

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики

Работа допущена к защите
Должность руководителя М
_____ И.О. Фамилия
« _____ » _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
РАБОТА БАКАЛАВРА
РАЗРАБОТКА МЕТОДА ОПРЕДЕЛЕНИЯ ЭКСПРЕССИИ ГЕНОВ ПО
ИЗОБРАЖЕНИЯМ МОЗГА ПЛОДОВОЙ МУШКИ
по направлению подготовки 01.03.02 Прикладная математика и информатика
Направленность (профиль) 01.03.02_04 Биоинформатика

Выполнил
студент гр. 3630102/70401

Р.А. Темиргалиев

Руководитель
должность,
степень, звание

К.Н. Козлов

Консультант
должность, степень

И.О. Фамилия

Консультант
по нормоконтролю

И.О. Фамилия

Санкт-Петербург
2021

СОДЕРЖАНИЕ

Введение	3
Глава 1. Постановка задачи и описание исходных данных.....	4
1.1. Постановка задачи.....	4
1.2. Описание исходных данных	4
1.3. Сложность задачи	5
1.4. Существующие методы решений	6
Глава 2. Описание используемых методов	8
2.1. Методы обработки изображений реализованные в пакете ProStack.....	8
2.2. Название параграфа	15
2.2.1. Название подпараграфа	15
2.3. Название параграфа	17
2.4. Выводы	23
Список использованных источников.....	25

ВВЕДЕНИЕ

Для исследования генной регуляции требуется получать количественные данные по экспрессии генов с учетом пространственной локализации.

Рассмотрим следующую задачу для решения которой изучается генная регуляция. Итак, сначала по изображениям мозга плодовой мушки измеряют уровень экспрессии генов, то есть получают количественные данные с учетом пространственной локализации генов. Далее полученные данные используют для изучения поведения мух в период спаривания - сравнивают уровни экспрессии генов в разных частях мозга у мушек разных полов с их поведением в период спаривания. Под поведением можно понимать их привлекательность друг другу, желание спариваться и др. Далее эти статистические связи от модельных объектов(мушек) можно попробовать распространить на более сложные организмы(мыши, собаки и др.)

В данной работе изучается получение количественных данных которые можно было бы использовать для приведенной задачи выше. Для выделения на экспериментальных изображениях комплексов молекул РНК будут использованы методы обработки изображений.

Целью данной работы является разработка алгоритма для выделения на экспериментальных изображениях комплексов молекул РНК и применение для анализа паттернов экспрессии генов в мозге плодовой мушки. Работа проводится на примере изображений плодовой мушки.

Для достижения поставленной цели требуется решить следующие задачи:

1. Изучить методы разделения каналов в экспериментальных биологических изображениях и подобрать пригодные для тестирования в имеющихся данных. Проверить работу методов на тестовых данных из соответствующих статей.
2. Модифицировать и запрограммировать отобранные методы для процедуры обработки имеющихся данных по экспрессии генов в мозге плодовой мушки, выделить настроечные параметры.
3. Получить количественные данные по экспрессии генов в мозге плодовой мушки по имеющимся изображениям.
4. Проанализировать различия в экспрессии генов для разных условий.

ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ И ОПИСАНИЕ ИСХОДНЫХ ДАННЫХ

1.1. Постановка задачи

Задача заключается в модификации и улучшении механизма обработки изображений реализованного в пакете **ProStack** [prostack].

В работе будут подробно описаны модификации, улучшения, и новые методы встроенные в пакет.

1.2. Описание исходных данных

Исходные данные представляют собой трехмерные двухканальные изображения полученные с помощью конфокального микроскопа(тут хочется написать про автора этих изображений).

Современные конфокальные микроскопы обычно имеют несколько фотоприемных каналов, благодаря которым можно получить изображения одновременно в нескольких спектральных областях, т.е. использовать несколько флуорохромов.

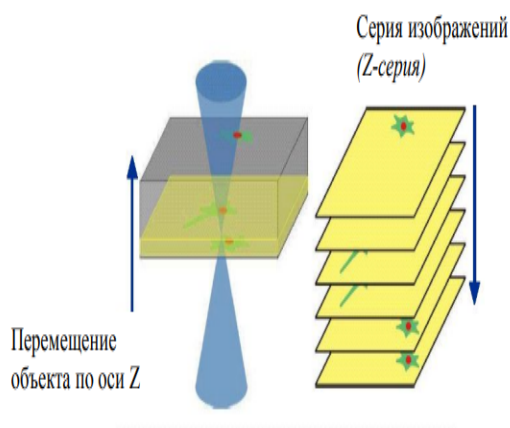


Рис.1.1. Получение серии оптических срезов(Z-серия).

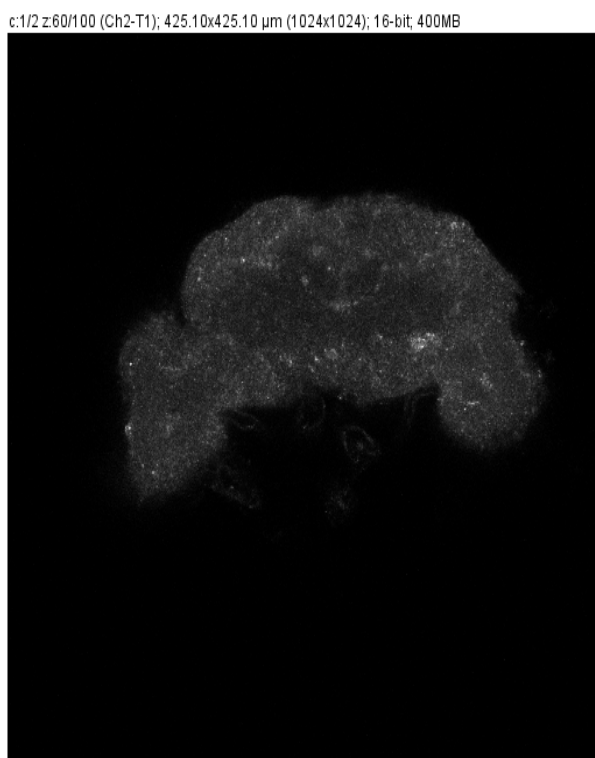


Рис.1.2. Пример трехмерного двухканального изображения мозга плодовой мушки.

1.3. Сложность задачи

Существует некоторая проблема в решении поставленной задачи. В исходных изображениях наблюдается паразитное свечение из одного канала микроскопа в другом, что вредит выделению частиц. Данное явление называется автофлуоресценцией. Существует несколько методов решения этой проблемы, которые позволяют уменьшить этот эффект, однако конкретный метод и параметры надо тестировать с конкретными изображениями. Также может понадобиться проводить предобработку и модифицировать последующие шаги всей процедуры.

1.4. Существующие методы решений

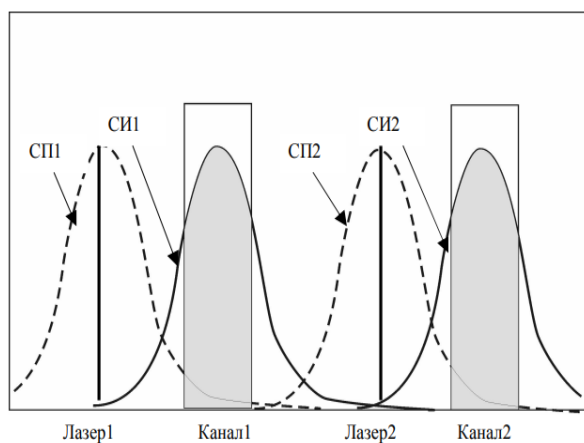


Рис.1.3. Перекрывание спектров полностью отсутствует. СП – спектры поглощения, СИ – спектры испускания флуорохромов 1 и 2.

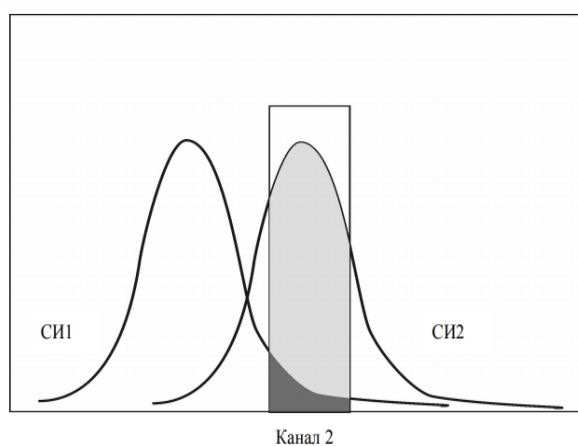


Рис.1.4. Слабое перекрытие спектров. Обозначения те же.

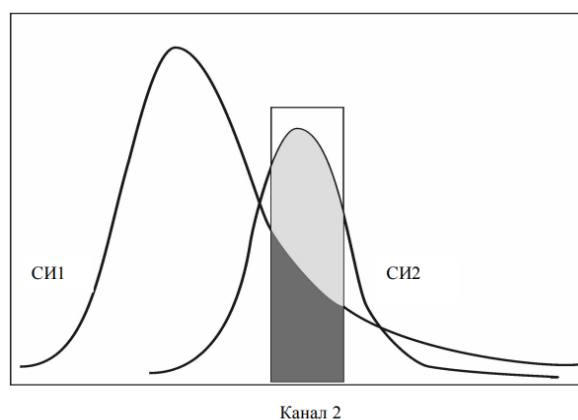


Рис. 4.7. Сильное перекрытие спектров. Обозначения те же.

Рис.1.5. Сильное перекрытие спектров. Обозначения те же

Существуют следующие возможные случаи взаимодействия сигналов от двух флуорохромов. Наилучший вариант показан на Рис.1.3 Перекрытия спектров нет.

На Рис. 1.4 представлен вариант со слабым перекрытием спектров. Часть спектра испускания первого флуорохрома попадает во второй фотоприемный канал. Для это случая нужна небольшая предобработка. Уменьшить перекрытие можно путём уменьшения мощности первого лазера. Чтобы яркость изображения не падала, нужно усилить первый канал, а также сдвинуть полосу приема второго канала вправо.

В последнем случае - когда перекрытие сильное(Рис. 1.5) необходимо применять последовательное сканирование. То есть сначала включить лазер и фотоприемник для первого канала, затем отключить и потоврить для второго канала (режим Multitrack на LSM).

Также существует программный способ уменьшения перекрытия спектров. Он основан на различных математических алгоритмах, учитывающих информацию о спектрах применяемых красителей, применяющих методы линейной алгебры, адаптивном или ручном разделении изображений по приемным каналам. Именно такой способ будет рассмотрен в данной работе.

Наиболее эффективным способом избежания перекрытия спектров является последовательное сканирование, однако у этого подхода есть несколько ограничений.Он требует использования специализированного оборудования и запатентованного программного обеспечения, что является ограничивающим фактором в его широком использовании. Также, например, получения изображений для каждого фотоприемного канала значительно увеличивает время получения изображений и объем данных.

ГЛАВА 2. ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ МЕТОДОВ

2.1. Методы обработки изображений реализованные в пакете ProStack

В данной работе производится модификация и улучшение методов обработки изображений реализованных в пакете ProStack.

В пакете ProStack реализованы стандартные и проблемно-ориентированные методы обработки изображений а также методы для получения количественных данных из изображений, полученных на световом или конфокальном микроскопе. Пакет имеет графический интерфейс, для построения сложных сценариев.

Механизм обработки изображения в данном пакете представляет из себя своего рода конвейер - множество зависимых друг от друга различных операций, записанных в один сценарий.

Все методы в рамках пакета разделены на десять классов.

- Комбинирование (Получение одного изображения из нескольких входов)
- Выделение объектов
- Корректировка (Повышение качества изображения)
- Сегментация (Разделение изображения на части/зоны)
- Восстановление
- Морфология (Морфологические операции)
- Геометрия (Изменение свойств изображений)
- Преобразование
- Арифметика (Алгебраические операции)
- Разное

Рассмотрим некоторые морфологические операции, которые реализованы в пакете, а также применялись для извлечение количественных данных из изображений мозга мушки.

Для улучшения сегментации(например обработки фона) применяют операцию морфологического размыкания - комбинацию операций эрозии и наращивания. Рассмотрим операции на простом примере. Пусть у нас есть следующее бинарное изображение и структурный элемент:

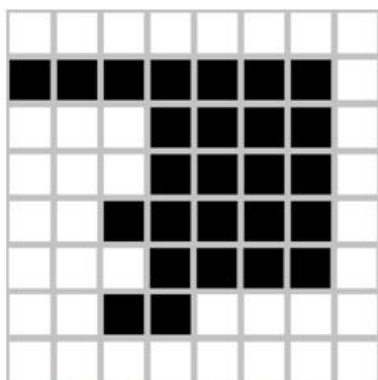


Рис.2.1. Бинарное изображение В

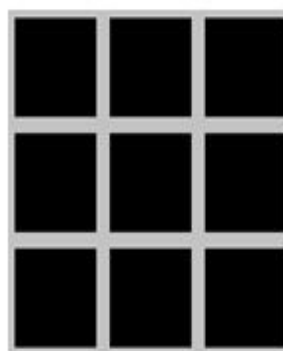


Рис.2.2. Структурный элемент S

Нарращивание — Структурный элемент S применяется ко всем пикселям бинарного изображения. Каждый раз, когда начало координат структурного элемента совмещается с единичным бинарным пикселем, ко всему структурному элементу применяется перенос и последующее логическое сложение с соответствующими пикселями бинарного изображения. Результаты логического сложения записываются в выходное бинарное изображение, которое изначально инициализируется пустыми значениями.

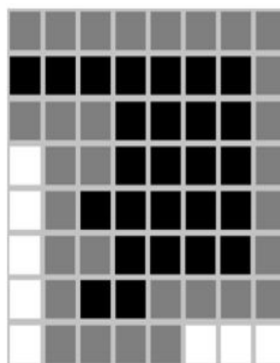


Рис.2.3. Нарращивание бинарного изображения структурным элементом

Эрозия — При выполнении операции эрозии структурный элемент тоже проходит по всем пикселям изображения. Если в некоторой позиции каждый единичный пиксел структурного элемента совпадет с единичным пикселем бинарного изображения, то выполняется логическое сложение центрального пиксела структурного элемента с соответствующим пикселем выходного изображения.

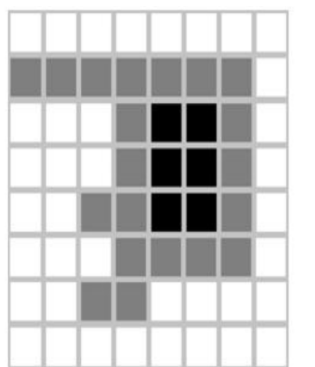


Рис.2.4. Эрозия бинарного изображения структурным элементом

В результате применения операции эрозии все объекты, меньшие чем структурный элемент, стираются, объекты, соединённые тонкими линиями становятся разъединёнными и размеры всех объектов уменьшаются.

Размыкание — Операция эрозии полезна для удаления малых объектов и различных шумов, но у этой операции есть недостаток – все остающиеся объекты уменьшаются в размере. Этого эффекта можно избежать, если после операции эрозии применить операцию наращивания с тем же структурным элементом. Размыкание отсеивает все объекты, меньшие чем структурный элемент, но при этом помогает избежать сильного уменьшения размера объектов. Также размыкание идеально подходит для удаления линий, толщина которых меньше, чем диаметр структурного элемента. Также важно помнить, что после этой операции контуры объектов становятся более гладкими.

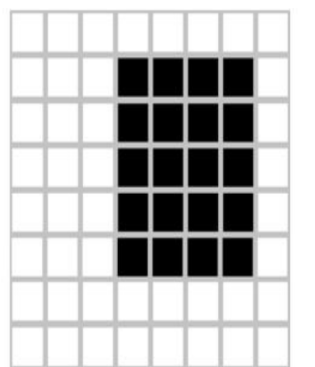


Рис.2.5. Размыкание бинарного изображения структурным элементом

Далее рассмотрим довольно распространенную операцию в обработке изображений - *выделение границ*. Данная операция, а именно оператор Кэнни был применен в этой работе для выделения комплексов молекул РНК.

Границы объектов на изображении в значительной степени уменьшают количество данных, которые необходимо обработать, и в то же время сохраняет

важную информацию об объектах на изображении, их форму, размер, количество. Главной особенностью техники обнаружения границ является возможность извлечь точную линию с хорошей ориентацией.

Граница является местным изменением яркости на изображении. Они, как правило, проходят по краю между двумя областями. С помощью границ можно получить базовые знания об изображении. Функции их получения используются передовыми алгоритмами компьютерного зрения и таких областях, как медицинская обработка изображений, биометрия и тому подобные. Обнаружение границ — активная область исследований, так как он облегчает высокоуровневый анализ изображений. На полутоновых изображениях существует три вида разрывов: точка, линия и граница. Для обнаружения всех трех видов неоднородностей могут быть использованы пространственные маски.

В технической литературе приведено и описано большое количество алгоритмов выделения контуров и границ. В данной работе рассмотрены наиболее популярные методы. К ним относятся: оператор Робертса, Собеля, Превитта, алгоритм Кэнни.

Фильтрация

Все следующие методы основываются на одном из базовых свойств сигнала яркости — **разрывности** (резких изменений значений интенсивности). Наиболее общим способом поиска **разрывов** является обработка изображения с помощью скользящей маски, называемой также фильтром, ядром, окном или шаблоном, которая представляет собой некую квадратную матрицу, соответствующую указанной группе пикселей исходного изображения. Элементы матрицы принято называть коэффициентами. Оперирование такой матрицей в каких-либо локальных преобразованиях называется **фильтрацией** или **пространственной фильтрацией**. Схема пространственной фильтрации иллюстрируется на рисунке 2.6:

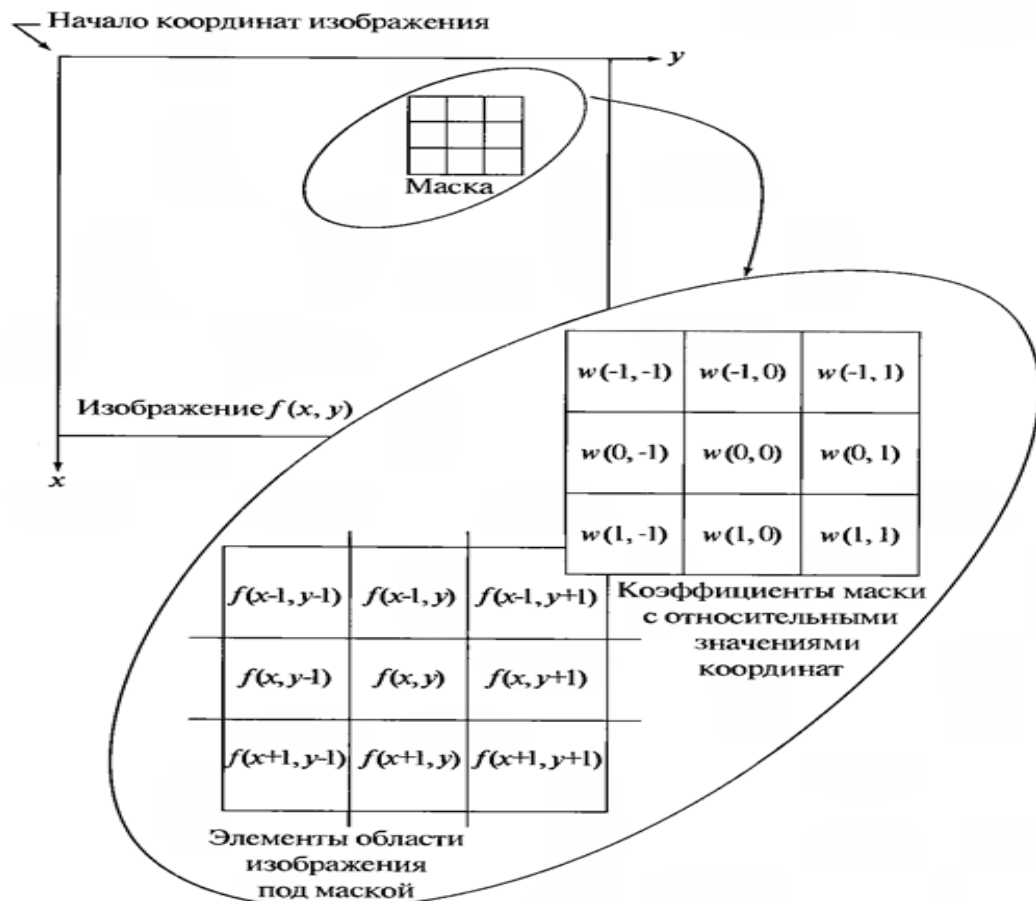


Рис.2.6. Схема пространственной фильтрации

Процесс основан на простом перемещении маски фильтра от точки к точке изображения; в каждой точке (x, y) отклик фильтра вычисляется с использованием предварительно заданных связей. В случае линейной пространственной фильтрации отклик задается суммой произведения коэффициентов фильтра на соответствующие значения пикселей в области, покрытой маской фильтра. Для маски 3×3 элемента, показанной на рисунке 2.6, результат (отклик) R линейной фильтрации в точке (x, y) изображения составит: $R = w(-1, -1) * f(x - 1, y - 1) + w(-1, 0) * f(x - 1, y) + \dots + w(0, 0) * f(x, y) + \dots + w(1, 0) * f(x + 1, y) + w(1, 1) * f(x + 1, y + 1)$ что, как видно, есть сумма произведений коэффициентов маски на значения пикселей непосредственно под маской. В частности заметим, что коэффициент $w(0, 0)$ стоит при значении $f(x, y)$, указывая тем самым, что маска центрирована в точке (x, y) .

При обнаружении перепадов яркости используются дискретные аналоги производных первого и второго порядка. Первая производная одномерной функции $f(x)$ определяется как разность значений соседних элементов:

$$\frac{df}{dx} = f(x + 1) - f(x).$$

Аналогично, вторая производная определяется как разность соседних значений

первой производной:

$$\frac{df^2}{dx^2} = f(x+1) - f(x-1) - 2f(x).$$

По определению, градиент изображения $f(x,y)$ в точке (x,y) — это вектор:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}.$$

Направление вектора градиента совпадает с направлением максимальной скорости изменения функции f в точке (x,y) . Важную роль при обнаружении контуров играет модуль этого вектора, который обозначается $|\nabla f|$ и равен

$|\nabla f| = \sqrt{G_x^2 + G_y^2}$. Эта величина равна значению максимальной скорости изменения функции f в точке (x,y) . **[find-contours-habr]**

Оператор Робертса

Пусть область 3×3 , показанная на рисунке ниже (см. 2.7), представляет собой значения яркости в окрестности некоторого элемента изображения.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Рис.2.7. Окрестность 3×3 внутри изображения

Один из простейших способов нахождения первых частных производных в точке z_5 состоит в применении следующего перекрестного градиентного оператора Робертса: $G_x = z_9 - z_1$ и $G_y = z_8 - z_6$

Эти производные могут быть реализованы путем обработки всего изображения с помощью оператора, описываемого масками на рисунке 2.8, используя процедуру фильтрации, описанную ранее.

-1	0
0	1

0	-1
1	0

Рис.2.8. Маски оператора Робертса

Реализация масок размерами 2×2 не очень удобна, т.к. у них нет четко выраженного центрального элемента, что существенно отражается на результате выполнения фильтрации. Но этот «минус» порождает очень полезное свойство данного алгоритма – высокую скорость обработки изображения.

Оператор Превитта

Оператор Превитта, так же как и оператор Робертса, оперирует с областью изображения 3x3, представленной на рисунке 2.7, только использование такой маски задается другими выражениями:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \text{ и } G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

В этих формулах разность между суммами по верхней и нижней строкам окрестности 3x3 является приближенным значением производной по оси x, а разность между суммами по первому и последнему столбцам этой окрестности – производной по оси y. Для реализации этих формул используется оператор, описываемый масками на рисунке 2.9, который называется оператором Превитта.

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Рис.2.9. Маски оператора Превитта

Оператор Собеля

Оператор Собеля тоже использует область изображения 3x3, отображенную на рисунке 2.7. Он довольно похож на оператор Превитта, а видоизменение заключается в использовании весового коэффициента 2 для средних элементов:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \text{ и } G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Это увеличенное значение используется для уменьшения эффекта сглаживания за счет придания большего веса средним точкам.

Маски, используемые оператором Собеля, отображены на рисунке ниже (см. рис. 2.10).

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Рис.2.10. Маски оператора Собеля

Рассмотренные выше маски применяются для получения составляющих градиента G_x и G_y . Для вычисления величины градиента эти составляющие необходимо использовать совместно:

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}$$

Еще один алгоритм который является модификацией вышеперечисленных а также применялся в данной работе:

Детектор границ Канни

Детектор границ Канни является одной из самых популярных алгоритмов обнаружения контуров. Впервые он был предложен Джоном Канни в магистерской диссертации в 1983 году, и до сих пор является лучше многих алгоритмов, разработанных позднее. Важным шагом в данном алгоритме является устранение шума на контурах, который в значительной мере может повлиять на результат, при этом необходимо максимально сохранить границы. Для этого необходим тщательный подбор порогового значения при обработке данным методом.

Алгоритм:

- Размытие исходного изображения
- Выполнить поиск градиента. Границы намечаются там, где градиент принимает максимальное значение
- Подавление не-максимумов. Только локальные максимумы отмечаются как границы
- Итоговые границы определяются путем подавления всех краев, не связанных с определенными границами.

В отличии от операторов Робертса и Собеля, алгоритм Канни не очень восприимчив к шуму на изображении.

2.2. Название параграфа

Название параграфа оформляется с помощью команды `\section{...}`, название главы — `\chapter{...}`.

2.2.1. Название подпараграфа

Название подпараграфа оформляется с помощью команды `\subsection{...}`.

Использование подподпараграфов в основной части крайне не рекомендуется. В случае использования, необходимо вынести данный номер в содержание. Название подпараграфа оформляется с помощью команды `\subsubsection{...}`.

Вместо подподпараграфов рекомендовано использовать перечисления.

Перечисления могут быть с нумерационной частью и без неё и использоваться с иерархией и без иерархии. Нумерационная часть при этом формируется следующим способом:

1. в перечислениях *без иерархии* оформляется арабскими цифрами с точкой (или длинным тире).
2. В перечислениях *с иерархией* — в последовательности сначала прописных латинских букв с точкой, затем арабских цифр с точкой и далее — строчных латинских букв со скобкой.

Далее приведён пример перечислений с иерархией.

- A. Первый пункт.
- B. Второй пункт.
- C. Третий пункт.
- D. По ГОСТ 2.105–95 [**gost-russian-text-documents**] первый уровень нумерации идёт буквами русского или латинского алфавитов (*для определенности выбираем английский алфавит*), а второй — цифрами.

1. В данном пункте лежит следующий нумерованный список:

- a) первый пункт;
- b) третий уровень нумерации не нормирован ГОСТ 2.105–95 (*для определенности выбираем английский алфавит*);
- c) обращаем внимание на строчность букв в этом нумерованном и следующем маркированном списке:
— первый пункт маркированного списка.

E. Пятый пункт верхнего уровня перечисления.

Маркированный список (без нумерационной части) используется, если нет необходимости ссылки на определенное положение в списке:

- первый пункт с *маленькой буквы* по правилам русского языка;
- второй пункт с *маленькой буквы* по правилам русского языка.

Оформление псевдокода необходимо осуществлять с помощью пакета `algorithm2e` в окружении `algorithm`. Данное окружение интерпретируется в

шаблоне как рисунок. Пример оформления псевдокода алгоритма приведён на рис.2.11.

Algorithm

Input: the many-valued context $\mathbb{M} \stackrel{\text{def}}{=} (G, M, W, J)$, the class membership $\varepsilon : G \rightarrow K$

Output: positive and negative binary contexts $\overline{\mathbb{K}}_+ \stackrel{\text{def}}{=} (\overline{G}_+, M, I_+)$, $\overline{\mathbb{K}}_- \stackrel{\text{def}}{=} (\overline{G}_-, M, I_-)$ such that i-tests found in $\overline{\mathbb{K}}_+$ are diagnostic tests in \mathbb{M} , and objects from $\overline{\mathbb{K}}_-$ are counter-examples

1. **for** $\forall g_i, g_j \in G$ **do**
2. **if** $i < j$ **then**
3. $\overline{G} \leftarrow (g_i, g_j);$
4. **for** $\forall (g_i, g_j) \in \overline{G}$ **do**
5. **if** $m(g_i) = m(g_j)$ **then**
6. $(g_i, g_j)Im;$
7. **if** $\varepsilon(g_i) = \varepsilon(g_j)$ **then**
8. $\overline{G}_+ \leftarrow (g_i, g_j);$
9. **else** $\overline{G}_- \leftarrow (g_i, g_j);$
10. $I_+ = I \cap (\overline{G}_+ \times M), I_- = I \cap (\overline{G}_- \times M);$
11. **for** $\forall \overline{g}_+ \in \overline{G}_+, \forall \overline{g}_- \in \overline{G}_-$ **do**
12. **if** $\overline{g}_+^{\uparrow} \subseteq \overline{g}_-^{\uparrow}$ **then**
13. $\overline{G}_+ \leftarrow \overline{G}_+ \setminus \overline{g}_+;$

Рис.2.11. Псевдокод алгоритма DiagnosticTestsScalingAndInferring
[Naidenova2017]

Обратим внимание, что можно сослаться на строчку 1 псевдокода из рис.2.11.

2.3. Название параграфа

Одиночные формулы также, как и отдельные формулы в составе группы, могут быть размещены в несколько строк. Чтобы выставить номер формулы напротив средней строки, используйте окружение `multlined` из пакета `mathtools`

следующим образом [Ganter1999]:

$$\begin{aligned}
 (A_1, B_1) &\leq (A_2, B_2) \Leftrightarrow \\
 &\Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow \\
 &\Leftrightarrow B_2 \subseteq B_1.
 \end{aligned}
 \tag{2.1}$$

Используя команду `\labelcref{...}` из пакета `cleveref`, допустимо оформить ссылку на несколько формул, например, (???? ??).

Пример оформления четырёх иллюстраций в одном текстово-графическом объекте приведён на рис.2.12. Это возможно благодаря использованию пакета `subcaption`.

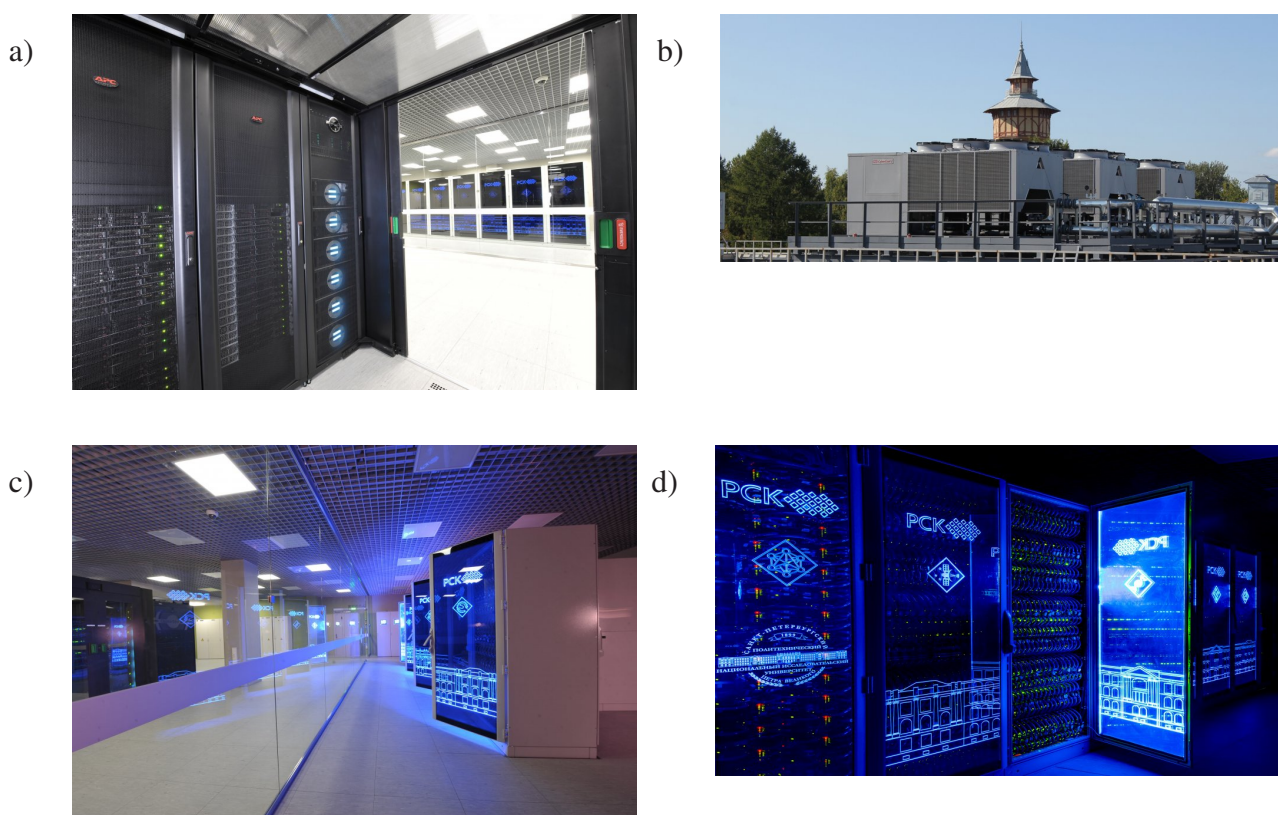


Рис.2.12. Фотографии суперкомпьютерного центра СПбПУ [spbpu-gallery]: *a* — система хранения данных и узлы NUMA-вычислителя; *b* — холодильные машины на крыше научно-исследовательского корпуса; *c* — машинный зал; *d* — элементы вычислительных устройств

Далее можно ссылаться на составные части данного рисунка как на самостоятельные объекты: рис.2.12а, рис.2.12b, рис.2.12с, рис.2.12d или на три из четырёх изображений одновременно: рис.2.12а–2.12с.

Приведём пример табличного представления данных с записью продолжения на следующей странице на табл.2.1.

Таблица 2.1

Пример задания данных из [Peskov2004] (с повтором для переноса таблицы на новую страницу)

G	m_1	m_2	m_3	m_4	K
1	2	3	4	5	6
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2
g_1	0	1	1	0	1

Продолжение табл. 2.1

1	2	3	4	5	6
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.2

Пример представления данных для сквозного примера по ВКР [Peskov2004]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица 2.3

Пример задания данных в табличном виде из [Peskov2004] (с помощью окружения minipage)

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2



Рис.2.13. Новый научно-исследовательский корпус СПбПУ [spbpu-gallery] (с помощью окружения minipage)

Вопросы форматирования текстово-графических объектов (окружений) не регламентированы в известных нам ГОСТах, поэтому предлагаем придерживаться следующих правил:

- **полужирный текст** рекомендуем использовать только для названий стандартных окружений с нумерационной частью, например, для представления *впервые*: **определение 1.1, теорема 2.2, пример 2.3, лемма 4.5**;
- *курсив* рекомендуем использовать только для выделения переменных в формулах, служебной информации об авторах главы (статьи), важных терминов, представляемых по тексту, а также для всего тела окружений, связанных с получением *новых существенных результатов и их доказательством*: теорема, лемма, следствие, утверждение и другие.

По аналогии с нумерацией формул, рисунков и таблиц нумеруются и иные текстово-графические объекты, то есть включаем в нумерацию номер главы, например: теорема 3.1. для первой теоремы третьей главы монографии. Команды \LaTeX выставляют нумерацию и форматирование автоматически. Полный перечень команд для подготовки текстово-графических и иных объектов находится в подробных методических рекомендациях [**spbp-bci-template-author-guide**].

Для удобства авторов названия стандартных окружений, рекомендованных к использованию, приведены в табл.2.4, а в табл.2.5 перечислены имена специально разработанных окружений для шаблонов SPbPU.

На базе пакета `tikz` разработано большое количество расширений [**ctan-tikz**], например, `tikzcd`, которые мы рекомендуем использовать для оформления иллюстраций.

В случае, если авторам потребовалось новое окружение, то создать его можно в файле `my_folder/my_settings.tex` согласно правилам, приведённым ниже.

1. Для перехода в режим создания окружений следует указать:
 - `\theoremstyle{myplain}` — окружения с доказательствами или аксиомами
 - `\theoremstyle{mydefinition}` — окружения, не связанные с доказательствами или аксиомами.
2. В команде создания окружения следует ввести краткий псевдоним (`m-new-env`) и отображаемое в pdf имя окружения (Название_окружения):
 - `\newtheorem{m-new-env-second}{Название_окружения} - [chapter]`.

Таблица 2.4

Стандартные окружения

Название окружения	Назначение
center	центрирование, аналог команды <code>\centering</code> , но с добавлением нежелательного пробела, поэтому лучше избегать применения <code>center</code>
itemize	перечисления, в которых нет необходимости нумеровать пункты (немаркированные списки)
enumerate	перечисления с нумерацией (немаркированные списки)
refsection	создание отдельных библиографических списков для глав
tabular	оформление таблиц
table	автоматическое перемещение по тексту таблиц, оформленных, например, с помощью <code>tabular</code> , для минимизации пустых пространств
longtable	оформление многостраничных таблиц
tikzpicture	создание иллюстраций с помощью пакета <code>tikz</code> [ctan-tikz]
figure	автоматическое перемещение по тексту рисунков, оформленных например, с помощью <code>tikz</code> или подключенных с помощью команды <code>\includegraphics</code> , для минимизации пустых пространств
subfigure	оформление вложенных рисунков в составе <code>figure</code>
algorithm	оформление псевдокода на основе пакета <code>algorithm2e</code> [ctan-algorithm2e]
minipage	оформление рисунков и таблиц без функций автоматического перемещения по тексту для минимизации пустых пространств
equation	оформление выключенных (не встроенных в текст с помощью <code>\$. . . \$</code>) одиночных формул на одной строке
multilined	оформление выключенных (не встроенных в текст с помощью <code>\$. . . \$</code>) одиночных формул в несколько строк
aligned	оформление нескольких формул с выравниванием по символу <code>&</code> .

Теорема 2.1 (о чем-то конкретном). *Текст теоремы полностью выделен курсивом. Допустимо математические символы не выделять курсивом, если это искажает их значения. Используется абзацный отступ, так как “Абзацы в тексте начинают отступом” в соответствии с ГОСТ 2.105–95. Название теоремы допустимо убрать. Доказательство окончено.*

Доказательство теоремы 2.1, леммы, утверждений, следствий и других подобных окружений (в последнем абзаце) завершаем предложением в котором сказано, что доказательство окончено. Например, доказательство теоремы 2.1 окончено.

Тело доказательства не выделяется курсивом. Тело следующих окружений также не выделяется сплошным курсивом: определение, условие, проблема, пример, упражнение, вопрос, гипотеза и другие.

Таблица 2.5

Специальные окружения

Название окружения	Текстово-графический объект
abstr	реферат (abstract)
m-theorem	теорема
m-corollary	следствие
m-proposition	утверждение
m-lemma	лемма
m-axiom	аксиома
m-example	пример
m-definition	определение
m-condition	условие
m-problem	проблема
m-exercise	упражнение
m-question	вопрос
m-hypothesis	гипотеза

Определение 2.1 (термин). В тексте определения только *важные термины* выделяются курсивом. Если определение носит лишь вспомогательный характер, то допустимо не использовать окружение `m-definition`, представляя текст определения в обычном абзаце. Ключевые термины при этом обязательно выделяются курсивом.

Вместо теоремо-подобных окружений для вставки небольших текстово-графических объектов иногда используются команды. Типичным примером такого подхода является команда `\footnote{text}`¹, где в аргументе `text` указывают текст *подстрочной ссылки (сноски)*. В них *нельзя добавлять веб-ссылки или цитировать литературу*. Для этих целей используется список литературы. Нумерация сносок сквозная по ВКР без точки на конце выставляется в шаблоне автоматически, однако в каждом приложении к ВКР нумерация, зависящая от номера приложения, выставляется префикс «П», например «П1.1» — первая сноска первого приложения.

2.4. Выводы

Текст заключения ко второй главе. Пример ссылок [**Book**; **Incollection**; **Manual**; **Mastersthesis**; **Misc**; **Phdthesis**; **Proceedings**; **Techreport**; **Unpublished**;

¹Внимание! Команда вставляется непосредственно после слова, куда вставляется сноска (без пробела). Лишние пробелы также не указываются внутри команды перед и после фигурных скобок.

badiou:briefings; 1—4], а также ссылок с указанием страниц, на котором отображены те или иные текстово-графические объекты [Naidenova2017] или в виде мультицитаты на несколько источников [Naidenova2017; Ganter1999]. Часть библиографических записей носит иллюстративный характер и не имеет отношения к реальной литературе.

Короткое имя каждого библиографического источника содержится в специальном файле `my_biblio.bib`, расположенном в папке `my_folder`. Там же находятся исходные данные, которые с помощью программы `Viber` и стилевого файла `Biblatex-GOST` [ctan-biblatex-gost] приведены в списке использованных источников согласно ГОСТ 7.0.5-2008. Многообразные реальные примеры исходных библиографических данных можно посмотреть по ссылке [ctan-biblatex-gost-examples].

Как правило, ВКР должна состоять из четырех глав. Оставшиеся главы можно создать по образцу первых двух и подключить с помощью команды `\input` к исходному коду ВКР. Далее в приложении ?? приведены краткие инструкции запуска исходного кода ВКР [latex-miktex; latex-texstudio].

В приложении ?? приведено подключение некоторых текстово-графических объектов. Они оформляются по приведенным ранее правилам. В качестве номера структурного элемента вместо номера главы используется «П» с номером главы. Текстово-графические объекты из приложений не учитываются в реферате.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Штейн Г. И.* Руководство по конфокальной микроскопии - СПб: ИНЦ РАН, // — Стандинформ, 2007. — (Сер.: ил.ISBN).
2. *Baharlou H. C. N. P. Bertram K. M., Sandgren K. J., Cunningham A. L., Harman A. N., Patrick E* AFid: A tool for automated identification and exclusion of autofluorescent ob-jects from microscopy images. bioRxiv, 566315 // — 2019. — URL: <https://doi.org/10.1101/566315>.
3. *Bradski G.* The OpenCV Library. — 2000. — P. 122–125.
4. *Kozlov K.N. Baumann P. Waldmann J. et al.* TeraPro, a system for pro-cessing large biomedical image. // Pattern Recognit. Image Anal. — 2013. — P. 23, 488–497. — URL: <https://doi.org/10.1134/S105466181304007X>.