進捗報告

1 今後の研究方針

人物の行動認識は, 与えられた動画中に映る人物が何の動作を行っているのか分類するタスクであり, 動画解析における主要な研究テーマの一つとして盛んに研究されている. そして, この応用研究として白黒映像に色情報を付与する自動着色や, 映像を滑らかにするフレーム補間, 動画に対する音声の自動生成などの生成タスクがある.

近年, アニメやゲームといった動画を扱うコンテンツ業界においても, データの複雑化・大規模化に伴って機械学習を用いた開発・創作支援が取り入れられている. 例えば, アニメ業界では中割作業を機械学習によって自動化したり [1], ゲーム業界ではキャラクタのセリフや音声から感情を推定して 3D モデルの表情に適応させる [2]などの事例がある.

しかし, 著作権などの観点から従来の動画解析を目的とした一般に公開されているデータセットは実空間を対象としたものがほとんどであるのが現状である.

そこで, 行動認識およびマルチメディアデータの自動生成を目的とした 3D モデルアクション動画データセットの構築に関する立案を行う.

2 Unity, ユニティちゃん

Unity（Unity3D）とは, Unity Technologies（日本法人はユニティ・テクノロジーズ・ジャパン株式会社）が開発・販売している, ゲームエンジンであり, アニメやゲームなど, 幅広い業界で用いられている.

ユニティちゃんとは, ユニティ・テクノロジーズ・ジャパン株式会社が提供する開発者のためのオリジナルキャラクターであり, キャラクターを自由に設定できるように利用規約に準じる形でアセット（素材）として無料配布している.

3 GameSynth

GameSynth [3] とは, tsugi 社によって開発された最新のプロシージャルサウンド合成技術を駆使してあらゆるジャンルのゲーム効果音を作成できる, サウンドデザインツールで, 効果音バリエーションの自動量産機能など, ゲーム開発・映像制作において役立つ機能も含まれている.

4 データセット制作フロー案

4.1 動作ラベルの選定

「歩いている, 走っている, 攻撃 (剣), 攻撃 (銃), 魔法 (火), ギターを弾いている, ピアノを弾いている, 階段を昇り降りしている, ...」といったアセット (素材) との兼ね合いで実現できそうなものを選定する予定であるが, まずは「歩いている」だけに絞って行うこととする. (最も簡潔な動作であるため.)

4.2 シーンの作成

各動作ラベルについて, 図 1 に示すようなシーンを Unity 上で作成する. 次に音声素材を GameSynth 等を用いて制作する. 動作ラベルに対応するアニメーションはユニティちゃんアセットに含まれているものや, Mixamo [4], Unity のアセットストアなどから集める. その他, 動作ラベルが「ピアノを弾いている」におけるピアノなど, 必要なアセットも Unity のアセットストアなどから集める. (この時, 各アセットのライセンスには注意しなければならない.)

4.3 シーンの録画

Unity 内の Record 機能を用いて t 秒間の動画を固定 FPS , 固定サンプリングレートで撮影する. 一回の撮影には必ず動作ラベルを表すアクションを起こさせるようにし, 以下のような情報を毎フレームごとに抽出可能である. ランダマイズするパラメータを指定することで, これを *interval* 秒おきに録画することでデータの大量作成が可能である.

- 距離, 方位角 (azimuth), 仰角 (elevation) (図 2 参照)
- ユニティちゃんのステートが idol か動作ラベルか
- 音イベント検知

今回は, 初めとして以下の条件下でデータセットを作成する.

- 60fps, 5 秒間の動画を繰り返し撮影する.
- 動作ラベルは idol (何もしていない状態) または walking.
- ユニティちゃんは目的地に向かって「歩く」, 「止まって」, また次の目的地に向かって「歩く」卯を繰り返す.
- 5 秒間の動画の中で, walking の割合が 2 割以上 8 割以下のデータのみを扱う.
- ユニティちゃんの歩行スピードは一定の範囲内でランダムに選ばれる.
- カメラの位置は固定とする.

[1] https://cacani.sg/
[2] https://cedec.cesa.or.jp/2020/session/detail/s5e58c8811bc98.html
[3] http://tsugi-studio.com/
[4] https://www.mixamo.com/

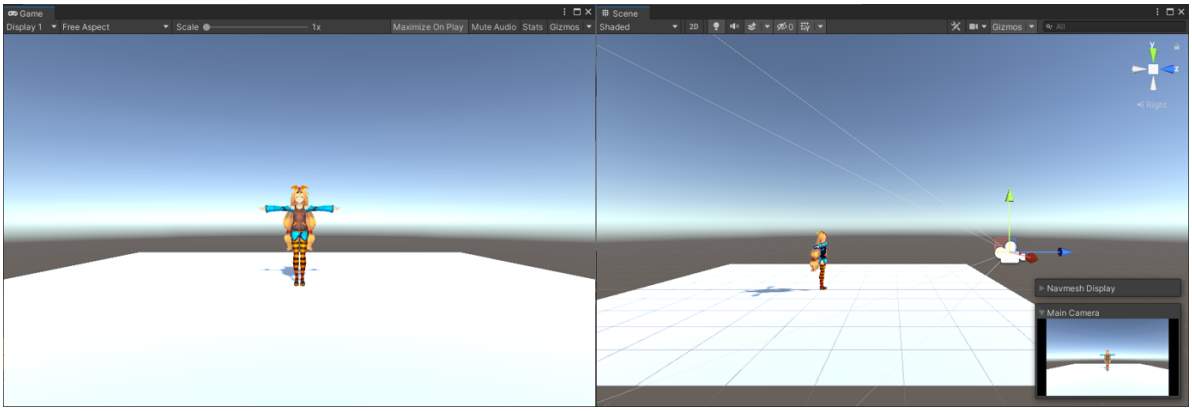

図 1: Unity シーン 概要

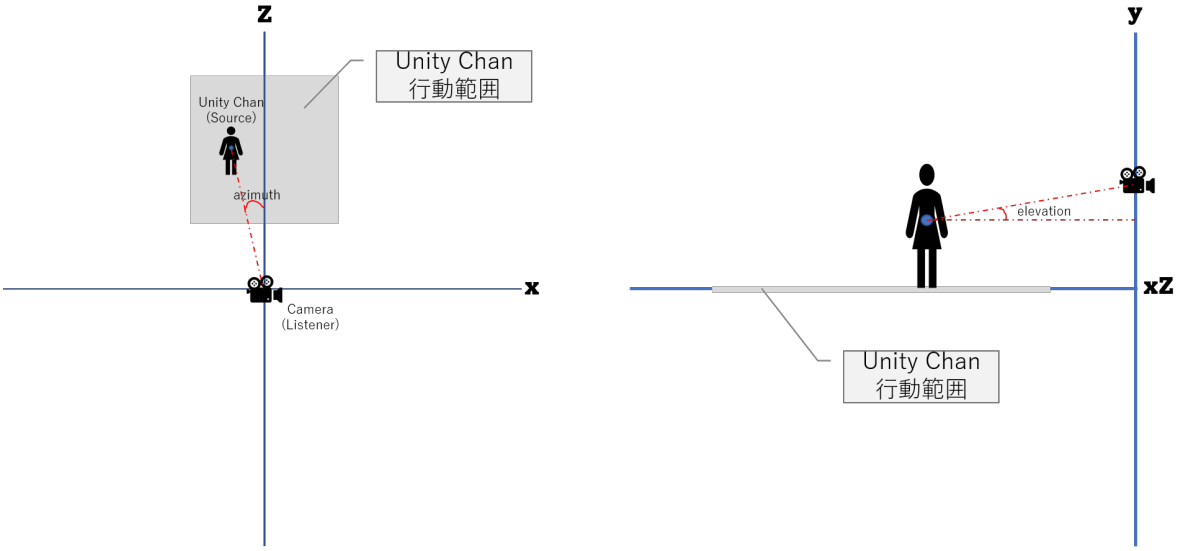

図 2: Unity シーン 座標系

図 3 に出力されるメタ情報の例を示す.

frame	distance	azimuth	elevation	state	audio_event
0	1.71729	-45	14	idle	
1	2.15534	-43	14.3	walk_default	walk_default
2	0.01812	-42.5	14.2	walk_default	
3	0.54333	-39	13.6	walk_default	walk_default
4	1.3783	-23	13.2	walk_default	
5	2.11827	-19	12.6	walk_default	walk_default
6	0.58305	-10	11.6	walk_default	
7	0.25785	-2	11.5	walk_default	walk_default
8	1.66969	3	11.3	walk_default	
9	2.59832	14	11.4	walk_default	walk_default
10	0.5851	16	11.2	idle	
11	0.50678	18	11	idle	
12	2.00431	29	10.4	idle	
13	1.57715	34	9.8	walk_default	walk_default

図 3: 出力メタ情報例

4.4 データセットの特徴

- ユニティちゃんを音源, カメラをリスナーとする 3D 音響空間であることから, 正確な相対位置をフレームごとに取得できることによって音源の位置推定タスクに有用であってほしい.
- 左から鳴った音は左側から聞こえ, 遠くで鳴った音ほど小さく聞こえる.
- 3D 音響空間のポストエフェクト (リバーブやエコー) も任意に指定可能.
- 例えば, 床のアセットに応じて音や反響パラメータを変化させることもできる.
- 実動画と比べてノイズが入らない.
- 同じ fps, フレーム数, サンプリングレートで出力されるので前処理が楽. エラーデータの削減.

5　現状と次にやること

現状

- 大量生成部分の実装はまだ未完成ではあるが, 何とかなりそう...?

次にやること

- 一旦実装を完成させる
- 作成したデータの量や例を提示する
- LSTM を用いて, 動作ラベルの時系列推定 (入力は連番画像, または音声, またはその両方)
- 距離, 方位角, 仰角を認識できるような映像のエンコーダを学習 (音源位置推定タスク)
- 音源位置を考慮した音声の生成タスクに取り組む