



## AR技術を用いた3Dキャラクターの描写に関する研究

メタデータ	言語: jpn 出版者: 宮崎大学工学部 公開日: 2020-06-21 キーワード (Ja): キーワード (En): 作成者: 坂本, 真人, 穂積, 圭輔, Hozumi, Keisuke メールアドレス: 所属:
URL	<a href="http://hdl.handle.net/10458/5901">http://hdl.handle.net/10458/5901</a>

# AR技術を用いた3Dキャラクターの描写に関する研究

坂本 真人<sup>a)</sup> ・ 穂積 圭輔<sup>b)</sup>

## Study on the Depiction of 3D Character by using AR Technology

Makoto SAKAMOTO, Keisuke HOZUMI

### Abstract

The promotion of the tourism nation became the important problem of our country. On the other hand, recently, there is a movement to take advantage of the Augmented-Reality (AR) in tourism. AR is technology to supplement the perception of reality by adding information. In this paper, we concentrate on description of 3D character to perform a guided tourist destination. The algorithm is implemented in the Visual C++ 2008 and C on a personal computer. We hope that this study will be practical.

**Keywords** : AR, C, Tourism, Visual C++, 3D Character

### ・ 1 はじめに

現在日本では、観光事業を力の入れるべき政策の1つとしている。各自治体でも独自に観光に力を入れており、中にはAR技術を活用している自治体も存在する[4]。そこで、ゆるキャラ等をARで表示し、その名所などを紹介することで有意義に過ごしていただくことができ、PRにもなると考えた。以上を踏まえて、本研究では観光地の紹介を行う3Dキャラクターの試作を行う。

### ・ 2 ARの説明

AR(Augmented Reality)とは、人間が視覚的に認識しているものに、情報を付加することによって現実の認識を補足する技術のことである。

#### ・ 2.1 ARの実現方法

##### ・ マーカー型 AR

マーカーを認識することによって付加情報の提示位置を特定し、付加情報を出現させることができる。

##### ・ マーカーレス型 AR

現実の環境に存在している物体や環境そのものを認識することにより提示位置を特定して情報を出現させることができる。

##### ・ ロケーションベース型 AR

GPS などから取得できる位置情報を利用して情報を提示する技術である。

#### ・ 2.2 ARToolKitについて

本研究はARToolKitを用いてマーカー型ARの作成、改良を行う[1]。ARToolKitを用いることで、カメラでマーカーを読み取り、3Dモデルを表示することができる。以下にARToolKitが提供する機能を示す。

a) 工学教育研究部 情報システム工学科担当 准教授

b) 情報システム工学科 学部4年生

- カメラからの画像の取得
- マーカーの検出とパターンの認識
- マーカーの3次元位置・姿勢の計測
- 実写画像と3次元CGの合成

### ・3 プログラム概要

本研究はARToolKitを用いて以下のようにプログラムを作成した。

#### ・3.1 開発環境

- OS : Windows 8.1
- 環境 : Visual C++ 2008
- 言語 : C言語
- ソフトウェア : Softalk, Metasequoia 4

#### ・3.2 今回実装した機能

先行研究では、テキストベースだったものを音声案内に変更した[1]。テキストベースの場合、字が読みづらいため、音声に変更することでその問題を解消できると考えた。また、モデルの向きが固定されており、別視点から見ると見せたい部分とは別の部分が見えてしまう恐れがあったため、モデルが常にカメラの視点に対して正面を向くように表示できるよう変更した。

#### ・3.3 プログラムの概要

1. GLUTの初期化
2. ビデオデバイスの設定
3. カメラパラメータの設定とロード
4. カメラパラメータのサイズ調整
5. カメラパラメータの初期化
6. パターンファイルのロード
7. ウィンドウの設定
8. ビデオキャプチャの開始
9. マウスボタンに関するコールバック関数を登録
10. メインループの開始
11. カメラ画像の取得と描画

#### 12. マーカーの検出と認識

#### 13. 画像のキャプチャ

#### 14. マーカーの信頼度比較

#### 15. マーカーの位置・姿勢計算

: 検出されたマーカーの情報からカメラに対するマーカーの3次元位置、姿勢の計算を行う。

- `arGetTransMat( &marker_info, patt_center, patt_width, patt_trans)`

初回時は上記の関数で座標変換行列を取得し、それ以降の座標変換行列の取得には次の関数を利用する。

- `arGetTransMatCont( &marker_info, patt_trans, patt_center, patt_width, patt_trans)`

カメラとマーカーを静止させていても、表示されているCGが震えて安定感のない表示になることがある。これを安定化させるために初回時に以下の関数で計算する。

- `arGetTransMat()`

それ以降は次の関数を用いて前フレームで取得した行列をもとに計算を行うことでCGが震える現象を多少抑えることができる。

- `arGetTransMatCont()`

なお、ここで `patt_trans` には計算された座標変換行列が格納される。

#### 16. バッファの内容表示

#### 17. 光源の設定

#### 18. 3Dオブジェクトを描画するための準備

: OpenGLのモデルビュー行列を次のように初期化する。

- `argDrawMode3d();`

さらに、以下の関数で射影変換の設定と仮想カメラの射影変換を行う。

- `argDraw3dCamara( 0, 0);`

#### 19. 座標変換行列の適用

: `arGetTransMat()`によって得られた座標変換行列を次のようにモデルビュー行列に適用させる。

- `argConvGlpara( patt_trans, gl_para);`

次に、以下の関数で値を変更する行列の指定を

行う。

- `glMatrixMode( GL_MODELVIEW );`  
さらに、行列の値の変更を以下の関数で行う。

- `glMatrixMode( GL_MODELVIEW );`

## 20. 3D オブジェクトの描画

: 以下の関数で Z バッファの初期化を行う。

- `glClear( GL_DEPTH_BUFFER_BIT );`

## 21. 特定機能の有効化、無効化

### 22. 3D モデルの表示

: 今回はビルボードと呼ばれる手法を応用しモデルの表示を行った[5]。ビルボードとは、常に視点(カメラ)を向く板状のポリゴンである。モデルが常にカメラに対して正面を向いてほしいため、この考え方を用いてモデル表示を行った。以下、関連する関数列挙する。

- マーカー座標系を平行移動する関数  
: `glTranslatef( 0.0, -80.0*size, 40.0*size );`
- 現在の幾何変換行列を取得する関数  
: `glGetDoublev( GL_MODELVIEW_MATRIX, v );`
- モデルビュー行列を初期化する関数  
: `glLoadIdentity();`
- モデルの表示位置の設定する関数  
: `glTranslatef( v[12], v[13], v[14] );`
- モデルの向きを調整する関数:  
: `glRotated( 180, 1.0, 0.0, 0.0 );`  
: `glRotated( angle, 0.0, 1.0, 0.0 );`
- モデルのサイズ変更する関数  
: `glScalef( size, size, size );`
- モデルの描画を行う関数  
: `mgoCallModel( model );`

### 23. 左のボタンが押されたとき

: マーカーを認識している間、以下の関数にしたがって音声再生される[2]。

- `#pragma comment( lib, "wimm.lib" );`  
`PlaySound( _T("sakamoto_ken_2.wav"), NULL, SND_FILENAME | SND_ASYNC )`

### 24. 右のボタンが押されたとき

## 25. 終了処理

### 3.4 Metasequoia による AR モデルの制作

Metasequoia とは、株式会社テトラフェイスが開発、販売している Windows 用 3DCG ソフトウェアである。本研究では、宮崎大学のオリジナルキャラクターである「みやだいもうくん」[3]の 3DCG の作成を行った(図 1)。



図 1: 鳥瞰図。

### 3.5 Softalk による音声ファイルの作成

今回、音声ファイルを作成するために Softalk を使用した。本研究では、「TOSHIBA male adult」という、若い男性をイメージした合成音声に以下の文章を読み上げさせ使用した。

「こちらは坂本研究室です。坂本研究室には院生 3 名、学部 4 年生 6 名、学部 3 年生 6 名が在籍しています。」

### 3.6 プログラムの実行結果

プログラムの実行の際、以下のマーカーを使用した(図 2)。



図 2: 使用するマーカー。

- 音声での案内  
モデルを表示している間、任意のタイミングでの

再生に成功した。

- モデルの向きについて

基本的なモデルの表示と改良したものの比較を行った。失敗した改良、成功した改良ともに示す(図3～図5)。

- ・ 横からの画像



図3：基本的な表示。



図4：失敗した改良例。



図5：成功した改良例。

- ・ 追加機能

離れた距離からマーカーを読み込んだ際、モデルが小さく見えたため、追加で拡大、縮小の機能を追加した。また、モデルがマウスドラッグで回転する機能を追加した。

#### ・ 4 考察

- 音声の再生

音声再生されている間に画面をクリックすると、初めから再生されてしまう問題がある。コールバック関数を実装して音声再生を行う必要がある。

- モデルの表示について

初めはカメラの横の移動に応じてモデルを回転させようとした。しかし、カメラの横の移動量が大きいため、モデルの回転とカメラの挙動が一致しなかった。そこで、プログラムの概要でも述べたビルボードと呼ばれる考え方を応用してモデルの表示を行った。その結果、モデルがカメラの正面を向くという私の意図した表示を行うことができた。しかし、カメラを遠ざけたとき、モデルの位置は固定されたままで、モデル小さく見えてしま

った。これを解決するために、モデルとカメラの距離を保って表示を行う必要がある。

- モデルの拡大・縮小

マーカーとの距離が離れたとき大きさを調節できるようになったが、拡大するたび、モデルが暗くなってしまった。

- モデルの向きを変更

今回はアニメーションの実装までたどり着かなかったが、将来的にアニメーションの機能を実装した際、アニメーションをいろんな視点から楽しむことができるようにするため実装した。

#### ・ 5 おわりに

本研究において観光地で案内を行うキャラクターの試作を行ったが、考察でも述べたが複数の問題が浮き彫りになった。また、本研究では実験をまだ室内でしか行っておらず、実際に外で使用する場合における問題点の抽出を行うことができていない。また、マーカーレスで行う場合やロケーションベースで行う場合の検証も必要であると考えられる。問題点は多々あるが少しずつ改善していき、より実用的なものにしていきたい。

#### 参考文献

- [1] 橋本 直：3D キャラクターが現実世界に誕生！、ARToolKit、拡張現実プログラミング入門、ASCII (2008)。
- [2] 小松 翼：AR 技術を用いた疑似フラ楽器の試作、平成 26 年度卒業論文、宮崎大学工学部 情報システム工学科 (2015)。
- [3] 宮崎大学 理念・目的・シンボルマーク・オリジナルキャラクター：<http://www.miyazaki-u.ac.jp/guide/ideal/ideal> [Online] 。
- [4] 関谷 裕樹：制作が容易な AR を利用した観光 AR の提案、平成 26 年度卒業論文、宮崎大学工学部 情報システム工学科 (2015)。
- [5] OpenGL プログラミングメモ：<http://www21.atwiki.jp/opengl/pages/1.html> [Online] 。