



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف ششم درس مبانی بینایی کامپیوتر

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۱/۱۴۰۰

مدرس: دکتر نادر کریمی

دستیاران آموزشی: بهنام ساعدی - محمدرضا مزروعی

۱

فرض می‌کنیم که نقاط $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_k, y_k)$ در فضای تصویر روی خط $y = mx + c$ قرار دارند. معادله خط نظیر هر یک از این نقاط در فضای هاف به ترتیب برابر $c = -x_1m + y_1, c = -x_2m + y_2, c = -x_3m + y_3, \dots, c = -x_km + y_k$ است. به عبارتی نقطه‌ی (m, c) روی تمامی خطوط با شیب $-x_i$ و عرض از مبدا y_i در فضای هاف $(m-c)$ قرار دارد. از این رو همه‌ی این خطوط در نقطه‌ی (m, c) هم‌رس‌اند.

۲

شیب یک خط در فضای تصویر در بازه‌ی $(-\infty, \infty)$ است. پس $m \in (-\infty, \infty)$. همچنین عرض از مبدا خط هم در فضای تصویر در بازه‌ی $(-\infty, \infty)$ است. پس $c \in (-\infty, \infty)$.

۳

هر دایره در فضای $x - y$ با ۳ پارامتر مختصات مرکز و شعاع به صورت متمایز مشخص می‌شود. در نتیجه فضای هاف ۳ بعدی است و هر نقطه در فضای هاف نظیر یک دایره در فضای $x - y$ است. همچنین دایره‌های نظیر هر دو نقطه‌ای که در فضای $x - y$ روی یک دایره قرار دارند، در فضای هاف از یک نقطه می‌گذرند که مختصات این نقطه (صرف نظر از مولفه‌ی نظیر شعاع) مختصات مرکز همان دایره در فضای $x - y$ است. در نهایت نظیر همه‌ی دایره‌ها (شعاع‌های دلخواه)ی که در فضای $x - y$ از یک نقطه می‌گذرند یک مخروط در فضای هاف تشکیل می‌شود. با استفاده از گسسته‌سازی فضا و مکانیزم انباشت‌گر (Accumulator) به طریق الگوریتم زیر عمل می‌کنیم.

۱. تقسیم‌بندی فضای پارامترها یا H به تعدادی سلول بر اساس دقت مورد انتظار با مقدار اولیه صفر

۲. به ازای هر نقطه (پیکسل سفید) با مختصات (x_i, y_i) در صفحه تصویر

(آ) به ازای تمام مقادیر r بین r_{min} تا r_{max}

□ محاسبه‌ی مقدار $H(x_j, y_j, r) = H(x_j, y_j, r) + 1$ به ازای تمام نقاط مثل (x_j, y_j) که روی دایره‌ی به مرکز

(x_i, y_i) و شعاع r قرار گرفته‌اند

۳. آنالیز داده‌ها در فضای هاف برای یافتن نقاط مطلوب (در واقع پیدا کردن مختصات نقطه‌ای که بیشترین H را دارد)

۴. به دست آمدن مرکز دایره‌ی متناظر

۴

Algorithm ۱.۴

همانطور که در سوال قبل توضیح داده شد الگوریتم به شکل زیر است.

۱. تقسیم‌بندی فضای پارامترها یا $centers$ به تعدادی سلول بر اساس دقت مورد انتظار با مقدار اولیه صفر

۲. به ازای هر نقطه (پیکسل سفید) با مختصات (i, j) در صفحه تصویر

(آ) به ازای تمام مقادیر r بین 40 تا 50

□ محاسبه‌ی مقدار $centers(k, l, radius) = centers(k, l, radius) + 1$ به ازای تمام نقاط مثل (k, l) که روی

دایره‌ی به مرکز (i, j) و شعاع $radius$ قرار گرفته‌اند

۳. آنالیز داده‌ها در فضای هاف برای یافتن نقاط مطلوب (در واقع پیدا کردن اندیس عنصر ماکسیمم آرایه‌ی $centers$)

۴. به دست آمدن مرکز/مراکز دایره/دایره‌های متناظر

توجه شود که برای سادگی در پیاده‌سازی، تابع `findCenters` در هر گام به ازای یک $radius$ خاص آرایه‌ی دو بعدی $centers$ را محاسبه می‌کند و سپس به آرایه‌ی ۳ بعدی `append` می‌شود.

۲.۴ Function

```

1 function centers = findCenters(I, radius)
2 %FINDCENTERS Summary of this function goes here
3 % Detailed explanation goes here
4 centers = double(zeros(size(I)));
5 for i = 1: size(I, 1)
6     for j = 1: size(I, 2)
7         if (i == 217 && j == 80)
8             x = 0;
9         end
10        if (I(i, j) ~= 0)
11            centers = updateCenters(I, centers, i, j, radius);
12        end
13    end
14 end
15 end
16
17 function new_centers = updateCenters(I, centers, i, j, radius)
18 %UPDATECENTERS Summary of this function goes here
19 % Detailed explanation goes here
20 new_centers = centers;
21 for k = i - radius - 1: i + radius + 1
22     for l = j - radius - 1: j + radius + 1
23         if ((k >= 1 && k <= size(I, 1)) && (l >= 1 && l <= size(I, 2)))
24             if (I(i, j) ~= 0 && round((double(i) - double(k)) ^ 2 + (double(j) - double(l)
                )) ^ 2) == radius ^ 2)

```

```

25         new_centers(k, 1) = new_centers(k, 1) + 1;
26     end
27 end
28 end
29 end
30 end

```

Driver code ۳.۴

```

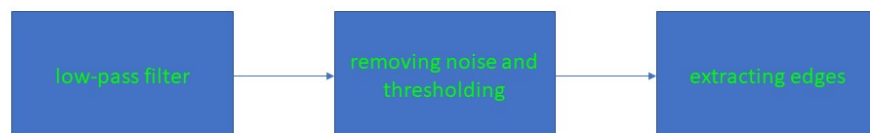
1  clc
2  clear
3  close all
4  imtool close all
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  min_radius = 30;
7  max_radius = 50;
8  thresh = 15;
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%1_test.tif
10 I = imread("images\Q4\DRIVE\Test\images\1_test.tif");
11 %%%%%%%%%low-pass filter
12 lowed_I = I;
13 lowed_I = imfilter(lowed_I, fspecial('average', 5));
14 %%%%%%%%%removing noise and thresholding
15 level = multithresh(lowed_I(:, :, 1), thresh);
16 lowed_I = imopen(lowed_I, strel('disk', 7));
17 lowed_I = imclose(lowed_I, strel('disk', 7));
18 K = lowed_I(:, :, 1) > level(thresh);
19 %%%%%%%%%extracting edges
20 K = imgradient(K);
21 %%%%%%%%%finding centers
22 centers = double(zeros([size(I, 1) size(I, 2) max_radius]));
23 for radius = min_radius: max_radius
24     centers(:, :, radius) = findCenters(K, radius);
25 end
26 max_centers = find(centers == max(centers(:)));
27 [x_i y_i r_i] = ind2sub(size(centers), max_centers);
28 RGB_i = insertShape(I, 'circle', [round(mean(y_i)) round(mean(x_i)) round(mean(r_i))], '

```

```

        LineWidth', 2);
29 figure, imshow(I, []);
30 figure, imshow(RGB_i, []);
31 [round(mean(y_i)) round(mean(x_i))]
32
33 %%%%%%%%%25_training.tif
34 J = imread("images\Q4\DRIVE\Training\images\25_training.tif");
35 %%%%%%%%%low-pass filter
36 lowed_J = J;
37 lowed_J = imfilter(lowed_J, fspecial('average', 5));
38 %%%%%%%%%removing noise and thresholding
39 level = multithresh(lowed_J(:, :, 1), thresh);
40 lowed_J = imopen(lowed_J, strel('disk', 7));
41 lowed_J = imclose(lowed_J, strel('disk', 7));
42 L = lowed_J(:, :, 1) > level(thresh);
43 %%%%%%%%%extracting edges
44 L = imgradient(L);
45 %%%%%%%%%finding centers
46 centers = double(zeros([size(J, 1) size(J, 2) max_radius]));
47 for radius = min_radius: max_radius
48     centers(:, :, radius) = findCenters(L, radius);
49 end
50 max_centers = find(centers == max(centers(:)));
51 [x_j y_j r_j] = ind2sub(size(centers), max_centers);
52 RGB_j = insertShape(J, 'circle', [round(mean(y_j)) round(mean(x_j)) round(mean(r_j))], '
        LineWidth', 2);
53 figure, imshow(J, []);
54 figure, imshow(RGB_j, []);
55 [round(mean(y_j)) round(mean(x_j))]

```

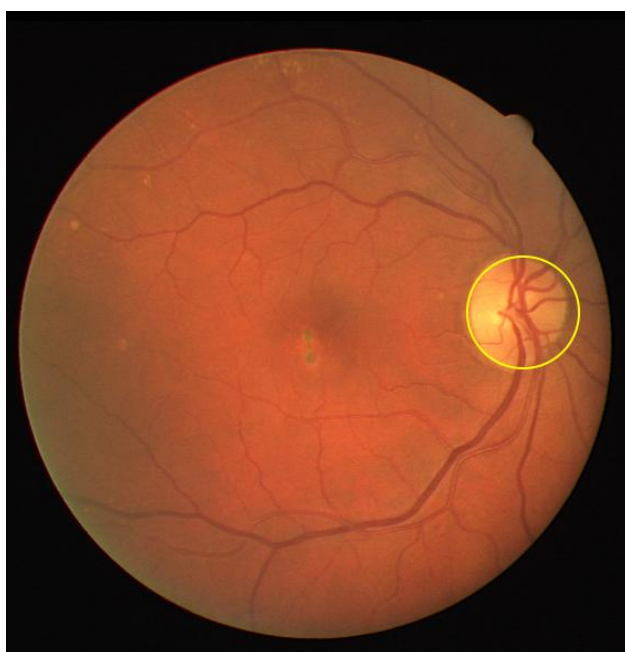
Block diagram ۴.۴

شکل ۱: بلوک‌دیگرام پیش‌پردازش

۵.۴ Output



شکل ۲



شکل ۳

منابع