

## 教育利用を目指した Mixed Reality システム

井上 祥史\*・伊藤 敏\*\*

(2007年12月7日受理)

### 1. はじめに

Mixed Reality は、視野の一部にシミュレーションによるテキストやCGによる仮想物体を表示して、人間の作業効率や実在感をサポートするシステムで、全て仮想オブジェクトを扱う Virtual Reality と Reality との中間に位置する。実像または実映像の中にシミュレーションによる情報を付加して、現実感をさらに増強させる意味で Mixed Reality は Augmented Reality とも言われる。Mixed Reality は手術や hands free が必要とされる作業現場での情報提供などに用いられており、教育分野でも構造体の内部構造の把握や古代遺跡の仮想体験などの理解支援として有力なツールとなることが期待されている。

Mixed Reality システムを利用する場合、利用者の必要に応じて機能を付加・変更できる柔軟なシステムであることが望ましい。しかしながら現有の Mixed Reality あるいは Augmented Reality システムはソフト及びハード的に複雑で、簡単にカスタマイズできる状況には無い。一方、Mixed Reality システムはいくつかの画像処理および視点決定を中心とする明確な機能を持つソフトウェア部品から構成されている。そのため個々のソフトウェア部品を汎用的な機能部品で代替して、それらを組み合わせることで Mixed Reality システムを構成できる可能性がある。そこで本研究では Mixed Reality の教育展開を図るために、汎用的なツールによるソフトウェア部品からシステムを構成し、その機能を視覚的に確認できる Mixed Reality システムを設計・開発した。

### 2. Mixed Reality システムの設計・開発

#### 2.1 Mixed Reality システムの概要

一般に Mixed Reality システム (以下 MR) の開発には次の3つの要件を決める必要がある：

1. 表示コンテンツ, 2. MR 画像の提示方法, 3. MR で用いる視点決定法である。

要件1. は視野の中に提示するコンテンツの種類で、テキスト提示と仮想物体提示方式とに分けられる。さらにテキスト提示では、単にマニュアルなどその場面に必要な情報を表示するもの、外部センサから取り込んだ情報を基に対象物の位置や温度、圧力などの値と共にシミュレーションで得た計算値などを表示するものがある。仮想物体提示では、観察者の位置をリア

---

\* 岩手大学教育学部 技術教育科

\*\* 岐阜聖徳学園大学経済情報学部

ルタイムで把握し、視点位置から見た仮想物体を提示する。

要件 2. の MR システムの表示方法には、実像の視野の中にテキストや仮想物体を配置する方式と、カメラを通した映像の中にそれらを配置する方式がある。実像の中に仮想オブジェクトを配置するためには、ハーフミラーを用いてディスプレイからの仮想オブジェクトと see-through の実像とを重ねて観察することになる。この方式では、実像の焦点距離と表示器の焦点距離とが一般に異なるために観察者は眼精疲労を起ししやすい。これに対して、カメラ映像の中に仮想物体を重ねる方式では、上記の疲労は起こらない代わりに間接的な映像であるため臨場感に欠ける。また、後者では一体画像であるためネットワーク転送に適するなどの特徴を持つが、一方で背景画像処理を必要とする。

要件 3. の MR における視点決定には、センサ（無線、GPS または超音波など）を用いて位置決定を行う方法と、カメラ画像の解析から観察者の位置決定を行う方法とがある。センサを用いる方法では多くのハードウェア資源が必要で、システム構成および設定が複雑になる傾向が強い。一方、画像解析から位置決定を行う方法として、マーカーを含むカメラ画像の解析から観察者と物体の相対的な 3 次元位置の決定を行う方式があり、代表的なソフトに ARToolKit<sup>1)</sup> がある。ARToolKit は汎用パソコンで手軽に運用できるが、定型処理以外の開発環境は複雑である。最近ではマーカーを用いないで映像の中の特定の対象を認識して 3 次元位置を決定する、より自然な MR システムが開発されているが、画像認識のために別サーバを必要とするなど課題も多い<sup>2)</sup>。

## 2.2 システムの設計及び開発

教育利用を意図した MR システムを設計するに当たって 1. 簡易に扱えること、2. 特別なハードウェアを必要としないこと、3. システム構築の各ステップで処理過程が視覚化できること、4. ソフトウェア部品が交換可能なこと、の 4 点に留意した。

簡易に扱うためにはハードウェアに依存しないようにソフト的な処理を行い、操作が煩雑にならないように基本的なデータ処理機能が揃っている環境が必要である。これらの条件を満たす環境を検討した結果、MR システムの基本構成を 1) コンテンツ提示は仮想物体方式、2) MR 画像表示は実像と重ねる表示方式、3) 視点決定法はカメラ画像のマーカーを用いる方法、とした。

図 1 にマーカーを用いる MR システムの概念図を示す。実装するにあたって 1) の仮想物体

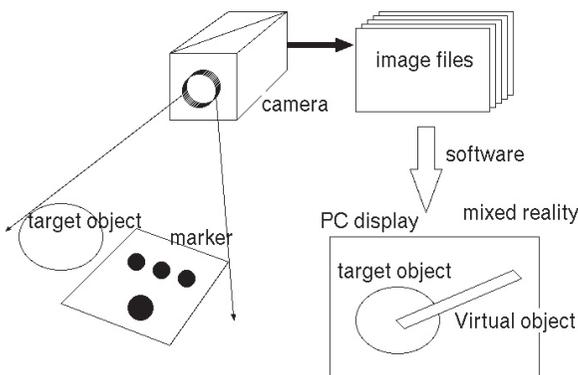


図 1. マーカーを用いる MR システム

の表示には 3D 表示が可能な Java3D を用いてシステムを構成する、2) の表示方法は仮想物体を表示する方式をとり、実映像と重ねる表示方式の場合には、必要な機能部品に交換することとした、3) のマーカーを用いる視点決定法では、静止画像からマーカーの位置関係を読み取る画像処理が必要である。このためにマーカーとターゲットの対象物をカメラで撮影し、一定時間ごとに静止画像をキャプチャーする。そして画像解析から得た位置情報に基づいて、あらかじめ読み込んだ仮想物体の視点を決定して CG 画像を表示する。この一連の操作を Java3D の割り込み処理で行い、MR のリアルタイム表示を実現することにした。図 2 に MR 管理システムに Java3D を用いたデータ処理の流れを示す。以下このシステムの視点決定のアルゴリズム、取得した画像の処理、PC に表示する 3D 画像の生成について述べる。

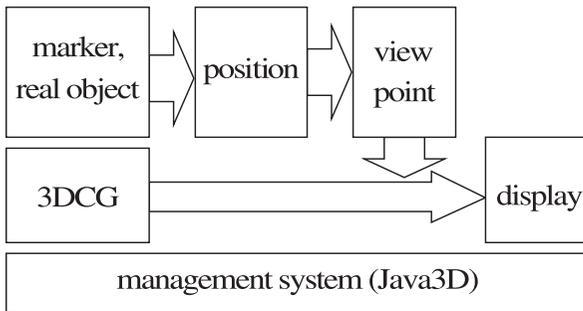


図 2. MR システムのデータ処理

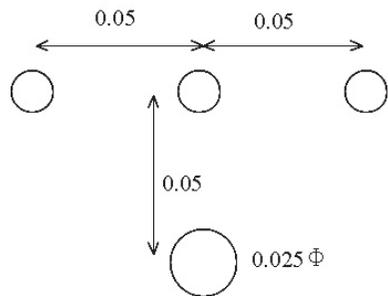


図 3. マーカーの構成. 単位は m

(1) 視点決定のアルゴリズム

視点決定には取得した画像とカメラの位置関係を知る必要がある。そこで、マーカー、用いる座標系、視点決定の順で処理概要についてまとめる。

**マーカー：**マーカーとカメラとの 3 次元の相対位置を決定するために、図 3 に示すように平面上に大小 4 つの赤い円から成るマーカーを作成した。このうち 1 つの円は大きく、他の 3 つの小円は同じ大きさにした。円とカメラとの距離が一定ならば、視点の角度が変化しても画像の楕円の長半径は一定であることを利用して、画像解析で得られた大円の長半径を用いて距離決定を行った。一方、3 つの小円を等距離に配置することによって、画像解析による中央の円の判定を容易にさせ、円の指標付けに利用した。また、画像解析によるマーカーの抽出を容易にするため、マーカーの色は原色に近い赤とした。

**座標系：**カメラ映像からキャプチャーした静止画像内の pixel 単位の 2 次元座標系を画像座標系、観察者の視点位置を原点とする m 単位の座標系をカメラ座標系  $(x, y, z)$  と表すことにする。画像座標系の原点は画像の中央にとる。いま、マーカーの円  $i$  の静止画像における重心位置を  $(u_i, v_i)$ 、カメラ座標系における円の中心位置を  $(x_i, y_i, z_i)$  とすると図 4 から次の関係が得られる。

$$u_i = f \frac{x_i}{z_i}, v_i = f \frac{y_i}{z_i} \tag{1}$$

ただし  $f$  はカメラの焦点距離に比例した光学定数で、あらかじめスケールで校正して決定しておく。固定焦点の USB カメラで  $350 \times 240$  pixel の画像の場合、 $f = 530$  程度の値となった。

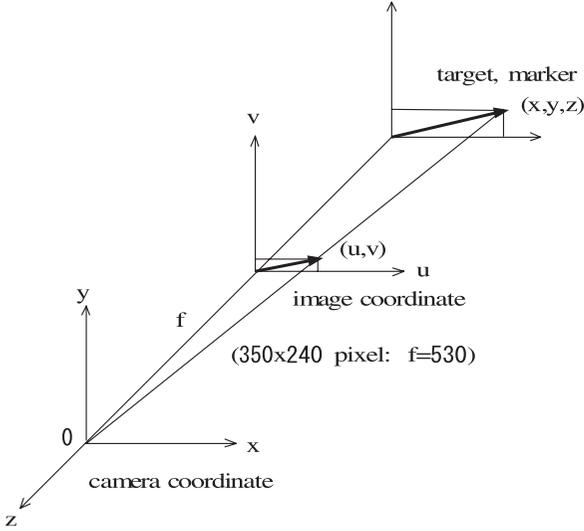


図4. 画像座標系とカメラ座標系

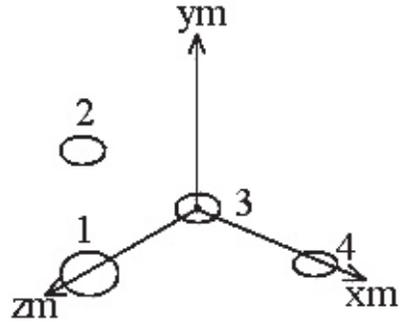


図5. マーカー上の local 座標

視点決定：視点決定には図5に示したマーカー上の local 座標軸

$$(x_m, y_m, z_m) = (r_4 - r_3, z_m \times x_m, r_1 - r_3) \quad (2)$$

を求める必要がある。ここで  $r_i = (x_i, y_i, z_i)$  である。画像位置  $(u_i, v_i)$  は観測値であるため (1) 式より円1, 3, 4の座標の未知数は  $z_1, z_3, z_4$  で表される。

一方、マーカーの円  $i$  と  $j$  の距離を  $l_{ij}$  とするとき

$$l_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \quad (3)$$

の関係が成り立つ。線形化するため

$$\frac{\partial l_{ij}^2}{\partial z_k} = 0, \quad i, j, k = 1, 3, 4 \quad (4)$$

の条件より初期値の次の連立方程式を得る<sup>3)</sup>。

$$\begin{pmatrix} 2a_1 z_{10} + b_{13} z_{30} & 2a_3 z_{30} + b_{13} z_{10} & 0 \\ 0 & 2a_3 z_{30} + b_{34} z_{40} & 2a_4 z_{40} + b_{34} z_{30} \\ 2a_1 z_{10} + b_{14} z_{40} & 0 & 2a_4 z_{40} + b_{14} z_{10} \end{pmatrix} \begin{pmatrix} \delta z_1 \\ \delta z_3 \\ \delta z_4 \end{pmatrix} = \begin{pmatrix} 0.05^2 - l_{13}^2 \\ 0.05^2 - l_{34}^2 \\ 0.05^2 \cdot 2 - l_{14}^2 \end{pmatrix} \quad (5)$$

ここで

$$a_i = 1 + \frac{u_i^2 + v_i^2}{f^2}, \quad b_{ij} = -2 \left( 1 + \frac{u_i u_j + v_i v_j}{f^2} \right) \quad (6)$$

である。これより

$$z_i = z_{i0} + \delta z_1 \quad (7)$$

から収束値を求めた。

一方、仮想物体は CG 作成ソフトの Shade で作成し、その座標系はカメラ座標系と同一である。従ってシステムに読み込まれた仮想物体の初期位置は、カメラ座標系の原点に座標軸に沿って配置される。local 座標系とカメラ座標系との間の回転行列を

$$M = \begin{pmatrix} xm_x & xm_y & xm_z \\ ym_x & ym_y & ym_z \\ zm_x & zm_y & zm_z \end{pmatrix} \quad (8)$$

とすると、仮想物体を  $M^T = M^{-1}$  だけ回転した後、 $(x_3 - x_0, y_3 - y_0, z_3 - z_0)$  だけ平行移動させて仮想物体の視点表示を行った。ただし  $(x_0, y_0, z_0)$  はマーカーと表示位置との差を表すオフセット値である。

## (2) 画像処理

マーカーを含む対象を USB カメラまたは DV カメラで撮影し、キャプチャーソフト LiveCapture2<sup>4)</sup> で 1 秒ごとに静止画像を得た。得られた静止画像を ImageJ<sup>5,6)</sup> を用いてマーカーの赤円を抽出して、それぞれの面積、重心位置、長半径を求めた。画像解析は 1. RGB 分解, 2. R-G, 3. 2 値化, 4. 面積・重心位置・長半径, 5. テキスト書き出し, の一連の操作から成り、これをマクロとして登録して Java3D から呼び出した。マクロ動作中は画像解析のステップごと処理画像が表示される。表 1 にマクロコマンドをリストした。また図 6 に画像解析の処理過程の画像を示す。赤円だけを 2 値画像として抽出するために Red 画像から補色に近い Green 画像の差を取り解析を行った。表 2 に解析の結果得られたテキストデータの例を示す。

表 1. ImageJ のマクロコマンド

```
selectWindow("Live (press space bar to stop)");
run("Copy to System");
saveAs("BMP", "C:\\My Documents\\live.bmp");
open("C:\\My Documents\\live.bmp");
//run("Size...", "width=640 height=480 constrain interpolate");
run("RGB Split");
//selectWindow("Live (press space bar to stop)");
imageCalculator("Subtract create", "live.bmp (red)", "live.bmp (green)");
//run("Image Calculator...", "image1=[live.bmp (red)] operation=Subtract image2=[live.bmp (green)]
create");
setAutoThreshold();
//run("Threshold...");
setThreshold(60, 230);
run("Convert to Mask");
run("Set Measurements...", "area center fit limit redirect=None decimal=3");
run("Analyze Particles...", "size=15-Infinity circularity=0.10-1.00 show=Nothing display clear");
saveAs("Measurements", "C:\\My Documents\\data.txt");
close();
close();
close();
close();
```

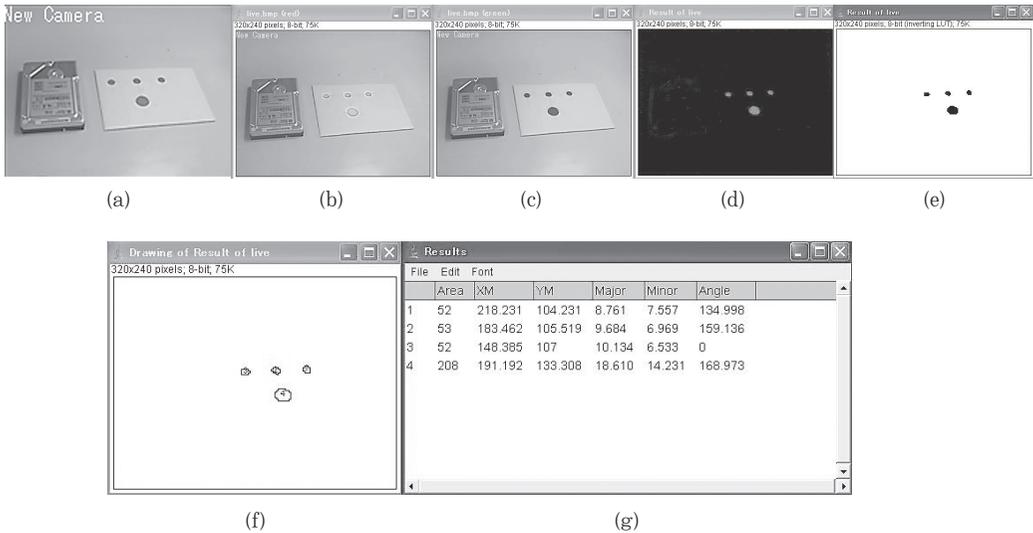


図6. 画像解析の過程. (a) キャプチャ画像, (b) R成分, (c) G成分, (d) R-G画像, (e) 2値化画像, (f) 粒子解析モード, (g) 解析結果のテキストデータ

表2. 画像解析データ (単位は dot, Angle の単位は degree, 左欄の数字は解析順序). (XM, YM) は重心位置, (Major, Minor, Angle) は楕円の (長半径, 短半径, 傾き角).

	Area	XM	YM	Major	Minor	Angle
1	29	138.052	127.81	8.269	4.465	6.314
2	29	166.879	130.052	8.472	4.358	2.963
3	27	196.87	132.278	7.359	4.671	0.773
4	119	160.088	149.416	15.845	9.562	4.942

### (3) 3D表示

MRシステムの記述にはJava3Dおよびクラスライブラリ saiga3D<sup>7)</sup>を用いた. Shadeで描いたCGはwrl形式で例えばVrmlLoader(disc.wrl)として読み込んだ. また, 連続的に画像解析を行うためにJava3DのBehaviorクラスのprocessStimulusメソッドの中で

```
Process process=Runtime.getRuntime().exec("C:¥¥ImageJ -port1 -macro c:¥¥macro.txt");
```

としてImageJのマクロmacro.txtを実行した. マクロでは表1のように, LiveCapture2から書き出した画像ファイルlive.bmpを読み込み, 結果をdata.txtとして書き出している. 図7に試作したMRシステムのブロック図を示す. なお, 図7における背景処理(BG)は未実装である.

## 3. 試作システムの特徴

図8にキャプチャー画像と, 仮想オブジェクトおよび比較のためにそれらを重ね合わせた画像を示す. ハードディスクとマーカーの映像に対して, 回転磁気円盤およびアームの仮想オブジェクトが適切に表示されている. 仮想オブジェクトにはマーカーとハードディスクとの移動量をオフセットとして与えている. カメラ位置を動かすと, 1秒程度の遅延はあるもののほぼリアルタイムで仮想オブジェクトも変化し, カメラアングルからマーカーが消えると仮想オブ

ジェクトは表示されない。このように視点変化に応じて仮想オブジェクトを表示する基本機能を満たしていることが確認できた。

また、第5の円をマーカー平面内で移動させると、円の動きに応じて別の仮想物体に平行移動の動きを与えることも確認できた。

システムを実用に供するためには、小型ディスプレイをハーフミラーから観察できる、あるいは仮想オブジェクトの背景に実映像を表示できるようにしなければならない。また、動作速度は画像解析画面のサイズに依存するため、処理速度を上げるためにはキャプチャサイズを小さくする必要があるが、小さくすると解析精度が劣ることになる。操作性を高めるために、マーカーと仮想オブジェクトとの移動量、画像解析における閾値の調整など複数のパラメータをマニュアル操作できるようにする、機能部品をJava3Dシステムにまとめるなど、いくつかの課題も残っている。

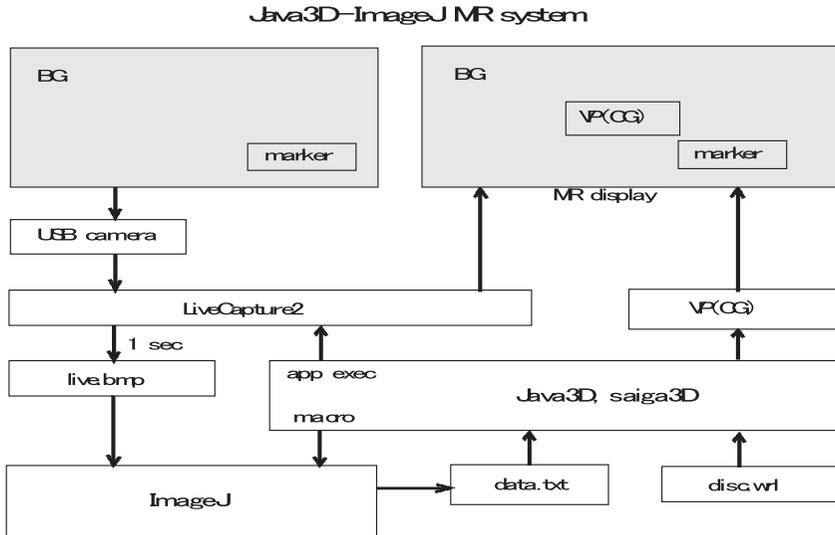


図7. MRシステムの全体構成. BGは背景, VPは視点操作を示す.

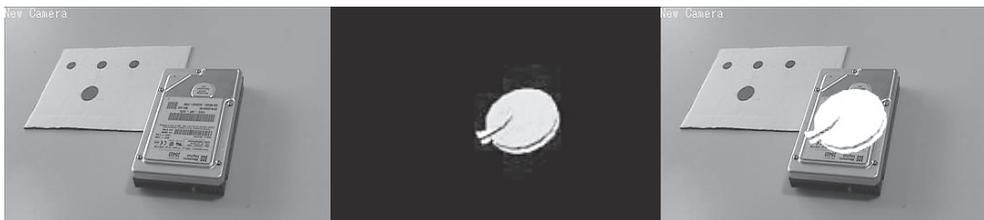


図8. ハードディスクの内部構造を表示するMR. 左からキャプチャ画像, 仮想オブジェクトおよび比較のための合成画像.

#### 4. MRの教育利用

試作したMRシステムは次のような教育利用に適した特徴を持つことを確認した。

1) ImageJ は動作中に解析画面のウィンドウを連続的に表示するため、画像処理過程を段階的に視覚的に確認することができる。バックグラウンドでの実行は単にウィンドウを隠すだけである。2) マーカーを追加して複数の仮想オブジェクトを独立に表示可能であるため、それらの相互作用を扱うことができる。3) 明確な設計目標に対して、機能部品からシステムを組み上げる適切な例題になっている。4) 機能部品のカスタマイズが容易な柔軟なシステムである、などの諸点を挙げる事ができる。

さらに、機能を拡張することによって以下のような展望も持つことができた。Java3D を使って、1) picking や衝突判定、回転や移動のアニメーションなどを表示できる。したがって構造物内部で回転する機械部品を動かしたりできる。2) 両眼視差に応じた 2 画面の表示で立体表示も可能、3) MR の映像表示の特徴を活かしてネットワークを利用した遠隔実験や観察などが可能、など多くの可能性を見出すことができた。

一方、仮想物体を扱う Virtual Reality は精細な映像や力覚の提示などによる没入感を安易に得られるため、扱いによっては弊害をもたらす。これらに配慮しながら、MR システムを用いて内部構造や物理パラメータなど不可視なものを可能な限り表現して理解支援を行うことは、教育活動にとって効果的でもあり必要でもある。実物を表示しながらさまざまな付加情報を表示する MR はこのような教育利用に適した特徴を持ち、今後の発展が期待される。

謝辞：本研究は、文部科学省科学研究費、基盤研究 C 「仮想実験室システムの開発」課題番号 18500696 より補助を受けた。

## 参考文献

- 1) <http://www.hitl.washington.edu/artoolkit/>
- 2) <http://www.altechads.jp/web/products/dfusion/markerless.html>
- 3) 坂井丈泰『GPS のための実用プログラミング』東京電気大学出版局, 2007.
- 4) <http://www.wisnet.ne.jp/~daddy/>
- 5) <http://www.bioarts.co.jp/~ijjp/ij/>
- 6) 小島清嗣・岡本洋一『画像解析テキスト改訂第 3 版』羊土社, 2006.
- 7) 広内哲夫『Java3D グラフィックス』セレンディップ, 2004.  
Tetsuo Hirouchi, <http://www.bunkyo.ac.jp/~hiro/saiga3d/index.html>