Image pyramids and their applications

Feb. 26, 2008

Image pyramids

- **Gaussian**
- Laplacian
- Wavelet/QMF
- Steerable pyramid

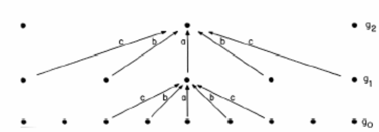
Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image The original image, level 0, meusures 257 by 257 pixels and each higher level array is roughly half the dimensdons of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

The computational advantage of pyramids

GAUSSIAN PYRAMID



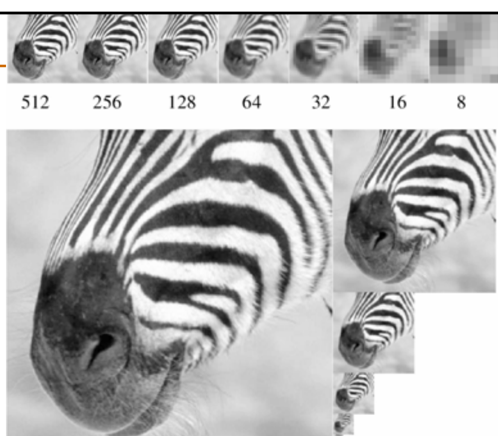
g_0 = IMAGE

g_L = REDUCE $[g_{L-1}]$

Fig 1. A one-dimensional graphic representation of the process which generates a Gaussian pyramid Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or "generating kernel" is used to generate all levels.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983



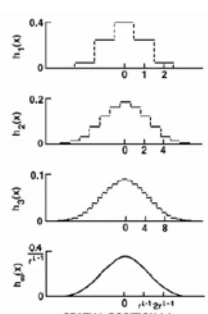


Fig. 2. The equivalent weighting functions $h_l(x)$ for nodes in levels 1, 2, 3, and infinity of the Gaussian pyramid. Note that axis scales have been adjusted by factors of 2 to aid comparison Here the parameter a of the generating kernel is 0.4, and the resulting equivalent weighting functions closely resemble the Gaussian probability density functions.

http://www-bcs.mit.edu/people/adelson/pub_pdfs/pyramid83.pdf

IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 4, APRIL 1983

Convolution and subsampling as a matrix multiply (1-d case)

$$U1 = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \end{pmatrix}$$

Next pyramid level

$$U2 = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 \end{pmatrix}$$

The combined effect of the two pyramid levels

>> U2 * U1 =

$$\begin{pmatrix} 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 31 & 20 & 10 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 31 & 40 & 44 & 40 & 30 & 16 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4 & 10 & 20 & 25 & 16 & 4 & 0 \end{pmatrix}$$

Image pyramids

- **Gaussian**
- **Laplacian**
- **Wavelet/QMF**
- **Steerable pyramid**

Image pyramids

- **Gaussian**
- **Laplacian**
- **Wavelet/QMF**
- **Steerable pyramid**

The Laplacian Pyramid

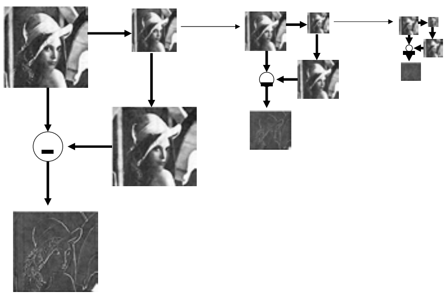
- **Synthesis**

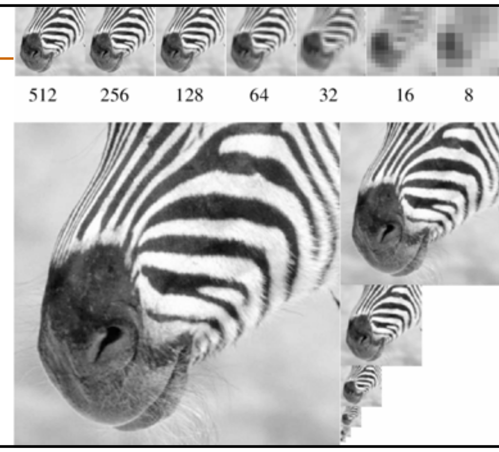
  preserve difference between upsampled Gaussian pyramid level and Gaussian pyramid level

  band pass filter- each level represents spatial frequencies (largely) unrepresented at other levels
- **Analysis**

  reconstruct Gaussian pyramid, take top layer

Laplacian pyramid algorithm





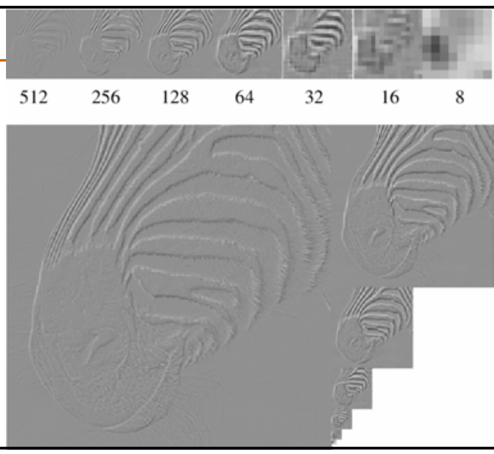


Image pyramids

- Gaussian
- Laplacian
- **Wavelet/QMF**
- Steerable pyramid

What is a good representation for image analysis?

(Goldilocks and the three representations)

- **Fourier transform domain tells you "what" (textural properties), but not "where". In space, this representation is too spread out.**
- **Pixel domain representation tells you "where" (pixel location), but not "what". In space, this representation is too localized**
- **Want an image representation that gives you a local description of image events—what is happening where. That representation might be "just right".**

Wavelets/QMF's

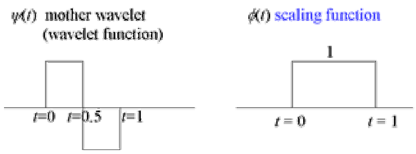
transformed image

$$\vec{F} = U\vec{f}$$

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

The simplest wavelet transform: the Haar transform



$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The inverse transform for the Haar wavelet

U=

$$\begin{matrix} 0.5000 & 0.5000 \\ 0.5000 & -0.5000 \end{matrix}$$

Apply this over multiple spatial positions

U =

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

The high frequencies

U =

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

The low frequencies

U =

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

The inverse transform

>> inv(U) =

$$\begin{pmatrix} 0.5000 & 0.5000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5000 & -0.5000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5000 & 0.5000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5000 & -0.5000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5000 & 0.5000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5000 & -0.5000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5000 & 0.5000 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5000 & -0.5000 \end{pmatrix}$$

Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

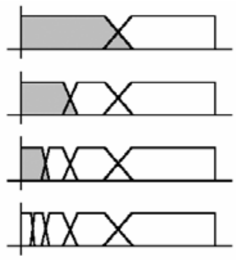
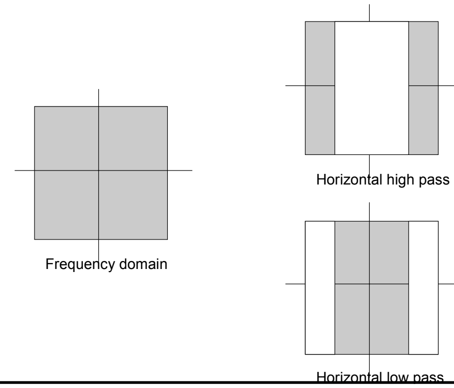
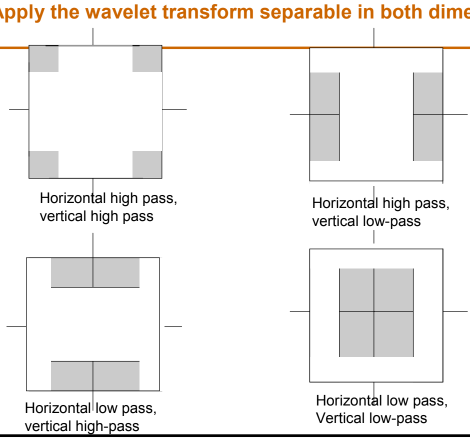


Figure 4.4: Octave band splitting produced by a four-level pyramid cascade of a two-band A/S system. The top picture represents the splitting of the two-band A/S system. Each successive picture shows the effect of re-applying the system to the lowpass subband (indicated in grey) of the previous picture. The bottom picture gives the final four-level partition of the frequency domain. All frequency axes cover the range from 0 to π.

Now, in 2 dimensions…



Apply the wavelet transform separable in both dimensions



Simoncelli and Adelson, in “Subband coding”, Kluwer, 1990.

To create 2-d filters, apply the 1-d filters separably in the two spatial dimensions

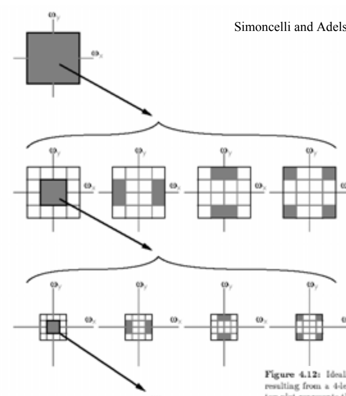
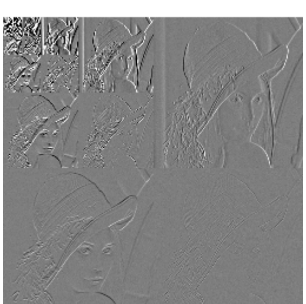


Figure 4.12: Idealized diagram of the partition of the frequency plane resulting from a 4-level pyramid cascade of separable 2-band filters. The top plot represents the frequency spectrum of the original image, with axes ranging from $-\pi$ to π. This is divided into four subbands at the next level. On each subsequent level, the lowpass subband (outlined in bold) is subdivided further.

Wavelet/QMF representation



Good and bad features of wavelet filters

- **Bad:**
  - Aliased subbands
  - Non-oriented diagonal subband
- **Good:**
  - Not overcomplete (so same number of coefficients as image pixels).
  - Good for image compression (JPEG 2000)

Image pyramids

- Gaussian
- Laplacian
- Wavelet/QMF
- Steerable pyramid

Steerable filters

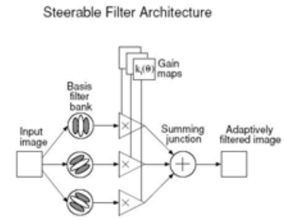


Figure 2-3: Steerable filter system block diagram. A bank of dedicated filters process the image. Their outputs are multiplied by a set of gain maps which adaptively control the orientation of the synthesized filter.

http://people.csail.mit.edu/billf/freemanThesis.pdf

Oriented Filters

- **Filter bank:**
  - Mix of edge, bar, spot filters at multiple scales and orientations

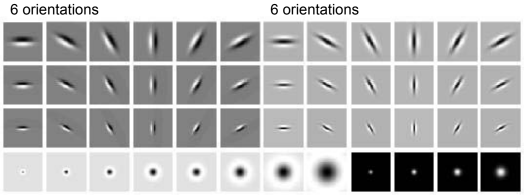
1st derivative of a gaussian

6 orientations

2nd derivative of a gaussian

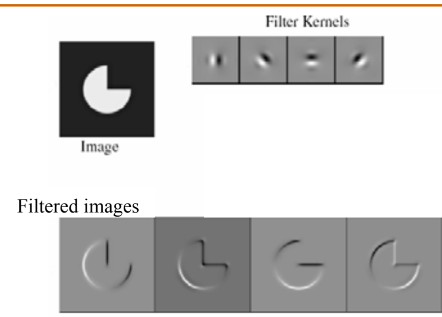
6 orientations

3 scales



8 Laplacian of Gaussian filters

4 Gaussian filters



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

Non-oriented steerable pyramid

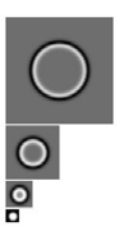


Figure 4: A 3-level $k = 1$ (non-oriented) steerable pyramid. Shown are the bandpass images and the final lowpass image.

http://www.merl.com/reports/docs/TR95-15.pdf

3-orientation steerable pyramid

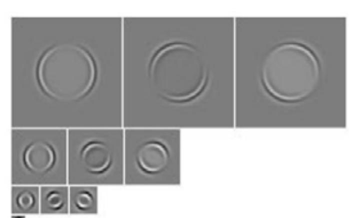


Figure 5: A 3-level $k = 3$ (second derivative) steerable pyramid. Shown are the three bandpass images at each scale and the final lowpass image.

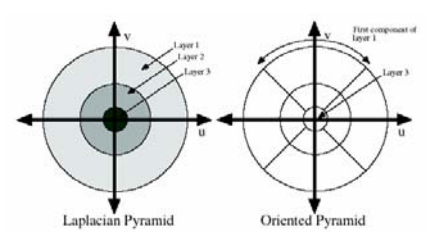
http://www.merl.com/reports/docs/TR95-15.pdf

Steerable pyramids

- **Good:**
  - Oriented subbands
  - Non-aliased subbands
  - Steerable filters
- **Bad:**
  - Overcomplete
  - Have one high frequency residual subband, required in order to form a circular region of analysis in frequency from a square region of support in frequency.

Oriented pyramids

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



	Laplacian Pyramid	Dyadic QMF/Wavelet	Steerable Pyramid
self-inverting (tight frame)	no	yes	yes
overcompleteness	4/3	1	4k/3
aliasing in subbands	perhaps	yes	no
rotated orientation bands	no	only on hex lattice [9]	yes

Table 1: Properties of the Steerable Pyramid relative to two other well-known multi-scale representations.

http://www.cns.nyu.edu/ftp/eero/simoncelli95b.pdf Simoncelli and Freeman, ICIP 1995

But we need to get rid of the corner regions before starting the recursive circular filtering

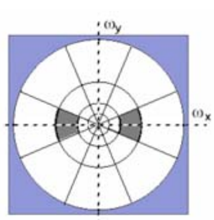


Figure 1. Idealized illustration of the spectral decomposition performed by a steerable pyramid with $k = 4$. Frequency axes range from $-\pi$ to π. The basis functions are related by translations, dilations and *rotations* (except for the initial highpass subband and the final lowpass subband). For example, the shaded region corresponds to the spectral support of a single (vertically-oriented) subband.

http://www.cns.nyu.edu/ftp/eero/simoncelli95b.pdf Simoncelli and Freeman, ICIP 1995

- **Summary of pyramid representations**

Image pyramids

- **Gaussian**

Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

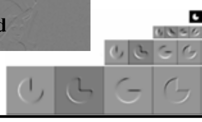
- **Laplacian**

Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- **Wavelet/QMF**

Bandpassed representation, complete, but with aliasing and some non-oriented subbands.

- **Steerable pyramid**



Shows components at each scale and orientation separately. Non-aliased subbands. Good for texture and feature analysis.

Schematic pictures of each matrix transform

Shown for 1-d images

The matrices for 2-d images are the same idea, but more complicated, to account for vertical, as well as horizontal, neighbor relationships.

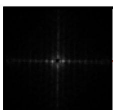
transformed image

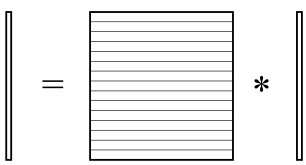
$$\vec{F} = U\vec{f}$$

Vectorized image

Fourier transform, or
Wavelet transform, or
Steerable pyramid transform

Fourier transform





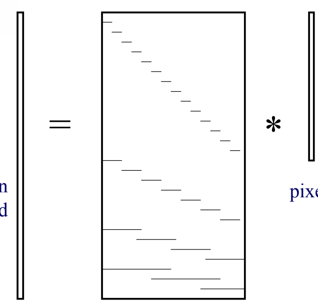
Fourier transform

Fourier bases are global: each transform coefficient depends on all pixel locations.

pixel domain image

Gaussian pyramid

Gaussian pyramid

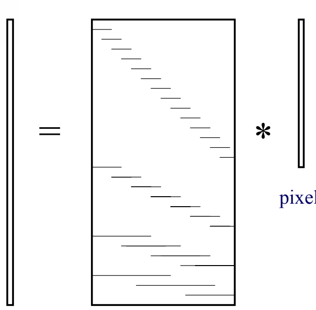


pixel image

Overcomplete representation. Low-pass filters, sampled appropriately for their blur.

Laplacian pyramid

Laplacian pyramid

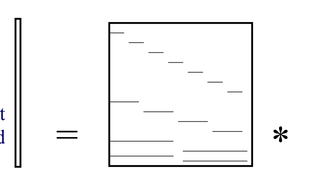


pixel image

Overcomplete representation. Transformed pixels represent bandpassed image information.

Wavelet (QMF) transform

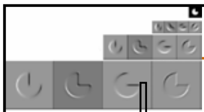
Wavelet pyramid

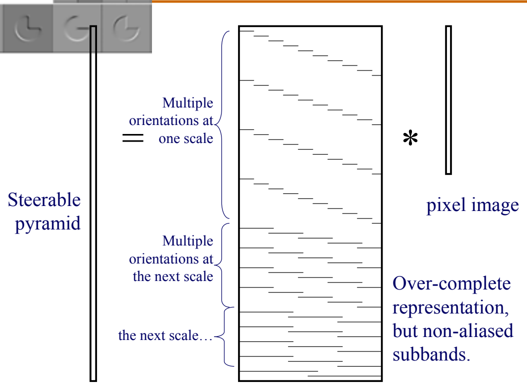


Ortho-normal transform (like Fourier transform), but with localized basis functions.

pixel image

Steerable pyramid





Matlab resources for pyramids (with tutorial)

http://www.cns.nyu.edu/~eero/software.html

Eero P. Simoncelli

Associate Investigator,
Howard Hughes Medical Institute

Associate Professor,
Neural Science and Mathematics,
New York University

Matlab resources for pyramids (with tutorial)

http://www.cns.nyu.edu/~eero/software.html

lcv Laboratory for Computational Vision

Home | People | Research | Publications | Software

Publicly Available Software Packages

- Texture Analysis/Synthesis - Matlab code is available for analyzing and synthesizing visual textures. README | Contents | ChangeLog | Source code (UNIX/PC, gzip'ed tar file)
- EPWIC - Embedded Progressive Wavelet Image Coder. C source code available.
- **matlabPyrTools** - Matlab source code for multi-scale image processing. Includes tools for building and manipulating Laplacian pyramids, QMF/Wavelets, and steerable pyramids. Data structures are compatible with the Matlab wavelet toolbox, but the convolution code (in C) is faster and has many boundary-handling options. README, Contents, Modification list, UNIX/PC source or Macintosh source.
- The Steerable Pyramid, an (approximately) translation- and rotation-invariant multi-scale image decomposition. MatLab (see above) and C implementations are available.
- Computational Models of cortical neurons. Macintosh program available.
- EPIC - Efficient Pyramid (Wavelet) Image Coder. C source code available.
- OBVIUS [Object-Based Vision & Image Understanding System]: README / ChangeLog / Doc (225k) / Source Code (2.25M).
- CL-SHELL [Gnu Emacs <-> Common Lisp Interface]: README / Change Log / Source Code (119k).

Why use these representations?

- **Handle real-world size variations with a constant-size vision algorithm.**
- **Remove noise**
- **Analyze texture**
- **Recognize objects**
- **Label image features**

E. H. Adelson | C. H. Anderson | J. R. Bergen | P. J. Burt | J. M. Ogden

Pyramid methods in image processing

The image pyramid offers a flexible, convenient multiresolution format that mirrors the multiple scales of processing in the human visual system.

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

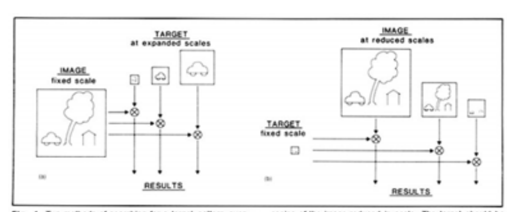


Fig. 1. Two methods of searching for a target pattern over many scales. In the first approach, (a), copies of the target pattern are constructed at several expanded scales, and each is convolved with the original image. In the second approach, (b), a single copy of the target is convolved with copies of the image reduced in scale. The target should be just large enough to resolve critical details. The two approaches should give equivalent results, but the second is more efficient by the fourth power of the scale factor (image convolutions are represented by "O").

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

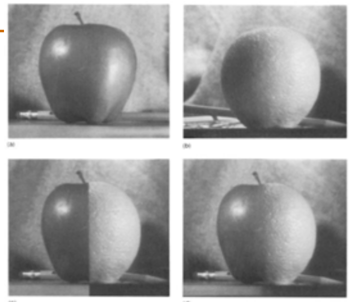


Fig. 10. *Image mosaics. The left half of image (a) is catinated with the right half of image (b) to give the mosaic in (c). Note that the boundary between regions is clearly visible. The mosaic in (d) was obtained by combining images separately in each spatial frequency band of their pyramid representations then expanding and summing these bandpass mosaics.*

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

Creating large depth-of-field (early approach)

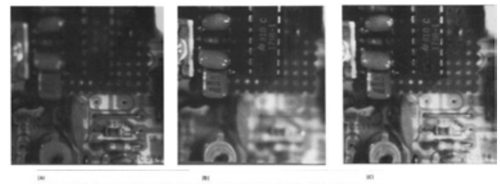


Fig. 9. *Multifocus composite image. The original images with limited depth of field are shown in (a) and (b). These are combined digitally to give the image will an extended depth of field in (c).*

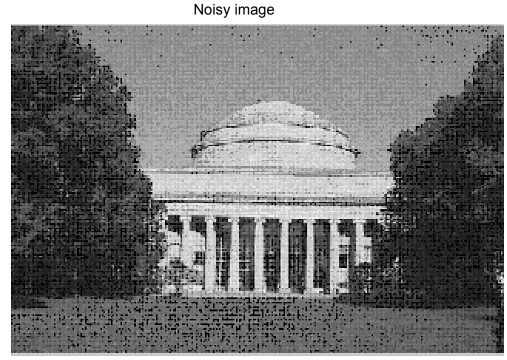
summe
Note th
values

http://web.mit.edu/persci/people/adelson/pub_pdfs/RCA84.pdf

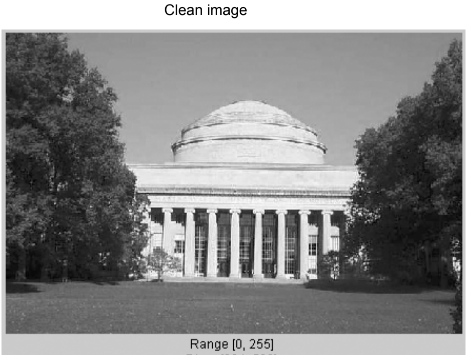
Image pyramids for noise removal

Image statistics (or, mathematically, how can you tell image from noise?)

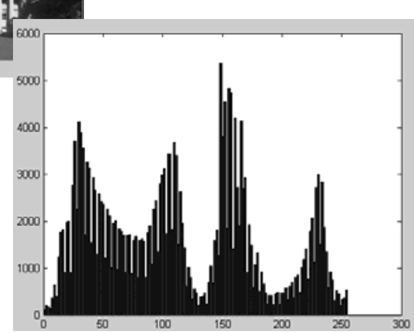
Noisy image



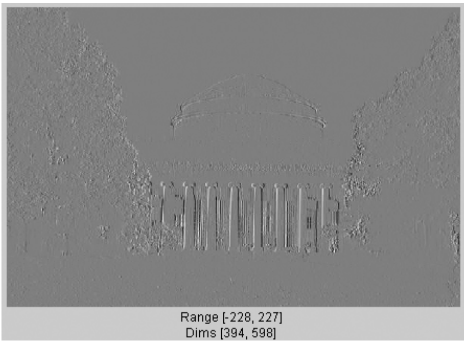
Clean image



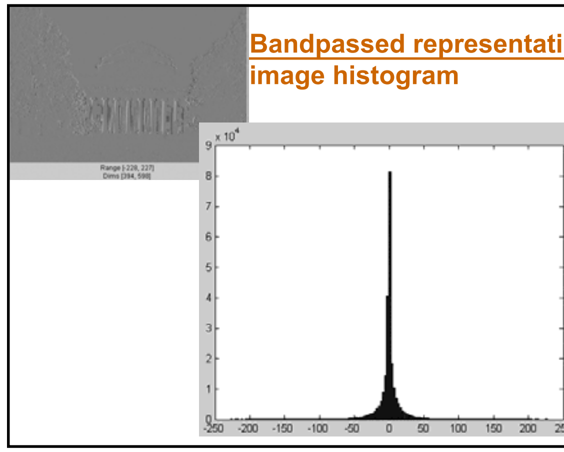
Pixel representation image histogram



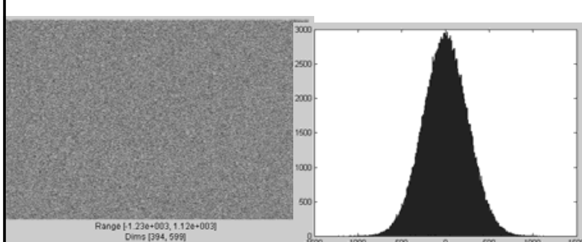
Bandpass filtered image



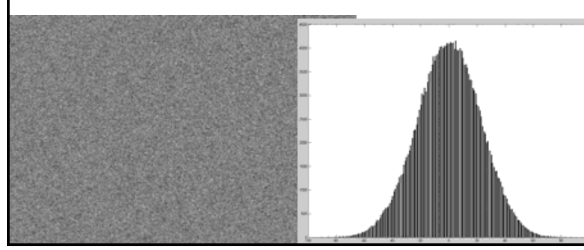
Bandpassed representation image histogram



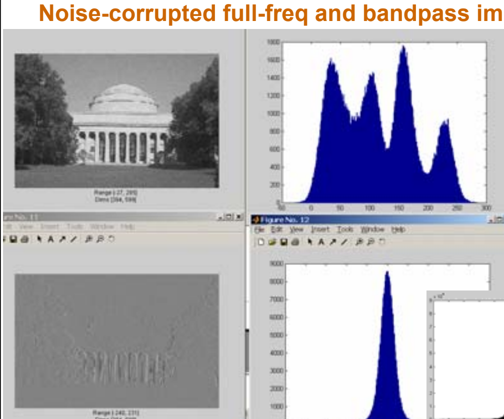
Pixel domain noise image and histogram



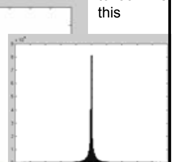
Bandpass domain noise image and histogram



Noise-corrupted full-freq and bandpass images



But want the bandpass image histogram to look like this



Noise removal results

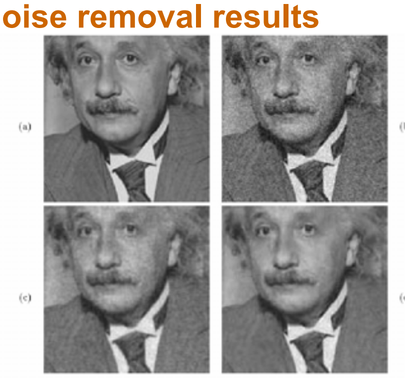


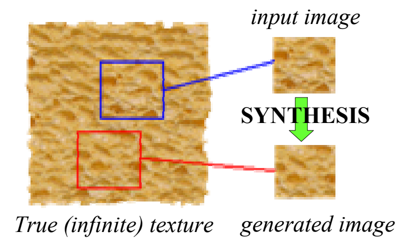
Figure 4: Noise reduction example. (a) Original image (cropped). (b) Image contaminated with additive Gaussian white noise (SNR = 9.00dB). (c) Image restored using (semi-blind) Wiener filter (SNR = 11.88dB). (d) Image restored using (semi-blind) Bayesian estimator (SNR = 13.82dB).

Simoncelli and Adelson, Noise Removal via Bayesian Wavelet Coring

http://www-bcs.mit.edu/people/adelson/pub_pdfs/simoncelli_noise.pdf

Image texture

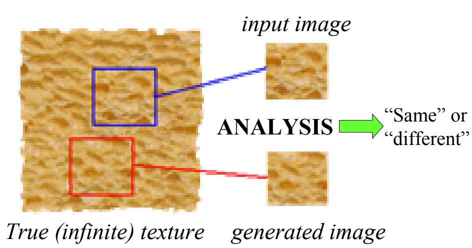
The Goal of Texture Synthesis



- **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“

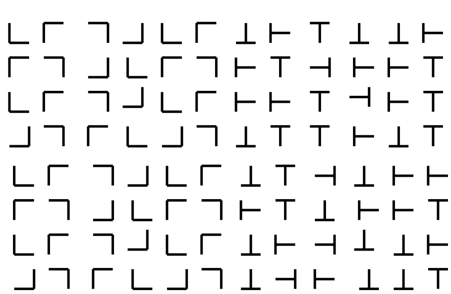
The Goal of Texture Analysis



Compare textures and decide if they’re made of the same “stuff”.

Pre-attentive texture discrimination

Pre-attentive texture discrimination

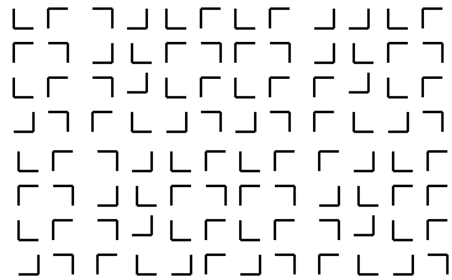


Pre-attentive texture discrimination

Same or different textures?

Pre-attentive texture discrimination

Pre-attentive texture discrimination

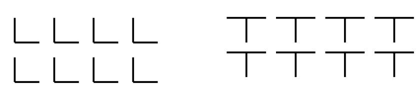


Pre-attentive texture discrimination

Same or different textures?

Julesz

- **Textons: analyze the texture in terms of statistical relationships between fundamental texture elements, called "textons".**
- **It generally required a human to look at the texture in order to decide what those fundamental units were...**



Influential paper:

Early vision and texture perception

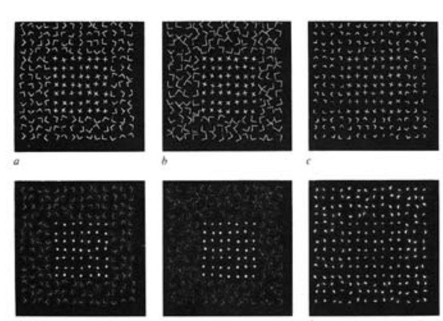
James R. Bergen* & Edward H. Adelson**

* SRI David Sarnoff Research Center, Princeton, New Jersey 08540, USA
** Media Lab and Department of Brain and Cognitive Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Learn: use filters.

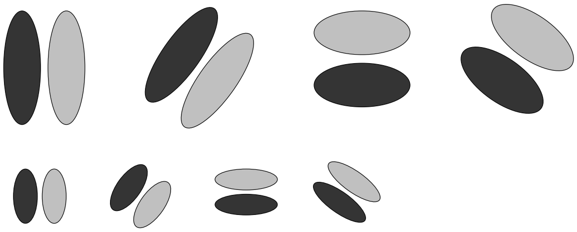
Bergen and Adelson, Nature 1988

Fig. 1 *Top row*. Textures consisting of Xs within a texture composed of Ls. The micropatterns are placed at random orientations on a randomly perturbed lattice. *a*, The bars of the Xs have the same length as the bars of the Ls. *b*, The bars of the Ls have been lengthened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is enhanced. *c*, The bars of the Ls have been shortened by 25%, and the intensity adjusted for the same mean luminance. Discriminability is impaired. *Bottom row*: the responses of a size-tuned mechanism *d*, response to image *a*; *e*, response to image *b*; *f*, response to image *c*.



Malik and Perona

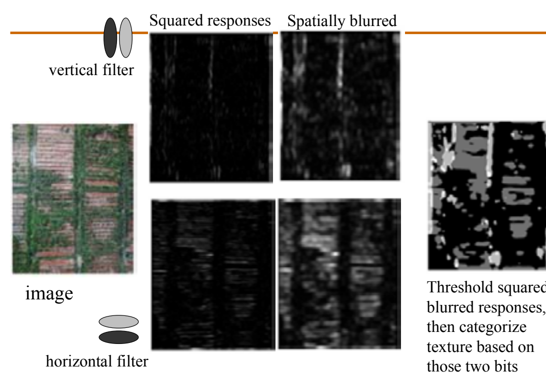
Learn: use lots of filters, multi-ori&scale.

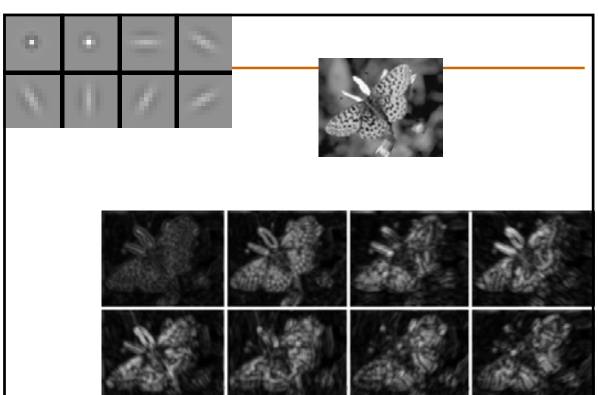


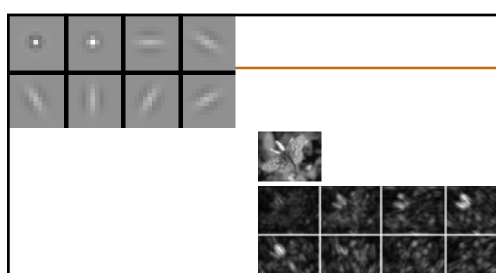
Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms. J OPT SOC AM A 7: (5) 923-932 MAY 1990

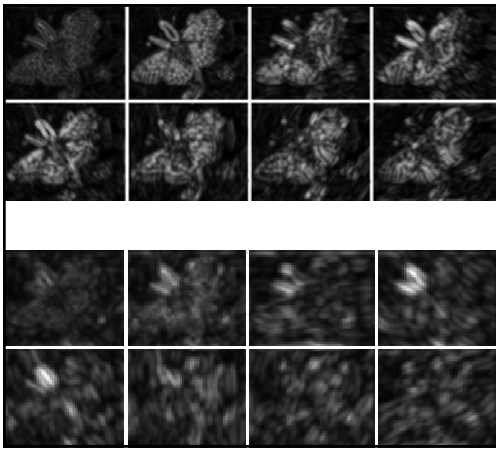
Representing textures

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









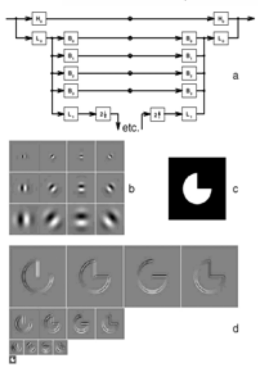
If matching the averaged squared filter values is a good way to match a given texture, then maybe matching the entire marginal distribution (eg, the histogram) of a filter's response would be even better.

Jim Bergen proposed this…

Pyramid-Based Texture Analysis/Synthesis

David J. Heeger* Stanford University

James R. Bergen† SRI David Sarnoff Research Center



SIGGRAPH 1994

Histogram matching algorithm

```
Match-histogram (im1,im2)
  im1-cdf  = Make-cdf(im1)
  im2-cdf  = Make-cdf(im2)
  inv-im2-cdf  = Make-inverse-lookup-table(im2-cdf)
  Loop for each pixel do
       im1[pixel]  =
         Lookup(inv-im2-cdf,
                Lookup(im1-cdf,im1[pixel]))
```

"At this im1 pixel value, 10% of the im1 values are lower. What im2 pixel value has 10% of the im2 values below it?"

The Problem … in Words

- **Given texture *I*, generate a texture *J* which**
  - Looks like the same texture
  - Has no obvious copying or tiling from *I*
  - Difference between *I* and *J* should be the same as the way *I* "differs from itself" [DeBonet97]
- **Things to watch for:**
  - 'Looks the same': what is the texture model?
  - 'Obvious copying': how is it avoided?
  - <u>Underlined text</u>: indicates algorithm parameter

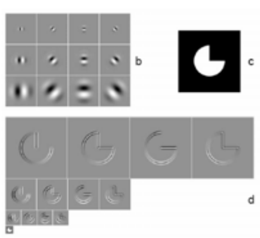
Classes of Algorithms

- **Multiresolution pyramids**
  [HeegerBergen95]
- **Pixel-by-pixel synthesis**
  [EfrosLeung99]
- **Multiresolution pixel-by-pixel**
  [DeBonet97], [WeiLevoy00], [Hertzmann et.al. 01], [Ashikhmin01]
- **Patch quilting**
  [EfrosFreeman01], [Kwatra et.al. 03], [WuYu04]
- **Geometric feature matching**
  [WuYu04], [Liu et.al. 04]

Heeger Bergen 1995

- **Seminal paper that introduced texture synthesis to the graphics community**
- **Algorithm:**
  Initialize *J* to noise
  Create multiresolution pyramids for *I* and *J*
  Match the histograms of *J*'s pyramid levels with *I*'s pyramid levels
  Loop until convergence
  Can be generalized to 3D

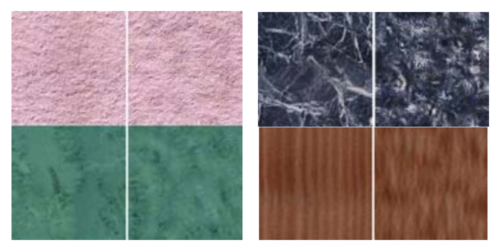
Heeger Bergen 1995 - Algorithm



- **Image pyramids**
  Gaussian
  Laplacian
- **Steerable pyramids [SimoncelliFreeman95]**
  b): multiple scales of oriented filters
  c): a sample image
  d): results of filters in b) applied to c)

Heeger Bergen 1995 - Results

Successes Failures



I *J* *I* *J*

Heeger Bergen 1995 - Results

Heeger Bergen 1995 - Verdict

- **Texture model:**
  Histograms of responses to various filters
- **Avoiding copying:**
  Inherent in algorithm
- **No user intervention required**
- **Captures stochastic textures well**
- **Does not capture structure**
  Lack of inter-scale constraints

De Bonet 1997

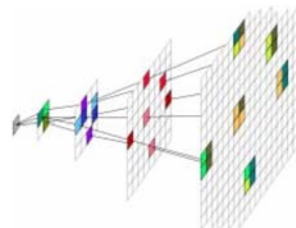
- **Propagate constraints downwards by matching statistics all the way up the pyramid**
- *Feature vector:* **multiscale collection of filter responses for a given pixel**
- **Algorithm:**

  Initialize *J* to empty image

  Create multiresolution pyramids for *I* and *J*

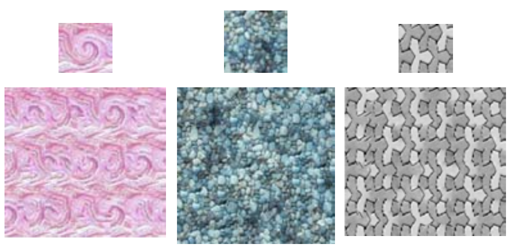
  For each pixel in level of *J*, randomly choose pixel from corresponding level of *I* that has similar feature vector

De Bonet 1997 - Algorithm



- **6 feature vectors shown**
- **Notice how they share parent information**

De Bonet 1997 - Results



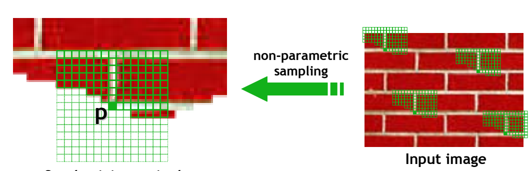
De Bonet 1997 - Verdict

- **Texture model:**

  Feature vector containing multiscale responses to various filters
- **Avoiding copying:**

  Random choice of pixels with 'close' feature vectors, but copying still frequent on small scale
- **Individual per filter thresholds are cumbersome**
- **Feature vectors used in later synthesis work**

Efros & Leung Algorithm



- **Assuming Markov property, compute P(**p**|N(**p**))**

  Building explicit probability tables infeasible

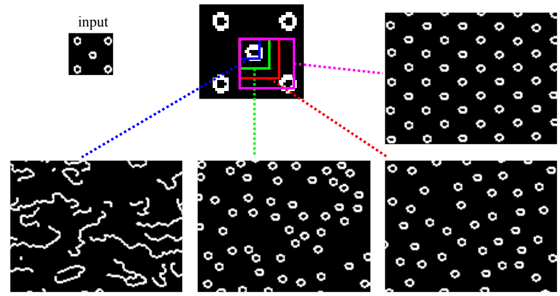
  – Instead, we *search the input image* for all similar neighborhoods — that's our pdf for **p**

  – To sample from this pdf, just pick one match at random

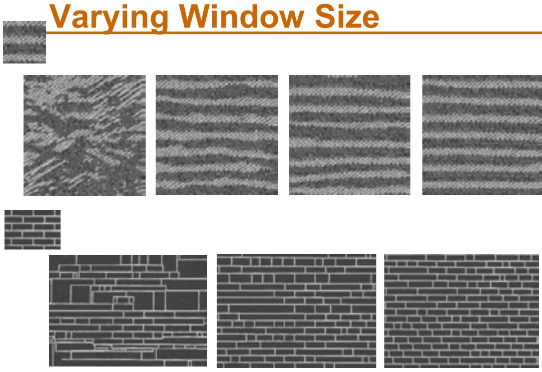
Some Details

- **Growing is in "onion skin" order**
  - Within each "layer", pixels with most neighbors are synthesized first
  - If no close match can be found, the pixel is not synthesized until the end
- **Using *Gaussian-weighted* SSD is very important**
  - to make sure the new pixel agrees with its closest neighbors
  - Approximates reduction to a smaller neighborhood window if data is too sparse

Neighborhood Window



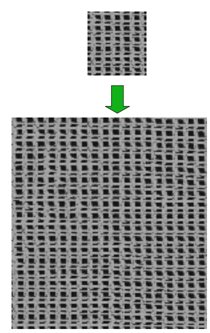
Varying Window Size



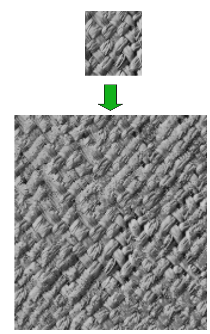
Increasing window size

Synthesis Results

french canvas

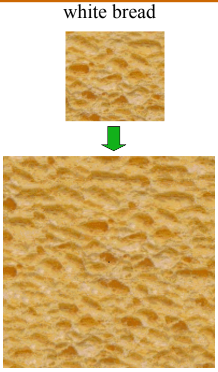


rafia weave

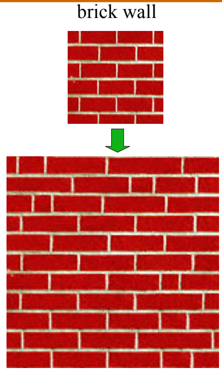


More Results

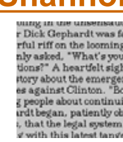
white bread



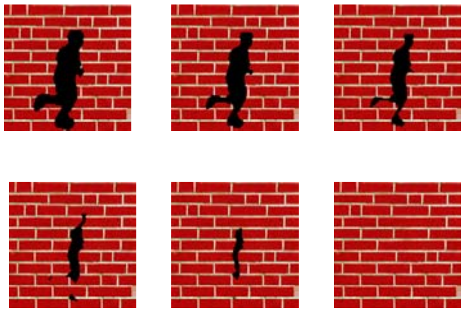
brick wall



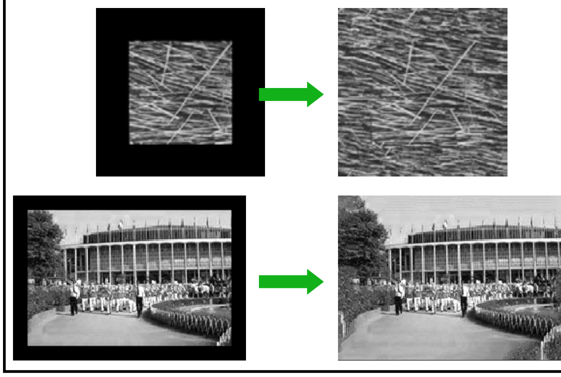
Homage to Shannon



Hole Filling



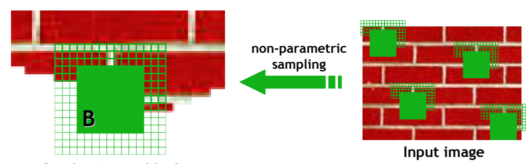
Extrapolation



Summary

- **The Efros & Leung algorithm**
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow

Image Quilting [Efros & Freeman]

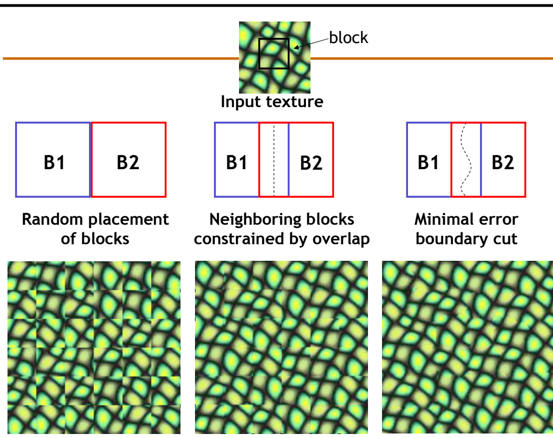


Synthesizing a block

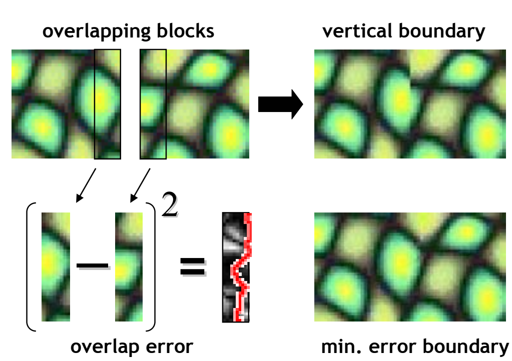
- **<u>Observation:</u> neighbor pixels are highly correlated**

<u>Idea:</u> unit of synthesis = block

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



Minimal error boundary



Our Philosophy

- **The "Corrupt Professor's Algorithm":**

  Plagiarize as much of the source image as you can

  Then try to cover up the evidence

- **Rationale:**

  Texture blocks are by definition correct samples of texture so problem only connecting them together