояние от глаз инженера до предмета	600	620
Расстояние от экрана или предмета до края стола	750	650

В рабочей зоне необходимо исключение резких и подвижных теней, отблесков. При планировке рабочего места необходимо учитывать зоны досягаемости рук оператора при расположении дисплея, клавиатуры, органов управления системы и периферийных устройств. Эти зоны, установленные на основании антропометрических данных человеческого тела, дают возможность рационально разместить как по горизонтали, так и по вертикали все элементы рабочего места.

Правильная организация рабочего места оператора ЭВМ предусматривает также соблюдение следующих параметров:

- 1) высота мыши с клавиатурой (62-88) см (над уровнем пола);
- 2) высота экрана (над уровнем пола) (90-128) см;
- 3) расстояние от экрана до края стола (0-115) см;
- 4) наклон экрана от минус 15⁰ до плюс 20⁰ к нормальному его положению;
- 5) расстояние от глаз оператора до экрана должно быть в пределах от 40 до 80 см.

В состав рабочего места входят учебный стенд, персональный компьютер, видеомонитор, клавиатура.

Органы управления, к которым относятся клавиатура и манипулятор “мышь” ЭВМ расположены в зоне досягаемости, ограниченной длиной руки, т.е. от 70 до 80 см. Такое расположение обеспечивает равномерную нагрузку обеих рук оператора.

К системам отображения информации, на данном рабочем месте, относятся: видеомонитор ЭВМ. Видеомонитор расположён в зоне пространства отображения информации ($\pm 15^\circ C$ от нормальной линии взгляда), что обеспечивает оптимальный зрительный поиск.

В результате анализа можно сделать вывод, что организация рабочего места, на котором выполнялась дипломная работа, удовлетворяет перечисленным выше требованиям правильной организации рабочего места оператора ЭВМ. Так, как и остальные условия работы в помещении являются удовлетворительными (микроклимат, освещение и т.д.), о чем писалось выше, то согласно ГОСТ 12.2.032-78.ССБТ данное рабочее место работника можно считать соответствующим общим эргономическим требованиям.

4.4 Общие требования безопасности перед началом, вовремя, по окончании работы и в случае аварийных ситуаций

4.4.1 Требования безопасности перед началом работы

Перед началом работы работник должен:

- проверить на рабочем месте наличие и пригодность средств защиты, инструмента и приспособлений, а также наличие электрического фонаря, средств пожаротушения, плакатов или знаков безопасности;
- проверить достаточность освещения рабочей зоны и на обслуживаемом оборудовании (отсутствие перегоревших ламп) наличие плафонов на светильниках;
- ознакомиться с работами производимыми предыдущими работниками, прочитав об этом в “Оперативном журнале”;
- при обнаружении, каких-либо отклонений в работе САР доложить технику.

4.4.2 Требования безопасности во время работы

- использовать инструмент и приспособления, предназначенные только для выполнения данной работы. Не допускать применения случайных приспособлений и предметов вместо инструмента;
- следует выполнять только ту работу, которая поручена;
- импульсные трубопроводы, на которых производится ремонт оборудо-

вания КИПиА, необходимо перекрыть при помощи запорной арматуры и снять давление;

Запрещается во время работы:

- – – эксплуатировать неисправное оборудование, а также оборудование с неисправными или отключенными устройствами аварийного отключения блокировок, защит и сигнализации;
- применять для отмывки и обезжикивания деталей и оборудования керосин, бензин, бензол, ацетон и другие, горючие и легковоспламеняющиеся вещества, при уборке помещений и оборудования горючие вещества, а также хлорпроизводные углеводороды;
- при обнаружении дефектов на оборудовании немедленно сообщить об этом своему руководству и руководству обслуживаемого объекта;
- во время работы необходимо поддерживать чистоту и порядок на рабочем месте;
- во время работы соблюдать противопожарные правила, знать местонахождение первичного противопожарного инвентаря, уметь его применять и не допускать использование противопожарных средств не по назначению;

4.4.3 Действия в аварийных ситуациях

При возникновении аварии или ситуации, которые могут привести к нежелательным последствиям, необходимо прекратить работу, принять меры по предупреждению несчастных случаев, выхода из строя оборудования, сообщить технику или заведующему кафедры, в котором возникла аварийная ситуация, а также принять меры обеспечения безопасности других лиц;

При возникновении аварийных ситуаций необходимо:

- прекратить работу;
- обесточить оборудование, если это угрожает жизни и здоровью персонала;
- принять меры по предупреждению несчастных случаев, выхода из строя оборудования;
- сообщить руководству;
- при возникновении пожара необходимо немедленно вызвать пожарную охрану, отключить электрооборудование, находящееся в зоне пожара. Приступить к тушению пожара с помощью первичных средств пожаротушения. Для

тушения пожара в электроустановках необходимо применять только углекислотные и порошковые огнетушители.

4.4.4 Требования безопасности по окончании работы

По окончании работы работник должен:

- выключить оборудование;
- привести в порядок рабочее место.

5 Технико-экономическое обоснование

5.1 Обоснование необходимости проводимого исследования

Настоящая дипломная работа исследует алгоритмы поиска смещений по парам изображений основываясь на дифференциальном алгоритме Лукаса-Канаде. На основе полученных векторных полей смещений можно строить карты поверхностной деформации твердого тела.

Целью дипломной работы является разработка программного обеспечения (ПО) для оценки деформаций поверхностей твёрдых тел, а также проведение исследований алгоритмов и методов как на модельных, так и на реальных оптических изображениях.

5.2 Планирование комплекса работ по разработке программного обеспечения

Основными задачами планирования работ являются:

- определение объёма предстоящих работ;
- распределение объёма работ на взаимосвязанные последовательные этапы;
- установление сроков выполнения работ;
- определение необходимых, для выполнения планируемых работ денежных, материальных и трудовых ресурсов.

При выполнении дипломной работы было задействовано два человека:

- руководитель (рук.);
- разработчик (разр.).

Руководитель выполняет контроль выполнения различных этапов работ, согласованность этапов выполнения работ между собой, корректирует действия разработчика, дает рекомендации по выполнению тех или иных работ. Разработчик реализует тот объем работ, который установлен руководителем в соответствие с техническим заданием.

Месячный оклад студента в ТУСУР равен 10483 рублей, с учетом 20 рабочих дней в месяце, и 8 часового рабочего дня, стоимость одного часа работ равна 65,51 рублей. Месячный оклад руководителя к.т.н., доцента в университете равен 14800 рублей, с учетом 24 рабочих дней, и 6 часового рабочего дня, стоимость одного часа работ равна 102,8 рубля.

График выполнения работ приведен в таблице 5.7.

Наименование этапов и содержание работ	Исполнитель (должность)	Трудоемкость		Процент от общей трудоемкости, %	Количество исполнителей, чел.	Стоимость одного часа работ, руб/ч	Общая стоимость работы, руб.	Продолжительность рабочего дня, час.	Срок исполнения, дни
		Нормо-часы, н-ч							
1 Постановка задачи	рук.	2	4%	1	102,8	205,6	6	1	
	разр.	10	2%	1	65,51	655,1	8	2	
2 Анализ научно-технической литературы	разр.	25	5%	1	65,51	1637,75	8	4	
3 Разработка и реализация алгоритма Лукаса-Канаде	разр.	120	26%	1	65,51	7861,2	8	15	
4 Разработка итеративного метода и суб-пиксельного смещения	рук.	15	30%	1	102,8	1542	6	3	
	разр.	70	15%	1	65,51	4585,7	8	9	
5 Разработка иерархического метода	рук.	25	50%	1	102,8	2570	6	5	
	разр.	70	15%	1	65,51	4585,7	8	9	
6 Проведение эксперимента на искусственных и реальных изображениях	рук.	3	6%	1	102,8	308,4	6	1	
	разр.	34	7%	1	65,51	2227,34	8	5	
7 Анализ результатов	рук.	10	20%	1	102,8	1028	6	2	
	разр.	40	9%	1	65,51	2620,4	8	5	
8 Создание основной части пояснительной записи	рук.	10	20%	1	102,8	1028	6	2	
	разр.	40	9%	1	65,51	2620,4	8	5	
9 Создание документации по охране труда и экономической части	разр.	40	9%	1	65,51	2620,4	8	5	
10 Подготовка графического материала	разр.	16	3%	1	65,51	1048,16	8	2	
11 Проверка пояснительной записи	рук.	10	20%	1	102,8	1028	6	2	
Всего: 11	разр.	50	100	1	65,51	30462,15	6	6	
	рук.	465	100	1	102,8	5140	8	59	

Таблица 5.7 – График выполнения работ

Зная длительность цикла каждого этапа и возможность их параллельно-последовательного выполнения, можно рассчитать срок завершения планируемых работ и составить ленточный и сетевой графики плана их выполнения. Поскольку работа не требует большого состава исполнителей, то ограничимся ленточным графиком планирования, представленным в виде таблицы 5.8.

Этапы работы	Исполнитель	Длительность, дн.	Продолжительность работ, недели													
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	рук.	1														
	разр.	2														
2	разр.	4														
3	разр.	15														
4	рук.	3														
	разр.	9														
5	рук.	5														
	разр.	9														
6	рук.	1														
	разр.	5														
7	рук.	2														
	разр.	5														
8	рук.	2														
	разр.	5														
9	разр.	5														
10	разр.	2														
11	рук.	2														
	разр.	1														

Таблица 5.8 – Ленточный график загрузки участников работ

5.3 Определение сметной стоимости проекта

5.3.1 Общие положения

Смета затрат для данной работы состоит из расходов, которые включают в себя следующие статьи:

- затраты на оборудование и амортизацию;
- расходы на оплату труда и отчисления на социальные нужды;
- затраты на основные и вспомогательные материалы;
- затраты на электроэнергию.

5.3.2 Затраты на оборудование и амортизацию

Основным оборудованием при проведении работы являются компьютер и принтер, которые постановлением Правительства Российской Федерации от 1.01.02 г. N 1 отнесены ко второй амортизационной группе – «имущество со

Этапы работы		Исполнитель	Длительность, дн.	Календарные даты
1 Постановка задачи	рук.	1	2.03.15 - 4.03.15	
	разр.	2		
2 Анализ научно-технической литературы	разр.	4	5.03.15 - 9.03.15	
	разр.	15		
3 Разработка и реализация алгоритма Лукаса-Канаде	рук.	3	25.03.15 - 5.04.15	
	разр.	9		
4 Разработка итеративного метода и суб-пиксельного смещения	рук.	5	6.04.15 - 17.04.15	
	разр.	9		
5 Разработка иерархического метода	рук.	1	17.04.15 - 24.04.15	
	разр.	5		
6 Проведение эксперимента на искусственных и реальных изображениях	рук.	2	25.04.15 - 1.05.15	
	разр.	5		
7 Анализ результатов	рук.	2	2.05.15 - 13.05.15	
	разр.	5		
8 Создание основной части пояснительной записки	рук.	2	13.05.15 - 19.05.15	
	разр.	5		
9 Создание документации по охране труда и экономической части	разр.	2	20.05.15 - 22.05.15	
	разр.	2		
10 Подготовка графического материала	рук.	2	25.05.15 - 29.05.15	
	разр.	1		

Таблица 5.9 – Календарный график загрузки участников

сроком полезного использования свыше 2 лет до 3 лет включительно». Месячная норма амортизации составляет 2,8% и для компьютера, и для принтера.

Результаты расчётов амортизационных отчислений приведены в таблице 5.10.

Наименование прибора, оборудования	Потребленное количество, шт.	Цена, руб.		Время использования по теме	Месячная норма амортизации, %	Сумма амортизации, руб.
		Единицы	Всего			
Ноутбук	1	23000	23000	3 мес.	2,8	1932
Принтер	1	7000	7000	3 мес.	2,8	588
Итого: 2520 руб.						

Таблица 5.10 – Смета затрат на оборудование

5.3.3 Расходы на оплату труда и отчисления на социальные нужды

Статья затрат учитывает выплаты по заработной плате за выполненную работу, исчисленные на основании тарифных ставок и должностных окладов в соответствии с принятой в организации-разработчике системой оплаты труда. В этой статье также отражаются премии, надбавки и доплаты за условия труда, оплата ежегодных отпусков, выплата районного коэффициента и некоторые другие расходы. Отчисления на социальные нужды учитывают страховые взносы.

Результаты расчета расходов на оплату труда участников проекта представлены в таблице 5.11.

Участники проекта	$ЗП_{пр}$	Премия	$RН, руб. 30\%$	ОЗП	$ДЗП 15\%$	ФОТ	$Страховые взносы, руб. 30\%$	Всего
рук.	30462,15	-	9138,645	39600,8	5940,1	45540,92	13662,3	59203,2
разр.	3952,5	-	1185,75	5138,25	770,7375	59095	1772,7	8452,43
Итого: 67655,7 руб.								

Таблица 5.11 – Расчет расходов на оплату труда участников проекта

5.3.4 Затраты на основные и вспомогательные материалы

Статья включает расходы по приобретению и доставке основных и вспомогательных материалов, необходимых для опытно-экспериментальной проработки решения, для изготовления макета или опытного оборудования. Сюда включаются и стоимость необходимых материалов для изготовления образцов и макетов, и материалов необходимых для оформления требуемой документации.

Размер транспортно-заготовительных расходов (ТЗР), определяемый в процентах от стоимости, примем 10%. Стоимость вспомогательных материалов принимается 10% от стоимости основных материалов с учетом ТЗР. Результаты расчёта стоимости материалов представлены в 5.12.

Наименование материала	Единицы измерения	Потребленное количество	Цена за единицу, руб.	Сумма, руб.
Ватман	шт.	5	50	250
Пачка бумаги	шт.	1	250	250
Картридж для принтера	шт.	1	4450	4450
CD-диск	шт.	1	30	30
Итого затраты на основные (с учетом ТЗР) и вспомогательные материалы: 5478				

Таблица 5.12 – Расчёт затрат на основные и вспомогательные материалы

5.3.5 Расходы на электроэнергию

Статья включает затраты по электроэнергии на технологические нужды. В настоящее время тариф на электроэнергию для населения г. Томска на 2015 год составляет 2,7 руб./ кВт ч. Введенный приказом от 26.12.2014 " №6/9 (691) «О тарифах на электрическую энергию для населения и потребителей, приравненных к категории население по Томской области на 2015 год», принятый департаментом тарифного регулирования Томской области.

Результаты расчётов приведены в 5.13.

Наименование прибора или оборудования	Количество, шт.	Потребляемая мощность, кВт	Часы работы	Тариф за 1 кВт·час, руб.	Стоимость электроэнергии, руб.
Ноутбук	1	0,05	560	2,7	75,6
Принтер	1	0,1	4	2,7	1,08
Освещение	1	0,6	560	2,7	907,2
Всего: 983,88 руб.					

Таблица 5.13 – Затраты на электроэнергию

5.3.6 Накладные расходы

Расчет накладных расходов сведем в 5.14.

5.3.7 Сводная смета затрат

На основании всех произведенных расчетов составим сводную смету затрат на выполнение работы в виде таблицы 5.15.

Услуга	Количество	Стоимость одной единицы, руб.	Сумма затрат, руб.
Изготовление плакатов	5 штук	100	500
Переплет	1 штука	50	50
Транспортные расходы	10 поездок	17	170
Итого: 720			

Таблица 5.14 – Накладные расходы

Наименование статей затрат	Всего, руб.
1. ФОТ со страховыми взносами	67655,7
2. Основные и вспомогательные материалы	5478
3. Амортизационные отчисления	2520
4. Затраты на электроэнергию	983,88
5. Накладные расходы	720
Итого себестоимость работ: 77357,58руб.	

Таблица 5.15 – Сводная смета затрат

5.4 Научно-технический эффект

Количественная оценка научно-технического уровня может быть произведена путем расчета результативности участников разработки по формуле:

$$K_{ny} = \sum_{i=1}^n (K_{dyi} \cdot d_i),$$

где K_{ny} – коэффициент научного или научно-технического уровня;
 n – количество факторов.

Весовые коэффициенты d для каждого из факторов устанавливались эксперты путем. При этом сумма коэффициентов значимости по всем факторам равна единице. Коэффициенты достигнутого уровня факторов также установлены эксперты путем.

Показатели	Значимость показателя	Достигнутый уровень	Значение i -го фактора
	d_i	K_{dyi}	$K_{dyi} \cdot d_i$
Новизна полученных или предполагаемых результатов	0,375	0,7	0,2625
Перспективность использования результатов	0,300	0,8	0,24
Завершенность полученных результатов	0,125	1	0,125
Масштаб возможной реализации полученных результатов	0,200	0,4	0,08
Результативность	$K_{ny} = \sum_{i=1}^4 (K_{dyi} \cdot d_i) = 0,7075$		

Таблица 5.16 – Оценка научно-технического уровня разработки

Рассчитанный коэффициент научно-технической результативности равен

0,7075. Полученное значение достаточно высоко, что говорит об эффективности проведенных работ выше среднего, однако отмечается необходимость дальнейшего развития проекта для достижения завершенности полученных результатов.

5.5 1.5 Социальный эффект

Гипотетический социальный эффект заключается в улучшении результатов исследований, проводимых в лаборатории механики полимерных композиционных материалов СО РАН, которые могут использовать программное обеспечение оценки деформации твердых тел, и усилении научного потенциала.

6 Заключение

В результате выполнения работы было разработано программное обеспечение для оценки деформаций поверхности твёрдого тела. ПО позволяет оценивать оптический поток по серии входных снимков и проводить исследования механизмов пластической деформации на мезоуровне.

При разработке ПО были рассмотрены и реализованы методы вычисления оптического потока на основе алгоритма Лукаса-Канаде, его пирамидалная версия, и итеративная модификация с субпиксельной точностью.

Тестирование ПО показало, что классический алгоритм Лукаса-Канаде локален и не способен определить смещение большее чем область поиска. Пирамидалная версия алгоритма решает эту проблему.

Список использованных источников

- 1 Кориков А. М. Сырямкин В. И., [У+FFFД] Корреляционные зрительные системы роботов / [У+FFFД] Кориков А. М., Сырямкин В. И.; Ed. by [У+FFFД] отделение. — Томск: Радио и связь, - Томское отделение, 1990.
- 2 Lucas, B. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA) / B. Lucas, T. Kanade // Proceedings of the 1981 DARPA Image Understanding Workshop. — 1981. — P. 121–130.
- 3 Конушин, [У+FFFД] Сложение за точечными особенностями сцены (point feature tracking) / [У+FFFД] Конушин // Компьютерная графика и мультимедиа. — 2003. — № 1.
- 4 Bouguet, J.-Y. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. — 2000. <http://robots.stanford.edu/cs223b04/algotracking.pdf>.
- 5 Tomasi, C. Good features to track / C. Tomasi, J. Shi // Proc. IEEE Conf. on Comp. Vision and Patt. Recog. — 1994. — P. 593–600.
- 6 team1, D. I. Debian gnu/linux installation guide / D. I. team1. — Debian Installer team2, 2015.
- 7 Group, T. H. Hdf5 user's guide / T. H. Group. — 2011.
- 8 Scott Chacon,. Pro Git : professional version control / Scott Chacon. — 2011. <http://progit.org/ebook/progit.pdf>.
- 9 Г.В. Смирнов, [У+FFFД] [У+FFFД] Безопасность жизнедеятельности / [У+FFFД] [У+FFFД] Г.В. Смирнов; Ed. by [У+FFFД] пособие. — ТУСУР, 2007.
- 10 “Гигиенические требования ПЭВМ и организации работы” / Ed. by [У+FFFД] 2.2.2/2.4.1340-03. — 2003.
- 11 “Типовая инструкция по охране труда при работе на персональном компьютере” / Ed. by [У+FFFД] Р-45-084-01. — 2001.

Приложение А (обязательное)

Текст графического модуля

```

import QtQuick 2.4
import QtQuick.Controls 1.3
Item {
    width: 400
    height: 240
    ComboBox {
        id: comboBox1
        x: 150
        y: 0
        model: [ "B-spline", "Bilinear", "Bicubic" ]
    }
    Label {
        id: countInterpolation
        x: 20
        y: 5
        text: qsTr("Interpolation")
    }
    SpinBox {
        id: countIteration
        x: 150
        y: 40
        minimumValue : 1
        maximumValue : 100
        value : 10
    }
    SpinBox {
        id: windowSearch
        x: 150
        y: 80
        minimumValue : 2
        maximumValue : 1000
        value : 16
    }
    SpinBox {
        id: stepGrid
        x: 150
        y: 120
        minimumValue : 1
        maximumValue : 500
        value : 16
    }
    Label {
        id: countIterationLabel
        x: 20
        y: 40
        text: qsTr("Count iteration")
    }
    Label {
        id: labelWindowSearch
        x: 20
        y: 80
        text: qsTr("Size window search")
    }
    Label {

```

```

        id: labelStepGrid
        x: 20
        y: 120
        text: qsTr( "Step for grid" )
    }
    TextField {
        id: outDirField
        objectName: "outDirField"
        x: 20
        y: 160
        placeholderText: qsTr( "Output directory" )
    }
    Button {
        id: buttonOutDirField
        x: 150
        y: 160
        text: qsTr( "Browse" )
    }
    CheckBox {
        id: debugCheckBox
        x: 20
        y: 200
        text: qsTr( "More log" )
        checked: false
    }
    CheckBox {
        id: pyramidCheckBox
        x: 150
        y: 200
        text: qsTr( "Use pyramid" )
        checked: true
    }
    function getcmd( )
    {
        return "lucas_kanade";
    }
    function getarg( )
    {
        var a =
        [
            "-l", outDirField.text + "/list", //load list image
            "-o", outDirField.text, //output directory
            "-i", countIteration.value, //count iteration (1 by default)
            "-w", windowSearch.value, //size window search (3px by default)
            "-g", stepGrid.value, //step for grid (5px by default)
            "-b", comboBox1.currentIndex,
            pyramidCheckBox.checked ? "-p" : "",
            debugCheckBox.checked ? "-d" : "",
            /*"-h", //show help
            "-v" //show version*/
        ];
        return a;
    }
    function outdir( )
    {
        return outDirField.text;
    }
}

```

Приложение Б
(обязательное)
Документация программного обеспечения

Lucas Kanade
1.4

Создано системой Doxygen 1.8.9.1

Пт 5 Июн 2015 02:43:45

Содержание

1	Lucas–Kanade	1
2	Алфавитный указатель структур данных	3
2.1	Структуры данных	3
3	Список файлов	3
3.1	Файлы	3
4	Структуры данных	3
4.1	Структура imageInform	3
4.1.1	Подробное описание	3
4.2	Структура subSize	4
4.2.1	Подробное описание	4
5	Файлы	4
5.1	Файл src/global.h	4
5.1.1	Подробное описание	5
5.2	Файл src/lk_func.h	5
5.2.1	Подробное описание	6
5.2.2	Функции	6
5.3	Файл src/lk_interpolation.h	9
5.3.1	Подробное описание	10
5.3.2	Функции	10
5.4	Файл src/lk_struct.h	10
5.4.1	Подробное описание	11

1 Lucas–Kanade

In computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion.

By combining information from several nearby pixels, the Lucas–Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

Version

1.2

Installation

You need install some packet: ##### Qt

```
1 $ sudo apt-get install qt-sdk
2 $ sudo apt-get install libarchive-dev libqt5webkit5-dev
```

CMAKE

```
1 $ sudo apt-get install cmake
```

HDF5 Software

```
1 $ wget http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.8.14.tar
2 $ tar -xvf hdf5-1.8.14.tar
3 $ cd hdf5-1.8.14
4 $ cmake CMakeLists.txt
5 $ make
6 $ sudo make install
```

DV(Deformation analys)

```
1 $ cp /media/flash_card/dv.zip
2 $ unzip dv.zip
3 $ cd dv/df-cl
4 $ cmake CMakeLists.txt
5 $ make
```

Lucas-Kanade

```
1 $ git clone git@github.com:IgorPolyakov/graduate-work.git
2 $ cd graduate-work
3 $ cmake CMakeLists.txt
4 $ make
```

Run

```
1 $ ./bin/lukas_kanade_qt -l <list_of_image>
```

Options

- l — load list image
- o — output directory
- i — count iteration (1 by default)
- w — size window search (3px by default)
- g — step for grid (5px by default)
- v — show version
- h — show help
- d — debug mod on

Todo's

- Added pyramid
- Write Tests
- Add Code Comments for doxygen

License

GNU GPL

Repository created in order to write a graduate work on specialty 220301.

2 Алфавитный указатель структур данных

2.1 Структуры данных

Структуры данных с их кратким описанием.

[imageInform](#)

The [imageInform](#) struct Хранение информации о размерах изображения

3

[subSize](#)

The [SubSize](#) struct Данный файл содержит в себе определения основных структур, используемых в программе

4

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

[src/global.h](#)

Заголовочный файл с описанием глобальных переменных

4

[src/lk_func.h](#)

Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе

5

[src/lk_interpolation.h](#)

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполяцию данных

9

[src/lk_struct.h](#)

Заголовочный файл с описанием используемых структур

10

[src/version.h](#)

??

4 Структуры данных

4.1 Структура imageInform

The [imageInform](#) struct Хранение информации о размерах изображения

```
#include <lk_struct.h>
```

Поля данных

- int [height](#)

Размер изображения по высоте

- int [width](#)

Размер изображения по ширине

4.1.1 Подробное описание

The [imageInform](#) struct Хранение информации о размерах изображения

Объявления и описания членов структуры находятся в файле:

- src/lk_struct.h

4.2 Структура subSize

The SubSize struct Данный файл содержит в себе определения основных структур, используемых в программе

```
#include <lk_struct.h>
```

Поля данных

- int **x_1**
x_1 Координата пикселя по x для левого изображения
- int **y_1**
y_1 Координата пикселя по y для левого изображения
- int **rc**
rc Радиус окна, т.е. если radiusCode = 2, то сетка 5x5
- int **step**
step - Шаг между векторами
- int **x_2**
x_2 Координата пикселя по x для правого изображения
- int **y_2**
y_2 Координата пикселя по y для правого изображения

4.2.1 Подробное описание

The SubSize struct Данный файл содержит в себе определения основных структур, используемых в программе

Объявления и описания членов структуры находятся в файле:

- src/lk_struct.h

5 Файлы

5.1 Файл src/global.h

Заголовочный файл с описанием глобальных переменных

```
#include <QString>
```

Переменные

- bool **g_isDebug**
Глобальная переменная Включить режим вывода в консоль отладочной информации
- int **g_sizeWindowSearch**
Глобальная переменная Размер окна поиска
- int **g_stepForGrid**
Глобальная переменная Размер шага сетки (через какое расстояние повторяются точки поиска)
- int **g_iteration**
Глобальная переменная Число итераций (уточнений)
- int **g_interpolation**

- Глобальная переменная Выбор метода интерполяции(0 - Б-сплайн, 1 - Билинейный, 3 - Бикубический)
- **QString g_outputFolder**
Глобальная переменная Директория для сохранения результатов работы
- **double g_fastProgBar**
Глобальная переменная Переменная хранящая промежуточный уровень прогресс бара
- **double g_slowProgBar**
Глобальная переменная Переменная хранящая полный уровень прогресс бара

5.1.1 Подробное описание

Заголовочный файл с описанием глобальных переменных

Данный файл содержит в себе определения глобальных переменных, используемых в программе

5.2 Файл src/lk_func.h

Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе

```
#include "dv.h"
#include "global.h"
#include "lk_interpolation.h"
#include "lk_struct.h"
#include <deprecated/dvfile.h>
#include <iostream>
#include <math.h>
#include <QDebug>
#include <QFile>
#include <vector>
#include "hdf5io.h"
```

Функции

- **Data2Db * resizeImage (Data2Db *image, int kK, QString prefix)**
resizeImage Функция масштабирования изображения, для построения пирамиды уменьшенных изображений
- **double * multipicMtrxAndVectr (double **array, int *vector)**
multipicMtrxAndVectr Произведение матрицы на вектор
- **int calcLvlPyramid (int cx, int cy, bool isPyramid)**
calcLvlPyramid Автоматическое вычисление количества уровней пирамиды на основе размеров изображения
- **std::vector< Data2Db * > * createPyramid_v2 (Data2Db *img, int lvl_pyramid, QString pref)**
createPyramid_v2 Выделение памяти для пирамиды изображений
- **Vec2d computeOptFlow (subSize *kernel, Data2Db *leftImg, Data2Db *rightImg, Vec2d &dv, double *d_mid_x, double *d_mid_y, double *d_max_x, double *d_min_x, double *d_min_y, double *d_max_y, double *d_avg_x, double *d_avg_y, double *v_mid_x, double *v_mid_y, double *v_max_x, double *v_min_x, double *v_min_y, double *v_max_y)**
computeOptFlow Вычисление вектора оптического потока
- **VF2d * computeGrid (Data2Db *leftImg, Data2Db *rightImg, VF2d *prev, QString info)**
computeGrid Строит сетку с ранее заданным шагом, в точках пересечения ищется вектор оптического потока.
- **void getImageInfo (imageInform *image, QString path)**
- **void inversion (Matx22d &A)**

- inversion Нахождения обратной матрицы(2 x 2).
- void **printProgressBar** (double fastProgBar, double slowProgBar)

printProgressBar Функция ответственная за отрисовку прогресс бара при использовании графического интерфейса
- void **saveVfResult** (VF2d &vf, QString info)

saveVfResult Сохранение векторного поля в формате VF. Просмотр возможен в программе dfcl
- void **derivativeVectorField** (VF2d &vf, QString info)

derivativeVectorField Вычисление поля деформации
- void **writeHdf5File** (QString path, ProtoData2D &layer, bool writeMode)

writeHdf5File Сохранение всех результатов в формат HD5F

5.2.1 Подробное описание

Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе

5.2.2 Функции

5.2.2.1 int calcLvlPyramid (int cx, int cy, bool isPyramid)

calcLvlPyramid Автоматическое вычисление количества уровней пирамиды на основе размеров изображения

Аргументы

in	cx	Ширина изображения
in	cy	Высота изображения
in	isPyramid	Вычислять ли уровень пирамиды

Возвращает

[out] Колличество уровней пирамиды

5.2.2.2 VF2d* computeGrid (Data2Db * leftImg, Data2Db * rightImg, VF2d * prev, QString info)

computeGrid Строит сетку с ранее заданным шагом, в точках пересечения ищется вектор оптического потока.

Аргументы

in	leftImg	указатель на массив яркостей первого кадра
in	rightImg	указатель на массив яркостей второго кадра
in	prev	указатель на векторное поле содержащий предыдущий уровень пирамиды
in	info	имя слоя содержащего векторное поле смещений

Возвращает

[out] двумерный массив содержащий векторное поле, в формате VF

5.2.2.3 Vec2d computeOptFlow (subSize * kernel, Data2Db * leftImg, Data2Db * rightImg, Vec2d & dv, double * d_mid_x, double * d_mid_y, double * d_max_x, double * d_min_x, double * d_min_y, double * d_max_y, double * d_avg_x, double * d_avg_y, double * v_mid_x, double * v_mid_y, double * v_max_x, double * v_min_x, double * v_min_y, double * v_max_y)

computeOptFlow Вычисление вектора оптического потока

Аргументы

in	kernel	структура содержащая сведения о местонахождении пикселя, размерах окна поиска и прочего
in	leftImg	массив яркостей первого кадра
in	rightImg	массив яркостей второго кадра
in	dv	векторное поле предыдущего поля

Возвращает

[out] vf вектор оптического потока

B-spline interpolation or Bilinear interpolation

5.2.2.4 std::vector<Data2Db*>* createPyramid_v2 (Data2Db * img, int lvl_pyramid, QString pref)

createPyramid_v2 Выделение памяти для пирамиды изображений

Аргументы

in	img	Указатель на оригинальное изображение
in	lvl_pyramid	Уровень пирамиды
in	pref	Префикс имени для создаваемого слоя

Возвращает

[out] Список изображений

5.2.2.5 void derivativeVectorField (VF2d & vf, QString info)

derivativeVectorField Вычисление поля деформации

Аргументы

vf	Векторное поле
info	Имя сохраняемого файла

0.81649658092 - Это $\sqrt{2/3}$

5.2.2.6 void inversion (Matx22d & A)

inversion Нахождения обратной матрицы(2 x 2).

Аргументы

in	A	Ссылка на массив
----	---	------------------

5.2.2.7 double* multiplicMtrxAAndVectr (double ** array, int * vector)

multiplicMtrxAAndVectr Произведение матрицы на вектор

Аргументы

in	**array	указатель на массив
in	*vector	указатель на вектор

Возвращает

[out] tmp результат произведения

5.2.2.8 void printProgressBar (double fastProgBar, double slowProgBar)

printProgressBar Функция ответственная за отрисовку прогресс бара при использовании графического интерфейса

Аргументы

fastProgBar	Промежуточный уровень
slowProgBar	Полный уровень

5.2.2.9 Data2Db* resizeImage (Data2Db * image, int kK, QString prefix)

resizeImage Функция масштабирования изображения, для построения пирамиды уменьшенных изображений

Аргументы

in	image	Структура содержащая сведения о размерах масштабируемого изображения
in	kK	Коэффициент уменьшения изображения
in	prefix	Префикс имени для создаваемого слоя

Возвращает

[out] указатель на массив масштабированных изображений

5.2.2.10 void saveVfResult (VF2d & vf, QString info)

saveVfResult Сохранение векторного поля в формате VF. Просмотр возможен в программе dfcl

Аргументы

in	vf	Указатель на векторное поле
in	info	Имя сохраняемого файла

5.2.2.11 void writeHdf5File (QString filename, ProtoData2D & layer, bool writeMode)

writeHdf5File Сохранение всех результатов в формат HD5F

Аргументы

filename	Имя сохраняемого файла
layer	Указатель на слой
writeMode	Режим записи для данного слоя

5.3 Файл src/lk_interpolation.h

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполяцию данных.

```
#include <deprecated/dvfile.h>
```

Функции

- double **bilinearInterpolation** (double delx, double dely, uchar q11, uchar q12, uchar q21, uchar q22, int x1, int y1)

bilinearInterpolation
- double **cubicInterpolate** (double p[4], double x)

cubicInterpolate Кубическая интерполяция
- double **bicubicInterpolate** (double p[4][4], double x, double y)

bicubicInterpolate Бикубическая интерполяция

5.3.1 Подробное описание

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполяцию данных.

5.3.2 Функции

5.3.2.1 double bicubicInterpolate (double p[4][4], double x, double y)

bicubicInterpolate Бикубическая интерполяция

Аргументы

in	p	
in	x	
in	y	

Возвращает

5.3.2.2 double bilinearInterpolation (double delx, double dely, uchar q11, uchar q12, uchar q21, uchar q22, int x1, int y1)

bilinearInterpolation

Аргументы

delx	Смещение по x
dely	Смещение по y
q11	Значение функции в левом верхнем углу
q12	Значение функции в левом нижнем углу
q21	Значение функции в правом верхнем углу
q22	Значение функции в правом нижнем углу
x1	Координата верхнего левого угла по x
y1	Координата верхнего левого угла по y

Возвращает

5.3.2.3 double cubicInterpolate (double p[4], double x)

cubicInterpolate Кубическая интерполяция

Аргументы

in	p	
in	x	

Возвращает

5.4 Файл src/lk_struct.h

Заголовочный файл с описанием используемых структур

Структуры данных

- struct **subSize**

The SubSize struct Данный файл содержит в себе определения основных структур, используемых в программе

- struct **imageInform**

The **imageInform** struct Хранение информации о размерах изображения

5.4.1 Подробное описание

Заголовочный файл с описанием используемых структур

