Computer Vision

Spring 2019 CS543/ECE549 Projects

Hsiu-Yao Chang (hsiuyao2)
Bo-Rong Chen (borongc2)
Hsin-Yu Hou (hsinyuh2)

Introduction

The aim of our project is to reconstruct 3D model from a single view perspective image. We refer [2] for reconstructing images. However, we further extend their works into two different field. Firstly, since our project is going to reconstruct the 3D view of tennis court [fig.1], it is different from their works, which only constructs a box. Besides, instead of using label or hand-engineering to compute the vanishing, we use Mask R-CNN to detect a particular object. We, in turn, develop a finding algorithm to find the key points. By utilizing these points, our work is able to automatically reconstruct the 3D views. Our main constructions are: 1) extend the original work [2] into reconstruct non-box-shaped objects, 2) make the whole process automatically, and 3) use OpenGL to produce the whole 3D view into video sequences.

We divide our work into two steps. First of all, in the original design, we have to label the vertices of object manually to get the vanishing point. After finding vanishing points, we can use homography transform to get the XY, YZ, XZ plans of the object and reconstruct the 3D model.

Moreover, we then push our project further by detecting the vertices and lines automatically. We first use Mask R-CNN to get the margin of object that we want to reconstruct [fig 6. 7]. After that, we implement a point detector to get six vertices of cuboid to find the vanishing points of the object. Last, we use the same technique above to construct the 3D model.

Details of the approach

The intriguing parts of our project are: 1) how to make the whole reconstructing processing automatically? In fact, we design a point-finding algorithm to find the key points around a bus image **[fig.6]**. However, the center point inside the bus image cannot be detected by Mask R-CNN. Hence, we utilize the vanishing points and the points around the edges to compute the center point [fig.8]. 2) Since we do not have enough time to retrain the Mask R-CNN to detect the whole tennis courts, we still present our original work on reconstructing 3D image of tennis courts by labeling and pinpointing the lines and key points. Under such circumstances, we present our work is able to reconstruct the whole 3D view of the tennis courts. And, we utilize off-the-shelf detecting algorithm, Mask R-CNN, and a bus image to develop the automatic algorithm to reconstruct the whole 3D images. In order to make the whole algorithm automatically, we make some assumptions. 3) We also present videos concerning our final results. The links are available here: Tennis(https://youtu.be/vJj32U-ZawM), Bus(https://youtu.be/1YJnzVArV38)

a. Assumption

In the second part of our project, considered the feasibility of this project, we assume the input of image has three vanishing points and the object that we want to reconstruct is a cuboid so that we can obtain six vertices from our point detector. In our point detector implementation, it so far can only handle this specific image due to the output from Mask R-CNN.

b. Finding vanishing points of object

To obtain the vertices in order to find the vanishing point, we use two strategies to implement it. The first one is manually labeling the vertices for tennis court in the basic functionality, the other method is using Mask R-CNN to get the margin object then detecting vertices by implementing vertices detector.

The second method, we get a mask image with value 0 and 1 from Mask R-CNN. We then apply line filter to the mask image to get margin images of object. However, since the margin is not straight, we have to find the vertices to construct the straight edges of the object. After knowing the vertices of object, we can compute the vanishing point by finding the intersection of two lines which are parallel in the real-world.

c. Homography transform and Projection

With three vanishing points, we construct the homography matrix that gives us XY, YZ and XZ plans of objects by switching the column of homography matrix.

d. Mask R-CNN

We use the pretrained Mask R-CNN from [1] to detect the mask of target object. In this process, we expect the output shape of the mask is as close as the shape of object, so we choose our input carefully because the pretrained model that can not output perfect masks of objects. The mask image return from the network is an array with value 0 and 1.

e. 3D Construction

With the input of the XY, YZ, XZ plan images, we reconstruct the model by using OpenGL. We extract and crop the three plans of object manually and transform it into a rectangle textures. Next, we use the referenced length to resize the texture and attach them to get the 3D model.

Some texture might overlap with the texture of other model. For example, the texture of net will also appear in the homographic result of tennis court. We made some adjustment and remove the overlapped texture for clearer representation of our result.

Results

Part 1. Reconstruct tennis court

Fig 1. Tennis court and the virtual axis for reconstruction

Fig 2. Cropped images after homography transformation

(left:Tennis Court view from z axis)

(right: y axis)

Fig 3. Cropped images of human

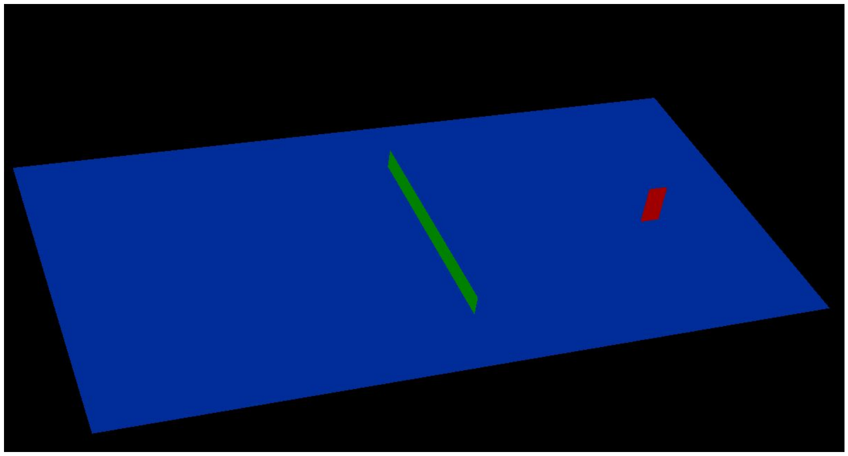

Fig 4. 3D Construction without texture

(Blue: tennis court; Green: net; Red: human)

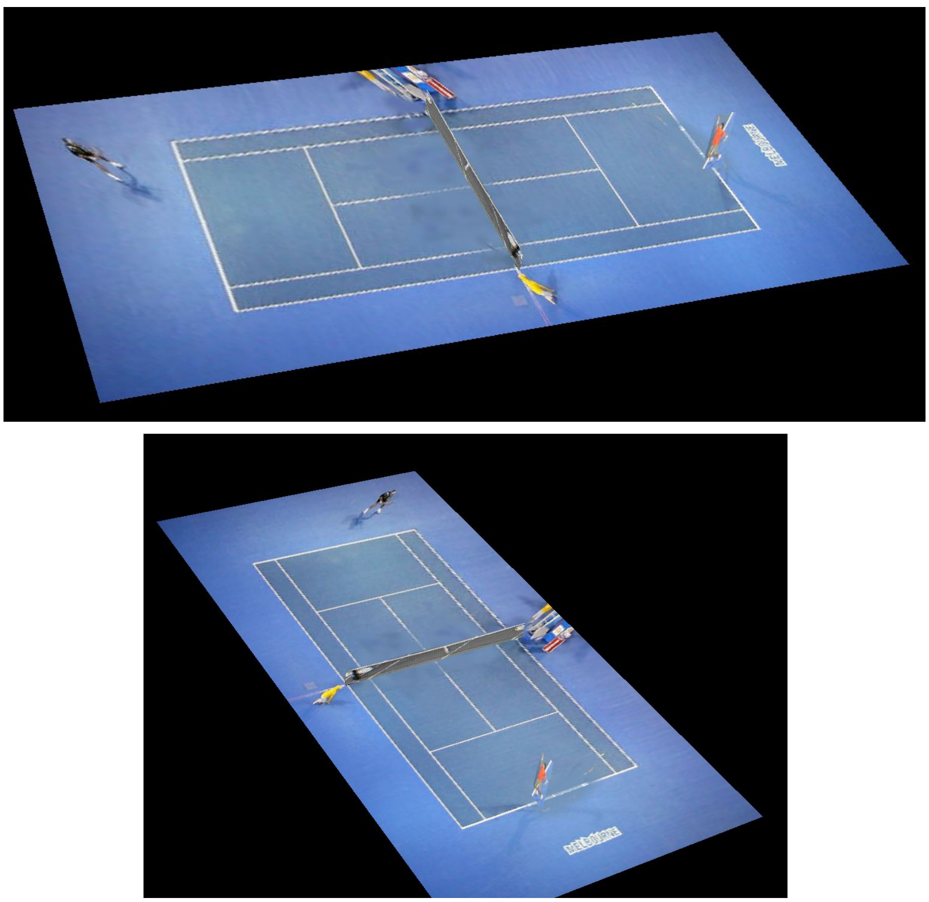

Fig 5. 3D Construction with texture

Part 2. Reconstruct bus using Mask R-CNN and point detector

Fig 6. Mask image from Mask R-CNN

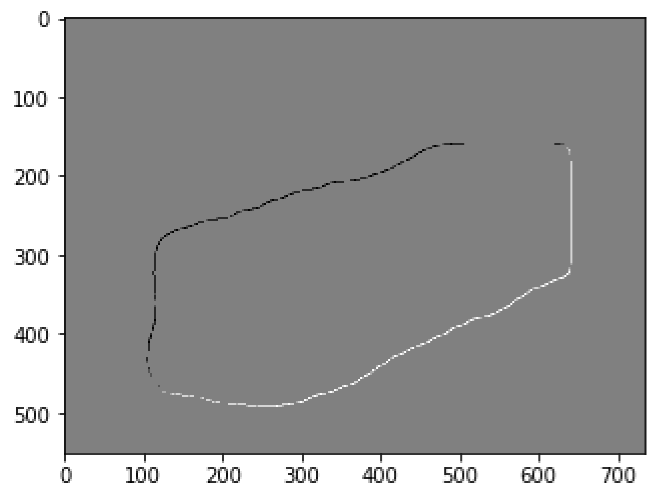

Fig 7. Margin of bus

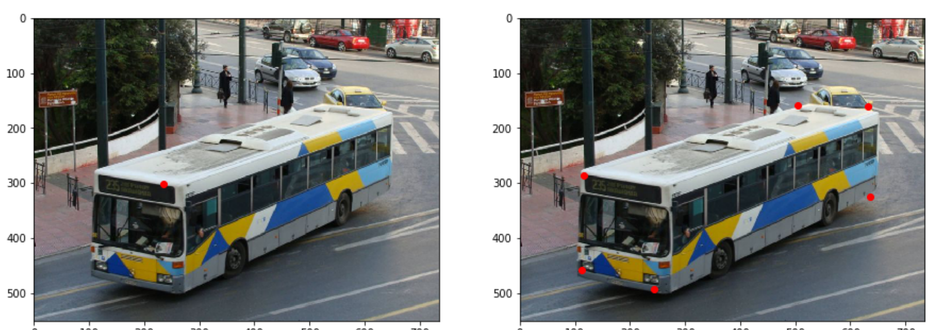

Fig 8. Computing the center point

Fig 9. Bus texture from homography

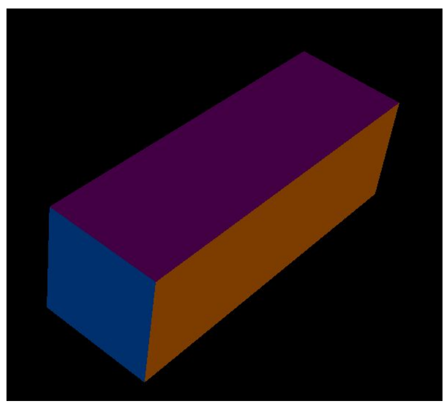

Fig 10. 3D Construction without texture

Fig 11. 3D Construction with texture

Discussion and conclusions

Line detection trial

In the part of finding vanishing points, we spend time on grouping lines to find the vanishing points. We have tried to use Hough line detection in CV2 to get the lines of tennis court; however, it is still arduous to group lines by vanishing points of the tennis court in that we cannot distinguish what lines belong to the right vanishing points, and what others belong to the left vanishing points. Therefore, for the tennis court, we still label the vertices manually to find vanishing points. And, we improve this difficult jobs with the help of Mask R-CNN, and use a bus image as an example.

Mask R-CNN input selection

Mask R-CNN can help us find the margin of objects; however, due to time limitation, we do not fine-tune the model to detect the tennis courts. Instead, we use another object that can be detected by the pretrained model, i.e. a bus image, for our improvement. When applying the neural networks into our project, we spend most of time finding the proper object from the COCO dataset because pretrained model can not detect objects we want. Also, in the COCO dataset, it is herculean to find a image for 3D reconstruction in that most images are overlap. Hence, we finally find a bus image to continue developing our automatic version of reconstructing 3D models.

Different origins to get texture

While doing the projection in tennis court case, we set two origins, one is at the corner of tennis court and the other is at the corner of the net. That is because if we use the same projection matrix one of them might not be preserved in the image. And, it is different from the original work [2], because if we just reconstruct a 3D model of a box image, we do not need to consider this points. In other words, our work is more general and suitable to reconstruct objects. It, however, still exists some future works. For example, our current work cannot reconstruct polygon objects. But, it can be improved by knowing vanishing points in advance. Then, we can pick up some points from the original images to find coordinates to reconstruct the 3D model.

Utilizing vanishing points to get larger region

To include more region of tennis court, we utilize the property of vanishing points. Since there is no obvious line or vertices located on the ground, we randomly select two points on the ground and form two lines from each points by connecting to X, Y vanishing points. Those two lines are parallel to the width and length of tennis court, so we calculate the intersection of two lines to be the new origin that contains more region.

To sum up, traditional methods, such as single-view reconstruction, can be combined with some state-of-the-art methods, like Mask R-CNN, to produce better and more user-friendly algorithm. Since Mask R-CNN renders a desirable results on targeting particular objects, it helps edge detectors to find more precise locations. So, by combined methods like our work, we believe that in the future more complicated methods can do many things in a one shot. If the time permit, we think we can make the whole algorithm automatically on reconstructing most objects, like the tennis courts, instead of limiting in Rod Laver Arena.

Statement of individual contribution

Hsiu-Yao Chang (hsiuyao2): Implement point detector and apply Mask R-CNN
Bo-Rong Chen (borongc2): Homography, Projection Transformation, Inspire the project
Hsin-Yu Hou (hsinyuh2): 3D construction, OpenGL, Model/Texture construction

References

[1] Mask R-CNN. https://github.com/matterport/Mask_RCNN

[2] 3D-Reconstruction-using-Single-View-Metrology.
https://github.com/kalyanghosh/3D-Reconstruction-using-Single-View-Metrology