

# 既存の Web コンテンツを立体的に表現する VR 向け Web ブラウザアプリケーションの開発

吉川遼<sup>†1†2</sup> 横山徹<sup>†3†4</sup> 八嶋有司<sup>†1</sup> 彦坂和里<sup>†1</sup>

**概要**：近年，VR 空間を活用した立体的な情報提示手法が多く見られるが，Web コンテンツの閲覧方法や Web コンテンツのデザインは平面を想定したものがほとんどである．本研究では，既存の Web コンテンツのマークアップ言語を応用しつつ，VR 空間内に Web コンテンツを立体的に表示する Web ブラウザアプリケーションを開発し，実験を通し既存の平面的な Web ブラウザとの閲覧体験の差異について存在感の観点から比較・検討をおこなった．

## 1. はじめに

21 世紀以降のスマートフォンやタブレット型端末の普及に伴い，これまで PC 上での閲覧を想定して設計されていた Web コンテンツは，様々な媒体と解像度を想定したレイアウトが要求されるようになった．レイアウトを実装するにあたって，Web ブラウザの表示領域（ビューポート）の幅や高さに合わせて，段組や画像といったコンテンツの幅や高さを動的に調整するリキッドデザインや，スタイルシートでメディアクエリを用い，コンテンツの幅や高さの条件を段階的に設定するレスポンシブデザインを用いることが昨今の Web デザインにおいては主流となっている．

しかしながら，上記のデザイン手法や，そのデザイン手法を実装する上で必要な HTML，CSS といったマークアップ言語において，昨今普及しつつある Virtual Reality (VR) で用いられるヘッドマウントディスプレイ（以下：HMD）での閲覧を想定したレイアウトや記述は考慮されていない．これまで Web コンテンツの表示を想定していた PC やスマートフォン，タブレット型端末の表示領域は，一般的に平面かつ長方形の形状をしている一方，HMD 内の表示領域は全天球型であり，これまで以上に柔軟な Web コンテンツのレイアウトが可能になることが期待される．また，VR 空間の三要素として挙げられている自律性 (Autonomy)，対話性 (Interaction)，臨場性 (Presence) の AIP キューブ[1]において，Web コンテンツの相互作用性は保持しつつも，特に臨場性の部分において，単なる平面上のレイアウトだけではなく，奥行き感やスケール感といった部分を強調することで，より臨場感や存在感のある Web コンテンツのレイアウトが可能になることが考えられる．

そこで本稿では著者らが提案している Web コンテンツに奥行き感やスケール感を付加し，表示する VR 向け Web ブラウザアプリケーション[2]の開発をおこなった．開発した Web ブラウザアプリケーションによる Web コンテンツ鑑賞が既存の VR 向け Web ブラウザアプリケーションによる鑑賞と比較し，奥行き感やスケール感といった存在感の知覚にどのような差異があるかを実験結果から考察する．

## 2. 先行事例・関連研究

VR 向けの Web コンテンツ提示手法に関する先行事例としては，AR，VR 向けの API である WebXR がある．HTML 内に WebXR 用の canvas 要素を記述し，その canvas 内で 3D オブジェクトを描画することで，HMD で当該 Web ページを閲覧すると，ブラウザ内で VR や AR アプリケーションを動作させることを可能にするが，既存の Web コンテンツを VR 向けに拡張する本研究とは目的が異なる．

また，Web コンテンツを VR 内に配置する関連研究としては Toyama et al. の「VRrowser」[3]や Egawa et al. の「Multi-Window Web Browser」[4]があるが，いずれも平面の Web コンテンツ表示領域を VR 空間内に配置しているという点で本研究とは方向性が異なる．

## 3. VR 向け Web ブラウザアプリケーション

### 3.1 アプリケーションの試作

本章では，PC やスマートフォン，タブレット型端末向けにレイアウト・記述された Web ページを維持しつつ，同一ページ内で VR 向けのレイアウトを実現する Web ブラウザアプリケーションを提案する．また，この機能を実現するためのアプリケーションの試作をおこなった．

### 3.2 アプリケーションの概要

本稿で提案する Web ブラウザアプリケーションは，Web ページのマークアップ言語である HTML，CSS に記載された VR 向けの要素やクラス名を取得し，画像やテキストなど個別の要素を，CSS で指定された座標や奥行きに関するプロパティの値を基に，VR 空間内に要素を配置する Web ブラウザアプリケーションである．

### 3.3 Web ページ内の VR 空間用要素の記述

VR 空間用の要素は以下のように実装した．まず HTML の要素や CSS のセレクタに特定のクラス名を用意しておき，CSS 上で当該クラス名が設定されている要素の display プロパティ値を hidden（非表示）に設定しておく．これにより，スマートフォンやコンピュータ上では，VR 空間用の要素は非表示となる．また VR 空間用の要素は HTML 上で

†1 名古屋文理大学情報メディア学部情報メディア学科

†2 名古屋大学大学院情報学研究科社会情報学専攻

†3 東京藝術大学音楽学部

†4 株式会社アマナ

個別に id を付加し, CSS にて各要素の id をセレクタとし, 位置や幅・高さ, 奥行きといったプロパティを個別に指定する. このように個別に指定することで, スマートフォンや PC 向けに記述されている HTML や CSS の内容を保持しつつ, VR 空間向けの記述を追記することが可能となる. 例えばスマートフォンと VR 空間で同一の画像を提示したい場合は, 通常の img 要素と特定のクラス名が付与された img 要素を同じ HTML 内に記述することで, 各端末で同一の画像を表示させることができる (図 1).

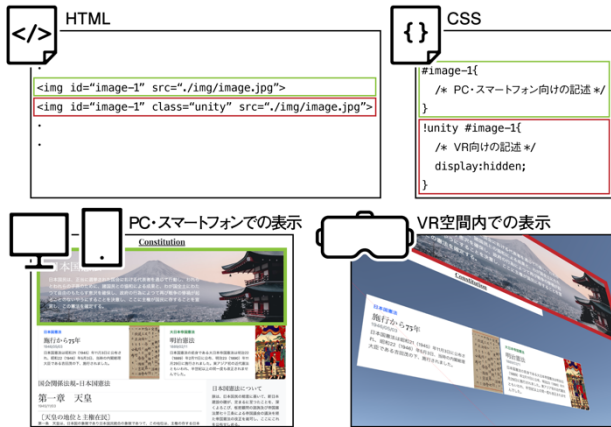


図 1 要素の記述例と表示例

### 3.4 Web ブラウザアプリケーションの構成

本研究で試作したアプリケーションは Unity で開発した. 開発において, 第 1, 第 3, 第 4 著者がアプリケーションの仕様策定を, また第 1 著者が操作機能の実装を, 第 2 著者が HTML, CSS 各要素の取得と VR 空間内に要素を配置する際の処理に関する実装を担当した.

アプリケーションの仕様策定アプリケーションは起動時に指定された URL から HTML, CSS を読み込む. 読み込んだ HTML ファイルから, VR 空間用のクラス名が設定されている要素のみを HTML から取得し, 読み込んだ CSS で指定された各要素のプロパティを配列として保持する.

アプリケーションは配列に格納された各プロパティとその値を基に要素を空間内に配置する. 具体的には, CSS の top, left プロパティを空間内の x 座標, y 座標に変換し, 空間内に配置する. また width, height プロパティを基に要素の幅と高さを空間内に反映させる. また z-index プロパティに指定される値に一定の係数を乗じ, 空間内の z 座標に変換し, 空間内に配置する. これらの処理を HTML から取得した各要素に適用し, 3D 空間上に Web ページとして構築する (図 2). この他, VR 空間用のクラス名が設定されている body 要素に background-color プロパティが指定されている場合には, その値を取得し, 3D 空間の背景色として反映することで, VR 空間においても, PC やスマートフォンと同様の背景が表示されるよう調整した.

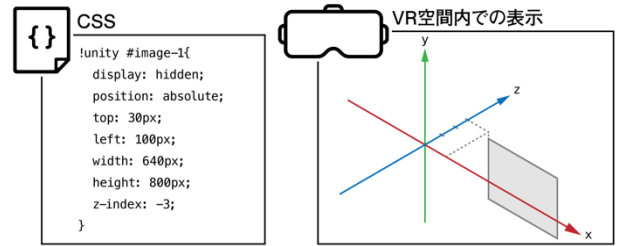


図 2 CSS 上での記述と空間内での座標配置

また, 要素の種類に応じて次のような処理を実装した. img 要素 (画像) については矩形を作成し, その矩形に要素の src 属性に指定された画像を読み込み, 貼り付けて表示させている. main, article, section, div 要素は矩形として扱い, 矩形の幅や高さ, 背景色を矩形の色として反映させている. h1 から h6 までの見出し要素, p, span 要素はテキストオブジェクトを作成し, CSS の font-size プロパティや font-weight プロパティを反映させ, 表示している (図 3).

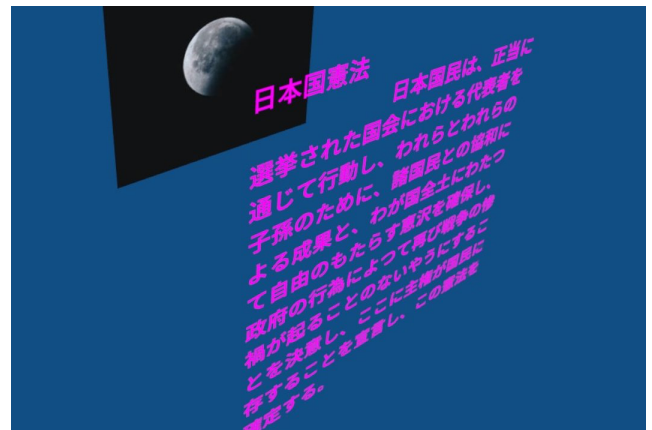


図 3 テキスト, 背景色の表示例

### 3.5 操作方法

本アプリケーションでは, ユーザーは自分の視点を移動させることで, 空間内に配置されたコンテンツの閲覧をおこなう. 左のコントローラーのスティック操作で前後左右への視点移動, 右のコントローラーのスティック操作で視点の上昇・下降, 左右のコントローラーのトリガーで視点の回転ができる.

## 4. 評価実験

### 4.1 実験の概要

3 で開発した Web ブラウザアプリケーションを用いた Web コンテンツの鑑賞が既存の VR 向け Web ブラウザアプリケーション (Meta Quest Browser : 以下, 既存ブラウザ) を用いた際と比較し, 存在感の知覚にどのような差異が生じるかを明らかにするために実験を行った. 実験には 18 歳から 22 歳までの大学生 10 名が参加した. 実験では各ブラウザでの鑑賞前に操作方法について実験実施者が説明し,

説明終了後、参加者は既存ブラウザと開発したブラウザそれぞれで Web コンテンツを鑑賞した。Web コンテンツ全体を鑑賞後、各ブラウザでの見え方の違いや操作感に関するアンケートならびにインタビューを実施した。なお、順序効果を考慮し、参加者のうち5名は開発ブラウザ→既存ブラウザの順番で、残り5名は開発ブラウザ→既存ブラウザの順番で体験するよう、ランダムで振り分けをおこなった。



図 4 実験の様子

#### 4.2 実験用 Web コンテンツ

実験用の Web コンテンツとして、デザイン思考に関する架空のイベントを紹介する Web ページを作成した。Web ページはいわゆる「ランディングページ」のように 1 ページで完結する縦長のレイアウトとなっている。ページ内に画像、見出し、テキストを配置し、デザイン思考やプロトタイプリングに関する説明や、架空のイベントの紹介、これまでのイベント概要や写真といったコンテンツを配置した。

なお、今回のページは上述の通り 1 ページで完結するような内容であるため、他ページや外部の Web サイトへのハイパーリンクなどは設置していない。VR 向けの記述として、背景画像は z-index プロパティの値を下げ、見出しや画像は z-index プロパティの値を上げることで、視界の前後で奥行き感をもたせるような設計を操作的に実施した。既存の VR 向けブラウザである Meta Quest Browser での表示例を図 5 に、開発したブラウザでの表示例を図 6 に示す。



図 5 既存の VR 向け Web ブラウザ (Meta Quest Browser) での表示



図 6 開発した VR 向け Web ブラウザでの表示

#### 4.3 結果・考察

アンケートの結果を図 7 に示す。なお、先述の順序効果について、既存ブラウザ→開発ブラウザの順で体験した群と開発ブラウザ→既存ブラウザの順で体験した群の各アンケート項目の平均値で t 検定を実施したが、有意差はみられなかった ( $p=0.41$ )。

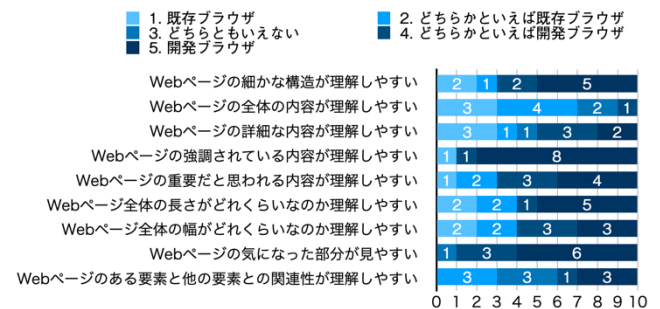


図 7 アンケート結果

アンケート結果より、「Web ページの強調されている内容が理解しやすい」「Web ページの気になった部分が見やすい」といった項目で開発したブラウザの方が理解しやすいと答えた割合が他の項目に比べて多かった。この結果から、z-index プロパティの値を空間内の z 座標に反映させる方法は、VR 向けの Web デザインとして、奥行き感を提示する上で有効である可能性が示唆された。

また、Web ページの気になった部分が見やすいという点に関しては、インタビューより「気になった部分の近くに寄ることができる」「写真に近寄って見ることができる」といった発言にあるように、今回実装した、自身の視点を上下前後左右に移動させる操作方法が効果的であった可能性がある。例えばスマートフォン上でも画像を拡大することは可能だが、画面が狭く、広範囲の詳細を同時に把握することは難しい。一方、今回開発したブラウザでは、対象に自身そのものが近づき、視野に対象の全体を収めつつ、対象の詳細を観察することができる。VR 上における Web コンテンツの適切なスケール変換については今後の課題でもあるが、対象そのものに近づくことができるという点は今回開発したブラウザの大きな特徴であるといえる。

しかしながら、「Web ページの全体の内容が理解しやすい

い「Web ページの全体の構造が理解しやすい」といった項目に関しては、比較的既存ブラウザの方が理解しやすいと答えた割合が多かった。これは既存ブラウザが PC やスマートフォンといった普段利用している端末に近いレイアウトで表示されること、また図 6 で示したような開発ブラウザで提示した奥行き感かつスケール感のある情報表示だと全体像が把握しづらい、といったことが原因として考えられる。また、テキスト情報量の多さや配置といった要因も考えられる。今回の実験参加者の視聴行動を更に詳細に分析しつつ、今回のようなテキストによる情報の多い Web ページ以外にも、画像や動画を中心とした Web ページなど様々な種類の Web ページで実験をおこない、VR 上でのレイアウトについて比較検討していく必要があるといえる。

奥行き感やスケール感といった Web コンテンツの存在感に関しては、インタビューから「要素が前面に出ていたり奥に配置されていたりすることで重要度がわかった」「壁画のような迫力があつた」といった意見も得られた。今回の画像やテキストの幅や高さは、PC 上で表示されている解像度を基に算出した値を用い、横幅 1900px 程度の 1 カラム型のレイアウトをベースに要素の位置を設定したため、実際に VR 空間内で鑑賞すると横幅や高さのあるコンテンツ配置となったが、スケール感を体験するにあたって、PC 上での要素の幅や高さの値が利用できる可能性が示唆された。また「360 度にコンテンツが配置されていると面白そう」「自分が振り向いた方向にコンテンツがあるとよい」といった意見も得られた。今回実験用に作成した Web ページは奥行きこそあるものの、コンテンツが一方向に集中していたことから、全天球型の VR 空間を十分に活用できていたとは言い難い。そのため、PC やスマートフォン向けのレイアウトを考慮しつつも、既存の Web サイトを VR 空間内においてどのようにレイアウトすることが Web デザインとして適切なのか、VR 向けの Web デザイン原則についても今後検討していく必要があるといえる。

最後に、アプリケーションの操作性に関しては「気になるコンテンツを選択すると自身の視点はその地点まで移動するとよい」「目線の移動でスクロールできるとよい」といった改善点がインタビューの意見から得られた。今回のアプリケーションでは自身の視点を移動させる操作方法を実装していたが、コンテンツが大きくなった分、移動に時間がかかり、見たいコンテンツにすぐたどり着けないといった事象もあつた。今回の実験参加者は 10 名中 3 名が VR 未経験者、5 名が VR を過去数回程度体験したことがあると回答していたため、コントローラーを使った VR 空間内での操作自体に不慣れであった可能性は十分に考えられる。今後、VR 経験者にも実験対象を広げつつ、よりスムーズにコンテンツのスクロールや視点の移動を可能にする操作方法の実装を検討していきたい。

## 5. おわりに

本稿では Web コンテンツに奥行き感やスケール感を付加し、表示する VR 向け Web ブラウザアプリケーションを提案し、試作したアプリケーションを用いることで、既存の VR 向けブラウザと比較し、奥行きのあるコンテンツ配置や自身の視点を要素の目の前まで移動可能にする操作方法を実装することで、要素の強調されている内容や気になる部分への理解が容易になる可能性が示唆された。

一方、既存ブラウザと比較し、Web ページ全体の内容理解や構造の理解のしやすさで劣る部分があつたほか、コンテンツを閲覧する際の操作性に課題もみられた。

また、今回試作したアプリケーションでは、VR 向けのレイアウトを実現するために、画像やテキストの座標 (top, left プロパティ) や要素の大きさ (width, height プロパティ)、奥行き (z-index プロパティ) を CSS 上で手動で設定する必要がある。今後は、より柔軟なレイアウトを実現するために、メディアクエリを用いたレイアウトの実現や、現在の Responsive Web Design で実装されているような可変レイアウトの VR 空間内への適用、空間内への要素の自動配置などの機能実装を進めていく予定である。

**謝辞** 本研究は JSPS 科研費 21K12573 の助成を受けたものです。また、実験にご協力いただきました名古屋文理大学の学生に感謝申し上げます。

## 参考文献

- [1] Zeltzer, D. Autonomy, interaction, and presence. *Presence: Teleoperators & Virtual Environments*. 1992, vol. 1, no. 1, p. 127-132.
- [2] 吉川 遼, 八嶋 有司, 彦坂 和里: 既存の Web デザインを VR 空間に転用するアプリケーションデザインの提案, 第 12 回社会情報学会中部支部研究会・第 7 回芸術科学会中部支部研究会・第 10 回情報文化学会中部支部研究会合同研究会論文集. 2021, p. 13-16.
- [3] Toyama, S. et al.. VRowser: a virtual reality parallel web browser. In *International Conference on Virtual, Augmented and Mixed Reality*, Springer, Cham. 2018, p. 230-244.
- [4] Egawa, R., and Ijiri, T. Multi-Window Web Browser with History Tree Visualization for Virtual Reality Environment. In *The Adjunct Publication of the 34th Annual ACM Symposium on User Interface Software and Technology*. 2021, p. 32-34.