VisAlg2DBase

**1、Pyramid**

共两个算子，一个是仅得到最高层图像，一个是可以得到每一层的图像信息。

**1.1**

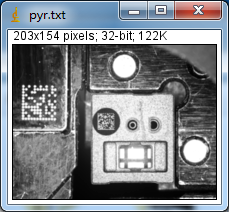
int CVisAlg2DBase::VisPyramid(IMG\_UBBUF src, unsigned char \* pDst, int & pyramid\_width, int & pyramid\_height, int level)

Author：Liu Ping

参数level=2



**图1.1 原图** （width=1624，height=1228）



**图1.2 第三层图（width=203，height=154）**

**分析：**函数输出level层的图像，以及该层图像的宽和高。当level输入成0的时候，函数返回-5，是函数ippiPyramidLayerDownGetSize\_8u\_C1R的返回值。

**1.2**

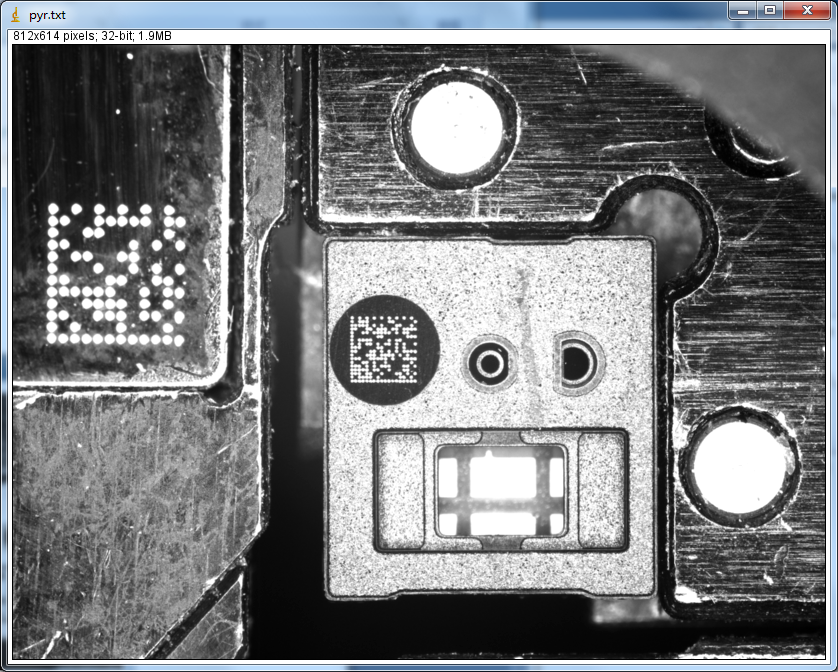
IppStatus CVisAlg2DBase::VisPyramid2(Ipp8u\* pSrc, IppiSize roiSize, IppiPyramid \*&pPyrStruct, Ipp8u \*\*&pPyrImage, int level)

Author: Shen Jiancheng

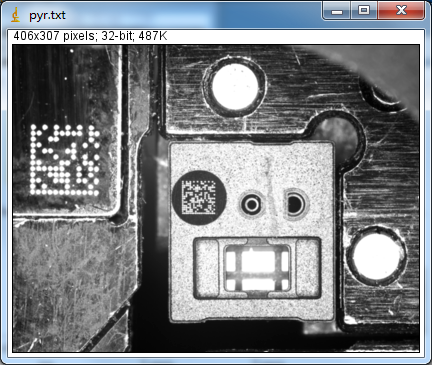
参数level=2



**图1.3 原图**



**图1.4 第一层**



**图1.5 第二层**

分析：可得到每一层的图像，图像信息保存在IppiPyramid \*&pPyrStruct,和Ipp8u \*\*&pPyrImage中。当level输入成0的时候，函数返回-5，是函数ippiPyramidGetSize的返回值。

**2、Filter**

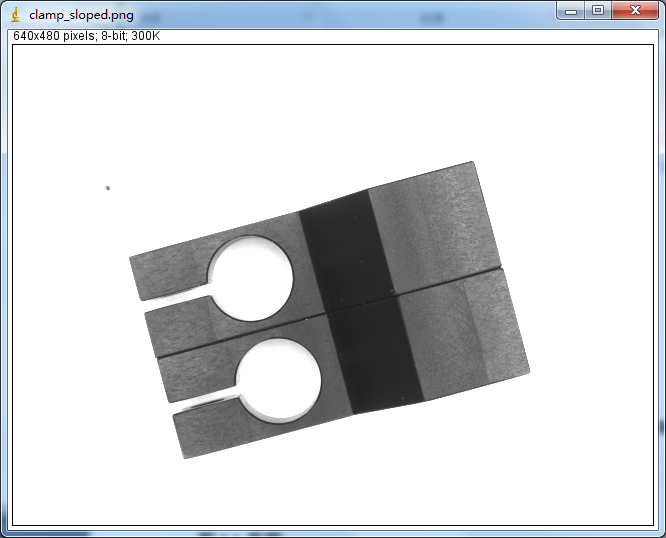
共有三种滤波器：均值、高斯、中值。

**2.1均值滤波**

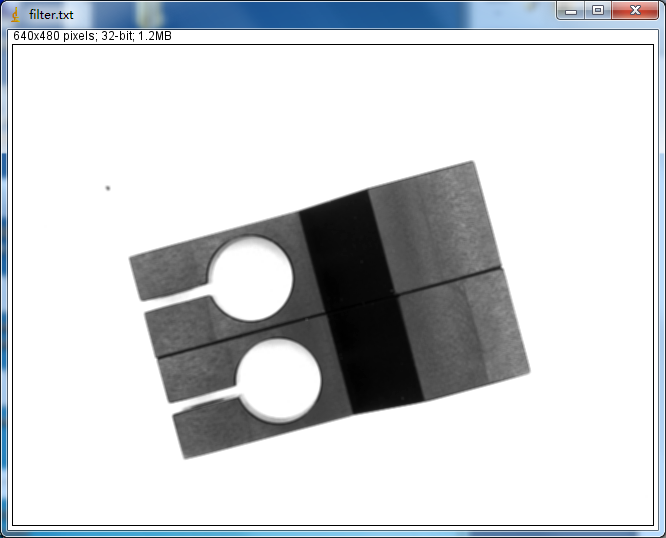
int CVisAlg2DBase::VisFilterMean(const unsigned char \*src, const int srcHeight, const int srcWidth, unsigned char \*dst, const unsigned char kernelSize, unsigned int divisor)

Author: Jiang He

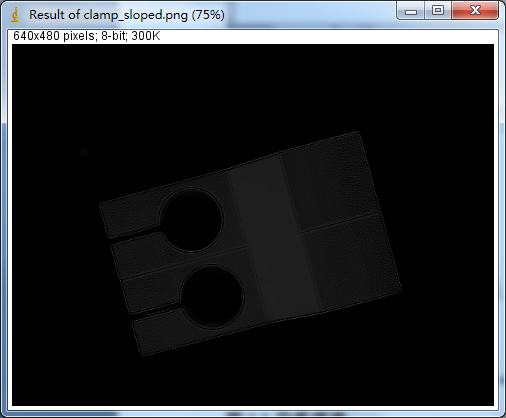
参数kernelSize =3，divisor=9

****

**图2.1 原图**

****

**图2.2 均值滤波**

****

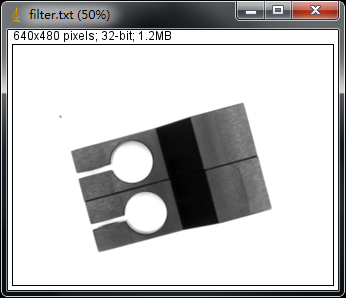
**图2.3均值滤波前后相减**

**2.2高斯滤波**

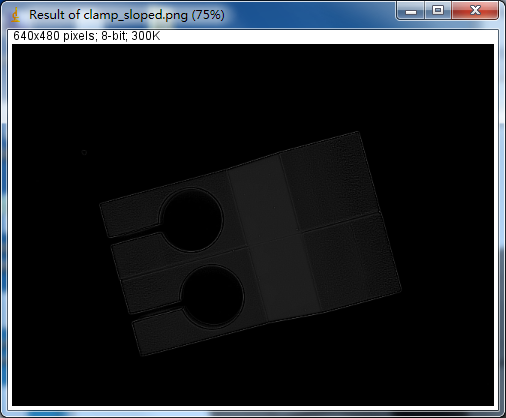
int CVisAlg2DBase::VisFilterGaussian(const unsigned char \*src, const int srcHeight, const int srcWidth, unsigned char \*dst, const unsigned char winWidth)

Author: Jiang He

参数winWidth =3

****

**图2.4 高斯滤波**

****

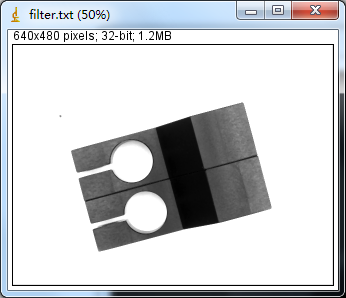
**图2.5 高斯滤波前后相减**

**2.3中值滤波**

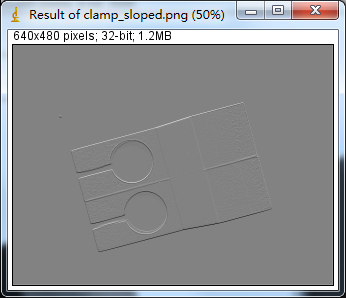
int CVisAlg2DBase::VisFilterMedian(const unsigned char \*src, const int srcHeight, const int srcWidth, unsigned char \*dst, const unsigned char winWidth)

Author: Jiang He

参数winWidth =3

****

**图2.4 中值滤波**

****

**图2.4 中值滤波前后相减**

**3、Fitting**

共四个算子：圆拟合、圆弧拟合、椭圆拟合、直线拟合。

**3.1圆和圆弧拟合**

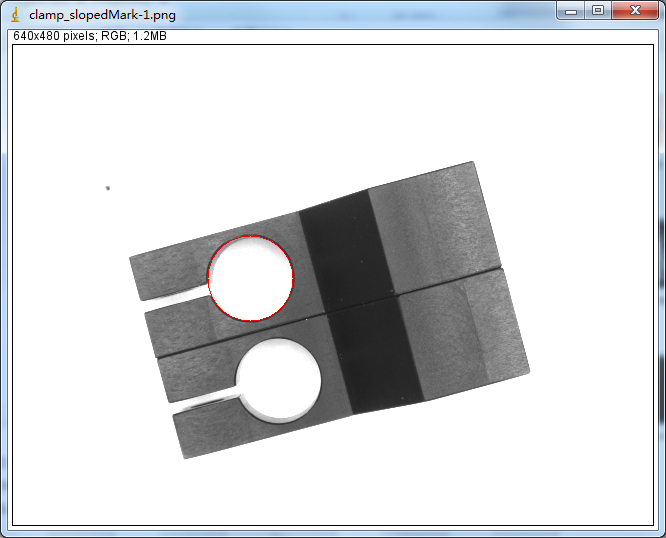
int VisFittingCircular(const float \*point\_pos, const int m, StructCircle &circular\_fit, const int iteration\_times);

Author: Shen Jiancheng

**参数设置1**：（255,193）（272,258）（206,263）（200,210）

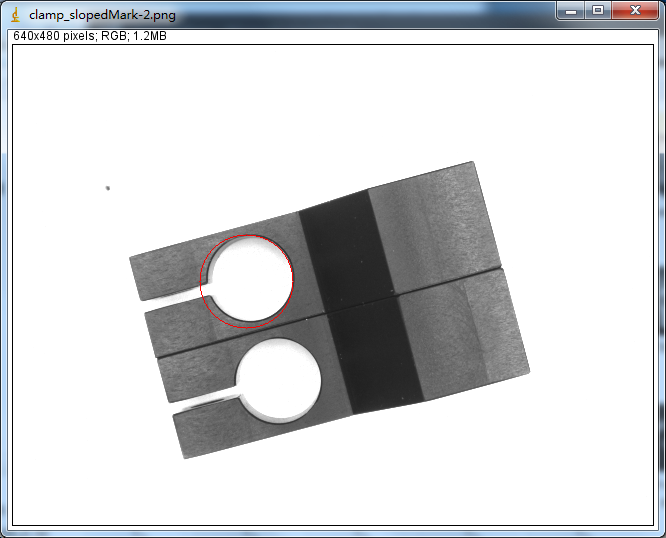
iteration\_times=20

Result：cirCen（236.966,232.522，r=43.3745



**参数设置2**：（266,202）（274,212）（280,228）

Result：cirCen（233.234,236.413)，r=47.5169



**分析：**1、输入点数较少的情况下，几个点距离越近，拟合的圆不标准。通常拟合圆是先进行边缘检测，用较多的点进行拟合，结果较准确。2、输入点坐标在一个数组，奇数是y坐标，偶数是x坐标，感觉这样有些反人类，用着不方便。一般把x坐标和y坐标放在两个数组，或者放在一个点类型的数组。

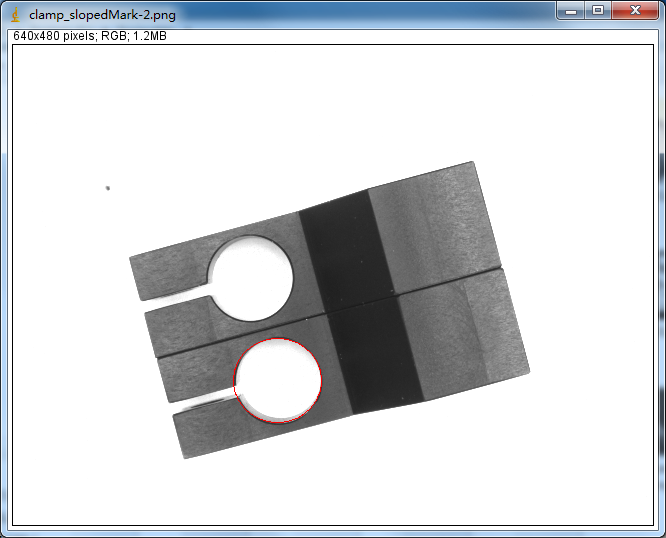
**3.2椭圆拟合**

bool CVisAlg2DBase::VisFitting\_Ellipse(Ipp32f \*xy, int m, EllipseStruct &resu)

Author: Shen Jiancheng

参数设置：Point[16] = {223,326,241,300,264,292,288,300,306,323,307,343,296,365,257,378}

Result：中心（264.78,335.596） 长半轴43.6028，短半轴42.4996



分析：同上，输入边缘点形式不方便。

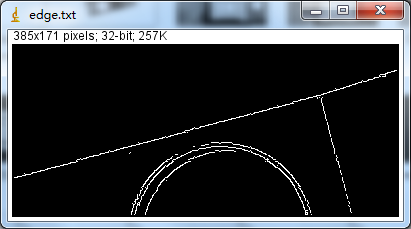
**3.3直线拟合**

IMG\_INT CVisAlg2DBase::GNERAL\_LINE\_FITTING(vector<IMG\_RCOORD>PointCor, IMG\_REAL &Slope, IMG\_REAL &B, IMG\_REAL Sigma, IMG\_INT FLAG)

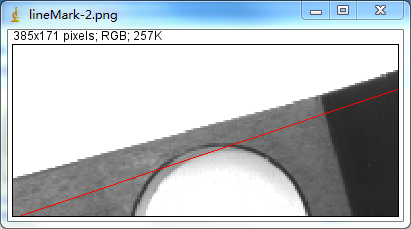
Author：Tan Ling

参数设置1：梯度强度100



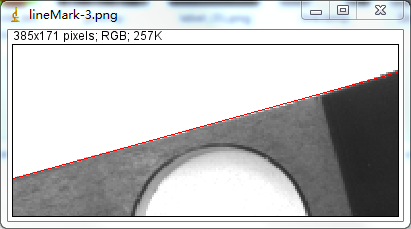
（边缘点）

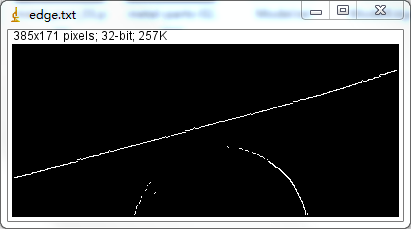
参数设置2：梯度强度200



（边缘点）

参数设置3：梯度强度300



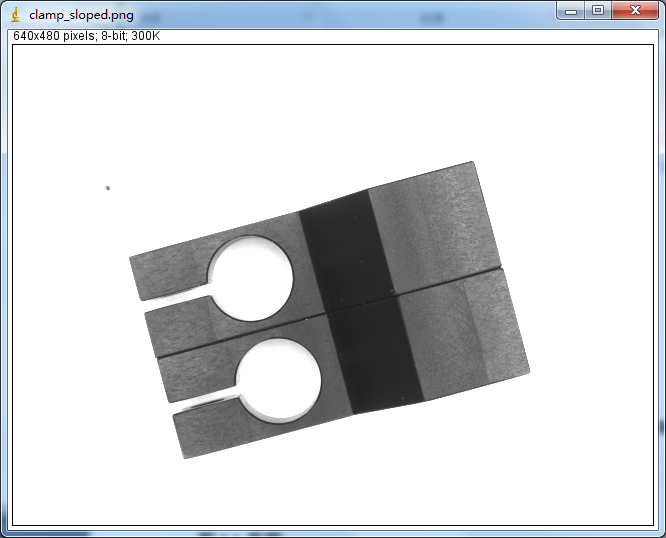
（边缘点）

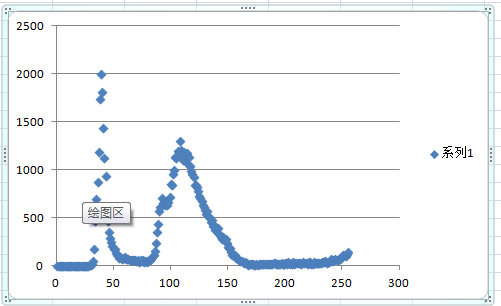
**分析：**当有较多的其它边缘点（非直线上）的边缘点时，拟合效果不理想。有待改进。

**4、Histogram**

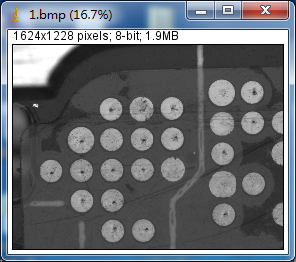
int VisHistogram(IMG\_UBBUF src, Ipp32u \* pHist, int nBins)

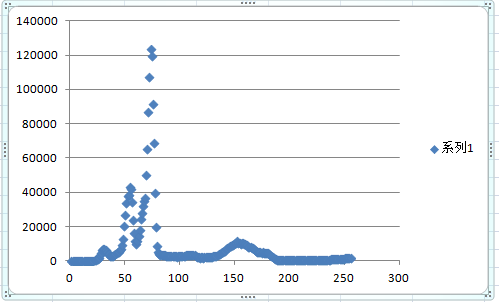
Author: Liu Ping



****

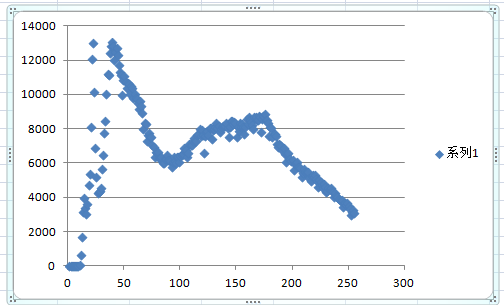
**图4.1** 图像1和灰度级分布情况



****

**图4.2**图像2和灰度级分布情况





**图4.3**图像3和灰度级分布情况

**5、Segment**

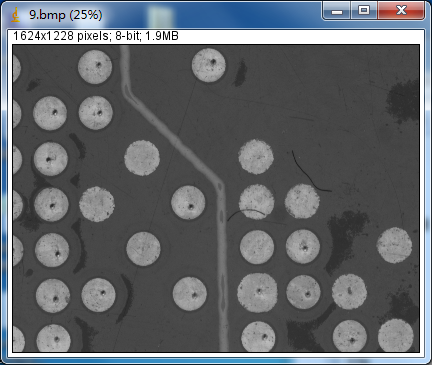
共5个算子：手动阈值分割、自动阈值分割、自动阈值计算不分割、动态阈值分割、数点阈值分割。

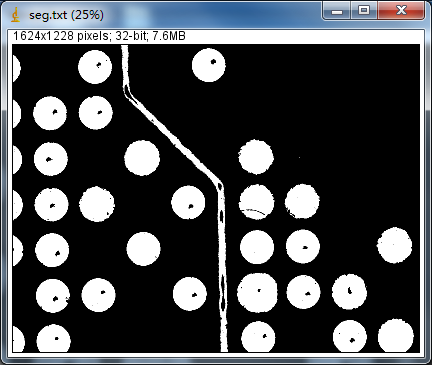
**5.1手动阈值分割**

int CVisAlg2DBase::VisSegmentManual(const unsigned char \* pSrc, const unsigned int nW, const unsigned int nH, unsigned char \* pDst, const unsigned char nT)

Author：Liu Ping

参数nT=100



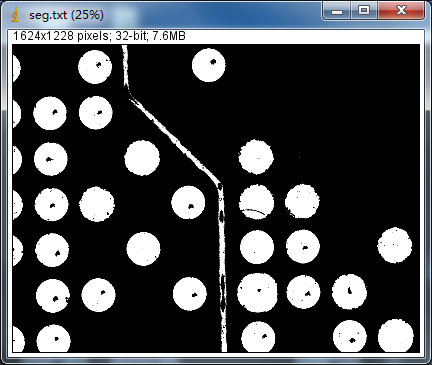


**图5.1** 手动输入阈值分割前后

**5.2自动阈值分割**

int VisSegmentOtsu(const unsigned char \* pSrc, const unsigned int nW, const unsigned int nH, unsigned char \* pDst)

Author：Liu Ping



**5.3自动阈值计算不分割**

int VisCalcOtsu(const unsigned char \* pSrc, const unsigned int nW, const unsigned int nH, unsigned char & nThres);

Author：Liu Ping

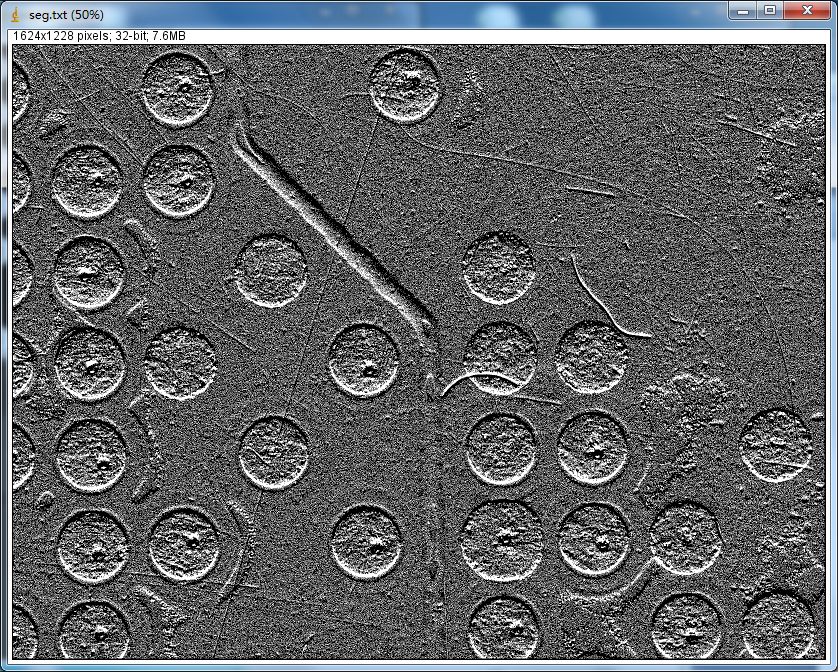
Result：nThres=107

**5.4动态阈值分割**

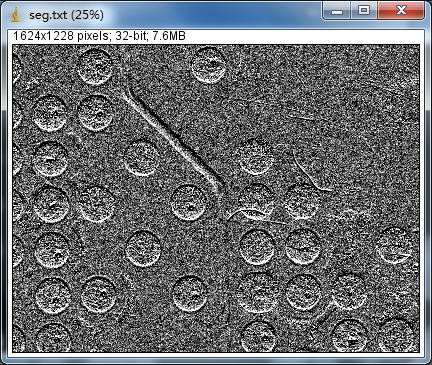
int VisSegmentDynamic(const unsigned char \* pSrc, const unsigned int nW, const unsigned int nH, unsigned char \* pDst, const unsigned int avgWinWidth);

Author：Liu Ping

参数1：avgWinWidth=3



参数2：avgWinWidth=5

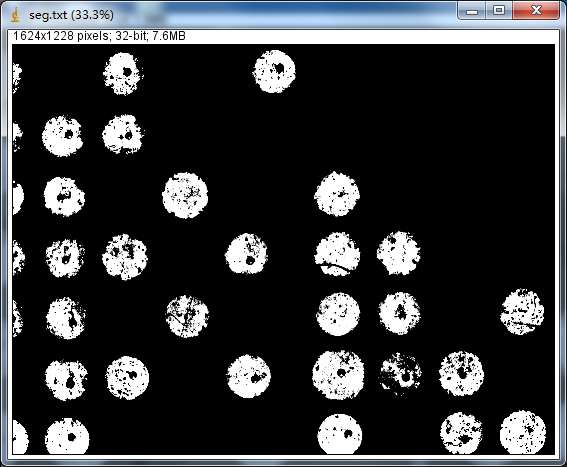


**5.5数点阈值分割**

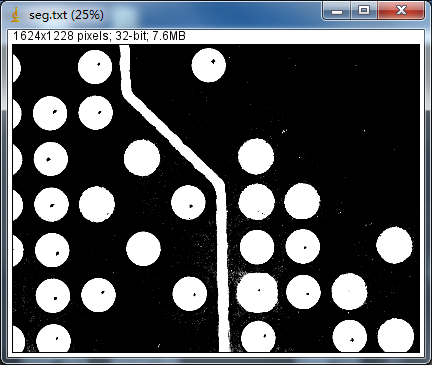
int VisSegmentCntDots(unsigned char \* pSrc, const unsigned int nW, const unsigned int nH, unsigned char \* pDst, const double fWhiteRatio);

Author：Liu Ping

参数1：fWhiteRatio =0.15

****

参数2：fWhiteRatio =0.25

****

**5.6分段求取阈值**

void VisGetBlobThreshold(Ipp8u \*srcRoi, IppiSize Roi, int BlobThreshold);

Author：Huang Yige

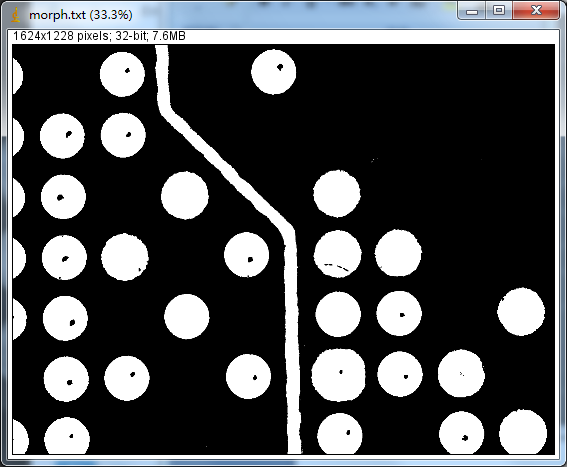
**分析**：只有输入参数，没有输出参数。有待确认。

**6、Morphology**

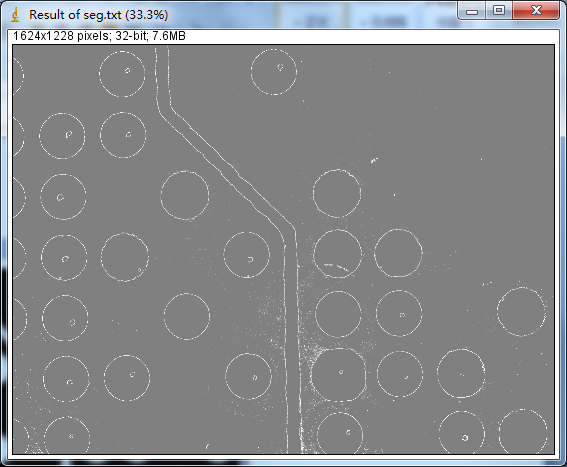
共8个算子：腐蚀、膨胀、开、闭、计算连通域信息、孔洞填充、寻找连通域的个数、计算图像的矩。

**6.1 腐蚀**

int VisMorphologyErode(unsigned char \* src1, int srcWidth, int srcHeight, unsigned char \* dst)



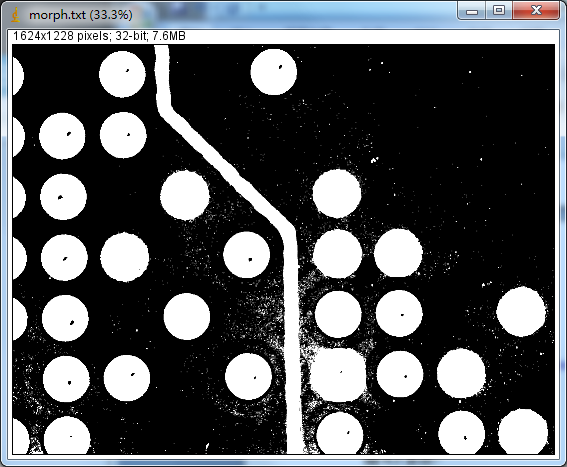
**图6.1** 腐蚀（maskSize=3）



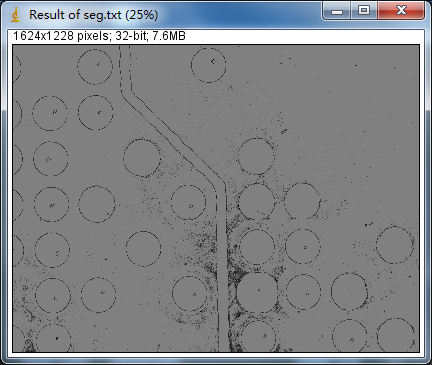
**图6.2** 腐蚀前后差值

**6.2膨胀**

int VisMorphologyDilation(unsigned char \* src1, int srcWidth, int srcHeight, unsigned char \* dst);



**图6.3** 膨胀



**图6.4** 膨胀前后差值

**6.3开运算**

int VisMorphologyOpen(IMG\_UBBUF src, IMG\_UBBUF dst)

Author：Liu Ping



**图6.5** 开运算



**图6.6** 开运算前后差值

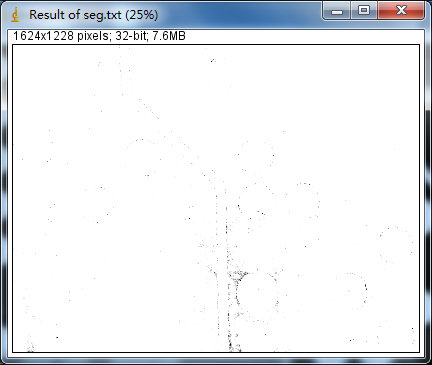
**6.4闭运算**

int VisMorphologyClose(IMG\_UBBUF src, IMG\_UBBUF dst)

Author：Liu Ping



**图6.7** 闭运算

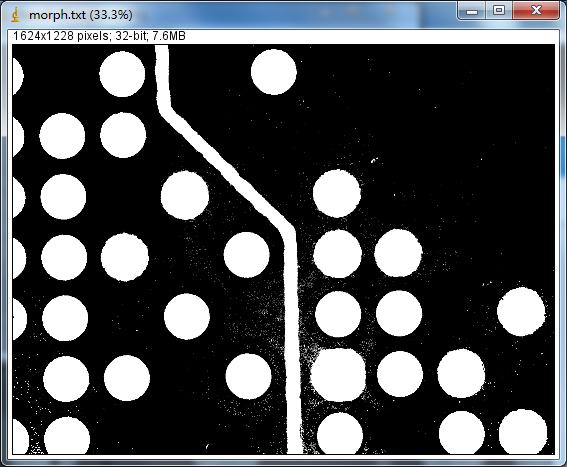


**图6.8** 闭运算前后差值

**6.5孔洞填充**

int VisHoleFill(unsigned char \* srcSeg, int srcWidth, int srcHeight, unsigned char \* dst)

Author：Jiang He

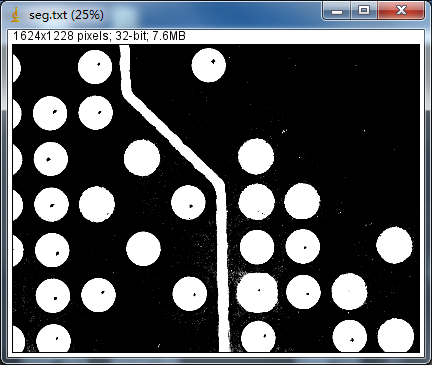


**6.6 计算图像的Moment**

void VisMoment(Ipp8u \*srcRoi, IppiSize Roi, double &hu)

Author：Huang Yige

Result：hu=0.00241162

（输入图像）

**6.7计算blob的个数**

void CVisAlg2DBase::VisLabelMarker(Ipp8u \*srcRoi, IppiSize Roi, int &markersNum)

Author：Huang Yige

Result: 5732 （输入图像同上）

**7、Gradient**

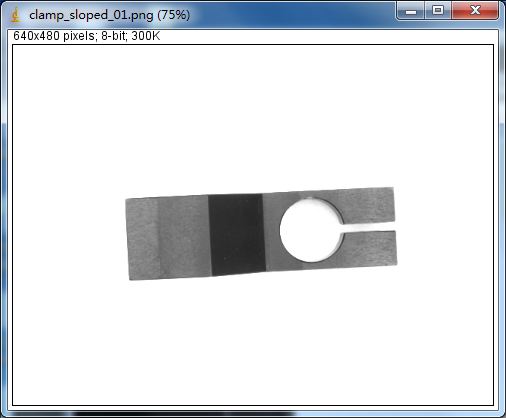
两个算子，有待增加。

**7.1 Sobel**

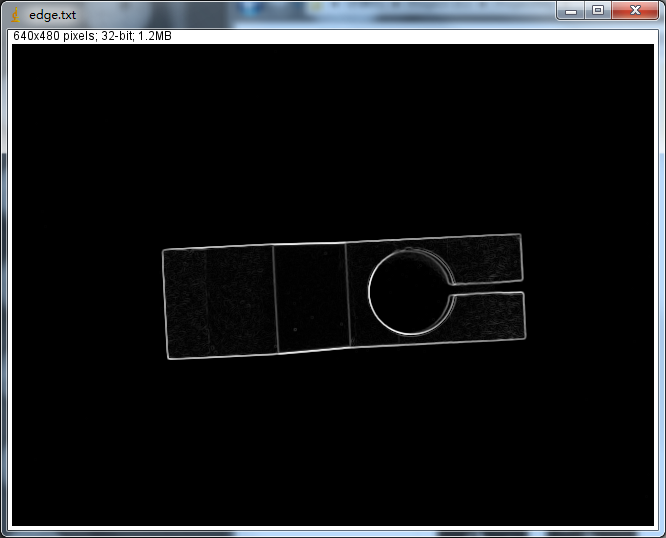
IMG\_INT VisEdge\_detection(IMG\_UBYTE \* srcRoi, IMG\_SIZE roiSize, int threshold, IMG\_WORD \* dstRoi, IMG\_UBYTE \* dstRoiE, edgeInformation \*& edgeArray, IMG\_INT & eNum)

Author：Jiang He

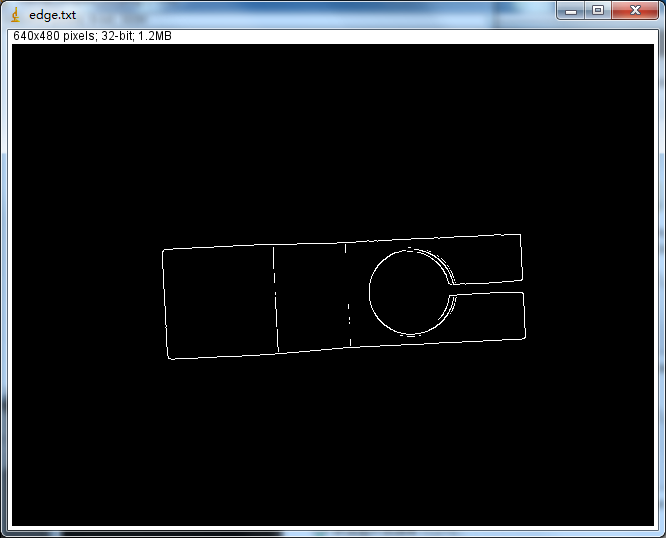
参数：threshold=200



**图7.1** 原图



**图7.2** Sobel梯度强度

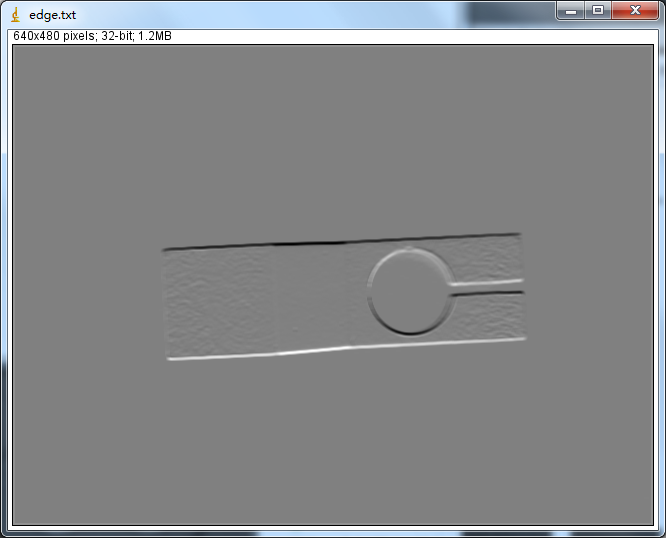
****

**图7.3** 边缘

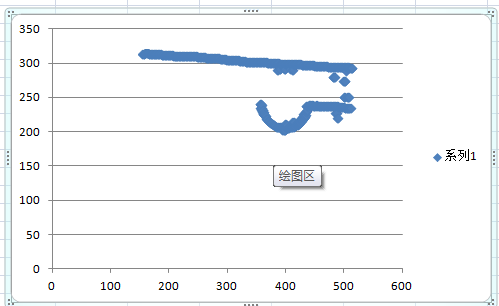
**7.2 Y方向**（上下矩形块之差得梯度）

int GradientCompute(unsigned char \* src, int srcWidth, int srcHeight, int rowNum, int colNum, float threshold, int \* dst, vector<edgeInformation>& edgePoint)

Author：Jiang He and Tan Ling

****

**图7.4** Y方向梯度强度



**图7.5** 边缘点（threshold=200）