Fractals

Part 4 : IFS

Martin Samuelčík
Department of Applied Informatics

Transformations in HS

- Contractions $\{f_1, f_2, \ldots, f_N\}$ with factors $\{s_1, s_2, \ldots, s_N\}$.
- W: H(X) -> H(X) $W(B) = \bigcup_{n=1}^{N} f_n(B) \quad \forall B \in H(X)$
- W is contraction
- W has fixed point in H(X)
- Fixed point = non-empty compact set
- Attractor, invariant

IFS

- Complete metric space (X,d)
- Finite set of contractions $f_1,\ldots,f_N$ with contraction factors $s_1,\ldots,s_N$
- Notation $(X,f_1,\ldots,f_N)$
- Hutchinson operator W
- Contraction factor $s = \max\{s_1,\ldots,s_N\}$

$$W(B) = \bigcup_{n=1}^{N} f_n(B) \quad \forall B \in H(X)$$

IFS 2

- Iterated function system
- Deterministic fractal
- Simple description of attractor
- Multiple reduction copy machine
- Attractor is limit of forward iterations
- Independent on initial set

IFS – computing attractors

- Deterministic & stochastic algorithms
- Deterministic:
- B in H(X)
- For n = 1 to infinity
  $W^n(B) = W(W^{n-1}(B))$
  Delete W^{n-1}, paint W^n
- Can be computed adaptively

Dimension of IFS attractor

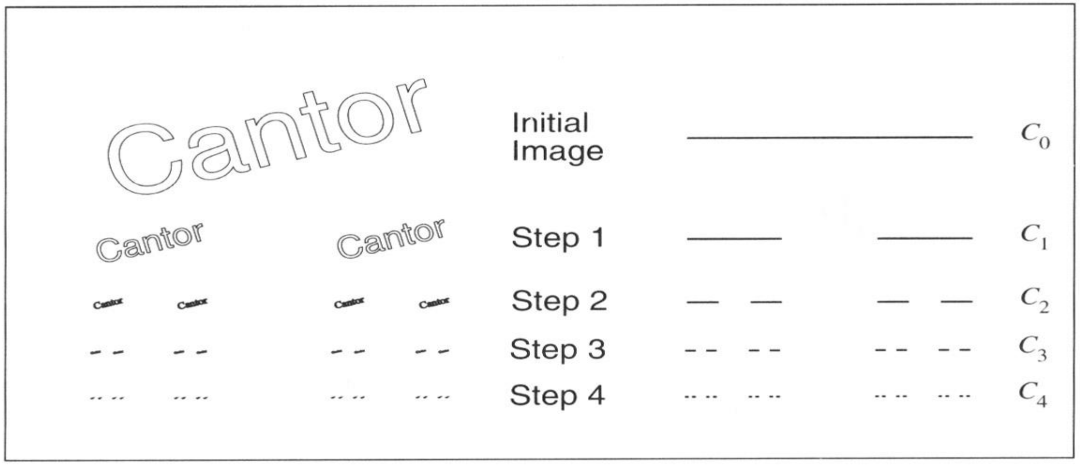
- In simple case
- $f_i(A)$ and $f_j(A)$ are disjunctive for attractor A and i <> j, this means no overlapping
- f_i are contractions with same factor c
- $D = \log(N) / \log(1/c)$
- s_1^D + ... + s_N^D = 1

Affine transformations

- $y = Ax + b$
- A consists of rotation, scaling
- b is translation
- We needs 3 points to determine it
- Fixed point
- Contraction (Euclidean metric):
  $a^2 + c^2 < 1,\ b^2 + d^2 < 1,$
  $a^2 + b^2 + c^2 + d^2 < 1 + (ad - cb)^2$

Classical fractals as IFS

- Cantor set
- f1 = [(1/3)*x,(1/3)*y]
- f2 = [(1/3)*x + 2/3,(1/3)*y]
- Attractor = Cantor set



Koch curve as IFS

$$f_1(\mathbf{x}) = \begin{bmatrix} 0.333 & 0 \\ 0 & 0.333 \end{bmatrix}\mathbf{x}$$

Scale by r

$$f_2(\mathbf{x}) = \begin{bmatrix} 0.167 & -0.289 \\ 0.289 & 0.167 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0.333 \\ 0 \end{bmatrix}$$

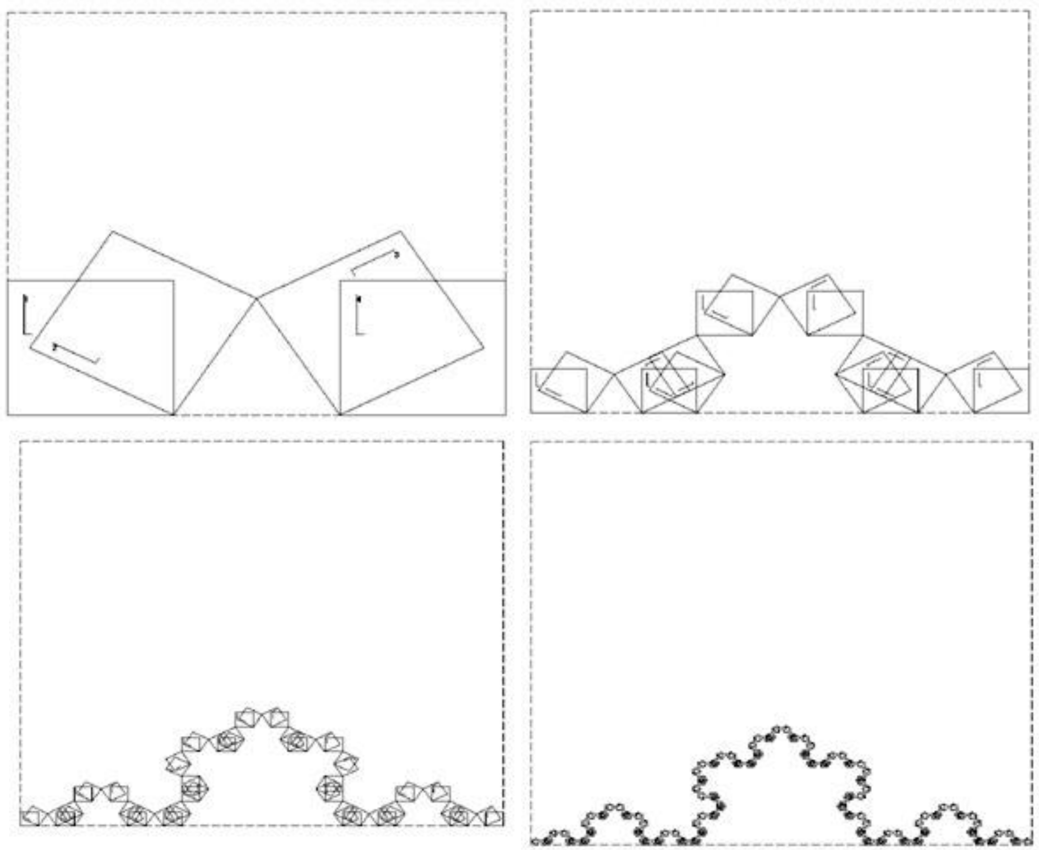
Scale by r, rotation by $60°$

$$f_3(\mathbf{x}) = \begin{bmatrix} 0.167 & 0.289 \\ -0.289 & 0.167 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0.500 \\ 0.289 \end{bmatrix}$$

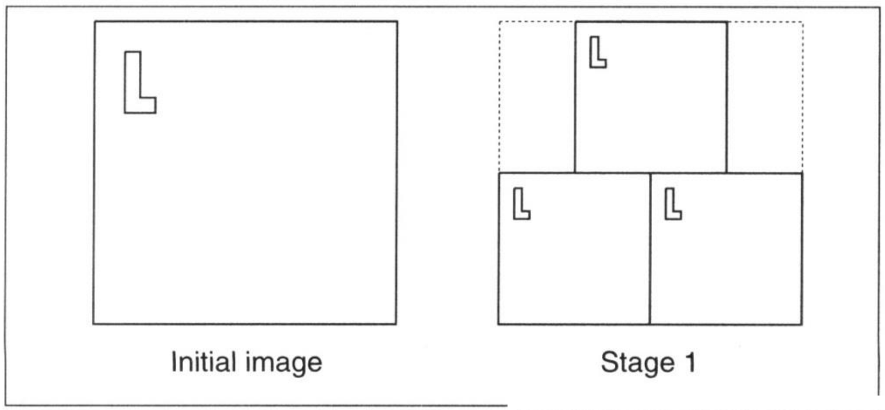
Scale by r, rotation by $-60°$

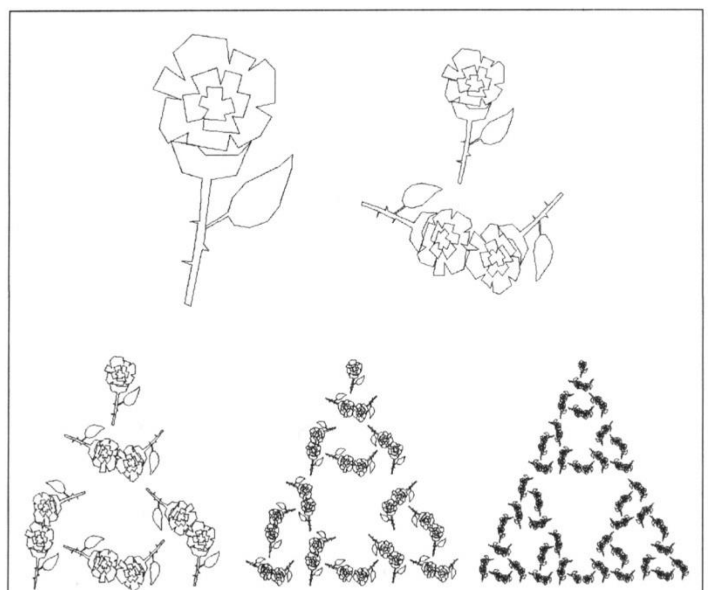
$$f_4(\mathbf{x}) = \begin{bmatrix} 0.333 & 0 \\ 0 & 0.333 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0.667 \\ 0 \end{bmatrix}$$

Scale by r



Sierpinski gasket as IFS





$$f_1(\mathbf{x}) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}$$

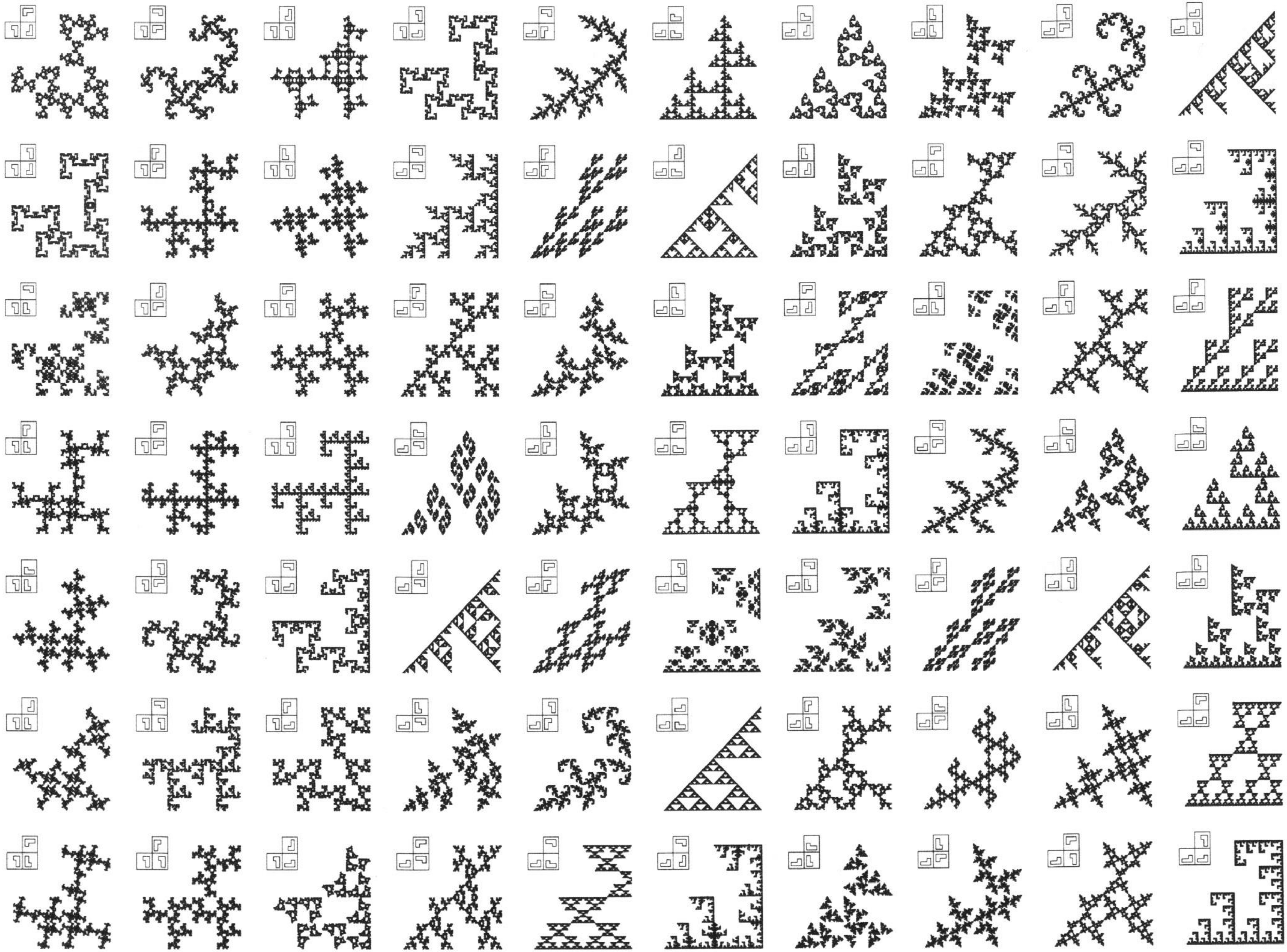
scale by r

$$f_2(\mathbf{x}) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

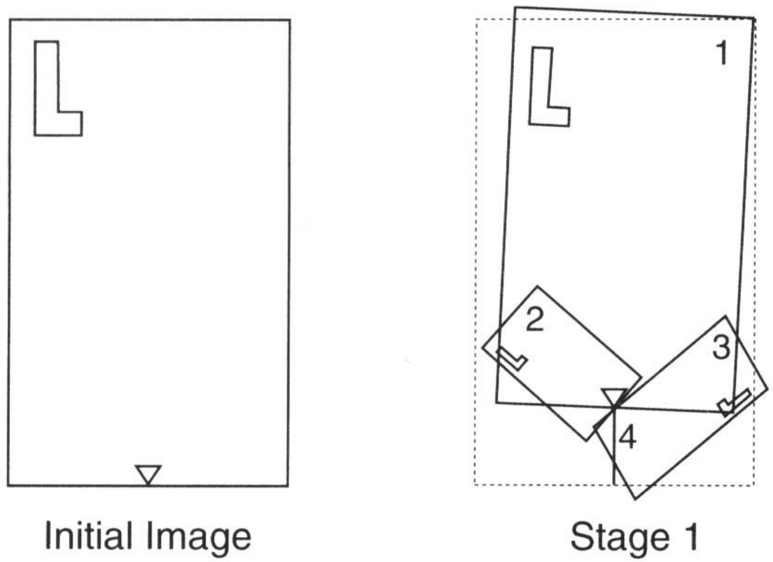
scale by r

$$f_3(\mathbf{x}) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0.250 \\ 0.433 \end{bmatrix}$$

scale by r



Barnsley's fern



$$f_1(x,y) = \begin{pmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

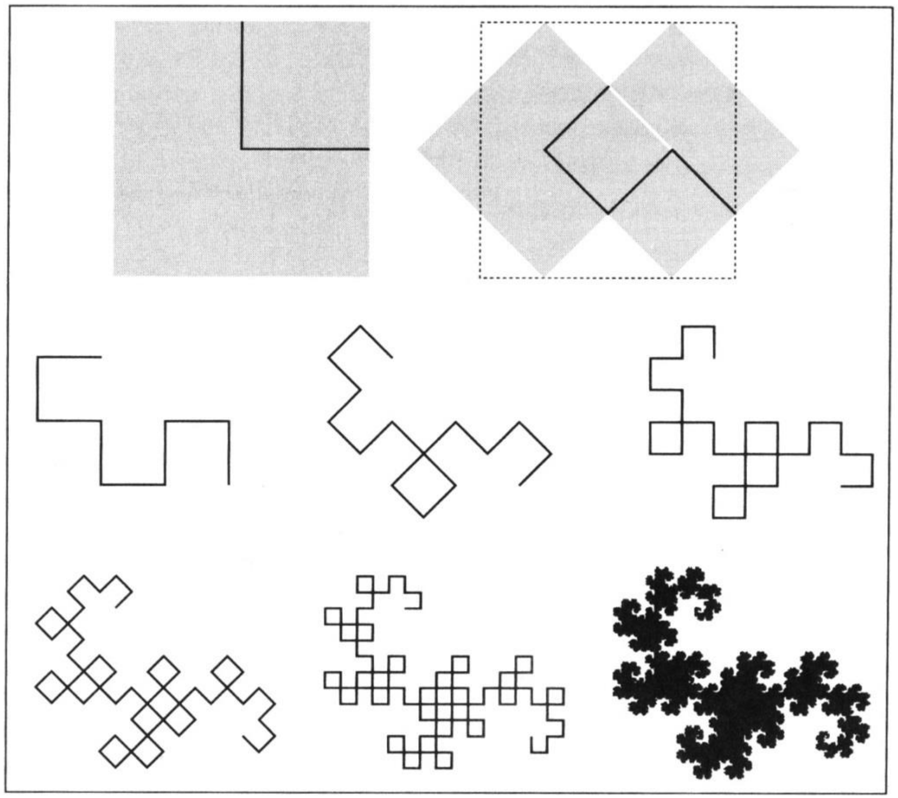
$$f_2(x,y) = \begin{pmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0.44 \end{pmatrix}$$

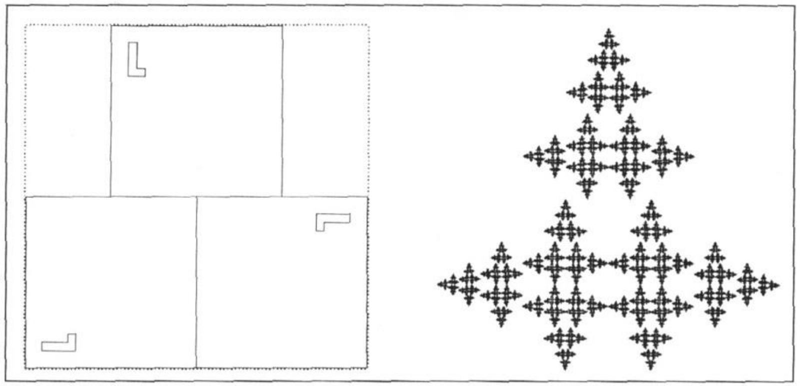
$$f_3(x,y) = \begin{pmatrix} 0.2 & -0.26 \\ 0.23 & 0.22 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$$

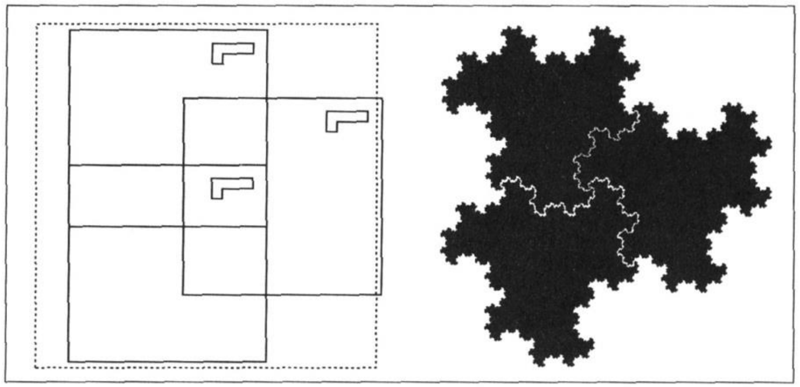
$$f_4(x,y) = \begin{pmatrix} 0 & 0 \\ 0 & 0.16 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Barnsley's fern 2

Other IFS examples



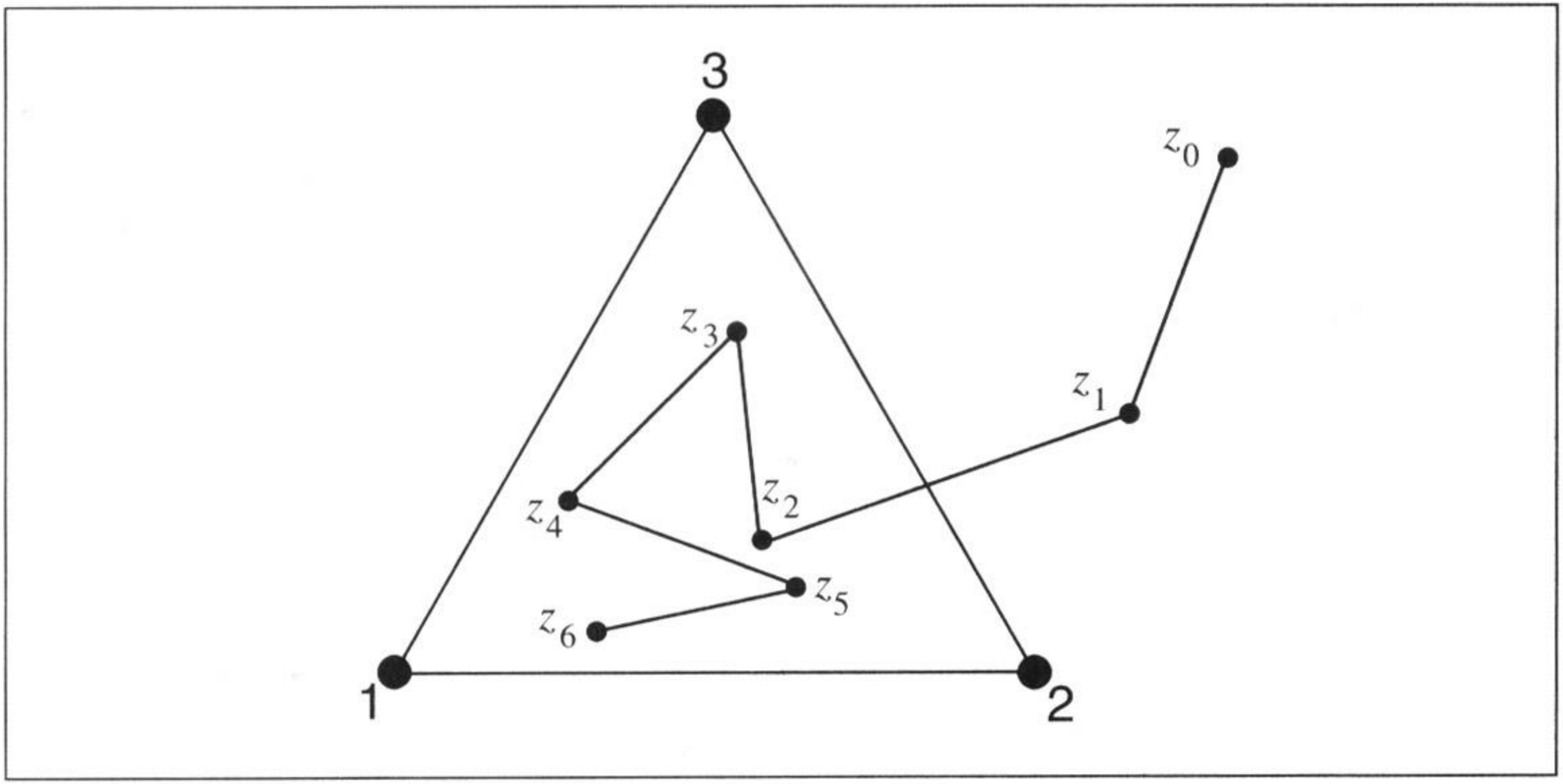




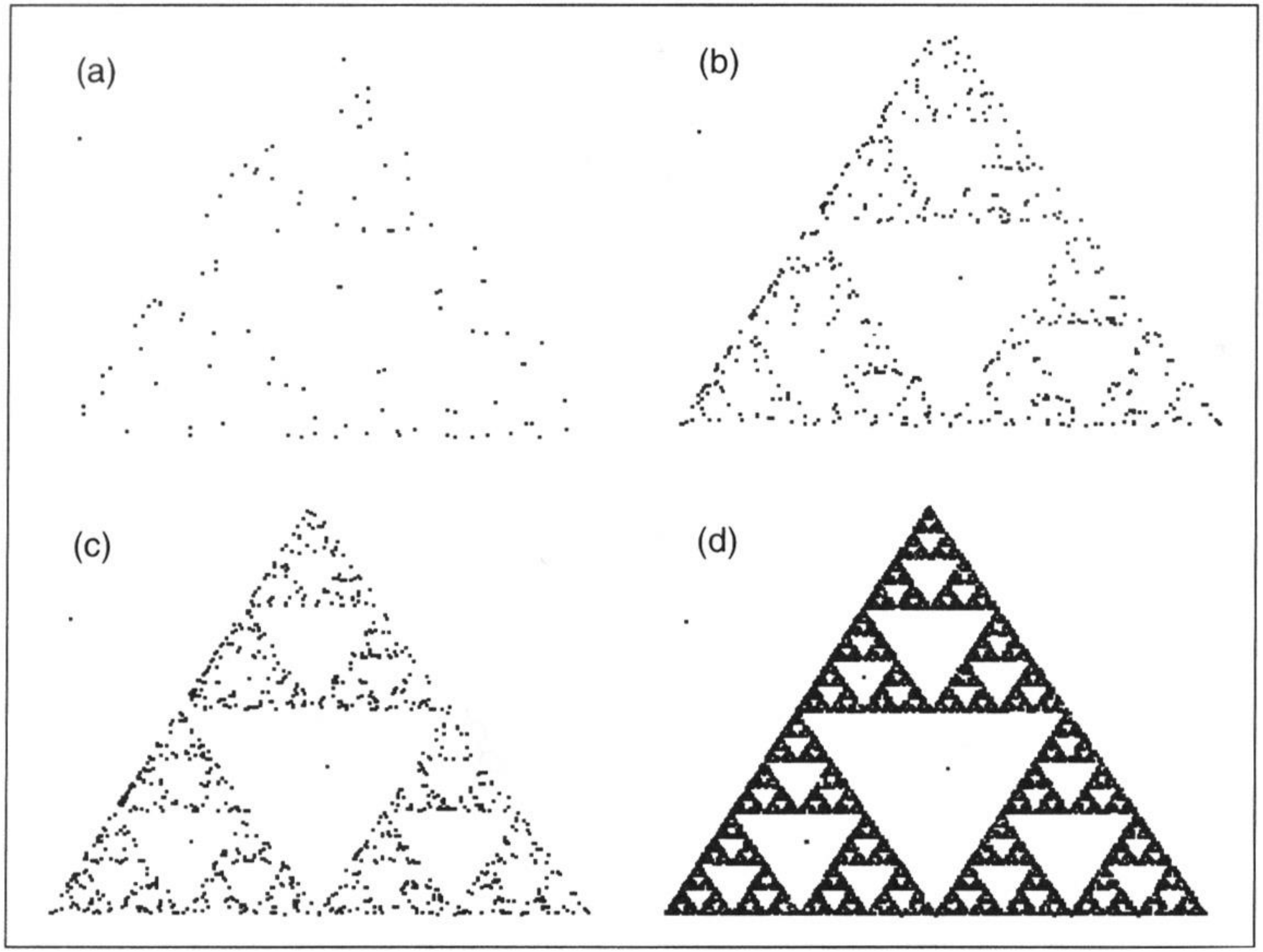
Chaos game with triangle

- Given 3 points A, B, C
- Points probabilities p1,p2,p3
- Starting with point z_0 from plane
- In step i pick randomly point from A,B,C with given probability
- Make center from picked point and z_i
- This center is z_{i+1}

Chaos game with triangle



And the Result is …



Chaos game generally

- Given IFS
- Given probabilities for each contraction in IFS - $p_1,\ldots,p_N$;$p_1+\ldots+p_N = 1$
- Pick starting point z_0
- In i-th step pick one contraction f_j
- $z_{i+1} = f_j(z_i)$

Chaos game

- Stochastic algorithm for computing attractor
- Using randomness
- Controlled with probability
- Attractor can appear faster
- Working just with points

Why is it working?

- We can have $A_0 = \{z_0\}$
- In IFS: $A_1 = \{f_1(z_0),\ldots,f_N(z_0)\}$
- In k-th step: N^k points
- Chaos game produces only one from these points; z_k in A_k
- So z_k is still close to attractor

Addressing system

- After k iterations we have some point
- Address: $s_1 s_2 \ldots s_k$; s_i in {1,2,...,N}
- In j-th step we picked s_j-th contraction
- Point in attractor = infinite address
- In IFS we have in k-th step all available addresses, in chaos game we have one

Generating attractor

- Given point P from attractor
- $P = s_1 s_2 \ldots$
- For some number m, the point $s_1 \ldots s_m$ is ε close to P
- If some point S contains sequence $s_1 \ldots s_m$, then it's ε close to P
- Such sequence will exists

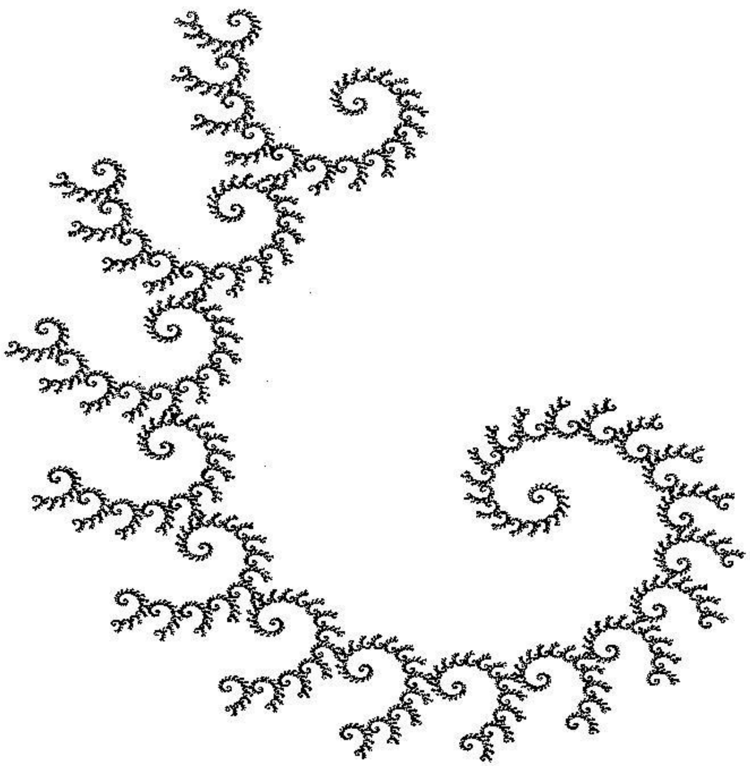
Picking good probabilities

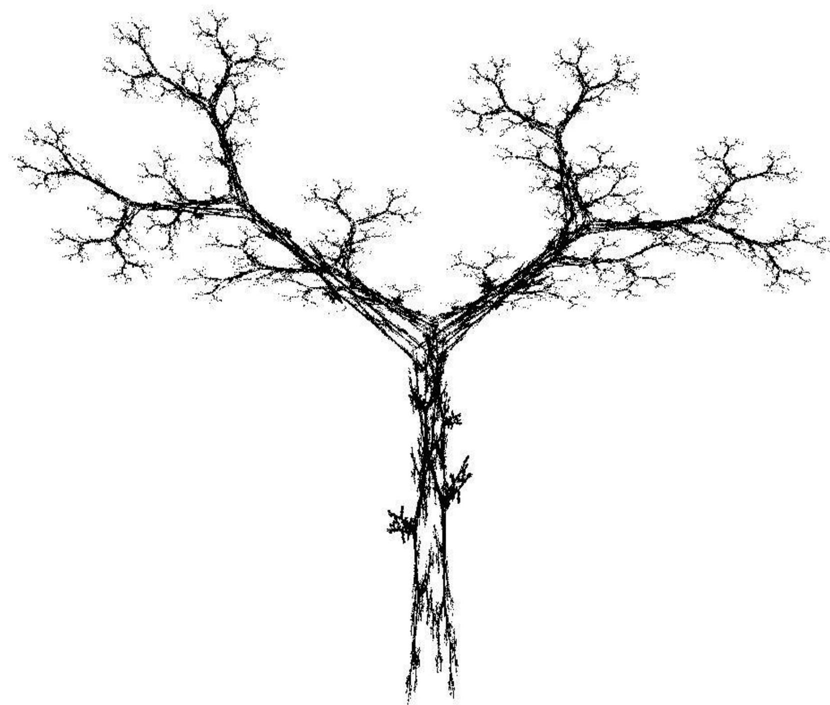
- Can be given
- Each probability is 1/N
- Heuristic methods:

$$p_i = \frac{|\det A_i|}{\sum_{i=1}^{N} |\det A_i|}$$

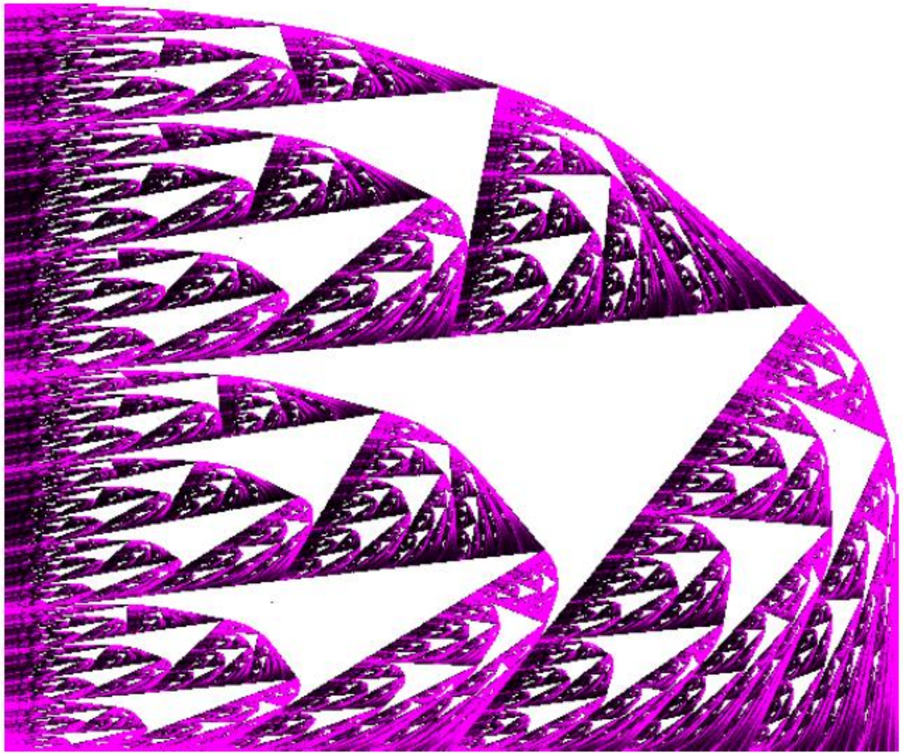
- Adaptive methods

Chaos game & IFS



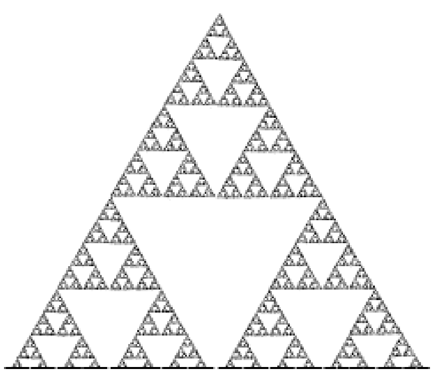
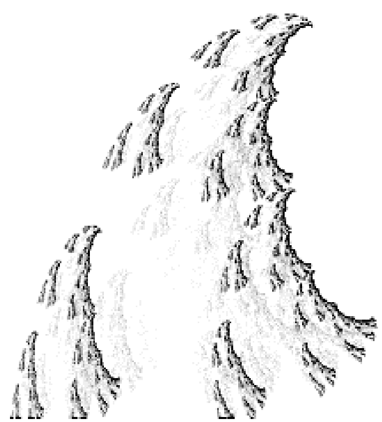


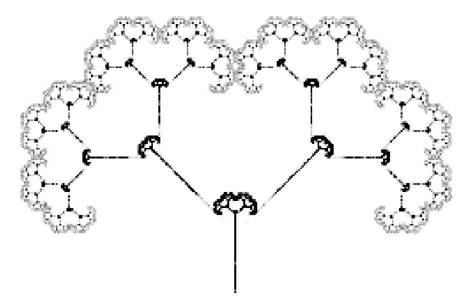
Other transformations



Morfing of IFS

```
// fern      { 0.850000, 0.040000,-0.040000, 0.850000, 0.000000, 1.600000, 0.850000},
             { 0.200000,-0.260000, 0.230000, 0.220000, 0.000000, 1.600000, 0.070000},
             {-0.150000, 0.280000, 0.260000, 0.240000, 0.000000, 0.440000, 0.070000},
             { 0.000000, 0.000000, 0.000000, 0.160000, 0.000000, 0.000000, 0.010000},
             { 1.000000, 0.000000, 0.000000, 1.000000, 0.000000, 0.000000, 1.000000},
// triangle  { 0.500000, 0.000000, 0.000000, 0.500000,-0.500000, 0.000000, 0.333333},
             { 0.500000, 0.000000, 0.000000, 0.500000, 0.500000, 0.000000, 0.333333},
             { 0.500000, 0.000000, 0.000000, 0.500000, 0.000000, 0.860000, 0.333334},
             { 1.000000, 0.000000, 0.000000, 1.000000, 0.000000, 0.000000, 1.000000},
             { 1.000000, 0.000000, 0.000000, 1.000000, 0.000000, 0.000000, 1.000000},
```



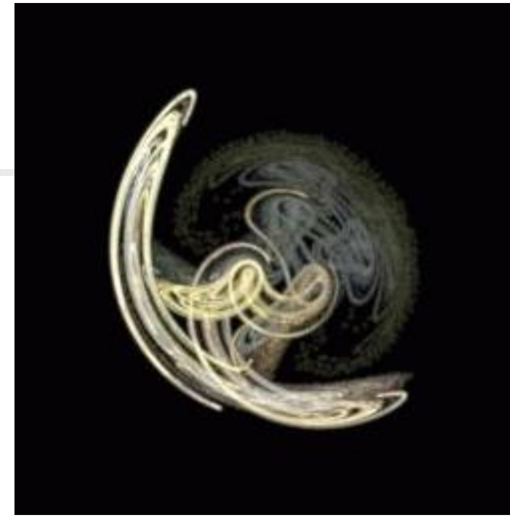
http://www.root.cz/clanky/plynuly-morfing-mezi-dvojici-ifs-systemu/

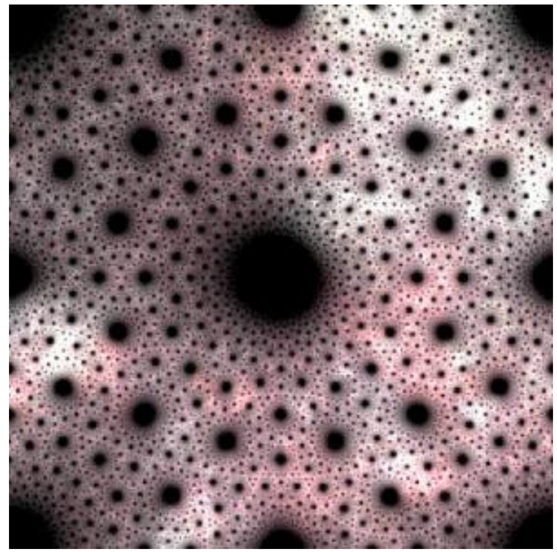
Fractal Flame

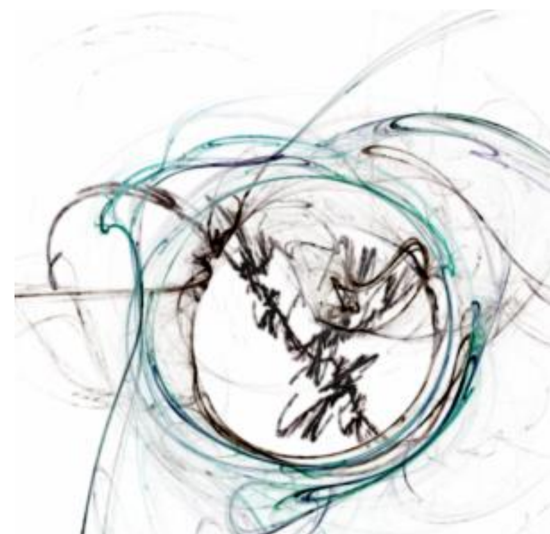
- Added variation function
- RWA – Random Walk Algorithm
- Each visited pixel – increased color

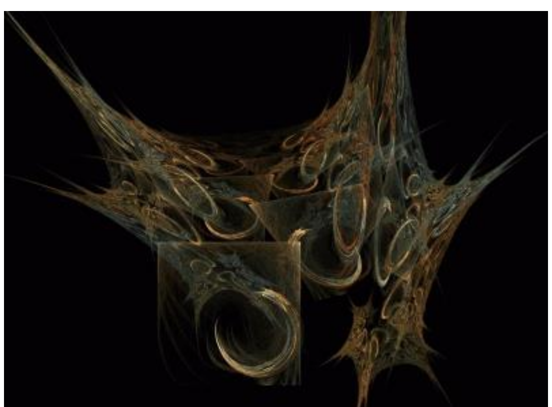
$V_0(x,y)$	(x,y)	linear
$V_1(x,y)$	$(\sin x, \sin y)$	sinusoidal
$V_2(x,y)$	$(x/r^2, y/r^2)$	spherical
$V_3(x,y)$	$(r\times\cos(\varphi+r), r\times\sin(\varphi+r))$	swirl
$V_4(x,y)$	$(r\times\cos(2\varphi), r\times\sin(2\varphi))$	horseshoe
$V_5(x,y)$	$(\varphi/\pi, r-1)$	polar
$V_6(x,y)$	$(r\times\sin(\varphi+r), r\times\cos(\varphi-r))$	handkerchief
$V_7(x,y)$	$(r\times\sin(\varphi r), -r\times\cos(\varphi r))$	heart
$V_8(x,y)$	$(\varphi\times\sin(\pi r)/\pi, \varphi\cos(\pi r)/\pi)$	disc
$V_9(x,y)$	$((\cos\varphi+\sin r)/r, (\sin\varphi-\cos r)/r)$	spiral
$V_{10}(x,y)$	$((\sin\varphi)/r, (\cos\varphi)r)$	hyperbolic
$V_{11}(x,y)$	$((\sin\varphi)(\cos r), (\cos\varphi)(\sin r))$	diamond
$V_{12}(x,y)$	$(r\times\sin^3(\varphi+r), r\times\cos^3(\varphi-r))$	ex
$V_{13}(x,y)$	$(r^{1/2}\times\cos(\varphi/2+\Omega), r^{1/2}\times\sin(\varphi/2+\Omega))$	julia
$V_{14}(x,y)$		
$V_{15}(x,y)$		
$V_{16}(x,y)$	$(2r/(r+1)x, 2r/(r+1)y)$	fisheye
$V_{17}(x,y)$		
$V_{18}(x,y)$		
$V_{19}(x,y)$		
$V_{20}(x,y)$	$(\cos(\pi x)\times\cosh(y), -\sin(\pi x)\times\sinh(y))$	cosine

Fractal Flame 2









End

End of Part 4