CS280 - Computer Vision - Spring 2023 - Assignment 1
Due: Tuesday, Feb 7, 2023, 11:59pm

1 Perspective Projection (15 points)

Humans can estimate the spatial layout of a scene from a single image. Can computers also do this? In computer vision, we have made some progress towards solving this problem (i.e. recovering depth/spatial layout from a single image; interested students may want to have a look at [1]). There are various methods through which spatial layout can be recovered from a single image. Some of these methods rely on the knowledge of perspective projections and vanishing points. Here, we will further our understanding of perspective projections.

1. (5 points) Show that the vanishing points of lines on a plane lie on the vanishing line of the plane.

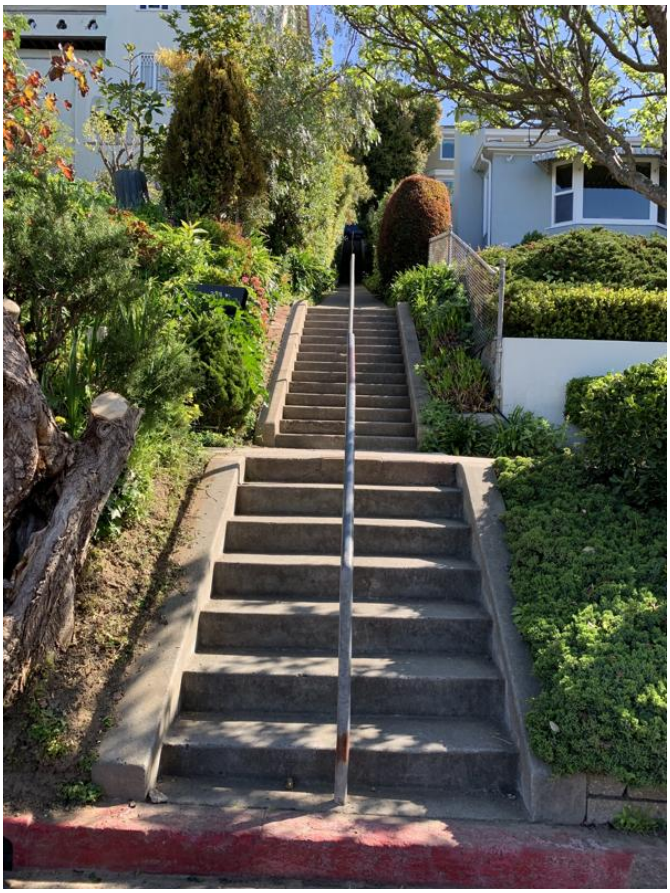
2. (5 points) Show that, under typical conditions, the silhouette of a sphere of radius r with center (X,0, Z) under planar perspective projection (XY is the image plane, and the center of projection is at the origin) is an ellipse of eccentricity $X/\sqrt{(X^2 + Z^2 - r^2)}$. Are there circumstances under which the projection could be a parabola or hyperbola? (Hint: See the definition of conic sections at: `http://en.wikipedia.org/wiki/Conic_section#Features`)

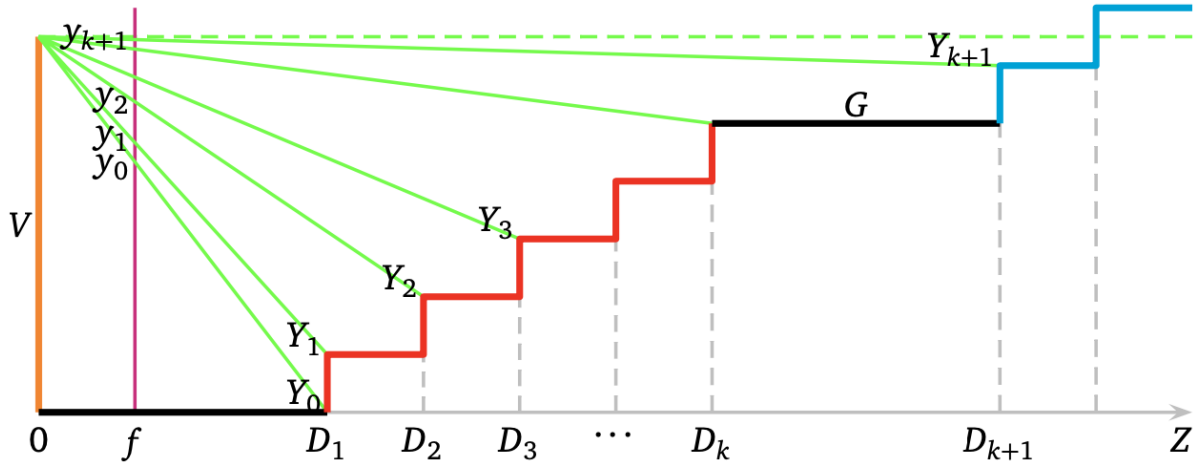
3. (5 points) An observer of height h is standing on a ground plane looking straight ahead. We want to calculate the accuracy with which she will be able to estimate the depth Z of points on the ground plane, assuming that she can visually discriminate angles to within $1'$. Derive a formula relating depth error δZ to Z. For simplicity, just consider points straight ahead of the observer($x = 0$). Given a Z value (say 10 m), your formula should be able to predict the δZ.

2 Visual Metrology: Infer Scene Geometry (15 points)

You are at the foot of a stair path and ready to walk up. From a single image taken of the stair path straight on (shown below with a schematic side view of the geometrical setup), is it possible to figure out how steep the stair path is and how much walk distance between the two flights? We assume that:

- All stairs are level with respect to the ground, and are of the same height H, width W and depth D.
- The camera is level and pointing straight at the stair path. That is, its optical axis is parallel to the ground plane and orthogonal to the front of the stairs.
- The camera's focal length f is known, its height V above ground is known, and the pixel size is known.
- Each stair can be segmented in the image, and its height and width can be measured.





1. (3 points) How is the steepness, or slope, of a flight of stairs related to the stair height H, width W, and depth D?

2. (3 points) Let D_t denote the scene depth of t-th stair from the camera. Let Y_t denote the elevation of the t-th stair with respect to the camera. Suppose each stair is of width W, height H, and depth D. For the first flight of k stairs, write down the 3D scene point equations for Y_t and D_t, $t = 0, 1, 2, \ldots, k$. Note the initial conditions at $t = 0$: $Y_0 = -V$, and D_0 denotes the depth of the foot of the stair, i.e. $D_0 = D_1$ and D_1 is marked in the diagram.

3. (3 points) The projection of Y_t in the image is y_t. What is the equation relating the 2D image point to the 3D scene point?

4. (3 points) The distance between the $(t + 1)$-th stair and the base of the staircase can be measured in the image, i.e., $y_{t+1} - y_0$. How is this 2D image measurement related to the 3D scene points, i.e., (Y_{t+1}, D_{t+1}) and (Y_0, D_0)?

5. (3 points) There is a gap, G, between two flights of stairs that you would need to walk through. After the first flight of k stairs, the first stair in the second flight is (Y_{k+1}, D_{k+1}) in 3D. What is its projection, y_{k+1}, in the image?

3 Training a Deep Neural Network (15 points)

In the following assignment, we want to train a convolutional network and visualize what it learns. The CIFAR-10 PyTorch tutorial is a good reference[1], if you are unsure of how to start. This assignment does not require access to GPUs, but you can check out Google Colab[2] if you want free GPU compute. **Please submit your code along with a PDF including your plots and visualizations.**

1. **Dataset:** Obtain the CIFAR-10 dataset [3]. The easiest way to do this is as presented in the PyTorch tutorial.
2. **Network Architecture:** Construct a lighter and smaller Residual Network (ResNet) architecture. A useful reference is the original ResNet paper by He et al [2]. (Some useful PyTorch code, for reference.[4]) Start with a ResNet of depth 18 and shrink it in depth and width (number of filters).
3. (10 points) Train your network to perform image classification by minimizng the cross-entropy between the network's prediction and the CIFAR-10 targets. Use your favorite optimizer to train your deep network. You should write out the definition of the cross-entropy loss function. Interpret it.
4. (5) As the training proceeds, plot the loss value and the classification accuracy on training and validation set, and choose the iteration with the least error.

[1] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
[2] https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c
[3] https://www.cs.toronto.edu/ kriz/cifar.html
[4] https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py

4 Visualize Your Networks (15 points)

Now we experiment with a few methods for gazing into the soul of the representations learned by your network.

4.1 Show your filters!

Visualize your filters by saving them as an image. For example, a layer with 64 filters of size 7x7x3 can be saved as 64 separate RGB images of size 7x7. Visualize these filters to identify any filters you learned about in class. Do this for two layers of the network.

1. (5 points) In the report, include a single image displaying all the filters from one layer.
2. (5 points) Select a few interesting ones and explain what these filters might be doing.

4.2 Embedding Space Visualization with tSNE

tSNE [3] is a popular method for nonlinear dimensionality reduction. The main idea is to project high dimensional data (e.g. features) to a 2d map, such that local structure (i.e. neighbors) is preserved. We will use tSNE to visualize the mapping learned by your networks. In particular, compute features of the layer before the classifier for 100 images, and use tSNE to project them to a 2d map. You can use the implementation of tSNE in scikit-learn.

1. (5 points) Plot the 2d map as a scatter plot and color the points according to their CIFAR class. Interpret it.

5 Instructions

1. This assignment is to be done individually.
2. Please submit the assignment using gradescope (Entry Code: Y7Z35V). Upload the following files:
   (a) `A PDF file`. The top of the first page should contain your name, student ID, and date of submission. The file should contain answers to all questions and all supporting images. Questions should be answered in order. Each problem should be on a new page.
   (b) `A tar/zip file`, containing any code you wrote for the assignment.
3. The HW is due on: Tuesday, Feb 7, 2023, 11:59pm.

References

[1] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 2000.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.