

计算机图像处理

COMPUTER IMAGE PROCESSING

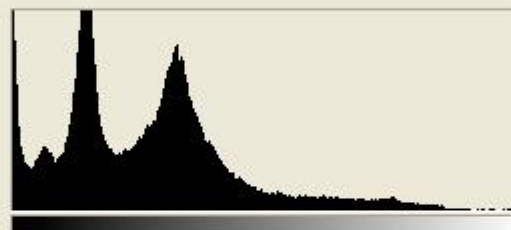
直方图修正

图像灰度统计直方图:



直方图

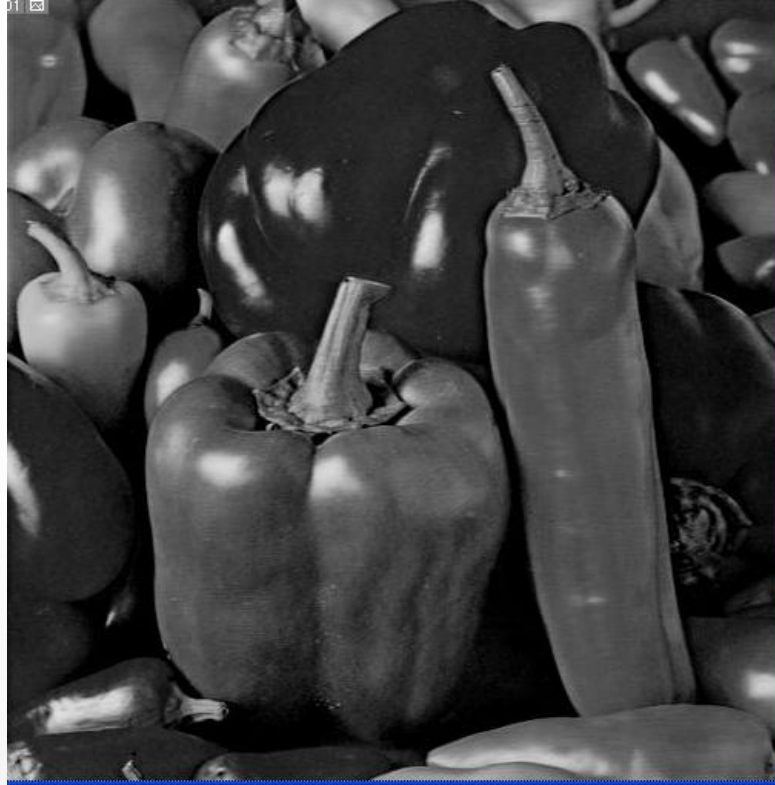
通道(C): 亮度



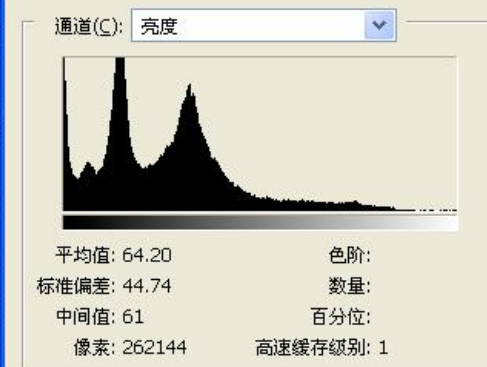
平均值: 64.20	色阶:
标准偏差: 44.74	数量:
中间值: 61	百分位:
像素: 262144	高速缓存级别: 1

$$p(s_k) = n_k / n$$

$$k = 0, 1, \dots, L-1$$

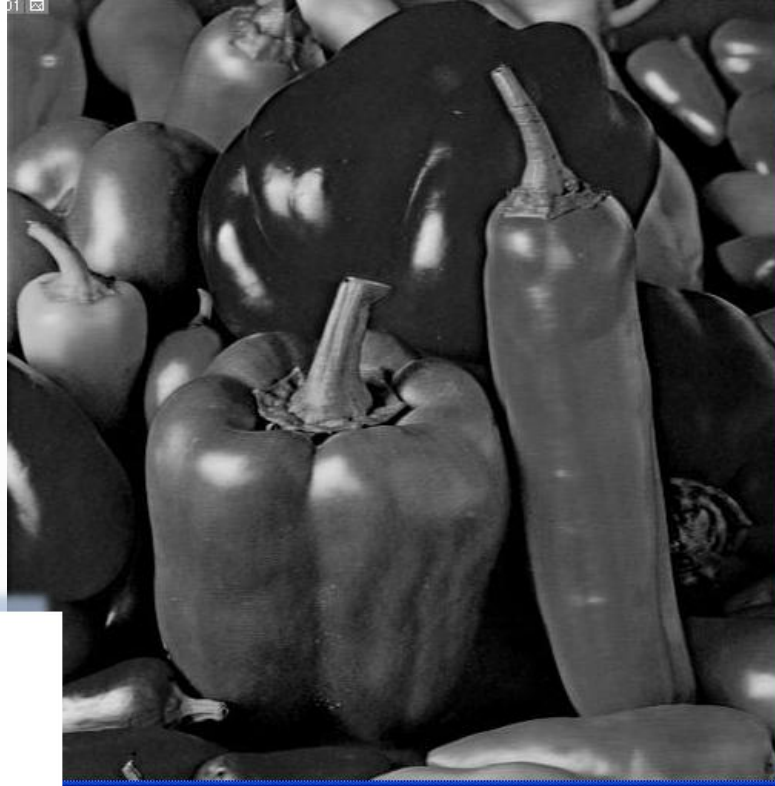


直方图

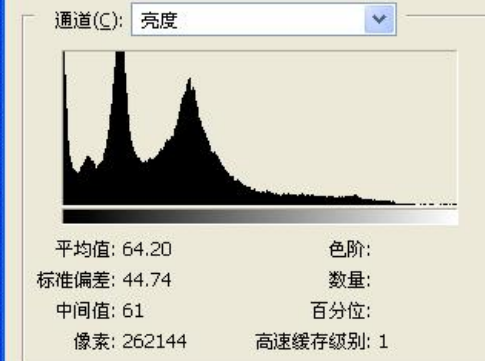


$$p(s_k) = n_k / n$$

$$k = 0, 1, \dots, L-1$$



直方图



```

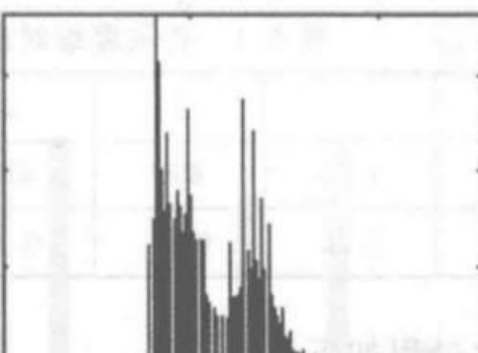
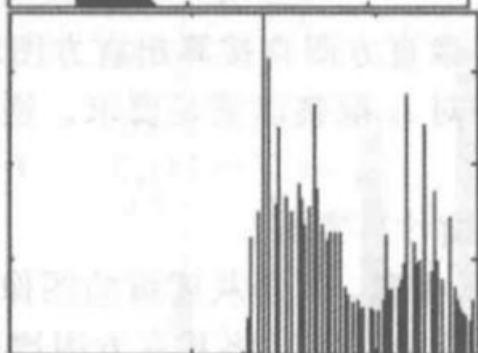
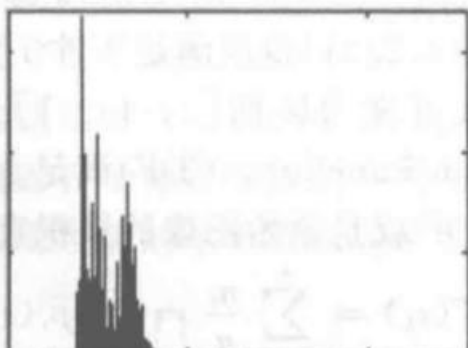
HXLBMPPFILE f;
float H[256];
int total;
int i;

memset(H,0,256*sizeof(float));
if (!f.LoadBMPFILE("c.bmp"))
return 1;

total = f.imagew*f.imageh;

.....

return 0;
    
```



直方图

通道(C): 亮度



平均值: 64.20

色阶:

标准偏差: 44.74

数量:

中间值: 61

百分位:

像素: 262144

高速缓存级别: 1



直方图

通道(C): 亮度



平均值: 64.20

色阶:

标准偏差: 44.74

数量:

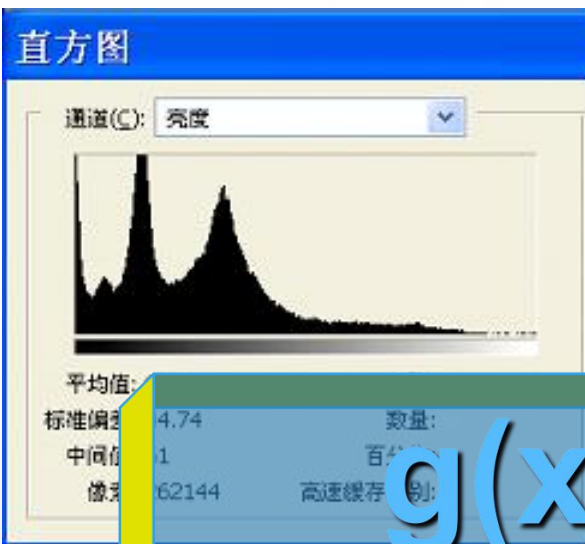
中间值: 61

百分位:

像素: 262144

高速缓存级别: 1



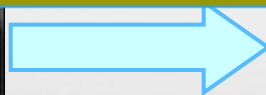


直方图均衡化



$$g(x,y) = T(f(x,y))$$

$T = ?$



T满足2个条件:

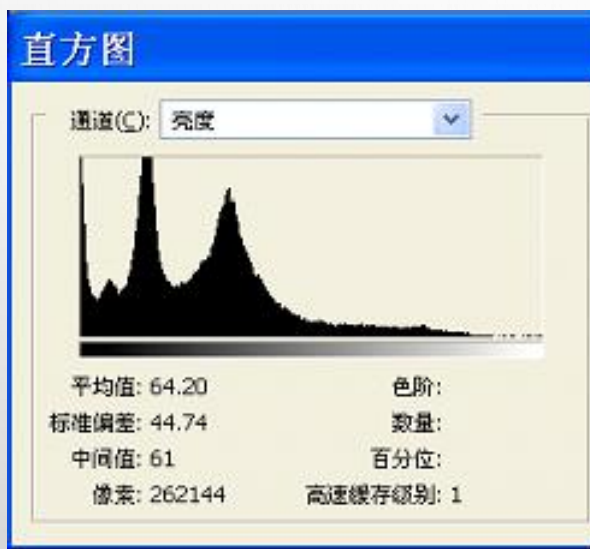
(1) T单值单增函数。

(2) 对 $0 \leq s \leq L-1$ 有 $0 \leq T(s) \leq L-1$

T满足2个条件:

(1) T单值单增函数。

(2) 对 $0 \leq s \leq L-1$ 有 $0 \leq T(s) \leq L-1$

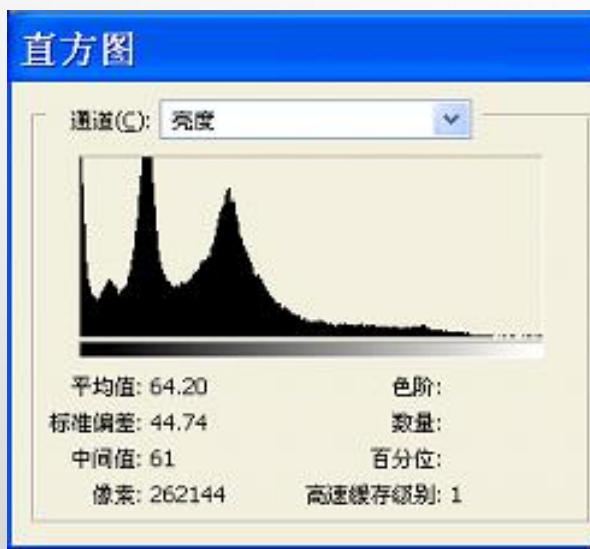


直方图均衡化

T满足2个条件:

(1) T单值单增函数。

(2) 对 $0 \leq s \leq L-1$ 有 $0 \leq T(s) \leq L-1$



直方图均衡化中T:

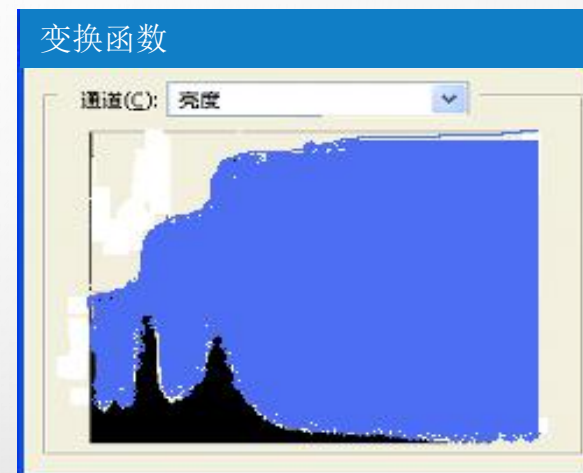
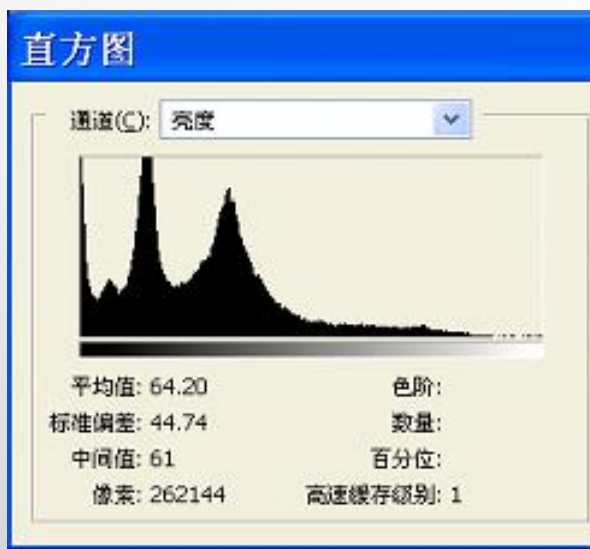
$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

直方图均衡化

T满足2个条件:

(1) T单值单增函数。

(2) 对 $0 \leq s \leq L-1$ 有 $0 \leq T(s) \leq L-1$



直方图均衡化中T:

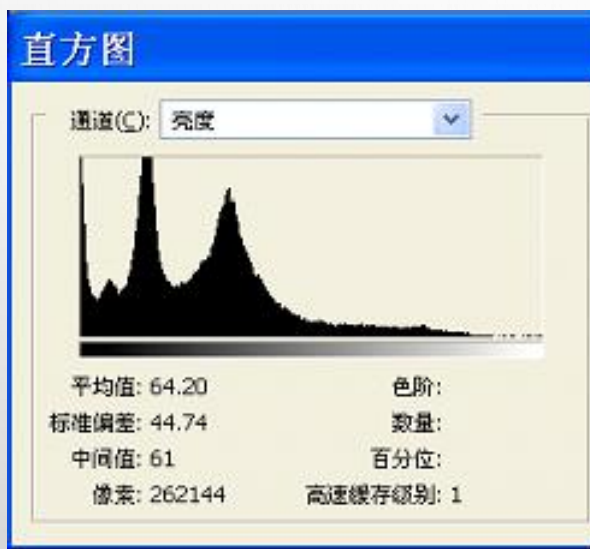
$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

直方图均衡化

T满足2个条件:

(1) T单值单增函数。

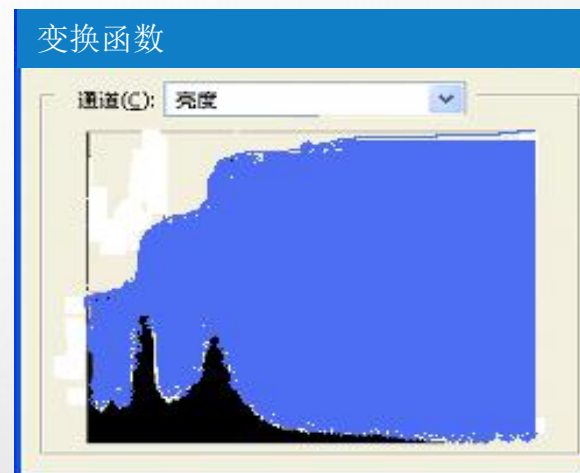
(2) 对 $0 \leq s \leq L-1$ 有 $0 \leq T(s) \leq L-1$

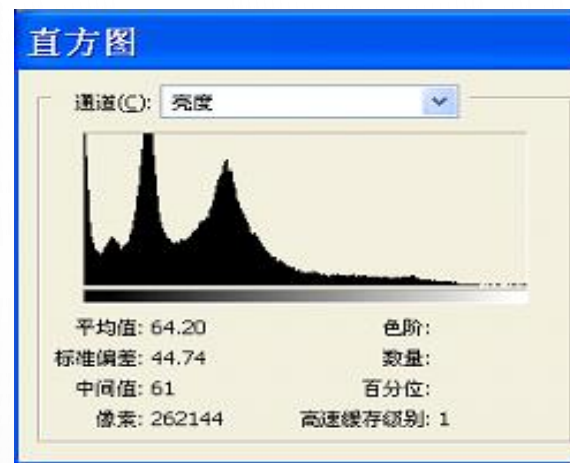


直方图均衡化中T:

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

$$t_k = \text{int}[(L-1)t_k + 0.5]$$



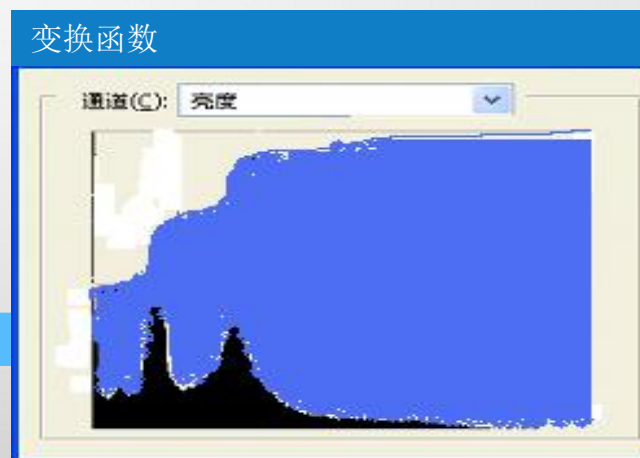


直方图均衡化中T:

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$



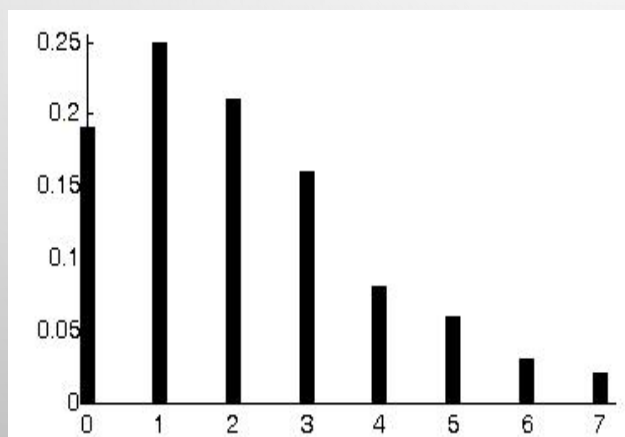
$$t_k = \text{int}[(L - 1)t_k + 0.5]$$



举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02



$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19							
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44						
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65					
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65	0.81				
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65	0.81	0.89			
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65	0.81	0.89	0.95		
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

$$t_k = T(s_k) = \sum_{i=0}^k \frac{n_i}{n} = \sum_{i=0}^k p_s(s_i)$$

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	t0	t1	t2	t3	t4	t5	t6	t7

用式 $t_k = \text{int}[(L-1)t_k + 0.5]$ 将 t_k 扩展到 $[0, L-1]$ 范围内并取整

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	1	3	5	6	6	7	7	7

用式 $t_k = \text{int}[(L-1)t_k + 0.5]$ 将 t_k 扩展到 $[0, L-1]$ 范围内并取整

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7	
像素 n_k	790	1023	890	655	329	245	122	81	
概率 $p(s_k)$	0.09	0.15	0.11	0.06	0.08	0.05	0.03	0.02	
t_k	0.09	0.04	0.05	0.01	0.09	0.05	0.08		
灰度级	1	3	5	6			7		

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	1	3	5	6		7		
	\underline{t}_0	\underline{t}_1	\underline{t}_2	\underline{t}_3		\underline{t}_4		

举例

原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	1	3	5	6		7		
	\underline{t}_0	\underline{t}_1	\underline{t}_2	\underline{t}_3		\underline{t}_4		
像素 \underline{n}_k	790	1023	850	656+329		245+122+81		

举例

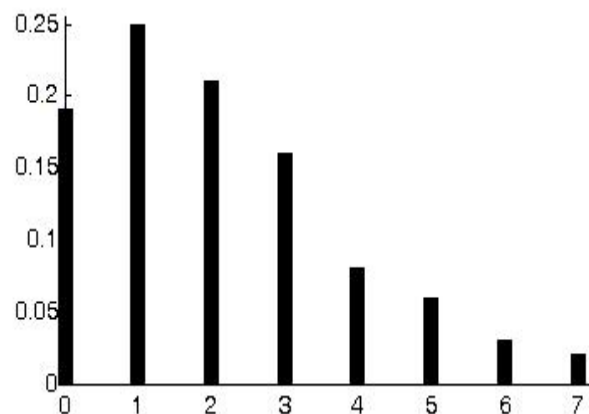
原始图像各灰度级对应的概率分布

灰度级 s_k	0	1	2	3	4	5	6	7
像素 n_k	790	1023	850	656	329	245	122	81
概率 $p(s_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

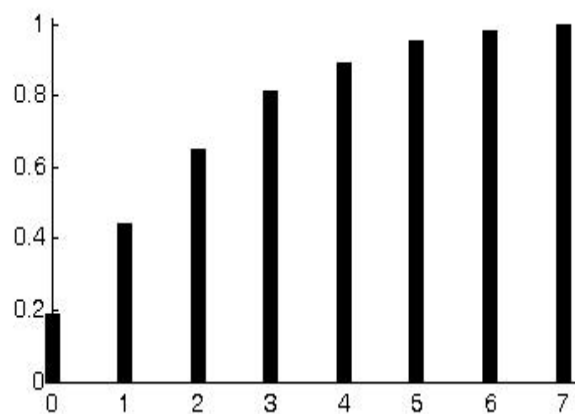
t_k	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1
灰度级	1	3	5	6	7			
	\underline{t}_0	\underline{t}_1	\underline{t}_2	\underline{t}_3	\underline{t}_4			
像素 \underline{n}_k	790	1023	850	985	448			
概率 $p(\underline{t}_k)$	0.19	0.25	0.21	0.24	0.11			

直方图均衡化

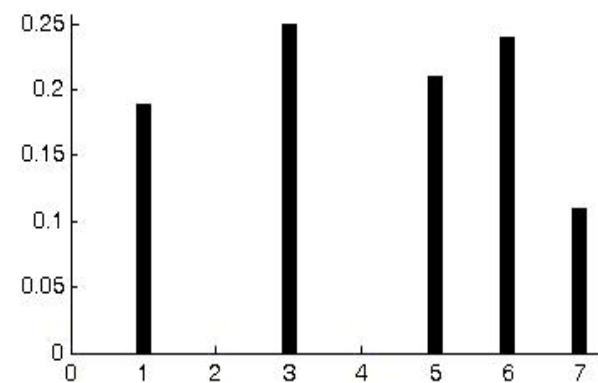
结果



原始直方图



变换函数

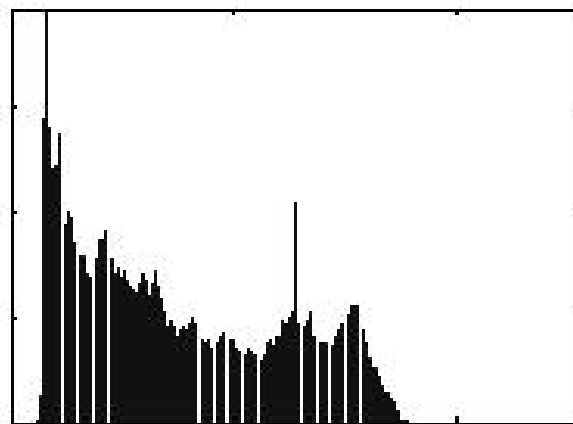
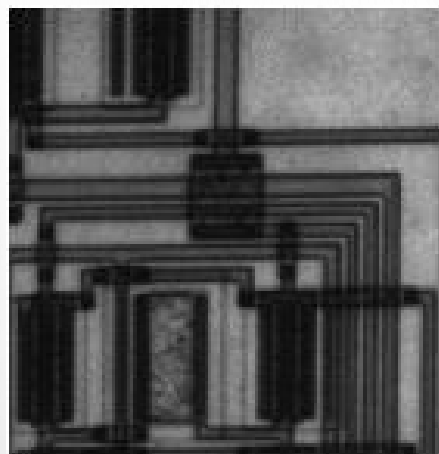


直方图均衡化结果

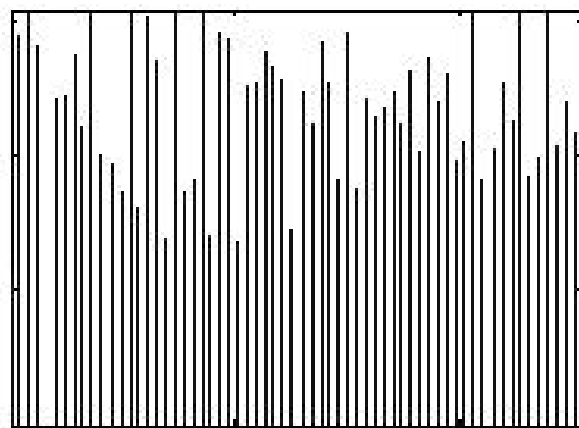
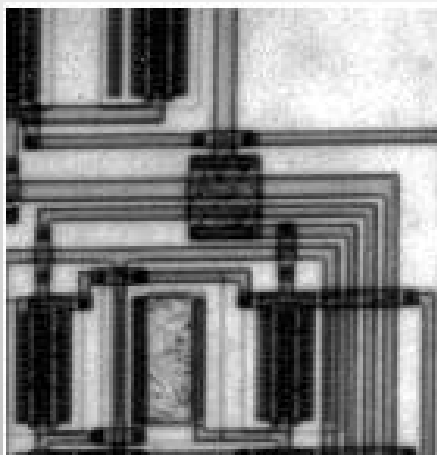
直方图均衡化

效果

原图



均衡化后效果图



程序？ 请改正大小写

- HXLBMPPFILE F;
- IF (F.LOADBMPFILE("C.BMP")) RETURN 1;
- FLOAT H[256], T[256];
- MEMSET(H,0,256*4);
- INT I,J; INT TOTAL = F.IMAGEH*F.IMAGEW;
- FOR (I =0; I<F.IMAGEH;I++)
- FOR (J =0;J<F.IMAGEW;J++)
- H[F.PDATAAT(I)[J]]+=1.0;
- FOR (I =0; I< 256;I++) H[I]/=(FLOAT) TOTAL;
-
- F.SAVEBMPFILE("2.BMP");
- RETURN 0;