

特別研究報告

題目

実空間確率場表現を利用した ネットワーク型仮想現実サービスのための レンダリング手法の提案と評価

指導教員

村田 正幸 教授

報告者

内田 悠斗

2021年2月9日

大阪大学 基礎工学部 情報科学科

実空間確率場表現を利用したネットワーク型仮想現実サービスのためのレンダリング手法の提案と
評価

内田 悠斗

内容梗概

近年、VR (Virtual Reality: 仮想現実) や MR (Mixed Reality: 複合現実) をはじめとした XR (X-Reality) は、その機器の低価格化および品質の向上を背景に、急速に普及が進んでいる。それに伴ってネットワークに流れるメディアとして、画像や動画などの2Dのメディアに加え、3Dのメディアが登場し、その割合は日々増加し、今後も3Dメディアを用いた様々なアプリケーション・サービスが登場するものと考えられる。

本報告では、遠隔地の情報を点群データとして取得し、ユーザが3Dオブジェクトとして実在の物体を体感できるネットワークサービスに着目し、点群データのVR機器への情報表現に取り組む。しかし、高解像度の点群データは、そのデータサイズが大きいためネットワークを介して送受信することは非現実的である。そこで、少量かつ低解像度の点群データをネットワークを介して送受信しつつ、低解像度の点群データをVR機器で描画する際に生じる3Dオブジェクトの凹凸を軽減するレンダリング手法を用いて3Dオブジェクトの補正を試みる。ただし、レンダリング手法による点群データの補正表示をVR上の空間全体に施すと補正処理による負荷が大きくなる。そのため、本報告では処理負荷を軽減するために、実空間の確率場情報を利用して一部の3Dオブジェクトの補正を行うレンダリング手法を提案する。確率場情報は、点群の各点に対して識別ラベルと確度、物体インスタンス情報が付与された情報であり、機械学習等の適用により得られる情報である。そのため、点群の確度が小さい場合、識別に必要な情報が不足しているものと考えられる。提案するレンダリング手法では、確率場情報の性質にもとづき、確度が小さい3Dオブジェクトに限定した補正表示を実施し、視認性を高めている。提案レンダリング手法を実装し、VR機器を用いて3Dオブジェクトを描画した結果、23.7%の点群に対してレンダリング手法を適用することで、視認性が向上することを確認した。また、その時の処理遅延時間は、すべての点群に対してレンダリング手法を適用する場合と比べて、84.1%~94.8%削減されることがわかった。

主な用語

仮想現実 (VR: Virtual Reality)、点群 (Point Cloud)、確率場 (Random Field)、インスタンス (Instance)、レンダリング (Rendering)

目次

1	はじめに	5
2	ネットワーク型仮想現実サービス	8
2.1	仮想現実	8
2.2	想定するサービス	8
2.3	点群データの既存レンダリング手法	9
3	実空間確率場表現に基づくレンダリング手法	12
3.1	確率場情報の利用	12
3.2	評価	17
3.2.1	実行環境	17
3.2.2	評価方法	18
3.2.3	評価結果	18
4	おわりに	25
	謝辞	26
	参考文献	27

目 次

1	確率場情報	6
2	提供サービスの概要	9
3	クラウドを用いたサービスシステム構成	10
4	エッジサーバを用いたサービスシステム構成	10
5	未補正レンダリングの描画例	11
6	補正レンダリングの描画例	11
7	描画する放物面	13
8	確度 0.9 以上の物体の未補正レンダリング例	15
9	確度 0.9 未満の物体の未補正レンダリング例	16
10	確度 0.9 未満の物体の補正レンダリング例	16
11	VR 装置 (Oculus Rift S)	17
12	未補正レンダリングの実行結果: 間引き無し点群データ ($r=0$)	20
13	提案レンダリング手法の実行結果: 間引き無し点群データ ($r=0$)	20
14	未補正レンダリングの実行結果: 間引いた点群データ ($r=3$)	21
15	提案レンダリング手法の実行結果: 間引いた点群データ ($r=3$)	21
16	未補正レンダリングの実行結果: 間引いた点群データ ($r=5$)	22
17	提案レンダリング手法の実行結果: 間引いた点群データ ($r=5$)	22
18	未補正による視認性低下の例: 間引いた点群データ ($r=3$)	23
19	提案レンダリング手法による視認性向上の例: 間引いた点群データ ($r=3$)	23
20	処理遅延時間の測定結果	24
21	補正点群数と処理遅延時間の関係	24

表 目 次

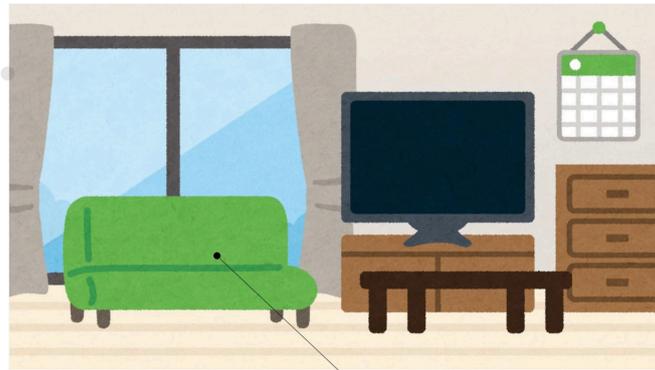
1	ラベル ID とカテゴリの対応	12
2	実行環境	17

1 はじめに

近年、VR (Virtual Reality: 仮想現実) や MR (Mixed Reality: 複合現実) をはじめとした XR (X-Reality) は、その機器の低価格化および品質の向上を背景に、急速に普及が進んでいる [1,2]。それに伴ってネットワークに流れるメディアとして、画像や動画などの 2D のメディアに加え、3D のメディアが登場し、その割合は日々増加している。例えば、VR 機器を用いて目の前にアーティストがいるかのような体験をするアプリケーションが登場している [3]。また、美術館などでは、仮想空間上に展示物を模した 3D オブジェクトを配置し、その 3D マップを VR で見て回ることでできるアプリケーションを提供する試みがなされている [4]。今後も 3D メディアを用いた様々なアプリケーション・サービスが登場するものと考えられる。

現在は、3D オブジェクトを作成する情報データとしてグラフィックデータまたは点群データが用いられている。グラフィックデータを用いて 3D オブジェクトを描画する際には、事前に場 (以降、シーン) を表現するプログラムを組んでおき、プログラムの内容によって指定された色や形の 3D オブジェクトが配置されたシーンを表示する。そのため、3D オブジェクトの細部を作り込むことができる利点がある。しかし、プリプログラムされた表示や動作以外のことができず、シーンごとに新たなプログラムを作成する必要がある。一方、点群データは、物体の座標情報と色情報の集合データであり、RGB-D カメラや 3D レーザースキャナによって採取される。点群データは点の情報の集合であるため、物体の表面の曲線や凹凸を 3D オブジェクトとして表現するためには多くのデータ量が必要である。しかし、点群データを元に 3D オブジェクトを描画する際、点を表示する汎用プログラムを作成することで、データの差し替えのみで新たな 3D オブジェクトを描画できる利点がある。そのため、例えば遠隔ショッピングサービスや遠隔ショールームサービスなど、ユーザが遠隔地の情報をリアルタイムに知覚し遠隔地の人やモノとインタラクトするサービスでは点群データを用いることが適していると考えられる。そこで本報告では、遠隔地の情報を点群データとして取得し、ユーザが 3D オブジェクトとして実在の物体を体感できるネットワークサービスに着目し、点群データの VR 機器への情報表現に取り組む。具体的には、現在画像で行われているネットショッピングの商品表示を 3D データとして VR 機器上に表示しあらゆる角度から自由に見ることができる遠隔ショッピングサービスや、実際にその場に赴かなければ見ることのできないショールームを点群データ化して VR 機器上に表示する遠隔ショールームサービスを想定している。

しかし、通常 3D オブジェクトの視認性を高めるためには、多くの点群データ量が必要となる。例えば、狛犬の 3D オブジェクトでは約 60MB、山や湖を含む風景の 3D オブジェクトには約 300MB の点群データ量が必要になる。そのため、高解像度の点群データをネットワークを介して送受信することは非現実的である。そこで本報告では、少量かつ低解像度の点群データをネットワークを介して送受信し、低解像度の点群データを VR 機器で描画する際に生じる 3D オブジェクトの凹凸を軽減するレンダリング手法を検討する。レンダリングとは、点群データを 3D オブジェクトとして PC や VR 機器のディスプレイ上に描画することである。既存のレンダリングでは、点群の各点を円として



座標 + 色 + ラベル + 確度 + 物体インスタンス

図 1: 確率場情報

描画（以降、未補正レンダリング）している。しかし、円として描画するとレンダリングされた 3D オブジェクトの細部や輪郭の視認性が低下することから、点群の各点を放物面として描画する手法や、ユーザから見た点群の立体的な位置関係を計算し、それを考慮した放物面として点群の各点を描画し、これによって 3D オブジェクトの補正表現を行う（以下、補正レンダリング）手法が提案されている [9]。ただし、既存のレンダリング手法で補正レンダリングを用いる場合、その対象は描画する 3D オブジェクトすべてであるため、処理負荷が大きくなる。処理負荷の増大は FPS（Frame Per Second：フレームレート）の低下を招き、仮想現実技術で求められる物理法則のための演算や実空間へのフィードバックを実装するための資源の余地がなくなる。FPS の保持はユーザに臨場感を与える効果があり [5]、処理負荷の増大はユーザの満足度を低下させる一因といえる。よって、ユーザが知覚する 3D オブジェクトの視認性を高めつつ、処理負荷を軽減するために、補正レンダリングを適用する範囲を必要な点群のみに限定する必要がある。補正レンダリングが必要とされる点群は、人間が物体を識別する際その細部や輪郭の情報が重要となるものや、欠落があり、それをもとにレンダリングされる 3D オブジェクトの識別が困難なものが挙げられる。例えば、ドアを表す 3D オブジェクトのうちドアノブを表す部分が不鮮明なとき、壁と誤認識してしまうことが考えられる。

本報告では、補正レンダリングが必要な 3D オブジェクトを特定する手段として、点群データに対して機械学習を実行することにより算出された確率場（Random Field）情報を使用することを提案する。本報告で使用する確率場情報は、物体インスタンス情報、およびその物体があらかじめ用意されたカテゴリのうちどのカテゴリに属しているのかを表すラベル情報に加え、ラベルに対する確度情報である。これらの情報が図 1 のように、点群の各点に対して座標情報、色情報などの点群情報とともに各点に付与されている。物体インスタンス情報とは点群の各点がどの物体の構成要素であることを意味しているのかを表す情報であり、確度は 0 から 1 で表され、1 に近づくほどラベルのカテゴリの物体である確率が高いことを意味している。確度が高いものは機械学習によるカテゴリの識

別が可能である程度の多い情報量を持つ点群であるといえるのに対して、確度の低い物体は点群の持つ情報量が少なく機械学習によって他のカテゴリとの判別がつきにくい点群である。したがって、機械学習の識別能力が人間の識別能力に近いものとした場合、物体インスタンスのラベルに対する確度が低い点群ほど、補正レンダリングが必要な点群であるといえる。

以上から、本報告では確率場情報が付与された点群のみに限定し補正レンダリングを適用するレンダリング手法を提案し、更に提案レンダリング手法を用いることで未補正レンダリングのみを用いるレンダリング手法より視認性が向上すること、補正レンダリングのみを用いるレンダリング手法より処理負荷が削減されることを確認する。

2 ネットワーク型仮想現実サービス

本章では、本報告で想定するネットワークを利用した仮想現実サービスについてと利用する既存レンダリング手法について述べる。まず、2.1節で仮想現実について説明する。次に、2.2節では想定するサービスについて説明し、2.3節では実装したアプリケーションに使用した既存レンダリング手法について説明する。

2.1 仮想現実

近年、VR (Virtual Reality: 仮想現実) や MR (Mixed Reality: 複合現実) をはじめとした XR (X-Reality) はその機器の低価格化および品質の向上 [1, 2] から急激に普及している。XR の登場によって、コンピュータ上に仮想的に作られた世界をあたかも現実の世界のように体感することができるようになりつつある。例えば、本報告で扱う仮想現実技術では、3D モデル (コンピュータ上に作成したグラフィックス) を用いて構築された仮想現実上で実空間 (現実世界) と変わらない体験することができるようになることを目標として研究が進められている。これを実現するために、3D モデルの視認性を向上させるだけでなく、実空間上で作用している物理法則を仮想空間上で再現できるようにすることや、仮想空間上で発生した事象による作用を実空間にいる我々が五感で感じられるようにすることが求められる。また、現在仮想現実技術を用いたサービスやアプリケーションは静的なデータを扱うことが主流である。これをネットワークを通し逐次更新される、動的なデータを扱うことも展望として挙げることができる。

3D モデルを作成するためにはコンピュータグラフィックスを作成する方法、または点群データとよばれる実空間を測量したデータを利用し作成する方法が挙げられる。本報告では、RGB-D カメラや 3D レーザースキャナによって測量することができる点群データを扱う。点群データは測量によって得られるため、これを用いて 3D モデルを作成すると測量対象物の寸法や構造を仮想空間上に正確に再現することができる。しかし、そのデータにはしばしばノイズが発生することからそれを除去する研究が進められている [6]。

2.2 想定するサービス

想定している遠隔ショッピング、遠隔ショールームサービスの全体システム構成を図 2 に示す。

クラウドコンピューティングを用いた場合は図 3 のように、エッジコンピューティングを用いた場合は図 4 のようになる。クラウドコンピューティング [7] を用いた場合、クラウドの持つ大規模な演算機能を利用し確率場情報の算出を実行することができる。エッジコンピューティング [8] を用いた場合、点群データを測定した場所で確率場情報を算出することができるため、よりネットワーク遅延が少なくユーザへとデータを届けることができる。

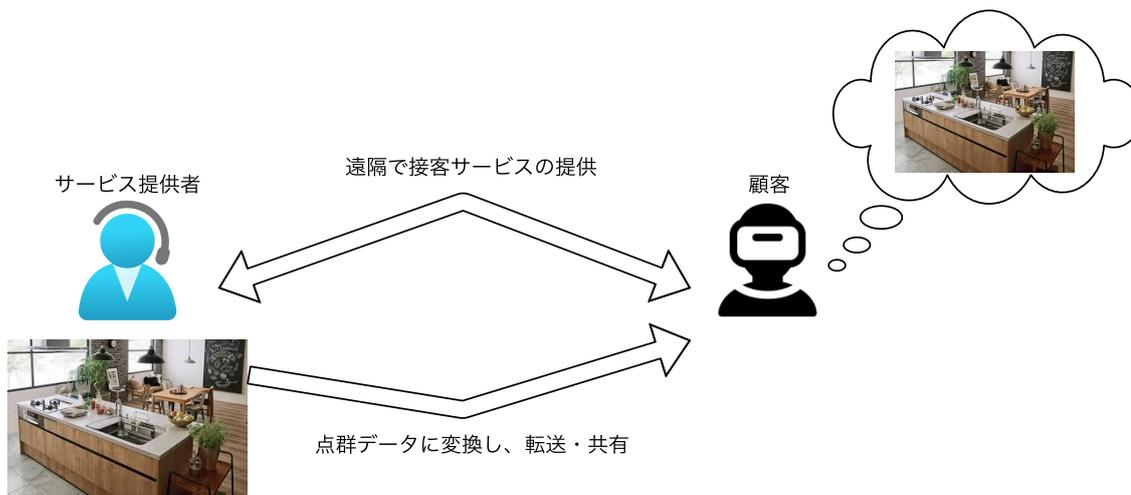


図 2: 提供サービスの概要

本報告では、そのうちの点線で囲われた部分の機能を担うアプリケーションを開発する。このシステムではまず、現実の物体を計測し点群データ化する。それをクラウド、あるいはエッジサーバに保存し確率場情報の算出を行う。次に、確率場情報の付与された点群データをユーザのサーバへとネットワーク経由で転送する。その際、ネットワークの輻輳を避けるために転送する点群データ量を削減する必要があることを想定する。

2.3 点群データの既存レンダリング手法

本報告では、視認性向上のために文献 [9] のレンダリング手法を応用している。文献 [9] のレンダリング手法では点群の各点を円として描画する未補正レンダリング (図 5) と、ユーザから見た点群の立体的な位置関係を計算し、それを考慮した放物面として点群の各点を描画する補正レンダリング (図 7) を用いている。

未補正レンダリングを行った場合、描画された円が互いに重なるため、各点の持つ輪郭や細部の情報が損なわれている。補正レンダリングを行った場合は、点の立体的な位置関係が考慮されているため、レンダリングされた各点を表す放物線が互いに干渉しあうことなく、各点の持つ細部や輪郭の情報を損なうことなく描画されている。しかし、既存レンダリング手法で補正レンダリングを用いる場合、その対象は描画する 3D オブジェクトすべてであるため、処理負荷が大きくなる。処理負荷の増大は FPS (Frame Per Second : フレームレート) の低下を招き、仮想現実技術で求められる物理法則のための演算や実空間へのフィードバックを実装するための資源の余地がなくなる。そこで、処理負荷を軽減するために、補正レンダリングを用いて描画する 3D オブジェクトを限定する機能が求められる。

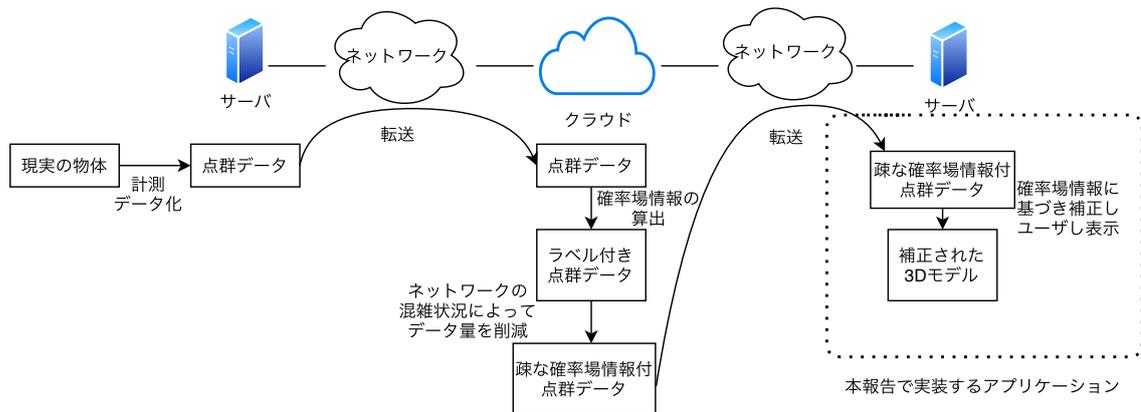


図 3: クラウドを用いたサービスシステム構成

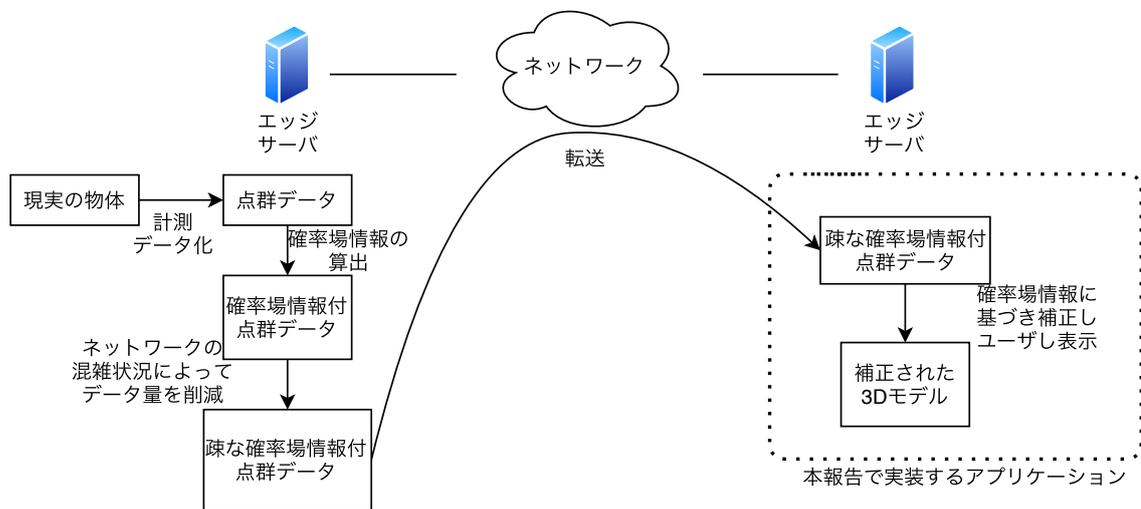


図 4: エッジサーバを用いたサービスシステム構成

また、既存のレンダリング手法では、描画される円や放物面の大きさを手動で決定している。円や放物面大きさが大きすぎると各点の持つ情報の正確性が保てない。逆に、小さすぎると描画された3Dオブジェクト同士の間隙ができてしまう。しかし、点群の密度は一様ではない。よって、点群の密度によって描画される円や放物面の大きさを自動で決定する機能が求められる。



図 5: 未補正レンダリングの描画例



図 6: 補正レンダリングの描画例

表 1: ラベル ID とカテゴリの対応

Label	Category	Label	Category	Label	Category
0	wall	7	door	14	refrigerator
1	floor	8	window	15	shower curtain
2	cabinet	9	bookshelf	16	toilet
3	bed	10	picture	17	sink
4	chair	11	counter	18	bathtub
5	sofa	12	desk	19	other furniture
6	table	13	curtain		

3 実空間確率場表現に基づくレンダリング手法

本章では 2 章で示したレンダリング手法を実空間確率場情報を利用するよう変更した、提案レンダリング手法についてその実装を説明し、評価する。まず、3.1 節にて本報告で利用する確率場情報についてと、それに基づく実装について説明し、3.2 節で評価を行う。

3.1 確率場情報の利用

本報告では 3 次元屋内点群データセットである ScanNet [10] に対し、点群の各点について確率場情報を付与したデータを使用する。確率場情報は、文献 [14] に基づいており、物体インスタンス情報、およびその物体があらかじめ用意されたカテゴリのうちどのカテゴリに属しているのかを表すラベル情報に加え、ラベルに対する確度情報が付与されている。ラベルとカテゴリの対応を表 1 に示す。

物体インスタンス情報とは点群の各点がどの物体の構成要素であることを意味しているのかを表す情報であり、確度は 0 から 1 で表され、1 に近づくほどラベルのカテゴリの物体である確率が高いことを意味している。したがって、機械学習の識別能力が人間の識別能力に近いものとした場合、物体インスタンスのラベルに対する確度が低い点群ほど、補正レンダリングが必要な点群であるといえる。例えば、点と確率場情報として、

3.1542 2.2542 0.7784 34 33 28 2 0.3947 0

が与えられると、座標 $(x, y, z) = (3.1542, 2.2542, 0.7784)$ にある色 $(R, G, B) = (34, 33, 28)$ の点は、カテゴリ 2 の cabinet (戸棚) である確度が 0.3947 の物体 $ID=0$ の物体を構成する点群であることを意味している。

また、本報告ではこの情報を利用することによって、点群の密度によって描画される円や放物面の大きさを自動で決定する機能を実装し視認性を向上させること、補正レンダリングを用いて描画する 3D オブジェクトを限定する機能を実装し処理負荷を低減させることを考える。

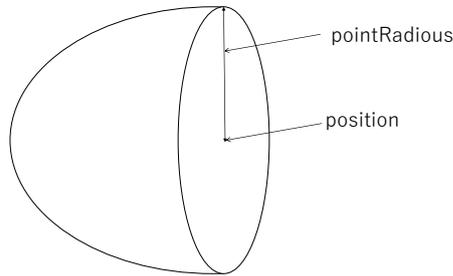


図 7: 描画する放物面

まず、点群データの密度が一様ではないことを考慮して、各物体における円や放物面の描画される大きさを適切なものにするために物体インスタンス情報を利用し、各物体における点群の密度によって、その物体を構成する各点をレンダリングする際のサイズ s を、

$$\frac{4}{3}\pi s^3 = \frac{V}{pts} \quad (1)$$

によって決定する。 pts は物体に所属する点群数、 V は物体を構成する点群の存在範囲の体積である。これによって、各点を半径 s の球とすると、ほぼ隙間なく点群の存在範囲を埋めることが可能である。式 (1) にもとづいて、描画する円または放物面の大きさを変更する関数 `CulPointRadius()` は、

Listing 1: 各点のレンダリングサイズの導出

```
public double CulPointRadius(){
    double volume = tightBoundingBox.xLength() * tightBoundingBox.yLength() *
        tightBoundingBox.zLength();
    double density = (double) volume / points;
    return pointRadius = (float) Math.Pow(density / 4, 1.0 / 3.0);
}
```

であり、戻り値の `pointRadius` の値を点の描画サイズに設定する。変数 `volume` は、物体を構成する点群の存在範囲の体積 V を表す。体積 V は、物体を構成する点群の x 軸方向、 y 軸方向、 z 軸方向のそれぞれの座標情報から最大と最小を求め、その 2 点間の距離により算出する。

次に、補正レンダリングを用いて描画する 3D オブジェクトを限定する機能の実装方法を述べる。まず、すべての点群データを補正レンダリングする場合の疑似コードをアルゴリズム 1 に示す。この疑似コードは文献 [9] に基づいているが、本報告では確率場情報を用いるため入力の点群データセット `Data` は、確率情報を含む (`position, color, label, probability, id`) を要素に持つものとして記述している。まず、`Data` に含まれる各点を、`id` ごとに `List[id]` の点群に分割する。そして、分割した `id` ごとの点群データに対して、先述の `CulPointRadius()` を用いて描画する半径 `pointRadius` を求め、座標 `List[id].position` に色 `List[id].color` で図 7 に示す放物面を点毎に描画する。

Algorithm 1 3D オブジェクトの描画 (すべて補正レンダリングの場合)

Require: *Data* is a list of (*position, color, label, probability, id*), $id = -1, ..$

```
1: for all  $d \in Data$  do
2:    $List[id] \leftarrow d + List[id]$ 
3: end for
4: for  $i = -1$  to  $id$  do
5:    $pointRadious \leftarrow CulPointRadious(List[id])$ 
6:    $drawParaboloids(List[id].position, List[id].color, pointRadious)$ 
7: end for
```

Algorithm 2 確度情報による補正/未補正レンダリングの決定

```
if  $List[id].probability < 0.9$  then
   $drawParaboloids(List[id].position, List[id].color, pointradious)$ 
else
   $drawCircles(List[id].position, List[id].color, pointradious)$ 
end if
```

提案レンダリング手法は、アルゴリズム 1 の 6 行目をアルゴリズム 2 に変更し、物体の確度情報 $List[id].probability$ にもとづいて、放物面として描画するか否かを決定する。

図 8 はラベル 5 (ソファ) である確度が 0.977 の物体の点群を未補正レンダリングした例である。それに対して、図 9 はラベル 2 (戸棚) である確度 0.857 である物体の点群を未補正レンダリングした例である。この 2 つを見比べると、前者はソファであることは読み取れるが、後者は戸棚と判断することは困難であり、特に戸棚に備わっている取っ手の視認性が低いことがわかる。このように、物体の点群の確度が高くなるとともに、人間が物体を識別するのに十分な特徴量を有すると考えられる。本報告では、物体の点群の確度が 0.9 を超えているものは、人間が物体を識別するのに十分な特徴量を持っているとみなし、確度が 0.9 を下回っている物体の点群を補正レンダリングで出力することで処理負荷の低減を図り、さらに物体の点群を人間が識別する補助を行う。図 9 に対して確度が 0.9 未満の物体の点群に対して補正を行った結果が図 10 であり、戸棚であることを人間が識別する補助をする。

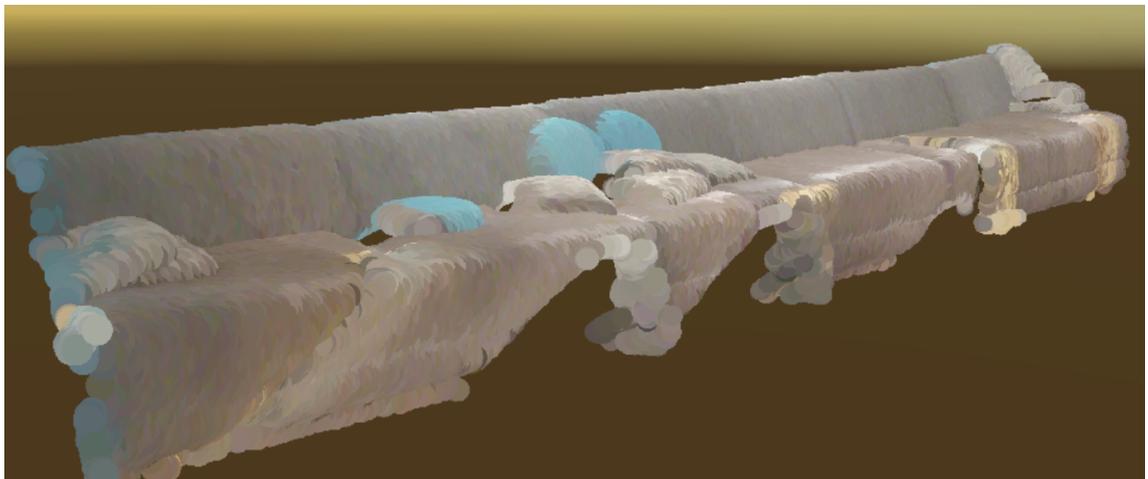


図 8: 確度 0.9 以上の物体の未補正レンダリング例



図 9: 確度 0.9 未満の物体の未補正レンダリング例



図 10: 確度 0.9 未満の物体の補正レンダリング例



図 11: VR 装置 (Oculus Rift S)

表 2: 実行環境

OS	Windows 10 pro 1903
CPU	AMD Ryzen Threadripper 3960X 24-Core Processor 3.80 GHz
GPU	Radeon RX 5500 XT
RAM	32.0 GB

3.2 評価

本節では提案レンダリング手法の評価を行う。まず、実行環境、評価方法について説明し、次に評価結果を説明する。

3.2.1 実行環境

視認性の評価を VR 機器 (Oculus Rift S [11]: 図 11) を用いて行った。

処理遅延の測定を表 2 に示す PC 上で Unity [12] Version 2020.1.15f1 上で行なった。VR 機器で実行すると、ある一定以上の FPS が出ないように設定されている VR 機器の性質によって、処理遅延を正確に計測できなかつたため、PC を用い計測した。

3.2.2 評価方法

提案レンダリング手法を用いて、ネットワークの輻輳を避けるため利用できる点群データ量に制約があった場合、VR オブジェクトの視認性を保ちつつ処理負荷が低減できることを実機を用いて確認する。評価する内容は提案レンダリング手法によって視認性が未補正レンダリング手法に対して向上していること、提案レンダリング手法によって補正レンダリング手法より処理負荷が削減されることである。

同一シーンに対して点群データの間引きを行うことで、ネットワーク越しにデータを転送することなく、ネットワークの輻輳を避けるためにデータ量が削減された状況を再現できる。そこで、ScanNet の点群データに対し r [cm] 以内に隣り合う点がないように間引いた点群データとして、 $r=0, 3, 5$ とし、

- 間引き無し点群データ ($r = 0, 9.02\text{MB}$)
- 3cm 以内に隣合う点群がないように間引いた点群データ ($r = 3, 2.92\text{MB}$)
- 5cm 以内に隣合う点群がないように間引いた点群データ ($r = 5, 1.09\text{MB}$)

の3つの点群データを用意した。ローカルに保存されたこれらの点群データを利用することで、視認性および処理遅延時間の計測を行う。

視認性の評価は未補正レンダリング手法と提案レンダリング手法によって VR 機器のディスプレイ上にレンダリングされた結果を比較し、処理負荷の測定は

- 補正レンダリングのみを用いた場合
- 提案レンダリング手法を用いた場合
- 未補正レンダリングのみを用いた場合

において動作させた場合の 6900 フレームの処理遅延時間の平均を計測し比較する。

3.2.3 評価結果

VR 機器のディスプレイへの描画結果を図 12~図 17 に示す。図 12、13 は 間引き無し点群データ ($r=0, 9.02\text{MB}$) を用いた場合の描画結果であり、図 14、15 は $r=3$ 、図 16、17 は $r = 5$ とした場合の描画結果である。

間引きを行うと物体を構成する点群数が減少するため、式 (1) より、各点の表示サイズが大きくなることが確認できる。これに伴い、物体の輪郭および細部の視認性が低下している。提案レンダリング手法を用いることでキッチン部分を主とした細部情報が識別に重要になる部分に補正レンダリングが施されていることを確認し、キッチン部分を代表例として図として掲載する。ここでは、補正

レンダリング手法によって取っ手部分やカウンター上に置かれた小物などの視認性が特に向上している。

$r=0$ の点群データ、 $r=3$ の点群データをレンダリングした結果では輪郭および細部における視認性の向上を確認した。 $r=3$ のデータに対する実行例の比較箇所を強調した、図 18、19 を用いて説明する。黒丸で示した部分が補正レンダリングを用いて描画することで視認性が向上した箇所である。補正レンダリングを用いることで上部の黒丸に示す箇所では戸棚が開いていることが立体的に表現されていた。下部の黒丸に示す箇所では取っ手の部分の視認性が向上している。対して、橙丸に示す部分が確度が高いため未補正レンダリングが用いられた箇所である。ここでは、未補正レンダリングでもごみ箱であることが識別できるため、未補正レンダリングでも視認性は損なわれない。 $r=5$ のデータに対する実行例（図 16、17）では補正レンダリングが行われている箇所でも視認性の向上は確認できない。これは、補正レンダリングによる補助をしても人間が識別できない程度にまで、間引きによって点群の情報量が低下してしまったためである。よって、点群データ量が少なすぎると、良好なサービス品質にならず効果も薄れるといえる。

また、処理遅延時間の測定結果は図 20 のようになった。

結果から、点群のサイズ数が少なくなればなるほど、補正の有無にかかわらず処理遅延時間が小さくなる。また、補正をかけると表示処理時間が大きくなるため、視認性の向上と処理遅延時間の短縮はトレードオフ関係にあるといえる。提案手法では補正を行わない場合に近しい処理遅延時間で視認性を向上させることに成功している。具体的には提案手法を用いることで、VR 空間全体に補正レンダリングを行う場合と比較し、補正を行わない場合からの処理遅延時間の上昇率を 84.1%～約 94.8% 削減した。このとき、提案手法では全点群のうち、約 23.7% の点群に対して補正を行っている。

ここで、間引きを無し点群データに注目し、表示処理遅延が小さくなる理由を考察する。間引きを行っていない 9.02MB のデータにおいて、補正する点群の数と表示処理遅延は図 21 のようになる。

図 21 をみると、補正する点群の数 20,000 点から 35,000 点の間で処理時間が大きく上がっており、その前後では線形的に変化していることが確認できる。これは、メモリ領域に描画情報をバッファリングする際、領域の解放と再割り当てが発生していることが原因として推測できる。

点群を補正する場合、その描画のためにバッファリングする情報は補正しない場合と比較して、より複雑な描画を行うために多くなる。よって、補正を行う点群数が増加するほど、全体としてバッファリングする情報量は増加する。メモリ資源は有限であるため、点群数が一定数を超えると領域の解放と再割り当てが必要となり、35,000 点以上の点群データに対する処理遅延時間が大きく増加している。



図 12: 未補正レンダリングの実行結果: 間引き無し点群データ ($r=0$)



図 13: 提案レンダリング手法の実行結果: 間引き無し点群データ ($r=0$)



図 14: 未補正レンダリングの実行結果: 間引いた点群データ ($r=3$)



図 15: 提案レンダリング手法の実行結果: 間引いた点群データ ($r=3$)



図 16: 未補正レンダリングの実行結果: 間引いた点群データ ($r=5$)



図 17: 提案レンダリング手法の実行結果: 間引いた点群データ ($r=5$)

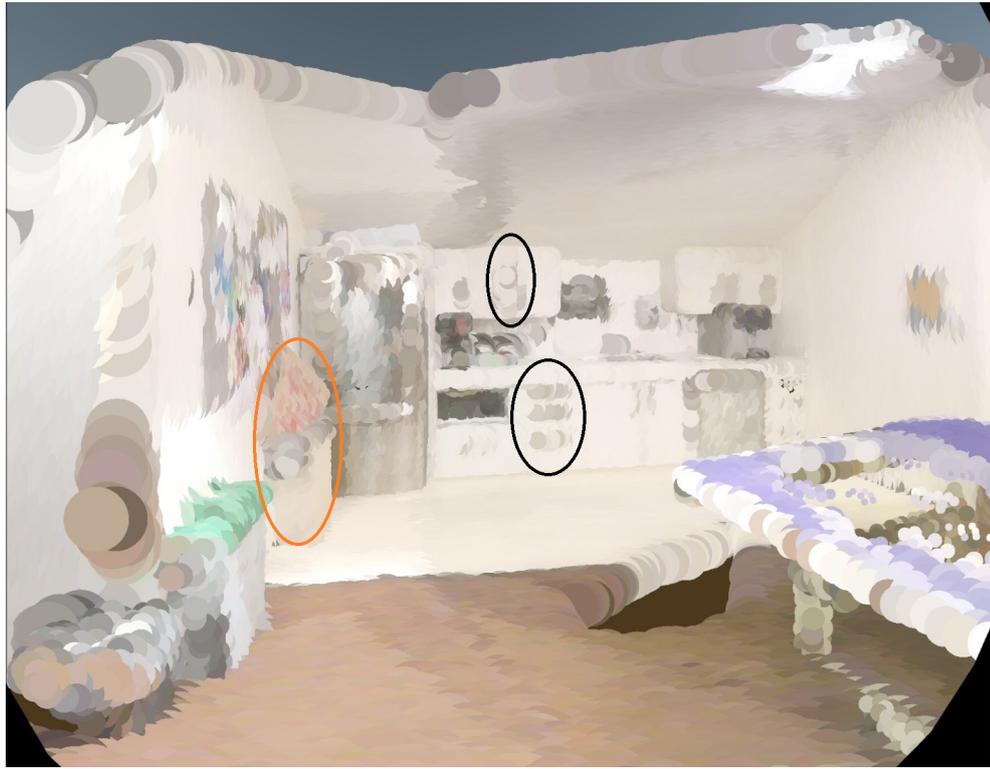


図 18: 未補正による視認性低下の例: 間引いた点群データ ($r=3$)



図 19: 提案レンダリング手法による視認性向上の例: 間引いた点群データ ($r=3$)

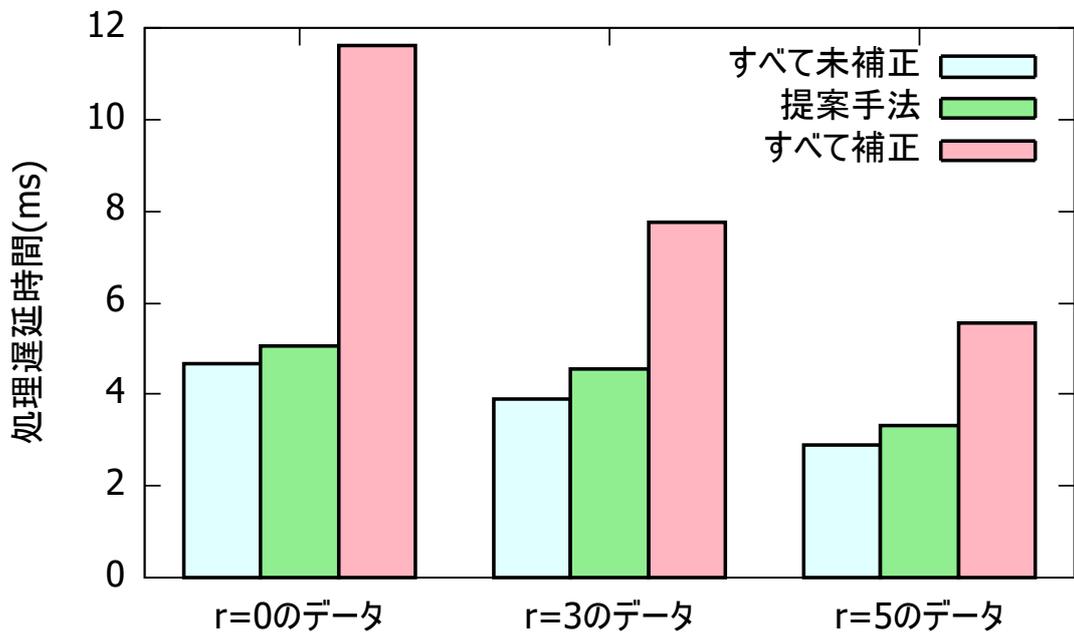


図 20: 処理遅延時間の測定結果

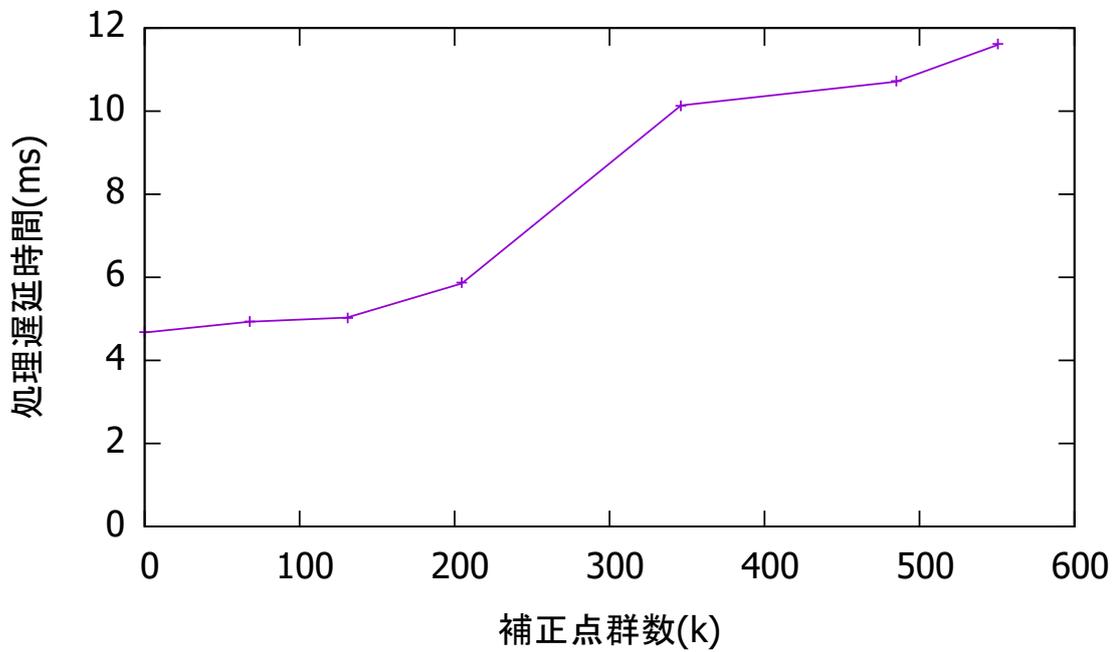


図 21: 補正点群数と処理遅延時間の関係

4 おわりに

現在、仮想現実をユーザが体験する VR 機器はゴーグル型であり、受け取る情報は視覚、聴覚のみである。しかし、視覚、聴覚以外の五感情報を受け取るようになると、そのために必要なデータ量や計算は増加する。それに対して、メモリ資源は有限のため、優先して処理すべきデータを取捨選択する必要がある。本報告では、仮想空間型ネットワークサービスを想定し、優先して処理するデータを決定するための指標として確率場情報を用い、実際に処理を行うことによる視認性の向上の検証および処理遅延時間を計測する検証を行った。検証の結果、大幅にデータ量を削減しない限りは、描画される 3D オブジェクトのうち、人間が識別できないであろう 3D オブジェクトに限定し、その細部および輪郭を補正表示する補正レンダリングを適用することで視認性を向上させることに成功し、すべてに補正レンダリングを適用する場合と比較し、すべてに非補正レンダリングを行う場合からの処理遅延時間の増加を約 84.1%～約 94.8%削減させることに成功した。また、補正レンダリングを適用する点群数を調整し実行することで、メモリ資源の領域解放および再割り当ての発生と処理遅延時間に関する推論を得た。この推論は、複雑な計算処理を行うデータを限定することで大幅に処理遅延時間を削減できるようになる可能性があることを意味する。今後の課題としては、データの取得およびネットワークを通してデータを転送する際に発生する遅延を含めてリアルタイム性に関する検証を行い、動的に変化するデータに対する検証実験を行うことが挙げられる。これによって、点群データを利用したリアルタイム仮想現実アプリケーションが実現に必要な条件を求める。加えて、動的に変化するデータを扱う際のデータ圧縮方法についても検討する必要がある。

謝辞

本報告を終えるにあたり、日頃より熱心にご指導、ご教授くださいました大阪大学大学院情報科学研究科の村田正幸教授に深謝いたします。大阪大学大学院情報科学研究科の荒川伸一准教授には、的確なご指摘やご助言を通して、研究の具体的な方向性を示していただきました。心より感謝申し上げます。また、平素からご指導くださいました大阪大学先導的学際研究機構大下裕一准教授、大阪大学大学院情報科学研究科小南大智助教、大阪大学経済学研究科経営学系専攻大歳達也助教に深く感謝いたします。最後に、様々な面で支えてくださった家族、友人、研究室の皆様に感謝の意を表して謝辞といたします。

参考文献

- [1] V. T. Minh, N. Katushin, and J. Pumwa, “Motion tracking glove for augmented reality and virtual reality,” *Paladyn, Journal of Behavioral Robotics*, pp. 160 – 166, Mar. 2019.
- [2] C. Trepkowski, D. Eibich, J. Maiero, A. Marquardt, E. Kruijff, and S. Feiner, “The effect of narrow field of view and information density on visual search performance in augmented reality,” in *Proceedings of 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 575–584, Mar. 2019.
- [3] VARK. <https://vark.co.jp>. (Accessed on 12/21/2020).
- [4] “「未来と芸術展」3D — 森美術館 - MORI ART MUSEUM.” <https://www.mori.art.museum/jp/digital/03/>. (Accessed on 01/04/2021).
- [5] L. Thibault, T. Jocelyne, R.-C. Amélie, and B. François, “Is it real? measuring the effect of resolution, latency, frame rate and jitter on the presence of virtual entities,” in *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pp. 5–16, Nov. 2019.
- [6] K. Wolff, C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung, “Point cloud noise and outlier removal for image-based 3D reconstruction,” in *Proceedings of the 2016 4th International Conference on 3D Vision (3DV)*, pp. 118–127, Dec. 2016.
- [7] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” in *Proceedings of the 2010 39th International Conference on Parallel Processing Workshops*, pp. 275–279, Sept. 2010.
- [8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet of Things Journal*, vol. 3, pp. 637–646, Oct. 2016.
- [9] S. M. Fraiss, “Rendering large point clouds in Unity,” Bachelor’s thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Sept. 2017.
- [10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “ScanNet: Richly-annotated 3D reconstructions of indoor scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2443, July 2017.

- [11] Oculus Rift S. https://www.oculus.com/rift-s/?locale=ja_JP. (Accessed on 01/24/2021).
- [12] Unity. <https://unity.com/ja>. (Accessed on 01/24/2021).
- [13] CloudCompare. <https://www.danielgm.net/cc/>. (Accessed on 02/10/2021).
- [14] 佐藤 広章, “三次元点群データを用いた実空間確率場表現に基づく物体推定手法の提案と評価,” 大阪大学基礎工学部情報科学科特別研究報告, February 2021.