

## PORÓWNANIE WYDAJNOŚCI JĘZYKÓW CIENIOWANIA CG I HLSL

Jerzy Dąbkowski

*Instytut Teleinformatyki, Politechnika Krakowska*

*Instytut Inżynierii Rolniczej i Informatyki, Uniwersytet Rolniczy w Krakowie*

Rafał Król

*Instytut Teleinformatyki, Politechnika Krakowska*

**Streszczenie.** Na podstawie zaprojektowanych dziesięciu testów szybkości wykonania identycznych programów cieniujących, jednego napisanego w języku HLSL, drugiego w języku Cg porównano wydajność tych języków w środowisku Windows XP. W zależności od rodzaju testu stwierdzono albo porównywalną wydajność obu języków albo przewagę szybkości działania programów napisanych w HLSL.

**Słowa kluczowe:** języki cieniowania, rendering, potok graficzny

Rozwój technologii programowalnych kart graficznych i powstanie języków cieniowania wysokiego poziomu z początkiem obecnej dekady, doprowadził do zmiany technologii przetwarzania potoku graficznego. Można dziś programować procesor graficzny (GPU) niezależnie od zadań jakie wykonuje główny procesor komputera (CPU). W wyniku zdarzeń na rynku funkcjonują trzy równorzędne (prawie identyczne co do specyfikacji) języki cieniowania wysokiego poziomu – Cg (**C** for **graphics**) [Fernando, Kilgard 2003] wspierany przez firmę NVIDIA i niezależny od platformy i stosowanego interfejsu graficznego (DirectX czy OpenGL), HLSL (**H**igh **L**evel **S**hading **L**anguage) firmy Microsoft – część biblioteki DirectX [Dempski 2003] począwszy od wersji 9.0 oraz – GLSL (**O**pen**G**L **S**hading **L**anguage) będący częścią biblioteki OpenGL 2.0. W pracy niniejszej porównano szybkość działania dwóch najpopularniejszych języków cieniowania: Cg i HLSL. Język Cg może być używany w różnych systemach operacyjnych (Windows, Linuks) i przy wsparciu różnych bibliotek graficznych, zaś HLSL tylko w systemie Windows we współpracy z biblioteką DirectX.

Środowisko programowo – sprzętowe, w którym przeprowadzono testy jest następujące:

- System operacyjny: WindowsXP.
- Biblioteka graficzna: DirectX v. 9.0c, OpenGL v. 1.4.
- Procesor CPU: AMD Athlon 3000+.
- Karta graficzna: ATI RADEON 9600.

Dla możliwie pełnego obrazu działania wymienionych języków w różnych warunkach renderingu, opracowano dziesięć testów wyszczególnionych w poniższej tabeli [NVIDIA 2006]. Osiem pierwszych testów należy do grupy testów statycznych, dwa ostatnie sprawdzają szybkość wykonania animacji i są testami dynamicznymi. Kształty renderowanych brył i rodzaje zastosowanych tekstur pokazano na rysunkach 1 i 2.

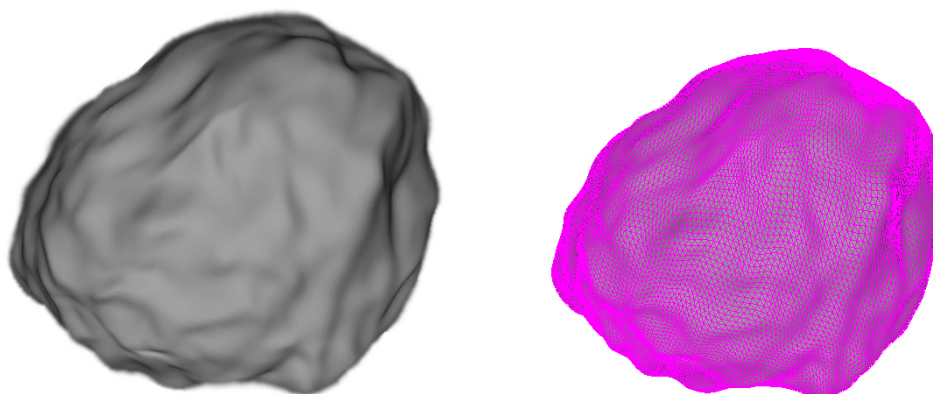
Czasy wykonania testów mierzono w cyklach procesora CPU za pomocą sygnałów przerwań. Technikę mierzenia liczby cykli dopracowywano doświadczalnie przez

wielokrotne dokonywanie pomiarów wstępnych i oczyszczanie przedziału czasowego z artefaktów związanych z czynnościami maszyny innymi niż mierzone efekty. Sposób mierzenia czasów wykonania był identyczny dla obydwu rodzajów programów cieniowania (Cg i HLSL) [Lovesey 2005, Petzold 1999].

Tabela 1. Rodzaje testów

Table 1. Test types

| Lp. | Nazwa   | Opis   |
|-----|---------|--|
| 1   | Test 1  | Proste przekształcenie wierzchołków bez ingerowania w jednostkę fragmentów.  |
| 2   | Test 2  | Nałożenie na siebie dwóch tekstur z przesunięciem.   |
| 3   | Test 3A | Zastosowanie techniki mapowania środowiska (tekstury kubicznej) na obiekt, przy czym główne obliczenia wykonywane są w jednostce wierzchołków.       |
| 4   | Test 3B | Podobnie jak powyżej, z tym że główne obliczenia wykonywane są w jednostce fragmentów.   |
| 5   | Test 4A | Zastosowanie techniki załamania światła (z wykorzystaniem mapowania środowiska), przy czym główne obliczenia wykonywane są w jednostce wierzchołków. |
| 6   | Test 4B | Test jak powyżej, z tym że główne obliczenia wykonywane są w jednostce fragmentów.   |
| 7   | Test 5  | Test efektu Fresnela oraz efektu chromatycznego rozszczepienia światła.  |
| 8   | Test 6  | Mapowanie środowiska wraz z zastosowaniem techniki rzucania cienia na samego siebie (na obiekt).   |
| 9   | Anim. 1 | Animacja położenia wierzchołków dająca w rezultacie efekt płynnego obiektu, bez ingerowania w jednostkę fragmentów.                                  |
| 10  | Anim. 2 | Zastosowanie techniki mapowania środowiska w połączeniu z animacją wierzchołków.   |

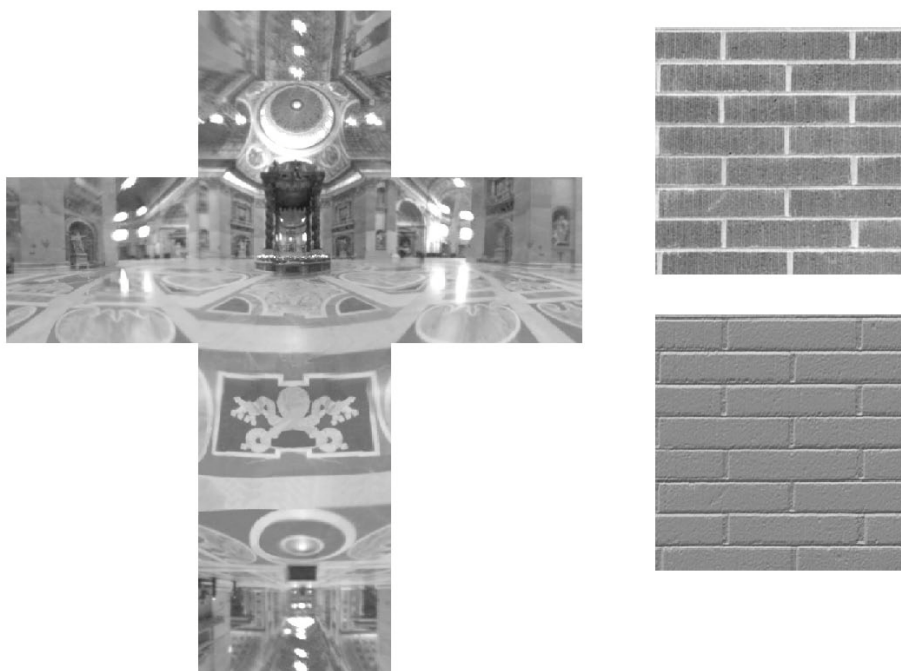


Rys. 1. Obiekt, zniekształcona kula wykorzystana do renderingu (powierzchnia i siatka wielokątów)  
 Fig. 1. Object, deformed sphere used for rendering (surface and lattice of polygons)

## Porównanie wydajności...

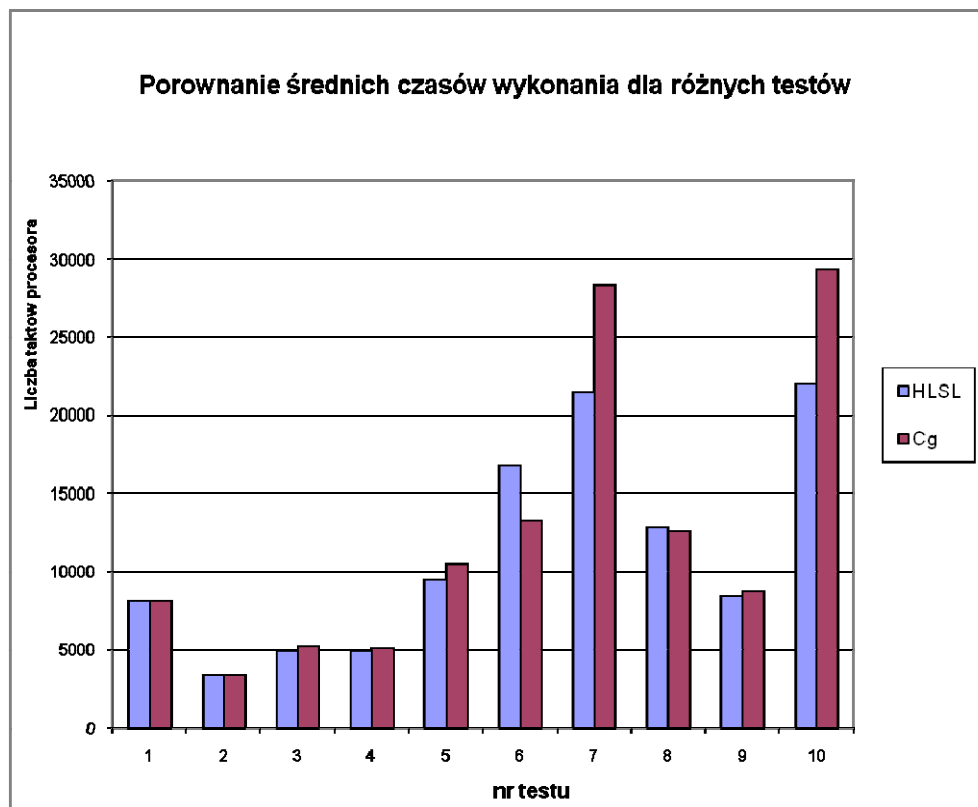
Wyniki w postaci średnich wartości z próby o liczności  $n=200$  przebiegów dla każdego rodzaju testu przedstawione są na rysunku 3. Widać, że dla większości testów wyniki dla Cg i HLSL nie różnią się istotnie. Wyjątek stanowią testy numer 6, 7 i 10. W teście szóstym przewagę wykazuje język Cg, zaś w testach 7 i 10 język HLSL. Ponieważ testy 5 i 6 różnią się jedynie środowiskiem wykonania (5 – jednostka wierzchołków, 6 – jednostka fragmentów) to można wnioskować, że mimo iż język HLSL został przez Microsoft zaprojektowany do pracy w warunkach w jakich były przeprowadzane testy i powinien wykazywać przewagę nad Cg, to gorzej współpracuje z jednostką fragmentów.

Ogólnie można wyniki przeprowadzonych testów podsumować w ten sposób, że oba języki są porównywalnie wydajne w renderowaniu grafiki z przewagą języka HLSL w przetwarzaniu efektów dynamicznych. W środowisku testowym (DirectX v. 9.0) język HLSL wykazuje spowolnienie w przetwarzaniu jednostki fragmentów.



Rys. 2. Rodzaje tekstur użytych w programie. Po lewej tekstura środowiska (kubiczna), wykorzystywana w technice odbicia i załamania światła. Po prawej stronie u góry widoczna jest tekstura ściany, natomiast po prawej stronie u dołu mapa wysokości dla tekstury ściany

Fig. 2. Types of textures used in the program. On the left-hand side, the environment (cubic) texture used in the light reflection and refraction technique. On the upper right-hand side, there is a wall texture visible, whereas on the lower right-hand side there is a height map for the wall texture



Rys 3. Wyniki testów  
 Fig. 3. Test results

## Bibliografia

- Fernando R., Kilgard M.J.** 2003. Programowanie grafiki w czasie rzeczywistym – Język Cg. Helion. ISBN: 83-7361-241-6.
- Dempski K.** 2003. DirectX – Rendering w czasie rzeczywistym, Helion, ISBN: 83-7361-120-7.
- Lovesey A.** 2005. A Comparison of Real Time Shading Languages. CS4983 Senior Technical Report. University of New Brunswick Canada. Materiały informacyjne Firmy.
- Petzold Ch. 1999. Microsoft Press. November 11. ISBN: 1-57231-995-X
- NVIDIA. 2006. Cg Toolkit. A Developer’s Guide to Programmable Graphics. Release 1.4.1, NVIDIA Corporation. Materiały informacyjne Firmy.

## **A COMPARISON OF CG VERSUS HLSL SHADERS EFFICIENCY**

**Abstract.** Based on ten tests especially designed for checking the execution time of two identical shading programs, one written in HLSL and the other one in Cg, there were comparisons made of the efficiency of these languages in Windows XP environment. Depending on the type of test one finds either equivalent efficiency or advantage of HLSL programs.

**Key words:** shading languages, rendering, graphics pipeline

**Adres do korespondencji:**

Jerzy Dąbkowski; e-mail: [rtdabkow@cyf-kr.edu.pl](mailto:rtdabkow@cyf-kr.edu.pl)  
Instytut Teleinformatyki  
Politechnika Krakowska  
ul. Warszawska 24  
31-155 Kraków