



# Analysis and Synthesis of Texture

---

CMPE 264: Image Analysis and Computer Vision

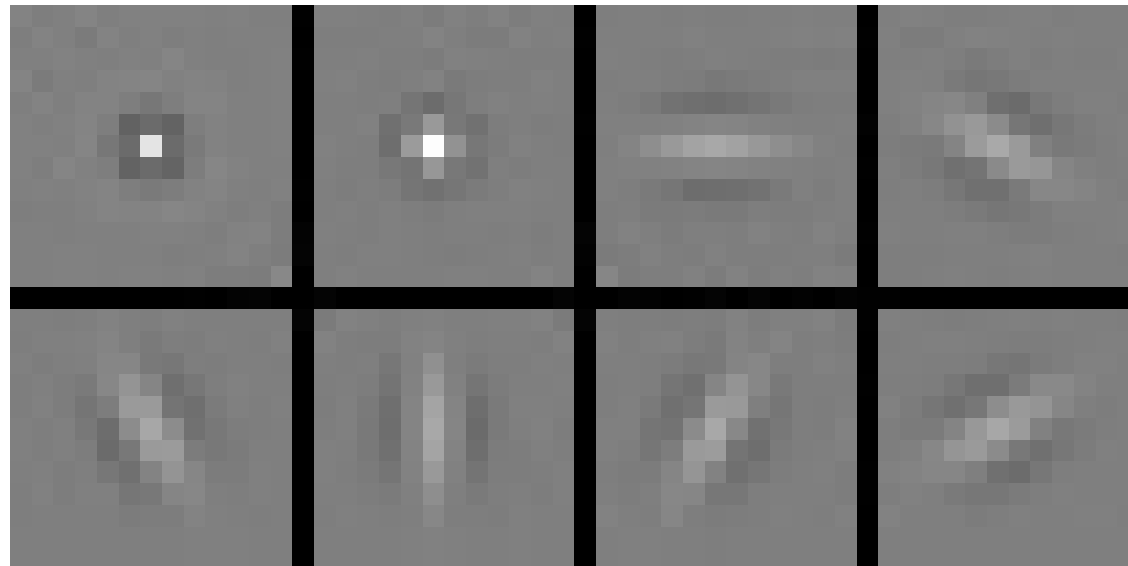
Hai Tao



## Extracting image structure by filter banks

---

- Represent image textures using the responses of a collection of filters
- An appropriately designed filter bank will extract useful information such as spots and edges
- An example of filter banks and the responses

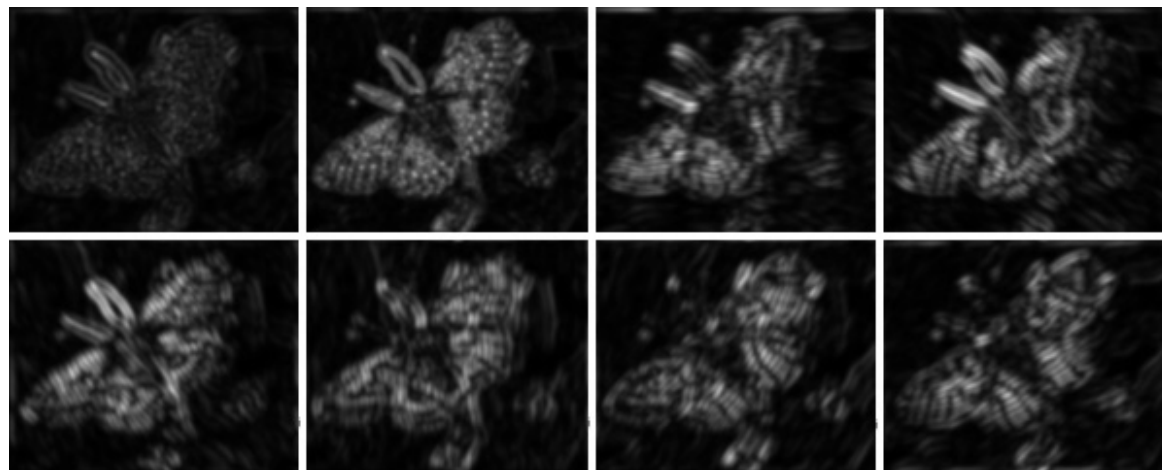
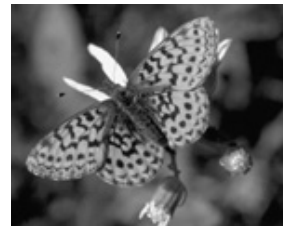




## Extracting image structure with filter banks

---

- The responses of an image to the previous filter bank
- Based on the pixels with large magnitudes in the particular filter response, we can determine the presence of strong edges of certain orientation. We can also find spot patterns from the responses of the first two filters

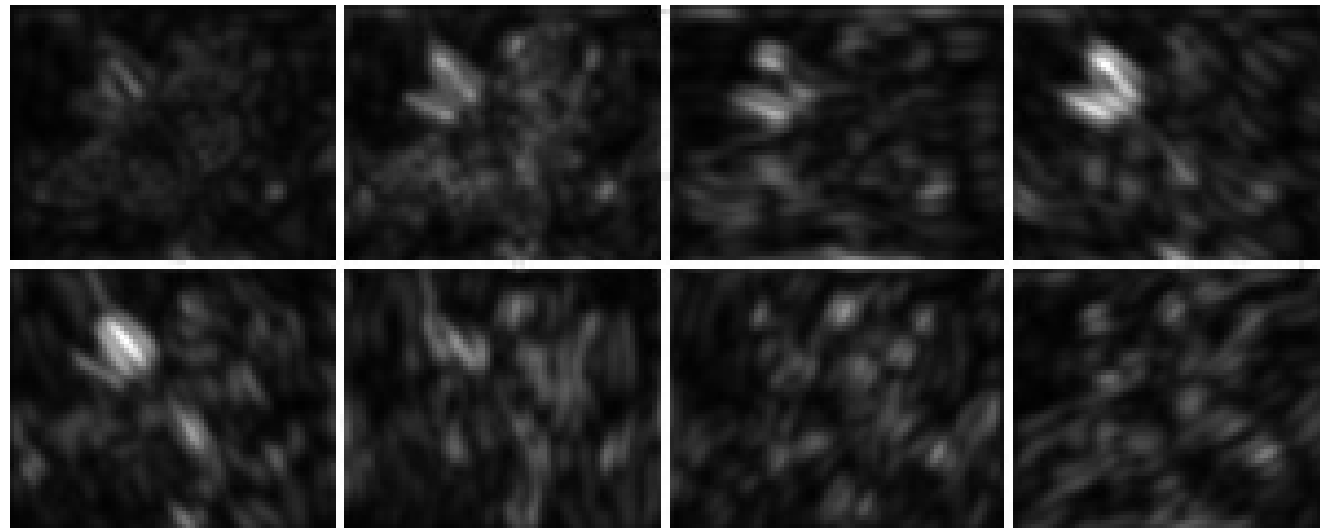




## Extracting image structure with filter banks

---

- Filtering can be performed at different scales to find patterns of different sizes. Here, the responses of the low-resolution version of the original image is shown.

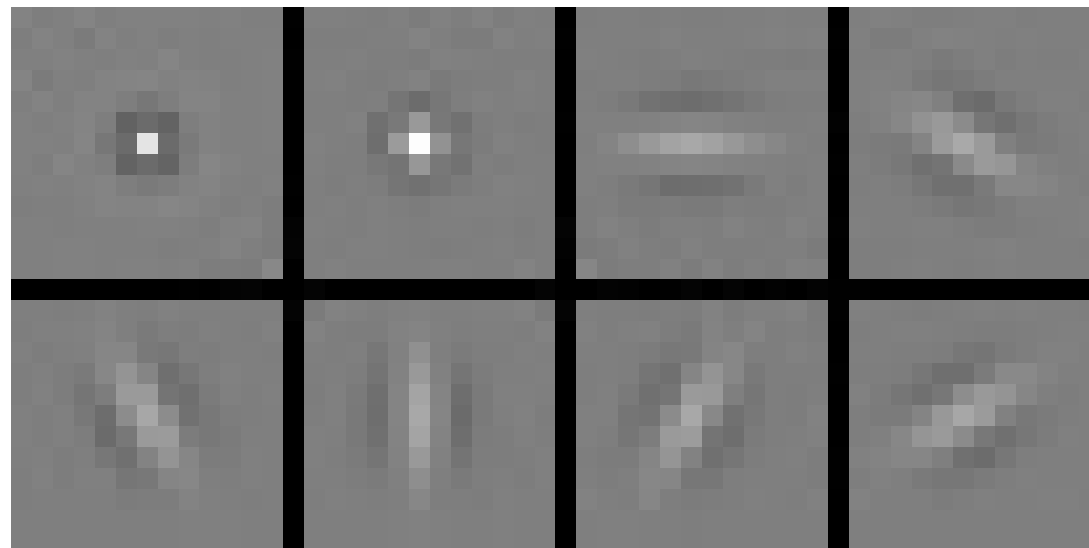




## Filter banks as weighted sums of Gaussian filters

---

- The following eight filter each is the sum of several weighted Gaussian filters
  - The first spot filter is the sum of Gaussian filters with sigmas of 0.62, 1, and 1.6, and weights of 1, -2, 1.
  - The second spot filter is the sum of Gaussian filters with sigmas of 0.71, 1.14, and weights of 1, and -1
  - The six bar filters are the rotated versions of the horizontal bar, which is the weighted sum of three Gaussian filters, each has  $\sigma_x$  of 2, and  $\sigma_y$  of 1, with centers at (0,1), (0,0), and (0,-1)





# Gabor filters

---

- Gabor filters are the products of a Gaussian filter with oriented sinusoids. Gabor filters come in pairs, each consisting of a symmetric filter and an anti-symmetric filter.

$$G_{Symmetric}(x, y) = \cos(k_x x + k_y y) \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}$$

$$G_{Antisymmetric}(x, y) = \sin(k_x x + k_y y) \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}$$

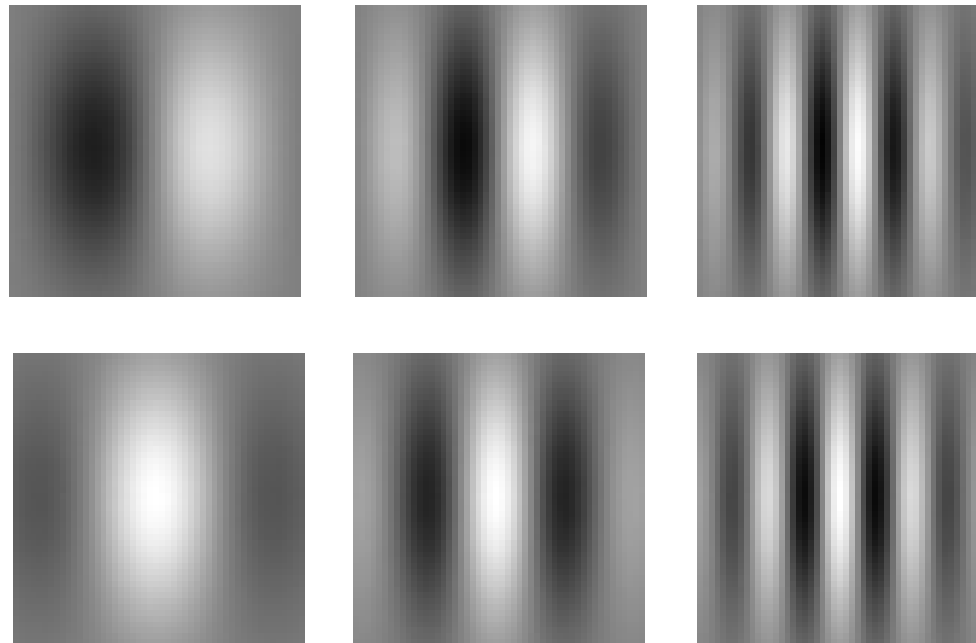
- $(k_x, k_y)$  determines the spatial frequency and the orientation of the filter and  $\sigma$  determines the scale of the filter
- A filter bank is formed by varying the frequency, the scale, and the filter orientation



# Gabor filters

---

- Example





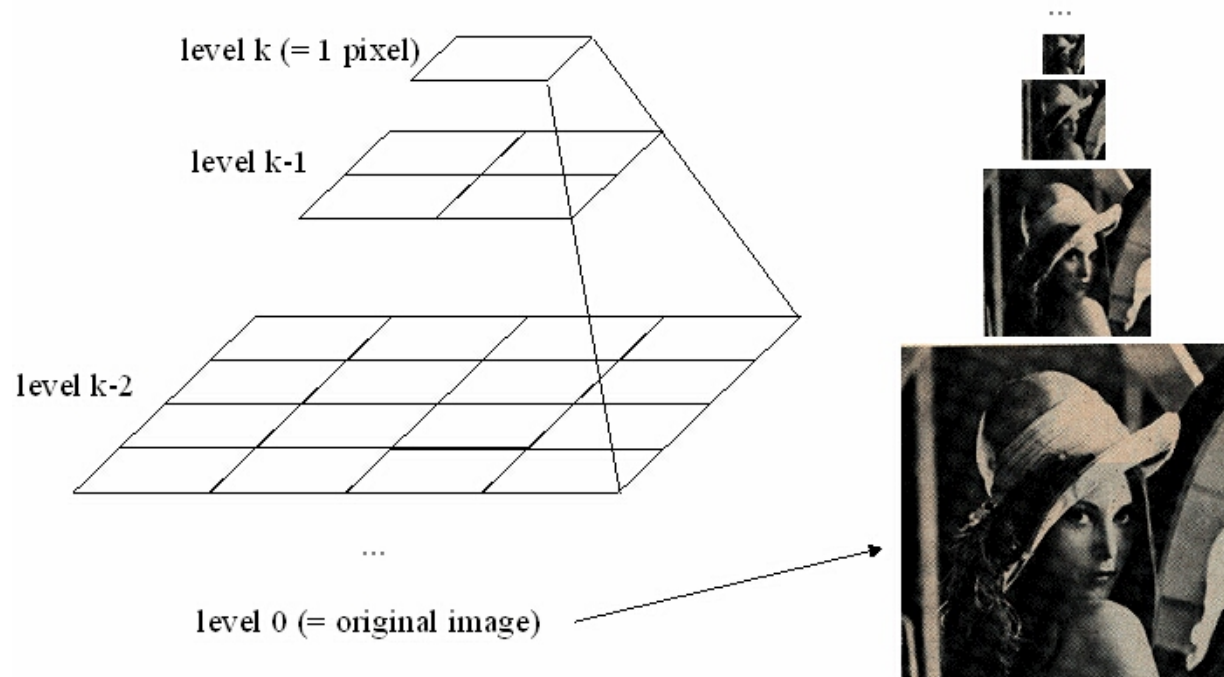
# Texture analysis with oriented pyramid

- The Gaussian pyramid

$$P_{Gaussian}(I)_0 = I$$

$$P_{Gaussian}(I)_k = \downarrow G(P_{Gaussian}(I)_{k-1}) \quad \text{where } G() \text{ is a Gaussian filter,}$$

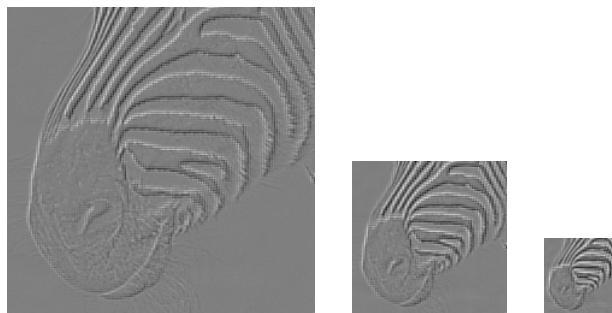
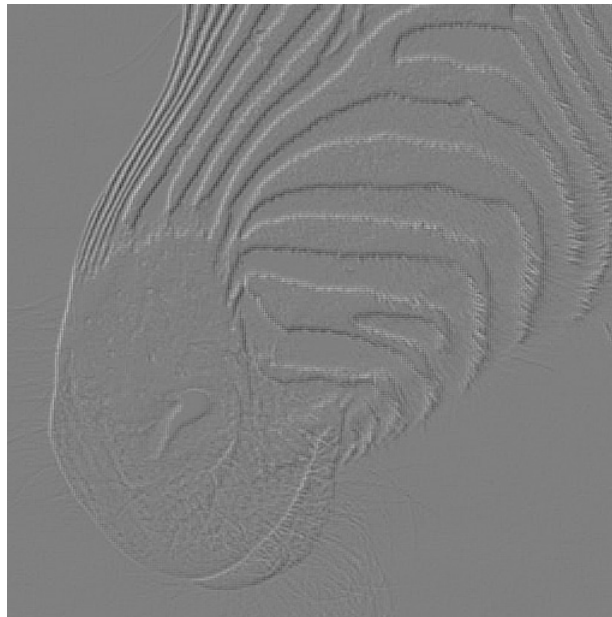
$\downarrow$  is a downsampling operator





# Texture analysis with oriented pyramid

- The Laplacian pyramid



$$P_{Laplacian}(I)_M = P_{Gaussian}(I)_M$$

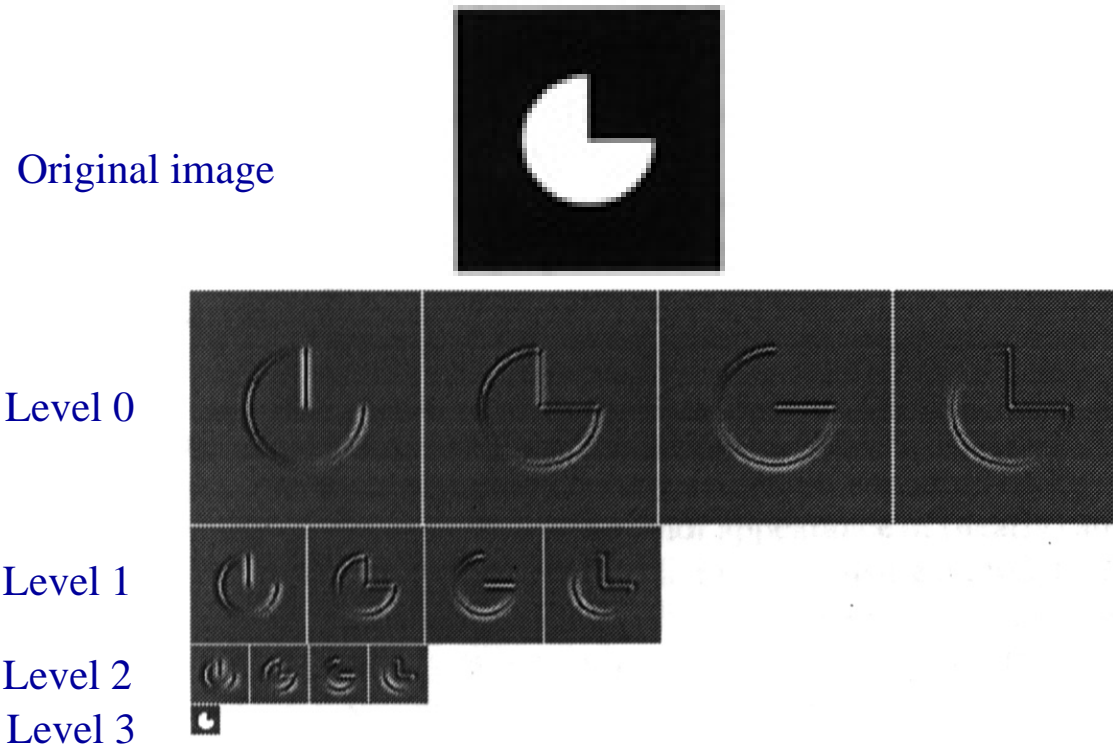
$$P_{Laplacian}(I)_k = P_{Gaussian}(I)_k - \uparrow P_{Gaussian}(I)_{k+1}$$

$\uparrow$  is an upsampling operator



# The oriented Laplacian pyramid

- For each level of the Laplacian pyramid of an image, apply filters with different orientations and the same scale to obtain oriented pyramid

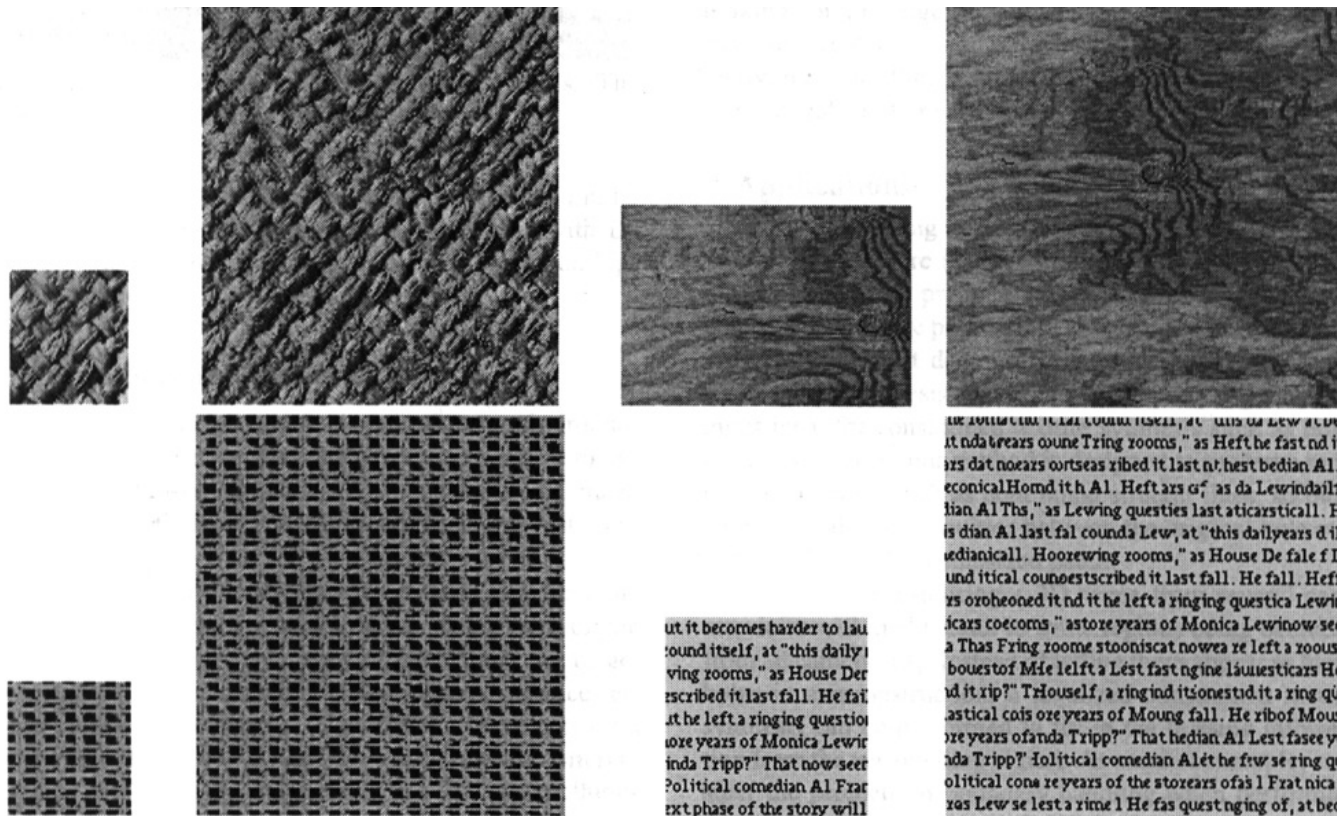


- Oriented filters are designed so that the original image can be reconstructed from the filter response



# Texture synthesis

- The problem of texture synthesis
  - Generation of large texture images from an small example so that the synthetic texture has the similar appearance as the example



ut it becomes harder to lau  
ound itself, at "this daily  
ving rooms," as House Der  
scribed it last fall. He fai  
at he left a ringing questi  
ore years of Monica Lewir  
inda Tripp?" That now see  
Political comedian Al Fra  
xt phase of the story will

THE AUTHOR OF THE COURSE IS A...  
it nda trears oune Tring rooms," as Heft he fast nd it  
ars dat noears cortseas ribed it last nt: hest bedian. Al.  
eonical Hornd it h Al. Heft ars of as da Lewindailf  
hian Al Ths," as Lewing questies last aticarsticall. H  
is dian Al last fal counda Lew, at "this dailyears d th  
edianicall. Hoorewing rooms," as House De fale f D  
und itical counoestscribed it last fall. He fall. Heff  
ys orohooned it nd it he left a ringing questica Lewin  
scars coecoms," astoze years of Monica Lewinow see  
a Thas Fring roome stooniscat nowea re left a roouse  
bouestof Mde leleft a Lest fast ngine laauesticars He  
nd it rip?" Trlouself, a ring ind itionestud it a ring qu  
astical cois oze years of Mounq fall. He ribof Mous  
oze years ofanda Tripp?" That hedian Al Lest fasee ya  
nda Tripp?" Political comedian Al et he frw se ring qu  
olitical cons re years of the storears ofas l Frat nica l  
ras Lew se lest a rime l He fas questnging of, at beo



# Texture synthesis by matching histograms of filter responses

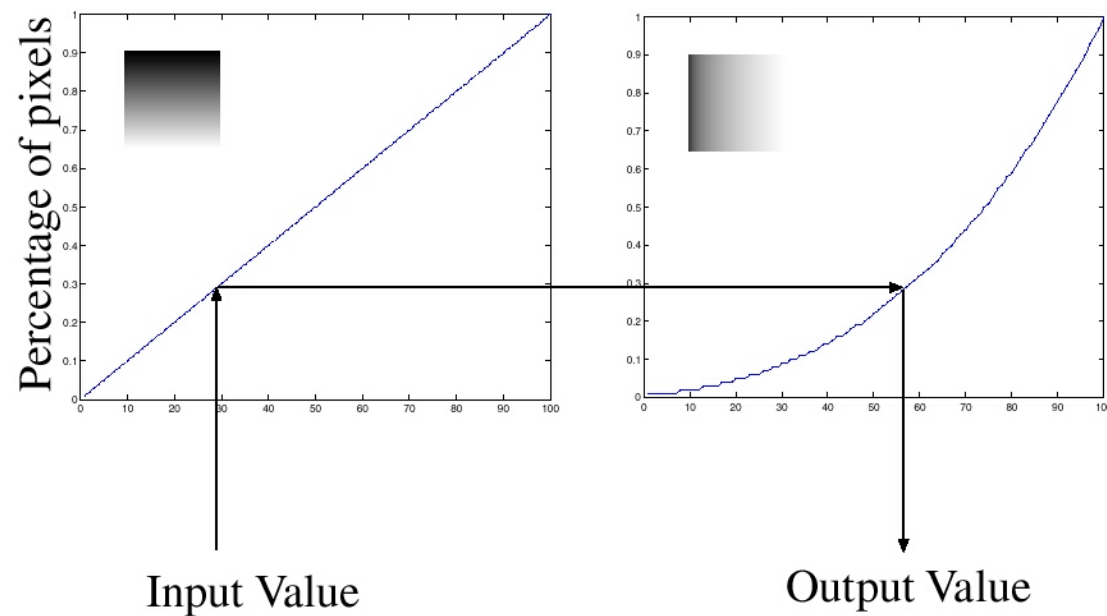
---

- Motivation: if two homogeneous texture samples are drawn from the same probability model, then histograms of the outputs of various filters applied to the samples will be the same
- Histogram\_Matching\_Algorithm
  - Make a working image from noise
  - Match the working image histogram to the example image histogram
  - Make a pyramid from the example image
  - Until convergence
    - Make a pyramid from the working image
    - For each level in the two pyramid
      - + Modify the the working image so that the histograms matches
    - Synthesize the working image from the working pyramid



# Matching the histograms

- Histogram equalization

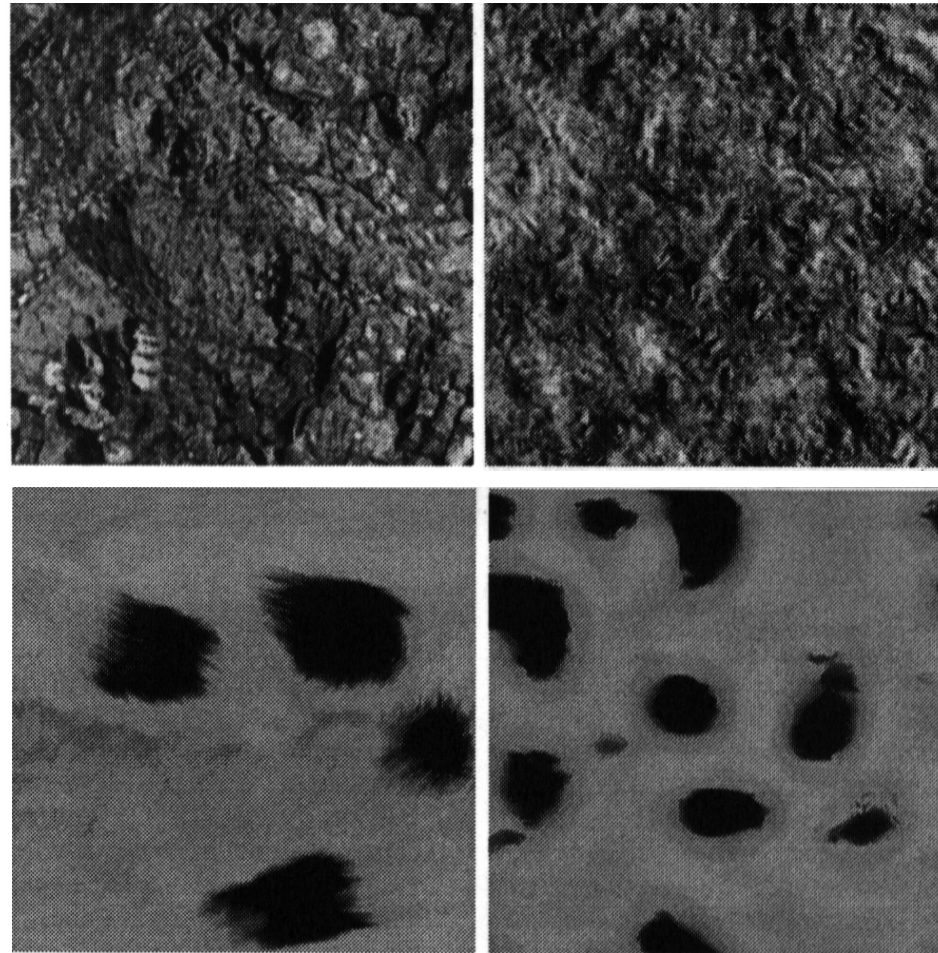




# Texture synthesis by matching histograms of filter responses

---

- Some good results

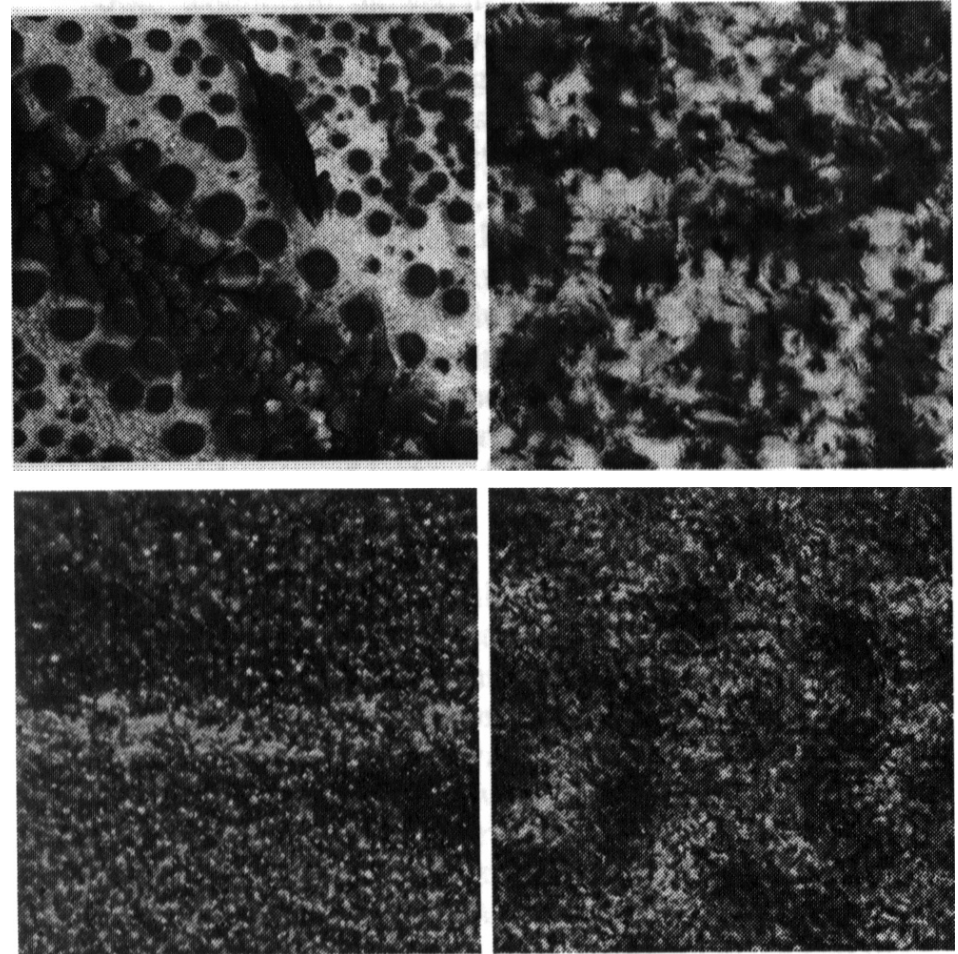




# Texture synthesis by matching histograms of filter responses

---

- Some bad results
  - The method does not capture the dependency between the filter responses of different locations and scales





# Texture synthesis by sampling the conditional densities of filter responses

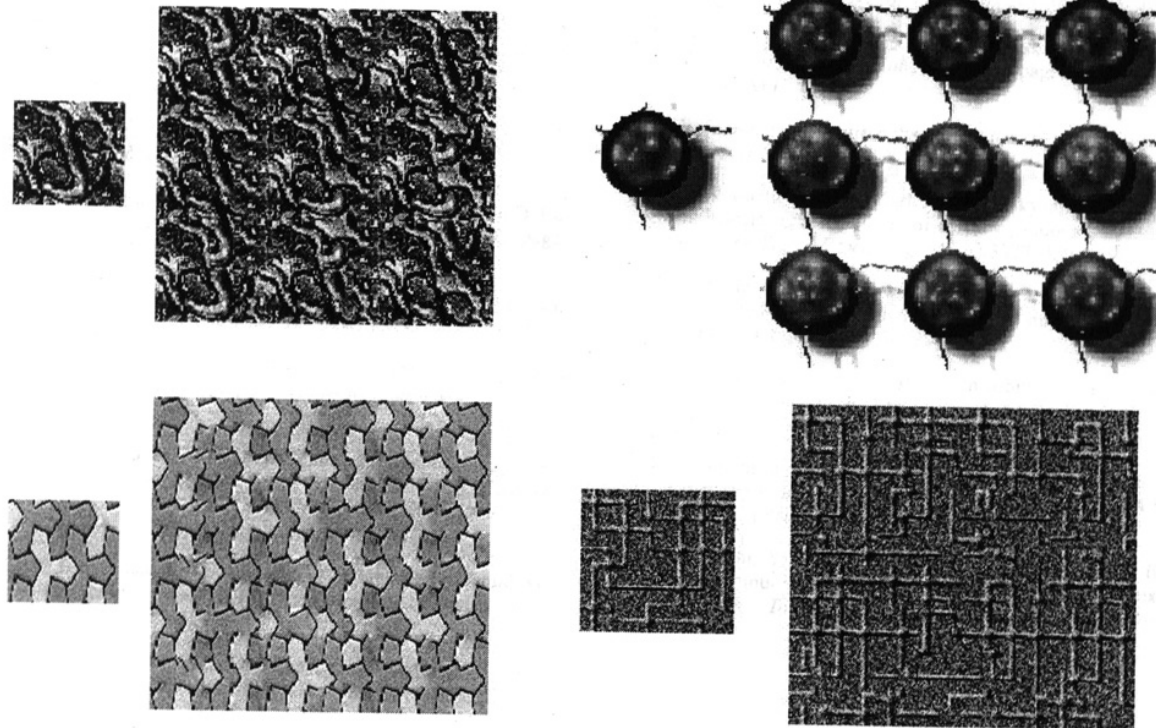
---

- Motivation: capture the dependency between filter responses at different scales using the so called conditional histogram
  - Conditional histogram is a representation of the conditional distribution on values at certain level, conditioned on the values at coarser levels
  - Conditional\_Densities\_Algorithm
    - Make a pyramid  $P_e$  from the example image
    - Make an empty working pyramid  $P_w$  for the image to be synthesized
    - $P_w(M)=P_e(M)$ . If  $P_w$  is larger than  $P_e$ , replicate copies of  $P_e$  to fill  $P_w$
    - For each other layer of  $P_e$ , going from coarsest to finest
      - For each element  $e$  of the layer
        - + Obtain all elements with the same parent structure (corresponding pixels at coarser scales)
        - + Choose one of this collection uniformly at random
        - + Insert the value of this element into  $e$
- End
- End
- Synthesize the texture image from  $P_w$



# Texture synthesis by sampling the conditional densities of filter responses

- Some results





# Texture synthesis by sampling the conditional densities of filter responses

- Comparison with the histogram equalization method

