

携帯端末上での 3 次元データ表示方法に関する検討

庄司こずえ 今野晃市
岩手大学工学部

1. はじめに

3 次元データ表示技術は、計算機性能の飛躍的な向上に伴って、さまざまな分野で応用されている。例えば、コンピュータグラフィクス(CG)による風景の描画や、不動産物件の表示はその一例であり、これらは我々の生活と密接にかかわり合っているといえる。最近では、作業マニュアルをインターネット上で 3 次元的に表示し、その解説を対話的におこなうなどの試みも行われており、3 次元画像表示はますます注目を浴びている技術となっている。

このような技術を携帯端末上に搭載することにより、移動中の人に対する情報の伝達・取得が容易になり、その利便性は飛躍的に向上すると考えられる。しかしながら、大規模な 3 次元データを携帯端末で扱うには、処理時間がかかってしまうなどの問題も多い。

以上のような背景で、本研究では、携帯端末上での 3 次元データ表示システムの構築を目的として、まず、PocketPC 上で 3 次元データの表示を行うプログラムを作成し、3 次元データを表示する際のデータ量と描画時間について比較検討をおこなった。本稿では、その詳細について報告する。

2. インターネット上の 3 次元データ

インターネット上で利用される 3 次元データは、3 次元データを表示するという用途だけではなく、目的に応じて、多様な情報との統合により活用範囲を広げている。従来、インターネット上の 3 次元技術として広く知られていたものとして VRML(Virtual Reality Modeling Language)があったが、現在では、さまざまな特徴を持ったインターネット上 3 次元技術が存在する。

その一つとして、XVL(eXtensible Virtual world description Language:ラティス・テクノロジー)がある。通常、3 次元データはそのデータ量が膨大であり、表示のためには多大な計算機パワーを消費する。もし、表示精度を上げればデータ量はさらに増大し、処理時間も余計にかかる

ことになる。したがって、インターネット上で利用効率をあげるには、データの軽量化と高精度を同時に実現しなければならない。XVL は 3 次元形状の表現方法を VRML のようなポリゴンではなく、パラメトリック曲面を使用することで、データの軽量化と高精度を実現している。他にも、Java 言語をベースにした、マルチプラットフォームを実現している Cult3D(Cycore)や、独自のレンダリングモジュールでハイクオリティなレンダリングを実現できる Viewpoint(Viewpoint)など、多種の技術が存在し、それぞれ用途や目的に応じて利用されている。

本研究で対象としている PocketPC プラットフォームでは、VRML ビューアである Pocket Cortona(Parallel Graphics)の開発が進められているが、大規模なデータを表示するには処理時間がかかるといった問題も抱えている。携帯端末上での 3 次元データ表示を実用化するには、大規模データの高速な表示が期待されている。

3. PocketPC 上での 3 次元データ表示

本研究では、データ量と描画時間の関係を把握することを目的としているので、既存のものを流用するのではなく新たに 3 次元データ(ポリゴンデータ)を表示するプログラムを作成して PocketPC に実装した。以下では、プログラムの主な内容と使用した開発ツールについて述べる。

3.1 3 次元データ表示プログラムの作成と実装

3 次元形状のデータとして、ポリゴンデータを使用した。3 次元データ表示プログラムの主な内容を以下にまとめる。

- ・モデリング部
3 次元形状データとして、ポリゴンデータを読み込む
- ・レンダリング部
陰面消去と光源に対する面の陰影付け
- ・インタラクティブ操作部

3次元画像の移動として、回転、平行移動、拡大縮小の機能

このような内容で作成したプログラムをPocketPCへ実装し、3次元データの表示を行った。

PocketPCはCOMPAQのiPAQ H3630を用いた。主な仕様を表1にまとめる。また、実際に実行させている様子を図1に示す。

表1 iPAQ H3630のシステム仕様

機能	仕様
プロセッサ	StrongARM SA-1110 206MHz
メモリ	32MB SDRAM



図1 プログラムのPocketPCへの実装

3.2 WindowsCE ソフト開発ツール

PocketPCでは、OSとしてMicrosoftのWindowsCEが搭載されている。そこで、アプリケーション開発ツールとして、Microsoftから無償で配布されているeMbedded Visual Tools 3.0を使用した。実際の開発は、Microsoft eMbedded Visual Tools 3.0をWindowsマシンにインストールし、エミュレーション上で行い、プログラムが完成したら、WindowsマシンにWindowsCEマシンを接続して実行ファイルのダウンロードを行う。

4. 評価実験と検討

今回作成した3次元データ表示プログラムを用いて、実際にPocketPC上での3次元表示を行い、3次元データ量（ポリゴン数）と一回の描画にかかる時間を測定した。実験に使用した形状は(a)部品、(b)石器、(c)魚、の3種類であり、それぞれの表示結果は図1に示した。また、各々の形状におけるポリゴン数、表示に要した処理時間を表2にまとめた。



(a) 部品 (b) 石器 (c) 魚

図2 PocketPC上での3次元データの表示結果

表2 データ量と描画時間

形状	ポリゴン数	比率	描画時間(秒)	比率
(a)	416	1	0.474	1
(b)	876	2.1	1.516	3.2
(c)	1341	3.2	3.559	7.5

これらの結果から、ポリゴン数が416からおよそ2倍に増加したとき描画時間は約3倍になり、ポリゴン数が3倍に増加すると描画時間は約7.5倍に増えていることが分かる。また、現状のスペックでは、一回の描画に時間がかかるために画像移動の処理がスムーズに行われないことも明らかになった。したがって、このままでは3次元データ表示システムとしての利用は難しく、今後、高速表示に向けプログラムの改善が必要であると考えられる。その一方で、WindowsCE上での3次元グラフィックスAPIの充実に期待したい。

5. まとめと今後の課題

PocketPC上での3次元データ表示プログラムを作成し、PocketPC上で3次元データ量と描画にかかる時間の比較をおこなった結果、ポリゴン数が増えると描画にかかる時間が大幅に増加してしまうことが確認できた。この結果をもとに、今後は、データの軽量化やプログラムの改善をおこない、3次元データの描画速度を短縮していかなくてはならない。

参考文献

- 1) 脇田 玲, Web3Dの現状と動向, 映像情報メディア学会誌, vol.56, No.10, pp26-30, 2002
- 2) 鳥谷 浩志, XVL ネットワーク 3D 規格実践ガイド, CQ 出版社, 2002
- 3) Microsoft eMbedded Visual Tools 3.0, <http://www.microsoft.com/japan/windows/embedded/>