

# Intel RealSense Viewer マニュアル（日本語）

本マニュアルでは、Intel RealSense Viewer のインストールから基本操作・フィルタ解説・推奨設定・データ収集の注意点までを体系的に説明します。視覚障害者支援ナビゲーション用の研究用途を前提に、屋外歩行でのデータ収集に必要な要点も盛り込みました。

---

## 1. RealSense SDK（Windows版）インストール

### 1.1 ダウンロード

公式 GitHub から最新版の Windows 用 RealSense SDK をダウンロードします。

1. <https://github.com/IntelRealSense/librealsense/releases> を開く
2. 「Assets」から **Intel.RealSense.SDK-Win10.exe** をダウンロード
3. 実行してインストール

### 1.2 インストーラで選択される主な構成

- RealSense Viewer
- Firmware Updater
- C++/Python SDK（必要に応じて）

RealSense Viewer だけであれば追加設定不要で使用可能です。

---

## 2. RealSense Viewer の基本操作

### 2.1 起動

Windows のスタートメニュー → **RealSense Viewer** で起動。

### 2.2 カメラの接続

- USB3.0（青ポート）かつ付属ケーブルを推奨
- 接続後、Viewer が自動認識

### 2.3 主要パネル

- **Sensors パネル**：RGB/Depth センサの設定
  - **Streaming パネル**：解像度・fps
  - **Post-Processing パネル**：深度フィルタ
  - **3D/2D Visualization パネル**：可視化設定
-

## 3. Depth (深度) 設定

### 3.1 解像度

RealSense D455 の一般的な選択肢： - 1280×720 (高解像度・屋外向け) - 848×480 (標準) - 640×480 (軽量・屋外でも安定)

あなたが収集した 640×480 は YOLO 学習に十分対応します。YOLOv8/YOLOv10 は 640×640 入力を想定しているため、標準的です。

### 3.2 表示レンジ (Min/Max Distance)

- 屋外歩行でのナビ用途 → **Min 1.0m / Max 16.0m** が適切
  - 4m以上が表示されないのはデフォルトが 4m だから
  - Max 距離を上げれば表示可能
- 

## 4. Depth Post-Processing フィルタ (重要)

あなたが現状 ON にしているフィルタ： - ✓ Decimation Filter - ✓ HDR Merge - ✓ Depth to Disparity - ✓ Spatial Filter - ✓ Hole Filling Filter - ✓ Disparity to Depth

これは RealSense 社が推奨する典型的なノイズ低減パイプラインとなっており、非常に良い構成です。

以下、それぞれの詳細を記述します。

---

## 5. フィルタ詳細解説 (研究者向け)

### 5.1 Decimation Filter (解像度縮小)

- Depth 画像の解像度を 1/2, 1/4 に縮小
- ノイズが減る + データ処理が高速
- 点群生成パイプラインでも有効

研究用途での利点：YOLO に深度特徴を混ぜる場合、サイズ縮小により軽量化できる

---

### 5.2 HDR Merge (HDR 深度合成)

- 複数露光データを合成し、暗所・逆光に強い深度を生成
- 屋外歩行では光源が一定でないため有効

注意：処理コストはやや高い

---

## 5.3 Depth to Disparity (深度→視差変換)

- Stereo カメラの基準データである「視差」に一旦変換
- Spatial / Temporal フィルタが最適に作用

RealSense の 公式推奨パイプラインの一部です。

---

## 5.4 Spatial Filter (空間ノイズ除去)

- 深度の平面性が向上、点状ノイズ除去
- 壁・道路など大きな平面が綺麗になる

ナビ用途では最重要フィルタ。

---

## 5.5 Hole Filling Filter (穴埋め)

- 深度が欠損したピクセル（黒い穴）を補完
- 遠距離（10m～）や日陰で生じやすい欠損を改善

YOLO などの学習に深度を使う場合に効果大。

---

## 5.6 Disparity to Depth (視差→深度)

- Disparity に変換した深度を元に戻す
- 終端処理として行われる

# 6. 推奨フィルタ構成 (Intel公式)

```
Decimation →  
Depth to Disparity →  
Spatial →  
Temporal (任意) →  
Hole Filling →  
Disparity to Depth
```

あなたの設定はこの公式推奨にはほぼ一致しています。

---

## 7. Viewer での録画 (bag記録)

### 7.1 設定

左上の赤丸ボタン → Record - 保存先 : 外付け SSD を推奨 - フォーマット : **.bag** (全センサ値を保持)

### 7.2 bag の特徴

- 再生時に全てのストリームを復元可能
- Depth → 16bit、RGB → 8bit
- Python / C++ で容易に解析可能

---

## 9. データ収集の注意点

データ収集の注意点

- 屋外は光条件が変化するため **HDR Merge** を強く推奨
- Depth は遠距離でノイズが増えるため **Spatial + Hole Filling** が効果的
- bag ファイルは巨大 (1分で 500MB~1.5GB)
- 外付け SSD (USB3.1/3.2) に保存すること
- 保存ボリュームが不足した場合の対策として 10分ごとの自動セグメント切り替えが有効

---

## 10. まとめ (あなたの設定は正しいか?)

あなたの現在の設定 : - ✓ Decimation - ✓ HDR Merge - ✓ Depth to Disparity - ✓ Spatial - ✓ Hole Filling - ✓ Disparity to Depth - ✓ Min=1m, Max=16m - ✓ 解像度 640×480

→ 研究用途として非常に良い。ほぼ最適構成です。

特に、屋外歩行データ、夜間/逆光への耐性、ノイズ低減など配置目的に合っています。

---

必要であれば、- Python で bag ファイルから RGB/Depth を画像化するツール - 深度を距離マップとして可視化するツール - YOLO 学習用に画像+深度を同期抽出するスクリプトも作成します。