

デバイスの制御およびVR空間描画用PCはCPU Intel Core i5-8400, メモリ 16GB, GPU NVIDIA GeForce GTX1080 TI / 11GB GDDR5X を搭載した Windows 10 pro PC を使用した. VR空間の構築には Unity ver. 5.6.2f1 (64bit), VR空間の観察には Oculus Rift CV 1 を用いた. VR空間の提示および伸縮距離の算出, デバイス先端部の駆動は毎秒 90 フレームで行っている.

3. 硬軟感提示システム

硬軟感を提示するために(1) 提示する視覚情報の表現力向上, (2) 先端伸縮による接触感提示手法の改良, (3) デバイスの先端を付け替えられる機構の搭載の3点の改良を加えた.

3.1 視覚情報の提示

まず, 仮想物体硬軟感のCG表現方法について述べる. 仮想物体は, ExtickTouchの先端の3次元座標 P_{Tip} から求めた値を入力として変形させる. まず, 仮想物体とデバイスの接触点 $P_{Virtual}$ を算出する. $P_{Virtual}$ とは, デバイス伸縮部分の後端から先端までを通る方向ベクトルと接触対象との交点を指す. 次に, 接触点と仮想物体内に侵入したデバイス先端の距離 d を求め, 仮想物体の変形に用いる値に加える力 F を弾性定数 k を用い以下の式で算出する (図3).

$$d = |P_{Virtual} - P_{Tip}| \quad (1)$$

$$F = k \cdot d \quad (2)$$

まず F の値が, 設定した F の上限 F_{MAX} 以上かを確認する. F の上限を設定する理由は, F の値が大きくなるほど仮想物体が変形する量が増加する仕組みとなっており, F の値を制限しなければ F の値が増加しつづけ, 仮想物体が際限なく変形してしまうためである. F が F_{MAX} 以上の場合, F の値を F_{MAX} に変更し, d の長さを制限するために以下の式で d を算出する処理を行う.

$$d = \frac{F_{MAX}}{k} \quad (3)$$

算出した F を用いて, 接触点 $P_{Virtual}$ を中心とし, 仮想物体のメッシュの各頂点に力を加えられたかのような変形を行う. メッシュの各頂点に加える力は $P_{Virtual}$ とメッシュの各頂点の距離の差 d_{base} を用いて逆2乗の法則により減衰する. また, メッシュの各頂点に元の頂点位置に戻ろうとする力 F_{spring} (式(4))を加え, これらの力を1フレーム毎に指数関数的に減衰する力 $F_{damping}$ (式(5)) (h は定数)を加えることで仮想物体の硬軟感を実現する. 以上の要素を考慮し, メッシュの各頂点の3次元座標 P_{moved} を式(6)より算出する [4]. ここで, d_{moved} は元のメッシュの頂点と描画されているメッシュの頂点の距離の差を表す3次元座標, t は時間である. 以上の式(1)から式(6)を用いて仮想物体の変形を

$$F_{spring} = k \quad (4)$$

$$F_{damping} = \frac{F_{spring}}{h} \quad (5)$$

$$P_{moved} = \left(\frac{F \cdot \Delta t}{1 + d_{base}^2} - d_{moved} \cdot F_{spring} \cdot \Delta t \right) \cdot (1 - F_{damping} \cdot \Delta t) \quad (6)$$

行った様子を図4に示す.

3.2 伸縮制御の変更

仮想物体の変形にデバイスの伸縮距離を合わせることで視覚情報に合わせた接触感を提示し, より現実に近い感覚提示を行う. 具体的なデバイス伸縮処理について図5に示す. まず, デバイス伸縮部分の後端から先端までを通る方向ベクトルと, 実世界に配置されている実物体と同じ位置に配置されている仮想物体との接触点 P_{Real} を算出する. 次に, $P_{Virtual}$ と P_{Real} の距離の差 l_{VMAX} を以下の式から求める.

$$l_{VMAX} = |P_{Virtual} - P_{Real}| \quad (7)$$

そして, VR空間内でデバイスが伸縮する長さ l_V を l_{VMAX} と d の差から算出する (図5).

$$l_V = l_{VMAX} - d \quad (8)$$

また, l_V はVR空間内での距離となっているため, このままでは実空間上のデバイスの伸縮に利用することができない. そこで, 以下の式を用いて, 実世界の寸法に合わせた後, リニアアクチュエータへ送信する値 l_{sign} へと変換する

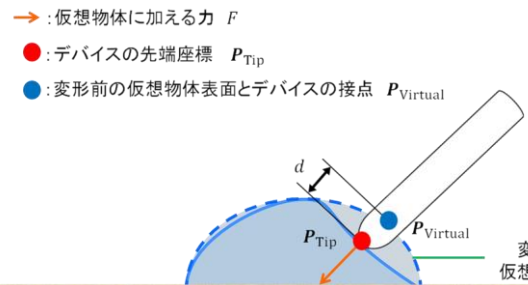


図3 仮想物体変形処理

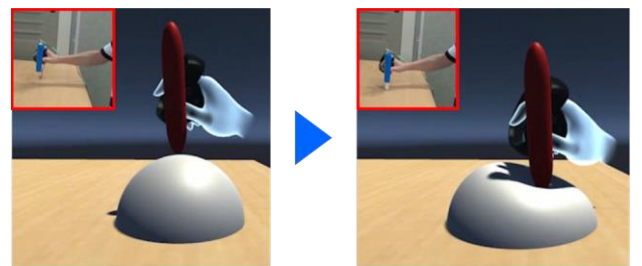


図4 仮想物体の変形の様子

ために 1024 段階に正規化を行う。

$$l_{\text{sign}} = \frac{l_v}{l_{\text{device}}} \cdot 1023 \quad (9)$$

ここで、 l_{device} とはデバイス伸縮部分の VR 空間上での長さ、つまりリニアアクチュエータの可動域である 10cm の VR 空間上での長さである。 l_{sign} の値が 1023 を超える場合、このデバイスでは接触感が提示できないため、伸縮不可と判定する。

また、従来システムにおけるデバイスの伸縮速度 (256 段階中 140) を今回のデバイスにそのまま用いた場合、デバイスで仮想物体を押し込んだ際に仮想物体表面に合わせてデバイスが即座に縮んでしまい、仮想物体を押し込む途中に発生するはずの連続的な接触感を提示することができなかった。そのため、デバイスが縮む場合の速度を 5 まで遅くすることで仮想物体を押し込む感覚を実現した。また、デバイスが伸びる場合の速度に関しては従来のシステムと同じアルゴリズムを利用した。具体的には、デバイスの伸縮距離が大きい場合はデバイスの伸縮速度を上げ、伸縮距離が小さい場合にはデバイスの伸縮速度を下げるという手法を利用した[3]。以下の式により伸縮速度 v を求める。

$$v = a \cdot v_{\text{MAX}} + (1 - a) \cdot v_{\text{min}} \quad (10)$$

$$a = \frac{l_{\text{sign}}}{l_{\text{MAX}}} \quad (11)$$

ここで、 v_{MAX} は最高伸縮速度、 v_{min} は最低伸縮速度、 l_{MAX} はデバイス先端の最大の長さである。

さらに、仮想物体が変形する上限まで変形している場合、その場に押し留める力が変形途中よりも強くなるよう、デバイスが伸びる速度を 256 段階中 250 に変更を行った。

3.3 先端付け替え機構

最後に、ExtickTouch の特徴である実物体との接触を活かした機構を採用することで、提示可能な硬軟感の向上を行った。前述の通り本デバイスの先端部分は 3D プリンタで作成した。したがって、机と接触した際の接触感は常に同様であり、仮想物体の材質や硬軟感までは提示できていないことが課題であった。そこで我々はデバイスの先端を変更することで異なる触覚を提示可能な先端付け替え機構を作成した。実際に作成した機構を図 6 に示す。

この機構に用いる先端については、今回は硬軟感提示システムとして組み込むために、硬さが 6 種類あるウレタンフォーム (八幡ねじ社製, 表 1) を用意し、これをペン先の形状 (半径 10mm, 高さ 15mm の円錐) に整形したものを作成した。作成した先端を図 7 に示す。実際に、デバイスの先端を作成した先端に変更すると、やわらかい物体に触れているように感じられた。この時、使用したウレタンの中でも硬さ#5や#6といったものは弾力も感じられたが、

硬さが#1 に近づいていくほど机と接地した際にウレタンがつぶれてしまい、弾力は感じにくくなった。全体として机と接触した際の衝撃が緩和され、従来の先端よりも仮想物体から得られる接触感が大きくなったという副次的な効果も得られた一方で、軟らかい先端では机と接触した際に変形して机と接触している表面積が大きくなってしまふことで、摩擦が大きくなりなぞる動作が少し行いづらくなっ

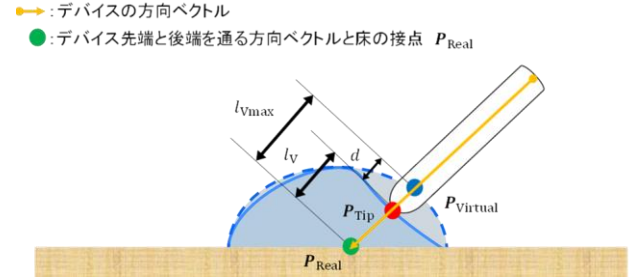


図 5 デバイス伸縮処理



図 6 先端付け替え機構



図 7 実装した先端

表 1 ウレタンフォーム

ウレタンフォームの種類	密度 (kg / m ³)	型番
#1	16 ± 1.5	4979874852335
#2	20 ± 2.0	4979874852373
#3	21 ± 2.0	4979874852410
#4	35 ± 3.0	4979874852458
#5	40 ± 4.0	4979874852496
#6	50 ± 5.0	4979874852533

てしまうという問題も見つかった。これについては今後さらに先端に用いる材質を検討し、軽減したいと考えている。

4. むすび

本稿では、VR 空間内において仮想物体との接触感を提示するデバイス ExtickTouch の表現力の向上を目的として改良した硬軟感提示システムについて述べた。この機構を用いることで接触感だけでなく、状況に応じて視覚情報を変更し、先端を付け替えることで硬軟感の提示も可能になった。今後は、このシステムで提示できる「視覚」「先端伸縮による接触感」「先端部分の材質」のパラメータを単独、あるいは組み合わせた場合に、接触感提示にどのような影響をおよぼすのかを評価実験によって明らかにし、表現したい硬軟感に合わせた適切な組み合わせを提示できるようにしていきたい。

参考文献

- [1] 3DSystems Inc., PHANToM (<https://ja.3dsystems.com/haptics-devices/3d-systems-phantom-premium>), (2018 年 9 月 12 日)
- [2] H. Benko, C. Holz, M. Sinclair, and E. Ofek, “Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers.” The Proceedings of 29th Annual Symposium on User Interface Software and Technology (UIST 2016), pp. 717-728, 2016.
- [3] 片岡, 山本, 大槻, 柴田, 木村 : ExtickTouch: 仮想物体への接触感を提示する先端伸縮型デバイスの開発, 第 27 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2019) USB Memory 論文集, 2019
- [4] Jasper Flick, Mesh Deformation, a Unity C# Tutorial (<https://catlikecoding.com/unity/tutorials/mesh-deformation/>), (2019 年 11 月 29 日)