

ゲーム用入力デバイスを用いた 3次元グラフィクスビューワソフトウェアの開発と医用画像表示

大川 貴志, 佐川 盛久, 早川 吉彦

北見工業大学工学部情報システム工学科 〒090-8507 北海道北見市 165 番地
(2011年3月2日受付, 2011年3月25日最終受付)

Development of three-dimensional graphic viewer software using game-device Wii[®] Remote and medical image displaying

Takashi OKAWA, Morihisa SAGAWA and Yoshihiko HAYAKAWA

Department of Computer Science, Faculty of Engineering, Kitami Institute of Technology
165, Koen-cho, Kitami, Hokkaido 090-8507 Japan

(Received on March 2, 2011. In final form on March 25, 2011)

Abstract : Three-dimensional (3D) virtual space is recognized by some ubiquitous devices (controllers) in the world of computer games. Wii[®] Remote (Nintendo, Japan) is the controller of Wii[®], and some sensors and interfaces are installed. 3D acceleration sensor enables to work as a 3D positional sensor. Our trial is to develop a 3D medical graphic viewer using a game-input-device, Wii Remote.

The 3D acceleration sensor can recognize 3D virtual space precisely and permit the intuitive operation. The programming language for a medical graphic viewer development was C#. We used the OpenSource Project, Wiimotelib 1.7, for C++/C#. 3D virtual place was created by XNA GameStudio 3.1 (Microsoft). Our development included the rotation, parallel displacement/shift/translation, scaling (expansion/reduction) of 3D objects, and various light-source settings.

As results, the 3D object was rotated using three axes of the 3D acceleration sensor and enlarged or reduced using Wii Remote's Up and Down buttons. Also the 3D object showed the parallel translation using the cross-joint button while a trigger button B pressed. A medical graphic viewer was developed using a game-input-device, Wii Remote. Various operations are demonstrated. Such intuitive operation shows potentials as a medical graphic viewer.

Key words : medical graphic viewer, 3D acceleration sensor, 3D virtual space

1. はじめに

医療における画像診断の世界では、様々なイメージング・モダリティで作られた画像をもとにした多様な三次元表現が可能となり、診療室のディスプレイには三次元画像が溢れている[1].

一方、コンピュータゲームの世界では、3次元ヴァーチャル空間を認識し、操作するための様々な入力デバイスが開発されている。入力デバイスを活用した直感的な操作が、ゲームにおける迫力の増強・臨場感の高揚に大いに貢献していると考えられる。また、その空間を製作する3次元コンピュータ・グラフィクス表現もハードウェアとソフトウェアの両面で目覚ましい進歩をしている[2-4].

一方で、ゲーム用のインターフェイスだけでなく、iPadのような人間の指そのものを入力インターフェイスとする「タブレットPC」はいち早く医療現場に導入されており、手術、遠隔医療、リハビリ、患者説明、医学教育、在宅医療などに応用が広がっている[5]。しかし、従来の空間認識デバイスやタブレットPCでは、その多機能性を活かす複雑な操作方法や手順があるがゆえに、医療従事者と患者とでは取り掛かりやすさが格段に違うと考えられる。たとえば、タブレットPCでは大きすぎるかもしれない。医療関係者と患者とのあいだで診断・治療(施術)に対する理解の深さが求められる昨今、「患者も医療用画像を扱うことができる」ということが重要であると思える。

以上の理由から、我々はゲーム用入力デバイスを使って医用三次元画像の操作が可能にならないかと考えた。以前には、3次元ヴァーチャル・マウスというような操作方法もあった[6]。本研究では、ゲーム用入力デバイスとコンピュータ・グラフィクスを組み合わせ、現実感のある操作によってヴァーチャル空間内の3次元オブジェクトを観察・認識し、操作するシステムの開発を行うことにした。今回は、ゲーム用入力デバイスとして「Wii[®] Remote (Wiiリモコン)」(任天堂, 以下 WiiRemote と記載)を用い、3次元ヴァーチャル空間の表示には「XNA GameStudio 3.1」(Microsoft, 以下 XNA GameStudio と略記)を使用した[7, 8].

2. 方法

2.1 開発システムの概要とオブジェクト

本開発システムでは、WiiRemoteの加速度センサで得られるデータを解析して、画面上に表示されている3次元グラフィクスをX, Y, Zの各方向に回転させるという操作を実現する。加えて、3次元オブジェクトの拡大と縮小や平行移動という座標変換、光源調整の機能を持つようにした。

今回、3次元ヴァーチャル空間に置いたオブジェクトは、成人男性骨格形状データ(産総研 H20PRO-905)である[9]。データ使用許諾条件に合意して、この全身骨格形状を利用した。

2.2 プログラミング言語とライブラリ

XNAGameStudio は 3次元ヴァーチャル空間を表現するためのフレームワークである[3]。XNAGameStudio については 2.3.1 で述べる。XNAGameStudio はプログラミング言語 C#に対するサービスであるので、この開発には C#を使用した。

WiiRemoteを使用するためのライブラリには「WiimotLib_1.7」を利用した[10]。これによって、プログラム上で WiiRemote を使えるようにした。WiiRemote とパソコン間のデータ通信は Bluetooth インターフェイスによって行われる。Bluetooth の信号受信には、Princeton 社の PTM-UBT3S と東芝製スタック (Stack) を使用した。

2.3 開発システムの構成

2.3.1 XNAGameStudio

3次元ヴァーチャル空間を表現する XNAGameStudio の利点は、3次元グラフィクスを手軽に扱えること、冗長なプログラムコードにならないことが挙げられる。これにより 3次元グラフィクスの差し替えも容易にできる。

XNAGameStudio の座標系は右手座標系であり、+X, +Y, +Z方向は、それぞれ水平方向の右、重力方向の上、および操作者の方向を向いている (Fig.1)。XNAGameStudio におけるオブジェクトの取り扱いは、“code sample details : Custom Model Importer” [11]を参考にした。

2.3.2 加速度センサによる 3次元オブジェクトの回転

WiiRemote の位置認識には赤外線センサを使用する。この位置認識を用いてオブジェクトを操作するのが通常である。しかし、今回のオブジェクトの操作は、この位置センサを使わずに、3軸加速度センサのデータのみを利用した。なぜならば、グラフィクス・ビューワという特性上、位置認識にそれほど重要な要素はない。赤外線 LED とそれを感知するセンサを使う環境に依存する場合、電源を含めた装置を必要とするだけでなく、赤外線 LED とセンサの正常動作が保証される認識範囲等が限定されるなどの問題があると予想されるからである[4][12]。

WiiRemote における加速度の検出軸は、Fig.2 に示すような3軸になっている。この加速度センサの時間分解能 (1秒間に 100 回程度、公式ではなくプログラムで算出した結果なので断定はできない。) と位置分解能 (Wii モーションプラスも装着した) は高く、送受信する時系列データをそのまま使うと 3次元グラフィクスが「ぎこちなく動く」という挙動を見せる。そこで、受信した加速度のデータを

Right-Handed System

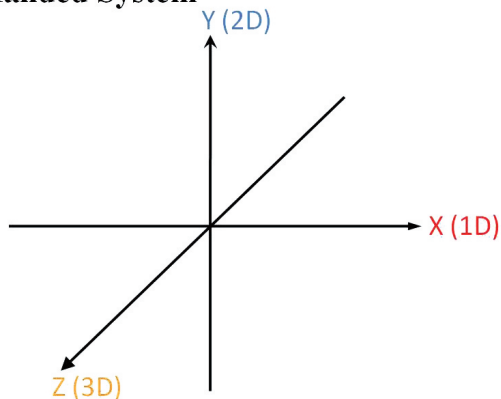


Fig.1 Right-Handed System. Keys : 1D is thumb (first digit). 2D is index finger (second digit). 3D is middle finger (third digit).

50 回ごと (時間に換算して 0.5 秒ごと) に平均値を計算し、動作の決定にその値を用いることで滑らかな 3次元グラフィクスの操作ができるようにした。

ここで使った変換とその説明は以下のとおりである。

```
Matrix transform = Matrix.Identity ;
基となるマトリクスを作成する。
transform *= Matrix.CreateRotationX (radz) ;
Z 軸方向の加速度の平均値を用いて X 軸を中心軸とし、
オブジェクトを回転させる。
transform *= Matrix.CreateRotationZ (rady) ;
Y 軸方向の加速度の平均値を用いて Z 軸を中心軸とし、
オブジェクトを回転させる。
transform *= Matrix.CreateRotationY (-radx) ;
X 軸方向の加速度の平均値を用いて Y 軸を中心軸とし、
オブジェクトを回転させる。
```

2.3.3 3次元オブジェクトの座標変換

WiiRemote 上で操作できる十字 (cross joint) キー、A ボタンおよび B ボタンなどを用いて、3次元オブジェクトの拡大縮小や平行移動といったことができるようにした。

WiiRemote のキーやボタンの操作によって出力されたデータをもとに三次マトリクスを作成し、それらのマトリクスを掛け合わせることで座標変換を行う。ここで使った変換とその説明は以下のとおりである。

```
transform *= Matrix.CreateScale (this.ca) ;
WiiRemote の十字キーの上下ボタンを押した回数に対応してスケールを作成する。
transform *= Matrix.CreateTranslation (this.position) ;
WiiRemote の B ボタン (トリガーボタン) が押されているときの各十字キーを押した状態に応じて、オブジェクトを平行移動させる。
最終的に、掛け合わされたマトリクスの各要素を用いて、3次元オブジェクトを描画する。
```

2.3.4 光源の調整

3次元グラフィクスを表現する上で欠かせないのが光源である。光源の設定条件がなければ、3次元のオブジェクトには見えない。光源の性質を記述するパラメータとして、以下のような項目を設定できるようにした。これらの項目はキーボードから操作できるようにした。

- LightEnable : ライトの有効フラグ
- DiffuseColor (Color) :

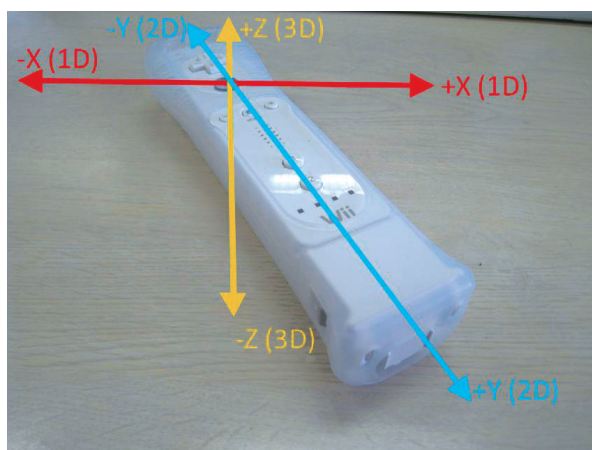


Fig.2 X-, Y-, Z-axes of WiiRemote acceleration sensor. Keys : 1D is thumb (first digit). 2D is index finger (second digit). 3D is middle finger (third digit).

拡散反射（オブジェクト表面から様々な方向への光の反射）

- SpecularColor (Color) :
鏡面反射（オブジェクト表面から一方向への光の反射）
- DirectionX, Y, Z : カメラの向き
- AmbientLightColor : 環境光（空間上の光）
- PreferPerPixelLighting :
ピクセル単位での描写をするか否か

3. 結果

3次元ヴァーチャル空間とオブジェクトからなるグラフィックスの初期状態はFig.3のとおりである。今回は赤外線センサによる位置の認識を行わないので、あとはこのデフォルトであるホームポジションからの変化・変位ということになる。このオブジェクトは、「独立行政法人産業技術総合研究所著作物 管理責任者：河内まき子・持丸正明「成人男性骨格形状データ」管理番号：H20PRO-905」である。なお、画面上には、操作キー・ボタンの説明、オブジェクトの位置と光源に関するデータが表示される。

WiiRemoteとディスプレイあるいはグラフィックス・ビューワとの関係はFig.4に示すような初期状態とした。

3.1 加速度センサによる3次元オブジェクトの回転

WiiRemote から送信された加速度データを格納し、50回ごとに平均した値を用いて3次元オブジェクトをX方向、Y方向、またはZ方向に回転させた。Fig.5とFig.6に

回転させた状態の1例を示す。

3.2 3次元オブジェクトの座標変換（拡大・縮小と平行移動）

WiiRemoteの十字キーの上・下ボタンで拡大・縮小を実行できるようにした。オブジェクトのスケールを拡大または縮小する方法と、カメラ（視点）の位置を変える方法があったが、今回は前者を採用した。Fig.7に拡大した例を、Fig.8に縮小した例を示す。

WiiRemoteのBボタン（トリガーボタン）を押している

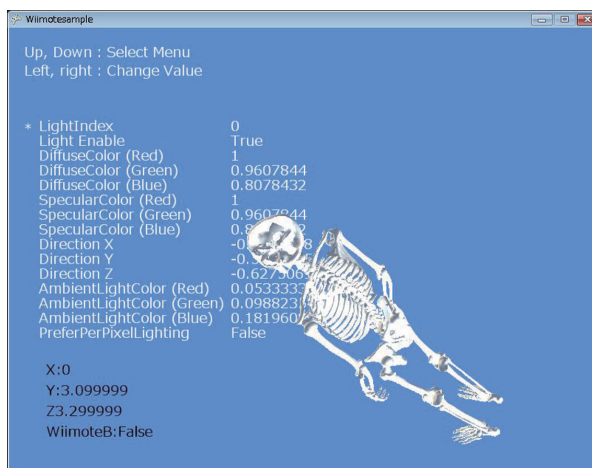


Fig.5 3D virtual space and an object : Example of the rotation by acceleration sensor

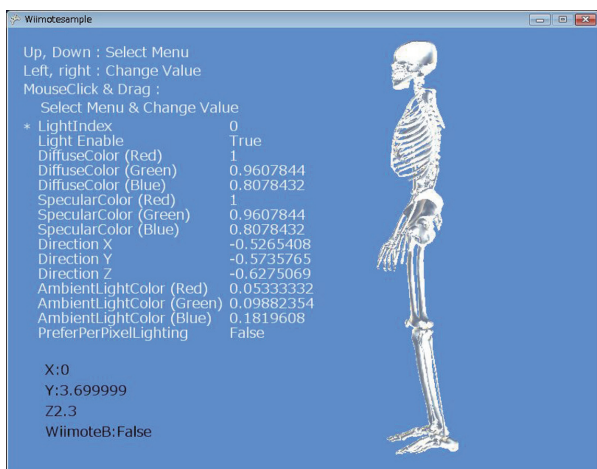


Fig.3 3D virtual space and an object : Default status



Fig.6 WiiRemote and 3D graphics viewer display : The relationship as stated in Fig.5



Fig.4 WiiRemote and 3D graphics viewer display : Default positional relationship

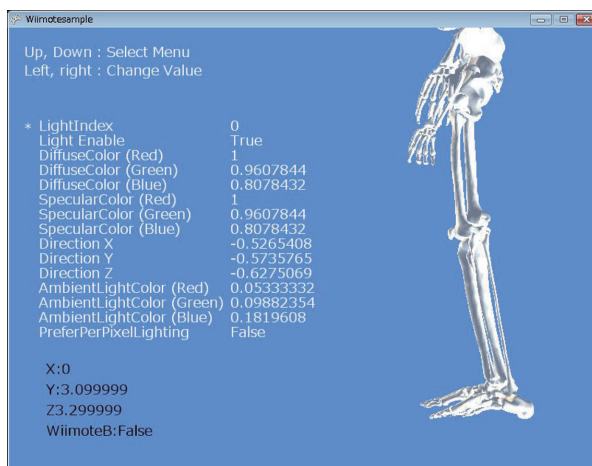


Fig.7 3D virtual space and an object : The object is expanded.

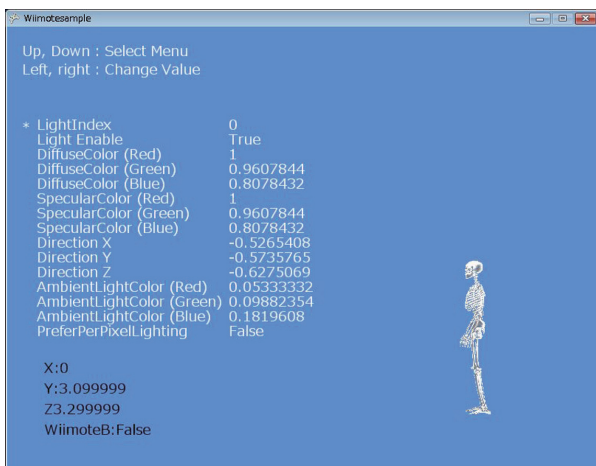


Fig.8 3D virtual space and an object: The object is reduced.

ときに十字キーを操作することで平行移動ができるようにした。Fig.9 にその例を示す。

3.3 光源について

光源の設定がない場合のオブジェクトをFig.10 に示す。真っ白なオブジェクトとして表示されてしまう。Fig.3 から Fig.9 に表示されているオブジェクトは、画面中に示されたパラメータの設定が行われている。

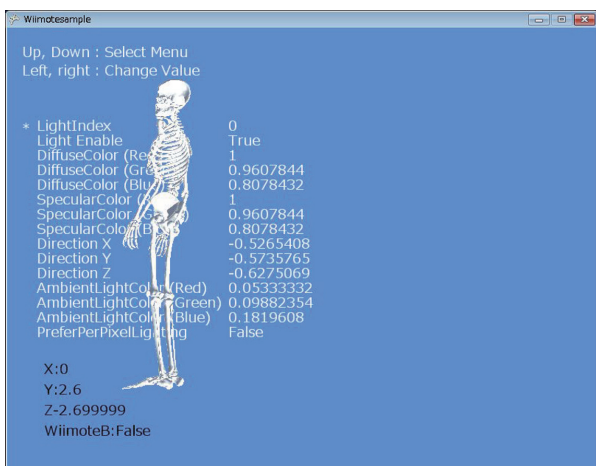


Fig.9 3D virtual space and an object: The object shows the parallel translation.

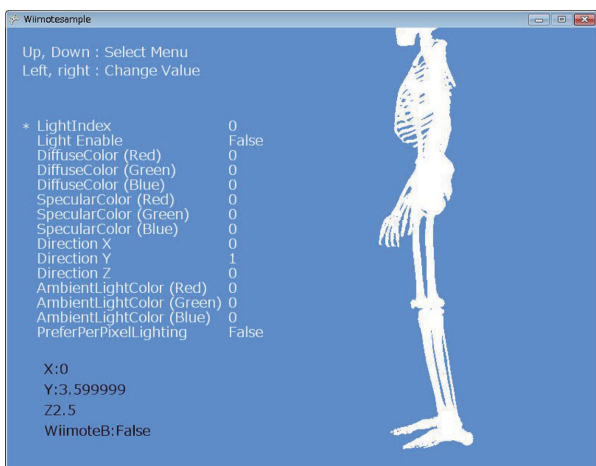


Fig.10 3D virtual space and an object: No light sources are set.

4. 考察

3次元ヴァーチャル空間を表現し、そこに置かれたオブジェクトを WiiRemote で操作するシステムを製作した。オブジェクトに光をあてて3次元的な表現を可能にした。そして、加速度センサによってオブジェクトを回転させ、またキーやボタンの操作によって拡大・縮小、平行移動ができるようにした。

今回は、3次元ヴァーチャル空間を表現するために XNAGameStudio を利用した。この欠点を挙げるならば、表示可能な3次元グラフィクスは基となるファイル形式に依存するので、プログラム上でファイル変換などの加工をする場合は手間がかかること、また対応するグラフィクスのファイル形式（フォーマット）の種類が少ないことである。現在、“.x (DirectX)”と“.fbx”が公式に使えるとされる3次元グラフィクス・ファイルである。なお、非公式ながら“.obj形式”のファイルも表示することができるようになっている[11]。

XNAGameStudio はプログラミング言語 C# に対するサービスであるので、本システム開発には C# を使用した。なお、プログラミング言語 Java においても、同様の WiiRemote の制御に関するライブラリ (WiiremoteJ) が存在する。ただし、そこには Bluetooth 用通信ライブラリ (BlueCove) が必要であるなどの制限があるうえ、Windows PC と WiiRemote 間の Bluetooth 通信において Java 言語は相性が悪いとされる。現時点では BlueCove の不具合であるとされているが、それも今回 C# を使用した理由である。

本システムでは、WiiRemote を使用するためのライブラリで加速度センサのデータを受け取って3次元オブジェクトが回転するようにした。加速度を送られるごと (1秒間に100回程度) に回転させるとオブジェクトがぎこちなく動いたので、WiiRemote から送信された加速度データを格納し、50回ごとに平均した値を用いて3次元オブジェクトを3軸方向に回転させた。これによってスムーズなオブジェクトの動作が実現した。

グラフィクス・ビューワとしての機能は限定されているものの、より直感的な動きで3次元グラフィクスを見ることができるようになった。マウスなどとは違い、ゲーム用デバイスならではのボタンの多さ、ひとつのパソコンに WiiRemote を複数台接続して認識させることができることなどから、その応用範囲は広いと予想される。今回使った XNAGameStudio の3次元グラフィクスの差し替えの容易さからも、WiiRemote、XNAGameStudio および WiimotLib ライブラリの組み合わせは汎用的である。今回は、成人男性骨格形状の表示を行った。医学・医療応用では、X線CT画像などによる3次元グラフィクスをこのヴァーチャル空間に表示して、より直感的な病巣の把握や手術のためのシミュレーションやナビゲーションが考えられる。

課題点を挙げるならば、赤外線 LED とセンサに依存している WiiRemote における位置認識の改善、そしてこのシステムへの取り込みが挙げられる。XNAGameStudio については、対応する3次元グラフィクス・ファイルが多様であることなどが求められる。

参考文献

- [1] 高原太郎, 今井裕:なるほど!! 医用3次元画像—考え方と処理法の虎の巻, 学研メディカル・秀潤社, 2003.
- [2] 白井暁彦, 小坂崇之, くるくる研究室, 木村秀敬:

- WiiRemote プログラミング, オーム社, 2009.
- [3] Kosaka laboratory Tips のホームページ, <http://www.kosaka-lab.com/tips/>
- [4] 白鳥貴亮, Hodgins JK: キャラクタ操作のためのインターフェイス: 加速度センサーを用いたインターフェイスの設計とその考察, 画像ラボ, 20(1), 33-39, 2009.
- [5] iPad で変わる医療, 日経メディカル, 2010年10月号, 日経BP社, 東京, 2010.
http://medical.nikkeibp.co.jp/inc/all/info/mag/nm/index_bn1010.html
- [6] Goodsitt MM, Chan HP, Hadjiiski L.: Stereomammography: evaluation of depth perception using a virtual 3D cursor. Medical Physics 27, 1305-1310, 2000.
- [7] Wii[®]のホームページ, <http://www.nintendo.co.jp/wii/>
- [8] XNA デベロッパーセンターのホームページ, <http://msdn.microsoft.com/ja-jp/xna/default>
- [9] 中村仁彦, 山根克, 諏訪元, 近藤修, 河内まき子, 川地克明, 持丸正明: 2008: 成人男性骨格形状データ (産総研 H20PRO-905)
<http://riodb.ibase.aist.go.jp/dhbodydb/bone/index.html>
- [10] WiimotLib_1.7 のホームページ (Managed Library for Nintendo's Wiimote), <http://wiimotelib.codeplex.com/>, <http://www.codeplex.com/WiimoteLib>
- [11] App Hub (Model Importer Sample)
http://create.msdn.com/en-US/education/catalog/sample/custom_model_importer
- [12] 佐川盛久, 妙瀬田泰隆, 早川吉彦: ゲーム用デバイスを用いた3次元空間認識システムの開発, 情報処理北海道シンポジウム 2009 講演論文集, 85-86, 2009.