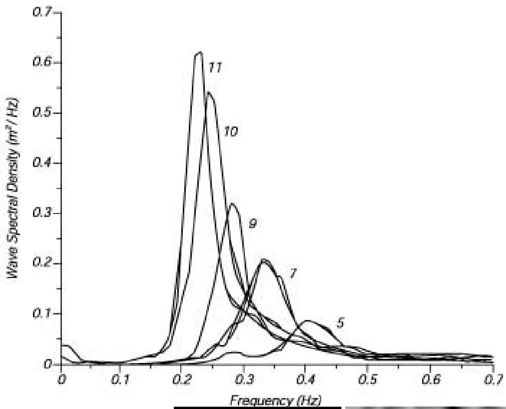
Fourierereries

1. Prelude: some Fourier strangenesses

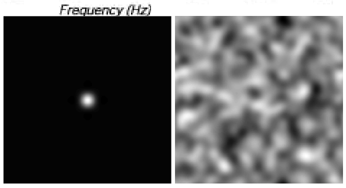
1.1. Are images/textures“features” encoded into modulus or phases ?



This is indeed a tricky question.

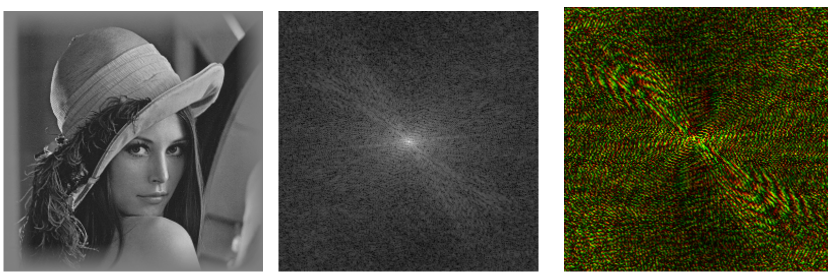
- For most physically-based spectrum, only the DSP (i.e. modulus²) is looked at. “Take random phases”, they say. e.g. http://www.wikiwaves.org/Ocean-Wave_Spectra

$$S_j(\omega) = \frac{\alpha g^2}{\omega^5} \exp\left[-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right]\gamma^r \qquad r = \exp\left[-\frac{(\omega-\omega_p)^2}{2\sigma^2\omega_p^2}\right]$$

  - Same for “wavy like” textures & patterns (generally thought).
  - Still, phases are compulsory:
    same DSP, zero phases vs random phases →
    
    (BTW, what does “random” exactly means here ?)
  - And big issues with interpolation: phases don’t meaningfully interpolate.

  → phases annoying, but not really encoding a lot ?

- Pretty different situation on figurative images like Lena:
  image f   modulus $\log(1+|\hat{f}|)$ phase



Keeping <u>only the phases</u> and replacing the modulus by *(left to right) 1, random, or some arbitrary function like* $|\vec{\omega}|^{-1.6}$ you do get back most of Lena, while *(extreme right) keeping only the modulus and using random phases* gives nothing meaningful:

Note that the third reconstruction tells that most of the shading is *also* encoded in phases !
→ almost nothing in the modulus and all in phases ?

- BUT check this fancy experiment :-)

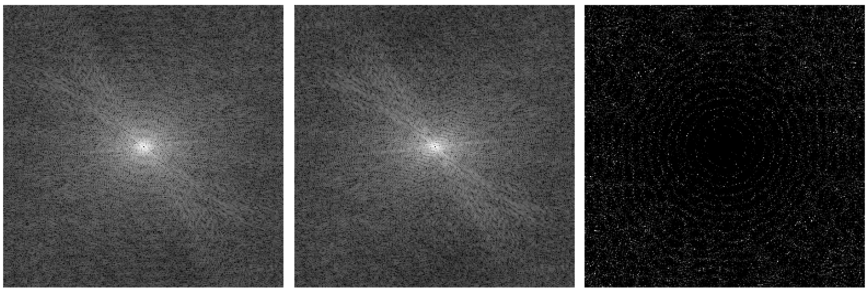
  suppose you only have the spectrum modulus of an image you want to recover, and the phases of a different-but-not-too-much reference image.
  Coupling the accurate modulus with the reference phases (where added objects are missing) does gives you back the missing objects ! (plus some noise).

  modulus of   + *phases of*   *reconstruct as:*

  → finally, the disk is totally encoded in the modulus ?

  My interpretation (temptatively):
  adding an object induces interference beats that also showing in the modulus.

  More in http://www.ysbl.york.ac.uk/~cowtan/fourier/coeff.html

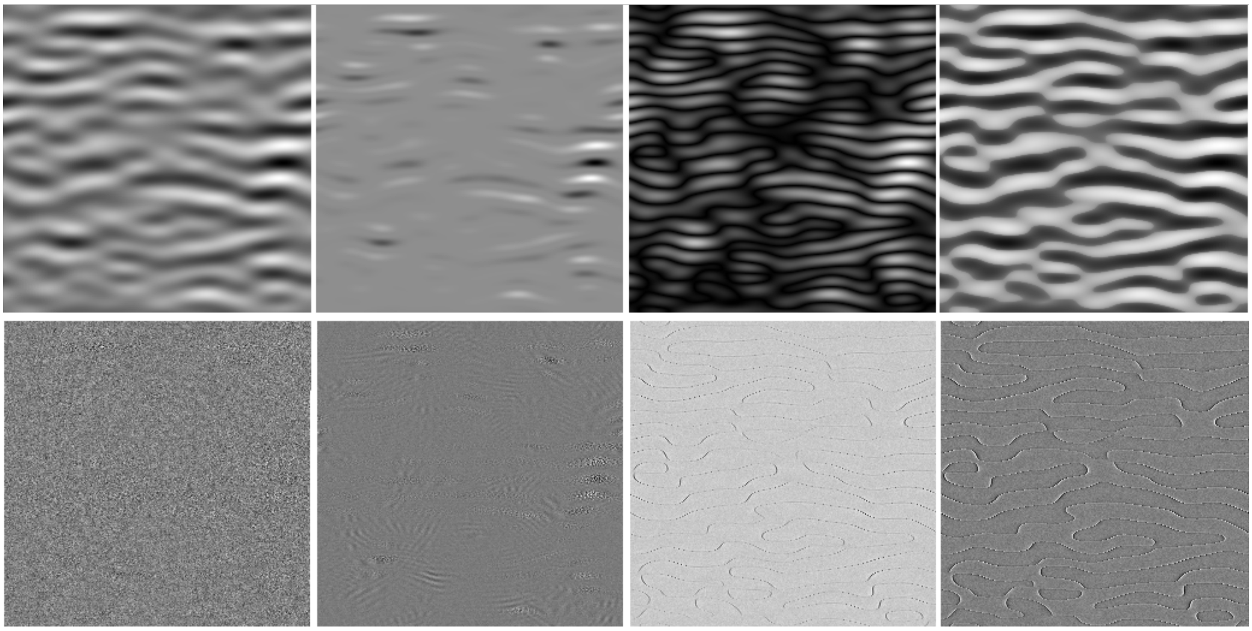
- Stranger and stranger:

  in"*The Importance of Phase in Signals*", AV Oppenheim - 1981, p537b-538, they propose an algorithm able to reconstruct the full shaded image only starting from the phases
  http://ieeexplore.ieee.org/ielx5/5/31301/01456290.pdf?tp=&arnumber=1456290&isnumber=31301
  (they refer to "Gerchberg-Saxton algorithm" , proved to always converge when iteratively reinjecting phases or modulus).

How phases-features appear along processing

Starting with a simple Gabor noise (spectrum modulus = bi-lobe, phases = uniform random), let see for different image transform (top row) if phases encode something, by showing the image resulting into keeping only the phase and resetting the modulus to 1.

f=simple Gabor bi-lobe | f^3 | $|f|$ | $sgn(f)*\sqrt{|f|}$

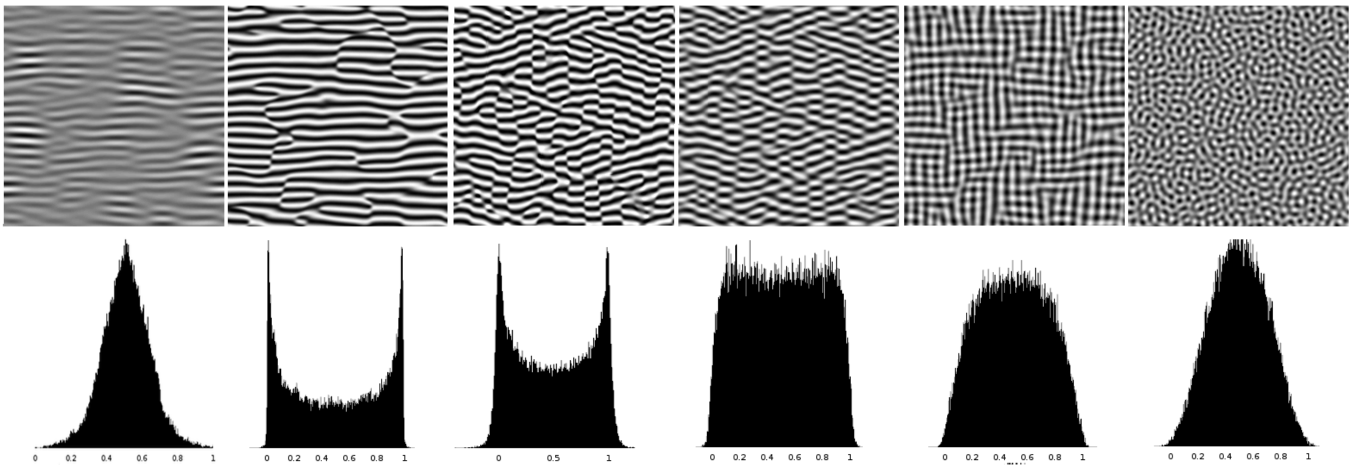


As we see, the initial “wavy” features are not at all phase features.
But any non-linear transform (visually enforcing features) encode feature-alignment conspiracy into phases.

1.2. Key appearance attributes not really talking to each-other

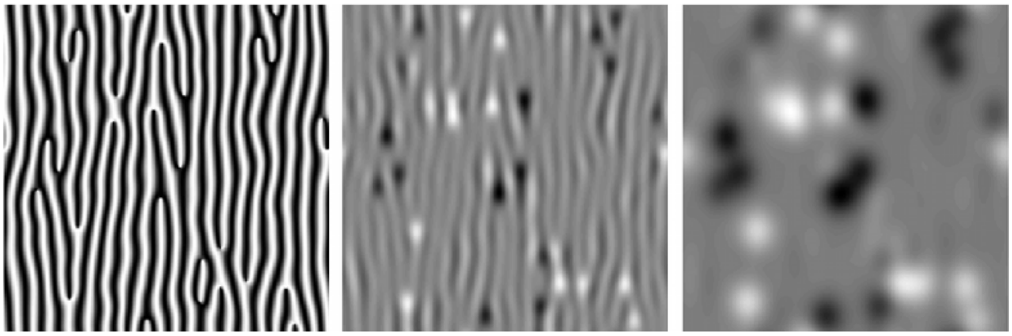
- DSP (profile of “waviness”)
- phases/contours (features boundaries = correlation / conspiracy between wave fronts)
- histogram profile & min-max bounding vs std-dev
- slopes min-max bounding (sharpness) vs std-dev(derivative)

1.2.1. Histograms



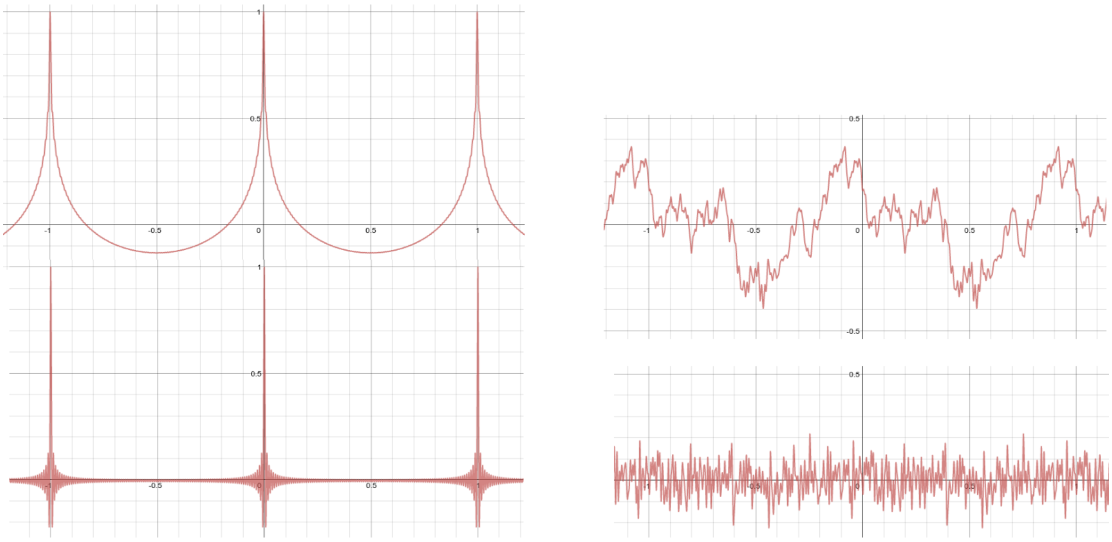
- std-dev is point-less (alone) for min-max… (alas we often want to bound min-max).

- very poor intuition/perception of histogram (intermediate values, influence of shape).



1.2.2. Min-max of image or of slopes :

texture module = $\frac{1}{\omega}$. *Top: image. Bottom: slope. Left: linear phases. Right: random phases.*



→
statistical sharpness (in normalized signal) relates to HF in spectrum.
but sharp feature is feature, and relates to phase conspiracy (much higher amplitude).

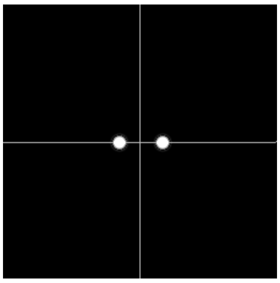
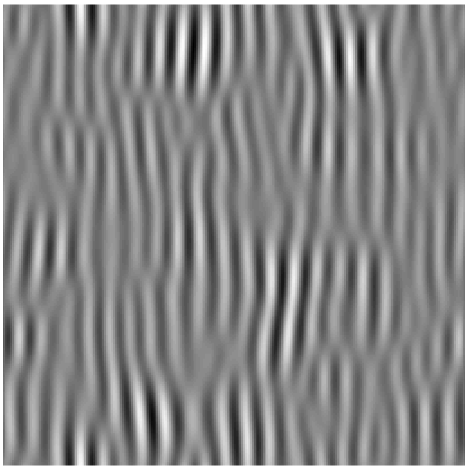
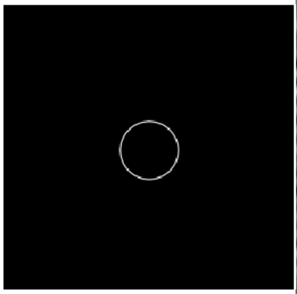
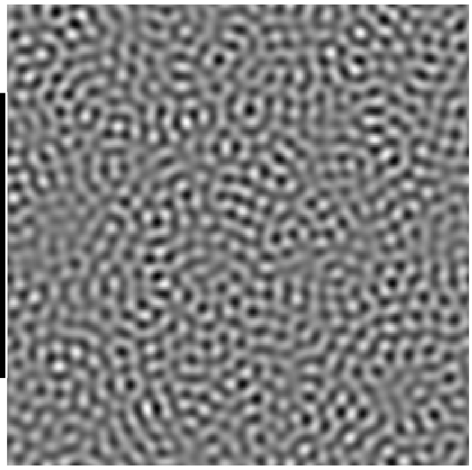
1.3. What is a texture ?

- patterns, features
- vs "wavyness", stochastics (generally thought)
- stationarity of descriptors beyond a "textural scale"
- vs gradient texture (stationarity of something else than pixels value :-)).

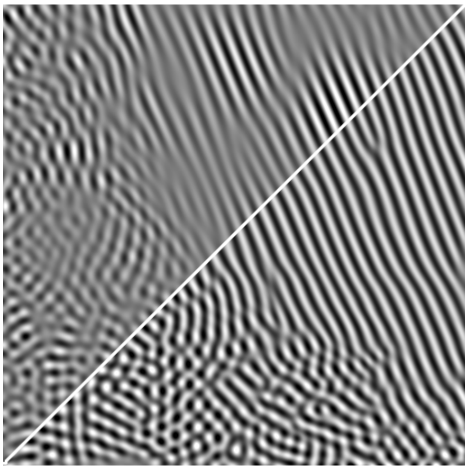
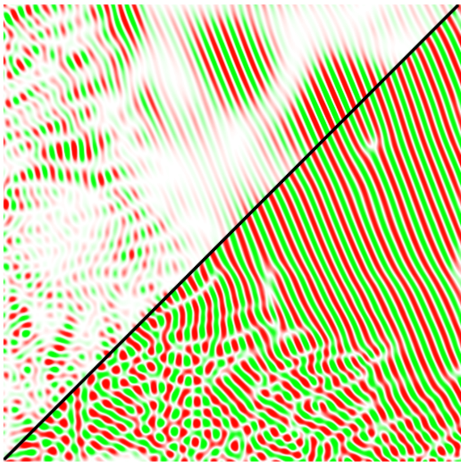
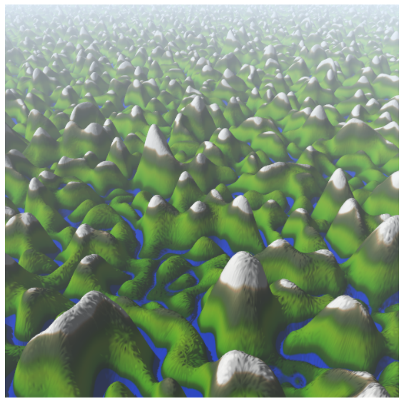
2. Revisiting Fourier texture synthesis

2.1. Motivations

Fourier synthesis (e.g., Gabor noise) used for “wavy” or “stochastics” textures.
issue: unwanted contrast oscillations.

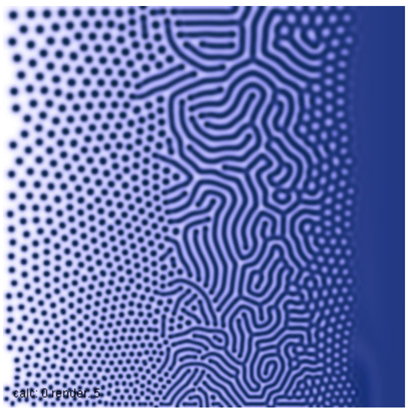
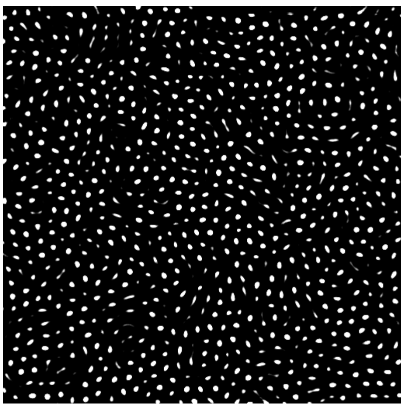
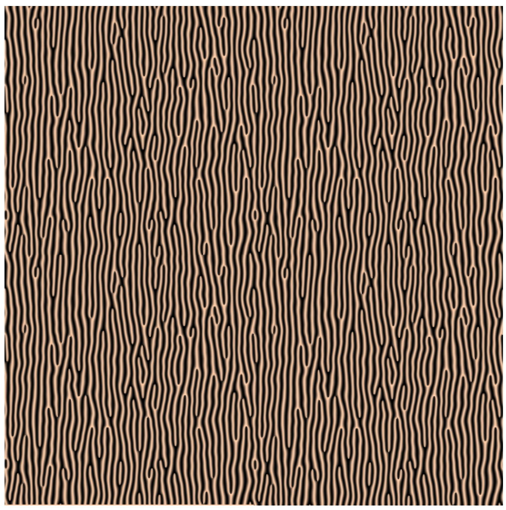





Worse with color maps:
(what we have, what we want)

What do we want ?

- sometime ok (water waves), sometime not (desert, ripples, zebras strips, cheetah spots):
- LF stationarity (beyond texture scale)
- pseudo-pattern with saturated min/max
- Note that Perlin conform to this, and Fourier texture not (while both supposed “stochastics”)
- “I put only strips frequency, I want only their strips and nothing else” (e.g. reaction-diffusion look)

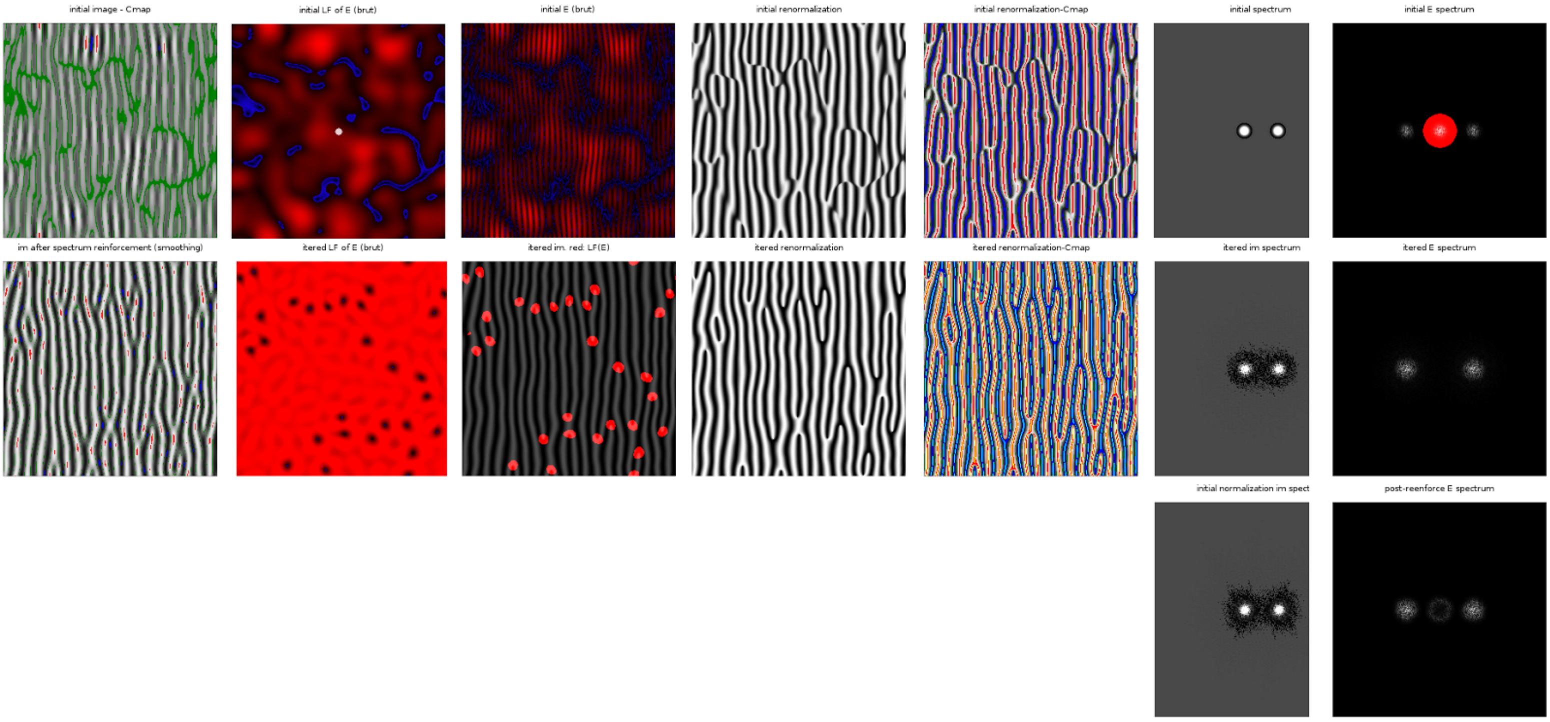




2.2. Where these LF oscillations come from ?

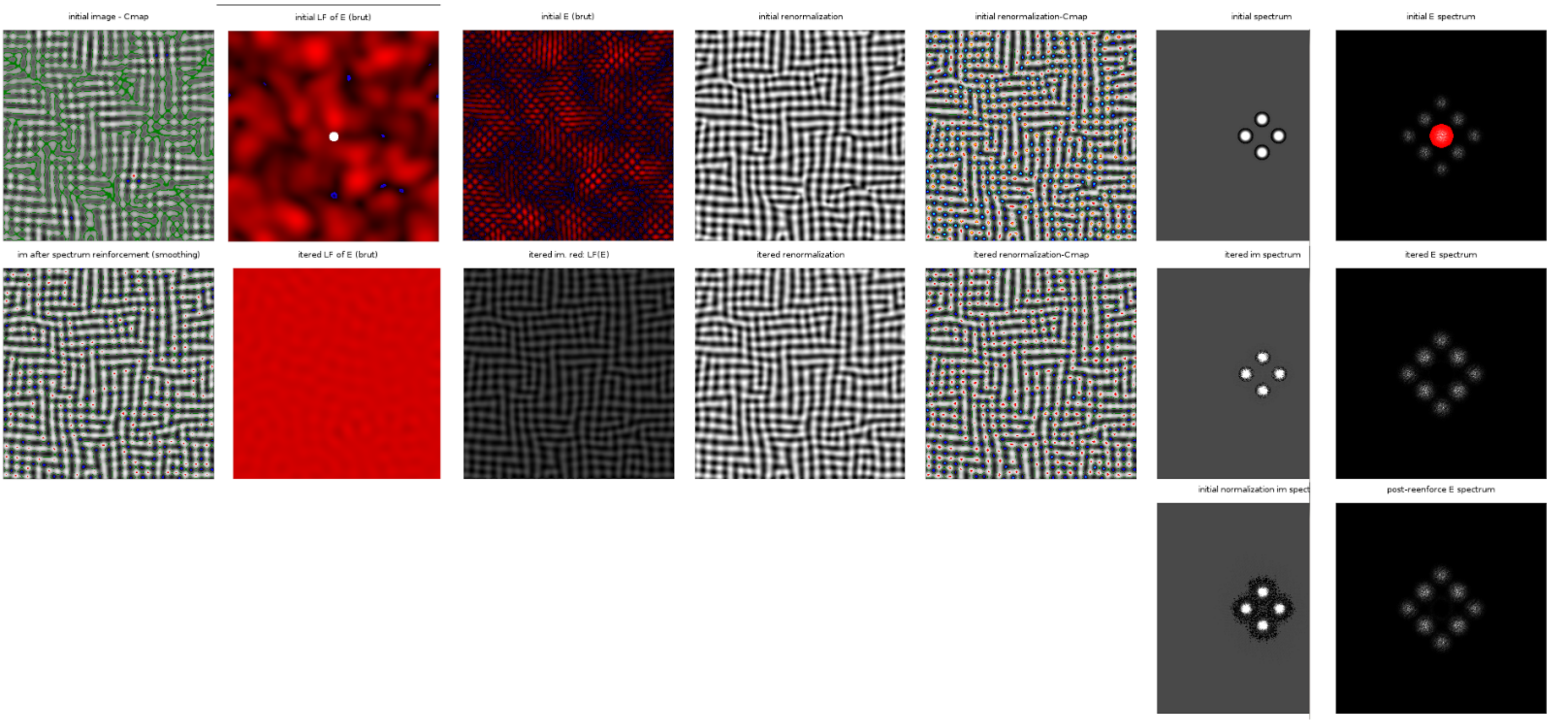
We did not put any LF in the spectrum, after all !

2.2.1. Lab

bi-lobe Gabor noise:



quadri-lobe Gabor noise:



→ **contrast := sqrt(FFT^{-1}(LF(variance spectrum))).** LF=whatever you want: texture scale / strip scale

→ **idea: normalize signal by the contrast.**

2.2.2. Welcome to the magic world of Fourier(f²) , aka ***Variance Spectrum***

(which is <u>not</u> the DSP, |Fourier|²(f))

completed recap:

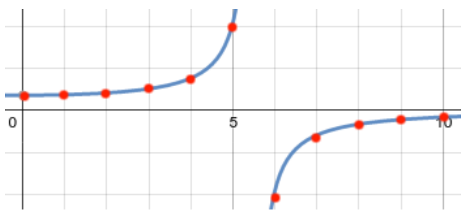
$op(f(x))(\omega)$ NB: signal f, $\bar{f}=0$, $F()$:=fourier	$op(\omega=0)$	$\int_\omega op$
$F(f)$ signal spectrum	$F(f)(0) = \bar{f}$	$\int_\omega F(f) = f(0)$
$\lvert F(f)\rvert^2$ power/energy spectrum (DSP)	$\lvert F(f)\rvert^2(0) = (\bar{f})^2$	$\int_\omega \lvert F(f)\rvert^2 = \int_x f^2 =$ E , variance
$F(f^2)$ spectrum of variance	$\lvert F(f^2)\rvert(0) =$ E , variance	$\int_\omega \lvert F(f^2)\rvert = f^2(0)$
$\lvert F(f^2)\rvert^2$ DSP of variance	$\lvert F(f^2)\rvert^2(0) =$ variance²	$\int_\omega \lvert F(f)\rvert^2 = \int_x f^4 =$ variance$_{LF(x)}$ of variance

- ➢ spectrum of variance decompose spatial fluctuations of windowed variance or radius $2\pi/\omega$
- ➢ DSP (variance of spectrum) decompose signal variance into contributions of signal spectral band.

  (make sense since linear despite the square: $\int (f+g)^2 = \int f^2 + \int g^2$ when $f \perp g$)

2.2.3. Image spectrum vs image sub-window spectrum

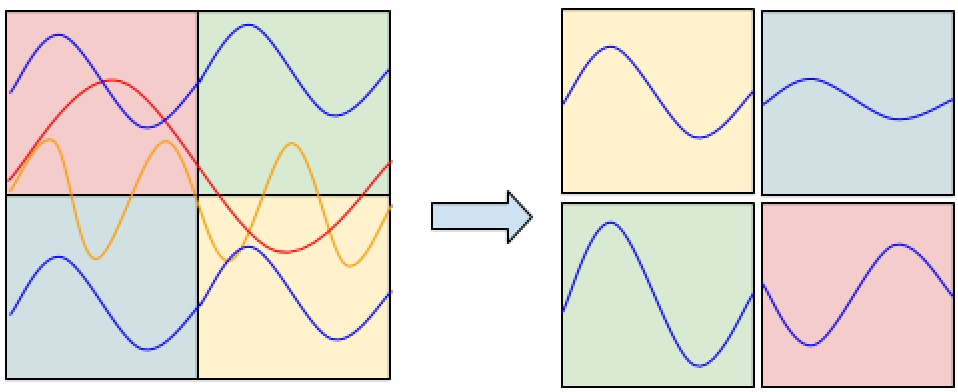
- Each sub-window spectrum only have $\omega_k^{sw} = \omega_{2k} = 2k.\omega_{window}$ ($\omega_{sub-window} = 2\omega_{window}$)
- *All* freqs ω_{1+2k} cause lower frequencies, not only ω_1 . $\omega_{>1}$ corresponds to LF variation of ω_k^{sw} spectrum through sub-windows.
- Beats between ω_{2k-l} and $\omega_{2k+l} = 2\cos(2\pi 2k\omega_{window}x)\cos(2\pi l\omega_{window}x)$: they alter ω_k^{sw} i.e., $FFT_{sw}(\omega_k^{sw}) = FFT(w_{2k}) +$ modulations per sub-window.
- Stated another way: ω_{2k+1} contributes to all ω_j^{sw} : $\int_0^1 \sin(2\pi(k+\frac{1}{2})x)\cos(2\pi jx) = \frac{1}{2\pi}(\frac{1}{k+j+\frac{1}{2}} + \frac{1}{k-j+\frac{1}{2}})$



Contribs at frequencies j caused by freq k= $5+\frac{1}{2}$:

|contrib| is max for j=k ($\rightarrow \frac{1}{2\pi}(2+\frac{1}{2k+\frac{1}{2}})$) and j=k+1 ($\rightarrow \frac{1}{2\pi}(-2+\frac{1}{2k+\frac{3}{2}})$)

Even the sub-window DC is modified: j=0 $\rightarrow \frac{1}{2\pi}(\frac{2}{k+\frac{1}{2}})$

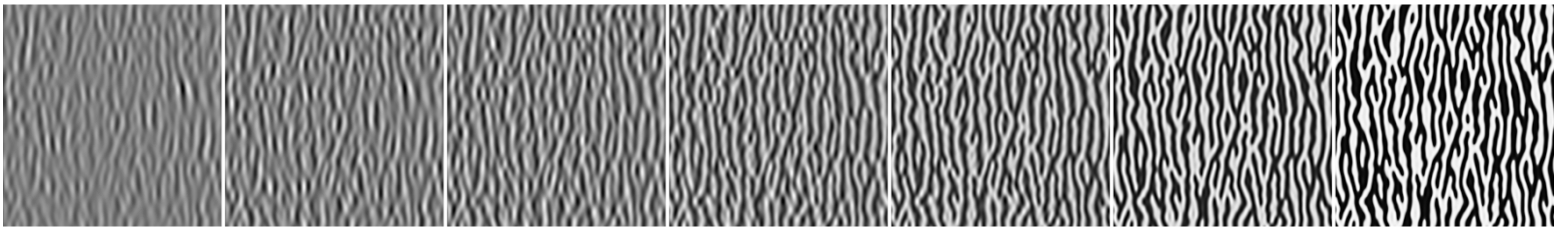


2.3. Operations in variance spectrum

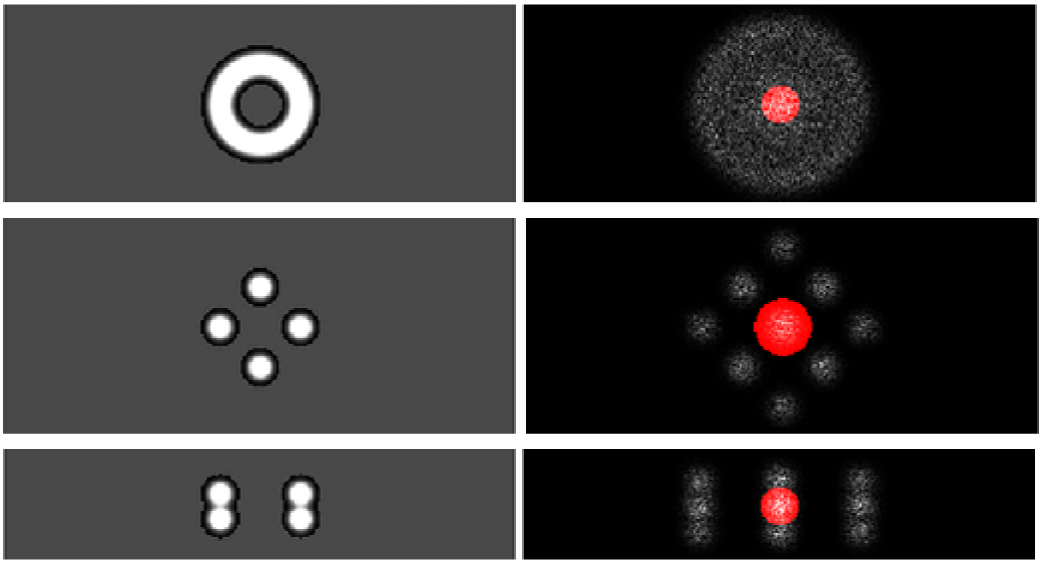
- FFT(f²)
- filter variance spectrum
- contrast := sqrt(FFT⁻¹)
- normalize f by contrast

2.3.1. LF filter size

from 0 to texture scale to strip scale to pixel scale.



Sometime objective choice, sometime less automatic

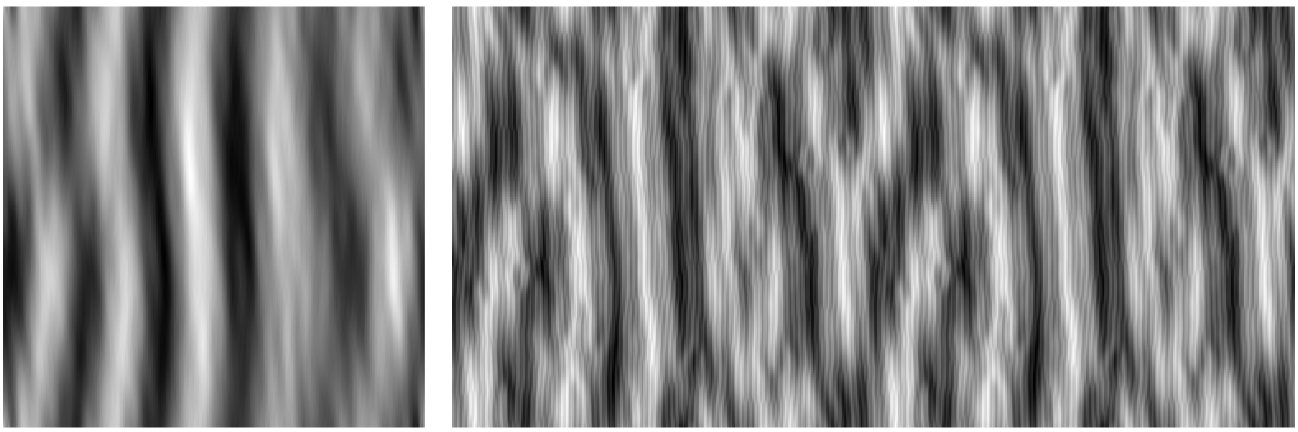


2.3.2. Separating components

For different wavelengths, it seems to work well :

left: without normalization

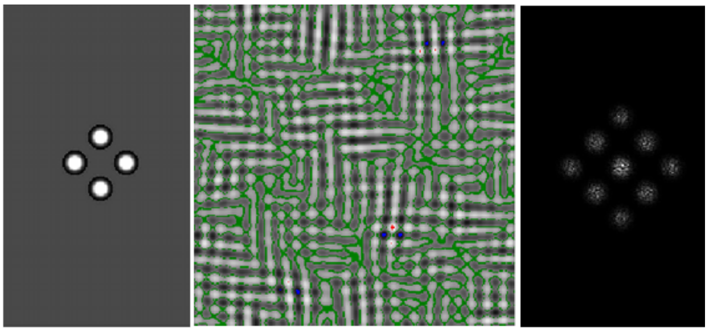
right: with



2.3.3. Filtering-out other lobes

E.g., **on quadri-lobe**, we can selectively suppress only the LF (variance fluctuations), or also the beat between the 2 directions, or keep only the beat and suppress the main signal in the 2 directions.
This are the 3 cases illustrated above (below the source Gabor image).

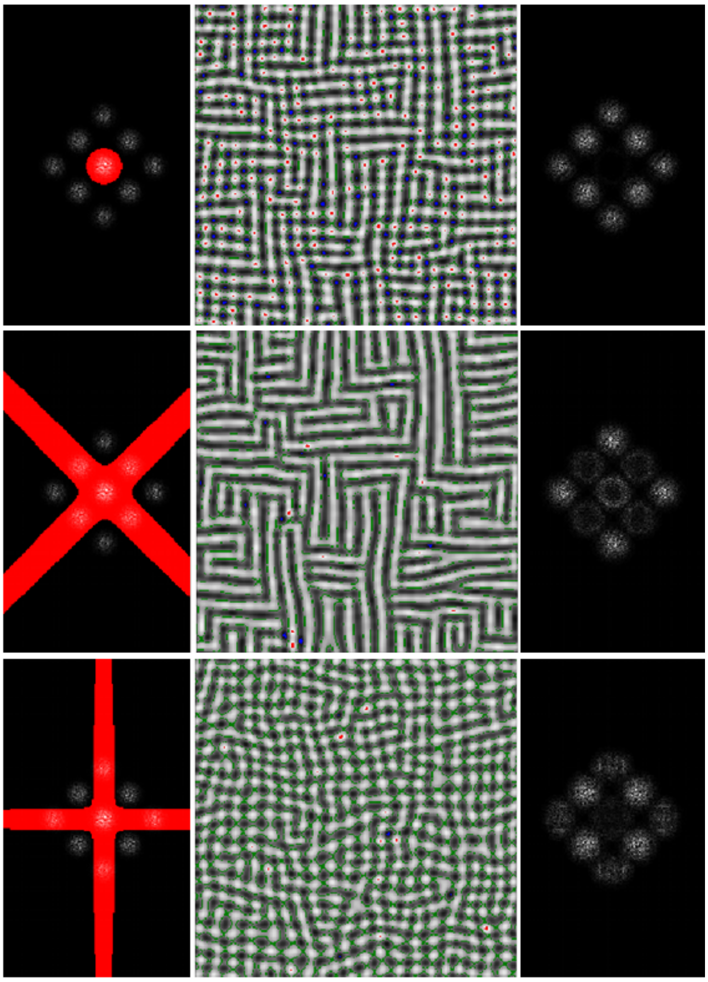
brute Gabor noise:
spectrum, image, E=fourier(f^2).



left: red: features selected for normalisation in spectrum. Signal will be normalized by 1/sqrt(ifft).

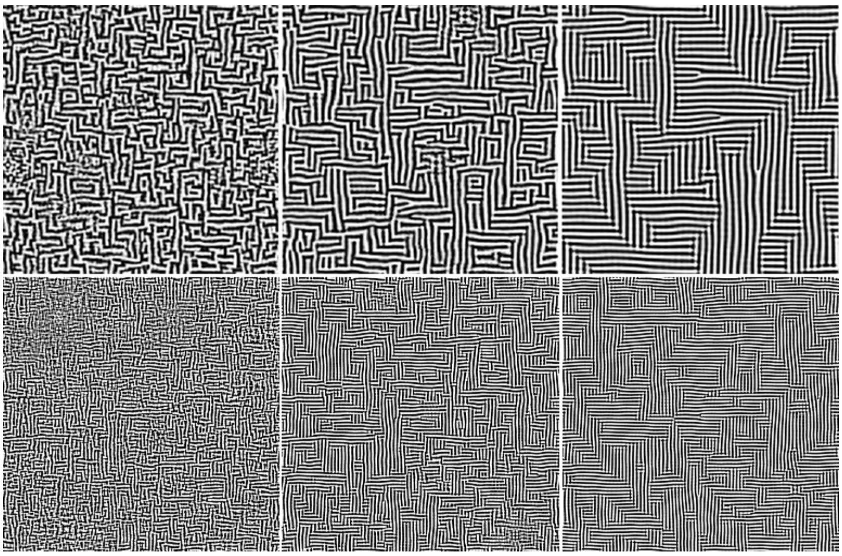
middle: the resulting signal (after iterating normalization+smoothing).

right: final E spectrum.
Note the ring recontaminating the clamped lobe, due to smoothing.



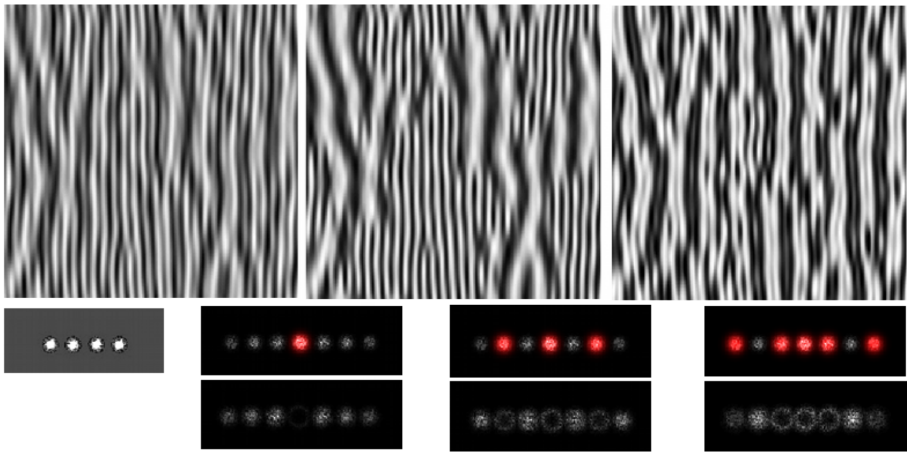
E

Did you know it was possible to make mazes with Fourier ? :-p
Middle case with various frequency and sigma lobes parameters.



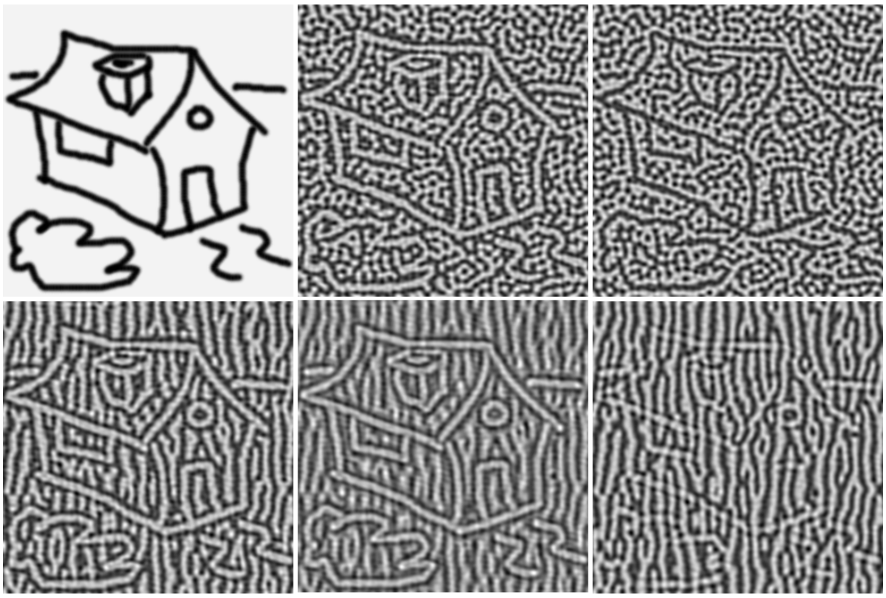
Double bi-lobe: (f+3f)

Left: cancels only LF of E.
Middle: cancels also interference.
Right: let only interferences.



2.4. More op

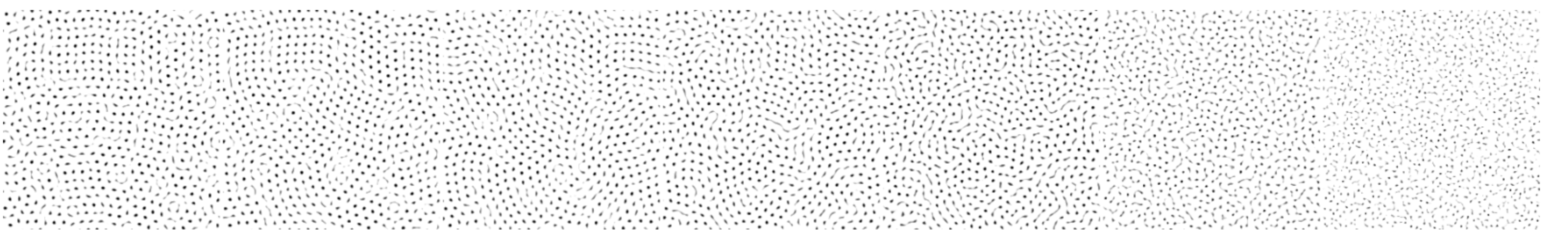
2.4.1. Mixing paint and Gabor



2.4.2. Reaction diffusion

Thresholding spatial gradient from blue noise to bi-lobe.

Thresholding blue noise, sigma = 1/16 → 4

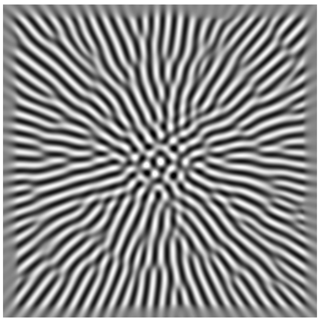


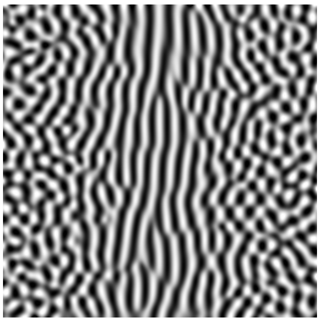
2.5. Image-space Fourier

Purpose:

- On the fly, on GPU, etc.
- On non-flat surfaces.
- Gabor noise with space-varying parameters.
- When periodicity is unwanted.

<u>Space-varying example</u>:
bi-lobe radially oriented relative to space cannot be generated through FFT. Simple normalization (cull the isotropic BF lobe) can be done in this simple case, but not the smoothing, for instance. Plus we might want space varying parameters in normalization and smoothing.





2.5.1. Convolving signal

Easy (LF) or costly (lobe)

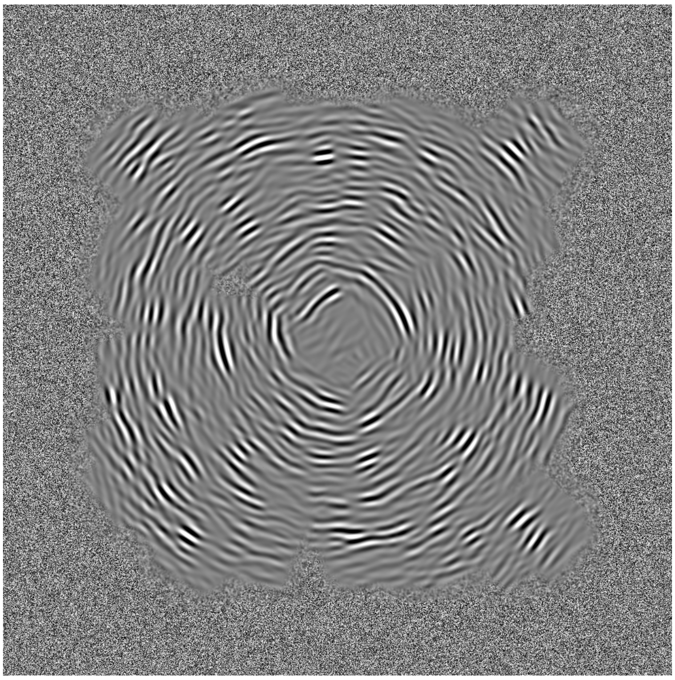
2.5.2. Windowed Fourier in tangent space

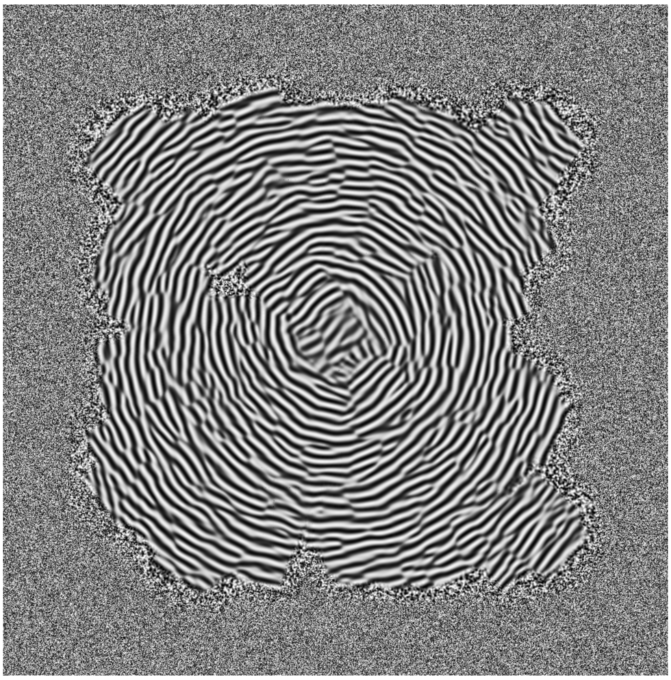
But this have several inconvenients:

- No space-varying is possible within the window. For very curvy, radial, or nody patterns this give poor polygonal or patched results.
- Inserting means blending, which corrupts the normalization we just did.
- Normalization could be done on the big image, but so long space varying treatments, including varying local Jacobian for not-flat surface.

Left: bi-lobe Gabor on rotated windows.
image = 1024x1024, 200 windows 128x128

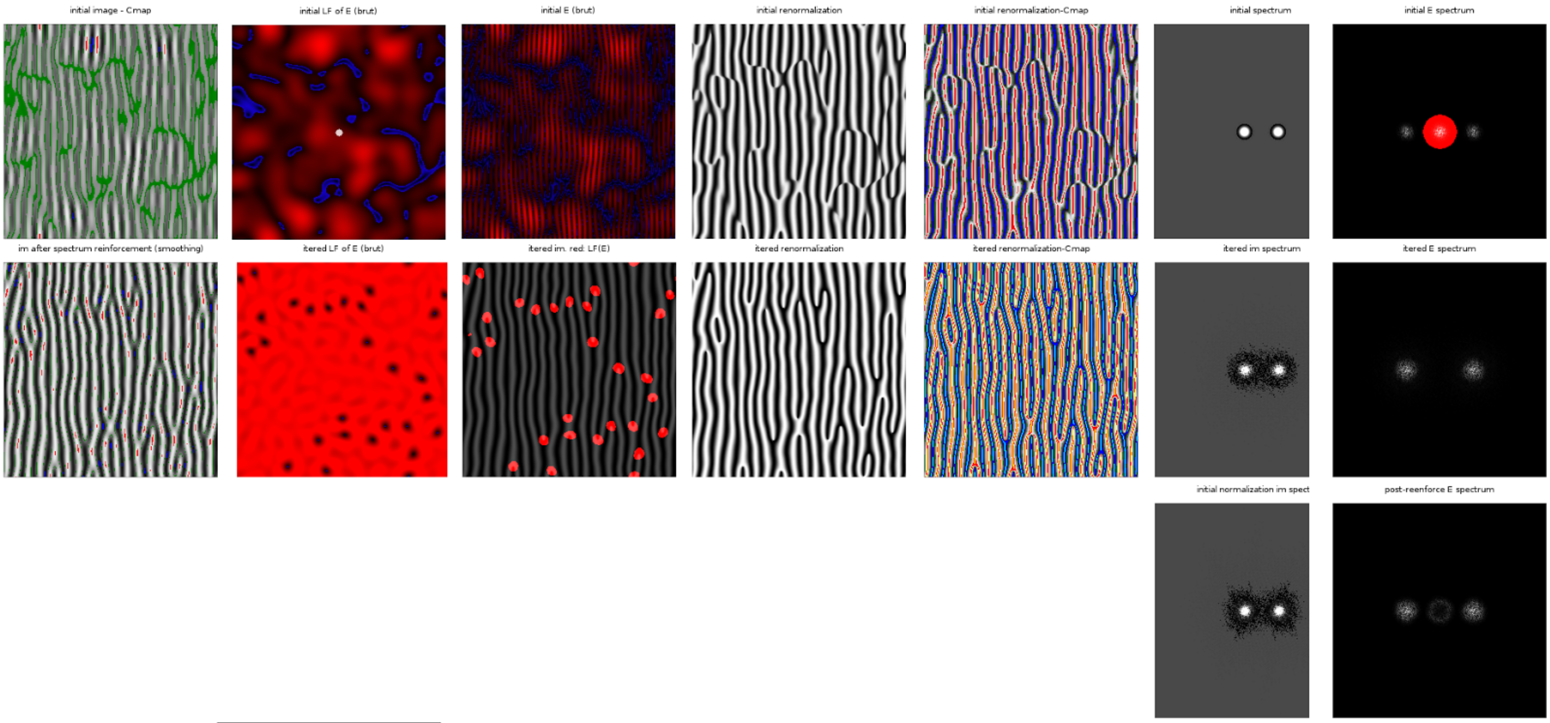
Right: global normalization (local looks ugly).

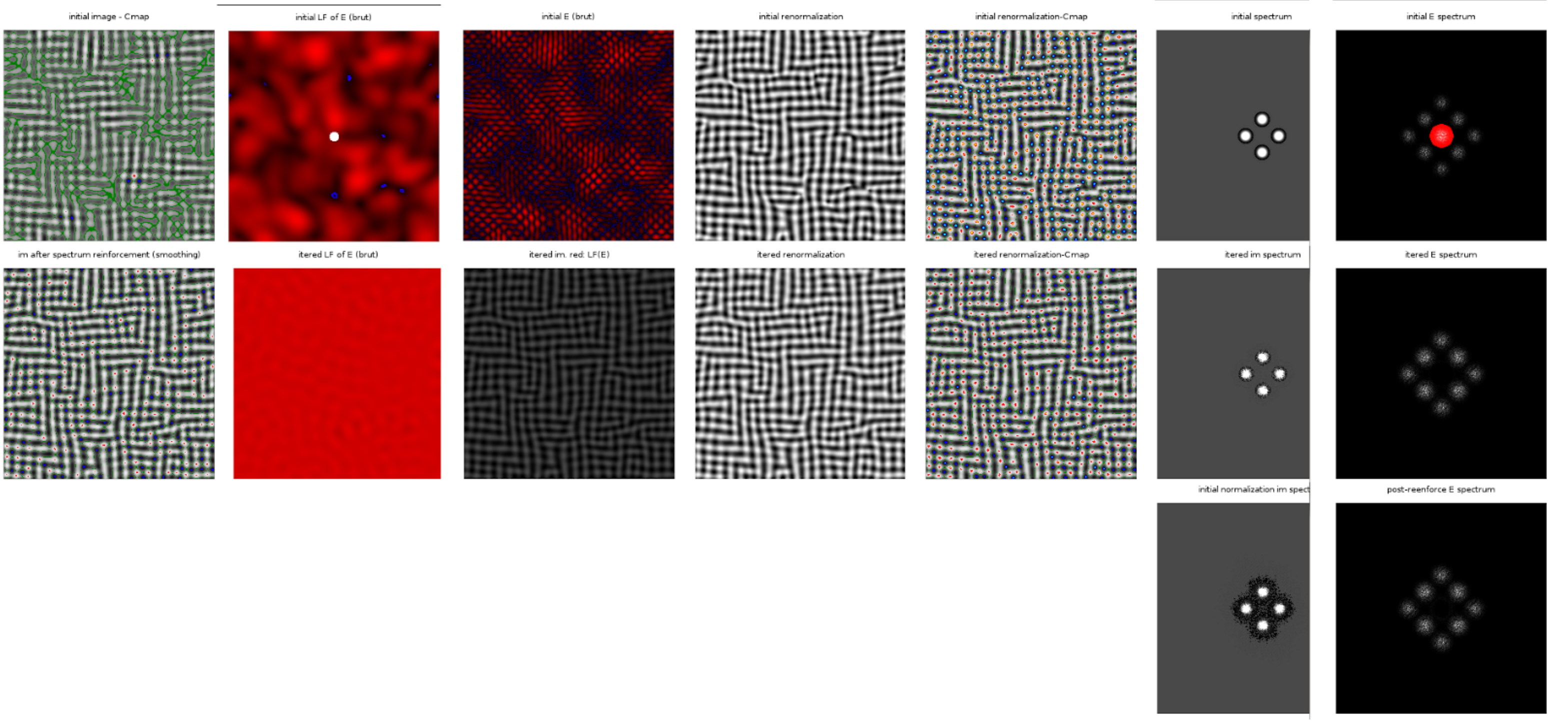




2.6. There is a wolf - Evil in the details, worsen while trying to fix it

2.6.1. More lab





2.6.2. Issues (sometimes)

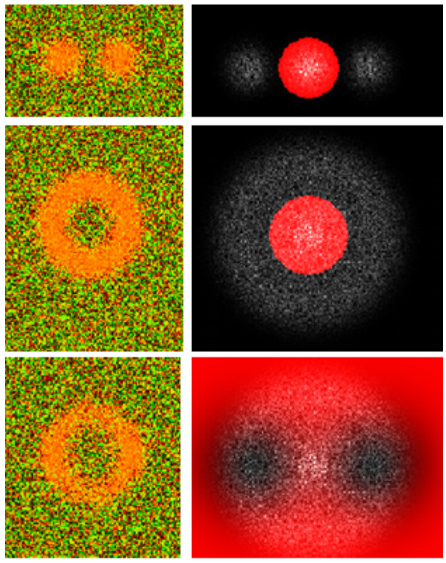
- normalization induces sharp features → smoothing requested → cause more issues
- undershoots: negative values in LF(contrast). Meaning ? what to do with that ?
- div by zero (or close).
- overshoots after normalization
- some remaining variance in max
- histogram bounds vs stddev
- histogram profile vs stddev
- which LF filter choosing ? (box/gauss in spectrum)

Smoothing: how ?

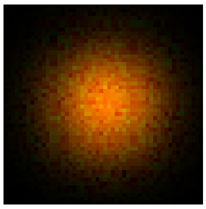
- cull added HF
  - reenforce initial spectrum template
  - reset initial spectrum
- smoother normalization
  - 1/(eps+contrast)
  - smooth(1/contrast)
  - 1/smooth(contrast)

NB: iterative :-(

2.6.3. What happens to phases / modulus ?



It occurs that **phases are quite well conserved**:
left shows difference of phases (before/after normalization+iteration) for bi-lobe, blue-noise, blue-noise E enforced to bi-lobe.
i.e. $RGB = (\Re(e^{i(\phi-\phi_0)}), \Im(e^{i(\phi-\phi_0)}), 0)$
Still, they are not exactly same: cf zoom on lobe →



Also the gaussian lobes in modulus are well conserved, but get salty:

→ what get encoded in the salt in phases and modulus?

2.7. To be continued…

- More understanding is needed for “smoothing”
- Applied math solvers of “enforce variance spectrum, keep signal spectrum”