

Urychlování algoritmu sledování paprsku metodou dělení prostoru a paralelizací výpočtu

Petr Bláha
BLAHA@KIV.ZCU.CZ

Václav Skala
SKALA@KIV.ZCU.CZ

Západočeská univerzita
Katedra informatiky a výpočetní techniky
Americká 42, Box 314
306 14 Plzeň

1 Úvod

Metoda sledování paprsku je všeobecně velmi dobře známá, avšak jejímu masovému rozšíření brání výpočetní složitost, která může být hrubě odhadnuta jako

$$O(M^2 p 2^k),$$

kde p je počet těles ve scéně,

k je počet štěpení paprsku na propuštěný a odražený,

M je počet pixelů obrazu v jednom směru.

Je zřejmé, že při $M > 512$ se každé urychlení metody sledování paprsku projeví drastickým způsobem. Ke zrychlení výpočtu lze v zásadě použít paralelizace (viz.kap.2) nebo technik využívajících některé urychlující metody (viz.kap.3).

2 Paralelizace metody sledování paprsku s využitím systému PVM

Jeden ze způsobů urychlení metody sledování paprsku je použití paralelního zpracování. Využitým paralelním výpočetním prostředkem je síť pracovních stanic s operačním systémem UNIX. Takovéto využití počítačové sítě umožňuje programový celek **PVM** (Parallel Virtual Machine).

2.1 PVM — Parallel Virtual Machine

PVM je programový celek dostupný formou free-ware na síti Internet, který umožňuje využívat obecně heterogenní síť pracovních stanic s operačním systémem UNIX jako jeden paralelní výpočetní prostředek — virtuální volně vázaný multiprocessor.

Distribuční podoba PVM obsahuje konfigurační soubory pro většinu běžně používaných hardwarových platform (od systému Linux na počítačích PC až po např. superpočítače Cray). Případné odlišnosti jednotlivých architektur, např. v implementaci datových formátů, jsou řešeny v rámci PVM a pro aplikaci běžící na multiprocesoru je tento zcela transparentní — s využitím funkcí knihoven PVM (popsaných dále) může spouštět procesy na jednotlivých uzlech multiprocesoru, zasílat jim zprávy, atd. bez ohledu na to, jakou hardwarovou architekturou je daný uzel ve skutečnosti realizován.

Základní princip činnosti PVM je následující: na každém počítači v konfiguraci virtuálního multiprocesoru musí být instalován a spuštěn komunikační démon PVMD, jednotlivé demony jsou mezi sebou ve spojení prostřednictvím síťových služeb UNIXu a aplikačním programům poskytují množinu funkcí umožňujících provádět na multiprocesoru všechny akce, důležité pro realizaci paralelního výpočtu.

Pro psaní aplikačních programů využívajících multiprocesor jsou k dispozici knihovny pro jazyk C nebo Fortran 77. Tyto knihovny poskytují následující skupiny funkcí:

1. ovládání a zjišťování konfigurace multiprocesoru — přidání, resp. ubrání pracovní stanice z konfigurace, zjištění síťových jmen všech použitých pracovních stanic, atd.
2. spouštění, rušení, zjišťování stavu aplikačních procesů
3. komunikace a synchronizace procesů — posílání zprávy, blokující a neblokující příjem zprávy, sdružování procesů do skupin, multicast a broadcast, atd.

Podrobné informace lze nalézt v [1].

2.2 Využití PVM pro paralelizaci metody sledování paprsku

Pro experiment s paralelizací sledování paprsku byl systém PVM použit v kombinaci s dalším volně šiřitelným programem **PovRay**. Tento poměrně široce používaný program se pro daný účel nabízí hned z několika důvodů — je dostupný ve formě zdrojových textů, umožňuje počítanou scénu rozdělit na samostatně počítané části a navíc existence původní verze pro jednu pracovní stanici umožní průkazné srovnání s paralelizovanou verzí a zjištění dosaženého zrychlení.

Popišme si nyní základní principy realizované paralelizace — použitá metoda je z terminologie paralelního programování známa jako **data partitioning**. V této metodě jsou zpracovávaná data rozdělena na určitý počet samostatně řešených menších částí. Na jejich řešení jsou nasazovány výpočetní procesy, běžící podle stejného programu, jejichž počet je dán počtem dostupných procesorů. Je dokázáno, že data musí být rozdělena na několikrát větší počet částí, než je počet těchto výpočetních procesů, aby se dostatečně projevilo zrychlení přinášené paralelizací.

Pro řízení výpočtu na virtuálním multiprocesoru byl s využitím knihovny PVM napsán program **MASTER**. Jednotlivé úkoly programu MASTER v průběhu výpočtu jsou:

1. **rozdělení vstupních dat** — v našem případě je provedeno s využitím již zmíněného vestavěného rysu programu PovRay umožňujícího dělení počítané scény na menší části. Scéna (ve smyslu výsledné zobrazovací plochy) je tedy rozdělena na horizontální pruhy (skupiny pixelů — skupiny primárních paprsků).
2. **spuštění výpočetních procesů** na jednotlivých pracovních stanicích v multiprocessoru: všechny procesy běží podle stejného programu, kterým je program PovRay mírně modifikovaný tak, aby byl schopen pomocí komunikačních služeb PVM převzít vstupní data a předat vypočtené výsledky programu MASTER.
3. **čekání na dokončení výpočtu** některým výpočetním procesem, převzetí výsledků od tohoto procesu a přidělení mu nové, ještě nezpracované části dat. Speciální případ nastává v závěrečné fázi výpočtu, kdy zbývá méně nezpracovaných částí dat než je počet výpočetních procesů — procesy jsou na nezpracované části dat nasazovány násobně, přičemž použitá strategie je taková, že nejprve se pro každou řešenou část dat procesy zdvojnásobí a teprve poté může být na některou část nasazen třetí proces, atd. Cílem této strategie je obejít potenciální úzké místo algoritmu, kdy jedna extrémně vytižená pracovní stanice by výsledně prodloužila práci celého multiprocessoru.
4. **sestavení dílčích výsledků** po vypočtení všech pruhů (částí dat) do celkového výsledku — v našem případě se jedná o sestavení několika True Color TGA souborů do jednoho velkého (tedy poměrně jednoduchý úkol)
5. **hlídání výpadků** jednotlivých pracovních stanic (výpočetních uzlů) a výpočetních procesů v průběhu výpočtu (možné příčiny — výpadky napájení, práce ostatních uživatelů sítě, atd.) a zajistit náhradní výpočet příslušných částí dat a případné nové spuštění vypadlých stanic či procesů

2.3 Experimentální ověření algoritmu

Výsledky testování navrženého programu na síti pracovních stanic Silicon Graphics Indy pro různé úrovně dělení zpracovávaných dat jsou uvedeny v tab.1.

pruhů \ stanic	2	3	4	5	6
10	3082	1806	1047	609	913
20	3463	1261	754	610	589
40	3401	1181	757	633	591
80	3797	1328	737	722	544
120	4107	1333	848	763	568
240	5600	1617	994	877	719

Tab.1 Doba výpočtu [s] na síti stanic SGI Indy

Je nutné zdůraznit, že použité pracovní stanice byly v průběhu výpočtu zatíženy běžným provozem ostatních uživatelů (zejména CAD systémy).

Tab.2 ukazuje závislost dosahovaného maximálního urychlení na počtu použitých pracovních stanic, tj. procesorů. Lze si všimnout charakteristického průběhu, který je ve shodě s Amdahlovým zákonem omezujícím maximální urychlení.

stanic	2	3	4	5	6
urychlení	1.16	3.05	4.76	5.91	6.62

Tab.2 Závislost urychlení na počtu procesorů

Obdobný experiment byl proveden i pro síť 6 pracovních stanic SUN4 Solaris, které v době výpočtu byly prakticky nezatížené. Výsledky, které jsou, až na absolutní hodnoty naměřených časů, shodné s předchozím experimentem, shrnují tabulky 3 a 4.

pruhů \ stanic	2	3	4	5	6
10	861	583	455	348	615
20	872	534	410	358	375
40	914	576	407	375	383
80	987	609	447	376	387
120	1066	655	476	430	391
240	1315	812	585	499	556

Tab.3 Doba výpočtu [s] na síti stanic SUN4 Solaris

stanic	2	3	4	5	6
urychlení	4.18	6.7	8.84	10.3	9.6

Tab.4 Závislost urychlení na počtu procesorů

3 Metoda dělení prostoru

Metoda dělení prostoru v algoritmech sledování paprsku je všeobecně známá a je velmi často používána v různých modifikacích s rovnoměrným, nerovnoměrným, adaptivním dělením, apod. Uvedené techniky jsou však paměťově velmi náročné. Vzhledem k tomu, že použité objekty jsou většinou malé, existuje možnost modifikace metody dělení prostoru. Princip spočívá v "nařezání" scény do pásů kolmých na vybranou osu. Pro každý pás se ukládá seznam objektů, které daný pás protíná. Při použití ortogonálních pásů pak průnik

těchto množin obsahuje objekty, které mohou incidovat s prostorem určeným jako průnik pásů. Uvedený způsob je zajímavý především tím, že podstatným způsobem minimalizuje paměťové nároky.

Lze ukázat, že paměťová efektivita η metody řezů vůči klasické metodě dělení prostoru v E^3 může být v případě malých objektů odhadnuta jako

$$\eta \doteq 11 \frac{N^2}{p},$$

kde N je počet řezů v jedné ose,

p je počet těles ve scéně.

Kromě uvedeného snížení paměťových nároků je navrhovaná metoda řezů rychlá i pro primární paprsky. Tab.5 ukazuje zrychlení metody řezů vůči testu minmax box, zatímco tab.6 ukazuje zrychlení vůči případu, kdy je použito koule jako ohraničujícího objemu při 256×256 pixelů na průmětně.

Počet objektů	velikost objektů (jeden rozměr boxu)					
	10	16	25	40	63	100
400	10.38	9.52	8.04	6.13	4.74	4.10
630	11.39	10.31	8.55	6.44	4.90	4.22
1000	11.96	10.72	8.83	6.59	4.98	4.28
1600	12.65	11.30	9.22	6.82	5.15	4.44
2500	13.06	11.64	9.48	6.99	5.26	4.55
4000	13.72	12.18	9.86	7.22	5.39	4.51
6300	14.03	12.43	10.03	7.30	5.43	4.65

Tab.5 Zrychlení vůči minmax boxu

Počet objektů	velikost objektů (jeden rozměr boxu)					
	10	16	25	40	63	100
400	26.11	23.40	19.10	13.79	9.86	7.64
630	26.82	25.30	20.29	14.43	10.17	7.82
1000	29.95	26.23	20.88	14.73	10.32	7.92
1600	31.91	27.77	21.87	15.24	10.59	8.11
2500	33.07	28.71	22.53	15.63	10.83	8.29
4000	33.97	29.40	22.96	15.83	10.91	8.35
6300	38.94	33.64	26.24	18.03	12.39	9.48

Tab.6 Zrychlení vůči ohraničující kouli

Podrobný teoretický rozbor metody a experimentální výsledky lze nalézt v [2].

4 Závěr

Jednou z možností urychlení časově náročných výpočtů je použití paralelního zpracování. Byla provedena praktická realizace paralelizace metody sledování paprsku nevyžadující speciální paralelní výpočetní prostředek. Na experimentálních výsledcích bylo ukázáno zrychlení výpočtu dosažitelné uvedeným způsobem. Z těchto výsledků je ovšem zřejmé, že s rostoucím počtem procesorů zrychlení dosažitelné přidáním dalšího procesoru klesá. Proto vedle paralelizace metod a použití paralelních systémů je další alternativou hledání jiných postupů, datových struktur apod. Uvedená aplikace metody dělení prostoru ukazuje na jednu z možností dalšího urychlování metody sledování paprsku.

5 Poděkování

Je milou povinností autorů poděkovat za pomoc a cenné připomínky studentům zaměření Počítačová grafika, zejména pak J.Frischovi a P.Sebránkovi, kteří se též částečně podíleli na realizaci projektu, resp. experimentálním ověření metod.

Literatura

- [1] A.Geist, A.Beuelin, J.Dongarra, W.Jiang, R.Manckel, V.Sunderam. PVM3 Users Guide and Reference Manual. Oak Ridge National Laboratory, September 1994
- [2] Skala,V.: Memory Saving Techniques for Space Subdivision Technique, Machine Graphics and Vision, Vol.2, No.3, pp.237-250, 1993

Abstract

The ray tracing technique is very often used for image synthesis. Many methods for its acceleration have been developed. This paper presents two possible solutions — modification of space subdivision method and application of parallel processing, with obtained experimental results.