ECE 285 – Assignment #3

Writing part

---

1 Exercise 1 (Eigenvectors of circulant matrices)

Show that the n eigenvectors, with unit norm, of any circulant matrix H

$$H = \begin{pmatrix} \nu_0 & \nu_{n-1} & \nu_{n-2} & \dots & \nu_2 & \nu_1 \\ \nu_1 & \nu_0 & \nu_{n-1} & \nu_{n-2} & \dots & \nu_2 \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ \nu_{n-1} & \nu_{n-2} & \dots & \nu_2 & \nu_1 & \nu_0 \end{pmatrix}$$

reads, for $k = 0$ to $n-1$, as

$$e_k = \frac{1}{\sqrt{n}} \left(1, \exp\left(\frac{2\pi i k}{n}\right), \exp\left(\frac{4\pi i k}{n}\right), \exp\left(\frac{6\pi i k}{n}\right), \dots, \exp\left(\frac{2(n-1)\pi i k}{n}\right) \right)^T .$$

Identify the corresponding eigenvalues λ_k.

Recall that the n eigenvectors (e_k) with unit norm of a matrix H must satisfy:

$$He_k = \lambda_k e_k, \quad \langle e_k, e_l \rangle = e_k^* e_l = 0 \quad \text{if} \quad k \neq l \quad \text{and} \quad \|e_k\|^2 = e_k^* e_k = 1 \ .$$

Hint: use the sum of a geometric series: For $r \neq 1$, $\displaystyle\sum_{p=a}^{b} r^p = \frac{r^a - r^{b+1}}{1 - r}$.

Practical part

---

2 Exercise 2 (Separable convolutions)

1. Modify `imconvolve_spatial.m` in order to implement cases where the kernel is separable. The function signature will be the following

   ```
   function xconv = imconvolve_spatial(x, nu, s1, s2, boundary, separable)
   ```

   Here `separable` is a string that can take one of the values: `'none'`, `'product'` or `'sum'`. When `separable = 'none'`, the behavior of this function is unchanged. To make this as the default behavior, add the following lines at the beginning of the function

   ```
   if ~exist('separable', 'var')
       separable = 'none';
   end
   ```

When `separable = 'product'`, `nu` is a cell-array composed of two handle functions, `nu = { nu1, nu2 }`, and the function performs the convolution of x by ν defined as

$$\nu(i, j) = \nu_1(i, 0)\nu_2(0, j).$$

Similarly, when `separable = 'sum'`, ν is defined as

$$\nu(i, j) = \nu_1(i, 0) + \nu_2(0, j).$$

Example: the following code

```
nu1   = imkernel('gaussian', tau, s1, 0);
nu2   = imkernel('gaussian', tau, 0,  s2);
nu    = { nu1, nu2 };
xconv = imconvolve_spatial(x, nu, s1, s2, 'mirror', 'product');
```

should produce the same result as

```
nu    = imkernel('gaussian', tau, s1, s2);
xconv = imconvolve_spatial(x, nu, s1, s2, 'mirror');
```

2. Download `assignment3.zip`, and extract the files

   - `assignment3/train.png`
   - `assignment3/race.png`

   Create a script `test_imconvolve_separable.m` that performs the convolution of the image `x = train` with the Gaussian and Box kernel. Compare the results and the execution times of the separable version and the non-separable one. Check that your results are consistent with the followings:

(a) Original x (b) x^{conv} non-separable (1.15s) (c) x^{conv} separable (0.08s)

Figure 1: Results of convolutions with a Gaussian kernel $\tau = 4$ and $s_1 = s_2 = 20$.

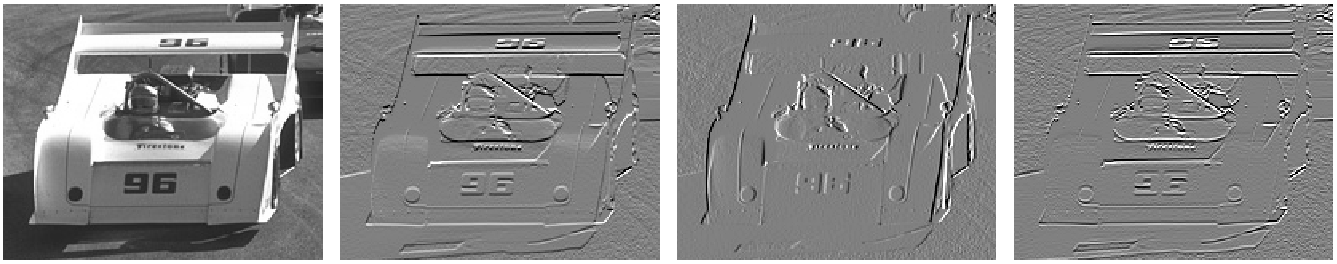
3 Exercise 3 (Derivative filters)

1. Complete `imkernel.m` such that it implements the following derivative filters: `grad1_forward` (forward discrete gradient in the first dimension), `grad1_backward`, `grad2_forward`, `grad2_backward`, `laplacian1` (Laplacian in the first dimension), `laplacian2` (Laplacian in the second dimension). Please refer to the class for proper definitions. For instance, the code for the forward discrete gradient in the second dimension reads as:

```
    case 'grad2_forward'
        nu = @(i, j) (+1 * (j == 1) + -1 * (j == 0)) .* (i == 0);
```

In this case, the arguments `tau`, `s1` and `s2` are ignored.

2. Create a script `test_derivative_filters.m`, that applies the 6 derivative filters on `x = race` and display the 6 results in a single figure. Use `linkaxes`, zoom and move on the images, and check that your results are consistent with Figure 2.



(a) Original x (b) $\nabla_1^{\text{forward}} x$ (c) $\nabla_2^{\text{backward}} x$ (d) $\Delta_1 x$

Figure 2: Results of derivative filters.

3. Create a script `test_derivative_filters_properties.m`, that, for `x = train` and `y = race`, compares the results of $\langle \nabla_1^{\text{forward}} x, y \rangle$ and $\langle x, \nabla_1^{\text{backward}} y \rangle$ for different boundary conditions.

   Next, compute $\| \nabla_1^{\text{backward}} \nabla_1^{\text{forward}} y - \Delta_1 y \|$ for different boundary conditions.

   What do you conclude?

4. Create in `imlaplacian.m`, the function

```
function l = imlaplacian(x, boundary)
```

   that returns the Laplacian of the image `x`.

   Hint: use `separable = 'sum'`.

5. Create in `imgrad.m`, the function

```
function g = imgrad(x, boundary)
```

   that for a $n_1 \times n_2$ image `x` returns a $n_1 \times n_2 \times 2$ array `g` corresponding to the discrete gradient vector field of `x` with periodical boundary conditions. More precisely, `g(:,:,1)` (resp., `g(:,:,2)`) corresponds to the forward discrete image gradient in the first (resp., second) direction.

6. The divergence of a two-dimensional vector field $f : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as the function $\operatorname{div} f : \mathbb{R}^2 \to \mathbb{R}$ such that

$$\operatorname{div} f = \frac{\partial f_1}{\partial s_1} + \frac{\partial f_2}{\partial s_2}.$$

   Create in `imdiv.m`, the function

```
function d = imdiv(f, boundary)
```

   that for a $n_1 \times n_2 \times 2$ vector field `f` returns its $n_1 \times n_2$ discrete divergence `d`. For periodical boundary conditions, the function should guarantee that

$$\operatorname{div} \nabla x = \Delta x.$$

   Check in `test_derivative_filters_properties.m`, that for periodical boundary conditions, $\| \operatorname{div} \nabla x - \Delta x \| = 0$ (up to machine precision) and compare $\langle \nabla x, \nabla y \rangle$ with $\langle \Delta x, y \rangle$. Conclude.