

ROZ

ROZPOZNÁVÁNÍ A ZPRACOVÁNÍ OBRAZU

Martin Petřek

5. února 2004

Obsah

1	Základní pojmy	2
1.1	Matematické transformace	3
1.2	Digitalizace	7
1.3	Kvantování	8
2	Předzpracování	9
2.1	Operace s Histogramem	9
2.2	Image enhancement methods	9
2.3	Odstranění šumu	11
2.3.1	Bílý šum	11
2.3.2	Šum typu sůl a pepř (salt & pepper)	13
2.3.3	Periodické poškození obrázku	13
2.3.4	Kvantifikace šumu	14
2.3.5	Aproximace	14
2.4	Detekce hran	15
2.4.1	Detekce rohů	18
2.5	Restaurace obrazu	20
2.5.1	Wienerův filtr	21
2.5.2	Problém slepé dekonvoluce	23
2.6	Geometrické zkreslení	24
2.7	Výběr řídicích bodů (control point selection)	25
2.8	Mapovací funkce	27
2.8.1	K metodě nejmenších čtverců	28
2.8.2	Vlastní mapování	29
3	Rozpoznávání dat (pattern recognition)	29
3.1	Statistické klasifikace	33
3.2	Klasifikace bez učení (shluková analýza)	36
3.3	Redukce dimenzionality (příznakového prostoru)	39
3.4	Suboptimální metody	41

4	Rozpoznávání v obraze	42
4.1	Segmentace	42
4.2	Matematická morfologie	45
4.3	Intuitivní příznaky	46
4.4	Transformační příznaky	47
4.5	Momentové invarianty	48
4.5.1	Změna měřítka	49
4.5.2	Rotace soustavy	50
4.6	Problém rekonstrukce	52
4.6.1	Invariant vůči konvoluci	52

1 Základní pojmy

Mějme danu spojitou funkci $f : L_1^2 \rightarrow \mathbf{R}$, tedy $f(x, y)$ integrabilní. Nazveme ji *spojitým obrazem*. Podobně *diskrétní obraz* definujeme jako posloupnost $F_{ij} \in \hat{n} \cup \{0\}$ pro $i, j \in \mathbf{Z}, n \in \mathbf{N}$. Libovolné zobrazení $\Phi : f(x, y) \rightarrow F_{ij}$ nazýváme *digitalizací*.

Digitalizace Digitalizaci rozlišujeme podle toho, zda diskretizaci provedeme v prostoru souřadnic či v oboru hodnot. Pak hovoříme o

- *vzorkování (prostor souřadnic)*
- *kvantování (obor hodnot)*

Ukazuje se, že vzorkování lze za určitých předpokladů provést tak, aby se žádná informace neztratila.

Předzpracování – sem patří následující operace:

- potlačení šumu
- zvýraznění hran
- doostření
- změna kontrastu

Analýza – je v podstatě získávání další dodatečné (neobrazové) informace z obrazu. Příkladem je třeba určení hranic, rozpoznávání objektů, atd.

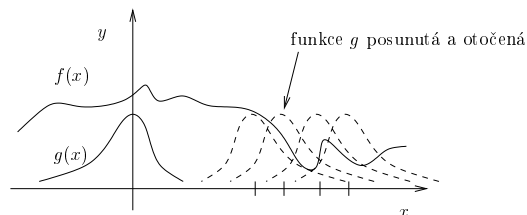
1.1 Matematické transformace

Konvoluce Definujeme *konvoluci* dvou funkcí f, g jako zobrazení $A : L_1 \times L_1 \rightarrow L_2$

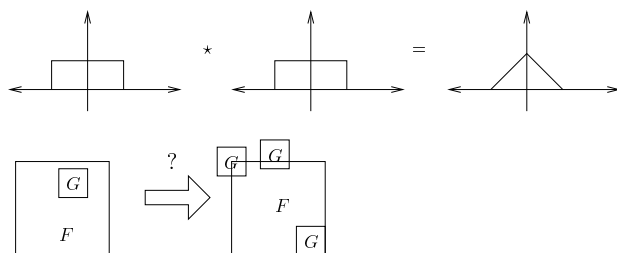
$$(f * g)(x) = \int f(t)g(x - t)dt \quad (1)$$

V podstatě se tedy jedná o průměrování funkce f jinou funkcí g , protože g je obvykle symetrická funkce s malým supportem (viz. obr. 1). Je-li např.

g obdelníkový puls, pak se jedná o klasické průměrování. Dost často se taky požaduje, aby výsledná funkce měla stejně omezený obor hodnot jako původní f , takže se volí $\int g = 1$.



Obrázek 1: K definici konvoluce v 1D.



Obrázek 2: Ukázka konvoluce dvou obdelníkových pulsů. A problém počítání prvku na okraji při diskrétní konvoluci ve 2D.

V diskrétním případě je F i G matice. Otočenou matici G posouváme po F a do každého bodu výsledné matice napíšeme součet součinů překrývajících se prvků. Na krajích matice F a v rozích je pak třeba zvolit postup, jak se chovat. Pokud zůstaneme s G uvnitř F , výsledná matice bude menší než původní. Většinou se dělá stejně velká matice jako F , ale stejně tak ji lze i rozšiřovat (viz. obr. 2).

Korelace Korelaci dvou funkcí definujeme podobně jako konvoluci s tím rozdílem, že funkce g se „neotáčí“. Je to tedy opět zobrazení $A : \mathbf{L}_1 \times \mathbf{L}_1 \rightarrow L_2$

$$(f \circledast g)(x) = \int f(t)g(t-x)dt \quad (2)$$

Spojité Fourierova transformace (CFT) Necht $f \in L_1$, pak *Spojité Fourierova transformace*¹ je definována vztahem:

$$F(u) = \int f(x)e^{-2\pi i u x} dx \quad (3)$$

Existence komplexní funkce F plyne z předpokladu, není ovšem zaručeno, že i $F(u)$ bude z L_1 . V podstatě se jedná o vyjádření původní funkce jako lineární kombinace sin a cos funkcí.

¹Fourierova transformace a Spojité Fourierova transformace se nepatrně liší.
 $FT(f(x)) = \int f(x)e^{-iux} dx, \quad IFT(F(u)) = \frac{1}{2\pi} \int F(u)e^{iux} du$

V diskrétním prostoru, nechť máme ortonormální bázi funkcí

$$\mathcal{B} = \{\sin mx, \cos nx, 1\} \quad (4)$$

na konečném dvojrozměrném intervalu. Funkci f jsme pak schopni vyjádřit ve tvaru $f = \sum c_k \varphi_k$, kde c_k jsou tzv. *Fourierovy koeficienty*. To, že jsme se omezili pouze na celočíselné násobky frekvencí má za následek fakt, že jsme schopni v celém oboru popsat jen periodické funkce. V obecném pojetí lze libovolnou funkci složit z bazických sin a cos funkcí, ovšem jejich frekvence musí být spojitě se měnící.

Spojité Fourierova transformace ve 2D (CFT2)

$$F(u, v) = \int f(x, y) e^{-2\pi i(ux+vy)} dx dy \quad (5)$$

Inverzní spojitá Fourierova transformace (ICFT)

$$f(x) = \int F(u) e^{2\pi i u x} du \quad (6)$$

Inverzní spojitá Fourierova transformace 2D (IFT2)

$$f(x, y) = \int F(u, v) e^{2\pi i(ux+vy)} du dv \quad (7)$$

Ze vztahů a z vlastností integrálu je zřejmé, že

$$f_1 + f_2 \xrightarrow{CFT} F_1 + F_2 \quad (8)$$

$$\alpha f \xrightarrow{CFT} \alpha F \quad (9)$$

Diracův puls (δ -funkce) V signálech se definuje speciální bodová funkce δ , která je všude nulová, jen v nule má takovou hodnotu, že $\int \delta(x) = 1$.

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ \cdot & x = 0 \end{cases} \quad (10)$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (11)$$

Platí tyto vztahy, jejichž důkaz přenecháme matematické analýze.

$$f * \delta = f \quad \forall f \quad (12)$$

$$FT(\delta) = 1 \quad (13)$$

$$FT(1) = \delta \quad (14)$$

Konvoluční teorém Vyjadřuje skutečnost, že konvoluci lze ve frekvenční oblasti převést na násobení a naopak.

$$FT(f * g) = F \cdot G \quad (15)$$

$$FT(f \cdot g) = F * G \quad (16)$$

Zde se symbolem \cdot nemyslí maticové násobení, ale násobení stejně rozměrných matic po prvcích. Pokud chceme ověřit konvoluční teorém v praxi, je potřeba diskretní data periodicky prodlužovat!

Poznamenejme, že při FT2, tedy dvojdimenzionální Fourierově transformaci, dostáváme komplexní data (*Fourierův obraz*). Protože jsme ovšem původně měli reálný signál, bude fourierův obraz symetrický podle středu. Co se týká náročnosti, má výpočet FT (1D) v základním tvaru složitost $O(n^2)$. V roce 1965 Cooley a Turkey ukázali urychlení výpočtu za předpokladu $n = 2^k$. Tato rychlá Fourierova transformace (FFT) se používá dodnes, lze ji sestavit pro obecné obdélníkové matice a její složitost je $O(n \log n)$. Naproti tomu konvoluce má složitost $O(n^2)$ pro každý bod výsledné matice, tedy je jasné, že při počítání konvoluce s rozměrnějšími maticemi je výhodnější použít konvolučního teorému a ve frekvenční oblasti pouze vynásobit matice. Například už pro obrázek 512×512 a konvoluční matici 11×11 je použití konvolučního teorému výhodnější.

Nyní ukažme jak vypadá Fourierův obraz *obdélníkového pulsu*. Nechť f je funkce definovaná takto:

$$f(x) = \begin{cases} c & \forall x \in \langle -\lambda, \lambda \rangle, c \in \mathbf{R} \\ 0 & \text{jinde} \end{cases} \quad (17)$$

pak

$$\begin{aligned} F(u) &= \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx = c \int_{-\lambda}^{\lambda} e^{-2\pi i u x} dx = \\ &= \frac{c}{-2\pi i u} [e^{-2\pi i u x}]_{-\lambda}^{\lambda} = c \frac{e^{-2\pi i u \lambda} - e^{2\pi i u \lambda}}{-2\pi i u} = \\ &= \frac{c}{\pi u} \sin(2\pi \lambda u) \end{aligned} \quad (18)$$

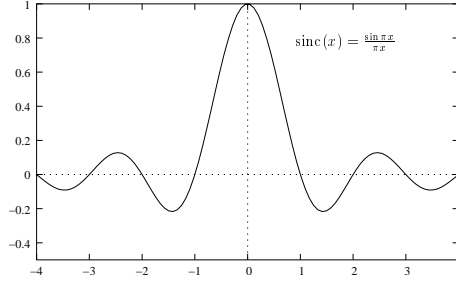
Pro $c = 2$ a $\lambda = \frac{1}{2}$ bude

$$F(u) = \frac{\sin(\pi u)}{\pi u} = \text{sinc}(u) \quad (19)$$

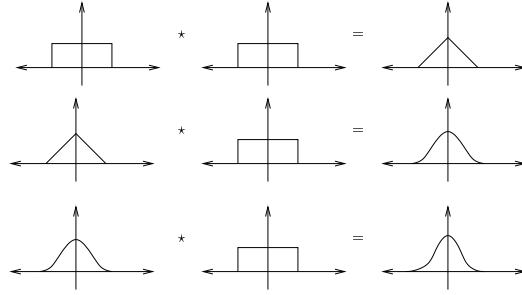
Pozn. Ve 2D vzniká funkce $\text{sinc}(x)\text{sinc}(y)$. Tato funkce není středově symetrická, jak by se mohlo zdát.

Při násobné konvoluci obdélníkových pulsů vznikají křivky, které se označují jako *B-spline*. Mimo jiné mají tu vlastnost, že aproximují Gaussovu funkci normálního provděpodobnostního rozložení.

Fourier Shift Theorem (FST) Máme-li funkci g posunutou vzhledem k f o konstantu a , dává Fourierova transformace vztah:



Obrázek 3: Fourierova transformace obdélníkového pulsu a vzniklá funkce $\text{sinc}(u)$.



Obrázek 4: Opakováním konvoluce obdélníkových pulsů vznikají B-spline křivky, které stále lépe aproximují Gaussovu rozdělovací funkci.

$$f(x-a) \xrightarrow{CFT} e^{-2\pi i a u} F(u) \quad (20)$$

Protože

$$\begin{aligned} f(x-a) &= g(x) \xrightarrow{CFT} \int g(x) e^{-2\pi i u(x-a+a)} dx = \\ &= \int f(x-a) e^{-2\pi i u(x-a)-2\pi i u a} dx = \\ &= e^{-2\pi i u a} \int f(x-a) e^{-2\pi i u(x-a)} dx = \left\{ \text{subst. } y = x-a \right\} = \\ &= e^{-2\pi i u a} \int f(y) e^{-2\pi i u y} dy = e^{-2\pi i u a} F(u) \end{aligned} \quad (21)$$

Tedy obraz je akorát násobený komplexní jednotkou $e^{-2\pi i a u}$, jeho amplituda má tedy stejnou velikost jako originál. Toho lze využít pro rozpoznávání, fázi lze zjistit dodatečně. Lze také ukázat, že nejpodstatnější část obrazové informace je obsažena ve fázi, amplituda nemá tak zásadní vliv na vzhled obrazu.

1.2 Digitalizace

$$f(x) \rightarrow \{\text{vzorkování}\} \rightarrow d(x) \rightarrow \{\text{kvantování}\} \rightarrow F \quad F \in \langle 0, 255 \rangle \quad (22)$$

Při vzorkování (*sampling*) vzniká otázka, jak hustě vzorkovat, abychom neztratili obrazové informace, eventuálně, zda existuje vzorkovací frekvence, která již zachytí všechny tyto informace. Shannonův teorém dává za určitých předpokladů kladnou odpověď na předchozí otázku.

Vzorkování lze považovat a za násobení původního spojitého signálu nekonečným polem δ -funkcí.

$$s(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x - i\Delta x, y - j\Delta y) \quad (23)$$

$$d(x, y) = f(x, y) \cdot s(x, y) \quad (24)$$

Lze ukázat, že Fourierův obraz nekonečného pole δ -funkcí bude opět nekonečné pole δ -funkcí s krokem $\frac{1}{\Delta x}, \frac{1}{\Delta y}$. Ve frekvenční oblasti pak dostáváme

$$S(U, V) = \frac{1}{\Delta x} \frac{1}{\Delta y} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta\left(\frac{u-i}{\Delta x}, \frac{v-j}{\Delta y}\right) \quad (25)$$

$$D(u, v) = F(u, v) * S(u, v) \quad (26)$$

Tedy zvyšováním vzorkovací frekvence (tj. zmenšováním Δx a Δy) současně zvyšujeme vzdálenost mezi Fourierovými obrazy dvou vzorků, protože funkce $D(u, v)$ ve vztahu (26) není nic jiného než opakováním funkce $F(u, v)$ ve vzálenosti Fourierových pulsů vzorkovací funkce. Pokud budeme chtít funkci zrekonstruovat z jejích Fourierových vzorků je nezbytně nutné, aby tyto vzorky nebyly poškozeny vzájemným překrytím. To je možné za dvou předpokladů

- Pokud je funkce frekvenčně omezená. U frekvenčně neomezených funkcí, kde neexistuje $f_{\max} < \infty$, je i jejich frekvenční spektrum neomezené, proto se vzorky budou překrývat vždy.
- Pokud je vzorkovací frekvence větší nebo rovna dvojnásobku maximální frekvence f_{\max} (tzv. Nyquistův limit)

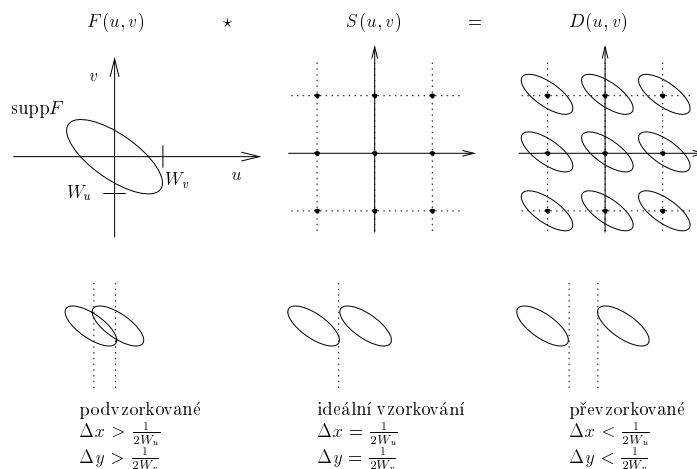
Což je zmíněný *Shannonův vzorkovací teorém*. Ideální rekonstrukční filtr má tedy ve Fourierově oblasti tvar obyčejné schodovité funkce a její spojitý vzor v časové oblasti má tvar již zmíněné funkce sinc.

$$\text{sinc}(x, y) = \frac{\sin(\pi x) \sin(\pi y)}{\pi x \pi y} \quad (27)$$

Pro frekvenčně neomezené funkce by byl teoreticky potřeba nekonečně malý krok. Vzorkováním tak vzniká velmi nepříjemný jev zvaný *Aliasing*.

Aliasing vzniká ve dvou případech

1. Pokud je původní funkce frekvenčně neomezená, tj. neexistuje žádná maximální frekvence a funkci tudíž nelze v diskrétním rastru reprezentovat přesně.
2. Funkce je sice frekvenčně omezená, tj. existuje $f_{\max} < \infty$, ale funkci vzorkujeme s frekvencí menší nežli $2f_{\max}$, tedy pod Nyquistovým limitem.



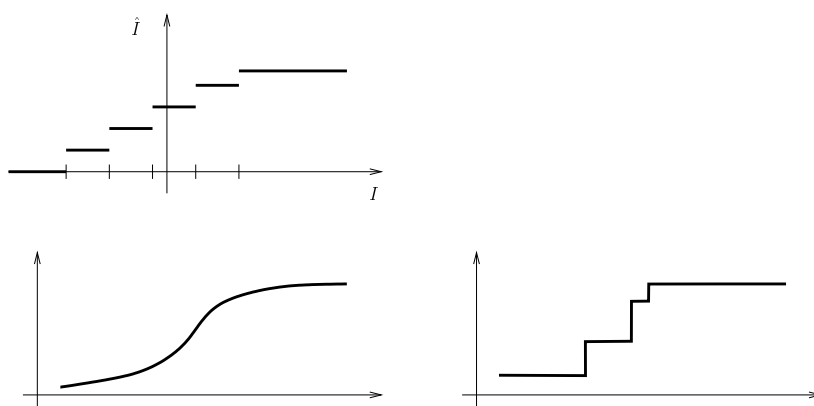
Obrázek 5: Vzorkování ve frekvenční oblasti a překrývání obrazů frekvenčně omezené funkce

1.3 Kvantování

Kvantování je zobrazení z \mathbf{R} do množiny $K = \{0, 1, \dots, l-1\}$. Počet intenzit l se většinou volí 256. Z povahy množin \mathbf{R} a K je jasné, že se jedná o ztrátovou operaci a není jednoznačný inverzní proces. Hodnoty intenzit $i \in K$ označujeme při kvantování jako tzv. *Kvantovací prahy*.

Kvantovací prahy můžeme volit několika způsoby. První a poslední je většinou určen snímacím zařízením. Dále se používají

- *ekvidistantní* – prahy jsou od sebe stejně vzdáleny, velmi často se používá
- *logaritmické (exponenciální)* – využívá se u ztmavených obrázků
- další volba je třeba taková, aby každá barva byla zastoupena přibližně stejným počtem pixelů



Obrázek 6: Kvantovací prahy a vznik falešných kvantizačních hran

Nedostatečným počtem úrovní intenzit vznikají falešné kvantizační hrany (false contour effect). Lidské oko se ovšem podle hran orientuje, proto se přidává aditivní šum, který tento nepříjemný efekt rozmaže (samozřejmě pouze tehdy, pokud nelze zvýšit počet prahů).

Máme-li omezenou paměť je třeba volit vhodně počet úrovní a prostorové rozlišení. Má-li obraz málo detailů, je lepší volit více hodnot pro kvantování s menším rozlišením, pro hodně detailů a hran upřednostnit větší rozlišení a méně úrovní pro kvantování.

Charakteristiky lidského oka Experimentálně byly zjištěny tyto charakteristiky zdravého lidského oka.

- prostorová rozlišovací schopnost 0,1 mm ze vzdálenosti 25 cm
- rozlišení šedi (odděleně) 40 úrovní
- rozlišení šedi (porovnání vedle sebe) 100 úrovní

Dále podle průzkumu, bylo vypořizováno, že standartní obrázek velikosti 512×512 se 128 stupni šedi zobrazený na plochu 5×5 cm pozorovaný ze vzdálenosti 25 cm se jeví jako **spojitý**.

2 Předzpracování

2.1 Operace s Histogramem

Histogram obrazu je funkce četností jednotlivých intenzit. Lze jej přirovnat k hustotě pravděpodobnosti

Kumulativní histogram je pak funkce intenzit, kde pro každou intenzitu je funkční hodnota rovna počtu bodů mající svou intenzitu menší nebo rovnou. Je to tedy obdoba distribuční funkce.

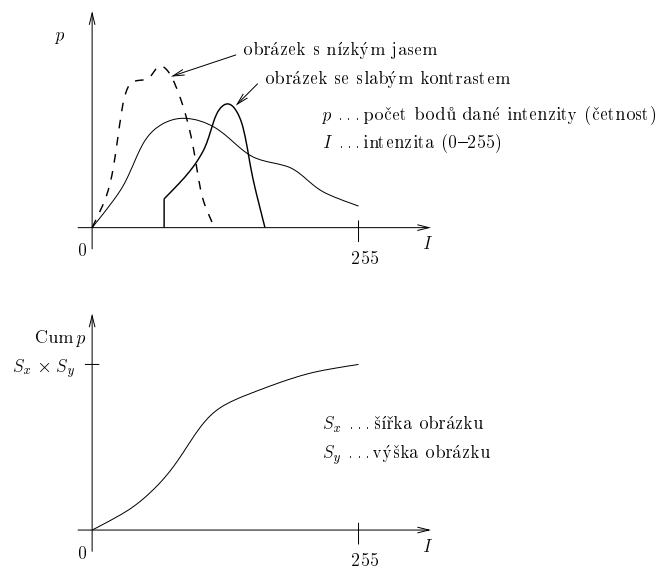
Předzpracování obrazu Patří sem

1. Metody pro vylepšení vzhledu (Image enhancement methods) – změna jasu a kontrastu, potlačení šumu, zvýraznění hran
2. Obnovení obrazu (Image restoration) – snažíme se rozpoznat a invertovat vzniklé poškození

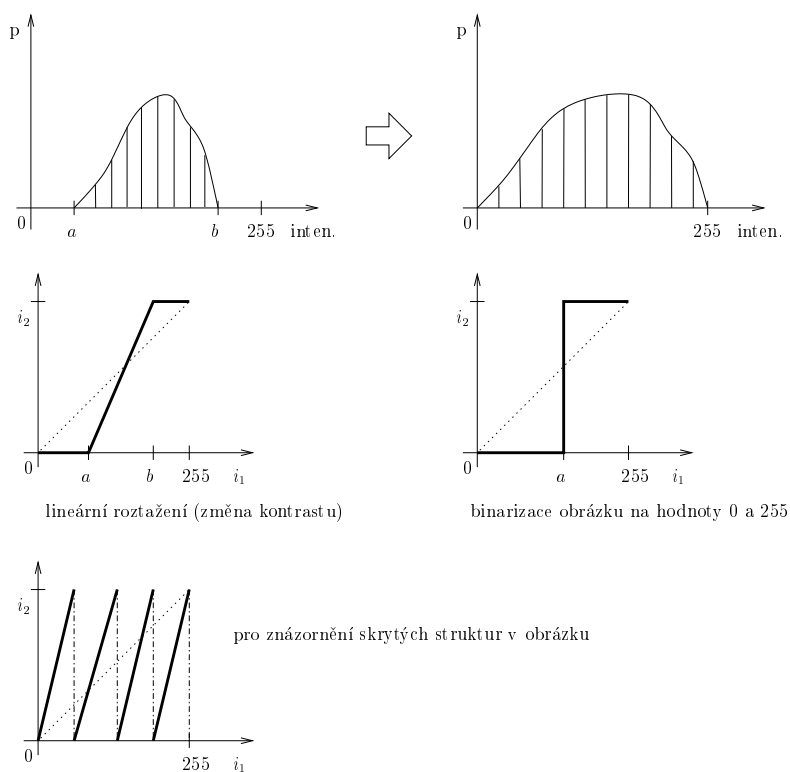
2.2 Image enhancement methods

Jas a kontrast Změna kontrastu a jasu se provádí změnou hodnot v histogramu. Realizuje se transformační funkcí $\psi : K \rightarrow K$. Hodnota výsledků závisí pouze na hodnotě v jednom bodě, jedná se tedy o *bodové operace*.

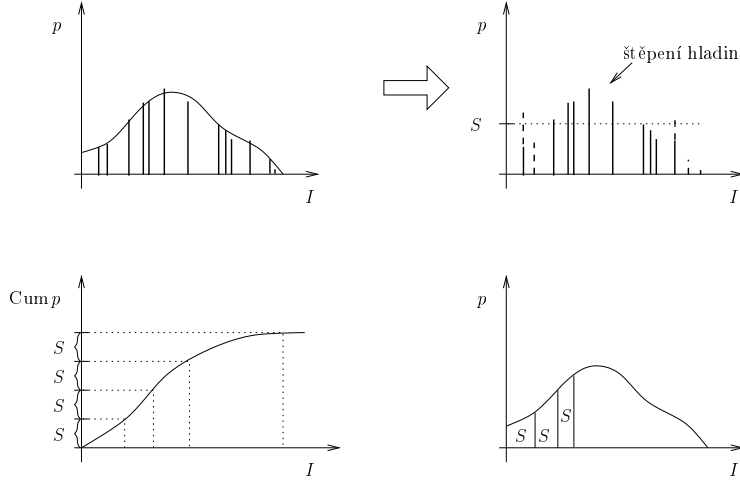
Ekvalizace histogramu je operace, kdy chceme převést histogram na požadovaný tvar (např. konstatní). K transformaci používáme distribuční funkci (kumulativní histogram). Dojde-li k tomu, že následující hodnota v histogramu pro danou intenzitu převyšuje požadovanou hodnotu, je potřeba provést štěpení (tj. nějakým způsobem ji rozdělit do více intenzit). Někdy se ekvalizuje i lokálně (změna jasu, kontrastu), postup probíhá v okně podobně jako při konvoluci.



Obrázek 7: Histogram a kumulativní histogram



Obrázek 8: Změna kontrastu v histogramu a znázornění transformační funkce (nahore a vlevo dole). Vpravo dole pak transformační funkce pro provedení binarizace obrázku.



Obrázek 9: Ekvalizace histogramu – převedení na konstantní histogram.

2.3 Odstranění šumu

Aditivní model šumu Předpokládejme tzv. *aditivní model šumu*², neboli na signálu nezávislý šum. Modelujeme jej pomocí funkce n .

$$f = f_{\text{org}} + n \quad (28)$$

V našem diskrétním pojetí budeme uvažovat funkci n jako náhodnou veličinu (NV), která měla $h \cdot w$ realizací ($h \dots$ výška, $w \dots$ šířka) nebo jako $h \cdot w$ náhodných veličin.

2.3.1 Bílý šum

Situaci si zjednodušíme předpokladem tzv. *Bílého šumu*, což jsou nezávislé, stejně rozdělené (iid) NV X , takové, že $EX = 0$. Analogie s bílým světlem plyne z toho, že tento šum obsahuje všechny frekvence.

Spočítejme *power spectrum* (PS) naší šumové funkce. Power spektrum odpovídá amplitudě spektra po Fourierově transformaci funkce n (tj. $PS = |N|^2$).

Pokud provedeme korelaci bílého šumu, dostáváme

$$n \circledast n = \sigma^2 \cdot \delta \xrightarrow{CFT} \sigma^2 \quad (29)$$

Ve frekvenční oblasti je ale

$$N \cdot N^* = |N|^2 \quad (30)$$

Vztah (29) plyne z vlastností šumu (tj. z nezávislosti NV). Vidíme, že power spectrum je rovno rozptylu a je to tudíž konstantní veličina (což je opět pro bílý šum charakteristické). V praxi by měla při korelaci vzniknout matice, která je všude nulová jen vprostřed je hodnota rozptylu.

²Existuje i multiplikativní model šumu, tj. $f = f_{\text{org}} \cdot n$.

Ostranění šumu Cílem je snížit rozptyl šumové funkce. Protože lidskému oku nejvíce „vadí“ vysoké frekvence šumu, provádí se tzv. *nízkofrekvenční filtr*, který tyto frekvence potlačuje. To s sebou nese i nepříjemnou vlastnost a to, že na vysokých frekvencích jsou zaznamenány i informace o hranách, které se tímto způsobem rozmazávají.

1. Průměrování – používáme například konvoluční maticí C

$$C = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (31)$$

nebo

$$C = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (32)$$

2. Průměrování podél hran – podél hran používáme jiný druh filtru (pokud ovšem hrany známe), v ostatních částech obrazu používáme obyčejné průměrování.
3. Metoda rotujícího okna – používají se dva typy konvoluční matice pro všech 8 směrů (4+4) v daném bodě. Spočítáme konvoluce a za výslednou hodnotu bereme tu, která vznikla v okně, jež má minimální rozptyl od této hodnoty. Děláme v podstatě „průměr přes maximální homogenitu“. Tato metoda se dost používá a dává dobré výsledky.
4. Prahování – výsledek měníme pouze pokud konvoluce v bodě překročila zadaný práh.
5. Mediánový filtr – v okně provedeme seřazení dat a prostřední prvek (medián) dáváme do výsledku.

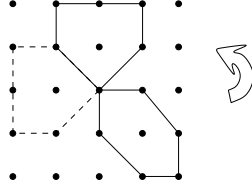
Tato metoda potlačuje šum, ale „okusuje“ okraje a rohy. Proto je vhodnější volit za výběrové okno třeba kříž.

Náročnost algoritmu pro okno šířky n by byla při klasickém třídění $O(n^4)$, při QUICKSORTu nebo HEAPSORTu $O(n^2 \log^2 n)$, případně $O(n \log n)$, když využijeme vkládání do utříděné posloupnosti. Novější algoritmus FMF (fast median filter) pracuje při složitosti $O(n)$. Urychlení je provedeno díky konstrukci a aktualizaci histogramu, hledání mediánu je pak velmi rychlé.

6. Zobecněný mediánový filtr – v okně provedeme seřazení dat a na setříděnou posloupnost aplikujeme váhovou funkci w rovnu např.

$$w = \frac{1}{4}(0, \dots, 0, 1, 2, 1, 0, \dots, 0) \quad (33)$$

Téměř všechny předchozí filtry jde provádět i ve frekvenční oblasti. Pokud víme, že má obraz jen jisté druhy hran (např. vodorovné), lze použít speciální filtr (např. $\frac{1}{4}(1 \ 1 \ 1)$). Ve frekvenční oblasti má pak tento filtr také speciální tvar.



Obrázek 10: Metoda rotujícího okna

2.3.2 Šum typu sůl a pepř (salt & pepper)

Tento speciální druh šumu vzniká při pořizování obrázku na snímacích zařízeních. Většina bodů je správně načtená, ale občas některý vypadne nebo se maximálně zesvětlí. Matematický model vypadá takto.

$$n = \begin{cases} +\infty & \text{s pravděp. } p \\ 0 & \text{jinak} \\ -\infty & \text{s pravděp. } p \end{cases} \quad (34)$$

Filtr, který tyto nesprávně načtené body opravuje, je opět v podobě konvolučního filtru s maticí tvaru

$$C = \frac{1}{8} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (35)$$

Aplikuje se pouze na tmavé a světlé body obrazu (v ostatních to nemá smysl) a dává uspokojivé výsledky.

Pokud provádíme opakované snímání (např. sledujeme experiment v laboratoři), používá se i triviální odstranění šumu pomocí průměrování více kopií, protože na každé je jiná realizace šumu, kterýžto v průměru vymizí. Pokud je možnost takového snímání, pak se jedná o ideální způsob, jak šum odstranit.

2.3.3 Periodické poškození obrázku

Pokud se na obraze vyskytuje jisté periodické poškození (např. pohled přes pleťo, mříž, fotka na vroubkovaném papíře), lze tyto „funkce“ z obrazu odstranit. Ve frekvenční oblasti totiž obdržíme amplitudové spektrum s nápadnými symetrickými píky mimo střed. Když tyto odstraníme, vymizí z obrazu i poškození.

2.3.4 Kvantifikace šumu

Pro měření „velikosti“ šumu v obraze se zavádí tzv. *signal noise ratio (SNR)*, což je funkce definovaná jako

$$SNR = \frac{|N|^2}{|F|^2}(u, v) \quad (36)$$

Tato funkce zachycuje tu vlastnost, že šum nám vadí hlavně při vysokých frekvencích. Pokud provedeme velké zjednodušení a budeme předpokládat, že obrázek (tj. signál) není prostorově korelovaný, dále budeme předpokládat bílý šum, dostáváme vztah

$$SNR = \frac{|N|^2}{|F|^2}(u, v) = \frac{\sigma_n^2}{\sigma_f^2}. \quad (37)$$

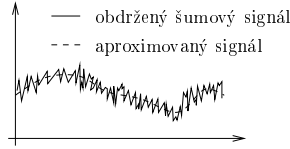
Pro rozumnou práci definujeme tzv. *odstup signálu od šumu*, který měříme v decibelech [dB] a definujeme jej jako

$$snr = -10 \log \frac{\sigma_n^2}{\sigma_f^2} \quad (38)$$

Čím vyšší je hodnota, tím lepší (kvalitnější) máme signál.

2.3.5 Aproximace

Pohlédneme-li na zašumělý signál jako na přibližnou funkci, můžeme se pokusit použít aproximačních metod k rekonstrukci původního (nezašuměného) signálu. Hledáme funkci f , která řeší rovnici



Obrázek 11: Odstraňování šumu z pohledu aproximace funkcí.

$$\min[\lambda \sum_{ij} \|f(x_{ij}) - u_{ij}\|^2 + \mathcal{J}(f)] \quad (39)$$

kde

$$\mathcal{J}(f) = \iint \left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \frac{\partial^2 f}{\partial x \partial y} + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \quad (40)$$

kde u_{ij} jsou naše naměřená data, koeficient λ udržuje hladkost výsledku a f je hledaná funkce. Tu předpokládáme ve tvaru *kubické spline* křivky, takže gradientními metodami jsme schopni nalézt její koeficienty. Ještě obecnější je rovnice pracující s váhami w_{ij}

$$\min[\lambda \sum_{ij} w_{ij} \|f(x_{ij}) - u_{ij}\|^2 + \mathcal{J}(f)], \quad (41)$$

přičemž váhy se dávají do míst předpokládaných hran.

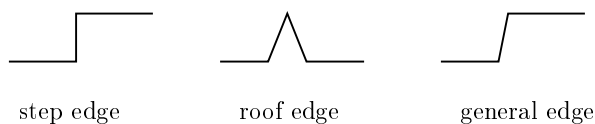
2.4 Detekce hran

Nejedná se již o předzpracování, spíše už se snažíme rozpoznat další neobrazové informace. *Hranou* pro nás bude výrazná změna intenzity obrazu. Rozlišujeme tedy

hrana (edge) – surová (obrazová) informace

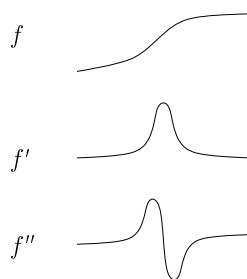
hranice (boundary) – silnější informace, popsána jinak než obrazově

Experimenty ukazují, že lidské oko se podle hran silně orientuje, ztráta hranové informace vede k zmatení a chybám v interpretaci vjemu. Detekce hran se realizuje několika způsoby. Jsou založeny na sledování 1. (resp. 2.) derivace funkce intenzity.

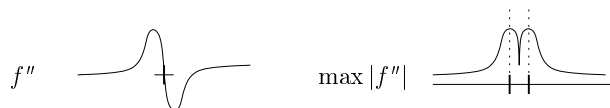


Rozdílné vnímání hran lidským okem při pozorování trojúhelníka

Obrázek 12: Typy hran pro rozpoznávání v obraze



Obrázek 13: Funkce obsahující hranu a její derivace



Obrázek 14: Hledání *zero-crossing point* pomocí $\max |\Delta f|$.

1. první derivace – *Robertsův detektor*: Jedná se o konvoluční filtr s maticí $C = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$ nebo $C = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Objevuje se však velká citlivost na šum, najdou se hrany i na místech, kde nejsou. Jistým vylepšením konvoluce s maticí

$$C = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad (42)$$

Pozn. Uvažovali jsem pouze svislé hrany, pro vodorovné hrany jsou matice symetrické.

2. první derivace – *Sobelův detektor*: Pro vodorovné hrany by konvoluční matice vypadala takto

$$C = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (43)$$

Pro zbylých 7 směrů jsou matice obdobné. Všeměrovou detekci pak realizujeme postupně a všech 8 hranových obrazů spojíme dohromady za použití maximového pravidla. Pokud nejprve obrázek vyhladíme, a pak na něj aplikujeme hranový detektor, dostaneme hranový obraz, který by vznikl jako aplikace Robertsova detektoru na větší okolí, takže žádný lepší výsledek se nedostaví.

$$C_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (44)$$

$$C_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (45)$$

$$C_3 = C_1 \circ C_2 = \begin{pmatrix} - & - & - \\ 0 & 0 & 0 \\ + & + & + \end{pmatrix} \quad (46)$$

$$(f * C_1) * C_2 = f * C_3 \quad (47)$$

3. druhá derivace – hledáme body, kde má druhá derivace průchod nulou. Neřešíme tedy rovnici $\Delta f = 0$, ale hledáme, zda se v Δf objevují velké přechody mezi kladnou a zápornou hodnotou (tzv. *zero-crossing points*). V těch místech pak zaznamenáme hranu. Tento způsob je velmi náchylný na šum (dokonce víc než Robertsův detektor). Marr a Hildreth navrhli modifikaci, při které se sleduje funkce

$$\Delta(G * f) = (\Delta G) * f \quad (48)$$

kde funkce G je kruhově symetrická Gaussova funkce. Takto provedená modifikace dává dobré výsledky. Matice ΔG se předpočítává a bývá široká 5×5 až 15×15 pixelů. Parametrizací Gaussovy funkce lze docílit výborných výsledků. Širší funkce detekuje podstatné hrany a je méně náchylná na šum, užší se chová jako předchozí detektory. Detektor má tendenci vytvářet uzavřené hrany (křivky). Tento fakt plyne z teorie Laplaceových operátorů a řešení Laplaceovy rovnice $\Delta f = 0$

$$\Delta = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (49)$$

Pro lokalizaci hrany se často používá místo složitějšího hledání „průchodu nulou“ jednoduché nalezení $\max |\Delta f|$, jak plyne z obrázku 14.

4. první derivace – *Prewittův detektor*. Jako konvoluční jádro pro detekci hran ve svislém směru se použije matice 3×3

$$h = \begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix} \quad (50)$$

5. první derivace – *Cannyho detektor*. Používá se velmi často. Byl vytvořen s požadavky:

- jedna hrana – jedna odezva
- přesná lokalizace hran
- nic nepřehlédnout
- nevytvářet zbytečné hrany

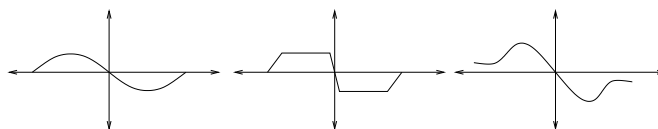
Postup:

- (a) Obraz se vyhladí pomocí konvoluce s Gaussovým jádrem, za účelem odstranění šumu
- (b) Provede se výpočet parciálních derivací, stačí i jednoduchý detektor, např. Roberts Cross function

$$\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$$

- (c) Provede se „non-maximum suppression“ – tj. na základě gradientu spočítaného z parciálních derivací odstraníme z obrázku body, které nejsou lokálními maximy ve směru kolmém na hranu
- (d) Provede se prahování celého obrázku, konkrétně prahování s hysterezí

Cannyho detektor je tvořen sadou odvozených filtrů, které závisí na parametrech.

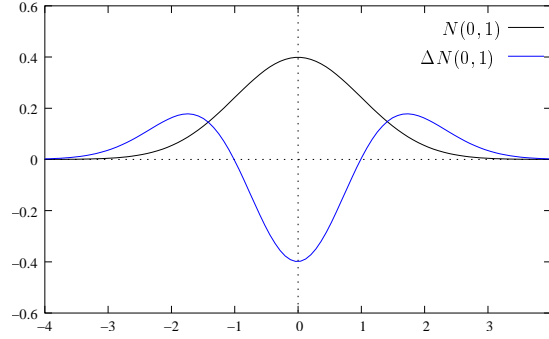


Obrázek 15: Ukázka sady filtrů *Cannyho detektoru*

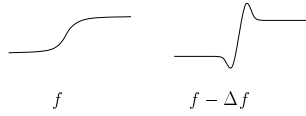
Zvýraznění hran Díky Laplaceově operátoru Δ jsme schopni v obrázku zvýraznit hrany. Stačí provést opět konvoluci s maticí C .

$$C = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad (51)$$

Vlastně jsme od funkce f odečetli Δf , jak ukazuje obrázek (viz. obr. 17).



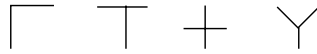
Obrázek 16: Gaussova funkce a její druhá derivace



Obrázek 17: Zvýraznění hran pomocí operátoru Δf .

2.4.1 Detekce rohů

Pro detekci rohů se nehodí klasické hranové filtry, protože charakter rohů je svým způsobem specifický rohové útvary nedávají patřičné odezvy na konvoluční jádro.



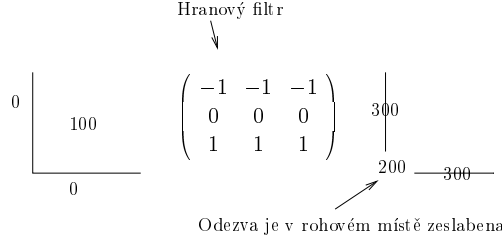
Obrázek 18: Typy rohů

- a)
 1. provedeme segmentaci obrazu
 2. procházíme hranice a detekujeme zlomová místa
 Tato metoda funguje dobře pro binární obrázky, pro klasické selhává.
- b) Corner response function (Baudetův detektor) – máme speciální konvoluční masku a procházíme obraz → dostáváme odezvu při výskytu rohu. Sleduje se veličina

$$C_B = \frac{f_{xx}f_{yy} - f_{xy}^2}{(1 + f_x^2 + f_y^2)^2} \quad (52)$$

Kitchen & Rosenfeld – vyhodnocuje se změna směru gradientu podél hran, tato změna se násobí velikostí lokálního gradientu. Funkce popisující směr gradientu je

$$\varphi(x, y) = \tan^{-1} \left(\frac{f_y}{f_x} \right) \quad (53)$$

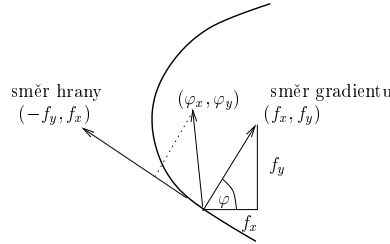


Obrázek 19: Odezva hranového detektoru na roh.

Při procházení po hraně se směr gradientu v případě výskytu rohu zákonitě musí změnit. Konstatní směr naopak ukazuje na přímou hranu. Změnu směru zjišťujeme pomocí směrových derivací φ . Směr hrany je kolmý na gradient, neboli je to směr $(-f_y, f_x)$. Funkce odezvy pak vypadá

$$\begin{aligned} C_{KR} &= \frac{(\varphi_x, \varphi_y)(-f_y, f_x)}{|(-f_y, f_x)|} |(f_x, f_y)| = \\ &= (\varphi_x, \varphi_y)(-f_y, f_x) = \frac{f_x^2 f_{yy} - 2f_x f_y f_{xy} + f_y^2 f_{xx}}{f_x^2 + f_y^2} \end{aligned} \quad (54)$$

Detektor je dost citlivý na šum a není invariantní na otáčení.



Obrázek 20: Detekce rohů metodou Kitchen & Rosenfelda.

Harrisův detektor

$$K = \frac{\overline{f_x^2 f_y^2} - (\overline{f_x f_y})^2}{\overline{f_x^2} + \overline{f_y^2}} \quad (55)$$

Pruh zde značí střední hodnotu přes okolí. Výsledkem je snížená citlivost na šum.

2.5 Restaurace obrazu

Nyní se zabýváme modelováním jiné než šumové degradace vzorového obrazu. Zajímá nás hlavně konvoluční degradace. Nechť g je náš obrázek, f bude ideální, nezkreslený obraz (originál). Symbol \mathcal{O} bude značit *operátor degradace* a n aditivní šum. Obecně případ modelujeme vztahem $g = \mathcal{O}(f)$. Předpokládáme speciální volbu operátoru \mathcal{O} ve tvaru

$$\mathcal{O} = T_G \circ T_I \quad (56)$$

kde T_G je geometrická degradace
a T_I je radiometrická degradace
výsledný model pak bude tedy

$$g = T_G \circ T_I(f) + n \quad (57)$$

Pokud nepředpokládáme geometrické zkreslení a uvážíme, že ve většině běžně dostupných zobrazovacích systémů lze T_I modelovat pomocí konvoluce, můžeme vztah přepsat na

$$g = f * h + n \quad (58)$$

Funkce h se zde označuje jako *Impulsní odezva* a jedná se o tzv. *point spread function* (polohově invariantní) funkci. Polohová invariance značí skutečnost, že výsledek vztahu ovlivňující dva pixely není závislý na jejich poloze, nýbrž, jen a pouze na jejich vzájemné vzdálenosti.

Předpokládejme nyní $n = 0$ v rovnici (58). Pak s touto můžeme provést Fourierovu transformaci. Dostáváme

$$G = F \cdot H \quad (59)$$

Funkce H se označuje jako *přenosová funkce*. Pokud známe h lze navrhnout *inverzní filtr* ve frekvenční oblasti.

$$F = G/H \quad (60)$$

Po inverzní Fourierově transformaci

$$\hat{f} = \mathcal{FT}^{-1}(F) \quad (61)$$

obříme obraz podobný originálu. Dělení ve vztahu (60) provádíme po složkách. Nulové body v H můžou následně vadit při inverzní Fourierově transformaci, proto provádíme bilineární interpolaci funkce F z nejbližšího okolí v místech, kde bylo H nulové.

Pro model se šumem bude

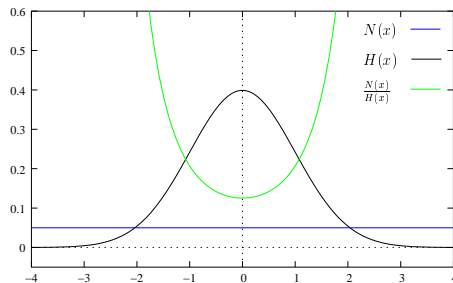
$$g = f * h + n \quad (62)$$

$$G = F \cdot H + N \quad (63)$$

$$F = \frac{G - N}{H} = \frac{G}{H} - \frac{N}{H} \quad (64)$$

Člen $\frac{N}{H}$, který jsme v předchozím zanedbávali může mít velmi zásadní vliv na výsledek a to i pro malá n jako ukazuje graf na obrázku 21. Tato metoda je proto v reálných obrázcích téměř nepoužitelná.

Další možností je například předpokládat H ve tvaru



Obrázek 21: Hodnoty funkcí $H(x)$, $N(x)$ a jejich podílu

$$H = \begin{pmatrix} a & b & a \\ b & c & b \\ a & b & a \end{pmatrix} \quad (65)$$

a řešit lineární soustavu rovnic

$$G = F \cdot H \quad (66)$$

2.5.1 Wienerův filtr

Požadavkem na Wienerův filtr W bylo

$$\hat{F} = G \cdot W \quad (67)$$

kde \hat{F} se definuje

$$\hat{F} = \|f - \hat{f}\|^2 \quad (68)$$

a hledáme

$$W = \operatorname{argmin} E(\|f - \hat{f}\|^2) \quad (69)$$

Ve vztahu (69) minimalizujeme přes všechny realizace šumu. Matice W není přesně daná, je to funkce obrazu! Po netriviálním odvození dostaneme výsledek

$$W = \frac{1}{H} \cdot \frac{|H|^2}{|H|^2 + \varphi(u, v)} \quad (70)$$

Funkce $\varphi(u, v)$ značí SNR (signal noise ratio). Připomeňme její definici ve frekvenční oblasti.

$$\varphi = \frac{|N|^2}{|F|^2} \quad (71)$$

Hodnoty $|N|^2$ a $|F|^2$ se nazývají výkonová spektra (power spectrum). Je vidět, že při nepřítomnosti šumu přechází Wienerův filtr na předchozí jednoduchý

filtr. Pokud neznáme (což je většinou) $|F|$ a $|N|$, můžeme zkusit uvažovat bílý šum a nekorelovanost původního obrazu.

$$f \circledast f = \delta(x) \sigma_f^2 = \begin{cases} \sigma_f^2 \cdot \delta(0) & \text{pokud leží přesně na sobě} \\ 0 & \text{jinde} \end{cases} \quad (72)$$

Budeme mít

$$\varphi = \frac{\sigma_n^2}{\sigma_f^2} \quad (73)$$

Postupujeme tedy tak, že s hodnotou φ začínáme na nízko ($\sim 10^{-3}$) a postupně ji zvyšujeme a sledujeme, co se děje s obrazem.

Stále neznámé h , které navíc může být tvaru $h = h_1 * h_2 * h_3 * \dots$, nelze obecně zjistit. Můžeme jej pouze předpokládat ve speciálních tvarech, abychom znali tvar impulsní odezvy. Reálně máme většinou jeden z těchto druhů degradace.

- a) špatné zaostření
- b) turbulence (např. snímek přes tlustou vrstvu atmosféry)
- c) rozmazání pohybem

ad c) Je-li pohyb lineární (tj. po přímce) má impulsní odezva tvar obdélníka.

- délka je úměrná délce expozice (při konstantní rychlosti pohybu)
- šířka bude obrácená hodnota délky, abychom zachovali $\int h = 1$ a nevnašeli tak do obrázku jas.
- ve směru kolmém na směr pohybu se tento obdélníček chová jako δ -funkce, tedy nic nemění.

Po Fourierově transformaci obdržíme ve frekvenční oblasti známou funkci sinc, natočenou do směru pohybu. Slangově se tato funkce označuje jako *vlnitý plech*³.

- nulové hodnoty funkce H budou ležet na přímkách kolmých na směr pohybu. Čím blíže budou k sobě, tím je rozmazání větší.
- stejné nulové body budou zachovány i ve funkci G . Můžeme tak najít směr a délku pohybu (během expozice). Získáváme odhad pro H do Wienerova filtru.

Podobně lze odhadovat impulsní odezvy na jiné druhy pohybu (např. dítě na houpačce). Některé druhy pohybů lze převést na známé užitím souřadnicové transformace. Třeba rotační pohyb gramofonové desky při snímání shora, převedeme na pohyb lineární s konstantní (úhlovou) rychlostí pomocí radiálních souřadnic.

Při nekonstatních rychlostech je situace obtížná, protože se vytrácí polohová invariance.

³Název zavedl J. Flusser :-)

ad a) Ideálním rozostřením světelného bodu je kruh.

- ve skutečnosti klesá jeho intenzita ke krajům
- impulsní odezva je tvaru válce – přenosová funkce má nulové body na soustředných kružnicích

$$H \sim \frac{J_1}{r} \quad \begin{matrix} J_1 & \dots \\ r & \dots \end{matrix} \quad \begin{matrix} \text{Besselova funkce 1. druhu} \\ \text{radiální souřadnice} \end{matrix} \quad (74)$$

Díky výrazu $\frac{1}{r}$ ve vztahu (74) jsou Besselovy kmity tlumené.

- stejné nulové body budou zachovány i ve funkci G . Postup je obdobný jako v případě lineárního posunutí. Změříme vzdálenost kružnic, odhadneme impulsní odezvu a aplikujeme Wienerův filtr.

ad b) Vzniká např. při delší expoziční době, než je perioda Brownova pohybu částic v atmosféře.

- impulsní odezva (a přenosová funkce) jsou Gaussovy funkce. Nemají tedy nulové body a nelze proto použít metodu z předchozích případů
- k odhadu se používají body, které nemění polohu, např. hvězdy ve snímcích oblohy (astronomie)

2.5.2 Problém slepé dekonvoluce

V obecném případě pro neznámé h nemá úloha jednoznačné řešení. Zavádíme proto dodatečné omezující podmínky.

- nezápornost originálního signálu (snímku)
- omezenost supportu h
- minimalizace jistého potenciálového funkcionálu výsledného obrazu

Jiným problémem je tzv. *vícekanálová slepá dekonvoluce*, kdy máme k dispozici několik stejných snímků s jinými impulsními odezvami. Tedy systém

$$g_1 = f * h_1 + n_1 \quad (75)$$

$$\vdots$$

$$g_l = f * h_l + n_l \quad (76)$$

Tato úloha je nepoměrně snažší, protože máme daleko více informací a z každého g_i jsme schopni zrekonstruovat jiné části obrázku. Výsledek tedy můžeme „poslepnout“ z několika takových částí. Také jsme schopni daleko lépe odstraňovat případné šumy, podobně jako při průměrování více kopií.

2.6 Geometrické zkreslení

Model geometrického zkreslení obrazu lze zapsat jako

$$g = T_G(f) \quad (77)$$

Máme-li snímek téže scény z různých pohledů (tj. s jiným geometrickým zkreslením) a potřebujeme-li zjistit odpovídající pixely (tj. aby stejné pixely měly stejné souřadnice), mluvíme o *Registraci obrazu (image registration)*. Využití spočívá např. v detekci časových změn. Nepřesná registrace vede na chybnou detekci resp. zjištění změn tam kde nejsou.

Registrace se provádí pomocí řídicích bodů (control points). Pokud jsou správně nalezeny odpovídající řídicí body v obou obrazech, je možné sestavit geometrické transformace a snímky zaregistrovat.

Požadavky na řídicí body

- větší počet
- detekovatelné v obou obrazech
- rozmístěny po celém snímku

Po určení řídicích bodů je třeba provést vzájemnou korespondenci. Tato operace se označuje jako (*matching*). K tomuto účelu musíme mít dostatečně robustní, spolehlivou metodu, neboť některé body mohou chybět, mohlo dojít k nepřesnostem atp.

Jako významné řídicí body se volí *rohy, průsečíky* – tyto jsou však citlivé na šum, nebo *těžiště uzavřených oblastí* – na šum více stabilní.

Hledání odpovídajících bodů se provádí

- topologicky (předpokládaný výskyt)
- prohledává se okolí bodu s ohledem na intenzity (barvy) okolních pixelů

Mapovací funkce (Mapping function design) Při vlastním mapování hledáme

$$u = f(x, y) \quad (78)$$

$$v = g(x, y) \quad (79)$$

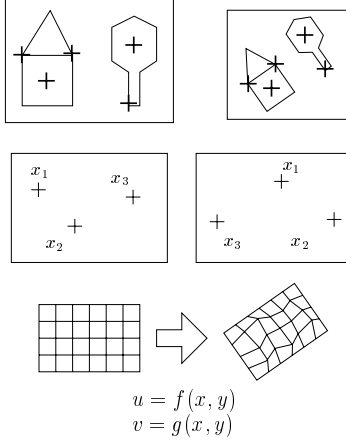
s podmínkou P1:

$$\begin{aligned} u_i &= f(x_i, y_i) \\ v_i &= g(x_i, y_i) \quad \forall i \in m, \text{ kde } m \text{ je počet řídicích bodů} \end{aligned} \quad (80)$$

Funkce f, g předpokládáme spojitě a dost hladké, navíc parametrizovatelné (např. afinní transformace). Parametry funkcí hledáme řešením následujících úloh

- *interpolační úloha* – s podmínkou P1
- *extrapolační úloha* – bez podmínky P1

Po nalezení funkcí již nic nebrání tomu, abychom obrazy namapovali bod po bodu.



Obrázek 22: Postup při registraci dvou snímků

2.7 Výběr řídících bodů (control point selection)

Metody dělíme na

- *Korelační (signálně závislé)* – korelaci provádíme se skutečnými obrazovými hodnotami. Jeden obraz (okno) posouváme po druhém a počítáme korelaci. Pokud existuje větší či menší shoda obrázku, bude na daném bodě ve výsledné matici vysoká hodnota. Při konstatní korelační matici pak tento postup odpovídá počítání vztahu

$$\frac{E[(X - EX)(Y - EY)]}{\text{var} X \text{var} Y} \quad (81)$$

což odpovídá vztahu

$$C(X, Y) = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} ((X_{ij} - \bar{X}) \cdot (Y_{ij} - \bar{Y}))}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - \bar{X})^2 \cdot (Y_{ij} - \bar{Y})^2}} \quad (82)$$

Metoda funguje dobře pro obrazy se stejnou intenzitou nebo pokud došlo ke globálním změnám kontrastu a jasů. Ostatní změny dost vadí. V základním provedení by metoda fungovala jen pro posun.

Provádí se modifikace, kdy je korelační matice trojrozměrná. Časová náročnost je ale příliš velká, proto se metoda používá pro malé rotace. Metoda dává dobré výsledky po hranové detekci v obou obrazech.

Urychlení výpočetní složitosti plyne z *Korelačního teoremu*

$$f \circledast g \xrightarrow{CFT} F \cdot G^* \quad (83)$$

kde symbol $(\cdot)^*$ značí komplexně sdruženou matici

- *Fázová korelace* – Využívá Fourier-shift-theoremu. Máme dány funkce $f(x)$ a $g(x) = f(x - a)$. Po Fourierově transformaci bude

$$f(x) \xrightarrow{CFT} F(u) \quad (84)$$

$$g(x) \xrightarrow{CFT} e^{iau} F(u) \quad (85)$$

a protože se snažíme získat vektor posunutí a , spočteme

$$\frac{F \cdot G^*}{|F||G|} = \frac{F^2 e^{iau}}{|F|^2} \xrightarrow{CFT^{-1}} \delta(x - a) \quad (86)$$

Tím vznikne dobře lokalizovatelný pík. Metoda funguje dobře, odpovídá vlastně detekci hran a korelaci v obrazové oblasti.

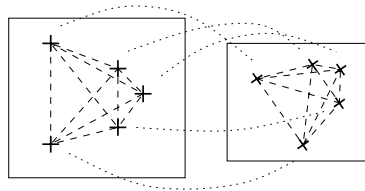
- *nepodobnost (SSDA)* – označuje se také jako „klasická korelace“. Hledá se bod, kde vztah $\sum |f_{ij} - g_{ij}|$ nabývá minima. Hledání minima zřejmě urychluje metodu, protože při napočítávání sumy, máme samé kladné hodnoty a jakmile jednou částečný součet převýší již vypočtené minimum, lze přímo přejít k dalšímu bodu.

Pro snadnou hardwarovou implementaci korelace jsou tyto metody dosti používány.

- *Point-based method* – pro zjištění korespondence řídících bodů (matching) se vytváří úplný graf a hledají se 4 parametry pro geometrickou transformaci (posun 2, rotace 1, měřítko 1). Porovnáváme po dvojici úseček a zjišťujeme, jak ostatní body padají do okolí řídících bodů.

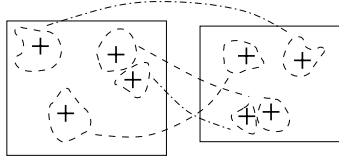
Další možnost je provést matching a zaznamenávat do parametrického prostoru provedenou transformaci. Pak hledat shluk v tomto prostoru a vybrat tak nejlepší parametr.

Pro urychlení se nekonstruuje úplný graf, ale třeba konvexní obálka nebo kostra grafu. Tím ovšem zavádíme do algoritmu vyšší náchylnost k chybám a nestabilitě.



Obrázek 23: Zjištění vzájemné korespondence řídících bodů (matching)

- *Region-based method* – pro okolí řídících bodů hledáme společné charakteristiky, které by měli být
 - *invariantní vzhledem k transformaci* – rotace, měřítko, ...
 - *dostatečně diskriminabilní*



Obrázek 24: Určování okolí bodů a jejich charakteristik (Region-based method)

2.8 Mapovací funkce

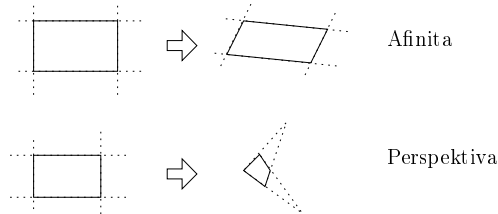
Předpokládejme transformační model

$$\begin{aligned} u &= a_0 + a_1x + a_2y \\ v &= b_0 + b_1x + b_2y \end{aligned} \quad (87)$$

Tento model (*Afinita*) je jeden z nejjednodušších, přesto velmi používaný. Zachovává rovnoběžnost přímek. Dalším modelem je *Perspektivní projekce*.

$$\begin{aligned} u &= \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y} \\ v &= \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y} \end{aligned} \quad (88)$$

Perspektivní projekce je obecnější transformace. Má však tu vlastnost, že při pozorování z větších vzdáleností v afinitu přechází, neboť c_1 a c_2 ve vztazích (88) budou malé.



Obrázek 25: Základní mapovací funkce

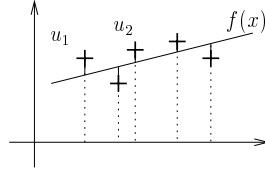
Jeden z nejobecnějších modelů má tvar

$$\begin{aligned} u &= a_0 + a_1x + a_2y + a_3xy [+ a_4x^2 + a_5y^2] \\ v &= b_0 + b_1x + b_2y + b_3xy [+ b_4x^2 + b_5y^2] \end{aligned} \quad (89)$$

Tento model je silně nelineární, nezachovává přímky ani rovnoběžnost. Používá se často.

2.8.1 K metodě nejmenších čtverců

Předpokládejme parametrizovatelnou funkci $f(x, a_1, \dots, a_l)$ a k ní předem zadanou množinu bodů $U = \{[x_1, y_1], \dots, [x_K, y_K]\}$. Definujme funkci d



Obrázek 26: K metodě nejmenších čtverců

$$d = \sum_{i=1}^K (f(x_i) - y_i)^2 \quad (90)$$

Protože hledáme minimum funkce d , mluvíme o *metodě nejmenších čtverců*. Položíme-li parciální derivace podle parametrů do rovnosti s nulou, dostáváme

$$\frac{\partial d}{\partial a_i} = 0 \quad \forall i \in \hat{K} \quad (91)$$

Což je obecně nelineární systém rovnic, po jehož vyřešení a ověření, že se jedná o minimum, získáme námi hledané koeficienty. Vztah (90) lze obecně psát ve tvaru $\|U - f\|_{\mathcal{N}}$, kdy norma $\|\cdot\|_{\mathcal{N}}$ definuje nějakým způsobem vzdálenost bodů od hledané funkce (křivky).

2.8.2 Vlastní mapování

Při vlastním mapování se používají přímé a nepřímé transformace.

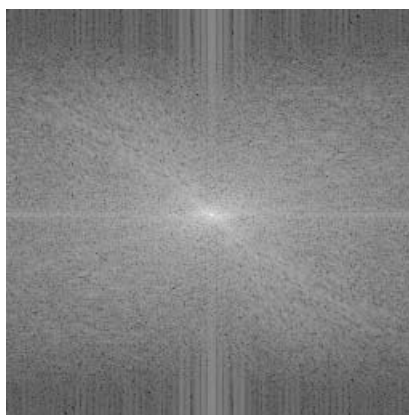
- *přímá* – procházíme všechny body ze zdrojového obrazu, transformujeme je a zapisujeme do výsledku. Vznikají situace, kdy se více bodů namapuje na stejné místo (přepisování) a výsledný obraz může obsahovat některé prázdné pixely (díry). Pro takové body je třeba provést interpolaci z okolních pixelů.
- *nepřímá* – procházíme všechny body výsledného obrazu a pomocí zpětné transformace určíme, který zdrojový bod se na něj obrazí.

Nelineární transformace se často provádí po trojúhelnících, na kterých se transformační funkce linearizuje (obdoba metody konečných prvků). Pokud se použijí lineární polynomy, má tento postup za následek nepřirozené zalamování hran na přechodových stranách trojúhelníkové sítě. Obecněji se proto definují *Radiální bázové funkce*, které tyto vlastnosti nemají.

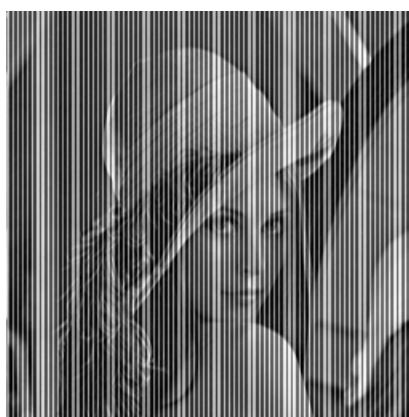
$$u = \sum a_i \varphi_i(r_i) \quad (92)$$



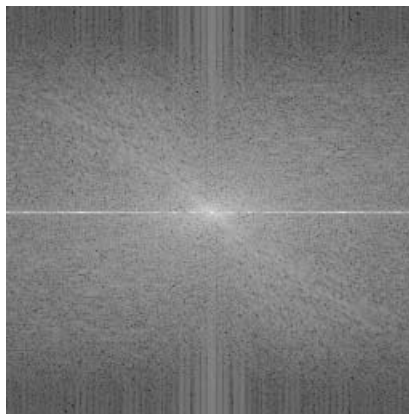
Obrázek 27: Originální obraz



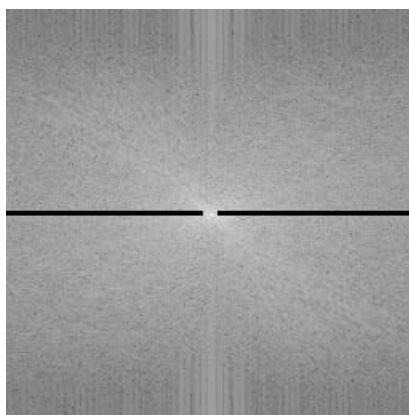
Obrázek 28: Frekvenční spektrum originálu



Obrázek 29: Obraz s periodickým poškozením



Obrázek 30: Frekvenční spektrum poškozeného obrazu



Obrázek 31: Upravené spektrum poškozeného obrazu



Obrázek 32: Výsledný restaurovaný obraz

3 Rozpoznávání dat (pattern recognition)

Úloha rozpoznávání Máme n tříd, máme rozhodnout o objektu \mathcal{O} do jaké třídy patří.

Existují 2 přístupy: strukturální (syntaktické) a příznakové (statistické)

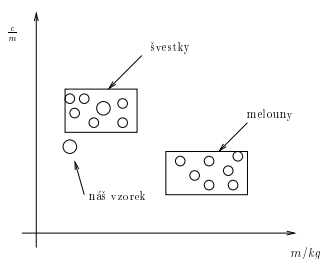
1. strukturální

- založeno na teorii jazyků, rozhodování, zda patří slovo do jazyka
- příklad: dům, střecha, zdi, dveře, okna

2. příznakové

- příklad: žokejové \times basketbalisté \rightarrow příznaky: výška, váha
- pro každý objekt máme příznakový vektor (obecně tvořen čísly, vektory, maticemi, funkcemi atd.)
- nutné definovat metriku na prostoru příznaků (někdy problém)

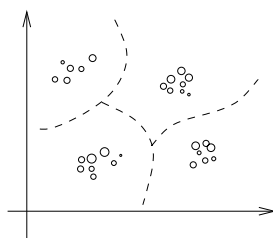
Rozdělení na třídy Nabízí se například klasifikátor beroucí minimální vzdálenost od těžišť jednotlivých tříd.



Obrázek 33: Rozdělení na třídy

- Většinou máme k dispozici tzv. *trénovací množinu* s daty.
- Otázka je podle jakého pravidla rozhodovat.

Příznakový prostor rozdělíme na oblasti (nadplochy), pak rozhodneme do které oblasti objekt padne.



Obrázek 34: Rozdělení do oblastí

Nejhorší případ je situace, kdy neznáme třídy (resp. počet), ani trénovací množinu. Pak v příznakovém prostoru provádíme tzv. *shlukování*.

Jak konstruovat nadplochy? Pouhé zjišťování těžiště, nedává dobré výsledky. Lze také počítat rozptyly a podělit vzdáleností od těžišť.

Další algoritmus: vezmu dva body (o,x) a udělám dělicí přímku, pak testuji další body z trénovací množiny a v případě špatné klasifikace posouvám, resp. natáčím přímku. (evidentně to lze udělat pouze pro lineárně separabilní množinu).

Příklady klasifikátoru: minimální vzdálenost, minimální vzdálenost od těžiště, nejbližší soused (NN- nearest neighbour, nebezpečné, citlivý na atypické prvky, není lineární klasifikátor)

NN – Nearest neighbour

- modifikace: k-NN klasifikátor (přiřadíme do třídy, kde je k -nejbližších bodů) $\rightarrow NN \equiv 1\text{-}NN$
- k je třeba volit nízké oproti četnosti skupiny

Stále nerespektujeme rozložení četností \rightarrow konstrukce statistických klasifikátorů.

$\omega_1, \dots, \omega_n$... individua
 $g_i(x)$... rozhodovací funkce
Hledáme $\mathcal{G} = \max_i g_i(x)$ (maximální ve smyslu nejlepší).

3.1 Statistické klasifikace

Náhodná veličina (NV) ξ , $E\xi$, $D\xi = E(\xi - E\xi)^2$.

$$Cov(\xi, \nu) = E(\xi - E\xi)(\nu - E\nu) \quad (93)$$

$$Cor(\xi, \nu) = \frac{Cov(\xi, \nu)}{\sqrt{D\xi D\nu}} \quad (94)$$

Bayesův klasifikátor

$$p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{\sum_j p(x|\omega_j)p(\omega_j)} \quad (95)$$

$p(\omega_i|x)$... pravděpodobnost, že individuum bude patřit do i -té třídy, přičemž jsme na něm naměřili příznakový vektor x

$p(\omega_i)$... pravděpodobnost i -té třídy v Ω

$p(x|\omega_i)$... pravděpodobnost, že na prvku ze třídy i můžeme naměřit vektor x

Pravděpodobnosti $p(\omega_i)$ určíme buď z relativních četností trénovací množiny, v tom případě je nutné zajistit, aby četnosti odpovídaly skutečnosti, nebo předpokládáme pro všechny třídy *stejnou* pravděpodobnost tj. $p(\omega_i) = \frac{1}{n}$, $\forall i \in \hat{n}$, kde n je počet tříd.

Nyní je třeba určit $p(x|\omega_i)$ pro každou třídu. Předpokládejme rozložení x ve třídě jako normální. Pomocí trénovací množiny odhadneme parametry tohoto rozložení. Pro jednorozměrný příznakový vektor máme

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (96)$$

Pro vícerozměrný příznakový vektor pak

$$p_N(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \Sigma^{-1}(\mathbf{x}-\mathbf{m})} \quad (97)$$

kde $\Sigma_{ij} = Cov(\xi_i, \nu_j)$ tj. kovariance i -tého a j -tého příznaku.

Příklad: Pro $N = 2$ bude

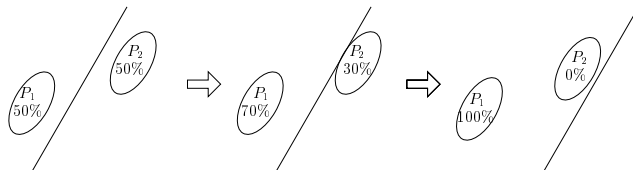
$$\Sigma = \begin{pmatrix} \sigma_1^2 & Cov \\ Cov & \sigma_2^2 \end{pmatrix}$$

Je-li $Cov = 0$ vytvoří případ $\sigma_1 = \sigma_2$ kružnici, $\sigma_1 \neq \sigma_2$ elipsu v základní poloze. Kovariance nenulová pak vede na elipsu v obecné poloze.

Každá třída má tedy vlastní Gaussovu funkci s vlastní kovarianční maticí Σ . Pokud budou pouze dvě třídy a $\Sigma_1 = \Sigma_2$, případ odpovídá Euklidovskému určování vzdáleností.

Pozn. Výraz $(\mathbf{x} - \mathbf{m})^T \Sigma^{-1}(\mathbf{x} - \mathbf{m})$ definuje tzv. Mahalanobisovu vzdálenost. Rovnost kovariančních matic je ekvivalentní skutečnosti, že je Bayesův klasifikátor *lineární*.

Pokud jsou tedy kovariančních matic stejné a změní se apriorní rozdělení pravděpodobností ($p(\omega_i)$), posunuje se rozdělovací přímka od množiny s větší pravděpodobností.



Obrázek 35: Posun rozdělovací přímky lineárního Bayesova klasifikátoru při změnách $p(\omega_i)$. Vpravo extrémní případ nulové četnosti.

Testování klasifikátoru, provádíme tak, že množinu předem klasifikovaných dat rozdělíme jednak na trénovací a testovací. Parametry určíme z trénovací množiny.

Možné chyby

- chyba modelu (není to např. normální rozložení)
- rozložení (byť normální) nemusí mít stejné parametry, někdy lepší 3 třídy klasifikovat jako dvě a pak tu jednu následně rozdělit
- chyby v odhadech kovarianční matice (celkem $\approx \frac{n^2}{2}$ prvků), pokud máme málo dat, použijeme předpoklad o stejných kovariančních maticích, pak na odhady lze použít data všech tříd.

- chyba předpokladu diagonální matice (příznaky jsou většinou silně korelované)
- curse of dimensionality (prokletí dimenze) – moc příznaku \Rightarrow stejná testovací data \Rightarrow více parametrů \Rightarrow chyby \Rightarrow Je potřeba aspon 10× více prvků než příznaků.

Odhady parametrů Maximum likelihood test:

$$\max_q \prod_{k=1}^N p(x_k|q) \quad (98)$$

$k \dots$ index v trénovací množině. Uděláme logaritmus a vyšetříme extrém funkce tj. $(\frac{\partial}{\partial q} = 0)$. Což vede na lineární rovnice a odhady parametrů

$$e = \frac{\sum x_i}{N} \quad (99)$$

$$\sigma^2 = \frac{1}{N} \sum (x_i - e)^2 \quad (100)$$

V předchozím jsme předpokládali parametrické odhady.

Neparametrické odhady používáme až selžou parametrické odhady.

- hustotu pravděpodobnosti na intervalech odhadneme pomocí relativních četností realizací
- vylepšení – odhad po bodech (posouváme interval po bodech)
- nejlepší určení spojitě hustoty pravděpodobnosti – metoda Parzenova okna

Provádíme průměrování relativních četností nějakou váhovou funkcí, aby centrální bod měl větší váhu, možné použít Gaussovu funkci. Dost záleží na tom, jak zvolíme šířku okna (support funkce) w . Extrémem jsou:

- $w = 1 \dots$ Dirackova δ -funkce, dostáváme jednotlivé realizace s pravděpodobností $\frac{1}{n}$
- $w = \max \dots$ dostaneme konstatní hustotu po celém intervalu.

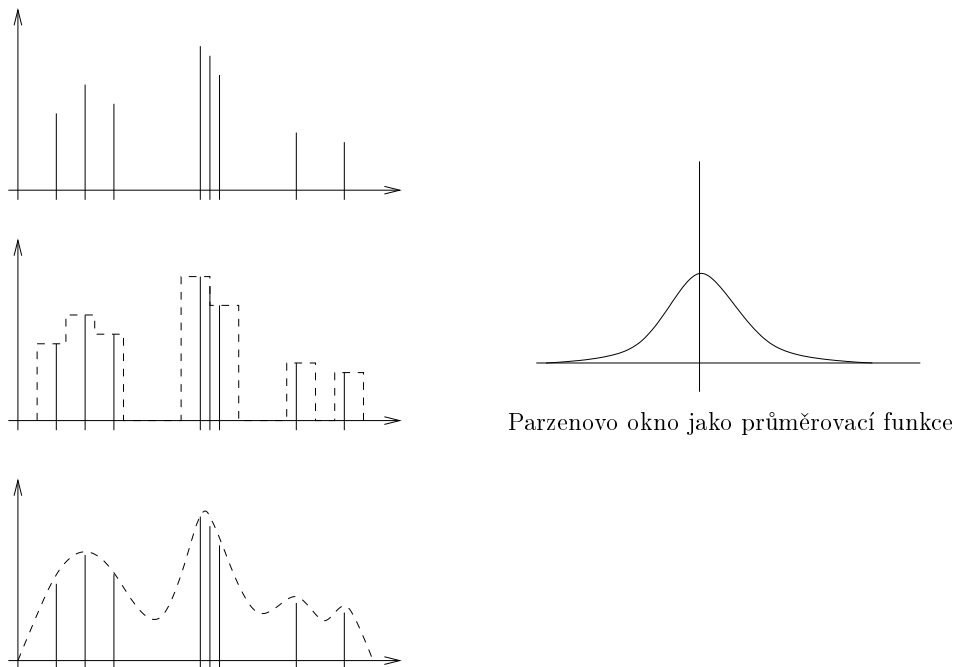
Malá okna vedou na přetrénování neboli *overtraining* – na trénovací množině to funguje bezchybně, ale na ostatních datech nespolehlivě.

Pokud klasifikace stále selhává a vše je již optimalizované, nezbyvá než zkusit změnit trénovací množinu nebo brát jiné příznaky. Často se používají kombinace více klasifikátorů $\mathcal{K}_1, \dots, \mathcal{K}_L$, protože některé fungují jen na určité příznaky. Pokud k -klasifikátor určí $p_k(\omega_i|x)$ bereme jako kombinovaný klasifikátor \mathcal{K} jednu z těchto možností:

$$p(\omega_i|x) = \max_i \prod_{k=1}^L p_k(\omega_i|x) \quad (101)$$

$$p(\omega_i|x) = \max_i \sum_{k=1}^L p_k(\omega_i|x) \quad (102)$$

$$p(\omega_i|x) = \max_i \max_{k \in \tilde{L}} p_k(\omega_i|x) \quad (103)$$



Obrázek 36: Určování četností

3.2 Klasifikace bez učení (shluková analýza)

Shlukovou analýzu neboli *clustering* používáme, když předem neznáme počet tříd. Shluk není přesně definován, ale znamená zhruba to, že rozptyly parametrů ve shluku jsou „malé“, naproti tomu vzdálenosti jednotlivých shluků „velké“.

V obecné definici můžeme shluk považovat za libovolnou podmnožinu dat, a tedy proces shlukování pak lze přirovnat k pokrytí celé množiny disjunktními podmnožinami.

Míru kvality shluků lze určit jako

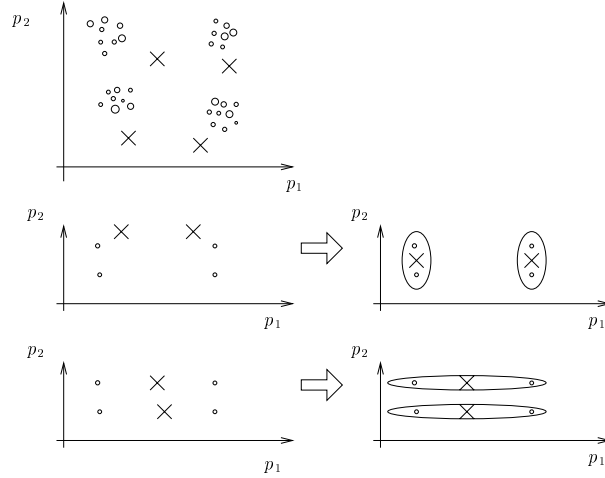
$$Q = \sum_{i=1}^N \sum_{x \in C_i} \|x - m_{C_i}\|^2 \quad (104)$$

Pokud známe počet shluků, metoda funguje dobře, pokud ne, snaží se udělat z každého prvku samostatný shluk.

N-means clustering algoritmus

1. vyberem N středních hodnot (\times)
2. klasifikujeme všechny objekty minimální vzdáleností \rightarrow dostáváme shluky
3. přepočítáme těžiště
4. opakujeme postup pokud těžiště mění své polohy

Často dává dobré výsledky, ovšem je zde silná závislost na počátečních podmínkách.



Obrázek 37: Ukázka shlukování a závislost procesu na počátečních podmínkách.

Jiné přístupy

- Data bereme jako jeden počáteční shluk a pro pevný bod zkoumáme, jak se změní kritérium kdybychom objekt přesunuli k tomuto shluku. Pokud se zmenší, přepočítáme těžiště a postup opakujeme.
- *Algoritmus ISODATA* Nejdříve máme tolik shluků jako je objektů a shluky spojujeme. Nebo máme počáteční shluk, který postupně rozdělujeme. Při tom se využívá konstrukce tzv. *dendrogramu*.
- Podmínky pro zastavení jsou např. počet shluků nebo rozptyl největšího shluku.

Kritéria pro spojení shluků

$$f = \min_{a,b} \rho(a,b) \quad (105)$$

$$f = \max_{a,b} \rho(a,b) \quad (106)$$

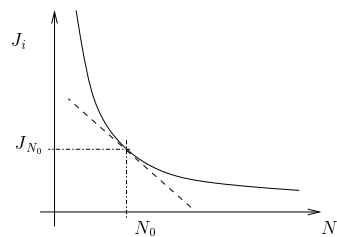
$$f = \rho(\mu_i, \mu_j) \quad (107)$$

Hausdorfova metrika:

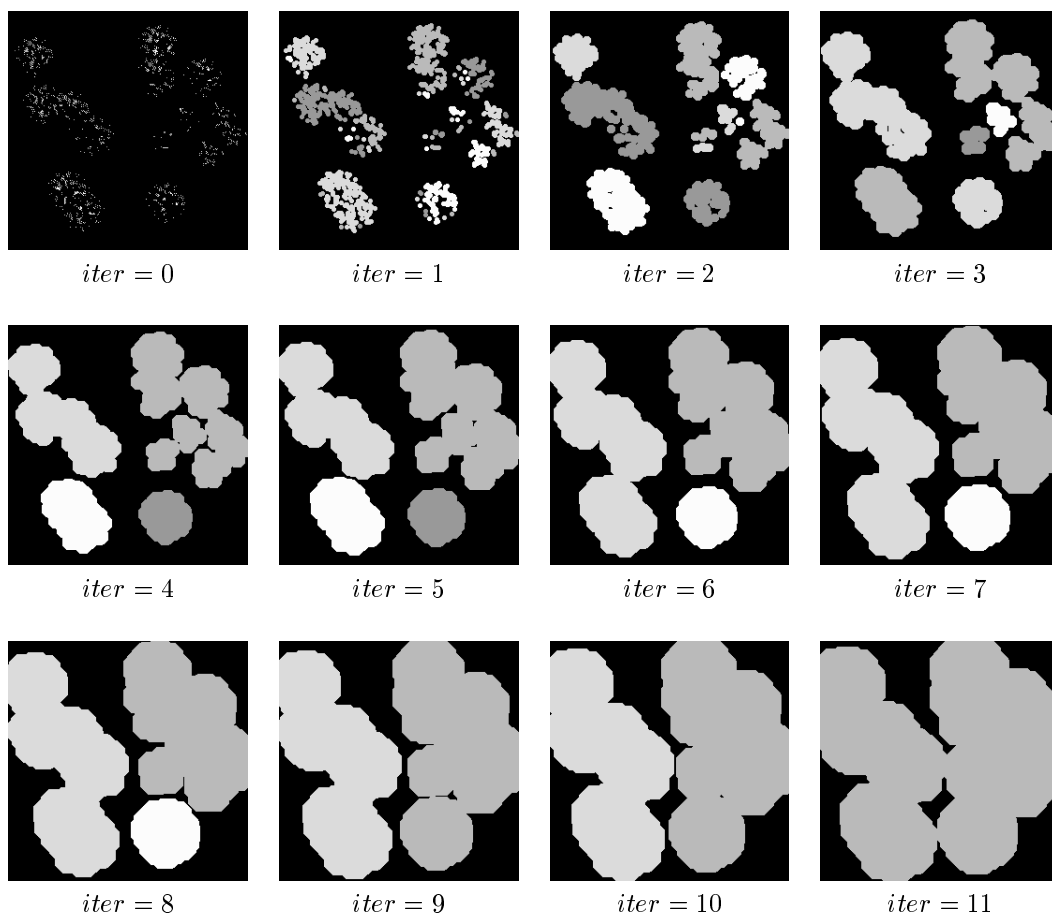
$$f = \max(\max_a \rho(a, B), \max_b \rho(b, A)) \quad (108)$$

Hausdorfova metrika je dosti náročná na výpočet. Je jasné, že shlukovací proces na zvolené metrice silně závisí.

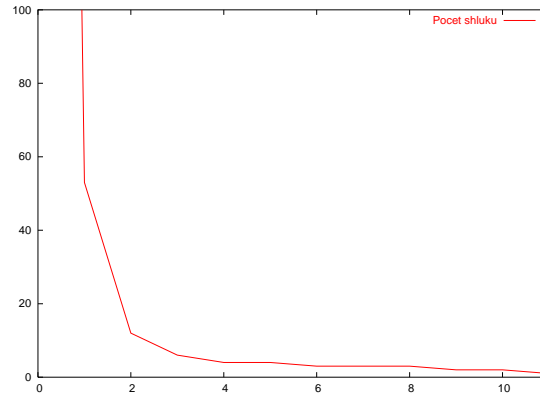
Pokud nemáme počet shluků, je úloha obtížná, potom rozdělujeme jeden shluk na malé shluky a počítáme hodnotu kritéria J_i . Pak vynášíme graf J_i pro i rovno $1, 2, \dots, N$ shluků a optimální počet shluků je pak N_0 takové, kde derivace křivky začne růst o poznání pomaleji (viz. obr. 38).



Obrázek 38: Ukázka vynášení globálního kritéria pro spočítané shluky a předpokládané optimální shlukování.



Obrázek 39: Ukázka shlukování ve 2D metodou matematické morfologie.



Obrázek 40: Graf závislosti počtu shluků na počtu iterací. Za optimální by se dala považovat druhá iterace s dvanácti shluky.

3.3 Redukce dimenzionality (příznakového prostoru)

Z naměřených dat máme většinou obrovské množství příznaků, ty jsou však silně korelované, proto se prostor snažíme redukovat.

- *One-class problem* – cílem je vybrat optimální reprezentaci (tj. x příznaků), které charakterizují všechny objekty celé množiny
- *Two-class problem* – cílem je najít $y < x$ příznaků, aby se jen velmi málo ztratila informace

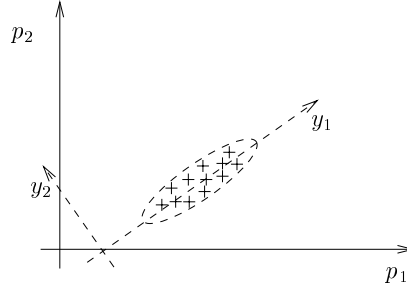
Přístupy:

1. Feature selection – vybíráme nějaké příznaky (používá se pro Two-class)
2. Feature extraction – hledáme zobrazení do méně rozměrného prostoru (používá se pro One-class)

One-class Hlavní myšlenka vychází z toho, že korelované příznaky vytváří v příznakovém prostoru (PP) více či méně protáhlou elipsu v níž můžeme kratší poloosu považovat za šum a za příznaky brát souřadnice průmětů bodů z PP do hlavní poloosy této elipsy. (viz. obr. 41).

Principle component transformation – PCT (Karhunen, Loeve) Tato transformace používá rotaci a hledá nejlepší úhel pro nekorelovanost. Je založena na transformaci SVD. Nekorelovaná matice (tedy ta, kterou hledáme) bude diagonální. Můžeme použít postup:

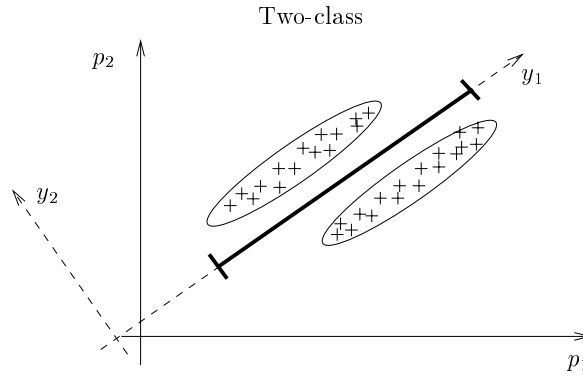
1. $K_x \dots$ matice korelace, pokud je diagonální, končím.
2. Provedeme SVD tj. $K_y = G K_x G^T$. Získáme K_y , která je diagonální. Příznaky jsou pak $y = Gx$.



Obrázek 41: Ukázka redukce dimenzionality korelovaných příznaků

Protože K_x je původně symetrická, lze diagonalizaci provést a matici transformace dokonce volit ortonormálně. Nalezená vlastní čísla pak jsou zároveň rozptyly; matici G sestavíme z přidružených vlastních vektorů.

Předchozí myšlenka, že malé rozptyly znamenají šum ale nefunguje pro *two-class*. Právě na základě jich jsme schopni třídy rozlišit (viz. obr. 42).



Obrázek 42: Two-class problém a nemožnost zanedbat příznaky s malým rozptylem, neboť hrozí ztráta rozlišovací schopnosti

Two-class a feature selection Dobře rozdělené třídy, uvažujeme-li jeden příznak, budou mít vysokou hodnotu funkce Φ

$$\Phi = \frac{(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)} \quad (109)$$

Výraz $(\mu_1 - \mu_2)^2$ se nazývá *interclass deviation*. Je vidět, že čím je větší, tím lépe jsou třídy odděleny. Ve vektorovém pojetí ($\sum ((\mu_1 - \mu_2)_i)^2$) tato funkce funguje dobře za předpokladu, že jsou příznaky nekorelované (to bohužel v praxi není příliš často).

Předchozí operace byly provedeny na trénovacích množinách, tím vznikly parametry pro třídy $\mathbf{m}_1, \dots, \mathbf{m}_K, \Sigma_1, \dots, \Sigma_K$, kde K je počet tříd. Mezi dvěma třídami tak můžeme definovat známou *Mahalanobisovu vzdálenost* pro třídy v příznakovém prostoru:

$$d_M = (\mathbf{m}_1 - \mathbf{m}_2)(\Sigma_1 + \Sigma_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)^T \quad (110)$$

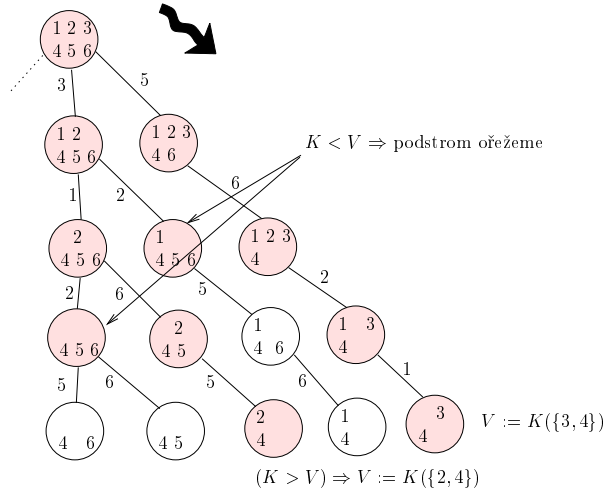
Reps. vylepšení od *Bhattacharge*

$$(111)$$

$$d_B = \frac{1}{4}d_M + \frac{1}{2} \ln \frac{|\frac{1}{2}(\Sigma_1 + \Sigma_2)|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (112)$$

Pokud chceme pouze vybrat příznaky (tj. D příznaků a chceme z nich vybrat d , tak aby $d \ll D$), vyčíslujeme d_M (resp. d_B) a hledáme d -tici, kdy d_M (resp. d_B) bude maximální.

Algoritmus Branch & Bound Myšlenkou algoritmu je efektivní procházení celého stromu příznaků (viz obr. 43). Strom má ve svých uzlech k -tice příznaků. Strom se odvíjí od kořene, kde je celká D -tice příznaků, k listům, ve kterých už je jen d -tice příznaků. V každém následujícím patře stromu ubude jeden příznak. Strom procházíme od kořene k listům a v každém uzlu spočteme kritérium ($K = d_M(\langle k\text{-tice} \rangle)$). Hodnota K se zanořováním hlouběji do stromu bude zmenšovat, až dojdeme k prvnímu listu. Zde si uložíme $V = K$ a jdeme k dalším uzlům. Nyní po výpočtu K testujeme zda $K > V$. Pokud to neplatí (tj. $K \leq V$), můžeme zahodit celý podstrom, protože hodnota K se bude dál jen snižovat a my se přece snažíme najít d -tici pro které je K maximální. Pokud se dostaneme opět do listu a je $K > V$, aktualizujeme hodnotu V na K a zapamatujeme si tento list. Tak postupujeme dále až projdeme celý strom.



Obrázek 43: Algoritmus Branch & Bound při výběru 2 ze šesti příznaků.

3.4 Suboptimální metody

Protože *fullsearch* příznaků je značně pomalý, provádí se místo něj následující heuristické algoritmy.

Algoritmus Sequential forward selection (SFS)

1. vyberu nejlepší příznak
2. přidám příznak, který vyváří nejlepší dvojici s předchozím
- \vdots
3. v k -tém kroku přidám příznak, který s předchozí $(k-1)$ -ticí vytváří nejlepší k -tici vzhledem ke kritériu.

V základním provedení má vady, protože nelze vyhodit příznak a přidáváme pouze po jednom. Takže vylepšení: přidávat j -tice (v rámci j -tice to je *fullsearch*). Algoritmus lze samozřejmě modifikovat místo procesu *forward* na odebírání tj. *backward*.

Algoritmus Plus l minus s Tento je opět rozšířením originálního SFS algoritmu. V daném kroku lze přidat l -tici a odebrat s -tici příznaků. V rozšířené verzi navíc l a s jsou dynamické. Algoritmus údajně velice dobře funguje.

4 Rozpoznávání v obraze

4.1 Segmentace

Patří sem:

- hledání objektů v obraze
- oddělení objektů od pozadí
- je třeba uchovat informace, kde se objekt nachází

Základní tři metody pro segmentaci obrazu:

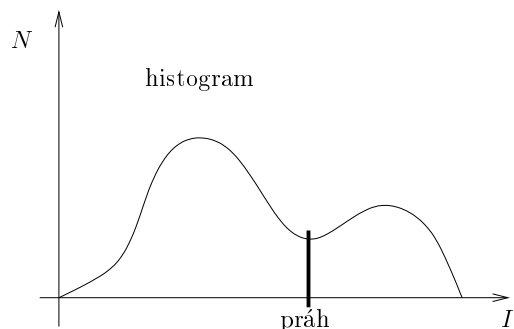
1. Prahování
2. Hrany
3. Růst oblasti (Region growing)

ad 1) Prahování V podstatě se jedná o rozdělení na pozadí a objekt. Ideální je použití na (skoro) binárních obrázcích (viz. obr. 44).

ad 2) Hrany *Hrana* je informace v obrazové oblasti. *Hranice* pak je informace vyšší úrovně, měla by být souvislá, jednobodová.

Určení hranic se většinou provádí následujícím postupem.

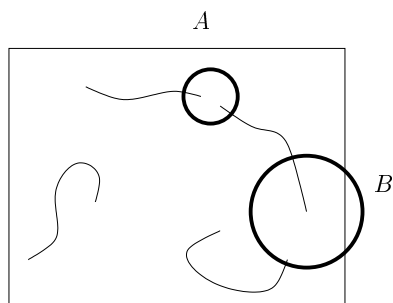
1. hranový detektor
2. binarizace
3. zbavení se *izolovaných bodů*
4. zmenšení tloušťky na jeden pixel



Obrázek 44: Histogram obrázku a zvolený práh pro segmentaci typu popředí/pozadí

5. napojování částí

Při napojování částí se prohledává okolí A koncového bodu jedné hranice a podle určitého kritéria se rozhoduje, zda sousední hranice v A napojit nebo ne. Další možností je použít hranové detektory založené na funkcích, ty často vytvářejí automaticky uzavřené křivky.

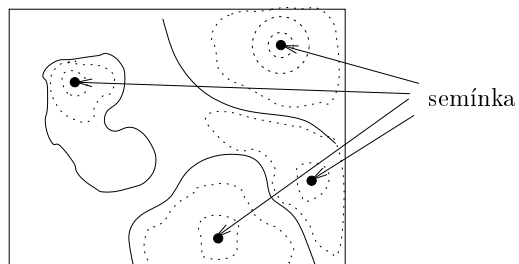


Obrázek 45: Prohledávání okolí a napojování hranice

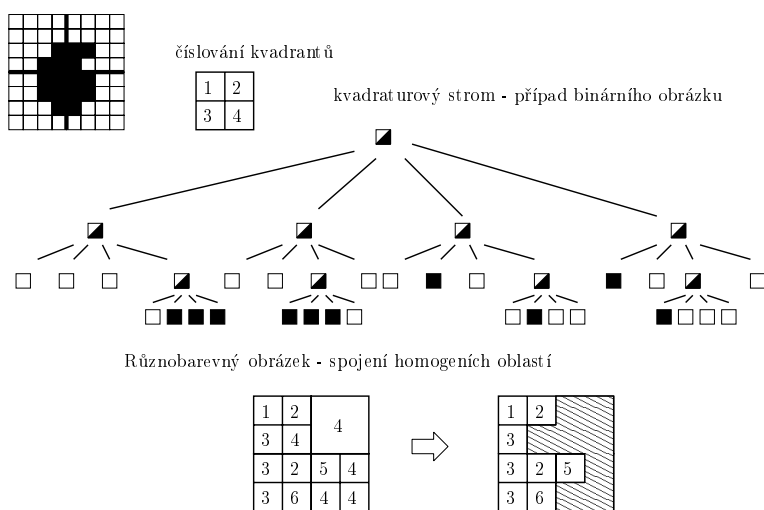
ad 3) Růst oblastí Je potřeba specifikovat tzv. semínka (*seed-points*), od nich se pak provádí buď numerická simulace roztahování křivky v závislosti na gradientu obrázku a křivosti nebo jisté prohledávání okolí a následný růst křivky do něj. Semínka buď určuje uživatel nebo se provede hrubá hranová detekce s velkým prahem a semínko se zvolí mimo hranu.

Někdy je výhodné objekt reprezentovat pomocí kvadraturového stromu, protože ten vytváří homogenní oblasti. Ty se pak můžeme snažit spojovat.

Popis hranice Uvažujeme-li nyní binární obrázek a v něm křivku, můžeme k její reprezentaci využít tzv. *chain codes* tedy řetězce čísel, kde následující číslo popisuje směr bodu oproti stávajícímu. Pokud definujeme 4 (absolutní) směry (0 – nahoru, 1 – doleva, 2 – doprava, 3 – dolů), lze křivku zapsat jako např. 020233231110. Při použití relativního vyjádření stačí 3 hodnoty (0 – rovně, 1 – doleva, 2 – doprava). Procházíme křivku a zapisujeme 0 pokud neměníme



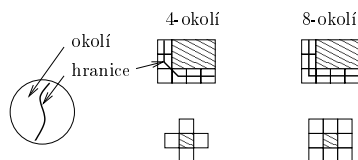
Obrázek 46: Region growing s počátečními body (semínky)



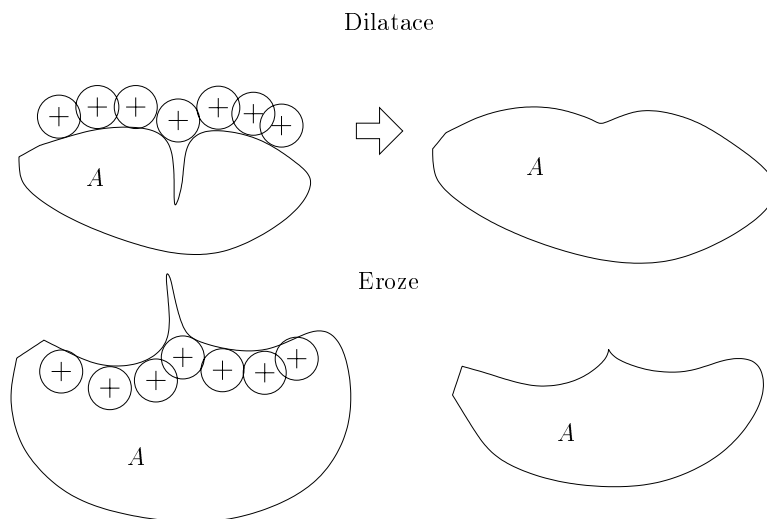
Obrázek 47: Vytvoření kvadrantového stromu a spojení oblastí do hranice

směr, 1 resp. 2 když křivka zatačí. Svislé a vodorovné části se pak vyobrazí jako sekvence nul.

Problém hranice V diskrétním rastru vzniká problém, zda za hraniční pixely považovat ty, které sdílí stěnu nebo zda stačí, sdílí-li společný bod. Následkem toho můžeme definovat čtyřokolí a osmiokolí bodu a tím i čtyř-spojitosť (resp. osmi-spojitosť).



Obrázek 48: Ukázka nejednoznačnosti hranice. Definice 4-okolí, 8-okolí



Obrázek 49: Dilatace a eroze objektu s kruhovým objektem

4.2 Matematická morfologie

Máme-li v obraze objekty s velmi členitou hranicí jde jistými operacemi tyto „záhyby“ zhladit.

Morfologické operace Předpokládejme v obraze dva objekty A, B reprezentované nějakým způsobem v obraze. Definujeme operace:

Dilatace \oplus (expanze): $A \oplus B = \{x | B_x \cap A \neq \emptyset\}$

Eroze \ominus : $A \ominus B = \{x | B_x \subset A\}$

Za objekt B volíme většinou malý kruh, takže dilatace odpovídá objíždění B po hranici A a zhlazování zálivů. Objekt se mírně rozšiřuje. Eroze odpovídá

objíždění hranice A z vnitřní strany a zahlazování výběžků. Výsledný objekt se zmenšuje (viz. obr. 49). Bude-li objekt B symetrický, pak z množinových operací vyjde vztah

$$(A \ominus B)^C = A^C \oplus B \quad \text{kde operace } (\cdot)^C \text{ značí doplněk množiny} \quad (113)$$

Tak můžeme definovat operace *otevření* a *uzavření* objektu:

Otevření: $A \circ B = (A \ominus B) \oplus B \rightarrow$ vyčistí výběžky

Uzavření: $A \bullet B = (A \oplus B) \ominus B \rightarrow$ vyčistí zálivy

Požadavky na příznaky:

1. **Invariance** – Objekty uvnitř třídy musí mít podobné vlastnosti.
2. **Diskriminabilita** – Příznaky musí být s to rozlišit příslušnost k dané třídě.
3. **Stabilita (robustnost)** – Je potřeba, aby příznaky rozumně reagovaly na případný šum. Stejně tak malá odlišnost objektu by se měla odrazit v malé změně hodnoty příznaku.
4. **Efektivita** – Nenáročnost na výpočet příznaku, pokud možno menší počet příznaků.

Předpokládejme příznakový prostor \mathcal{P} a metriku ρ , dále G souvislou, omezenou oblast v \mathbf{R}^2

Příznaky můžeme rozdělit na:

1. **Intuitivní (visual features)** – barva, plocha, obvod, ...
2. **Transformační** – jako příznaky se berou koeficienty potřebné k dané transformaci
3. **Momenty** – převod objektu do polynomiální báze $\mathcal{X} = \{x^p y^q\}$
4. **Diferenciální** – například části hranice aproximované hladkou křivkou

4.3 Intuitivní příznaky

- obsah (P), obvod (o) – Tyto příznaky nejsou invariantní na změnu měřítka, navíc spousta úplně odlišných objektů může mít stejný obsah resp. obvod.
- minimální opsaný obdélník (poměr jeho stran)
- Kompaktnost – $C := \frac{4\pi P}{o^2}$, pro jednotkový kruh vyjde 1
- Míra konvexnosti – tj. poměr obsahu objektu P a obsahu jeho konvexního obalu P' . $(\frac{P}{P'})$
- Eulerovo číslo – tj. počet komponent C minus počet děr H . $(C - H)$

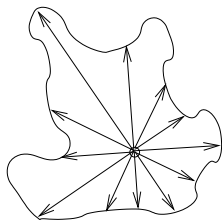
Výše uvedené příznaky se používají k předtřídění při velkém množství tříd. Obecnou nevýhodou je slabá diskriminabilita, výhodou naopak rychlé napočítání

Jak zajistit invarianty vzhledem k translaci? – stačí funkce přepočítávat s ohledem na těžiště.

4.4 Transformační příznaky

Chceme-li charakterizovat nějakým způsobem objekt, lze použít tzv. *shape-vector*, což je posloupnost vzdáleností od těžiště objektu k jeho okraji po jistých úhlových přírůstcích.

Složky Shape-vectoru (a_k) nejsou primárně invariantní na rotaci objektu ani na změnu měřítka. Pokud chceme zajistit invarianci na změnu měřítka, vybereme maximální vzdálenost, od ní začneme a složky podělíme touto maximální vzdáleností. Problém vzniká, pokud je takových složek více, resp. když je spousta složek blízkých této vzdálenosti.



Obrázek 50: Konstrukce Shape-vectoru

Invarianci na rotaci přímo nezajistíme, museli bychom rotovat posloupnost. To ovšem můžeme nahradit operací korelace. Která nám nalezení shody zajistí.

Druhý způsob je obejít všechny možné rotace trikem. Pokud provedeme *FFT*, dostaneme komplexní vektor (A_k). Z *FFT* shift-teorému plyne, že můžeme porovnávat absolutní hodnoty složek tohoto vektoru. Ty totiž budou invariantem, neboť počáteční posun (v tomto případě počáteční úhel, od kterého tvoříme shape-vector) se po *FFT* převede na komplexní jednotku, která násobí původní funkci, takže v absolutní hodnotě se vektor nezmění. Dva Shape-vectory budou po složkách splňovat vztah

$$A'_k = e^{-i\pi\varphi} A_k \quad (114)$$

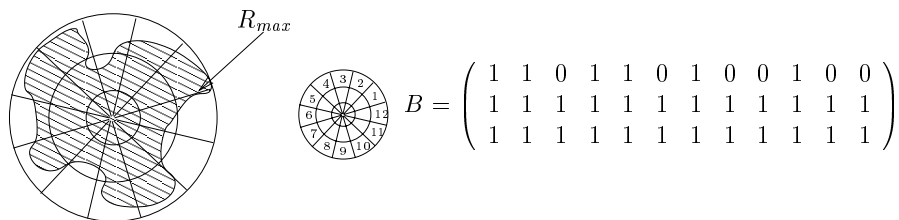
tedy

$$|A'_k| = |e^{-i\pi\varphi}| |A_k| = |A_k| = \sqrt{\Re^2 A_k + \Im^2 A_k} \quad (115)$$

Shape-vectory dobře fungují jen na tzv. *hvězdicovité objekty*. Objekty ale bývají složitější. Proto vezmeme kružnicovitou síť, začneme opět od maximálního bodu a pokud objekt překrývá více než polovinu segmentu, zapisujeme do tzv. *tvarové matice* 1 jinak 0 (viz. obr. 51).

- + každý objekt má stejně velkou matici, stejným počtem mezikruží zajistíme i invarianci na změnu měřítka
- velké paměťové nároky
- stejná míra váhy pro velké i malé segmenty → poloměry se volí, aby segmenty měly stejnou plochu

Metoda údajně velmi dobře funguje.



Obrázek 51: Příznaky tvořené mezikruhovými segmenty

Rozpoznávání polozakrytých objektů Výše uvedené příklady nefungují, protože zakrytí změní celý vektor. Tvarová matice také nelze použít, protože se podstatně změnilo těžiště. U binárních objektů lze vzít lokální vlastnosti hranice – diferenciální invariance.

Definuje se metrika jakožto délka shody s hranicí v databázi. To je sice dobré, ale je potřeba hranici zapsat i s derivacemi vyšších řádů, tím je to zase náchylné na šum, což opět vede na malou použitelnost v praxi.

Dalším nápadem je třeba rozsekat objekt na menší objekty a počítat klasické příznaky a ty pak v rámci objektu globálně sloučit. Tato metoda funguje dobře na polygonech, takže nejdříve je potřeba objekt *aproximovat polygonem*, což někdy bývá obtížné. Při porovnání dvou objektů musíme totiž udělat i stejné dělení.

Pro hladkou křivku můžeme za příznaky brát *inflexní body (IB)* tj. řešení rovnice $\ddot{x}\dot{y} - \dot{x}\ddot{y} = 0$. Tyto invarianty se chovají dobře při afinní transformaci. Pro představu stačí vzít počet inflexních bodů. Problém vzniká při napočítávání. Hodnoty bývají blízké nule, takže je třeba vhodně volit práh. Objekt rozdělíme podle inflexních bodů, ty následně spojíme a bereme „odřezky“ pro opakování postupu. Hledání IB je nestabilní.

Lepší a spolehlivější metody zatím nebyly objeveny.

Jak použít morfologie k popisu objektu? Definujeme tzv. *Morfologické spektrum (MS)*. S různými objekty (například kruhy proměnného poloměru) provádíme nad objektem operaci *otevření* a zaznamenáváme do grafu příslušnou plochu (viz. obr. 52). Tato charakteristika bývá celkem spolehlivá, i když lze najít objekty, které mají stejné MS a přitom vypadají různě.

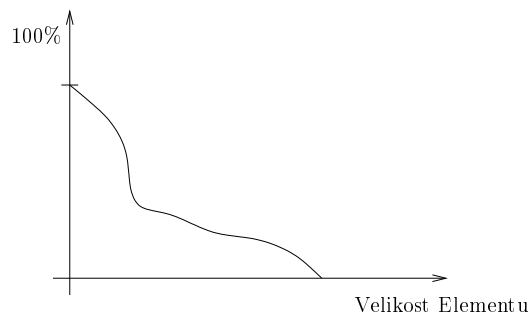
4.5 Momentové invarianty

Předpokládejme, že máme obraz zadán jako funkci $f : \mathbf{R}^2 \rightarrow \mathbf{R}$. Dále nechť $\mathcal{X} = \{x^p y^q\}$ je báze polynomů na \mathbf{R}^2 . Průměty funkce $f(x, y)$ do báze \mathcal{X} nazýváme *Momenty*.

Základní (tzv. geometrické) momenty definujeme vztahem

$$m_{pq} = \int x^p y^q f(x, y) dx dy \quad \text{pro } p, q = 0, 1, \dots \quad (116)$$

Lze ukázat, že funkce, která je počástech spojitá a s omezeným supportem je jednoznačně svými momenty určena a naopak.



Obrázek 52: Příklad morfologického spektra objektu

Pro souřadnice těžiště objektu pak vyjdou vztahy:

$$\begin{aligned} x_t &= \frac{m_{10}}{m_{00}} \\ y_t &= \frac{m_{01}}{m_{00}} \end{aligned} \quad (117)$$

Pak definujeme takzvané *Centrální momenty* jako:

$$\mu_{pq} = \int (x - x_t)^p (y - y_t)^q f(x, y) dx dy \quad \text{pro } p, q = 0, 1, \dots \quad (118)$$

Že lze centrální momenty vyjádřit jako funkci geometrických momentů je zřejmé, vzhledem k tomu, že se jedná o dvě báze stejného prostoru.

Invarianty vůči geometrickým transformacím. Hledejme nyní funkce momentů tak, aby se při zadaných geometrických transformacích zachovávaly.

Aplikace je pak zřejmá například při rozpoznávání znaků (OCR), rozpoznávání fotografií, apod.

4.5.1 Změna měřítka

Transformace změny měřítka je definována jako

$$\begin{aligned} x' &= ax \\ y' &= ay \end{aligned} \quad (119)$$

Je pak ale

$$\begin{aligned} m'_{pq} &= \int x'^p y'^q f'(x', y') dx' dy' = \int (ax)^p (ay)^q f(x, y) a^2 dx dy = \\ &= a^{p+q+2} \int x^p y^q f(x, y) dx dy = a^{p+q+2} m_{pq} \end{aligned} \quad (120)$$

protože absolutní hodnota z Jakobiánu transformace je $|\mathcal{J}| = a^2$

Chceme-li tedy zkonstruovat invariant na změnu měřítka stačí ho předpokládat ve tvaru

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{00}^\omega} \quad (121)$$

Pak bude

$$\nu'_{pq} = \frac{\mu'_{pq}}{\mu'^\omega_{00}} = \frac{a^{p+q+2}\mu_{pq}}{a^{2\omega}\mu_{00}^\omega} = \frac{a^{p+q+2}}{a^{2\omega}}\nu_{pq}$$

protože má být $\nu'_{pq} = \nu_{pq}$

Je potřeba, aby $a^{p+q+2-2\omega} = 1$, čehož docílíme volbou

$$\omega = \frac{p+q}{2} + 1 \quad (122)$$

4.5.2 Rotace soustavy

Transformace rotace o úhel φ je předepsána rovnicemi

$$\begin{aligned} x' &= x \cos \varphi - y \sin \varphi \\ y' &= x \sin \varphi + y \cos \varphi \end{aligned} \quad (123)$$

Spočteme-li například momenty do řádu dva bude

$$m'_{20} = \int (x \cos \varphi - y \sin \varphi)^2 f(x, y) dx dy \quad (124)$$

a po výpočtu

$$m'_{20} = m_{20} \cos^2 \varphi - 2m_{11} \cos \varphi \sin \varphi + m_{02} \sin^2 \varphi \quad (125)$$

podobně

$$m'_{02} = m_{20} \sin^2 \varphi + 2m_{11} \cos \varphi \sin \varphi + m_{02} \cos^2 \varphi \quad (126)$$

Sečtením obou rovnic dostaneme

$$m'_{20} + m'_{02} = m_{20} + m_{02} \quad (127)$$

což je invariant na otáčení. Dalším takovým invariantem je výraz

$$(m_{20} - m_{02})^2 + 4m_{11}^2 \quad (128)$$

Lze ukázat, že další invarianty pro otáčení už nenajdu.

Momenty do 2. řádu určují tzv. *objektovou elipsu*. Je-li $\mu_{11} = 0$ bude objekt symetrický. Středově symetrické objekty mají dokonce všechny centrální momenty lichého řádu rovny nule.

Classical moments invariants

$$\Phi_1 = \mu_{20} + \mu_{02} \quad (129)$$

$$\Phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (130)$$

$$\Phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (131)$$

$$\Phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (132)$$

$$\begin{aligned} \Phi_5 = & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + \\ & + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \end{aligned} \quad (133)$$

$$\begin{aligned} \Phi_6 = & (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + \\ & + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \end{aligned} \quad (134)$$

$$\begin{aligned} \Phi_7 = & (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + \\ & - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \end{aligned} \quad (135)$$

Hledání invariantů je velmi obtížné. Jde však použít trik, který toto hledání ulehčí. Definujeme-li si komplexní momenty podle vztahu

$$C_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x + iy)^p (x - iy)^q f(x, y) dx dy \quad (136)$$

po převodu do polárních souřadnic bude

$$C_{pq} = \int_0^{\infty} \int_0^{2\varphi} r^{p+q} e^{i(p-q)\theta} \tilde{f}(r, \theta) dr d\theta \quad (137)$$

Pokud nyní provedeme rotaci soustavy o úhel φ dostaneme vztah

$$C'_{pq} = C_{pq} \cdot e^{-i(p-q)\varphi} \quad (138)$$

Z čehož plyne, že $|C_{pq}|$ bude invariant protože $e^{-i(p-q)\varphi}$ je komplexní jednotka. Invarianty jde ale hledat i ve tvaru $C_{pq} \cdot C_{kl}$ tak, aby se komplexní jednotka vykrátily. Například $C_{pq}C_{qp}$ bude invariant. Obecně

$$I = \prod_{j=1}^n C_{p_j q_j}^{k_j} \quad (139)$$

za podmínky

$$\sum_{j=1}^n k_j (p_j - q_j) = 0 \quad (140)$$

bude invariantem. Tímto způsobem lze najít všechny invarianty Φ_1, \dots, Φ_7 . Žádný další s předchozími nezávislý už neexistuje. Je například

$$C_{00} = m_{00} \quad (141)$$

$$C_{10} = m_{10} + im_{01} \quad (142)$$

$$C_{20} = m_{20} - m_{02} + 2im_{11} \quad (143)$$

$$C_{11} = m_{20} + m_{02} \quad (144)$$

Stabilita předchozích invariantů na šum se ukazuje jako dobrá.

Jak počítat momenty? Klasická diskretizace typu

$$m_{pq} = \sum_{ij} i^p j^q f_{ij} \quad (145)$$

se sice používá, nicméně je matematicky špatně nebo spíš málo přesná.

Pro případ jednorozměrného momentu počástech konstantní funkce $f(x)$ bude

$$m_p = \int x^p f(x) dx = \sum_i f_i \int_{A_i} x^p dx \quad \text{kde } A_i \equiv \langle i, i+1 \rangle \quad (146)$$

$$= \sum_i f_i \left[\frac{x^{p+1}}{p+1} \right]_i^{i+1} = \sum_i f_i \left[\frac{(i+1)^{p+1}}{p+1} - \frac{i^{p+1}}{p+1} \right] \quad (147)$$

Bylo dokázáno, že při klasickém výpočtu momentu řádu b je chyba úměrná momentům řádu $(b-2)$.

Výpočty momentů mohou být dosti náročné, pro binární obrázky, lze výpočty urychlovat a to buď

- dekomponovat na obdélníky, řádky, sloupce, ...
- použít Greenovu formuli a integrovat přes hranici objektu

4.6 Problém rekonstrukce

Uvažujme funkci

$$\begin{aligned} F(u) &= \int e^{-iux} f(x) dx = \int \sum \frac{(-iux)^n}{n!} f(x) dx = \\ &= \sum \frac{(-i)^n}{n!} u^n \int x^n f(x) dx = \sum \frac{(-i)^n}{n!} u^n m_n \end{aligned} \quad (148)$$

Takže pokud máme momenty, můžeme sestavit řadu (148), pak provést inverzní Fourierovu transformaci a následnou rekonstrukci obrazu.

Pro rekonstrukci jsou optimální ortogonální báze funkcí (nevznikají zbytečné redundance). Báze $\mathcal{X} = \{x^p y^q\}$ však ortogonální není, proto se hledalo vyjádření momentů v bázích OG polynomů (Čebyšev, Lagrange). Tyto pochopitelně vedou na lepší výsledky při rekonstrukci. Z hlediska momentových invariantů ale nepřináší nic nového.

4.6.1 Invariant vůči konvoluci

Při aplikacích získáváme občas rozmazaná nebo jinak poničená data. Bylo by proto výhodné, kdyby existovaly invarianty na nějaký druh konvoluce $g = f \star h$. Ukazuje se, že ve speciálních případech existují.

Nechť h je konvoluční jádro takové, že $\int h = 1$ a h je středově symetrická, pak po Fourierově transformaci můžeme psát:

$$G = F \cdot H \quad (149)$$

$$|G| = |F| |H| \quad (150)$$

$$ph(G) = ph(F) + ph(H) \quad (151)$$

A protože H je nyní reálná funkce (h symetrické), je

$$ph(H) = \begin{cases} 0 \\ \pi \end{cases} \quad (152)$$

A tedy například

$$\tan ph(G) = \tan ph(F) \quad (153)$$

Takže tangenta fáze je invariantem vůči konvoluci. Je nutné ovšem poznamenat, že nejsme schopni provést zpětnou rekonstrukci, protože fázová změna $+\pi$ je pro obraz podstatná.

Má-li jádro h například rotační symetrii bude invariantem i výraz $\mu_{20} - \mu_{02}$

$$\mu_{20}^g = \mu_{20}^f \mu_{00}^h + \mu_{20}^h \mu_{00}^f \quad (154)$$

$$\mu_{02}^g = \mu_{02}^f \mu_{00}^h + \mu_{02}^h \mu_{00}^f \quad (155)$$

a po odečtení

$$\mu_{20}^g - \mu_{02}^g = (\mu_{20}^f - \mu_{02}^f) \mu_{00}^h + (\mu_{20}^h - \mu_{02}^h) \mu_{00}^f \quad (156)$$

protože je ale $\mu_{00}^h = 1$ a $\mu_{20}^h = \mu_{02}^h$ máme

$$\mu_{20}^g - \mu_{02}^g = \mu_{20}^f - \mu_{02}^f \quad (157)$$

Disclaimer

VZHLEDEM K BEZPLATNÉMU POSKYTNUTÍ PRODUKTU SE NA PRODUKT NEVZTAHUJE ŽÁDNÁ ZÁRUKA, A TO V MÍŘE POVOLENÉ ZÁKONEM. POKUD NENÍ PÍSEMNĚ STANOVENO JINAK, POSKYTUJÍ DRŽITELÉ AUTORSKÝCH PRÁV PO PŘÍPADĚ JINÉ STRANY PRODUKT „TAK, JAK JE“, BEZ ZÁRUKY JAKÉHOKOLIV DRUHU, AŽ VÝSLOVNĚ NEBO VYPLÝVAJÍCÍ, VČETNĚ, ALE NIKOLI JEN, ZÁRUK PRODEJNOSTI A VHODNOSTI PRO URČITÝ ÚČEL. POKUD JDE O KVALITU A VÝKONNOST PRODUKTU, LEŽÍ VEŠKERÉ RIZIKO NA VÁS. POKUD BY SE U PRODUKTU PROJEVILY ZÁVADY, PADAJÍ NÁKLADY ZA VŠECHNU POTŘEBNOU ÚDRŽBU, OPRAVY ČI NÁPRAVU NA VÁŠ VRUB.

V ŽÁDNÉM PŘÍPADĚ, S VÝJIMKOU TOHO, KDYŽ TO VYŽADUJE PLATNÝ ZÁKON, ANEBY KDYŽ TO BYLO PÍSEMNĚ ODSOUHLASENO, VÁM NEBUDE ŽÁDNÝ Z DRŽITELŮ AUTORSKÝCH PRÁV ODPOVĚDNÝ ZA ŠKODY, VČETNĚ VŠECH OBECNÝCH, SPECIÁLNÍCH, NAHODILÝCH NEBO NÁSLEDNÝCH ŠKOD VYPLÝVAJÍCÍCH Z UŽÍVÁNÍ ANEBY NESCHOPNOSTI UŽÍVAT PRODUKTU (VČETNĚ ALE NIKOLI JEN, ZTRÁTY NEBO ZKRESLENÍ DAT, NEBO TRVALÝCH ŠKOD ZPŮSOBENÝCH VÁM NEBO TŘETÍM STRANÁM, NEBO SELHÁNÍ FUNKCE PRODUKTU V SOUČINNOSTI S JINÝMI PRODUKTY), A TO I V PŘÍPADĚ, ŽE TAKOVÝ DRŽITEL AUTORSKÝCH PRÁV NEBO JINÁ STRANA BYLI UPOZORNĚNI NA MOŽNOST TAKOVÝCH ŠKOD.