

設定ファイル

設定ファイルは JSON 形式です。このファイル名をコマンドライン引数に指定してください。指定した設定ファイルが作業ディレクトリに存在しなければ、デフォルト値を設定したその名前のファイルを作成します。ファイル名を指定しなければ、`config.json` というファイル名が用いられます。

```
{
  "left_camera": <左のカメラの番号またはムービーファイル名, 負なら使用しない ※1>,
  "left_image": "<左のカメラを使用しないときに使う静止画像ファイル名>",
  "right_camera": <右のカメラの番号またはムービーファイル名, 負なら使用しない ※1>,
  "right_image": "<右のカメラを使用しないときに使う静止画像ファイル名>",
  "capture_width": <カメラからキャプチャする画像の幅, 0 ならカメラから取得>,
  "capture_height": <カメラからキャプチャする画像の高さ, 0 ならカメラから取得>,
  "capture_fps": <カメラからキャプチャする際のフレームレート, 0 ならカメラから取得>,
  "ovrvision_property": <Ovrvision Pro の動作モード ※2>,
  "texture_repeat": <THETA S で撮影した正距円筒図法の画像を用いる時は true, 通常は false>,
  "stereo": <表示するディスプレイの立体視方式 ※3>,
  "use_secondary": <全画面表示をプライマリディスプレイに行う場合は 0, それ以外なら 1 以上>,
  "depth_near": <モデル表示の前方面までの距離, 単位は m>,
  "depth_far": <モデル表示の後方面までの距離, 単位は m>,
  "display_center": <表示するディスプレイの高さの 1/2, 単位は m>,
  "display_distance": <表示するディスプレイまでの距離, 単位は m>,
  "screen_samples": <魚眼レンズ画像の平面展開に使用するメッシュの格子点数>,
  "zoom": <操縦者の視野のズーム率の初期値, 標準は 1.0>,
  "fisheye_center_x": <操縦者側の魚眼レンズのイメージサークルの中心の水平位置>,
  "fisheye_center_y": <操縦者側の魚眼レンズのイメージサークルの中心の垂直位置>,
  "fisheye_fov_x": <操縦者側の魚眼レンズの水平方向の画角の初期値, 単位ラジアン>,
  "fisheye_fov_y": <操縦者側の魚眼レンズの垂直方向の画角の初期値, 単位ラジアン>,
  "vertex_shader": "<入力画像の書式に対応したバーテックスシェーダのファイル名 ※4>",
  "fragment_shader": "<入力画像の書式に対応したフラグメントシェーダのファイル名 ※5>",
  "port": <画像のネットワーク転送に使用するポート番号>,
  "host": "<画像のネットワーク転送先の IP アドレス>",
  "role": <役割, スタンドアローンなら 0, 作業側（ロボット）側 1, 操縦側（科学者）側 2>,
  "texture_quality": <ネットワーク転送する画像の JPEG 圧縮の品質, 単位 %>,
  "texture_reshape": <ネットワーク転送する画像をドーム画像に変形するなら true>,
  "texture_samples": <ドーム画像の変形に用いるメッシュの格子点数>,
  "remote_fov_x": <ドーム画像に変形するロボット側の平面カメラの水平方向の画角, 単位ラジアン>,
  "remote_fov_y": <ドーム画像に変形するロボット側の平面カメラの垂直方向の画角, 単位ラジアン>,
  "scene": <重畳表示するシーンに固定したシーングラフノード ※6>,
  "target": <重畳表示する操縦者の頭の動きに追従するシーングラフノード ※6>,
  "remote": <重畳表示するロボットの頭の動きに追従するシーングラフノード ※6>,
}
```

```

"tracking": <HMD に表示する際にヘッドトラッキングするなら true>,
"tracking_delay_left": <左目の画像に対してヘッドトラッキングを遅らせるフレーム数>,
"tracking_delay_right": <右目の画像に対してヘッドトラッキングを遅らせるフレーム数>
}

```

※1 映像入力を選択

	left_camera < 0	left_camera >= 0
right_camera < 0	静止画像ファイルを使用	左カメラのみを使用（単眼）
right_camera >= 0	Ovrvision Pro を使用	左右のカメラを使用（両眼）

（ここに "movie.mp4" のようにムービーファイルを指定することもできます）

※2 Ovrvision Pro のプロパティ (OVR::Camprop)

0	OVR::OV_CAM5MP_FULLL	2560x1920	15fps x2
1	OVR::OV_CAM5MP_FHD	1920x1080	30fps x2
2	OVR::OV_CAMHD_FULLL	1280x960	45fps x2
3	OVR::OV_CAMVR_FULLL	960x950	60fps x2
4	OVR::OV_CAMVR_WIDE	1280x800	60fps x2
5	OV_CAMVR_VGA	640x480	90fps x2
6	OVR::OV_CAMVR_QVGA	320x240	120fps x2
7	OVR::OV_CAM20HD_FULLL	1280x960	(Only USB2.0) 15fps x2
8	OVR::OV_CAM20VR_VGA	640x480	(Only USB2.0) 30fps x2

（Ovrvision Pro を HMD に取り付けるときは 3 を指定してください）

※3 立体視の設定 (StereoMode)

NONE	単眼視
LINEBYLINE	インターレース（未実装）
TOPANDBOTTOM	上下2分割
SIDEBYSIDE	左右2分割
QUADBUFFER	クワッドバッファステレオ（フレームシーケンシャル）
OCULUS	Oculus Rift DK1 / DK2 または CV1

（Oculus Rift DK1 / DK2 と CV1 では実行ファイルが異なります）

※4 パーテックスシェーダのソースファイル

rectangle.vert	平面カメラを使用して視線を回転する（Ovrvision Pro）
fisheye.vert	光軸を水平にした魚眼カメラ
pixpro.vert	光軸を垂直にした魚眼カメラ（PIXPRO SP360 4K）
panorama.vert	正距円筒図法
theta.frag	RICOH THETA S のライブビデオ映像

※5 フラグメントシェーダのソースファイル

normal.frag	テクスチャ座標の画素をそのまま使う（通常）
panorama.frag	正距円筒図法
theta.frag	RICOH THETA S のライブビデオ映像

※6 シーングラフノード

<pre>{ "position": [<x>, <y>, <z>], "rotation": [<x>, <y>, <z>, <angle>], "scale": [<x>, <y>, <z>], "controller": <共有メモリ上の変換行列の番号> "model": "<三角形分割された Alias OBJ 形状データファイル名>", "children": [<シーングラフノード>, : <シーングラフノード>] }</pre>	
"position"	位置 <x>, <y>, <z>
"rotation"	回転軸ベクトル <x>, <y>, <z>, 回転角 <angle>
"scale"	拡大率 <x>, <y>, <z>
"model"	三角形分割された Alias OBJ 形状データファイル名
"children"	このノードの下位に接続するノード

（シーングラフノードのプロパティは省略可能）

設定ファイルの例

```
{
```

```
  "stereo": 5,
```

- 0 なら単眼視になります。Oculus Rift DK1/DK2/CV1 を使うときは 5 を指定してください。

```
  "use_secondary": 1,
```

- 表示を行うディスプレイの番号を指定します。0 ならプライマリディスプレイを使います。立体視は常にフルスクリーン表示になります。

```
  "screen_samples": 1271,
```

- 全天球画像の平面展開に使うメッシュの分割数です。数が多いほど変形の精度が上がりますが、そんなに上げる必要はありません。

```
  "display_center": 0.9,
```

```
  "display_distance": 1,
```

- 平面ディスプレイで立体視を行う場合のディスプレイの中心位置と視点からの距離です。単位は m です。

```
"depth_near": 0.1,  
"depth_far": 10,
```

- 視点から前方面および後方面までの距離です。単位は m です。

```
"zoom": 1.0,
```

- ズームの初期値です。Oculus Rift では 1 にしておく必要があります。

```
"capture_width": 0,  
"capture_height": 0,  
"capture_fps": 0,
```

- 映像のキャプチャに使うカメラの解像度とフレームレートです。0 にしておけばカメラに問い合わせますが、うまくいかないときは値を指定してください。

```
"fisheye_center_x": 0.0,  
"fisheye_center_y": 0.0,
```

- 映像のキャプチャに使うカメラ画像のイメージサークルの中心を、画像全体を縦横 [-1, 1] とする領域に対する相対値で指定してください。

```
"fisheye_fov_x": 1.0,  
"fisheye_fov_y": 1.0,
```

- 映像のキャプチャに使うカメラの魚眼レンズの画角の 1/2 をラジアンで指定してください。通常のカメラや RICOH THETA では 1 で構いません。

```
"ovrvision_property": 3,
```

- 映像のキャプチャに Ovrvision Pro を使うときのモードを指定します。ヘッドトラッキングを行わない場合（Ovrvision Pro を Oculus Rift に取り付ける場合）は 3 を指定してください。

```
"left_camera": -1,  
"left_image": "theta_left.jpg",  
"right_camera": -1,  
"right_image": "theta_right.jpg",
```

- 映像のキャプチャに使うカメラの番号、あるいは“ムービーファイル名”を left_camera および right_camera に指定してください。ここに -1 を指定したときは、それぞれ left_image および right_image に指定した静止画ファイルを使います。なお right_image が -1 のときは単眼視、left_image が -1 で right_image が 1 のときは Ovrvision Pro を使います。

```
"texture_repeat": false,
```

- キャプチャする映像が正距円筒図法（緯度経度画像）のときは true にしてください。

```
"vertex_shader": "theta.vert",  
"fragment_shader": "theta.frag",
```

- 全天球画像の平面展開に使うシェーダプログラムのソースファイルを指定してください。RICOH THETA S のライブストリーミングの二重魚眼画像を使うときは、それぞれ theta.vert、theta.frag を指定してください。UVC Blender による正距円筒図法（緯度経度画像）では、それぞれ panorama.vert と normal.frag を指定してください。

```
"port": 12345,
```

- 画像伝送に使うポート番号を指定してください。

```
"host": "192.168.30.1",
```

- 画像伝送する先の受信サーバの IP アドレスを指定してください。

```
"role": 0,
```

- コンピュータの役割を指定します。スタンドアローン（ネットワーク転送を行わない）なら 0, 作業者（ロボット）側なら 1, 操縦者（科学者）側なら 2 を指定してください。

```
"texture_quality": 50,
```

- 画像伝送時に送信側で JPEG 圧縮する際の品質を 0～100 の値で指定してください。

```
"texture_reshape": false,
```

- 画像伝送時に受信側で全天球画像に変形する場合は true にしてください。

```
"texture_samples": 1271,
```

- 平面画像の全天球画像変換に使うメッシュの分割数です。

```
"remote_fov_x": 1,
```

```
"remote_fov_y": 1,
```

- 平面画像の全天球画像変換の際のカメラの縦横の画角を指定してください。例えば 1280×720 画素（HD）のカメラで水平方向の画角が 60° であれば,

```
"texture_samples": 1271,
```

- 平面画像の全天球画像変換に使うメッシュの分割数です。

```
"scene": "scene.json",
```

```
"target": {
```

```
  "position": [ 0, 0, -0.5 ],
```

```
  "scale": [ 0.5, 0.5, 0.5 ],
```

```
  "model": "target.obj"
```

```
},
```

```
"remote": {
```

```
  "position": [ 0, 0, -0.5 ],
```

```
  "scale": [ 0.5, 0.5, 0.5 ],
```

```
  "model": "remote.obj"
```

```
}
```

```
}
```