

AR技術を用いた仮想フラ楽器の試作

坂本 真人^{a)}・本田 義和^{b)}

The Experiment of Virtual Musical Instruments for Hula by Using AR Technology

Makoto SAKAMOTO, Yoshikazu HONDA

Abstract

Hula is a Hawaiian traditional dance form accompanied by chant (oli) or song (mele). There are many sub-styles of hula, with the main two categories being Hula 'Auana and Hula Kahiko. Ancient hula, as performed before Western encounters with Hawaii, is called kahiko. It is accompanied by chant and traditional musical instruments such as the Ipu, the Pu'ili, the Uli'uli, and so forth. On the other hand, in general, the musical instruments are expensive, and we need the space for the big instruments. Moreover, various troubles are often taking place by performance of the musical instruments between the neighborhoods. Therefore, in this paper, we deal with the Hawaiian musical instruments by using the AR, and study the production of the virtual Hawaiian traditional musical instruments for Hula experimentally. The algorithm is implemented in the Visual Studio 2010 on a personal computer.

Keywords: AR, Hula, Kahiko, Musical instruments, Virtual reality, Visual Studio

1. はじめに

近年の日本では趣味としてフラが浸透しつつあり、最近ではダイエット効果があることが判明したためフラ人口がさらに増加すると考えられる。今後の普及に伴いフラを始めようとする方が抱える問題がある。1つめは楽器の購入である。古典フラか現代フラによって使用する楽器が違い、何種類も買う必要となった場合負担が大きい。次にフラの練習場所の確保である。フラ教室以外で例えば家で練習しようとした際、家が集合住宅であれば騒音問題に発展する可能性がある。

本論文では、AR技術を駆使し、PCとwebカメラを使用して様々な疑似フラ楽器の表示と実際に手にとった形でフラを踊れるプログラムを試作した。

2. 原理

2.1 ARの説明

拡張現実感技術とは、我々が現実の環境から得る知覚情報にデジタル技術の情報を付加することにより現実世界の情報を強化する技術である。英語ではAugmented Realityであり、一般的には頭文字をとっ

てAR技術と呼ばれている。現在、AR技術はコンピュータ処理能力向上やAR向けのデバイスが安価になったこともあり、AR技術を用いた商品の研究・開発が活発になっている(図1参照)。



図1. AR商品化の一例³⁾.

今回行うAR表現方法はマーカーARである。これは、マーカーから情報を読み取ってマーカー上に情報を表示させる方法である。商品化されているARはこの表示方法が一般的であり、ある程度の知識があれば簡単に試せるというメリットがある。試作段階ということで、本論文では容易にプログラミングできるマーカーARを採用した¹⁻³⁾。

2.2 アルゴリズム

2.2.1 複数マーカーによる表現

a) 情報システム工学科准教授

b) 情報システム工学科学部生

今回はAR マーカーをサイコロ状にした。(図2 参照) カメラでAR マーカーを読み取る際に少しでもマーカーが隠れてしまうと表示することができず、マーカーを持って踊ることを考えた場合に隠れるおそれがある。マーカーをサイコロ状にして各マーカーで表示するモデルの角度を調整したことにより、どの角度から見ても表示できるようにした。

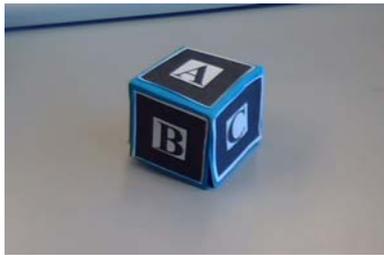


図 2. 複数のマーカー。

(1) マーカーの定義

マーカーを読み込ませるために A, B, C, D, F のマーカーがあることを指定する。

```
#define COUNT 5
* patt_name[5] = { "Data/patt.a",
  "Data/patt.b", "Data/patt.c", "Data/patt.d",
  "Data/patt.f" };
第1～5引数：パターンファイル名
```

上記5つのパターンファイルを読み込む。

```
for(i=0; i<COUNT; i++){
if ( ( patt_id[i] = arLoadPatt(patt_name[i])) <
0 ) {
printf("パターンファイルの読み込みに失敗しました\n");
return -1;
}
}
```

マーカーの個数 (COUNT) 分信頼度の比較をして patt_id に一致し、かつ信頼度が最も高いマーカーの番号を k に格納する。そして比較した回数を a とする。

```
for(a=0; a<COUNT; a++){
k = -1;
for( j = 0; j < marker_num; j++ ) {
if ( patt_id[a] == marker_info[j].id ) {
if ( k == -1 ) k = j;
else if ( marker_info[k].cf <
```

```
marker_info[j].cf ) k = j;
}
}
```

3D オブジェクトの描画をするために a の値をおく

```
DrawObject(a);
```

Switch文でマーカー毎に角度と表示位置の調整していく。

(2) モデルの回転

MetasequoiaではY軸が上を向いている に対し、AR 表示の際はZ軸が上を向いている。これを修正するためにX軸まわりに90度回転させる。

```
glRotatef( 90.0, 1.0, 0.0, 0.0 );
第1引数：回転する角度 (単位：度)
第2～4引数：回転軸となるベクトル
```

(3) モデルの平行移動

モデルが複数表示する際にきれい重なるようにするため、サイコロの辺の長さの30mm分調整をおこなう。

```
glTranslatef( 0.0, -30.0, 0.0 );
引数：X, Y, Zの各軸の移動量 (単位：ミリメートル)
```

2.2.2 MetasequoiaによるARモデル制作

本論文では、メタセコイアの独自形式である mqqo 形式でARモデルを表示させる。mqqo形式の3Dポリゴンモデルをプログラムで表示させるために、橋本氏¹⁾が開発したMQOファイルを読み込んでOpenGLで表示するためのC/C++用のライブラリであるGLMetaseqを使用する。GLMetaseq.hとGLMetaseq.cをプログラムのプロジェクトに追加し、プログラム中でインクルードする。

(1) GLMetaseqの初期化

最初に初期化を行う。argInit()関数の後に行う。

```
mqqoInit();
```

(2) MQOファイルの読み込み

MQOファイルからモデルデータを読み込むために使用する。MQOファイルをロードし、MQO_MODEL型の変数にモデルの情報が記録される。

```
MQO_MODEL model;
```

```
*mqo_name= "Data/パイリ3.mqo";
model = mqoCreateModel( mqo_name, 1.0 )
第1引数：モデル名
第2引数：スケール (1.0で当倍)
```

(3) モデルの描画

ファイルから読み込んだモデルを描画するために使用し、実行することにより3Dポリゴンモデルを描画できる。

```
mqoCallModel( model );
```

(4) モデルの消去

プログラムの終了時に不要になったモデルを削除させるために使用する。ファイルの読み込みを行ったすべての変数に対してモデルの削除を行う。

```
mqoDeleteModel( model );
```

(5) GLMetaseqの終了

プログラムの終了時にGLMetaseqの終了処理を行うために使用される。プログラムを終了させる際、mqoDeleteModel() 関数でモデルを削除後にこの関数を使用する。

```
mqoCleanup();
```

2.2.3 wave ファイルの再生

waveファイルを再生するためにplaysound() 関数を使用する。この関数を使用する場合にはwinmm.libというライブラリファイルを#pragmaでリンクする必要がある。

```
#pragma comment(lib, "winmm.lib")
PlaySound( _T("パイリ2.wav"), NULL,
SND_ASYNC|SND_NOSTOP);
第1引数：waveファイル名
第2引数：HMODULE型
第3引数：再生フラグ
SND_ASYNC：非同期再生
SND_NOSTOP：前のサウンドが流れている場合は再生しないようにする。
```

第3引数を上記のようにした理由が音と画像を連携させるためである。SND_ASYNC だけの場合、playsound の処理を実行しようとするときにすぐにマーカーを読み取ってまた playsound の処理になってしまうため音が流れなくなってしまう。

しかし、SND_NOSTOP があれば playsound の処理中にマーカーを読み取っても処理が継続されるため、音がならないという問題が解決する。

3. 実行結果

3.1 開発環境

本研究では開発環境に visual studio 2010 professional およびC言語を用いてプログラミングを行った。また、ARモデルの制作にMetasequoiaLE R3.0を使用した。

3.2 シミュレーション結果

フラ楽器のARモデル表示との結果を以下に示す(図3~5参照)。AR表示では時々モデルが重なったり、表示されなかったり、安定性に欠けていた。一方、音の再生では画面が停止するなどの問題もなく鳴らすことができた。

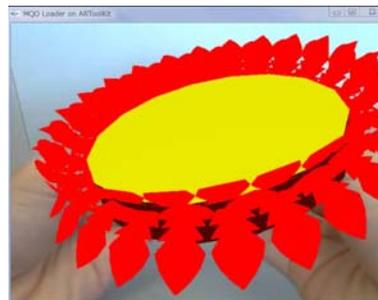


図 3. ウリウリの AR モデル。

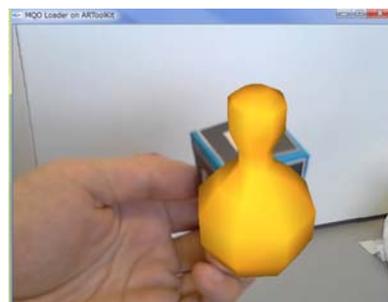


図 4. イプの AR モデル。

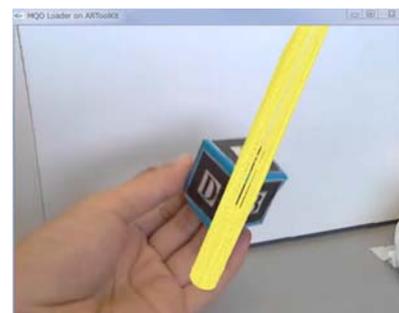


図 5. パイリの AR モデル。

4. 考察

実行結果から AR モデル表示と音の再生という最低限の実装はできた。しかし、AR モデルの表示の正確性がないことや音の再生でいつ鳴らすかなどの設定ができていないという課題が残っている。

AR モデル表示の正確性では WEB カメラの性能に大きく左右されてしまう。最近では WEB カメラが内蔵されている PC が販売され、安価で買えるため WEB カメラ自体はすぐに入手できるが、WEB カメラの解像度、焦点距離、レンズ歪みなどの特性上モデル表示の正確性が欠けてしまう場合がある。一般的な性能の WEB カメラでより正確に表示ができるよう工夫しなければならない。

また、音の再生の設定では 1 人のフラダンサーの踊りを対象にプログラム中の条件文を設定することは安易であるが、複数のフラダンサーがこのソフトを使用するとダンスの種類や癖などを踏まえて条件を絞ることがとても難しい。

5. おわりに

前述したように、日本では趣味としてフラが浸透しており、最近ではダイエット効果があることが判明したためフラ人口がさらに増加している。2012 年に宮崎で開催された「モク・オ・ケアヴェ・インターナショナル・フェスティバル日本大会」を見に行き、実際にフラやハワイの文化に触れることができた。大会を勝ち抜いてきた強豪ハラウのチームによる力強く美しいフラにとっても感動したのと同時に、練習を相当積み重ねてきた成果であると感じた。

本研究では、AR で疑似フラ楽器を試作した。フラの練習の一助にしてもらいたいため今後考察で述べたような問題点の改善を行いたい。そして、他のフラ楽器の AR モデル制作や、マーカーレス AR で実装していくことでより扱いやすいプログラムにしていきたい。

最後に、日頃お世話になっている宮崎市ブアマエオレ・メレ・フラクラブ主宰の海野比呂美先生、神戸市ケ・アラ・スクール・オブ・フラ主宰のカオル ケアラヌヘアオブアラニ マエダ先生に深く感謝の意を表す。また、両主宰の家本である著名なクム・フラ Kawaikapuokalani k. Hewett 先生に敬意を表す。

参考文献

- 1) 橋本直：3D キャラクターが現実世界に誕生！
ARToolKit 拡張現実感プログラミング入門，
ASCII，2008.

- 2) 工学ナビ, ARToolKit を使った拡張現実感プログラミング. [Online]

http://kougaku-navi.net/ARToolKit/index.html#Overview_of_program

- 3) 任天堂, NINTENDO 3 DS. [Online]

<http://www.nintendo.co.jp/3ds/software/built-in/ar/>