

# GPU パイプラインを用いたフィルタリングによる 手メッシュ描画の安定化

○堤 大河, 松林 篤, 牧野 泰才, 篠田 裕之 (東京大学)

## Stabilization of Hand Mesh Visualization via GPU-Accelerated Point Cloud Filtering

○Taiga TSUTSUMI, Atsushi MATSUBAYASHI, Yasutoshi MAKINO,  
and Hiroyuki SHINODA (Univ. of Tokyo)

Abstract: In real-time hand mesh rendering, processing point cloud noise from depth sensors poses a challenge due to the trade-off between robustness and computational cost. To address this challenge, this study proposes a non-iterative processing pipeline that runs on a GPU. The proposed method effectively enhances the quality of the input point cloud by sequentially applying spatial filtering, pose estimation using 3D Principal Component Analysis (3D-PCA), and subsequent temporal filtering based on the estimated pose. Experimental results confirmed that the method reduces outliers and improves the temporal stability of the mesh rendering while maintaining 60 Hz real-time performance. This research demonstrates that improving input quality without iterative calculations is an effective approach for achieving both real-time performance and robustness.

### 1. 緒言

裸眼立体ディスプレイ[1,2]と空中超音波触覚フィードバック[3,4]を組み合わせた非接触インタフェースは、ヒューマン・コンピュータ・インタラクション (HCI) 分野において、その応用が期待され、研究開発が推進されている。この種のシステムにおいて、「矛盾した光学的オクルージョン」が解決すべき重要な課題の一つとして認識されている。これは、ユーザの手が背後の空中像を物理法則に従って遮蔽できず、手が空中像を透過する現象を指す。この課題の解決には、ユーザの手の形状をリアルタイムかつ安定的にメッシュとして再構成し、描画する技術が不可欠である。

本研究では、この安定した手メッシュ描画を実現するための基盤技術を提案する。まず、本研究で対象とするシステム構成を図 1 に示す。複数の深度カメラ (Intel RealSense D415) が特定のインタラクション領域を囲むように配置されており、ユーザはこの領域内に片手を挿入して操作する。この構成では、ユーザの手のみが密な点群として計測されることを想定している。しかし図 2 に示すように、モーションブレンダーやセンサの誤計測に起因する、手の本来の輪郭から離れた位置に点群が外れ値として発生しうる。これらの外れ値は、点群から手の位置および姿勢を推定するトラッキング処理の精度低下を招き、結果として生成されるメッシュに歪みやちらつきを生じさせる要因となる。本研究の目的は、このような外れ値をリアルタイムに除去し、安定したトラッキングを実現することにある。なお、図 1 のように計測対象をインタラクション領域内の片手に限定することで、取得される点群全体の分散が最大となる方向が、ユーザの腕の伸展方向と高い相関を持つという仮定が成り立ち、これが後述する推定アルゴリズムの妥当性を担保する。

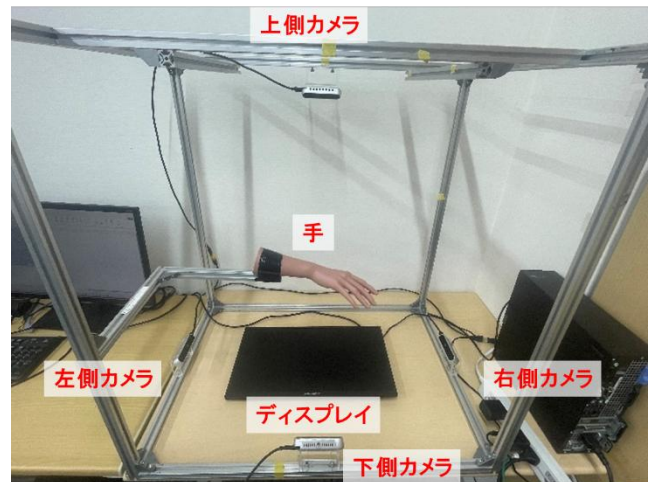


図 1 評価システムのハードウェア構成図

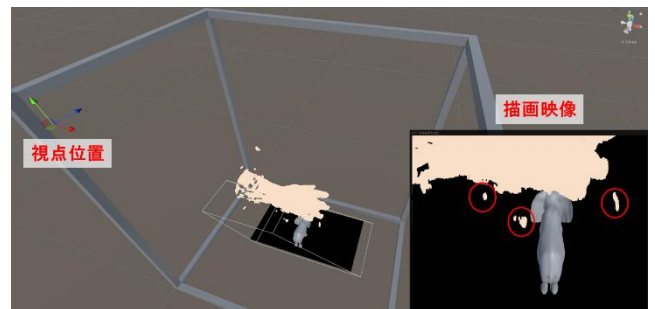


図 2 仮想環境における空中像提示  
描画映像において、赤円の箇所にノイズが生じる。

このトラッキングにおける外れ値問題に対し、既存研究は「頑健性」と「リアルタイム性」のトレードオフに直面してきた。RANSAC[5,6]に代表される反復的手法は外れ値に頑健だが、計算負荷の高さから、60 Hz といった厳格なリアルタイム要件を満たすことが課題となる。一方、Schröder ら[7]が提案した、事前に収集したデータから構築した高次元の姿勢空間モデルに観測点群を当てはめる主成分分析 (Pose-Space PCA) ベ

ースの手法は高速である。一方、性能が事前モデルに依存し、予期せず発生するノイズに対しては、Pose-Space PCA が原理的に有する外れ値への感度という課題があった。

これに対し本研究は、入力データの品質をリアルタイムに改善するという点に着目した、従来とは原理が異なるアプローチを提案する。本アプローチの特徴は、本研究で設計した GPU アクセラレーションを活用した処理パイプラインにあり、これは現実の 3 次元座標空間上で直接行う主成分分析 (3D-PCA) と密に連携して動作する点である。具体的には、前フレームの推定結果を現フレームにおける手の存在領域の予測モデルとして利用し、時間的連続性に基づいて外れ値を除去する。この外れ値が除去された点群のみを、現実の 3 次元座標空間上で直接主成分を算出する、計算負荷の低い 3D-PCA に入力するという一連の処理を統合的に設計した。この事前モデルを必要としない構成により、60 Hz のリアルタイム性を維持しつつ外れ値の影響を抑制し、安定した手メッシュ描画を実現する。

## 2. 関連研究

深度センサから取得した点群に基づくリアルタイム手トラッキングは、HCI 分野における重要な研究課題である。本研究が解決を目指す核心的なトレードオフ、すなわち「外れ値への頑健性」と「リアルタイム性の確保」という二つの要件に関連するアプローチを概観する。

### 2.1. 外れ値への頑健性向上へのアプローチ

点群に含まれる外れ値は、トラッキング精度を著しく低下させる主要因である。この課題に対し、RANSAC[5]に代表される頑健推定アルゴリズムが広く利用されてきた。これらはデータからランダムにサブセットを抽出し、モデルを当てはめるといった試行を繰り返すことで、外れ値の影響を確率的に排除する。近年では、MAGSAC++[6]のような計算効率を改善した手法も提案されている。しかし、これらの手法は仮説検証に基づく反復計算を本質的に伴うため、フレームごとに処理を完結させる必要があるインタラクティブシステム、特に 60 Hz という厳格なリアルタイム要件下では、依然として計算負荷が課題となる。

### 2.2. リアルタイム性確保へのアプローチ

リアルタイム性を最優先する場合、非反復的な手法が有効である。Schröder らは、事前に収集したデータから Pose-Space PCA を用いて低次元の姿勢空間を構築し、計算をその空間内に制約することでリアルタイム性と安定性を両立するモデルベース手法を提案した[7]。このアプローチは計算効率に優れるが、Pose-Space PCA が共分散行列の算出に全てのデータ点を用いるという数論的特性上、入力点群に外れ値が含まれる場合には推定結果がその影響を直接受けてしまう脆弱性を持つ。

## 2.3. 本研究の位置づけ

上述の通り、既存研究は「頑健性」と「リアルタイム性」のいずれかに重点を置いており、両立は困難であった。これに対し本研究では、推定アルゴリズム自体 (出力側) で頑健性を担保するのではなく、その前段 (入力側) でデータ品質を改善するという新しいアプローチを採用する。具体的には、GPU 上で完結する高速なフィルタリングパイプラインを 3D-PCA の前段に配置することで、反復計算を伴う手法を回避しつつ、外れ値の影響を抑制する。この構成により、「外れ値への頑健性」と「リアルタイム性の確保」という二律背反の課題の両立を図り、安定した手メッシュ描画を実現する。

## 3. 提案手法

### 3.1. システム概要

本研究で提案するリアルタイム直線の特徴推定の全体構成を図 3 に示す。本システムは、入力された生点群データから、後段の直線特徴推定に適した加工済みデータを生成する一連の処理パイプラインとして設計される。システムの中核を成すのは、Unity の Compute Shader を用いて GPU 上で実行されるフィルタリングパイプラインである。このパイプラインは、3 次元座標空間上で直接行う 3D-PCA へ入力する点群からノイズおよび外れ値を除去し、後段の CPU 上で実行される 3D-PCA による推定結果のフレーム間変動を抑制する役割を担う。さらに、3D-PCA による推定結果は次フレームのフィルタリング処理へフィードバックされ、時間的連続性を利用したノイズ除去を可能にする。この GPU と CPU が連携するアーキテクチャにより、システム全体として 60 Hz のリアルタイム性を維持しつつ、手メッシュ描画の安定性を確保する。

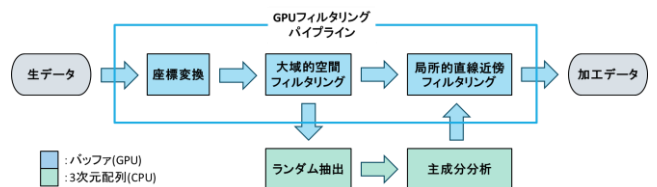


図 3 提案手法の処理パイプライン全体図

図中の青枠は GPU 上で実行される処理、緑枠は CPU 上で実行される処理を示す。

### 3.2. 設計方針：リアルタイム性の確保

非接触インタラクションにおいては、ユーザの操作に対する即時的なフィードバックが、体験の質に直結する。この要請に基づき、本研究では処理パイプライン全体が 60 Hz (フレームあたり約 16.67 ms) 以内で安定して動作することを設計目標とする。このリアルタイム性を確保するため、計算コストの高い反復的最適化手法は避け、非反復的な行列演算によって完結する 3D-PCA を直線特徴推定の中核アルゴリズムとして採用する。一方で、3D-PCA が本質的に有する外れ値への感度という課題は、その前段に GPU アクセラレーシ

ョンを活用した多段階フィルタリング処理を配置することで解決する。

### 3.3. GPU アクセラレーションによる点群フィルタリング

直線の特徴推定の精度は、入力される点群に含まれるノイズおよび外れ値の有無に左右される。この点を踏まえ本手法では、GPU の並列処理能力を利用して複数のフィルタリング処理を実行するパイプラインを、Unity の Compute Shader 上に構築した。本パイプラインは以下の二段階で構成される。

#### 3.3.1. 大域的空間フィルタリング

第一段階では、深度カメラの有効測定距離 (0.3 m ~ 10 m) に基づき、範囲外の点群を除去する。その後、あらかじめ設定した三次元直方体領域外の点群も除去する。この処理により、対象である手以外の点群を削減し、後続処理における計算負荷を低減する。

#### 3.3.2. 局所的直線近傍フィルタリング

第二段階の処理では、手運動がフレーム間で連続的に変化するという時間的一貫性の仮定に基づき、局所的フィルタリングを実施する。本手法では、前フレーム ( $t-1$ ) で主成分分析 (3.4 節) により推定された直線モデルを、現フレーム ( $t$ ) の手領域の予測モデルとして用いる。

前フレームにおける点群の重心を  $q_{t-1}$ 、第一主成分ベクトルを  $v_{t-1}$  とすると、直線モデル  $L$  は次式で表される。

$$L(s) = q_{t-1} + sv_{t-1}, \quad s \in R$$

本研究では、この予測モデル  $L$  の周囲に半径  $r$  を閾値とする円筒状の選択領域を設定し (図 4)、その内部に含まれる点群のみを保持する。すなわち、現フレームの点群  $p_t$  がフィルタを通過する条件は、直線  $L$  に対する最短距離が  $r$  以下である場合であり、次式で定式化される。

$$\|(p_t - q_{t-1}) - ((p_t - q_{t-1}) \cdot v_{t-1})v_{t-1}\| \leq r$$

この条件を満たさない点は、センサ誤差や外乱による点群と見なされ除去される。これにより、予測モデルから大きく外れた、センサ誤差や外乱に起因すると考えられる点が除去される。この処理は、後段で生成される手メッシュのフレーム間変動を抑制することに繋がる。

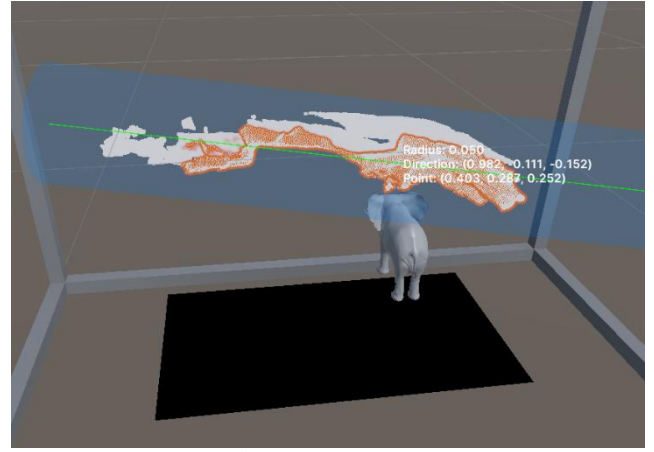


図 4 局所的直線近傍フィルタリングの概念図  
前フレームの直線モデル (緑) を基準に円筒フィルタを適用し、内部の点 (オレンジ) のみを保持する。

### 3.4. 主成分分析に基づく直線の特徴推定

上記のフィルタリング処理を通過した点群に基づき、手の三次元位置および姿勢を表す直線モデルを推定する。前段のフィルタリング後に残存する点群は、図 5 に示すように、推定モデルの近傍に集中して分布する。図 5 は、各点と前フレームで推定された直線モデルとの垂直距離と、その距離に該当する点数の関係を示したものであり、点群の分布が 55 mm 以内に集中していることが確認できる。評価に用いたデータセットにおいて、この範囲を超える外れ値は全点群の平均 0.82 % に過ぎなかった。そのため、後述するランダムサンプリングにおいてこれらの外れ値が抽出される確率は低く、推定結果への影響は限定的であると考えられる。

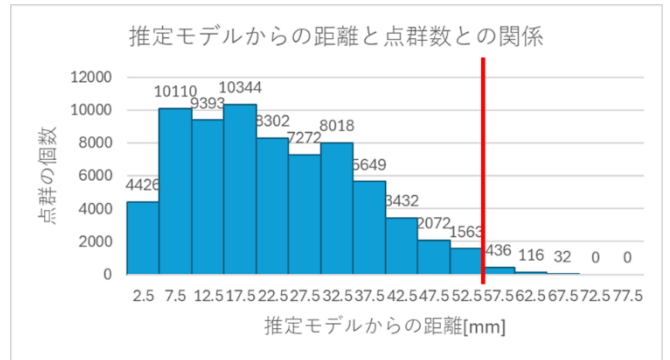


図 5 推定モデルからの距離と点群数との関係  
赤線は、局所的直線近傍フィルタリングで外れ値と判断する距離の閾値(55 mm)を示す。

このフィルタリングの有効性を決定づける半径 55 mm という閾値は、本研究の対象である手全体を確実に内包しつつ、無関係なノイズを効果的に排除するというトレードオフを考慮して設定されている。参考文献[8]によれば若年成人男性の平均手幅は 84.1 mm であるが、この平均値のみを基準に閾値を設定すると、個人差や親指を広げた際の寸法変化に対応できず、手指の一部が領域外となりデータが欠損する可能性がある。トラッキングの安定性を担保する上で、このようなデ

ータの欠損は推定精度を著しく低下させるため、閾値は想定される最大寸法を包含する値である必要がある。すなわちこの閾値は、手指の点群を欠損させないための「下限」としての役割と、背景ノイズを混入させないための「上限」としての役割を同時に担っており、その具体的な値として 55 mm が設定された。

前述のフィルタリング処理によって点群が推定モデルの近傍に集中するという分布特性は、本手法におけるランダムサンプリング戦略の適用を可能にする。すなわち、フィルタリング後の点群から一定の割合（本研究では 1%）の点をランダムに抽出することで、対象となるモデルの形状および姿勢を統計的に代表するサンプルを得ることができる。この戦略により、処理負荷を一定に保ちながら計算コストを抑制でき、60 Hz のリアルタイム処理に対応可能となる。

サンプリングによって得られた点群  $P = \{p_i | i = 1, \dots, N\}$  に対しては、3D-PCA を適用し、手の位置および姿勢を推定する。まず幾何学的中心、すなわち重心  $\bar{p}$  を次式で定義し、手の代表位置とする。

$$\bar{p} = \frac{1}{N} \sum_{i=1}^N p_i$$

次に、各点と重心との差分ベクトルに基づき、点群の空間分布を表す  $3 \times 3$  の共分散行列  $C$  を構築する。

$$C = \frac{1}{N} \sum_{i=1}^N (p_i - \bar{p})(p_i - \bar{p})^T$$

最後に、共分散行列  $C$  の固有値問題を解くことで、点群の主成分方向を求める。得られた固有値のうち最大のものを  $\lambda_1$  とし、対応する固有ベクトル  $v_1$  を第一主成分とする。第一主成分  $v_1$  は点群の分散が最大となる方向、すなわち手の伸展方向を示す。したがって、重心  $\bar{p}$  が手の位置、方向ベクトル  $v_1$  が手の姿勢をそれぞれ表し、これらはフレームごとに逐次算出される。

## 4. 実装

本研究では、提案手法の性能を評価するため、以下の仕様に基づいて実験システムを構築した。

### 4.1. ハードウェア構成

評価システムの全体構成は、緒言に掲載した図 1 の通りである。システムは、幅 740 mm、奥行 540 mm、高さ 760 mm の筐体に収められた実験装置と、これを統合的に制御する PC から構成される。

実験装置は、手トラッキングを行うトラッキングユニットと、空中像を提示する表示ユニットの 2 つのモジュールで構成される。トラッキングユニットには、製造元が提供する近距離での手検出用のプリセットを適用した Intel RealSense D415 を 4 台設置し、各カメラは解像度  $640 \times 480$  pixel で動作する。表示ユニットには、有効表示領域  $230 \text{ mm} \times 410 \text{ mm}$  の QHD 解像度ディスプレイを搭載し、像を提示する。各装置の制御および提

案アルゴリズムの実行には、CPU に Intel Core i9-13900KF、GPU に NVIDIA GeForce RTX 4080 を搭載した PC を用いた。

### 4.2. ソフトウェア構成

本システムは、ゲームエンジン Unity (Version 2022.3.61f1) 上で開発された。深度カメラからのデータ取得、3.3 節で述べた GPU フィルタリングパイプライン、3.4 節での主成分分析まで、提案手法に基づく一連の処理はすべて Unity 上で実装されている。また、空中像の提示についても Unity を用いて行った。

具体的には、まず操作対象となる仮想物体を Unity の三次元空間内に構築し、キャリブレーションにより実空間の座標系と整合させた。次に、ユーザの視点位置に対応する Unity 空間内の特定点に透視投影を行う仮想カメラを配置し、視野角を調整することで物理ディスプレイの四隅と視野境界を一致させた。最終的に、この仮想カメラによるレンダリング映像を物理ディスプレイに出力し、4.1 節で述べた光学系を介してユーザ眼前に空中像として結像させる。本構成の概念図は、緒言に掲載した図 2 の通りである。

## 5. 評価

提案手法の有効性を検証するため、技術的性能評価を実施した。

### 5.1. 実験方法

提案手法の処理性能を定量的に評価するため、比較実験を行った。本実験では、フィルタリング後の点群から全体の 1% をランダムに抽出し、処理に用いた。

比較の公平性を確保するため、まず 4 台の Intel RealSense D415 から、手が静止している状態と空間内で動いている状態の 2 種類のシナリオについて深度データを同時に録画した。次に、同一の録画データを各条件の入力として使用し、システムの初期化完了後、20 秒間 (1200 フレーム) にわたり処理を実行した。

実験条件は、手の状態 (静止、運動)、点群フィルタリングの有無 (有、無)、描画プリミティブ (Sprite、Cube) の 3 つの要素で構成した。これらの条件を組み合わせた各ケースについて平均処理時間を算出し、リアルタイム性 (60 Hz) の要件に対する有効性を評価した。

### 5.2. 実験結果

#### 5.2.1. 処理性能

全 32 条件で計測したシステムの平均処理時間を表 1 および表 2 に示す。

表1 静止状態における平均処理時間と差分

カメラ位置	描画プリミティブ	フィルタあり [ms]	フィルタなし [ms]	差分 [ms]
下側	Cube描画	5.142	5.432	-0.290
	Sprite描画	4.246	4.148	0.098
右側	Cube描画	6.329	6.667	-0.338
	Sprite描画	4.691	4.669	0.022
左側	Cube描画	4.135	4.056	0.079
	Sprite描画	4.195	3.954	0.241
上側	Cube描画	3.972	4.071	-0.099
	Sprite描画	4.026	4.03	-0.004

表2 運動状態における平均処理時間と差分

カメラ位置	描画プリミティブ	フィルタあり [ms]	フィルタなし [ms]	差分 [ms]
下側	Cube描画	4.072	4.117	-0.044
	Sprite描画	4.374	4.132	0.243
右側	Cube描画	5.870	6.107	-0.237
	Sprite描画	5.022	4.814	0.208
左側	Cube描画	4.067	4.293	-0.226
	Sprite描画	4.247	4.222	0.025
上側	Cube描画	5.564	5.089	0.474
	Sprite描画	3.943	4.212	-0.269

全条件において、平均処理時間は目標閾値である16.67 ms (60 Hz) を下回った。計測された平均処理時間は、最大 6.67 ms、最小 3.95 ms であった。最大値は右側カメラで手が静止し、フィルタを無効にした状態で Cube 描画を行った際に、最小値は左側カメラで手が静止し、フィルタを無効にした状態で Sprite 描画を行った際にそれぞれ記録された。フィルタリング処理の有無による影響を見ると、フィルタを有効にした場合、無効時と比較して処理時間が最大で 0.47 ms 増加した。また、描画プリミティブに関しては、Cube 描画が Sprite 描画より平均で 0.63 ms 長い処理時間を要した。

### 5.2.2. 点群フィルタリング効果

点群フィルタリングの有効性を評価するため、各条件下でフィルタにより除去された点群数、フィルタ適用後に残存した総点群数、および除去比率を算出した。結果を表3に示す。

表3 各条件下における点群フィルタリングの効果

カメラ位置	手指の状態	破棄点群数	残存点群数	破棄比率 [%]
下側	静止状態	1481	51503	2.88
	運動状態	5499	22837	26.56
右側	静止状態	1321	20671	6.39
	運動状態	1008	32253	2.95
左側	静止状態	2062	29837	6.91
	運動状態	477	19948	2.36
上側	静止状態	2490	32235	7.72
	運動状態	538	20839	2.53

除去された点群数が最も多かったのは、下側カメラで手が運動している条件であり、平均 5,499 個の点群

が除去された。これは同条件下の入力点群全体の 26.56 % に相当する。また、この除去数は、同カメラで手が静止している場合と比較して約 3.7 倍であった。

さらに、このフィルタリングが描画品質に与える定性的な効果を視覚的に示す。図6は、特に外れ値が多く観測された手運動中の特定フレームにおいて、フィルタを無効にした場合の描画結果である。手の本来の輪郭から離れた位置に、多数の点群が視覚的ノイズとして描画されていることがわかる。

次に、この図6と同一フレームに本手法のフィルタを適用した結果を図7に示す。図6で見られた外れ値が効果的に除去され、手の形状がより明瞭になっている。この比較から、本手法が定量的に点群を除去するだけでなく、最終的な描画品質と安定性の向上に直接的に寄与していることが視覚的に裏付けられる。

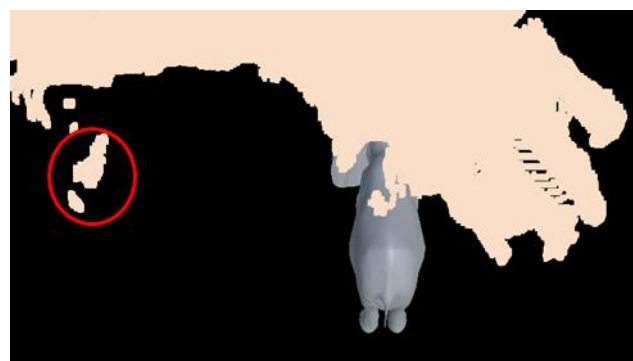


図6 フィルタ無効時の描画結果

赤円の内にノイズが生じる。

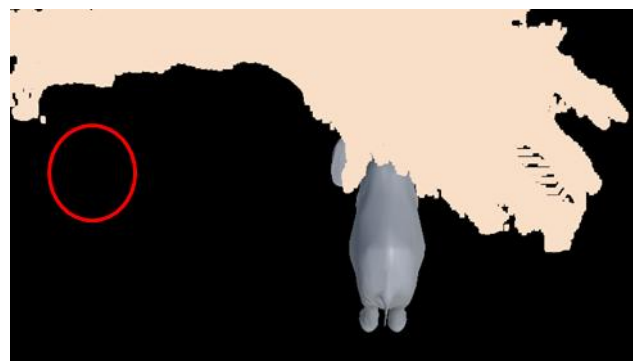


図7 フィルタ有効時の描画結果

赤円の内のノイズが除去されている。

## 6. 考察

### 6.1. 結果の解釈

実験結果は、提案手法が 60 Hz のリアルタイム処理要件を維持しつつ、入力点群に含まれる外れ値を最大で約 26 % 除去可能であることを示した。

処理性能の観点では、システムの平均処理時間は全条件で目標閾値 16.67 ms を下回り、最大でも 6.67 ms であった。この 9 ms 以上の時間的余裕は、将来的な機能拡張、例えばより高度な描画処理や触覚フィードバック計算の統合が可能であることを示唆している。特に、GPU パイプライン上で実行されるフィルタリング処理の追加負荷は最大約 0.47 ms に留まった。この結果は、

GPU の並列処理を用いる本手法の設計が、処理全体への時間的影響を 0.47 ms に抑制しつつ、入力点群に対する前処理を組み込むことを可能にした、と解釈できる。

堅牢性の観点では、フィルタリング処理の追加計算負荷を最大 0.47 ms に抑えながら、手運動時に発生する外れ値（入力点群の約 26 %）を除去できることが示された。入力点群からこれらの外れ値を除去する本手法の前処理が、最終的な手メッシュ描画のフレーム間変動を抑制し、安定化に寄与することを示唆している。

さらに、カメラ視点によって取得点群数が変動した事実は、単一視点では手の側面観測やオクルージョンによる情報欠落が生じる可能性を示唆する。本研究で採用した 4 台のカメラ構成は、複数視点からの情報を相互補完することで、単一視点に比べてより完全な点群データを取得する一つの手段である。

## 6.2. 本手法の限界

本研究は、リアルタイム性と堅牢性を両立する有効な手法を提示したが、いくつかの技術的限界も存在する。

第一に、フィルタリング閾値の一般性である。本手法で用いた局所的直線近傍フィルタリングの半径（55 mm）は、成人男性の手を想定して設定されており、実験環境下では有効に機能した。しかし、ユーザの手の大きさや形状が大きく異なる場合や、手を大きく広げた場合と閉じた場合とで、最適な閾値は変動しうる。現状の固定閾値では、このような多様な条件下でフィルタリングが過剰または不十分になる可能性がある。

第二に、非常に高速かつ複雑な動きへの追従性である。本手法は、フレーム間の動きが連続的であるという前提に基づいている。そのため、ユーザが極めて高速に手を振る、あるいは急に握るなど、フレーム間で点群の分布が劇的に変化する非連続的な動きに対しては、前フレームの予測モデルが有効に機能せず、フィルタリングが破綻する可能性がある。

## 6.3. 今後の課題と展望

上記の限界点を克服し、本研究を発展させるため、以下の方向性が考えられる。

まず、「閾値の一般性」という課題に対し、動的な閾値調整メカニズムの導入が有効である。例えば、大域的フィルタリングを通過した点群の分散や数をリアルタイムに評価し、手の開閉度合いに応じてフィルタの半径を適応的に変化させるアプローチが考えられる。これにより、多様なユーザや手の状態に対して、より頑健なフィルタリングが期待できる。

次に、「高速な動きへの追従性」の課題に対しては、機械学習モデルの統合が有望な解決策となりうる。例えば、過去数フレームの運動履歴から次フレームの手の位置と形状を予測するような時系列モデルを導入することで、予測精度を高め、急激な動きに対する追従性を向上させることが可能である。

これらの課題解決を通じて、本研究で確立した基盤技術をさらに発展させ、より多様な状況下においても

安定したインタラクションを提供する、より完成度の高い空中像システムの実現を目指す。

## 7. 結言

本研究では、空中インタラクションにおける手メッシュ描画の安定化を目的とし、その基盤となる手トラッキング手法を提案した。提案手法は、GPU アクセラレーションを活用した二段階の点群フィルタリングと、確率的サンプリングを用いた主成分分析を組み合わせることで、60 Hz のリアルタイム処理を維持しつつ、トラッキング結果のフレーム間変動を抑制する。

本手法は、空中インタラクションにおける「矛盾した光学的オクルージョン」の問題を軽減するための技術的基盤を提供する。今後は、本研究で確立したリアルタイムトラッキング基盤を発展させ、6.2 節で述べた課題、すなわち動的な閾値調整メカニズムの導入や高速な動きへの追従性を高める機械学習モデルの統合といった改良に取り組むことが、次の研究課題となる。

## 謝辞

本研究の一部は CAO-NEDO SIP 23201554-0 の支援を受けている。

## 参考文献

- [1] H. Kakeya, K. Okada, and H. Takahashi, "Time-division quadruplexing parallax barrier with subpixel-based slit control," *ITE Trans. Media Technol. Appl.*, vol. 6, pp. 237–246, 2018.
- [2] H. Kakeya, A. Hayashishita, and M. Ominami, "Autostereoscopic display based on time-multiplexed parallax barrier with adaptive time-division," *J. Soc. Inf. Disp.*, vol. 26, pp. 595–601, 2018.
- [3] T. Iwamoto, M. Tatezono, and H. Shinoda, "Noncontact method for producing tactile sensation using airborne ultrasound," in *Proc. 6th Int. Conf. EuroHaptics*, 2008, pp. 504–513.
- [4] T. Hoshi et al., "Noncontact tactile display based on radiation pressure of airborne ultrasound," *IEEE Trans. Haptics*, vol. 3, no. 3, pp. 155–165, 2010, doi: 10.1109/TOH.2010.4.
- [5] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [6] D. Barath, J. Noskova, and J. Matas, "MAGSAC++, a fast, reliable and accurate robust estimator," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 401–410.
- [7] M. Schröder, J. Maycock, H. Ritter, and M. Botsch, "Real-time hand tracking using synergistic inverse kinematics," in *Proc. ACM SIGGRAPH Symp. Interact. 3D Graph. Games (I3D)*, 2014, pp. 73–82.
- [8] 産業技術総合研究所, "AIST 日本人寸法データベース 1991-92," 1992. [Online]. Available: <https://www.airc.aist.go.jp/dhrt/91-92/>. [Accessed: 22-Sep-2025].