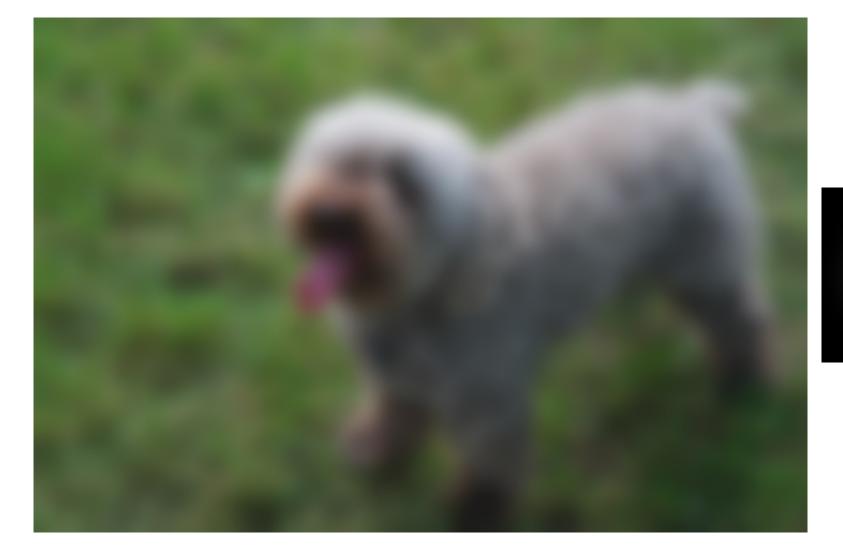
Lecture 3: Image pyramids

Announcements

- Reminder: PS1 due next Weds.
- Section this week:
 - Complex numbers and frequencies
 - Fourier transform
- Suggested reading: Szeliski chapter or Torralba, Isola, Freeman chapter

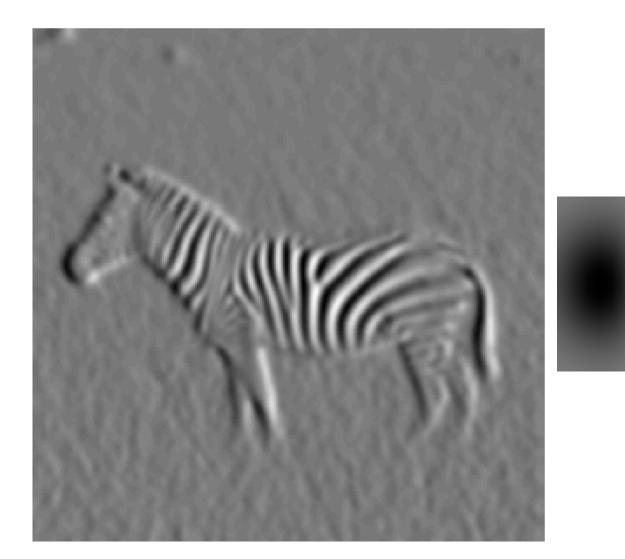
• Image pyramids Image statistics • Texture synthesis

Today

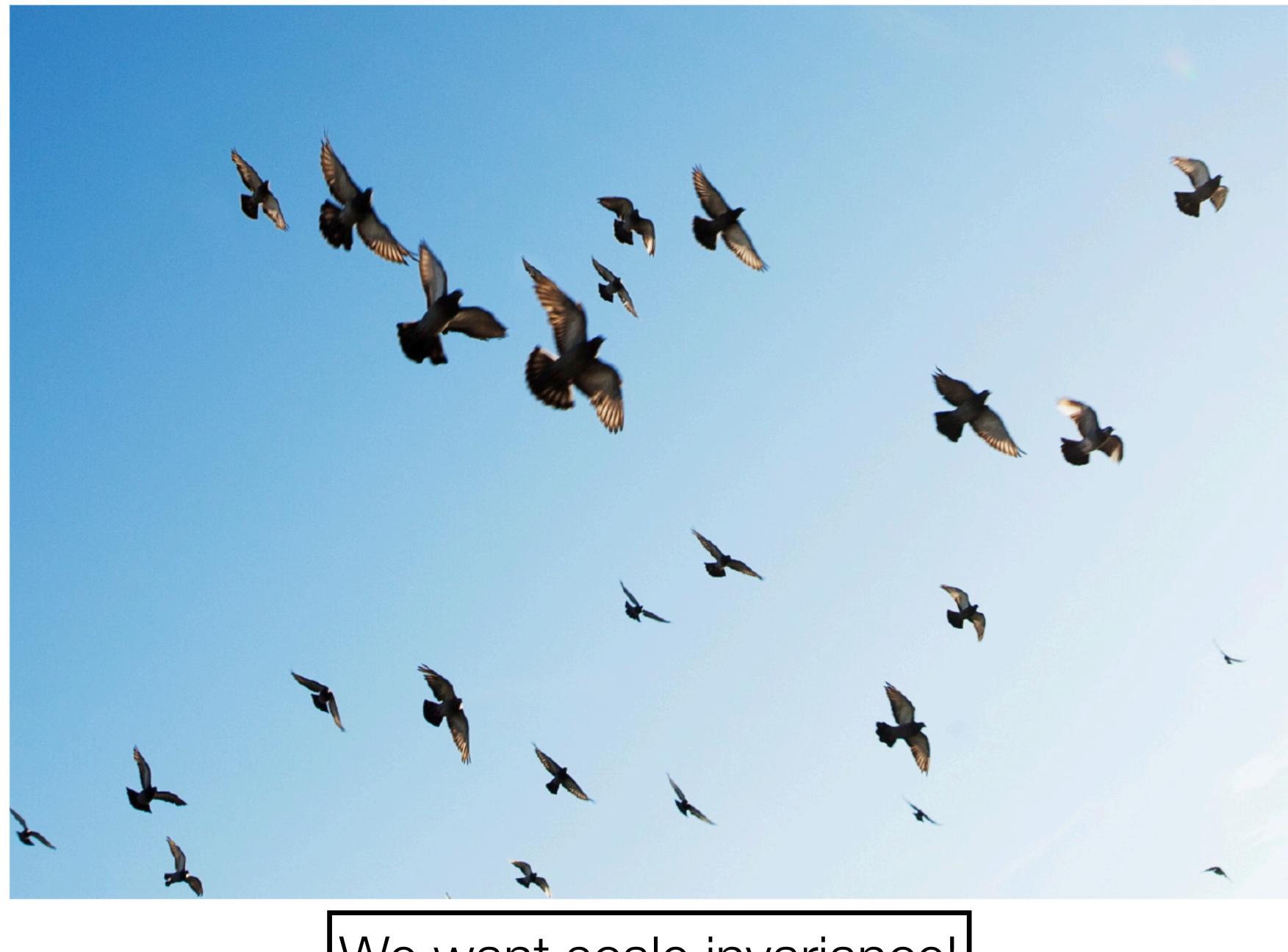




Last class: linear filtering



Derivative filters



We want scale invariance!



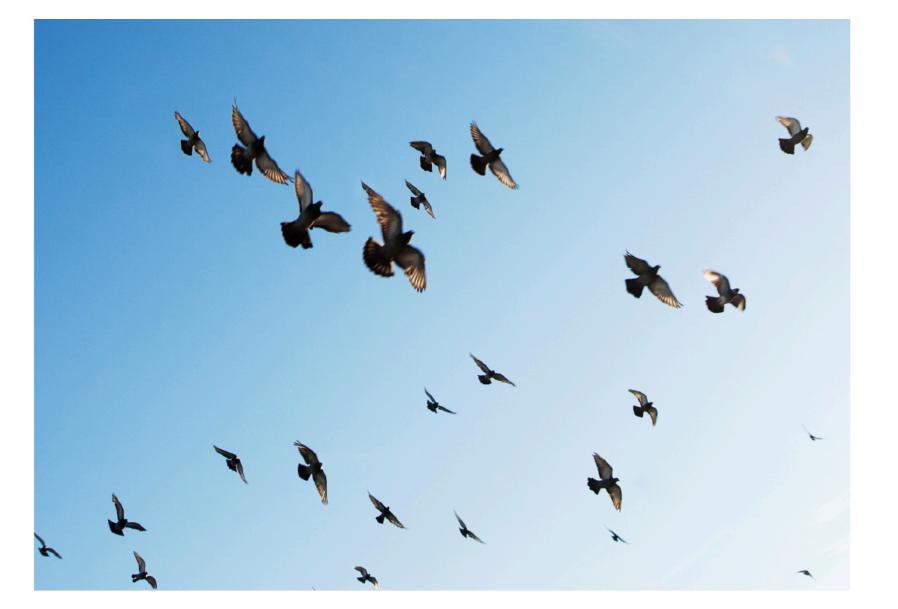
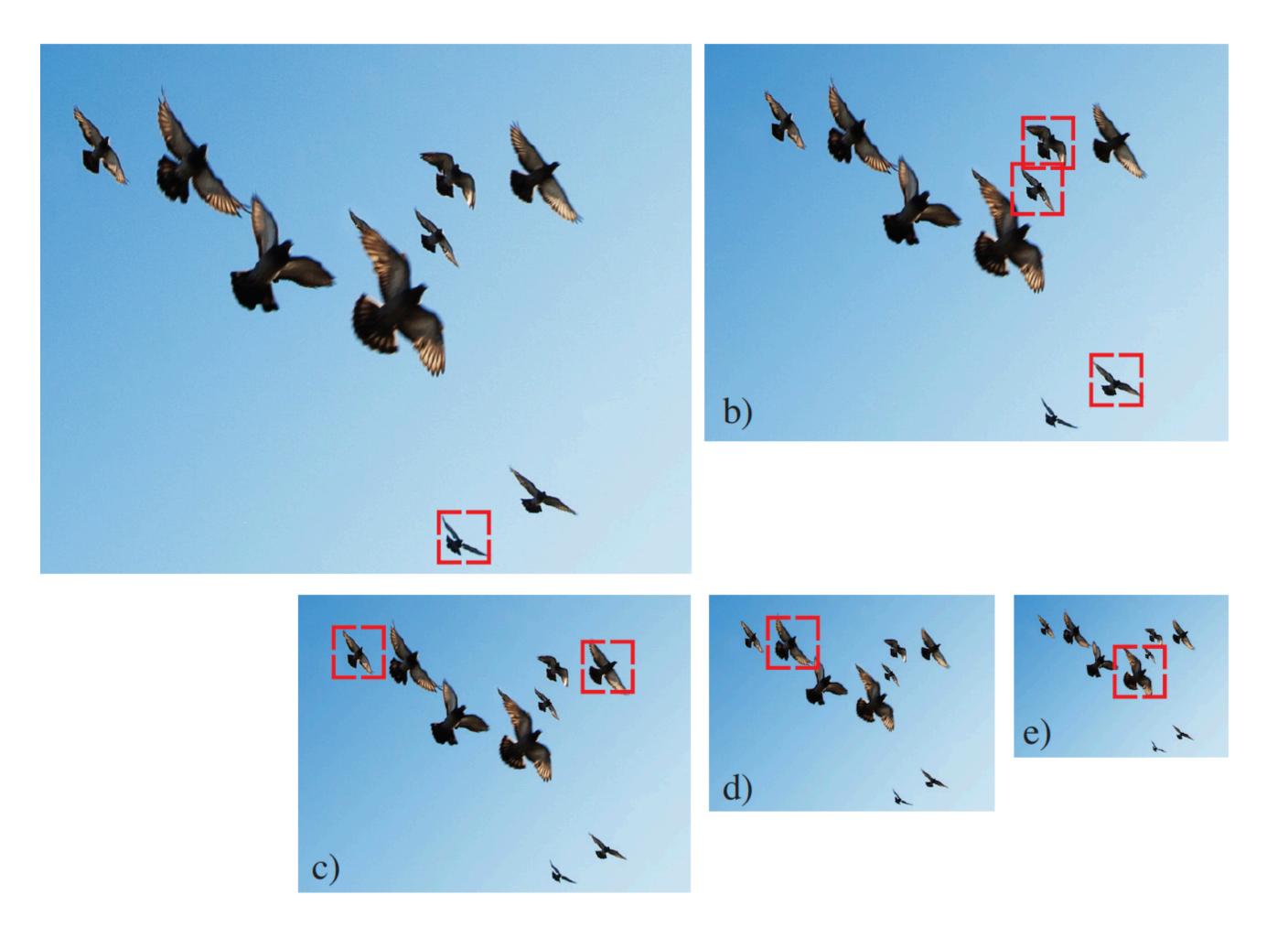


Image pyramids



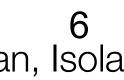
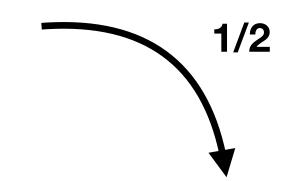




Image pyramid



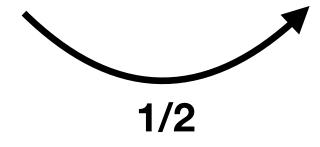




Subsampling and aliasing

103×128

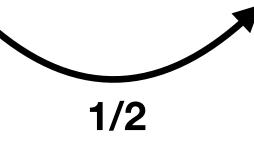




Idea #1: Throw away every other pixel.

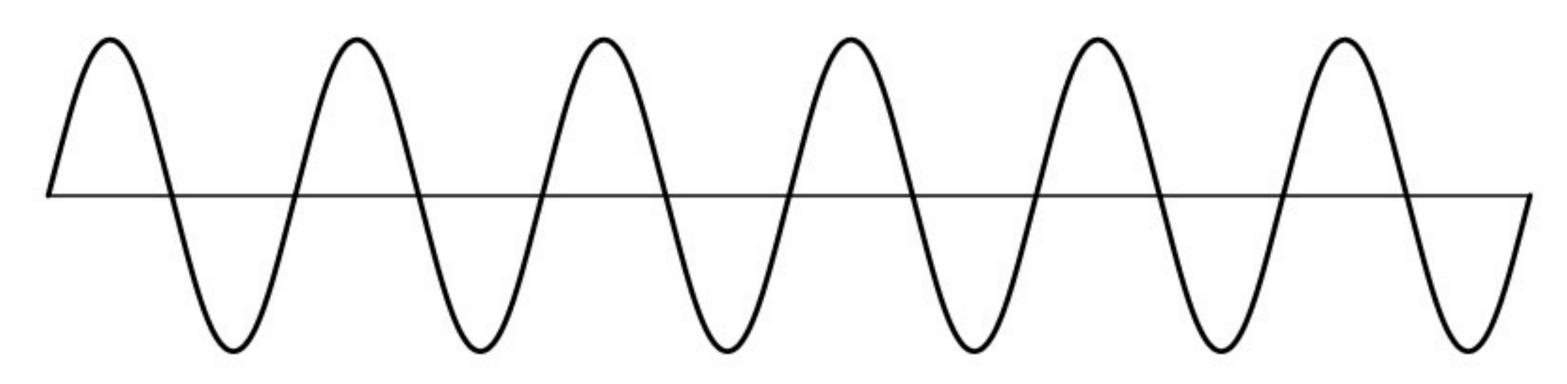
52×64

26×32



What's happening?

Consider a sinusoid:





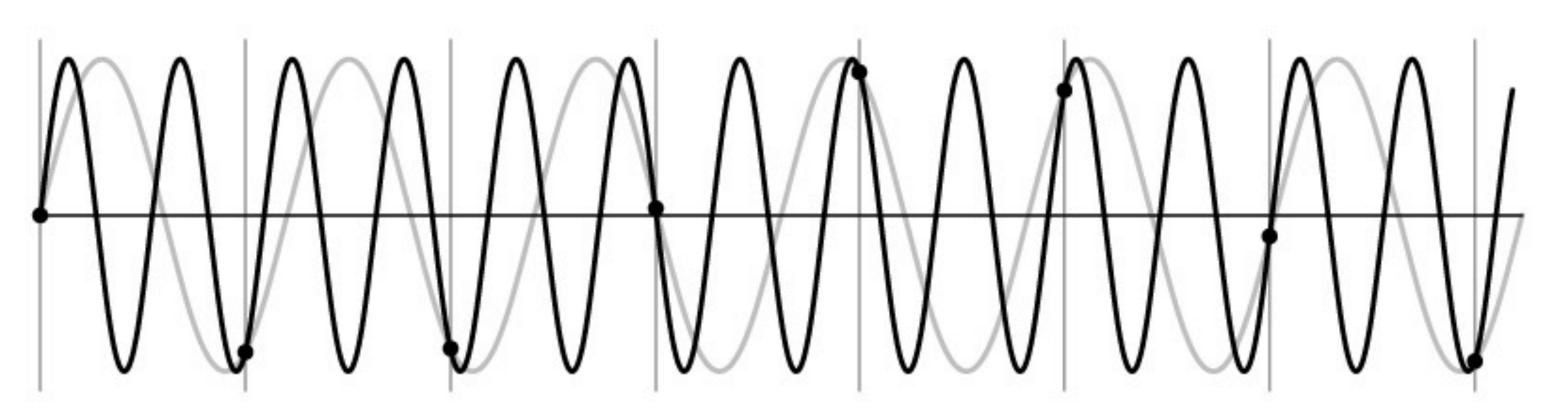
Source: S. Marschner

9



Undersampling

- What if we "missed" things between the samples?
- As expected, information is lost
- Unexpectedly: indistinguishable from low-frequency sinusoid!
- Also indistinguishable from higher frequencies
- Aliasing: signals "traveling in disguise" as other frequencies





Removing aliasing

Remove the high frequencies first! • Blur the image before downsampling • Next class we'll see why blurring does this



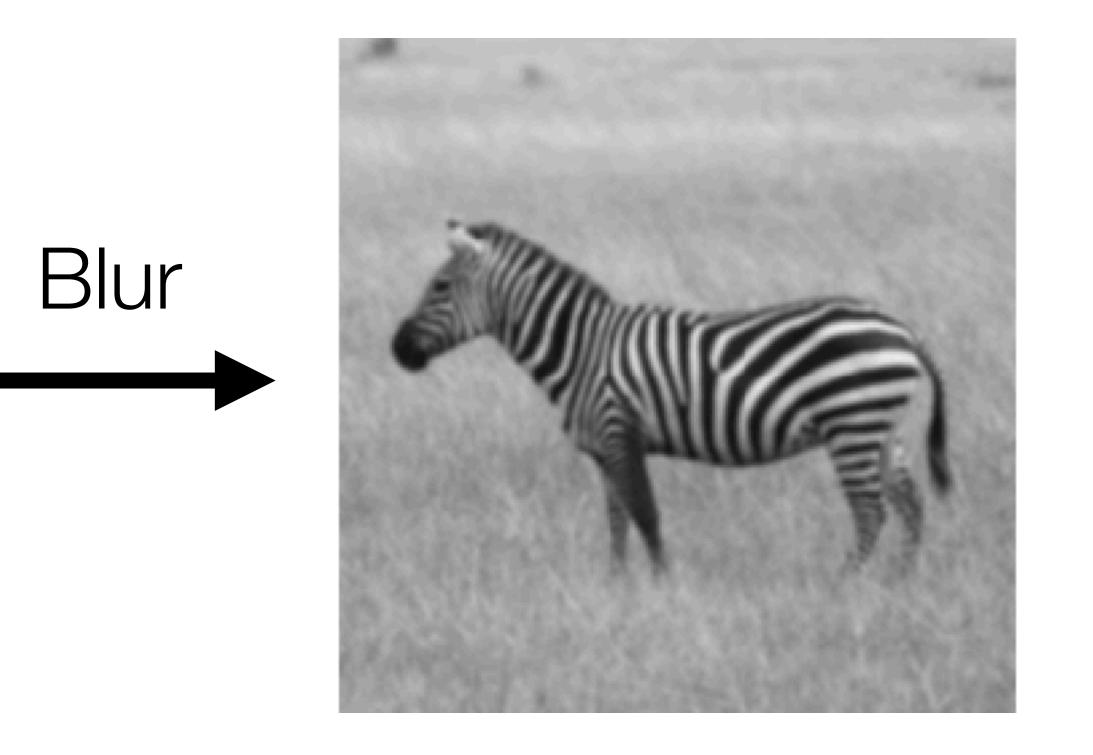


Blur



For each level: 1. Blur input image with a Gaussian (or binomial) filter





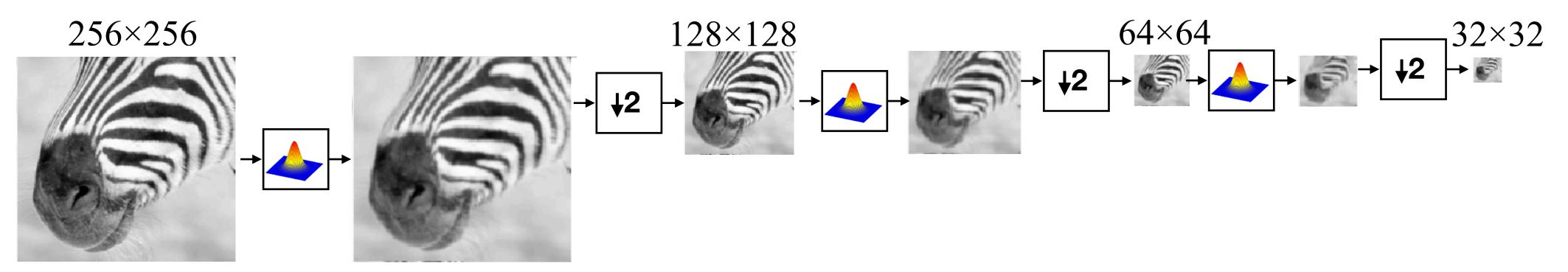


For each level: 1. Blur input image with a Gaussian (or binomial) filter 2. Downsample (throw away every other pixel)











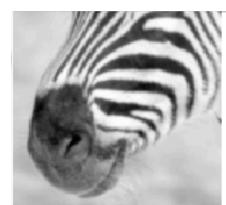
512×512



(original image)

256×256 128×128 64×64 32×32







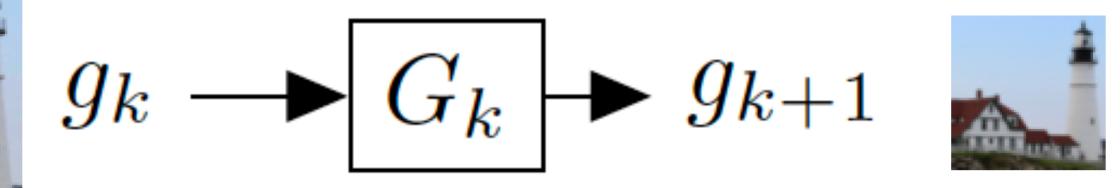


Source: Torralba, Freeman, Isola. Image from Forsyth & Ponce





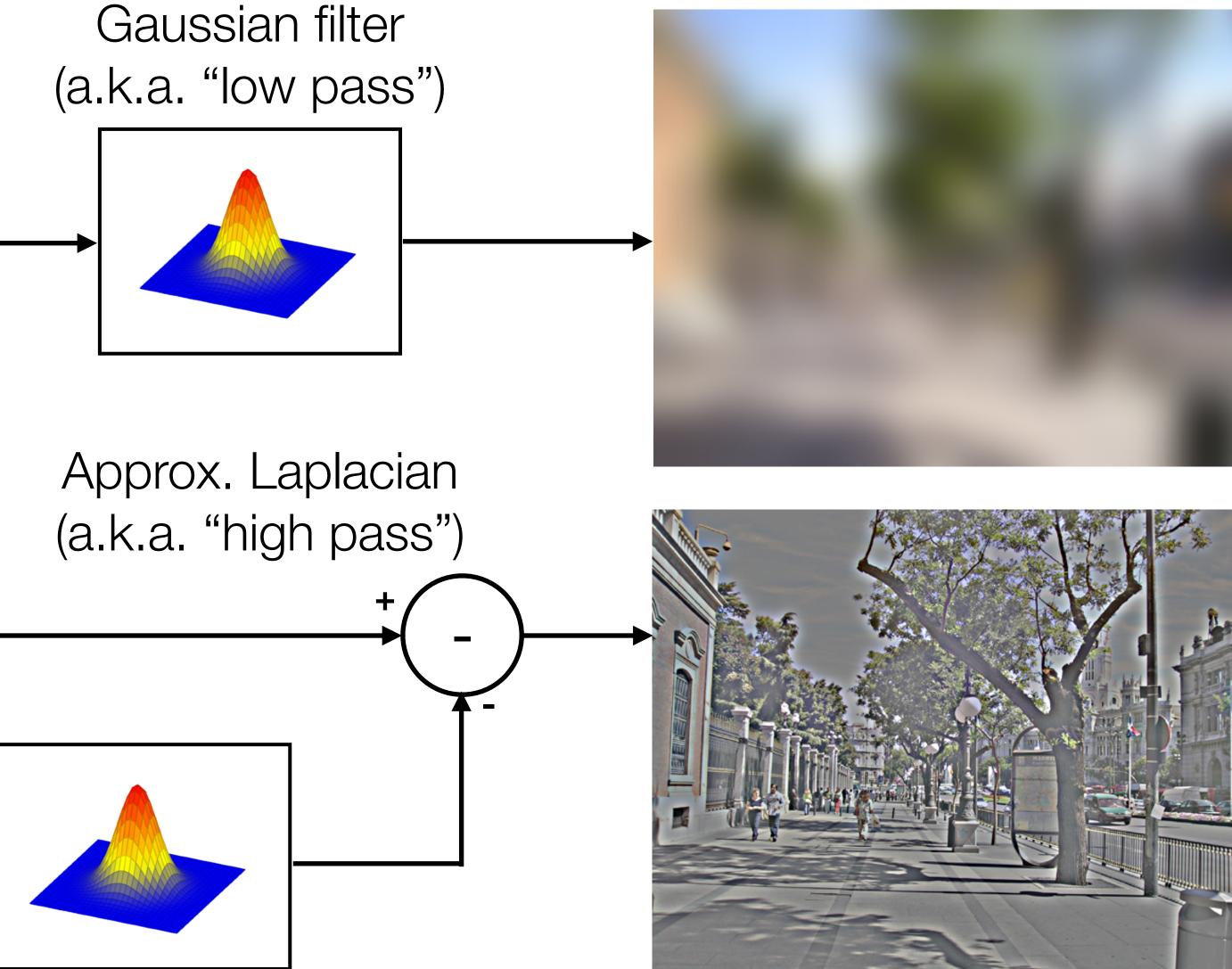
For each level 1. Blur input image with a Gaussian filter 2. Downsample image











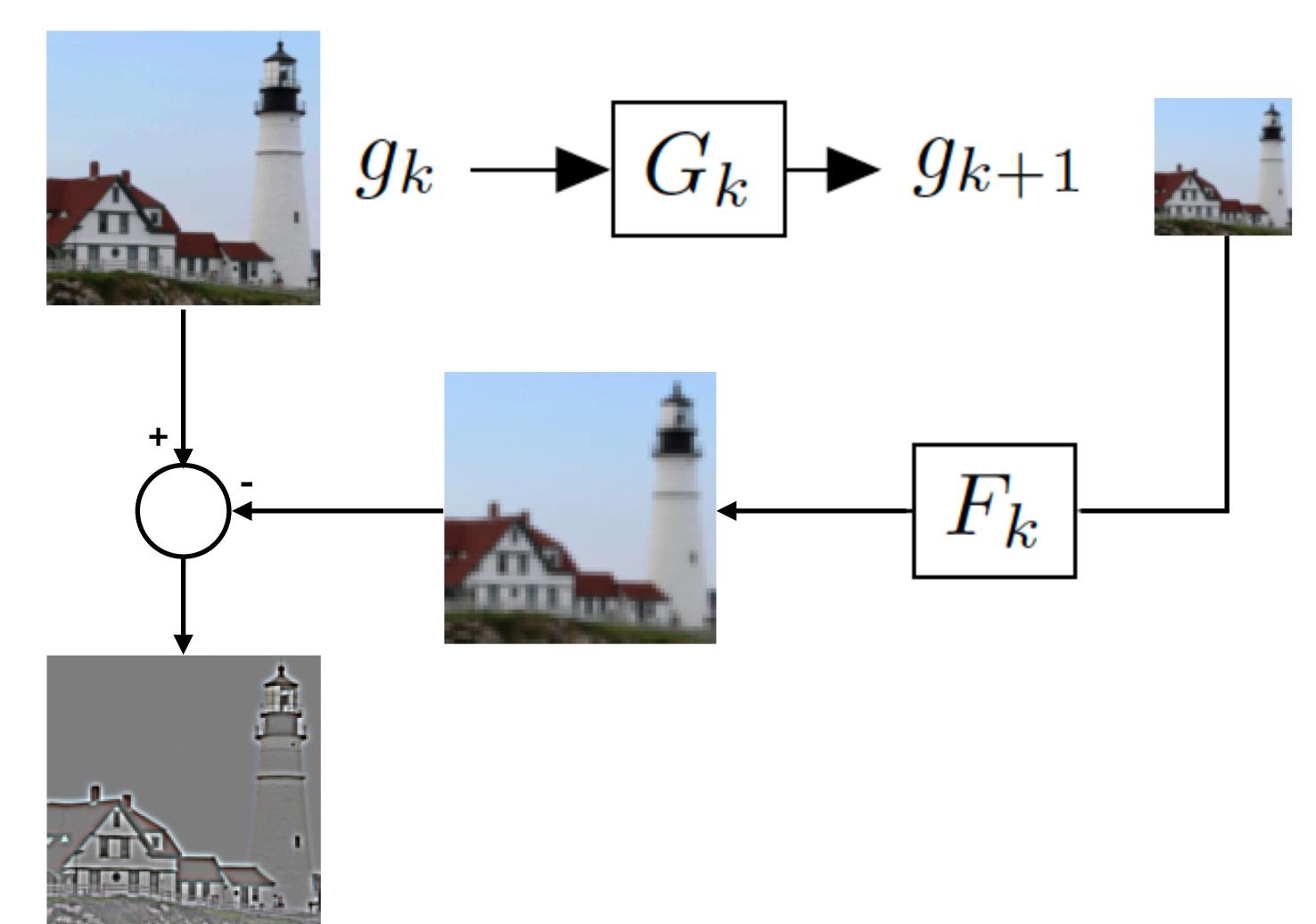
Recall: Laplacian







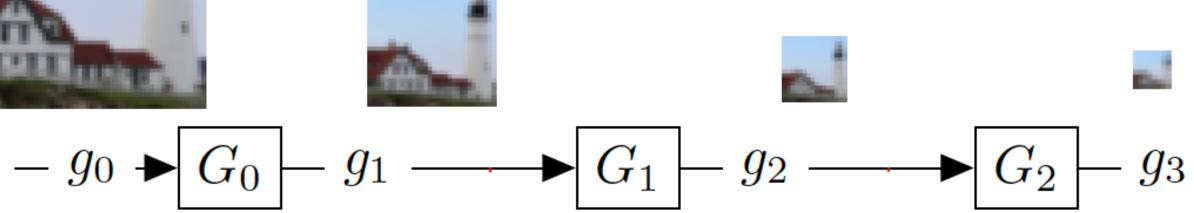
Compute the difference between upsampled Gaussian pyramid level k+1 and Gaussian pyramid level k. Recall that this approximates the blurred Laplacian.





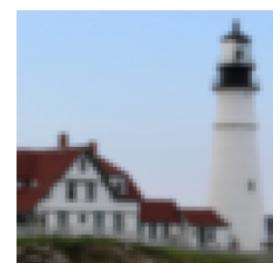




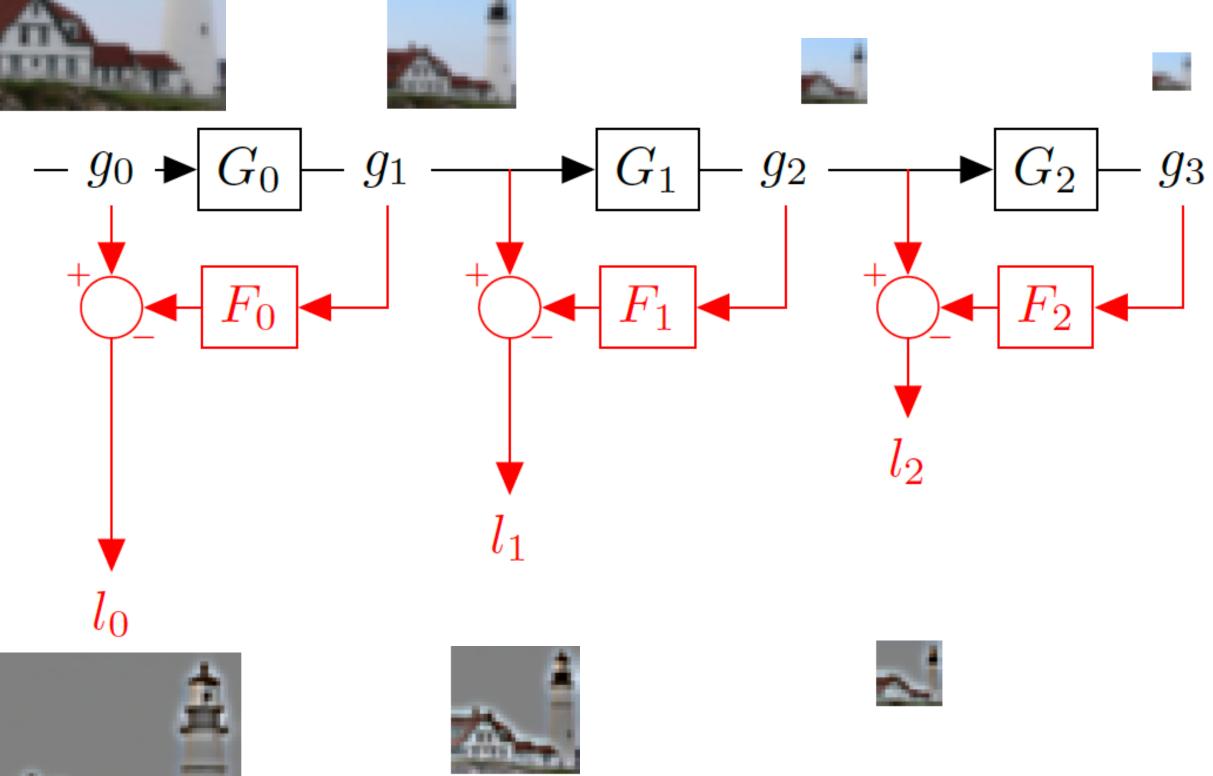


Gaussian pyramid









Gaussian pyramid



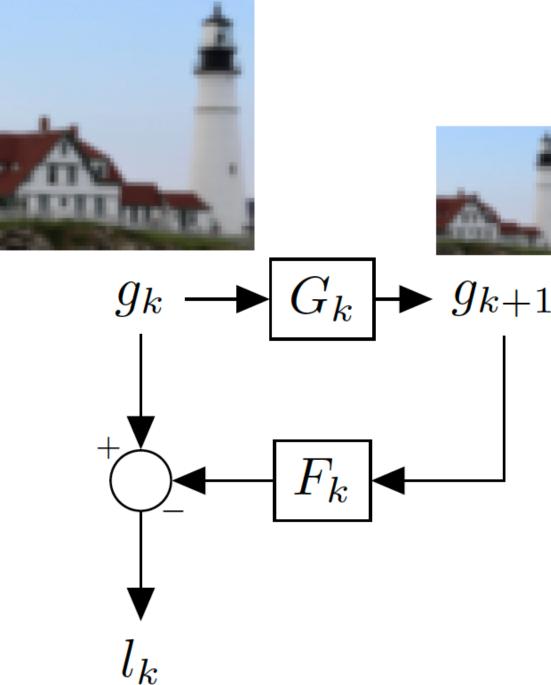
Source: Torralba, Freeman, 1201a



Blurring and downsampling:

Upsampling, blurring, and subtraction:

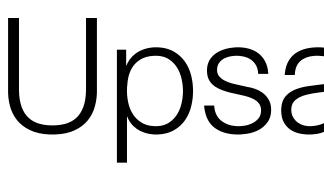
$$l_k = F_k(g_k, g_{k+1}) = ????$$



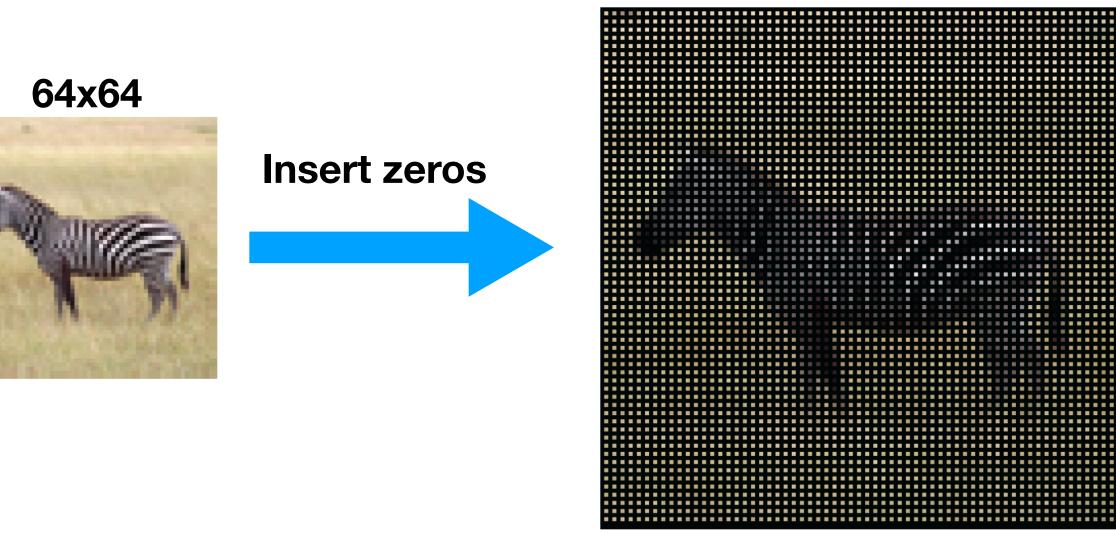


 $g_{k+1} = G_k(g_k) = \text{downsample}(\text{blur}(g_k))$

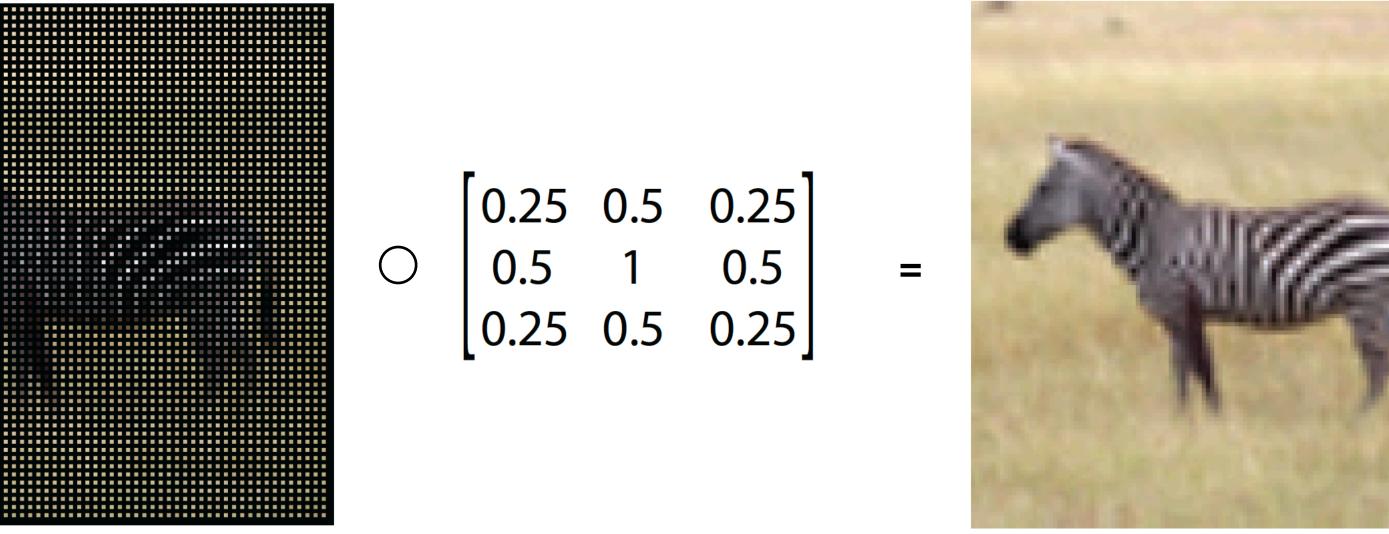




128x128



Upsampling



Source: Torralba, Freeman, Isala

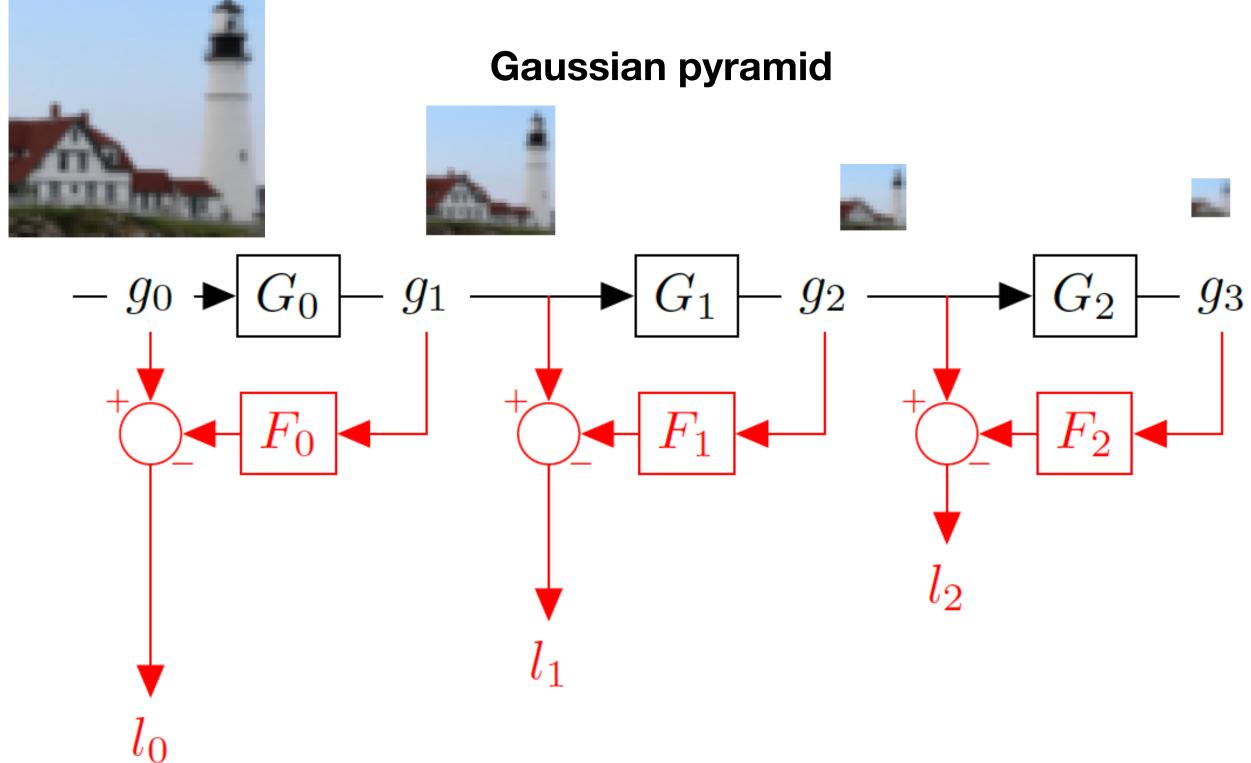
128x128















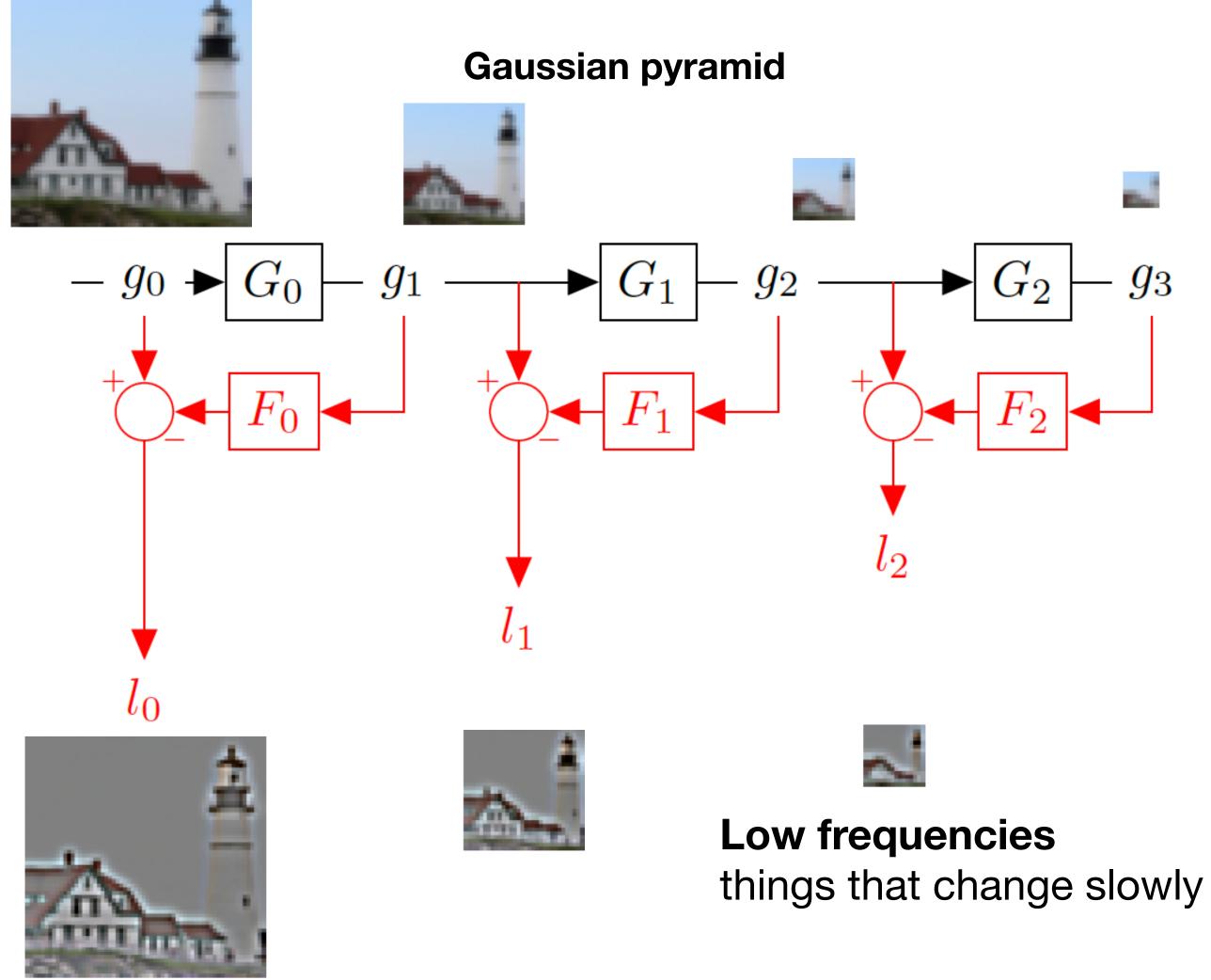


Laplacian pyramid

Source: Torralba, Freeman, 1831a







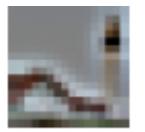
High frequencies things that change fast





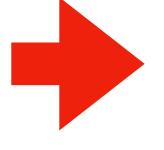


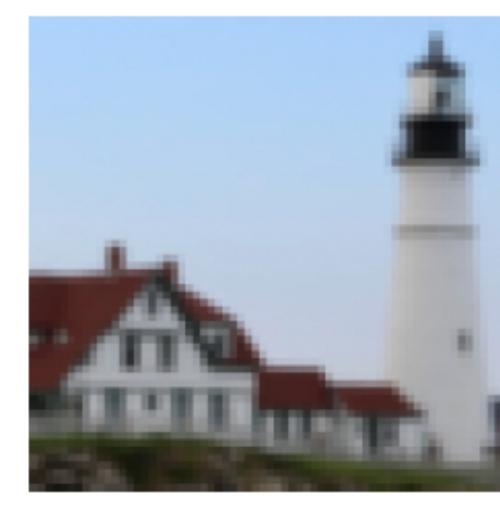
Laplacian pyramid





Gaussian residual



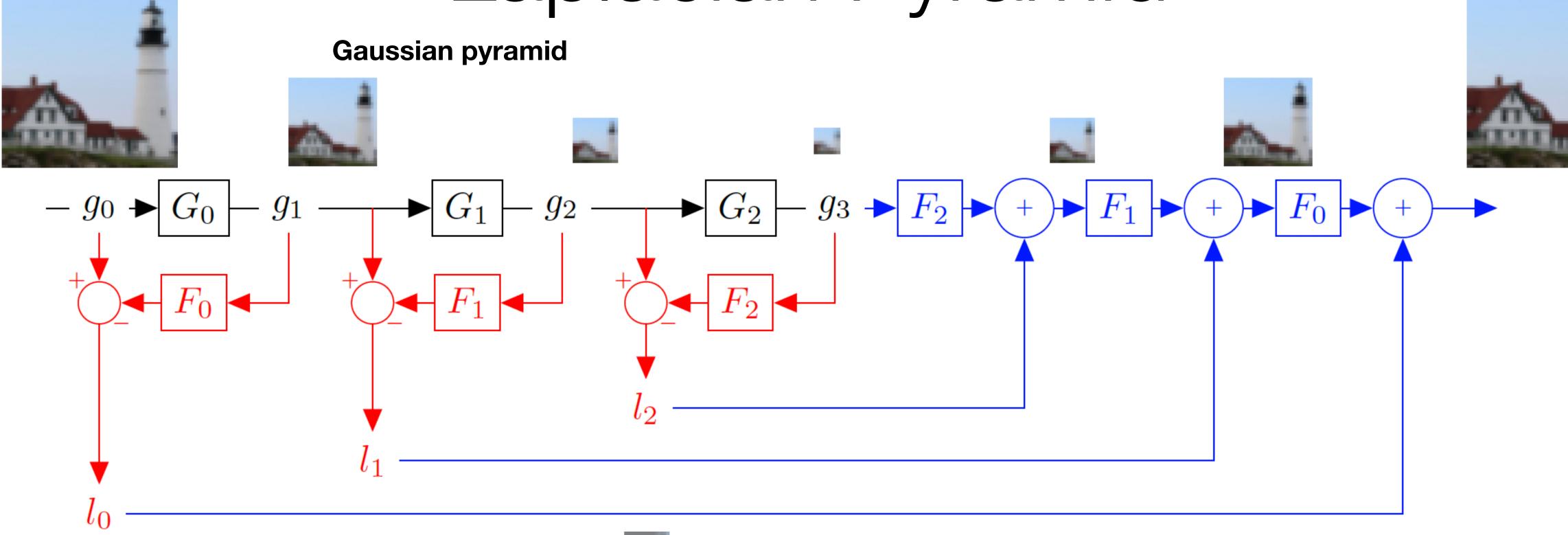


Can we invert the Laplacian Pyramid?

Source: Torralba, Freeman, 1851a











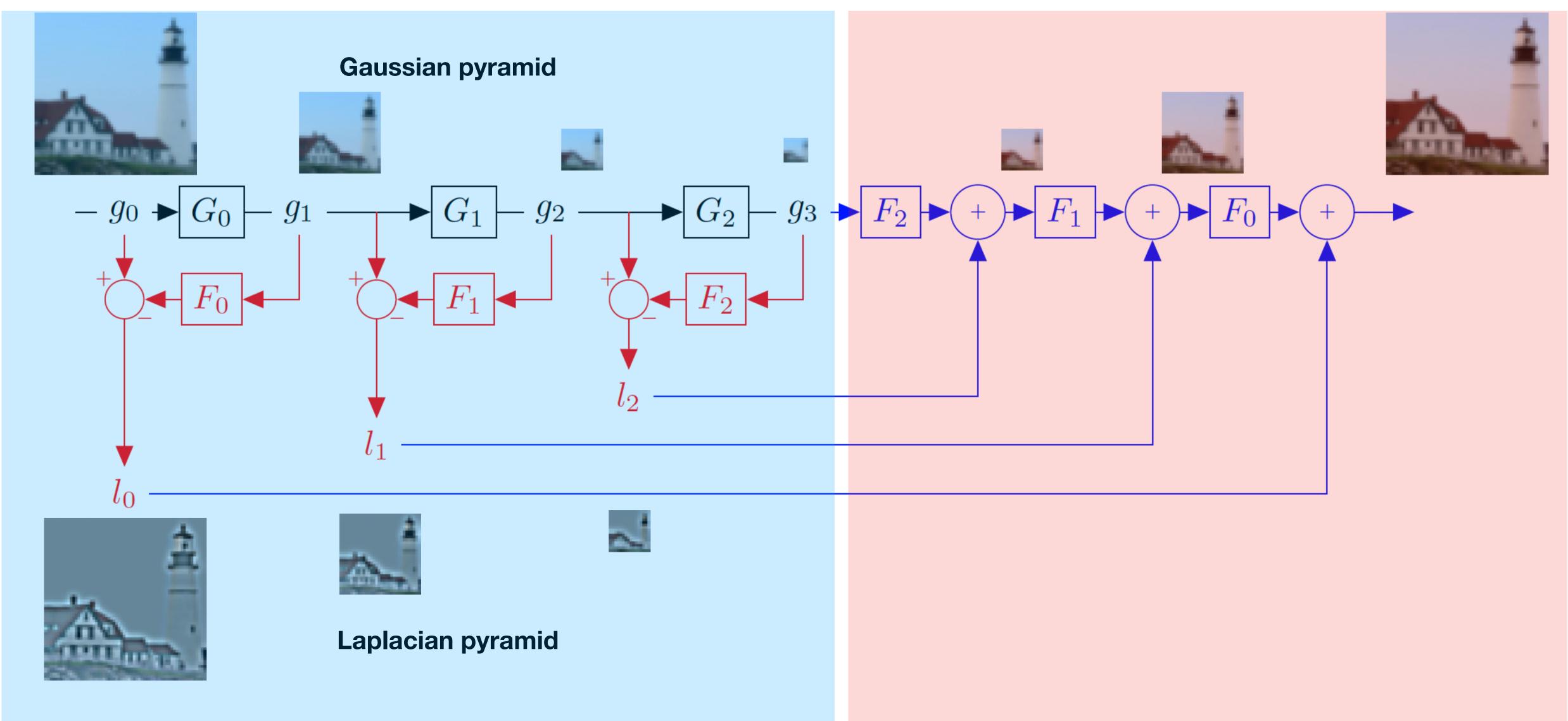


Laplacian pyramid

Source: Torralba, Freeman, 1861a







Analysis/Encoder

Synthesis/Decoder



Laplacian pyramid applications

- Texture synthesis
- Image compression
- Noise removal
- Computing image "keypoints"

Source: Torralba, Freeman, 1281a







Image Blending









Image Blending

Source: Torralba, Freeman, 1801a





JA

JB

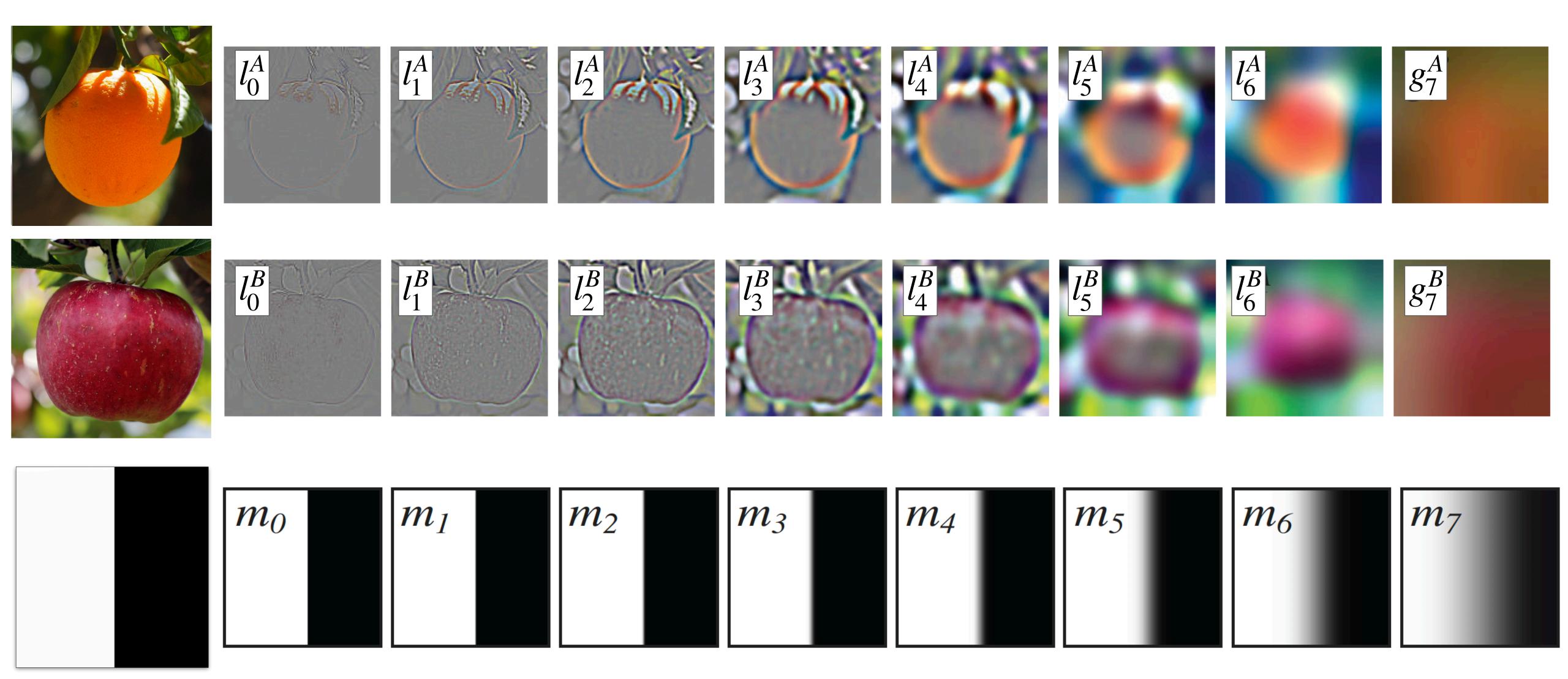
Image Blending



$I = m * I^A + (1 - m) * I^B$



Image Blending with the Laplacian Pyramid



 $l_k = l_k^A * m_k + l_k^B * (1 - m_k)$

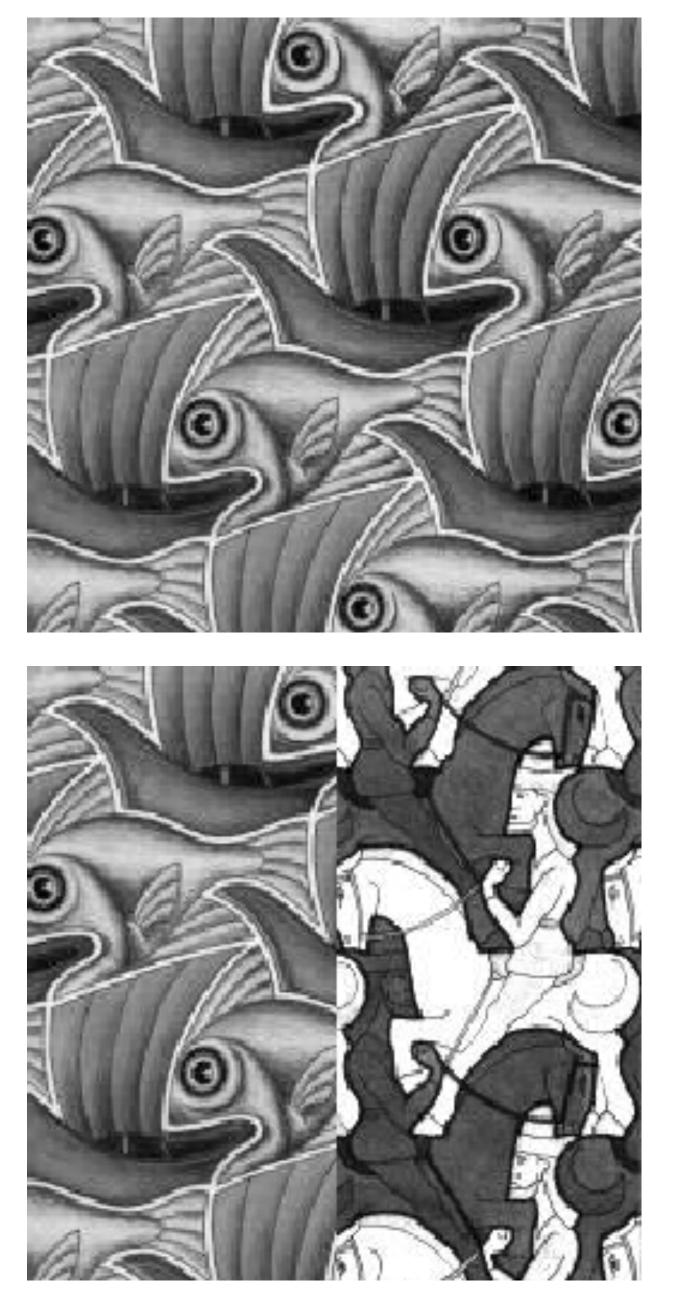


Image Blending with the Laplacian Pyramid

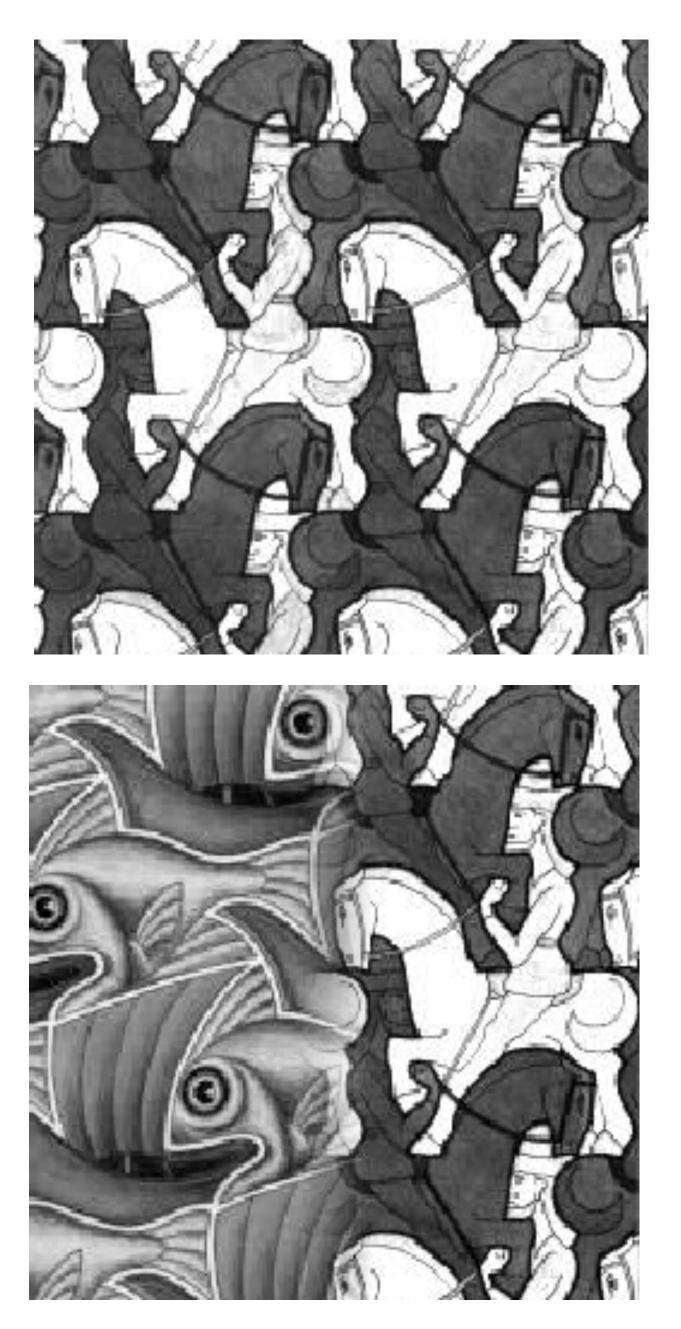








Simple blend



With Laplacian pyr.

Source: A. Efros





Photo credit: Chris Cameron





Image Blending (PS2 problem)

- Build Laplacian pyramid for both images: L_A , L_B
- Build Gaussian pyramid for mask: G
- Build a combined Laplacian pyramid
- Collapse L to obtain the blended image

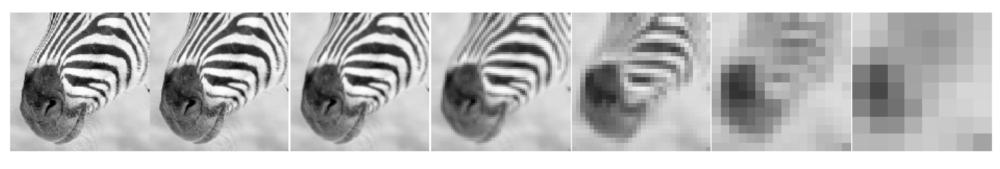




Source: Torralba, Freeman, 1361a





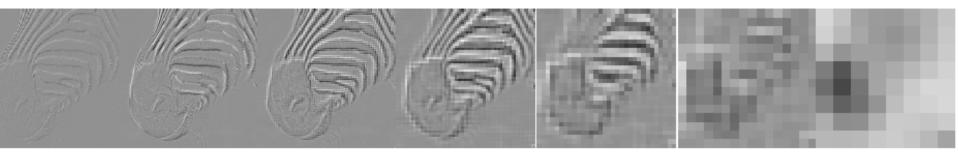


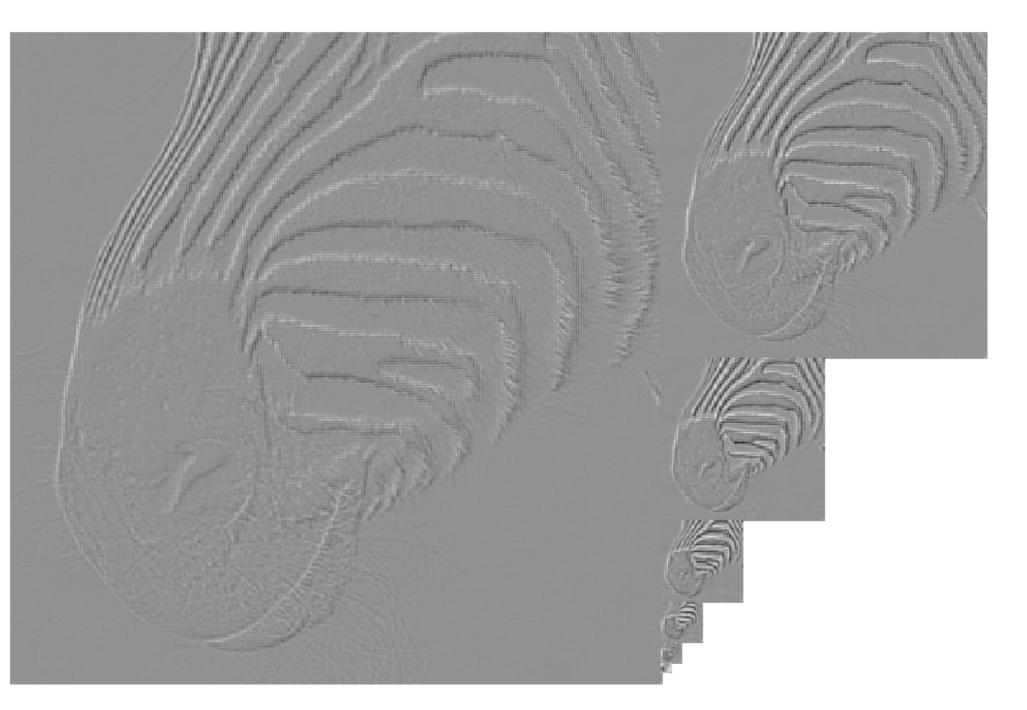


Gaussian Pyramid

And many more: steerable filters, wavelets, ... (and later) convolutional networks!

Image pyramids

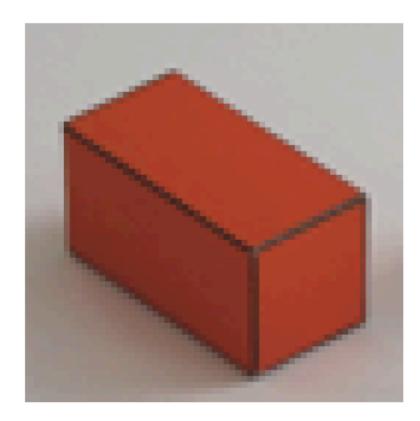




_aplacian Pyramid



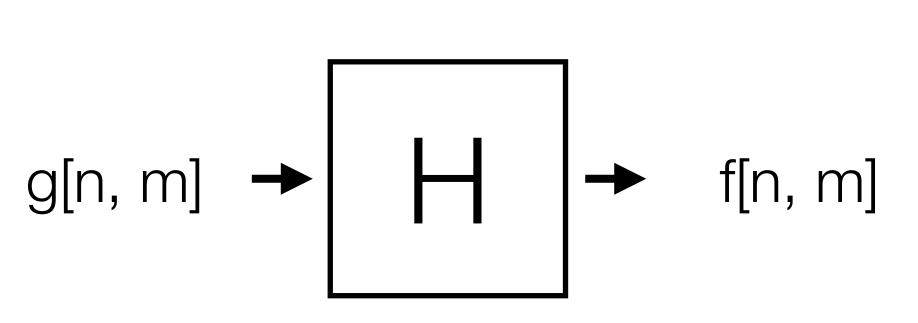
Are Gaussian/Laplacian pyramids linear filters?

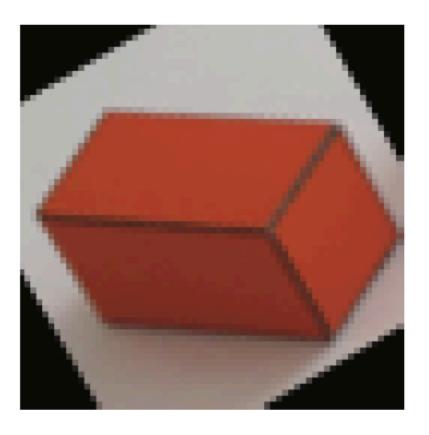




Recall: linear filter

Equivalent to matrix multiplication with some matrix *H*:

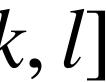


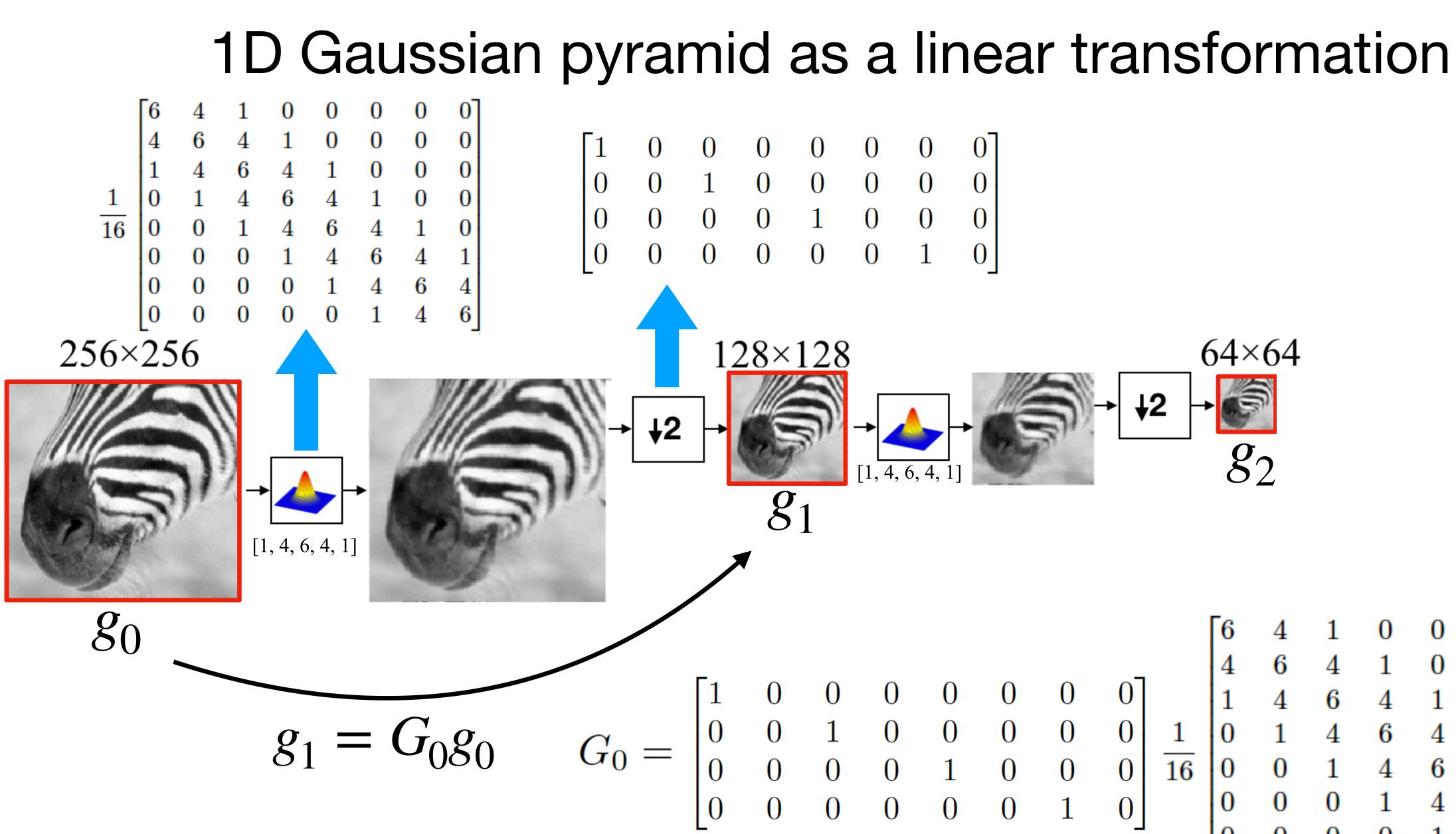


N-1 N-1 $f[n,m] = \sum \sum h[n,m,k,l]g[k,l]$ k=0 l=0

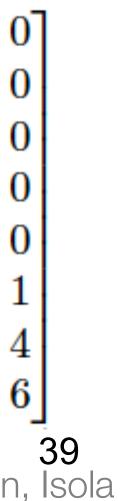
f = Hg

38



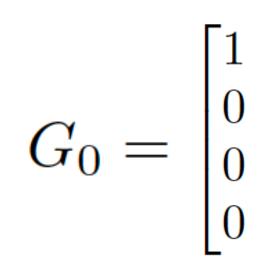


$$\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \\ \end{bmatrix}$$

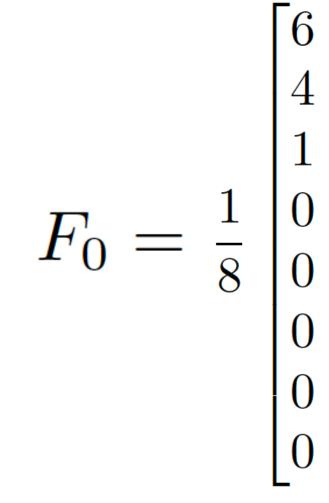


Laplacian Pyramid

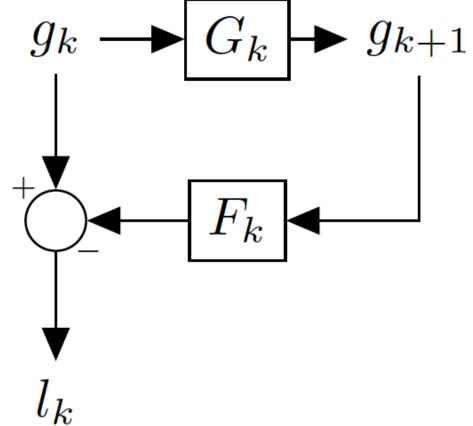
Blurring and downsar



Upsampling and blur

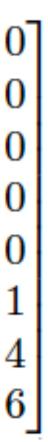






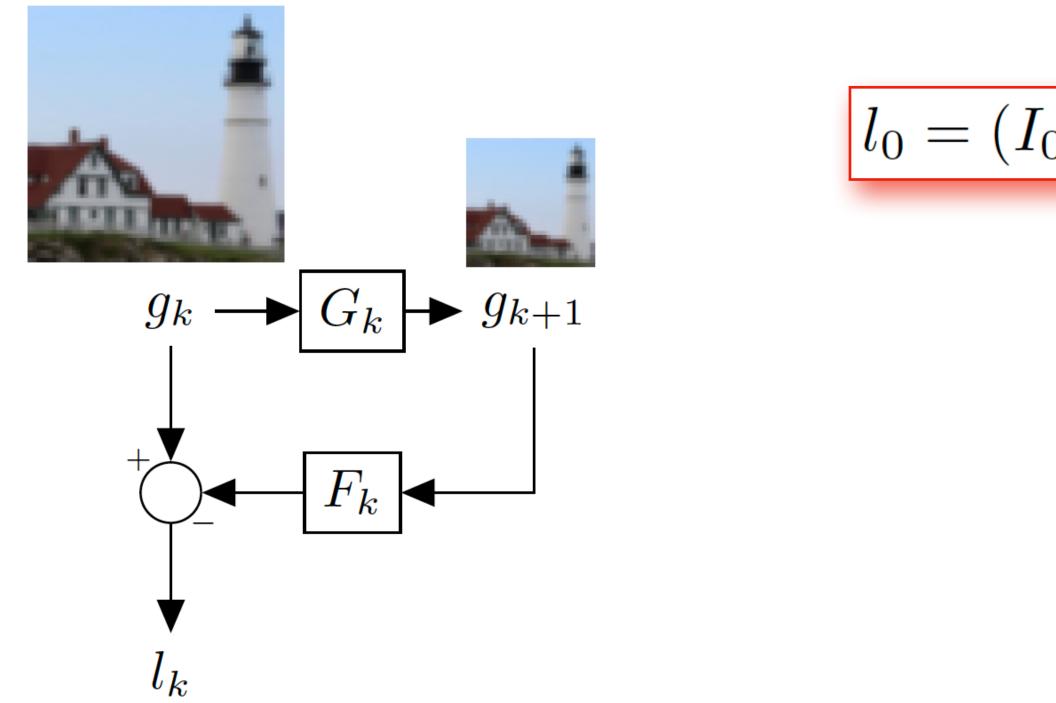


mpling:								$\begin{bmatrix} 6\\4 \end{bmatrix}$	$\frac{4}{6}$	1	0	0 0	0 0	0 0
0	0	0	0	0	0	0		$ ^{4}_{1}$	4	4 6	4	1	0	0
0	1	0	0	0	0	0	1	0	1	4	6	4	1	0
0	0	0	1	0	0	0	16	0	0	1	4	6	4	1
0	0	0	0	0	1	0		0	0	0	1	4	6	4
Lownsampling by 2)							0	0	0	0	1	4	6	
	X		•	0,	,			0	0	0	0	0	1	4
rin	g:										(blur)		
	4	1	0	0	0	0	0	Γ1	0	0	()]		
	6	4	1	0	0	0	0	0	0	0	()		
	4	6	4	1	0	0	0	0	1	0	()		
	1	4	6	4	1	0	0	0	0	0	()		
	0	1	4	6	4	1	0	0	0	1	()		
	0	0	1	4	6	4	1	0	0	0	()		
	0	0	0	1	4	6	4	0	0	0	-	L		
	0	0	0	0	1	4	6	0	0	0	()		
			(blur) (Upsampling by 2)											
			-		-	-								
$l_0 = (I_0 - F_0 G_0) g_0$ Source: Torralba, Free												eem		
			L											





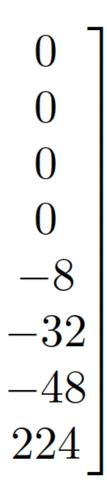
The Laplacian Pyramid





$l_0 = (I_0 - F_0 G_0)g_0$

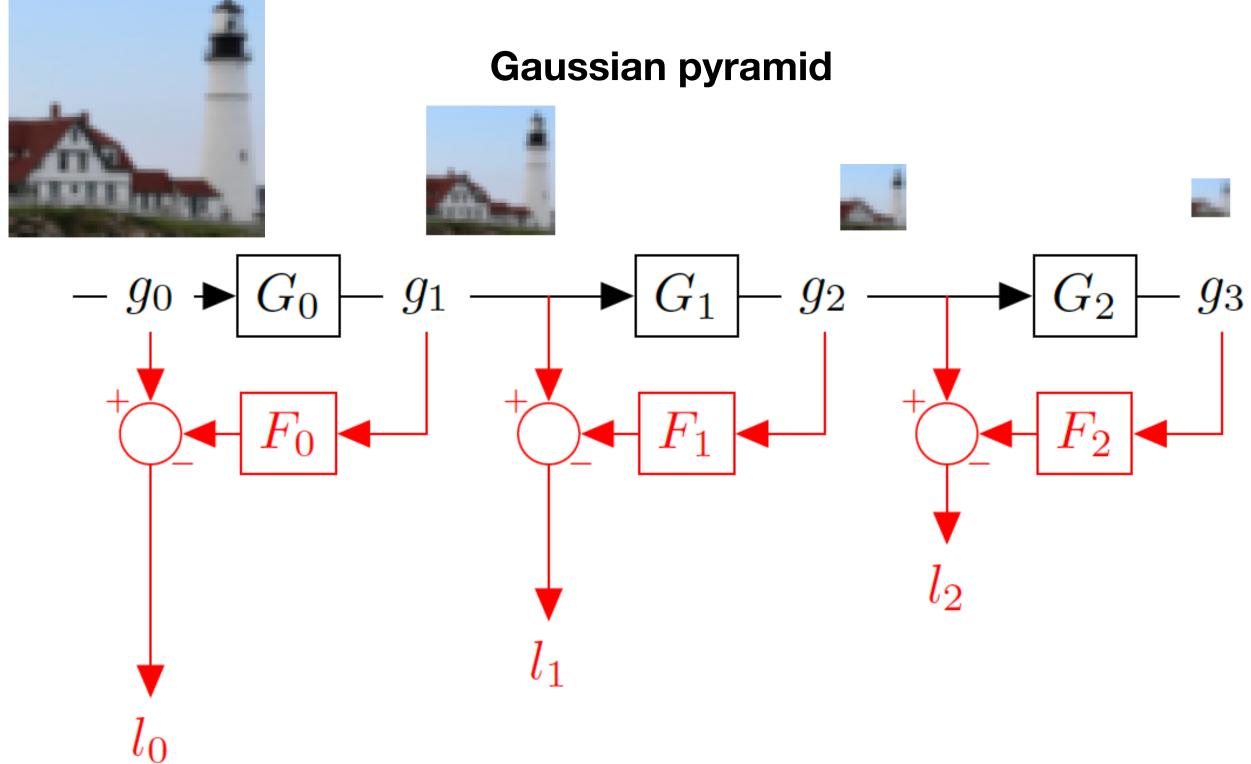
	1 82	-56	-24	-8	-2	0	0	
	-56	192	-56	-32	-8	0	0	
	-24	-56	180	-56	-24	-8	-2	
		-32						
-256		-8						
	0	0	-8	-32	-56	192	-56	-
	0	0	-2	-8	-24	-56	182	-
	0	0	0	0	-8	-32	-48	







Laplacian Pyramid

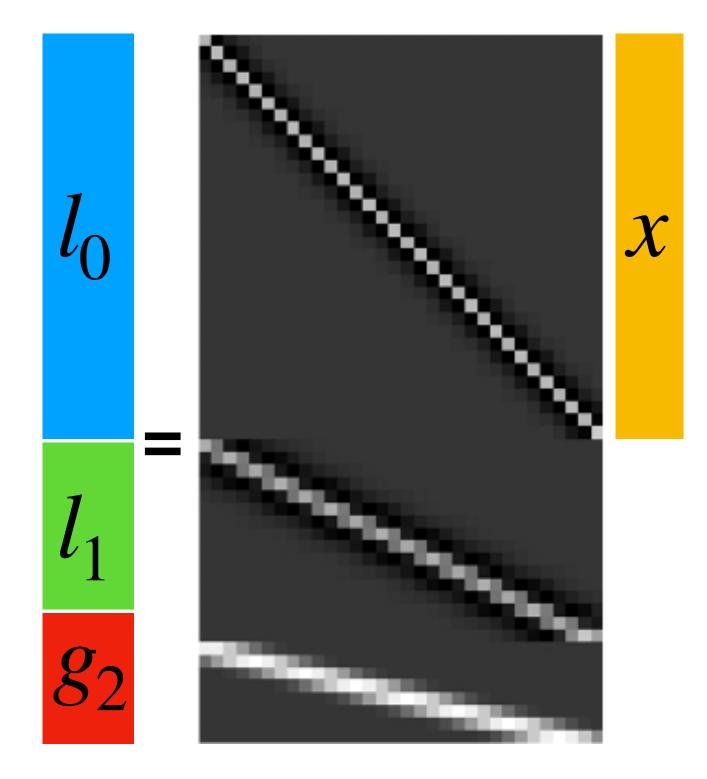








Laplacian pyramid



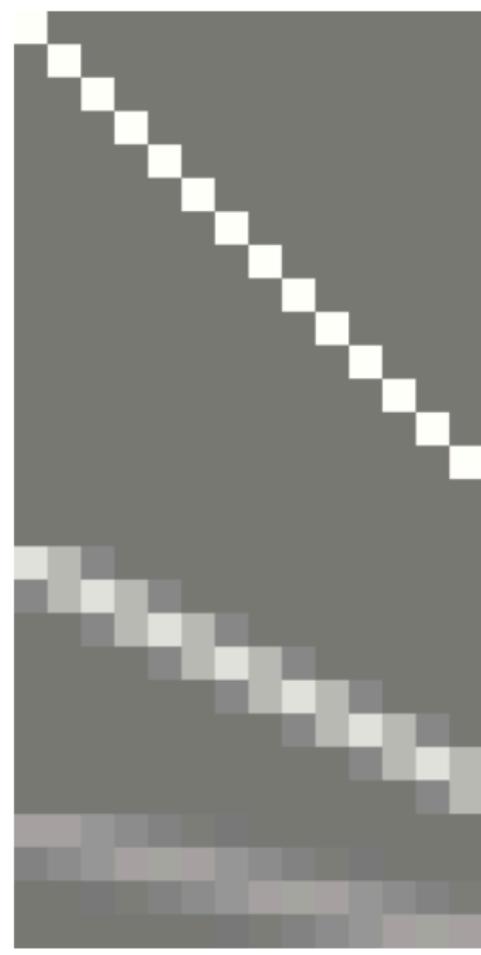
Source: Torralba, Freeman, 1981a





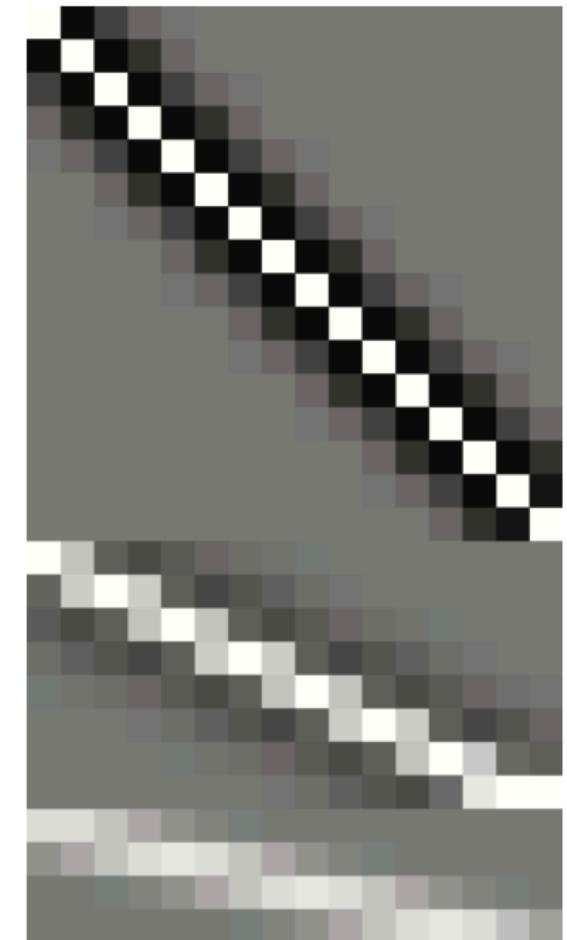
Linear Image Transforms





Gaussian pyramid

28x16



Laplacian pyramid



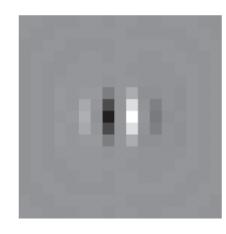
One other pyramid: the Steerable Pyramid





Steerable Pyramid

Oriented gradient

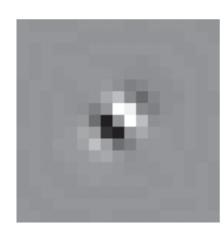


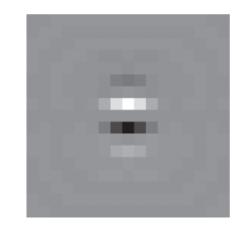


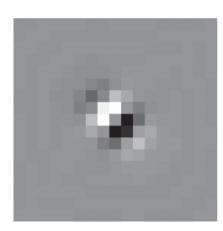
160 ant (0)8

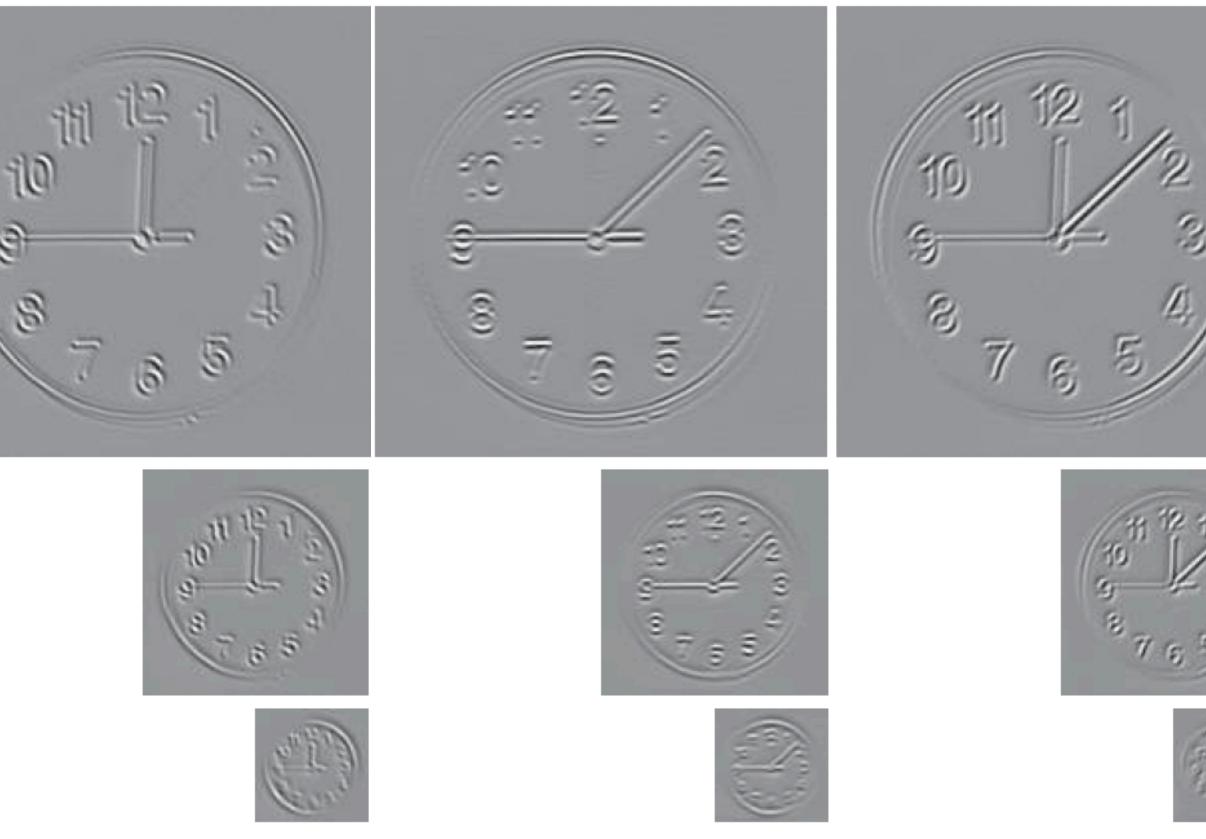














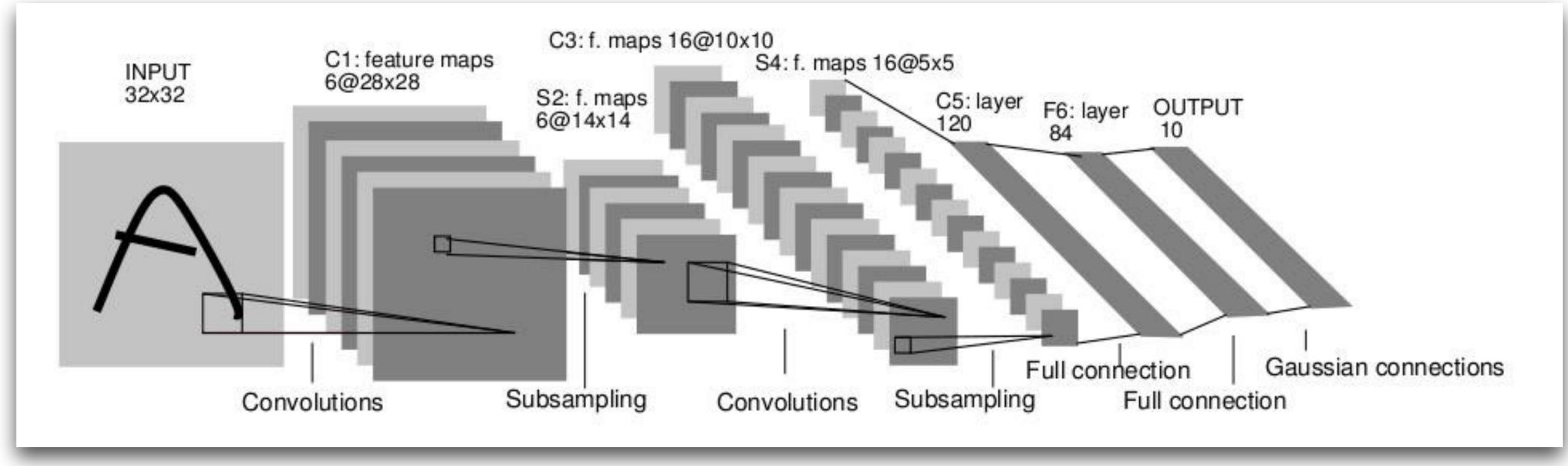






For more powerful nonlinear models...

 Later, we'll study a powerful nonlinear model, convolutional neural networks • Their main building blocks are linear filters.



[LeCun et al. 1989]

• Image pyramids Image statistics • Texture synthesis

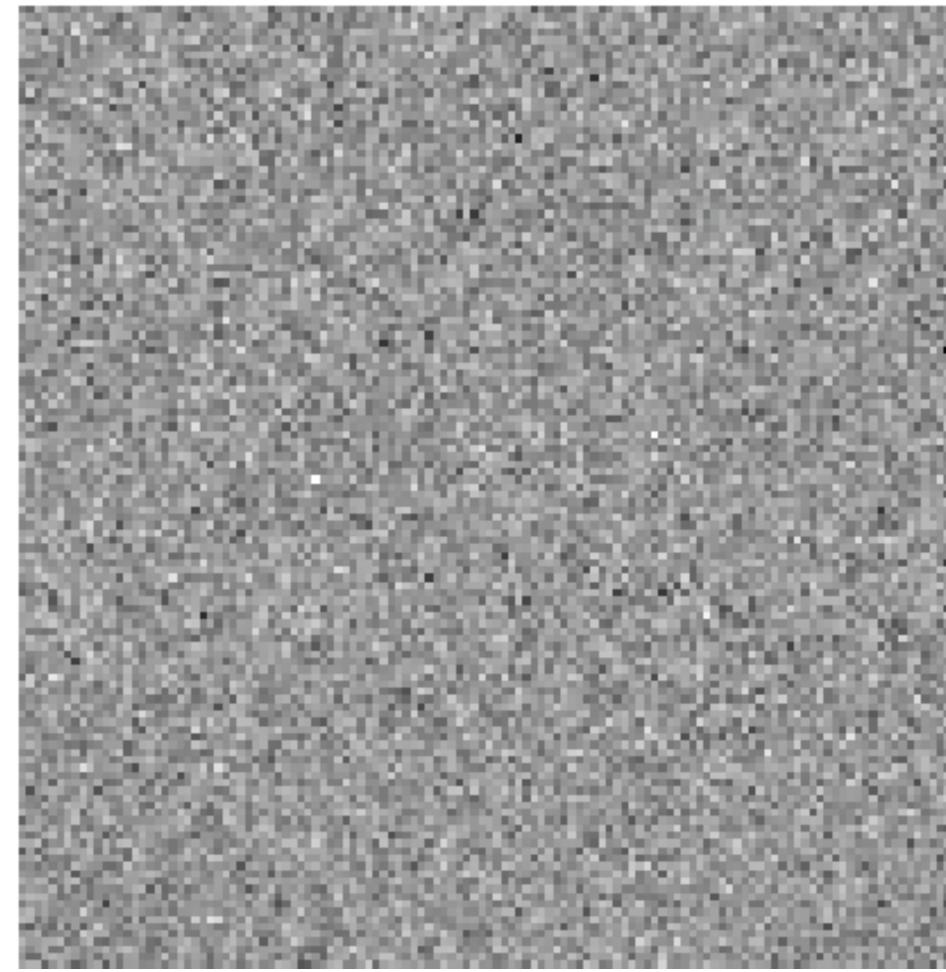
Today

47

What makes an image "natural"?



Natural image



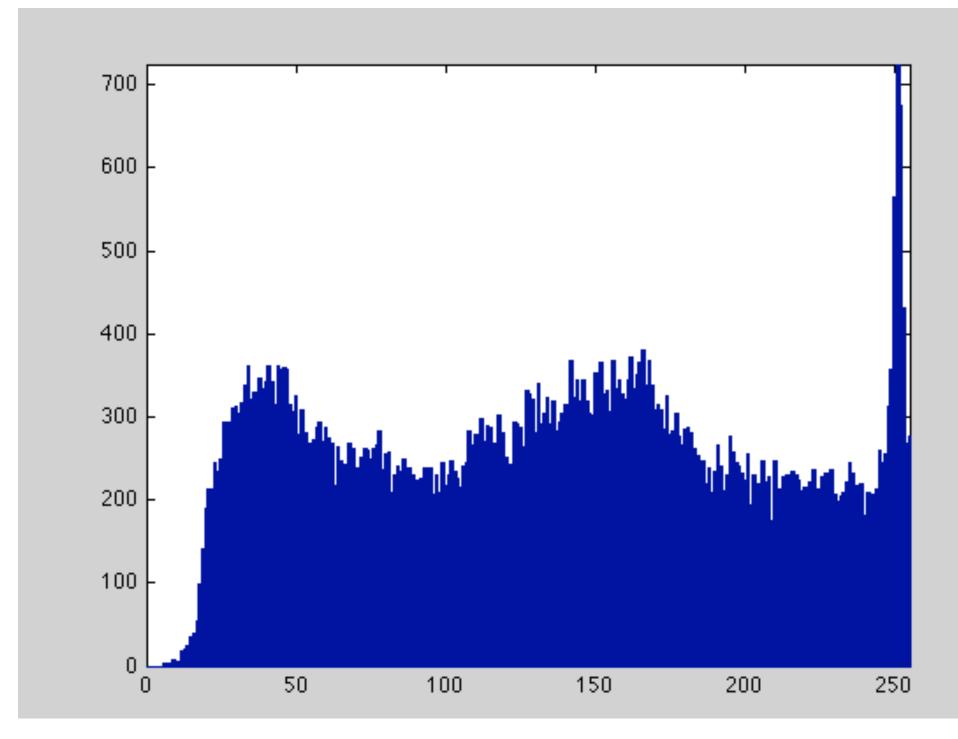
"Fake" image



Is it the distribution of pixel intensities?



No real structure here...

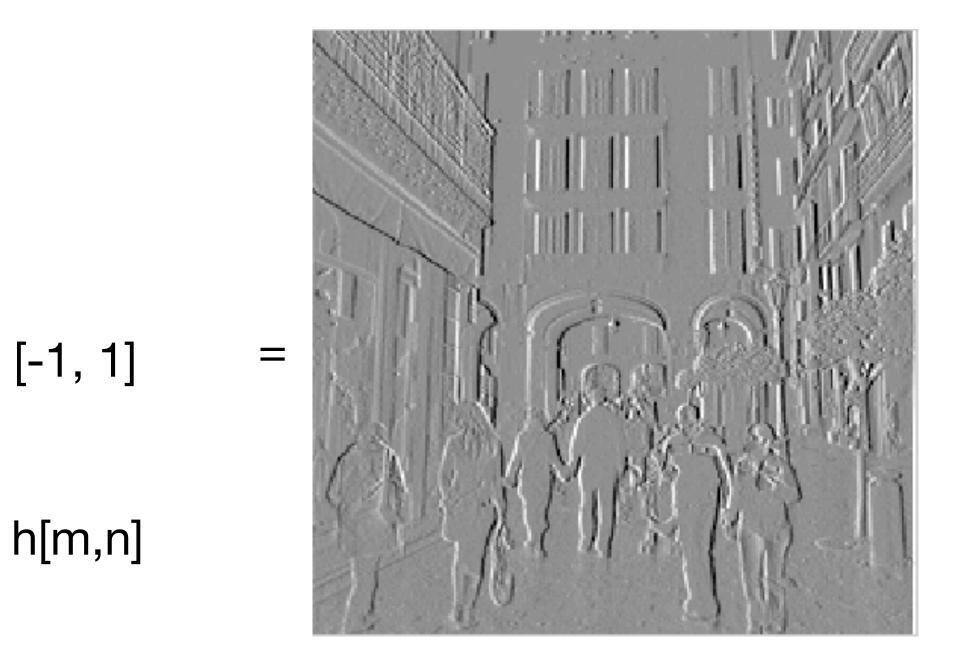


Intensity histogram

What about gradients?



g[m,n]



f[m,n]

50 Source: Torralba, Freeman, Isola

nan, Isola

What about gradients?

[-1, 1]⊺

h[m,n]

=



g[m,n]

f[m,n]

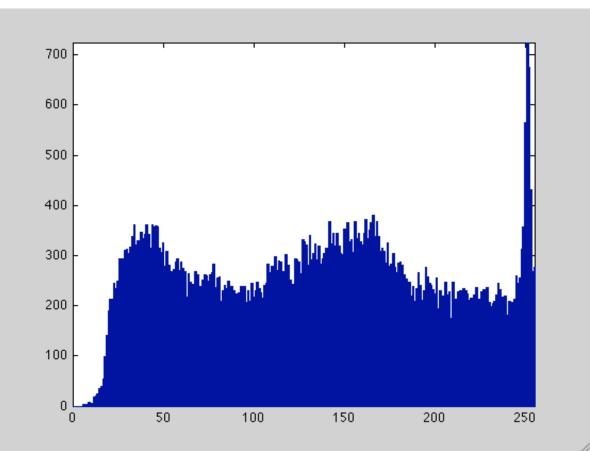
51 Source: Torralba, Freeman, Isola

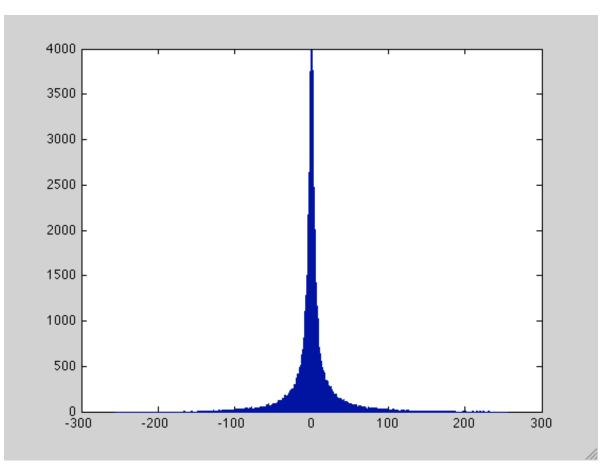
nan, Isola

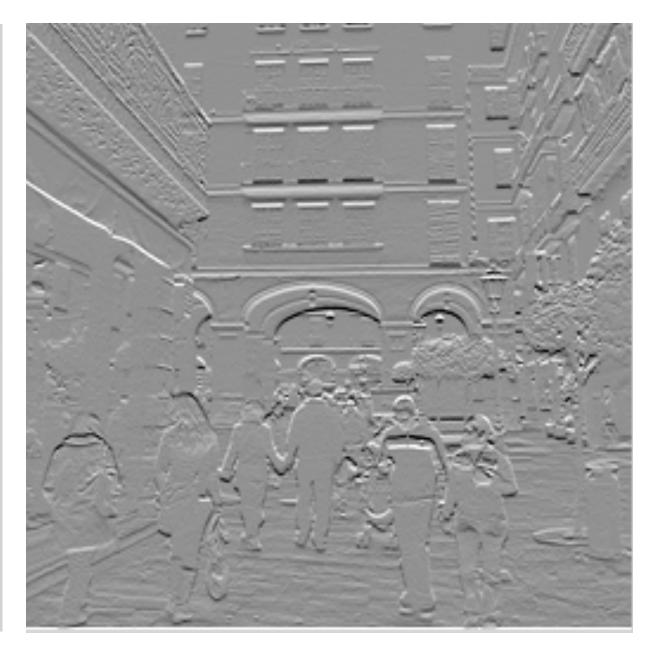
Filter response distribution is pretty consistent!

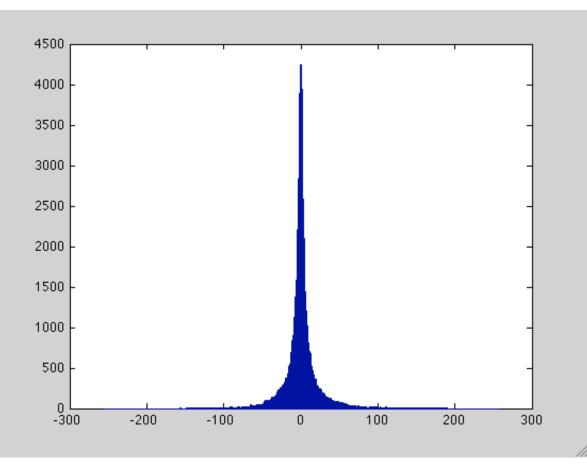






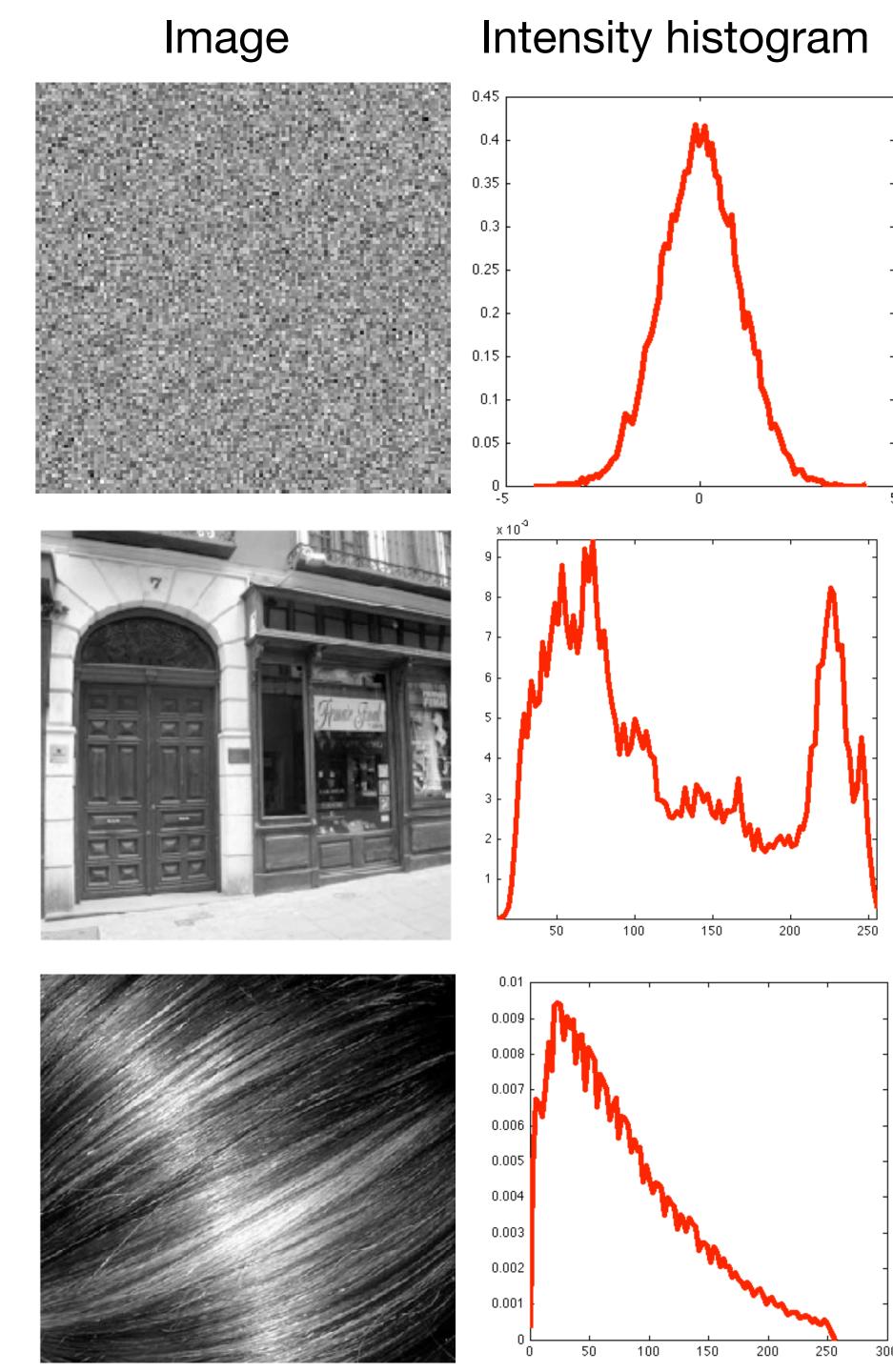




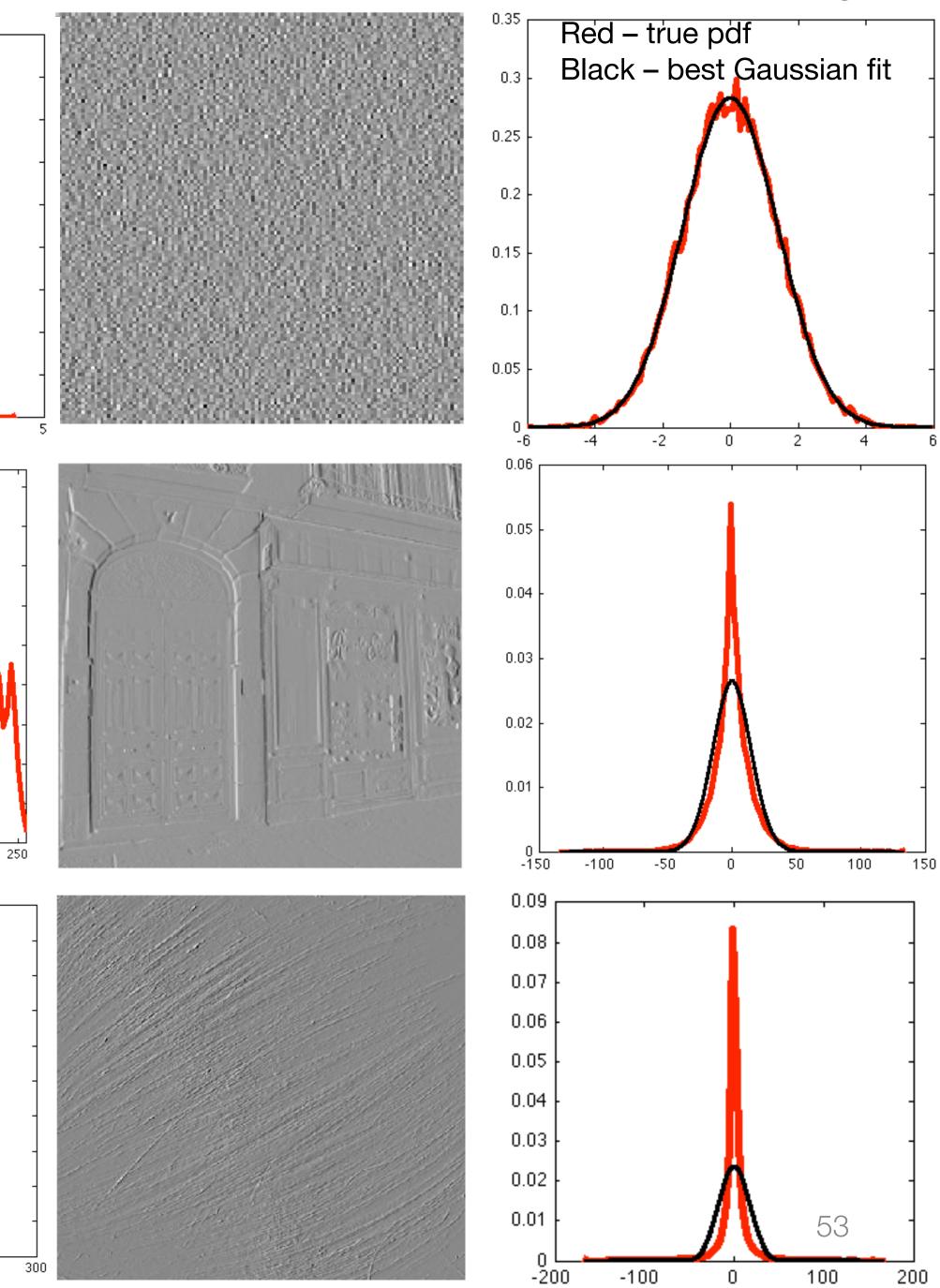


52 Source: Torralba, Freeman, Isola

nan, Isola



[1 -1] filter output [1 -1] output histogram



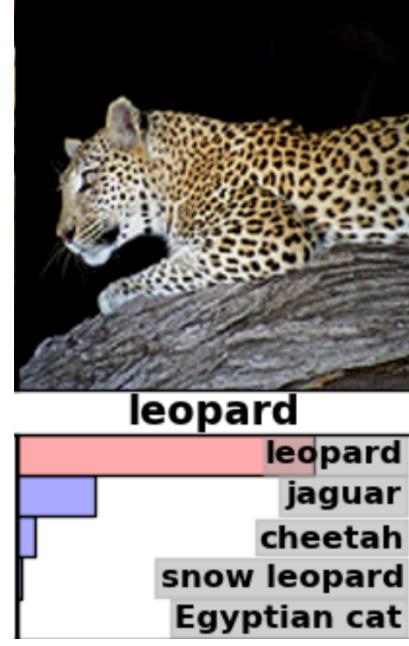
Applications of image statistics



Compression



Image restoration



Learning (later in course)



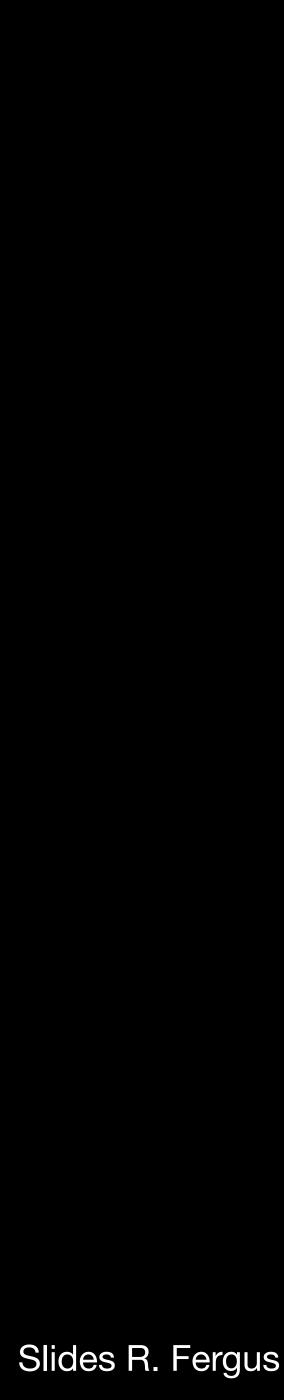




Taking a picture...

What the camera give us... How do we correct this?



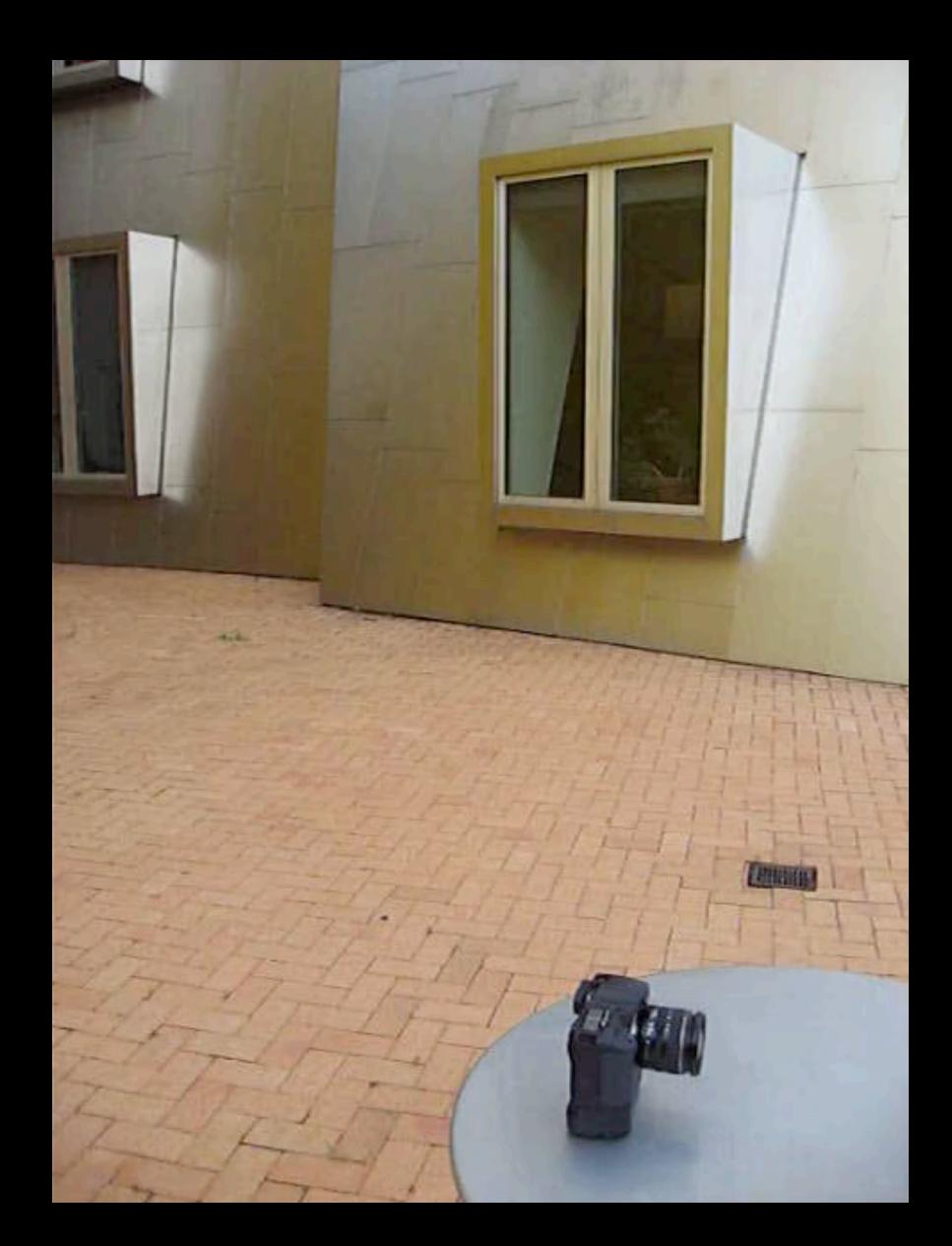


Deblurring



Slides R. Fergus

Deblurring



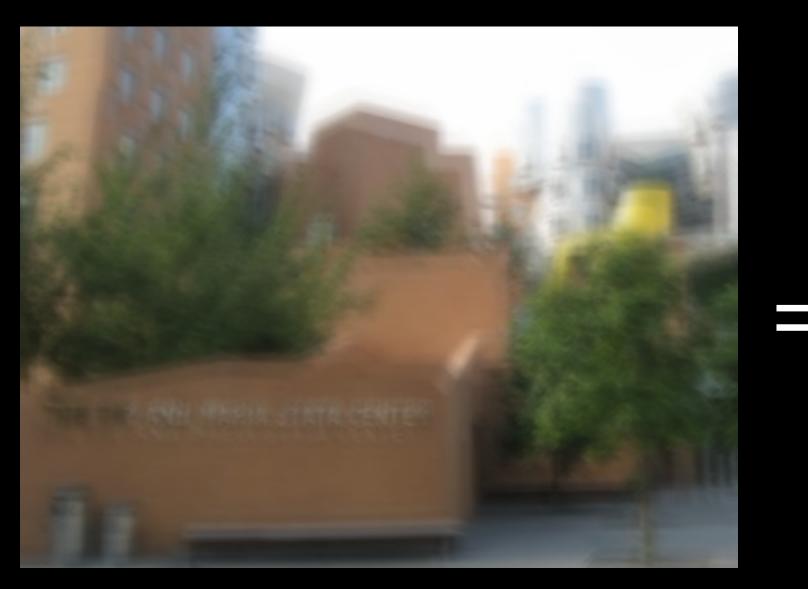
Slides R. Fergus

Deblurring



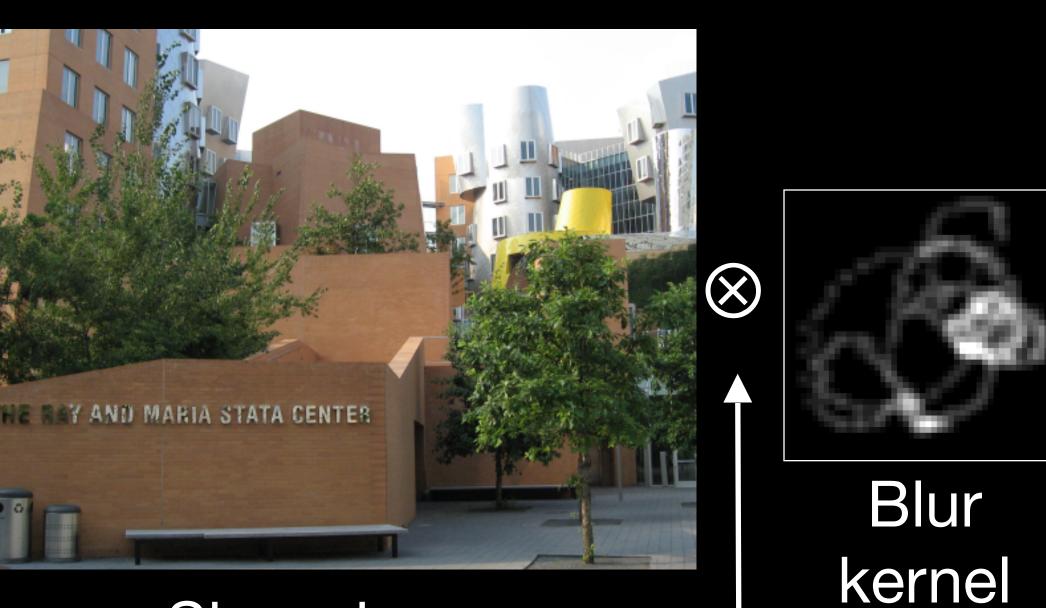
Slides R. Fergus

Image formation process



Blurry image

Input to algorithm



59

Sharp image

Desired output

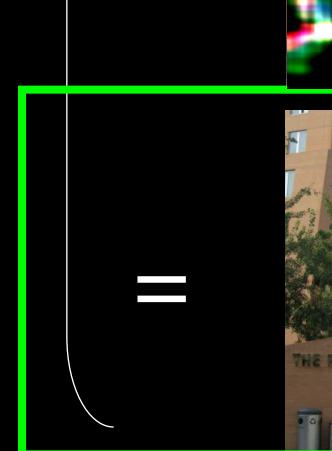
Convolution operator



Multiple possible solutions



Blurry image



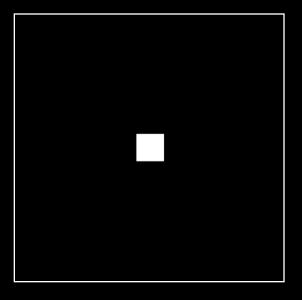
Sharp image

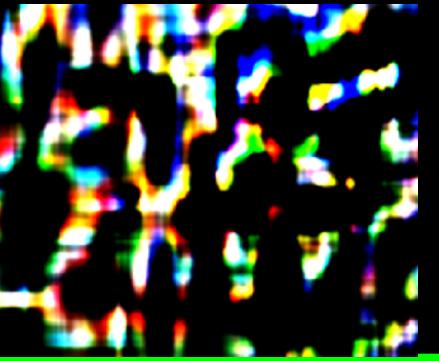


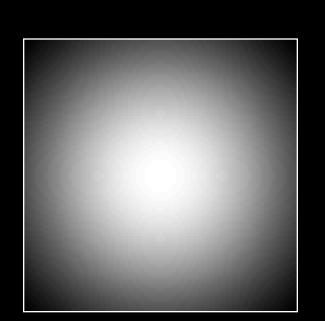


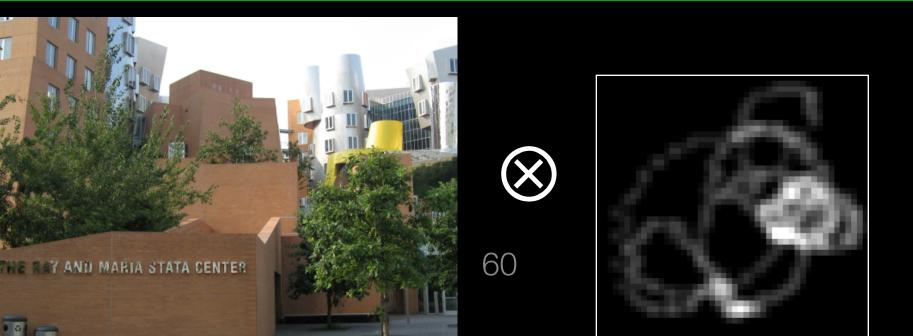
 \bigotimes











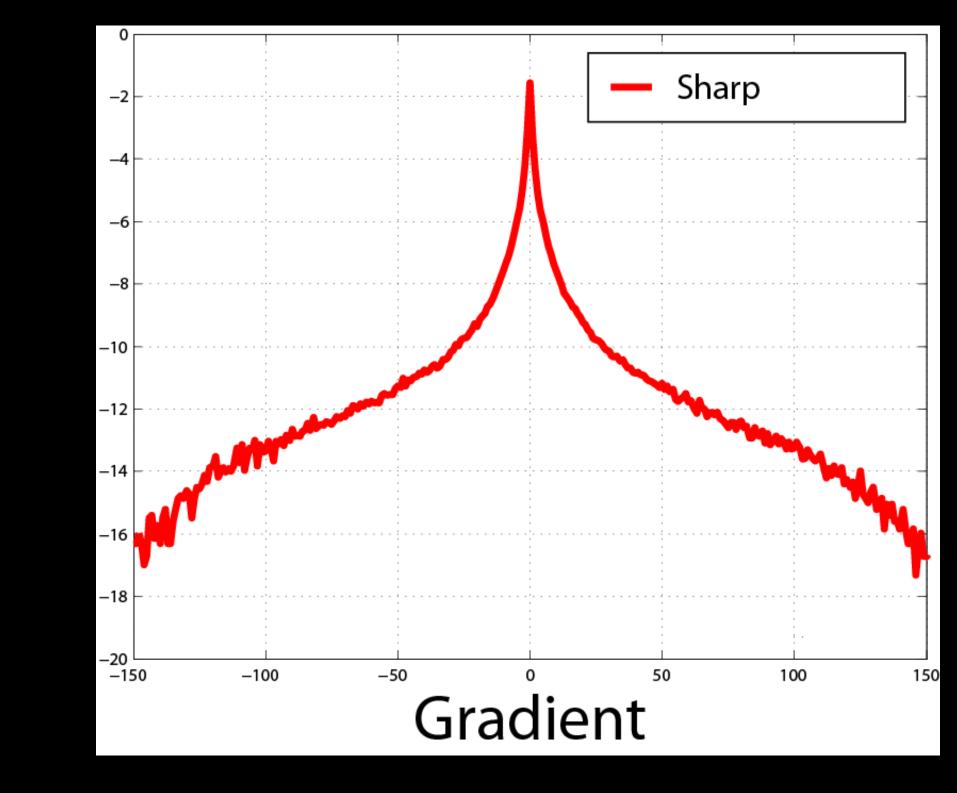
Source: R. Fergus



Natural image statistics

Characteristic distribution with heavy tails Histogram of image gradients





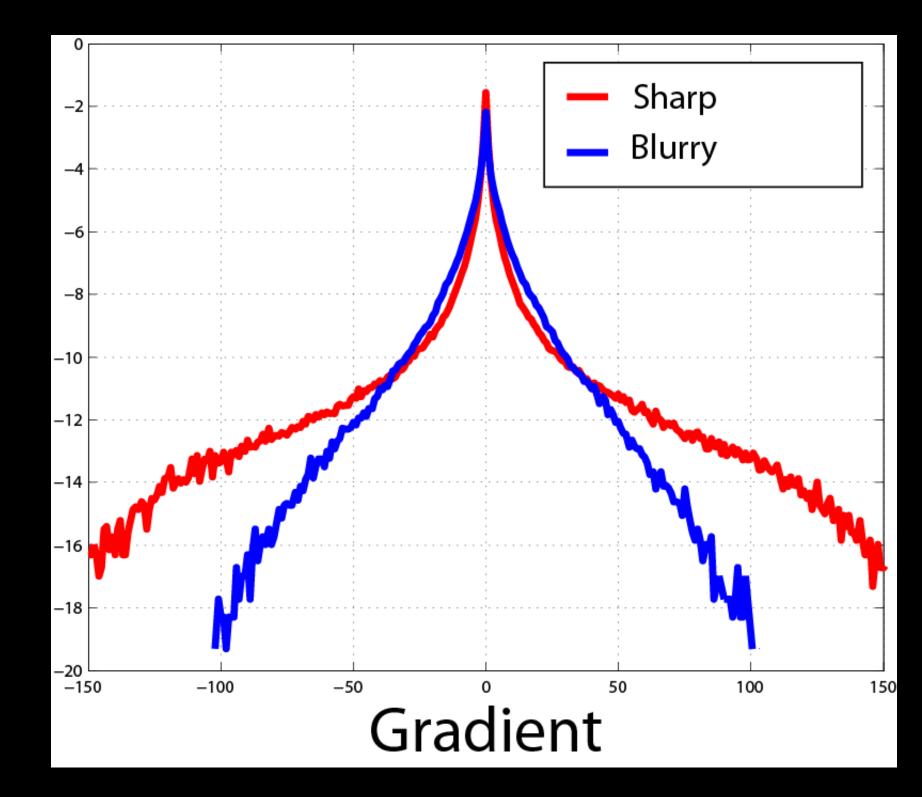




Blurry images have different statistics



Histogram of image gradients



62

Source: R. Fergus

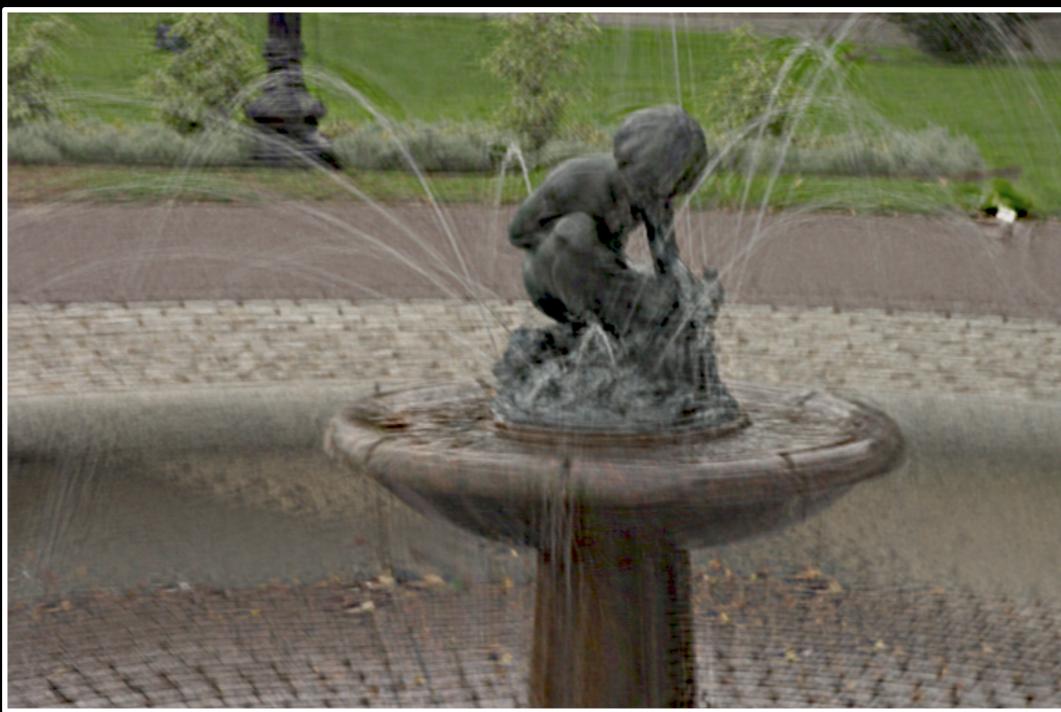


Removing motion blur



Original photograph

Solve for an image with a distribution of edge gradients that "matches" a normal image.



After matching filter distribution

Source: R. Fergus











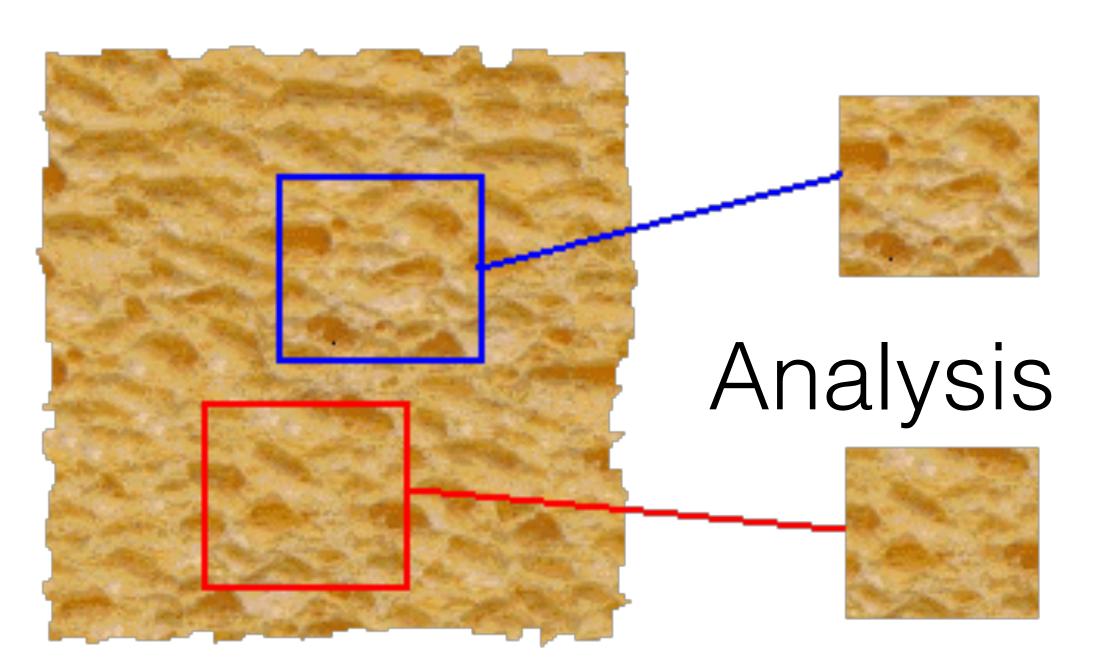


• Image pyramids Image statistics Texture synthesis



64

Texture analysis



True (infinite) texture

"stuff". Are these textures similar?

"Same" or "different"

What we'd like: are they made of the same



How do humans analyze texture?

ハマンションションションショントレンションションション 「くて下くシア シンハートア ハアシア ション ション ション バンシンシンシンションションションションションションシンシンシン バルドマンシュー・コンハンドショー・レードノンシン くこうアランシューション シーションレーション シーション - レノノレアイ シアンイレーイントアハハレンイン ノノトリアア シアアアイトアレンシャレンシインアンションアンラン ハンイ×++××+××ンーレンインレンシンノアンシンイトンン · イン* + × + × + × × ノハレビシンイビレイレーハノンレン · 」 L + x + + x x + × ハ < 1- 」 ハ 」 「 ん く ~ 」 く 」 「 ハ 」 · ハンキャ××××××× · ノハン ノハーバレン ーシーレン くくしょう - ノレントンスノー・ハンバティイハレーシアレンシーノーハハ - イトンシアイトハント - ハイト マノイトハイイナノールシム ーハ 「シンドくトーレンバンバンシンベンシーレンシンベンシー フィレハンシーレイレビンシンイントレインシン コンドハンハドトレドシントノドレイシントノン コバアンハノーバンイデブリンイベンアルシーフシートレイ くくて インファンション く ファント ショインシントインショ

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



Béla Julesz





Julesz Conjecture

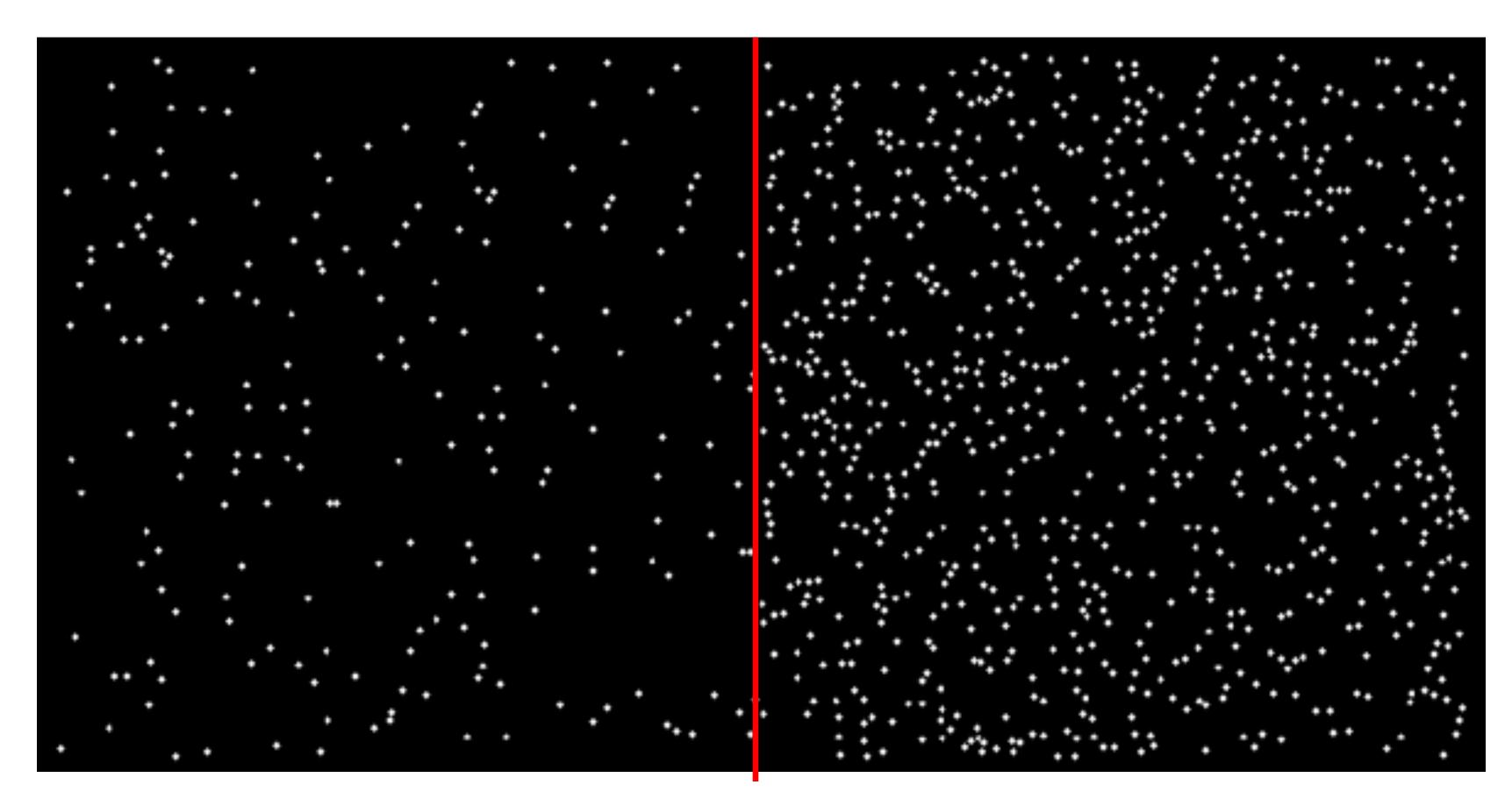
Textures cannot be spontaneously discriminated if they have the same first-order and second-order statistics and differ only in their third-order or higher-order statistics.

Somewhat imprecise (and later proved wrong)



Béla Julesz

1st order statistics differ

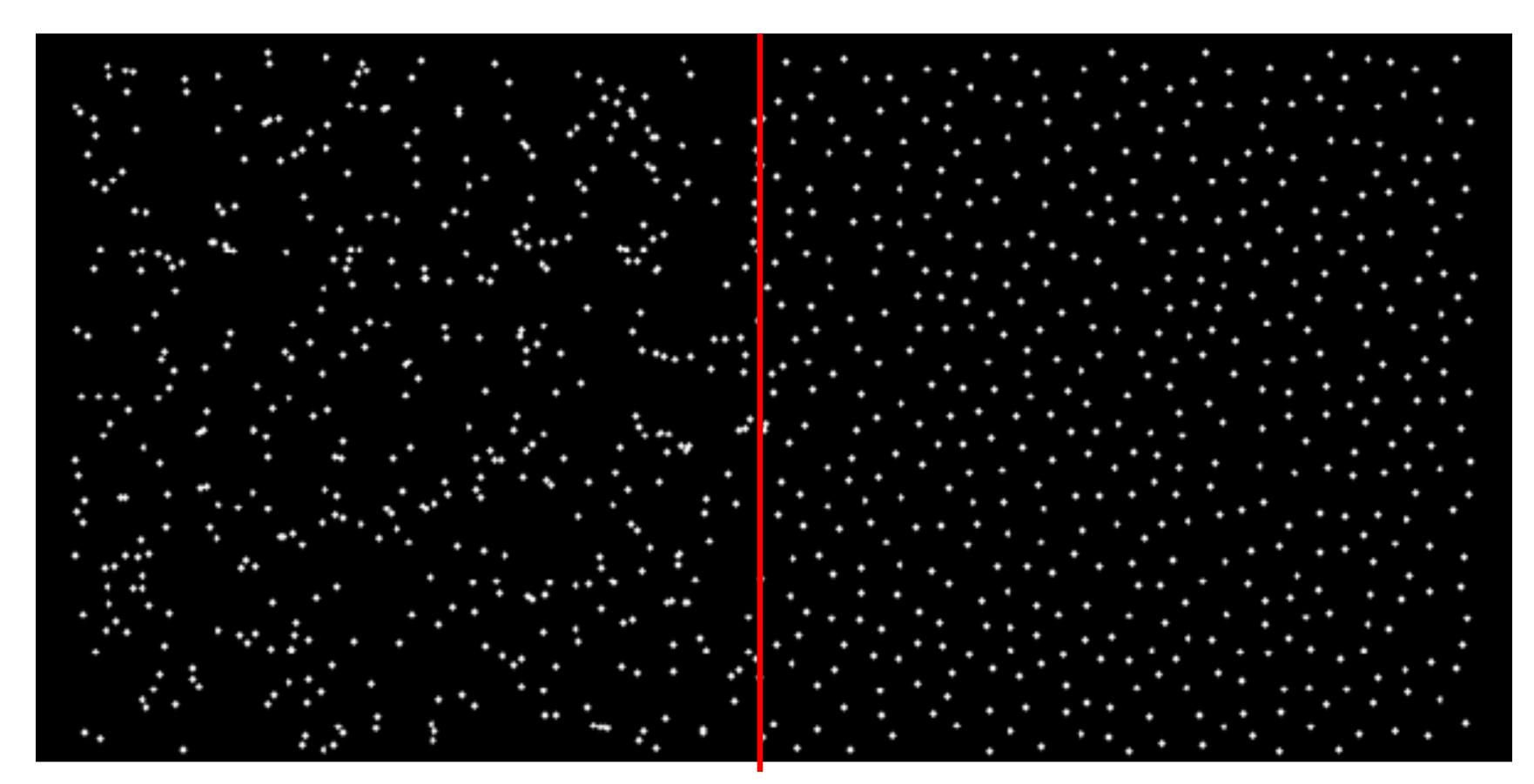


5% white

20% white



2nd order statistics differ



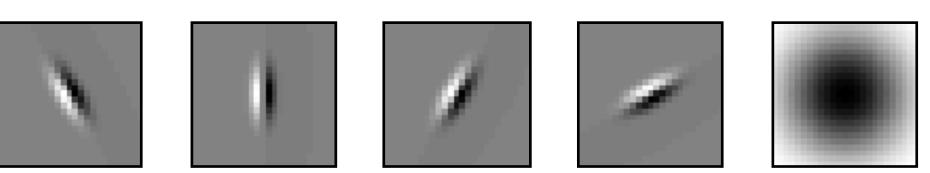
10% white



How can we represent texture in natural images?

Idea #1: Record simple statistics (e.g., mean, standard deviation) of absolute filter responses

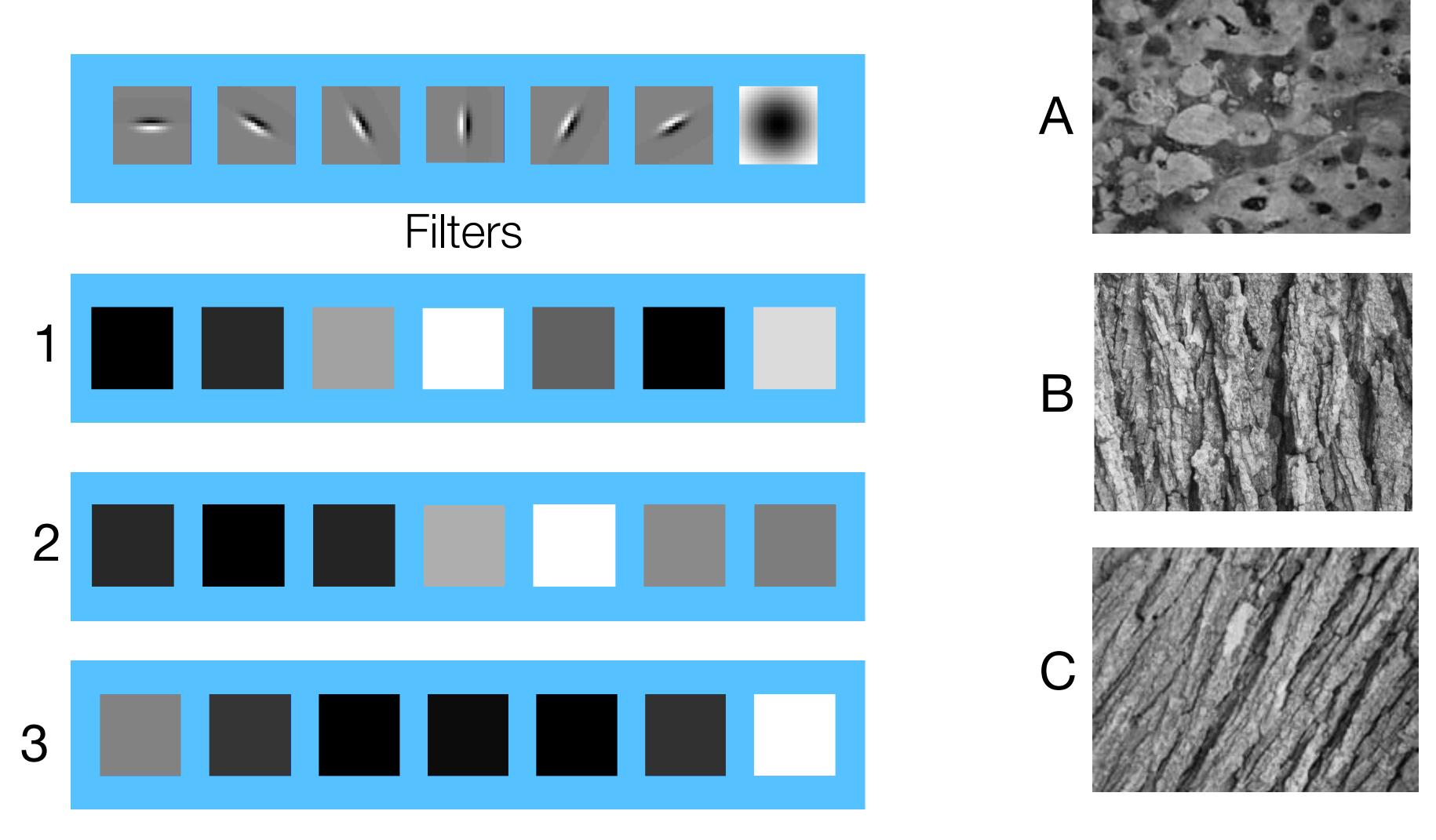








Can you match the texture to the response?



Mean abs. responses



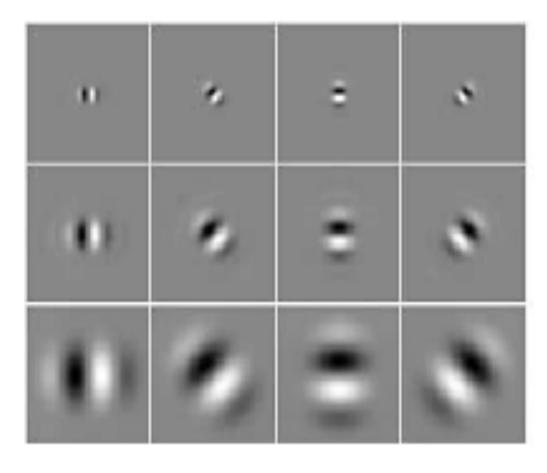
How can we represent texture?

- Generalize this to "orientation histogram"
- Idea #2: Histograms of filter responses • One histogram per filter



Steerable pyramid decomposition

Filter bank

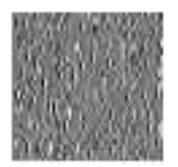






80



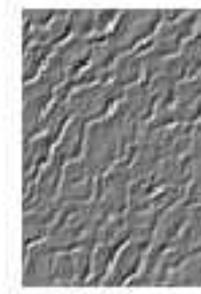


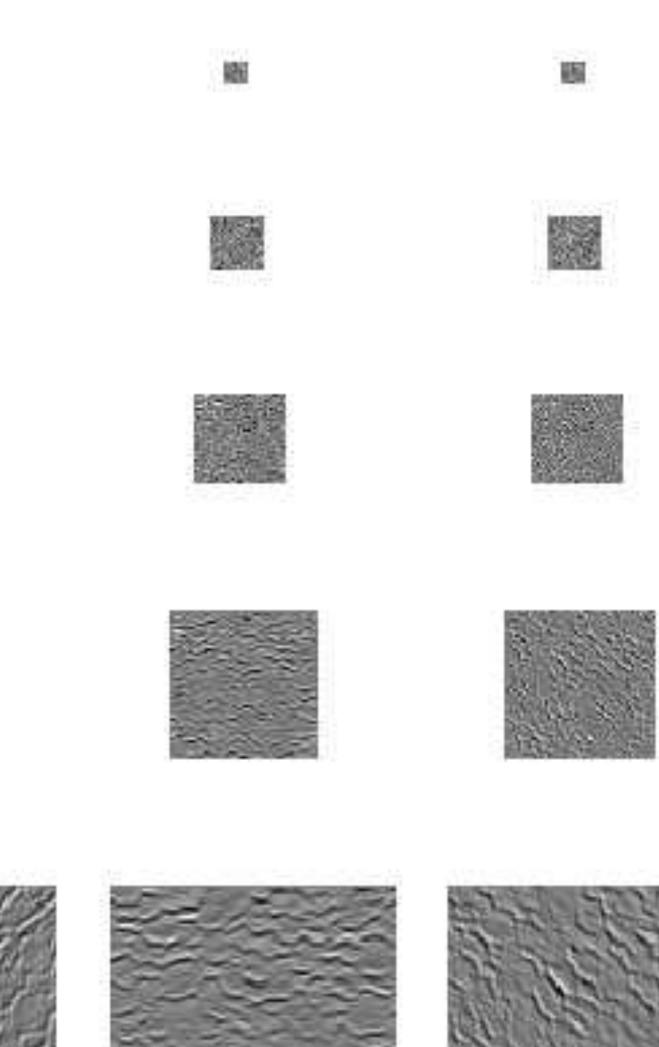


Input image



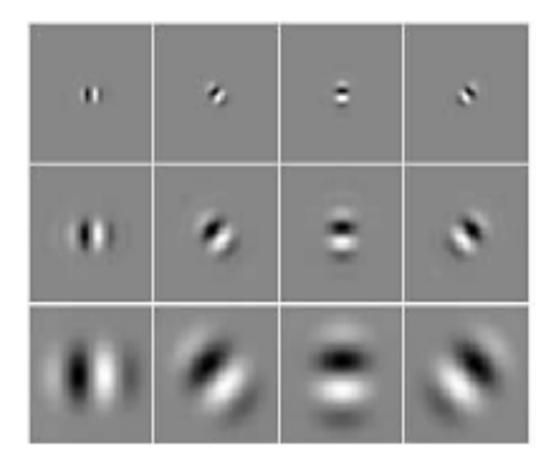








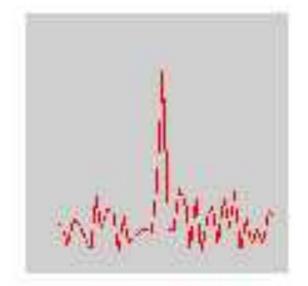
Filter response histograms

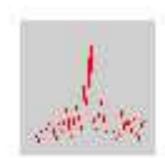




MM

Filter output



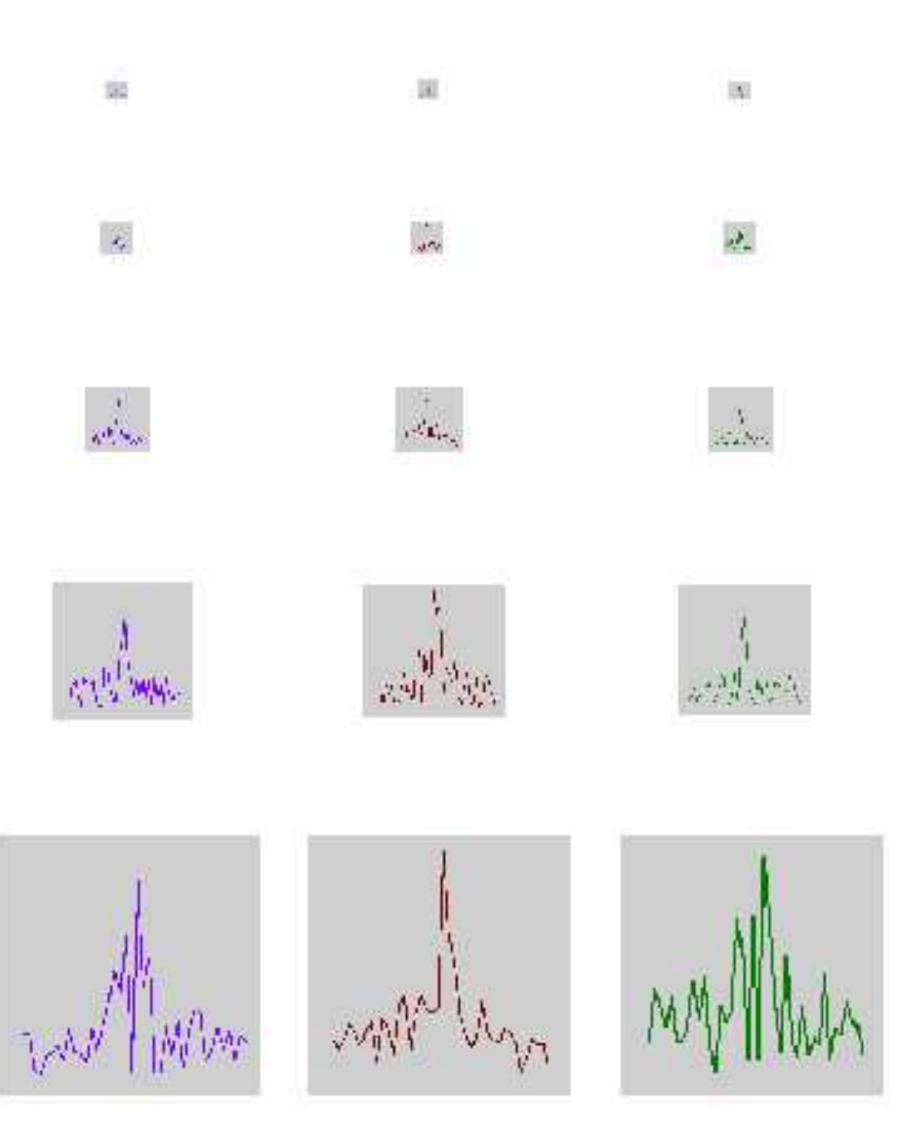




12



Are





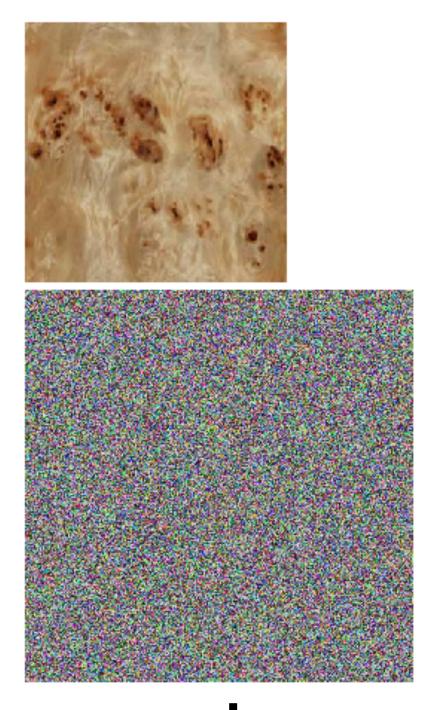
Start with a noise image as output.

Iterative algorithm [Heeger & Bergen, 95]:

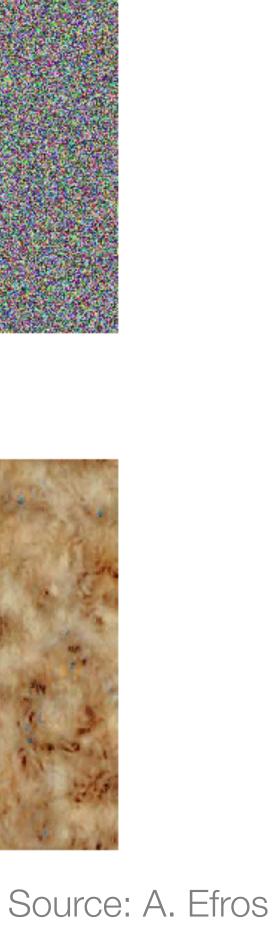
- Match pixel histogram of output image to input
- Decompose input/output images using a Steerable Pyramid
- Match histograms of input and output pyramids
- Reconstruct image and repeat

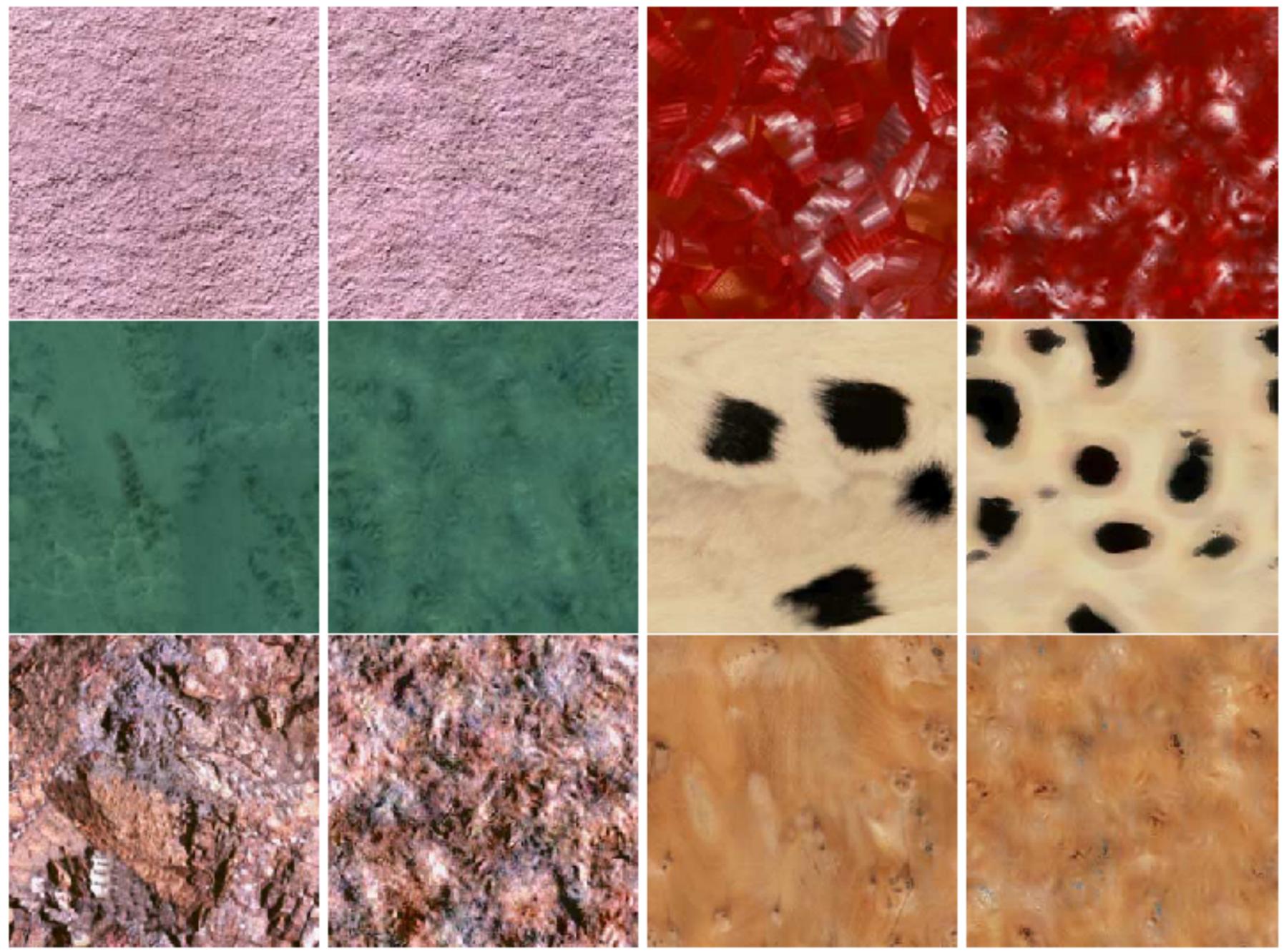
Later in the class we'll see a simpler optimization method on neural nets [Gatys et al. 2015]

Texture synthesis







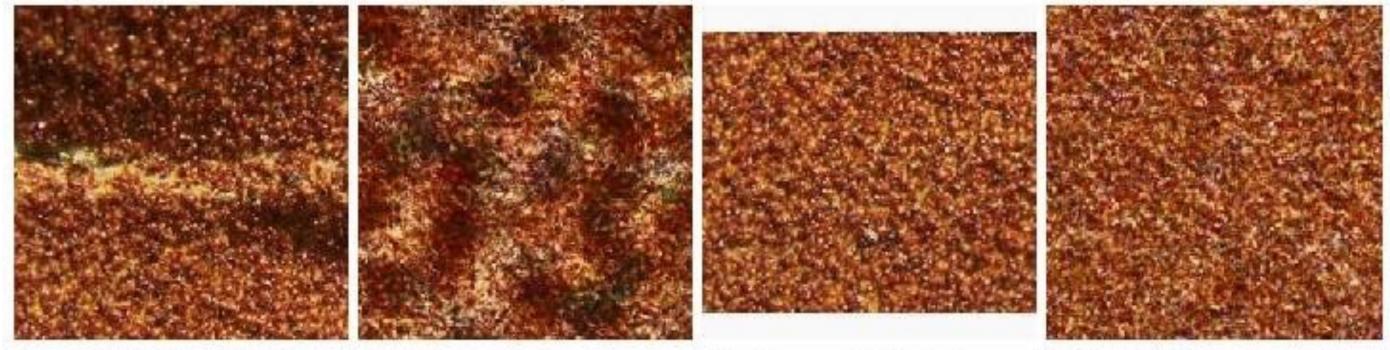








Failure cases



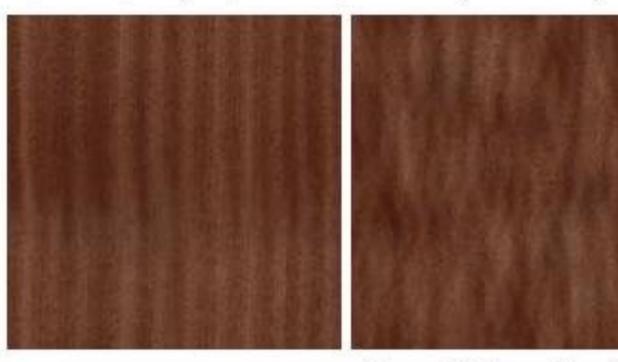
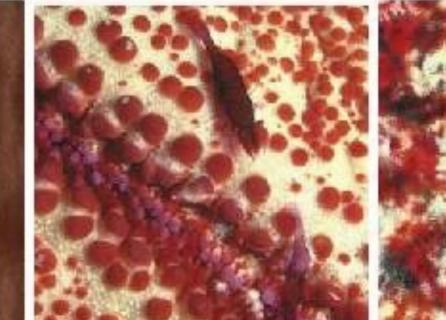






Figure 7: (Left pair) Inhomogoneous input texture produces blotchy synthetic texture. (Right pair) Homogenous input.



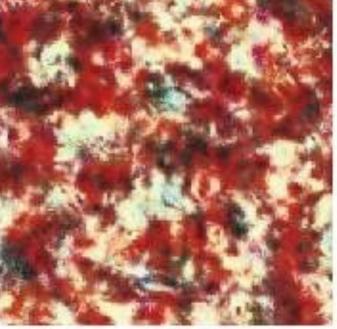


Figure 8: Examples of failures: wood grain and red coral.

Figure 9: More failures: hay and marble.



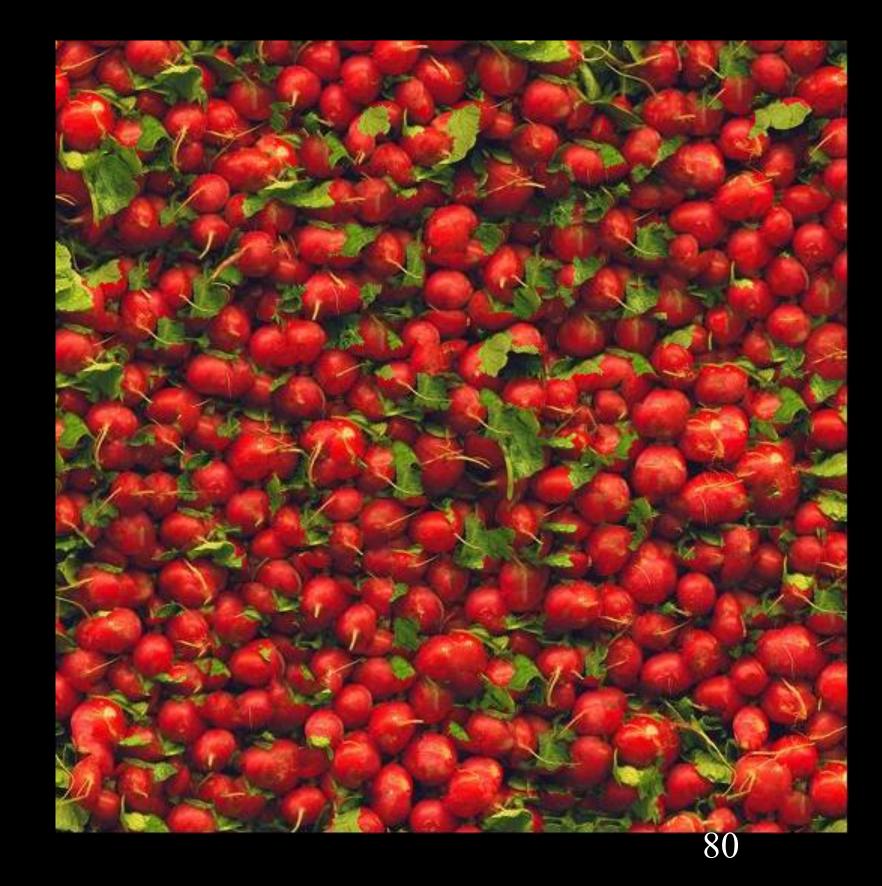
Nonparametric texture synthesis: who needs pyramids or filters?

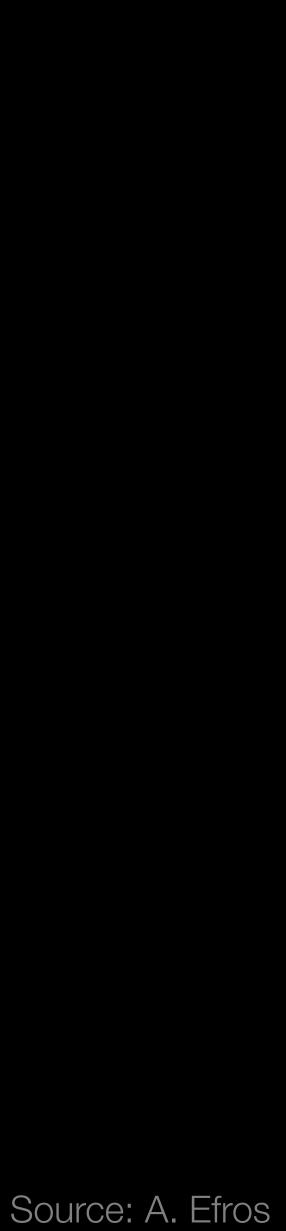
Modeling local neighborhoods

Model $p(\mathbf{p} \mid N(\mathbf{p}))$, Probability of pixel given its neighbors

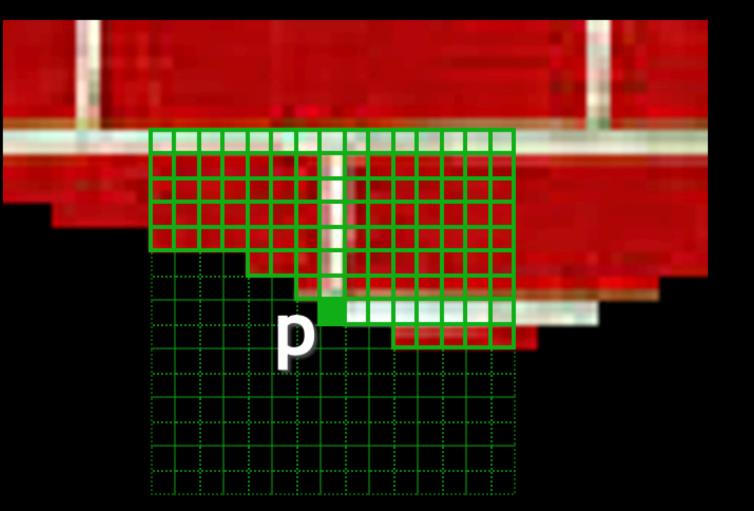








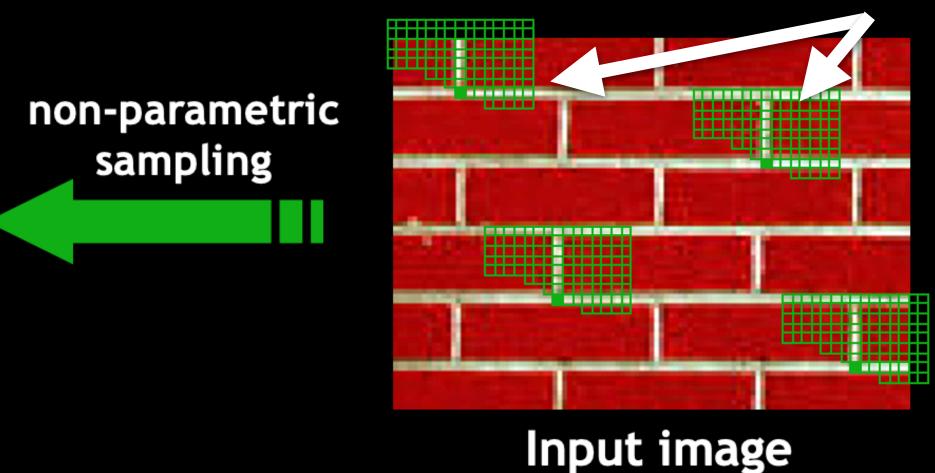
Efros & Leung Algorithm

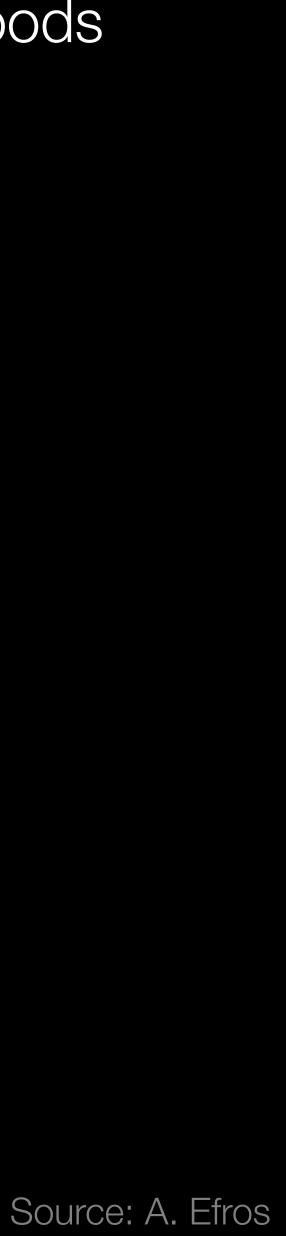


Synthesizing a pixel

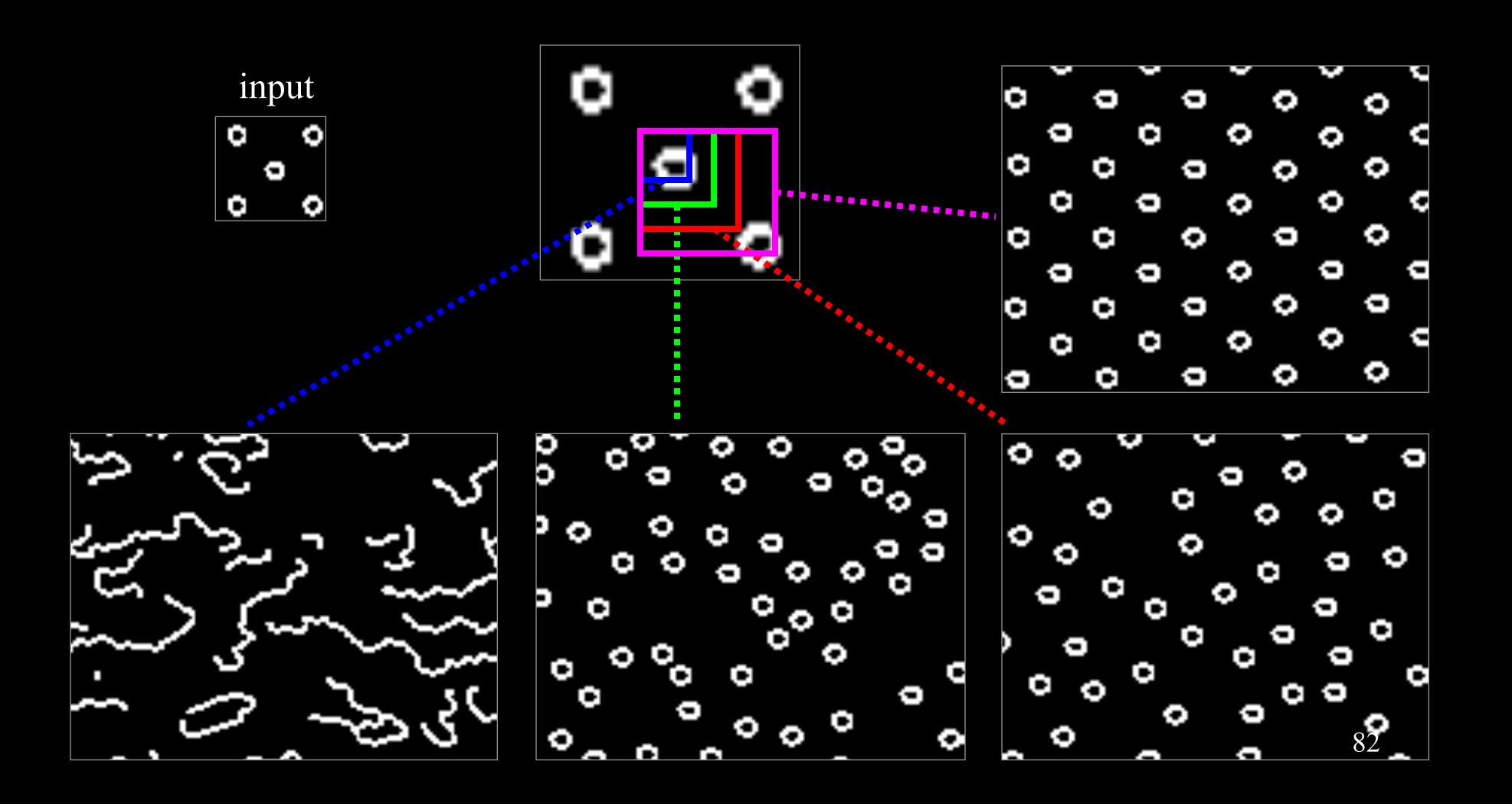
- Synthesize one pixel at a time. Want to sample: P(p|N(p)), where N(p) are the already filled-in neighbors
 - Building explicit probability tables is hard
- Instead, we search the input image for all similar neighborhoods — that's our distribution for p
- To sample from this distribution, just pick one match at random

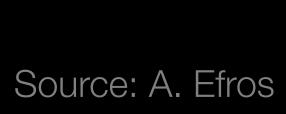
Most similar neighborhoods



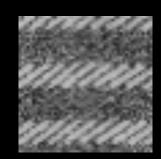


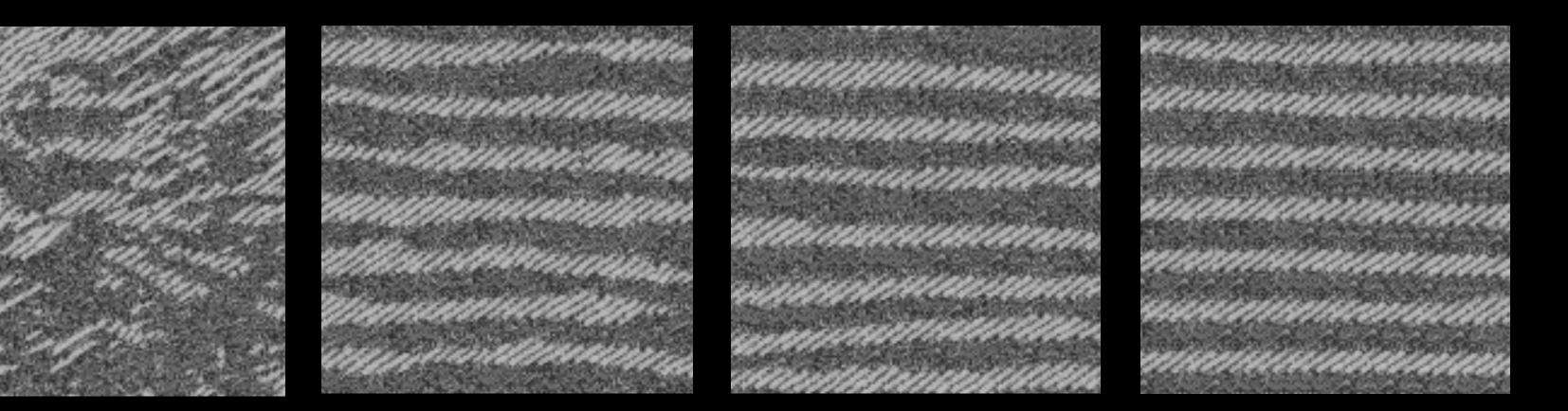
Neighborhood Window

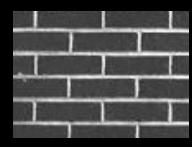


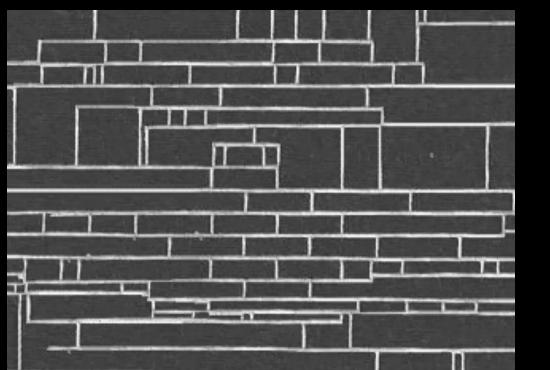


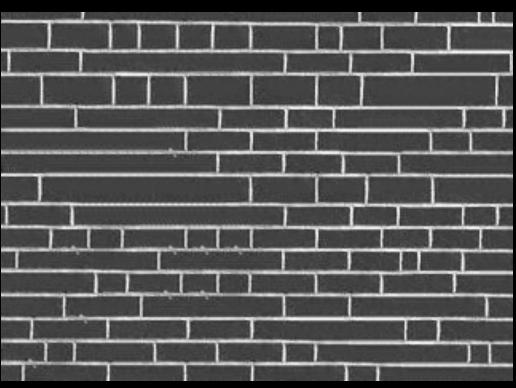






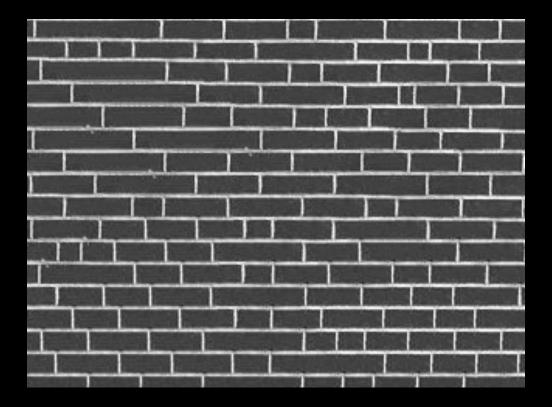




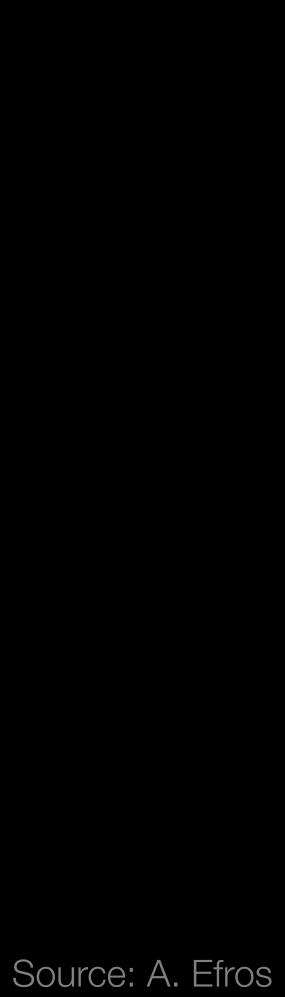


Increasing window size

Varying Window Size

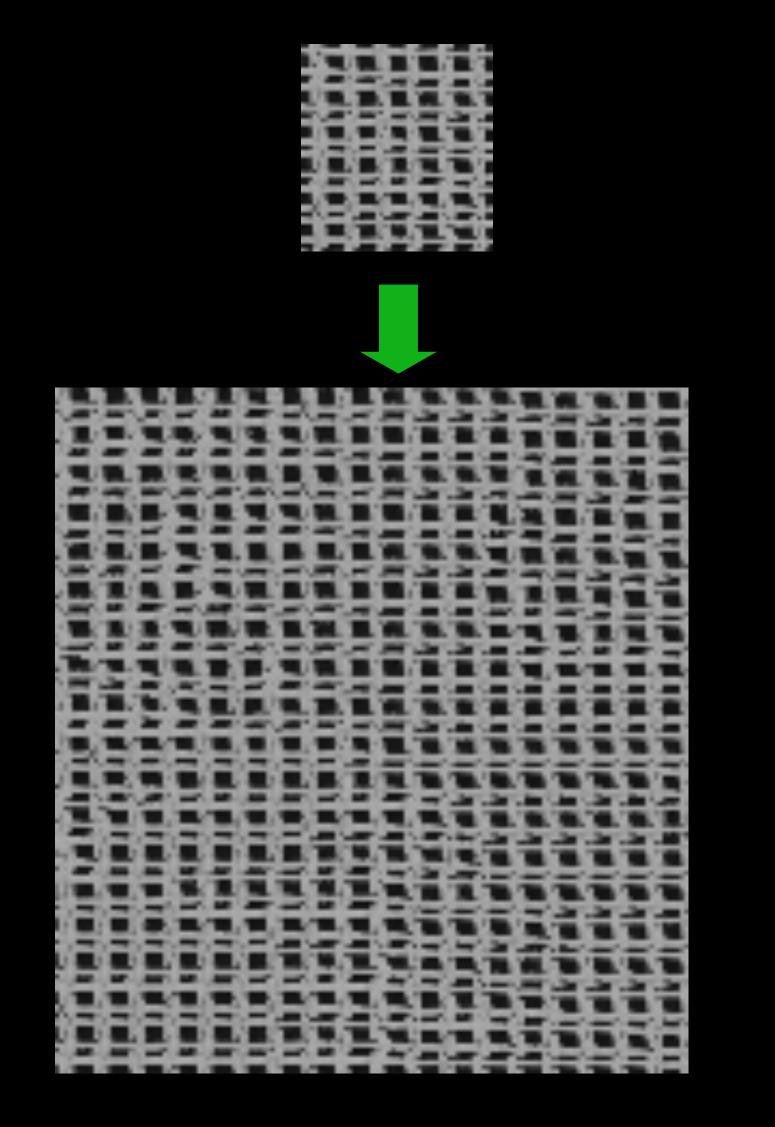




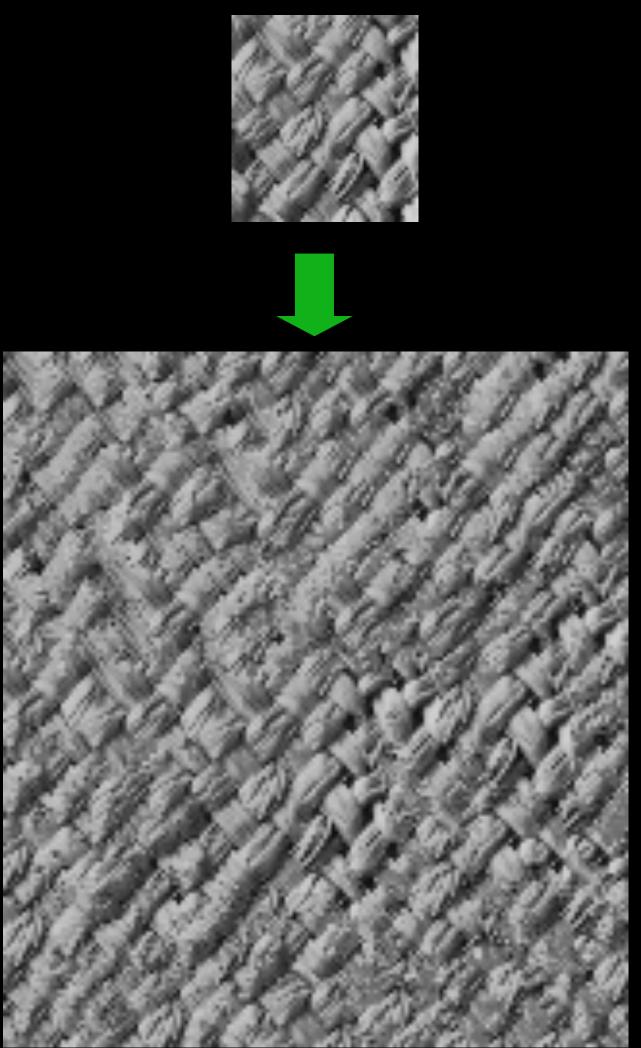


Synthesis Results

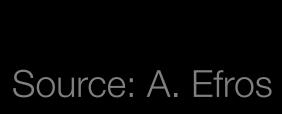
french canvas



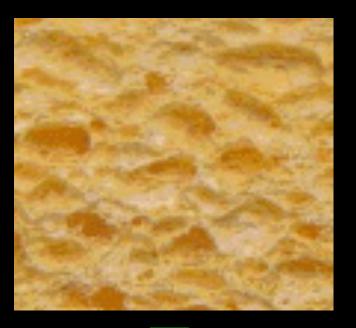
rafia weave

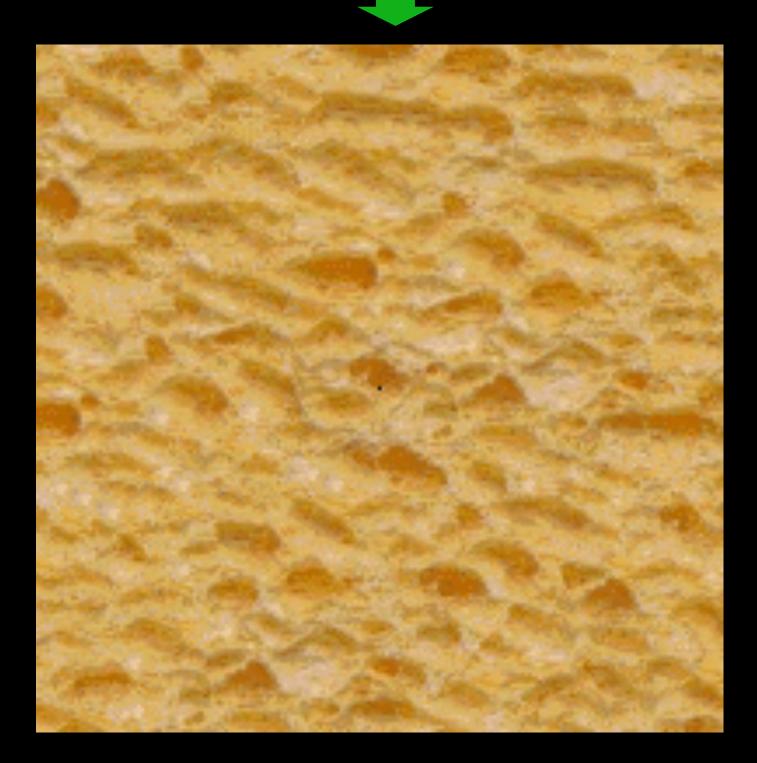


84



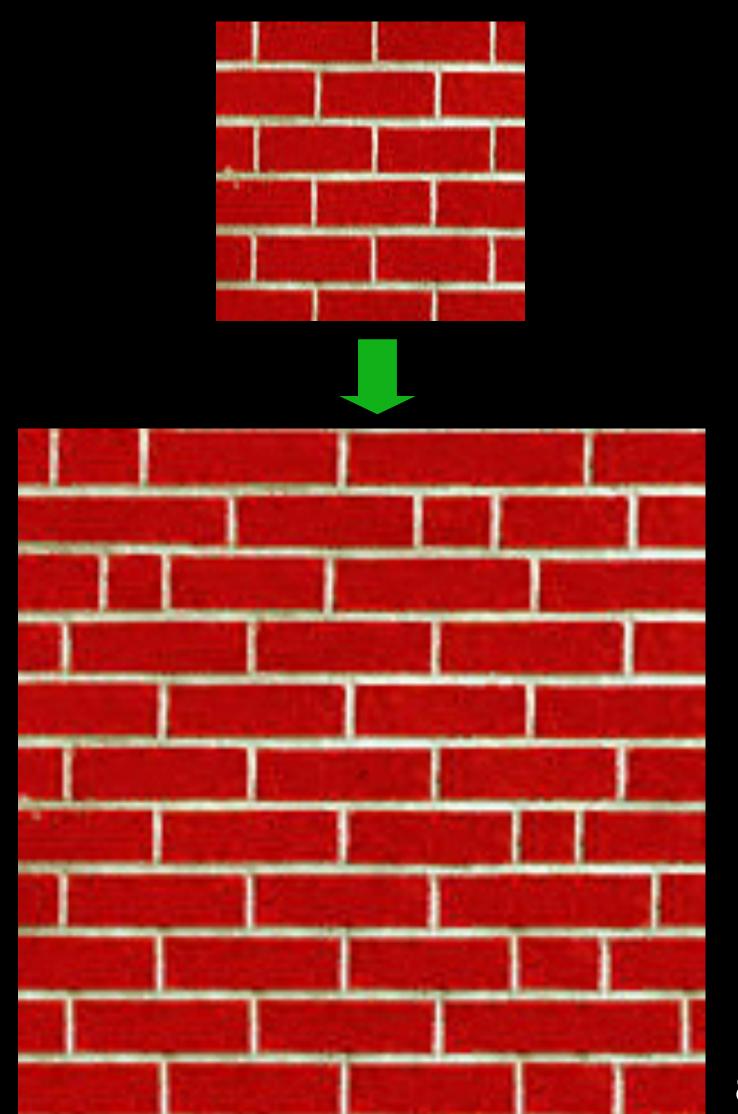
white bread



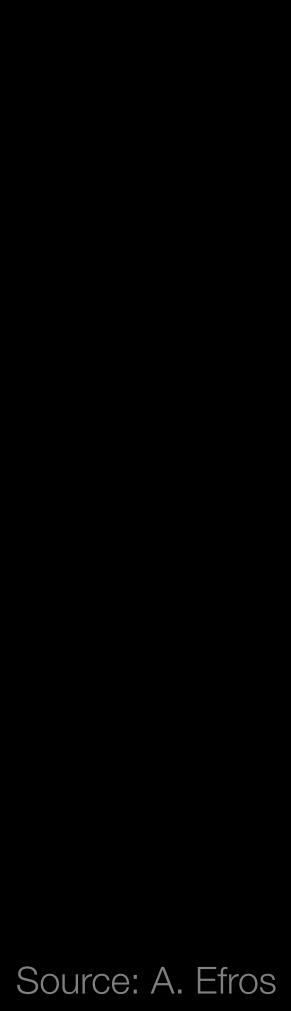


More Results

brick wall



85

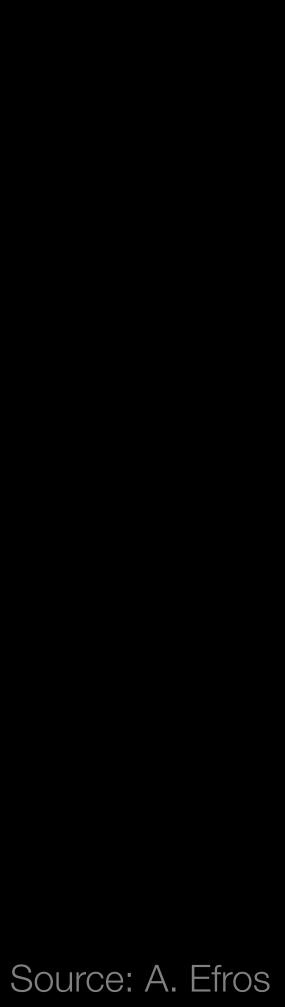


Homage to Shannon

onng m me unsensauor r Dick Gephardt was fai rful riff on the looming : nly asked, "What's your tions?" A heartfelt sigh [.] story about the emergen es against Clinton. "Boy g people about continuin ardt began, patiently obs s, that the legal system h g with this latest tanger

elor a_JS? tirtfooear/ htil ²thuhmr te opm, hrist^ydh þnr titseu id of Pule of the alles of the bar inter and the solo of the solo bnt u muabry S utonu C fui hes, ",ithenly n Anf er Eloaeunoh," dthf_{ptd1} Uni in e oui ba hooegahmti $1a_{1,1}$ երի f hsk as vs.m, d tin ^hElt nit. ry_{en} Joenn'n P (Un) n, Dirweiurb mtime ?'rt sy npsaunpun or

thaimn. them ."Whephartfe lartifelintomimen el ck Clirtioout omaim thartfelins.f out 's anestc the ry onst wartfe lck Gepmtoomimeationl sigab Chiooufit Clinut Cll riff on, hat's yordn, parut tly : ons ycontonsteht wasked, paim t sahe loo riff on l nskoneploourtfeas leil A nst Clit, "Wieontongal s k Cirtioouirtfepe.ong pme abegal fartfenstemem tiensteneltorydt telemephinsverdt was agemen ff ons artientont Cling peme asırtfe atiıh, "Boui s nal s fartfelt sig pedr‡l•dt ske abounutie aboutioo tfeonewwas your aboronthardt thatins fain, ped, ains. them, pabout wasy arfuut countly d, In A h ole emthrängboomme agas fa bontinsyst Clinüt : ory about continst Clipeoµinst Cloke agatiff out 0 stome minemen tly ardt beoraboul n, thenly as t G cons faimeme Diontont wat coutlyohgans as fan ien, phrtfaul, "Wbaut cout congagal comininga: mifmst Clivy abon 'al coountha.emungairt tf oun Vhe looorystan loontieph. Intly on, theoplegatick 🕻 ul fatiezontly atie Diontiomt wal s f tbegàe ener mthahgat's enenhinmas fan. "intchthory abons y

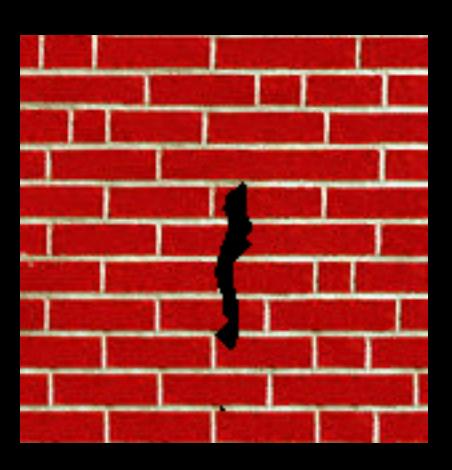


Hole Filling (a.k.a. Inpainting)

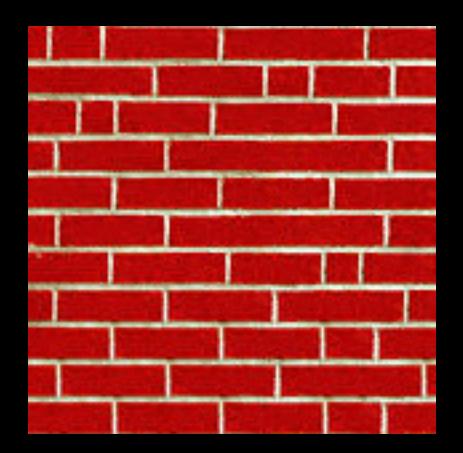


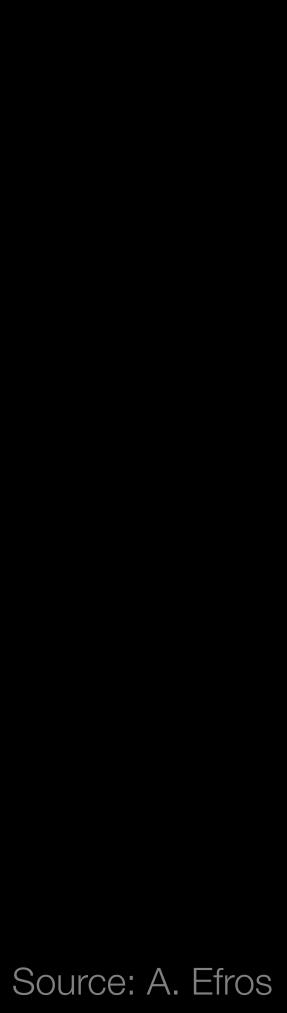




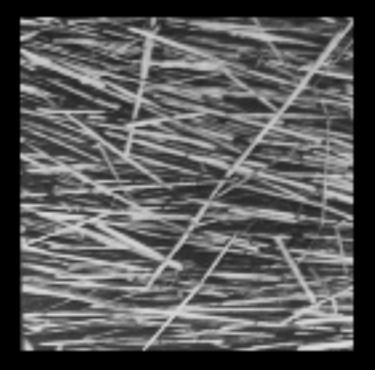




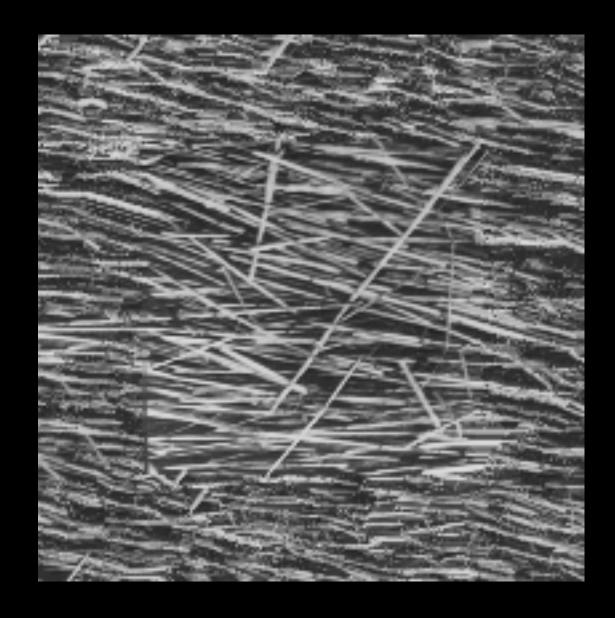




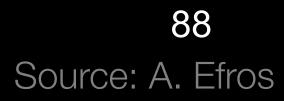
Extrapolation











• Image pyramids Image statistics • Texture synthesis

Today