

```

clear all;

close all;

% ori = imread('x1.jpg');

ori = imread('x2.jpg');

% ori = imread('x3.jpg');

% ori = imread('pollen.tif');

% ori = imread('ly_tmp.jpg');

ori = im2uint8(ori);

a = ori;

figure,imshow(a,[]);

[M, N] = size(a);

se = ones(3);

lowestBasin = min(a(:));

highestPeak = max(a(:));

index = find(a == lowestBasin);      % 找出 a 的最低海拔

b = false(size(a));

b(index) = 1;                      % 一开始的连通域

figure(1),dis_start = show_process0(a,b);      % 显示刚有水洼时的景象

con = max(max(bwlabel(b)));          % 找出出现水洼时的连通域的个数

a(index) = lowestBasin + 1;

dam = false(size(a));

dam = im2uint8(dam);                % 水坝

```

```

c = b;

% 下面开始涨水

for i = lowestBasin + 1 : 254

    b = false(size(a));

    index = find(a == i);

    b(index) = 1;

    if(max(max(bwlabel(b))) >= con)

        con = max(max(bwlabel(b)));          % 如果连通域增加或不变，则更新连通域的个数

        c = b;

    elseif(max(max(bwlabel(b))) < con)        % 如果连通域减少，则一定有水洼合并了，通过 c 返回上一步

        d = c;

        while(1)

            e = d;                            % e 是上一步的 d

            d = imdilate(d, se);              % 不断膨胀 d

            if(max(max(bwlabel(d))) < con)    % 直到其连通域减少

                break;

            end

        end

        g = bwlabel(e);                      % 回到上一步，求 e 的各个连通域

        max(max(g))

        tmp = zeros(size(a));

        tmp = im2double(tmp);

        for m = 1:con                        % 对任何一个连通域

            h = false(size(a));

```

```

ind1 = find(g==m);

h(ind1) = 1;           % 把其单独孤立开来

tmp = tmp + imdilate(h,se) - h;           % 提取其边界，加到 tmp 上

end

ind = find(tmp > 1);           % 有多个边界相交的情况，把交集提取出来，是要修水坝的地方

a(ind) = 255;

dam(ind) = 255;

p = false(size(a));

del = find(a == i);

p(del) = 1;

c = p;           % 连通域减少的情况，c 与修过水坝的 a 相关联，得到二值图

end

figure(1),dis_process = show_process1(ori,c,dam);

% figure(3)对应于 figure(1)展示其 3D 效果，其中函数 figure_rgb 与 show_process1 基本相同

figure3_rgb = figure_rgb(ori,c,dam);

figure3_gray = rgb2gray(figure3_rgb);

cmap = colormap;

xxcolormap = double(rgb2ind(figure3_rgb,cmap));

figure3_gray = double(figure3_gray);

figure(3), mesh(figure3_gray,xxcolormap);

% figure(3)对应于 figure(1)展示其 3D 效果，其中函数 figure_rgb 与 show_process1 基本相同

figure(2), imshow(process2(a,dam,i));           % 展示最终结果

```

```

% figure(4)对应于 figure(2)展示其 3D 效果

figure4_rgb = process2(a, dam, i);

figure4_gray = rgb2gray(figure4_rgb);

cmap = colormap;

xxcolormap = double(rgb2ind(figure4_rgb, cmap));

figure4_gray = double(figure4_gray);

figure(4), mesh(figure4_gray, xxcolormap);

% figure(4)对应于 figure(2)展示其 3D 效果


if(i == 254)

    ee = e - imerode(e, se);

    dam(find(ee)) = 255; % 把最终的各连通域的边界加到水坝上

    figure(2), imshow(process2(a, dam, i));

end


for u = 1 : numel(index)

    if(a(index(u)) ~= 255)

        a(index(u)) = i+1; % 把 a 中的水位抬高 1 像素

    end

end

i

con

end

figure(1), dis_final = show_process1(ori, c, dam);

```