タイトル	Three.js ライブラリによる全天球画像の疑似立体視表示
著者	菊地, 慶仁; Kikuchi, Yoshihito; 間野, 絢也; Mano, Jun'ya
引用	工学研究 : 北海学園大学大学院工学研究科紀要(17): 39-44
発行日	2017-09-30

研究論文

Three.js ライブラリによる全天球画像の疑似立体視表示

菊 地 慶 仁* ・ 間 野 絢 也**

Pseudo stereoscopic display of omnidirectional image utilizing “Three.js” library

Yoshihito Kikuchi* and Jun’ya Mano**

要 旨（Abstract）

近年，撮影者の全周を撮影可能な全天球カメラが商品化され，個人の日常生活でも活用されている．撮影された画像は全天球画像と呼ばれ Equirectangular 形式で保存される．この形式は地図における正距円筒図法と同じ原理で，360 度の周囲を球として考えると赤道付近の画像は歪みなく表現できるが，球の極に当たる画像の上下限の端の部分が地図の上下の辺に拡張されてしまう問題点がある．このため画像を見るためにスマートフォンなどで専用のビューアを用いることが一般的となっている．この方式では利便性に制限があるため，本研究では情報の閲覧と公開に広く用いられているウェブブラウザを用いて表示することを試みる．また表示画面を 2 面に分けて疑似立体表示を試みた内容についても報告を行う．

1．全天球カメラとファイル保存形式の概要

1.1 全天球カメラ RICOH THETA について

ここでは，全天球カメラの量産モデルとして 2014 年 11 月に出荷が始まった RICOH 製 THETA[1] を中心に紹介する．

図 1 に本研究で用いた ROCOH THETA m15 の正面図と背面図を示す．THETA は，筐体上部の前後にある魚眼レンズと前面のシャッターのみを持つ．各魚眼レンズが撮影できる画像は 1024×1024 ドットサイズで，本体に保存される際には前後のカメラによる撮影画像ファイルを結合して 1024×2048 ドットサイズ一枚の画像に合成される．

撮影された画像を保存するために本体には 4 GByte の記憶容量がある．また THETA は Wifi ステーションとして機能できるのでスマートフォンやタブレットに Wifi 経由で画像ファイルや動画を保存することができる．

1.2 全天球画像の構造及び表示方法

図 2 に RICOH THETA m15 で撮影した全天球画像の例を示す．この図は前後面二基の魚眼レンズで撮影した生の画像そのままで Dual fisheye（両眼魚眼）形式と呼ばれる．THETA で撮影した動画ファイルなどは，この形式のままで動画として保存される．

次に Equirectangular（正距円筒図法）形式を図 3 に示す．これは，魚眼レンズで撮影した画像の外周部分を地図の赤道に，画像の中心部分を地図の極に写像し，さらに極の部分は，本来は一点なのだが，これが直線に拡張されている．このために地図の経線に相当する直線は本来は集中して極の一点で交差するのだが，Equilectugular 形式では上下に平行な直線となり極に相当する画面上下の辺の直線に等間隔に交差することになる．従って極部分の図形は大きく歪んでしまうので本来どのような画像が撮影されているかを直観的に認識することが難しくなる．

* 北海学園大学大学院工学研究科電子情報生命工学専攻
Graduate School of Engineering (Electronics, Information, and Life Science Eng.), Hokkai-Gakuen University
** HIS ホールディングス株式会社（北海学園大学工学部電子情報工学科卒）
HIS HOLDINGS, INC. (Graduated from Hokkai-Gakuen University)

図 1　ROCOH THETA 本体装置図（左正面図，右背面図）[1)]

図 2 と図 3 はほぼ同時に撮影しているので 2 種類の画像特徴を比較することができる．図 2 の撮影者の手が画像の下端全体に広がってしまっており，この部分だけを見ると手として認識することは難しくなっている．

1.3　全天球画像の表示方法と問題点

図 4 にスマートフォン上での THETA の基本アプリケーションのスクリーンショットを示す．THETA 本体にもシャッターはあるが，シャッター優先/ISO 優先/オートなどの露出設定やインターバル撮影など多様な設定を本体のみで行うことは不可能なので，スマートフォン上の上記アプリケーションで設定し撮影する必要がある．また撮影した画像の閲覧も本体単独では不可能なのでスマートフォンに転送した後に基本ソフトで閲覧する必要がある．図 4 に示されている画像は通常カメラのファインダーのように THETA のカメラからの画像をスマートフォンに示しているが，ここでは Equilectangular 形式のままで表示している．

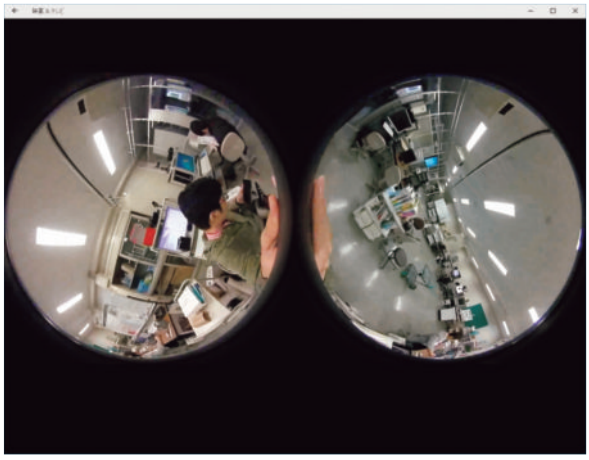

図 2　Dual fisheye 形式画像例

スマートフォン及びパソコンで通常閲覧を行うには専用のビューアーソフトを用いて行う．このソフトでは図 5 に示すように Equilectangular 画像を球面に写像し，球の中心部分に視点を置いて球面の一部をスマートフォンのスクリーンにレンダリングすることで行っている．このようにすると，描画できる領域は一部となるが極地点をレンダリングしても画像が歪むことが無く，どのような対象が写っているか正確に認識することができる．

Equilectangular 形式画像の表示は前述のように PC もしくはスマートフォンなどの専用のアプ

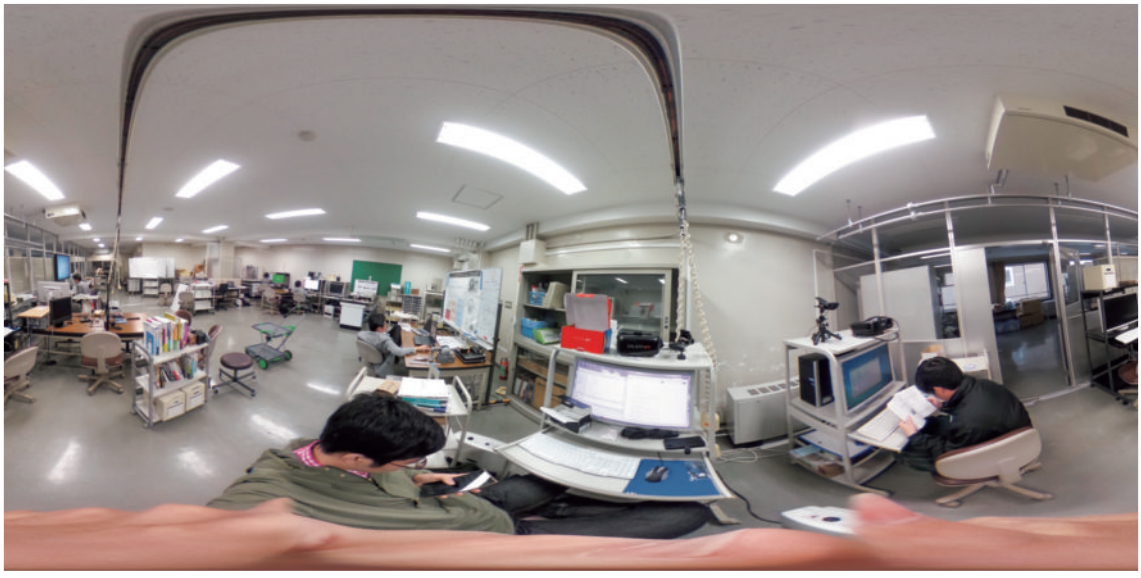

図 3　Equilectangular 形式画像例

図 4　スマートフォン向け RICOH THETA 操作用標準ソフト[1)]

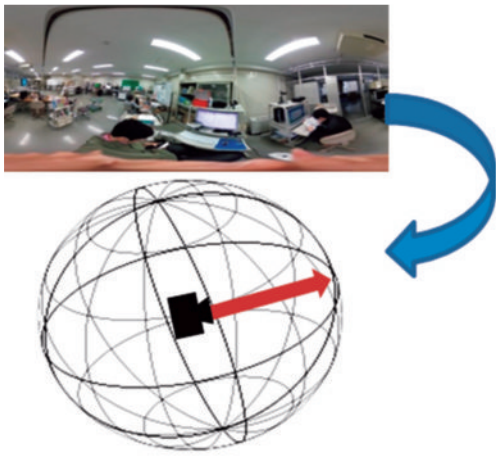

図 5　Equilectangular 形式画像を球オブジェクトに写像し内部に視点を置いたレンダリング手法

リケーションを用いることが一般的である．しかしながら，インターネット上に存在するサイトをブラウザで閲覧する形で撮影した画像の内容を公開することは今後も継続的に行われていくと考えられる．そこで本研究では，表示の原理は，これまで解説した専用アプリケーション上での，Equilectangular 形式画像を球面に写像し一部をブラウザ上にレンダリングする点は同じとし，PC もしくはスマートフォン上の一般的なブラウザで表示することを目的とする．

またこの際にブラウザ上での立体視表示も試みる．近年は図 6 のような，中にスマートフォンを組み込んで簡易的な VR ビューアとして利用するためのキットが発売されている．通常のシステムでは CG 画像を 360 度全周でレンダリングし，その一部を VR ヘッドマウントディスプレイに出力して立体視を得ている．Equilectangular 形式画像から右目左目用に 2 つの画像を生成できれば非常に簡便に疑似的な立体視を得ることが可能と考えられるからである．

2．関連技術と本研究での課題

第二章ではブラウザ上でのプログラマブルな画像表示技術を述べ，本研究における課題をまとめる．

図 6　スマートフォンをマウントして用いる簡易 VR ヘッドマウントディスプレイ

図 8　立体視用左右画像をスマートフォンで表示し VR ヘッドセットに装着して立体視を確認

2.1　ブラウザ上での画像のプログラマブルな表示方法

ブラウザで表示されるページを記述する際には HTML と呼ばれるマークアップ言語が用いられる．現在は HTML5 と呼ばれる規格が最新となっている[2]．

HTML5 では旧来に比べて多数のタグが規格に導入されて文書構造の定義が明確に行われるようになっている．その中に canvas と呼ばれるタグがあり，図形描画を行う領域を宣言するために用いられる．ページ中に宣言された canvas エリア内部には単純に画像ファイルを表示することも可能であるが，ウェブページに付随している Javascript によって動的に図形描画を行うことが可能となっている．

次にウェブページ上でのプログラムとして Javascript を用いたライブラリが存在する．Javascript 自体は Java に似た構文を持つスクリプト言語であるが，現在は複雑なページ動作を表現するスクリプトを Javascript の基本言語要素のみを用いた開発は少なく，目的に応じたライブラリを読み込んで，そのライブラリ関数を呼び出してデータの定義や表示を行っている．ライブラリは 1 つの JavaScript ファイルとして提供されることが多い．

ライブラリは単純に画面の装飾として簡単な動作を行わせるレベルからポリゴンメッシュで構成された 3D 画像を表示し，ブラウザ上で対話的な 3D アプリケーションを構築することも可能となってきている．3D グラフィック表示は Windows に準拠しない世界では OpenGL[3] が用いられている．この OpenGL に準拠してプログラミングを行うための Javascript の関数群が WebGL[4] である．WebGL の実装形態としては WebGLStudio.js[5] や Three.js[6] などが存在する．

2.2　本研究の課題

これまで述べてきた Equilectangular 形式画像の表示方法やブラウザ上での図形描画に基づいて，本研究での課題は以下の通りと考えられる．

1）WebGL に対応する Javascript を用いて Equilectangular 形式画像を球面に写像しレンダリングを行うこと．

2）立体視を行うために，2 つの canvas エリアを設けて，それぞれのエリアで視点を左右にオフセットさせた 2 枚のレンダリング画像を生成させること．またスマートフォンの姿勢の変化を取り込んで描画する領域を変化させることで VR ヘッドセット装着者の姿勢の変化に対応した描画を行うこと．

3　本研究での開発

3.1　開発手法

今回の開発では，HTML5 で記述した Equilectangular 形式画像を描画するウェブページを作成することになる．描画エリアの構成はブラウザの背景色及び余白と最小限のみを記述し，Equilectangular 形式画像の描画に関するアルゴリズムは全て JavaScript で行った．

3.2　開発環境

HTML5 及び JavaScript のエディティングにはフリーの Web ページ開発ツールである AptanaStudio3[7] を用いた．また PC 上のブラウザではスマートフォン上での動作が確認出来ないため，なんらかの形で Web サーバーを使用する

必要があった．そこで，GitHub[8] が提供している静的な Web ページをホスティングするサービスである GitHub Pages を用いた．

3.3　全天球画像/動画の表示

Javascript プログラム中で描画を行う対象の canvas エリアを指定し，球体オブジェクトを用意する．次に無限ループでレンダリングを実行するプログラムとして用意する．球面に全天球画像/動画をレンダリングして視線の方向の領域を canvas エリアに描画する．このとき，視点を球体内部から外側に向くように設置する．

ステレオ表示を行うには，ブラウザ上の canvas エリアを画面の左右に配置する．次に Equilectangular 形式画像を写像した球体内に視点を 2 つ中心から等距離だけ左右にオフセットさせて配置する．そして左の視点からの画像を左目用画像表示の canvas エリアに，右側の視点からの画像を右目用画像表示の canvas エリアに表示するように指定する．

4．動作実験

4.1　実験結果

開発したウェブページを PC 上のブラウザで閲覧し Equilectangular 形式画像をレンダリングしていることを確認した．

次にステレオ表示させた全天球画像/動画をスマートフォンのブラウザ上に表示させた状態で VR 用ヘッドマウントディスプレイに装着し閲覧することで目的としていた立体視を得ることができた（図 7）．

4.2　問題点

今回は，装着者の首振りの見地とレンダリングの連動までは実現できなかった．

全天球動画に関しては，開始の動作が遅い，動画が再生できない等，幾つかの問題が見られた．動作が遅い問題に関しては，動画容量を減らすことにより改善出来る見込みがある．動画が動かない問題に関しては，スマートフォンに組み込まれているブラウザが動画の自動再生に対応していない可能性がある．

図 7　Equilectangular 形式画像から立体視用左右画像の生成

5．結論

本報告では以下の報告を行った．

1）Equilectangular 形式画像をウェブブラウザで閲覧することを目的として，Javascript で記述したアルゴリズムを用いて HTML5 形式で記述されたウェブページに描画する方式を検討した．

2）検討した方式を実際に実装し，動作を確認した．

3）動画の再生については処理速度及びスマートフォンブラウザの動画再生能力によって制限を受けることが判った．

今後の課題としては，Dualfisheye 形式の動画をそのままブラウザでレンダリング処理する点がある．また後継機種の RICHO THETA S にはライブストリーミング機能があるので，得られる動画ストリームをステレオ表示することなどが考えられる．

参考文献

1）RICOH THETA 公式サイト
https://theta360.com/ja/
2）HTML5.1 公式サイト
https://www.w3.org/TR/html51/
3）OpenGL 公式サイト
https://www.opengl.org/
4）WebGL 公式サイト
https://www.khronos.org/registry/webgl/specs/latest/

2.0/
5）WebGLstudio 公式サイト
http://webglstudio.org/
6）Threejs 公式サイト
https://threejs.org/
7）Aptanastudio 公式サイト
http://www.aptana.com/
8）GitHub 公式サイト
http://www.github.com