

Lucas Kanade

1.4-19-gb49fe34

Создано системой Doxygen 1.8.9.1

Вт 16 Июнь 2015 14:53:06

Содержание

1	Lucas–Kanade	1
2	Алфавитный указатель структур данных	2
2.1	Структуры данных	2
3	Список файлов	2
3.1	Файлы	2
4	Структуры данных	3
4.1	Структура <code>imageInform</code>	3
4.1.1	Подробное описание	3
4.2	Структура <code>subSize</code>	3
4.2.1	Подробное описание	4
5	Файлы	4
5.1	Файл <code>src/global.h</code>	4
5.1.1	Подробное описание	4
5.2	Файл <code>src/lk_func.h</code>	4
5.2.1	Подробное описание	5
5.2.2	Функции	6
5.3	Файл <code>src/lk_interpolation.h</code>	9
5.3.1	Подробное описание	9
5.3.2	Функции	9
5.4	Файл <code>src/lk_math.h</code>	10
5.4.1	Подробное описание	11
5.4.2	Функции	11
5.5	Файл <code>src/lk_struct.h</code>	11
5.5.1	Подробное описание	11

1 Lucas–Kanade

Целью настоящей работы является разработка программного обеспечения (ПО) для оценки деформаций поверхностей твёрдых тел, а также проведение исследований алгоритмов и методов как на модельных, так и на реальных оптических изображениях.

В работе исследовано влияние метода интерполяции изображений с субпиксельной точностью с использованием итеративного подхода на расчёт оптического потока(векторного поля).

Проект выполнен с использованием следующих средств разработки: языка программирования C++(Qt), среды разработки QtCreator 3, Sublime 3. Система контроля версий git.

Версия

1.4

Установка

Для начала, необходимо настроить окружение: ## Qt

```
1 $ sudo apt-get install qt-sdk libarchive-dev libqt5webkit5-dev
```

CMAKE

```
1 $ sudo apt-get install cmake
```

HDF5 Software

```
1 $ wget http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.8.14.tar
2 $ tar -xvf hdf5-1.8.14.tar
3 $ cd hdf5-1.8.14
4 $ cmake CMakeLists.txt
5 $ make
6 $ sudo make install
```

DV(Deformation analys)

```
1 $ cp /media/flash_card/dv.zip
2 $ unzip dv.zip
3 $ cd dv/df-cl
4 $ cmake CMakeLists.txt
5 $ make
```

Lucas-Kanade

```
1 $ git clone git@github.com:IgorPolyakov/graduate-work.git
2 $ cd graduate-work
3 $ cmake CMakeLists.txt
4 $ make
```

Запуск

```
1 $ ./bin/lucas_kanade -l <list_of_image>
```

Опции

- l — load list image
- o — output directory
- i — count iteration (1 by default)
- w — size window search (3px by default)
- g — step for grid (5px by default)
- v — show version
- h — show help
- d — debug mod on

Список дел

- Добавить юнит-тесты

Лицензия

GNU GPL

2 Алфавитный указатель структур данных

2.1 Структуры данных

Структуры данных с их кратким описанием.

imageInform	The imageInform struct Хранение информации о размерах изображения	3
subSize	The SubSize struct Данный файл содержит в себе определения основных структур, используемых в программе	3

3 Список файлов

3.1 Файлы

Полный список документированных файлов.

src/global.h	Заголовочный файл с описанием глобальных переменных	4
src/lk_func.h	Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе	4
src/lk_interpolation.h	Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполирование данных	9
src/lk_math.h	Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за подсчет статистических показателей	10
src/lk_struct.h	Заголовочный файл с описанием используемых структур	11
src/version.h		??

4 Структуры данных

4.1 Структура imageInform

The [imageInform](#) struct Хранение информации о размерах изображения

```
#include <lk_struct.h>
```

Поля данных

- int [height](#)
Размер изображения по высоте
- int [width](#)
Размер изображения по ширине

4.1.1 Подробное описание

The `imageInform` struct Хранение информации о размерах изображения

Объявления и описания членов структуры находятся в файле:

- `src/lk_struct.h`

4.2 Структура `subSize`

The `SubSize` struct Данный файл содержит в себе определения основных структур, используемых в программе

```
#include <lk_struct.h>
```

Поля данных

- `int x_1`
x_1 Координата пикселя по x для левого изображения
- `int y_1`
y_1 Координата пикселя по y для левого изображения
- `int rc`
rc Радиус окна, т.е. если `radiusCode = 2`, то сетка 5x5
- `int step`
step - Шаг между векторами
- `int x_2`
x_2 Координата пикселя по x для правого изображения
- `int y_2`
y_2 Координата пикселя по y для правого изображения

4.2.1 Подробное описание

The `SubSize` struct Данный файл содержит в себе определения основных структур, используемых в программе

Объявления и описания членов структуры находятся в файле:

- `src/lk_struct.h`

5 Файлы

5.1 Файл `src/global.h`

Заголовочный файл с описанием глобальных переменных

```
#include <QString>
```

Переменные

- `bool g_isDebug`
g_isDebug Включить режим вывода в консоль отладочной информации
- `int g_sizeWindowSearch`
g_sizeWindowSearch Размер окна поиска

- int [g_stepForGrid](#)
g_stepForGrid Размер шага сетки (через какое расстояние повторяются точки поиска)
- int [g_iteration](#)
g_iteration Число итераций (уточнений)
- int [g_interpolation](#)
g_interpolation Выбор метода интерполирования (0 - Б-сплайн, 1 - Билинейный, 3 - Бикубический)
- QString [g_outputFolder](#)
g_outputFolder Директория для сохранения результатов работы
- double [g_fastProgBar](#)
g_fastProgBar Переменная для отображения текущего прогресса исполнения программы
- double [g_slowProgBar](#)
g_slowProgBar Переменная для отображения общего прогресса исполнения программы

5.1.1 Подробное описание

Заголовочный файл с описанием глобальных переменных

Данный файл содержит в себе определения глобальных переменных, используемых в программе

5.2 Файл src/lk_func.h

Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе

```
#include "dv.h"
#include "global.h"
#include "lk_interpolation.h"
#include "lk_struct.h"
#include "lk_math.h"
#include <deprecated/dvfile.h>
#include <iostream>
#include <math.h>
#include <QDebug>
#include <QFile>
#include <vector>
#include "hdf5io.h"
```

Функции

- Data2Db * [resizeImage](#) (Data2Db *image, int kK, QString prefix)
resizeImage Функция масштабирования изображения, для построения пирамиды уменьшенных изображений
- double * [multiplicMtrxAndVectr](#) (double **array, int *vector)
multiplicMtrxAndVectr Произведение матрицы на вектор
- int [calcLvlPyramid](#) (int cx, int cy, bool isPyramid)
calcLvlPyramid Автоматическое вычисление количества уровней пирамиды на основе размеров изображения
- std::vector< Data2Db * > * [createPyramid_v2](#) (Data2Db *img, int lvl_pyramid, QString pref)
createPyramid_v2 Выделение памяти для пирамиды изображений
- Vec2d [computeOptFlow](#) ([subSize](#) *kernel, Data2Db *leftImg, Data2Db *rightImg, Vec2d &dv, double *d_mid_x, double *d_mid_y, double *d_max_x, double *d_min_x, double *d_min_y, double *d_max_y, double *d_avg_x, double *d_avg_y)
computeOptFlow Вычисление вектора оптического потока

- VF2d * `computeGrid` (Data2Db *leftImg, Data2Db *rightImg, VF2d *prev, QString info)
computeGrid Строит сетку с ранее заданным шагом, в точках пересечения ищется вектор оптического потока.
- void `inversion` (Matx22d &A)
inversion Нахождения обратной матрицы(2 x 2).
- void `printProgressBar` (double fastProgBar, double slowProgBar)
printProgressBar Функция ответственная за отрисовку прогресс бара при использовании графического интерфейса
- void `saveVfResult` (VF2d &vf, QString info)
saveVfResult Сохранение векторного поля в формате VF. Просмотр возможен в программе dfcl
- void `derivativeVectorField` (VF2d &vf, QString info)
derivativeVectorField Вычисление поля деформации
- void `writeHdf5File` (QString path, ProtoData2D &layer, bool writeMode)
writeHdf5File Сохранение всех результатов в формат HDF5F

5.2.1 Подробное описание

Заголовочный файл с описанием функций. Данный файл содержит в себе определения основных функций, используемых в программе

5.2.2 Функции

5.2.2.1 int calcLvlPyramid (int cx, int cy, bool isPyramid)

calcLvlPyramid Автоматическое вычисление количества уровней пирамиды на основе размеров изображения

Аргументы

in	cx	Ширина изображения
in	cy	Высота изображения
in	isPyramid	Вычислять ли уровень пирамиды

Возвращает

[out] Количество уровней пирамиды

5.2.2.2 VF2d* computeGrid (Data2Db * leftImg, Data2Db * rightImg, VF2d * prev, QString info)

computeGrid Строит сетку с ранее заданным шагом, в точках пересечения ищется вектор оптического потока.

Аргументы

in	leftImg	указатель на массив яркостей первого кадра
in	rightImg	указатель на массив яркостей второго кадра
in	prev	указатель на векторное поле содержащий предыдущий уровень пирамиды
in	info	имя слоя содержащего векторное поле смещений

Возвращает

[out] двумерный массив содержащий векторное поле, в формате VF

```
5.2.2.3 Vec2d computeOptFlow ( subSize * kernel, Data2Db * leftImg, Data2Db * rightImg, Vec2d
    & dv, double * d_mid_x, double * d_mid_y, double * d_max_x, double * d_min_x,
    double * d_min_y, double * d_max_y, double * d_avg_x, double * d_avg_y )
```

computeOptFlow Вычисление вектора оптического потока

Аргументы

in	kernel	структура содержащая сведения о местонахождении пикселя, размерах окна поиска и прочего
in	leftImg	массив яркостей первого кадра
in	rightImg	массив яркостей второго кадра
in	dv	векторное поле предыдущего поля

Возвращает

[out] vf вектор оптического потока

B-spline interpolation or Bilinear interpolation

5.2.2.4 std::vector<Data2Db*>* createPyramid_v2 (Data2Db * img, int lvl_pyramid, QString pref)

createPyramid_v2 Выделение памяти для пирамиды изображений

Аргументы

in	img	Указатель на оригинальное изображение
in	lvl_pyramid	Уровень пирамиды
in	pref	Префикс имени для создаваемого слоя

Возвращает

[out] Список изображений

5.2.2.5 void derivativeVectorField (VF2d & vf, QString info)

derivativeVectorField Вычисление поля деформации

Аргументы

vf	Векторное поле
info	Имя сохраняемого файла

0.81649658092 - Это $\sqrt{2/3}$

5.2.2.6 void inversion (Matx22d & A)

inversion Нахождения обратной матрицы(2 x 2).

Аргументы

in	A	Ссылка на массив
----	---	------------------

5.2.2.7 double* multiplicMtrxAndVectr (double ** array, int * vector)

multiplicMtrxAndVectr Произведение матрицы на вектор

Аргументы

in	**array	указатель на массив
in	*vector	указатель на вектор

Возвращает

[out] tmp результат произведения

5.2.2.8 void printProgressBar (double fastProgBar, double slowProgBar)

printProgressBar Функция ответственная за отрисовку прогресс бара при использовании графического интерфейса

Аргументы

fastProgBar	Промежуточный уровень
slowProgBar	Полный уровень

5.2.2.9 Data2Db* resizeImage (Data2Db * image, int kK, QString prefix)

resizeImage Функция масштабирования изображения, для построения пирамиды уменьшенных изображений

Аргументы

in	image	Структура содержащая сведения о размерах масштабируемого изображения
in	kK	Коэффициент уменьшения изображения
in	prefix	Префикс имени для создаваемого слоя

Возвращает

[out] указатель на массив масштабированных изображений

5.2.2.10 void saveVfResult (VF2d & vf, QString info)

saveVfResult Сохранение векторного поля в формате VF. Просмотр возможен в программе dfcl

Аргументы

in	vf	Указатель на векторное поле
in	info	Имя сохраняемого файла

5.2.2.11 void writeHdf5File (QString filename, ProtoData2D & layer, bool writeMode)

writeHdf5File Сохранение всех результатов в формат HDF5

Аргументы

filename	Имя сохраняемого файла
layer	Указатель на слой
writeMode	Режим записи для данного слоя

5.3 Файл src/lk_interpolation.h

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполирование данных.

```
#include <deprecated/dvfile.h>
```

Функции

- double [bilinearInterpolation](#) (double delx, double dely, uchar q11, uchar q12, uchar q21, uchar q22, int x1, int y1)
bilinearInterpolation Билинейная интерполяция — в вычислительной математике расширение линейной интерполяции для функций двух переменных.
- double [cubicInterpolate](#) (double p[4], double x)
cubicInterpolate Кубическая интерполяция
- double [bicubicInterpolate](#) (double p[4][4], double x, double y)
bicubicInterpolate Бикубическая интерполяция

5.3.1 Подробное описание

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполирование данных.

5.3.2 Функции

5.3.2.1 `double bicubicInterpolate (double p[4][4], double x, double y)`

`bicubicInterpolate` Бикубическая интерполяция

Аргументы

in	p	Массив значений
in	x	Массив смещений по оси x
in	y	Массив смещений по оси y

Возвращает

[out] Интерполированное значение

5.3.2.2 `double bilinearInterpolation (double delx, double dely, uchar q11, uchar q12, uchar q21, uchar q22, int x1, int y1)`

`bilinearInterpolation` Билинейная интерполяция — в вычислительной математике расширение линейной интерполяции для функций двух переменных.

Аргументы

in	delx	Смещение по x
in	dely	Смещение по y
in	q11	Значение функции в левом верхнем углу
in	q12	Значение функции в левом нижнем углу
in	q21	Значение функции в правом верхнем углу
in	q22	Значение функции в правом нижнем углу
in	x1	Координата верхнего левого угла по x
in	y1	Координата верхнего левого угла по y

Возвращает

[out] Интерполированное значение

5.3.2.3 `double cubicInterpolate (double p[4], double x)`

`cubicInterpolate` Кубическая интерполяция

Аргументы

in	p	Массив значений
in	x	Массив смещений

Возвращает

[out] Интерполированное значение

5.4 Файл src/lk_math.h

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за подсчет статистических показателей.

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <QDebug>
#include "dv.h"
```

Функции

- double expectation (double &var, int size)
 - double variance (double &var, int size)
 - double std_deviation (double &var, int size)
 - void [show](#) (double *var, int size)
- [show](#) Функция счета статистических показателей

5.4.1 Подробное описание

Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за подсчет статистических показателей.

5.4.2 Функции

5.4.2.1 void show (double * var, int size)

[show](#) Функция счета статистических показателей

Аргументы

in	var	Ссылка на массив данных.
in	size	Длина массива.

5.5 Файл src/lk_struct.h

Заголовочный файл с описанием используемых структур

Структуры данных

- struct [subSize](#)
The SubSize struct Данный файл содержит в себе определения основных структур, используемых в программе
- struct [imageInform](#)
The [imageInform](#) struct Хранение информации о размерах изображения

5.5.1 Подробное описание

Заголовочный файл с описанием используемых структур