Point-cloud rendering systems based on stereoscopic devices are useful for recognizing the current situation of factories. However, in existing methods, details of point-clouds have to be omitted to achieve a sufficient rendering speed. In this research, we develop an efficient point rendering method for supporting maintenance tasks. In our method, we adjust the resolution of point-clouds to the one of stereoscopic devices for achieving high rendering performance. In addition, our system allows users to optionally increase the resolution at the user-specified points to investigate the details. We implemented our stereoscopic rendering method and evaluated the performance of our method.

Key words: Point-cloud, point-based rendering, stereoscopic rendering, head mounted display

1. 結言

近年、老朽化が進んだインフラ設備や生産設備などの保全作業の効率化が重要な課題となっている。保全管理作業を効率化するため、これまでに、補修箇所の自動検出などの手法が検討されてきた。一方で現在の把握や検査においては、熟練作業員による目視作業も不可欠である。しかし、現場の詳細な3次元情報が取得できず、立体視デバイスを用いて、これまで現場で行っていた現

状態の把握や検査が、現場へと行かないとなるためである。保全作業の飛躍的な効率化が期待できる。

近年では、安定した立体視デバイスが開発されており、仮想環境の立体視が身近なものになっている。また、地上型レーザスキャ

ナ(TLS)を用いれば、設備の高密度点群を短時間で得ることができる。そこで、本研究では、大規模な高密度点群を用いて、立体視

デバイスによる保全点検作業を支援する手法について考える。

仮想環境上での活動の点検を行うにあたっては、高密度な点群が必要である。高い密度な点群のデータ量が大きく、描画のコ

ストが非常に多い。一方で高密度な描画を行うために点群を間引くと、点検を行うために十分な情報を得られなくなるという問題があ

る。これまで、我々は、立体視デバイスの解像度に合わせて点群密度を調整することで、多様な点群立体視を実現する手法を提案した[1]。しかし、仮想空間上での移動に制限があり、また、点検業務に必要な詳細情報が行えないような問題があった。

そこで本研究では、これまでの手法を拡張し、高品質な大規模点群の描画と、ユーザが注視箇所をインスタラティブに詳細化できる手法について検討する。なお、本研究では、立体視デバイスとしてOculus Rift CV1を対象とするOculus Touch(図1)を用いて、提案手法の実装と評価を行う。

2. 検查業務に適した点群立体視

2.1 検査業務で要求される描画機能

点検等の作業を考慮した場合、目視検査は現状把握に有効な手段である。そのためには、立体視デバイスで仮想空間内を観察して全体を把握すること、注目箇所をより詳細に観察できることが求められる。

そこで本研究では、立体視デバイスの解像度に合わせた「大規模的描画」、注視点を描画可能な最高密度で描画する「詳細描画」、

その二つを実現することを考慮。大規

模的描画においては、立体視

デバイスのコントローラで移動点群を指定すると、ユーザの視点が

その位置に移動し、描画が更新される。また、詳細描画においては、ユーザが仮想空間中に注視範囲を指定すると、その範囲の点

群を詳細化して描画する。詳細化した部分では、ユーザがその部

分に近づいてみると、細部まで再現された描画が得られる。

2.2 大規模な点群描画

大規模な点群は描画コストが多く、全てを描画するとフレームレートが低下し、浸入感やリアルタイム性を損なう。

一方で、立

体視デバイスには解像度の上限があり、Oculus Rift CV1の場合は

片目あたり1080x1200ピクセル以上の点を描画する必要がある。

また、ノスタフレームポイントの描画は可能であるが、そのた

め、高密度な点群は、立体視デバイスの解像度に合わせて間引く

ことができ、また、不可視領域の点群は、描画対象から除外する

ことができる。

本研究では以下の手順で大規模な点群描画を行う。

まず、国本らの手法[1]で、図2に示すようなデスモップを作

成する。デスモップとは、視点に最も近い点のみを抽出した可

視点の集合のことをとり、その解像度は、立体視デバイスの解像

度に合わせて設定される。ここでは、横軸を方位角、縦軸を仰角

とした二次元座標上に点群を投影し、同一のピクセル数が複数の点

が存在する場合は、視点からの距離が最も近いものを選択する。

この際、処理の高速化のために、視点から十分離れた点でスキ

ャン化された点群はデスモップ作成の処理から除外する。また、

立体視デバイスから視線方向の情報を取り、視野の外にある

点群を描画対象から除外する。視線の極座標ベクトルがわかる、

視野の範囲を計算することで計算が可能である。図3において、実際

に描画を行った様子を示す。

ユーザが指定した地点に高速に移動するために、前処理として、

DTM (Digital Terrain Model)を作成しておく。地表上に格子を生

成し、格子に含まれる点群で座標が最も小さい値を探求する。移

動地点は、指定した地点の3次元座標を地表面とする。
3. 評価実験・考察

本研究では、ユーザがコントローラーで注視範囲を指定することで、その範囲を詳細化する。ここでは、被験者のすべての点群を読込込みとした。本手法では、選択区の半径が小さいときは 90 fps 以上の描画性能が実現できるが、半径が大きくなるにつれて描画すべき点の個数が大きくなるため、描画性能が低下していくことを確認した。

そこで、描画処理をする点群数と fps との関係を検証した。立体視デバイスは Oculus Rift CV1、コントローラには Oculus Touch を用いた。実行環境は、CPU: Intel Core i7-6700K、GPU: GeForce GTX 1080、RAM: 64.0 GB で、使用した言語は C++である。

描画すべき点の個数を徐々に増加させていく fps を測定した結果を図 6 に示す。我々の検証実験では、描画点群数が 2000 万点程度までは、Oculus Rift CV1 の最高値である 90 fps が実現でき、その範囲を超えると fps は急激に低下した。

このことから、詳細点群を描画する場合には、被描画点数が上限である 2000 万点近辺を超えないように設定する必要があることが明らかになった。すなわち、上限の点数を $N_{\text{max}}$ 、大局的描画の点数を $N$ とするとき、詳細点群を $N_{\text{max}} - N$ 個以下まで間引き必要がある。なお、本手法では、間引き事によって不自然なパタークがでないように、乱数を用いて間引きしている。

4. まとめ

リアルタイム性を損ねない大規模点群の高速描画を実現した。また、選択された範囲の点群を高速に抽出し、詳細に描画する手法を示した。また、本手法を実装して評価を行い、高速描画できることは明らかになった。

今後は、詳細化範囲の自動抽出や、階層的な点群の詳細化手法を検討していく。また、点群のセグメントーションを行い、距離に応じて点の描画サイズを変えていく手法についても検討する。

参考文献
[1] 鳥本大樹，丹羽健，増田宏：立体視に適した大規模点群のレンダリング（第 2 報），精密工学会講演会 2015