



THE VIRTUAL REALITY SOCIETY OF JAPAN

VR 空間における、高速かつ即時性の高い文字入力手法に関する 検討

Fast and immediate text input technique in Virtual Reality

岡本幸樹

Kouju Okamoto

デジタルハリウッド大学 (〒 101-0062 東京都千代田区神田駿河台 4-6 御茶ノ水ソラシティ アカデミア 3F/4F,
a20dc036@dhw.ac.jp)

概要: VR 空間での文字入力には主にバーチャルキーボードや物理キーボード等が用いられている。しかし、それらの入力手法は高速性に欠けていたり、コントローラーとは別にデバイスを用意しないといけなかったりなどの欠点を抱えている。そこで、それらの欠点を克服したものとして、コントローラーのボタンを用いた文字入力手法を検討する。

キーワード: 文字入力手法, 文字入力

1. はじめに

現在、使い易さや習得のしやすさに焦点を合わせた研究はいくつも存在している [8][2]。しかし、習得がしやすいものが最終的に効率の良いものとは限らない。例えば従来の PC で使われるテキストエディタにおける vim やその系統はその最たる例で、習得しやすくないものの、習得してしまうと非常に快適に操作が出来るようになる。このように、学習コストが高くても最終的なパフォーマンスが高くなる文字入力手法というもの存在するはずだと考えられる。

又、文字入力の際に既存のコントローラーから別のデバイスに持ち替える手間を無くしたかったことから外部デバイスやセンサーは用いず、既存のコントローラーを使用したソフトウェア的な解決方法を模索した。

2. 先行研究

2.1 ジョイスティックを利用した文字入力手法

VR でのコントローラーを利用した文字入力手法として、PizzaText[8] がある。この手法では、1 つないし複数の文字が円を扇型に切り分けた各部分に配置され、それを手持ち型のゲームコントローラーの 2 つあるジョイスティックを用いて選択する。具体的には、片方のジョイスティックで扇形になった部分を選び、もう片方のジョイスティックでその扇形内に配置された文字を選ぶようになっている。

2.2 コントローラーの 3 次元的な動きを利用した文字入力手法

DayDream Lab の Drum Keyboard[2] はドラム型のキーボードをハチで叩くような UI をしており、一般的によく使われている aim-and-shoot による文字入力よりも速度が出るとされている。しかし、VR 空間上にドラムキーボードを表示する都合上それを見て打つ必要があり、姿勢や手の位

置に制限がかかる。VR 空間内でテキストを入力する際は体が常に同じ姿勢であるとは限らないため、これは何かしら改善の余地がある。

他にも、キューブを 3x3x3 に並べたキューブキーボード [3] 等が提案されている。これは 3x3x3 に並べた立方体のそれぞれに文字を割り当て、ユーザーがコントローラーを動かした軌跡から単語を推測する手法となっている。トリガーを押している間のみ追従して動くため、手の位置の制限は存在しないものと考えられる。論文にはキーが把握しづらい等の問題点が上げられていたが、ハプティックフィードバックやキーボードの向きを揃える等の改善を行うことでキーボードが視界外にあったとしても入力が可能なのではないかという希望が持てる。しかし、この手法は基本的には単語予測によって単語を確定していく必要がある。単語予測を行う前の文字列も選択することが可能ではあるものの、その場合本来打ちたかった文字以外が含まれていたとしてもそのまま選択されてしまうことになる。プログラミングを行う際やマイナーな分野の単語を扱いたい場合等にはそういった辞書外の単語を使う機会が多いと考えられ、入力する文字をより直接的に選択出来る仕組みを用いた方が良いのではないかと思われる。

3. 検討した手法

広く用いられているレイキャスト式以外の、コントローラーを主体とした文字入力手法について、コントローラーの 3 次元空間的な動きを利用したもの [7][2][3] やジョイスティックを用いた手法 [8] の研究はそれなりに行われているようであったが、ボタンを用いているものは殆んど見付けられなかった。しかし、ボタン入力は人間が慣れているものでもあるし物理的なフィードバックも存在しており、これを

用いた入力手法に関して研究する余地があると考えられる。

ボタンのみを用いた入力方法が採用されない・研究されない要因は、コントローラーの種類によってその配置や個数が異なること、そして VR アプリケーション自体で何らかの機能が割り当てられていることではないかと推測されるので、本研究ではコントローラーを Meta Quest 2 Controller に限定し、VR アプリケーション内でボタン入力を出来る限りホストアプリケーションに伝えないための手法として NeosVR 上でのボタンのみを用いた入力手法について実装した。

3.1 定義

3.1.1 Button Per Character (BPC)

Button Per Character (BPC) は、1 文字を入力するために入力する必要のあるボタンの押し込み、ボタンのタッチ、ボタンの力の個数。「押し込み」はボタンの押下や、ジョイスティックの押し込みを示す。「タッチ」はボタンやジョイスティックに触れてはいるが押し込んでいない状態を指す。「力」はボタン等の押し込み具合やジョイスティックの傾き具合を 0 1 に正規化したものを指す。

例として、BPC 2 以下で、というと 2 つ以内のボタンの押し込み・ボタンのタッチの組合せで、という意味を指す。左トリガーの押し込みで 1 文字入力するのならば BPC=1、X ボタンと Y ボタン同時押しであれば BPC=2、X ボタンにタッチしながら左トリガー押し込みの場合でも BPC=2 となる。

3.2 設計

3.2.1 使用する入力

NeosVR 内で片方の Meta Quest 2 Controller から受け取れるイベントを、表 1 に示した。

この全てを入力として用いることも出来るが、そのままだと誤動作に繋がりそうな入力も存在するのである程度使用しない入力を決定する。

誤った入力が行われないように、原則「タッチ」のみでは文字の入力を行わないものとする。又、ジョイスティックの方向入力以外の「力」の入れ具合は「押し込み」と比べて感覚的なものになってしまうこと、指を中途半端な状態で

表 1: Meta Quest 2 Controller の片側から取得可能なイベント

	押し込み	タッチ	力
ジョイスティック	O	O	O
Trigger	O	O	O
Grip	O	O	O
A/X	O	O	x
B/Y	O	O	x
Rest	X	O	x

保持しなくてはいけないことが指の疲れに繋がることから扱わないこととする。

一般的に、ジョイスティックを押し込む際にジョイスティックの傾きも入力してしまうことが多く、「押し込みのみ」の入力が少しやりづらいと感じたためこれも使わないものとする。

NeosVR 特有の問題として、ジョイスティックの入力はアプリケーション側に伝えないようにする手法があるものの X,Y,A,B ボタンの押し込みに関してはそれが出来ない。又、他の VR アプリケーションでも文字入力中にも何かしらの機能を埋め込んでいる可能性は高いと考えられる。そこで、一旦それらのボタンの押し込みも利用しないものとした。

これを基に、「必ず使用しなければいけないボタン」は左右トリガー、左右グリップのうち 1 つ、「使用可能なボタン」はそれに加えて左右ジョイスティックタッチ、X、Y、A、B 各ボタンのタッチとなる。尚、ジョイスティックの傾き入力は取得可能であるものの、今回はボタン入力による文字入力手法について検討しているため、使用しないものとする。

3.2.2 文字列編集用の機能の配置

現実で文字列を入力する際には、文字列編集は重要な役割を果たすと考えられる。そのため、本来であれば編集用に様々な機能が必要となる。しかし、今回の目的は文字入力であるため、最低限必要な機能としてカーソルの移動と文字削除に絞ることとした。そこで、それらをアルファベットの配置を検討する前に優先的に配置している。アルファベットよりも優先して配置されているのは、今後日本語等に拡張したいと考えた時にこういった文字列編集用の機能はボタン配置が変わらないことが望ましいと考えられるからである。具体的には、カーソル移動は両手ジョイスティックのタッチと左右それぞれのトリガーの組合せ、文字削除は左手の Rest ボタンのタッチにしている。

この配置は、直感的であることと使用頻度が高いと考えられるので簡単に入力出来ることを重要視して暫定的に決定されたものである。尚 Rest ボタンのタッチのみで文字削除の機能を持っているのは「タッチのみでの入力を行わない」という原則に反しているが、他の押し込めるボタンはアルファベットの入力に用いたかったことと、Rest に関しては普段指が乗って誤って押すことがなかったことから例外として使用している。

3.2.3 アルファベットの配置

アルファベットは、基本的に Norvig の研究 [6][4] から得たアルファベットの出現頻度に基づいて、頻出するものほど少ないボタンの組合せで入力出来るようにした。個人的な感覚で、母音が片側に集まっていると分かりやすいなど感じたため、上記の法則を基にすつつも母音は左手に集めている。右ではなく左を選んだ理由は特なく、この研究の前段階としてコンセプトを試していた際に仮で左手に母音を置いていたものを継承しているだけである。

頻出する文字をより小さい BPC で打ちたいので、まずはボタン入力の一覧を作成し、そのうち BPC の小さいものから割り当てるものとした。同じ BPC の場合、覚え安さを考えて配置することとした。尚覚えやすさに関しては現在は根拠がなく、感覚で設定している。

BPC を小さくしたいかつ、今後編集機能を充実させる事も考えると不必要に使用するボタンを増やしたくないと考えた。そこで、まずは「BPC 2 以下で全アルファベットを入力すること」を目標とし、そのためにはいくつのボタンを組合せればよいかを考える。

必ず必要なボタンが 4 種類あり、それとその他の n 個のボタンを合わせた集合の中から 1 つないし 2 つ取り出した組み合わせの数が 26 以上の最小の数となればよいので、「ボタンを 1 つ押す場合」と「必要なボタンの中から 2 つ選んで押す場合」「必要なボタンの中から 1 つと、その他の n 個のボタンの中から 1 つ選んで押す場合」の 3 通りが考えられる。

$$4 + 6 + 4n \geq 26 \quad (1)$$

$$4n \geq 16 \quad (2)$$

$$n \geq 4 \quad (3)$$

よって、ボタンは 4 つ存在すればよい。

ジョイスティックタッチは、文字編集用の機能に割り当てている都合で被らせたくないと考えた。なので、X、Y、A、B 各ボタンのタッチを利用することとした。

また、これを行った上でもまだ数カ所使用不可能な組み合わせが存在する。例えば、

最終的な対応表は図 2 に示した。この図で、Yt は Y ボタンタッチ、Xt は X ボタンタッチ、lg は左グリップ、lt は左

トリガー、At は A ボタンタッチ、rg は右グリップ、rt は右トリガーを示す。横線が引かれている場所は、タッチのみである等で使用不可能であり対応する文字の割り当てがされていないものか、表の形式上不要な箇所であることを示す。

3.3 実装

実装には NeosVR[1] 上で扱えるノードプログラミング言語である LogiX を用いた。これは、普段使いすることで使っているうちに改善点が見付けられると考えたこと、開発するのに慣れていたこと、不特定多数に配付して試してもらおうと思った際に exe を配るよりも NeosVR 内ではよく行われているインゲームのツールを渡す手法で行えるのでやりやすいこと、細かい設定の変更等を VR 空間で使用しながら行えるため毎度ビルドしたりする手間がなく、プロトタイピングに適していると考えたことなどが理由となっている。

他の技術候補としては Unity を用いたスタンドアロンの VR アプリケーションの作成と SteamVR overlay を用いた開発が上げられる。Unity を用いたスタンドアロンアプリケーションは性能を評価したり等するのには適しているが、普段使いしている内に改善していくような開発手法を取ろうとしていた為適さなかった。SteamVR overlay は普段使い出来ないという問題を克服出来るが、ボタン入力がホストアプリケーションに伝わるのを止めることが出来ず、適さなかった。

実際に制作したのは本手法を実装したキーボードと、実験に必要な値を計測するためのツールである。比較対象とする仮想キーボードは NeosVR の標準のものを用いた。

その特性上完全に最適化された実装を作れてはいないものの、比較対象とする仮想キーボードも NeosVR 上の同じ仕組みを使って作られていることから実装による差はそこまで出ないのではないかと仮定した。

表 2: ボタンと文字の対応表

	no-key	lg	lt	rg	rt	At	Yt	Xt	Bt
lg	a	—	—	—	—	p	y	u	z
lt	e	o	—	—	—	c	b	i	x
rg	n	f	d	—	—	h	v	w	j
rt	t	m	l	s	—	r	k	g	q

3.4 現在の課題・今後の展望

まだ手法自体も開発を初めたばかりであり、最適な配置の模索はおろか他の手法との比較実験も行えていない。まずはしっかりとした実験の場を設け、適切な人数の被験者を基にしたデータを集めたい。

又、比較実験を行う手法についても模索の段階であり、適切なフレーズセットの選定や適切な比較項目についても検討を重ねたい。特に、本手法は「初めの学習曲線を見捨てても、慣れた後の使い勝手が良い」ことを重視しているため、「慣れた後の使い勝手」について測定する方法を模索していきたいと考えている。

参考文献

- [1] Neos metaverse. <https://neos.com/>. "Accessed: 2023-07-18".
- [2] Google VR. Andrey Doronichev, Group Product Manager. Daydream labs: exploring and sharing vr's possibilities. <https://developers.googleblog.com/2016/05/daydream-labs-exploring-and-sharing-vrs.html>. "Accessed: 2023-07-07".
- [3] 直貴. 立体キーボードを用いた vr 向け文字入力手法. Master's thesis, 筑波大学大学院, 2020.
- [4] harry guinness. The best starting words to win at wordle. <https://www.wired.com/story/best-wordle-tips/>, 2022.
- [5] I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, p. 754–755, New York, NY, USA, 2003. Association for Computing Machinery.
- [6] Peter Norvig. English letter frequency counts: Mayzner revisited or etaoinsrhldcu. <http://norvig.com/mayzner.html>. "Accessed: 2023-07-07".
- [7] Caglar Yildirim and Ethan Osborne. Text entry in virtual reality: A comparison of 2d and 3d keyboard layouts. In *HCI International 2020 – Late Breaking Papers: Virtual and Augmented Reality: 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings*, p. 450–460, Berlin, Heidelberg, 2020. Springer-Verlag.
- [8] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. Pizzatext: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 2018.