

Formát digitálních obrázků

Obrázek 1

- Máme tři druhy cípu. (Většina sávací 2, ryby, ještěři a ptáci 4)
- Barvy, které lze pozorovat lidským okem se dají vyjádřit jako (in. komb. R, G, B . Čili: $(R, G, B) \in [0, 1]^3$
 $\{0, 1, \dots, 255\}^3$ (24 bitů, ale používá se až 48 bitů)
- Emisní zařízení - monitor - vyzařují kombinaci těchto barev (Aditivní model)
- Absorbční zařízení - čistě materiální - absorbují kombinaci těchto barev. (Subtrahční model)

$$C = 1 - R \quad \text{azurová}$$

$$M = 1 - G \quad \text{purpurová}$$

$$Y = 1 - B \quad \text{žlutá}$$

K

černá - v praxi je lepší nahradit smíšením všech tří barev

červená - kombinace totiž příliš prokáže

- kombinace je nekladnější

- pro jemné detaily, jako písmo, musí být perfektně seřazena

- RGB model je ~~vysoká~~ výsoká korelovaný.

- Při přechodu z CB TV vysílače na barvu se objevil YUV model.

$$Y = 0.299R + 0.587G + 0.114B \in [0, 1] \quad \text{Jas}$$

$$U = B - Y \in [-0.886, 0.886]$$

$$V = R - Y \in [-0.701, 0.701]$$

} Barvosné složky

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- V počítačích se tomto říká YCbCr model, $(R, G, B) \in \{0, \dots, 255\}^3 \Rightarrow (Y, C_b, C_r)$

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

2. řádek matice jsme získali jako $\frac{1}{2 \cdot 0.886}$

3. řádek jako $\frac{1}{2 \cdot 0.701}$

Vektorový formát (EPS) Nevhodný pro steganografii.

Obraz 2

Rastrový formát (BMP, PNG)

- Matice, kde každá složka obsahuje údaj o barvě příslušného pixelu.
- Bezstrátový formát.
- BMP umožňuje RLE
- PNG používá DEFLATE (jako ZIP) (LZ77 + Huffman)

Paletový formát (PNG, GIF)

- Matice, kde složky udávají číslo barvy
- Navíc paleta, která číslům přiřazuje barvu
- Paleta má obvykle 2-256 barev.
- GIF používá LZW kompresi (LZW84)
- Možnosti:
 - 1) Defaultní paleta
 - 2) Paleta šita na míru obrazu.
- Pokud obrazek obsahuje jen omezené množství barev, tak není problém
- Sestavení palety:
 - 1) Vyberu nejčastější barvy - nic na
 - 2) Budu usilovat o to, aby i významné ale odlišné barvy byly v paletě (median-cut algorithm)
- Snížení barevné hloubky obrazu
 - 1) Každou barvu zaochutím na nejbližší barvu v paletě
 - 2) Zaochutím a chybou zaochutění přenesu na dosud neochochutěné pixely.

Floyd - Steinberg dithering (Chybové difuze)

Postupujeme po řádcích a modifikujeme pixely obrazu

...	a_{ij}	$a_{i,j+1}$
$a_{i+1,j-1}$	$a_{i+1,j}$	$a_{i+1,j+1}$

$$\bar{a}_{ij} = \text{nejbližší barva v paletě k } a_{ij}$$
$$e_{ij} = a_{ij} - \bar{a}_{ij}$$

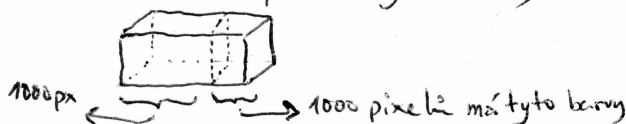
např. v paletě
mám číselnou hodnotu
 $0010FD = 0000FF$

...	$\bar{a}_{ij}$	$a_{i,j+1} + \frac{7}{16} e_{ij}$
$a_{i+1,j-1} + \frac{3}{16} e_{ij}$	$a_{i+1,j} + \frac{5}{16} e_{ij}$	$a_{i+1,j+1} + \frac{1}{16} e_{ij}$

$$a_{i,j+1} = a_{i,j+1} + \frac{7}{16} e_{ij}$$

median-cut: Na barvy se díváme jako body v prostoru $\{0, 255\}^3$. Každá barva (bod) má vliv podle toho, jak moc je zastoupena v obrazech.

- Tuto rozdělnu na 256 kvádřů v každém kvádři spočítáme těžiskové barvy $\rightarrow$ vloží se do palety.
- Rozdělení na kvádry se provádí rekursivně:
 - 1) barvy obracejíme nejmenším kvádrem.
 - 2) kvádr rozdělnu podle nejdelší osy v medianu.



- Obvykle ztratou
- Spáření - přechod do zcela jiné reprezentace, kde lze částečně zanedbat, aniž by se tím zásadně snížila kvalita obrázku.

JPEG kóduvání:

1) Rozklad na jas Y a barvoslepné složky C_b a C_r .

Kazda 2 techto slozek datu matice s hodnotami $\{0, 1, \dots, 255\}$.
S kazdou datu pracujeme samostatne.

2) Rozdělení na bloky a podzorkování

- Obrázek se rozdělí na bloky 8×8 pixelů.
- Barvoslepe složky se obvykle podvazávají předtím než jsou rozděleny na bloky.
- Pokud strany obrázku nejsou dělitelné 8 doplní se pixely, které se zase při dekódování ořežou.

Co je podvzorkování (subsampling):

- Barvonosna časť obrázku se zmenší typicky o $1/2$ v každém směru

ve skutečnosti: $16 \times 16 \rightarrow 8 \times 8$

- lze ověřen zveřejnit jen v jednom směru nebo v oběma směry



- Pri: lehotoučim se zase naložie na pivochi veličnost
- ~~ke~~ ~~z~~eti Standard pripraviť: jine ponery treba $\leftarrow \frac{1}{2}$ $\uparrow \frac{1}{4}$
- Pro C_b a C_r lze volit různé ~~spis~~ ponery podrozchovani.

3) Diskretní kosinová transformace. (Od každé hodnoty odečteme 128.)

Pro každý blok velikosti 8×8 B spočítáme blok velikosti 8×8 d jeho lineární kombinací: koeficient lin. komb.

$$d[k, l] \quad d_{k, l} = \sum_{i, j=0}^T f(i, j, k, l) \quad R_{i, j}$$

$$\text{Lde } f(i, j, k, l) = \frac{w_k w_l}{4} \cos\left(\frac{\pi}{16} k(2i+1)\right) \cos\left(\frac{\pi}{16} l(2j+1)\right) \text{ ~~and~~$$

a $w_0 = \sqrt{1/2}$, $w_i = 1_{p_{10} i} > 0$.

Inverzni DCT je ~~stejný~~:

$$B_{ij} = \sum_{k,l=0}^7 f(i,j,k,l) \, d_{kl}$$

$$\begin{pmatrix} DC & AC & AC & -AC \\ AC & AC & & \\ : & & & \\ AC & & & AC \end{pmatrix}$$

- Existují implementace DCT které ~~pracují~~ pracují s celými čísly a jsou tedy „přátelštější“ k počítačům, ale dávají méně přesné výsledky. $\Rightarrow$ Mohou ovlivňovat statistickou distribuci DCT koeficientů.
- DCT je vlastně změna báze v prostoru všech matic velikosti 8×8 .
- Nová báze je $\{(f(i,j,k,l))_{i,j} : \forall k, l \in \{0,1,\dots,7\}\}$.

Tato báze je ON vzhledem k dot product.

- DCT koeficienty mohou teoreticky ležet v rozsahu $[-1024, 1024]$ nebývají však rovnoměrně rozloženy (koncentrací okolo 0)

4) Kvantizace

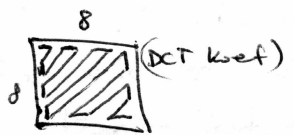
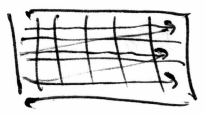
- Zde dochází ke ztrátě dat
- ~~Stará~~ Q matice 8×8
- Standard zadává sadu matic indexovaných číslem $q \in \{1 \dots 100\}$ ale do JPEGu lze zadat vlastní kvantizační matici.

$$D_{k,l} = \text{round} \left(\frac{d_{k,l}}{Q_{k,l}} \right), \quad k, l \in \{0, \dots, 7\}$$

- Čím vyšší hodnota $Q_{k,l}$ tím více se přibližuje koeficient k nule.
- Při rekonstrukci $d'_{k,l} = D_{k,l} Q_{k,l} \neq d_{k,l}$ (obecně)
- Pro barevnost složí se obvykle používají jiná matice než pro jasovou složku.

5) Bez ztrátová komprese

- Koeficienty se převedou na posloupnost



~~Nemulové koeficienty budou mít Huffmanovým kódováním (malé hodnoty mají kratší vyjádření, velké hodnoty dlouhé)~~

- Pro každý nemulový koeficient spočítáme kolik nul mu předchází a potom ho společně s tímto údajem zakódujeme Huffmanovým kódováním.
- Tímto se říká seřazení kódování (zig-zag, next block, zigzag, next block...)
- Alternativa je progresivní kódování. Nejprve kódujeme koeficienty v levém horním rohu každého bloku, potom všechny koeficienty na pozici (0,1), potom všechny na pozici (1,0) (tzn. opět zig-zag) (číslo opět zig-zag). Toto umožňuje postupně načítání obrázku.
- Formát JPEG je mnohem obecnější toto je jen nejpřehlednější příklad.

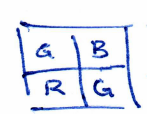
REJPA: IE neumí progresivní

Akvízie obrázku

→ 2000 hmot. fyz.

Obráz 5

- Založená na fotoelektrickém jevu  ⇒ Měříme napětí, které vznikne v důsledku akumulace náboje. (Fotodioda )
⇒ Měříme intenzitu, nikoliv barvu.

- Měření barvy: Před senzor se vloží Bayerův filtr: 

Převod na pixely:  →  $\bar{G} = (G_1 + G_2) / 2$ (nic moc)

- Interpolace  →   zprůměrníme oběma R a oběma B
 $(\frac{R_1 + R_2}{2}, G, \frac{B_1 + B_2}{2})$

 $(R, \frac{G_1 + G_2 + G_3 + G_4}{4}, \frac{B_1 + B_2 + B_3 + B_4}{4})$

(v okolí hran vznikají zipové artefakty)

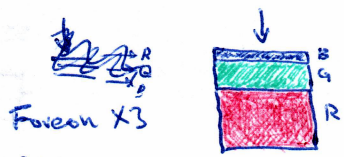
- Chytrá interpolace: Např. volí směr interpolace tak, aby byla maximalizována homogenita

Tento proces se nazývá demosaicing

- Vytváří vztahy mezi sousedními pixely.
- Modifikace pixelů tyto vztahy využívá.
- Analýza těchto vztahů lze detekovat např. falšování fotek nebo steganografii.

- Alternativy - Světlo rozdělení 

- Využit toho, že ~~světlo~~ světlo proniká hlouběji do silikonového čipu



- Další úpravy: • Upravení barev $R' = g_R R$ $\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} g_R & 0 & 0 \\ 0 & g_G & 0 \\ 0 & 0 & g_B \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$

Řeší problém ohledně osvětlení

Předpokládá se, že něco na obrázku je barev popř. řádek a podle toho se upraví, aby tomu tak skutečně bylo.

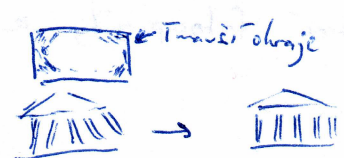
- Gamma korekce:

Problém - kamera vnímá jas lineárně

lidé ale vnímají jas logaritmicky (chrání proti oslnění?)
nelineárně

$$\begin{matrix} R' = R^p \\ G' = G^p \\ B' = B^p \end{matrix} \quad \text{ kde } p = \frac{1}{2.2}$$

- Odstranění šumu
- Odstranění defektů
- Digitální ostření
- Úprava chromy - problém při vysokém zoomu
- Úprava perspektivy - focení budov ze spodu



Toto vše včetně JPEG komprese vloží do obrázku (obrázku) vztahy, které je třeba při ukládání respektovat.

Šum

Obráz 6

Zdroje šumu: • Nedokonalosti a nečistoty v materiálu čipu. — Různá citlivost pixelů.

- Nedokonalosti v Bayerově filtru
- Teplo (Johnson-Nyquist noise) — šum vznikající přímo v elektronice
- Defektivní pixely (horší nebo zapadlé mrtvé)
- Odrazy světla uvnitř optiky fotoaparátu.
- Nečistoty na optice.
- Přexponování pixelu může vést k přetečením záboje do okolních pixelů.
- Amplifikace je významným zdrojem šumu při nízkých světelných podmínkách. Má konstantní rozdělení

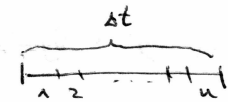
Amplifikace: 

Náboj → Napětí → Amplifikované napětí → Digit.

- Kvantizovaný šum: Demaskování, filtry, uložení do JPEG
- Výstřelový šum: vyplatí se kvantové povahy světla
počet fotonů, které dopadnou na snímač je velká veličina
Nechť X je počet fotonů, které dopadly na ~~snímač~~ pixel
a λ průměrný počet fotonů dopadajících na pixel za sekundu
pak $P(X=k) = \frac{e^{-\lambda \Delta t} (\lambda \Delta t)^k}{k!}$ a $EX = \text{var } X = \lambda \Delta t$

Zvyšujeme-li hodnotu $\lambda \Delta t$, pak ~~šum~~ relativní šum klesá:

$$\frac{\sqrt{\text{var } X}}{EX} = \frac{1}{\sqrt{\lambda \Delta t}} \rightarrow 0$$

Odkud se bere Poissonovo rozdělení? 

Předpokládejme, že v čase $\frac{\Delta t}{n}$ dopadne nejvýše jeden foton.

potom pravděpodobnost dopadu je $\frac{\lambda \Delta t}{n}$.

$$P(X=k) = \binom{n}{k} \left(\frac{\lambda \Delta t}{n}\right)^k \left(1 - \frac{\lambda \Delta t}{n}\right)^{n-k} = \frac{n(n-1)\dots(n-k+1)}{k!} \frac{(\lambda \Delta t)^k}{n^k} \frac{\left(1 - \frac{\lambda \Delta t}{n}\right)^n}{\left(1 - \frac{\lambda \Delta t}{n}\right)^k} \xrightarrow{\text{pro } n \rightarrow \infty} \frac{(\lambda \Delta t)^k}{k!} e^{-\lambda \Delta t}$$

- Některý šum je tedy pro dané zařízení stálý a jiný nikoliv.
- Stálý šum může sloužit jako otisk prostu daného zařízení (lze ho např. určit z nějaké jednobarevné plochy, ale snímků je potřeba víc, aby se eliminoval vhodný šum) — není nutné stačí použít denoising filter.
- Některé fotoaparáty se snaží tento stálý šum určit a odečíst od fotek.