Semestrální práce z předmětu APG

Rekonstrukce informace o sousednosti z STL dat

Autor: Jan Bařtipán

bartipan@students.zcu.cz

A03043

Úvod

Ve strojírenském odvětví se při rychlé výrobě prototypů (Rapid prototyping – RP) pro výměnu dat hojně používa formát STL. Jde o čistý seznam trojúhelníků, kde každý vrchol trojuhelníku je reprezentován jeho souřadnicemi. Tato reprezentace má však jednu podstatnou nevýhodu, nenese žádnou informaci o sousednosti trojuhelníků. Naštěstí tuto informaci lze zpětně vytvořit níže popsaným algoritmem.

Implementace algoritmu pro rekonstrukci informace o sousednosti trojuhelníků z STL souboru vyžaduje strukturu, která by udržoval seznam vrcholů a zároveň v tomto seznamu umožňovala ryhclé vyhledávání konkrétního prvku. Jako velmi dobrá struktura se pro tento účel jeví hashovací tabulka. Pokud je dobře navržena hashovací funkce, nalezení konkrétního prvku v tabulce má očekávanou algorimickou složitost O(1)[1].

Hashovací tabulka

V [1] byla navržena tato hashovací funkce:

$$h(x,y,z)=([3\ [|x|Q]/Q+5\ [|y|Q]/Q+7\ [|z|Q]/Q]\,T)\ mod\ T$$

kde x,y,z jsou souřadnice vrcholu,

Q určuje počet desetinných míst pro kvantování (v [1] Q = 1000,0)

T je velikost hashovací tabulky

a mod je operace modulo, tedy zbytek po celočíselném dělení

Tato funkce poskytuje dobré výsledky pro relativně malé objemy dat. Pro větší objem dat (řádově statisíce vrcholů a více) vytváří příliš velké shluky vrcholů na jedné pozici v tabulce.

V [2] je navrženo zlepšení, které vytváří malé shluky vrcholů i pro velké objemy dat. Vylepšená hashovací funkce vypadá následovně:

$$h(x,y,z)=(uint)[(\alpha x+\beta y+\gamma z)\,C+0{,}5]\ mod\ T$$

kde x,y,z jsou souřadnice vrcholu,

α, β, γ jsou koeficienty (v [1] volené jako 3, 5, 7)

C je konstanta, její zvolení je vysvětleno níže

T je velikost hashovací tabulky

operace uint je převod z reálného čísla na celé číslo odtržením části za destinou čárkou, přičtením 0,5 docílíme lepšího rozložení hodnoty (zaokrouhlení)

a mod je operace modulo, tedy zbytek po delení

Konstanta C musí být zvolena tak, aby nedošlo k přetečení mezí neznamenkového integeru. Nazvěme

$$\xi=\alpha x+\beta y+\gamma z$$

adresním prostorem hashovací funkce h(x,y,z). Dále předpokládejme, že

$$x\in\langle 0,X_{max}\rangle, y\in\langle 0,Y_{max}\rangle, z\in\langle 0,Z_{max}\rangle$$

pak

$$\xi_{max}=\alpha X_{max}+\beta Y_{max}+\gamma Z_{max}$$

je maximální hodnota, kterou může adresní prostor získat. Aby nedošlo k přetečení musí pro konstantu C platit:

$$C_1\xi_{max}\le 2^{32}-1$$

Dále musí být konstanta C taková, aby z vypočtené hodnoty ξ byl zachován nejméně stejný počet bitů, který odpovídá velikosti tabulky T. Proto další omezující podmínka pro konstatntu C je:

1 Ve skutečnosti nalezení prvku v tabulce má složitost O(k), kde k je průměrná delka clusteru.

$$C_2 = 2^{32} - 2^k$$

Kde hodnota k je:

$$k = \log_2 T$$

Spojením obou podmínek pro konstantu C dostáváme:

$$C = min\, C_1, C_2$$

Dále se budeme zabývat vhodnou volbou velikosti hashovací tabulky. Hodnota funkce h(x,y,z) se bude počítat velmi často, a je proto důležité, aby její výpočet byl rychlý. Velmi drahé je počítání zbytku po celočíselném dělení. Pokud však zvolíme velikost hashovací tabulky T tak, aby byla mocninou čísla 2, lze operaci modulo T nahradit (rychlejším) bitovým and.
Tedy pro $T = 2^k, k \in \mathbb{N}$ lze operaci X mod T nahradit za operaci X & (T – 1), kde symbol & představuje bitové and.
Velikost tabulky resp. naplnění tabulky α (load factor) nepřímo ovlivňuje délku clusterů. Při zvolení α = 30% až 50% lze očekávat průměrnou délku clusteru 1,5 až 2,5. Počet vrcholů, které budeme zpracovávat, Nv je roven trojnásobku počtu trojuhelníků Nt. Není však třeba vytvářet tabulku velikosti opovídající počtu (duplicitních) vrcholů Nv. Empericky je dokázáno, že jeden (unikátní) vrchol je sdílen průměrně 6-ti trojúhelníky. Výsledný vztah pro velikost hashovací tabulky je tedy:

$$T \geq \frac{N_V}{q_{avg}\,\alpha}$$

tedy:

$$T = \frac{N_V}{6 \cdot 0{,}5} = \frac{3 \cdot N_T}{3} = N_T$$

Rekonstrukce

Algoritmus pro rekonstrukci informace o sousednosti trojúhelníků využívá následujících vlastností datových struktur:

- na jednu pozici v hashovací tabulce H lze uložit více různých vrcholů
- každý vrchol v H si uchovává odkazy na všechny trojúhelníky, které jej sdílí

Může se stát, že vrchol, který sdílejí trojúhelníky A a B je zadán v při definci trojuhelníku (v STL souboru) A jinou hodnotou než při definci trojúhelníku B. Proto při posuzování shodnosti vrcholů je třeba brát v úvahu konstantu ε. Prohlásíme vrcholy V a W za shodné, pokud $\|V - W\| \leq \epsilon$. ε lze zvolit jako polovinu nejkratší hrany. V [4] se však uvádí, že pokud se tak učinní, začnou se při rekonstrukci trojúhelníkové sitě strácet malé trojúhelníky. Proto raději volím ε jako konstantu.
Abych mohl zvolit konstantu C pro hashovací funkci, musím nejdříve jedním průchodem přes všechny vrcholy najít obalovací kvádr. Ze znalosti v absolutní hodnotě největších hodnot jednotlivých souřadnic jsem schopen určit ξ_{max} a tedy i konstatntu C. Teprve druhý průchod rekonstuuje informaci o sousednosti. Algoritmus vypadá následovně:

Pro všechny trojúhelníky $T_i \in T^{(STL)}$:
    Vytvoř strukturu S_i pro trojúhelník T_i
    Pro každý vrchol $A_i \in T_i$
        Pokud je vrchol A_i s ohledem na prahovou hodnotu ϵ již v tabulce,
            ulož příslušnému vrcholu odkaz na trojúhelník T_i
        vlož do tabulky H vrchol A_i a k tomuto vrcholu odkaz na trojúhelník T_i

do S_i ulož odkaz na příslušný vrchol v tabulce H
reprezentaci trojúhelníku S_i přidej do množiny S

Po ukončení algoritmu získáme množinu všech trojúhelníků S s odkazy na jednotlivé vrcholy a dále hashovací tabulku H, ve které jsou uloženy neduplicitní vrcholy s odkazy na trojúhelníky, které tento vrchol sdílejí.
Předzpracování (nalazení obalovacího kvádru) má algoritmickou složitost O(n), kde n je počet všech (i duplicitních) vrcholů. Sám algorimus pro rekonstrukci informace o sousednosti trojúhelníků má složitost O(n k), kde n je opět počet všech (i duplicitních) vrcholů a k je průměrná délka shluku.

Závěr

Předzpracování (nalazení obalovacího kvádru) má algoritmickou složitost O(n), kde n je počet všech (i duplicitních) vrcholů. Sám algorimus pro rekonstrukci informace o sousednosti trojúhelníků má složitost O(n k), kde n je opět počet všech (i duplicitních) vrcholů a k je průměrná délka shluku.
Hashovací tabulka s funkcí navženou výše dává pro testovací soubor „happy budha“ výborné výsledky (velmi krátké shluky – délka maximálního shluku = 5) a přístup k prvku do této tabulky je proto velmi krátký, což je pro zpracování rozsáhlých modelů klíčové.

Použitá literatura:

[1] – Glassner,A.: Building Vertex Normals from an Unsturctured Polygon List, Grahpics Gem IV, str. 60-73, Accademic Press,Inc 1994
[2] – Skala,V., Kuchař,M.: Hash Function for Geometry Reconstruction in Rapid Prototyping, Algoritmy2000 proceedings, Slovakia, str. 379-387, 2000
[3] – Skala,V., Kuchař,M.: The Hash Function and the Principle of Duality, Proceedings of Computer Grahpics International – CGI 2001
[4] – Kuchař,M.: Diplomová práce