

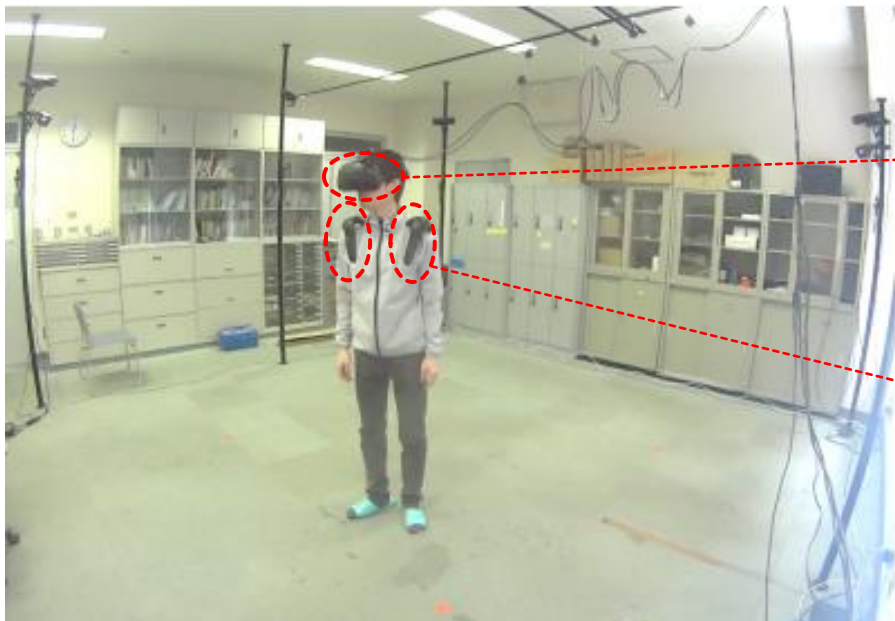
引き継ぎ資料

目次

- ・ 眼球・頭部協調運動収集システム(村上さん)⇒スライド3-5
 - ・ Demo⇒スライド6-15
 - ・ VR間違い探しの答え⇒スライド16-18
 - ・ Unity Back Up⇒スライド19-24
 - ・ 眼球・頭部協調運動モデルにおける位置移動の有無の影響についての調査⇒スライド25-26
 - ・ 眼球・頭部協調運動モデルの作成⇒スライド27-36
 - ・ 視線推定ライブラリの使い方⇒スライド37-48
 - ・ 視線推定ライブラリの利用例⇒スライド49-52
- ※視線推定ライブラリ(ライブラリ化前)の使い方⇒スライド53-60

眼球・頭部協調運動データ 収集システム(村上さん)

眼球・頭部協調運動データ収集システム^[1]



Eye tracker
(Pupil Labs)



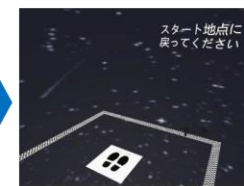
Controller
(HTC VIVE)

3種類

シーンバリエーション



間違い探しタスクを実施



1分間でシーンを
覚える

間違いを探す

眼球・頭部協調運動データ収集システム^[1]

ハードウェア



HTC VIVE
HMD & Controllers



Pupil Labs
Eye-trackers

様々な状況下における人の眼球・頭部運動データを
効率的に収集するVRシステム

VRゴーグルに映し出されるシーン



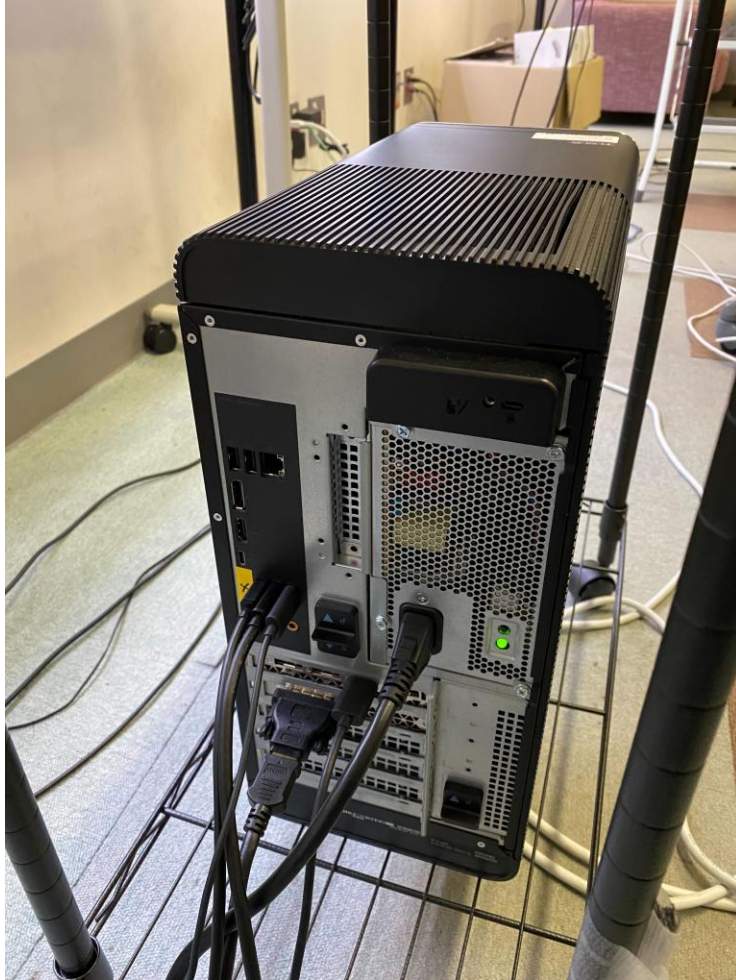
図書館

実環境



Demo

配線確認



PCの配線



VRゴーグルの配線

手順①

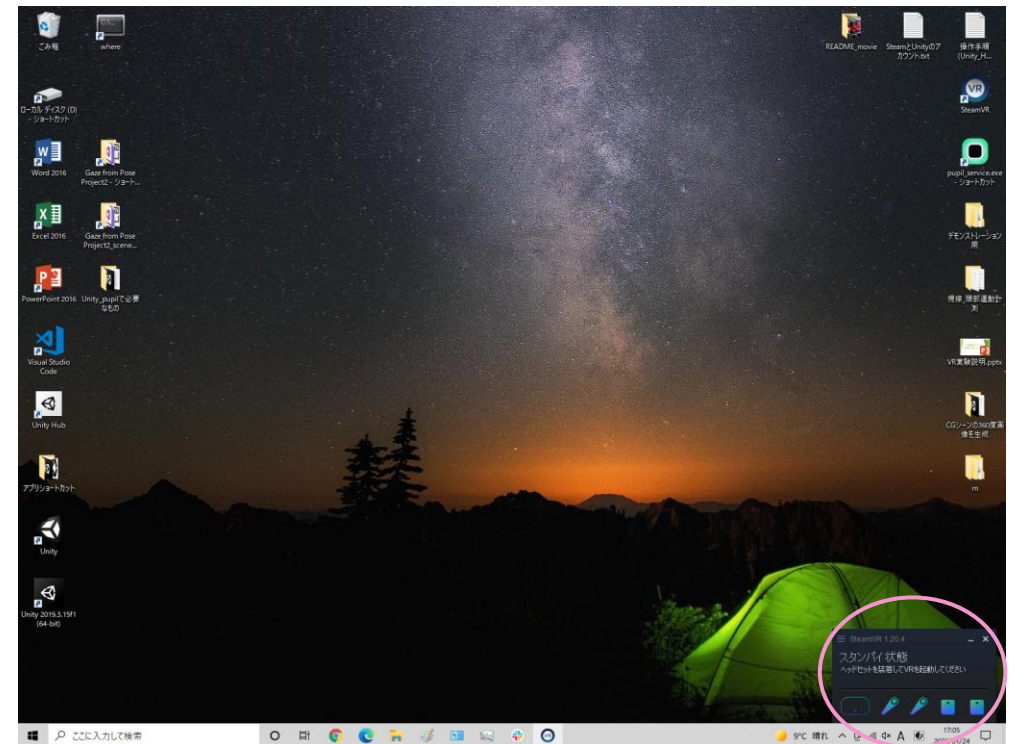
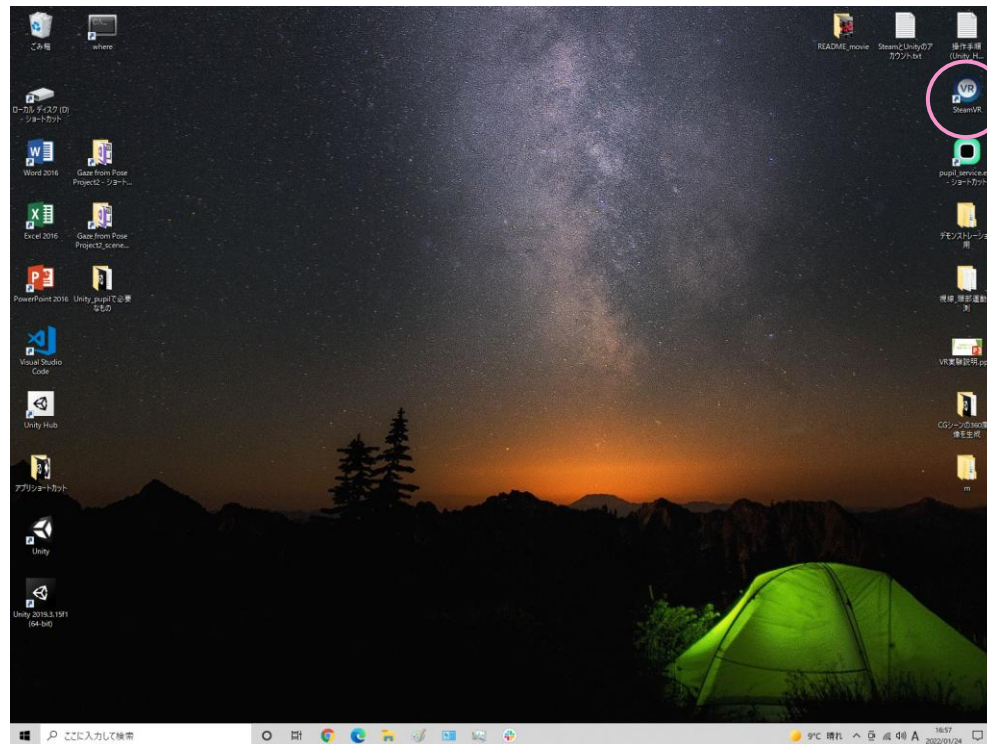
- ・ VRゴーグルをベースステーションが検出できる位置に置く。



ベースステーション×2
(ポール上部にある)

手順②

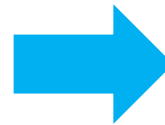
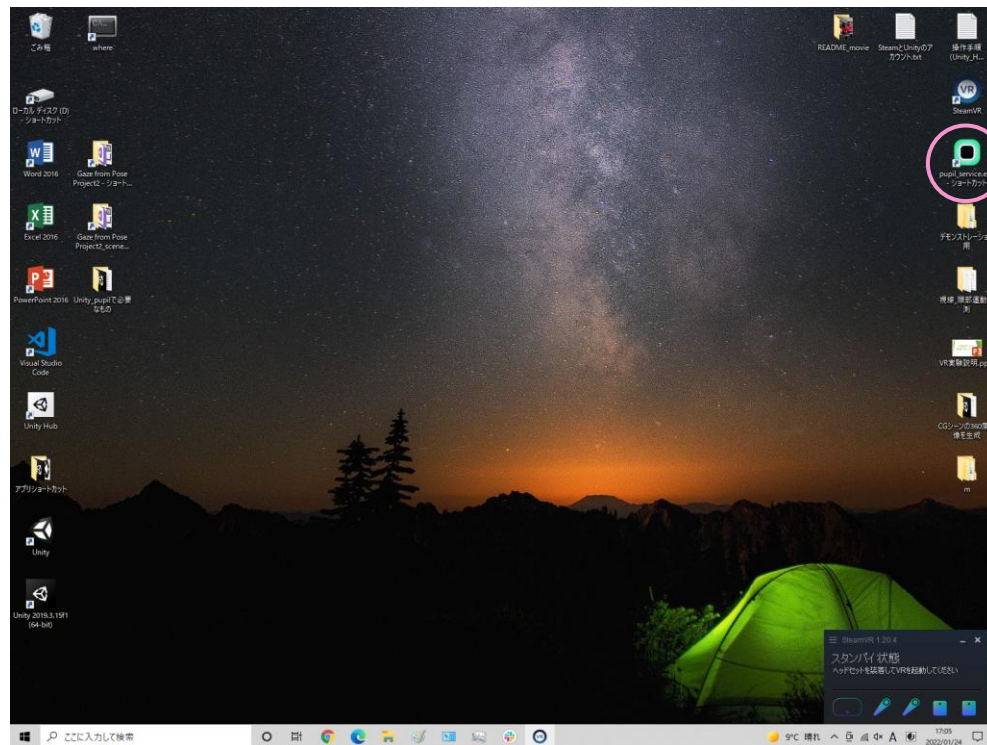
- Steam VR(画面右端)をダブルクリック.



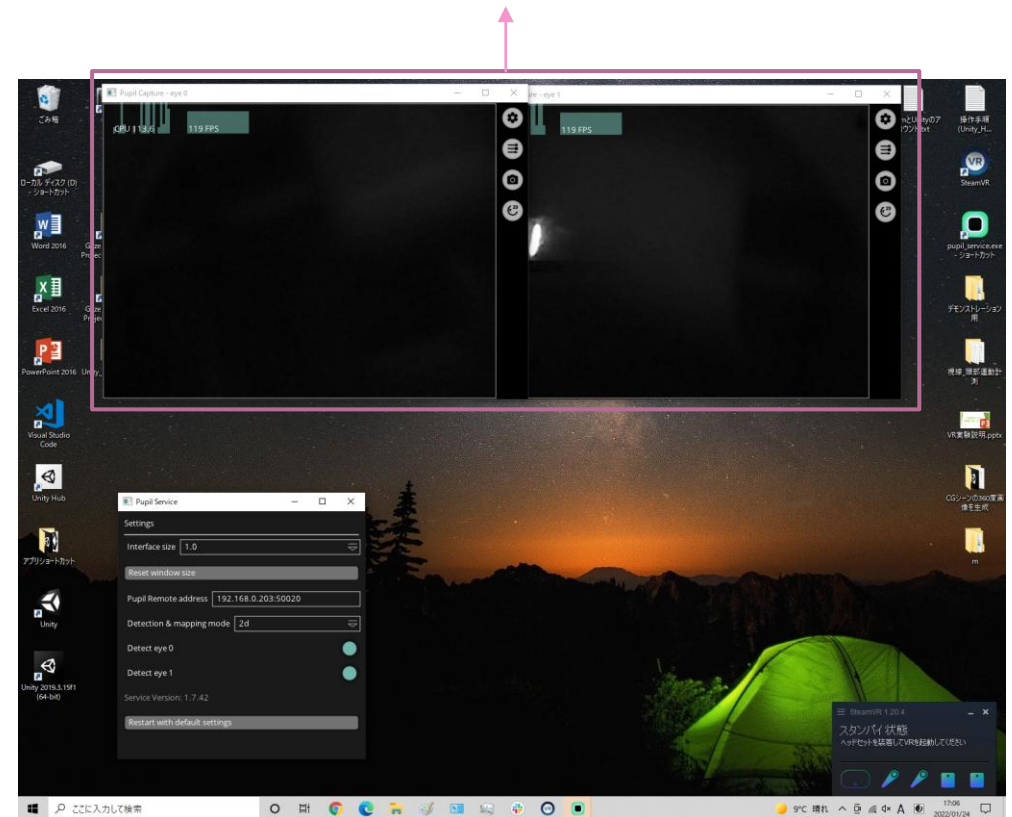
checkpoint

手順③

- pupil_service.exe(画面右端)をダブルクリック.

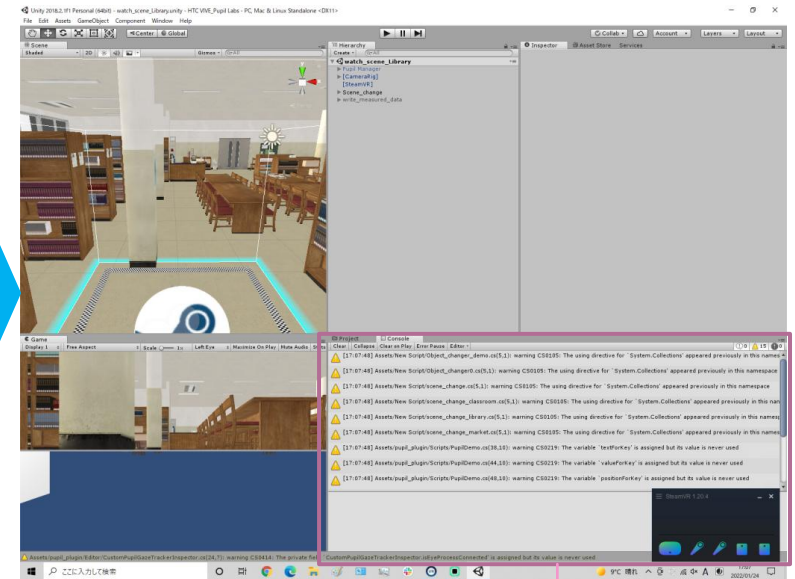
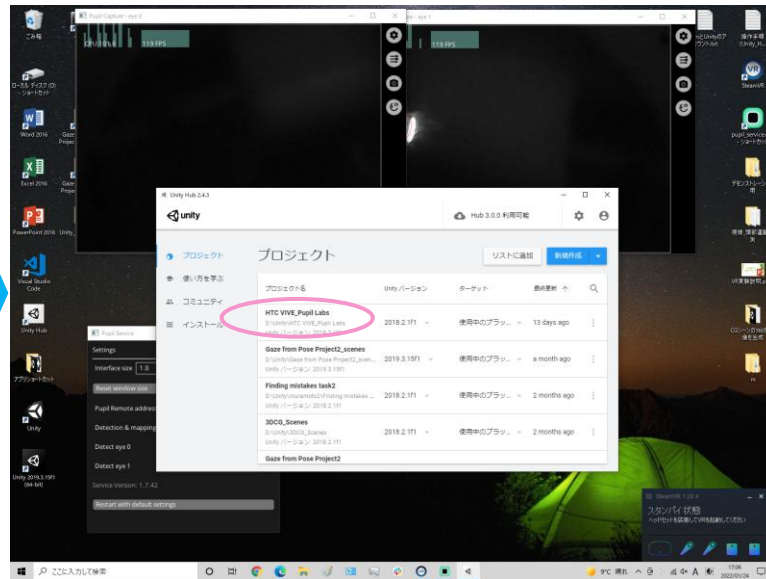
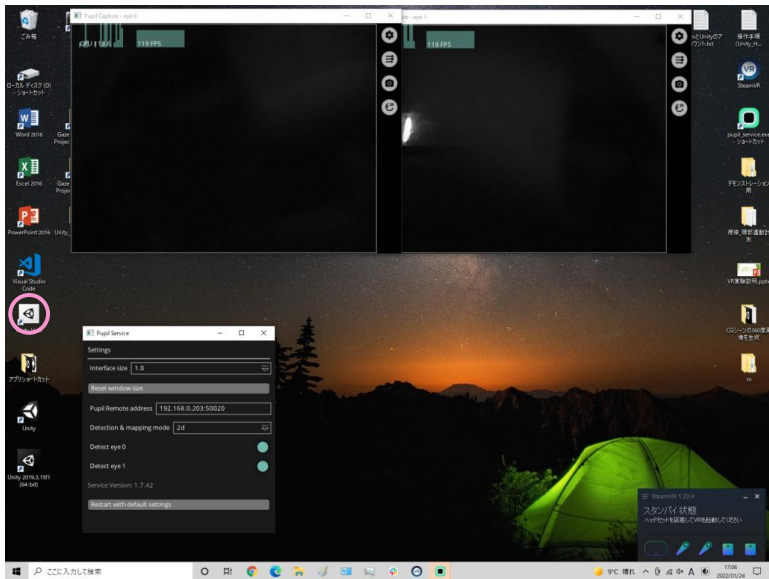


アイトラッカーのカメラ映像
(VRゴーグルを被ると眼球が映る)



手順④

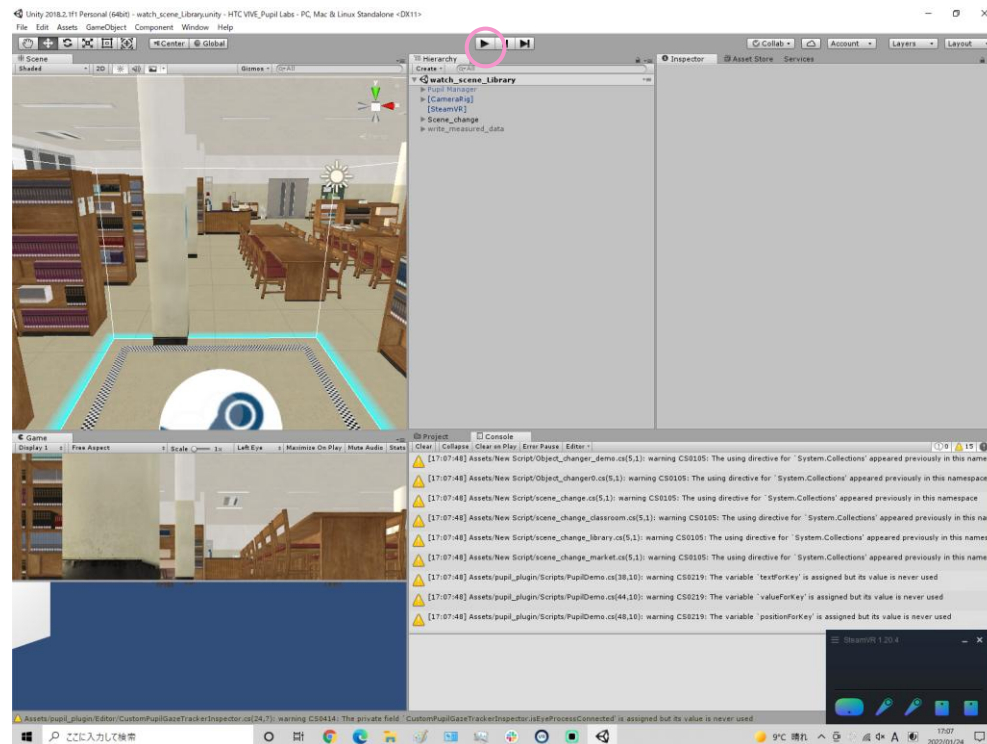
- Unity Hub(画面左端)を開いて, HTC VIVE_Pupil Labsを選択.



Consoleウィンドウにwarning(警告)が出ているが, 実行に影響は無し

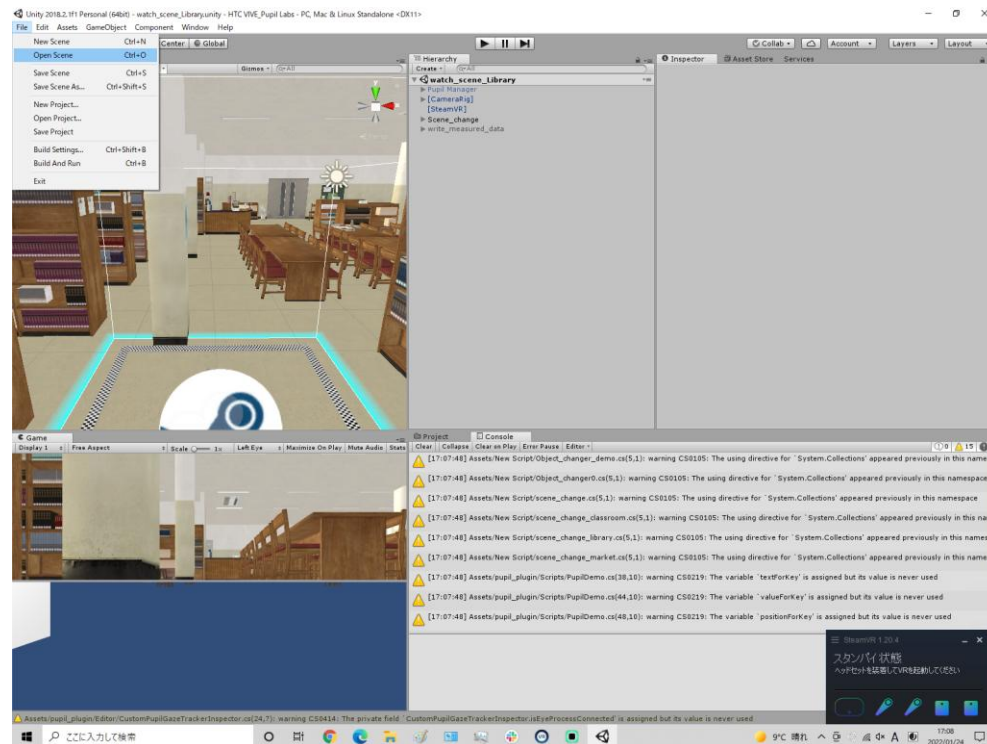
手順⑤

- ・▶(再生ボタンをクリック). (終了時には、もう一度クリックすること!)



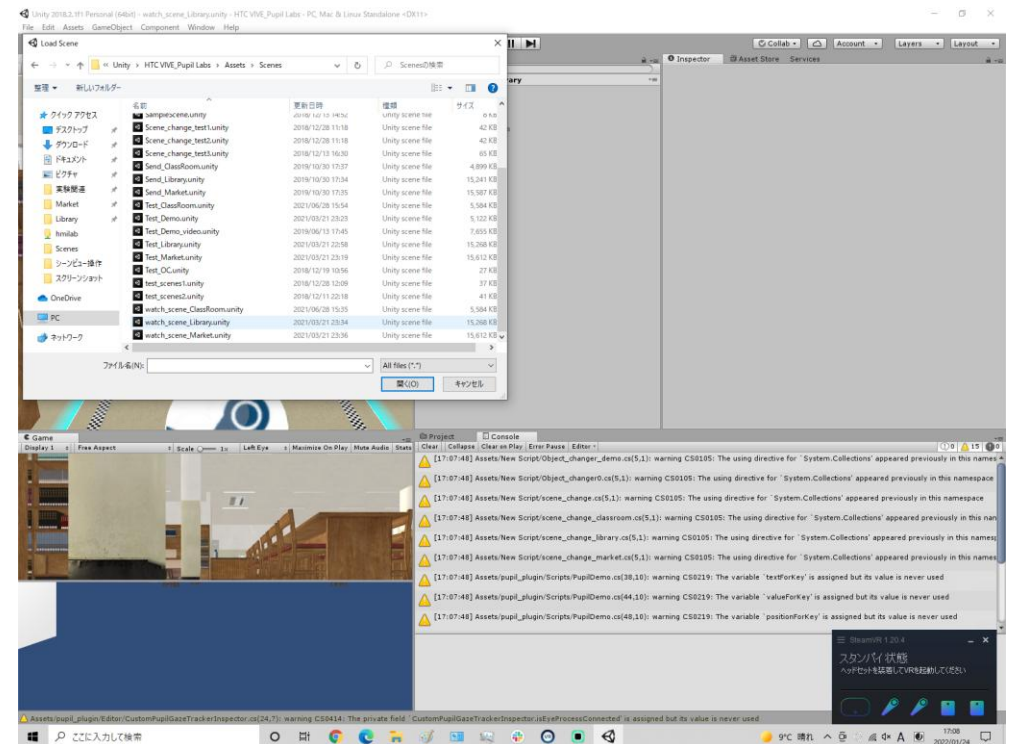
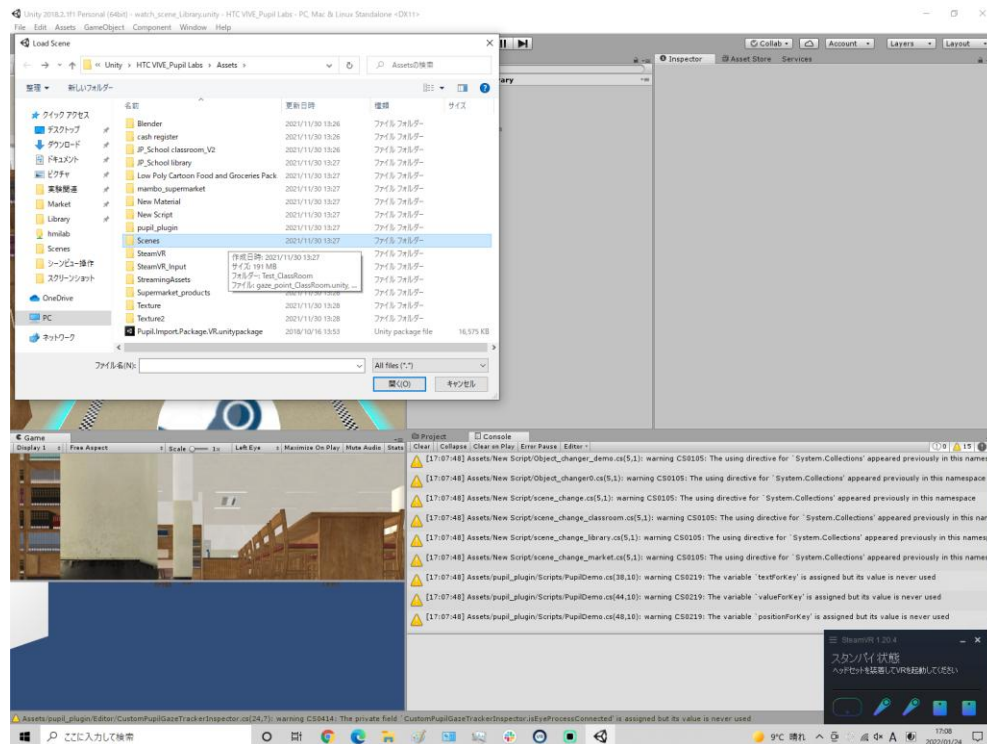
手順⑥

- ・シーンを変える時は、左上のFile→Open Sceneから選択.

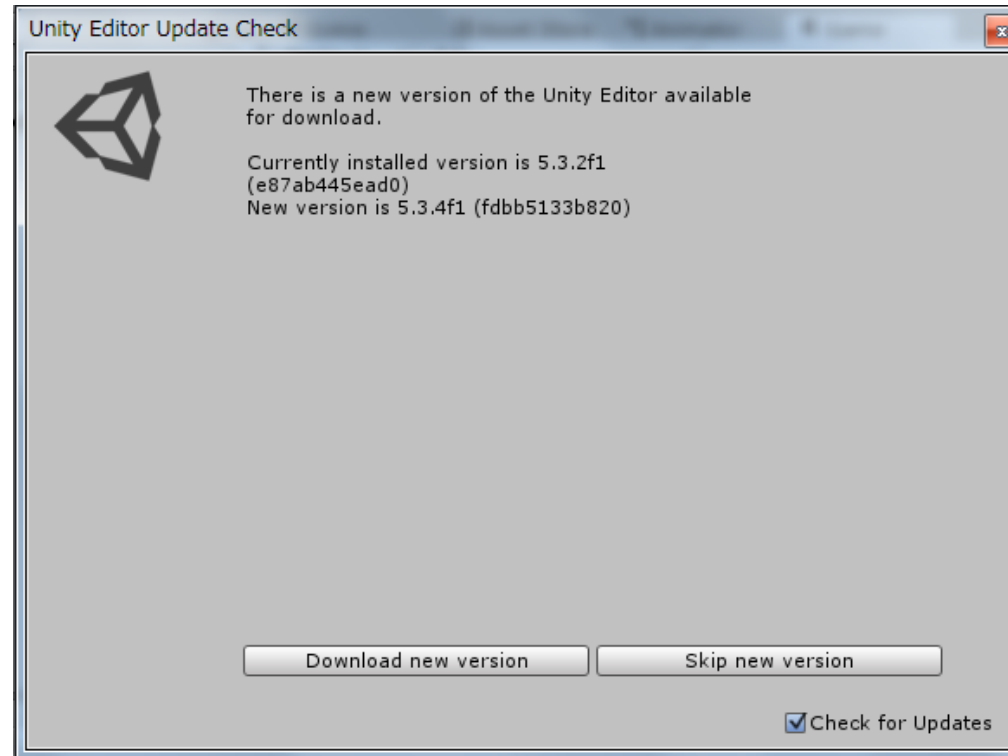


手順⑦

- watch_scene_Library.unity watch_scene_Market.unity watch_scene_ClassRoom.unityから選択.



Unityの注意点



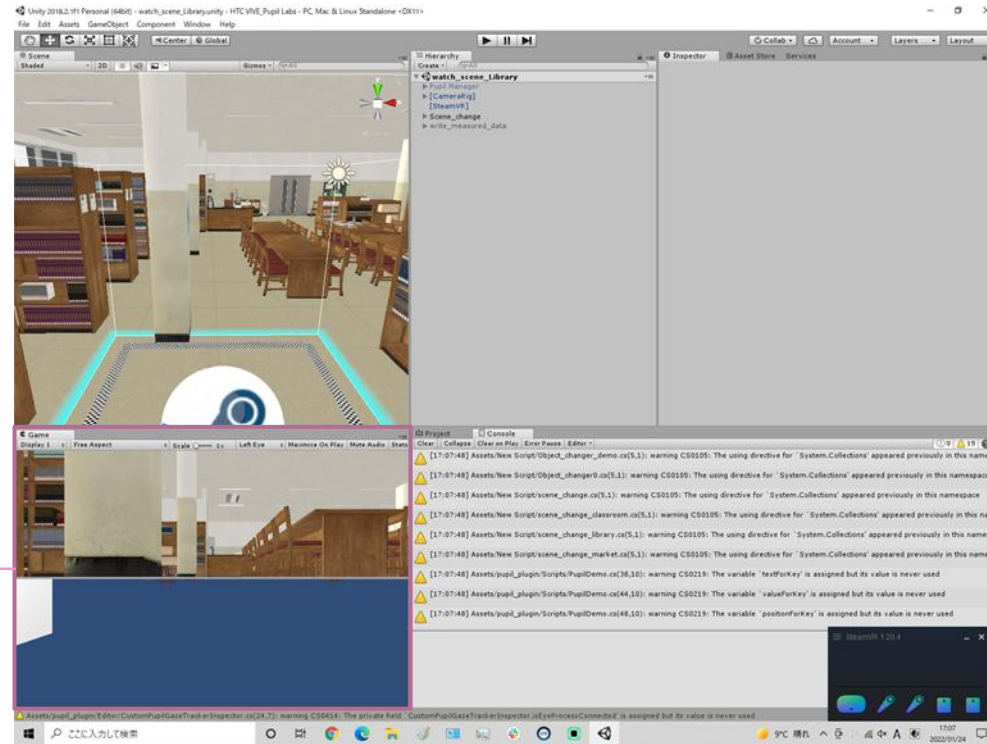
Unityを開いた際に、このようなウィンドウが表示される場合がある。
これは、Unityのバージョンの変更を促すもので、**skip new version**を選択する!!!

VR間違い探しの答え

VR間違い探しの答え

- L-2 : リンゴの貼り紙の位置が変わった
- L-3 : 左手前の本棚から本が消えた(見つけにくい)
- L-4 : 右後ろの壁紙が変わった
- L-5 : 右奥の椅子が移動した
- L-6 : 左の本棚の本が全て黄色になる(気付きにくい)
- M-2 : パンが1個無くなる
- M-3 : Winter Sale→Summer Sale
- M-4 : 買い物かごが減る
- M-5 : タマネギとネギが入れ替わる
- M-6 : 右下のクリーナーボトル(黄)が2列になる
- C-2 : 教卓が消える
- C-3 : 右手前の辞書が変わる(赤と青)
- C-4 : 黒板上の定規が消える
- C-5 : 左の本が増える(青とピンク)
- C-6 : 椅子が消える(そうじ当番表近く)

Gameウィンドウ



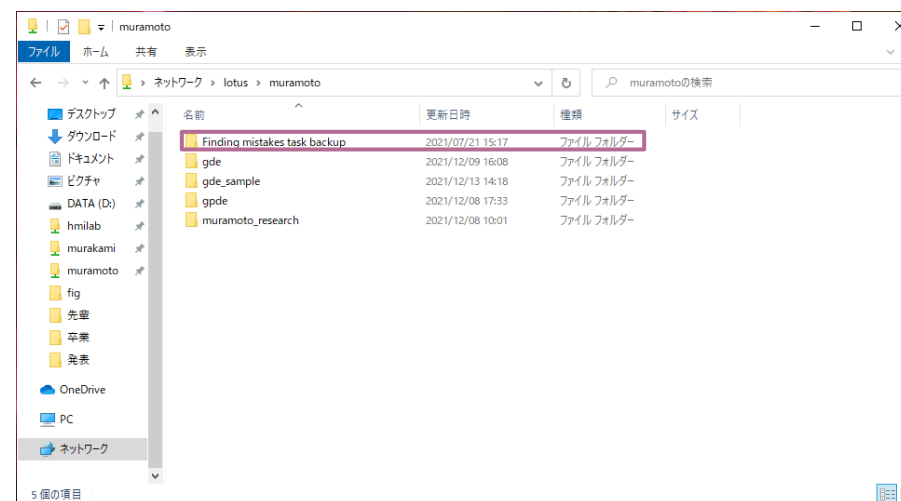
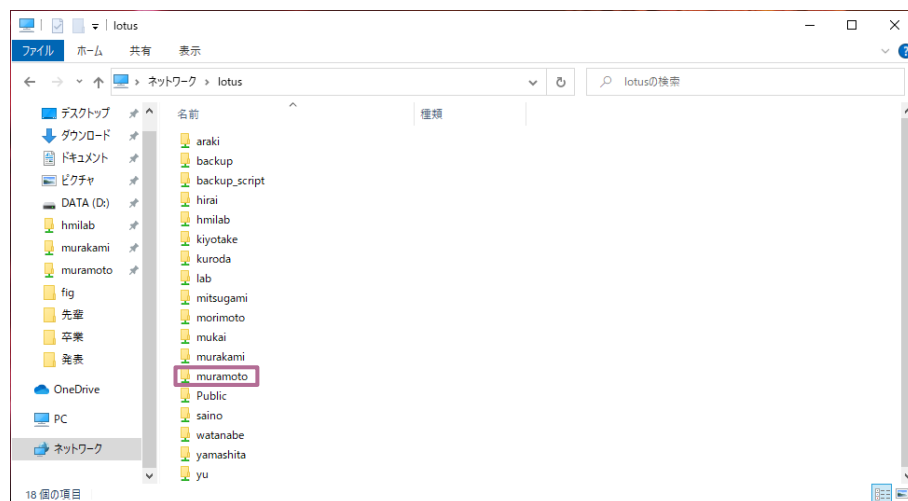
Gameウィンドウをクリックした状態で

1キー：基準シーン
2キー：1スライド前の○-2に相当
3キー：1スライド前の○-3に相当
4キー：1スライド前の○-4に相当
5キー：1スライド前の○-5に相当
6キー：1スライド前の○-6に相当
スペースキー：初期位置

Unity Back Up

Unity Back Up

- lotusのmuramotoフォルダ内のFinding mistakes task backupフォルダは、Unityのバックアップフォルダである。(もしもの場合、使うこと!)



手順①

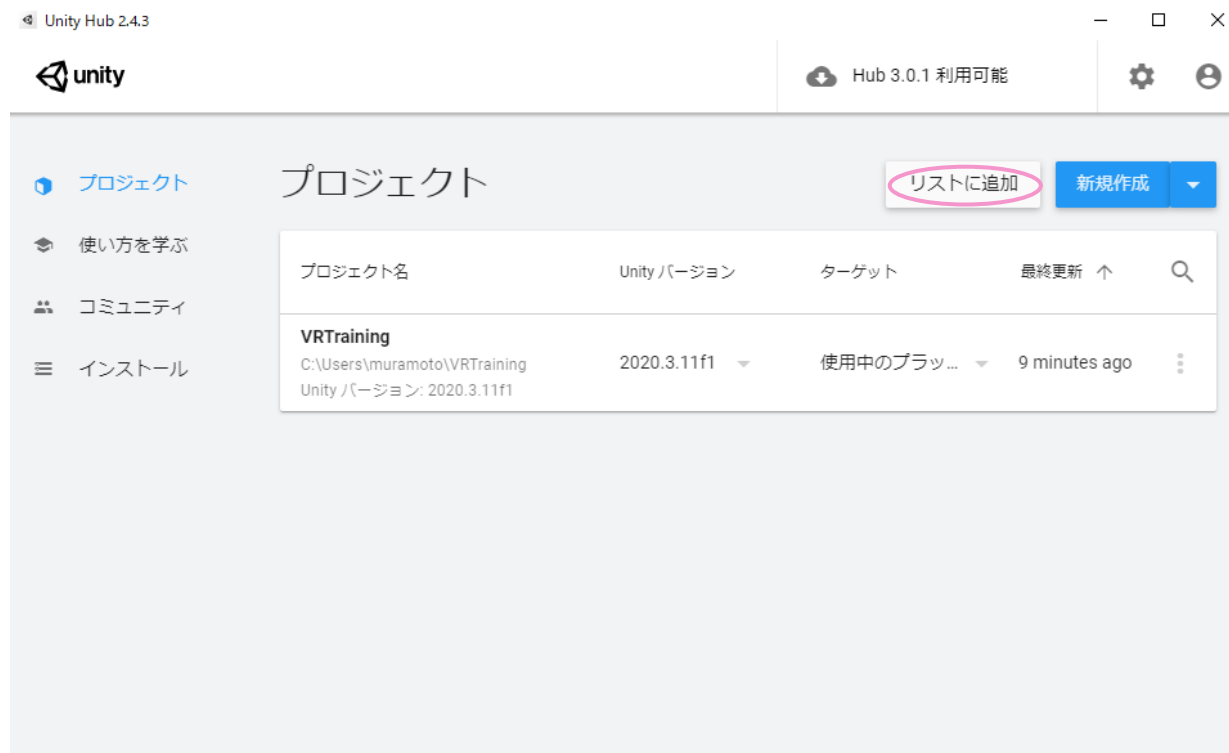
- Finding mistakes task backupフォルダをD:¥Unity内に(実験室PC)コピーする.



Finding mistakes task backupフォルダ

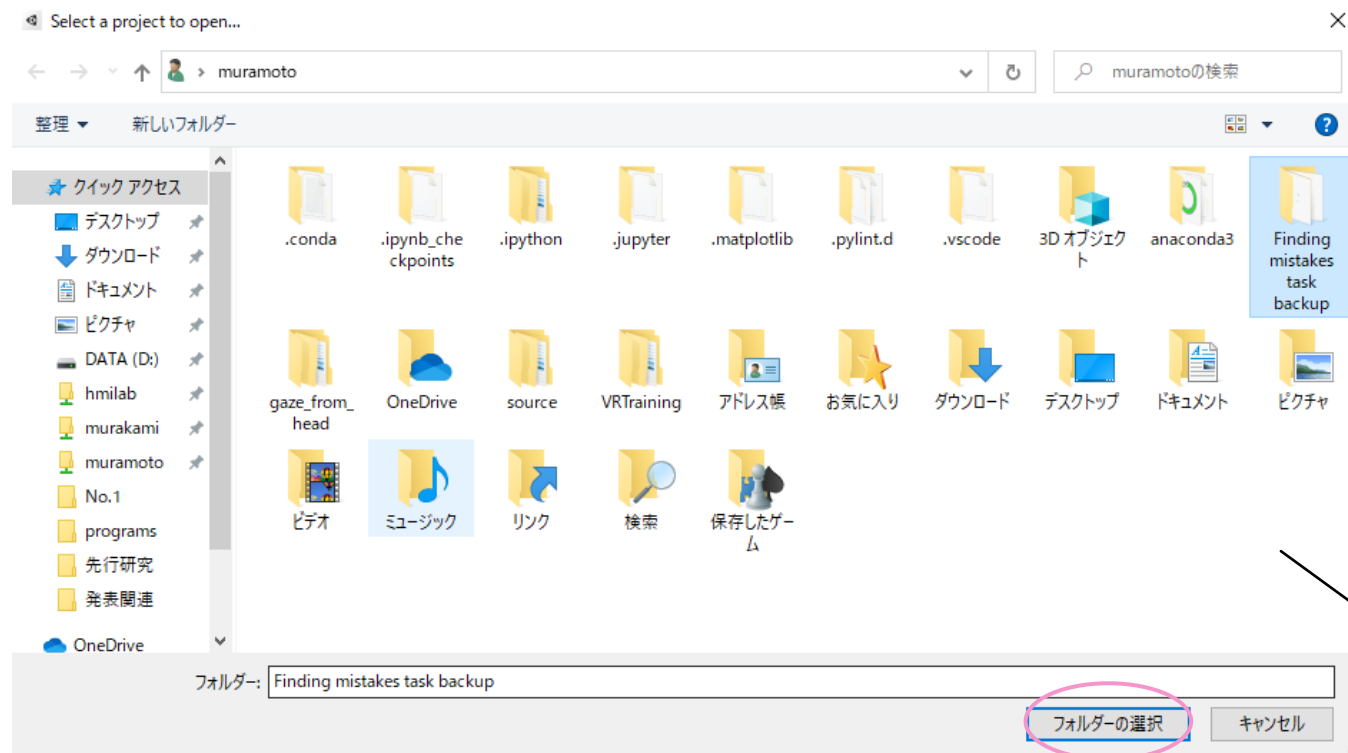
手順②

- Unity Hubを開き，リストに追加をクリック。



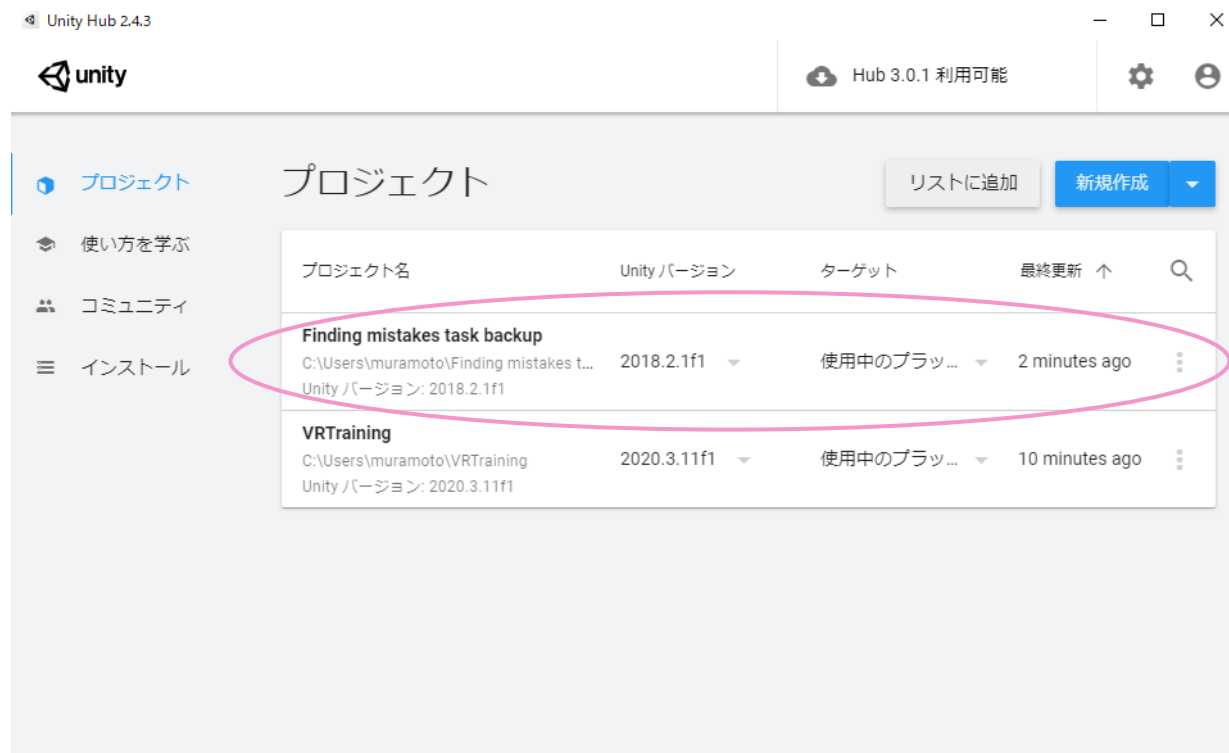
手順③

- Finding mistakes task backupを選択する。パス=D:¥Unity¥Finding mistakes backup



手順④

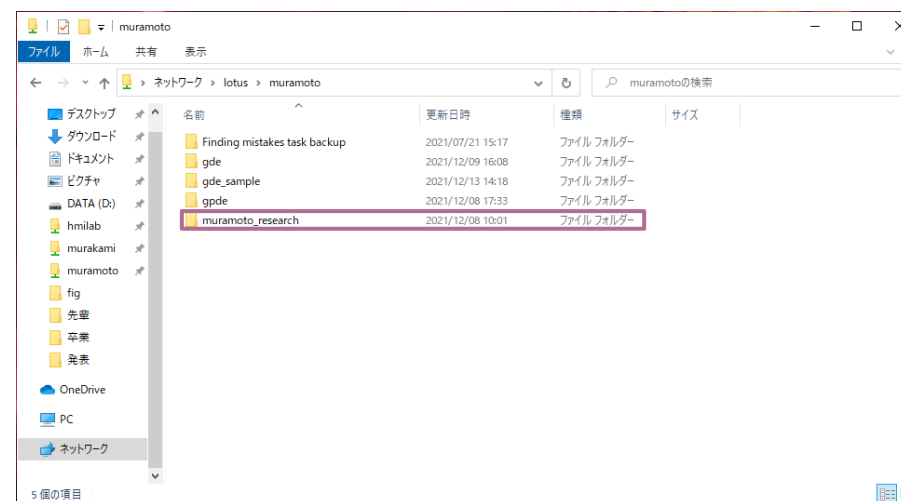
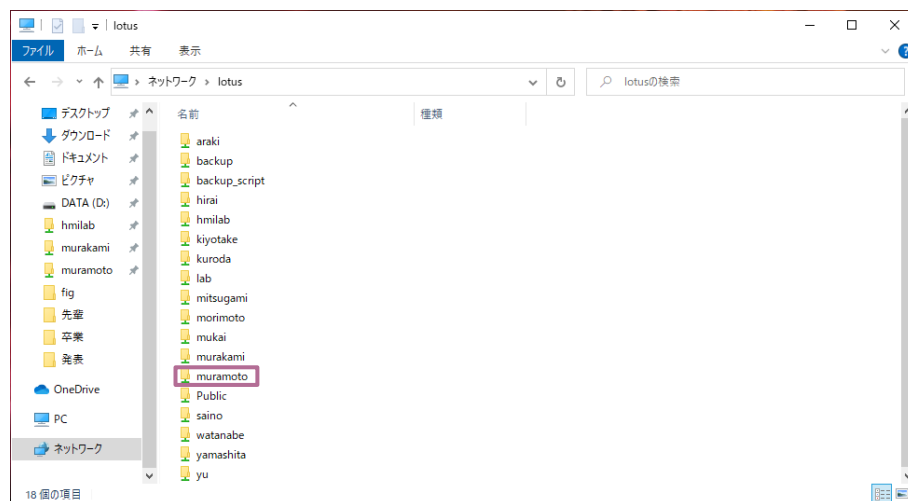
- Unityバージョンが2018.2.1f1になっていることを確認し, Finding mistakes task backupをクリック.



眼球・頭部協調運動モデルにおける 位置移動の有無の影響についての調査

互換性調査

- lotusのmuramotoフォルダ内のmuramoto_researchフォルダは、眼球・頭部協調運動モデルにおける位置移動の有無の影響についての調査に関するフォルダである。

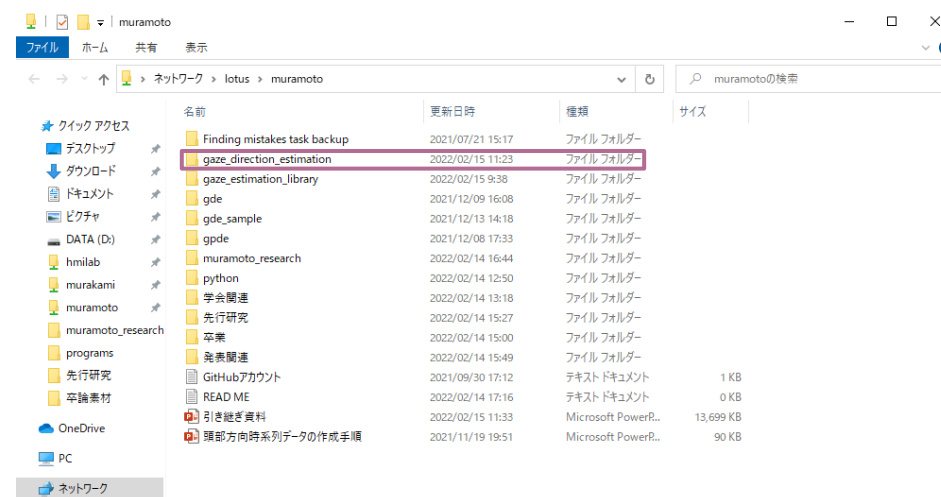
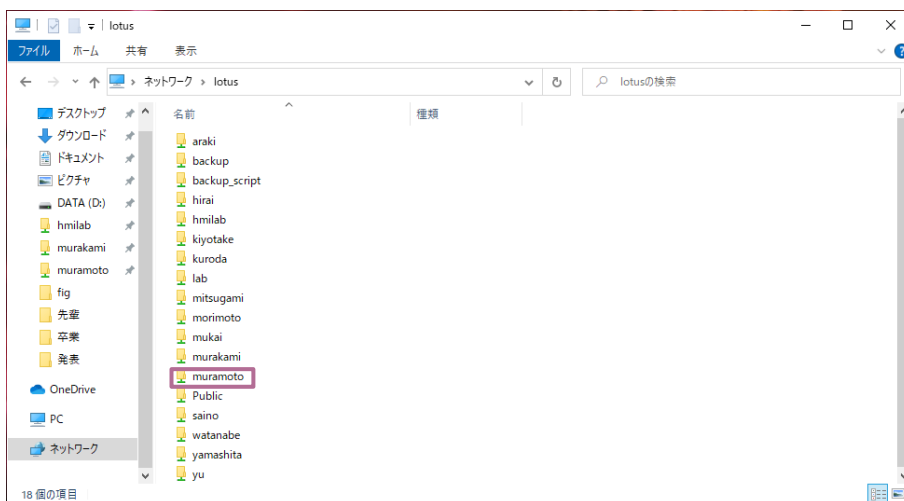


※詳細については、muramoto_researchフォルダ内のtrain_○ test_○フォルダ/Read meを参照してください
m=村上データセット(移動有りデータセット) s=Saliency in VR データセット(移動無しデータセット)

眼球・頭部協調運動モデルの作成

gaze_direction_estimationフォルダ

- lotusのmuramotoフォルダ内のgaze_direction_estimationフォルダは、眼球・頭部協調運動モデル(移動有りモデル・移動無しモデル)の作成に使用したフォルダである。



眼球・頭部協調運動モデルの作成(フロー図1)



例:Library_1.csv

time	hpx	hpy	hpz	...
0.00E+00	2.76E-01	1.62E+00	3.18E-01	
3.33E-02	2.74E-01	1.62E+00	3.27E-01	

⋮

ステップ1



VR_mistake_rawdata_to_degree.py



例:Library_1_deg.csv

time	head_h	head_v	gaze_h	gaze_v
0.00E+00	-4.17E+00	-6.59E-01	3.28E+00	1.60E+01
3.33E-02	-4.19E+00	-1.24E+00	3.58E+00	1.64E+01

⋮

ステップ2



murakami_1.py



例:Library_1_head_gaze.csv

head_h	head_v	gaze_h	gaze_v
2.74E+01	5.70E+00	9.63E+00	5.90E+00
3.09E+01	5.23E+00	1.12E+01	6.72E+00

⋮



例:Library_1_features.csv

h_-21	h_-20	...	v_5	v_6
-3.16E+01	-3.16E+01		-1.01E+00	-1.14E+00
-3.51E+01	-3.49E+01		-6.67E-01	-6.92E-01

⋮

⋮

移動有りデータセット

実験参加者16名(No.1-No.16), タスク15回(Library_1~ClassRoom_5)

オレンジ

移動無しデータセット

実験参加者117名(E0098-E1085), シーン22種(cubemap_0000~cubemap_0021) イエロー



例:cubemap_0000_head_gaze.csv

time	head_h	head_v	gaze_h	gaze_v	...
0.00E+00	1.19E+00	2.52E+00	-1.12E+00	-4.93E-02	
3.33E-02	1.28E+00	2.44E+00	-1.21E+00	7.36E-03	

⋮

ステップ4



Saliency_1.py



例:cubemap_0000_head_gaze.csv

head_h	head_v	gaze_h	gaze_v
-2.26E+00	3.59E+00	-1.14E+01	-1.96E+00
-3.29E+00	3.58E+00	-2.16E+01	-1.08E+00

⋮



例:cubemap_0000_features.csv

h_-21	h_-20	...	v_5	v_6
3.45E+00	3.54E+00		1.35E-01	2.34E-01
4.57E+00	4.55E+00		2.36E-01	2.74E-01

⋮

⋮

眼球・頭部協調運動モデルの作成(フロー図2)

ステップ3

murakami_2.py



head_h	head_v	gaze_h	gaze_v
2.74E+01	5.70E+00	9.63E+00	5.90E+00
3.09E+01	5.23E+00	1.12E+01	6.72E+00

例:No.1_head_gaze.csv

⋮



h_-21	h_-20	...	v_5	v_6
-3.16E+01	-3.16E+01	...	-1.01E+00	-1.14E+00
-3.51E+01	-3.49E+01	...	-6.67E-01	-6.92E-01

例:No.1_features.csv

⋮

⋮

ステップ6

gaze_direction_1.py



移動有りモデル

ステップ5

Saliency_2.py



head_h	head_v	gaze_h	gaze_v
-2.26E+00	3.59E+00	-1.14E+01	-1.96E+00
-3.29E+00	3.58E+00	-2.16E+01	-1.08E+00

例:E0098_head_gaze.csv

⋮



h_-21	h_-20	...	v_5	v_6
3.45E+00	3.54E+00	...	1.35E-01	2.34E-01
4.57E+00	4.55E+00	...	2.36E-01	2.74E-01

例:E0098_features.csv

⋮

⋮

ステップ6

gaze_direction_1.py



移動無しモデル

ステップ1(スライド29の青矢印)



左手座標系

- ・ フレームレート30(fps)にリサンプリングした計測データ(VR_gaze-head_data_in_VR_30fps内のcsvファイル)から頭部と視線方向の水平角・垂直角を算出し, CSVファイルとして保存する.
- ・ 使用するプログラムはVR_mistake_rawdata_to_degree.py
- ・ 出力先はgaze-head_data_in_VR_30fps_degreeフォルダ
- ・ 頭部と視線方向の水平角・垂直角⇒右手座標系

計測データ



例:Library_1.csv

(VR_gaze-head_data_in_VR_30fps
¥No.1)

time	h_p_x	h_p_y	h_p_z
0.00E+00	2.76E-01	1.62E+00	3.18E-01
3.33E-02	2.74E-01	1.62E+00	3.27E-01
6.67E-02	2.74E-01	1.62E+00	3.35E-01

• • •



頭部と視線方向の水平角・垂直角



例:Library_1_deg.csv

(gaze-head_data_in_VR_30fps_degree
¥No.1)

time	head_h	head_v	gaze_h	gaze_v
0.00E+00	-4.17E+00	-6.59E-01	3.28E+00	1.60E+01
3.33E-02	-4.19E+00	-1.24E+00	3.58E+00	1.64E+01
6.67E-02	-4.03E+00	-5.16E-01	3.90E+00	1.80E+01

•
•
•

ステップ2 (スライド29の青矢印)

- ・ 頭部と視線方向の水平角・垂直角(gaze-head_data_in_VR_30fps_degree内のcsvファイル)から 頭部・視線方向データ, 頭部方向時系列データをシーンごとに作成し, CSVファイルとして保存する.
- ・ 使用するプログラムはmurakami_1.py
- ・ 出力先はcreated_datasetフォルダ
- ・ 頭部・視線方向データ, 頭部方向時系列データ⇒右手座標系

頭部と視線方向の水平角・垂直角



time	head_h	head_v	gaze_h	gaze_v
0.00E+00	-4.17E+00	-6.59E-01	3.28E+00	1.60E+01
3.33E-02	-4.19E+00	-1.24E+00	3.58E+00	1.64E+01
6.67E-02	-4.03E+00	-5.16E-01	3.90E+00	1.80E+01

例:Library_1_deg.csv

(gaze-head_data_in_VR_30fps_degree
¥No.1)

・
・
・

シーンごと

頭部・視線方向データ



head_h	head_v	gaze_h	gaze_v
2.74E+01	5.70E+00	9.63E+00	5.90E+00
3.09E+01	5.23E+00	1.12E+01	6.72E+00
3.49E+01	5.06E+00	1.63E+01	9.22E+00

例:Library_1_head_gaze.csv
(created_dataset¥SD_5.0¥No.1)

・
・
・

頭部方向時系列データ



h_-21	h_-20	h_-19
-3.16E+01	-3.16E+01	-3.14E+01
-3.51E+01	-3.49E+01	-3.45E+01
-3.90E+01	-3.86E+01	-3.77E+01

例:Library_1_features.csv
(created_dataset¥SD_5.0¥No.1)

・
・
・

...

ステップ3 (スライド30の青矢印)

- ・ 頭部・視線方向データ, 頭部方向時系列データを実験参加者ごとに作成し, CSVファイルとして保存する.
- ・ 使用するプログラムはmurakami_2.py
- ・ 出力先はcreated_datasetフォルダ
- ・ 頭部・視線方向データ, 頭部方向時系列データ⇒右手座標系



例:Library_1_head_gaze.csv
(created_dataset¥SD_5.0¥No.1)

頭部・視線方向データ



例:Library_1_features.csv
(created_dataset¥SD_5.0¥No.1)

頭部方向時系列データ



例:No.1_head_gaze.csv
(created_dataset¥SD_5.0¥No.1)

頭部・視線方向データ



例:No.1_features.csv
(created_dataset¥SD_5.0¥No.1)

頭部方向時系列データ

シーンごと

実験参加者ごと

ステップ4 (スライド29の青矢印)

- ・ 頭部と視線方向の水平角・垂直角(dataset_head_gaze_direction内のcsvファイル)から 頭部・視線方向データ, 頭部方向時系列データをシーンごとに作成し, CSVファイルとして保存する.
- ・ 使用するプログラムはSaliency_1.py
- ・ 出力先はdataset_gaze-distribution2フォルダ
- ・ 頭部・視線方向データ, 頭部方向時系列データ⇒右手座標系

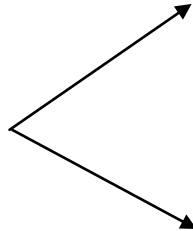
頭部と視線方向の水平角・垂直角



time	head_h	head_v	gaze_h	gaze_v
0.00E+00	1.19E+00	2.52E+00	-1.12E+00	-4.93E-02
3.33E-02	1.28E+00	2.44E+00	-1.21E+00	7.36E-03
6.67E-02	1.26E+00	2.35E+00	-1.24E+00	1.33E-01

例:cubemap_0000_head_gaze.csv
(dataset_head_gaze_direction¥E0098)

・
・
・



シーンごと

頭部・視線方向データ



head_h	head_v	gaze_h	gaze_v
-2.26E+00	3.59E+00	-1.14E+01	-1.96E+00
-3.29E+00	3.58E+00	-2.16E+01	-1.08E+00
-4.74E+00	3.55E+00	-2.27E+01	-2.38E-01

例:cubemap_0000_head_gaze.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)

・
・
・

頭部方向時系列データ



h_-21	h_-20	h_-19
3.45E+00	3.54E+00	3.52E+00
4.57E+00	4.55E+00	4.46E+00
6.00E+00	5.91E+00	5.81E+00

...

例:cubemap_0000_features.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)

・
・
・

ステップ5 (スライド30の青矢印)

- ・ 頭部・視線方向データ, 頭部方向時系列データを実験参加者ごとに作成し, CSVファイルとして保存する.
- ・ 使用するプログラムはSaliency_2.py
- ・ 出力先はdataset_gaze-distribution2フォルダ
- ・ 頭部・視線方向データ, 頭部方向時系列データ⇒右手座標系



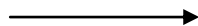
頭部・視線方向データ

例: cubemap_0000_head_gaze.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)



頭部方向時系列データ

例: cubemap_0000_features.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)



頭部・視線方向データ

例: E0098_head_gaze.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)



頭部方向時系列データ

例: E0098_features.csv
(dataset_gaze-distribution2¥SD_5.0¥E0098)

シーンごと

実験参加者ごと

ステップ6 (スライド30の青矢印)

- ・ 頭部・視線方向データ, 頭部方向時系列データから
眼球・頭部協調運動モデル(移動有りモデル・移動無しモデル)を作成する.
- ・ 使用するプログラムはgaze_direction_1.py
- ・ 出力先はmodel_h_out_pathフォルダ, model_v_out_pathフォルダ
→プログラム実行時にフォルダが作成される



移動有りモデル

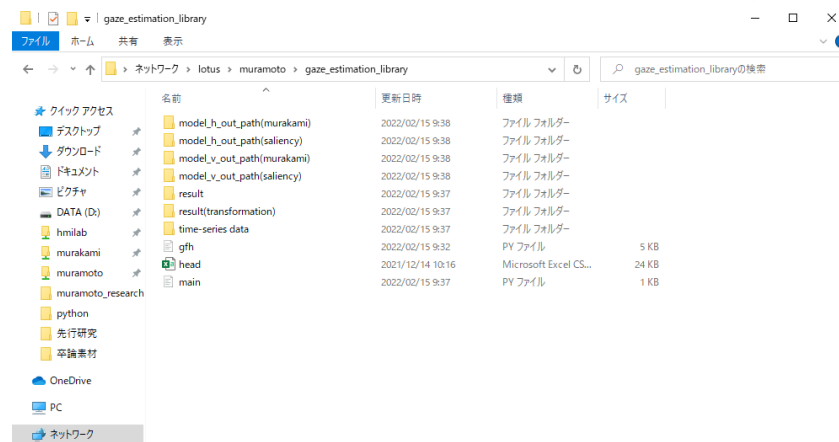
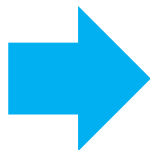
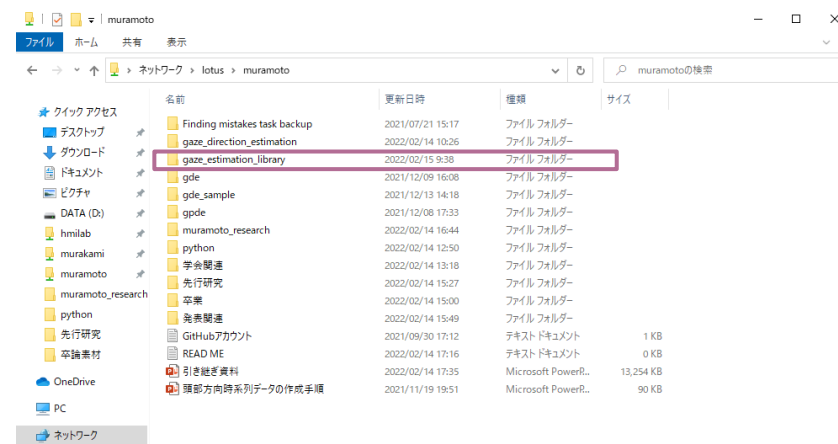
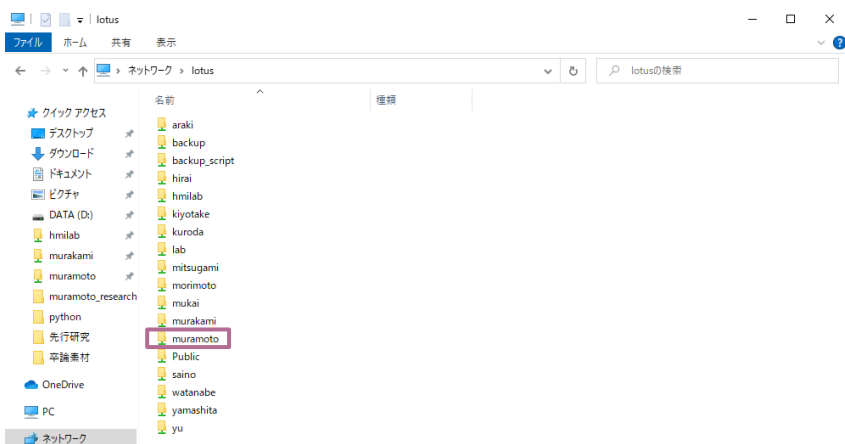


移動無しモデル

視線推定ライブラリの使い方

準備

- lotusのmuramotoフォルダ内のgaze_estimation_libraryフォルダをデスクトップ等にコピーする。



中身について

- head.csv → フレームレート30(fps)にリサンプリング, ± 180 度を跨いだ際の処理_(1)をした頭部運動データ
- model_h_out_path(murakami)フォルダ, model_v_out_path(murakami)フォルダ, model_h_out_path(saliency)フォルダ, model_v_out_path(saliency)フォルダ, → 学習済みモデル_(2)
- time-series dataフォルダ → 頭部方向データと頭部方向時系列データ
- resultフォルダ → 推定結果(頭部基準の視線)
- result(transformation) フォルダ → 推定結果を変換(頭部基準の視線から世界座標基準の視線へ)

Oculus Goの
視線復元

推定された視線と頭部→右と下が正, 左と上が負の座標系

360度を足し込む

(1)について



時間	頭部データ(計測時)
0.0[sec]	179[deg]
1.0[sec]	-179[deg]
2.0[sec]	-177[deg]



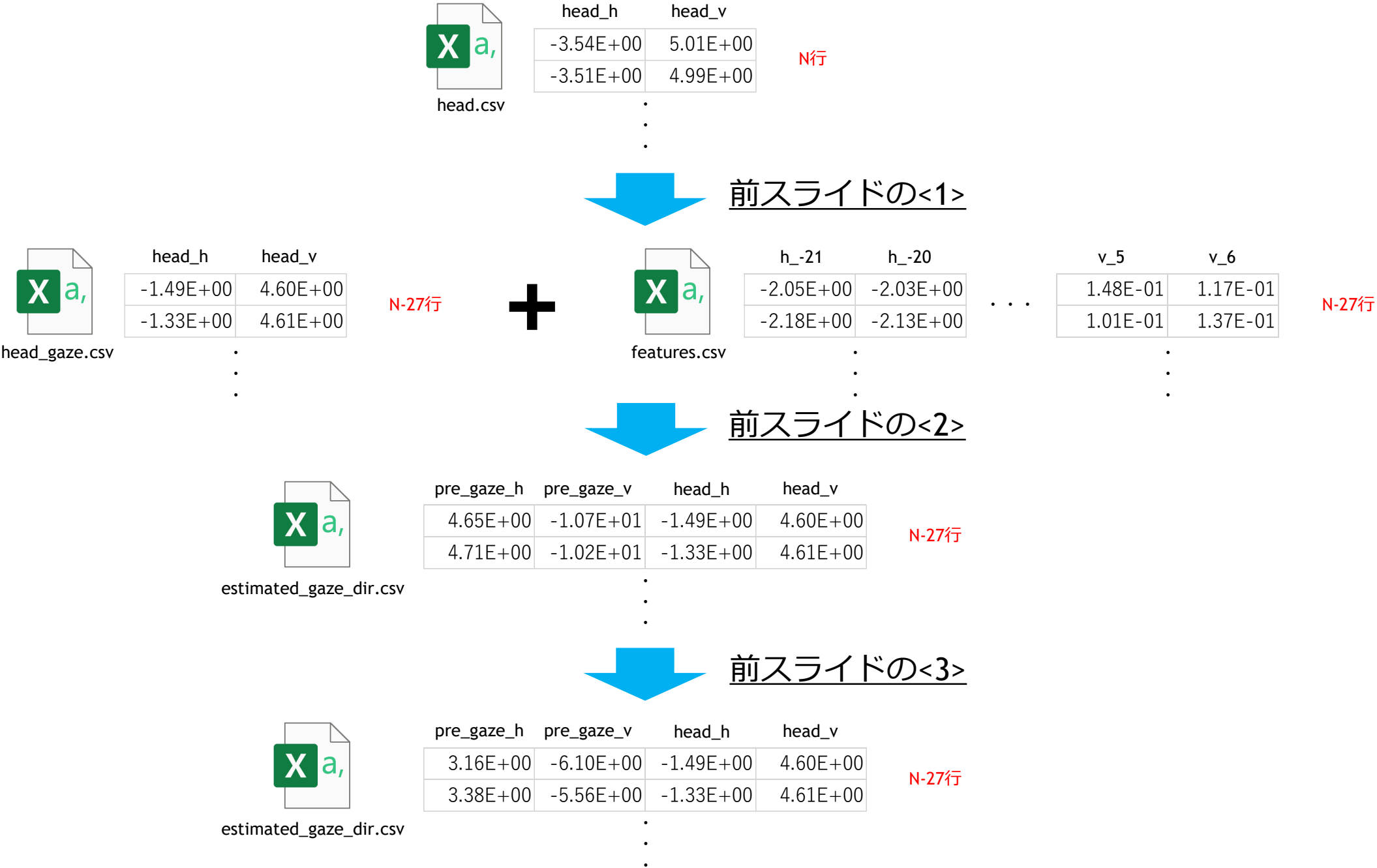
時間	頭部データ(処理後)
0.0[sec]	179[deg]
1.0[sec]	181[deg]
2.0[sec]	183[deg]

(2)学習済みモデルは, 移動有りモデル(murakami), 移動無しモデル(saliency)の2種類存在する.

視線推定ライブラリ(GFH)

```
1  import gfh
2
3  # 頭部運動データの読み込み
4  head_seq = gfh.load_file( 'head.csv' ) ← <1>
5
6  # 視線推定
7  gaze_seq = gfh.estimate_gaze( head_seq, model = "3DoF" ) ← <2>
8
9  # 頭部基準から世界座標基準へ変換
10 gaze_seq_global = gfh.trans_to_global( gaze_seq ) ← <3>
```

視線推定ライブラリのフロー図



gfh.pyとmain.pyについて

- gfh.py⇒main.pyで使用する関数を定義している.
- main.py

```
1 import gfh
2
3 # 頭部運動データの読み込み
4 head_seq = gfh.load_file('head.csv')
5
6 # 視線推定
7 gaze_seq = gfh.estimate_gaze(head_seq, model = "3DoF")
8
9 # 頭部基準から世界座標基準へ変換
10 gaze_seq_global = gfh.trans_to_global(gaze_seq)
```

頭部運動データのcsvファイルを指定

移動有りモデルなら**6DoF**
移動無しモデルなら**3DoF**を指定

lightgbmのインストール

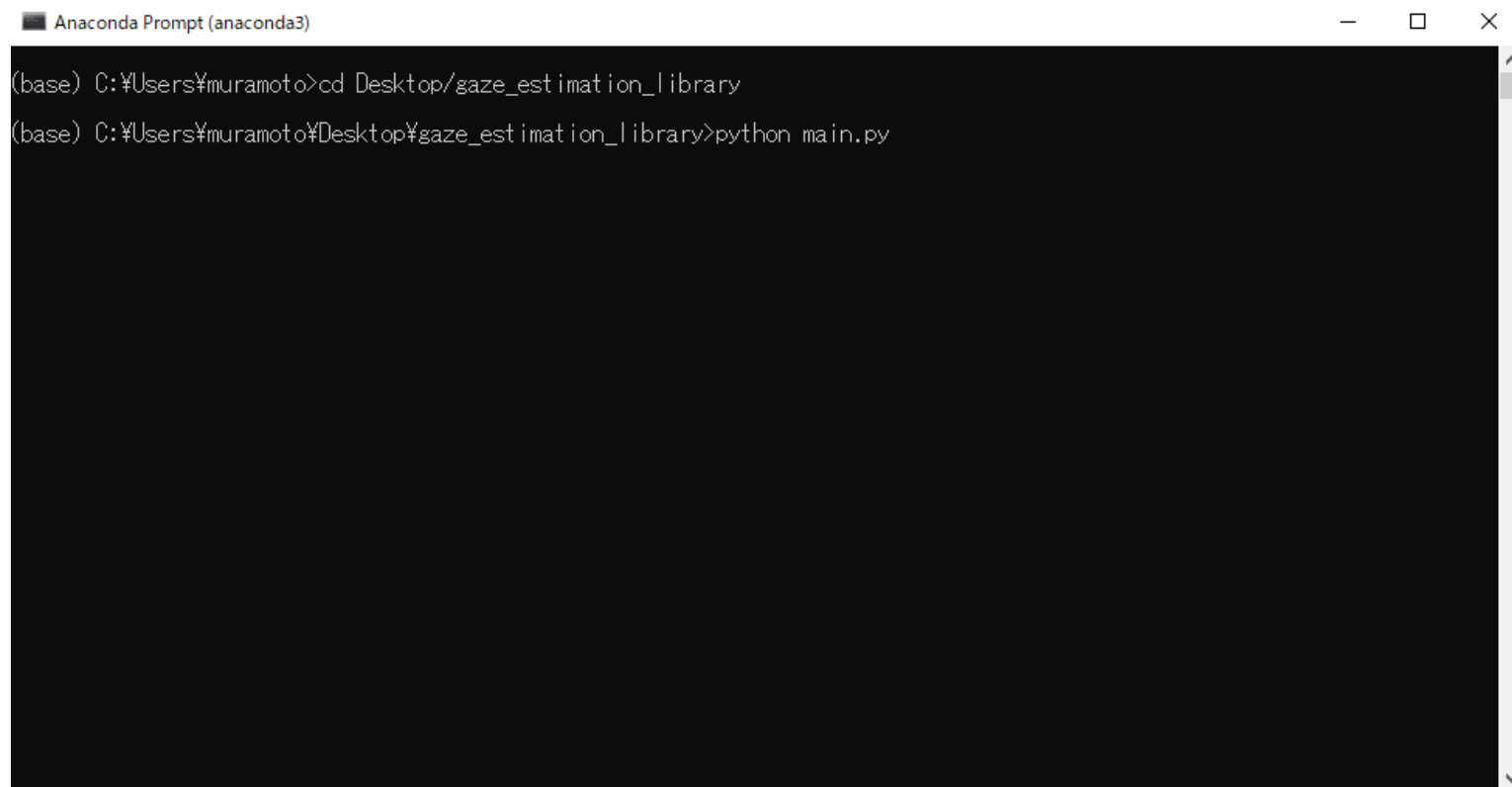
- Anaconda上で, `conda install -c conda-forge lightgbm`と入力する. (既に, インストール済みの場合は不要)



LightGBM

実行

- ・ デスクトップにgaze_estimation_libraryフォルダを配置した場合の実行例



```
Anaconda Prompt (anaconda3)
(base) C:\Users\muramoto>cd Desktop/gaze_estimation_library
(base) C:\Users\muramoto\Desktop\gaze_estimation_library>python main.py
```


time-series dataフォルダ

- head_gaze.csv(頭部方向データ)



head.csv

head_h	head_v
-3.54E+00	5.01E+00
-3.51E+00	4.99E+00
-3.46E+00	4.84E+00

•
•
•

N行

3.87E+02	1.81E+00
3.87E+02	1.76E+00
3.87E+02	1.71E+00

頭部の参照時間幅
 $t_s = -0.7(\text{sec})$, $t_e = 0.2(\text{sec})$



head_gaze.csv

head_h	head_v
-1.49E+00	4.60E+00
-1.33E+00	4.61E+00
-1.21E+00	4.69E+00

•
•
•

N-27行

3.87E+02	2.25E+00
3.87E+02	2.09E+00
3.87E+02	2.05E+00

※行数が減っている(頭部の参照時間幅の影響により)

time-series dataフォルダ

- features.csv(頭部方向時系列データ)



head.csv

N行

head_h	head_v
-3.54E+00	5.01E+00
-3.51E+00	4.99E+00
-3.46E+00	4.84E+00

•
•
•

3.87E+02	1.81E+00
3.87E+02	1.76E+00
3.87E+02	1.71E+00

頭部の参照時間幅
 $t_s = -0.7(\text{sec})$, $t_e = 0.2(\text{sec})$



features.csv

N-27行56列

h_-21	h_-20	h_-19
-2.05E+00	-2.03E+00	-1.97E+00
-2.18E+00	-2.13E+00	-2.11E+00
-2.25E+00	-2.24E+00	-2.26E+00

•
•
•

※行数が減っている(頭部の参照時間幅の影響により)

resultフォルダ

- estimated_gaze_dir.csv(推定結果<頭部基準>)



gaze_h(推定)	gaze_v(推定)	head_h(計測)	head_v(計測)
4.65E+00	-1.07E+01	-1.49E+00	4.60E+00
4.71E+00	-1.02E+01	-1.33E+00	4.61E+00
4.10E+00	-1.00E+01	-1.21E+00	4.69E+00

•
•
•

N-27行

1.69E+00	-1.24E+01	3.87E+02	2.25E+00
1.97E+00	-1.16E+01	3.87E+02	2.09E+00
-1.93E+00	-1.21E+01	3.87E+02	2.05E+00

result(transformation)フォルダ

- estimated_gaze_dir.csv(推定結果<世界座標基準>)



gaze_h(推定)	gaze_v(推定)	head_h(計測)	head_v(計測)
3.16E+00	-6.10E+00	-1.49E+00	4.60E+00
3.38E+00	-5.56E+00	-1.33E+00	4.61E+00
2.89E+00	-5.33E+00	-1.21E+00	4.69E+00

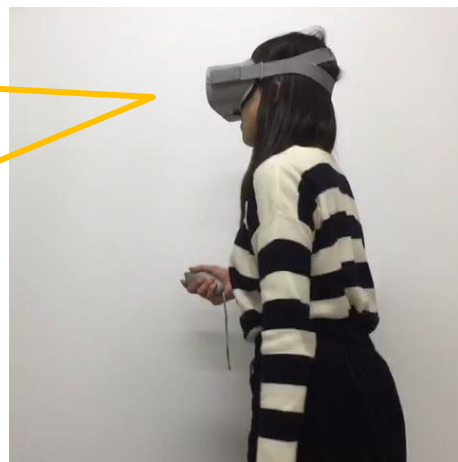
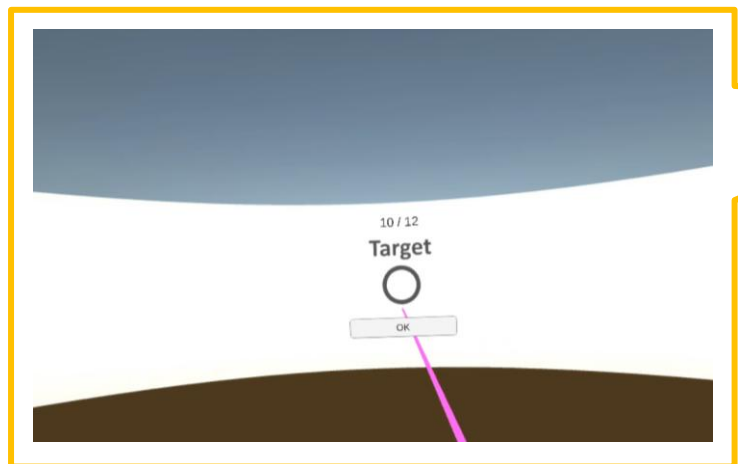
•
•
•

N-27行

3.89E+02	-1.02E+01	3.87E+02	2.25E+00
3.89E+02	-9.54E+00	3.87E+02	2.09E+00
3.85E+02	-1.01E+01	3.87E+02	2.05E+00

視線推定ライブラリの利用例

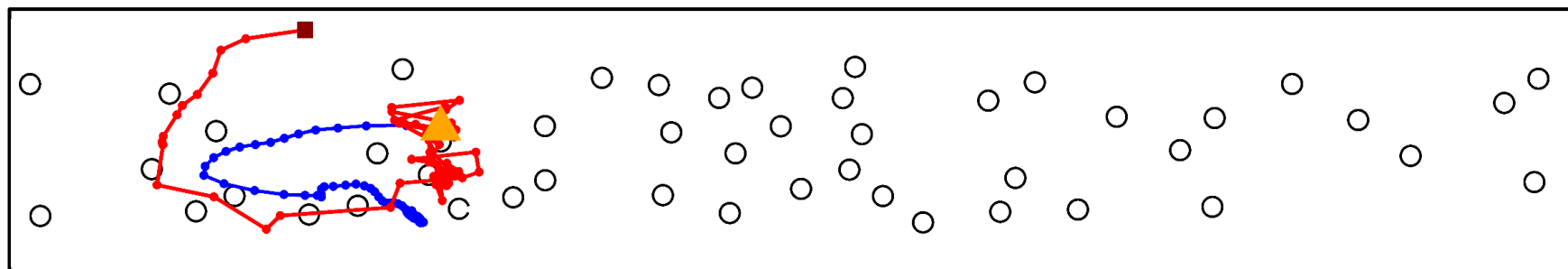
利用例



アイトラッカーを搭載できないため、分析の対象は、ユーザの頭部運動のみ

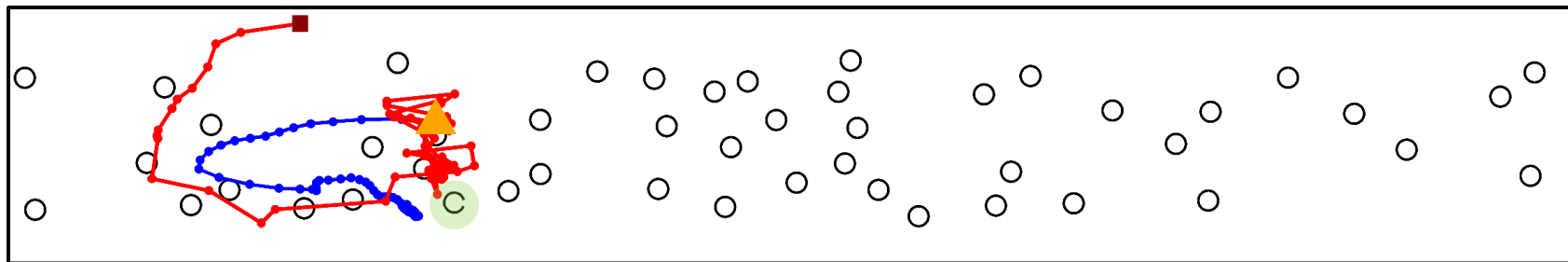
VR視覚探索アプリケーション[2]

— : 頭部(計測) — : 視線(推定)

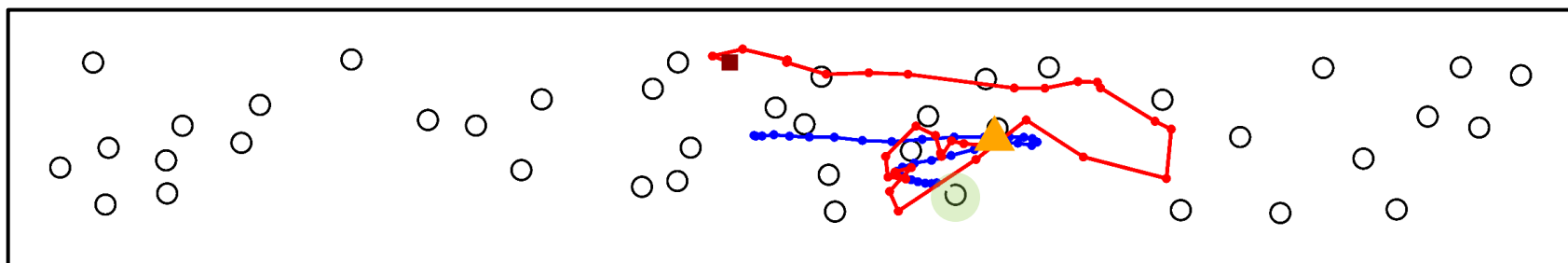


視線推定が良好な例

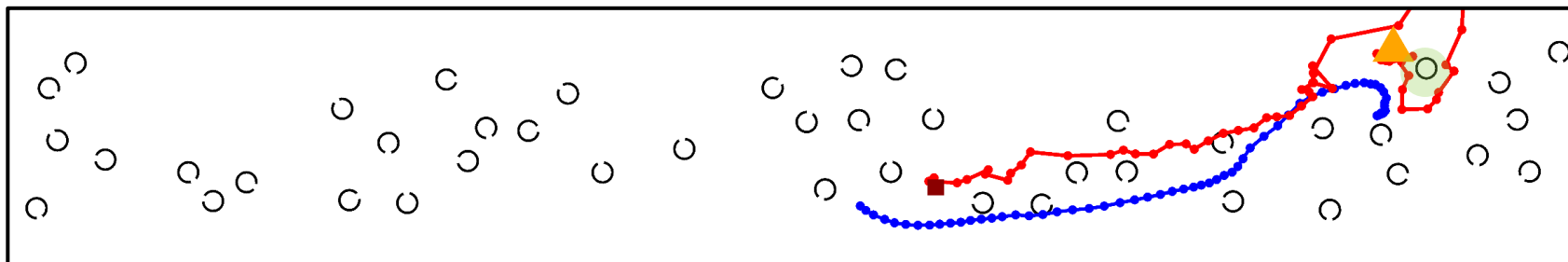
推定結果A



推定結果B



推定結果C

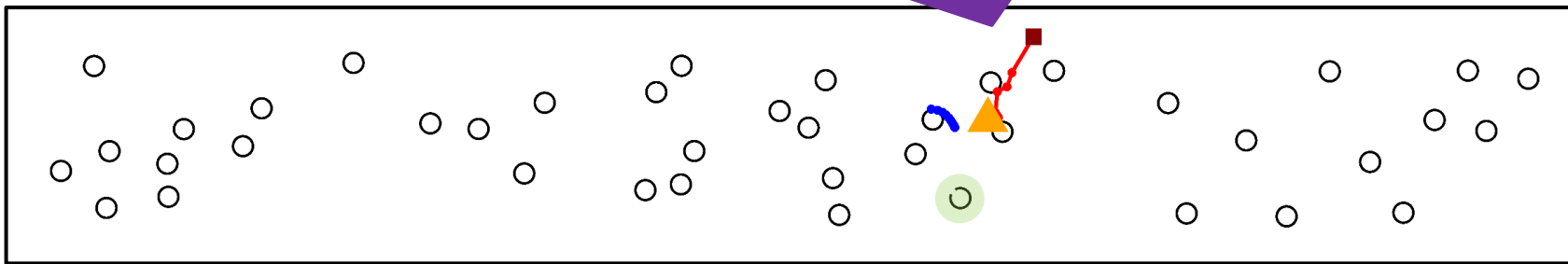


- : 頭部(計測)
- : 視線(推定)
- : 開始点(視線)
- ▲ : 終了点(視線)

視線推定が不正確・失敗するケース

頭部運動が
非常に少ない

推定結果D



- : 頭部(計測)
- : 視線(推定)
- : 開始点(視線)
- ▲ : 終了点(視線)
- : 途中経過(10秒毎)

推定結果E

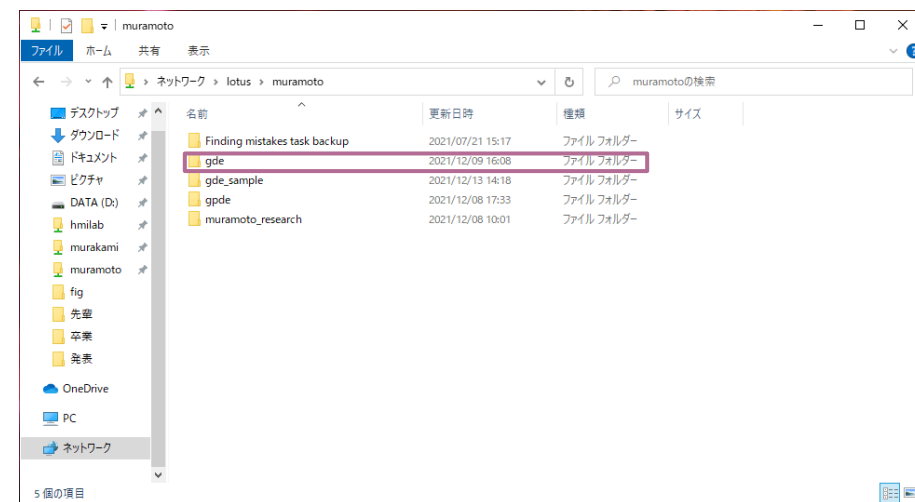
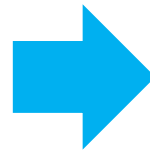
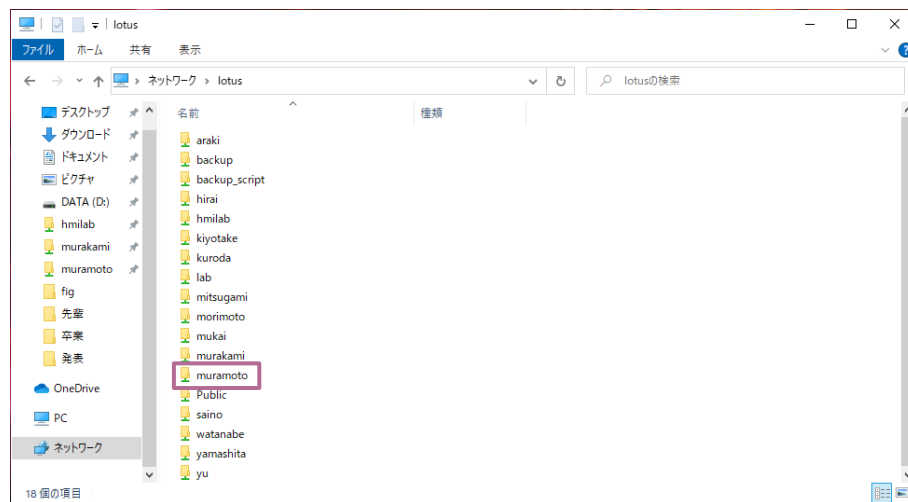


不自然な頭部運動

視線推定ライブラリ (ライブラリ化前)の使い方

準備①

- lotusのmuramotoフォルダ内のgdeフォルダをデスクトップ等にコピーする。




準備②





- GitHubのmitsugamilab/gaze_from_headリポジトリのgdeフォルダをダウンロードし、その中身を先程デスクトップ等にコピーしたgdeフォルダ内に移す.

https://github.com/mitsugamilab/gaze_from_head

main ▾ 1 branch 0 tags Go to file Add file ▾ Code ▾

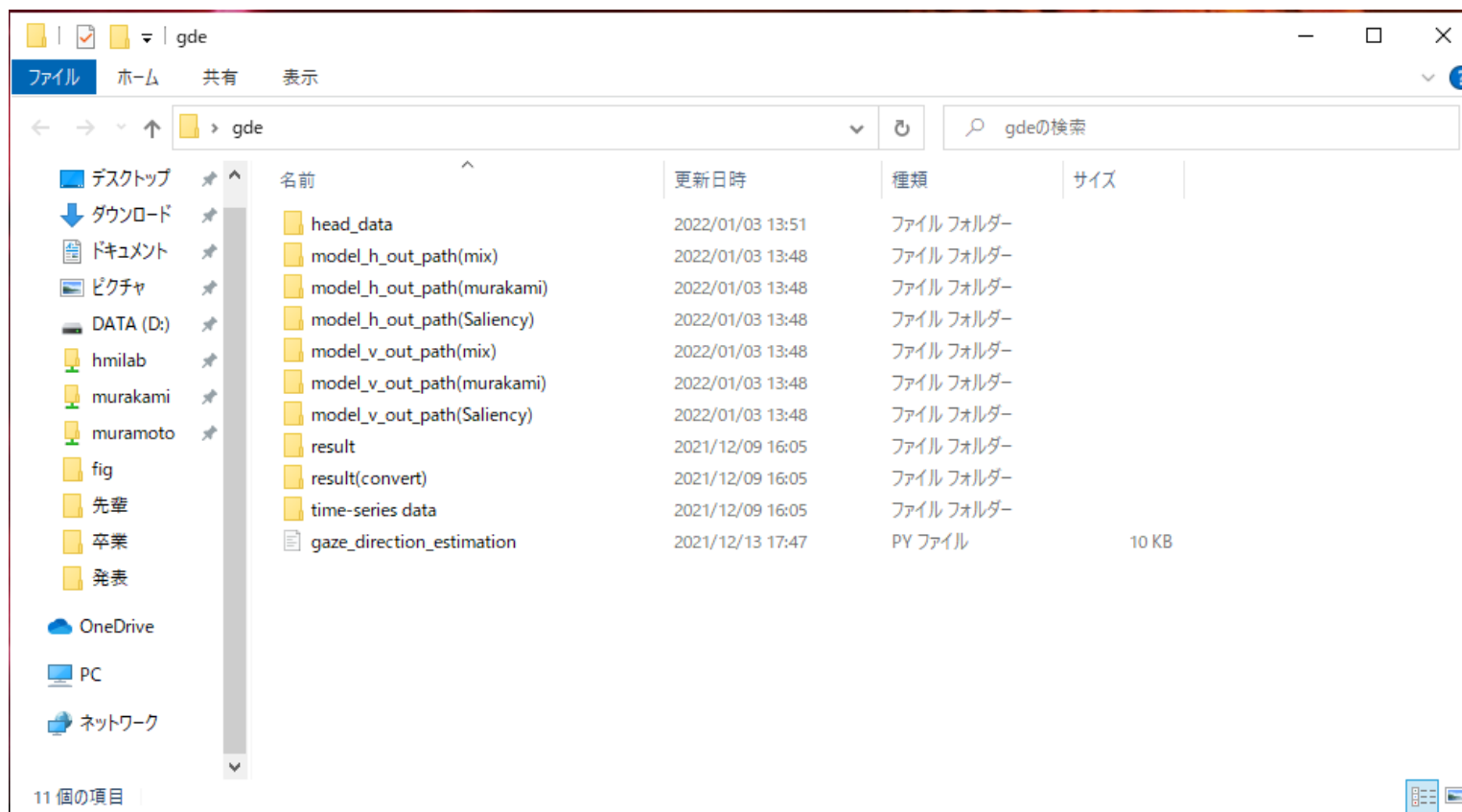
 MuramotoShuto Create README.md

2b7b32e 20 days ago 68 commits

	gde	gde	20 days ago
	gde_sample	gde_sample	20 days ago
	gpde	gpde	20 days ago
	README.md	Create README.md	20 days ago

確認

- gdeフォルダ内の構成が以下になっていることを確認する。



各フォルダについて

- head_dataフォルダ → 頭部運動データ
- model_h_out_path(murakami)フォルダ, model_v_out_path(murakami)フォルダ,
model_h_out_path(Saliency)フォルダ, model_v_out_path(Saliency)フォルダ, → 学習済みモデル_(1)
model_h_out_path(mix)フォルダ, model_v_out_path(mix)フォルダ
- time-series dataフォルダ → 頭部方向データと頭部方向時系列データ
- result フォルダ → 推定結果
- result(convert)フォルダ → 推定結果を変換

(1)学習済みモデルは、移動有りモデル(murakami)、移動無しモデル(Saliency)、移動有り・無しモデル(mix)の
3種類存在する。

lightgbmのインストール

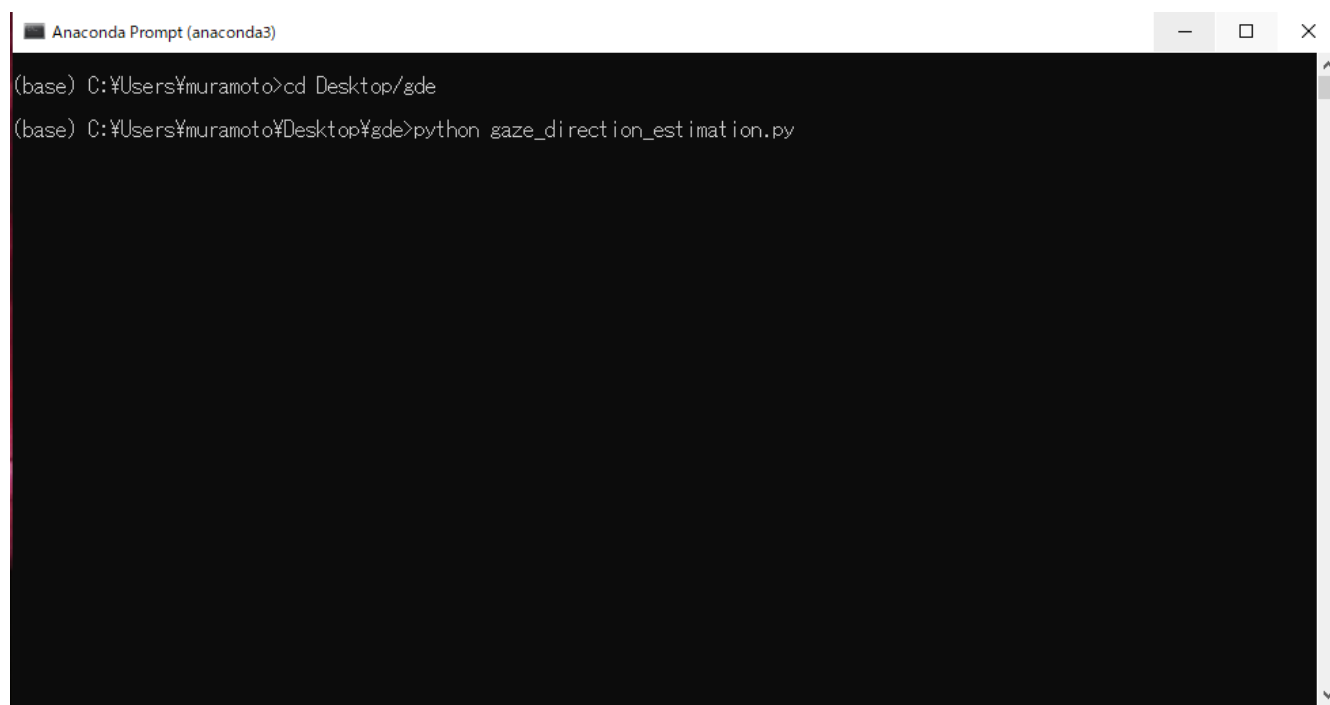
- Anaconda上で, `conda install -c conda-forge lightgbm`と入力する. (既に, インストール済みの場合は不要)



LightGBM

実行

- ・ 視線方向推定プログラムを実行する。(↓デスクトップにgdeフォルダを配置した場合の実行例)



```
Anaconda Prompt (anaconda3)
(base) C:\Users\muramoto>cd Desktop/gde
(base) C:\Users\muramoto\Desktop\gde>python gaze_direction_estimation.py
```

補足

- ・ 学習済みモデルを変更したい場合. (初期状態は移動無しモデル(Saliency))



gaze_direction_estimation.pyの100行~105行目を
適宜コメントアウトする!!!

```
100 #model_h_out_path = "../gde/model_h_out_path(murakami)/"
101 #model_v_out_path = "../gde/model_v_out_path(murakami)/"
102 model_h_out_path = "../gde/model_h_out_path(Saliency)/"
103 model_v_out_path = "../gde/model_v_out_path(Saliency)/"
104 #model_h_out_path = "../gde/model_h_out_path(mix)/"
105 #model_v_out_path = "../gde/model_v_out_path(mix)/"
```