Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Ижевский государственный технический университет имени

М. Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе № 2

по дисциплине

«Конструирование программного обеспечения»

Выполнили

ст. гр. Б08-191-1,2 Максимов А.Н.

Торхов Д.И.

Принял

Ижевск

2016

ТРЕБОВАНИЯ

Лабораторная работа №2

Подсистемы:

Схема системы

Содержание

Перечень рисунков

Перечень таблиц

1.ВВЕДЕНИЕ

1.1 Назначение системы

1.2 Область применения системы

1.3 Определения, акронимы, аббревиатуры

1.4 Ссылки

1.5 Обзор системы

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

2.6 Характеристики пользователя

2.7 Допущения и зависимости

2.8 Оперативные сценарии

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

1.ВВЕДЕНИЕ

1.1 Назначение системы

Система предназначена для тестирования и демонстрации алгоритмов аэронавигации БПЛА.

1.2 Область применения системы

Система применяется на персональных компьютерах под управлением операционной системой семейства Windows.

1.3 Определения, акронимы, аббревиатуры

БПЛА - беспилотный летательный аппарат.

1.4 Ссылки

1.5 Обзор системы

Система представляет собой приложение, работающее на компьютерах под управлением операционной системой семейства Windows, которое демонстрирует работу некоторых алгоритмов аэронавигации по снимкам, полученных с камеры БПЛА. Для демонстрации пользователь выбирает параметры обработки снимков, снимки, подтверждает операцию, после чего получает результат.

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

- выбор файлов обработки

- выбор параметров обработки

- обработка изображений

- демонстрация результата

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

Система работает с только с изображениями формата bmp.

2.6 Характеристики пользователя

Пользователь - конечный пользователь системы. Использует систему для тестирования или демонстрации алгоритмов аэронавигации.

2.7 Допущения и зависимости

В некоторых случаях система может давать неверный результат из-за различных факторов таких как поворот БПЛА или схожесть объектов на снимке.

2.8 Оперативные сценарии

Система предполагает следующий сценарий: пользователь выбирает изображения для обработки, параметры обработки, после чего система обрабатывает выбранные изображения и демонстрирует результат пользователю.

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

Система находится на персональном компьютере пользователя.

3.1.2 Износостойкость

3.1.3 Адаптируемость

При какой либо ошибке обработки изображения система информирует пользователя, прерывает текущую обработку и продолжает работу в нормальном режиме.

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

Система не является полностью надежной, т.к. возможны невереные результаты.

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

Обработчик изображений:

interface IBitmapEditor

{

int[,] GetColorHistogram();

void Contrast(double threshold);

void Filter(int dimension);

List<MapObject> GetMapObjects(int minuendDimension, int subtrahendDimension, double threshold);

Invariants GetInvariants(Rectangle where);

}

Детектор контуров:

interface IContourDetector

{

Bitmap Gradient(Bitmap source, out Bitmap gradientMagnitude, out Bitmap gradientAngle, out Bitmap gradientDirection);

Bitmap GetContours(Bitmap image, ContoursAlgorithmParams param);

}

Объект на карте:

interface IMapObject

{

List<Point> Points { get; }

Point Middlepoint { get; }

MapObjectType Type { get; }

double GetDistanceTo(MapObject mapObject);

double GetDistanceTo(Point point);

double GetAngleTo(MapObject mapObject);

double GetAngleTo(Point point);

}

Описание особенностей точек:

interface IPointDescriptor<T>

{

double [] Values { get; }

double GetDistinctWith(T other);

}

Базовая обработка изображений:

interface IPreProcessor

{

Bitmap EqualiseHistogram(Bitmap source);

Bitmap GaussFilter(Bitmap source, int windowSize);

Bitmap LinearFilter(Bitmap source, int windowSize);

Bitmap MedianFilter(Bitmap source, int windowSize);

Bitmap StretchHistogram(Bitmap source, double lowThreshold, double highThreshold);

Bitmap ToBlackAndWhite(Bitmap source);

}