CS x476 Project 6

Zhaodong Yang
zyang645@gatech.edu
903748903

detect_3d_box(): Code (10 points)

Please screenshot your "**detect_3d_box()**" here (only the two steps you implemented, not the whole function)

```
################################################################################
# TODO: YOUR CODE HERE
################################################################################
# Step-1: Call inference and get the result.
hm, displacements = inference(image, model_path)

# Step-2: Decode bounding boxes from inference result .
boxes = decode(hm, displacements)




################################################################################
#                              END OF YOUR CODE
################################################################################
```

Part 1: (8 points)

Briefly describe your understanding about the pipeline of mediapipe’s Objectron detection. Describe the stages and their required input/outputs

The first stage takes in an image and use MobileNetv2 to estimate a heatmap and displacement fields. The heatmap is actually a Gaussian distribution which take center of the detected object as mean. The second stage takes in the heatmap and the displacement and calculate the position of the box vertices.

Part 1: (4 points)

Is it possible to recover a single 3D point from a 2D point of a monocular image (which means a single image taken by a single camera)?

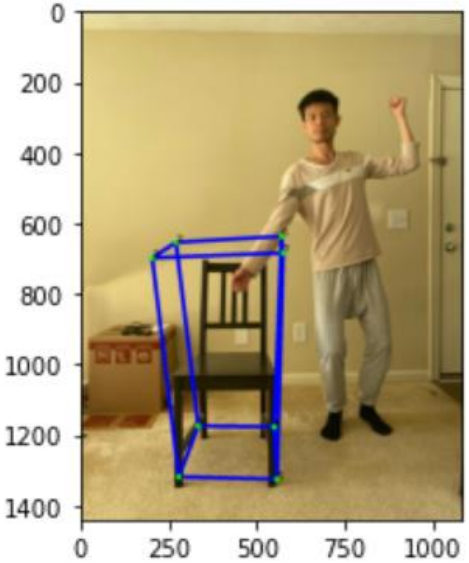
No. We need the depth information. And we can not estimate depth information from a single monocular image.

Part 1: (4 points)

Why is it possible to estimate a 3D object from a monocular image (like mediapipe's Objectron)? What other assumptions or data is needed to accomplish this.

Because mediapipe's objectron uses MobilePose-Shape which is supervised. We need depth assumptions to accomplish this.

Part 1: (4 points)




```
################################################################################
# TODO: YOUR CODE HERE
################################################################################
# Step-1: Call inference and get the result.
hm, displacements = inference(image, model_path)

# Step-2: Decode bounding boxes from inference result .
boxes = decode(hm, displacements)




################################################################################
#                             END OF YOUR CODE
################################################################################
```

Part 2: (5 points)

After you did camera calibration, you get a more accurate K, the intrinsic matrix of the camera, can you describe what is the meaning of the five non-zero parameter in K?

The fx and fy is the focal lengths for the sensor x and y dimensions. The entry s is skew between the sensor axes. The cx and cy is the image center expressed in pixel coordinates.

Part 2: (5 points)

In the K (intrinsic matrix), there is one value representing fx and another one representing fy, what is the unit of those two values? Why? In practice, when fx is not equal to fy, what does this imply?

The unit of those two values is pixel. Because fx = f * sx. F is the focal length of the physical lens which has a unit of millimeter, and sx is number of pixels per unit length along x axis, which has a unit of pixel/mm. So as fy. As a result, the unit of fx and fy is pixel. If fx is not equal to fy, it means the pixel or the individual imager elements is rectangle instead of square.

Part 2: (6 points)

You also performed the transformation from world to camera by using the equations below.

1) Previously what we did was from world coordinate to camera coordinate. If we perform the inverse, which is from camera coordinate to world coordinate, will also have a similar equation1, but the w and c will change. Then there will be a ctw in the change equation, what does ctw represent?

2)Using the equation2 and equation3 below, can we describe why the P matrix can project 3D points in world coordinate to 2D points on image plane? (Hint: the P matrix achieves two coordinate transform)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} {}_{w}\mathbf{R}_{c}^{\mathrm{T}} & -{}_{w}\mathbf{R}_{c}^{\mathrm{T}}\ {}_{w}\mathbf{t}_{c} \\ 0^{\top} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{eq.1}$$

$$\mathbf{P} = \mathbf{K}\ {}^{w}\mathbf{R}_{c}^{\top}[\mathbf{I} \mid -{}^{w}\mathbf{t}_{c}] \quad \text{eq.2}$$

$$z\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad \text{eq.3}$$

1) The ctw is the translation of the world frame in the camera coordinate.

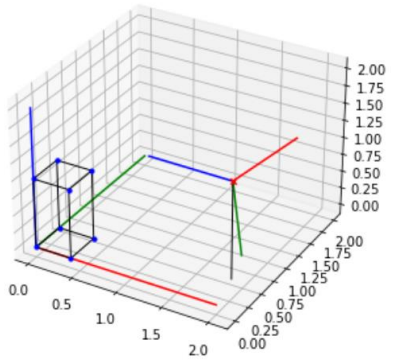
2) Because the [I| - wtc] decreases the dimension of 3D point to 2D point, while wtc and wRtc transform points from world frame to camera frame at the same time. Finally K transform point from camera frame to image frame.

Part 2: (4 points)

According to the box and camera plot at the end of Part 2. Where is the camera? Where is it facing?

<text answer here>

The camera if in front of the chair. It is facing towards the chair.



hand_pose_img(): Code (10 points)

Please screenshot your "**hand_pose_img()**" here (only the two steps you implemented, not the whole function)

```
################################################################################
# TODO: YOUR CODE HERE
################################################################################
# Reminder: check the mediapipe documentation first!
# Step-1: Pass the image to the pose model.
results = pose.process(image)

# Step-2: Get landmark from the result.
landmark = results.pose_landmarks

################################################################################
#                              END OF YOUR CODE
################################################################################
```

Part 3: (4 points)

Please describe an application for pose estimation and explain why it is useful.

When making anime, some producers will use actual actors to play anime character and capture their motion to make anime more vivid. They will use pose estimation to capture the motion of actors.In that way, they can enable the model of the anime character to perform the same motion as human actors.

Part 3: (4 points)

If you are going to do a pose detection project, what kind of pose do you want to detect and explain why these pose are important for you.

I want to detect my pose when I work out. Then I can know better if my motion is correct. Or even compare my pose with standard pose and calculate the difference.

Part 3: (6 points)

What are the two main steps associated with pose detection used in mediapipe (Hints, read the blog post of mediapipe's pose detection)?

The first steps is using a detector to locate the person/pose region-of-interest within the frame. The second step is to use a tracker to predicts pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input.

Part 3: (4 points)


```
################################################################################
# TODO: YOUR CODE HERE
################################################################################
# Reminder: check the mediapipe documentation first!
# Step-1: Pass the image to the pose model.
results = pose.process(image)

# Step-2: Get landmark from the result.
landmark = results.pose_landmarks

################################################################################
#                              END OF YOUR CODE
################################################################################
```

Part 4: (6 points)

How would you estimate depth information from a 2D image, given that the person’s feet and the chair are on the same floor and are both visible in the image?

I would at first detect the 3D bounding box of the person’s feet and get the pose of his feet. Then I would assume his feet length a constant value to get the depth value of his feet. As I know the chair and the feet are on the same floor I can also calculate the depth information of the chair.

check_hand_inside_bounding_box(): Code (10 points)

Please screenshot your "**check_hand_inside_bounding_box()**" here (only the steps you implemented, not the whole function)

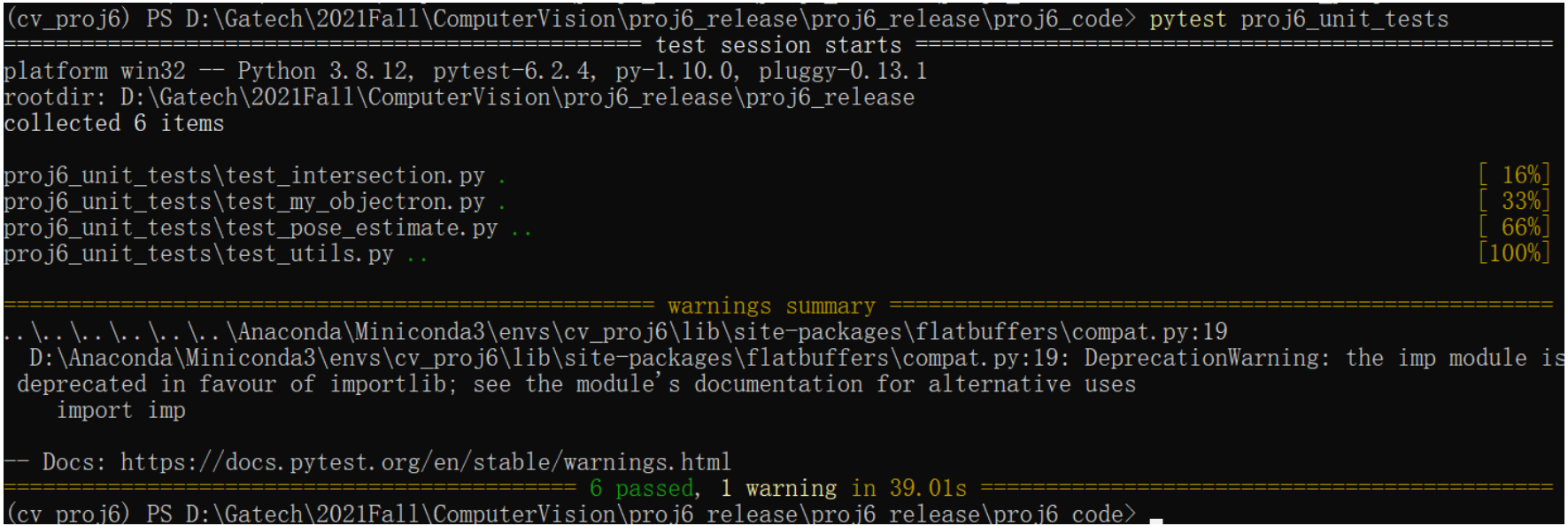
```
################################################################################
# TODO: YOUR CODE HERE
################################################################################
transform = np.amin(pts[:,:3], axis = 0)
boundary = np.amax(pts[:,:3] - [transform],axis = 0)
hand_transformed = hand - transform

inside = np.all(np.less_equal(hand_transformed, boundary)) and np.all(np.less_equal([0,0,0],hand_transformed))


################################################################################
#                              END OF YOUR CODE
################################################################################
```

Part 5: (6 points)

Given the 3D coordinates of eight vertices of a box in space, and one 3D point, describe how do you detect whether this point is inside or outside the box?

I used np.amin() to get the minimum value for each dimension from the box vertices. Then subtract the result of np.amin() from original hand and box coordinate. Then I would check if all elements of subtracted hand coordinate are positive or equal to zero at first. After that I checked if the coordinate of the hand is less than or equal to the maximum value for each dimension from the box vertices. If the subtracted hand coordinate passed both two tests then it is inside the box, otherwise it is not.

Screenshot the result of running “pytest” on this page:



Extra Credit (for all): Your own image

Insert your picture before interacting with the object and other picture after the interaction happens (the bounding box changes color)

Extra Credit (for grad students): Interaction Video

<Screenshot the code you implemented in **process_video()** here>

Extra Credit (for grad students): Interaction Video

<Tell us where to access your final video of part 1 or 2, Discuss what you found out. If you have a two-chair video, you don’t have to record the one-chair video again. >

Extra Credit (for grad students): Interaction Video

< What kind of factors determined how accurate the intersection detection was?>