

Největší souvislá plocha v RLE komprimovaném obrázku bez dekomprese

Tomáš Krejčí

Září 2012

1 Cíl práce

Cílem práce je navrhnout algoritmus, kterému bude na vstup předán obrázek komprimovaný metodou RLE a jako výstup je očekávána velikost největší souvislé plochy v obrázku.

Souvislou plochou je myšlena taková množina bodů obrázku ve které mají všechny body stejnou barvu a dotýkají se celou hranou, nikoliv pouze rohem.

2 RLE Komprese

RLE komprese je bezztrátová kompresní metoda používaná zejména pro komprimování obrázků s nízkým počtem barev a velkými stejnobarevnými plochami.

Jejím cílem je nahradit posloupnost stejnobarevných bodů mnohem kratší informací o barvě bodu a počtu jeho opakování. Například posloupnost znaků AAAAABCD4DDDDAAA se změní v 5ABC4D3A (Z původních 14 znaků na 8 to znamená kompresní poměr 57%).

3 Algoritmus

Předpokládá se, že vstupní obrázek je zkomprimován nejlépe jak je to jen možné. To zejména znamená, že se nikde nevyskytují dva po sobě jdoucí běhy stejné barvy, které by šli sloučit do jediného.

Celý algoritmus lze rozdělit do dvou částí. Vytvoření grafu a výpočet plochy.

- **Vytvoření grafu.** Principem je každý běh (včetně běhů jednotlivé délky) nahradit odpovídajícím vrcholem grafu, takto vzniklý vrchol ohodnotit délkou běhu ze kterého vznikl a hranou spojit vrcholy spolu sousedící podle deficice souvislé plochy výše. Načtení probíhá následujícím způsobem:

```

while nacti novy beh a oznac ho B do begin
  pridej do grafu vrchol odpovidajici behu B
  nove pridany vrchol oznac v
  for vsechny vrcholy w sousedi v do begin
    if w zhora sousedi s v a maji stejnou barvu then begin
      spoj hranou vrcholy v a w
    end
  end
end
end

```

Jsou celkem čtyři možnosti odkud může jeden běh sousedit s jiným. Z obou stran, zezhora a zespoda.

O strany se nemusíme starat, tam žádný běh se stejnou barvou nebude. Kdyby byl, šely by tyto spojit a vznikl by tak jeden souvislý běh a tím se zvýšil kompresní poměr. To by bylo ve sporu s předpokladem na nejlepší možnou kompresi.

Protože obrázek čteme sekvenčně obzvláště se nám hodí kontrolovat při načítání pouze horní hranu běhu. Dolní hrana bude obstarána dalšími postupně načítanými vrcholy.

- **Výpočet plochy.** V tuto chvíli máme neorientovaný graf s ohodnocenými vrcholy. Cílem je vyšetřit komponentu mající nevyšší součet hodnot všech jejích vrcholů. To je realizováno jednoduchým prohledáním do hloubky a postupným přičítáním hodnot ve vrcholech.

4 Implementace

Jako vstupní soubor byl vybrán formát PCX, který je ve většině případů komprimován metodou RLE.

Vstupní soubor musí:

- být validním RLE obrázkem
- komprimovaný algoritmem RLE
- mít právě osm bitů na pixel

- mít šířku menší než osm nebo mocninu dvou
- mít právě jednu vrstvu

Protože formát PCX podporuje běhy o maximální délce 63 pixelů, je třeba hlídat sousedy i z levé strany běhu nikoliv pouze po jeho horní hraně.

5 Analýza složitosti

Při načítání grafu se algoritmus dotáže na každý běh nejvýše dvakrát. Jednou při jeho samotném načtení a podruhé při načítání hran grafu. Časová složitost načítání je tedy $O(N)$, kde N je počet běhů v daném obrázku.

Při prohlédávání grafu do hloubky je každý vrchol navštíven právě jednou. Časová složitost $O(N)$.

Celkově je tedy algoritmus v lineární časové složitosti vzhledem k počtu běhů obrázku.

Paměti netřeba více než na uložení všech vrcholů grafu. Opět složitost $O(N)$.

6 Závěr

Ve finální implementaci bylo použito mnoho předpokladů na vstupní obrázek. Protože jde o algoritmus pro výpočet největší souvislé plochy, nikoliv však o program mající za úkol implementovat čtečku souborů ve formátu PCX, shledávám tyto předpoklady jako nepříliš omezující a nemají žádný vliv na celkovou funkcionální algoritmu.

Celkově algoritmus běží svižně a nemá problémy i s většími soubory.

Jako další možnosti pro zdokonalení programu se nabízí

- Odstranění předpokladů na vstupní soubor až do případného zobecnění na libovolný PCX-validní soubor.
- Nahrazení rekursivního prohledávání výsledného grafu za prohledávání s pomocí vlastního zásobníku. To by mohlo až řádově zvýšit možnou velikost vstupního souboru. Při použití rekurze je celý algoritmus nejvíce omezen velikostí systémového zásobníku.