On Generalizations of Weighted Finite Automata and Graphics Applications

Jürgen Albert
Informatik II
University of Würzburg

Contents

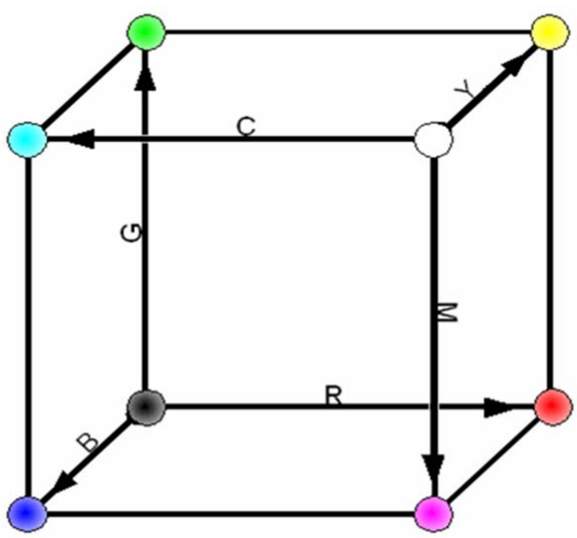
- Color, grayscale and bi-level images
- Color-cube
- Finite state acceptors, bi-level images
- Addressing in the unit-interval
- Bintrees
- Cartoons and video-clips
- Parametric WFA,
- Motivation, examples
- Iterated Function Systems
- Splines, texturing
- Open questions

Color Depths

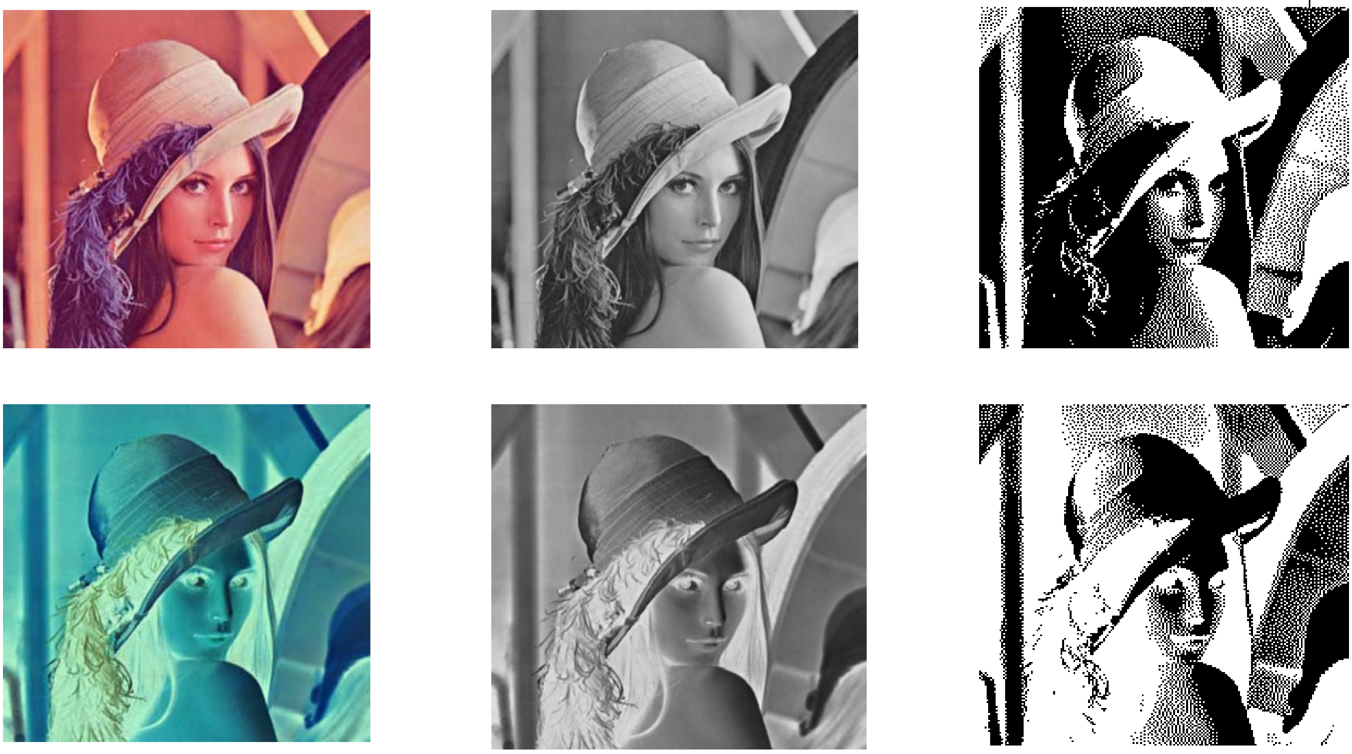
24 Bits („true color“) vs. 4 Bits (color table) vs. 1 Bit

Color-Cube

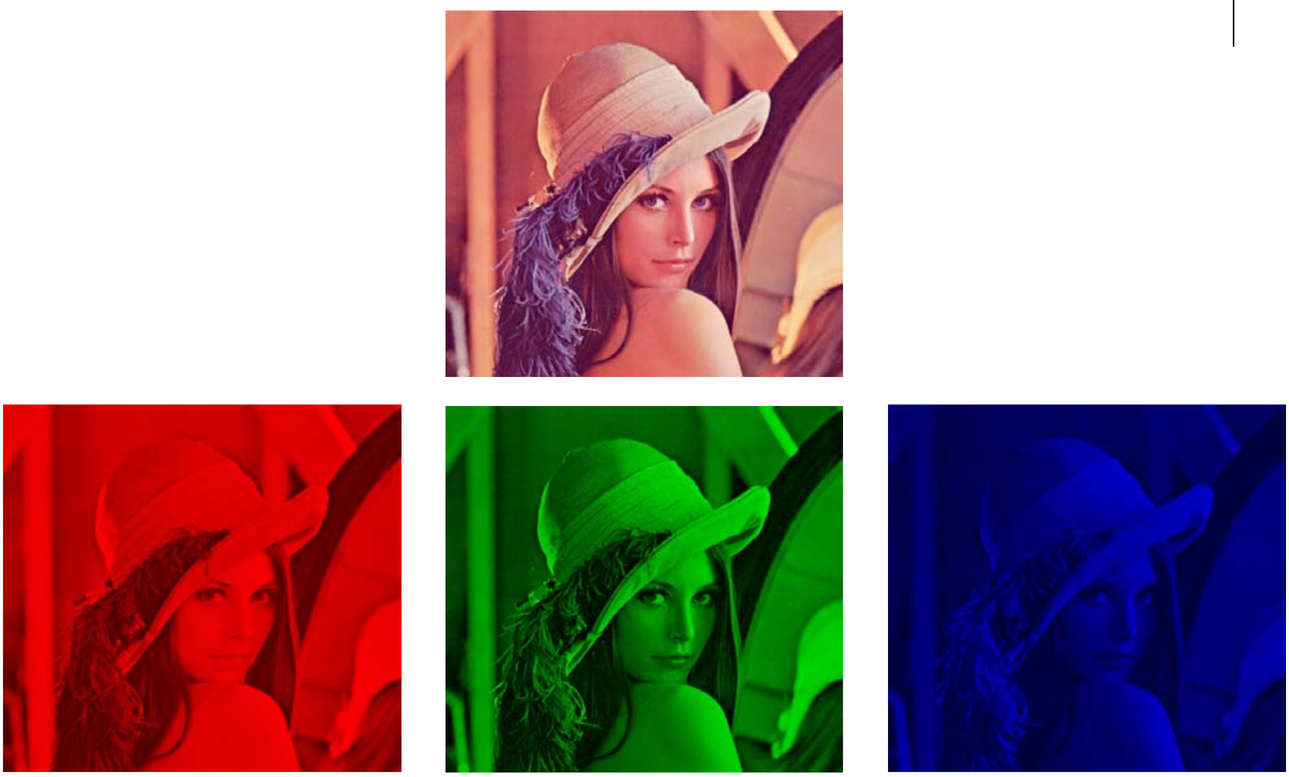
For custom digital photography in general 8 Bits for each component:
Black = (0, 0, 0), White = (255, 255, 255)



Negatives - Complements to White



Separation into Components (Red, Green, Blue)



Colors + Global Transparency, „Alpha-Blending“ (1)

Colors + Global Transparency, „Alpha-Blending“ (2)

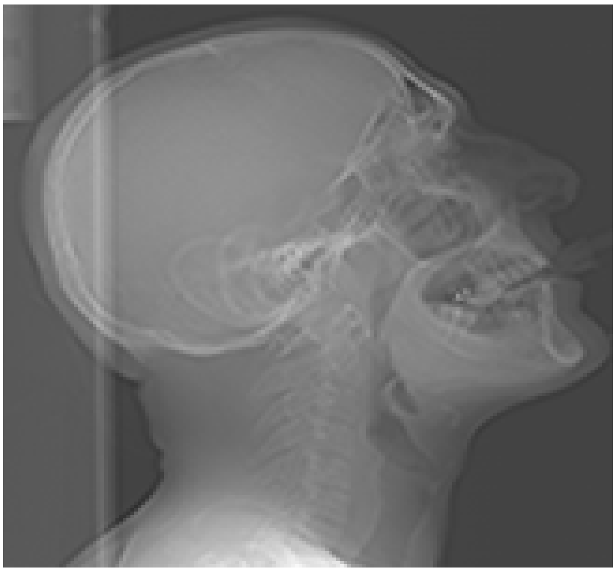
1

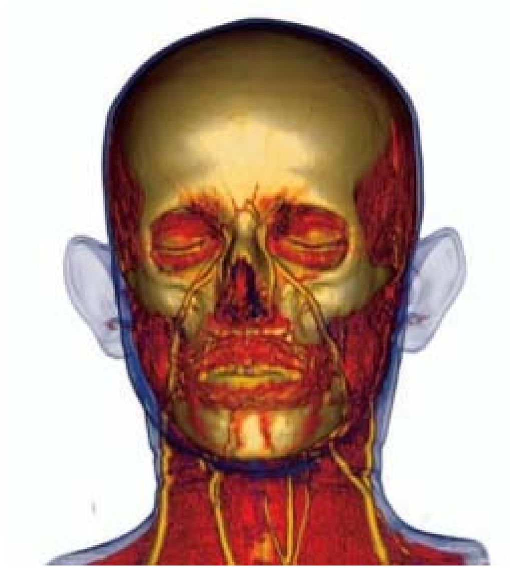
Von Sant Kilian dem Ersten Bischoff Zu
Wirtzburg

DER HEILIG KILIAN, den man auch sonst (wie hernach gesagt wierdt) *Killenam* genent hat, ist in dem Kunigraich Schottlandt, aus Sachßischen ansehenlichen eltteren geborn, vnd von denselbigen in seiner Kindthait vnd Jugent Zu der Schulen gethan worden, da er in allerley Kunsten, Sonderlich aber in der Heyligen geschrifft, so wol gelernet vnd verfarn hate, das er Letzlich Zu ainem Abt, das ist Ainem Vatter, Vorsteher vnd Regirer, d(er) Andern

Medical Imaging, Varying Transparency Values

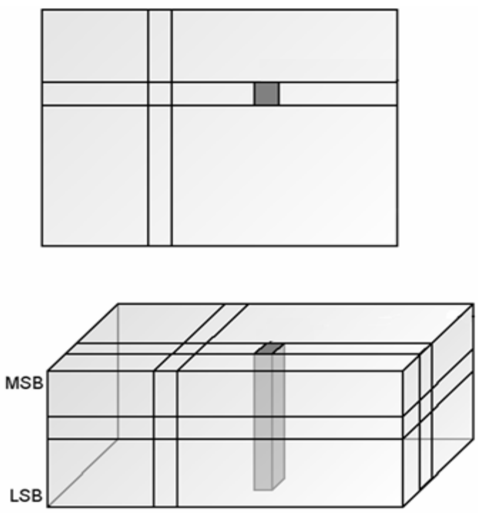
Voxel densities often as 12-Bit-Integers, selecting density-intervals and rendering e.g. as in the „Visible Human Project“





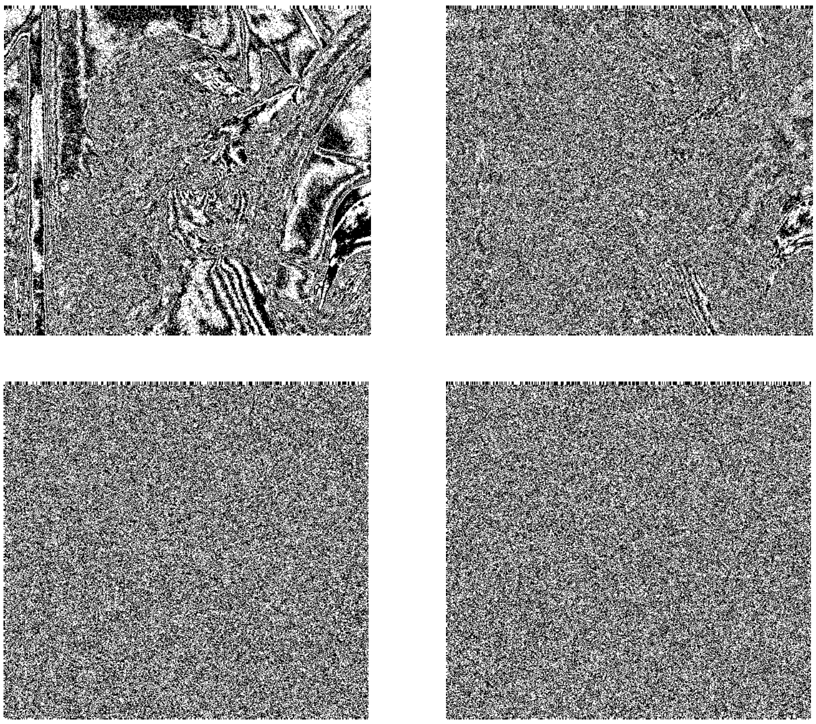
Bitplane Encoding

- Each pixel-value is specified by its color components and its transparency via a fixed number of bits.
- Each of these bit-positions defines a bit-plane.
- The whole image is representable as a stack of bi-level images.



8-Bit-Grayscale: Most Significant Bit-planes

8-Bit-Grayscale: Least Significant Bit-planes



Words and Intervals

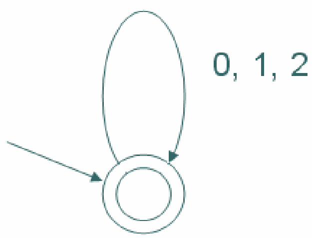
- For any alphabet S = {0, 1, …, p-1}
  a word $w = a_1 a_2 \ldots a_n$ in S^* is associated with the half-open interval
  $[\sum a_i p^{-i}, \sum a_i p^{-i} + p^{-n})$

- In the p-adic number-system this is denoted by
  $[0.a_1 a_2 \ldots a_n, 0.a_1 a_2 \ldots a_n + 0.00 \ldots 1)$

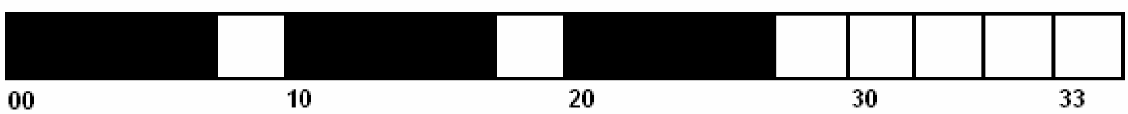
- E.g. for p = 4 the word w = 103 addresses the half-open interval
  [0.103, 0.110) of length 4^{-3}

Acceptors for Bi-level Images

Let $A = (\{q_1\}, \{0, 1, 2, 3\}, M, q_1)$ be a finite state acceptor with transitions

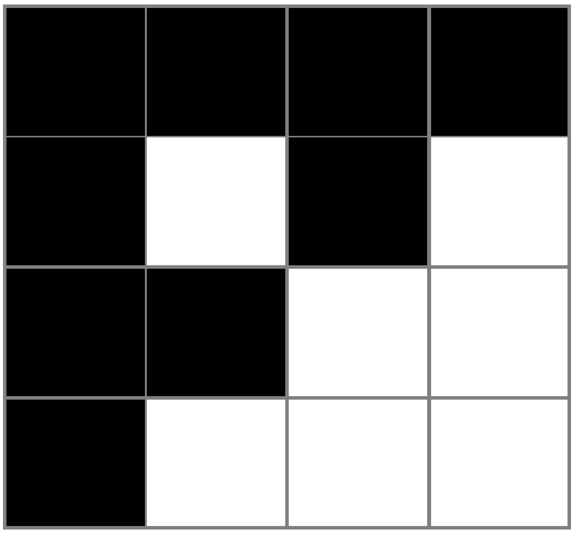


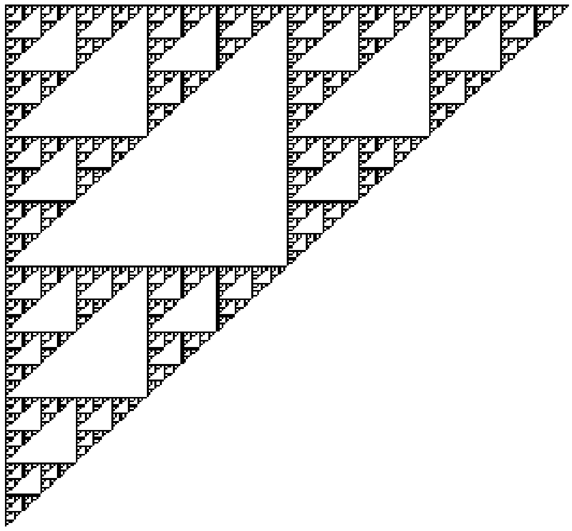
The acceptance pattern for sequences over {0, 1, 2, 3} of length 2 is



2D-Interpretation

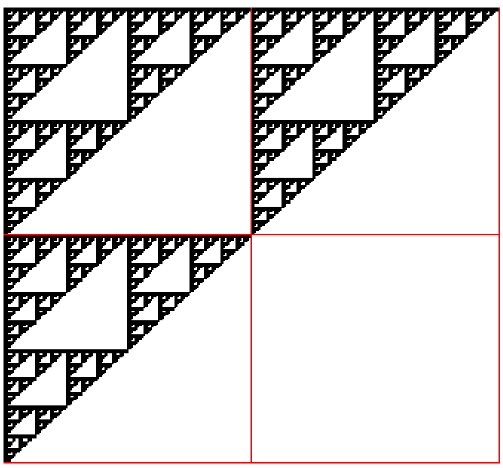
Acceptance patterns for automaton A and input-lengths 2 and 8.
Generation of bi-level images:

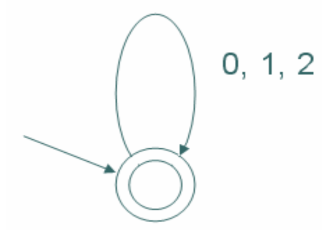




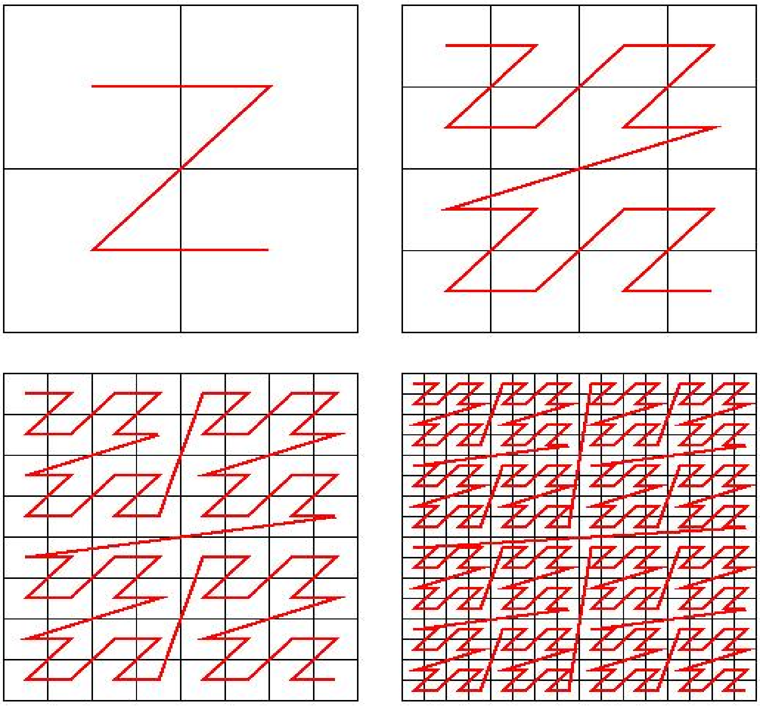
Automaton Inference

For simple cases a generating automaton can directly be inferred from the image. „Zooming“ into the quadrants 0, 1 and 2 yields scaled copies of the complete picture.





Morton-Order (Z-Order)

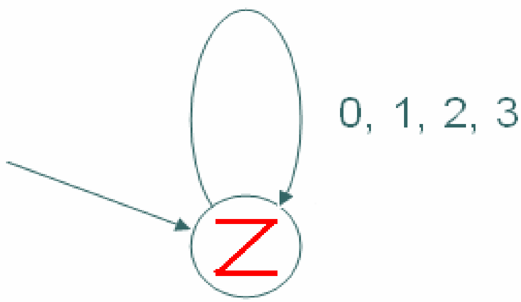


Addressing $2^k \times 2^k$ pixels in hierarchical and self-similar way.

Rotations and flips of Z are also very common in practice

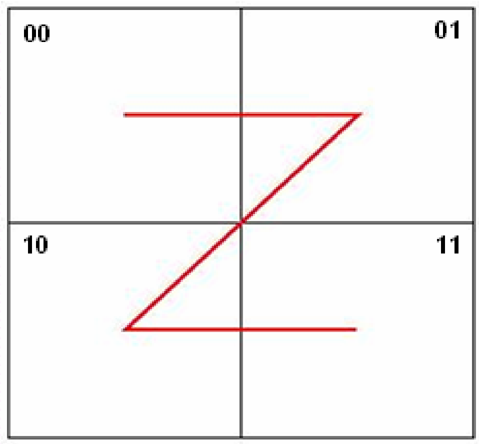
Self-Similar Morton-Order

Self-similarity holds for the traversals of the Morton-Order in a trivial manner:



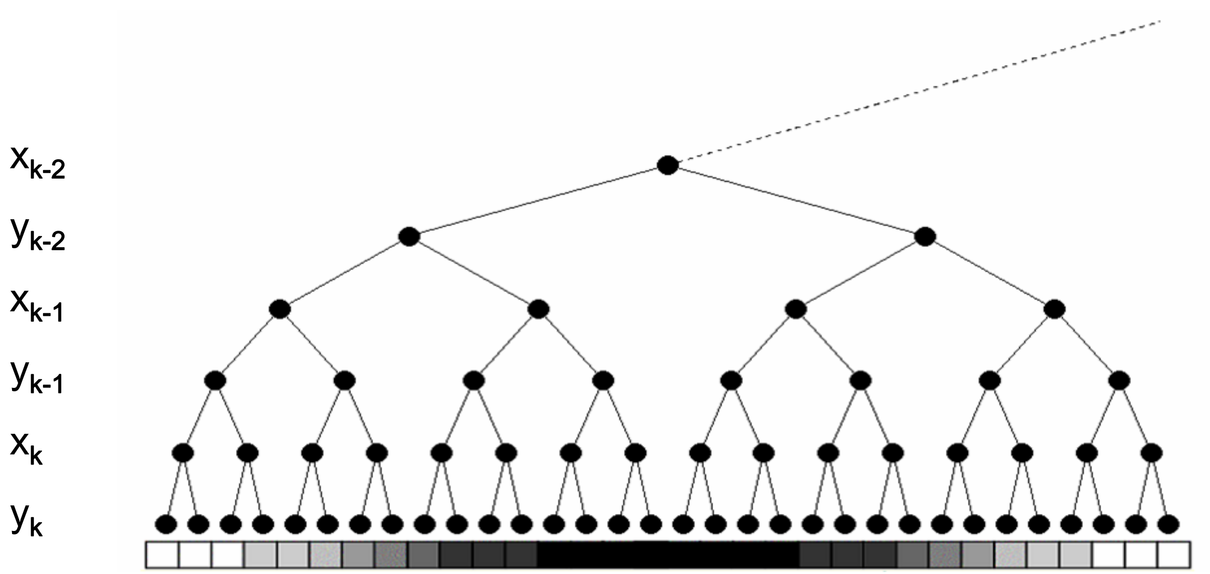
Binary Alphabet

Addressing pixels / voxels requires only a binary alphabet „with interleaving of the dimensions“:



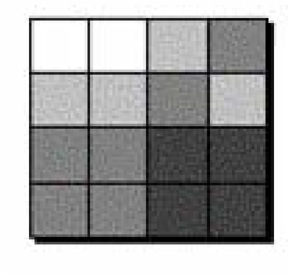
2-D Bintree-Interpretation

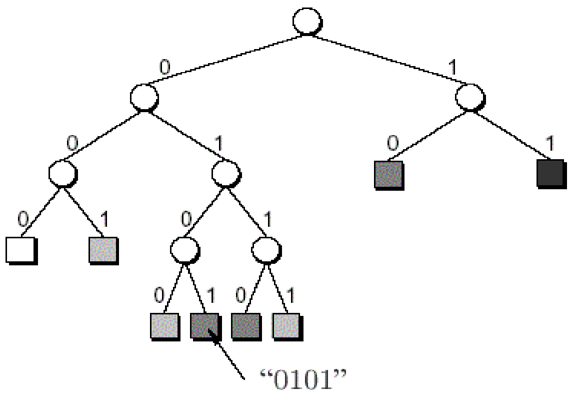
Any image (black/white, grayscale, color) can be addressed by a bintree with interleaving the x- and y-bits of the addresses:



Bintree Addressing

For any image, a tree can define a corresponding finite automaton: In its simplest form each pixel corresponds to an accepting state in a DFA, the root is the initial state.





WFA - Definition

A **Weighted Finite Automaton** (**WFA**) $A = (Q, \Sigma, W_\Sigma, I, F)$ consists of:

- Set of **states** $Q = \{q_1, ..., q_n\}$
- **Input-alphabet** $\Sigma = \{0,1,\ldots, p-1\}$, ({0,1} is sufficient)
- Set of **weight-matrices** $W_\Sigma = \{W_0, W_1, ...\}$, where $W_i \in IR^{n \times n}$
- **Initial distribution** (row vector) $I \in IR^n$
- **Final distribution** (column vector) $F \in IR^n$

The image attached to the input-sequence $w = i_1\, i_2 \ldots i_t$ („the address“) is computed by A as:

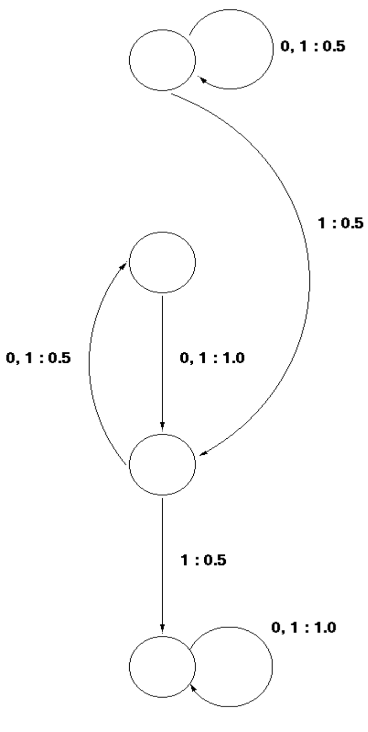
$$f_A(w) = I \bullet W_{i1} \bullet W_{i2} \bullet \ldots \bullet W_{it} \bullet F$$

WFA - Example

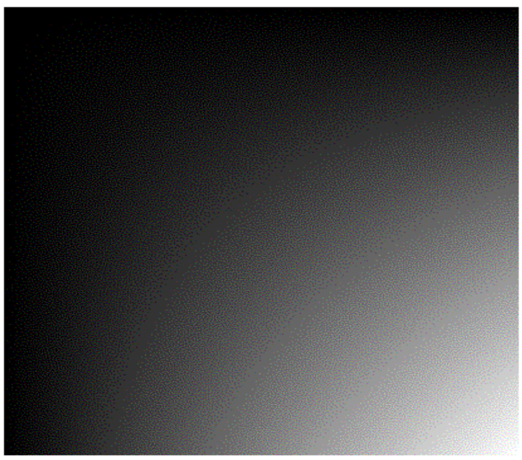
Set of states	$Q = \{q_1, q_2, q_3, q_4\}$
Set of input-symbols	$\Sigma = \{0, 1\}$
Transition-matrices	$W = \{M_0, M_1\}$
Initial distribution	$I = (0, 0, 0, 1)$
Final distribution	$F = (1, 0.5, 0.5, 0.25)^T$

$$M_0 = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.5 \end{pmatrix}, M_1 = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.5 \end{pmatrix}$$

WFA - Example

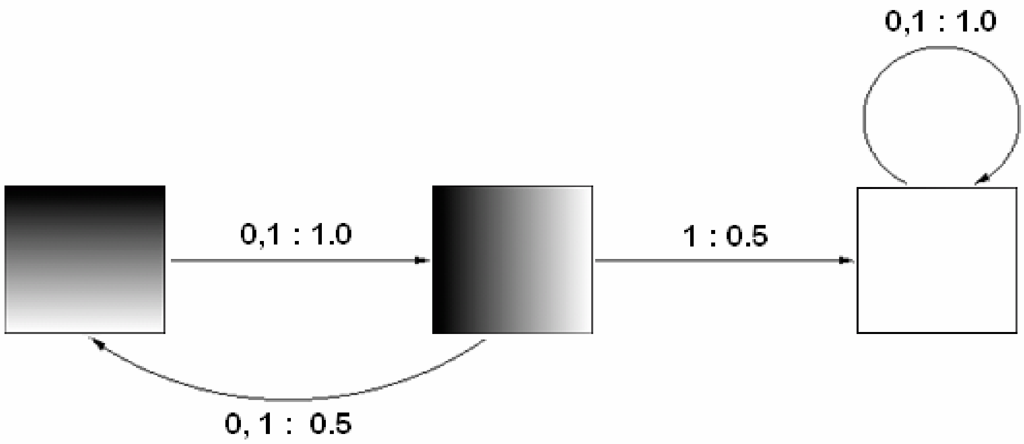


Graph for weight-matrices and grayscale image of the computed function $x \bullet y$ for $x, y \in [0, 1)$



WFA – Example, Subautomata

The WFA for x • y contains the subautomata for the linear slopes in y and x and the constant 1.



WFA – Polynomials

$p_m(x) = x^m$, $m \geq 1$. Let $b_i \in \{0,1\}$ and

$x_1 = 0.b_1\, b_2\, b_3 \ldots b_t$, $x_2 = 0.b_2\, b_3 \ldots b_t$,

$$x_1^m = \left(\frac{1}{2}b_1 + \frac{1}{2}x_2\right)^m = \frac{1}{2^m}\sum_{i=0}^{m}\binom{m}{i} b_1^{m-i} \cdot x_2^i.$$

For $b_1 = 0$: $$x_1^m = \frac{1}{2^m}x_2^m$$

For $b_1 = 1$: $$x_1^m = \frac{\binom{m}{0}}{2^m}x_2^0 + \frac{\binom{m}{1}}{2^m}x_2^1 + \cdots + \frac{\binom{m}{m}}{2^m}x_2^m.$$

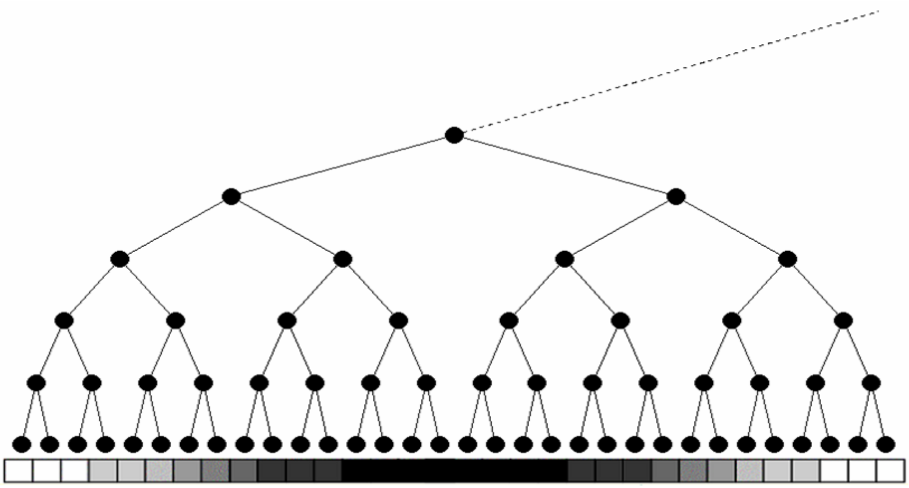
WFA - Polynomials

- WFA can represent polynomials in a very economical way.
- Any polynomial of degree m can be computed by some WFA with m+1 states ("Line-Automata")
- Polynomials are the only smooth functions that WFA can generate exactly for arbitrary resolution. (Culik, Karhumäki, Kari, Steinby, Droste)
- Even the square-root function can only be approximated well, (Karhumäki, Terlutte, et. al.)

WFA Inference Problem

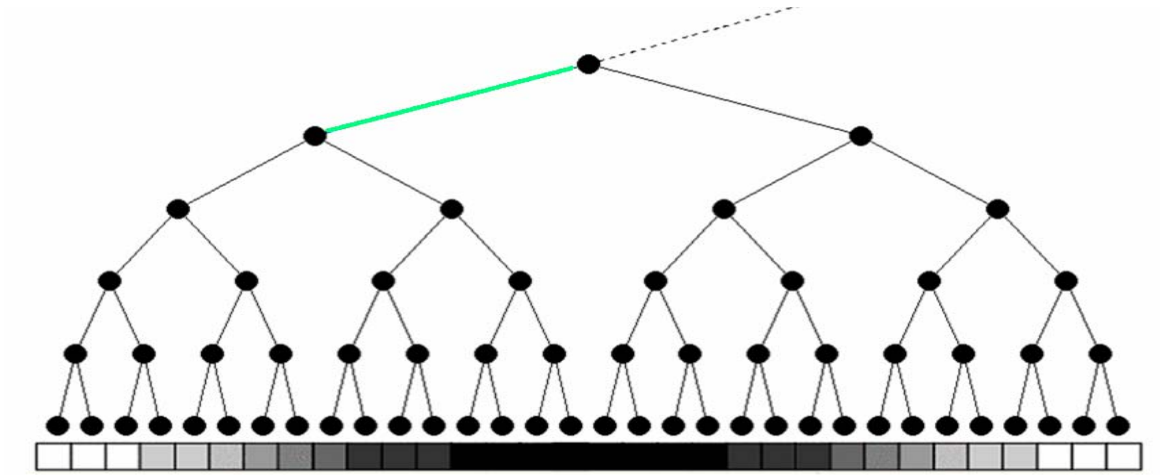
For a given function (a k-dimensional grid of real-valued intensities) find a WFA which **approximates this function well**, and which can be stored in a **small number of bytes**.

Efficient heuristic implemented by Culik and Kari.



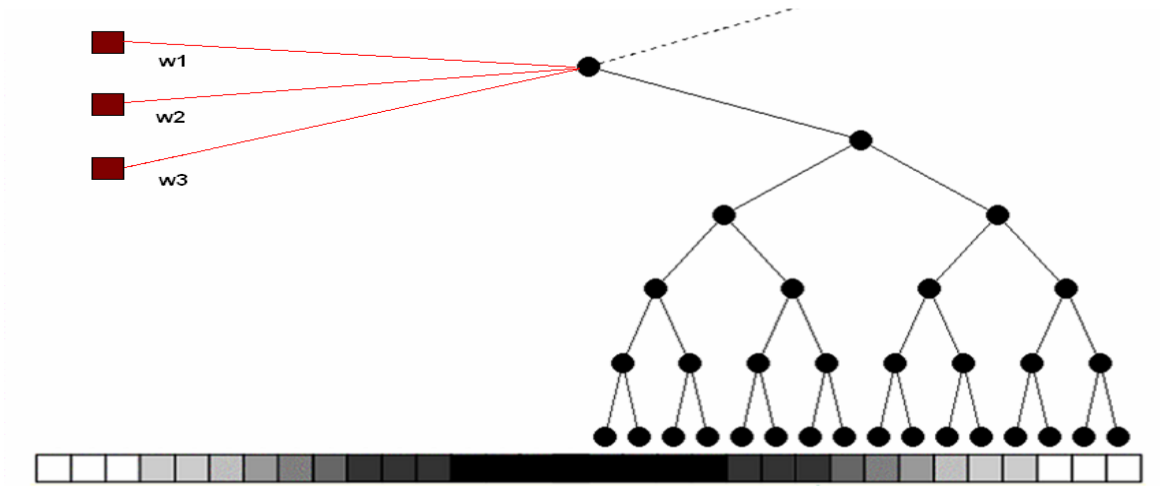
Coding Decisions in the Bintree

At each inner node recursively check, whether to
- approximate the current sub-image by a **linear combination** or
- **subdivide** it further



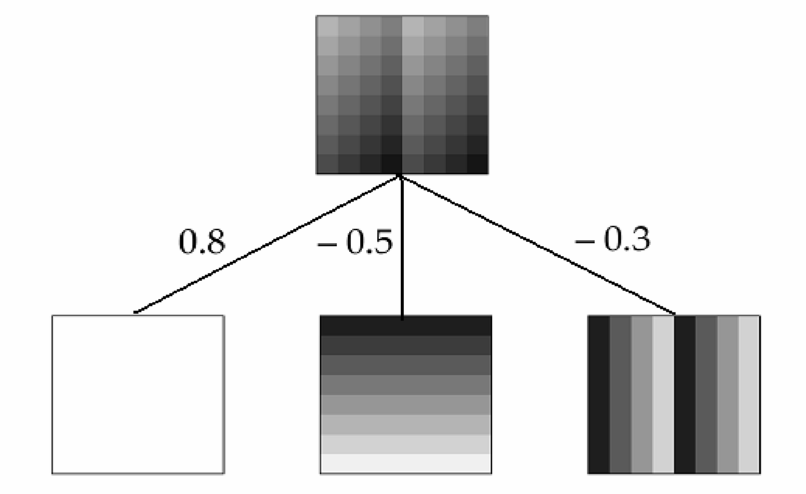
Coding Decisions in the Bintree

Choosing a linear combination:

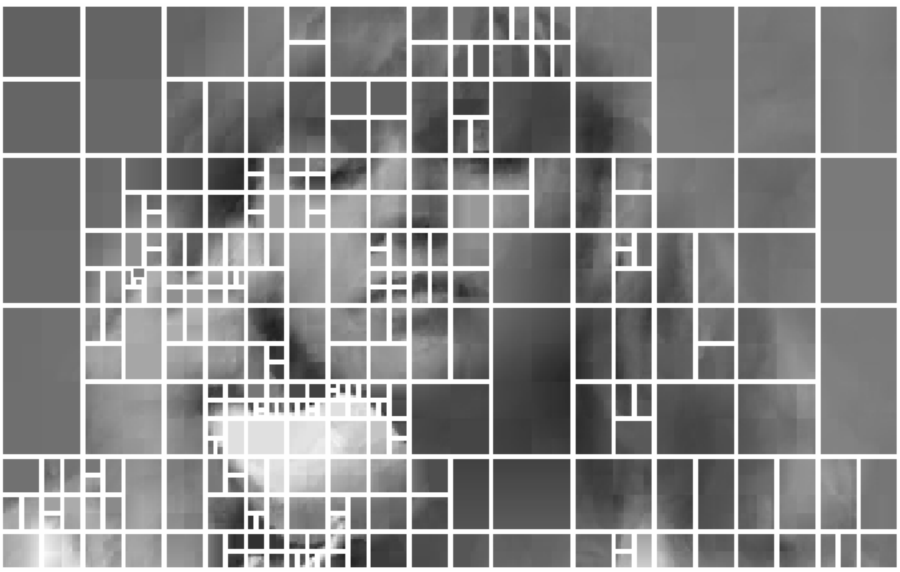


Cost-Function

A cost-function compares the locally generated **error** and the storage **space** needed for the current approximation and thus checks the build-up of linear combinations in each step



Expl. Bintree Decomposition



JPEG Baseline vs. WFA for Low-Bitrates

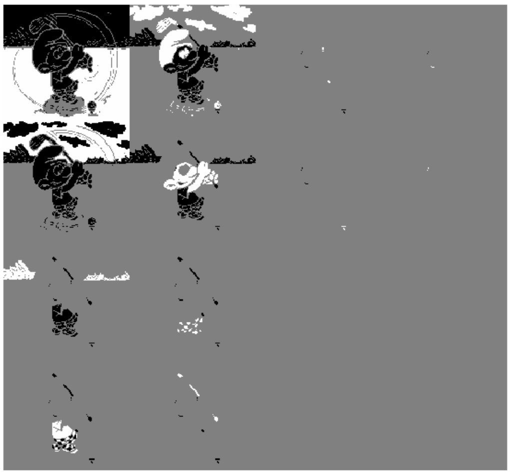
JPEG 2000 vs. WFA

Testimage „Cafe“, (detail) 390 x 280 x 8bpp:

original, JPEG 2000, WFA



WFA for Cartoon-like Images



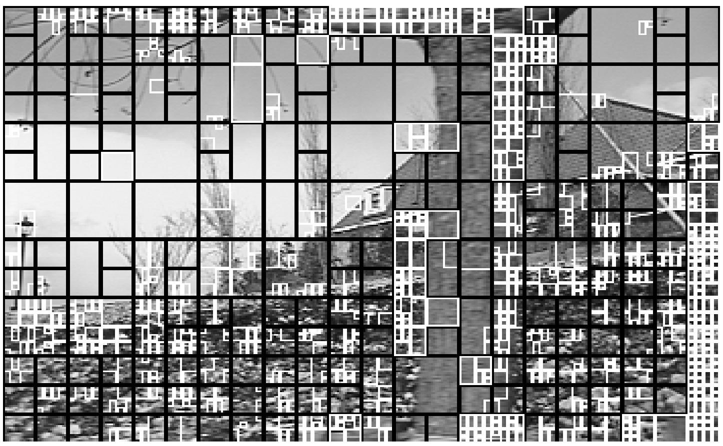
- Separate all colors into bit-planes.
- Reconstruct by painting **layer over layer**.
- „**Don‘t care**“ values (in gray) are those which will be repainted later.
- Replace addition and multiplication in linear combinations of WFAs by **Boolean operators**. In experiments **XOR** showed best performance.
- In the cost-function count the number of wrong pixels to control error-rate vs. file-size.

WFAs and Videosequences

Exploiting temporal redundancies similar to MPEG-x, ….
Different characteristics of frame-sequences:

Motion Estimation in Fast Actions, „Head and Shoulder“ Sequences

MWFA-Image-Partitioning and Motion-Vectors

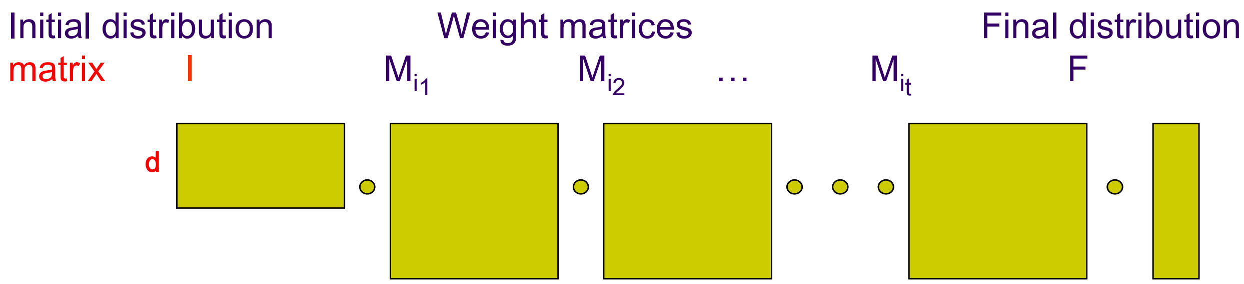


Motivations for Extending WFA

- In many practical applications there are finite higher dimensional grids for space, time, color-space, …
- To display e.g. a square-root function is not really harder than to draw a square-function on the screen
- There are other types of self-similarities not covered by WFA

Parametric WFA (PWFA)

PWFA generalize WFA by replacing the **initial distribution vector by a matrix**. Now a d-dimensional vector of real values is computed for arbitrary input sequences $w = i_1\, i_2 \ldots i_t$.

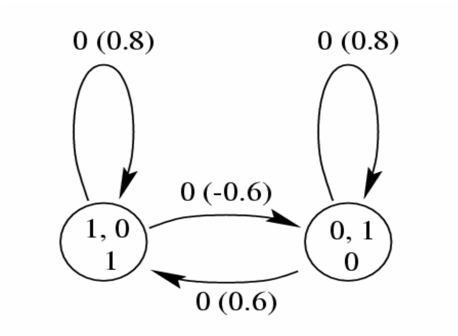


A point x belongs to the result-set S(A) defined by a PWFA A, if there are infinitely many w s.t. $x = f_A(w)$ or the are points $f_A(w)$ arbitrarily close to x. The resulting d-dimensional vectors x are interpreted as relations, e.g. for d=3: bi-level in 3D or grayscale in 2D

PWFA over Unary Alphabet

Rotations by an irrational angle e.g. $\cos^{-1}(0.8)$ applied to the final distribution point (1, 0) yield the points of the unit circle.

$S(A) = \{(\cos(t), \sin(t)) \mid t \in \mathbb{R}\}$

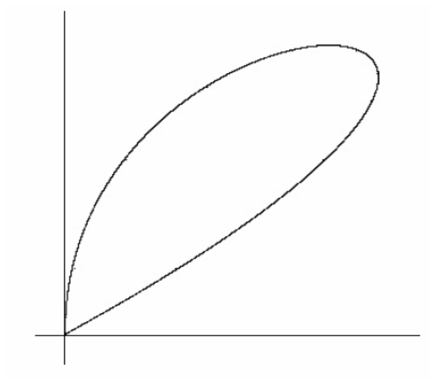


First 50 and 1000 points:

(a) (b)

Polynomial Curves

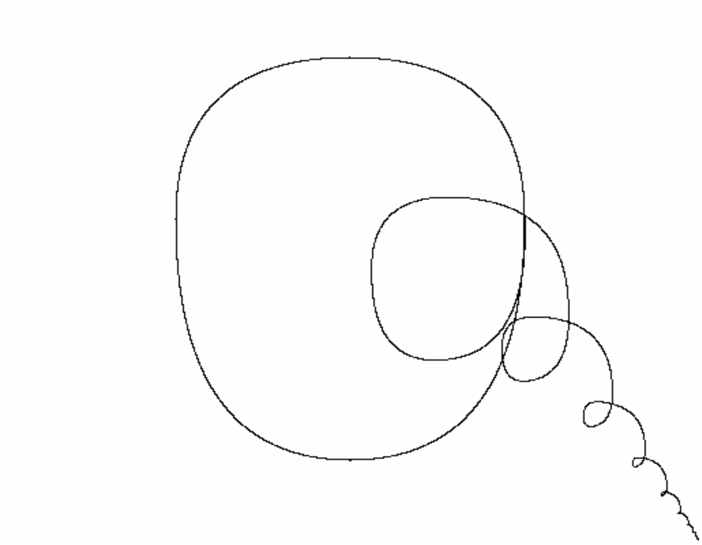
For $(t^2 - t^3, t - t^2)$, $0 \le t \le 1$, a PWFA with 4 states can compute the 2D-curve-segment:



Any d-dimensional curve with parametric representation by d polynomials of maximal degree m can be computed by a PWFA with m+1 states.

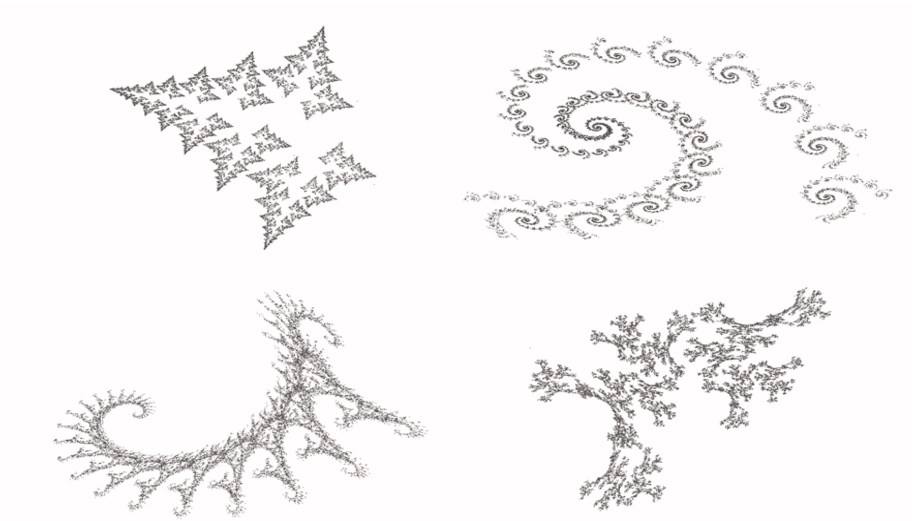
Segments and Splines

Expl.: Piecewise combination of parabola-chunks
(9 states PWFA):



Fractal Compression

Exploiting self-similarities in the pictures is comparable to dictionary-methods in text-compression in that „a new part of a picture is described by one or more references to previously coded parts".



Iterated Function Systems (IFS)

Affine Transformations in IR^m form the basis for IFS.

In IR^2 these are just:

$$w\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = A\bar{x} + B$$

Thus, affine transformations can contain scaling, rotation and translation.

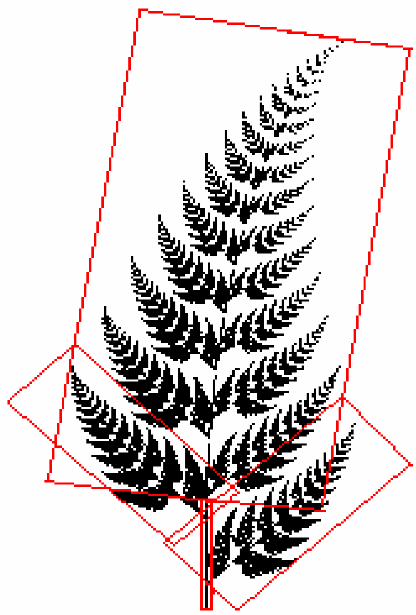
A transformation $f : X \rightarrow X$ is „contractive“, if there is an s such that

$0 \leq s < 1$ and $d(f(x), f(y)) \leq s\, d(x, y)$ for all x, y in X.

IFS are defined via sets of contractive affine transformations.

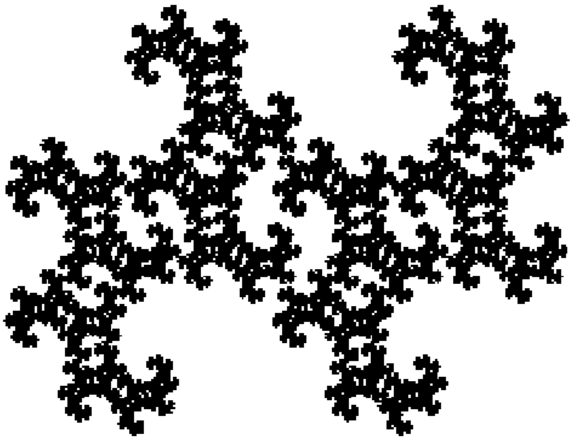
Affine Transformations for „Fern“

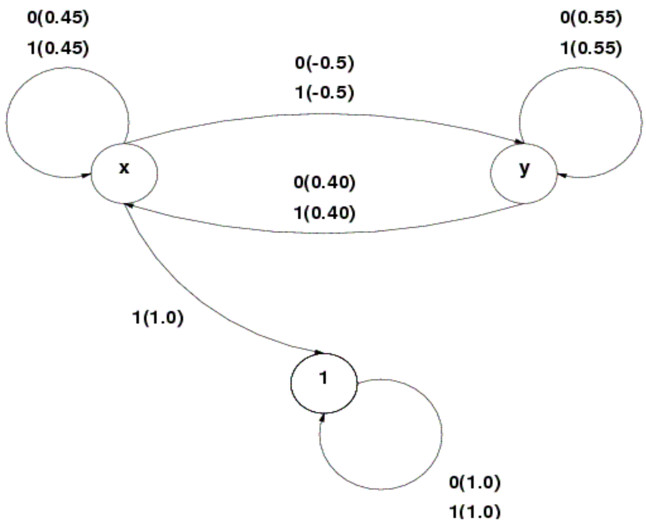
Four affine transformations for: upper part, new left and right leaf, new part of stem



PWFA-Simulation of Iterated Function Systems

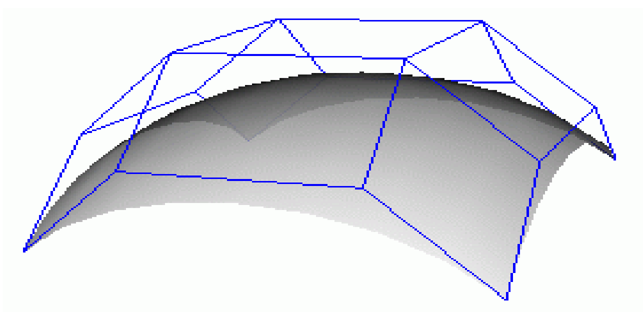
Any 2D IFS with k contractive affine maps can be simulated by a PWFA with 3 states and k labels.   Expl. „Dragon“:



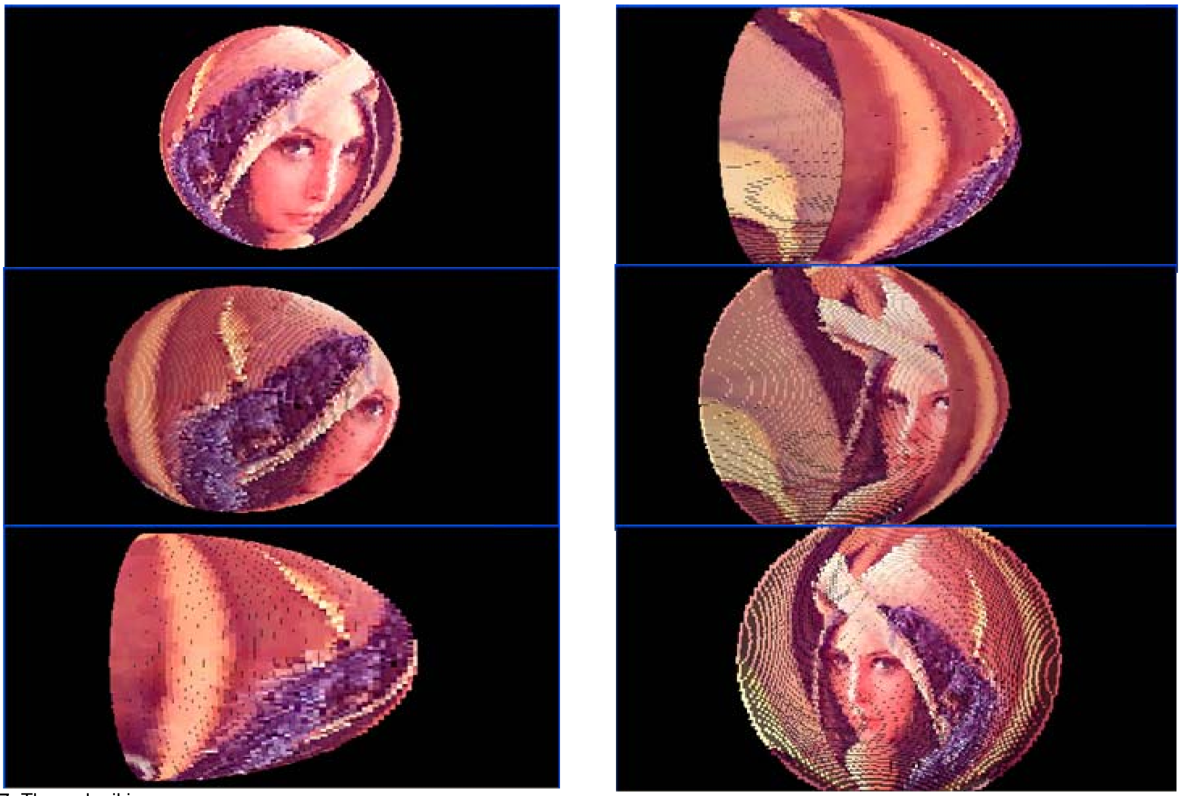


Spline Surfaces

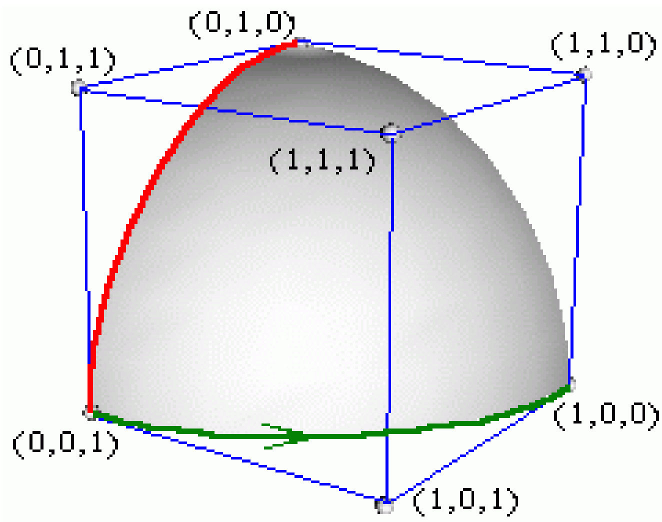
- E.g. the Bezier spline surfaces (or patches) are constructed of two Bezier curves
- The control points now make a control polyhedron of the surface.
- Moving the control points of one Bezier curve along a set of Bezier curves to define a surface



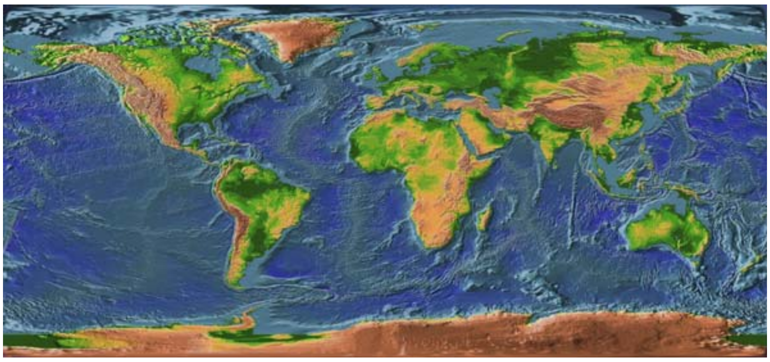
PWFA Spline Patches and Textures

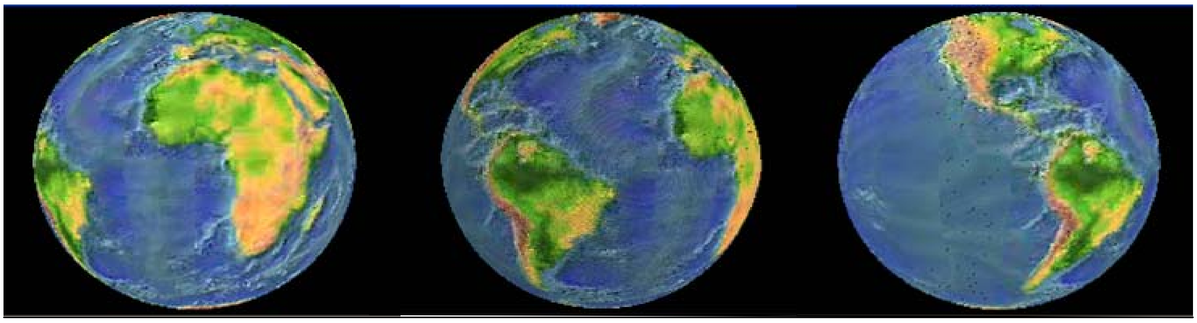


Spheres and Control Points



Spheres and Textures





Open Problems

- Pure WFA-encoding of video-clips and volume-data
- WFA: Encoding efficiency: „space for time“
- Inference heuristics for PWFA or for PWFA-subfamilies
- Applications of PWFA in „augmented reality“
- Efficiency for the representation of 3D-spline-patches
- …..

References (1)

- K. Culik II and J. Kari. Image compression using weighted finite automata. Computers and Graphics, 17(3):305--313, 1993.
- M. Barnsley, Fractals Everywhere, 2nd ed. Boston, Academic Press, 1993.
- K. Culik II, J. Karhumäki, Finite automata computing real functions, SIAM J. Comp. 23/4, 789-814, 1994.
- U. Hafner, J. Albert, S. Frank, and M. Unger. Weighted finite automata for video compression. IEEE J. Selected Areas in Comm. 16(1):108-119, 1998.
- J. Albert, J. Kari, Parametric Weighted Finite Automata and Iterated Function Systems, Proc. Fractals in Engineering, 248-255, 1999
- D. Derencourt, J. Karhumäki, et al. On computational power of weighted finite automata. Fund. Inform., 25(3): 285–293, 1996.

References (2)

- K. Culik II, V. Valenta, Finite automata based compression of bi-level and simple color images, Comput. Graphics 21 (1997) 61-68.
- IEEE Transact. Vis. Comp. Graphics, p. 1559, vol. 12 (6) 2006
- G. Tischler, Real Functions computed by Parametric Weighted Finite Automata, Proc. CAI 2005, Thessalonki
- http://en.wikipedia.org/wiki/Spline_(mathematics)
- http://www.mizuno.org/gl/bs/
- G. Tischler, Properties and Applications of Parametric Weighted Finite Automata, JALC 10/2-3, 347-365, 2005
- G. Tischler, Parametric Weighted Finite Automata for Figure Drawing, CIAA 2004, LNCS 3317, 259-268, 2005
- G. Tischler, J. Albert, J. Kari, Parametric Weighted Finite Automata and Multidimensional Dyadic Wavelets, Proc. Fractals in Engineering 2005, Tours, France