

# ステレオ画像における オクルージョン領域の検出および深度推定

## Occlusion Area Detection and Depth Estimation of Stereo Images

情報工学専攻 関根 匠海  
Information and System Engineering Takumi SEKINE

### 概要

左画像および右画像の各画素における視差の算出に DP ステレオ照合を用いた深度画像には、特にオクルージョン領域において水平線上バースト誤りが頻繁に混入する。深度画像の品質を向上させるためには、オクルージョン領域を検出し、該当領域の深度を再推定することが鍵である。そこで本研究は、DP ステレオ照合において脱同期的に連鎖する照合誤りに着目した、オクルージョン領域検出の新手法を提案する。その結果、提案手法は既存手法と同程度の正確度を維持しながら、約 1.9 倍高速にオクルージョン領域を検出できた。

さらに本研究は、ステレオ内視鏡から得られる映像入力に対して、入力の左右分割や歪み補正等の前処理の後、DP ステレオ照合によって深度画像を生成し、オクルージョン処理、フィルタ処理を経て最終的に得られた深度画像を出力し、深度画像とテクスチャ画像から 3D オブジェクトを生成し、Sony 社製の空間再現ディスプレイへ 3D オブジェクトをストリーム出力するシステムを開発した。本システムは平均プログラム実行時間 193.8 [ms]、平均遅延時間 634.2 [ms] にて動作する。特に、本システムの平均プログラム実行時間のうちオクルージョン処理が占める割合は約 2.2% であり、本研究は本システムに実装された提案手法が高速に動作することを確認した。

キーワード：ステレオ内視鏡, DP ステレオ照合, オクルージョン, 裸眼立体視, 深度画像。

## 1 序論

ステレオ内視鏡を AR/VR 空間への現実世界入力デバイスとして捉えたシステム、すなわち、3次元内視鏡映像化システムの開発は、次世代型医療のもつ未来像の一つである。スマート医療に期待される AR/VR は、コンパクトなセンサーとコンピュータとを組み合わせることで 3次元形状を実時間かつ正確に取り込めることを要求する。3次元内視鏡映像化システムは現在のところ、この要件を満たす唯一の装置である [1]。

ステレオ内視鏡の映像入力において、左画像および右画像の各画素における視差の算出に DP ステレオ照合を用いた深度画像には、左画像に映っているが右画像には映っていない領域、すなわち、オクルージョン領域において水平線上バースト誤りが頻繁に混入する。よって、深度画像の品質を向上させるためには、オクルージョン領域を検出し、該当領域の深度情報を改善することが鍵となる。そこで本研究は、既存手法 [4] と同程度の正確度を維持しながらより高速に動作する、オクルージョン領域検出の新手法を提案する。

さらに本研究は、提案手法を実装した 3次元内視鏡映像化システムを開発する。本システムは、ステレオ内視鏡 ( $\phi 5.4$  [mm] 3D 硬性内視鏡) からの入力映像をもとに深度画像を生成し、深度画像に対してオクルージョン処理およびフィルタ処理を適用し、得られた深度画像とステレオ内視鏡の左画像から 3D オブジェクトを生成し、3D オブジェクトを Sony 社製の空間再現ディスプレイ (Spatial Reality Display: SR ディスプレイ) へストリーム出力する。本研究は、本システムの遅延時間および実行時間を計測し、特に実行時間のうちオクルージョン処理が占める時間を調査する。

## 2 提案手法によるオクルージョン領域の検出

### 2.1 アルゴリズムのコンセプト

図 1 に示すようなオクルージョン領域に対して DP ステレオ照合をおこなうと、深度画像には水平線上バースト誤りが頻繁に混入する。これは、照合誤りが脱同期的に、または同期的に連鎖することが原因である。本研究は、脱同期的に連鎖した深度値を検出することによって、オクルージョン領域を検出する手法を提案する。

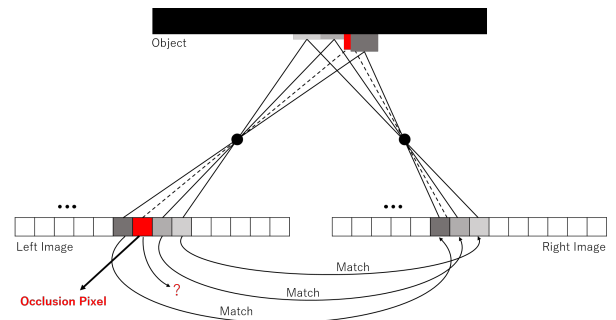


図 1: オクルージョン領域の例

### 2.2 アルゴリズム

提案手法によるオクルージョン領域の検出アルゴリズムを Algorithm 1 に示す。また、アルゴリズムの流れを図 2 に示す。ここで、1 行目の EF はエンハンスフィルタ (Enhance Filter: EF)、9 行目の MF は中央値フィルタ (Median Filter: MF) をあらわす。EF は、立体画像における奥行きをより明確に表すために、深度値の上限と下限を引き延ばすフィルタである。これにより、照合誤りの脱同期的な連鎖を検出しやすくなる。また MF

は、オクルージョン画像のノイズ除去のためにおこなう。ただし、MFのカーネルを3 [px] × 3 [px] とする。9行目で出力される Occlusion は、オクルージョン領域を黒 (true)、非オクルージョン画像を白 (false) にてあらわした二値画像となる。

---

**Algorithm 1:** 提案手法によるオクルージョン領域検出

---

**Input :** Depth Image: Depth, Threshold: TH  
**Output:** Occlusion Image: Occlusion

---

```

1 Depth := EF(Depth);
2 Occlusion := {false}
3 for (y = 0; y < imageHeight; y ++):
4   for (x = 0; x < imageWidth - 1; x ++):
5     if (|Depth(x, y) - Depth(x + 1, y)| ≤
        TH):
6       Occlusion(x, y) := true;
7     else:
8       Occlusion(x, y) := false;
9 Occlusion := MF(Occlusion)

```

---

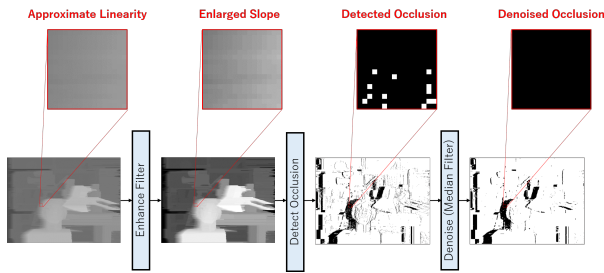


図 2: 提案手法によるオクルージョン領域の検出の流れ

### 3 オクルージョン処理の性能評価

#### 3.1 使用するデータセット

プログラム実験には、縦 480 [px], 横 640 [px] の筑波大学データセット [3] を用いる。

#### 3.2 実験環境

各データセットに対するプログラム実験環境を表 1 に示す。

表 1: 各データセットに対するプログラム実験環境

|            |   |
|------------|---|
| CPU        | Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz |
| RAM        | 16.0 GB   |
| OS         | Windows 10 Pro                                    |
| 言語         | C 言語  |
| コンパイラ      | gcc (tdm64-1) 10.3.0                              |
| コンパイルオプション | -O2 -lpthread                                     |

#### 3.3 オクルージョン領域検出の実行時間

オクルージョン領域検出について、Beak 手法 [4] と提案手法の実行時間を表 2 に示す。ここで Beak 手法は、オクルージョン領域検出のために左画像および右画像を基準とした視差マップを用いる。本研究はこれらの視差マップを得るために、左画像および右画像を基準とした DP ステレオ照合をおこなう。よって、Beak 手法の実行

時間には 2 回分の DP ステレオ照合の実行時間が含まれている。なお、表中の実行時間は、図 3 (a3) および、図 3 (a4) の生成プログラムを 1000 回繰り返し実行した際の値をあらわす。

提案手法は 1 回の DP ステレオ照合から得られる深度画像を入力として用いる。したがって、各手法の実行時間の平均値の差は  $542.2 - (26.5 + 255.7) = 260$  [ms] である。これは、提案手法が Beak 手法よりも  $542.2 / (26.5 + 255.7) \approx 1.92$  倍高速であることを示している。

表 2: オクルージョン領域検出の各手法の実行時間

| 各統計値      | Beak 手法 | 提案手法  | DP ステレオ照合 |
|-----------|---------|-------|-----------|
| 最小値 [ms]  | 405.4   | 18.10 | 180.8     |
| 最大値 [ms]  | 606.2   | 42.72 | 344.9     |
| 中央値 [ms]  | 543.2   | 26.44 | 255.5     |
| 平均値 [ms]  | 542.2   | 26.55 | 255.7     |
| 標準偏差 [ms] | 0.647   | 0.155 | 0.445     |

#### 3.4 検出されたオクルージョン領域の誤謬率

Beak 手法と提案手法のそれぞれにより検出されたオクルージョン領域の誤謬率を表 3 に示す。ここで、オクルージョン検出の誤謬率とは、正解データにおける非オクルージョン領域の画素数のうち、プログラム上ではオクルージョン画素と判定された画素の割合を表す。非オクルージョン検出の誤謬率も同様である。また、各手法により検出したオクルージョン領域を図 3 に示す。

表 3: 各手法におけるオクルージョン検出の誤謬率

| 各誤謬率               | Beak 手法 | 提案手法 |
|--------------------|---------|------|
| オクルージョン検出の誤謬率 (a)  | 0.03    | 0.03 |
| 非オクルージョン検出の誤謬率 (a) | 0.50    | 0.47 |
| オクルージョン検出の誤謬率 (b)  | 0.05    | 0.05 |
| 非オクルージョン検出の誤謬率 (b) | 0.65    | 0.65 |
| オクルージョン検出の誤謬率 (c)  | 0.03    | 0.04 |
| 非オクルージョン検出の誤謬率 (c) | 0.59    | 0.56 |

#### 3.5 深度再推定により生成された深度画像の誤謬率

Jang 手法 [5] による深度再推定を適用した深度画像に対する、正解オクルージョン領域における誤謬率を表 4 に示す。また、各手法により求めたオクルージョン領域を利用した、Jang 手法 [5] による深度再推定を適用した深度画像を図 4 に示す。ただし、この節における誤謬率を、正解深度との深度値の差が 10% 以上の画素の割合とする。

図 4 の (a8), (b8), (c8) のいずれも、オクルージョン領域の深度を再推定することによって、深度の境界情報を改善し、かつ、水平線上バースト誤りを軽減している。

## 4 3次元内視鏡映像化システム

### 4.1 全体像

3次元内視鏡映像化システムの全体像を図 5 に示す。本システムは、内視鏡コントローラ、3D コンバータ、

表 4: 各オクルージョンを用いた深度再推定後の深度画像の正解オクルージョン領域の誤謬率

| 各画像の誤謬率 | DP 深度 | Beak→Jang | 提案手法 →Jang |
|---------|-------|-----------|------------|
| (a)     | 0.462 | 0.387     | 0.384      |
| (b)     | 0.640 | 0.663     | 0.651      |
| (c)     | 0.520 | 0.490     | 0.492      |

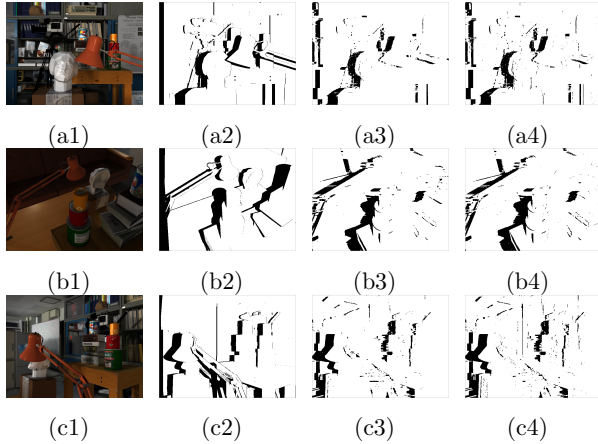


図 3: 左画像および各手法によるオクルージョン. 1 列目: 左画像, 2 列目: 正解オクルージョン, 3 列目: Beak, 4 列目: 提案手法

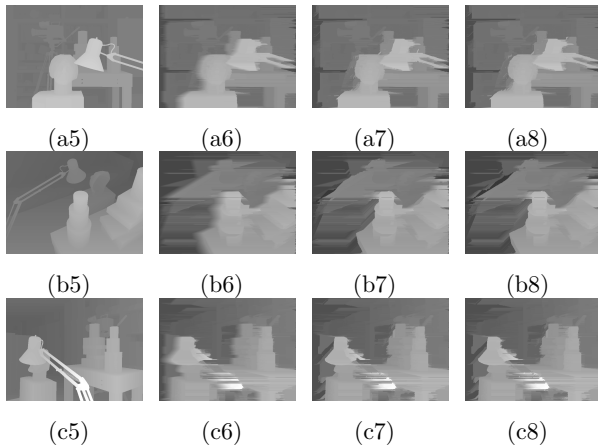


図 4: 各オクルージョンに対する Jang 深度再推定後の深度. 1 列目: 正解深度, 2 列目: DP 深度, 3 列目: Beak→Jang, 4 列目: 提案手法 →Jang.

キャプチャボード, 演算用コンピュータ, SR ディスプレイからなる.

#### 4.2 前処理

ステレオ内視鏡の入力画像を図 6 (a) に示す. これを DP ステレオ照合の入力へと変換するために前処理をおこなう. 前処理は, 入力画像の左右分割, 歪み補正 [6], 高さ調整, 外接楕円のトリミング, 正方形サイズへのリサイズ, 内接正方形のトリミングの順におこなわれる.

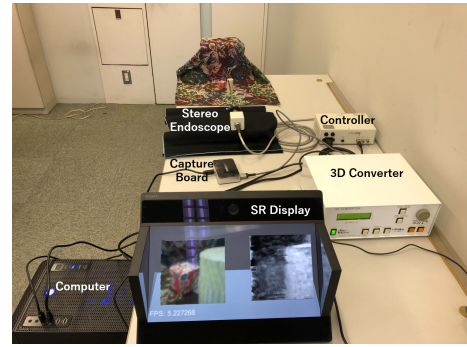


図 5: 3次元内視鏡映像化システム全体像

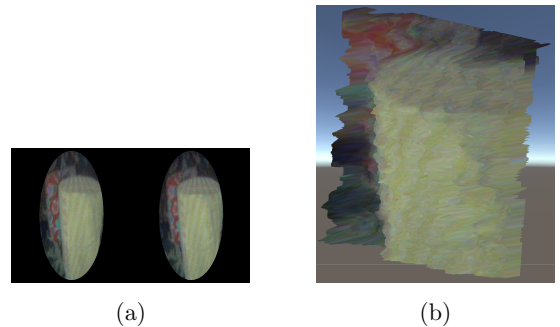


図 6: (a) ステレオ内視鏡の入力 (b) 3D オブジェクト

#### 4.3 深度画像生成

まず, 左画像および右画像を入力として DP ステレオ照合をおこなうことにより深度画像を生成する. つぎに, Algorithm 1 を用いてオクルージョン画像を生成する. つぎに, Jang 手法 [5] によるオクルージョン領域の深度再推定を用いて, オクルージョン領域の深度を再推定する. つぎに, MF, 選択的平均化フィルタ (Selective Average Filter: SAF), EF を順に適用し, 最終的に得られた画像を深度画像として用いる. ただし, 水平線上バースト誤りを軽減するために, カーネルが縦 9 [px] × 横 1 [px] の MF を用いる. また SAF とは, 深度画像の境界情報を保持したまま画像全体に平均値フィルタを適用する手法である.

#### 4.4 3D オブジェクト生成

3D オブジェクトは, 深度画像に対応する奥行きをもつオブジェクトの表面にテクスチャを貼ったものである. 本研究は, 三角形メッシュを用いて 3D オブジェクトを生成する. 生成した 3D オブジェクトを図 6 (b) に示す. これを SR ディスプレイへ表示することにより, 3D オブジェクトの裸眼立体視が可能となる.

### 5 3次元内視鏡映像化システムの性能評価

#### 5.1 実験環境

3次元内視鏡映像化システムのプログラム実験環境を表 5 に示す.

#### 5.2 システムのプログラム実行時間

3次元内視鏡映像化システムのプログラムを 1000 回繰り返した際の実行時間, およびオクルージョン処理の実行時間を表 6 に示す.



表 5: 3次元内視鏡映像化システム実験環境

|        |   |
|--------|---|
| CPU    | AMD Ryzen Threadripper 3970X 32-Core Processor 3.70GHz    |
| GPU    | NVIDIA GeForce RTX 3090 (RAM 24.0GB) × 2 (NVLink SLIにて接続) |
| RAM    | 32.0 GB   |
| OS     | Windows 10 Pro  |
| 言語     | CおよびC#  |
| 物理エンジン | Unity 2020.3.13f1   |

表 6: 3次元内視鏡映像化システムの実行時間

| 各統計値      | 3次元内視鏡映像化システム | オクルージョン処理 |
|-----------|---------------|-----------|
| 最小値 [ms]  | 176           | 5         |
| 最大値 [ms]  | 344           | 18        |
| 中央値 [ms]  | 191           | 5         |
| 平均値 [ms]  | 193.832       | 4.312     |
| 標準偏差 [ms] | 12.107        | 0.565     |

### 5.3 システムの遅延時間

タイマーを映した PC の画面をステレオ内視鏡の撮影範囲内に置き、PC と SR ディスプレイの両方を映した写真をスマートフォンにて撮影する。この状況において、PC に映る時間と SR ディスプレイに映る時間の差分を 3次元内視鏡映像化システムの遅延時間とする。本実験を 100 回おこなった際の各遅延時間を表 7 に示す。

表 7: 3次元内視鏡映像化システムの遅延時間

| 各統計値      | 遅延時間  |
|-----------|-------|
| 最小値 [ms]  | 500   |
| 最大値 [ms]  | 790   |
| 中央値 [ms]  | 630   |
| 平均値 [ms]  | 634.2 |
| 標準偏差 [ms] | 63.56 |

### 5.4 深度およびオクルージョンの出力結果

本システムが生成した深度画像およびオクルージョン画像を図 7 に示す。(a1) - (c1) の左画像に存在するオクルージョンは、(a3) - (c3) のようにおおむね検出できている。さらに、(a4) - (c4) のように、円柱の深度の境界線上に発生する水平線上バースト誤りを部分的に軽減し、深度の境界情報を改善している。

## 6 結論

本研究はまず、エンハンスフィルタ、水平方向の一次微分フィルタ、中央値フィルタの組み合わせによるオクルージョン領域検出の手法を提案した。提案手法は、DP ステレオ照合により生成した深度画像を入力としたとき、Beak 手法と同程度の正確度を維持しながら、より高速にオクルージョン領域を検出できる。提案手法は、DP ステレオ照合において照合誤りが脱同期的に連鎖する領域を特定するために、上記の組み合わせを用いる。オクルージョン領域検出の手法において同様のアプローチをとる先行事例は存在せず、提案手法には独自性がある。

さらに本研究は、高品質な 3D オブジェクト生成のた

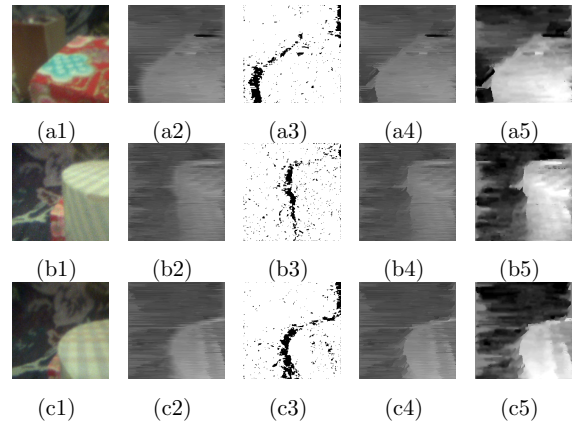


図 7: 各シーンに対する出力結果. 1 列目: L 画像, 2 列目: Depth, 3 列目: Occlusion, 4 列目: ReEstimated-Depth 5 列目: EF(SAF(MF(ReEstimatedDepth)))

めの鍵となるオクルージョン処理を実装した 3次元内視鏡映像化システムを開発した。本研究は、本システムが平均プログラム実行時間 193.8 [ms]、平均遅延時間 634.2 [ms] にて動作することを確認した。特に、平均プログラム実行時間におけるオクルージョン処理の実行時間は約 2.2% である。したがって本研究は、本システムの平均プログラム実行時間に対して提案手法が高速に動作することを示した。

DP ステレオ照合における照合誤りの同期的な連鎖を検出する手法の発見、ステレオ内視鏡の入力映像に点在するカラーノイズの軽減、ならびに、本システムの実行時間および遅延時間の改善が今後の課題である。

## 参考文献

- [1] 鈴木寿, 片井均, “3次元内視鏡映像化システムの開発,” スマート医療テクノロジー ~AI、ビッグデータの活用による次世代手術システムと医療経営~, NTS Inc., 東京, pp.259-268, 2019.
- [2] 伊藤惇, “コンピュータアーキテクチャを考慮した深度推定エンジンの高度チューニング,” 中央大学大学院情報工学専攻修士論文, 2021.
- [3] S.Martull, M.Peris, and K.Fukui, “Realistic CG stereo image dataset with ground truth disparity maps,” *In Trak-Mark*, 2012.
- [4] E.T.Baek and Y.S.Ho, “Occlusion and Error Detection for Stereo Matching and Hole-Filling Using Dynamic Programming,” *Electronic Imaging*, 2016 (2016) pp.1-6.
- [5] W.S.Jang and Y.S.Ho, “Discontinuity preserving disparity estimation with occlusion handling,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 7, pp. 1595-1603, 2014.
- [6] 宇都木修一, 鈴木寿, “消失点を利用した角度の算出に基づくカメラキャリブレーション,” 信学技報, vol. 118, no.459, PRMU2018-132, pp. 99-104, 2019.