Visual Texture (in human and machine)

Somewhere in Cinque Terre, May 2005

CS194: Intro to Computer Vision and Comp. Photo
Alexei Efros, UC Berkeley, Fall 2022

What is Texture?

- Texture depicts spatially repeating patterns
- Many natural phenomena are textures


radishes

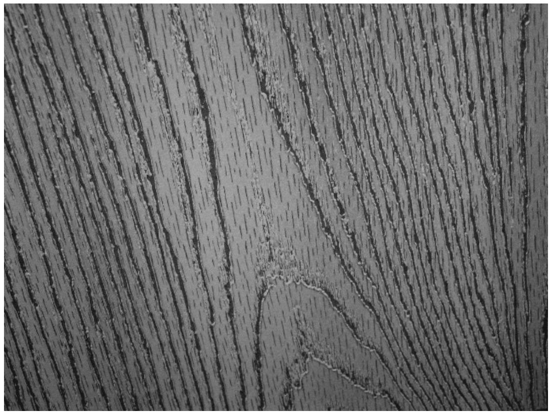

rocks

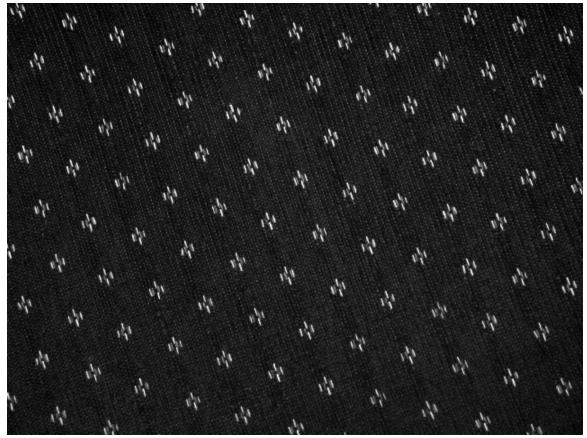

yogurt

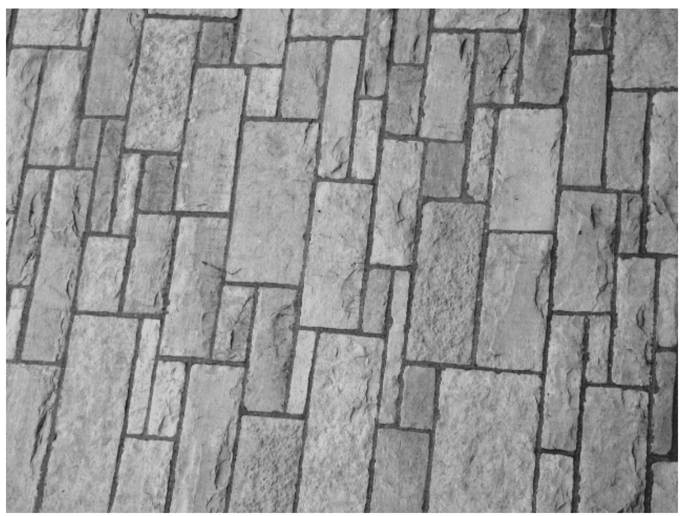
Texture as “stuff”

Source: Forsyth

Texture and Material

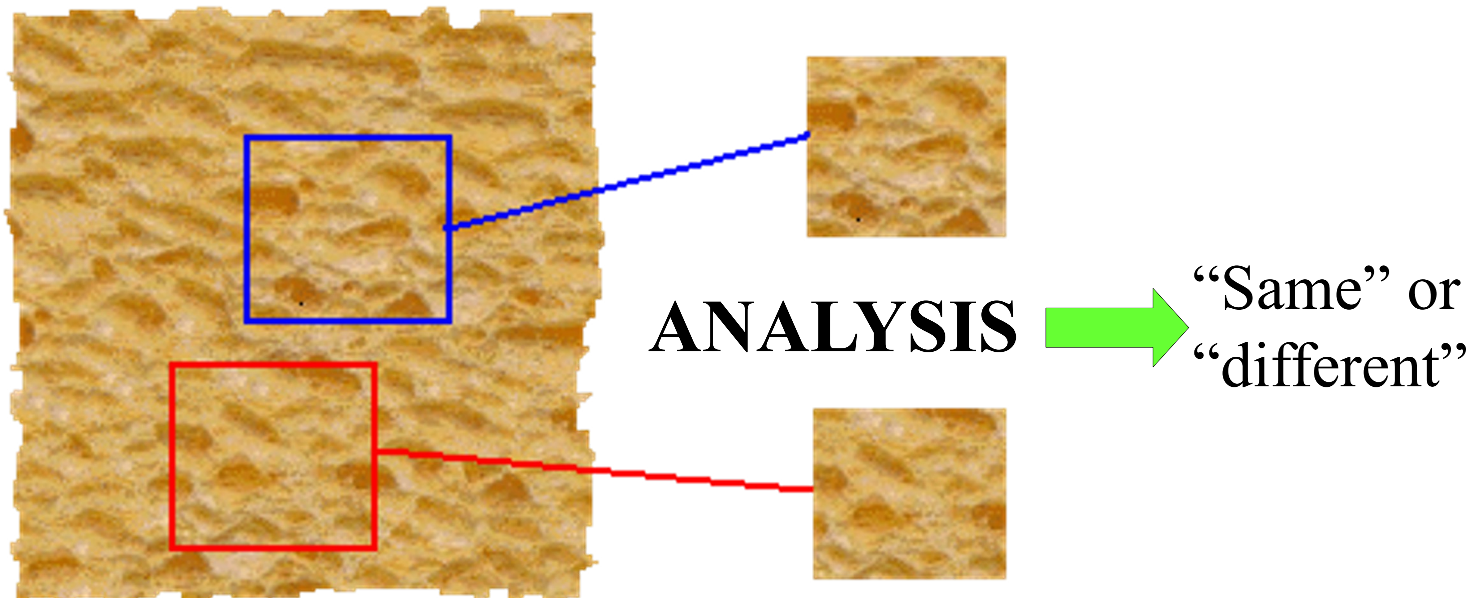






http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database/samples/

Texture Analysis



True (infinite) texture

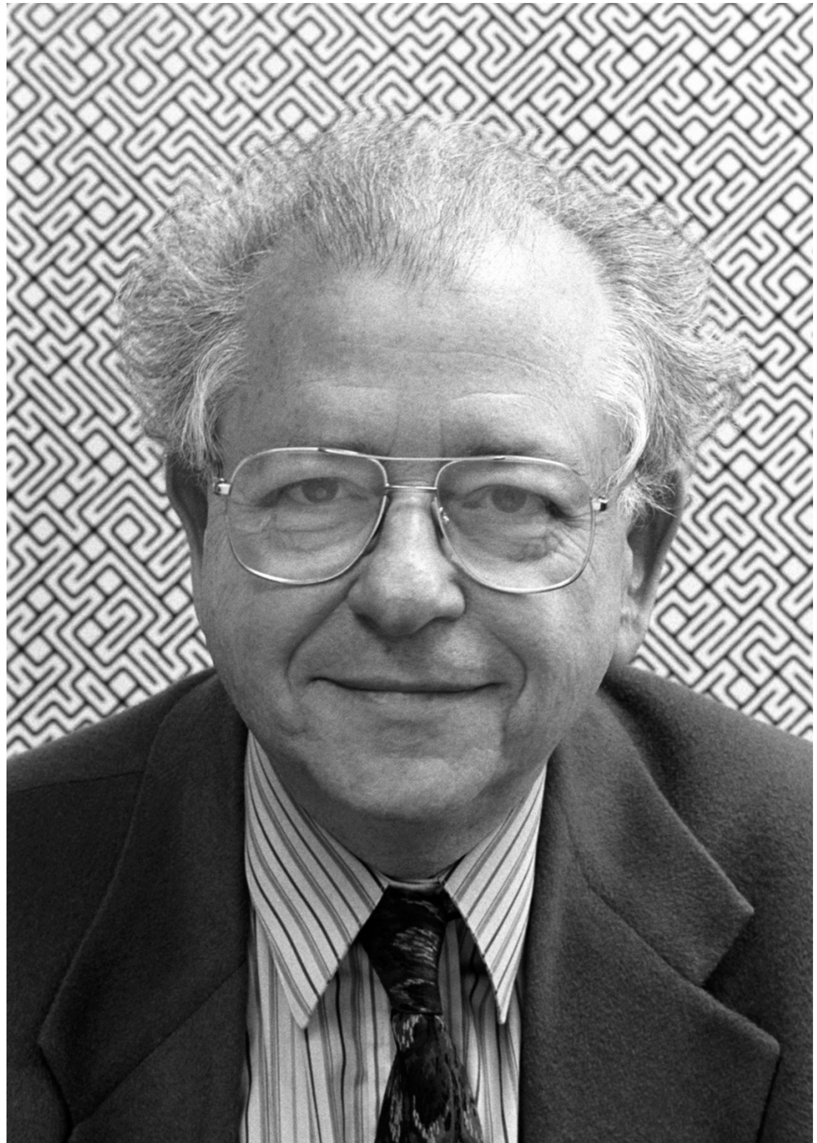
Compare textures and decide if they're made of the same "stuff".

When are two textures similar?

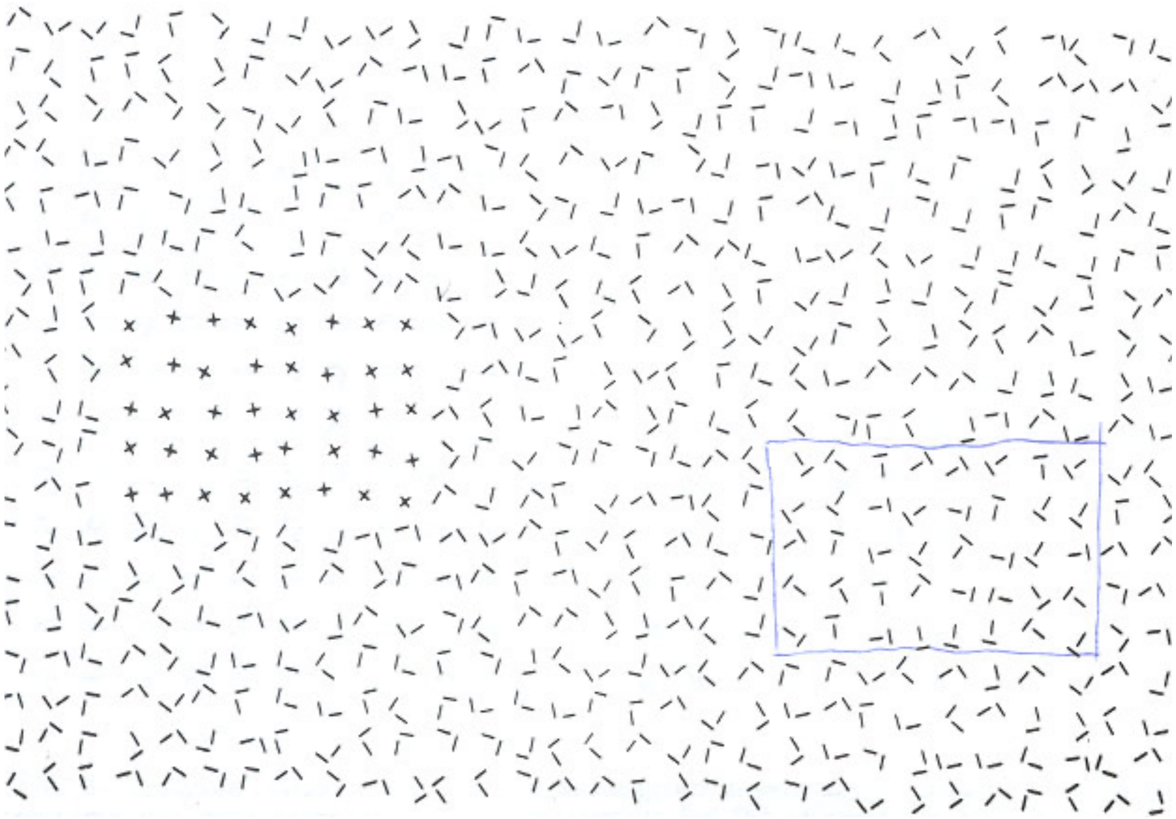




Béla Julesz, father of texture



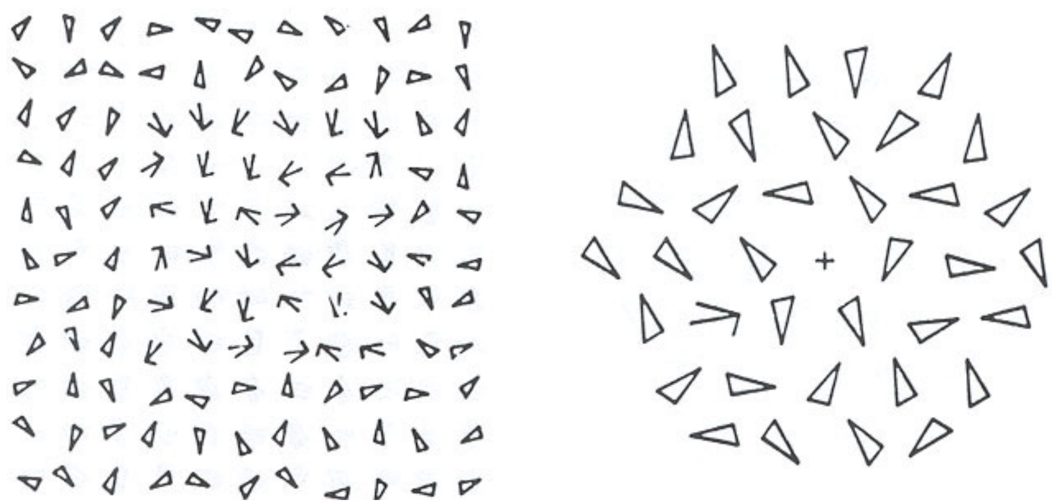
Texton Discrimination (Julesz)



Human vision is sensitive to the difference of some types of elements and appears to be "numb" on other types of differences.

Search Experiment I

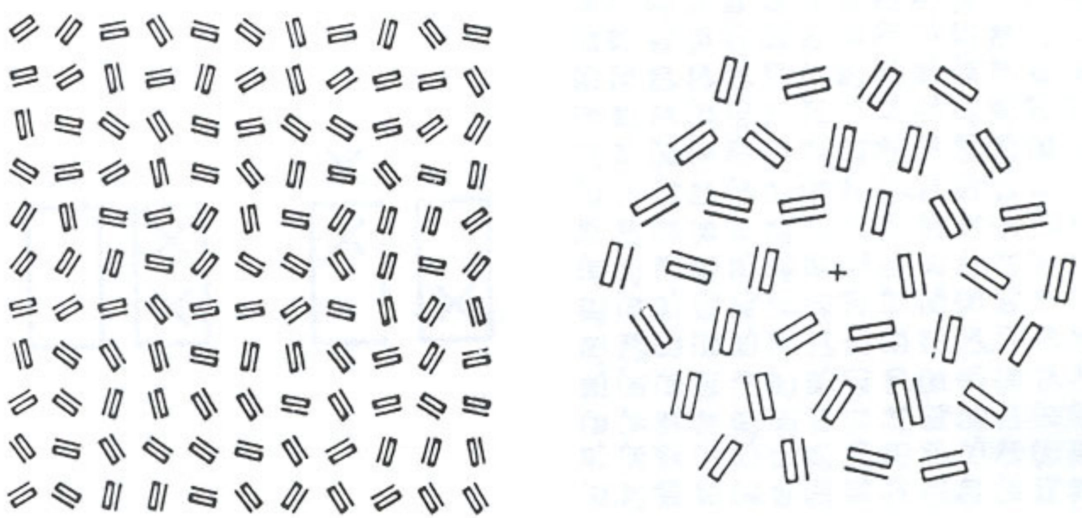
(a)



The subject is told to detect a target element in a number of background elements.
In this example, the detection time is independent of the number of background elements.

Search Experiment II

(a)



In this example, the detection time is proportional to the number of background elements,
And thus suggests that the subject is doing element-by-element scrutiny.

Preattentive vs Attentive Vision (Julesz)

Human vision operates in two distinct modes:

1. Preattentive vision

parallel, instantaneous *(~100--200ms),* without scrutiny, independent of the number of patterns, covering a large visual field.

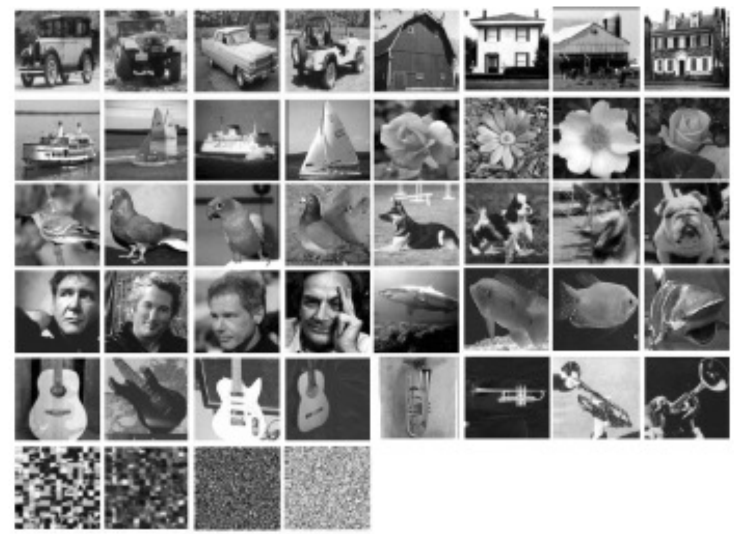
2. Attentive vision

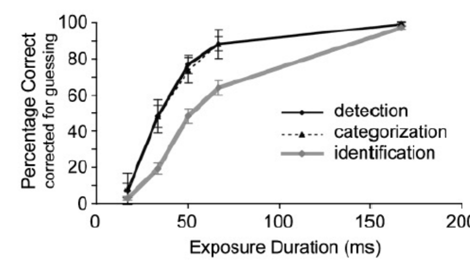
serial search by focal attention in 50ms steps limited to small aperture.

Evidence for Pre-attentive Recognition (Thorpe)

On a task of judging **animal vs no animal**, humans can make mostly correct saccades in 150 ms (Kirchner & Thorpe, 2006)

- Comparable to synaptic delay in the retina, LGN, V1, V2, V4, IT pathway.
- Doesn't rule out feed back but shows feed forward only is very powerful



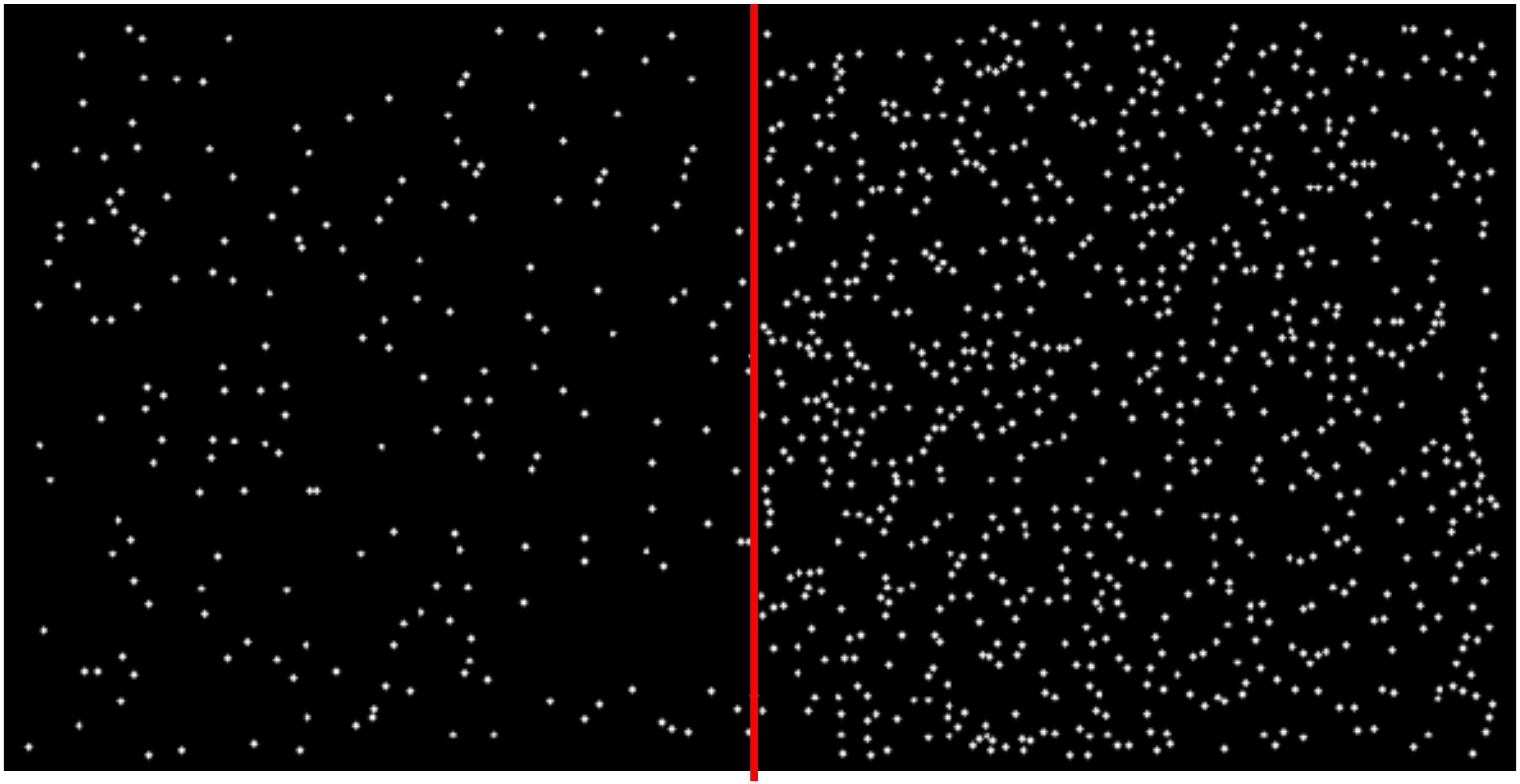


Detection and categorization are practically simultaneous (Grill-Spector & Kanwisher, 2005)

Julesz Conjecture

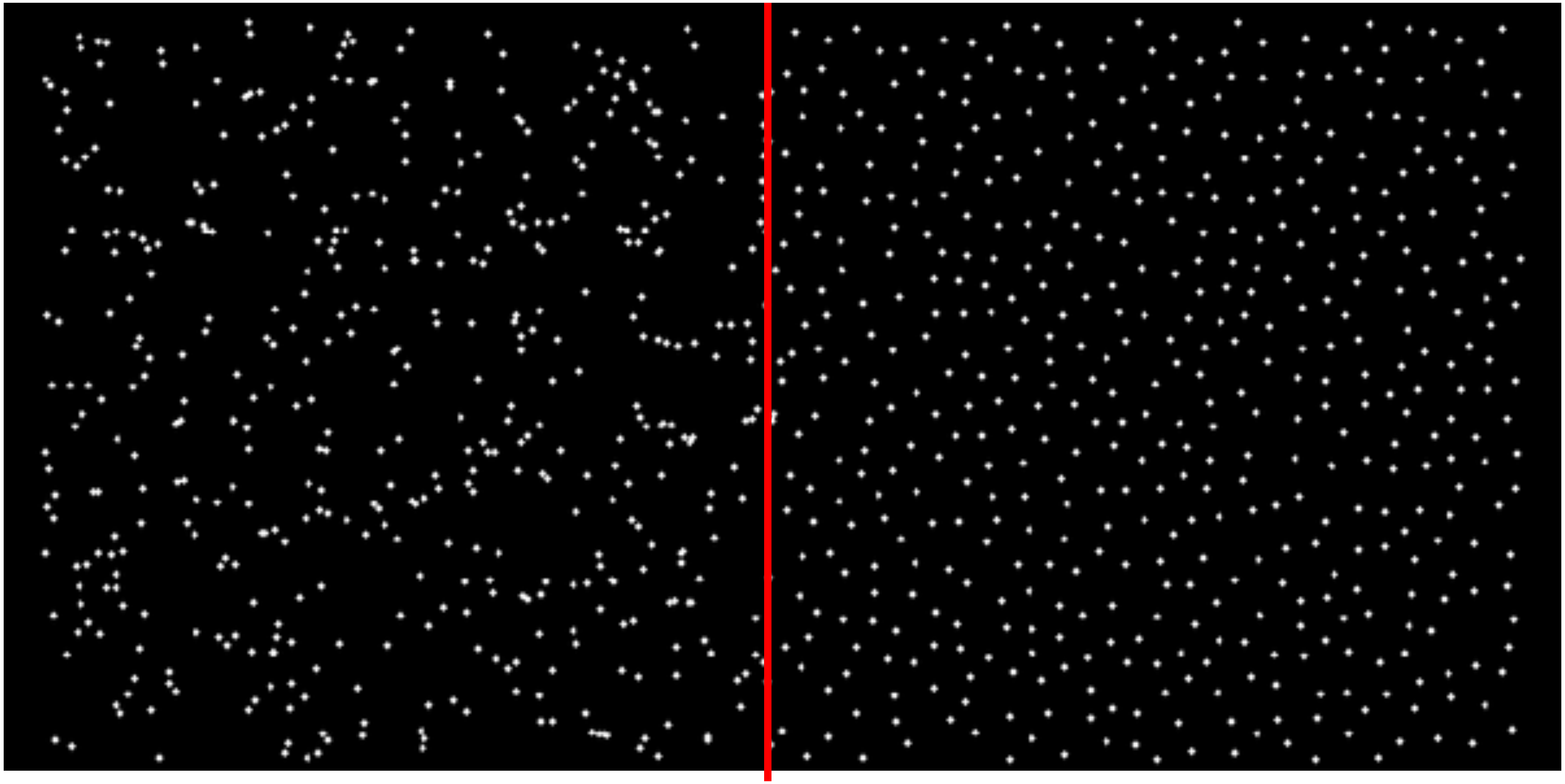
Textures cannot be spontaneously discriminated if they have the ***same first-order and second-order statistics*** *of texture features (textons) and differ only in their third-order or higher-order statistics.*

1^{st} Order Statistics



5% white   20% white

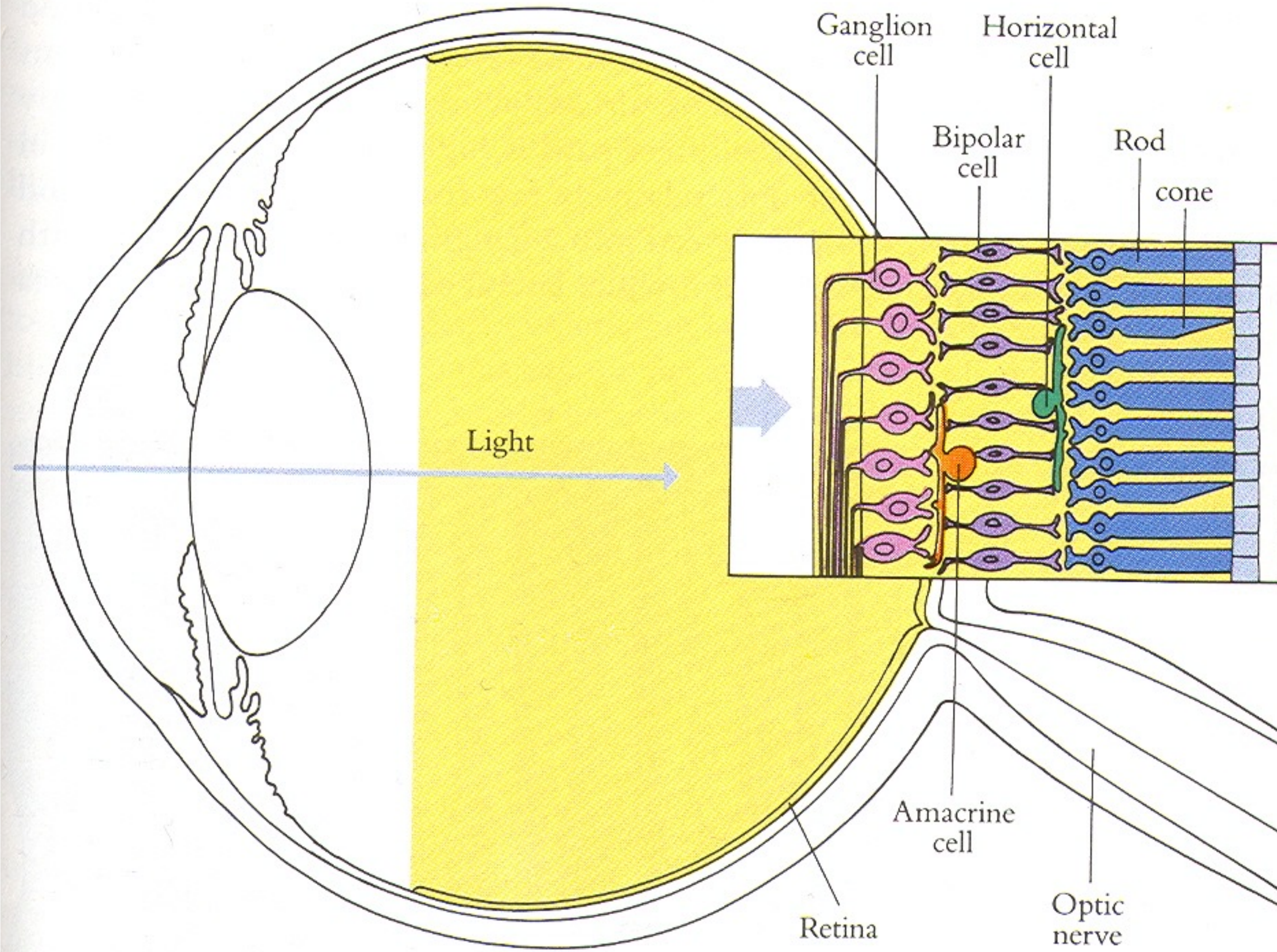
2nd Order Statistics



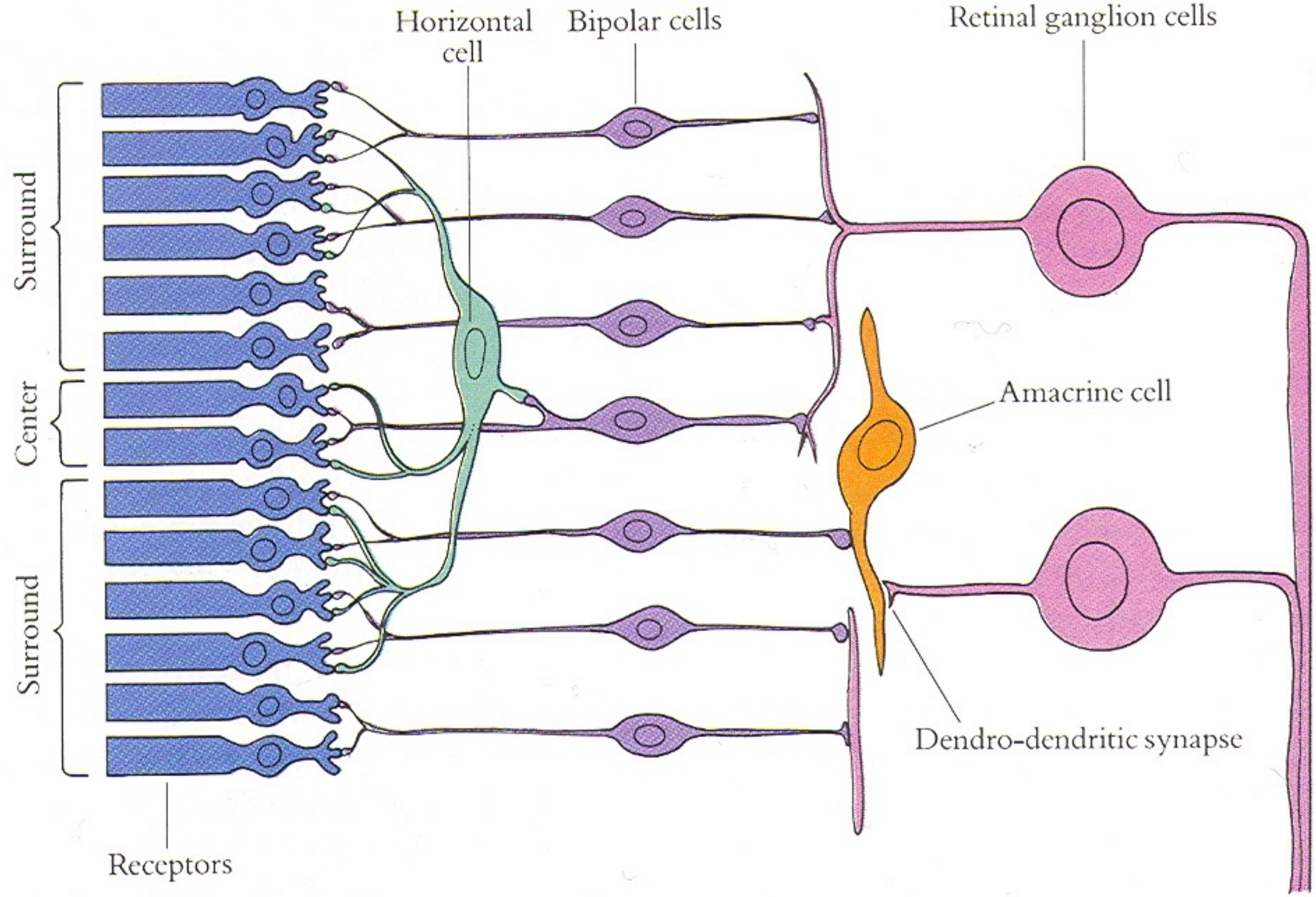
10% white

Big Question

What is the statistical unit (texton) of texture in real images?

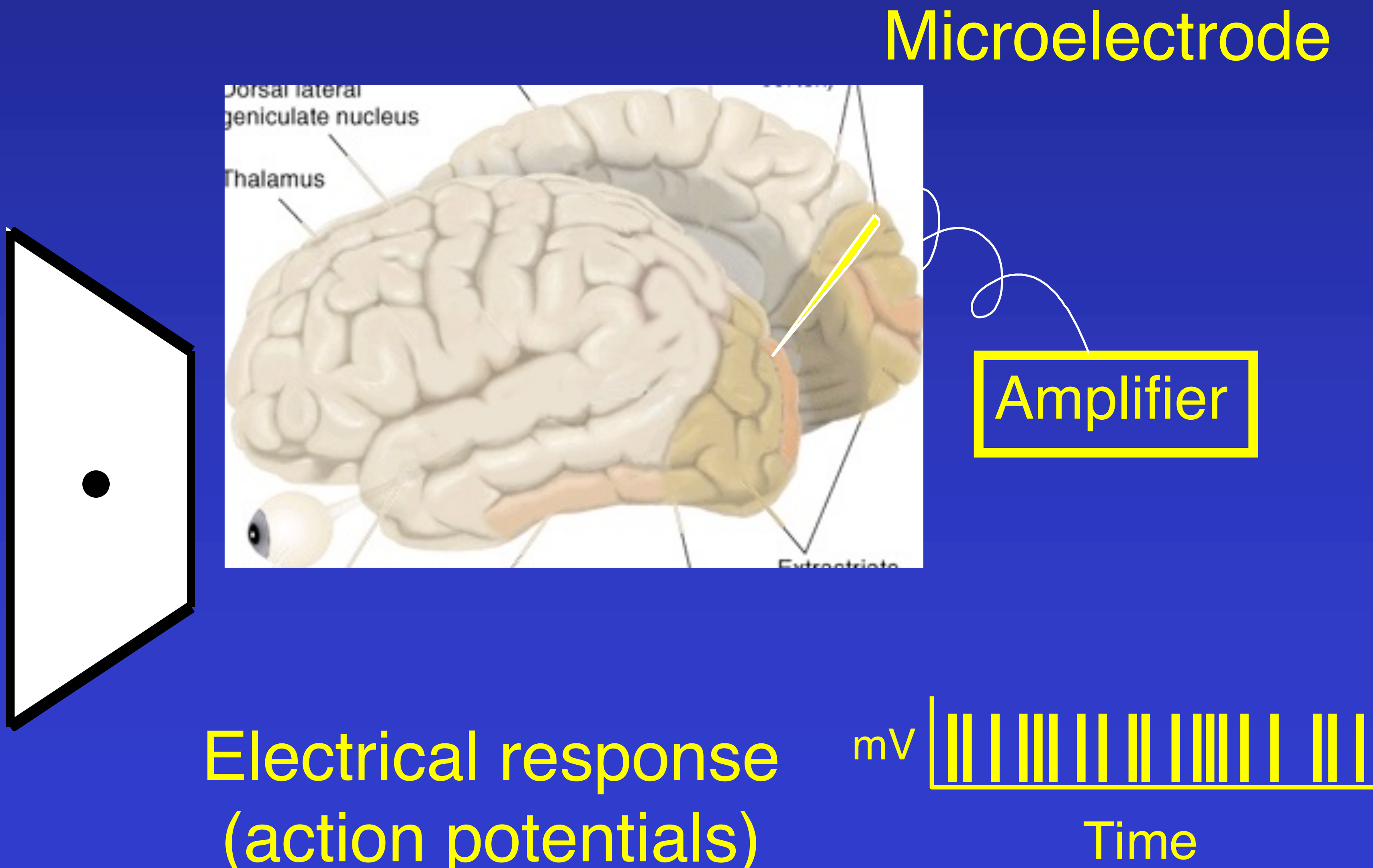


Ganglion cell
Horizontal cell
Bipolar cell
Rod
cone
Light
Amacrine cell
Retina
Optic nerve


Horizontal cell
Bipolar cells
Retinal ganglion cells
Surround
Center
Surround
Amacrine cell
Dendro-dendritic synapse
Receptors

Single Cell Recording

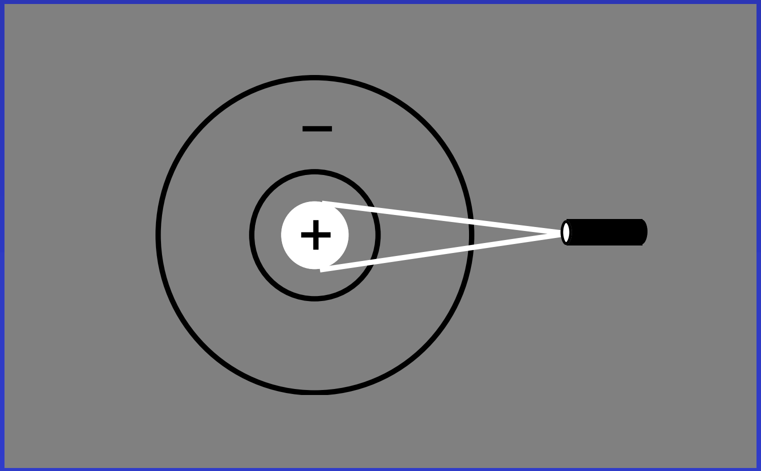
Single Cell Recording

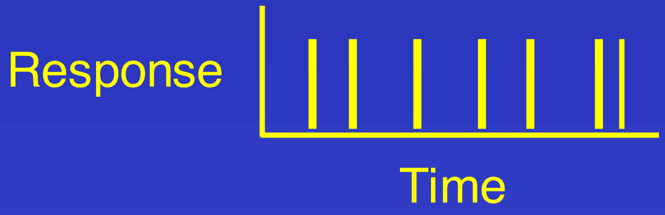


Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround



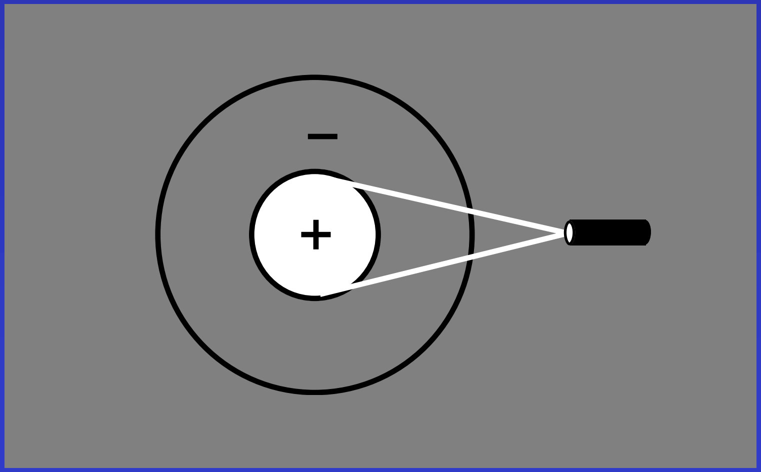


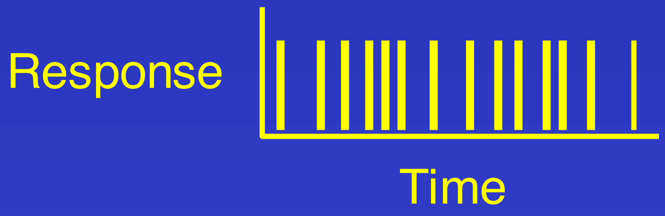
Stimulus condition

Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:
On-center Off-surround



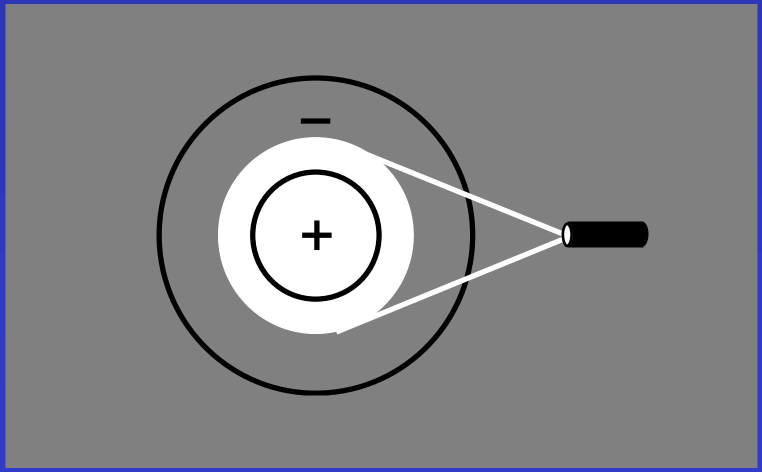


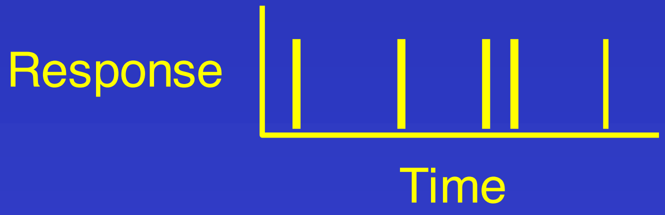
Stimulus condition

Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:
On-center Off-surround



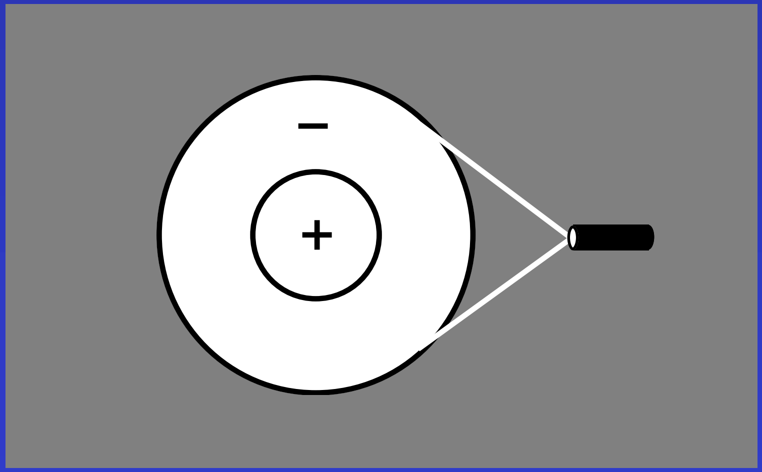


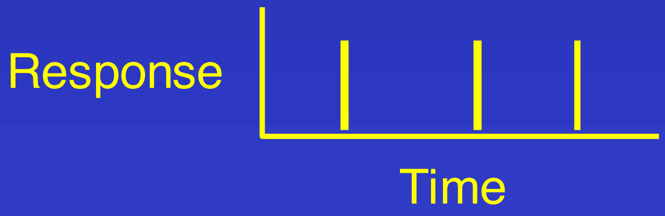
Stimulus condition

Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:
On-center Off-surround





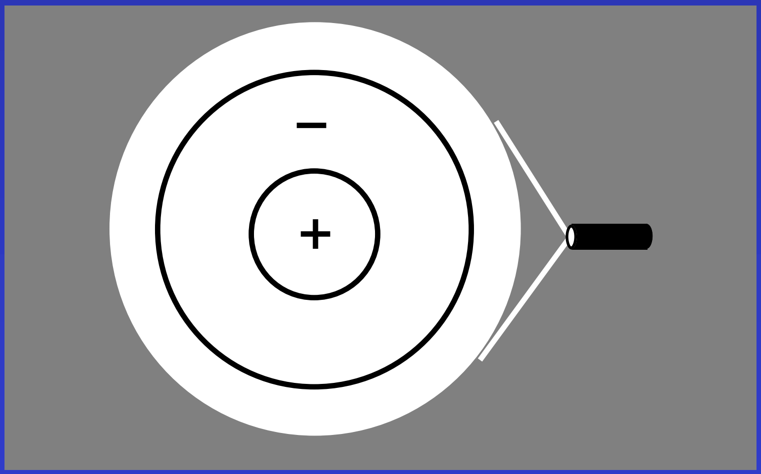
Stimulus condition

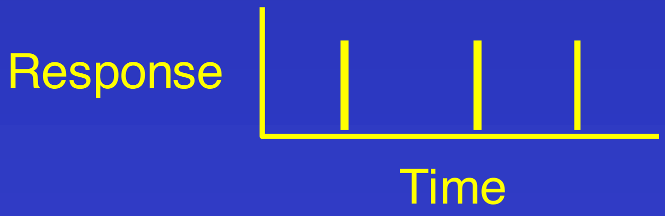
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround





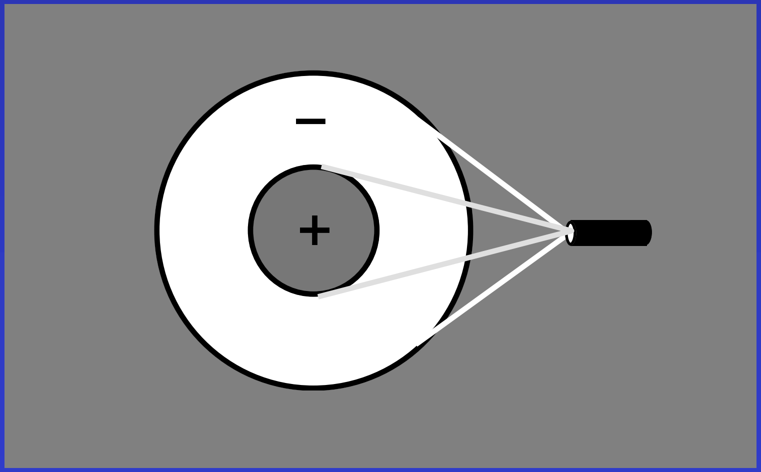
Stimulus condition

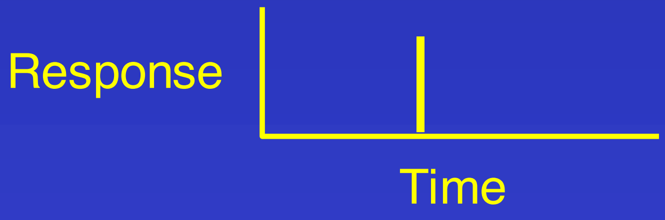
Electrical response

Retinal Receptive Fields

Receptive field structure in ganglion cells:

On-center Off-surround



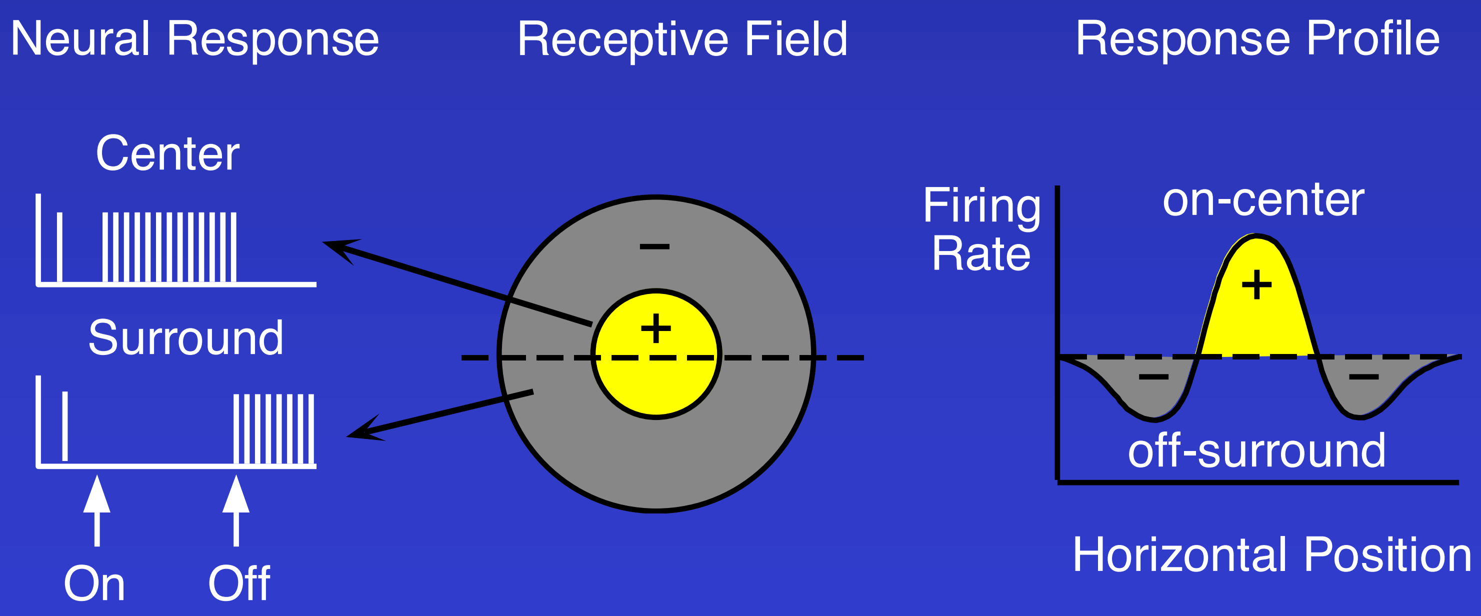


Stimulus condition

Electrical response

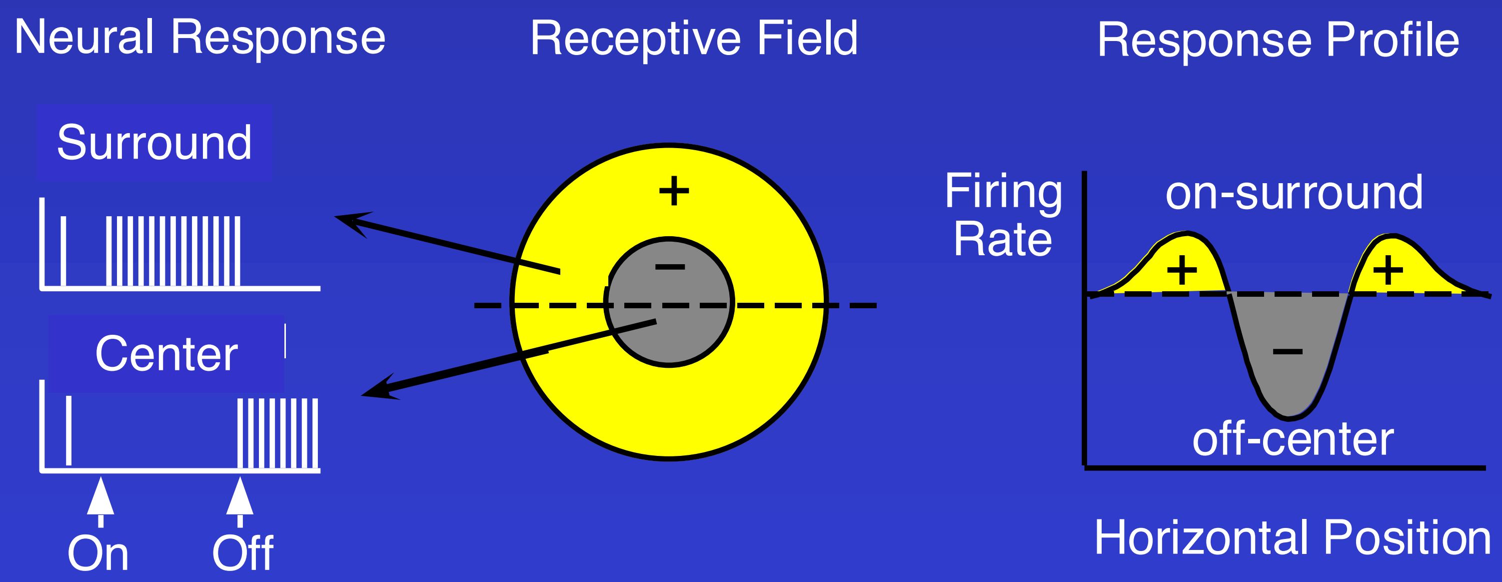
Retinal Receptive Fields

RF of On-center Off-surround cells

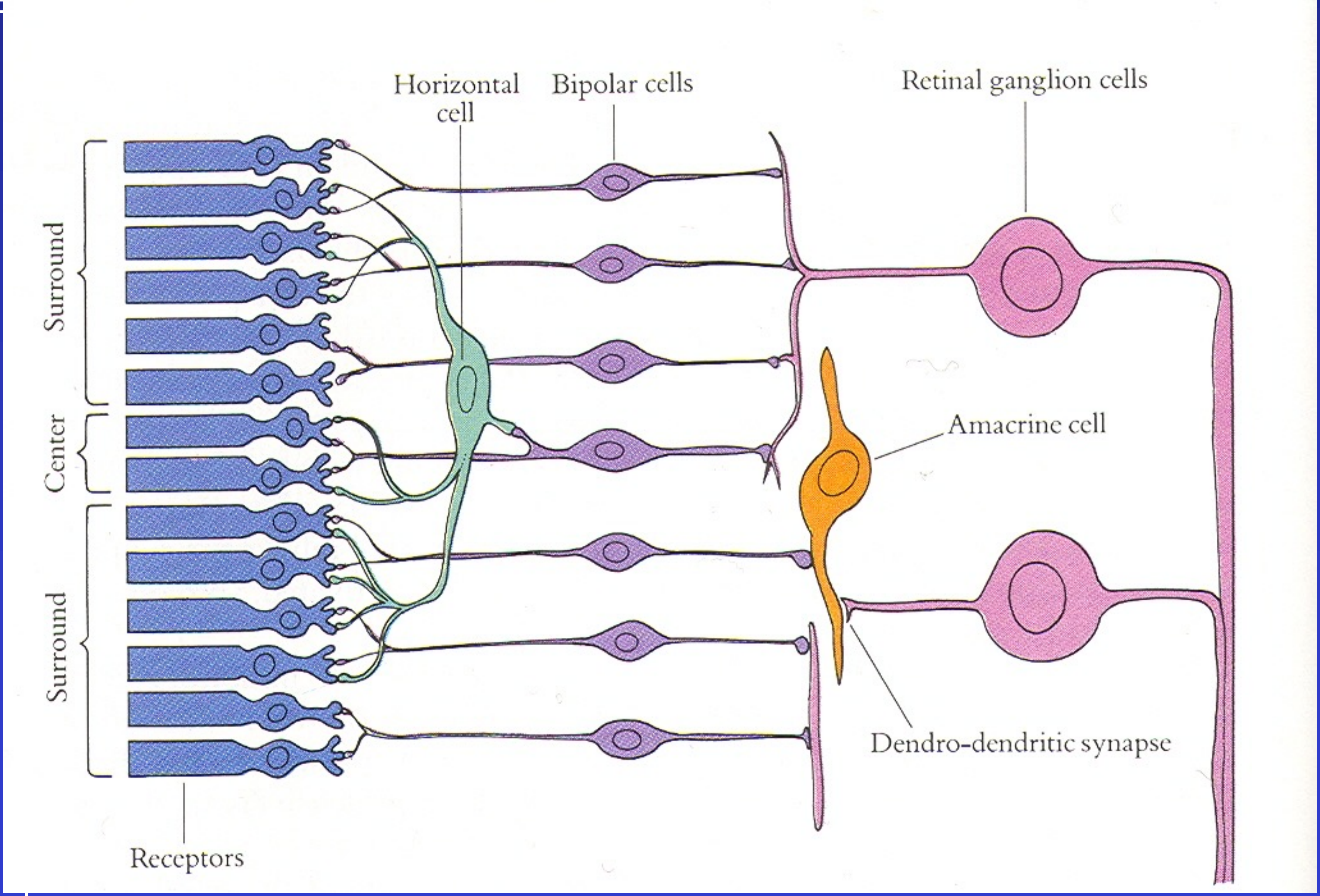


Retinal Receptive Fields

RF of Off-center On-surround cells

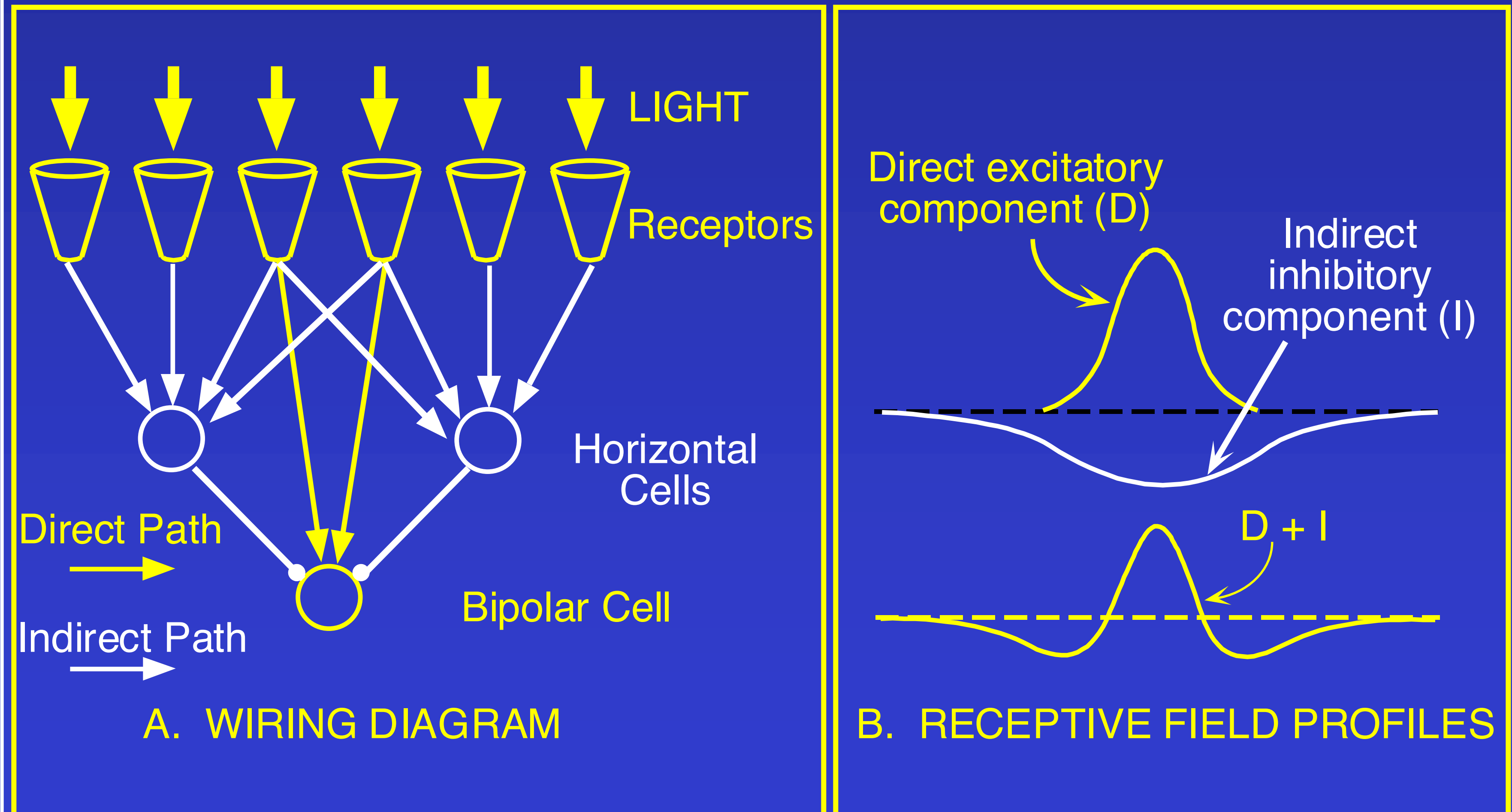


Retinal Receptive Fields

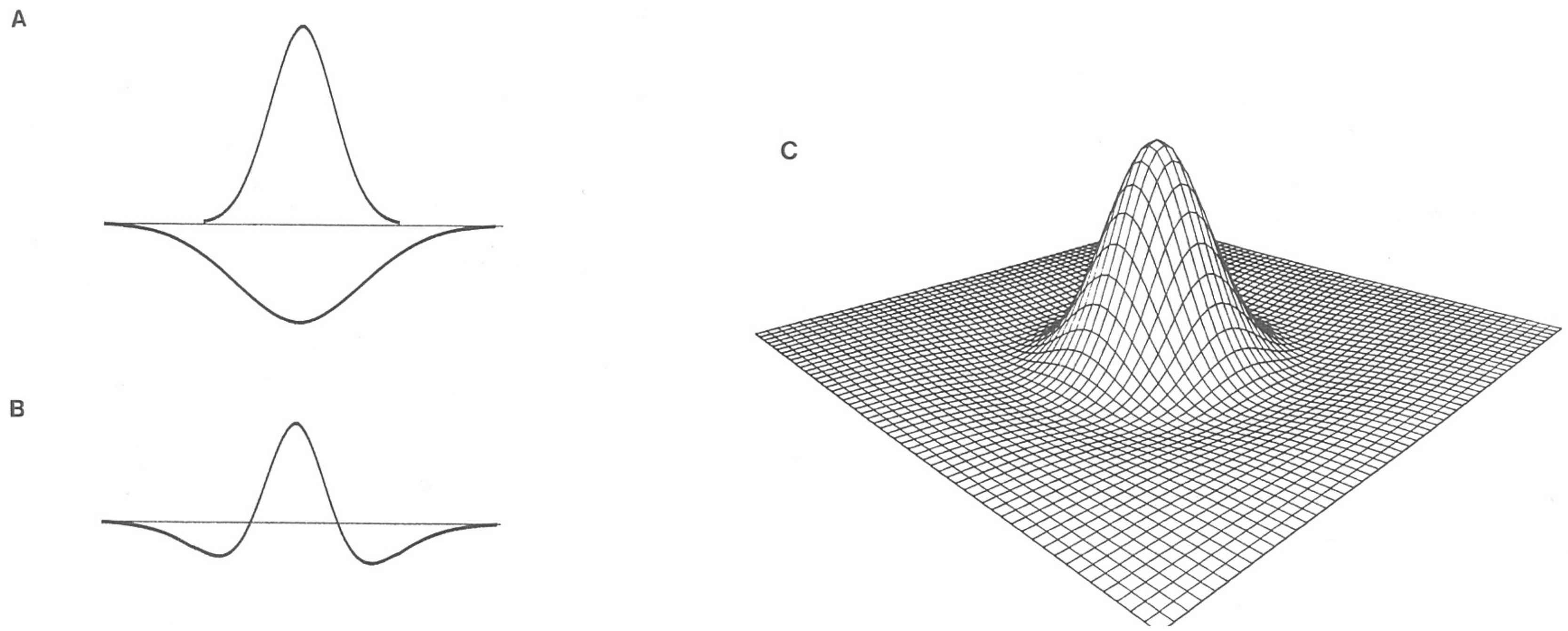


Retinal Receptive Fields

Receptive field structure in bipolar cells



The receptive field of a retinal ganglion cell can be modeled as a "Difference of Gaussians"



$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}$$

Receptive Fields

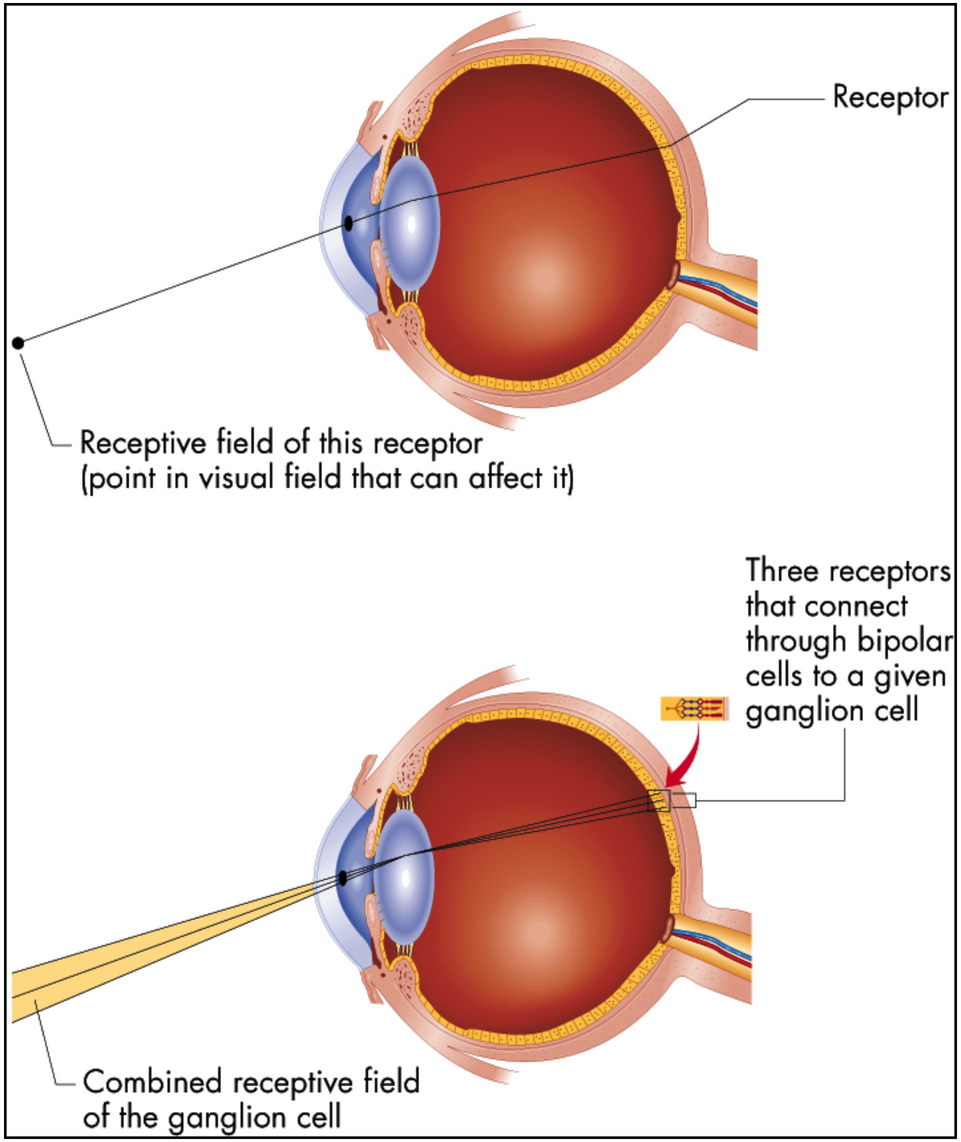
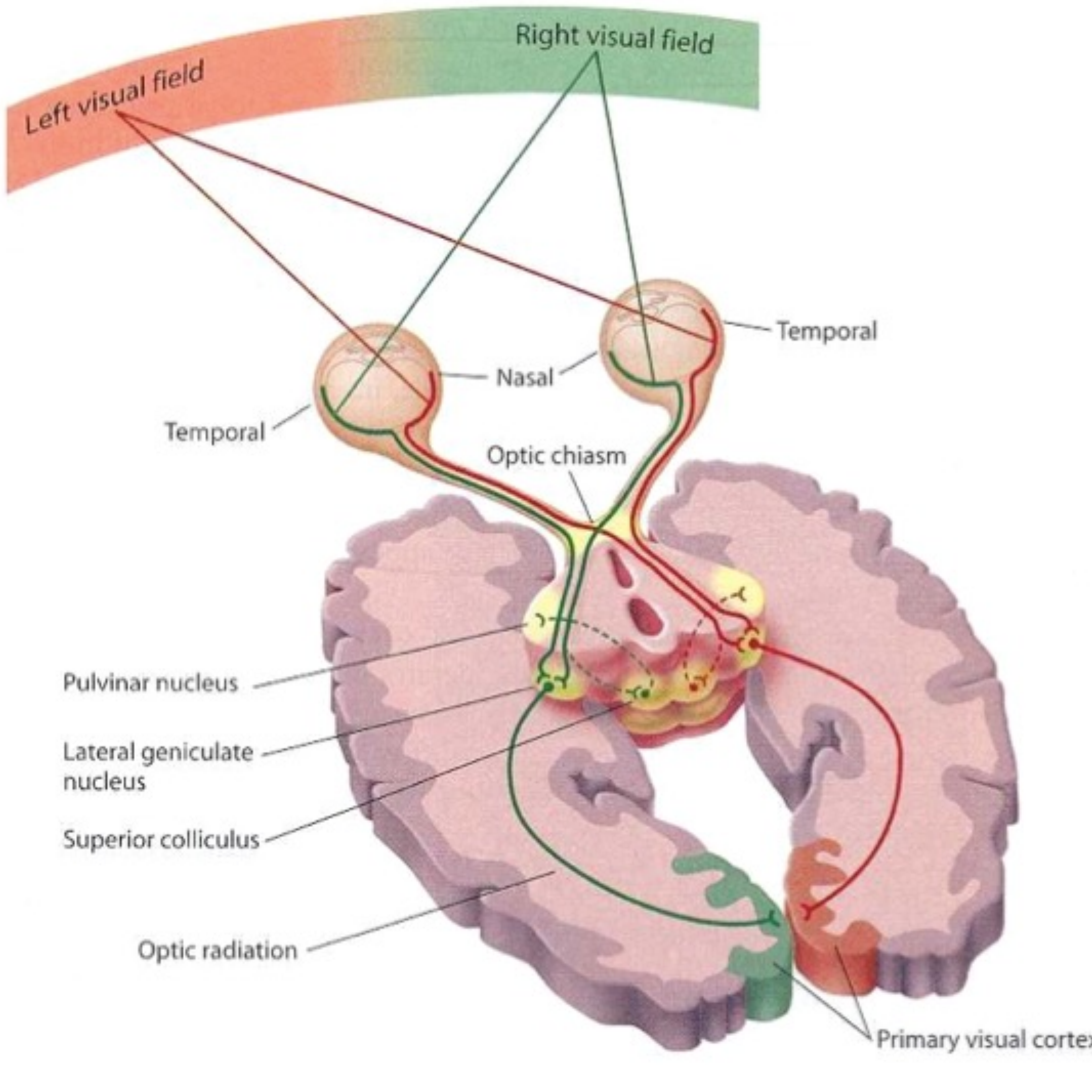


Figure 6.16 Receptive fields
The receptive field of a receptor is simply the area of the visual field from which light strikes that receptor. For any other cell in the visual system, the receptive field is determined by which receptors connect to the cell in question.

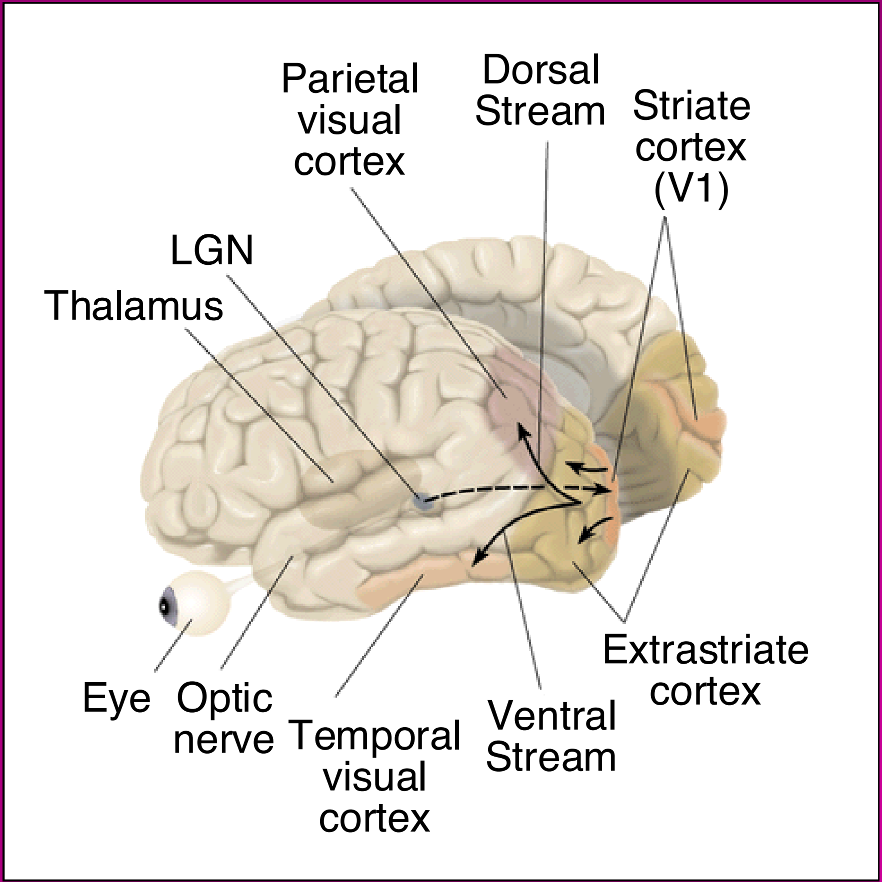
Anatomy of Pathway to Visual Cortex



Visual Cortex

Cortical Area V1

aka:
Primary visual cortex
Striate cortex
Brodman’s area 17



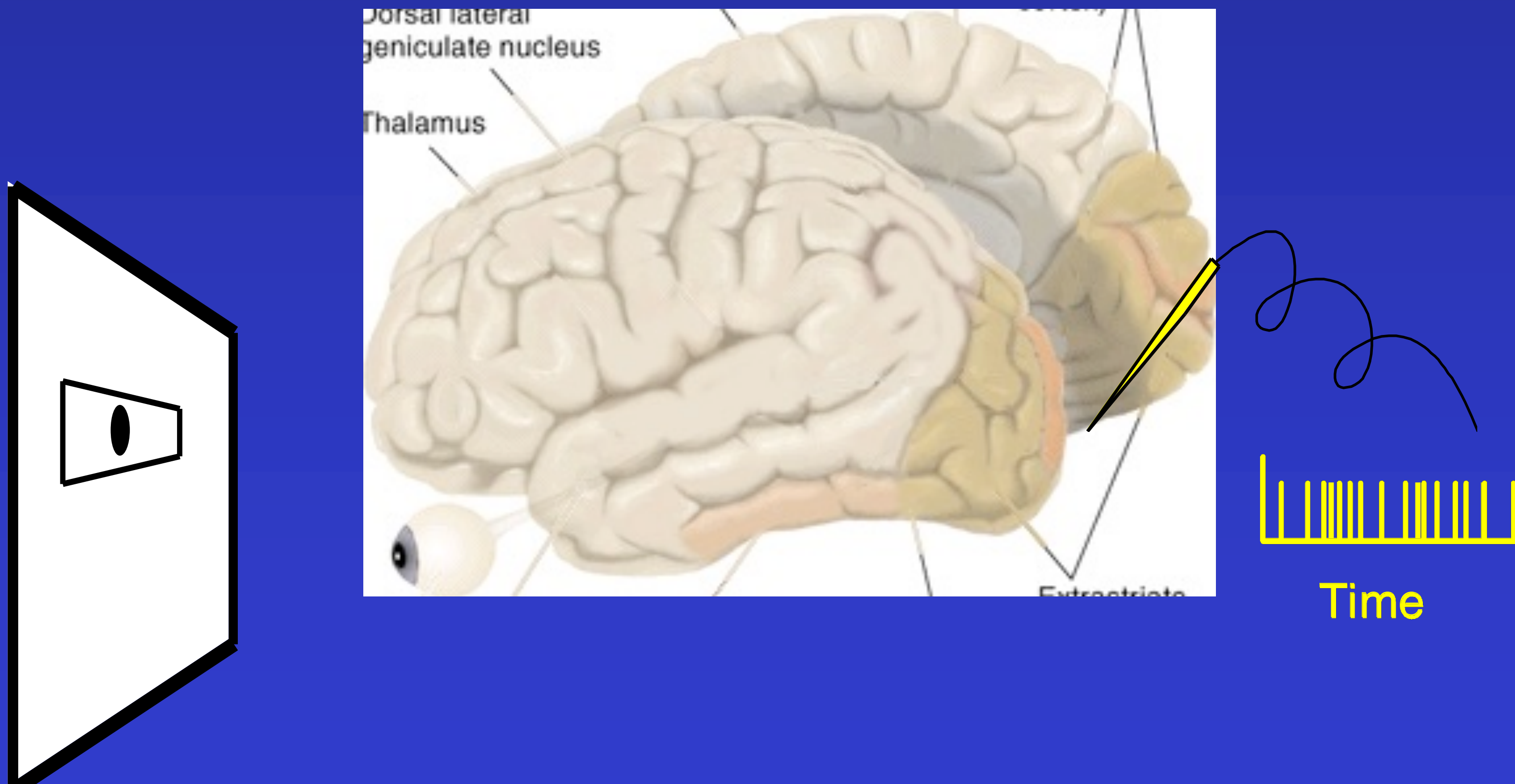
Cortical Receptive Fields

Single-cell recording from visual cortex

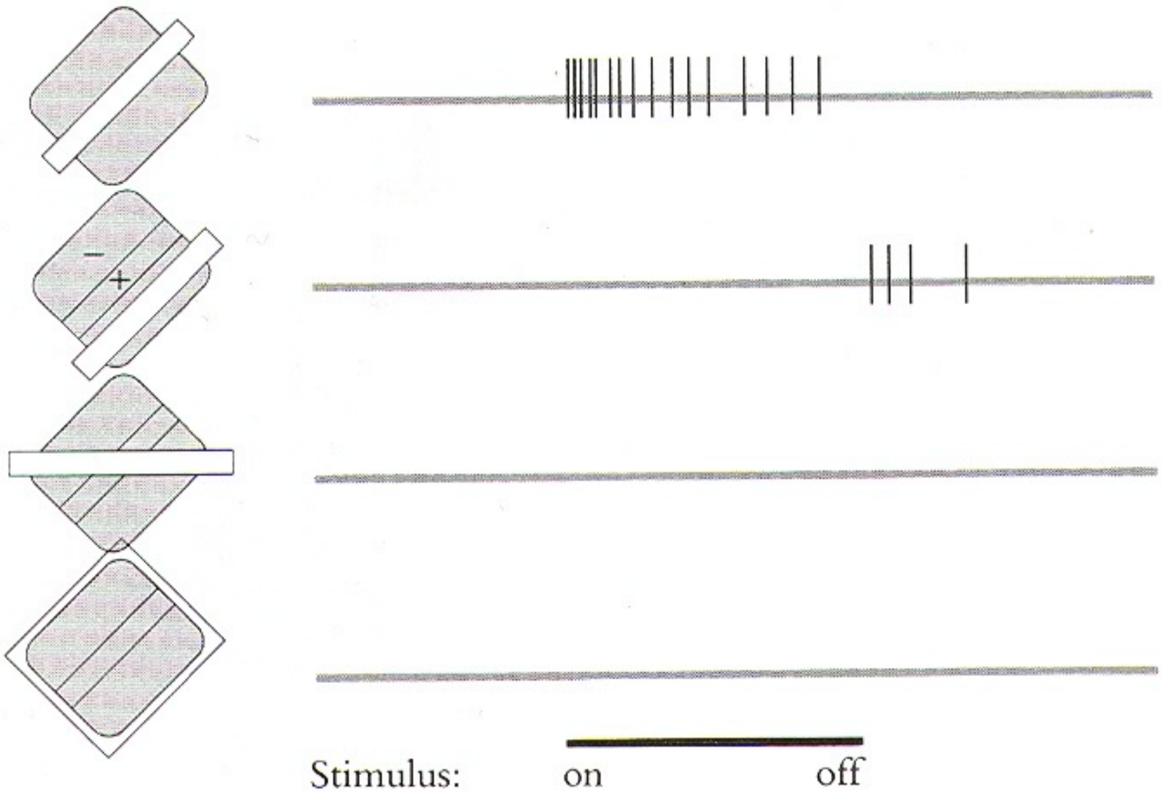
David Hubel & Thorston Wiesel

Cortical Receptive Fields

Single-cell recording from visual cortex



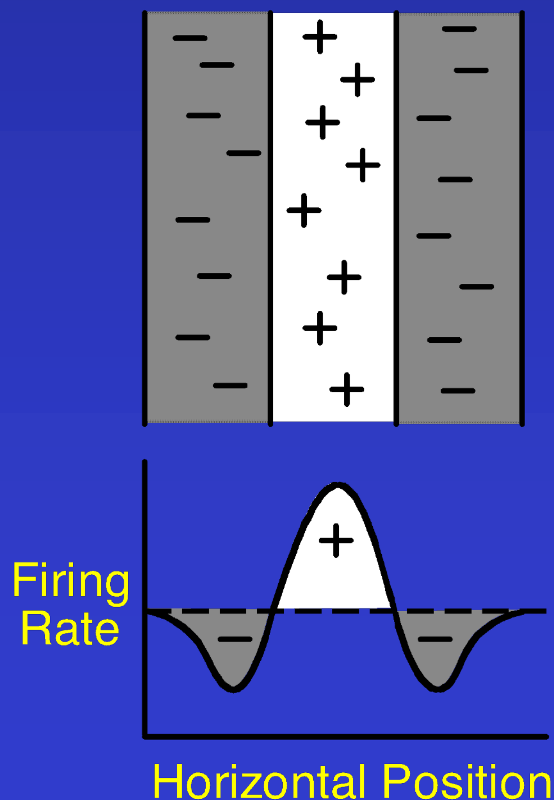
https://www.youtube.com/watch?v=IOHayh06LJ4


Stimulus:
on
off

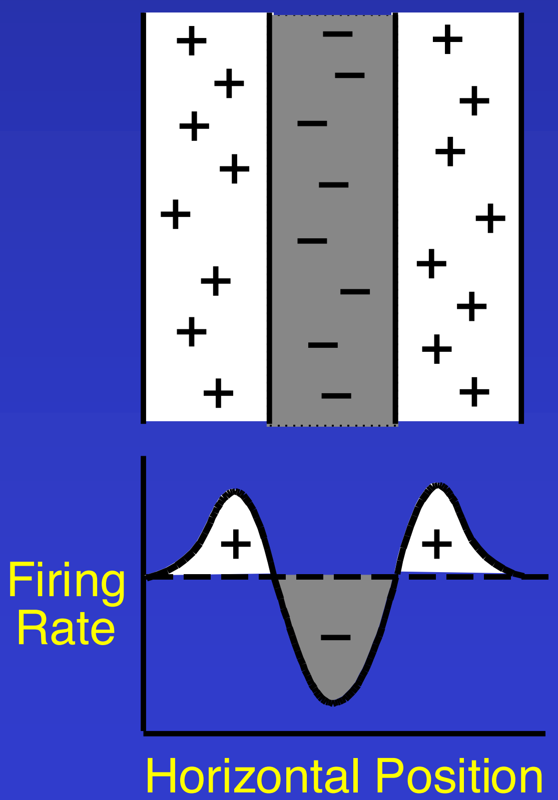
Cortical Receptive Fields

Simple Cells: “Line Detectors”

A. Light Line Detector



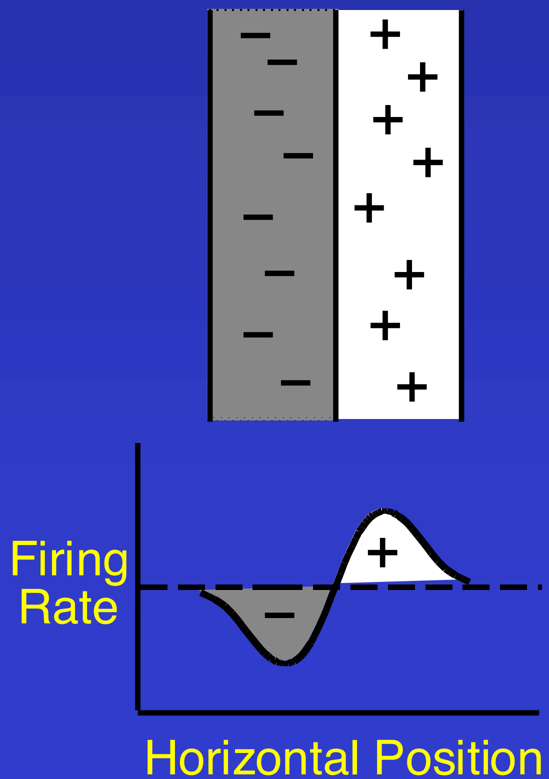
B. Dark Line Detector



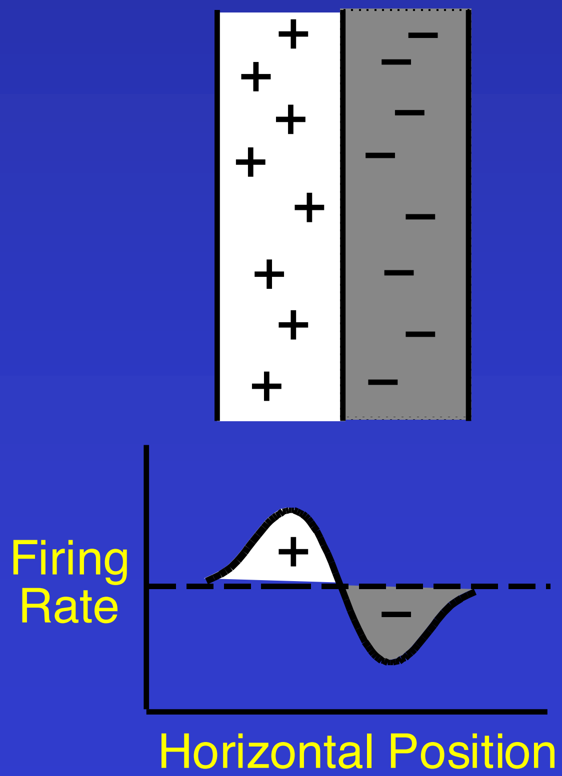
Cortical Receptive Fields

Simple Cells: "Edge Detectors"

C. Dark-to-light Edge Detector

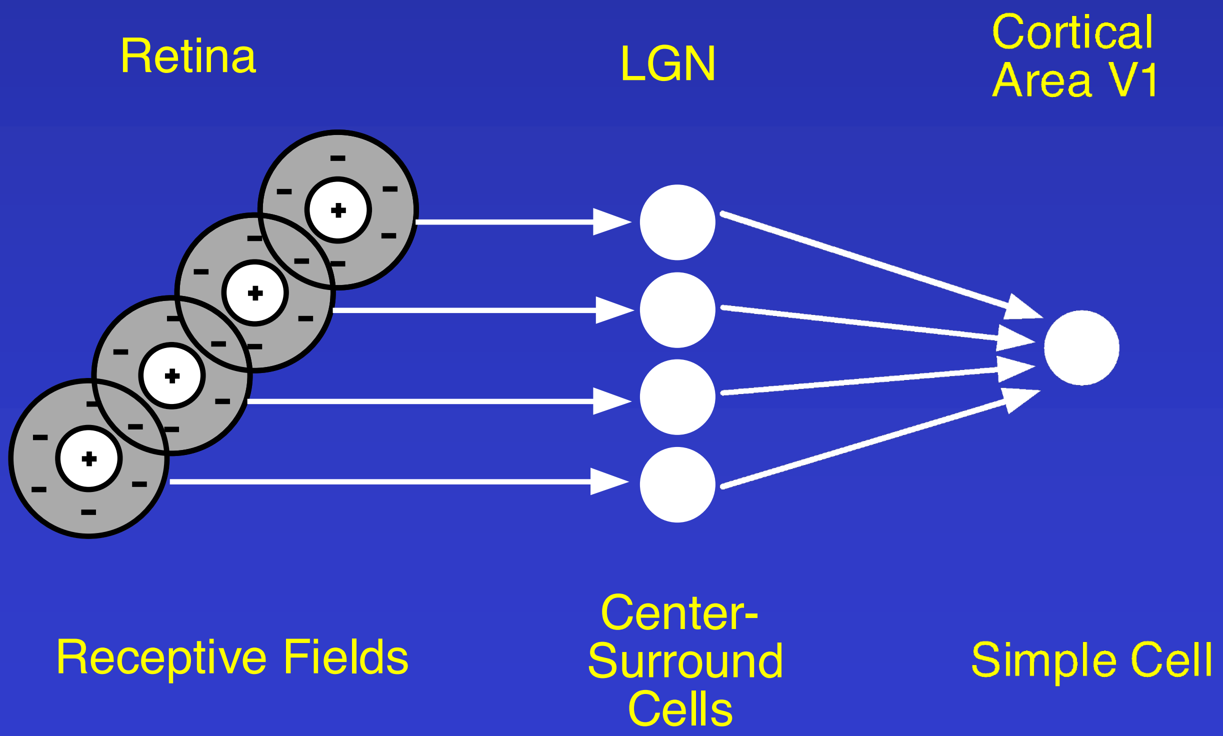


D. Light-to-dark Edge Detector

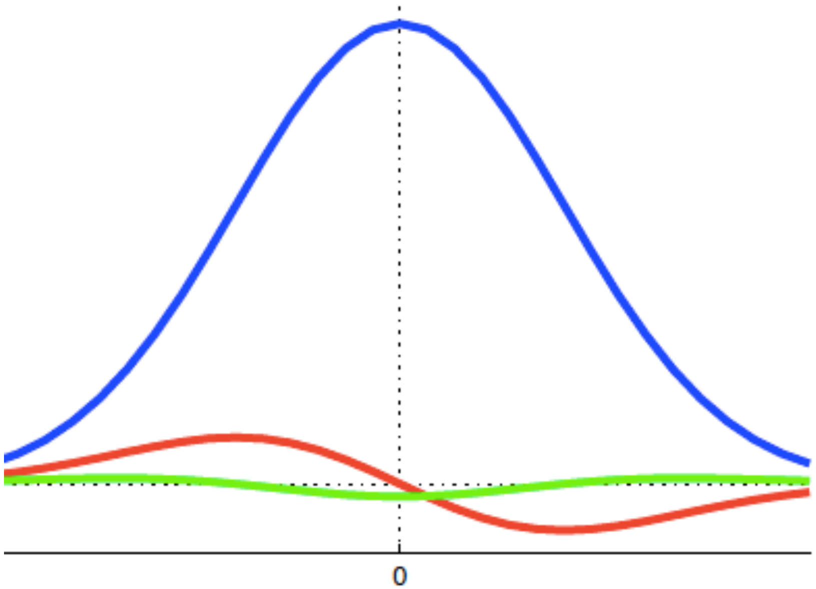


Cortical Receptive Fields

Constructing a line detector



The 1D Gaussian and its derivatives



$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$G'_\sigma(x) = \frac{d}{dx} G_\sigma(x) = -\frac{1}{\sigma}\left(\frac{x}{\sigma}\right) G_\sigma(x)$$

$$G''_\sigma(x) = \frac{d^2}{dx^2} G_\sigma(x) = \frac{1}{\sigma^2}\left(\frac{x^2}{\sigma^2} - 1\right) G_\sigma(x)$$

$G'_\sigma(x)$'s maxima/minima occur at $G''_\sigma(x)$'s zeros. And, we can see that $G'_\sigma(x)$ is an odd symmetric function and $G''_\sigma(x)$ is an even symmetric function.

Oriented Gaussian Derivatives in 2D

$$f_1(x,y) = G'_{\sigma_1}(x)G_{\sigma_2}(y) \tag{10.4}$$
$$f_2(x,y) = G''_{\sigma_1}(x)G_{\sigma_2}(y) \tag{10.5}$$

We also consider rotated versions of these Gaussian derivative functions.

$$Rot_\theta f_1 = G'_{\sigma_1}(u)G_{\sigma_2}(v) \tag{10.6}$$
$$Rot_\theta f_2 = G''_{\sigma_1}(u)G_{\sigma_2}(v) \tag{10.7}$$

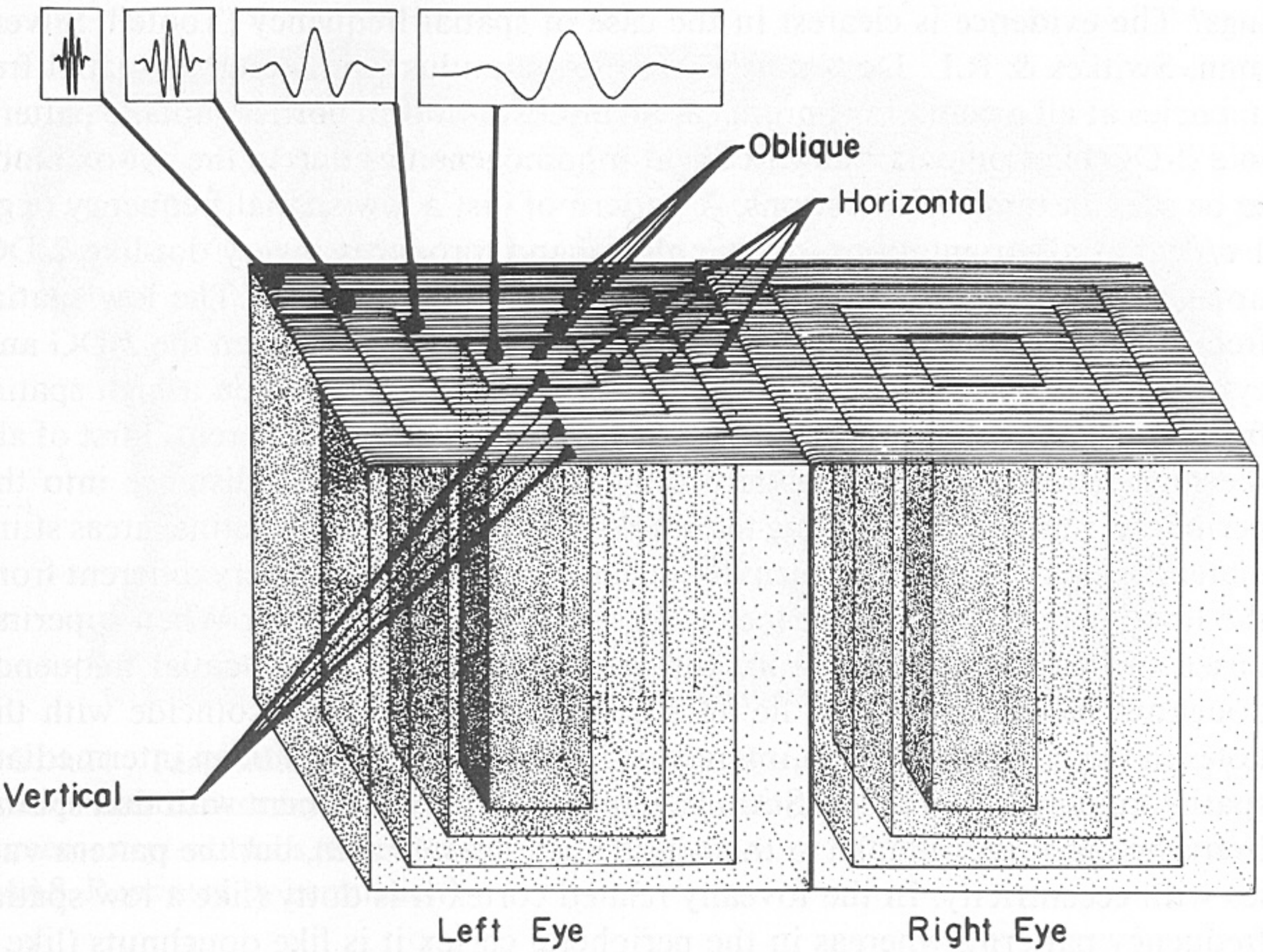
where we set

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

These are useful when we convolve with 2D images, e.g. to detect edges at different orientations.

Oriented Gaussian First and Second Derivatives

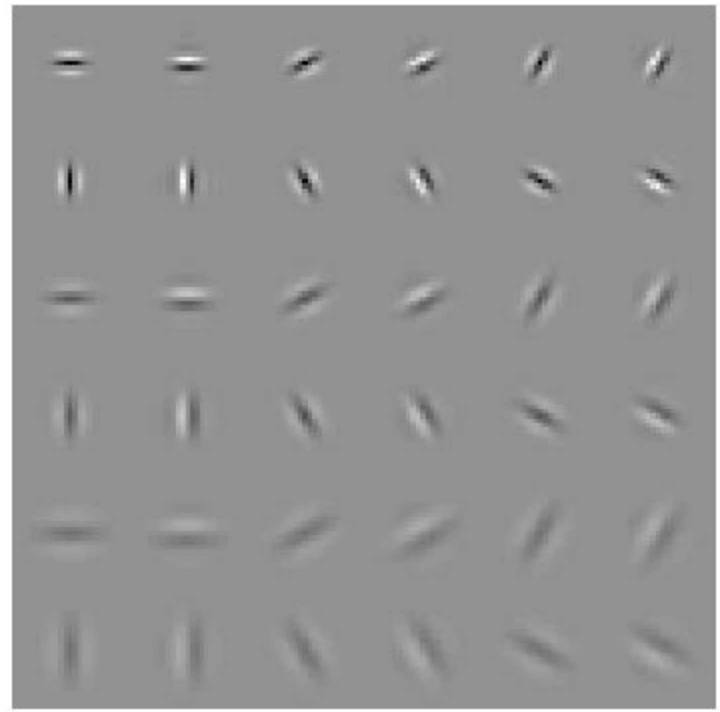
Hypercolumns in visual cortex



Model of Striate Module in Monkeys

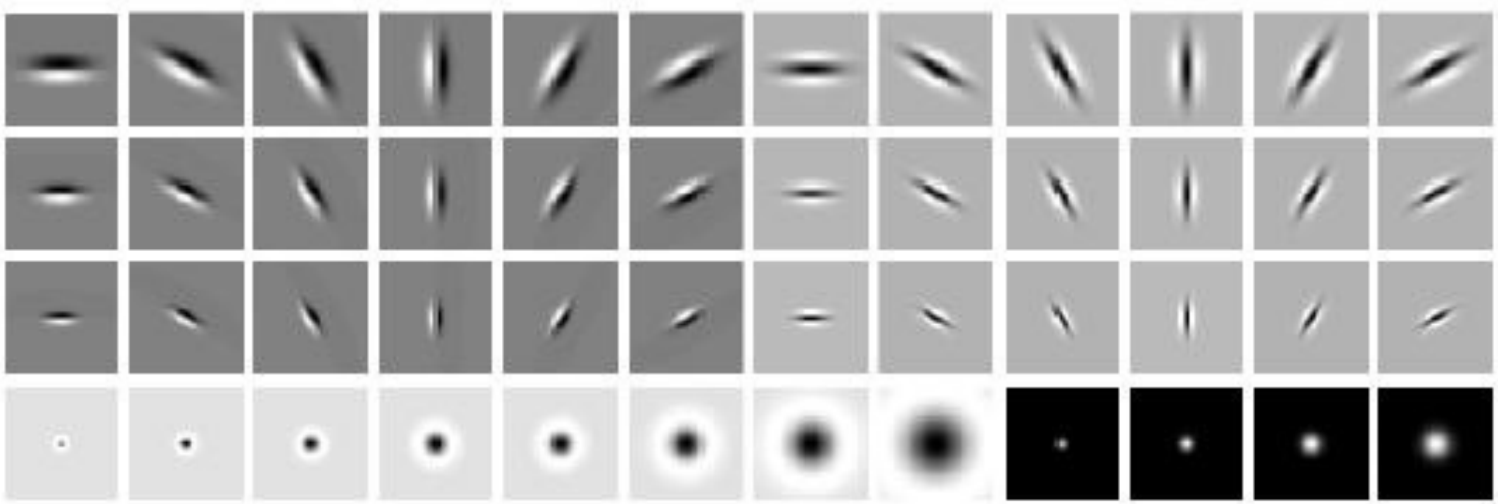
Modeling hypercolumns

- Elongated directional Gaussian derivatives
- Gabor filters could be used instead
- Multiple orientations, scales



Overcomplete representation: filter banks

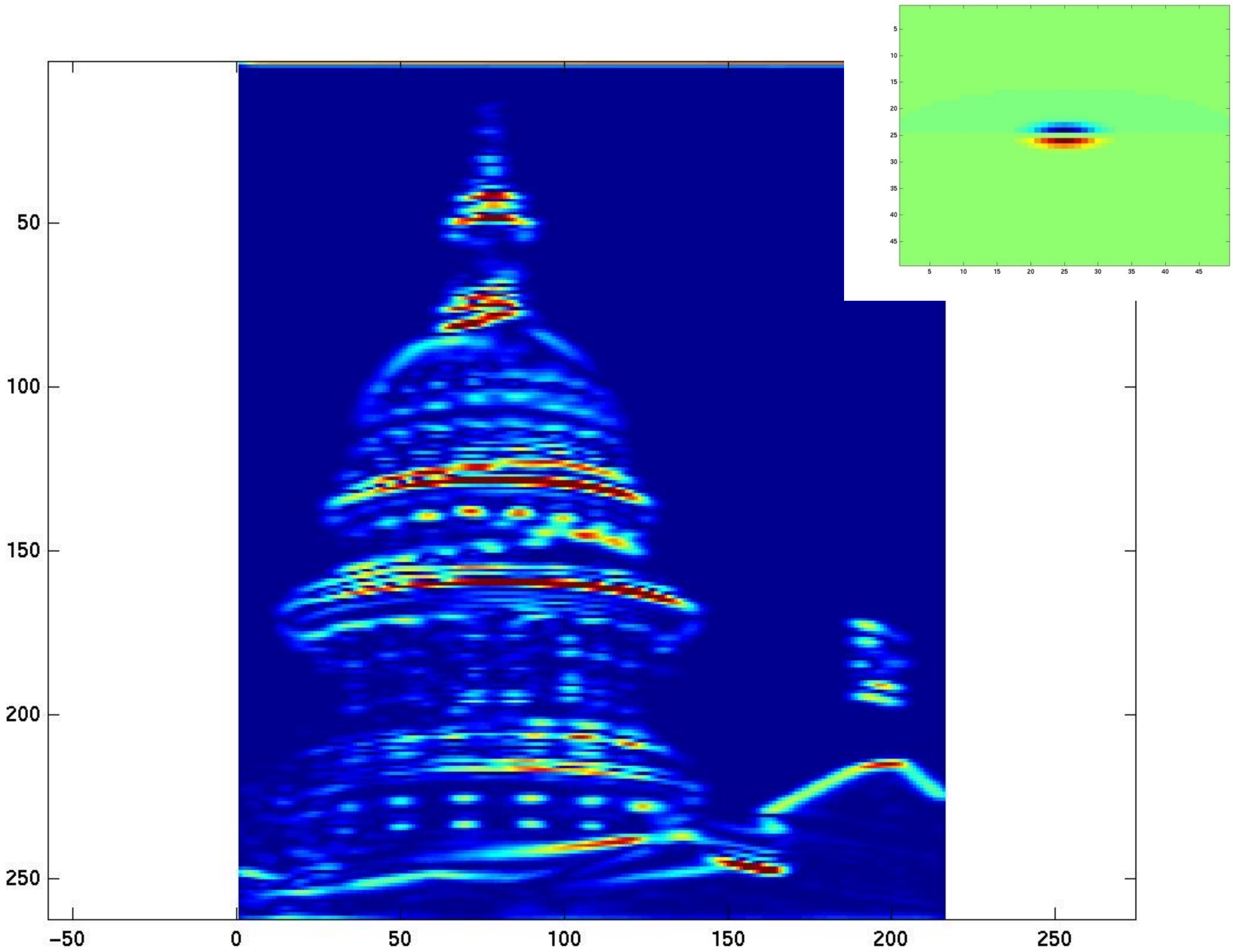
LM Filter Bank



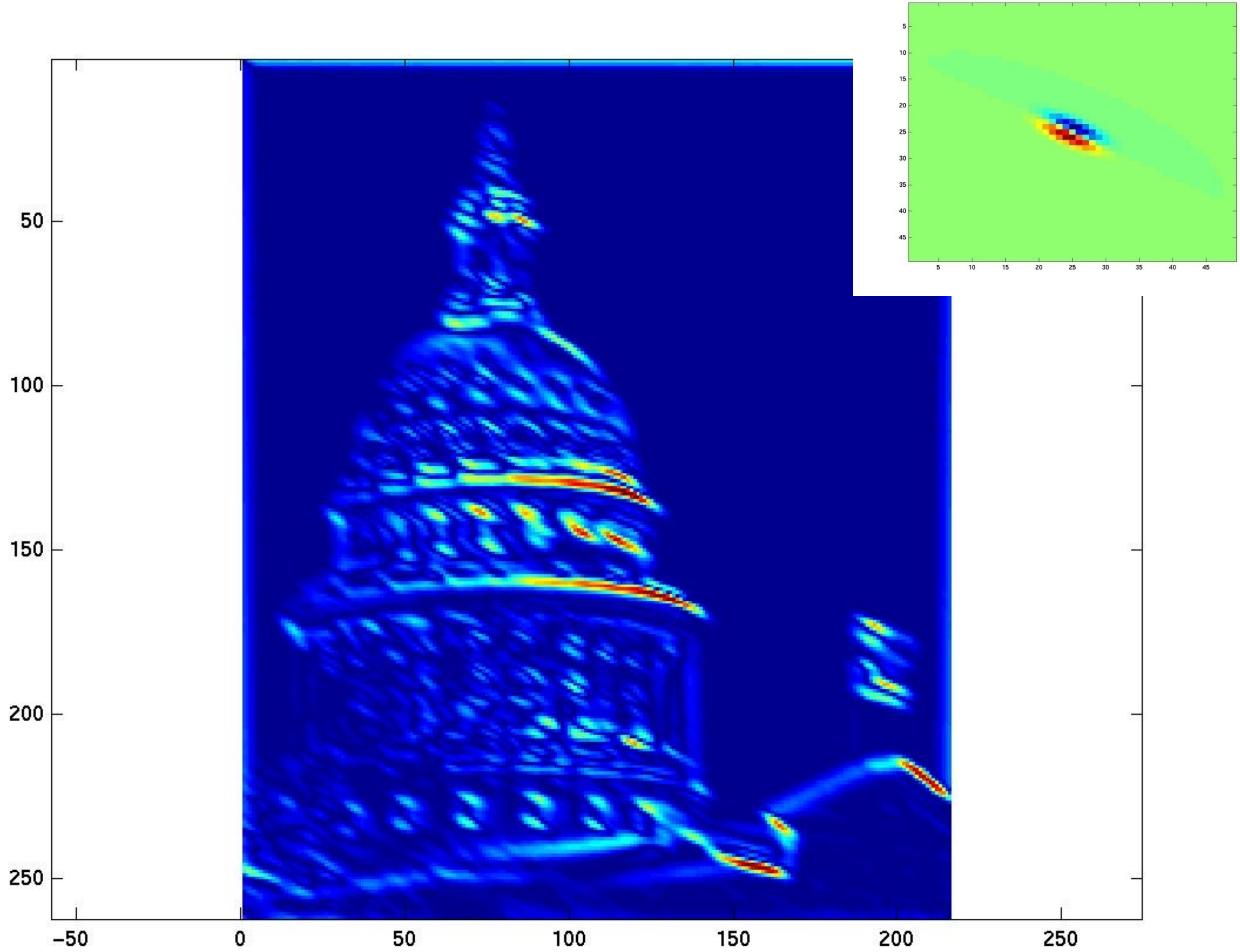
Code for filter banks: *www.robots.ox.ac.uk/~vgg/research/texclass/filters.html*

Image from http://www.texasexplorer.com/austincap2.jpg

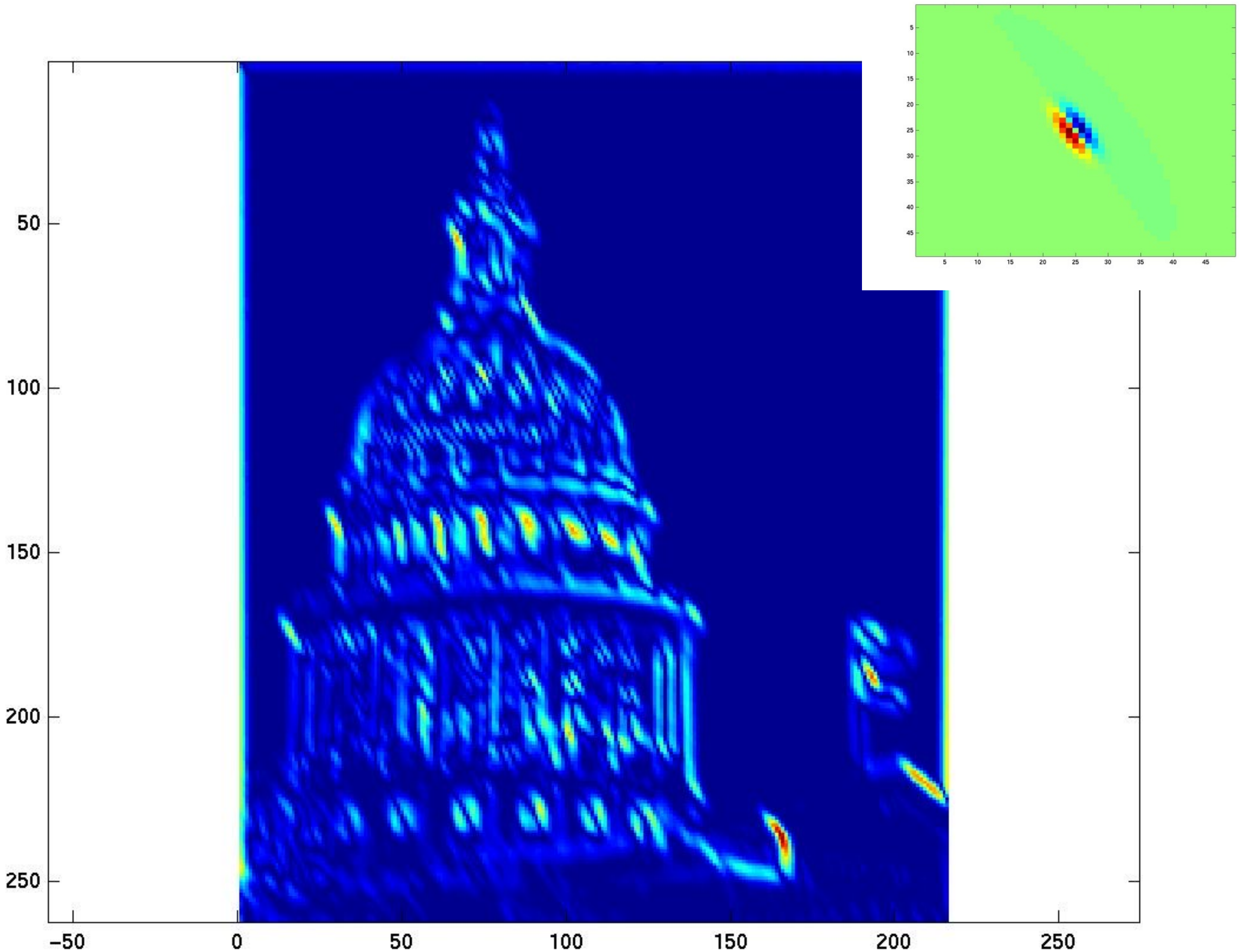
Kristen Grauman



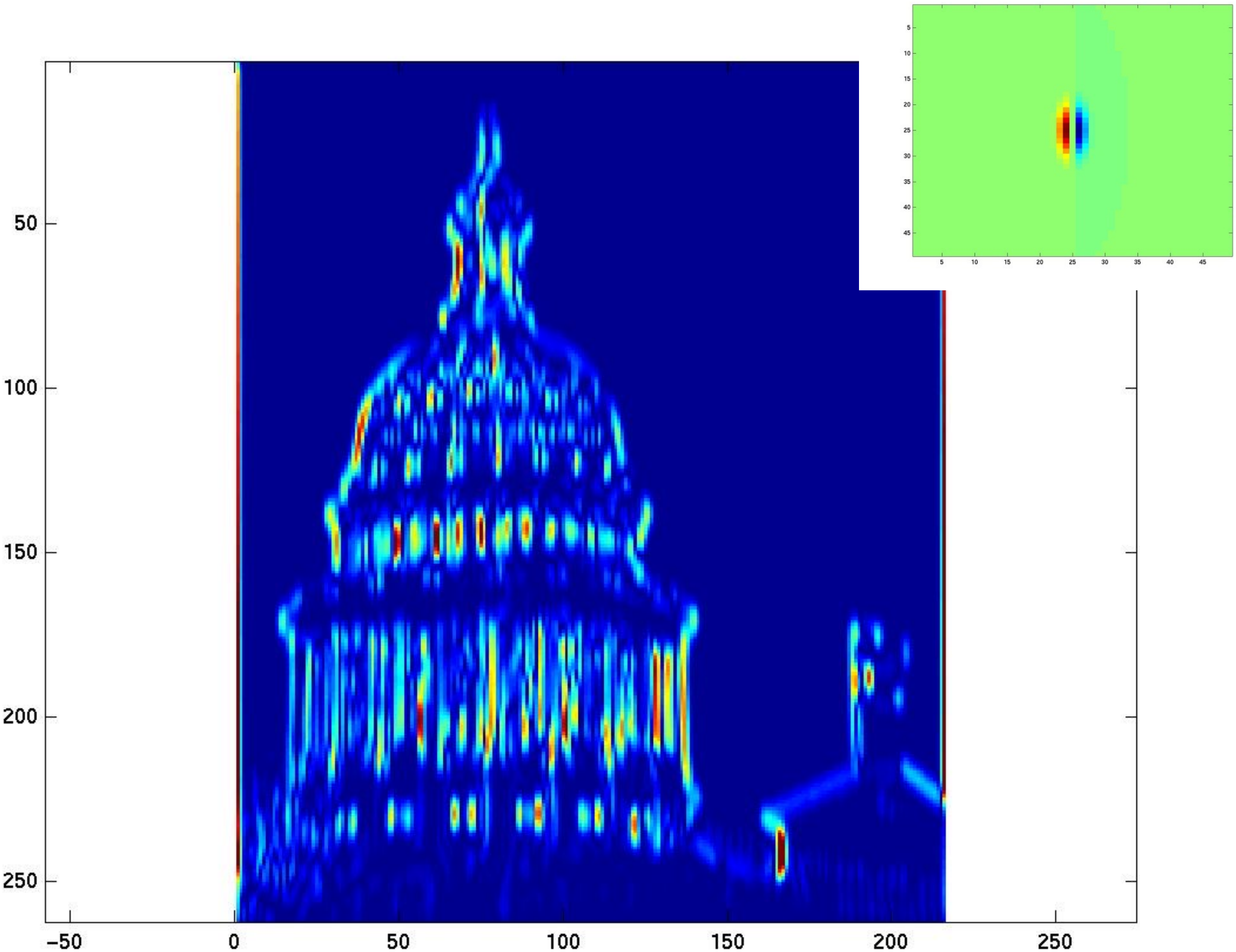
Kristen Grauman



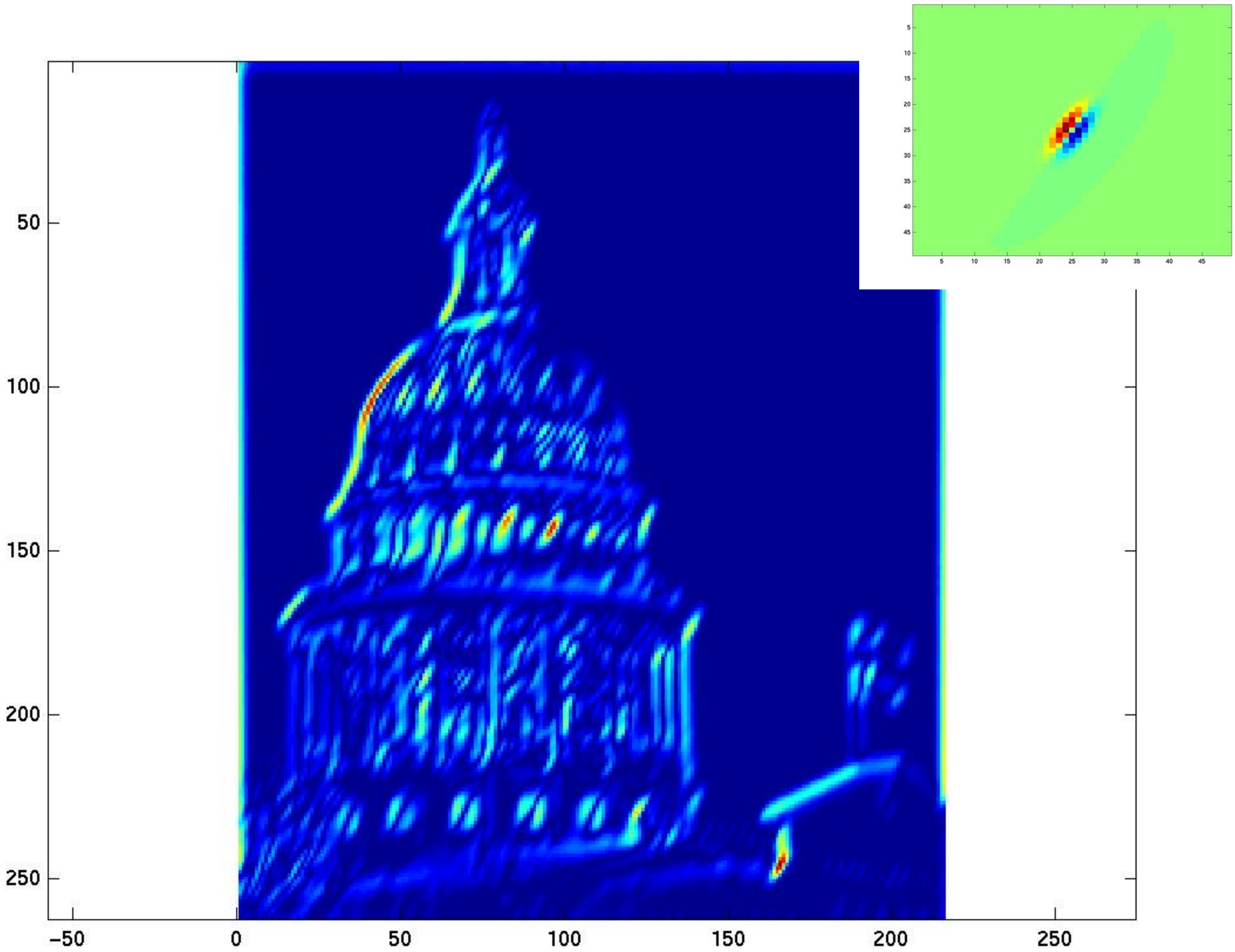
Kristen Grauman



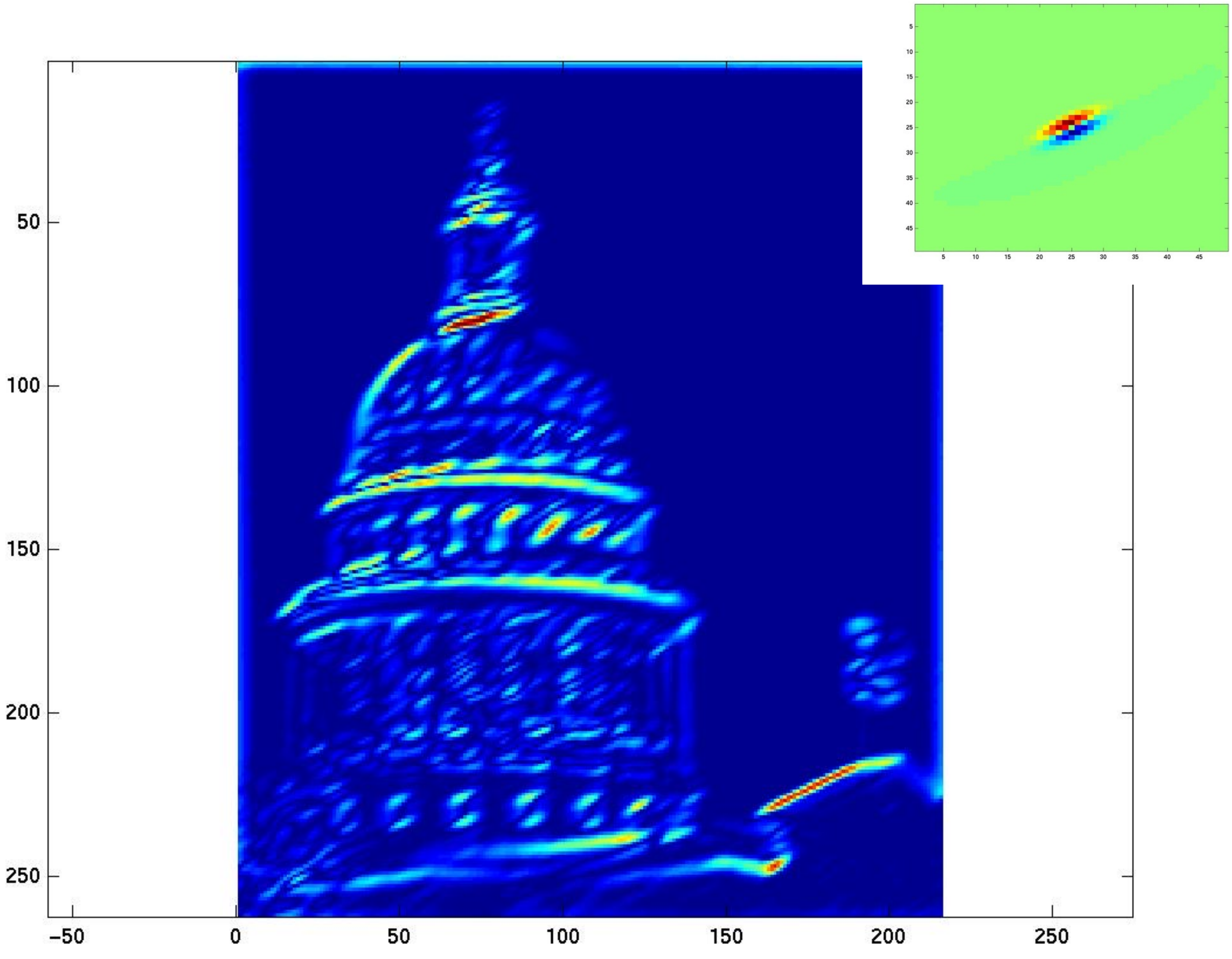
Kristen Grauman



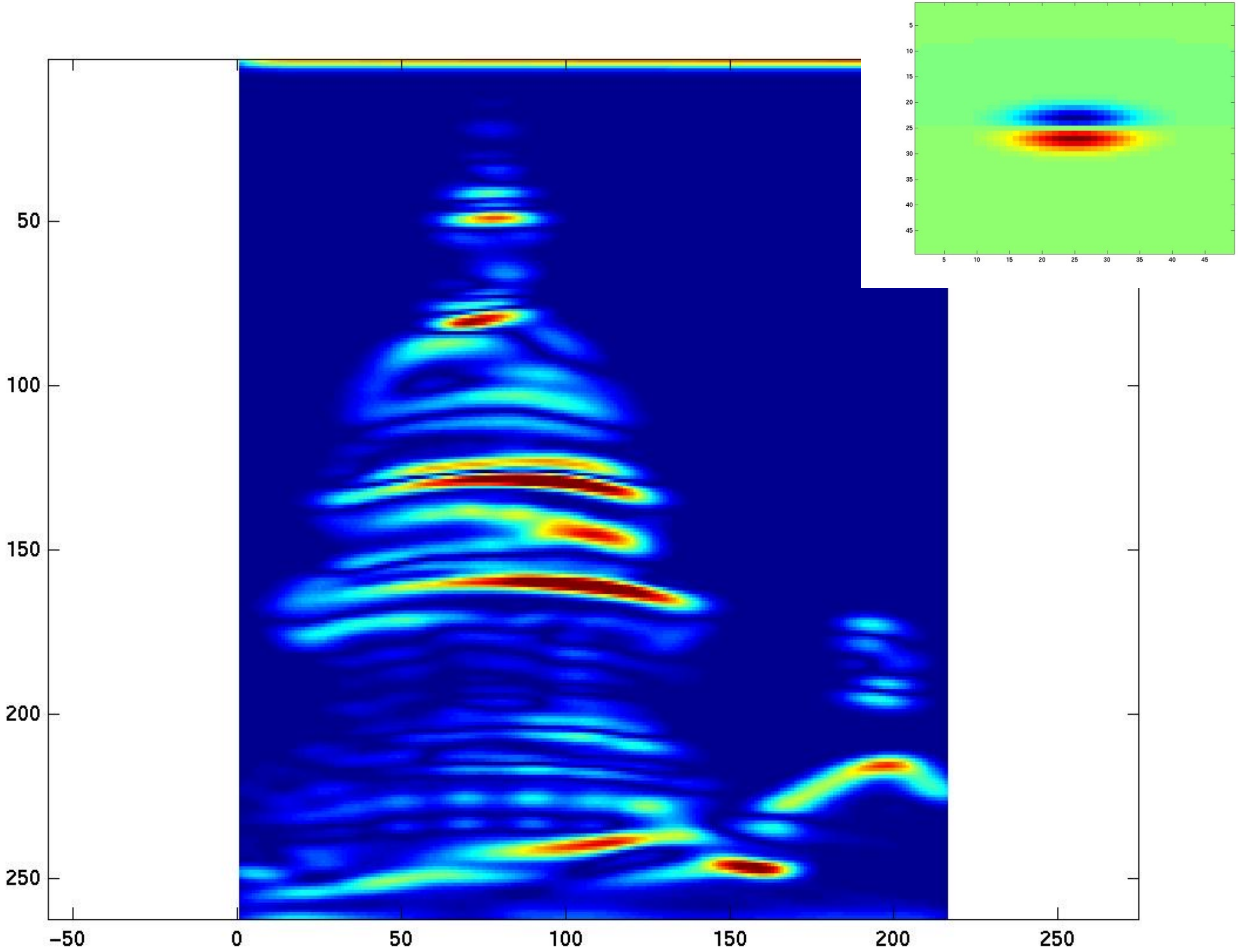
Kristen Grauman



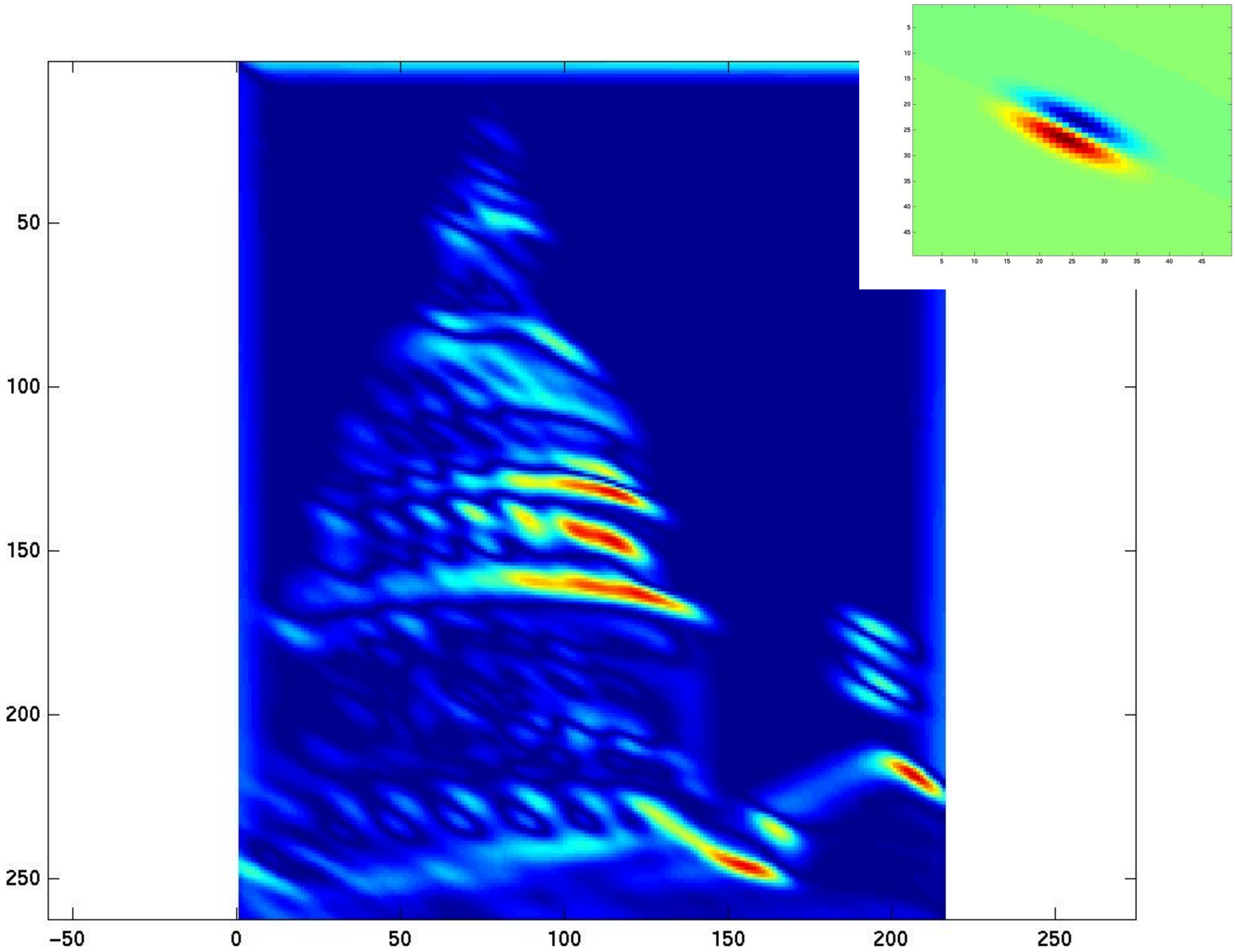
Kristen Grauman



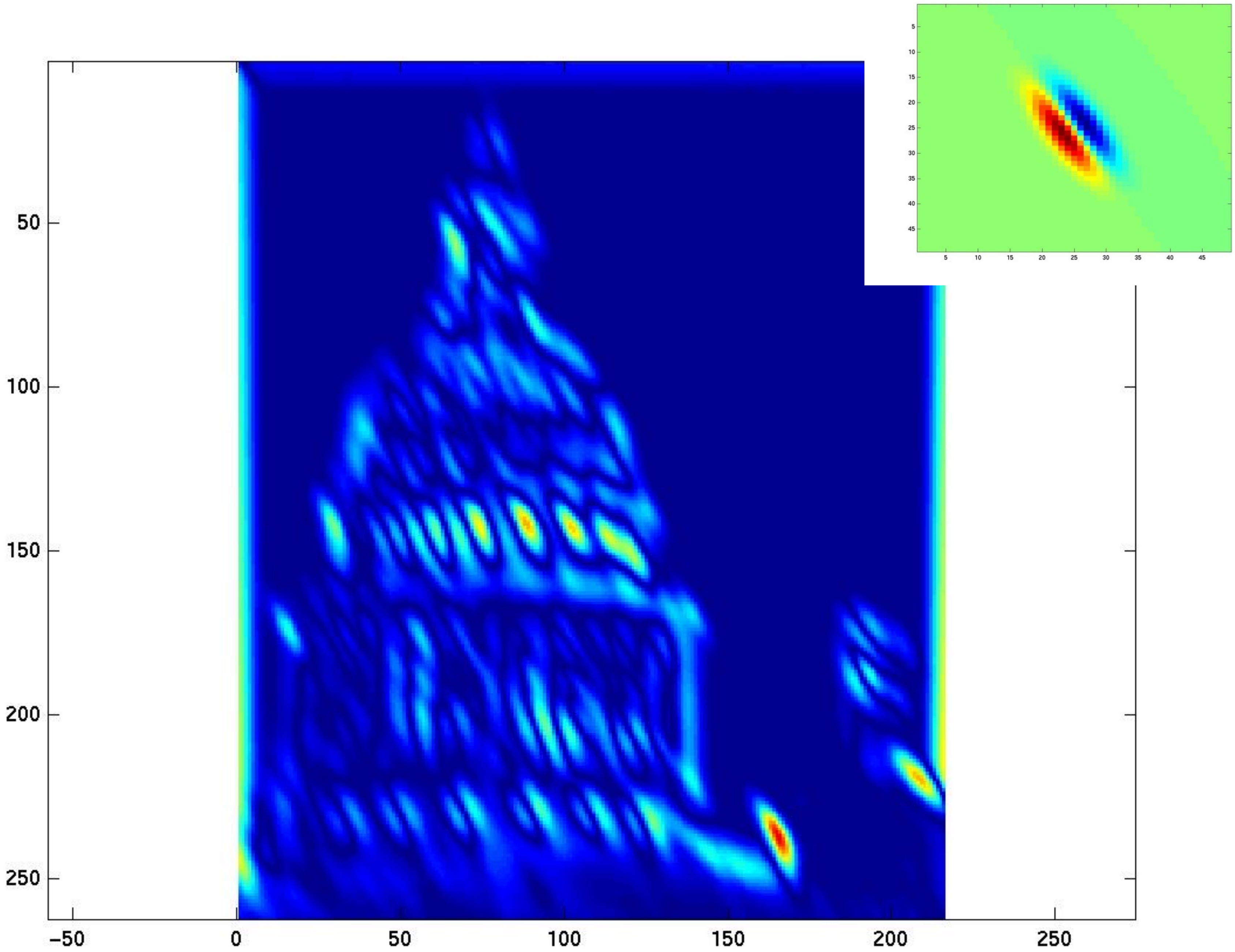
Kristen Grauman



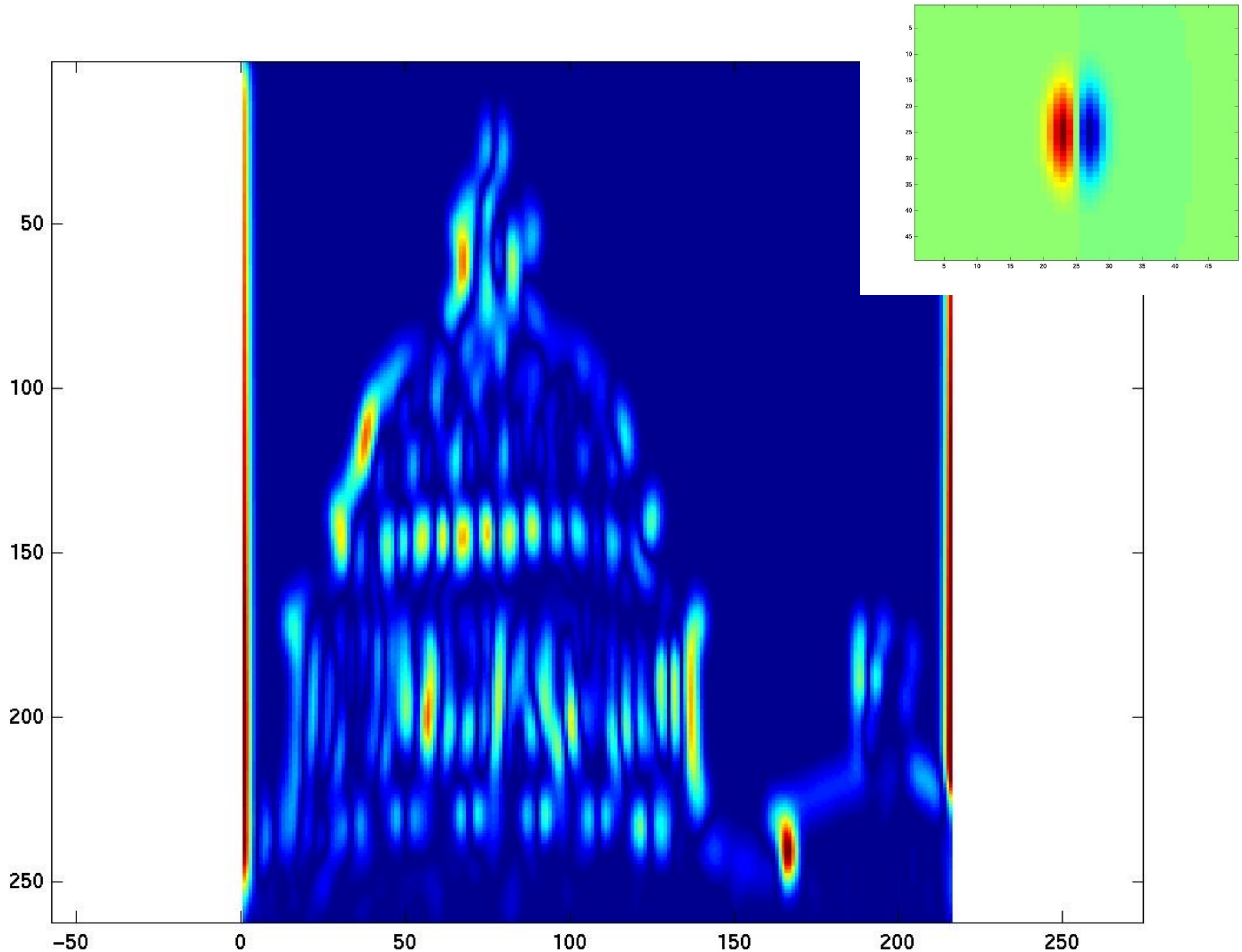
Kristen Grauman



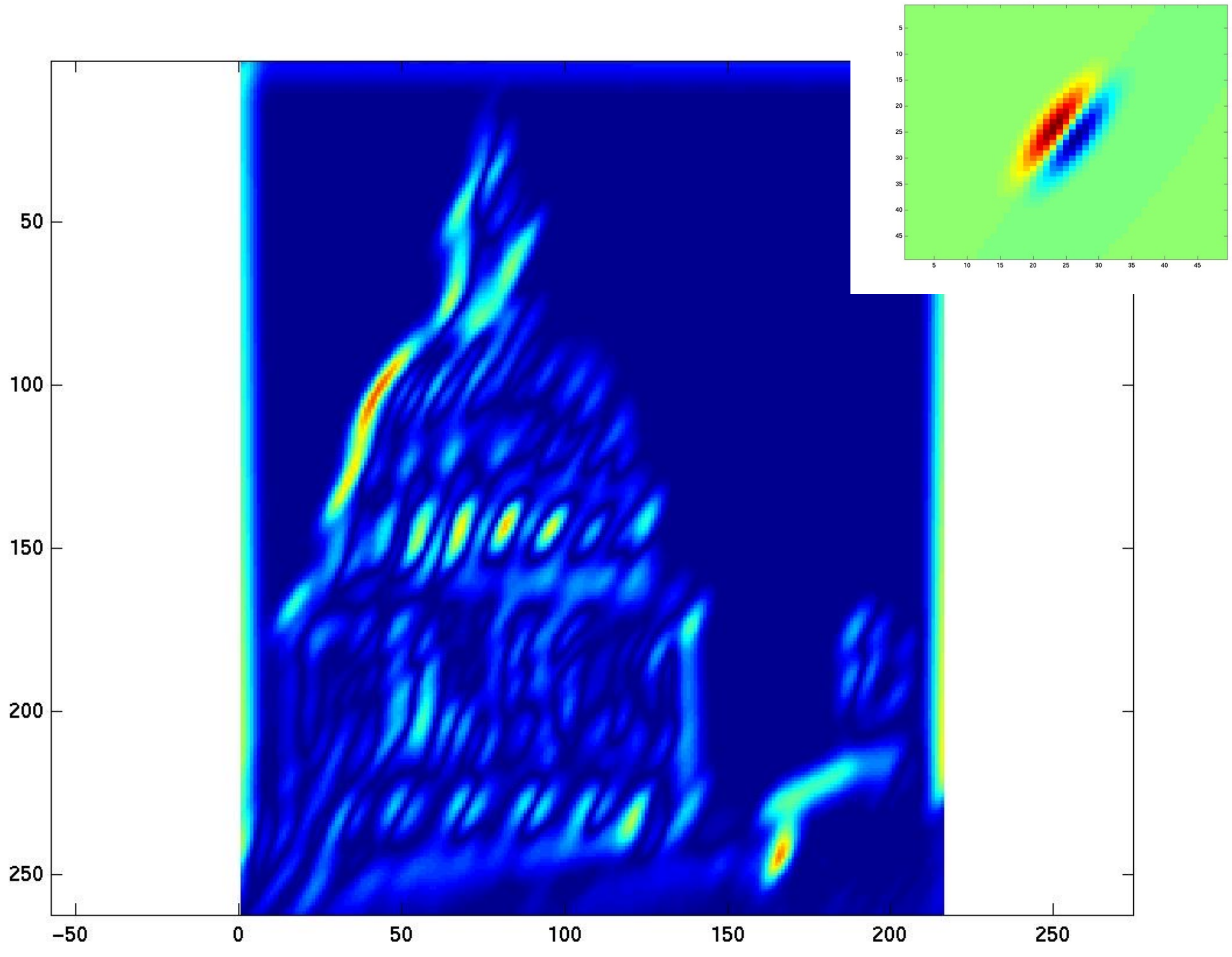
Kristen Grauman



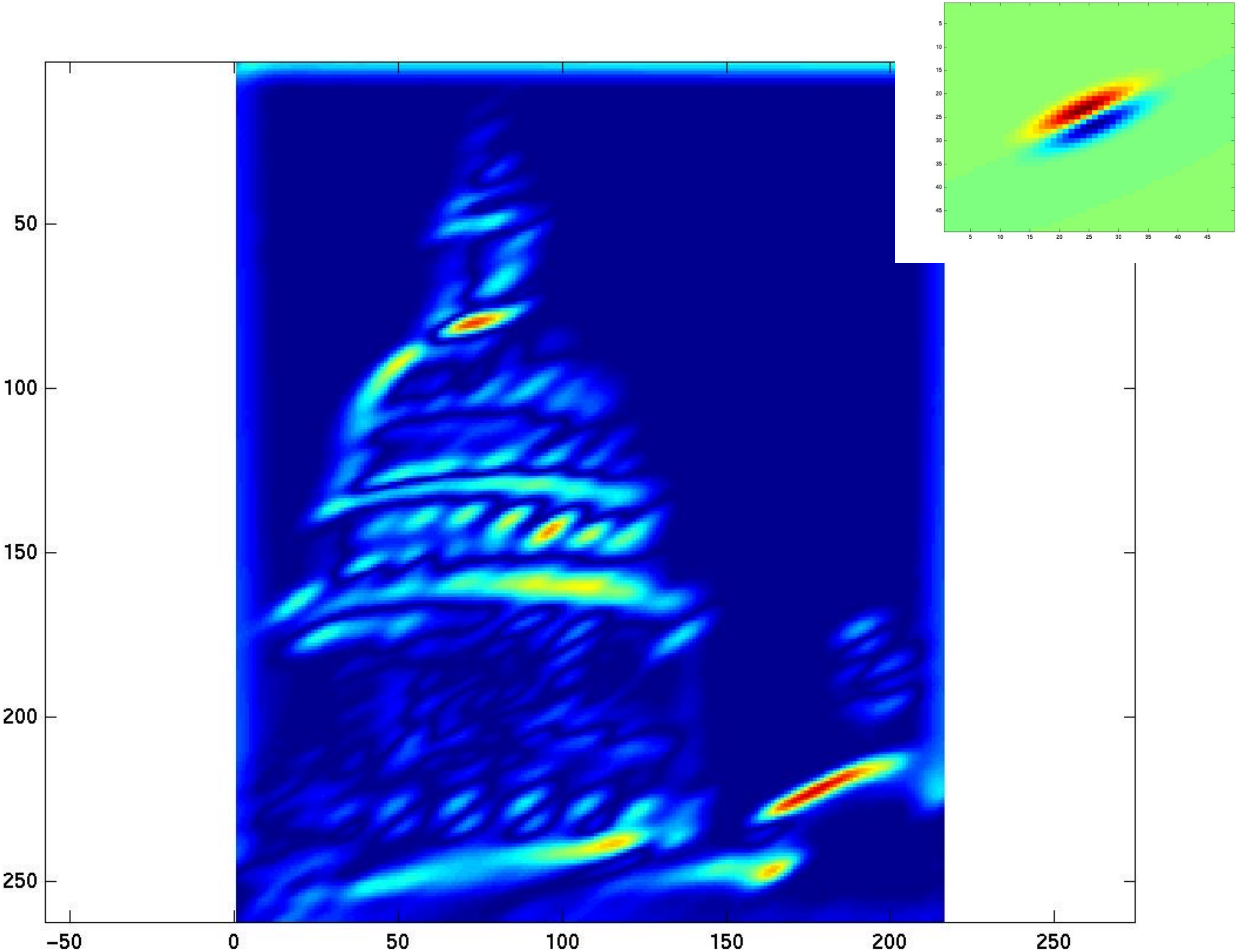
Kristen Grauman



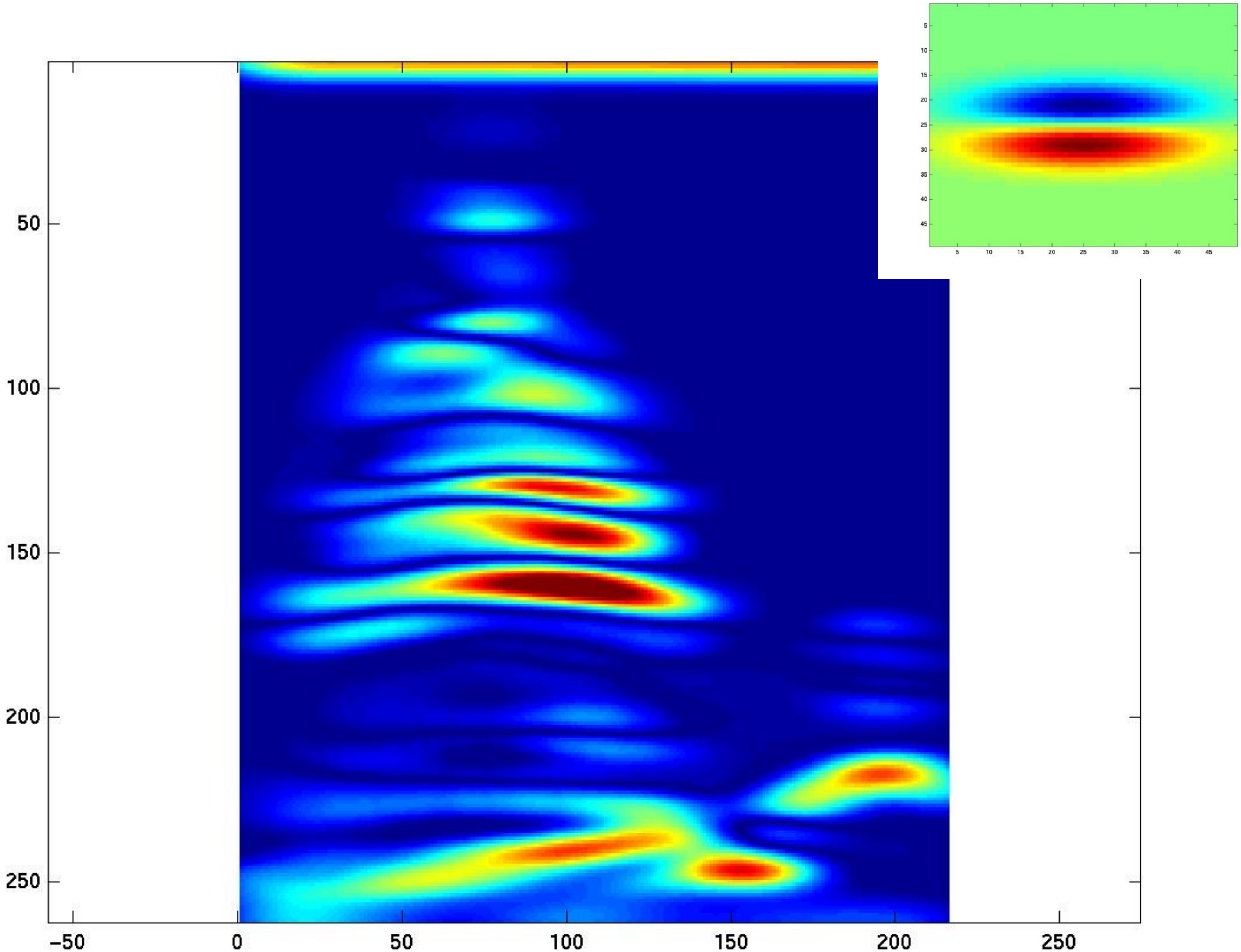
Kristen Grauman



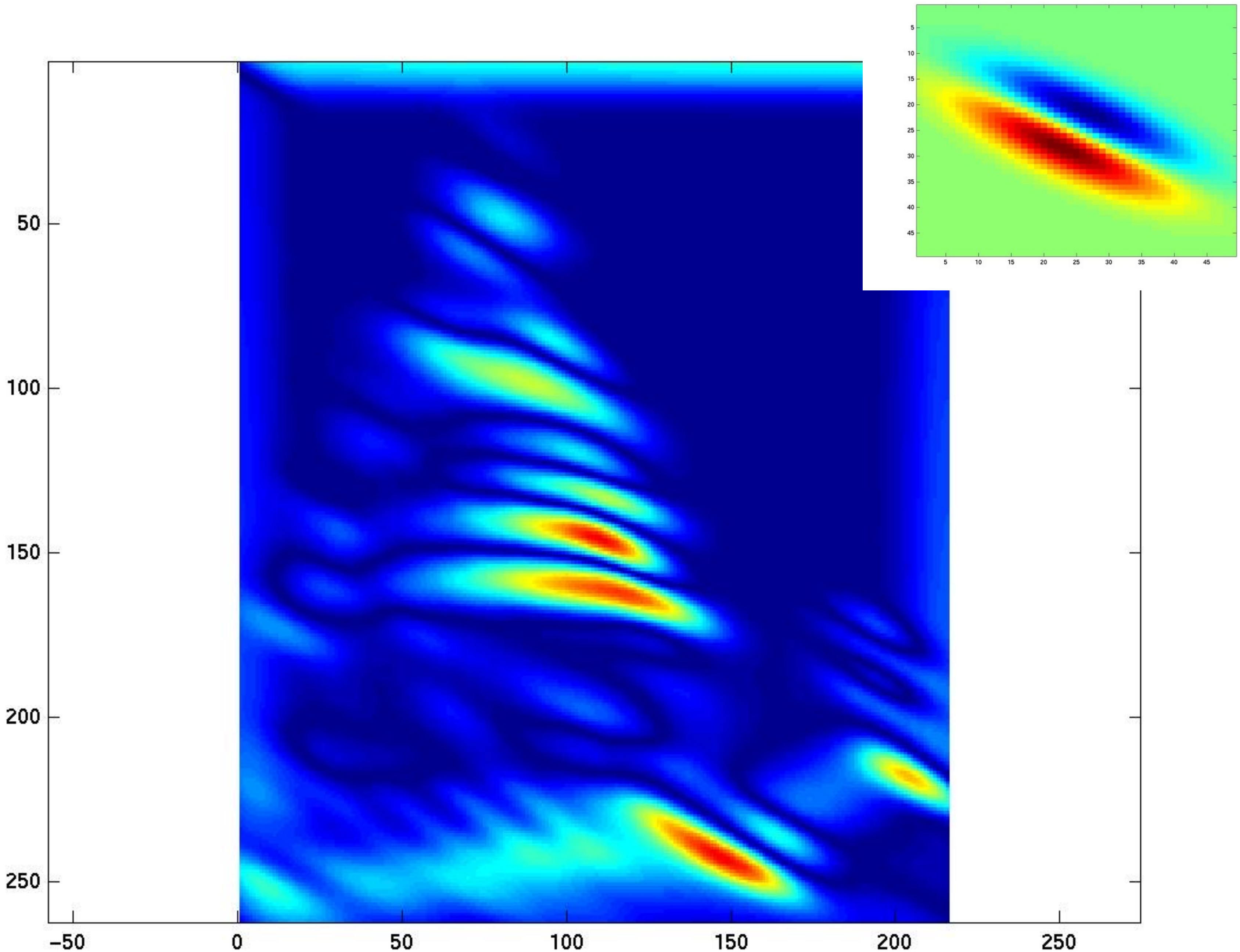
Kristen Grauman



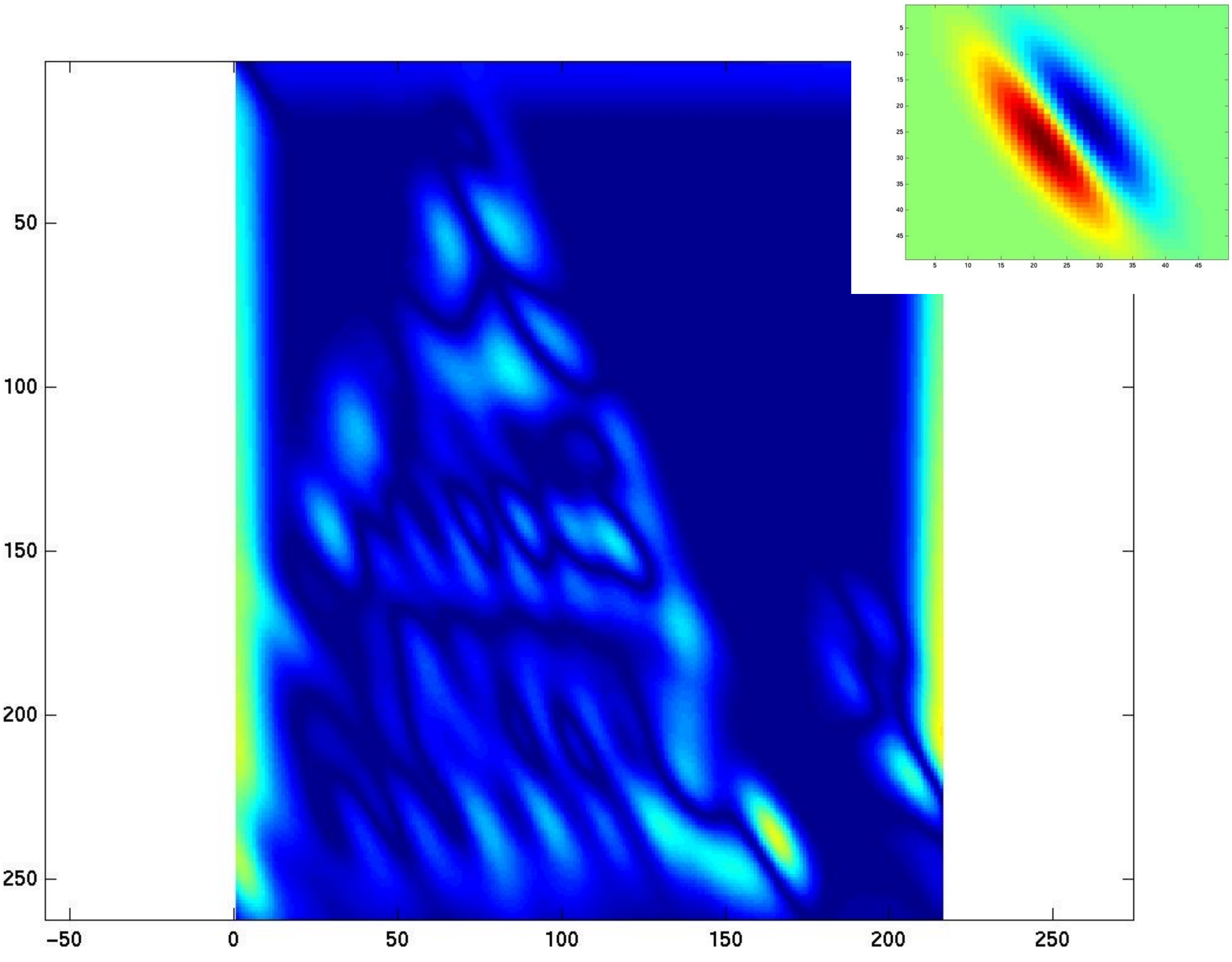
Kristen Grauman



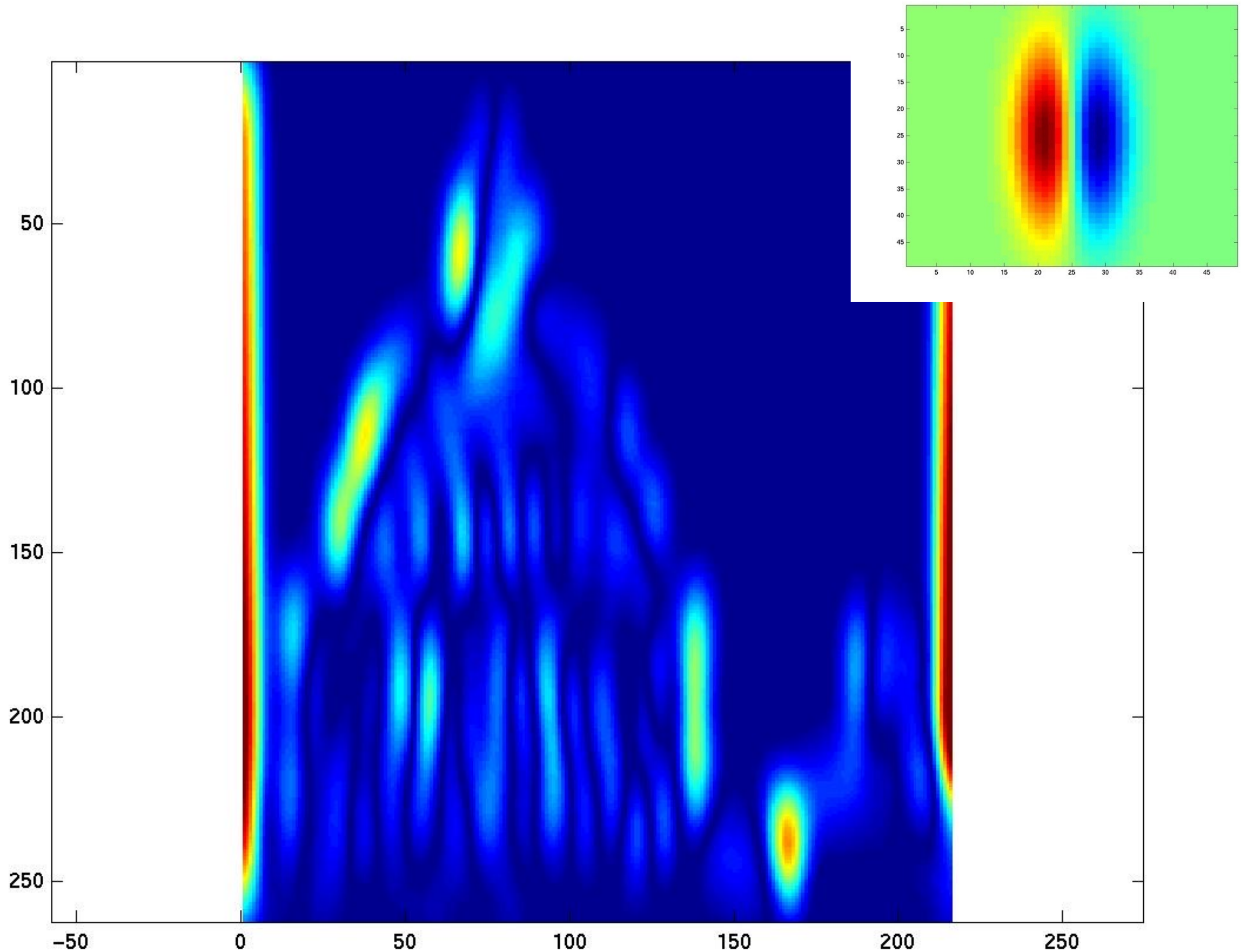
Kristen Grauman



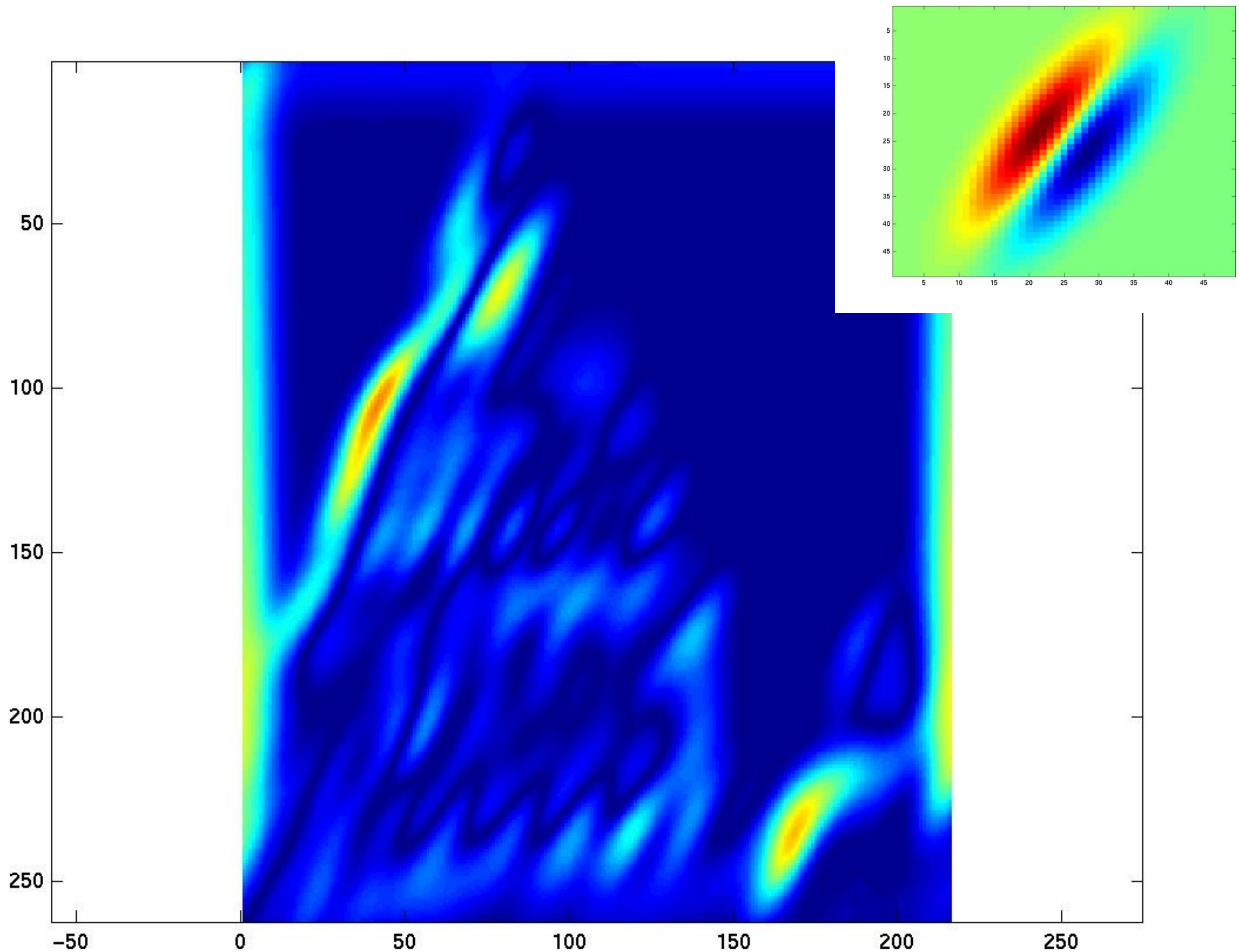
Kristen Grauman



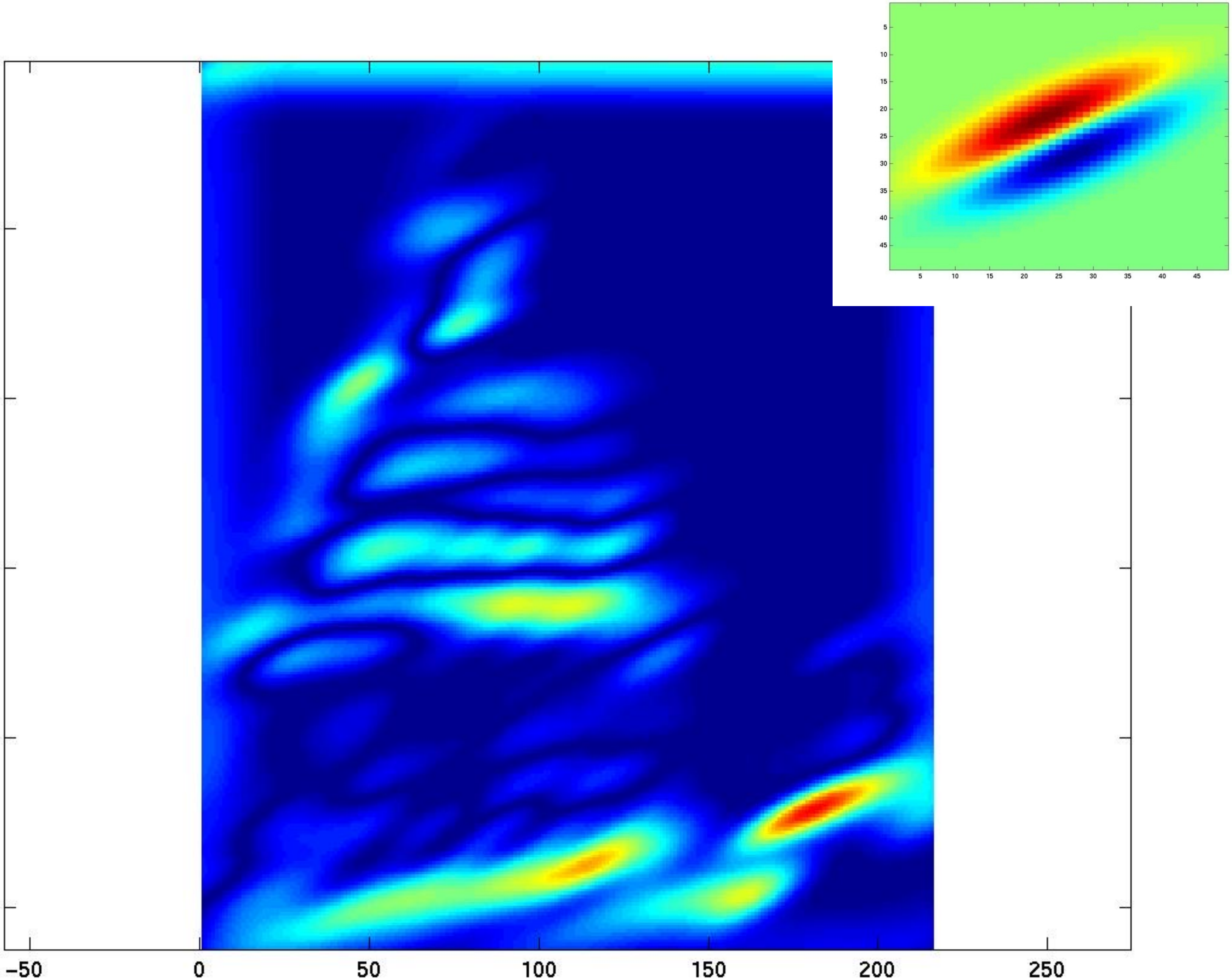
Kristen Grauman



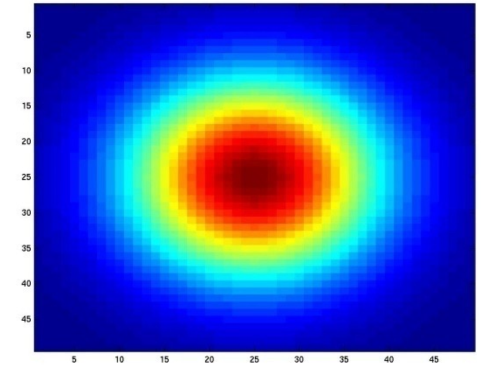
Kristen Grauman



Kristen Grauman



Kristen Grauman



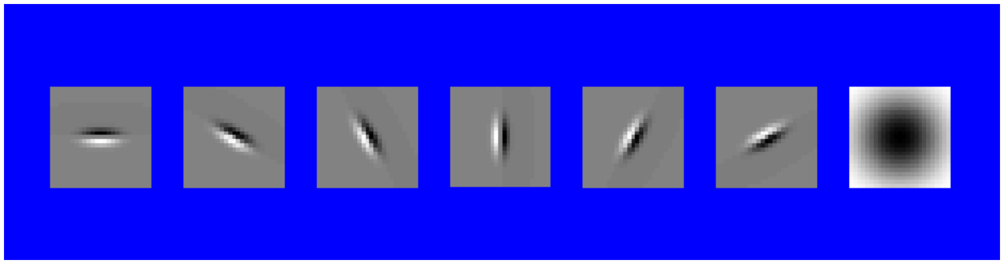
Kristen Grauman

How can we represent texture?

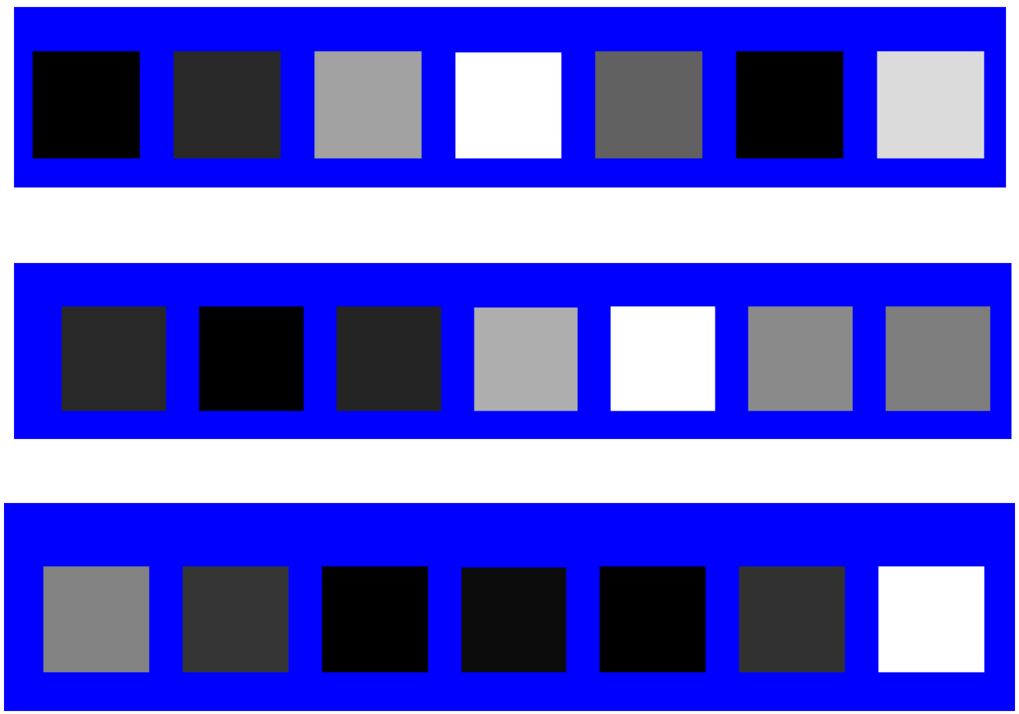
- Idea 1: Record simple statistics (e.g., mean, std.) of absolute filter responses

Can you match the texture to the response?

Filters

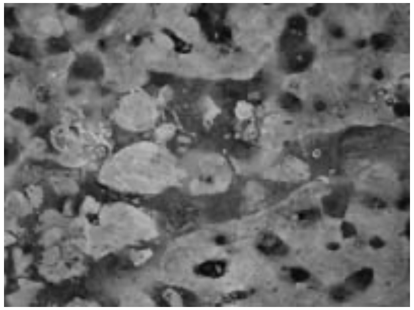


1
2
3



Mean abs responses

A



B

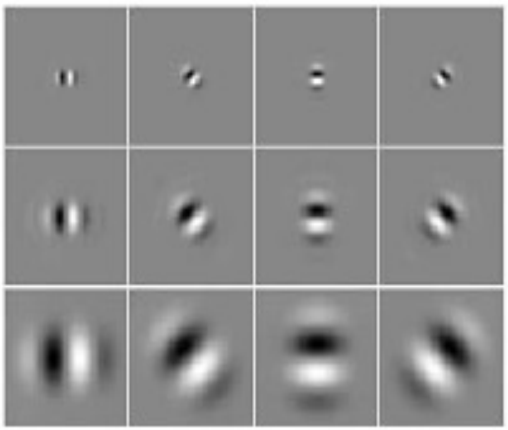
C

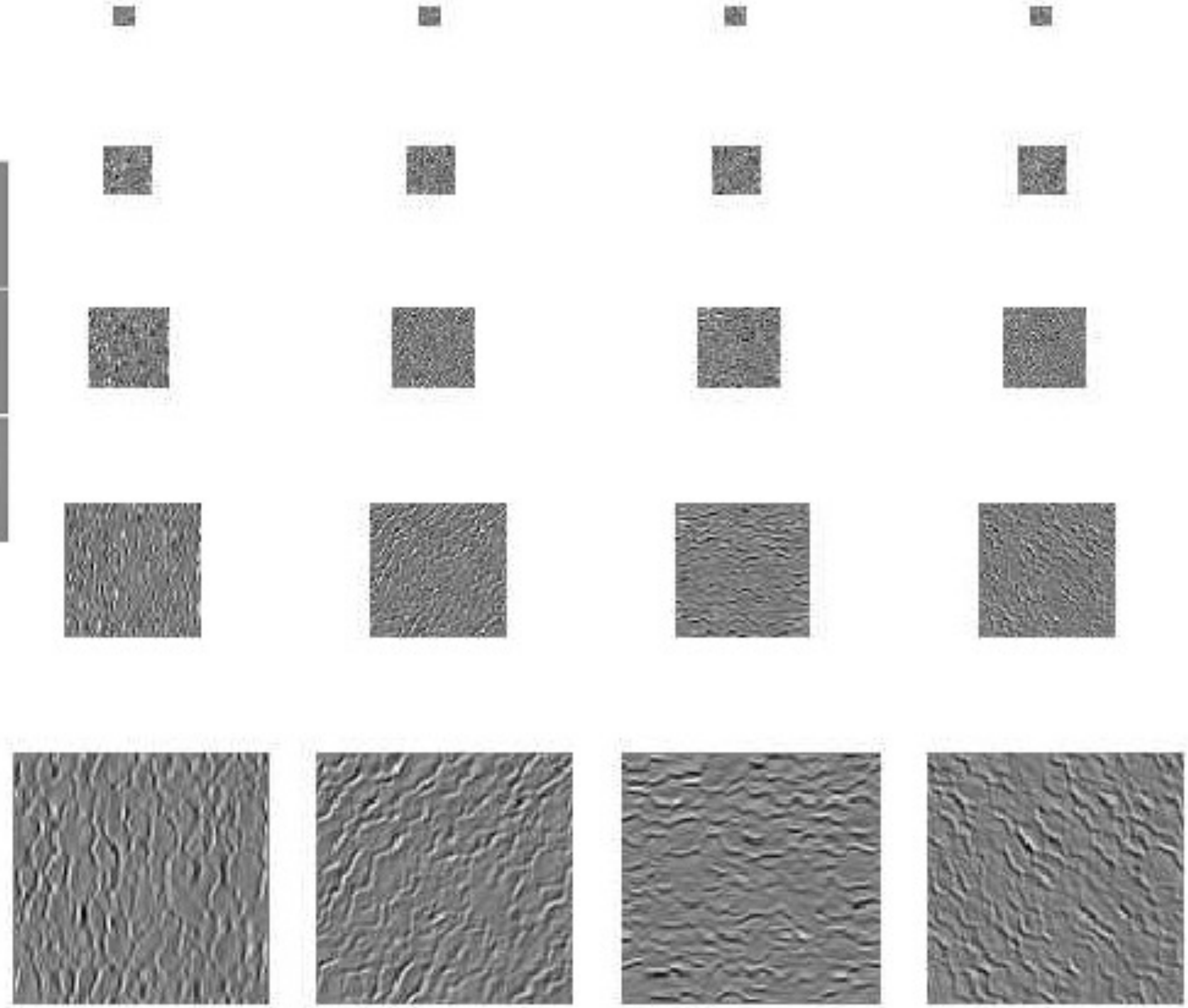
How can we represent texture?

- Can be thought of as an single “orientation histogram”
- Idea 2: Marginal histograms of filter responses
  - one histogram per filter

Multi-scale filter decomposition

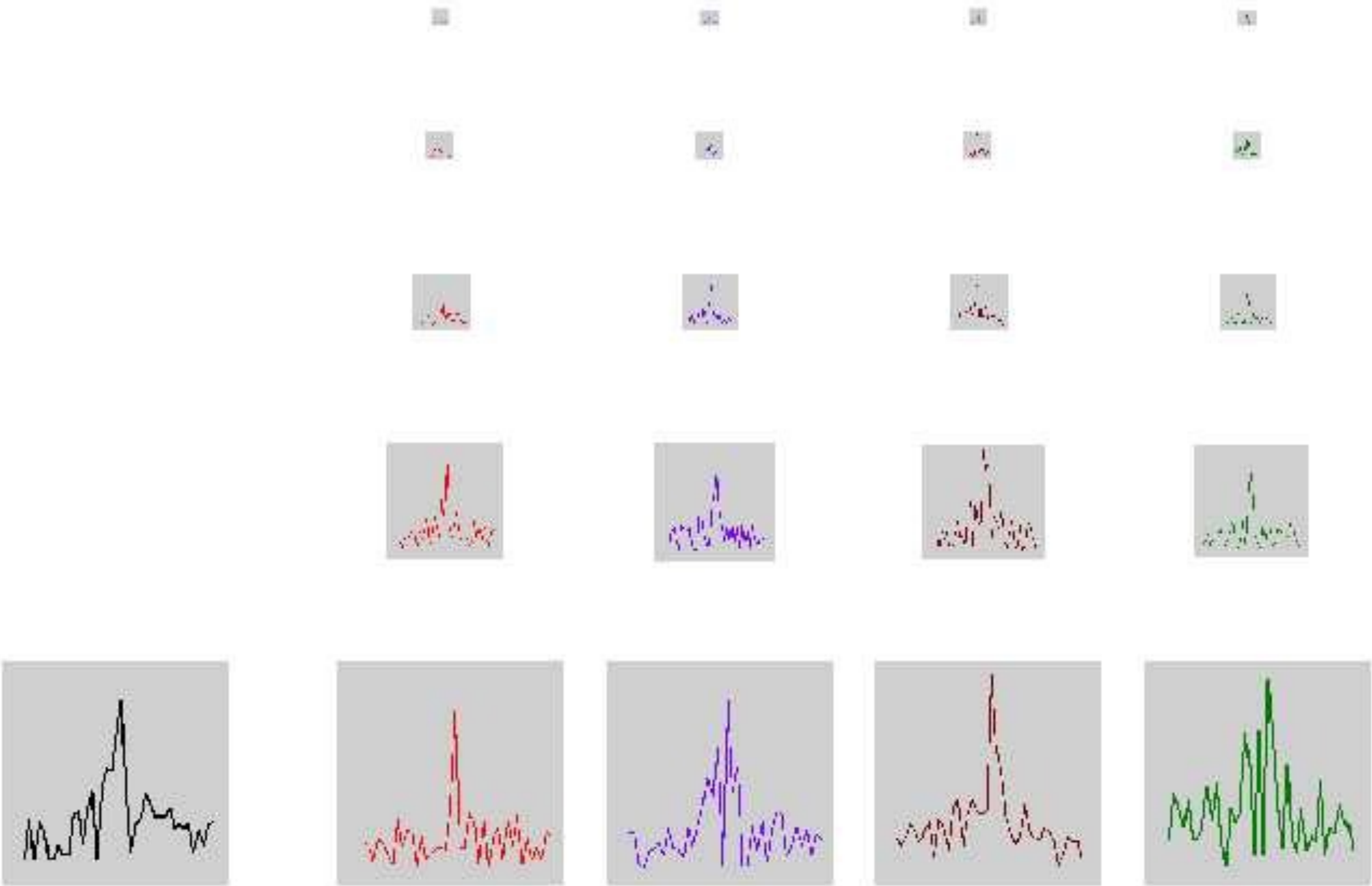
Filter bank





Input image

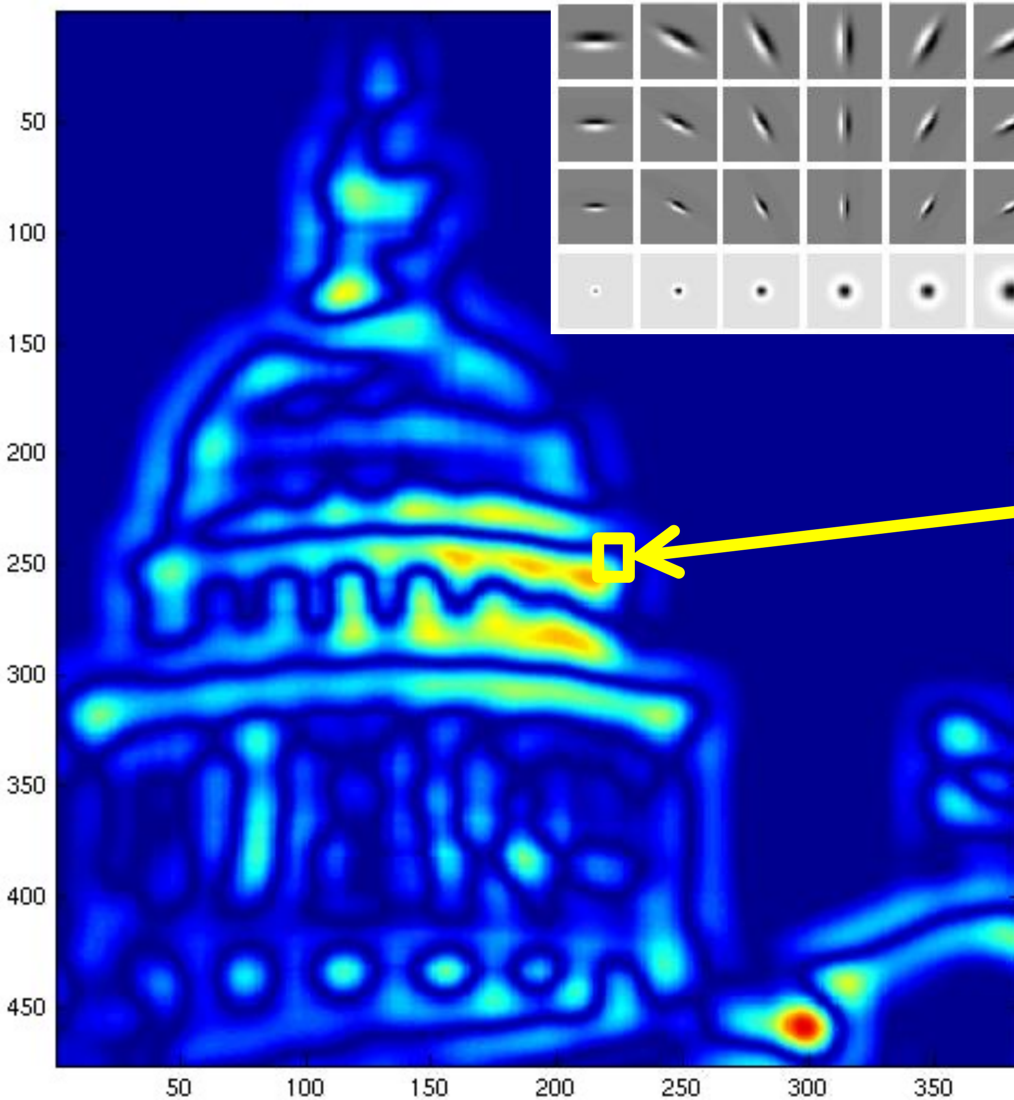
Filter response histograms

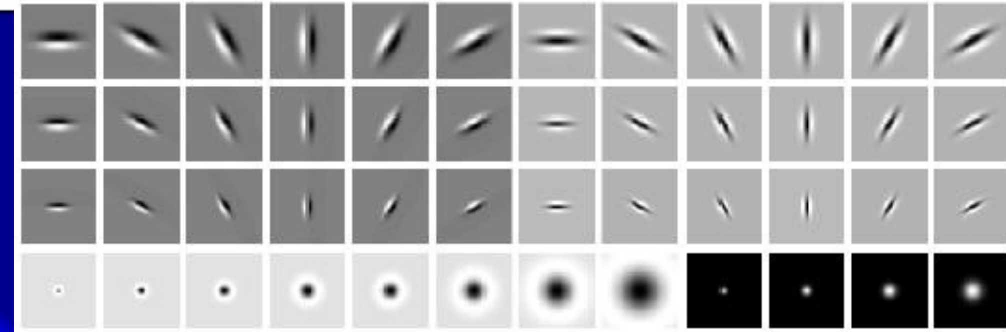


How can we represent texture?

- Marginal filter response histograms don't talk to each other (in a direct way)
- Idea 3: Histograms of joint responses (textons)

Filter Response



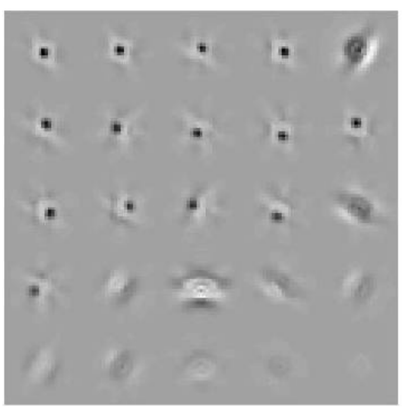


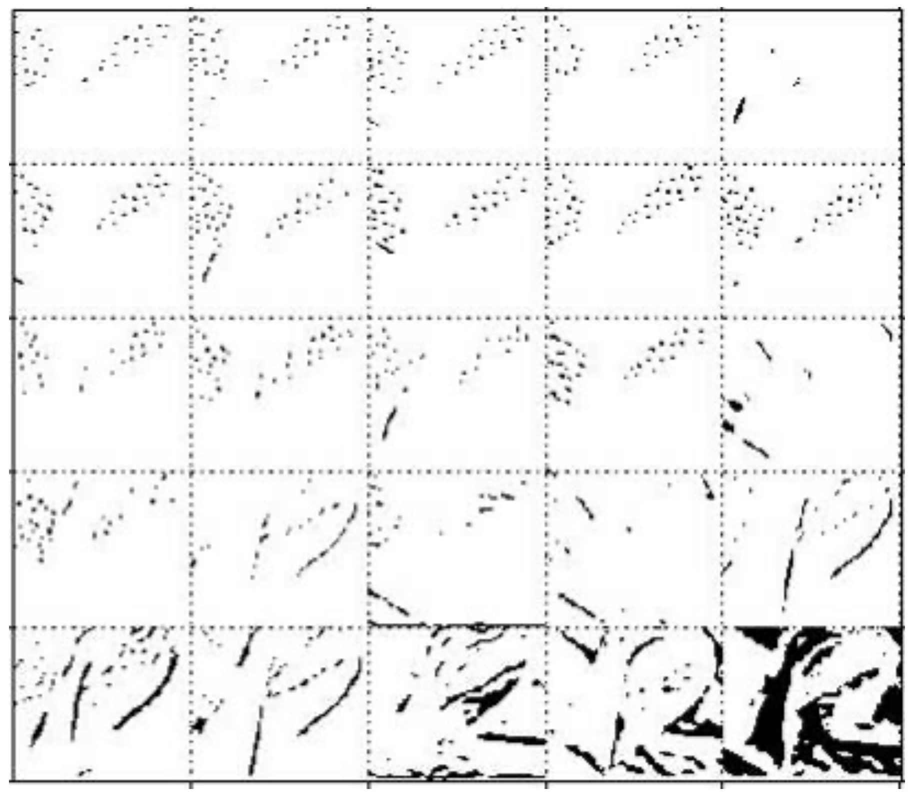
[r1, r2, …, r38]

We can form a feature vector from the list of responses at each pixel.

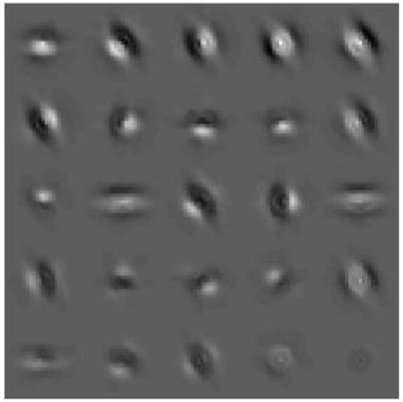
Textons (Malik et al, IJCV 2001)

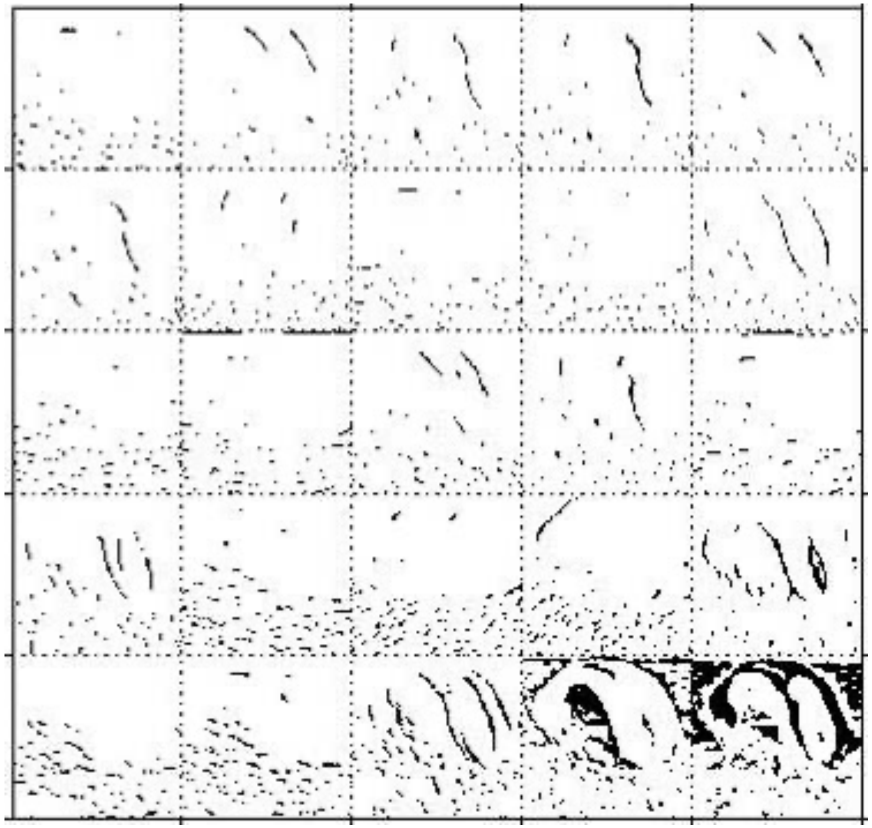
- Cluster vectors of filter responses





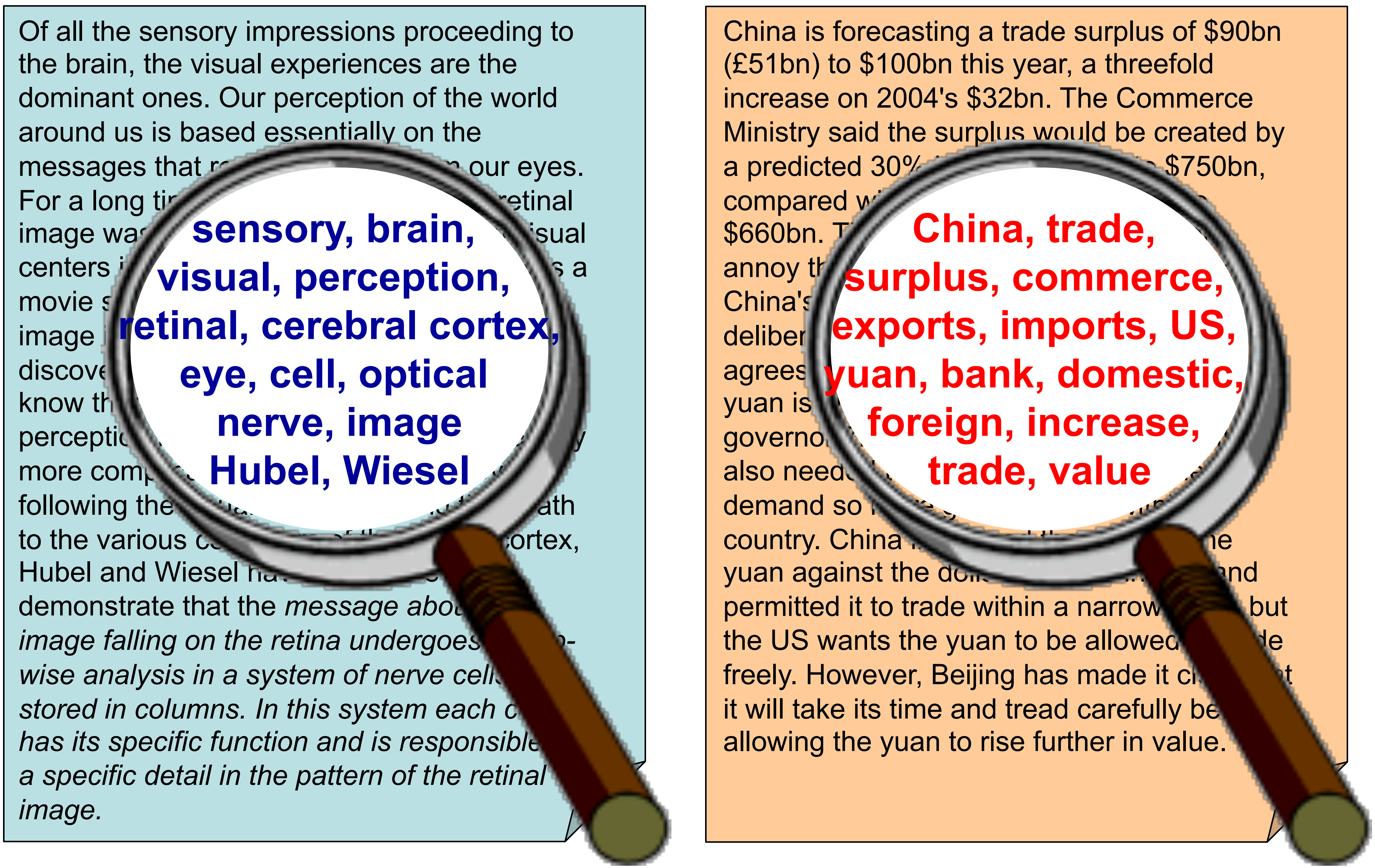
Textons (cont.)

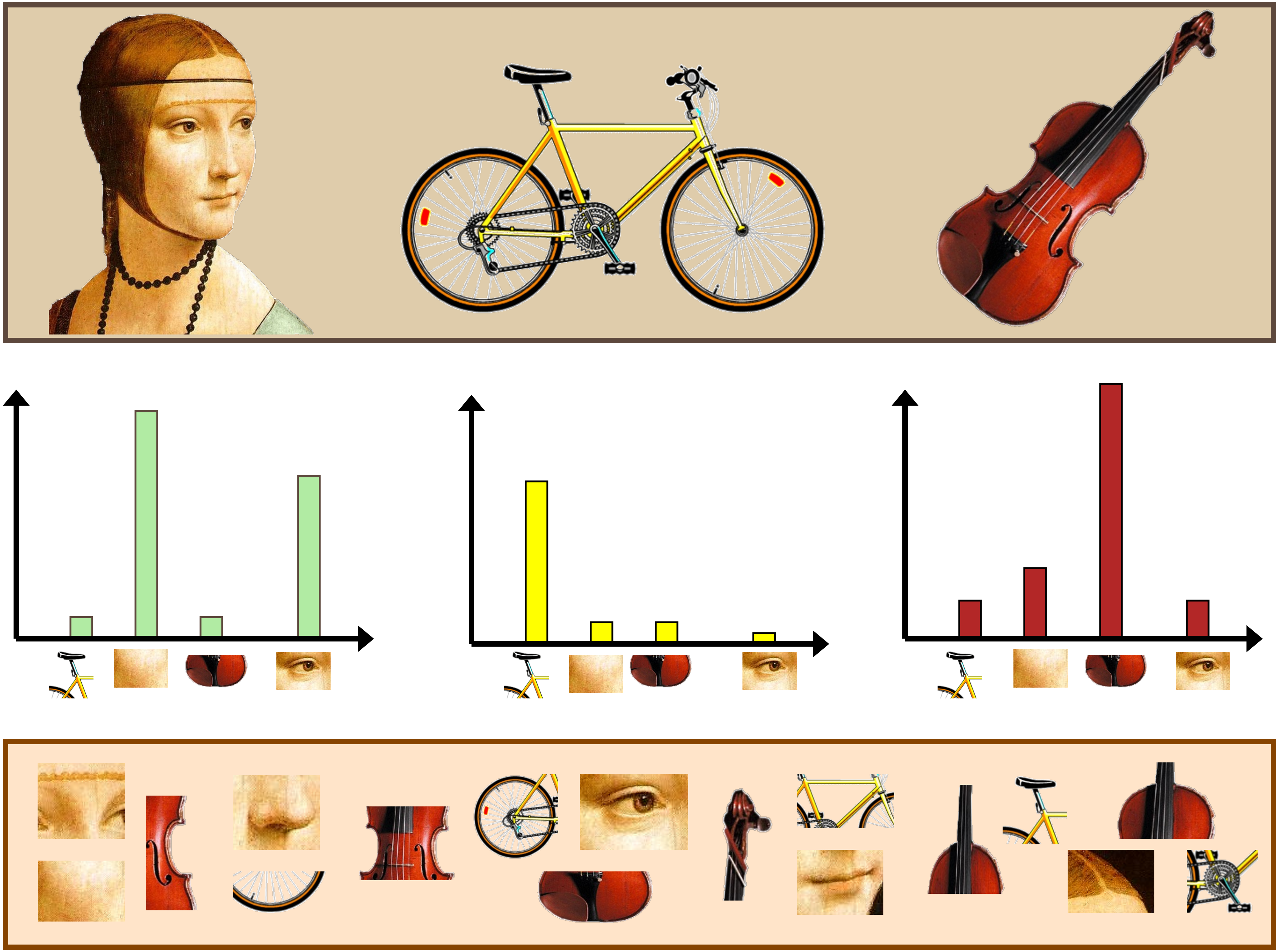




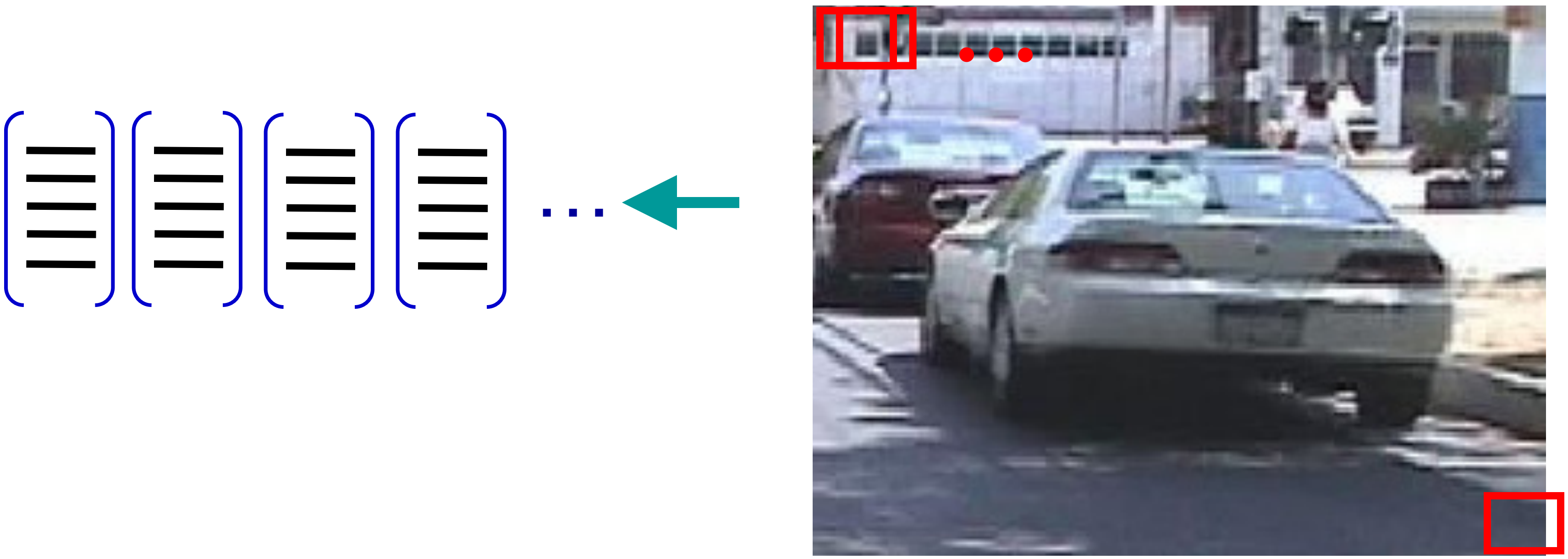
Object → **Bag of 'words'**

Analogy to documents

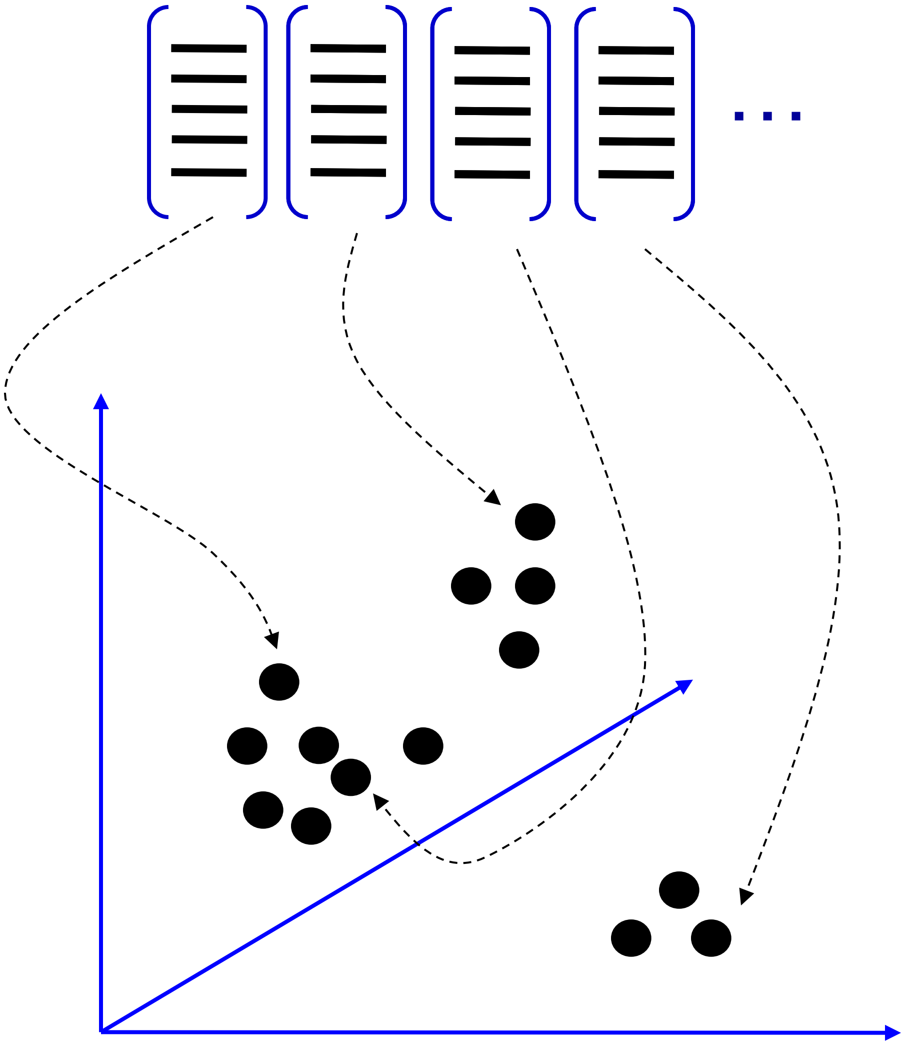




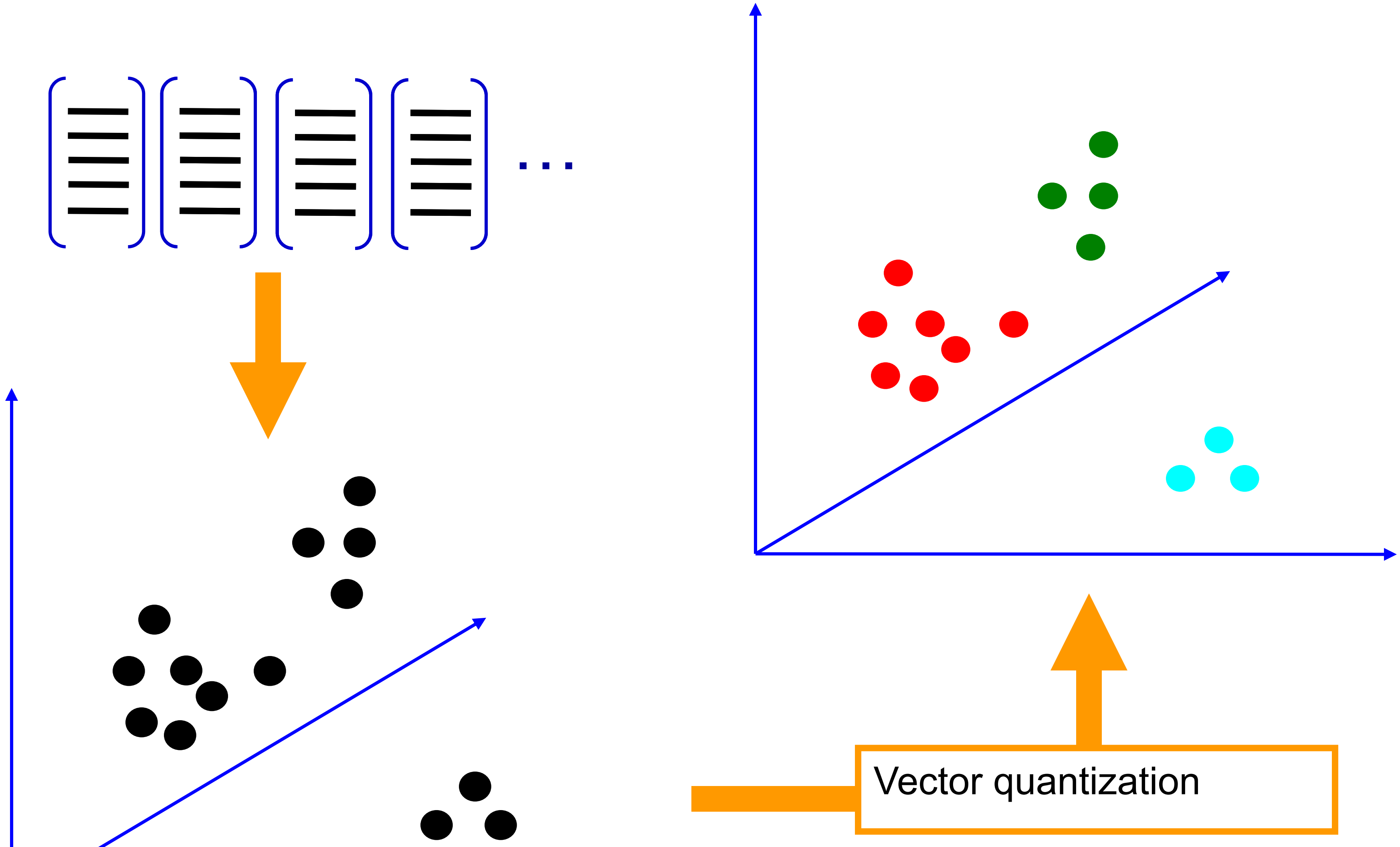
Patch Features



dictionary formation



Clustering (usually k-means)



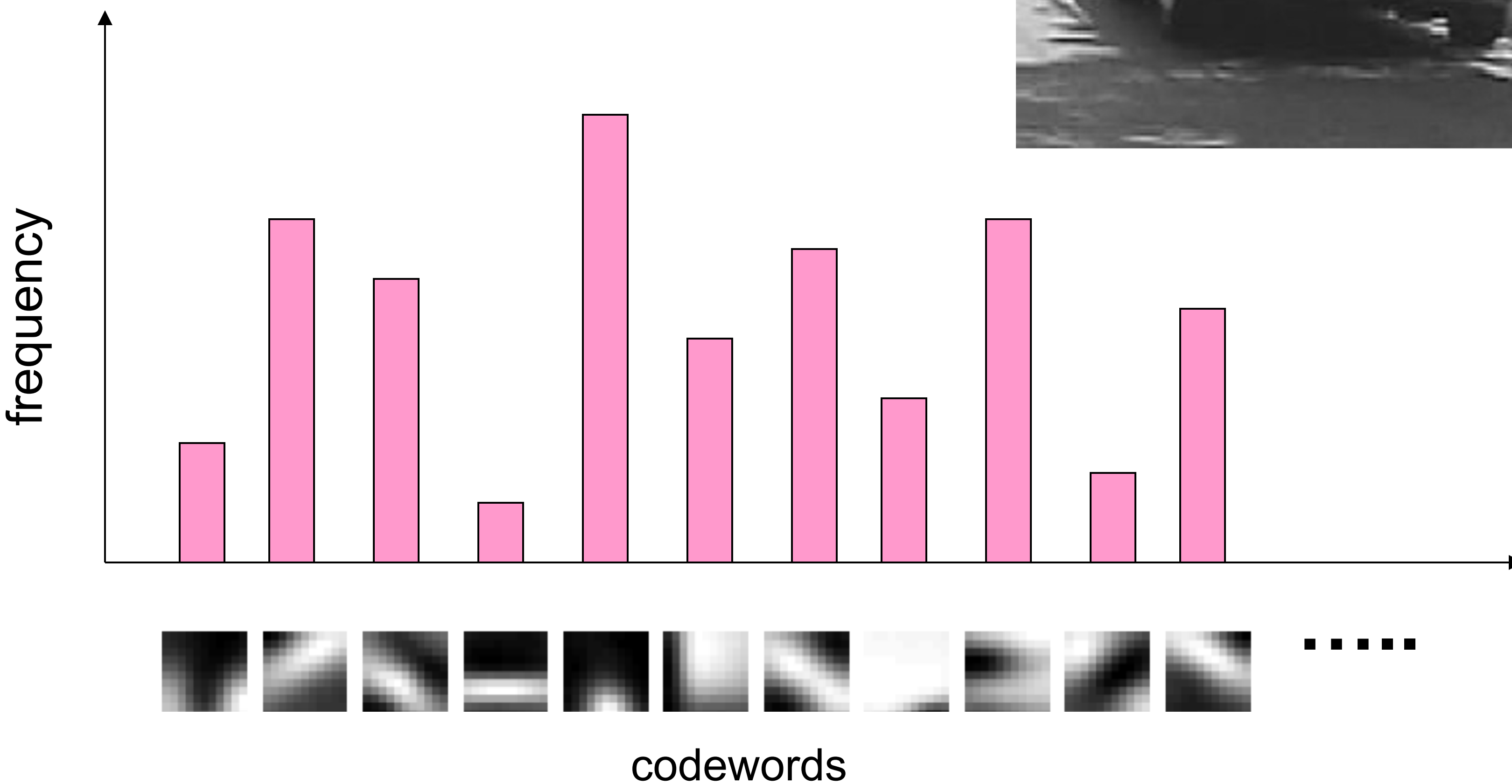
Slide credit: Josef Sivic

Feature Representation

Visual words, aka textons, aka keypoints:
K-means clustered pieces of the image

- Various Representations:
  - Filter bank responses (textons)
  - Image Patches
  - SIFT descriptors
- Either image-specific or “universal” dictionary

Image representation



Scene Classification (Renninger & Malik)

beach

mountain

forest

city

street

farm

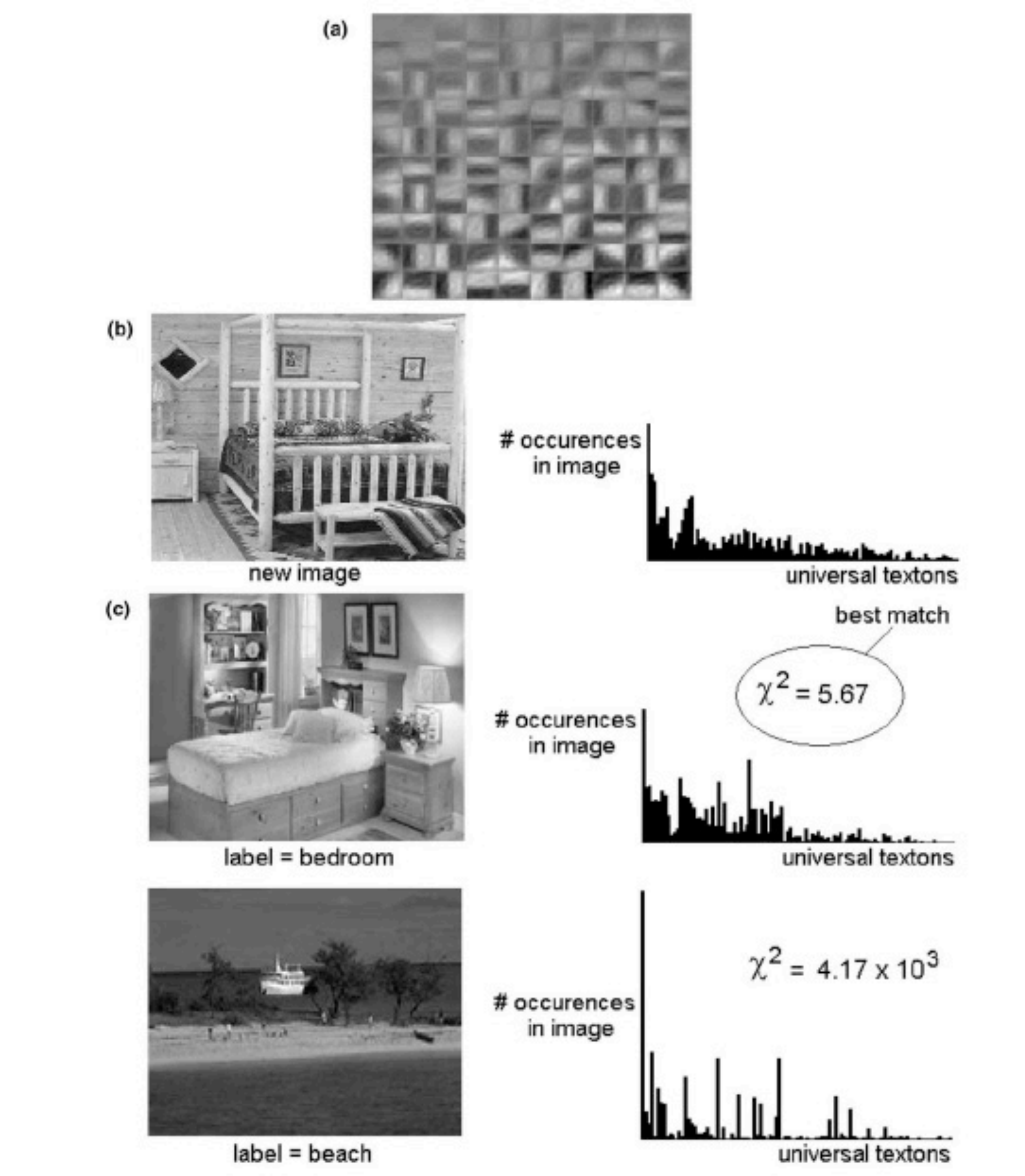
kitchen

livingroom

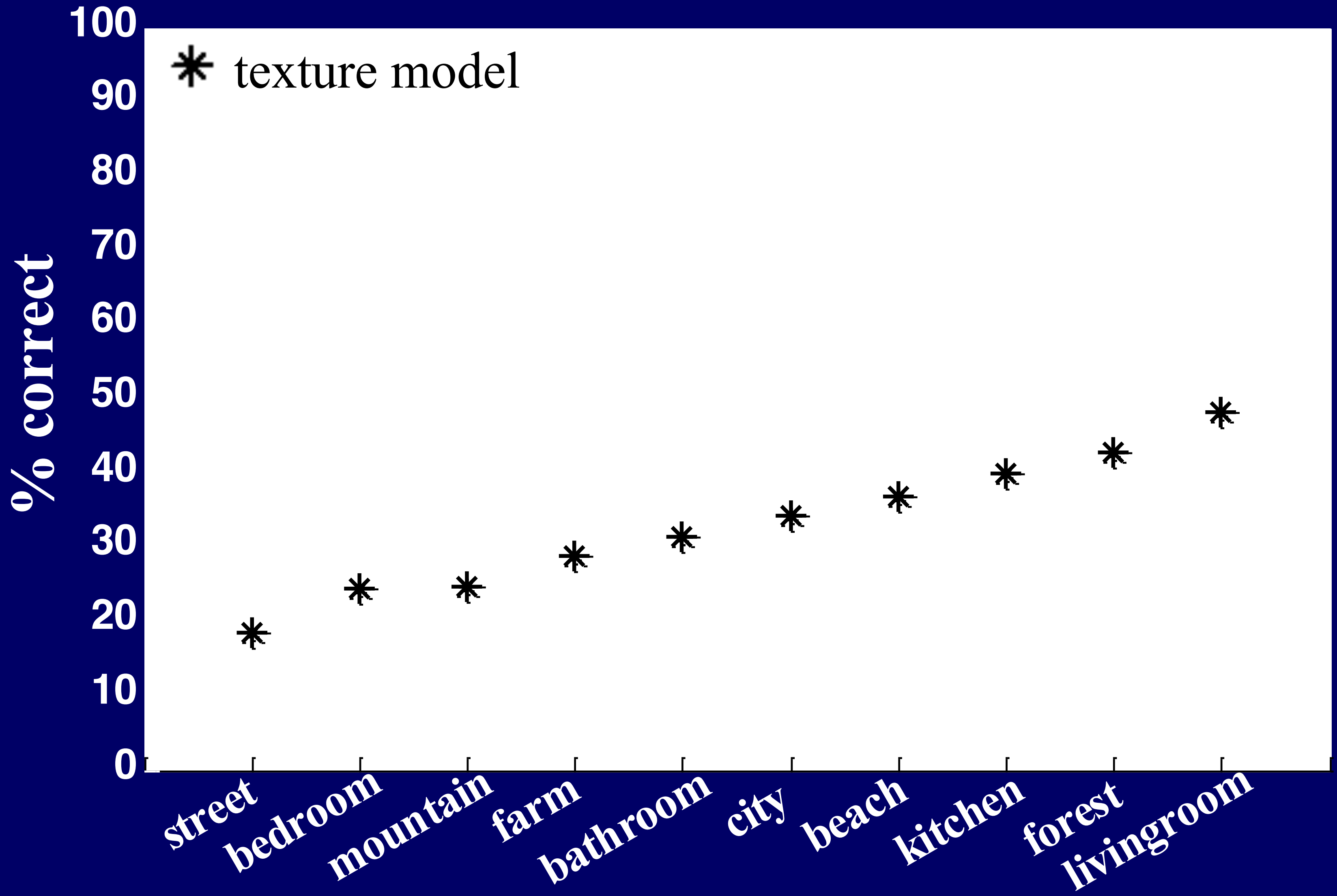
bedroom

bathroom

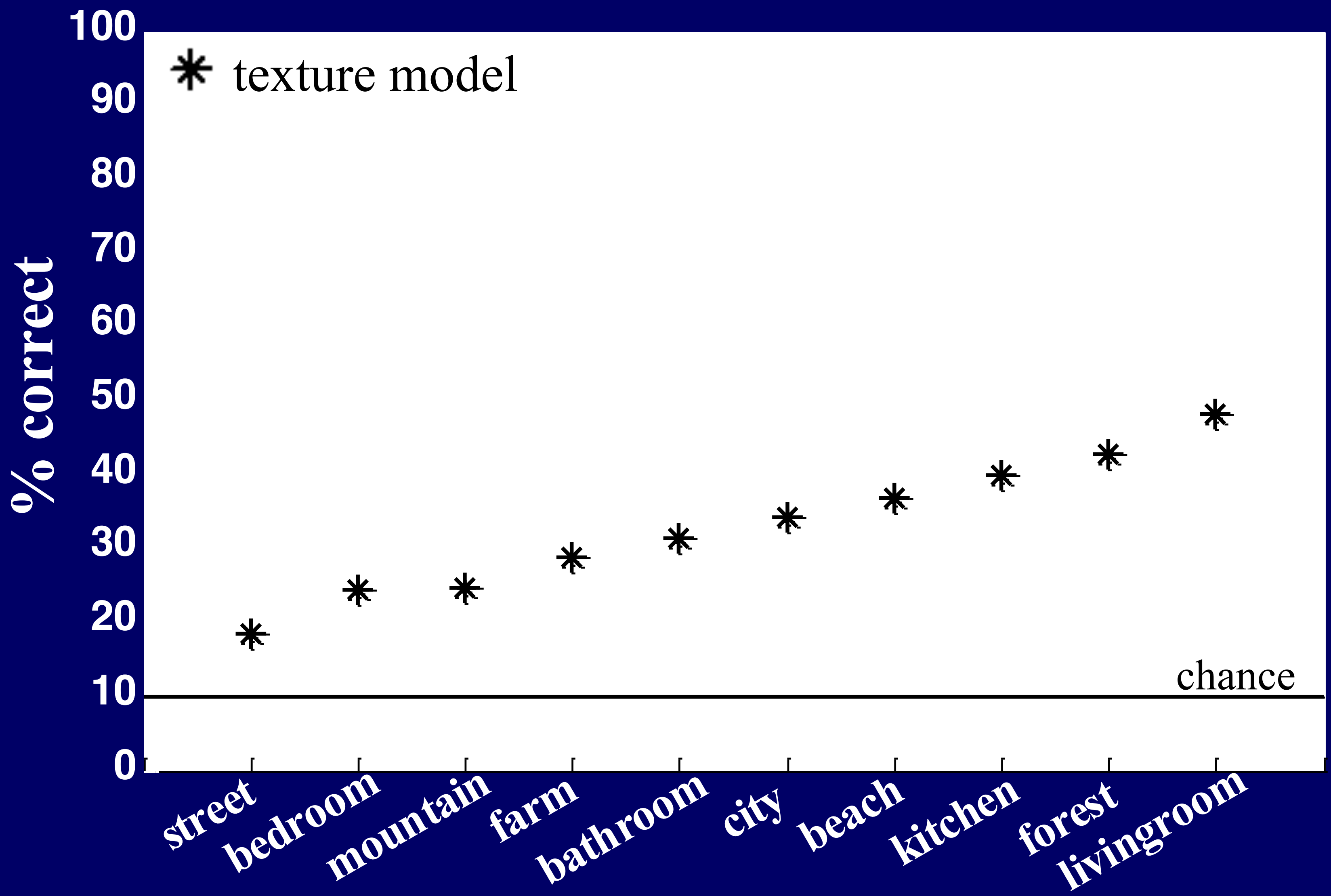
Texton Histogram Matching



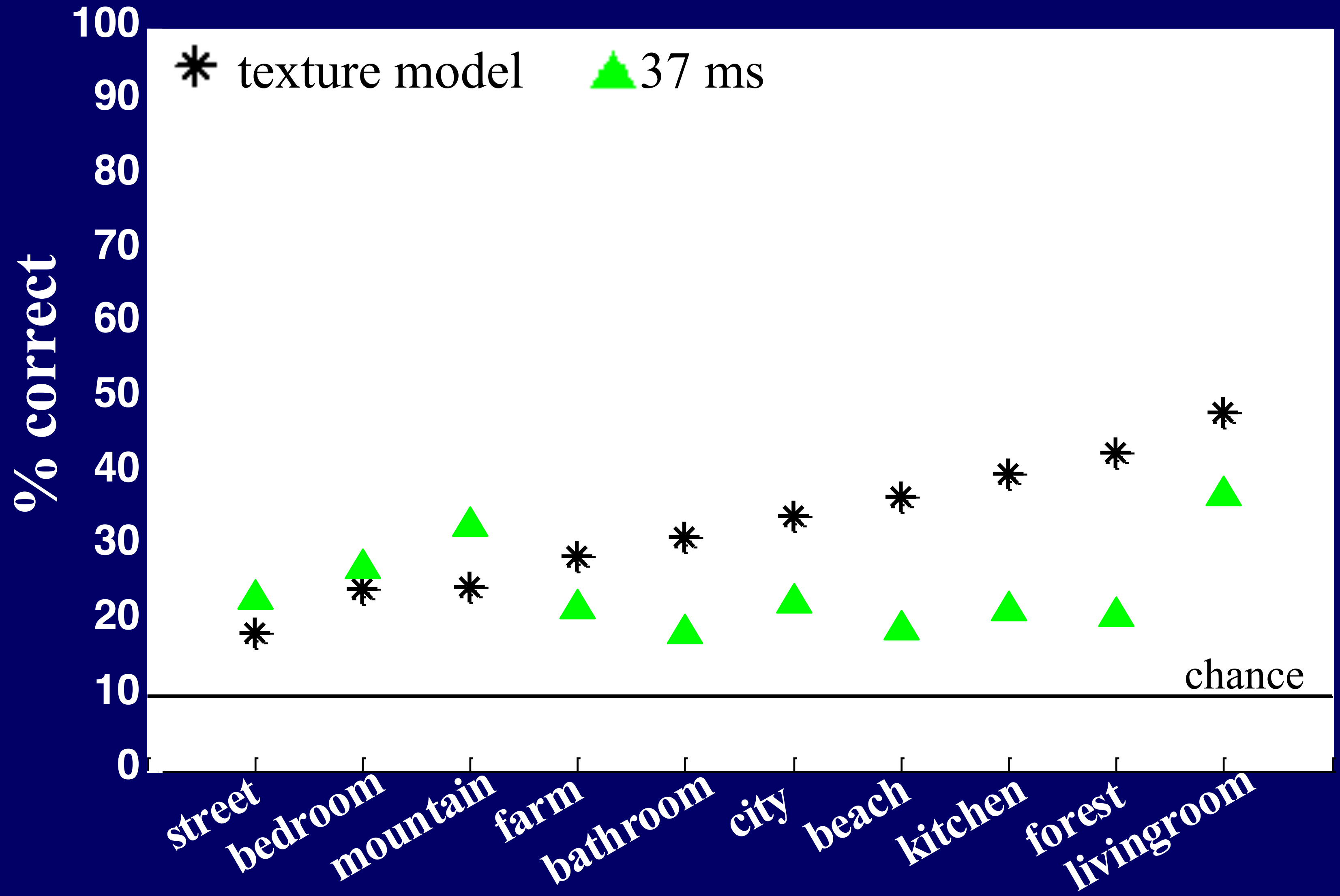
Discrimination of Basic Categories



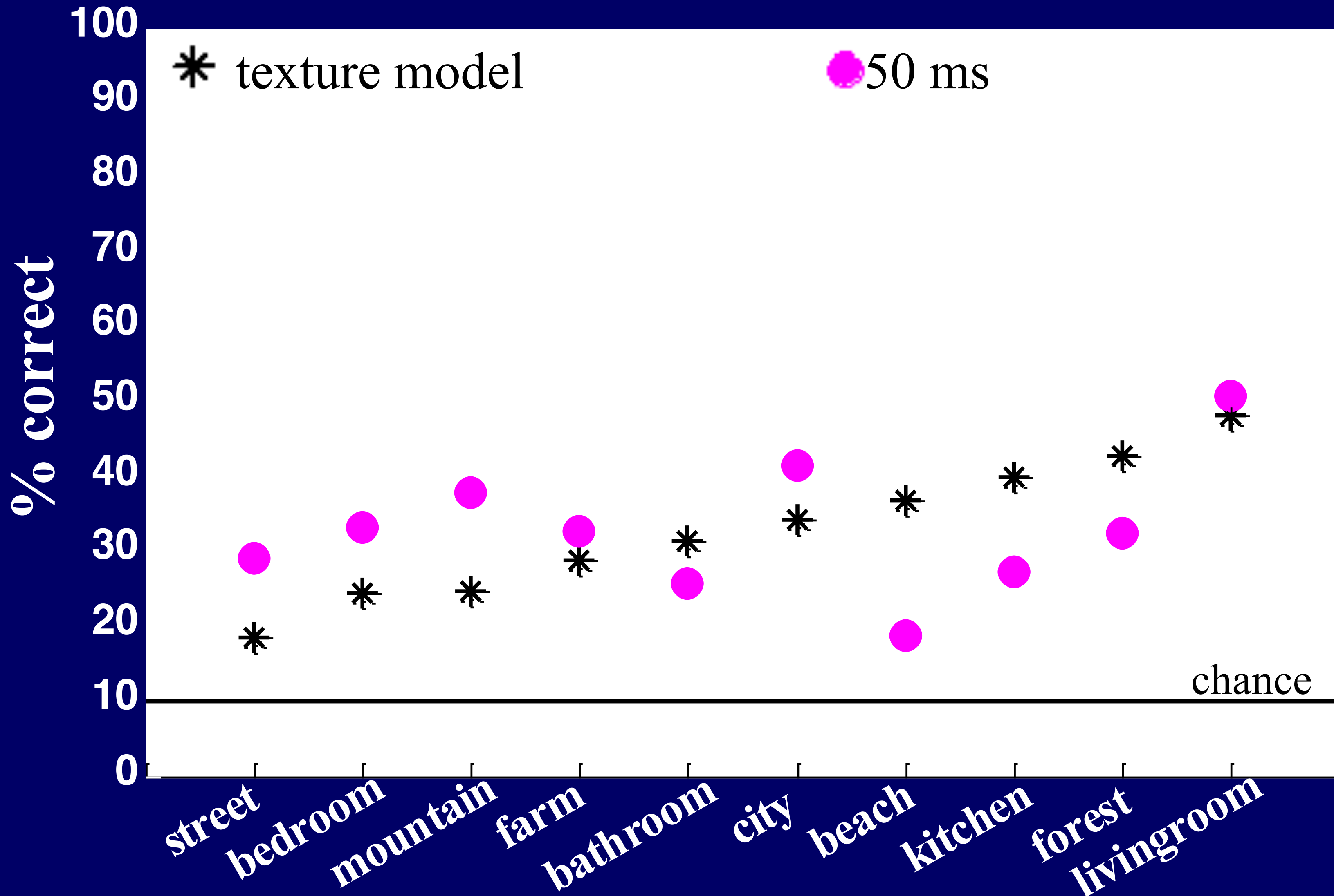
Discrimination of Basic Categories



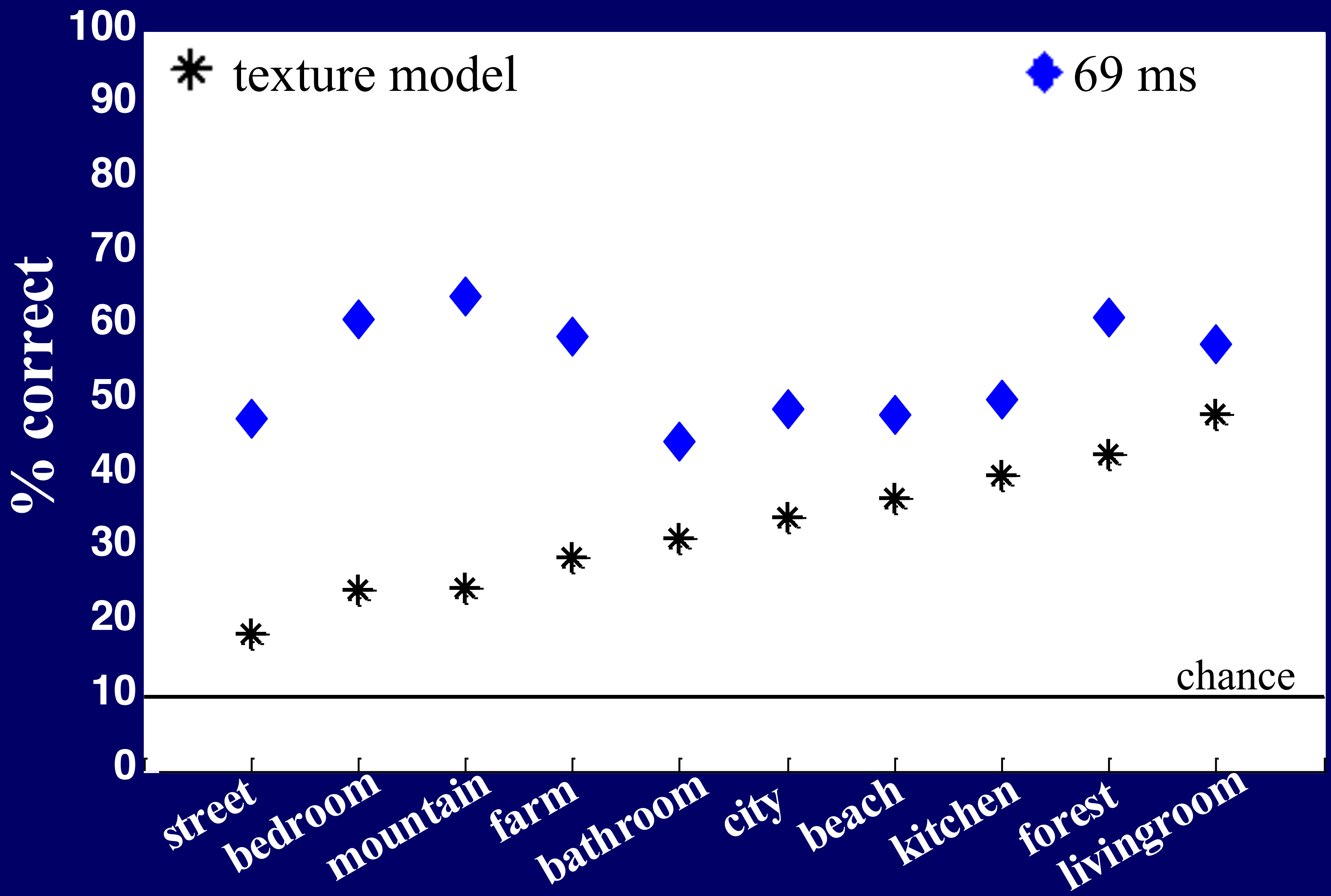
Discrimination of Basic Categories



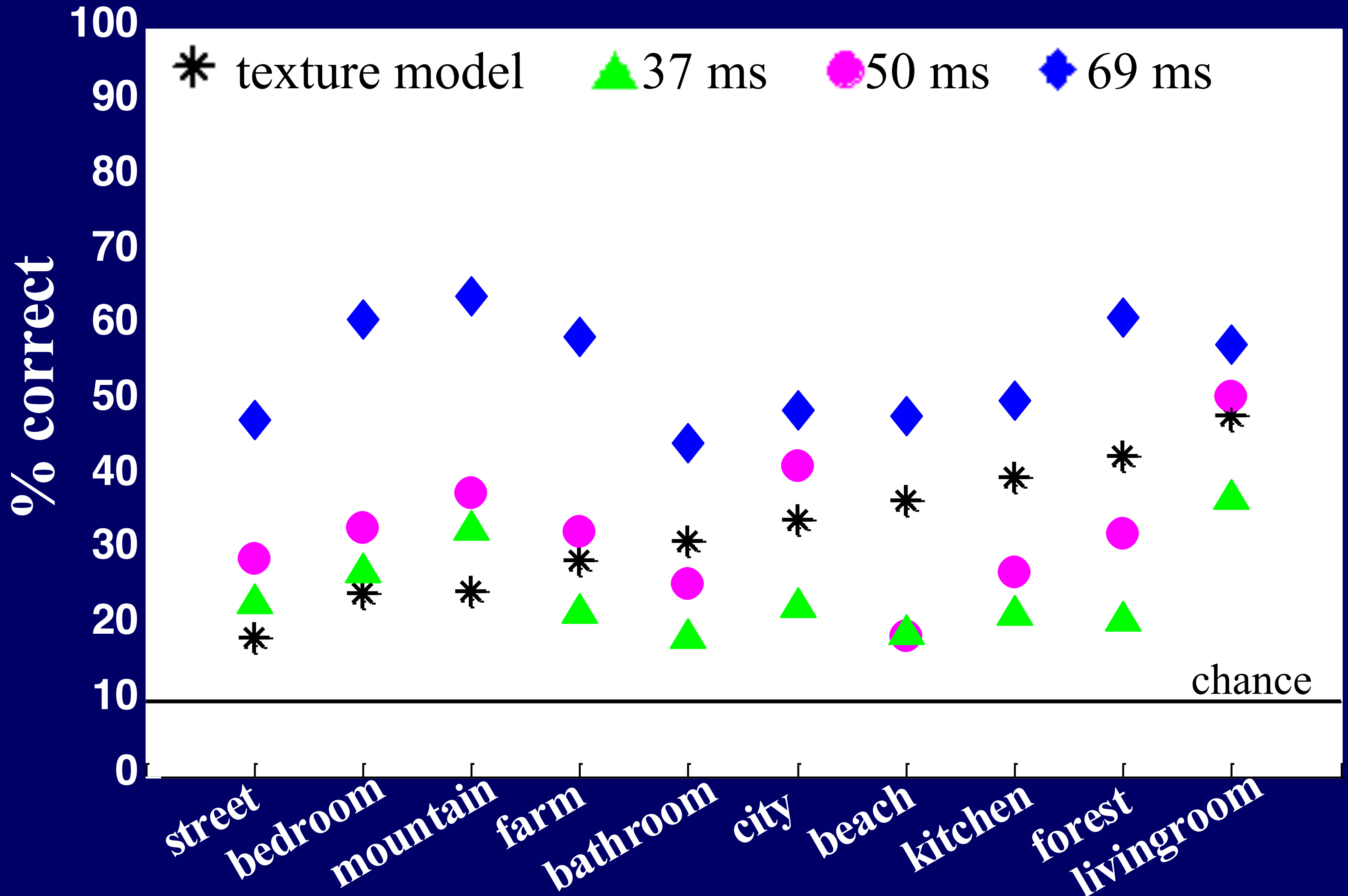
Discrimination of Basic Categories



Discrimination of Basic Categories



Discrimination of Basic Categories



Scene Recognition using Texture

Why these filters?

Wavelet-like receptive fields emerge from a network that learns sparse codes for natural images.

Bruno A. Olshausen[1] and David J. Field

$$E = -[\text{preserve information}] - \lambda[\text{sparseness of } a_i]\,, \tag{2}$$

where λ is a positive constant that determines the importance of the second term relative to the first. The first term measures how well the code describes the image, and we choose this to be the mean square of the error between the actual image and the reconstructed image:

$$[\text{preserve information}] = -\sum_{x,y}\left[I(x,y) - \sum_i a_i\,\phi_i(x,y)\right]^2 . \tag{3}$$

Learned filters

a.