

LOSSLESS IMAGE COMPRESSION

Jaroslav Němec

Bachelor Degree Programme (3), FIT VUT

E-mail: xnemec14@stud.fit.vutbr.cz

Supervised by: Bařina David

E-mail: ibarina@fit.vutbr.cz

ABSTRACT

This paper shows how to compress images without losing any information. The compress ratios are in the 2,5 : 1 rates. The paper also describes a suitable color model choosing, how to use and count prediction and code/decode the prediction error.

1. ÚVOD

Důvodů pro použití komprese je mnoho. Jedním z hlavních důvodů je úspora místa při ukládání na disk. Někdo by mohl namítnout, že kapacity dnešních pamětí jsou dostatečně velké. To je sice pravda, ale s kapacitou pamětí roste i velikost uložených dat. Navíc u téměř každého typu dat se setkáváme s nadbytečností. Dalším velmi dobrým důvodem je přenos takových dat po síti. Jistě si dokážeme představit, že nekomprimovaný obrázek se bude déle posílat než obrázek komprimovaný. Co se týká porovnání bezztrátové a ztrátové komprese je zřejmé, že při dekomprimaci u bezztrátově zakódovaných dat můžeme provést kompletní rekonstrukci obrazu. Ztrátová komprese si všímá spíše nedokonalosti lidského sluchu/zraku. Jako první pokus o kompresi je zaznamenáno Braillovo písmo nebo například Morseova abeceda (1844).

2. VOLBA VHODNÉHO BAREVNÉHO MODELU

Nejznámějším barevným modelem je model RGB. Představme si, že k zobrazení jednoho bodu v obrázku (pixelu) je nutné tři hodnot. Tyto hodnoty určují podíl na výsledné barvě. Hodnoty se pohybují v rozmezí 0-255. Jejich složením vznikne bod v obraze o dané barvě. Při zkoumání kompresních metod bylo zjištěno, že tento barevný model není vhodný ke kompresi, protože snižuje kompresní poměr. Proto se při kompresi využívá barevný model YUV[1]. Při každé transformaci RGB barevného modelu na jakýkoliv jiný je nutné, aby byl dokonale reverzibilní. Derivátů barevného modelu YUV je mnoho. V některých se používají výpočty v plovoucí řádové čárce, což může vést k nepřesné zpětné transformaci a samozřejmě ke ztrátě informace. Transformace z RGB do YUV je dána následujícími vztahy:

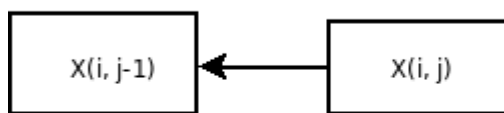
$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (1)$$

$$U = B - Y \quad (2)$$

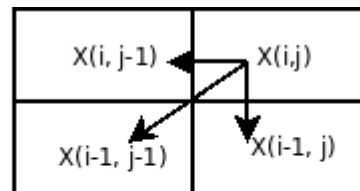
$$V = R - Y \quad (3)$$

3. PREDIKCE HODNOTY PIXELU

Bezeztrátová komprese obrazu využívá statických vlastností obrazu. Sousední body v obraze jsou si velmi často podobné (prostorová korelace). Díky této vlastnosti dokážeme predikovat hodnotu následujícího pixelu ze sousedních hodnot. Naší snahou a ideálním stavem je, kdy dokáže prediktor[2] určit hodnotu naprosto přesně. Rozdíl mezi predikovanou hodnotou a skutečnou se nazývá chyba predikce. Aplikací této predikce získáváme filtr. Můžeme predikovat z hodnoty předešlého pixelu nebo ze sousedních pixelů. Na počátku samotné komprese predikujeme hodnoty a zakódujeme chybu predikce (tedy ne samotné hodnoty pixelů). Při dekompresi opět provádíme predikci hodnoty pixelu a pro rekonstrukci pouze sečteme hodnotu predikce a dekódovanou chybu predikce. Na obrázku jsou vyobrazeny dva možné způsoby určování predikce.



$$P = X(i, j-1)$$



$$P = A \cdot X(i, j-1) + B \cdot X(i-1, j-1) + C \cdot X(i-1, j)$$

Kde A, B, C jsou vhodně zvolené konstanty

4. RLE ALGORITMUS

Jedná se o nejprimitivnější algoritmus komprese (tzv. délka běhu), který vyhledává opakující se sekvence v obraze a kóduje je[3]. To znamená, že vždy uloží daný znak a počet jeho výskytů. Nevýhodou je, že pokud se znaky neopakují, tak dochází ke zvětšení dat. Postup je uveden v následujícím příkladu.

Vstup: AAAAEEERRROOOO

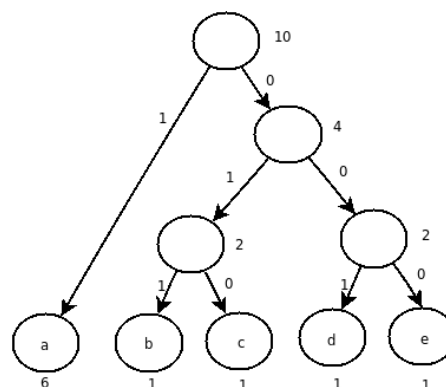
Výstup: [4A][3E][3R][4O]

5. HUFFMANOVO KÓDOVÁNÍ

Na úvod předesílám, že existují dvě varianty Huffmanova kódování – statická a dynamická/adaptivní[4]. Nyní se budu zabývat statickou variantou. Huffmanovo kódování se aplikuje až po užití filtru. Princip toho kódování spočívá v přidělování různě dlouhých prefixních kódů jednotlivým znakům v obraze. Délka těchto kódů se řídí jejich četností v obraze. Čím četnější znak, tím kratší kód. Tyto kódy jsou pak bitově uloženy do souboru. Nejdůležitější je tedy spočítat četnosti jednotlivých znaků v obraze. Z implementačního hlediska vytváříme binární strom. Listy stromu tvoří znaky v obraze (při užití filtru jsou to rozdíly mezi skutečnou hodnotou a predikcí). Výsledný kód, který přiřadíme znaku, je tvořen cestou od kořene stromu k danému znaku. Z toho vyplývá, že při sestavování stromu postupujeme odspodu. Huffmanovo kódování je jedním z nejefektivnějších algoritmů použitých při kompresi. Jeho nevýhodou je nutnost dvou průchodů – při sestavování stromu a při kódování posloupnosti. Další nepříjemností, která není ovšem nijak kritická, je uložení stromu k zakódovaným datům (pro účely dekompresi), což snižuje kompresní poměr. Celý postup je ukázán na následujícím příkladě.

Znak	Výskyt	Pravděpodobnost	Prefixový kód
a	6	60%	1
b	1	10%	011
c	1	10%	010
d	1	10%	001
e	1	10%	000

Tabulka 1: Pravděpodobnostní model



Obrázek 1: Huffmanův strom

6. VÝSLEDKY

Nejzásadnější problém u bezztrátové komprese obrazu je nalezení vhodného barevného modelu a volba nej přesnějšího prediktoru. Testování vlivu prediktoru bylo provedeno na množství obrázků. Testování probíhalo porovnáním kompresního poměru v závislosti na prediktoru. Barvy byly reprezentovány RGB modelem. V tabulce uvádím reprezentativní vzorek dat.

Obrázek	Nekomprimované	Prediktor č.1	Prediktor č.2	Kompresní poměr
č.1	216 kB	169,7 kB	169,5 kB	1,27 / 1,27
č.2	768,1 kB	503,8 kB	482,3 kB	1,52 / 1,59
č.3	1,2 MB	813 kB	801 kB	1,51 / 1,53
č.4	2,7 kB	623 B	638 B	4,43 / 4,33

Označení prediktorů vychází z obrázku v kapitole č.3. U většiny testovacích obrázků zvýšil prediktor č.2 kompresní poměr. U obrázku č.4 prediktor č.2 snížil kompresní poměr oproti poměru s prediktorem č.1. Pokud k tomu dojde, je rozdíl mezi kompresními poměry minimální. Z toho vyplývá, že obecně lepšího kompresního poměru dosáhneme s prediktorem č.2. Velikosti komprimovaných souborů jsou srovnatelné s velikostí obrázku ve formátu PNG.

7. ZÁVĚR

V této práci byl popsán obecný postup při implementaci bezztrátové komprese obrazu. Kapitoly 2-5 ukazují jednotlivé části, které jsou běžně používány. Použitím všech těchto částí dostáváme nejlepší kompresní poměr.

PODĚKOVÁNÍ

Tato práce vznikla částečně za podpory grantu VUT FIT, FIT-S-10-2 a specifického výzkumu MSM0021630528.

LITERATURA

- [1] Lossless and Fine-Granularity Scalable Near-Lossless Color Image Compression, <http://www.win.tue.nl/wic2004/28.pdf>
- [2] Predictores, <http://www.stanford.edu/~mbax/ee392c/node5.html>
- [3] RLE, <http://www.data-compression.info/Algorithms/RLE/index.htm>
- [4] Huffman Coding, http://en.wikipedia.org/wiki/Huffman_coding