

Lucas Kanade

1.4-19-gb49fe34

Создано системой Doxygen 1.8.9.1

Чт 11 Июн 2015 03:39:10

## Содержание

### 1 Lucas–Kanade

In computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion.

By combining information from several nearby pixels, the Lucas–Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

#### Version

1.2

#### Installation

You need install some packet: ##### Qt

```
1 $ sudo apt-get install qt-sdk
2 $ sudo apt-get install 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
```

#### ## Run

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

## Options

- 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

## Todo's

- Added pyramid
- Write Tests
- Add Code Comments for doxygen

## License

## GNU GPL

Repository created in order to write a graduate work on specialty 220301.

## 2 Алфавитный указатель структур данных

### 2.1 Структуры данных

Структуры данных с их кратким описанием.

#### [imageInform](#)

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

#### [subSize](#)

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

## 3 Список файлов

### 3.1 Файлы

Полный список документированных файлов.

#### [src/global.h](#)

Заголовочный файл с описанием глобальных переменных ??

#### [src/lk\\_func.h](#)

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

src/lk_interpolation.h	Заголовочный файл с описанием функций. Данный файл содержит в себе определения функций отвечающих за интерполирование данных	??
src/lk_math.h		??
src/lk_struct.h	Заголовочный файл с описанием используемых структур	??
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](#)  
Глобальная переменная Включить режим вывода в консоль отладочной информации
- [int g\\_sizeWindowSeach](#)  
Глобальная переменная Размер окна поиска
- [int g\\_stepForGrid](#)  
Глобальная переменная Размер шага сетки (через какое расстояние повторяются точки поиска)
- [int g\\_iteration](#)  
Глобальная переменная Число итераций (уточнений)
- [int g\\_interpolation](#)  
Глобальная переменная Выбор метода интерполирования(0 - Б-сплайн, 1 - Билинейный, 3 - Бикубический)
- [QString g\\_outputFolder](#)  
Глобальная переменная Директория для сохранения результатов работы
- [double g\\_fastProgBar](#)  
Глобальная переменная Переменная хранящая промежуточный уровень прогресс бара
- [double g\\_slowProgBar](#)  
Глобальная переменная Переменная хранящая полный уровень прогресс бара

#### 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 HD5F

### 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 HD5F

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
- 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	
in	x	
in	y	

## Возвращает

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

## bilinearInterpolation

## Аргументы

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

## Возвращает

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

## cubicInterpolate Кубическая интерполяция

## Аргументы

in	p	
in	x	

## Возвращает

## 5.4 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

## 5.4.1 Подробное описание

## Заголовочный файл с описанием используемых структур