**Computer and Robot Vision**

**Homework#3**

R01944040 柳成蔭

這次的作業是進行直方圖均衡Histogram Equalization，對原lena圖的histogram進行均衡化后得到亮暗分佈更均衡的圖像。

我使用VS2012編寫程式

1. 原圖的histogram

統計每個值對應的pixel的個數，記錄在histogram[]中。

|  |
| --- |
| Mat\_<uchar>::iterator iteK=img.begin<uchar>(),  iteKEnd=img.end<uchar>();  while (iteK!=iteKEnd)  {  histogram[(\*iteK)]++;  iteK++;  } |

根據對應的數量作為高度，畫出直方圖。

|  |
| --- |
| Mat imgHistogram(300,300,CV\_8UC1,Scalar(0));  Point a,b;  a.x=20; a.y=280;  b.x=275; b.y=280;  line(imgHistogram,a,b,CV\_RGB(255,255,255),1,8,0);  a.x=20; a.y=280;  b.x=20; b.y=0;  line(imgHistogram,a,b,CV\_RGB(255,255,255),1,8,0);  for (int k = 0; k <= 255; k++)  {  a.x=20+k; a.y=280;  b.x=20+k; b.y=280-( histogram[k]/10 );  line(imgHistogram,a,b,CV\_RGB(255,255,255),1,8,0);  } |

原圖和畫出的histogram：



1. Histogram Equalization

計算每個值對應點個數所占的比例，并累加起來得到累加比例分佈。

|  |
| --- |
| //cumPercentile初始化  double cumPercentile[256];  memset(cumPercentile,0,sizeof(cumPercentile));  //累加比例分佈  cumPercentile[0]=histogram[0];  for(int i=1;i<=255;i++)  {  cumPercentile[i]=cumPercentile[i-1]+(double)(histogram[i])/(double)(imgSize);  } |

之後把原圖中每個pixel的亮度替換成該亮度值的累加比例（累加比例需要對應到0~255的範圍）。

|  |
| --- |
| //均衡后的圖  Mat imgEqualization;  img.copyTo(imgEqualization);  //替換  Mat\_<uchar>::iterator iteEqu=imgEqualization.begin<uchar>(),  iteEquEnd=imgEqualization.end<uchar>();  while (iteEqu!=iteEquEnd)  {  (\*iteEqu)=(int)(cumPercentile[(\*iteEqu)]\*255.0+0.5);  iteEqu++;  } |

之後統計新圖的histogram并畫出均衡后的lena圖和它的histogram：

|  |
| --- |
| //histogram初始化  int hisEqualization[256];  memset(hisEqualization,0,sizeof(hisEqualization));  //統計histogram  Mat\_<uchar>::iterator iteHisEqu=imgEqualization.begin<uchar>(),  iteHisEquEnd=imgEqualization.end<uchar>();  while (iteHisEqu!=iteHisEquEnd)  {  hisEqualization[(\*iteHisEqu)]++;  iteHisEqu++;  }  //畫histogram  Mat imgHisEqualization(300,300,CV\_8UC1,Scalar(0));  Point aEqu,bEqu;  aEqu.x=20; aEqu.y=280;  bEqu.x=275; bEqu.y=280;  line(imgHisEqualization,aEqu,bEqu,CV\_RGB(255,255,255),1,8,0);  aEqu.x=20; aEqu.y=280;  bEqu.x=20; bEqu.y=0;  line(imgHisEqualization,aEqu,bEqu,CV\_RGB(255,255,255),1,8,0);  for (int k = 0; k <= 255; k++)  {  aEqu.x=20+k; aEqu.y=280;  bEqu.x=20+k; bEqu.y=280-( hisEqualization[k]/10 );  line(imgHisEqualization,aEqu,bEqu,CV\_RGB(255,255,255),1,8,0);  } |

Histogram Equalization後的結果：

