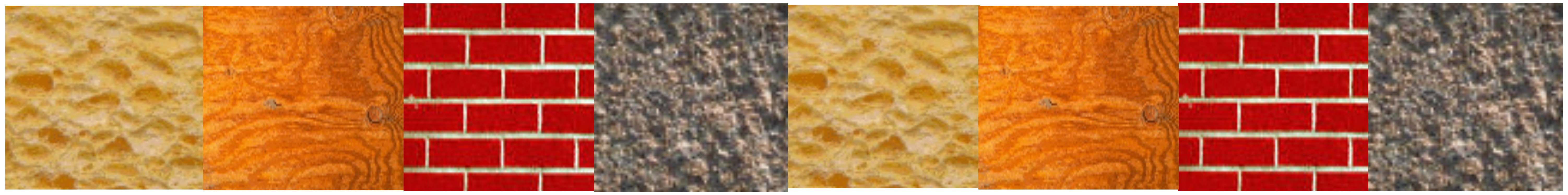
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 11: Texture (cont)

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today (**October 5, 2018**)

Topics:

— Texture Synthesis

— Texture **Analysis**

— **iClicker** quiz

Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 6.1-6.3, 3.1-3.3

— **Next** Lecture: N/A

Reminders:

— **Assignment 2**: Face Detection in a Scaled Representation is **February 8th**

— **Assignment 3**: Texture Synthesis will be out **February 8th**

Today's "**fun**" Example: Colour Constancy

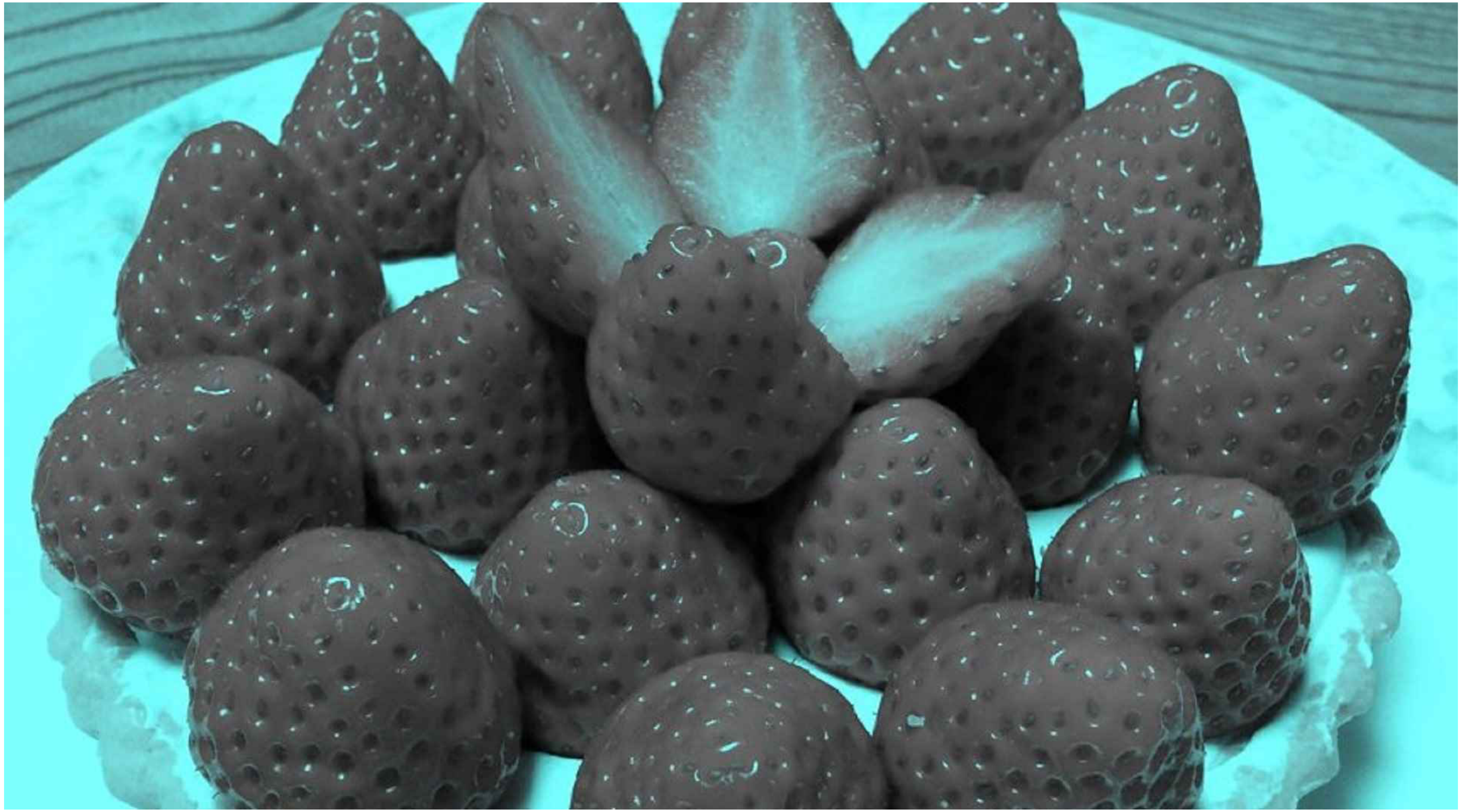


Image Credit: Akiyosha Kitoaka

Today’s “**fun**” Example: Colour Constancy

— Some people see a white and gold dress.

— Some people see a blue and black dress.

— Some people see one interpretation and then switch to the other

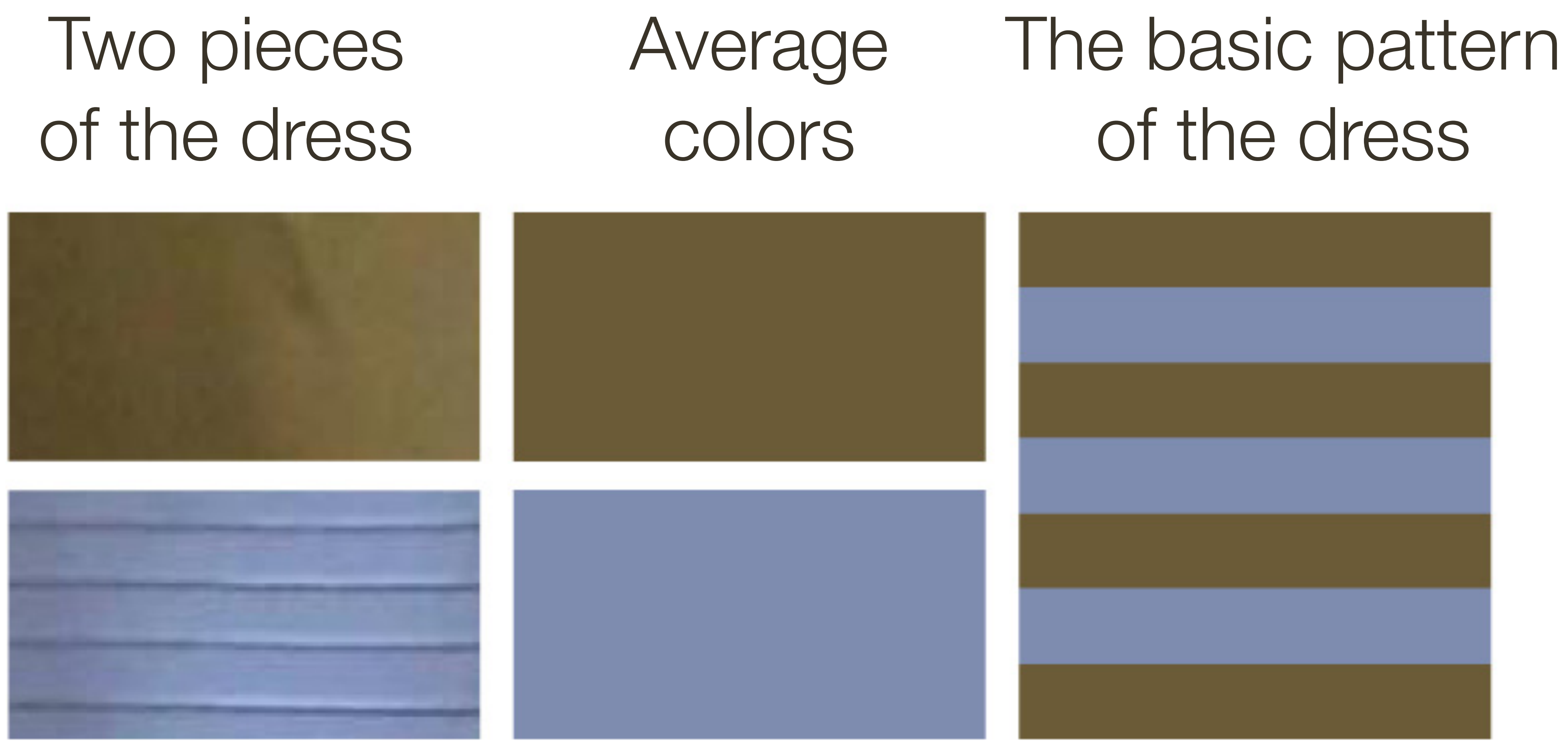
https://www.nytimes.com/interactive/2015/02/28/science/white-or-blue-dress.html

Today's "**fun**" Example: Colour Constancy

— Some people see a white and gold dress.

— Some people see a blue and black dress.

— Some people see one interpretation and then switch to the other



https://www.nytimes.com/interactive/2015/02/28/science/white-or-blue-dress.html

Today's "**fun**" Example: Colour Constancy

IS THE DRESS IN SHADOW?

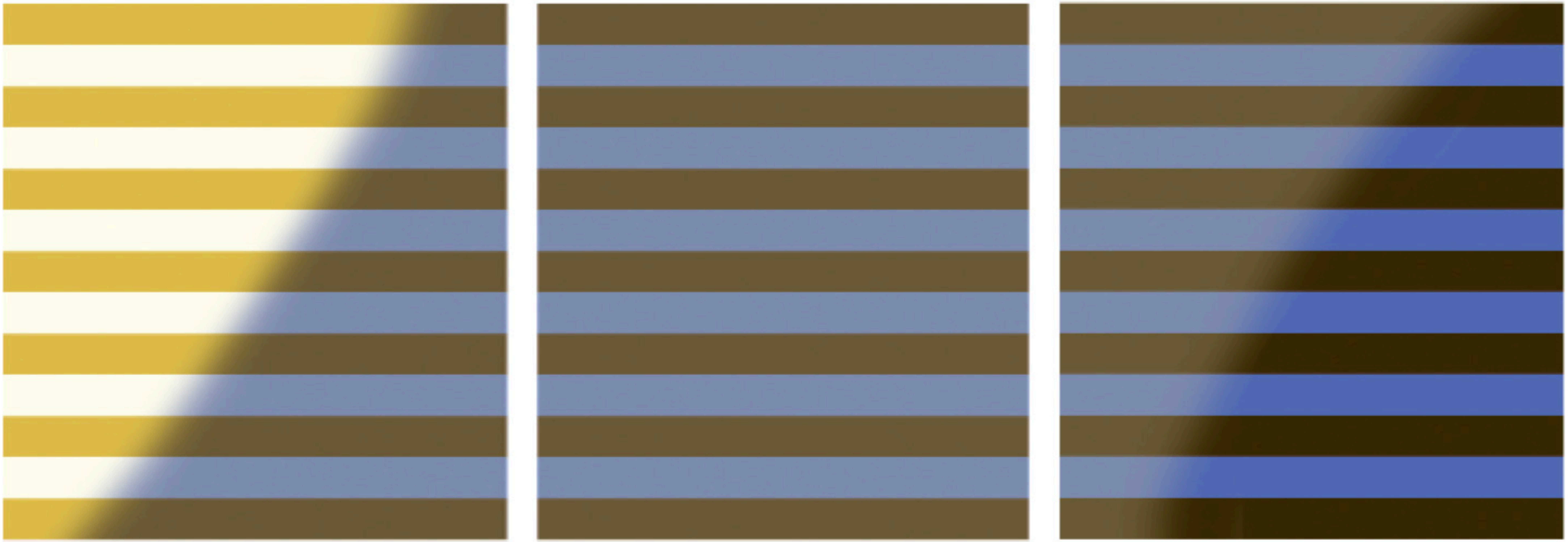
If you think the dress is in shadow, your brain may remove the blue cast and perceive the dress as being white and gold.

THE DRESS IN THE PHOTO

If the photograph showed more of the room, or if skin tones were visible, there might have been more clues about the ambient light.

IS THE DRESS IN BRIGHT LIGHT?

If you think the dress is being washed out by bright light, your brain may perceive the dress as a darker blue and black.



https://www.nytimes.com/interactive/2015/02/28/science/white-or-blue-dress.html

Today's "**fun**" Example: Colour Constancy

https://www.nytimes.com/interactive/2015/02/28/science/white-or-blue-dress.html

Lecture 10: Re-cap of Texture

(Functional) **Definition**:

Texture is detail in an image that is at a scale too small to be resolved into its constituent elements and at a scale large enough to be apparent in the spatial distribution of image measurements

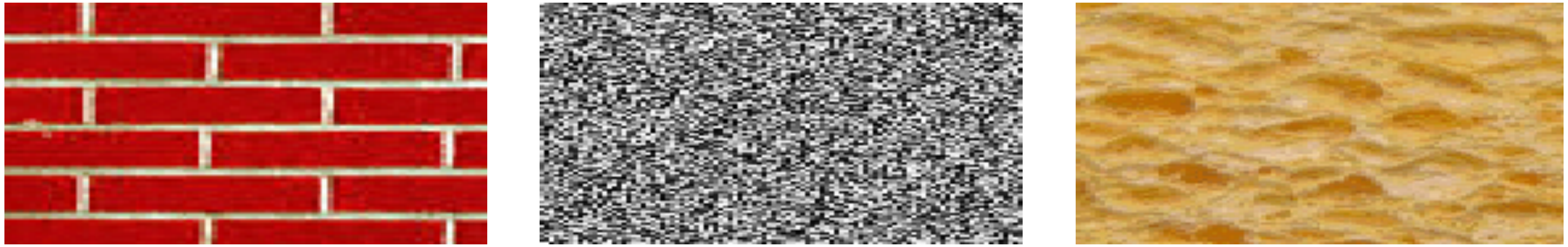


Figure Credit: Alexei Efros and Thomas Leung

Lecture 10: Re-cap of Texture

We will look at two main questions:

1. How do we represent texture?
   → Texture **analysis**

2. How do we generate new examples of a texture?
   → Texture **synthesis**

We begin with texture synthesis to set up **Assignment 3**

Lecture 10: Re-cap of Texture

Objective: Generate new examples of a texture We take a “data-driven" approach

Idea: Use an image of the texture as the source of a probability model

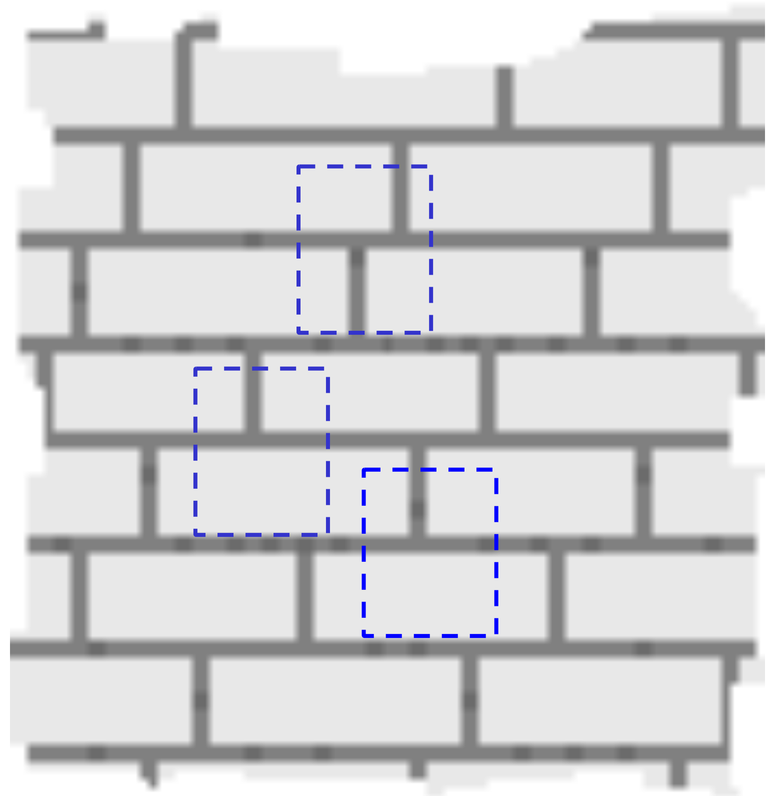
— Draw samples directly from the actual texture

— Can account for more types of structure

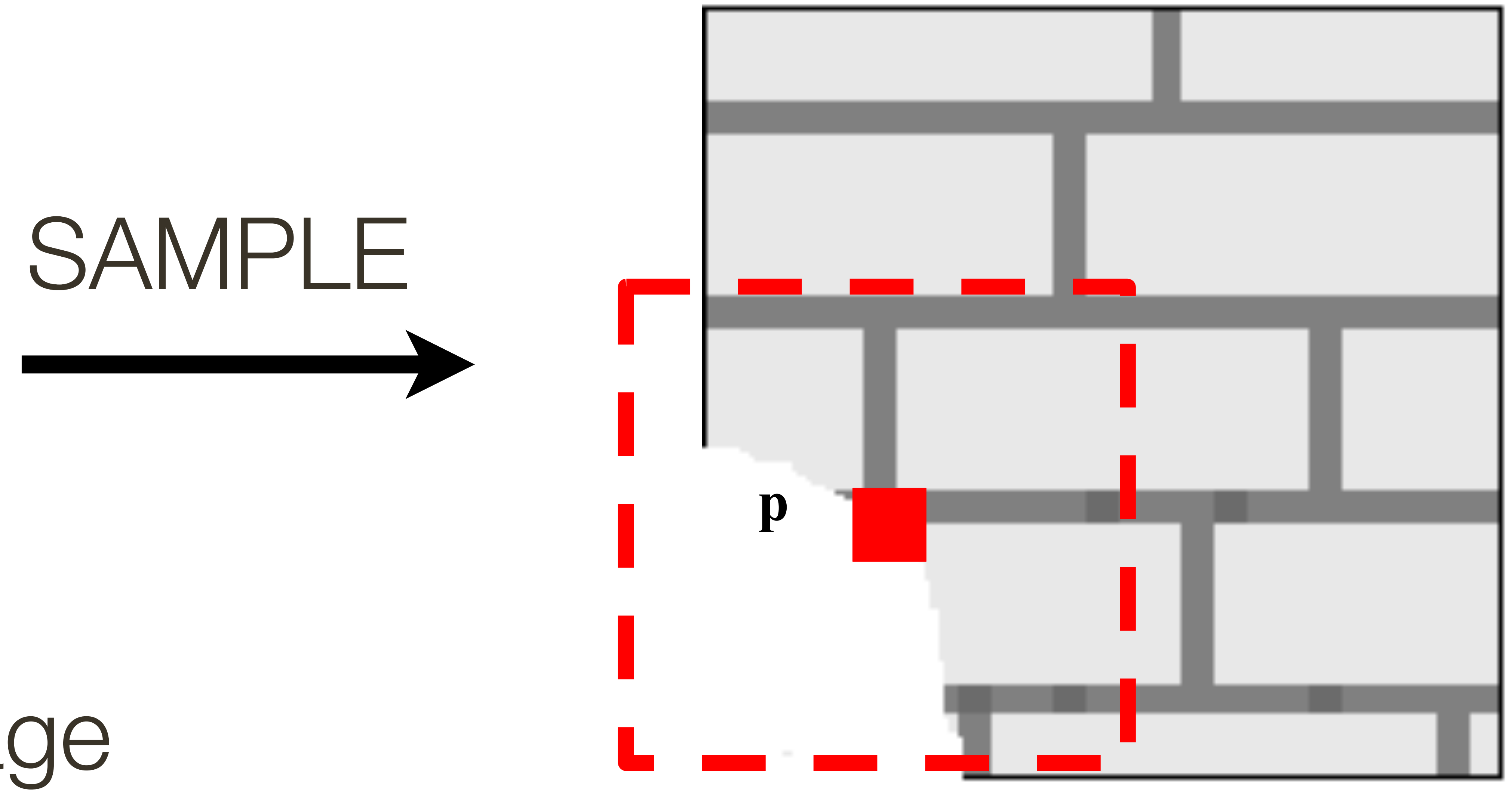
— Very simple to implement

— Success depends on choosing a correct “distance”

Lecture 10: Re-cap of **Efros** and Leung



Infinite sample image



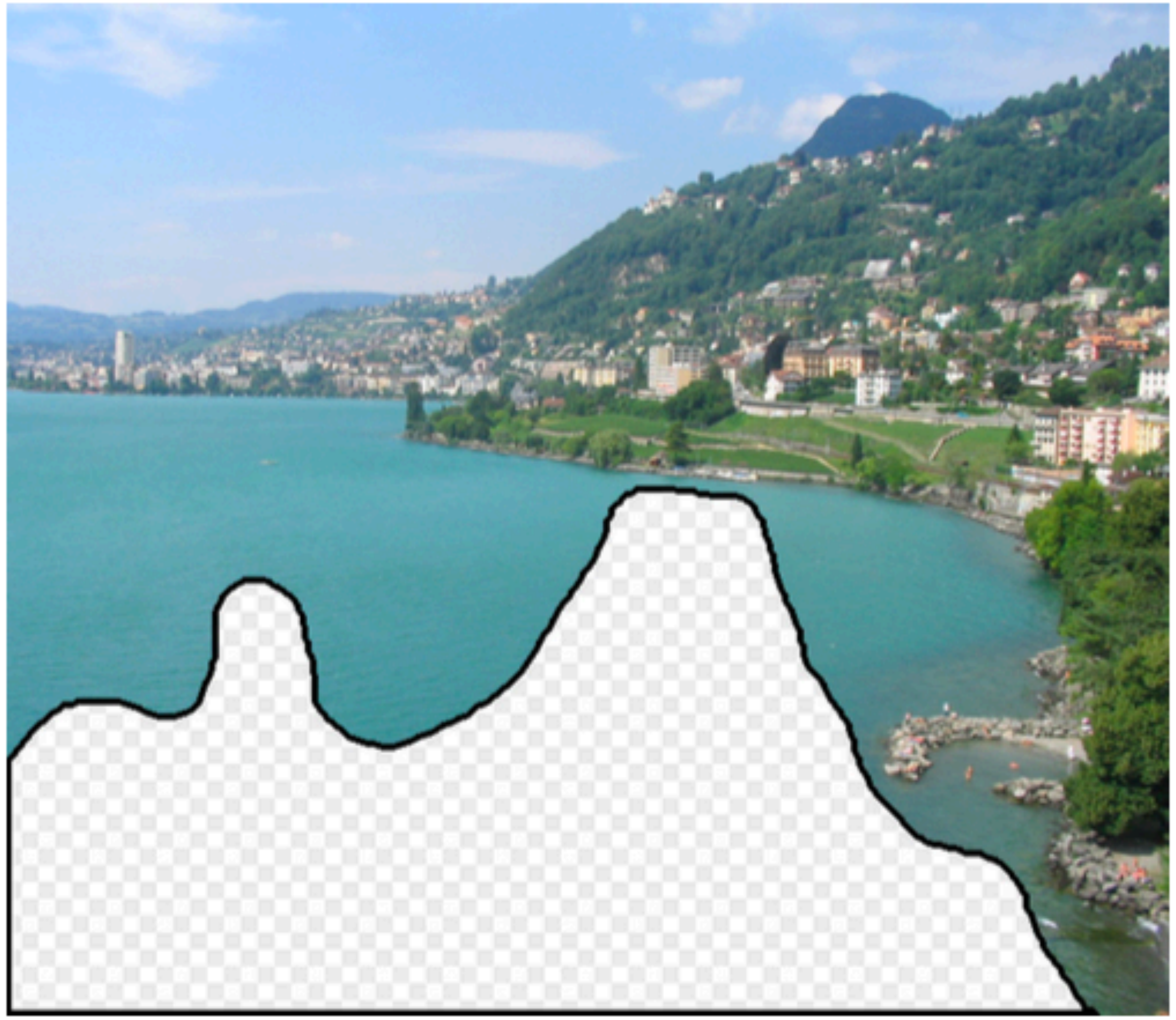
— What is **conditional** probability distribution of *p*, given the neighbourhood window?

— Directly search the input image for all such neighbourhoods to produce a **histogram** for *p*

— To **synthesize** *p*, pick one match at random

Lecture 10: Re-cap

Original Image



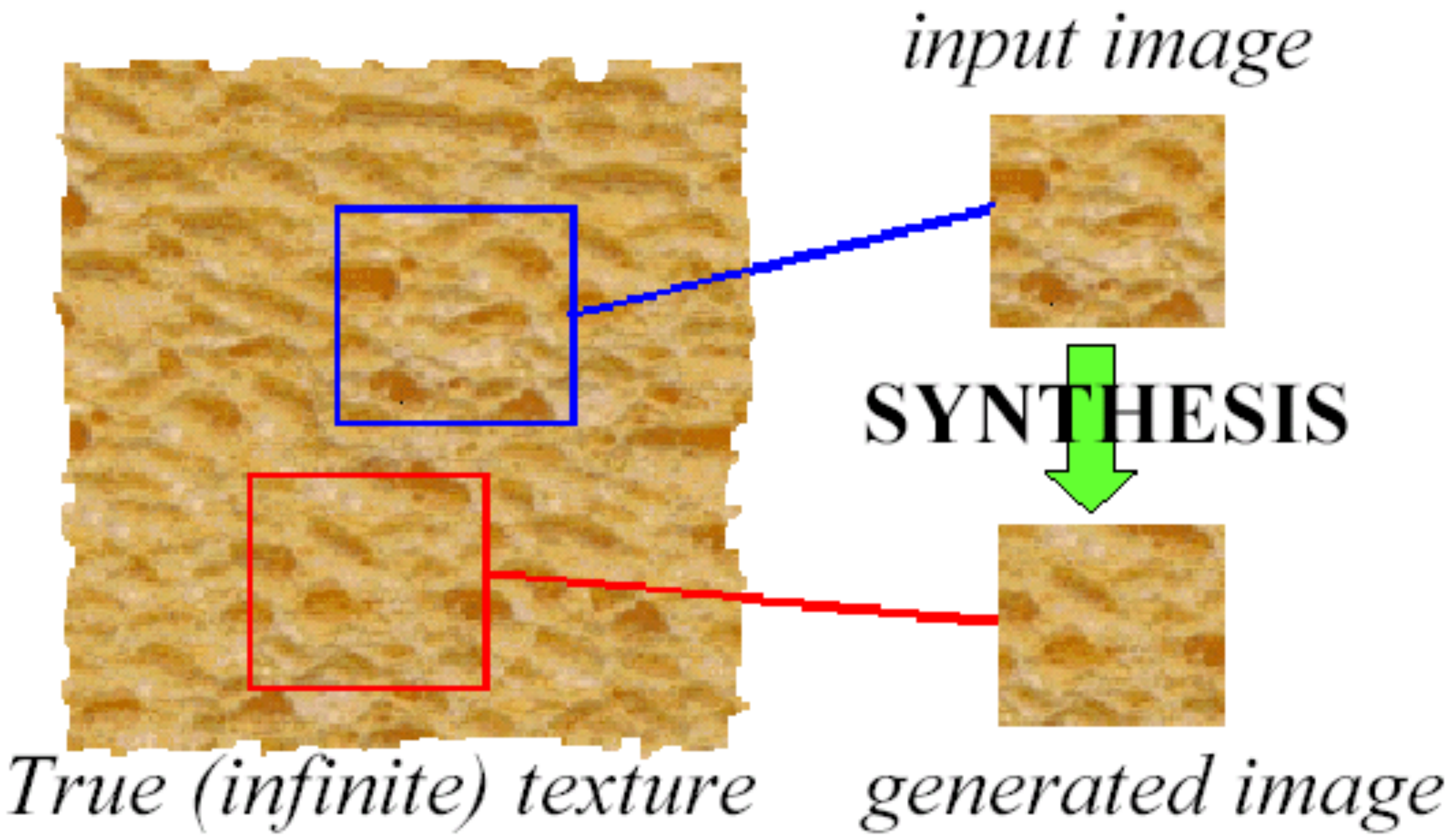
Input

Figure Credit: Hays and Efros 2007

Lecture 10: Re-cap

Figure Credit: Hays and Efros 2007

Goal of Texture **Synthesis**

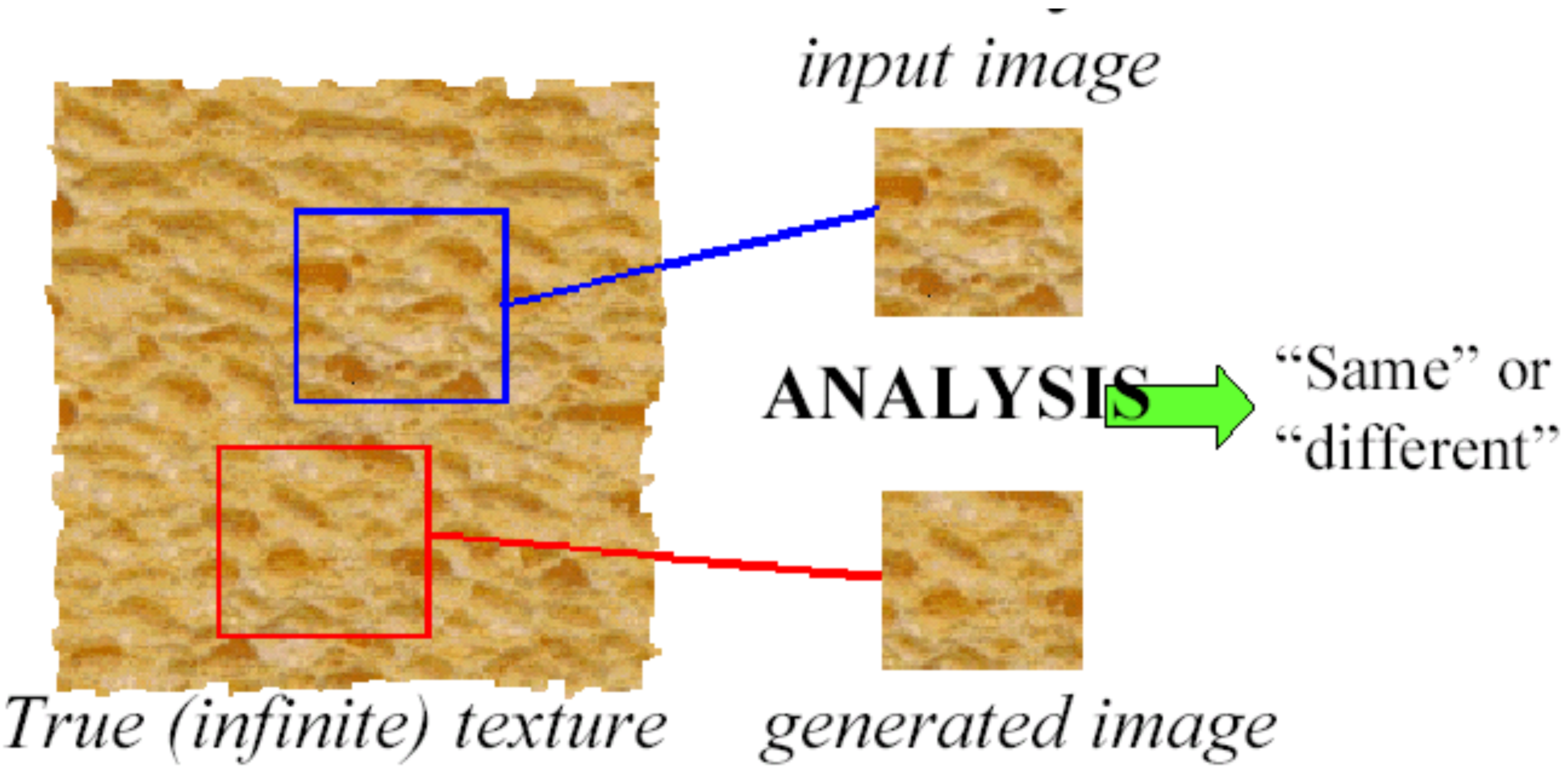


Given a finite sample of some texture, the goal is to synthesize other samples from that same texture

— The sample needs to be "large enough“

Credit: Bill Freeman

Goal of Texture **Analysis**



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

Credit: Bill Freeman

Definition of Texture (Re-Cap)

Recall that texture is easy to recognize but hard to define

— A functional definition was presented last class

We need representations that differ in ways that are easy to observe when two textures are significantly different.

Recall that textures can often be thought of as patterns composed of repeated instances of one (or more) identifiable elements, called **textons**

— e.g. bricks in a wall, spots on a cheetah

Texture **Segmentation**

Question: Is texture a property of a point or a property of a region?

Texture **Segmentation**

Question: Is texture a property of a point or a property of a region?

Answer: We need a region to have a texture.

Texture **Segmentation**

Question: Is texture a property of a point or a property of a region?

Answer: We need a region to have a texture.

There is a “chicken–and–egg” problem. Texture segmentation can be done by detecting boundaries between regions of the same (or similar) texture. Texture boundaries can be detected using standard edge detection techniques applied to the texture measures determined at each point

Texture **Segmentation**

Question: Is texture a property of a point or a property of a region?

Answer: We need a region to have a texture.

There is a “chicken–and–egg” problem. Texture segmentation can be done by detecting boundaries between regions of the same (or similar) texture. Texture boundaries can be detected using standard edge detection techniques applied to the texture measures determined at each point

We compromise! Typically one uses a local window to estimate texture properties and assigns those texture properties as point properties of the window’s center row and column

Texture **Representation**

Question: How many degrees of freedom are there to texture?

Texture **Representation**

Question: How many degrees of freedom are there to texture?

(**Mathematical**) Answer: Infinitely many

(**Perceptual Psychology**) Answer: There are perceptual constraints. But, there is no clear notion of a "texture channel" like, for example, there is for an RGB colour channel

Texture **Representation**

Observation: Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

Idea: Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

Texture **Representation**

Observation: Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

Idea: Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

Question: What filters should we use?

Answer: Human vision suggests spots and oriented edge filters at a variety of different orientations and scales

Texture **Representation**

Observation: Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

Idea: Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

Question: What filters should we use?

Answer: Human vision suggests spots and oriented edge filters at a variety of different orientations and scales

Question: How do we “summarize”?

Answer: Compute the mean or maximum of each filter response over the region
— Other statistics can also be useful

Texture **Representation**

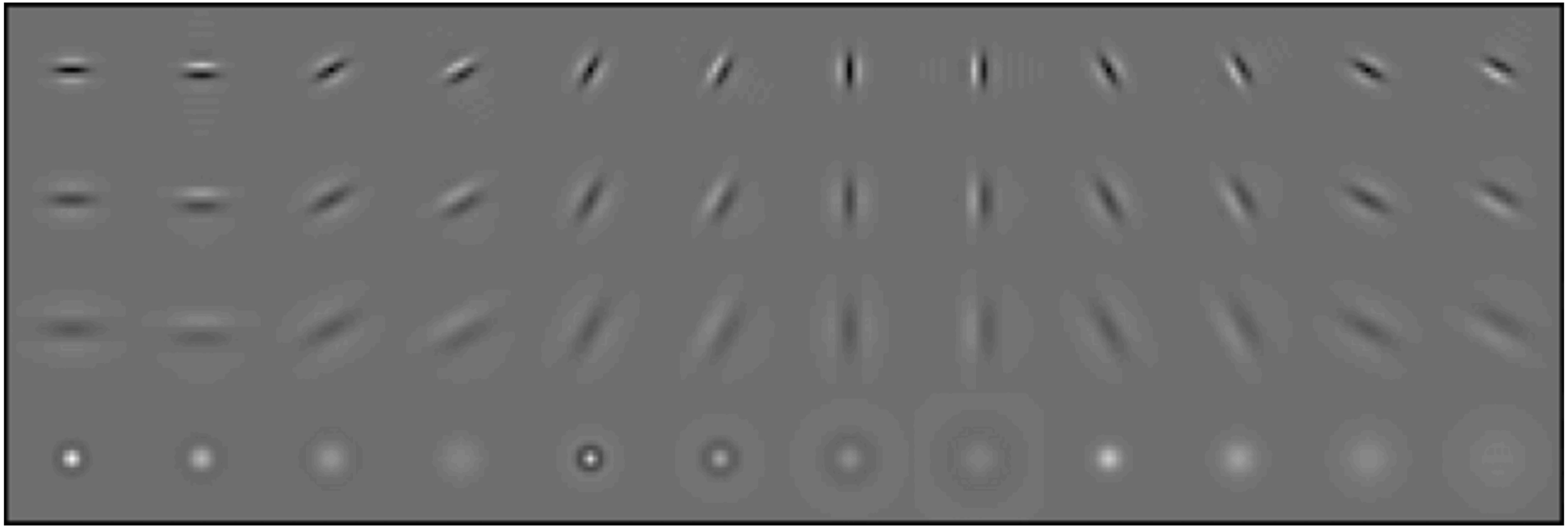
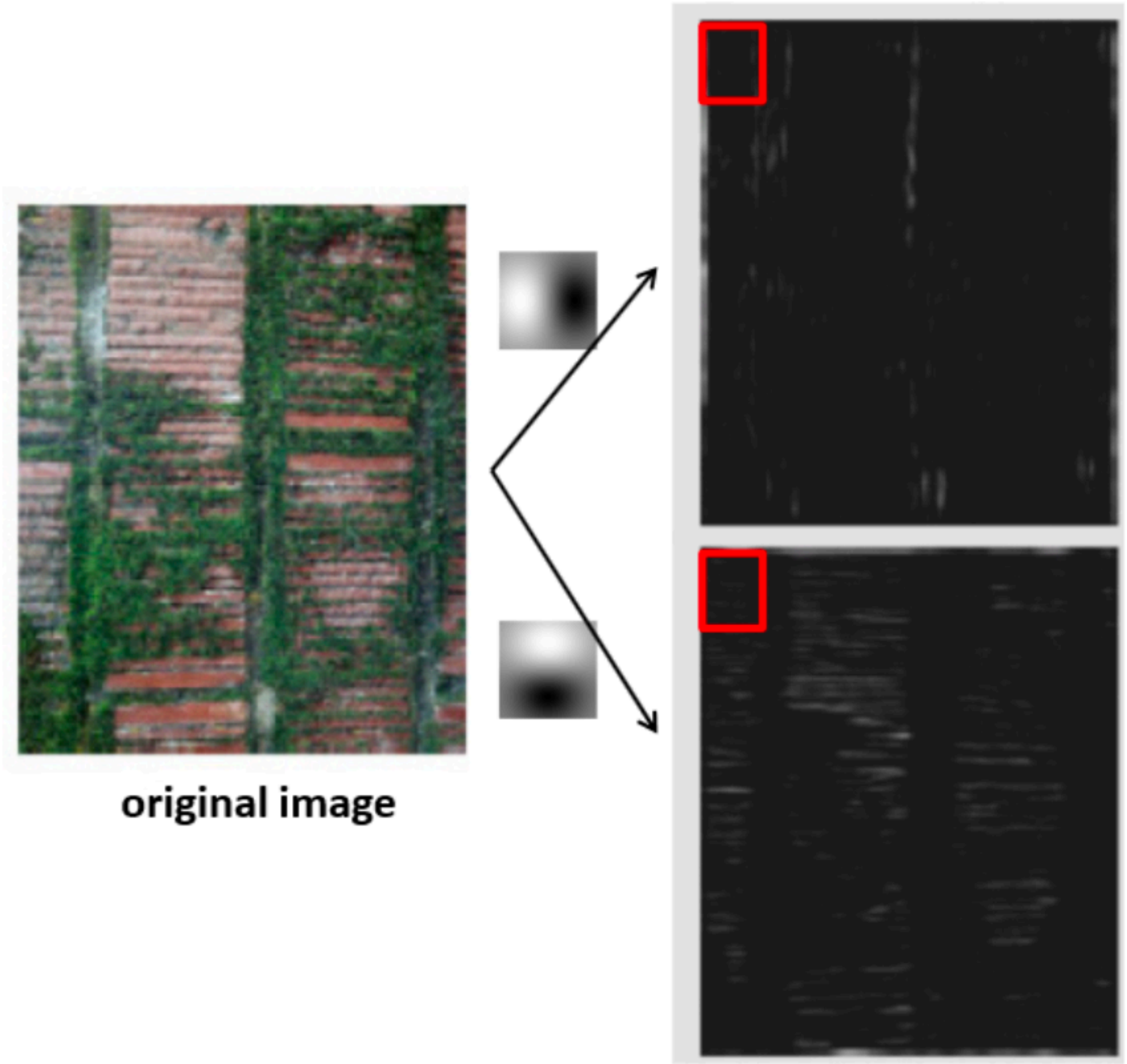


Figure Credit: Leung and Malik, 2001

Texture **Representation**



original image

derivative filter responses, squared

	mean d/dx value	mean d/dy value
Win. #1	4	10

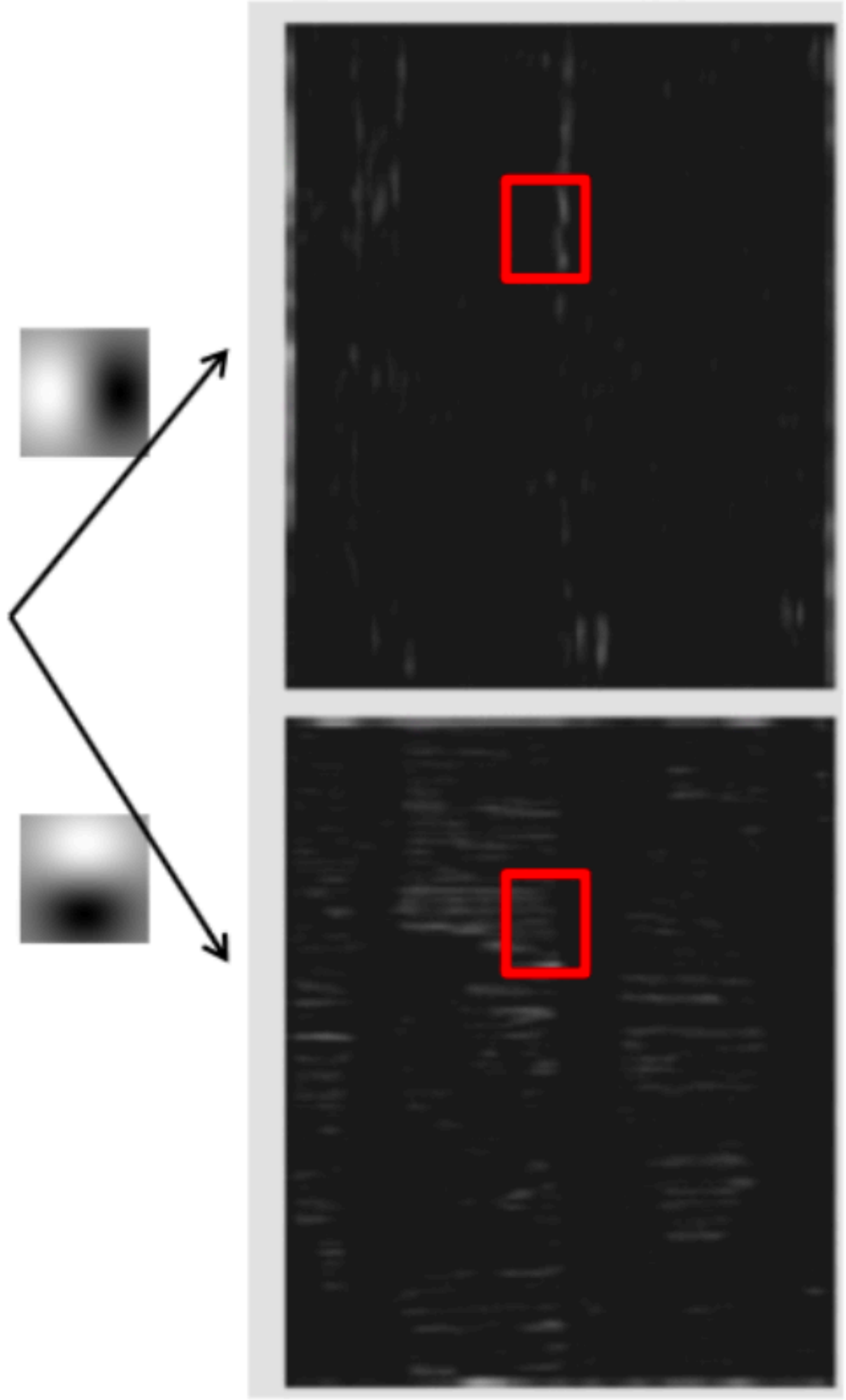
⋮

statistics to summarize patterns in small windows

Slide Credit: Trevor Darrell

Texture **Representation**

original image



derivative filter responses, squared

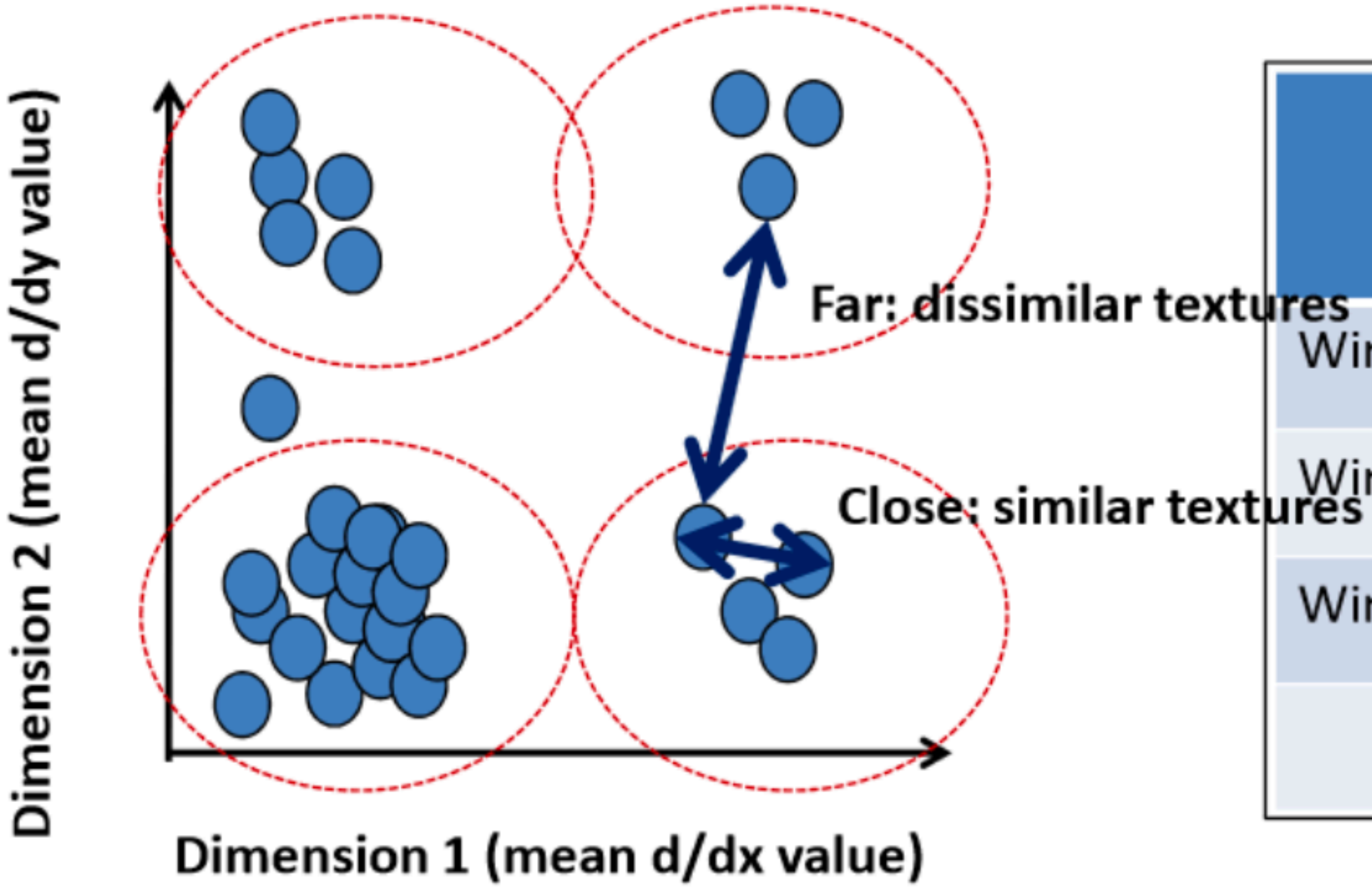
	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

statistics to summarize patterns in small windows

Slide Credit: Trevor Darrell

Texture **Representation**



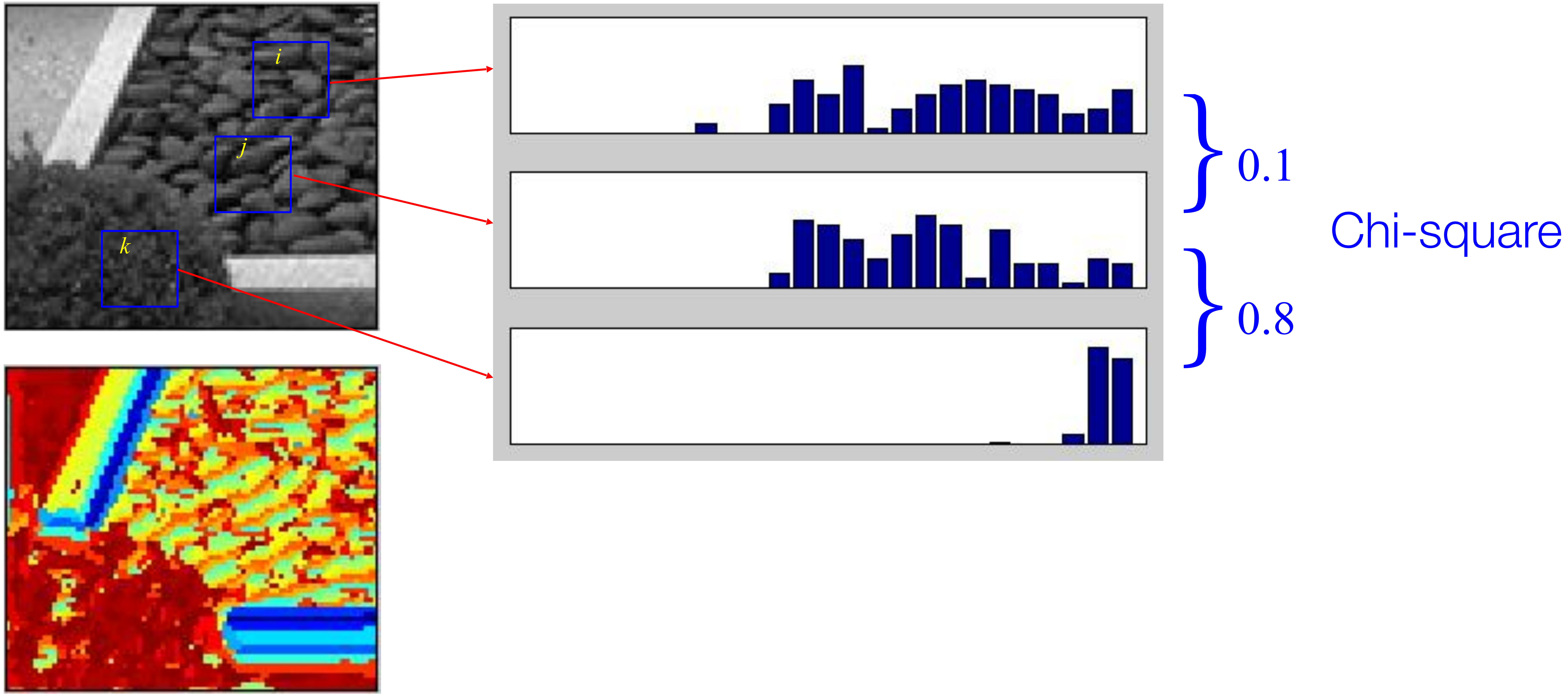
	mean d/dx value	mean d/dy value
Win. #1	4	10
Win.#2	18	7
⋮		
Win.#9	20	20

⋮

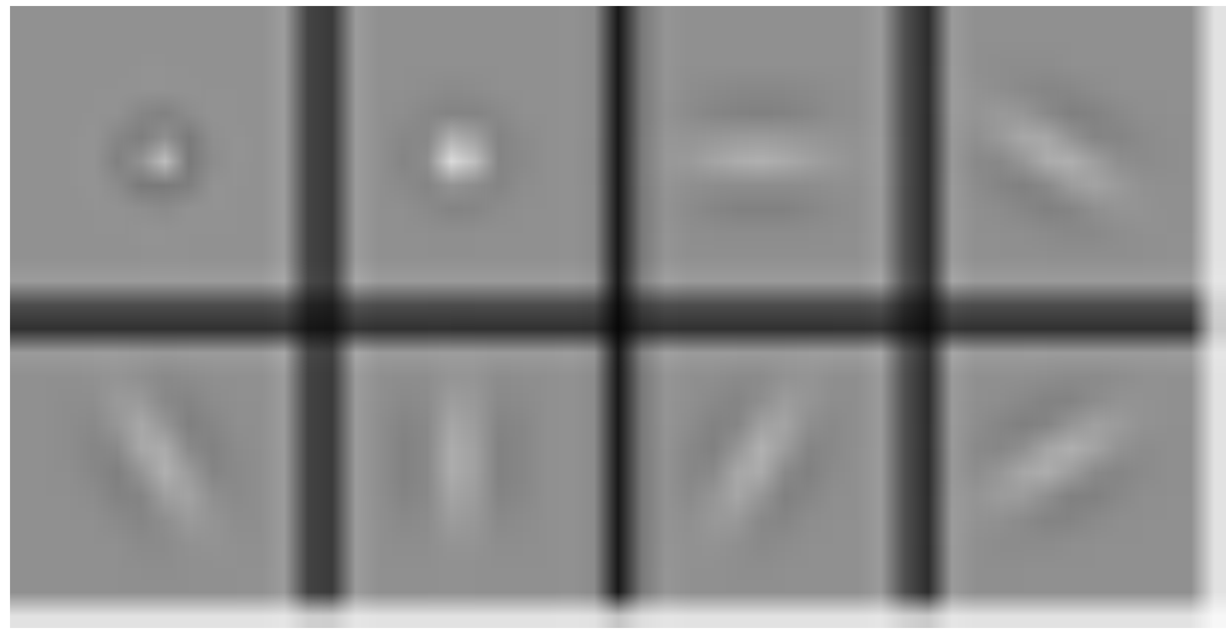
statistics to summarize patterns in small windows

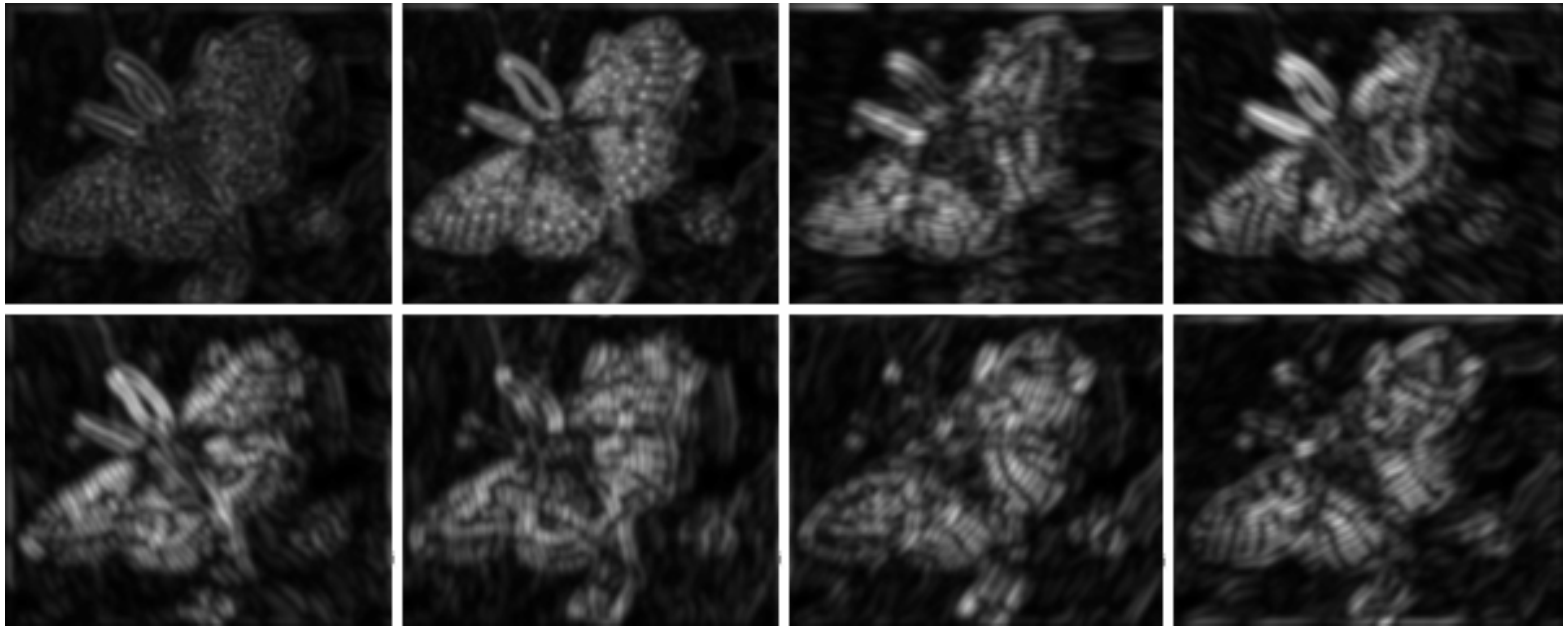
Slide Credit: Trevor Darrell

Texture **Representation**



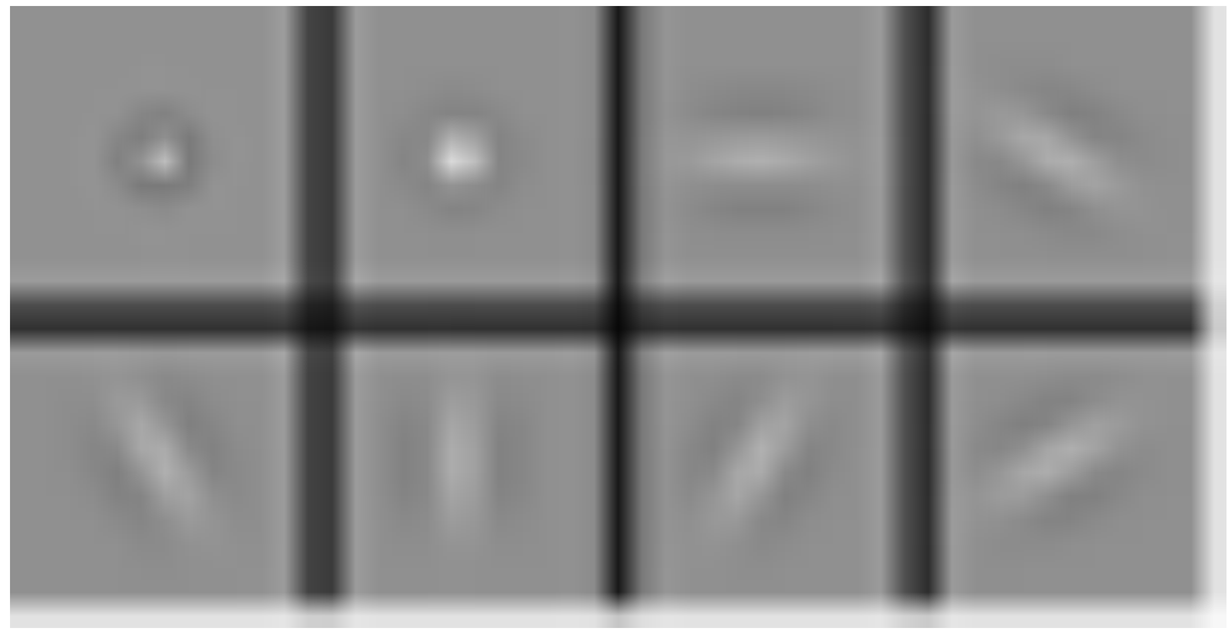
Spots and Bars (Fine Scale)

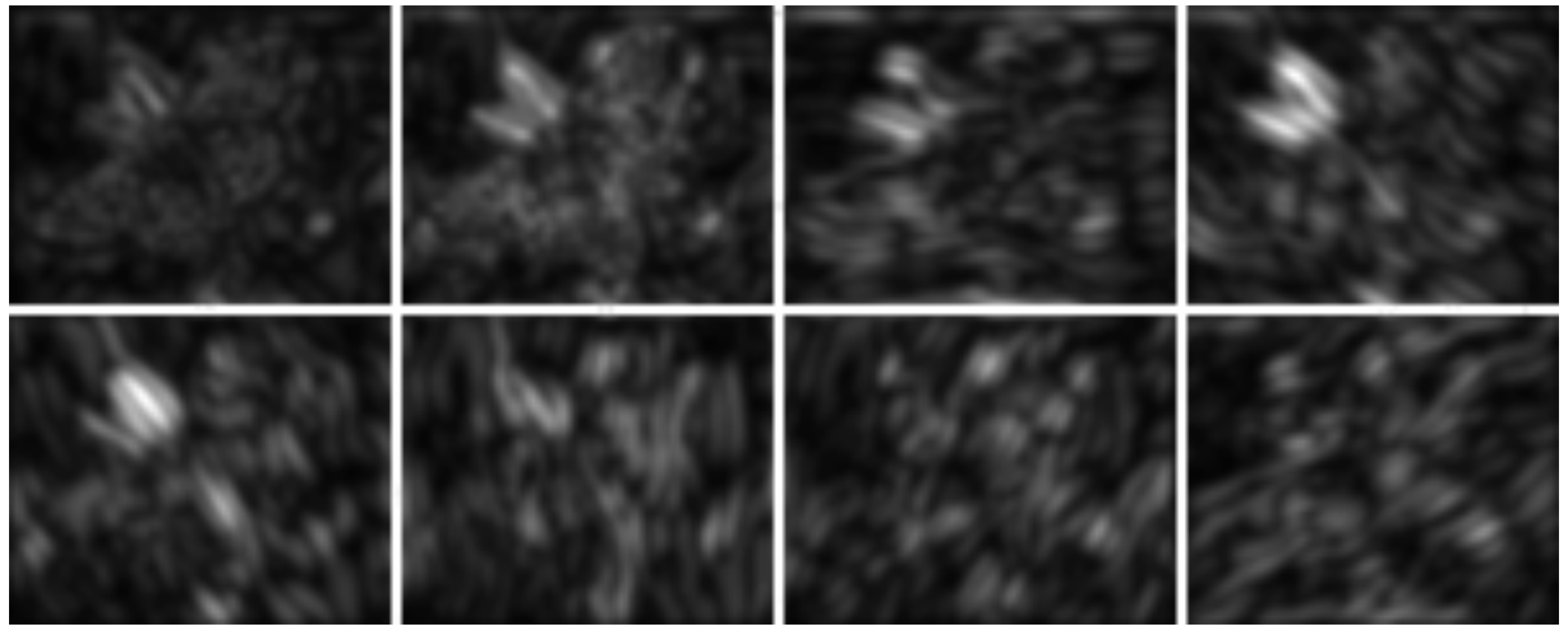




Forsyth & Ponce (1st ed.) Figures 9.3–9.4

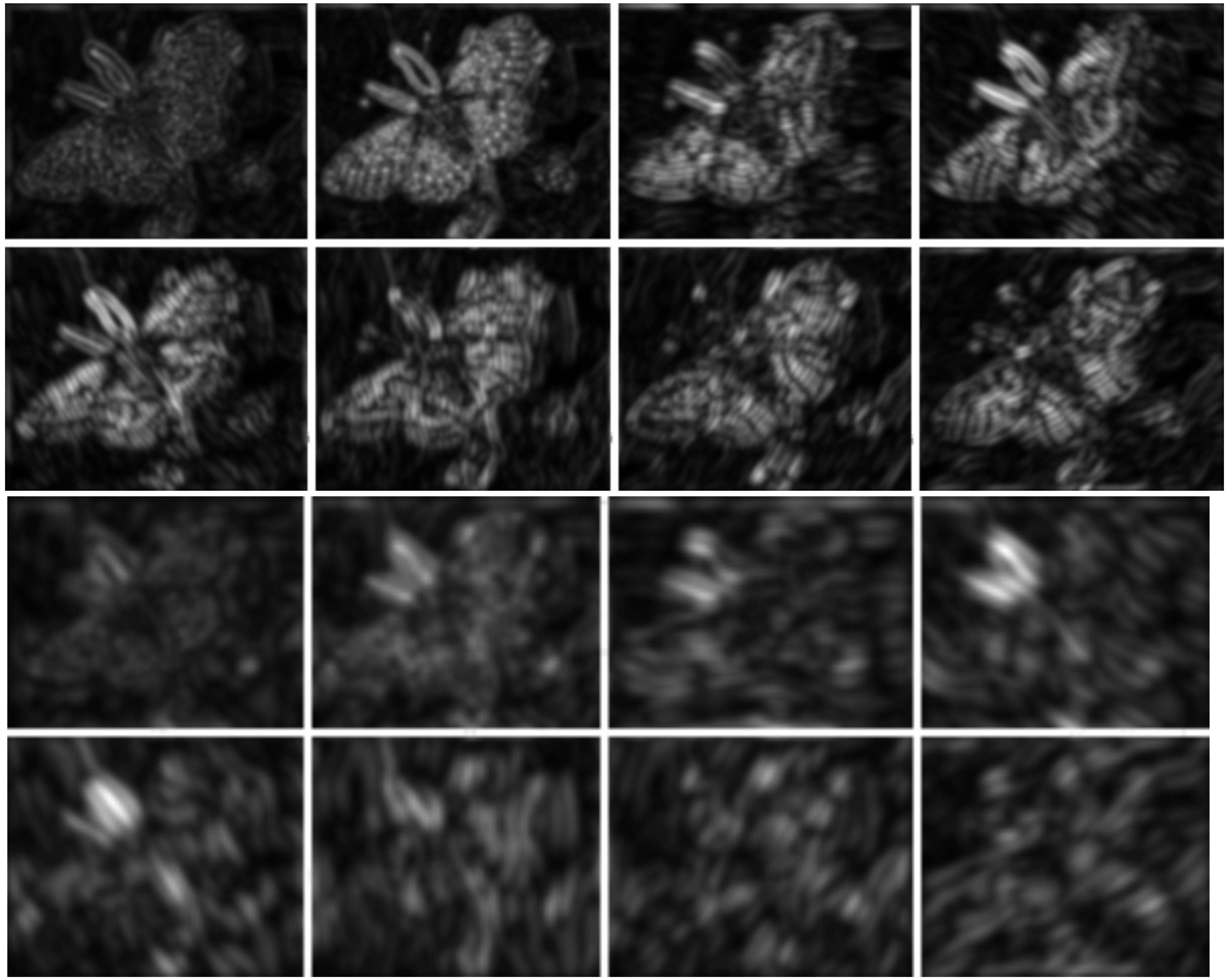
Spots and Bars (Coarse Scale)





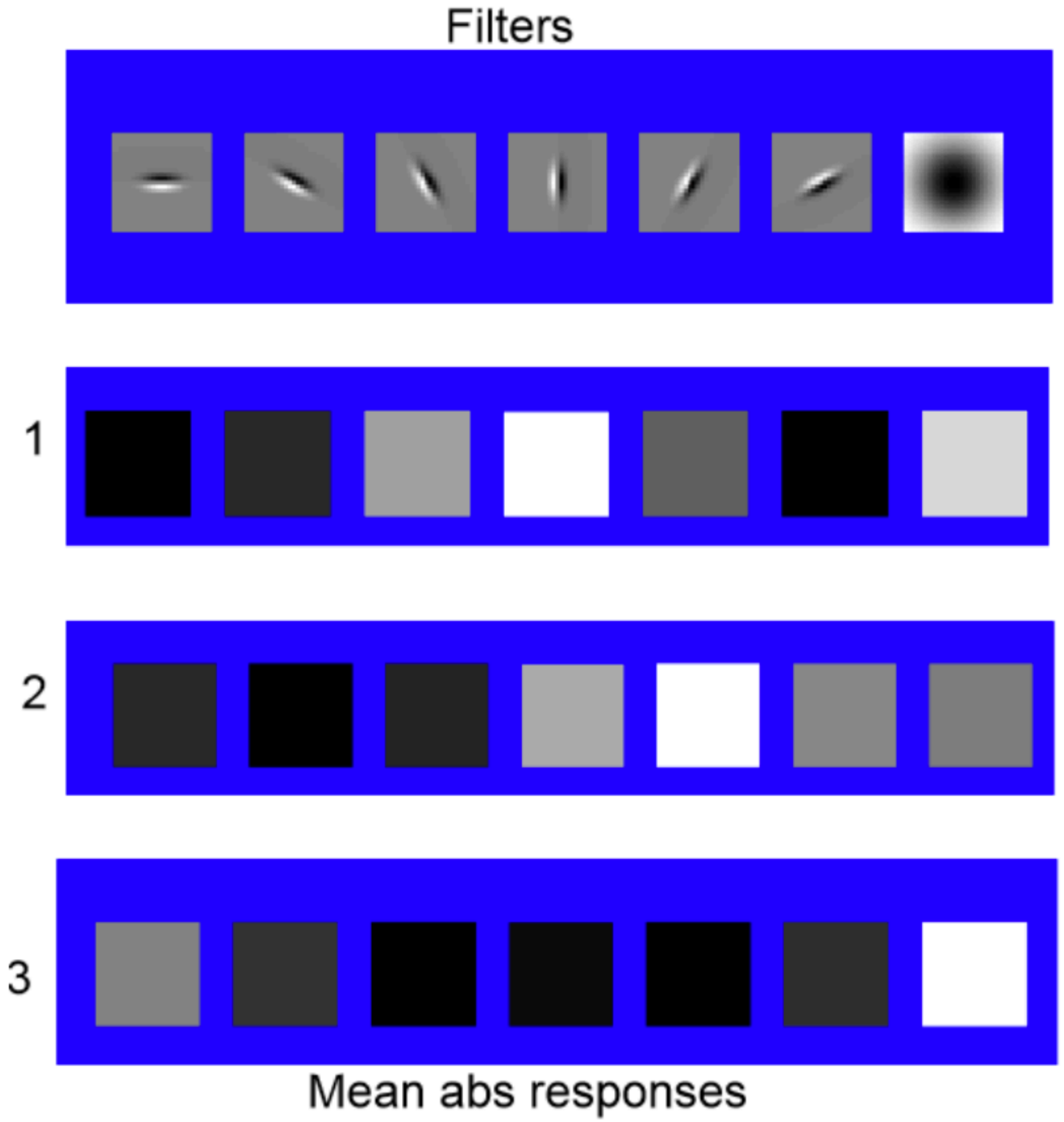
Forsyth & Ponce (1st ed.) Figures 9.3 and 9.5

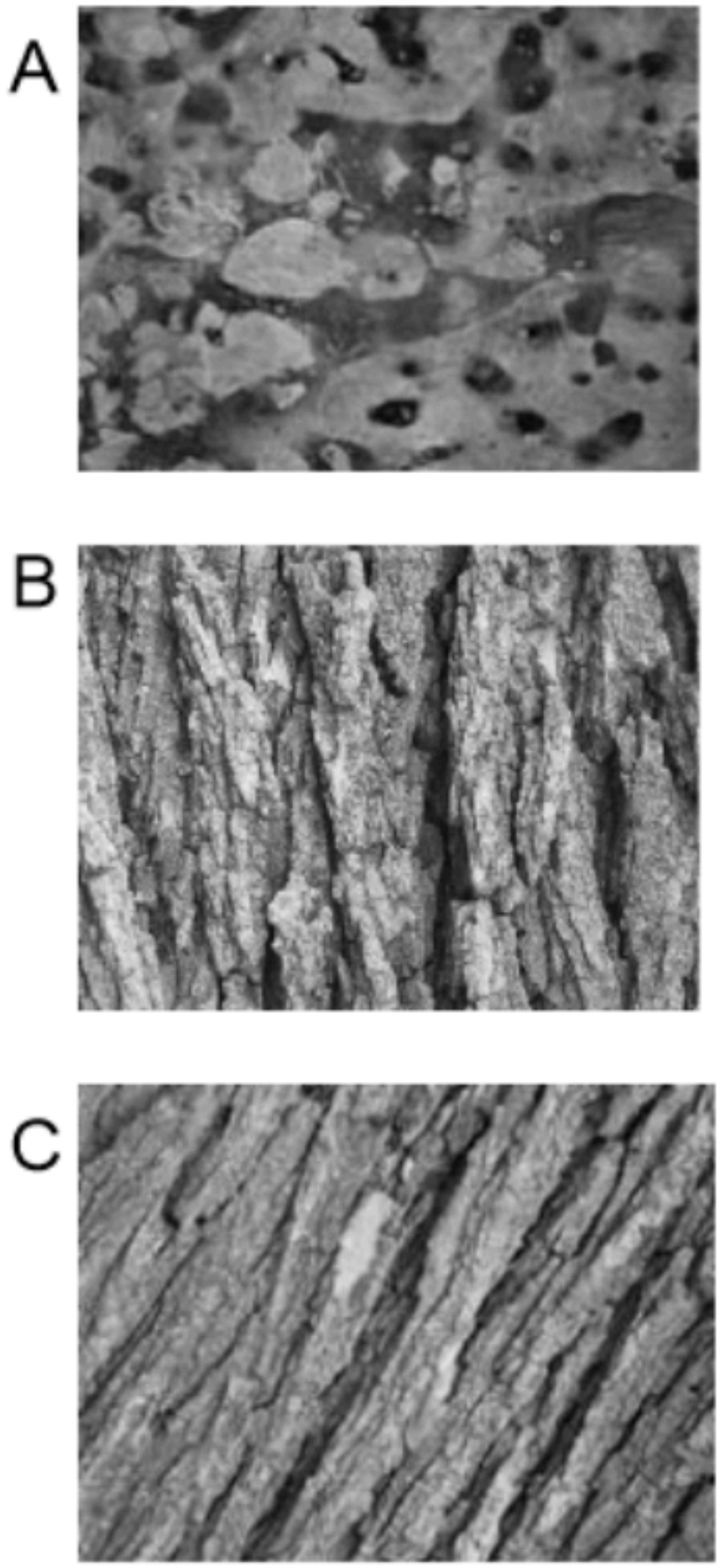
Comparison of Results



Forsyth & Ponce (1st ed.) Figures 9.4–9.5

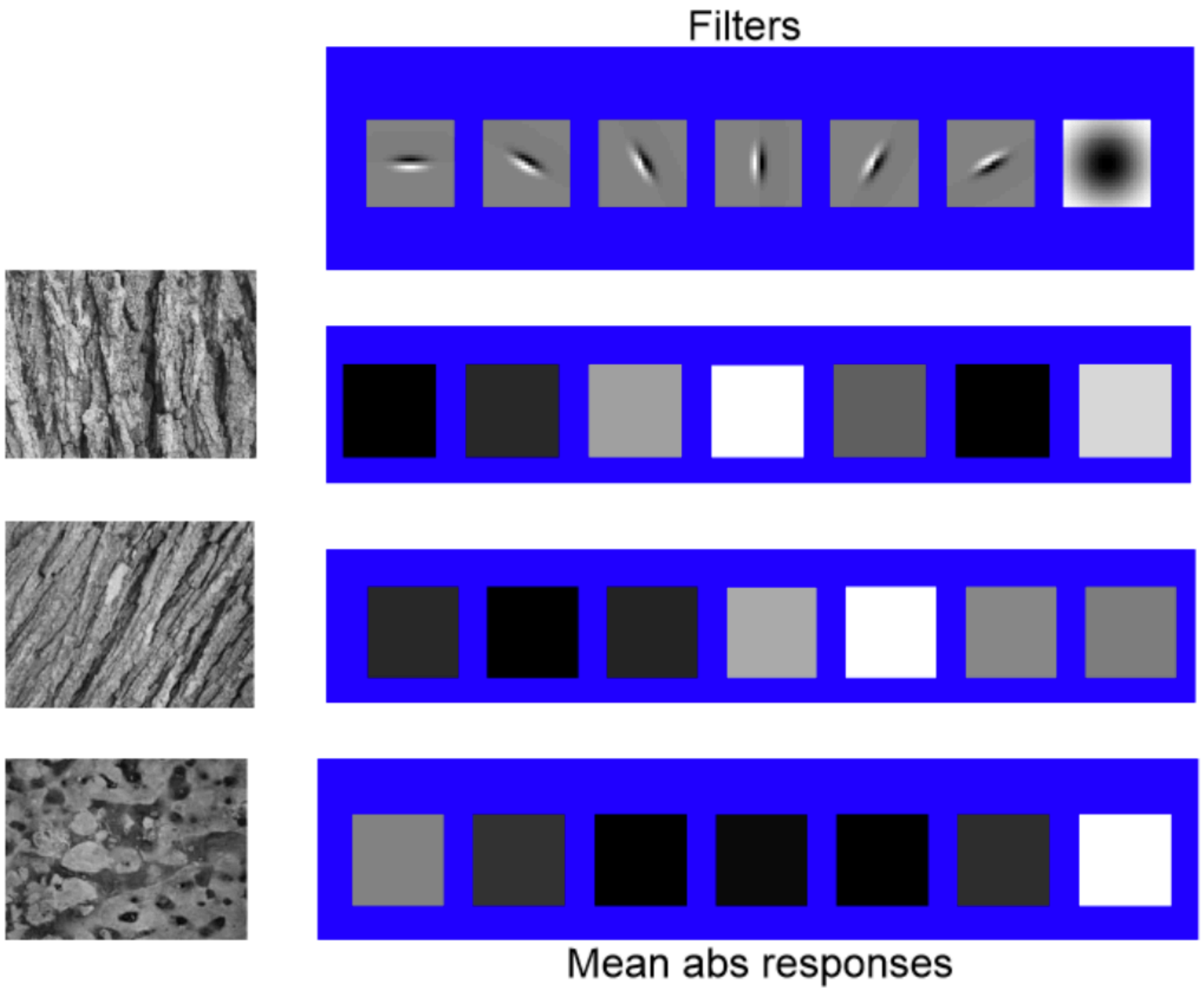
A Short **Exercise**: Match the texture to the response





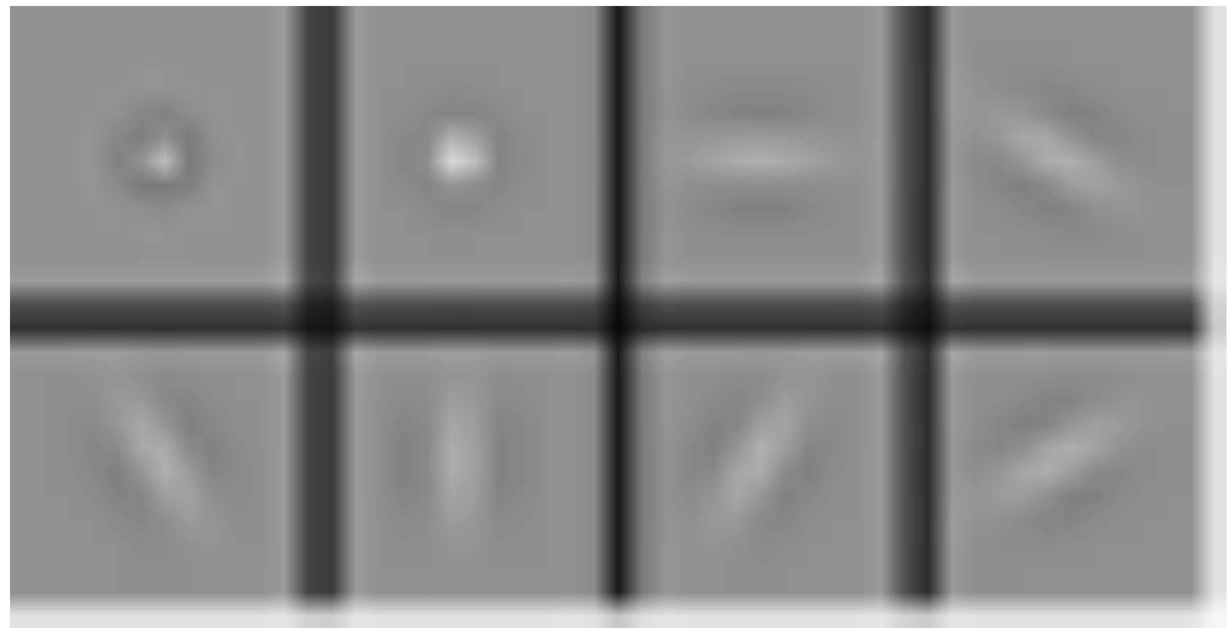
Slide Credit: James Hays

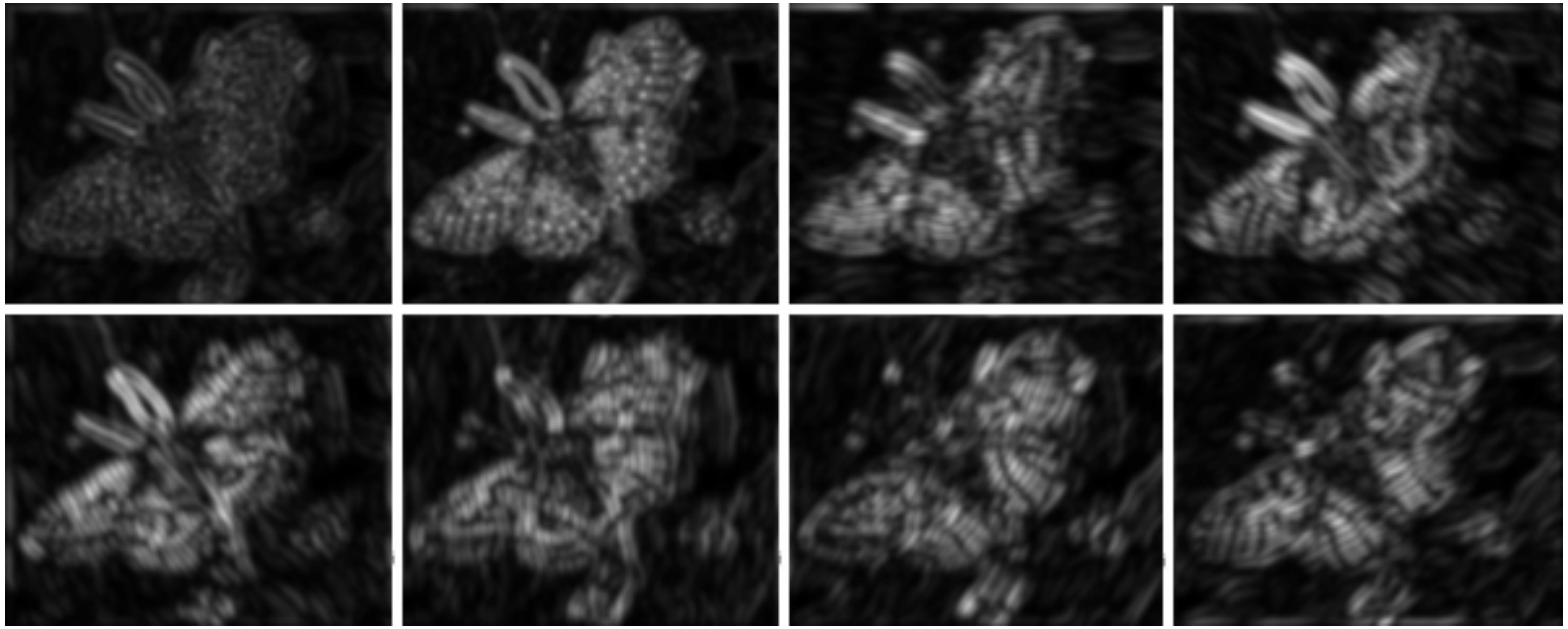
A Short **Exercise**: Match the texture to the response



Slide Credit: James Hays

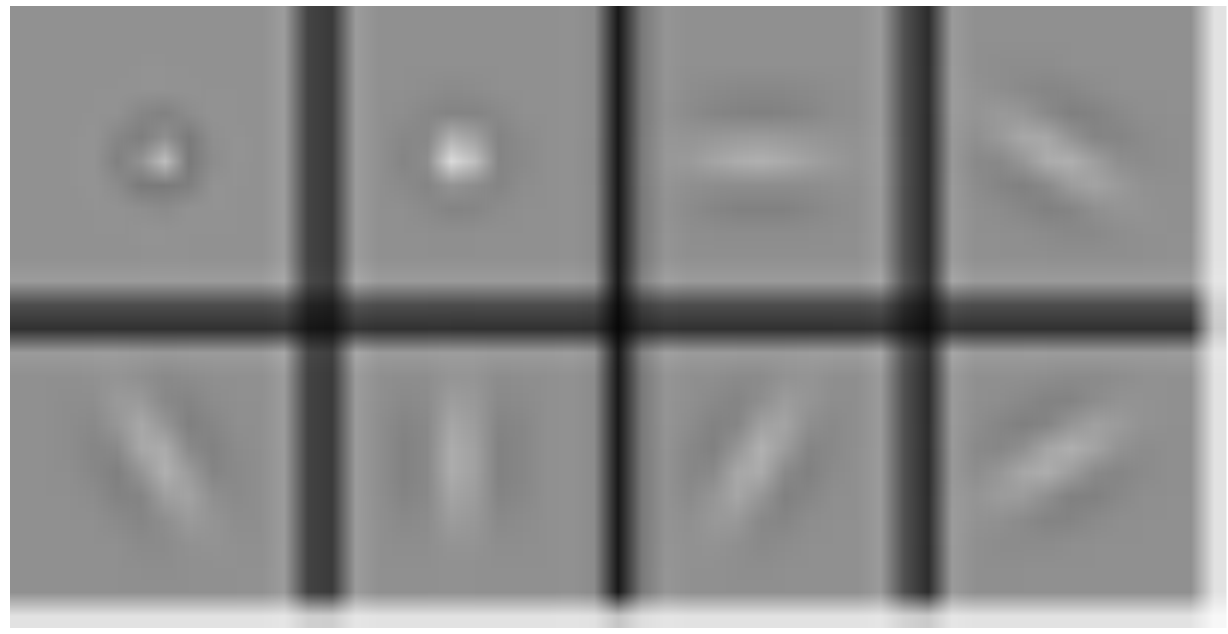
Spots and Bars (Fine Scale)

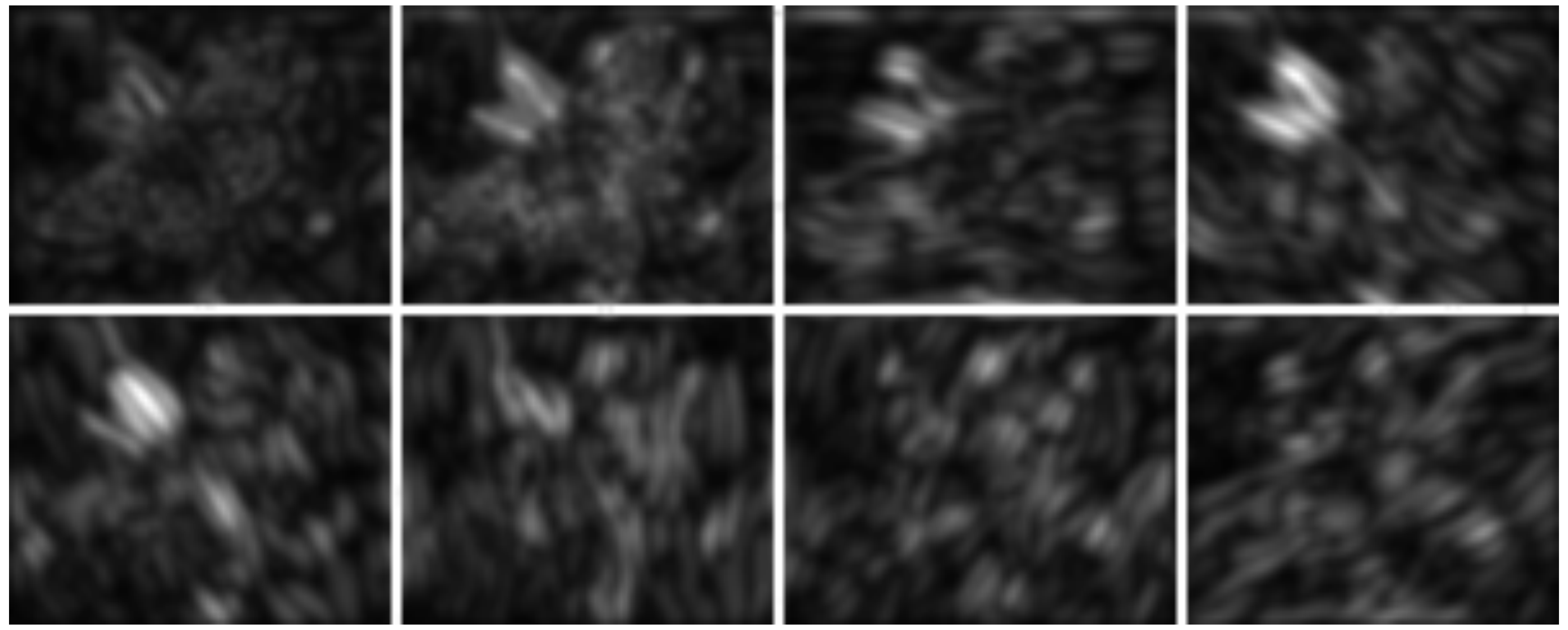




Forsyth & Ponce (1st ed.) Figures 9.3–9.4

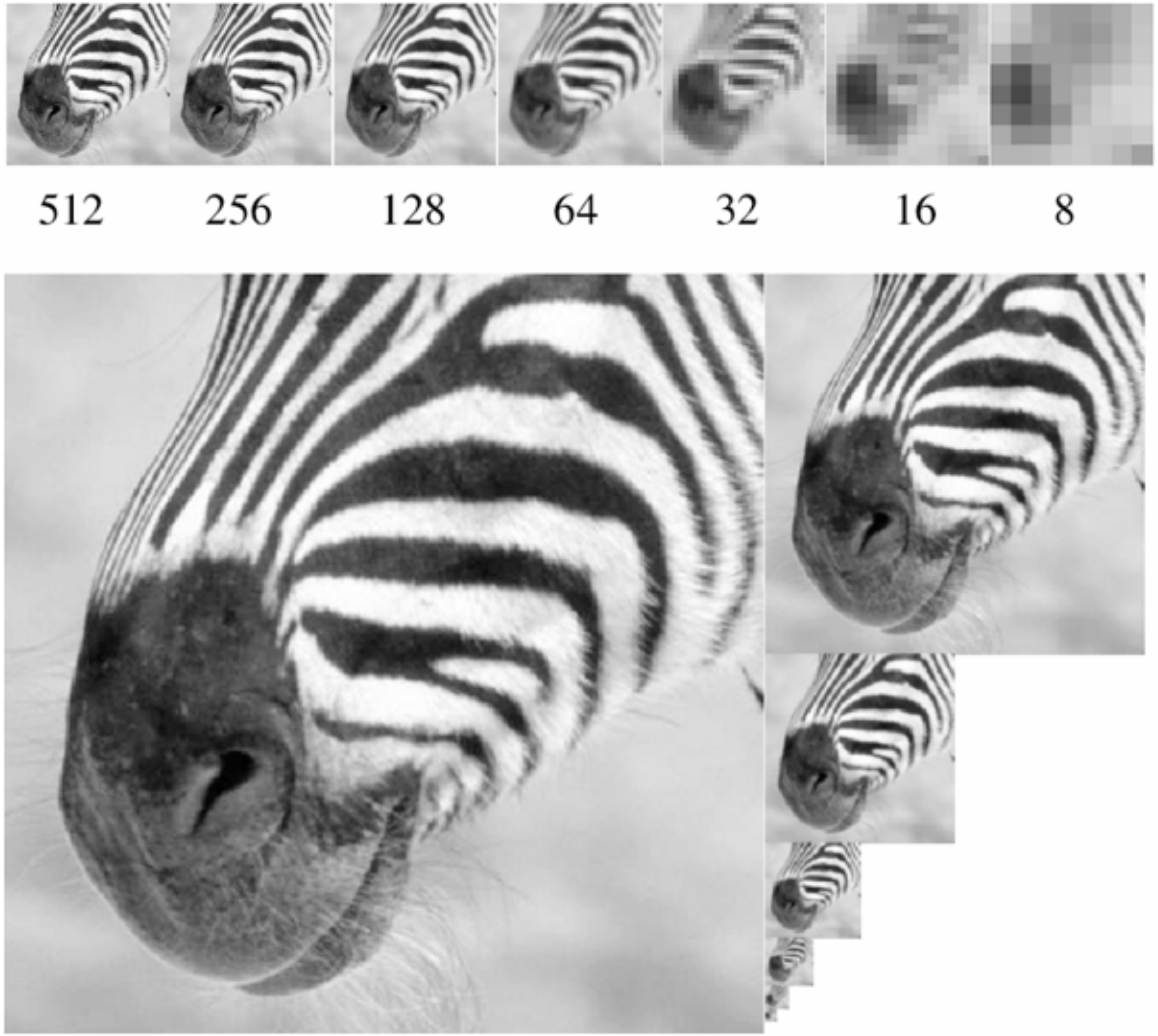
Spots and Bars (Coarse Scale)





Forsyth & Ponce (1st ed.) Figures 9.3 and 9.5

Gaussian Pyramid



Forsyth & Ponce (2nd ed.) Figure 4.17

What happens to the details?

— They get smoothed out as we move to higher levels

What is preserved at the higher levels?

— Mostly large uniform regions in the original image

How would you reconstruct the original image from the image at the upper level?

— That's not possible

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Laplacian Pyramid

Building a **Laplacian** pyramid:

— Create a Gaussian pyramid

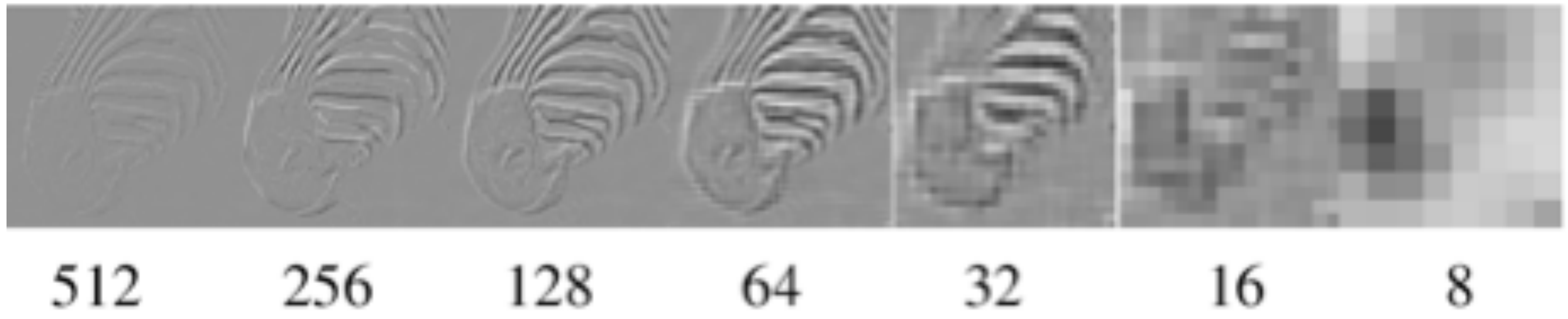
— Take the difference between one Gaussian pyramid level and the next (before subsampling)

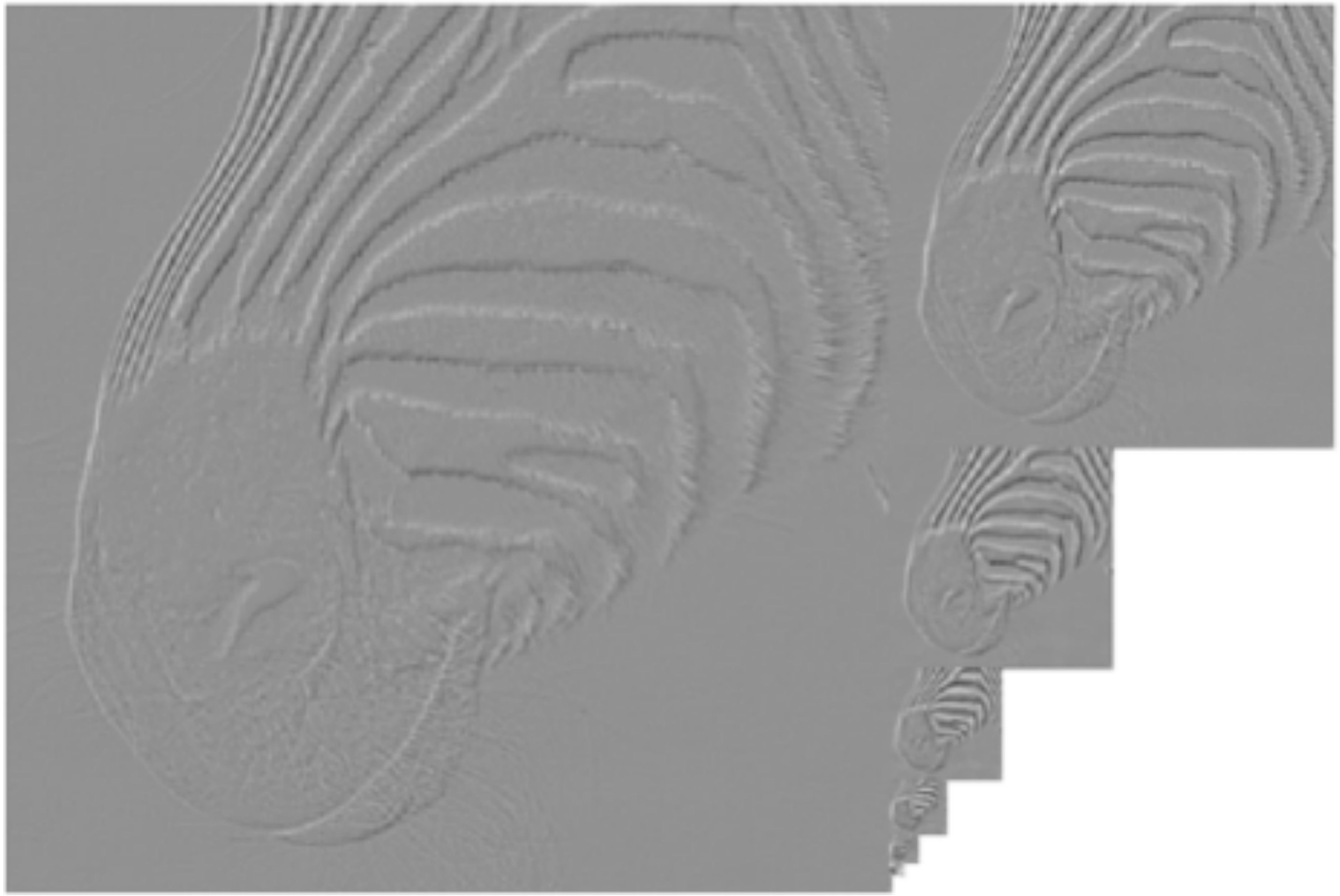
Properties

— Also known as the difference-of-Gaussian (DOG) function, a close approximation to the Laplacian

— It is a band pass filter – each level represents a different band of spatial frequencies

Laplacian Pyramid





At each level, retain the residuals instead of the blurred images themselves.

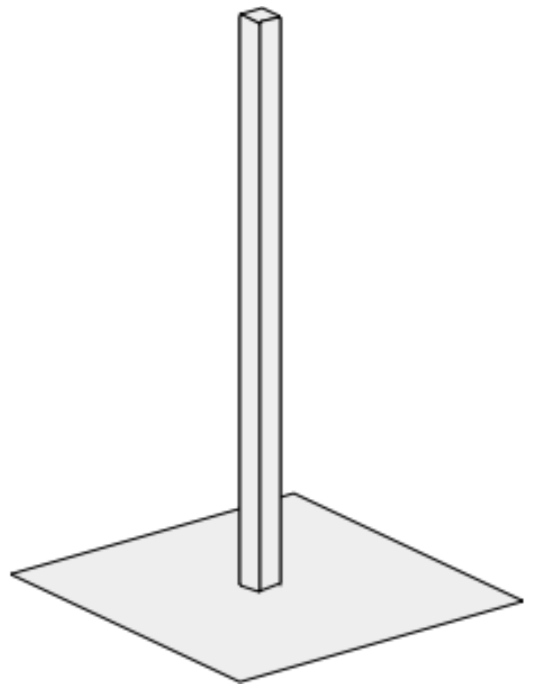
Why is it called Laplacian Pyramid?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

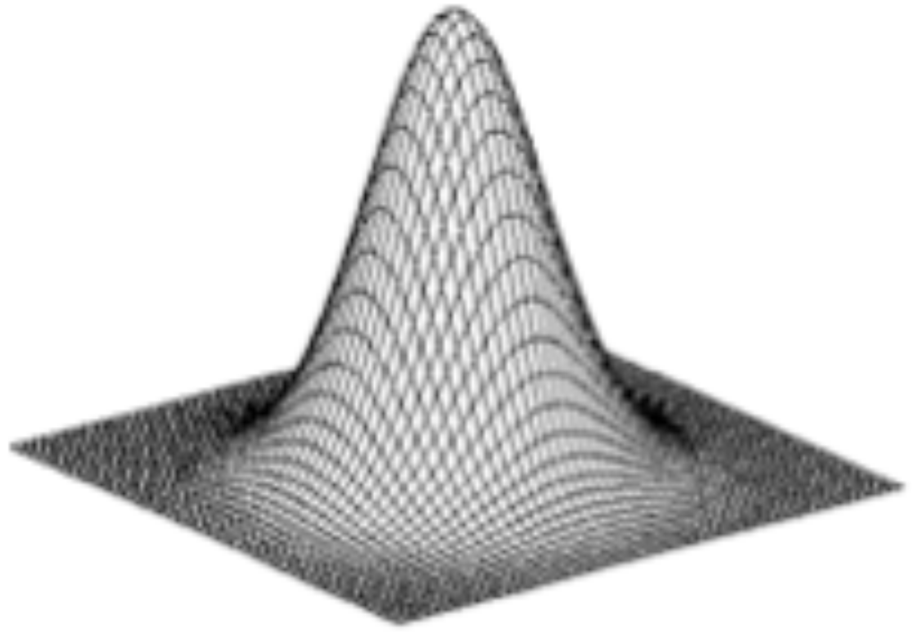
Why **Laplacian** Pyramid?

-

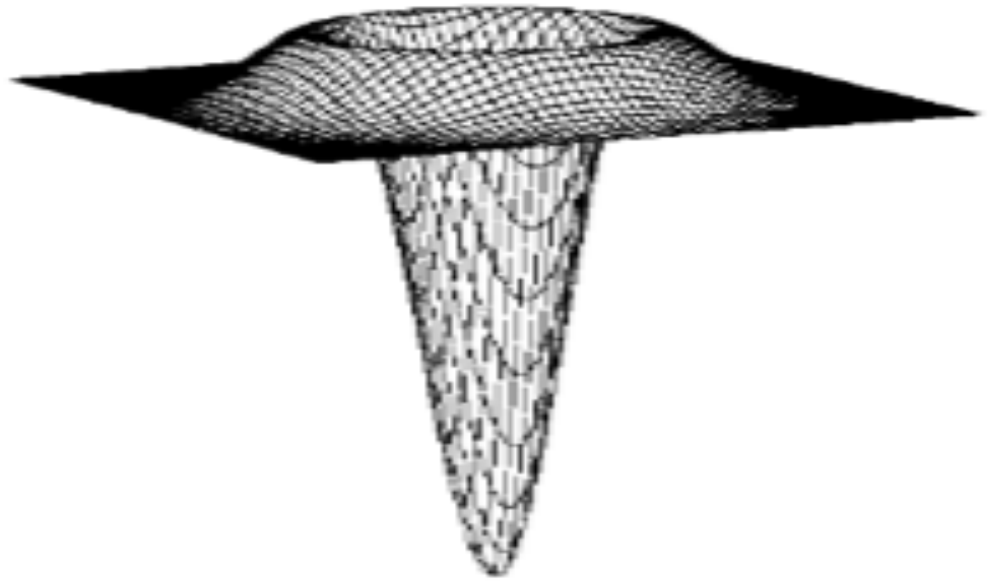
=



-



$\approx$



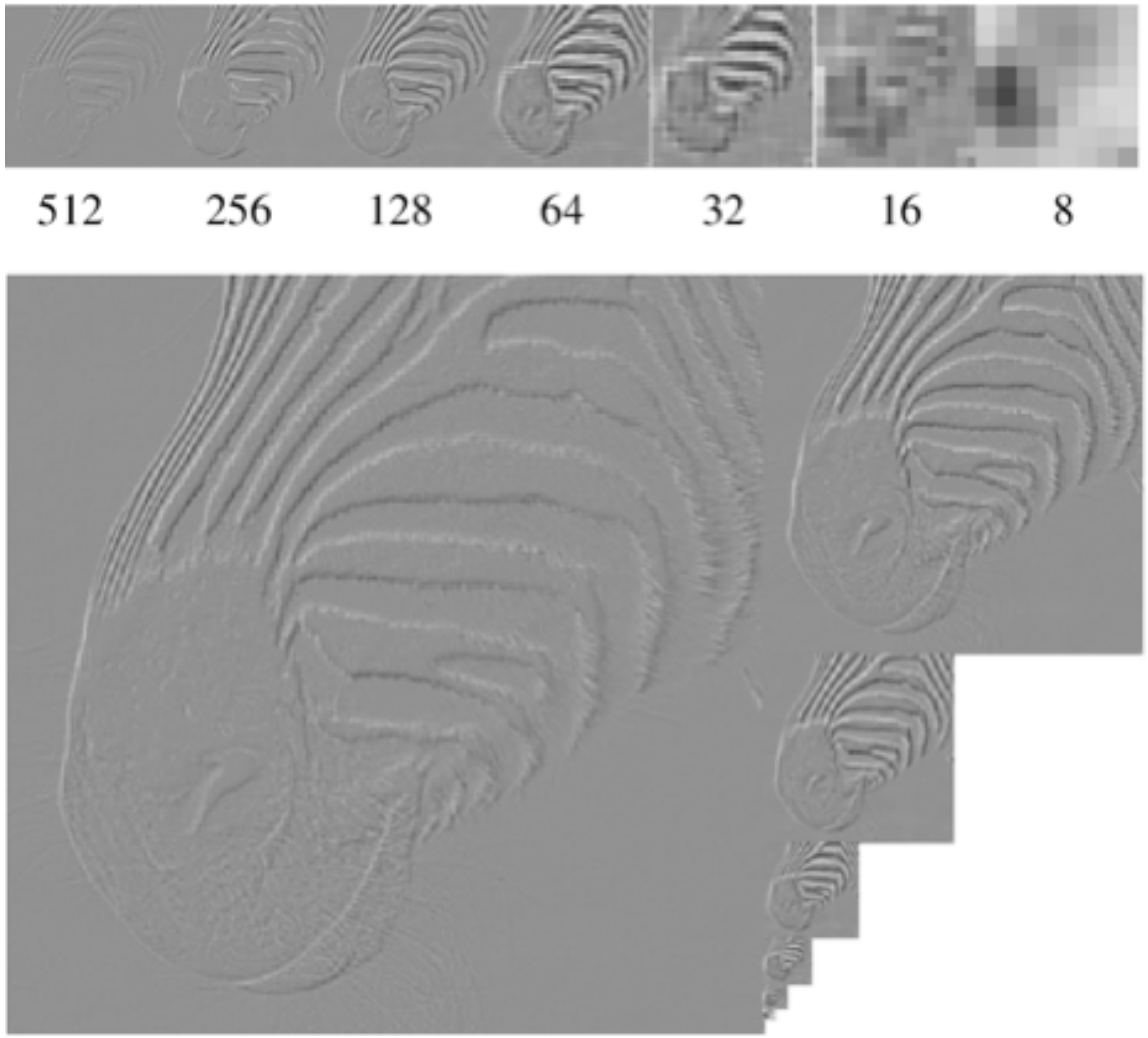
unit

Gaussian

Laplacian

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Laplacian Pyramid



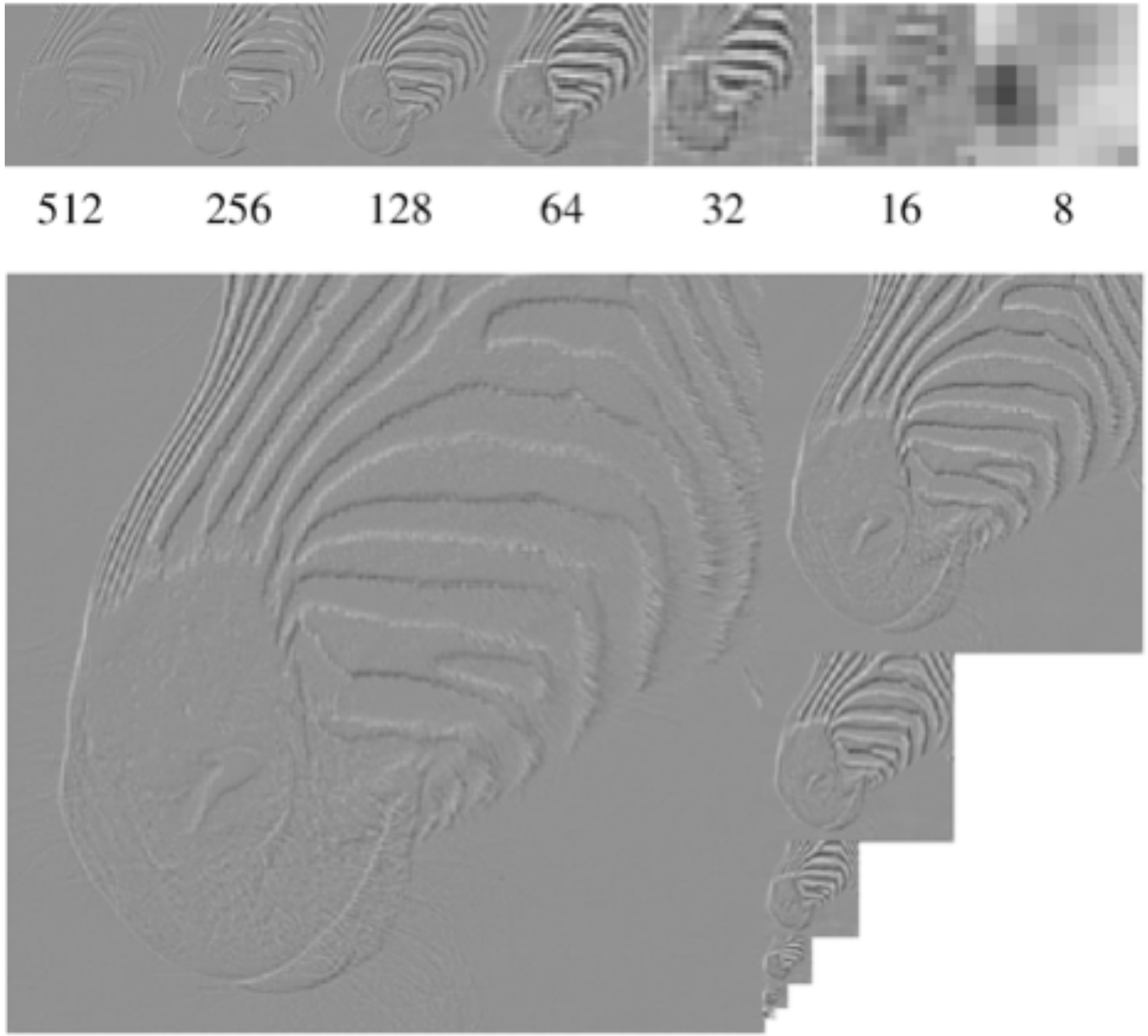
At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Can we reconstruct the original image using the pyramid?
— Yes we can!

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Laplacian Pyramid



At each level, retain the residuals instead of the blurred images themselves.

Why is it called Laplacian Pyramid?

Can we reconstruct the original image using the pyramid?
— Yes we can!

What do we need to store to be able to reconstruct the original image?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Let's start by just looking at **one level**

=

+

level 0

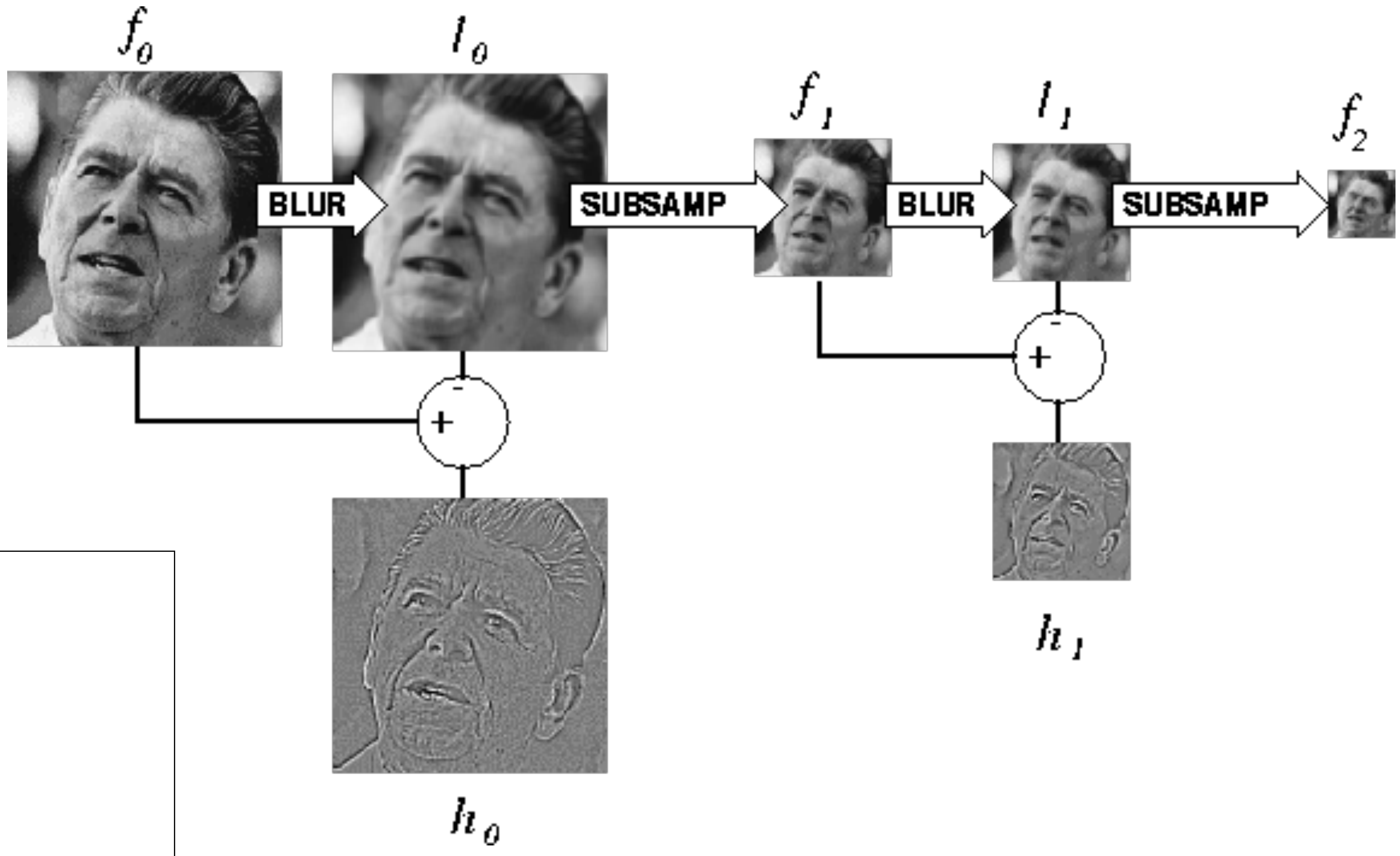
level 1 (upsampled)

residual

Does this mean we need to store both residuals and the blurred copies of the original?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Constructing a **Laplacian** Pyramid



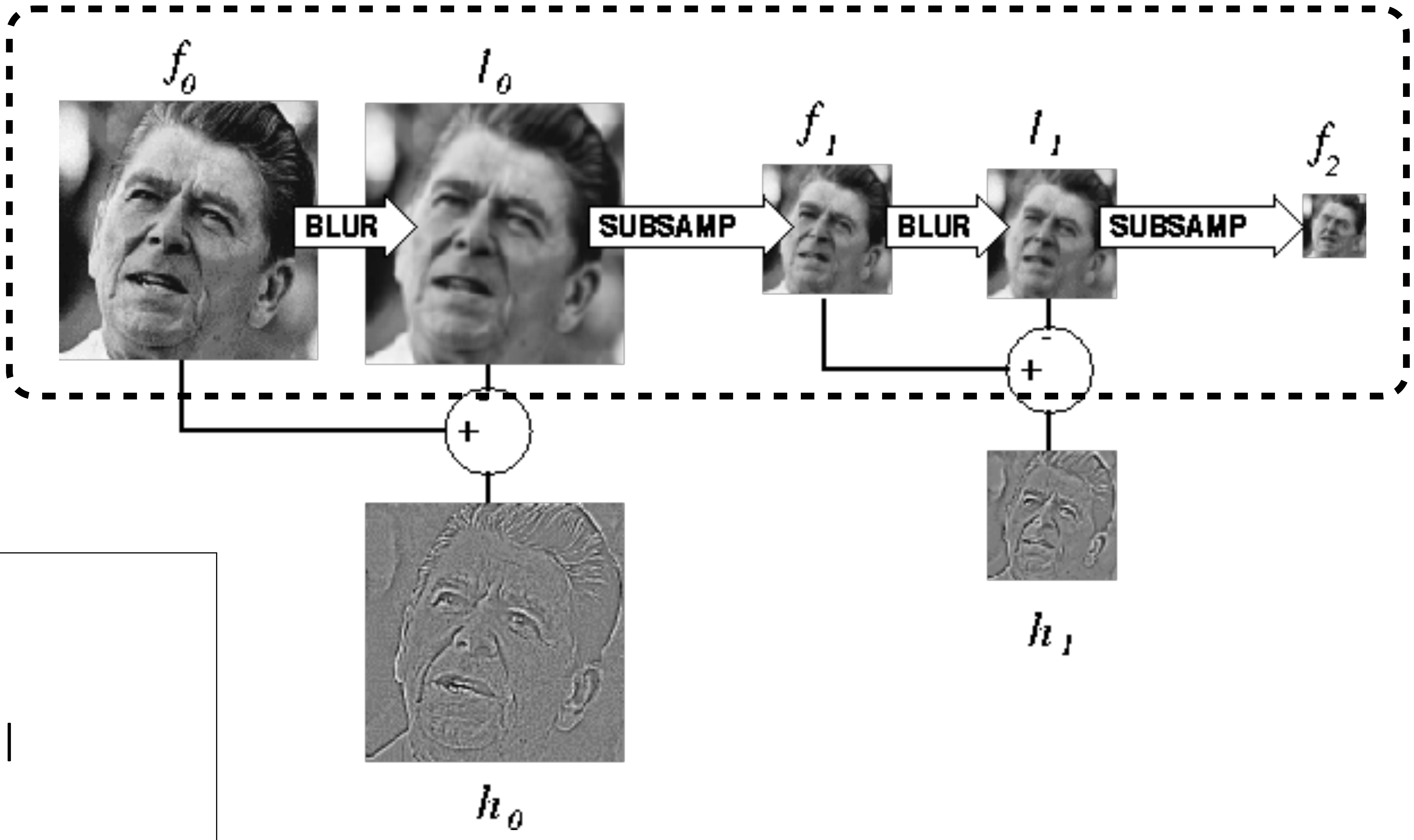
Algorithm

repeat:
    filter
    compute residual
    subsample
until min resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Constructing a **Laplacian** Pyramid

What is this part?



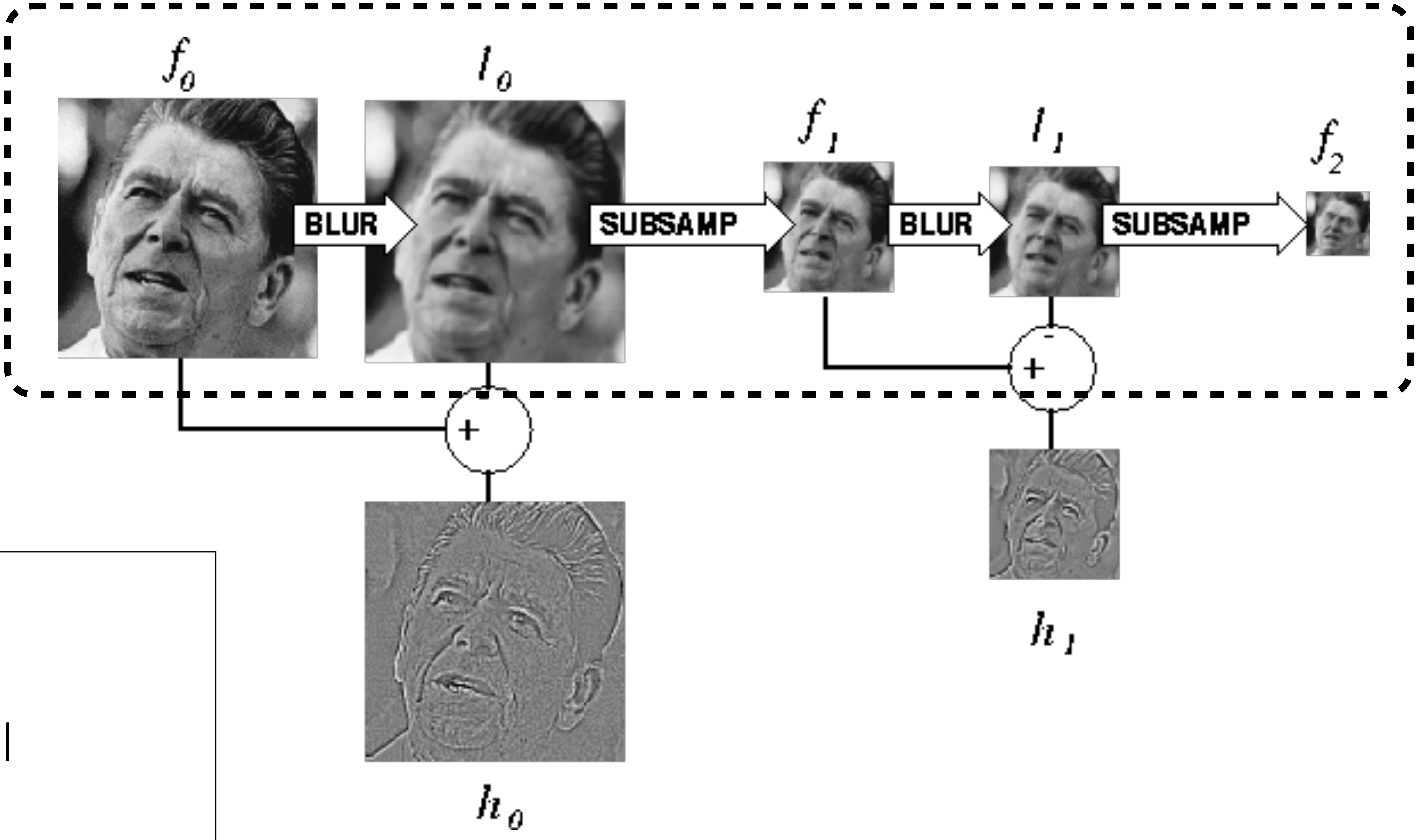
Algorithm

repeat:
filter
compute residual
subsample
until min resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Constructing a **Laplacian** Pyramid

It's a Gaussian Pyramid

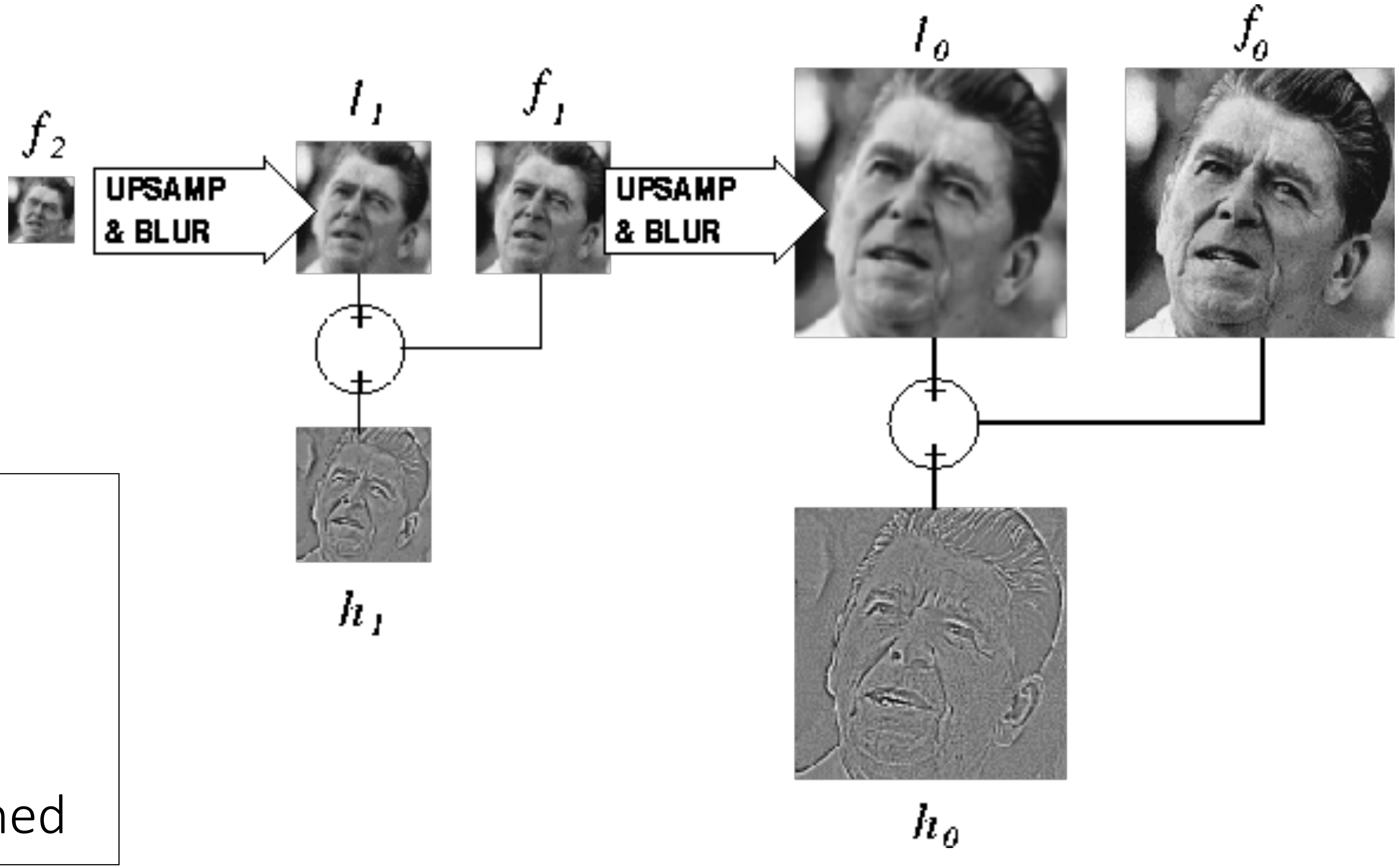


Algorithm

repeat:
filter
compute residual
subsample
until min resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Reconstructing the Original Image



Algorithm

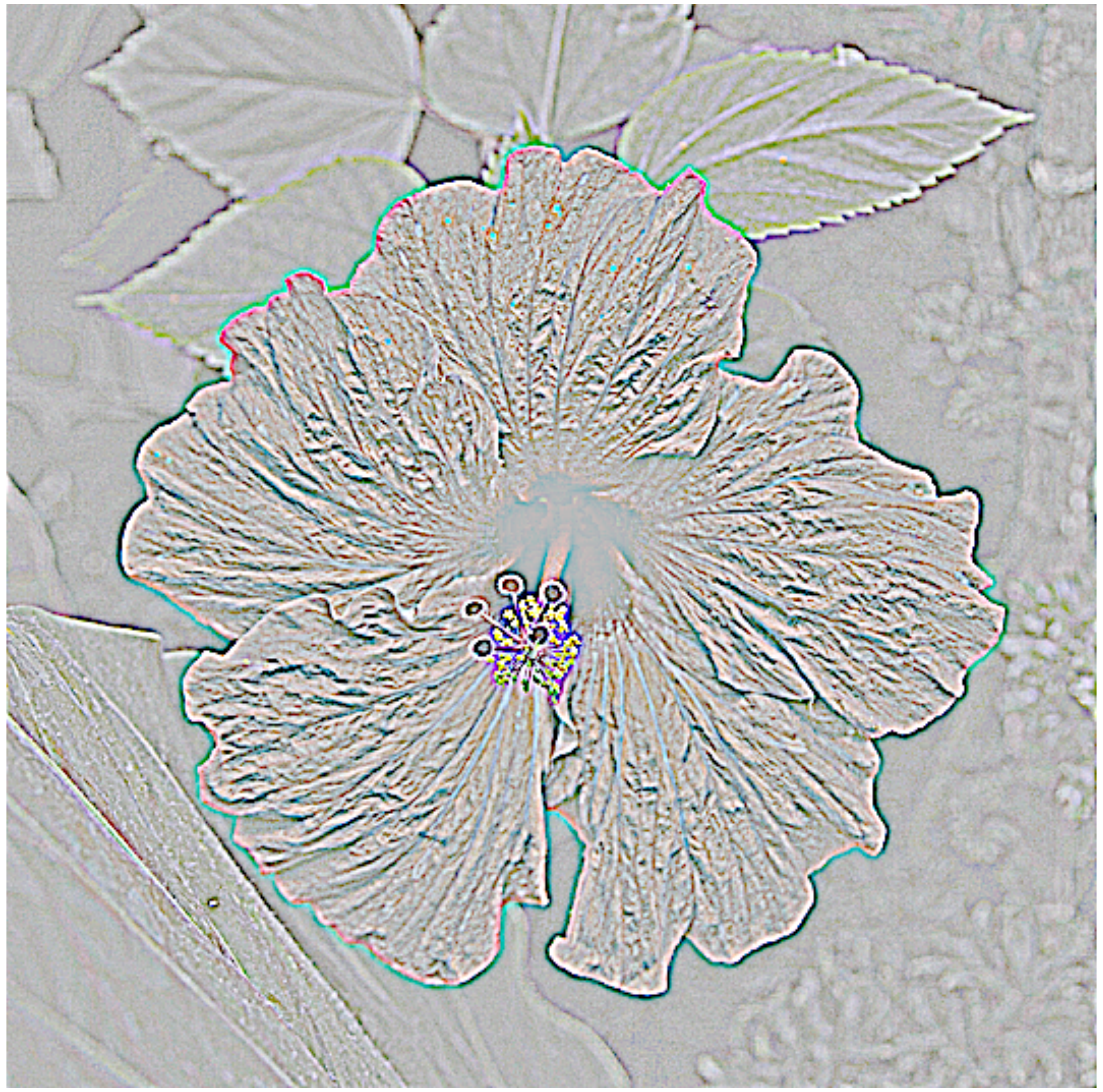
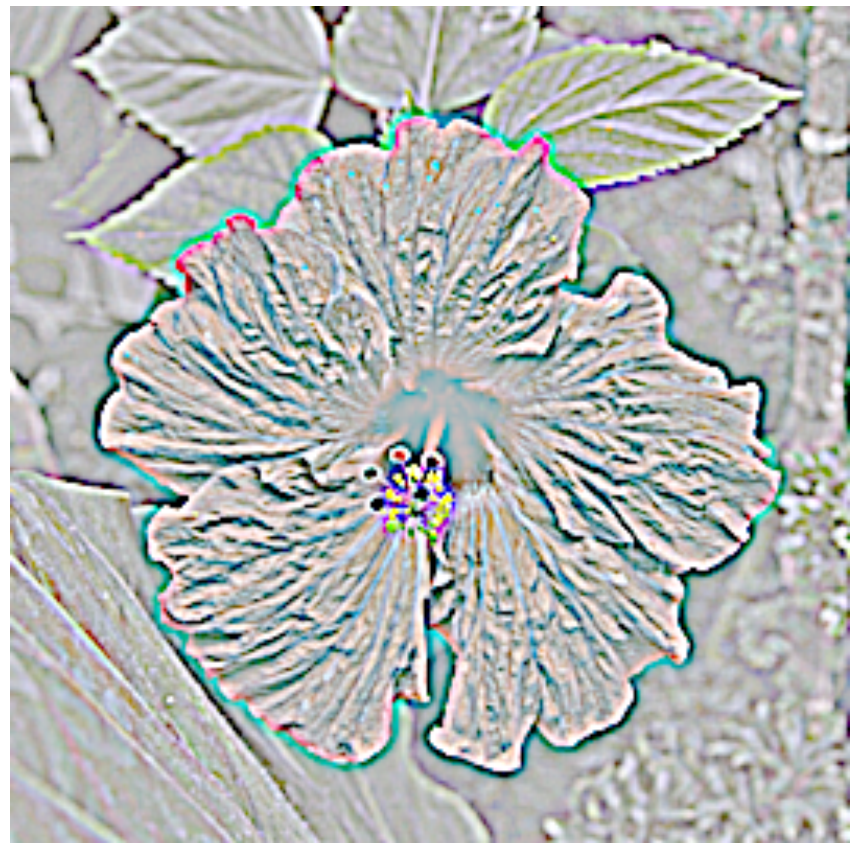
repeat: upsample sum with residual until orig resolution reached

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Gaussian vs **Laplacian** Pyramid



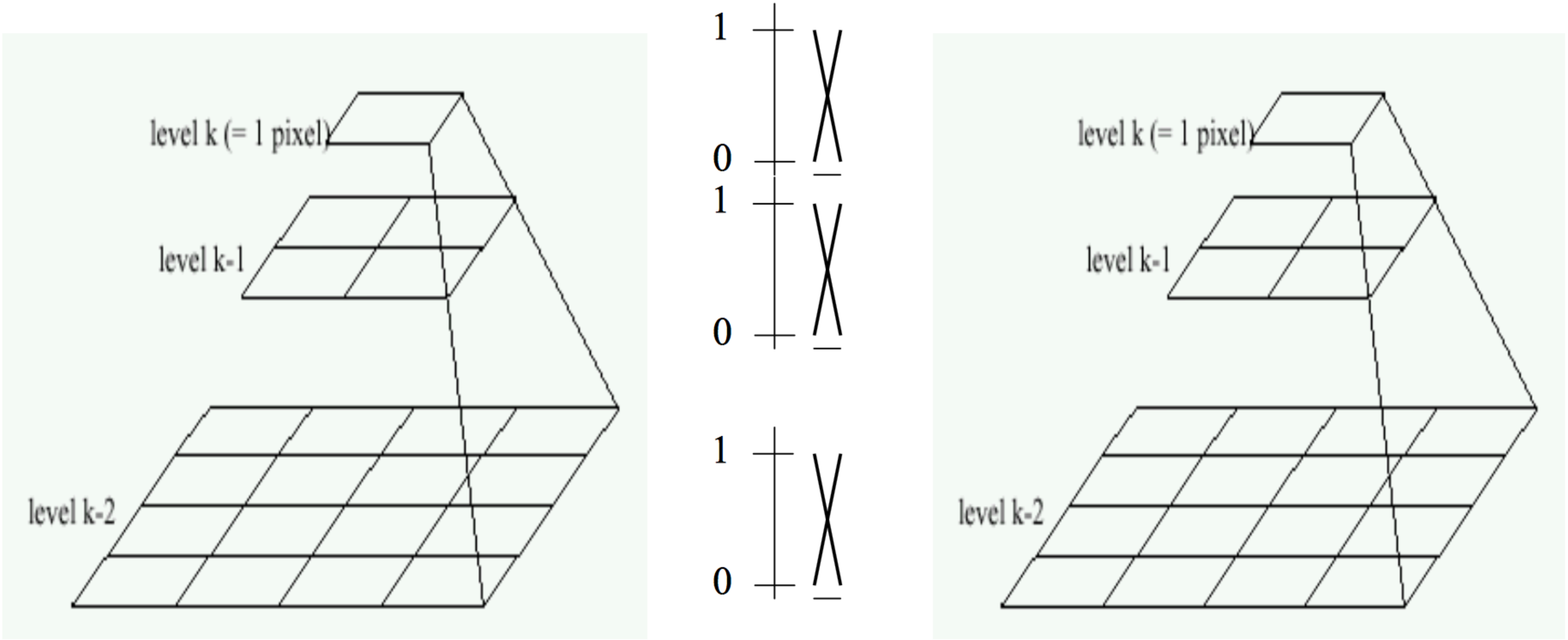

Shown in opposite order for space

Which one takes more space to store?

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

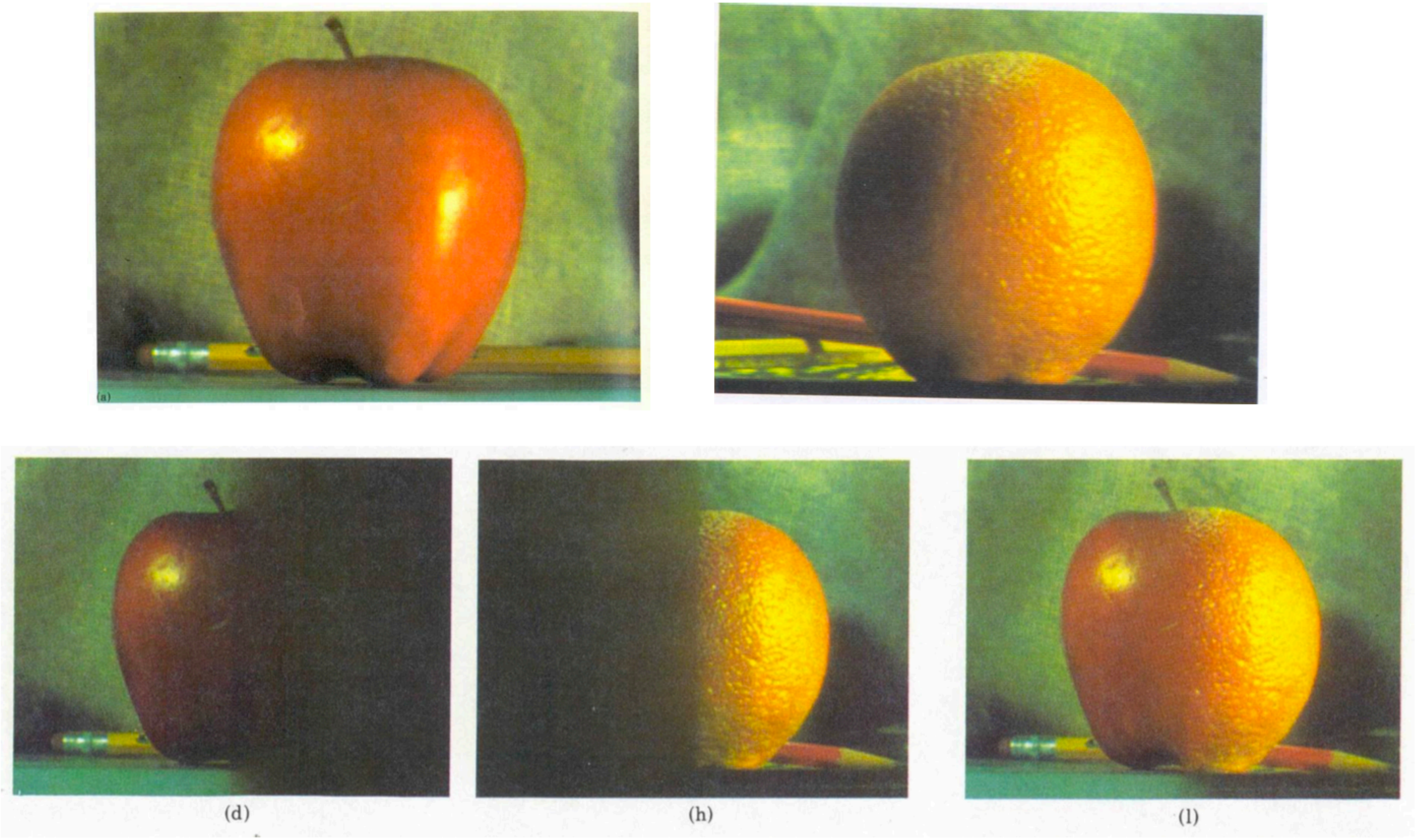
Aside: Image Blending



Left pyramid blend Right pyramid

Burt and Adelson, “A multiresolution spline with application to image mosaics,”ACM Transactions on Graphics, 1983, Vol.2, pp.217-236.

Aside: Image Blending



Burt and Adelson, “A multiresolution spline with application to image mosaics,”ACM Transactions on Graphics, 1983, Vol.2, pp.217-236.

Aside: Image Blending

Algorithm:

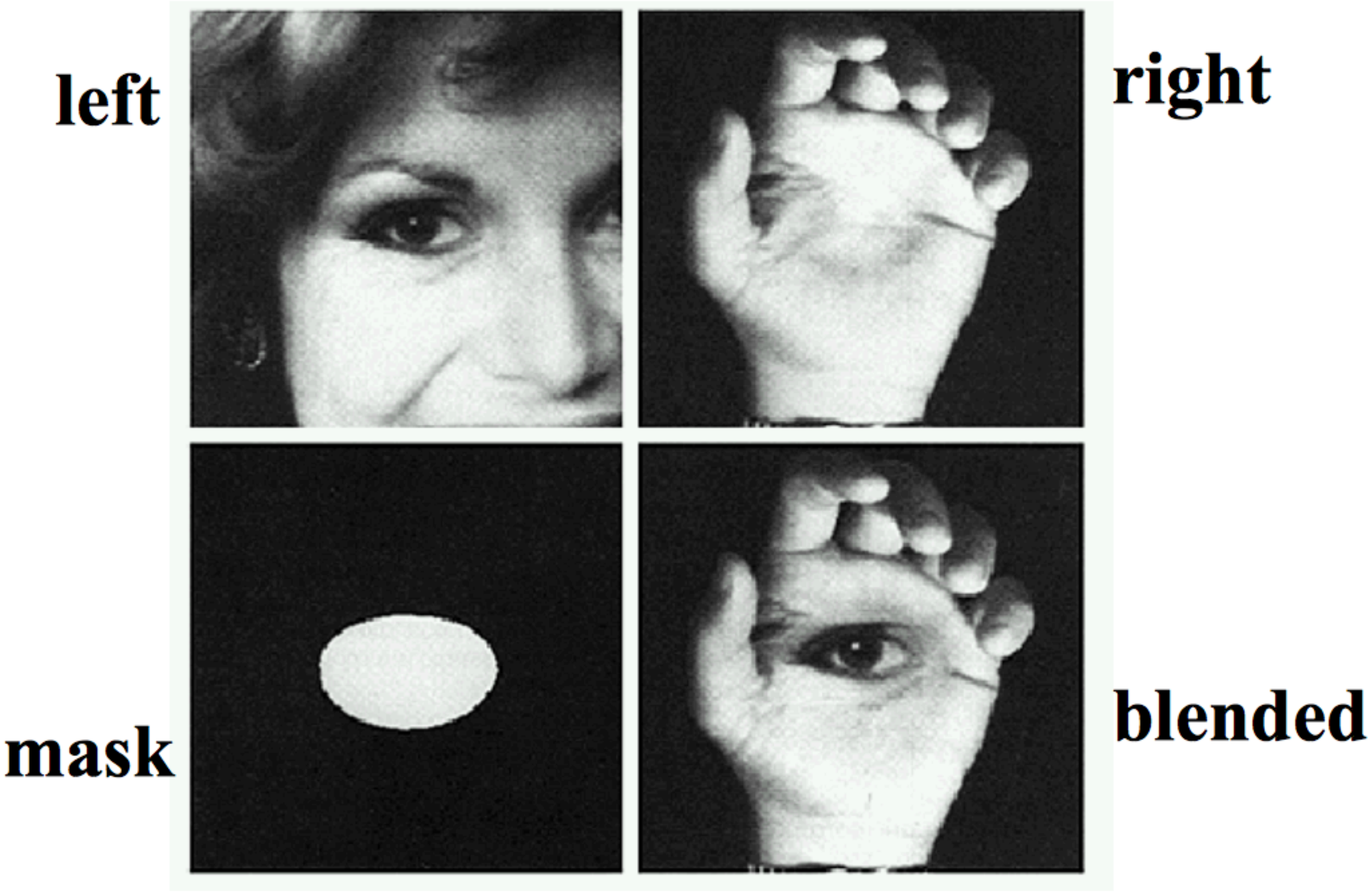
1. Build Laplacian pyramid LA and LB from images A and B

2. Build a Gaussian pyramid GR from mask image R (the mask defines which image pixels should be coming from A or B)

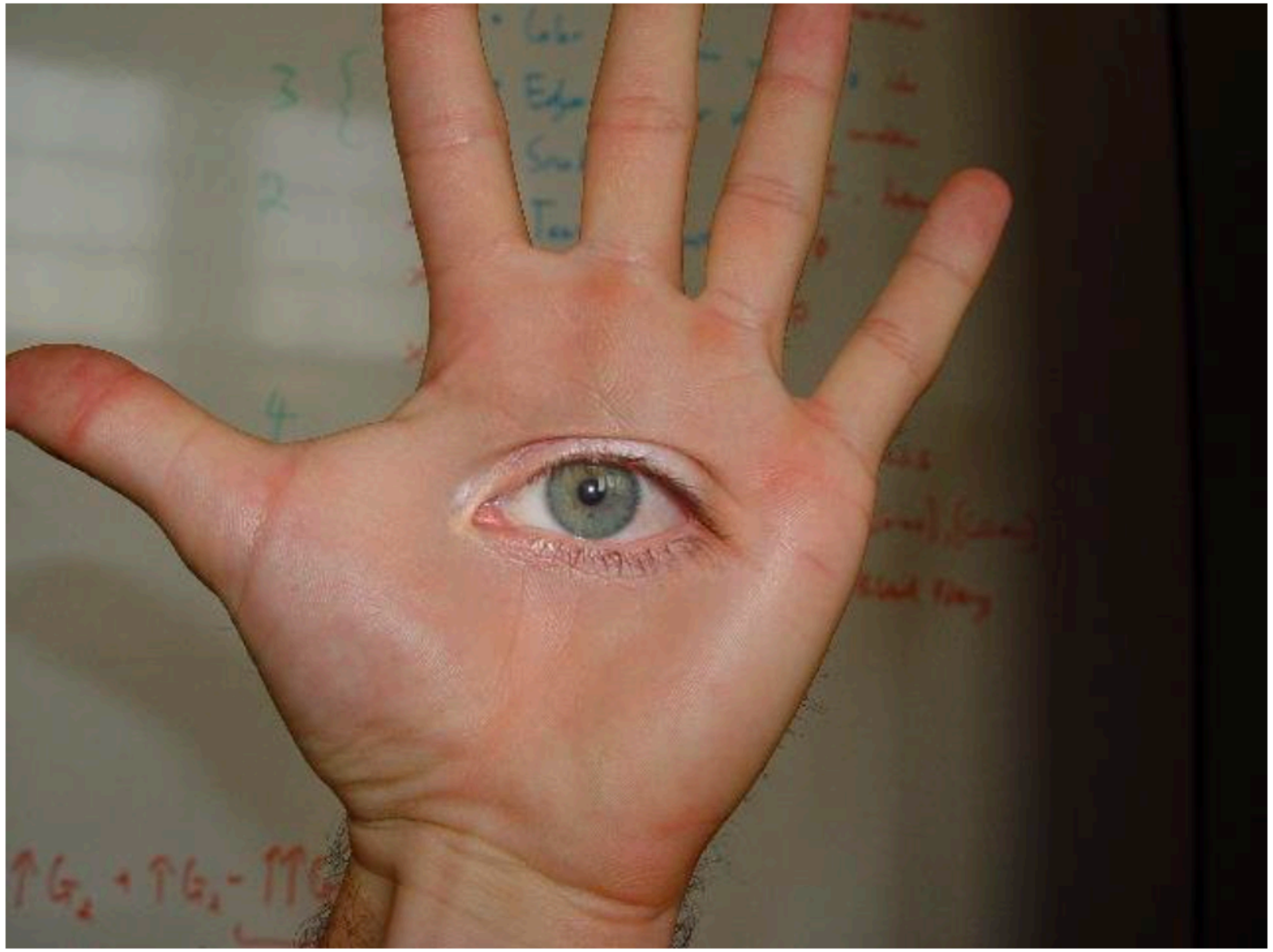
3. From a combined (blended) Laplacian pyramid LS, using nodes of GR as weights: $LS(i,j) = GR(i,j) * LA(i,j) + (1-GR(i,j)) * LB(i,j)$

4. Reconstruct the final blended image from LS

Aside: Image Blending



Aside: Image Blending



© david dmartin (Boston College)

Aside: Image Blending

© Chris Cameron

Oriented Pyramids

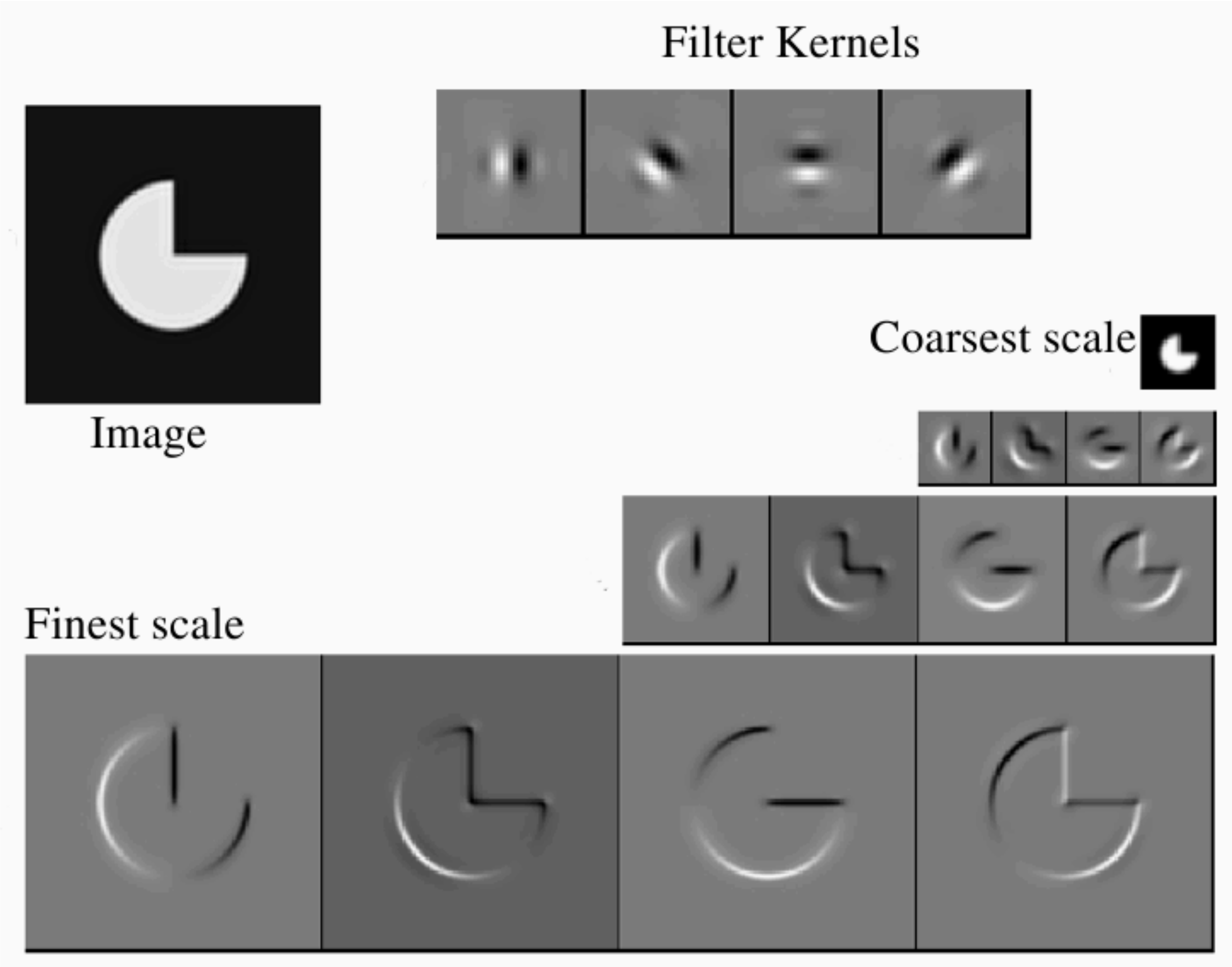
Laplacian pyramid is orientation independent

Idea: Apply an oriented filter at each layer

— represent image at a particular scale and orientation

— Aside: We do not study details in this course

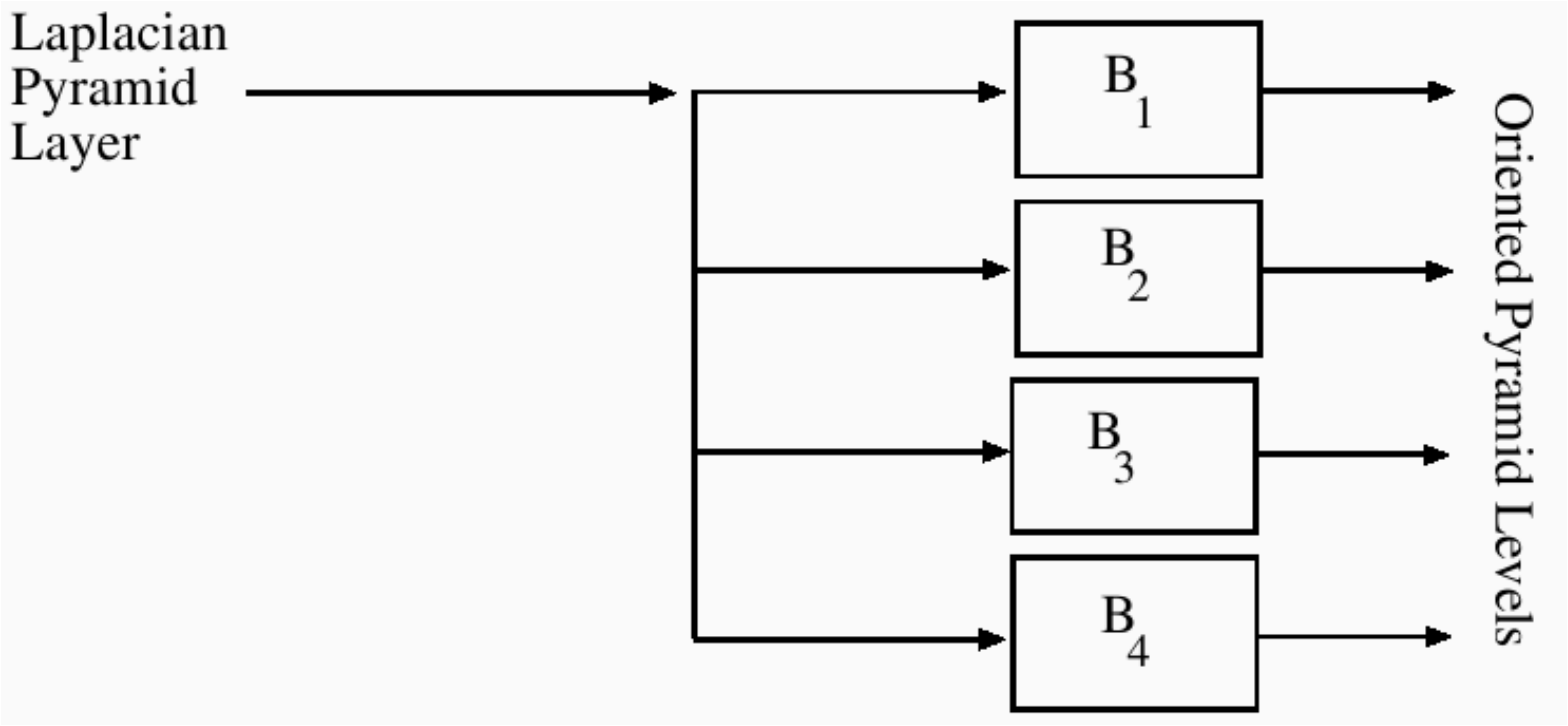
Oriented Pyramids



Forsyth & Ponce (1st ed.) Figure 9.13

Oriented Pyramids

Oriental Filters



Forsyth & Ponce (1st ed.) Figure 9.14

Final Texture Representaation

Steps:

1. Form a Laplacian and oriented pyramid (or equivalent set of responses to filters at different scales and orientations)

2. Square the output (makes values positive)

3. Average responses over a neighborhood by blurring with a Gaussian

4. Take statistics of responses
— Mean of each filter output
— Possibly standard deviation of each filter

Summary

Texture representation is hard

— difficult to define, to analyze

— texture synthesis appears more tractable

Objective of texture **synthesis** is to generate new examples of a texture

— Efros and Leung: Draw samples directly from the texture to generate one pixel at a time. A “data-driven" approach.

Approaches to texture embed assumptions related to human perception