P115

4

贴标签函数文件labeling.m:

function g = labeling( origF, x )

if x==4

%模板坐标相对偏移

offsets = [ -1,0;

0,-1];

%需要比较的模板个数

labelCount = 2;

expand = 1;

elseif x==8

offsets = [ -1,-1;

-1,0;

-1,1;

0,-1];

labelCount = 4;

expand = 1;

end

%数组扩展，避免边界处理

[m,n] = size(origF);

m = m+2\*expand;

n = n+2\*expand;

%产生一个边界为0的扩展数组

f = zeros(m, n, 'int8');

f(1+expand:m-expand,1+expand:n-expand) = origF(:,:);

%标号数组

g = zeros(m, n, 'int8');

%需要比较的模板的标号

labels = zeros(labelCount,1,'int8');

label=0;

N=0;

inUsedLabelBitmap = [];

for i=1+expand:m-expand

for j=1+expand:n-expand

if f(i,j)==0

continue;

end

%无连通的标志

flagNone = 1;

minLabel = label;

for k=1:labelCount

if f(i+offsets(k,1), j+offsets(k,2)) == 1

labels(k) = g(i+offsets(k,1), j+offsets(k,2));

%记录最小标签

if labels(k) < minLabel

minLabel = labels(k);

end

flagNone = 0;

else

labels(k) = 0;

end

end

%无连通

if flagNone == 1

%另起一个标号

label = label+1;

g(i,j) = label;

N = N+1;

inUsedLabelBitmap = [inUsedLabelBitmap(:,:), 1];

continue;

end

%有连通

sameLabel = 0;

flagDiff = 0;

%较大标签数组

bigLabels = [];

%处理标号

%如果标号都相同，则flagDiff为0，sameLabel为标号

%否则flagDiff为1

for k=1:labelCount

if labels(k)~=0

if sameLabel==0

sameLabel = labels(k);

elseif flagDiff==0 && sameLabel~=labels(k);

flagDiff = 1;

end

if labels(k)~=minLabel

bigLabels = [bigLabels(:,:),labels(k)];

N = N-1;

inUsedLabelBitmap(labels(k)) = 0;

end

end

end

%连通的标号都一样

if flagDiff==0

g(i,j)=sameLabel;

continue;

end

%连通的标号不一样

%使用最小的标号

g(i,j) = minLabel;

[bigCount, foo] = size(bigLabels);

%遍历前面的数

for ii = 1+expand:m-expand

for jj = 1+expand:n-expand

if ii>=i && jj>=j

break;

end

for k = 1:bigCount

%将标号改为较小的那个标号

if g(ii,jj)==bigLabels(k)

g(ii,jj) = minLabel;

break;

end

end

end

end

end

end

%序号重排

if label > N

i = 1;

rearrangeMap = zeros(label, 1, 'int8');

for j=1:label

if inUsedLabelBitmap(j)==1

rearrangeMap(j) = i;

i = i+1;

end

end

for i=1+expand:m-expand

for j=1+expand:n-expand

if g(i,j)==0

continue;

end

g(i,j) = rearrangeMap(g(i,j));

end

end

end

%去除扩展边界

g = g(1+expand:m-expand, 1+expand:n-expand);

end

腐蚀函数文件erosion.m：

function g = erosion( origF, s )

maxOffset = max(max(abs(s(:,1))),max(abs(s(:,2))));

[tmplCount, foo] = size(s);

%数组扩展，避免边界处理

expand = maxOffset;

[m,n] = size(origF);

m = m+2\*expand;

n = n+2\*expand;

%产生一个边界为0的扩展数组

f = zeros(m, n, 'int8');

f(1+expand:m-expand,1+expand:n-expand) = origF(:,:);

g = zeros(m, n, 'int8');

for i=1+expand:m-expand

for j=1+expand:n-expand

if f(i,j)==0

continue;

end

matchAll = 1;

for k=1:tmplCount

if f(i+s(k,1), j+s(k,2))==0

matchAll = 0;

break;

end

end

g(i,j) = matchAll;

end

end

%去除扩展边界

g = g(1+expand:m-expand, 1+expand:n-expand);

end

膨胀函数文件dilation.m:

function g = dilation( origF, s )

maxOffset = max(max(abs(s(:,1))),max(abs(s(:,2))));

[tmplCount, foo] = size(s);

%数组扩展，避免边界处理

expand = maxOffset;

[m,n] = size(origF);

m = m+2\*expand;

n = n+2\*expand;

%产生一个边界为0的扩展数组

f = zeros(m, n, 'int8');

f(1+expand:m-expand,1+expand:n-expand) = origF(:,:);

g = zeros(m, n, 'int8');

for i=1+expand:m-expand

for j=1+expand:n-expand

if f(i,j)==1

g(i,j)=1;

continue;

end

matchAny = 0;

for k=1:tmplCount

if f(i+s(k,1), j+s(k,2))==1

matchAny = 1;

break;

end

end

g(i,j) = matchAny;

end

end

%去除扩展边界

g = g(1+expand:m-expand, 1+expand:n-expand);

end

测试文件binaryProcessDemo.m:

f = [ 0,0,0,1,0,0,0,0;

0,1,1,1,0,0,0,0;

0,0,1,1,1,1,0,0;

0,1,0,1,1,1,1,1;

0,1,0,1,1,1,0,0;

0,1,1,0,0,1,0,0]

disp('4连通的贴标签');

g1 = labeling(f, 4)

disp('8连通的贴标签');

g2 = labeling(f, 8)

s1 = [ -1,-1; -1,0; -1,1;

0,-1; 0,1;

1,-1; 1,0; 1,1 ];

s2 = [ -2,-2; -2,-1; -2,0; -2,1; -2,-2;

-1,-2; -1,-1; -1,0; -1,1; -1,2;

0,-2; 0,-1; 0,1; 0,2;

1,-2; 1,-1; 1,0; 1,1; 1,2;

2,-2; 2,-1; 2,0; 2,1; 2,2 ];

%腐蚀

disp('3\*3腐蚀');

g3 = erosion(f,s1)

disp('5\*5腐蚀');

g4 = erosion(f,s2)

%膨胀

disp('3\*3膨胀');

g5 = dilation(f,s1)

disp('5\*5膨胀');

g6 = dilation(f,s2)

%开运算

disp('3\*3开运算');

g7 = dilation(g3,s1)

disp('5\*5开运算');

g8 = dilation(g4,s2)

%闭运算

disp('3\*3闭运算');

g9 = erosion(g5,s1)

disp('5\*5闭运算');

g10 = erosion(g6,s2)

测试结果：

>> binaryProcessDemo

f =

0 0 0 1 0 0 0 0

0 1 1 1 0 0 0 0

0 0 1 1 1 1 0 0

0 1 0 1 1 1 1 1

0 1 0 1 1 1 0 0

0 1 1 0 0 1 0 0

4连通的贴标签

g1 =

0 0 0 1 0 0 0 0

0 1 1 1 0 0 0 0

0 0 1 1 1 1 0 0

0 2 0 1 1 1 1 1

0 2 0 1 1 1 0 0

0 2 2 0 0 1 0 0

8连通的贴标签

g2 =

0 0 0 1 0 0 0 0

0 1 1 1 0 0 0 0

0 0 1 1 1 1 0 0

0 1 0 1 1 1 1 1

0 1 0 1 1 1 0 0

0 1 1 0 0 1 0 0

3\*3腐蚀

g3 =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

5\*5腐蚀

g4 =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

3\*3膨胀

g5 =

1 1 1 1 1 0 0 0

1 1 1 1 1 1 1 0

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 0

5\*5膨胀

g6 =

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

3\*3开运算

g7 =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 1 1 1 0 0

0 0 0 1 1 1 0 0

0 0 0 1 1 1 0 0

0 0 0 0 0 0 0 0

5\*5开运算

g8 =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

3\*3闭运算

g9 =

0 0 0 0 0 0 0 0

0 1 1 1 0 0 0 0

0 1 1 1 1 1 0 0

0 1 1 1 1 1 1 0

0 1 1 1 1 1 0 0

0 0 0 0 0 0 0 0

5\*5闭运算

g10 =

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 1 1 1 1 0 0

0 0 1 1 1 1 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0