dlib\_shape\_predictor库研究

目 录

[1 shape\_predictor.h 3](#_Toc525903365)

[2 full\_object\_detection.h 3](#_Toc525903366)

[参考文献 1](#_Toc525903367)

# shape\_predictor.h

namespace impl

{

struct split\_feature

{

unsigned long idx1;

unsigned long idx2;

float thresh;

};

struct regression\_tree

{

vector<split\_feature> splits;

vector<matrix<>> leaf\_values;

num\_leaves();

inline const matrix<float,0,1>& operator()(

const std::vector<float>& feature\_pixel\_values,

unsigned long& i) const

}

inline point\_transform\_affine unnormalizing\_tform (const rectangle& rect)

template <typename image\_type, typename feature\_type>

void extract\_feature\_pixel\_values (

const image\_type& img\_, const rectangle& rect,

const matrix<float,0,1>& current\_shape,

const matrix<float,0,1>& reference\_shape,

const std::vector<unsigned long>& reference\_pixel\_anchor\_idx,

const std::vector<dlib::vector<float,2> >& reference\_pixel\_deltas,

std::vector<feature\_type>& feature\_pixel\_values)

计算current和reference shape的转换，提取对于坐标的像素点

}

class shape\_predictor

{

private:

matrix<float,0,1> initial\_shape;

std::vector<std::vector<impl::regression\_tree> > forests;

std::vector<std::vector<unsigned long> > anchor\_idx;

std::vector<std::vector<dlib::vector<float,2> > > deltas;

public:

template <typename image\_type>

full\_object\_detection operator()(const image\_type& img,

const rectangle& rect) const

}

# full\_object\_detection.h

class full\_object\_detection

{

private:

rectangle rect;

std::vector<point> parts;

public:

point& part(unsigned long idx);

rectangle& get\_rect();

unsigned long num\_parts() const;

}

# points\_transforms.h

template <typename T>

point\_transform\_affine find\_similarity\_transform (

const std::vector<dlib::vector<T,2> >& from\_points,

const std::vector<dlib::vector<T,2> >& to\_points)

参考文献“Least-squares estimation of transformation parameters between two point patterns”的公式34-43

template <typename T>

point\_transform\_affine find\_affine\_transform (

const std::vector<dlib::vector<T,2> >& from\_points,

const std::vector<dlib::vector<T,2> >& to\_points)

class point\_transform\_affine

{

private:

matrix<double,2,2> m;

dlib::vector<double,2> b;

}

# rectangle.h

class rectangle

{

private:

long l;

long t;

long r;

long b;

}

# matrix.h

const matrix\_exp subm (const matrix\_exp& m, long row, long col, long nr, long nc);

函数功能：截取矩阵m的以row和col为起点的，大小为nr和nc的子矩阵。

const matrix\_exp colm (const matrix\_exp& m, long col);

函数功能：返回矩阵m的第col列。

参考文献

1. Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun. Deep Residual Learning for Image Recognition. CVPR 2016.