Assignment 4

CSC420 Introduction to Image Understanding

Author: Tingfeng Xia, University of Toronto

Date: Friday 4th December, 2020

I Deep Learning

We have the computation graph, written symbolically (with the corresponding variable names in the code `q1.py`)

$$\texttt{sum1} = \Sigma_1 = w_1 x_1 + w_2 x_2 \tag{I.1}$$
$$\texttt{sum2} = \Sigma_2 = w_3 x_3 + w_4 x_4 \tag{I.2}$$
$$\texttt{sigma1} = \sigma_1 = \sigma(\Sigma_1) \tag{I.3}$$
$$\texttt{sigma2} = \sigma_2 = \sigma(\Sigma_2) \tag{I.4}$$
$$\texttt{sum3} = \Sigma_3 = w_5 \sigma_1 + w_6 \sigma_2 \tag{I.5}$$
$$\texttt{sigma3} = \sigma_3 = \sigma(\Sigma_3) \tag{I.6}$$
$$\texttt{yhat} = \hat{y} = \sigma_3 \tag{I.7}$$
$$\texttt{L} = L = \|\hat{y} - y\|_2^2 \tag{I.8}$$

In `q1.py`, I calculated the forward pass and then back propagated the error signal. It prints the following result, including intermediate values:

```
-> Initialization ... <-
x1 = 0.90000, x2 = -1.10000, x3 = -0.30000, x4 = 0.80000
w1 = 0.75000, w2 = -0.63000, w3 = 0.24000,
w4 = -1.70000, w5 = 0.80000, w6 = -0.20000
target y = 0.50000
-> Start Forward Pass <-
sum1 = 1.36800, sigma1 = 0.79706, sum2 = -1.43200
sigma2 = 0.19279, sum3 = 0.59909, sigma3 = 0.64545
yhat = 0.64545, L = 0.02116
-> Start Back Propagation <-
dLdL = 1.00000, dLdyhat = 0.29090, dLdsigma3 = 0.29090
dLdsum3 = 0.06657, dLdsigma2 = -0.01331, dLdsum2 = -0.00207
-> End Result: dLdw3 = 0.0006215780
```

Then, our desired final result is

$$\frac{\partial L}{\partial w_3} = .0006215780 \tag{I.9}$$

II Camera Models

II.I Part 1 First we expand as hinted,

$$p = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = KP = K \begin{bmatrix} X_0 + td_x \\ Y_0 + td_y \\ Z_0 + td_z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 + td_x \\ Y_0 + td_y \\ Z_0 + td_z \end{bmatrix} \tag{II.1}$$

$$= \begin{bmatrix} f(X_0 + td_x) + p_x(Z_0 + td_z) \\ f(Y_0 + td_y) + p_y(Z_0 + td_z) \\ Z_0 + td_z \end{bmatrix} \tag{II.2}$$

then we can solve for x, y in where $p = (wx, wy, w)^\top$. We have

$$x = \frac{wx}{w} = \frac{f(X_0 + td_x) + p_x(Z_0 + td_z)}{Z_0 + td_z} \tag{II.3}$$

and

$$y = \frac{wy}{w} = \frac{f(Y_0 + td_y) + p_y(Z_0 + td_z)}{Z_0 + td_z} \tag{II.4}$$

which are (parametric) coordinates for the line on the image plane. To find the pixel coordinates o the vanishing point corresponding to the line, it suffices to take limit of t tends to infinity,

$$x_v = \lim_{t\to\infty} \frac{f(X_0 + td_x) + p_x(Z_0 + td_z)}{Z_0 + td_z} = \lim_{t\to\infty} \frac{ftd_x + p_x td_z}{td_z} = \frac{fd_x + p_x d_z}{d_z} \tag{II.5}$$

and

$$y_v = \lim_{t\to\infty} \frac{f(Y_0 + td_y) + p_y(Z_0 + td_z)}{Z_0 + td_z} = \lim_{t\to\infty} \frac{ftd_y + p_y td_z}{td_z} = \frac{fd_y + p_y d_z}{d_z} \tag{II.6}$$

Then, the vanishing point is, on the image plane, (x_v, y_v), defined everywhere except for when $d_z = 0$, in which case vanishing point does not exist.

II.II Part 2 What we have shown in Part 1 is true for all d. Now that we want to show for all lines that are on the plane, then it suffices to constraint d such that $\langle n, d\rangle = 0$, where n is the normal vector to the plane. This gives us an extra equation $n_x d_x + n_y d_y + n_z d_z = 0$ Our goal here is to show that $\alpha x_v + \beta y_v =$ constant that is independent of d for some α, β. (Notice that x_v, y_v is already independent of P, the starting point.) We can start with the inner product constraint, which is

$$n_x d_x + n_y d_y + n_z d_z = 0 \tag{II.7}$$

then since $n_z d_z \neq 0$ (otherwise there is no point in proving this) we have

$$\begin{aligned} & n_x d_x + n_y d_y + n_z d_z = 0 && \text{(II.8)} \\ \implies & \frac{n_x d_x + n_y d_y}{n_z d_z} + 1 = 0 && \text{(II.9)} \\ \implies & \frac{n_x}{n_z}\frac{d_x}{d_z} + \frac{n_y}{n_z}\frac{d_y}{d_z} + 1 = 0 && \text{(II.10)} \\ \implies & \frac{d_x}{d_z} = -\frac{n_z}{n_x} - \frac{n_z n_y d_y}{n_x n_z d_z} && \text{(II.11)} \\ \implies & \frac{d_x}{d_z} = -\frac{n_z}{n_x} - \frac{n_y d_y}{n_z d_z} && \text{(II.12)} \end{aligned}$$

We can substitute this into what we had in part 1, i.e.

$$\begin{aligned} x_v &= \frac{f d_x + p_x d_x}{d_z} = f\frac{d_x}{d_z} + p_x && \text{(II.13)} \\ &= f\left(-\frac{n_z}{n_x} - \frac{n_y d_y}{n_z d_z}\right) + p_x && \text{(II.14)} \\ &= -f\frac{n_z}{n_x} - f\frac{n_y d_y}{n_z d_z} + p_x = (\dagger) && \text{(II.15)} \end{aligned}$$

also,

$$y_v = \frac{f d_y + p_y d_z}{d_z} = f\frac{d_y}{d_z} + p_y \implies \frac{d_y}{d_z} = \frac{y_v - p_y}{f} \tag{II.16}$$

Then,

$$\begin{aligned} y_v = (\dagger) &= -f\frac{n_z}{n_x} - f\frac{n_y(y_v - p_y)}{n_z f} + p_x && \text{(II.17)} \\ &= \frac{f n_z}{n_x} - \frac{n_y y_v}{n_z} + \frac{n_y p_y}{n_z} + p_x && \text{(II.18)} \end{aligned}$$

from where we can simplify

$$\begin{aligned} & n_x n_z x_v = -f n_z n_z - n_y n_x y_v + n_x n_y p_y + n_x n_z p_x && \text{(II.19)} \\ \implies & \underbrace{n_x n_z}_{\alpha} x_v + \underbrace{n_y n_x}_{\beta} y_v = \underbrace{-f n_z n_z + n_x n_y p_y + n_x n_z p_x}_{\text{constant independent of } d} && \text{(II.20)} \end{aligned}$$

which has exactly the form that we want in the first place, and it illustrates the linear relationship between x_v and y_v.

III Projection

III.I Part 1 Please check my implementation in file `q3.py`, which is based on the starter code provided. I shall present the derivation used to solve for X, Y inside the code. Recall that the projection matrix is defined as

$$\mathrm{P} = \underbrace{\begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsics } K} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\underbrace{\begin{bmatrix} \mathrm{R}_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}}_{\text{rotation}} \underbrace{\begin{bmatrix} \mathrm{I}_{3\times3} & \mathrm{T}_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}}_{\text{translation}}}_{\begin{bmatrix} R & t \end{bmatrix} = \begin{bmatrix} R & -c \end{bmatrix}} \tag{III.1}$$

which is a known 3 by 4 real matrix. Let's now write out the relationship between what we have and what we want

$$P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11}X + p_{12}Y + p_{13}Z + p_{14} \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} \\ p_{31}X + p_{32}Y + p_{33}Z + p_{34} \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ a \end{bmatrix} \tag{III.2}$$

where a, ax, ay and all the $p_{(..)}$ are known and X, Y, Z are unknowns. We can obtain the following solvable linear relationship

$$\underbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}}_{\text{known}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{unknown}} = \underbrace{\begin{bmatrix} ax - p_{14} \\ ay - p_{24} \\ a - p_{34} \end{bmatrix}}_{\text{known}} \tag{III.3}$$

Also note that due to the choice of Z axis is $(0, 0, -1)$, I had to flip Y direction in my code. Figures III.1, III.2, III.3 are screenshots of the output. Since they are 3D, it would be the best if you run the code and view it in your browser, but the screenshot can give you a rough idea without having to execute the code.

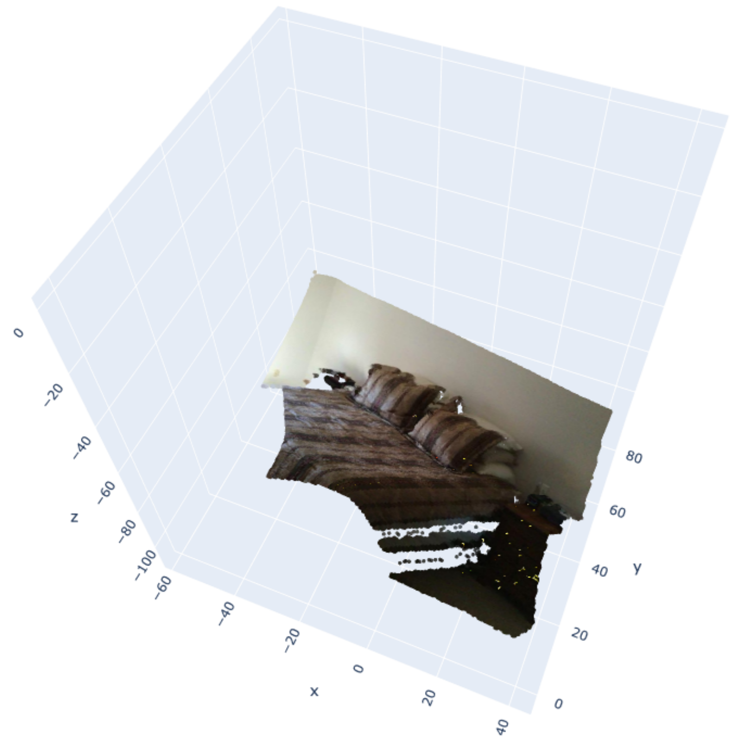

Figure III.1: Input 1, screenshot of reconstruction

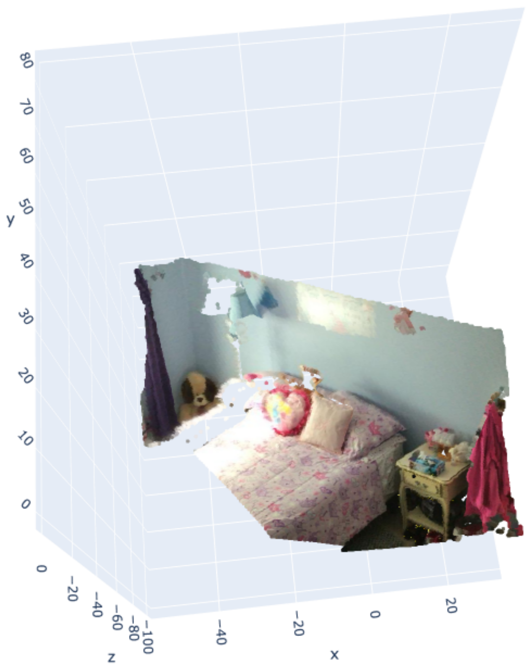

Figure III.2: Input 2, screenshot of reconstruction

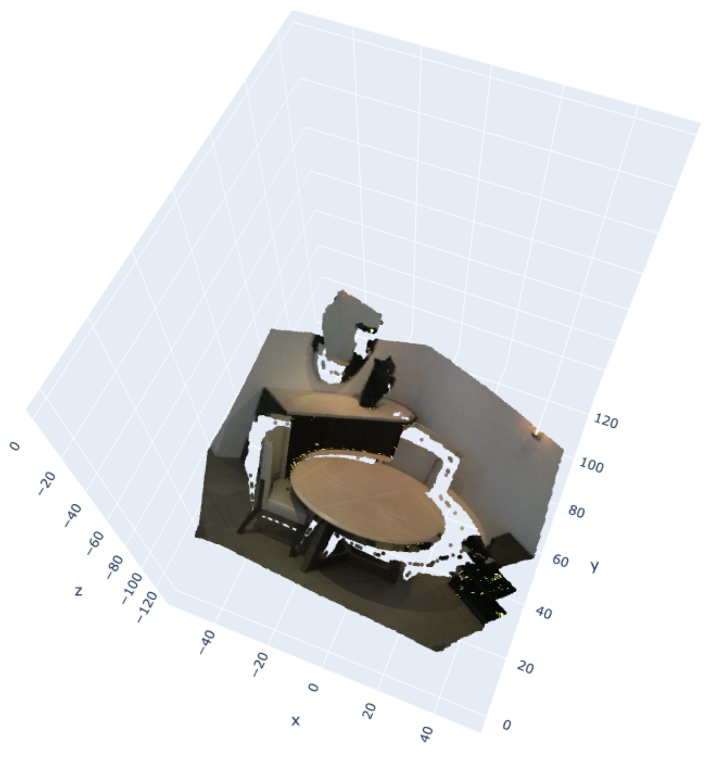

Figure III.3: Input 3, screenshot of reconstruction