

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2010
2010 JSME Conference on Robotics and Mechatronics

ROBOMECH2010 in ASAHIKAWA

ロボティクス・メカトロニクス・フロンティア・ビッグバン
Robotics・Mechatronics・Frontier・Big-Bang

June 13 Sun. - 16 Wed., 2010

Asahikawa TAISETSU Arena

主催 社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



ARToolKit を用いた拡張現実システムのための RT コンポーネント

RT-Components for Augmented Reality Systems using the ARToolKit

○張 敏浩 (デジタルクラフト) 李 賢徳 (産総研) 金 湘宰 (デジタルクラフト)
栗原 眞二 (産総研) ビグズ ジェフ (産総研) 正 安藤 慶昭 (産総研)

Minho JANG, Digital Craft Inc., minho-jang@aist.go.jp

Hyundeok LEE, National Institute of Advanced Industrial Science and Technology
Sangjae KIM, Digital Craft Inc.

Shinji KURIHARA, National Institute of Advanced Industrial Science and Technology
Geoffrey BIGGS, National Institute of Advanced Industrial Science and Technology
Noriaki ANDO, National Institute of Advanced Industrial Science and Technology

Augmented Reality (AR) is technology to augment virtual computer generated image into image of real world environment. Recently some libraries for AR technology are available, and ARToolKit is the one of famous toolkit for AR. In this paper, ARToolKit RT-Component is developed for various AR systems. Reusability and modularity is discussed and, a component design and implementation is shown.

Key Words: 画像処理, RT コンポーネント, Reusability

1. はじめに

拡張現実 (Augmented Reality : 以下 AR) システムを RT コンポーネント (Robot Technology Component) 化することにより, 再利用可能な AR システムの研究開発を行っている. AR はコンピュータを用いて, 実世界に付加情報として仮想的な物体や情報を合成提示する技術として, 様々な分野で応用が期待されている.

本稿では, 拡張現実を比較的容易に実現するためのソフトウェアツールキットとして注目を集めている ARToolKit を RT コンポーネント化 (以下, ARToolKit-RTC) する方法についてモジュール分割とデータポートのデータ型の観点から議論する.

2. ARToolKit 概要

ARToolKit (Augmented Reality Tool Kit) は Kato [1] により開発された AR システムを容易に構築するためのライブラリである. 現在は米国の ARToolWorks 社から GPL と商用のデュアルライセンス形式でリリースされている. GPL (GNU General Public License) バージョンは sourceforge からダウンロード可能である [2]. 本稿では, GPL バージョンの ARToolKit 2.72.1 を使用する [3].

2.1 拡張現実 (Augmented Reality: AR)

拡張現実とは, コンピュータを用いて, 実世界に付加情報として仮想的な物体や情報を合成提示する技術である. 実世界にある物体に説明的な情報や関連情報を表示することで, 新たなサービスを提供できる可能性があることから, 研究のみならず産業的にも期待されている技術である.

拡張現実システムにおいて最も重要な技術は, 使用者が対象を観察する位置・姿勢をカメラ画像などからリアルタイムに計算するアルゴリズム部分である. ARToolKit では, 画像処理技術を用いてマーカを認識し, カメラとマーカの相対位置および姿勢を計算し, マーカ座標系上に仮想的な物体の映像を合成提示している.

2.2 ARToolKit アーキテクチャ

ARToolKit はマルチプラットフォームとして Windows, Linux, SGI Irix, Mac OS X に対応したライブラリが提供されている. 提供される関数は大別すると 4 つのモジュールから構成される. ARToolKit のアーキテクチャを図 1 に示す.

ARToolKit の機能の中心となるマーカ認識と位置・姿勢を計算する部分である “AR Calculation Module” の他に, カメラからの画像取得を行う “Video Capture Module”, カメラ画像に OpenGL とその補助ライブラリである GLUT (OpenGL Utility Toolkit) を用いて三次元モデルを合成す

る “Graphics Processing Module”, OpenVRML を利用し, 描画する三次元モデルを VRML データから作成するための “VRML Drawing Module” などから構成される.

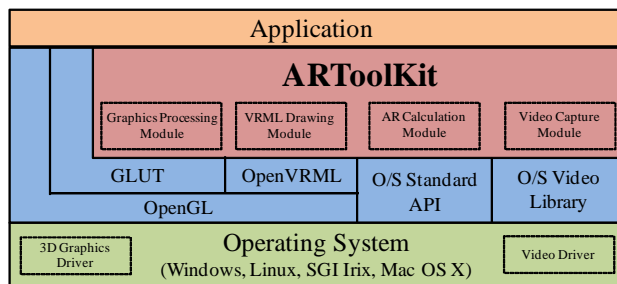


Fig.1 ARToolKit Architecture

3. RT コンポーネント設計

本節では, RT コンポーネントの設計について説明する.

3.1 RT コンポーネント構成

ARToolKit RT コンポーネントの構成図を図 2 に示す.

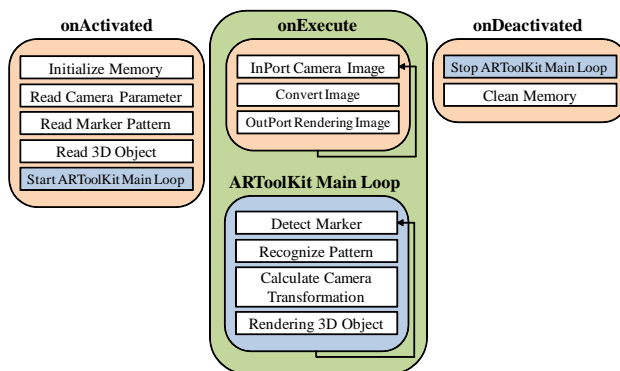


Fig.2 System layout of ARToolKit-RTC

RT コンポーネントの活性化処理を行う onActivate 関数では, メモリイメージの初期化を行うとともに, ARToolKit 側で必要となるカメラパラメータ, マーカパターンファイル, 三次元オブジェクトの VRML ファイルをロードする. こう

することで、コンポーネントを再起動することなく、カメラやマーカ、三次元オブジェクトの変更を行うことができる。

その後、別スレッドで ARToolKit のメインループを呼び出すと同時に、RT コンポーネントのメインループである、onExecute 関数内で、InPort から入力されたカメラ画像を ARToolKit で処理可能な型に変換し、ARToolKit へ渡す。ARToolKit 側で処理が行われた合成画像は onExecute 側でビットマップイメージとして取得され、OutPort から出力される。

3.2 インターフェース

ARToolKit-RTC は 1 つの InPort と 1 つの OutPort で構成される。インターフェースの詳細を図 3 に示す。


RTC	Configuration	Port	Data Details	Data Type
	<ul style="list-style-type: none"> •Camera Parameter File •Marker File •VRML File •Monochrome Thresh Value 	InPort	Camera Image	RTC::CameraImage
		OutPort	Camera Image + 3D Object	RTC::CameraImage

Fig.3 Interface of ARToolKit-RTC

InPort から入力されたカメラ画像上で、マーカが認識されると三次元オブジェクトが描画される。カメラ画像と三次元オブジェクトを合成した画像データが OutPort から出力される。InPort および OutPort では、画像情報を格納するためのデータ型である、CameraImage データ型を利用する。

本 RT コンポーネントでは、設定可能ないくつかのパラメータが、RT コンポーネントのコンフィギュレーションパラメータとして定義されている。コンフィギュレーションパラメータの詳細を以下に示す。

カメラパラメータファイル名 使用するカメラの焦点距離やレンズ歪等のパラメータ情報を記載したファイルを、コンポーネントを実行した場所からの相対パスで指定する。カメラが変更された場合、このファイルを入れ替えるだけで対応可能である。

マーカファイル名 ARToolKit が認識するマーカ画像のファイル名を指定する。認識するマーカを実行時に変更することが可能である。

VRML モデルファイル名 描画する三次元モデルの VRML ファイル名を指定する。このファイル名を変更することで、描画する三次元オブジェクトを実行時に変更可能である。

2 値化の閾値 マーカ認識のための 2 値化の閾値を指定する。マーカの認識率を向上させるために 0~255 の範囲で変更することが可能である。

3.3 CameraImage データ型

ARToolKit-RTC のデータポートに使用するデータ型として CameraImage データ型を用いる。CameraImage データ型はカメラ等のイメージ情報を格納するための構造体である。OpenRTM-aist-1.0.0-RELEASE の IDL (Interface Definition Language) ファイルである InterFaceDataType.idl の中で定義されている。構造体定義を以下に示す。

```

struct CameraImage
{
    Time tm; // Time stamp
    unsigned short width; // Image pixel width
    unsigned short height; // Image pixel height
    unsigned short bpp; // Bits per pixel
    string format; // Image format
    double fDiv; // Scale factor for images
    sequence<octet> pixels; // Raw pixel data
};

```

4. 仮想空間の構築

本節では、ARToolKit を用いて仮想空間を構築する方法を説明する。

4.1 マーカ認識

ARToolKit では、カメラ映像の中からマーカのパターンを認識するための前処理として、入力画像を二値化している。上述のように、閾値を 0~255 の範囲で指定可能であり、一般的には 100~130 付近の値が適当であるが、カメラ、環境光等により変化する。閾値を 100 にした場合の 2 値化後の画像を図 4 に示す。



Fig.4 Monochrome for Marker tracking

ARToolKit において最も重要な関数の一つである arGetTransMat() は、検出されたマーカ情報から、カメラ視点に対する三次元位置・姿勢を推定する。これはすなわち、カメラ座標系を

$$p_c = [X_c \ Y_c \ Z_c \ 1]^T,$$

マーカ座標系を

$$p_m = [X_m \ Y_m \ Z_m \ 1]^T,$$

としたとき

$$p_c = T p_m \quad (1)$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (2)$$

である、同次変換行列 T を推定することに他ならない。実際には、式 (2) の $r_1 \sim r_9, t_{\{x,y,z\}}$ を arGetTransMat() が 3×4 の 2 次元配列 patt_trans[3][4] (式 (3)) に格納し返す。

$$\text{patt_trans}[3][4] = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \quad (3)$$

4.2 三次元オブジェクト表示

上で求めた同時変換行列から、三次元オブジェクトをマーカ座標系上に以下の手順で描画する。

- 座標変換行列を ARToolKit の形式 (para[3][4]) から OpenGL の形式 (para[16]) に変換。
- 値を変更する行列として、モデルビュー行列を指定。
- モデルビュー行列の値を置き換える。

その後、三次元オブジェクトに対して座標変換行列を適用し描画を以下の手順で描画を行う。以下は、ARToolKit の OpenVRML 関数を利用して、VRML ファイルを描画する例である。

```

glClear(GL_DEPTH_BUFFER_BIT); // Z バッファの初期化
glEnable(GL_DEPTH_TEST); // 隠面処理の適用
arVrmlDraw(vrml_id); // VRML の描画
glDisable(GL_DEPTH_TEST); // 隠面処理の解除

```

4.3 オフスクリーンレンダリング

OpenGLで描画したデータは、すべてピクセルとして、画面に表示される。その一連の作業のことをレンダリングと呼ぶ。三次元のレンダリングは現在の一般的なPCにおいては、ビデオカードのGPU (Graphic Processing Unit) とVRAM (Video RAM) 上で行われている。

ARToolKitの機能の再利用性と、他のRTコンポーネントとの連携を考慮するならば、ARToolKitコンポーネントは、マーカ認識と三次元レンダリングのみを行い、実際の描画は描画用RTコンポーネントで行うことが望ましい。たとえば、レンダリング後の画像に、さらに画像処理を施したい場合、ARToolKitコンポーネントからレンダリング後のイメージを出力する必要がある。また、その際にARToolKitコンポーネント自身の表示ウィンドウは表示・非表示を任意に切り替え可能であることが望ましい。

これを実現するために、ARToolKitコンポーネントでは、オフスクリーンレンダリングを行っている。OpenGLにおいてオフスクリーンレンダリングを行うためには、OpenGLの拡張ライブラリであるGLEW (The OpenGL Extension Wrangler Library) が提供するFBO (Frame Buffer Object) を利用する。FBOは、OpenGLのレンダリングを画面上ではなく別途用意したメモリイメージ上で行うための仕組みである。

GLEWを用いてオフスクリーンレンダリングを行う方法を以下に示す。

```
GLuint fbo;
CameraImage outport_image;

// GLEW と FBO の初期化
glwInit();
InitTexture();
InitRenderbuffer();
InitFramebuffer();

// FBO の設定
glBindFramebufferEXT(GL_FRAMEBUFFER_EXT, fbo);
// 三次元オブジェクトの描画
DrawObject();
// FBO のバッファデータの読み込む
glReadPixels(0, 0, inport.width, inport.height, GL_RGB,
             GL_UNSIGNED_BYTE, outport_image.pixels[0]);
// バッファのクリア
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

カメラ画像取得コンポーネントと画像表示コンポーネントをそれぞれ、ARToolKit-RTCの入力と出力に接続し実行した。その際のRTコンポーネントの接続図を図6に示す。

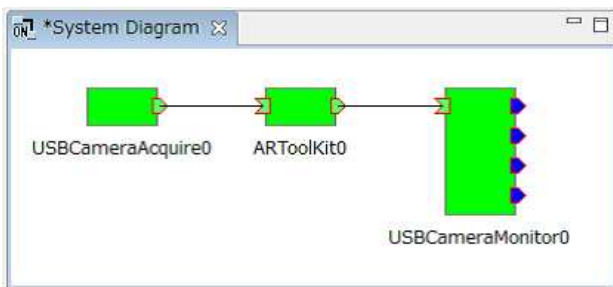


Fig.5 System Diagram of RTC

産総研の女性型ヒューマノイドロボットであるHRP-4CのVRMLモデルファイル [4] を用いて描画した結果を図6に示す。

カメラからの画像のキャプチャを行っているのは、汎用的なカメラコンポーネントであり、画像出力ポートの型がCameraImageデータ型であれば、他のカメラコンポーネントでも接続可能である。また、画像表示も汎用の画像表示コンポー

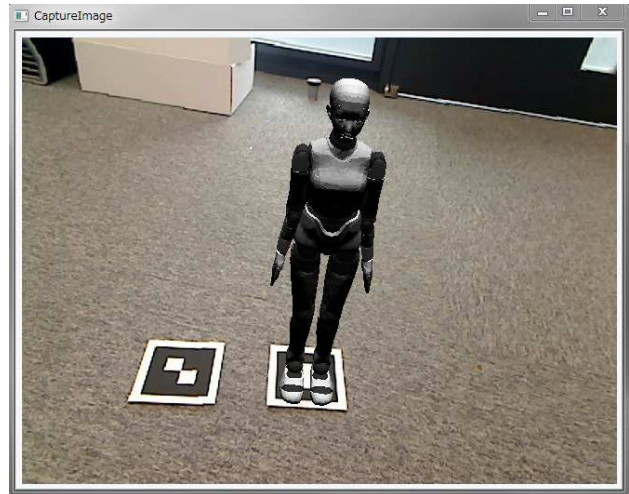


Fig.6 Drawing 3D Object of 「HRP-4C」 Robot model

ネットであり、例えば、ARToolKitコンポーネントと画像表示コンポーネントの間に、任意の画像処理コンポーネントを挿入して、ARToolKitでレンダリングされた画像に処理を加えることも可能である。

5. おわりに

本稿では、拡張現実感システムを実現するライブラリARToolKitをRTコンポーネント化する方法について示した。GLEWライブラリのFBOを利用し、レンダリングをオフスクリーンで行うことで、再利用性や他のコンポーネントとの連携が容易なARToolKit RTコンポーネントを実現した。

作成したRTコンポーネントは、OpenCV [5] 等、画像処理を行うRTコンポーネントと組み合わせることで、従来のARToolKitのみでは難しかった、レンダリング後の画像の処理も容易に実現可能である。

ARToolKitと類似の拡張現実を実現する技術として、ARTag [6] (Columbia Univ.)、BazAR [7] (CVLab, Switzerland)、HandyAR [8] (California Univ.)、PTAM [9] (Oxford Univ.) 等がある。

これらは、リアルタイムに特徴点を自動抽出することで、マーカを使用せずに拡張現実システムを実現している。今後は、こうしたシステムに対してRTミドルウェアを適用しコンポーネント化を行う予定である。

文献

- [1] Kato, H., et al., "Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System", IWAR 99, 1999
- [2] ARToolKit, sourceforge site, <http://sourceforge.net/projects/artoolkit/>
- [3] ARToolKit Home Page, <http://www.hitl.washington.edu/artoolkit/>
- [4] HRP-4C 外装付きモデルファイル, 産業技術総合研究所 H21PRO-1056
- [5] OpenCV Official Page, <http://opencv.willowgarage.com/wiki/>
- [6] ARTag Home Page, <http://www.artag.net/>
- [7] BazAR Home Page, <http://cvlab.epfl.ch/software/bazar/>
- [8] HandyAR Home Page, <http://ilab.cs.ucsb.edu/projects/taehee/HandyAR/>
- [9] PTAM Home Page, <http://www.robots.ox.ac.uk/~gk/PTAM>