ELEN 4810 Homework 2

Due **Friday, October 10, 5 PM**. Please submit your responses via CourseWorks. Please combine your responses to the analytical and computational problems in a single pdf file.

Please submit a single PDF of name `yourunihere hw02.pdf` - e.g., "`abc1234 hw02.pdf`" containing responses to both analytical and computational problems. For the coding problems, submit your solution as `yourunihere hw02.m` - e.g., "`abc1234 hw02.m`".

Thanks!

Analytical Questions

Please complete problems 2.36, 2.62, 2.71 in Oppenheim and Schafer (3rd Edition). Justify your answers!

Computational Question

Finding Waldo

In the lecture, we briefly introduced the correlation operator. The purpose of this problem is to understand the relationship between correlation and convolution, and to do some very simple image processing. You have been provided with two images `waldo.jpg` and `whereswaldo.jpg`. Here we will try to match the *template image* `waldo.jpg` - a profile of Waldo - to the original *reference image* `whereswaldo.jpg`.

Correlation operations

Notation. In describing the correlation operator the following notation is used. For the sake of clarity signals are denoted with bold font.

- The *support* of a 2D (possibly complex) signal $\mathbf{x}$ is the set of indices

$$\mathrm{supp}\,(\mathbf{x}) \doteq \{m, n : \mathbf{x}[m,n] \neq 0\}\,.$$

- We say $\mathbf{x}$ is *supported in* the index set S if $\mathrm{supp}(\mathbf{x}) \subseteq S$, and let $\mathbf{x}_S$ denote the signal

$$\mathbf{x}_S[i,j] \doteq \begin{cases} \mathbf{x}[i,j], & \text{if } [i,j] \in S \\ 0 & \text{elsewhere.} \end{cases}$$

- Denote the *inner product*

$$\langle \mathbf{x}, \mathbf{y} \rangle \doteq \Sigma_{i,j} \mathbf{x}[i,j] \mathbf{y}^*[i,j]$$

and the ℓ_2-norm by

$$\|\mathbf{x}\|^2 \doteq \langle \mathbf{x}, \mathbf{x} \rangle = \Sigma_{i,j} |\mathbf{x}[i,j]|^2 .$$

- The *shift of* $\mathbf{x}$ *by* (p,q) is denoted by

$$\left(\mathcal{D}_{p,q}^{-1} x\right)[m,n] \doteq \mathbf{x}[m+p, n+q].$$

- Finally, for a positive integer M we define an index set $[M]_* \doteq \{0, \ldots, M-1\}$. Note that this description is using zero-based indexing, which is different from MATLAB.

Correlation. Let $\mathbf{a}$ denote the template image supported in $S = [M]_* \times [N]_*$, and $\mathbf{b}$ denote the larger reference image in which we would like to detect the template. The *cross-correlation* between $\mathbf{a}$ and $\mathbf{b}$ is a signal given by the inner product between signal shifts

$$\mathbf{r}_{\mathbf{a},\mathbf{b}}[p,q] \doteq \left\langle \mathbf{a}, \mathcal{D}_{p,q}^{-1}\mathbf{b} \right\rangle \tag{1}$$

$$= \sum_{i,j} \mathbf{a}[i,j] \cdot \mathbf{b}^*[i+p, j+q]. \tag{2}$$

Our signals of interest will be real-valued, so $\mathbf{b}^* = \mathbf{b}$, and since $\mathbf{a}[i,j] = 0$ for $(i,j) \notin S$,

$$\mathbf{r}_{\mathbf{a},\mathbf{b}}[p,q] = \sum_{i \in [M]_*, j \in [N]_*} \mathbf{a}[i,j] \cdot \mathbf{b}[i+p, j+q]. \tag{3}$$

This can be implemented in MATLAB directly using the function `xcorr2()` or by using the convolution function `conv2()`. Both approaches result in a matrix corresponding to the possible shifts (p,q) such that $\mathbf{r}_{\mathbf{a},\mathbf{b}}[p,q] \neq 0$. This is necessarily a finite window. Why?

The normalized correlation. The correlation can be useful in certain situations, but what we would really like for template matching is a function $\mathbf{f}_{\mathbf{a},\mathbf{b}}[p,q]$ which is maximized at the locations (p,q) where the reference and template images match or "have the same pattern".

Unfortunately, cross-correlation fails this criterion – the correlation of a nonnegative template with a nonnegative image will be large anywhere the image is bright. To obtain a more suitable objective function, recall that for two vectors $\mathbf{u}$ and $\mathbf{v}$, the *normalized* inner product $\left\langle \frac{\mathbf{u}}{\|\mathbf{u}\|_2}, \frac{\mathbf{v}}{\|\mathbf{v}\|_2} \right\rangle = \cos\angle(\mathbf{u}, \mathbf{v})$. This is maximized iff $\mathbf{u} = c \cdot \mathbf{v}$ for some $c > 0$, which seems much closer to what we want.

Applying the normalized inner product between $\left(\mathcal{D}_{p,q}^{-1}\mathbf{b}\right)_S$ and $\mathbf{a}$ gives rise to the *normalized correlation*

$$\chi_{\mathbf{a},\mathbf{b},}[p,q] = \left\langle \frac{(\mathcal{D}_{p,q}^{-1}\mathbf{b})_S}{\|(\mathcal{D}_{p,q}^{-1}\mathbf{b})_S\|}, \frac{\mathbf{a}}{\|\mathbf{a}\|} \right\rangle \tag{4}$$

$$= \frac{\sum_{i \in [M]_*, j \in [N]_*} \overline{\mathbf{a}}[i,j] \cdot \mathbf{b}[i+p, j+q]}{\sqrt{\sum_{i,j} |\mathbf{a}[i,j]|^2} \cdot \sqrt{\sum_{i' \in [M]_*, j' \in [N]_*} |\mathbf{b}[i'+p, j'+q]|^2}}. \tag{5}$$

Notice that the inner product is taken with $\mathbf{b}$ restricted to the appropriate window. The normalized correlation can be implemented by normalizing the output of `xcorr2()` or `conv2()`. In particular, notice that the term

$$\sum_{i' \in [M]_*, j' \in [N]_*} |\mathbf{b}[i' + p, j' + q]|^2$$

can also be written as a cross-correlation. *Hint*: Take $\mathbf{r}_{\mathbf{1}_S, |\mathbf{b}|^2}$, where $\mathbf{1}_S[i, j]$ is equal to 1 for $(i, j) \in S$ and zero elsewhere.

Specification

Please implement the function in the skeleton file `detectcorr.m`, which attempts to find Waldo in the input image. Its usage is as follows:

```
NCcorr = detectcorr(template name, reference name),
```

where `template_name` and `reference_image` should be the names of the template and reference images. The given skeleton code first creates a *double* array representation of both images by applying `imread()` and `im2double()`, then using `rgb2gray()` to convert the images to grayscale. Next, the function should compute the normalized correlation `Ccorr` by normalizing the output of `corr()` between the two images. Figure 1 shows an example of a correct detection using normalized correlation maximization:

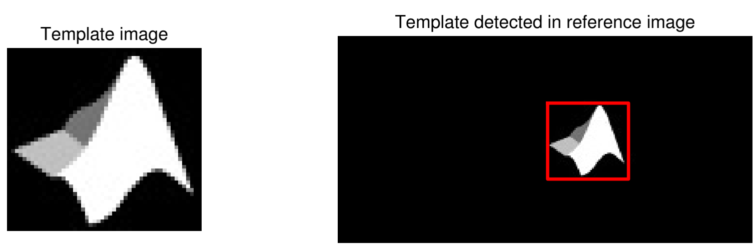

Figure 1: Template detection. Left: template image. Right: reference image, with detected template bounding box.

Observations (don't need to submit these)

- Enter the following commands:
  - `Ccorr = detectcorr('waldo.jpg', 'whereswaldo.jpg');`
  - `imagesc(Ccorr); colorbar;`

  Find the point achieving the maximum. Is it easily distinguishable in the correlation image? What are some possible explanations for this? Potential consequences in practical applications?
- Try adding different levels of random noise to the reference using `sqrt(noisepwr)*randn()`. Does `detectcorr()` still work?
- What would happen if Waldo is rotated in either the template or reference image?
- How can you make the detector more robust against the aforementioned issues?