CSC420: Introduction to Image Understanding
Assignment 1 Solutions

Ioan Andrei Bârsan
iab@cs.toronto.edu

September 29, 2017

Problem 1

Jake will write the grading annotations for this assignment on MarkUS.

(a) The concept of a 1D impulse $\delta(t)$ can be extended to 2D as follows:

$$\delta(u,v) = \begin{cases} 1, & u = 0 \wedge v = 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

(b) The notion of impulse shifting can also be extended to represent a 2D impulse located at $u = m, v = n$:

$$\delta'(u,v) = \begin{cases} 1, & u = m \wedge v = n \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Alternatively, the following definition would also work (and is even somewhat more consistent with the solution for (d)).

$$\delta'(u,v) = \delta(u-m, v-n). \tag{3}$$

(c) The code for producing the 2D impulses is straightforward; they simply correspond to values of 1 in an image which is otherwise all zeros. Example results can be seen in Figure 1, and the code to generate that plot is shown in Listing 2.

(d)

$$I(u,v) = \sum_{i=1}^{M} \sum_{j=1}^{N} I(i,j)\delta(u-i, v-j). \tag{4}$$

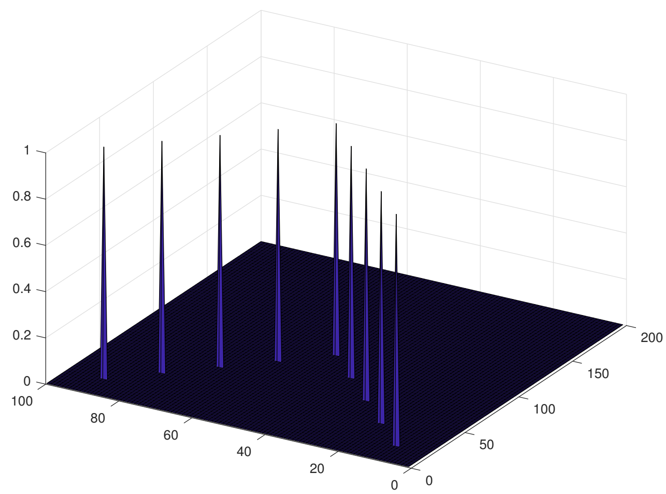

Figure 1: A 3D plot of an image of zeros containing a few impulses.

```
img = zeros(100, 200);

img(10, 20) = 1;
img(20, 40) = 1;
img(30, 60) = 1;
img(40, 80) = 1;
img(50, 100) = 1;
img(60, 80) = 1;
img(70, 60) = 1;
img(80, 40) = 1;
img(90, 20) = 1;

surf(img);
```

Figure 2: The MATLAB code for generating Figure 1.

Problem 2

Jake will write the grading annotations for this assignment on MarkUS.

(a) $n \times n$ image, $m \times m$ filter: the general case computational complexity of the $h * I$ convolution is $\mathcal{O}(m^2 n^2)$.

(b) If the filter is separable, then the complexity is $\mathcal{O}(mn^2)$ (the constant factor 2 is dropped from the big O notation).

(c)
- $\mathbf{F}_1$ is not separable, as highlighted by its singular value decomposition, which yields three nonzero singular values (i.e., rank($\mathbf{F}_1 = 3$). This means that the filter matrix cannot be expressed as the outer product of two vectors. This calculation can be performed very easily using either numpy or MATLAB's `svd` function. The values of the singular values are $\sigma_0 = 43.65$, $\sigma_1 = 15.18$, and $\sigma_2 = 0.03$.
- $\mathbf{F}_2$ is separable, since its singular value decomposition yields just one nonzero singular value ($\sigma_0 = 13.08$). The filter can therefore be expressed as the outer product of two vectors. Infinitely many solutions exist, such as

$$\mathbf{F}_2 = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} 2 & 1 & 2 \end{bmatrix} \tag{5}$$

or

$$\mathbf{F}_2 = \begin{bmatrix} 2.4888 \\ -0.8296 \\ -2.4888 \end{bmatrix} \begin{bmatrix} -2.4108 & -1.2054 & -2.4108 \end{bmatrix}. \tag{6}$$

The solution can be computed either by hand, or using the output of the singular value decomposition.

Problem 3

Andrei will write the grading annotations for this assignment on MarkUS.

Main part

Please see the attached MATLAB code. The threshold used is 0.678, and the results can be seen in Figure 3. 13/19 digits can be detected. Increasing the threshold can lead to more true positives, but at the cost of increasing the false positive rate.

Extra credit

Performance can be improved by using templates cropped from the image, as these templates are better tuned to our application domain. For instance, the digits in the given image are slightly rotated, unlike the templates, leading to slight imprecisions during detection. Using templates cropped from the image can alleviate this issue.

Figure 3: Digit detection result using $T = 0.678$.

For the implementation, we expect the student's results to be slightly better than those produced using the standard digit templates.