I11 - DZO – Digitální obraz

Reprezentace digitálních obrazů, reprezentace barev, geometrické a jasové transformace, interpolace. Filtrace a konvoluce, zpracování ve Fourierově oblasti, odstranění šumu. (Digitální obraz)

https://moodle.fel.cvut.cz/course/view.php?id=5088 (přednášky 1:9)

Reprezentace digitálních obrazů

Proč je počítačové vidění těžké ?
Najděme alespoň 6 příčin.

m p 12/22

3D → 2D přináší ztrátu informace díky vlastnostem perspektivní transformace (matematická abstrakce, dírková komora).

Měřený jas je dán složitým fyzikálním postupem vytváření obrazu. Zář (angl. radiance) (≈ jas) závisí na typu světelných zdrojů, jejich poloze, intenzitě, poloze pozorovatele, lokální geometrii povrchu a odrazivosti povrchu. Obrácená úloha je špatně podmíněna.

Nevyhnutelná přítomnost šumu v každém měření ve skutečném světě.

Příliš mnoho dat Stránka A4, 300 dpi, 8 bit per pixel = 8.5 Mbytes.
Neprokládané video 512 × 768, RGB (24 bit) = 225 Mbits/sekundu.

Nutnost zahrnout interpretaci (bude brzy diskutováno).

Lokální okno v kontrastu s potřebou globálního pohledu.

Jednou z lidských schopností je zasazovat vjemy do rámce (kontextu), který je konkrétnímu člověku známý. Tuto schopnost formalizujeme pojmem interpreace, který matematicky vyjádříme jako zobrazení.

Interpretace: pozorování → logický model světa *nebo jinak* syntax → sémantika

Příklady:

- Pohled z okna → {prší, neprší}.
- Jablko na běžícím pásu → {třída 1, třída 2, třída 3}.
- Dopravní scéna → vyhledávání čísla auta.

Pixely odpovídají vzorkům
(nikoliv malým čtverečkům)

m p 7/49

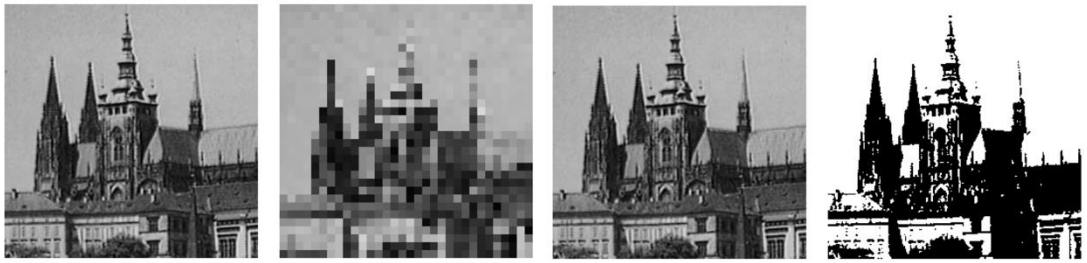
Digitalizace

- Digitalizace = vzorkování & kvantizace hodnoty obrazové funkce (též intenzity).
  - Vzorkování vybere ze spojité obrazové funkce vzorky. Výsledkem jsou vzorky v diskrétním rastru (je jich konečný počet). Hodnota vzorku zůstává "spojitá", tj. reálné číslo.
  - Kvantování rozdělí reálnou hodnotu vzorku na konečný počet hodnot (též přihrádek). U šedotónového obrazu např. na 256 hodnot.


Příklad: *šedý klín* 6 *šedotónových přihrádek*

- Digitální obraz se obvykle reprezentuje maticí.
- Pixel = akronym, angl. picture element.

Vzorkování, příklad 3 **Kvantování, příklad 4 (binární obraz)**

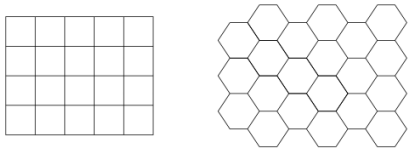


Originál 256×256 32×32 Originál 256 jasových úrovní 2 jasové úrovně

Vzorkování obrazu

Vzorkování obrazu zahrnuje dvě úlohy:

1. Vzor pro vzorkování (=uspořádání vzorkovacích bodů do pravidelného rastru).



(a) (b)

2. Rychlost/vzdálenost vzorkování (vyjadřuje Nyquist-Shannonova věta o vzorkování).
   - Vzorkovací frekvence musí být $2\times >$ než maximální frekvence v signálu; což je nejvyšší frekvence rekonstruovatelná z vzorkovaného signálu. Větu odvodíme, až budeme umět Fourierovu transformaci.
   - V obrazech se musí velikost vzorku (pixelu) být dvakrát menší než nejmenší detail, který chceme zaznamenat.

Několik definic vzdálenosti ve čtvercové mřížce

Euklidovská vzdálenost

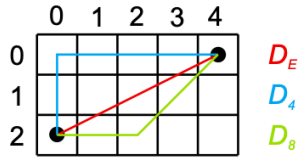
$$D_E((x,y),(h,k)) = \sqrt{(x-h)^2 + (y-k)^2}\,.$$

Vzdálenost městských bloků (též vzdálenost na Manhattanu)

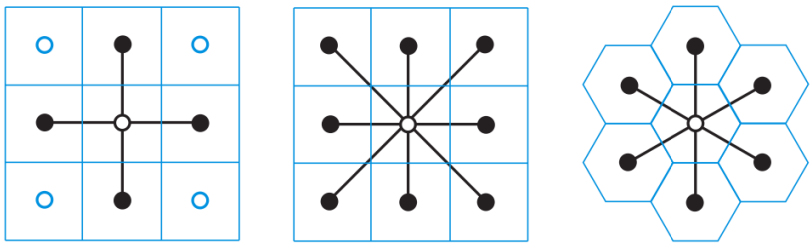
$$D_4((x,y),(h,k)) = \mid x-h \mid + \mid y-k \mid \;.$$

Vzdálenost na šachovnici (z pohledu šachového krále)

$$D_8((x,y),(h,k)) = max\{\mid x-h \mid, \mid y-k \mid\}\,.$$



Množina složená ze samotného pixelu (uprostřed, nazývaný reprezentativní pixel nebo reprezentativní bod) a jeho sousedé ve vzdálenosti 1.

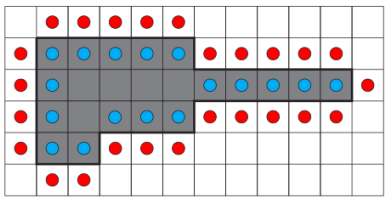


4-okolí 8-okolí 6-okolí

Oblast = souvislá množina

Hranice oblasti

- Hranice oblasti je množina pixelů oblasti majících alespoň jednoho souseda nepatřícího do oblasti.
- Spojitá obrazové funkce $\Rightarrow$ nekonečně tenká hranice.
- V digitálním obraze má hranice konečnou tloušťku. Je nutné rozlišovat vnitřní a vnější hranici.

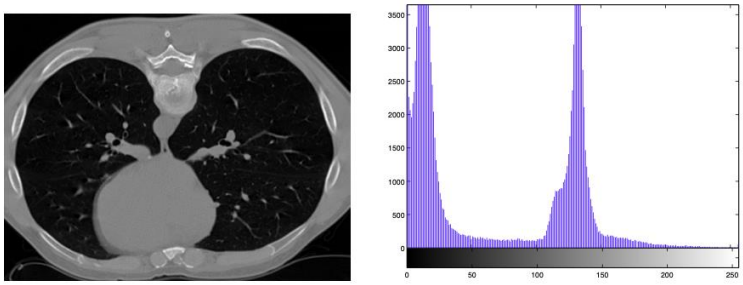


Konvexní množina = její každé dva body lze spojit úsečkou ležící uvnitř množiny.

konvexní nekonvexní

Konvexní obal, jezero, záliv.

- Histogram hodnot jasu je odhadem hustoty pravděpodobnosti jevu, že pixel bude mít určitou jasovou hodnotu.



výchozí obraz histogram hodnot jasu

- Jas obrazu charakterizuje celkovou jasnost nebo tmavost obrazu.
- Kontrast obrazu charakterizuje rozdíl (odlišení) mezi jasem/barvou objektů nebo oblastí v obraze. Například sněžná liška na sněhu má nízký kontrast a tmavý pes na sněhu má vysoký kontrast.
- Barevná sytost obrazu je podobná vlastnost jako kontrast s tím, že místo zvětšování odlišnosti objektů/oblastí v šedotónové reprezentaci obrazu, se uvažuje odlišení objektů/oblastí v barevném vyjádření.
- Ostrost obrazu je definovaná jako kontrast hran, tj. kontrast podél hran v obraze, tj. ve směru gradientu jasu obrazu. Když zvýšíme ostrost, zvýšíme kontrast jen blízko hran. V málo se měnících částech obrazu hodnotu obrazové funkce neměníme.

Reprezentace barev:

Barva popisuje vjemy vznikající souhrou tří jevů:

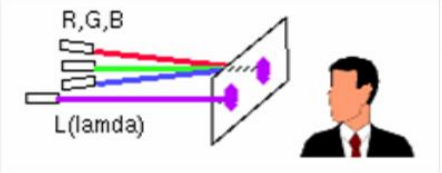
- Barva souvisí s vlastnostmi pozorovaného objektu.
- Barva souvisí se zdroji osvětlení scény a jejich vlastnostmi.
- Barva souvisí s mechanismy vnímání člověkem.

Viděná barva je způsobena vlastnostmi pigmentů (částic), které pohltí některé vlnové délky ze spektra přicházejícího světla.

- Metamerismus je obecně definován jako dva různé jevy, které jsou vnímány stejně.
- Smícháním červené a zelené vznikne žlutá (metamerismus). Žlutou lze také získat pomocí spektrální barvy, což je záření jediné vlnové délky mezi zelenou a červenou.
- Lidské vnímání barev je tedy "klamáno", že směs červené a zelené je totéž jako fyzikálně vytvořená žlutá.

Jak definovat barevný prostor?

- Tři typy čípků na sítnici vybízejí definovat barvu jako veličinu ve trojrozměrném (3D) vektorovém prostoru.
- Jak takový barevný prostor definovat?



Myšlenka experimentálního postupu:

- Posvítit světlem jedné vlnové délky λ na promítací plátno.
- Člověk nastavuje tři "potenciometry" ovlivňující intenzitu tří základních světel (tzv. funkce vyvažující barvy) R=645,2 nm; G=525,3 nm; B=444,4 nm, až se mu podaří dosáhnout stejného vjemu.
- Víme již, že je to možné právě díky barevnému metamerismu.

Barevný rozsah ve 2D, barevný trojúhelník CIE

Souřadnice x, y.

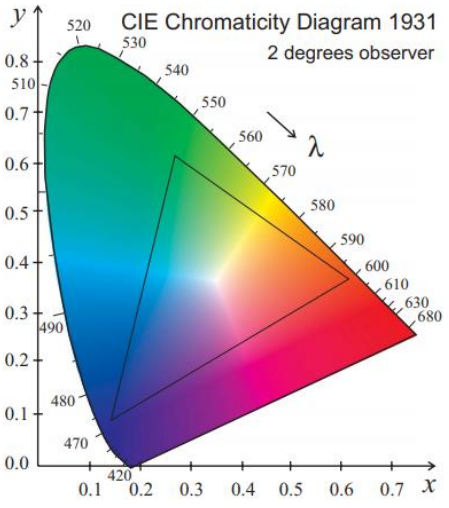
$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = 1-x-y$$

Všechny viditelné spektrální barvy jsou na okraji "podkovy", též nepřesně barevného trojúhelníku.

Všechny viditelné barvy, které lze namíchat, leží uvnitř "podkovy".

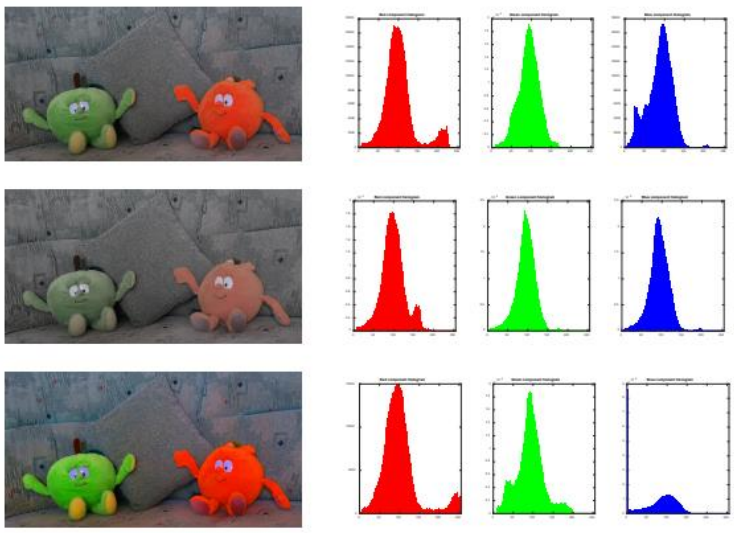


- **Odstín barvy** (angl. hue) odpovídá dominantní vlnové délce, projekci barvy na okraje barevného trojúhelníka, kde jsou spektrální barvy.
- Jména barev vlastně odpovídají odstínům. Liší se ovšem v různých kulturách.
- **Sytost barvy** (angl. saturation) popisuje, jak je barva vzdálena od neutrální šedé. Popisuje také, jak je dominantní vlnová délka (odstín) znečištěna jinými vlnovými délkami.

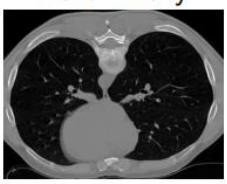
- Hodnota obrazové funkce odpovídá barvě/jasu 3D bodu

můžeme zobecnit histogram na barevné obrázky - RGB, každá složka zvlášť:

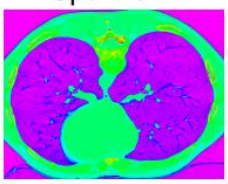
Ilustrace barevné sytosti



Beze změny



Spektrum



- Pseudobarva zobrazuje šedotónový obrázek jako barevný obrázek tak, že hodnotu intenzity převádí na barvu podle vyhledávací tabulky nebo funkce.

Geometrické a jasové transformace, interpolace

Vstupem je obraz, výstupem je obraz.

Obraz se neinterpretuje.

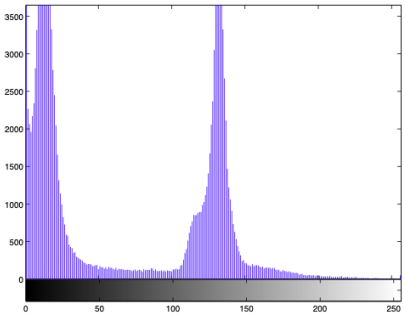
Cíl

- Zvýšení kontrastu (jen pro prohlížení obrazu člověkem).
- Potlačit zkreslení (např. korekce geometrického zkreslení díky zakřivenosti Země u družicového snímku).

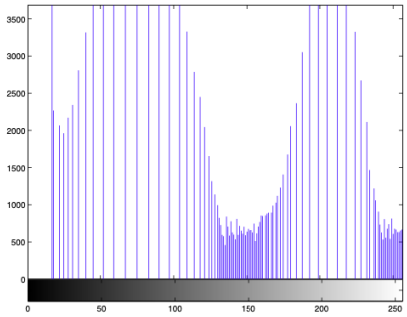
Ekvalizace histogramu

Cílem je:

- zvýšit kontrast úplným využitím jasové stupnice (pro pozorovatele – člověka),
- jasově obraz normalizovat (např. pro automatické porovnávání, vyhledávání).

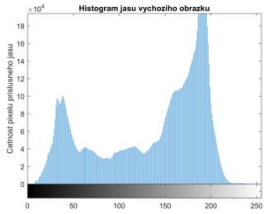


histogram původního obrazu



histogram po ekvalizaci

Příklad zvýšení kontrastu ekvalizací histogramu



výchozí obrázek

ekvalizovaný histogram

maximální kontrast (Photoshop)

Jasové korekce

- Nový jas $f(i,j)$ závisí na poloze i,j vstupního obrazu $g(i,j)$.
- Často multiplikativní model poruchy: $f(i,j) = e(i,j)\, g(i,j)$.

---

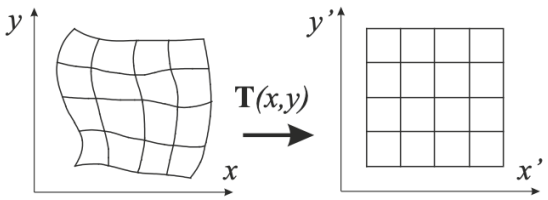
2 postupy:

1. Opravné koeficienty získány snímáním etalonové plochy známého jasu c, např. při kompenzaci nerovnoměrného osvětlení (vypnout AGC!). Po sejmutí získáme

$$f_c(i,j) = e(i,j)\, c \quad \Rightarrow \quad e(i,j) = \frac{f_c(i,j)}{c}$$

2. Proložení pozadí analytickou plochou a její odečtení od původního obrazu.

Geometrická transformace, motivace ve 2D



Rozepíšeme vektorovou transformaci $\mathbf{T}$ do dvou složek

$x' = T_x(x,y),\ y' = T_y(x,y).$

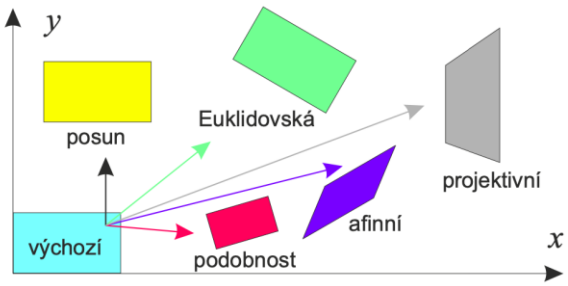
- Co musí být splněno, aby vzájemný převod byl jednoznačný?
- Geometrické transformace se se používají pro
  - zvětšování, posouvání, pootáčení, zkosení 2D obrazu (tj. podle předem známé transformace).
  - odstraňování geometrických zkreslení (příslušná geometrická transformace se často odhaduje z obrazu).

- Digitální obraz poskytuje pouze vzorky v diskrétním rastru.

- Uvažujme příklad: Obrázek jsem posunuli o polovinu pixelu ve smiru osy x. Viz obrázek vpravo dole.
- Problém: Nemáme k dispozici hodnotu obrazové funkce $f(x+0.5, y)$.
- Řešení: Interpolujeme chybějící hodnotu ze sousedních hodnot obrazové funkce.
  Proč pouze ze sousedních hodnot?

1. *Transformace souřadnic bodu* počítá se nová poloha každého bodu ve spojitých souřadnicích (reálná čísla), protože výsledek může být mimo rastr.
   Problémy:
   - Část obrazu leží mimo obraz.
   - Transformace díky nutné aproximaci nejsou invertovatelné.
2. *Aproximace jasové funkce* hledá celočíselnou hodnotu jasu v celočíselné pozici, která nejlépe odpovídá nově vypočítané neceločíselné poloze x', y'.

Základní geometrické transformace v rovině



Inspirace R. Szeliski.

Když se geometrická transformace v závislosti na pozici v obraze příliš náhle nemění, postačují aproximační polynomy nízkého stupně $m = 2$ nebo $m = 3$.

- Bilineární transformace

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_3 xy \,, \\ y' &= b_0 + b_1 x + b_2 y + b_3 xy \,. \end{aligned}$$

- Afinní transformace, která je speciálnější, zahrnuje v praxi potřebnou rotaci, translaci a zkosení.

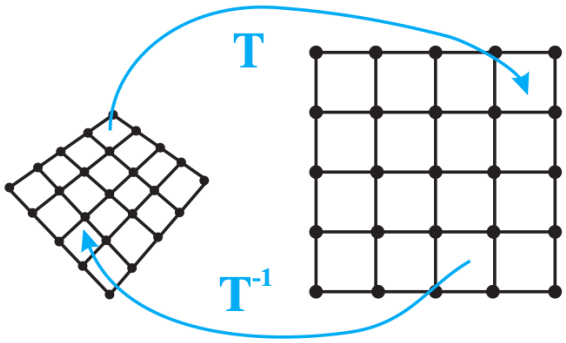
$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y \,, \\ y' &= b_0 + b_1 x + b_2 y \,. \end{aligned}$$

Afinní transformace maticově

Afinní zobrazení se po zavedení homogenních souřadnic vyjádří

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} .$$

- Vstupní obraz se zobrazí pomocí transformace $\mathbf{T}$ do výstupního obrazu.
- Duální vyjádření:
  - Dopředná transformace: $(x', y') = \mathbf{T}(x, y)$.
  - Zpětná transformace: $(x, y) = \mathbf{T}^{-1}(x', y')$.
- Obě transformace se liší kvůli nezbytnosti jasové aproximace v diskrétní mřížce.



Rotace (otočení) [editovat | editovat zdroj]

Rotací rozumíme otočení bodu kolem středu vztažné soustavy o daný úhel. Rotace je určena pouze úhlem α.

$$\mathbf{P}' = [X\cos(\alpha) - Y\sin(\alpha), X\sin(\alpha) + Y\cos(\alpha)]^T.$$

Transformační matice pro rotaci:

$$\mathbf{R} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Scaling (změna měřítka) [editovat | editovat zdroj]

Scaling je transformace změny měřítka. Je určena změnou velikosti podle souřadnicových os $[S_x, S_y]$.

$$\mathbf{P}' = [X * S_x, Y * S_y]^T.$$

Transformační matice pro změnu měřítka:

$$\mathbf{S} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Jsou-li koeficienty $[S_x, S_y]$ záporné, dochází ke „změně měřítka v opačném směru", t.j. ke středové symetrii.

Pokud je $S_x = S_y$, je možné se stejným efektem použít matici

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & S_x \end{pmatrix}$$

Translace (posunutí) [editovat | editovat zdroj]

Translace je transformace posunu. Je určena vektorem posunutí $\vec{p} = [p_x, p_y]$, který udává, kterým směrem a jak daleko bude bod posunut. T.j. $\mathbf{P}' = P + \vec{p}$.

Transformační matice pro posun:

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{pmatrix}$$

Shear [editovat | editovat zdroj]

Shear je transformace zkosení. Je určena mírou zkosení ve směrech souřadnicových os $[Z_x, Z_y]$. $\mathbf{P}' = [X + Z_x Y, Y + Z_y X]^T$.

Transformační matice pro zkosení:

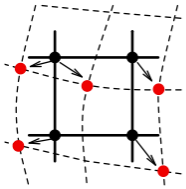
$$\mathbf{SH} = \begin{pmatrix} 1 & Z_x & 0 \\ Z_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

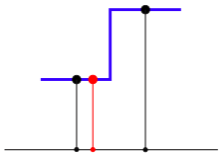
Tři používané interpolační metody

Pro interpolaci hodnot 2D obrazové funkce, např. po geometrických transformacích, se používají tři aproximační metody:

1. Nejbližší soused.
2. Bilineární interpolace. Lineární 1D interpolace se použije nejdříve v jedné a potom druhé dimenzi.
3. Bikubická interpolace.

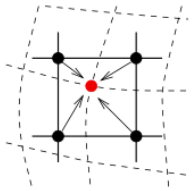
1. Nejbližší soused:

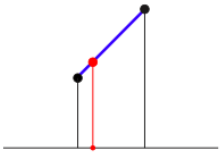




1D nejbližší soused

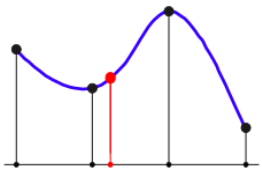
2. Bilineární interpolace:





1D lineární

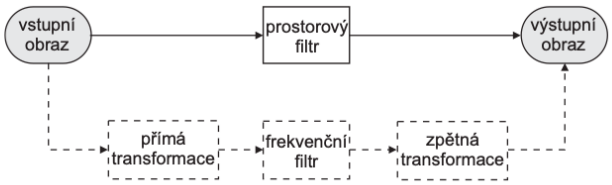
3. Bikubická interpolace:



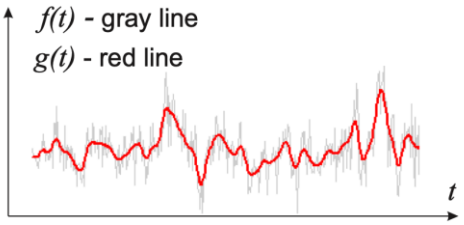
1D kubická

Filtrace a konvoluce, zpracování ve Fourierově oblasti, odstranění šumu.

- Filtrace odstraňuje nežádoucí části obrazu, např. odstraňuje šum.
- **Zdůraznění charakteristik** obrazu pro další zpracování, např. nalézání hran.



Filtrování signálu v 1D, motivační příklad

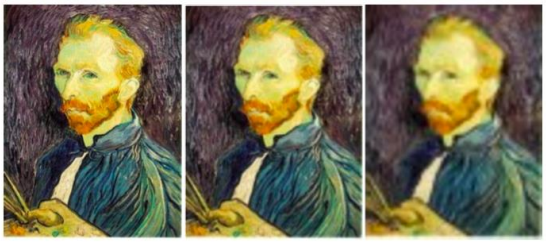


- Výstup filtru h se spočítá pomocí konvolučního integrálu

$$g(t) = f(t) * h(t) = \int_{-\infty}^{\infty} f(t)\, h(t-\tau)\, \mathrm{d}\tau = \int_{-\infty}^{\infty} f(t-\tau)\, h(t)\, \mathrm{d}\tau\,.$$

Filtrování obrázku ve 2D, motivační příklad

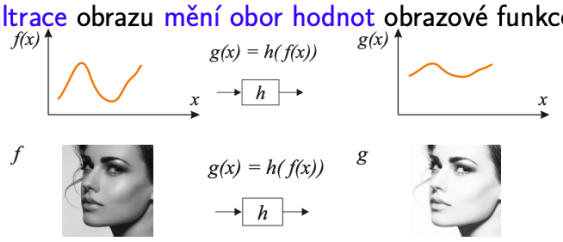
vyhlazování



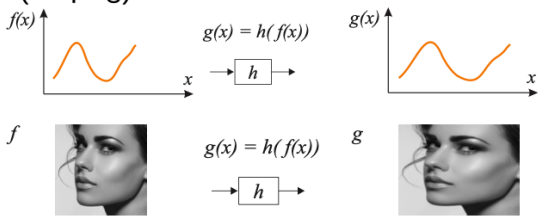
ostření

Filtrace obrazu × deformace obrazu

Filtrace obrazu mění obor hodnot obrazové funkce.



Deformace (warping) obrazu mění definiční obor obrazové funkce.



Šum v obraze

- Šumem (v obrazu) jsou náhodné změny (obrazové) funkce, které jsou většinou nežádané.
- Pro 2D obrazovou funkci $f(x, y)$ mohou být náhodné změny v hodnotě funkce f a/nebo v náhodnosti polohy vzorků (x, y). Druhý případ se uvažuje méně často.
- Šum se nejčastěji modeluje pravděpodobnostními metodami, např. pravděpodobnostním rozdělením šumu.

Šum

- Nechť $f(x, y) = I(x, y)$ je obraz; Nechť $N(x, y)$ je šum.
- Aditivní šum abstrahujeme jako $I + N$.
- Multiplikativní šum abstrahujeme jako $I(1 + N)$.

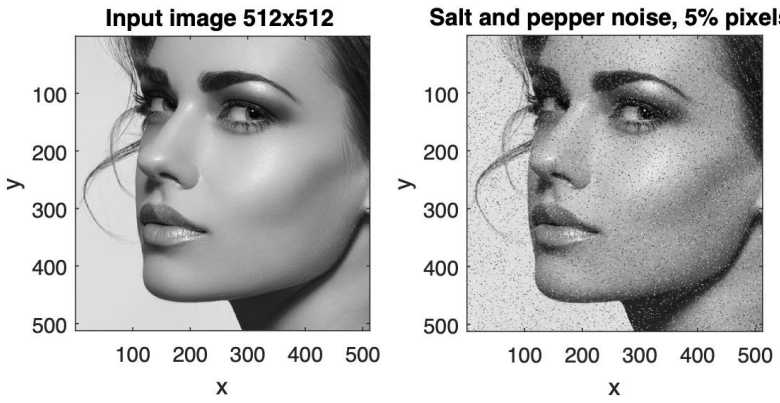
Šum nezávislý na signálu

- Šum sůl a pepř, také impulsní nebo binární šum, tvoří náhodné výskyty černých a bílých pixelů.
- Gaussovský šum představují náhodné změny v intenzitě odpovídající gaussovskému (normálnímu) rozdělení.

Šum závislý na signálu

- "Speckle" šum, také granulární nebo interferenční šum, se modeluje pomocí násobení náhodného čísla s hodnotou obrazové funkce.
- Poissonův šum předpokládá, že každý pixel v obrazu $f(x, y)$ odpovídá Poissonově rozdělení parametru $\lambda = f_0(x, y)$, kde $f_0(x, y)$ je obraz bez šumu, který chceme odhadnout. $P(f(x, y) = k) = \frac{\lambda^k}{k!} e^{-\lambda}$.

Příklad, šum sůl a pepř, 5% pixelů

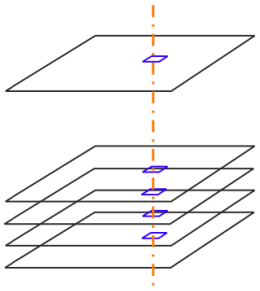


Vyhlazování z více obrazů bez rozmazání

Předpoklady: n obrazů téže neměnné scény, u nichž lze předpokládat náhodné poruchy nezávislé na signálu.

Správné hodnota jasu: $f(i,j)$ se odhaduje pro každý pixel obrazu z náhodné populace tvořené pixely v téže pozici ve všech vstupních obrazech $g_k(i,j)$ např. obyčejným průměrováním,

$$f(i,j) = \frac{1}{n} \sum_{k=1}^{n} g_k(i,j) \, .$$



n obrázků položených na sebe

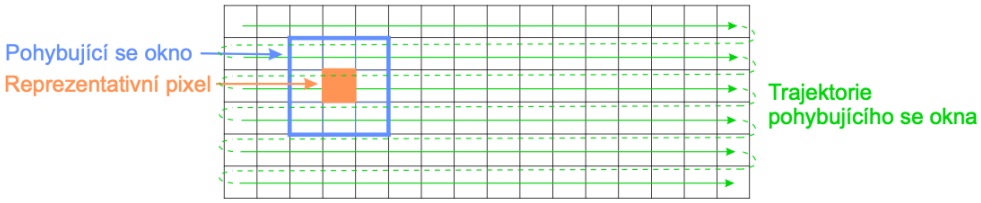
Příklad: Potlačení tepelného šumu kamery u přesných měření. Typicky se správná hodnota odhaduje přinejmenším z 50 obrazů.

Problém: potřebujeme filtrovat šum z jediného obrazu

- Nezbývá než se spolehnout na nadbytečnost údajů v obraze.
- Sousední pixely mají převážně stejnou nebo podobnou hodnotu jasu.
- Hodnotu jasu můžeme opravit na základě analýzy bodů v okolí. Použije se typický reprezentant nebo kombinace několika hodnot z okolí.
- Problém rozmazávání na jasových přechodech.

Obecná lokální filtrace

- Správná hodnota (nová výstupní hodnota) se odhaduje z hodnot obrazové funkce v malém okolí $\mathcal{O}$ nynějšího (reprezentativního) pixelu. Tímto okolím je často malý obdélník nazývaný také filtrační okno.
- Představme si, že se celý obraz systematicky prochází filtračním oknem, např. řádek po řádku shora dolů a zleva doprava.



- Výsledek analýzy obrazové funkce v okolí $\mathcal{O}$ se zapisuje do výstupního obrazu do stejného místa jako jsou souřadnice reprezentativního pixelu ve vstupním obrazu. Metoda se (někdy) nazývá filtrace pohybujícím se oknem.
- Obecně mohou být vlastnosti filtru v každém pixelu obrazu jiné.

Lokální lineární předzpracování

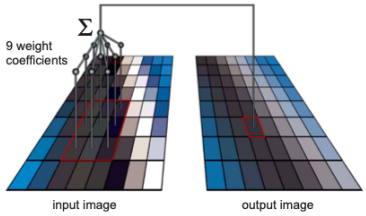
- Nová hodnota je vypočítána jako lineární kombinace (konvoluce) hodnot obrazové funkce v okolí.
- Připomínka: Linearita operátoru, tj. dvě vlastnosti: aditivita a homogenita.
- U skutečných obrazů je předpoklad linearity narušen.
  - Hodnotu pixelu nelze vynásobit libovolným skalárem nebo výsledek součtu nemůže být jakýkoliv. Problém saturace díky omezenému rozsahu hodnot obrazové funkce, typicky interval $\langle 0, 255 \rangle$.
  - Problémy na okraji obrazu. Maska přesahuje.

Diskrétní 2D konvoluce

- Příspěvek jednotlivých pixelů v okolí $\mathcal{O}$ je vážen v lineární kombinaci koeficienty h podle

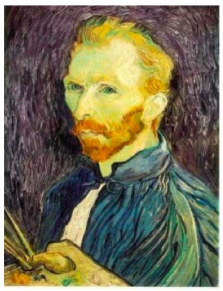
$$g(x,y) = \sum_{(m,n)} \sum_{\in \mathcal{O}} h(x-m,\, y-n)\; f(m,n)\;.$$

- Konvoluční jádro h, též konvoluční maska, udávající váhu hodnot obrazové funkce v okolí v součtu.
- Často se používá obdélníkové okolí $\mathcal{O}$ s lichým počtem řádků a sloupců, a tak může reprezentativní bod ležet uprostřed konvoluční masky.



Příklad lineární filtrace v prostoru obrazu

$=$



$*$ $\quad h(x,y)$

filtrovaný obraz g | výchozí obraz f | konvoluční jádro h též funkce filtru, zde Gaussián $h(x,y) = e^{-\frac{x^2+y^2}{\sigma^2}}$

Obyčejné průměrování

Průměrování v okolí 3×3

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$


Průměrování 3×3


Průměrování 7×7


Průměrování 11×11


Průměrování 25×25

Dvě varianty zvýraznění pixelů blíže středu masky

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Separabilní filtry

- Filter je separabilní, když $h(x,y) = h_1(x) \cdot h_2(y) = \delta * h_1(x,\cdot) * h_2(\cdot,y)$, kde δ je Diracova funkce.
- Separablilní filtry se dají implementovat postupnými 1D konvolucemi díky asociativitě konvoluce. Nechť jsou M, N rozměry obrázku a m, n rozměry konvoluční masky h. Separabilita sníží výpočetní složitost z $\mathcal{O}(MNmn)$ na $\mathcal{O}(MN(m+n))$.
- Příklad: binomický 2D filtr rozměru 5×5 (prvek filtru je součtem dvou předchozích čísel v Pascalově trojúhelníku).

$$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} \\ h_1 \\ \\ \end{bmatrix} \begin{bmatrix} \quad h_2 \quad \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

Separabilita $\Rightarrow$ úspora výpočtů 2

- Pro náš 5×5 filtr potřebuje přímý výpočet 25 násobení a 24 sčítání pro každý pixel.
- Při použití separabilního filtru stačí 10 součinů a 8 součtů.

Filtrace mediánem:

- Výpočet mediánu je pro diskrétní obrazovou funkci jednoduchý. Stačí uspořádat vzestupně hodnoty jasu v lokálním okolí a medián určit jako prvek, který je uprostřed této posloupnosti.
- Aby se snadno určil prostřední prvek, používají se posloupnosti s lichým počtem prvků, typicky okolí 3×3, 5×5, atd.
- Výpočet ještě urychlí skutečnost, že k nalezení mediánu stačí částečné uspořádání posloupnosti.

Medián, příklad

100	98	102
99	105	101
95	100	255

- Oranžové podbarvení označuje reprezentativní pixel filtrační masky (použitého okolí).
- Aritmetický průměr = 117,2
- Medián: 95 98 99 100 100 101 102 105 255
- Robustní, protože snese až 50 % vychýlených hodnot.

Filtrace mediánem, šum sůl a pepř
5 % pixelů zašuměno

Aditivní šum sůl a pepř

Průměrování 3×3

Průměrování 7×7

Originál 512×512

Mediánová filtrace 3×3

Mediánová filtrace 7×7

1D Fourierova transformace, úvod

- Fourierova transformace je základním nástrojem pro (lineární) zpracování signálů a v teorii řízení.
- Dovoluje vzájemně jednoznačný převod signálů z/do časové reprezentace $f(t)$ do/z frekvenční reprezentace $F(\xi)$.
- Umožňuje analyzovat frekvenční obsah (spektrum) signálu.
- FT je vhodná pro periodické signály.
- Když signál není periodický, potom lze použít krátkodobou FT nebo lineární integrální transformaci s bázovými funkcemi lokalizovanými v čase nebo 2D prostoru. Příkladem je vlnková transformace (wavelets), Gaborovy filtry.

Fourierova Tx definice: spojitý případ

$\mathcal{F}\{f(t)\} = F(\xi)$, kde ξ [Hz=s^{-1}] je frekvence a $2\pi\xi$ [s^{-1}] je úhlová frekvence.

Fourierova Tx	Inverzní Fourierova Tx
$F(\xi) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i \xi t}\, \mathrm{d}t$	$f(t) = \int_{-\infty}^{\infty} F(\xi)\, e^{2\pi i \xi t}\, \mathrm{d}\xi$

Pro digitální obrazy Fourierova transformace vždy existuje, protože digitální obrazy jsou omezené a mají konečný počet nespojitostí.

Konvoluce $*$ definovaná pro 1D signály používá překlopené jádro h

$$(f * h)(t) \equiv \int_{-\infty}^{\infty} f(\tau)\, h(t-\tau)\, \mathrm{d}\tau\,.$$

- Všechny dvojice (časový signál $\leftrightarrow$ Fourierův obraz) jsou vázány principem nejistoty.
- Signál o krátké době trvání má široké frekvenční spektrum a obráceně.
- (trvání signálu) $\cdot$ (šířka spektra) $\geq \frac{1}{\pi}$

Neperiodické signály

Fourierova transformace předpokládá periodický signál. A co když potřebujeme zpracovávat neperiodický signál? Dva obvyklé přístupy.

1. Zpracovat signál po malých částech (oknech) a předpokládat, že vně je signál periodický.
   - Přístup zavedl Dennis Gabor v roce 1946 a nazývá se krátkodobá Fourierova transformace (Short time Fourier transform).

     Dennis Gabor, 1900-1979, vynálezce holografie, Nobelova cena za fyziku 1971, studoval v Budapešti, PhD v Berlíně 1927, odešel před nacisty do Británie v 1933.
   - Pouhé rozsekání signálu na obdélníková okna není dobré, protože na rozhraní oken jsou nespojitosti. Ty se ve spektru projeví nežádoucími vysokými frekvencemi.
   - Proto se signál obvykle konvoluje s tlumící váhovou funkcí, obvykle Gaussián nebo Hammingova funkce, zajišťující nulovou hodnotu signálu na okraji a vně okna.
2. Použití složitějších bázových funkcí, např. vlnek ve vlnkové (wavelets) transformaci.

- Diskrétní Fourierova transformace

$$F(k) \equiv \sum_{n=0}^{N-1} f(n)\, e^{-\frac{2\pi i}{N} kn}$$

Spektrum $F(k)$ je periodicky prodloužené s periodou N.

- Inverzní diskrétní Fourierova transformace

$$f(n) \equiv \frac{1}{N} \sum_{k=0}^{N-1} F(k)\, e^{\frac{2\pi i}{N} kn}$$

- Výpočetní složitost je $\mathcal{O}(N^2)$.

- Rychlá Fourierova transformace (FFT – fast Fourier transform) je efektivní algoritmus pro výpočet diskrétní Fourierovy transformace a její inverze.
- Tvrzení: FFT má výpočetní složitost $\mathcal{O}(N \log_2 N)$.

2D Fourierka:

Myšlenka. Obrazová funkce $f(x, y)$ se rozloží na lineární kombinaci harmonických (sínusovek, kosínusovek, obecněji ortonormálních) funkcí.

Definice přímé transformace. u, v jsou prostorové frekvence.

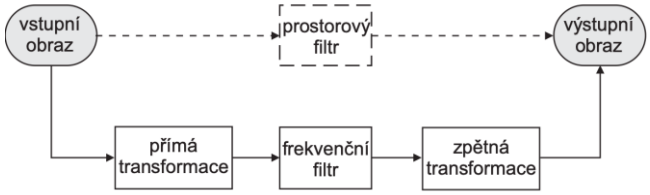
$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i(xu+yv)}\,\mathrm{d}x\,\mathrm{d}y$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, e^{2\pi i(xu+yv)}\,\mathrm{d}u\,\mathrm{d}v$$

Diskrétní:

$$F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\,\exp\left[-2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)\right],$$

$$u = 0, 1, \ldots, M-1\,, \qquad v = 0, 1, \ldots, N-1\,,$$



Konvoluce ve frekvenčním spektru

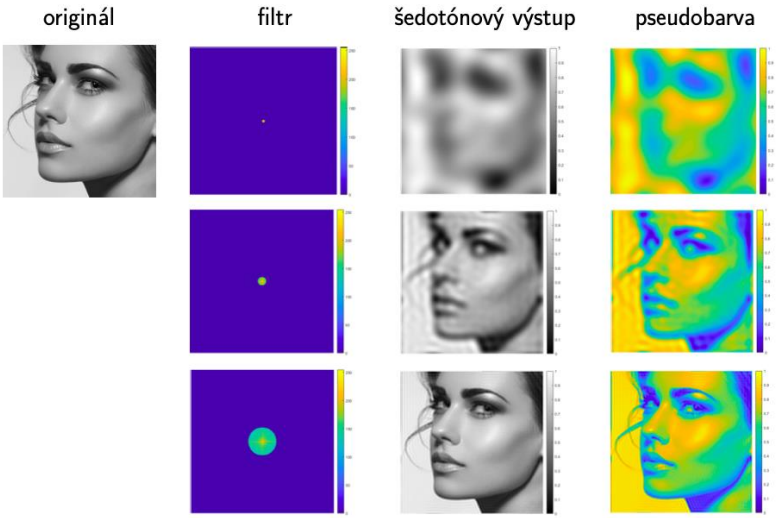
- Násobení v matici prvek po prvku.
- (Vysoká) rychlost je dána využitím rychlé FFT.

Mějme matici a rozměru $M \times N$ a matici b rozměru $P \times Q$.

Konvoluci $c = a * b$ lze spočítat postupem:

1. Doplň matice a, b nulami, aby měly rozměr alespoň $M + P - 1$, $N + Q - 1$ (obvykle až do velikosti mocniny dvou kvůli FFT).
2. Vypočítej 2D FFT matic a, b (v MATLABu pomocí `fft2`). Výsledkem jsou matice A, B.
3. Vynásob komplexní Fourierova spektra $C = A .* B$, součin prvek po prvku.
4. Výsledek konvoluce c získej inverzní Fourierovu transformací (v MATLABu pomocí `ifft2`).

Dolnopropustní filtrace, ořezání v ostrém kruhu, r=5, 15, 50



Příklad, fourierovská filtrace, Abs(Sobelův filtr)