

3次元仮想空間を用いたズーミング可能な ステレオフォトビューア

広内 哲夫

A stereo-photographic viewer with which zooming can be made by using 3-dimensional virtual space

Tetsuo Hirouchi

Abstract

In this paper, proposal is made with a new method concerning a stereo-photographic-viewer allowing stereographic-photographs to be viewed by executing zooming with the use of 3DCG virtual space. Furthermore SPV-F realized based on the said method is introduced.

Just exclusive use of virtual space is liable to cause minute slippage in the left and right images (stereo-pairs) of stereoscopic-photography to be expanded during zooming-in. This results in failure of stereoscopic vision. To avoid this trouble, "a fusion equation" was introduced based on stereoscopic view in the virtual space.

SPV-F fabricated by using the fusion equation referred to above can beautifully generate stereoscopic images in a fine region without being accompanied by appearance of double images, even when zooming-in is made to the extent of resolution of the stereoscopic photography.

1. はじめに

最近の立体視ブームで、情報機器や娯楽関連機器の分野(パソコン、カメラ、テレビ、ゲーム機、携帯電話、映画)では、多くの立体視技術が開発され、実用に供され始めた。そこで筆者も、立体写真をズーミングしながら閲覧できる SPV と呼ばれるビューアソフトを開発した。SPV においては、立体写真のズーミングを制御する「融像式」と呼ばれる数学的手続きが2次元グラフィックスである Java2D を用いて実現されたのである。その内容はすでに参考文献 1 に報告している(文献1)。

しかし、筆者は SPV を開発する以前に、3次元グラフィックスの仮想空間に基づく融像式を考案し、この融像式を搭載した試行的なビューアソフトを開発した。本論文ではそれを SPV-F と呼ぶことにする。この SPV-F は、VRML 言語、JavaScript 言語および HTML 言語を基盤として実現されたものである。3次元版の SPV-F は、その2次元版である SPV と同様に、ステレオフォトビューア(注→次ページ)と呼ばれる概念的なビューアに基づく、立体写真をズーミングしながら閲覧できる立体写真閲覧ソフトである。

現在実用に供されている SPV(2次元版)は、SPV-F(3次元版)におけるステレオフォトビューア

の概念を原型としている。そこで本論文では、3次元グラフィックスの仮想空間のもとでの融像式の導出の仕方と、それを搭載したSPV-Fを紹介することによって、「3次元グラフィックスにおける融像式は2次元グラフィックスのそれと原理的に同じである」ことを示すことにする。本論文においては、その基本となるステレオフォトビューアを説明する必要性から、SPVを紹介した参考文献1の内容と一部が重複している箇所(第2章、第5章)がある。また、説明を簡略化した箇所(第8章)もあるので、必要に応じて参考文献1を参照して頂きたい。

2. ステレオフォトビューアが対応すべき課題

本論に入る前に、立体写真を立体視する場合の問題点と、最適な遠近的立体感を実現するステレオフォトビューアが具備すべき機能を指摘しておく。

2.1 課題1 –ズームイン操作による2重像の発生への対応

立体写真を単にズームインしていくと、画面に投影された左右画像のズレは拡大していく。そして、倍率が一定以上の値となると、左右画像のズレが閲覧者の立体視能力の限界を超え、画面に2重像が現れ、閲覧者にとって立体像が融像しないという現象が生じる。このような状況に対応するために、ステレオフォトビューアに導入する機能には、画面の倍率が拡大する過程において、ズレの拡大を抑制して2重像の発生を防止する機能を持たせる必要がある。

2.2 課題2 –ズームインなのに遠ざかる立体像への対応

ズームイン操作を行っている場合、本来は近づくべき立体像が徐々に遠ざかっていく、あるいはズームアウト操作を行っている場合、遠ざかるべき立体像が徐々に近づいてくる、といった現象がみられる。立体写真の光景内容によっては、閲覧者が注意深く観察しないと視認できない場合もある。しかし、このような立体像の動きは、人々の感覚からすると不自然である。この原因は、左右の画像の間でのズレの大きさが、ズームインの場合には拡大する、あるいはズームアウトの場合には縮小することに起因する。このような現象を解消しないと、立体写真のズームインは不自然になってしまう。このような防止機能も持たせる必要がある。

2.3 課題3 –強い飛び出し効果への対応

立体写真と言えば、かつては飛び出し効果を狙ったものが多かったが、立体ディスプレイ装置を用いて立体写真を眺めるような時代になると、画面よりも後方に引っ込んだ奥行き感のある、眼に優しい立体写真が好まれる。しかし、通常の立体写真閲覧ソフトを用いて立体ディスプレイ上に立体像を再生する場合、2台のカメラのレンズ間隔は適切であっても、立体像全体が画面から飛び出しているように強く感じられ、閲覧者の目に負担を強いることになる。これが極端になると、立体像は2重像となる。このような場合、ステレオフォトビューアに導入する機能には、遠近的立体感

(注→前ページ) 一般に立体写真を単にズームインする場合、立体ディスプレイ装置上の微細な領域をズームインする過程で、①融像していた立体像が2つの画像 – 2重像へと分裂し、閲覧者が立体像を融像できない現象、②立体像が手前に迫って来ないで後退してしまう現象がしばしば起こる。この現象を防ぐ数学的な手続きが、立体像の融像を解析することによって導出された融像式である。この融像式を持つ概念的なビューアを、筆者はステレオフォトビューアと呼ぶ。

が適度な奥行き感覚になるように、立体像全体に渡って飛び出し感を抑える機能を持たせる必要がある。

3. 一般の 3DCG が具備する基本機能

ステレオフォトビューアを開発するには、3DCG の 3次元仮想空間を使用するので、一般の 3DCG が持っている機能を説明しておく。3DCG 関係の応用ソフトは一般には、以下のような組み合わせによって開発されることが多い。

- ① OpenGL-API と C / C++ 言語の組み合わせ
- ② Java3D-API と Java 言語の組み合わせ
- ③ VRML 言語と JavaScript 言語と HTML 言語の組み合わせ

上記①と②では C / C++ 言語と Java 言語がそれぞれの API の親言語になるが、③では JavaScript 言語と HTML 言語が強いていえば VRML 言語の親言語に対応するといえる。なお、今回開発したステレオフォトビューアに基づく SPV-F は、VRML 言語と JavaScript 言語と HTML 言語の組み合わせで開発している。それについては第 10 章で述べる。

これらの OpenGL-API、Java3D-API、および VRML 言語は、3DCG の基本ソフトであるがゆえに、共通した機能を持っている。ステレオフォトビューアを開発するために必要となる共通の機能とは、以下の通りである (文献², ³)。

- ① 画像(テクスチャーと呼ばれる)を仮想物体に貼り付ける機能
- ② 仮想空間を動き回るウォークスルー機能
- ③ 視点(仮想カメラ)の設定
- ④ 仮想物体を回転、平行移動、拡大縮小する座標変換機能(アフィン変換機能)
- ⑤ アニメーション機能

ソフトウェア開発の立場からは、ステレオフォトビューアは一般の画像処理技術等を用いても開発することは可能である。しかし、①の以外の機能を自前で作る(用意する)ことは、かなり厄介なことである。特に②と③の機能については、一般の画像処理技術等には仮想空間の概念がないため、この機能を作るのは厄介である。⑤の機能については、フォトフレームのように自動閲覧や自動ズームを行う場合に利用される機能である。3DCG の基本ソフトには、これらの機能が用意されているので、ステレオフォトビューアの設計には、3DCG 技術を応用することにしたのである。

4. ステレオフォトビューアにおける仮想空間

4. 1 仮想空間の構成

最初に 3DCG が作り出す仮想空間の適用から述べよう。図 1 は、ステレオフォトビューアが構成する 3次元仮想空間を斜め上から眺めたものである。そこには仮想物体である 2枚の薄い平板が重なるように存在し、2枚の平板には撮影用の 2台のカメラで撮られた 2枚の画像(ステレオペア)が 3DCG のテクスチャー機能によってそれぞれ貼られている。

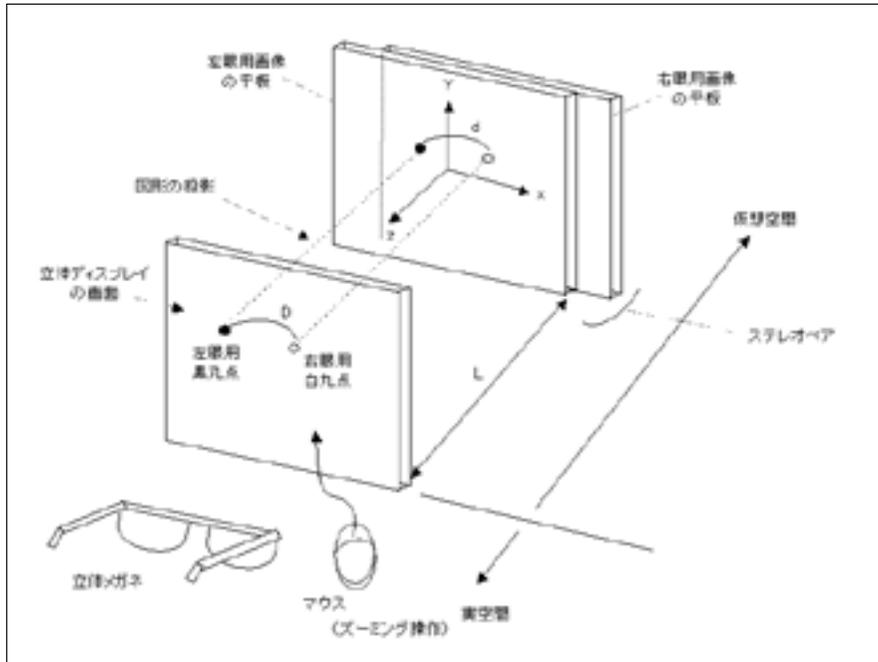


図1 ステレオフォトビューアの仮想空間の構成

図1中では分かり易くするため2枚の平板が前後に分けて描かれているが、実際には2枚の平板はともに同じ微小な厚さであり、 x 軸上に沿って重ね合わされている。手前の平板には左眼用の画像、後ろの平板には右眼用の画像が貼られている。ただし、左カメラで撮影された画像を左眼用画像、右カメラで撮影された画像を右眼用画像に使用するものとする。そして左眼用画像の平板の中心は、仮想空間における座標系の原点に固定して据えられて、右眼用画像の平板は、上下、左右の方向に可動し、 z 軸を中心にプラス・マイナス方向に回転する。座標系は、右に向かって正方向の x 軸、上に向かって正方向の y 軸、原点から画面に向かって正方向の z 軸となっている。ディスプレイ画面からある適切な距離を離して仮想空間の原点が置かれる。

仮想空間にはアバターと呼ばれる閲覧者の代理人がいる。アバターとは、閲覧者に代わって仮想世界に入り込み、仮想世界を眺める者をいう。アバターは自分の左右の目に超小型の仮想カメラを装備していて、それによって仮想世界を眺めている。仮想カメラは互いに結合し運動し合い、その左右のレンズの光軸はともに z 軸に平行で、その向きは常に z 軸負の方向である。仮想カメラの左右のレンズを通して眺めた仮想世界の光景が閲覧者の眺めるディスプレイ画面に投影される。右画像を仮想カメラの右レンズで写し、左画像を左レンズで写すことになる。そして、左右の画像が、重ね合わされて立体ディスプレイ画面に投影される。アバターは座標系の z 軸正方向側に存在し、アバターの装着する仮想カメラの左右のレンズが、仮想世界を眺めるアバターの視点となる。アバターは左右の視点を持ち、その視点の間隔を w とする。

ディスプレイ画面の外側(画面から外に向かった z 軸の正方向)の実空間には閲覧者がいて、立体視を行う。立体装置がメガネ式の場合には立体メガネを装着して、また裸眼式の場合には裸眼のまま、左右の画像が重ね合わされて投影されたディスプレイ画面を、閲覧者は眺め、そこで立体像

を融像する。閲覧者は3DCGの機能であるマウスを用いて、アバターに仮想世界内部をウォークスルーさせることによって、立体写真(画像)の任意の箇所をズームしながら眺める。アバターの視点は、ディスプレイ画面で閲覧者が眺める画像の領域を決定する。以降、閲覧者の操作するアバターの視点を単に「視点」と呼ぶことにする。

閲覧者の仮想空間におけるズームの範囲は、ズームインして画像に接近できる限界の位置から、ズームアウトして画像から後退する限界の位置までの距離であり、この距離を「視点から画像までの距離」あるいは単に「距離」と呼び、その値を λ (ギリシャ文字のラムダである：図1ではLと示されている)で表す。 λ は仮想空間のz軸によって測られ、その起点は平板上の画像位置を零とする。

4.2 画像の画面への投影

ステレオペアを構成する2画像は透視投影されてディスプレイ画面上に重なるように映し出される。図1では、ステレオペアとしての左眼用画像(左画像)の特定箇所(図中では黒丸点で示された位置)と右眼用画像(右画像)のそれに対応する箇所(図中では白丸点で示された位置)が、丸点としてディスプレイ画面に投影されている。なお、左右画像には、光景の左右それぞれが対応する点同士が無数に存在するが、ここではそのうちの一組を取り上げている。平板に貼られた左右画像の丸点間の距離(これを本論文では「ズレの値」と呼ぶ)は d である。このディスプレイ画面上に透視投影された左右の丸点間の間隔は D である。なお、この間隔 D が遠近の立体感を生み出すのであるが、これについては後述する。

仮想世界を眺めるアバターの装着した超小型の仮想カメラ(視点)を介して左右の光景がそのまま、ディスプレイ画面に重ね合わされて映し出されるのであるが、ここで、仮想世界、仮想カメラ、ディスプレイ画面の関係を述べておく。3DCG技術は、アバターの視点を介して仮想空間の様子を仮想カメラのフィルム面(仮想画面)に投影し、そのフィルム面(仮想画面)に写し出された光景をそのままディスプレイ画面に表示する。その仕様は次の通りである。

仮想カメラが写す仮想空間の画像の範囲(領域)は、仮想カメラが置かれた位置から画像までの距離 λ とある関係が成り立つ。すなわち距離 λ が2倍になれば、フィルム面(仮想画面)に写る光景の領域は2倍(長さ比)である。すなわち、①距離 λ とフィルム面に映し出された左右の丸点間の間隔 D は、そのときのズレの値 d のもとで反比例関係にあり、この関係を保ったまま、仮想カメラの捉えた画像領域がディスプレイ画面に表示される。また、画像と仮想カメラの距離 λ が一定であれば、左右画像間におけるズレの値 d は、仮想カメラを介してディスプレイ画面に比例する大きさの間隔 D で投影される。すなわち、②画像のズレの値 d と画面上に映し出された左右の丸点間の間隔 D は、そのときの距離 λ のもとで比例関係にある。

上記の①と②の関係を以下にまとめておく。

- ① 仮想世界における距離 λ とディスプレイ画面上に投影された間隔 D は、ズレの値 d が一定のもとで反比例関係にある。
- ② 仮想世界における左右画像間のズレの値 d とディスプレイ画面上に投影された間隔 D は、距離 λ が一定のもとで比例関係にある。

この2つの関係を式で表わすと、以下のようになる。この式は、第6章で用いられる大切な式である。

$$D \propto d / \lambda \quad (\text{式1})$$

5. 融像の解析

5. 1 ズレの値 d とズレの間隔 D の座標系

遠近ファクターの役割を数学的な式として定める前に、融像の現象の解析を行う。図2は、図1を上部から y 軸の負の方向に向かって眺めたものである。平板に貼られた画像が、ディスプレイ画面に投影される。閲覧者はディスプレイ画面上の投影点の白丸点を右眼で、黒丸点を左眼で眺め、閲覧者の脳は、画面上のズレの間隔 D に基づいて立体像を融像するのである。

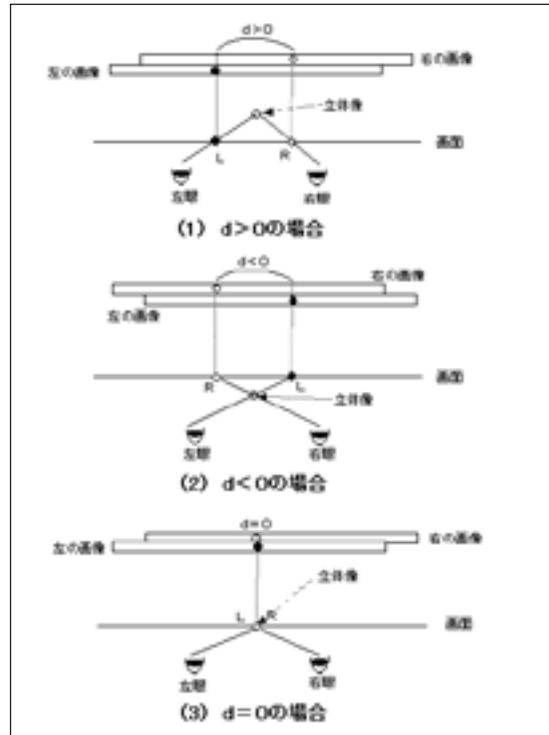


図2 左右画像から再生される立体像

「画像間のズレの値 d 」の座標系は、以下に説明する1次元座標系(x 軸)である。図2(1)に示すように左画像の黒丸点を原点(起点)として固定し、それに対応する右画像の白丸点との距離を画像間のズレの値 d とする。画像の白丸点が右側に移動すれば正の移動とし、そのズレの値 d は正值とする。また、白丸点が左側に移動すればズレの値 d は負の移動とし、そのズレの値 d は負値とする。従って、画像間のズレの値 d は正、零、負の値を取る。

「画面上のズレの間隔 D 」の座標系は、「画像間のズレの値 d 」の座標系と同じで1次元座標系(x 軸)ある。ズレの値 d は、ズレの間隔 D として投影される。4.2節で述べたように、 D と d は比例関係にあるので、ズレの間隔 D の座標系は、投影された黒丸点を原点とした1次元座標系(x 軸)であり、その正負の方向はズレの値 d の x 軸座標と同じである。

立体像の融像する位置の座標系は、画面に垂直の1次元座標系(u 軸)であり、画面の中央を原点

として、画面奥を正方向とする。図2においては、紙面上部に向かう方向がu軸の正方向となる。

5. 2 融像する立体像の位置

「画面上のズレの間隔Dが正の値として増大すると、立体像は画面奥の正方向に後退し、また間隔Dが負の値としてその絶対値が増大すると、立体像は画面手前の負方向に前進する」という現象が知られている。この現象に基づいて、閲覧者によって視認される立体像の位置を定性的に説明しよう。

図2(1)に示すように、画像間のズレの値dが正の値とし、それを増加するように右画像をx軸正方向に移動させると、画面上に投影された白丸点は同じくx軸正方向に移動するので、画面上のズレの間隔Dは正の値として増大する。そのため、立体像の位置はu軸正方向に移動するので、閲覧者は立体像が後退するように融像する。また、図2(2)に示すように、画像間のズレの値dが負の値とし、その絶対値を増大するように右画像をx軸負方向に移動させると、画面上に投影された白丸点は同じくx軸負方向に移動するので、画面上のズレの間隔Dは負の値としてその絶対値が増大する。そのため、立体像の位置はu軸負方向に移動するので、閲覧者は立体像が近づくように融像する。また、図2(3)に示すように、左右の画像の黒丸点と白丸点がズレなくある一点で重なった場合は、画像間のズレの値dは0なので、画面上のズレの間隔Dは0の値となる。そのため、融像する立体像の位置はu軸の零となるので、閲覧者は、画面上の位置に立体像が存在するように感じる。このような立体像の動きは、図2の作図からの分析で明らかであるが、その数式の取り扱いには6.6節で触れる。

ここに示したように、画像間のズレの値dを増大・減少させることによって、立体像は後退・前進をするので、この現象を利用して、立体像の動きを制御する数学的な式を組み立てることを、以下に試みる。

6. 融像式の導出

以上のことを踏まえ、4.2節で示した「距離 λ 」、「画像間のズレの値d」、および「画面上のズレの間隔D」の間で成り立つ式1を用いて、本論文の主題である3次元仮想空間を用いた場合の融像式を導くことにする。その最初の導出は発見的に行われたので、それに沿って段階的に行うことにする。

6. 1 前提

まず、ズームインあるいはズームアウトを行っている間、立体像を特定の位置に停止しているように視認させる方法を考えよう。それには、図2に示す画面上のズレの間隔Dを、ズームインあるいはズームアウトによって視点から画像までの距離 λ が変化しても、一定の値に保つ必要がある。4.2節で導いた式1を用いると、この間隔Dを距離 λ に関わらず一定にするには、ズレの値dと距離 λ を比例の関係にすればよい。

間隔Dの一定値を k_1 とし、さらに定数 k_2 を導入すると、式1は以下のようにになる。

$$D = k_1 = k_2 \times d / \lambda \quad (k_2 > 0) \quad (\text{式2})$$

k_1 / k_2 を a として上式を書き直すと、以下の d と λ に関する比例式、

$$d = a \times \lambda \quad (a > 0) \quad (式3)$$

が得られる。式3を式2に代入すると、間隔 D は、以下のように距離 λ に関係なく一定値となる。

$$D = k_2 \times a \quad (式4)$$

なお、図2における左画像の黒丸点の位置は固定されているものとするが、この場合において a を正値とする理由は、立体像を画面の奥で融像させるためである。負値とすると、立体像全体が画面から飛び出して、一般に視認しにくくなる。逆に右画像の白丸点の位置を固定し、左画像の黒丸点を可動とした場合には、 a は負値とする。

6. 2 立体像の駆動

6. 1節の条件下では立体像は前進も後退もしない。そこで、ズームインの時に立体像が自然に閲覧者に向かって前進するために、ズレの値の式3を変形しよう。実用に供するためにも、画面上に投影された画像に関する2つの距離 λ を導入する。1つは視点が画像に最も接近する距離 λ_0 であり、もう1つは視点が画像から最も後退する距離 λ_1 である、前者を「接近限界距離」、後者を「後退限界距離」と呼ぶことにする。すると、ズームングにおける距離 λ の取り得る範囲は、以下の不等式で表わされる。

$$\lambda_0 \leq \lambda \leq \lambda_1 \quad (式5)$$

ところで、式3において、その距離 λ を、以下のように $(\lambda - \lambda_0)$ で置き換える。

$$d = a \times (\lambda - \lambda_0) \quad (式6)$$

この置き換えの狙いは、「距離 λ が接近限界距離 λ_0 のとき、式2の間隔 D を0として、図2(3)の状況を作り出すこと」である。それを確認しよう。式6を、式2に代入すると、

$$D = k_2 \times a \times (1 - \lambda_0 / \lambda) \quad (式7)$$

となる。距離 λ が λ_0 のとき、間隔 D は0となり、黒丸点と白丸点は画面上で重なり合うことになる。

ズームングにおいて距離 λ は λ_1 から λ_0 になるので、その変化に従って、間隔 D は、正の値から0に近づいていく。従ってズームインにおいては、式6が成立する限り、立体像は閲覧者に近づいてくる。ズームアウトにおいては、その逆になることは明らかである。

6. 3 前進/後退ファクターの導入

式7から、視点の画像からの距離 λ の変化によって、立体像は前進したり後退したりする。このような状態でも閲覧者には、画面全体における立体像の動きは自然な動きのように感じるかもしれ

ない。しかし、もう1つ改良を加えることにする。立体像の動きを強制的に変化させるために、以下のような「前進／後退ファクター」と呼ぶものを導入する。

$$f = \{(\lambda - \lambda_0) / (\lambda_1 - \lambda_0)\}^n \quad (\text{式 8})$$

このファクター f は、距離 λ が λ_0 の時は0の値、距離 λ が λ_1 の時は1の値とる関数である (λ が $\lambda_0 \leq \lambda \leq \lambda_1$ の範囲で、 f は $0 \leq f \leq 1$ の範囲にある)。冪乗の指数 n は、立体像の“前進／後退の勢い”を調節するものであり、 $0 \leq n \leq 2$ の実数値を取るものとする。前進／後退ファクター f を補正因子として式6と式7の定数 a に乗ずることによって、ズーミングによる立体像全体の前進と後退を自然な動きにするように制御できるのである。

6.4 ズレの式と間隔Dの式の導出

式6の定数 a に、補正因子として式8を作用させると、以下の「ズレの式」と呼ぶ式が導かれる。

$$\begin{aligned} d &= (a \times f) \times (\lambda - \lambda_0) \\ &= \{a / (\lambda_1 - \lambda_0)\}^n \times (\lambda - \lambda_0)^{n+1} \end{aligned} \quad (\text{式 9})$$

また式7の定数 a に、同様に補正因子として式8を作用させると、以下の間隔Dに関する式が導かれる。

$$\begin{aligned} D &= k_2 \times (a \times f) \times (1 - \lambda_0 / \lambda) \\ &= k_2 \times \{a / (\lambda_1 - \lambda_0)\}^n \\ &\quad \times \{(\lambda - \lambda_0)^n \times (1 - \lambda_0 / \lambda)\} \end{aligned} \quad (\text{式 10})$$

式10は距離 λ に関して単調増加関数である。

6.5 2重像防止定数と融像式の定式化

ズレの式(式9)を用いて、さらに次のような式を定める。

$$t = \text{「ズレの式」} + \text{「2重像防止定数」} \quad (\text{式 11})$$

具体的には、次の式である。

$$\begin{aligned} t &= d + b = (a \times f) \times (\lambda - \lambda_0) + b \\ &= \{a / (\lambda_1 - \lambda_0)\}^n \times (\lambda - \lambda_0)^{n+1} + b \end{aligned} \quad (\text{式 12})$$

この式は、ズレの式に「2重像防止定数」と呼ぶ定数 b を付加したもので、この式を「融像式」と呼ぶことにする。(この式は2重像の発生を防ぎ、閲覧者が最適で遠近的立体感のある立体像を融像するための式である。このことから“融像式”と命名した)。

2重像防止定数 b は、2重像の発生防止に関するパラメータである。定数 b は、式9のズレの

式 d を 0 (に近い値) とするとき、左画像用平板 (位置は固定されている) と右画像用平板との間でのズレの値である。定数 b の値を適切に定めれば、ズームイン限界においても 2 重像の発生なしに、立体視を可能とするが、その値は視覚実験から定める。

ズレの式 d から算定される値は、閲覧者が目視で確認できる画面上のズレの間隔 D に対応する、左右画像におけるズレの値であるが、定数 b の値は、画像間のズレの値 d を 0 としたときの、左画像用平板 (位置は固定されている) に対する右画像用平板そのものの絶対的なズレの値である。

6. 6 定数 a の役割 - 遠近感制御定数

閲覧者の視認する立体像の奥行き (融像位置) u は理論的に導かれるが、本節では天下り式にその式を示し (文献 4)、それを融像式の定数 a の役割についての説明に用いる。

$$u = \{D / (e - D)\} \times h \quad (e > D) \quad (\text{式 } 13)$$

ここで、 e は閲覧者の瞳孔間隔、 h は閲覧者とディスプレイ画面までの距離、 D は 6. 4 節で定めた間隔 D (式 10) である。 u の座標系は、画面の中心が原点で、画面奥に向かって正とする。

融像式 t (式 12) の定数 a は、立体像の遠近感を調整するパラメータに成り得るものである。というのは、視点から画像までの距離 λ が一定の場合、立体像の奥行きについて、その位置 u (式 13) は間隔 D に関して単調増加関数であり、間隔 D は定数 a に比例する。そして、定数 a は融像式 t に基づくものなので、その結果、立体像の奥行き u は融像式の定数 a に関する単調増加関数となる。このことから、融像式 t における定数 a の値が増加すると、立体像の奥行き u は増加し、定数 a の値が減少すると、立体像の奥行き u も減少することが分かる。このため定数 a を「遠近感制御定数」と呼ぶが、その値は視覚実験から定める。

7. 融像式の特徴

7. 1 融像式のグラフ表現

式 12 の融像式の様子をグラフで示そう。視点から画像までの距離 λ が接近限界距離 λ_0 の値をとるとき、定数 b は視覚実験から定まり、そのときの値を b_0 、融像式 t の値を t_0 とすると、

$$\begin{aligned} t_0 &= (a / (\lambda_1 - \lambda_0)^n) \times (\lambda_0 - \lambda_0)^{n+1} + b_0 \\ &= b_0 \end{aligned}$$

である。また、視点から画像までの距離 λ が後退限界距離 λ_1 の値をとるとき、定数 a は視覚実験から定まり、そのときの値を a_1 、融像式 t の値を t_1 とすると、

$$\begin{aligned} t_1 &= (a_1 / (\lambda_1 - \lambda_0)^n) \times (\lambda_1 - \lambda_0)^{n+1} + b_0 \\ &= a_1 \times (\lambda_1 - \lambda_0) + b_0 \end{aligned}$$

である。

接近限界距離 λ_0 、後退限界距離 λ_1 、および上記の t_0 、 t_1 を用いて、融像式の概念的な様子を図

3にグラフで示す(図中では λ はLと表記されている)。前進/後退ファクターの冪乗の指数 n が1である場合の融像式は、点 (λ_0, t_0) 、点 (λ_1, t_1) を通り、点 (λ_0, t_0) を頂点とする下向きに凸(トツ)の2次関数である。また、前進/後退ファクターの冪乗の指数 n が0である場合の融像式は、傾きが a_1 の直線(1次関数)である。なお、視点から画像までの距離 λ の取り得る範囲(ズーム可能な範囲)は、式5から、 $\lambda_0 \leq \lambda \leq \lambda_1$ である。

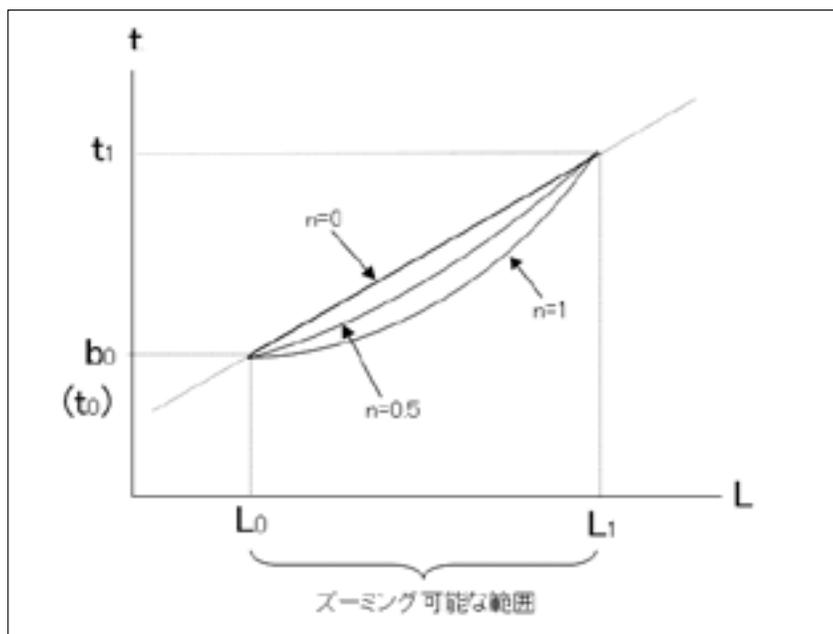


図3 実用的な融像式のグラフ表現(概念図)

7.2 融像式の働き

融像式の動作を、2重像防止定数 b が零の場合について調べてみる。すなわち、ズレの式 d

$$d = (a / (\lambda_1 - \lambda_0)^n) \times (\lambda - \lambda_0)^{n+1} \quad (\text{式9の下段式の再掲})$$

の動作を説明することになる。

(1) 値 $d > 0$ の場合

定数 a が正値の場合である。左右画像によって画面上に投影された丸点が生成する立体像は、図2(1)のようにディスプレイ画面の後方に融像する。そして、マウス操作によって、視点を画像に近づける(λ を $\lambda_1 \rightarrow \lambda_0$)と、ズレの式の値 d は正値から0に近づき、それに応じて平板に貼られた左右画像から投影された両丸点の間隔 D は式10によって正値から0に近づく。その場合、立体像の位置を定める式13の u は減少するので、立体像は閲覧者には近づくように感じられる。逆に視点を画像から遠ざける(λ を $\lambda_0 \rightarrow \lambda_1$)と、ズレの式の値 d は大きくなり、平板に貼られた左右画像から投影された両丸点の間隔 D は大きくなる。その場合、立体像の位置を定める式13の u は増加

するので、立体像は閲覧者から後退するように感じられる。

(2) 値 $d < 0$ の場合

融像式において $d < 0$ というのは、定数 a が負値である場合である。これまで融像式の定数 a は正値を採用すると述べてきたが、定数 a が負値の場合には、次のような不都合なことが生じる。画面上に透視投影された左右画像の黒丸点と白丸点の位置が本節(1)項とは逆になり、この場合には、図2(2)のように立体像はディスプレイ画面より手前に融像し、一般の閲覧者には見難いといえる。そして、マウス操作による立体像の動きは、本節(1)項とは反対であり、視点を画像に近づける(λ を $\lambda_1 \rightarrow \lambda_0$)と、ズレの値 d は負値から0に近づき、平板に貼られた左右画像から投影された両丸点の間隔 D は式10から負値から0に近づく。その場合、立体像の位置を定める式13の u は負値から0に近づくので、立体像は閲覧者から後退するように感じられる。逆に視点を画像から遠ざける(λ を $\lambda_0 \rightarrow \lambda_1$)と、間隔 D は0から負値となっていく。その場合、立体像の位置を定める式13の u は0から負値となるので、立体像は閲覧者に向かってくるに感じられる。このような立体像は人間の感覚からすると、矛盾する動きとなるので、それ故、定数 a は正値を採用する必要がある。

(3) 値 $d = 0$ の場合

ちょうど平板に貼られた左右画像の投影する両丸点(黒丸点と白丸点)のズレがなく重なる場合であり、このとき、視点の距離 λ は λ_0 となり、ズレの式の値 d は零であり、画面上に投影された両丸点の間隔 D も0(零)ある。この場合、図2(3)のように立体像はディスプレイ画面上で融像する。

7. 3 融像式の役割と特性

融像式の役割と特性は、上記の各所で述べてきたが、以下にまとめておく。

(1) 2重像の発生を防ぐ役割は、定数 b が担う

2重像の発生を防止する役割は、融像式の定数 b が担っている。定数 b はパラメータとして、視覚実験によって、接近限界距離 λ_0 で2重像が発生しないように、その値が決定される。それゆえ、この定数を「2重像防止定数」と呼ぶ。これにより、第2章の課題1を解決することが出来る。

(2) 立体像の遠近感を調整する遠近ファクターは、定数 a が担う

遠近立体感を調整する実質的な役割は、融像式の定数 a が担っている。定数 a はパラメータとして、視覚実験によって、後退限界距離 λ_1 で立体像の遠近感(奥行き感)が最良になるように、その値が決定される。それゆえ、この定数を「遠近感制御定数」と呼ぶ。これによって第2章の課題3を解決することが出来る。

(3) 立体像の前進/後退の調節は、前進/後退ファクター f の冪乗の指数 n が担う

立体像の前進/後退を行うのは、前進/後退ファクターである。指数 n は0から2の間の値を取る定数である(n は経験的に導入されたパラメータであるため、2以上であってもかまわないが、実用的な観点から上限を2とする)。融像式を用いたこのファクターの閲覧者に対する効果は、指数 n によって変わってくる。指数 n の値が0から2に増加するに従って、前進/後退の効果は距離 λ に比例するが如く、 λ が大きくなればなるほどより強くなる。指数 n は任意性のあるパラメータ

なので、ステレオフォトビューアに基づく立体写真閲覧ソフトの開発者による事前の視覚実験によって、個人差を吸収する平均的な値を決定するのがよい。これによって第2章の課題2を解決することが出来る。

(4) 接近限界距離 λ_0 と後退限界距離 λ_1 は、ステレオフォトビューアの定数とみなし得る

接近限界距離 λ_0 と後退限界距離 λ_1 は、ステレオフォトビューアの定数とみなすこともできる。前者の距離はカメラの解像度に関係するが、この値は、撮影された立体写真の内容(例えば夜間風景、霧の風景、夏の海岸での風景、人物等)によっても変化すると思われる。それ故、個々の立体写真毎に接近限界距離 λ_0 を定めてもよい。しかし、事前の視覚実験によれば、一般的なデジタルカメラでは同じ値を使用しても差し支えない。従ってこの値は、ステレオフォトビューアに基づく立体写真閲覧ソフトの開発者によって定める値としてもよい。なお、距離を比較する場合には、基準を定めなければならないが、ステレオフォトビューアにおいては、後退限界距離 λ_1 を基準と定めることにすると、接近限界距離 λ_0 は後退限界距離 λ_1 の0.05～0.1倍程度である。

(5) 閲覧者が視認する立体像の動きは、間隔 D に基づく

左右画像間のズレに関する最良の再配置は、融像式 t が担っているが、閲覧者が視認する立体像の動きは、融像式 t と連動する間隔 D が担っている。従って、間隔 D を用いると、閲覧者に視認される立体像の融像位置や立体像の倍率をシミュレートすることが可能となる。このシミュレーションは、参考文献1で示したので、本論文では省略する。

8. 視覚実験による定数の決定とステレオペアのズレ補正

8.1 定数 a 、 b の決定

融像式の定数 a 、 b は、パラメータとみなし、視覚実験によってその値を決定する。最初に、画像のズーム対象領域に対して視点から画像までの距離 λ を接近限界距離 λ_0 でズームインする。そして、左右の画像(黒丸点と白丸点)がズレなく重ね合わさるように、定数 b の値を決定する。その後、距離 λ を後退限界距離 λ_1 でズームアウトする。そして、立体像の奥行きが最良になるように、定数 a の値を決定する。この一連の実験は、立体写真毎に行うものとする。この視覚実験の説明は、参考文献1で詳しく示したので、本論文では省略する。

8.2 ステレオペアのズレ補正

2台の普通のカメラ(ただし同じメーカー製で同じ型)を用いて立体写真を撮影した場合、注意深く作業を行っても2つの左右画像の間で上下のズレや傾きによるズレが生じて、そのままではステレオペアとして適当でないことが起こる。(立体写真専用カメラを用いても、ズームを行うと、その解像度の限界点付近では、同じようなズレがしばしば生じる)。2台のカメラを雲台に取り付ける際に発生する極わずかな設置ミスがそのズレの原因となるのである。このズレは、写真全体をズームアウトして眺めるような目的でその立体写真を使用する場合は、それ程問題にならないが、ズームインを行い狭い領域を精査して眺めるような目的の場合は、左右画像間のズレが顕著になるので、その立体写真はステレオペアとして使用出来ないことになる。

そこでステレオフォトビューアには、ステレオペア(左目用画像と右目用画像)の形成を妨げる左

右画像間のズレを補正するには、以下の3つを考える必要がある。

- ①左目用画像と右目用画像の間における上下のズレの補正(補正値は p)
- ②左目用画像と右目用画像の間における左右のズレの補正(補正値は t)
- ③左目用画像と右目用画像の間における傾きのズレの補正(補正値は θ)

このズレを補正するための方法(機構)については、すでに詳しく紹介したので、本論文では省略する。参考文献1を参照されたい。

9. 画像の再配置の方法

9. 1 アフィン変換

左右画像の間で生じるズレを補正する実用上の手段は、ステレオフォトビューアが用いる3DCGの座標変換機能であり、それを使用することである。ズレの補正とは、右画像用平板を仮想空間内で再配置することであるが、それは簡単な座標変換によって行うことが出来る。その座標変換とは拡大縮小、回転、平行移動のアフィン変換である。一般の3DCGはその内部に 4×4 のアフィン変換行列を持っており、変換は行列演算によってリアルタイムに行われる。

アフィン変換行列を用いて右画像用平板の位置を、基準となる左画像の位置に対して適切に座標変換(再配置)する方法を示そう(文献2)。中心が座標系の原点に置かれた右画像用平板の4次元行ベクトルを $[0 \ 0 \ 0 \ 1]$ 、座標変換(再配置)を行ったときのその平板のその行ベクトルを $[x \ y \ z \ 1]$ とする。なお、そのベクトルの4番目の要素の1は同次座標表現における1である。8. 2節に示した3つのズレ補正の手続き①、②、③を手順1、手順2、手順3と呼ぶことにするが、これらの手続きにおける補正の変換行列の要素は表1に示されている。(この表に記載された変換行列は概念的なものであり、実際には原点の定め方によってはさらなる変換が必要となる場合がある)。この表には、後述するアスペクト比の補正の手順を手順4と呼び、その変換行列も記載されているが、この補正については9. 3節で述べる。また、それら間の変換順序は9. 4節で述べるが、手順1から手順4までの順で変換(再配置)を行う必要がある。すると、そのときの右画像用平板の4次元行ベクトル $[x \ y \ z \ 1]$ は、以下の式によって座標変換(再配置)が行われるのである。

$$[x \ y \ z \ 1] = [0 \ 0 \ 0 \ 1] M_{t1} M_{t2} M_r M_s \quad (\text{式 14})$$

表1 4×4 アフィン変換行列

	手順1 (上下のズレ補正)	手順2 (左右のズレ補正)	手順3 (傾きのズレ補正)	手順4 (アスペクト比の補正)
変換行列	M_{t1}	M_{t2}	M_r	M_s
変換行列の要素	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & p & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

なお、基準となる左画像用平板に対する座標変換に対しては、それは固定されているので、変換行列 M_{11} (要素 p は零)、 M_{12} (要素 t は零)、 M_r (要素の三角関数の θ は零) は単位行列となり、結果として手順4のアスペクト比の補正変換のみが行われる。

9.2 融像式を用いた画像の再配置の説明

手順2の左右画像のズレ補正について、6.5節で述べた融像式を以下に再掲し、それを用いて、「3DCGの持つ仮想空間の機能は、左右画像における左右のズレの補正を如何にして行うのか」を説明する。

$$t = (a / (\lambda_1 - \lambda_0)^n) \times (\lambda - \lambda_0)^{n+1} + b \quad (\text{式12の下段の再掲})$$

$$(a > 0, \lambda_0 \leq \lambda \leq \lambda_1)$$

右画像用平板の x 軸に対する最適な再配置の値 t は、視点から平板までの距離 λ を用いて融像式によって求められる。 λ_0 は接近限界距離、 λ_1 は後退限界距離である。(定数 a と b の設定については、8.1節で述べた)。ところで値 t は、表1に示すようにアフィン変換行列 M_{12} の要素となっている。閲覧者が仮想空間においてズームリングを開始すると、その間、3DCGの持つ仮想空間の機能は距離 λ を常に計測し、この値 t は、距離 λ によって融像式から算出される。そして、この値 t に基づいてアフィン変換行列 M_{12} が演算され、右画像用平板の4次元行ベクトル $[x \ y \ z \ 1]$ が算出される。そして、この行ベクトルを用いて、3DCGの持つ仮想空間の機能は、右画像用平板を最適な位置へと再配置するのである。

9.3 アスペクト比の変換

手順1から手順3までの補正に加え、画像に関する手順4のアスペクト比(横/縦の値)の補正を行う必要がある。通常の写真はアスペクト比が1.333であり、XGA(4対3型)解像度対応のパソコンを用いて同型画面の立体ディスプレイに出力する場合には、アスペクト比の問題は生じない。しかし、それを横長の最近のワイドに表示すると横に像が広がるために、横幅を調整する座標変換を必要がある。例えば、XGA(4対3型)解像度対応のパソコンを用いて5対3型画面のワイド型立体ディスプレイに出力する場合、そのままでは立体像の幅は太くなるので、本来の立体像の幅に戻すため、パソコン側の画像のアスペクト比を0.8倍にする必要が生じる。このときの修正されたアスペクト比は 1.333×0.8 となるが、写真のアスペクト比(1.333)を基準とした比率を換算アスペクト比と定めると、そのときの画像の換算アスペクト比は0.8となる。

また、立体フォーマットがサイドバイサイド・フォーマットである立体ディスプレイを表示装置とする場合、ステレオフォトビューアの立体フォーマットも同様のサイドバイサイド・フォーマットにする必要がある。そのため、左右画像それぞれの換算アスペクト比をさらに $1/2$ 倍しなければならない。ちなみに、XGA型解像度対応のパソコンから5対3型画面のワイド型立体ディスプレイにサイドバイサイド・フォーマットで出力する場合には、左右画像のそれぞれの換算アスペクト比は、 0.5×0.8 、すなわち0.4ということになる。

換算アスペクト比を q で表わすと、画像用の左右平板の拡大縮小についての座標変換は、実際にVRMLを用いた場合は、 y 軸の値を q の逆数($1/q$)倍にする必要がある。 q の逆数がアフィン変換行列 M_s の要素になっている。

9.4 座標変換の順序

行列演算においては、その演算順序が重要である。順番を間違えると正しい結果を得ることは出来ない。補正のための行列演算の順序は、手順1→手順2→手順3→手順4の順序で行うものとする。ただし、手順1と手順2は、行列の演算順序には依存しないので、数学的にはどちらが先でもかまわないが、実際には、最初は未調整の立体写真で目視確認実験を行うので、手順1の上下の補正(この補正を最初に行わないと立体像が融像しない場合もある)を先に行う必要がある。

9.5 座標変換の事例

上記に示したアフィン変換が実際の3DCG言語においては、どのように記述されるのかを、VRML言語を用いて(文献³)説明しよう。そのプログラムの主要部分を図4に示す。他の3DCG言語でも記述の流れは、ここに示す例とほぼ同じと考えてよい。

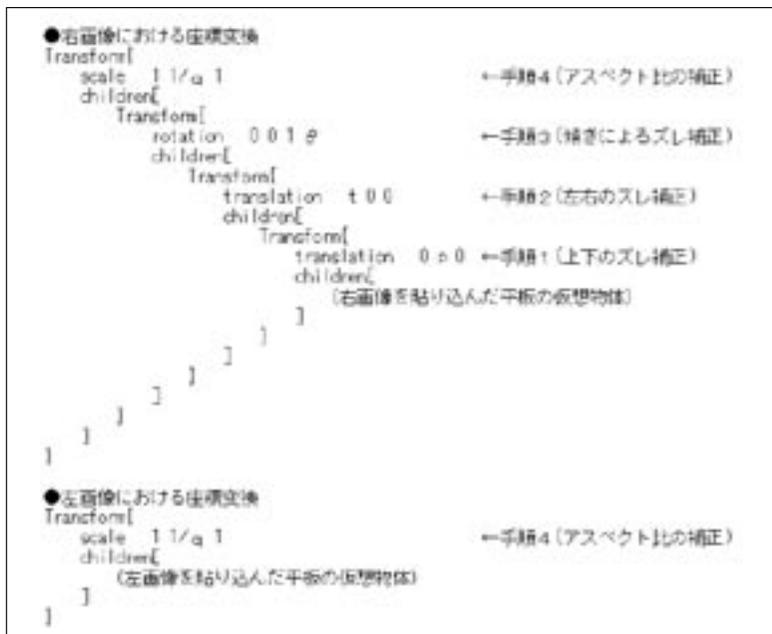


図4 座標変換の事例(VRML言語を用いた例)

図4では、左画像用平板を基準として、右画像用平板に対してx軸とy軸に沿った平行移動(手順1と手順2)とz軸を中心とした回転(手順3)を行うことによって、右画像用平板を左画像平板に対して再配置している。その結果、左右の画像間のズレを調整することが出来るのである。このように3DCG言語(図4ではVRML言語)の座標変換機能を使用すると、仮想空間における右画像用平板の再配置という主要課題を簡単に記述すること出来る。なお、図4には、手順1(上下方向のズレ補正の座標変換)、手順2(左右方向のズレ補正の座標変換)、手順3(傾きによるズレ補正の座標変換)、手順4(アスペクト比の補正の座標変換)のコメントが記載されているのに注目されたい。

10. ステレオフォトビューア方式に基づく SPV-F の開発

10.1 SPV-F の実現方法とそのプログラム連携構造

筆者はステレオフォトビューアを設計する当たり、3DGCの仮想空間を利用することを着想し、その一般的な概念を第9章までに示した。本章以降はその設計概念に基づいて筆者によって開発されたSPV-Fの説明である。

SPV-Fでは3DCGとして、VRML言語を採用した。VRML言語はそれ自身がプログラム言語(ただタグ言語)である。VRML言語で記述されたソースコードは、WebブラウザのプラグインであるVRMLビューアによって解釈されレンダリングされる。WebブラウザとVRMLビューアについてはInternet Explorer(IEと略記)とCortona3D Viewer(Cortonaと略記)を用い、VRMLを外部からコントロールするプログラム言語としてJavaScript言語を用いた。また左右画像を生成するためにHTML言語を用いた。従ってSPV-Fの立体フォーマットは、サイドバイサイド・フォーマットを採用している。

上記の言語を組み合わせ、立体像を再生する方法は、VRMLコンテンツを立体視するテーマのもとすでに論文(文献5)を発表している、それを参照して頂きたい。本論文では、その方法を立体写真に応用したものである。図5にIE、Cortonaおよび言語間の連携関係を示し、以下には言語などが司る機能を箇条書きにしておく。なお、SPV-Fは、OSがWindows XP、WebブラウザはIE Ver.6.0、VRMLビューアはCortona Ver.5.1の組み合わせのもとで開発を行った。

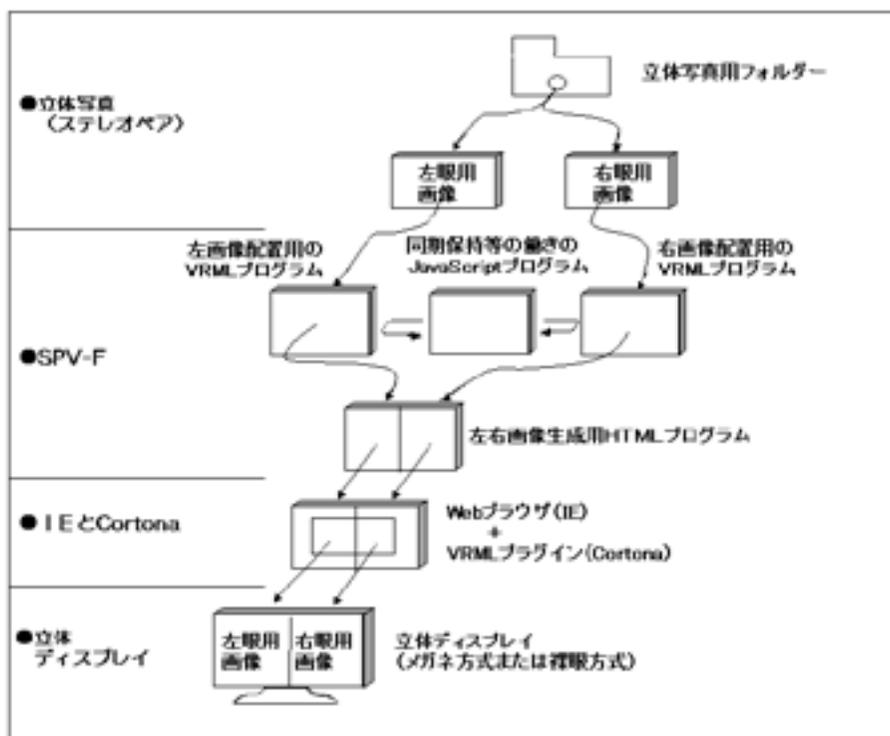


図5 SPV-Fにおけるプログラムの連携構造

● VRML 言語

左右画像を適切に再配置し遠近的立体感を設定するプログラムとして働く。

- ・ 仮想空間の設定
- ・ 変換行列の設定
- ・ ウォークスルー機能(ズーム機能)の活用
- ・ テクスチャ機能の活用
- ・ 視点の設定
- ・ アニメーション機能(自動ズーム機能)の活用

● JavaScript 言語

VRML のレンダリングを司る Cortona に指令を出す。

- ・ ズームにおける左右画像の同期保持
- ・ ユーザからの指令を解析
- ・ 立体写真の情報保存

● HTML 言語

左右 2 画面を生成するためのプログラムとして働く。

- ・ 左右の 2 画面の設定
- ・ ユーザインターフェースの設定

● Cortona3D Viewer (Cortona)

VRML ソースコードを解析して、実際の画像をレンダリングする。

- ・ 3DCG の持つ基本機能(行列演算など)の実現
- ・ VRML Automation 機能による左右画像と視点の同期等の実現(文献 6)

● Internet Explorer (IE)

HTML ソースコードを解析して、実際の 2 画面を生成する。

JavaScript ソースコードを解析して、JavaScript に 2 画面等の同期を行わせる。

- ・ Cortona の実装

10. 2 SPV-F の実際

SPV-F には、各種の操作ボタンが用意されている。そのボタンを以下に示す。

●ズレ補正機能ボタン

ズレを補正して立体写真としての完全なステレオペアを作るボタンである。

- ・ X 補正・・・左右のズレを補正する(融像式の定数 b を決定する)。
- ・ Y 補正・・・上下のズレを補正する。
- ・ R 補正・・・傾きによるズレを補正する。

●深度決定機能ボタンとアスペクト比決定機能ボタン

好みの遠近的立体感を定め、アスペクト比を設定するボタンである。

- ・ 深度決定・・・遠近的立体感を定める(融像式の定数 a を決定する)。
- ・ A 比決定・・・立体像をウィンドウ表示画面のアスペクト比に適合させる。

●探索機能ボタン

立体写真の置かれた仮想空間をマウス操作によって探索するボタンである。

- ・ ズーム・・・立体写真の任意箇所をズームする。

- ・シフト・・・立体写真の平面上の任意箇所を閲覧する。
- 閲覧機能ボタン
- 立体写真を閲覧する際に使用するボタンである。
- ・「進む」・・・次の立体写真を見る。
 - ・「戻る」・・・前に表示した立体写真を見る。
 - ・自動・・・フォトフレームのように自動表示を行い、繰り返し立体写真を見る。
(自動ズームも行う。)

図6に実際に開発したSPV-Fの画面を掲げ、図7に画面の下部に設定されている操作ボタンの拡大図を示す。なお、図7には示されていないが、実際には操作ボタンの下部には、ズレの補正情報が表示されるステータス領域がある。



図6 SPV-Fの画面



図7 SPV-Fの操作ボタン

ズレ補正ボタンについての使用法について、多少述べておく。これはカメラ撮影する際に発生するステレオペアのズレを補正するためのボタン群であり、閲覧前の事前準備として用いる。閲覧者によって補正ボタンがクリックされる度に、SPV-F 内部で微小な値が作られる。この値に基づいてクリック毎に座標変換が瞬時に行われ、仮想空間内の右画像用平板が左画像用平板を基準として再配置されていく。閲覧者はその度に、左右画像から生成される立体像を自身の目で確認する。例えば Y 補正(+と-の2種類)を行うプラスボタンをクリックすると、右画像用平板が上方に微小移動する。最適な立体像が生成するまで、Y 補正のプラスボタンあるいはマイナスボタンをクリックして立体像の状態を微調整するのである。

11. おわりに

本論文では、3次元仮想空間を用いて立体写真のズーミングを制御する融像式を提案し、それを SPV-F と名付けた立体写真閲覧ソフトを開発した。この融像式を用いることにより、3次元仮想空間内で解像度限界まで任意領域をズーミングしても、微細な部分に関する立体像を、2重像の発生を伴わない自然の動きのもとで再生することが可能となったと言える。

ちなみに SPV-F の立体写真提示能力は以下の通りである。富士フィルム社製のデジタルカメラ A220 を 2 台用いて、1200 万画素数のステレオペアを作成し、SPV-F を用いて実験したところ、10～15 倍程度までのズーミングが可能であった。そして、2重像の発生しない、また自然な動きの立体像を再生することが出来た。使用した立体ディスプレイ装置は、Dimen 社製 G170P の偏光メガネ式ディスプレイ装置で、解像度は XGA(1024 × 768)であった。

参考文献

- (1) 広内哲夫：「融像式を用いたズーミング可能なステレオフォトビューア」、情報研究（文教大学情報学部紀要）、pp.17～pp.56、第46号(2012)
- (2) 広内哲夫：『Java3D グラフィックス—基礎から立体アニメーションまで—』、セレンディップ（小学館発売）(2004)
- (3) 広内哲夫：『Web3D グラフィックス—VRML で創る仮想世界—』、ピアソン・エデュケーション（2001）
- (4) 河合隆史、田中見和：『次世代メディアクリエイター入門 I —立体映像表現—』、カットシステム（2003）
- (5) 広内哲夫：「ウェブサイトの VRML コンテンツを立体視化するステレオビューアの開発」、情報研究（文教大学情報学部紀要）、pp.19～pp.40、第43号(2010)
- (6) 荒屋真二：「荒屋研究室のホームページ」（VRML Automation の項）
<http://www.fit.ac.jp/~araya/jis/>（2008年に閲覧）