Semestrální práce na IZI425 – Teorie kódování a šifrování

JPEG komprese

Borek Bernard, leden 2004

1 Úvod do problematiky

1.1 Problémy uchovávání nestrukturovaných dat

Digitální záznam byl primárně určen pro uchovávání textu, popřípadě jiných dobře strukturovaných informací. Není problém uložit mnohaset stránkovou knihu do několika málo kilobajtů stejně tak jako není problém mít v databázi uloženo např. celé účetnictví nějaké firmy. Časem se logicky vyvinula potřeba digitálně uchovávat i data zcela jiného ražení, a to data obrazová a zvuková, která jsou oproti strukturovaným datům nebo textům zásadně odlišná svou rozmanitostí a mnohdy také formální nepopsatelností. Jestliže například jednoznačně víme, jaké znaky se můžou v textu vyskytnout (i když jich je kvůli asijským znakům hodně), např. u fotografických dat o jejich struktuře předem nevíme vůbec nic. Právě o tom, jak jsou statická obrazová data uchovávána, bude tato práce. Důraz bude kladen na formát JPEG a na některé nové nastupující technologie.

1.2 Dva přístupy uchovávání obrazových dat

Existují dva zcela odlišné způsoby, jak lze obrazová data uchovávat. První způsob je vektorový, který obrazová data kóduje do matematických vzorců, které jsou při vykreslování na obrazovku nebo na jiné výstupní zařízení opět nějakým programem interpretovány. Tento způsob je velmi efektivní, protože např. čtverec vyplněný zelenou barvou o velikosti 1000 x 1000 pixelů lze jednoduše zakódovat např. jako počátek=(50,50) AND strana=(1000) AND výplň=(RGB(00FF00)). Tato data zaberou pár bajtů a také interpretovány budou velmi rychle.

Druhý způsob uchovává obraz bitmapově, tedy skutečně jako mapu jednotlivých pixelů. Na záznamové médium musí být zaznamenána informace o každém jednotlivém pixelu obrazu, takže třeba náš zelený čtverec bude tentokrát kódován jako pozice=(1,1)+barva=(RGB(00FF00)) AND pozice=(1,2)+barva=(RGB(00FF00)) AND … AND pozice=(1000,1000)+barva=(RGB(00FF00)). I když se zatím nebudeme zajímat o to, jak je paměťově náročné uložit informaci o jednom pixelu, už nyní musí být jasné, že uložit 1 000 000 hodnot není možno s úsporností vektorového záznamu ani vzdáleně porovnávat.

Vektorový způsob kódování je bohužel možno použít jen na speciální obrazová data, zatímco bitmapový přístup lze uplatnit na cokoliv. V praxi se navíc ve většině případů budeme setkávat s bitmapovými daty, protože koneckonců, náš svět je bitmapový.

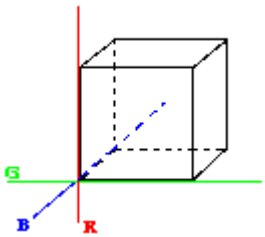
Ještě je důležité poznamenat, že volba způsobu kódování je na člověkovi. Proto někdy můžeme narazit na text v PDF uložený bitmapově, k čemuž opravdu není žádný rozumný důvod.

1.3 Kódování barev

Pro další popis bude důležité vědět, jak jsou v digitálním světě kódovány barvy. Existuje několik barevným systémů (nebo též barevných prostorů), přičemž pro nás bude základem systém RGB.

Písmena RGB jsou zkratkou anglických názvů barev Red (červená), Green (zelená) a Blue (modrá). Pomocí těchto tří barev jsou potom složeny úplně všechny další barevné odstíny, přičemž hodnota barvy se obvykle zapisuje jako uspořádaná trojice,

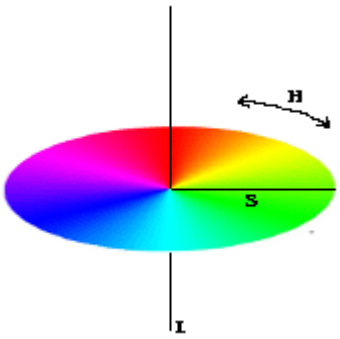
nejčastěji v desítkové nebo v šestnáctkové soustavě. Tento model je subtraktivní, což znamená, že skládání barev postupně výsledek zesvětluje. Budeme-li tedy mít stupnici 0 až 255, kde 0 znamená žádný podíl dané barvy a 255 naopak plný podíl, barva (0,0,0) bude černá (žádna barva v ní nemá svůj podíl) a (255,255,255) bude bílá (všechny barvy „svítí" naplno). Opakem tohoto modelu je například model CMYK (Cyan, Magenta, Yellow a blacK), který používají tiskárny – čím víc barvy na papír nanesou, tím je výsledek tmavší.



Systém RGB lze tedy vyjádřit jako krychli.

Hodnoty 0 až 255 (hexadecimálně 00 až FF) nebyly vybrány náhodně, každá složka barvy se skutečně obvykle kóduje 8 bity ($2^8 = 256$), což znamená, že jeden pixel zabírá 24 bitů neboli 3 bajty. Nyní si lze jednoduše udělat představu o tom, jak je takový nekomprimovaný čtverec o straně 1000 pixelů v paměti velký – zabírá zhruba 3 MB. Pokud máme tedy uchovávat fotku o velikosti 10 na 15 cm při rozlišení 600 pixelů na palec (běžná jednotka rozlišení tiskáren, jinak řečeno 240 pixelů na centimetr), potřebujeme uchovávat obrázek o rozměrech (10 * 240) * (15 * 240) = 8 640 000 pixelů, tj. zhruba 26 MB. To je i na dnešní velkokapacitní harddisky příliš velké číslo, které je nutno řešit kompresí.

Než se jí však budeme věnovat, uvedeme ještě jeden často používaný barevný model, model HSB (ze slov Hue – odstín, Saturation – nasycení a Brightnes – světlost. Ten nevypadá jako krychle, ale jako válec, kde svislou osou je světlost (B), kolmou osou je nasycení (S) a „obvodovou" osou je odstín. Model HSB je jen jeden z modelů, které zohledňují osu světelnosti, a je uveden především pro jeho názornost. V praxi se častěji používá model YCbCr (YUV), kde první písmeno Y je tzv. luminance a zbývající dvě jsou tzv. chrominance, tedy komponenty, které popisují barvu.



2 Komprese

Jak bylo naznačeno v předchozí podkapitole, je skladování obrazových dat v nekomprimované podobě velmi náročné na prostor, nehledě na to, že ještě před pár lety si podobný „luxus" počítače, respektive jejich pevné disky vůbec nemohly dovolit. Myšlenka komprese je proto logicky už poměrně starou záležitostí.

2.1 Bezztrátová komprese

Rozlišujeme komprese dvou druhů. Tím prvním je bezztrátová komprese, která při snížení velikosti zachovává 100% informací. Používány jsou podobné algoritmy jako např. při kompresi ZIPem. Mezi nejznámější bezztrátové formáty patří GIF, který se rozmohl především díky internetu. Navíc velikost dat velmi účinně snižoval používáním tzv. barevné palety, kde každý soubor s sebou nesl informaci o tom, jakých barevných hodnot nabývají jeho obrazové body a jeden pixel potom nebylo nutné kódovat 24 bity, ale bylo-li barev například jen 8, stačili bity 3. GIF se proto velmi hodil např. pro loga používaná na webových stránkách nebo na jiné

jednoduché grafické prvky. Jeho popularitu také umocňovala možnost definovat jednu barvu, která bude „průhledná", takže nebylo problém umisťovat nepravidelná loga na barevná pozadí webových stránek nebo třeba psaných dokumentů.

O GIFu jsem mluvil v minulém čase, protože se jeho autoři rozhodli tento formát licencovat. To vedlo k nástupu velmi podobného formátu jménem PNG (Portable Network Graphics), který už podporoval i paletu true color a alfa kanál, jakési vylepšení průhlednosti. I tento formát se však stále hodí spíše jen pro jednoduché obrazy a pro komprimování „reálných" obrazů je jeho kompresní poměr stále nedostatečný.

2.2 Ztrátová komprese

Obecně řečeno se jedná o takovou kompresi, kde po dekomprimování získáme méně než 100% původních dat, neboli se při kompresi určitá část obrazové informace ztratí. Ztrátové kompresní algoritmy jsou velmi důmyslné a využívají nedokonalosti lidského zraku. Díky tomu mohou dosahovat vysokých kompresních poměrů (kompresní poměr je výraz typu 20 : 1, kde 20 je velikost nekomprimovaných dat a 1 je velikost dat komprimovaných – zde tedy např. kompresní poměr 20 : 1 znamená dvacetinásobné zmenšení velikosti souboru oproti obyčejné nekomprimované bitmapě).

Zaměříme se především na JPEG, na závěr ale zabrousíme také mezi sofistikovanější a modernější formáty.

3 JPEG

Jméno JPEG vychází z názvu konsorcia Join Photographic Experts Group, které tento standard vyvinulo a v roce 1988 zavedlo do praxe. JPEG se potom především díky nástupu Internetu stal tak neuvěřitelně rozšířeným, že i v dnešní době, kdy už existují vyspělejší formáty, se stále používá v téměř 100% případů.

3.1 JPEG komprese krok za krokem

JPEG je založen na poměrně vysoké matematice, proto se pokusíme vystihnout to nejdůležitější a popsat celý kódovací proces.

Krok 1: Transformace do vhodného systému barev

Jak již bylo řečeno, ztrátové kompresní algoritmy (a JPEG zvlášť) vycházejí z nedokonalostí lidského zraku. Bylo experimentálně zjištěno, že lidské oko je velmi citlivé na světlost, ale výrazně méně na barevné odstíny (možná to neplatí o některých ženách, které svým mužům láskyplně vysvětlují, že jejich svetr není zelený, ale pistáciově tyrkysový). Přeneseme-li se tedy do barevného modelu HSB, můžeme říci, že lidské oko je výrazně citlivější na svislou osu B než na osy ostatní (H a S). Prvním krokem je tedy transformace barev z RGB do nějakého modelu zohledňujícího světelnost (HSB, YUV, …). Tato transformace sice není jednoznačná, ale zaokrouhlovací změny jsou minimální.

Krok 2: Zjednodušení os H a S

1	2	3	4	5	6	7	8	9	10
1		3		5		7		9	

Tato jednoduchá tabulka demonstruje, jak se lze jednoduše a elegantně zbavit zhruba 50% dat. Lidské oko nemá šanci toto zjednodušení postřehnout (na ose B by to ovšem poznalo velmi rychle).

Krok 3: Rozdělení do bloků 8x8

Celý obraz je rozdělen do čtverečků 8x8 pixelů. 8x8 pravděpodobně proto, že jak 8 tak 64 jsou mocninami dvojky a s těmito čísly si počítače „rozumí", což bylo v 80. letech dost podstatné kvůli rychlému výpočtu.

Krok 4: DCT (diskrétní kosinová transformace)

V tomto kroku je potřeba převést údaje signálu (hodnoty HSB) na frekvenční údaje. Skutečné hodnoty jednotlivých políček VŠECH bloku lze totiž popsat obecnou funkcí, takže cílem tohoto kroku je pouze spočítat koeficienty této funkce. Potom bude stačit přenášet pouze je a původní blok půjde vždy dopočítat. Zatím jsme toho moc neušetřili (koeficientů je pořád 64), ale data teď mají jiný význam – bod o souřadnicích (0,0) představuje průměrnou barvu celého bloku a ostatní body nesou informaci o tom, jak se odstín mění.

Krok 5: Komprese

V předchozím kroku jsme z každého bloku získali 64 frekvenčních koeficientů, které se nyní vydělí kvantizačními koeficienty z kvantizační matice, která je sestavena empiricky. A jak je tato komprese účinná? Každý z koeficientů z kroku 4 se pohybuje v rozmezí -1024 až 1023, což odpovídá zhruba 11 bitům paměti. Nejmenší koeficient z kvantizační tabulky je kolem 10, takže největší přenášený koeficient bude zabírat zhruba 8 bitů (ostatní samozřejmě stejně nebo ještě méně). Síla komprese se nastavuje právě v tomto kroku úpravou kvantizační matice.

Krok 6: Bezztrátová komprese

Posledním krokem je normální bezztrátová komprese, aby se docílilo nejmenší možné velikosti výsledných dat.

3.2 Výhody a nevýhody JPEGu

JPEG je nepochybně velmi dobrým formátem a i při slušné kompresi ještě lidský zrak nemusí nic poznat. Nicméně technologie JPEGu je už poměrně stará a tak například nepodporuje 48 bitovou barevnou hloubku nebo obrázky větší než 64000 x 64000 pixelů.

4 Další ztrátové kompresní algoritmy

Doba jde dál a objevují se alternativní kompresní algoritmy. My se krátce podíváme na dva nejnadějnější: JPEG 2000 a fraktálová komprese.

4.1 JPEG 2000

Na tomto formátu pracuje skupina JPEG od roku 1995 a první verze světlo světa spatřila skutečně roku 2000. Vyjmenujeme si hlavní rysy tohoto formátu:

- je založen na vlkové (wavelet) transformaci
- lepší, rychlejší a kvalitnější komprese (asi o 20 až 30% lepší kompresní poměr než JPEG)
- odolnost proti chybám datového toku
- kvalitní zpracování počítačové grafiky, která na rozdíl od přirozeného světa obsahuje mnoho ostrých přechodů (kontrastní přechody jsou velkou slabinou JPEGu)
- progresivní transmise, což je něco jako postupné zobrazování od nejnižší kvality až po nejvyšší
- dobrá podpora metadat

Mezi další zajímavé vlastnosti patří velmi příjemná práce s obrázkem v browseru, pokud je nainstalovaný příslušný plugin (otáčení, zoomování), možnost zobrazení v nižším než nejvyšším rozlišení tento formát rovněž pasuje do role favorita na internetu. Zatím je však JPEG natolik zaběhnutý, že bude pro JPEG 2000, který je navíc zčásti licencovaný, asi poměrně obtížné prosadit se.

4.2 Fraktálová komprese

Toto je nesmírně zajímavý algoritmus, který vychází z teoretických poznatků, které tvrdí, že obrazce reálného života se periodicky opakují v různých velikostech. Tato komprese je tedy založená na tom, že v obraze hledají kousky obrazu samotného. Později byl algoritmus zefektivněn tak, že se nehledá přímo obraz celého obrazu, ale i jeho částí.

Z tohoto stručného popisu už vyplývají některé zajímavé skutečnosti, jako např. to, že obraz lze nekonečně přibližovat a tím se v obraze logicky od určité úrovně objevují detaily, které v originále nebyly. Příjemnou vlastností této komprese je fakt, že se nikdy nesetkáme s klasickými čtvercovými artefakty z JPEGu, protože fraktály mají daleko přirozenější tvary.

Budoucnost této zajímavé komprese je trochu sporná, protože ačkoliv se na kompresi např. fotek velmi hodí, pro tradiční geometrii je jen těžko použitelná. Navíc i u oněch fotek je lepší než JPEG až při vyšších kompresních poměrech. Posledním, ale zdaleka ne nejméně významným břemenem je licencování této technologie.

5 Závěr

JPEG je i přes své relativní stáří stále velmi dobrým formátem pro uchovávání dat. Jeho velkou „výhodou" je téměř dokonalá podpora na mnoha zařízeních a na mnoha platformách. Jeho největší slabinou ani tak nejsou některá technologická omezení, ale spíše fakt, že JPEG příliš dobře nezvládá kompresi jednoduchých geometrických obrazů. Můžeme proto čekat, že bude postupem času nahrazen nejspíše nějakou formou standardu JPEG 2000.

Obsah

Zdroje

[1] Do hlubin JPEG, http://www.builder.cz/art/homepage/dohlubinjpg.html

[2] JPEG pro každého, http://www.grafika.cz/art/webdesign/clanek562289929.html

[3] JPEG2000: revoluční formát pro kompresi obrázků?, http://www.grafika.cz/art/polygrafie/jpeg2000.html

[4] Kompresní formáty, http://www.grafika.cz/art/webdesign/clanek1140805886.html

[5] Fraktální komprese, http://www.prikryl.cz/cze/htmlseminarka.php?id=fraktal