AA 274A: Principles of Robot Autonomy I
Problem Set 2

Name: Li Quan Khoo
SUID: lqkhoo (06154100)

October 13, 2020

Problem 1: Camera Calibration

(i) (code)

(ii) (code)

(iii) (code)

(iv) (code)

(v) (code)

Problem 2: Line Extraction

(i) (code)

(ii) TODO

Problem 3: Linear Filtering

(i) (a)

$$F = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \tag{1}$$

(b)

$$F = \begin{bmatrix} 2 & 3 & 0 \\ 5 & 6 & 0 \\ 8 & 9 & 0 \end{bmatrix} \tag{2}$$

(c) This kernel is doing discrete difference (differentiation) of the image along the horizontal axis. It could be used to perform edge detection tasks.

$$F = \begin{bmatrix} 2 & 2 & -2 \\ 5 & 2 & -5 \\ 8 & 2 & -8 \end{bmatrix} \tag{3}$$

(d) This is an isotropic, normalized Gaussian kernel performing blurring on the image. It could be used to filter out high frequency information on either axis.

$$F = \frac{1}{16}\begin{bmatrix} 21 & 36 & 33 \\ 52 & 80 & 68 \\ 57 & 84 & 69 \end{bmatrix} \tag{4}$$

(ii) I'm assuming that the question is alluding to the fact that performing correlation (or convolution) over an input of depth d means we end up summing over d individual correlations or convolutions performed on individual input channels.

This is saying that:

$$G(i,j) = \sum_{w} \left(\sum_{u} \sum_{v} F_w(u,v) \cdot I_w(i+u, j+v) \right) \tag{5}$$

where $G(i,j)$ is a correlation operation at position i, j of the image, and F and I are flattened vector representations of the kernel and current image patch (including padding) that we are operating over. Therefore, writing out the matrices explicitly:

$$G(i,j) = \sum_{w} \begin{bmatrix} - & F_w^{\mathsf{T}}(u,v) & - \end{bmatrix} \begin{bmatrix} | \\ I_w \\ | \end{bmatrix} \tag{6}$$

which is of course equal to taking the dot product of f, which is a single big vector f of length $u \cdot v \cdot w$ with a single big vector $t(i,j)$ of length $u \cdot v \cdot w$ as expressed below:

$$G(i,j) = \begin{bmatrix} F_1^{\mathsf{T}} & \dots & F_w^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} I_1 \\ \vdots \\ I_w \end{bmatrix} = f^{\mathsf{T}} t_{i,j} \tag{7}$$

(iii) (code)

(iv) Naive implementation as above using a loop: Runtimes are 0.79, 1.36, 0.77, 0.80 seconds.

Vectorized implementation batching all image pixels: Runtimes are 0.13, 7.55, 0.12, 0.34 seconds.

To answer the first hint, no it does not. For a mono-channel image, each individual patch could be flattened and stacked into a $u \cdot v$ by $h \cdot w$ array and processed in parallel.

To answer the second hint, the total number of addmul operations are $u \cdot v \cdot w \cdot h$ as we are applying a filter with a receptive field of u by v over a single-channel input of size w by h. If the filter could be expressed as an outer product, the total cost would be $(u + v) \cdot w \cdot h$.

Lastly, we could implement Winograd's minimal filtering algorithm that pre-computes intermediate values that depend only on kernel weights with the motivation of saving redundant computation.

(v) We use the result that any $m \times n$ matrix of rank 1 could be expressed as a vector outer product uv^{T}. This is obvious, because the column rank of any vector u has to be equal to 1. To answer the question, if we know that a matrix is rank 1 and we simply wish to recover u and v^{T}, these correspond to the orthogonal matrices after performing SVD on the original matrix. Alternatively, u could be any of its columns and v is the single nonzero row left over after performing Gaussian elimination, up to a constant factor k.

Additionally, it is easy to see that for any $m \times n$ matrix of rank r, we can express it as a linear combination of r matrices, which themselves could be expressed as the outer product of the linearly independent rows and columns of the original matrix. This is a generalization of the above result.

(vi) (code)

(vii) Convolution with a flipped filter in all its dimensions would produce the same output as correlation with an unmodified filter.

In other words,

$$G(i,j) = \sum_{u=1}^{k}\sum_{v=1}^{l} F(u,v) \cdot I(i-u, j-v) = \sum_{u=1}^{k}\sum_{v=1}^{l} F(k-u, l-v) \cdot I(i+u, j+v) \tag{8}$$

Problem 4: Template Matching

(i) (code)

(ii) (code)

(iii) TODO

Problem 5: Stop Sign Detection and FSM in ROS

(i) TODO

(ii) (code)

(iii) TODO

(iv) TODO

(v) null

(vi) TODO

(vii) (code)

(viii) TODO

Extra Problem: Image Pyramids

(i) (code)

(ii) TODO

(iii) (code)

(iv) TODO

(v) (code)

(vi) TODO

(vii) (code)

(viii) (code)

(ix) TODO