dlib库里实现人脸对齐采用的是shape\_predictor.h里定义的shape\_predictor类，产生人脸特征点的代码为：

shape\_predictor sp; // 声明一个shape\_predictor的实例

deserialize(argv[1]) >> sp; // 从argv[1]中加载参数

// img为图片，rect为人脸框，full\_object\_detection为特征点的类

// shape包含了输出的68个人脸特征点坐标，通过重载运算符()实现

full\_object\_detection shape = sp(img, rect);

人脸对齐的算法大致可以分成三个步骤：提取特征点灰度，经过回归树和去归一化输出。

* 1. 提取特征点灰度

一个shape\_predictor类包括四个private变量：

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;

其中，initial\_shape为初始化的特征点坐标，forests为回归树，anchor\_idx和deltas分别为锚点索引和偏差。

提取特征点灰度由函数

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)

其中，current\_shape为待估shape，reference\_shape为初始化的initial\_shape。该函数先计算current\_shape和initial\_shape的变换矩阵，以及人脸矩形框到单位矩形的去归一化变换矩阵，变换矩阵的求法参考文献[3]的公式(34)到公式(43)；然后根据文献[1]的公式()，将reference\_shape中的特征点坐标映射到current\_shape上，并去归一化到图片的坐标系统中；最后根据映射的坐标，提取相应位置的图片灰度强度，并传递到变量feature\_pixel\_values中。