



به نام خدا



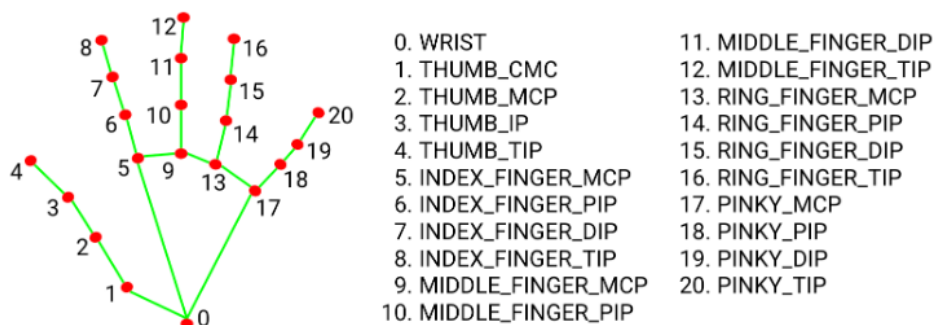
دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
مکاترونیک

گزارش مینی پروژه ۵

نام و نام خانوادگی	کامیار رحمانی
شماره دانشجویی	۸۱۰۱۹۹۴۲۲
تاریخ ارسال گزارش	۱۴۰۲/۳/۲۶

• بخش دنبال کردن دست توسط لاکپشت

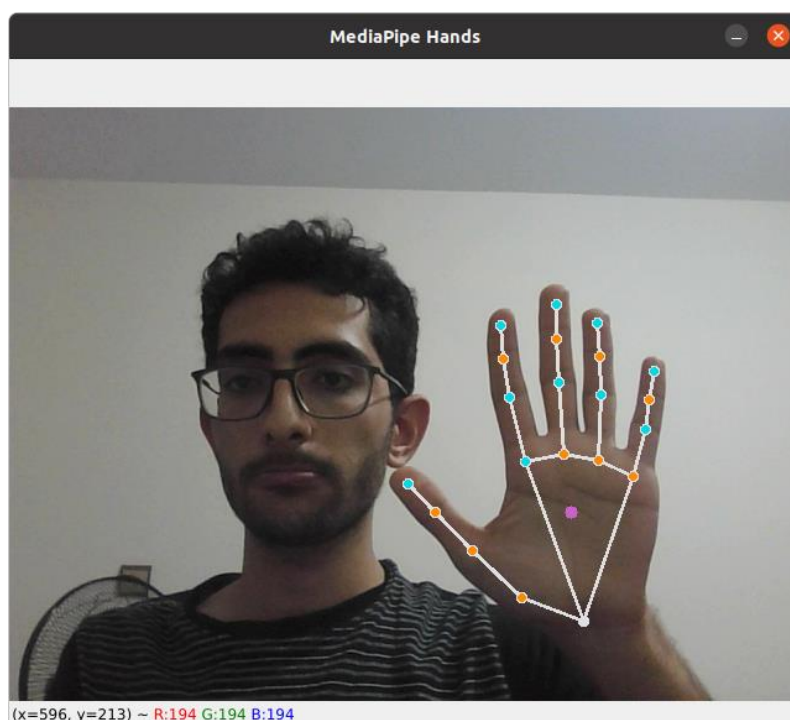
در ابتدا یکی از چالش هایی که داشتم این بود که مرکز دست به صورت آماده با استفاده از کتابخانه قابل نمایش نبود و تنها مختصات نقاط قابل بدست آوردن با کتابخانه بود:



با توجه به شکل بالا میتوان حدس زد که با میانگین گرفتن از مختصات نقاط 0 و 5 و 9 و 13 و 17 و 1 می توان مختصات مرکز را بدست آورد ولی وقتی که کد مربوط به آن را نوشتم دقیقا در مرکز دست قرار نگرفت و با اضافه کردن نقطه 1 به نقاط قبلی و میانگین گرفتن از کل آنها توانستم مرکز دست را بدست آورم:

```
center_x = (landmarks[0].x + landmarks[5].x + landmarks[9].x + landmarks[13].x + landmarks[17].x + landmarks[1].x) / 6
center_y = (landmarks[0].y + landmarks[5].y + landmarks[9].y + landmarks[13].y + landmarks[17].y + landmarks[1].y) / 6
```

نتیجه:



وقتی میخواستیم با استفاده از دستور `cv2.circle` مرکز دست را به صورت نقطه نمایش دهم خطای `type` میگرفتم و این به این دلیل بود که باید قبل از دادن مختصات مرکز به `cv2.circle` آنها را به `integer` تبدیل میکردم.

اما وقتی تابع `circle` را اجرا کردم نقطه مرکز دست را ندیدم. این به این دلیل بود که سایز پنجره `cv2` برای نمایش وبکم 480 در 640 بود و بنابراین مختصات بدست آمده را در این اعداد ضرب کردم تا `scale` انجام شود:

```
cv2.circle(image, (int(center_x*image.shape[1]), int(center_y*image.shape[0])), 5, (200, 100, 200), -1)
```

برای اینکه مختصات دست را به لاکپشت ارسال کنم در ابتدا به نظرم آمد که از یک `message` ساختگی خودم استفاده کنم که در آن دو متغیر از جنس `Float64` تعریف کردم :

```
1 float32 centerx
2
3 float32 centery
```

یک روش دیگر این بود که این دو پیام را تبدیل به یکی کنم. اولین ایده ساختن یک لیست شامل این دو بود. برای اینکه لیست را منتقل کنم از یکی `message` های استاندارد `ros` به نام `Float64MultiArray` استفاده و به صورت زیر پیام را `publish` کردم:

```
center= Float64MultiArray()
center.data = [center_x , center_y]
pub.publish(center)
rate = rospy.Rate(1000)
```

برای دریافت این پیام در نود `subscriber` ابتدا `data` را دریافت کردم و عنصر اول و دوم آن را جداسازی نمودم.

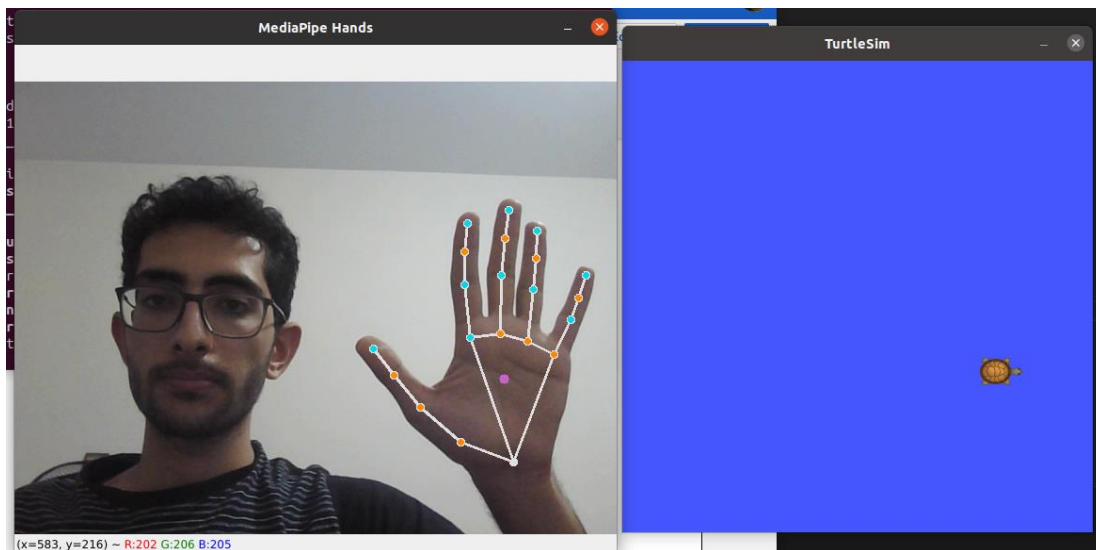
ولی بعد از نود مربوط به لاکپشت را ایجاد کردم لاکپشت دایما با دیوار برخورد میکرد و باید مجددا مختصات را با توجه به اندازه پنجره turtlesim اسکیل میکردم. طول و عرض پنجره turtlesim تقریباً 11*11 است و scale کردن را به صورت زیر انجام دادم:

```
(center[0]*11), 11-(center[1]*11)
```

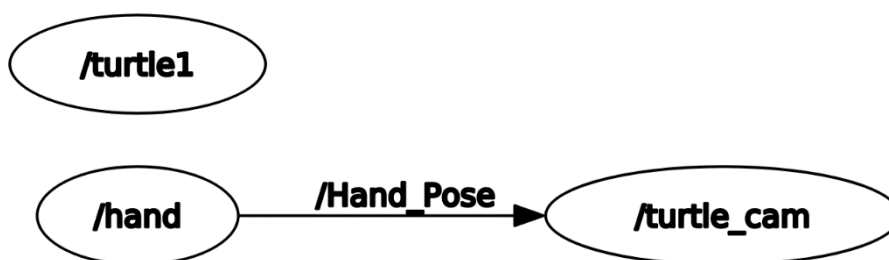
با استفاده از دستور TeleportAbsolute لاکپشت را با توجه به مختصات دست جابجا کردم و برای اینکه رد لاکپشت به جا نماند از دستور SetPen استفاده کردم:

```
rospy.wait_for_service('turtle1/set_pen')
pen_setter = rospy.ServiceProxy('turtle1/set_pen', SetPen)
pen_setter(100, 0, 0, 0, 1)
teleporter = rospy.ServiceProxy('turtle1/teleport_absolute', TeleportAbsolute)
teleporter((center[0]*11), 11-(center[1]*11), 0.0)
```

در نهایت نتیجه به صورت زیر شد:



نمایش نودها در: rqt_graph:



در شکل بالا topic به اسم Hand_Pose از نود hand به نود turtle_cam منتقل می شود. نود turtle1 نیز مربوط به لاکپشت است.

• بخش سرویس

در این بخش برای تعریف کردن سرویس از نود جدید استفاده نکردم. از همان نود hand که مختصات دست را در آن بدست می آورديم استفاده کردم.

```
service1 = rospy.Service('save_image_service', Service1, handler=image_save)
service2 = rospy.Service('gray_image_service', Service2, handler=image_save_gray)
```

در شکل بالا دو سرویس save_image_service و gray_image_service را تعریف کردم و در توابع image_save و image_save_gray با استفاده از دستور cv2.imwrite عکس ها را ذخیره کردم.

لیست سرویس ها:

```
kamiar@kamiar-IdeaPad-Gaming-3-15IMH05:~$ rosservice list
/clear
/gray_image_service
/hand/get_loggers
/hand/set_logger_level
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/save_image_service
/spawn
/turtle1/get_loggers
/turtle1/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtle_cam/get_loggers
/turtle_cam/set_logger_level
```

همانطور که مشخص است دو سرویس تعریف شده به لیست سرویس ها اضافه شدند.

لیست تاپیک ها:

```
kamiar@kamiar-IdeaPad-Gaming-3-15IMH05:~$ rostopic list
/Hand_Pose
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
```

لیست **message** ها: چون لیست مسیج ها زیاد بود فقط بخشی که استفاده کردم را قرار دادم یعنی

Float64MultiArray

```
std_msgs/Bool
std_msgs/Byte
std_msgs/ByteMultiArray
std_msgs/Char
std_msgs/ColorRGBA
std_msgs/Duration
std_msgs/Empty
std_msgs/Float32
std_msgs/Float32MultiArray
std_msgs/Float64
std_msgs/Float64MultiArray
std_msgs/Header
std_msgs/Int16
std_msgs/Int16MultiArray
std_msgs/Int32
std_msgs/Int32MultiArray
std_msgs/Int64
std_msgs/Int64MultiArray
std_msgs/Int8
std_msgs/Int8MultiArray
std_msgs/MultiArrayDimension
std_msgs/MultiArrayLayout
std_msgs/String
std_msgs/Time
std_msgs/UInt16
std_msgs/UInt16MultiArray
std_msgs/UInt32
std_msgs/UInt32MultiArray
std_msgs/UInt64
std_msgs/UInt64MultiArray
std_msgs/UInt8
std_msgs/UInt8MultiArray
```

لیست نودها:

```
kamilar@kamilar-IdeaPad-Gaming-3-15IMH05:~$ rostopic list
/hand
/rosout
/turtle1
/turtle_cam
```