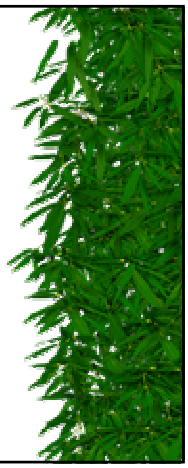
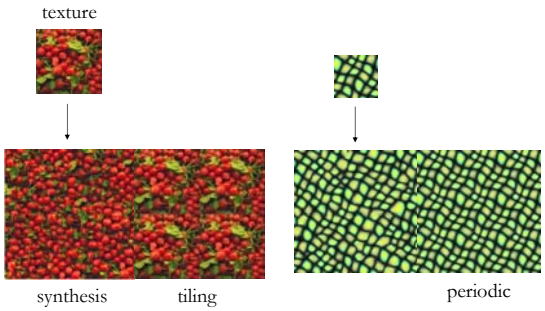
# Texture Synthesis an overview

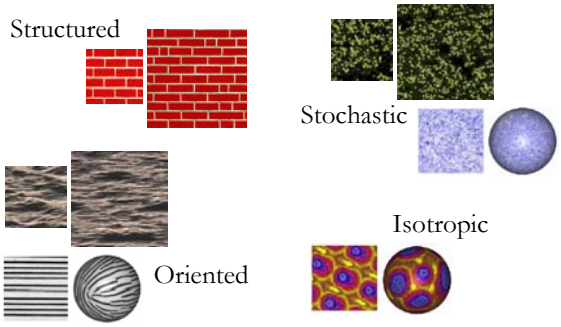


Iddo Drori
Tel-Aviv University, 2001

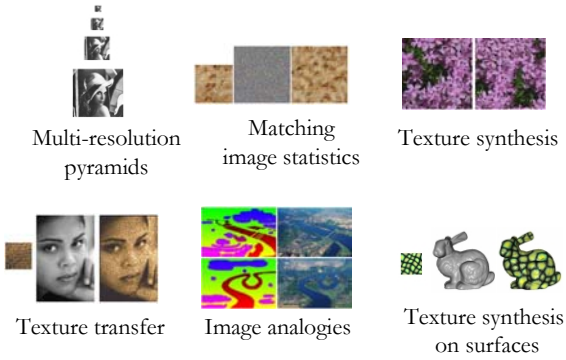
## Introduction



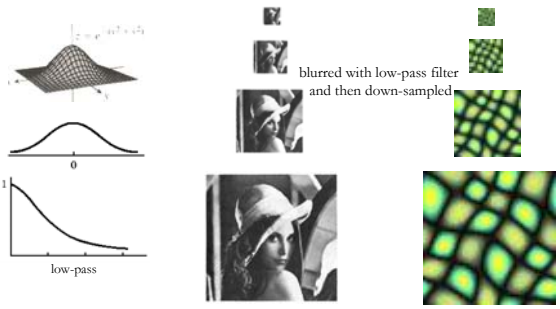

## Texture properties




## Overview

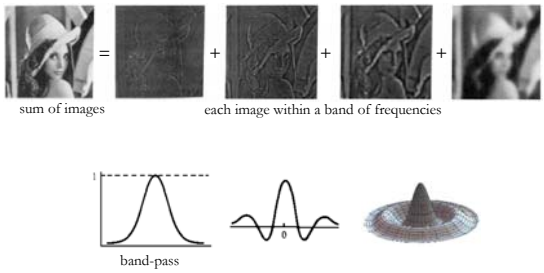



## Gaussian pyramid



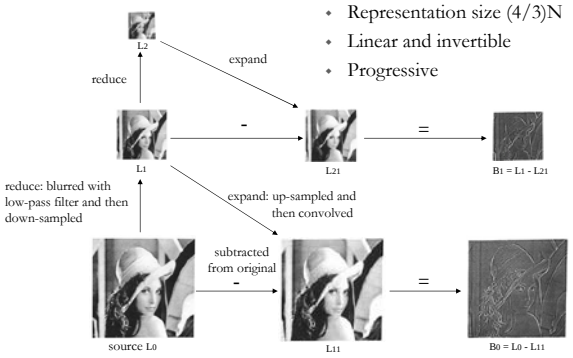

- Images contain objects and features of different size, which may be viewed over a range of distances, and therefore a transformation should analyze the image at different scales.

## Laplacian pyramid



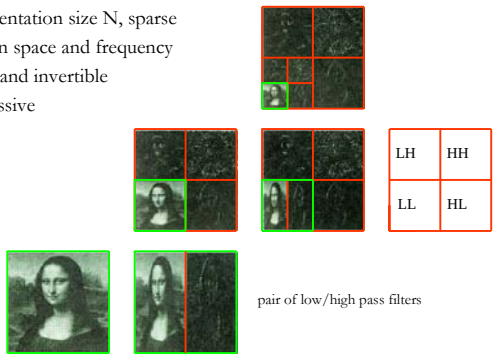
## Laplacian pyramid

- Representation size (4/3)N
- Linear and invertible
- Progressive




## Wavelet

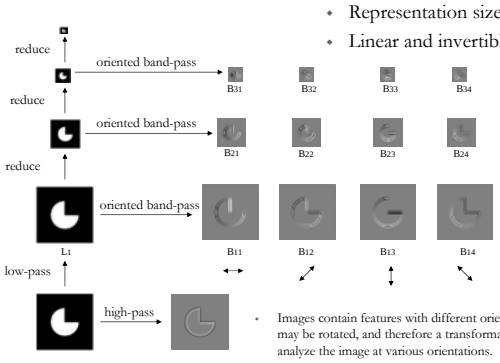
- Representation size N, sparse
- Local in space and frequency
- Linear and invertible
- Progressive




pair of low/high pass filters

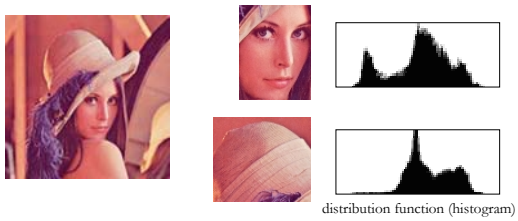
## Steerable pyramid

- Representation size k(4/3)N
- Linear and invertible




- Images contain features with different orientation, and may be rotated, and therefore a transformation should analyze the image at various orientations.

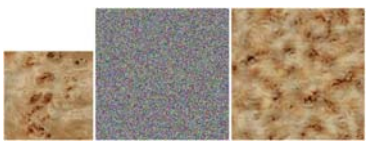
## Image statistics



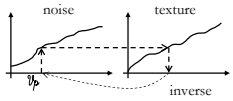
distribution function (histogram)

- Image statistics vary locally.
- Images have sharp local features like edges as well as large homogeneous regions, and generally defy simple statistical models for their structure.

## Matching image statistics



- Decompose noise and texture image into pyramids.
- Match between distributions (histograms) of corresponding pyramid levels.



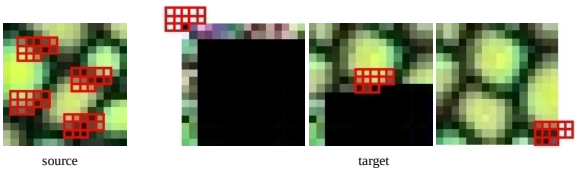

cumulative distribution functions

- Reconstruct noise pyramid.

## Texture model

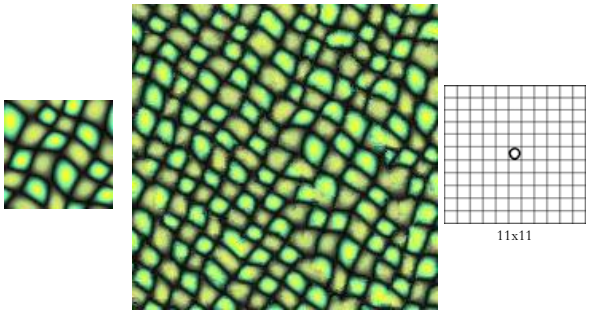
- Stationary - under a proper window size, the observable portion always appears similar.
- Local - each pixel is predictable from a small set of neighboring pixels and independent of the rest of the image.

## Algorithm




- Assign some random pixels (from source) to target.
- Synthesize target one pixel at a time.
- Find best matching neighborhood in source and assign that pixel to target.


## Synthesis results with different neighborhood sizes



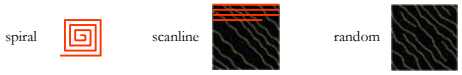

- To capture the structure with a small neighborhood→multi-resolution

## Multi-resolution algorithm

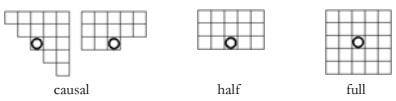
- Decompose source into (Gaussian) pyramid.
- For each level from coarse to fine, and each target pixel in scanline order
  - Search for best matching neighborhoods at multiple resolutions.

## Variants

1. Synthesis **Unit**: pixel/block
2. **Order** to synthesize target



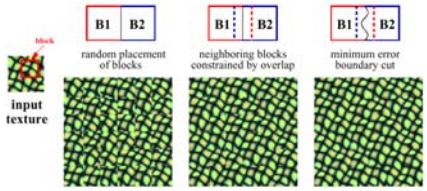

3. **Neighborhood** to match




4. **Search method** of source texture: naive/random/coherent/TSVQ
5. **Similarity metric:** weighted sum of square differences/other

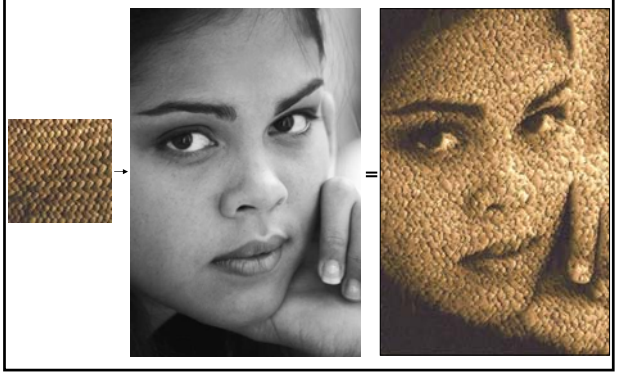
## Image Quilting

## Algorithm



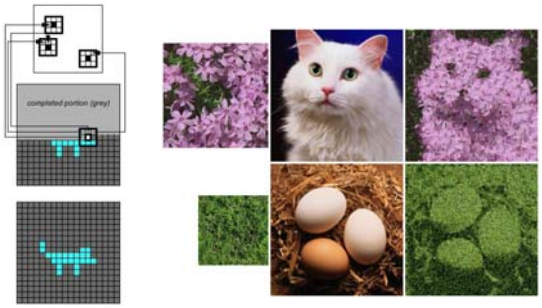

- Go through the image to be synthesized in scanline order in steps of block-overlap.
- For every location, search the input texture for a set of blocks that satisfy the overlap constraints within some error tolerance. Randomly pick one.
- Compute the error surface between the newly chosen block and the old blocks at the overlap region. Find the minimum cost path along this surface (dynamic programming/Dijkstra/greedy) and make it the boundary of the new block. Paste block onto texture.

## Texture transfer

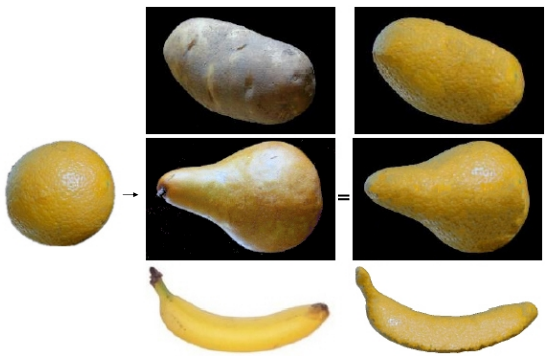


## Applications - Texture Transfer

## Algorithm



α(legitimate texture) + (1−α)(match corresponding image)



## Image analogies (filter by example)

**A to A' like B to B'**

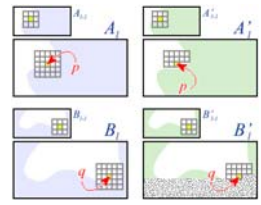
## $A_1,\ldots,A_n : A_1',\ldots,A_n' :: B : B'$


## Algorithm

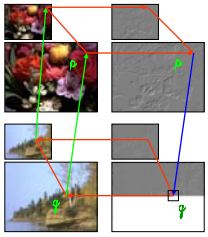
- decompose A, A', B into Gaussian pyramids.
- for each level $\ell$ from coarse to fine, for each pixel q in $B'\ell$

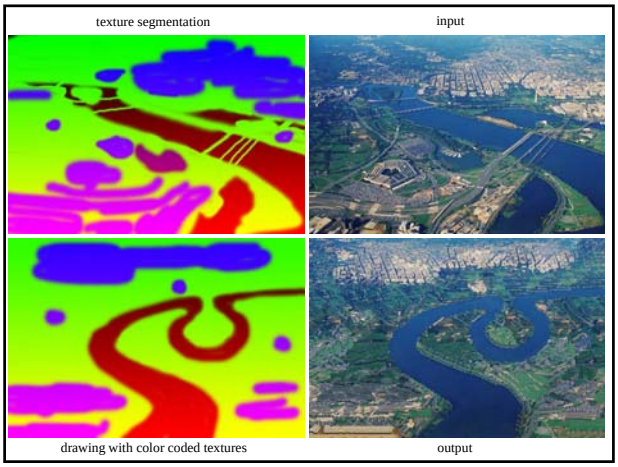
  $p = \text{match}(A\ell\text{-}1, A\ell, A'\ell\text{-}1, A'\ell, B\ell\text{-}1, B\ell, B'\ell\text{-}1, B'\ell, q)$

  $B'\ell(q) = A'\ell(p)$



assumes dense correspondence between A and A'



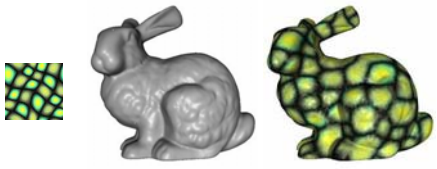



## Applications – Artistic Filters (Cont.)


## Applications– "Texture By Number"

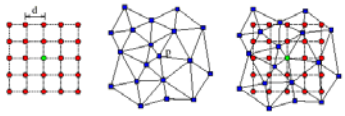
By color-labeling source image parts a realistic synthesized image can be created


## Texture synthesis on surface



## Extension of 2D algorithm

- Mesh pyramid
- Sampling - uniform density to minimize stretching an distortion.
- Local texture orientation - handle oriented textures, avoid seems.
- Synthesis order - define neighbors.

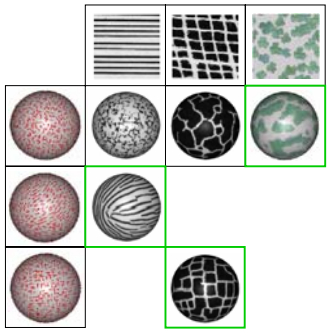


## Algorithm

- Decompose texture into image pyramid and model into mesh pyramid.
- Retile surfaces of mesh from fine to coarse, uniformly distributing the mesh vertices.
- Assign texture orientation to each mesh vertex.
- Assign a color to each mesh vertex, randomly from the corresponding level in the image pyramid.
- For each level from coarse to fine, and each mesh vertex
  - locally flatten and resample mesh neighborhood
  - Search for best matching neighborhoods

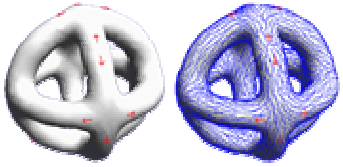
## Local texture orientation

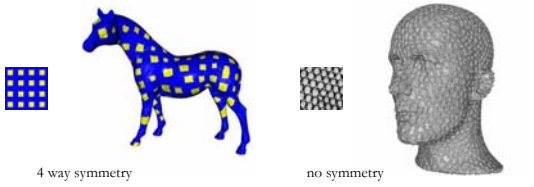
- Random
- Symmetry - 2 way
- Symmetry - 4 way



## Local texture orientation

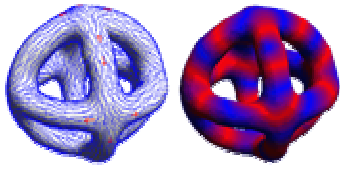
- User defined and interpolation

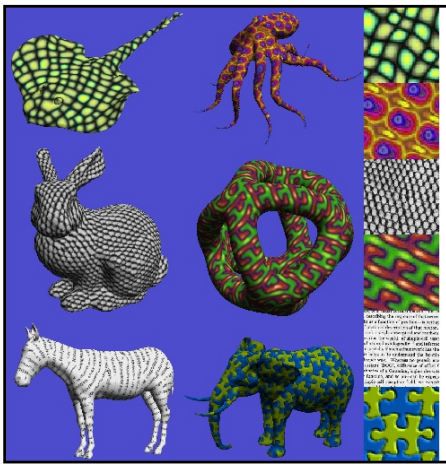



## Synthesis order

- Random
- Defined by orientation field





texture matches user specified orientation

isotropic texture onto model with branching surfaces

texture does not tile scale rows of model

periodic texture onto model with complex topology

# References


- Pyramid-Based Texture Analysis/Synthesis, David Heeger and James Bergen, SIGGRAPH 1995.
- Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images, Jeremy De Bonet, SIGGRAPH 1997.
- Texture Synthesis by Non-parametric Sampling, Alexei Efros and Thomas Leung, ICCV 1999.
- Fast Texture Synthesis using Tree-structured Vector Quantization, Li-Yi Wei and Mark Levoy, SIGGRAPH 2000.
- Synthesizing Natural Textures, Michael Ashikmin, I3DG 2001.
- Image Quilting for Texture Synthesis and Transfer, Alexei Efros and William Freeman, SIGGRAPH 2001.
- Image Analogies, Aaron Hertzmann, Charles Jacobs, Nuria Oliver, Brian Curless, David Salesin, SIGGRAPH 2001.
- Learning Low-Level Vision, William Freeman, Egon Pasztor, Owen Carmichael, IJCV 2000.
- Texture Synthesis on Surfaces, Greg Turk, SIGGRAPH 2001.
- Texture Synthesis over Arbitrary Manifold Surfaces, Li-Yi Wei and Mark Levoy, SIGGRAPH 2001.
- Texture and Shape Synthesis on Surfaces, Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin, EGRW 2001.