MLRF Lecture 02

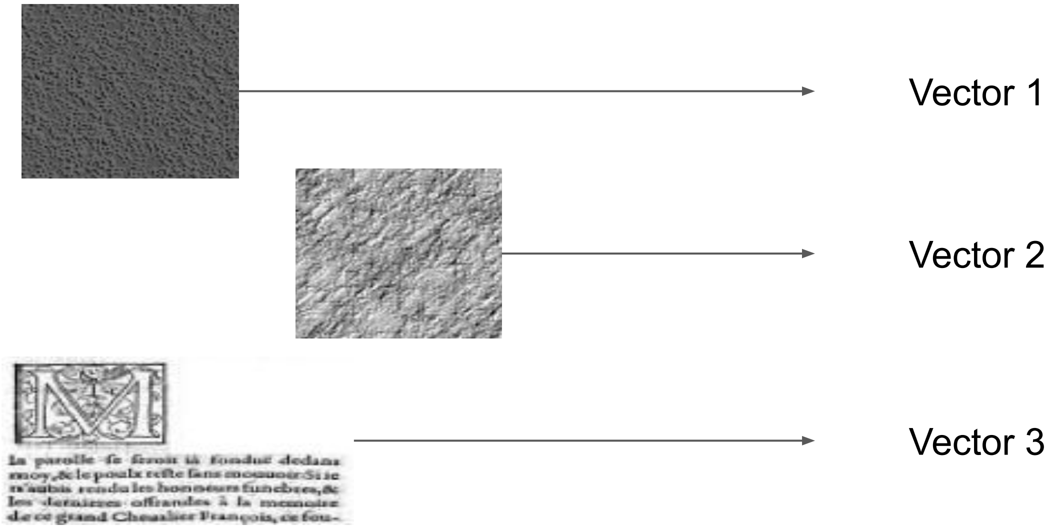
J. Chazalon, LRDE/EPITA, 2019

Texture descriptors

Lecture 02 part 04

What are they useful for?

To describe (then match, group, classify…) relatively large and regular images.



A tentative taxonomy (of a selection of approaches)

Statistical

- GLCM (Grey Level Co-occurrence Matrix)
- Fractal dimension

Frequency-based

- Fourier transforms
- Difference-of-Gaussian filter
- Gabor filters
- Wavelets

Model-based

- Markov Random Fields
- Convolutional Neural Networks

Statistical approaches

GLCM (Grey Level Co-occurrence Matrix)

From an image patch of grayscale image (usually 16 levels), compute the matrix

$$C_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{n}\sum_{y=1}^{m}\begin{cases}1, & \text{if } I(x,y)=i \text{ and } I(x+\Delta x, y+\Delta y)=j\\ 0, & \text{otherwise}\end{cases}$$

Where, for each cell (i,j) we add 1 when the pixel I(x,y) has value i and the pixel I(x+Δx, y+Δy) has value j.

Example, for Δx = Δy = 1:

Image

$$\begin{matrix} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{matrix}$$

GLCM

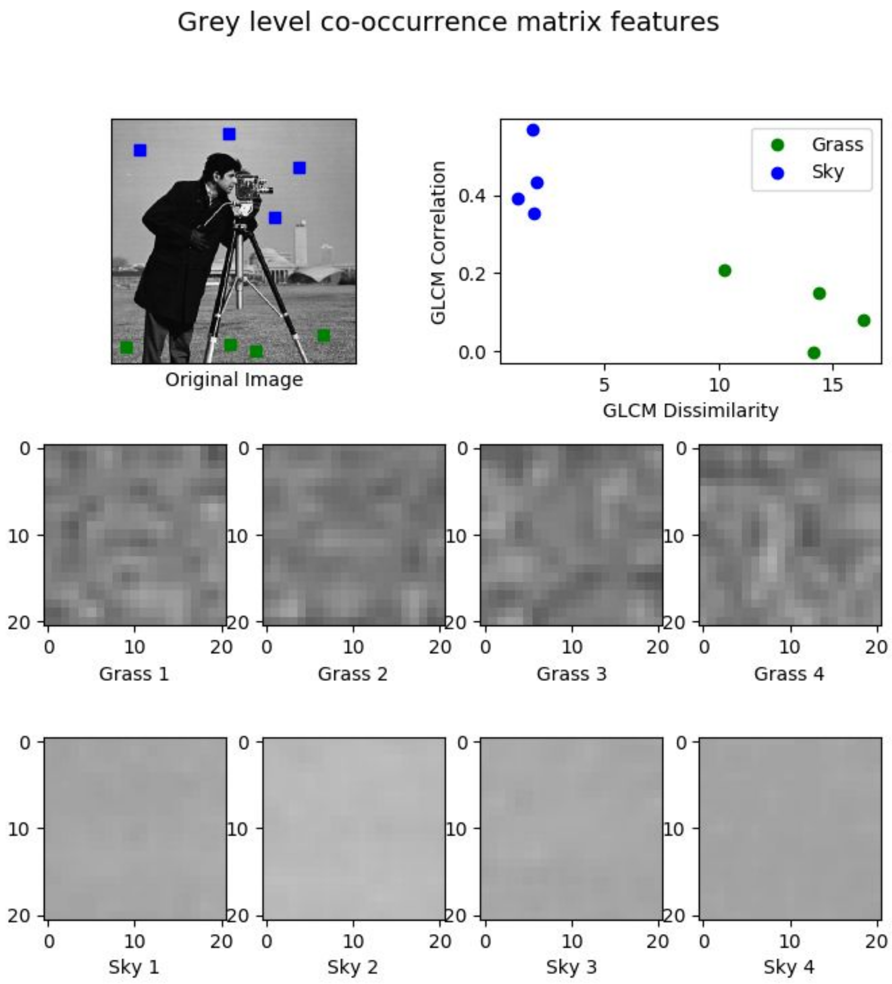
$$C = \frac{1}{16}\begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$

Robert M Haralick; K Shanmugam; Its'hak Dinstein (1973). "Textural Features for Image Classification". *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-3 (6): 610–621.

Using GLCM

We usually compute statistics on such matrix like:

- Contrast $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$
- Dissimilarity $\sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$
- Homogeneity $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- ASM $\sum_{i,j=0}^{levels-1} P_{i,j}^2$
- Energy $\sqrt{ASM}$
- Correlation $\sum_{i,j=0}^{levels-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right]$



https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_glcm.html

Fractal dimension

“How a smaller version of myself is equal to myself?”

Method of Range:

- Take 2 windows of size l_1=9 and l_2=5, centered on the same pixel.
- Compute the brightness range r_1 for window 1 and r_1 for window 2.
- Estimate the fractal dimension as $D = \frac{r_1 - r_2}{\ln l_1 - \ln l_2}$

D: ratio between the difference of the ranges in each window and the proportion of the length of each window, in log scale.

Input image

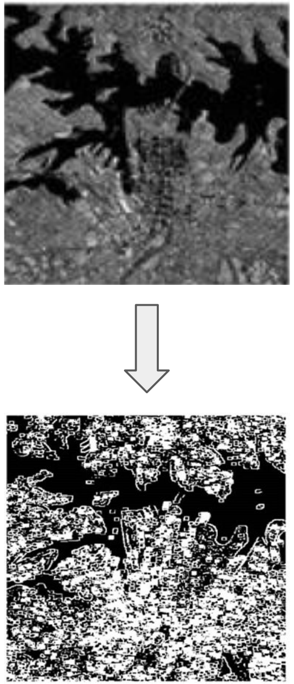
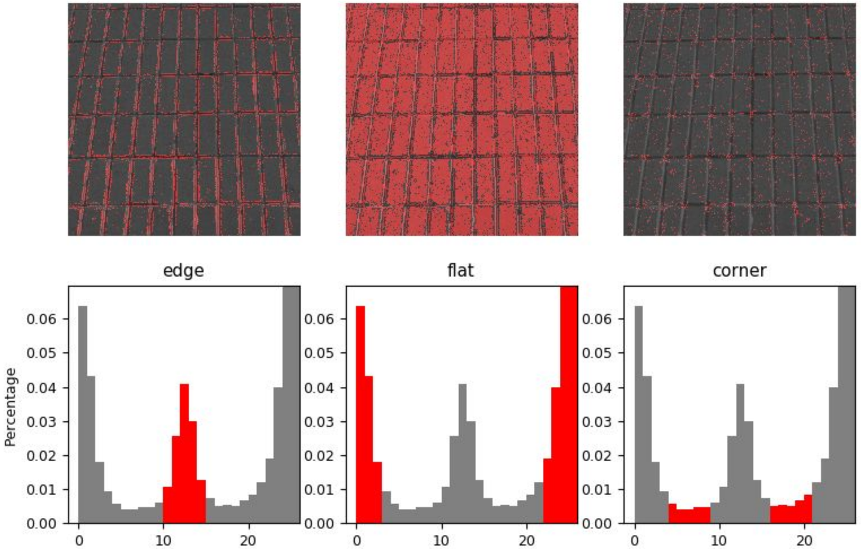


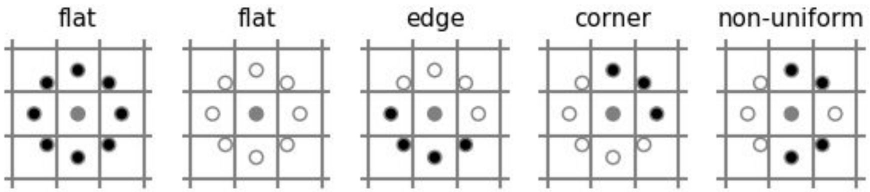
Image showing places where D is above a global threshold.

Marrón. https://doi.org/10.4236/jsip.2018.93014

Local Binary Patterns

LBP looks at points surrounding a central point and tests whether the surrounding points are greater than or less than the central point (i.e. gives a binary result).





T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.
https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_local_binary_pattern.html#sphx-glr-auto-examples-features-detection-plot-local-binary-pattern-py

Frequency-based approaches

Fourier transform

For each possible frequency, sum pixel contributions from original image.

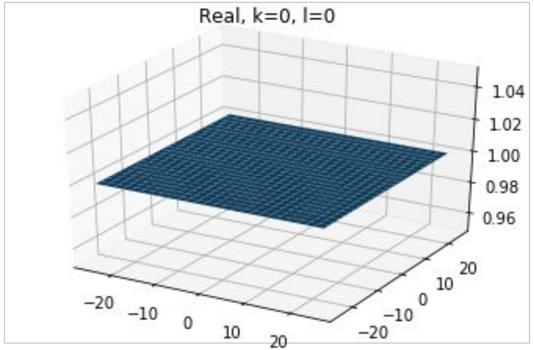
$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)\, e^{-\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)}$$

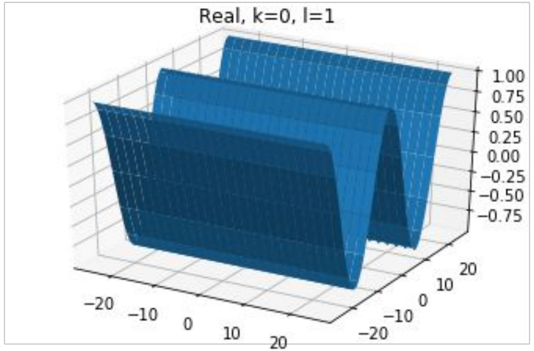
The exponent term can be viewed as the filter for the target frequency over the spatial image.

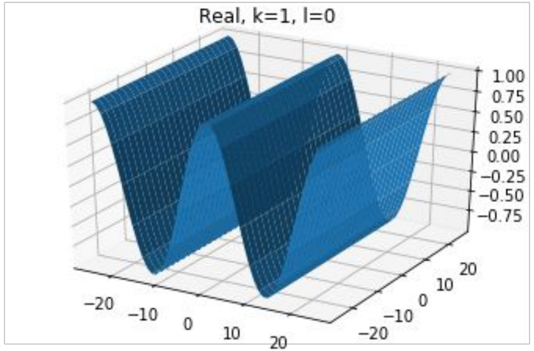
Next: values of the exponent term (only) for various k, l.

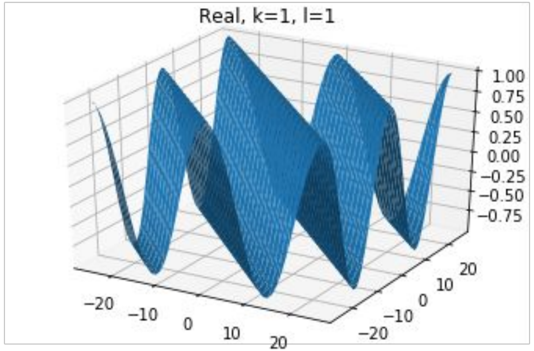
https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm

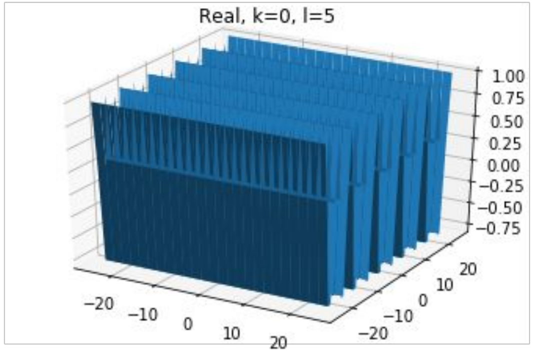
Values of the exponent term (only) for various k, l.

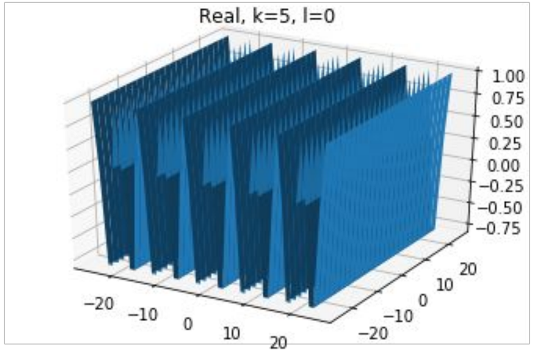


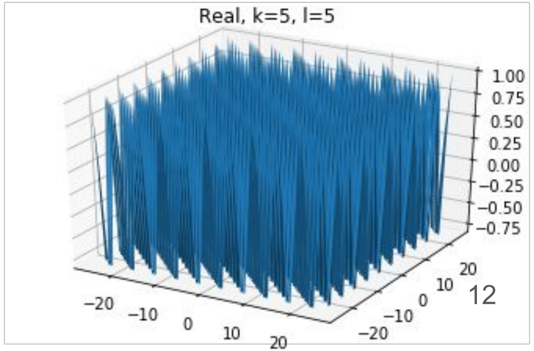




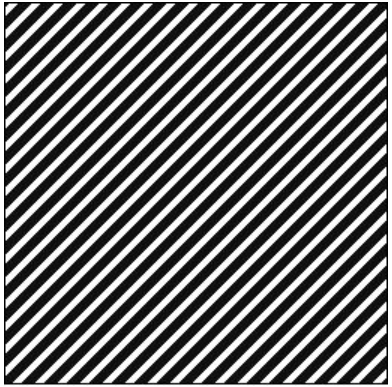


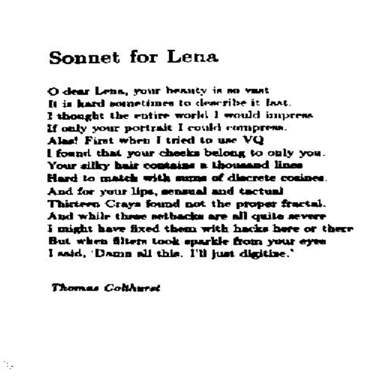






Fourier transform

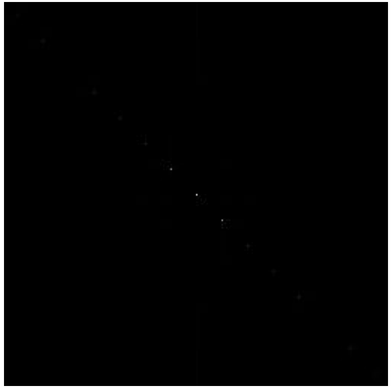


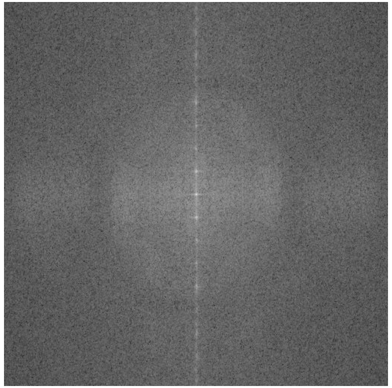


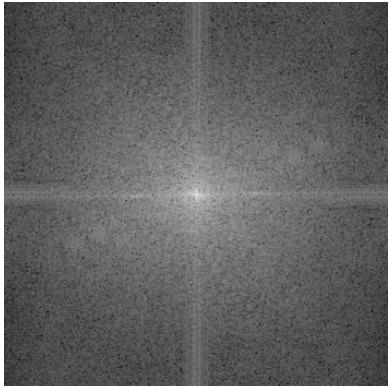
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst







https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm

Difference-of-Gaussian filter

Take a image.

Blur it.

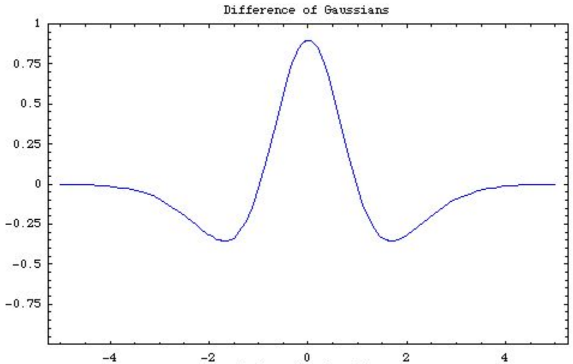
Take the difference.

Difference-of-Gaussian filter

It is a band-pass filter.

$$\Gamma_{\sigma,K\sigma}(x,y) = I * \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} - I * \frac{1}{2\pi K^2 \sigma^2} e^{-(x^2+y^2)/(2K^2\sigma^2)}$$

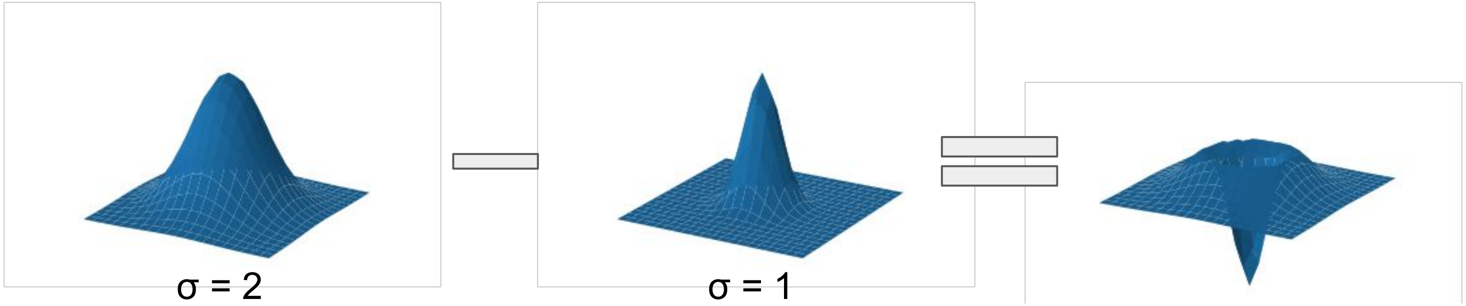
$$\Gamma_{\sigma,K\sigma}(x,y) = I * \left(\frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} - \frac{1}{2\pi K^2 \sigma^2} e^{-(x^2+y^2)/(2K^2\sigma^2)}\right)$$



Difference-of-Gaussian filter

Intuition

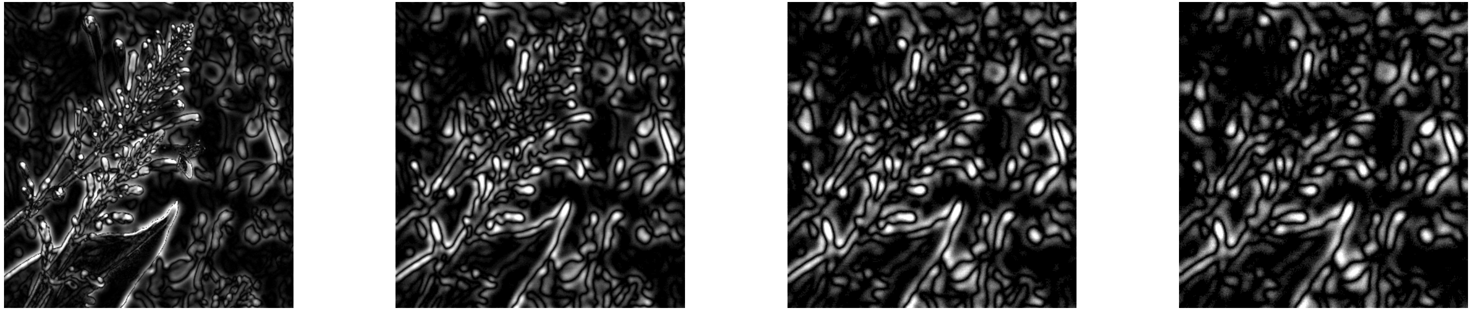
- Gaussian (g) is a low pass filter
- Strongly reduce components with frequency $f < \sigma$
- $(g*I)$ low frequency components
- $I - (g*I)$ high frequency components
- $g(\sigma1)*I - g(\sigma2)*I \Leftarrow$ Components in between these frequencies
- $g(\sigma1)*I - g(\sigma2)*I = [g(\sigma1) - g(\sigma2)]*I$



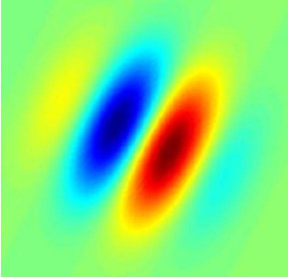
Difference-of-Gaussian filter

Many applications.

Indicates the “size” of the “stable” region around a pixel at a given freq. band.



Gabor filters



Gabor filters allow to select both a frequency band and an orientation.

$$G_c[i,j] = Be^{-\frac{(i^2+j^2)}{2\sigma^2}} \cos(2\pi f(i\cos\theta + j\sin\theta))$$

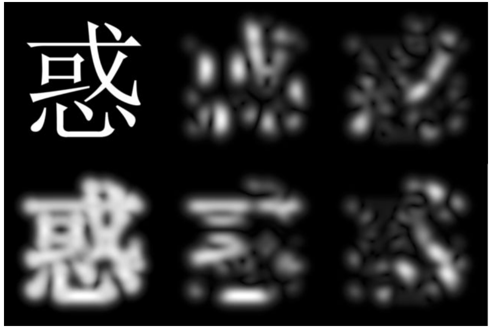
$$G_s[i,j] = Ce^{-\frac{(i^2+j^2)}{2\sigma^2}} \sin(2\pi f(i\cos\theta + j\sin\theta))$$

B and C: scaling factors

f : frequency selection

θ: angle selection

σ: size of the image region being analysed



Wavelets

Much like Gabor filters, with potentially more complex patterns.

It turns the image into a grid of coefficients based on an orthogonal basis of small finite waves, or “wavelets”.

(Used in JPEG-2000.)

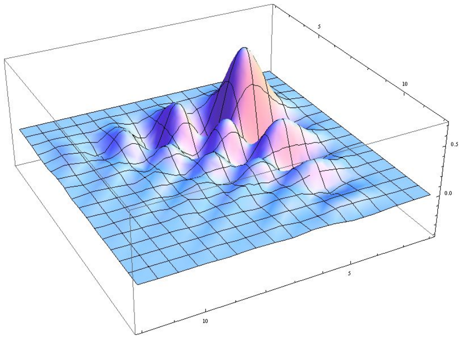


Image: **JonMcLoone** at **English Wikipedia**

Learning-based approaches

Markov Random Fields
Convolutional Neural Networks

Learn to produce a high response for some texture samples / patches.

The filter bank is not orthogonal in general, but rather overcomplete, ie the original signal can be recovered using a small subset of filters.

Highly tunable, but a good random sampling over the possible patch patterns can gives good results too.

(More on than next year, hopefully.)