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 Власов В.Г.

Ижевск

2016

ТРЕБОВАНИЯ

1. Пользовательский интерфейс
   1. Выбор изображений для обработки
   2. Задание параметров алгоритмов обработки изображений
   3. Просмотр промежуточных и окончательных результатов обработки
2. Предварительная обработка изображений
   1. Повышение контрастности изображения
   2. Фильтрация изображения
   3. Преобразование изображения в оттенки серого
   4. Нахождение разности двух изображений
3. Обработка особых точек
   1. Инварианты Ху в качестве дескрипторов точек
   2. Сопоставление последовательных кадров в пространстве дескрипторов
   3. Нахождение смещения между кадрами
   4. Построение траектории движения летательного аппарата на основе смещений между последовательными картами
4. Выделение контуров объектов на изображении
   1. Вычисление градиента изображения
   2. Построение контуров
   3. Описание контуров с помощью цепных кодов
   4. Построение топологического графа на основе цепных кодов
   5. Определение текущего положения на спутниковых снимках местности на основе сопоставления контуров объектов
5. Хранение данных и параметров
   1. Сохранение последнего использовавшегося списка параметров работы алгоритмов
   2. Сохранение промежуточных результатов обработки

Лабораторная работа №2

Подсистемы:

1. Подсистема интерфейса пользователя
   1. Модуль загрузки изображений для обработки
   2. Модуль задания параметров алгоритмов
   3. Модуль просмотра результатов обработки
2. Подсистема обработки изображений
   1. Модуль предварительной обработки
   2. Модуль выделения контуров
   3. Модуль выделения особых точек
3. Подсистема хранения данных
   1. Модуль хранения параметров программы
   2. Модуль хранения результатов обработки

Схема системы

AirNavSystem

Хранение параметров алгоритмов

IDataStorage

<Algorithm

Params>

IDataStorage

<Bitmap>

Хранение результатов обработки

Хранилище данных

IDataStorage

<T>

IPoint

Descriptor

Обработчик особых точек

Предварительная обработка

IImage

PreProcessor

Детектор контуров

IContour

Detector

Обработчик изображений

IImage

Processor

Просмотр результатов обработки

Выбор параметров алгоритмов

Выбор изображений для обработки

Интерфейс пользователя

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

Предварительная обработка

Обработчик изображений

Просмотр результатов обработки

Выбор параметров алгоритмов

Выбор изображений для обработки

Интерфейс пользователя

AirNavSystem

Построение траектории движения

Сопоставление особых точек на соседних кадрах

Сопоставление выделенных контуров с эталонными

Описание контуров

Выделение особых точек

Выделение контуров

Хранение параметров алгоритмов

Хранение результатов обработки

Хранилище данных

Обработчик особых точек

Детектор контуров

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 неверные результаты.

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

Детектор контуров:

interface IContourDetector

{

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

Bitmap GetContours(Bitmap image, ContoursAlgorithmParams param);

ChainCode[] DescribeContours(Bitmap Contours);

TopologicalGraph BuildGraphFromChainCode(ChainCode chainCode);

}

Сохранение данных на диск:

interface IDataStorage<T>

{

T[] Load();

void Store(params T[] objects);

}

Базовая обработка изображений:

interface IImagePreProcessor

{

int[,] GetColorHistogram(Bitmap source);

Bitmap Contrast(Bitmap source, double threshold);

Bitmap Filter(Bitmap source, int dimension);

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);

}

Нахождение траетории и местоположения БПЛА:

public interface IImageProcessor

{

Point GetCurrentLocation(ContoursAlgorithmParams param, Bitmap image, Bitmap areaMap);

Point[] GetTrajectory(PointsAlgorithmParams param, params Bitmap[] images);

}

Объект на изображении:

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 точек:

interface IPointDescriptor<T>

{

double [] Values { get; }

double GetDistinctWith(T other);

}

Детектор точек:

interface IPointsDetector

{

Invariants GetInvariants(Rectangle where);

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

}