

三次元モデル作成と顔の特徴点座標抽出

B4 田川幸汰

1 概要

進捗報告として三次元モデルの作成と顔の特徴点座標抽出を発表する。
三次元モデルの作成、顔の特徴点抽出を達成するために用いた技術と、その結果を以下の章で説明する。

2 三次元モデルの作成

三次元モデルを Blender のアドオンである FaceBuilder を用いて作成した。

2.1 FaceBuilder

FaceBuilder は、人間の顔や頭部のモデリングを簡単に行うためのツールである。FaceBuilder を使用すると、写真やビデオから自動的に顔のモデルを作成することができる。また顔の形状や解像度の詳細な調整、テクスチャの適用など、モデルのカスタマイズもサポートしている。他の Blender の機能とも連携し、リグやアニメーションの作成など、さまざまな作業を補助することができる。FaceBuilder は以下のように使う。

1. FaceBuilder のタブから [Create new head] を選択
2. 下絵となる写真を一枚以上 [Add Images] より追加
3. open した画像を選択して [Align face] を選択
4. 顔のラインに合わせてメッシュを微調整し [Create texture] を選択
5. 様々な 3 次元、2 次元形式でモデルやテクスチャをエクスポートする

2.2 実行結果

2.2.1 入力画像

入力として、今回は図 1 の 1 枚の画像を用いた。



図 1: 入力画像

2.2.2 出力画像

出力された 3 次元モデルは図 2 である。正面はきれいにテクスチャが貼られているが、側面、裏面は画像に写っていないので貼られていない。複数枚の写真を用いて側面を描画することも可能であるが、顔の正面に写るマスク部分の描画であれば、顔画像一枚でも問題ないと思う。



図 2: 出力画像

2.2.3 頂点座標の出力

blender のテキストエディタで python コードを記述し、作成した 3 次元モデルの頂点座標を出力する。図 3 は blender の 3d ビューを表した図である。blender の 3d ビューはシーン、コレクション、オブジェクトといったように階層化されている特徴がある。python の bpy ライブラリを用いることで、3 次元モデルのメッシュ情報にアクセスできる。メッシュ情報には大量の頂点座標がベクトル形式で記述されている。

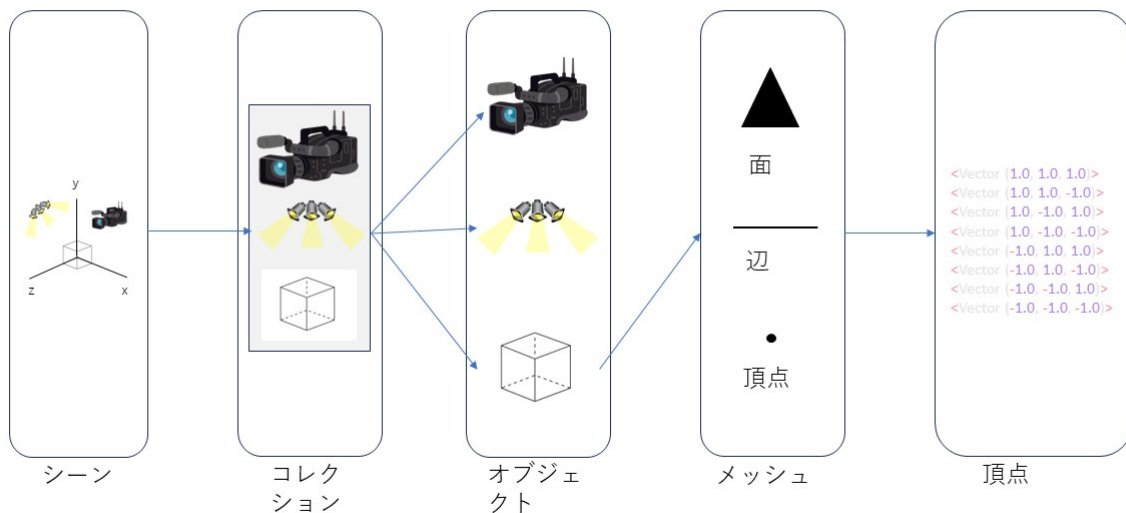


図 3: blender の 3d ビューのイメージ

図 4 は実際に出力された頂点座標を、python の matplotlib ライブラリを用いて 3 次元座標空間に出力した図である。なお、頂点座標は X,Y,Z 軸ともに正規化は行われていない。

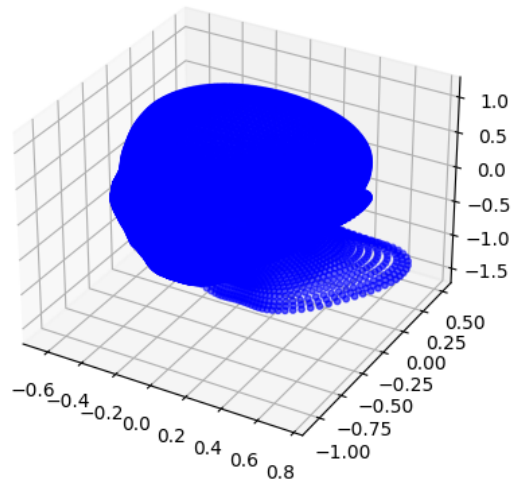


図 4: 頂点座標

3 顔の特徴点抽出

Google Mediapipe の FaceMesh の機能を用いて、顔の画像からフェイスマッシュを作成する。

3.1 FaceMesh

Google が開発したリアルタイムの顔のマーキングシステムである。FaceMesh は、ビデオや Web カメラの入力から、顔の輪郭、目、鼻、口などのさまざまな顔の特徴点を検出することができる。また、深層学習モデルに基づいており、ニューラルネットワークを使用して顔の特徴点を予測する。入力画像またはビデオフレームを FaceMesh に渡すと、顔の特徴点の 3D 座標と 2D 画像上の位置を返す。これにより、顔の構造や表情の解析、AR(拡張現実) アプリケーションの開発などに応用される。

3.2 実行結果

図 5 に FaceMesh を実行した画像を示す。このように顔上の特徴点を得ることができている。また、座標点を取得す

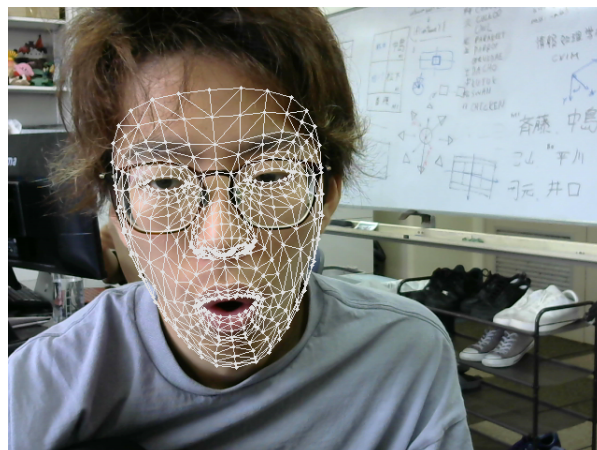


図 5: facemesh

することもできる。なお X, Y, Z 軸は正規化されていることに気をつけなければならない。

3.3 顔の特徴点を基に 3 次元モデルを張り付け

Python の OpenGL ライブラリを用いて、Facemesh で得られた顔の特徴点から三次元モデルを張り付けた。

3.3.1 OpenGL

OpenGL(Open Graphics Library) は、2D および 3D グラフィックスを描画するためのクロスプラットフォームなグラフィックス API で、OpenGL は、コンピュータ上でリアルタイムのグラフィックス処理を行うための一連の関数と命令のセットである。これらの関数を使用して、点、線、三角形などのプリミティブ（基本図形）を描画し、テクスチャを貼り付け、光源やマテリアルの効果を適用することができる。

3.3.2 グラフィックス処理

OpenGL で用いた様々なグラフィックス処理と、それに用いた関数を下に記述する。

射影変換行列... カメラ座標系に計算された行列を、OpenGL のウィンドウが映し出すスクリーン座標系に変換するための行列。平行投影変換と、透視投影変換に分けられる。透視投影変換を行うことで、カメラで写された範囲をどの程度の範囲で、どこまで見ることができるかという問題を解決する。glMatrixMode(GL_PROJECTION) で射影行列を選択し、glFrustum() で透視変換行列を作成。

モデルビュー行列... 物体が存在する世界座標系を、その物体を見ている視点であるカメラ座標系に変換するための行列。視界変換と、モデリング変換からなるが、これは同一の処理を行うことを指すため、モデルビューとして一括される。glMatrixMode(GL_MODELVIEW) でモデルビュー行列を選択し、glLoadMatrixf() で作成。

光の設定... OpenGL では光を演出するために光源と反射を処理することができる。glLightfv() でパラメータを設定。glEnable(GL_LIGHTING) で光源を有効にする。

カメラ位置の設定... オブジェクトの二次元座標と三次元座標を求めることで、カメラの姿勢、姿勢を特定することができる。これを PnP 問題という。solvepnp() は PnP 問題を解く関数であり、オブジェクトの二次元座標と三次元座標を引数に取り、カメラの回転ベクトル、並進ベクトルを返す。

3.3.3 実行結果

図 6 に実行した画像を示す。このように顔上の特徴点と 3 次元オブジェクトを表示できている。



図 6: facemesh と 3 次元オブジェクト



4 研究計画

現時点で進行中の研究や、次回の発表時までに勧めたい研究計画を以下にまとめる。

進行中

- マスクの有無による顔上の特徴点の検出の違い
- mgo 形式の 3 次元顔モデルを FaceMesh 上の特徴点に合わせて貼り付け (モデルと facemesh の座標合わせ)

次週以降

- 顔上の特徴点を用いた類似研究の論文の査読
- 3 次元顔モデルの貼り付けの修正
- facemesh の Z 軸がどのように正規化されているか調べる

参考文献

- [1] OpenGL 入門, <http://wisdom.sakura.ne.jp/system/opengl/index.html>, 閲覧日 2023/6/16