Aplikovaná informatika

Podklady předmětu

Aplikovaná informatika

pro akademický rok 2013/2014

Radim Farana

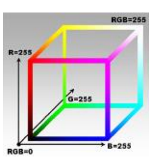
13

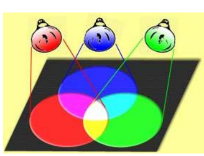
Obsah

- Ztrátová komprese.
  - Komprese JPEG.
  - Waweletová komprese.
- Fraktální komprese.

Modely barev

- ***RGB*** (Red-Green-Blue),

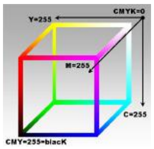


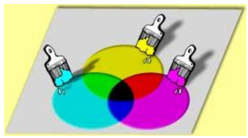


- Nestejné vnímání (necitlivost na modrou)
- relativní jas = 0,3 R + 0,59 G + 0,11 B

Modely barev

- ***CMYK*** (Cyan-Magenta-Yellow-blacK)



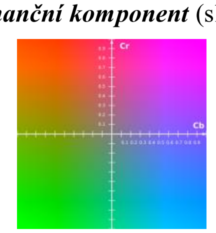


Modely barev

- ***HSB*** (Hue-Saturation-Brightness), který používá tři veličiny:
  - ***Hue*** (odstín), udává se ve stupních a popisuje barvu v rozsahu 0° - 360° na okraji tzv. barevného kola.
  - ***Saturation*** (sytost), udává sytost barvy v procentech od 0 % (bod je šedý v úrovni dané Brightness, tedy bez barvy) do 100 % (bod je barevný zcela podle Hue).
  - ***Brightness*** (jas), popisuje v procentech černobílý jas bodu od černé 0 % po bílou 100 %.

Modely barev

- ***YCbCr***, kde Y představuje složku jasu (luminance) a Cb s Cr představují modrý a červený ***chrominanční komponent*** (složku).



JPEG (Joint Photographic Experts Group)

- V současné době patří mezi nejvíce používané komprese u obrázků.
- Je vhodná pro komprimaci fotografií, nevhodná pro např. technické výkresy (čárové výkresy) – dochází k viditelnému rozmazání.

Princip:

- Části obrazu se transformují do frekvenční oblasti (výsledkem je matice „frekvenčních" koeficientů.
- Z matice koeficientů se odstraní koeficienty odpovídající vyšším frekvencím (rychlejší změny jasu – např. hrany v obraze).
- Zbývající koeficienty se vhodným způsobem zkomprimují.

JPEG

- JPEG komprimuje obrazy s 24 bitovou reprezentací barev (True Color). Algoritmus a z něj vycházející formát ukládání grafických dat byl převzat do norem ISO a CCITT.
- Dosahuje kompresního poměru 25: 1, při horší kvalitě výsledného obrazu až 80: 1 i více, využívá postupu nazvaného ***sekvenční komprese*** a z něj vycházející ***progresivní komprese***.

Převod do modelu YCbCr

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ -0{,}1687 & -0{,}3313 & 0{,}5 \\ 0{,}5 & -0{,}4187 & -0{,}0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

- Přičtení hodnoty 128 ke složkám ***chrominance*** má pouze technický význam, všechny tři složky jsou pak vyjádřeny pouze nezápornými hodnotami, takže se s nimi lépe pracuje.

Rozklad obrazu

- Obraz je rozdělen na čtvercové bloky po 8×8 bodech.
- V každém bloku je nejdříve od hodnot v něm odečtena hodnota 128, tím se hodnoty z intervalu 0 ÷ 255 převedou do intervalu -128 ÷127, čímž se sníží jejich absolutní velikost,
- následně je na každý blok samostatně provedena diskrétní kosinová transformace (***DCT – Discrete Cosine Transform***).

DCT

$$F(u,v) = \frac{1}{4} C(u) C(v) \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right)$$

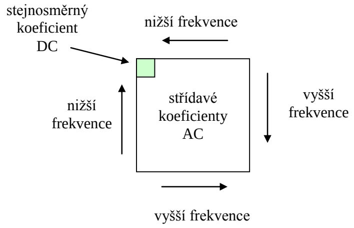
$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u,v = 0 \\ 1 & u,v \neq 0 \end{cases}$$

- odděleně pro jasovou složku a dvě barvové složky.

Význam koeficientů

- Koeficient $F(0, 0)$ reprezentuje stejnosměrnou složku (hodnoty kosinů jsou konstantní), ostatní koeficienty vyjadřují střídavé složky. Směrem k pravému dolnímu rohu (tj. směrem ke koeficientu $F(7, 7)$) se frekvence koeficientů zvyšují.

Význam koeficientů



Příklad

původní data

40	44	60	65	71	48	66	42
41	45	55	69	109	56	69	55
42	48	56	67	76	104	75	80
52	64	71	87	99	120	132	120
59	62	75	84	97	150	156	150
70	64	60	72	78	149	154	120
75	72	65	62	87	75	78	95
77	74	69	89	67	65	65	82

snížení o 128

-88	-84	-68	-63	-57	-80	-62	-86
-87	-83	-73	-59	-19	-72	-59	-73
-86	-80	-72	-61	-52	-24	-53	-48
-76	-64	-57	-41	-29	-8	4	-8
-69	-66	-53	-44	-31	22	28	22
-58	-64	-68	-56	-50	21	26	-8
-53	-56	-63	-66	-41	-53	-50	-33
-51	-54	-59	-39	-61	-63	-63	-46

DCT

-396	-125	-14	22	-8	6	-4	-1
-67	8	-45	-1	3	-3	-10	21
-104	101	-19	-21	33	-7	-14	16
28	-34	16	15	-21	15	-3	7
22	-5	2	-19	-2	12	-3	-5
-16	15	6	-4	-2	-5	1	1
-4	-9	8	10	-21	17	13	-25
-11	3	4	-6	3	-10	14	-4

Kvantování

- koeficienty $F(u, v)$ se dělí hodnotami kvantovací matice Q

$$C(u,v) = \mathrm{Int}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

kvantovací matice Q

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	92
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

výsledek

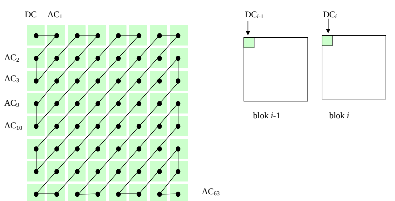
-25	-11	-1	1	0	0	0	0
-6	1	-3	0	0	0	0	0
-7	8	-1	-1	1	0	0	0
2	-2	1	1	0	0	0	0
1	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

velké množství koeficientů je nulových

QM-coder

- Poslední fází je statistické kódování hodnot matice **C** a jejich uložení na výstup.
- Používá se k tomu ***QM-coder***, což je již adaptivní způsob ***aritmetické komprese***, nebo ***Huffmanův kód***.
- Zatímco ostatní části standardu JPEG jsou volně dostupné, QM-coder je patentovaný, což je značná překážka při jeho použití v JPEG.
- Proto se v praxi převážně používá Huffmanovo kódování, i když s ním je účinnost komprese o 10 až 15 procent nižší.

Postup kódování bloku



- U stejnosměrných koeficientů se kóduje jejich rozdíl $DC_i - DC_{i-1}$,

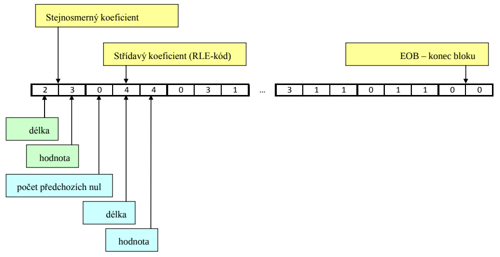
Postup kódování bloku

- Kódování je ve tvaru dvou hodnot. První hodnota uvádí počet nul od předchozího nenulového koeficientu (metoda RLE) a druhá hodnota reprezentuje nenulový koeficient.
- Vlastní hodnota nenulového střídavého koeficientu je dále kódována do tvaru dvojice čísel. První číslo udává délku čísla a druhé je vlastní hodnota koeficientu.

Postup kódování bloku

- Kódování každého nenulového koeficientu tedy představuje trojici čísel – počet nul, kód délky hodnoty koeficientu a samotná hodnota koeficientu.
- Pro zvýšení účinnosti se tato trojice neukládá na výstup přímo, ale používá se při tom statické ***Hufmanovo kódování***.

Příklad



Kód délky bloku

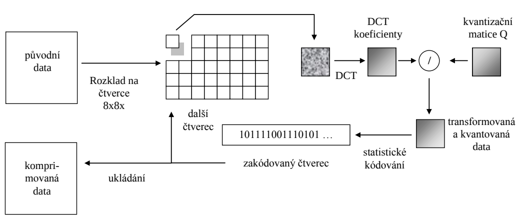
- Vyjádření celočíselné hodnoty pomocí dvojice *délka* + *hodnota* se nazývá kódování čísla s proměnnou délkou. Provádí se podle následující tabulky:

Délka	Rozsah hodnot
1	-1, 1
2	-3 ÷ -2, 2 ÷ 3
3	-7 ÷ -4, 4 ÷ 7
4	-15 ÷ -8, 8 ÷ 15
5	-31 ÷ -16, 16 ÷ 31
…	

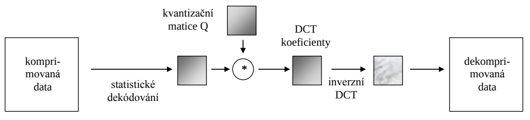
Kód délky bloku

- V tabulce 1 udává délka – kolika bity je zakódována vlastní hodnota čísla, např. hodnoty v rozsahu -15 ÷ -8, 8 ÷ 15 budou zakódovány pomocí 4 bitů.
- Samotné kódování je jednoduché. Nejprve se záporné hodnoty přičtením vhodného čísla převedou na nezáporné, např. hodnoty -15 ÷ -8 se přičtením čísla 15 převedou na hodnoty 0 ÷ 7, čímž získáme souvislý interval 0 ÷ 15 a hodnoty jeho prvků přímo představují čtyřbitová čísla 0000B ÷ 1111B.

Komprese JPEG



Dekomprese JPEG



Inverzní DCT

$$f(i,j)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)$$

$$C(u),C(v)=\begin{cases}\frac{1}{\sqrt{2}} & u,v=0\\ 1 & u,v\neq 0\end{cases}$$

Parametry komprese

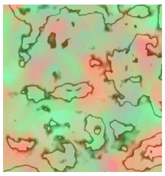
- Úroveň kvantování a tím i kvalitu obrazu a velikost kompresního poměru může uživatel měnit pomocí Q-faktoru, který lze volit v rozmezí 1−100.
- Při kvantování jsou koeficienty matice Q děleny hodnotou 50/Q-faktor.
- Z toho plyne, že čím bude Q-faktor menší, tím budou kvantovací koeficienty větší. Tím zároveň budou menší vypočítané koeficienty matice **C**, což v důsledku znamená větší kompresní poměr, ale i vyšší ztráty a tím i zhoršení kvality obrazu.
- Implicitně bývá nastavena hodnota Q-faktoru = 75. Při ní se dosahuje typický kompresní poměru 1:10 a současně zůstává

Parametry komprese

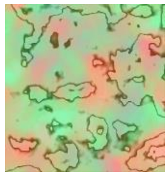
- Ještě vyšší komprese dosahuje ***progresivní komprese***, která pracuje na stejném principu, ale postupně provádí kompresi od nejvyššího stupně komprese a v jednotlivých cyklech kompresi opakuje se snižujícím se stupněm komprese. Díky více cyklům lze dosáhnout dalšího zmenšení výsledného souboru (za cenu zhoršení jeho kvality).

Komprese JPEG

Původní obrázek ve formátu BMP
(velikost **66 614** B)



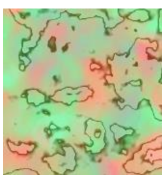
Bezeztrátová komprese JPEG
(velikost **56 463** B),



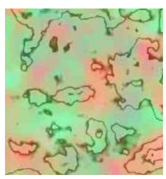
Informační obsah

Malé ztráty informace

Odstranění vysokých frekvencí (šum)
(velikost **13 816** B)

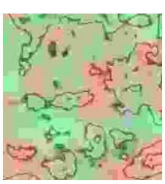


Zmenšení na desetinu obsahu
(velikost **5 696** B),



Velké ztráty informace

Odstranění detailů 5 % velikosti
(velikost **2 596** B)



Zmenšení na 3 % velikosti
(velikost **2 001** B),

JPEG 2000

- Jeho základem je použití ***diskrétní vlnkové (waveletové) transformace***, skalárního ***kvantování*** a entropického kódování. Díky tomu dosahuje lepších vlastností.

Vlnková transformace

- Vlnková transformace (wavelet transform) je výsledkem snahy o časově-frekvenční popis signálů.
- Fourierova transformace poskytuje informaci o tom, jaké frekvence se v signálu vyskytují, ale nevypovídají o jejich poloze v čase, proto je vhodná pro popis stacionárních signálů.
- Řešením problému je použití okna, které ohraničí krátký úsek signálu a umožní určení spektra signálu v tomto intervalu.

Vlnková transformace

- Myšlenkou vlnkové transformace je vhodnou změnou šířky okna a jeho tvarem dosáhnout optimálního poměru rozlišitelnosti v čase a frekvenci.
- Pro nízké frekvence je okno širší, pro vysoké užší. Toto okno se nazývá ***mateřská vlnka*** (mother wavelet) ψ. Pomocí parametru s (měřítko) je možné měnit její šířku, parametrem τ (poloha) se mění její poloha na časové ose:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \qquad s,\tau \in R,\ s \neq 0$$

Vlnková transformace

- Spojitá vlnová transformace je definována pro signály s konečnou energií $\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty$

$$Wf(\tau, s) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \overline{\psi}\left(\frac{t-\tau}{s}\right) dt$$

kde je $\overline{\psi}$ – číslo komplexně sdružené.
Výsledkem pro jednorozměrný signál je dvourozměrná funkce, nazývaná vlnkové koeficienty *Wf*(*τ*, *s*).

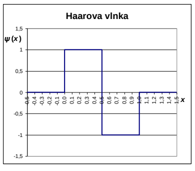
Mateřská vlnka

- Příkladem používaných mateřských vlnek je vlnka ***Mexican hat***,

$$\psi(x) = \frac{2}{\sqrt{3}} \pi^{\frac{1}{4}} \left(1 - 2x^2\right) e^{-x^2}$$

- nebo ***Haarova vlnka***

$$\psi(x) = \begin{cases} 1 & 0 \le x < \frac{1}{2} \\ -1 & \frac{1}{2} \le x < 1 \\ 0 & \text{jinak} \end{cases}$$



Vlnková transformace

- Vhodnou volbou závislostí parametrů s a τ můžeme vytvořit z vhodné vlnky ψ ortonormální bázi:

$$s = 2^p \quad \tau = 2^p k \quad p, k \in Z$$

$$\psi_{k,p}(t) = \frac{1}{\sqrt{2^p}} \psi\left(\frac{t - 2^p k}{2^p}\right)$$

kde *p* odpovídá měřítku.

Aplikovaná informatika 37

Vlnková transformace

- Díky ortonormalitě pak takto zvolená vlnka umožňuje neredundantní dekompozici signálu, tzv. analýzu s mnoha rozlišeními.
- Tento princip je základem ***diskrétní vlnkové transformace*** (Discrete Wavelet Transform – DTW).

Aplikovaná informatika 38

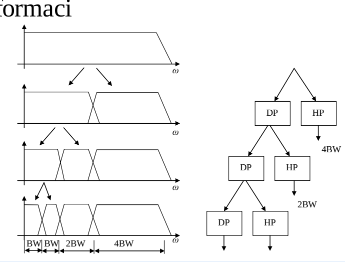
Vlnková transformace

- Vlnková funkce ψ se chová jako pásmová propust filtrující vstupní signál kolem centrální frekvence, která je závislá na měřítku mocninou dvou.
- V následujícím měřítku je filtrována horní polovina pásma předchozí dolnofrekvenční části signálu.
- S rostoucí frekvencí roste šířka pásma (BW) tohoto filtru.
- Činitel kvality **Q** je tak konstantní pro celou množinu měřítkem odvozených filtrů. Pro zvolené minimální měřítko však zůstává nepokryt rozsah nízkých frekvencí. Proto je od vlnky odvozena měřítková funkce φ (scaling function), která má charakter dolní propusti.

Aplikovaná informatika 39

Vlnková transformace

- Frekvenční pohled na diskrétní vlnkovou transformaci



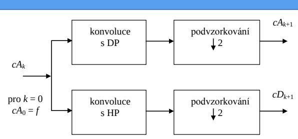
Realizace vlnkové transformace

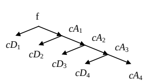
- Diskrétní vlnkovou transformaci je možno realizovat rychlým algoritmem, využívajícím ***FIR (Finite Impulse Response) filtry*** a ***podvzorkováním*** (decimací).
- Oba filtry, dolní propust DP (*h* - scaling filter) a horní propust HP (*g* - wavelet filter), tvoří pár kvadraturních zrcadlových filtrů (QMF), které mají komplementární propustná pásma.
- Výstupy obou filtrů jsou podvzorkovány na polovinu vstupních vzorků. Horní propust poskytuje koeficienty tzv. detailů DWT (cD), dolní propust koeficienty tzv. aproximace (cA).

Realizace vlnkové transformace

- Díky decimaci je celkový počet koeficientů po jednom kroku stejný jako počet vstupních vzorků. Koeficienty aproximace je možné dále analyzovat shodným rozkladem filtry a obdržet tak další soubor koeficientů aproximace a detailů. Tak je možno pokračovat až do vyčerpání vstupní sekvence

Realizace vlnkové transformace





- Jednotlivé konvolute s dedikací je možno formalizovat

$$cA_{p+1}(k) = \sum_{m=1}^{N} h(m-2k)cA_p(m)$$

$$cD_{p+1}(k) = \sum_{m=1}^{N} g(m-2k)cA_p(m)$$

JPEG 2000

- Oproti formátu JFIF pracuje JPEG 2000 s obrázkem jako celkem a převádí je na popisy pomocí vlnkové transformace.
- Převod je víceprůchodový, počet průchodů určuje kompresní poměr a kvalitu dekomprimovaného obrázku (méně průchodů = vyšší kompresní poměr = nižší kvalita obrazu).
- Každému průchodu odpovídá zvláštní datový blok komprimovaného souboru. Pro rozměrné obrázky však rychle rostou nároky na paměť.
- Proto se používá technika rozdělení vstupního obrazu na menší části – ***dlaždice*** (tile), tento proces je nazýván tilling.

JPEG 2000

- Následně se každá dlaždice komprimuje odděleně. K tomu je použita L-úrovňová DWT s biortogonálními spline vlnkami 9/7 (v aritmetice s pohyblivou řádovou čárkou) nebo s biortogonálními spline vlnkami 5/3 (v celočíselné aritmetice, v případě bezeztrátové komprese).
- L-úrovňová transformace má L+1 rozlišení obrazu – L je rozlišení obrazu a 0 je nejnižší frekvence. Ze 4 podpásem (subbandů) rozlišení j se složí jeden s rozlišením $j + 1$.

JPEG 2000

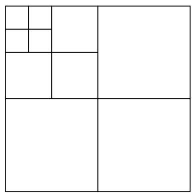
- Následuje ***kvantování*** jednotlivých podpásem – reálné koeficienty (v pohyblivé řádové čárce) se kvantují na celá čísla pomocí uniformního kvantování, krok kvantování je možné volit zvlášť pro každé podpásmo (a tím také částečně řídit kvalitu).

$$q = sign(y)\left\lfloor \frac{|y|}{\Delta_b} \right\rfloor$$

kde je q – kvantovaná hodnota,
y – transformovaný koeficient podpásma,
Δ_b – kvantovací krok, často má hodnotu 10.

JPEG 2000

- Po kvantování je každé podpásmo rozděleno na packet partitions – čtvercové nepřekrývající se oblasti



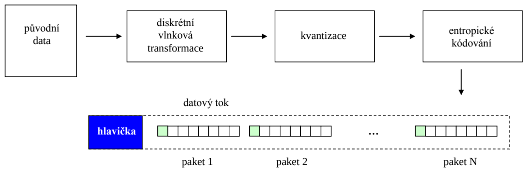
JPEG 2000

- Každá packet partition je dělena na bloky stejné velikosti (kromě bloků na kraji obrazu), všechny bity všech koeficientů v bloku jsou pak kódovány pomocí ***EBCOT*** (Embedded Bitplane Coding with Optimal Truncation) v pořadí od nejvýznamnějšího po nejméně významný:
  - Koeficienty se rozdělí do bitových rovin.
  - Bitové roviny obsahující pouze nuly jsou přeskočeny (uchová se jen jejich počet), začne se první rovinou obsahující alespoň jednu jedničku.
  - Bity se z roviny načítají po čtyřřádkových blocích, v rámci bloku po čtyřbitových sloupcích.
  - Každá rovina je pak kódována ve třech průchodech:
    - Significance Propagation – zakódují se bity, které nejsou důležité, ale sousedí s alespoň jedním důležitým (v některém z 8 směrů).
    - Magnitude Refinement – zakódují se bity z míst, které se byly v minulých rovinách důležité.
    - Cleanup Pass – zakódují se všechny zbylé bity.

JPEG 2000

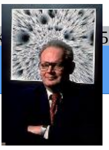
- Takto získané bity se kódují binárním kontextovým aritmetickým kodérem – MQ kodérem. Kontext koeficientu je tvořen stavem jeho 8 sousedů (okno 3×3 bity). Výsledný proud bitů je rozdělen na pakety – paket spojuje jeden ze tří průchodů všech bloků přes celou packet partition. Pomocí vypouštění méně důležitých paketů se řídí kvalita komprese.

JPEG 2000



Fraktální komprese

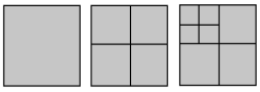
- Hledání podobných tvarů
  - uložení informace o tvaru, posunutí, otočení, změně velikosti a barvy.

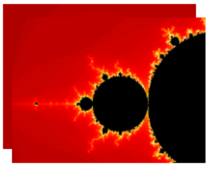


Benoit B. Mandelbrot

* 20. 11. 1924 Warsaw, Poland (French mathematician)

http://www.math.yale.edu/mandelbrot/





rychlá dekomprese.

Dimenze

- S fraktální geometrií úzce souvisí otázka měření dimenzí a členitosti objektů. Obvykle pracujeme s topologickou dimenzí, kterou můžeme chápat tak, že spojitým zobrazením přiřadí každé křivce přímce, každé ploše rovinu atd.
- Naproti tomu **Hausdorfova dimenze** je mírou členitosti objektu. Předpokládejme, že část křivky je rozdělena na N stejných částí, přičemž každá takto vzniklá část je stejná jako původní část křivky s tím, že je zmenšena v poměru r. Platí vztah $N.r = 1$. Pro čtverec rozdělený na $N.N$ částí je $N.r^2 = 1$, pro krychli obdobně $N.r^3 = 1$ atd., obecně je tedy dimenze d objektu exponentem zmenšovacího poměru:

$$N.r^d = 1$$

Aplikovaná informatika 52

Dimenze

- Eukleidovské metriky předpokládají celočíselné dimenze objektů. Nás ale zajímá, co se stane, pokud dimenze *d* nebude celočíselná.
- Dimenze *d* pak bude představovat fraktální dimenzi.
- Logaritmováním předchozího výrazu pak získáme experimentální způsob určení ***Hausdorfovy dimenze***, neboli ***fraktální dimenze***:

$$d = \frac{\log N}{\log \frac{1}{r}} = -\frac{\log N}{\log r}$$

Aplikovaná informatika 53

Soběpodobnost

- ***Soběpodobnost*** (matematicky se tato vlastnost nazývá ***invariance vůči změně měřítka***) je taková vlastnost objektu, že objekt vypadá podobně, ať se na něj díváme v jakémkoliv zvětšení.
- Soběpodobnost je hlavním znakem fraktálních útvarů a většinou je také považována za jejich definici.

Aplikovaná informatika 54

Soběpodobnost

- Soběpodobná množina A *n*-dimenzionálního Euklidovského prostoru IR^n je taková množina, pro níž existuje konečně mnoho kontrahujících zobrazení $\phi_1, \ldots, \phi_n$ takových, že A vznikne jako:

$$A = \bigcup_{i=1}^{n} \phi_i(A)$$

Soběpodobnost

- Soběpodobná množina vzniká opakováním sebe sama při použití transformací jako je změna měřítka, rotace, posunutí, zkosení apod., jsou invariantní vůči změně měřítka apod.
- Princip opakování tvarů je přitom v přírodě velmi častý, viz např. postupný růst schránky mlžů, tvořící archimedovu spirálu.

Afinní transformace

- ***Afinní transformace*** v IR je transformace f: IR → IR ve tvaru:

$$\mathrm{f}(x) = a.x + b,\ \forall\ x \in \mathrm{IR}$$

kde *a*, *b* – reálné konstanty.
Na daném intervalu I = (0, 1) představuje f(I) nový interval délky $|a|$. Levý krajní bod 0 intervalu je přenesen do *b* a f(I) leží nalevo nebo napravo od něj podle znaménka parametru *a*.

Afinní transformace

- ***Afinní transformace*** v IR^2 je dvojdimenzionální transformace
f: $\mathrm{IR}^2 \rightarrow \mathrm{IR}^2$ ve tvaru:

$$w(x, y) = (a.x + b.y + e, c.x + d.y + f), \forall\ x, y \in IR^2$$

Kde *a*, *b*, *c*, *d*, *e*, *f* – reálné konstanty.
Nebo v maticovém tvaru:

$$\mathrm{w}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} e \\ f \end{vmatrix} = \mathbf{A.x} + \mathbf{T}$$

Afinní transformace

- Inverzní afinní transformace je reprezentována následovně:

$$\mathrm{w}^{-1}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} d & -b \\ -c & a \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} -d & b \\ c & -a \end{vmatrix} . \begin{vmatrix} e \\ f \end{vmatrix}$$

Za podmínky $a.d - b.c \neq 0$

Afinní transformace

- Speciálními případy afinní transformace jsou ***dilatace*** (pro $a = d$ jde o ***změnu měřítka***):

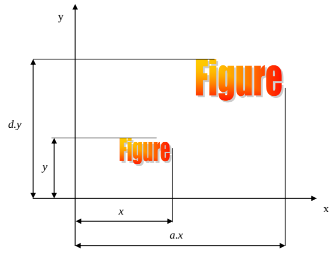
$$\mathrm{w}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

- ***rotace***:

$$\mathrm{w}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

Afinní transformace

- Příklad afinní transformace – dilatace



Afinní transformace

- Jako podobnostní pak nazýváme afinní transformaci, pokud má některý z následujících tvarů:

$$\mathrm{w}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} r.\cos\alpha & -r.\sin\alpha \\ r.\sin\alpha & r.\cos\alpha \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} e \\ f \end{vmatrix},$$

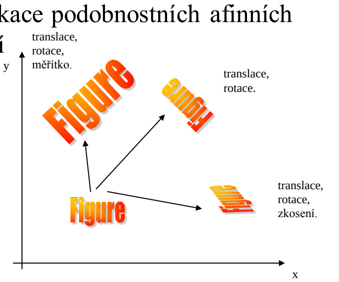
$$\mathrm{w}\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} r.\cos\alpha & r.\sin\alpha \\ r.\sin\alpha & -r.\cos\alpha \end{vmatrix} . \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} e \\ f \end{vmatrix}.$$

- Pro translaci (e, f), měřítko r dané nenulovým reálným číslem a úhel rotace α, kde

$$0 \le \alpha \le 2\pi$$

Afinní transformace

- Příklad aplikace podobnostních afinních transformací



Afinní transformace

- Pro Eukleidovskou metriku

$$\mathrm{d}(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}, \ \forall\ x, y \in \mathrm{IR}^2$$

můžeme každé afinní transformaci w přiřadit jediné reální číslo s:

$$\rho(\mathrm{w}(\mathbf{x}) - \mathrm{w}(\mathbf{y})) < s.\rho(\mathbf{x} - \mathbf{y})$$

kde s je nejmenší možné reálné číslo, které splňuje výše uvedenou podmínku.

Afinní transformace

- Velikost koeficientu *s* je důležitá pro typ transformace:
  - $s < 1$ je w kontrakcí,
  - $s = 1$ je w symetrií,
  - $s > 1$ je w extrakcí.
- Jejich využití je základem fraktální komprese využívající ***systému iterovaných funkcí*** (IFS).

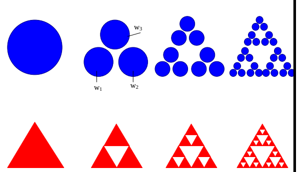
Afinní transformace

- Příklad: ***Sierpinského trojúhelník*** s transformacemi

$$w_1\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} 0,5 & 0 \\ 0 & 0,5 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} 0 \\ 0 \end{vmatrix}$$

$$w_2\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} 0,5 & 0 \\ 0 & 0,5 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} 0,5 \\ 0 \end{vmatrix}$$

$$w_3\left(\begin{vmatrix} x \\ y \end{vmatrix}\right) = \begin{vmatrix} 0,5 & 0 \\ 0 & 0,5 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} 0,25 \\ 0,5 \end{vmatrix}$$



Afinní transformace

- Pro aplikaci na kompresi černobílého obrazu rozšíříme afinní transformaci do podoby:

$$w\left(\begin{vmatrix} x \\ y \\ z \end{vmatrix}\right) = \begin{vmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ z \end{vmatrix} + \begin{vmatrix} e \\ f \\ o \end{vmatrix}$$

Kde *s* a *o* slouží k modifikaci jasové složky.

Fraktální komprese

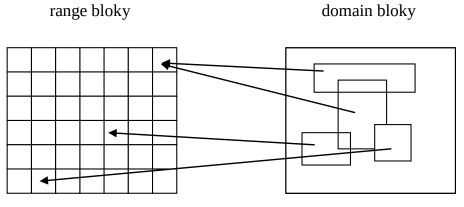
- Algoritmus komprese nejprve rozdělí komprimovaný obraz na nepřekrývající se range bloky velikosti 8x8 (4x4) bodů pokrývající celý obraz.
- Následně se vyhledávají domain bloky, které jsou range blokům podobné a mohou se překrývat.
- Procházíme obraz zleva doprava, shora dolů s krokem k bodů a které mají dvojnásobnou velikost než range bloky.

Fraktální komprese

- V každém doménovém bloku jsou průměrovány sousední body a jsou uloženy do nového doménového bloku stejné velikosti jako range blok. Novým doménovým blokem přepíšeme blok původní. Následně pro každý range blok R_i najdeme v souboru doménových bloků ten blok D^B, který se mu nejvíce podobá

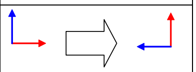
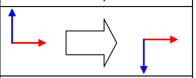
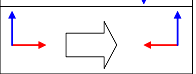
Fraktální komprese

- Princip rozmístění range a domain bloků v obrazu



Fraktální komprese

- Základní používané afinní transformace

Popis	Matice A transformace	Příklad
Rotace o 270°	$\begin{vmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	
Překlopení přes horizontální osu	$\begin{vmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	
Překlopení přes vertikální osu	$\begin{vmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	

Fraktální komprese

- Pro každý doménový blok D*j* a transformaci w_t (t = 1, 2, …, 8) se vypočte
  $\mathrm{D}_j^t = w_t(\mathrm{D}_j)$
  a podle následujících vztahů se určí koeficienty *s* a *o*:

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

Fraktální komprese

- Koeficienty *s* a *o* jsou následně ***kvantovány*** a pro kvantované koeficienty se podle následujícího vztahu vypočítá chyba podobnosti bloků :

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

Fraktální komprese

- Najdeme blok s minimální chybou

  $E(D^t_j, R_i)$

  (t = 1, 2, …, 8) a v souboru doménových bloků najdeme nejpravděpodobnější blok:

  $D^B = \min_{t=1,2,\dots,N_D} (D_j)$

Kde je N_D – počet doménových bloků

Fraktální komprese

- Výstupem je pak kód $w_i = (e_i, f_i, m_i, o_i, s_i)$. Posloupnost těchto transformací je možno následně komprimovat bezeztrátovou kompresí.
- Je zřejmé, že doba komprese bude značná, oproti tomu dekomprese probíhá i přes iterační postup výrazně rychleji.