Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Иванищев

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Крамаренко

Лабораторная работа №4

Методы выделения границ. Алгоритм Кэнни.

В рамках данной лабораторной работы будет рассматриваться решение задачи выявления границ объектов на изображении. Данная задача является необходимой частью более сложных и полных задач – выявление объекта на изображении, идентификация объекта, распознавание текста на фрагментах изображения, детектор движения в видеопотоке, обнаружение объекта в видеопотоке, трекинг объекта и многих других.

В данной работе контур рассматривается как совокупность пикселей, в окрестности которых наблюдается скачкообразное изменение функции яркости. Точки контура представляют собой границу объекта, отделяющую его от фона. В дальнейшем в данной работе также для обозначения контура будет использоваться понятие граница, подразумевающее границу яркости.

В данной работе под изображением понимается функция двух вещественных переменных 𝐼(𝑖,𝑗), где 𝐼 ‒ это интенсивность (яркость) в точке с координатами (𝑖,𝑗).

В данной лабораторной работе будут рассмотрены градиентные методы, в частности, комбинированный градиентный метод, являющийся наиболее часто применяемым для решения поставленной задачи – алгоритм Кэнни.

Градиентные методы основаны на достаточно простой идее, заключающейся в том, что на границе объекта происходит сильный скачок яркости изображения. Получаем, если в данном пикселе сильно меняется значение яркости, то данный пиксель является границей. Для определения количественного изменения яркости необходимо вспомнить, что изображение рассматривается, как функция двух переменных. И яркость пикселя – это функция. Тогда – скорость изменения яркости – это скорость изменения функции, то есть частные производные функции яркости, т.е. градиент данной функции.

Таким образом получим, если длина градиента яркости пикселя больше длины градиента соседей и больше некоторой пороговой величины, то данный пиксель считается границей.

Теперь перейдем к разбору градиентного комбинированного метода, который называется детектор границ Кэнни.

Алгоритм состоит из следующих шагов:

1) создание черно-белого изображения;

2) применение фильтрации для подавления шумов (размытие Гаусса);

3) вычисление градиентов функции яркости;

4) подавление немаксимумов (если значение градиента пикселя больше соседних и больше некоторой пороговой величины, то пиксель определяется как граничный, иначе значение пикселя подавляется);

5) двойная пороговая фильтрация (сравнение величины градиента с двумя пороговыми значениями).

**Задание 1** Реализовать метод, который принимает в качестве строки полный адрес файла изображения, читает изображение, переводит его в черно белый цвет и выводит его на экран применяет размытие по Гауссу и выводит полученное изображение на экран.

В данном задании мы реализуем пункт 1-2 алгоритма. Реализовано при помощи функции GaussianBlur, разобранной в прошлой лабораторной работе.

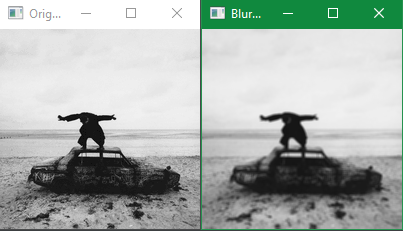


Рисунок 1 – оригинальное изображение и размытое по Гауссу.

**Задание 2.** Модифицировать построенный метод так, чтобы в результате

вычислялось и выводилось на экран две матрицы – матрица значений длин и матрица значений углов градиентов всех пикселей изображения.

При выполнении этого задания мы реализуем пункт 3 алгоритма-вычисление градиента функции яркости.

Изображение в нашем случае представляет из себя функцию яркости I(x,y), где x,y – координаты пикселя изображения. Градиентом функции двух переменных в точке 𝑥0, 𝑦0 называется вектор

∇𝐼(𝑥0, 𝑦0 ) = ( 𝜕𝐼 𝜕𝑥 (𝑥0, 𝑦0 ), 𝜕𝐼 𝜕𝑦 (𝑥0, 𝑦0 ) ),

полученный из значений частных производных в точке 𝑥0, 𝑦0, для наглядности преобразований обозначим градиент как

∇𝐼(𝑥0, 𝑦0 ) = (𝐺𝑥,𝐺𝑦)

Величиной градиента будем считать его длину, вычисляемую согласно определению длины вектора

Изображение выглядит как Шрифт, текст, белый, число

Автоматически созданное описание

Здесь возникает интересная особенность: понятие частной производной определено для непрерывной на области определения функции, в то время как рассматриваемая функция изображения определена на конечном множестве натуральных точек. Для разрешения данной ситуации необходимо рассмотреть численные методы вычисления частных производных. В алгоритме Кэнни рассматривается оператор Собеля для вычисления частных производных. Будем применять операцию свертки из прошлой лабораторной работы. Тогда оператор Собеля удобно представить в виде применения операции свертки с двумя матрицами свертки вида:

Изображение выглядит как Шрифт, белый, рукописный текст, типография

Автоматически созданное описание

После вычисления частных производных для всех внутренних пикселей изображения (все, кроме первых и последних строки и столбца) необходимо вычислить длину градиента для каждого пикселя. Алгоритм Кэнни строится на том предположении, что границей будут являться такие пиксели, градиент которых будет локальным максимумом. Для реализации этого алгоритма необходимо сравнить величину градиента пикселя с соседями и если длина градиента больше, то мы имеем дело с границей.

Но важно отметить, что величину градиента нужно сравнивать не со всеми соседними пикселями, а с теми пикселями, по направлению которых значение функции отличается сильнее всего. Для этого необходимо вычислить направление, в котором функция меняется наиболее быстро. Для этого найдем величину угла градиента по формуле (1)

Далее нам нет необходимости точно знать, какова величина угла. Нам необходимо для направления выбрать лишь один из соседних 8 пикселей. То есть необходимо округлить величину угла до 45 градусов. Для определения угла будем использовать следующую схему:

Изображение выглядит как текст, снимок экрана, Красочность, диаграмма

Автоматически созданное описание

Рисунок 2 – схема округления угла до 45 градусов.

В итоге мы имеем 2 матрицы - матрица значений длин и матрица значений углов градиентов всех пикселей изображения.

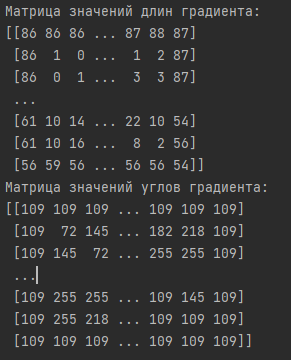


Рисунок 3 – полученные матрицы.

**Задание 3**. Модифицировать метод так, чтобы он выполнял подавление немаксимумов и выводил полученное изображение на экран. Рассмотреть изображение, сделать выводы.

Подавление немаксимумов означает следующую идею: **границей будет считаться пиксель, градиент которого максимален в сравнении с пикселями по направлению наибольшего роста функции.**

В граничные пиксели выставляем значение яркости 0 (черный), в остальные-255 (белый) или наоборот.

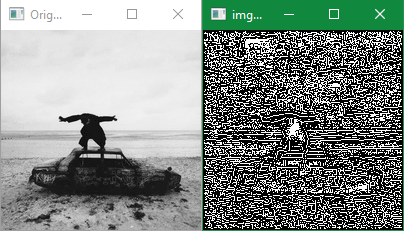


Рисунок 4 – границы изображения после подавления немаксимумов.

На данном изображении можно увидеть контуры объектов, если сильно присмотреться, но все же результат не является конечным, так как в качестве границы выбрано достаточно большое количество пикселей, не являющихся границей объекта, а лишь показывающих изменение яркости.

**Задание 4.** Модифицировать метод так, чтобы он выполнял двойную пороговую фильтрацию и выводил полученное изображение на экран.

Для того, чтобы выделить непосредственно границы, применим такой этап, как двойная пороговая фильтрация. Для этого необходимо выбрать два пороговых значения для градиента, используя значения максимального градиента. И далее производим следующую фильтрацию для пикселей, которые уже выбраны как граница на предыдущем шаге: если значение градиента меньше нижней границы, то пиксель не граница, если значение градиента выше верхней границы, то пиксель точно граница.

После такого фильтра останутся пиксели, значение градиента которых заключено между границами. Для них воспользуемся следующим предположением: если пиксель – это граница, то он не может быть отдельной границей, рядом должен быть еще пиксель с границей. Исходя из этого предположения добавим проверку на то, что рядом с границей есть другая граница, для чего необходимо проверить 8 пикселей вокруг заданного.

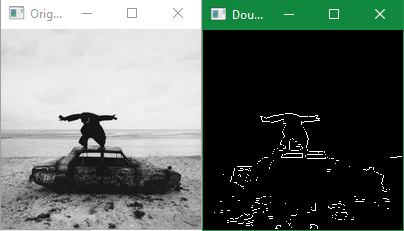


Рисунок 5 - выделенные контуры на изображении после применения двойной пороговой фильтрации.

**Задание 5.** Провести опыты для различных параметров размытия и различных пороговых значений градиента, определить наилучшие параметры для Вашего изображения.

В сравнительном анализе удалось заметить, что более низкое значение порогового значения приведет к тому, что больше пикселей могут быть отмечены как граница. Чем меньше стандартное отклонение и/или размер ядра, тем более сильное размытие будет применено к изображению. В результате, мелкие детали и шум будут сглажены, что поможет более эффективно выделять настоящие границы объектов.

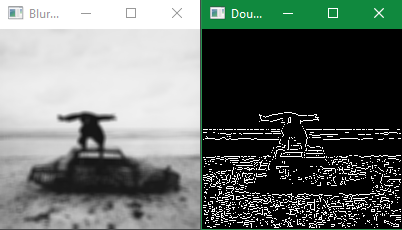


Рисунок 6 - итоговое значение с хорошо отрисованными контурами. (значение отклонения = 6, размер ядра = 5).

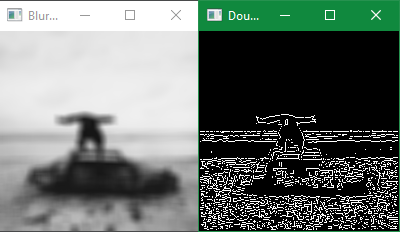


Рисунок 7 - итоговое значение с детально отрисованными контурами

(высокие значения параметров-границы прорисованы детально).

**Задание 6 (самостоятельно)** Реализовать алгоритм Канни на другом языке программирования.

Для выполнения этого задания я использовал гем ruby-gems. Результат работы встроенного алгоритма Кэнни представлен на рисунке 8.



Рисунок 8 – алгоритм Кэнни на Ruby.