# 目录

[目录 1](#_Toc3403258)

[金相图像的分割 2](#_Toc3403259)

[1. 方法概述 2](#_Toc3403260)

[2. 工程包含文件 2](#_Toc3403261)

[3. 本程序运行环境和所需库函数 3](#_Toc3403262)

[4. 算法流程图 4](#_Toc3403263)

[5. 代码详解 4](#_Toc3403264)

[6. 软件介绍 8](#_Toc3403265)

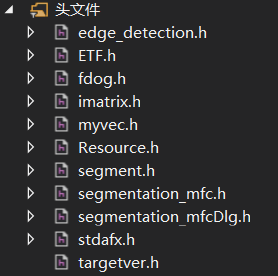
# 金相图像的分割

## 方法概述

本分割方法主要针对同类材料图像数目非常少的情况。方法主要分为两个部分：第一部分主要是通过对像素进行聚类实现初步分割得到区域块，具体包括高斯模糊，使用meanshift对像素进行分类以及使用floodfill将像素合并成区域块。第二部分主要是通过对第一部分分割得到的区域块进行聚类并进一步合并区域块，具体包括特征提取提取区域块的特征，以及使用DBSCAN对区域块进一步聚类，然后合并相邻的同类区域块。

## 2. 工程包含文件

**Segmentation\_mfc（项目）**





1. Segmentation\_mfc.h, Segmentation\_mfcDlg.h, Resource.h, stdafx.h, targetver.h, Segmentation\_mfc.cpp, Segmentation\_mfcDlg.cpp, stdafx.cpp是MFC界面相关的源文件，包括消息响应等，主要是消息控件、按钮。这部分代码和功能没关系，主要是界面。
2. ETF.h, fdog.h, imatrix.h, myvec.h, ETF.cpp, fdog.cpp这些是边缘检测相关代码，是由fdog方法的提出者公开的，这部分代码一般不要改动。
3. edge\_detection和edge\_detection.cpp是本方法对现有fdog的改进进行边缘检测，主要是采用了双阈值。一般处理图像时不需用到边缘检测。
4. segment.h，segment\_func.cpp是具体分割方法的各种功能实现，基本上大部分与算法相关的函数都包含在这两个文件中，也是核心。
5. dbscan.py,kmeans.py是对区域块进行聚类时调用的python脚本

## 3. 本程序运行环境和所需库函数

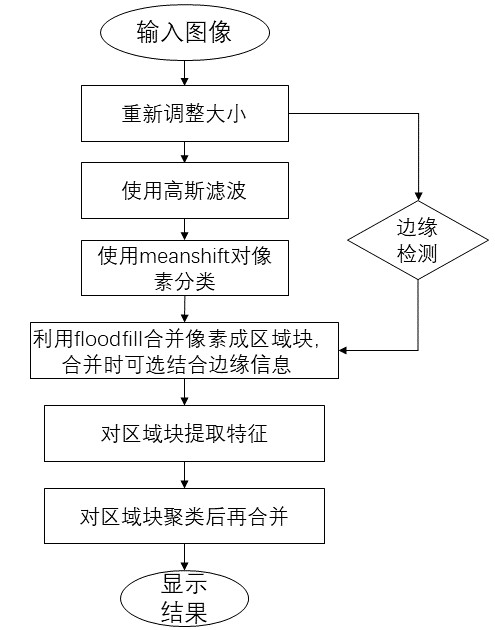
操作系统：Windows系统

运行语言：程序在Visual studio2012下使用C++开发

程序开发使用的函数库：C++下的OpenCV2.4.9, python运行环境及机器学习相关库(numpy,scikit-learn)

可执行程序运行所需环境：需安装python运行环境及机器学习相关库(numpy,scikit-learn)

## 算法流程图



## 代码详解

**5.1 主要分割函数（segment.h，segment\_func.cpp）**

**5.1.1结构体**

typedef struct ConnectedComp

{

int area;

std::vector<cv::Point> pointSet; //区域块的点集

int mean; //区域块的灰度均值

double var; //区域块的灰度方差

double hogHist[hogBins]; //区域块的hog

double lbpHist[lbpBins];//区域块的lbp,没用上

int num;

}

这个结构体主要代表的是区域块的信息

**5.1.2主要函数**

1. void segmentImg::segmentMain(std::string filename,int dim,int hs,int hr,int edge\_mode,int edge\_count,int clustering\_mode,int kmeans\_k,double dbscan\_eps,int dbscan\_min){

功能：算法主函数

作者：赖传滨

改写：无

输入：filename:图片路径

Dim:高斯滤波大小

hr,hs: meanshift参数,颜色范围和空间范围

edge\_mode:是否使用边缘检测

edge\_count:是否第一次边缘检测

clustering\_mode:聚类采用dbscan还是kmeans

kmeans\_k:kmeans参数k

dbscan\_epa, dbscan\_min: dbscan参数

返回：无。显示分割后的图像以及微观结构个数。

1. void segmentImg::segmentMainSimple(std::string filename,int edge\_mode,int edge\_count,int clustering\_mode,int kmeans\_k,double dbscan\_eps,int dbscan\_min){

功能：如果想要在保持初步分割相关参数不变的情况下，调整聚类部分相关参数时，调用此函数，可以重复利用初步分割的结果无需重新进行meanshift、边缘检测等步骤。

作者：赖传滨

改写：无

输入：filename:图片路径

edge\_mode:是否使用边缘检测

edge\_count:是否第一次边缘检测

clustering\_mode:聚类采用dbscan还是kmeans

kmeans\_k:kmeans参数k

dbscan\_epa, dbscan\_min: dbscan参数

返回：无。显示分割后的图像以及微观结构个数。

1. void doubleThresholdEdge(Mat \_grayImage, Mat image, Mat &edge\_result)

功能：双阈值fdog边缘检测方法

作者：赖传滨

改写：无

输入：\_grayImage: 待检测灰度图

image: 图像（主要是提供图像大小信息）

edge\_result:边缘检测结果

返回：边缘检测结果

1. int segmentImg::newfloodfill( Mat &\_image, Mat &\_mask,Point seedPoint, int loDiff, int upDiff, int flag\_floodfill ,int range\_area,vector<ConnectedComp> &ccompSet)

功能：floodfill合并相似像素成区域块，此处使用递归实现，需设置较大的堆栈保留大小（属性->链接器->系统）（可改为宽度优先遍历实现，则无需设置）

作者：赖传滨

改写：无

输入：\_image: meanshift处理后的图像

\_mask: 掩码，标识已填充的像素

seedPoint: 种子点

loDiff,upDiff：像素差异的阈值

flag\_floodfill:4邻域还是8邻域（默认8邻域）

range\_area:区域块大小阈值，太小删去

ccompSet:空的区域块结构体vector

返回：每个区域块的面积，区域块结构体集合vector

1. void segmentImg::outputMeanAndVar(vector<ConnectedComp> &ccompSet, Mat &\_gray)

功能：计算并输出区域块的灰度均值到txt文件

作者：赖传滨

改写：无

输入：ccompSet：区域块集合

\_gray：原图像灰度图

返回：无，输出每个区域块的灰度均值到results\\mean.txt文件

1. void segmentImg::outputHOG(vector<ConnectedComp> &ccompSet,Mat &\_gray)

功能：计算并输出区域块的hog特征到txt文件

作者：赖传滨

改写：无

输入：ccompSet：区域块集合

\_gray：原图像灰度图

返回：无，输出每个区域块的hog特征到results\\hogHist.txt文件

1. void segmentImg::clustringAmerging\_k(Mat &\_result, Mat &\_gray, vector<ConnectedComp> &ccompSet, int kmeans\_k)

功能：读取kmeans聚类结果（由kmeans.py生成的results\\kmeans\_labels.txt文件）并据此对区域块进行合并(上色)

作者：赖传滨

改写：无

输入：\_result：结果图（无用）

\_gray：原图像灰度图

ccompSet: 待合并区域集合

kmeans\_k: kmeans参数

返回：无，返回最终合并结果

1. void segmentImg::clustringAmerging\_db(Mat &\_result, Mat &\_gray, vector<ConnectedComp> &ccompSet)

功能：读取dbscan聚类结果（由dbscan.py生成的results\\dbscan\_labels.txt文件以及results\\dbscan\_nlabels.txt文件）并据此对区域块进行合并(上色)

作者：赖传滨

改写：无

输入：\_result：结果图（无用）

\_gray：原图像灰度图

ccompSet: 待合并区域集合

返回：无，返回最终合并结果

**5.2 边缘检测相关函数（edgedetection.h，edgedetection.cpp）**

**5.2.1主要函数**

1. void newFDoG(Mat grayImage, Mat image, Mat &result\_image, double tao)

功能：对图像采用FDOG方法检测边缘

作者：赖传滨

改写：无

输入：grayImage：待检测图像

image: 主要提供图像大小信息

result\_image：检测结果

tao: FDoG参数，控制细节的多少

返回：给定tao值下的fdog边缘检测结果

**5.3 聚类相关脚本（kmeans.py，dbscan.py）**

C++代码中是用system("python dbscan.py");和system("python kmeans.py");来调用这两个脚本的

**5.3.1 脚本解释**

1. dbscan.py

说明: 读取dbvar.txt中有关dbscan参数的设置, hogHist.txt和mean.txt文件中有关区域块的特征值，对特征值归一化和聚类，输出聚类结果

作者：赖传滨

改写：无

输出：results\\dbscan\_labels.txt文件（每一个区域块的类别标签）

Results\\dbscan\_nlabels.txt文件（总类数）

1. kmeans.py

说明: 读取读取results\\kmeansvar.txt中有关kmeans参数的设置，hogHist.txt和mean.txt文件中有关区域块的特征值，对特征值归一化和聚类，输出聚类结果

作者：赖传滨

改写：无

输出：results\\kmeans\_labels.txt文件（每一个区域块的类别标签）

**5.4. 界面控件相应函数（segmentation\_mfcDlg.cpp）**

**5.4.1 主要函数**

1. Void Csegmentation\_mfcDlg::OnBnClickedButtonSegment()

功能：按下分割按钮，执行分割函数

作者：赖传滨

改写：无

界面输入：dim:高斯滤波大小

hs:meanshift空间阈值

hr:meanshift颜色阈值

k:kmeans参数

eps:dbscan参数,核心点邻域大小

minpts:dbscan参数，核心点邻域内最少点数

输出：分割后的图像

## 软件介绍

双击运行segmentation\_mfc.exe即可执行程序：

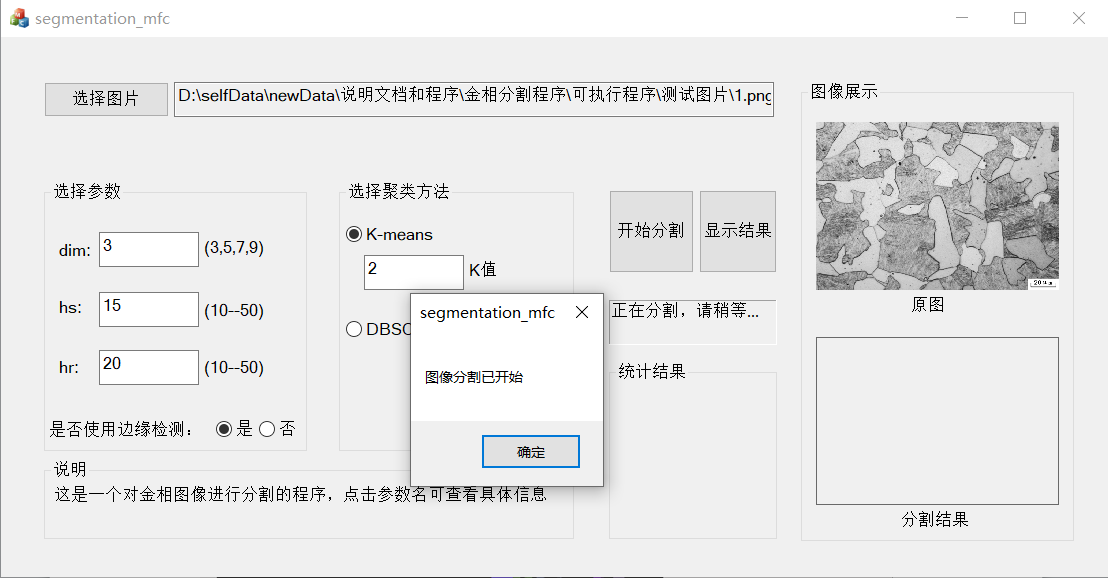


1. **软件各个功能**
2. **选择图片**：选择需要分割的图片
3. **选择参数:** 此部分有3个参数需要设定，dim是高斯核大小，hs是meanshift方法空间域值，hr是颜色阈值，这三个参数都有默认值。
4. **是否使用边缘检测:** 选择是否使用边缘检测。（测试图片需要边缘检测，其他图片不一定需要）
5. **选择聚类方法:** 可以选kmeans和dbscan作为区域聚类的方法
6. **开始分割:** 点击此按钮会执行分割程序
7. **显示结果:** 点击此按钮，会显示分割结果
8. **统计结果：**显示金相类别数以及各类微观结构的数目
9. **软件执行流程**

先点击选择按钮，选择需要分割的图片，选择的图片会显示在右侧的“原图”框内。然后设置参数，参数都有默认值。需注意，本程序在聚类过程中调用了python脚本，所以需要安装有python运行环境以及相关库(numpy,scikit-learn)。



点击“开始分割”：



分割结束后，点击“显示结果”按钮显示最终的分割结果：



分割结果的好坏取决于参数的设置，分割中间结果以及最终结果都会会保存在results目录下，最终分割结果的文件名为“result\_clustering.png”。微观结构的排序结果也会保存，文件名为“component 1.png”、“component 2.png”以此类推，一张排序图代表一类微观结构。

另外还会生成中间结果方便查看中间结果的好坏，来帮助调节参数。中间结果有meanshift结果“meanshift\_result\_gray.png”，floodfill结果“flood.png”，边缘检测结果“EdgeFresult.png”

