2008－HCI－130 (5)
2008／11／6

ブラウザ上でのインタラクティブな3次元仮想環境を実現するサーバレンダリングに関する検討

安田 敏宏[*1] 伊藤 雄一[*1] 櫻井 智史[*1] 山口 徳郎[*1] 岸野 文郎[*1]

A Study on Server Rendering for Interactive 3D Virtual Environments on a Web Browser

Toshihiro Yasuda[*1] Yuichi Itoh[*1] Satoshi Sakurai[*1] Tokuo Yamaguchi[*1] Fumio Kishino[*1]

Abstract – There are several studies on a server rendering approach for mobile devices. It builds 3D virtual environments and renders suitable images for user's requests on the server side, while only shows requested images on the client side. And it can provide 3D virtual environments that every user can use easily, even if they have thin client's devices. In this paper we propose a novel technique which builds interactive networked 3D virtual environments on a web browser by using a server rendering approach. For achieving realtime interactions, we also propose a method that shows the imitated images using CSS(Cascading Style Sheets) as the rendered images to reduce server's burden. In addition, we implement a system using our approach, and simulate its response time. As a result, our proposed method can shorten the response time by 98 % from the response time with an ordinal method.

Keywords : Virtual Reality, Web3D, Networked Virtual Environments, Server Rendering

1. はじめに

ネットワーク経由で3次元仮想環境を扱う機会が増えている．例えば，アバタを操作し，他のユーザとコミュニケーションを取ることができるメタバース[1] [2]や，多数のユーザが同時に参加するMMORPG[3]などがある．しかし，これらの多くは，サーバからネットワーク経由で取得した3Dデータをクライアント側でレンダリングする手法を用いており，インタラクションの応答速度がクライアントのマシンスペックに大きく依存してしまうという問題があった．また，3次元仮想環境の構築処理のためにクライアント側に専用のアプリケーションやプラグインが必要となる問題もある．これらにより，OS，内部プラグインのバージョンの違いや利用者のPCスキルによっては，利用が困難になる場合がある．一方で，携帯端末等において構築される仮想環境では，サーバレンダリングがよく用いられる[4] [5]．これは，サーバ側で動的にレンダリングした画像を，クライアント側に送信することで，仮想環境を実現する手法である．これにより，処理性能の低いクライアントであっても仮想環境を閲覧することができる．

そこで本稿では，サーバレンダリングにより，既存のウェブブラウザのみでインタラクティブな3次元仮想環境を実現する手法を提案する．ウェブブラウザはWebを利用する際に最もよく使われるツールであり，ほとんどの端末で標準に搭載されている．そのため，PC初心者であっても扱いやすい仮想環境を提供することができる．また，ウェブブラウザで画像を表示することから，専用のアプリケーションやプラグインのインストールが不要となる．さらに，サーバ側の負荷軽減手法として，CSS(Cascading Style Sheets)を用いた擬似画像の提示による画像生成回数の削減手法を提案し，リアルタイム性を持った仮想環境を実現する．

*1: 大阪大学大学院情報科学研究科
*1: Graduate School of Information Science and Technology, Osaka University

2. 関連研究

ネットワーク経由の3次元仮想環境を実現する手法は，クライアント側で仮想環境を構築する手法(クライアントレンダリング)，クライアントとサーバとで処理を分散する手法，そしてサーバ側で行う手法(サーバレンダリング)に大別される．

2.1 クライアントレンダリング

この手法は，サーバ側の負荷が少ないため，多数のクライアントを想定した大規模のシステム構築に適している．商用サービスで最もよく用いられているGeometry Replicationは，クライアント側で3Dデータを保持し，サーバ側の命令に基づいて仮想環境をレンダリングする手法である．さらに，この手法は3Dデータの保持方法に関して2種類に分けることができる．全てのデータをクライアント側で保持する手法と，必要な時にサーバからダウンロードする手法である．前者はMMORPG[3]などのオンラインゲーム，後者はSecond Life[1]やLively[2]などで用いられている．しかし，クライアント側に専用のアプリケーションやプラグインが必要であるため，手軽に利用することができない．また，レンダリング速度がクライアント側のマシンスペックに大きく依存するため，全てのクライ

アントへ同品質の仮想環境を提供することを保証できないという問題もある．

2.2 クライアントとサーバで処理を分散する方式

最も複雑な処理である仮想環境の構築をサーバとクライアントで分散させることで，リアルタイム性を維持しつつクライアントへの負荷を軽減することができる．Impostors は，サーバ側でレンダリングした 3D オブジェクトの画像を，クライアント側で構築した単純な仮想環境にテクスチャとして貼り付けることで複雑な仮想環境を構築する手法である [6] [7]．さらに，Jeschke らは視野内の環境構築に必要なテクスチャだけを送信することで必要データ量を削減し，リアルタイム性を実現している [8]．しかし，仮想環境の構築はクライアント側で行うため，やはり専用のアプリケーションが必要である．

2.3 サーバレンダリング

仮想環境の構築およびレンダリングをサーバ側で行い，クライアント側での処理量を大幅に軽減することで，スペックの低い端末でも仮想環境を閲覧することができる．Azzedine らは，サーバ側で仮想環境の構築と，要求に基づいたレンダリングを行い，クライアント側で受信画像を用いてパノラマ画像を生成することで，携帯端末での仮想環境の閲覧を可能にしている [4] [5]．さらに，サーバ側がクライアント側からの要求に優先度を設定し，要求される画像を予測しあらかじめ送信することで，1 秒以内の更新速度を実現している．しかし，これらの手法はサーバ側にかかる負荷が大きいため，リアルタイムに応答することが困難である．

3. サーバレンダリングによるウェブブラウザ上での仮想環境の実現

本章では，サーバレンダリングを用いて，ウェブブラウザ上で閲覧できるインタラクティブな 3 次元仮想環境を構築する手法を提案する．図 1 にシステム構成を示す．この手法では，サーバ側で仮想環境を構築し，要求された視点位置からの画像をレンダリングする．一方，クライアント側では，ウェブブラウザで利用できる Web 技術のみを用いてレンダリング条件をサーバへ送信し，受信した画像の表示のみを行う．これにより，クライアントに仮想環境構築の負荷をかけることなく，ウェブブラウザ上でインタラクティブな 3 次元仮想環境を実現することができる．また，クライアントが 3 次元情報を保持していた場合，複数のクライアントが同時に環境に変化を与えると，データ間に矛盾が生じることがある．しかし，この手法ではサーバが仮想環境の情報を一括管理しているため，クライアント間で仮想環境の整合性を維持できる．次節では，本手法を適用したシステムの詳細について述べる．

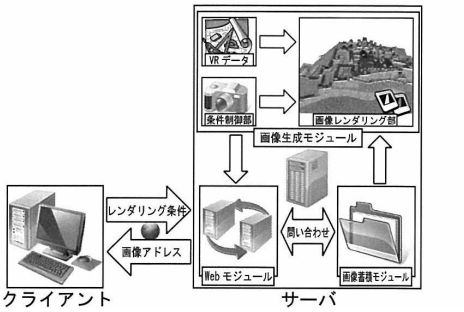

図 1 システム構成
Fig. 1 System architecture.

3.1 サーバ

サーバの主な機能は，仮想環境を構築すること，クライアントから指定された視点位置などのレンダリング条件に応じて仮想環境をレンダリングすること，さらに，生成した画像をキャッシュして再利用することの 3 つである．これらの機能をモジュールとして実装し，サーバの処理量増加に伴う負荷分散やシステム構成の拡張，変更を容易に行える構成とした．

Web モジュール: Web モジュールは，クライアントからレンダリング条件を受信し，画像ファイルのアドレスを送信する．まず，クライアントからレンダリング条件を受け取ると，それと同一条件の画像が既に画像蓄積モジュールに保存されているかを問い合わせる．保存されている場合は，該当画像ファイル名を受け取る．また，保存されていない場合には，画像生成モジュールへレンダリング要求を出し，画像のレンダリングが終了後，そのファイル名を受け取る．そして，受け取ったファイル名から該当画像のアドレスを生成し，クライアントに送信する．

画像蓄積モジュール: 画像蓄積モジュールは，サーバ側でキャッシュの役割を担う部分であり，生成した画像を保持する．また，画像ファイル名をレンダリング条件のハッシュ値で管理することで，条件に対して一意にファイル名を決定する．

画像生成モジュール: 画像生成モジュールは，仮想環境を構築し，要求画像をレンダリングする．まず，画像レンダリング部が，あらかじめ VR データから，仮想環境を構築しておく．そして，条件制御部からレンダリング条件が渡されると，それに基づいて画像をレンダリングする．さらに，レンダリングされた画像を画像蓄積モジュールに保存し，画像ファイル名を Web モジュールへ渡す．

3.2 クライアント

クライアントは，レンダリング条件の送信と，受信画像の表示を行う．レンダリング条件は，仮想環境内

のカメラ位置，画角，パン・チルトの角度で構成される．また，レンダリング条件は，各パラメータ値の絶対指定と，前進や右折のように現在位置からの相対指定が可能である．そして，これらの機能をウェブブラウザで利用できるHTMLやJavaScript，Ajaxなどの Web 技術を用いて実装することで，特殊なプラグインやアプリケーションを用いず，なおかつOSなどのプラットフォームにも依存しないサービスの提供が可能となる．

3.3 画像生成回数の削減手法

サーバレンダリングでは，多くの複雑な処理を行うためサーバ側が高負荷となる．そこで，サーバ側の負荷軽減手法として，**CSSを用いた擬似画像提示による画像生成回数の削減手法**を提案する．一般的に，ユーザは仮想環境を閲覧する際には，目的のオブジェクトを探して仮想環境内を動き回る探索タスクと，目的のオブジェクトをじっくり閲覧する観察タスクを繰り返し行っている[9]．その際，探索タスク中は画像の精度よりも更新速度が重要とされており，観察タスク中は更新速度よりも詳細な画像が必要とされる．そこで，解像度や仮想環境の描画精度が低いが，更新速度の速い擬似画像を探索タスク中に提示し，観察タスク中にのみ高精細な生成画像を提示することで，サーバ側の画像生成の回数を削減することができると考えられる．

擬似画像の提示には，ウェブブラウザとの親和性があり，比較的負荷が少ないCSSを用いる．CSSとはHTMLの要素をどのように表示するかを指示するものであり，表示画像の位置や大きさなどを指定することができる．これにより，ウェブブラウザのみで仮想環境を実現できるという本稿の提案手法の特長を保ちつつ，サーバの負荷軽減による応答時間の短縮ができると考えられる．あらかじめ各オブジェクトを様々な視点位置からレンダリングした画像をサーバ側にキャッシュしておく．そして，インタラクションが行われると，サーバ側でレンダリング条件とVRデータから各オブジェクトの画面上での位置やサイズを算出し，クライアントに送信する．さらに，クライアント側で必要なオブジェクトの画像をサーバ側のキャッシュから取得し，受信データに応じてウェブブラウザ上にCSSで配置することで擬似画像を提示する．これにより，画像をレンダリングするよりも短い処理時間で現在の視点位置での仮想環境の状況を把握できる品質の画像を提示することが可能となる．

4. 実装結果と考察

サーバを，CPU: Pentium 4 3GHz，メモリ: 1GB，nVidia GeForce 6600GT チップセットを搭載したPC上で構築した．また，OSはWindows XPで，Apache

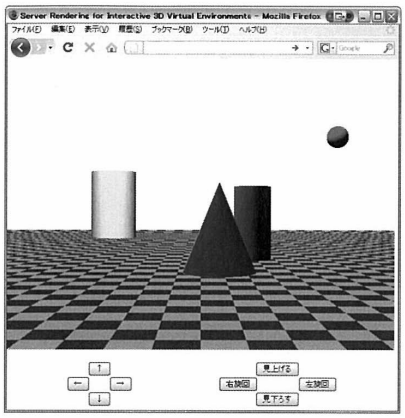

図2 実行画面例
Fig. 2 A screen capture of client's browser.

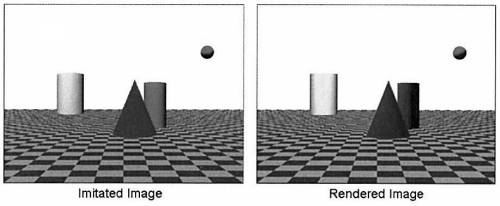

図3 擬似画像と生成画像
Fig. 3 Screen captures of the imitated image and the rendered image.

HTTP Server 2.2.4 および，PHP 5.2.4 を動作させており，クライアント・サーバ間の通信は1GbpsのLAN上で行った．

図2は，実装システムにおけるクライアントでの実行画面例であり，ウェブブラウザのみで3次元仮想環境を閲覧できていることが分かる．図3はCSSを用いた提案手法により生成された擬似画像と，同一視点位置からの生成画像である．この図から，擬似画像はオブジェクトの形状や，光源による陰影が正しく再現できていないものの，十分に仮想環境の状況を把握できる品質であることがわかる．また，擬似画像からレンダリング画像への切り替えの際に違和感は無く，探索タスクから観察タスクへスムーズに移行できることを確認した．

また，システムの応答時間を算出するため，end-to-endの遅延を測定した．出力画像のサイズは，携帯端末で表示可能な640pixel × 480pixelとし，形式は転送量軽減を考慮しJPEG形式とした．クライアントが1つの要求を出してから，対応する画像表示が完了するまでを1試行とし，画像生成時間と転送・処理時間に分けて測定した．なお，測定にはFirefoxのアドオンであるFirebugを用い，50試行の平均 (Ave.)

表 1　測定結果 (msec)
Table 1　Results of response time.

	画像生成		転送・処理		合計 (遅延)	
提示画像	Ave.	SD	Ave.	SD	Ave.	SD
生成画像	1051.0	19.2	18.2	3.6	1069.2	19.6
擬似画像	9.1	0.3	11.5	1.3	20.6	1.3

と標準偏差 (SD) を算出した．画像生成時間は，クライアントが要求を出してから，サーバでレンダリングと画像ファイルの保存が完了するまでの時間とする．また，転送・処理時間は，サーバからのレスポンスが返ってきてから，画像をダウンロードし，表示するまでの時間とする．そして，これらの合計をシステムの応答時間とする．測定条件は，提案手法を用いない場合 (生成画像を提示) と，用いる場合 (CSS による擬似画像を提示) とした．測定結果を表 1 に示す．この結果から，提案手法を用いない場合には，サーバ側での画像生成時間が遅延のほとんどを占めていることが分かる．これが，提案手法を用いることで 99.2%程度短くなっている．これは，提案手法を用いる場合は，サーバ側の処理が，各オブジェクトの位置やサイズといったデータの計算を行うのみでよいためである．また，転送・処理時間も提案手法を用いることで 37.2%程度短くなっている．これは，クライアントが各オブジェクトのレンダリング画像を取得する際にウェブブラウザのキャッシュを利用しており，視点変更毎の画像ダウンロード量が大幅に減少したためと考えられる．このように，提案手法を用いることで画像生成時間，転送・処理時間共に短縮され，それに伴いシステムの応答時間が 98.1%短縮された．また，応答時間は先行研究 [10] でリアルタイム性のある Web ページの更新速度として示されている 100msec 以内を達成しており，探索タスク中はリアルタイムな仮想環境の提供が可能となった．

観察タスク中の応答時間の短縮手法として，視点位置の予測によるサーバ側のキャッシュヒット率の向上手法が考えられる．利用者が要求するであろう視点位置をサーバ側で予測し，その位置での画像をあらかじめ生成することでキャッシュのヒット率が向上し，応答時間を短縮することができる．先行研究でも，利用者の移動可能な範囲を制限することで現在の視点位置から次に要求されるであろう視点位置を予測する手法 [4] や，3 次元仮想環境内で，最も多くの情報量を持つ視点を得る手法 [11] などが提案されているが，利用者の視点要求，前処理の高速化，また，最良視点の算出など解決すべき問題も多く，今後の課題である．

また，複雑な仮想環境に対応するために，複数オブジェクトを 1 つのオブジェクトとして扱う手法が考えられる．提案手法は，多くのオブジェクトが存在する複雑な仮想環境では擬似画像の生成に必要な画像枚数が増大してしまう．そこで，複数オブジェクトを 1 つのオブジェクトとして扱うことで必要画像の枚数を削減し，複雑な仮想環境を構築することができると考えられる．

5.　おわりに

本稿では，サーバレンダリングを用い，ウェブブラウザ上でインタラクティブな仮想環境を実現する手法を提案した．また，サーバ側の負荷軽減のため CSS を用いた擬似画像の提示による画像生成回数の削減手法を提案した．さらに，実際に本手法を適用したシステムを構築し，応答時間の測定を行うことでリアルタイムな応答が可能であることを確認した．

今後は，広大でポリゴン数の多い仮想環境への対応と，実際のネットワークや様々なクライアントを用いた提案手法の評価を行う．また，サーバレンダリングにおける LOD の導入や，要求視点位置の予測手法など，さらなる負荷軽減手法を検討する予定である．

参考文献

[1] Second Life: http://jp.secondlife.com/; Linden Research, Inc. (2003).
[2] Lively: http://www.lively.com/; Google (2008).
[3] FINAL FANTASY XI: http://www.playonline.com/ff11/index.shtml SQUARE ENIX (2002).
[4] Boukerche, A., Nelem Pazzi, R. W.: Scheduling and buffering mechanisms for remote rendering streaming in virtual walkthrough class of applications; In Proceedings of WMuNeP, pp.53-60 (2006).
[5] Boukerche, A., Nelem Pazzi, R. W.: Remote rendering and streaming of progressive panoramas for mobile devices; In Proceedings of MULTIMEDIA, pp.691-694 (2006).
[6] Shade, J., Salesin ,D. H., DeRose, T., Snyder, J.: Hierarchical image caching for accelerated walkthroughs of complex environments; In Computer Graphics Proceedings, Annual Conference Series, pp.75-82 (1996).
[7] Dobbyn, S., Hamill, J., O'Conor, K., O'Sullivan, C.: Geopostors: a real-time geometry impostor crowd rendering system; In Proceedings of SI3D, pp.95-102 (2005).
[8] Jeschke, S., Wimmer, M., Schumann, H., Purgathofer, W.: Automatic impostor placement for guaranteed frame rates and low memory requirements; In Proceedings of SI3D, pp.103-110 (2005).
[9] Tan, S. D., Robertson, G. G., Czwerwinski, M.: Exploring 3d navigation: combining speed-coupled flying with orbiting; In Proceedings of SIGCHI, pp.418-425 (2001).
[10] Shneiderman, B., Plaisant, C.: Designing the user interface fourth Edition strategies for effective human-computer interaction, Addison Wesley Computing (2004).
[11] Vazquez, P. P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy In Proceedings of VMV, pp.273-280 (2001).