陈伟灿 2023302181107

计算直方图

更新像素灰度值

计算映射关系

计算累计分布函数

计算直方图：计算每个灰度级的出现次数，即lCount[k]

计算累计分布函数(CDF)：

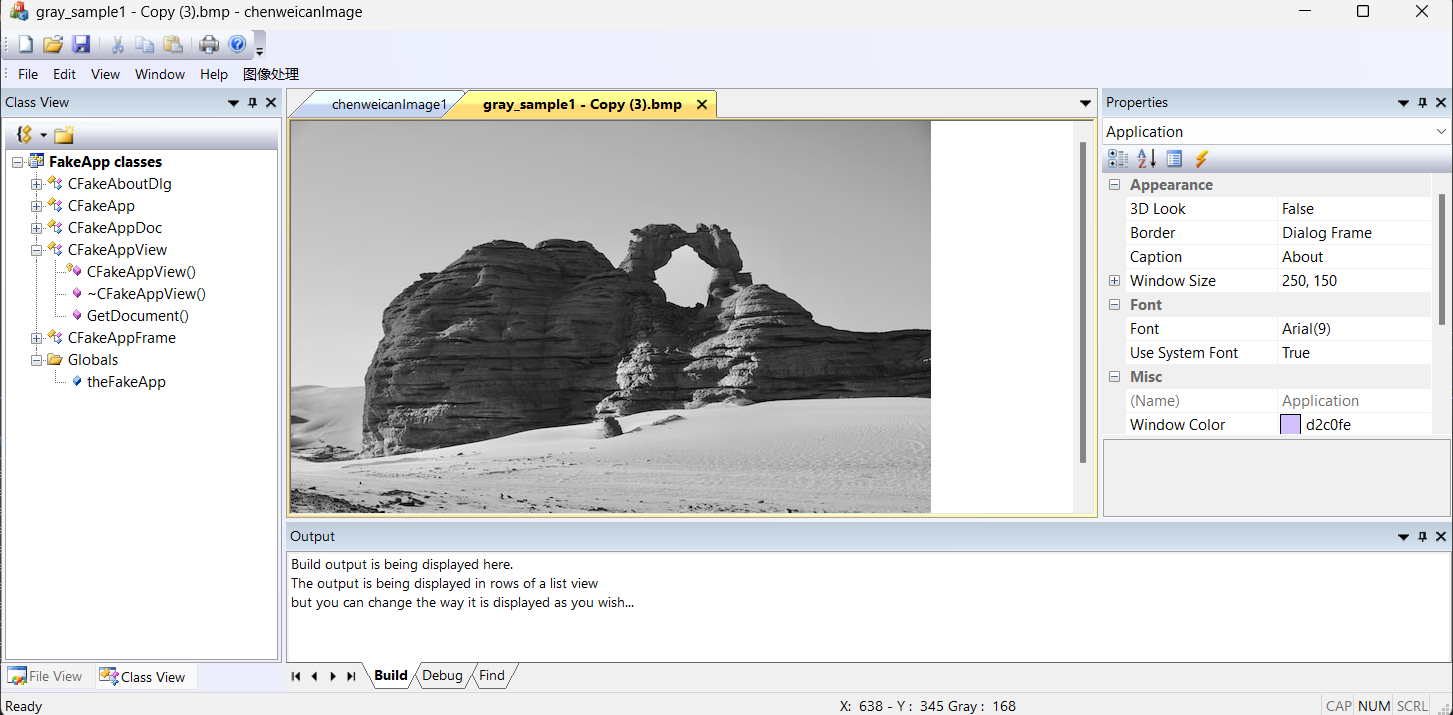
计算映射关系：

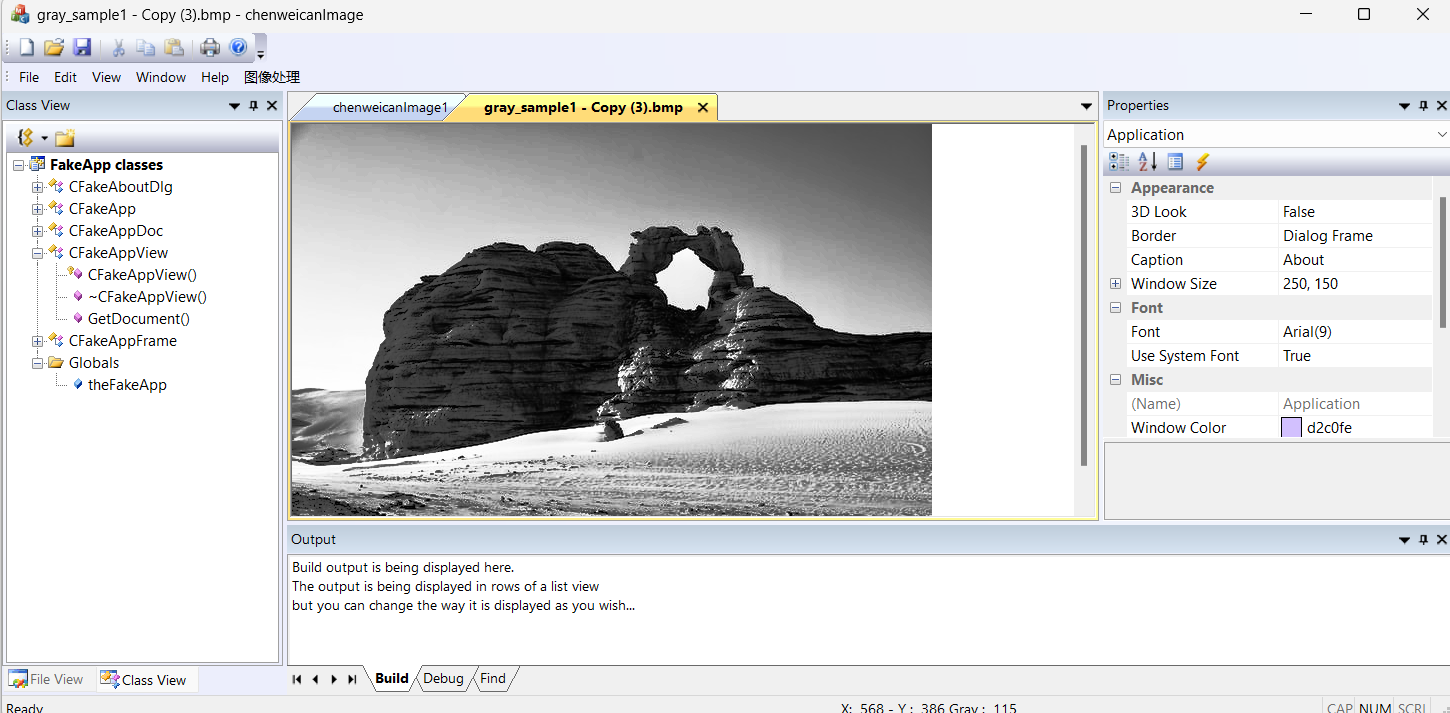
L-1保证像素取值在0到L-1内，在这里L为256

这里，映射关系储存到bMap里

更新像素灰度值

示例：





代码：

BOOL WINAPI InteEqualize(unsigned char\* lpDIBBits, LONG lWidth, LONG lHeight) {

// 指向源图像的指针

unsigned char\* lpSrc;

BYTE bMap[256];

LONG lCount[256];

// 图像每行的字节数

LONG lLineBytes = (((lWidth \* 8) + 31) / 32 \* 4);

// 重置计数为0

for (LONG i = 0; i < 256; i++)

lCount[i] = 0;

// 计算各个灰度值的计数

for (LONG i = 0; i < lHeight; i++) {

for (LONG j = 0; j < lWidth; j++) {

lpSrc = (unsigned char\*)lpDIBBits + lLineBytes \* i + j;

lCount[\*(lpSrc)]++;

}

}

// 计算灰度映射表

for (LONG i = 0; i < 256; i++) {

LONG lTemp = 0;

for (LONG j = 0; j <= i; j++)

lTemp += lCount[j];

bMap[i] = (BYTE)(lTemp \* 255 / lHeight / lWidth);

}

for (LONG i = 0; i < lHeight; i++) {

for (LONG j = 0; j < lWidth; j++) {

lpSrc = (unsigned char\*)lpDIBBits + lLineBytes \* (lHeight - 1 - i) + j;

\*lpSrc = bMap[\*lpSrc];

}

}

return TRUE;

}

void CchenweicanImageView::OnInteequalize(){

CchenweicanImageDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

unsigned char\* pBits = pDoc->m\_pBits;

int nWidth = pDoc->imageWidth;

int nHeight = pDoc->imageHeight;

int nColorBits = pDoc->m\_nColorBits;

if (nColorBits != 8){

MessageBox(\_T("目前只支持256色位图的运算！"), \_T("系统提示"), MB\_ICONINFORMATION | MB\_OK);

return;

}

if (InteEqualize(pBits, nWidth, nHeight)){

// 设置脏标记

pDoc->SetModifiedFlag(TRUE);

// 更新视图

pDoc->UpdateAllViews(NULL);

}

else{

// 提示用户

MessageBox(\_T("分配内存失败！"), \_T("系统提示"), MB\_ICONINFORMATION | MB\_OK);

}

EndWaitCursor();

}