Task 1

Image processing in MATLAB, Fourier coefficients

Christian Riess (christian.riess@fau.de)

1 Supplements on Fourier Coefficients, Series and Transforms

A Fourier series expansion of a periodic function $f(t)$ with period $T > 0$ is possible, if f

- is continuous and piecewise continuously differentiable. Then, the Fourier series converges pointwise and homogeneously.
- has a bounded total variation over one period, and the function values of f at each position t correspond to the average of the limits towards this value from the left and the right. In this case, the Fourier series converges only pointwise.
- belongs to the function space $L^2([c, c + T])$, limited to a period $[c, c + T]$. In this case, the Fourier series is convergent with respect to the L^2 norm.

The function $f(t)$ can be represented by a series of sine and cosine functions. The frequencies of these functions are integer multiples of the fundamental frequency $\omega = 2\pi/T$:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + b_n \sin(n\omega t) \tag{1}$$

For practical applications, this sum is truncated after a finite number of elements. This yields an approximation of f, represented by a trigonometric polynomial with the coefficients

$$a_n = \frac{2}{T}\int_c^{c+T} f(t)\cos(n\omega t)dt; \quad n \geq 0 \ , \tag{2}$$

$$b_n = \frac{2}{T}\int_c^{c+T} f(t)\sin(n\omega t)dt; \quad n > 0 \ . \tag{3}$$

This way of writing the Fourier series expansion (or Fourier analysis, respectively) differs from the version in the lecture notes and the lecture. Validate that both versions are equivalent. Here is the first actual exercise questions:

1. What special form do the coefficients of a Fourier series have if f is an even function? What special form do the coefficients have if f is an odd function?

2 Basic image processing in Matlab

A code template and an example image for this task are available at the website of the exercises. In case you have never written matlab code before, it may be very helpful to first go through Exercise 0 on the web page. The execution of matlab code is optimized for matrix operations. Thus, matlab encourages a very particular programming style: express as many operations as possible in terms of matrix operations.

1. Load the image `lena.png`.
2. Use matlab's workspace to find out the data type of the image data. Use for this task only the green channel of the image, stored as `double` values. Rescale the values to a range of $[0;1]$.
3. Calculate the gradient along x and y direction using central differences and the gradient magnitudes (see pseudo code below). Do *NOT* use a `for` loop for the implementation. Scale the gradient images to $[-1;1]$ and the magnitude image to $[0;1]$. Visualize the images together with the source images with a suitable color mapping.
4. Store the gradient magnitudes as an image on the hard drive in the file `gradient.png`.

```
Image (size X,Y);
for x = 2:X-1
    for y = 2:Y-1
        xg = Image(x+1, y) - Image(x-1, y); // Gradient X
        yg = Image(x, y+1) - Image(x, y-1); // Gradient Y
        GradientMagnitude(x-1, y-1) = sqrt(xg * xg + yg * yg);
=> GradientMagnitude (size X-2, Y-2);
```

Useful functions in matlab: `imread, imwrite, imtool, sqrt`

3 Triangle and Saw Tooth

We define a triangle function

$$\text{triangle}(t) = |t| \tag{4}$$

on the domain $-\frac{\pi}{2} \leq t \leq \frac{\pi}{2}$ with period $T = \pi$, and a saw tooth function

$$\text{sawtooth}(t) = t \tag{5}$$

on the domain $-\pi \leq t \leq \pi$ with period $T = 2\pi$.

1. Plot these functions in matlab on the domain $-2\pi \leq t \leq 2\pi$.
2. Calculate **analytically** the coefficients of the Fourier series.
3. Implement functions in matlab that approximate triangle(t) and sawtooth(t) by their first n Fourier coefficients.
4. Plot the approximations together with the original (exact) functions.
5. Vary the number of approximation coefficients to better understand the convergence behavior of the approximations.
6. Compute the root mean quadratic differences (RMSD) and the maximum absolute differences between approximation and exact functions with respect to the number of approximation coefficients n. Plot these errors.

4 Calculation of Fourier coefficients

We define $f(t)$ as

$$f(t) = e^{\frac{-t^2}{10}} \left(\cos(2t) + 2\sin(4t) + 0.4\cos(2t)\cos(40t)\right) \ . \tag{6}$$

1. Plot this function in matlab.
2. For which values of n do you expect Fourier coefficients that are clearly different from 0 (e.g., say, larger than $|0.1|$)?
3. Implement a function in matlab that computes the n first Fourier coefficients.
4. Plot the resulting values for a_n and b_n. Check whether your assumption was correct.
5. Vary the number of approximation coefficients and compare the approximation with the exact function.
6. Introduce a threshold. Use for the approximation only coefficients that are larger (in absolute) than the threshold.

Useful functions in matlab: `sin, cos, sum, mean, max, abs`