



AR技術を用いた仮想フラ楽器の試作 (第II報)

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, 小松, 翼, Komatsu, Tsubasa メールアドレス: 所属:
URL	http://hdl.handle.net/10458/5593

AR 技術を用いた仮想フラ楽器の試作 (第 II 報)

坂本 眞人^{a)}・小松 翼^{b)}

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

Makoto SAKAMOTO, Tsubasa KOMATSU

Abstract

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, we have dealt with the Hawaiian musical instruments by using the AR technology, and studied the production of the virtual Hawaiian traditional musical instruments for Hula experimentally. In this paper, we continue the fundamental study on this study, and show the program for this AR technology that was updated. The algorithm is implemented in the Visual C++ 2010 Express and C on a personal computer.

Keywords: AR, Hula, Kahiko, Musical instruments, Virtual reality, Visual C++

1. はじめに

フラはポリネシア文化の影響を受けて古代ハワイで誕生し、豊作や戦などに対して神々への祈りを捧げるためのものであったが、その後生活に結びつく様々な踊りも始まり、カラカウア王の時代には古典フラだけではなく、欧州の文化と融合させた現代フラも開花した。以後、優雅なフラのスタイルが次々と築き上げられたのである。日本では、近年フラがすっかり浸透し、最近ではダイエット効果やリラクゼーション効果があることが判明したため今後のフラ人口の増加が考えられる。これに伴いフラを始めようとする方が抱える問題がある。1つ目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に負担が大きい。2つ目は練習場所の確保である。フラ教室以外で、例えば自宅で練習しようとした際、自宅が集合住宅などであれば騒音問題に発展する可能性がある。

本論文では、先行研究である「AR を使用した疑似フラ楽器」に改良を加え、より実用的なものを目指した[2]。

2. 原理

2.1 AR の説明

AR (Augmented Reality) とは、我々が現実の環境から

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

受ける知覚情報などにコンピュータで作り出された情報を付加することで、現実世界の情報を強化する技術である。現在では、コンピュータの処理能力が向上して AR 向けのデバイスが比較的安価になったことにより、AR 技術の研究や商品化が活発になっている[3]。

今回用いる AR 表現方法は前回同様にマーカー型 AR である。この場合、マーカーは認識の対象となるためのある特定のパターンを持った図形であり、認識されることにより付加情報の提示位置を特定し、付加情報の出現キーとなるマーカーを利用する。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。

2.2 アルゴリズム

2.2.1 マーカーによる表現

今回は 2 つの方法を考え、2 種類の AR マーカーを用いた[4]。

方法 1 では、サイコロ状の AR マーカーを用い、同一のマーカーでモデルと音を表現するようにした。マーカーは 2 つ (B, C) に絞り、B を高音、C を低音とした (図 2.1)。

方法 2 では、図 2.1 のマーカーの下方に新たなマーカーを追加し、モデル表示と音の再生を分けた。モデル表示を (B, C)、音の再生を高 (高音)・低 (低音) とした (図 2.2)。



図 2.1 : 方法 1 の AR マーカー。



図 2.2 : 方法 2 の AR マーカー。

(1) マーカーの定義

ここでは、方法 2 についての定義を記す。マーカーを読み込ませるために高,低,B,C のマーカーがあることを指定する。

```
#define PATT_ID_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) だけ信頼度の比較を行い、patt_id に一致して、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 の値を送る。

```
DrawObject(a);
```

Switch文でマーカー毎に角度と表示位置を調整する。

(2) モデルの回転

MetasequoiaではY軸が上を向いているのに対し、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 30 mm分調整を行う。

```
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

本研究では、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/イプ.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) モデルの削除
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("イプ (高音).wav"), NULL,
SND_ASYNC|SND_PURGE);
```

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：サウンドの再生を停止する。

第3引数を上記のようにした理由はモデル表示と音の再生を分けるためである。SND_ASYNC だけの場合、playsound の処理を実行しようとするたび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_PURGE

があれば playsound の処理中に音が連続再生する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 C++ 2010 Express 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Metasequoia4を使用した。

3.2 シミュレーション結果

フラ楽器のARモデル表示の結果を以下に示す(図3.1)。ARモデルの表示では、モデルが表示されなかったり、同時に複数表示されたりと安定性に欠けていた。また、音の再生では鳴らしたいタイミングで音を鳴らせるようにはなったが、不意に音が鳴ってしまうことがあり、こちらも安定性に欠けていた。



図 3.1：イプのARモデル。

4. 考察

実行結果から、鳴らしたいタイミングで音を鳴らせるようになったので楽器に近づけることはできた。しかし、ARモデルの表示の正確性、不意に音が鳴ってしまうといった課題が残っている。

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カメラ自体は安価で販売されており入手しやすいが、解像度、焦点距離、レンズの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モデル表示ができるように工夫しなければならない。

また、音の再生でもマーカーを必要としているので、WEBカメラの性能に左右されてしまう。そして、今回は1人を対象にプログラム中の条件文を設定したが、複数人がこのソフトを使用するとダンスの癖などを踏まえて条件を絞ることが難しい。

5. おわりに

前述したように、フラは誰でも気軽に始めることができ、最近ではダイエット効果やリラクゼーション効果があることもわかり、今後のフラ人口の増加が期待される。また、フラのように気軽に他国の文化に触れることができる機会はそう多いものではないので大切にしていきたい。

本研究では、イプと呼ばれるひょうたんによって出来た打楽器の疑似フラ楽器を試作した。今後は考察で述べた問題点などをさらに改善し、他のフラ楽器についても疑似楽器を試作していくことで、バラエティに富んだより実用的なものにしていきたい。

謝辞

最後に、日頃お世話になっているプアマエオレ・メレ・フラクラブ主宰の海野比呂実先生（宮崎市）、ケ・アラ・スクール・オブ・フラ主宰のクムフラ カオル ケアラアヌヘアオブアラニ マエダ先生（神戸市）に深く感謝の意を表す。また、両主宰の家元である著名なロエア（クムフラの師）Kawaikapuokalani K. Hewett 先生（ハワイ州）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] 本田 義和：ARを使用した疑似フラ楽器の試作、平成24年度卒業論文、宮崎大学工学部情報システム工学科.
- [3] ASCII.jp×デジタル. [Online]
<http://ascii.jp/elem/000/000/514/514146/>
- [4] 工学ナビ Engineering Navi. [Online]
<http://kougaku-navi.net/ARToolKit/>