

1K-1

3次元グラフィックス・ワークステーションの評価

- PHIGS/PHIGS+ -

吹田和彦 中島美也子 肥後野恵史 諸田勝義 岡明男 吉岡謙
株式会社 東芝 東芝CAEシステムズ株式会社

1. はじめに

当社では、コンピュータ・グラフィックスの技術計算分野において、PHIGS/PHIGS+について、その性能、開発環境、利用動向等を調査中である。今回、この調査の一環として、PHIGS/PHIGS+をサポートしている、いくつかの、3次元グラフィックス・ワークステーションのグラフィックス・パフォーマンスの評価を試みた。

評価には、PHIGS/PHIGS+を用いた同一のアプリケーション・プログラムを用いた。これにより、グラフィックス・データ作成時間、および、表示時間を測定し比較を行った。

本論では移植に際して、生じた問題点に関して、特に、それぞれのPHIGS/PHIGS+における言語バインディングの違い、インプリメンテーションの違いに着目し、考察した結果を述べる。

2. PHIGS/PHIGS+概要

(1) ISO標準ソフトウェア

ISOにおける、グラフィックスのAPI (アプリケーション・プログラム・インタフェース) 標準化作業の一環として、PHIGSおよびPHIGS+の作業がある。API標準パッケージは、アプリケーション開発の開発コストおよび期間を大幅に削減する。それは、次の3点に起因する。API標準パッケージはデバイス独立である。また、ソフトウェア開発におけるグラフィックス記述の手間を大幅に削減する。さらに、開発したソフトウェアは、移植が容易である。

(2) PHIGS/PHIGS+とは

PHIGSの関数は階層構造を持った3次元グラフィックス・データの定義、修正、表示を対話的にとりあつたうことができる。アプリケーションをPHIGSで記述することによって、対話型の3次元グラフィック・システムを構築することが容易になる。

PHIGS+はPHIGSの機能拡張であり、例えばシェーディングをサポートする機能がある。

(3) ISO標準化作業とPHIGS/PHIGS+

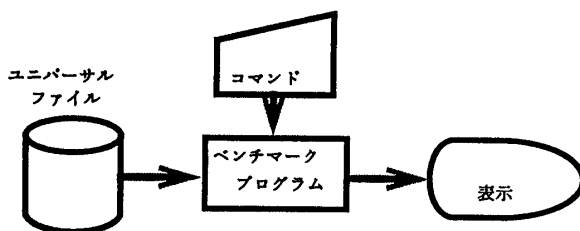
PHIGSは、1989年にはIS 9592の正式文書が公布される予定である。また、PHIGS+は、1988年から、標準案として提案され、標準化作業が進められている。

3. グラフィックス・ワークステーション評価手法

(1) アプリケーション・プログラム概要

ベンチマークテストプログラムはFEMのポスト処理用ソフトウェアベースにして、ベンチマーク用に改造した。

- ・使用ライブラリ・・・PHIGS/PHIGS+またはPHIGS
- ・使用言語・・・・・・C
- ・入力ファイル・・・・ユニバーサルファイル



(2) アプリケーション・プログラムの移植

ソースコードは、各機種で同一のものが使えるようにした。そのためにヘッダ等、システムに依存する部分はプリプロセッサのIFDEF-ENDIFを利用した。

PHIGS/PHIGS+の関数で少しでも用法が異なるものがあつた場合は、それらの関数を集めて一つのファイルにして、変更点を明確にした。

外部仕様の変更はなるべく避けるが、ライブラリがその機能をサポートしない場合は、その機能は実現しなかった。

Evaluating engineering workstations for 3Dgraphics -PHIGS/PHIGS+-

Kazuhiko SUITA¹, Miyako NAKAJIMA¹, Shigefumi HIGONO¹, Katuyoshi MOROTA², Akio OKA², Ken YOSHIOKA²

¹ TOSHIBA Coporation, ² TOSHIBA CAE systems, Inc.

(3) 評価機種概要

- 1 グラフィック・スーパーワークステーション
- 2 エンジニアリング・ワークステーション+高速グラフィックス・プロセッサ
- 3 エンジニアリング・ワークステーション+グラフィックス・プロセッサ
- 4 エンジニアリング・ワークステーション

(4) 測定項目

図形データの作成時間と表示時間をワイヤフレームモデルとサーフェスモデルについて測定し、各機種のPHIGS/PHIGS+のパフォーマンスの目安とした。今回のテストでは、それに加えて以下の項目について、パフォーマンスを測定した。

- 1 異なるベクトル数、ポリゴン数によるパフォーマンス
- 2 ストラクチャ構造の変化によるパフォーマンス
- 3 並列処理を行うものには、プロセッサ数の変化とパフォーマンス
- 4 使用するプリミティブの種類によるパフォーマンス (外部仕様は同じ)

(5) 測定方法

対象機種がワークステーションであったため、UNIXの関数`gettimeofday`を利用してELAPSE TIMEを算出した。

図形データ作成時間は、各モデル(ワイヤフレームモデル、サーフェスモデル)のストラクチャーの作成開始から、終了までの時間を測定する。

表示時間は、X軸回りに回転させながら繰り返し表示させた時の時間を計測し、1回当たりの表示時間を算出する。

シェーディングはフラット・シェーディングを用いた。

4. 移植に際して生じた問題点

(1) 言語バインディングと移植性

C言語バインディングを持たず、FORTRANバインディングを持つPHIGS/PHIGS+をサポートするワークステーションへの移植では、CプログラムからFORTRAN関数を呼ぶように変更した。そのため、使用する全ての関数について関数を置き換えた。少々手間がかかった。

(2) PHIGS+拡張と移植性

PHIGS+拡張に、ライティングとシェーディング機能があるが、PHIGS/PHIGS+ワークステーションからPHIGSワークステーションへの移植の際に問題になり、外部仕様の変更を余儀なくされた。すなわち、出力イメージは、シェーディングしないものとなった。

(3) インプリメンテーションによるちがいと移植性

PHIGS/PHIGS+は、各インプリメンテーションごとに機能拡張が行われている。たとえば、Double Bufferや、それぞれのワークステーションの持つウィンドウシステムのWorkstationとしての利用方法などである。これらについて、アプリケーション・プログラムの移植の際、若干の変更を要した。

また、ワイヤフレームの隠線消去処理の有無により外部仕様の変更が避けられない場合があった。

そのほか、各機種のPHIGS+拡張プリミティブのインプリメンテーションに差があった。

5. 測定結果の概要

(1) メモリとパフォーマンス

データ量が増えると、パフォーマンスの劣化が起きた。グラフィックス・データの作成の限界のデータ量および、グラフィックス・データの表示の限界のデータ量に機種によるばらつきがあった。

(2) ストラクチャ構造とパフォーマンス

ストラクチャ構造を複雑にするとパフォーマンスが劣化した。限界のデータ量も小さくなった。これも使用した機種によりばらつきがあった。

(3) PHIGS+拡張プリミティブとパフォーマンス

PHIGS+拡張プリミティブを利用することによりパフォーマンスが向上するものがあった。

(4) パラレルプロセッシングとパフォーマンス

機種により、効果の違いが見られた。

6. 終わりに

本論ではPHIGS/PHIGS+ベンチマーク・テストにおいて、アプリケーション・プログラムの移植に伴う問題点を明らかにした。以下に、その問題点を列記する。

1. 言語バインディングと移植性
2. インプリメンテーションによるちがいと移植性
3. PHIGS+拡張と移植性

当社ではPHIGS/PHIGS+の調査を今後とも継続する所存である。今後も、調査のうえで明らかにできた問題点等を報告する予定である。