

運動イメージと遂行動作のフィードバックシステム

Development of Feedback System of Motor Imagery and Performance

岡 本 敦* 鶴 原 清 志** 船 橋 孝 恵***
横 山 慶 子**** 水 藤 弘 吏***** 池 上 康 男*****
山 本 裕 二*****

Atsushi OKAMOTO* Kiyoshi TSURUHARA** Takae FUNABASHI***
Keiko YOKOYAMA**** Hiroshi SUITO***** Yasuo IKEGAMI*****
Yuji YAMAMOTO*****

We developed a feedback system of motor imagery and actual performance. For the visualization of motor imagery, we used a puppet-type motion-capture system with 3D animation software. To create computer graphics from actual movement, we captured the actual movement by digital video and analyzed the movement using motion-analysis software. We converted the kinematics of the movement to the BVH format to create computer graphics of actual movements. The movements of dart throwing were examined using this system, and learners of the game could compare their own motor imagery to their actual performance. This feedback system should be useful in motor-learning processes.

1. はじめに

運動遂行前にこれから行う運動のイメージを描くことは運動学習の中ではよく見られる。そして運動後に遂行動作と遂行前の運動イメージとの違いが比較可能となれば、その運動イメージを修正することによって運動学習が促進されると考えられる。本研究ではダーツ投げを課題として、運動前のイメージと実際の遂行動作のCG（コンピュータ・グラフィックス）を作成し、両者を比較することが可能なシステムの構築を試みた。

2. 方法

2.1 運動イメージの可視化

運動イメージの可視化はオー・エイ・エス株式会社製のマリオギア（人形式3DCG モーションキャプチャーシステム、図1）を用いた。マリオギアはポテンシオメータによって関節の角変位を検出することによって、人形の操作に同期して画面上の3Dモデルを動かす、操作者が直感的に身体動作を作成できるシステムである。このマリオギアとAutodesk社製MotionBuilderによって遂行前の運動イメージのCGを製作した。

具体的には、操作者がいくつかの姿勢をマリオギアで作成し、この作成された姿勢をMotionBuilderのキーフレームとして登録する。その後、作成されたいくつ

* 東海学園大学
** 三重大学教育学部
*** 九州大学大学院人間環境学府
**** 名古屋大学大学院教育発達科学研究科
***** 名古屋大学総合保健体育科学センター
* Tokai Gakuen University
** Faculty of Education, Mie University
*** Graduate School of Human-Environment Studies, Kyushu University
**** Graduate School of Education and Human Development, Nagoya University
***** Research Center of Health, Physical Fitness, and Sports, Nagoya University



『MarioGear』は、ビデオ撮影された映像を背景に置き、人形の形をしたセンサーシステムを操作して、画面上の3Dモデルにキーフレームごとにポーズを与え、キーフレーム間を中間補完して自然でスムーズなモーションデータを生成させるものです。

【必要システム】

OS : Windows98、Windows2000
 グラフィックAPI : DirectX7a以上
 メモリ : 128Mバイト以上
 インターフェイス : RS232C(USB 変換も可能)

【MarioGear 基本仕様】


人形 : センサーユニット40個 (最大64個/センサーBOX)
 付属ソフト : オーサリングソフト、キャリブレーションソフト
 キャプチャーレート : 40fps (CPU、ビデオボードに依存します)
 センサーBOX : 1個 (64チャンネル)
 その他 : 備品


【センサー 精度】


回転範囲 : 270度
 解像度 : 0. 25度


図1 マリオギアの仕様

イメージの CG



構え



バックスイング



リリース



フォロースルー

遂行動作の CG



構え



バックスイング



リリース


フォロースルー

遂行動作のビデオ映像


構え


バックスイング


リリース



フォロースルー

図2 イメージのCG、遂行動作のCGと遂行動作のビデオ映像の比較

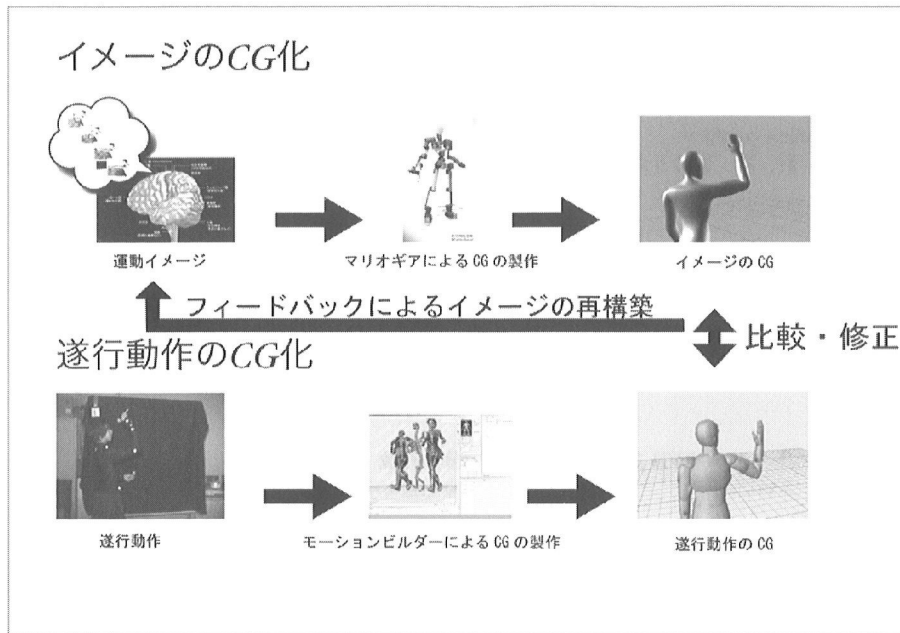


図3 イメージと遂行動作のCG製作の流れとその比較によるフィードバック

かのキーフレーム間の角度変化を Bézier 曲線で補間することによって、一連の動作が3次元CGのアニメーションとして作成される。その後、各姿勢間の時間間隔を操作者が調整することによって、実際の運動イメージに最も近いと思われる動作になるように調整することができる。

2.2 遂行動作のCG化

実際の動作はビデオカメラによって側方より2次元で撮影し、得られたビデオ映像を株式会社ディケイエイチ製のFrame-DIASによって座標解析し、身体各関節の変位を求めた。なお、座標解析を容易にするために、関節位置には反射マーカを貼付した。

この身体各関節の変位から関節の角変位の時系列データを求め、BVH形式のファイルに変換した(付録参照)。そしてBVH形式のファイルをMotionBuilderで読み込み、実際の身体動作のCGを製作した。

3. 結果と考察

本研究で製作したイメージと遂行動作のCG並びに実際の遂行動作を図2に示した。ここではダーツ投げ

を課題として用いたが、遂行動作のCGの製作がビデオ映像を座標解析した2次元座標データであったため、若干、肩関節角度や前腕の回内・回外に誤差が生じた。しかし、図2に見られるように運動イメージと遂行動作をCG化することによって、試技者が両者を比較可能であることが示された。

運動イメージと遂行動作を同じようにCGとして表現することによって両者を比較することが可能となり、学習者が運動イメージを修正、再構築することによって、運動学習が促進されることが考えられる。今後、遂行動作のCG化にモーションキャプチャーシステムを使うことによって実際の身体動作のCG化が容易となるだけでなく、3次元での動作解析が必要とされる動作のCG化も容易となり、より実用的なシステムの構築が可能になると考えられる。

謝辞

本研究の一部は、文部科学省科学研究費補助金基盤研究(B)(No. 17300206)の補助に基づくものである。記して謝意を表す。

(2008年1月10日受付)

付録 動作解析結果から角度変位を求め BVH フォーマットのファイルに変換するプログラム

```

/*
dias2bvvh.c
dias2bvvh <BVH_head file> <infile> <outfile>
2次元 DLT で、小指、手首、肘、肩、腰の5点を取り、そこから肘関節、肩関節の角度を求め、Mocap で CG がかかるように BVH 形式のデータに変換する
*/

#include "stdio.h"
#include "math.h"

#define MAXDT      3000
#define MAXFL      6
#define MAXDM      2
#define  MAXCOM    191 /*      BVH ファイルのヘッダ部分の行数      */
#define PI         3.14152
#define  FINGER    4
#define  WRIST     3
#define  ELBOW     2
#define  SHOULDER  1
#define  HIP       0
#define  MAXBVH    34
#define  BVHXYZ    3

/* Data buffer */
float
    da[MAXFL][MAXDM][MAXDT],sx1[MAXDT],sx2[MAXDT],sy1[MAXDT],sy2[MAXDT],sx3[MAXDT],sy3[MAXDT],sx4[MAXDT],sy4[MAXDT],sx5[MAXDT],sy5[MAXDT];
float  bvh[MAXBVH][BVHXYZ][MAXDT];
float  hand[MAXDT],f_arm[MAXDT],u_arm[MAXDT],trunk[MAXDT];
char   comment[128],label1[128],label2[128],label3[128];
float  filmspeed,A0,B1;
int    dtpnts,n_fl,release;
int    cutoff;
float  sampling;
FILE   *fp,*frp;
char   *bvhflnm,*datflnm,*outflnm;

main(c,v)
int c;
char **v;
{
    int    rdbvhfl(),rddatfl(),sign(),n_fl,i,j;
    float  trans();
    float  angletan2();

    if (c<3)    error();
    else
    {
        bvhflnm = *(++v);
        datflnm = *(++v);
        outflnm = *(++v);
    }

    rdbvhfl();

/* 引数で与えられたデータファイルを読み込み、データ数を返す
n_fl=rddatfl();
*/

```

```

printf("data number = %d\n",n_fl);

/*      それぞれの点を別の配列に置きなおす      */
trans(n_fl,da,WRIST,0,sx2);
trans(n_fl,da,WRIST,1,sy2);
trans(n_fl,da,ELBOW,0,sx3);
trans(n_fl,da,ELBOW,1,sy3);
trans(n_fl,da,SHOULDER,0,sx4);
trans(n_fl,da,SHOULDER,1,sy4);

printf("trans end\n");

/*      前腕ベクトル (hand->f_arm) の角度      */
angletan2(n_fl,sx2,sy2,sx3,sy3,f_arm);

/*      上腕ベクトル (f_arm->shoulder) の角度      */
angletan2(n_fl,sx3,sy3,sx4,sy4,u_arm);

wtf(n_fl,outflnm);

return( 1 );
}

int rdbvhfl()
{
int count=0;
int i,j;
int nf;
int dummy,dummy2;
char status[100];

if(fp=fopen(bvhflnm,"r"))
{
if(frp=fopen(outflnm,"w"))
{
printf("Now reading data file %s\n",bvhflnm);
printf("Please wait!\n\n");
for(i=0;i<MAXCOM;i++)
{
fgets(comment,128,fp);
{
fputs(comment,frp);
}
}
fclose(frp);
fscanf(fp,"%s%d",label1,&nf);
printf("%s\t%d\n",label1,nf);
fscanf(fp,"%s%s%f",label2,label3,&filmspeed);
printf("%s %s\t%f\n",label2,label3,filmspeed);
for(i=0;i<5;i++)
for(j=0;j<MAXBVH;j++)
{
fscanf(fp,"%f%f%f",&bvh[j][2][i],&bvh[j][0][i],&bvh[j][1][i]);
}
}
fclose(fp);
}
else
fprintf(stderr,"Can't open the file!!\n");

return( nf );
}

```

```

}

int rddatfl()
{
    int count=0;
    int i,j;
    int nf;
    int dummy,dummy2;
    char status[100];

    if(fp=fopen(datflnm,"r"))
    {
        printf("Now reading data file %s\n\n",datflnm);
        printf("Please wait!\n\n");
        fscanf(fp,"%d,%d,%*f",&nf,&dtpts);
        printf("n = %d\tchnl = %d\tfreq = %f\n",nf,dtpts,filmspeed);
        for(i=0;i<nf;i++)
        {
            for(j=0;j<dtpts;j++)
            {
                fscanf(fp,"%f,%f",&da[j][0][i],&da[j][1][i]);
            }
            count++;
        }
        fclose(fp);
    }else
        fprintf(stderr,"Can't open the file!!\n");

    return( count );
}

wtfi(nf,wtfnm)
int nf;
char *wtfnm;
{
    int i,j;

    if(fp=fopen(wtfnm,"a"))
    {
        printf("Now writing new data file <%s>\n\n",wtfnm);
        printf("Please wait!\n\n");
        fprintf(fp,"%s\t%d\n",label1,nf+5);
        fprintf(fp,"%s %s\t%f\n",label2,label3, filmspeed);
        /* T スタンスを 5 コマ分元のファイルから読み込んだ分を書き込む */
        for(i=0;i<5;i++)
        {
            for(j=0;j<MAXBVH;j++)
                fprintf(fp,"%f\t%f\t%f\t",bvh[j][2][i],bvh[j][0][i],bvh[j][1][i]);
            fprintf(fp,"\n");
        }
        /* 肩と肘の Y 軸周りの角度, 29 番目の Y と 31 番目の Y だけに実測値を書き込む */
        for(i=0;i<nf;i++)
        {
            for(j=0;j<28;j++)
                fprintf(fp,"%f\t%f\t%f\t",bvh[j][2][i],bvh[j][0][i],bvh[j][1][i]);
            fprintf(fp,"%f\t%f\t%f\t",bvh[28][2][i],bvh[28][0][i],-u_arm[i]); /* 肩の角度 29 */
            fprintf(fp,"%f\t%f\t%f\t",bvh[29][2][i],bvh[29][0][i],bvh[29][1][i]);
            fprintf(fp,"%f\t%f\t%f\t",bvh[30][2][i],bvh[30][0][i],-(f_arm[i]-u_arm[i])); /* 肘の角度 31 */
            fprintf(fp,"%f\t%f\t%f\t",bvh[31][2][i],bvh[31][0][i],bvh[31][1][i]);
            fprintf(fp,"%f\t%f\t%f\t",bvh[32][2][i],bvh[32][0][i],bvh[32][1][i]);
        }
    }
}

```

```

fprintf(fp,"%f\t%f\t%f\n",bvh[33][2][1],bvh[33][0][1],bvh[33][1][1]);
    }
    fclose(fp);
}else
    fprintf(stderr,"Can't open the file!! \n");

return( 1 );

}

error()
{
    printf("parameter error !!\n\n\n");
    printf("parm: input flnm..... output flnm\n");
    exit(1);
    return( 1 );
}

float trans(n,b,pnts,xyz,y)
int n,pnts,xyz;
float *b,*y;
{
    int i;
    for(i=0;i<n;i++)
        y[i]=*(b+pnts*MAXDM*MAXDT+xyz*MAXDT+i);
return(*y);
}

float angletan2(n,x1,y1,x2,y2,ang)
int n;
float *x1,*y1,*x2,*y2,*ang;
{
    int i;
    float vx1,vy1,temp;

    for(i=0;i<n;i++)
    {
        vx1=x1[i]-x2[i];
        vy1=y1[i]-y2[i];

        ang[i]=atan2((double)vy1,(double)vx1)*180.0/PI;
    }
return(*ang);
}

```

