

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»

38087

регистрационный номер

Компьютерные системы

название секции

**Численное определение площади поверхности, ограниченной
произвольным замкнутым контуром, с применением компьютерного
зрения**

название работы

Автор:

Цыпурский Борис Игоревич

фамилия, имя, отчество

ГБОУ Школа №1155, 10

наименование учебного заведения, класс

Научный руководитель:

Юсупова Кристина Олеговна

фамилия, имя, отчество

ГБОУ Школа №1155

место работы

учитель информатики

звание, должность

подпись научного руководителя

Численное определение площади поверхности, ограниченной произвольным замкнутым контуром, с применением компьютерного зрения

Аннотация

Площади известных геометрических фигур определяются с помощью формул, изучаемых в курсе планиметрии. Но на практике чаще возникает необходимость оценки площади некоторой плоской фигуры, которая ограничена произвольным контуром. Для решения такой задачи следует использовать методы вычислительной математики, так как теории геометрии для этого недостаточно. В качестве примера, где может быть необходимо определение площади плоской фигуры произвольной формы, может быть оценка территории, пораженной огнем во время пожара, или пострадавшей при извержении вулкана.

Одним из методов оценки площади в такой ситуации является метод Монте-Карло. В применении данного метода был обнаружен недостаток, который в работе предложено решить с помощью возможностей компьютерного зрения. Таким образом, целью работы является реализация алгоритма для численной оценки площади заданного на снимке участка местности произвольной формы методом Монте-Карло на языке программирования C++ с применением библиотеки компьютерного зрения OpenCV.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ.....	6
1 Описание алгоритма метода Монте-Карло	6
2 Программная реализация алгоритма вычисления площади для тестовых фигур.....	8
3 Тестирование разработанной программы на известных геометрических фигурах.....	11
4 Определение площади участков земной поверхности произвольной формы по снимку	16
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	19
Приложение А. Отзыв.....	20

ВВЕДЕНИЕ

При проведении землеустроительных работ или экстренной оценки ущерба при стихийных бедствиях используются различные способы вычисления площадей участков земли, самый популярный из которых – графический. Для осуществления такого способа используются данные измерений на плане, карте или снимках со специальных приборов. Например, общая площадь участка в границах координат характерных точек определяется по плану с помощью разбиения искомой площади на участки элементарного вида (треугольники, прямоугольники, трапеции) и дальнейшего суммирования площадей этих элементарных фигур. Точность такого подхода значительно ниже в случае, когда участки имеют криволинейную форму, которую трудно аппроксимировать простыми фигурами. В таких случаях специалисты применяют прозрачные палетки с нанесенными на них делениями, при наложении которых на план определяют количество делений, входящих в контур участка, и оценивают площадь участка, определив площадь одного деления с учетом масштаба. Недостаток такого метода состоит в том, что количество неполных квадратов приходится оценивать экспертно («на глаз»), в результате чего ухудшается точность метода [1].

В данной работе представлена идея другого варианта определения площади участка криволинейной формы, основанная на программном анализе снимка измеряемой территории с прибора специального назначения, для которого известны параметры съемки. С помощью численного метода и средств библиотеки OpenCV для анализа изображений был разработан алгоритм, позволяющий с хорошей точностью и с минимальными временными затратами определить площадь фигуры любой выпуклой формы. Работоспособность программы в проекте продемонстрирована на тестовой выборке, для каждого элемента которой дана оценка точности вычислений. В работе обозначены недостатки выбранного численного метода, которые были устранены с помощью авторского подхода и возможностей работы с цифровыми изображениями на C++.

В данном проекте была поставлена следующая цель: реализовать алгоритм для численной оценки площади заданного на снимке участка местности произвольной формы методом Монте-Карло на языке программирования C++ с применением библиотеки компьютерного зрения OpenCV.

Для достижения поставленной цели были выполнены следующие задачи:

1. выполнить программную реализацию алгоритма определения площади фигуры методом Монте-Карло на языке C++;
2. решить проблему определения принадлежности точки области внутри искомой фигуры с помощью средств библиотеки OpenCV;
3. устранить недостаток метода Монте-Карло, связанный со снижением точности при генерации большого количества точек со случайными координатами;
4. протестировать программу на изображениях с известными геометрическими фигурами и убедиться в работоспособности алгоритмов с помощью оценки погрешности вычислений;
5. адаптировать алгоритм под применения на реальных снимках местности с пораженной территорией, например, в результате пожара или извержения вулкана.

ОСНОВНАЯ ЧАСТЬ

1 Описание алгоритма метода Монте-Карло

В качестве ключевого метода, с помощью которого программа определяет площадь фигур, применяется метод Монте-Карло. Рассмотрим алгоритм данного метода [2].

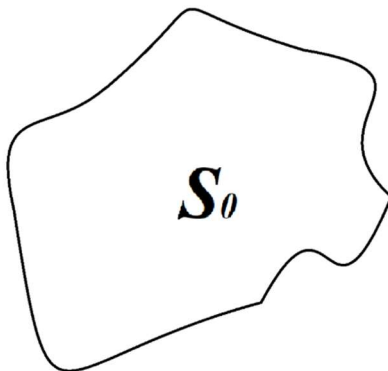
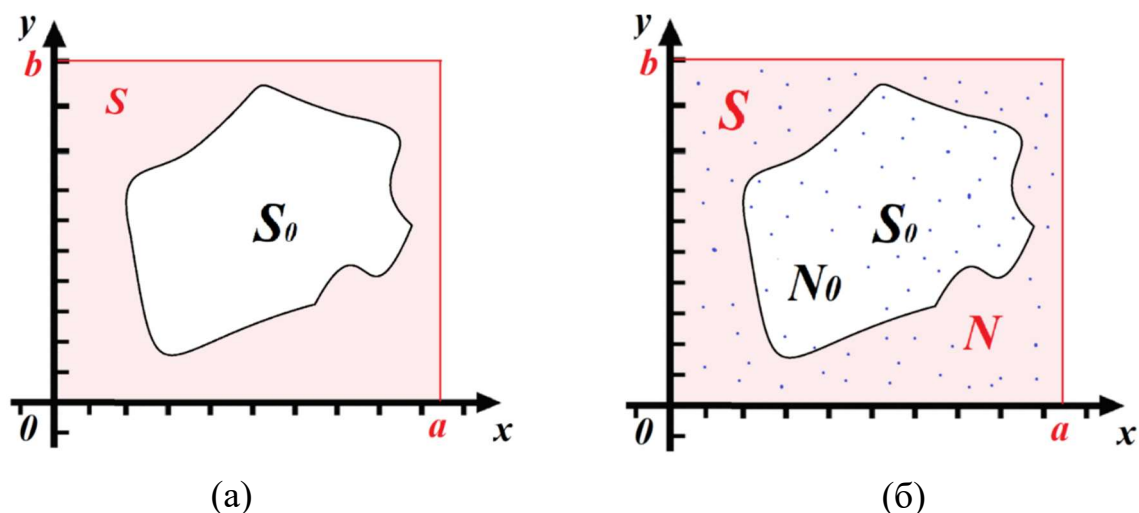


Рисунок 1 – Фигура, площадь которой необходимо определить

На рисунке 1 изображена фигура, площадь которой (S_0) необходимо определить методом Монте-Карло. Для этого выполним следующие действия:

- заключаем фигуру, площадь которой необходимо найти, внутри произвольной фигуры известной площади S (рис. 2а);
- равномерно разбрасываем N точек со случайными координатами внутри фигуры площади S . Определяем количество точек, попавших внутрь фигуры площади S_0 . Обозначим это количество N_0 (рис. 2б).



а – «Заключение в фигуру известной площади», б – «Равномерный разброс точек»

Рисунок 2 – Первые шаги метода Монте-Карло

Для площадей фигур справедливо соотношение (1), где S_0 – искомая площадь фигуры, S – площадь описанной фигуры, N – общее количество точек, N_0 – количество точек, попавших в искомую фигуру. Формула соотношения имеет следующий вид

$$\frac{S_0}{S} \approx \frac{N_0}{N}. \quad (1)$$

Из предыдущего выражения получим формулу для приближенного вычисления площади произвольной фигуры S_0

$$S_0 \approx S \cdot \frac{N_0}{N}. \quad (2)$$

Далее именно на этом методе будет выстраиваться вычисление площади фигур произвольной формы. В процессе создания вычислительного инструмента были обнаружены недостатки данного метода такие как: сложность определения критерия принадлежности точки внутренней области фигуры и подбор оптимального значения количества генерируемых точек, при котором точность вычисления площади будет минимальна. Но решение данных проблем было выведено, что позволило получить достаточно быстрый и точный программный комплекс для решения задач по определению площади участков земной поверхности.

2 Программная реализация алгоритма вычисления площади для тестовых фигур

Рассмотрим полную структурную схему алгоритма, который лежит в основе вычислительного инструмента для определения площади тестовых фигур, то есть геометрических фигур, точное значение площади которых нам известно (рис. 3). Данный алгоритм служит основой для программы, которая работает со снимками необходимой местности, сделанными на специализированное устройство, параметры съемки которого учтены в этой программе.



Рисунок 3 – Структурная схема программы

Рассмотрим реализацию каждого блока схемы нового метода, когда в качестве исходного изображения подаются снимки известных геометрических фигур. На рисунке 4 приведен код главной функции программы, где происходит загрузка изображения, а далее его обработка с помощью вспомогательных подпрограмм.


```

signed main() {
    setlocale(LC_ALL, "Rus");
    string path = "C:/Users/Борис/Downloads/прямоуг.png";
    Mat image = imread(path, IMREAD_UNCHANGED);
    Mat imgResize;
    if (image.empty()) {
        cout << "Could not open or find the image" << '\n';
        cin.get();
    }

    cvtColor(image, imgGray, COLOR_BGR2GRAY);
    GaussianBlur(imgGray, imgBlur, Size(3, 3), 3, 0);
    Canny(imgBlur, imgCanny, 25, 75);
    Mat kernel = getStructuringElement(MORPH_RECT, Size(3, 3));
    dilate(imgCanny, imgDil, kernel);
    getContours(imgGray, image);
    resize(image, imgResize, Size(877, 551));
    cout << karlo(image) - P * 2 << '\n';
    cout << image1.size();
    bool o = imwrite("C:/Users/Борис/Downloads/2_4(5).png", image);
    imshow("img", imgResize);
    waitKey(0);
    return 0;
}

```

Рисунок 4 – Главная программа

В первую очередь, изображение анализируется с помощью средств библиотеки OpenCV на обнаружение контура, соответствующего определенным цветовым диапазонам. После того, как программа выделит контур, производится заливка внутренней области изображения в пределах контура (рис. 5). Это необходимо для реализации точной проверки принадлежности точки внутренней части фигуры в методе Монте-Карло для вычисления значения площади.

```

void getContours(Mat imgGray, Mat image) {
    Mat thresh1;
    Mat imgResize1;
    threshold(imgGray, thresh1, 150, 255, THRESH_BINARY);
    vector<vector<Point>> contours2;
    vector<Vec4i> hierarchy2;
    findContours(thresh1, contours2, hierarchy2, RETR_TREE, CHAIN_APPROX_NONE);
    //Mat image_copy2 = image1.clone();
    drawContours(image1, contours2, -1, Scalar(0, 255, 0), FILLED);
    resize(image1, imgResize1, Size(877, 451));
    imshow("None approximation", imgResize1);
    waitKey(0);
}

```

Рисунок 5 – Функция определения контура и заливки внутренней области

```

double karlo(Mat image) {
    double square = 0;
    int s = 1000000;
    double x, y;
    int cnt = 0;
    for (int i = 0; i < s; ++i) {
        uniform_real_distribution<double> distX(0, image.rows);
        uniform_real_distribution<double> distY(0, image.cols);
        x = distX(gen);
        y = distY(gen);
        d.emplace_back(x, y);
    }
    function1();
    for (int i = 0; i < d.size(); ++i) {
        if (cvet(image, d[i].first, d[i].second)) {
            ++cnt;
        }
    }
    square = cnt / (double)d.size() * image.cols * image.rows;
    return square;
}

```

Рисунок 6 – Функция генерации точек со случайными координатами

После того, как область искомой фигуры определена, выполняется алгоритм метода Монте-Карло. Сначала происходит генерация случайных точек на изображении (рис. 6), проверка цвета каждой точки (рис. 7) и после определения точек внутри заливки на предыдущем шаге области выполняется оценка площади фигуры. Отметим, что в качестве фигуры известной площади выступает вся область самого исходного изображения, где и «разбрасываются» заданное количество точек N случайным образом.

```

bool cvet(Mat image, double x, double y) {
    Vec3b color = image.at<Vec3b>(int(x), int(y));
    if (color[0] == 0 && color[1] == 255 && color[2] == 0) {
        return true;
    }
    else {
        return false;
    }
}

```

Рисунок 7 – Проверка цвета пикселя с заданными координатами

Стоит отметить, что при генерации большого количества точек на всем изображении несколько точек могут совпасть друг с другом. Этот эффект

неблагоприятно сказывается на точность метода Монте-Карло. Поэтому в работе для таких случаев было реализовано удаление дубликатов точек после генерации. Такой подход при тестировании показал свою эффективность.

3 Тестирование разработанной программы на известных геометрических фигурах

Для того, чтобы убедиться в работоспособности программы, в том, что результатам вычисления данного инструмента можно доверять на реальных задачах, необходимо произвести тестирование программы. Для тестирования были выбраны png-файлы с белым фоном, на которых изображены контрастные геометрические фигуры с известной точной площадью. На рисунке 8 представлен пример тестовых изображений.

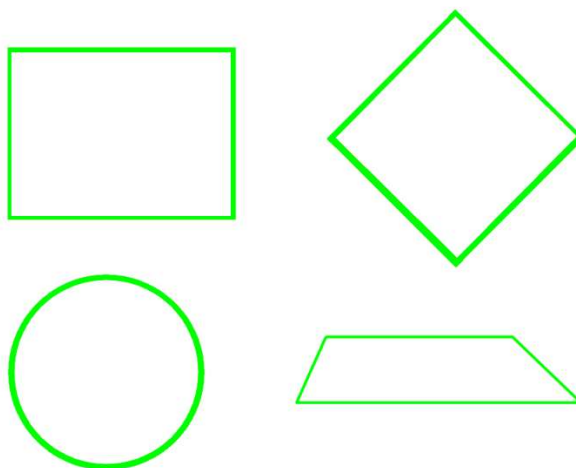


Рисунок 8 – Пример элементов тестовой выборки

В алгоритме была реализована авторская идея по определению принадлежности точки внутренней части фигуры: после распознавания контура фигуры, ее внутреннюю часть программа заливает цветом (рис. 9), и если точка (пиксель) с заданными координатами имеет цвет той самой заливки, то точка находится внутри фигуры.

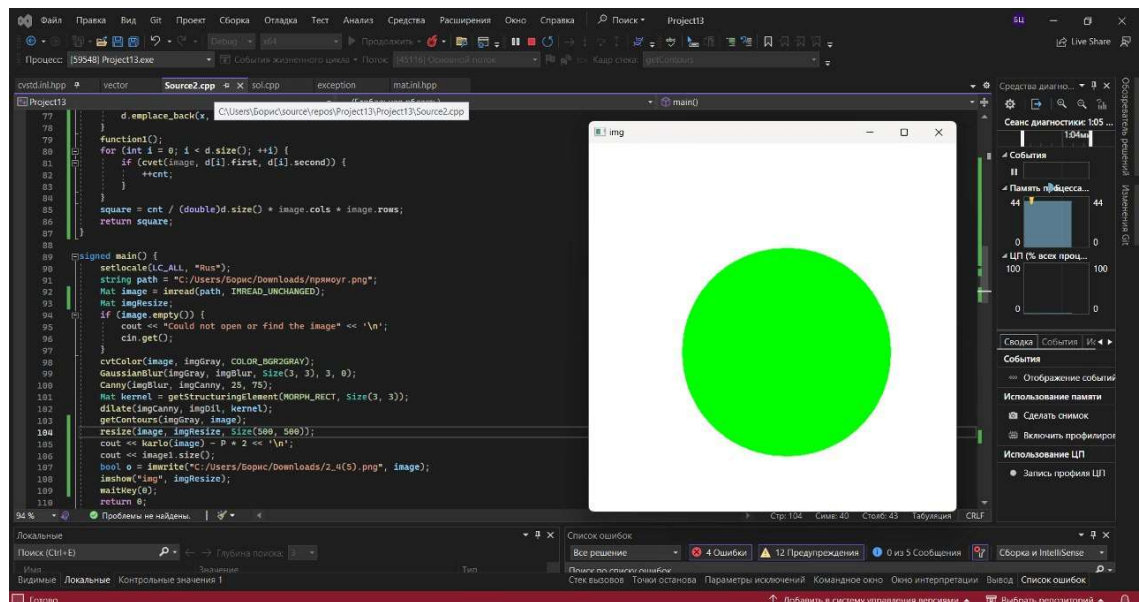


Рисунок 9 – Пример программной заливки

После численного определения площади с помощью созданного алгоритма была составлена таблица погрешностей вычисления в зависимости от различных значений количества генерируемых в методе Монте-Карло точек (таблица 1 – 3). Точность метода, действительно зависит от количества точек со случайными координатами: оно должно быть соизмеримо с размером изображения. Проиллюстрируем, как происходит равномерное распределение точек по области изображения, на рисунке 10.

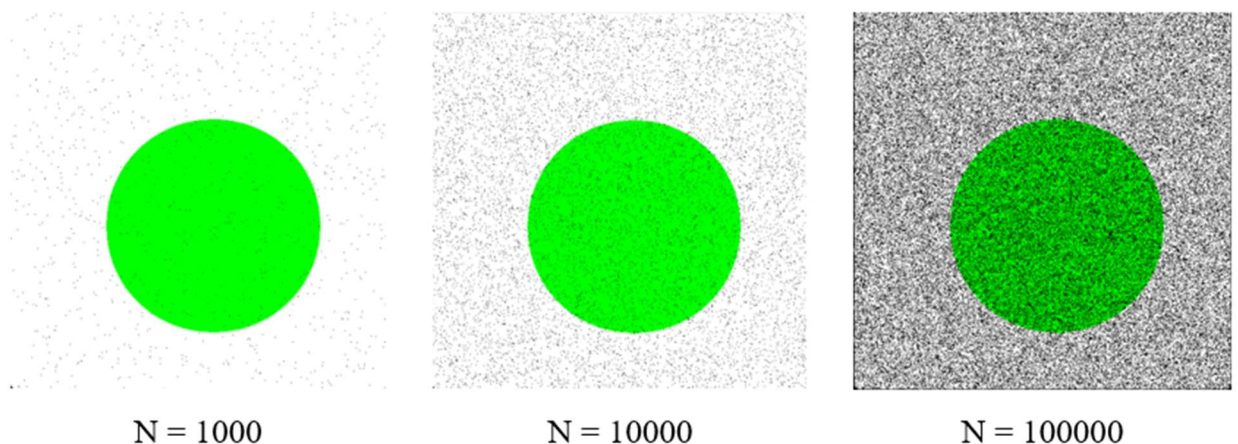


Рисунок 10 – Распределение сгенерированных случайным образом точек

Из рисунка 10 видно, что малое значение количества точек, приведет к большой ошибке вычисления за счет того, что неплотно область изображения покрыта множеством этих точек. Но также известно, что и очень большое значение количества точек приведет к их многократному наложению на области, что также сразу отражается на точности определения площади. В проекте было решено применить единый подход к решению данного недостатка в методе Монте-Карло: обеспечить в программе генерацию большого количества точек со случайными координатами, но из полученного множества точек исключить совпадающие точки. Данный подход позволил избежать рост погрешности вычисления с увеличением количества точек на этапе генерации.

Таблица 1 — Оценка точности вычислений программы для некоторых фигур при количестве генерируемых точек $N = 1000$

	Прямоугольник	Ромб	Окружность	Трапеция
Точное значение площади внутри контура S , пкс	80196	69288	62426,4	33271
Оценка площади методом Монте-Карло S' , пкс	81600.3	67198.1	63812	33808
Абсолютная погрешность, $ S - S' $	1404.3	401.9	1385.6	537
Относительная погрешность, %	1.7	0.6	2.2	1.6

Таблица 2 — Оценка точности вычислений программы для некоторых фигур
при количестве генерируемых точек $N = 1000$

	Прямоугольник	Ромб	Окружность	Трапеция
Точное значение площади внутри контура S , пкс	80196	69288	62426,4	33271
Оценка площади методом Монте-Карло S' , пкс	80550.3	67973.1	62337	33008
Абсолютная погрешность, $ S - S' $	354.3	373.1	89,4	263
Относительная погрешность, %	0.4	0.55	0.1	0.8

Из представленных таблиц можно сделать вывод, что при увеличении количества точек со случайными координатами N , которые «разбрасываются» случайным образом по всему изображению, погрешность определения площади фигуры на изображении становится меньше. Но, если данное значение N увеличивать и дальше, то с какого-то момента анализ точности вычислений будет отражать противоположный эффект – погрешность определения площади фигуры будет расти. Такая закономерность связана с многократным совпадением многих точек друг с другом, что в несколько раз искусственно увеличивает значение площади: наложенные друг на друга точки с одинаковыми координатами создают эффект нескольких фигур на одном изображении. Поэтому в программе был реализован контроль за удалением дубликатов точек при генерации. В таблице 3 приведено сравнение точности вычислений до применения этого способа и после.

Таблица 3 — Оценка точности вычислений программы для некоторых фигур
при количестве генерируемых точек $N = 1000000$

	Прямоугольник	Ромб	Окружность	Трапеция
Точное значение площади внутри контура S , пкс	80196	69288	62426,4	33271
С присутствием совпадающих точек				
Оценка площади методом Монте-Карло S' , пкс	80833.1	69287.1	62508	33938.4
Абсолютная погрешность, $ S - S' $	637.1	1687	81.57	667,4
Относительная погрешность, %	0.79	2.5	0.13	2
С учетом программного исключения совпадающих точек				
Оценка площади методом Монте-Карло S' , пкс	80555.3	69200	62505.6	33502
Абсолютная погрешность, $ S - S' $	359.3	88	79.6	231
Относительная погрешность, %	0.4	0.1	0.1	0.7

На этапе тестирования мы убедились, что вычисления происходят с достаточной точностью, чтобы применить этот инструмент в отношении снимков реальных объектов.

4 Определение площади участков земной поверхности произвольной формы по снимку

После того, как работоспособность нового алгоритма оценки площади фигуры подтверждена на примерах и соответствующих результатах вычисления погрешности, можно показать, как работает данный инструмент с изображением на снимке. На рисунке 11 представлен пример исходного снимка со спутника, на котором видна траектория распространения потока лавы. Именно в таких ситуациях необходима мгновенная автоматическая оценка площади пораженной территории с хорошей точностью. Необходимую скорость и точность как раз может обеспечить предложенный в работе алгоритм. Рассмотрим этапы обработки программой исходного изображения.



Рисунок 11 – Исходный снимок извержения вулкана

Первым шагом программа генерирует точки со случайными координатами по всему снимку. Далее идет проверка принадлежности цвета пикселя заданному диапазону для определения количества точек, попавших внутрь искомой области. Стоит отметить, что данный алгоритм настраивается под работу с определенным устройством съемки (спутник, дрон), то есть настройки обработки палитры устанавливаются в алгоритме. Кроме того, оценка площади участка изображения происходит в пикселях, поэтому для оценки площади реальной территории необходимо учесть масштаб съемки, что также определяется исходя из настройки съемки с высоты. Для лучшего визуального

представления необходимая область закрашивается однородным цветом (рис. 12). После чего производится оценка площади выделенной области методом Монте-Карло с использованием фильтрации совпадающих точек. Стоит отметить, что пиксели, попавшие в заданный цветовой диапазон, но не относящиеся к основной фигуре, алгоритмом не учитываются при вычислении площади. Таким образом, мы получили универсальный инструмент по вычислению площади, который позволяет его применять к различным системам фиксации изображений с высоты путем адаптации алгоритма к параметрам заданного устройства.

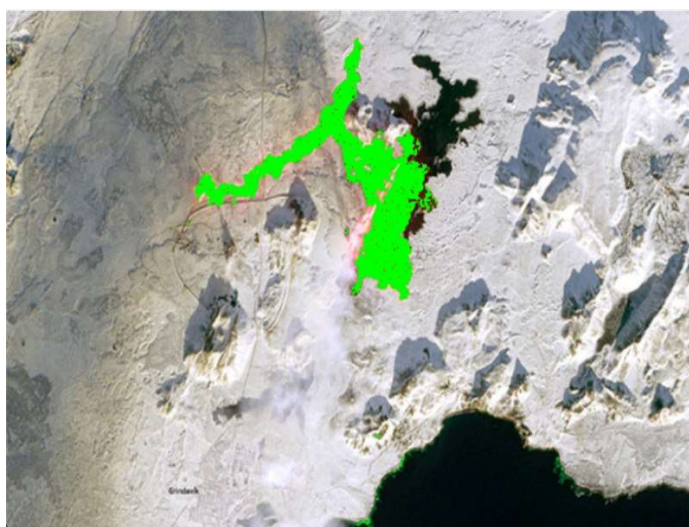


Рисунок 12 – Выделение цветом искомой части изображения на снимке

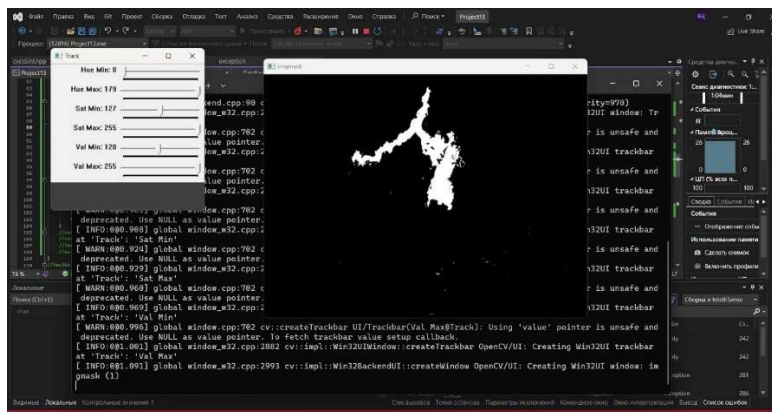


Рисунок 13 – Настройка параметров цветового формата изображения

На рисунке 13 продемонстрирован этап настройки параметров цветового формата для снимков с заданного устройства. В данный момент идет работа над автоматической подборкой значений интервалов.

ЗАКЛЮЧЕНИЕ

Проделанные в данном проекте исследования позволили нам получить быстрый и достаточно точный вычислительный инструмент, который способен решить важную задачу: оценить площадь фигуры произвольной формы, что часто встречается на практике, но так мало имеет эффективных путей решения.

Реализация алгоритма производилась с помощью языка программирования C++ с применением библиотеки компьютерного зрения OpenCV, реализация проекта происходила в среде разработки Visual Studio. Полученная программа была протестирована на специальной выборке: результаты тестирования подтвердили обоснованность использования в основе алгоритма метода вычислительной математики для определения площади фигуры – метода Монте-Карло. В ходе работы были выявлены недостатки метода, а также разработаны пути устранения этих недостатков, что позволило добиться высокой точности вычислений с его помощью.

Среди главных достоинств именно метода Монте-Карло стоит назвать также возможность его применения, когда контур фигуры произвольный и определен лишь визуально. В сочетании с компьютерным зрением метод работает с графическим представлением искомой фигуры. В то время как, например, определение площади с помощью интеграла требует математического описания контура – функции, описывающей границы фигуры. Такой подход был бы намного затратнее по времени и памяти и был бы нерационален для экстренных ситуаций, когда запуск процесса оценки площади территории должен быть незамедлительным и выдать результат почти мгновенно.

Перспективным направлением развития проекта является расширение границ его применения: адаптация программы к конкретному средству съемки и тестирование алгоритма экспериментальным образом с учетом масштаба съемки.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Курс лекций по геодезии [Электронный ресурс] URL: <https://studfile.net/preview/4295364/> (дата обращения: 15.09.2023)
2. Библиотека МЭШ. Сценарий урока 2199146 [Электронный ресурс] URL: <https://uchebnik.mos.ru/> (дата обращения: 1.12.2023)

Приложение А. Отзыв



Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский
университет «МЭИ» (ФГБОУ «МЭИ»)
111250, г. Москва,
вн.тер.г. муниципальный округ Лефортово,
ул. Красноказарменная, д. 14, стр. 1
Тел.: (495) 362-75-60, факс: (495) 362-89-38
E-mail: universe@mpei.ac.ru
<http://www.mpei.ac.ru>

№ _____

«__» _____ 20__ г.

ОТЗЫВ

на проектную работу
(проектную / исследовательскую)

по теме «Численное определение площади поверхности, ограниченной произвольным
замкнутым контуром, с применением компьютерного зрения»
(наименование работы)

учащегося 10 класса ГБОУ Школа №1155
(класс) (наименование образовательной организации)
Цыпурский Борис Игоревич
(фамилия, имя, отчество (при наличии), заполняется на каждого участника)

В проекте «Численное определение площади поверхности, ограниченной произвольным замкнутым контуром, с применением компьютерного зрения» автором обозначена проблема определения площади фигуры произвольной формы. Автор привел подробное обоснование актуальности данной проблемы: в работе продемонстрированы ситуации, когда возникает необходимость решить подобную задачу в жизни.

В рамках проекта предложен новый системный подход по определению площадей поверхностей произвольной формы по снимку. Разработанный алгоритм подробно описан в проекте, а его работоспособность подтверждена на множестве тестовых примеров. Кроме того, автором произведен анализ точности вычислений и сравнение с другими методами, в следствие чего явно отмечено преимущество нового метода.

Цели работы автором достигнуты, актуальность представленных в работе достижений и их практическая значимость подтверждены. Созданный в проекте программный комплекс с заложенным в него новым алгоритмом определения площади произвольных фигур по снимку может быть адаптирован под выбранное устройство съемки, например, с целью своевременной оценки площади территории, пораженной от лесных пожаров.

доцент
(должность)

(подпись)



Лямасов А.К.
(И.О. Фамилия)

М. П.