Computational Vision

Learning invariances

- Finish invariance learning
- Start texture classification

Hypothesis

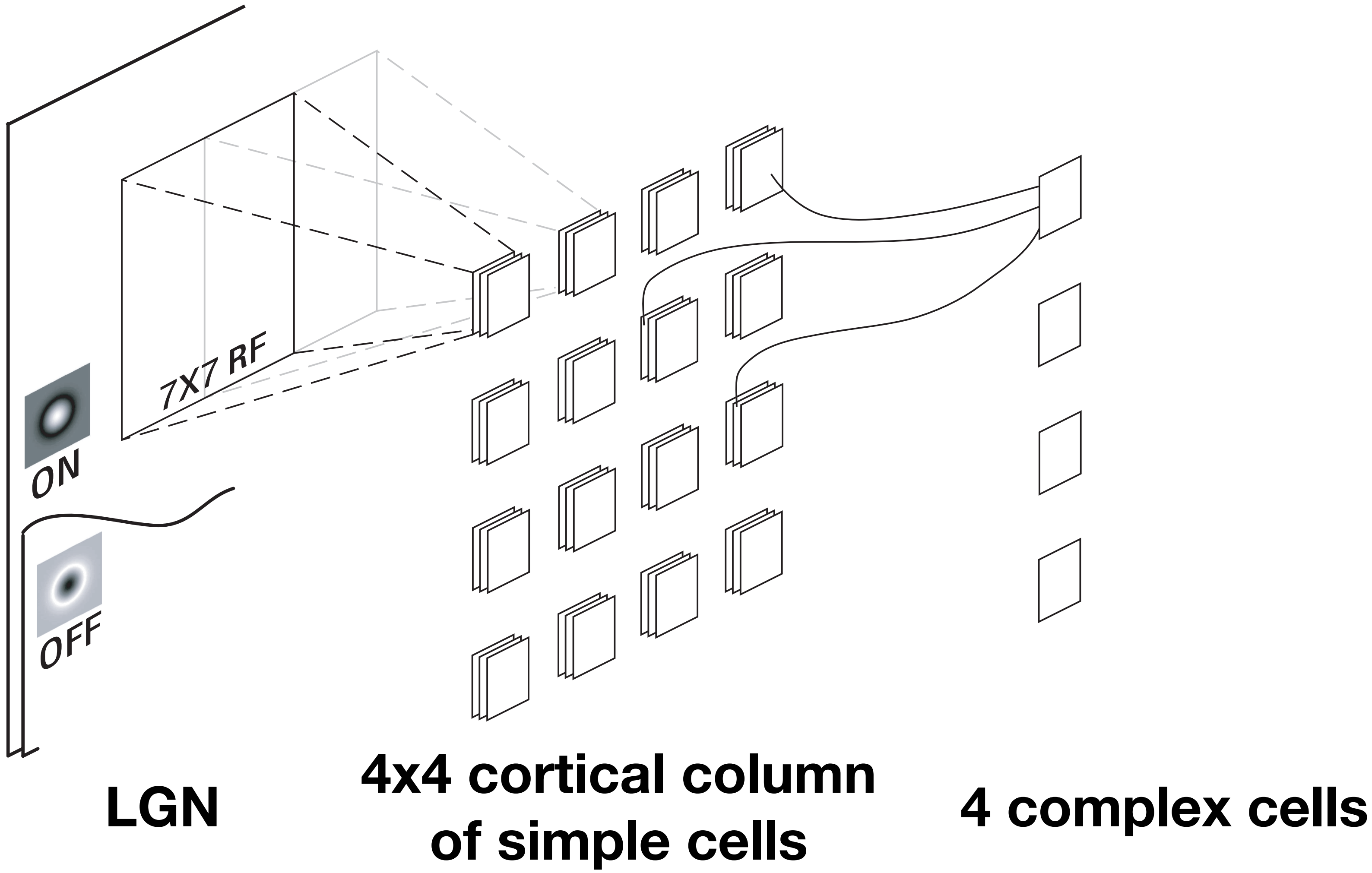
- Simple cells correspond to learning correlations in space
- Complex cells correspond to learning correlations in time

'What fires together, wires together'

'What fires together in close temporal proximity, wires together'

Masquelier et al 2007 (see also Foldiak 1991)

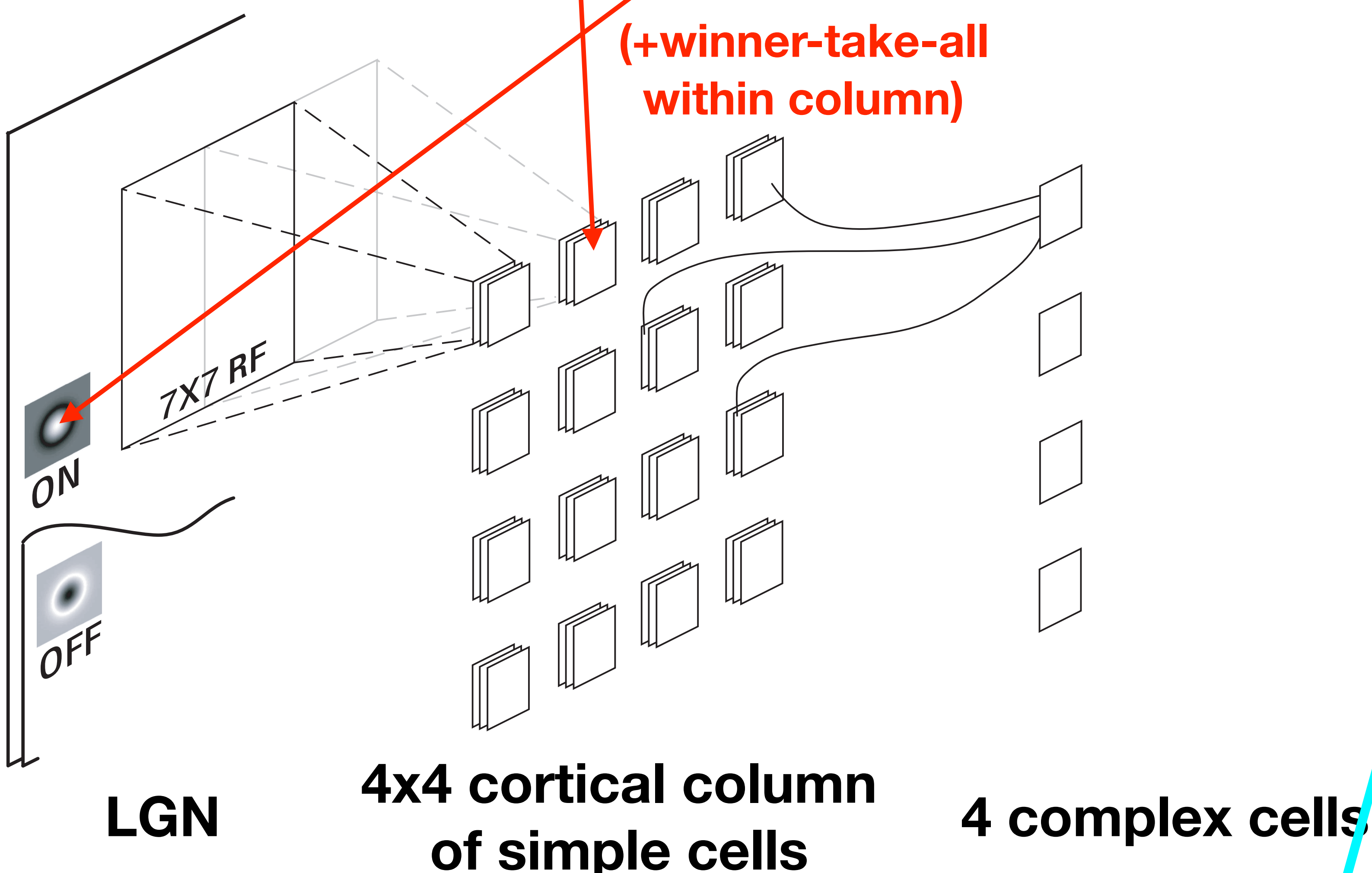
Learning invariances from temporal continuity

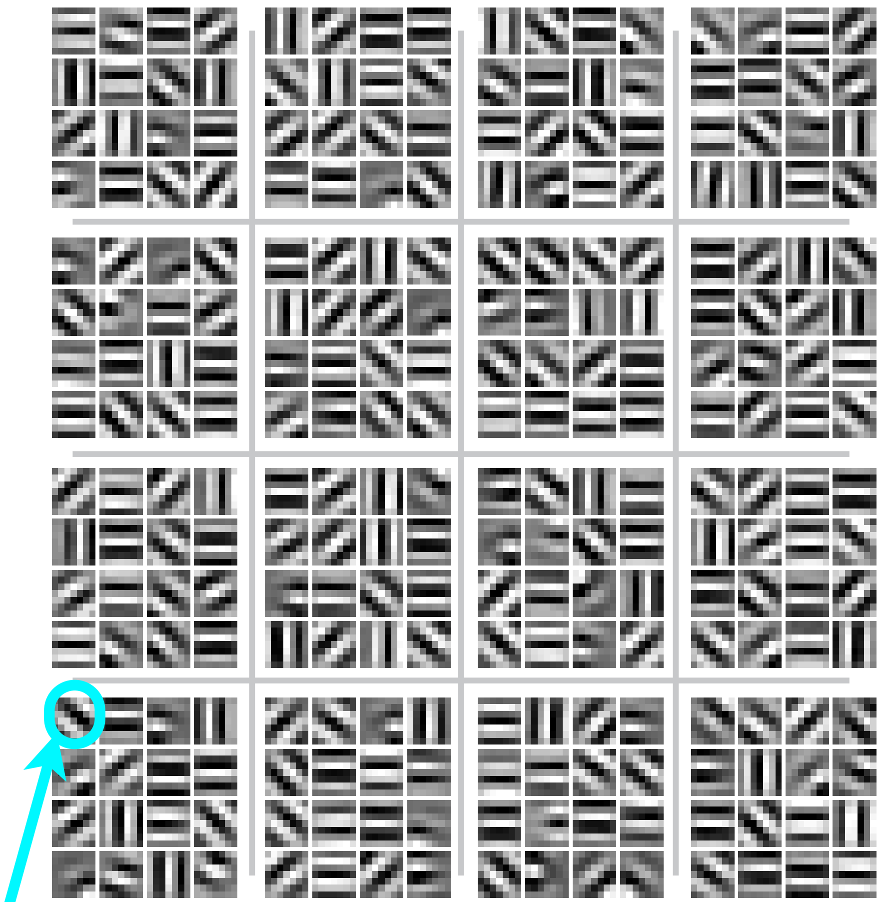


Masquelier et al 2007 (see also Foldiak 1991)

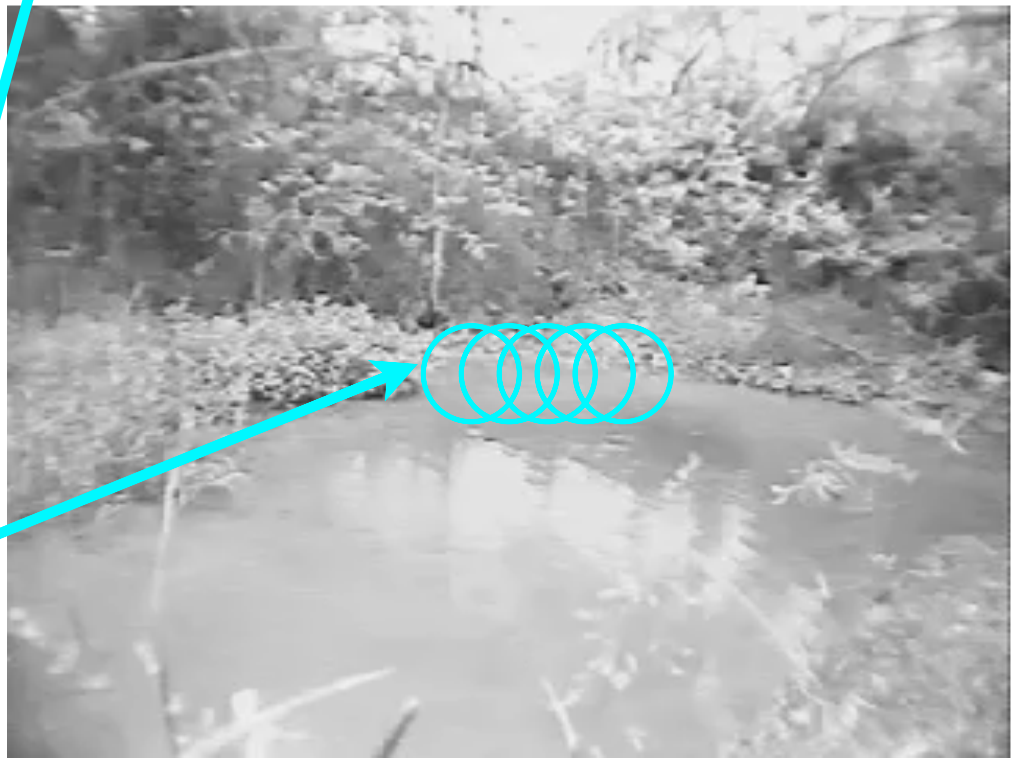
Learning invariances from temporal continuity

$$\Delta \boldsymbol{w} = \alpha \cdot y \cdot (\boldsymbol{x} - \boldsymbol{w})$$





simple cells

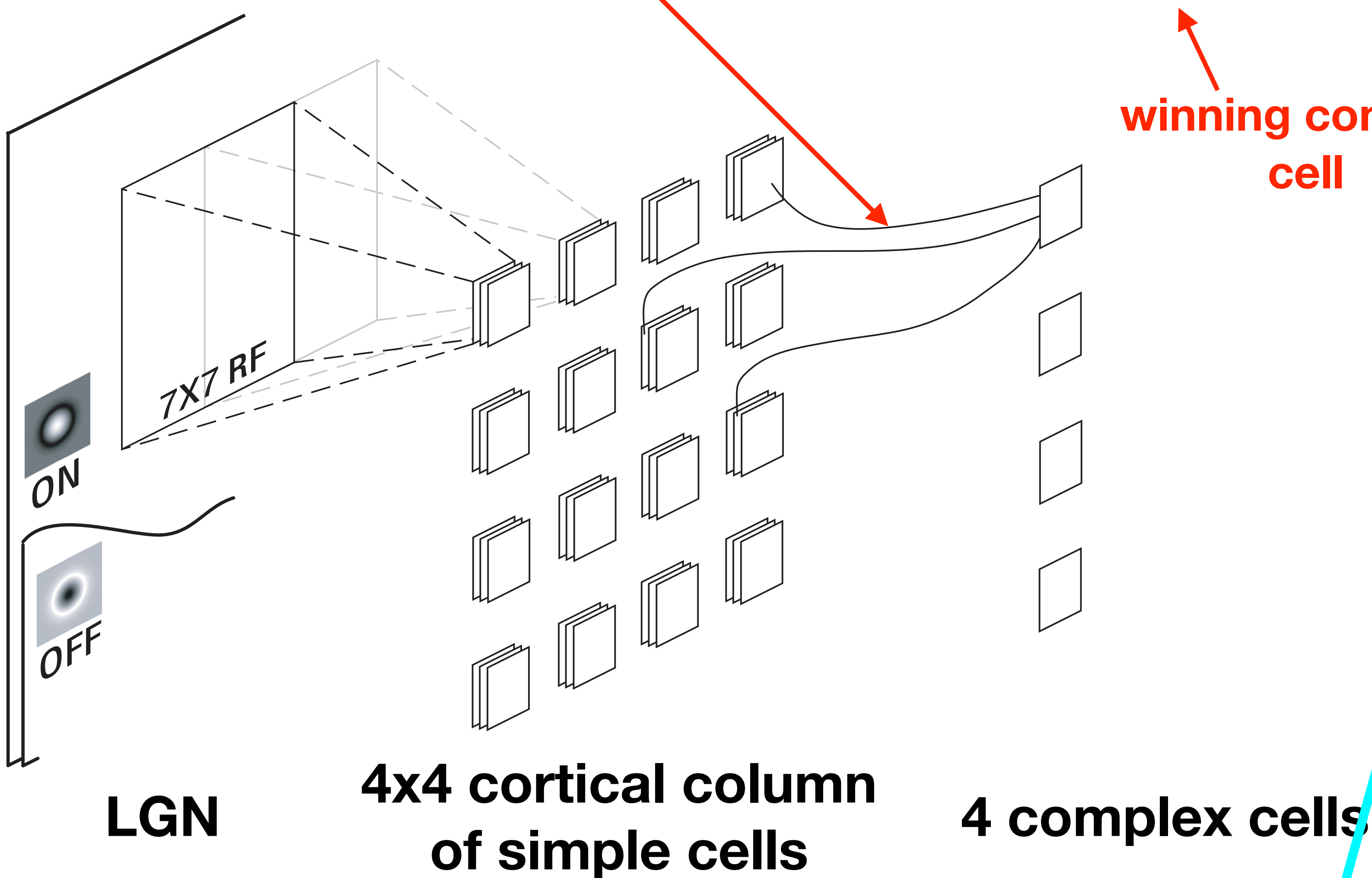


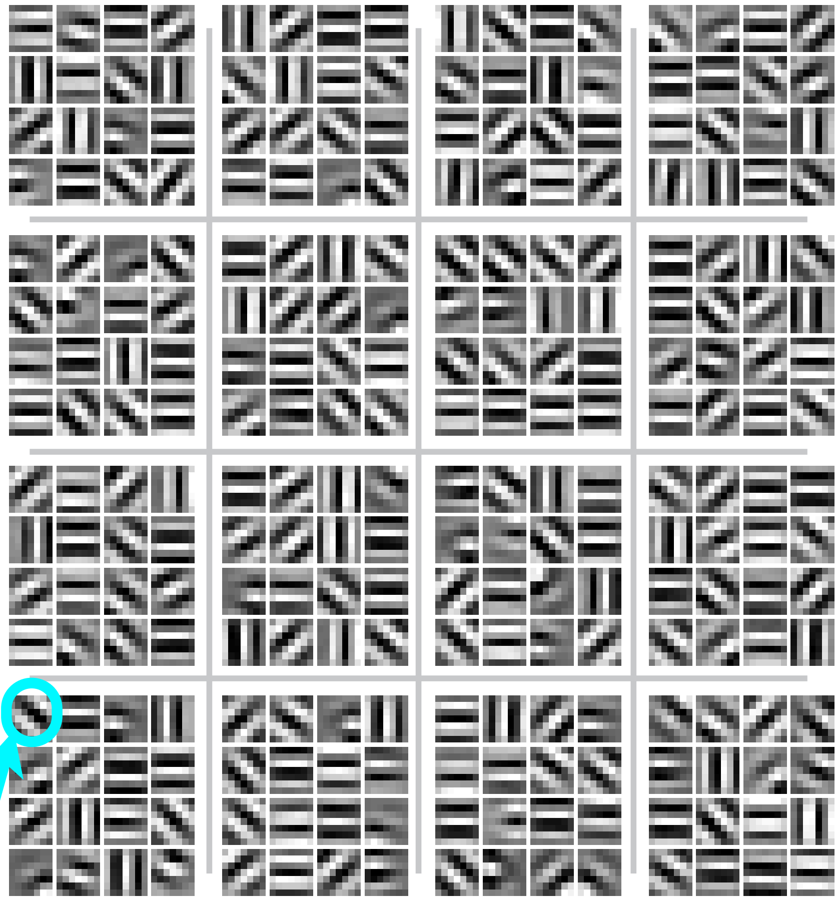
Masquelier et al 2007 (see also Foldiak 1991)

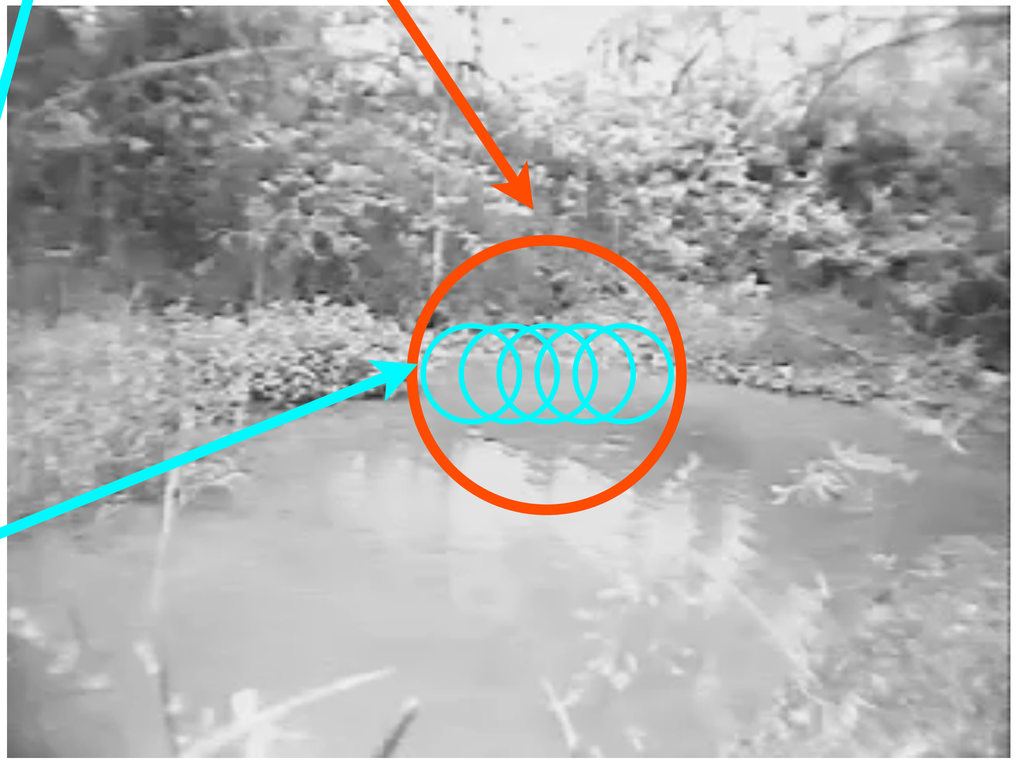
movie courtesy Wolfgang Einhauser

Learning invariances from temporal continuity

$$\Delta w_{iJ_{t-\Delta t}} = \begin{cases} a^{+} \cdot w_{iJ_{t-\Delta t}} \cdot (1 - w_{iJ_{t-\Delta t}}) & \text{if } i = I_t \\ a^{-} \cdot w_{iJ_{t-\Delta t}} \cdot (1 - w_{iJ_{t-\Delta t}}) & \text{otherwise.} \end{cases}$$





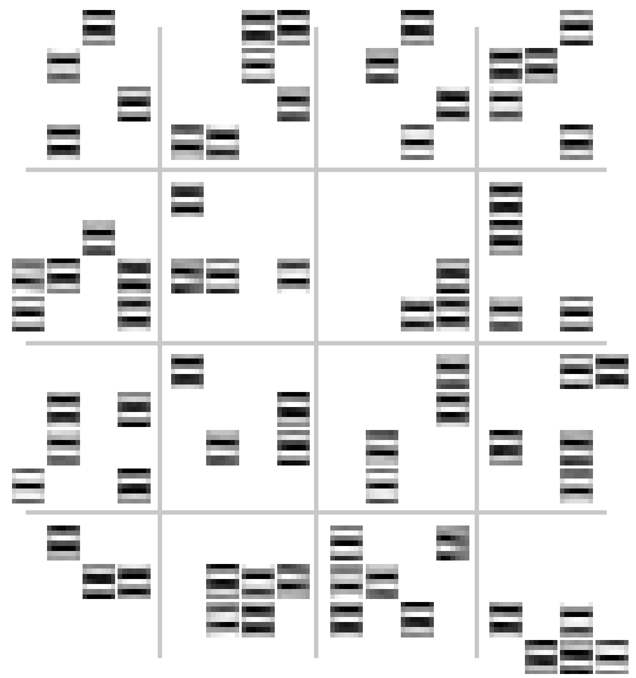


Masquelier et al 2007 (see also Foldiak 1991)

movie courtesy Wolfgang Einhauser

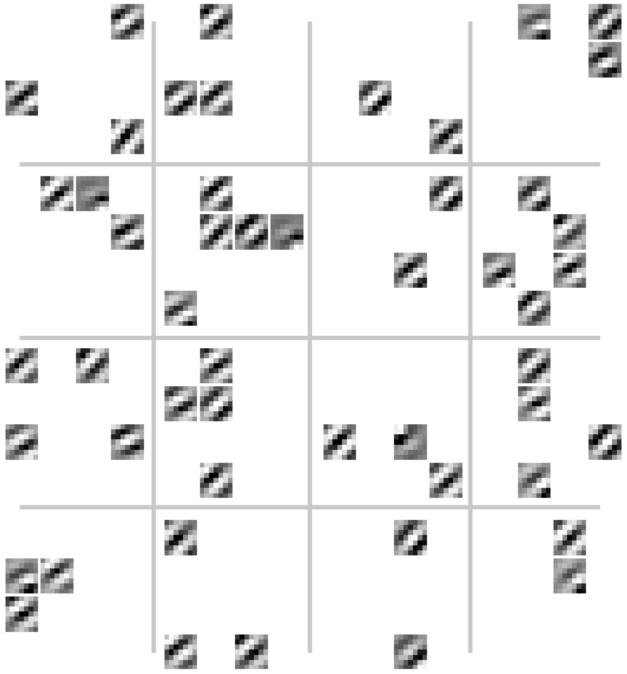
Masquelier et al 2007

Learning the invariance from temporal continuity

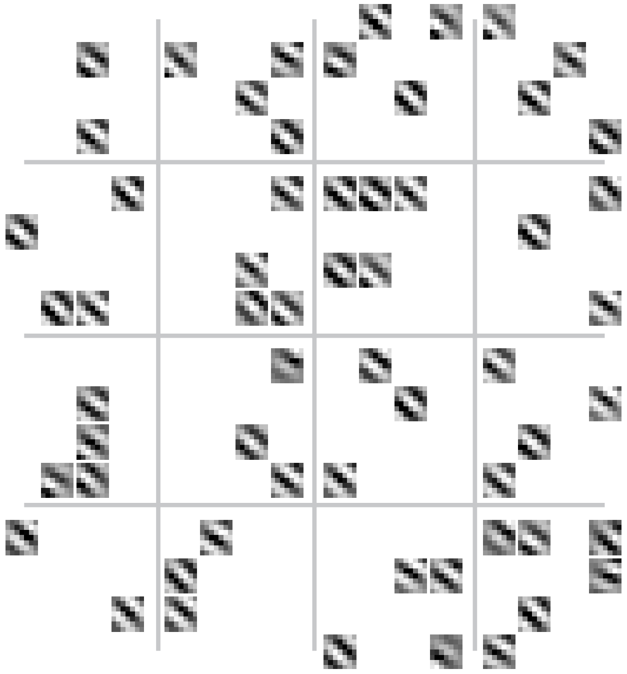


(a) S_1 units (n=73) that remain connected to C_1 unit # 1 after learning

(b) S_1 units (n=35) that remain connected to C_1 unit # 2 after learning



(c) S_1 units (n=59) that remain connected to C_1 unit # 3 after learning



(d) S_1 units (n=38) that remain connected to C_1 unit # 4 after learning

Slow feature analysis

$$\Delta_j := \Delta(y_j) := \langle \dot{y}_j^2 \rangle \quad \textit{is minimal} \tag{1}$$

under the constraints

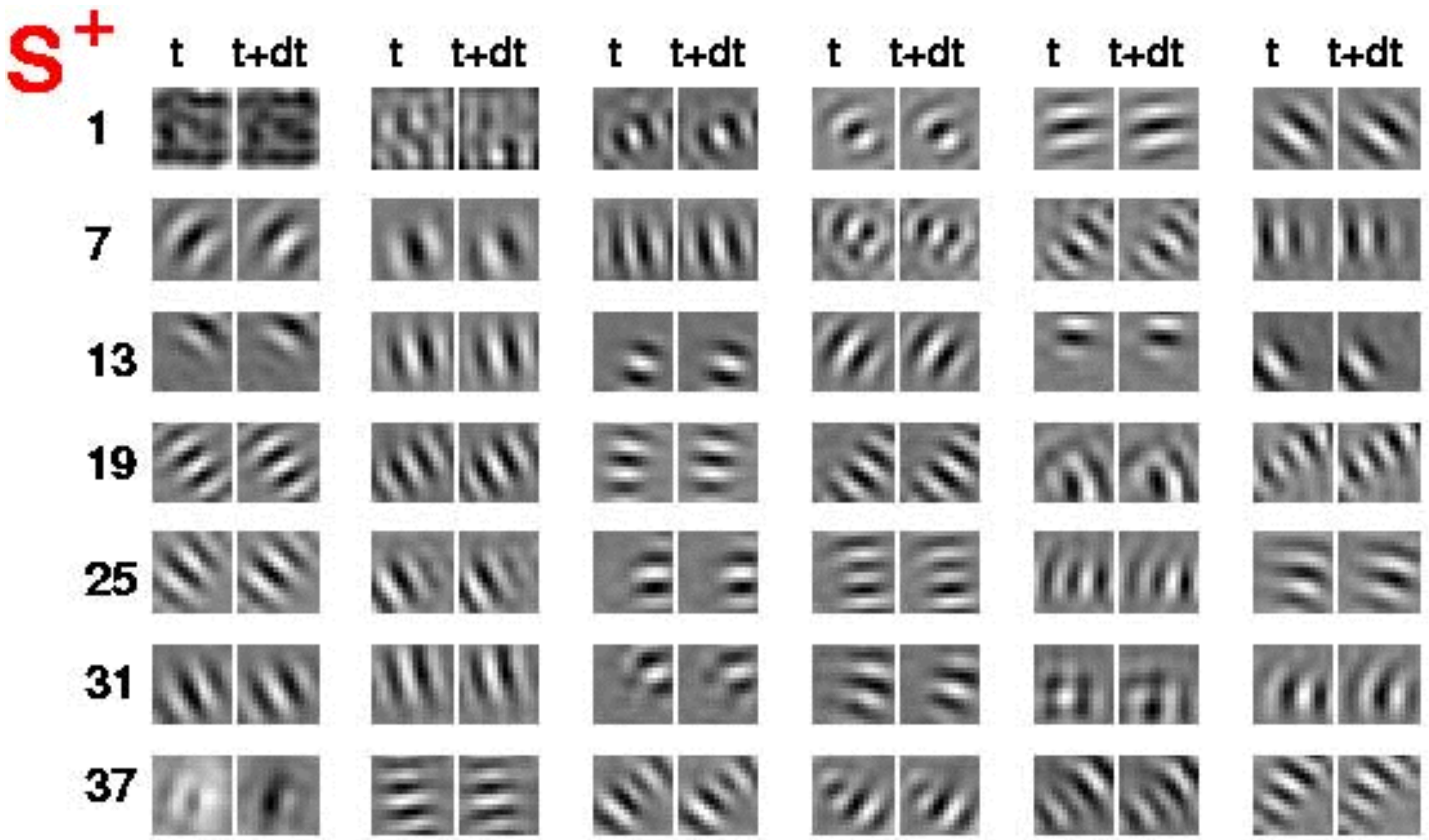
$$\begin{aligned}
\langle y_j \rangle &= 0 \qquad \textit{(zero mean)}\ , & (2)\\
\langle y_j^2 \rangle &= 1 \qquad \textit{(unit variance)}\ , & (3)\\
\forall\, j' < j: \quad \langle y_{j'}\, y_j \rangle &= 0 \qquad \textit{(decorrelation)}\ , & (4)
\end{aligned}$$

where the angle brackets indicate temporal averaging, i.e. $\langle f \rangle := \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t)\, dt$.

$$y_j(t) = g_j(\mathbf{x}(t)) = \mathbf{w}_j^T \mathbf{h}(\mathbf{x}(t)) = \mathbf{w}_j^T \mathbf{z}(t)$$
$$\Delta(y_j) = \langle \dot{y}_j^2 \rangle = \mathbf{w}_j^T \langle \dot{\mathbf{z}} \dot{\mathbf{z}}^T \rangle \mathbf{w}_j$$

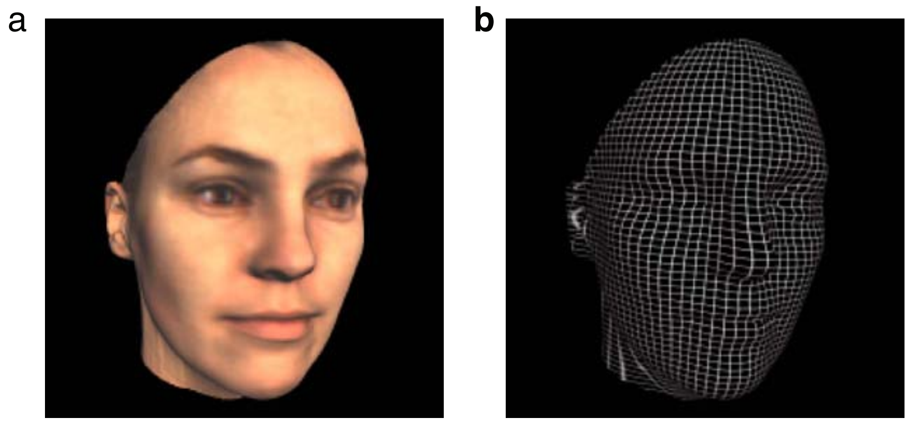
Wiskott & Sejnowski 2002

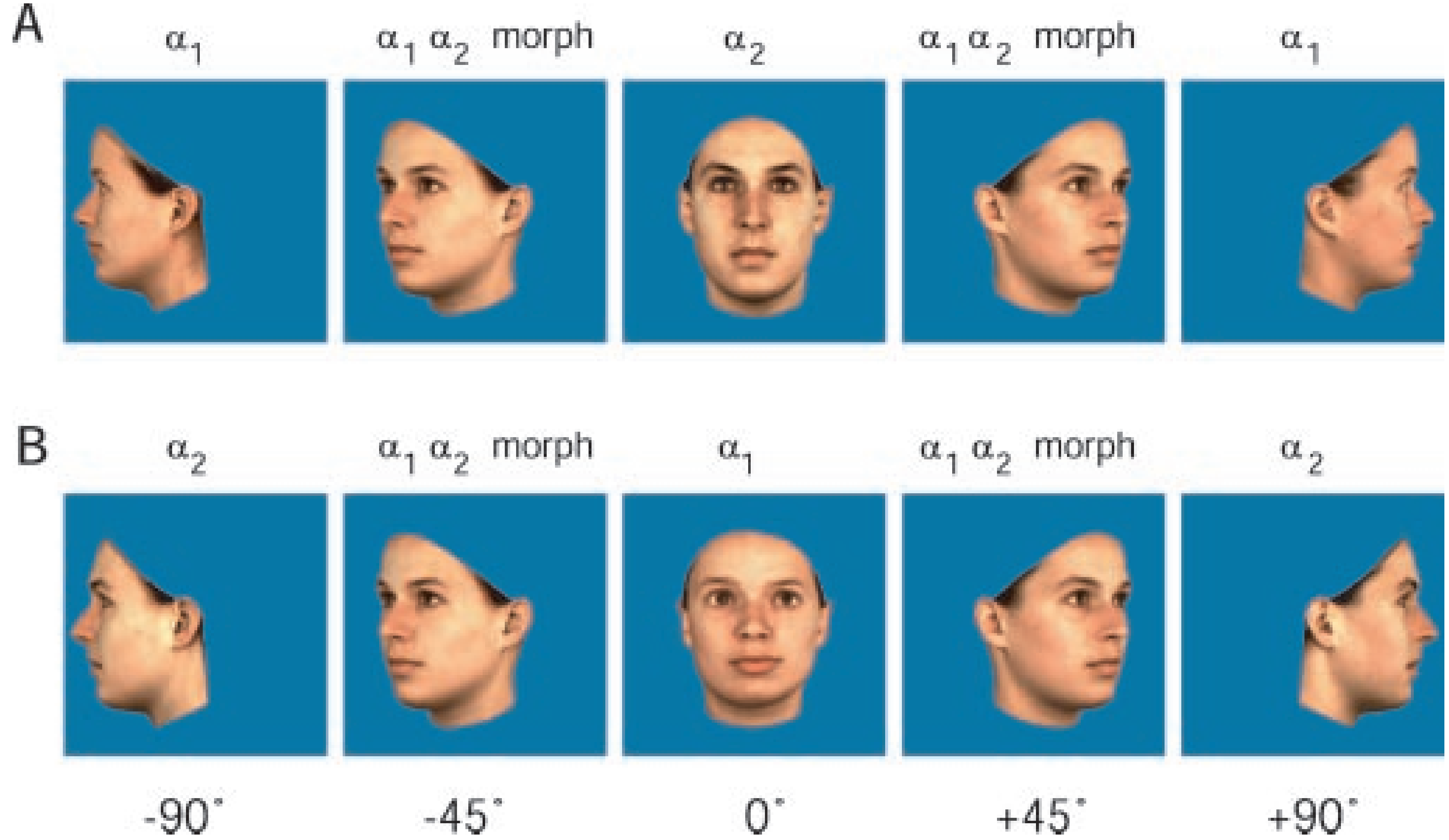
Slow feature analysis



source: scholarpedia

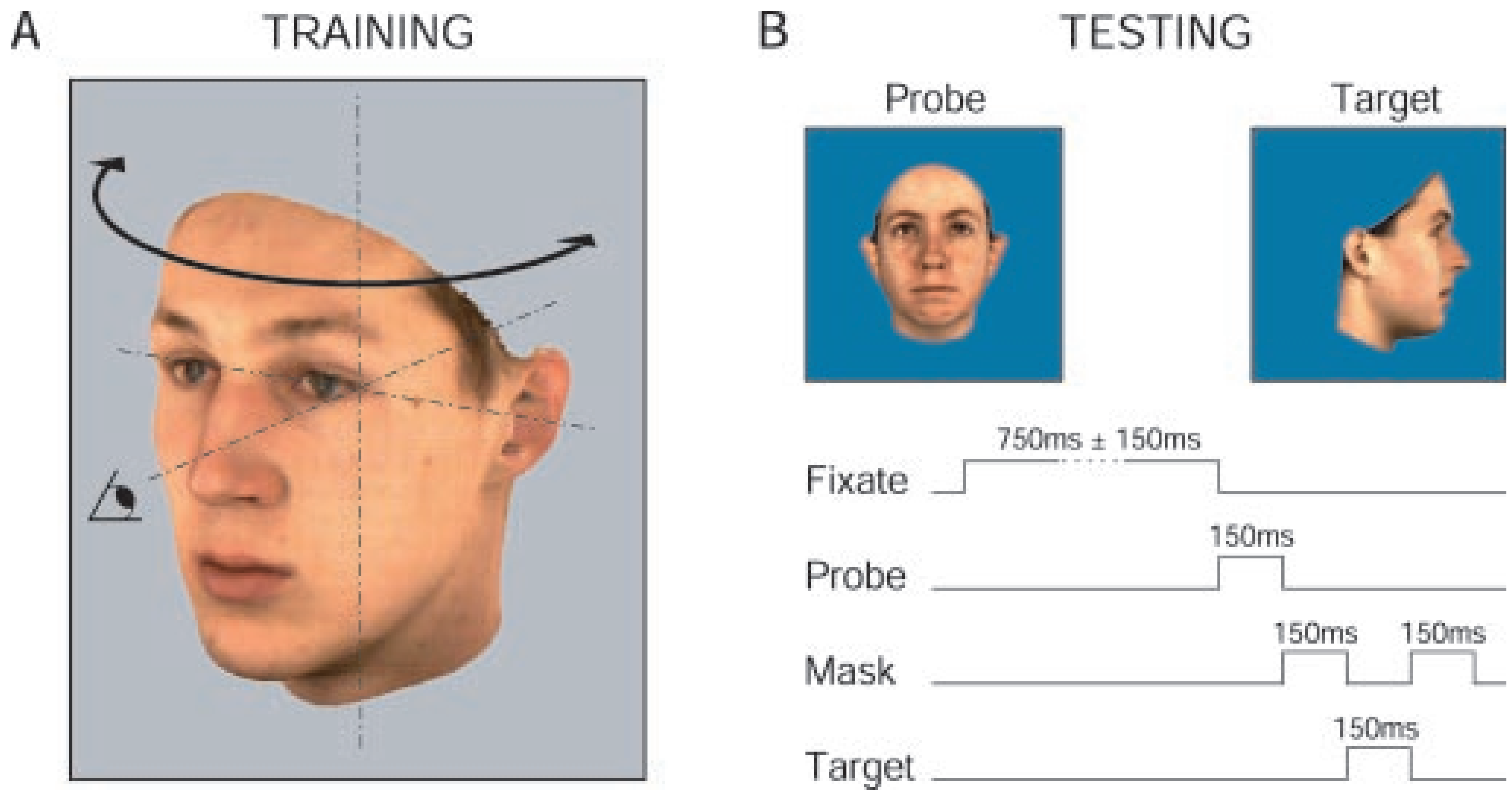
Effects of temporal associations on learning and memory





Wallis & Bulthoff '01

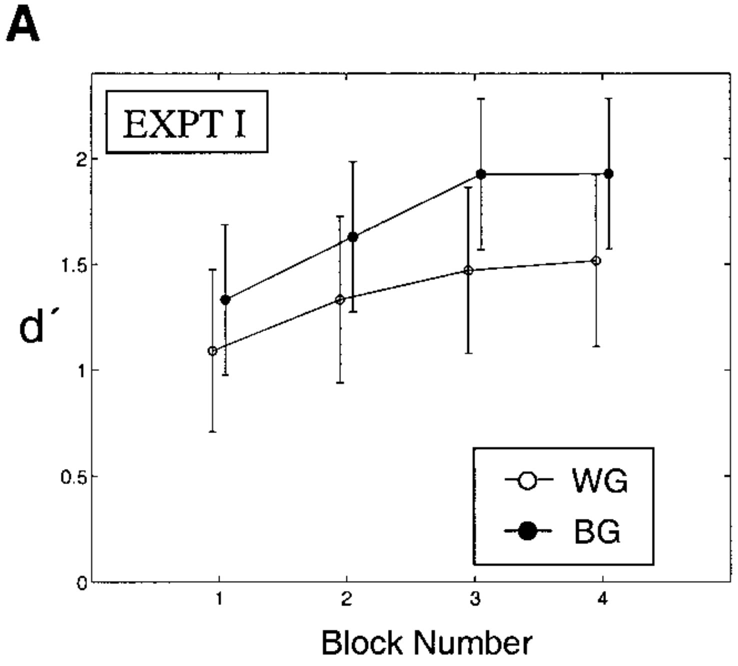
Effects of temporal associations on learning and memory



Wallis & Bulthoff ’01

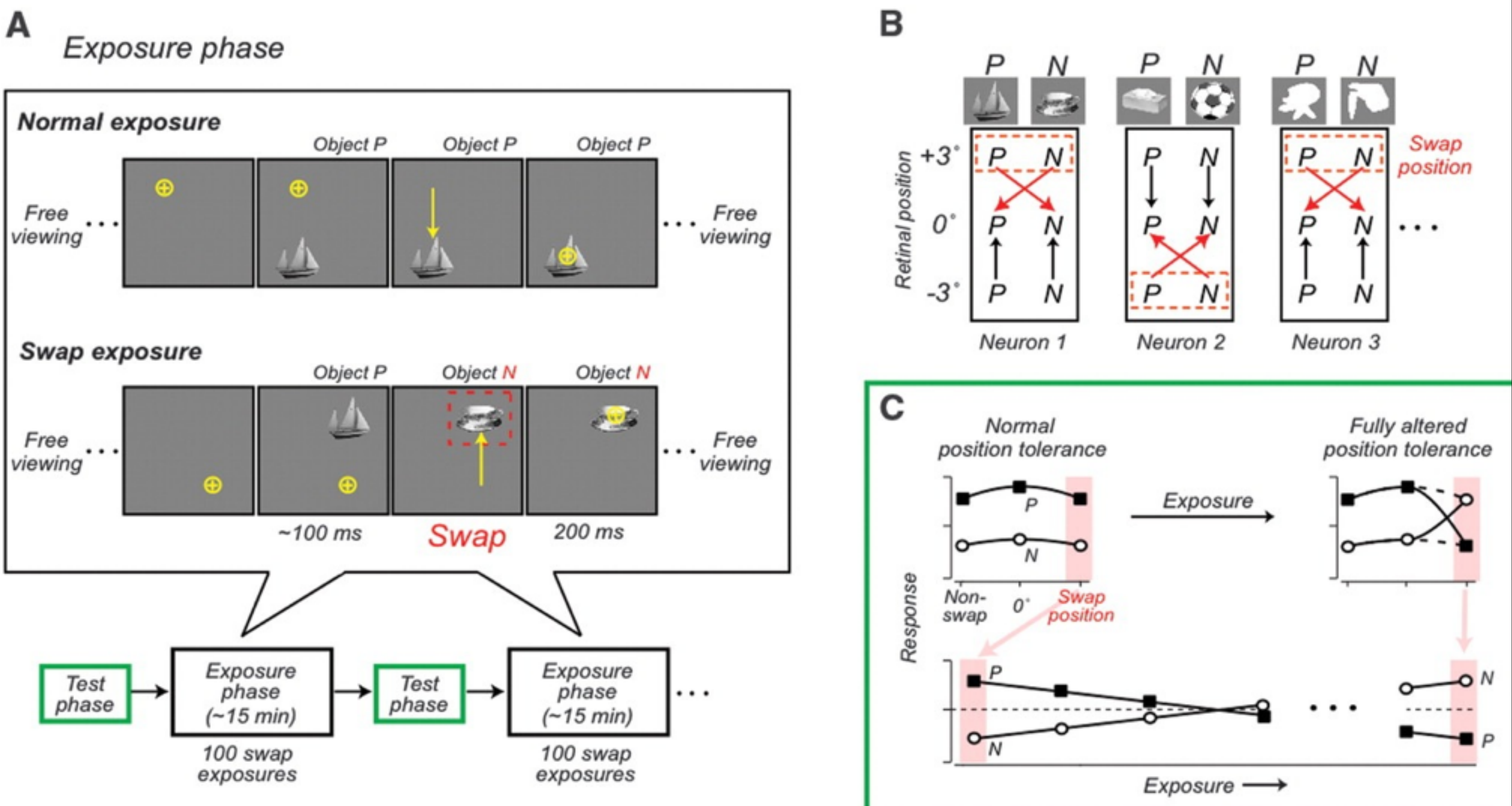
Effects of temporal associations on learning and memory

Discrimination worst for prototypes that are part of the same training sequence



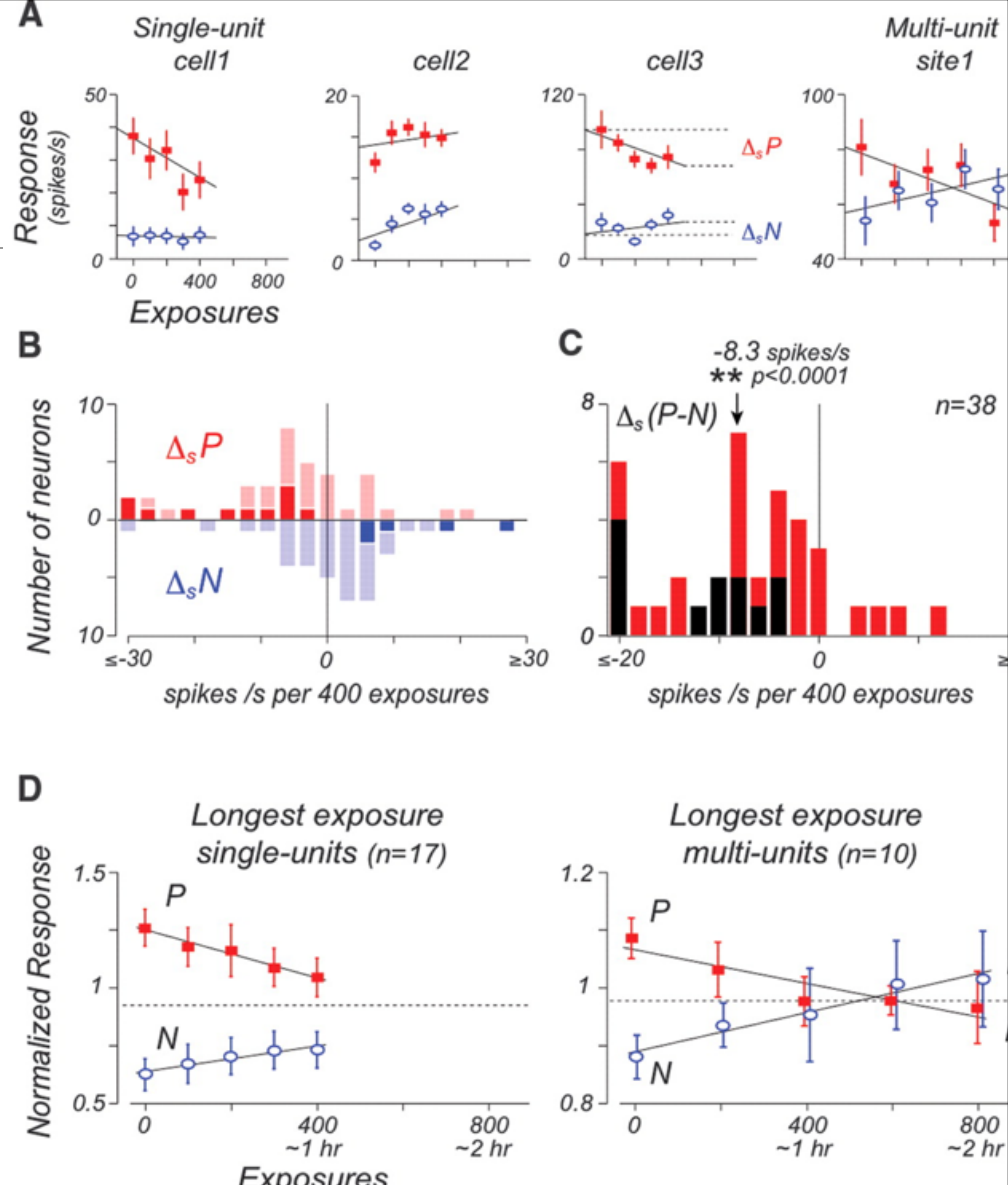
Wallis & Bulthoff '01

Learning in IT



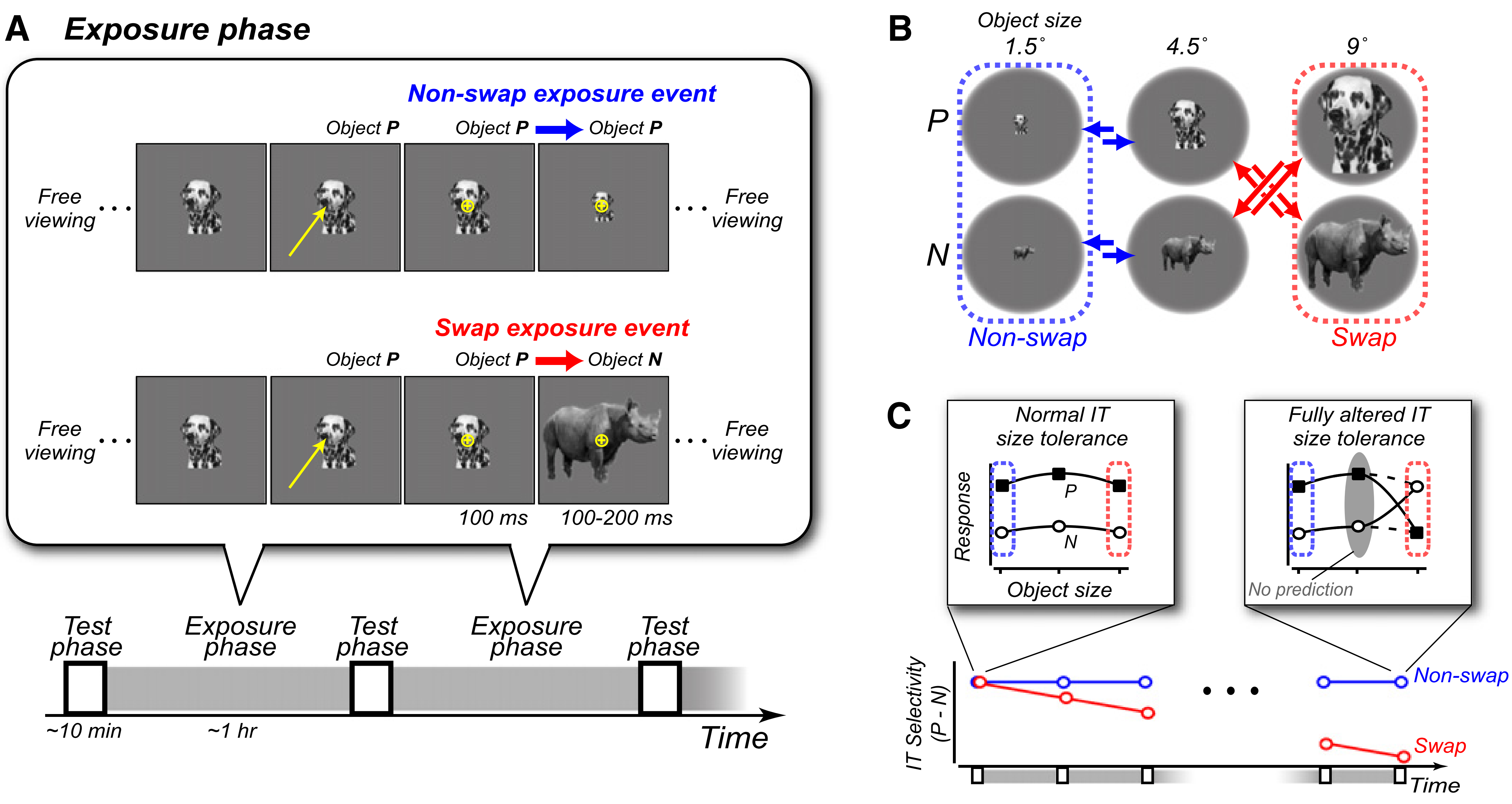
Li & DiCarlo ’08

Learning in IT



Li & DiCarlo '08

Learning in IT



Li & DiCarlo '10

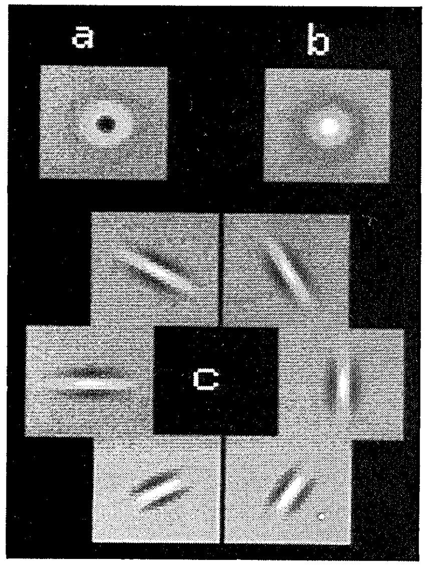
Computational Vision

Learning invariances

- Finish invariance learning
- Start texture classification

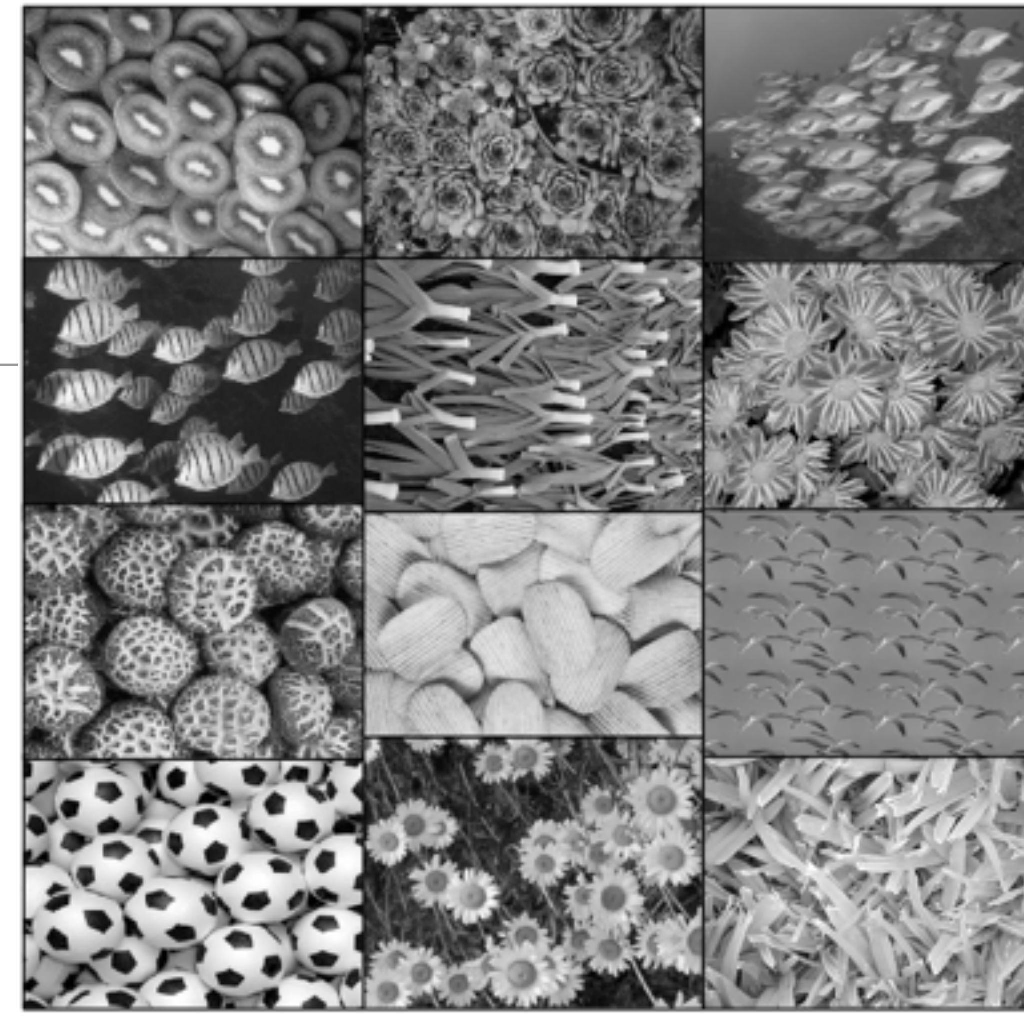
So far in this class...

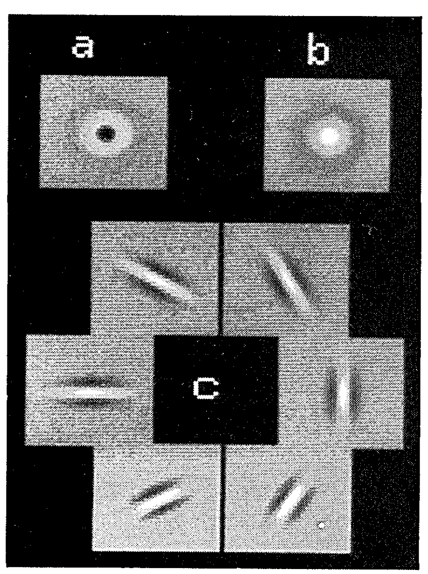
- The building blocks of vision:
  - spatial filters: edges, blobs, bars, etc
  - spatio-chromatic filters: color
  - spatio-temporal filters: motion (soon)



From early to mid-level vision

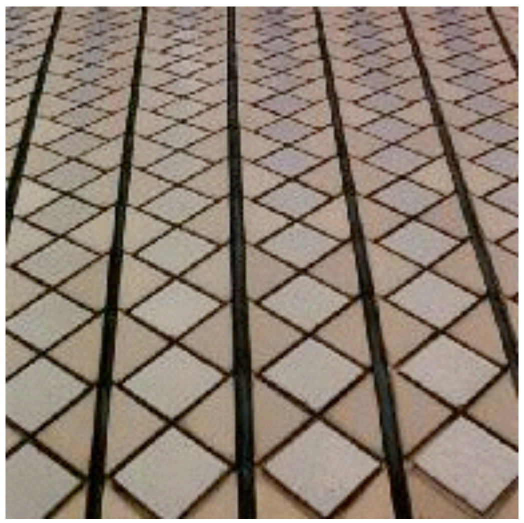
- Surface / 3D shape
- Texture
- Material properties
- etc

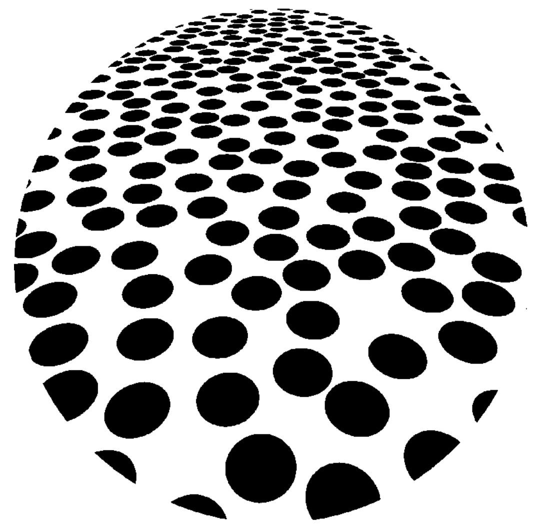




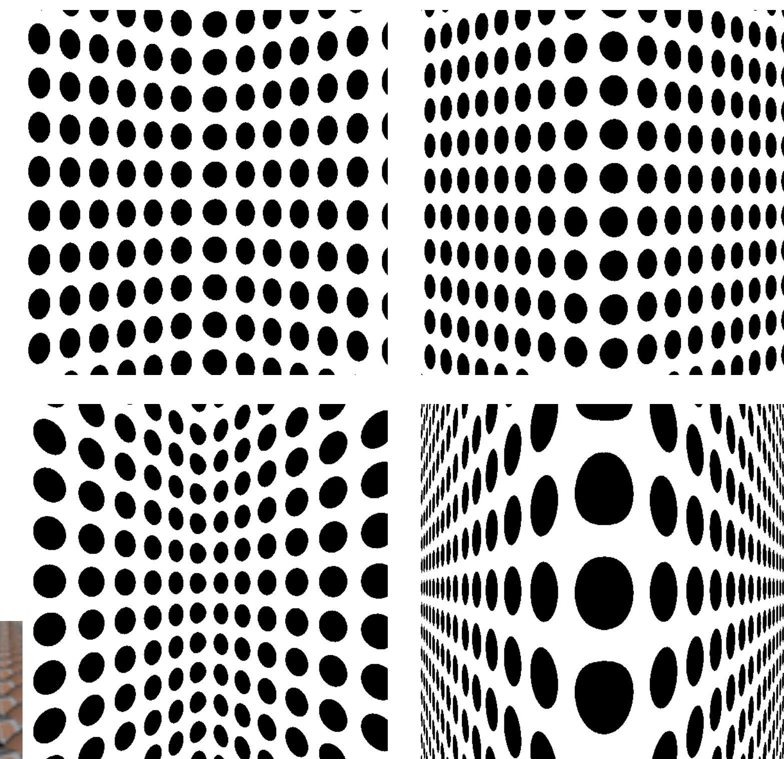
?

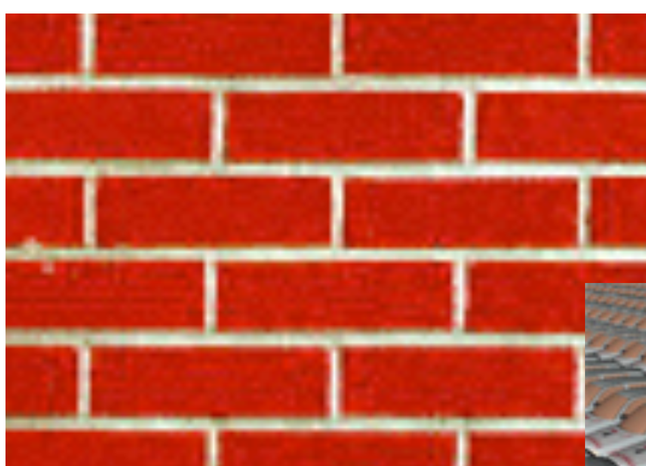
On textures

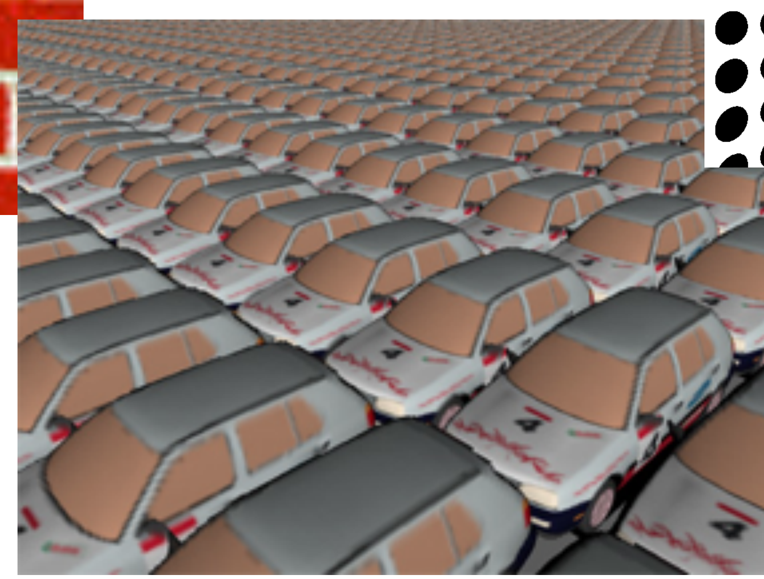




Shape/structure from texture





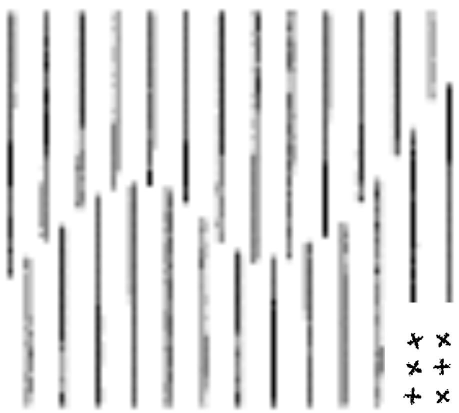


On textures

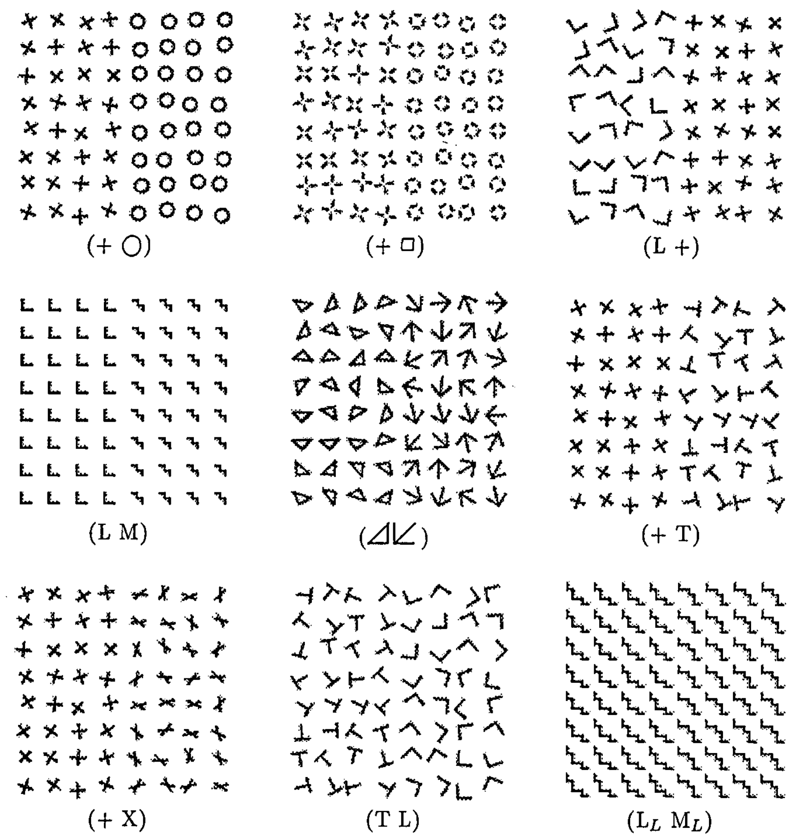
Affordances

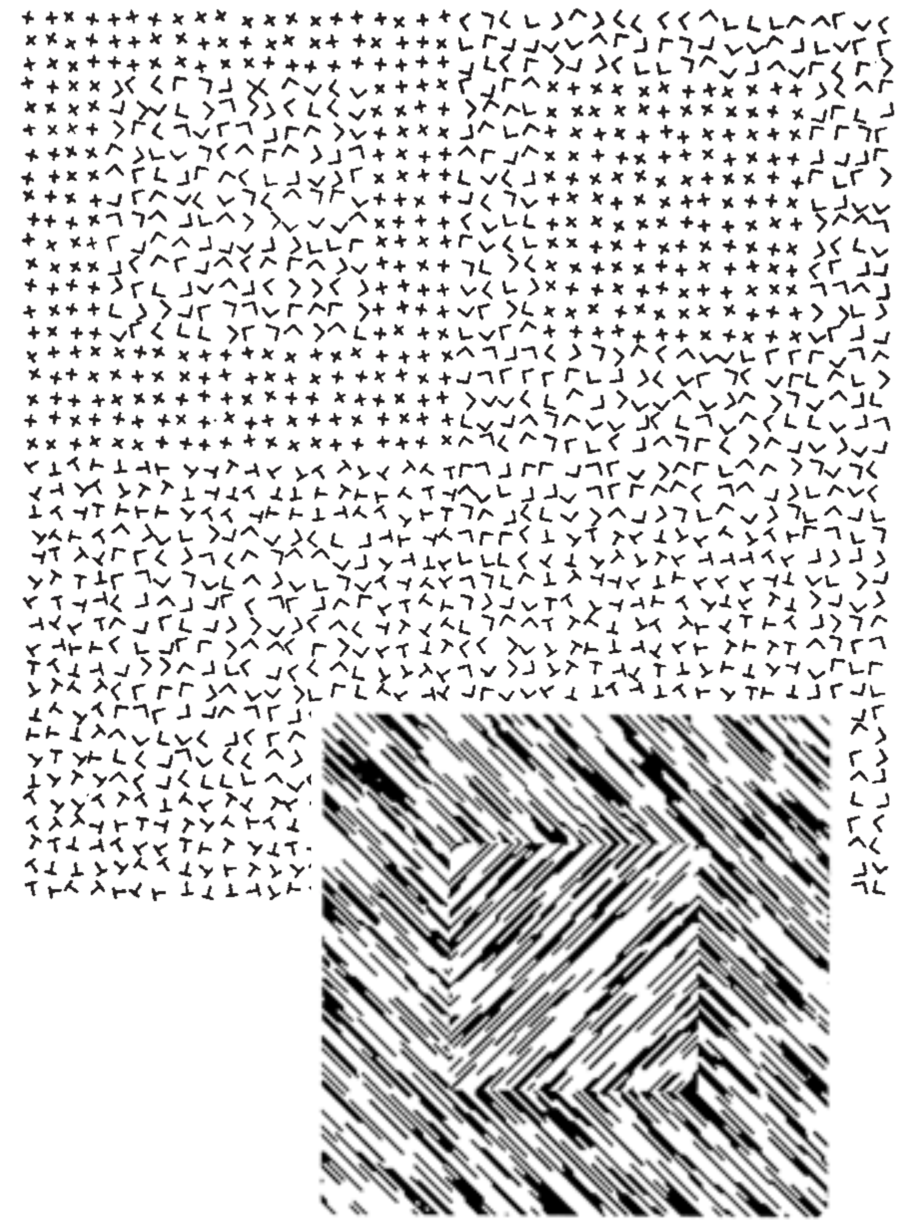
Ted Adelson (MIT)

Textures in the lab



A



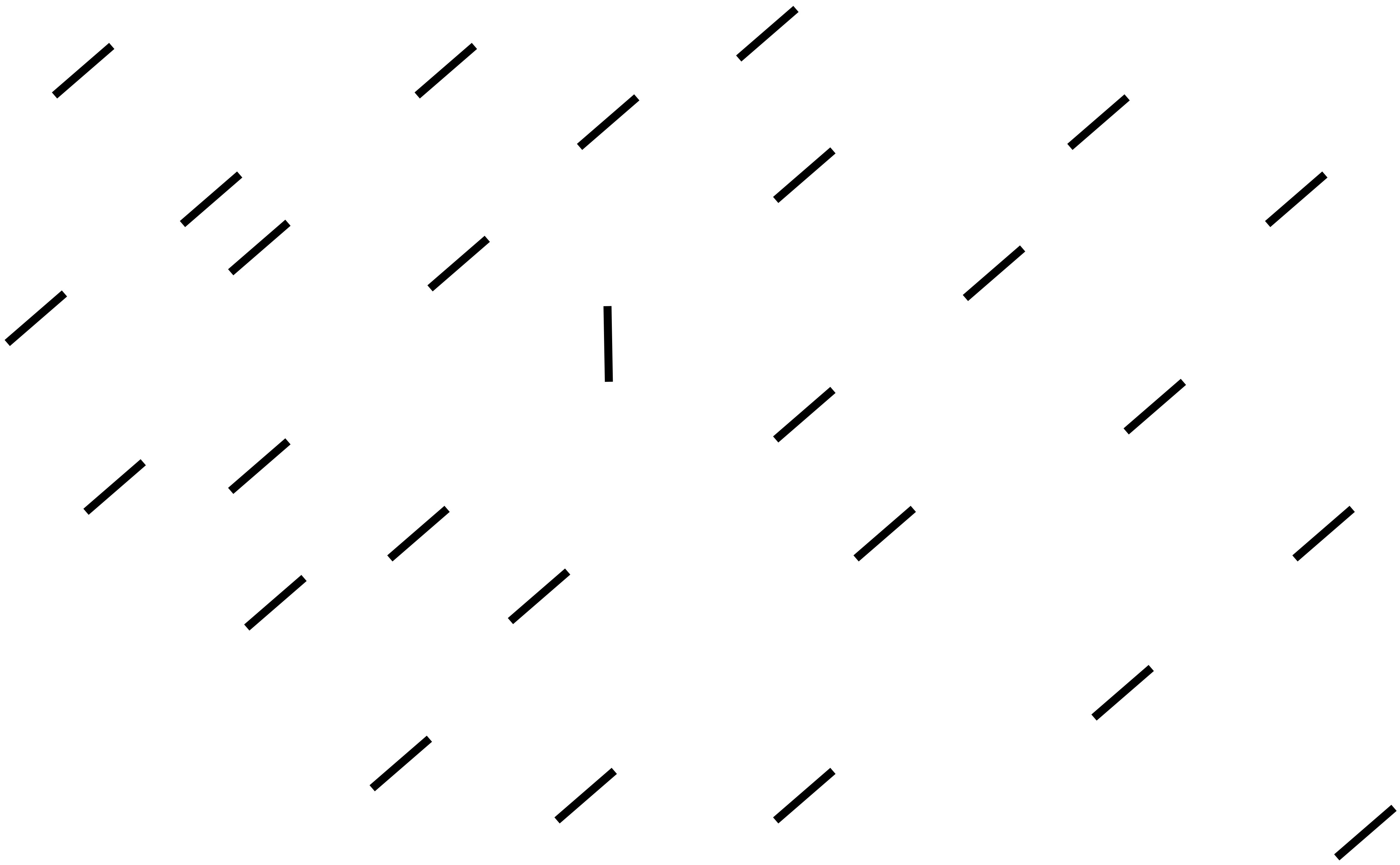


Attentive vs. pre-attentive vision: Searches

Find this 

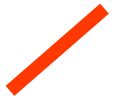
Treisman, Wolfe and many others

Attentive vs. pre-attentive vision: Searches



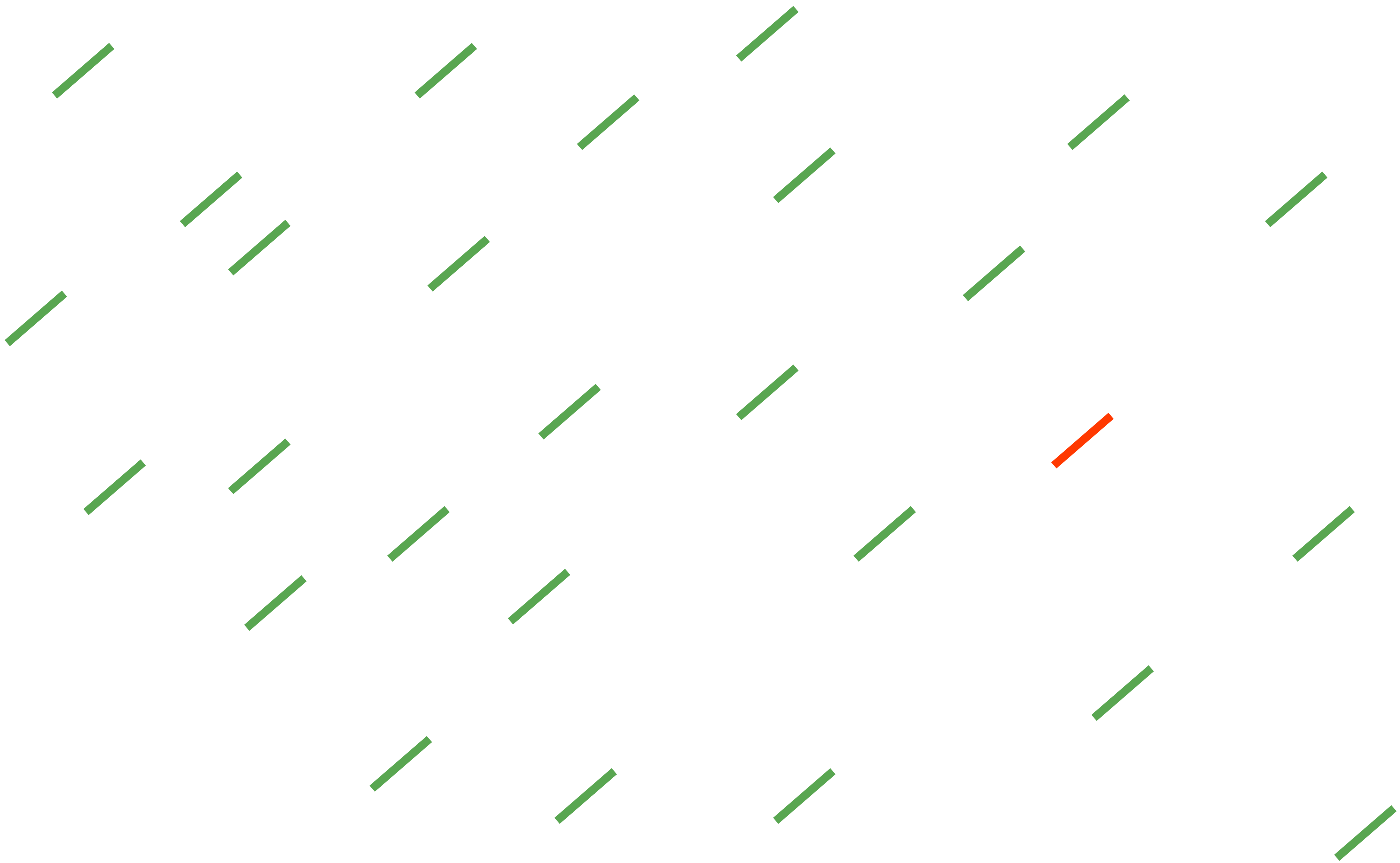
Treisman, Wolfe and many others

Attentive vs. pre-attentive vision: Searches

Find this 

Treisman, Wolfe and many others

Attentive vs. pre-attentive vision: Searches



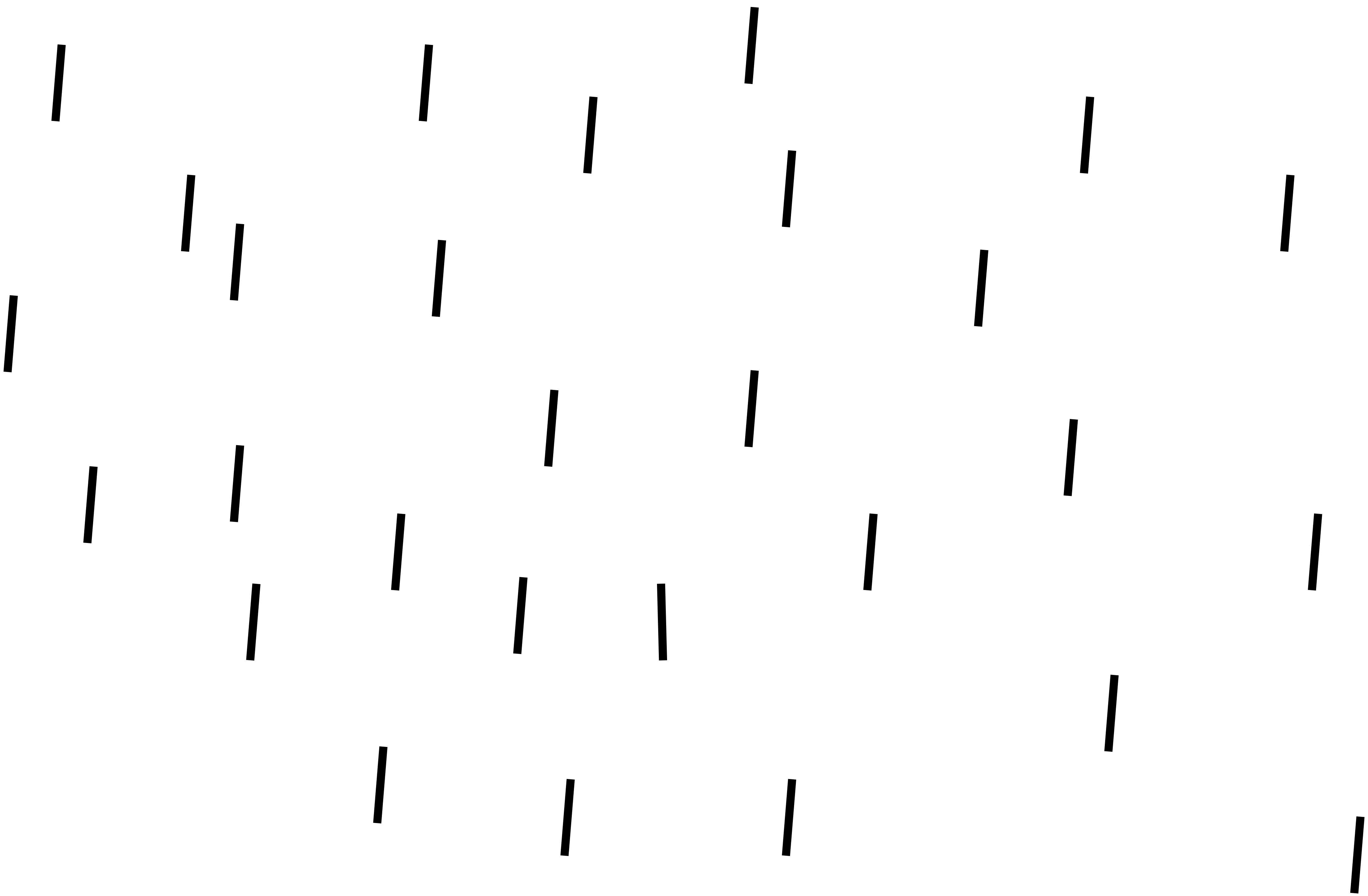
Treisman, Wolfe and many others

Attentive vs. pre-attentive vision: Searches

Find this 

Treisman, Wolfe and many others

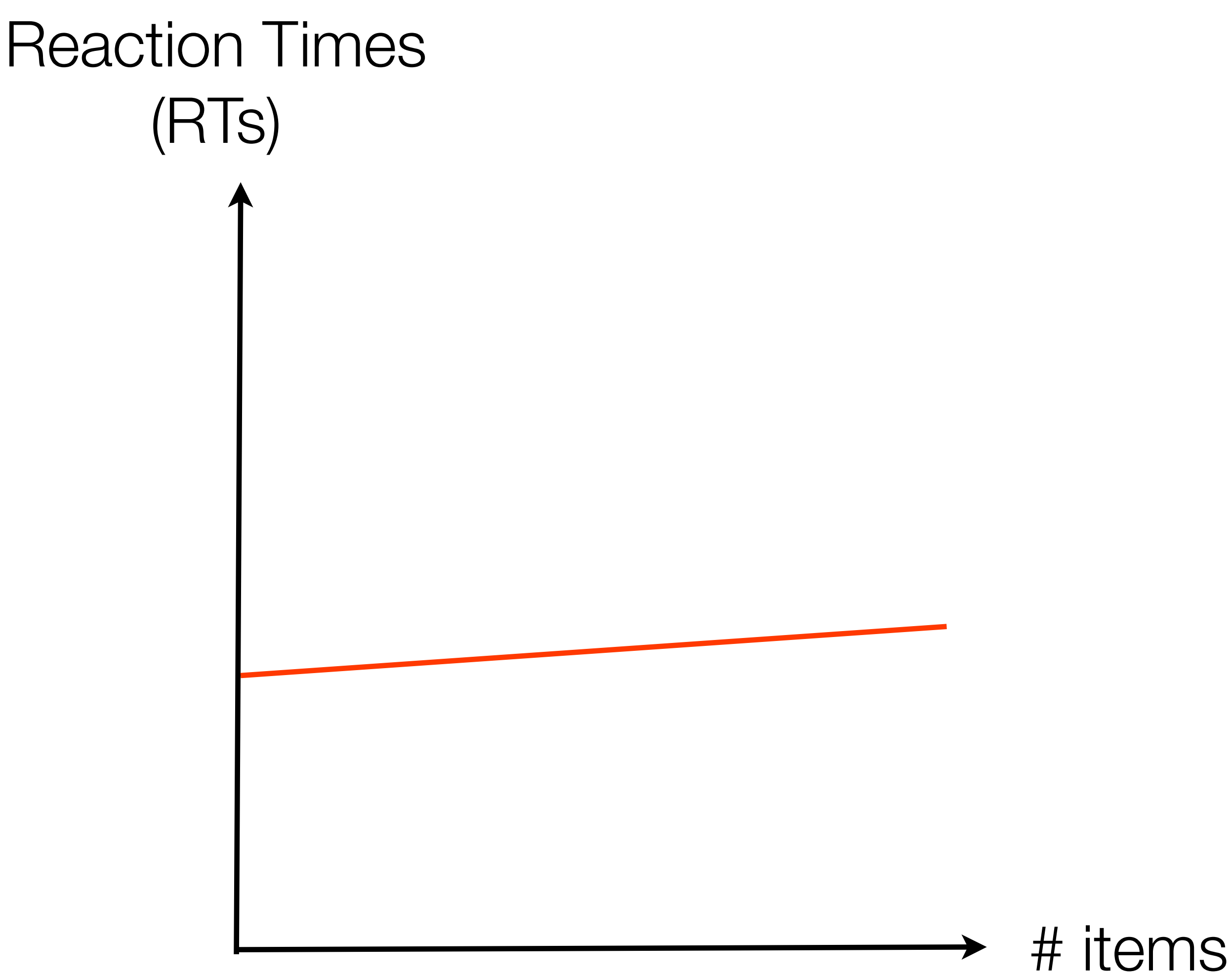
Attentive vs. pre-attentive vision: Searches



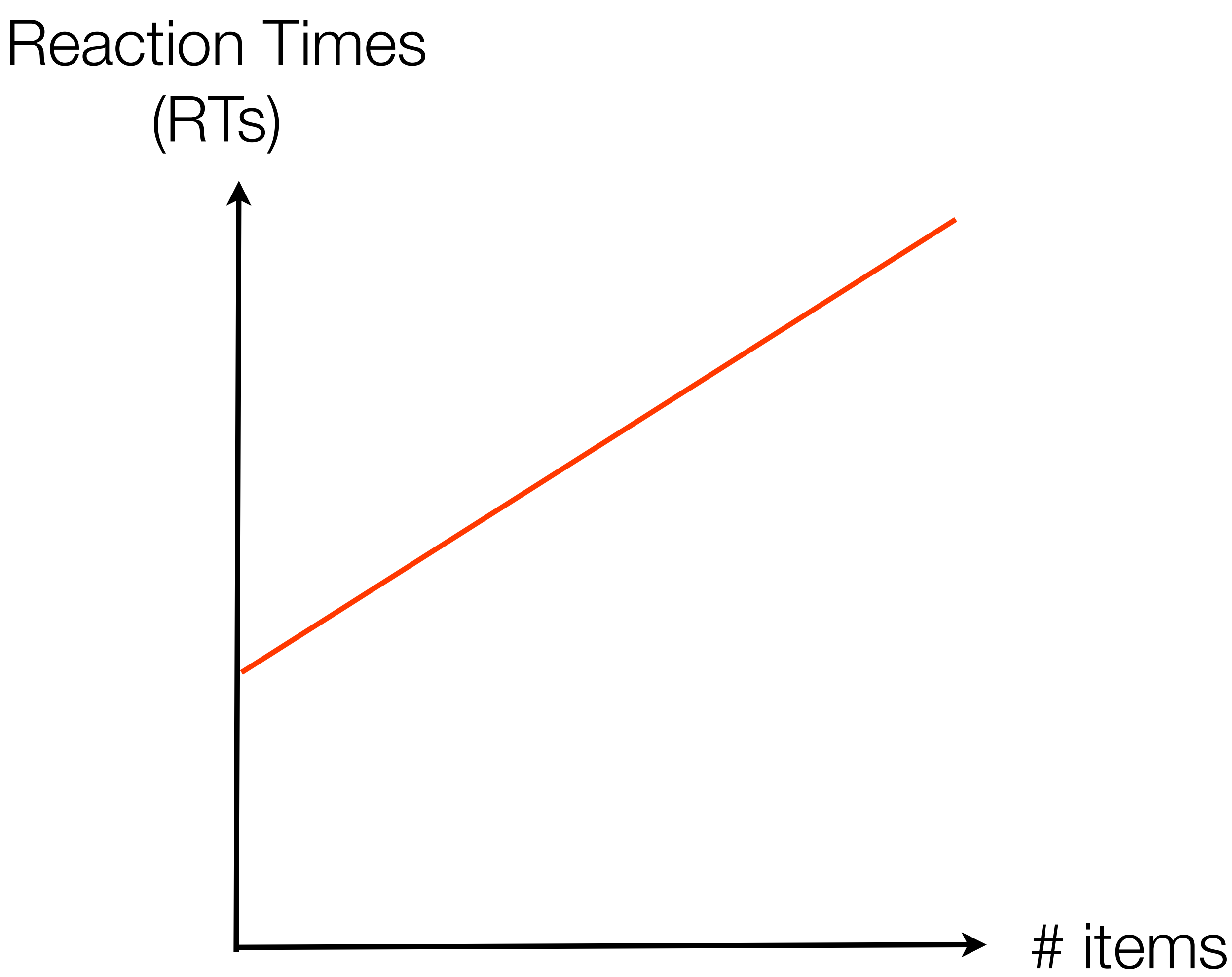
Treisman, Wolfe and many others

Attentive vs. pre-attentive vision: Searches

Parallel / pre-attentive



Serial / attentive



Treisman, Wolfe and many others

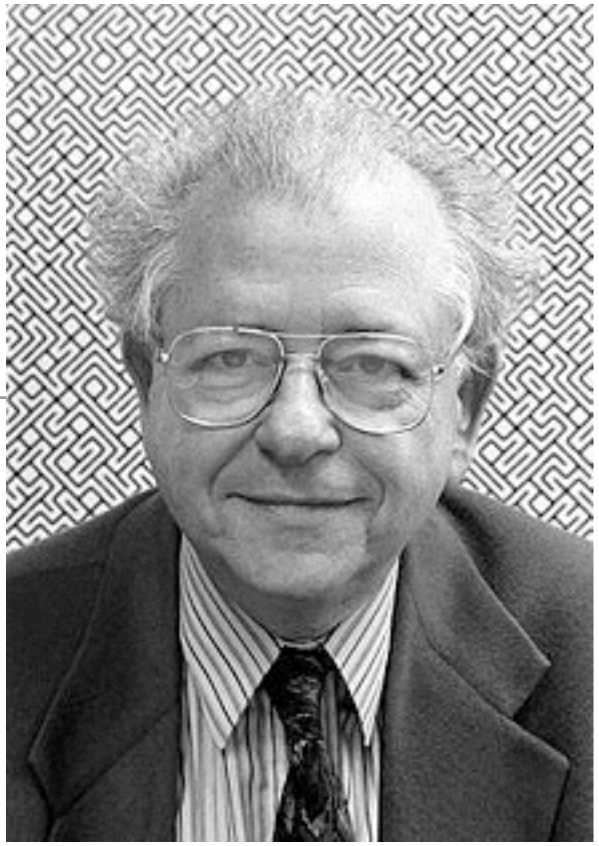
Rapid presentation paradigms

- Ss get the gist of a scene from ultra-rapid image presentations
  - No time for eye movements
  - No top-down / expectations
- Coarse initial ***base*** representation based on parallel processing of arrays of feature detectors

Potter 1971; Biederman 1972; Thorpe et al 1996; Li et al 2002; Evans & Treisman 2005; Serre et al 2007; see Fabre-Thorpe 2011 for review

Texture discrimination

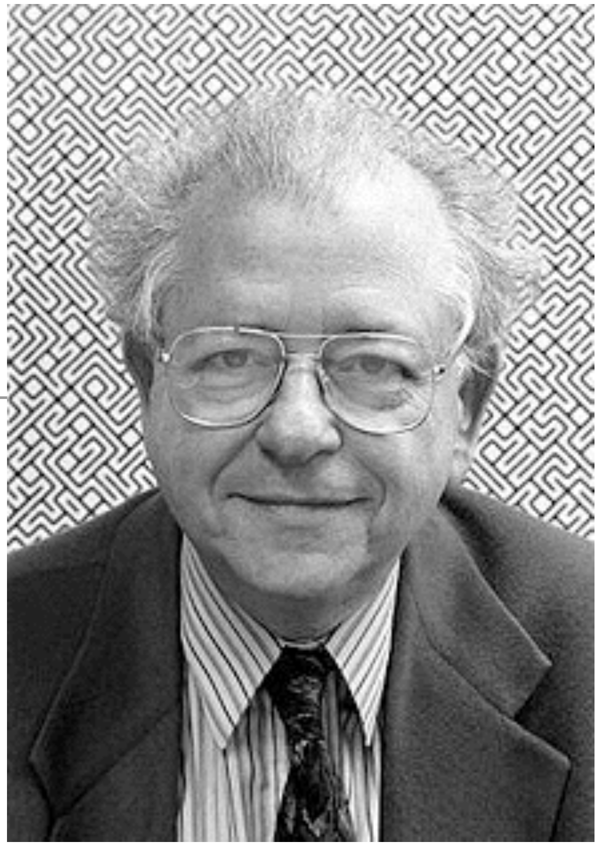
- Pre-attentive texture discrimination possible when basic elements have different first-order statistics (i.e., different contrast distributions)



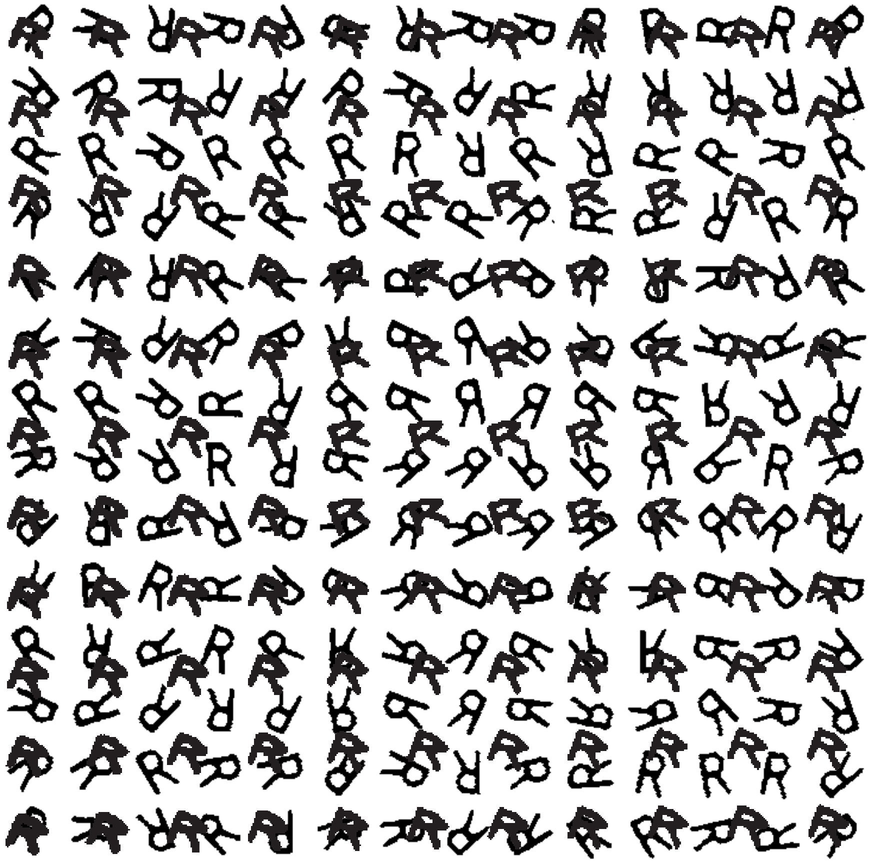
Béla Julesz (1928–2003)

Texture discrimination

- Pre-attentive texture discrimination possible when basic elements have different second-order statistics:
  - different sizes
  - different orientations

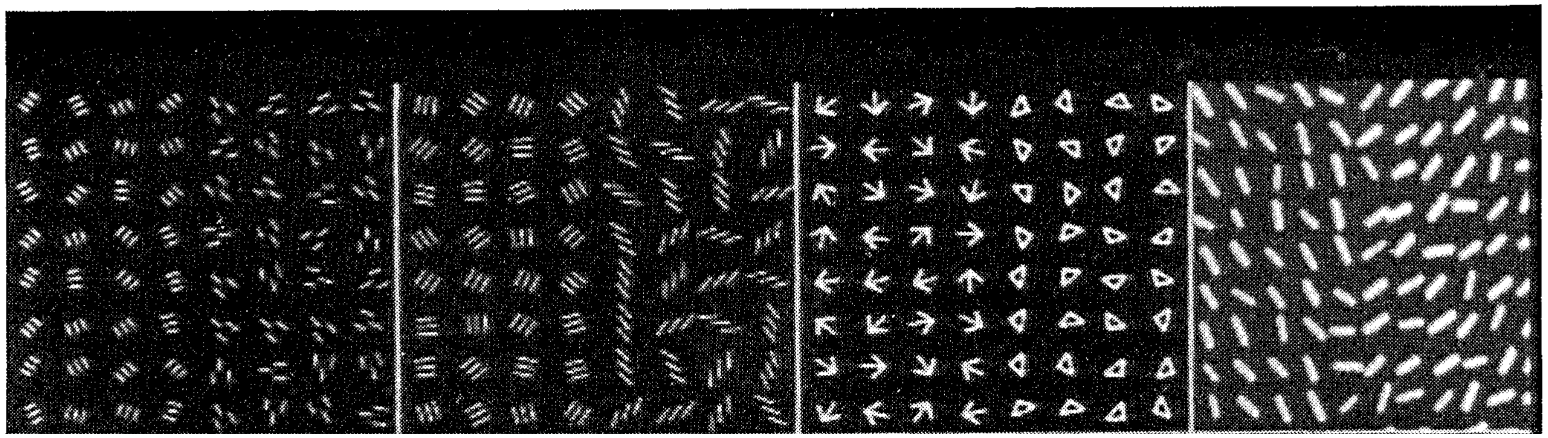


Béla Julesz (1928–2003)



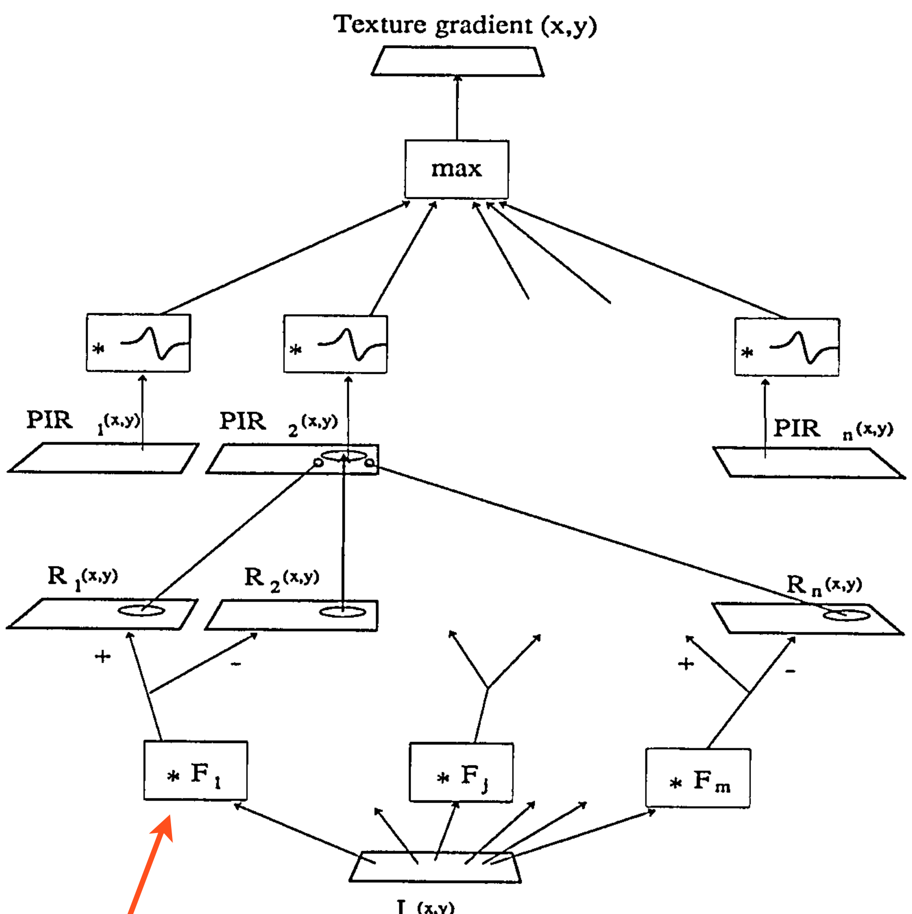
Filter-Rectify-Filter (FRF) model

texture

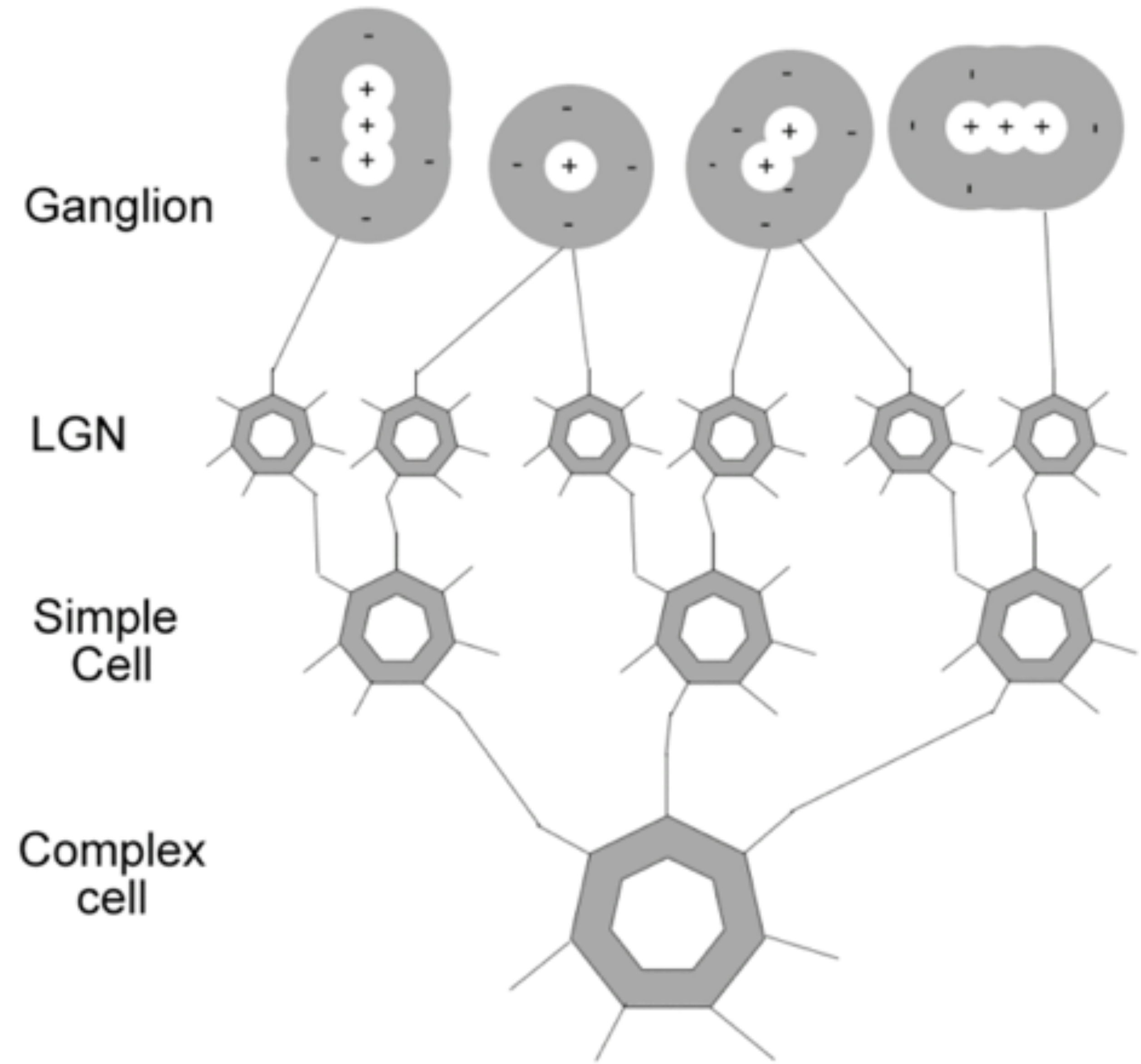


Malik & Perona '90

A model for texture boundary detection

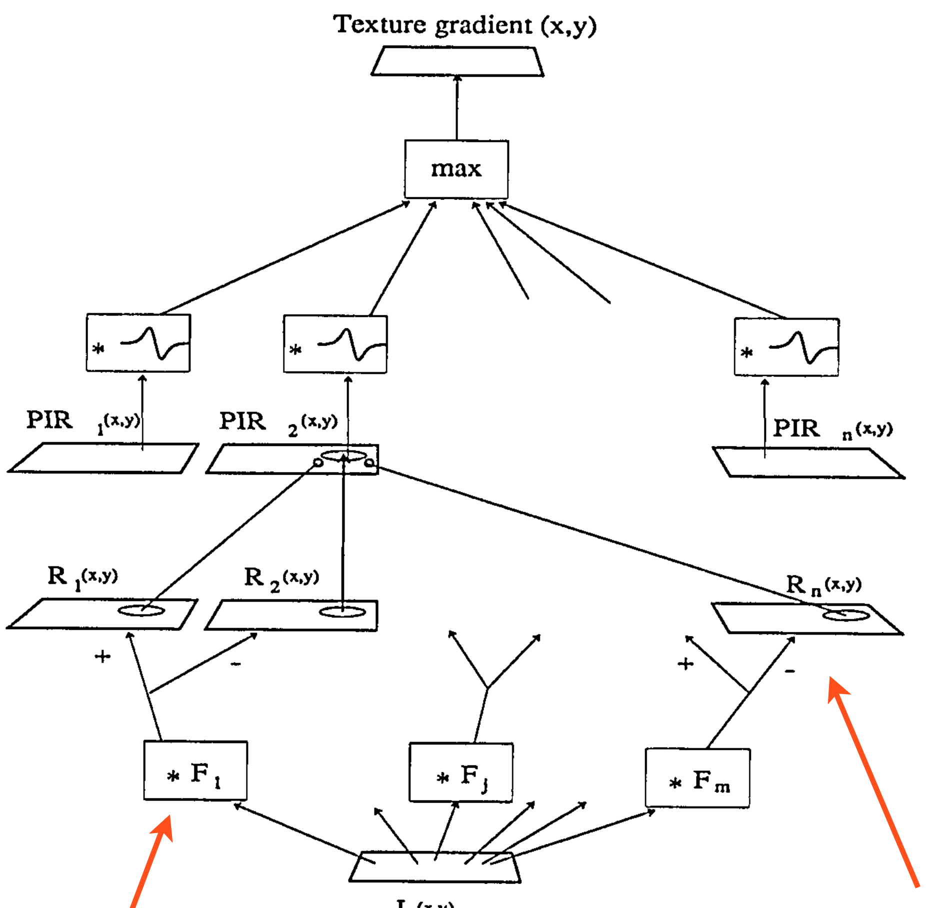


Gabor pyramid
(simple cells)



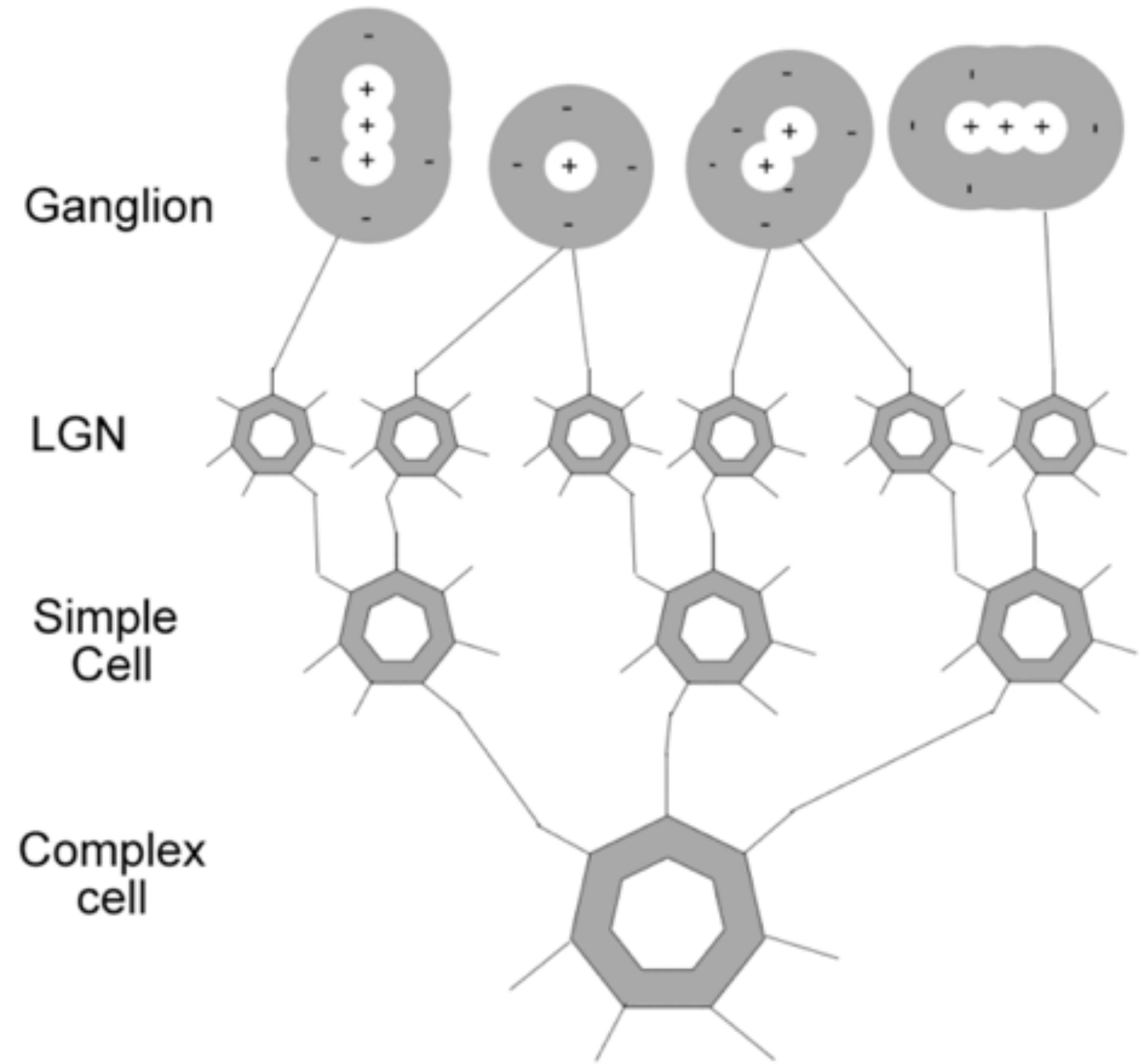
Malik & Perona '90

A model for texture boundary detection



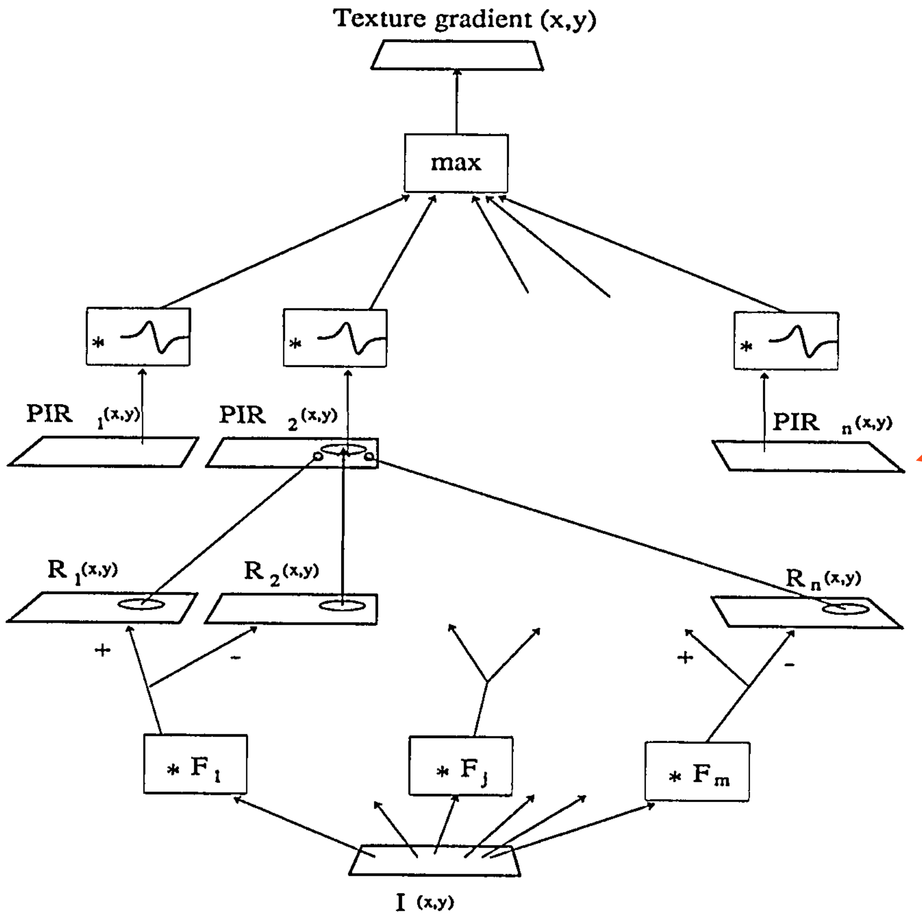
Gabor pyramid (simple cells)

Local 'averaging' (like complex cells) but no pooling over phase



Malik & Perona '90

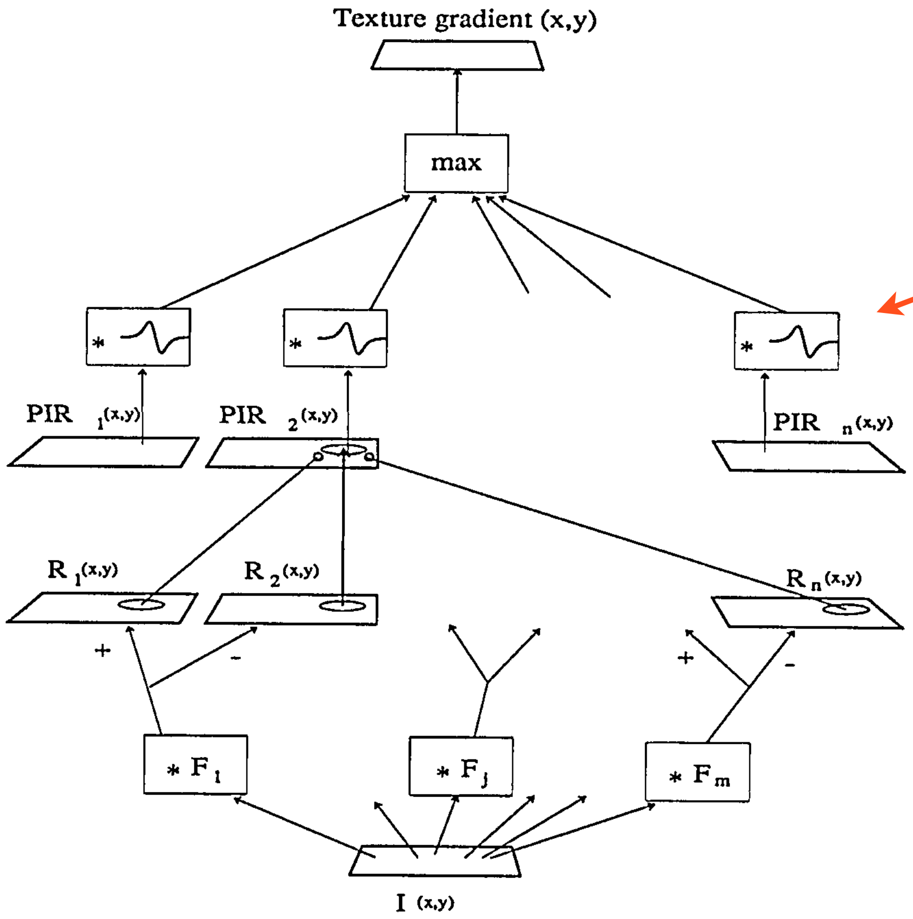
A model for texture boundary detection



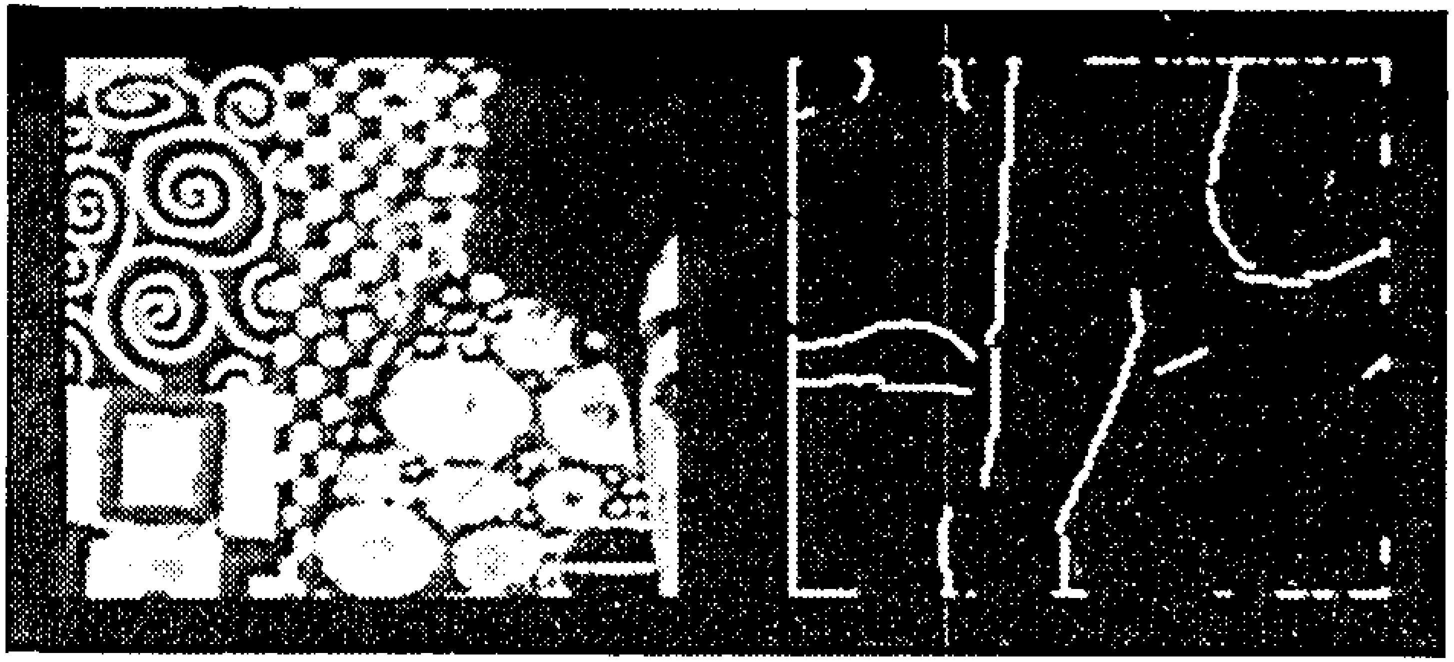
normalization / gain control

Malik & Perona ’90

Texture boundaries

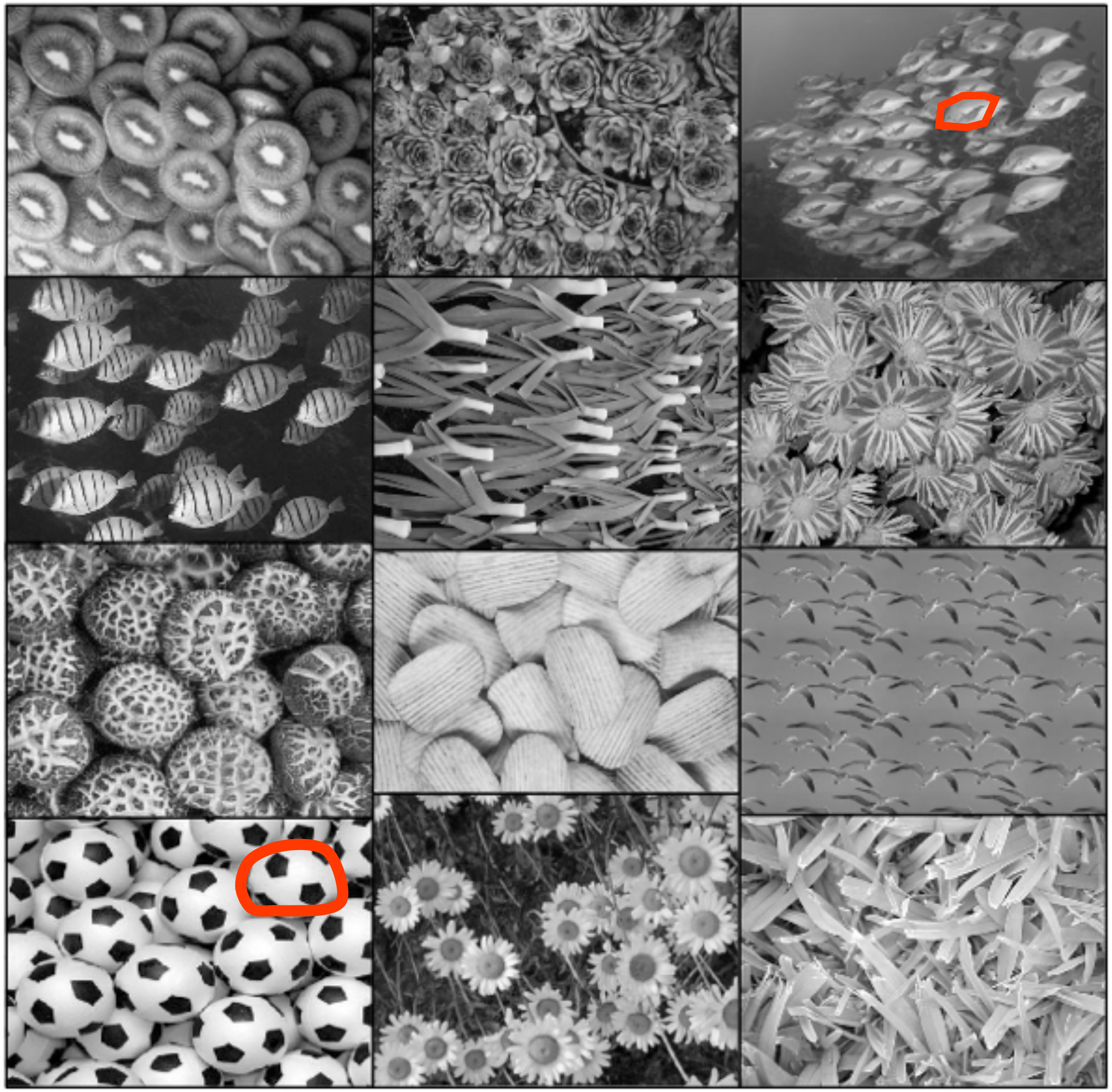


Local derivative



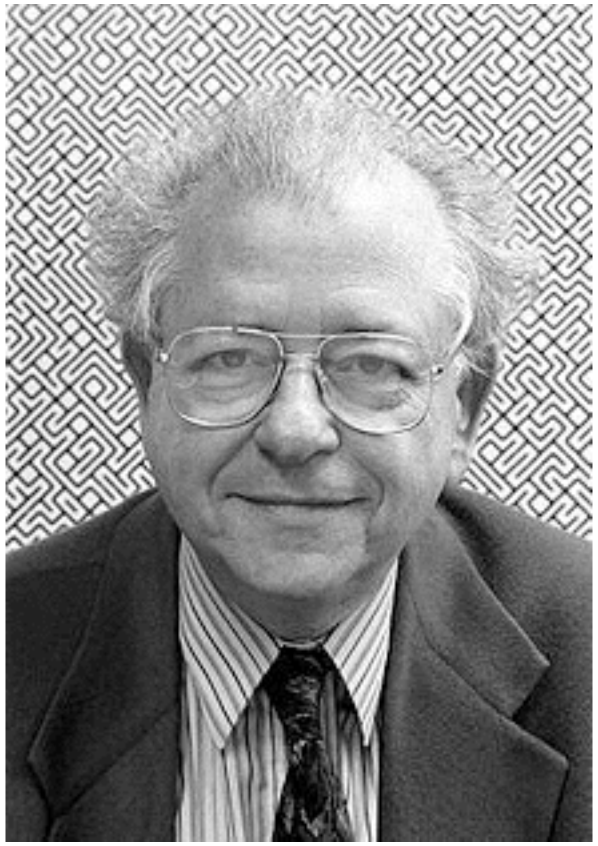
Malik & Perona '90

Orientation-based vs. texton-based

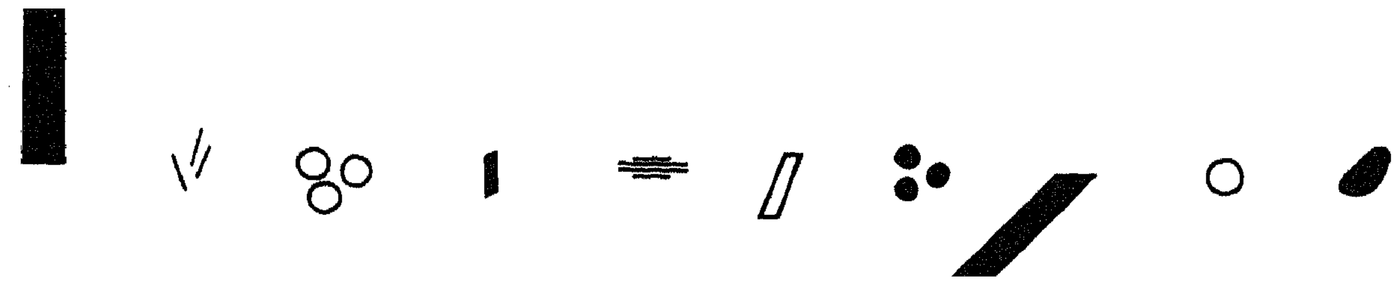


Textons

- Textons refer to fundamental micro-structures in natural images
- Considered the atoms of pre-attentive human visual perception (Julesz)



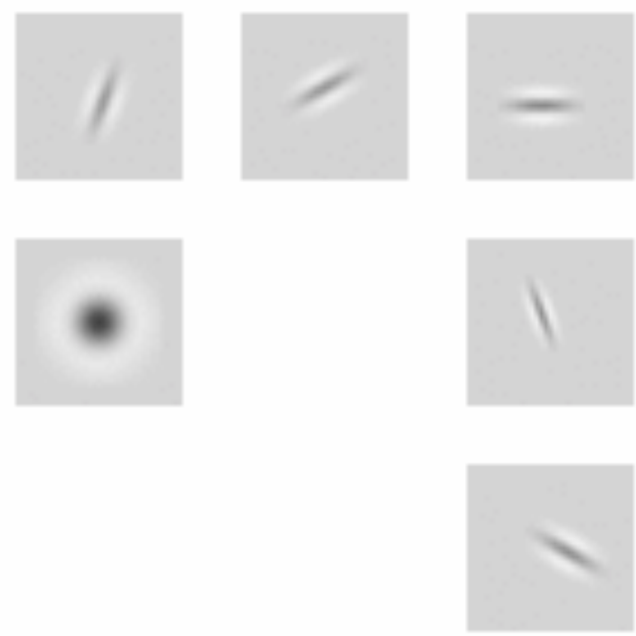
Béla Julesz
(1928–2003)

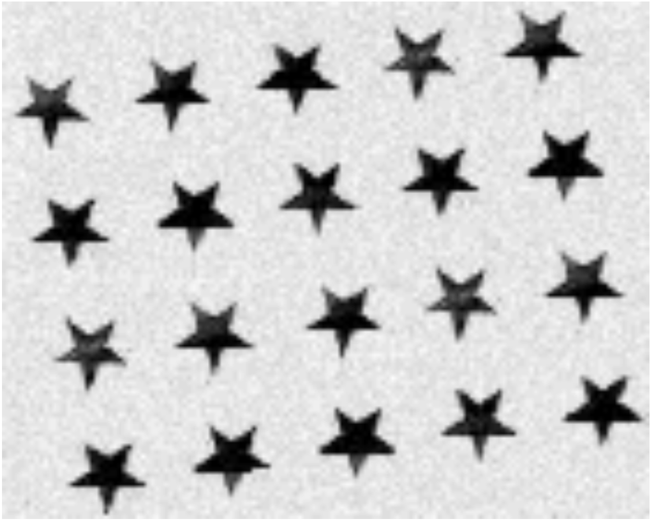


Marr's tokens

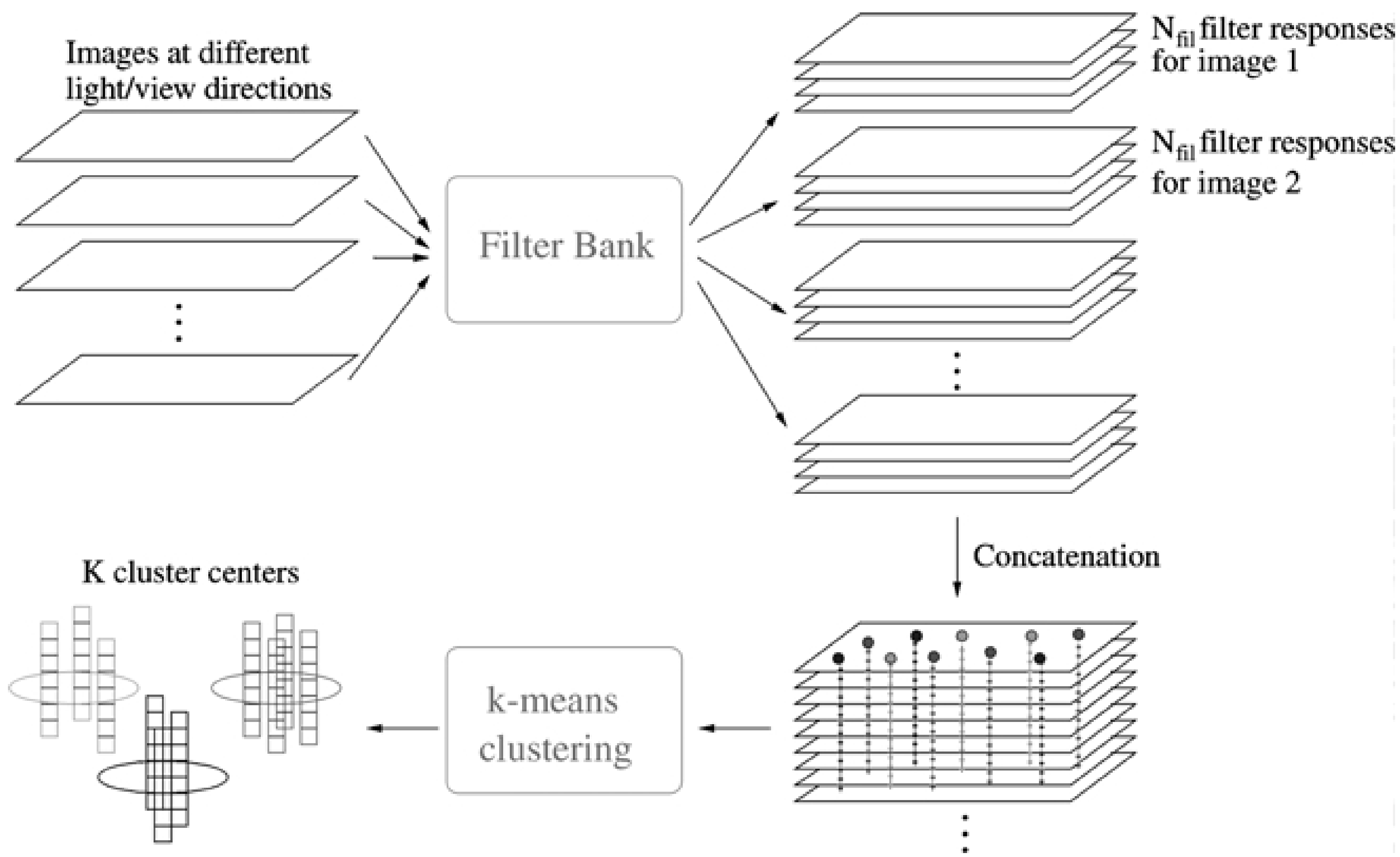
How to represent textons?

Key insight: by building units tuned to combinations of orientations



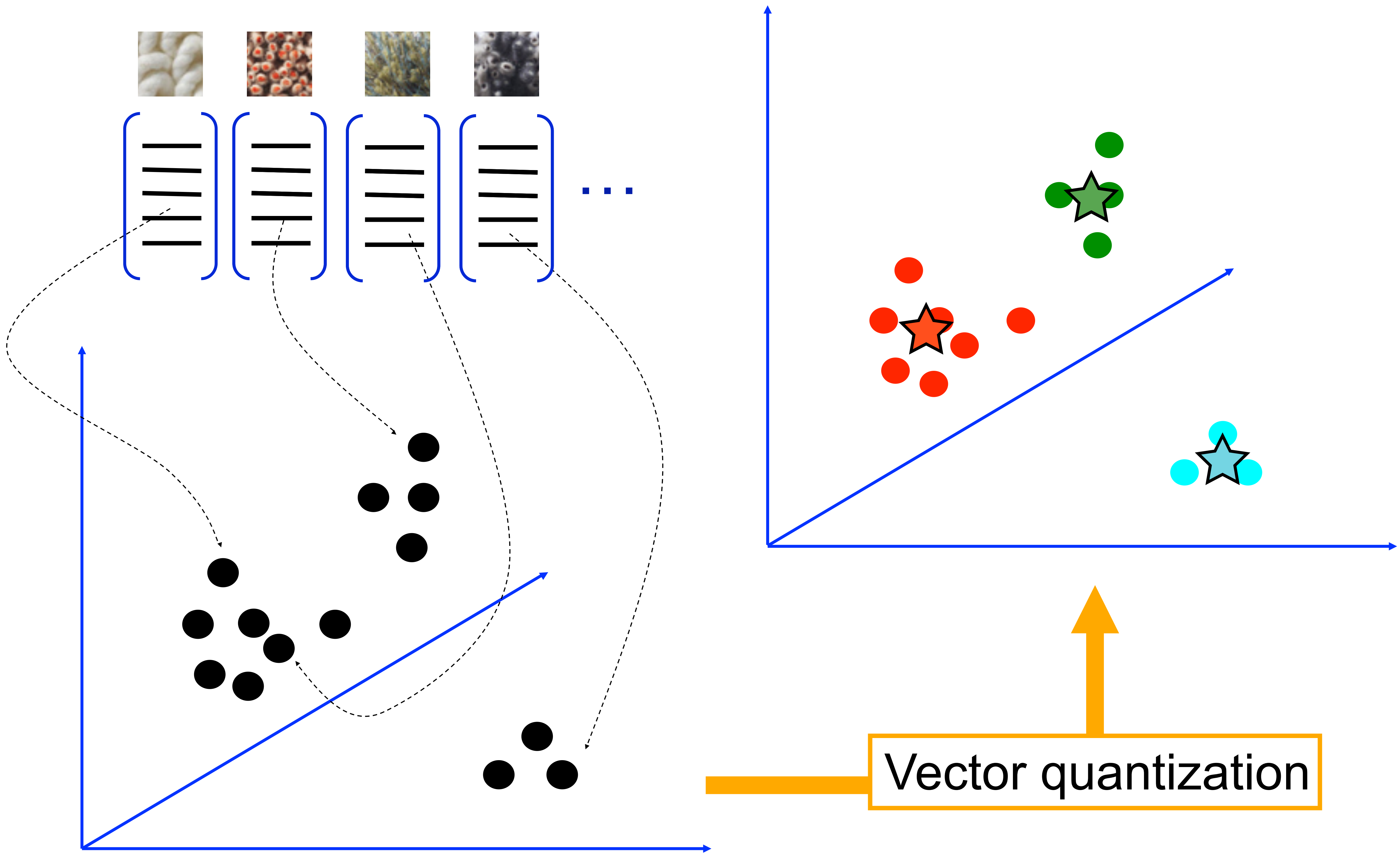


Simple algorithm for building a dictionary/ codebook of textons (your next assignment)



Leung & Malik '01

Simple algorithm for building a dictionary/ codebook of textons (your next assignment)

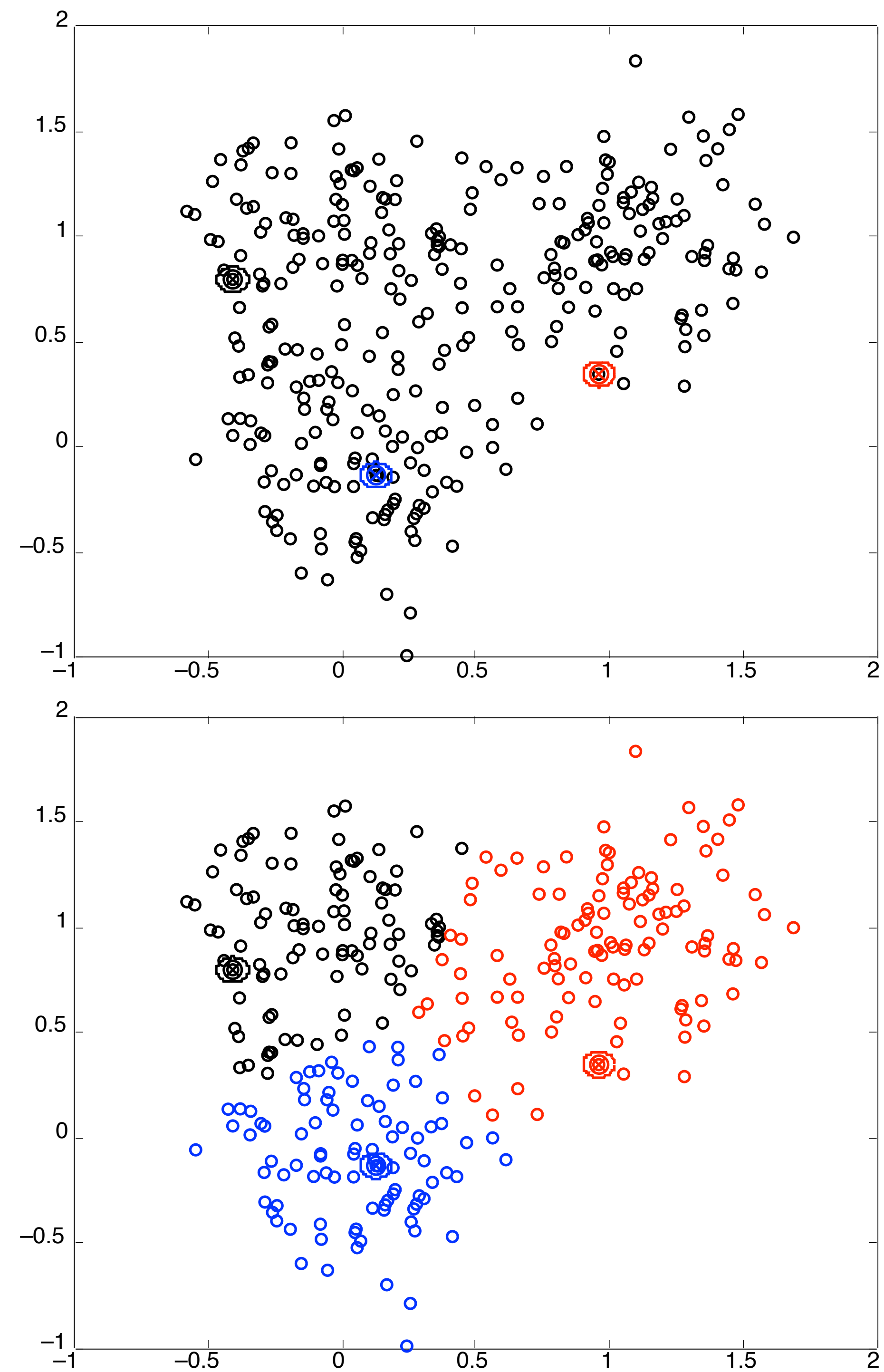


K-mean clustering

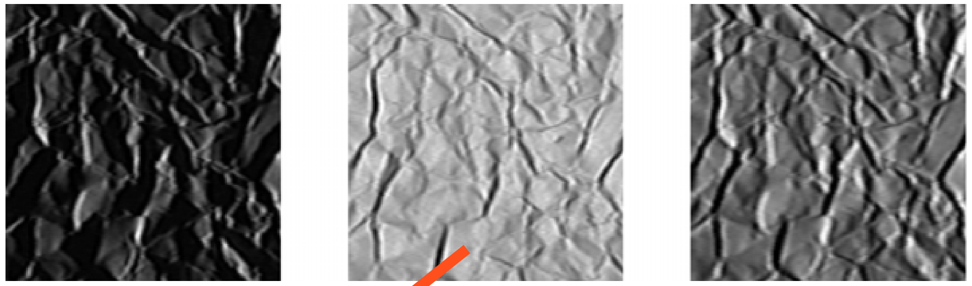
- The procedure:

    1. Pick k arbitrary centroids (cluster means)
    2. Assign each example to its “closest” centroid
    3. Adjust the centroids to be the means of the examples assigned to them
    4. Goto step 2 (until no change)

- The algorithm is guaranteed to converge in a finite number of iterations



Learning material properties



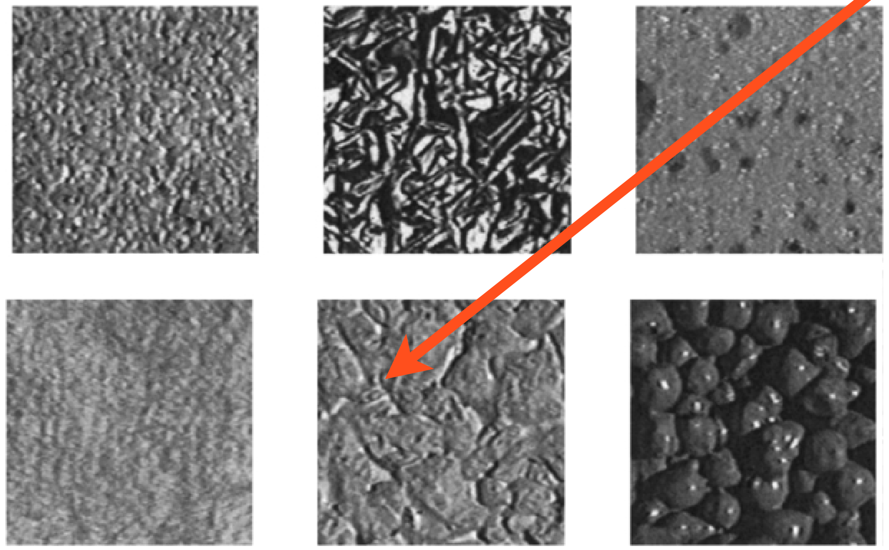
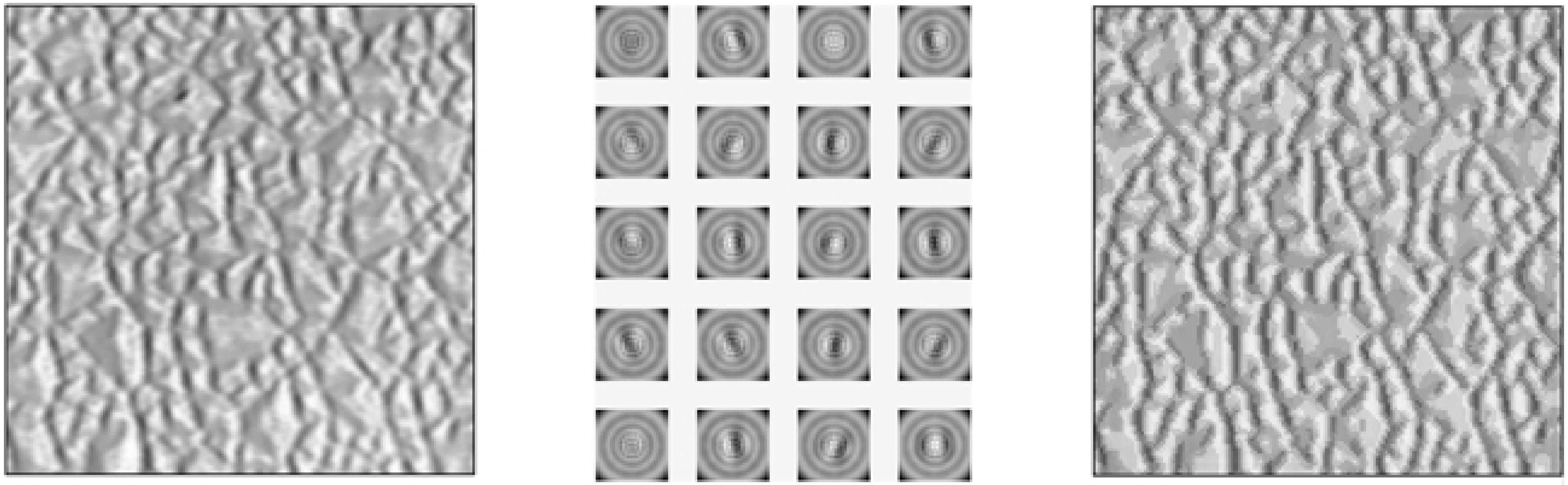
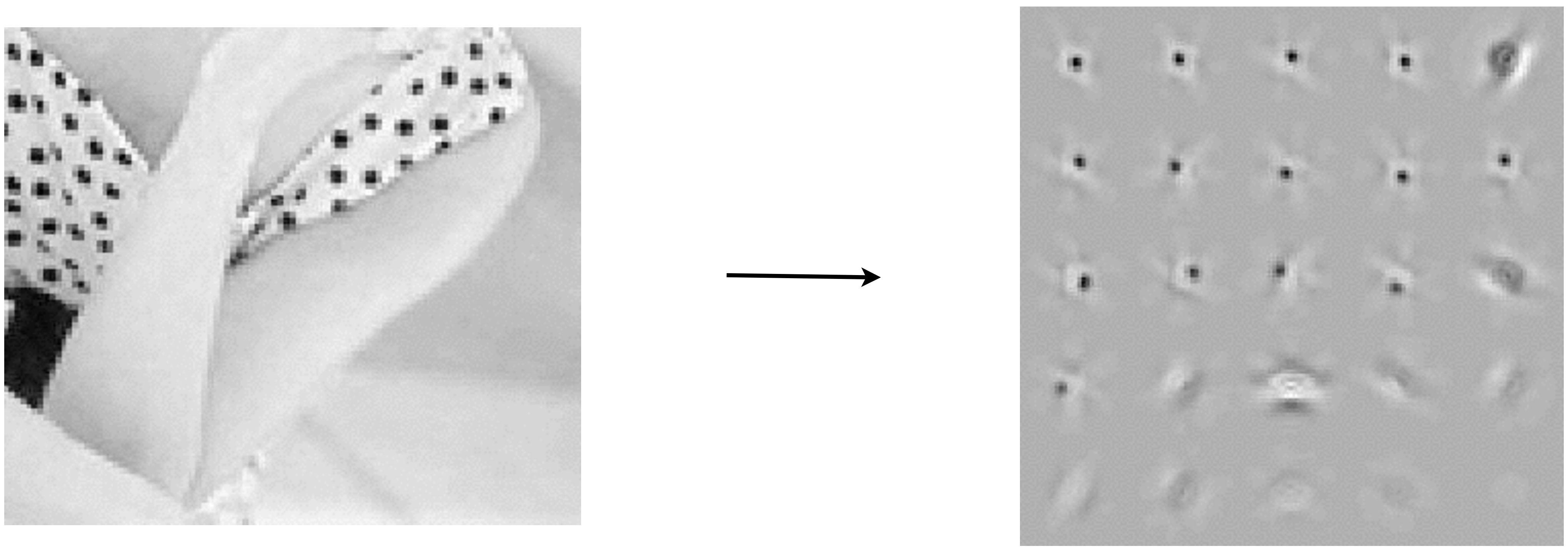


Figure 1. Some natural 3D textures from the Columbia-Utrecht database (Dana et al., 1999). Left to right: "Pebbles", "Aluminum Foil", "Sponge", "Rabbit Fur", "Concrete" and "Painted Spheres". These textures illustrate the effects caused by the 3D nature of the material: *specularities*, *shadows*, and *occlusions*.



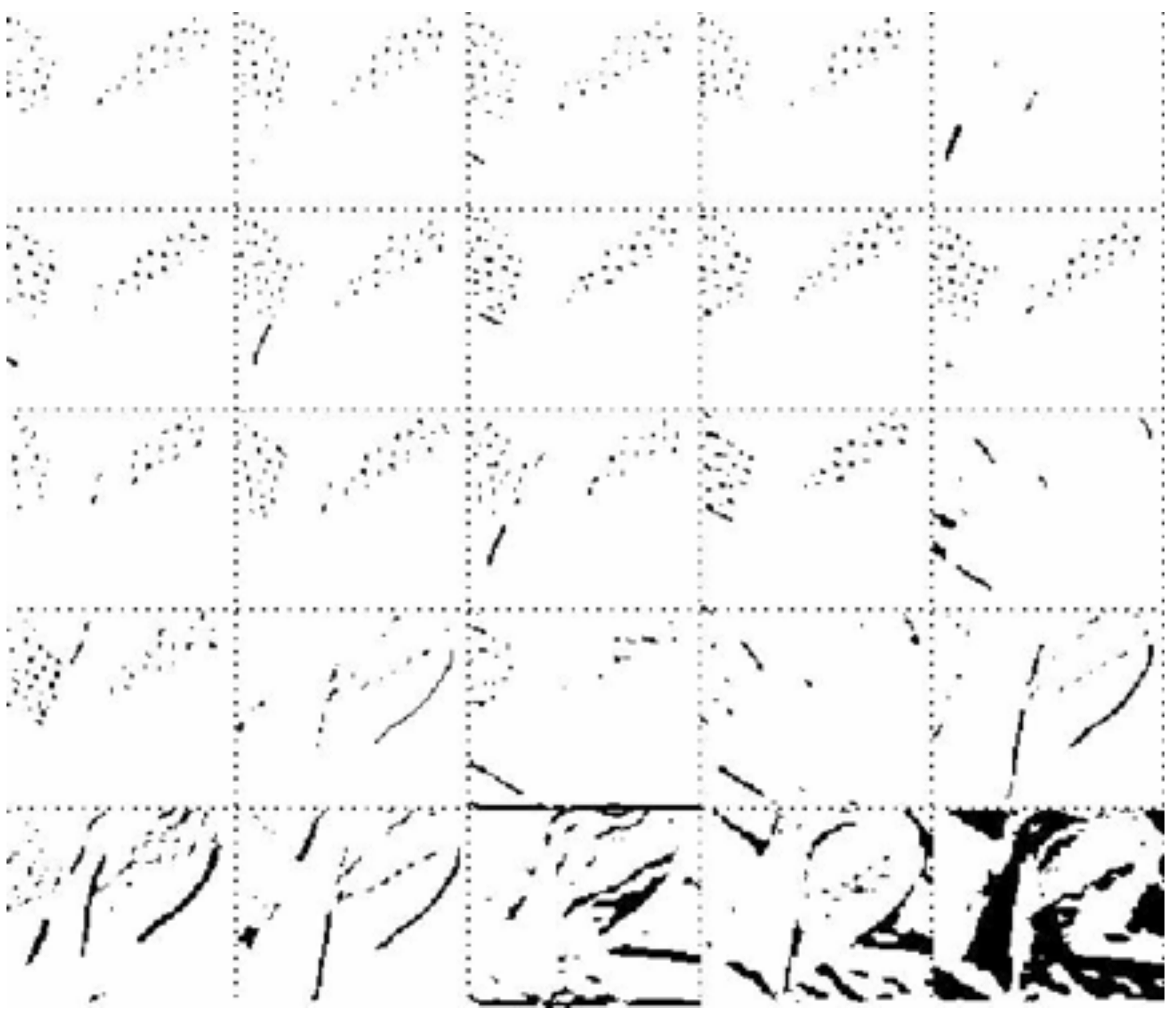
Leung & Malik '01

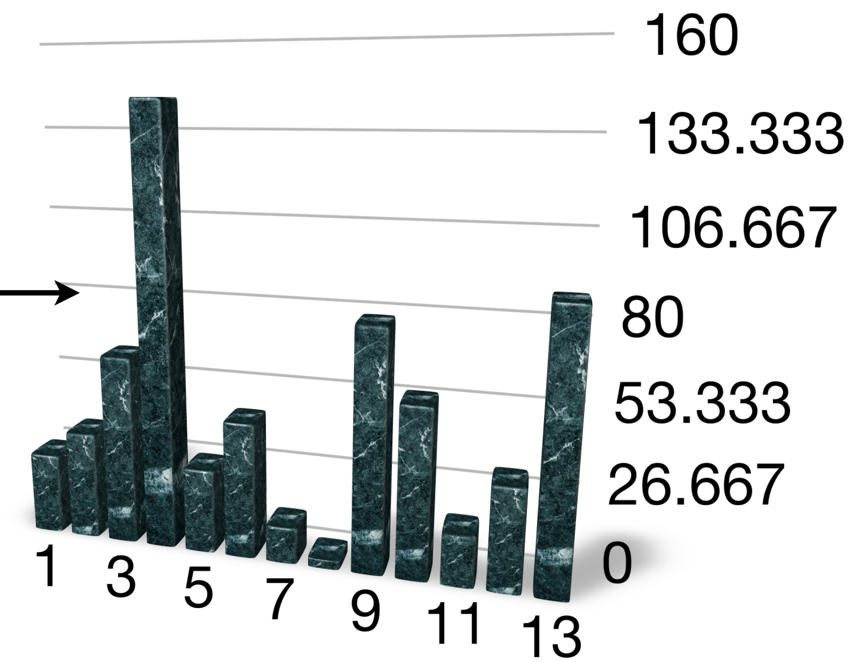
Learning textons cont'd



Leung & Malik '99

Learning textons cont'd





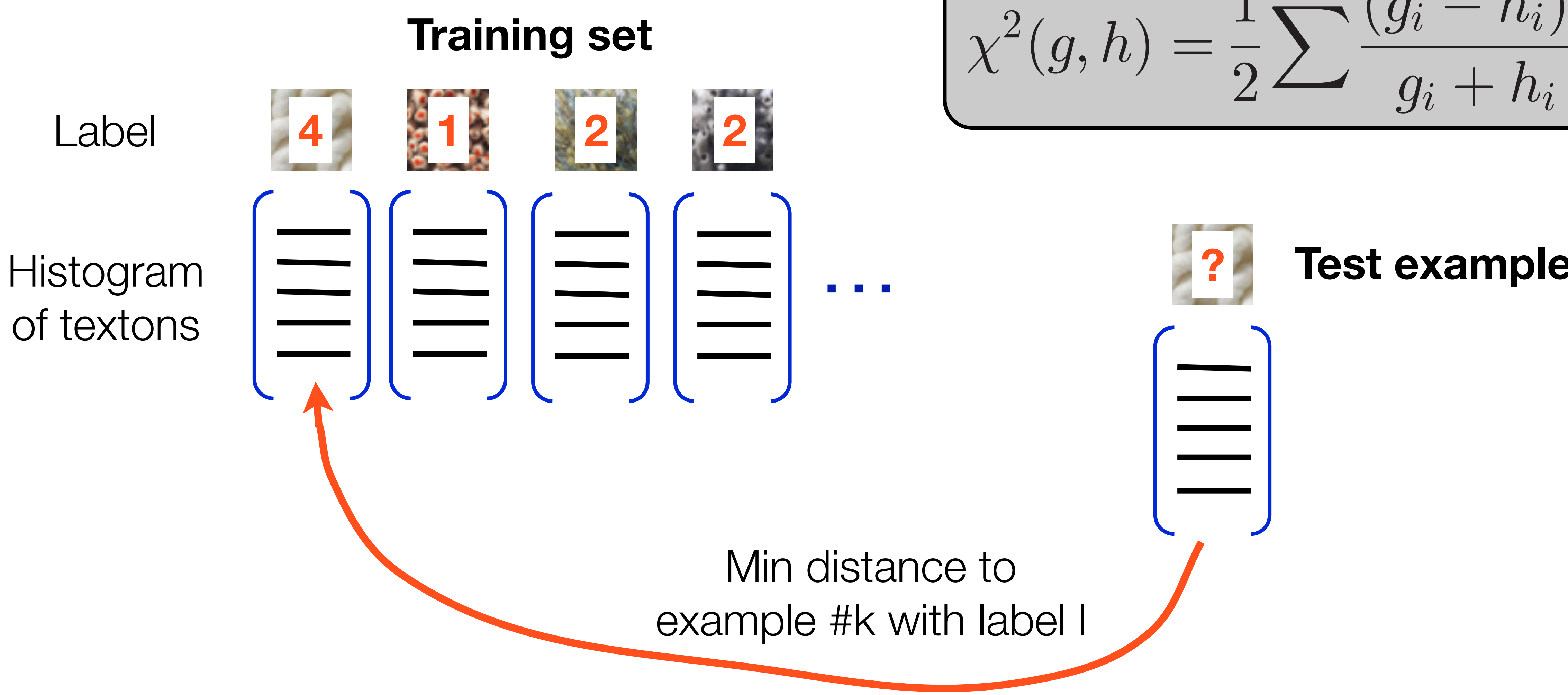
normalize

Leung & Malik '99

Classification with Nearest Neighbor

chi-square distance

$$\chi^2(g,h) = \frac{1}{2}\sum \frac{(g_i - h_i)^2}{g_i + h_i}.$$

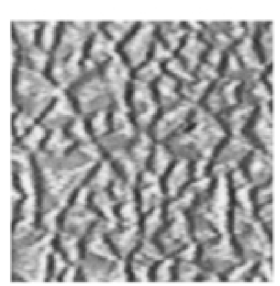


More in your next assignment...

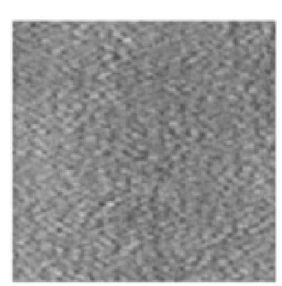
Leung & Malik '99

Material classification

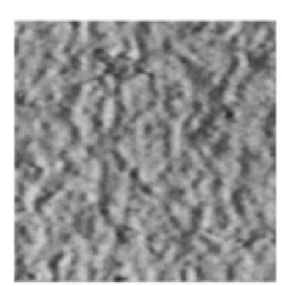
	Felt	Terrycloth	Rough Plastic	Leather	Sandpaper	Pebbles	Plaster–a	Plaster–b	Rough Paper	Artificial Grass	Rough Shingle	Aluminum Foil	Cork	Rough Tile
Felt	**0.6**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Terrycloth	0.0	**1.0**	0.0	0.0	0.3	0.0	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0
Rough Plastic	0.0	0.0	**0.9**	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
Leather	0.2	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sandpaper	0.0	0.1	0.0	0.0	**1.0**	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Pebbles	0.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
Plaster–a	0.0	0.1	0.2	0.0	0.1	0.0	**1.0**	0.7	0.0	0.0	0.0	0.0	0.0	0.0
Plaster–b	0.0	0.2	0.1	0.0	0.0	0.0	0.8	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0
Rough Paper	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	**0.5**	0.0	0.0	0.0	0.0	0.0
Artificial Grass	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.1	0.1	0.0	0.0
Roof Shingle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	**1.0**	0.1	0.0	0.0
Aluminum Foil	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	**1.0**	0.0	0.0
Cork	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.2
Rough Tile	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**0.9**



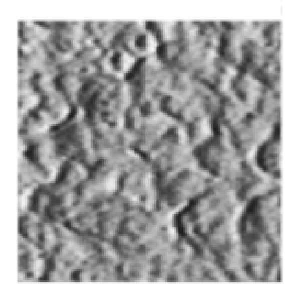
Rough Plastic



Leather



Plaster-a



Plaster-b

Leung & Malik ’01

Image segmentation

Martin et al ’04

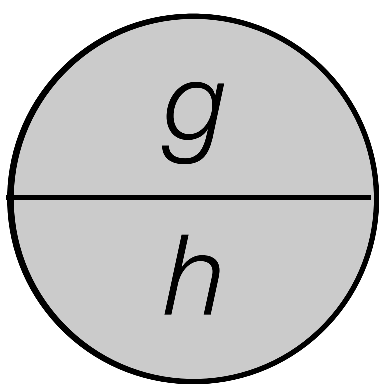
(a) (b) (c) (d)

Image segmentation

Martin et al ’04

chi-square distance

$$\chi^2(g,h) = \frac{1}{2}\sum \frac{(g_i - h_i)^2}{g_i + h_i}.$$



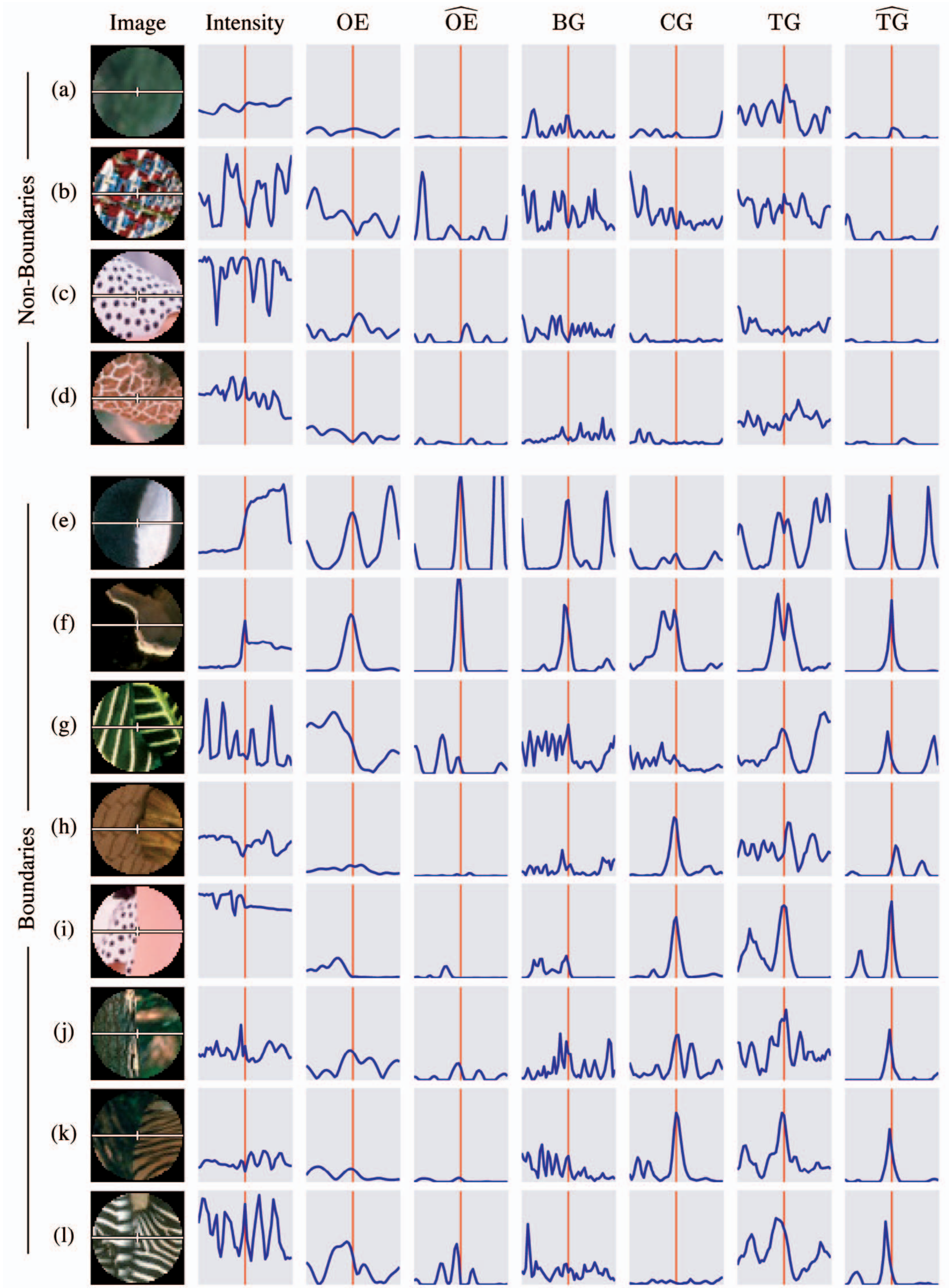
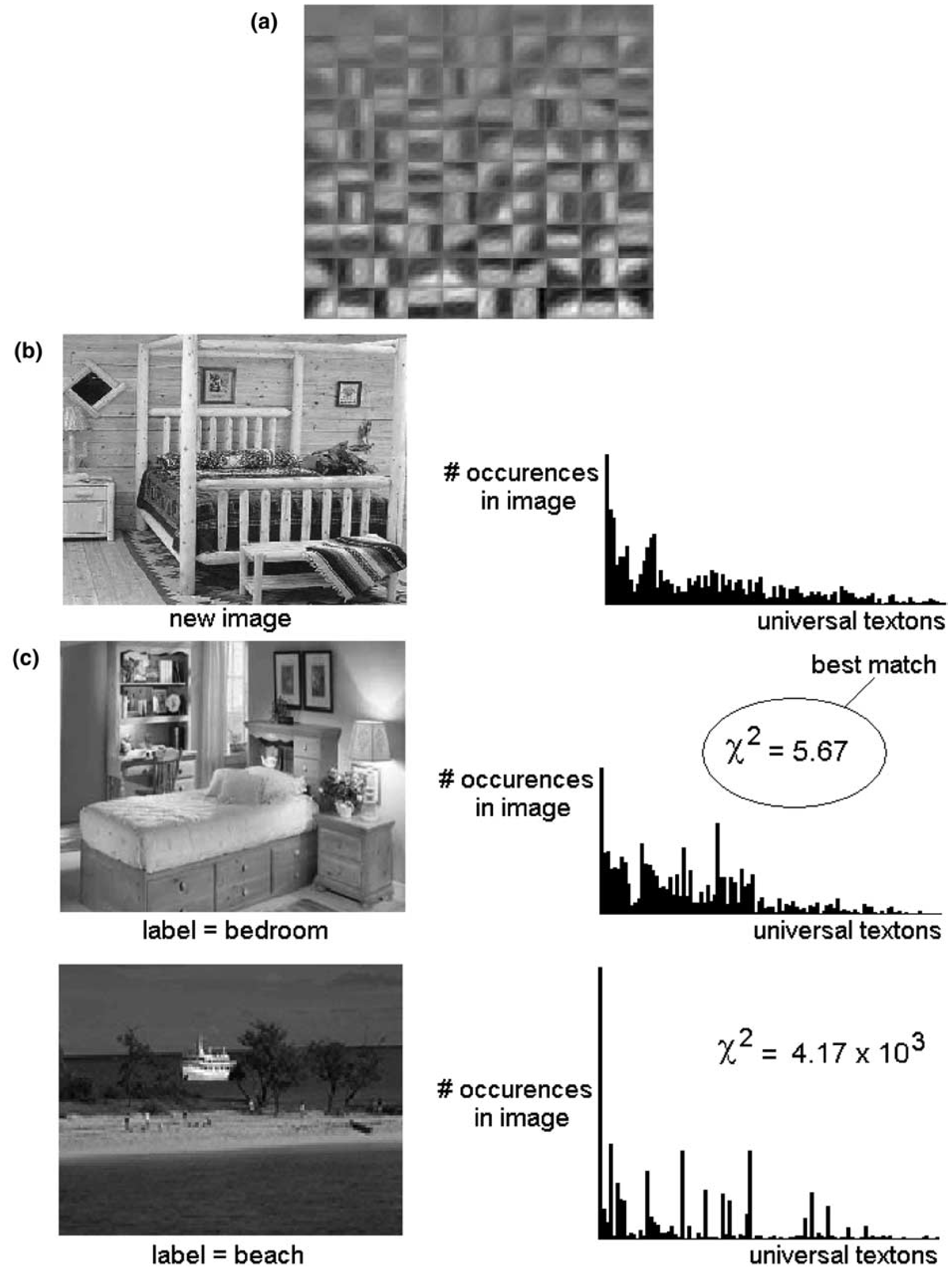


Fig. 2. Local Image Features. In each row, the first panel shows an image patch. The following panels show feature profiles along the patch’s horizontal diameter. The features are raw image intensity, oriented energy OE, brightness gradient BG, color gradient CG, raw texture gradient TG, and localized texture gradient $\widehat{\mathrm{TG}}$. The vertical line in each profile marks the patch center. The scale of each feature has been chosen to maximize performance on the set of training images—2 percent of the image diagonal (5.7 pixels) for OE, CG, and TG, and 1 percent of the image diagonal (3 pixels) for BG. The challenge is to combine these features in order to detect and localize boundaries.

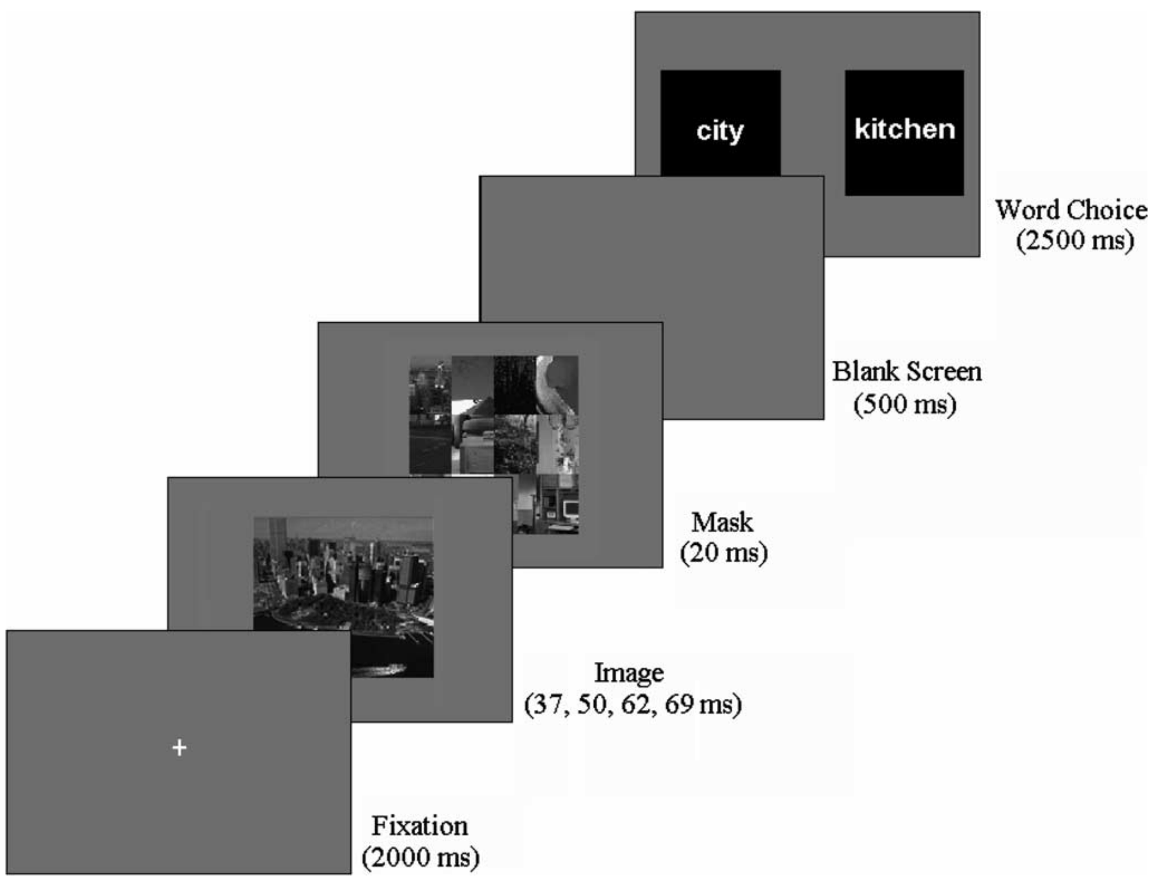
Scene classification with texton histograms

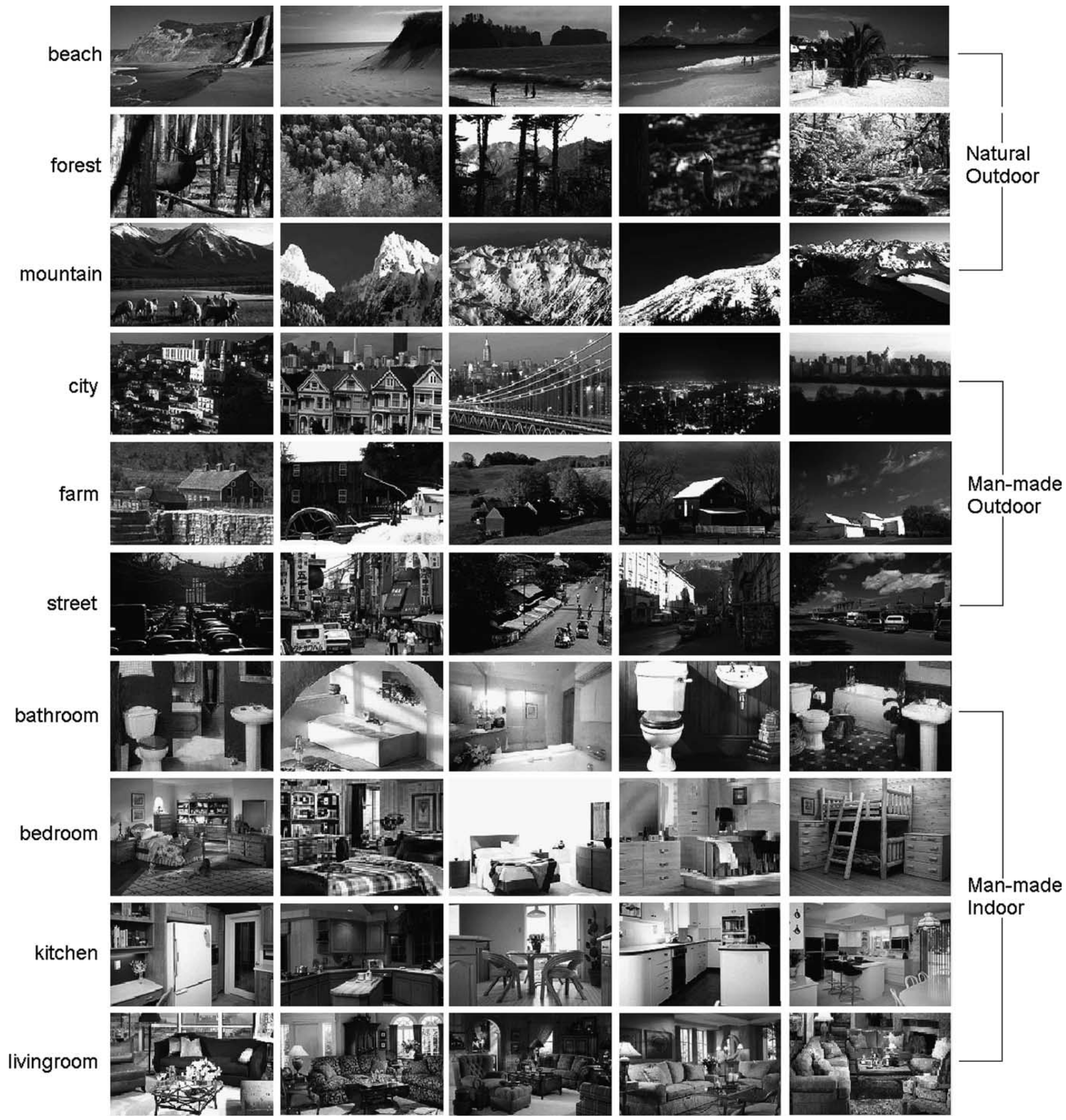
Generic / universal dictionary



Reninger & Malik '04

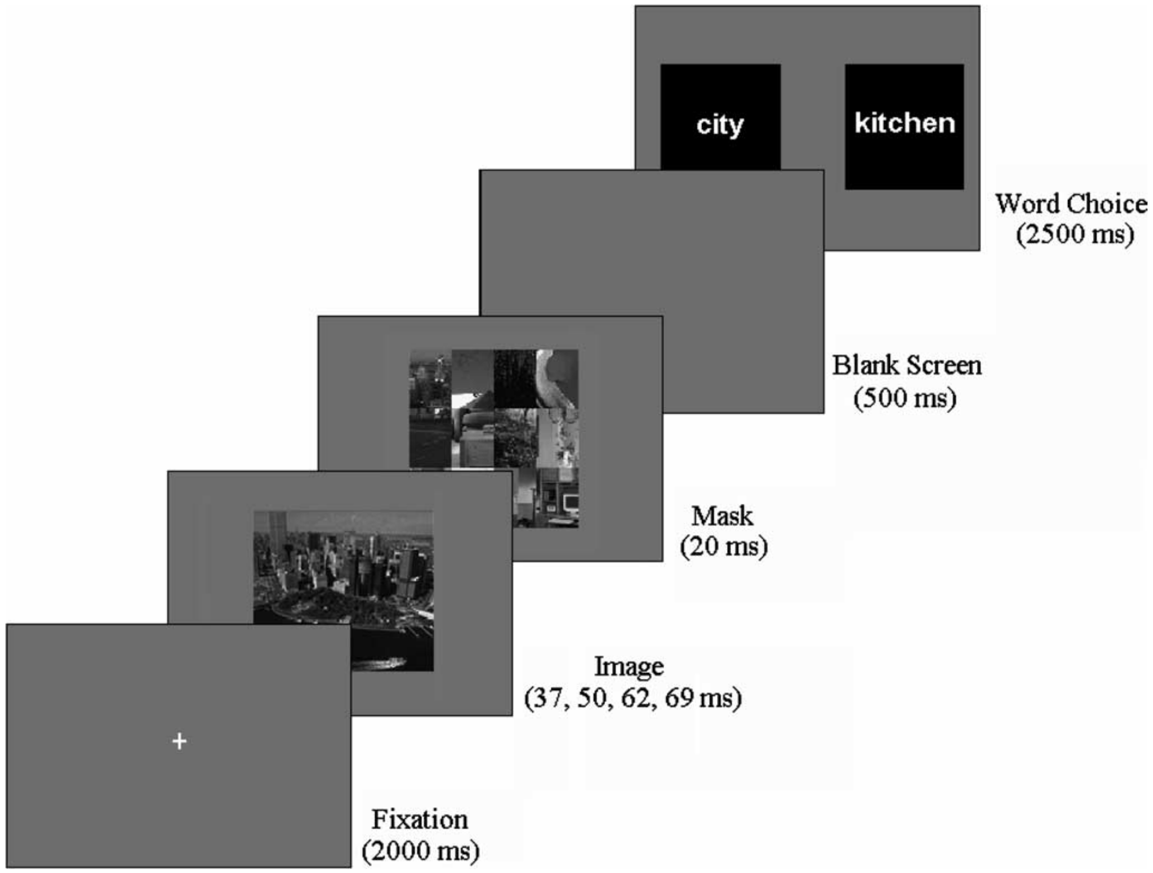
Scene classification with texton histograms





Reninger & Malik '04

Scene classification with texton histograms



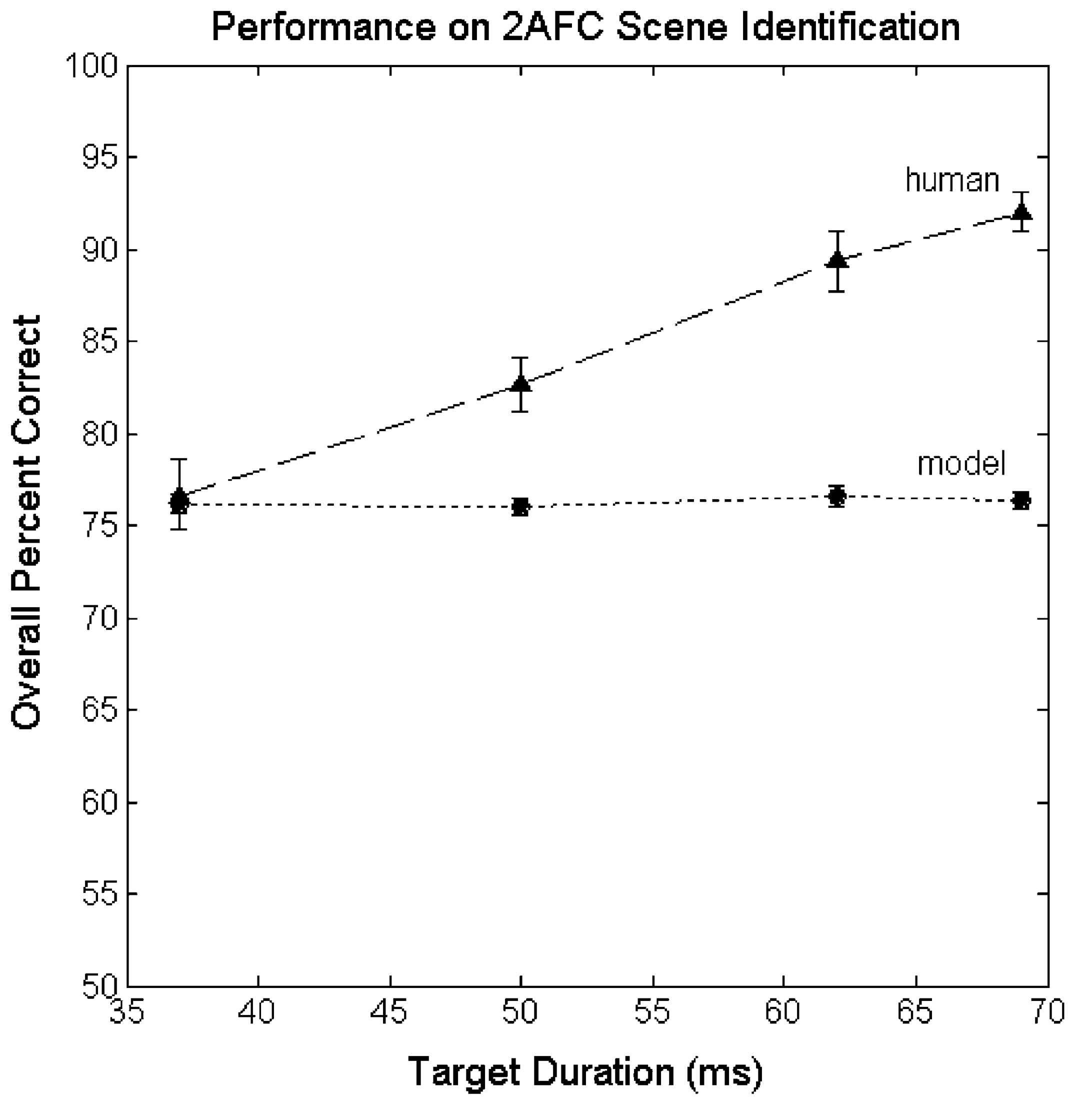


Fig. 5. Subject accuracy in the 2AFC scene discrimination task improves with increased presentation time. The percent correct is plotted with its 95% confidence intervals for 48 subjects (11, 15, 8 and 14 subjects at 37, 50, 62 and 69 ms). Chance performance is 50% correct.

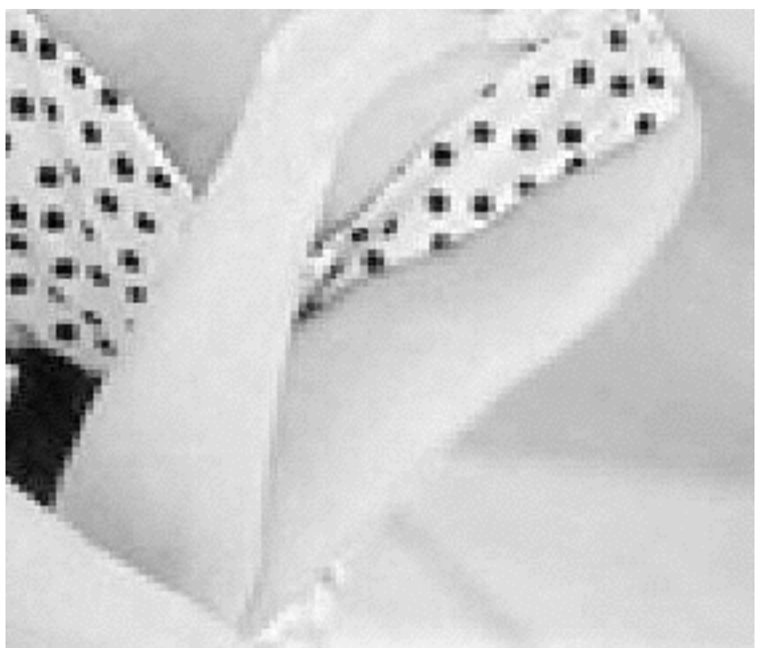
Reninger & Malik ’04

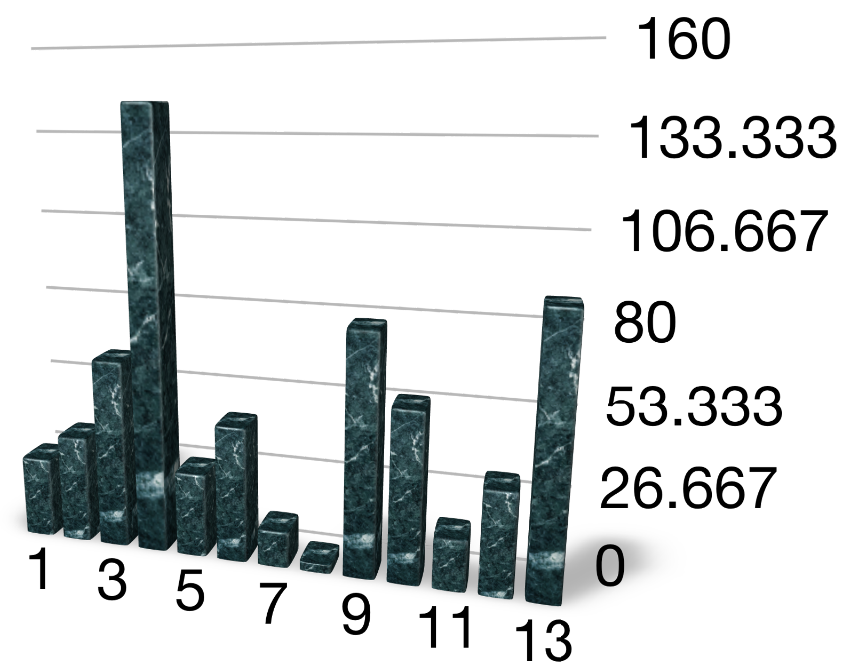
Computational Vision

Scenes, textures & surfaces

- Finish texture analysis
  - Texture synthesis
  - Structure-from-texture
- Start motion processing

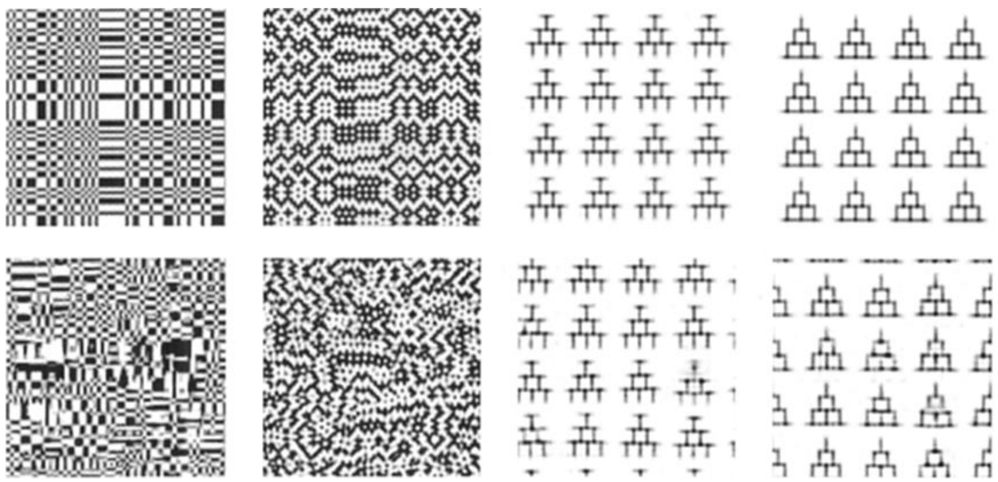
Texture synthesis

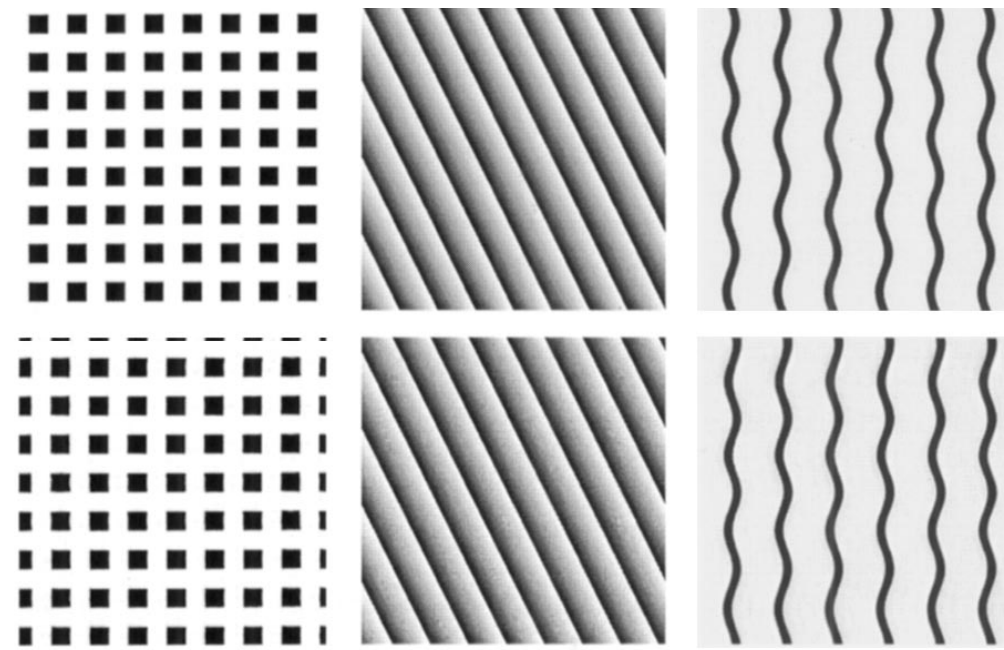


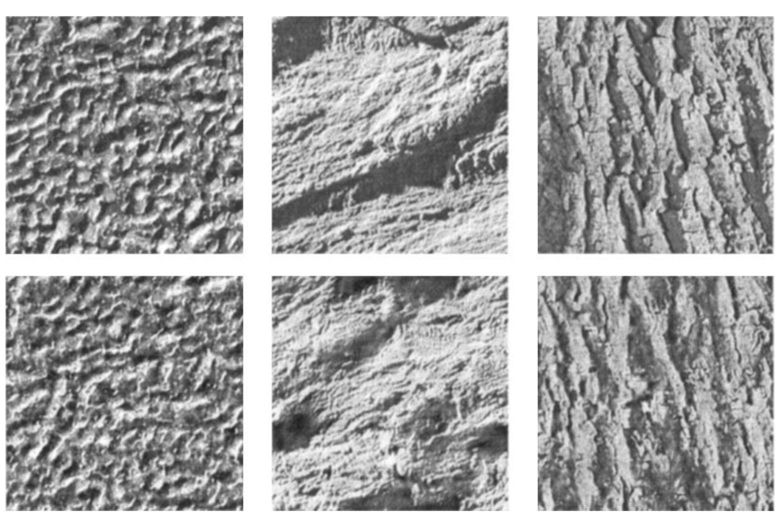


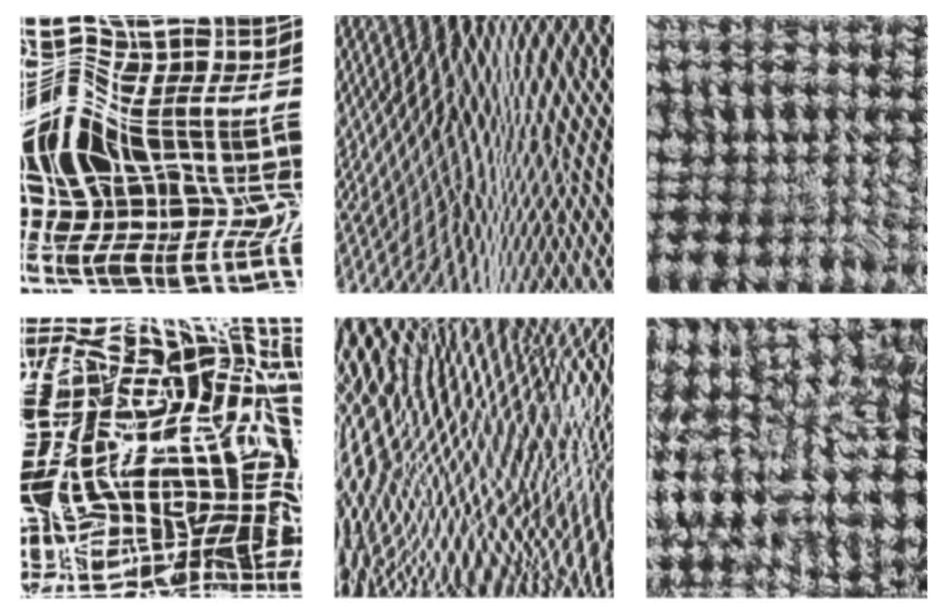
noise

Texture synthesis





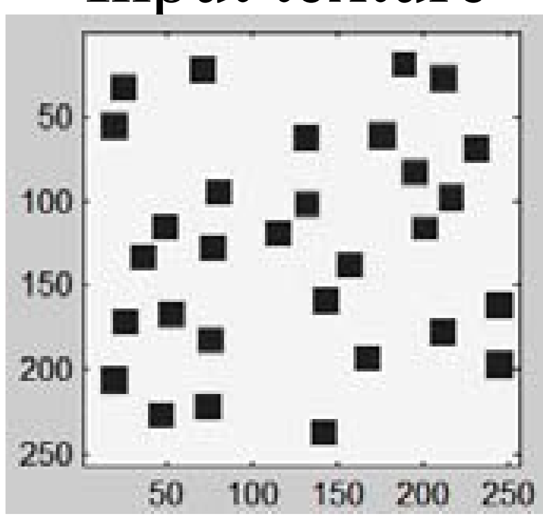




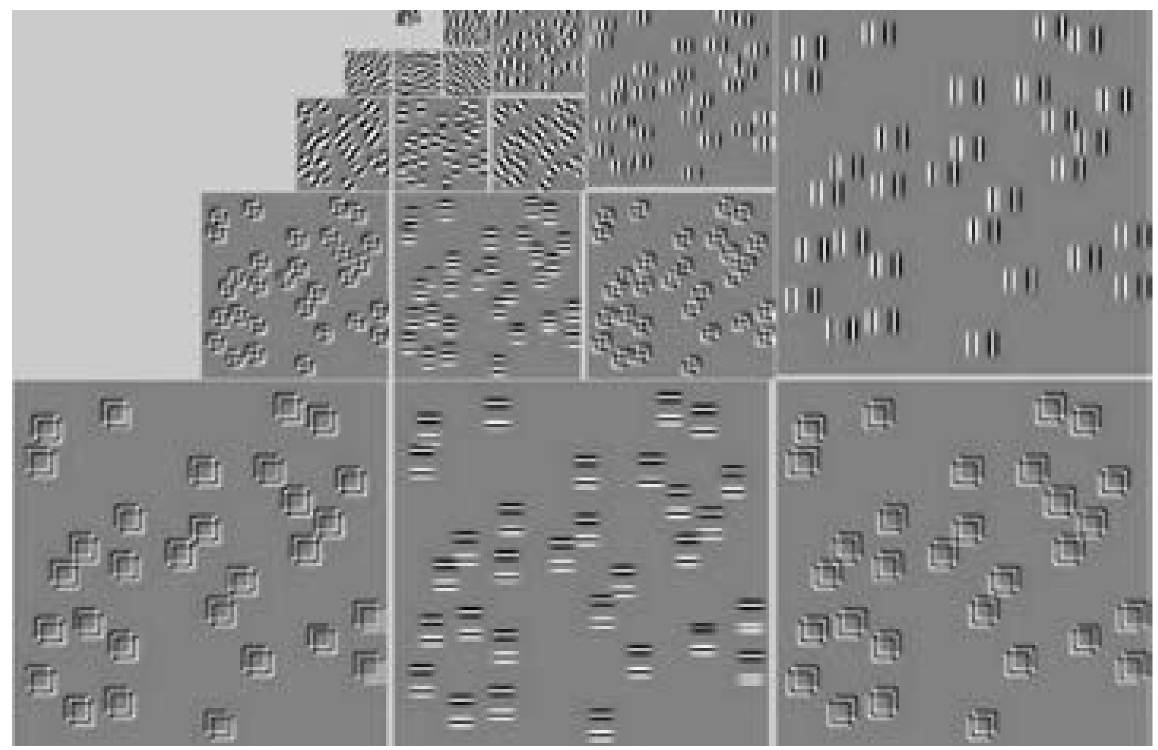
Heeger & Bergen ’96; Portilla & Simoncelli ’00

1. Multi-scale image decomposition

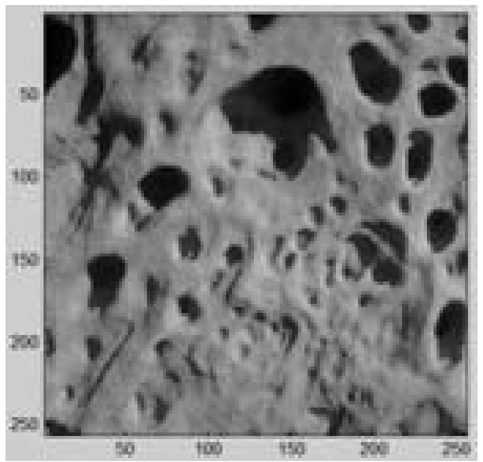
Input texture

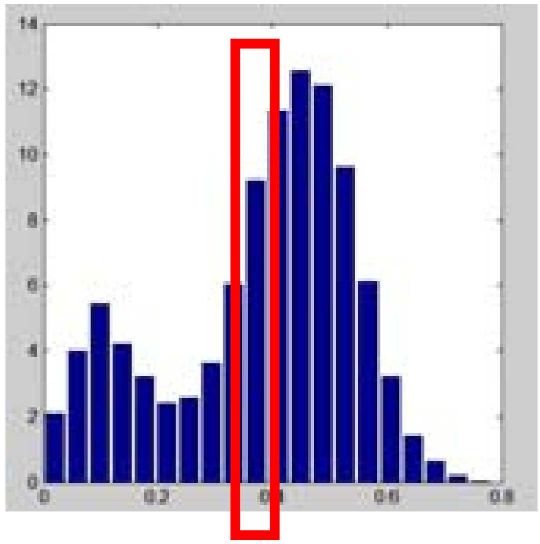


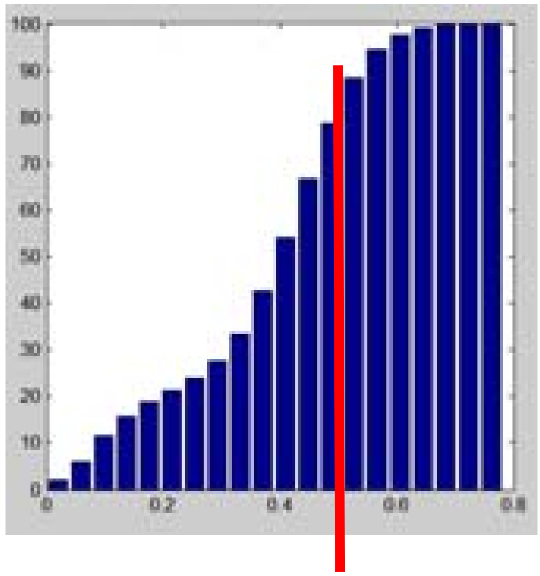
Steerable pyramid



2. Histogram matching

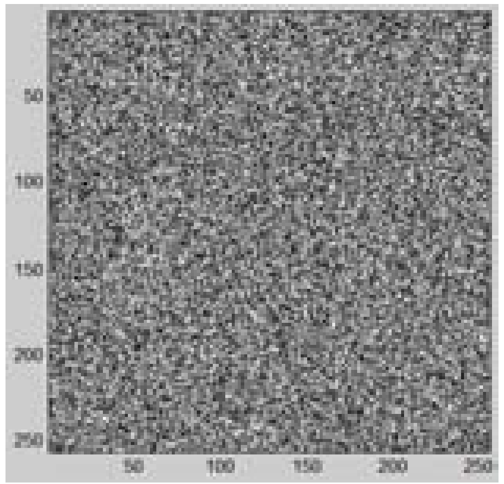


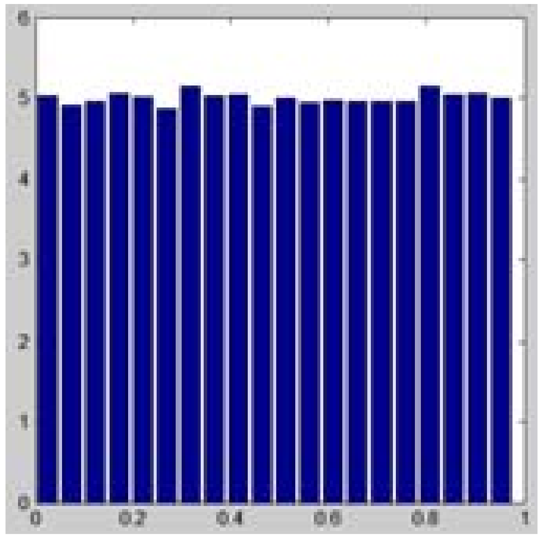


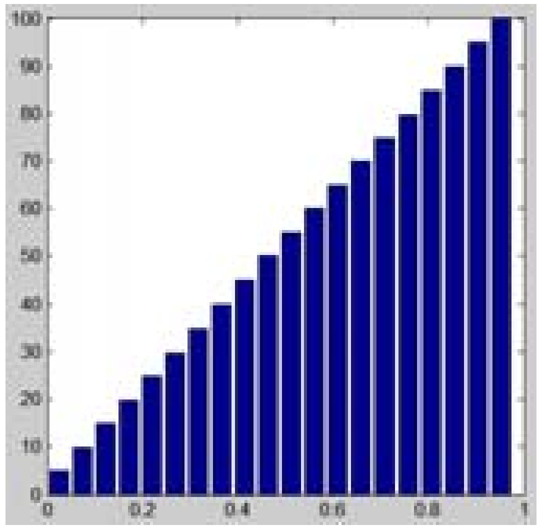


9% of pixels have an intensity value within the range[0.37, 0.41]

75% of pixels have an intensity value smaller than 0.5



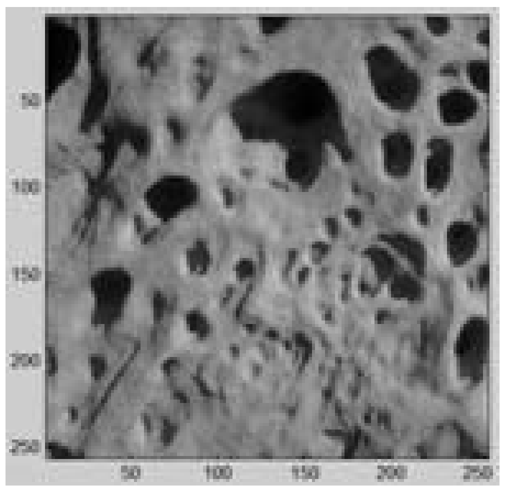


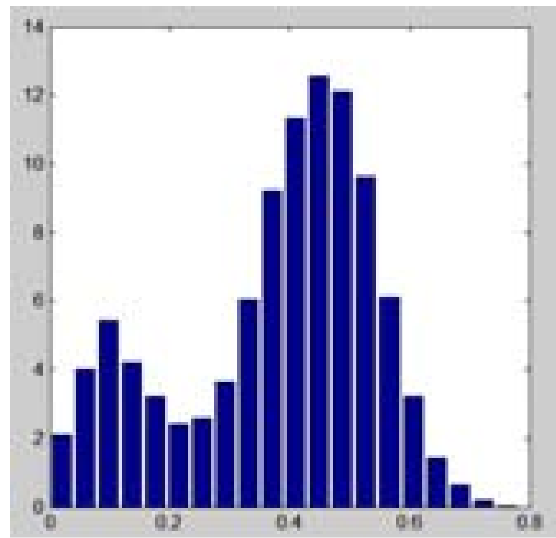


5% of pixels have an intensity value within the range[0.37, 0.41]

2. Histogram matching

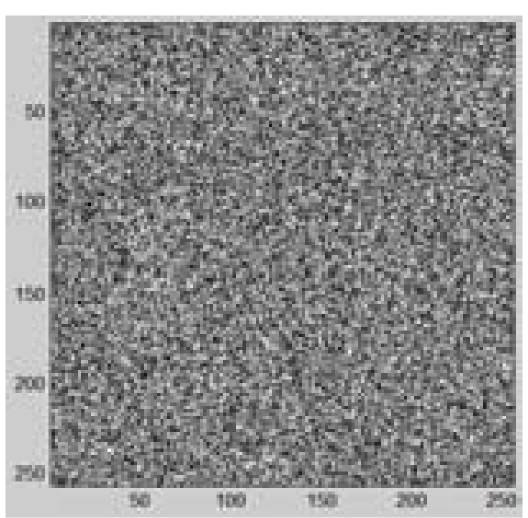
Z(x,y)

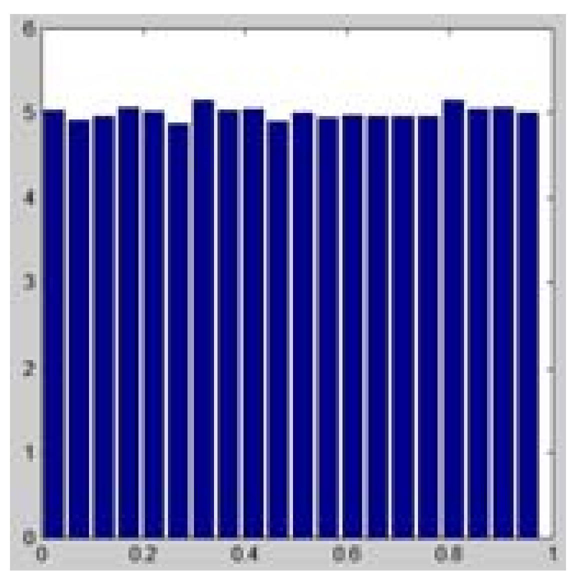




?

Y(x,y)





We look for a transformation of the image Y

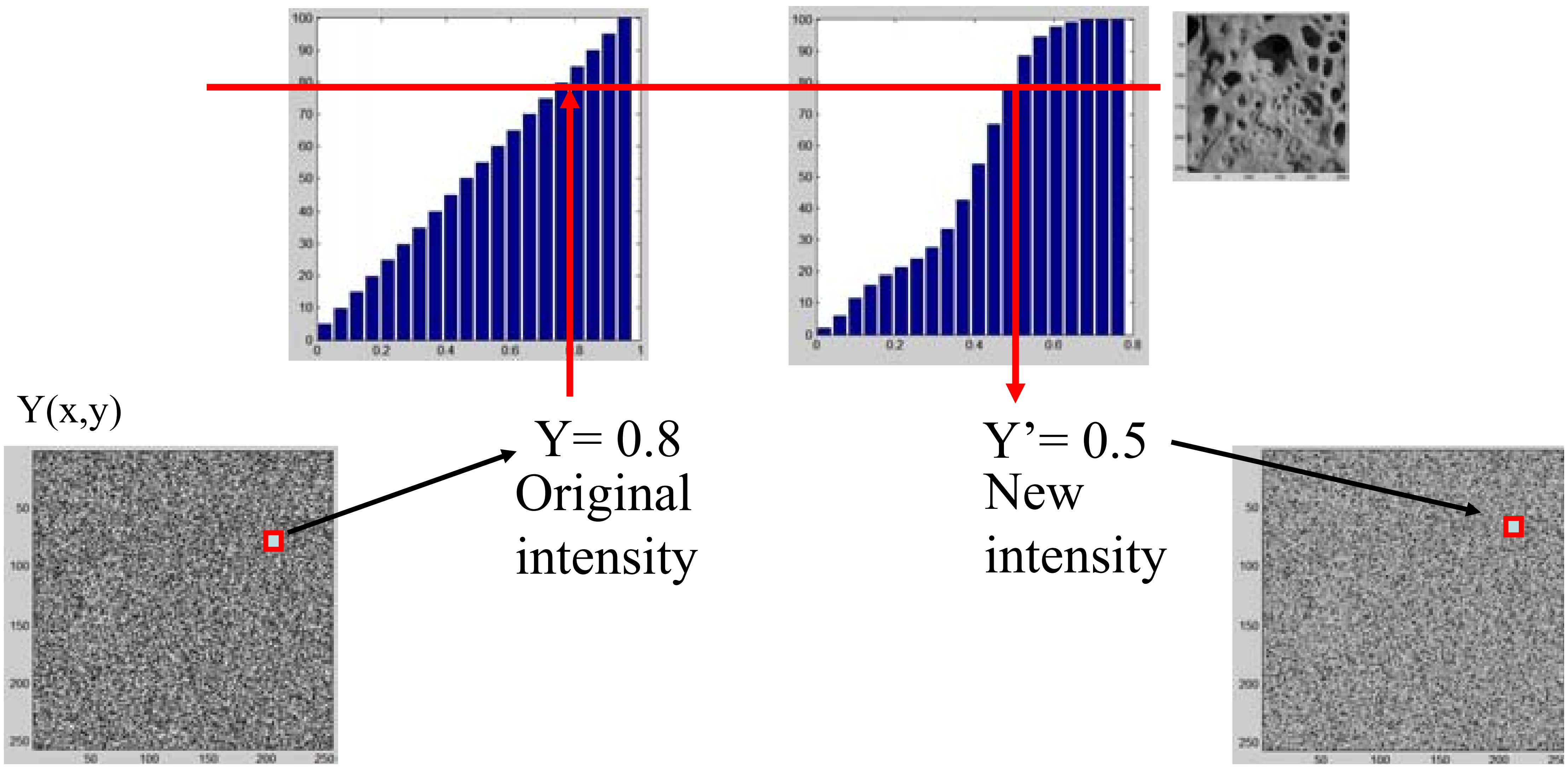
Y' = f (Y)

Such that
Hist(Y) = Hist(f(Z))

2. Histogram matching

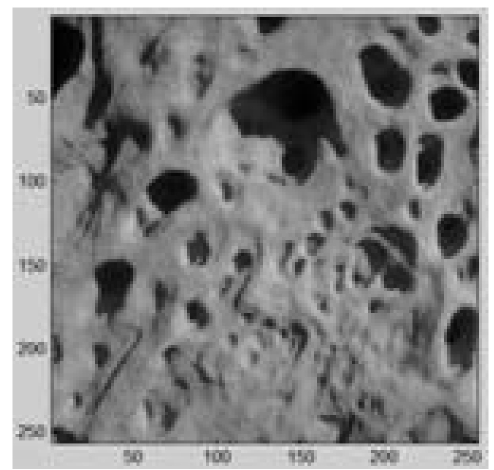
The function f is just a look up table: it says, change **all the pixels of value Y into a value f(Y).**

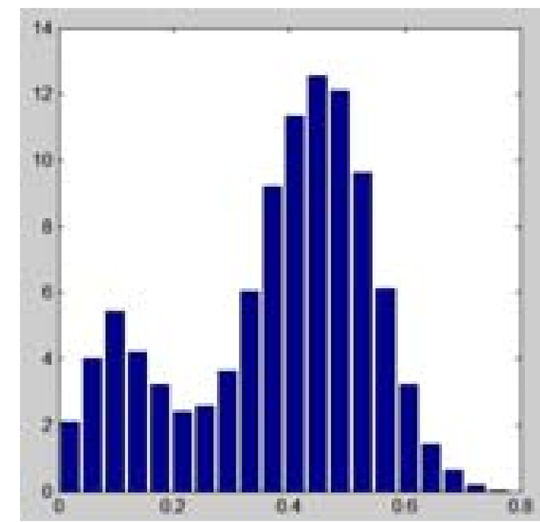
Y’ = f (Y)

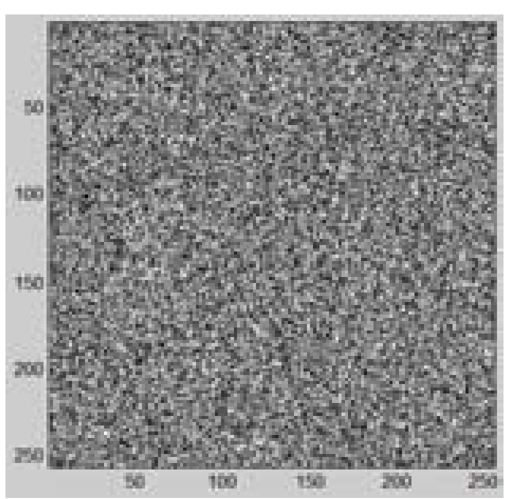


2. Histogram matching

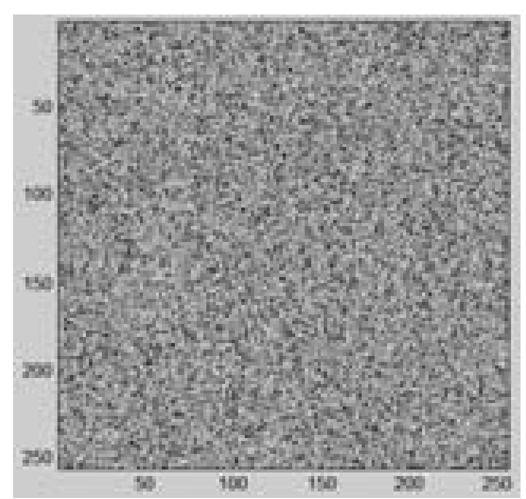
(1) pixels

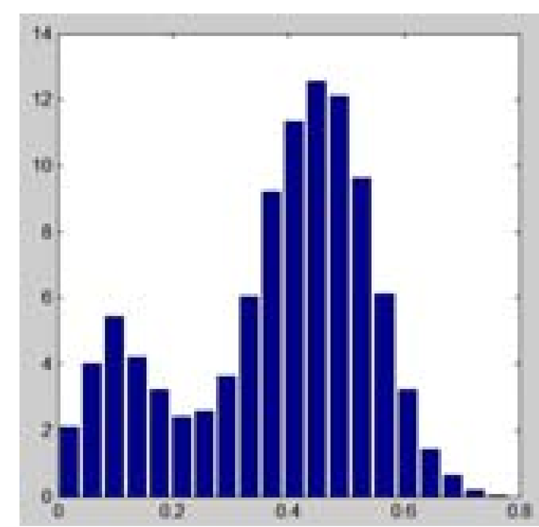






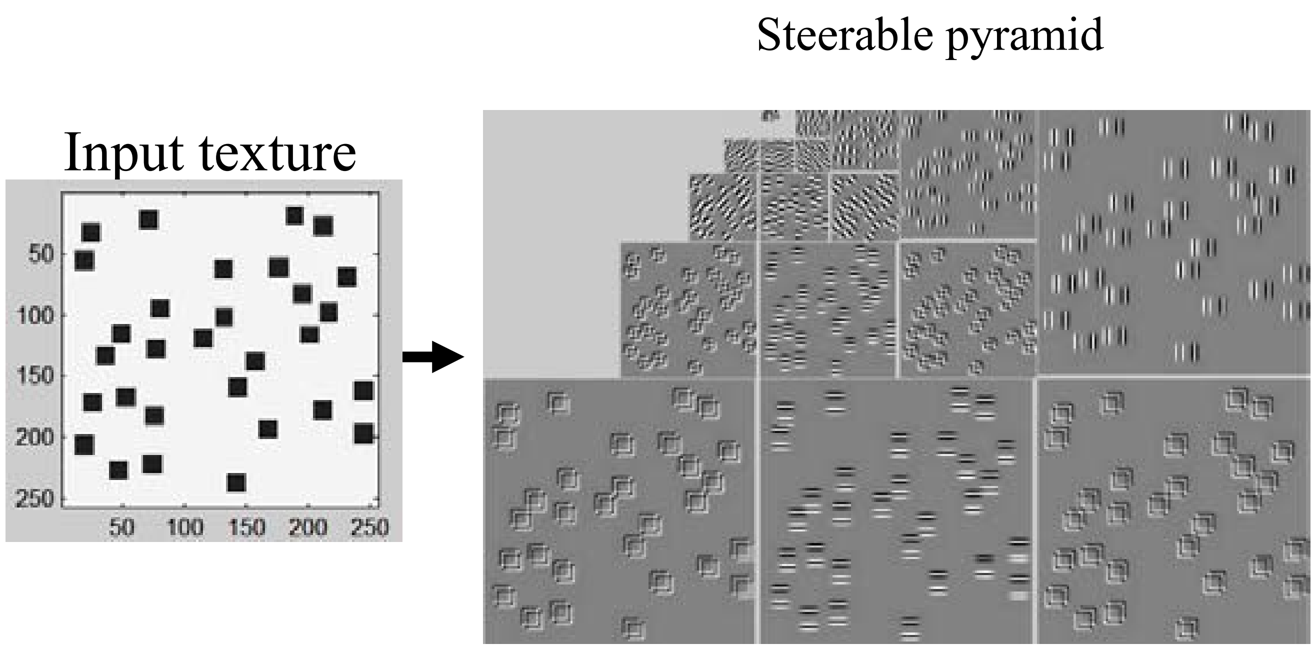
$Y' = f(Y)$

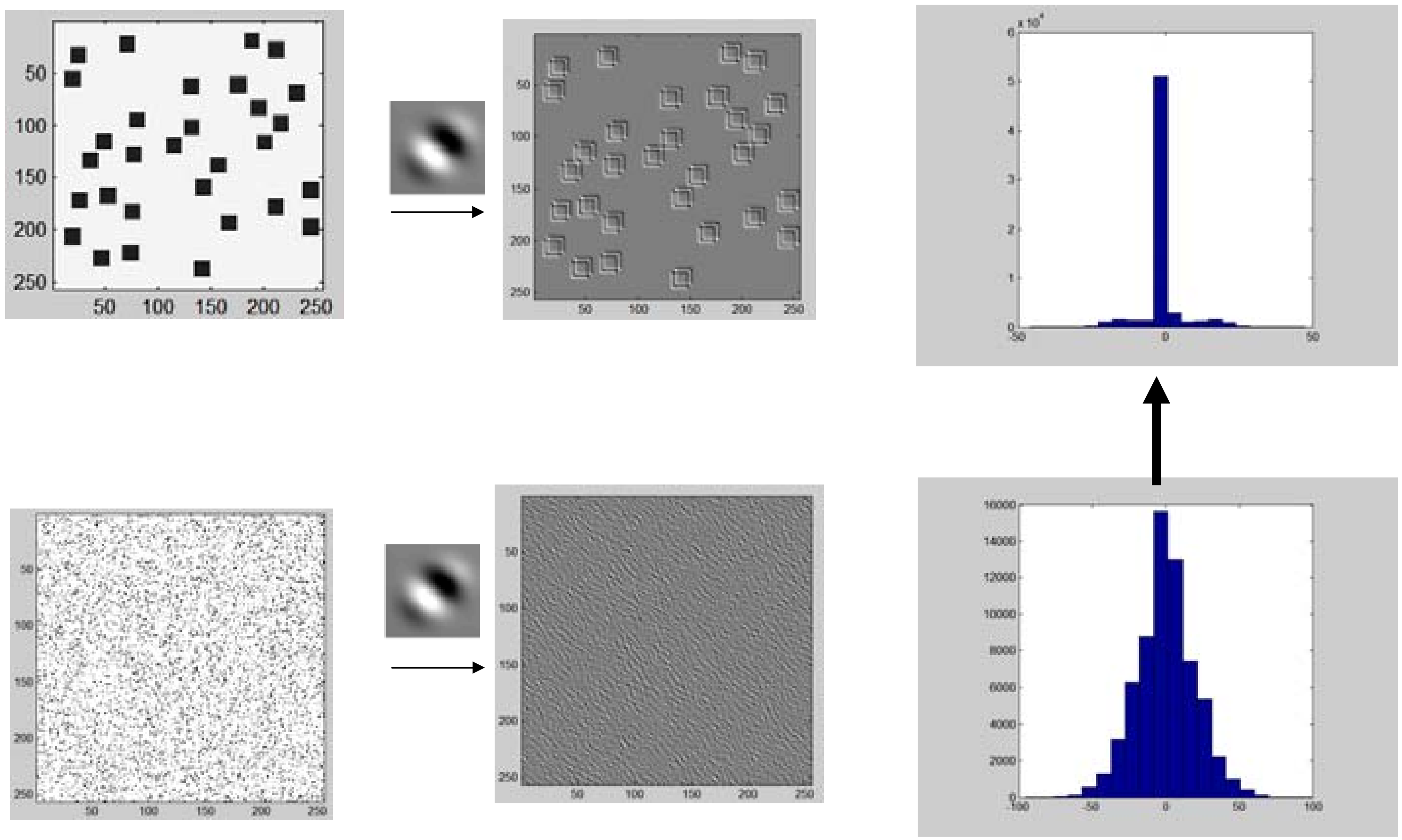




2. Histogram matching

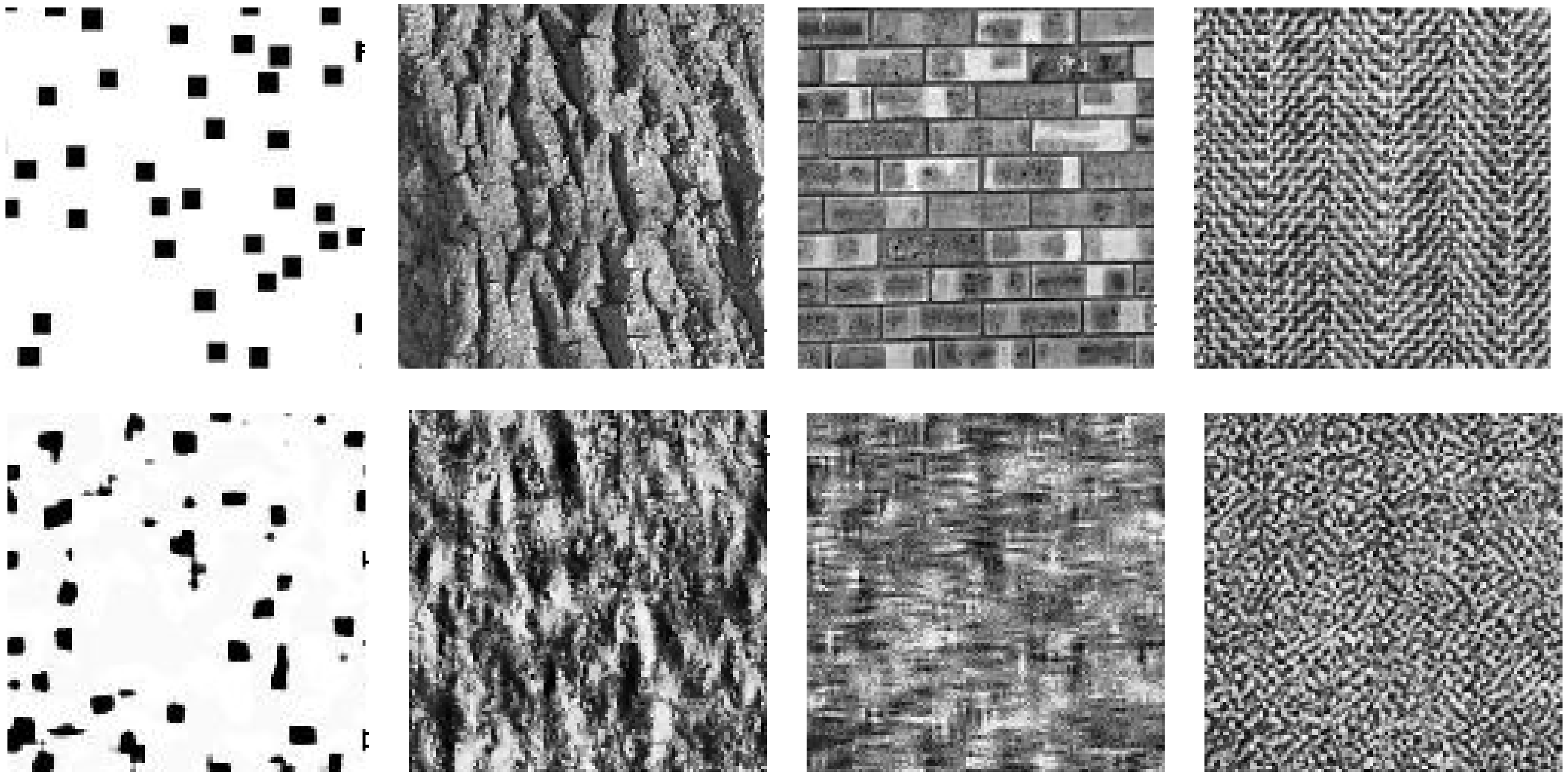
(2) each subband





Results

Marginal histograms (Heeger-Bergen)



Portilla & Simoncelli '00

Contours statistics

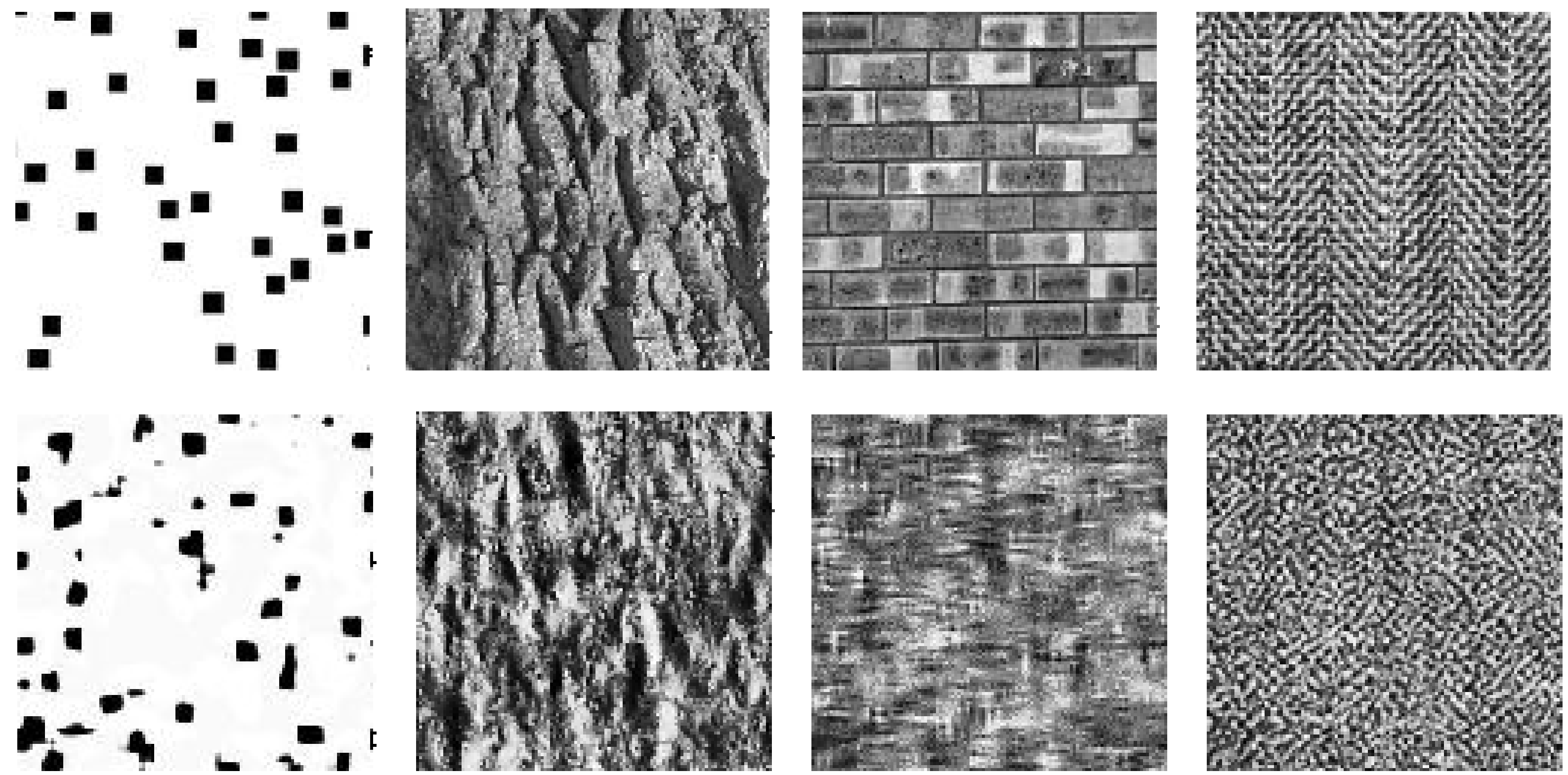
- Remember what we said about image statistics of neighboring orientations and contours grouping....

Demo

Geisler '01

Results

Marginal histograms (Heeger-Bergen)



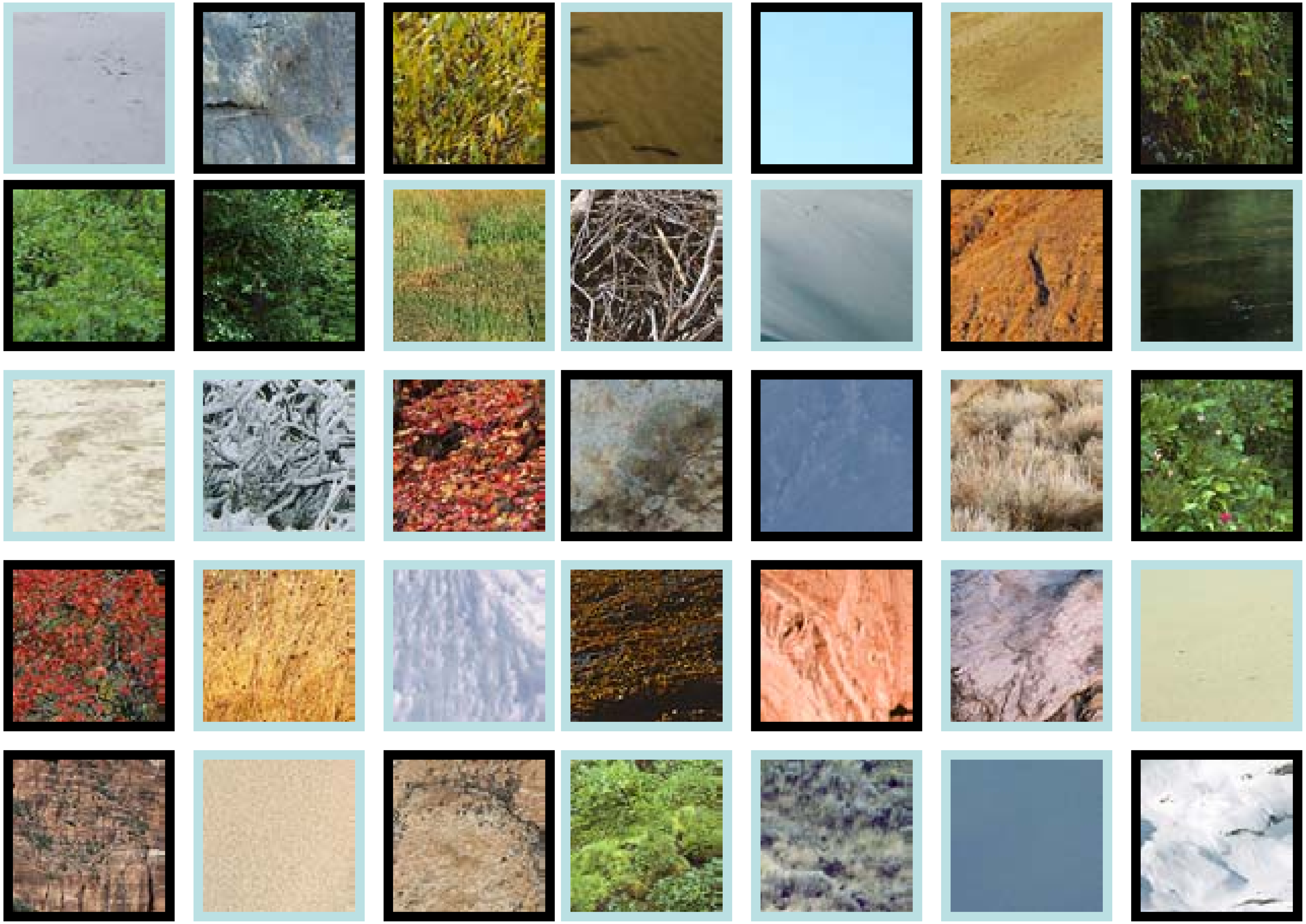
Portilla & Simoncelli ‘00

Computational Vision

Scenes, textures & surfaces

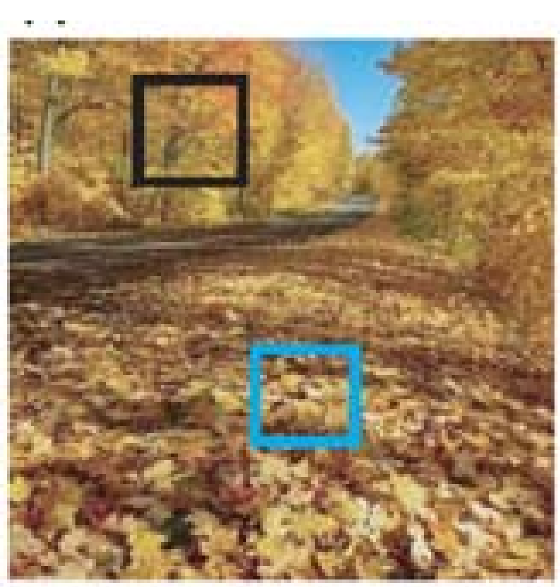
- Finish texture analysis
  - Texture synthesis
  - Structure-from-texture
- Start motion processing

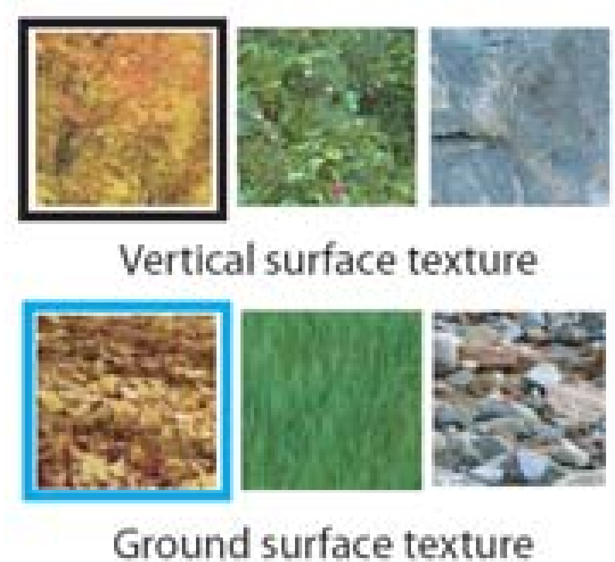
Vertical or ground surface?



source: Aude Oliva

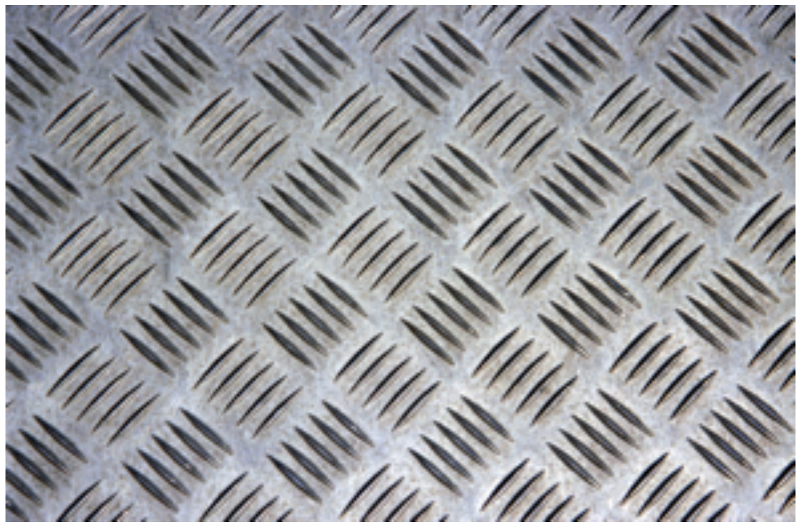
Local texture gradients

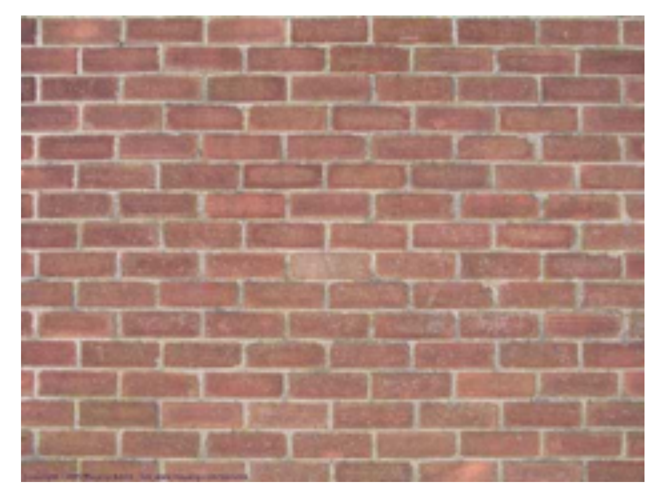




source: Aude Oliva

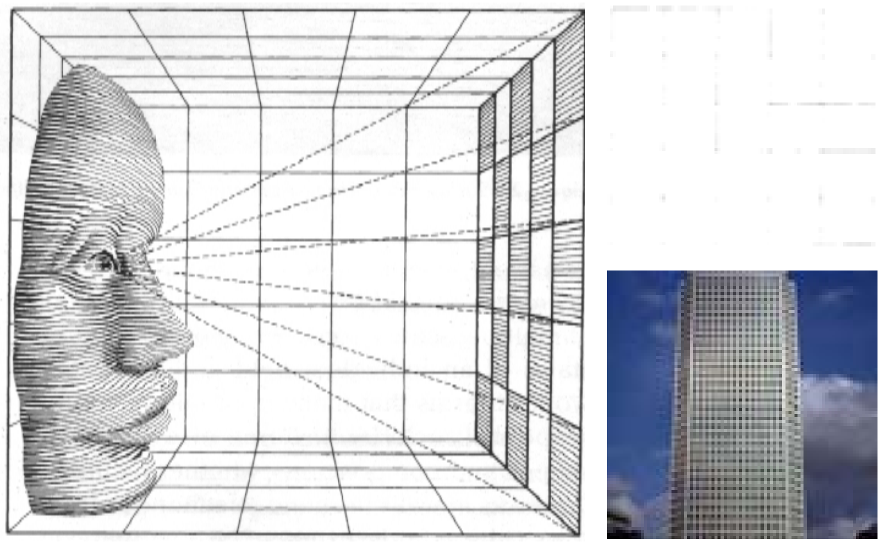
Representing texture elements: Texels



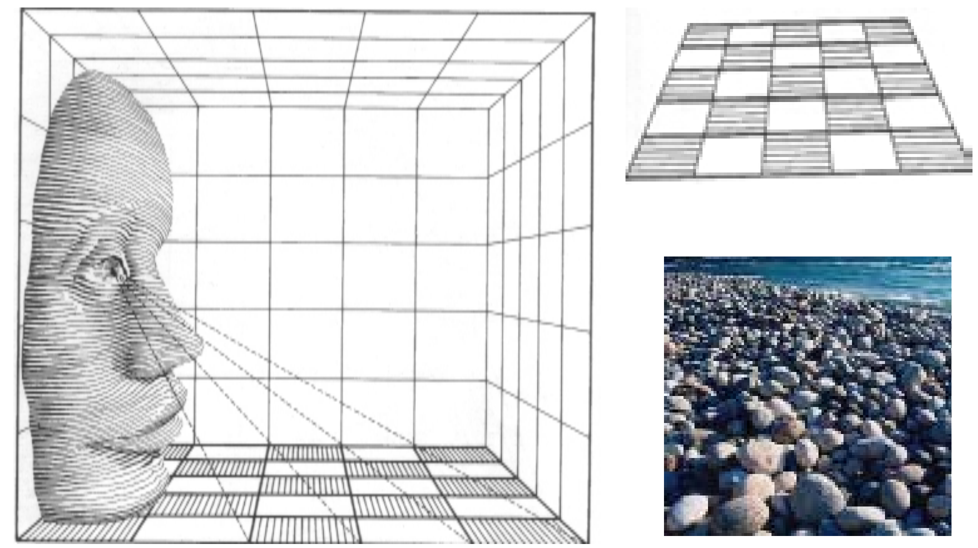


Surface orientation from texture gradient

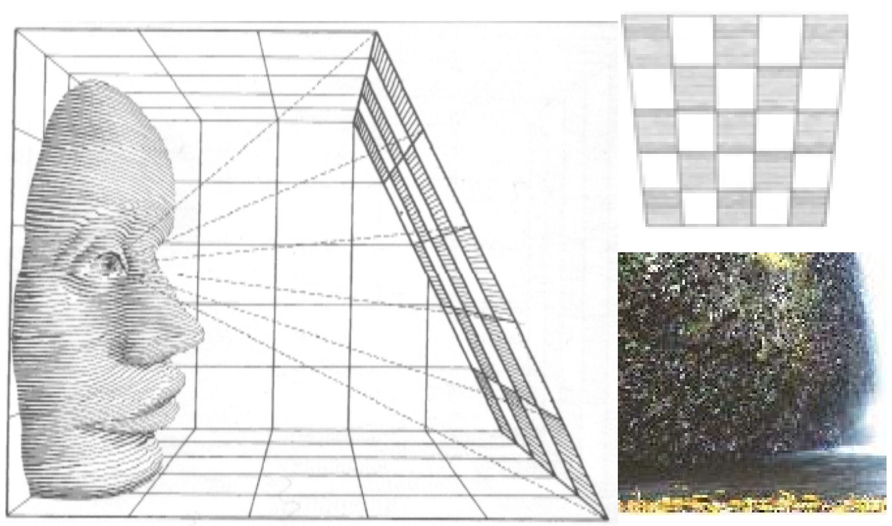
A flat frontal surface
Gradient is constant



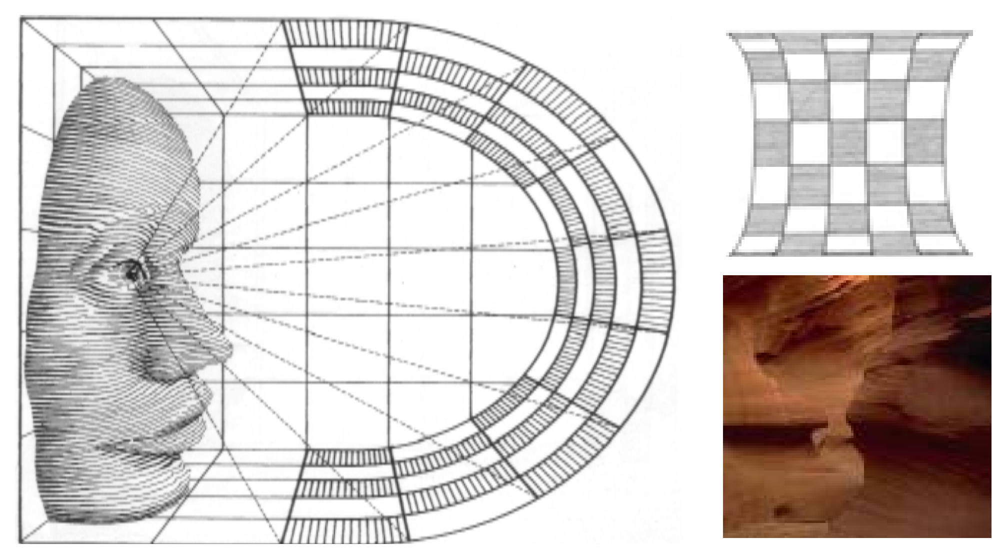
A flat longitudinal surface
Gradient decreases



A flat slanting surface
Gradient decreases



A rounded surface
Gradient changes from small to large to small
as the surface curves from a longitudinal
to a frontal



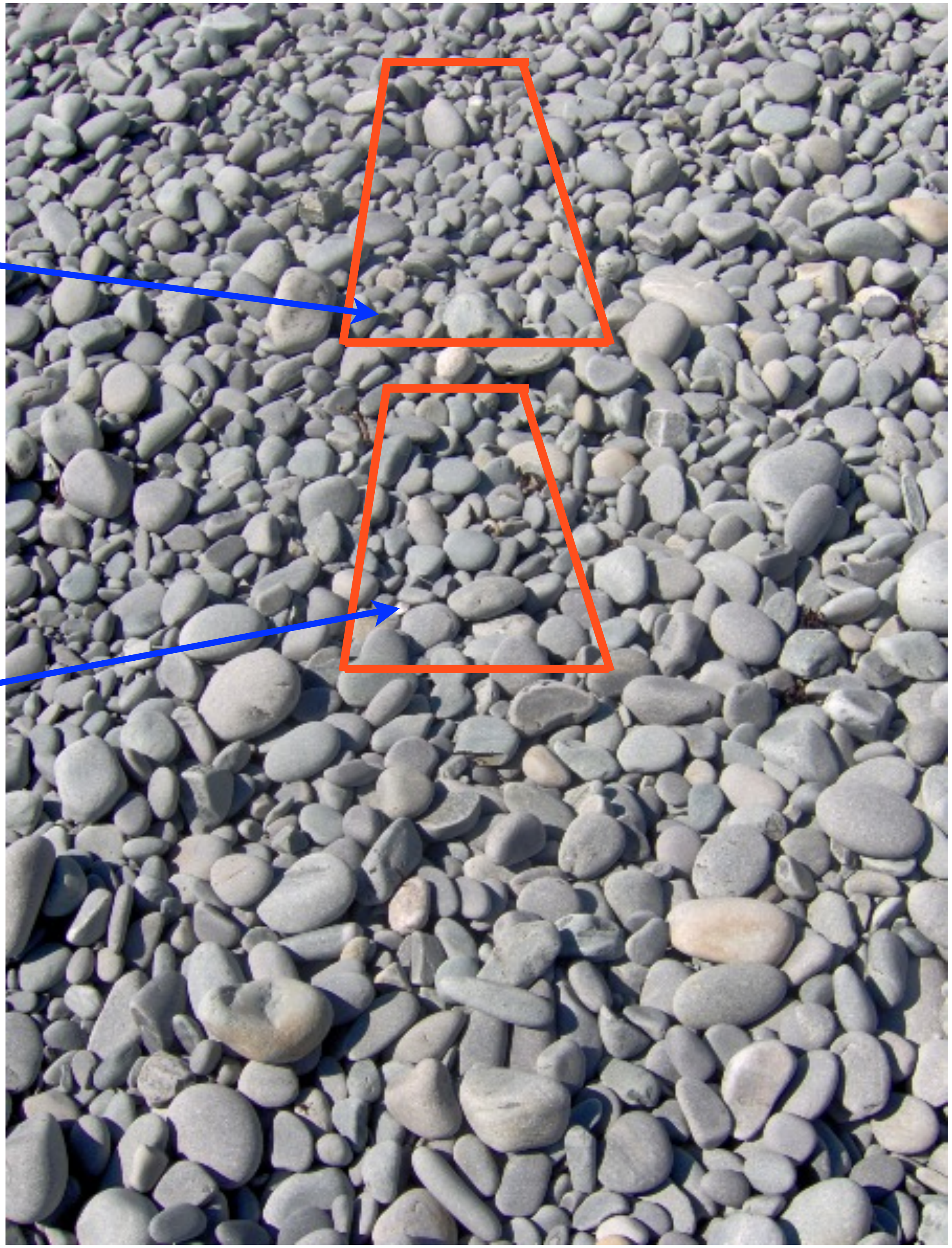
source: Aude Oliva

Cues to surface orientation

- Texture ‘gradients’
  - Texture density

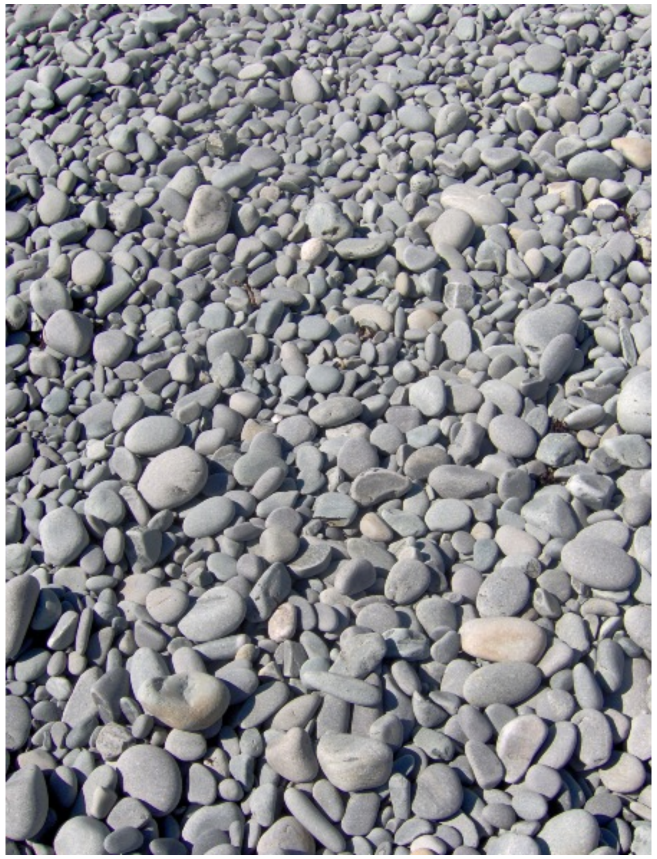
more stones

fewer stones



Assumptions: Textures are homogenous

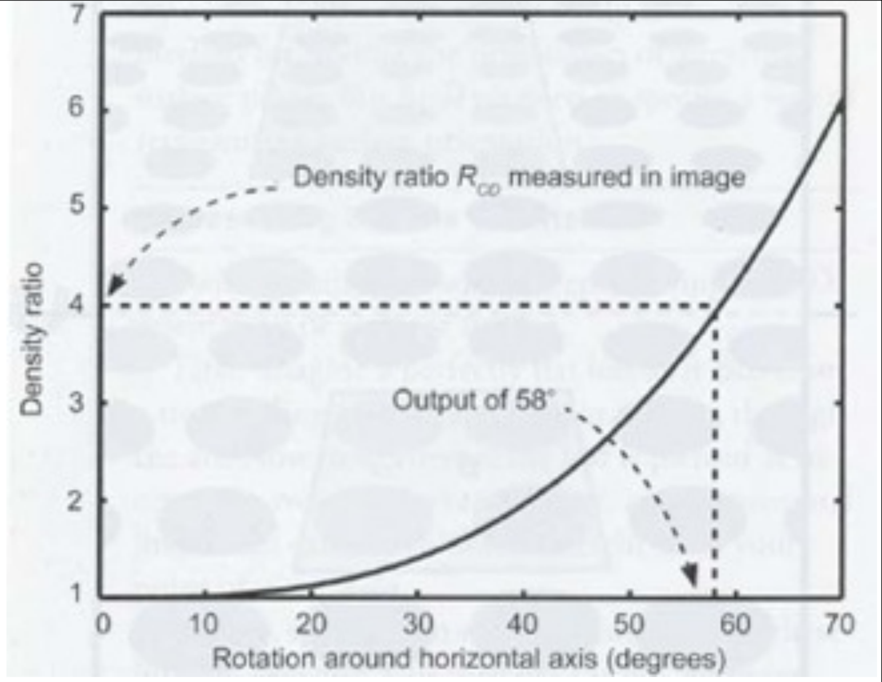
Next door beach



Mars

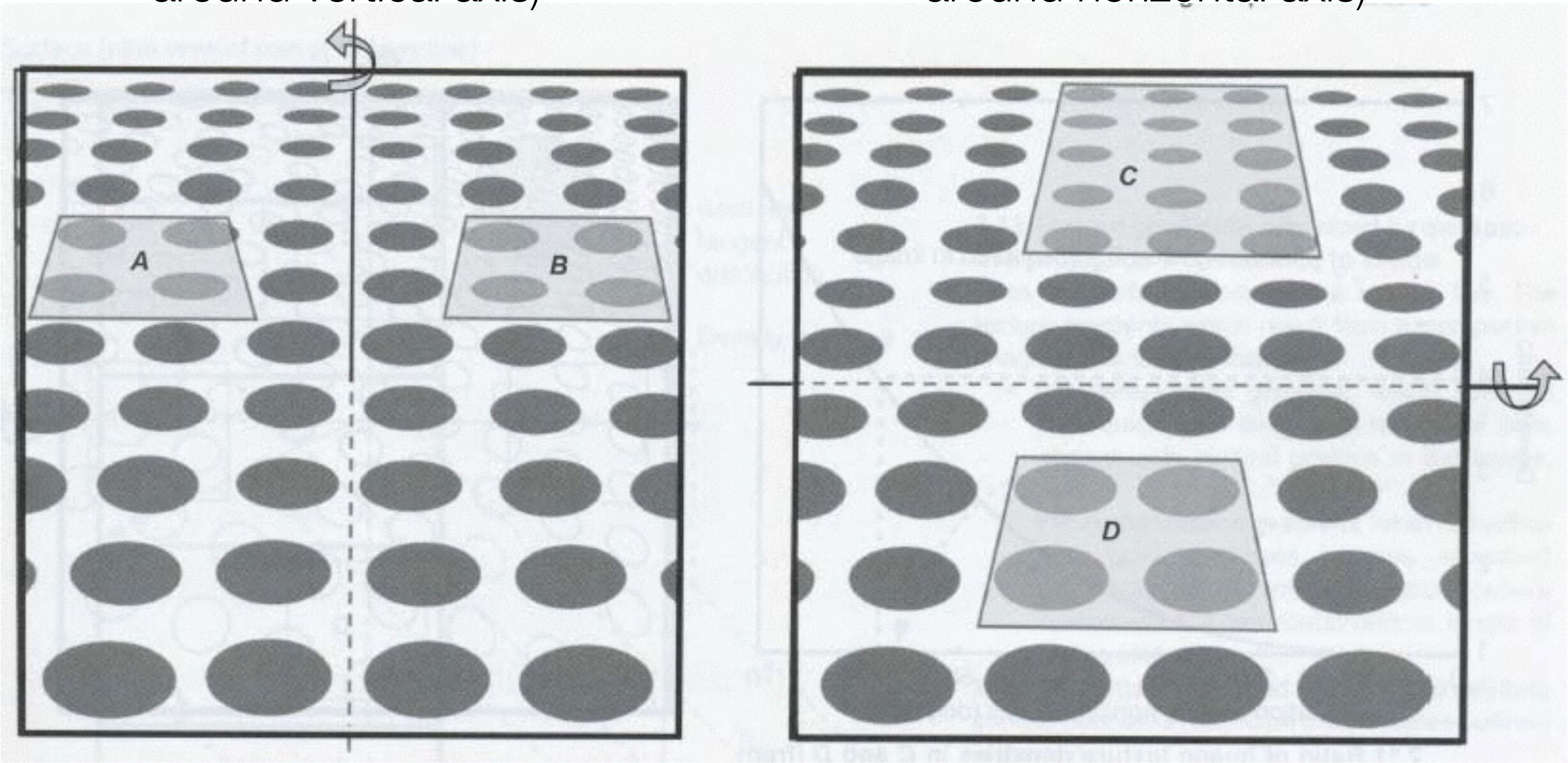
Surface orientation = f(# texture elements)

Finding p and q: An algorithm



Finding p (rotation around vertical axis)

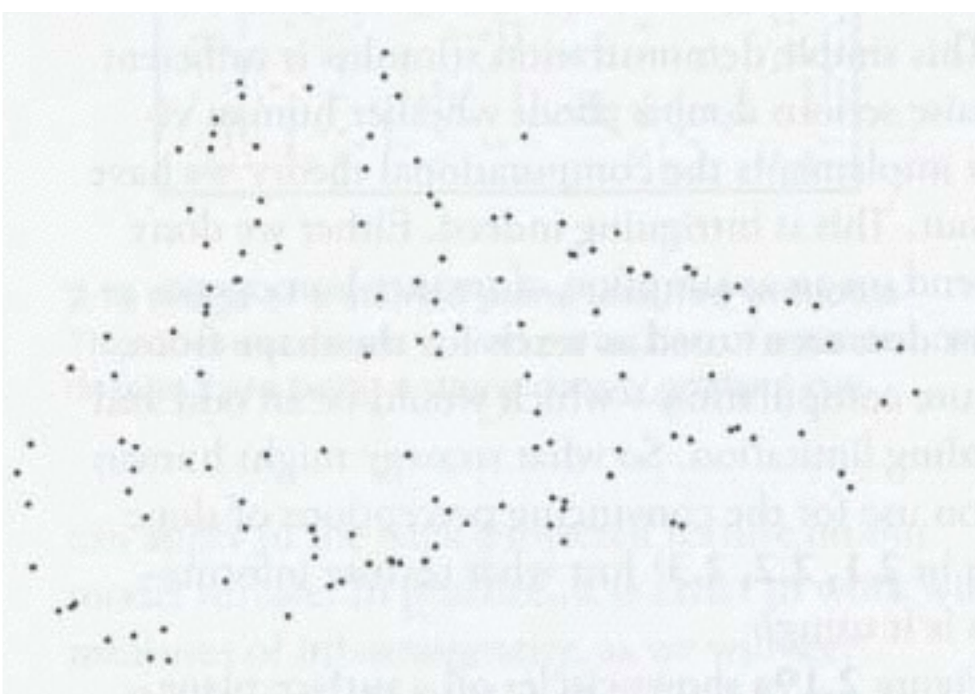
Finding q (rotation around horizontal axis)



Source: Seeing: The computational approach to biological vision. Frisby & Stone. MIT Press.

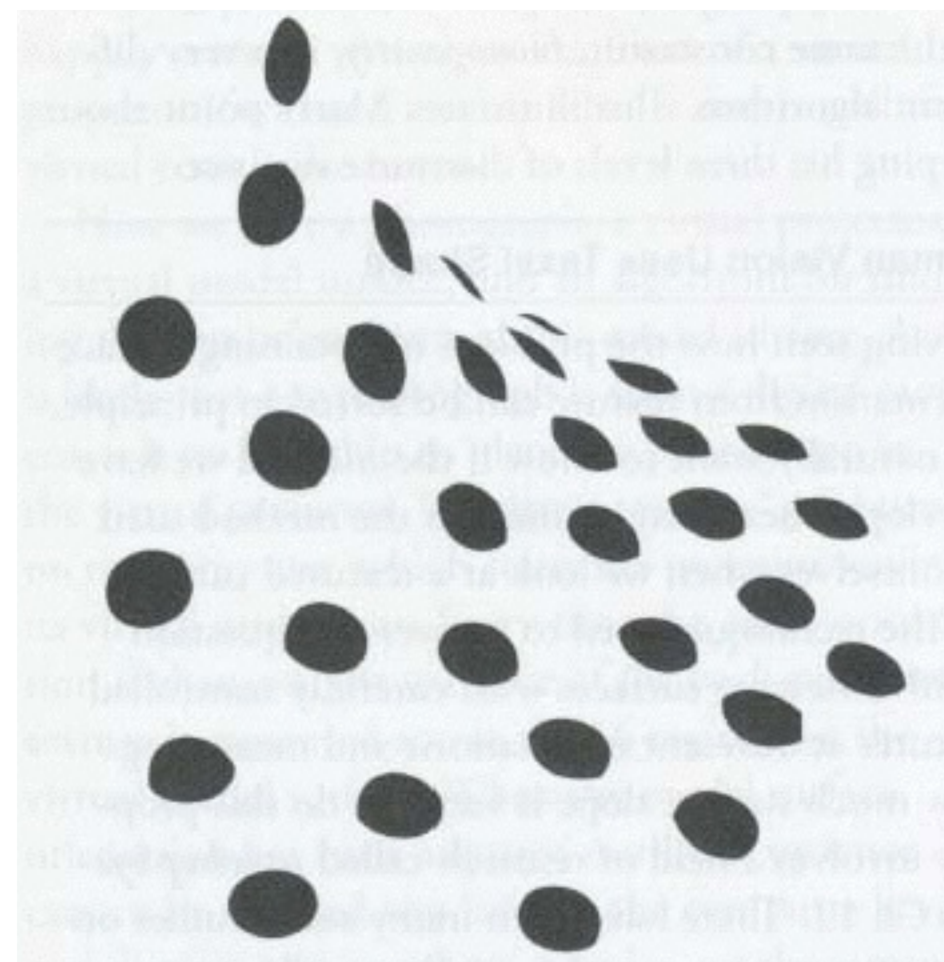
Beyond the homogeneity constraint

Texture gradients:
no shape information



VS.

Texture shape:
ellipses = slanted circles

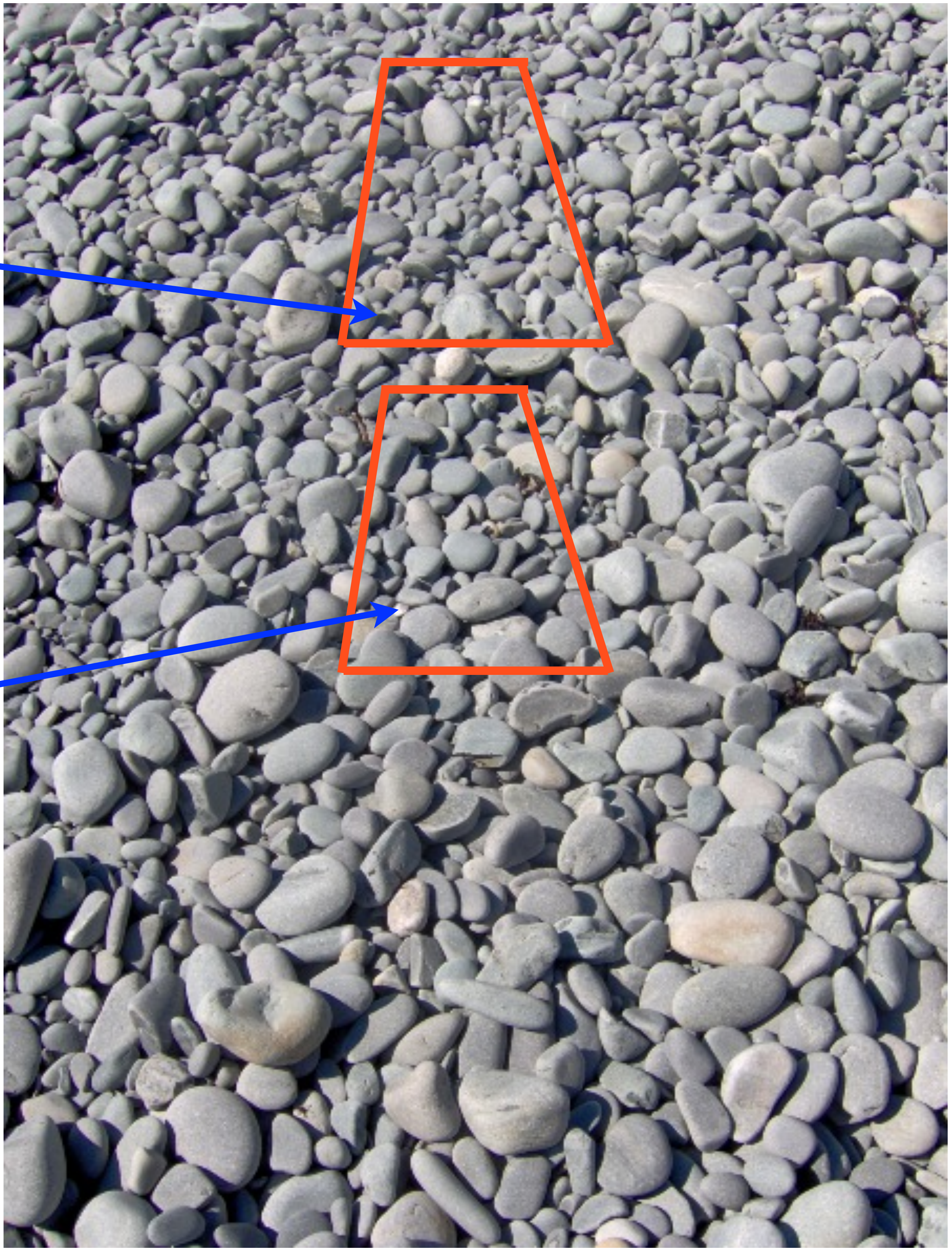


Cues to surface orientation

- Texture ‘gradients’
  - Texture density
  - Texture scale/area/spatial frequencies

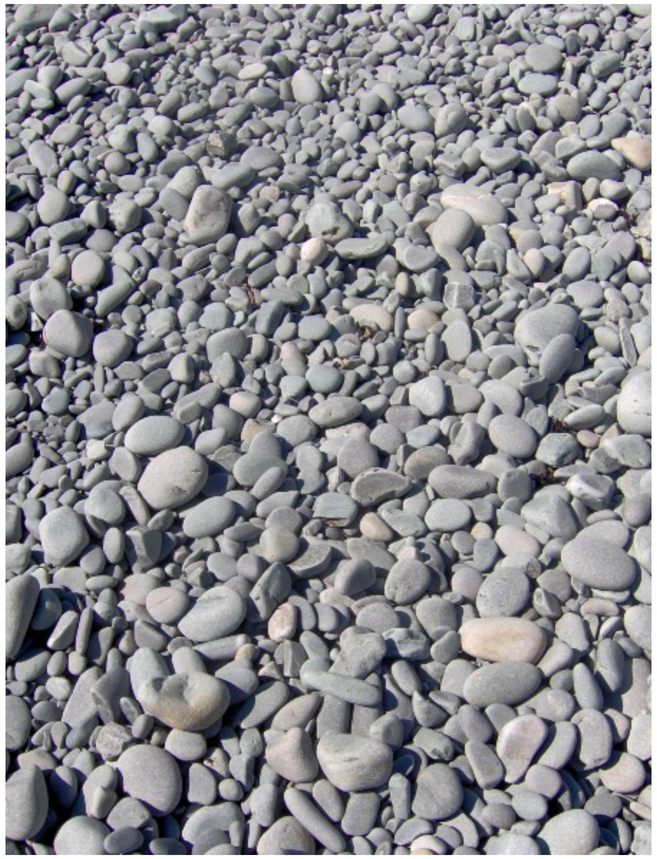
smaller stones

larger stones



Assumptions: Textures are isotropic

Next door beach

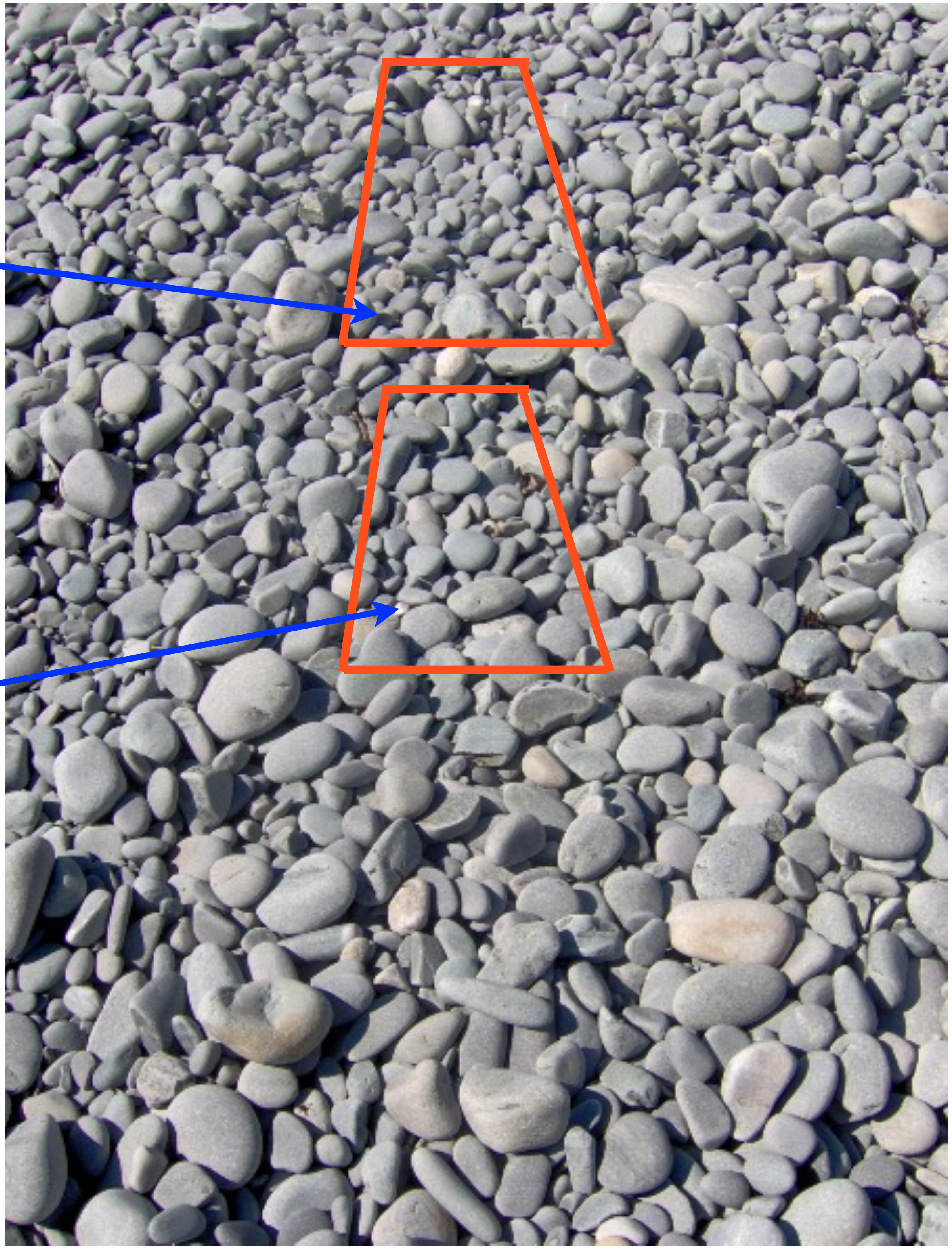


Mars

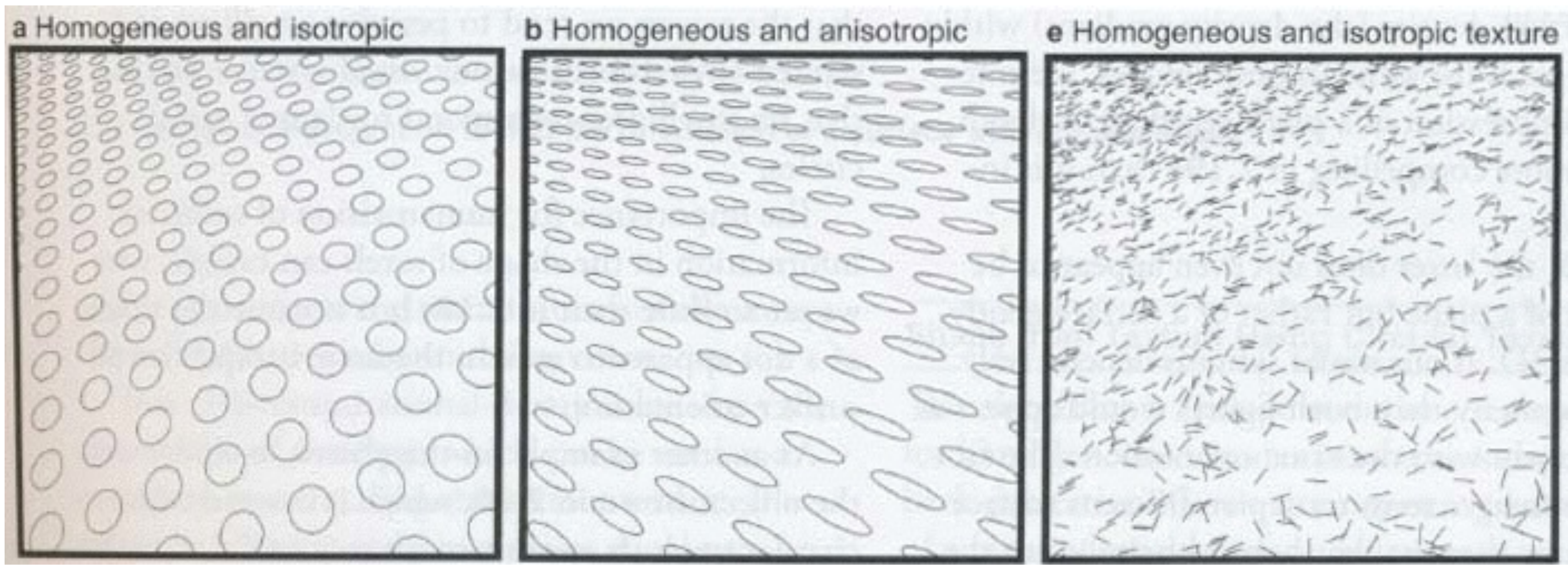
Surface orientation = f(Shape of the elements)

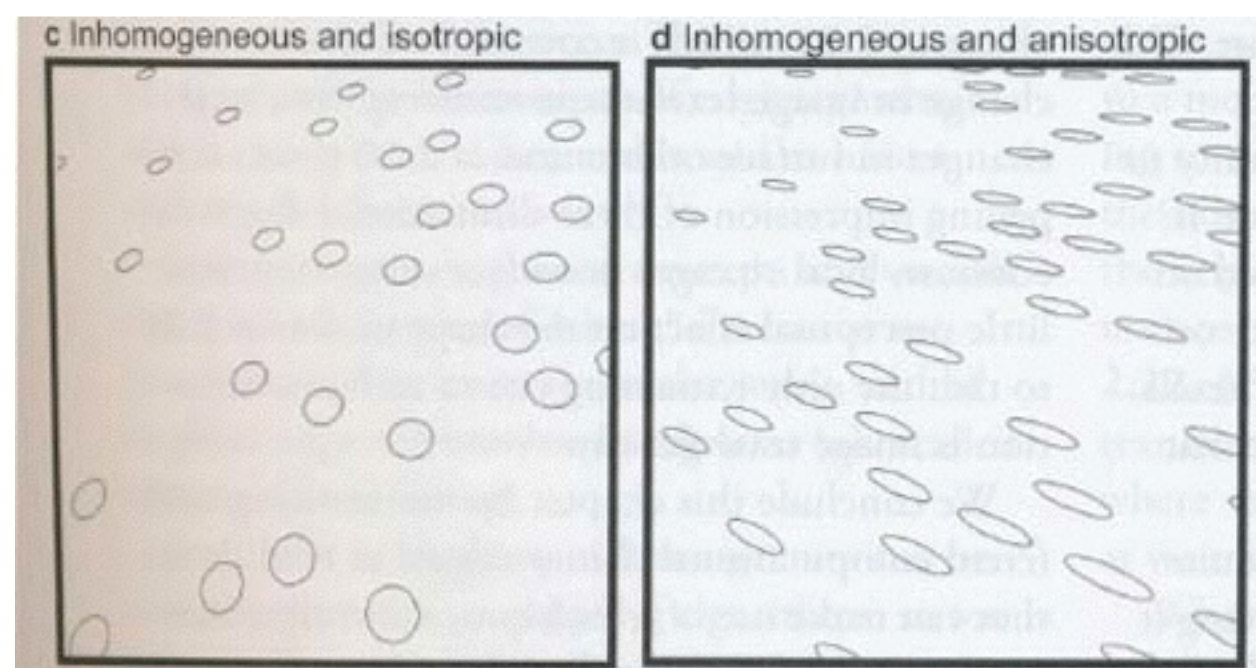
Cues to surface orientation

- Texture 'gradients'
  - Texture density
  - Texture scale/area/spatial frequencies
  - Perspective projection



Which strategy is being used?





Texture gradient vs. shape

- Short answer: Both

Surface Orientation from Texture: Isotropy or Homogeneity (or Both)?

RUTH ROSENHOLTZ,*‡ JITENDRA MALIK†

Received 23 June 1995; in revised form 9 April 1996

We examine two models for human perception of shape from texture, based on two assumptions about the surface texture: isotropy and homogeneity. Observers made orientation judgments on planar textured surfaces. Surface textures were either isotropic or anisotropically stretched or compressed. If subjects used an isotropy assumption, they would make biased orientation estimates for the anisotropic textures. In some conditions some observers showed no bias for the anisotropic textures relative to the isotropic textures. In general, even when the observers showed bias, the biases were significantly less than those predicted if the observer used only deviation from isotropy as a cue. Observers appear to use both the deviation from isotropy and a texture gradient or affine texture distortion cue for shape from texture. © 1997 Elsevier Science Ltd.

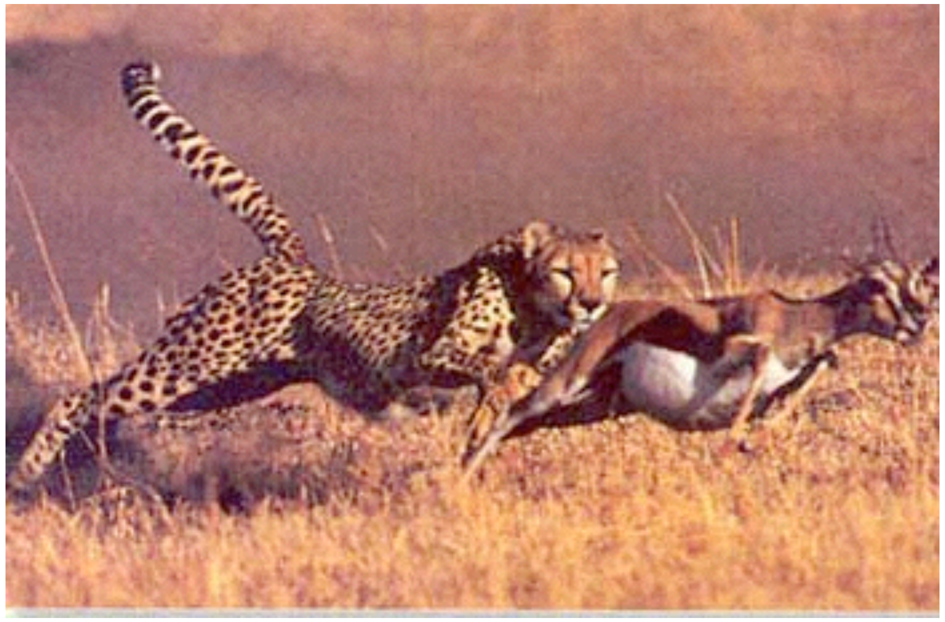
Shape from texture Homogeneity Isotropy Texture

Computational Vision

Scenes, textures & surfaces

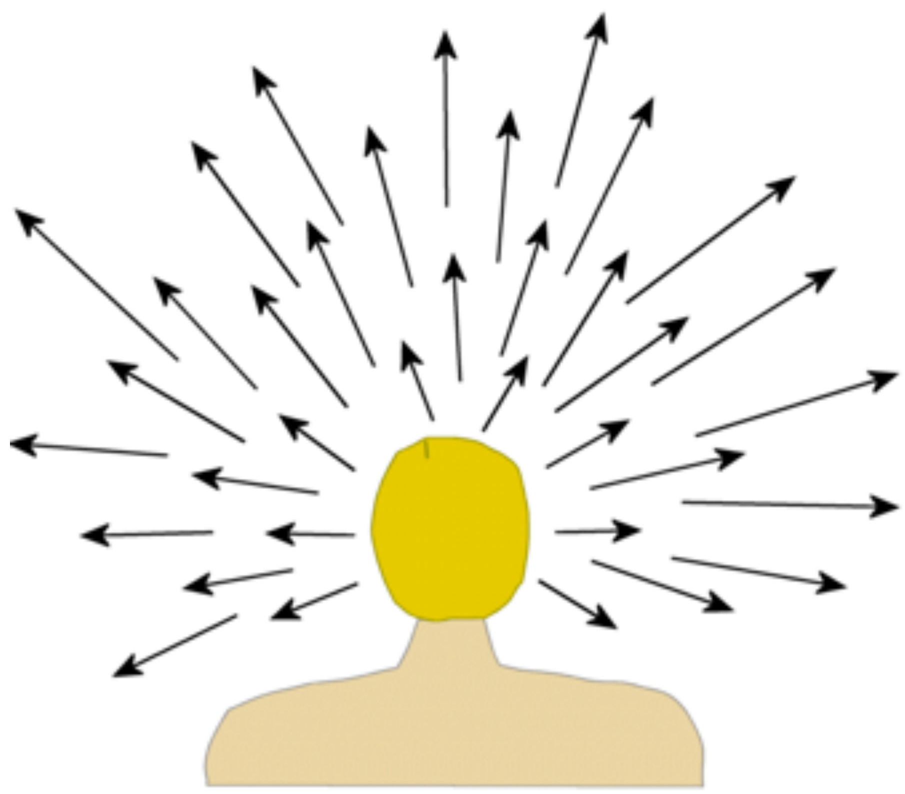
- Finish texture analysis
  - Texture synthesis
  - Structure-from-texture
- Start motion processing

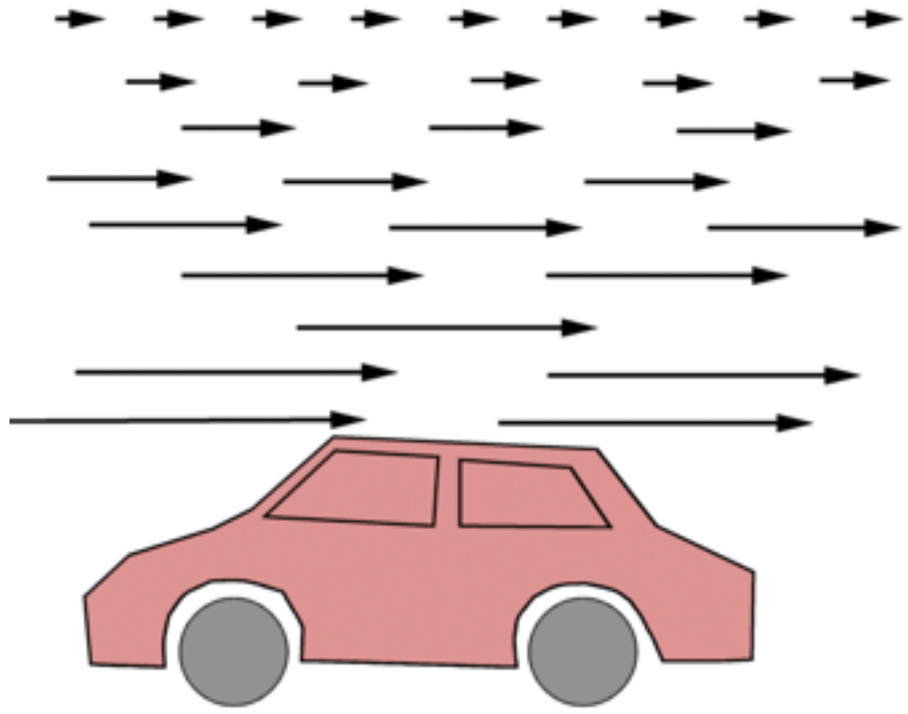
Motion as the most primitive form of vision



Motion perception in humans

Motion perception in humans





http://psych.hanover.edu/krantz/motionparallax/motionparallax.html

Seeing without motion

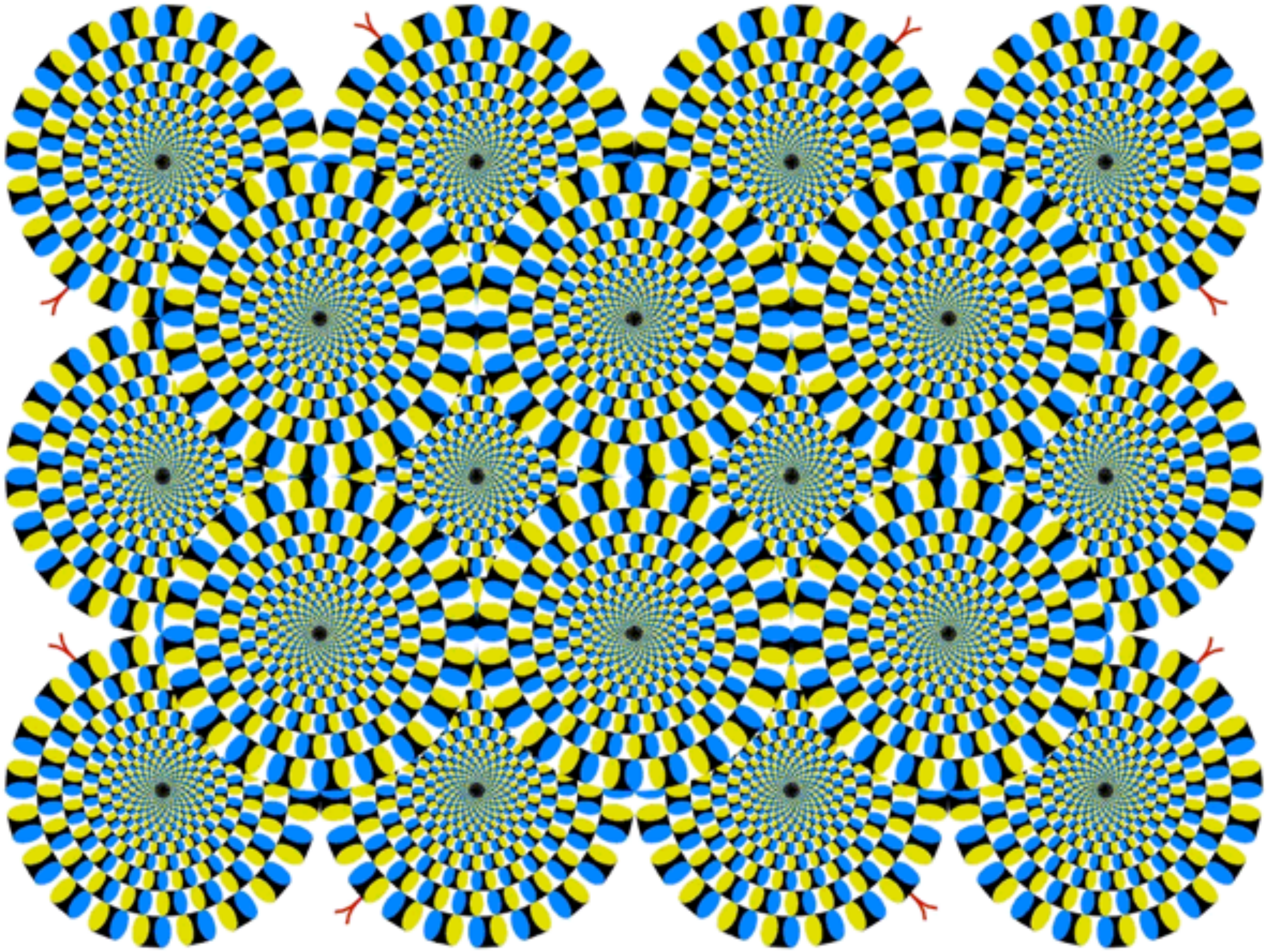
- Patient LM: 43 yr old, stroke with bilateral posterior parietal and occipital regions
- Complete loss of motion perception
- Comment from the scientists who have been working with her for years [Zihl et al ’83]:
    - She had difficulty for example, in pouring tea or coffee because the fluid appeared to be frozen like a glacier [...]
    - In a room where more than two people were walking [...] she usually left the room because:
        - “people were suddenly here or there but I have not seen them moving”

Computational problem of motion processing

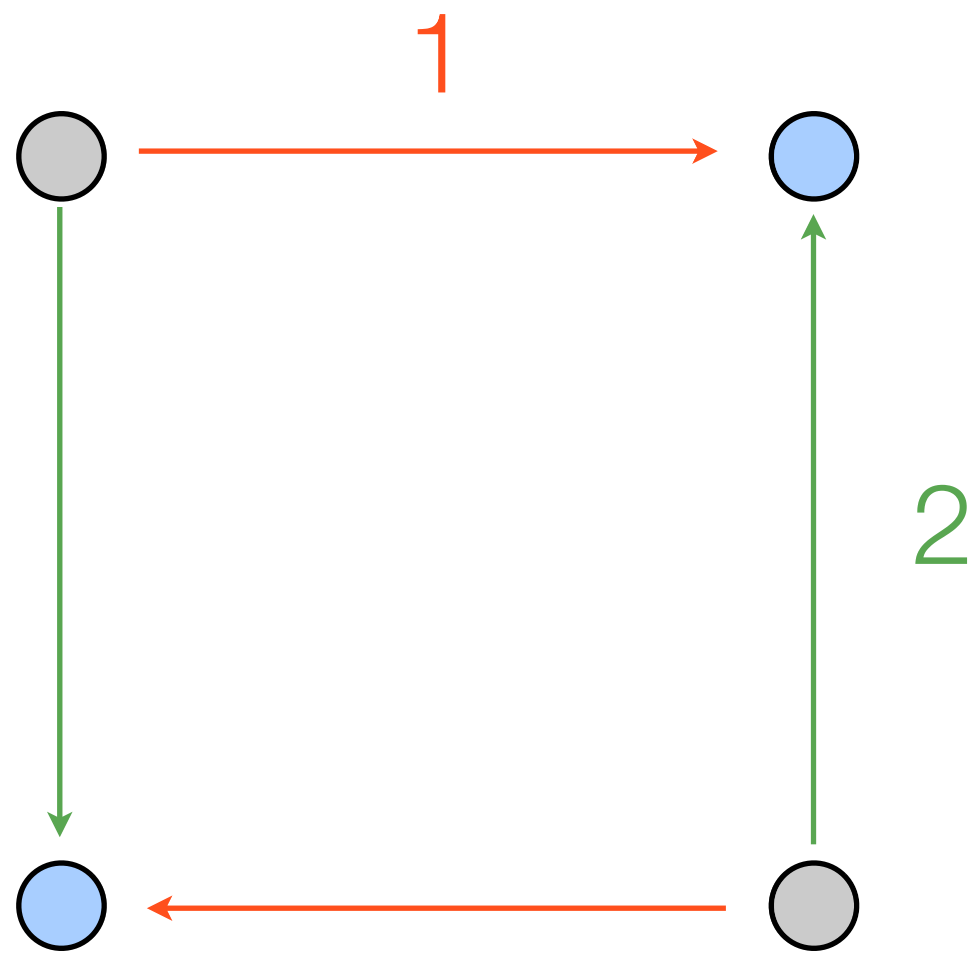
- How to get from the dynamic optical event on the retina (2D) to the veridical perception of moving objects (3D) within a generally stationary environment
  - Ill-posed problem! (not enough constraints for the solution to be unique)
- Hard problem because motion measured by photoreceptors on the retina can come from 2 sources:
  - moving objects
  - moving retinas (heads, eyes, body)
  - Moving objects can also appear static on the retina (think about smooth pursuit – object tracking by the eye of a moving object!)

Motion as a correspondence problem

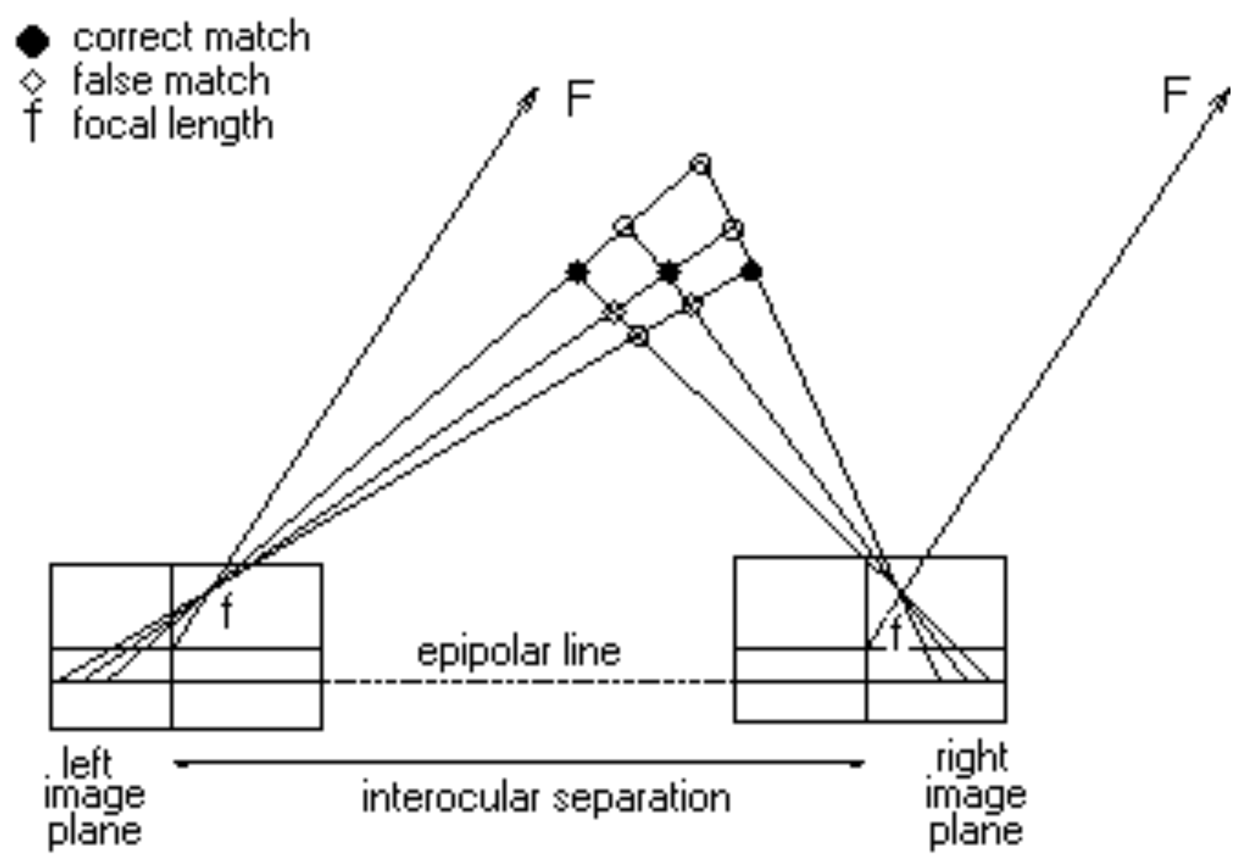
- Matlab demo
- see also: http://www.michaelbach.de/ot/mot_Ternus/index.html



Motion as a correspondence problem

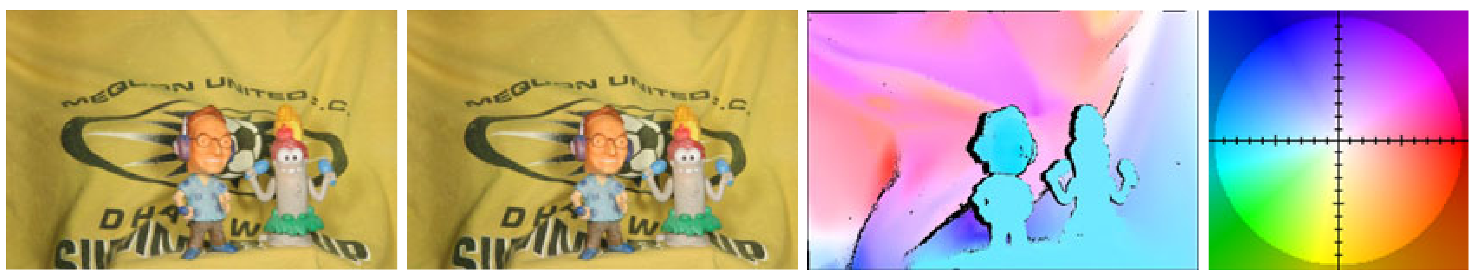


Another correspondence problem: stereo



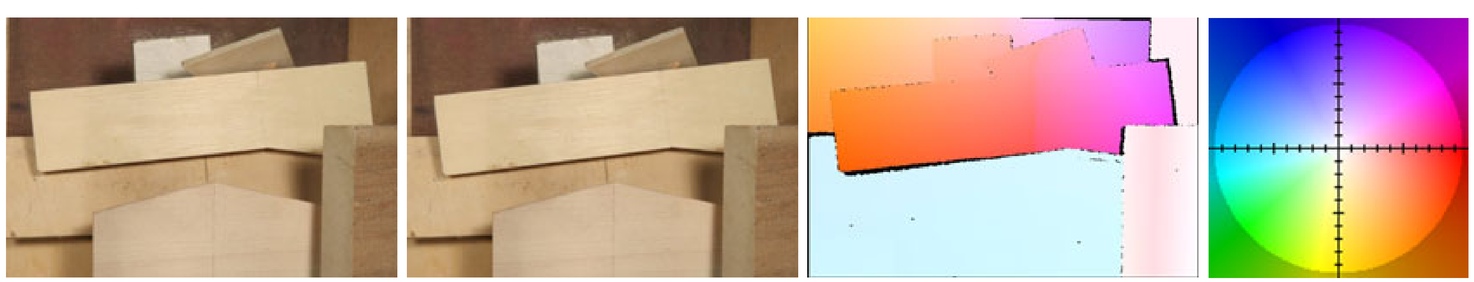
Optic flow

Army frame 0 | **Army** frame 1 | **Army** GT flow | flow color coding



Mequon frame 0 | **Mequon** frame 1 | **Mequon** GT flow | flow color coding

Schefflera frame 0 | **Schefflera** frame 1 | **Schefflera** GT flow | flow color coding

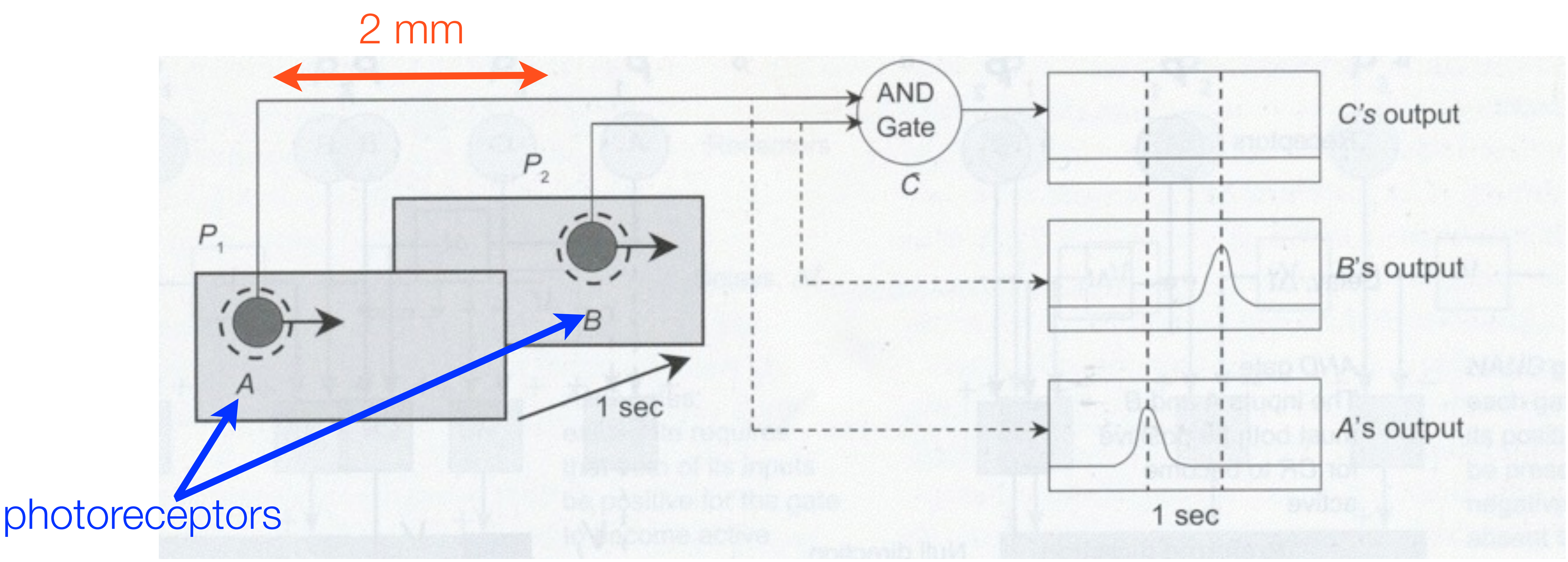


Wooden frame 0 | **Wooden** frame 1 | **Wooden** GT flow | flow color coding

Computational models of motion detection

Reichardt models (correlation models / delay-and-compare networks)
Marr-Ulllman (temporal derivatives / edge-based models)
Adelson-Bergen (spatial-frequency-based models)

Basic motion detector

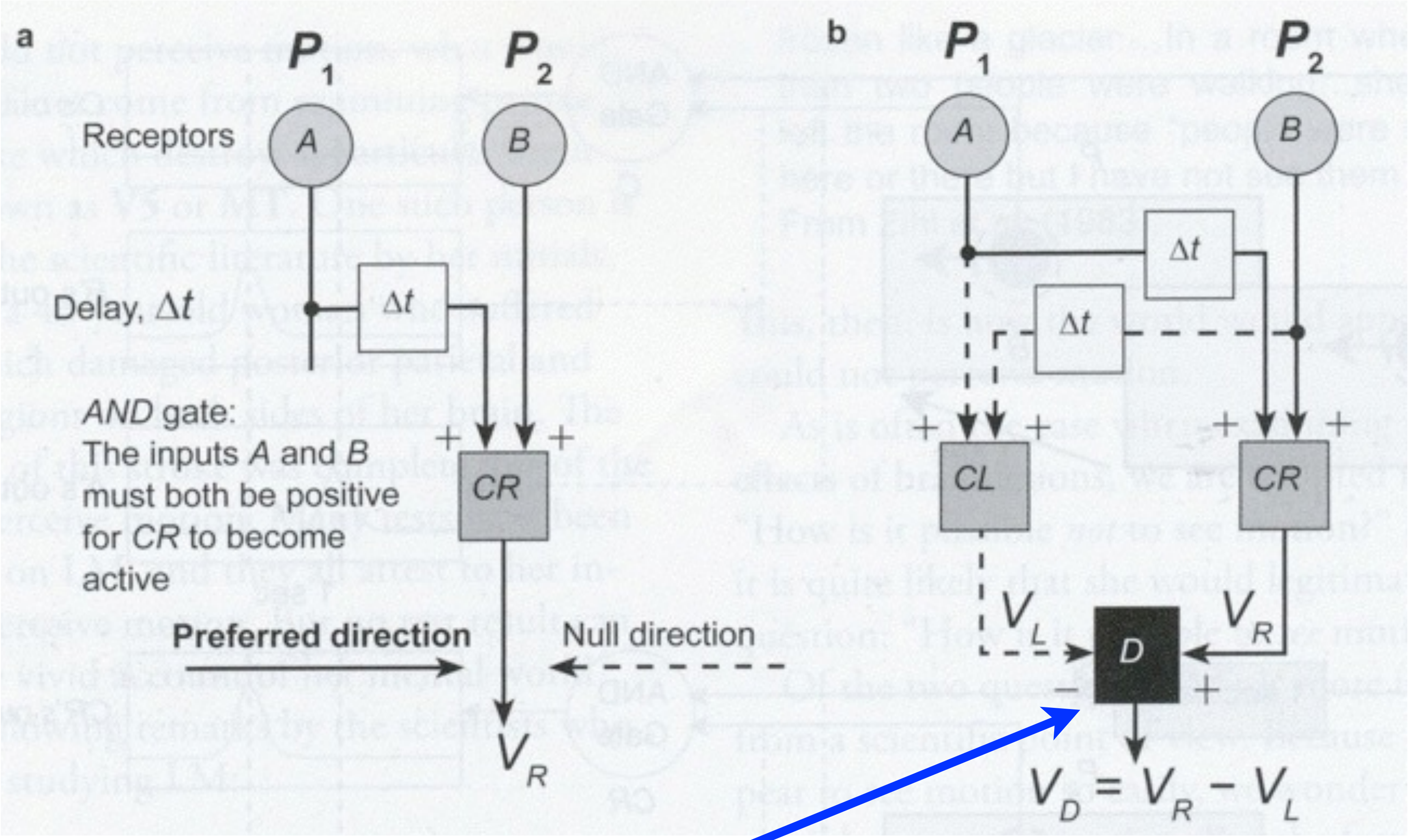


Basic motion detector

- Selective to:
    - speed
    - direction
- Very sensitive to noise in practice
- One way to reduce the influence of noise is to use ***opponent pairs*** of such detectors

Reichardt motion detector (1961)

- Initially a model of the optomotor response of the beetle

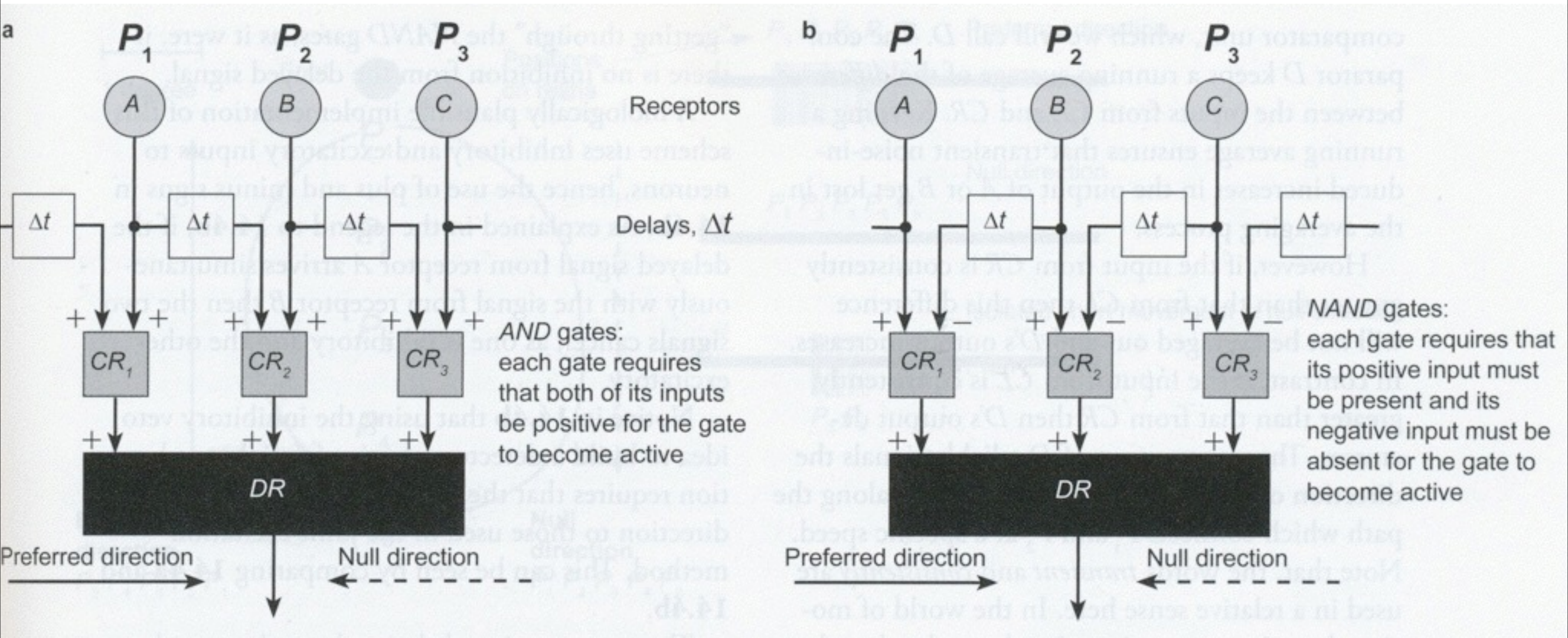


comparator unit (average out signals between CL and CR)

Beyond the Reichardt motion detector

Joint excitation method

Inhibitory veto



Barlow & Levick ’65 (model of the rabbit retina)