

## تمرین گروهی اصول رباتیکز سری دوم

محمدامین شفیعی ۹۷۲۴۳۰۴۱ - امیرحسام بلندی ۹۷۴۳۰۱۰

نکته: در پروژه ویبائتری که در فایل آپلوده شده قرار دارد تمام سنسور ها و اجزای مورد نیاز تمرین بر روی ربات نصب شده اند.

### سوال اول)

در این سوال پس از تنظیم مقادیر اولیه گفته شده در سوال؛ اسکن کامل را انجام دادیم و قسمت های مختلف سوال را حل کردیم:

**لینک گوگل کولب بخش های سوال:**

<https://colab.research.google.com/drive/1Ijdma00hCNgltyr4mu1F5DLggTs5blv6?usp=sharing>

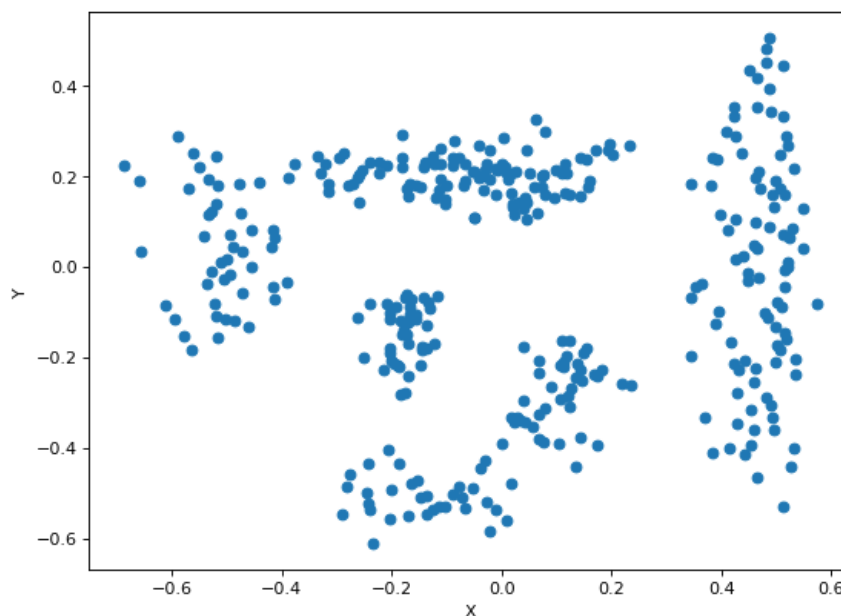
کد کنترلر و پروژه ی ویبائتر این سوال هم در فایل آپلود شده موجود است.

(الف)

در این قسمت و سایر قسمت های سوال؛ با توجه به این که مقدار خروجی سنسور به صورت  $X$  و  $Y$  نیست؛ باید آن را به  $X$  و  $Y$  تبدیل کنیم. روند تبدیل به صورت زیر است:

$X = \text{points}[i] * \text{math.cos}(\text{math.radians}(\text{deg}))$

$Y = \text{points}[i] * \text{math.sin}(\text{math.radians}(\text{deg}))$



شکل-۱) scatter نقاط دریافتی (۳۶۰ درجه)

(ب)

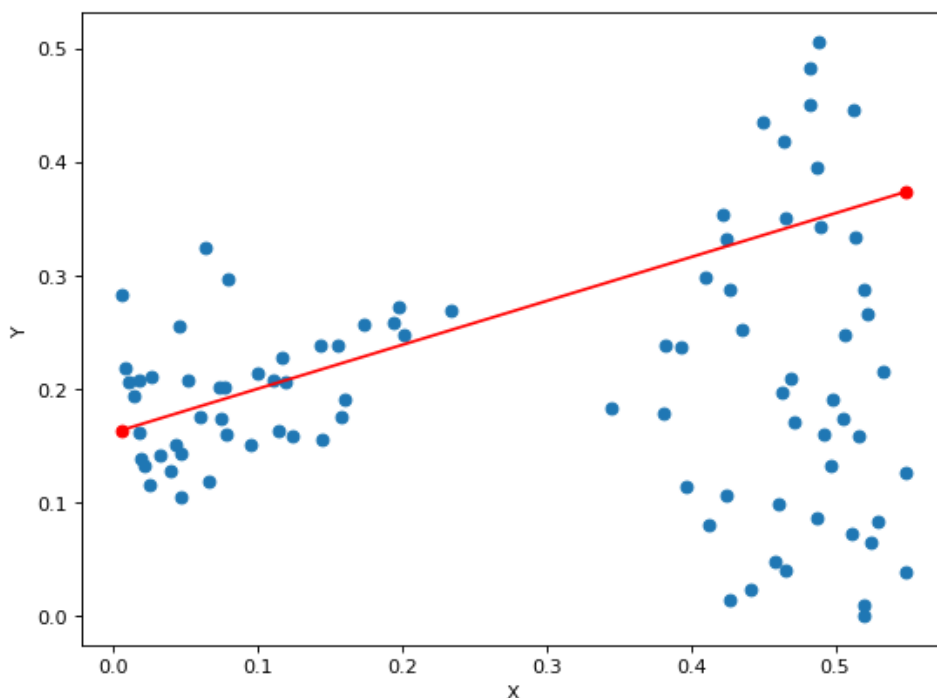
همانطور که در قسمت قبل مقادیر سنسور را به مقادیر دستگاه  $x$  و  $y$  تبدیل کردیم؛ در این قسمت نیز همانگونه عمل میکنیم.

در مرحله ی بعد با استفاده از معادلات زیر مقادیر  $a$  و  $b$  خط مورد نظر را به دست می آوریم:

$$\begin{cases} ax_1 + by_1 = 1 \\ ax_2 + by_2 = 1 \\ \vdots \\ ax_i + by_i = 1 \\ \vdots \\ ax_N + by_N = 1 \end{cases} \equiv \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_i & y_i \\ \vdots & \vdots \\ x_N & y_N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_N$$

$$\begin{aligned} Xp &= c \Rightarrow X^T X p = X^T c \\ \Rightarrow (X^T X)^{-1} X^T X p &= (X^T X)^{-1} X^T c \\ \Rightarrow p &= (X^T X)^{-1} X^T c \end{aligned}$$

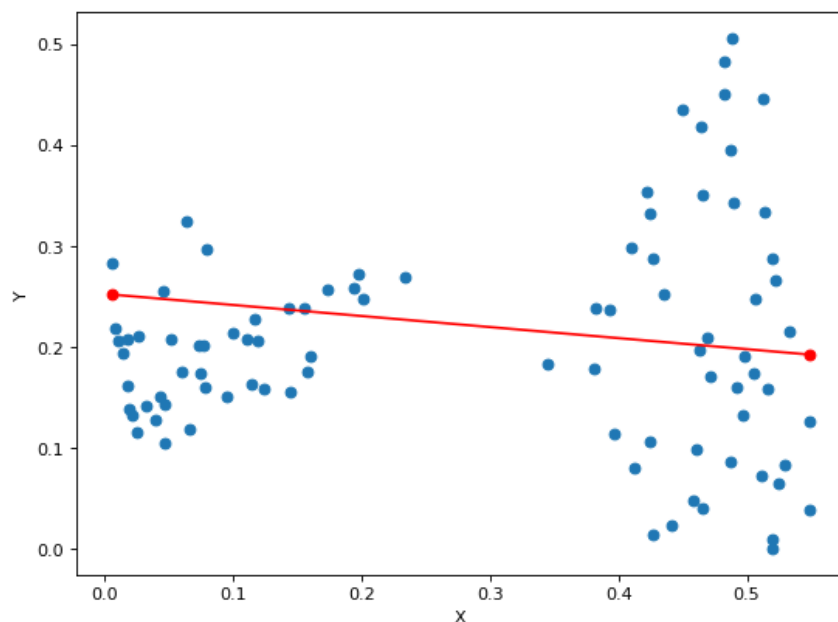
در نهایت پس از محاسبه ی  $a$  و  $b$  خط را رسم کرده ایم:



شکل-۲) خط محاسبه شده با استفاده از الگوریتم شبه معکوس

(ج)

در این قسمت هم الگوریتم RANSAC را مطابق شبه کد زیر پیاده سازی کردیم:

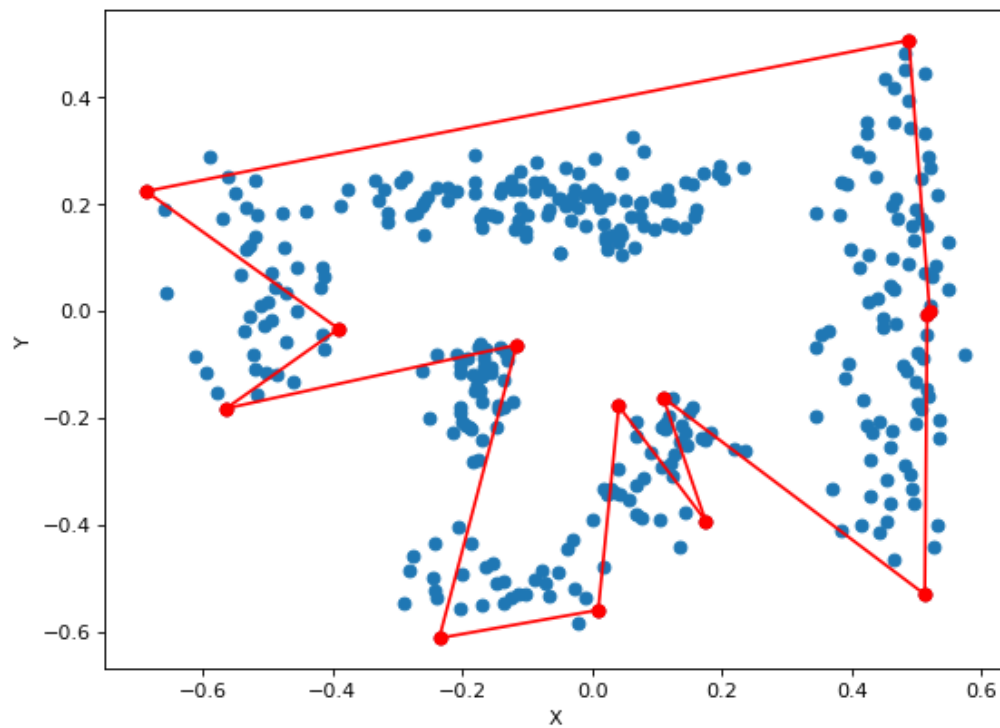


شکل-۳) خط محاسبه شده با استفاده از الگوریتم RANSAC

(د)

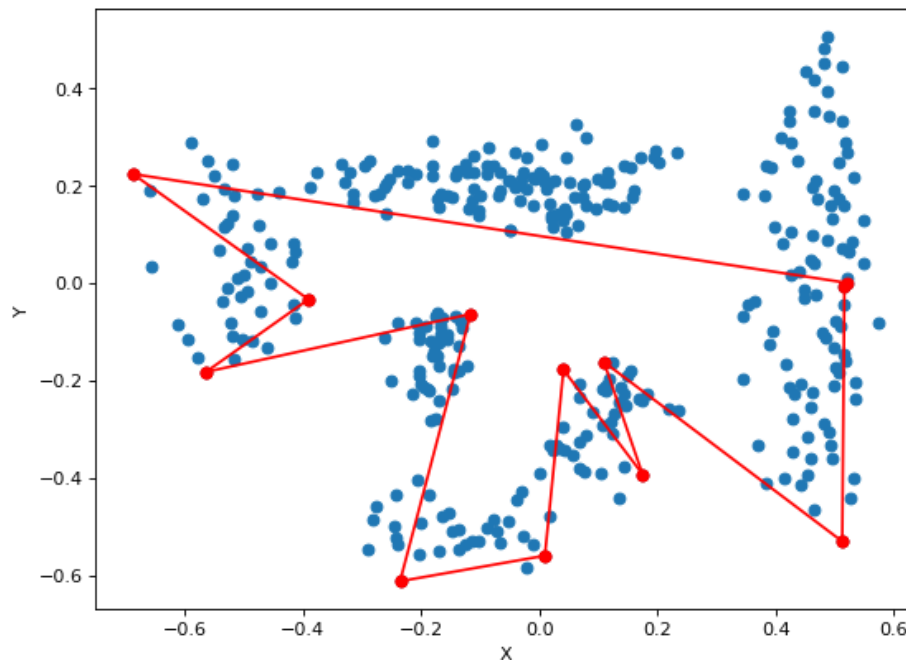
در این قسمت تمام داده (نه صرفاً ۹۰ درجه ی اول) را به دستگاه X و Y میبریم. ابتدا قسمت Split را اجرا میکنیم و سپس قسمت Merge.

به طور کلی روند قسمت split به اینگونه است که بین ابتدا و انتهای بازه داده شده از نقاط یک خط در نظر میگیرد و سپس نقطه ای که بیشترین فاصله از آن را دارد در صورتی که از یک حد آستانه ای که به عنوان ورودی داده می شود بزرگتر باشد؛ یک نقطه ی میانی در نظر میگیرد و خط را به دو قسمت تقسیم می کند و دو سگمنت جدید ایجاد میکند و این روند را به صورت بازگشتی انجام میدهد.



شکل-۴) نتیجه ی اجرای بخش split از الگوریتم split & merge

پس از این که قسمت split اجرا شد باید قسمت merge را اجرا کنیم تا سگمنت هایی که می توانستند خود یک سگمنت بزرگتر باشند را شناسایی کنیم. برای اینکار با مقایسه ی خط ها می توانیم مشخص کنیم که دو سگمنت متوالی فعلی امکان یکی شدن را دارند یا خیر.



شکل-۵) اجرای قسمت merge

همانطور که مشاهده می شود دو تا از سگمنت ها با یکدیگر merge شده اند و یک سگمنت جدید ساخته اند.

نکته ای که در مورد این قسمت وجود دارد این است که به علت نویزی بودن دیتا بهترین شکل اجرای الگوریتم را نمی بینیم.

### سوال یک - قسمت امتیازی)

(الف)

برای این سوال ابتدا هشت سنسور سونار رو به صورت متقارن بر روی ربات نصب میکنیم. و مقادیر گفته شده را تنظیم میکنیم.

(ب)

با توجه به اینکه قرار است ۲۵۶ نمونه برداریم و همچنین با توجه به اینکه نیاز داریم ۳۶۰ درجه را اسکن کنیم؛ بنابراین هر نمونه را برای پوشش  $1.40625 = 360/256$  در نظر میگیریم.

هر سونار هم ۳۲ نمونه تولید خواهد کرد که با توجه تقارن سونار ها ۴۵ درجه ی متوالی خواهد بود. به این ترتیب سونار اول صفر تا ۴۵ درجه را اسکن میکند سونار دوم ۴۵ تا ۹۰ درجه و....(که ما به صورت هشت فایل مقادیر را ذخیره کرده ایم).

همچنین این فایل ها در کنار تمرین قرار داده شده اند.

با کنار هم قرار دادن این هشت و فایل خروجی اسکن سونار ها؛ ۳۶۰ را به صورت کامل اسکن کرده ایم و به حل قسمت های سوال می پردازیم.

(ج)

یکی از چالش های موجود وجود برخی نقاط نامطلوب در داده های ورودی بود که اعمال یک حد آستانه(threshold)؛ مقادیر نامطلوب را حذف کرده ایم. همچنین اسکیل بزرگ ورودی هم می تواند؛ محاسبات پیش رو را تحت تاثیر قرار دهد. به همین جهت ورودی بر ۲۰۰۰ تقسیم کرده ایم.

### قسمت های مشابه سوال اصلی)

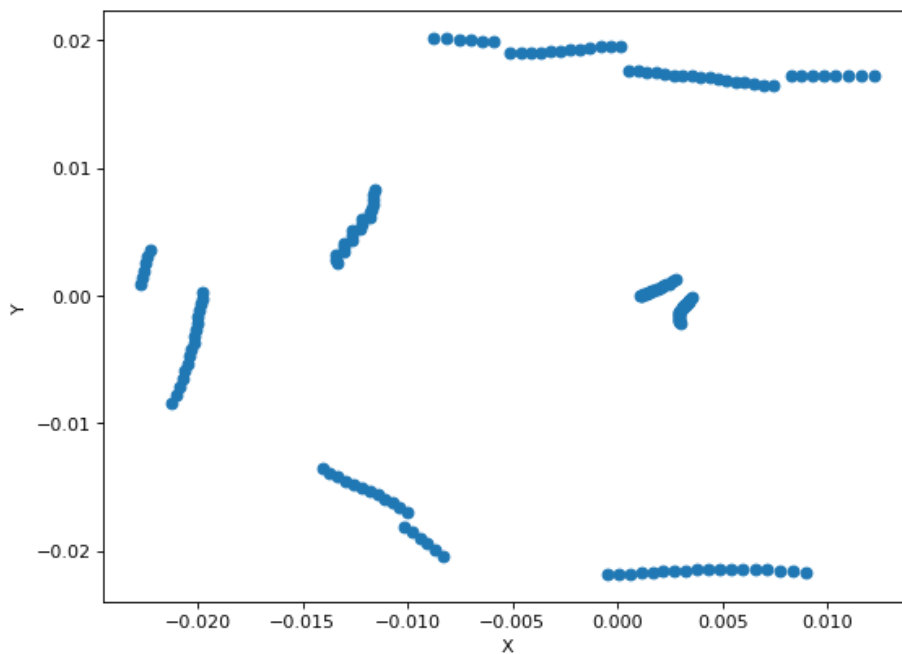
**لینک گوگل کولب این بخش ها:**

<https://colab.research.google.com/drive/14b-svMfd2P2zCgvJF4hGPR-W05u8Hzaw?usp=sharing>

کد کنترلر و پروژه ی ویباتر این سوال هم در فایل آپلود شده موجود است.

(الف)

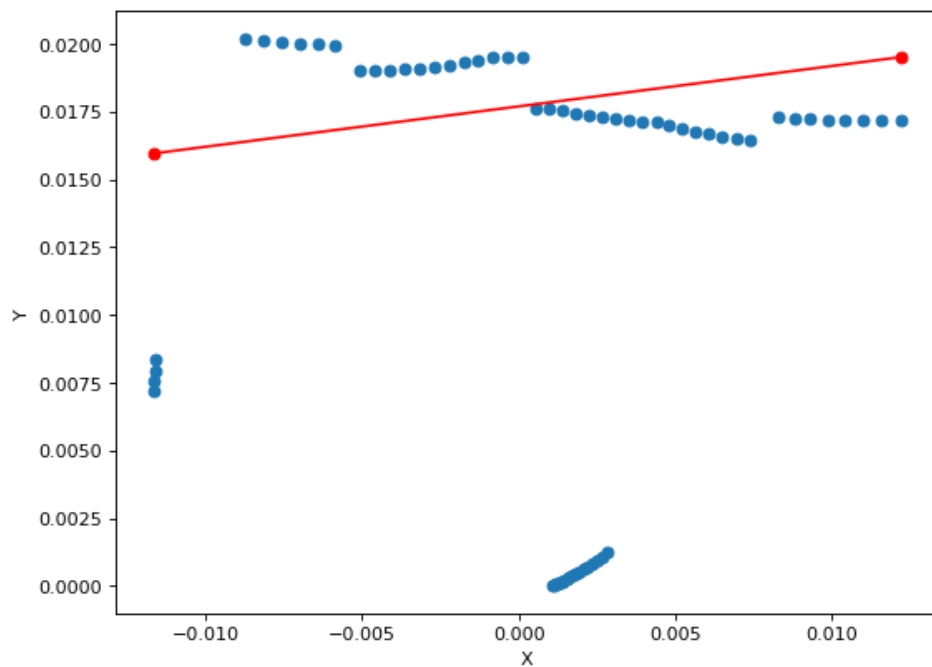
همانند سوال قبل ابتدا باید به دستگاه  $x$  و  $y$  منتقل کنیم. سپس scatter را رسم میکنم:  
(اسکتر رسم شده پس از اعمال اصلاحات گفته شده در قسمت قبل است.)



شکل-۶) scatter نقاط به دست آمده با رینگ متقارن و ۸ تایی سونار

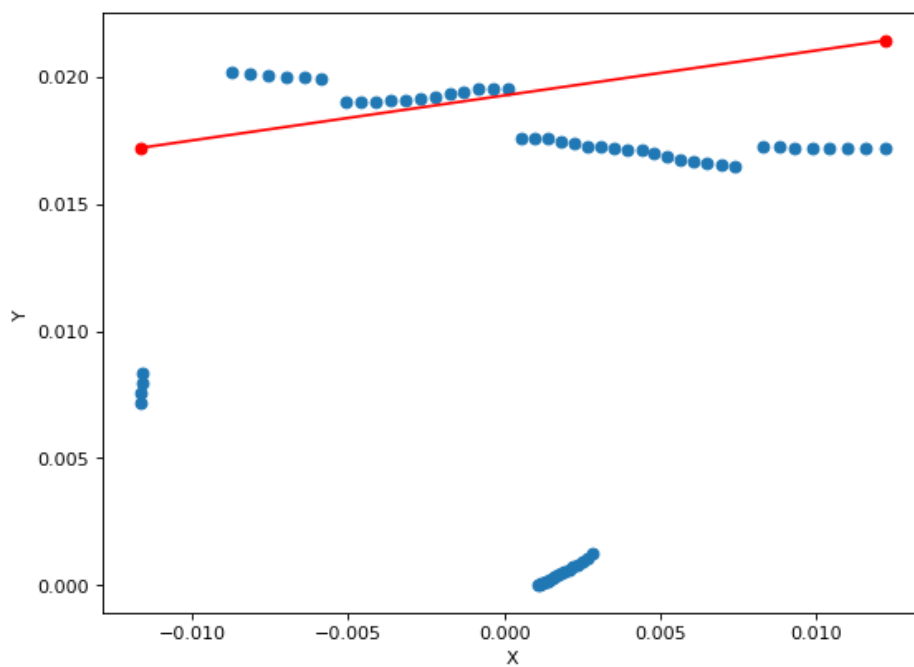
(ب)

در این قسمت هم مطابق سوال اول بر روی نود درجه ی ابتدایی الگوریتم شبه معکوس را اجرا میکنیم:



شکل-۷) نتیجه ی اجرای الگوریتم شبه معکوس

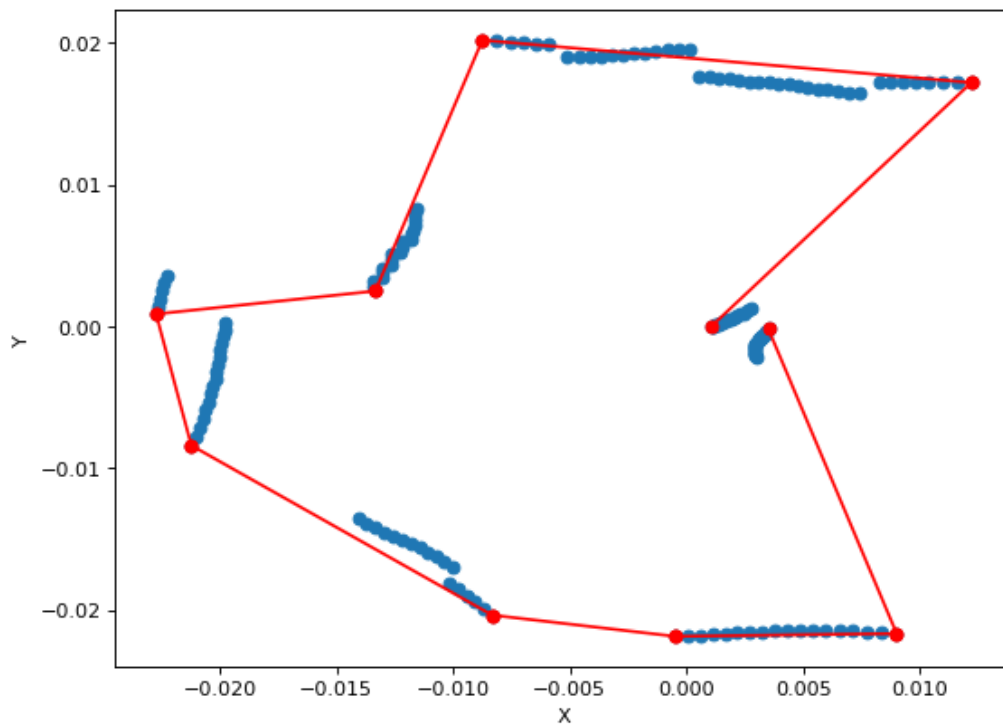
(ج)



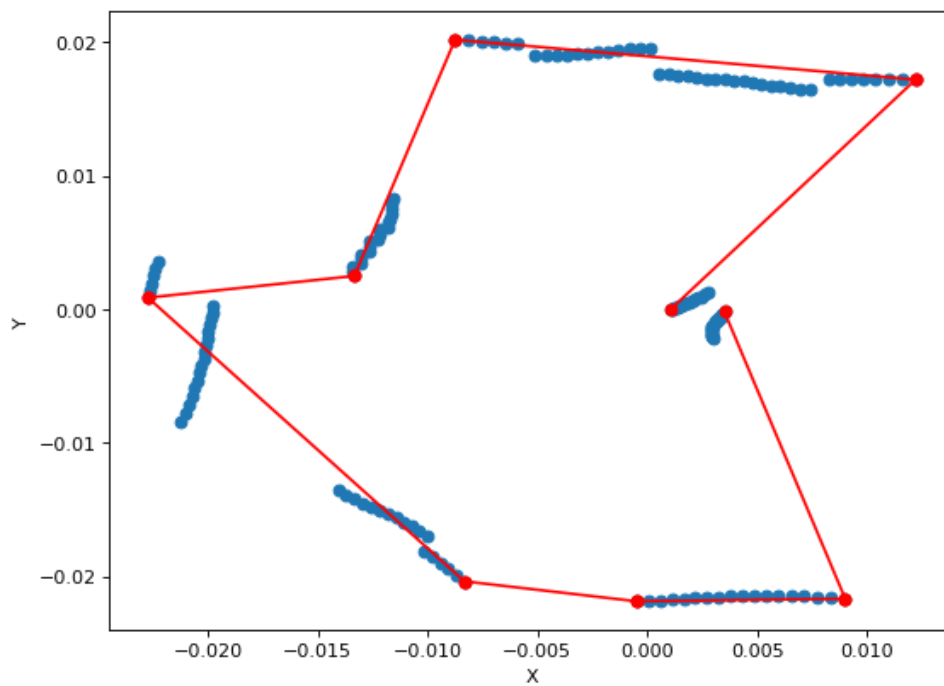
شکل-۸) نتیجه ی اجرای الگوریتم RANSAC

(د)

در این قسمت هم الگوریتم split & merge را مطابق سوال اصلی و بر روی تمام ۳۶۰ درجه اجرا میکنیم.



شکل-۹) پس از اجرای split



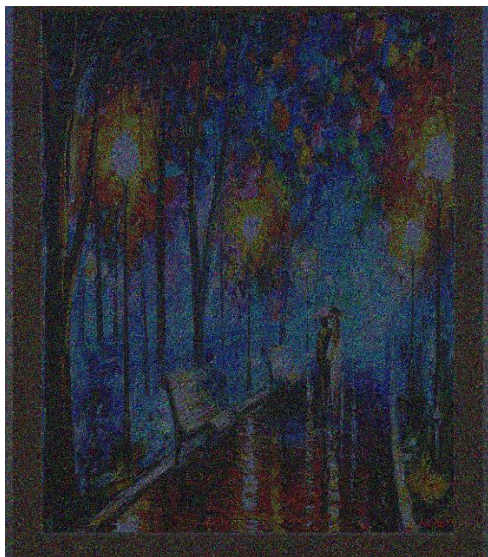
شکل-۱۰) پس از اجرای merge

همانطور که مشاهده می شود در اینجا هم دو مورد از سگمنت ها به یکدیگر مرج شده اند و یک سگمنت واحد ساخته اند.

سوال سوم)



برای این سوال در محیط ویباتز پس از اینکه camera را نصب کردیم؛ ابتدا نویز ۰.۲ را در آن تنظیم میکنیم. سپس width و height را بزرگتر از مقدار اولیه که ۶۴ پیکسل است قرار می دهیم تا عکس بزرگتر و مناسب تری را دریافت کنیم.



شکل-۱۱) نمونه تصویر دریافتی از دوربین

پس از دریافت تصویر از دوربین باید با استفاده از فیلتر گاوسی سعی کنیم تصویر را هموار کنیم و اثر نویز را تا حد ممکن از بین ببریم.

برای این سوال، دو راه مختلف را امتحان کردیم:

۱. ابتدا فیلتر گاوسی را اعمال کردیم و سپس تصویر را به تصویری grayscale تبدیل کردیم؛ نهایتاً کرنل های prewit

۲. ابتدا تصویر را grayscale کردیم و سپس فیلتر گاوسی را اعمال کردیم. پس از آن کرنل های prewit افقی و عمودی را اعمال کردیم تا لبه ها نمایان شوند.

نکته ی مهم دیگر هم استفاده از تابع abs بود. مقادیر ممکن است پس از اعمال فیلتر های مذکور منفی شوند؛ یا از ۲۵۵ بزرگتر شوند. برای این که مقادیر در بازه ی ۰ تا ۲۵۵ باشند راه های زیر را امتحان کردیم:

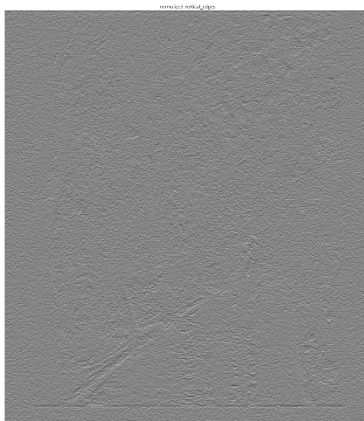
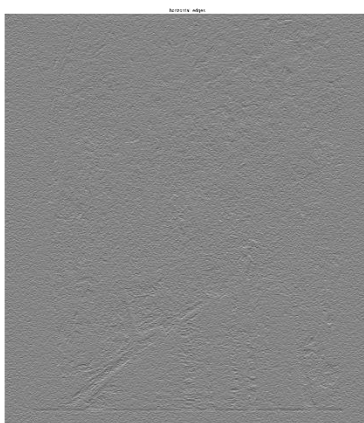
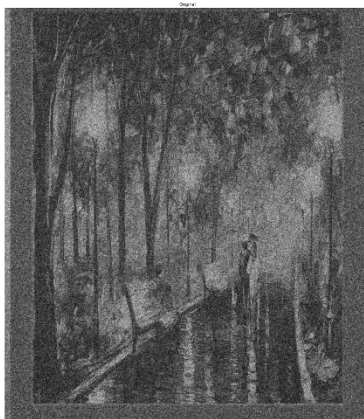
۱. مقادیر را به صفر تا ۲۵۵ اسکیل کردیم

۲. تابع abs را اعمال می کنیم.

۳. مقادیر منفی را ۰ و بیش از ۲۵۵ را خود عدد ۲۵۵ در نظر گرفتیم.

**لینک گوگل کولب سوال سوم:**

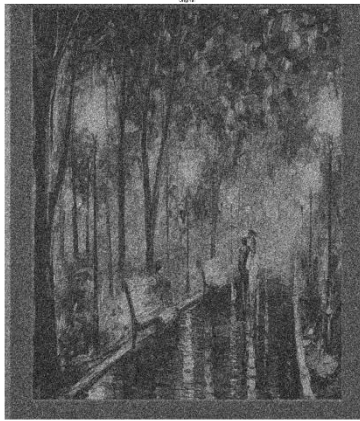
<https://colab.research.google.com/drive/11OPvl3rCfOWes8p2bGlXguQRmnbGkQ1e?usp=sharing>



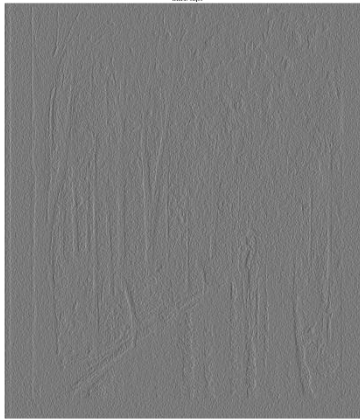
در شکل بالا تصاویر یک تا چهار به ترتیب عبارتند از:

۱. عکس اصلی پس از اعمال فیلتر گاوسی و grayscale
۲. عکس حاصل از اعمال فیلتر horizontal prewitt روی عکس اول
۳. عکس حاصل از تبدیل مقادیر منفی عکس دوم به صفر و مقادیر بیش از ۲۵۵ به ۲۵۵
۴. عکس حاصل از اسکیل کردن کلیه ی مقادیر عکس دوم به بازه ی ۰ تا ۲۵۵

Copyright



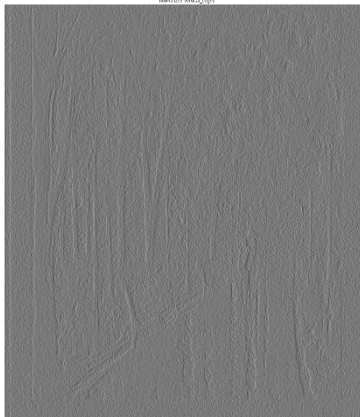
central edges



Involved central edges



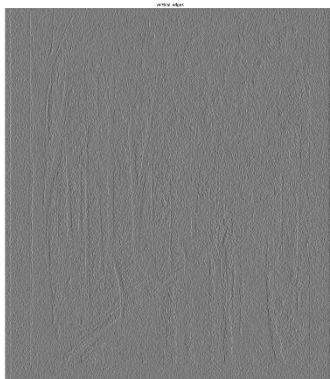
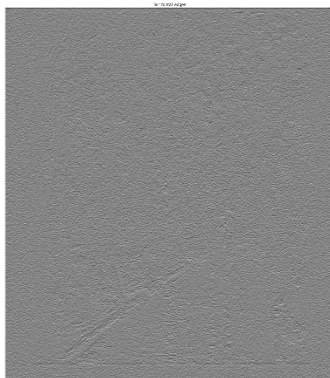
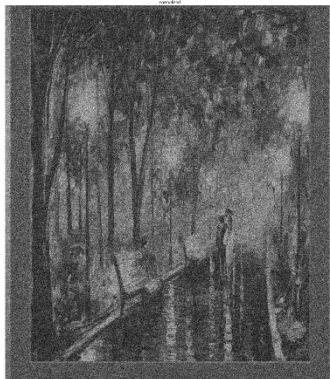
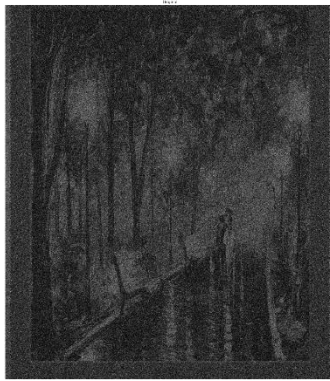
normalized vertical edges



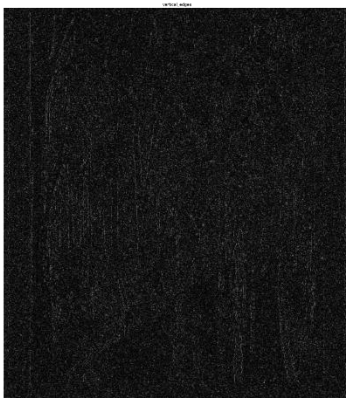
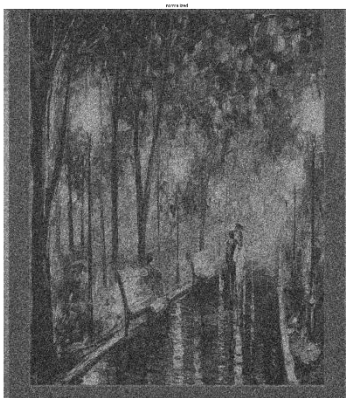
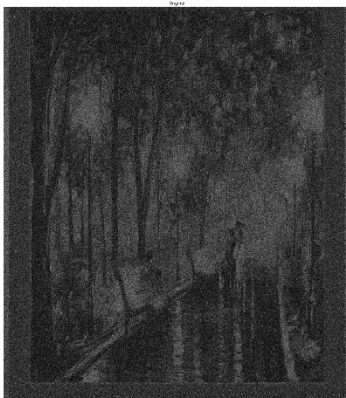
در شکل بالا تصاویر یک تا چهار به ترتیب عبارتند از:

۱. عکس اصلی پس از اعمال فیلتر گاوسی و grayscale
۲. عکس حاصل از اعمال فیلتر vertical prewitt روی عکس اول
۳. عکس حاصل از تبدیل مقادیر منفی عکس دوم به صفر و مقادیر بیش از ۲۵۵ به ۲۵۵
۴. عکس حاصل از اسکیل کردن کلیه ی مقادیر عکس دوم به بازه ی ۰ تا ۲۵۵

در زیر هم به کاری گیری صرفا تابع abs و ترتیب های مختلف فیلتر گاوسی و grayscale را مشاهده میکنم.

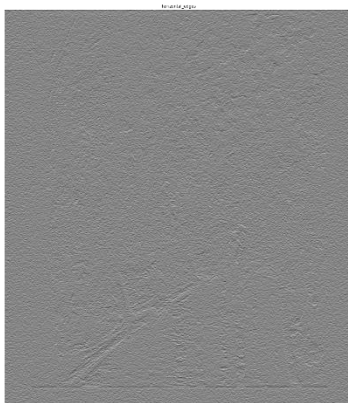
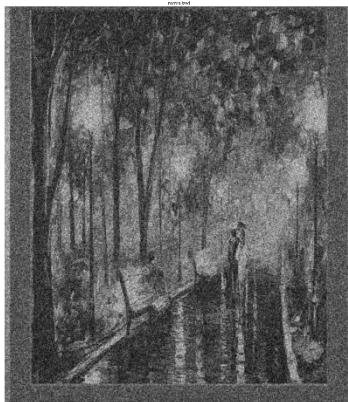
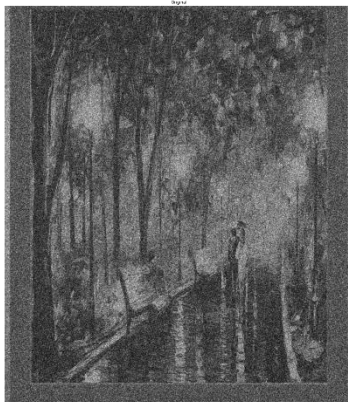


شکل-۱۱) ابتدا به به grayscale و سپس استفاده از فیلتر گاوسی و بدون استفاده از تابع abs

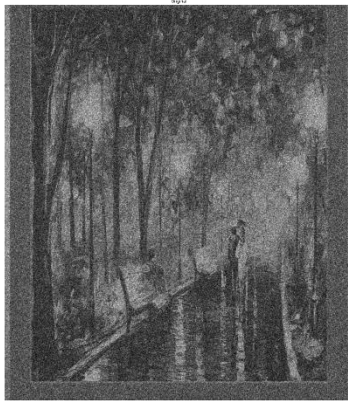


شکل-۱۲) همان روند قبل و با استفاده از تابع abs





شکل-۱۳) ابتدا اعمال فیلتر گاوسی و سپس تبدیل به grayscale و بدون استفاده از abs



شکل-۱۴) همان روند قبل و با استفاده از تابع abs

همانطور که مشاهده می شود عملا تفاوت خاصی بین ترتیب اعمال فیلتر گاوسی و grayscale وجود ندارد.

همچنین کد کنترلر و پروژه ی ویبائز این سوال هم در فایل آپلود شده موجود است.