

立体視に適した大規模点群のレンダリング

電気通信大学 ○岡本 大樹, 丹羽 健, 増田 宏

Point-Cloud Rendering Suitable for Stereopsis

The University of Electro-Communications: Hiroki Okamoto, Takeru Niwa, Hiroshi Masuda

The usability of point-clouds will be largely increased when they are displayed in a stereoscopic view as following eye directions. However, it is not easy to render point-cloud at high-quality and high-speed. Because the view directions are constantly changing. Since inexpensive stereoscopic devices have been developed in recent years, the real-time rendering using common PCs are strongly desired. We discuss stereoscopic visualization of point-clouds for realizing high-speed and high-quality rendering.

Key words: Point-Cloud, HMD,

1. はじめに

レーザスキャナによる 3 次元計測は、工場やプラントの施設・設備管理など多くの分野で利用され始めている。また最近では、発電プラントの改修や歴史的な文化遺産の保護・調査に用いられた事例などがある。

3次元レーザスキャナによるスキャニングでは、広範囲の実フィールドを非接触かつ短時間に 3次元座標データ（点群）として取得することが出来る。このような点群は数千万から数億点に及ぶ。レーザスキャナが普及するにつれ、大規模環境を計測した大規模点群は一般的なものとなりつつある。

一方で、近年では安価な立体視デバイスが開発されている。このようなデバイスを用いることで、大規模点群を立体視することができれば、点群に基づいた現況把握や検査業務の支援などに適用できると思われる。さらに、大規模点群の立体視を安価な立体視デバイスと PC で実現できるならば、様々な分野での応用が期待できる。

しかしながら、大規模点群の立体視を一般的な PC 環境において高品質かつ効率的に行うことは容易ではない。最近では数億点規模の点群を可視化する商用システムも現れているが、視点の移動に追従できるほどには高速ではない。一方、高速表示のために点を間引くと、手前の点群の間から奥の点群が見えてしまい、表示品質が劣化するという問題が発生する。

そこで、本稿では立体視を前提にして、こうした問題点を改善できる高速かつ高品質なレンダリング手法を検討する。

2. 立体視

立体視の手法として主に、平行法・交差法・余色法がある。ここでは平行法（図 1）について述べる。人間の目は一般的に 6~7cm 離れているため、左右の目に見える映像は微妙に異なっている。この差を視差といい、この 2つの映像を重ね合わせることで立体視が実現される。平行法では、ヘッドマウントディスプレイ（HMD）を用いて、左右の目に違った映像を見せることで擬似的な視差を作ることで立体視を実現している。

立体視が可能な安価な HMD として Oculus VR, Inc. の Oculus Rift（図 2）が知られている。Oculus Rift は一般的な液晶ディスプレイを持ち、魚眼レンズとあらかじめ逆補正された画像を組み合わせることで、広視野角を実現している。さらに、頭の動きに映像が追従するヘッドトラッキングを用いている。

3. 高速で高品質な大規模点群表示

3.1 手法の概要

市販されている点群処理ツールで大規模点群を表示する際、視点移動時は点群を間引くことでリアルタイム性を確保し、視点が固定されたときに点群を高精細に表示する処理が行われている。しかし、このような方法は HMD を用いた立体視では適切ではない。HMD を使用した場合、視線は常に動い

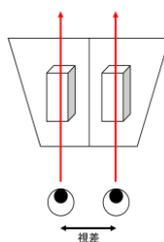


図 1 平行法による立体視



図 2 OculusRift

ていて固定されないため、視点移動中も常に高品質のレンダリングが求められる。また、ユーザにとって不自然に見えるためには、視線方向に追従したリアルタイム性が非常に重要である。

この問題に対応するために、視点位置から可視となる点群を選択し、それらが隙間を生じないように各点の描画サイズを調整することを考える。可視点群の選択では、デプス値だけではなく、視点位置が短距離変化した場合のオクルージョンの変化への対応についても考える。また、描画時の隙間に関しては、隣接する 2 点が連続な面上にあるときには、透視投影画像が連続になるように点の描画サイズを制御する。

3.2 表示対象点群の抽出

ここでは、HMD を装着したユーザは、同じ位置から周囲を見渡す場合を想定する。ただし、立体視では視点は固定ではな

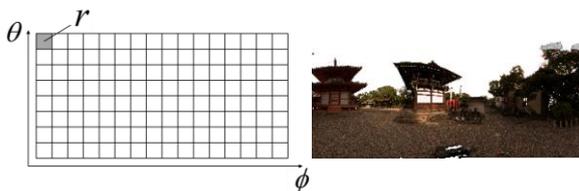


図 3 展開画像

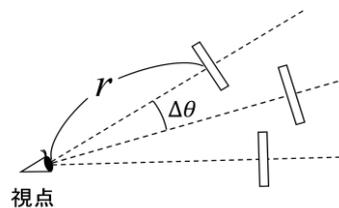


図 4 デプス値 r による選定

く、微小距離だけ動く。

視点から見えていない点は描写する必要がないので、まず、隠れて見えない点を除去する。ここでは、視点付近を原点とした極座標にすべての点座標を変換し、緯度 θ と経度 φ を主軸とする画像上に投影する (図 3)。投影においては、 θ, φ の値を整数値に変換して、ピクセルごとに最前にある点を保持する。(図 4)。

ただし、立体視においては、視差を持った二つの画像を生成する必要があるため、視点は固定されておらず、原点付近で微小距離変動する。そのため、一番手前の点だけを選択した場合においては、ユーザが視線方向を移動させたときに隙間が生じる可能性がある。

この問題を解決するために、図 3 の展開画像上で、隣接する点の距離が不連続となる境界点を検出し、境界付近の点については、背後の点も保持するようにする (図 5)。

境界点の検出においては、図 6 に示すように、隣り合うピクセル上の点が同一面上にあるときは、その距離が $r\theta$ に近くなることを利用する。ここでは、あるピクセルの 4 近傍の点のすべてで、点間距離が $3r\theta$ 以上となるとときに境界点とした。

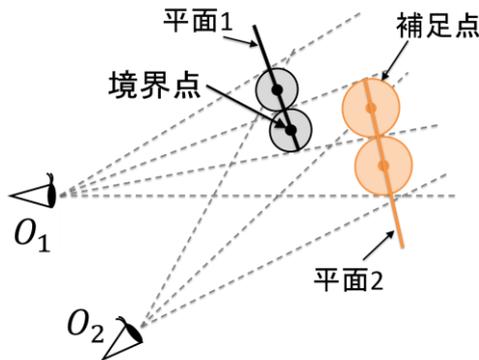


図 5 平面境界点の補足

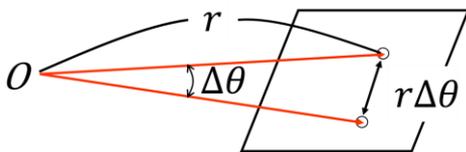


図 6 同一平面の判定

3.2 高品質表示

各ピクセルに点を一つだけ配置することで、レンダリングする点の個数を減らすことができる。しかしながら、レーザー光源からの距離が大きくなるに従って点群密度が小さくなるため、視線位置によっては、点群が投影されるピクセルに隙間が生じる可能性がある。また、立体視を前提として考えると、ユーザの視線移動に伴い、不連続な面の境界付近で隙間が生じる可能性がある。これらは大規模点群を検査などに使用する際に、実フィールドの詳細な情報の欠如に繋がる。

ある連続な面上の 2 点を考える。視点がレーザー光源にあるときは、2 点が隣接するピクセルに投影され、隙間なく表示される。しかし、視点位置が変わったときは、離れたピクセル上に投影され、ピクセル間に隙間が生じる可能性がある。これは、三次元上での点間の相対距離は変わらないが、点を描写する際に二次元投影面に射影されることで、視点の位置により点間の絶対距離が変化するためである。

この隙間を埋めるために、図 7 のように点間の距離 d を求め、この距離 d に基づきポイントサイズを計算する。これにより 2 点が常に接するようになるため、視点がどの位置であっても隙間が生じない表示が実現できる。

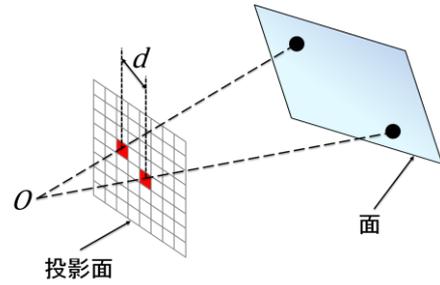


図 7 投影面に射影された点間距離

3.3 リアルタイム性の検証

展開画像の解像度を 720*360, 1040*540, 1440*720, 2160*1080, 2880*1440 に設定し、レンダリング速度を検証した。以下に解像度とフレームレートの関係 (図 8, 表 1) と実際のレンダリング画像 (図 9) を示す。本手法の検証に用いた PC は、実装 RAM : 16GB, プロセッサ : Intel Core i5-4440 CPU @3.10GHz の 64bitPC である。

HMD のリフレッシュレートが 60Hz であるので、60fps 以上の描画速度が望ましい。解像度を 2880*1440 とすると図 12 のように高解像度となるが、fps は 3.45 と大幅に低下した。本研究の環境では、適切な解像度は、1440*720 となった。

表 1 各解像度における fps

解像度	720 *360	1080 *540	1440 *720	2160 *1080	2880 *1440
fps	437.39	182.28	65.95	6.14	3.45

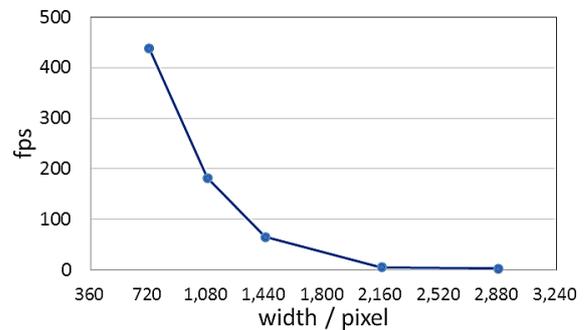


図 8 各解像度における fps



(a) 1440*720

(b) 2880*1440

図 9 手法適用後の点群

4. まとめ

立体視を前提とした大規模点群の高速かつ高品質なレンダリング手法について提案した。本手法を用いて、大規模点群を高速かつ高品質にレンダリングすることができた。

今後は、より高速かつ高品質なレンダリングが実現できるようプログラムを再構築する必要がある。また HMD に本手法を実装し、立体視を可能にする予定である。