Visual Computing:

Selected topics:

Sparsity and Texture

Prof. Marc Pollefeys

ETH

inf | Informatik
Computer Science

Image denoising

$$\underbrace{\mathbf{y}}_{\text{measurements}} = \underbrace{\mathbf{x}_{orig}}_{\text{original image}} + \underbrace{\mathbf{w}}_{\text{noise}}$$

Slides from Julien Mairal

Sparse representations for image restoration

$$\underbrace{\mathbf{y}}_{\text{measurements}} = \underbrace{\mathbf{x}_{orig}}_{\text{original image}} + \underbrace{\mathbf{w}}_{\text{noise}}$$

Energy minimization problem - MAP estimation

$$E(\mathbf{x}) = \underbrace{\frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2}_{\text{relation to measurements}} + \underbrace{Pr(\mathbf{x})}_{\text{image model (-log prior)}}$$

Some classical priors

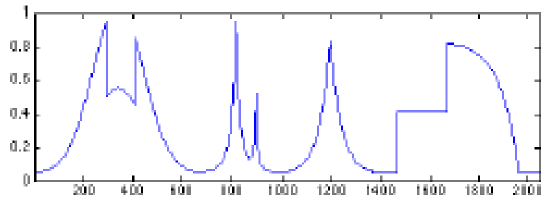
- Smoothness $\lambda\|\mathcal{L}\mathbf{x}\|_2^2$
- Total variation $\lambda\|\nabla\mathbf{x}\|_1^2$
- MRF priors
- ...

ETH

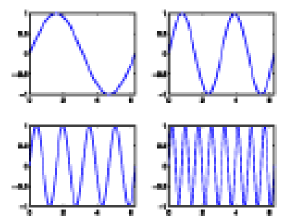
Slides from Julien Mairal

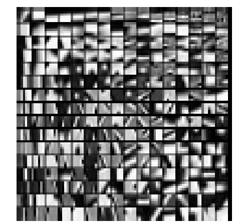
What is a sparse linear model

Let $\mathbf{x}$ in $\mathbb{R}^m$ be a signal.



Let $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_p] \in \mathbb{R}^{m \times p}$ be a set of normalized "basis vectors".
We call it **dictionary**.





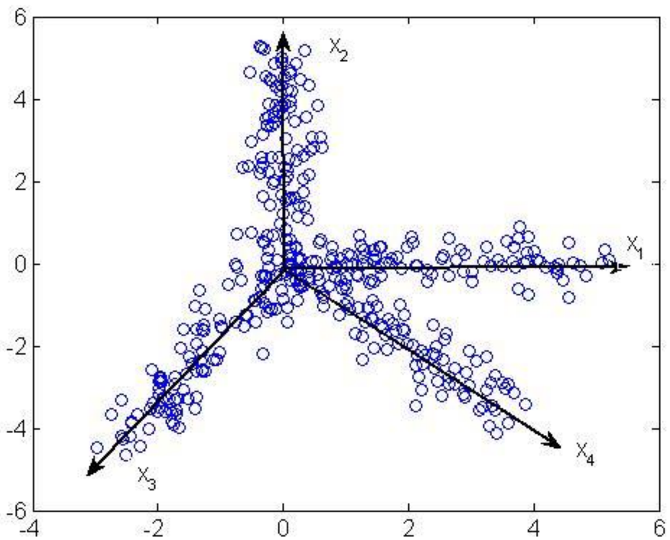
$\mathbf{D}$ is "adapted" to $\mathbf{x}$ if it can represent it with a few basis vectors—that is, there exists a **sparse vector** α in $\mathbb{R}^p$ such that $\mathbf{x} \approx \mathbf{D}\alpha$. We call α the **sparse code**.

$$\underbrace{\begin{pmatrix} \mathbf{x} \end{pmatrix}}_{\mathbf{x} \in \mathbb{R}^m} \approx \underbrace{\begin{pmatrix} \mathbf{d}_1 & \big| & \mathbf{d}_2 & \big| & \cdots & \big| & \mathbf{d}_p \end{pmatrix}}_{\mathbf{D} \in \mathbb{R}^{m \times p}} \underbrace{\begin{pmatrix} \alpha[1] \\ \alpha[2] \\ \vdots \\ \alpha[p] \end{pmatrix}}_{\alpha \in \mathbb{R}^p, \textbf{sparse}}$$

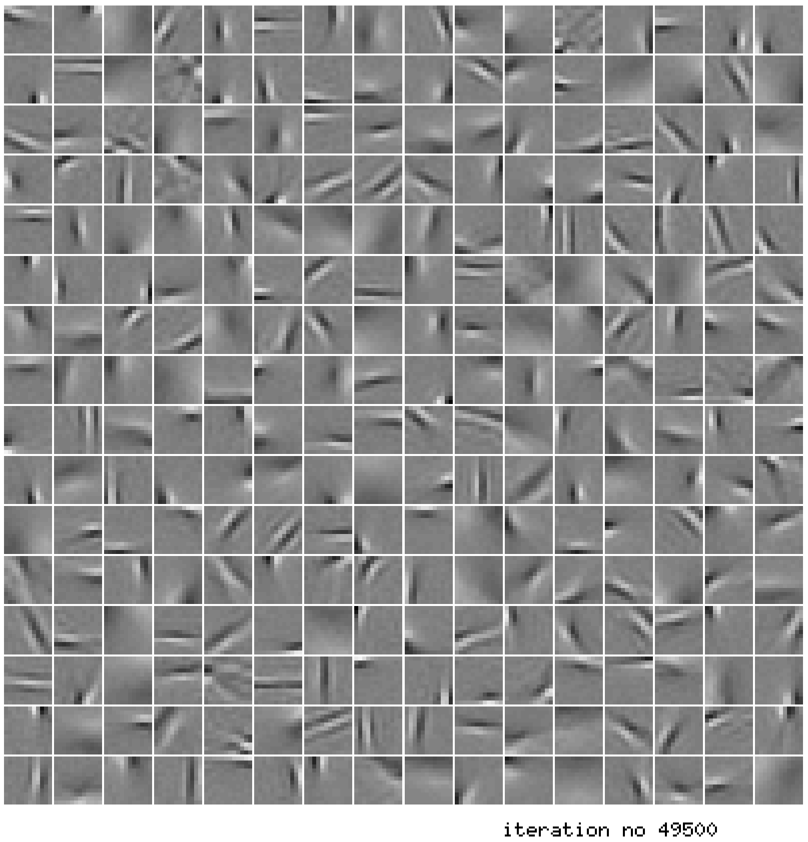
Slides from
Julien Mairal

Overcomplete Sparse Dictionary

Simple 2D example
- Complete basis has 2 elements
- Overcomplete Sparse Dictionary with 4 elements allows sparse representation



Learned example (Ranzato NIPS2006)



Important idea

Why Sparsity?

A dictionary can be good for representing a class of signals, but not for representing white Gaussian noise.

Sparse Decomposition problem

$$\min_{\alpha \in \mathbb{R}^p} \underbrace{\frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2}_{\text{data fitting term}} + \underbrace{\lambda\psi(\boldsymbol{\alpha})}_{\text{sparsity-inducing regularization}}$$

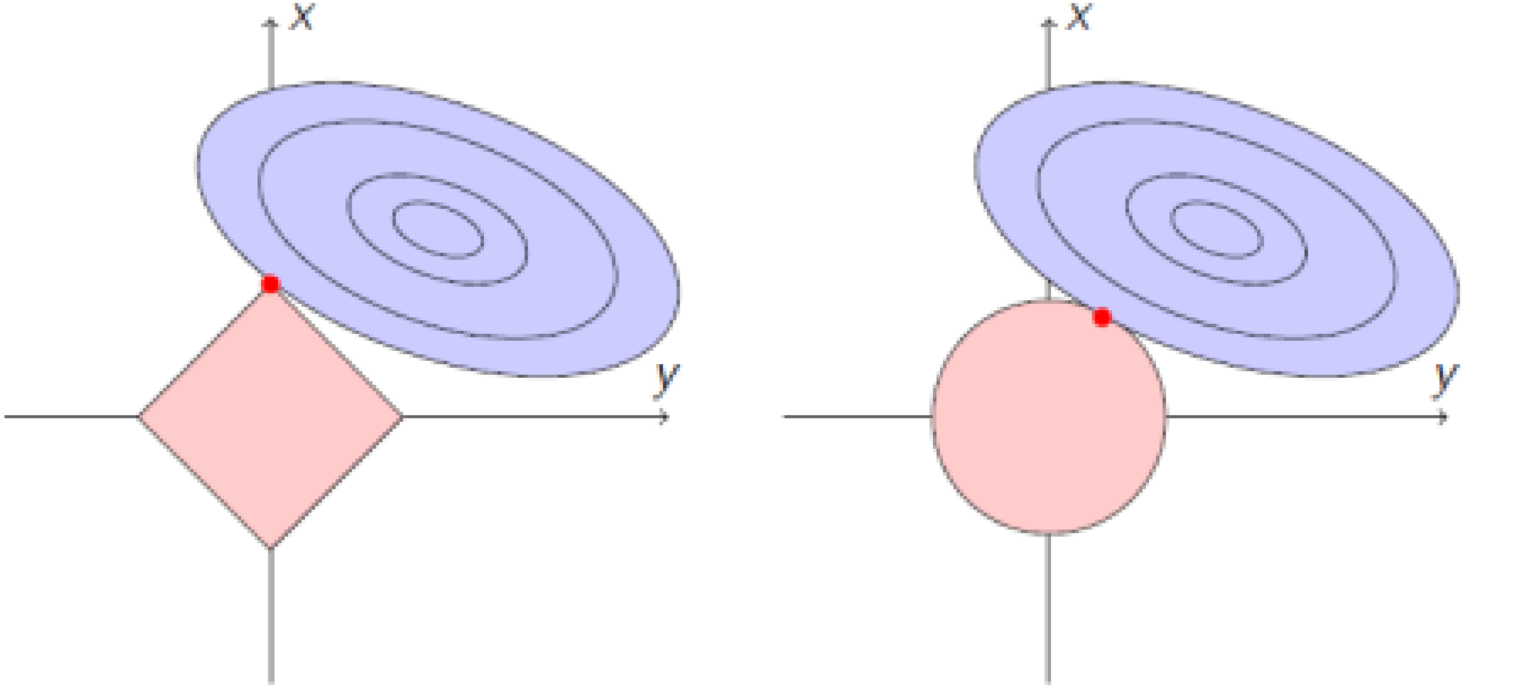
ψ induces sparsity in $\boldsymbol{\alpha}$. It can be

- the ℓ_0 "pseudo-norm". $\|\boldsymbol{\alpha}\|_0 \triangleq \#\{i \text{ s.t. } \boldsymbol{\alpha}[i] \neq 0\}$ (NP-hard)
- the ℓ_1 norm. $\|\boldsymbol{\alpha}\|_1 \triangleq \sum_{i=1}^{p} |\boldsymbol{\alpha}[i]|$ (convex),
- ...

This is a selection problem. When ψ is the ℓ_1-norm, the problem is called Lasso [Tibshirani, 1996] or basis pursuit [Chen et al., 1999]

Why does ℓ_1 induce sparsity

Geometric explanation



$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1$$

$$\min_{\alpha \in \mathbb{R}^p} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 \;\; \text{s.t.} \;\; \|\alpha\|_1 \leq T.$$

Sparse representations for image restoration

Designed dictionaries

[Haar, 1910], [Zweig, Morlet, Grossman ~70s], [Meyer, Mallat, Daubechies, Coifman, Donoho, Candes ~80s-today]. . . (see [Mallat, 1999])

Wavelets, Curvelets, Wedgelets, Bandlets, . . . lets

Learned dictionaries of patches

[Olshausen and Field, 1997], [Engan et al., 1999], [Lewicki and Sejnowski, 2000], [Aharon et al., 2006] , [Roth and Black, 2005], [Lee et al., 2007]

$$\min_{\alpha_i, \mathbf{D}\in\mathcal{C}} \sum_i \underbrace{\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2}_{\text{reconstruction}} + \underbrace{\lambda\psi(\alpha_i)}_{\text{sparsity}}$$

- $\psi(\alpha) = \|\alpha\|_0$ (“ℓ_0 pseudo-norm”)
- $\psi(\alpha) = \|\alpha\|_1$ (ℓ_1 norm)

ETH

Slides from Julien Mairal

Sparse representations for image restoration

Solving the denoising problem

[Elad and Aharon, 2006]

- Extract all overlapping 8×8 patches $\mathbf{x}_i$.
- Solve a matrix factorization problem:

$$\min_{\alpha_i, \mathbf{D} \in \mathcal{C}} \sum_{i=1}^{n} \underbrace{\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2}_{\text{reconstruction}} + \underbrace{\lambda\psi(\boldsymbol{\alpha}_i)}_{\text{sparsity}},$$

with $n > 100,000$

- Average the reconstruction of each patch.

Slides from Julien Mairal

Sparse representations for image restoration

K-SVD: [Elad and Aharon, 2006]

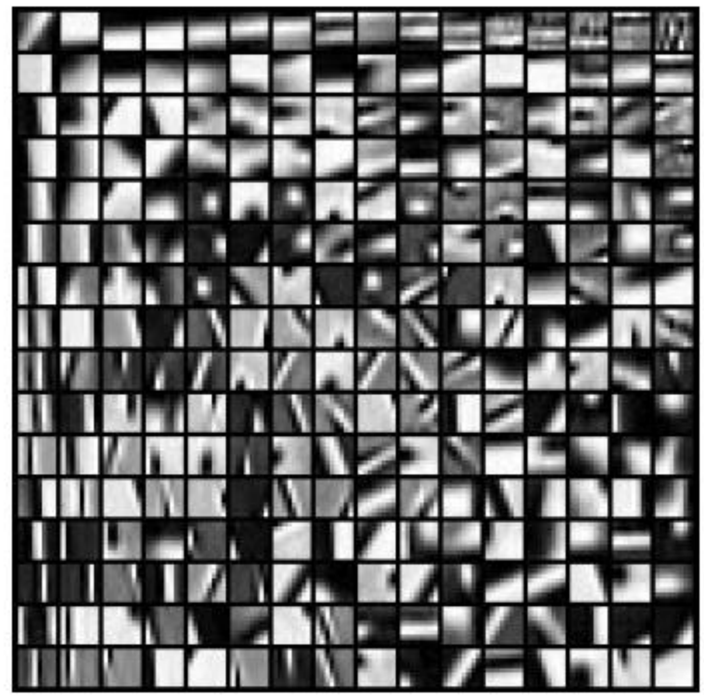


Figure: Dictionary trained on a noisy version of the image boat.

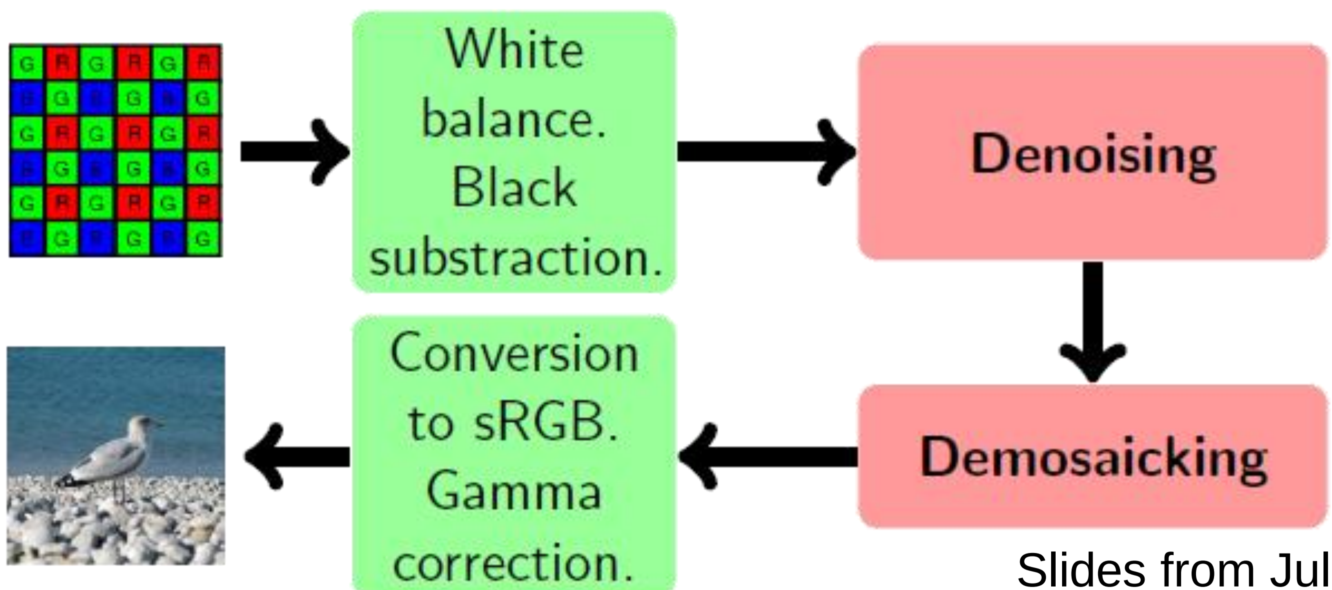
Slides from Julien Mairal

Sparse representations for image restoration

Inpainting, Demosaicking

$$\min_{\mathbf{D}\in\mathcal{C},\alpha} \sum_i \frac{1}{2}\|\beta_i \otimes (\mathbf{x}_i - \mathbf{D}\alpha_i)\|_2^2 + \lambda_i \psi(\alpha_i)$$

RAW Image Processing

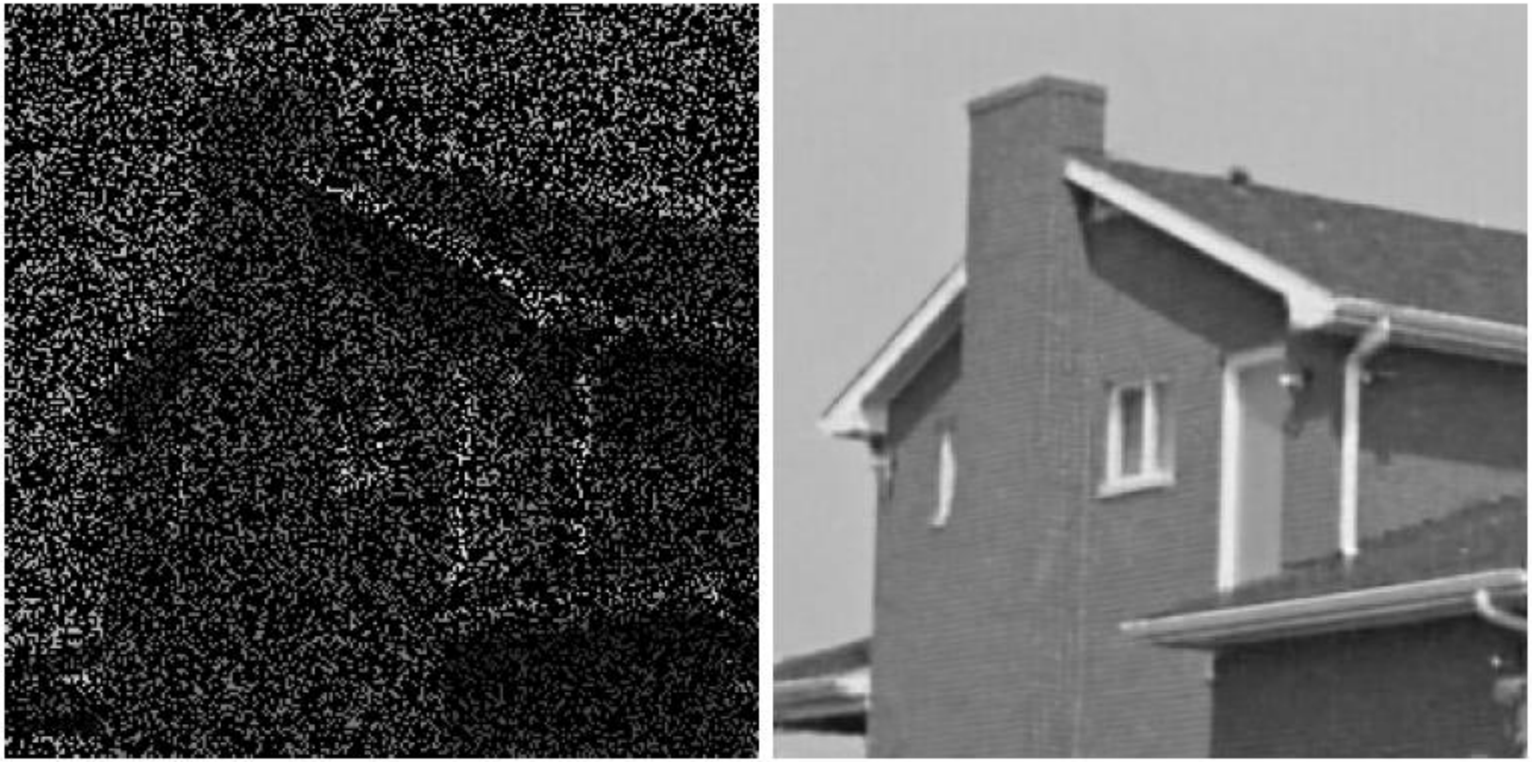


ETH

Slides from Julien Mairal

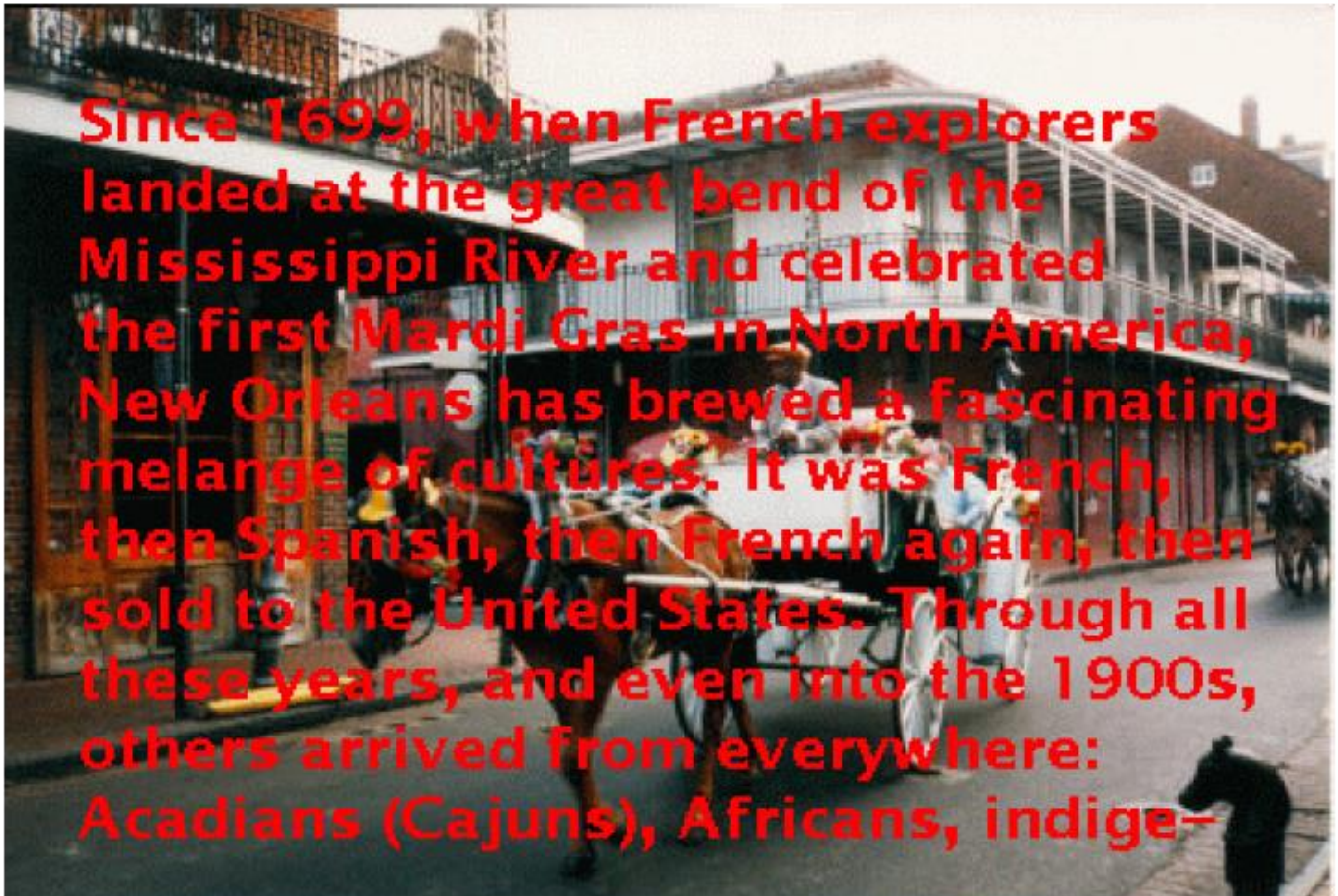
[Mairal, Bach, Ponce, Sapiro, and Zisserman, 2009b]

[Mairal, Sapiro, and Elad, 2008d]



Slides from Julien Mairal

Inpainting, [Mairal, Elad, and Sapiro, 2008b]



Inpainting, [Mairal, Elad, and Sapiro, 2008b]

Slides from Julien Mairal

Key ideas for video processing

[Protter and Elad, 2009]

- Using a 3D dictionary.
- Processing of many frames at the same time.
- Dictionary propagation.

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

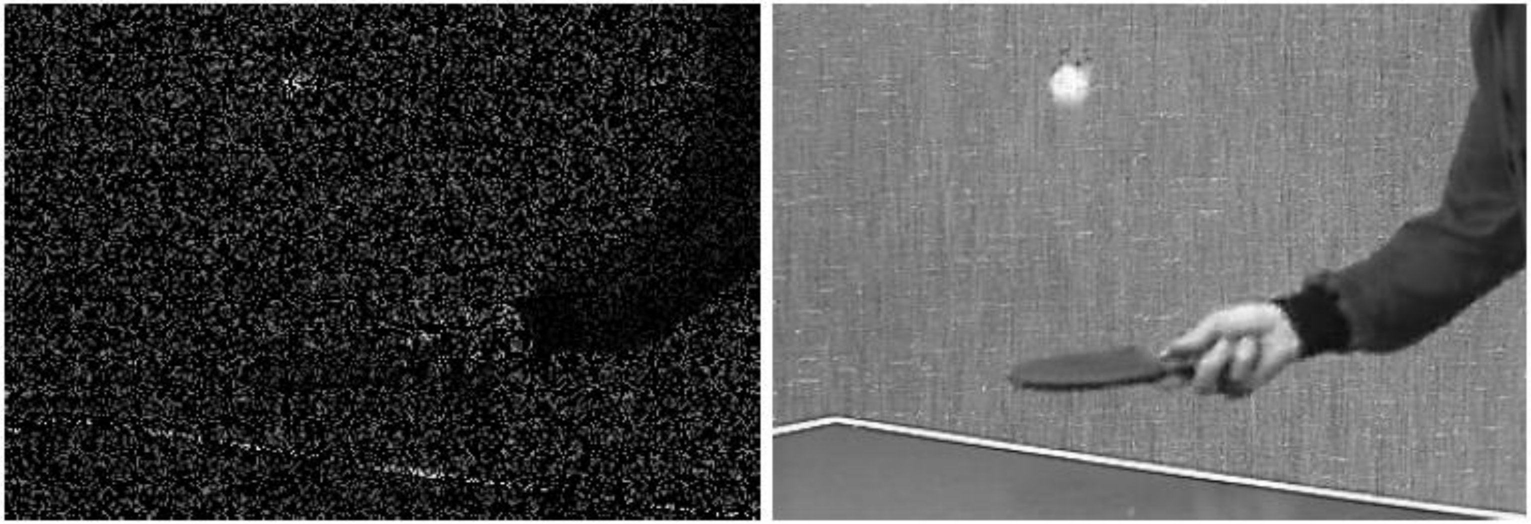


Figure: Inpainting results.

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

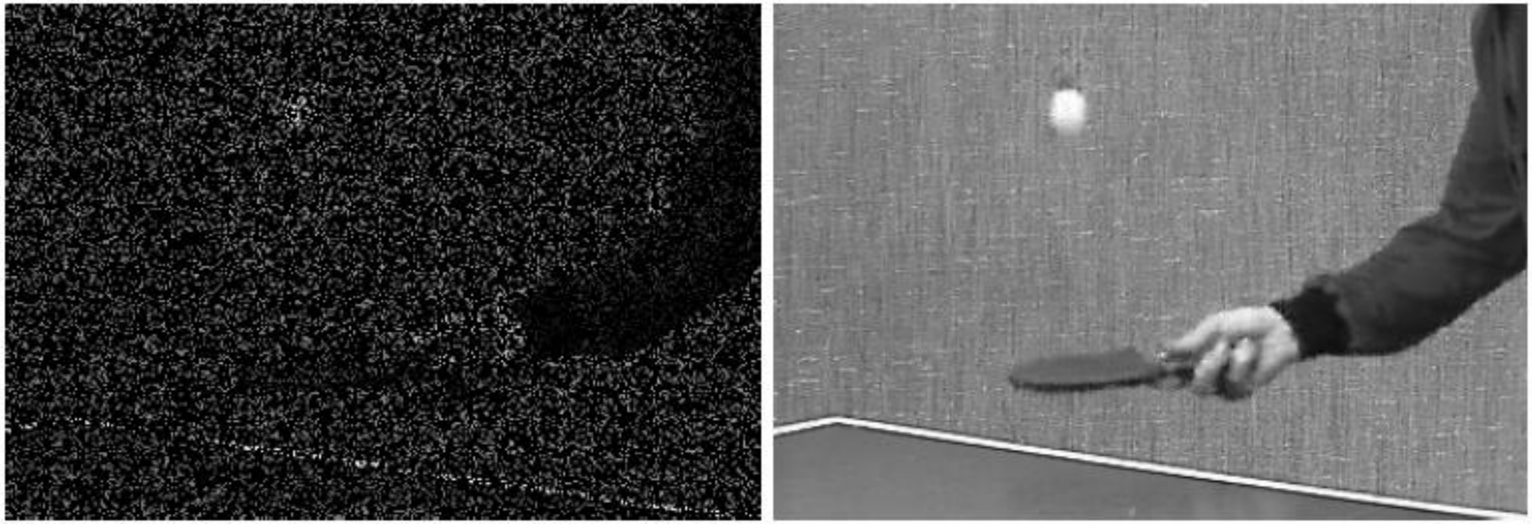


Figure: Inpainting results.

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

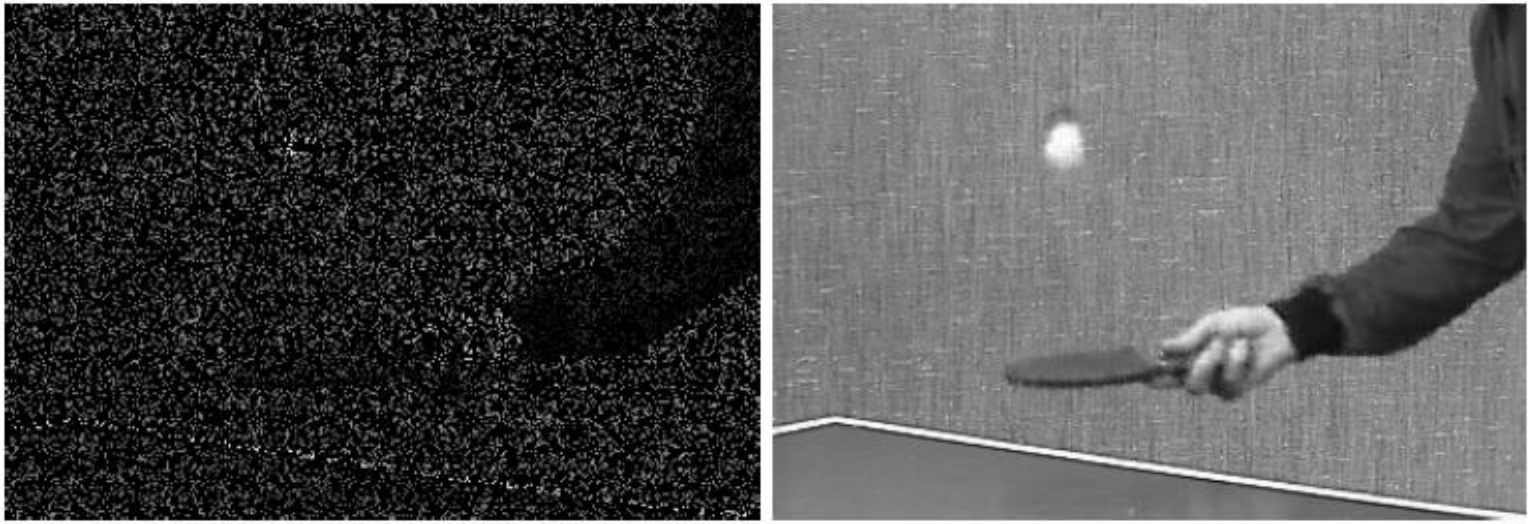


Figure: Inpainting results.

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

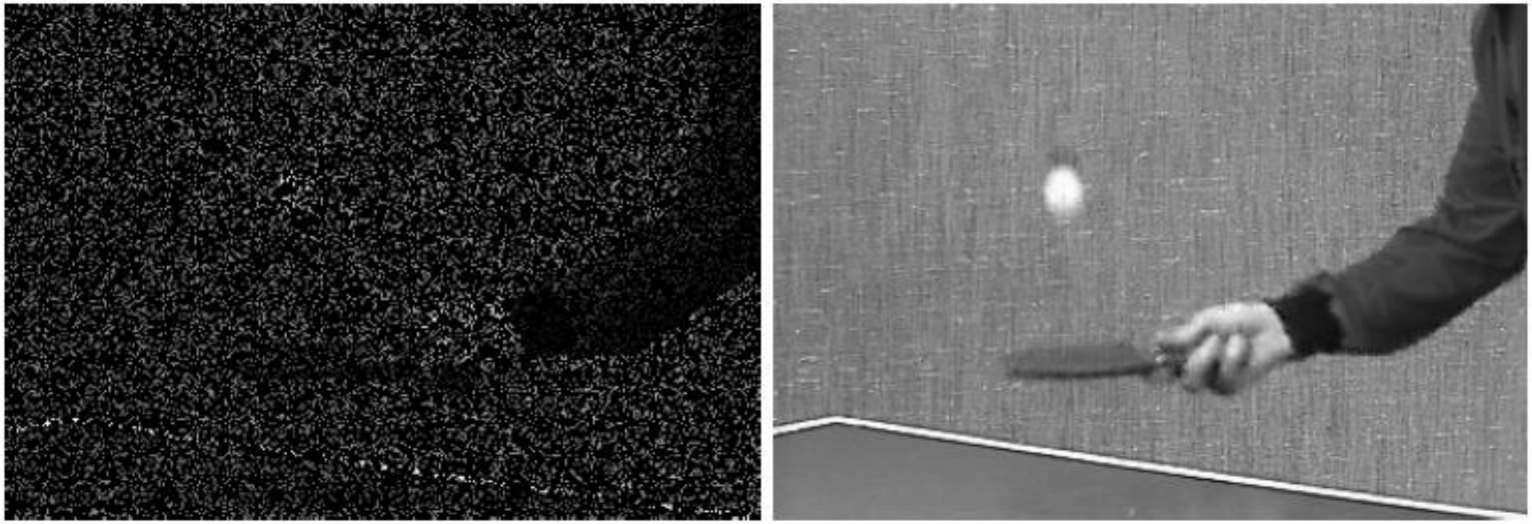


Figure: Inpainting results.

Inpainting, [Mairal, Sapiro, and Elad, 2008d]

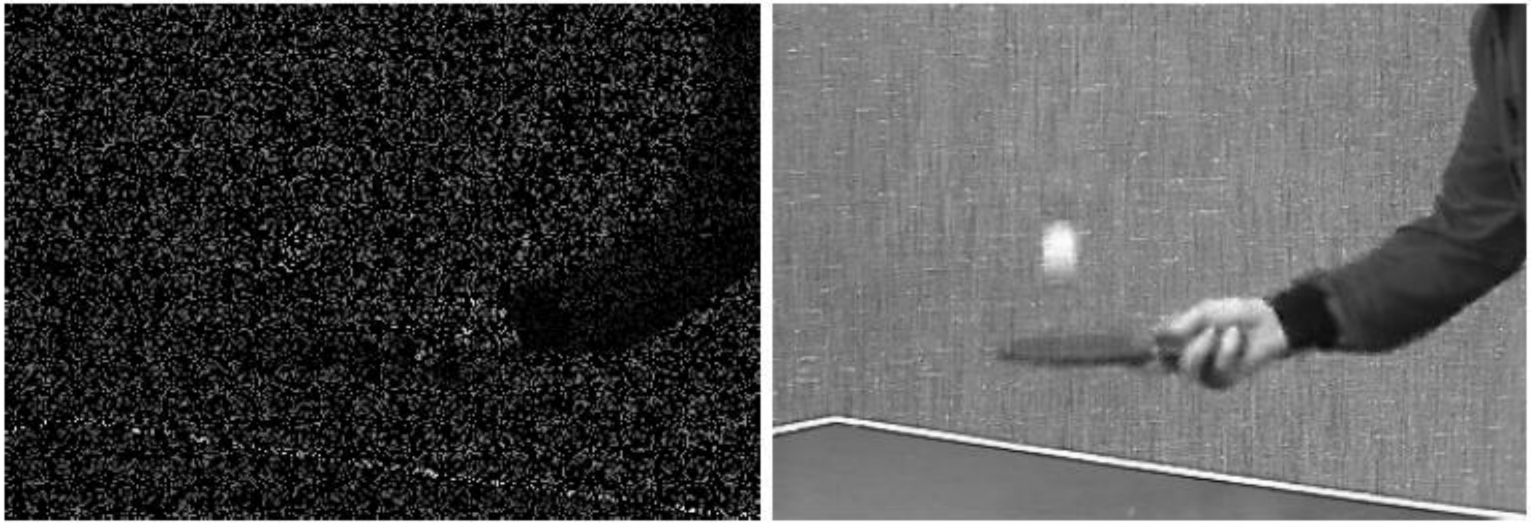


Figure: Inpainting results.

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

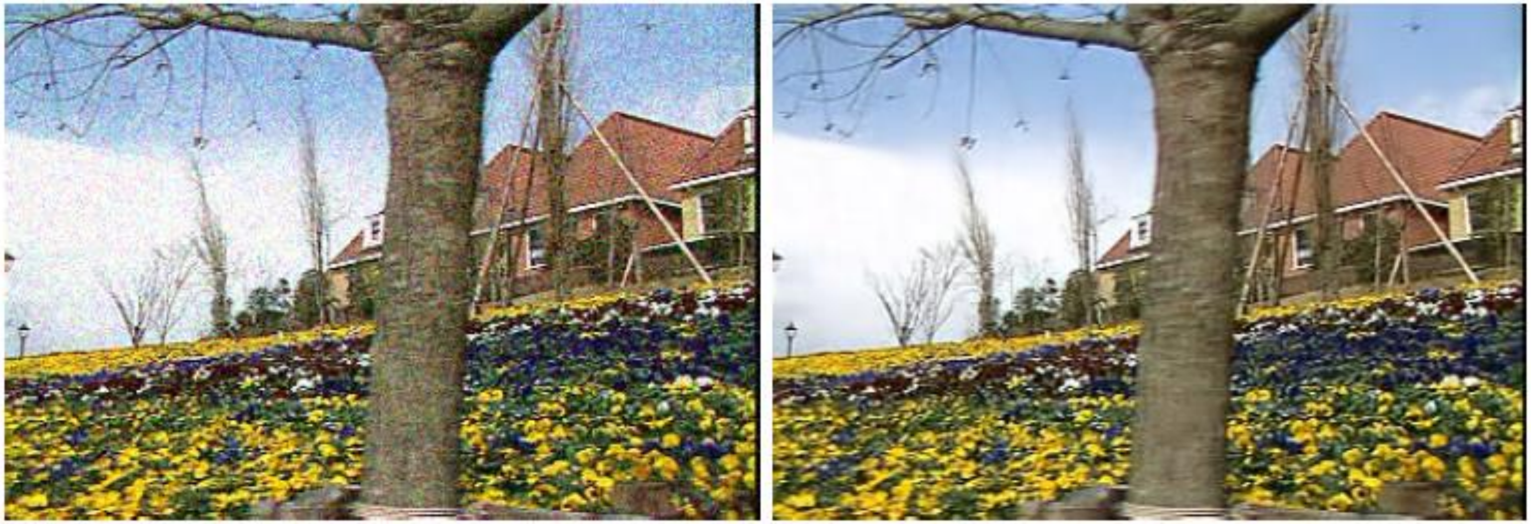


Figure: Denoising results. $\sigma = 25$

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

Color video denoising, [Mairal, Sapiro, and Elad, 2008d]

Figure: Denoising results. $\sigma = 25$

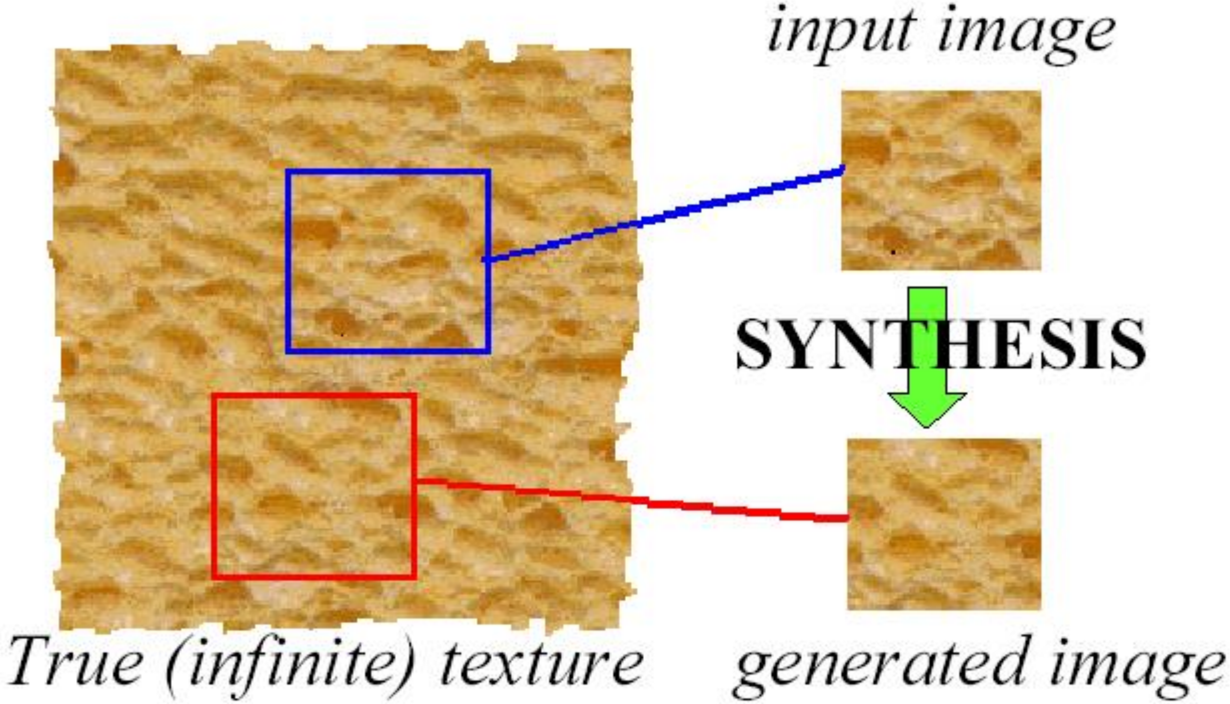
Texture

- Key issue: representing texture
  – Texture analysis/segmentation
    - key issue: representing texture
  – Texture synthesis
    - useful; also gives some insight into quality of representation
  – Shape from texture

link

Computer graphics: texture mapping

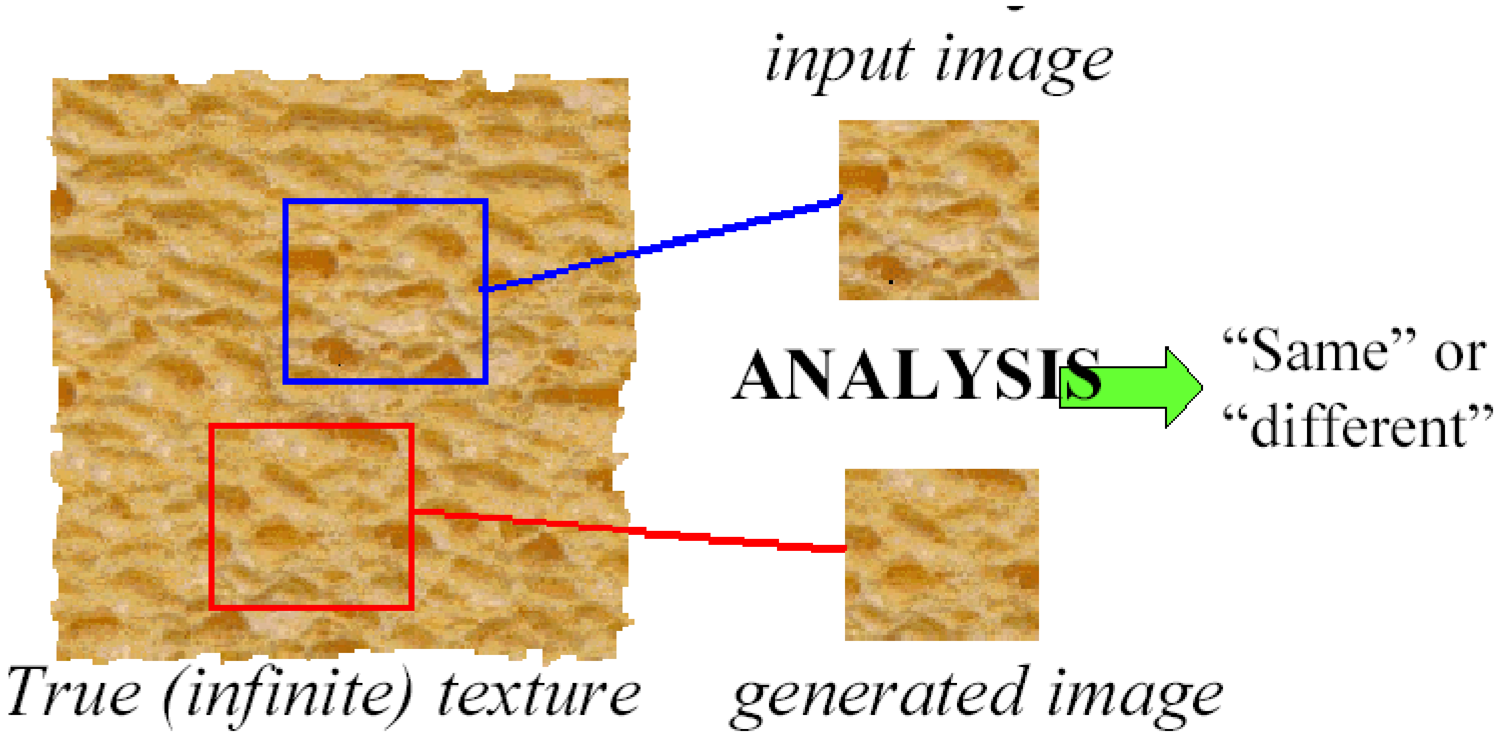
Texture synthesis



Given example, generate texture sample

(that is large enough, satisfies constraints, …)

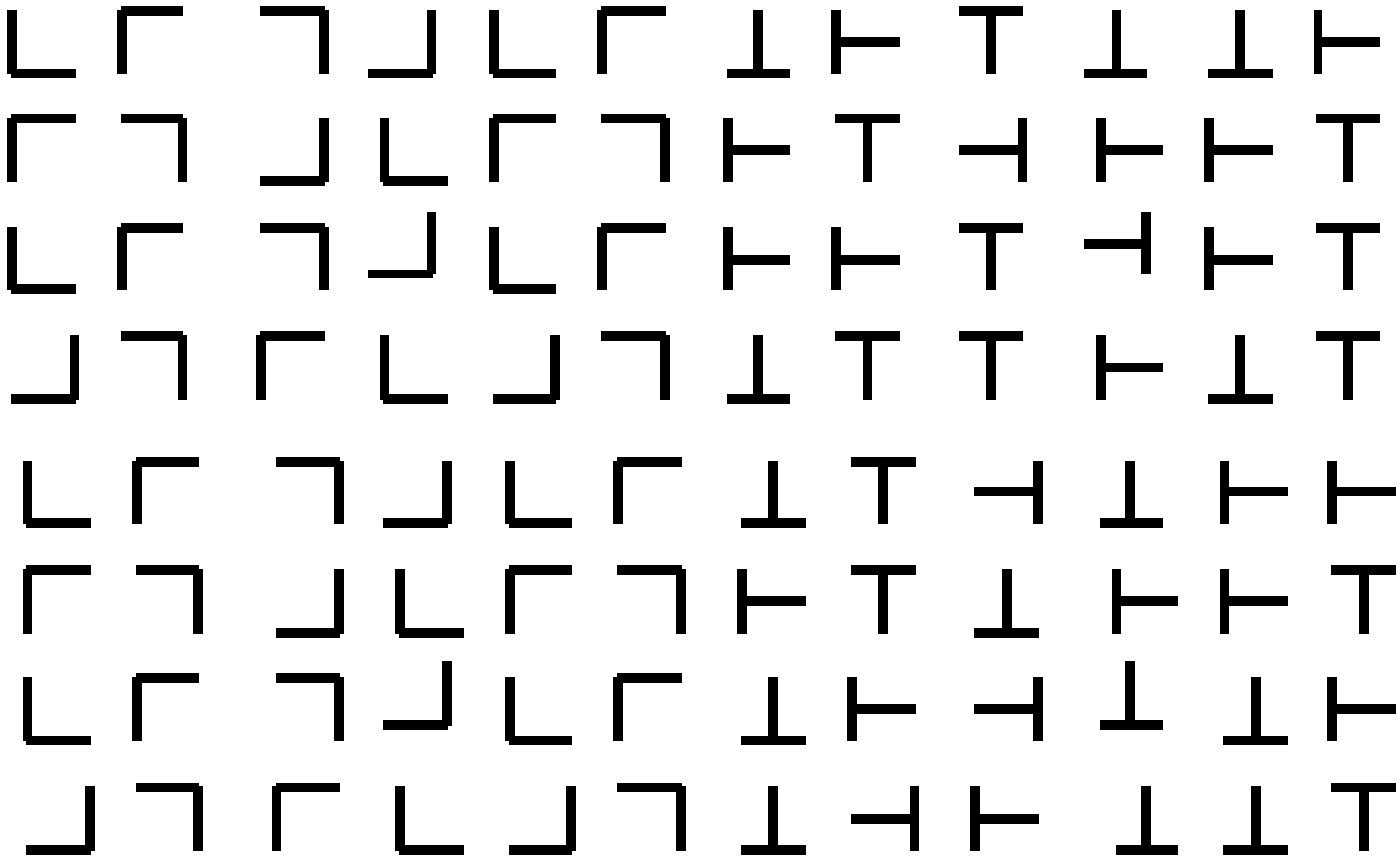
Texture analysis



Compare; is this the same "stuff"?

pre-attentive texture discrimination

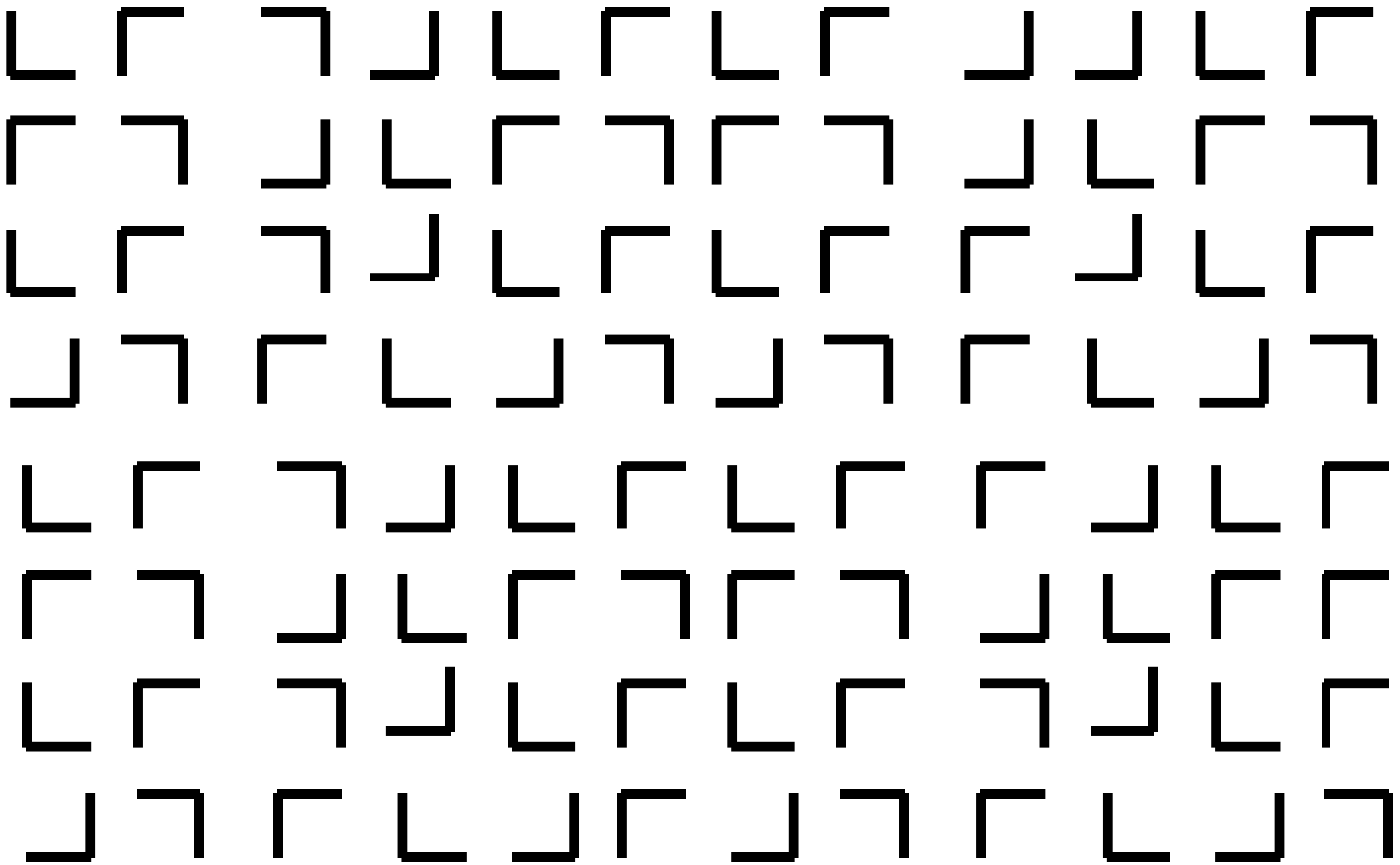
pre-attentive texture discrimination



pre-attentive texture discrimination

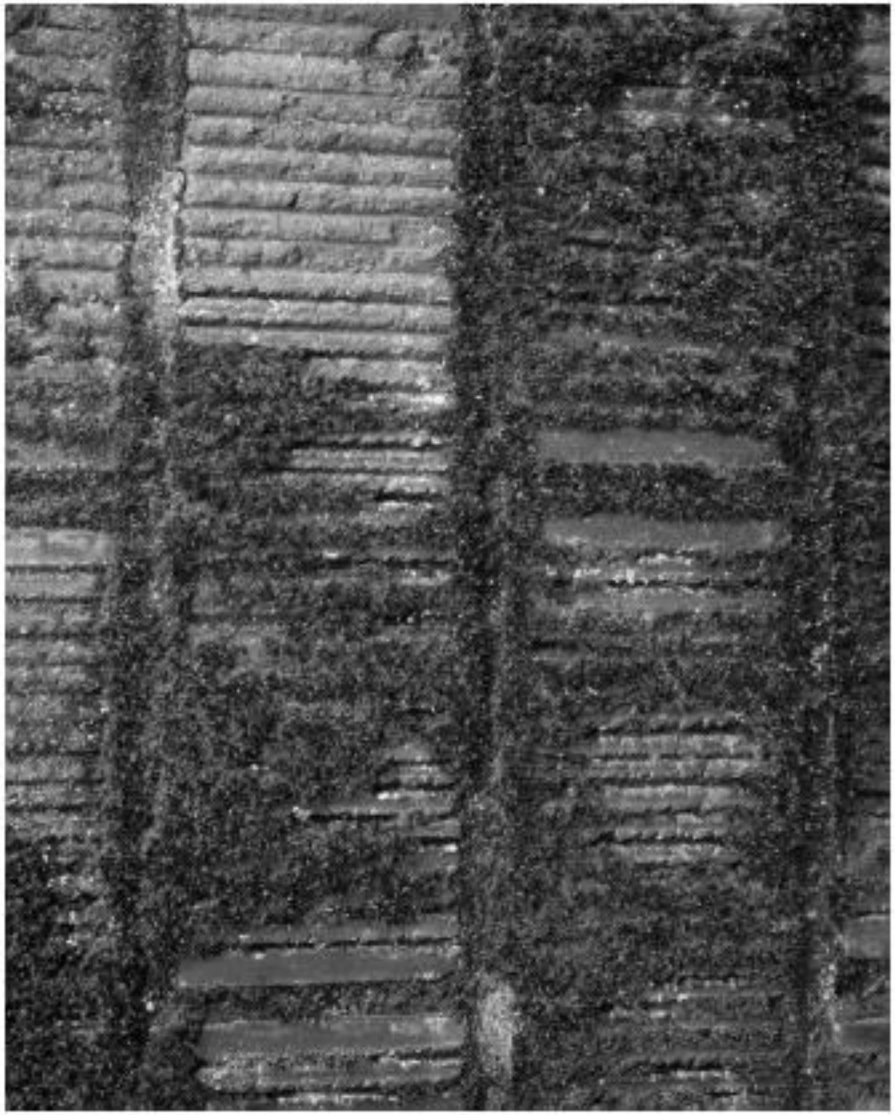
- same or not?

pre-attentive texture discrimination



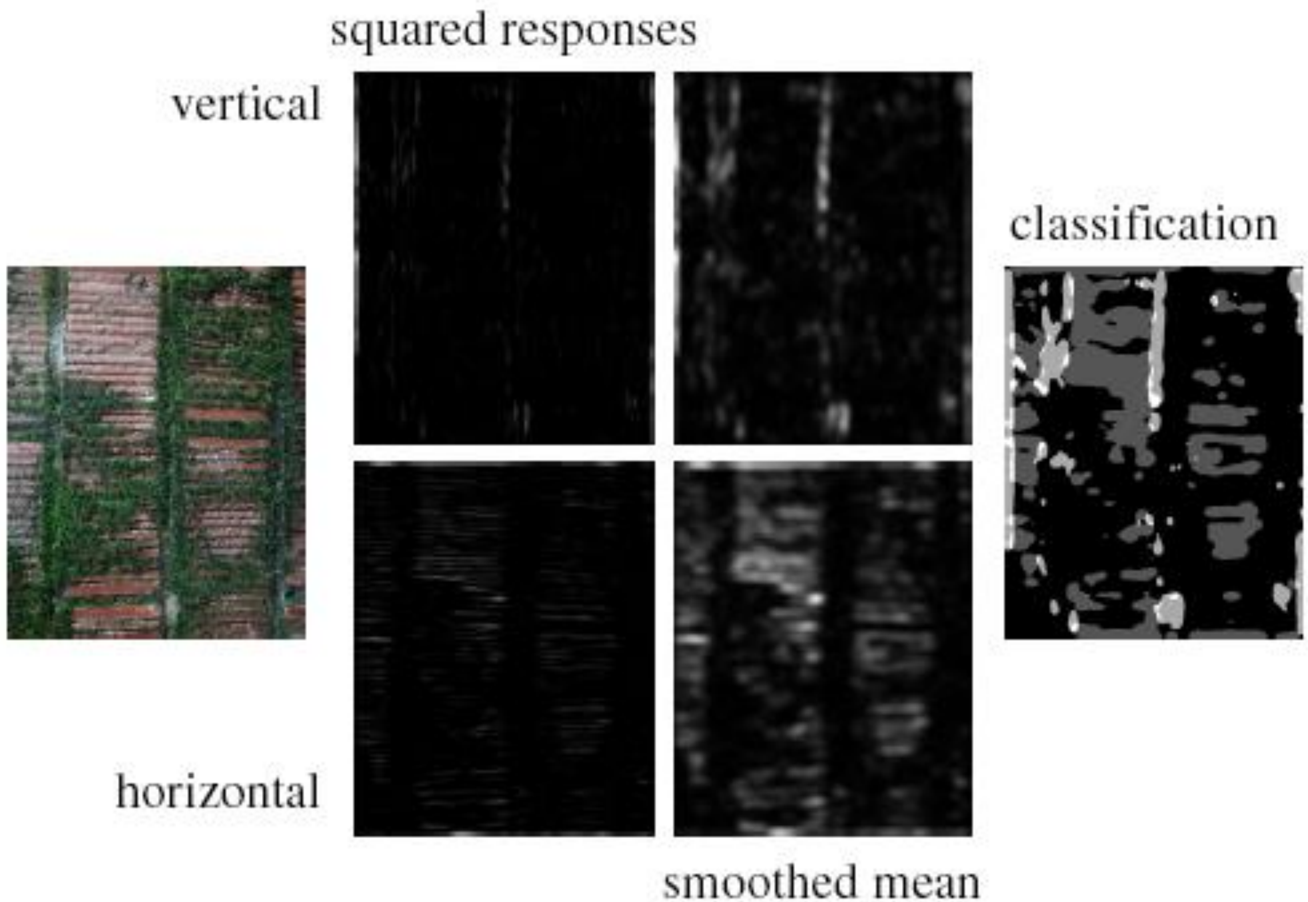
pre-attentive texture discrimination

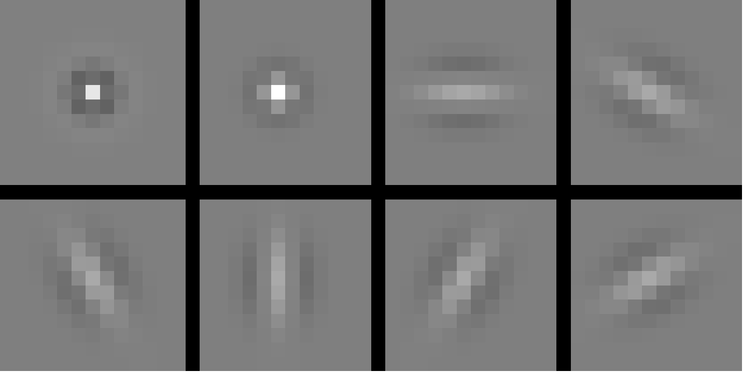
- same or not?

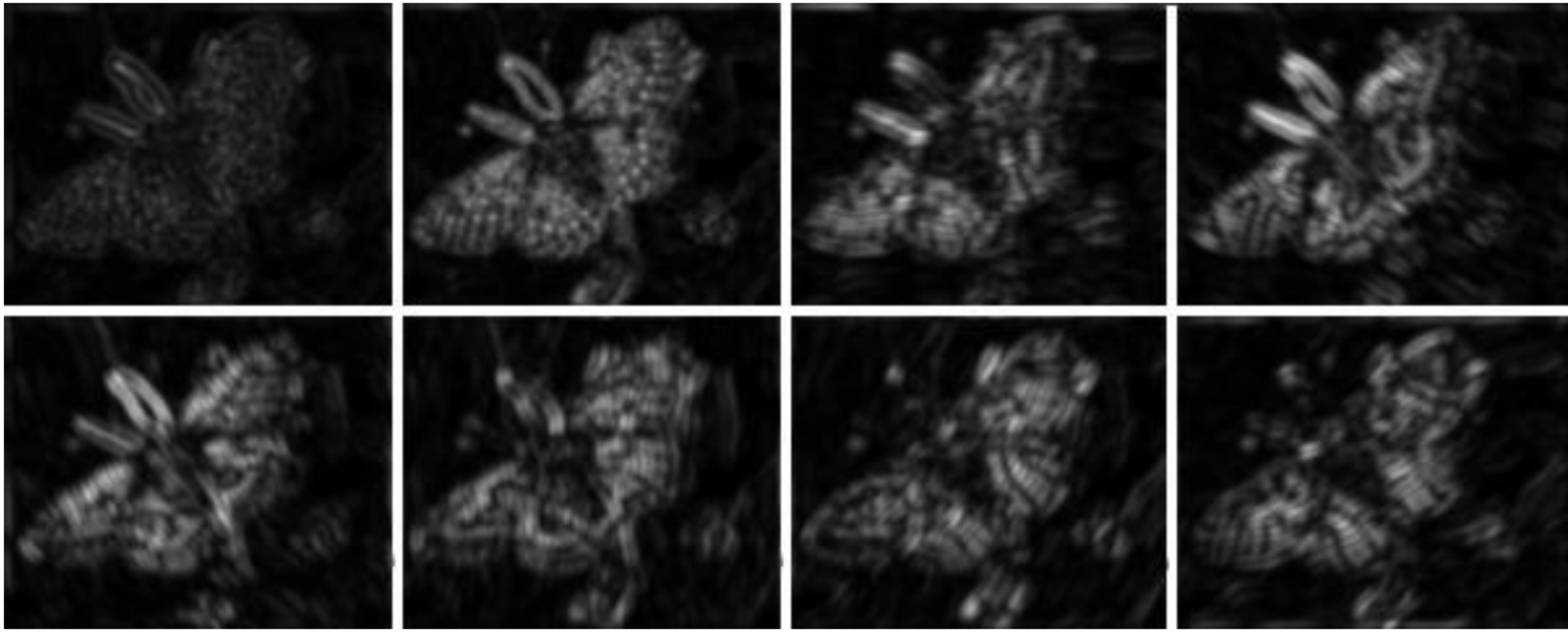


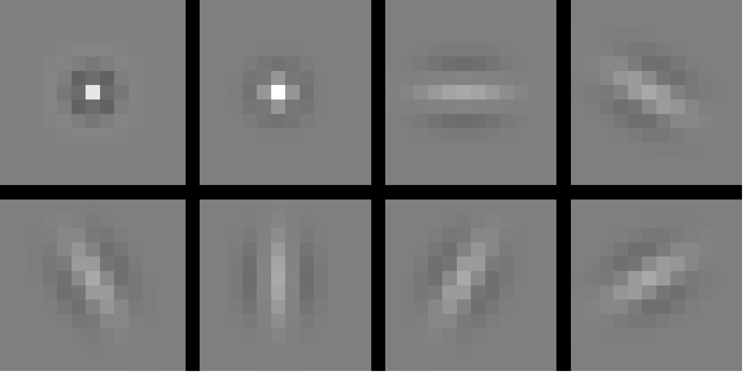
Representing textures

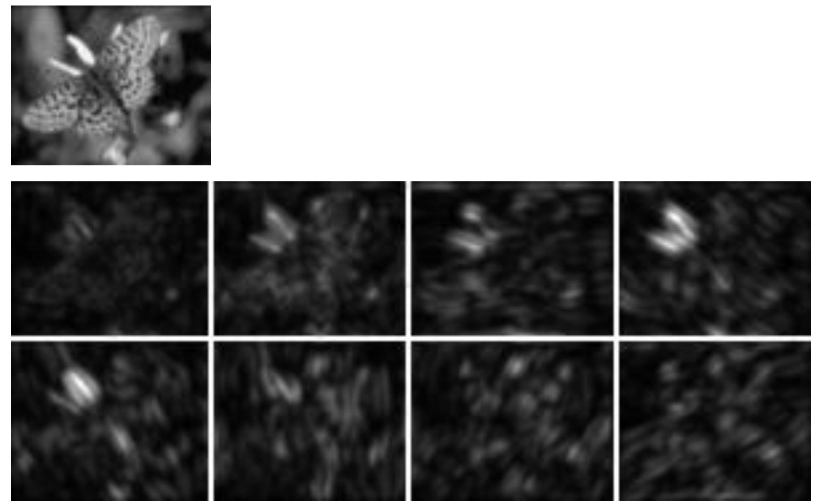
- Textures are made up of quite stylised subelements, repeated in meaningful ways
- Representation:
  - find the subelements, and represent their statistics
- But what are the subelements, and how do we find them?
  - recall normalized correlation
  - find subelements by applying filters, looking at the magnitude of the response
- What filters?
  - experience suggests spots and oriented bars at a variety of different scales
  - details probably don't matter
- What statistics?
  - within reason, the more the merrier.
  - At least, mean and standard deviation
  - better, various conditional histograms.

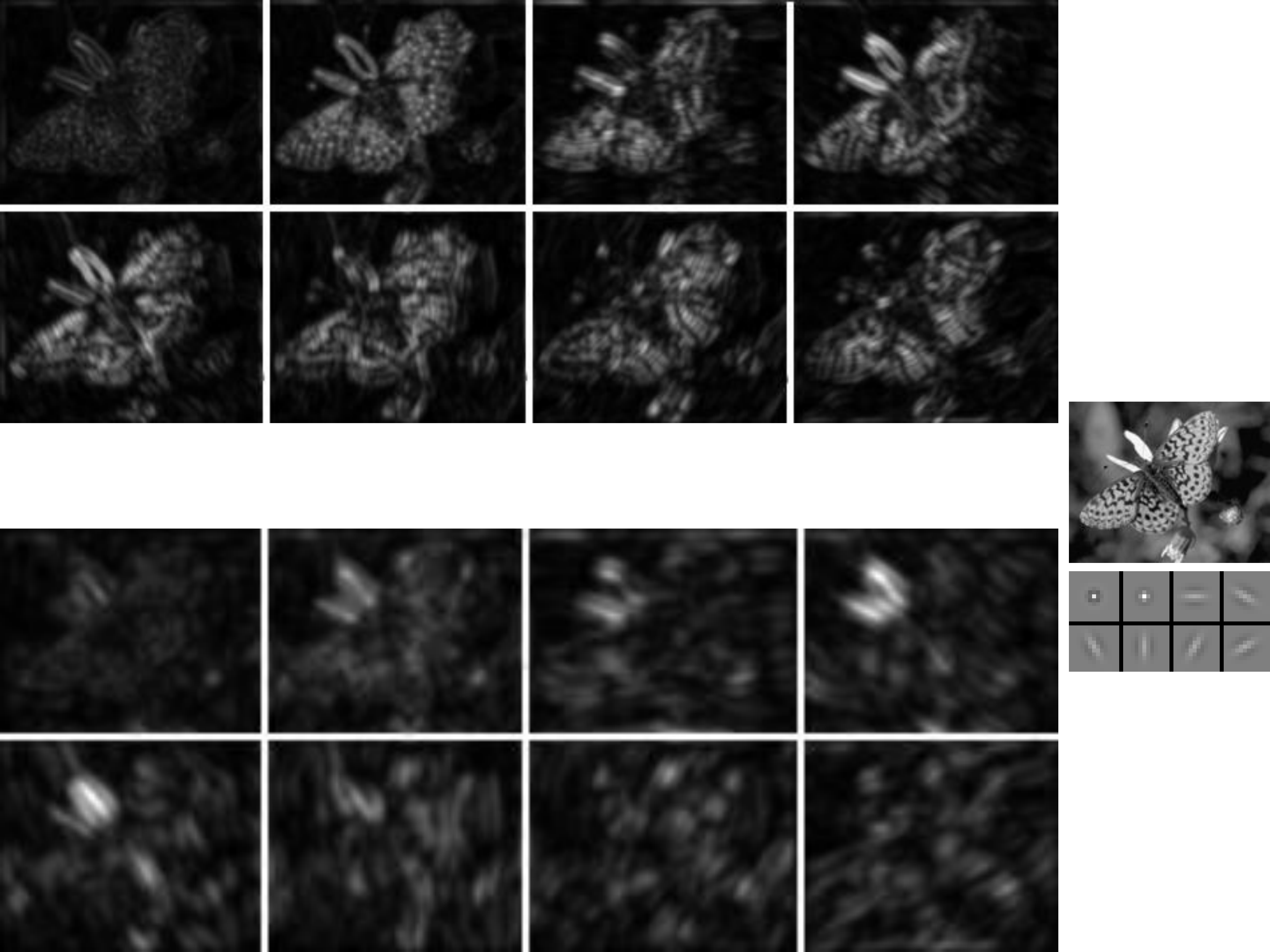

squared responses
vertical
classification
horizontal
smoothed mean

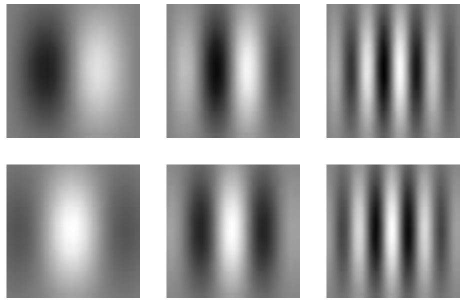






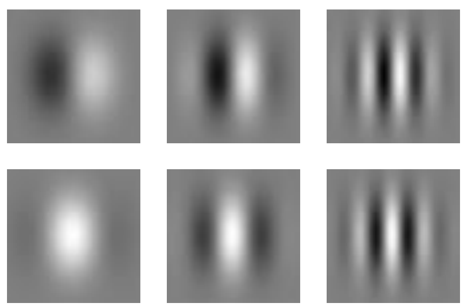


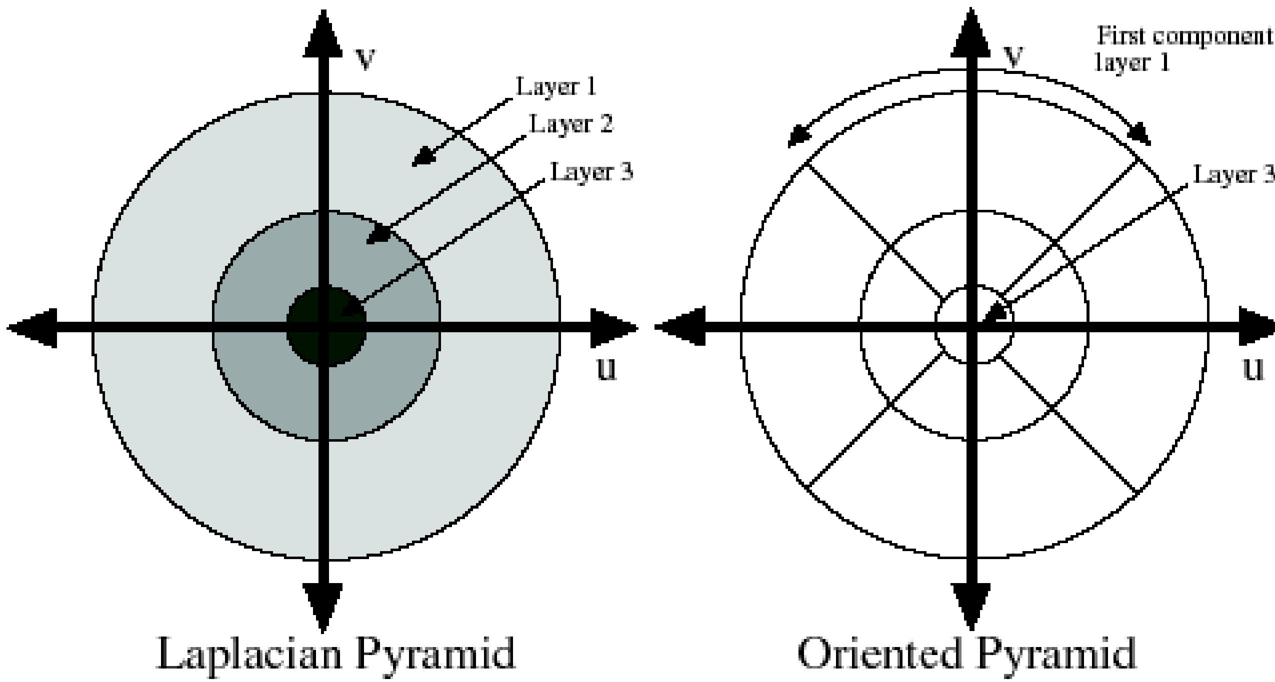




Gabor filters at different scales and spatial frequencies

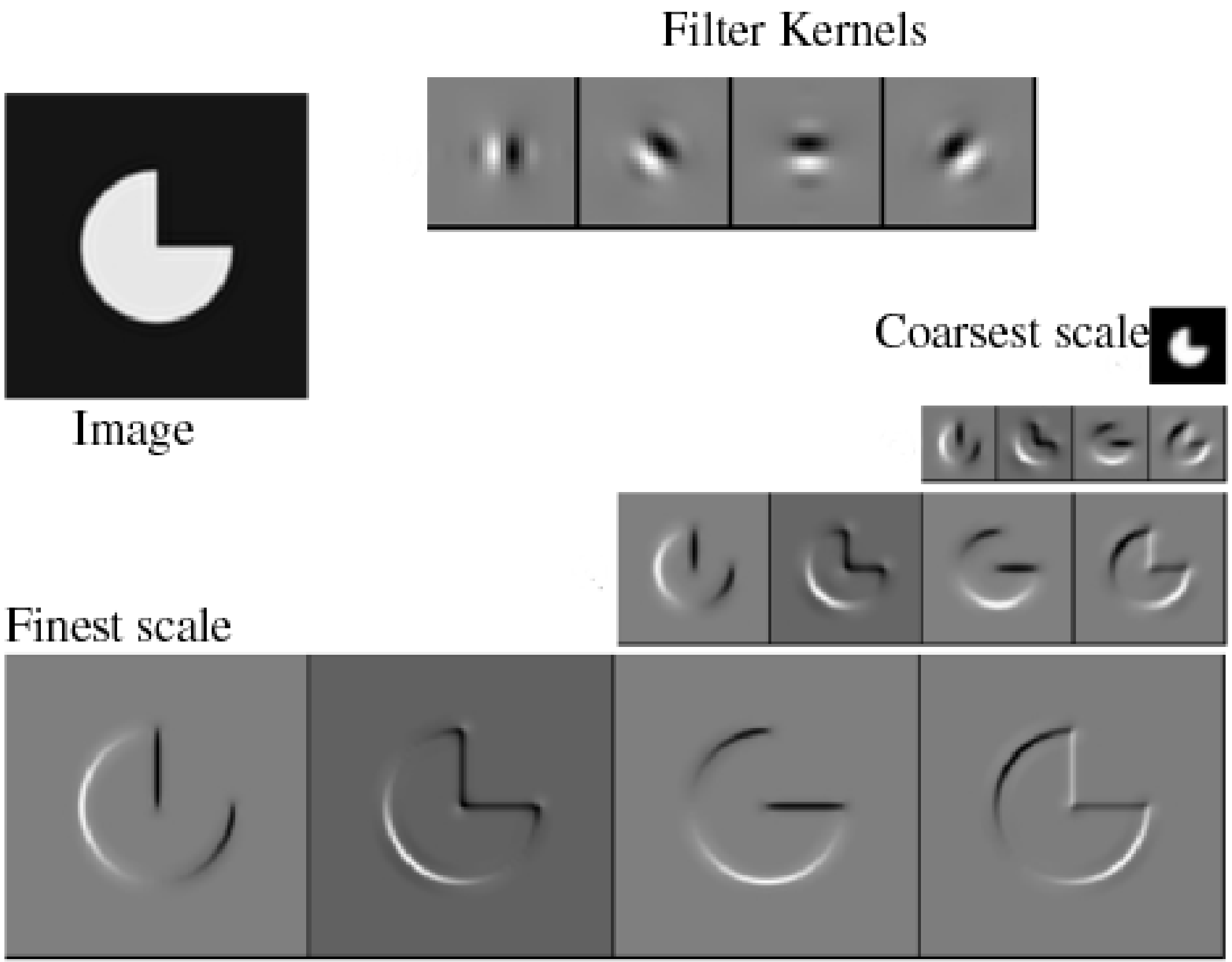
top row shows anti-symmetric (or odd) filters, bottom row the symmetric (or even) filters.



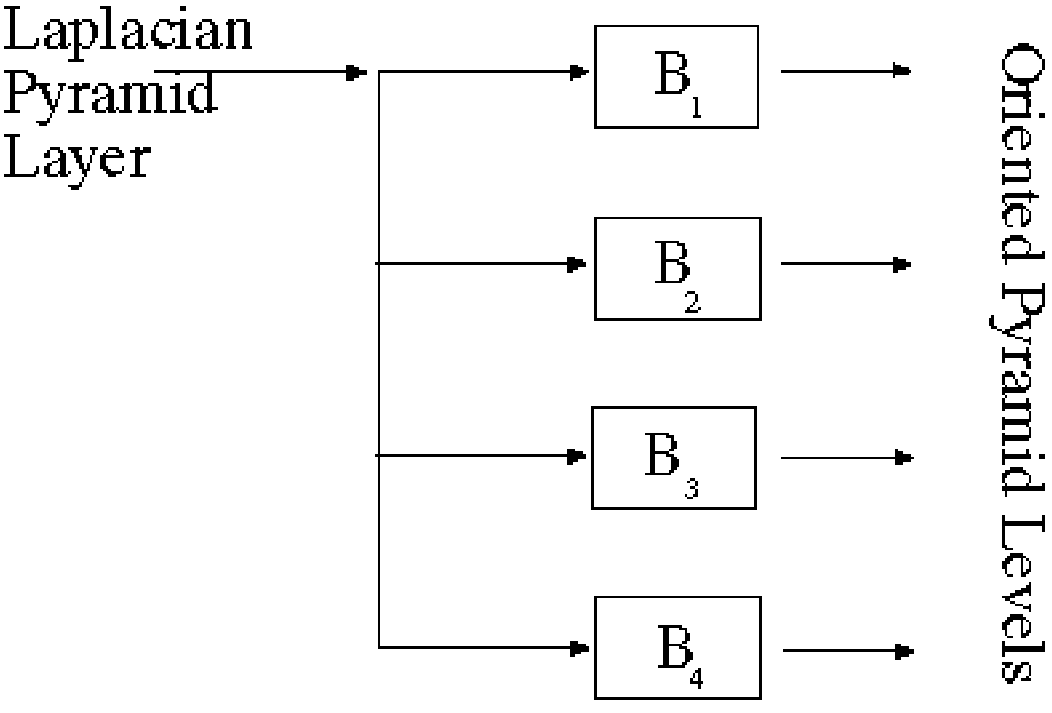

v
Layer 1
Layer 2
Layer 3
u
Laplacian Pyramid
v
First component of
layer 1
Layer 3
u
Oriented Pyramid

Oriented pyramids

- Laplacian pyramid is orientation independent
- Apply an oriented filter to determine orientations at each layer
  - by clever filter design, we can simplify synthesis
  - this represents image information at a particular scale and orientation

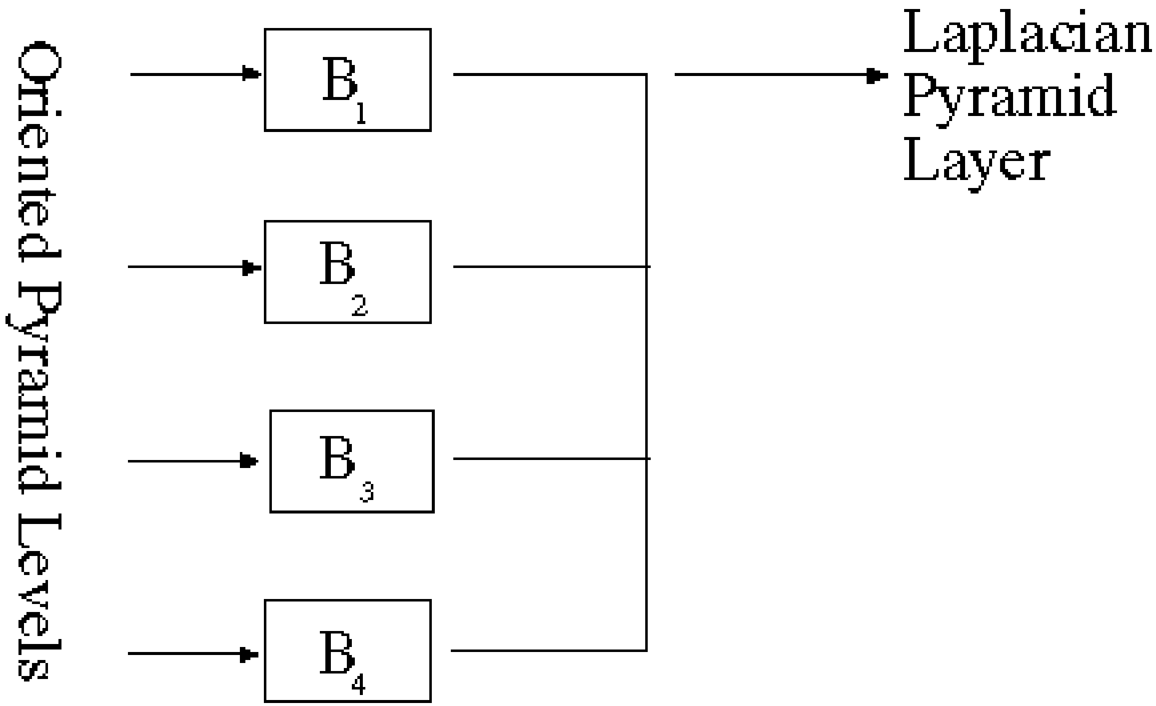


Reprinted from "Shiftable MultiScale Transforms," by Simoncelli et al., IEEE Transactions on Information Theory, 1992, copyright 1992, IEEE



Laplacian Pyramid Layer
B_1
B_2
B_3
B_4
Oriented Pyramid Levels

Analysis


Oriented Pyramid Levels
B1
B2
B3
B4
Laplacian Pyramid Layer

synthesis

Final texture representation

- Form an oriented pyramid (or equivalent set of responses to filters at different scales and orientations).
- Square the output
- Take statistics of responses
  - e.g. mean of each filter output (are there lots of spots)
  - std of each filter output
  - mean of one scale conditioned on other scale having a particular range of values (e.g. are the spots in straight rows?)

Texture synthesis

- Use image as a source of probability model
- Choose pixel values by matching neighbourhood, then filling in
- Matching process
  - look at pixel differences
  - count only synthesized pixels

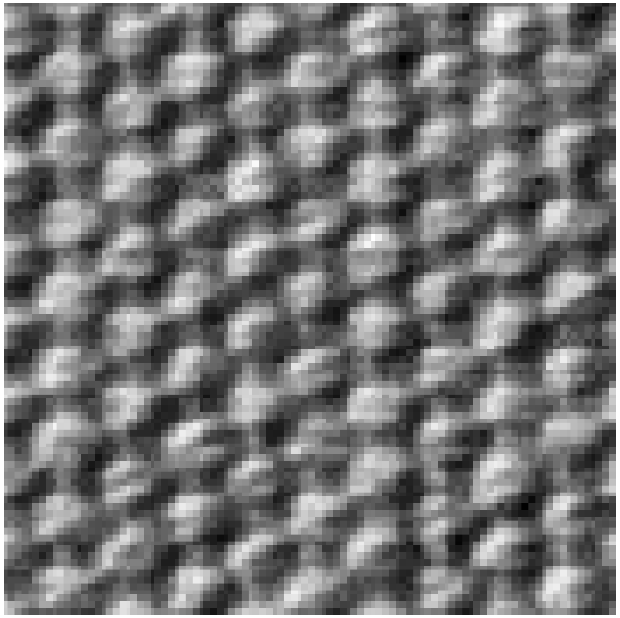
Histograms: principle

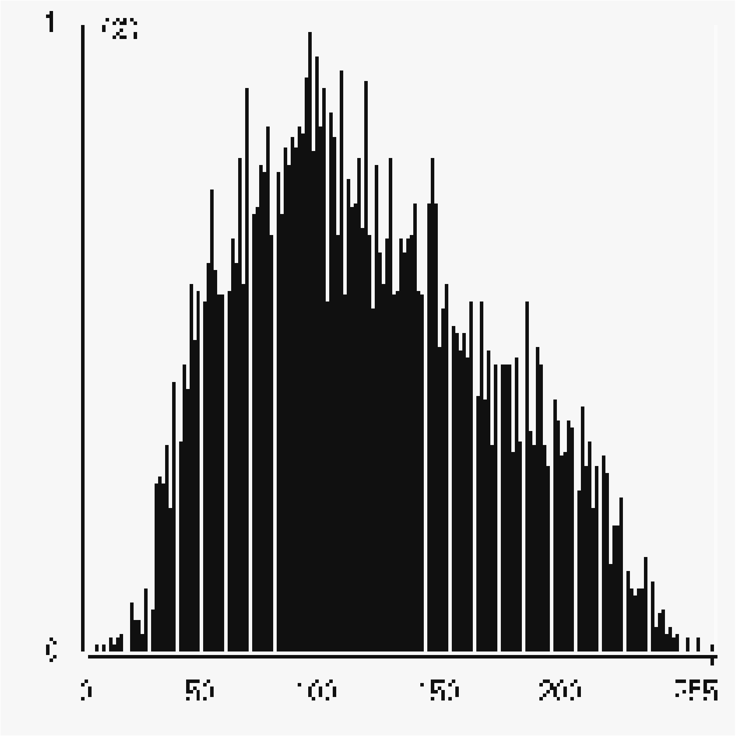
Intensity probability distribution

Captures global brightness information in a compact, but incomplete way

Doesn't capture spatial relationships

Histograms : example

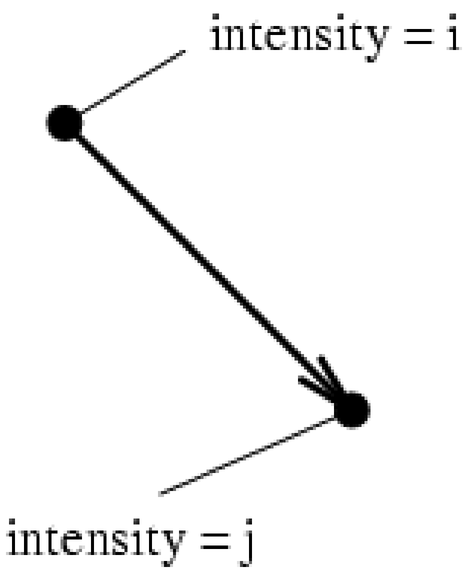


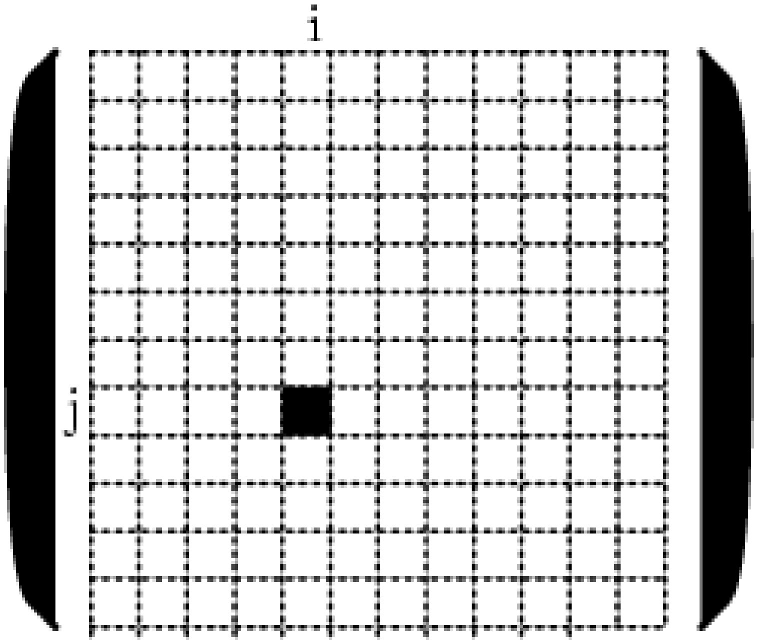


Histograms: co-occurrence matrix

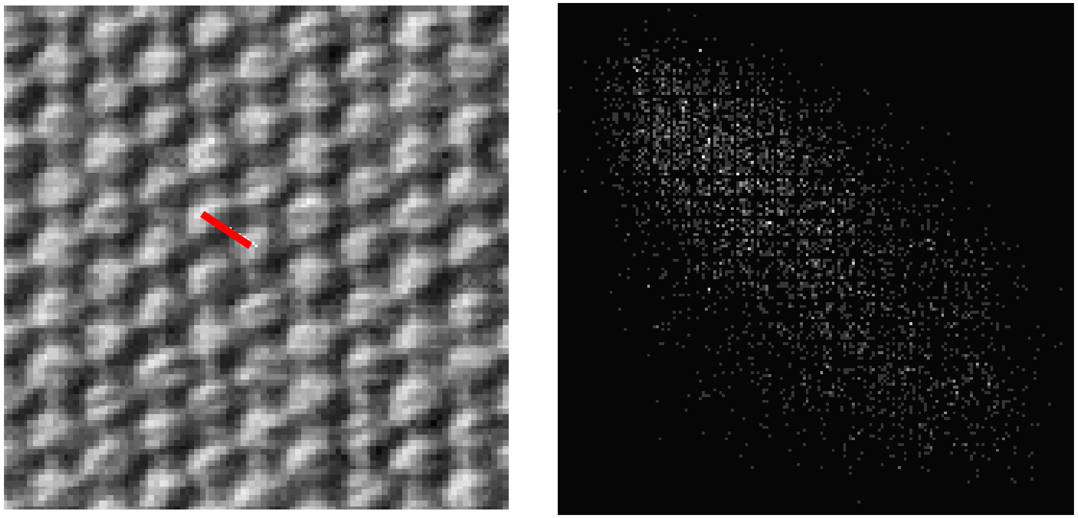
Co-occurrence matrix = probability distributions for intensity pairs

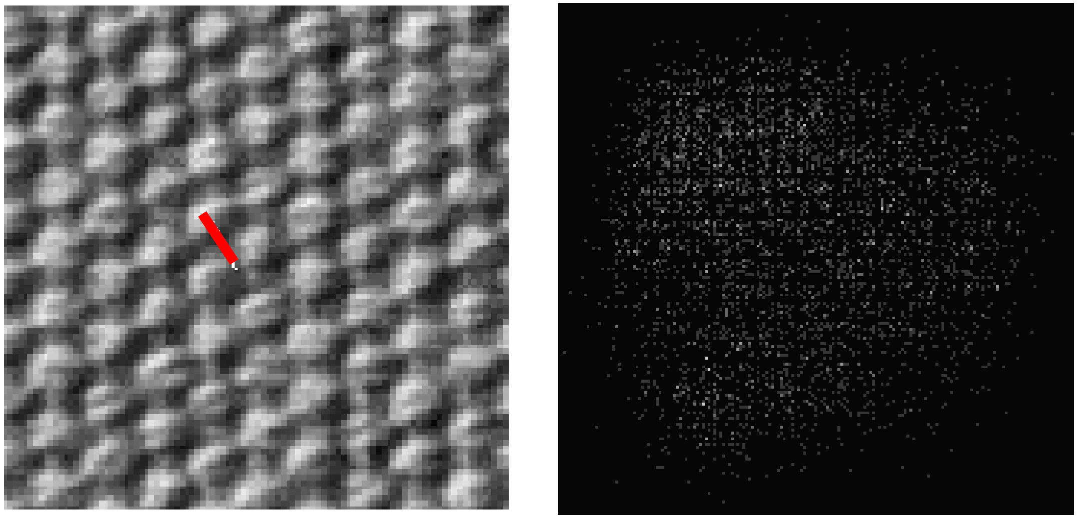
Contains information on some aspects of the spatial configurations



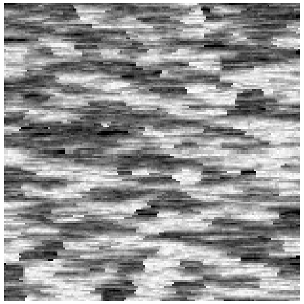
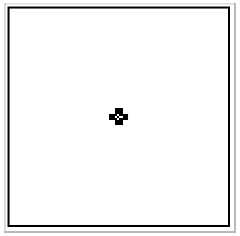


Histograms : co-occurrence matrix

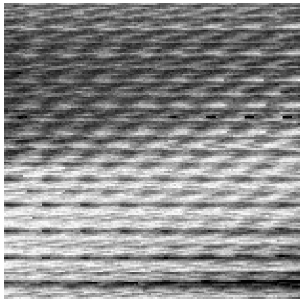
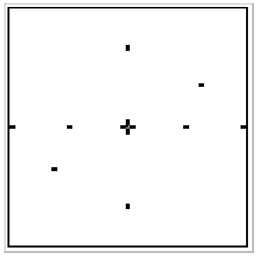




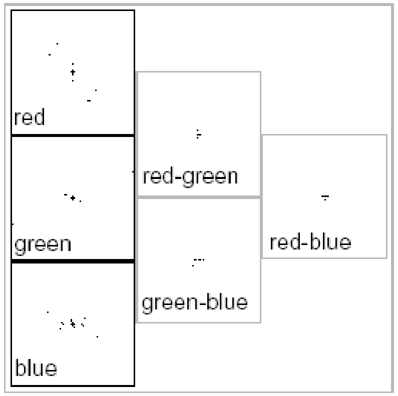
Texture synthesis [Zalesny & Van Gool 2000]

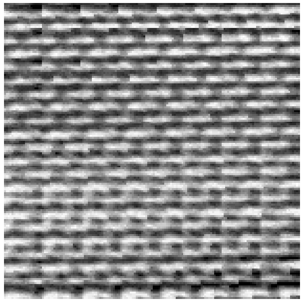
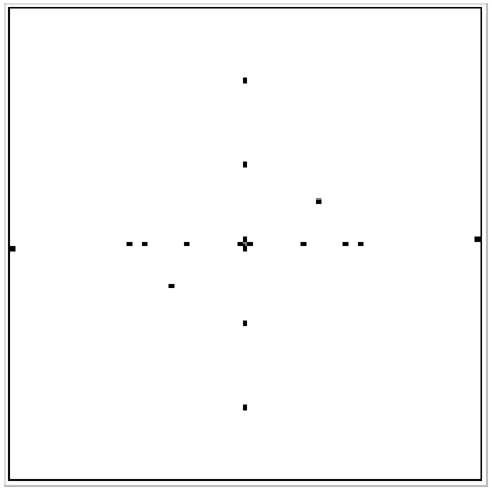



2 analysis iterations

6 analysis iterations



9 analysis iterations

View-dependent texture synthesis

[Zalesny & Van Gool 2000]

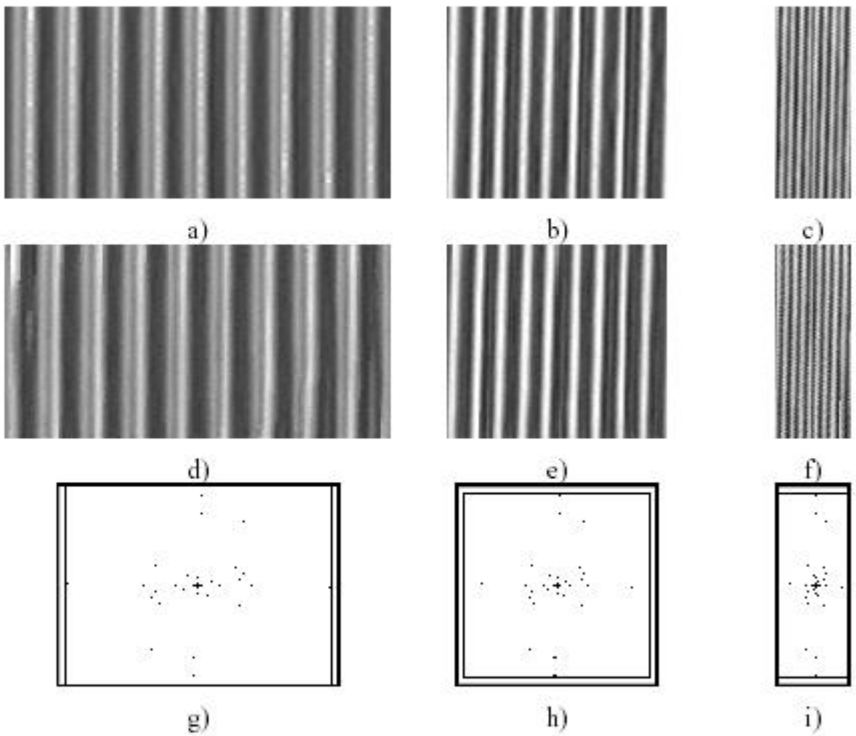
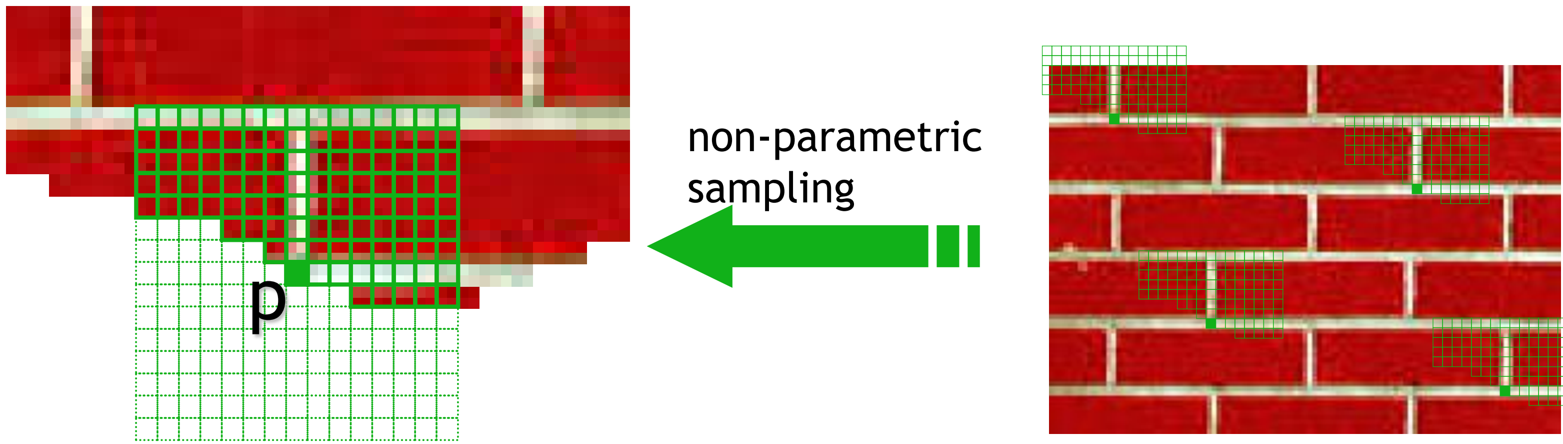


Fig. 13. Straw (CUReT, 40b); a) original image for a perpendicular view; b) synthesized texture based on a model for a); c) original image for an oblique view (68°); d) result of contracting b); e) texture synthesis based on a completely new model extracted from c); f) texture synthesis based on a transformed neighborhood system for b) and new difference distributions from c).

Image-based approaches

- No difficult analysis
- Let’s ‘Cut & Paste’

Efros & Leung ’99



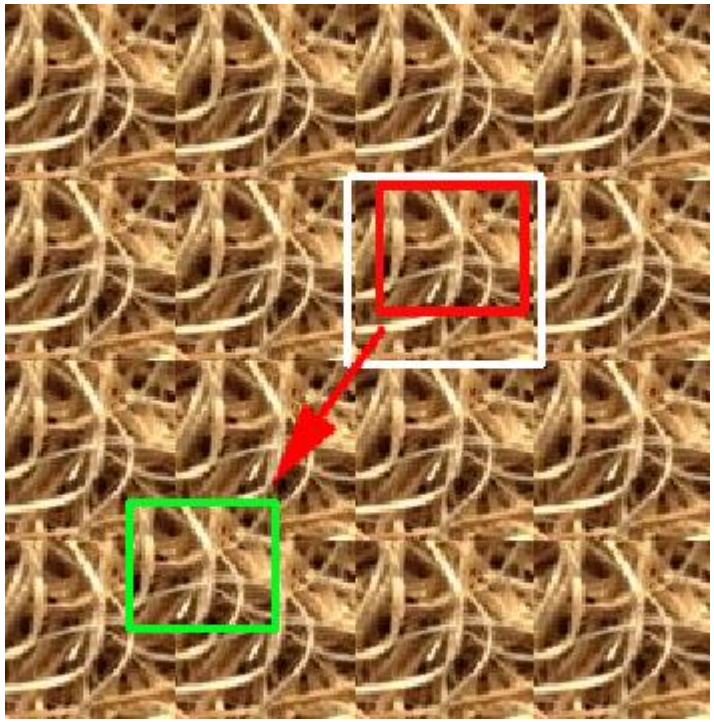
Synthesizing a pixel

Input image

- Assuming Markov property, compute P(**p**|N(**p**))
  - Building explicit probability tables infeasible
  - Instead, let’s *search the input image* for all similar neighborhoods — that’s our histogram for **p**
- To synthesize **p**, just pick one match at random

Chaos Mosaic [Xu, Guo & Shum, ‘00]

input



idea

result

- Process: 1) tile input image; 2) pick random blocks and place them in random locations 3) Smooth edges

Used in Lapped Textures [Praun et.al,’ 00]

Chaos Mosaic [Xu, Guo & Shum, ‘00]

input

result

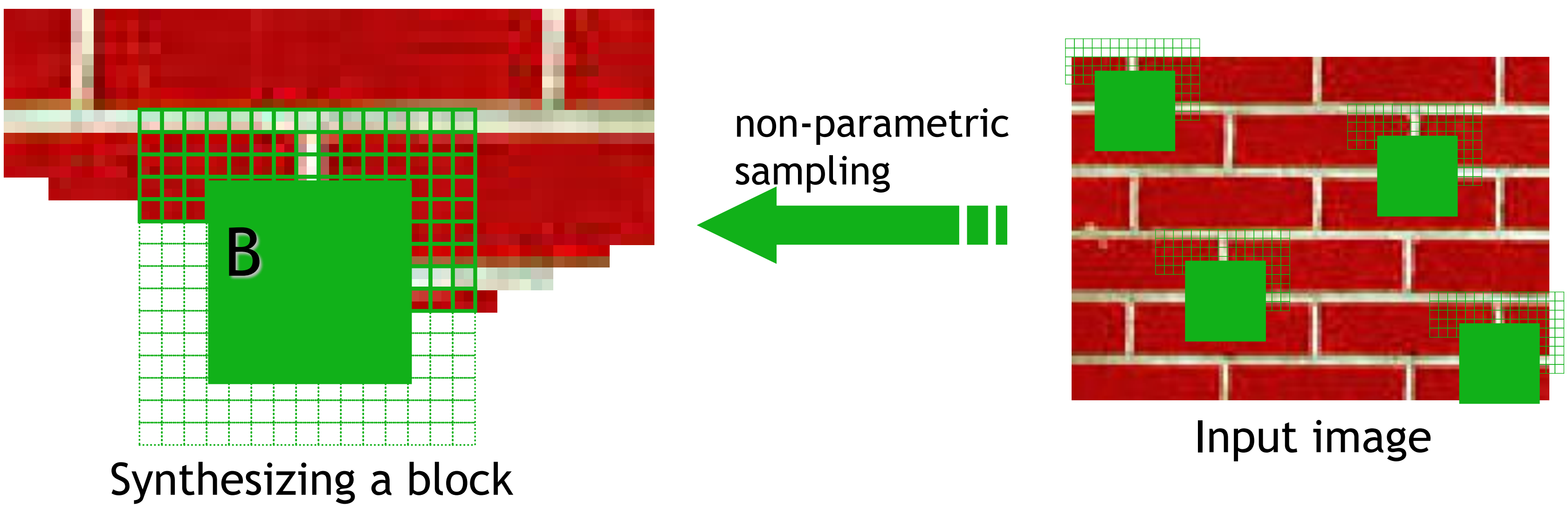
Of course, doesn’t work for structured textures

Image Quilting [Efros & Freeman, ‘01]

- Idea:
  - let’s combine random block placement of Chaos Mosaic with spatial constraints of Efros & Leung

- Related Work (concurrent):
  - Real-time patch-based sampling [Liang et.al. ’01]
  - Image Analogies [Hertzmann et.al. ’01]

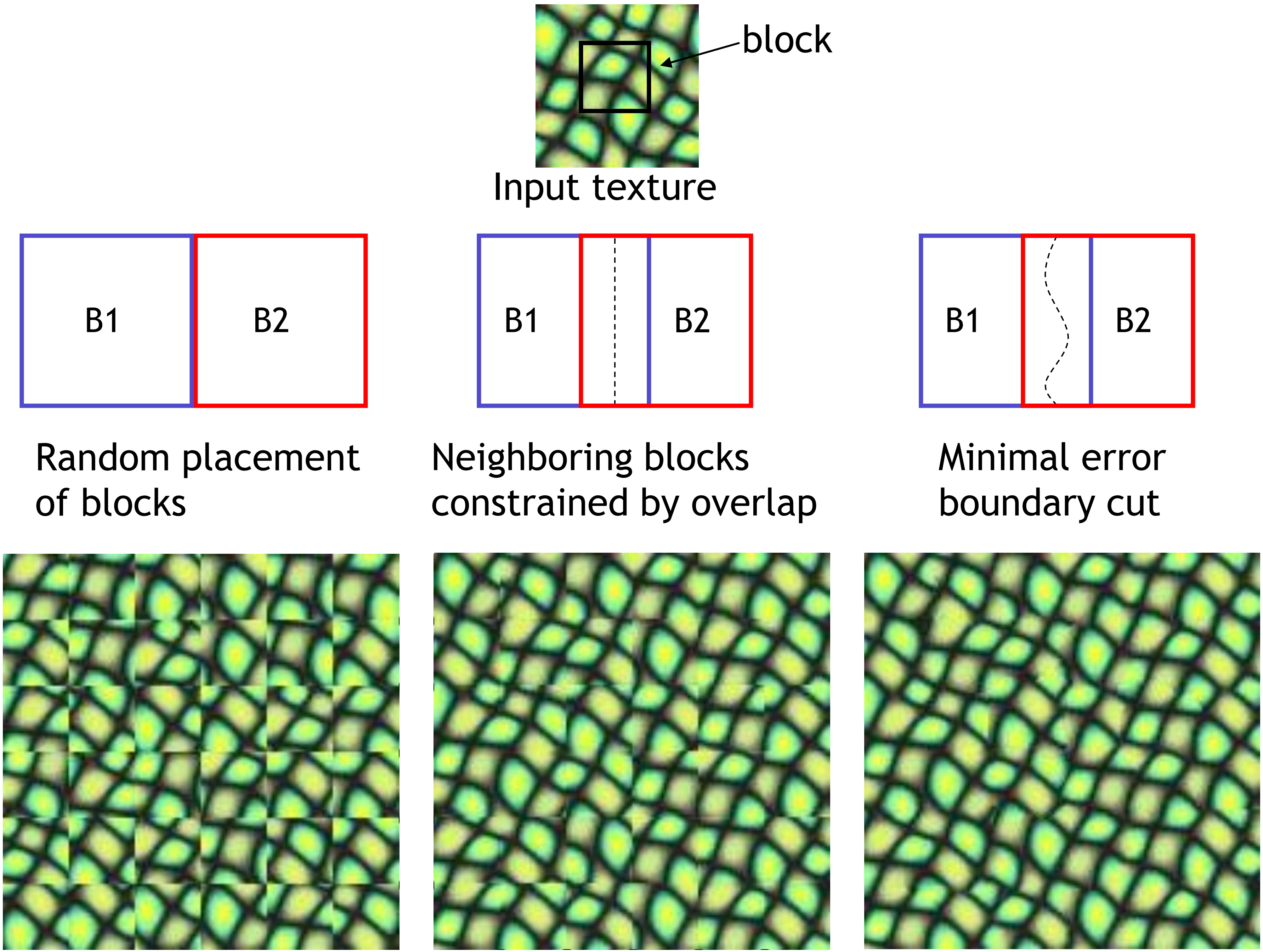
Efros & Leung ’99 extended



- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

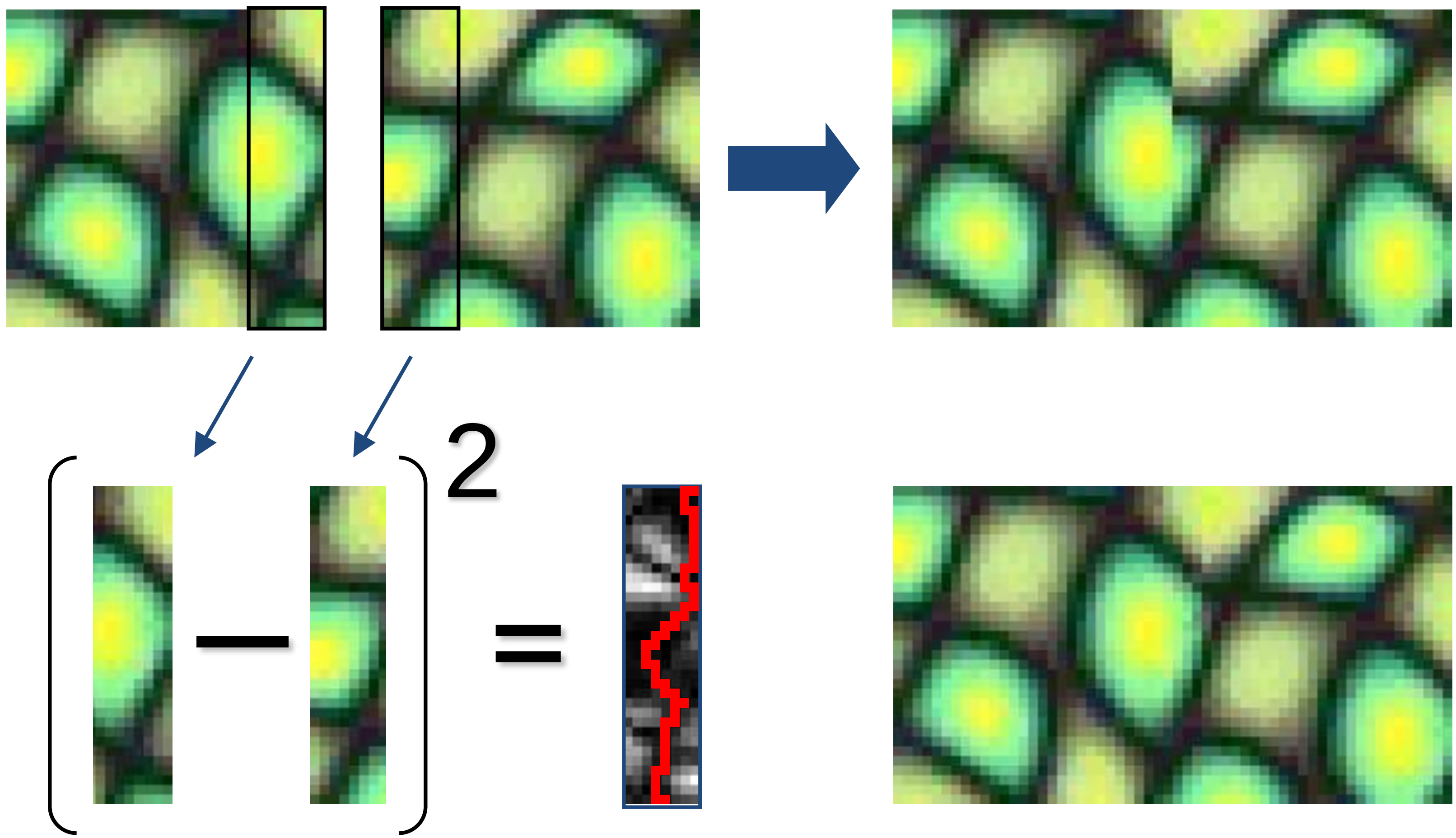
- Exactly the same but now we want P(B|N(B))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!


block
Input texture
B1
B2
B1
B2
B1
B2
Random placement of blocks
Neighboring blocks constrained by overlap
Minimal error boundary cut

Minimal error boundary

overlapping blocks

vertical boundary

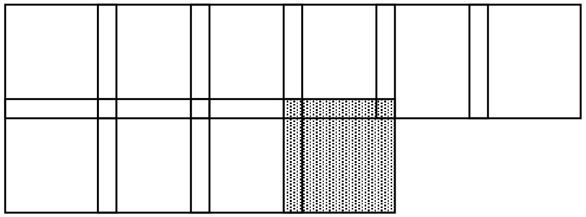


overlap error

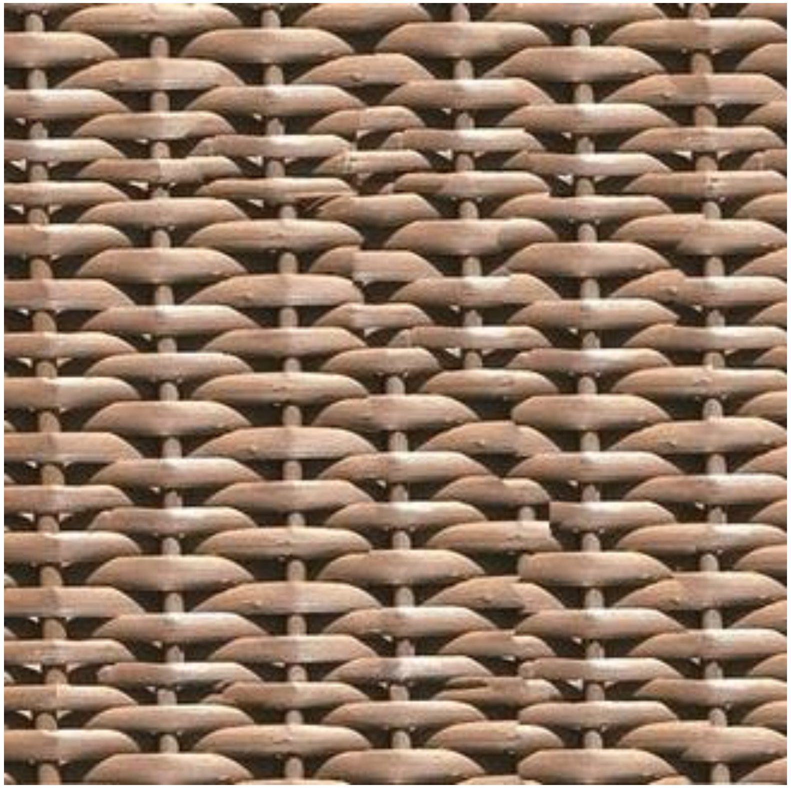
min. error boundary

Algorithm

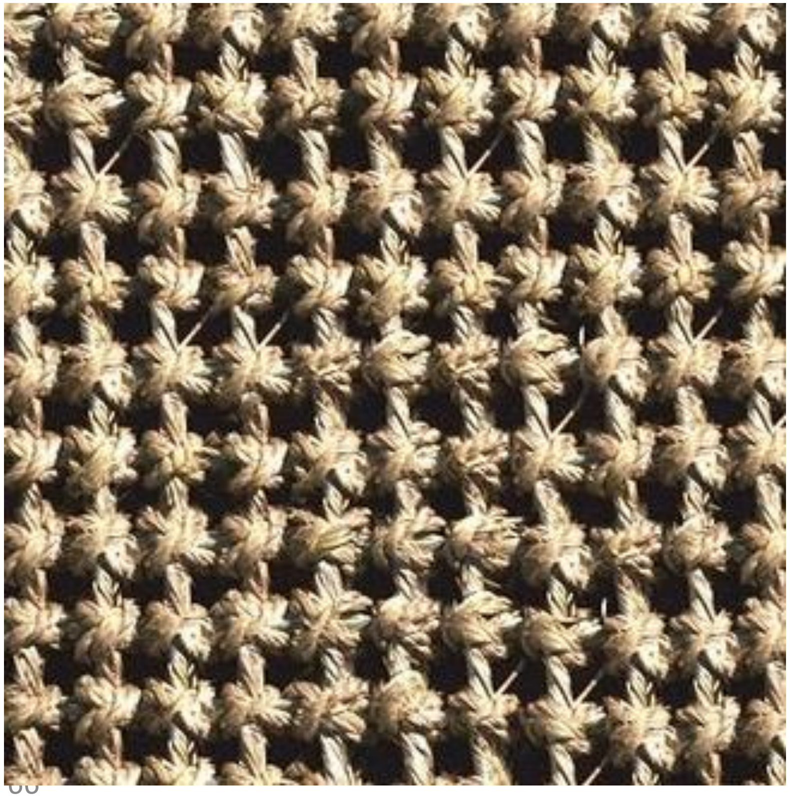
– Pick size of block and size of overlap
– Synthesize blocks in raster order



– Search input texture for block that satisfies overlap constraints (above and left)
– Paste new block into resulting texture
  - use dynamic programming to compute minimal error boundary cut







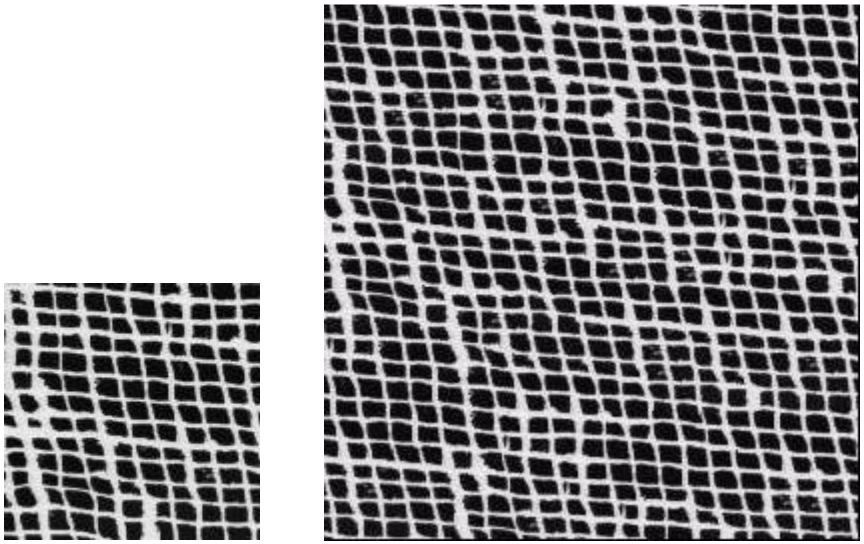


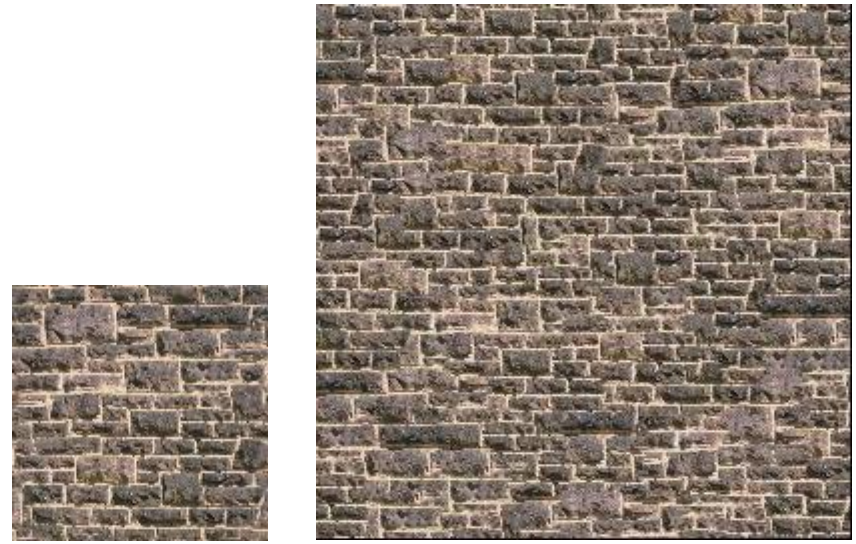



College Inn


College Inn





Failures

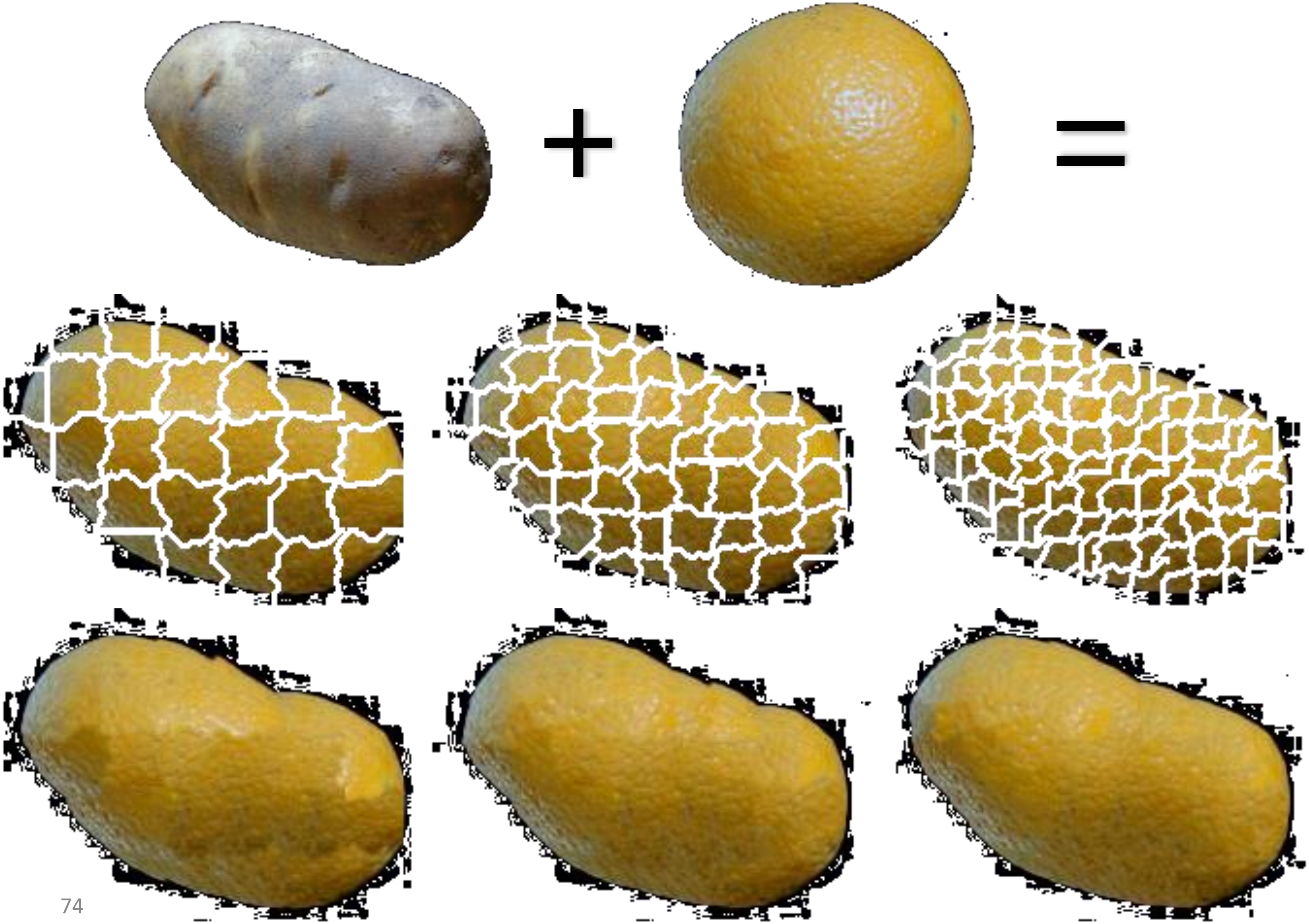
(Chernobyl
Harvest)

Texture Transfer

Take the texture from one object and "paint" it onto another object

- This requires separating texture and shape
- That's HARD, but we can cheat
- Assume we can capture shape by boundary and rough shading

Then, just add another constraint when sampling: similarity to underlying image at that spot



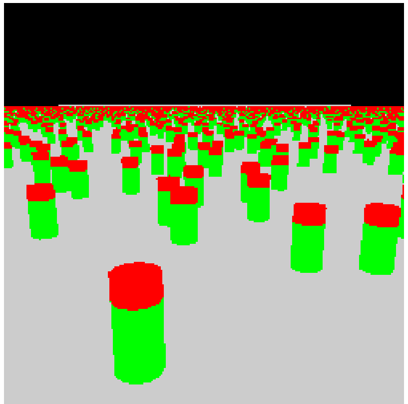
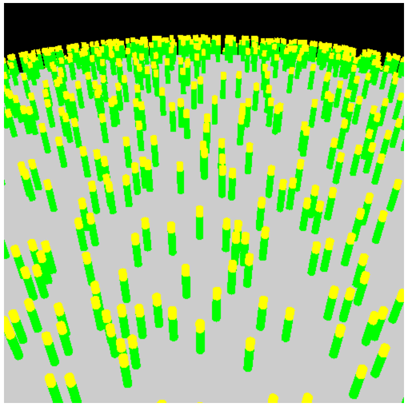
Shape from textures

- Possibility to recover normals both for deterministic and probabilistic textures

Why do we see more flowers in the distance?

[Leung & Malik CVPR97]



Perpendicular textures

Some recent results at CVG

Live 3D Reconstruction on Mobile Phone

Tanskanen et al. ICCV2013

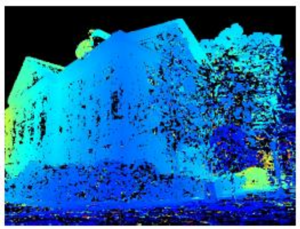
Live reconstruction with interactive feedback
All calculations performed on mobile devices


astrivis
Home
Technology
Gallery
Jobs
About Us
astrivis
capture the world in 3D with your mobile device
anywhere - anytime - in real-time
SAMSUNG

ETH

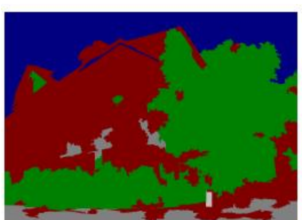
Joint 3D reconstruction and class segmentation

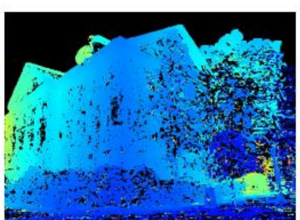
(Haene et al CVPR13)



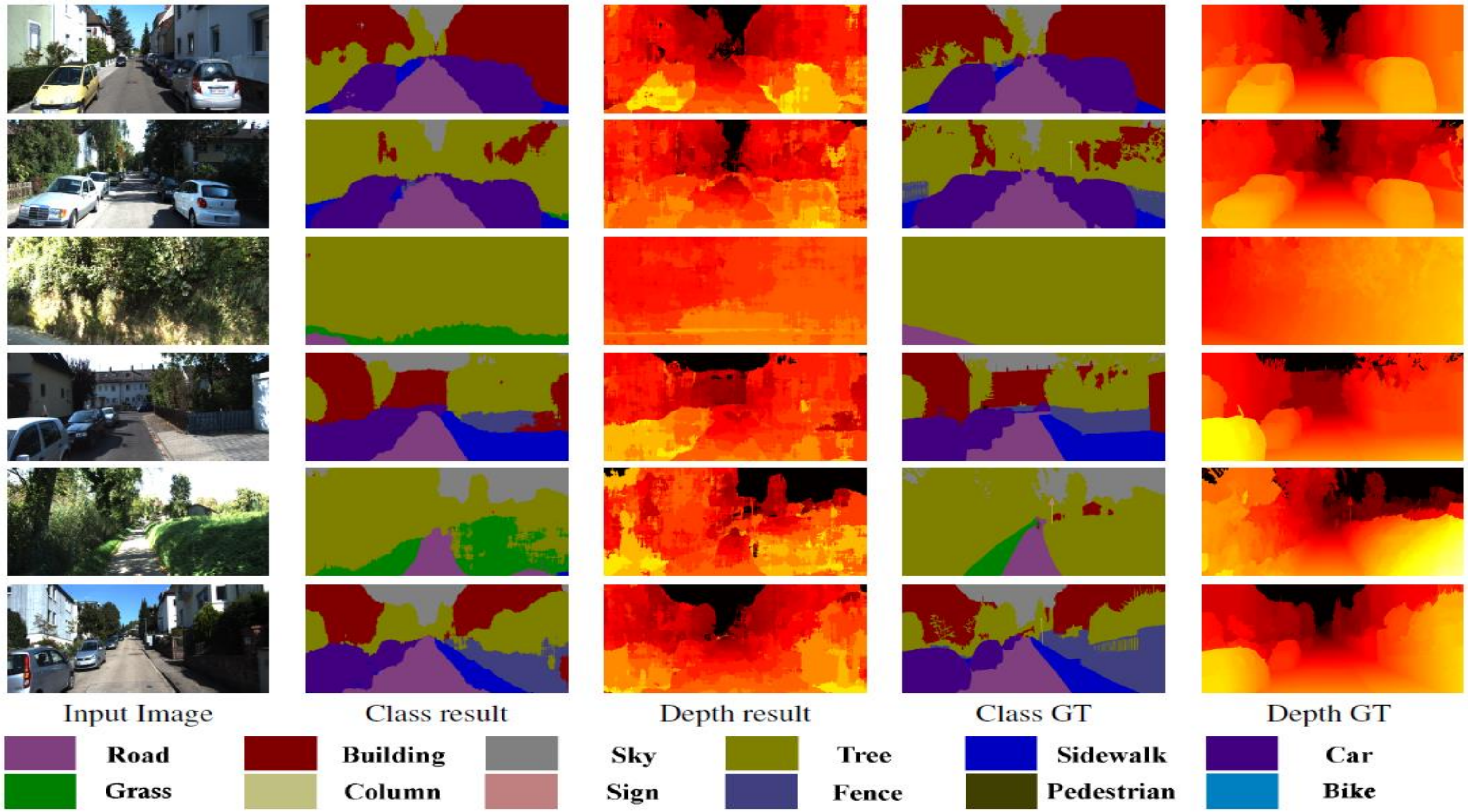
reconstruction only
(uniform smoothness prior)

joint reconstruction
and segmentation
(ground, building, vegetation, stuff)





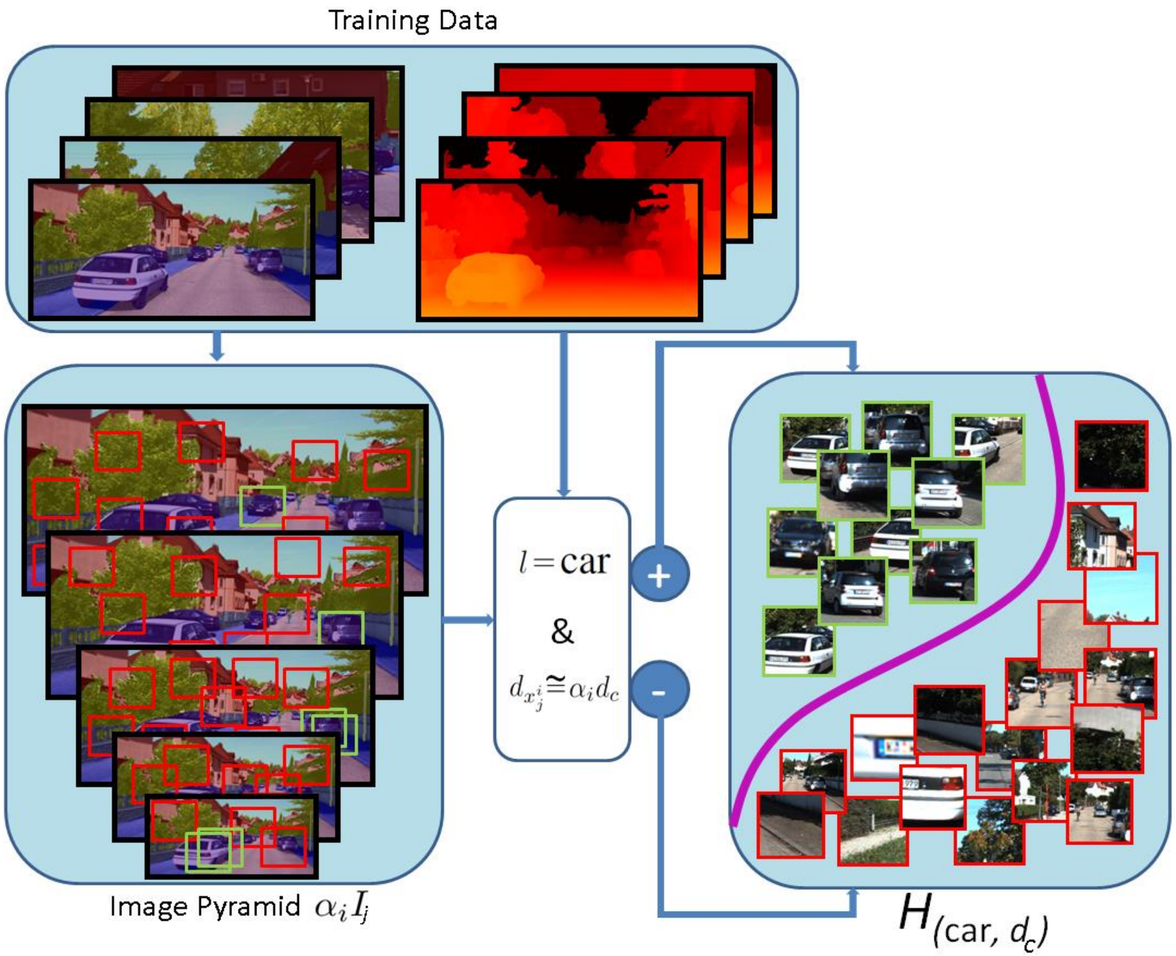
Joint classification of class and depth?



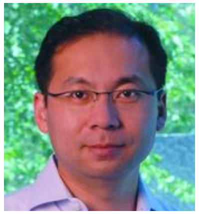
	Global	Average	Building	Tree	Sky	Sidewalk	Fence	Road	Grass	Column	Car
Class-only classifier	80.2	66.2	87.0	82.8	89.7	68.4	31.6	**84.8**	61.2	7.3	83.2
Joint classifier	**82.4**	**72.2**	**87.2**	**84.6**	**91.6**	**76.5**	**39.4**	83.2	**69.9**	**28.5**	**88.9**

Learning joint classifier for class and depth

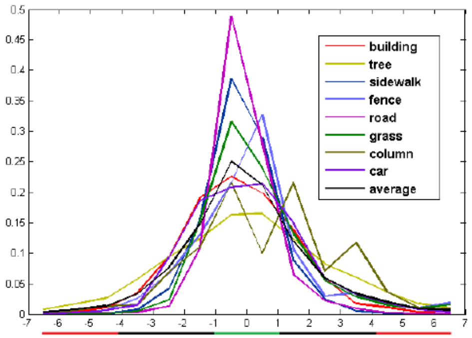
(Ladicky, Shi and Pollefeys CVPR 2014)



Lubor Ladicky



Jianbo Shi

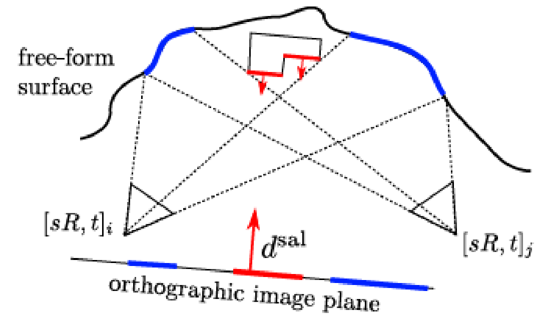


How well do we estimate depth?

proposed approach is both efficient and unbiased towards specific depths

Automatic Registration of RGB-D Scans via Salient Directions (Zeisl et al. ICCV13)

original photographs from two stations



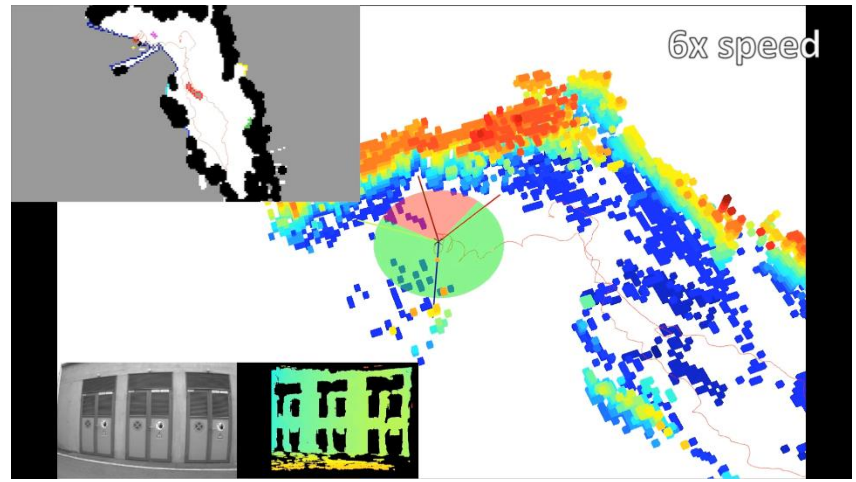
normalized views dir A

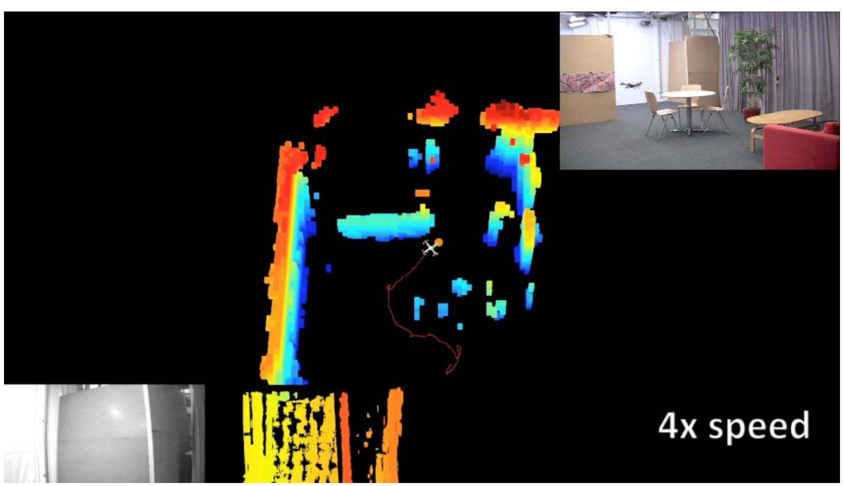
normalized views dir B

multiple aligned 3D scans

Vision-based autonomous drones and robots

(Meier et al. ICRA11, Fraundorfer et al. IROS12, Scaramuzza et al. RAM14...)





PX4 autopilot

More on PixHawk:
http://pixhawk.ethz.ch

pixhawk

auterion

Computer Vision on mobile Robots

Real-time processing on constrained platforms is feasible

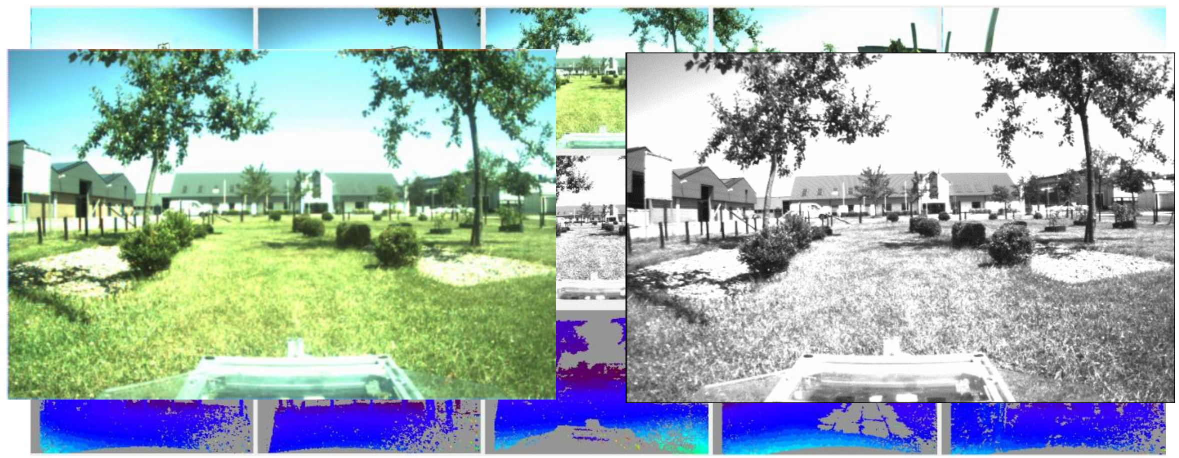
Pascal Gohl, **Dominik Honegger**, Sammy Omari, Markus Achtelik, Marc Pollefeys and Roland Siegwart. **Omnidirectional Visual Obstacle Detection using Embedded FPGA**. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2015.

Lightweight, low-power embedded 360 depth camera system

5 stereo pairs
360^{o} SGM stereo
5W / 100g

Multi Camera System – Performance

The system can process ten image stream simultaneously and estimate disparity values



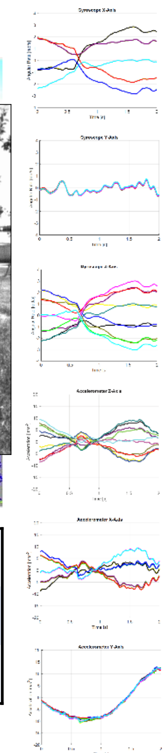
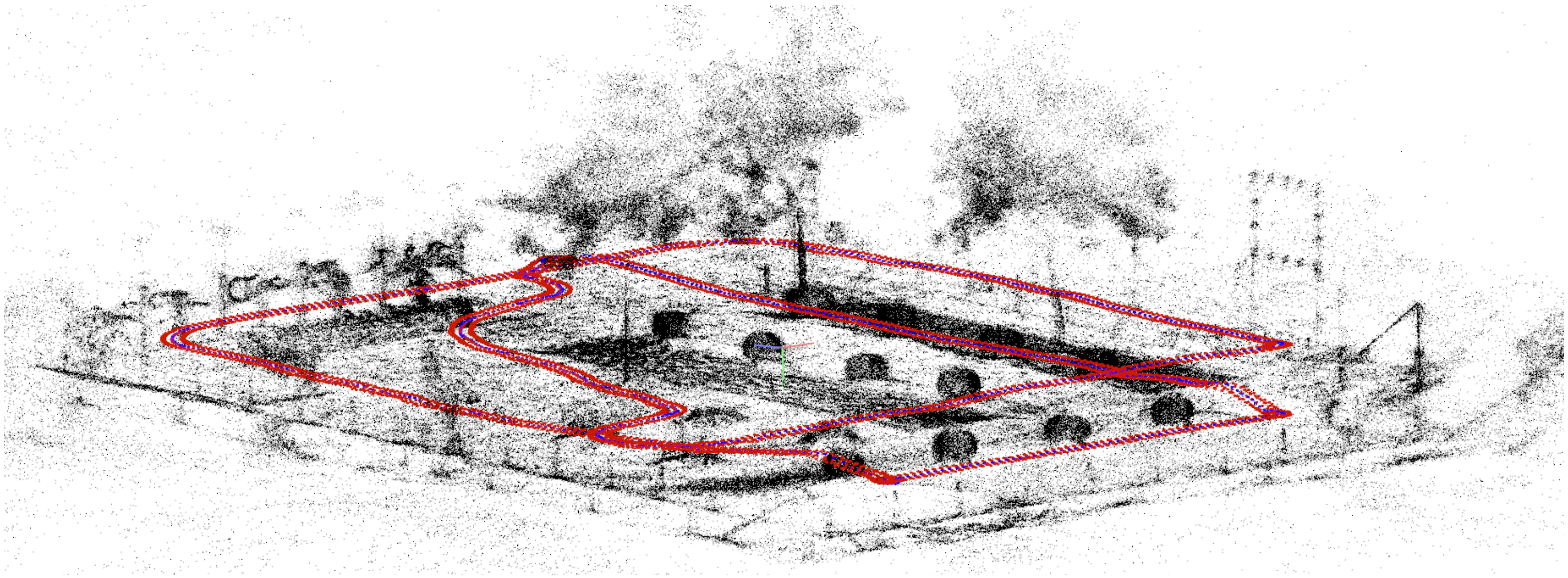


Image Dimension	752x480 Pixel	IMU Data	6-Axis 800 Hz
Update Rate	12 Hz	Synchronization	Hardware
Power Consumption	5.5 W	USB UVC	Compliant

Applications – 360˚ SLAM System

Localization and mapping on a garden robot

Autonomous Vehicles – SLAM



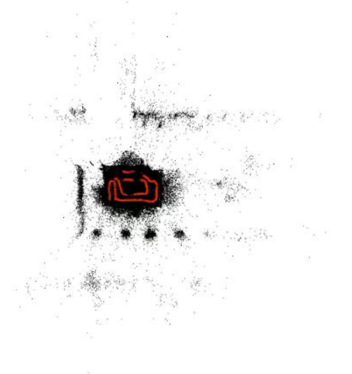
[video credit: Marcel Geppert]

Simultaneous Localization and Mapping using Generalized Cameras
Master thesis by Marcel Geppert
supervised by Johannes Schönberger and Torsten Sattler

Horizon 2020
European Union funding
for Research & Innovation

Applications – 360° SLAM System

Localization and mapping on a garden robot



Marcel Geppert **Simultaneous Localization and Mapping using Generalized Cameras.** Master's thesis, ETH Zürich (2017)

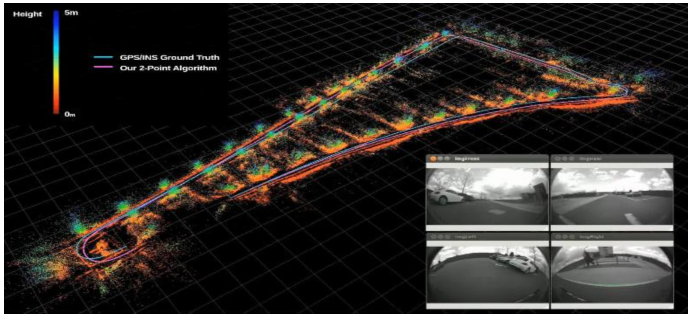
Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: **Pixelwise view selection for unstructured multi-view stereo**. In: European Conference on Computer Vision (ECCV). (2016)

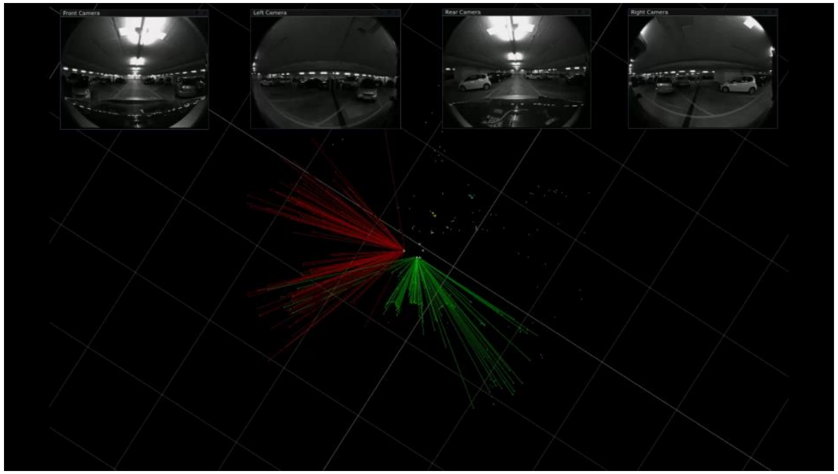
3D mapping for autonomous driving

(Lee et al CVPR13;
Heng et al. ICRA14;
Haene et al…)

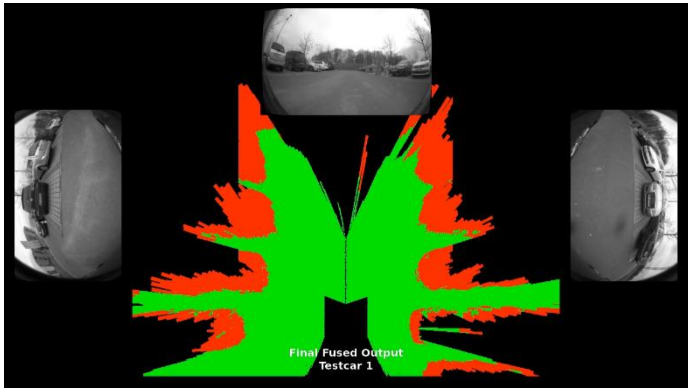
V-CHARGE

Dense real-time temporal fisheye stereo (GPU)





omnidirectional visual simultaneous localization and mapping



cameras as main sensors: *no lasers, no GPS*

Semantic Stixels (stereo + semantic)

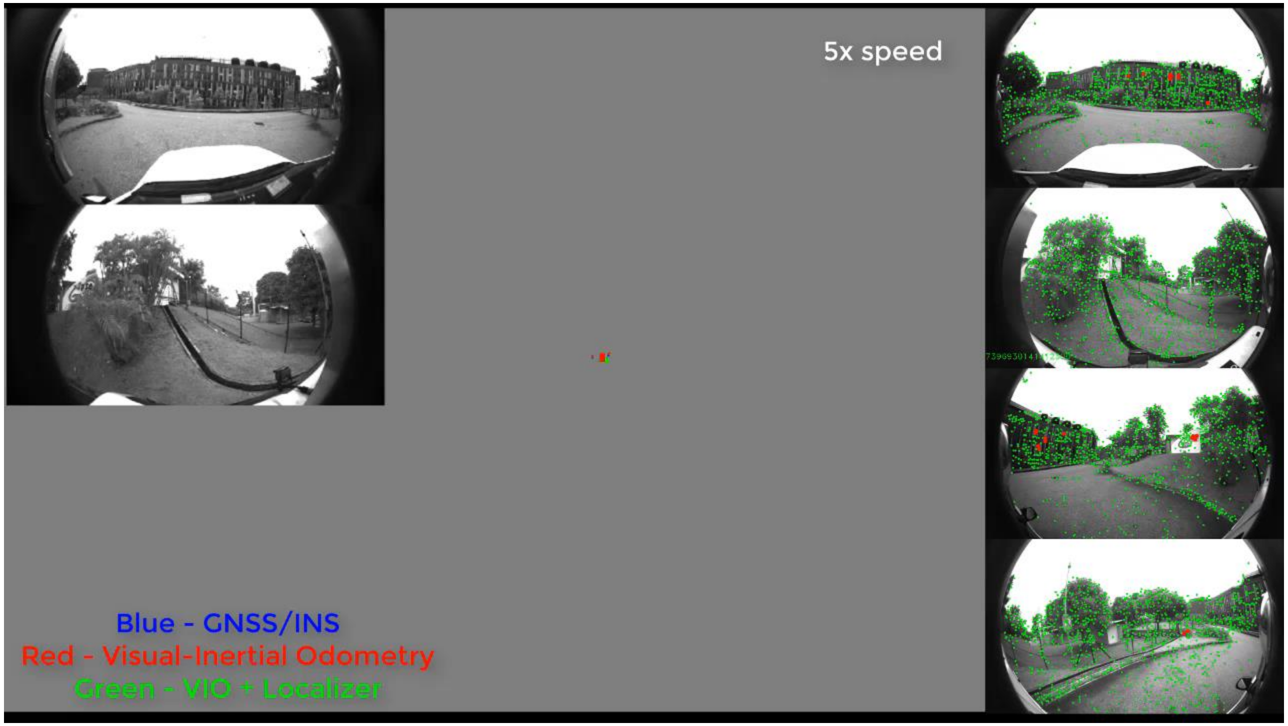
Schneider et al. IV 2016

AutoVision

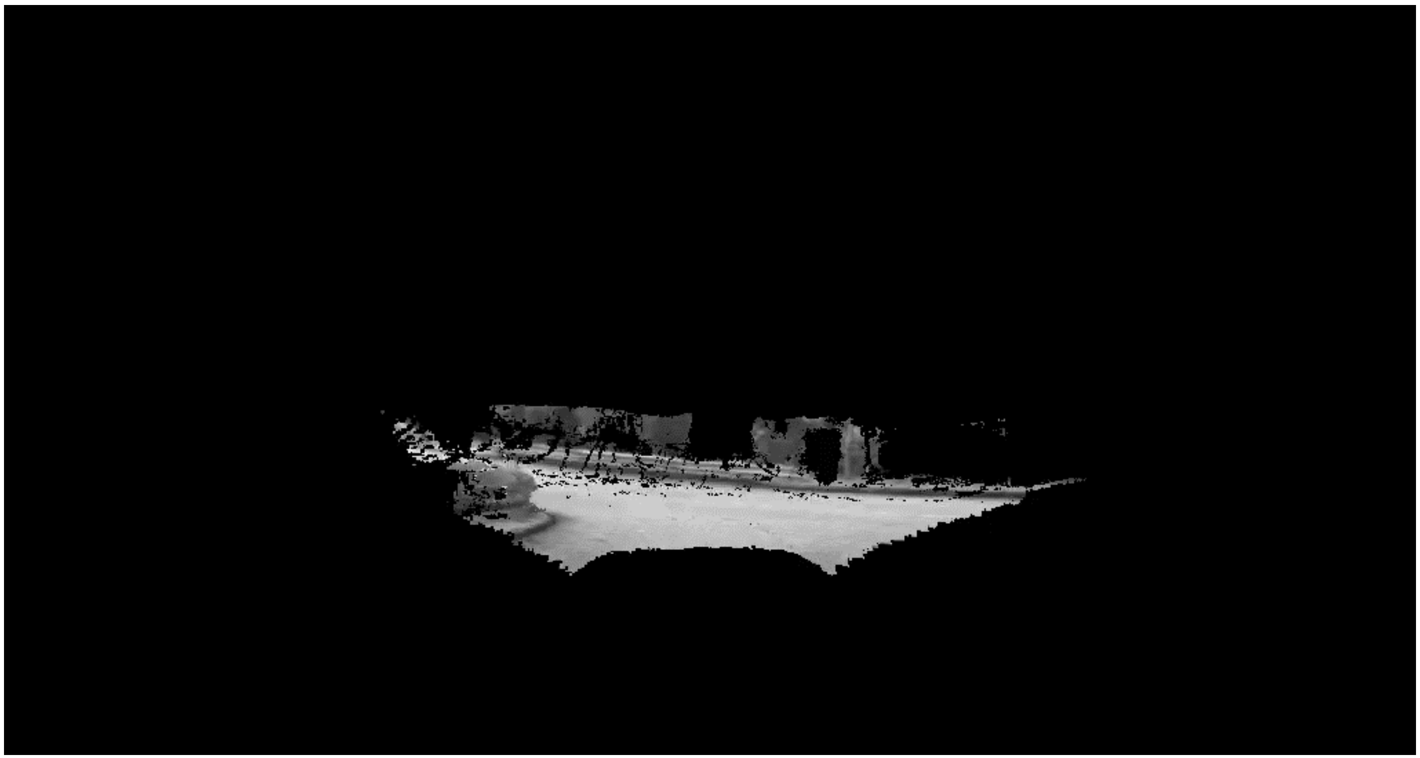
Collaborative project between DSO – NUS – ETH

Specifics:
- Passive sensing only
- All weather, day/night
- Off-road capable

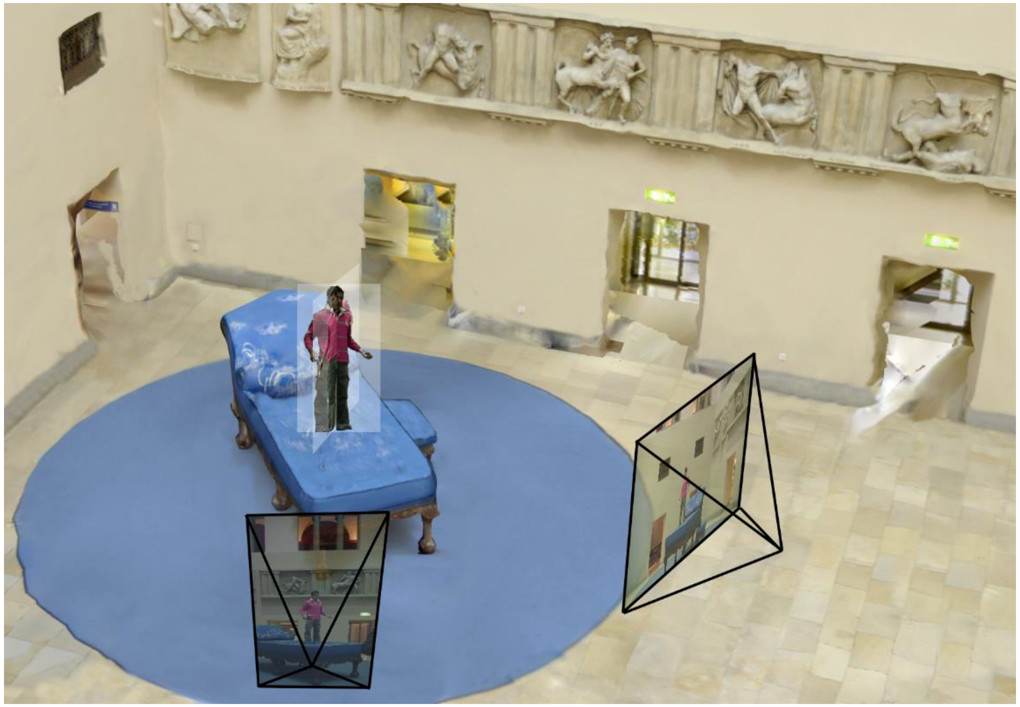
VIO + relocalizer



Dense 3D surface reconstruction



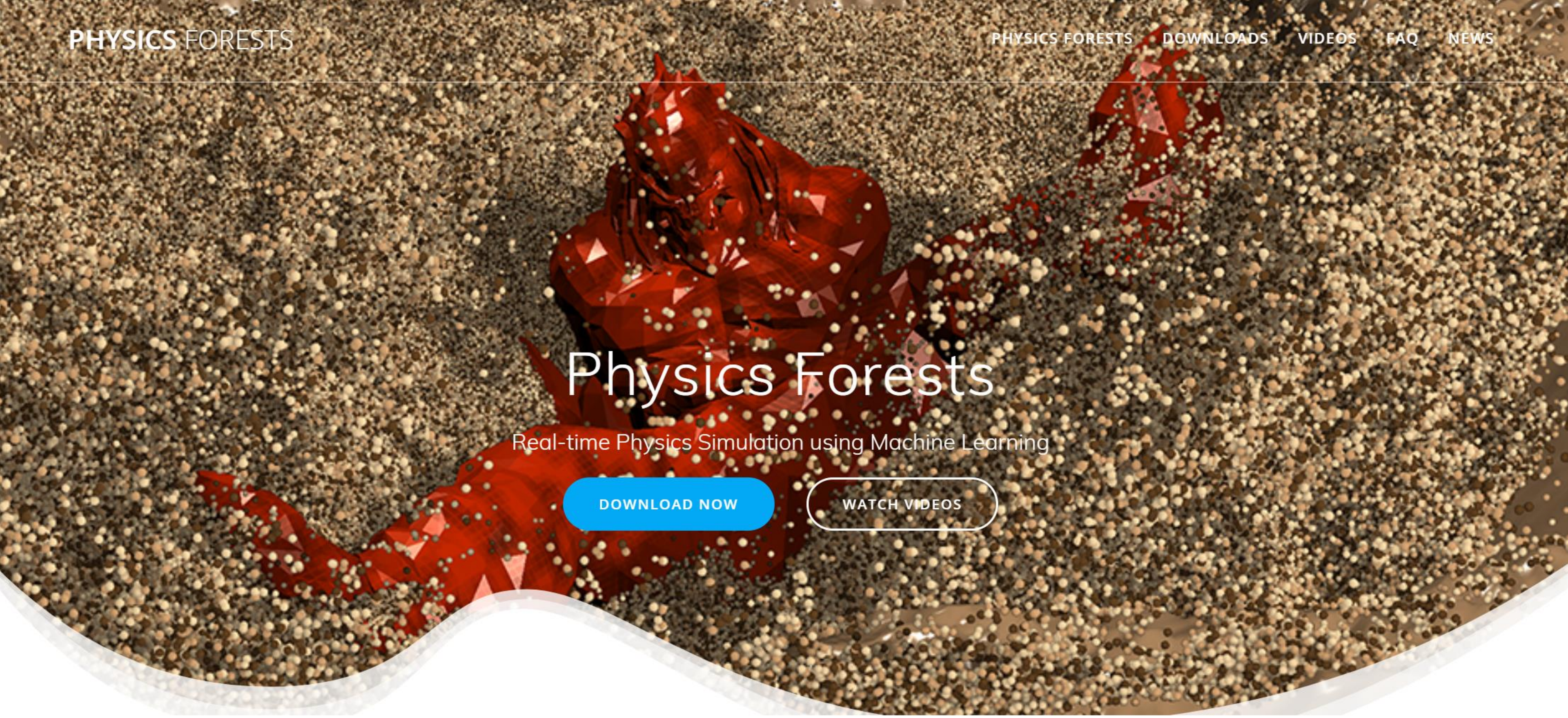
Unstructured Video-Based Rendering



Interactive viewer, more results & datasets available at:
http://cvg.ethz.ch/research/unstructured-vbr/

ETH

erc Starting Grant 4D Video



Physics Forests

Real-time Physics Simulation using Machine Learning

DOWNLOAD NOW WATCH VIDEOS

What's next?

Master Studies

- Visual Computing track (Computer Science)
- Robotics, Systems and Control

Courses:

- **Computer Vision, Computer Graphics, Machine Learning**
- 3D Photography, Shape Modeling and Geometry Processing
- Prob. Graphical Models for Image Analysis, Prob. AI, etc.
- Physically-Based Simulation, Game-Programming Lab, CV Lab
- Multi-Media Communication, Computational Vision
- Mathematical Foundations of Computer Graphics and Vision

Do not hesitate to contact us for projects!

Next Thursday:

Radon transform!