دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف پنجم درس مبانی بینایی کامپیوتر

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

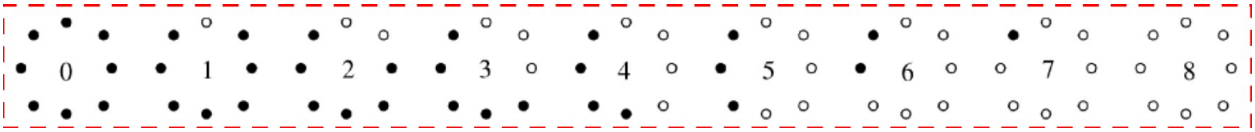
نیم سال تحصیلی: بهار ۱۴۰۰/۱۴۰۱

مدرّس: دکتر نادر کریمی

دستیاران آموزشی: بهنام ساعدی - محمدرضا مزروعی

۱

از آنجایی که این نسخه از LBP یکنواخت است تعداد transitionها در الگوی باینری به دست آمده حداکثر ۲ است. همچنین ناوردا بودن آن نسبت به چرخش باعث می‌شود که این حالت‌ها صرفا در تعداد صفر (یا یک) با یکدیگر تفاوت داشته باشند. به مثال زیر برای $LBP^{riu2}_{8,1}$ توجه کنید.



شکل ۱: الگوهای مختلف در $LBP^{riu2}_{P,R}$

همانطور که می‌بینیم تعداد بین‌ها در $LBP^{riu2}_{8,1}$ برابر با ۱۰ خواهد بود. پس به ازای $LBP^{riu2}_{P,R}$ تعداد صفرها از 0 تا P می‌تواند متغیر باشد. تعداد بین‌های هیستوگرام برابر با تعداد حالات مختلف این الگوها به علاوه‌ی یک (برای همهی الگوهای غیریکنواخت یک بین در نظر می‌گیریم) است. پس تعداد بین‌ها در هسیتوگرام نظیر آن برابر $P+2$ خواهد بود.

۲

۱.۲ الف

۱.۱.۲ Algorithm

برای شمارش سلول به صورت بازگشتی عمل می‌کنیم. به ازای هر پیکسل سفید آن را سیاه می‌کنیم و سپس الگوریتم را روی پیکسل‌های سفید در همسایگی ۳ در ۳ آن اجرا می‌کنیم. اگر همسایه‌ی سفید نداشت یکی به شمارش‌گر سلول‌ها اضافه می‌شود. این کار را تا جایی ادامه می‌دهیم که همه‌ی پیکسل‌ها پیمایش شوند (هیچ پیکسل سفیدی در تصویر باقی نماند). برای رفع مشکل یکی شمرده شدن سلول‌های به هم چسبیده از عملگر erosion با المان ساختاری دایره به شعاع ۵ استفاده می‌کنیم.

۲.۱.۲ Function

```
function number_of_cells = countCells(I)
%COUNTCELLS Summary of this function goes here
%   Detailed explanation goes here
    J = I;
    number_of_cells = 0;
    for i = 1: size(I, 1)
        for j = 1: size(I, 2)
            if (J(i, j) == 1)
                number_of_cells = number_of_cells + 1;
                J = countRemoveCell(i, j, J, number_of_cells);
            end
```

```
        end
    end
end

function J = countRemoveCell(i, j, I, id)
%COUNTREMOVECELL Summary of this function goes here
%   Detailed explanation goes her
    J = I;
    if (J(i, j) == 1)
        J(i, j) = 0;
        if (i - 1 >= 1)
            J = countRemoveCell(i - 1, j, J, id);
            if (j - 1 >= 1)
                J = countRemoveCell(i - 1, j - 1, J, id);
                J = countRemoveCell(i, j - 1, J, id);
            end
            if (j + 1 <= size(I, 2))
                J = countRemoveCell(i - 1, j + 1, J, id);
                J = countRemoveCell(i, j + 1, J, id);
            end
        end
        if (i + 1 <= size(I, 1))
            J = countRemoveCell(i + 1, j, J, id);
            if (j - 1 >= 1)
                J = countRemoveCell(i + 1, j - 1, J, id);
            end
            if (j + 1 <= size(I, 2))
                J = countRemoveCell(i + 1, j + 1, J, id);
            end
        end
    end
end
```

Driver code ۳.۱.۲

```
clc
clear
```

```
close all
imtool close all
%%%%%%%%%%%%%%%%%
I = imread("images\Q2\Cells.tif");
level = multithresh(I, 2);
J = double(I >= level(1));
%figure, imshow(J, []);
se = strel('disk', 5);
K = imerode(J, se);
%figure, imshow(K, []);
x = countCells(J)
y = max(max(bwlabel(J)))
z = countCells(K)
w = max(max(bwlabel(K)))
```

۲.۲ ب

۱.۲.۲ Algorithm

برای برچسب‌گذاری سلول‌ها و نهایتا محسابه‌ی مساحت و میانگین سطح روشنایی آن‌ها به صورت بازگشتی عمل می‌کنیم. یک map (ماتریس هم‌اندازه با تصویر ورودی) نظیر تصویر ورودی که درواقع برچسب هر پیکسل (که در کدام ناحیه قرار دارد) می‌سازیم. به ازای هر پیکسل سفید در map پیکسل نظیر آن را برچسب می‌دهیم و آن را سیاه می‌کنیم و سپس الگوریتم را روی پیکسل‌های سفید در همسایگی ۳ در ۳ آن اجرا می‌کنیم. اگر همسایه‌ی سفید نداشت یکی به شمارش‌گر ناحیه اضافه می‌شود. این کار را تا جایی ادامه می‌دهیم که همه‌ی پیکسل‌ها پیمایش شوند (هیچ پیکسل سفیدی در تصویر باقی نماند).

۲.۲.۲ Function

```
function saveCellsData(I, filename, path)
%SAVECELLSDATA Summary of this function goes here
%   Detailed explanation goes here
    level = multithresh(I, 4);
    J = double(I >= level(1));
    x = labelCells(J);
    cells_data = zeros(3, max(max(x)));
    cells_data(1, :) = 1: max(max(x));
    for i = 1: max(max(x))
        cells_data(2, i) = sum(sum(x == i));
        cells_data(3, i) = sum(sum((uint8(x == i) .* I))) / cells_data(2, i);
    end
```

```
    writematrix(transpose(cells_data), [path '/' filename]);
end

function map = labelCells(I)
%LABELCELLS Summary of this function goes here
%   Detailed explanation goes here
    map = uint32(zeros(size(I)));
    id = 0;
    for i = 1: size(I, 1)
        for j = 1: size(I, 2)
            if (I(i, j) == 1)
                id = id + 1;
                [I, map] = labelRemoveCell(i, j, I, id, map);
            end
        end
    end
end

function [J, new_map] = labelRemoveCell(i, j, I, id, map)
%labelRemoveCell Summary of this function goes here
%   Detailed explanation goes her
    J = I;
    new_map = map;
    if (J(i, j) == 1)
        J(i, j) = 0;
        new_map(i, j) = id;
        if (i - 1 >= 1)
            [J, new_map] = labelRemoveCell(i - 1, j, J, id, new_map);
            if (j - 1 >= 1)
                [J, new_map] = labelRemoveCell(i - 1, j - 1, J, id, new_map);
                [J, new_map] = labelRemoveCell(i, j - 1, J, id, new_map);
            end
            if (j + 1 <= size(I, 2))
                [J, new_map] = labelRemoveCell(i - 1, j + 1, J, id, new_map);
                [J, new_map] = labelRemoveCell(i, j + 1, J, id, new_map);
            end
```

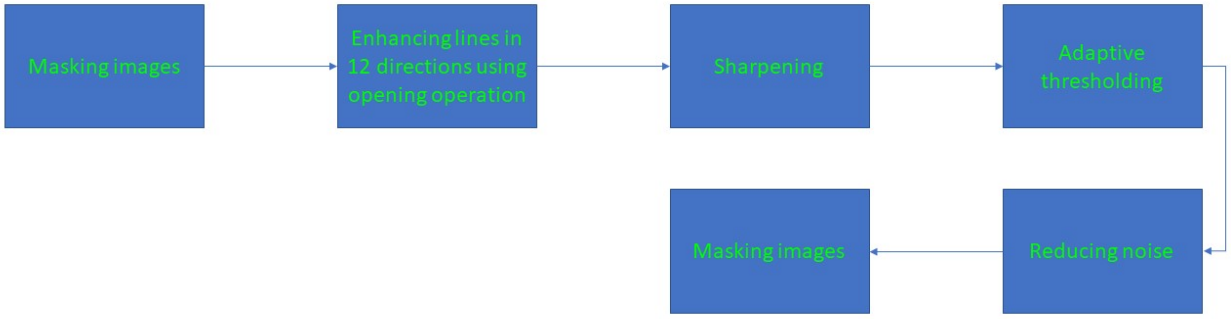
```
        end
        if (i + 1 <= size(I, 1))
            [J, new_map] = labelRemoveCell(i + 1, j, J, id, new_map);
            if (j - 1 >= 1)
                [J, new_map] = labelRemoveCell(i + 1, j - 1, J, id, new_map);
            end
            if (j + 1 <= size(I, 2))
                [J, new_map] = labelRemoveCell(i + 1, j + 1, J, id, new_map);
            end
        end
    end
end
```

۳.۲.۲ Driver code

```
clc
clear
close all
imtool close all
%%%%%%%%%%%%%%%%%
I = imread("images\Q2\Cells.tif");
I = I(1: size(I, 1) - 1, :);
saveCellsData(I, 'cells_data.xls', '.');
```

۳

۱.۳ Block diagram



شکل ۲: بلوک‌دیاگرام روش پیاده‌سازی شده

۲.۳ Algorithm

ابتدا ماسک تصاویر را اعمال می‌کنیم تا حاشیه‌ی آن‌ها کاملا سیاه شود. سپس به وسیله‌ی عملگر opening و با المان ساختاری خط به طول ۷، خطوط را در ۱۲ جهت مختلف تقویت می‌کنیم تا هم رگ‌های ریز نمایان‌تر شوند و هم نویزهای غیرخطی گرفته شوند. سپس تصویر را به وسیله‌ی unsharp masking شارپ می‌کنیم. با استفاده از روش آستانه‌گذاری وفقی تصویر را به تصویر سیاه‌سفید تبدیل می‌کنیم. جهت تقویت رگ‌ها روی تصویر عملگر dilation با المان ساختاری خط به طول ۶ در راستای افقی اعمال می‌کنیم. در نهایت به وسیله‌ی فیلتر میانه نویز را کاهش می‌دهیم.

۳.۳ Driver code

```
clc
clear
close all
imtool close all
%%%%%%%%%%%%%%%%%
n = 20;
path = 'images\Q3\DRIVE\Test\';
```

```
temp1 = imread([path 'images\1_test.tif']);
temp2 = imread([path '2nd_manual\1_manual2.gif']);
images = uint8(zeros([n size(temp1)]));
first_manual = uint8(zeros([n size(temp2)]));
second_manual = uint8(zeros([n size(temp2)]));
mask = uint8(zeros([n size(temp2)]));
for i = 1: n
    images(i, :, :, :) = imread([path 'images\' num2str(i) '_test.tif']);
    first_manual(i, :, :, :) = imread([path '1st_manual\' num2str(i) '_manual1.gif']);
    second_manual(i, :, :, :) = imread([path '2nd_manual\' num2str(i) '_manual2.gif']);
    mask(i, :, :, :) = imread([path 'mask\' num2str(i) '_test_mask.gif']);
end
%%%%%%%%%%%%%%%%%%%%%%%%%mask
masked_images = images;
for i = 1: n
    for j = 1: size(masked_images(i, :, :, :), 2)
        for k = 1: size(masked_images(i, :, :, :), 3)
            if (mask(i, j, k) == 0)
                masked_images(i, j, k, 1) = 0;
                masked_images(i, j, k, 2) = 0;
                masked_images(i, j, k, 3) = 0;
            end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%enhancing lines in 12 directions using opening operation
linearly_opened_images = masked_images;
for i = 1: n
    for j = 1: 15: 180
        se = strel('line', 7, j);
        linearly_opened_images(i, :, :, :) = imopen(squeeze(linearly_opened_images(i, :,
    :, :)), se);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%sharpening
sharped_images = linearly_opened_images;
```

```
for i = 1: n
    sharped_images(i, :, :, :) = imsharpen(squeeze(sharped_images(i, :, :, :)), 'Radius',
     1, 'Amount', 5);
end
%%%%%%%%%%%%%%%%%%%%%%%%%adaptive thresholding
thresh_images = sharped_images;
for i = 1: n
    S = imfilter(squeeze(thresh_images(i, :, :, :)), fspecial('average', [21 21]));
    K = 0.93;
    T = K * S;
    thresh_images(i, :, :, :) = squeeze(thresh_images(i, :, :, :)) < T;
end
%%%%%%%%%%%%%%%%%%%%%%%%%reduce noise
noise_canceled_images = thresh_images;
se = strel('line', 6, 0);
for i = 1: n
    noise_canceled_images(i, :, :, :) = imdilate(squeeze(noise_canceled_images(i, :, :,
    :)), se);
end
%
for i = 1: n
    noise_canceled_images(i, :, :, 1) = medfilt2(squeeze(noise_canceled_images(i, :, :,
    1)), [3 3]);
    noise_canceled_images(i, :, :, 2) = medfilt2(squeeze(noise_canceled_images(i, :, :,
    2)), [3 3]);
    noise_canceled_images(i, :, :, 3) = medfilt2(squeeze(noise_canceled_images(i, :, :,
    3)), [3 3]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%mask
final_images = noise_canceled_images;
for i = 1: n
    for j = 1: size(final_images(i, :, :, :), 2)
        for k = 1: size(final_images(i, :, :, :), 3)
            if (mask(i, j, k) == 0)
                final_images(i, j, k, 1) = 0;
                final_images(i, j, k, 2) = 0;
```

```
                final_images(i, j, k, 3) = 0;
            end
        end
    end
    %imwrite(imadjust(squeeze(final_images(i, :, :, 2))), ['images\Q3_results\1\' num2str
    (i) '.tif']);
    imwrite(imadjust(squeeze(final_images(i, :, :, 2))), ['images\Q3_results\2\' num2str(
    i) '.tif']);
end

%%%%%%%%%%%%%%%%%%%%%%%%calc parameters
sensitivity = double(zeros([1 n]));
specificity = double(zeros([1 n]));
accuracy = double(zeros([1 n]));
for i = 1: n
    [tp, tn, fp, fn] = calcParameters(squeeze(final_images(i, :, :, 2)), squeeze(
    second_manual(i, :, :)));
    sensitivity(i) = double(tp / (tp + fn));
    specificity(i) = double(tn / (tn + fp));
    accuracy(i) = double((tp + tn) / (tp + tn + fp + fn));
end
sen_mean = mean(sensitivity)
spe_mean = mean(specificity)
acc_mean = mean(accuracy)
sensitivity;
specificity;
accuracy;
```

۴.۳ Results

۱.۴.۳ پزشک اول

Accuracy	Specificity	Sensitivity	
90.66	91.25	84.63	**1**
91.25	91.98	84.91	**2**
88.82	89.56	82.08	**3**
89.65	90.14	84.80	**4**
90.12	91.06	81.07	**5**
90.84	92.45	75.93	**6**
89.69	90.50	81.64	**7**
89.99	91.07	78.50	**8**
92.22	94.13	70.59	**9**
88.65	89.21	82.50	**10**
89.73	90.65	80.45	**11**
90.03	90.84	81.42	**12**
90.69	92.19	76.80	**13**
89.47	89.75	86.25	**14**
80.02	79.03	92.87	**15**
90.67	0.9191	78.25	**16**
92.08	93.96	71.68	**17**
91.30	92.43	78.18	**18**
87.27	87.09	89.26	**19**
90.66	91.23	83.40	**20**
89.69	**90.52**	**81.26**	**Average**

۲.۴.۳ پزشک دوم

Accuracy	Specificity	Sensitivity	
90.68	91.18	85.45	**1**
91.48	92.03	86.62	**2**
88.93	89.16	86.58	**3**
89.50	89.86	85.72	**4**
90.35	90.63	87.24	**5**
90.54	92.09	75.51	**6**
89.26	89.41	87.38	**7**
89.71	90.07	84.71	**8**
92.02	94.01	69.42	**9**
88.63	88.75	87.12	**10**
90.07	90.55	84.81	**11**
90.04	90.58	83.86	**12**
90.56	92.28	75.24	**13**
89.34	89.42	88.36	**14**
80.20	79.22	92.36	**15**
91.12	91.91	82.60	**16**
92.69	93.79	78.84	**17**
92.10	93.46	78.63	**18**
87.97	88.15	86.35	**19**
90.72	92.13	76.85	**20**
89.80	**83.18**	**83.18**	**Average**

منابع