



没入球面ディスプレイの設計開発を支援する 投影シミュレータ

橋本 渉, 水谷 泰治, 西口 敏司

大阪工業大学 情報科学部 (〒573-0196 大阪府枚方市北山 1-79-1, wataru.hashimoto@oit.ac.jp)

概要: 本研究は球面型没入ディスプレイ構築をサポートするシミュレータを開発することを目的としている。球面ディスプレイを構築するには、ドームスクリーンへの特殊な歪み補正を考慮した投影系の光学設計を行う必要がある。しかし、実際に製作される光学系はシミュレーション通りの精度が保証されるわけではない。使用する際に改めて光学系の微調整が必要となる。本研究では、投影系の光学設計と同時に、光学系の調整や歪み補正が実行可能な投影シミュレータを開発したので報告する。

キーワード: 球面スクリーン, 光学設計, 調整, 歪み補正

1. はじめに

没入型映像システムとは、あたかも体験者が映像空間に存在するかのような臨場感を与える映像システムの総称である。昨今、HMDを中心とした没入映像システムが活況を呈しているが、人間の周囲をスクリーンで覆ってしまう、いわゆる没入型ディスプレイも没入型映像システムの一つとして知られている。

没入型ディスプレイの大きな問題は、人間を映像で囲むという仕組み上、大掛かりになり容易に設置できない、という点である。設置容積を小さくするために、魚眼レンズや凸面鏡によりドーム状の曲面に映像を投影したり、超短焦点プロジェクタを用いて投影距離を短縮するなどの工夫がされている。しかし、それらの工夫を実現するためには、光学系のシミュレーションが必要となり、結局は容易に設置できないという問題に帰着する。また、それらに用いられる光学系は設置精度に繊細であるため、必ずしも設計通りの投影ができるわけではない。設置後には、正しい見え方になるような微調整が不可欠である。さらに、光学系による映像の歪みや、ずれによる映像の補償を表示コンテンツに適用しなければならない。

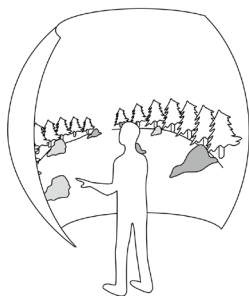


図 1: 没入球面ディスプレイの概念図

本研究の目的は、簡単に没入型ディスプレイを設計し、かつ容易に調整やコンテンツ生成ができる開発環境を構築することである。ここでは、没入型ディスプレイの中でも小規模に実現できる球面ディスプレイ(図 1)を対象とし、球面ディスプレイの開発フローを整理して、開発の障害となっている点を抽出する。そして、専門的知識や経験がなくても装置を構築できるようなシミュレータの開発を目指す。

2. 球面ディスプレイの特徴

球面ディスプレイの開発に際して、参考になるのがプラネタリウムである。ドーム投影は従来から確立されている技術であり、全天周に投影するためのリソース[1]やソフトウェア[2]が豊富である。最近ではプロジェクタ等の普及により DIY で製作している例もある。プラネタリウムは没入型球面ディスプレイの一種と言えるが、VR で利用される球面ディスプレイは以下の特徴を持つ。

・対話性

ユーザが映像とインタラクションすることを前提としている。したがって、投影される映像は予めレンダリングされたものではなく、即時生成しなければならない。

・リア投影

ユーザを完全に覆う場合や、ユーザの影などによる干渉を避ける場合には、スクリーンの裏面からリア投影することがある。

・視点位置の依存性

球面スクリーンの曲率半径が小さくなると、映像を歪みなく観察することのできるスポットが小さくなるため、理想視点の位置の依存性が高くなる。

上記の特徴を踏まえて、球面ディスプレイの開発フローを整理する。

3. 球面ディスプレイの開発フロー

球面スクリーンの開発の流れを大まかに分類すると、以下の4つのフローに分けられる。

(1). 光学設計

光学シミュレーション等により、スクリーンやプロジェクタ、反射鏡などの大きさや形状、配置、リア・フロントの投影方法などを決定する。設計のポイントは、プロジェクタによる投影画像がスクリーン面に合焦しているか、プロジェクタの光がユーザや構造物に遮られているか、プロジェクタの画素が有効に利用できているか、である。プロジェクタを複数配置する場合は、映像の継ぎ目部分を考慮しなければならない。設計の結果に基づいて、コンテンツ生成で使用される映像の歪み補償のための補正テーブルが取得される。

(2). 試作

光学設計の結果に基づいて球面スクリーンを構築する。最も製作が困難なものは球面状のスクリーンである。曲率が大きいものは FRP 成型や発泡スチロールの削り出しによる特注となる。

(3). 調整

試作されたディスプレイには誤差がつきものである。とくに光学系ではわずかな角度のずれで大きく光軸がずれてしまうため、どうしても設計通りの結果が得られない。そこで、設計時に得られた補正テーブルを表示し、誤差によるずれを修正する作業が必要となる。たとえば図2は光学設計によって得られた補正テーブル(緯線・経線)を表示しているが、左側の写真では明らかにずれていることがわかる。そのため、スクリーンやプロジェクタを物理的に動かしたり、補正テーブルを画面上で調整し、右側の状態に近づける必要がある。外部から投影状態をカメラで観察し、自動的に調整する方法もある。

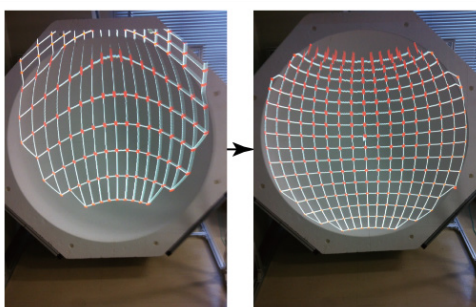


図2:補正テーブルの調整(左:調整前,右:調整後)

(4). コンテンツ生成

球面に正しい見えの映像を投影するためには、あらかじめ歪んだ映像を生成する必要がある。一般的には、全方位の映像を一度レンダリングし、その映像に歪みを加えて再度レンダリングする手法がとられる。調整で得られた補正テーブルに基づいて、歪み映像を生成する。Unityなどのゲームエンジンを用いた生成方法も提案されている[3]。

以上の通り、(1)で設計された装置は、製作や組み立てに伴う誤差によって(3)による調整作業が不可欠になる。補正テーブルの点の数が増えれば増えるほど、歪み補正の精度が高まるが、点の数に比例して調整に手間がかかる。特にディスプレイを頻繁に展示等に持ち出す場合は、毎回この調整作業が必要となる。

本研究では、この調整作業を軽減するための工夫として、光学設計と調整を同時に行うことができるシミュレータを提案する。具体的には、調整の段階で配置に関するパラメータを微調整し、歪み補正結果を確認しながら微調整することを目指す。配置に関するパラメータとは、プロジェクタやスクリーン、反射鏡などの3次元的な位置姿勢などである。要するに、実際に表示されている歪み補正の状態から、誤差を含めた本来のパラメータを推測することを意味する。実際の配置状態が推測できれば、補正テーブルの点を任意に増加させることも容易である。

4. シミュレータの設計

シミュレータを設計するにあたり、全方位映像の投影における歪み補正について整理しておく。全方位映像の投影には、魚眼レンズの投影範囲をカバーできるドームマスターフォーマット、Ricoh Thetaのような360度画像を世界地図のように長方形に引き延ばす正距円筒図法などがあるが、ここではレンダリングが容易で、調整時の結果表示にも活用できるcube-mapを用いることにする。図3に示す通り、cube-mapではユーザの全方位を立方体で覆い隠すことになるので、全6方位の映像をレンダリングし、テクスチャマッピング等を用いて球面状に貼りつければよいことになる。

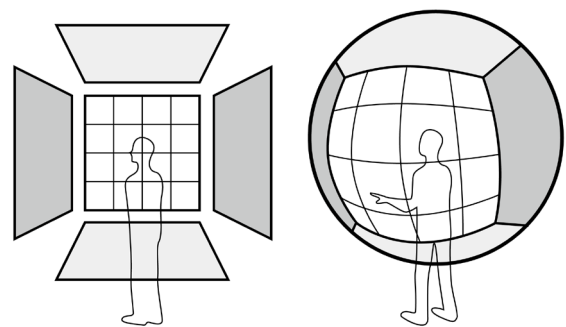


図3:cube-mapによる全方位映像レンダリング

cube-map映像を球面スクリーンに投影するためには、図3の右図のような歪み補正が必要となる。これを実現するため、図4に示すユーザの理想視点Vからcube-mapの特徴点Mの延長上にあるSと、プロジェクタから投影される光線PIとの関係を求め、PIとSの関係の集合に基づいて歪み補正をおこなうことになる。

図4の場合、PIは凸面鏡による反射を経て投影面Sに到達している。凸面鏡の位置や曲率が与えられれば、PからCの反射を経てSを求めることができる。しかしながら、

SとPがわかっている状態で、PIとCとの交点を求めるのは簡単ではない。言い換えると、プロジェクタの光線を順方向に追跡するのは容易であるが、逆算するのは困難である、ということである。なお、逆算が困難なのは曲面反射のようなケースであり、単純な平面反射などの光学系であれば、逆算することは不可能ではない。

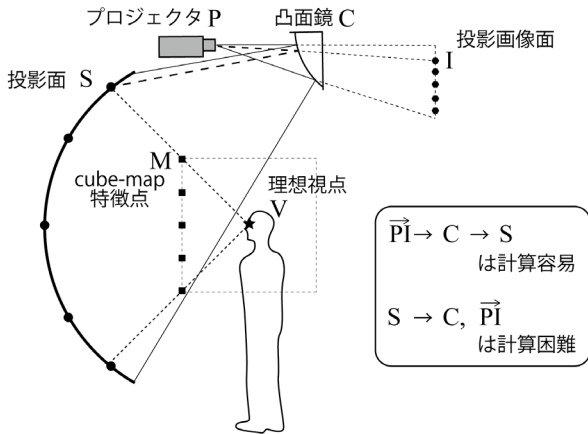


図 4: Cube-map の特徴点に対応するプロジェクタ-投影画像面上の点を算出する

先に述べたように、歪み補正に必要な情報はプロジェクタの投影画像面と、cube-map 等による特徴点の対応関係である。投影シミュレータにおいて、前述のように特徴点から投影画像面が逆算することができればよいが、逆計算できない場合は、別の方法を考えなければならない。

ここでは、投影面に表示される有効な画素すべてを順方向 (PI→C→S) で計算しておき、VM の延長上にある近似点を利用することにした。これらの光線追跡計算はすべて互いに独立であるため、並列化による処理高速化が容易である。しかし、VM の延長上にある S の近似点を求めるために、投影面に表示される有効な画素全てから全点探索をおこない、最も近い点を求めると、cube-map の点が増えた場合に探索回数が増加するため処理時間を要する。投影シミュレータの応答性が悪くなれば、パラメータ調整して即座に補正結果を確認することができず、調整に時間を要してしまう。最も近い点の探索回数を減らすために、探索アルゴリズムとして山登り法を適用し、全点探索を避けることにする。

以上の方法により、光学設計のパラメータを微調整しながら、歪み補正に必要な対応点の探索処理をできるだけ短縮し、補正結果を確認できるシミュレータを実現する。

5. シミュレータの実装

作成した投影シミュレータを図 5 に示す。スクリーンやプロジェクタ、反射鏡の 3 次元的な位置姿勢に関するパラメータは右上の 2 つのウィンドウで操作可能である。左上のウィンドウで投影状態を確認し、左下のウィンドウで歪み補正映像をレンダリングする。左下のウィンドウを最大

化しプロジェクタから出力することで、パラメータを変更しながら実際の投影状態を確認することが可能となる。歪み補正画像で出力されている特徴点は cube-map の配置となっているが、cube-map が鉛直軸に 45 度回転した状態となっている。本来なら半球をカバーするため図 3 のように 5 面レンダリングする必要があるが、cube-map を回転させることでレンダリングを 4 面に低減させるためである。

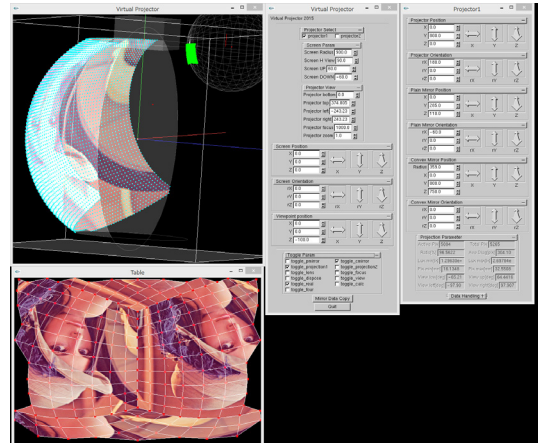


図 5: 作成した投影シミュレータ

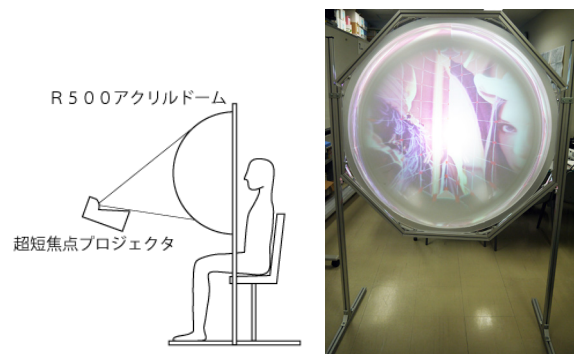


図 6: シミュレータ試用環境：直径 1000mm リア投影ドーム環境 (左) と投影状態 (右)

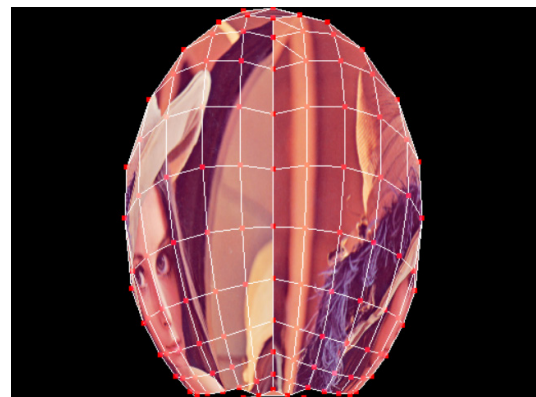


図 7: 歪み補正テーブルに基づいて生成された補正画像

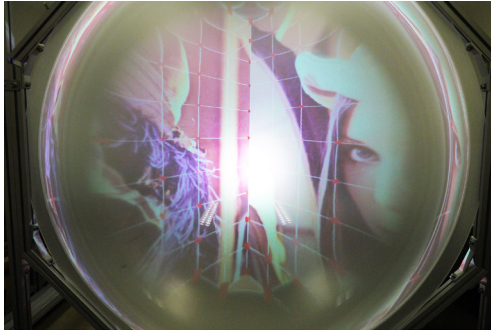


図8: 図7の補正画像を実際に投影している様子
(理想視点の位置より約1m後方で撮影)

実際に投影シミュレータを球面ディスプレイに試用し、調整による補正を試みた例を示す。図6は直径1000mmの亚克力ドームスクリーンで、背面から超短焦点プロジェクタ (NEC NP-UM330XJL-N2) により投影している環境である。これらのプロジェクタやスクリーンに関する不変パラメータを入力し、20分程度の微調整の末に得られた補正テーブルが図7のとおりである。cube-mapによる特徴点の合計数は102点である。実際にスクリーンに投影したものが図8である。キューブマップの左右の面が歪みのない状態が表示されているのが確認できる。

全点の光線追跡と対応関係の探索計算に要する合計時間は全点探索で7.3秒、OpenMPによる最適化された全点探索で6.5秒、山登り法とOpenMPの併用で0.14秒であった。パラメータを変更すると0.14秒後に補正テーブルが生成され、補正結果が画面に反映されることになる。若干

の遅延が感じられるものの、ほぼ実時間で結果を反映できていることがわかる。

6. まとめ

球面ディスプレイの光学設計と調整を同時に実行できるシミュレータを開発した。このシミュレータの特徴は、投影状態を観察しながらパラメータを変更することで、現実の配置状態を近似的に求めることができ、補正に必要な特徴点を任意に増やすことのできる点である。得られた補正テーブルは現時点ではOBJ形式により出力し、Unity等のゲームエンジンにおいてコンテンツ生成ができるようにしている。今後は、専門知識がないユーザでも球面ディスプレイを開発できるよう、シミュレータのインターフェースに改良を加えることが必要である。

謝辞 本研究はJSPS科研費JP15K00290の助成を受けたものです。We also appreciate Prof. Corchado and Dr. De la Prieta in University of Salamanca for assisting with this research.

参考文献

- [1] IMERSA, 2016, <http://imersa.org/>
- [2] (株)オリハルコンテクノロジーズ, 2016, <http://orihalcon.co.jp/>
- [3] Paul Bourke: Dome: Immersive gaming with the Unity game engine, Proc of CGAT09, pp 265-272, 2009