# Image Approximation with Fourier Wavelets*

November 18, 2019

## 1 Image Loading and Displaying

To load and visualize images, the toolboxes provide direct commands.

```
n = 256; % Size of image
M = load_image('lena',n);

figure
imageplot(M)
```

An image is just a matrix $f \in \mathbb{R}^N$ of $N = N_0 \times N_0$ pixels.

To implement subplots within the `imageplot`

```
figure
imageplot(M(1:50,1:50),'Zoom',1,2,1)
imageplot(-M,'Reversed contrast',1,2,2)
```

### 1.1 Blurring

Blurring is achieved by computing a convolution $(f * g)$ with a kernel $g$.

**Definition 1** *A **convolution** is a mathematical operation on two functions $(f, g)$ that produces a third function expressing how the shape of one is modified by the other. The term convolution refers to both the result function and to the process of computing it. It is defined as the integral of the product of the two functions after one is reversed and shifted.The convolution of $f$ and $g$ is written $f * g$, denoting the operator with the symbol $*$. It is a particular kind of integral transform:*

$$(f * g) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

```
k = 9; % size of the kernel (the larger, the more blurred)
g = ones(k,k);
g = g/sum(g(:)); % normalize

fg = perform_convolution(f,g);
```

---

```
% Blurred image
figure
imageplot(fh,'Blurred',1,2,1)
imageplot(f,'Original',1,2,2)
```

**Blurred**                    **Original**



Figure 1: Blurring by use of Convolution

## 1.2 Fourier Transform

To measure error between an image $f$ and its approximation $f_M$, we use the signal-to-noise-ratio (SNR) measure

**Definition 2** *The signal to noise ratio (SNR) is defined as*

$$SNR(f, f_M) = -20 \log_{10} \left( \frac{\|f - f_M\|}{\|f\|} \right)$$

*which is a quantity expressed in decibels (dB). The higher the SNR, the better the quality.*

**Definition 3** *A basis B of a vector space V over a field F (such as the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$) is a linearly independent subset of V that spans V. This means that a subset B of V is a basis if it satisfies the two following conditions:*

- **Linear independence property:** *for every finite subset $\{b_1, \ldots, b_n\}$ of B and every $a_1, \ldots, a_n$ in F, if $a_1 b_1 + \cdots + a_n b_n = 0$, then necessarily $a_1 = \cdots = a_n = 0$;*

- **Spanning property:** *for every (vector) $v \in V$, it is possible to choose $v_1, \ldots, v_n$ in F and $b_1, \ldots, b_n$ in B such that $v = v_1 b_1 + \cdots + v_n b_n$.*

**Definition 4** *Any n orthogonal vectors which are of unit length*

$$\langle u_i, u_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

*form an orthonormal basis of $\mathbb{R}^n$*

**Definition 5** *The Fourier orthonormal basis is defined as*

$$\psi_m(k) = \frac{1}{\sqrt{N}} e^{\frac{2i\pi}{N_0} \langle m,k \rangle}$$

*where $0 \leq k_1, k_2 < N_0$ are position indexes, and $0 \leq m_1, m_2 < N_0$ are frequency indexes.*

The Fourier transform $\hat{f}$ is the projection of the image on this Fourier basis.

$$\hat{f}(m) = \langle f, \psi_m \rangle.$$

The Fourier transform is computed in $O(Nlog(N))$ operation using the FFT algorithm (Fast Fourier Transform). Note the normalization by $N = \sqrt{N_0}$ to make the transform orthonormal.

The following snippet of code shows the Fourier transform of the original image, conservation of energy, and the log of the Fourier magnitude $\left( \log \left( \left\| \hat{f}(m) \right\| + \varepsilon \right) \right)$. The function `fftshift` shifts the zero-frequency component to the center of the array.

```
% Normalized Fast Fourier Transform
F = fft2(f)/n0;

% Check conservation of the image
fprintf('Energy of image: %5.5f \n',norm(f(:)))
fprintf('Energy of Fourier: %5.5f \n',norm(F(:)))

% Compute the log¡ of the Fourier magnitude for some small epsilon
epsi = 1e-2;

% Shift the zero frequency component to the center of the array
L = fftshift(log(abs(F)+1e-1));

% Display
clf;
imageplot(L,'Log(Fourier Transform)')
```

## 1.3   Linear Fourier Approximation

An approximation is obtained by retaining a certain set of index $I_M$

$$f_M = \sum_{m \in I_M} \langle f, \psi_m \rangle \psi_m.$$

A linear approximation is obtained by retaining a **fixed** set $I_M$ of $M = |I_M|$ coefficients. The important point is that $I_M$ does not depend on the image $f$ to be approximated.

For the Fourier transform, a low pass linear approximation is obtained by keeping only the frequencies within a square.

$$I_M = \{m = (m_1, m_2), -q/2 \leq m_1, m_2 < q/2\}$$
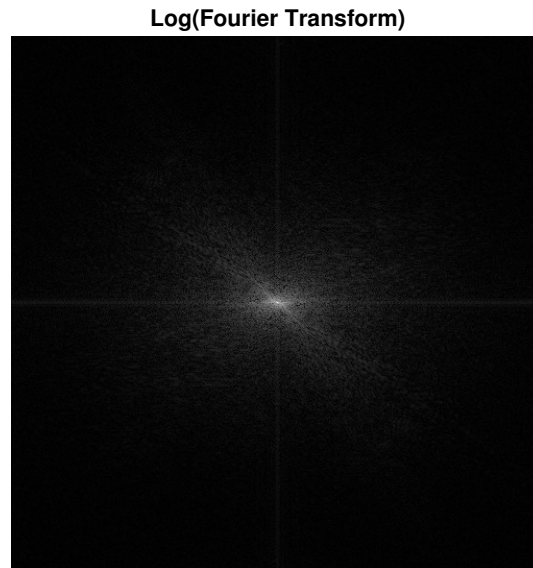
**Log(Fourier Transform)**

Figure 2: Log of Fourier Magnitude

where $q = \sqrt{M}$.

This can be achieved by computing the Fourier transform, setting to zero the $N - M$ coefficients outside the square $I_M$ and then inverting the Fourier transform.

**Example 1** *Perform the linear Fourier approximation with M coefficients. Store the result in the variable* fM *and display.*

```
% Number of kept coefficients
M = n0^2/64;
% Bound of interval
q = sqrt(M);

% Compute Centered Fourier transform
F = fft2(f);

% Linear approximation pre-allocation
F1 = zeros(n0,n0);

% Choose a square in the middle of the image
sel = (n0/2-q/2:n0/2+q/2) + 1;

% Take the points of the square to the linear approx
F1(sel,sel) = F(sel,sel);

% Invert the Fourier and keep real terms
fM = real(ifft2(F1));

% Plot
figure
```

```
imageplot(f,'Original',1,2,1)
imageplot(F1,['Linear Fourier Approximation with ',num2str(snr(f,fM),4)])
```

## 1.4 Non-Linear Fourier Transform

A non-linear approximation is obtained by taking the $M$ largest coefficients. This is equivalently computed using a threshold for the coefficients.

$$I_M = \{m, |\langle f, \psi_m \rangle| > T\}$$

Figures 3a and 3b show the linear and non-linear approximations for a different set of parameter values.
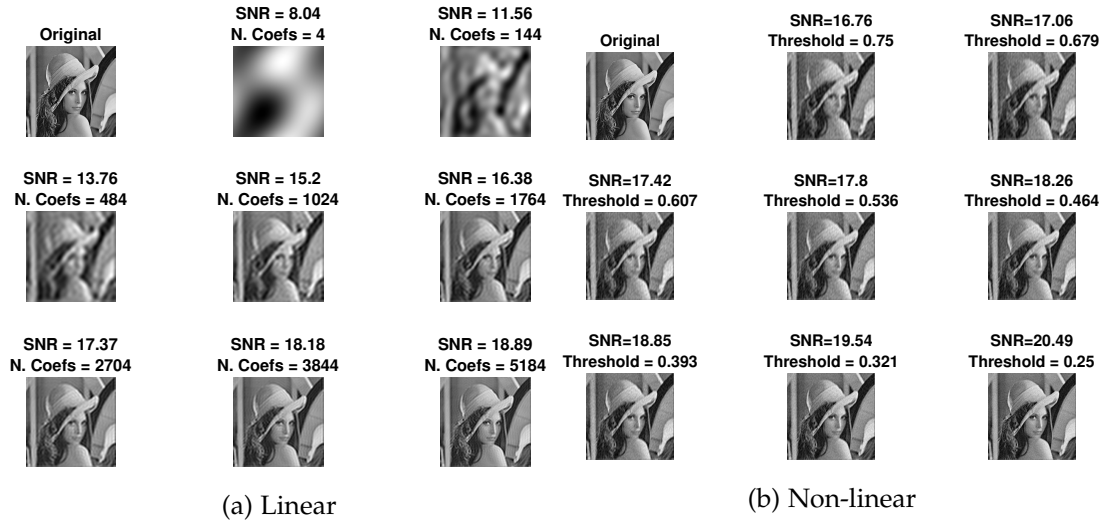


(a) Linear          (b) Non-linear

Figure 3: Fourier Approximations for Different Parameter Values