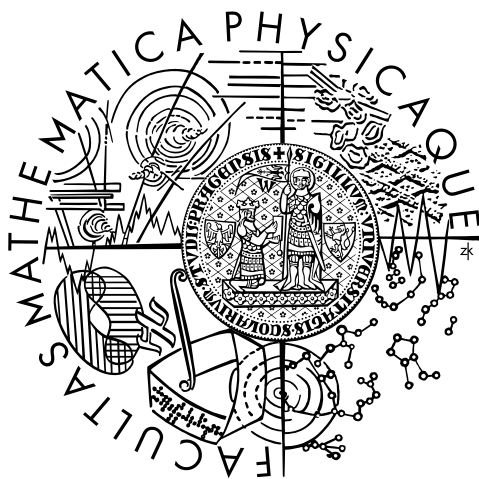


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jiří Šilhán

Podobnost bitmapových obrázků

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Mareš
Studijní program: Informatika, programování

2008

Rád bych na tomto místě poděkoval všem lidem, kteří mne při tvorbě této práce podporovali, nabídli radu, či pomocnou ruku. Obzvláště děkuji svému vedoucímu, Mgr. Martinu Marešovi, za rychlou zpětnou vazbu a spousty podnětných připomínek.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 26. července 2008

Jiří Šilhán

Obsah

1	Úvod	6
2	Popis systému k hledání obrázků dle obsahu (CBIR)	8
3	Rozbor příznaků nižší úrovně	10
3.1	Barva	10
3.1.1	Globální popis	11
3.1.2	Lokální popis	14
3.2	Textura	15
3.2.1	Kookurenční matice	15
3.2.2	Transformace	16
3.2.3	Banky směrových filtrů	17
3.2.4	MPEG7	18
3.3	Tvar / obrys	19
3.3.1	Metody dělicí obrys podle význačných bodů	20
3.3.2	Zjednodušování tvaru pomocí Diskrétního křivkového rozvoje	23
3.3.3	Tvarový kontext	24
3.3.4	Fourierovy deskriptory	24
3.3.5	Momentové invarianty	26
3.3.6	Statistiky úhlů paprsků (Beam Angle Statistics)	26
3.3.7	Tensor Scale Descriptor	28
3.3.8	MPEG 7	29
3.4	Prostorové vztahy	30
3.4.1	Iconic indexing	30
3.4.2	Binární vztahy mezi regiony	31
3.4.3	Ostatní	32
3.5	Jiné	32
4	Typy segmentací	33
4.1	Shlukování o K-průměrech	33
4.2	Blob world	33
4.3	JSEG segmentace	34
4.4	Ostatní	34

5	Sumarizace a adaptivita signatur	35
5.1	Sumarizace popisů do signatur	35
5.1.1	Histogram	35
5.1.2	Lokalizovaný histogram	35
5.1.3	Shluková analýza	36
5.1.4	Ostatní	36
5.2	Rozdělení signatur podle jejich adaptivity	36
6	Popis měření podobnosti pomocí signatur	38
6.1	Rozdělení	38
6.2	Samotný vektor příznaků	39
6.2.1	Lineární prostor	39
6.2.2	Nelineární varieta	40
6.3	Podobnost signatur založených na regionu	40
6.3.1	Hausdorffova vzdálenost	41
6.3.2	Mallowsova vzdálenost	41
6.3.3	Earth movers vzdálenost	42
6.3.4	Integrated region matching (IRM) vzdálenost	42
6.3.5	Vylepšení proti základní porovnávací myšlence	43
6.4	Porovnávání lokálních příznaků	44
6.4.1	Vektorová kvantizace a generování seznamů kódů	44
6.4.2	Kullbackova-Leiblerova (K-L) vzdálenost	44
6.4.3	Fraktálové kódování	44
6.4.4	Ostatní metody	44
6.5	Ostatní způsoby měření	45
6.5.1	Anchoring (ukotvení)	45
7	Úvod do příznaků vyšší (sémantické) úrovně	46
7.1	Ontologie objektů	46
7.2	Strojové učení	46
7.2.1	Učení s učitelem (Supervised) učení	47
7.2.2	Učení bez učitele (Unsupervised)	47
7.3	Relevanční zpětná vazba	47
7.4	Sémantická šablona	48
7.5	Fúze mimoobrazových informací s obrazovými	48
8	Způsoby hodnocení systémů CBIR	49
8.1	Testovací data	49
8.2	Hodnocení porovnávacích metod	49
9	Závěr	51
	Literatura	52

Název práce: Podobnost bitmapových obrázků

Autor: Jiří Šilhán

Katedra (ústav): Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Mareš

e-mail vedoucího: mares@kam.mff.cuni.cz

Abstrakt: V předložené práci mapujeme metody používané pro vyhledávání obrázků dle obsahu. Rozebíráme příznaky založené na barvě, tvaru, textuře a prostorových vztazích. Zkoumáme metody konstrukce signatur na základě těchto příznaků a způsoby jejich porovnávání. Též se dotýkáme technik segmentace a metod hledajících sémantický význam ukrytý v obrázcích. V závěru zmiňujeme metody srovnávání systémů pro vyhledávání obrázků dle obsahu.

Klíčová slova: CBIR, vyhledávání, zpracování obrázků

Title: Similarity of bitmap images

Author: Jiří Šilhán

Department: Department of Applied Mathematics

Supervisor: Mgr. Martin Mareš

Supervisor's e-mail address: mares@kam.mff.cuni.cz

Abstract: In the present work we study methods used for content based image retrieval. We analyse features based on color, shape, texture and spatial relations. We review methods for construction signatures based on this features and their comparison. We touch the segmentation techniques and methods for obtaining hidden semantics in image content. At last but not least we study some of evaluation metrics for content based image retrieval systems.

Keywords: CBIR, retrieval, image processing

Kapitola 1

Úvod

Stále rychlejší rozvoj digitální techniky umožňuje tvořit a archivovat čím dál tím větší množství obrazového a audiovizuálního materiálu. S takovým rozvojem vyvstává potřeba efektivně uspořádat, třídít a vyhledávat.

K orientaci ve stále větších fotografických sbírkách se používá kategorizace, ta však přináší nutnost každý obrázek ručně anotovat. Význam je však skrytý i uvnitř obrazu, bylo by krásné ho umět popsat automaticky, nebo alespoň zhodnotit nějak podobnost obsažené informace. I takové hodnocení by nám ulehčilo práci se spoustou obrazového materiálu.

Metody které toto umí alespoň v omezené míře už byly nalezeny a s úspěchem se v praxi používají. Nejdříve se prosadily metody pracující s omezeným rozsahem rozpoznávané informace (už například OCR je příkladem takového systému), nyní však nastupují pokusy nasadit takovéhle systémy pro porovnávání/vyhledávání obrázku podle obsahu (dále *CBIR* z anglického *Content Based Image Retrieval*) v obecnější rovině, například i pro vyhledávání obrázků na webu. (viz. Google, Flickr, YouTube) Stoupá využití digitálního obrazu a jeho zpracování v mnoha disciplínách, nejen na webu, ale i v medicíně (popisují publikace [148], [101]), digitálních knihovnách, kultuře (publikace [58]), ochraně práv a v dalších odvětvích.

V souvislosti se systémy CBIR se zmiňují 2 druhy problémů (popisují Smeulders et al. [119], dále se zmiňují publikace [139], [23], či [82]):

Sensory gap – popisuje neurčitost jakéhokoli popisu obrázku, neboť k tomuto jednomu popisu se může vztahovat mnoho rozdílných obrázků. Je to propast mezi objektem samotným a informací kterou dokážeme získat ze záznamu tohoto objektu.

Semantic gap – popisuje obtížnost propojení vizuální podobnosti se sémantickou. Přesněji jde o problém spojení nízkourovňových příznaků obrázku, s významem (sémantikou), který bychom chtěli obsahu obrázku přiřadit.

V případě, že chceme vyhledávat obrázky bylo by dobré navrhnout nějakou metriku k porovnání jejich podobnosti. Ta by měla mít určité vlastnosti (těmi se pro metriku podobnosti tvaru zabývají například Latecki a Lakamper [74]).

Metrika by měla umožnit rozpoznání viditelně podobných objektů, které nejsou matematicky identické. Měla by být schopna oddělit zkreslení (digitalizační šum či např. segmentační chyby). Neměla by být závislá na měřítku, orientaci a v některých případech umístění objektů v obrázku. Měla by být schopná oklasifikovat jakýkoli obrázek.

Cílem práce je zmapovat alespoň známější metriky (a postupy) použité k porovnání obrazu dle obsahu. Metrika se většinou skládá z algoritmů extrakce příznaků obrazu, výpočtu signatury z těchto příznaků a algoritmu jak tyto signatury porovnat.

V kapitole 2 je uveden stručný popis obecného systému pro vyhledávání dle obrazu. Kapitola 3 se zabývá jednotlivými (nízkoúrovňovými) příznaky pomocí kterých se dá obraz popsat. V kapitole 4 je pak popsáno několik algoritmů používaných pro segmentaci obrazu na jednotlivé objekty. Způsoby jak se z těchto příznaků vytvoří signatura jsou zkoumány v kapitole 5. V kapitole 6 lze nalézt popis způsobů měření podobnosti pomocí signatur. Zběžnému popisu technik vyšší úrovně (většinou propojení se sémantikou) se věnuje kapitola 7. Způsoby porovnávání vyhledávacích systémů a závěr jsou pak obsahem kapitol 8 a 9.

Vzrůstající poptávka po indexaci a vyhledávání obrázků podle obsahu podpořila rozvoj mnoha metod. Shrnutím a srovnáním těchto se v poslední době zabývají práce (které též byly notnou inspirací této): za roky 94 – 2000 Smeulders et al. [119], pozdější Snoek a Worring [122], Torres a Falcao [130]. Speciálně vyššími technikami (spíše sémantikou ukrytou v obrazu) se zabývají Liu et al. [82]. Velice obsáhlá a asi nejaktuálnější je pak práce [23], kde lze najít široký popis metod používaných k vyhledávání nejen podle obsahu obrázku, ale i spojení s jinými obory.

Z českých prací se pak tímto tématem zabývají například Beneš a Zitová [7] (popisují jeden reálný systém a rozebírají metody použité nejen v něm).

Kapitola 2

Popis systému k hledání obrázků dle obsahu (CBIR)

Většina systémů CBIR, ostatně stejně jako jiné vyhledávací systémy, používá podobné schéma práce (schéma typického CBIRu viz obrázek 2.1). Do databáze, či nějaké chytré datové struktury, jsou uloženy ohodnocené obrázky. Ohodnocení je stěžejním bodem systému, neboť právě na něm nejvíce záleží efektivita a efektivnost celého systému. Nad touto databází se pak pokládají vyhledávací dotazy.

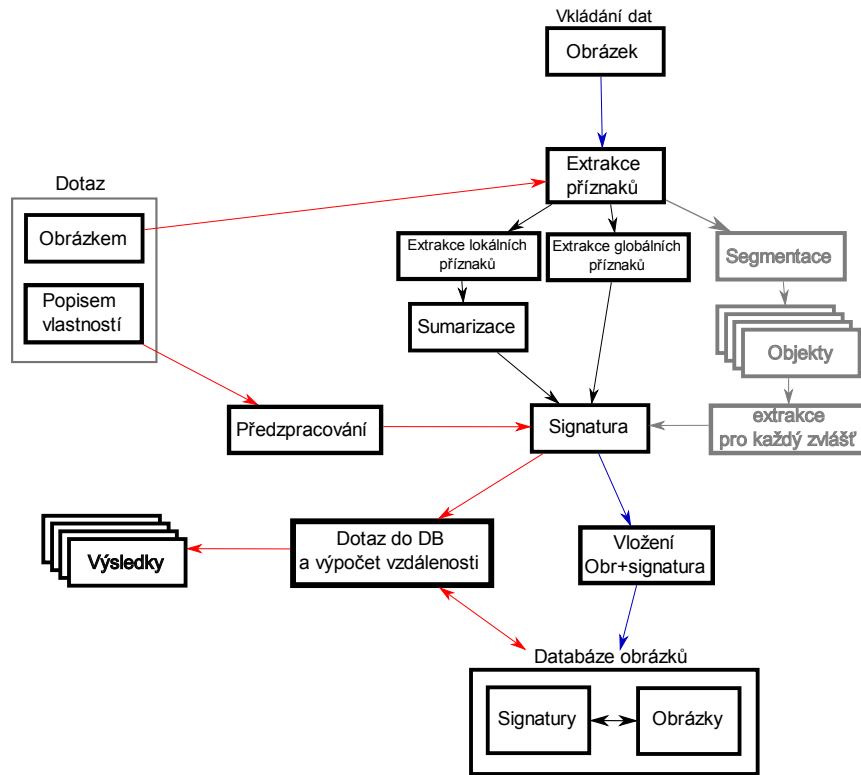
Jednotlivé systémy CBIR lze rozdělit podle způsobu pokládání dotazu. Buď jde o typ hledání podobného obrázku k existujícímu podkladu, tedy uživatel zadává konkrétní předlohu. Nebo uživatel popíše vlastnosti jaké by měl obsah hledaného obrázku mít. První kategorii lze dále dělit podle typu předlohy, buď se jedná o reálný obrázek, nebo jde o náčrt obrázku v prostředí systému, který umožňuje získat od uživatele další informace (např priority vlastností, jednoduše vektorový tvar oblastí apod.).

Při dotazu obrázkem většinou prochází předloha stejnou ohodnocovací procedurou jako při vkládání obrázků do databáze. V případě dotazu pomocí popisu vlastností či nákresem se ohodnocování liší (v případě popisu vlastností může jít i o jednoduchý intervalový dotaz nad signaturami).

Ohodnocování začíná zkoumáním příznaků obrázku. Příznaky lze dělit na globální a lokální. Globální příznaky se vztahují k celému obrázku, kdežto lokální většinou k pixelu či skupině (bloku nebo i segmentu). Pro zpracování lokálních příznaků je třeba vyřešit normalizaci jejich počtu (počet bloků pro různě velké obrázky může být různý), či příznaky nějak sumarizovat. V případě že se snažíme hledat obrázek obsahující podobné objekty, je vhodné obrázky segmentovat a popisovat segmenty separátně. Obrázek pak vlastně rozdělím na několik podobrázků které klasifikuji podobně jako u jiných systémů celek (takový systém pak bývá označován jako RBIR – Region Based Image Retrieval), navíc mohou přidat popis tvaru regionu (segmentu) a uspořádání regionů.

Z příznaků se pak tvoří signatura, popis který umožňuje dobré vyhledávání. V jednodušším případě jde o vektor čísel, kde každé číslo popisuje nějakou vlastnost. Obrázek si pak lze představit jako bod v n -rozměrném prostoru, kde každý rozměr je jedna z vlastností.

Obrázek 2.1: Schéma obecného systému k vyhledávání obrázků dle obsahu.



V poslední době je patrný posun k sémantice, systémy se snaží spojit příznaky s významem (respektive najít vztah), popsat lépe obsahovou složku (občas označovanou jako příznaky vyšší úrovně). K tomuto se používá například algoritmů strojového učení.

Kapitola 3

Rozbor příznaků nižší úrovně

Požadavky na příznaky (přesněji na jejich popis) se liší podle typu využití. Pro vyhledávání typů materiálu při restaurování obrazů, bude jistě lepší klást důraz na jinou množinu než při hledání čínských znaků, či obecném vyhledávání. Otázka výběru typů příznaků pak získává trošku filozofický nádech hledání podstaty podobnosti.

V případě hledání snímků podobných objektů, by měly být příznaky nejspíše invariantní k afinním transformacím (zamýšlí se nad tím například článek [117]). Tímto způsobem lze totiž simulovat v menší míře změnu perspektivy. Při požadavku invariance ke změně perspektivy ve větší míře se ale dostáváme k postupům jako jsou konstrukce 3D deskriptorů či ke zkoumání sémantiky obrázku.

Vhodná je též otázka použití segmentace a chápání obrázku jako množiny objektů místo celkového pohledu. V poslední době můžeme pozorovat u systémů CBIR posun od globálních popisů k segmentování a k distribucím lokálních příznaků. Globální popis umožňuje jednodušší metody porovnání, kdežto popis založený na regionech (tedy segmentovaný) je chápán jako bližší lidskému vnímání (viz Wang et al. [140]).

3.1 Barva

Při popisu barevných vlastností obrazu je obecně dobrá myšlenka převést obrázek do barevného prostoru, kde jsou složky navzájem co nejméně korelované (tedy prostoru s co nejmenší redundancí). Častěji používané systémy jsou LUV, Lab, YUV, HSV. Standard MPEG7 dokonce využívá vlastní barevný systém hue-min-max-difference (HMMD), kde se k popisu barvy používá trojice tón (hue) barvy, největší a nejmenší RGB složka, alternativně jako trojice tón, rozdíl minima a maxima RGB složek, aritm. průměr minima a maxima RGB složek. Na optimalizaci barevného systému byl kladen důraz spíše v minulých letech, nyní se pozornost přesouvá k sumarizaci příznaků.

Standard MPEG7 [91] uvádí pro popis barevných vlastností obrázků deskriptory: barevný histogram kódovaný Haarovou transformací (tzv. *škálovatelný deskriptor barvy*), *histogram barevné struktury*, *barevné rozvržení* a z lokálních *dominantní barvy*.

3.1.1 Globální popis

Barevný histogram

Barevný histogram je velmi hrubým popisem zastoupení barev obrázku používaným v raných CBIR systémech (viz například článek [125]). Postup při konstrukci barevného histogramu je následující:

1. Rozdělíme barevný prostor na přihrádky, ne nutně stejně veliké.
2. Spočítáme počet pixelů spadající do každé přihrádky v celém obrázku.
3. Výsledné počty (respektive procentuální zastoupení) se většinou prezentují jako vektor hodnot.

Barevný histogram je popis invariantní k otočení či posunutí, jeho prostorová složitost je $O(n)$ kde n je počet přihrádek a lze ho zkonstruovat v čase $O(m)$ kde m je počet pixelů v obrázku. Porovnávání obrázků pomocí histogramů se provádí jako výpočet euklidovské (jinak též L_2) či manhattanské (L_1) vzdálenosti mezi dvěma vektory (histogramy). Využití barevných histogramů je podrobněji popsáno Strickerem a Orengem [124].

Barevný korelogram

Struktura popisující prostorové vztahy mezi dvojicemi barev se nazývá barevný korelogram. V publikaci [50], mimo jeho návrhu, autor experimentálně ukazuje jak tento popis překonává barevné histogramy. Neformálně jde o tabulku γ indexovanou dvojicemi barev. Obsah pole γ_{c_i, c_j} je vektor, jehož k . prvek je pravděpodobnost že ve vzdálenosti k od pixelu barvy c_i bude pixel barvy c_j . Barvy před zpracováním prochází kvantizací, takže struktura není příliš prostorově náročná (přesněji $O(m^2 d)$ kde m je počet barev po kvantizaci a d je maximální vzdálenost do které korelogram počítám). Huang et al. [50] též popisují postup jak barevný korelogram zkonstruovat s časovou složitostí $O(n^2 d)$. Vzdálenost pomocí korelogramu článek [50] určuje podle upravené L_1 metriky:

$$d(I_1, I_2) = \sum_{i, j \in (1..m), k \in (1..d)} \frac{|\gamma_{c_i, c_j}^{(k)}(I_1) - \gamma_{c_i, c_j}^{(k)}(I_2)|}{1 + \gamma_{c_i, c_j}^{(k)}(I_1) + \gamma_{c_i, c_j}^{(k)}(I_2)} \quad (3.1)$$

Kde I_1, I_2 jsou hodnocené obrázky, m počet barev po kvantizaci a $d()$ výsledná vzdálenost.

Vektor barevné koherence

Vektor barevné koherence (popisuje publikace [104]) je postup obdobný histogramu, navíc rozšířený o určení zdali jsou pixely spadající do dané přihrádky obsaženy ve spojitě barevné oblasti. Obrázek je nejdříve zprůměrován přes 8 okolí, načež jsou kvantizovány jeho barvy na n odstínů. Pak se provede určení spojitých komponent (obrázek je chápán jako graf, kde pixely jsou vrcholy a hrany mezi nimi existují

pokud jsou ve svém 8-okolí a navíc mají stejnou kvantizovanou barvu). Ze spojitých komponent určíme které pixely jsou:

koherentní ve chvíli, kdy je součástí spojitě komponenty o nejméně π pixelech,
nekoherentní v opačném případě.

Vektor barevné koherence sestrojím ze dvojic počtů koherentních (α_j) a nekoherentních (β_j) pixelů pro každou barvu (j):

$$V_e = \langle (\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n) \rangle \quad (3.2)$$

Vzdálenost se pak spočítá jako součet rozdílů odpovídajících složek:

$$d(I_1, I_2) = \sum_{j=1}^n |\alpha_1 - \alpha_2| + |\beta_1 - \beta_2| \quad (3.3)$$

Díky kvantizaci je tento postup citlivý na nahuštění pixelů do jedné přihrádky, což je například u RGB způsobeno zvýšením jasu celého obrázku, nabízí se řešení v podobě použití jiného barevného modelu. Též je citlivá volba hranice π .

Klasifikace vnitřních a hraničních pixelů (BIC histogram)

Dalším rozšířením myšlenky barevných histogramů je metoda navržená Stehlingem Nascimentem a Falcaem [123].

Barevný prostor (RGB) obrázku je kvantizován (v práci [123] na 64 ($4 \times 4 \times 4$) odstínů). Jednotlivé pixely jsou klasifikovány na hraniční a vnitřní (border/interior) podle pravidla:

hraniční – alespoň jeden pixel ze 4-okolí má jinou barvu, nebo je pixel na okraji obrázku,

vnitřní – ostatní případy.

4-okolí se používá pro menší výpočetní náročnost (oproti 8-okolí). Pro každou kategorii se pak spočítá barevný histogram. K porovnání histogramů zavádí publikace [123] tzv d_{Log} vzdálenost:

$$d_{Log}(H_1, H_2) = \sum_{i=0}^{i < M} |f(H_1[i]) - f(H_2[i])| \quad (3.4)$$

kde H_1, H_2 jsou histogramy, M počet jejich přihrádek a f je funkce:

$$f(x) \begin{cases} 0 & \text{pokud } x = 0 \\ 1 & \text{pokud } 0 < x \leq 1 \\ \lceil \log_2 x \rceil + 1 & \text{v jiném případě} \end{cases} \quad (3.5)$$

Použití takové míry umožňuje zmenšit efekt homogenního pozadí (pokud pozadí pokrývá velkou plochu a barva jeho pixelů se vejde do jedné přihrádky, jsou obrázky s pozadím stejné barvy klasifikovány jako podobnější než obrázky se stejným popředím a pozadím v jiné barvě).

Kvůli úspoře místa se jako signatura ukládají hodnoty $f(x)$ místo původních, dojde tak k 50-procentní úspoře (pak také stačí porovnávat $f(x)$ klasickou L_1 metrikou). Pak mi na jeden BIC histogram stačí 64 bajtů (64 přihrádek po 4 bitech pro 2 kategorie).

Histogram barevné struktury

Deskriptor uvedený v rámci standardu MPEG 7 [91] uvádí další možné vylepšení barevného histogramu. Používá barevný prostor HMMD nelineárně kvantizovaný do několika úrovní od 32 do 184 odstínů (přesně popsáno v publikaci [91]). Pro zajištění škálovatelnosti uvádí MPEG 7 postup jak převádět deskriptory o různých úrovních kvantizace (samozřejmě se vždy snižuje úroveň toho s úrovní vyšší). Definuje „*structuring element*“ což je mřížkový průzor o rozměrech 8x8, který je postupně posouván po obrázku. Pro každou kvantizovanou barvu se pak spočítá počet pozic při průchodu průzoru, ve kterém byla vidět alespoň jedním z okének. Deskriptor je pak tento histogram normalizovaný celkovým počtem pozic průzoru v obrázku. Porovnávání se provádí běžnými metodami srovnání histogramů (L_1 metrika apod.).

Barevné momenty

Pokus co nejvíce shrnout vlastnosti barevného rozložení v obrázku vyústil v deskriptor popisovaný v díle [124]. Obrázek je popsán prvními 3 statistickými momenty pro každou barevnou složku (v HSV prostoru). Momenty vypadají takto:

průměr (1. moment)

$$E_i = \left(\frac{1}{N}\right) \sum_{j=1}^N p_{i,j} \quad (3.6)$$

rozptyl (2. moment)

$$\sigma_i = \sqrt{\left(\frac{1}{N}\right) \sum_{j=1}^N (p_{i,j} - E_i)^2} \quad (3.7)$$

šikmost (3. moment)

$$s_i = \sqrt[3]{\left(\frac{1}{N}\right) \sum_{j=1}^N (p_{i,j} - E_i)^3} \quad (3.8)$$

kde $p_{i,j}$ je hodnota i . barevné složky j . pixelu, N je počet pixelů v obrazu. Rozdíl obrázků se pak určí jako:

$$d(I_1, I_2) = \sum_{i=1}^3 w_{1,i} |E_{1,i} - E_{2,i}| + w_{2,i} |\sigma_{1,i} - \sigma_{2,i}| + w_{2,i} |s_{1,i} - s_{2,i}| \quad (3.9)$$

kde $w_{j,i}$ je důležitost j . momentu i . barvy.

Protože pracujeme s malou podmnožinou momentů není tato vzdálenost metrikou (nestejná rozdělení můžou mít nulovou vzdálenost). Navíc je tento popis velmi hrubý a nemá nijak velikou diskriminabilitu.

Barevné rozvržení

K popisu prostorového rozložení barev v obrázku se používá metody *Barevného rozvržení*. Obrázek je rozdělen do oblastí (v případě MPEG7 [91], je to 8×8 obdélníků při použití YCbCr barevného prostoru). Pro každou oblast je pak spočtena reprezentativní (většinou průměrná) barva. Z těchto hodnot se pak vytvoří vektor který je deskriptorem (jedná se vlastně o jakési podvzorkování obrázku). MPEG 7 navíc přidává zpracování tohoto 8×8 obrázku pomocí diskrétní cosinové transformace. Deskriptorem pak je několik kvantizovaných nízkofrekvenčních koeficientů (více v díle [91]) Pro určení rozdílnosti dvou deskriptorů barevného rozložení se pak v publikaci [91] používá míra:

$$d(D_1, D_2) = \sqrt{\sum_i w_{Y,i} (D_{1,Y}[i] - D_{2,Y}[i])^2} + \sqrt{\sum_i w_{Cb,i} (D_{1,Cb}[i] - D_{2,Cb}[i])^2} + \sqrt{\sum_i w_{Cr,i} (D_{1,Cr}[i] - D_{2,Cr}[i])^2}$$

kde D_1, D_2 jsou kompletní deskriptory, $D_{i,j}[k]$ k . složka i . deskriptoru barvy j a $w_{i,j}$ je váhový koeficient pro j . složku deskriptoru barvy i (vyšší váhy jsou přidělovány složkám odpovídajícím koeficientům pro nízké frekvence).

3.1.2 Lokální popis

Průměrná barva přes určitý blok

Nejjednodušším lokálním příznakem pro popis barvy je její průměr v daném regionu. Není to však dobrá volba pro regiony s nehomogenní barevností. Pro specifické úlohy se hodí, navíc lze rozšířit o váhové ohodnocení.

Dominantní barva přes určitý blok

Nevýhodám průměrování se vyhýbá výběr *dominantní barvy*. Jedná se o barvu která má v daném regionu největší zastoupení, prakticky je to ta kterou v regionu „vi-

díme“. Určím ji jednoduše jako průměrnou hodnotu barvy v nejobsáhlejší přihrádce barevného histogramu.

Například v článku [83] se dominantní barva počítá takto:

1. spočítám histogram přes HSV prostor (o $10 \times 4 \times 4$ přihrádkách)
2. vyberu přihrádku s největším obsahem
3. průměrná HSV hodnota vybrané přihrádky histogramu je dominantní barva

Oproti tomu MPEG7 [91] nejdříve podrobí obrázek (či segment) barevnému shlučování, po kterém zbude jen několik barev. Popis je pak tvořen vektorem n -tic (barva, její procentuální zastoupení, barevný rozptyl (volitelně)) a číslem určujícím celkovou homogenitu dominantních barev v objektu. Pro popis se většinou využívá maximálně 8 dominantních barev. Porovnávání je prováděno pomocí upravené L_2 metriky (více viz práce [91]).

3.2 Textura

Další vlastností obrazu někde na půl cesty mezi velice lokální barvou a globálním tvarem je textura. Má charakter veličiny popisující vlastnosti okolí pixelu (a z toho vyvstává otázka jak velké okolí zvolit, zabývají se jí Carson et al. [16]). Není to nijak přesně definovaná veličina, vznikla jako pokus zachytit hladkost, zrnitost, opakování, směrovost a vzor v obrazu. Používá se k zachycení vlastností povrchu/vzoru, typicky lze pomocí textury rozlišit objekty s povrchem z kůže, stromy, cihly, obložky. Je to lokální vlastnost důležitá pro segmentaci (obzvláště v medicíně a mapování), protože dokáže rozlišit dva objekty, které ač jsou stejné barvy, mají jiný povrch, respektive jeho vzorek. Textura je většinou počítána nad šedotónovou variantou obrázku (například nad složkou Y z $YCbCr$ barevného prostoru).

3.2.1 Kookurenční matice

Jedna z nejstarších technik, dodnes využívána pro popis textury je *kookurenční matice* definována již v 70. letech (popsána například v publikaci [45]). Jedná se o popis parametrizovaný posunem v ose x (d_x) a y (d_y). Formálně je to matice C o velikosti $n \times n$ (pro n rovno počtu odstínů), kde políčko $C(i, j)$ udává počet pixelů odstínu j , které jsou posunuty v ose x o d_x a v ose y o d_y oproti jakémukoli pixelu odstínu i . Pro udržení rozumné velikosti se před výpočtem kookurenční matice podrobí obrázek kvantizaci barev. K popisu jednoho obrázku je nutno vypočítat několik různě parametrizovaných kookurenčních matic (pro různé směry a velikosti posunutí), navíc jsou tyto matice většinou řídké. Pro tyto vlastnosti se používá místo samotných matic jejich matematických popisů. Jedná se převážně o vlastnosti občas nazývané *Haralickovy příznaky* jako entropie, kontrast, homogenita, rozptyl a další (více viz práce [45], [72]) a [7]). Rozšíření kookurenčních matic pro použití na prostorová data lze najít v článku [72].

3.2.2 Transformace

Obsáhlá skupina metod k popisu textur je založena na myšlence využití technik pro zpracování obrazu a komprese. Jde o aplikaci nějaké transformace a následný výběr či zpracování jejích koeficientů.

Problém ovšem nastává pokud chci takovouto metodu použít na segment neobdélníkového tvaru. Transformace jsou většinou definovány pro obdélníkové ba dokonce čtvercové oblasti, takže je třeba segment nějak upravit na tuto formu.

Řešení tohoto problému jsou diskutována například Liem et al. [82]:

1. Budeme počítat texturové příznaky z bloků pixelů, řekněme 4×4 . (pak se stává výsledek lokálním popisem)
2. Rozšíříme region na obdélník přidáním nějakého pozadí, to nám ale ovlivní popis.
3. Použijeme největší vepsaný obdélník. Tato metoda funguje dobře, pokud je textura homogenní a obdélník nese dostatek informace k popisu (pokud je tvar pokroucený a úzký, není v obdélníku dostatek reprezentativního obsahu).
4. Použijeme složitější metodu. Například metoda prezentovaná v práci [84] doplní tvar na obdélník a pak hledá koeficienty pro nejlepší popsání oblasti pomocí iterativní projekce mezi obrázkem a jeho transformací.

Wavelety

Waveletová transformace popsána prací [24] je hojně využívána pro schopnosti separace frekvencí, práci v několika rozlišeních a určení směrovosti. Je to technika využívaná též v moderních způsobech komprese obrazu (například JPEG2000). Bohužel nedokáže zaručit invarianci k posunutí, otočení a v určitém smyslu i měřítka.

Energie pásma jako příznak V publikaci [57] je použita jednoúrovňová 9/7 bi-ortogonální dekompozice (pro popis doporučuji publikaci [18]) k určení lokálních texturových příznaků pro segmentaci malých obrázků (200×200 pixelů). Dekompozice se provádí na okně 9×9 pixelů a pro další využití se vybírají jen LH a HL pásma (tedy Low-pass v horizontálním směru, high-pass ve vertikálním a opačně). Okno se postupně posouvá po celém obrázku a z výsledných pásem se jako příznak počítá medián energie koeficientů.

$$E = \frac{1}{mn} \sqrt{\sum_{i=0}^m \sum_{j=0}^n c_{i,j}^2} \quad (3.10)$$

Kde E je energie, m, n jsou rozměry pásma a $c_{i,j}$ je koeficient s daným indexem (relativně v pásmu). Textury pak dělí na 4 typy – hladká (nízká energie v obou pásmech), horizontální (vysoká v HL), vertikální (vysoká v LH) a komplexní (vysoká v obou).

Systém *Simplicity* [140], používá k segmentaci a k následnému popisu texturovaných obrázků podobný postup. Používá jinou waveletu – Haarovu a Daubechies-4, barevný prostor LUV (pro popis textury jen L složku). Místo putujícího okna dělí obrázky na bloky, a nad nimi počítá jednu úroveň dekompozice. Z tohoto získá 3 příznaky, energii LH pásma (zde se projeví horizontální pruhy v textuře), energii HL pásma (zde vertikální) a energii HH pásma.

Podobný postup popisují Li a Tian [80]. Dekompozice je však prováděna nad celým obrázkem a je dvouúrovňová. Získáme tak 9 pásem. Z kombinací energií těchto pásem je pak vytvořeno 7 deskriptorů, vektorů. Autor testuje pro dekompozici Haarovu a některé Daubechies wavelety.

Waveletové příznaky jako parametry statistických modelů V článku [110] je popsán statistický model pro popis texturovaných obrázků. Model je parametrizovaný několika statistickým atributy odvozenými z koeficientů komplexní waveletové transformace. K dekompozici je použito schéma „steerable pyramids“ [118], které umožňuje rozklad parametrickými směrovými filtry (oproti Malatově rozkladu se Low-pass pásmo rozkládá na počet pásem určený požadovanými směry). Dále je prezentován způsob generování náhodných obrázků odpovídajících danému modelu.

Další ze systémů [28] využívajících statistické modelování se snaží specifikovat každé z pásem waveletového rozkladu (3-úrovňový rozklad Daubechies-4 waveletou, tedy popis textury pomocí 2×9 čísel) pomocí 2 parametrů zobecněného modelu gaussovské hustoty (viz práce [135]). K porovnání modelů používá Kullbackovu-Leiblerovu vzdálenost popsanou v článku [92].

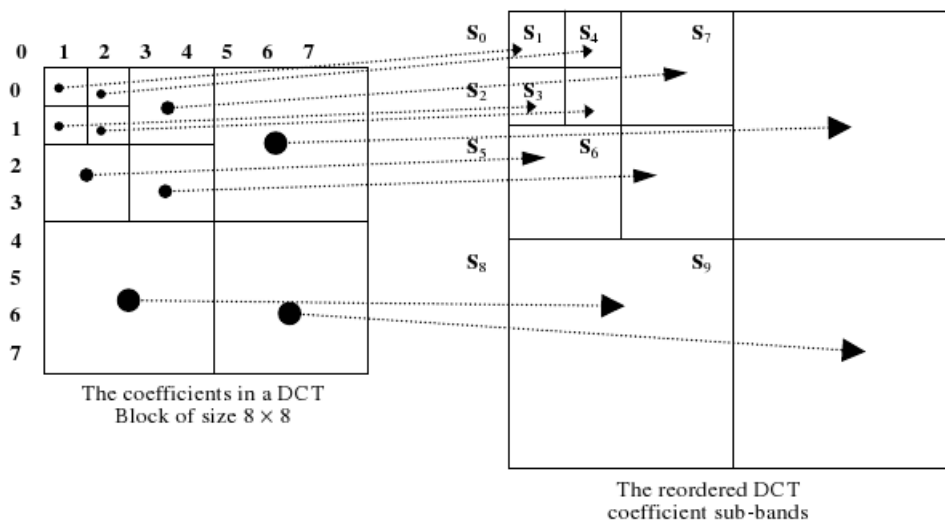
Diskrétní cosinová transformace

Diskrétní cosinová transformace je nejčastěji využívaná technika pro ztrátovou kompresi obrazu. A tak je nasnadě využít již spočtených koeficientů pro popis obrázku (jak píše Huang [51]). Navíc pro texturové popisy přispívá to, že je (např ve standardu JPEG [105]) *DCT* počítána pro bloky, takže jde o lokální popis. Huang [51] používá k popisu denormalizované a přeuspořádané koeficienty (tak aby byla frekvenční pásma rozdělena do oblastí, viz obrázek 3.1). Deskriptor je pak vektor statistických příznaků (pro každé pásmo 2 čísla) pásem. Vzdálenost mezi deskriptory se počítá jako normalizovaná manhattanská. Popis je DCT bloků 4×4 pixely nad obrázkem v několika měřítkách se pak v práci [79] používá ke klasifikaci textur pomocí SVM.

3.2.3 Banky směrových filtrů

Schéma směrových filtrů je vcelku jednoduchá myšlenka. Pokud chceme postihnout určitou frekvenci v určitém směru, s určitým fázovým posunem, proč nepoužít masku (filtr) ve frekvenčním pásmu (například diskrétní Fourierovy transformace). Takové masky lze generovat pomocí parametrizované funkce (parametry jsou frekvence, směr, měřítko, fázový posun...). Skupinám filtrů pokrývajícím určité parametry se říká *Banky filtrů*. Jejich aplikace se pak provádí konvolucí ve frekvenčním pásmu, výsledek lze popsat pomocí jednoduchých statistických metod. Nejznámější a nejčastěji

Obrázek 3.1: Schéma přeuspořádání koeficientů, zdroj [51].



využívané jsou *Gaborovy filtry*.

Pokus o simulaci lidského vnímání textury popisuje článek [86], používá mimo jiné konvoluce se směrovými filtry. V publikaci [61] můžeme najít popis využití banky Gaborových filtrů k neasistované segmentaci obrazu podle texturových příznaků.

V práci [90] je popsáno využití Gaborových filtrů k extrakci texturových příznaků v systému *Netra* [89]. Podobné Gaborovy filtry jsou též použity pro konstrukci *Deskriptoru homogenní textury* ve standardu MPEG7 [91].

3.2.4 MPEG7

Texturové příznaky definované standardem popisuje publikace v [91], též v ní můžeme najít výsledky testování na Brodatz texturách a množině obrázků Corel stock photos.

Deskriptor homogenní textury

Pro kvantitativní charakteristiku texturově homogenních regionů (k vzájemnému porovnávání obrázků) používá MPEG7 banky Gaborových filtrů. Jde o banku 30 filtrů o 6 směrech (rovnoměrně pokrývá rozpětí ϕ) a 5 vzdáleností od středu frekvenčního pásma (30 filtrů).

Obrázek popíšeme pomocí průměru a směrodatné odchylky textury a dvojice energie (ve frekvenčním pásmu) a směrodatné odchylky energie pro aplikaci každého filtru. Pro dvojice se používá logaritmické měřítko. Deskriptorem je pak n-tice [průměr frekv. amplitudy, sm. odchylka, energie 1. filtru, ..., e.30. filtru, odchylka 1., ..., o.30. filtru]. Porovnání se provádí pomocí normalizované L_1 metriky.

Lokální distribuce hran (EHD – edge histogram deskriptor)

Obdoba barevného rozvržení pro textury se nazývá *deskriptor hranových histogramů*. Popisuje rozložení hran v regionu, což je důležité pokud region není texturově homogenní. Postup je vcelku přímočarý. Region je rozdělen do 4×4 podobrázků (tedy celkově 16 podobrobázků). V každém podobrázku počítáme histogram hran. Rozlišujeme 5 typů hran – *vertikální, horizontální, 45 diagonální, 135 diagonální a neorientované*. Výpočet hranových histogramů provedeme rozdělením podobrázku na ještě menší podjednotky a použitím hranových detektorů.

Takovým postupem získáme 16 histogramů po 5 přihrádkách. Kvantizujeme každou přihrádku na 3 bity a tím vytvoříme deskriptor o 240 bitech. Vzdálenost počítáme jako L_1 metriku nad samotným deskriptorem, či jeho odvozeními (nahrazení 3-bitových hodnot pomocí adekvátního vzoru před kvantizací, sestavení rozšířeného histogramu nad deskriptorem) Deskriptor hranových histogramů je výhodný k reprezentaci přírodních obrázků pro vzájemné porovnávání obrázků.

Texture browsing deskriptor

V roce 87 definoval Tamura [126] 6 příznaků k popisu textury doteď jsou zmiňovány pod názvem *Tamurovy příznaky*. Jedná se o hrubost (coarseness), směrovost (directionality), pravidelnost (regularity), kontrast, linelikeness, drsnost (roughness)

MPEG 7 z těchto vlastností používá první 3 ke konstrukci tzv. *Texture browsing deskriptoru*. Zbylé jsou závislé na prvních 3. *Texture browsing deskriptoru* je deskriptor založený na aplikaci bank měřítkově a směrově citlivých pásmových filtrů (používá stejného filtrování jako *deskriptor homogenní textury*). Ve 14 bitech dokáže popsat hrubost, směrovost i pravidelnost. Pro případ více směrů a hrubostí/měřítek v obrázku je omezen počet příznaků dané veličiny na maximálně 2, a jedná se pak o dvojice (tedy směr+hrubost). Příznaky jsou v deskriptoru rozděleny takto:

směrovost – 3 bity pro 2 různé exempláře, rozsah 0–150 stupňů, po 30, 0 znamená beze směru,

hrubost – 2 bity pro 2 různé exempláře,

pravidelnost – 2 bity, rozsah 0–3, 0 znamená náhodná, 3 přesně pravidelná.

Texture browsing desk. může být díky stejné filtraci počítán společně s deskriptorem homog. textury.

3.3 Tvar / obrys

Klíčovým atributem segmentovaných regionů je tvar. Je to jedna z odměn za správné vysegmentování obrázku, bohužel, je velmi odvislý od přesné segmentace. Velká důležitost se tvarovým atributům přikládá v popisu lidských výtvorů a vůbec tříd objektů, které mají společný tvar, ale vyskytují se v mnoha barevných či povrchových modifikacích.

Bylo by dobré, aby příznak popisující tvar byl :

1. kompaktní
2. jednoduchý
3. odolný vůči šumu
4. invariantní alespoň k afinním transformacím
5. pokud možno dobře indexovatelný

Podrobným popisem a dělením metod popisujících tvar v obrázku se zabývá například práce [147]. Rozděluje metody na ty co jsou založené na popisu regionu a na ty které popisují jen hranici.

V publikaci [93] je popsán posun od globálních tvarových příznaků k lokálnějším. Mezi globální (využívá systém QBIC [38]) počítáme například kulatost, excentricita, kruhovost, orientaci, plochu, obvod, množinu čtverců/trojúhelníků která pokrývá tvar a jiné. Taková reprezentace má jednu zásadní nevýhodu, popisuje tvar kompletní takže v případě překrytí či oříznutí je nepoužitelná. Lokální reprezentace by měly tuto nevýhodu eliminovat (proto se budeme zabývat převážně lokálními popisy).

3.3.1 Metody dělí obrys podle význačných bodů

Velikou skupinou jsou metody, které tvar (konturu) rozdělí podle bodů s nějakými specifickými vlastnostmi a popíší segmenty (nebo jen ty význačné body). Popisy pak mohou podléhat sumarizaci, nebo se z nich vytvoří víceméně lokální popis umožňující porovnávání částí tvarů.

K popisu algoritmů se nám bude hodit znát definici *zakřivení*:

Máme-li křivku zadanou pomocí parametrické rovnice

$$\vec{r}(u) = (x(u), y(u)) \quad (3.11)$$

kde u je volný parametr, pak *zakřivení* spočítáme pomocí vzorce:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}} \quad (3.12)$$

kde $\dot{x}(u), \dot{y}(u)$ jsou první a $\ddot{x}(u), \ddot{y}(u)$ druhé derivace.

Největší změna zakřivení

Článek [93] uvádí metodu založenou na popisu tvaru zjednodušením na body s největší změnou zakřivení. Tvar je pak reprezentován pomocí seřazeného seznamu segmentů. Segment se sestává z několika význačných bodů (těch s největší změnou). Kvůli invarianci segmentů k afinním transformacím se každý segment normalizuje podle velikosti a orientace prvních 2 bodů (tak aby 1. bod měl souřadnice (0,0) a 2. (1,0)).

Deskriptor segmentu pak tvoří:

seznam souřadnic bodů (po normalizaci) $((x_1, y_1), \dots, (x_n, y_n))$

popis normalizační transformace $P = (S, T_x, T_y, \phi)$, kde S je měřítko, T_x, T_y posunutí po daných osách a ϕ je úhel otočení.

Tvar je pak popsán pomocí skupiny těchto segmentových popisů, případně rozšířeného o body dotyku a další specifiky. Při porovnávání segmentů se používá euklidovské vzdálenosti mezi body (krom prvních 2, ty jsou díky normalizaci všude stejné).

Minimum zakřivení

Beretti et al. [8] dělí vyhlazenou křivku tvaru v lokálních minimech zakřivení. Každý segment popíšeme pomocí *maxima zakřivení* a *orientace* (úhel mezi středem spojnice krajních bodů, maxima zakřivení a vodorovné osy). V případě že je tvar kruh (a tedy nemá žádné minimum), popíšeme ho jako jeden segment se zakřivením 0 a „*do not care*“ orientací.

Deskriptor objektu je tedy uspořádaná množina dvojic (zakřivení a orientace). To že jsou 2 tvary podobné pak odpovídá tomu, že sdílí segmenty s podobným zakřivením a orientací.

Míru podobnosti určíme podle následujících pravidel:

podobnost segmentů – metrika :

$$d(seg_1, seg_2) = a \times |seg_{1,zakriveni} - seg_{2,zakriveni}| + (1-a) \times |seg_{1,orientace} - seg_{2,orientace}| \quad (3.13)$$

kde a je váha důležitosti zakřivení a seg_1, seg_2 jsou popisy segmentů.

podobnost tvarů Vezmeme tvar s menším počtem segmentů. Pro každý jeho segment vybereme nejbližší nepřirazený segment v druhém tvaru. Vzdálenost určíme jako aritmetický průměr ze vzdálenosti dvojic které jsou bližší než předem daný práh (pak se považují za nepodobné).

Nejedná se o metriku, protože nefunguje trojúhelníková nerovnost. Tvary jejichž podmnožiny se rovnají si vůbec nemusí celkově odpovídat.

Tento způsob popisu není invariantní vůči otáčení, možná by pomohlo normalizovat otočení pozicí nejpodobnějších segmentů. Těž může vydávat špatné výsledky v závislosti na prohození segmentů. Ke zlepšení efektivity vyhledávání segmentů v článku [8] doporučují index segmentů sestavený pomocí M-stromu popsáném v publikaci [17].

Inflexní body

Snaha zlepšit porovnávání u překrytých či ořízých tvarů vyústila v postup popisovaný v práci [107]. Tento postup se na rozdíl od jiných dá aplikovat na otevřené tvary.

Zde popsaný algoritmus nejdřív aproximuje tvar pomocí bikubického B-spline (více viz Smith [121]). Z této aproximace určíme zakřivenost a následně i inflexní body. Pak můžeme konečně tvar rozdělit na konvexní a konkávní segmenty oddělené inflexními body.

Segmenty popíšeme pomocí 3 příznaků:

rotační úhel – úhel mezi tečnami v obou bodech inflexe, kladný pro konvexní, záporný pro konkávní segmenty,

délka – délka segmentu (normalizovaná),

plocha – plocha uzavřená obloukem a spojnici inflexních bodů (normalizovaná).

Pomocí technik dynamického programování pak najdeme nejvíce si odpovídající dvojice segmentů a zhodnotíme podobnost (je to nejnižší cena párování segmentů).

Konturové a segmentové význačnosti (Contour and Segment Saliences)

Dva podobné postupy založené na aplikaci „*Image Foresting Transform*“ (více viz publikace [31]) a oblastech vlivu prezentují Torres a Falcao [131]. Snaží se najít význačné body, které formují tvar a pomocí nich ho popsat, zároveň se pokouší vyhnout ovlivnění šumem (u hladkého tvaru jsou takovéhle body většinou ty s velkou změnou zakřivení, pro zašuměný tvar je ale zakřivení příliš lokální vlastnost).

Obrázek je chápán jako graf, kde pixely jsou vrcholy spojené hranami (v rámci okolí). Pixely kontury jsou použity jako tzv „kořenové pixely“. Oblast vlivu jednoho kořenového pixelu je pak množina pixelů, k nimž vede nejkratší cesta přes hrany právě z tohoto kořenového pixelu.

K určení oblastí vlivu se využívá upraveného Dijkstrova algoritmu pro vyhledávání nejkratší cesty. Modifikace spočívá v paralelním počítání s více zdroji a v použití obecné funkce k ohodnocení cest. Tento algoritmus se nazývá „*Image Foresting Transform*“.

Když máme určené oblasti vlivu, můžeme vybrat kandidáty na význačné body. Mohly by to být ty konturové body které mají největší oblasti vlivu (můžeme použít histogram příslušnosti a prahování). Tímto postupem sice získáme zaručeně význačné body, ale můžeme v případě velice zašmodrchaného tvaru některé zanedbat (blízké se vyruší). Algoritmus popsaný v článku [131] používá vnitřní a vnější kostry k zachycení všech význačných bodů. Body se klasifikují na konvexní a konkávní podle poměru vnitřní a vnější oblasti vlivu (pokud je vnější větší tak je konvexní).

Deskriptor se pak skládá z hodnot význačnosti (plocha oblasti vlivu, u konkávních bodů násobená -1) význačných pixelů a jejich umístění na obrysu.

K porovnávání se používá heuristický srovnávací algoritmus používaný též v algoritmu CSS, viz 3.3.8.

Segment Saliences Rozšířením myšlenky význačností kontury jsou segmentové význačnosti, popsané též v práci [131]. Tento algoritmus místo význačností bodů, využívá význačností segmentů (jejich počet je předem pevně daný, kontura se na ně

dělí stejnoměrně). Pro každý segment spočítáme součety vnitřních a vnějších oblastí vlivu pro všechny pixely jeho kontury. Deskriptorem je pak vektor rozdílů těchto součtů (tedy pro každý segment jedno číslo). Deskriptory se porovnávají pomocí algoritmu OCS, stejně jako v postupu 3.3.6.

3.3.2 Zjednodušování tvaru pomocí Diskrétního křivkového rozvoje

V publikaci [74] se autor snaží vyvinout postup k porovnávání tvarů odolný vůči zašumění. Prostředkem tomu má být co největší zjednodušení tvaru (a tímto pádem i odstranění šumu) při zachování podobnosti s originálem. K pochopení postupu je třeba si uvědomit, že „každá digitální křivka může být chápána jako polygon, bez ztráty informace“. Polygon mohu přesně popsat posloupností jeho bodů. Zjednodušení křivky tedy provádím tak, že z této posloupnosti jeden bod vyřadím, aby se výsledný tvar změnil co nejméně.

Cenu změnu mohu určit takto:

$$K(us_1, us_2) = \frac{uhel(us_1, us_2) \times delka(us_1) \times delka(us_2)}{delka(us_1) + delka(us_2)} \quad (3.14)$$

Kde $uhel()$ je úhel mezi úsečkami ve společném bodu, $delka()$ je normalizovaná délka vůči celé délce tvaru a us_1, us_2 jsou úsečky, které nahrazují spojnicí okolních bodů.

Tedy ze všech možných bodů postupně odebíráme ty, které mi tvar změní minimálně. Tento postup zastavím ve chvíli když by rozdíl podobnosti zjednodušeného a originálního tvaru překročil určitou mez.

Tento postup je bohužel omezen jen na tvary, které jsou uzavřené a nemají průsečíky samy se sebou. K určení podobnosti je tvar rozdělen na konvexní oblouky, mezi kterými se algoritmus snaží najít co nejlepší korespondenci zachovávající pořadí oblouků. Postup bere v potaz, že oblouk v prvním tvaru nemusí odpovídat jenom jednomu z druhého a používá přiřazování oblouk-skupina, skupina-oblouk a oblouk-oblouk (oblouky jsou spojovány do větších). Po nalezení korespondence se vzdálenost křivek určí jako součet vzdáleností spárovaných oblouků. Vzdálenost oblouků určíme pomocí vzorce:

$$S_a(o_1, o_2) = \left(\int_0^1 (T(o_1)(s) - T(o_2)(s) + \phi_0)^2 ds \right) \max(l(o_1), l(o_2)) \max\left(\frac{l(o_1)}{l(o_2)}, \frac{l(o_2)}{l(o_1)}\right) \quad (3.15)$$

kde obl_1, obl_2 jsou oblouky, $l()$ je relativní délka oblouku vztažená k celé křivce, $T(o)(p)$ je *tangentová funkce* oblouku o s pozicí $p \in [0, 1]$. *Tangentová funkce* je kroková funkce popisující úhel svíraný obloukem v daném místě s vektorem $(0, 1)$. Pro podrobnější popis algoritmu a přesnou definici *tangentové funkce* doporučujeme nahlédnout do práce [74].

3.3.3 Tvarový kontext

Kdybychom dokázali určit vztah bodu k celému zbytku tvaru, mohli bychom porovnávat přímo bod s bodem a lépe tak určit správné párování. Řešení nabízí článek [6], popisuje takzvaný *tvarový kontext* postup, který k porovnávání používá mimo jiné popis vztahu bodu k ostatním v kontuře.

Tento popis je realizován za předpokladu, že je kontura po částech hladká a že jí reprezentuje n bodů. Pro každý bod se spočítá log-polární histogram, histogram pozice ostatních bodů. Jedná se o 2D histogram kde je každá přihrádka určená směrem kterým leží v ní obsažené body vůči zkoumanému a logaritmem vzdálenosti od něj (logaritmus umožňuje vyšší rozlišení blízkého okolí). Deskriptorem tvaru je tedy uspořádaná posloupnost vektorů (histogramů).

K porovnávání tvarů je nejdříve třeba určit odpovídající body. K vyjádření ceny spárování bodů p_i, p_j se používá statistických metod, a to X^2 (Chi kvadrát) testu:

$$C(p_i, p_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (3.16)$$

Do této ceny je též dobré zahrnout i případnou podobnost v ostatních lokálních příznacích (textura, barevnost).

K určení podobnosti tvaru se nejdřív spárují dvojice nejvíce podobných bodu (k minimalizaci cen párování dvojic přes všechny permutace se používá algoritmus popsáný v publikaci [66]). Pro spárované body můžeme určit transformaci, pomocí Thin Plate Spline modelu [30], kterou na sebe body převedeme.

Vzdálenost je určena jako vážený součet cen transformace nejlépe odpovídajících bodů, tedy:

$$D(P, Q) = \frac{1}{n} \sum_{p \in P} \arg \min_{q \in Q} C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} \arg \min_{p \in P} C(p, T(q)) \quad (3.17)$$

Kde P, Q jsou množiny bodů a $T(p)$ je bod p po aproximované transformaci.

V práci jsou rovněž obsaženy návrhy na zlepšení invariance v rámci měřítka (normalizace přes průměr vzdálenosti mezi všemi dvojicemi ve tvaru) a rotace (při počítání orientace se bere tečna jako osa x (resp vektor (1,0))).

3.3.4 Fourierovy deskriptory

Metodou s mnoha variantami a variacemi jsou *Fourierovy deskriptory*. Jedná se o zpracování tvaru pomocí diskrétní Fourierovy transformace (dále DFT) a následná reprezentace pomocí nějak zpracovaných koeficientů.

Obecný popis takové metody vypadá přibližně takto

1. Aplikujeme DFT na nějakou reprezentaci tvaru. Často využívané jsou například komplexní souřadnice bodů kontury nebo funkce vzdálenosti bodu od těžiště kontury.

2. Normalizujeme tvar ve frekvenční oblasti (kvůli invarianci).
3. Vybereme omezený počet koeficientu ve frekvenční oblasti jako popis.

Pěkné srovnání některých použitých metod popisuje například publikace [68].

Fourierovy koeficienty a time warping vzdálenost Pokračováním myšlenky Fourierových koeficientů je postup popsáný Bartolinim [5]. Oproti Fourierovým koeficientům nezahazuje informace obsažené ve fázovém spektru (ta bývá zahozena kvůli invarianci k otáčení a nezávislosti na startovním bodu). Autor též popisuje důvody proč se ve fázi může skrývat kritické množství informace.

Postup algoritmu je následující:

1. Namapujeme konturu jako periodickou komplexní funkci – $z_i = x_i + iy_i$.
2. Na této funkci provedeme DFT.
3. Zajistíme invarianci (posun do počátku, pootočení, posunutí startovního bodu, měřítko)
4. Vybereme jako popis jen M (řádově 32) nízkofrekvenčních koeficientů.
5. Nad těmito koeficienty provedeme inverzní DFT.
6. Určíme vzdálenost pomocí *Dynamic time warping* vzdálenosti (dále DTW).

DTW vzdálenost se namísto euklidovské používá kvůli lepšímu zhodnocení fázového posunu. Podle autora použití DTW vzdálenosti k porovnání deskriptorů přináší lepší přesnost (+ 30% precision, 20 % recall (viz kapitola o porovnávání CBIR systémů) ve srovnání s klasickými Fourierovy deskriptory viz).

DTW vzdálenost je definována následovně:

$$d_{DTW}(\emptyset, \emptyset) = 0 \quad (3.18)$$

$$d_{DTW}(seq_1, \emptyset) = d_{DTW}(\emptyset, seq_2) = \infty \quad (3.19)$$

$$d_{DTW}(seq_1, seq_2) = \sqrt{vzd(prv(seq_1), prv(seq_2)) + \min \begin{cases} d_{DTW}(seq_1, zb(seq_2)) \\ d_{DTW}(zb(seq_1), seq_2) \\ d_{DTW}(zb(seq_1), zb(seq_2)) \end{cases}} \quad (3.20)$$

kde d_{DTW} je vzdálenost mezi dvěma sekvencemi bodů, $prv()$ vybírá první prvek sekvence, $zb()$ všechno krom prvního a $vzd()$ je libovolná funkce k určení vzdálenosti mezi dvěma body (v tomto případě 2. mocnina euklidovské). Kvůli výpočetní efektivitě je omezeno maximální odchýlení od diagonály (tedy rekurze kdy se použije jedna sekvence celá).

3.3.5 Momentové invarianty

Asi nejfousatější metoda dodnes používaná k popisu tvarů jsou momentové invarianty. Jde o globální popis pomocí momentů respektive jejich invariantních úprav. Metoda je zmíněna například v práci [7].

Momenty můžeme dělit na:

základní

$$m_{pq} = \int \int x^p y^q f(x, y) dx dy \quad (3.21)$$

kde $p, q \geq 0$

centrální

$$\mu_{p,q} = \int \int (x - x_t)^p (y - y_t)^q f(x, y) dx dy \quad (3.22)$$

kde $p, q \geq 0$, a x_t, y_t jsou souřadnice těžiště

Součet $p + q$ nazýváme řádem momentu.

Jak tedy z momentů vytvořit invarianty. Centrální momenty jsou samy o sobě invariantní vůči posunu. Invarianci k měřítku získáme například pomocí vydělení mocninou momentu nultého řádu. Dalšími výpočty lze získat invarianci k jiným transformacím. V roce 1962 Hu [53] uvedl jeho 7 rotačních invariantů (později se zjistilo, že jsou redundantní a některé závislé mezi sebou), pak se postupně objevila další rozšíření (například článek [39]).

3.3.6 Statistiky úhlů paprsků (Beam Angle Statistics)

Invariantní k posunu, otočení, měřítku křivek, necitlivý na šum a překrývání, ale omezený na uzavřené křivky – takový je deskriptor popisovaný v práci [4]. Je založený na paprscích („beams“) vedoucích mezi konturovými body a na statistickém popisu úhlů mezi nimi. Autor v práci ukazuje, že tento postup předčí všechny MPEG7 deskriptory

Mějme hranici definovanou jako spojenou posloupnost bodů $p(i)$:

$$B = \{p(1), \dots, p(N)\} \quad (3.23)$$

$$p(i) = (x(i), y(i)) \quad i = 1, \dots, N \quad (3.24)$$

kde N je počet bodů a $p(i) = p(i + N)$.

Pak pro $p(i)$ definujeme paprsky („beams“) jako:

$$L(p(i)) = \{V_{i+j}, V_{i-j}\} \quad (3.25)$$

kde V_{i+j} a V_{i-j} jsou vektory směřující k bodům $p(i + j)$ a $p(i - j)$.

Definujme K -okolí:

$$p(i \pm K) \in \eta_K(p(i)) \quad \forall p(i), i = 1, \dots, N, K = 1, \dots, N/2 \quad (3.26)$$

nyní spočítáme sklon každého paprsku V_{i+k} :

$$\theta_{V_{i+l}} = \tan^{-1} \frac{\Delta y_{i+l}}{\Delta x_{i+l}}, \quad l = \pm k \quad (3.27)$$

z těchto sklonů pak pro k -okolí bodu $p(i)$ určíme úhel:

$$C_K(i) = (\theta_{V_{i-k}} - \theta_{V_{i+k}}) \quad (3.28)$$

Tento úhel $C_K(i)$ lze brát jako náhodnou veličinu s funkcí hustoty $P_K(C_K(i))$ a náskres $C_K(i)$ vůči i pro každé K jako výstup stochastického procesu, který generuje tvar v různých měřítkách.

Jako deskriptor se pak používá m . moment náhodné veličiny $C_K(i)$ definovaný jako:

$$E[C^m(i)] = \sum_K C_K^m P_K(C_K(i)) \quad m = 0, 1, 2, 3, \dots \quad (3.29)$$

kde E je střední hodnota, $P_K(C_K(i))$ funkce hustoty $C_K(i)$. (maximální hodnota K je $N/2$ (N = počet bodů), $C_K(i)$ jde k 0, současně s tím jak se K blíží $N/2$)

V článku [4] aproximují $P_K(C_K(i))$ pomocí histogramu $C_K(i)$ v každém bodě $p(i)$.

Každý bod kontury je pak reprezentován jako vektor momentů paprskových úhlů:

$$\Gamma(i) = [E[C^1(i)], E[C^2(i)], \dots] \quad (3.30)$$

Celá kontura je pak reprezentována vykreslením $C_K(i)$ pro všechny hraniční body. V práci [4] ukazují, že 1. a 2. moment postačují k popisu tvaru. Deskriptorem je pak vektor obsahující vektory popisů bodů.

Porovnávání Porovnávání se provádí pomocí algoritmu nejbližší optimální podposloupnosti (Optimal correspondence subsequence [143]). Minimalizuje vzdálenost mezi dvěma vektory pomocí deformací.

Postup:

- Převědeme vektory na stejnou délku pomocí samplování BAS funkce.
- Najdeme vhodnou korespondenci bodů.
- Spočítáme vzdálenost jako součet 2. mocnin euklidovské vzdálenosti v každém bodě (bude minimální v rámci všech permutací pokud jsme našli správnou korespondenci).

Tento algoritmus je označován jako „Optimal correspondence algorithm“:

Optimal correspondence algorithm

$$\begin{aligned} D(0, 0) &= 0 \\ D(i, 0) &= i * t \quad [i = 1 \dots N] \\ D(j, 0) &= j * t \quad [j = 1 \dots M] \end{aligned}$$

```

for  $i = 1$  to  $N$  do
  for  $i = 1$  to  $M$  do
     $D(i, j) = \min\{$ 
       $[D(i - 1, j - 1) + \text{dist}(\Gamma(i), \Pi(j))],$ 
       $[D(i - 1, j) + t],$ 
       $[D(i, j - 1) + t]$ 
     $\}$ 
  end for
end for
return  $D(N, M)$ 

```

kde t beru jako vzdálenost mezi danými dvěma body, ne jako konstantu.
Složitost tohoto algoritmu je kvadratická.

3.3.7 Tensor Scale Descriptor

K porovnání tvaru lze též použít velmi lokálního příznaku známé jako „*tensor scale*“ (viz Saha [115]). Jde příznak každého pixelu který popisuje homogenní oblast obklopující zkoumaný pixel pomocí parametrického vyjádření elipsy. K popisu tvaru obsažených v obrázku pak nepotřebují segmentaci.

Elipsa v bodě p je definovaná pomocí:

- $\text{Orientace}(p) = \text{úhel mezi } t_1(p) \text{ a vodorovnou poloosou}$
- $\text{Anisotropie}(p) = \sqrt{1 - \frac{|t_2(p)|^2}{|t_1(p)|^2}}$
- $\text{Tloušťka}(p) = |t_2(p)|$

kde $|t_1(p)|$ a $|t_2(p)|$, $|t_1(p)| \geq |t_2(p)|$, označují délky poloos elipsy se středem v bodě p .

Abychom pro každý bod našli elipsu, musíme jím proložit přímky pod úhlem od 0 do 179 stupňů od vodorovné osy. Pak postupujeme po přímkách a zaznamenáváme si průsečíky hranic s přímkami. Po průchodu všech přímek posuneme vzdálenější body na přímkách tak, aby byly dvojice na přímce středově souměrné. Nakonec použijeme PCA (Principal component analysis [67]) k určení nejlepší elipsy. Toto provedeme s každým pixelem v obrázku.

Rozšířením tohoto postupu se zabývalo mnoho prací. Například v publikaci [96] autoři zrychlují vyhledávací část tím, že se přímo hledá bližší bod (zařídíme postup po přímce ve chvíli, kdy jsme našli první průsečík), používají dvouprahové hledání okrajů segmentu. K nalezení nepoužívají PCA ale vlastního, efektivnějšího algoritmu. Autoři též navrhnou deskriptor pro šedotónové obrázky – histogram lokálních orientací *Tensor Scale Descriptor* – podobnost se pro něj počítá jako rozdíl histogramů po korekci posunutí (resp. otočení – posunutím histogramu otočím orientaci).

V publikaci [3] autor uvádí deskriptor *Tensor Scale Descriptor with Influence Zones*, přináší další zrychlení výpočtu tensor scale pomocí využití image foresting transformace (viz článek [32], či [31]).

3.3.8 MPEG 7

Standard MPEG 7 (viz publikace [10], [91] a [98]) přinesl 3 deskriptory k popisu tvarů. *3D tvarový deskriptor*, ten zde popisovat nebudeme, zvědavého čtenáře odkazují na citované publikace. Deskriptor k popisu regionů – *Zernikovy momenty* a deskriptor určený k popisu kontur *Curvature scale space*.

Zernikovy momenty

Zernikovy momenty jsou pokračování v myšlence momentových invariantů. Jsou to projekce obrazové funkce na komplexní polynomy, které tvoří úplnou ortogonální množinu vyplňující vnitřek jednotkového kruhu (bázi). Popsány jsou například v článku [71].

Rotační invarianci získáme, když místo momentu použijeme jeho absolutní hodnotu (tedy zahodíme fázi - to se nám ale nemusí vyplatit – viz práce [5]). Invarianci k posunu a měřítku musíme zajistit normalizací. Pro obecný popis (deskriptor) je obtížné určit počet momentů, který nám popíše jakýkoli tvar dostatečně.

Curvature scale space

Popisem kontury, který by měl být odolný vůči šumu a invariantní k afinním transformacím (viz Mokhtarian Abbasi [98]) je *Curvature scale space*. Jde o popis bodů s nulovým zakřivením přes několik měřítek. Tvar se postupně zjednodušuje konvolucí s gaussovskou funkcí a během toho se sledují jeho vlastnosti.

Používáme parametrické vyjádření kontury a definici zakřivení, popsanou v 3.3.1. Pak se používá zjednodušení křivky pomocí vyhlazení gaussovskou funkcí. Pokud $g(u, \sigma)$ je 1-D jádro gaussovské funkce se šířkou (std. odchylkou) σ , pak $X(u, \sigma)$ a $Y(u, \sigma)$ odpovídají komponentám zjednodušené křivky:

$$X(u, \sigma) = x(u) \times g(u, \sigma), Y(u, \sigma) = y(u) \times g(u, \sigma) \quad (3.31)$$

Vzhledem k vlastnostem konvoluce, lze derivace jednotlivých komponent popsat:

$$X_u(u, \sigma) = x(u) \times g_u(u, \sigma), X_{uu}(u, \sigma) = x(u) \times g_{uu}(u, \sigma) \quad (3.32)$$

kde $g_u(u, \sigma)$ a $g_{uu}(u, \sigma)$ jsou první a druhá derivace gaussovské funkce odpovídajících parametrů dle u .

Pak zakřivení zjednodušené křivky odpovídá:

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u^2(u, \sigma) + Y_u^2(u, \sigma))^{3/2}} \quad (3.33)$$

Čím větší je σ tím více je tvar vyhlazený.

Postupně zvětšujeme σ až získáme konvexní křivky bez bodů nulového zakřivení a zaznamenáváme parametrické souřadnice bodů kde zakřivení prochází nulou (viz odkaz [42]). Takto získáme 2D obrázek (*CSS image*) kde svislou osou je σ a vodorovnou parametrické vyjádření umístění bodů s nulovým zakřivením (nulové body mi vykreslí oblouky trvanlivosti).

K srovnání tvaru:

- Kruhově posuneme *CSS image* tak aby maxima (přes σ) odpovídala stejné hodnotě u .
- Srovnáváme od maxima největších po dvojicích.
- Cena každé dvojice je dána jejich euklidovskou vzdáleností (tu přičítám k celkové ceně).
- Pokud není stejný počet maxim přičítám σ zbylých maxim k výsledku.

3.4 Prostorové vztahy

V předešlých kapitolách jsme popsali metody k hodnocení regionů, barevnosti, textur, tvarů. V obrázku nám ale zůstávají informace dalších typů. Jedním z nich je rozmístění segmentů ve scéně. Tato informace je velmi často popisována pomocí technik z rodiny „iconic indexing“ (symbolické dělení), existují však i jiné postupy.

3.4.1 Iconic indexing

Rodina metod založena na jednoduché myšlence, objekty v obrázku označíme za symboly (viz Chang [54] a [55]) a pak popíšeme uspořádání těchto symbolů (například pomocí 2D stringu či jiné struktury).

2D String

2D string [54] je popis uspořádání objektů v obrázku pomocí 2 řetězců, z nichž každý popisuje uspořádání podle jedné z os (respektive promítnutí uspořádání na tuto osu).

Popíšeme si ho na malém příkladu. Mějme obrázek obsahující objekty slunce, domeček a strom v následujícím uspořádání:

slunce
domeček strom

Takový obrázek popíšeme následujícím 2D Stringem: (domeček slunce>strom , domeček strom>slunce). > je oddělovač „sloupečků“ a uvnitř sloupce jsou symboly řazené podle pořadí od osy (v některých pracích opačně) a odděleny mezerou.

Pro 2D String byla popsána spousta rozšíření a variant. Zmíňme například rozšíření o znaky = a : pro označení zdali se objekty překrývají/nepřekrývají.

2D Be-String Jedním z posledních rozšíření myšlenky 2D Stringu je 2D Be-String (begin-end), popsáný v práci [144]. Symbol je zde reprezentován pomocí jeho nejmenšího opsaného obdélníku zarovnaného podle os. Typický 2D Be string vypadá takto:

$$(u, v) = (d_0x_1d_1x_2d_2\dots d_{n-1}x_nd_n, d_0y_1d_1y_2d_2\dots d_{n-1}y_nd_n) \quad (3.34)$$

kde d_n jsou bud takzvané *dummy objekty* nebo prázdné řetězce, a x_n, y_n symboly.

dummy objekt – umísťuje se v prípade že medzi okolnými symboly je prázdny prostor (v prípade d_0 medzi osou x a x_1)

prázdne řetězce – opačný případ

Pořadí ve sloupcích je opět ve směru od osy.

Podobnost se počítá jako vážený součet podobností přes osy. Kde podobnost přes osu odpovídá nule v případě prázdneho stringu (krom dummy objektu) m . V opačném případě spočítáme podobnost takto:

$$d(Q, D) = 1 - \frac{(Q_l + D_l - 2L)}{(4N + 1)} \quad (3.35)$$

kde Q, D jsou stringy, Q_l jejich délky, L délka nejdelšího společného podřetězce (mezi stringy) přes danou osu a N značí počet objektů v dotazovém obrazu.

3.4.2 Binární vztahy mezi regiony

Další možnost jak popsat uspořádání objektů v obrázků je popisovat vzájemné vztahy mezi jednotlivými objekty (tedy objekt-objekt). Jedním z příkladů je například algoritmus popisující *Prostorový kontext* obrázku [112], který popisuje vztahy regionů pomocí 6 typů umístění: vlevo, vpravo, nahoře, dole, dotyk a před. Trochu složitější je postup nazývaný *Virtual Image*, který si popíšeme podrobněji.

Virtual image

V publikaci [106] autor rozebírá reprezentaci pomocí popisu prostorových vztahů mezi dvojicemi objektů separátně přes obě osy. (tedy vztahy přes x , vztahy přes y)

Vztahy jsou popsány následně:

$A < B$ – A je s B disjunktní ($konec(A) < zacatek(B)$),

$A = B$ – A se přesně kryje s B ($zacatek(A) = zacatek(B)$, $konec(A) = konec(B)$),

$A|B$ – A se dotýká hranou B ($konec(A) = zacatek(B)$),

$A\%B$ – nemají stejnou hranu a A obsahuje B ($zacatek(A) < zacatek(B)$, $konec(A) > konec(B)$),

$A(B$ – stejná počáteční hrana, A obsahuje B ($zacatek(A) = zacatek(B)$, $konec(A) > konec(B)$),

$A)B$ – stejná konečná hrana, A obsahuje B ($zacatek(A) < zacatek(B)$, $konec(A) = konec(B)$),

A/B – A částečně překrývá B ($zacatek(A) < zacatek(B) < konec(A) < konec(B)$).

Obrázek se pak reprezentuje pomocí:

seznamu objektů – $Ob = \{A, B, C, D\}$,

vztahů přes osu X – $Rel_x\{A < B, A \% C, A(D, B < A, \dots D < C)\}$,

vztahů přes osu Y – $Rel_y\{A < B, A < C, A < D, B < A, \dots D|C\}$.

(v případě že $A < B$ už nepíšeme $B < A$).

Při porovnávání podobnosti tohoto popisu se porovnávají nejen obsažené objekty, ale i relace mezi nimi (kombinace různých relací jsou různě ohodnocené). Pro podrobnější popis porovnávání doporučujeme pročíst práci [106].

3.4.3 Ostatní

Dalšími technikami popisujícími prostorové uspořádání objektů v obrázku jsou například *obalové obdélníky*. *Obalové obdélníky* popsané v práci [94] používají k zachycení uspořádání těžiště regionu a jeho minimálního obalového obdélníku. Zajímavý postup k ohodnocení rozvržení nabízí též *vážené procházky* [9]

3.5 Jiné

Myslet si, že můžeme popsat všechny možné metody a postupy užívané k porovnávání obrázků by bylo bláhové.

Jsou zde například metody založené na různých lokálních invariantech, rohových bodech, bodech zájmu a dalších technikách používaných například ve stereo-matchingu. Za všechny zmiňme alespoň některé. Příznaky invariantními k měřítku, afinním transformacím a dokonce různému osvětlení se zabývá publikace [95]. Metodu detekce význačných bodů založenou na waveletech popisuje práce [128]. Využití *Earth movers vzdálenosti* [113] k porovnávání lokálně invariantních příznaků zkoumá práce [41].

Pro reprezentace obrázku pomocí rozložení lokálních deskriptorů/invariantů se objevují na jedné straně postupy, kde jsou deskriptory úplně nezávislé (pytel příznaků kde každý reprezentuje část regionu), po naprosto propojené (constelation model [35]). Při vysoké provázanosti deskriptorů se ale musí omezit počet částí, protože s ním exponenciálně roste složitost algoritmu. Kompromis nám například přináší *topologie hvězdy* [36], nebo *hierarchie*, kde nejnižší úroveň odpovídá lokálním příznakům [11] či *geometrická struktura* v níž jsou lokální příznaky prostorově závislé na jejich nejbližších sousedech [14].

Výhody a nevýhody bodů zájmu pro barevné obrázky diskutují Gouet a Boujemaa [40]. Výkonnostní srovnání různých detektorů invariantních bodů zájmu najdeme v práci [95].

Z jiného soudku je celkový popis obrázku jen pomocí Daubechies wavelety systém WBIIS [141].

Doslova úplně jiná kapitola jsou metody využívající k porovnání obrázku například nějakého sémantického uchopení obrázku.

Kapitola 4

Typy segmentací

Správná a přesná segmentace je důležitá pro zachycení tvarové informace v obrázku a pro jeho správné rozdělení na regiony. V této kapitole si popíšeme nejpoužívanější postupy a zmíníme některé další.

4.1 Shlukování o K -průměrech

Postup shlukování o K -průměrech je jednoduchý:

1. Máme předem stanovený počet shluků K .
2. Nějakým (náhodně, rovnoměrně, závisí na algoritmu) způsobem vybereme K bodů v prostoru popisů.
3. Určíme pro každý popis (vektor, pixel) bod, ke kterému má nejbliže.
4. Pro každou skupinu příslušnosti spočítáme průměr, kterým nahradíme původní bod.
5. Iterujeme zpět na 3 pokud nějaké body změnily příslušnost.

Tento postup využívají například Wang et al. [140]. Dělí obrázek na bloky 4×4 , nad nimiž si počítá barevné a texturové příznaky, pak segmentuje obrázek K -průměrovým algoritmem.

Dalším vylepšením je KMCC (*K-means with connectivity constraint*) [94]. Bere v potaz blízkost objektů pomocí definice nového centra pro K -průměrový algoritmus. Hodí se na nehomogenní objekty.

4.2 Blob world

Další algoritmus používá EM (Expectation-Maximization) algoritmus k nalezení parametrů modelu rozdělení pomocí směsi gaussovských funkcí (popisují Carson et al. [16], používají Feng a Chua [34]).

1. Zjistí vlastnosti (barva/textura/pozice) pro každý pixel, odhadni zhruba parametry modelu.
2. Použij iterativní EM algoritmus k nalezení přesnějšího odhadu parametrů.

4.3 JSEG segmentace

Deng a Manjunath [25] popisují algoritmus zvaný „JSEG“ segmentace (použitý použitý v článcích – [34], [81]). Tento postup obchází nutnost odhadnout parametry texturového modelu. Produkuje regiony homogenní texturou i barvami, tedy jde o dobrý postup k segmentaci přírodních scén.

1. Kvantizujeme barvy obrázku do několika tříd, záměnou barvy pixelu za jeho třídu získáme mapu tříd (class-map) obrázku.
2. Provedeme segmentaci nad mapou tříd.

4.4 Ostatní

Vedle algoritmu Blobworld, dalším postupem který využívá technik statistického modelování je mean shift uvedený v publikaci [19].

Algoritmus popsáný v článku [140] pracuje s waveletovými koeficienty vysokých frekvencí, nespojuje ale okraje, detekuje objekt na který bylo ostřeno a pak rozlišuje popředí a pozadí. Rekurzivně reklasifikuje bloky na popředí a objekt zájmu postupně podle pásma.

Postup který nejdříve segmentuje podle texturových příznaků a pak upravuje výsledek podle barevných je popsáný Chen et al. [57].

Segmentaci pouze homogenních barevných regionů, pomocí použití přímého shlukování v barevném prostoru popisují Comaniciu a Meer [20].

Zjednodušování křivek k segmentaci používá postup [33]. Různé statistické metody typu Markovových řetězců, metody Monte carlo a dalších popisují Tu a Zhu [132].

Segmentaci jako dělení grafu (vrcholy jsou pixely a hranové ohodnocení je dáno podobností vrcholů) pojímá publikace [116]. Malik et al. [87] rozšiřují tento postup na segmentaci texturovaných obrázku užitím příznaků rozdílů textury a kontury.

Po redukcí závislosti porovnávacích metod na přesné segmentaci volají Carson et al. [16]. Tento problém částečně řeší Wang et al. [140] pomocí objektu zájmu a klasifikace jen na 2 skupiny, dále práce [78] a [76] pomocí charakterizace prostorového rozložení barvy a textury pomocí statistických metod.

Kapitola 5

Sumarizace a adaptivita signatur

U některých postupů můžeme získaný popis použít již jako signaturu, u jiných jde například o rozdělení, které musíme nějak popsat či sumarizovat. V rozdělení sumarizací se budeme volně inspirovat dělením z [23].

5.1 Sumarizace popisů do signatur

5.1.1 Histogram

Histogram je dříve nejčastější reprezentace rozdělení. Máme popis jako příznakové vektory ve tvaru $x_{i,j} \in R^d$ (d -rozměrný euklidovský prostor). Rozdělím prostor R^d do k přihrádek a spočítám procentuální zastoupení spadajících $x_{i,j}$ pro každou přihrádku. Histogram je pak k -rozměrný vektor (f_1, f_2, \dots, f_k) , kde f_l je frekvence l . přihrádky.

Rozšíření této myšlenky přináší například práce [44] a jeho histogram s více rozlišeními (multi-resolution). Spolu s vlastním porovnávacím algoritmem se ukazuje efektním na texturované obrázky. Více rozlišení histogramu nejen, že zachovává jednoduchoost a efektivitu, navíc zachycuje prostorové variace v obrázku.

Článek [62] popisuje kvantizaci pomocí směsi gaussovských funkcí což se ukazuje jako lepší způsob získání histogramu oproti klasické uniformní kvantizaci a vektorové kvantizaci s kvadratickou chybou.

5.1.2 Lokalizovaný histogram

Klasický histogram zanedbává umístění (respektive průměrné hodnoty) přihrádek. Tuto nevýhodu rozebírá práce [114]. Pro měření „vzdálenosti“ rozdělení je umístění kyblíků rozhodující. Článek [114] tedy navrhuje použití *Earth Movers vzdálenosti*. Při použití EMD je histogram reprezentován jako množina dvojic příznakový vektor, frekvence. Prakticky jde o $\{(z_1, f_1), (z_2, f_2), \dots, (z_k, f_k)\}$, kde $z_l \in R^d$ je centrum přihrádky.

Práce [75] dokazuje že EMD použité na frekvence je ekvivalentní Mallowsově vzdálenosti [88], což je pravá metrika pro obecné pravděpodobnostní vzdálenosti.

Histogram je speciální rozdělení ve smyslu konečného/spočetného počtu příhrádek, který je navíc stejný pro různé obrázky.

5.1.3 Shluková analýza

Proč bychom museli používat pevná z_l ? Proč nenechat volbu k na vlastnostech obrázku? Ano, necháme – použijeme shlukování. Použití shlukování k sumarizaci je popsáno například v publikacích [26, 140].

- Mějme data $\{x_{i,j}, i \geq 1, j \geq 1\}$, aplikujeme shlukovací proceduru: například *K-průměrové shlukování*.
- Máme k shluků – pak průměr všech $x'_{i,j}$ v l . shluku označíme jako z'_l .
- Procentuální zastoupení $x'_{i,j}$ v l -tém shluku označíme f'_l .
- Množina pixelů (i, j) , pro která jsou $x'_{i,j}$ ve stejném shluku tvoří relativně homogenní oblast, protože shlukování vybírá nejbližší hodnoty $x'_{i,j}$.

Signatura je pak stejná jako u lokalizovaného histogramu. Kvůli výběru relativně homogenních oblastí se při zahrnutí umístění pixelu v obrázku do kritéria pro shlukování používá tento postup pro segmentaci, je to také důvod proč se tato signatura považuje za „region based“.

Taková signatura je více kompaktní než histogramy, protože umožňuje aby se z_l adaptovalo podle obrázku. Publikace [26] i [140] tvrdí že region-based signatura je výpočetně efektivnější pro vyhledávání, zároveň obchází problémy, které jsou spojeny s dřívějšími postupy jako redukce dimenze a deskriptor pomocí barevných momentů.

Článek [140] říká že motivací pro shlukovou analýzou není jen dynamický histogram, ale spíš to že barevně/texturově homogenní region odpovídá objektu v obrázku, takže navíc k z'_l můžu přidat další informace příslušející celému regionu.

5.1.4 Ostatní

Dalšími způsoby sumarizace lokálních vlastností je například pro *vektory lokálních příznaků* postup *spojité funkce hustoty* ([28]) což je přesnější než diskrétní popis s konečným počtem vektorů.

Jinou technikou je *stochastické modelování*, které navíc oproti spojité funkci hustoty bere v potaz prostorovou závislost vektorů. Popisuje ho například článek [77], užívá ho k porovnávání čínských inkoustových maleb. Nevýhoda sofistikovanějších modelů je jejich výpočetní náročnost. Též je nepříjemné, že ve výsledku jsou méně intuitivní než jiné reprezentace a tak je jejich využití omezeno.

5.2 Rozdělení signatur podle jejich adaptivity

Signatury lze též dělit dle jejich přizpůsobení vlastnostem obrázku či potřebám uživatele. Průzkum a porovnání výkonu (v rámci výběru příznaků) popisuje publikace [43].

statické – Sem spadají signatury generované vždy stejným způsobem, nehledě na vlastnosti obrázku a potřeby uživatele.

adaptivní podle obrazu – Toto jsou signatury jejichž vlastnosti se liší podle vlastností obrázku (pro různý obsah potřebujeme různé vyjadřovací prostředky). Typickým příkladem je systém *Simplicity* [140]. Dělí obrázky podle kategorie (ta je nejdříve určena automaticky – grafika/fotografie, texturovaný/netexturovaný). Pro každou kategorii se sledují jiné příznaky. Je nutno zajistit, aby robustnost získaná různými signaturami neovlivnila efektivitu indexování a vyhledávání. Při velkém počtu možných příznaků se používá výběr jejich podmnožin, nebo udílení vah podle důležitosti, zde jsou některé příklady:

- automatický výběr podmnožin pro strojové učení (popsané v [145]),
- výběr používající boosting [129],
- evoluční algoritmy [69],
- Bayesovská klasifikace [15],
- algoritmus, který vybírá příznaky tak, aby měly co nejmenší redundanci [97].

Adaptivní dle uživatele – jsou signatury, jejichž podoba se řídí potřebami uživatele. Našly široké uplatnění v medicínské sféře. Například publikace [100] popisuje způsob získání vah příznaků z anotací uživatelů. Používá se metod s vyšší úrovní sémantiky jako je například *relevance feedback* a další.

Kapitola 6

Popis měření podobnosti pomocí signatur

V minulých kapitolách jsme rozebírali jaké příznaky sledovat v obraze a jak z nich tvořit signatury. Už v kapitole o nízkourovňových příznacích jsme rozebírali otázku jaké vlastnosti od metod porovnávání obrázků požadujeme. Též jsme si na ni z části odpověděli. Položme si tuto otázku znovu, naposledy a odpovězme s přihlédnutím na vlastnosti, které bychom očekávali od metod porovnávání signatur.

Jaké vlastnosti by tedy měla mít míra, či metrika k porovnání podobnosti obrazu? Měla by být:

- odpovídající významu/sémantice,
- odolná vůči šumu,
- efektivně spočitatelná,
- nezávislá na pozadí/okolí,
- alespoň lokálně lineární (měla by platit trojúhelníková nerovnost na sousedech).

V této kapitole čerpáme z podrobného a rozsáhlého rozboru popsaného v práci [23]. Zájemce o podrobnější rozbor a další informace odkazují na tuto publikaci.

6.1 Rozdělení

Metody porovnávání signatur lze dělit podle jejich způsobu návrhu podle (na):

- reprezentace příznaků jako vektory / nevektorové reprezentace / soubory (celky),
- podobnosti založené na oblasti / celkové podobnosti / kombinace,
- měření podobnosti v lineárním prostoru / v nelineární varietě (lokálně euklidovský prostor),
- role, kterou hrají segmenty ve výpočtu podobnosti,

- stochastické, fuzzy nebo deterministické míry podobnosti,
- použití učení s učitelem nebo plně automatického (pro metody, které využívají strojového učení)

V článku [23] je velmi přehledné dělení popsané následovně:

Typ signatury	Matematická formulace	Vzdálenosti	Technika
vektor příznaků	jeden vektor	euklidovská geodetická	manifold embedding(varieta)
signatura založená na regionu	skupina/y vektorů	vážený součet vzdáleností vektorů výběr regionů	optimal matching a lineární programování uživatelský vstup optimalizace
souhrn lokálních příznakových vektorů	seznamy kódů pravděpodobnostní hustota	textové Kullback-Leibler	vektorová kvantizace, textové vyhledávání odhad hustoty

Jednoduché příznaky typu příznakového vektoru jsou efektivní pro výpočet, lze je škálovat. Většinou však nemusí být dostatečně efektivní/ diskriminabilní při popisu/vyhledávání. V průběhu let je viditelný posun od globálních vektorů, k signaturám založeným na oblasti, případně dále k sémantice či porovnávání rozdělení, také v případě globálních popisů od lineárních metrik ke geodetickým.

6.2 Samotný vektor příznaků

Pokud je obrázek interpretován jako jeden samostatný vektor, je obtížné měřit vizuální podobnost v jakémkoli lineárním prostoru, a proto se nahrazuje typicky euklidovská míra geodetickou. Předpoklad je, že vizuální podobnost lépe koresponduje s nelineárním podprostorem, než s originálním prostorem.

6.2.1 Lineární prostor

Nejčastěji používané metriky v lineárním prostoru jsou varianty metrik Minkowského typu, a to L_1 metrika známá jako *manhattanská* a L_2 známá jako *euklidovská*.

Metriky Minkowského-typu

Mějme dva p -rozměrné vektory (x_1, x_2, \dots, x_p) a (y_1, y_2, \dots, y_p) .

Pak je obecná Minkowského metrika je definována následně:

$$d(X, Y) = \left(\sum_{i=1}^p |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (6.1)$$

Pro $r = 2$ jde o *euklidovskou* vzdálenost, pro $r = 1$ o *manhattanskou* vzdálenost. Existuje též vážená verze kde k zadání přiberu p -rozměrný vektor vah $w_i, i = 1, \dots, p$:

$$d(X, Y) = \left(\sum_{i=1}^p w_i |x_i - y_i|^r \right)^{\frac{1}{r}} \quad (6.2)$$

6.2.2 Nelineární varieta

Na nelineární varietu musíme aplikovat geodetickou míru (globálně to není metrika, ale lokálně ano). Tento postup je popsán publikacemi [48], [47], [46] a [149].

Článek [137] prezentuje „*Multiresolution manifold vzdálenost (MRMD)*“. Obrázky jsou reprezentovány jako variety, vzdálenost je pak ta, která minimalizuje chybu transformace jedné variety na druhou (vlastně měřím posunutí). Vzdálenost se pak úspěšně používá v zarovnávání obrázků (rozpoznávání tváří, detekce scén ve videu).

Existují též metody učení variet (vedou k nelineární redukci dimenze): Locally-linear Embedding, Isomap, Multi-dimensional scaling [27].

6.3 Podobnost signatur založených na regionu

Postupy pro porovnávání obrázků podle typu porovnání množin a vlastností obsažených regionů lze rozdělit do dvou skupin:

1. Jeden s jedním (někdy také *Hard matching*):
 - každý region se porovnává jen s jedním protějškem z druhého obrázku,
 - protějšek je vybírán pomocí nějakého kritéria nejlepšího párování (nejpodobnější apod..),
 - celková podobnost je vážený součet podobností mezi všemi dvojicemi, váhy většinou odpovídají velikosti regionu či jinému příznaku důležitosti
2. Množina na množinu (někdy také *Soft matching*):
 - každý region může být srovnáván s více protějšky,
 - typicky se zhodnotí všichni se všemi,
 - typickými reprezentanty jsou například EMD [113] – více 6.3.3 a IRM [78] – více 6.3.4,
 - počet regionů nemusí být stejně veliký,
 - umožňuje omezit chyby napáchané nepřesnou segmentací.

Mějme signaturu jen ve formě vážené posloupnosti příznakových vektorů:

$$I = ((z_1, p_1), (z_2, p_2), \dots, (z_n, p_n)) \quad (6.3)$$

Kde z_i jsou příznakové vektory a p_i jsou odpovídající váhy.

Pak označíme srovnávané signatury:

$$I_m = ((z_1^{(m)}, p_1^{(m)}), (z_2^{(m)}, p_2^{(m)}), \dots, (z_n^{(m)}, p_n^{(m)})); m = 1, 2 \quad (6.4)$$

Typický postup metody z prvního tábora by bylo nalezení korespondence dvojic regionů mezi obrázky, klasifikace jejich podobnosti a sumarizace těchto výsledků.

Jako příklad metody druhého tábora bereme postup použitý například v práci [140] (systém Simplicity).

Postup je následující:

- Přiřadíme váhu $s_{i,j}$ každé dvojici z_i^1 a $z_j^{(2)}$, $1 \leq i \leq n_1$, $1 \leq j \leq n_2$, která určuje důležitost podobnosti dané dvojice (tedy neporovnávají se jen dvojice ale každý s každým). Tímto by se z části měl zmírnit efekt špatné segmentace.
- Váhy jsou výsledkem omezení, nejčastěji se používá $\sum_i s_{i,j} = p_j^{(2)}$ a $\sum_j s_{i,j} = p_i^{(1)}$.
- Ve chvíli kdy máme zvolené váhy, vzdálenost spočítáme takto:

$$D(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{i,j} d(z_i^{(1)}, z_j^{(2)}) \quad (6.5)$$

kde vzdálenost $d(\cdot)$ může být definovaná libovolně v závislosti na systému.

6.3.1 Hausdorffova vzdálenost

- Při výpočtu Hausdorffovy vzdálenosti je každé $z_i^{(1)}$ porovnáváno se svým nejbližším kolegou z druhého vektoru, řekněme $z_{i'}^{(2)}$.
- Jde tedy o typ jeden na jednoho.
- Vzdálenost mezi vektory je maximum všech $d(z_i^{(1)}, z_{i'}^{(2)})$ symetrizovaná tak, že se spočítá jak vzdálenost I_1, I_2 tak I_2, I_1 a vybere se ta větší.

Shrnutí:

$$D_H(I_1, I_2) = \max(\max_i \min_j d(z_i^{(1)}, z_j^{(2)}), \max_j \min_i d(z_j^{(2)}, z_i^{(1)})) \quad (6.6)$$

Hausdorffova vzdálenost je použita v práci [70].

6.3.2 Mallowsova vzdálenost

Pokud bychom se zamysleli, jak dobře určit váhy $s_{i,j}$ pro dvojice $z_i^{(1)}, z_j^{(2)}$ v rovnici 6.3, mohlo by nás napadnout hledat $s_{i,j}$ tak, aby $D(I_1, I_2)$ byla minimalizovaná přes určitá omezení $s_{i,j}$.

Předpokládejme:

$$\sum_i p_i^{(1)} = 1; \sum_j p_j^{(2)} = 1 \quad (6.7)$$

(toto dokážeme pomocí normalizace (pokud nechci žádném z obou obrázků dát větší důležitost), v praxi p_i často odpovídají pravděpodobnostem a tím pádem je

předpoklad automaticky splněn) Pokud $p_i^{(1)}$ odpovídá důležitosti regionu $z_i^{(1)}$ a $\sum_j s_{i,j}$ zohledňuje celkový vliv $z_i^{(1)}$ na výpočet $D_H(I_1, I_2)$, pak je přirozené požadovat omezení $\sum_j s_{i,j} = p_i^{(1)}$ pro všechna i a podobně $\sum_i s_{i,j} = p_j^{(2)}$ pro všechna j . Nebylo by též špatné omezit $s_{i,j} \geq 0$ pro všechna i, j .

Z těchto požadavků nám vychází:

$$D(I_1, I_2) = \min_{s_{i,j}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{i,j} d(z_i^{(1)}, z_j^{(2)}) \quad (6.8)$$

Což odpovídá Mallowsově vzdálenosti pro diskrétní rozdělení [88].

6.3.3 Earth movers vzdálenost

Další metoda patřící do skupiny porovnávání množina na množinu je Earth movers vzdálenost (v překladu bagrová vzdálenost) [113], [114].

- Bere vzdálenost mezi obrázky jako nejmenší „práci“ kterou musím vykonat k tomu, abych z jednoho histogramu udělal druhý (respektive transformace s nejmenší cenou, která z jednoho rozdělení udělá druhé).
- Umožňuje matchovat různé délky reprezentací rozdělení.
- Je použita například v práci [64].
- Článek [75] dokazuje že EMD použité na frekvence (tedy p_i a p_j jsou pravděpodobnosti) je ekvivalentní Mallowsově vzdálenosti [88], což je pravá metrika pro obecné pravděpodobnostní vzdálenosti.
- pro přesný popis doporučujeme publikace: [113] či [114].

6.3.4 Integrated region matching (IRM) vzdálenost

Dalším postupem je IRM vzdálenost popsána v článku [78]. Používá princip *Most similar highes priority* (MSHP) – největší priorita pro nejpodobnější. $s_{i,j}$ mají stejná omezení jako v Mallowsově vzdálenosti, $D(I_1, I_2)$ se však nepočítá přes minimalizaci.

Místo toho MSHP princip říká:

- Vyber dvě oblasti v obrázku s nejmenší vzdáleností přes všechny dvojice
- Přiděl jim největší prioritu, tedy největší možná $s_{i,j}$.
- Prováděj toto rekurzivně dokud nejsou ohodnoceny všechny regiony (dvojice se nevrací).

Tedy mám $\sum_j s_{i,j} = p_i^{(1)}$ a $\sum_i s_{i,j} = p_j^{(2)}$ pro všechna i, j . Tento postup je rychlejší než výpočet Mallowsovy vzdálenosti, přičemž má stejné pokud ne lepší výsledky. Též (jak je popsáno v publikaci [142]) snižuje vliv nepřesné segmentace.

6.3.5 Vylepšení proti základní porovnávací myšlence

Popsané postupy jsou nejpoužívanější, avšak existuje mnoho dalších.

Například využití vektorové kvantizace pro tvorbu kódových knih je popsáno v článku [65].

Stromový postup používaný pro shlukování je popsán v publikaci [29]. Snažíme se o ještě větší zrychlení vyhledávání, chceme zkrátit vyhledávací cestu a zmenšit množinu kterou musíme prohledávat. Nejdřív podle nějakého jednoduchého příznaku (nejčastěji vektor) zařadím do kterého shluku a pak použiji silnější deskriptor k vyhledávání už jen uvnitř shluku

Úpravu IRM vzdálenosti zahrnující vícehodnotové (fuzzy) vyhledávání uvádí Chen a Wang [56].

Využití kontextových informací, boostingu, učení modelů a kategorizace (pro efektivní vyhledávání v přeplácaných obrázcích) je popsáno Amoresem et al. [1].

Vyhnout se segmentaci a ukázat, že taková metoda je efektivnější než metody založené na potenciálně nepřesné segmentaci se snaží publikace [49], popisuje okénkové procházení přes měřítko a místo.

Postup založený na hrubém ohraničení popředí/pozadí pomocí obdélníků a následném vyhledávání jen pomocí popředí přináší publikace [22].

Dobrý nápad je zapojit uživatele do výběru místa zájmu. Zabývá se jím například práce [70], která popisuje systém vyhledávající uživatelem nakreslený objekt.

Jiný způsob zapojení uživatele je dotaz pomocí určení oblasti získané segmentací jako vzoru. Tento způsob dotazování používá například systém Blobworld [16].

- Uživatel vybere několik z oblastí získaných segmentací v obrázku a určí důležitost pro typy příznaků.
- Oblastem se říká bloby (podle tvaru gaussovské funkce pomocí kterých jsou modelovány).
- Každý blob b_i je reprezentován pomocí vektoru příznaků (barev a textury) v_i .
- Pro dotazový blob b_i a jakýkoli b_j z databáze má ten nejpodobnější skóre:

$$\mu_i = \max_j \exp \left(\frac{(v_i - v_j)^T \Sigma (v_i - v_j)}{2} \right) \quad (6.9)$$

Kde Σ je matice vah pro příznaky získaná od uživatele.

- Pro sloučené dotazy se míra rozšiřuje pomocí fuzzy logiky.
- klady: přesnější zadání, lepší pochopení
- zápory: důraz na správné zadání, závislost na uživateli

Hledání obrázků obsahujících transformovaný či zmenšený/zvětšený objekt bez zapojení uživatele popisuje publikace [103].

6.4 Porovnávání lokálních příznaků

Pokud jsme již lokální příznaky nějak nesumarizovali do jiné formy, je třeba použít nějakou metodu která dokáže porovnat buď velké množství lokálních vektorů, nebo funkce popisující rozdělení hodnot popisujících lokální vlastnosti.

6.4.1 Vektorová kvantizace a generování seznamů kódů

Metoda inspirovaná kompresními algoritmy je vektorová kvantizace [151], kvantizujeme příznakové vektory na blocích obrázku, pak můžeme textově vyhledávat, použít míru textového vyhledávání.

6.4.2 Kullbackova-Leiblerova (K-L) vzdálenost

Pro porovnávání dvou rozdělení se používá Kullbackova-Leiblerova vzdálenost (publikace [28], [92]), jinak též relativní entropie. Hodí se na texturované obrázky, kde není tak důležitá segmentace, spíše rozdělení hodnot příznakových vektorů.

Je to asymetrická míra rozdílu mezi 2 rozděleními $f()$ a $g()$, definována:

$$K(f, g) = \int_{-\infty}^{+\infty} f(x) \log \frac{f(x)}{g(x)} dx \quad (6.10)$$

diskrétně:

$$K(f, g) = \sum_x f(x) \log \frac{f(x)}{g(x)} \quad (6.11)$$

6.4.3 Fraktálové kódování

Další výlet do krajin kompresních algoritmů je použití myšlenky fraktálové komprese [108].

- Rozdělím obrázek na bloky.
- Pro každý blok hledám afinní transformaci která by ho zobrazila z nějakého doménového.
- Pokud ji nenajdu, prohlásím blok za doménový.

Tuto metodu využíváme většinou pro texturové databáze. Porovnání se provádí jako porovnávání histogramů bloků.

6.4.4 Ostatní metody

Soubory lokálních příznakových vektorů a rozdělení nízkoúrovňových příznaků se hodnotí pomocí Waldova-Wolfowitzova testu [127].

Pravděpodobnostní nástroje k vyhledávání obrázků popisují publikace [63] a [136].

Publikace [136] se snaží začlenit výběr příznaků, reprezentace, a podobnostní míru do kombinovaného Bayesovského vzorce s cílem minimalizovat pravděpodobnost chyby.

V postupu popsaném článkem [60] jsou primitivní příznaky hierarchicky sdružovány a jejich vzájemné vztahy jsou použity k charakterizaci.

6.5 Ostatní způsoby měření

6.5.1 Anchoring (ukotvení)

V případě, že nemáme explicitní vektorovou reprezentaci stačí nám metrika k porovnání dvou obrázků. Lze použít *anchoring* [102], který vytvoří vektorovou reprezentaci tak, že každou dimenzi vztáhne k jednomu z předem daných obrázků (tzv. *vantage images*).

Zadání:

- metrika $d(F_i, F_j)$ mezi jakýmkoli párem obrázků
- množina K vybraných obrázků (*vantage images*)

Pak můžu pomocí tzv „*Vantage space*“ zobrazení $V : F \rightarrow \mathbb{R}^K$ mapovat jakýkoli obrázek F_i na jeho vektorovou reprezentaci $V(F_i)$, tímto způsobem:

$$V(F_i) = \langle d(F_i, A_1), \dots, d(F_i, A_K) \rangle . \quad (6.12)$$

Kapitola 7

Úvod do příznaků vyšší (sémantické) úrovně

Moderní postupy se stále více přichylují k hledání sémantiky v obrázku. Metody k dosažení tohoto cíle jsou popsány v publikaci [82], ze které v této kapitole ve velké míře čerpáme. Zvědavějšího čtenáře odkazují na tento článek, zde uvedeme jen lehký úvod.

7.1 Ontologie objektů

Ontologie objektů je způsob k definování pojmů vyšší úrovně pomocí obvyklé lidské slovní zásoby. Nejdříve definujeme nízké sémantiky (nejen pro regiony), jako pojmenování intervalů low-level příznaků (například nahoře/dole/uprostřed, světle zelená/tmavě zelená, velký/malý/střední, středně/více protáhlý). Těmto příznakům se říká *ontologie objektů*. Jde vlastně o takové slovníčky.

Tohoto slovníčku se pak použije k vytvoření vyšší sémantiky (dalšího slovníčku).

Například oblohu můžeme popsat jako objekt, o kterém se dá říct že je nahoře, modrý a homogenní. Typický příklad systému popisují Mezaris et al. [94], používají kvantizace barvy v Lab prostoru, pozice, velikosti a protáhlosti tvaru.

Další techniky používané ke generování ontologií jsou například kvantizace barevných a texturových příznaků, pojmenovávání barev (10–20 odstínů). Tyto techniky jsou popsány a porovnány v například Conwayem [21], pojmenovávání textur řeší Rao a Lohse [111]

Tato metoda umožňuje jednoduchý návrh systému a lze jí s úspěchem užít na aplikace s jednoduššími sémantickými vlastnostmi.

7.2 Strojové učení

Ke zvládnutí komplexnější sémantiky lze s úspěchem použít strojového učení k asociování low-level příznaků s požadavky.

7.2.1 Učení s učitelem (Supervised) učení

Metody založené na učení s učitelem se snaží předpovědět/odhadnout hodnotu výsledné míry (na základě toho jak byla oklasifikovaná trénovací množina) pro daný objekt.

Pro učení s učitelem se používá metod jako *support vector machine* (SVM) př. publikace [34], *bayesovské klasifikátory*, *neuronové sítě*, *rozhodovací stromy*.

Mají několik nevýhod:

- potřebují velký oklasifikovaný vzorek,
- při změně domény je třeba je dotrénovat.

Tyto problémy se v určité míře snaží řešit *bootstrapping* postup (popisuje například publikace [34]), což je trénování více různých klasifikátorů na malé množině. Výsledek se rozpoznává/klasifikuje všemi najednou.

7.2.2 Učení bez učitele (Unsupervised)

Učení bez učitele nedokáže přiřadit žádnou výslednou míru. Ve většině algoritmů jde spíš o popis organizace dat či shlukování. Typickými algoritmy jsou K -průměrové shlukování a další shlukovací algoritmy.

7.3 Relevanční zpětná vazba

Relevanční zpětná vazba je algoritmus popsany v publikacích [148, 23, 82]. Dal by se přirovnat k postupu který běžně podstupujeme například při výběru zboží.

Postup je jednoduchý:

1. Získáme základní výsledek, ten ukážeme uživateli.
2. Uživatel vybere jeden z výsledků který se mu nejvíce líbí (či nějak sdělí relevanci výsledku).
3. Aplikujeme postup strojového učení na uživatelovo rozhodnutí (v jednoduchosti upravíme množinu nabízeného tak, aby byla blízko vybraným výsledkům). Pokud není spokojen s výběrem iterujeme na 1. bod.

Postupy užívané na cestě k výsledku jsou převyvažování (měníme váhy příznaků, podle toho jaké uživatel vybírá alternativy), posun centra parametrů odhadovaného požadovaného obrázku a další. Nevýhoda relevanční zpětné vazby je, že uživatel musí projít většinou 5–6 iterací, což je pro většinu uživatelů příliš velké zdržení.

7.4 Sémantická šablona

Sémantická šablona je propojení mezi vysokoúrovňovým konceptem a nízkoúrovňovými příznaky obrázku.

Můžeme ji definovat jako

$$ST = \{C, F, W\},$$

kde C je koncept (tedy například klíčové slovo), F je těžiště příznaků obrázků odpovídajících konceptu a W je váha aplikovaná na příznakové vektory.

Sémantické šablony může buď vytvářet expert přímou definicí kombinací příznaků a konceptu, nebo jsou tvořeny systémem fungujícím na principu relevantní zpětné vazby, kdy uživatel cykluje u definice odpovídajících výsledků pro daný koncept dokud není spokojený.

Příkladem takové šablony je třeba lexikon textur popsany v publikaci [85].

7.5 Fúze mimoobrazových informací s obrazovými

V reálném použití většinou obrázky neexistují absolutně odděleně od ostatního obsahu. Například na webu můžeme najít spoustu informací přímo spojených s obrázkem (od jména souboru, přes EXIF data obsažená ve fotografiích, až po informace obsažené v HTML tagu obrázku a jeho okolí). Taková data pak pomáhají při kombinování klasifikátorů (přidáme klasifikátor klasifikující podle textu).

Kapitola 8

Způsoby hodnocení systémů CBIR

K hodnocení systémů CBIR se používá metrik užívaných k hodnocení běžných vyhledávacích algoritmů (textu apod.). Často se porovnává jen vizuální výstup (snímky obrazovky) pro určité dotazy. Navíc pokud je dotazem obrázků je relevance výsledku silně subjektivní. Pěkný popis problematiky najdeme u Muller et al. [99].

8.1 Testovací data

Důležitým předpokladem pro spravedlivé porovnání CBIR systémů je použití stejných testovacích dat při testu na všech srovnávaných systémech.

S postupem času se vyrojily množiny dat které se staly prakticky standardem pro porovnání dat.

Jednou z nich je například sbírka jménem Corel Stock Photos. Byla to sbírka obrázků opatřená tagy které popisovaly obsah obrázku. Bohužel se dnes už na internetu nenachází její prezentace.

Pro porovnání textur se standardem stala množina nazvaná Brodatz textures [12].

Dalším a aktualizovaným zkušebním nástrojem je sbírka Caltech (Caltech101 [13], aktualizace – Caltech256), obsahuje 101 (256) různých kategorií fotografií.

V posledních letech proběhlo několik workshopů v rámci nichž se na velkých datových sbírkách srovnávaly různé algoritmy nejen na vyhledávání obrázků dle obsahu, každý měl svoji speciální sbírku dat, za všechny jmenujme například ImagEval 2006 [59], který prováděl srovnání nad více než 130 000 vybranými obrázky.

8.2 Hodnocení porovnávacích metod

Různé metody hodnocení systémů CBIR rozebírají pánové Huijsmans a Sebe [52], my zde uvedeme jen několik základních způsobů měření dokonalosti systémů. Krom různých měřítek přesnosti a úspěšnosti (která zde popisujeme) se též používá klasických technik k porovnávání algoritmů a systémů, jako je časová složitost nebo třeba i uživatelská přívětivost.

Mějme systém, který v závislosti na zadaném dotazu vrací vybranou množinu obrázků. Míry používané k hodnocení vyhledávacích systémů (nejen pro vyhledávání obrázků) jsou:

Přesnost (precision) – je vyjádřena jako poměr relevantních obrázků kterými systémem odpověděl ku počtu všech obrázků kterými systém odpověděl, tedy

$$Pr(dotaz) = \frac{pocet_rel(odpoved)}{pocet(odpoved)},$$

Úplnost (recall) – je vyjádřena jako poměr relevantních obrázků kterými systém odpověděl ku počtu všech relevantních ze kterých mohl vybrat. Je většinou nepřímo úměrná přesnosti a přímo úměrná počtu vybíraných obrázků (čemuž se říká *scope*), tedy

$$Rec(dotaz) = \frac{pocet_rel(odpoved)}{pocet_rel(prohledavana_mnozina)},$$

Sumarizační křivky – Přesnost, úplnost a velikost výsledku (*scope*) se často znázorňují pomocí vztahových křivek, jako jsou *precision-recall* (přesnost × úplnost), nebo *precision-scope* (přesnost × počet vrácených) křivek,

Průměrné ohodnocení (average precision) – je rozšíření míry přesnosti o ohodnocení výsledků (které dodává systém jako hodnocení podobnosti každého vydaného obrázku vůči zadání dotazu). Je definováno jako průměr z ohodnocení obrázků, kterými systém odpověděl na dotaz,

Střední průměrné ohodnocení (mean average precision) – je průměr průměrného ohodnocení přes několik dotazů.

Jistě jsme nemohli popsat všechny možné způsoby hodnocení systémů CBIR. Zvědavý čtenář může nahlédnout například do popisu hodnocení těchto algoritmů které používá standard MPEG 7 ([91]), či si přečíst podrobnější článek [52] na toto téma.

Kapitola 9

Závěr

V práci jsme se seznámili s principy, na kterých fungují systémy vyhledávající obrázky podle obsahu. Probrali jsme příznaky, kterých si systémy všímají. Nízkoúrovňové příznaky, založené na barvě, tvaru, textuře či rozložení jsme prozkoumali a popsali podrobněji. Příznaky vyšší sémantické úrovně jsme prošli zběžně, neboť nespádaly do zaměření této práce. Též jsme se dotkli tématu správného rozdělení obrazu na objekty (segmentace). Podrobněji byly zkoumány postupy konstrukce, porovnávání signatur a určení podobnosti z jejich srovnání. Nakonec jsme se seznámili s některými způsoby srovnávání systému CBIR.

Je škoda, že nezbyl čas alespoň některé metody naprogramovat a porovnat prakticky. Tímto směrem by se měl pohybovat případný další rozvoj práce. Široké obzory se též rozvírají nad zkoumáním příznaků vyšší sémantické úrovně.

Literatura

- [1] Amores, J., Sebe, N., Radeva, P., Gevers, T., and Smeulders, A. 2004. *Boosting contextual information in content-based image retrieval*. In Proc. MIR Workshop, ACM Multimedia.
- [2] Amores, J., Sebe, N., and Radeva, P. 2005. *Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors*. In Proc. IEEE CVPR.
- [3] Andalo F. A. , Miranda P., Torres R. da S., and Falcão A. X. .*A New Shape Descriptor based on Tensor Scale* . 8th International Symposium on Mathematical Morphology, Rio de Janeiro. ISMM 2007, 2007. p. 141–152.
- [4] Arica N. and Vural F. T. Y., *BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics*, Pattern Recognition Letters, 24(9–10):1627–1639, June 2003.
- [5] Bartolini, I., Ciaccia, P., and Patella, M. 2005. *Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance*. IEEE Trans. Pattern Analysis and Machine Intelligence 27, 1, 142–147.
- [6] Belongie, S., Malik, J., and Puzicha, J. 2002. *Shape matching and object recognition using shape contexts*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 4, 509–522.
- [7] Beneš M., Zitová B., *Nephele: Databáze restaurátorských zpráv s možností vyhledávání podle textové obrazové informace* . (Interní publikace DAR – ÚTIA AV ČR 2005/43) ÚTIA AV ČR, Praha 2005, 53 pp.
- [8] Berretti, S., Del Bimbo, A., and Pala, P. 2000. *Retrieval by shape similarity with perceptual distance and effective indexing*. IEEE Trans. Multimedia 2, 4, 225–239.
- [9] Berretti, S., Del Bimbo, A., and Vicario, E. 2003. *Weighted walkthroughs between extended entities for retrieval by spatial arrangement*. IEEE Trans. Multimedia 5, 1, 52–70.
- [10] Bober M. *MPEG-7 Visual Shape Descriptors* . IEEE Transactions on Circuits and Systems for Video Technology, 11(6):716–719, June 2001.

- [11] Bouchard, G. and Triggs, B. 2005. *Hierarchical part-based visual object categorization*. In Proc. IEEE CVPR.
- [12] Brodatz P., *Textures – a photographic album for artists and designers*. New York: Dover Publications Inc., 1966.
- [13] Caltech101, 2004.
http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html.
- [14] Carneiro, G. and Lowe, D. 2006. *Sparse flexible models of local features*. In Proc. ECCV.
- [15] Carneiro, G. and Vasconcelos, N. 2005. *Minimum bayes error features for visual recognition by sequential feature selection and extraction*. In Proc. Canadian Conf. Computer and Robot Vision.
- [16] Carson, C., Belongie, S., Greenspan, H., and Malik, J. 2002. *Blobworld: Image segmentation using expectation-maximization and its application to image querying*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 8, 1026–1038.
- [17] Ciaccia P., Patella M., and Zezula P., *M-tree: An efficient access method for similarity search in metric spaces*. in Proc. Int. Conf. VLDB, 1997, pp. 522–525.
- [18] Cohen A., Daubechies I., Feauveau J. C., *Biorthogonal bases of compactly supported wavelets*, Commun. Pure Appl. Math., v. 45, p. 485–560, 1992
- [19] Comaniciu, D. and Meer, P. 2002. *Mean shift: A robust approach toward feature space analysis*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 5, 603–619.
- [20] Comaniciu D., Meer P., *Robust analysis of feature spaces: color image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 750–755.
- [21] Conway D.M., *An experimental comparison of three natural language color naming models*, Proceedings of the East-West International Conference on Human-Computer Interactions, St. Petersburg, Russia, 1992, pp. 328–339.
- [22] Dagli, C. and Huang, T. S. 2004. *A framework for grid-based image retrieval*. In Proc. IEEE ICPR.
- [23] Datta R. , Joshi D., Li J., and Wang J.Z., *Image Retrieval: Ideas, Influences, and Trends of the New Age*, ACM Computing Surveys, Vol. 40, No. 2, April, 2008.
- [24] Daubechies I., *Ten Lectures on Wavelets*. Philadelphia: SIAM, 1992

- [25] Deng, Y. and Manjunath, B. 2001. *Unsupervised segmentation of color-texture regions in images and video*. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 8, 800–810.
- [26] Deng, Y., Manjunath, B. S., Kenney, C., Moore, M. S., and Shin, H. 2001. *An efficient color representation for image retrieval*. IEEE Trans. Image Processing 10, 1, 140–147.
- [27] de Silva, V. and Tenenbaum, J. 2003. *Global versus local methods in nonlinear dimensionality reduction*. In Proc. NIPS.
- [28] Do, M. N. and Vetterli, M. 2002. *Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance*. IEEE Trans. Image Processing 11, 2, 146–158.
- [29] Du, Y. and Wang, J. Z. 2001. *A scalable integrated region-based image retrieval system*. In Proc. IEEE ICIP.
- [30] Duchon J., *Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces*, Constructive Theory of Functions of Several Variables, W. Schempp and K. Zeller, eds., pp. 85–100, Berlin:Springer-Verlag, 1977.
- [31] Falcão A.X., Lotufo R. A., Araujo G.. *The Image Foresting Transformation*. Relatorio Tecnico IC-00–12, 2000
- [32] Falcão A. X., Stolfi J., and Lotufo R. A.. *The Image Foresting Transform: Theory, Algorithms, and Applications*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(1):19–29, Jan 2004
- [33] Feng H., Castanon D.A., Karl W.C., *A curve evolution approach for image segmentation using adaptive flows*. Proceedings of the International Conference on Computer Vision (ICCV’ 01), 2001, pp. 494–499.
- [34] Feng H., Chua T.-S., *A bootstrapping approach to annotating large image collection*, Workshop on Multimedia Information Retrieval in ACM Multimedia, November 2003, pp. 55–62.
- [35] Fergus, R., Perona, P., and Zisserman, A. 2003. *Object class recognition by unsupervised scale-invariant learning*. In Proc. IEEE CVPR.
- [36] Fergus, R., Perona, P., and Zisserman, A. 2005. *A sparse object category model for efficient learning and exhaustive recognition*. In Proc. IEEE CVPR.
- [37] Finlayson, G. 1996. *Color in perspective*. IEEE Trans. Pattern Analysis and Machine Intelligence 18, 10, 1034–1038.
- [38] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. 1995. *Query by image and video content: The qbic system*. IEEE Computer 28, 9, 23–32.

- [39] Flusser Jan and Boldys Jiri and Zitova Barbara, *Moment Forms Invariant to Rotation and Blur in Arbitrary Number of Dimensions*, 2003, <http://citeseer.ist.psu.edu/flusser03moment.html>
- [40] Gouet, V. and Boujemaa, N. 2002. *On the robustness of color points of interest for image retrieval*. In Proc. IEEE ICIP.
- [41] Grauman, K. and Darrell, T. 2005. *Efficient image matching with distributions of local invariant features*. In Proc. IEEE CVPR.
- [42] Gunjan Agarwal and Siddhartha K Goel *Object Recognition through Curvature Scale Space*, <http://www.cse.iitk.ac.in/amit/courses/AI/99/gunjan/>
- [43] Guyon, I. and Elisseeff, A. 2003. *An introduction to variable and feature selection*. J. Machine Learning Research 3, 1157–1182.
- [44] Hadjidemetriou, E., Grossberg, M. D., and Nayar, S. K. 2004. *Multiresolution histograms and their use for recognition*. IEEE Trans. Pattern Analysis and Machine Intelligence 26, 7, 831–847.
- [45] Haralick Robert M., Shanmugam K., Dinstein Its'hak *Textural Features for Image Classification*, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6): 610–621, 1973
- [46] He, J., Li, M., Zhang, H.-J., Tong, H., and Zhang, C. 2004a. *Manifold-ranking based image retrieval*. In Proc. ACM Multimedia.
- [47] He, X., Ma, W.-Y., and Zhang, H.-J. 2004. *Learning an image manifold for retrieval*. In Proc. ACM Multimedia.
- [48] He, X. 2004. *Incremental semi-supervised subspace learning for image retrieval*. In Proc. ACM Multimedia.
- [49] Hoiem, D., Sukthankar, R., Schneiderman, H., and Huston, L. 2004. *Object-based image retrieval using the statistical structure of images*. In Proc. IEEE CVPR.
- [50] Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W.-J., and Zabih, R. 1999. *Spatial color indexing and applications*. Int. J. Computer Vision 35, 3, 245–268.
- [51] Huang Y.L., *A Fast Method for Textural Analysis of DCT-Based Image*, Journal of Information Science and Engineering, vol. 21, no. 1, pp. 181–194, 2005.
- [52] Huijsmans, D. P. and Sebe, N. 2005. *How to complete performance graphs in content-based image retrieval: Add generality and normalize scope*. IEEE Trans. Pattern Analysis and Machine Intelligence 27, 2, 245–251.
- [53] Hu. M. K., *Visual Pattern Recognition by Moment Invariants* . IRE Transactions on Information Theory, 8(2):179–187, 1962.

- [54] Chang, S., Shi, Q., and Yan, C. 1987. *Iconic indexing by 2-d strings*. IEEE Trans. Pattern Analysis and Machine Intelligence 9, 3, 413–427.
- [55] Chang, S., Yan, C., Dimitroff, D., and Arndt, T. 1988. *An intelligent image database system*. IEEE Trans. Software Engineering 14, 5, 681–688.
- [56] Chen, Y. and Wang, J. Z. 2004. *Image categorization by learning and reasoning with regions*. J. Machine Learning Research 5, 913–939.
- [57] Chen, J., Pappas, T., Mojsilovic, A., and Rogowitz, B. 2002. *Adaptive image segmentation based on color and texture*. In Proc. IEEE ICIP.
- [58] Chen, C.-C., Wactlar, H., Wang, J. Z., and Kiernan, K. 2005. *Digital imagery for significant cultural and historical materials – an emerging research field bridging people, culture, and technologies*. Int. J. on Digital Libraries 5, 4, 275–286
- [59] ImagEVAL, 2006.
<http://www.imageval.com>.
- [60] Iqbal, Q. and Aggarwal, J. K. 2002. *Retrieval by classification of images containing large manmade objects using perceptual grouping*. Pattern Recognition J. 35, 7, 1463–1479.
- [61] Jain, A. and Farrokhnia, F. 1990. *Unsupervised texture segmentation using gabor filters*. In Proc. Int. Conf. Systems, Man and Cybernetics.
- [62] Jeong, S., Won, C. S., and Gray, R. 2004. *Image retrieval using color histograms generated by gauss mixture vector quantization*. Computer Vision and Image Understanding 9, 1–3, 44–66
- [63] Jin, R. and Hauptmann, A. 2002. *Using a probabilistic source model for comparing images*. In Proc. IEEE ICIP
- [64] Jing F., Li M., Zhang L., Zhang H.-J., Zhang B., *Learning in region-based image retrieval*. Proceedings of the International Conference on Image and Video Retrieval (CIVR2003), 2003, pp. 206–215.
- [65] Jing, F., Li, M., Zhang, H.-J., and Zhang, B. 2004a. *An efficient and effective region-based image retrieval framework*. IEEE Trans. Image Processing 13, 5, 699–709.
- [66] Jonker R. and Volgenant A., *A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems*, Computing, vol. 38, pp. 325–340, 1987.
- [67] Kass M. and Witkin A., *Analyzing oriented patterns*, Computer Vision Graphics and Image Processing, 37(3):362–385, 1987

- [68] Kauppinen H., Seppanen T., and Pietikainen M., *An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pp. 201–207, Feb. 1995.
- [69] Kim, Y. S., Street, W. N., and Menczer, F. 2000. *Feature selection in unsupervised learning via evolutionary search*. In Proc. ACM SIGKDD.
- [70] Ko, B. and Byun, H. 2002. *Integrated region-based image retrieval using region's spatial relationships*. In Proc. IEEE ICPR.
- [71] Kontanzad A. and Hong Y.H., IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 489–497, May 1990.
- [72] Kurani A., Xu D. H., Furst J. D., Raicu & D. S., *Co-occurrence matrices for volumetric data*, The 7th IASTED International Conference on Computer Graphics and Imaging – CGIM 2004, Kauai, Hawaii, USA, in August 16–18, 2004.
- [73] Laaksonen, J., Koskela, M., and Oja, E. 2002. *Picsom-self-organizing image retrieval with mpeg-7 content descriptors*. IEEE Trans. Neural Networks 13, 4, 841–853
- [74] Latecki, L. J. and Lakamper, R. 2000. *Shape similarity measure based on correspondence of visual parts*. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 10, 1185–1190.
- [75] Levina, E. and Bickel, P. 2001. *The earth mover's distance is the mallows distance: Some insights from statistics*. In Proc. IEEE ICCV.
- [76] Li, J. and Wang, J. Z. 2003. *Automatic linguistic indexing of pictures by a statistical modeling approach*. IEEE Trans. Pattern Analysis and Machine Intelligence 25, 9, 1075–1088
- [77] Li, J. and Wang, J. Z. 2004. *Studying digital imagery of ancient paintings by mixtures of stochastic models*. IEEE Trans. Image Processing 13, 3, 340–353.
- [78] Li, J., Wang, J. Z., and Wiederhold, G. 2000. *Irm: Integrated region matching for image retrieval*. In Proc. ACM Multimedia.
- [79] Li Yong, Chen Xiujuan, Fu Xuezheng, Belkasim, S., 2007, *Multi-Level Discrete Cosine Transform for Content-Based Image Retrieval by Support Vector Machines*, Image Processing, 2007. ICIP 2007. IEEE International Conference, Volume: 6, VI – 21-VI – 24
- [80] Li Xiaobin, Tian Zheng *Wavelet Energy Signature: Comparison and Analysis*. ICONIP (2) 2006: 474–480

- [81] Liu Y., Zhang D.S., Lu G., Ma W.-Y., *Region-based image retrieval with perceptual colors*, Proceedings of the Pacific-Rim Multimedia Conference (PCM), December 2004, pp. 931–938.
- [82] Liu, Y., Zhang, D., Lu, G., and Ma, W. 2007. *A survey of content-based image retrieval with high-level semantics*. Pattern Recogn. 40, 1 (Jan. 2007), 262–282.
- [83] Liu Y., Zhang D.S., Lu G., Ma W.-Y., *Region-based image retrieval with perceptual colors*, Proceedings of the Pacific-Rim Multimedia Conference (PCM), December 2004, pp. 931–938.
- [84] Liu Y., Zhou X., Ma W.Y., *Extraction of texture features from arbitrary-shaped regions for image retrieval*. International Conference on Multimedia and Expo (ICME04), Taipei, June 2004, pp. 1891–1894.
- [85] Ma, W.-Y. and Manjunath, B. 1998. *Texture thesaurus for browsing large aerial photographs*. J. American Society for Information Science 49, 7, 633–648.
- [86] Malik, J. and Perona, P. 1990. *Preattentive texture discrimination with early vision mechanisms*. J. Optical Society of America A 7, 5, 923–932.
- [87] Malik, J., Belongie, S., Leung, T. K., and Shi, J. 2001. *Contour and texture analysis for image segmentation*. Intl. J. Computer Vision 43, 1, 7–27.
- [88] Mallows, C. L. 1972. *A note on asymptotic joint normality*. Annals of Mathematical Statistics 43, 2, 508–515.
- [89] Ma W.Y., Manjunath B., *Netra: a toolbox for navigating large image databases*, Proceedings of the IEEE International Conference on Image Processing, 1997, pp. 568–571.
- [90] Manjunath, B. and Ma, W.-Y. 1996. *Texture features for browsing and retrieval of image data*. IEEE Trans. Pattern Analysis and Machine Intelligence 18, 8, 837–842.
- [91] Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., and Yamada, A. 2001. *Color and texture descriptors*. IEEE Trans. Circuits and Systems for Video Technology 11, 6, 703–715.
- [92] Mathiassen, J. R., Skavhaug, A., and Bo, K. 2002. *Texture similarity measure using kullback-leibler divergence between gamma distributions*. In Proc. ECCV.
- [93] Mehrotra, R. and Gary, J. E. 1995. *Similar-shape retrieval in shape data management*. IEEE Computer 28, 9, 57–62.
- [94] Mezaris V., Kompatsiaris I., Strintzis M.G., *An ontology approach to object-based image retrieval*, Proceedings of the ICIP, vol. II, 2003, pp. 511–514.
- [95] Mikolajczyk, K. and Schmid, C. 2004. *Scale and affine invariant interest point detectors*. Intl. J. Computer Vision 60, 1, 63–86.

- [96] Miranda P., Torres R. da S., and Falcão A. X. *TSD: A Shape Descriptor Based on a Distribution of Tensor Scale Local Orientation* . In XVIII Brazilian Symposium on Computer
- [97] Mitra, P., Murthy, C., and Pal, S. 2002. *Unsupervised feature selection using feature similarity*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 3, 301–312.
- [98] Mokhtarian F., Abbasi S., *Shape similarity retrieval under affine transforms* . Pattern Recognition 35 (2002) 31–41.
- [99] Muller, H., Muller, W., Squire, D. M., Marchand-Maillet, S., and Pun, T. 2001. *Performance evaluation in content-based image retrieval: Overview and proposals*. Pattern Recognition Letters 22, 5, 593–601.
- [100] Muller, H., Michoux, N., Bandon, D., and Geissbuhler, A. 2004. *A review of content-based image retrieval systems in medical applications – clinical benefits and future directions*. Intl. J. Medical Informatics 73, 1, 1–23.
- [101] Muller, H., Michoux, N., Bandon, D., and Geissbuhler, A. 2004. *A review of content-based image retrieval systems in medical applications – clinical benefits and future directions* . Intl. J. Medical Informatics 73, 1, 1–23.
- [102] Natsev, A. and Smith, J. 2002. *A study of image retrieval by anchoring*. In Proc. IEEE ICME.
- [103] Natsev, A., Rastogi, R., and Shim, K. 2004. *Walrus: A similarity retrieval algorithm for image databases*. IEEE Trans. Knowledge and Data Engineering 16, 3, 301–316.
- [104] Pass G., Zabih R., Miller J. *Comparing Images Using Color Coherence Vectors* . In Proceedings of the fourth ACM international conference on Multimedia, pages 65–73, 1996.
- [105] Pennebaker W. B. and Mitchell J. L., *JPEG – Still Image Data Compression Standard*, Newyork: International Thomsan Publishing, 1993.
- [106] Petraglia, G., Sebillio, M., Tucci, M., and Tortora, G. 2001. *Virtual images for similarity retrieval in image databases*. IEEE Trans. Knowledge and Data Engineering 13, 6, 951–967.
- [107] Petrakis, E. G. M., Diplaros, A., and Milios, E. 2002. *Matching and retrieval of distorted and occluded shapes using dynamic programming*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 4, 509–522.
- [108] Pi, M., Mandal, M. K., and Basu, A. 2005. *Image retrieval based on histogram of fractal parameters*. IEEE Trans. Multimedia 7, 4, 597–605.
- [109] Plataniotis K.N., Venetsanopoulos A.N., *Color Image Processing and Applications*, Springer, Berlin, 2000.

- [110] Portilla, J. and Simoncelli, E. 2000. *A parametric texture model based on joint statistics of complex wavelet coefficients*. Int. J. Computer Vision 40, 1, 49–71.
- [111] Rao A.R., Lohse G.L., *Towards a texture naming system: identifying relevant dimensions of texture* . IEEE Proceedings of the Fourth Conference on Visualization, 1993, pp. 220–227.
- [112] Ren W., Singh M., Singh C., *Image retrieval using spatial context* . Ninth International Workshop on Systems, Signals and Image Processing (IWSSIP'02), Manchester, November, 2002.
- [113] Rubner, Y., Tomasi, C., and Guibas, L. J. 2000. *The earth mover's distance as a metric for image retrieval*. Int. J. Comput. Vision 40, 99–121.
- [114] Rubner, Y., Tomasi, C., and Guibas, L. J. 2002. *A metric for distribution with applications to image databases*. In Proc. IEEE ICCV.
- [115] Saha P.K. .*Tensor Scale: A Local Morphometric Parameter With Applications to Computer Vision and Image Processing* . Technical Report 306, Medical Image Processing Group, Department of Radiology, University of Pennsylvania, September 2003.
- [116] Shi, J. and Malik, J. 2000. *Normalized cuts and image segmentation*. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 8, 888–905.
- [117] Schmid, C. and Mohr, R. 1997. *Local grayvalue invariants for image retrieval*. IEEE Trans. Pattern Analysis and Machine Intelligence 19, 5, 530–535.
- [118] Simoncelli, E.P. and Freeman, W.T. *The steerable pyramid: A flexible architecture for multi-scale derivative computation*. In Second Int'l Conf. on Image Proc., Washington, DC, October 1995. Vol. III, IEEE Sig. Proc. Society, pp. 444–447.
- [119] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. 2000. *Content-based image retrieval at the end of the early years*. IEEE Trans. Pattern Anal. Mach. Intell. 22, 12 (Dec. 2000), 1349–1380.
- [120] Smith, J. and Chang, S.-F. 1997b. *Visualeek: a fully automated content-based image query system*. In Proc. ACM Multimedia.
- [121] Smith A.R., *Spline Tutorial Notes* . Tech memo 77, Computer Division, Lucasfilm, May 1983. Also tutorial notes at SIGGRAPHs 83 and 84, (<http://www.alvyray.com/Memos/MemosPixar.htm#SplineTutorial>).
- [122] Snoek, C. G. M. and Worring, M. 2005. *Multimodal video indexing: A review of the state-of-the-art*. Multimedia Tools and Applications 25, 1, 5–35.

- [123] Stehling R.O., Nascimento M.A., Falcão A.X., *A Compact and Efficient Image Retrieval* . Approach Based on Border/Interior Pixel Classification. In Proceedings of the 11th ACM International. Conference on Information and Knowledge Management, pages 102–109, McLean, Virginia, USA, November 2002.
- [124] Stricker M. A. and Orengo M. *Similarity of Color Images* . Storage and Retrieval for Image and Video Databases (SPIE), pages 381–392, 1995.
- [125] Swain, M. and Ballard, B. 1991. *Color indexing*. Int. J. Computer Vision 7, 1, 11–32.
- [126] Tamura H., Mori S., Yamawaki T., *Texture features corresponding to visual perception* , IEEE Trans. Syst. Man Cybern. 8 (6) (1978) 460–473.
- [127] Theoharatos, C., Laskaris, N. A., Economou, G., and Fotopoulos, S. 2005. *A generic scheme for color image retrieval based on the multivariate wald-wolfowitz test*. IEEE Trans. Knowledge and Data Engineering 17, 6, 808–819.
- [128] Tian, Q., Sebe, N., Lew, M. S., Loupias, E., and Huang, T. S. 2001. *Image retrieval using wavelet-based salient points*. J. Electronic Imaging 10, 4, 835–849.
- [129] Tieu, K. and Viola, P. 2004. *Boosting image retrieval*. Intl. J. Computer Vision 56, 1/2, 17–36.
- [130] Torres, Falcão, 2006 *Content-Based Image Retrieval: Theory and Applications*
- [131] Torres R. da S., Falcão A.X., *Contour salience descriptors for effective image retrieval and analysis*, Image and Vision Computing Volume 25, Issue 1, SIB-GRAPI, January 2007, Pages 3–13.
- [132] Tu, Z. and Zhu, S.-C. 2002. *Image segmentation by data-driven markov chain monte carlo*. IEEE Trans. Pattern Analysis and Machine Intelligence 24, 5, 657–673.
- [133] Tuytelaars, T. and van Gool, L. 1999. *Content-based image retrieval based on local affinely invariant regions*. In Proc. VISUAL.
- [134] Unser, M. 1995. *Texture classification and segmentation using wavelet frames*. IEEE Trans. Image Processing 4, 11, 1549–1560.
- [135] Varanasi M. K. and Aazhang B., *Parametric generalized Gaussian density estimation*, J. Acoust. Soc. Amer., vol. 86, pp. 1404–1415, 1989.
- [136] Vasconcelos, N. and Lippman, A. 2000b. *A probabilistic architecture for content-based image retrieval*. In Proc. IEEE CVPR.
- [137] Vasconcelos, N. and Lippman, A. 2005. *A multiresolution manifold distance for invariant image similarity*. IEEE Trans. Multimedia 7, 1, 127–142.

- [138] Vasconcelos, N. 2004. *On the efficient evaluation of probabilistic similarity functions for image retrieval*. IEEE Trans. Information Theory 50, 7, 1482–1496.
- [139] Veltkamp R.C., Tanase M. *Content-Based Image Retrieval Systems: A Survey*. Technical Report UU-CS-,2000–02
- [140] Wang, J., Li, J., and Wiederhold, G. 2001. *Simplicity: Semantics-sensitive integrated matching for picture libraries*. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 9, 947–963.
- [141] Wang, J., Wiederhold, G., Firschein, O., and Wei, S. 1998. *Content-based image indexing and searching using daubechies' wavelets*. Int. J. Digital Libraries 1, 4, 311–328.
- [142] Wang J.Z., Li J., Chan D., Wiederhold G., *Semantics-sensitive retrieval for digital picture libraries* . Digital Library Magazine, vol. 5(11), 1999.
- [143] Wang, Y.P., Pavlidis, T., *Optimal correspondence of string subsequences*, IEEE Trans. PAMI 12, 1080–1087, 1990.
- [144] Wang, Y. H. 2003. *Image indexing and similarity retrieval based on spatial relationship model*. Information Sciences – Informatics and Computer Science 154, 1–2, 39–58.
- [145] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. 2000. *Feature selection for svms*. In Proc. NIPS.
- [146] Zhang, R. and Zhang, Z. 2004. *Hidden semantic concept discovery in region based image retrieval*. In Proc. IEEE CVPR.
- [147] Zhang D. and Lu. G., *Review of Shape Representation and Description* . Pattern Recognition, 37(1):1–19, Jan 2004.
- [148] Zhou, X. S. and Huang, T. S. 2003. *Relevance feedback in image retrieval: A comprehensive review*. Multimedia Systems 8, 536–544.
- [149] Zhou, D., Weston, J., Gretton, A., Bousquet, O., and Scholkopf, B. 2003. *Ranking on data manifolds*. In Proc. NIPS.
- [150] Zhu, S.-C. and Yuille, A. 1996. *Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation*. IEEE Trans. Pattern Analysis and Machine Intelligence 18, 9, 884–900.
- [151] Zhu, L., Zhang, A., Rao, A., and Srihari, R. 2000. *Keyblock: An approach for content-based image retrieval*. In Proc. ACM Multimedia.