Understanding and controlling contrast oscillations in stochastic texture algorithms using Spectrum of Variance

supplemental material

Fabrice NEYRET *(LJK / Grenoble University & INRIA)*, Eric HEITZ *(Unity Technologies)*

Please find below the paper's figures in full size, plus a few extra images.

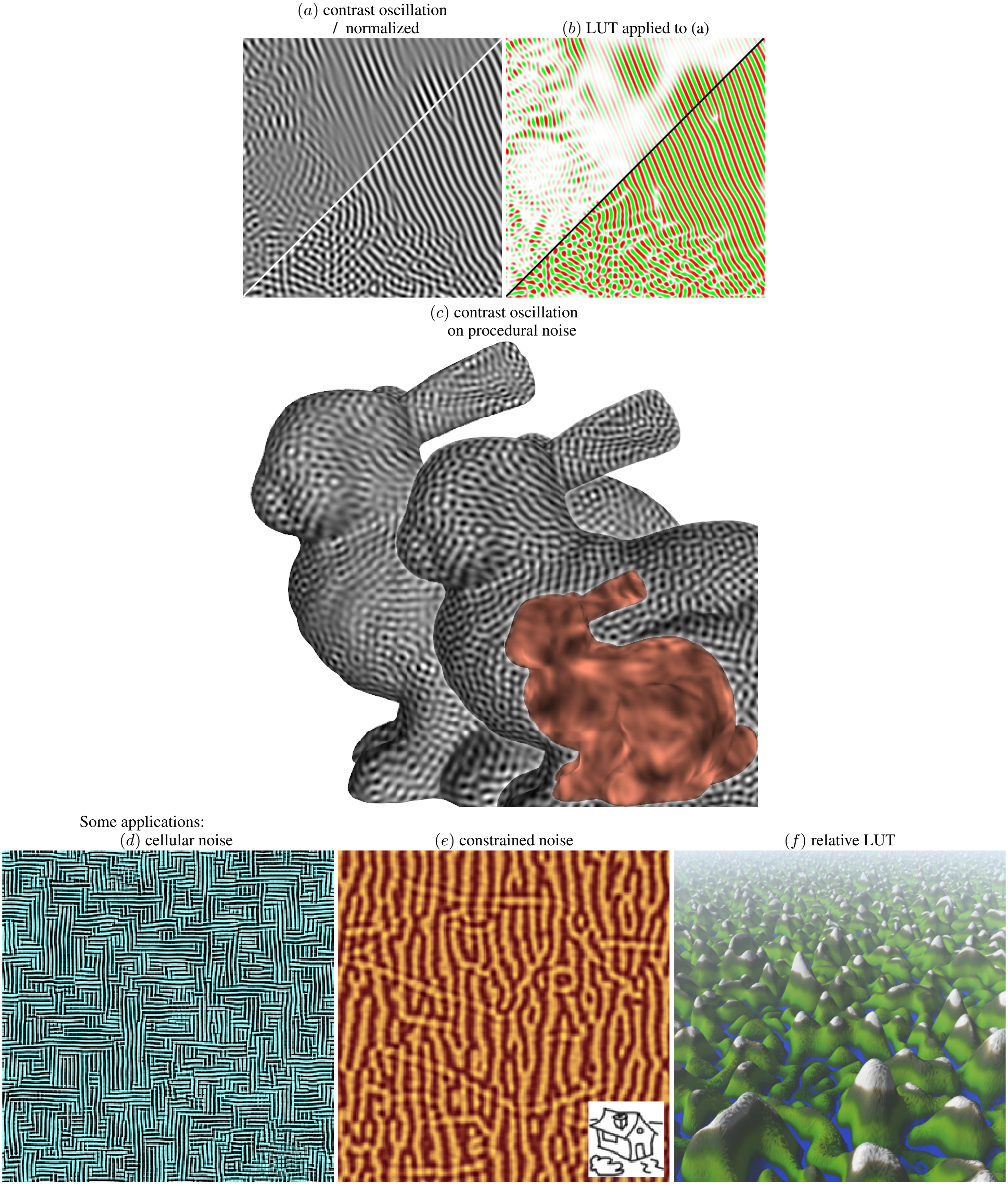


Figure 1: *Power-spectrum based texturing algorithms (e.g., Gabor, Fourier synthesis) suffer from unexpected low frequency contrast variations* (a,b,c top) *even when the spectrum has no low frequency (the contrast field is display in red in* (c)*). This prevents precise authoring with non-linear transform, like color LUT* (b top)*. Our renormalization method allows to control the stationarity* (a,b,c bottom)*. It also opens many doors for noise authoring such as the generation of reaction-diffusion-like strips and spots* (b bottom)*, cellular-like patterns* (d)*, content constraints* (e)*, or the parametrization of height maps relative to local extrema* (f).

Figure 2: *Typical results from Figure 7 of [Lagae et al. 2009] show contrast oscillation* (left) *despite no low frequencies are present in the power spectrum* (snippets)*, and pleasant but totally uncontrolled heterogeneities in the resulting pattern* (middle and right).

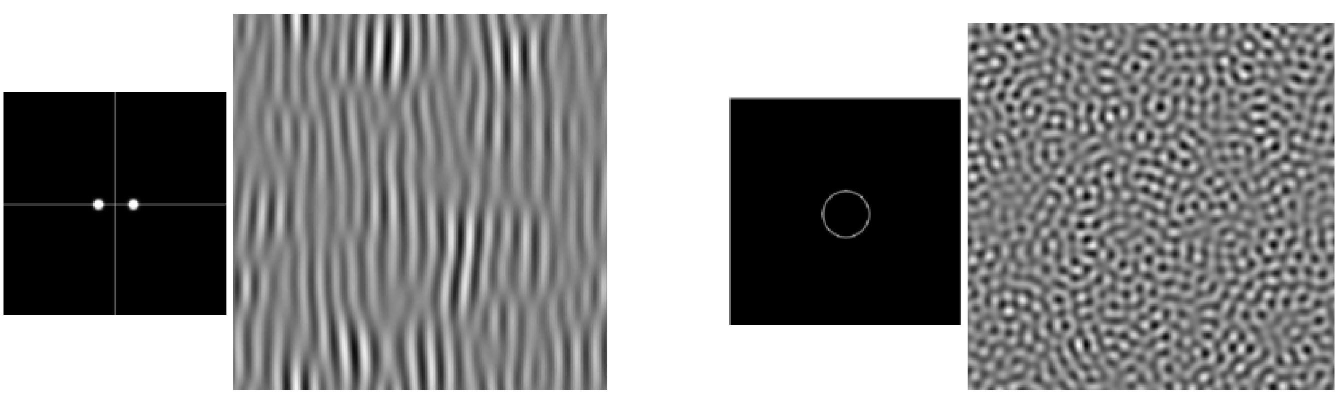


Figure 3: *Bi-lobe* (left) *and blue noise* (right). *Despite no low frequencies are present in the PSD* (black snippets)*, the contrast of the generated texture varies, at low frequency.*

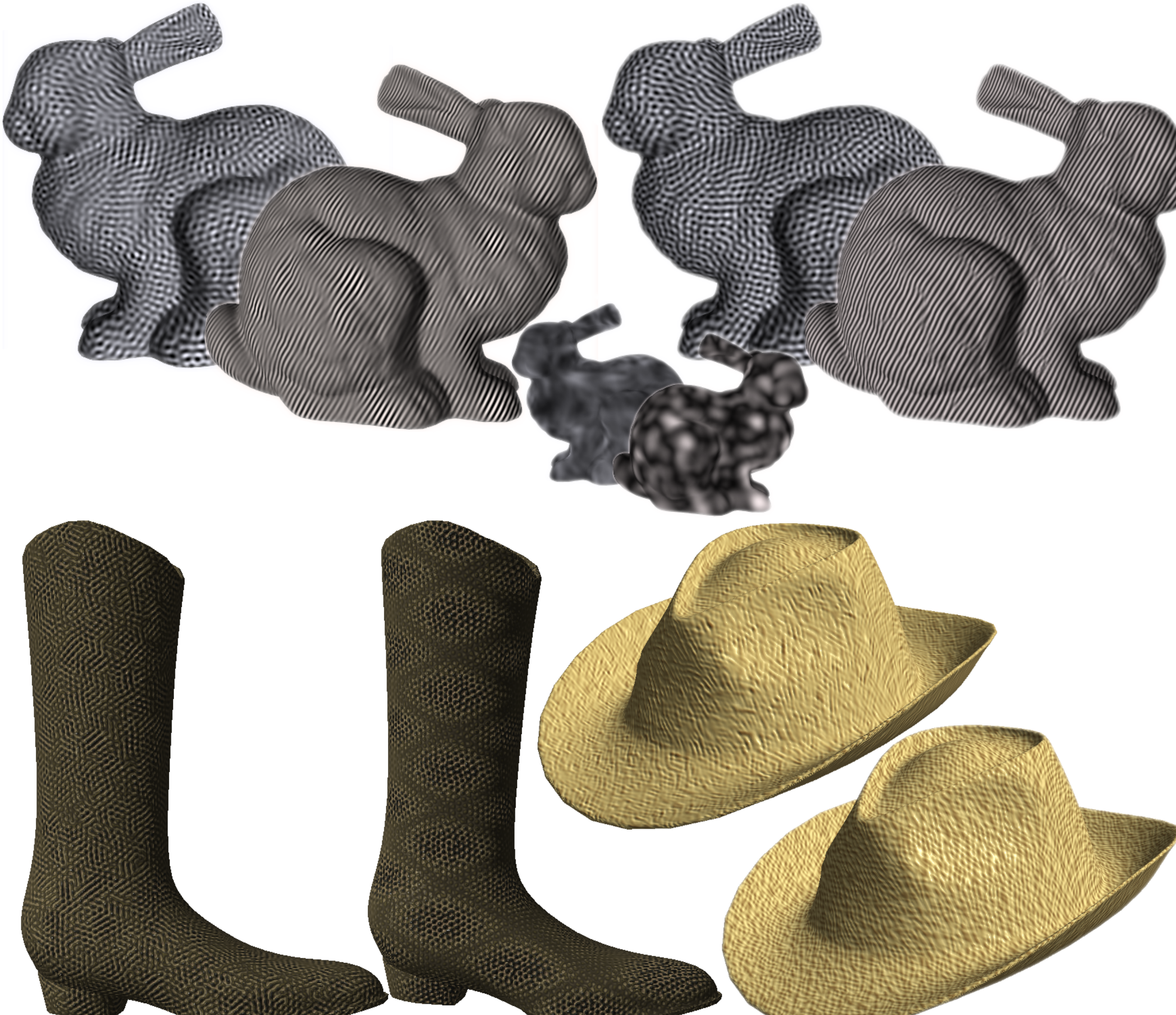
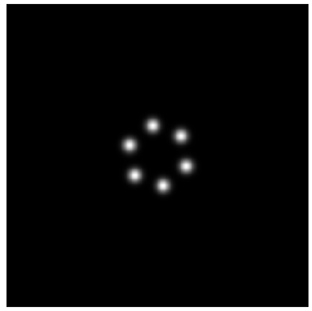
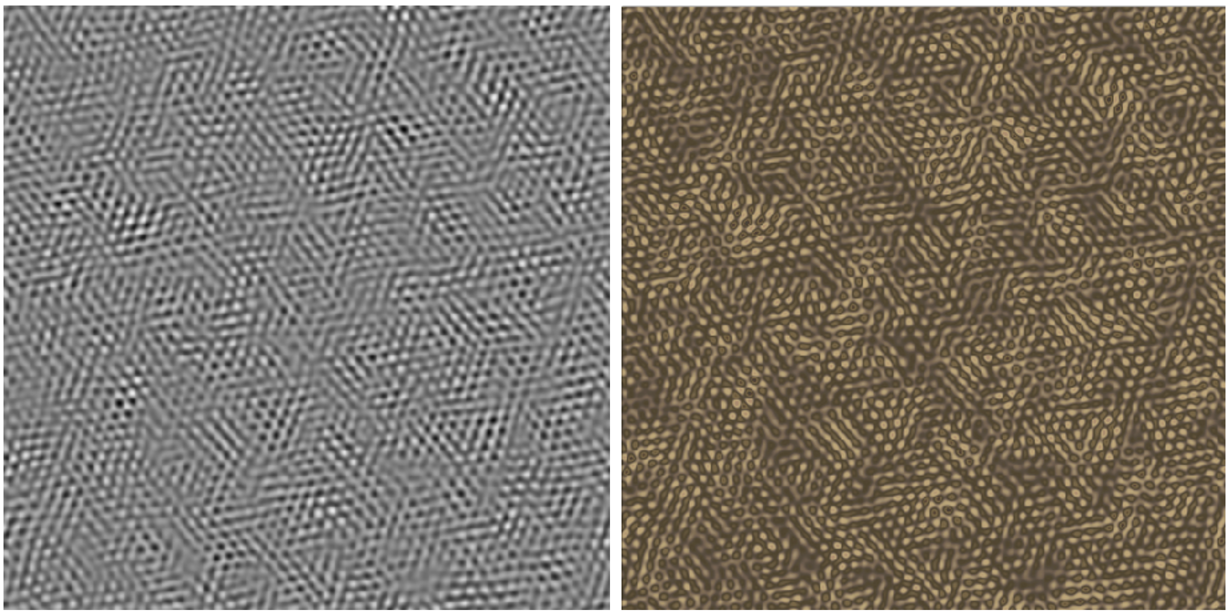


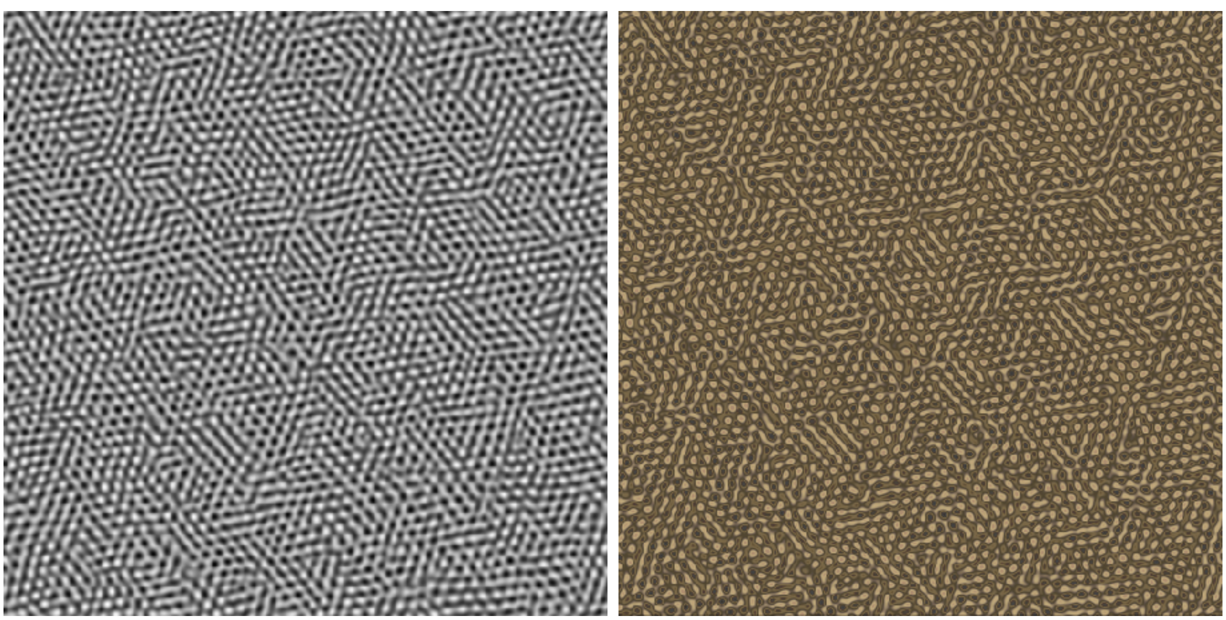
Figure 4: Left: *Two procedural Gabor noises (bi-lobe strips and blue-noise spots) showing unspecified contrast variations (explicitly displayed in snippet), and its correction. Here, the variations even misleads relief perception via false shading. Note that this example is implemented using non-parametric real-time GPU Gabor noise, with contrast evaluated on mesh vertices in the vertex shader.* Middle: *The boot of [Lagae et al. 2009] (cf Figure 2) without the artifacts, resp., with controlled variance applied to the renormalized Gabor before the color LUT.* Right: *The hat of [Lagae et al. 2009] without and with correction.*



(a) PSD imitating the one of [Lagae et al. 2009]'s boot seen in Figure 2.



(b) corresponding raw noise texture, without and with color LUT.



(c) renormalized noise texture, without and with color LUT.

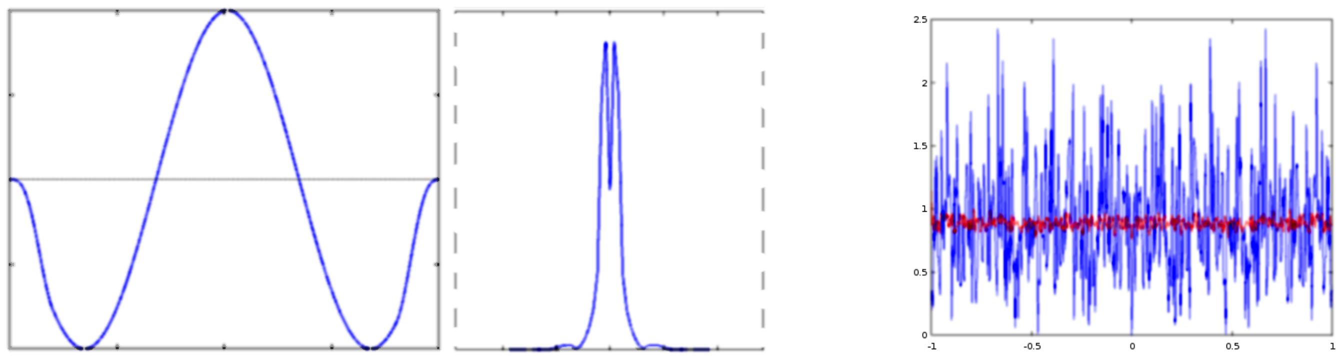


Figure 5: (Left): *Inharmonic sinus (windowed by a smooth kernel to ensure continuous-derivable wrapping) and its Fourier modulus.* (Right): *PSD of a realization of white noise by points distribution and Bernoulli's weights.* Red: *average of 100 realizations converges towards flat PSD.*

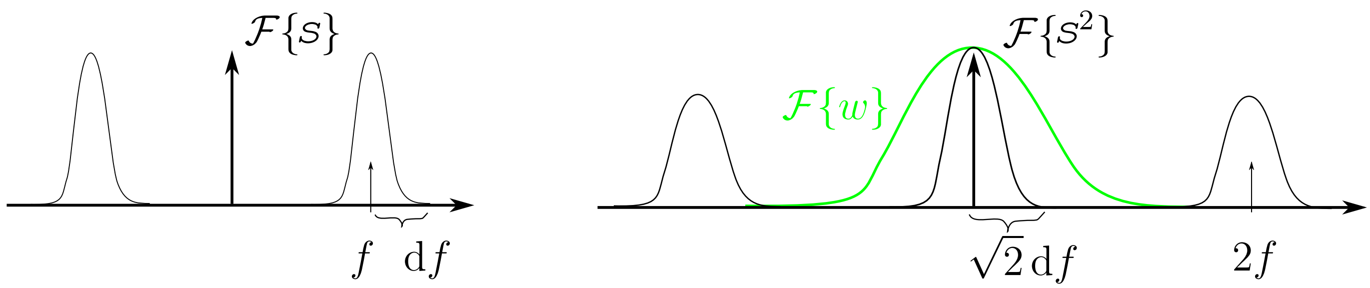


Figure 6: *Spectrum of a bi-lobe noise* (left) *and of its square* (right). *The windowed second moment of the signal corresponds to the filtering of the latter by $\mathcal{F}(w)$* (in green)*: it results in a LF lobe in Fourier, thus corresponding to LF contrast variations in image space.*

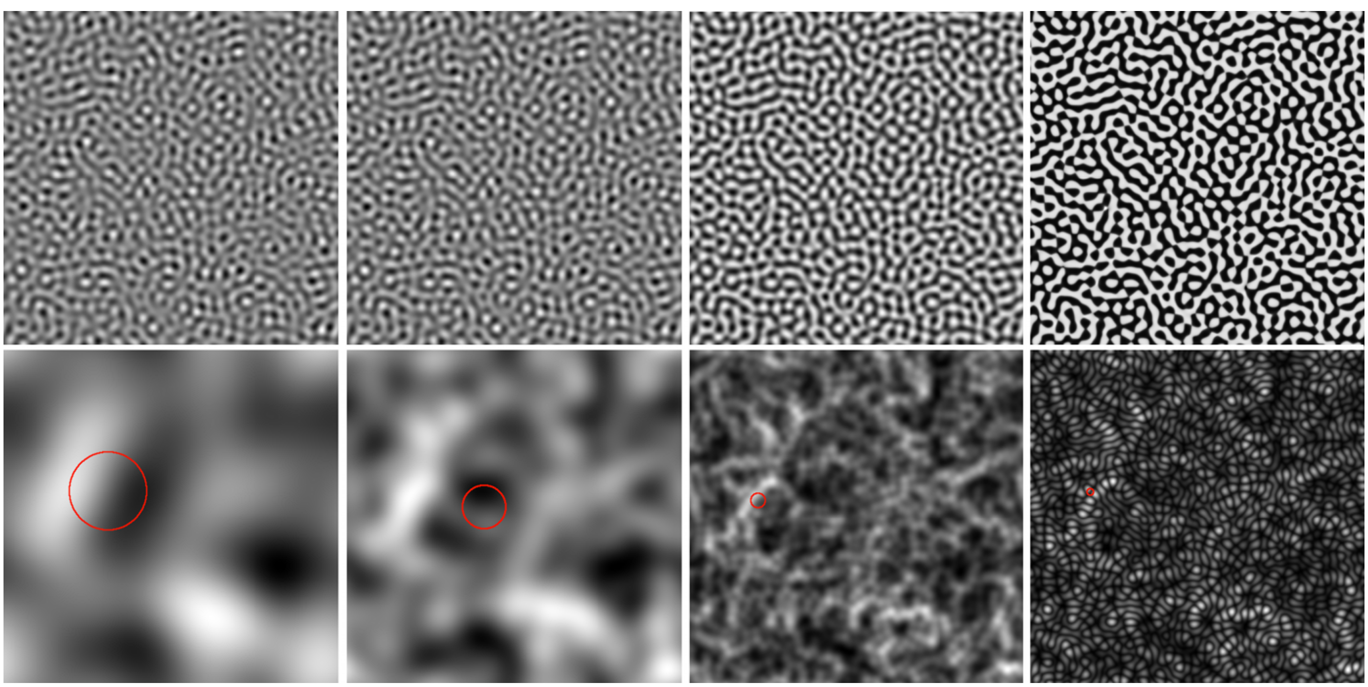


Figure 7: top: *Blue noise normalized using different stationarity scales L.* bottom: *corresponding normalization field, i.e., variance filtered at the target window size (figured in red). NB: $[min, max]$ is remapped to $[0, 1]$. The lower the frequency, the lower the amplitude of variations. On the third column the stationarity scale is the same as the wavelength of waves: each wave saturates the dynamics. On the fourth column the window scale is just a few pixels: the texture is forced to produce regions of min vs max value, with a sudden transition width equal to the stationarity scale.*

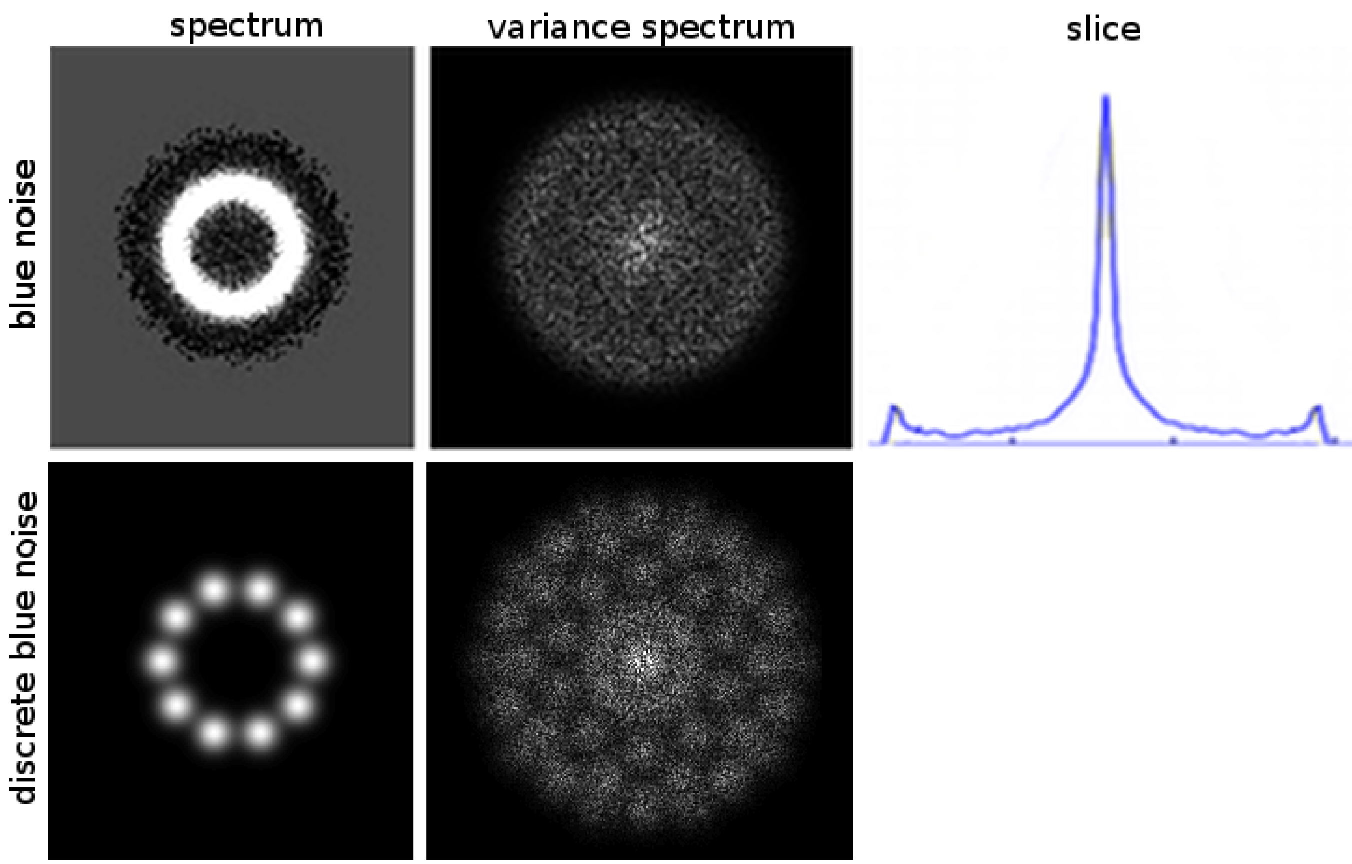


Figure 8: Top*: With blue noise spectrum* (left)*, the auto-convolution of the ring gives a central lobe not well separated to the "HF" ring in the variance spectrum* (middle and right)*. if "full renormalization" is aimed at, the exact stationarity scale to choose is arbitrary.* Bottom*: If the ring is approximated by a series of lobes, we recover a separate central lobe.*

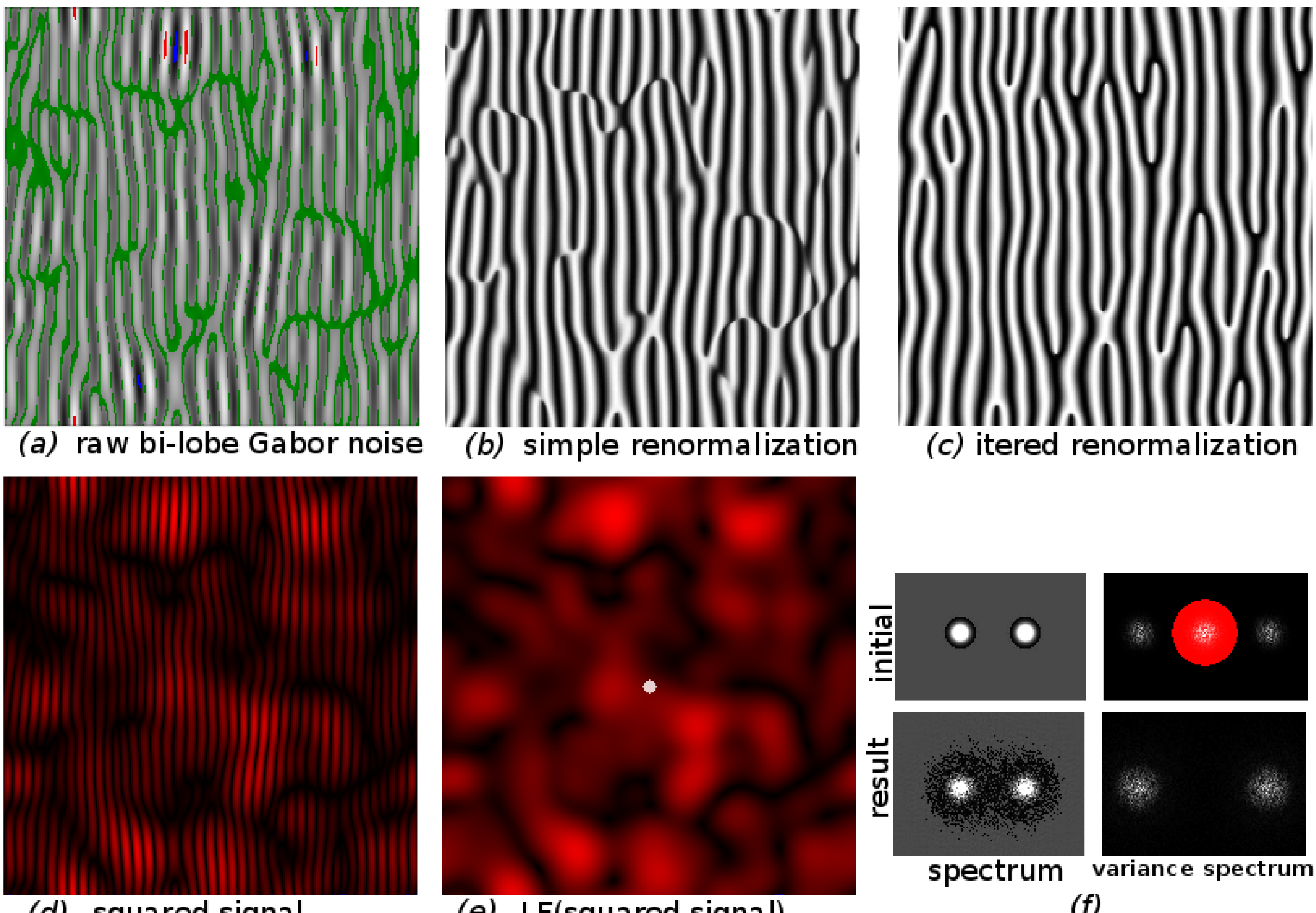


Figure 9: (a): *Raw bi-lobe Gabor noise. zero contrast locus are marked in green.* (b): *Simple normalization (showing locus of sudden inversion).* (c): *Our high quality scheme, iterating several normalization and spectrum reprofiling passes.* (d): *Squared signal.* (e): *Windowed variance (i.e., LF-filtered squared signal).* (f): *Signal spectrum (left) and variance spectrum (right) before (top) and after (bottom) normalization.*

```
im = real(ifft2(Kernel.*exp(2*I*pi*rand))); % Gabor noise
imf_footprint  = (Kernel>0);                % template for spectrum reprofiling
% imf_footprint = Gauss(0,KernelRadius);    % smooth variant
minC = 1e-2;                                % to avoid div0 when normalizing

for i = 1:N  % --- iterations (for Quality mode)

    % --- renormalization (for both modes) -----------------------
    immean = mean(im(:));
    imE = (im-immean).^2;
    imEf = fft2(imE);               % variance spectrum
    imEf .*= LF_filter;             % keep only central lobe of variance spectrum
    imE = real(ifft2(imEf));        % envelope (i.e. LF) of contrast variations
    imE = abs(imE);                 % get rid of (rare & small) negative values

    imC = sqrt(imE);                % variance -> contrast ( = std-dev )
    imC /= max(imC(:));             % normalized contrast: max = 1 -> keep untouched

    imK = (minC+1)/(minC+imC);      % renormalization factor
    im = immean  + (im-immean).*imK;
  % im = newmean + (im-immean).*imK * newStd/max(imC(:)); % variant

    % --- spectrum reprofiling (for Quality mode) ----------------
    immean = mean(im(:));
    imf = fft2(im-immean);
    imf .*= imf_footprint;          % clamping spectrum to its initial footprint
    im = real(ifft2(imf));          % to smooth-out high slopes at degenerated locus
    im = .5+.5*im/max(abs(im)(:));% normalize image to [0,1]

end
```

Figure 10: *Octave/Matlab pseudo-code for quality normalization, relying on the FFT.*

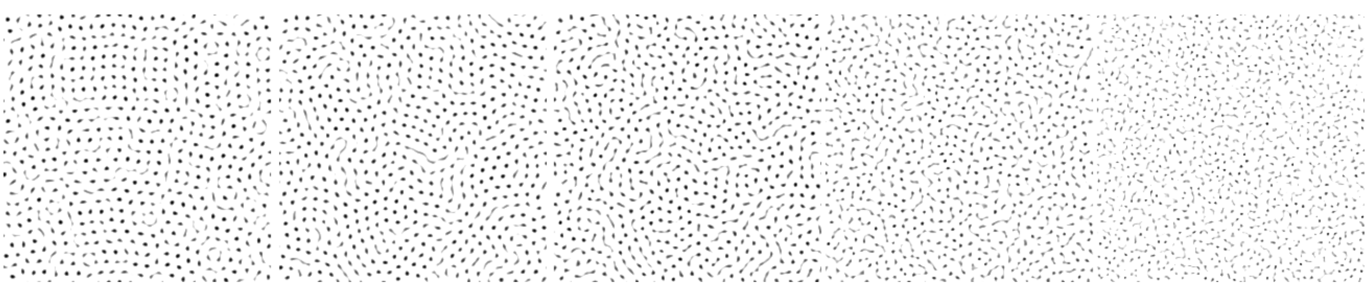


Figure 11: *Points-like distributions obtained with a LUT selecting the peaks of normalized blue noise. The better the normalization the smaller can be the spots (with no miss or excessive deformation).*

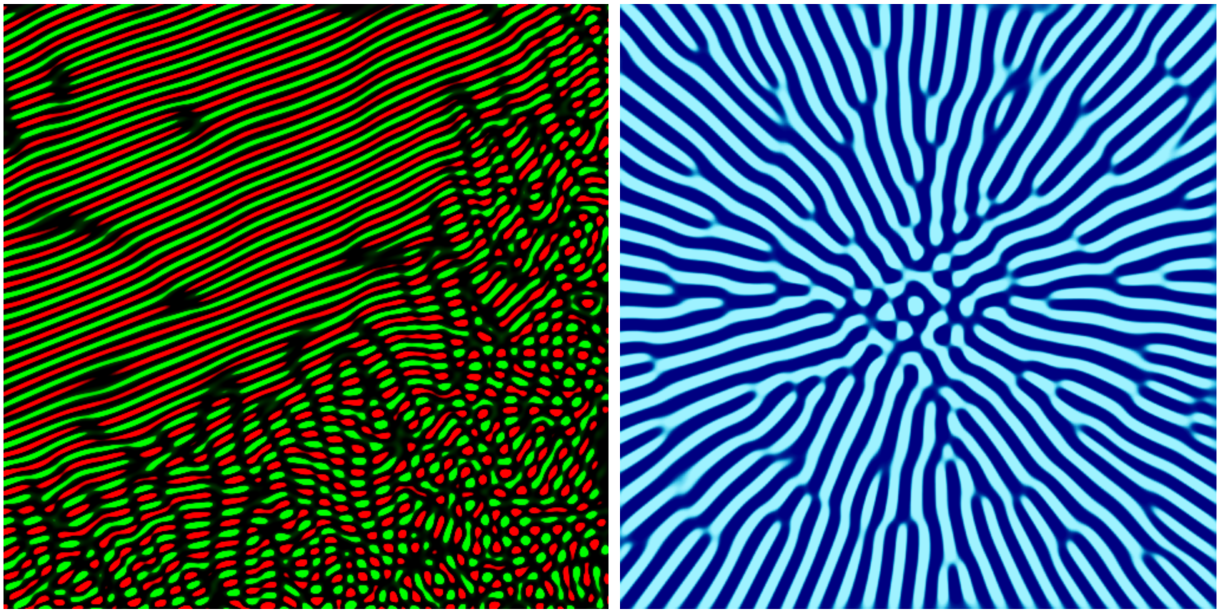


Figure 12: *Reaction-diffusion looking pattern obtained by applying a color look-up table on a totally normalized (stationarity scale = pattern size) spatially varying Gabor noise. In these figures, no spectrum reprofiling was done.*

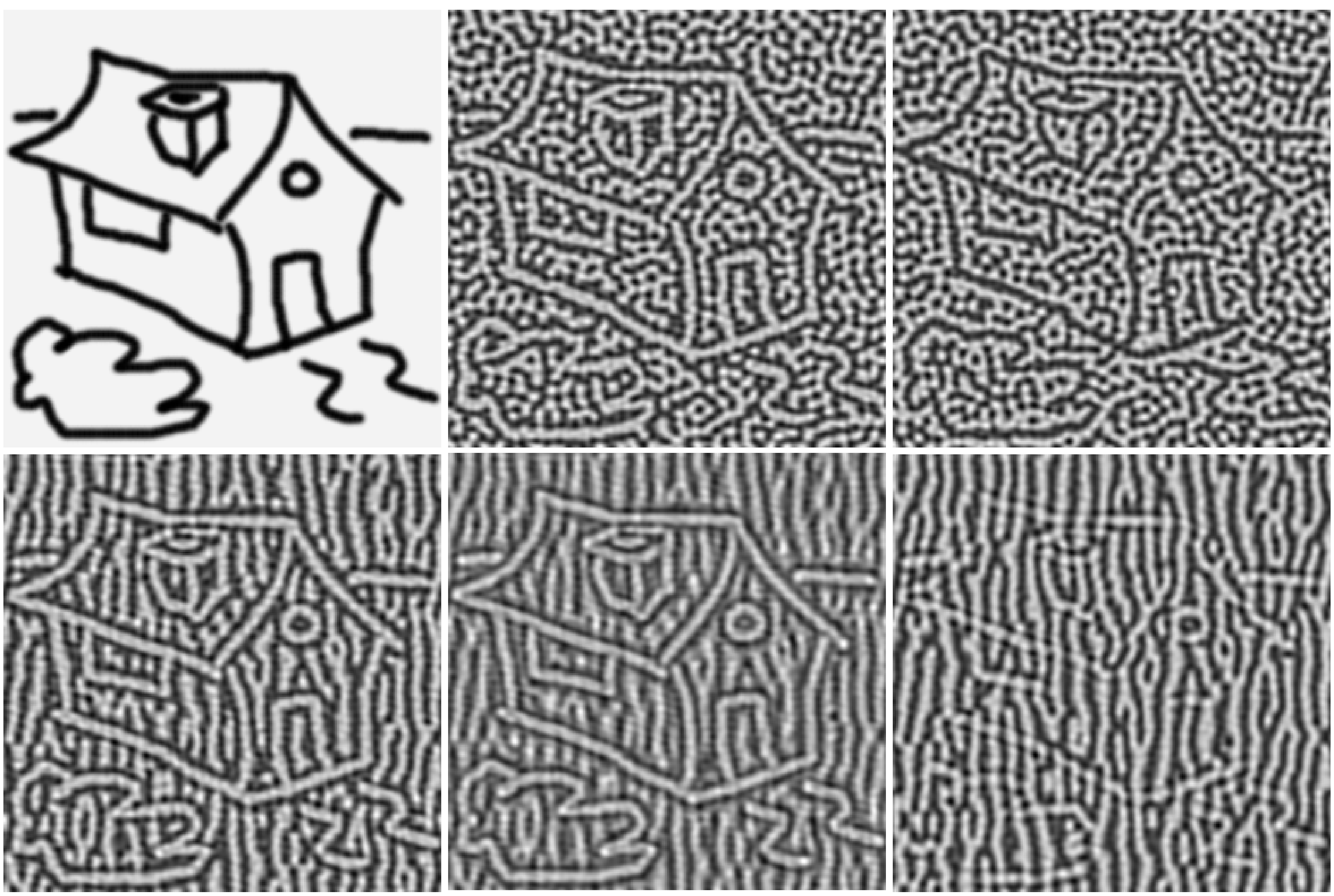


Figure 13: *Normalization of* `lerp`*(drawing,noise,w), with different noises and weights w. The drawing can be either an image or interactive. Here the drawing was given at once, but similar results are obtained by interactively painting it over the generated noise during the iterations loop.*

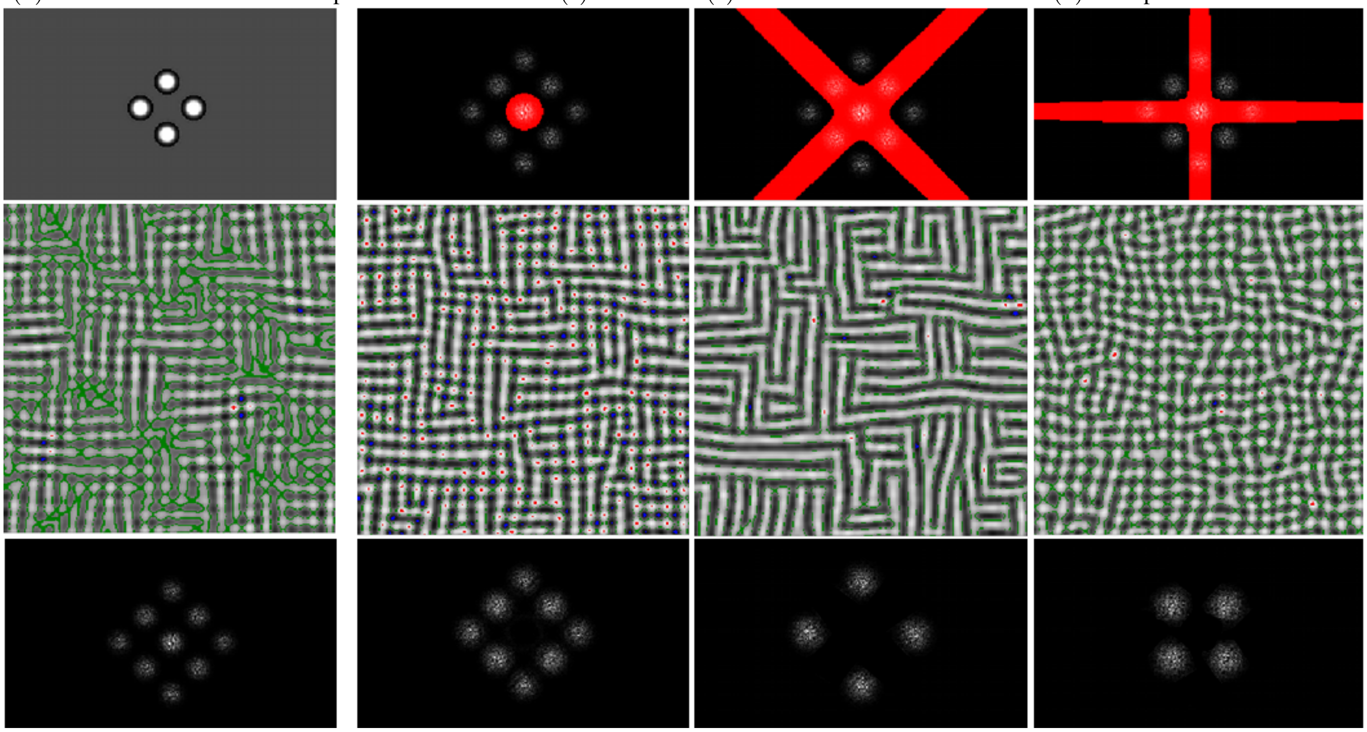


Figure 14: *The variance spectrum of a quadri-lobe Gabor noise shows 9 lobes: 4 correspond to the initial lobes, 4 to interferences between lobes, plus the LF lobe. Besides the simple filtering-out of the central lobe (b), we can choose to remove interferences (c), or to keep only them (d).* a, top and bottom*: initial spectrum and variance spectrum.* b,c,d top and bottom*: variance spectrum before and after selected normalization. In red: the filter selecting the features collected for renormalizing the signal.*

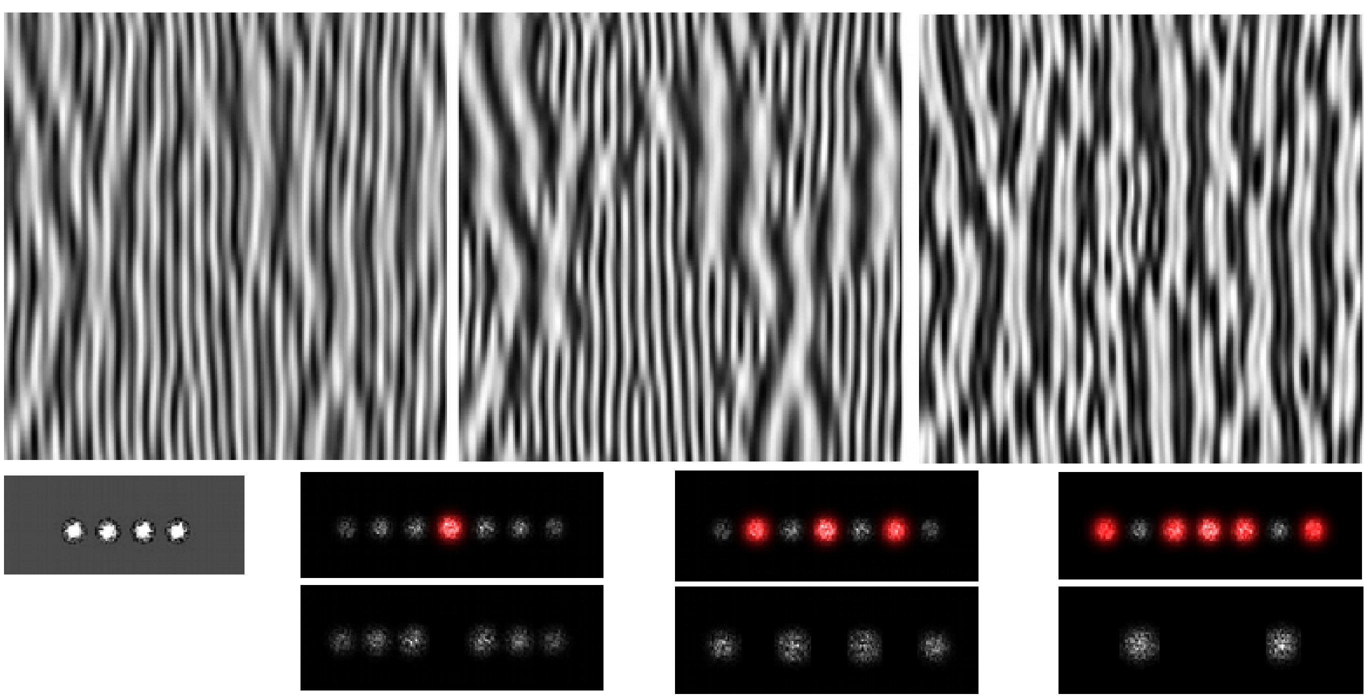


Figure 15: *Same as Fig. 14 with aligned quadri-lobes.*

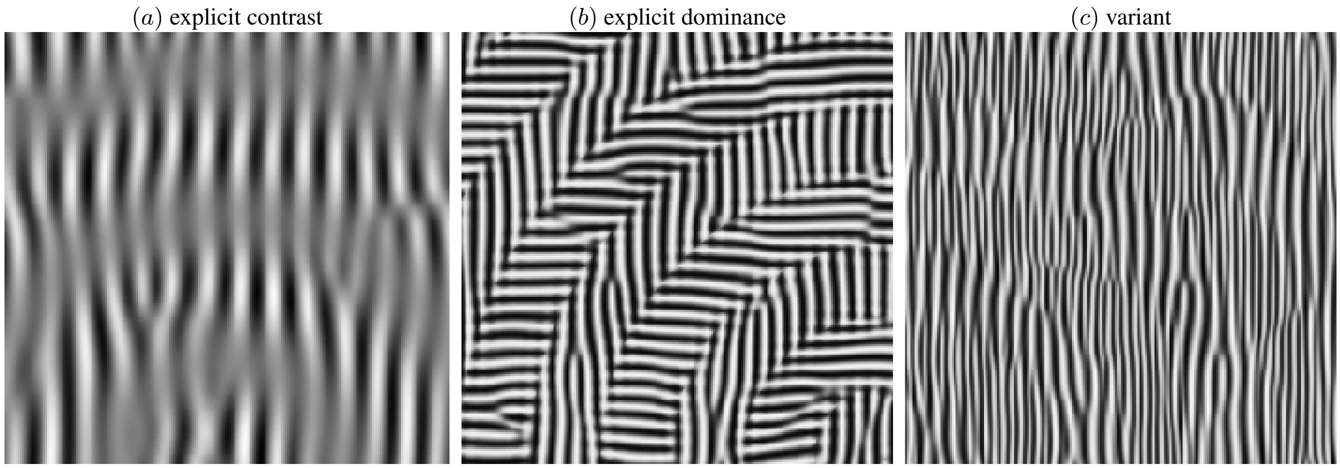


Figure 16: (a): *Explicit contrast control by multiplying totally renormalized signal by a target contrast map.* (b, c): *Explicit control of the lobe preference by separate pre-normalizing, lerping according to a dominance map, and re-normalizing. PSD = orthogonal vs aligned quadri-lobes.*

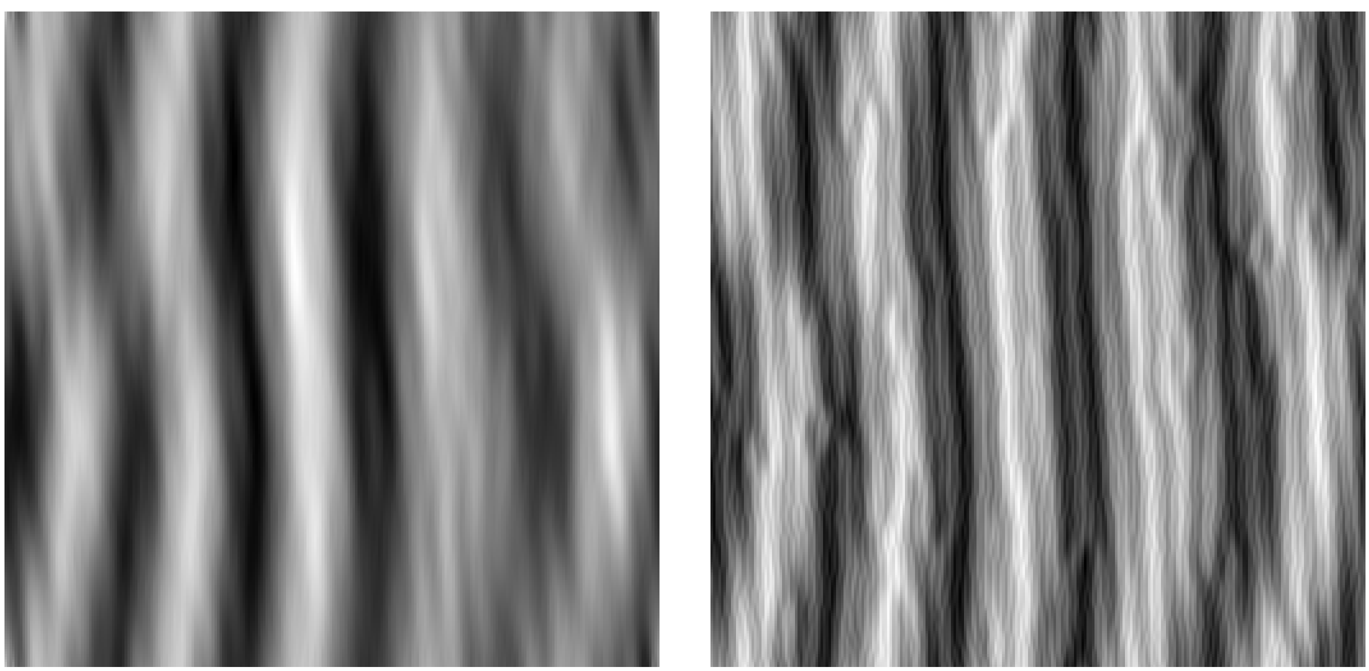


Figure 17: *For multiscale lobes (here, aligned hexa-lobes), we can normalize bi-lobe layers separately* (right) *rather than together* (left)*, to preserve the perceptual hierarchy.*