

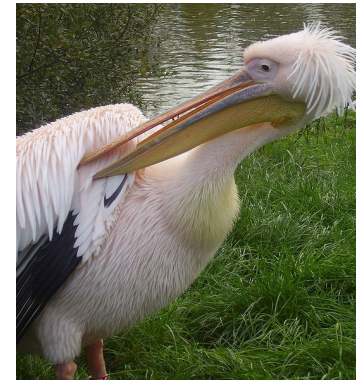
# Fraktálová komprese obrázků

Marek Fišer  
25. 11. 2010

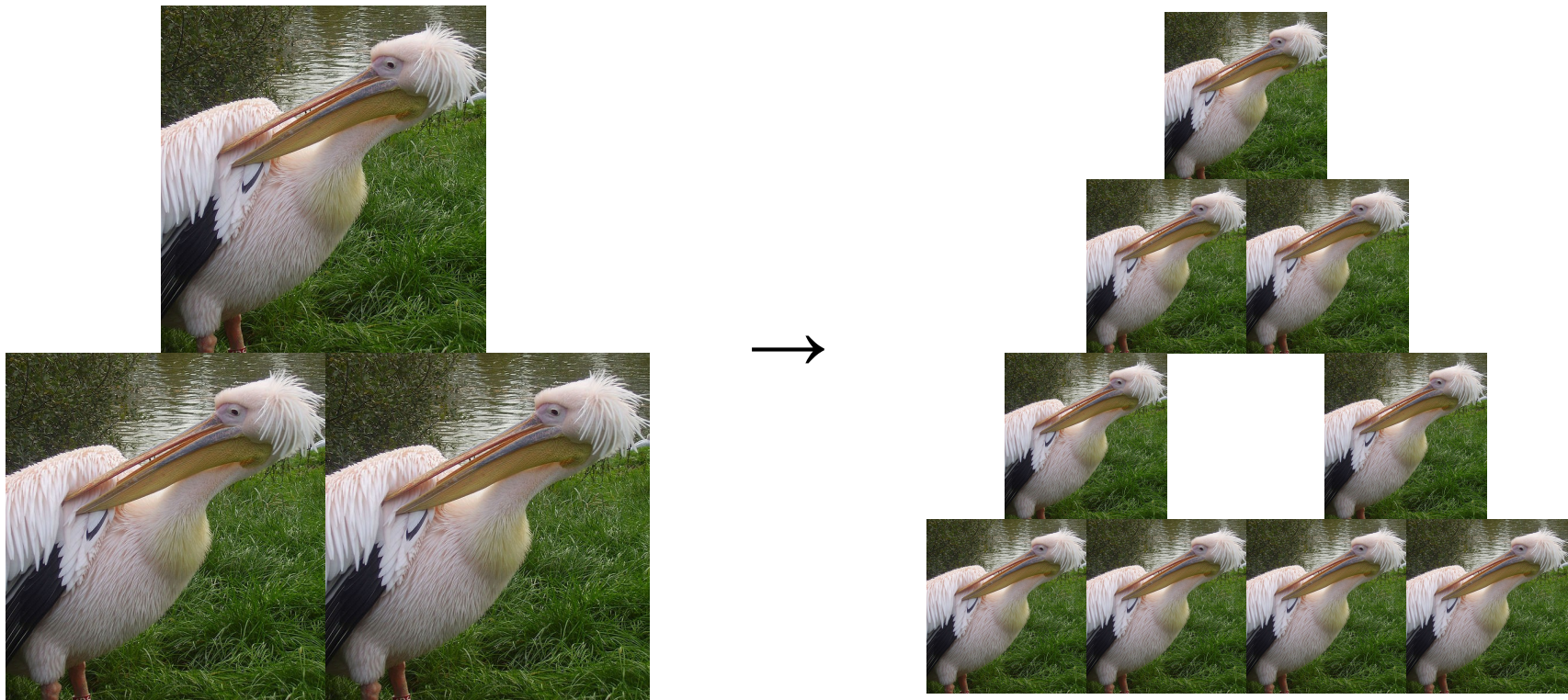
# Co je to fraktál?

- množina, jejíž Hausdorffova dimenze je (ostře) větší než dimenze topologická
- je soběpodobný
- mívá na první pohled velmi složitý tvar, ale je generován opakovaným použitím jednoduchých pravidel

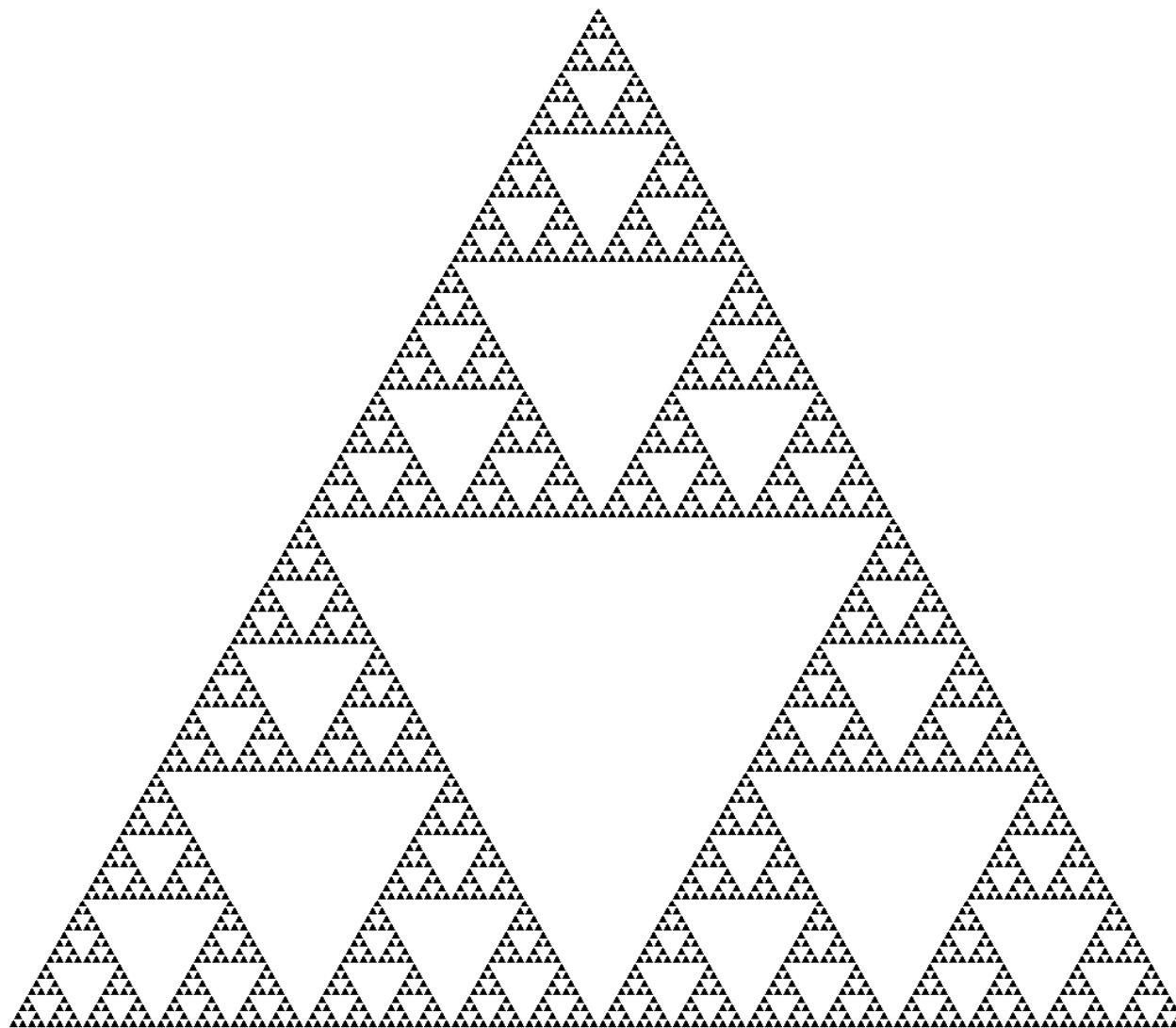
# Divná kopírka – motivace



# Divná kopírka

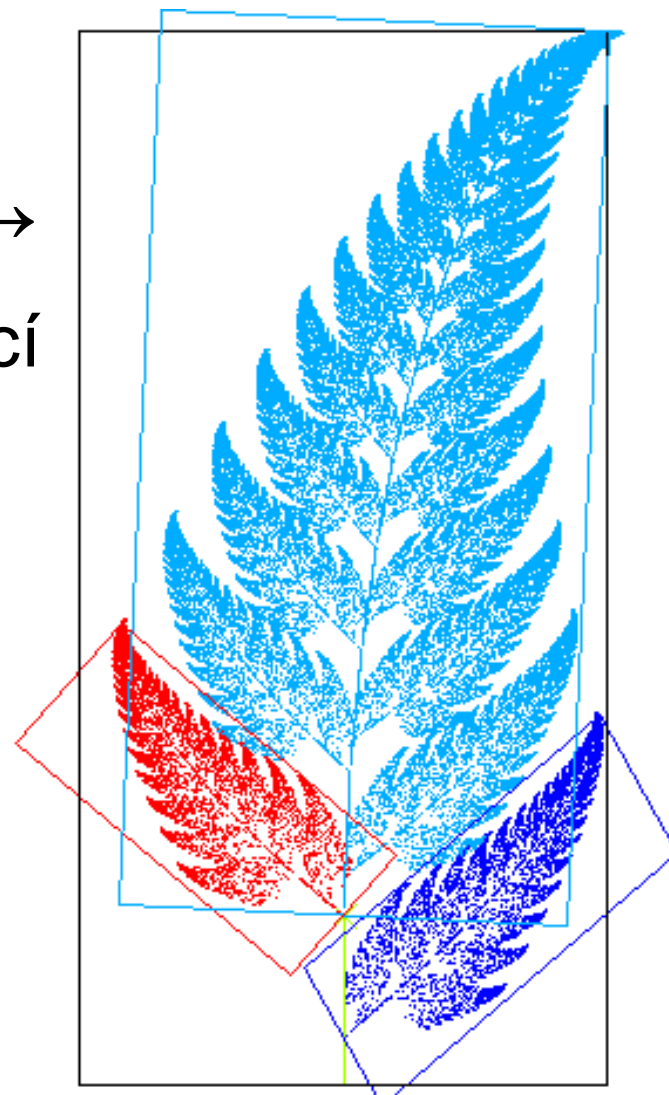


# Divná kopírka – výsledek



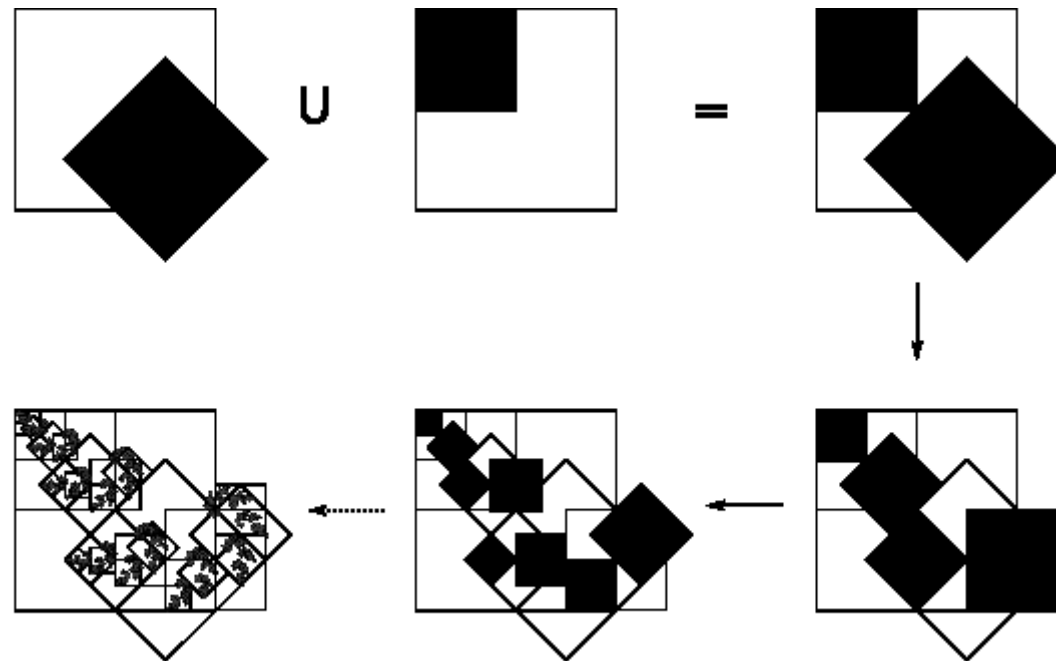
# Vylepšení kopírky

- rozšíření o transformace
- Barnsleyova kapradina →
  - lze uložit pomocí 4 transformací
  - každá transformace potřebuje 6 čísel
  - tj  $6 \times 4 \times 32 = 768$  bitů
  - nekonečný detail



# IFS – Iterated function system

- fraktál je tvořen sjednocením kopií sama sebe
- každá kopie je transformovaná nějakou funkcí



# Fraktálová komprese obrázků

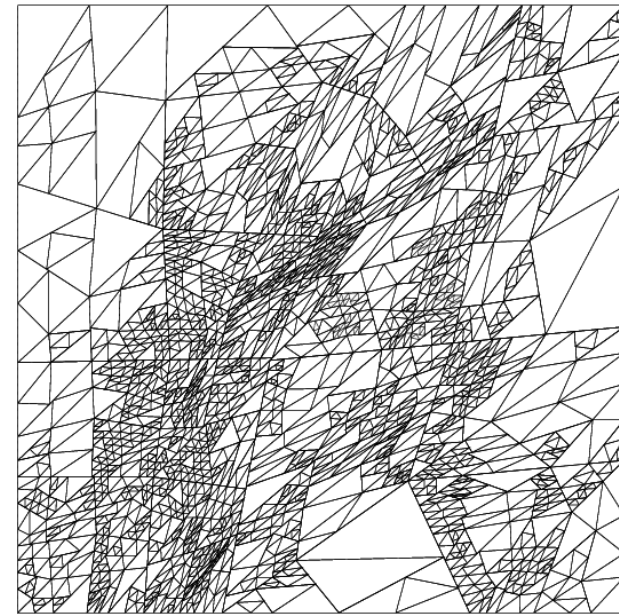
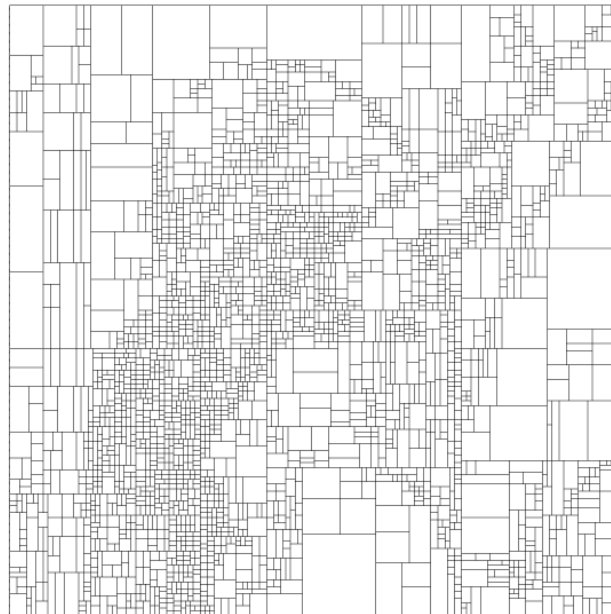
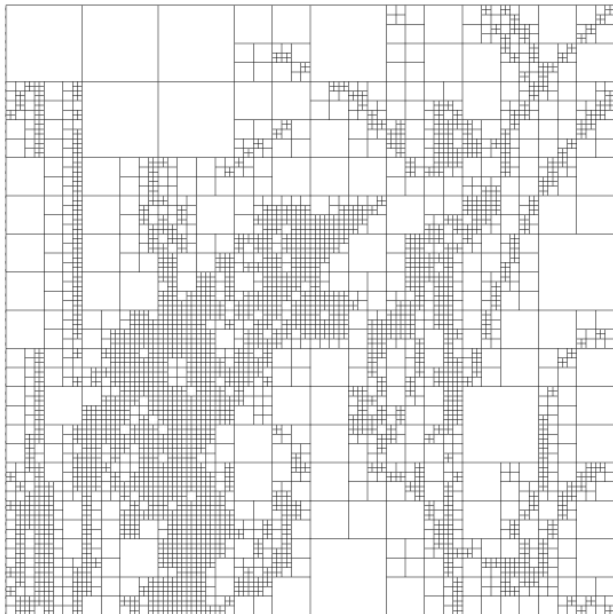
- IFS po částech
- po jakých částech?
- jaké transformace povolíme?
- jak je vypočítat?
- jak je uložit?
- jak docílit toho, aby se výsledek ustálil?
- z čeho začít?





# Po jakých částech?

- rovnoměrné čtverce
- quadtree
- obdelníky
- trojúhelníky



# Jaké transformace povolíme?

- v zásadě libovolné...
- posunutí (v  $x$  a  $y$ )
- otočení (0, 90, 180, 270)
- převrácení (horizontálně, vertikálně, dle osy  $x = y$ , dle osy  $x = -y$ )
- úprava jasu a kontrastu (zde se musí myslet na konvergenci)

# Jak transformace vypočítat?

- těžko
  - brutal-force niky nezklame
  - nebude to pomalé?
  - obrázek 1024x768, blok 8x8
    - zkusíme každých 8 pixelů blok 16x16 tj  $95 * 127 = 12\ 065$  pozic „okénka“, k tomu 8 transformací okénka, to jsme na  $96\ 520$  a to musíme zkoušet pro každý blok 8x8, kterých je  $128x96 = 12\ 288$  tj  $148\ 254\ 720$  porovnání obrázků 8x8 + počítání transformací při zkoušení
- neradil bych vám používat GetPixel() a SetPixel()

# Jak transformace uložit?

- co nejmenší :)
- vymyslet si je tak, aby to šlo
- v mém ukázkovém programu:
  - posunutí 16 bitů + 16 bitů (zde by šlo šetřit)
  - otočení a převrácení 3 bity
  - multiplikační konstanta 12 bitů
  - aditivní konstanta 9 bitů
  - dohromady 56 bitů = 7 bytů na část

# Konvergence transformací

- proč by to mělo konvergovat?
- jak toho docílit?
  
- kopírovat z větší části do menší (získávání detailů)
- při transformování násobit číslem menším než jedna
  - geometrická posloupnost (odpověď na otázku „Z čeho začít?“)

# Shrnutí

- uložené máme pouze informace o tom co je s čím podobné
- díky zmenšování a správnému transformování výsledek konverguje
- nezáleží na počátečním stavu
- obrázek nemá nativní velikost – díky způsobu generování lze obrázek vytvořit přímo v rozlišení v jakém ho potřebujeme (není třeba interpolovat)

Ukázka



# Zdroje

- <http://cs.wikipedia.org/wiki/Fraktál>
- [http://en.wikipedia.org/wiki/Barnsley\\_fern](http://en.wikipedia.org/wiki/Barnsley_fern)
- [http://en.wikipedia.org/wiki/Iterated\\_function\\_system](http://en.wikipedia.org/wiki/Iterated_function_system)
- <http://en.wikipedia.org/wiki/Pelican>