به نام خدا

پردازش تصویر

تمرین شماره ۵

بخش بندی وثبت تصاویر

امین سخایی

۹۷۳۳۰۳۶

استاد درس

دکتر حامد آذرنوش

سوال شماره ۱:

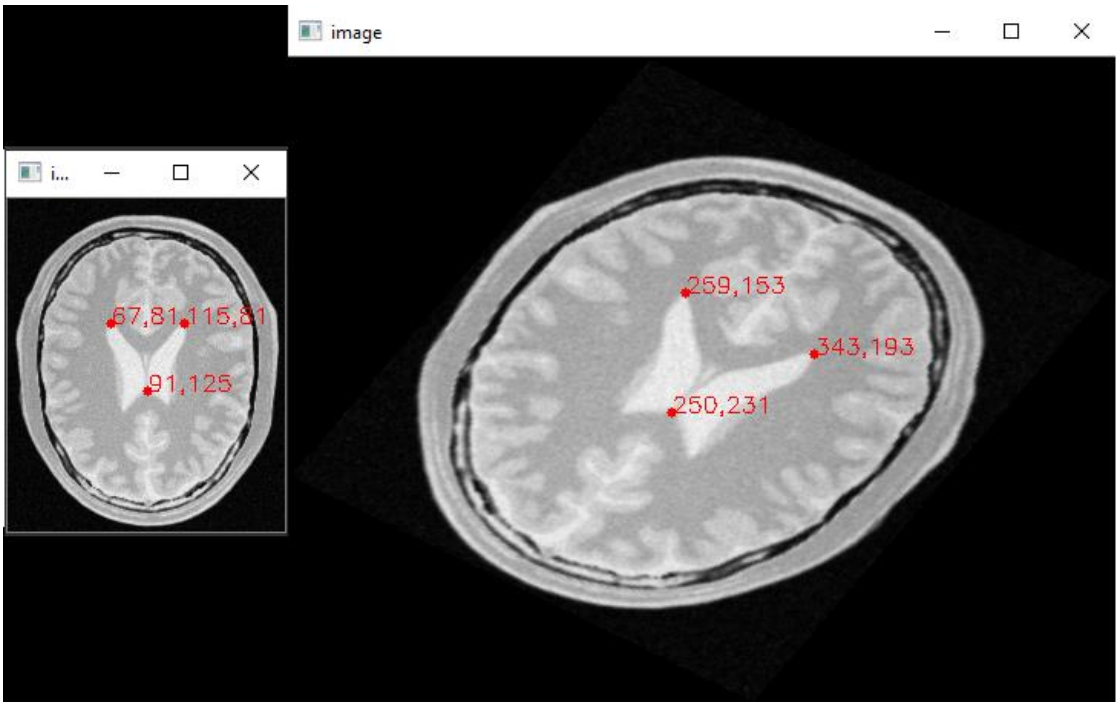
تصویر MRIF با تبدیل Affine به تصویر MRIS تبدیل شده است.برای register کردن تصویر دوم روی اول با استفاده از Feature based Registration باید نقاطی را از تصویر تصویر اول استخراج کرده و بیابیم تحت چه تبدیلی به تصویر دوم نگاشت شده است.ماتریس تبدیل Affine به شکل زیر است:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{aligned} x'_i &= a_{11}\, x_i + a_{12}\, y_i + a_{13} \\ y'_i &= a_{21}\, x_i + a_{22}\, y_i + a_{23} \end{aligned}$$

۲ معادله و ۶ مجهول در اختیار داریم بنابراین با استخراج سه نقطه متناظر در دو تصویر و حل ۶ معادله و ۶ مجهول ضرایب تبدیل را بدست می آوریم.

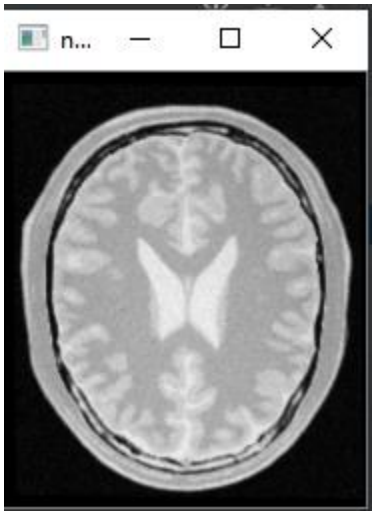
مختصات انتخاب شده در دو تصویر:



با حل معادلات تابع تبدیل به صورت زیر به دست می آید:

```
[[  0.40277778   0.35416667 -91.50694444]
 [ -0.25462963   0.53472222  65.13657407]]
```

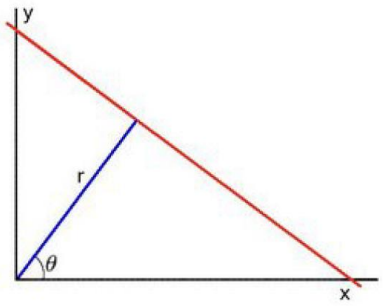
خروجی برنامه:



سوال شماره ۳:

بخش آ) تبدیل هاف:

هر خط را می توان در دو مختصات کارتزین و قطبی نمایش داد.



به دلیل اینکه در مختصات دکارتی مقدار شیب خط می تواند مقادیر بین $(-\infty, \infty)$ را به خود اختصاص دهد و در تبدیل هاف باید مقادیر پارامتر ها محدود باشد از مختصات قطبی در این تبدیل استفاده می کنیم. بنابراین معادله خط را می توان به شکل زیر نوشت:

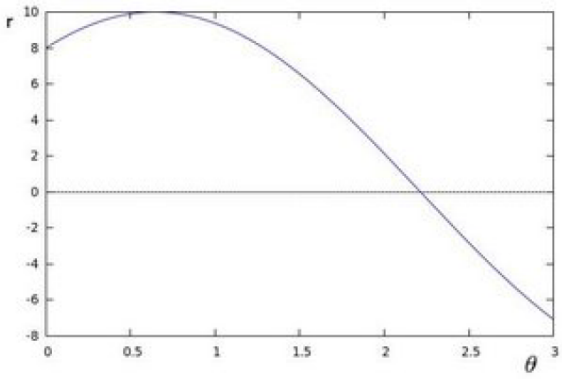
$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

$$r = x\cos\theta + y\sin\theta$$

به طور کلی، به ازای هر نقطه به صورت$(x_.,y_.)$ می توانیم خانواده خطوطی که از آن نقطه می گذرند را به صورت زیر تعریف کرد:

$$r = x_.\cos\theta + y_.\sin\theta$$

هر جفت (r_θ, θ)، نشان دهنده یک خط است که از نقطه $(x_.,y_.)$ می گذرد. اگر خانواده تمام خطوطی که از نقطه $(x_.,y_.)$ می گذرند را در صفحه $\theta - r$ رسم کنیم، یک شکل سینوسی بوجود می آید. برای مثال برای $x_. = ۸, y_. = ۶$شکل زیر بوجود می آید:



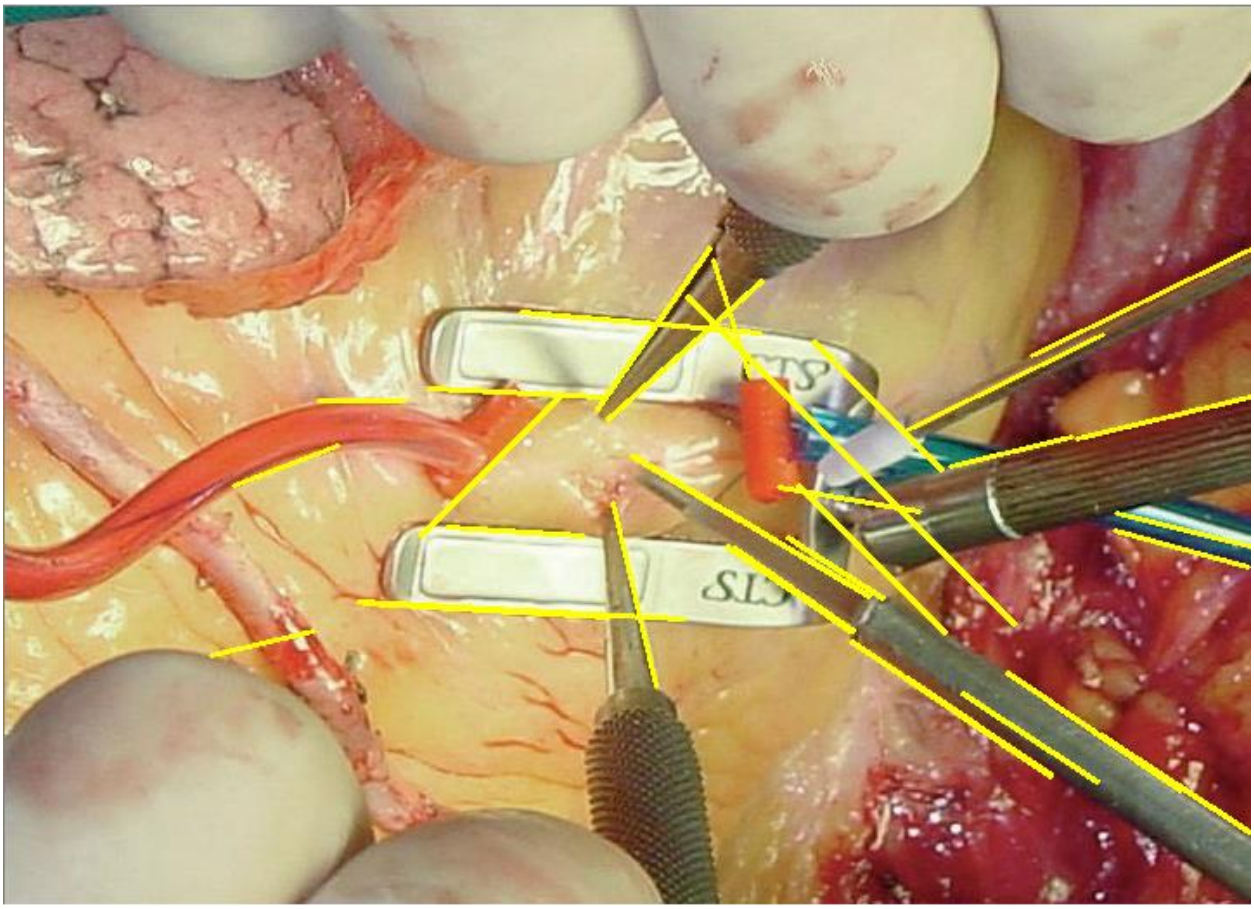
مشابه کار بالا را می توان برای تمام نقاط موجود در یک تصویر انجام داد.

به صورت کلی می توان یک خط را با به دست آوردن نقاط برخورد بین منحنی ها کشف کرد. هرچه تعداد بیشتری منحنی در یک برخورد وجود داشته باشد، به این معنی است که خطی که نشان دهنده آن برخورد است، تعداد بیشتری نقطه دارد. تبدیل هاف تعداد برخورد های منحنی همه ی نقاط تصویر را بدست می آورد و اگر تعداد برخورد ها بیشتر از آستانه تعیین شده باشد، آن را به عنوان خط با پارامترهای (θ, r_θ) در نظر می گیرد.

برای کشف خطوط ابتدا فیلتر میانه (به منظور کاهش نویز و جلوگیری از خطا) بر تصویر اعمال کرده و لبه های آنرا استخراج می کنیم سپس تبدیل خط هاف را بر آن اعمال می کنیم.

در این مسئله از cv.HoughLinesP استفاده کردیم که بسیار بهینه تر از cv.HoughLines می باشد و به عنوان خروجی به جای بردار (θ, r_θ) نقاط انتهایی خطوط کشف شده $(x_., y_., x_1, y_1)$ را بر می گرداند.

خروجی:



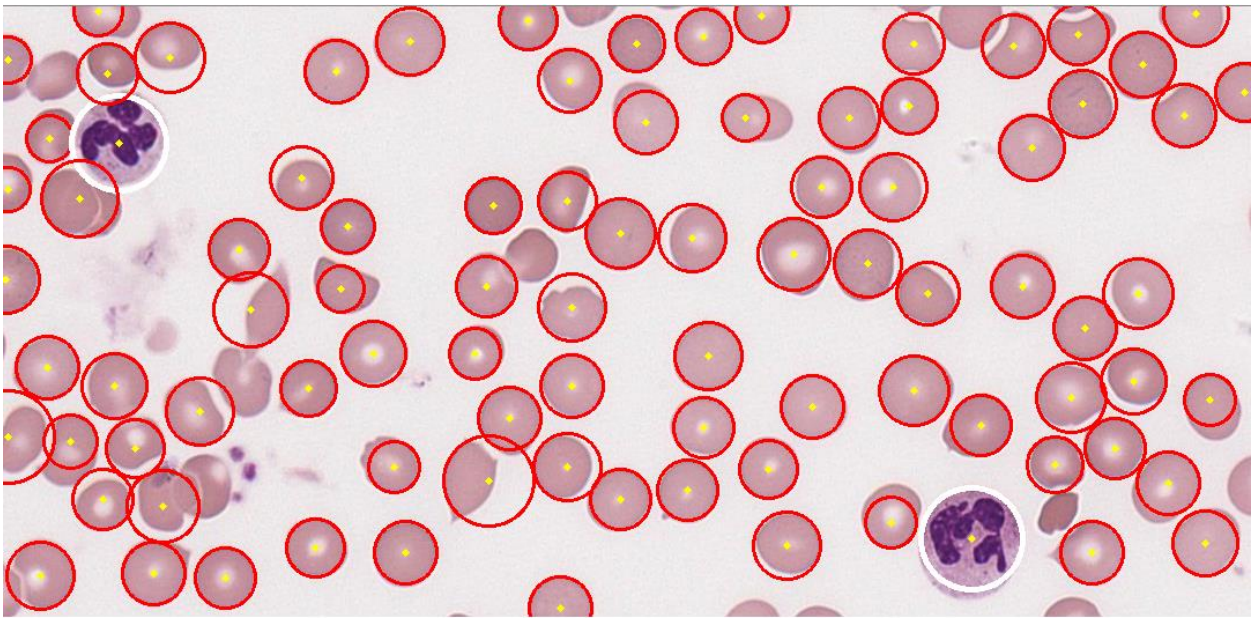
$\text{threshold} = ۸۵, \text{minLineLenght} = ۵۰, \text{maxLineGap} = ۲۰$

بخش ب) تبدیل دایره هاف بسیار مشابه تبدیل خط هاف است اما در تشخیص خط به دو پارامتر (r, θ) اما در تشخیص دایره به سه پارامتر $(x_{center}, y_{center}, r)$ نیاز داریم که (x_{center}, y_{center}) مختصات مرکز و r شعاع دایره می باشد.

برای کشف دایره ها ابتدا فیلتر میانه (به منظور کاهش نویز و جلوگیری از خطا) و سپس تبدیل دایره هاف را بر تصویر اعمال می کنیم. (به منظور بهینه ساز در opencv از روش گرادیان هاف به جای روش استاندارد استفاده شده است.)

در انتها از یک دستور شرطی و مقایسه با شعاع ۴۰،دور گلبول های قرمز دایره قرمز و دور گلبول های سفید دایره سفید می کشیم.

خروجی:



$param1 = ۲۴۵, param2 = ۱۳, minRadius = ۰, maxRadius = ۴۵$

تعداد سلول ها:

```
White blood cells =  2
Red blood cells =  87
```

سوال شماره ۴:

- Sobel:

دو ماسک عمودی و افقی را روی تصویر اعمال می کند.این الگوریتم برای لبه یابی عمودی مناسب می باشد.

−1	−2	−1	−1	0	1
0	0	0	−2	0	2
1	2	1	−1	0	1

Sobel

- Prewitt:

همانند sobel می باشد و فقط در ضرایب ماسک متفاوت اند.

−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Prewitt

- LoG:

ابتدا تصویر با فیلتر گووسی هموارمی‌شود سپس با عبور از فیلتر لاپلاسین لبه ها در تصویر آشکار می‌شود. به خودی خود اثر LoG هایلایت کردن لبه ها می باشد.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

- Canny:

الگوریتم Canny یک سطح آستانه دارد که این آستانه تفاضل سطح شدت ها می باشد.

در هر قسمتی که شدت روشنایی کم باشد،لبه یابی ضعیف و هر قسمتی که شدت روشنایی زیاد باشد لبه یابی بهتر و قوی تر انجام می شود.زمانی که نیاز به لبه یابی قوی باشد با در نظر گرفتن شیب ها از canny استفاده می کنیم.

Canny سه سطح آستانه مختلف برای شدت روشنایی ها دارد:

اگر اختلاف سطح شدت از آستانه اول بیشتر باشد آن سطح لبه است.اگر از آستانه دوم کوچک تر باشد، لبه ای وجود ندارد و اگر بین این دو مقدار باشد یک لبه ی ضعیف وجود دارد.(در Canny لبه ها پیوسته می باشند.)

- Roberts:

دو ماسک عمودی و افقی روی تصویر اعمال می کند.این الگوریتم به نویز حساسیت زیادی دارد و پیکسل های کمتری را برای تقریب گرادیان به کار می برد.همچنین نسبت به Canny قدرت کمتری دارد.

−1	0	0	−1
0	1	1	0

Roberts

خروجی برنامه :