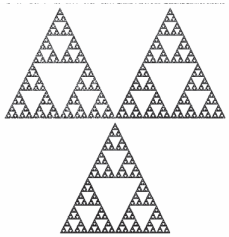


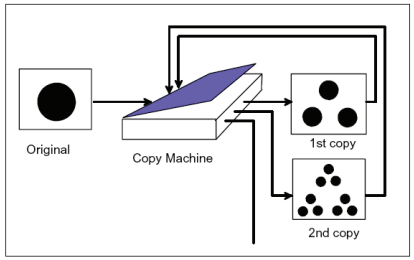
Fraktálová komprese obrazu

Úvod

- Termín fraktál poprvé použil Benoit Maldebrot (1975)
- Některé definice pojmu fraktál:
  - Fraktál je nerovný nebo fragmentovaný geometrický tvar, který může být rozdělen na části, které jsou (alespoň přibližně) menší kopií celku. Fraktály jsou obecně „sobě-podobné" (self-similar) to je malá část vypadá jako celý obraz
  - Fraktál je obraz nebo snímek, který může být kompletně popsán matematickým algoritmem (v libovolném rozlišení)
  - Fraktál je pevný bod (attractor) systému iterovaných funkcí (Iterated Function System)

Systém iterovaných funkcí

- soubor kontraktivních zobrazení
- nejlépe lze IFS vysvětlit pomocí kopírovacího stroje (Multiple Reduction Copying machine) s následujícími vlastnostmi:
  1. Kopírka obsahuje skupin čoček, nastavených tak, že mohou vytvářet překrývající se kopie originálu
  2. Každá čočka zmenšuje velikost originálu
  3. Kopírka pracuje iteračně ve zpětnovazebním režimu tj.,výstup je znovu přiveden na vstup



- Matematicky lze každou čočku popsat jako tzv. kontraktivní afinní zobrazení, které mění měřítko vstupu (zmenšuje), natáčí ho a kopíruje na výstup.

$$w_i = \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

tj. každý bod vstupního obrazu (x,y) bude transformován do výstupního obrazuumístěn v nové pozici x', y', pro které platí

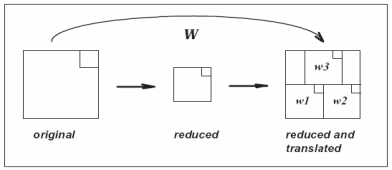
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

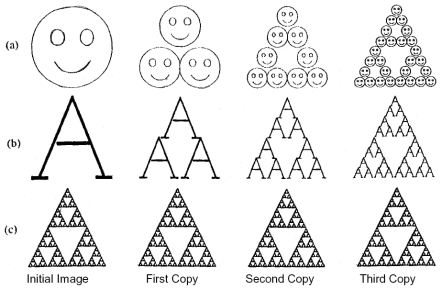
- Krontraktivní zobrazení : $x_1'=w(x_1)$, $x_2'=w(x_2)$

$$d(x_1,x_2)=s\cdot d(x_1',x_2') \qquad 0 < s < 1$$

Příklad IFS systému: (Sierpinského trojúhelník)

$$w_1 = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \quad w_2 = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix} \quad w_3 = \begin{bmatrix} 0.5 & 0 & 0.25 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$





Zobecnění IFS systému pro šedotónové obrazy

- Pro šedotónové obrazy je IFS systém trojrozměrný a zobrazení má tvar

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = w_i \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

kde s_i a o_i slouží k modifikaci jasu

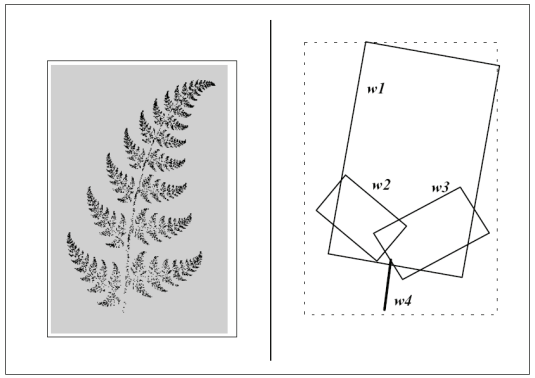
Další příklad IFS fraktálu: (Barnsleyova kapradina)

$$w_1 = \begin{bmatrix} 0.85 & 0.04 & 0.00 \\ -.04 & 0.85 & 1.6 \end{bmatrix}$$

$$w_2 = \begin{bmatrix} 0.20 & -.26 & 0.00 \\ 0.23 & 0.22 & 1.6 \end{bmatrix}$$

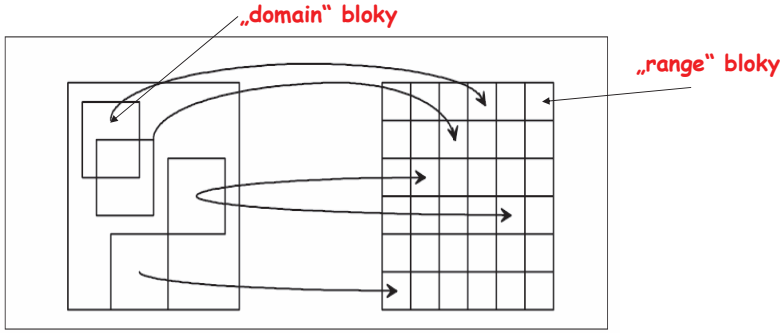
$$w_3 = \begin{bmatrix} -.15 & 0.28 & 0.00 \\ 0.26 & 0.52 & 0.44 \end{bmatrix}$$

$$w_4 = \begin{bmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.16 & 0.00 \end{bmatrix}$$



Základní princip algoritmu fraktálové komprese

Základní princip spočívá v rozdělení komprimovaného obrazu na „range" bloky (nepřekrývají se) a vyhledávání „domain" bloků (mohou se překrývat) které jsou „range" blokům podobné



„domain" bloky se mohou vyskytovat buď v základním tvaru nebo v transformované podobě. Používají se následující transformace

1. Rotace o 0°
2. Rotace o 90°
3. Rotace o 180°
4. Rotace o 270°
5. Překlopení přes horizontální osu
6. Překlopení přes vertikální osu
7. Překlopení přes hlavní diagonálu
8. Překlopení přes vedlejší diagonálu

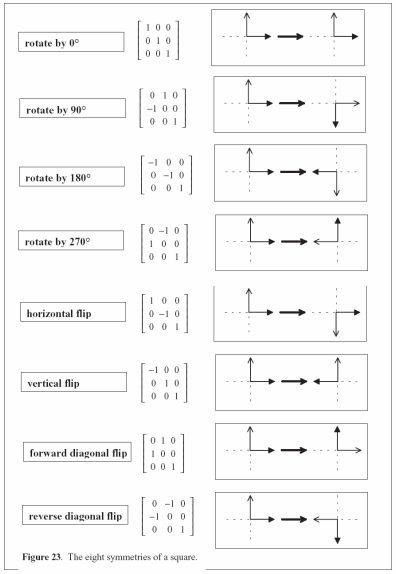


Figure 23. The eight symmetries of a square.

a) Pro každý doménový blok D_j a transformaci m_t (t=1,2,…,8) se vypočte $D_j^t=m_t(D_j)$ a na základě následujících rovnic se stanoví koeficienty ***s*** a ***o***

$$s=\frac{n\cdot(\sum_{i=1}^{n}d_i\cdot r_i)-(\sum_{i=1}^{n}r_i)\cdot(\sum_{i=1}^{n}d_i)}{n\cdot\sum_{i=1}^{n}d_i^2-(\sum_{i=1}^{n}d_i)^2}$$

$$o=\frac{1}{n}\left(\sum_{i=1}^{n}r_i-s\sum_{i=1}^{n}d_i\right)$$

b) Koeficienty ***s*** a ***o*** se kvantizují

c) Pro kvantizované koeficienty se podle následující rovnice vypočte chyba podobnosti bloků $E(D_j^t,R_i)$

$$E(D,R)^2=\frac{1}{n}\cdot\left[\sum_{i=1}^{n}r_i^2+s\cdot\left(s\cdot\sum_{i=1}^{n}d_i^2-2\cdot\sum_{i=1}^{n}d_i\cdot r_i+2\cdot o\cdot\sum_{i=1}^{n}d_i\right)+o\cdot\left(o\cdot n-2\cdot\sum_{i=1}^{n}d_i\right)\right]$$

Detailní algoritmus fraktálové komprese

1. **Segmentace obrazu** – komprimovaný obraz je rozdělen do bloků velikosti 8x8 (4x4) pixelů. Tyto bloky pokrývají celý obraz a nepřekrývají se. Tyto bloky se nazývají „range" bloky R_i

2. **Vytvoření souboru doménových bloků (domain pool)** – procházíme obraz zleva do prava shora dolů s krokem *k* pixelů a vytvoříme seznam tzv. doménových bloků, které mají dvojnásobnou velikost „range" bloků. V každém doménovém bloku jsou průměrovány sousední pixely a jsou uloženy do nového doménového bloku stejné velikosti jako „range" blok. Novým doménovým blokem přepíšeme blok původní.

For i=1 to N_R opakuj kroky 3 a 4 (N_R je počet „range" bloků)

3. **Vyhledávání** - pro každý „range" blok R_i nalezneme v souboru doménových bloků blok D^B, který se mu nejvíce podobá.

d) Nalezneme blok D_j^t s minimální chybou $E(D_j^t,R_i)$ t=1,2,…,8

e) V souboru doménových bloků nalezneme nejpodobnější blok tj.

$$D^B=\min_{t=1,2,\ldots,N_D}(D'_j)$$

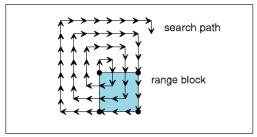
N_D je počet doménových bloků

4. **Výstupem je kód $w_i=(e_i,f_i,m_i,o_i,s_i)$**

5. **Výstupní posloupnost transformací je možné kódovat metodou bezztrátové komprese**

Strategie vyhledávání (vytvoření souboru doménových bloků)

1. **Metoda „hrubé" síly (heavy brute force)** – velikost kroku k=1. Časová složitost je $O(n^2)$, pro šedotónový obrazek vyžaduje algoritmus 2^{37}=128 Gflops
2. **„Light" Brute Force** – velikost kroku k>1. Kvalita výsledného obrazu může být v tomto případě horší protože přeskakujeme některé části obrazů, které by mohli být podobné s range blokem
3. **Omezená oblast vyhledávání** - oblast vyhledávání doménových bloků se redukuje pouze na okolí (např. kvadrant) testovaného „range" bloku.
4. **Lokální spirálové vyhledávání** – doménové bloky jsou vyhledávány na spirále začínající v pozici „range" bloku . Vyhledávání končí jakmile se nalezne vhodný doménový blok



5. **Hledání ve stejném místě jako je odpovídající range blok.** Jedná se o rychlé vyhledávání (složitost O(n)) s nízkou kvalitou
6. **Kategorizované vyhledávání** – každý doménový blok je zařazen do jedné ze 72 kategorií (3 třídy, 24 kategorií v každé třídě. Postup klasifikace bloků je následující: nejprve je blok rozdělen do 4 kvadrantů a pro každý kvadrant se vypočítá průměrná hodnota pixelů podle následujících rovnic:

$$A_i = \frac{1}{n}\sum_{j=1}^{n} r_j^i$$

$$V_i = \sum_{j=1}^{n}((r_j^i)^2 - A_i^2)$$

Poté je blok natočen do kanonické pozice – tj. pozice ve které je „světlost" kvadrantů shodná z některým z následujících vzorů (světlost=průměrná hodnota jasu pixelů)

class 1 class 2 class 3