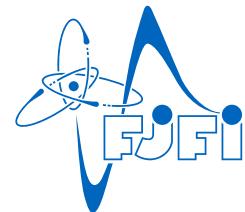




ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta jaderná a fyzikálně inženýrská



Katedra matematiky

Rozpoznávání a zpracování obrazu

Poznámky k přednáškám (vytvořeno: 2. 1. 2018)

Contents

Úvodní slovo	VII
1 Matematické základy	1
1.1 Konvoluce a korelace	1
1.2 Fourierova transformace	3
1.2.1 Filtrace ve frekvenční oblasti	6
1.3 Otázky	7
2 Digitalizace	9
2.1 Vzorkování	9
2.1.1 Aliasing	10
2.2 Kvantování	12
2.3 Charakteristiky lidského oka	12
2.4 Barevné prostory	12
2.4.1 RGB prostory	13
2.4.2 CMYK prostory	13
2.4.3 HSV prostory	13
2.5 Otázky	14
I Předzpracování	15
3 Metody pro vylepšení vzhledu	17
3.1 Jas a kontrast	17
3.2 Ekvalizace histogramu	18
3.3 Šum	18
3.3.1 Bílý šum (AGWN)	20
3.3.2 Odstranění šumu	20
3.3.3 Šum typu sůl a pepř (<i>salt & pepper</i>)	22
3.3.4 Periodické poškození obrazu	22
3.3.5 Kvantifikace šumu	22
3.4 Detekce hran	23
3.4.1 Detektory používající první derivaci	23
3.4.2 Detektory používající druhou derivaci	26

Contents

3.4.3	Detektory nepoužívající derivace	27
3.4.4	Detektory pracující ve frekvenční oblasti	27
3.4.5	Zvýraznění hran	27
3.5	Houghova transformace	27
4	Restaurace obrazu	29
4.1	Radiometrické zkreslení (radiometrický inverzní problém)	29
4.1.1	Inverzní filtr	29
4.1.2	Wienerův filtr	30
4.1.3	Základní typy radiometrických degradací	31
4.1.4	Problém slepé dekonvoluce	32
4.2	Geometrické zkreslení (geometrický inverzní problém)	32
4.2.1	Výběr kandidátů na řídící body (<i>Control point selection</i>)	33
4.2.2	Korespondence řídících bodů (<i>Control point matching</i>)	33
4.2.3	Odhadnutí modelu transformace souřadnic	35
4.2.4	Vlastní transformace	37
4.3	Otázky	37
II	Rozpoznávání dat (<i>pattern recognition</i>)	39
5	Segmentace	41
5.1	Prahování (<i>Thresholding</i>)	41
5.2	Růst oblastí (<i>Region growing</i>)	42
5.3	Spojování hran (<i>Edge linking</i>)	43
5.4	Matematická morfologie	43
6	Volba příznaků	45
6.1	Jednoduché vizuální příznaky	45
6.2	Úplné příznaky	46
6.2.1	Řetězové kódy (<i>chain code</i>)	46
6.2.2	Polygonální approximace	46
6.2.3	Radiální funkce	47
6.2.4	Tvarový vektor (<i>shape vector</i>)	47
6.2.5	Tvarová matice (<i>shape matrix</i>)	47
6.3	Transformační příznaky (<i>Transform coefficient features</i>)	48
6.3.1	Fourierovy deskriptory	48
6.4	Diferenciální invarianty	49
6.4.1	<i>Curvature Scale Space (CSS)</i>	50
7	Momentové invarianty	51
7.1	Základní vztahy	52
7.1.1	Geometrické momenty	52
7.2	Invarianty na posun	52
7.3	Invarianty na změnu měřítka a posun	53

7.4	Rotační invarianty	53
7.4.1	Obecná konstrukce	54
7.4.2	Konstrukce pomocí komplexních momentů	54
7.4.3	Pseudoinvarianty	55
7.4.4	Normalizace na rotaci	56
7.4.5	Rotační symetrie	57
7.5	Invarianty na afinní transformaci	58
7.5.1	Grafová metoda	58
7.5.2	Afnní invarianty skrz normalizaci	59
7.6	Invarianty na elastické transformace	60
7.7	Invarianty na konvoluci	60
7.7.1	Invarianty na konvoluci v FT doméně	61
7.8	Ortogonalní momenty	61
7.8.1	Ortogonalní momenty na čtverci	62
7.8.2	Ortogonalní momenty na disku	62
7.8.3	Rekonstrukce obrazu z geometrických momentů	63
7.9	Algoritmy pro výpočet momentů	63
7.9.1	Dekompoziční metody	63
7.9.2	Boundary-based metody	64
8	Waveletová transformace	65
8.1	Spojitá waveletová transformace	66
8.2	Diskrétní waveletová transformace	66
8.2.1	Vnořené podprostory (<i>Multi resolution analysis, MRA</i>)	68
8.2.2	Dekompozice funkce f	69
8.3	Aplikace waveletové transformace	71
8.3.1	Kompresie	71
8.3.2	Odstranění šumu a poškození	72
9	Klasifikace bez učitele (shluková analýza)	75
9.1	Wardovo kritérium	75
9.2	Ostatní metody shlukování	76
9.2.1	Sekvenční shlukování	76
9.2.2	Shlukování na základě grafů (<i>graph based clustering</i>)	76
9.3	Iterativní metody shlukování	76
9.3.1	<i>N-means clustering</i>	76
9.3.2	Iterační minimalizace Wardova kritéria	76
9.4	Hierarchické metody shlukování	77
9.4.1	Aglomerativní přístup	77
9.4.2	Divizivní přístup	78
9.5	Další kritéria kvality	79
9.6	Aplikace v <i>image processing</i>	79

Contents

10 Klasifikace s učitelem	81
10.1 Minimum distance klasifikátory	81
10.1.1 NN klasifikátor (<i>Nearest neighbor</i>)	82
10.1.2 k -NN klasifikátor (k - <i>Nearest neighbor</i>)	82
10.2 Lineární klasifikátory	83
10.2.1 <i>Support vector machine</i> (SVM)	83
10.3 Bayesův klasifikátor	84
10.3.1 Odhady apriorní pravděpodobnosti	84
10.3.2 Parametrické odhady class-conditional pravděpodobnosti	85
10.3.3 Neparametrické odhady class-conditional pravděpodobnosti	88
10.3.4 Aplikace Bayesova klasifikátoru v multispektrálním snímání země	88
10.4 Rozhodovací stromy	88
10.5 Účinnost klasifikace (<i>classification performance</i>)	88
10.6 Kombinované klasifikátory	89
11 Redukce dimenzionality příznakového prostoru	91
11.1 Principle Component Transform (PCT)	91
11.2 Multi-class problem	92
11.2.1 Two-class problem	93
11.2.2 Optimální metody	93
11.2.3 Suboptimální metody	93

Úvodní slovo

Tyto poznámky vznikly jako materiál ke státní závěrečné zkoušce z předmětů ROZ1, ROZ2 a SFTO v roce 2017. Vzhledem k tomu, že se jedná o studenty vytvořené poznámky, tak není možné zaručit jejich bezchybnost. Berte proto tyto poznámky pouze jako pomocný materiál a nespoléhejte při učení pouze na ně.

Oprava chyb

Jak již bylo zmíněno výše, poznámky jsou studentské a tedy obsahují chyby. Například část *Příznakové metody (signálově nezávislé)* je velmi nepřesná a potřebovala by předělat.

Verze dokumentu:

- Martin Petřek (původní verze),
- Ondřej Ticháček (první přepracování),
- Václav Mácha, Jana Vacková (současná verze).

1

Matematické základy

1.1 Konvoluce a korelace

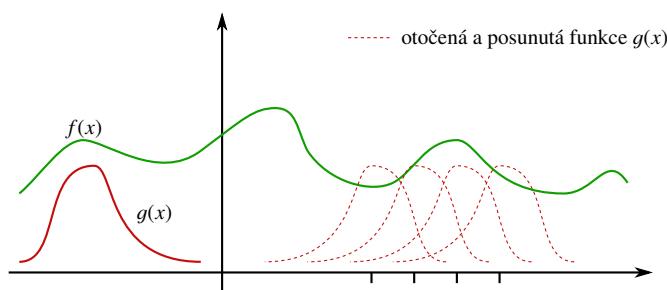
Definice 1.1 (Konvoluce a korelace v 1D). Konvolucí rozumíme zobrazení $* : L_1 \times L_1 \rightarrow L_1$ definované pro funkce f, g vztahem

$$(f * g)(x) = \int_{\mathbb{R}} f(t)g(x-t)dt, \quad \text{pro spojitého případ,}$$
$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n-m), \quad \text{pro diskrétní případ.}$$

Korelací rozumíme zobrazení $\otimes : L_1 \times L_1 \rightarrow L_1$ definované pro funkce f, g vztahem

$$(f \otimes g)(x) = \int_{\mathbb{R}} f(t)g(t-x)dt, \quad \text{pro spojitého případ,}$$
$$(f \otimes g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(m-n), \quad \text{pro diskrétní případ.}$$

Princip konvoluce je znázorněn na obr. 1.1. V podstatě se jedná o průměrování funkce f jinou funkcí g , protože g je většinou symetrická funkce s malým nosičem. Je-li např. g obdélníkový puls, pak se jedná o klasické průměrování. Dost často se požaduje, aby výsledná funkce $f * g$ měla stejně omezený obor hodnot jako původní funkce f , z tohoto důvodu se volí g splňující vlastnost $\int_{\mathbb{R}} g = 1$. Rozdíl korelace oproti konvoluci je, že funkce g se v korelacích „neotáčí“.



Obr. 1.1: Znázornění principu konvoluce v 1D.

1.1. Konvoluce a korelace

Definice 1.2 (Diracova δ funkce). Diracovou δ funkcí rozumíme funkci $\delta(x)$ splňující

$$\delta(x) = \begin{cases} " + \infty " & \dots x = 0 \\ 0 & \dots x \neq 0 \end{cases}, \quad \wedge \quad \int_{\mathbb{R}} \delta(x) dx = 1.$$

Lemma 1.3. Základní vlastnosti konvoluce:

1. $f(x) * g(x) = g(x) * f(x),$
2. $f(x) * (g(x) + h(x)) = f(x) * g(x) + f(x) * h(x),$
3. $f(x) * (g(x) * h(x)) = (f(x) * g(x)) * h(x),$
4. $a(f(x) * g(x)) = (af(x)) * g(x) = f(x) * (ag(x)),$
5. $f(x) * \delta(x) = \delta(x) * f(x) = f(x).$

Definice 1.4 (Konvoluce a korelace v 2D). Konvoluce a korelace ve 2D jsou pro matice F, G definovány po prvcích následovně

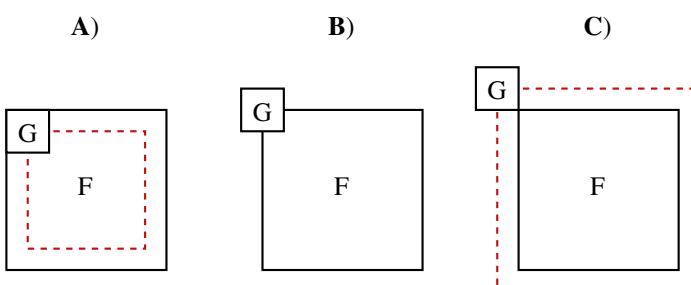
$$H[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k G[u, v] F[i - u, j - v],$$

$$H[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k G[u, v] F[i + u, j + v].$$

V případě konvoluce se jedná o posouvání matice G , kterou nazýváme maska, po matici F a do každého bodu výsledné matice napíšeme součet součinů překrývajících se prvků. V případě korelace se maska aplikuje obráceně. Koeficienty v masce udávají váhu jednotlivým pixelům. Aby při aplikaci masky na obrázek nedocházelo ke zvyšování jasu, je třeba, aby celková energie masky byla 1.

Při aplikaci masky na obrázek dochází na okrajích k tzv. okrajovému efektu. Na obr. 1.2 jsou znázorněny možnosti aplikace masky. Ve většině případů chceme, aby výsledný obrázek byl stejně velký jako původní obrázek F , proto volíme typ *same*. V takovém případě je ovšem nutné původní obrázek rozšířit. Toto rozšíření lze dělat několika způsoby:

- **přidání nul (zero padding):** po aplikaci masky dojde k rozmazání a ztmavnutí okrajů a vzniká hrana,
- **zrcadlení (mirror extension):** nedochází ke ztmavnutí ani rozmazání a nevzniká hrana,
- **periodické prodloužení (periodic extension):** toto vyžadují některé matematické věty (např. konvoluce).



Obr. 1.2: Ukázka okrajového efektu při diskrétní konvoluci ve 2D. Typy: **A) valid**, **B) same** a **C) full**.

1.2 Fourierova transformace

Fourierova řada slouží k zápisu periodického průběhu pomocí funkcí sin a cos. Základní myšlenka zápisu funkce ve formě řady z těchto funkcí je rozklad vektoru do ortogonální báze. Mějme ortogonální bázi $\mathcal{B} = \{\phi_n(x)\} = \{1, \sin mx, \cos nx\}$ v diskrétním prostoru $L_2(\langle 0, 2\pi \rangle)$ se skalárním součinem definovaným následovně

$$\langle f, g \rangle = \int_0^{2\pi} f(x) g(x) dx.$$

Funkci $f \in L_2(\langle 0, 2\pi \rangle)$ jsme pak schopni vyjádřit pomocí Fourierovy řady

$$f = \frac{a_0}{2} + \sum_{k=1}^{+\infty} [a_k \cos(kx) + b_k \sin(kx)],$$

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x) dx, \quad a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx \quad k \geq 1, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx, \quad k \geq 1,$$

kde a_k a b_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 spojité se měnící.

Fourierova transformace je integrální transformace převádějící signál mezi časově a frekvenčně závislým vyjádřením pomocí harmonických signálů. V podstatě se jedná o vyjádření původní funkce jako lineární kombinace sin a cos funkcí. Slouží pro převod signálů z časové oblasti do oblasti frekvenční. V jednodimenzionálním případě jsou bázové funkce ve tvaru

$$\{\phi_u(x)\} = \{e^{-2\pi i ux}\}$$

a ve dvoudimenzionálním případě jsou bázové funkce ve tvaru

$$\{\phi_{uv}(x, y)\} = \{e^{-2\pi i(ux+vy)}\}$$

a jejich reálná a imaginární část je tvořena „vlnitými plechy“.

Definice 1.5 (Spojitá Fourierova transformace (CFT) a inverzní spojitá Fourierova transformace (IFT)). Nechť $f \in L_1$, pak spojitá Fourierova transformace je definována vztahem

$$F(u) = \mathcal{F}[f(x)](u) = \int_{\mathbb{R}} f(x) e^{-2\pi i ux} dx \quad \text{v 1D},$$

$$F(u, v) = \mathcal{F}[f(x, y)](u, v) = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i(ux+vy)} dx dy \quad \text{ve 2D}.$$

Inverzní Fourierovu transformaci definujeme následovně

$$f(x) = \mathcal{F}^{-1}[F(u)](x) = \int_{\mathbb{R}} F(u) e^{2\pi i ux} du \quad \text{v 1D},$$

$$f(x, y) = \mathcal{F}^{-1}[F(u, v)](x, y) = \int_{\mathbb{R}^2} F(u, v) e^{2\pi i(ux+vy)} du dv \quad \text{ve 2D}.$$

Poznámka 1.6. Fourierova transformace a spojitá Fourierova transformace se nepatrně liší

$$\mathcal{F}[f(x)] = \int_{\mathbb{R}} f(x) e^{-iux} dx, \quad \mathcal{F}^{-1}[F(u)] = \frac{1}{2\pi} \int_{\mathbb{R}} F(u) e^{iux} du.$$

1.2. Fourierova transformace

Definice 1.7 (Diskrétní Fourierova transformace). Diskrétní Fourierova transformace a inverzní diskrétní Fourierova jsou definovány vztahy

$$F(k) = \mathcal{F}[f(n)](k) = \sum_{n=0}^{N-1} f(n) e^{-\frac{1}{N} 2\pi i n k}, \quad \forall k \in \{0, 1, \dots, N-1\},$$

$$f(n) = \mathcal{F}^{-1}[F(k)](n) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} F(k) e^{\frac{1}{N} 2\pi i n k}, \quad \forall n \in \{0, 1, \dots, N-1\}.$$

Existence komplexní funkce F plyne z předpokladu, není ovšem zaručeno, že i $F(u)$ bude z L_1 . Díky tomu, že $F(u)$ je komplexní funkce (spektrum), lze jí rozložit do tvaru

$$F(u) = |F(u)| e^{i \arg F(u)},$$

kde $|F(u)|$ je tzv. amplitudové spektrum a $\arg F(u)$ fázové spektrum.

Lemma 1.8. *Vlastnosti Fourierovy transformace:*

1. $\mathcal{F}[af(x) + g(x)](u) = a\mathcal{F}[f(x)](u) + \mathcal{F}[g(x)](u),$
2. $\mathcal{F}[a \cdot f(x)](u) = |a|^{-1} \mathcal{F}[f(x)]\left(\frac{u}{a}\right),$
3. $\mathcal{F}[\delta(x)] = 1,$
4. $\mathcal{F}[1](u) = \delta(u),$
5. $\mathcal{F}[R(f(x))](u) = R(\mathcal{F}[f(x)](u)),$

kde $R(\cdot)$ značí rotaci.

Věta 1.9 (Fourier Shift teorém). *Posuneme-li funkci f o konstantu a , dává Fourierova transformace vztah:*

$$\mathcal{F}[f(x-a)](u) = e^{-2\pi i u a} F(u).$$

Proof. Upravujme

$$\begin{aligned} \mathcal{F}[f(x-a)](u) &= \int_{\mathbb{R}} f(x-a) e^{-2\pi i u (x \pm a)} dx = e^{-2\pi i u a} \int_{\mathbb{R}} f(x-a) e^{-2\pi i u (x-a)} dx = \left| \begin{array}{l} y = x - a \\ dy = dx \end{array} \right| = \\ &= e^{-2\pi i u a} \int_{\mathbb{R}} f(y) e^{-2\pi i u y} dy = e^{-2\pi i u a} F(u). \end{aligned}$$

□

Tedy obraz je akorát násobený komplexní jednotkou $e^{-2\pi i u a}$, tj. jeho amplituda má 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.

Věta 1.10 (Konvoluční teorém). *Pro Fourierovu transformaci platí následující vztahy*

$$\mathcal{F}[f(x) * g(x)](u) = F(u) \cdot G(u),$$

$$\mathcal{F}[f(x) \cdot g(x)](u) = F(u) * G(u),$$

kde symbol \cdot značí násobení stejně rozměrných matic po prvcích.

Proof. Upravujme

$$\begin{aligned}\mathcal{F}[f(x) * g(x)](u) &= \int_{\mathbb{R}} \left[\int_{\mathbb{R}} f(t)g(x-t)dt \right] e^{-2\pi iux}dx \stackrel{a)}{=} \int_{\mathbb{R}} f(t) \left[\int_{\mathbb{R}} g(x-t)e^{-2\pi iux}dx \right] dt = \\ &\stackrel{b)}{=} \int_{\mathbb{R}} f(t) \left[e^{-2\pi itu}G(u) \right] dt = \int_{\mathbb{R}} f(t)e^{-2\pi itu}dt \cdot G(u) = F(u) \cdot G(u),\end{aligned}$$

kde jsme v rovnosti *a*) použili Fubiniho větu a v rovnosti *b*) větu 1.9. Druhou rovnost dokážeme obdobně

$$\begin{aligned}\mathcal{F}[f(x) \cdot g(x)](u) &= \int_{\mathbb{R}} f(x)g(x)e^{-2\pi iux}dx \stackrel{a)}{=} \int_{\mathbb{R}} \left[\int_{\mathbb{R}} F(t)e^{2\pi itx}dt \right] g(x)e^{-2\pi iux}dx = \\ &\stackrel{c)}{=} \int_{\mathbb{R}} F(t) \left[\int_{\mathbb{R}} g(x)e^{-2\pi i(u-t)x}dx \right] dt = \int_{\mathbb{R}} F(t)G(u-t)dt = F(u) * G(u),\end{aligned}$$

kde jsme v rovnosti *a*) použili inverzní Fourierovu transformaci a v rovnosti *b*) Fubiniho větu. \square

Pokud chceme ověřit konvoluční teorém v praxi, je potřeba diskrétní data periodicky prodlužovat! Co se týká náročnosti, má výpočet diskrétní Fourierovy transformace v základním tvaru složitost $O(N^2)$. V roce 1942 pánové Danielson a Lanczos zjistili, že posloupnosti délky $N = 2^k$ lze vyjádřit jako součet dvou diskrétních Fourierových transformací délky $\frac{N}{2}$, kde v jedné jsou liché a ve druhé sudé vzorky. V roce 1965 Cooley a Turkey zobecnili tuto metodu i pro posloupnosti délky $N \neq 2^k$. Tato rychlá Fourierova transformace (FFT) se používá dodnes. Lze ji sestrojit pro obecné obdélníkové matice a její složitost je $O(N \cdot \log_2 N)$. Naproti tomu konvoluce má složitost $O(N^2)$ pro každý bod masky, tedy je jasné, že při počítání konvoluce s rozměrnějšími maticemi je výhodnější použít konvolučního teorému a ve frekvenční oblasti pouze vynásobit matice.

Příklad 1.11 (Funkce $\text{sinc}(x)$). Nyní ukažme jak vypadá Fourierův obraz obdélníkového pulsu. Nechť f je funkce definovaná pro $c \in \mathbb{R}$ a $\lambda \in \mathbb{R}^+$ jako

$$f(x) = \begin{cases} c & \dots x \in (-\lambda, \lambda), \\ 0 & \dots \text{jinde}, \end{cases}$$

pak

$$F(u) = \int_{\mathbb{R}} f(x)e^{-2\pi iux}dx = c \int_{-\lambda}^{\lambda} e^{-2\pi iux}dx = \frac{c}{-2\pi iu} [e^{-2\pi iux}]_{-\lambda}^{\lambda} = c \frac{e^{-2\pi iu\lambda} - e^{2\pi iu\lambda}}{-2\pi iu} = \frac{c}{\pi u} \sin(2\pi\lambda u).$$

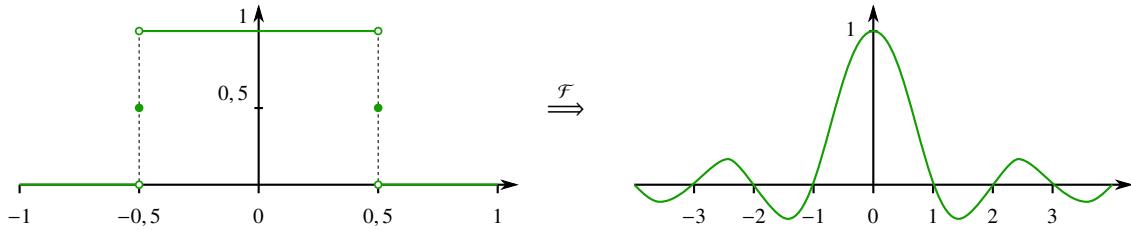
Pro $c = 1$ a $\lambda = \frac{1}{2}$ dostaneme funkci sinc

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

Ve 2D vzniká funkce $\text{sinc}(x, y)$, která je ve tvaru

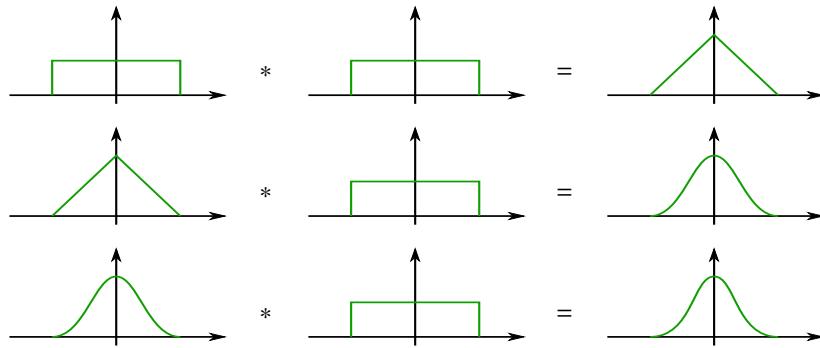
$$\text{sinc}(x, y) = \text{sinc}(x)\text{sinc}(y) = \frac{\sin(\pi x)\sin(\pi y)}{\pi^2 xy}.$$

1.2. Fourierova transformace



Obr. 1.3: Fourierova transformace obdélníkového pulsu a vzniklá funkce $\text{sinc}(x)$.

Tato funkce není středově symetrická, jak by se mohlo zdát. Další velice zajímavou vlastnost obdélníkového pulsu získáme, pokud budeme provádět konvoluci těchto pulsů. Opakováním konvoluce obdélníkových pulsů totiž vznikají tzv. B-spline křivky, které stále lépe approximují Gaussovou funkci, viz obr. 1.4.



Obr. 1.4: Ukázka opakované konvoluce obdélníkových pulsů a vznik B-spline křivek.

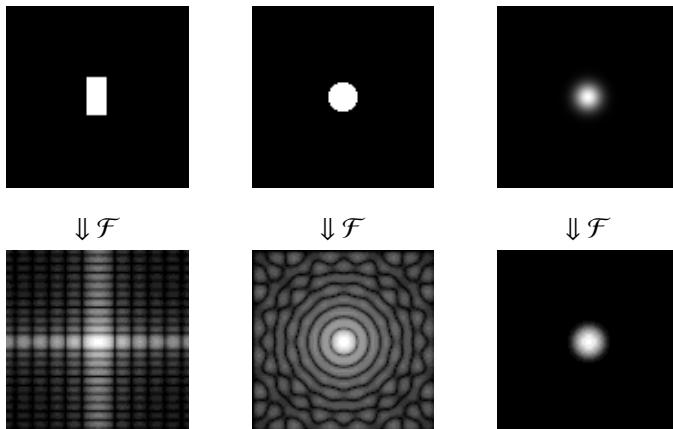
Další důležitý signál je ve 2D obdélníkový signál, kruhový signál a gaussovský signál. Amplitudové spektrum obdélníkového signálu je tvořeno „protáhlou“ funkcí sinc. Amplitudové spektrum kruhového signálu je tvořeno Besselovou funkcí a spektrum gaussovského signálu je opět Gaussova funkce. Tyto signály a jejich amplitudová spektra jsou na obr. 1.5 a na obr. 1.6 jsou jejich 3D znázornění.

1.2.1 Filtrace ve frekvenční oblasti

Pokud provedeme Fourierovu transformaci 2D obrazu, získáme její projekci do prostoru bazických funkcí a dozvíme se, jak jsou v obrazu zastoupeny jednotlivé frekvence. Protože jsme ovšem původně měli reálný signál, bude Fourierův obraz symetrický podle středu. V bodě $(0, 0)$ se nachází celková energie (součet intenzit jednotlivých pixelů) Fourierova obrazu, protože

$$F(0, 0) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) e^{-2\pi i (\frac{1}{N} n 0 + \frac{1}{M} m 0)} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m).$$

Naopak u obvodu jsou vyobrazeny vysoké frekvence. Protože Fourierova transformace detektuje pravidelnosti, tak pravidelné obrazce se Fourierovou transformací promítou na jednotlivé body. Navíc lze z Fourierova obrazu odvodit hlavní směry ve Fourierově vzoru. Ve Fourierově obrazu budou totiž linie kolmé na skutečný směr, tedy na směr v původním obrazu, viz obr. 1.7.



Obr. 1.5: Ukázka dalších důležitých signálů a jejich amplitudové spektrum.

Uvedených vlastností Fourierovy transformace lze dobře využít pro filtrování nechtěných frekvencí. Na obr. 1.8 jsou zobrazeny základní filtry pro filtraci ve frekvenční oblasti. *High-pass* filtry propouštějí vysoké frekvence a tak zachovávají hrany a detaily. Naopak *low-pass* filtry vysoké frekvence nepropouští a výsledný obraz rozmazávají.

1.3 Otázky

- *Zdvojnásobují se data při FT?*

Ne, FT je symetrická středově (n je reálné)

- *Nejvyšší frekvence u DFT?*

Pro $n = 0$ to je konstanta, $n = 1$ je to pul sinu, $n = 2$ je to sin. Takže nejvyšší vlnová délka jde přes 2 body a tím, že je to symetrické tak je to v $n = N/2$.

- *Co nese více informací – amplituda nebo fáze?*

Fáze (tu vizuální).

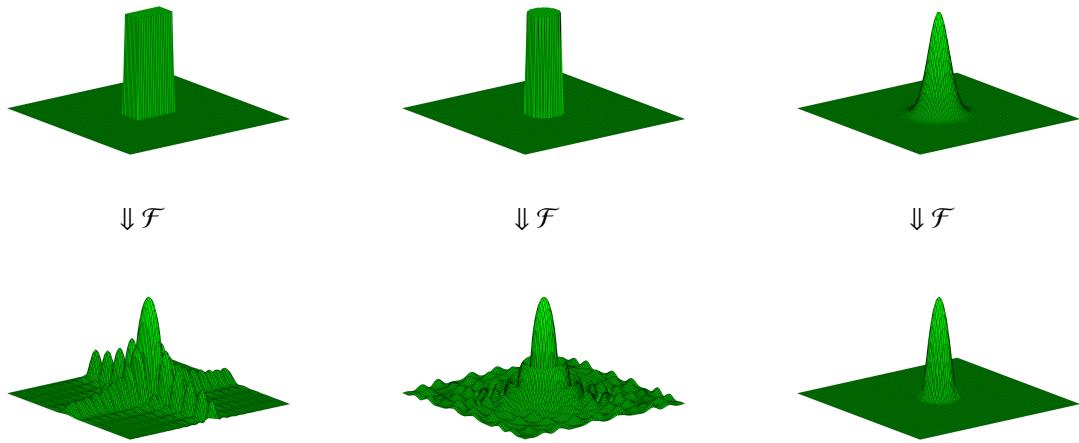
- *Co se stane, když amplitudu nahradím jedničkami a fázi nechám?*

Po inverzní FT dostanu černý obrázek a obrysy budou bílé.

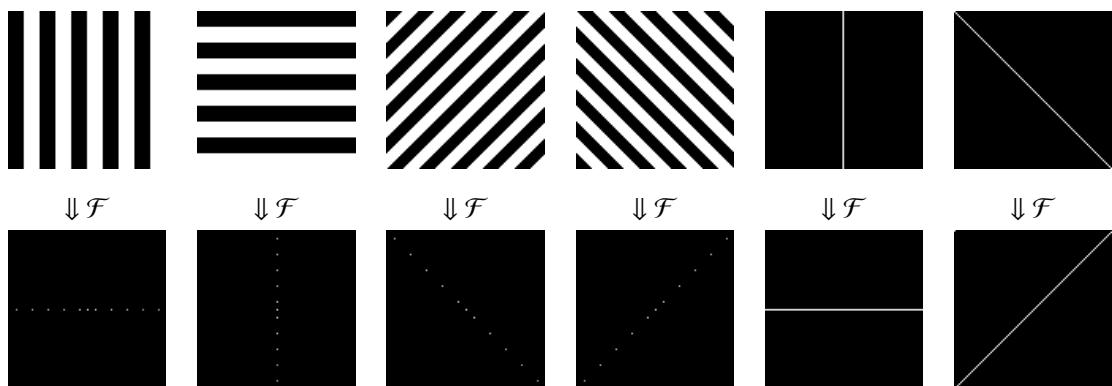
- *Porovnání rychlosti výpočtu s konvolucí*

Při malém filtrovi je rychlejší počítat v obrazové oblasti, ale při velkém je lépe přejít do frekvenční.

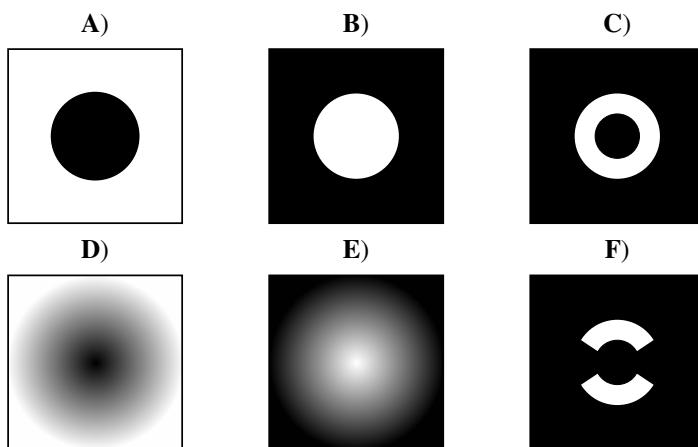
1.3. Otázky



Obr. 1.6: Reprezentace signálu z obr. 1.5 ve 3D, kde třetí rozměr reprezentuje intenzitu v daném pixelu.



Obr. 1.7: Ukázka amplitudového spektra vybraných pravidelných obrazů.

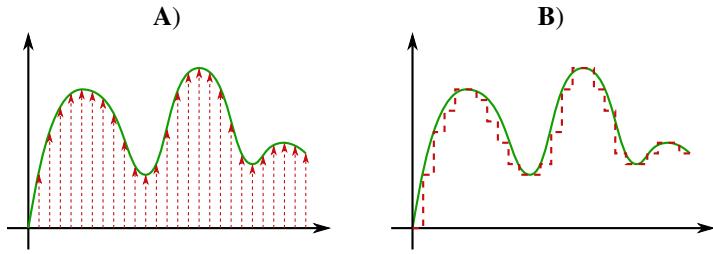


Obr. 1.8: Filtry ve frekvenční oblasti: **A)** *high-pass filtr*, **B)** *low-pass filtr*, **C)** *band-pass filtr*, **D)** *Gaussovský high-pass filtr*, **E)** *Gaussovský low-pass filtr* a **F)** *směrový filtr*.

2

Digitalizace

Digitalizaci rozlišujeme podle toho, zda diskretizaci provedeme v prostoru souřadnic či v oboru hodnot. Pak hovoříme o *vzorkování (prostor souřadnic)* nebo *kvantování (obor hodnot)*.



Obr. 2.1: Znázornění principu **A)** vzorkování a **B)** kvantování.

2.1 Vzorkování

Vzorkování lze považovat za násobení původního spojitého signálu nekonečným polem δ -funkcí, tj. použijeme vzorkovací funkci

$$s(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta(x - i\Delta x, y - j\Delta y), \quad (2.1a)$$

pomocí níž získáme výsledný digitální obraz

$$d(x, y) = f(x, y) \cdot s(x, y), \quad (2.1b)$$

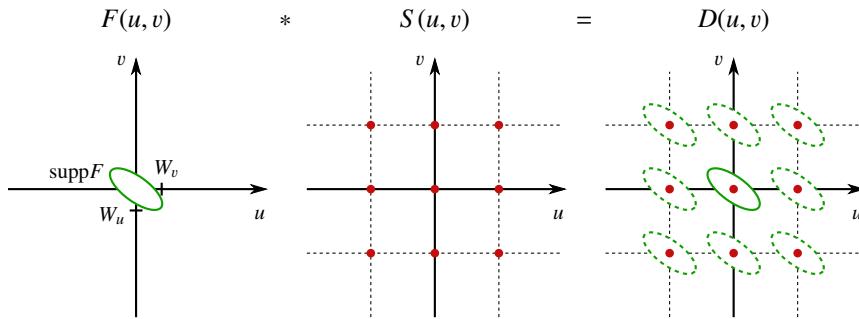
kde funkci $s(x, y)$ nazýváme vzorkovací funkce a $d(x, y)$ je výsledný digitální obraz. Lze ukázat, že Fourierův obraz nekonečného pole δ -funkcí bude opět nekonečné pole δ -funkcí s krokem $\frac{1}{\Delta x}, \frac{1}{\Delta y}$. Pokud použijeme Fourierovu transformaci na rovnice (2.1), dostáváme

$$S(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \delta\left(u - \frac{i}{\Delta x}, v - \frac{j}{\Delta y}\right), \quad (2.2a)$$

$$D(u, v) = F(u, v) * S(u, v). \quad (2.2b)$$

2.1. Vzorkování

Tedy zvyšováním vzorkovací frekvence (tj. zmenšováním Δx a Δy) současně zvyšujeme vzdálenost mezi Fourierovými obrazy dvou vzorků, protože funkce $D(u, v)$ ve vztahu (2.2) není nic jiného než opakování funkce $F(u, v)$ ve vzdálenosti Fourierových pulsů vzorkovací funkce, viz obr. 2.2. Pokud budeme chtít funkci zrekonstruovat z jejích Fourierových vzorků, je nezbytně nutné, aby tyto vzorky nebyly poškozeny vzájemným překrytím. Shannonův teorém dává odpověď, jak hustě vzorkovat, abychom neztratili obrazové informace.



Obr. 2.2: Znázornění vzorkování ve frekvenční oblasti.

Věta 2.1 (Shannonův vzorkovací teorém). *Aby bylo možné zrekonstruovat funkci $d(x, y)$ z jejího Fourierova obrazu $D(u, v)$, je třeba, aby splňovala následující podmínky.*

1. *Musí být frekvenčně omezená (frekvenčně neomezené funkce, tj. neexistuje $f_{\max} < \infty$, mají totiž neomezené frekvenční spektrum, a proto se vzorky budou vždy překrývat).*
2. *Vzorkovací frekvence musí být větší nebo rovna dvojnásobku maximální frekvence f_{\max} (tzv. Nyquistův limit), tzn.*

$$\Delta x \leq \frac{1}{2W_u}, \quad \Delta y \leq \frac{1}{2W_v}.$$

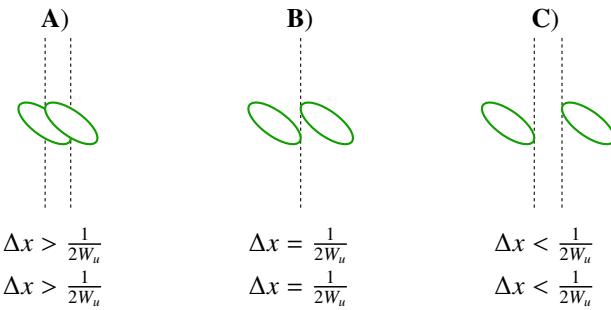
Poznámka 2.2. Uvažujme, že existuje maximální frekvence f_{\max} a předpokládejme, že se v obrazu nevyskytuje. Potom je ideálním vzorkováním Nyquistův limit, viz obr. 2.3. V případě, že se maximální frekvence v obrazu vyskytuje, tak při použití tohoto vzorkování dojde k překryvu maximální frekvence, a tedy ke ztrátě informace.

V závislosti na volbě vzorkovací frekvence může dojít ke třem situacím, viz obr. 2.3. Ideální rekonstrukční filtr má tedy ve frekvenční oblasti tvar obyčejné schodovité funkce (konkrétně je to obdélníkový puls) a její spojitý vzor v časové oblasti má tvar již zmíněné funkce sinc.

2.1.1 Aliasing

Pro frekvenčně neomezené funkce by byl teoreticky potřeba nekonečně malý krok. Vzorkováním tedy může vznikat velmi nepříjemný jev zvaný *aliasing*, který nastává ve dvou případech.

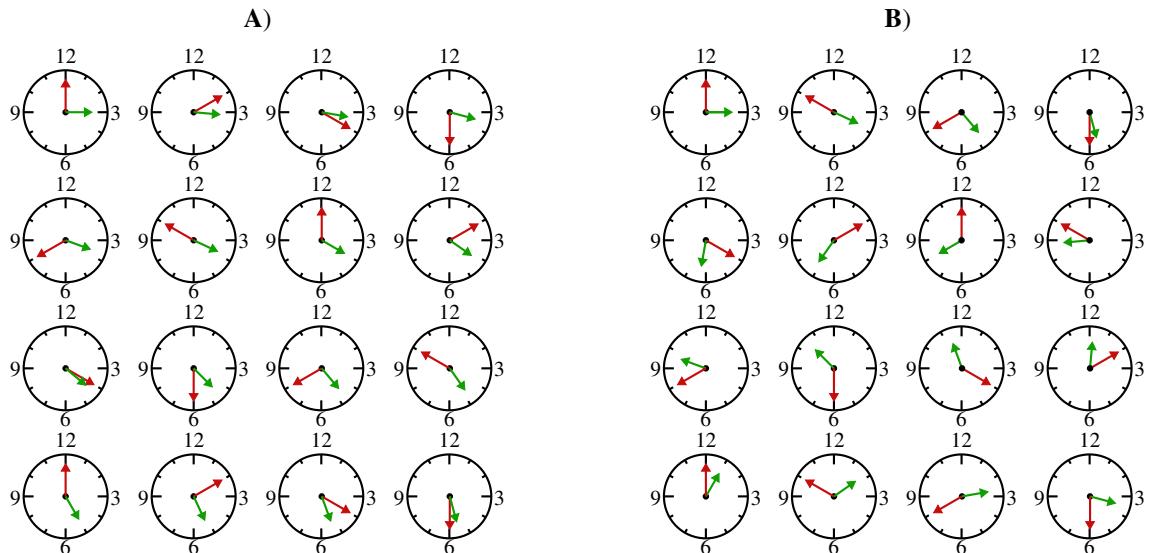
1. Pokud je původní funkce frekvenčně neomezená, tj. neexistuje žádná maximální frekvence, a funkci tudíž nelze v diskrétním rastru reprezentovat přesně.
2. Funkce je sice frekvenčně omezená, tj. existuje $f_{\max} < \infty$, ale funkci vzorkujeme s frekvencí menší než $2f_{\max}$, tedy pod Nyquistovým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



Obr. 2.3: Překrývání obrazů frekvenčně omezené funkce: **A)** převzorkování, **B)** ideální vzorkování a **C)** podvzorkování.

V těchto případech dochází k překrytí spekter ve frekvenční oblasti, čímž ztrácíme vysokofrekvenční informaci. Pokud například budeme snímat hodiny a nezvolíme dostatečné vzorkování (v časové oblasti), budou ručičky vypadat, jako by se pohybovaly pozpátku, viz obr. 2.4. Ve fotografii je tento jev znám jako *Moiré efekt*, a je to tedy jev, kdy na snímku vznikají falešné nízké frekvence. Pro odstranění se používají následující techniky.

- Zvýšení vzorkovací frekvence (to ale nejde vždy).
- Odstranění vysokých frekvencí ještě před vzorkováním, čímž zabráníme překrytí spekter ve frekvenční oblasti a vzniku falešných frekvencí. Je tedy třeba použít nějaký filtr, např. optiku (mírné rozostření).



Obr. 2.4: Ukázka **A)** dostatečného a **B)** nedostatečného vzorkování v časové oblasti.

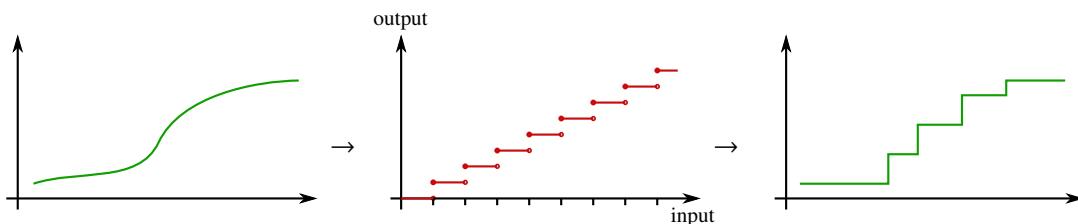
2.2. Kvantování

2.2 Kvantování

Kvantování je zobrazení z \mathbb{R} do množiny $K = \{0, 1, \dots, L - 1\}$. Počet intenzit L se většinou volí 256. Z povahy množin \mathbb{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 prah je většinou určen snímacím zařízením. Dále se používají následující.

- *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.

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.



Obr. 2.5: Ukázka kvantování pomocí ekvidistantního kvantizéru a vzniku falešných kvantizačních hran.

2.3 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 vyzkoušeno, že standardní obrázek velikosti 512×512 se 128 stupni šedi zobrazený na plochu 5×5 cm pozorovaný ze vzdálosti 25 cm se jeví jako **spojitý**.

2.4 Barevné prostory

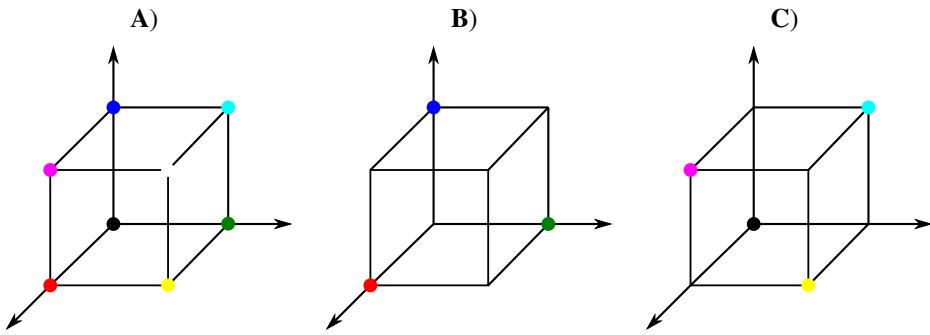
Barevný prostor je předem definovaná množina barev, kterou je schopno určité zařízení snímat, zobrazit nebo reprodukovat. Ve většině případů je založen na barevném modelu, ale na rozdíl od barevného modelu má barevný prostor standardizované odstíny základních barev.

2.4.1 RGB prostory

Tyto barevné prostory jsou založeny na barevném modelu RGB, který se skládá ze tří základních barev: červené (*Red*), zelené (*Green*) a modré (*Blue*). Model RGB využívá aditivního míchání barev a používá se pro zobrazení barev v monitorech, projektorech, televizích atd. Na rozdíl od modelu CMYK nepotřebuje vnější světlo.

2.4.2 CMYK prostory

Tyto barevné prostory jsou založeny na barevném modelu CMYK, který tvoří čtyři základní barvy: azurová (*Cyan*), purpurová (*Magenta*), žlutá (*Yellow*) a černá (*black*). Barevný model CMYK využívá subtraktivní míchání barev a používá se výhradně pro tisk.



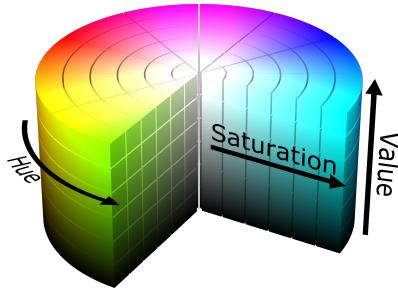
Obr. 2.6: Ukázka barevných prostorů: **A)** základní barvy a jejich kombinace, **B)** RGB prostor, **C)** CMYK prostor.

2.4.3 HSV prostory

HSV (*Hue, Saturation, Value*), také známý jako HSB (*Hue, Saturation, Brightness*), je barevný model, který nejvíce odpovídá lidskému vnímání barev. Skládá se ze tří složek (nejsou to základní barvy), u nichž je nutno hlídat hodnoty, protože může docházet k nesmyslným kombinacím.

- **Hue:** převládající barevný tón (odstín). Měří se jako poloha na standardním barevném kole (0° až 360°). Obecně se odstín označuje názvem barvy.
- **Saturation:** sytost barvy, příměs jiné barvy. Někdy též chroma, síla nebo čistota barvy, představuje množství šedi v poměru k odstínu. Měří se v procentech od 0% (šedá) do 100% (plně sytá barva). Na barevném kole vzrůstá sytost od středu k okrajům.
- **Value:** hodnota jasu, množství bílého světla. Relativní světlost nebo tmavost barvy. Jas vyjadřuje, kolik světla barva odráží, dalo by se také říct přidávání černé do základní barvy. Měří se opět v procentech.

2.5. Otázky



Obr. 2.7: HSV barevný prostor.

2.5 Otázky

- Je dobré mít pravoúhlé vzorkování?

Efektivnější by bylo jiné, třeba hexagonální, aby spektra pokrývali největší plochu a zároveň se nepřekrývala.
Ale většina scannerů a dalších přístrojů má pravoúhlé vzorkování, a to kvůli jednodušší konstrukci.

- Když mám hodně členitou scénu, co je více potřeba, jemnější vzorkování nebo kvantování?

Vzorkování.

- Když mám v obraze hodně velké plochy, scéna není tak členitá, co je více potřeba, jemnější vzorkování nebo kvantování?

Kvantování.

Part I

Předzpracování

3

Metody pro vylepšení vzhledu

3.1 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*.

- **Pixel:** nejmenší (bezrozměrná) jednotka digitální rastrové grafiky, 1 bod obrazu charakterizovaný jasem a barvou.
- **Histogram:** funkce četností jednotlivých intenzit. Lze jej přirovnat k hustotě pravděpodobnosti. Předpokládejme L úrovní jasu a označme n_i , $i \in K := \{0, 1, \dots, L-1\}$ počet pixelů s intenzitou i , tzn. absolutní četnost intenzity i v obraze. Dále označme N celkový počet pixelů. Histogram je pak vektor absolutních četností

$$H = (n_0, n_1, \dots, n_{L-1}).$$

Hustotu pravděpodobnosti intenzity můžeme odhadnout jako

$$f_i = \frac{n_i}{N}.$$

- **Kumulativní histogram:** funkce intenzit, kde pro každou intenzitu je funkční hodnota rovna počtu pixelů, které mají svou intenzitu menší nebo rovnu dané intenzitě, tj.

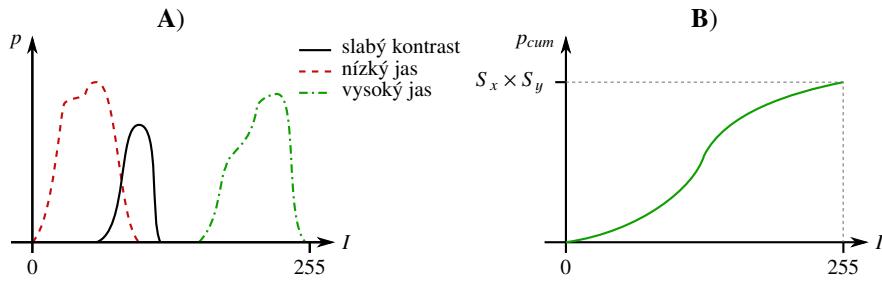
$$H_{cum} = \left(n_0, n_0 + n_1, \dots, \sum_{k \leq i} n_k, \dots, \sum_{k \leq L-1} n_k \right).$$

S použitím odhadu hustoty pravděpodobnosti intenzity z histogramu lze odhadnout distribuční funkci vztahem

$$F_I(i) = \sum_{k \leq i} f_k = \frac{1}{N} \sum_{k \leq i} n_k. \quad (3.1)$$

- **Kontrast:** rozptyl histogramu. Malý kontrast tedy znamená, že rozptyl histogramu je úzký. Pokud chceme změnit kontrast obrázku vynásobíme nebo vydělíme hodnoty intenzit pixelů, tzn. „roztahujeme“ histogram.
- **Jas:** střední hodnota histogramu. Pokud chceme změnit jas, tak musíme přičíst nebo odečíst hodnotu k intenzitě pixelů obrázku, tzn. „posunujeme“ histogram.

3.2. Ekvalizace histogramu



Obr. 3.1: Ukázka A) histogramu a B) kumulativního histogramu. Písmeno I značí intenzitu pixelu, p počet pixelů s danou intenzitou a p_{cum} počet pixelů s danou nebo menší intenzitou. Dále S_x značí šířku obrázku v pixelech a S_y výšku obrázku v pixelech.

3.2 Ekvalizace histogramu

Jde o tzv. vyrovnání histogramu, resp. jako zploštění. Jasy se transformují tak, aby každý z nich „měl pro sebe“ takovou část jasové osy, jaké je jeho zastoupení v obrázku. Jako přechodníková funkce se používá normovaný kumulativní histogram, tedy odhad distribuční funkce (3.1). Dojde-li k tomu, že následující hodnota v histogramu pro danou intenzitu převyšuje požadovanou hodnotu, nebo naopak požadované hodnoty nedosahuje, je potřeba provést štěpení sloupců do více intenzit, případně sloučení sloupců do jedné intenzity. Spojení sloupců problém není, ale jak vysoké sloupce dělit? Používá se následující algoritmus (pro obrazy s intenzitou v rozmezí $(0, 255)$): Označme počet pixelů obrazu I písmenem S . Budeme procházet všechny intenzity postupně od nejmenší po největší, tzn. od černé barvy po bílou. Pro k -tou intenzitu najdeme všechny pixely v původním obrazu s danou intenzitou, označme je n_k , a nastavíme jim novou intenzitu ve tvaru

$$I_{new} = \text{round} \left[255 \cdot \frac{n_k + 2S_k}{2S} \right],$$

kde S_k je počet zpracovaných pixelů z předchozích kroků a 255 značí intenzitu bílé barvy. Někdy se ekvalizuje i lokálně (změna jasu, kontrastu), postup probíhá v okně podobně jako při konvoluci. Ukázka ekvalizace histogramu a porovnání s obyčejným roztažením na všechny možné intenzity je na obr. 3.3.

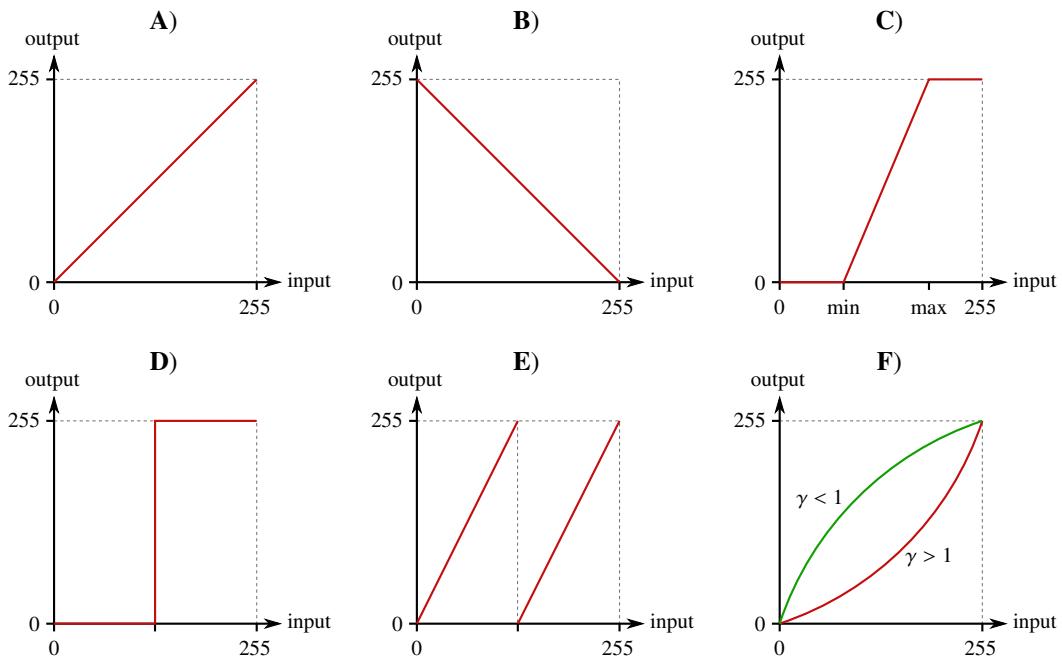
3.3 Šum

Šum v obrázku je přidaná falešná informace nahodilého původu. Může vznikat například při zpracování na senzoru fotoaparátu (prachové částice), nebo při zpracování ve fotoaparátu (kvantizační šum). Abychom si zjednodušili situaci, předpokládejme aditivní model šumu, neboli na signálu nezávislý šum. Modelujeme jej pomocí funkce n

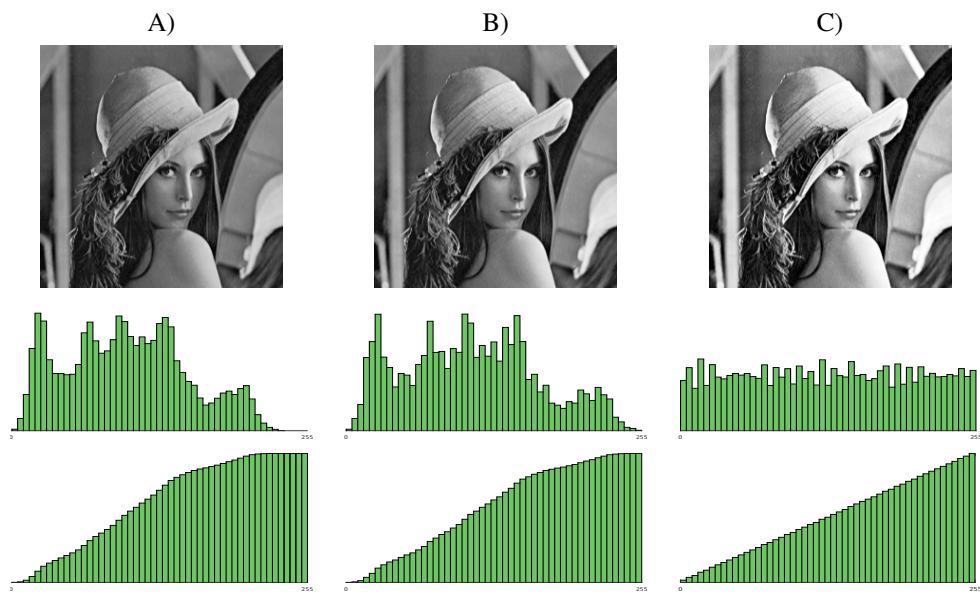
$$f = f_{\text{org}} + n.$$

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

Poznámka 3.1. Existuje i multiplikativní model šumu, tj. $f = f_{\text{org}} \cdot n$.



Obr. 3.2: Ukázka přechodníkových funkcí: **A)** identita, **B)** negativ, **C)** roztažení intenzit na maximální rozsah, tedy zvýšení kontrastu, **D)** binarizace obrazu, **E)** neprostá přechodníková funkce, **F)** gama korekce proti zkreslení zobrazení monitorem ve tvaru $output = c \cdot (input)^\gamma$, $\gamma \in \mathbb{R}^+$.



Obr. 3.3: Ukázka roztažení a ekvalizace histogramu: **A)** originální obraz a jeho histogram, **B)** obraz s roztaženým histogramem a **C)** obraz s ekvalizovaným histogramem. V prvním řádku je obraz, ve druhém jeho odhad hustoty pravděpodobnosti intenzity z histogramu a ve třetím je odhad distribuční funkce intenzity z kumulativního histogramu.

3.3. Šum

3.3.1 Bílý šum (AGWN)

Situaci si zjednodušíme předpokladem bílého šumu (AGWN z anglického *Additive Gaussian White Noise*), což jsou nezávislé, stejně rozdelené (iid) náhodné veličiny X , s normálním rozdelením (gaussovským) a nulovou střední hodnotou, tj. $X \sim \mathcal{N}(0, \sigma^2)$. Analogie s bílým světlem (bílé světlo obsahuje všechny frekvence) plyne z nulové střední hodnoty a z toho, že míra šumu je stejná na všech pixelech a pixel od pixelu nezávislá, tzn. šum jakožto náhodná veličina na jednotlivých pixelech je nekorelovaný s šumem na ostatních pixelech. Protože kovarianční matici jednoznačně zadává normální rozdelení, tak je pro normální veličiny nekorelovanost to samé jako nezávislost.

Spočítjeme *power spektrum* naší šumové funkce. Power spektrum odpovídá dle definice druhé mocnině amplitudového spektra Fourierovy transformace šumové funkce n , tj. $PS = |N|^2$. Pokud provedeme korelací bílého šumu, dostaváme

$$n \otimes n = \sigma_n^2 \cdot \delta \implies \mathcal{F}[n \otimes n] = \sigma_n^2. \quad (3.2)$$

Tento vztah plyne z vlastnosti šumu, tj. z nezávislosti. Z vlastnosti Fourierovy transformace ovšem víme, že platí

$$\mathcal{F}[n \otimes n] = N \cdot N^\star = |N|^2 \implies |N|^2 = \sigma_n^2,$$

kde výslednou rovnost dostaváme díky vztahu (3.2). Vidíme, že power spektrum je rovno rozptylu, a je to tudíž konstantní veličina, což je opět pro bílý šum charakteristické. V praxi by měla při korelací vzniknout matici, která je všude nulová, jen uprostřed je hodnota rozptylu.

3.3.2 Odstranění šumu

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

3.3.2.1 Lineární filtry (konvoluční filtry)

- Průměrování v čase:** lze použít v případě, že je scéna statická, tzn. nehýbe se. Uděláme N nezávislých snímků dané scény g_1, g_2, \dots, g_N , sečteme v jednotlivých pixelech a vydělíme počtem snímků, tzn.

$$g_k = f + n_k \quad \wedge \quad n_k \sim \mathcal{N}(0, \sigma_n^2) \implies g = \frac{1}{N} \sum_{k=1}^N g_k = f + \tilde{n} \implies \tilde{n} \sim \mathcal{N}\left(0, \frac{\sigma_n^2}{N}\right).$$

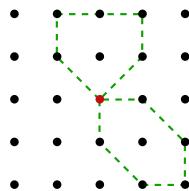
Šum klesá s hodnotou σ^2/N a navíc tato metoda nepřináší žádné degradace.

- Průměrování:** používáme například konvoluční maticí C (průměrování prosté nebo vážené)

$$C = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{nebo} \quad C = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

- Průměrování podél hran:** podél hran používáme jiný druh filtru, v ostatních částech obrazu používáme obyčejné průměrování. Pro použití musíme mít apriorní informaci o hranách a to proto, že detektory hran reagují na šum stejně jako na hrany.

- 4. Metoda rotujícího okna:** v okolí daného bodu definujeme 8 oblastí pro všechny 8 směrů. V každé této oblasti spočítáme rozptyl a vybíráme oblast s nejmenším rozptylem. Ve vybrané oblasti spočítáme průměr a nahradíme jím bod uprostřed masky. Tato metoda se dost používá a dává dobré výsledky. Její výhodou je, že respektuje směr a umístění hran. Nevýhodou je, že vytváří malé oblasti se stejnou intenzitou. Tato charakteristická vlastnost je způsobena principem metody, protože pokud vybereme oblast s nejmenším rozptylem, tak je pravděpodobné, že s posunutím o jeden pixel bude vybrána ta samá oblast, a tedy dosadíme stejný průměr.



Obr. 3.4: Metoda rotujícího okna na okolí bodu velikosti 5×5 .

- 5. Filtry ve frekvenční oblasti:** ve frekvenční oblasti odstraníme nebo utlumíme vysoké frekvence pomocí hladkých *low-pass* filtrů.

Poznámka 3.2. Všechny tyto filtry mohou být kombinovány s prahováním, kdy výsledek měníme pouze pokud konvoluce v bodě překročila zadaný práh.

3.3.2.2 Nelineární filtry

- 1. Mediánový filtr:** v okně provedeme seřazení dat a prostřední prvek (medián) dáváme do výsledku. Tato metoda potlačuje šum, ale „okusuje“ okraje a rohy, např. pokud máme čáru širokou jeden pixel, tak jí mediánový filtr „sežere“. Proto je vhodnější volit za výběrové okno třeba kříž. Navíc pokud je zašuměných více než 50 % pixelů, metoda bere šum jako originální obrázek a šumu po filtrace ještě přibude. Tento filtr je vhodnější na šum typu sůl a pepř. Při použití na bílý šum moc dobře nefunguje.

Náročnost algoritmu pro symetrické okno šířky n (tedy obsahující n^2 prvků) by byla při použití algoritmu *Quicksort* $O(n^4)$. V případě použití algoritmu *Heapsort* je náročnost $O((n \log n)^2)$. Pro další zrychlení lze využít 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.

- 2. Zobecněný mediánový filtr:** v okně provedeme seřazení dat a na setříděnou posloupnost aplikujeme váhovou funkci w rovnou např.

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

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

3.3. Šum

3.3.2.3 Metody zachovávající hrany

Jedná se např. o minimalizaci funkcionálu (funkcionál energie), anizotropní difuzi a splainové metody.

3.3.3 Šum typu sůl a pepř (salt & pepper)

Tento speciální druh šumu vzniká při pořizování obrázku na snímacích zařízeních a je to tzv. impulsní šum. Většina bodů je správně načtená, ale občas některý vypadne nebo se maximálně zesvětlí. Matematický model pro tento šum je ve tvaru

$$f_{i,j} = \begin{cases} I_{\max} & \dots \text{ s pravděpodobností } p, \\ \text{nemění se} & \dots \text{ s pravděpodobností } 1 - p - q, \\ I_{\min} & \dots \text{ s pravděpodobností } q, \end{cases}$$

kde $r = p + q \in \mathcal{U}(0, 1)$ udává míru zašumění, I_{\min}, I_{\max} je minimální a maximální možná intenzita a $f_{i,j}$ je pixel obrazu na pozici (i, j) . Filtr, který tyto nesprávně načtené body opravuje, je opět v podobě konvolučního filtru s maticí tvaru

$$C = \frac{1}{8} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Aplikuje se pouze na tmavé a světlé body obrazu (v ostatních to nemá smysl) a dává uspokojivé výsledky. Lze také použít mediánový filtr. Pokud provádíme opakované snímání (např. sledujeme experiment v laboratoři), používá se i triviální odstranění šumu pomocí průměrování více kopí. Protože na každé kopii je jiná realizace šumu, tak v průměru vymizí. Pokud je možnost takového snímání, pak se jedná o ideální způsob jak šum odstranit.

3.3.4 Periodické poškození obrazu

Pokud se na obraze vyskytuje jisté periodické poškození (např. pohled přes pletivo, mříže, fotka na vroubkovaném papíře), lze tyto „funkce“ z obrazu odstranit. Ve frekvenční oblasti totiž obdržíme amplitudové spektrum s nápadnými symetrickými píky mimo střed. Když je odstraníme, vymizí z obrazu i poškození, viz obr. 3.5.

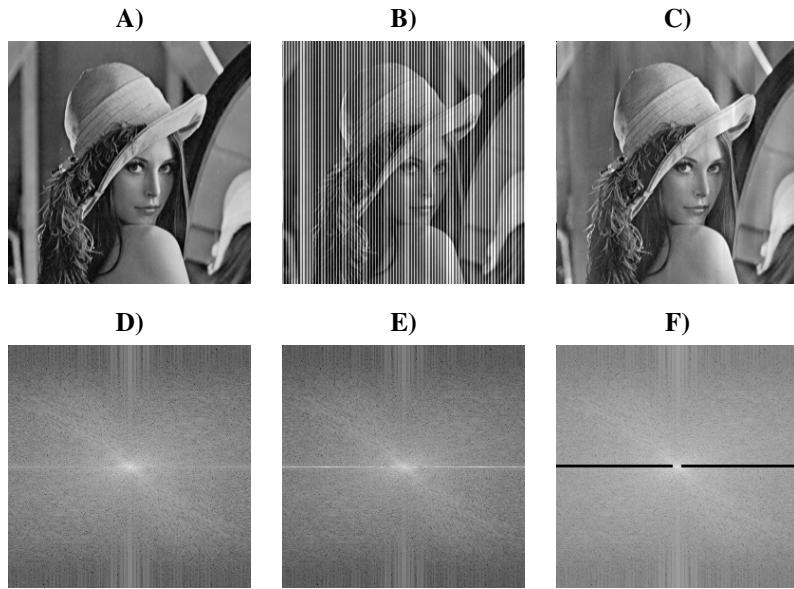
3.3.5 Kvantifikace šumu

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

$$SNR = \frac{|F|^2}{|N|^2} (u, v), \quad (3.3)$$

kde $|N|^2$ a $|F|^2$ se nazývají výkonová spektra (*power spectrum*). Tato funkce zachycuje tu vlastnost, že šum nám vadí hlavně při vysokých frekvencích. Uvažujme velké zjednodušení a předpokládejme, že obrázek není prostorově korelovaný, tj. $|F|^2 = \sigma_f^2$, a bílý šum, tj. $|N|^2 = \sigma_n^2$. Tímto zjednodušením dostáváme vztah

$$SNR = \frac{|F|^2}{|N|^2} (u, v) = \frac{\sigma_f^2}{\sigma_n^2} \quad \forall u, v. \quad (3.4)$$



Obr. 3.5: Periodické poškození: **A)** originální obraz, **B)** obraz s periodickým poškozením a **C)** výsledný restaurovaný obraz a frekvenční spektra: **D)** originálního obrazu, **E)** obrazu s periodickým poškozením a **F)** upravené spektrum obrazu s periodickým poškozením.

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

$$snr = 10 \log_{10} \frac{\text{Var}(f)}{\text{Var}(n)} = 10 \log_{10} \frac{\sigma_f^2}{\sigma_n^2}.$$

Čím vyšší je hodnota odstupu signálu od šumu, tím lepší (méně zašuměný) máme signál. Pro oko je hodnota 15 dB dostačující. V praxi σ_n^2 a σ_f^2 většinou neznáme, takže je odhadujeme jako celek.

3.4 Detekce hran

Nejdříve se již o předzpracování, spíše už se snažíme rozpoznat další neobrazové informace. *Hranou* pro nás bude výrazná změna intenzity obrazu. Experimenty ukazují, že lidské oko se podle hran silně orientuje. Ztráta hranové informace vede ke zmatení a chybám v interpretaci vjemu. Detekce hran se realizuje několika způsoby, které jsou založeny na sledování 1. derivace, resp. 2. derivace funkce intenzity.

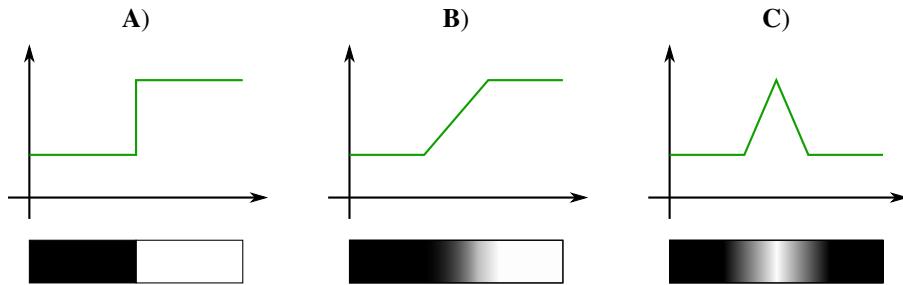
3.4.1 Detektory používající první derivaci

Tyto detektory hledají body, kde má první derivace maximum, tedy $\arg \max_x |\nabla f(x, y)|$. Gradient funkce (obrazu) získáme approximací

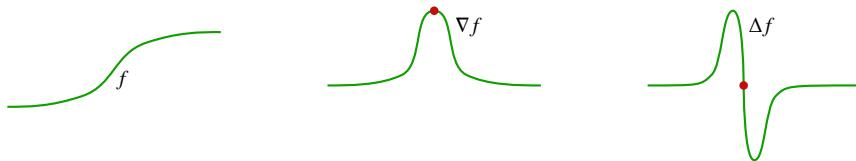
$$\nabla f(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right], \quad \frac{\partial f(x, y)}{\partial x} \sim \frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \sim f(x + 1, y) - f(x, y),$$

což odpovídá konvoluci s jádrem $C = (-1, 1)$.

3.4. Detekce hran



Obr. 3.6: Typy hran pro rozpoznávání v obraze: **A)** ideální hrana (*step edge*), **B)** typická pozvolná hrana (*ramp edge*), **C)** střechovitá hrana (*roof edge*).



Obr. 3.7: Funkce f obsahující hranu a její první a druhá derivace.

3.4.1.1 Robertsův detektor:

Jedná se o konvoluční filtr s maticemi

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

pro všeobecnou detekci hran. Nevýhoda tohoto filtrov je velikost konvolučních masek, díky níž detekuje i šum jako hranu. Tato vada je způsobena tím, že detektor používá derivaci jen v jednom bodě.

Poznámka 3.3. Všeobecnou detekci realizujeme postupnou aplikací všech 8 konvolučních masek, tj. získáním 8 hranových obrazů. Tyto hranové obrazy poté spojíme dohromady za použití maximového pravidla, tzn. bereme vždy největší hodnotu v daném pixelu ze všech 8 hranových obrazů. Pro zlepšení hranové detekce se dále používá prahování. Tento postup všeobecné detekce se používá i pro další konvoluční hranové filtry.

3.4.1.2 Detektor Prewittové :

Jistým vylepšením Robertsova detektora je detektor Prewittové, který pro detekci hran používá konvoluční masky ve tvaru

$$C = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix},$$

a jejich obdobny pro další směry.

3.4.1.3 Sobelův detektor:

Výhoda Sobelova detektora proti Robertsovou detektoru je ta, že počítá derivaci ze tří bodů (z centrálního bodu a bodů okolo). Je tedy robustnější proti šumu. Má celkem 8 konvolučních masek ve tvaru

$$C = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}.$$

Konvoluční masky pro zbylých 6 směrů vznikají rotací a vypadají obdobně.

Poznámka 3.4. Pokud nejprve obrázek vyhladíme (rozmažeme), a pak na něj aplikujeme Sobelův detektor, dostaneme hranový obraz, který by vznikl aplikací Robertsova detektora na větší okolí, tzn. žádný lepší výsledek se nedostaví

$$C_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad C_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \implies C_3 = C_1 \circ C_2 = \begin{pmatrix} - & - & - \\ 0 & 0 & 0 \\ + & + & + \end{pmatrix},$$

$$(f * C_1) * C_2 = f * C_3.$$

3.4.1.4 Kirschův detektor:

Podobně jako Sobelův detektor tento detektor pracuje s 8 konvolučními maskami, které jsou ve tvaru

$$C = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}, \quad C = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}.$$

Konvoluční masky pro zbylých 6 směrů vznikají rotací a vypadají obdobně. Vše směrová detekce probíhá stejně jako u Sobelova detektoru.

3.4.1.5 Cannyho detektor:

Používá se velmi často. Byl vytvořen s požadavky aby detekoval všechny hrany, každou hranu detekoval právě jednou a přesně ji lokalizoval (střed hrany) a aby nevytvářel hrany navíc. Princip Cannyho detektora je následující:

1. Odstranění šumu vyhlazením pomocí konvoluce s Gaussovým jádrem.
2. Výpočet parciálních derivací, stačí i jednoduchý detektor, např. *Roberts Cross function*

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

Dále se provede „non-maximum suppression“. Úkolem této procedury je vybrat z hodnot gradientů (stanovených v předchozím kroku) jen lokální maxima. Respektive odebrat body, které nejsou maximem („detekce hřbetů“), čímž zajistíme, že hrana bude detekována v místě největšího gradientu.

3.4. Detekce hran

3. Prahování s hysterezí (*Double threshold*). V předchozím kroku jsme určili, kde přesně leží hrany, ale dosud jsme se nezabývali jejich významem. V tuto chvíli jsou označeny i ty nejmenší hrany, protože i ty mají své lokální maximum. Zvolíme minimální t_{min} a maximální t_{max} práh, mezi kterými může gradient kolísat. Pokud hodnota gradientu daného pixelu leží nad větším prahem t_{max} je označen jako *silný*. Pokud posuzujeme bod, jehož hodnota leží mezi t_{min} a t_{max} , je označen jako *slaby*. Silné pixely zachováváme vždy a slabé zachováváme, pokud sousedí se silným pixelem.

Cannyho detektor je tvořen sadou odvozených filtrů, které závisí na parametrech. Nastavením parametrů lze docílit výborných výsledků.

3.4.2 Detektory používající druhou derivaci

Tyto detektory hledají body, kde má druhá derivace průchod nulou. Pro druhou derivaci platí

$$\frac{\partial}{\partial x^2} = (-1, 1) * (-1, 1) = (1, -2, 1)$$

z čehož lze odvodit tvar Laplaceova operátoru

$$\Delta = \frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

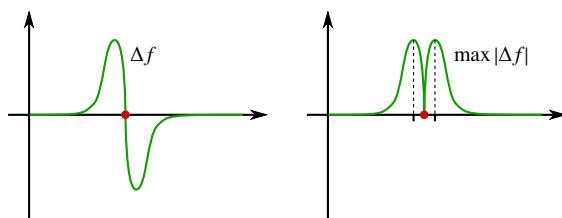
Neřešíme tedy rovnici $\Delta f = 0$, ale hledáme, zda se v Δf objevují velké přechody mezi kladnou a zápornou hodnotou (tzv. *zero-crossing points*). V těchto místech pak zaznamenáme hrany. Tento způsob je velmi náchylný na šum (dokonce víc než Robertsův detektor).

3.4.2.1 Marr Hildrethův detektor (LoG):

Marr a Hildreth navrhli modifikaci, při které se sleduje funkce

$$\Delta(G * f) = \Delta G * f,$$

kde G je kruhově symetrická Gaussova funkce. Matice ΔG se předpočítává a bývá široká 5×5 až 15×15 pixelů. Parametrisací 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 $\Delta f = 0$. Pro lokalizaci hrany se často používá místo složitějšího hledání „průchodu nulou“ jednoduché nalezení $\max |\Delta f|$, viz obr. 3.8.



Obr. 3.8: Hledání *zero-crossing point* pomocí $\max |\Delta f|$.

3.4.3 Detektory nepoužívající derivace

3.4.3.1 Whitening

Tato metoda spočívá v tom, že amplitudy Fourierovy transformace nahradíme konstantní funkcí. Zůstává nám tím harmonická informace. Poté provedeme konvoluci s maskou C , a tím docílíme, že hrany zůstanou jen v určitém směru. Budeme ztráct informaci o hranách s větší šířkou.

3.4.4 Detektory pracující ve frekvenční oblasti

Chceme-li ve frekvenční oblasti detekovat hrany pod určitým úhlem, musíme se ve spektru signálu dívat ve směru kolmém na tento úhel. Po převedení do frekvenční oblasti lze využít směrových filtrů, viz obr. 1.8.

3.4.5 Zvýraznění hran

Abychom při hledání hran dosahovali lepších výsledků, můžeme před samotnou detekcí hrany zvýraznit.

- Jednou z možností je použití konvoluce. Celkem potřebujeme masky pro 4 směry, které jsou ve tvaru

$$C = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{pmatrix}, \quad C = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix},$$

a pro zbylé 2 směry obdobně. Po konvoluci se ještě provádí prahování maximálních hodnot.

- Další možností je použití Laplaceova operátoru Δ (odvození tvaru Laplaceova operátoru viz sekce 3.4.2). Hrany zvýrazníme odečtením Laplaceova operátoru od funkce $f - \Delta f$. Při použití approximace to odpovídá konvoluci s maskou C , tzn.

$$f - \Delta f \implies \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \implies C = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

- Poslední metodou jak zvýraznit hrany je tzv. neostré maskování (*unsharp masking*). Nejdříve odečteme od originálního signálu vyhlazený signál f_{smooth} pomocí *low-pass* filtru čímž získáme *high-pass* reprezentaci f_{high} . Pokud *high-pass* reprezentaci přičteme k původnímu signálu, dostáváme „zaostřený“ signál f_{sharp}

$$f_{high}(x, y) = f(x, y) - f_{smooth}(x, y),$$

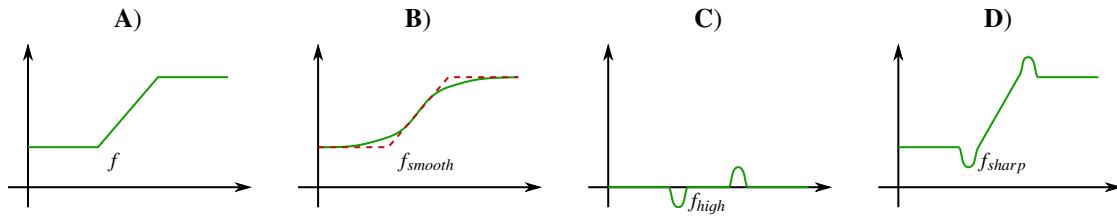
$$f_{sharp}(x, y) = f(x, y) + k \cdot f_{high}(x, y),$$

kde k je škálovací konstanta. Na obr. 3.9 je zobrazen princip neostrého maskování.

3.5 Houghova transformace

Jedná se o metodu nalezení parametrického popisu objektů v obraze. Abychom metodu mohli použít, je třeba znát analytický popis hledaných objektů. Je to tedy vhodná metoda na detekci jednoduchých objektů, jako např. přímka, kruh, elipsa, případně objektů, jejichž hranici lze popsat jednoduchými křivkami.

3.5. Houghova transformace



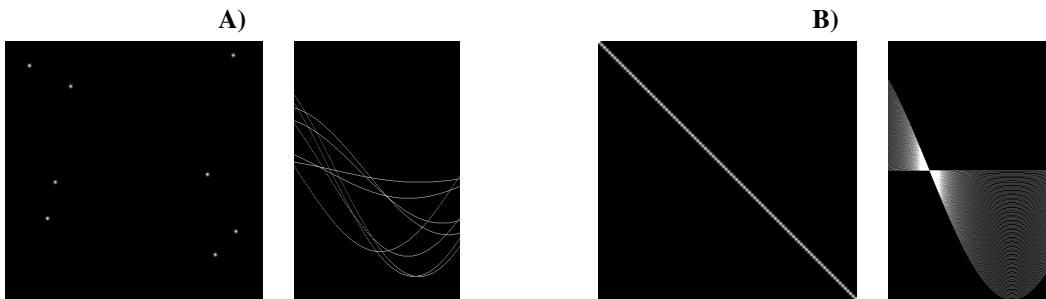
Obr. 3.9: Princip neostrého maskování: **A)** originální signál, **B)** vyhlazený signál, **C)** *high-pass* reprezentaci, **D)** „zaostřený“ signál.

Každá přímka lze popsát pomocí úhlu θ , který svírá kolmice na přímku procházející středem souřadné soustavy s osou souřadné soustavy, a vzdálenosti r přímky od počátku souřadné soustavy. Z tohoto popisu lze odvodit vztah

$$r = x \cdot \cos \theta + y \cdot \sin \theta.$$

Máme tedy vztah, kterým můžeme z klasické souřadné soustavy přejít do soustavy tvořené dvojicí proměnných r a θ , která se nazývá Houghův prostor. Pokud provedeme tuto transformaci, tak se nám přímka promítne na bod. Bod se promítne na křivku, která reprezentuje všechny přímky procházející daným bodem v klasické souřadné soustavě. Průsečíky křivek v Houghově prostoru reprezentují přímky. Fungování Houghovy transformace je znázorněno na obr. 3.10.

Pokud tedy chceme hledat hrany pomocí této transformace, tak provedeme transformaci pro celý obraz a v Houghově prostoru hledáme průsečíky, které reprezentují přímku. Většinou nelze kvůli deformacím nalézt přímo jednoznačný průsečík, a tak hledáme shluky bodů. To je také největší problém metody.



Obr. 3.10: Houghova transformace působící na body A), resp. přímku B).

4

Restaurace obrazu

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

$$\mathcal{D} = \mathcal{T}_G \circ \mathcal{T}_I,$$

kde \mathcal{T}_G je *geometrická degradace* a \mathcal{T}_I je *radiometrická degradace*. Výsledný model lze zapsat ve tvaru

$$g = \mathcal{T}_G \circ \mathcal{T}_I(f) + n.$$

4.1 Radiometrické zkreslení (radiometrický inverzní problém)

Pokud nepředpokládáme geometrické zkreslení a uvážíme, že ve většině běžně dostupných zobrazovacích systémů lze \mathcal{T}_I modelovat pomocí konvoluce, můžeme vztah přepsat na tvar

$$g = f * h + n. \quad (4.1)$$

Funkce h se zde označuje jako *impulsní odezva* a jedná se o tzv. *point spread function* (polohově invariantní funkci).¹ Polohová invariance značí skutečnost, že výsledek vztahu ovlivňující dva pixely není závislý na jejich poloze, ale pouze na jejich vzájemné vzdálenosti.²

4.1.1 Inverzní filtr

Rovnici (4.1) můžeme převést do frekvenční oblasti Fourierovou transformací a dostaneme

$$G = F \cdot H + N \implies F = \frac{G - N}{H} = \frac{G}{H} - \frac{N}{H}, \quad (4.2)$$

kde dělení 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. Funkce H se označuje jako *přenosová funkce*. Pokud známe h , lze navrhnout *inverzní filtr* ve frekvenční oblasti a po inverzní

¹Lze ji získat třeba vyfocením bodu: $\delta * h = h \implies$ výstup je *point spread function*.

²Toto je příliš velké omezení pro 3D scény, kde se rozmažání mění s hloubkou ostrosti.

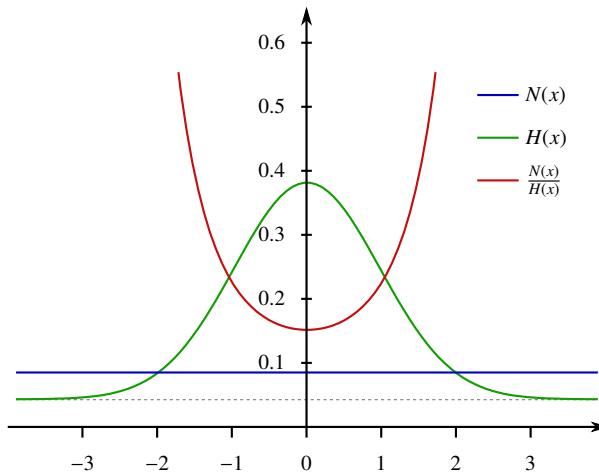
4.1. Radiometrické zkreslení (radiometrický inverzní problém)

Fourierově transformaci dostáváme odhad originálu následovně

$$\hat{f} = \mathcal{F}^{-1}[F] = \mathcal{F}^{-1}\left[\frac{G}{H} - \frac{N}{H}\right].$$

V modelu bez šumu zanedbáváme ve vztahu (4.2) člen $-N/H$ a v takovém případě inverzní filtr funguje velmi dobře.

Pro model se šumem může mít ovšem člen $-N/H$ velmi zásadní vliv na výsledek, a to i pro malá n , jak ukazuje graf na obr. 4.1. Z tohoto důvodu je volba inverzního filtru pro model se šumem nepoužitelná.



Obr. 4.1: Hodnoty funkcí $H(x)$, $N(x)$ a jejich podílu.

4.1.2 Wienerův filtr

Místo inverzního filtru se pro zašuměný obrázek používá filtr Wienerův. Požadavkem na Wienerův filtr W bylo, aby byl lineární, tzn. ve tvaru

$$\hat{F} = G \cdot W, \quad (4.3)$$

a získaný odhad originálního obrazu byl co nejlepší. Předpis pro matici W dostaneme řešením minimalizační úlohy

$$W = \arg \min_{\hat{f}} \mathbb{E} [\|f - \hat{f}\|^2], \quad (4.4)$$

kde f je originální obraz a \hat{f} je odhad obrazu po rekonstrukci. Minimalizaci provádíme přes všechny **lineární filtry**, a střední hodnota je brána přes všechny možné realizace šumu. Matice W není přesně daná, je to funkce obrazu! Po netriviálním odvození dostaneme výslednou matici W ve tvaru

$$W = \frac{1}{H} \cdot \frac{|H|^2}{|H|^2 + SNR^{-1}(u, v)},$$

kde funkce $SNR(u, v)$ značí *signal-to-noise ratio*, viz (3.3). Druhý člen v uvedeném součinu je tzv. korekční člen, který potlačuje frekvence, kde převládá šum. Pokud neznáme $|F|$ a $|N|$ (většinou neznámé), můžeme zkoušet uvažovat bílý šum a nekorelovanost původního obrazu, viz (3.4). V takovém případě tedy máme SNR konstantní ve tvaru

$$SNR(u, v) = \frac{\sigma_f^2}{\sigma_n^2} = \text{const} \quad \forall u, v.$$

Dále z nekorelovanosti původního obrazu platí

$$f \otimes f = \delta(x) \sigma_f^2 = \begin{cases} \sigma_f^2 \cdot \delta(0) & \dots \text{ pokud leží přesně na sobě,} \\ 0 & \dots \text{ jinde.} \end{cases}$$

Při odhadování SNR postupujeme tak, že s hodnotou SNR začínáme na nízké úrovni $\sim 10^{-3}$ a postupně ji zvyšujeme třeba do $\sim 10^3$ a sledujeme, co se děje s obrazem. Když Wienerův filtr aplikujeme na nezašuměný obrázek (tedy $|N|^2 = 0$), pak matici W přechází do tvaru $W = 1/H$ a celkem získáváme jednoduchý inverzní filtr.

Pokud SNR při rekonstrukci nadhodnotíme, tak dojde k zvýraznění šumu. V případě podhodnocení SNR dojde k rozostření obrazu, protože potlačíme více vyšších frekvencí.

4.1.3 Základní typy radiometrických degradací

Nyní se budeme zabývat odhadem přenosové funkce H pro Wienerův filtr. Impulsní odezva h je neznámá a navíc může být ve tvaru $h = h_1 * h_2 * h_3 * \dots$. Obecně impulsní odezvu nelze zjistit. Můžeme ji pouze předpokládat ve speciálních tvarech. Reálně rozlišujeme tři základní druhy degradace: rozmazání pohybem, špatné zaostření a turbulence. Obvykle jsou zastoupeny všechny základní degradace najednou, ale jedna výrazně převládá, a ostatní tedy můžeme zanedbat.

4.1.3.1 Rozmazání pohybem

Je-li pohyb lineární (tj. po přímce), má impulsní odezva tvar obdélníka. Předpokládejme konstantní rychlosť pohybu, jinak je situace obtížná, protože se vytrácí polohová invariance. V takovém případě je délka obdélníku úměrná délce expozice a šířka je převrácená hodnota délky, abychom zachovali $\int h = 1$ a nevnášeli tak do obrazu jas. Ve směru kolmém na směr pohybu se tento obdélník chová jako δ -funkce, a tedy nic nemění. Po Fourierově transformaci obdržíme ve frekvenční oblasti známou funkci $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$, natočenou do směru pohybu. Slangově se tato funkce označuje jako *vlnitý plech*, viz obr. 1.5 a obr. 1.6.

Nulové hodnoty funkce H budou ležet na přímkách kolmých na směr pohybu, viz obr. 4.2. Čím blíže budou nulové body k sobě, tím je rozmazání větší. Stejně nulové body budou zachovány i ve funkci G . Můžeme tak najít směr a délku pohybu (během expozice).

Podobně lze odhadovat impulsní odezvy i 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.

4.1.3.2 Špatné zaostření (defokusace)

Ideálním rozostřením světelného bodu je kruh. Ve skutečnosti ovšem klesá jeho intenzita ke krajům. Impulsní odezva je tvaru válce a přenosová funkce je ve tvaru

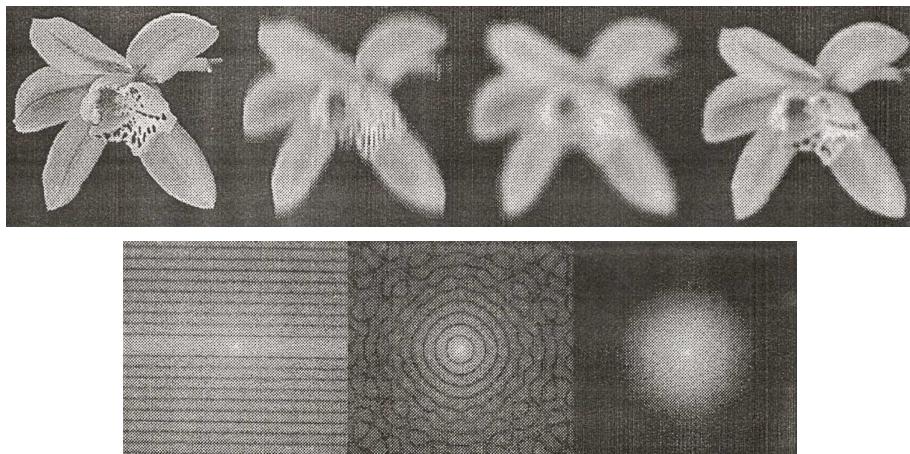
$$H \sim \frac{1}{r} J_1(r), \quad (4.5)$$

kde $J_1(r)$ je Besselova funkce 1. druhu a r je radiální souřadnice, viz obr. 1.5 a obr. 1.6. Díky výrazu $1/r$ ve vztahu (4.5) jsou Besselovy kmity tlumené. Nulové body přenosové funkce jsou na soustředných kružnicích, viz obr. 4.2. Stejně nulové body budou zachovány i ve funkci G . Postup je obdobný jako v případě lineárního posunutí. Změříme-li vzdálenost kružnic, můžeme odhadnout impulsní odezvu.

4.2. Geometrické zkreslení (geometrický inverzní problém)

4.1.3.3 Turbulence

Vzniká např. při snímání přes tlustou vrstvu atmosféry s delší expoziční dobou, než je perioda Brownova pohybu častic v atmosféře. Impulsní odezva, a tedy i přenosová funkce, jsou Gaussovy funkce, viz obr. 1.5 a obr. 1.6. 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).



Obr. 4.2: Základní typy degradací. Nahoře zleva původní obrázek, obrázek rozmazený pohybem, defokusací a turbulencí. Dole FT příslušné impulzní odezvy.

4.1.4 Problém slepé dekonvoluce

V obecném případě pro neznámé h nemá úloha jednoznačné řešení. Zavádíme proto dodatečné omezující podmínky, které nám řešení zjednoduší, např. nezápornost originálního signálu (snímku), omezenost nosiče h a minimalizace jistého potenciálového funkcionálu výsledného obrazu.

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

$$g_1 = f * h_1 + n_1,$$

⋮

$$g_k = f * h_k + n_k.$$

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

4.2 Geometrické zkreslení (geometrický inverzní problém)

Model geometrického zkreslení obrazu lze zapsat jako

$$g = T_G(f).$$

Máme-li snímek též scény z různých pohledů (tj. s jiným geometrickým zkreslením) a potřebujeme-li zjistit odpovídající pixely (tj. aby stejné pixely měly stejně 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. Rozlišujeme čtyři základní kategorie registrace obrazu.

- Vícepohledová (*different viewpoints, multi-view*).
- Vícečasová (*different times, multi-temporal*).
- Multimodální (*different modalities, multi-modal*).
- Napasování scény na model registrace (*scene to model registration*).

Registrace se provádí pomocí řídících bodů (*control points*). Pokud jsou správně nalezeny odpovídající řídící body v obou obrazech, je možné sestavit geometrické transformace a snímky registrovat. Postup registrace lze rozdělit do čtyř kroků, kterým se věnuje další část textu.

4.2.1 Výběr kandidátů na řídící body (*Control point selection*)

Na řídící body klademe základní požadavky a to, aby jich byl dostatečný počet, byly rozmístěny po celém obrazu, aby byly dobře automaticky detektovatelné v obou obrazech a aby byly invariantní vůči transformaci. Jako významné řídící body se volí *rohy, průsečíky* (citlivé na šum), *těžiště uzavřených oblastí* (na šum více stabilní), nebo *extrémní křivosti křivek*. V prvním kroku registrace se řídící body vybírají zvlášť na referenčním obrazu a na registrovaném obrazu. Ve druhém kroku je poté třeba provést vzájemnou korespondenci. K tomuto účelu musíme mít dostatečně robustní, spolehlivou metodu, neboť některé body mohou chybět, mohlo dojít k nepřesnostem atp.

4.2.2 Korespondence řídících bodů (*Control point matching*)

Jak už bylo řečeno, druhým krokem výběru řídících bodů, je nalezení vzájemné korespondence mezi kandidáty na řídící body v referenčním a registrovaném obrazu. Na základě korespondence kandidátů poté volíme vhodné řídící body. Hledání odpovídajících si bodů se provádí buď topologicky (předpokládaný výskyt), nebo se prohledává okolí bodu s ohledem na intenzity (barvy) okolních pixelů. Rozlišujeme dva základní druhy metod.

4.2.2.1 Signálově závislé metody

Tyto metody pracují s intenzitou obrázku.

1. **Obrazová korelace:** korelací provádíme se skutečnými obrazovými hodnotami. Jeden obraz (okno) posouváme po druhém a počítáme korelací. Pokud existuje větší či menší shoda obrazů, bude na daném bodě ve výsledné matici vysoká hodnota. Při konstantní korelační matici pak tento postup odpovídá počítání dle vztahu

$$C(X, Y) = \frac{\mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)]}{\sqrt{\text{Var}(X)} \cdot \sqrt{\text{Var}(Y)}} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - \bar{X})(Y_{ij} - \bar{Y})}{\sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - \bar{X})^2} \cdot \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (Y_{ij} - \bar{Y})^2}}.$$

V této podobě se metoda také nepoužívá, protože je výpočetně a časově náročná a maximum bývá někdy „ploché“. Urychlení výpočetní složitosti plyne z *Korelačního teoremu*

$$\mathcal{F}[f \otimes g] = F \cdot G^*$$

4.2. Geometrické zkreslení (geometrický inverzní problém)

kde symbol $(\cdot)^*$ značí komplexně sdruženou matici. Dále lze provádět modifikace: korelace hran, rohů, korelace ve frekvenční oblasti (fázová korelace), pyramidální reprezentace (*pyramidal representation*).

Metoda funguje dobře pro obrazy se stejnou intenzitou nebo pokud došlo ke globálním změnám kontrastu a jasu. 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 veliká, proto se tato modifikace používá pouze pro malé rotace.

U této metody se většinou nedetekují řídící body v druhém obrázku, ale hledají se nejvyšší korelace vzhledem k řídícím bodům prvního obrázku. Nejčastěji hledáme malý výrez na velkém obrázku. Hodnoty intenzit se liší pouze lineárně. Metoda dává dobré výsledky po hranové detekci v obou obrazech.

2. **Fázová korelace:** Fáze Fourierovy transformace je blízká hranám obrázků. Ty jsou výhodnější kvůli nezávislosti na barvách a mají menší prostorovou korelaci. Nevrací se do obrazové oblasti pro počítání korelace, ale zůstane se ve frekvenční oblasti, kde se využívá *Fourier Shift Theorem* (FST), viz věta 1.9. Předpokládáme-li, že první a druhý obrázek jsou stejné, ale jen posunuté, máme dány funkce $f(\vec{x})$ a $g(\vec{x}) = f(\vec{x} - \vec{d})$, kde $\vec{x}, \vec{d} \in \mathbb{R}^2$. Po Fourierově transformaci získáváme vztahy

$$f(\vec{x}) \xrightarrow{\mathcal{F}} F(\vec{u}), \quad g(\vec{x}) \xrightarrow{\mathcal{F}} e^{-2\pi i \vec{u} \cdot \vec{d}} F(\vec{u}),$$

kde $\vec{u} \cdot \vec{d}$ značí standardní skalární součin. Amplituda je u obou Fourierových obrazů stejná, ale fáze se změnila známým způsobem. Definujeme tzv. *Cross-power* spektrum jako

$$\frac{F \cdot G^*}{|F||G|} = \frac{|F|^2 e^{2\pi i \vec{d} \cdot \vec{u}}}{|F|^2} \xrightarrow{\mathcal{F}^{-1}} \delta(\vec{x} - \vec{d}).$$

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

3. **Nepodobnost (SSDA):** označuje se také jako „klasická korelace“. Jde o použití jiné míry podobnosti než korelace, tzn. ne L_2 norma, ale L_1 norma, např. $\sum |f_{ij} - g_{ij}|$. Hledáme bod, kde norma nabývá minima. Výpočet je velmi rychlý, protože suma je neklesající, jakmile tedy jednou částečný součet převýší již vypočtené minimum, lze přímo přejít k dalšímu bodu.

Poznámka 4.1. Pro snadnou hardwarovou implementaci korelace jsou tyto metody (Fázová korelace a nepodobnost (SSDA)) dosti používané.

Poznámka 4.2. Lze rozšířit na obecnější transformace, např. pro rotaci je okénko kruh. V takovém případě se ale stává úloha výpočetně velice náročná. Dá se nejdříve projet prostým posunutím tam, kde je nalezeno maximum, tak to začneme natáčet. Vybereme úhel natočení, kde je korelace maximální, a poté projdeme opět celý obrázek s tímto natočením.

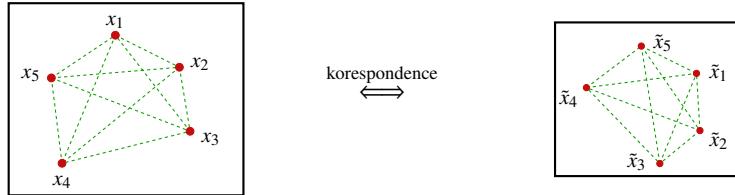
4. **Pyramidální reprezentace:** Snižujeme rozlišení obrázků o dvojnásobek. Začínáme na nízkém rozlišení, kde najdeme „nadějně body“ a u vyššího rozlišení počítáme jen v okolí těchto bodů.

4.2.2.2 Příznakové metody (signálově nezávislé) (???)

Jde o napočítávání vlastností (*příznaků*) segmentovaných objektů obrazu. Příznaky by měly být invariantní k transformaci, kterou předpokládáme. Více o příznacích viz kapitola 6.

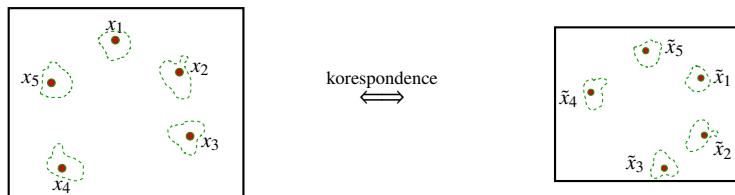
1. **Point-based method:** pro zjištění korespondence řídících bodů se vytváří úplný graf a hledají se 4 parametry pro geometrickou transformaci (2×posun, 1×rotace, 1×měřítko). Porovnáváme po dvojici hrany grafu

a zjišťujeme, jak ostatní body padají do okolí řídících bodů. Další možností je provést matching a zaznamenávat do parametrického prostoru provedenou transformaci. Pak hledat shluv 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.



Obr. 4.3: Zjištění vzájemné korespondence řídících bodů (*matching*) pomocí úplného grafu.

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



Obr. 4.4: Určování okolí bodů a jejich charakteristik (*Region-based method*).

3. **Kombinatorické:** Využívá globální informace o kandidátních bodech a jejich vzájemné polohy, z níž hledá jejich korespondenci. Zkouší všechny možné kombinace a hledá tu nejlepší. Jde o minimalizaci funkce.
4. **RST (rotation, scaling, translation):** Libovolnou úsečku můžeme namapovat na libovolnou úsečku. V základním případě se každá dvojice bodů namapuje na každou dvojici z druhého obrázku. Namapujeme úsečku a podle ní transformujeme ostatní body a koukáme se, kolik bodů se shoduje se vzory, tzn. počítáme „zásahy“. Pokud toto uděláme pro všechny dvojice bodů, tak transformace, která má nejvíce zásahů, je hledaná transformace.

4.2.3 Odhadnutí modelu transformace souřadnic

Při vlastním mapování hledáme tzv. mapovací funkce

$$\begin{aligned} u &= f(x, y), & \text{s podmínkou } \mathcal{P}_1: u_i = f(x_i, y_i), \\ v &= g(x, y), & v_i = g(x_i, y_i), \quad \forall i \in \hat{m}, \end{aligned}$$

kde m je počet řídících bodů. Funkce f, g předpokládáme spojité, dost hladké a parametrizovatelné (např. afinní transformace). Parametry funkcí hledáme řešením následujících úloh:

- *interpolační úloha* – s podmínkou \mathcal{P}_1 ,

4.2. Geometrické zkreslení (geometrický inverzní problém)

- *extrapolační úloha* – bez podmínky \mathcal{P}_1 .

Po nalezení mapovacích funkcí můžeme obrazy namapovat bod po bodu. Pro mapování chceme použít transformaci, která promítne řídící body referenčního obrazu na odpovídající řídící body na obrazu registrovaném. Transformační funkce mají obecný tvar

$$\tilde{x} = T_x(x), \quad \tilde{y} = T_y(y),$$

a mohou platit pro celý obrázek (globální transformace), ale jednotlivé dílčí části můžou mít i odlišné transformace (lokální). Většinou se transformace předpokládá jako jedna z následujících.

1. **Afinní transformace:** affinní model je jeden z nejjednodušších, ale přesto se hojně používá. Pro Euklidův prostor zahrnuje posunutí, otáčení, změnu měřítka, zkosení, zrcadlení, a jejich skládání. Transformační funkce má tvar

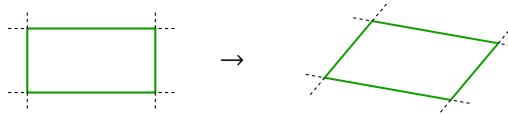
$$\tilde{x} = a_0 + a_1x + a_2y, \quad \tilde{y} = b_0 + b_1x + b_2y. \quad (4.6)$$

Tato transformace zobrazuje čtverec na rovnoběžník, převádí přímky na přímky (nebo bod) a zachovává jejich rovnoběžnost. Obecněji převádí affinní podprostory na affinní podprostory. Pro její určení jsou potřeba tři body, ale v praxi se počítá z mnohem více bodů pomocí metody nejmenších čtverců. Lze zapsat ve tvaru

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \mathbb{T} \begin{pmatrix} x \\ y \end{pmatrix} + \vec{p},$$

kde \mathbb{T} jsou koeficienty transformace bez posunu a \vec{p} jsou koeficienty posunu. Tyto matice vyjádřené pomocí vztahů (4.6) mají tvar

$$\mathbb{T} = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}, \quad \vec{p} = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix}.$$



Obr. 4.5: Ukázka působení affinní transformace na obdélník.

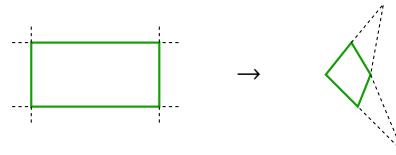
2. **Perspektivní projekce:** perspektivní projekce je obecnější transformace. Má však tu vlastnost, že při pozorování z větších vzdáleností v afinitu přechází, neboť c_1 a c_2 v transformačních funkciích

$$\tilde{x} = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y}, \quad \tilde{y} = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y},$$

budou malé. V praxi se při transformacích nepoužívá, protože nevede na lineární soustavu a nejde ji „rozumně“ řešit. Tato transformace zobrazuje čtverce na jakýkoli čtyřúhelník a vystihuje promítání roviných předmětů fotoaparátem. Pro její určení jsou potřeba čtyři body.

3. **Nelineární transformace:** tento model se používá velmi často. Je silně nelineární a nezachovává přímky ani rovnoběžnost. Transformační funkce má tvar

$$\tilde{x} = a_0 + a_1x + a_2y + a_3xy \left[+a_4x^2 + a_5y^2 \right], \quad \tilde{y} = b_0 + b_1x + b_2y + b_3xy \left[+b_4x^2 + b_5y^2 \right].$$



Obr. 4.6: Ukázka působení perspektivní projekce na obdélník.

4.2.4 Vlastní transformace

Vlastní mapování zabírá nejvíce výpočetního času. Musí se převzorkovávat, protože nové pixely mají neceločíselné souřadnice. Používají se přímé a nepřímé transformace.

- **Dopředná (Forward):** 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.
- **Zpětná (Backward):** procházíme všechny body výsledného obrazu a pomocí zpětné transformace určíme, který zdrojový bod se na ně transformuje.

4.2.4.1 Lokální transformace

Obrázek rozdělíme na trojúhelníky pomocí řídících bodů a na každém trojúhelníku počítáme affinní transformaci. Nemá spojité derivace a řeší se pomocí kubické transformace s 10 parametry, kde se předepří spojitosti derivací. Nejčastěji se používá TPS (*Thin-Plate-Splines*), kdy se hledá minimální křivost plochy ideálně neformovatelného ocelového plátu fixovaného v řídících bodech.

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.

4.3 Otázky

- Přes co probíhá minimalizace a přes co střední hodnota v odvození Wienerova filtru?

Minimalizace probíhá přes všechny lineární filtry a střední hodnota přes šum.

- Co znamená že je obraz f nekorelovaný?

Korelací se myslí korelace prostorová, tedy že hodnoty pixelů na různých pozicích jsou nekorelované

- Co se stane, pokud hodnotu SNR ve Wienerově filtru podhodnotíme, a co když nadhodnotíme?

Je nutné si rozmyslet limitní případy, tedy $SNR \rightarrow 0$ a $SNR \rightarrow \infty$. Pokud SNR nadhodnotíme, objeví se na obrázku artefakty vysokofrekvenčního šumu. Vysoké frekvence šumu způsobují výraznější poškození v inverzním filtru, a tedy i ve Wienerově filtru s podhodnoceným SNR. Pokud naopak SNR podhodnotíme, obrázek ztrácí detaily a bude rozmazený.

Part II

Rozpoznávání dat (*pattern recognition*)

Máme n tříd a chceme rozhodnout o objektu O , do jaké třídy patří. Existují dva typy metod: strukturální (syntaktické) a příznakové (statistické)

- **Strukturální rozpoznávání:** Používá se v případě, kdy je struktura důležitější než měřitelné vlastnosti objektu, např. dům, střecha, zdi, dveře, okna, atd. Není vhodné na jednoduché úlohy (zbytečně složité). Používá se třeba při klasifikaci složitých scén, např. osoba lze špatně popsat pomocí příznaků (různé výrazy tváře atd.), ale lze popsat strukturně. Dále se používá při klasifikaci EKG, EEG křivek, kde jde o strukturu výchylek a nikoliv o jejich velikost. Založeno na teorii jazyků, rozhodování, zda patří slovo do jazyka. V této přednášce se strukturálním rozpoznáváním nezabýváme.
- **Statistické rozpoznávání (feature based):** Objekty jsou popisovány pomocí příznaků (měřitelné veličiny, např. žokejové × basketbalisté → příznaky: výška, váha). Každý objekt má více příznaků, které dohromady tvoří příznakové vektory. Obecně může být každý příznak jiného typu, např. může být tvořen čísly, maticemi, vektory, funkciemi, atd. Příznak je tedy bod ve vícedimenzionálním prostoru s nějakou metrikou. Většinou se jedná o eukleidovský prostor, ale nemusí tomu tak být vždy, např. pokud by jeden příznak měl rozsah $(0, 1)$ a druhý rozsah $(0, 10^6)$, tak by eukleidovská metrika nefungovala. Navíc ne vždy musíme mít nutně metriku, občas stačí pseudometrika (hledáme pouze nějakou míru shody objektů).

Dále už se budeme zabývat pouze statistickým rozpoznáváním.

5

Segmentace

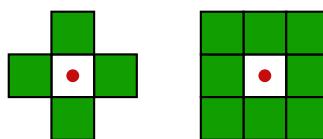
Segmentace obrazu se používá pro rozdělení obrazu na části a detekci objektů v obrazu. Je tedy třeba nalézt hranice objektů a oddělit objekty od pozadí. Navíc je nutné zachovat informaci o poloze objektu v obrazu. Jedná se ovšem o nekorektní definici, protože nikdo není schopen pořádně definovat, co je to objekt. Pokud např. dostaneme fotku, tak jak zvolíme, co je na ní objekt a co pozadí? Je to subjektivní rozhodnutí. Nejprve tedy definujeme, co by měl splňovat 2D objekt.

- **2D objekt:** binární, konečný (souvislý) a jeho hranice je tvořena jednoduchou uzavřenou křivkou, která je celá vidět.

V diskrétním rastru vzniká problém jak definovat sousední bod (pixel). Nabízejí se dvě možnosti jak k definici přistoupit, a to tzv. 4-sousednost a 8-sousednost, viz obr. 5.1. Vždy je důležité vědět, která definice se uvažuje. V podstatě se jedná o pojem ekvivalentního pojmu okolí. Pokud tedy máme zaveden pojmem okolí bodu, můžeme definovat hranici objektu.

- **Hranicní bod:** bod objektu, v jehož okolí je alespoň jeden pixel náležející do pozadí, tzn. který není součástí objektu.

Se zavedeným popisem objektu a jeho hranice ve 2D lze přistoupit k základním metodám segmentace.



Obr. 5.1: Definice 4-sousednosti a 8-sousednosti v diskrétním rastru.

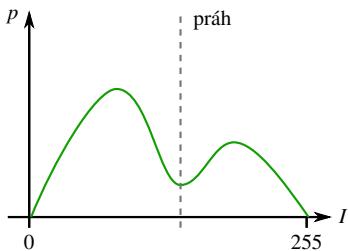
5.1 Prahování (*Thresholding*)

Jedná se o nejčastěji používanou metodu segmentace, která je nejstarší a také nejjednodušší. Její výhodou je snadná realizace a rychlosť, díky které ji lze provádět v reálném čase. Nevýhodou je naopak to, že lze jen velmi obtížně naprogramovat, aby fungovala automaticky. Navíc lze použít pouze na určitou třídu obrazů, kdy objekty a

5.2. Růst oblastí (Region growing)

pozadí jsou jasově snadno rozlišitelné. Jedná se tedy o třídu obrazů, které mají ideálně bimodální histogram, tzn. histogram má dvě lokální maxima, kde jedno odpovídá objektu a druhé pozadí, viz obr. 5.2. Pokud nalezneme lokální minimum mezi nimi, tedy práh, tak jsme schopni oddělit pozadí od objektu. Jednoduše pixely s intenzitou, která je menší než nalezený práh, položíme rovny 0, a naopak pixely s větší intenzitou položíme rovny 1. Tímto získáme binární obraz se separovanými objekty od pozadí. Obecně lze volit i více prahů, pokud histogram obsahuje více lokálních maxim, ale ukazuje se, že prahování pomocí více prahů funguje pouze lokálně.

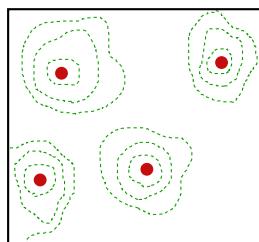
Jednoduché prahování dobře funguje jen u obrázků, které jsou z principu binární a jsou jen poškozené, např. pokud naskenujeme černobílý text. Vylepšením může být lokální prahování, které realizujeme pomocí výběrového okna (v praxi celkem dobře funguje). Pro volbu hodnoty prahu lze použít Otsuovu metodu, která se snaží rozdělit histogram na dva shluky, které budou dobře separovatelné. Další možnosti je provedení tzv. procentního prahování, pro které je ovšem nutné mít apriorní znalost, kolik procent plochy obrazu pokrývají objekty a kolik pozadí.



Obr. 5.2: Histogram obrázku a zvolený práh pro segmentaci typu objekt vs. pozadí.

5.2 Růst oblastí (Region growing)

Prvním krokem této metody je zvolení tzv. zárodečných bodů (*seed points*). Tyto body budou určit uživatel do objektů, které chce separovat, nebo se provede hrubá hranová detekce s velkým prahem a zárodečné body se umístí mimo hrany. Po zvolení zárodečných bodů se prohledává okolí jednotlivých bodů a na základě kritéria homogeneity hledáme podobné body a rozhodujeme, zda patří či nepatří do objektu, viz obr. 5.3.. Jako kritérium lze volit například intenzitu jasu. Nepoužívá se ovšem pouze segmentace podle barvy, ale také segmentace podle textury, např. textura dřeva, cihel, atd.

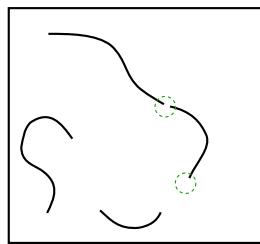


Obr. 5.3: Růst oblastí (Region growing) se zárodečnými body (červené body).

5.3 Spojování hran (Edge linking)

Tato metoda nejprve pomocí hranového detektoru naleze výrazné hrany, ty se pak snaží nějakým způsobem spojit a vytvořit uzavřené křivky, které by tvořily hranice objektů. Určení hranic se většinou provádí následujícím postupem: nalezení hran hranovým detektorem \Rightarrow binarizace prahováním s vysokým prahem (zůstane jen vysoký gradient) \Rightarrow zbavení se *isolovaných bodů*, např. pomocí morfologických operací \Rightarrow zmenšení tloušťky na jeden pixel \Rightarrow napojování hran, viz obr. 5.4.

Při napojování hran hledáme v okolí koncového bodu hrany počáteční bod jiné hrany a podle určitého kritéria se rozhodujeme, zda nalezenou hranu připojit nebo ne. Pokud ovšem použijeme pro nalezení hran např. Marr Hildrethův detektor (viz 3.4.2.1) nebo obecně detektory hledající *zero crossing points*, získáme už z podstaty detektoru uzavřené hrany. Získané oblasti ale nemusí být uzavřené správně a mohou vést ke špatné segmentaci.



Obr. 5.4: Prohledávání okolí a napojování hranice.

5.4 Matematická morfologie

Matematická morfologie je nástroj pro extrakci požadovaných částí obrazu a používá se při předzpracování obrazu a segmentaci, s důrazem na tvar hledaných objektů. Jedná se o metody založené na nelineárních operacích v obrazu. Rozlišuje se binární a šedotónová morfologie, my budeme ovšem probírat pouze binární.

Binární obraz lze vyjádřit jako 2D bodovou množinu s počátkem v bodě O . Označme X množinu bodů tvořících objekty v obraze s hodnotou 1 a X^c body doplňku této množiny, které tvoří pozadí obrazu s hodnotou 0. Morfologická transformace ψ je dána relací mezi obrazem a jinou bodovou množinou B . Tuto množinu nazýváme strukturní element a má svůj lokální počátek O_b , kterému říkáme aktuální (reprezentativní) bod. Pro libovolnou morfologickou transformaci existuje duální transformace ψ^* , tzn.

$$(\forall \psi)(\exists \psi^*) : \psi(X) = (\psi^*(X^c))^c.$$

Definujme základní morfologické operace:

- **Binární dilatace (expanze):** zaplňuje díry a zálivy menší než strukturní element. Původní objekt zvětší.

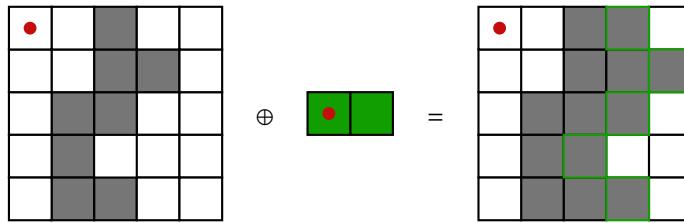
$$X \oplus B = \{p \in \mathbb{E}^2 \mid p = x + b \text{ pro } x \in X \wedge b \in B\},$$

tj. jedná se o sjednocení posunutých bodových množin. Dilatace je komutativní, asociativní a invariantní vůči posunu. Navíc je to rostoucí transformace, tzn. je-li $X \subseteq Y$ a platí-li $O_b \in B$, pak $X \subseteq Y \oplus B$.

- **Binární eroze:** odstraní objekty menší než strukturní element. Původní objekt se zmenší.

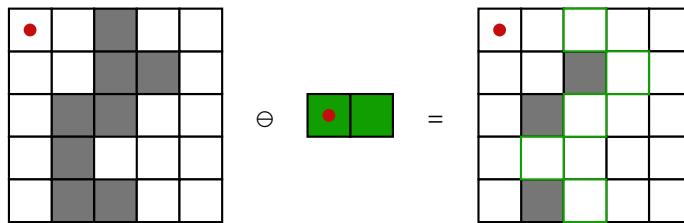
$$X \ominus B = \{x \in \mathbb{E}^2 \mid x + b \in X \text{ pro } \forall b \in B\},$$

5.4. Matematická morfologie



Obr. 5.5: Znázornění principu binární dilatace.

tj. jedná se o průnik všech posunů objektu X o vektory $-b \in B$. Eroze je invariantní vůči posunu, antiextenzivní, tzn. platí-li $O_b \in B$, pak $X \ominus B \subseteq X$, a zachovává inkluzi, tzn. $X \subseteq Y \Rightarrow X \ominus B \subseteq Y \ominus B$. Binární eroze je duální operace k binární dilataci.



Obr. 5.6: Znázornění principu binární Eroze.

Pomocí těchto základních operací lze například získat obrys objektu. Označme obrys objektu X symbolem ∂X (tedy hranici objektu tloušťky 1), pak platí $\partial X = X \setminus X \ominus B$. Dále definujme následující operace.

- **Otevření (opening):** z původního objektu odstraní výběžky menší než strukturální element. Jedná se o erozi následovanou dilatací, tj.

$$X \circ B = (X \ominus B) \oplus B.$$

Pokud platí $X \equiv X \circ B$, tak říkáme, že X je otevřený vzhledem k B .

- **Uzavření (closing):** v původním objektu zaplní díry menší než strukturální element. Jedná se o dilataci následovanou erozí, tj.

$$X \bullet B = (X \oplus B) \ominus B.$$

Pokud platí $X \equiv X \bullet B$, tak říkáme, že X je uzavřený vzhledem k B .

Poznámka 5.1 (Popis objektu pomocí matematické morfologie). Definujeme tzv. morfologické spektrum. Nad různými objekty (např. kruhy proměnného poloměru) provádíme operaci otevření a zaznamenáváme do grafu příslušnou plochu. Tato charakteristika bývá celkem spolehlivá, i když lze najít objekty, které mají stejné morfologické spektrum a přitom vypadají různě.

6

Volba příznaků

Pokud máme v obrazu nalezené objekty, je třeba nalézt příznaky a objekty od sebe rozlišit. Chceme tedy přejít z obrazového prostoru do prostoru příznakového. Na příznakový prostor klademe základní požadavky.

- **Invariance:** objekty uvnitř třídy musí mít stejné příznaky. Pokud např. víme, že objekty v dané třídě se liší pouze otočením, tak budeme volit příznak, který je invariantní vůči rotaci, ale nemusí být invariantní třeba vůči změně velikosti. Je důležité, aby deformace, vůči kterým hledáme invariant, vždy tvořily grupu, a to kvůli nutnosti uzavřenosti na skládání.

Poznámka 6.1. Pokud uvažujeme posun $\tilde{x} = x + \alpha_1$, $\tilde{y} = y + \alpha_2$, nebo změnu velikosti $\tilde{x} = s_2 \cdot x$, $\tilde{y} = s_1 \cdot y$, tak nedochází k mísení souřadnic. Naopak při rotaci $(\tilde{x}, \tilde{y})^T = \mathbb{A}(x, y)$ k mísení souřadnic dochází, a nejedná se tedy o separabilní transformaci, a proto je nejtěžší nalézt invarianty vůči rotaci. Invarianty vůči rotaci převedou nějakým „trikem“ rotaci na posun, např. změnou souřadné soustavy.

- **Diskriminabilita:** objekty patřící do různých tříd by měly mít různé hodnoty příznaků (invariance jde většinou proti diskriminabilitě).
- **Robustnost:** je potřeba, aby příznaky rozumně reagovaly na případný šum. Stejně tak malá odlišnost objektu by se měla odrazit v malé změně hodnoty příznaku.
- **Nezávislost:** příznaky by na sobě měly být nezávislé, tzn. žádná složka vektoru příznaků není funkce jiných.
- **Úplnost:** schopnost přesné rekonstrukce objektu z jeho příznaků až na grupu deformací, na které jsou příznaky invariantní. Tato vlastnost není vždy potřeba, protože většinou chceme objekty pouze klasifikovat a ne poté zpětně rekonstruovat. Výhodou úplných příznaků je, že poskytují výbornou diskriminabilitu, a to z toho důvodu, že pracují s úplnou informací a zanedbávají například jen rotaci, velikost, atd.

Od příznakového prostoru očekáváme, že v něm příznaky budou tvořit malé shluky, které jsou daleko od sebe. Dále se budeme zabývat hlavními kategoriemi příznaků.

6.1 Jednoduché vizuální příznaky

Tato kategorie příznaků se používá pro jednoduché úlohy, kdy máme málo tříd, které se od sebe hodně liší. Pokud chceme tyto příznaky použít pro složitější úlohy, je třeba úlohu „předklasifikovat“.

6.2. Úplné příznaky

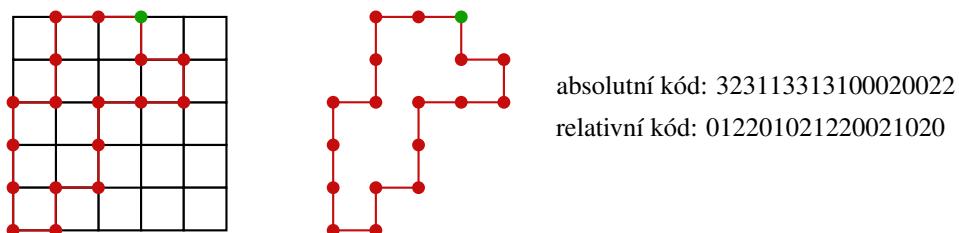
- **Obsah P nebo obvod O :** tyto příznaky nejsou invariantní na změnu měřítka, navíc spousta úplně odlišných objektů může mít stejný obsah resp. obvod.
- **Kompaktnost:** míra podobnosti ke kruhu. Je dána poměrem $4\pi P/O^2$, kde P je obsah a O je obvod.
- **Rektangularita:** poměr plochy objektu k ploše minimálního opsaného obdélníku.
- **Podlouhlost (elongation):** poměr stran minimálního opsaného obdélníku, tzn. míra podobnosti ke čtverci.
- **Konvexita:** poměr obsahu objektu A a obsahu jeho konvexního obalu C_A , tzn. $P(A)/P(C_A)$.
- **Eulerovo číslo:** $C - H$, kde C je počet komponent a H je počet dér.

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

6.2 Úplné příznaky

6.2.1 Řetězové kódy (*chain code*)

Jedná se o hledání hranice objektu a její popis pomocí řetězce čísel, kde následující číslo popisuje směr, jakým vede hranice ze stávajícího bodu. Pokud vezmeme čtyři směry a označíme je čísly: 0 – nahoru, 1 – doleva, 2 – doprava, 3 – dolů, jedná se o absolutní kód. Pokud ovšem objekt otočíme, tak se nám popis rozsype, a proto se používá tzv. relativní (diferenční) kód, který je vůči rotaci invariantní. Při použití relativního vyjádření stačí pouze hodnoty: 0 – rovně, 1 – doleva, 2 – doprava. Procházíme křivku a zapisujeme 0, pokud neměníme směr, 1 resp. 2 když křivka zatáčí. Svislé a vodorovné části se pak vyobrazí jako sekvence nul. Je také možné uvažovat diagonální směry, pak je třeba pro reprezentaci křivky osm čísel, resp. sedm při použití relativního kódu. Tato metoda popisu se moc nepoužívá, protože je těžké definovat metriku, která by dobře reprezentovala vizuální podobu objektů.



Obr. 6.1: Ukázka absolutního a relativního řetězového kódu.

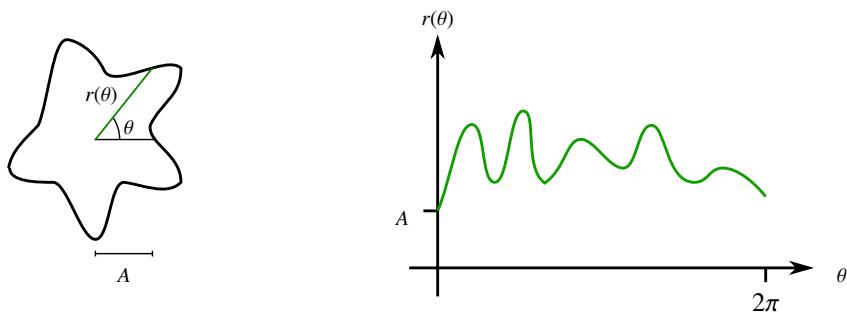
6.2.2 Polygonální approximace

Tato metoda approximuje objekty pomocí polygonů. Pro použití je předem nutné vědět počet vrcholů objektu a z tohoto důvodu se používá v podstatě jen v případech, kdy víme, že objekty jsou polygony a známe jejich počet vrcholů.

6.2.3 Radiální funkce

Uvažujme hvězdicové objekty, tzn. libovolná polopřímka vedoucí z těžiště objektu má pouze jeden průsečík s hranicí objektu. Nejedná se o stejnou vlastnost jako konvexnost. Platí, že každý konvexní objekt je hvězdicový, ale opačná implikace neplatí. Pokud budeme zaznamenávat vzdálenost těžiště od hranice objektu v polárních souřadnicích, dostaneme reprezentaci objektu, tzn. zaznamenáváme $r(\theta)$ pro $\forall \theta \in (0, 2\pi)$.

Při provedení rotace dojde v záznamu funkce $r(\theta)$ pouze k cyklickému posunutí. Protože víme, že při posunutí se u Fourierovy transformace nemění amplituda a fáze se změní známým způsobem (viz věta 1.9), můžeme vzít pouze amplitudy Fourierovy transformace, čímž získáme invariant vůči rotaci. Díky této vlastnosti také nezáleží na tom, kde začneme se vzorkováním úhlu θ , protože jiný začátek vzorkování se opět projeví pouze cyklickým posunutím. Jediným problémem zůstává omezenost pouze na hvězdicové objekty.



Obr. 6.2: Ukázka popisu objektu pomocí radiální funkce.

6.2.4 Tvarový vektor (*shape vector*)

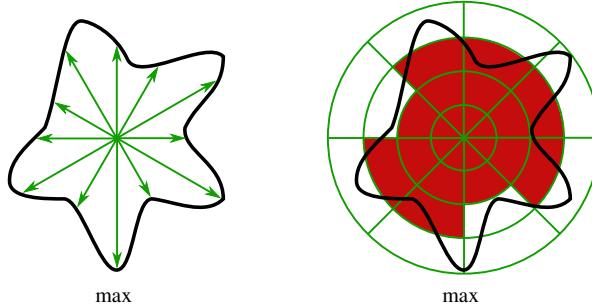
Chceme-li charakterizovat nějakým způsobem objekt, lze použít tzv. tvarových vektorů, což je posloupnost vzdáleností od těžiště objektu k jeho okraji po jistých úhlových přírůstcích, tzn. jedná se o navzorkovanou radiální funkci. Vzorkování začínáme v místě, kde je vzdálenost těžiště objektu od jeho hranice největší. Složky tvarového vektoru nejsou primárně invariantní na rotaci objektu ani na změnu měřítka. Invariantnosti na rotaci lze dosáhnout stejným postupem jako u radiální funkce. Abychom zajistili invariantní na změnu měřítka, vybereme maximální vzdálenost těžiště od hranice a složky tvarového vektoru touto vzdáleností vydělíme. Problém nastává, pokud existuje více složek tvarového vektoru s maximální hodnotou. Pokud ovšem zvolíme dostatečné vzorkování, tak se více maximálních hodnot projeví pouze cyklickým posunutím tvarového vektoru. Jestliže tedy přijmeme, že musíme zkoumat všechny cyklické posuny tvarového vektoru, tak nezáleží na tom, kde vzorkování začneme.

6.2.5 Tvarová matice (*shape matrix*)

Tvarové vektory a radiální funkce dobře fungují jen na hvězdicové objekty, ale objekty bývají složitější. Uvažujme kružnicovitou síť, která překrývá objekt a začíná v místě, kde je maximální vzdálenost těžiště od hranice. Tato síť jde zapsat jako matice, která obsahuje 1, pokud objekt překrývá více než polovinu segmentu, jinak 0. Jde tedy o polární převzorkování celého objektu. Jako metriku lze například volit počet segmentů, ve kterých se matice shodují, a velikost vzorkování volíme tak, aby segmenty blízko středu byly přibližně čtvercové. Díky tomu, že pro všechny objekty bude stejný počet segmentů, a tedy stejně velká tvarová matice, dostáváme invariantní na změnu měřítka. Kvůli zvolenému typu vzorkování ovšem nastává otázka, zda je problém, že segmenty jsou různě velké,

6.3. Transformační příznaky (*Transform coefficient features*)

a v různých částech objektu tedy uvažujeme různé rozlišení. Toto by se dalo obejít volbou vzorkování tak, aby segmenty stejné byly, nebo volbou čtvercové sítě. Nicméně ani jedna z těchto možností nefunguje.



Obr. 6.3: Reprezentace objektu pomocí tvarového vektoru a tvarové matice.

6.3 Transformační příznaky (*Transform coefficient features*)

Do této kategorie spadají například waveletová transformace (*wavelet transform*), momentové invarianty a Fourierovy deskriptory, kterými se budeme zabývat. Momentovým invariantům a waveletové transformaci jsou věnovány vlastní kapitoly.

6.3.1 Fourierovy deskriptory

Fourierovy deskriptory jsou rozšířením radiální funkce o invariantnost vůči změně měřítka. Pokud provedeme Fourierovu transformaci radiální funkce a z ní vezmeme amplitudu, tak Fourierovými deskriptory rozumíme prvních N koeficientů Fourierovy transformace. Abychom získali invariantnost vůči změně měřítka, tak sadu Fourierových deskriptorů dělíme nultým Fourierovým koeficientem, který odpovídá střední hodnotě radiální funkce

$$C_k := \frac{|F(k)|}{|F(0)|}.$$

Protože fáze Fourierovy transformace nese podstatnou část informace, tak není možné zpětně zrekonstruovat objekt, a nejedná se tedy o úplný příznak. Navíc jsme omezeni pouze na hvězdicové objekty a spojitý případ.

V praxi se tedy používá jistá modifikace, která umožňuje použití i pro nehvězdicové objekty v diskrétním případě. Vezmeme hranici objektu a představíme si ji jako komplexní funkci

$$z_n = x_n + i \cdot y_n.$$

Vzhledem k tomu, že se jedná pouze o posunutí (přičítáme konstantu), změní se pouze nultý Fourierův koeficient, který má nyní význam vzdálenosti od počátku. V případě, že objekt posuneme, tzn. $\tilde{z}_n = z_n - c$, tak se nám změní vzdálenost od počátku, což opět vede pouze ke změně nultého Fourierova koeficientu, tedy

$$\tilde{Z}(k) = \sum_{n=0}^{N-1} (\tilde{z}_n - c) e^{-\frac{1}{N} 2\pi i n k} = \sum_{n=0}^{N-1} z_n e^{-\frac{1}{N} 2\pi i n k} - c \cdot \sum_{n=0}^{N-1} e^{-\frac{1}{N} 2\pi i n k} = \begin{cases} Z(0) - c \cdot N & \dots \quad k = 0 \\ Z(k) & \dots \quad k \neq 0 \end{cases}.$$

Pokud nultý koeficient zahodíme, tak dostáváme invariant na posun. Dále platí, že rotace v komplexních číslech je pouze přenásobení konstantou $\tilde{z}_n = z_n \cdot e^{i\varphi}$, a tedy

$$\tilde{Z}(k) = \sum_{n=0}^{N-1} (z_n \cdot e^{i\varphi}) e^{-\frac{1}{N}2\pi ink} = e^{i\varphi} \cdot \sum_{n=0}^{N-1} z_n e^{-\frac{1}{N}2\pi ink} = e^{i\varphi} \cdot Z(k) \implies |\tilde{Z}(k)| = |Z(k)|.$$

Abychom získali invariantnost vůči změně měřítka $\tilde{z}_n = c \cdot z_n$, tak sadu Fourierových deskriptorů dělíme prvním Fourierovým koeficientem (nejdělíme nultým, protože ten jsme zahodili) a dostáváme vztah

$$C_k := \frac{|\tilde{Z}(k)|}{|\tilde{Z}(1)|} = \frac{\left| \sum_{n=0}^{N-1} (c \cdot z_n) e^{-\frac{1}{N}2\pi ink} \right|}{\left| \sum_{n=0}^{N-1} (c \cdot z_n) e^{-\frac{1}{N}2\pi in} \right|} = \frac{c}{c} \cdot \frac{|Z(k)|}{|Z(1)|} = \frac{|Z(k)|}{|Z(1)|}.$$

Protože ale pořád zahazujeme fázi, tak se nejedná o úplný příznak.

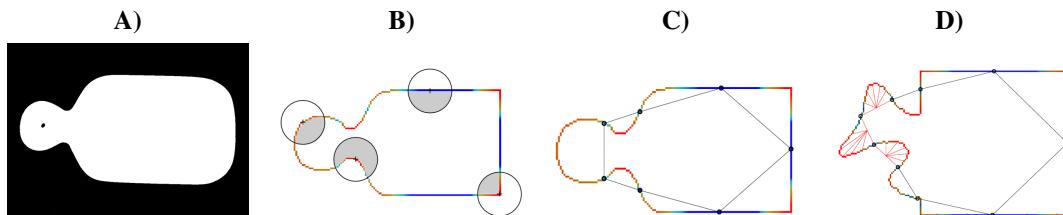
6.4 Diferenciální invarianty

Všechny výše uvedené příznaky jsou globální, tzn. je třeba vidět celý objekt, aby bylo možné tyto příznaky použít. Abychom byli schopni najít příznaky i částečně zakrytých objektů, budeme hledat lokální příznaky. Můžeme například zkoumat derivaci hranice. Protože na derivaci hranice potřebujeme znát pouze okolí bodu, tak v bodech hranice, které nejsou zakryté, bude mít derivace stejnou hodnotu. Můžeme tedy porovnávat derivaci hranice v těch bodech, kde není zakrytá, a podle toho hledat míru shody. Příkladem diferenciálního příznaku je křivost, která je definována vztahem

$$C(t) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}.$$

Pokud jsme schopni nalézt inflexní body, tj. řešit rovnici $\ddot{x}\dot{y} - \dot{y}\ddot{x} = 0$, tak je používáme pro popis místo lokálních extrémů, tzn. porovnáváme objekty v inflexních bodech hranice. Inflexní body se totiž na rozdíl od lokálních extrémů dobře zachovávají při affiní transformaci. Jestliže je inflexních bodů málo, tak metoda nefunguje, a proto se používá pouze na velmi členité objekty. Nevýhodou diferenciálních invariantů je velmi velká nestabilita.

Abychom zvýšili stabilitu, je možné provádět porovnávání jiným způsobem. Použijeme Inflexní body, které jsou affině a projektivně invariantní. Podle inflexních bodů „rozsekáme“ hranici na části a tyto části poté popisujeme globálními příznaky. Takovým příznakům říkáme semi-diferenciální invarianty.



Obr. 6.4: A) Originální objekt a B) ukázka principu diferenciálních invariantů a C) semi-diferenciálních invariantů. Obrázek D) ilustruje rozpoznávání částečně zakrytého objektu. Tento objekt je rozdělen podle inflexních bodů a v každé části hledáme příznaky pomocí tvarového vektoru.

6.4. Diferenciální invarianty

6.4.1 Curvature Scale Space (CSS)

Pokud budeme narovnávat hranici objektu, tak se k sobě budou inflexní body přibližovat a postupně zanikat. Rychlosť, jakou se k sobě body přibližují, je dobrý popis objektu, a navíc je rotačně invariantný. Existuje i modifikace, která je invariantný vůči afinní transformaci.

Momentové invarianty

V obecném případě je vztah mezi originálním obrazem $f(x, y)$ a pozorovaným obrazem $g(x, y)$ dán vztahem $g = \mathcal{D}(f)$, kde symbolem \mathcal{D} značíme operátor degradace. Problémem je, že neznáme předpis operátoru \mathcal{D} , nebo ho známe pouze parametricky. K řešení můžeme použít několik postupů.

1. **Hrubá síla:** rozšíříme trénovací množinu o všechny možné degradace operátorem \mathcal{D} .
2. **Inverze \mathcal{D} (normalizační přístup):** definujeme normalizovanou polohu a do té se snažíme obrázek převést (např. rozmazaný obrázek → dekonvoluce, rotace → otočení do nějaké konkrétní polohy). Řešení inverze ovšem bývá obtížné a numericky nestabilní.
3. **Popis pomocí invariantních příznaků:** obraz popisujeme pomocí příznaků, které jsou invariantní na daný druh degradace. Částečně je tato úloha a inverze operátoru \mathcal{D} ekvivalentní (často lze získat invarianty přes inverzní úlohy) a nemusí to být vždy nejlepší přístup.

Definice 7.1. Invariantem rozumíme funkcionál I definovaný na obrazovém prostoru, který pro všechny přípustné operátory degradace \mathcal{D} a libovolný obraz f splňuje vztah

$$I(f) = I(\mathcal{D}(f)).$$

Je důležité zmínit, že požadujeme, aby degradace tvořily grupu, tzn. aby množina degradací obsahovala jednotkový prvek, existoval v ní inverzní prvek a byla uzavřená na kompozici. Ve skutečnosti není třeba grupa, ale stačí semi-grupa, ale grupa se většinou uvažuje pro jednoduchost. Nejdůležitější vlastností je uzavřenosť. Pokud bychom tuto vlastnost neměli, tak nemusí opakování deformace patřit do třídy deformací, a tedy nemáme invarianty na opakování deformace. Pokud bychom přece jenom našli invariant, který je invariantní vůči neuzavřené třídě deformací, tak je určitě invariantní vůči nejmenší grupě obsahující tuto třídu deformací. Tedy je to invariant vůči větší třídě a neexistuje invariant, který by byl invariantní pouze vůči uvažované třídě deformací. Můžeme tedy uvažovat větší třídu deformací a nic se pro nás nezmění.

Pokud máme definovanou grupu deformací na prostoru invariantů, můžeme přistoupit k jeho rozkladu. Tento rozklad můžeme udělat dvěma (obecně jinými) způsoby.

1. **Pomocí orbitů:** definujme grupovou akci z $\mathcal{D} \times I \rightarrow I$. Orbity $\mathcal{D}(f)$, tzn. množiny všech možných degradací, které mohou vzniknout, jsou třídy ekvivalence uvažovaného klasifikačního problému. Invarianty jsou charakteristiky orbitu a chceme, aby daný orbit plně popisovaly.

7.1. Základní vztahy

2. **Pomocí invariantů:** definujeme ekvivalenci vztahem $I(f) = I(g)$. Tako získaný rozklad je obecně hrubší než rozklad pomocí orbitů.

Pokud jsou oba uvedené faktorprostory stejné, tak pro každý $f \neq \mathcal{D}(g)$ platí $I(f) \neq I(g)$ a máme tzv. úplný systém invariantů. Ne vždy ovšem potřebujeme úplný systém invariantů, např. pokud chceme rozpoznávat tváře pouze zaměstnanců a ne všech lidí na planetě, potřebujeme, aby bylo pomocí vybraných invariantů možné rozlišit pouze skupinu zaměstnanců.

7.1 Základní vztahy

Nyní se budeme zabývat momentovými invarianty. Jedná se o projekci do polynomální báze. Nechť $f(x, y)$ je po částech spojitá funkce definovaná na $\Omega \subset \mathbb{R} \times \mathbb{R}$ a $\{\mathcal{P}_{pq}(x, y)\}$ je množina polynomů definovaná na Ω . Pak je projekce dána vztahem

$$M_{pq} = \int_{\mathbb{R}^2} \mathcal{P}_{pq}(x, y) f(x, y) dx dy.$$

Protože momenty nejsou (na rozdíl od Fourierových koeficientů $\sum c_k \varphi_k = f$) algebraické souřadnice vyjádřené pomocí polynomální báze (jedná se pouze o průměty), nelze zpětnou rekonstrukci provést obvyklým způsobem

$$f(x, y) \neq \sum M_{pq} \mathcal{P}_{pq}(x, y).$$

Momenty jsou algebraické souřadnice vyjádřené pomocí polynomální báze pouze v případě, že je systém polynomů ortogonální.

7.1.1 Geometrické momenty

Velmi často se používají geometrické momenty, které jsou definovány následujícím vztahem

$$m_{pq}^f = \int_{\mathbb{R}^2} x^p y^q f(x, y) dx dy,$$

kde řádem momentu rozumíme číslo $p + q$. S jejich pomocí můžeme snadno nalézt souřadnice těžiště objektu, které mají tvar

$$x_t = \frac{m_{10}^f}{m_{00}^f}, \quad y_t = \frac{m_{01}^f}{m_{00}^f},$$

kde m_{00}^f představuje v případě binárních obrazů plochu. Pokud budeme obraz považovat za hustou pravděpodobnosti a normalizujeme $m_{00}^f = 1$, pak m_{10}^f , m_{01}^f jsou střední hodnoty a m_{20}^f , resp. m_{02}^f je vertikální, resp. horizontální rozptyl. Nakonec moment m_{11}^f představuje kovariaci.

Věta 7.2 (Jednoznačnost). *Pokud je $f(x, y)$ po částech spojitá funkce a Ω je omezená, potom*

$$f(x, y) \iff \{m_{pq}\}, \quad p, q = 0, 1, 2, \dots, \infty.$$

7.2 Invarianty na posun

Abychom získali invarianty vůči posunu zavádíme centrální geometrické momenty ve tvaru

$$\mu_{pq}^f = \int_{\mathbb{R}^2} (x - x_t)^p (y - y_t)^q f(x, y) dx dy.$$

Mezi geometrickými a centrálními geometrickými momenty platí následující vztahy

$$\mu_{00} = m_{00}, \quad \mu_{10} = \mu_{01} = 0, \quad \mu_{pq} = \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} (-1)^{k+j} x_t^k y_t^j m_{p-k, q-j}^f.$$

7.3 Invarianty na změnu měřítka a posun

Uvažujme změnu měřítka $\tilde{x} = s \cdot x$, $\tilde{y} = s \cdot y$ a dosadíme do centrálního geometrického momentu

$$\tilde{\mu}_{pq}^f = \int_{\mathbb{R}^2} (\tilde{x} - \tilde{x}_t)^p (\tilde{y} - \tilde{y}_t)^q \tilde{f}(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y} = \int_{\mathbb{R}^2} s^p (x - x_t)^p s^q (y - y_t)^q f(x, y) s^2 dx dy = s^{p+q+2} \mu_{pq}^f.$$

Víme, že platí $\tilde{\mu}_{00}^f = s^2 \mu_{00}^f$ a označíme

$$v_{pq}^f = \frac{\mu_{pq}^f}{(\mu_{00}^f)^\omega} \implies \tilde{v}_{pq}^f = \frac{\tilde{\mu}_{pq}^f}{(\tilde{\mu}_{00}^f)^\omega} = \frac{s^{p+q+2} \mu_{pq}^f}{s^{2\omega} (\mu_{00}^f)^\omega} \stackrel{!}{=} v_{pq}^f.$$

Protože chceme získat invariantnost na změnu měřítka, požadujeme, aby byla platila uvedená rovnost, která je splněna pro $\omega = \frac{p+q}{2} + 1$. Tyto momenty se nazývají normalizované momenty.

7.4 Rotační invarianty

Uvažujme rotaci ve tvaru

$$\begin{aligned} \tilde{x} &= x \cos \theta - y \sin \theta, \\ \tilde{y} &= x \sin \theta + y \cos \theta. \end{aligned}$$

Dosazením do obecného tvaru pro geometrické momenty získáme vztah

$$\tilde{\mu}_{pq} = \int_{\mathbb{R}^2} (x \cos \theta - y \sin \theta)^p (x \sin \theta + y \cos \theta)^q f(x, y) |d\tilde{x} d\tilde{y}|,$$

který ovšem nelze obecně jednoduše řešit. Pro malé p, q lze odvodit (*M. K. Hu - 7 invariants of the 3rd order (1962)*) invarianty ve tvaru

$$\begin{aligned} \phi_1 &= \mu_{20} + \mu_{02}, & \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2, & \dots, \\ \phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2, & \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2, & \dots. \end{aligned}$$

Nicméně tyto invarianty jsou závislé, např. platí

$$\phi_3 = \frac{\phi_5^2 + \phi_7^2}{\phi_4^3},$$

a také jsou nekompletní, např. nelze jednoznačně vyjádřit μ_{11}

$$\mu_{11}^2 = \frac{1}{4} \left(\phi_2 - \left(\frac{\phi_6}{\phi_4} \right)^2 \right).$$

Navíc víme, že do třetího řádu máme celkem 10 momentů, moment nultého řádu použijeme na normalizaci, momenty prvního řádu na translaci, tzn. zbývá 7 momentů a na rotaci potřebujeme jeden parametr. Celkem tedy může být nejvýše 6 invariantů na rotaci do řádu 3. Na snadné úlohy tyto invarianty stačí (např. rozpoznávání siluet letadel), ale pro náročnější mají nízkou diskriminabilitu. Pokud bychom do těchto 7 invariantů dosadili normalizované momenty, získali bychom invarianty na posun, rotaci i změnu měřítka, tzv. TRS invarianty.

7.4. Rotační invarianty

7.4.1 Obecná konstrukce

Uvažujme transformaci do polárních souřadnic

$$\begin{aligned} x &= r \cos \theta, & r &= \sqrt{x^2 + y^2}, \\ y &= r \sin \theta, & \theta &= \arctan\left(\frac{y}{x}\right), \end{aligned}$$

a zaved' me tzv. cirkulární momenty ve tvaru

$$c_{pq} = \int_0^\infty \int_0^{2\pi} \mathcal{R}_{pq}(r) e^{i\xi(p,q)\theta} f(r, \theta) r d\theta dr,$$

kde $\xi(p, q)$ je funkce od p, q , a $\mathcal{R}_{pq}(r) e^{i\xi(p,q)\theta}$ představuje prvky množiny polynomů vyjádřené v polárních souřadnicích. Pokud budeme otáčet obrázek o α , tak se nám hodnota momentu změní následujícím způsobem

$$\tilde{c}_{pq} = e^{i\xi(p,q)\alpha} \cdot c_{pq},$$

což je analogie k *Fourier shift theorem*, viz věta 1.9. Otázkou nyní je jak se zbavit členu $e^{i\xi(p,q)\alpha}$. Můžeme například položit $|\tilde{c}_{pq}| = |c_{pq}|$, címkž získáme invariant, ale nejedná se o úplný systém invariantů. Pokud chceme úplný systém, tak musíme znát konkrétní tvar funkce $\xi(p, q)$, jinak tuto úlohu nelze řešit nějakým obecným způsobem.

7.4.2 Konstrukce pomocí komplexních momentů

Zaved' me komplexní momenty následujícím vztahem

$$c_{pq}^f = \int_{\mathbb{R}^2} (x + iy)^p (x - iy)^q f(x, y) dx dy.$$

Pro tyto momenty můžeme snadno odvodit vztahy s normálními momenty

$$\begin{aligned} c_{pq} &= \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} (-1)^{q-j} \cdot i^{p+q-k-j} \cdot m_{k+j, p+q-k-j}, \\ m_{pq} &= \frac{1}{2^{p+q} \cdot i^q} \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} (-1)^{q-j} \cdot c_{k+j, p+q-k-j}. \end{aligned}$$

Navíc pro komplexní momenty platí vztah $c_{qp} = c_{pq}^*$, kde hvězdička značí komplexní sdružení. Pokud vyjádříme komplexní momenty v polárních souřadnicích, tzn. dosadíme transformaci $x + iy \rightarrow re^{i\varphi}$ a $x - iy \rightarrow re^{-i\varphi}$ získáme následující vztah

$$c_{pq}^f = \int_{\mathbb{R}} \int_0^{2\pi} r^{p+q} e^{i\varphi(p-q)} f(r, \varphi) r dr d\varphi = \int_{\mathbb{R}} \int_0^{2\pi} r^{p+q+1} e^{i\varphi(p-q)} f(r, \varphi) dr d\varphi,$$

ze kterého je jasné vidět, že komplexní momenty jsou pouze speciálním případem cirkulárních momentů pro volbu $\xi(p, q) = p - q$.

Z vlastností cirkulárních momentů a ze znalosti funkce $\xi(p, q)$ pro komplexní momenty, můžeme snadno odvodit, jak se tyto momenty chovají při rotaci

$$\tilde{f}(r, \theta) = f(r, \theta + \alpha) \implies \tilde{c}_{pq} = e^{-i(p-q)\alpha} \cdot c_{pq}.$$

Výraz $e^{-i(p-q)\alpha}$ vyjadřuje otočení c_{pq} v komplexní rovině, tzn. fáze je posunutá o $(p - q)\alpha$ a amplituda se nemění. Následující věta shrnuje jak konstruovat invarianty na rotaci.

Věta 7.3. Necht' $n \geq 1$ a dále k_i, p_i a q_i , $i \in \hat{n}$ jsou nezáporná celá čísla splňující vztah

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

Potom invariant vůči rotaci má tvar

$$I = \prod_{j=1}^n c_{p_j q_j}^{k_j}.$$

Předchozí věta říká, že zřetězíme invarianty tak, aby se jejich otočení v komplexní rovině vzájemně vyrušila (součet otočení na jednu i druhou stranu je 0), a tím získáme rotační invariant, tj.

$$\tilde{I} = \prod_{j=1}^n \tilde{c}_{p_j q_j}^{k_j} = \prod_{j=1}^n e^{-i(p_j - q_j)k_j \alpha} \cdot c_{p_j q_j}^{k_j} = e^{\underbrace{-i\alpha \sum_{j=0}^n k_j (p_j - q_j)}_{=0}} \prod_{j=1}^n c_{p_j q_j}^{k_j} = I.$$

Těchto invariantů je nekonečně mnoho, ale spoustu z nich je závislých. Vlastností součinových invariantů je, že pokud je jeden z momentů v součinu blízký nule, tak celý součin je blízký nule. Toto může být velký problém, např. pro symetrické objekty je hodně momentů nulových.

Nyní se zabýveme tím jak vybrat bázi invariantů. Chceme, aby báze byla nezávislou a úplnou množinu invariantů. V tomto kontextu rozumíme úplností to, že jsme schopni zpětně spočítat všechny komplexní momenty až na jeden (jeden stupeň volnosti), a nezávislostí rozumíme to, že neexistuje invariant, který by byl funkcí ostatních. Následující věta shrnuje jak konstruovat bázi invariantů.

Věta 7.4. Necht' máme komplexní momenty až do řádu $r \geq 2$. Bázi \mathcal{B} rotačních invariantů až do řádu r lze konstruovat následujícím způsobem

$$(\forall p, q : p \geq q \wedge p + q \leq r) (\Phi(p, q) = c_{pq} c_{q_0 p_0}^{p-q}),$$

kde p_0, q_0 jsou libovolné indexy splňující

$$c_{p_0 q_0} \neq 0, \quad p_0 - q_0 = 1, \quad p_0 + q_0 \leq r.$$

Důkaz, že se jedná o bázi invariantů, plyne z nezávislosti momentů c_{pq} . V každém invariantu se vyskytuje jiný moment, což znamená, že jsou invarianty nezávislé. Problém nastává u důkazu úplnosti, tzv. *Inverse problem*. Orientace obrazu z těchto invariantů nemůže být obecně získána. Proto zavádíme podmínku $c_{p_0 q_0} \in \mathbb{R}^+$, čímž ztratíme jeden stupeň volnosti. Díky této podmínce a vlastnosti komplexních momentů $c_{qp} = c_{pq}^*$ platí $c_{q_0 p_0} = c_{p_0 q_0}^* = c_{p_0 q_0}$ a celkem dostáváme

$$\Phi(p_0, q_0) = c_{p_0 q_0} \cdot c_{q_0 p_0} = c_{p_0 q_0}^2 \implies c_{p_0 q_0} = \sqrt{\Phi(p_0, q_0)}, \quad c_{pq} = \frac{\Phi(p, q)}{c_{q_0 p_0}^{p-q}}.$$

Příklad 7.5. Příklad báze třetího řádu, $p_0 = 2, q_0 = 1$ obsahující 6 různých reálných invariantů:

$$\begin{aligned} \Phi(1, 1) &= c_{11} \in \mathbb{R}, & \Phi(2, 0) &= c_{20} c_{12}^2 \in \mathbb{C}, \\ \Phi(2, 1) &= c_{21} c_{1,2} \in \mathbb{R}, & \Phi(3, 0) &= c_{30} c_{12}^3 \in \mathbb{C}. \end{aligned}$$

7.4.3 Pseudoinvarianty

Zabýveme se tím, jak se chovají rotační invarianty při zrcadlení obrazu. Zrcadlení obrazu odpovídá platnosti vztahu $\bar{f}(x, y) = f(x, -y)$, z čehož získáváme

$$\bar{c}_{pq} = c_{pq}^*, \quad \overline{\Phi(p, q)} = \overline{c_{pq} c_{p_0 q_0}^{p-q}} = c_{pq}^* \cdot (c_{q_0 p_0}^*)^{p-q} = \Phi(p, q)^*.$$

Invarianty, které jsou reálnými částmi Φ , jsou invarianty vůči zrcadlení. Naopak invarianty, které jsou imaginárními částmi Φ , mění při zrcadlení znaménko, a tedy invarianty vůči zrcadlení nejsou. Invariance na zrcadlení ovšem může být v některých případech na škodu.

7.4. Rotační invarianty

7.4.4 Normalizace na rotaci

Normalizace pomocí momentů spočívá v předdefinování některých momentů. Většinou se používají momenty nižších řádů, protože jsou stabilnější a méně náchylné na šum. V případě invariantů na posun je normalizace jednoduchá a spočívá v posunutí těžiště do počátku souřadnic. Toho docílíme normalizací momentů prvního řádu na 0, tzn. volbou $\tilde{m}_{10} = \tilde{m}_{01} = 0$. Obdobně lze postupovat v případě invariantů na změnu měřítka. Budeme chtít, aby byl nultý moment roven 1, tzn. $\tilde{m}_{00} = 1$. Pokud budeme uvažovat změnu měřítka danou transformací $\tilde{x} = sx$, $\tilde{y} = sy$, tak lze snadno ukázat, že platí $\tilde{m}_{00} = s^2 m_{00}$ a navíc pro $m_{00} \neq 0$ vždy takové s existuje.

Poznámka 7.6. Pokud moment použijeme jako normalizační podmínku, tak už tento invariant nelze použít pro rozpoznávání, protože má pro všechny objekty stejnou hodnotu.

Při normalizaci na rotaci využijeme podobnou myšlenku, jako v sekci 11.1 o metodě PCT. Předpokládejme, že používáme invarianty normalizované na posun a změnu měřítka, tzn. předpokládáme nulové střední hodnoty $\tilde{\mu}_{10} = \tilde{\mu}_{01} = 0$ a $\tilde{\mu}_{00} = 1$. Budeme požadovat, aby byl moment druhého řádu $\tilde{\mu}_{11}$ nulový. Tento moment odpovídá v případě nulových středních hodnot kovarianci. Myšlenka je tedy taková, že zvolíme takovou souřadnou soustavu, ve které platí $\tilde{\mu}_{11} = 0$. Vezmeme momentovou matici

$$\begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix}$$

a budeme jí diagonalizovat, viz sekce 11.1 o PCT. Protože tato matice je symetrická, můžeme sestavit OG bázi danou vlastními vektory této matice a pomocí ní provést diagonalizaci. Maticově můžeme normalizaci popsat následovně

$$\mathbb{M} = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix}, \quad \tilde{\mathbb{M}} = \mathbf{G}^T \mathbb{M} \mathbf{G} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} = \begin{pmatrix} \tilde{\mu}_{20} & 0 \\ 0 & \tilde{\mu}_{02} \end{pmatrix},$$

což znamená, že řešíme úlohu na vlastní čísla ve tvaru $|\tilde{\mathbb{M}} - \lambda \mathbb{I}| = 0$. Spočteme tedy vlastní vektory momentové matice a vezmeme větší z nich. Porovnáme tento vektor s osou x , abychom měli větší rozptyl ve směru x . Celkem získáme úhel, o který musíme rotovat obraz

$$\tan 2\theta = \frac{-2\mu_{11}}{\mu_{20} - \mu_{02}}.$$

Tento úhel ovšem není jednoznačný (mohou to být 4 způsoby natočení obrazu). Rotací totiž obraz srovnáme pouze podle hlavních os souřadné soustavy, ale už neřešíme orientaci na těchto osách. Musíme proto dodefinovat normalizační podmínky, abychom měli jednoznačnou normalizaci $\tilde{\mu}_{20} \geq \tilde{\mu}_{02}$ a $\tilde{\mu}_{30} \geq 0$. První podmínka zaručuje, že má obraz větší rozptyl v ose \tilde{x} , a druhá, že má obrázek víc hmoty vpravo od těžiště. Tento způsob normalizace se nazývá *principal axis normalization*.

Poznámka 7.7. V posledních dvou podmírkách používáme pro normalizaci momenty $\tilde{\mu}_{20}$, $\tilde{\mu}_{02}$ a $\tilde{\mu}_{30}$, ale nepředpisujeme přesnou hodnotu, a tak je lze použít i pro rozpoznávání.

7.4.4.1 Referenční elipsa objektu

Referenční elipsu objektu definujeme jako elipsu, která má stejně druhé momenty jako objekt. Touto elipsou můžeme objekt popsat a normalizaci provést pro elipsu. Normalizace spočívá v tom, že elipsu dostaneme do

polohy, kdy její hlavní osa splývá s osou x a vedlejší osa s osou y . Poté platí vztahy

$$\tilde{\mu}_{20} = \frac{\pi a^3 b}{4}, \quad \tilde{\mu}_{02} = \frac{\pi a b^3}{4},$$

kde a, b , je hlavní resp. vedlejší osa.

7.4.4.2 Normalizace pomocí komplexních momentů

Předpokládejme, že komplexní moment c_{pq} je reálný, kladný. Poté pro úhel otočení platí vztah

$$\alpha = \frac{1}{p-q} \cdot \arctan \left(\frac{\operatorname{Im}(c_{pq})}{\operatorname{Re}(c_{pq})} \right).$$

Pokud platí, že $p - q > 1$, pak při jedné otočce obrazu proběhne $p - q$ otoček v bázi momentů. Omezíme se tedy na případ $p - q = 1$, kdy jedna otočka obrazu odpovídá jedné otočce v bázi momentů a normalizace je v takovém případě jednoznačná. Pokud zvolíme $p = 2$, a $q = 0$, tak normalizace vede na tradiční normalizaci *principal axis normalization*

$$c_{20} = m_{20} - m_{02} + 2im_{11}.$$

Pro $c_{20} > 0$ je ovšem normalizace pořád nejednoznačná, např. nemůžeme rozlišit, zda je objekt orientován na ose x vpravo nebo vlevo. Pokud vezmeme $c_{21} > 0$ tak už je to jednoznačné.

Poznámka 7.8. Ve skutečnosti není nutné a často ani žádoucí obrazy normalizovat, protože transformace vede na nepřesnosti a je pomalá. Stačí spočítat transformační rovnice a poté pouze transformovat momenty spočítané v nenormalizované soustavě. Argumentem proti normalizaci je, že pokud uděláme chybu při výpočtu normalizačních podmínek, tak se tato chyba propaguje do všeho dál. Z tohoto důvodu se častěji počítají přímé invarianty. Pokud máme požadavek, že chceme úplný systém invariantů, je situace jiná. V takovém případě je jednodušší spočítat normalizované invarianty, protože ty už rovnou tvoří úplný a nezávislý systém.

Poznámka 7.9 (Užití TRS invariantů v praxi). V podstatě vždycky se používají pro binární obrazy, např. rozpoznávání objektů podle kontur. Dále se používají pro registraci obrazu...

7.4.5 Rotační symetrie

Definice 7.10. Objekt má N -fold rotační symetrii, pokud je identický s původním objektem při rotaci o $\frac{2\pi j}{N}$ pro všechna $j \in \hat{N}$.

Poznámka 7.11. Každý osově symetrický objekt s S osami souměrnosti je N -fold symetrický pro $N = S$. Naopak, pokud je objekt N -fold symetrický, tak je buď osově symetrický s $S = N$ osami souměrnosti, nebo není osově symetrický vůbec.

Z vlastností N -fold symetrických objektů plyne vztah pro cirkulární momenty

$$\tilde{c}_{pq} = e^{-(p-q)\frac{2\pi}{N}} \cdot c_{pq} \stackrel{!}{=} c_{pq}.$$

Aby tato rovnost platila, tak buď musí platit $c_{pq} = 0$ nebo

$$\frac{p-q}{N} \in \mathbb{Z} \implies e^{-(p-q)\frac{2\pi}{N}} = 1.$$

Tím dostáváme snadnou podmínu na testování, zda je daný moment nulový. Čím větší je N , tím méně je netriviálních invariantů. Pro $N = \infty$ je pouze jeden netriviální moment, a to $c_{pp} \neq 0$. Je důležité používat pouze netriviální momenty, a to z důvodů konstrukce rotačních invariantů, viz dříve. Následující věta shrnuje postup, jak se při vytváření báze invariantů vyhnout nulovým momentům pro N -fold symetrické objekty.

7.5. Invarianty na affinní transformaci

Věta 7.12. Nechť máme komplexní momenty až do řádu $r \geq 2$. Bázi \mathcal{B} rotačních invariantů až do řádu r pro N -fold symetrické objekty lze konstruovat následujícím způsobem

$$\left(\forall p, q : p \geq q \wedge p + q \leq r \wedge k = \frac{p - q}{N} \in \mathbb{Z} \right) (\Phi(p, q) = c_{pq} c_{q_0 p_0}^k),$$

kde p_0, q_0 jsou libovolné indexy splňující

$$c_{p_0 q_0} \neq 0, \quad p_0 - q_0 = N, \quad p_0 + q_0 \leq r.$$

Pokud bychom uvažovali nesymetrické objekty, tak v předchozí větě představuje člen

$$c_{q_0 p_0}^{\frac{p-q}{N}}$$

N -tou odmocninu z komplexního čísla. Jenže nevíme kterou, protože jich je N , a z tohoto důvodu se tento způsob tvorby báze pro nesymetrické objekty nepoužívá. Pokud je $\frac{p-q}{N} \in \mathbb{Z}$, tak uvedený člen není odmocnina. Je důležité volit správně příznaky s přihlédnutím k N -fold symetrickým objektům, tzn. pro různé typy objektů se hodí jiné volby indexů p_0 a q_0 .

7.5 Invarianty na affinní transformaci

Affinní transformace je dána transformačními vztahy

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

Vlastností affinní transformace je, že zachovává rovnoběžnost a také je dobrou approximací perspektivní projekce v případě hodně vzdálených malých objektů. Perspektivní transformace popisuje projekci 3D objektů na 2D plochu centrální kamerou. Protože invarianty na perspektivní projekci neexistují pro konečnou ani nekonečnou množinu momentů (formálně existují jako nekonečná řada momentů kladných i záporných indexů), tak tuto projekci nelze použít. Z tohoto důvodu je affinní transformace důležitá.

7.5.1 Grafová metoda

Uvažujme obraz $f(x, y)$ a dva libovolné body $(x_1, y_1), (x_2, y_2)$ z nosiče tohoto obrazu. Definujme tzv. křížový součin (*cross-product*) vztahem

$$T_{12} = x_1 y_2 - x_2 y_1.$$

Po působení affinní transformací získáváme vztah $\tilde{T}_{12} = J \cdot T_{12}$, kde J je Jakobián transformace. Hlavní myšlenka generování affinních invariantů pomocí grafové metody je následující. Předpokládejme několik bodů a integrujme jejich křížové součiny přes nosič obrazu f . Tyto integrály lze vyjádřit pomocí momentů a pokud se zbavíme Jakobiánu transformace vhodnou normalizací, získáme affinní invarianty. Předpokládejme $N \geq 2$ a definujme funkcionál I vztahem

$$I(f) = \int_{\mathbb{R}^{2N}} \prod_{k,j=1}^N T_{kj}^{n_{kj}} \cdot \prod_{i=1}^N f(x_i, y_i) dx_i dy_i,$$

kde n_{kj} je nezáporné celé číslo. Stačí uvažovat pouze $j > k$, protože pro křížové součiny platí $T_{kj} = -T_{jk}$ a $T_{kk} = 0$. Pokud na tento funkcionál budeme působit affinní transformací, získáme

$$\tilde{I} = J^\omega |J|^N I(f), \quad \omega = \sum_{k,j} n_{kj},$$

kde člen ω nazýváme vahou invariantu a N nazýváme stupněm invariantu. Pokud funkcionál I normalizujeme pomocí momentu nultého řádu $\mu_{00}^{\omega+N}$, získáme vztah

$$\left(\frac{I(\tilde{f})}{\mu_{00}^{\omega+N}} \right) = (\operatorname{sgn} \delta)^\omega \left(\frac{I(f)}{\mu_{00}^{\omega+N}} \right).$$

Pokud je ω sudé, jedná se o pravé invarianty, v opačném případě se jedná o pseudoinvarianty. Pokud má daná transformace záporný Jakobián, tak získáme převrácený obraz, ale většinou bývá Jakobián kladný. Uvedené invarianty jsou nulové pro objekty s axiální symetrií.

Myšlenka předchozí metody má analogii v teorii grafů. Každý invariant generovaný uvedeným vztahem lze vyjádřit pomocí grafu, kde body (x_k, y_k) představují vrcholy grafu a křížový součin T_{kj} představuje hranu grafu. Pokud je $n_{kj} > 1$, tak příslušný člen $T_{kj}^{n_{kj}}$ představuje n_{kj} hran spojujících k -tý a j -tý vrchol. Celkem tedy víme, že počet uzlů grafu odpovídá stupni invariantu N a celkový počet hran grafu je roven váze invariantu ω .

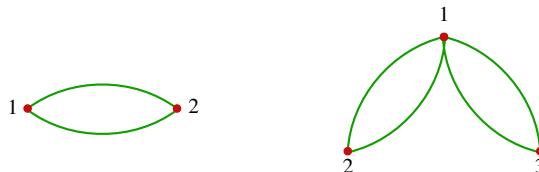
Příklad 7.13. Nechť $N = 2$ a $n_{12} = 2$, potom

$$I(f) = \int_{\mathbb{R}^4} (x_1 y_2 - x_2 y_1)^2 \cdot f(x_1, y_1) f(x_2, y_2) dx_1 dy_1 dx_2 dy_2 = 2(m_{20} m_{02} - m_{11}^2).$$

Obdobně pro $N = 3$ a $n_{12} = 2, n_{13} = 2, n_{23} = 0$, získáváme

$$\begin{aligned} I(f) &= \int_{\mathbb{R}^6} (x_1 y_2 - x_2 y_1)^2 (x_1 y_3 - x_3 y_1)^2 \cdot f(x_1, y_1) f(x_2, y_2) f(x_3, y_3) dx_1 dy_1 dx_2 dy_2 dx_3 dy_3 = \\ &= m_{20}^2 m_{04} - 4m_{20} m_{11} m_{13} + 2m_{20} m_{02} m_{22} + 4m_{11}^2 m_{22} - 4m_{11} m_{02} m_{31} + m_{02}^2 m_{40}. \end{aligned}$$

Vyjádření těchto invariantů pomocí grafů je uvedeno na obrázku 7.1.



Obr. 7.1: Grafy odpovídající invariantům z uvedených příkladů.

Poznámka 7.14 (Závislosti invariantů). Když hledáme invarianty potřebujeme, aby byly pokud možno nezávislé. Je tedy třeba z množiny všech invariantů odstranit invarianty triviální (tzn. nulové nebo identické s jinými invarianty), reducibilní (tzn. invarianty, které jsou produktem nebo lineární kombinací jiných invariantů) a irreducibilní (tzn. invarianty tvořené polynomy nebo polynomy produktů jiných invariantů). Pokud postupně odstraňujeme invarianty podle uvedených vlastností, tzn. nejdřív triviální, pak reducibilní a nakonec irreducibilní, tak nám zbude množina nezávislých invariantů. Nejtěžším krokem je odstranění irreducibilních invariantů. Odstraněním uvedených závislostí přijdeme o drtivou většinu invariantů.

7.5.2 Afinní invarianty skrz normalizaci

Je mnoho možností jak určit normalizační podmínky. Abychom mohli získat invarianty na affinní transformaci pomocí normalizace, musíme provést dekompozici affinní transformace. Existuje několik možných dekompozicí, např. můžeme affinní transformaci popsat jako následující řadu transformací:

1. horizontální a vertikální translace: $m_{10} = m_{01} = 0$,

7.6. Invarianty na elastické transformace

2. změna měřítka: $c_{00} = 0$,
3. první rotace: $c_{20} = \mathbb{R}^+$,
4. stretching: $c_{20} = m_{20} - m_{02} = 0 \iff m_{20} = m_{02} \iff \mu_{20} = \mu_{02}$,
5. druhá rotace: $c_{21} = \mathbb{R}^+$, tato normalizace je nejtěžší, protože už nemáme momenty malých řádů a používáme moment 3. řádu, který je celkem nestabilní,
6. zrcadlení.

V praxi pro normalizaci druhé rotace hledáme moment c_{ij} , který má co nejmenší řád a je prokazatelně nulový.

7.6 Invarianty na elastické transformace

Jako elastickou transformaci si můžeme představit např. text na nerovném povrchu, rybí oko, atp. Formálně tyto transformace můžeme zapsat následovně

$$\tilde{f}(x, y) = f(r(x, y)),$$

kde $r(x, y)$ je polynom řádu většího než jedna. Uvažujme nyní transformaci ve tvaru $\tilde{x} = ax^2$, potom

$$\tilde{m}_p = \int (\tilde{x})^p f(\tilde{x}) d\tilde{x} = \left| \begin{array}{l} \tilde{x} = ax^2 \\ d\tilde{x} = 2axdx \end{array} \right| = \int (ax^2)^p f(x) 2ax dx = 2a^{p+1} \int x^{2p+1} f(x) dx = 2a^{p+1} m_{2p+1},$$

tzn. moment se transformuje na moment vyššího řádu a to by „slušný“ moment dělat neměl. Vždy by mělo docházet k transformaci na momenty stejného nebo nižšího řádu. Protože tomu tak v uvedeném případě není, nemůže existovat invariant z konečného počtu momentů.

7.7 Invarianty na konvoluci

Předpokládejme deformaci obrazu konvolucí, tzn. deformaci ve tvaru

$$g(x, y) = (f * h)(x, y),$$

kde $h(x, y)$ je tzv. *point spread function* (PSF) systému, viz kapitola 4. Budeme hledat invarianty na konvoluci, tzn.

$$I(f * h) = I(f),$$

pro libovolnou středově symetrickou PSF funkci h , tzn. 2-fold symetrickou. Předpoklad středově symetrické PSF funkce není nijak omezující, protože běžné senzory a optické systémy disponují středovou symetrií. Dále po PSF požadujeme, aby měla jednotkový integrál. Tyto invarianty jsou občas nazývány invarianty na rozmazání (*blur invariants*) a to z toho důvodu, že obvyklé PSF funkce se chovají jako *low-pass* filtr.

Pro komplexní momenty (pro geometrické momenty platí stejný vztah) deformovaného obrazu platí

$$c_{pq}^{(g)} = \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} c_{kj}^{(h)} c_{p-k, q-j}^{(f)}.$$

Otázkou tedy je, jak eliminovat člen $c_{kj}^{(h)}$. Postup jak tento člen eliminovat je popsán v následující větě.

Věta 7.15. Nechť je funkcionál $c : L_1(\mathbb{R}^2) \times \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{R}$ definován pro $(p+q)$ liché

$$c_{(p,q)}^{(f)} = \mu_{pq}^{(f)} - \frac{1}{\mu_{00}^{(f)}} \sum_{n=0}^p \sum_{\substack{m=0 \\ 0 < n+m < p+q}}^q \binom{p}{n} \binom{q}{m} c_{(p-n,q-m)}^{(f)} \cdot \mu_{nm}^{(f)},$$

a pro $(p+q)$ sudé $c_{(p,q)}^{(f)} = 0$. Potom pro libovolnou funkci f , nezáporná celá čísla p, q , a středově symetrickou PSF funkci h platí

$$c_{(p,q)}^{(f*h)} = c_{(p,q)}^{(f)}.$$

7.7.1 Invarianty na konvoluci v FT doméně

Budeme se zabývat konstrukcí invariantů pomocí projekčního operátoru P_N , který je definovaný vztahem

$$(P_N f)(r, \theta) = \frac{1}{N} \sum_{j=1}^N f(r, \theta + \alpha_j),$$

tzn. jedná se o součet N otočení daného obrazu. Přejděme do frekvenční oblasti pomocí Fourierovy transformace

$$g(x, y) = (f * h)(x, y) \implies G = F \cdot H,$$

kde $G = \mathcal{F}(g)(u, v)$, $F = \mathcal{F}(f)(u, v)$ a $H = \mathcal{F}(h)(u, v)$. Potom jako invariant na konvoluci můžeme ve frekvenční oblasti volit

$$I_N^{(f)}(u, v) = \frac{\mathcal{F}(f)(u, v)}{\mathcal{F}(P_N f)(u, v)},$$

což plyne z následujících rovností

$$I_N^{(f*h)}(u, v) = \frac{\mathcal{F}(f * h)(u, v)}{\mathcal{F}(P_N(f * h))(u, v)} = \begin{cases} \mathcal{F}(f * h) = \mathcal{F}(f)\mathcal{F}(h) \\ P_N(f * h) = (P_N f) * h \end{cases} = \frac{\mathcal{F}(f)(u, v)\mathcal{F}(h)(u, v)}{\mathcal{F}(P_N f)(u, v)\mathcal{F}(h)(u, v)} = I_N^{(f)}(u, v).$$

Ve jmenovateli vztahu pro invariant mohou být nuly, ale protože předpokládáme, že jsme na prostoru funkcí s omezeným nosičem (*image space*), tak pro takové funkce má Fourierova transformace neomezený nosič a může být nulová pouze na množině míry nula.

7.8 Ortogonální momenty

Uvažujme množinu ortogonálních polynomů a označme ji $\{\mathcal{P}_{pq}(x, y)\}$. Potom ortogonální momenty definujeme klasicky

$$v_{pq} = \int_{\Omega} \int \mathcal{P}_{pq}(x, y) f(x, y) dx dy.$$

Motivací pro zavedení ortogonálních momentů je stabilní výpočet pomocí rekurentních formulí a jednodušší a stabilnější zpětná rekonstrukce obrazu. Rozlišujeme dva typy ortogonalnosti: na jednotkovém čtverci a na jednotkovém disku.

7.8. Ortogonální momenty

7.8.1 Ortogonální momenty na čtverci

Ortogonální momenty na čtverci jsou definovány následovně

$$v_{pq} = n_p n_q \int_{\Omega} \int p_p(x) p_q(y) f(x, y) dx dy,$$

kde $p_k(x)$ je systém jedno dimenzionálních ortogonálních polynomů. Běžně se používají Legendreovy, Čebyševovy nebo Gegenbauerovy, které jsou všechny definovány na intervalu $(-1, 1)$. Další možností jsou Jacobiho polynomy na intervalu $(0, 1)$ nebo $(-1, 1)$, Laguerreovy na intervalu $(0, \infty)$ nebo Hermiteovy na intervalu $(-\infty, \infty)$. Dále budeme uvažovat pouze Legendreovy polynomy definované vztahem

$$p_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n.$$

Ortogonalita plyne z následujícího vztahu

$$\int_{-1}^1 p_m(x) p_n(x) dx = \frac{2}{2n+1} \delta_{mn}.$$

Pro výpočet Legendreových momentů platí vztah ve tvaru

$$\begin{aligned} \lambda_{mn} &= \frac{(2m+1)(2n+1)}{4} \int_{-1}^1 \int_{-1}^1 p_m(x) p_n(y) f(x, y) dx dy, \\ \lambda_{mn} &= \frac{(2m+1)(2n+1)}{4} \sum_{p=0}^m \sum_{q=0}^n a_{mp} a_{nq} m_{pq}, \end{aligned}$$

7.8.2 Ortogonální momenty na disku

Ortogonální momenty na disku jsou definovány následovně

$$v_{pq} = n_p n_q \int_0^{2\pi} \int_0^1 R_{pq}(r) e^{-iq\varphi} f(r, \varphi) r dr d\varphi,$$

kde $R_{pq}(r)$ je radiální část a $e^{-iq\varphi}$ je úhlová část. Běžně se používají: Zernikeovy, pseudo-Zernikeovy, OG Fourier-Mellineho, atd. Zernikeovy polynomy jsou definované vztahem

$$\begin{aligned} A_{nl} &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 V_{nl}^*(r, \varphi) f(r, \varphi) r dr d\varphi, \\ V_{nl}(r, \varphi) &= R_{nl}(r) e^{-il\varphi}, \\ R_{nl}(r) &= \sum_{s=0}^{(n-l)/2} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+l}{2} - s\right)! \left(\frac{n-l}{2} - s\right)!} \cdot r^{n-2s}. \end{aligned}$$

Ortogonalita plyne z následujícího vztahu

$$\int_0^{2\pi} \int_0^1 V_{nl}^*(r, \varphi) V_{mk}(r, \varphi) r dr d\varphi = \frac{\pi}{n+1} \delta_{mn} \delta_{kl}.$$

Pro výpočet Zernikeových momentů platí vztah ve tvaru

$$A_{nl} = \frac{n+1}{\pi} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} V_{nl}^*(r, \varphi) f(x, y).$$

Jedná se o mapování Kartézských souřadnic x, y do polárních souřadnic r, φ . Celý obrázek je namapován uvnitř jednotkového disku, z čehož plyne, že jsou tyto momenty invariantní na posunutí a škálování. Další vlastností Zernikeových momentů je tzv. rotační vlastnost

$$\tilde{A}_{nl} = e^{-il\theta} A_{nl},$$

tzn. velikost se zachovává, ale posunuje se fáze o $l\theta$.

7.8.3 Rekonstrukce obrazu z geometrických momentů

První možností je rekonstrukce přímo z geometrických momentů, které lze docílit řešením následujících rovnic

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y).$$

Tento způsob ovšem funguje pro velmi malé obrázky, protože pro větší je systém rovnic špatně podmíněný. Další možností je rekonstruovat obraz skrze Fourierovu transformaci

$$F(u, v) = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \frac{(-2\pi i)^{p+q}}{p! q!} \left(\frac{u}{M}\right)^p \left(\frac{v}{N}\right)^q m_{pq}.$$

7.9 Algoritmy pro výpočet momentů

7.9.1 Dekompoziční metody

Objekt je rozložen na K disjunktních bloků B_k (obvykle obdélníkových) takových, že platí

$$m_{pq}^{\Omega} = \sum_{k=1}^K m_{pq}^{B_k},$$

kde $K \ll N^2$. Jednotlivé dekompoziční metody se od sebe liší dekompozičním algoritmem, tvarem bloků nebo způsobem, jakým jsou momenty daného bloku počítány. Výpočetní náročnost dekompozičních metod se často ignoruje a doufá se, že pro jeden blok bude $m_{pq}^{B_k} \sim O(1)$ a celkem tedy $m_{pq}^{\Omega} \sim O(K)$. Nicméně toto nemusí platit a náročnost může být vyšší. Dekompozice je účinná, pokud se počítá velké množství momentů objektu, ale některé objekty efektivně dekomponovat nelze.

1. **Delta metoda (Zakaria et al.):** Jedná se o dekompozici na řádky, která lze zapsat ve tvaru

$$m_{pq}^{G_k} = y_0^q \sum_{i=0}^{x_0+\delta-1} i^p.$$

Pro výpočet této sumy existují rekurzivní formule, viz prezentace z přednášky. Nicméně výpočet lze zjednodušit přímou integrací, tzn.

$$m_{pq}^{B_k} = y_0^q \int_{i=0}^{x_0+\delta} i^p dx = \frac{y_0^q}{p+1} ((x_0 + \delta)^{p+1} - x_0^{p+1}).$$

2. **Obdélníkové bloky:** Jedná se o dekompozici na množinu řádků, které mají stejný začátek a konec. Tuto dekompozici můžeme zapsat ve tvaru

$$m_{pq}^{G_k} = \sum_{i=0}^{x_1} i^p \cdot \sum_{j=0}^{y_1} j^q,$$

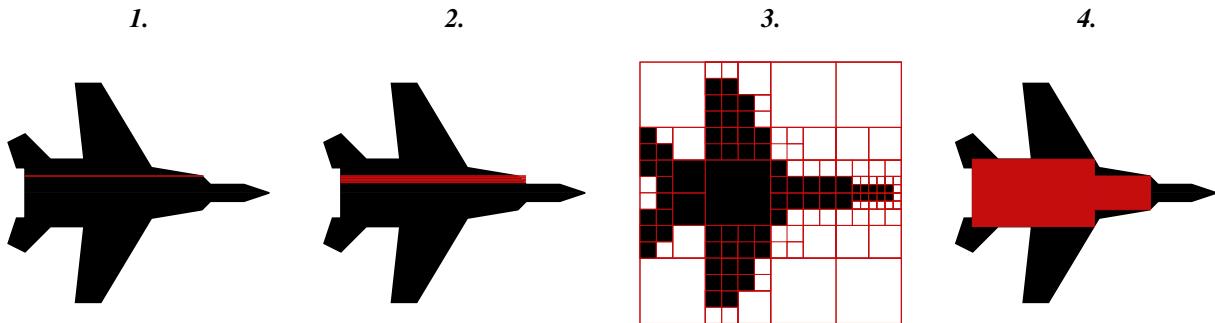
a zjednodušení můžeme opět provést přímou integrací, čímž získáme vztah

$$m_{pq}^{B_k} = \frac{1}{(p+1)(q+1)} (x_1^{p+1} - x_0^{p+1})(y_1^{q+1} - y_0^{q+1}).$$

3. **Hierarchická dekompozice:** Tato dekompoziční metoda rozkládá obraz na homogenní čtverce a existují verze *bin-tree* a *quad-tree*, které se liší tím, na kolik čtverců dělíme. Moment pro blok po zjednodušení přímou integrací je ve tvaru jako pro obdélníkové bloky.

7.9. Algoritmy pro výpočet momentů

4. **Morfologická dekompozice:** Rekurzivní dekompozice na největší vepsané čtverce, kde jejich středy hledáme pomocí morfologické operace eroze. Moment pro blok po zjednodušení přímou integrací je opět ve tvaru jako pro obdélníkové bloky.



Obr. 7.2: Ukázky principu dekompozice pro uvedené metody.

7.9.2 Boundary-based metody

Tyto metody vycházejí z tzv. Greenova teorému

$$\oint_{\partial\Omega} h(x, y) \, dx + g(x, y) \, dy = \iint_{\Omega} \left(\frac{\partial g}{\partial x} - \frac{\partial h}{\partial x} \right) \, dx \, dy.$$

Pokud budeme volit funkce h a g ve speciálním tvaru, dostaneme

$$g(x, y) = \frac{x^{p+1} y^q}{p+1} \quad \wedge \quad h(x, y) = 0 \quad \Rightarrow \quad m_{pq}^{\Omega} = \frac{1}{p+1} \oint_{\partial\Omega} x^{p+1} y^q \, dy.$$

Křívkový integrál lze počítat pomocí sumace pixel po pixelu, případně lze použít nějakou approximaci, např. polygonální.

Další metoda je založena na diskrétním Greenovu teorému, který říká

$$m_{pq}^{\Omega} = \sum_{(x,y) \in \partial\Omega_+} y^q \cdot \sum_{i=1}^x i^p - \sum_{(x,y) \in \partial\Omega_-} y^q \cdot \sum_{i=1}^x i^p,$$

kde množiny $\partial\Omega_+$ a $\partial\Omega_-$ jsou definovány následovně

$$\partial\Omega_+ = \{(x, y) | (x, y) \notin \Omega \wedge (x+1, y) \in \Omega\}, \quad \partial\Omega_- = \{(x, y) | (x, y) \in \Omega \wedge (x+1, y) \notin \Omega\}.$$

Tato metoda je ekvivalentní dekompoziční delta metodě a lze ji zjednodušit pomocí přímé integrace a předpočítání.

Složitost boundary-based metod závisí na délce hranice a předpokládá se, že detekce hranice je rychlá. Tyto metody jsou účinné pro objekty s jednoduchou hranicí a můžou být použity i pro malý počet momentů, což je rozdíl oproti dekompozičním metodám. Pro objekty se složitou hranicí jsou neúčinné.

8

Waveletová transformace

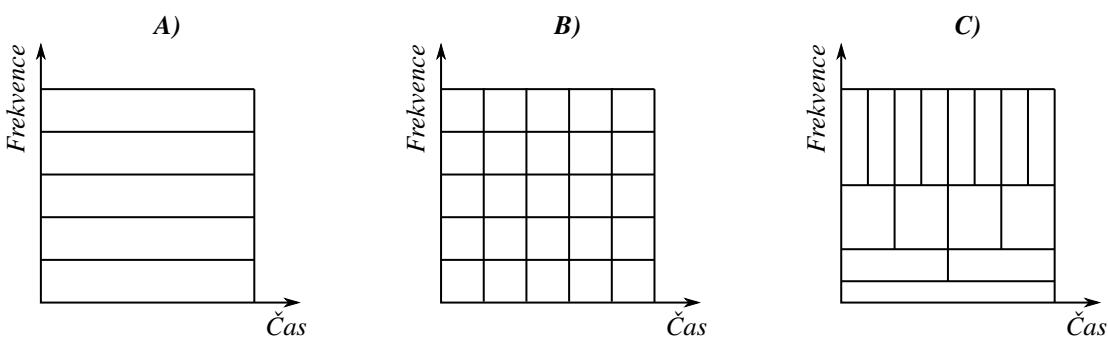
V oblasti zpracování signálů po dlouhou dobu dominovala Fourierova transformace (FT). S její pomocí získáme informaci o frekvencích obsažených v analyzovaném signálu, nicméně nedokážeme zjistit čas, ve kterém se tyto frekvence vyskytují. To je dán povahou bázové funkce, kterou je v FT komplexní exponenciála, tedy periodická funkce s energií rozloženou po celém jejím průběhu. Kvůli této vlastnosti nelze získat odezvu od konkrétního místa (času) v signálu. Poměrně jednoduchým řešením je neprovádět analýzu celého signálu, ale postupně po menších částech, definovaných plovoucím oknem. Tento přístup je znám jako krátkodobá Fourierova transformace (*short-time Fourier transform, STFT*)

$$STFT_f(a, b) = \int_{-\infty}^{\infty} f(t) w^*(t - b) e^{-2\pi i at} dt,$$

kde $w(t)$ značí výběrové okno a člen $h(t) = w^*(t) e^{-2\pi i at}$ představuje prvek báze. Zde ale nastává problém s volbou velikosti plovoucího okna. Čím je okno menší, tím je lepší časové rozlišení analýzy. Se zmenšujícím se oknem ale také ubývá množství frekvencí, které lze ve zkoumaném signálu identifikovat. Toto platí samozřejmě i naopak, tedy se zvětšujícím se oknem se zvyšuje frekvenční rozlišení a zmenšuje rozlišení časové. Tento problém je úzce svázán s tzv. Heisenbergovým principem neurčitosti

$$\Delta t * \Delta f \geq \frac{1}{4\pi},$$

kde Δt značí časové rozlišení a Δf frekvenční rozlišení. Tento nedostatek STFT řeší waveletová transformace (*wavelet transform, WT*) volbou jiné bázové funkce, tzv. wavelety, která má většinu energie umístěnou v malém konečném intervalu. Rozdíl mezi charakteristikami jednotlivých transformací je znázorněn na obr. 8.1.



Obr. 8.1: Porovnání časového a frekvenčního rozlišení jednotlivých transformací: **A)** FT, **B)** STFT a **C)** WT.

8.1 Spojitá waveletová transformace

Spojitá waveletová transformace rozkládá signál pomocí dilatovaných a posunutých wavelet. Název waveleta pochází z anglického *wavelet*, kde *wave...* značí to, že funkce osciluje, a *...let* značí to, že je funkce dobře lokalizovatelná kolem 0, tzn. rychle „mizí“. Formálně je waveleta ψ funkce z prostoru funkcí integrovatelných s druhou mocninou, tzn. $\psi \in L^2(\mathbb{R})$, s následujícími vlastnostmi:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0, \quad \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty, \quad \int_{-\infty}^{\infty} \frac{|\psi(t)|}{|t|} dt < \infty, \quad \mathcal{F}(\psi) = \begin{cases} 0 & \dots \text{ v } 0, \\ 0 & \dots \text{ v } \infty. \end{cases}$$

Definujme waveletovou bázi, která je tvořena wavelety $\psi_{a,b}$, které jsou odvozeny z mateřské wavelety ψ (*mother wavelet*) posunutím o faktor b a dilatovaných faktorem a , tzn. ve tvaru

$$\mathcal{D} = \left\{ \psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \text{ pro } a \in \mathbb{R}^+, b \in \mathbb{R} \right\}.$$

Dosazením do obecného předpisu pro integrální transformaci, získáme rovnici spojité waveletové transformace ve tvaru

$$W_f(a, b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt = \int_{-\infty}^{\infty} f(t) \psi_{a,0}^*(t-b) dt,$$

kde symbol $*$ značí komplexní sdružení. Kompaktně lze uvedený vztah zapsat následovně

$$W_f(a, b) = \langle f(t), \psi_{a,b} \rangle.$$

Pokud uvedený vztah porovnáme se vztahem pro STFT, tak je jasné, že jediná změna je ve volbě bazických funkcí. Dále definujme inverzní waveletovou transformaci

$$f(t) = c \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} W_f(a, b) \psi_{a,b}(t) da db,$$

kde konstanta c závisí na waveletu ψ . Uvedená waveletová transformace je ale vysoce redundantní, protože počítá transformaci pro všechny možné posuny b a všechny možné dilatace faktorem a . Z tohoto důvodu její přímá implementace nemá velký význam a přistupuje se k diskretizaci faktorů a, b .

8.2 Diskrétní waveletová transformace

Jak již bylo řečeno, spojitá waveletová transformace vypočtená pro všechna možná měřítka a posunutí wavelety produkuje obrovské množství dat. Je tedy nutno provést výpočet jen pro určitou množinu pozic a měřítek wavelety. K tomuto slouží tzv. Mallatův algoritmus, kde změna měřítka a posuv probíhá na dvojkové mřížce, tzv. dyadické mřížce. Díky vhodné volbě závislosti mezi posuvem a změnou měřítka vytvoříme neredundantní dekompozici a z posunutých a dilatovaných wavelet tak vytvoříme ortonormální bázi. Parametr a vzorkujeme na logaritmické stupnici následovně $a = a_0^{-m}$. Parametr b poté vzorkujeme hustěji pro malá a a používáme vztah $b = n \cdot a_0^{-m} \cdot b_0$. Volba parametrů $a_0 = 2$ a $b_0 = 1$ vede na dyadickou síť.

Pokud tedy pro diskretizaci zvolíme dyadickou síť, dostáváme ortonormální bázi \mathcal{D} prostoru $L^2(\mathbb{R})$ ve tvaru

$$\mathcal{D} = \left\{ \psi_{m,n}(t) = \sqrt{2^m} \psi(2^m t - n) \text{ pro } m, n \in \mathbb{Z}, -\infty < m, n < \infty \right\},$$

tzn. báze se skládá z binárně škálovaných wavelet, tzn. zmenšují se o faktor 2, s dyadickým posunem, tzn.

posunem o $\frac{n}{2^m}$. Když máme ortonormální bázi, tak můžeme funkci $f(t)$ vyjádřit pomocí této báze ve tvaru

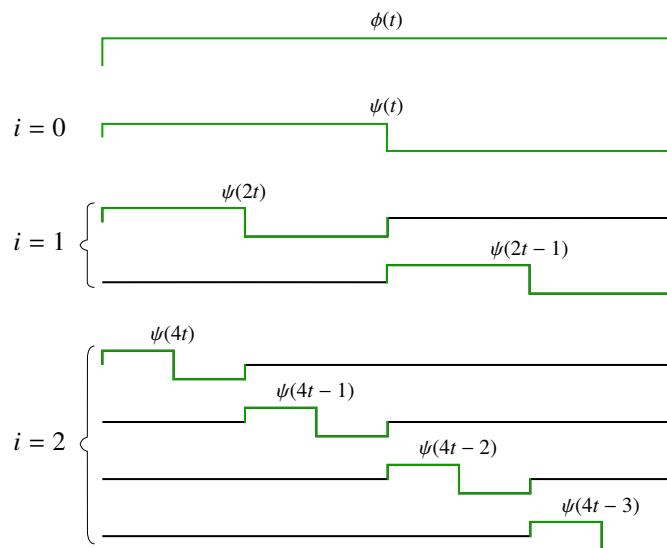
$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{m,n} \psi_{m,n}, \quad c_{m,n} = \langle f(t), \psi_{m,n} \rangle,$$

tzn. koeficienty $c_{m,n}$ jsou výstupem diskrétní waveletové transformace funkce $f(t)$. Nutno podotknout, že takto popsaná funkce $f(t)$ je stále spojitá, protože jsme diskretizovali pouze proměnné a, b . Dále se budeme zabývat tím, jak navzorkovat funkci f .

Uvedenou transformaci můžeme ještě zjednodušit. Zavedeme tzv. kompaktní dyadicou transformaci. Předpokládejme funkci $f(t)$ a waveletu $\psi_{m,n}(t)$, která je nenulová na jednotkovém intervalu $\langle 0, 1 \rangle$. Dále předpokládejme $j = 2^m + n$, kde $m = 0, 1, \dots$ a $n = 0, 1, \dots, 2^j - 1$. Potom můžeme zapsat waveletu ve tvaru

$$\psi_j(t) = \sqrt{2^m} \psi(2^m t - n),$$

kde pro libovolné j je m největší celé číslo, které splňuje $2^m \leq j$, a pro n platí vztah $n = j - 2^m$. Na obrázku 8.2 jsou zobrazeny rozložení intervalu $\langle 0, 1 \rangle$ pomocí Haarovy wavelety pro různé hodnoty j .



Obr. 8.2: Rozložení intervalu $\langle 0, 1 \rangle$ pomocí Haarovy wavelety pro různé hodnoty j .

Nyní můžeme přistoupit k diskretizaci funkce f , tzn. k zavedení kompaktní waveletové transformace ve tvaru

$$f(t) = \sum_{j=1}^N c_j \psi_j, \quad c_j = \langle f(t), \psi_j \rangle,$$

kde výstupem už je diskrétní funkce. Následující tabulka uvádí srovnání Fourierových řad a transformací s transformacemi waveletovými.

8.2. Diskrétní waveletová transformace

Fourierova transformace	spojitá funkce \Rightarrow spojité funkce
Fourierova řada	periodická funkce \Rightarrow řada koeficientů
Diskrétní Fourierova transformace	navzorkovaná funkce \Rightarrow navzorkované spektrum
Spojitá waveletová transformace	spojitá funkce \Rightarrow spojité a, b
Waveletová řada	spojitá funkce \Rightarrow řada koeficientů
Diskrétní waveletová transformace	navzorkovaná funkce \Rightarrow konečná řada koeficientů

8.2.1 Vnořené podprostory (*Multi resolution analysis, MRA*)

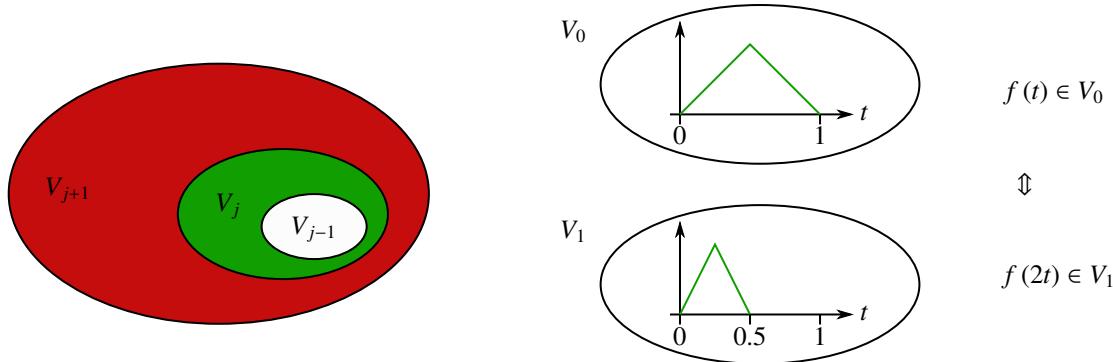
Prostor $L^2(\mathbb{R})$ lze rozložit na uzavřené disjunktní podprostory V_i , viz obr. 8.3, kde každý podprostor V_i odpovídá jednomu měřítku, tzn.

$$L^2(\mathbb{R}) = \bigcup_{m \in \mathbb{Z}} V_m, \quad \bigcap_{m \in \mathbb{Z}} V_m = \{0\}.$$

Pro tyto prostory navíc platí následující vztahy

$$f(t) \in V_0 \implies f(2^k t) \in V_k, \quad f(t) \in V_k \implies f(2t) \in V_{k+1}, \quad f(t) \in V_k \iff f(t - j) \in V_k.$$

Znázornění uvedených vlastností je na obr. 8.3.



Obr. 8.3: Vnořené prostory a jejich vlastnosti.

Uvedené podprostory lze plně určit pomocí tzv. škálovací funkce (též někdy nazývaná vyhlazovací funkce)

$$\phi_{ij}(t) = \sqrt{2^i} \phi(2^i t - j),$$

která se také někdy nazývá otcovská waveleta (*father wavelet*). Škálovací funkce musí být kompaktní, tzn. nulová krom určitého konečného intervalu, musí rychle klesat jak ϕ , tak i $\mathcal{F}(\phi)$, a musí platit

$$\int_{-\infty}^{\infty} \phi(t) dt \neq 0 \quad (= 1).$$

Tato funkce (se svými posuny o j) tvoří ortonormální bázi podprostoru V_i . Označme symbolem $P_i(f)$ ortonormální projekci funkce f do podprostoru V_i , potom platí

$$f = \lim_{i \rightarrow \infty} P_i(f).$$

Stejně jako škálovací funkce tvoří se svými posuny ortonormální bázi podprostoru V_i , tak i waveletová funkce $\psi_{i,j}$ tvoří se svými posuny ortonormální bázi podprostoru W_i , kde W_i je ortonormálním doplňkem prostoru V_i do prostoru V_{i+1} , tzn. platí $V_{i+1} = V_i \oplus W_i$.

8.2.2 Dekompozice funkce f

Nyní se budeme zabývat tím jak dekomponovat funkci f následujícím vztahem

$$P_{V_j}f = P_{V_{j_0}}f + \sum_{k=j_0}^{j-1} P_{W_k}f, \quad (8.1)$$

kde P_{V_i} představuje ortonormální projekci f do podprostoru V_i a $P_{W_j}f$ představuje ortonormální projekci f do podprostoru W_j . V předchozím vzorci představuje člen $P_{V_{j_0}}f$ základ a členy $P_{W_k}f$ představují detailly různého měřítka. Pro uvedené ortonormální projekce platí následující vztahy

$$\begin{aligned} (P_{V_j}f)(t) &= \sum_{k=0}^{2^j-1} c_{j,k} \cdot \phi_{j,k}(t), & c_{j,k} &= \int_{-\infty}^{\infty} f(t) \phi_{j,k}(t) dt, \\ (P_{W_j}f)(t) &= \sum_{k=0}^{2^j-1} d_{j,k} \cdot \psi_{j,k}(t), & d_{j,k} &= \int_{-\infty}^{\infty} f(t) \psi_{j,k}(t) dt. \end{aligned}$$

V praxi se škálovací funkce a waveletová funkce rozkládají pomocí filtrů. Tento rozklad lze zapsat jako dilatační rovnice

$$\phi(t) = \sqrt{2} \sum_j h_j \cdot \phi(2t - j), \quad \psi(t) = \sqrt{2} \sum_j g_j \cdot \psi(2t - j),$$

kde platí $\phi(2t - j), \psi(2t - j) \in V_1$, $\phi(t) \in V_0$ a $\psi(t) \in W_0$. Dále h_j, g_j jsou tzv. *quadrature mirror filters* (platí $|H|^2 + |G|^2 = 1$), kde h_j je *low-pass filter*, g_j je *high-pass filter* a platí pro ně vztahy

$$\sum_j |h_j|^2 < \infty, \quad \sum_j h_j = \sqrt{2}, \quad \sum_j g_j = 0.$$

Je tedy vidět, že škálovací funkce a waveleta je určena dvojcí filtrů. Jako příklad uveďme Haarovo waveletu, která je definována následovně

$$\psi(t) = \begin{cases} 1 & \dots \quad t \in \left(0, \frac{1}{2}\right) \\ -1 & \dots \quad t \in \left(\frac{1}{2}, 1\right) \\ 0 & \dots \quad \text{jinak} \end{cases}, \quad \phi(t) = \begin{cases} 1 & \dots \quad t \in (0, 1) \\ 0 & \dots \quad \text{jinak} \end{cases}.$$

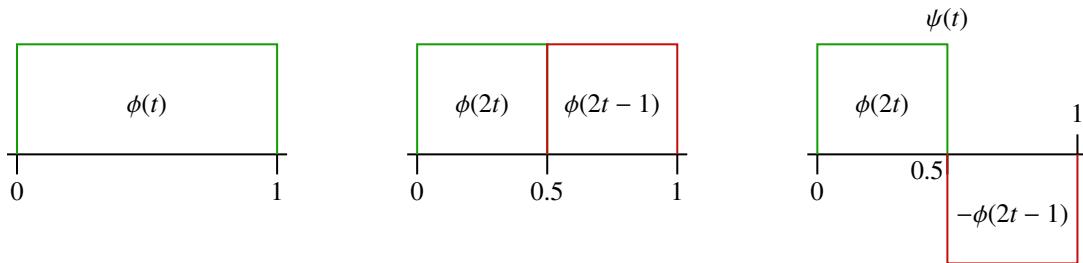
Pokud definujeme filtry, můžeme pomocí nich tuto waveletu a škálovací funkci vyjádřit

$$\begin{aligned} h &= \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right), & \phi(t) &= \sqrt{2} \sum_{j=0}^1 h_j \cdot \phi(2t - j) = \sqrt{2} \left(\frac{1}{\sqrt{2}} \phi(2t) + \frac{1}{\sqrt{2}} \phi(2t - 1) \right) = \phi(2t) + \phi(2t - 1), \\ q &= \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right), & \psi(t) &= \sqrt{2} \sum_{j=0}^1 g_j \cdot \phi(2t - j) = \sqrt{2} \left(\frac{1}{\sqrt{2}} \phi(2t) - \frac{1}{\sqrt{2}} \phi(2t - 1) \right) = \phi(2t) - \phi(2t - 1). \end{aligned}$$

Toto vyjádření je znázorněno na obr. 8.4. Pro filtry h_j, g_j platí následující vztahy

$$h_j = \int_{-\infty}^{\infty} \phi(t) \phi(2t - j) dt, \quad g_j = \int_{-\infty}^{\infty} \psi(t) \phi(2t - j) dt.$$

8.2. Diskrétní waveletová transformace

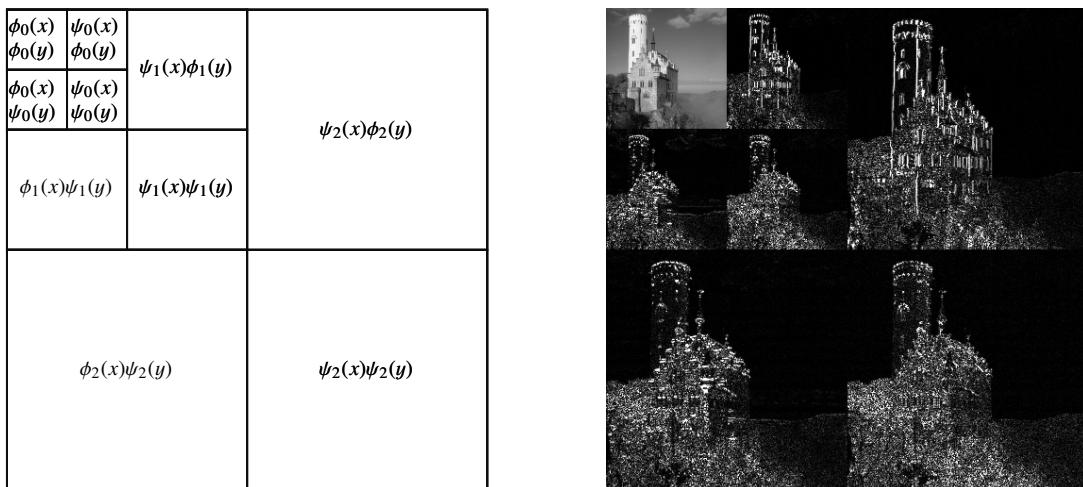


Obr. 8.4: Znázornění vyjádření Haarovy wavelety pomocí škálovací funkce a filtrů.

Díky filtrům h_j, g_j můžeme nalézt rekurentní vztahy pro výpočet koeficientů $c_{j,k}$ ortonormální projekce do prostoru V_j , resp. koeficientů $d_{j,k}$ ortonormální projekce do prostoru W_j ve vztahu (8.1), následovně

$$c_{j-1,k} = \sum_n h_{n-2k} \cdot c_{j,n}, \quad d_{j-1,k} = \sum_n g_{n-2k} \cdot c_{j,n}, \quad c_{j+1,k} = \sum_n h_{k-2n} \cdot c_{j,n} + \sum_n h_{k-2n} \cdot c_{j,n}.$$

Příklad dekompozice obrazu pomocí waveletové transformace je uveden na obr. 8.5.



Obr. 8.5: Ukázka dekompozice obrazu.

Poznámka 8.1. Waveletová transformace je tedy proces určení koeficientů $c_{j_0,k}, d_{j,k}$. Na wavelety klademe požadavek, aby byly kompaktní, tzn. měli konečný počet nenulových momentů, a byly lokalizovatelné v čase a frekvenci. Také požadujeme nulovost prvních m momentů, tzn.

$$0 = \int_{-\infty}^{\infty} \psi(t) dt = \dots = \int_{-\infty}^{\infty} t^{m-1} \psi(t) dt.$$

Dále očekáváme jednoduchou konstrukci a reprezentaci, invarianci vzhledem k některým operacím, hladkost, spojitost, diferencovatelnost, symetrii a dobré vlastnosti vzhledem k počtu nulových momentů.

Poznámka 8.2 (Rychlosť výpočtu). Rychlá Fourierova transformace (FFT) má náročnosť $O(n \cdot \log_2 n)$. Oproti tomu rychlá waveletová transformace (FWT) má náročnosť pouze $O(n)$.

8.3 Aplikace waveletové transformace

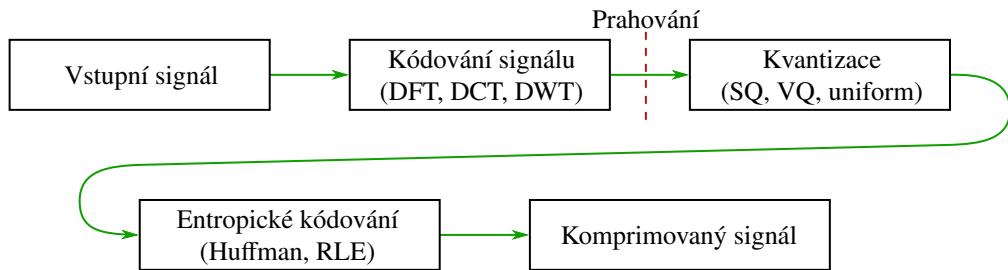
8.3.1 Kompresce

Při kompresi se snažíme eliminovat redundantní informaci a méně důležité informace. Rozlišujeme tři základní druhy redundancy.

1. Prostorová redundancy: sousední hodnoty v obrazu jsou korelované.
2. Frekvenční redundancy: frekvenční hodnoty ze stejného pixelu jsou korelované.
3. Časová redundancy: frames malé změny v sekvenci.

Waveletová transformace provádí dekorelací dat. Z vlastností diskrétní waveletové transformace víme, že koeficienty WT představují amplitudu wavelety v daném místě, rozlišení a čase.

První obecně známé využití waveletové transformace v kódování obrazu bylo její užití ve standardu FBI pro kompresi otisků prstů. Pro kompresi obecného obrazu se nejvíce ujal standard JPEG poprvé představený v roce 1992. Kompresce podle standardu JPEG je realizována kvantizací a kódováním koeficientů diskrétní kosinové transformace. Algoritmus je poměrně jednoduchý a lze dosáhnout dobrého kompresního poměru, nicméně je zde prostor k vylepšení, např. v kvalitě obrazu nebo odolnosti vůči chybám. I z těchto důvodů byl v roce 2000 představen standard JPEG 2000. Schéma komprese (obr. 8.6) je velmi podobné schématu JPEG, vstupní obraz je tedy transformován a výsledné koeficienty jsou kvantovány a zakódovány. Hlavní rozdíl je ve využití diskrétní waveletové transformace namísto kosinové a v použití kódování založeném na algoritmu EBCOT.



Obr. 8.6: Schéma komprese obrazu.

- **Prahování:** rozlišujeme ztrátovou a bezeztrátovou kompresi. Při ztrátové provádíme prahování a koeficienty, které jsou menší než daný práh, pokládáme rovny nule. Ostatní koeficienty pokládáme rovny 1.
- **Vektorová kvantizace (blokové kódování):** hledáme kódovou knihu $Y = \{y_i : i = 1, 2, \dots, N\}$ (y_i je kódové slovo), která by nejlépe reprezentovala danou množinu vektorů nD z \mathbb{R}^n . Jedná se o NP úplný problém. Používá se Linde-Buzo-Gray algoritmus, který je podobný algoritmu *k-means clustering*. Na začátku se určí velikost N a náhodně se vybere N kódových slov (centroidy). Poté proběhne shlukování a jako nová kódová slova se vezme průměr. Pokud se kódové slovo změnilo, tak opakujeme shlukování a aktualizaci kódových slov. Jinak končíme.
- **Kódování:** pro kódování se používá například RLE kódování (*run length coding*), které zaznamenává nejdělsí souvislý běh po spirále od středu. V takovém kódování stačí zaznamenávat první hodnotu a změny. Další možností je kódovat amplitudy koeficientů pomocí Huffmanova instantního kódu.

8.3. Aplikace waveletové transformace

8.3.1.1 Embedded Zerotree Wavelet Encoding

Jiný druh kódování, kdy modelujeme závislosti mezi koeficienty waveletové transformace. Předpokládáme, že waveletový koeficient je nevýznamný vzhledem k parametru T , pokud je v absolutní hodnotě menší nebo roven než T . Toto kódování je založeno na hypotéze, že pokud je nevýznamný koeficient v hrubé škále, tak bude nevýznamný každý koeficient též prostorové orientace v téže prostorové lokaci jemnější škály. Nevýhodou tohoto kódování je, že je obtížné dekódování pouze části obrazu a také špatné vzpamatovávání se z chyb. Vylepšení je tzv. *Embedded Block Coding with Optimized Truncation* (EBCOT). Tento algoritmus dělí každý sub-band na bloky stejné velikosti (32x32), které pak separátně kóduje. Výhodou tohoto přístupu je možnost paralelního kódování, využití lokálních informací, omezený dopad chyb a možnost náhodného přístupu.

8.3.2 Odstranění šumu a poškození

David Donoho a Iain Johnstone navrhli řešení problému ztráty důležitých vlastností obrazu při redukci šumu. Jejich metoda kombinuje diskrétní waveletovou transformaci s prahovací technikou. Oddělení signálu od šumu umožňuje schopnost waveletové transformace soustředit energii signálu do malého počtu koeficientů s vysokou hodnotou. Naopak energie šumu je rozprostřena do velkého počtu malých koeficientů a proto je možné šum pomocí prahování oddělit od původního signálu. K získání odhadu původního obrazu je nakonec jednoduše použita inverzní transformace.

Předpokládejme signál konečné délky

$$g_i = f_i + n_i, \quad i = 1, \dots, N,$$

kde n_i představuje aditivní gaussovský šum. Potom

$$G_i = F_i + N_i,$$

představuje signál převedený pomocí diskrétní waveletové transformace, která je určena maticí W_i , tzn. platí

$$G_i = W_i g_i.$$

K matici W_i existuje inverzní matice W_i . Schéma odstranění šumu je potom následující.

1. Výpočet diskrétní waveletové transformace $G_i = W_i g_i$.
2. Provedení prahování v transformační doméně a získání odhadu nezašuměného signálu v transformační doméně \hat{F} .
3. Výpočet inverzní matice diskrétní waveletové transformace a získání odhadu nezašuměného $\hat{f} = W_i^{-1} \hat{F}$.

Pro prahování lze použít tzv. měkké (*soft*) prahování definované následovně

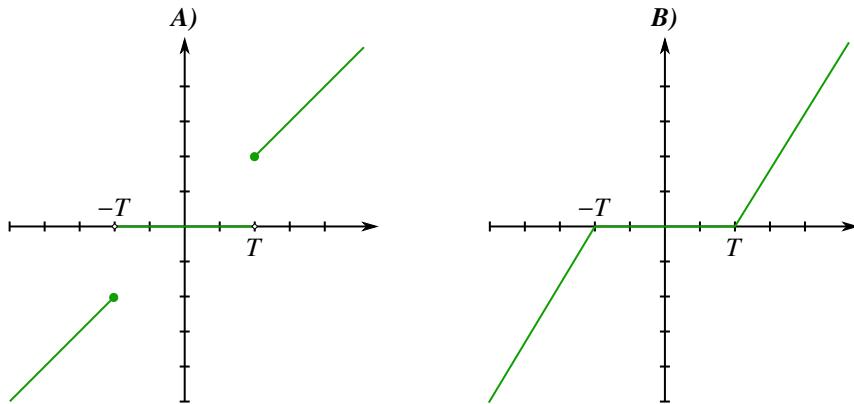
$$\hat{F} = \begin{cases} \text{sgn}(Y) (|Y| - t) & \dots \quad |Y| \geq T, \\ 0 & \dots \quad |Y| < T, \end{cases}$$

a tvrdé (*hard*) prahování definované vztahem

$$\hat{F} = \begin{cases} Y & \dots \quad |Y| \geq T, \\ 0 & \dots \quad |Y| < T. \end{cases}$$

Rozdíl mezi měkkým a tvrdým prahováním je znázorněn na obr. 8.7. Při použití měkkého prahování je odhad s vysokou pravděpodobností minimálně stejně hladký jako původní signál. Výpočetní náročnost metody je $O(n)$.

Tvrde prahování lépe zachovává hrany, naopak měkké prahování dává hladší (a líbivější) výsledky. Mnohdy se na detailní úrovni používá měkké prahování a na ostatní tvrdé.



Obr. 8.7: A) tvrdé prahování, B) měkké prahování.

Velký vliv na výsledek má volba hodnoty prahu. Nejčastěji se používá univerzální práh Donoho-Johnstone, a to jak kvůli rychlosti, tak kvůli automatickému nastavení. Práh T je určen vztahem

$$T = \hat{\sigma} \sqrt{2 \log n},$$

kde n je délka signálu a $\hat{\sigma}$ je směrodatná odchylka. Idea je taková, že odstraníme koeficienty, které jsou menší než očekávané maximum předpokládaného šumu délky n . Často se používá jen pro první odhad prahu. V praxi se používají prahy nezávislé na velikosti signálu.

9

Klasifikace bez učitele (shluková analýza)

Shlukovou analýzu neboli *clustering* používáme, když předem neznáme počet tříd. Nejprve zavedeme, co to je shluk. Po shluku požadujeme, aby byl kompaktní a dobře separovatelný od ostatních shluků. Definovat shluk formálně je složité a zatím neexistuje korektní definice.

- **Shluk:** za shluk budeme považovat množinu C splňující tři vlastnosti.
 1. Neprázdnost, tj. $\exists a, a \in C$.
 2. Mezi každými dvěma body ve shluku existuje cesta, jejíž každý úsek je menší než t , kde $t > 0$ je parametr zadaný uživatelem.
 3. V datech neexistuje žádný bod $b \notin C$, na který by šlo z libovolného bodu shluku C přeskočit skokem menším než t .

Tato definice shluku ovšem není korektní, a to z důvodu, že volba uživatelského parametru je subjektivní. V obecné definici můžeme shluk považovat za libovolnou podmnožinu dat, a tedy proces shlukování lze přirovnat k pokrytí celé množiny disjunktními podmnožinami. Je možné porovnat různá shlukování a na základě kritéria kvality vybrat, které shlukování je nejlepší. Ani tím se ovšem nevyhneme problému subjektivity, protože volba kritéria kvality je také velice subjektivní. Spokojíme se tedy s tím, že dobré shlukování má malé shluky daleko od sebe.

9.1 Wardovo kritérium

Jedna z možností jak určovat míru kvality shluků je Wardovo kritérium ve tvaru

$$J = J(N) = \sum_{i=1}^N \sum_{x \in C_i} \|x - m_i\|^2,$$

kde m_i je těžiště shluku C_i , $\forall i \in \hat{N}$. Tímto kritériem lze srovnávat pouze dvě shlukování se stejným počtem shluků. V L_2 normě se vlastně jedná o součet rozptylů shluků v jednotlivých dimenzích. Dává nám tedy informaci o tom, zda jsou shluky „malé“, ale už nám neřekne nic o tom, jak jsou daleko od sebe. Problémem je, že musíme mít pevně daný počet shluků N . Pokud by tomu tak nebylo, tak minimum bude nastávat pro shluky velikosti 1. Tedy pro libovolné shlukování je funkce $J(N)$ monotónní a pro N rovno počtu prvků, které klasifikujeme, je kritérium vždy rovno 0. Je tedy vidět, že shlukování je vlastně optimalizační úloha. Dalším problémem je, že kritérium předpokládá nekorelované příznaky a konvexní shluky. Tento se však dá vyřešit vhodnou transformací.

9.2 Ostatní metody shlukování

9.2.1 Sekvenční shlukování

Není potřeba znát N . Definujeme uživatelský parametr d a vybereme bod jako zárodek shluku. Vezmeme další bod a pokud je jeho vzdálenost od shluku menší než d , tak ho zařadíme do shluku, pokud je vzdálenost větší než d , tak vytvoříme nový shluk. Je to rychlá metoda, ale strašně závisí na pořadí procházení dat a na volbě uživatelského parametru d . Reálně nepoužitelná metoda.

9.2.2 Shlukování na základě grafů (*graph based clustering*)

Na datech zkonztruujeme graf a nalezneme jeho nejkratší kostru. Odstraníme-li libovolnou hranu, tak se kostra rozpadne na dvě komponenty, protože neobsahuje žádný cyklus. Pokud tedy nalezneme nejdelší hranu a odstraníme jí, dostaneme dva shluky. Stejným postupem můžeme pokračovat. Odstranění nejdelší hrany se ovšem neukazuje jako nejlepší volba. Vylepšením může být například odstranění hrany, která se nejvíce nehodí do okolí, tzn. v nějakém smyslu nekonzistentní hrany. Míru nekonzistence můžeme například volit takto: vezmeme bod a všechny hrany, které do něj vedou → spočteme průměr délky hran a poměr s nejdelší hranou → získáváme určitou míru nekonzistence nejdelší hrany. Tato metoda není zas tak špatná, ale v praxi se moc neprosadila.

9.3 Iterativní metody shlukování

Tyto metody typicky potřebují znalost počtu shluků N .

9.3.1 *N-means clustering*

Předpokládáme, že známe počet shluků N . Postup metody je následující.

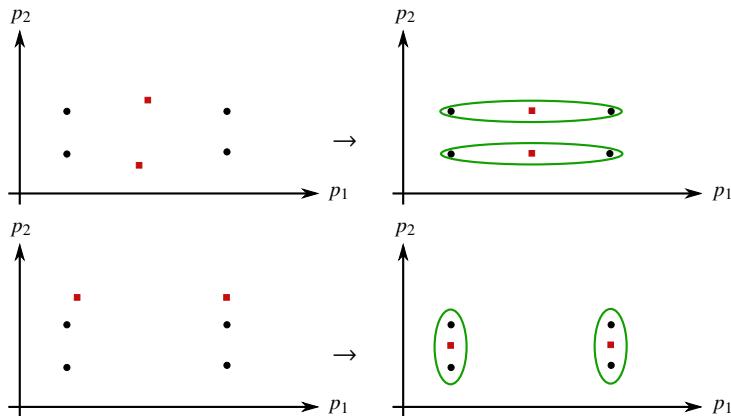
1. Vybereme N těžišť shluků.
2. Klasifikujeme každý bod do těchto N shluků pomocí kritéria minimální vzdálenosti.
3. Aktualizujeme polohy těžišť shluků a pokud se změnily, opakujeme krok 2, jinak končíme.

Jedná se o velmi rychlou metodu a často dává dobré výsledky. Algoritmus neminimalizuje Wardovo kritérium J a je silně závislý na počátečních podmínkách. Kvůli této závislosti může docházet ke špatnému shlukování, viz obr. 9.1. Pokud použijeme algoritmus na data s různými počátečními podmínkami a pokaždé konverguje jinam, tak úloha není vhodná ke shlukování.

9.3.2 Iterační minimalizace Wardova kritéria

Předpokládáme, že známe počet shluků N . Postup metody je následující.

1. Vezmeme počáteční rozdělení shluků, např. z *N-means clustering*.
2. Pro každý bod provedeme minimalizaci Wardova kritéria, tzn. přesouváme ho mezi shluky a hledáme ten, pro který je Wardovo kritérium nejmenší.
3. Opakujeme pro všechny body, dokud se přesouvají.



Obr. 9.1: Ukázka závislosti procesu shlukování při použití N -means na počátečních podmínkách. Červené body značí těžiště shluku.

Algoritmus opět závisí na počátečních podmínkách, ale už ne tolik jako předchozí metoda. Přestože je metoda optimální v každém kroku, nemusí nalézt globální minimum Wardova kritéria. Naleze lokální minimum, které závisí na počátečních podmínkách.

Poznámka 9.1. Isodata: balík procedur na shlukování (komerční nástroj). Základem jsou výše zmíněné metody v kombinaci s hierarchickými metodami.

9.4 Hierarchické metody shlukování

Hierarchické metody se používají, pokud neznáme počet shluků N .

9.4.1 Agglomerativní přístup

Na začátku je každý bod brán jako jeden shluk a na základě vzdálenosti/podobnosti shluků vždy dva nejpodobnější spojíme. Pokračujeme až do nějaké koncové podmínky. Otázkou tedy je jak volit koncovou podmínku a jak porovnávat vzdálenost/podobnost shluků. Nevýhodou této metody je, že mohou vzniknout nejednoznačnosti už na začátku shlukování, které se projeví až později ve velkých shlucích. Předchozí kroky není možné změnit. Agglomerativní metody dělíme podle kritéria, dle kterého se vybírají nejpodobnější shluky.

9.4.1.1 Volba kritéria pro spojení shluků

Chceme definovat metriku (stačí pseudometrika nebo semimetrika) na množinách (shlucích). Nabízí se několik možností:

9.4. Hierarchické metody shlukování

Nejbližší soused:	$d(A, B) = \min_{a,b} \rho(a, b),$
Nejvzdálenější soused:	$d(A, B) = \max_{a,b} \rho(a, b),$
Centroidní metoda:	$d(A, B) = \rho(m_A, m_B),$
Hausdorfova metrika:	$d(A, B) = \max \left(\max_a \rho(a, B), \max_b \rho(b, A) \right),$ $d(A, B) = J(A \cup B) - J(A, B).$

Předposlední uvedená metrika je tzv. Hausdorfova metrika. Její výpočet je ovšem zbytečně náročný, a proto si vystačíme s jednoduššími metrikami. Je jasné, že shlukovací proces na zvolené metrice silně závisí a volba metriky je tedy subjektivní záležitost související s požadavky na výsledné shluky.

9.4.2 Divizivní přístup

Divizivní metody berou množinu bodů, kterou mají zpracovat, jako jeden shluk a ten dále dělí na menší shluky, čímž vytváří hierarchický systém. V každém kroku vybíráme jeden shluk a ten rozdělíme na dva nové. Je tedy nutné vždy vyřešit otázku, který shluk budeme dělit a jakým způsobem ho budeme dělit. V každém kroku tedy řešíme malou úlohu divizivního shlukování. Pokud budeme dělit shluky bez nějaké koncové podmínky, získáme nakonec jednoprvkové shluky. Nevýhodou těchto metod je, že nalezení optimálního rozkladu množiny n objektů na dvě podmnožiny vyžaduje prozkoumání $2^{n-1} - 1$ možností. Tento postup je kvůli exponenciální časové složitosti prakticky proveditelný jen pro malý počet objektů.

9.4.2.1 MacNaughton-Smithova metoda

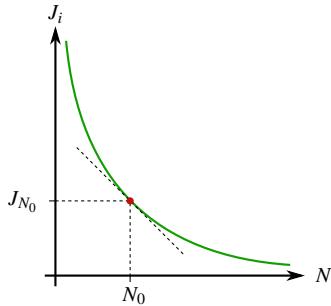
Tato metoda se snaží snížit časovou náročnost divizivních algoritmů až na kvadratickou, ale za cenu toho, že výsledné rozdelení nemusí být optimální. Je tedy aplikovatelná i na rozsáhlejší množiny objektů při nevelkých náročích na výpočetní čas. V porovnání s aglomerativními přístupy je ale stále pomalejší.

Pomocí středních vzdáleností vybereme bod uvnitř shluku a vytvoříme nový shluk. Na základě rozdílů středních vzdáleností bodů z původního shluku a bodů ze shluku nového postupně rozhodujeme kam zařadit body původního shluku. Rozhodujeme tedy, zda bod necháme v původním shluku, nebo ho přesuneme do shluku nového. Výhodou oproti aglomerativnímu přístupu jsou jednoznačnější výsledky pro větší shluky.

9.4.2.2 Volba optimálního počtu shluků

Pokud nemáme počet shluků N pevně dán, je úloha obtížná. Rozdělujeme jeden shluk na malé shluky a počítáme hodnotu kritéria J_i . Pokud spočteme J_i pro $i \in \hat{N}$, můžeme tyto hodnoty vynést do grafu v závislosti na N . Optimální počet shluků je pak N_0 takové, kde derivace vynesené křivky začne růst o poznání pomaleji viz obr. 9.2.

Poznámka 9.2. Existuje ještě tzv. *model based parametric clustering*, což je fitování dat pomocí gaussovské směsi. Problémem je určení počtu komponent směsi.



Obr. 9.2: Ukázka vynášení globálního kritéria pro spočítané shluky a předpokládané optimální shlukování.

9.5 Další kritéria kvality

Kromě Wardova kritéria existují i jiná kritéria, která lze použít na porovnávání kvality shlukování. Definujme si nejprve následující pojmy.

- **Between cluster matrix (BCM):** kovarianční matice z těžišť všech shluků

$$B = \sum_{i=1}^N n_i (m_i - m) (m_i - m)^T.$$

- **Within cluster matrix (WCM):** kovarianční matice shluků W_i a průměrná kovarianční matice

$$W = \sum_{i=1}^N W_i,$$

$$W_i = \sum_{x \in C_i} (x - m_i) (x - m_i)^T.$$

- **Total scatter matrix (TSM):** kovarianční matice všech bodů bez ohledu na shluk

$$T = \sum_x (x - m) (x - m)^T.$$

V předchozích vztazích je n_i počet prvků a m_i těžiště shluku C_i , dále m je celkové těžiště a x prvek shluku. Pro tyto matice platí vztah $T = W + B$. Další kritéria můžeme tvořit následovně

$$\text{tr}(W) = \text{tr}\left(\sum_{i=1}^N W_i\right) = \sum_{i=1}^N \text{tr}(W_i) = \sum_{i=1}^N \sum_{x \in C_i} \|x - m_i\|^2 = J.$$

To tedy znamená, že minimalizace stopy WCM je ekvivalentní minimalizaci Wardova kritéria. Dalšími možnostmi jsou

$$\min \det(W), \quad \max \text{tr}(W^{-1}B).$$

9.6 Aplikace v *image processing*

- Segmentace: špatně definovaná úloha, protože objekt je shluk, shluk je objekt a nikdo neví, co je přesně objekt.
- Klasifikace multispektrálních obrázků.

9.6. Aplikace v *image processing*

- Shlukování v parametrickém prostoru: RANSAC, registrace a matching.

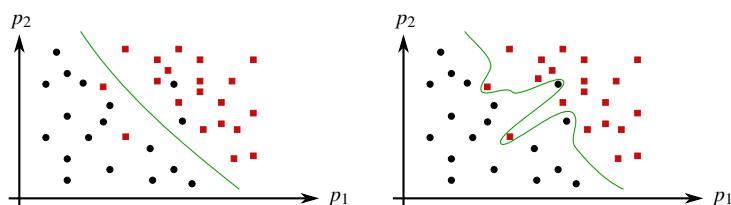
10

Klasifikace s učitelem

Při klasifikaci s učitelem pracujeme s tzv. trénovacími daty, tj. daty se známou klasifikací, na kterých „náladíme“ klasifikátor, a ten pak můžeme použít ke klasifikaci.

- **Trénovací množina:** měla by obsahovat reprezentanty všech tříd a pro každou třídu všechny hlavní typy příznaků. Měla by být spolehlivě klasifikovaná (bez chyb) a dostatečně velká. Vybírat by jí měl někdo, kdo rozumí dané oblasti, odkud data pochází.
- **Klasifikátor:** nechť je každá třída charakterizována svou diskriminační funkcí $g_i(x)$, pak klasifikací rozumíme maximalizaci diskriminační funkce, tzn. pokud platí $g_i(x) > g_j(x), \forall j \neq i$, tak x patří do i -té třídy. Diskriminační funkce definuje rozhodovací hranici v příznakovém prostoru.

Při trénování klasifikátoru může nastat jev, který se nazývá přetrénování dat. K tomuto jevu dochází, pokud se snažíme mít data až příliš dokonale klasifikovaná, viz obr. 10.1. Pokud například v datech bude odlehlé pozorování nebo chyba, a my přesto budeme chtít, aby klasifikátor klasifikoval trénovací množinu správně, tak po natrénování při skutečné klasifikaci nebude fungovat.



Obr. 10.1: Správné klasifikování a přetrénování dat.

10.1 Minimum distance klasifikátory

Jedná se o jednoduché klasifikátory, které klasifikují data podle nejmenší vzdálenosti od tříd.

10.1. Minimum distance klasifikátory

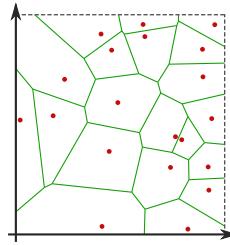
10.1.1 NN klasifikátor (*Nearest neighbor*)

Tento klasifikátor pracuje s diskriminačními funkcemi ve tvaru

$$g_i(x) = \frac{1}{\text{dist}(x, \omega_i)},$$

kde ω_i je nejbližší soused. Obecně vytváří tento klasifikátor lomenou čáru, která odděluje jednotlivé třídy. Výhodou je, že při klasifikaci respektuje jednotlivé body. Naopak nevýhodou je extrémní citlivost na odlehlá pozorování. Vylepšením může být použití vzdálenosti k nejbližšímu těžišti místo k nejbližšímu sousedovi. Tím získáme robustnost na odlehlá pozorování, ale klasifikátor už nebude respektovat tvar ani počet prvků jednotlivých tříd.

Pokud máme třídy obsahující vždy jen jeden bod, tak NN klasifikátor vytvoří tzv. Voronoiovy polygony. Tyto polygony rozdělují rovinu na oblasti $S(x)$, kde x je prvek trénovací množiny, pro které platí, že libovolný prvek této oblasti $y \in S(x)$ je blíže k bodu x než k jakémukoliv jinému prvku trénovací množiny. Pokud uvažujeme euklidovskou metriku, tak dostaneme vždy konvexní polygony. Při volbě jiné metriky můžeme dostat i nekonvexní polygony. Duální úlohou k hledání Voronoiových polygonů je tzv. Delaunayho triangulace.



Obr. 10.2: Voronoiovy polygony. Body trénovací množiny jsou znázorněny červenými tečkami a rozhodovací hranice jsou zelené.

10.1.2 k -NN klasifikátor (*k -Nearest neighbor*)

Jiným vylepšením NN klasifikátoru, které je robustní na odlehlá pozorování, je k -NN klasifikátor. Existují dvě verze tohoto klasifikátoru.

1. Najdu k nejbližších sousedů a podle největšího zastoupení sousedů ve třídách klasifikuj do určité třídy.
Tato metoda nám nedá vždy odpověď, není deterministická, a proto je nutné další rozhodovací kritérium pro případ, kdy obdržíme z více tříd stejný počet sousedů. Výhodou ovšem je, že jsme schopni odhadnout složitost.
2. Hledáme nejbližší sousedy, dokud jich z jedné třídy není k , a do této třídy poté klasifikujeme. Jedná se o deterministickou úlohu, ale složitost není možné odhadnout. V praxi se používá více než první metoda.

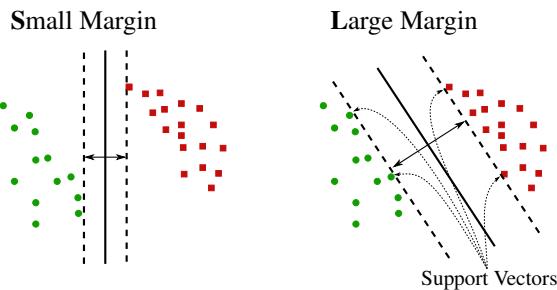
Otázkou je jak správně volit hodnotu k . Čím menší k zvolíme, tím je metoda náhylnější na přetrénování. Obecně lze říci, že by měla být řádově menší než počet prvků trénovací množiny. Většinou se volí $k \in \langle 2, 5 \rangle$. Metoda se hodí pro menší skupiny dat.

10.2 Lineární klasifikátory

Chceme nalézt nadrovinu v příznakovém prostoru, která bude oddělovat jednotlivé třídy. Úlohu klasifikace tedy převádíme na jednodušší úlohu hledání hranice. Klasifikace ovšem nemusí být správná. Jak bychom tedy hranici hledali? Začneme osou mezi dvěma body z různých tříd a postupně přidáváme další body. Pokud „padají“ na správnou stranu přímky, tak s přímkou nic neděláme. V opačném případě přímku posouváme a nakláníme, aby byly body správně separované. Lepší přístup je upravovat přímku vždy, i když padají nové body na správnou stranu, a to např. minimalizací rozdílů středních vzdáleností od přímky.

10.2.1 Support vector machine (SVM)

Support vector machine je metoda strojového učení, která hledá nadrovinu takovou, že v prostoru příznaků optimálně rozděluje trénovací data. Optimální nadrovinou je taková, že body leží v opačných poloprostorech a hodnota minima vzdáleností bodů od roviny je co největší. Jinými slovy, okolo nadroviny je na obě strany co nejširší pruh bez bodů, tzn. maximální odstup (*maximal margin*). Metoda se tedy snaží konstruovat dvě rovnoběžné nadroviny tak, aby separovaly třídy a byly co nejdále od sebe. Na popis těchto dvou nadrovin stačí pouze body, které nadroviny protínají. Tyto body se nazývají podpůrné vektory (*support vectors*) a odtud název metody. Vlastní rozhodovací nadrovinu je s nalezenými nadrovinami rovnoběžná a vede mezi nimi. SVM je ze své přirozenosti binární, tedy rozděluje data do dvou tříd a rozdělující nadrovinu je lineární funkcí v prostoru příznaků.

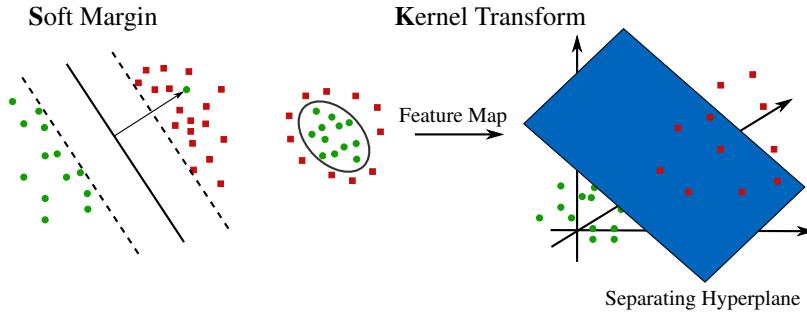


Obr. 10.3: Princip SVM: porovnání neoptimálního a optimálního lineárního klasifikátoru.

Nevýhodou této metody je, že podpůrné vektory jsou většinou extremální body. Metoda je tedy velice citlivá na odlehálo pozorování a šum. Jistým vylepšením je zavedení ztrátové funkce, kterou budeme minimalizovat. Do ztrátové funkce budou přispívat body, které jsou špatně klasifikované. Už tedy nebude vyžadovat, aby rozdělovací přímka dělila všechny body, tzv. *hard margin* přístup, ale umožníme neúplné rozdělení, tzv. *soft margin* přístup, viz obr. 10.4. Body, které jsou špatně klasifikované, poté penalizujeme a celkově se snažíme minimalizovat zvolenou ztrátovou funkci. Dále také tato metoda nezohledňuje počty bodů v jednotlivých množinách a jejich tvar. Rozhodovací přímku prostě posune v poměru k té množině, kde je více prvků.

Často nemusí existovat dvě rozdělující přímky, pokud nejsou třídy lineárně separovatelné. Důležitou součástí techniky SVM potom je tzv. jádrová transformace (*kernel transformation*). Jedná se o transformaci prostoru příznaků dat do prostoru transformovaných příznaků (typicky vyšší dimenze). Tato jádrová transformace umožňuje převést původně lineárně neseparovatelnou úlohu na úlohu lineárně separovatelnou v prostoru vyšší dimenze, na kterou lze dále aplikovat optimalizační algoritmus pro nalezení rozdělující nadroviny. Pokud takovou nadrovinu získáme, tak zpětnou transformací do původního prostoru příznaků získáme nelineární rozhodovací nadplochu. Princip jádrové transformace je uveden na obr. 10.4.

10.3. Bayesův klasifikátor



Obr. 10.4: Princip SVM: soft margin přístup a jádrová transformace.

10.3 Bayesův klasifikátor

Idea Baesova klasifikátoru je, že hodnoty příznaků jsou náhodné veličiny a příslušnost k třídám modelujeme pomocí hustoty pravděpodobnosti. Jedná se tedy o statistický klasifikátor, který je založen na Bayesově principu

$$P(\omega_i | x) = \frac{p(x|\omega_i) P(\omega_i)}{p(x)}, \quad (10.1)$$

kde

- **podmíněná pravděpodobnost výskytu ve třídách (class-conditional probability):** $p(x|\omega_i)$ vyjadřuje pravděpodobnost, že na prvku z i -té třídy můžeme naměřit příznakový vektor x ;
- **apriorní pravděpodobnost i -té třídy (a prior probability):** $P(\omega_i)$ vyjadřuje pravděpodobnost výskytu i -té třídy v Ω , tzn. v „reálu“ ne jen v trénovací množině;
- **aposteriorní pravděpodobnost (a posterior probability):** $P(\omega_i | x)$ vyjadřuje pravděpodobnost, že se v i -té třídě může vyskytnout prvek s příznakovým vektorem x ;
- **celková pravděpodobnost (total probability):** $p(x) = \sum_{j=1}^c p(x|\omega_j) P(\omega_j)$ vyjadřuje pravděpodobnost, že naměříme příznakový vektor x .

Cílem je maximalizovat aposteriorní hustotu pravděpodobnosti a Bayesův vzorec používáme proto, že nejsme schopni aposteriorní pravděpodobnost spočítat přímo. Ze vztahu (10.1) ovšem plyne, že maximalizace aposteriorní pravděpodobnosti je ekvivalentní maximalizaci součinu $p(x|\omega_i) P(\omega_i)$ a v případě shodných apriorních pravděpodobností pouze maximalizaci pravděpodobnosti $p(x|\omega_i)$. V řeči diskriminačních funkcí vypadá zápis následovně

$$g_i(x) = P(\omega_i | x) \iff g_i(x) = p(x|\omega_i) P(\omega_i) \iff g_i(x) = \ln p(x|\omega_i) + \ln P(\omega_i). \quad (10.2)$$

Dále se budeme zabývat tím, jak tyto pravděpodobnosti odhadovat.

10.3.1 Odhad apriorní pravděpodobnosti

Apriorní pravděpodobnosti $P(\omega_i)$ určíme buď z relativních četností trénovací množiny, v tom případě je nutné zajistit, aby četnosti odpovídaly skutečnosti. Další možností je předpokládat pro všechny třídy stejnou apriorní pravděpodobnost, tj. $P(\omega_i) = \frac{1}{c}$, $\forall i \in \hat{c}$, kde c je počet tříd.

10.3.2 Parametrické odhady class-conditional pravděpodobnosti

Parametrické odhady můžeme použít pouze v případě, že známe hustotu pravděpodobnosti. Předpokládejme tedy, že class-conditional pravděpodobnost má normální (gaussovskou) hustotu pravděpodobnosti, a budeme odhadovat parametry Gaussova rozdělení.

10.3.2.1 1-dimenzionální příznakový prostor

Pro jednorozměrný příznakový vektor máme hustotu pravděpodobnosti pro příznak ve tvaru

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right].$$

- **Odhad pomocí výběrových statistik:** pro nejjednodušší odhad můžeme použít výběrový průměr a rozptyl

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n x_k, \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \hat{\mu})^2.$$

- **Metoda maximální věrohodnosti (maximal likelihood estimator):** předpokládejme, že máme soubor iid náhodných veličin x_1, x_2, \dots, x_N a jejich hustotu pravděpodobnosti uvažujeme jako funkci parametrů $\vec{\theta} = (\theta_1, \theta_2 \dots \theta_k)$, tzn. $p(x|\vec{\theta})$. V našem případě se jedná o prvky z trénovací množiny. Zavedeme věrohodnostní funkci

$$L(x_1, x_2, \dots, x_N | \vec{\theta}) = \prod_{i=1}^N p(x_i | \vec{\theta}) \iff \Lambda = \ln L(x_1, x_2, \dots, x_N | \vec{\theta}) = \sum_{i=1}^N \ln p(x_i | \vec{\theta}).$$

Maximálně věrohodný odhad parametrů dostáváme řešením maximalizační úlohy

$$\hat{\vec{\theta}}_{MLE} = \arg \max_{\vec{\theta} \in \Theta} L(x_1, x_2, \dots, x_N | \vec{\theta}),$$

což je ekvivalentní hledání extrému funkce Λ . Spočteme tedy její parciální derivace podle parametrů $\theta_1, \theta_2 \dots \theta_k$ a položíme je rovny 0. Řešením této soustavy rovnic získáme požadovaný odhad parametrů.

10.3.2.2 d -dimenzionální příznakový prostor

V d -dimenzionálním případě máme hustotu pravděpodobnosti pro příznakový vektor ve tvaru

$$p_d(\vec{x}) = \frac{1}{(2\pi)^{d/2} \sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right],$$

kde $\Sigma_{i,j} = \Sigma_{j,i} = \text{Cov}(x_i, x_j)$, tzn. jedná se o kovarianci i -tého a j -tého příznaku. Matici Σ nazýváme kovarianční maticí. Každá třída má tedy vlastní Gaussovou hustotu pravděpodobnosti s vlastní kovarianční maticí.

- **Odhad pomocí výběrových statistik:** pro nejjednodušší odhad můžeme použít

$$\hat{\vec{\mu}} = \frac{1}{n} \sum_{k=1}^n \vec{x}_k, \quad \hat{\Sigma} = \frac{1}{n} \sum_{k=1}^n (\vec{x}_k - \hat{\vec{\mu}})(\vec{x}_k - \hat{\vec{\mu}})^T.$$

Poznámka 10.1. Pro připomenutí: mějme náhodné veličiny x a y . Kovarianci a korelace definujeme následovně

$$\text{Cov}(x, y) = \mathbb{E}[x - \mathbb{E}[x]] \mathbb{E}[y - \mathbb{E}[y]], \quad \text{Cor}(x, y) = \frac{\text{Cov}(x, y)}{\sqrt{\text{Var}(x) \text{Var}(y)}}.$$

10.3. Bayesův klasifikátor

Je důležité si uvědomit, co vlastně udává kovarianční matice Σ . Z její konstrukce víme, že matice vypadá takto

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \text{Cov}(x_1, x_2) & \cdots & \text{Cov}(x_1, x_d) \\ \text{Cov}(x_1, x_2) & \sigma_2^2 & \cdots & \text{Cov}(x_2, x_d) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(x_1, x_d) & \text{Cov}(x_2, x_d) & \cdots & \sigma_d^2 \end{pmatrix},$$

tzn. na diagonále máme rozptyly jednotlivých příznaků a mimo diagonálu jejich vzájemné kovariance. Je také patrné, že je kovarianční matice symetrická. Vlastní vektory kovarianční matice nám udávají vychýlení vrstevnic (v contour plotu) a vlastní čísla udávají rozptyly v souřadné soustavě dané vlastními vektory. Rozdělovací hranice mezi jednotlivými třídami jsou v d -dimenzionálním příznakovém prostoru kuželosečky.

Nyní uvažujme pouze 2-dimenzionální příznakový prostor. Kovarianční matici máme tedy ve tvaru

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \text{Cov}(x_1, x_2) \\ \text{Cov}(x_1, x_2) & \sigma_2^2 \end{pmatrix}.$$

Poměr vlastních čísel nám udává, jak moc je elipsa reprezentující gaussovskou hustotu (v contour plotu) „sešláplá“. Pokud dosadíme tvar hustoty pravděpodobnosti příznakového vektoru do diskriminační funkce (10.2) dostáváme pro i -tou třídu

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}) - \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i). \quad (10.3)$$

Klasifikace je pro nás tedy v tomto případě ekvivalentní porovnávání dvou gaussovských rozdělení. V případě, že máme dvě gaussovská rozdělení, která mají stejné kovarianční matice a jsou od sebe pouze posunutá, dostáváme jako rozhodovací hranici přímku. Toto plyně z řešení rovnosti

$$g_1(\vec{x}) = g_2(\vec{x}) \iff g_1(\vec{x}) - g_2(\vec{x}) = 0.$$

Protože mám stejné kovarianční matice a platí vztah (10.3), dostáváme

$$\begin{aligned} 0 &= -\frac{1}{2}(\vec{x} - \vec{\mu}_1)^T \Sigma^{-1} (\vec{x} - \vec{\mu}_1) + \ln P(\omega_1) + \frac{1}{2}(\vec{x} - \vec{\mu}_2)^T \Sigma^{-1} (\vec{x} - \vec{\mu}_2) - \ln P(\omega_2) \\ \iff 0 &= 2\vec{\mu}_1^T \Sigma^{-1} \vec{x} - \vec{\mu}_1^T \Sigma^{-1} \vec{\mu}_1 + 2 \ln P(\omega_1) + 2\vec{\mu}_2^T \Sigma^{-1} \vec{x} + \vec{\mu}_2^T \Sigma^{-1} \vec{\mu}_2 - 2 \ln P(\omega_2) \\ \iff 0 &= 2(\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma^{-1} \vec{x} - \vec{\mu}_1^T \Sigma^{-1} \vec{\mu}_1 + 2 \ln P(\omega_1) + \vec{\mu}_2^T \Sigma^{-1} \vec{\mu}_2 - 2 \ln P(\omega_2) \\ \stackrel{\text{ozn.}}{\iff} 0 &= \vec{A}\vec{x} + b \end{aligned}$$

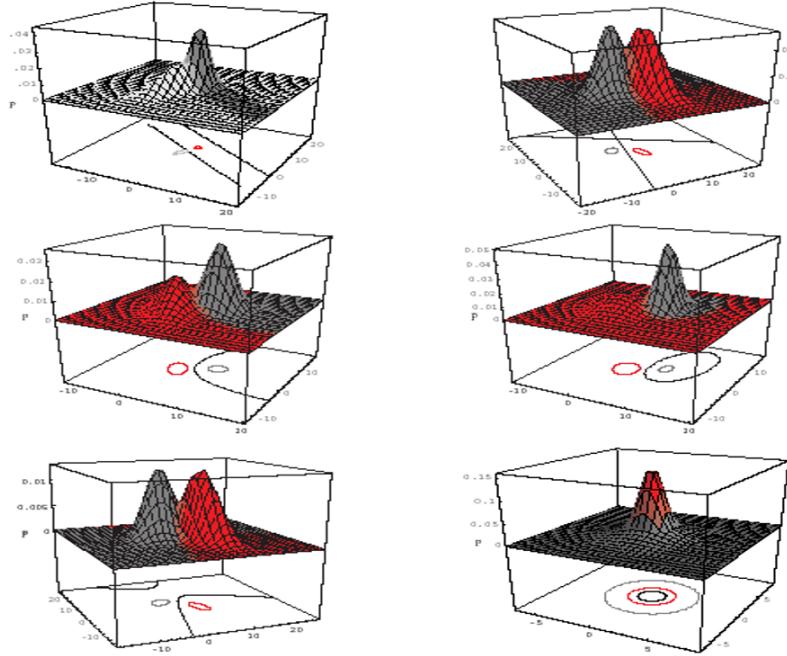
V takovém případě je klasifikátor lineární. Platí i opačná implikace, tedy pokud je klasifikátor lineární, tak jsou kovarianční matice stejné. Této vlastnosti se dá využít pro lepší odhad kovarianční matice, protože na její odhad můžeme použít všechna data a odhad je pak přesnější. To, zda jsou kovarianční matice stejné, testujeme statisticky. Další příklady jak vypadají rozdělovací hranice pro různé tvary a vzájemné polohy gaussovských rozdělení jsou uvedeny na obr. 10.5.

V případě, že uvažujeme stejnou apriorní pravděpodobnost pro všechny třídy, můžeme zjednodušit diskriminační funkci do tvaru

$$g_i(\vec{x}) = -\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}).$$

Definujme Mahalanobisovu vzdálenost vztahem

$$d_M(\vec{x}) = (\vec{x} - \vec{\mu})^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}),$$



Obr. 10.5: Různé vzájemné polohy gaussovských rozdělení a rozdělovací hranice (rovnoběžky, různoběžky, přímka,...).

který udává vzdálenost bodu od třídy. Z tvaru Mahalanobisovy vzdálenosti plyne, že její minimalizace je ekvivalentní maximalizaci diskriminační funkce. Tedy klasifikaci můžeme provádět minimalizací Mahalanobisovy vzdálenosti.

Poznámka 10.2. Protože všude předpokládáme normální rozložení dat, měli bychom tuto normalitu otestovat. To není problém pro jedno-dimenzionální data, ale pro více-dimenzionální už to problém je. V praxi se toto šídí tím, že otestujeme normalitu všech marginálních rozdělení, a poté prohlásíme, že i sdružené rozdělení je normální. Tato implikace ovšem platí pouze opačným směrem. Normalita marginálních rozdělení nám nic neříká o normalitě sdruženého rozdělení. Není to tedy korektní postup, ale v praxi se kvůli časové úspore používá. Je tedy možné to použít, ale musíme vědět, co děláme špatně a proč to děláme špatně.

10.3.2.3 Co dělat pokud třídy nejsou normálně rozdělené

V případě, že máme data, která nejsou normálně rozdělená, nabízí se nám několik možností jak postupovat.

- První možností je odhadování parametrů jiného než gaussovského rozdělení, které by nám data popisovalo.
- Další možností je popsat data pomocí gaussovských směsí. Většinou se používají součtové směsi. Může dobře fungovat, pokud podtřídy mají normální rozdělení. K tomuto jevu může například dojít, když vezmeme moc velké třídy. Pokud třeba vezmeme dohromady váhy dětí, dospělých a slonů, tak nejspíš normální rozdělení mít nebudou, ale jednotlivě ano.
- Poslední možností jsou neparametrické odhadování, kterým se budeme věnovat v další části.

10.3.3 Neparametrické odhadы class-conditional pravděpodobnosti

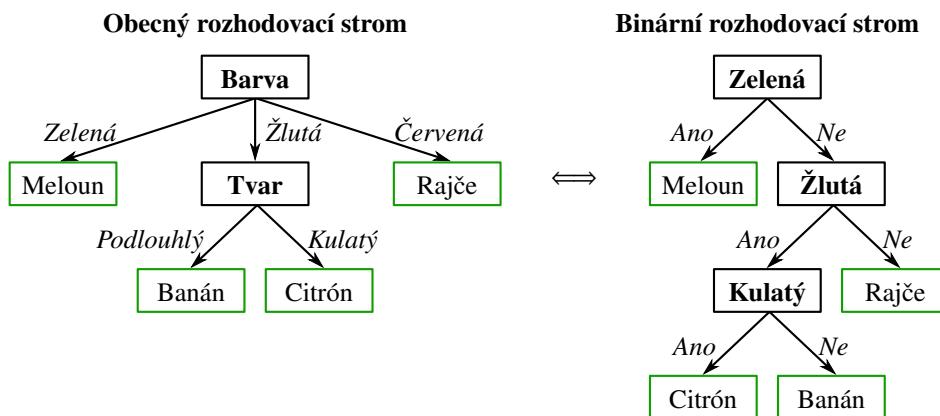
Hustotu pravděpodobnosti na intervalech odhadneme pomocí relativních četností jejích realizací. Provádíme průměrování relativních četností pomocí váženého výběrového okna w , např. *Parzen window*. Dost záleží na tom, jakou zvolíme šířku okna, tzn. nosíč funkce w . Extrémem jsou volby $w = 1$, kdy dostáváme jednotlivé realizace s pravděpodobností $\frac{1}{n}$ a $w = \max$, kdy dostaneme konstantní hustotu po celém intervalu. Malá okna vedou na přetrénování a velká naopak na vyhlazení dat. V případě spojitéh dat na velikost okna nezáleží.

10.3.4 Aplikace Bayesova klasifikátoru v multispektrálním snímání země

Bayesů klasifikátor se velice často používá v multispektrálním snímání země. V této úloze jako objekty bereme jednotlivé pixely a příznaky pro nás jsou hodnoty intenzity pixelů ve spektrálním pásmu (od 4 o několika set). Jedná se například o klasifikování družicových snímků, kde příznaky mohou být třeba voda, poušť, les, resp. pásmo intenzit, která jím odpovídají. Trénovací množiny se vybírají ručně pomocí tematických map. Typický počet tříd je 2 až 16.

10.4 Rozhodovací stromy

Jedná se o nemetrické klasifikátory. Obecné rozhodovací stromy mohou mít více větví v každém uzlu. Existuje ovšem způsob, jak libovolný rozhodovací strom převést na strom binární, který má v každém uzlu dvě větve, viz obr. 10.6. Typicky tedy rozhodovací stromy pracují s příznaky „Yes - No“. V praxi se moc nepoužívají, v podstatě jen na jednoduché úlohy.



Obr. 10.6: Převedení obecného rozhodovacího stromu na binární rozhodovací strom.

10.5 Účinnost klasifikace (*classification performance*)

Nyní se zabýveme otázkou jak testovat kvalitu klasifikátoru. Chceme zjistit jak moc úspěšný klasifikátor byl, tzn. kolik dat klasifikoval správně. Toto testování je nutné provádět na množině dat.

- **Test na trénovací množině:** optimistický odhad chyby, ale je to blbost. Na této množině jsme klasifikátor trénovali, takže by měl fungovat dobře.

- **Test na testovací množině:** pesimistický odhad chyby. U testovací množiny musíme stejně jako u množiny trénovací vědět, jak mají být data správně klasifikována.

Pokud budeme intuitivně pouze porovnávat „počet zásahů“, dostaneme sice jakousi představu o kvalitě klasifikátoru, ale je to zavádějí. Různé chyby totiž mohou mít různou důležitost. Typickým příkladem je určování nemoci u pacientů. Pokud totiž řekneme, že pacient je nemocný, a začneme ho léčit, i když ve skutečnosti není, tak se dopustíme méně závažné chyby, než když nemocného pacienta prohlásíme za zdravého a léčit ho nebude.

Jestliže máme všechny třídy stejně zastoupeny a všechny chyby při klasifikaci nám vadí stejně, pak je možné kvalitu klasifikátoru charakterizovat jedním číslem. Jinou možností pro vyjádření kvality jsou tzv. *confusion table*. Uvažujme příklad, kdy mám 27 zvířat (8 koček, 6 psů a 13 králíků) a confusion table v následujícím tvaru.

		Klasifikace		
		Kočky	Psi	Králíci
Skutečné třídy	Kočky	5	3	0
	Psi	2	3	1
	Králíci	0	2	11

Tato tabulka udává, že ze skutečného počtu 8 koček klasifikátor 3 klasifikoval jako psy atd. V případě, že jsou všechny mimo diagonální prvky přibližně stejné, můžeme tuto tabulku zkomprimovat do jednoho čísla, které značí úspěšnost.

Důvodem, proč klasifikátor nefunguje přesně, může být například špatná trénovací množina. Pokud jsou již v trénovací množině data špatně klasifikovaná, není možné klasifikátor natrénovat správně. Tedy vybrat dostatečně kvalitní trénovací množinu je velice důležité.

10.6 Kombinované klasifikátory

Pokud chceme zlepšit úspěšnost klasifikace, máme několik možností jak toho dosáhnout. Můžeme zkoušit změnit příznaky, nebo příznaky přidat. Ve druhém případě ovšem můžeme narazit na problém vysoké dimenze (tzv. *curse of dimensionality*). Další možností je zvolit větší a kvalitnější trénovací množinu, nebo použít jiný klasifikátor, případně jiný parametrický model.

Pokud ani jedna z předchozích možností nepomáhá nebo je není možné provést, lze použít kombinaci více nezávislých klasifikátorů $\mathcal{K}_1, \dots, \mathcal{K}_L$. Protože některé fungují jen na určité příznaky. Otázkou je, jak zkombinovat výsledky od jednotlivých klasifikátorů. Nechť k -tý klasifikátor určí $p_k(x|\omega_i)$. Jako kombinovaný klasifikátor můžeme například brát většinovou volbu. To je ale takové „nestatistiké“ řešení. Protože pokud je nějaký klasifikátor úplně špatný, z této volby to nepoznáme. Když už kombinujeme více klasifikátorů, tak chceme, aby i ten nejhorší z nich dosahoval co nejlepších výsledků. Lepší volby kombinací jsou:

$$p(x|\omega_i) = \max_i \prod_{k=1}^L p_k(x|\omega_i), \quad p(x|\omega_i) = \max_i \min_{k \in \hat{L}} p_k(x|\omega_i), \quad p(x|\omega_i) = \max_i \max_{k \in \hat{L}} p_k(x|\omega_i).$$

Ovšem ani pokud zvolíme jednu z předchozích možností, nemáme zaručeno, že kombinovaný klasifikátor bude lepší než nejlepší klasifikátor.

11

Redukce dimenziality příznakového prostoru

Důvodem snahy o redukci dimenziality je, že často vede ke zlepšení klasifikace a zrychlení výpočtu. Nevýhodou je možná ztráta informace, a proto je nutné najít způsob, jak při transformaci zahodit co nejméně informace. Máme D -dimenzionální příznakový prostor a chceme jeho dimenzi transformací zredukovat na n -dimenzionální, kde $n \ll D$, tj.

$$(x_1, x_2, \dots, x_D) \xrightarrow{\mathcal{T}} (y_1, y_2, \dots, y_n).$$

Rozlišujeme dva základní principy.

1. **Feature extraction:** vytváříme nový příznakový prostor a příznaky ztrácejí jejich původní význam. Máme transformaci $\mathcal{T} : \mathbb{R}^D \rightarrow \mathbb{R}^n$. Pro $n = 1$ získáme například nový prostor takto

$$y_1 = \sum_{k=1}^D x_k.$$

2. **Feature selection:** vybíráme pouze podmnožinu příznaků, kterou budeme používat. Výhodou je, že příznaky neztrácejí význam a ty, co nechceme použít, nemusíme ani měřit, což vede k urychlení klasifikace.

Podle toho jaká máme data, rozlišujeme dva základní problémy redukce dimenziality.

11.1 Principle Component Transform (PCT)

Uvažujme, že nedisponujeme trénovací množinou, tzn. máme všechna data a nemáme je roztríďená. Tato data mohou být silně korelovaná a korelace se chceme zbavit. Pokud totiž máme korelované příznaky, tak nám stačí uvažovat jeden a ostatní z něho poté můžeme dopočítat. Na tento problém se používá přístup feature extraction. Uvažujme 2D příznakový prostor. Hlavní myšlenka vychází z toho, že korelované příznaky vytváří v příznakovém prostoru více či méně protáhlou elipsu, v níž můžeme kratší poloosu považovat za šum. Za příznaky můžeme brát souřadnice průmětů bodů z příznakového prostoru do hlavní poloosy této elipsy. Princip metody je následující.

1. Provádíme OG transformaci příznakového prostoru, konkrétně rotaci $y = \mathcal{T}x$. Příznakový prostor rotujeme tak dlouho, dokud nejsou nové příznaky y nekorelované, tzn. kovarianční matici Σ_y je diagonální.

11.2. Multi-class problem

Spočítáme vlastní čísla a vlastní vektory kovarianční matice Σ_x . Vlastní vektory příslušné k různým vlastním číslům jsou ortogonální a vlastní vektory příslušné jednomu vlastnímu číslu lze ortogonalizovat. Rotaci příznakového prostoru můžeme „seskládat“ z vlastních vektorů a dostáváme vztah

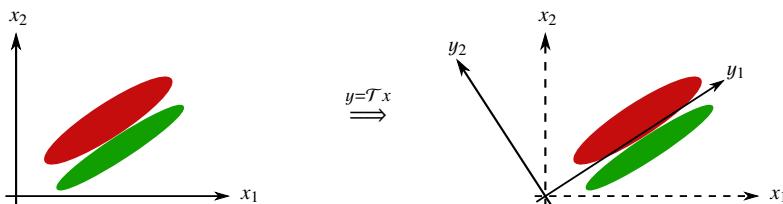
$$\mathcal{T}^T \Sigma_x \mathcal{T} = \Sigma_y \implies y = \mathcal{T}x.$$

V předchozím vztahu je místo inverze rotační matice její transpozice, ale protože je tato matice složená z vlastních vektorů, tak jsou tyto pojmy ekvivalentní. Nově vzniklá kovarianční matice Σ_y má na diagonále rozptyly v nové souřadné soustavě. Důležité je zmínit, že tato transformace je vždy možná a navíc víme, jak sestrojit transformační matici \mathcal{T} . Existence reálných vlastních čísel plyne ze symetrie, reálnosti a pozitivní definitnosti matice Σ_x .

2. Seřadíme příznaky podle rozptylů a vybereme si prvních n příznaků s největšími rozptyly, které se nazývají hlavní komponenty (*principal components*). Tento výběr je možné udělat až po transformaci, protože u korelovaných dat nelze seřadit podle rozptylů. Toto jsou příznaky, které použijeme ke klasifikaci a ostatní „zahodíme“, címž zredukujeme dimenzi úlohy.

Pokud by vlastní čísla byla stejná, nepředstavují data v příznakovém prostoru elipsu, ale kružnice a rotací bychom nic nezměnili. K tomu ovšem může dojít pouze pokud $\Sigma_x = k \cdot \mathbb{I}$, a to znamená, že už původní příznaky jsou nekorelované.

Tato metoda ovšem není moc dobrá pro klasifikaci, protože se u ní vyskytuje problém se separabilitou. PCT totiž hodnotí vliv jednotlivých příznaků pouze na základě jejich rozptylů, ale ty se nemusejí shodovat s jejich diskriminační silou, viz 11.1. Lze jí tedy použít, ale je nutné si být vědom této špatné vlastnosti.



Obr. 11.1: Ilustrace proč je PCT špatná pro klasifikační úlohy.

11.2 Multi-class problem

Předpokládejme, že máme trénovací množinu, tzn. máme data roztríděná do jednotlivých tříd. Lze tedy mluvit o separabilitě příznaků a můžeme vybírat příznaky, které chceme používat. Na tento problém se používá přístup feature selection. Metody pro redukci dimenzionality pro klasifikační úlohy musejí brát v úvahu diskriminační sílu (separabilitu) jednotlivých příznaků. Cílem je maximalizovat diskriminabilitu mezi třídami, otázkou ale je, jak ji měřit. V případě, že bychom chtěli použít Wardovo kritérium, tak nebude fungovat. U shlukování funguje, protože pracujeme na stejném příznakovém prostoru, tzn. máme zafixované příznaky a podle nich rozdělujeme data. Zde je to naopak, máme fixované třídy (trénovací množiny) a vybíráme příznaky.

11.2.1 Two-class problem

Pro počet tříd $N = 2$ a pokud jsou nekorelované a normální, se používá kritérium

$$\frac{(m_1 - m_2)^2}{(\sigma_1^2 + \sigma_2^2)},$$

které celkem dobře funguje. Další používané kritérium pro měření diskriminability je

$$\text{tr}(W^{-1}B),$$

kde B je between cluster matrix a W je within cluster matrix, viz sekce 9.5. Pokud je počet tříd $N = 2$ a obě trénovací množiny mají stejný počet prvků, pak platí

$$\max \text{tr}(W^{-1}B) \sim \max d_M = \max (\vec{m}_1 - \vec{m}_2)(\Sigma_1 + \Sigma_2)^{-1}(\vec{m}_1 - \vec{m}_2)^T,$$

kde d_M je Mahalanobisova vzdálenost. Tato vzdálenost ovšem neumí dobře rozlišit \vec{m}_1 a \vec{m}_2 , která jsou blízko u sebe. Vylepšením, které více zohledňuje tvar jednotlivých tříd je Bhattacharyaova vzdálenost

$$d_B = \frac{1}{4}d_M + \ln \frac{|\Sigma_1 + \Sigma_2|}{2\sqrt{|\Sigma_1||\Sigma_2|}}.$$

Obě tyto vzdálenosti bývají používány po dvojicích i v případě multi-class problému. Je důležité zdůraznit, že zmíněné způsoby měření diskriminability předpokládají normálně rozdělené třídy. Normalitu je tedy třeba testovat, např. pomocí Pearsonova testu a pro jiné třídy není možné metody použít. Lze ještě užít jako approximaci pro některé unimodální třídy. Dále budeme hledat algoritmus, který povede na maximalizaci zvolené míry diskriminability.

11.2.2 Optimální metody

Jedná se o metody používající *fullsearch* příznaků nebo jeho modifikace. Jejich výhodou je, že zaručují nalezení globálního optima. Naopak nevýhodou je velká náročnost a proto se v praxi téměř nepoužívají.

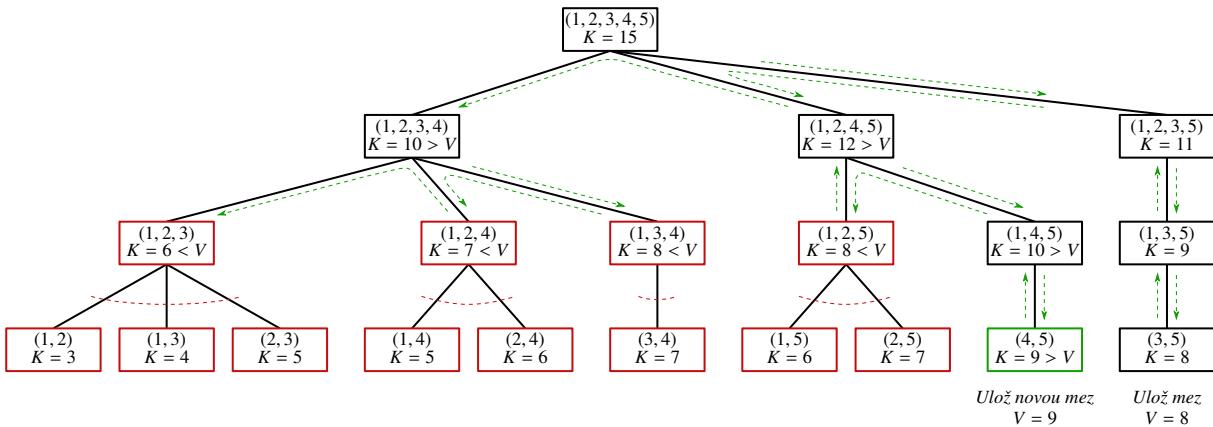
11.2.2.1 Algoritmus *Branch & Bound*

Myšlenkou algoritmu je efektivní procházení celého stromu příznaků, viz obr. 11.2. Strom se odvíjí od kořene, kde je celá D -tice příznaků, k listům, ve kterých už jsou jen k -tice příznaků, $k < D$. V každé následující úrovni stromu ubude jeden příznak. Strom procházíme od kořene k listům a v každém uzlu spočteme kritérium $K = d_M((k-tice))$, které určuje kvalitu popisu pomocí daných příznaků. Protože v každé úrovni ubíráme jeden příznak, bude se hodnota kritéria K se zanořováním hlouběji do stromu zmenšovat. Při procházení stromu tedy nejdříve projdeme jednu větev až do požadované úrovně (tuto úroveň určíme podle toho, kolik chceme použít příznaků pro popis). V této úrovni si uložíme hodnotu kritéria $V := K$ a jdeme k dalším uzlům, ve kterých 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 snažíme K maximalizovat. Pokud se v průběhu procházení opět dostaneme až do požadované úrovně a platí $K > V$, aktualizujeme hodnotu V a zapamatujeme si tento list jako aktuální maximum. Tak postupujeme dále až projdeme celý strom.

11.2.3 Suboptimální metody

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

11.2. Multi-class problem



Obr. 11.2: Algoritmus Branch & Bound při výběru dvou z pěti příznaků. Použité kritérium je ve tvaru $K(p_1, p_2, \dots, p_k) = \sum p_i$.

11.2.3.1 Algoritmus Sequential forward selection (SFS)

Tento algoritmus v prvním kroku najde nejlepší příznak a k němu hledá další další příznak, který s ním vytváří nejlepší dvojici. Tento postup poté opakuje. V k -tému kroku přidává algoritmus příznak, který s předchozí $(k - 1)$ -ticí vytváří nejlepší k -tici vzhledem k danému kritériu kvality kritériu.

V základním provedení má algoritmus vady, protože nelze vyhodit příznak a přidáváme pouze po jednom příznaku. Vylepšení může spočívat v tom, že přidáváme j -tice, kde v rámci j -tice probíhá *fullsearch*. Algoritmus lze samozřejmě modifikovat místo procesu *forward* na odebírání tj. *backward*. Dalším rozšířením tohoto algoritmu může být tzv. plus l minus s algoritmus. Od originálního SFS algoritmu se liší tím, že v daném kroku lze přidat l -tici a odebrat s -tici příznaků. V rozšířené verzi navíc l a s jsou dynamické.