

NUIによる簡易3DVRアニメーション編集システムの構築

2016SC003 浅井裕也

指導教員：河野浩之

1 はじめに

近年のモーションキャプチャ技術の低コスト化とそれを利用したVR(Virtual Reality)の普及[2]には目覚ましいものがあり、今後は個人でのVtuberなど3DVRアニメーション制作の需要が高まってくると予想される。そこで本研究では、家庭用にも販売されているHMD(Head Mount Display)を用いたモーションキャプチャによる低コストで簡易的なアニメーション編集システムを提案する。

しかし、HMDを装着すると、視界が塞がれるため、通常のPC作業には不便がある。そこでHMDを装着したままVR空間内を操作しアニメーション制作・編集やVtuber配信ができるようなインターフェースが必要であると考え、ジェスチャやポーズ認識を用いたNUI(Natural User Interface)の構築を試みた。

2 3DアニメーションとVRについての先行研究

3Dアニメーション制作についての関連研究には、キャラクターのモーション作成の簡易化を目指したものがある。古川らは、Kinectを用いて人間の動きを取得し、各モーション間の関連性を自動的に構造化する手法、および、モーション間のつなが目を滑らかに表示するための3次元形状補間手法を提案している[3]。

ジェスチャ認識を用いたコンピュータなどの操作についての関連研究には、山田らの発表者のジェスチャをスライドオブジェクトに反映させプレゼンテーションの効果的支援手法の研究がある[5]。また、市川らのVR空間をジェスチャを用いたNUI(Natural User Interface)により操作するシステムの研究がある[1]。松原らは距離画像センサを用いたジェスチャ認識により、画面に対する手の位置や近づくに応じてメニュー操作できるUI技術を開発した[4]。

3 3DVRアニメーション編集システムの提案

3.1 3DVRアニメーション編集システム

まずHMDを用いたモーションキャプチャ環境を構築する。次に特定のジェスチャやポーズを判断し、音楽を流したり表情を変えるなど、NUIによるアニメーション編集に役立つシステムを構築する。本研究ではコントローラのトリガーを引いている間に行った動作について判定し、ジェスチャ認識させたくない場合と分けて使用することが出来るようにする。

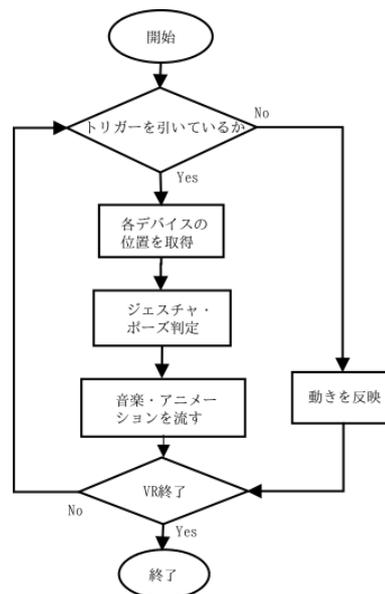


図1 制作するシステムのフローチャート

3.2 VR開発に使用されるゲームエンジンとPC用HMDの比較

まず一般的にVR開発に使われるゲームエンジンの比較を行う。表1に示した通り、Unityはシェアがトップということもあり、日本語の情報が多く、研究・開発を行い易い。よって本研究ではUnityを使用する。次に使用するHMDの比較をする。PC用として代表的なのはHTC VIVE Pro, Oculus Rift S, Windows MRである。本研究では全身モーションキャプチャを利用したジェスチャやポーズ認識を行うため、表2に示した通り、トラッキングが可能な関連機器があるHTC VIVE Proを使用する。

表1 ゲームエンジンのシェア

シェア	ゲームエンジン
1位 (50~60%)	Unity
2位	Unreal Engine
その他	Lumberyard, Godot, OROCHI

表2 PC用HMDの比較

HMD	性能
HTC VIVE Pro	360度トラッキング・関連機器を用いたモーショントラッキング
Oculus Rift S	360度トラッキング
Windows MR	トラッキング範囲が狭い

4 3DVR アニメーション編集システムの構築

4.1 構築環境

使用する機材は HTC VIVE Pro と VIVE Tracker3 個である。使用ソフトは Unity, SteamVR, AviUtl である。使用アセット (Unity 内での VR 開発に使用) は FinalIK, Steam VR Plugin, OBS Studio, OVRlipSync, AniLipSync-VRM, UniVRM である。

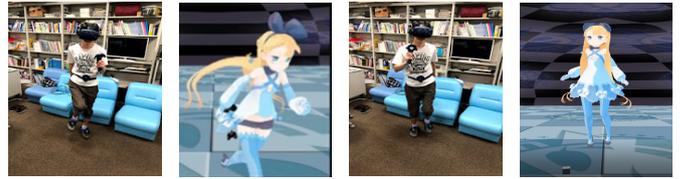
4.2 NUI の VR 空間への適応

本研究では特定のジェスチャとポーズをして狙った操作をするのが目的のため、各ジェスチャとポーズ毎に行いたい操作のプログラムを組む。「ダッシュ・ジャンプする」「驚き・怒り・喜び・悲しみなどの表情変化」「BGM を流す・止める」「カメラの切り替え」の4つについて、NUI を用いて VR 空間またはキャラクターを操作する。図 4.2 は「ダッシュ・ジャンプする」の操作を表している。

Listing 1 ダッシュ・ジャンプをさせるプログラムの一部

```
1  if (ground && ControllerInput.l == false){
2  playerPos = transform.position;
3  if (vRTest.v > 0.05f && (vRTest.x>0.05f ||
4      vRTest.y>0.05f)){
5      vrik.enabled = false;
6      transform.rotation =vRTest.head.transform.
7          rotation;
8      transform.position =new Vector3( vRTest.
9          head.transform.position.x,0, vRTest.
10         head.transform.position.z);
11     //走るアニメーションを再生
12     animator.SetBool("Running", true);
13 }else{
14     animator.SetBool("Running", false);
15     vrik.enabled = true;
16 }if (vRTest.w>0.05f && (vRTest.x > 0.05f ||
17     vRTest.y > 0.05f)){
18     transform.rotation = vRTest.head.transform
19         .rotation;
20     transform.position = new Vector3(vRTest.
21         head.transform.position.x, 0, vRTest.
22         head.transform.position.z);
23     vrik.enabled = false;
24     //thrustの分だけ上方に力がかかる
25     rb.AddForce(transform.up * thrust);
26     if (rb.velocity.magnitude > 0)
27     //速度が出ていたら前方と上方に力がかかる
28     rb.AddForce((transform.forward * thrust) +
29         (transform.up * thrust));
30 } }
```

Listing1 では、別のプログラムで取得した速度 v (左手)・ w (右手)・ x (左足)・ y (右足) によってダッシュ・ジャンプの判定をしている。ControllerInput.l と ground はそれぞれ表情変化の操作を行っているかどうか、床に触れているかどうかを判定する bool 型変数である。速度の条件が満たされた時は、4・15 行目でモーションキャプチャした情報をキャラクターに反映する VRIK の機能を停止する。また、VRIK を停止するとキャラクターが初期位置に戻ってしまうため、再生する前に 5・6、13・14 行目のように位置と回転を最新の情報に更新する。そしてアニメーションが停止された時は VRIK を再び適用する。



(1) ダッシュ VR 内 (2) ジャンプ VR 内

図 2 ジェスチャの操作例

5 アニメーション編集と考察

これまでは大きな動きをさせたい場合は実際に動いて撮るしかなく、高価な機材・設備でないと難しかったが、このシステムにより小さな動きで安全に撮ることが出来た。また、動画編集ソフトで簡単な編集をするだけで済み、表情やモーションを後から編集する必要が無くなった。現状の個人でのアニメーション制作・編集へのハードルをかなり下げることが出来たと言える。

6 結び

本研究では、VR 空間で HMD を用いたモーションキャプチャを行い、NUI によるアニメーション制作・編集に役立つシステムを構築した。かかったコストは 255,641 円程度に収まった。今後の課題としては、他のモデルでも同じようにアニメーションを制作・編集出来るようにすること、表情やアニメーションの多彩化などがある。

参考文献

- [1] 市川 ひまわり, 飯島 沙織, 新田 善久, “VR 空間におけるジェスチャを用いた Natural User Interface の研究,” 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI), 2 号, pp. 1-8, 2018.
- [2] “Breakthrough みんなの目元に VR [第 2 部:コンシューマー用途] 誰でも「Vtuber」時代へ 格安ツールやセンサーが後押し,” 日経エレクトロニクス, 08 号, pp. 28-33, 2018.
- [3] 古川 真行, 赤木 康宏, 河合 由起子, 川崎 洋, “Kinect を用いた簡易 3D アニメーション制作システムの提案,” インタラクション 2014 論文集, 情報処理学会, pp. 166-171, 2014.
- [4] 松原 孝志, ボンダン ステシアワン, 松本 和己, 徳永 竜也, 中島一州, “3 次元ジェスチャ操作によるテーブル型インタラクティブデジタルサイネージの開発,” 情報処理学会論文誌コンシューマ・デバイス&システム (CDS), 4 巻, 3 号, pp. 1-8, 2014.
- [5] 山田 裕之, 丹羽 佑輔, 白松 俊, 大園 忠親, 新谷 虎松, “ジェスチャーで操作可能なスライドオブジェクトに基づくプレゼンテーション支援システムについて,” 日本ソフトウェア科学会第 30 回 (2013 年度) 大会講演論文集, 2013.