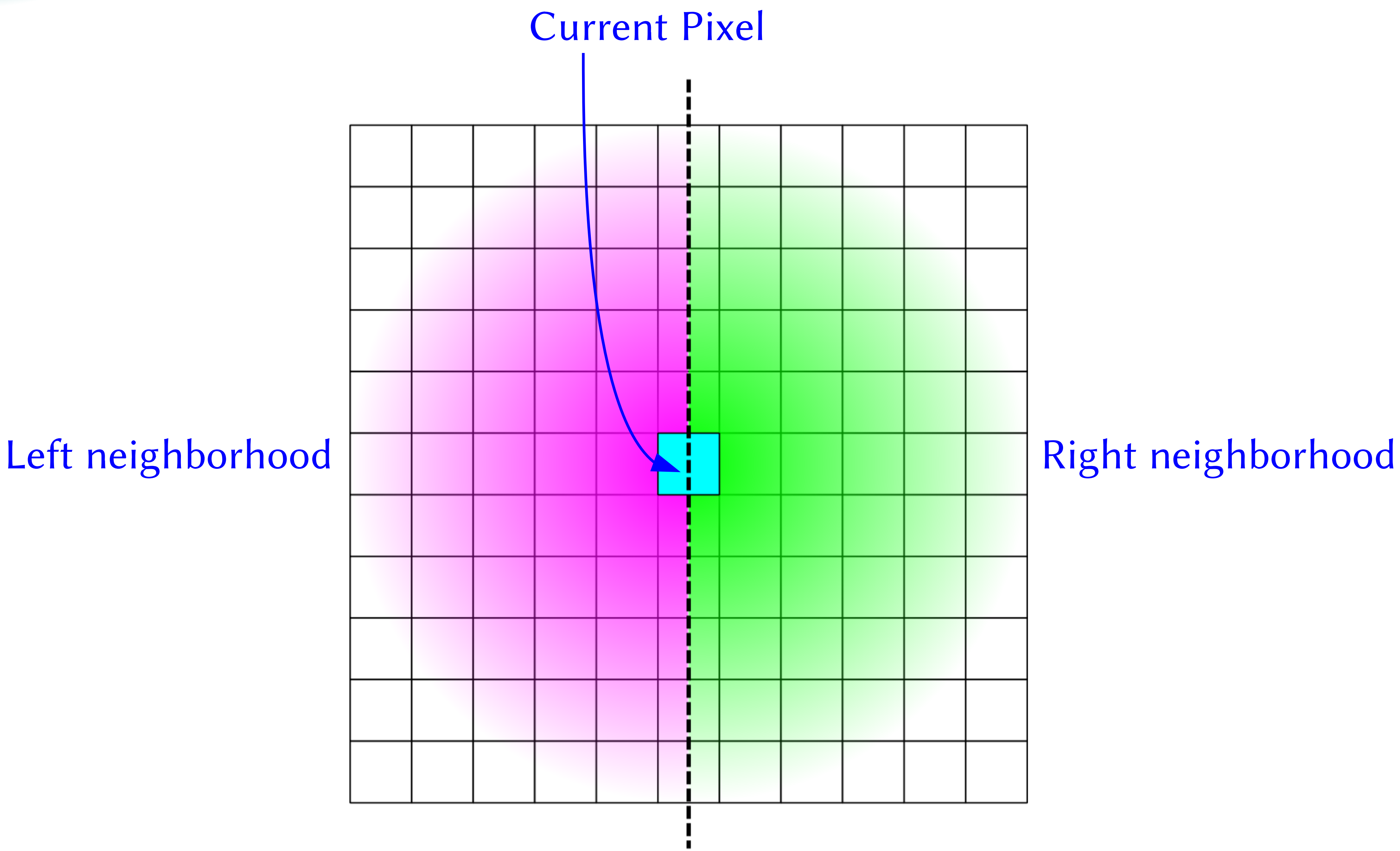
Multiscale Image Analysis with Stochastic Texture Differences

Nicolas Brodu, Hussein Yahia

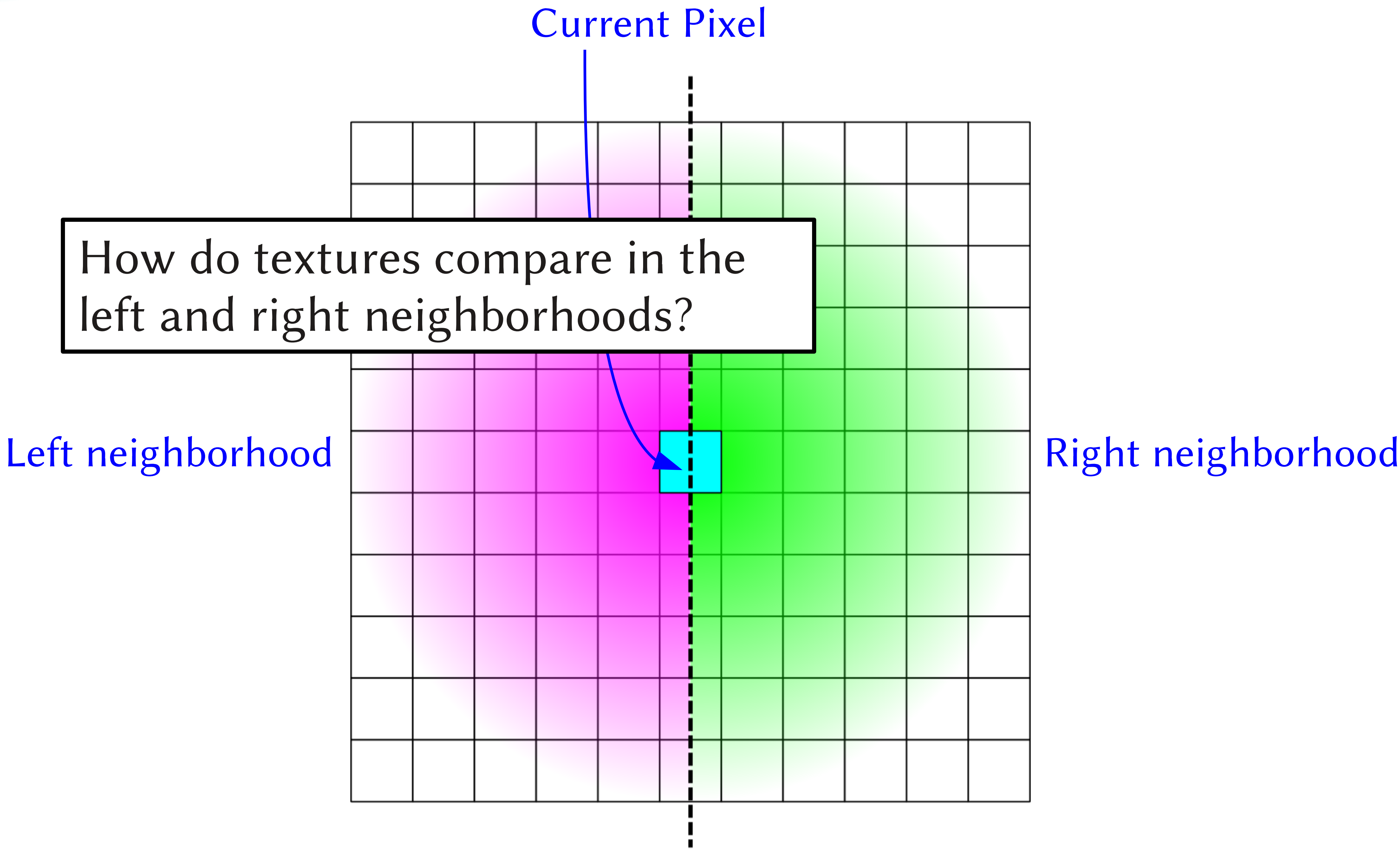
INRIA Bordeaux Sud-Ouest

16 / 01 / 2014

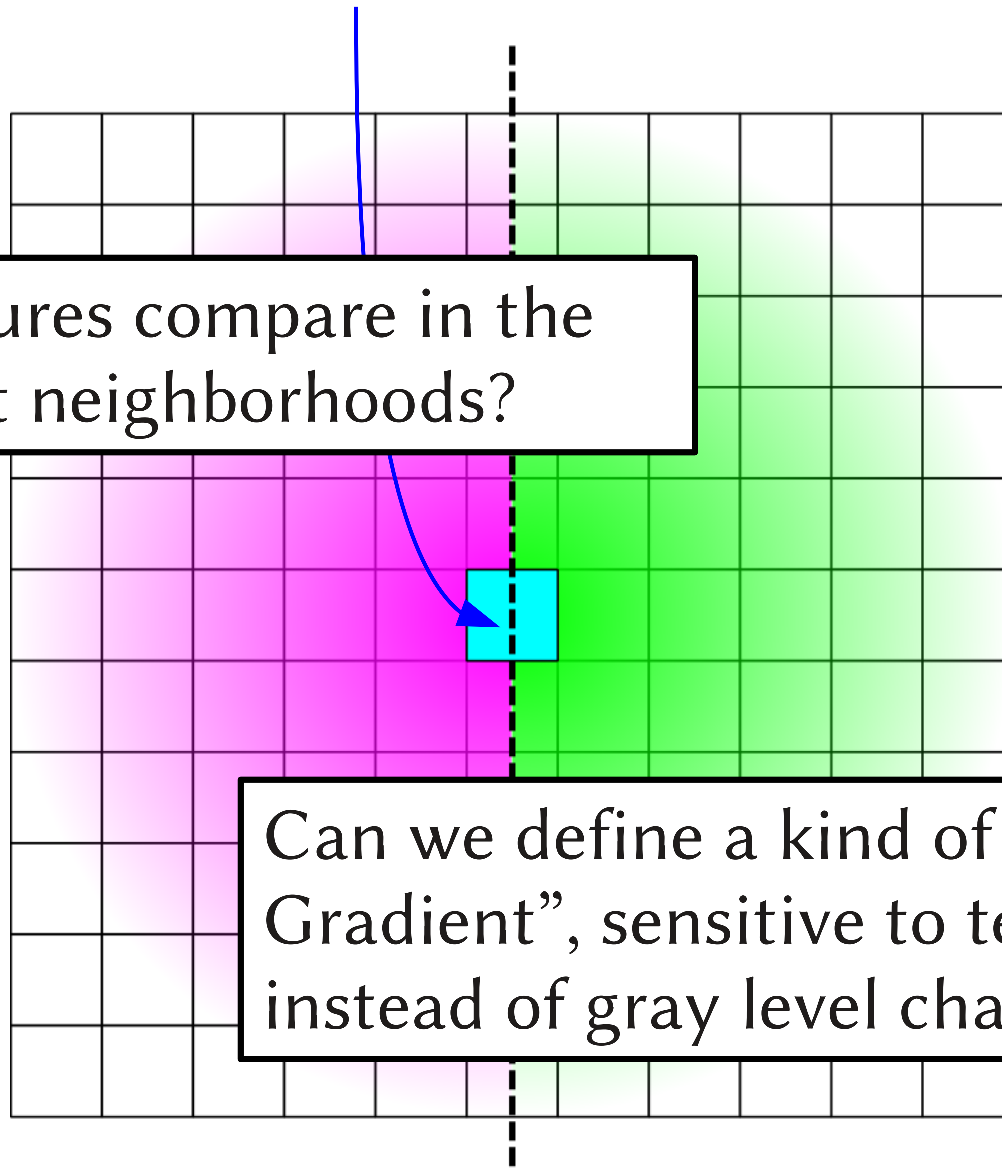
The Idea



The Idea


Current Pixel
How do textures compare in the left and right neighborhoods?
Left neighborhood
Right neighborhood

The Idea

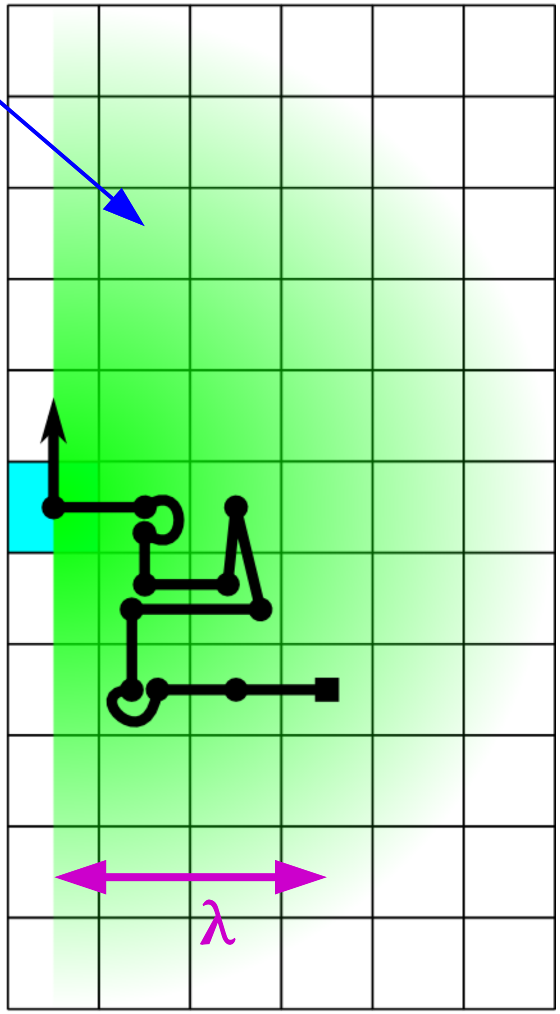


How do textures compare in the left and right neighborhoods?

Can we define a kind of “Texture Gradient”, sensitive to texture changes instead of gray level changes?

Texture Representation

Probability distribution H
to visit each pixel:
- Half-Gaussian
- dev. λ = **length scale**
  for the neighborhood



Random path:
- Start point ■ drawn
  according to H
- Transitions drawn so H is
  the limit distribution of
  the implied Markov Chain
- Average spatial extent = λ
  ⟺ average path length = m

Texture Representation

Pixel values $v \in V$ (e.g. [0..255])

Series of pixel values $s \in V^m$

m=13

s = (233, 194, 120, 120, 134, 191, 170, 191, 134, 133, 133, 108, 159)

Texture Representation

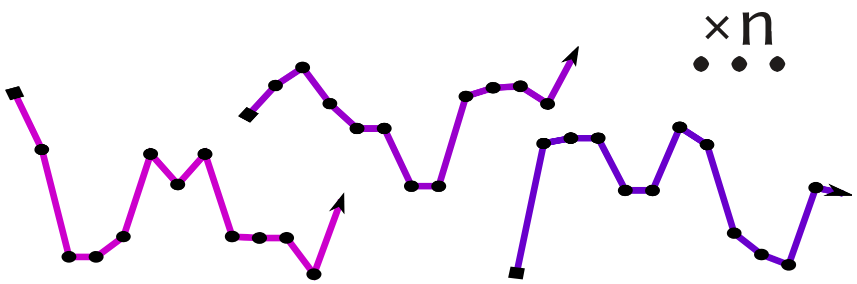
Texture = Probability distribution over V^m

– Distribution of **sequences**

– Collectively, these sequences characterize how pixel values evolve in the texture

Observed series = samples

– Collect *n* observed series



Estimator for the probability distribution

– Use a Reproducing Kernel Hilbert Space $\mathcal{H}$

– Use a characteristic kernel k such that $k(s,\cdot) \in \mathcal{H}$

– Empirical estimator: $P \triangleq \frac{1}{n}\sum_{i=1}^{n} k(s_i,\cdot)$

Comparing textures

Texture on the left = P, on the right = Q

- Use the RKHS norm: $d(P,Q) = \|P-Q\|_{\mathcal{H}} = \sqrt{\langle P-Q, P-Q\rangle_{\mathcal{H}}}$
- $d^2(P,Q) \triangleq \left(\sum_{i,j} k(s_i,s_j) + \sum_{i,j} k(t_i,t_j) - 2 \sum_{i,j} k(s_i,t_j)\right)/n^2$

  with $\{s\}$ and $\{t\}$ samples from P and Q

This is the MMD test (Gretton *et al.*, 2012)

- Beats χ^2 or Kolmorgorov-Smirnov esp. for small n
- Error in $O(n^{-1/2})$ does **not** depend on $\dim(V^m)$, just n.

Valid for **any** kernel $\Rightarrow$ not limited to gray scale

- Vector data (RGB, hyperspectral), strings, graphs…

Scalar pixels (e.g. gray scale)

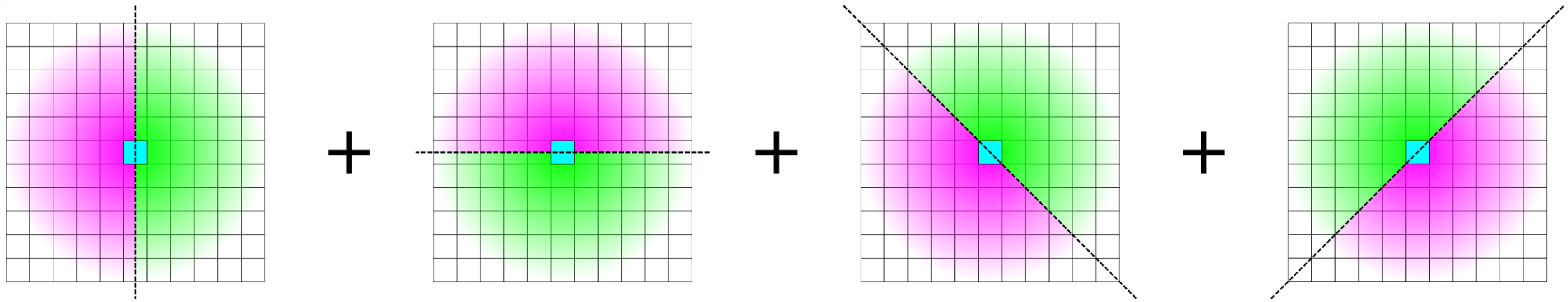
Scaling the data: $s_i' = s_i/\kappa$

- κ is the characteristic data scale, e.g. gray level difference, at which the texture is best described
- small κ : sensitive to small variations, but large gradients give similar k(s,t) values
- large κ : distinguish large gray level gradients, but small variations (e.g. noise) are ignored

The inverse quadratic kernel

- $k(s,t) = 1/(1+\frac{1}{m}\|s'-t'\|^2)$, using the norm in V^m
- Characteristic and faster than the Gaussian kernel
- Normalize by m: comparable kernel values $\forall\ \lambda$

Combining directions



Diagonals

– Use the same scale λ and adapt the Markov chain

Combining directions

– Product space $\mathcal{H}_{L/R} \times \mathcal{H}_{U/D} \times \mathcal{H}_{DL/UR} \times \mathcal{H}_{DR/UL}$

$\Longrightarrow$ Norm is $d(x)^2 = d_{L/R}(x)^2 + d_{U/D}(x)^2 + d_{DL/UR}(x)^2 + d_{DR/UL}(x)^2$

for each pixel x, LRUD = Left Right Up Down, + diagonals

Easy extensions if need be

– Any angle θ, voxels / higher dimensions, anisotropy...

Result: an edge detector...

Original gray scale image

d(x) for each pixel x
with λ=1.5, κ=0.36 (for v=0..1)
white/black is min/max d(x)

…THAT CAN ANALYZE AT ≠ SCALES…

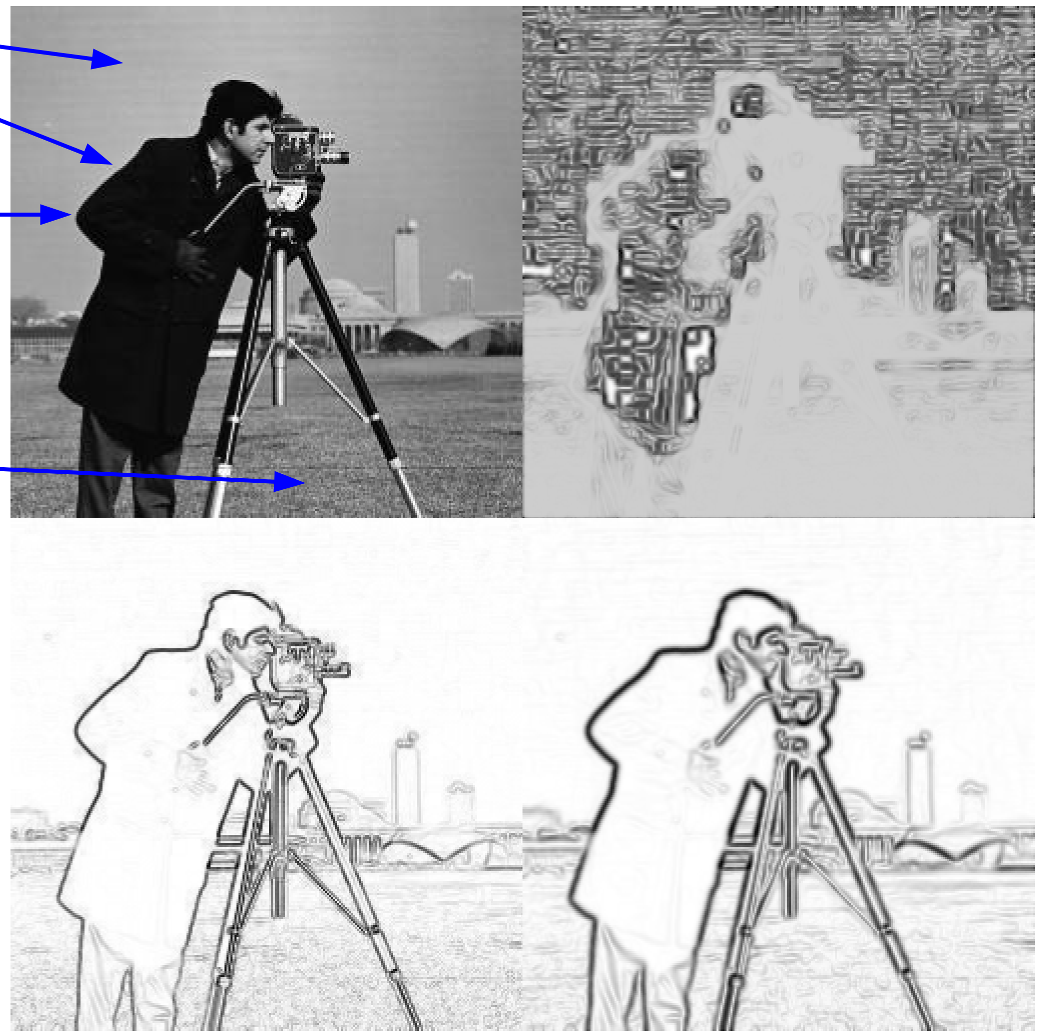
JPEG quantization artifacts

Large contrast differences: coat, tripod, handle

Fine texture with white / med-grey contrast

Edges at small λ:
- sharp, but some JPEG blocks still visible
- λ too small for the grass texture

Original | λ=3, κ=1/256



λ=1, κ=1 | λ=3, κ=1

Edges at small κ:
- sensitive to small gray diff.: JPEG artifacts, texture within the coat
- ignore med/large diff.: no grass, no coat edges, no tripod poles

Edges at large κ:
- Ignore JPEG artifacts
- Highlight large contrasts

Edges at medium λ:
- smoother
- grass texture < λ is matched

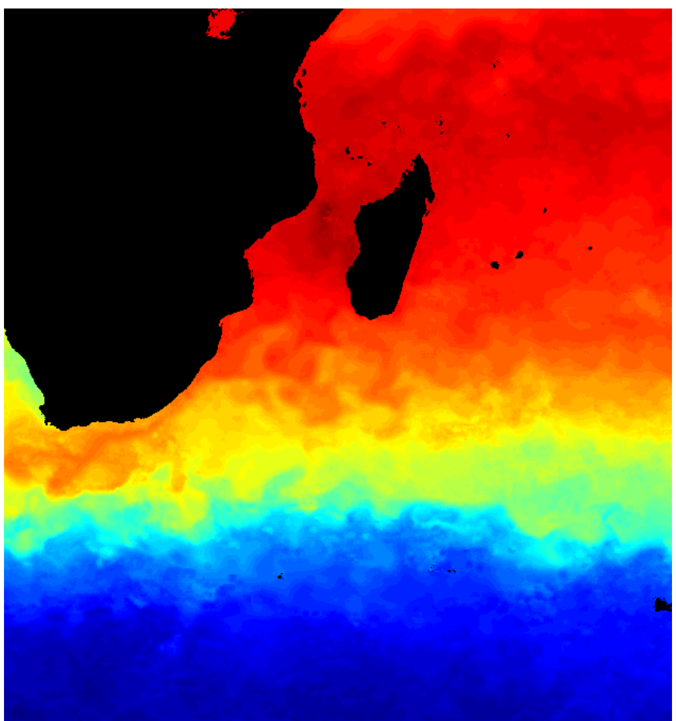
Application to remote sensing

Spatial (λ) and data (κ) scales are known a priori

– Small κ is not necessarily noise: e.g. large trends = sensor drift and small variations carry information

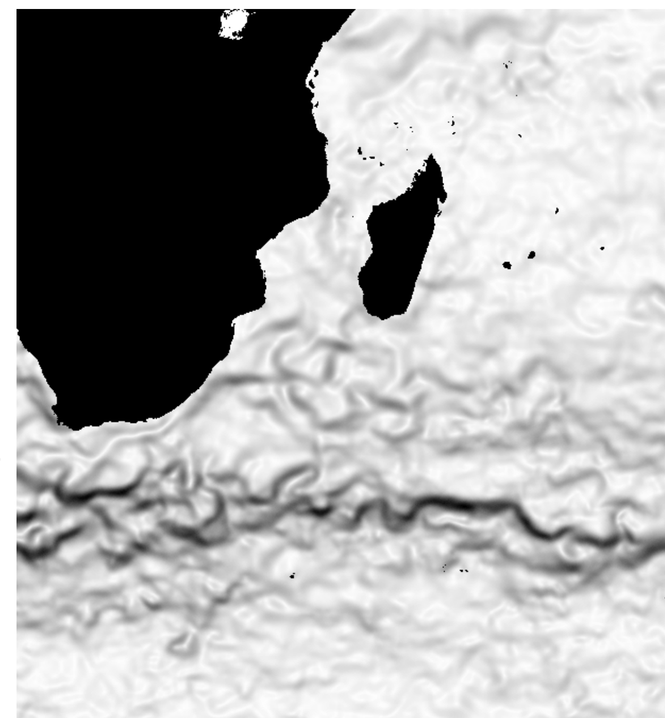
– Over- or sub- sampled signals: λ should match the characteristic physical scale, not the sampling rate

Example: Sea Surface Temperature



Left:
8-day composite MODIS data
blue = -1.2°C to red = 31.5°C
black = land masses (no data)

Right: analysis at:
$\lambda \approx 75$km (at center)
$\kappa \approx 1$°C
$\Longrightarrow$ typical oceanic current scales



DETECTING CHARACTERISTIC SCALES

Finding λ and κ without a priori information

- Analyse for a given pair of scales λ and κ
- Retain 20% of the most discriminative points
- Reconstruct the image from these points
- Compare with the original

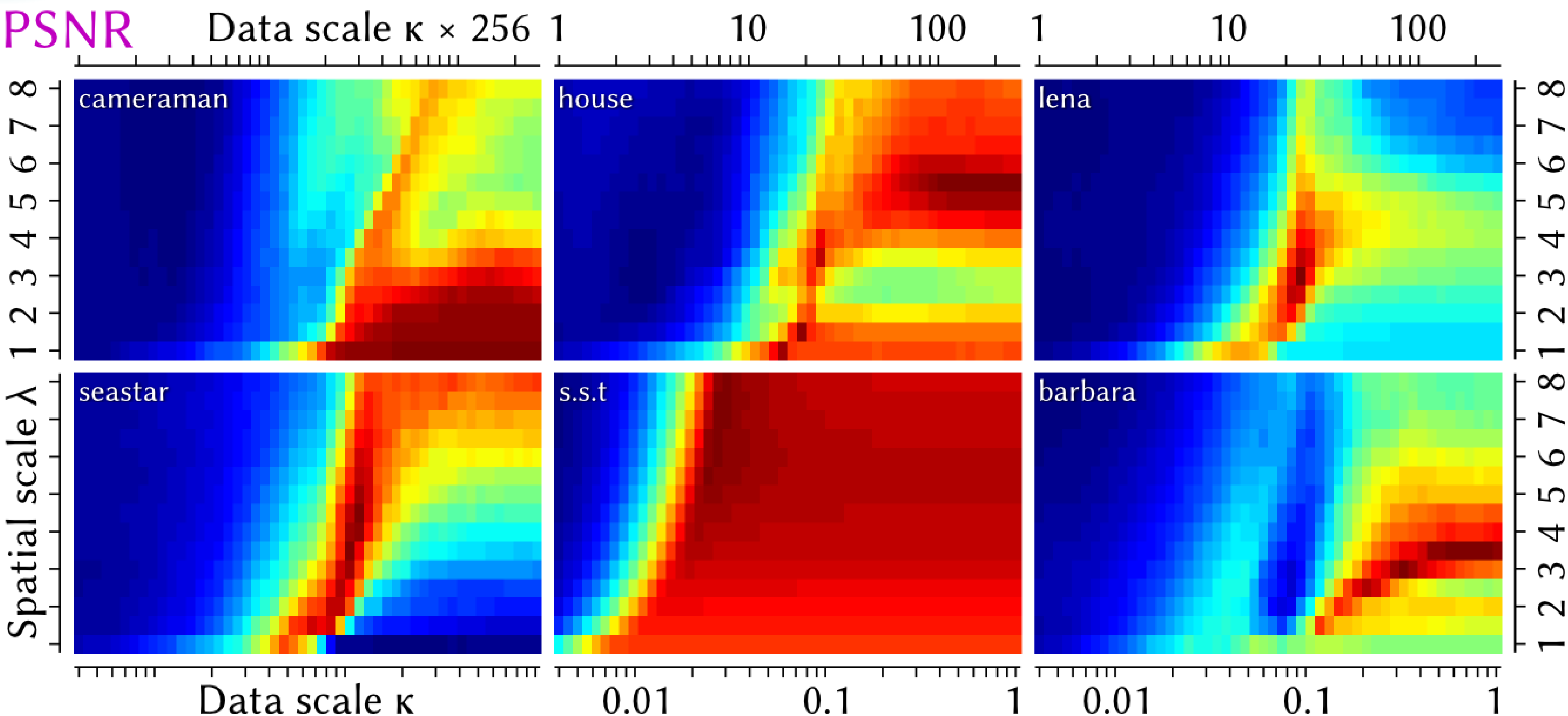
Hypothesis

- **IF** the points carry most of the information in the image.
  THEN the reconstruction will be “good”.

Reconstruction accuracy as a proxy for good λ, κ

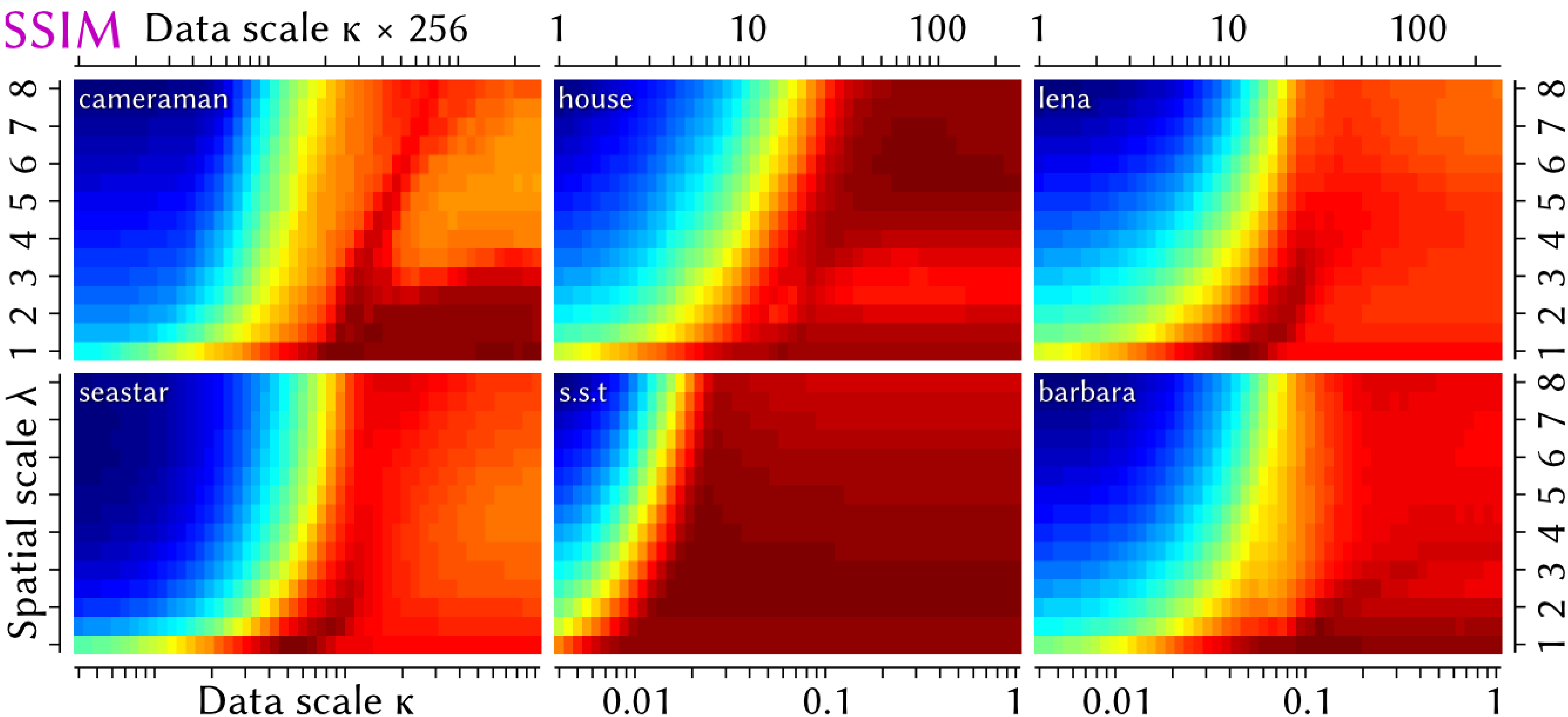
- Accuracy using the Peak Signal to Noise Ratio (PSNR)
- Accuracy using the Structural SIMilarity index (SSIM)

Multiscale analysis: PSNR



- Definite zones of high accuracy ⟹ best λ, κ
- Local maxima (cameraman, house) ⟹ objects in image have ≠ properties
- Zones at low κ (s.s.t, house) have high PSNR but match noise:
  house: the brick texture, sea surface temperature (s.s.t): noise at 0.1°C

Multiscale analysis: SSIM



– Same general transitions => irrelevant λ, κ below

– Zones of local maxima SSIM ≠ PSNR

$\Longrightarrow$ Pareto front for best λ, κ. Common maxima = best λ, κ ?

– Something special for Barbara at λ≈2.5 and large κ, in both PSNR / SSIM

BARBARA, λ≈2.5, LARGE GRADIENT?

Quite obvious!

but a nice validation of the method

Reconstruction from 20% points

λ=1.5, κ=0.12, PSNR=17.6, SSIM=0.73

λ=3.5, κ=0.64, PSNR=17.7, SSIM=0.67

Texture edges preserved, details $\lesssim \lambda$ smoothed out!

Colored Textures

The method is valid for **any** kernel acting on V^m

– Especially vector data: Color spaces, hyperspectral, etc.

For RGB triplets $v = \{r,g,b\} \in V$

– Conversion to Lab space with D65 white point: $\ell(v) \in \mathcal{L}$

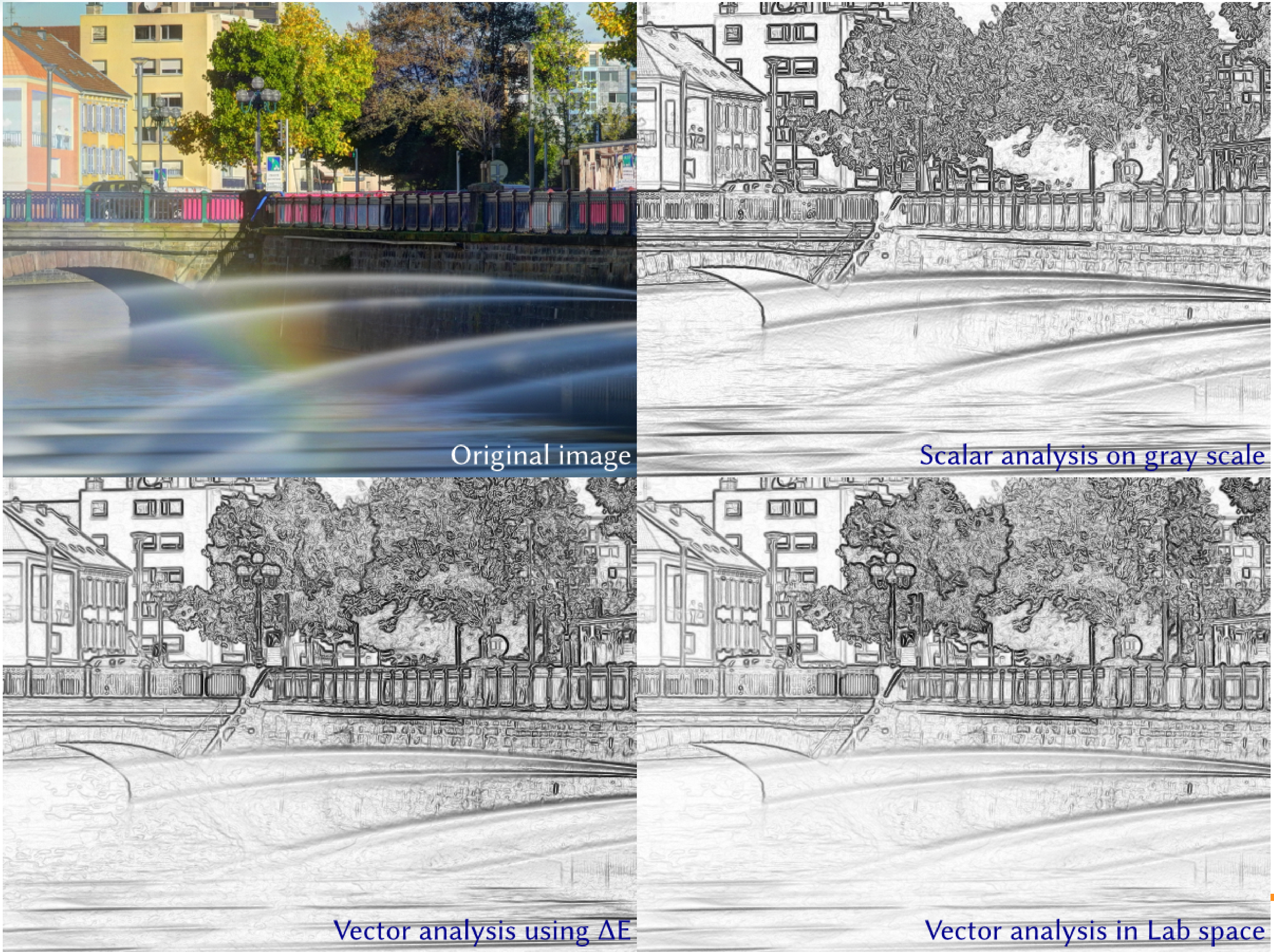
– Using one of the two operators:

- $\delta_1(v,w) = \|\ell(v)-\ell(w)\|_{\mathcal{L}}$: Lab is perceptually uniform $\Rightarrow$ norm in Lab is presumably a sensitivity to color difference
- $\delta_2(v,w) = \Delta E(\ell(v)-\ell(w))$: CIE DE 2000 updated formula for a better perceptual uniformity

– Then apply an updated kernel

$$k_{1,2}(s,t) = 1/(1+\tfrac{1}{m}\Sigma(\delta_{1,2}(v,w)/\kappa)^2\ )$$

Color Results



Summary

Multiscale Image Analysis

– Find or use the characteristic spatial (λ) and data value (κ) scales within the image

With Stochastic Texture Differences

– Statistical description of the texture

– Norm of a difference (in Hilbert space)

$\Longrightarrow$ behaves like a norm of a gradient (=difference)

… but for textures!

And for vector data

– Shown with RGB, but anything with a kernel is OK

N. Brodu, H. Yahia, “Multiscale image analysis with stochastic texture differences”, to appear online in 2015.