

実時間の点群情報に対する力覚インタラクション手法

山崎健一^{†1} 橋本渉^{†2} 西口敏司^{†2}

本研究では、点群データで表現された実物体に対する力覚レンダリング手法について述べる。実物体の点群データは、デプスカメラによって手軽に得られるようになってきた。リアルタイムに取得される点群データを力覚提示する場合、デプスカメラにおける情報獲得に要する時間に対し、力覚提示における情報提示に必要な時間とのギャップが大きすぎるため、力覚提示の障害となっている。そこで本研究では、力覚提示における時間的なギャップを埋めるため、点群データを時間軸上で補間し、その補間方法にICPを導入することを提案する。本稿では、あらかじめ獲得しておいた点群データとそのフレームを用いて、提案する補間処理を適用し、その効果について評価をおこなう。

Haptic Interaction Method of Live Streaming Point Clouds

KEN'ICHI YAMASAKI^{†1} WATARU HASHIMOTO^{†2}
SATOSHI NISHIGUCHI^{†2}

This paper describes a haptic rendering method for real objects represented by point clouds. Recently, point clouds data of real objects can be easily obtained by using depth sensing camera. However there are problem caused by gap between video capturing rate and haptic rendering rate when haptic technologies are applied to live streaming point clouds data. Therefore, this paper proposes a method of time-scale interpolation of point clouds data based on iterative closest point algorithm. Then we have performed haptic rendering for previously stored data by applying the proposed interpolation process, confirmed the efficiency of the interpolation method by evaluations.

1. はじめに

実空間を認識するのに利用される装置の1つとして深度センサがある。深度センサを使うことで、非接触で3次元の構造を動的に把握することが出来る。深度センサで取得したデータは、仮想空間上に点群データとして反映させることが出来る。最近では、低価格な深度センサが発売され手軽に使えるようになってきた。また、メッシュデータ、ボクセルデータ、点群データなど様々なデータで表現された形状に対する力覚提示の研究が盛んに行われている。そこで、深度センサによる取得したデータから表現された点群データに対して力覚提示をすることが出来れば、遠くにあつて実際に触れないものや、触ると危険などの理由で直接触れないものを触っているような感覚を提示することが可能となる。例えば、博物館などに展示されている実際に触れないものや、檻などに入った直接接触には危険な動物を触っているような感覚を提示することが出来るようになる(図1)。また、テレプレゼンスなど遠隔地に対する提示においても実用性があると考えられる。

点群データに基づいた力覚提示をおこなう試みは、深度センサの普及に伴って活発に研究されている。例えば、点群データに対して三角形メッシュを張る等といった前処理を施してから力覚提示を行う方法がある[1]。しかし、深度センサで取得するデータはリアルタイムで更新されていく

ため、点群データもリアルタイムで更新される。そのため、計算量が多い前処理を施すと更新レートが低下してしまう。また、深度センサで取得するデータの更新速度は、力覚提示における更新速度に対して遅すぎるため、データの急激な変化に対して正確な力覚提示ができないという更新速度のギャップ問題がある。

そこで本研究では、深度センサで取得した点群データに対して、力覚提示の対象となる局所データを抽出し時間的に補間することで、更新速度のギャップを埋め、補間フレームに対して力覚提示が可能かを試みる。

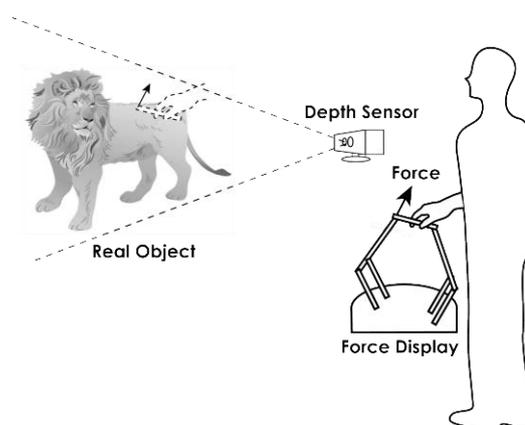


図1 点群情報に対する力覚インタラクションの概念図
Figure 1 The concept of haptic interaction for point clouds

2. 関連研究

深度センサによって得られたデータに対して力覚提示をする方法として、Fredrikらはproxyという概念を点群デ

^{†1} 大阪工業大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka Institute of Technology
^{†2} 大阪工業大学情報科学部
Faculty of Information Science and Technology, Osaka Institute of Technology

ータに適用している [2]. proxy とは力覚提示装置のカーソル (Human Interface Point: 以下 HIP) そのものに同心球を定義し, 同心球に含まれる点群データから反力ベクトルを導出するものである.

点群データに対する接触判定のメリットは, データの参照が最小限ですむことである. 深度センサから取得したデータは, カメラ座標で表現されている. そのため, 直交座標系で表現されている proxy の座標を深度センサに使われているカメラ座標系に変換し, 変換した座標を中心とした正方領域のデータだけを参照する. それにより, proxy の内部に含まれるか含まれないかの判断をするために参照するデータの数を減らすことができる.

3. 提案手法

3.1 問題点

深度センサの更新速度は力覚提示装置の更新速度に比べてはるかに遅い. そのため, 高速に移動する動的実物体のデータにおいては, フレーム間でのデータの変化量が大きくなってしまふ. Proxy の同心球の大きさを超える移動量の場合, 同心球をすりぬけてしまふなど, 接触判定が正常に行われず, 力覚提示に支障を来すという問題が起こる.

3.2 時間的補間方法

高速に変化する動的実物体のデータに対しての接触判定が正しく行われぬのは, 観測される時間間隔の変化量が大きいことによる問題である. その問題の解決方法として, フレーム間を補間するために新たにフレームを作成し, フレーム間の変化量を軽減させることができる.

まず, ある時間 t と $t+1$ において, 実物体は回転と等速直線運動をすると仮定する. この仮定のもと, 時刻 t および $t+1$ において実物体を観測して得られる点群データがどの程度回転, 移動したかを推定する手法として, 本研究では, ICP(Iterative Closest Point)アルゴリズムを用いる. ICPアルゴリズムは, 最近傍点の対応付けと, 対応付けからの幾何変換からの推定の2つの処理を交互に行うことで位置合わせを行う. ICPアルゴリズムを適用させると 3×3 の回転行列 \mathbf{R} と 1×3 の並進行列 \mathbf{T} からなる 4×4 の同次変換行列が得られる.

しかし, ICPアルゴリズムは点群数が増えると, 計算量も膨大になる. 計算に要する時間が長くなり, 力覚提示に支障を来しかねない. そこで, ICPの対象となる点群データを HIP 付近の局所領域のみに限定する.

逆に, ICPの対象となる点群データを小さい領域に限定してしまうと, ICPにおける特徴点抽出がうまくいかない場合もある. このため, 画像のマッチングや物体認識等に用いられている SIFT(Scale-Invariant Feature Transform)を適用する. SIFTは, 特徴点であるキーポイントの検出と特徴量の記述をおこなうアルゴリズムである. 検出したキーポイントに対して, 回転やスケール変化, 照明変化などに強

い特徴量を記述するため, フィルタリングした局所的データから SIFT を用いて検出した特徴点を用いて, ICP を適用し, 位置合わせを行う.

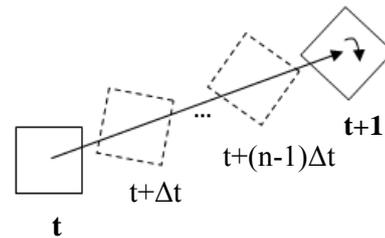


図 2 2フレーム間の補間方法

Figure 2 Interpolation method between two frames

位置合わせにより得られた同次変換行列の \mathbf{R} と \mathbf{T} の要素から, 移動経路上の要素を求める. 回転行列 \mathbf{R} の回転移動量を分割して微小回転を表現するためには, \mathbf{R} より等価回転軸と回転角を算出することになる[3]. 回転角を ϕ とすると回転角を 0 から ϕ までステップ数 n で増加させることにより, 移動経路上の回転行列 \mathbf{R} を求めることができる. 並進行列 \mathbf{T} については, ステップ数 n で線形に増加させることで, 移動経路上の並進行列 \mathbf{T} を求めることができる. この変換行列を適用させ, 線形的に補間することで図 2 のように 2 つの点群データの間を補間でき, $n-1$ 個の補間データを得ることが出来る. 補間した $n-1$ 個のデータを補間フレームとして扱うことで時間的補間をおこなう.

3.3 反力の計算

本研究では Fredrik らの方法[2]を参考に, proxy を用いて次のようなアルゴリズムで接触判定をおこなうことにした. 具体的には, 図 3 のように 3 つの同心円を定義し, proxy のどの円に点群データが含まれているかによって以下の処理を行う.

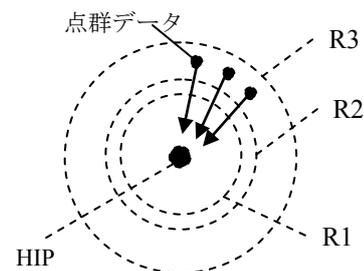


図 3 Proxy と点群による接触判定

Figure 3 Collision detection between proxy and point clouds

- R3 内に含まれる点群データから, 面法線を計算する.
- R1 内に含まれていた場合, 面法線の方に proxy を動かす.
- R1 内に含まれていなくて, R2 内に含まれていた場合, 面法線と直交した方向に proxy を動かす.

- R1 内にも R2 内にも含まれていない場合、力覚提示装置の座標と proxy の座標からなるベクトルの方向に proxy を動かす。

この処理を繰り返すことで、点群データとの接触判定を行う。また反力については、HIP が proxy の位置に近づくような力を発生させることにした。バネ定数により、HIP と proxy の距離に比例した反力が発生する。

4. 実験

提案した手法の有効性を確かめるため評価実験を行った。まず、補間方法の有効性と計算量を確かめる実験を行った。その後、あらかじめ保存しておいた深度センサで取得した 2 つのデータに対して提案手法を適用し、補間フレームのデータにおいて力覚提示が出来るのかを確かめる実験を行った。実験では、図 4 のような 3 自由度の力覚提示装置と、深度センサとして Microsoft 社の Kinect を用いた。実験に用いた計算機の性能を表 1 に示す。



図 4 3 自由度力覚提示装置
Figure 4 3DOF-force display

表 1 計算機の仕様

Table 1 Specifications of the computer

OS	Windows7 64bit
メモリ	8GB
CPU	Intel® Core i7-3770 3.40GHz

4.1 補間処理に関する予備実験

補間処理が適切に実行されているかどうかを確かめるために、1 辺が 10cm の立方体が置かれた実空間を用意する。この空間において、(a)移動前から左に移動(b)し、さらに奥に移動(c)した後で、右回転しながら左に移動(d)したと想定し、あらかじめ点群データを保存しておいた(図 5)。このデータを用いて、立方体付近の局所領域データを抽出し、補間可能かどうかを確かめる実験をおこなった。局所領域として、中心から 20 ピクセル、フレーム間の補間ステップ数 n は 2 として行った。

補間結果を俯瞰的に見たデータを図 6 に示す。実験結果から局所的なデータのみを用いて左右・前後の平行移動、回転運動において補間が出来ていることがわかる。

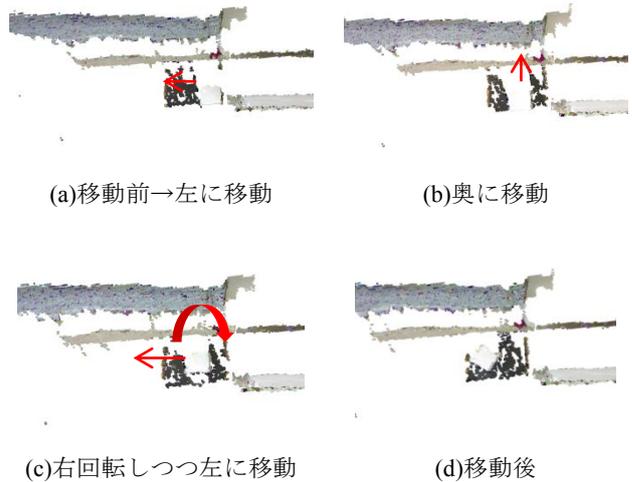


図 5 立方体の移動に関する点群データ

Figure 5 The motion of a cube represented by point clouds data

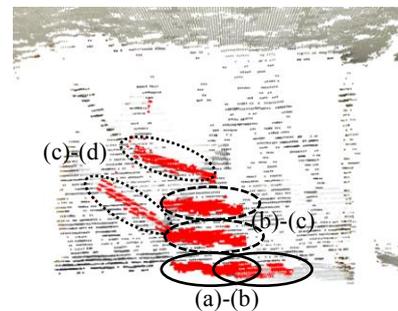


図 6 ICP を用いた 4 フレーム間の補間結果

Figure 6 The result of interpolation between four frames by using ICP algorithm

4.2 計算量に関する予備実験

提案手法の計算量がどの程度なのかを確認するために、局所領域の領域幅を変えながら、補間処理に要する時間を測定した。ここでは、補間の前段階として、局所領域の抽出、特徴抽出、ICP に要した前処理の時間と、補間フレーム作成に要する時間を測定した。補間フレーム作成の所要時間は、100 万フレームを計算した場合の 1 フレーム当たりの計算所要時間である。測定結果を表 2 に示す。

表 2 局所領域の大きさと補間ステップ数における計算所要時間

Table 2 Required Time of Computation for the size of localized region and the number of interpolation step between two frames

		処理に要する時間[ms]	
		前処理	補間フレーム作成
局所領域 の大きさ [pixel]	10	2	0.002
	20	12	0.012
	30	31	0.026
	40	58	0.051

この結果より、局所領域の大きさが所要時間に関係していることがわかる。一般に局所領域を大きくすると、ICPの精度が高くなり補間結果も正確になるが、この結果は計算コストが増大することを意味する。一方、前処理に対して補間フレーム作成の計算量は極めて小さいものとなっている。したがって、前処理のICPによって同次変換行列さえ求めれば、補間フレームの生成は計算機に負荷をかけることなく、分割数を増加させることができると考えられる。

4.3 補間データに対する力覚提示

補間フレームに対する力覚提示が有効かどうかを確認するため、補間フレームの有無による物体移動時の力覚提示を比較してみた。比較に用いた点群データは、立方体を上方向に持ち上げたものである。図7(a)は立方体を動かす前、(b)は動かした後の点群データと proxy, HIP の位置を示している。立方体を上方向に動作させると、proxy が立方体の表面を追従できず、めり込んでしまっている。このまま立方体を上方向に動かすと、HIP が立方体を貫通してしまうことになる。

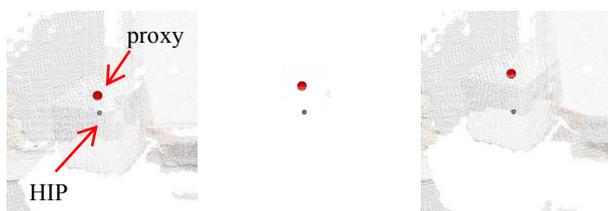


(a)移動前 (b)移動後

図7 立方体の移動時におけるHIPとproxy(補間なし)

Figure 7 The behavior of HIP and proxy when the cube moves to upper direction (without interpolation)

上記と同じ点群データにおいて、補間ステップ数10、局所領域の領域の大きさを20ピクセルとして補間をおこなった場合の様子を図8に示す。(b)の補間フレームはちょうど(a)と(c)の中間フレームである5ステップ目に相当する。補間フレームの点群が薄いのは、proxyの周辺20ピクセルのみを局所領域としており、局所領域のみ点群を補間しているからである。



(a)移動前 (b)補間フレーム (c)移動後

図8 立方体の移動時におけるHIPとproxy(補間あり)

Figure 8 The behavior of HIP and proxy when the cube moves to upper direction (with interpolation)

図をみても確認できるように、proxyが立方体の表面に追従していることがわかる。HIPは立方体の表面に追従できていないが、HIPにはproxyとバネで結ばれているよう

な力覚が生じるため、いずれ立方体の表面に到達し、貫通することはなくなる。

4.4 提案手法に関する考察

ここまで、あらかじめ保存しておいたオフラインデータに対して提案手法を適用してきたが、オンラインデータに適用する場合について検討する。図9は補間処理を実時間で実行した場合のタイムラインである。デプスカメラからt, t+1フレームが獲得されると、両フレームの前処理から同次変換行列が計算され、n分割の力覚提示用補間フレームが作成されていることを意味する。表2で力覚提示処理の計算負荷が小さいことが示されているので、図のように力覚提示と前処理を並行することが期待できる。したがって、tフレーム獲得時から1フレームと前処理分の遅れは生じるものの、オンラインデータに対する補間フレームによる力覚提示が可能だと考えることができる。

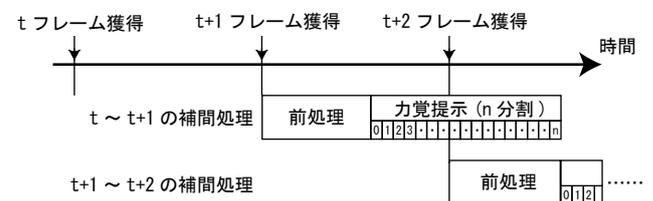


図9 補間処理のタイムライン

Figure 9 A timeline of desired interpolation

5. まとめ

本研究では、深度センサと力覚提示装置の更新速度のギャップを埋めるために、局所的に点群データを補間することで擬似フレームを作成し、更新速度のギャップを埋める方法を提案した。この方法の有効性を確かめるために評価実験を行った。実験の結果、局所的な点群データを用いて補間が出来ることがわかった。また、補間データに対して力覚提示が出来ることもわかった。

現時点ではデプスカメラからあらかじめ取り込んでおいた点群データに対して補間をおこなっているが、計算量の観点からリアルタイムデータに対する適用の可能性について示すことができた。

今回はICPの対象物体が立方体に限定されていたが、今後は様々な形状に対して本手法を適用してみるとともに、物体の移動だけでなく変形にも適用可能か検討したい。

参考文献

- 1) Sreeni K. G., Subhasis Chaudhuri: Haptic Rendering of Dense 3D Point Cloud Data, IEEE Haptic Symposium, pp.333-339, 2012
- 2) F.Ryden, S.N. Kosari, H.Chizeck.: Proxy Method for Fast Haptic Rendering from Time Varying Point Clouds; Intelligent Robots and System (IROS), IEEE/RSJ International Conference, pp.2614-2619, 2011
- 3) 3) 長谷川健介, 増田良介:基礎ロボット工学-制御編-, 昭晃堂,1994