Využití LSI a M-stromu při indexování a vyhledávání obrázků

Tomáš Skopal[1], Michal Kolovrat[2] a Václav Snášel[2]

[1]Katedra softwarového inženýrství, MFF UK Praha,
Malostranské nám. 25, 118 00, Praha
[2]Katedra informatiky, FEI, VŠB – Technická Univerzita Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba
tomas@skopal.net, {michal.kolovrat,vaclav.snasel}@vsb.cz

Abstrakt. Při práci s multimediálními dokumenty musíme často řešit problém, jak v nich efektivně a rychle vyhledávat. Multimediální dokumenty většinou reprezentujeme vektory ve vysokodimenzionálním prostoru, neboť v takto modelované kolekci dokumentů lze jednodušeji definovat sémantiku i mechanismus samotného vyhledávání. V tomto příspěvku prezentujeme vytváření vektorů obrázků (extrakci vlastností z obrázků) v modelu LSI (resp. pomocí singulárního rozkladu). Současně ukážeme vliv takové extrakce na efektivitu indexování/vyhledávání pomocí datové struktury M-strom. Vzhledem k aplikaci modelu LSI na obrázky poukážeme rovněž na některé zajímavé souvislosti, které při klasickém použití LSI na textových kolekcích nejsou přímo patrné.

Klíčová slova: LSI, podobnostní vyhledávání obrázků, M-strom

1 Úvod

V současné době, s nárůstem informačních technologií, vzniká obrovské množství multimediálních dat – mezi tato data patří zejména textové, obrazové a zvukové dokumenty.

Multimediální data je potřeba nějakým způsobem vhodně reprezentovat a organizovat, aby v nich následně šlo jednoduše a rychle vyhledávat [6]. Jednotlivé multimediální dokumenty většinou reprezentujeme vektory ve vysokodimenzionálním prostoru, neboť v takto modelované kolekci dokumentů lze jednodušeji definovat sémantiku i mechanismus samotného vyhledávání podle obsahu (content-based search). V současnosti se stále ve velké míře používá sekvenční způsob vyhledávání, kdy je celá kolekce dokumentů (resp. její reprezentace, tj. kolekce vektorů) postupně sekvenčně načtena a hledané dokumenty jsou nacházeny pomocí vzájemného porovnáváním všech dokumentů v kolekci s dotazovým dokumentem (pomocí nějaké funkce podobnosti, definované pro dva vektory dokumentů). Tento způsob vyhledávání však již v současné době není možný, neboť kolekce dokumentů jsou velmi rozsáhlé a sekvenční průchod takovou kolekcí by trval neúnosně dlouho. Proto byly navrženy přístupy a metody, které nejdříve

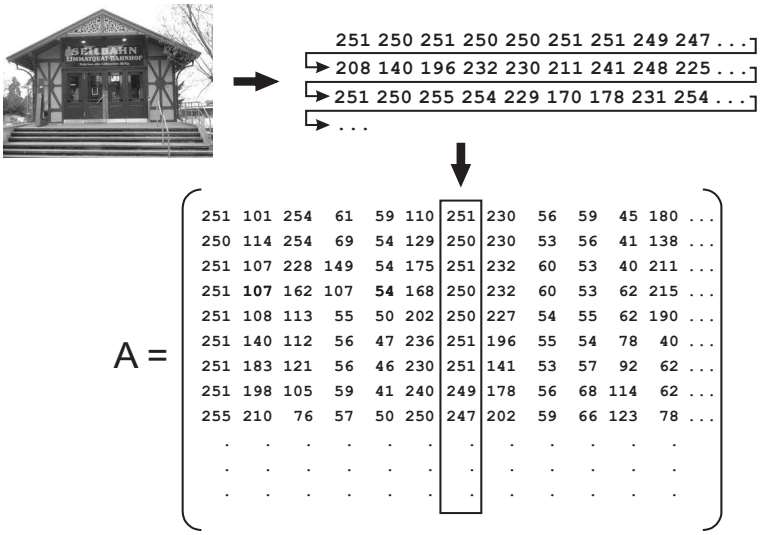
kolekci předzpracují do předem navržené struktury – indexu. Cílem indexování je tedy umožnit rychlé vyhodnocení uživatelského dotazu pomocí indexu. Na Obr. 1 uvádíme vzorek z experimentální kolekce 730 obrázků budov (posléze transformovaných do velikosti $80 \cdot 60$ pixelů v 256 odstínech šedi).

Obr. 1. Vzorek z kolekce obrázků budov.

2 Extrakce vektorů vlastností pomocí LSI

Metod pro extrakci vlastností z obrázků (a tím vytvoření jejich vektorové reprezentace) existuje celá řada. Nejčastěji jsou používány různé histogramy barev [3], textur [17], apod. Pro doménově specifické kolekce (např. otisky prstů, oční duhovky, obličeje) pak existuje mnoho speciálních přístupů. Pro přehled současných metod extrakce vlastností z obrázků odkazujeme na [13].

V našem příspěvku jsme použili jiný přístup [11, 12], kdy je celý obrázek velikosti $x \cdot y$ (resp. jasy/stupně šedi všech jeho pixelů) předeven na vektor dimenze $x \cdot y$. Transformace obrázku do vektoru je provedena zcela triviálně, a to "naskládáním" řádků obrázku za sebe, čímž vznikne vysokodimenzionální "jasový" vektor obrázku. V této fázi reprezentace obrázku ještě nejde o extrakci vlastností, podle kterých by se dalo posléze smysluplně vyhledávat, obrázek (resp. jeho jasová verze) lze bezezbytku zrekonstruovat z jasového vektoru. Nicméně díky tomu, že obrázky jsou reprezentovány jasovými vektory, můžeme celou kolekci obrázků (o velikosti n) reprezentovat maticí A řádu $x \cdot y \times n$, kde jednotlivé vektory obrázků tvoří sloupce matice (viz Obr. 2, $x = 80$, $y = 60$, $n = 730$, tj. matice A je řádu 4800×730).



Obr. 2. Transformace obrázku do vektoru jasů a jeho umístění v matici A.

2.1 Klasické LSI jako rozšíření vektorového modelu DIS

V oblasti *Dokumentografických informačních systémů* (Information Retrieval) [1, 10] se s modelem LSI (latent semantics indexing) setkáváme v kontextu reprezentace a indexování textových dokumentů ve vektorovém modelu [7, 2]. Textová kolekce obsahuje dohromady m unikátních termů, přičemž m-rozměrné vektory dokumentů (sloupce matice A) reprezentují frekvence/váhy jednotlivých termů v textových dokumentech. Pomocí singulárního rozkladu (SVD) matice A:

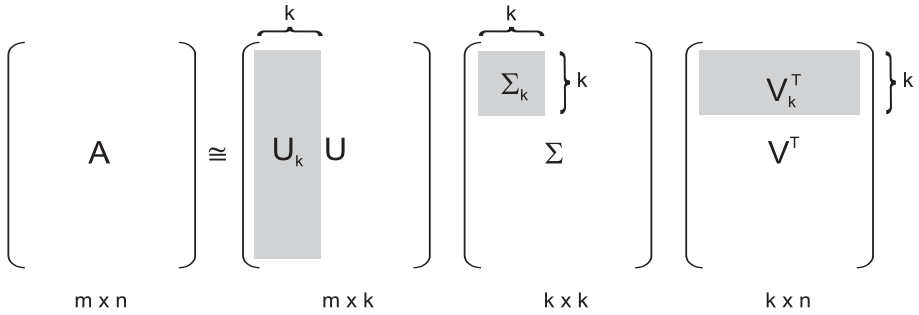
$$A = U \Sigma V^T$$

se v textové kolekci naleznou tzv. *vektory konceptů* (levé singulární vektory – sloupce v U), které lze interpretovat jako jednotlivá témata přítomná v kolekci. Vektory konceptů tvoří bázi v původním vysokodimenzionálním prostoru a jsou to v zásadě lineární kombinace termů (termy jsou nezávislé). Důležitou vlastností singulárního rozkladu je fakt, že vektory konceptů jsou uspořádány podle "důležitosti" (důležitost vyjadřují hodnoty singulárních čísel σ_i, uložené sestupně v diagonální matici Σ) – podle toho, v jaké míře a v kolika dokumentech se (ne)vyskytují. Tím je určeno, které koncepty jsou vzhledem ke kolekci sémanticky důležité (odtud indexování *latentní* (skryté) sémantiky), a které nedůležité, ty působí jako statistický šum. Sloupce matice ΣV^T obsahují vektory dokumentů (tzv. *vektory pseudo-dokumentů*), které jsou ale nyní vyjádřené v bázi U, tj. v *bázi konceptů*. Jeden vektor pseudo-dokumentu odpovídá nějaké lineární kombinaci všech nalezených vektorů konceptů, tj. příslušný dokument v nějaké míře (třeba i nulové nebo záporné) obsahuje každý z konceptů.

Vzhledem k tomu, že pouze prvních k konceptů lze považovat za sémanticky důležité (příslušná singulární čísla jsou vysoká), můžeme původní rozklad aproximovat jako

$$A \approx U_k \Sigma_k V_k^T$$

kde v U_k je prvních k nejdůležitějších vektorů konceptů, v Σ_k jsou příslušná singulární čísla a ve $\Sigma_k V_k^T$ vektory pseudo-dokumentů vyjádřené pouze pomocí prvních k vektorů konceptů (viz Obr. 3). Jinými slovy, SVD promítá původní m-rozměrné vektory dokumentů do prostoru dimenze k ($k \ll m$). Aproximace SVD rozkladu (rank-k SVD) matice A lze dosáhnout tak, že buď provedeme úplný SVD rozklad a posléze příslušné matice "ořežeme", anebo zvolíme některou z časově méně náročných numerických metod, která počítá přímo redukovaný rozklad (např. Lanczos, Arnoldi). Pokud chceme v kolekci dokumentů vyhledávat,



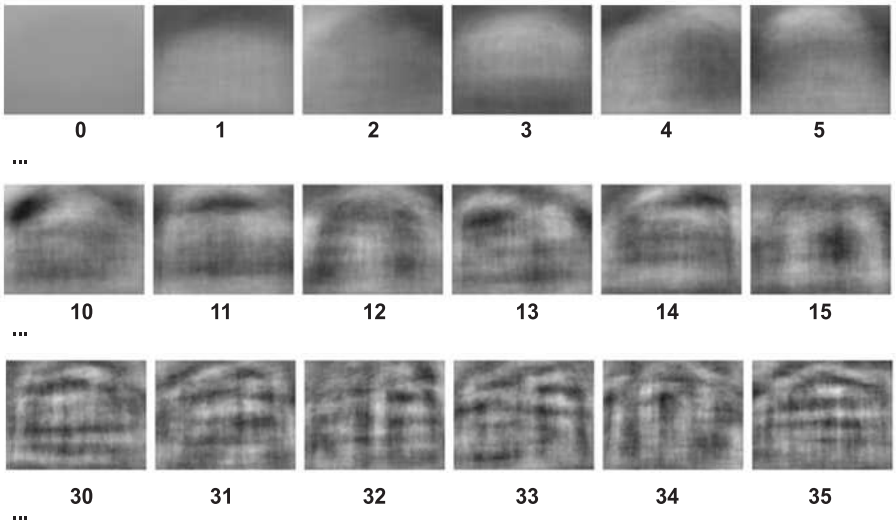
Obr. 3. Aproximace SVD rozkladu (rank-k SVD).

porovnáváme vektory pseudo-dokumentů s *vektorem pseudo-dotazu* na základě nějaké míry podobnosti (v klasickém vektorovém modelu DIS i v modelu LSI pro textové kolekce je to např. kosinová míra). Abychom mohli porovnávat vektory pseudo-dokumentů, potřebujeme z klasického vektoru dotazu q (reprezentovaného stejným způsobem jako jsou vektory dokumentů v A) zkonstruovat jeho projekci do U_k, tj. $U_k^T q$

2.2 LSI pro kolekce obrázků

Vzhledem k tomu, že SVD rozklad můžeme aplikovat na libovolnou matici, je model LSI aplikovatelný na libovolné typy dat. Interpretace matice A jako matice termů v dokumentech je pouze předmětem aplikace, tj. aplikace LSI jako rozšíření klasického vektorového modelu DIS. Obecně tedy můžeme model LSI adaptovat pro indexování libovolné multimediální kolekce, kterou lze reprezentovat sadou vektorů shodné dimenze. Přitom nezáleží na konkrétním způsobu extrakce vlastností z dokumentů, jedinou podmínkou je sestavení matice A tak, aby vektory vlastností dokumentů tvořily sloupce.

V našem případě nic nebrání sestavit matici A z jasových vektorů obrázků, jak bylo popsáno na začátku této kapitoly. Pomocí SVD matici A rozložíme stejně jako při klasickém použití LSI, tj. získáme aproximaci rozkladu $A \approx U_k \Sigma_k V_k^T$.



Obr. 4. Bázové obrázky (vizualizace konceptů, resp. singulárních vektorů) s příslušným pořadím singulárních čísel σ_i.

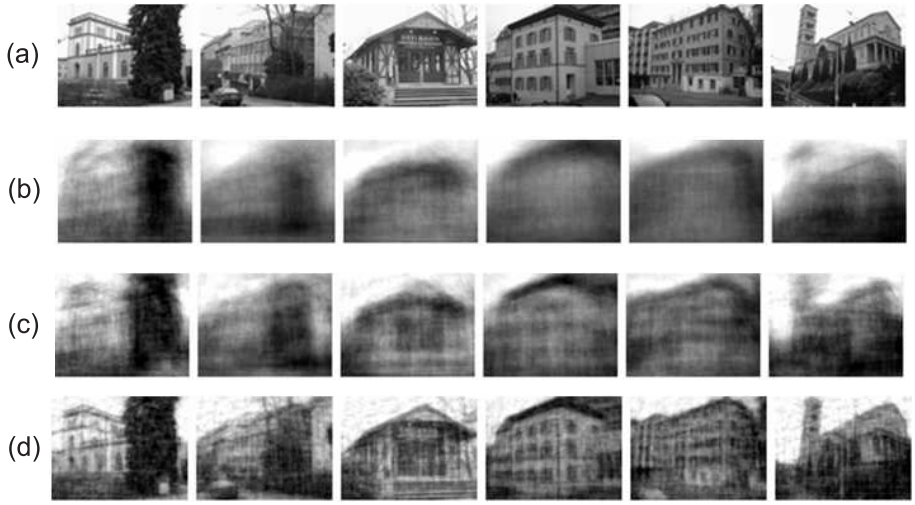
Zajímavá je interpretace a zejména vizualizace vektorů konceptů (báze U), které přestavují jakési "bázové obrázky", ze kterých je každý obrázek v kolekci složen (viz Obr. 4). Jinými slovy, vhodnou kombinací bázových obrázků dostaneme libovolný obrázek z kolekce (čím vyšší k, tím přesnější rekonstrukce obrázku).

Je zde jistá spojitost s diskrétní kosinovou transformací (DCT), využívanou při kompresi JPEG, kde je používáno např. 64 bázových obrázků pro obrazové segmenty velikosti 8×8, ovšem těchto 64 bázových obrázků je konstantních (jedná se o kombinace diskrétních kosinových funkcí s různou frekvencí) a dají se vygenerovat nezávisle na kolekci.

Na Obr. 4 vidíme několik nejdůležitějších bázových obrázků – ty můžeme interpretovat následovně. První bázový obrázek tvoří průměr jasu v celé kolekci (průměrný jas budovy), další báze reprezentují hrubé tvary (obrysy budov, rozhraní budov a pozadí) a báze s nejnižšími singulárními čísly σ_i (resp. s vyšším pořadím i) postupně popisují čím dál méně frekventované detaily v různých obrázcích (např. konkrétní tvary oken a dveří na budovách).

Rekonstrukce obrázků. Abychom získali vizuální představu o tom, jak velké (resp. malé) k je ještě dostačující pro popis obsahu obrázku, můžeme obrázky zpětně rekonstruovat tak, že pro dané k provedeme rozklad a matici A rekonstruujeme jako $A \approx U_k \Sigma_k V_k^T$.

Na Obr. 5a vidíme původní obrázky, resp. obrázky, které byly rekonstruované plným rozkladem $A = U \Sigma V^T$ (kde k je rovno hodnosti matice U^T). Naopak, pro velmi malá k je rekonstrukce velmi nedokonalá, viz Obr. 5b, kde $k = 15$. Přesto lze i z takto hrubě reprezentovaných obrázků rozeznat tvary původních budov. Uvědomme si, že rekonstruovaný obrázek je kombinací pouze 15 bázových ob-



Obr. 5. (a) Původní obrázky. Rekonstrukce obrázků z aproximace $A \approx U_k \Sigma_k V_k^T$ pro (b) $k = 15$, (c) $k = 50$, (d) $k = 250$.

rázků, tj. namísto 4800 hodnot jasu pixelů je potřeba pouze 15 "vah" konceptů! V případě $k = 50$ (viz Obr. 5c) je rekonstrukce dokonalejší, lze rozeznat přítomnost některých hrubých detailů, např. oken. Pro $k = 250$ je již rekonstrukce na takové úrovni, že původní budovy lze jednoznačně rozpoznat (viz Obr. 5d).

Z uvedeného příkladu lze usoudit, že pokud budeme chtít vyhledávat v kolekci obrázků podle hrubých obrysů, bude výhodnější použít pouze několik málo souřadnic vektorů pseudo-obrázků, naopak v případě, že budeme vyhledávat objekty také podle detailů, zvolíme větší počet souřadnic, tj. vyšší k.

Poznámka: Z příkladu lze rovněž usoudit, že celá metoda by mohla být využita pro kompresi celých kolekcí obrázků – výhodou oproti kompresi JPEG, resp. použití DCT, jsou bázové obrázky "šité na míru" dané kolekci (ne pouhé kombinace kosinů), což by se mělo projevit menším počtem bázových obrázků potřebných pro dostatečně kvalitní rekonstrukci. Nevýhodou je fakt, že spolu s vektory pseudo-obrázků potřebujeme uchovávat i bázové obrázky kolekce.

2.3 Funkce podobnosti pro porovnávání obrázků

Po provedení SVD rozkladu obdržíme kolekci vektorů pseudo-obrázků, které je potřeba nějak porovnávat s vektorem pseudo-dotazu (vzniklým transformací dotazového obrázku). Obecně se v oblasti vyhledávání v obrazových kolekcích používá mnoho nejrůznějších podobnostních měr, některé z nich nemají ani analytické vyjádření (např. ohodnocení natrénovanou neuronovou sítí).

V případě reprezentace obrázků histogramy barev/jasů se pro jejich porovnávání často používá kvadratická forma [8, 14], ve které je možné příbuznost

barev zachytit korelačními váhami mezi jednotlivými barvami (obecně souřadnicemi vektoru). Kvadratická forma je vlastně zobecněním Euklidovské vzdálenosti (kde neexistují korelace mezi souřadnicemi, všechny dimenze jsou považovány za nezávislé).

Jelikož v případě modelování obrázku vektorem pseudo-obrázku jsou souřadnice nekorelované (souřadnice obsahují váhy vektorů konceptů, které tvoří bázi a tudíž jsou lineárně nezávislé), postačí, když použijeme pro porovnání vektorů obyčejnou Euklidovskou vzdálenost, nebo jinou Minkowského metriku (L_p metriku). Podobnost modelovaná vzdáleností je interpretována tak, že čím vzdálenější jsou oba vektory, tím méně podobné jsou si obrázky. Nulovou vzdálenost mají identické obrázky (resp. jejich vektory). Výhoda Euklidovské vzdálenosti spočívá také v tom, že je to metrika, a proto lze celou kolekci obrázků (jejich vektorovou reprezentaci) indexovat pomocí metrických přístupových metod (*metric access methods* [4]), což posléze umožňuje v kolekci rychle vyhledávat. Rychlost vyhledávání je dána dvěma typy nákladů spořebovaných během vyhodnocování dotazu. Je to jednak počet diskových přístupů a jednak počet aplikací dané metriky (množství výpočtů vzdáleností).

3 M-strom

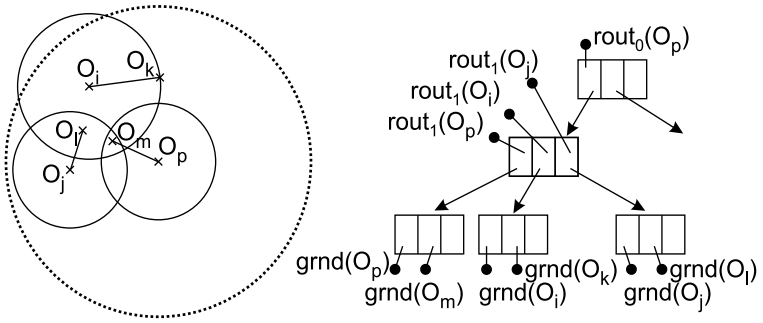
Jednou z metrických přístupových metod je M-strom [5, 9, 16], umožňující indexovat kolekci dokumentů modelovaných objekty metrického prostoru $\mathcal{M} = (\mathbb{U}, d)$, kde $\mathbb{U}$ je univerzum objektů (např. vektorů dokumentů) a d je metrika. Podobně jako mnoha struktur v jiných oblastech indexování, i struktura M-stromu je založena na myšlence B^+-stromu, tj. je vyvážená, dynamická a stránkovaná (umožňující efektivní perzistenci). Konkrétní index M-stromu představuje hierarchii metrických regionů (každý uzel představuje jeden region), respektive hierarchii shluků objektů v těchto regionech.

3.1 Struktura

Listy M-stromu obsahují záznamy $grnd(O_i)$ (ground entries) samotných indexovaných objektů O_i, zatímco vnitřní uzly obsahují tzv. *směrovací záznamy* $rout(O_j)$ (routing entries). Směrovací záznamy popisují tzv. *metrické regiony*, které vymezují v metrickém prostoru oblast, v níž se nacházejí objekty uložené v listech příslušného podstromu. Metrický region je popsán hyper-koulí se středem v nějakém objektu a příslušným pokrývajícím poloměrem (covering radius). Příklad hierarchie metrických regionů (pro Euklidovskou vzdálenost a 2D prostor) a příslušného M-stromu je uveden na Obr. 6.

3.2 Vyhodnocování dotazů na podobnost

Indexování objektů v M-stromu je realizováno pouze pomocí dané metriky d. Díky tomu je snadné implementovat dva základní typy dotazů na podobnost. Je to jednak *rozsahový* dotaz, který slouží k nalezení takových objektů, pro které



Obr. 6. Hierarchie regionů v metrickém prostoru a příslušný M-strom.

je vzdálenost od objektu dotazu menší než daná prahová hodnota a dále dotaz na h *nejbližších sousedů*[1] (h-NN dotaz), kterým získáme prvních h nejméně vzdálených objektů od objektu dotazu.

Vyšší efektivita (rychlost) vyhledávání v M-stromu (vůči např. prostému sekvenčnímu průchodu množiny objektů) spočívá v postupném odfiltrování těch větví M-stromu, které obsahují (vzhledem k dotazu) irelevantní objekty. Korektnost filtrování zaručuje zejména axiom trojúhelníkové nerovnosti metriky, což je klíčová vlastnost využívaná všemi metrickými přístupovými metodami.

3.3 Použití M-stromu pro indexování obrázků

Do M-stromu lze zaindexovat libovolnou kolekci reprezentovanou objekty v metrickém prostoru, můžeme jej tedy použít k indexování vektorů pseudo-obrázků podle vzdálenosti – zvolili jsme Euklidovskou vzdálenost. Podobnostní vyhledávání v kolekci obrázků provádíme pomocí rozsahových dotazů nebo dotazů na h nejbližších sousedů.

4 Experimenty

S výchozí kolekcí 730 obrázků budov jsme provedli dva druhy experimentů. Z hlediska přesnosti a úplnosti (precision and recall) jsme zkoumali úspěšnost samotné metody extrakce vlastností z obrázků pomocí LSI, z hlediska efektivity jsme pak testovali náklady na vyhodnocení dotazu pomocí M-stromu.

4.1 Kvalita vyhledávání

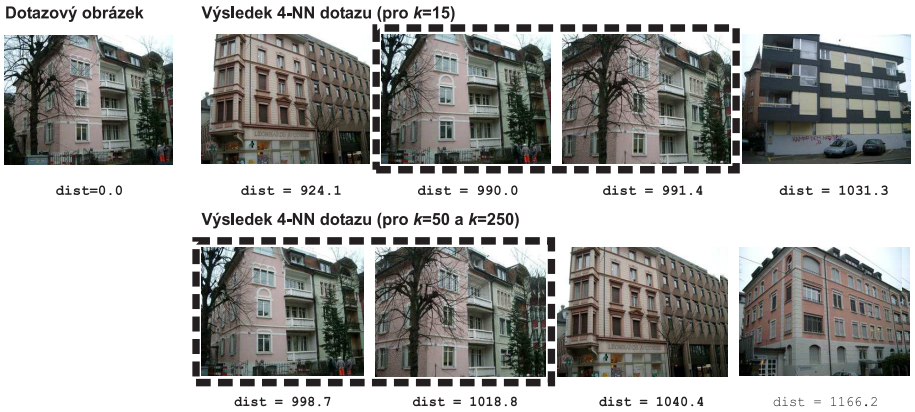
Vzhledem k tomu, že kolekce byla složena ze skupin po pěti obrázcích zobrazujících stejnou budovu z různých pohledů (viz Obr. 7), rozhodli jsme se pro úlohu

[1] Obvykle se uvádí parametr k, tj. dotaz na k nejbl. sousedů (k-NN dotaz). Parametr k jsme již zavedli u SVD, proto u dotazů budeme používat h, aby nedošlo k záměně.

identifikace budovy (jedna z nejtěžších úloh vyhledávání podle podobnosti). Zaindexovaná byla samozřejmě celá kolekce 730 obrázků, členění na skupiny uvnitř kolekce bylo pouze logické.

Obr. 7. Skupina obrázků téže budovy z různých pohledů.

Testování probíhalo tak, že jsme náhodně vybrali 50 skupin obrázků, přičemž z každé skupiny jsme dále náhodně vybrali jeden obrázek, položili 4-NN dotaz k tomuto obrázku a očekávávali, zda ve výsledku dotazu obdržíme zbývající 4 obrázky z dané skupiny. Pokud byly nalezeny všechny 4 obrázky, byla přesnost odpovědi 100%, pro 3 správně nalezené obrázky byla přesnost 75%, atd. Ačkoliv pořadí obrázků v odpovědi určuje míru podobnosti k dotazovému obrázku, pro stanovení přesnosti nám na pořadí nezáleželo vzhledem k malému počtu obrázků v odpovědi. Úplnost odpovědi v tomto případě odpovídala přesnosti, neboť velikost odpovědi se rovnala počtu relevantních obrázků v kolekci, tj. 4.

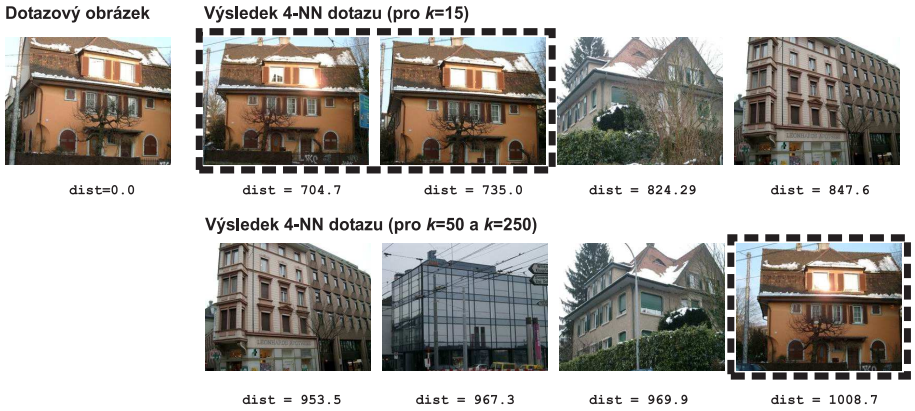


Obr. 8. Odpověd na první dotaz.

Na Obr. 8 vidíme příklad dotazu a odpovědí pro různá k (tj. pro popis obrázků pomocí k bázových obrázků). Přesnost odpovědí byla 50%, pro $k = 50$ a $k = 250$ navíc hledaná budova obsadila první dvě místa v odpovědi.

Na Obr. 9 vidíme 4-NN dotaz pro jiný obrázek. Přesnost odpovědí byla pro $k = 15$ opět 50%, pro $k = 50$ a $k = 250$ pouze 25%. Tento výsledek je zdánlivě v rozporu s faktem, že pro vyšší k získáme přesnější rekonstrukci obrázků, a tedy

by měla být i vyšší přesnost vyhledávání. Taková představa je ale mylná, protože, jak jsme podotkli v kap. 2.2, důležitější (frekventované) koncepty reprezentují hrubé tvary, kdežto méně důležité (tj. méně frekventované) koncepty popisují různé detaily. V uvedeném příkladu zřejmě došlo k situaci, kdy dotazování na příliš velké detaily dotazového obrázku použitím vyššího k způsobilo vyhledání irelevantních obrázků, které měly (ačkoliv to není přímo vizuálně patrné v samotných obrázcích) vyšší zastoupení detailů přítomných v obrázku dotazu. Naopak, relevantní obrázek (při vyhledání zařazený až na 4. místo) měl díky jinému pohledu na budovu poněkud odlišné detaily.



Obr. 9. Odpověď na druhý dotaz.

Zhodnocení. Pro 50 4-NN dotazů bylo dosaženo průměrné přesnosti 43% (tj. v každé odpovědi byly v průměru 1–2 správné obrázky), přičemž jako nejvýhodnější se ukázalo použití $k = 15$. Vzhledem k tomu, že jednotlivé obrázky v rámci skupin představovaly budovu z různých pohledů, často i velmi odlišných, lze považovat přesnost 43% jako velmi dobrou. Z toho lze také usoudit, že uvedená metoda LSI pro obrázky je poměrně odolná vůči prostorovým transformacím. Pokud bychom ještě zmírnili identifikační požadavek tak, že stačí, aby se ve 4 vyhledaných obrázcích vyskytoval alespoň jeden správný, dosáhli bychom identifikace téměř každé budovy.

4.2 Rychlost vyhledávání

V druhé skupině experimentů jsme testovali náklady na vyhledávání obrázků pomocí indexů M-stromu. Pro parametry $k = 15, k = 50, k = 250$, tj. pro každou sadu k-rozměrných vektorů, byl sestaven jeden index. Konstrukce probíhala metodou `MinMax + MultiWay`, detaily viz [16, 15]. Výsledky efektivity dotazování jsou uvedeny v Tabulce 1. Kapacita uzlů M-stromu byla stanovena na 10

objektů, přičemž průměrná naplněnost uzlů (v tabulce označeno jako UTIL) dosahovala cca 70%.

Při dotazování jsme sledovali počet diskových přístupů k uzlům (v tabulce označeno jako I/O) a počet aplikací metriky (v tabulce označeno COMP). Náklady jsou vyjádřeny procentuelně jako (a) podíl přístupů/aplikací vzhledem k velikosti indexu M-stromu, resp. počtu záznamů v indexu M-stromu (b) podíl přístupů/aplikací vzhledem k velikosti sekvenčního souboru, resp. počtu obrázků. Sledovány byly průměrné, maximální a minimální hodnoty nákladů (bylo provedeno 50 4-NN dotazů).

Tabulka 1. Efektivita vyhledávání v M-stromu.

Parametr	UTIL	4-NN dotazy											
k	%	I/O			COMP			I/O			COMP		
		% M-stromu			% M-stromu			% sekv.			% sekv.		
		Avg	*Min*	*Max*	*Avg*	*Min*	*Max*	*Avg*	*Min*	*Max*	*Avg*	*Min*	*Max*
$k = 15$	68	**54**	17	76	**41**	11	58	**93**	29	131	**48**	13	68
$k = 50$	71	**63**	29	88	**49**	20	74	**104**	48	145	**58**	24	88
$k = 250$	70	**65**	24	91	**51**	16	75	**112**	41	157	**60**	19	88

Zhodnocení. Z tabulky můžeme usoudit, že nejlepších výsledků bylo dosaženo pro $k = 15$, kdy bylo potřeba načíst polovinu indexu M-stromu, tj. ekvivalent 93% sekvenčního souboru. Počet aplikací metriky byl nižší, v přepočtu 48% aplikací nutných pro sekvenční vyhodnocení. Uvedené výsledky nejsou zcela přesvědčivé z hlediska výhodnosti M-stromu oproti sekvenčnímu zpracování dotazu, nicméně je třeba podotknout, že experimenty byly prováděny na velmi malé kolekci (pouze 730 obrázků). V budoucnu bychom chtěli metodu vyzkoušet na kolekcích o deseti- až statisících obrázků, kde by se měl potenciál M-stromu projevit v podstatně větší míře.

5 Závěr

V tomto příspěvku jsme popsali specifika použití modelu LSI pro reprezentaci obrázků spolu s využitím M-stromu jako indexovací struktury pro efektivní vyhledávání. Díky aplikaci LSI na obrázky jsme také vizualizovali některé aspekty LSI, které nejsou při klasickém použití ve vektorovém modelu DIS přímo patrné.

Tento výzkum je částečně podporován grantem GAČR 201/05/P036.

Reference

1. R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., 1999.
2. M. W. Berry and M. Browne. *Understanding search engines: mathematical modeling and text retrieval.* Society for Industrial and Applied Mathematics, 1999.

3. C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.
4. E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Compututing Surveys*, 33(3):273–321, 2001.
5. P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of the 23rd Athens Intern. Conf. on VLDB*, pages 426–435. Morgan Kaufmann, 1997.
6. S. Deb. *Multimedia Systems and Content-Based Image Retrieval*. Information Science Publishing, 2003.
7. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
8. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems (JIIS)*, 3(3/4):231–262, 1994.
9. M. Patella. *Similarity Search in Multimedia Databases*. PhD thesis, Dipartmento di Elettronica Informatica e Sistemistica, Bologna, `http://www-db.deis.unibo.it/Mtree/index.html`, 1999.
10. J. Pokorný, V. Snášel, and D. Húsek. *Dokumentografické informační systémy*. Karolinum, Praha, 1998.
11. P. Praks, J. Dvorský, and V. Snášel. Latent Semantic Indexing for Image Retrieval Systems. In *SIAM Conference on Applied Linear Algebra (LA03), The College of William and Mary, Williamsburg, USA*, 2003.
12. P. Praks, L. Machala, and V. Snášel. Iris Recognition Using the SVD-Free Latent Semantic Indexing. In *Workshop on Multimedia Data Mining (MDM/KDD 2004), Seattle, WA, USA*, 2004.
13. Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.
14. T. Seidl and H.-P. Kriegel. Efficient user-adaptable similarity search in large multimedia databases. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)*, pages 506–515, 1997.
15. T. Skopal. *Metric Indexing in Information Retrieval*. PhD thesis, Technical University of Ostrava, `http://urtax.ms.mff.cuni.cz/~skopal/phd/thesis.pdf`, 2004.
16. T. Skopal, J. Pokorný, M. Krátký, and V. Snášel. Revisiting M-tree Building Principles. In *Proceedings of the 7th East-European conference on Advances in Databases and Informations Systems (ADBIS), LNCS 2798, Springer-Verlag, Dresden, Germany*, pages 148–162, 2003.
17. H. Tamura, S. Mori, and T. Yamawaki. Textual features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6), 1978.

Annotation:

An Application of LSI and M-tree in Image Retrieval

In this paper we present a method of LSI-based image feature extraction and, simultaneously, we demonstrate how such an extraction can be used for image retrieval using the M-tree. Moreover, we show several interesting properties, which are not directly visible when using the LSI on text collections.