

# 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

広内 哲夫

## A stereo-photographic viewer with which zooming can be made by using a fusion equation

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 a fusion equation and 2DCG. Furthermore an SPV realized based on the said method is introduced.

Just exclusive use of 2DCG 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 stereoscopic space. Additionally for the parameters required for the fusion equation, a method to determine them was adopted based on a judgment experiment of the stereoscopic image visually observed by users.

The SPV 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. On the other hand, the SPV, into which a mechanism allowing the slippage of the left and right images produced to be adjusted while stereoscopic photography is being taken is taken, can make out perfect stereo-pairs.

### 1. はじめに

#### (1) 立体写真をズーミングする機能

最近の立体ブームで、パソコン、カメラ、テレビ、ゲーム機、携帯電話、映画などの情報関連技術に立体技術が本格的に適用され始めた。その中で古くからある両眼視差を利用した立体写真は、現在でも多くの愛好者がいて、同好の士の集いも盛んである。しかしながら、世の中で使われている立体写真閲覧装置(パソコン立体表示ソフトも含む)は、単に写真を立体的に眺めるものだけで、それ以上の機能を持ったものは少ないと言える。このような状況に鑑み、筆者はパソコンと立体ディスプレイ装置を用いる立体写真閲覧ソフトの新しい仕組みを考案した。これは滑らかにズーミングしながら立体写真の閲覧を可能ならしめるものであり、一般の立体写真閲覧ソフトを開発する際に

応用可能な汎用性の高い方式といえよう。筆者はこの方式を「ステレオフォトビューア」と呼んでいる。

通常、本論文で示すような特別の工夫を施さない、単に立体写真をズームするソフトウェアを用いた場合、立体ディスプレイ上の微細な領域を拡大(ズームイン)する過程で、立体像が手前に迫って来ないで後退してしまうという、人間の感覚に合わないような動き方をする現象が起こったり、あるいはそれまで融像していた立体像が2つの画像へと分裂していき、閲覧者が立体像を融像出来ないという状態が生じたりする。前者は立体写真のズーミングにおける立体視の本質に関わる困った現象である。また後者は閲覧者の立体視能力がその限界を超え、立体像の融像がもはや出来ないという状態である。

特に後者の立体像の融像不能を本論文では、「融像崩壊による2重像」(以降、単に「2重像」と呼ぶ)と呼ぶことにする。このときの2重像は、単に平坦な2つの画像がズレて重なっただけのものである。(なお、立体ディスプレイ装置の品質や性能から生じる「クロストークによるゴースト」と呼ばれる2重像と似た現象もあるが、これは本論文の主題ではない)。

そこで、筆者はステレオフォトビューアにおいて、2次元コンピュータグラフィックス(2DCG)の画像処理技術を利用して、立体像の動きや2重像が発生する原因を検討した。そして立体像融像の仕組みを分析し、ステレオフォトビューアにおいてズーミングを司る「融像式」を定義した。融像式とは、2重像の発生を抑え、立体像を人間の感覚に合うように動かす数学的な式である。

筆者はステレオフォトビューアの方式に基づいて、「SPV」と名付けた立体写真閲覧ソフトを開発した。SPVは2DCGの画像処理技術を用いたズーミングの機構と2重像の発生を抑える融像式を備えている。これによりSPVを用いた閲覧者は、立体写真を滑らかにズーミングしながら希望する領域を立体的にかつ鮮明に眺めることができるのである。

## (2) 立体写真(ステレオペア)を編集する機能

立体写真は、2台のカメラを雲台と呼ばれる支え台に取り付けて、2個のレンズの間隔を経験的に定めて左右画像(一般にはステレオペアと呼ばれる)を撮影し、ステレオペアを適当な立体写真閲覧装置(19世紀中頃からレンズ、鏡、プリズムを用いて作られている)を用いて眺めるというやり方で、昔から愛好者に楽しまれている。この方式は、2個のレンズの間隔を雲台上で自由に変わることが出来るために、近景から遠景までの立体写真を撮影することが可能である。最近日本の大手カメラ会社が立体写真専用カメラ(左右の画像を撮る2台のカメラが1セットになっている)を販売し始めたが、そのカメラはレンズの間隔が固定されているので、人物撮影などの近景限定向きとなっている。そのため、雲台を用いてカメラ2台を使用するという従来からの方式は、現在においても、遠景撮影をする愛好者にとっては捨てがたい魅力がある。

ところで立体ディスプレイ装置を用いる場合、立体写真は撮影したままでは、効果的なステレオペアとはならない。2台のカメラの雲台への気付かない取り付けミスや撮影時のカメラの傾きにより、若干ではあるが左右画像の重ね合わせにズレが生ずることが普通に起こるのである。そこで、左右画像を上下・左右でトリミングしたり、一方の画像を回転して傾きを直したりして、適切なステレオペアになるように編集する必要がある。これは結構厄介な作業であるので、最近ではアドビ社のプレミアと呼ばれる画像・映像編集ソフトが用いられることが多い。

そこで筆者は、左右画像を編集して完全なステレオペアを生成する機構もステレオフォトビューアに備える必要があると考えた。この機構には2DCGの持つアフィン変換(座標変換)を応用して

いる。SPVのステレオペア編集機能を用いると、初心者でも、立体写真専用カメラではない、普通の2台のカメラで撮影した立体写真から奥行き感のある立体像を再生することが出来るのである。

☆

☆

☆

ステレオフォトビューアをソフトウェアとして具現化したSPVは、立体ディスプレイ装置(メガネ式、裸眼式ともに可)、立体テレビあるいは立体プロジェクターのスクリーン上に、立体像をそれぞれ再生することが出来る。

第2章で閲覧者の立体感覚を分析し、第3章から第10章でステレオフォトビューアの考え方とその方式を示し、第11章では実現されたSPVを具体的に紹介する。

## 2. 立体視感覚の分析

### 2.1 立体視感覚の模式的な表現

筆者はステレオフォトビューアを設計し、試作用プロトタイプのビューアを開発している最中に、立体ディスプレイ画面上に再生された立体像から得る立体視感覚は、閲覧者の置かれた立体写真閲覧環境によってかなり変化することに気が付いた。閲覧環境とは、立体ディスプレイ装置とパソコン立体表示ソフトの特性が作るものである。(立体像の眺め方によっても、閲覧者による立体感は多少変化するが、これは本論文の主題から外れるので、本論文では触れないことにする)。

2台のカメラで撮影された立体写真の立体感は、立体写真の撮り方やそのときの撮影風景等によって変わるので、個々の立体写真はそれぞれ“固有の”立体感を持っている。そこで、その立体感を立体写真の「固有立体感」と呼ぶことにする。この固有立体感による個々の立体写真の奥行き感は、周知のようにその撮影の時点の状況で決定されるので、原理的に変更することが出来ない。これについては、2.2節で再考する。

しかしながら、閲覧者の置かれた立体写真閲覧環境(据え置き型の立体ディスプレイ装置とパソコン立体表示ソフトの特性)による立体感は、固有立体感とは多少異なる立体感である。この立体感とは、同じ立体写真であっても立体写真閲覧環境が異なれば、立体像が立体ディスプレイ画面の前方に存在し、後方に存在したり、あるいは画面付近に存在したりするような立体感である。このように閲覧者が感じる立体感は、立体写真の固有立体感に何らかの作用が加わったような立体感である。

そこで、閲覧者の感じる立体感を「遠近的立体感」、何らかの作用を「遠近ファクター」と呼ぶことにすると、筆者は、遠近的立体感を以下のような模式で表現できることに気が付いた。(ただしこの模式は数式ではなく、概念的なイメージを表わしていることに注意されたい)。

$$\text{「遠近的立体感」} = \text{「固有立体感」} \times \text{「遠近ファクター」}$$

この模式は、「閲覧者が感じる立体写真の立体感は、立体写真固有の立体感に何らかの遠近感を発現させる因子が作用した結果の立体感である」ということを表わしている。そこで閲覧者の感じる遠近的立体感を、次節以降で説明するように、立体写真の固有立体感と遠近ファクターに分けて分析することにする。

このように「遠近的立体感」、「固有立体感」、「遠近ファクター」の3つに分けて立体写真の立体感を考察することは、ステレオフォトビューアを設計する際の見通しを良くする。というのは、例

えば立体写真をズームインしているとき、閲覧者は左右画像(ステレオペア)による立体像をある時点から2重像として見てしまい、立体像を融像できないという状態が生じるが、この2重像の発生を抑える工夫を、固有立体感と遠近ファクターの分析から具体化することが出来るのである。

## 2.2 固有立体感

立体写真に3台の車が写っていて、それらが車A、車B、車Cの順に手前から奥に並んでいるとする。(立体写真は、そのステレオペアの間で上下あるいは傾きにズレがあれば、それが除去されているものとする。これはステレオペアが立体像を生じる重要な要件である。画像の重なるのズレを調整する方法については、第9章で述べる)。レンズや鏡などを用いた伝統的な覗き見方式の立体閲覧装置を通してそれを眺めた閲覧者が、3台の車の配置関係に注目した場合に感じる立体像の奥行き感を、すでに述べたように本論文では「固有立体感」と呼ぶ。立体写真の固有立体感は、写真撮影の際にすでに決定され、原理的には変更することができない。

立体写真において固有立体感を生み出す要因は、その風景を撮影した2台のカメラの2個のレンズ間隔である。立体写真のステレオペア同士は、左右カメラのレンズ間隔に起因する水平方向のズレを作り出している。閲覧者がステレオペアを適切に眺めた場合、閲覧者の脳がそのズレの間隔(すなわち両眼視差)に基づいて立体像を融像するのである。その間隔が大きくなるに従って、立体写真による立体像の奥行き感はより増すように感じられる。ステレオペア同士のズレの間隔がある値よりも大きくなると、閲覧者にとって立体像の融像が難しくなり、その限界を過ぎると立体像は2重像へと崩壊する。またそのズレの間隔が小さいと奥行き感は減り、間隔が零の場合には、結果的に2枚の同一画像を見ているのと同じで、立体感は感じられない。

立体写真におけるこのような撮影の条件は昔から知られている。従って一般的には、後に立体ディスプレイ装置とパソコン立体表示ソフトを用いて閲覧しようとする立体写真(ステレオペア)は、事前に2台のカメラが適切なレンズ間隔になる状態で撮影しておく必要がある。

なお、立体写真を撮る2台のカメラは同一型で同性能のカメラである必要があり、その2つのレンズの向きが同一平面で平行になるように雲台に設置され、そしてその雲台は、水平に保たれるようにするものとする。(立体写真で固有立体感を生じさせる方法としてレンズ間の間隔以外に、輻輳と呼ばれる方法がある。この方法は2つのレンズの向きを互いに交差するように設置する方法であるが、本論文における立体写真撮影においては、上記の理由から輻輳は採用しないものとする)。

## 2.3 遠近的立体感

ステレオフォトビューアは、パソコン上で実行され、その出力である立体像を据え置き型の立体ディスプレイ装置上に再生する。閲覧者がそのビューアを通して立体写真を眺めた場合の立体感、すなわち遠近的立体感について、2.2節で述べた3台の自動車立体写真の例を用いて説明しよう。立体ディスプレイ装置には当然のことながら画面があり、その立体写真による立体像の中で3台の自動車を、閲覧者は“その画面を通して眺めている”ことになる。このとき、閲覧者は、写真が撮られたときに決まる固有立体感のもとで3台の車配列の奥行き感を感じている。3台同士のそれぞれの相対的な奥行き感だけに注目すると、そのときの立体感は、固有立体感の場合に感じられる際の立体感とほとんど同じである。しかし、筆者はそれだけではなく、閲覧者は別の立体感も感じていることに気が付いたのである。その立体感とは、閲覧環境によって、①「場面Ⅰ：車3台ともが画面よりも近くに飛び出して見える」、あるいは②「場面Ⅱ：車3台ともが画面より遠くに引っ込

んで見える」、あるいは③「場面Ⅲ：車3台のうちの1台ないし2台の車が画面付近にあるように見える」というような、3つの場合のいずれかになるのである。このように、立体ディスプレイ装置とパソコン用立体表示ソフトを用いる場合には、閲覧者にとっては、固有立体感に加え更なる遠近的な立体感が加わることになる。この立体感を「遠近的立体感」と呼ぶことは、2.1節で述べたことである。

試行用プロトタイプ・ビューアを開発し、それを用いた事前の実験で分かったことであるが、閲覧者は、立体写真から視認される立体空間全体を一塊に感知し、そしてその立体空間の位置を、画面を基準とする遠近として感じるのである。閲覧者の遠近的立体感は、立体写真の固有遠近感に遠近ファクターが作用する訳であるが、その遠近ファクターに当たるものが、ステレオフォトビューアの機能ということになる。このようなイメージを表現する模式を2.1節で示したが、以下に再掲しておく。

$$\text{「遠近的立体感」} = \text{「固有立体感」} \times \text{「遠近ファクター」}$$

上記した車の立体写真閲覧において、閲覧者が最初に場面Ⅰの立体感を感じていても、ステレオフォトビューアに遠近ファクターの機能が備わっていれば、閲覧者がそれを操作して、場面Ⅱの立体感、あるいは場面Ⅲの立体感を感じることが出来るのである。その具体的なイメージ図を図1に示す。描かれた3つの図の中で、黒枠は画面を、また点線で示された立方体は立体空間を意味する。図の左側は立体空間が画面より手前に、図の真ん中はそれが画面よりも後方に、図の右側はそれが画面付近に、それぞれ配置されており、閲覧者にとってはそのように感じられるのである。

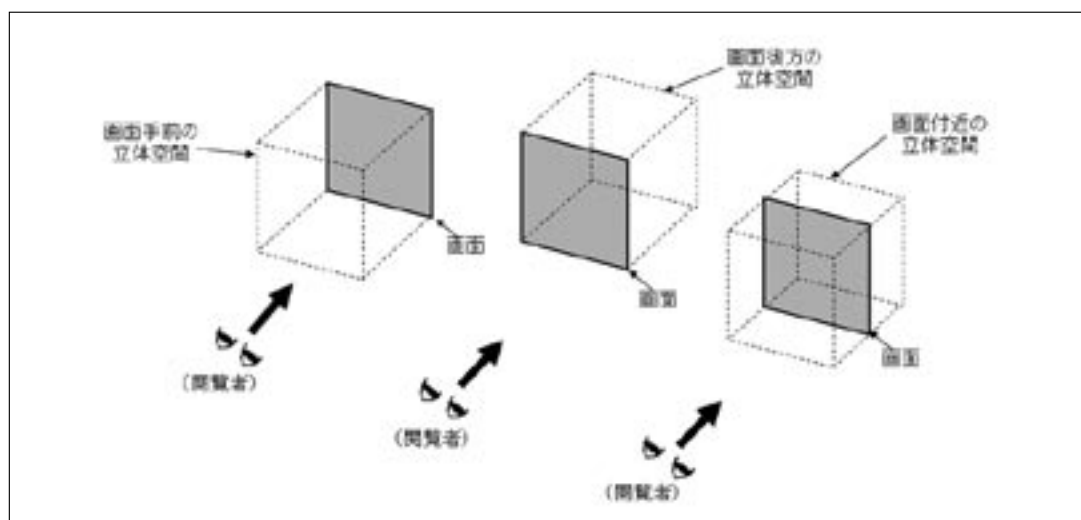


図1 立体空間のイメージ図

このような立体感覚は状況によっても変わるが、据え置き型の立体ディスプレイ装置を使用する場合に明確に生じる感覚である。すなわち、立体写真の再生する立体空間の全体が、立体ディスプレイ画面位置に対して、どのような“遠近的な”位置関係で配置されているかである。立体空間の

配置のされ方によって、上記の車配置において、場面Ⅰ、場面Ⅱ、場面Ⅲの立体感が出現する。(これは閲覧者の遠近的立体感からの立場からすると、2重像に関係した重要な問題となる。これが本論文のテーマとなっているので、それについては第5章から第8章で述べる)。

立体ディスプレイ装置と試行用プロトタイプ・ビューアを用いて立体写真を閲覧するとき、この遠近的立体感を筆者は強く感じたのである。遠近的立体感は、立体写真から立体像を立体ディスプレイ上に再生するステレオフォトビューアの特性に依存する。立体像の見やすさは主観的なものであるが、立体像は、画面の前に飛び出している(画面より近くに視認される)よりも、多少画面の後ろに引っ込んでいる(画面より多少遠くに視認される)方が閲覧者にとっては見やすいと、一般には言われている。

なお、レンズや鏡などを用いた古典的な覗き見方式の立体写真閲覧装置あるいは立体HMD(ヘッド装着型の超小型立体ディスプレイ)を用いて立体写真を眺めた場合は、個人差もあるが遠近的立体感は弱く、一見しただけでは立体写真の固有立体感のみが強く感じられる。遠近的立体感を感覚するには、視野角(ディスプレイ分野ではなく視覚分野で用いる用語である)の大きさが関係しているようである。

## 2.4 遠近ファクターへの注目

立体ディスプレイ上に立体写真を立体像として再生した場合には、その立体像を視認する閲覧者は、2.1節で示した固有立体感に遠近ファクターが作用して生じた遠近的立体感を感じている。しかし、これまで立体ディスプレイ上に立体像を再生するに当たっては、この遠近ファクターを十分に把握しないで立体感を論じるのが普通であった。ステレオフォトビューアを設計するような工学的な立場からは、遠近的立体感のもとで遠近ファクターを考察する方が、見通しよく議論を進めることが出来るのである。そこで次章以降では、遠近ファクターに注目し、これを適切に取り扱うことにより、立体写真をズーミングするステレオフォトビューアの設計が可能であることを示す。

## 3. 一般の2DCG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、2DCGの画像処理技術を使用するので、一般の2DCGが持っている機能を説明しておく。2DCG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 OpenCV-API と C / C++言語の組み合わせ
- ② Java2D-API と Java 言語の組み合わせ
- ③ HTML5 言語と JavaScript 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HTML5言語の画像処理機能とJavaScript言語を用いるが、強いて言えば、後者が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は、Java2D-APIとJava言語の組み合わせで開発している。それについては第11章で述べる。

これらのOpenCV-API、Java2D-APIは、2DCG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。SPV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。

- ① 画像(テクスチャーと呼ばれる)を表示する機能
- ② 画像を回転、平行移動、拡大縮小する座標変換機能(アフィン変換機能)

ソフトウェア開発の立場からは、SPV におけるユーザインターフェースやズーム機能、フォトフレームのような自動閲覧などについては、親言語の機能を用いて作成することになる。

#### 4. ステレオフォトビューアにおける立体視化

#### 4.1 画像と画面の関係

最初に 2DCG において、ステレオペアと呼ばれる 2 枚一組の立体写真(左眼用画像と右眼用画像)が立体ディスプレイ画面に描画される概念を簡単に説明述べよう。図 2 は、ステレオフォトビューアにおいて取り扱われる左右の画像の概念的イメージを斜め上から描いたものである。そこには 2 枚の薄い平板が重なるように置かれており、2 枚の平板には 2 台のカメラで撮影された 2 枚の左右の画像(ステレオペア)が 2DCG のテクスチャー機能によってそれぞれ貼られているものとする。

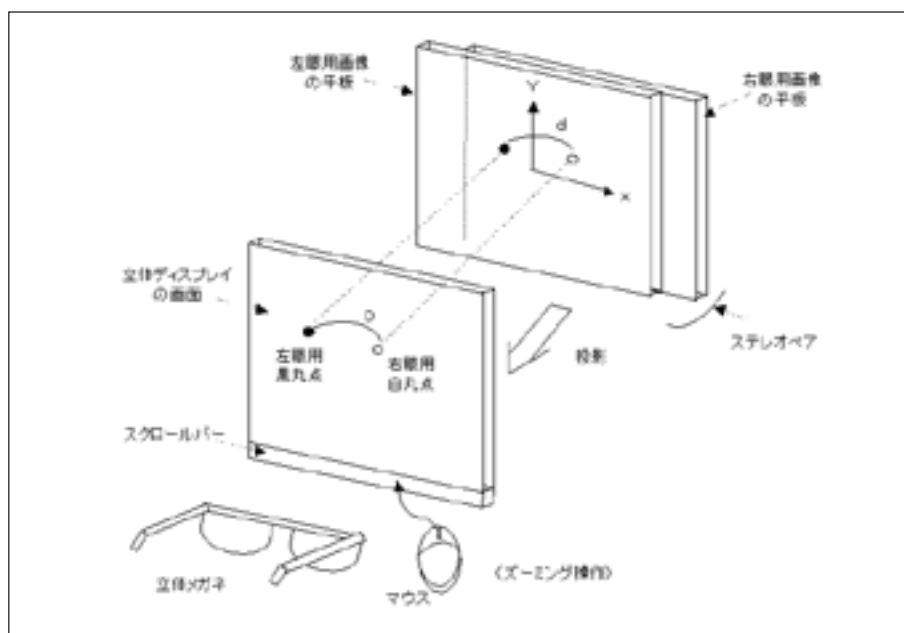


図2 ステレオフォトビューアにおける画像配置の概念的イメージ

図2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、概念的には2枚の平板はともに同じ厚さであり、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は左眼用画像、右カメラで撮影された画像は右眼用画像として使用される。そして左眼用画像の平板の中心は、ステレオフォトビューアに固定したxy平面座標系の原点に、固定されて配置されている。しかし、右眼用画像の平板は、上下、左右の方向に可動し、その座標系の原点を中心にプラス回りまたはマイナス回りに回転する。平面座標系は、右に向かって正方向のx軸、上に向かって正方向のy軸となっている。そして、2つの平板に貼られた左右の画像は、重ね合わされて立体ディスプレイ画面に平行投影される。

閲覧者は、その投影された画像を立体視する。立体ディスプレイ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で、投影されたディスプレイ画面上に重ね合わされて投影された左右の画像を眺めて立体像を融像する。その際、閲覧者はマウスを用いたズーミング操作により、立体写真(画像)の任意の箇所をズーミングしながら眺めるのである。

#### 4.2 画像の画面への投影

ステレオペアを構成する2つの左右画像は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2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1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)を  $d$  とする。そしてディスプレイ画面上に投影された左右画像の丸点間の間隔を  $D$  とする。この間隔  $D$  が遠近的立体感を生み出すのであるが、これについては後述する。

ところで、画面上に投影された画像の倍率を  $m$  とすると、①倍率  $m$  と画面上に投影された左右画像の丸点間の間隔  $D$  は、そのときのズレの値  $d$  が一定もとで比例関係にある。また、倍率  $m$  が一定であれば、平板に貼られた左右画像間のズレの値  $d$  は、ディスプレイ画面のサイズに比例する間隔  $D$  の大きさに投影される。すなわち、②ズレの値  $d$  と画面上に投影された左右画像の丸点間の間隔  $D$  は、そのときの倍率  $m$  が一定のもとで比例関係にある。

上記の①と②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。

$$D \propto m \times d \quad (\text{式1})$$

### 5. 融像における問題点

マウス操作を行うことによってズーミングが出来るステレオフォトビューアの基本構造は第4章の通りであるが、この構成だけでは立体写真の持つ固有立体感から閲覧者が感覚する遠近的立体感をステレオフォトビューアが再生するという課題を解決できない。というのは、ズーミングして画面の倍率を上げても2重像が発生しないように、第2章で定めた遠近ファクターの作用をステレオフォトビューアに導入して、閲覧者の感じる遠近的立体感を調整するような何らかの機能を遠近ファクターに持たせる必要がある。これは第6章の「融像式」と呼ばれる式に繋がる課題であるが、その前に問題点と解決案を、以下に指摘しておく。

#### 5.1 問題点1(ズームイン操作による2重像の発生)

ステレオフォトビューアを用いて立体写真を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(これは虫メガネで微細部分が拡大されて見えるのと同じである)。倍率が一定以上の値となると、その左右画像のズレが限界を超え、画面に2重像が現れ、閲覧者にとっては立体像が融像しない。立体感覚は主観的な感覚であるがゆえに、厳密に言えば、左右画像間におけるズレの限界値は人によって異なるが、その差は小さいと言える。このような場合、ステレオフォ



トビューアに導入する遠近ファクターの機能には、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、遠近的立体感において2重像の発生を防ぐ機能を持たせる必要がある。

## 5.2 問題点2(強い飛び出し効果)

立体写真と言えはかつては、飛び出し効果を狙ったものが多かったが、現在のように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、ステレオフォトビューア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感じられることがある。これは立体視感覚として強烈であり、閲覧者にとっては目に負担がかかりすぎると言える。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遠近ファクターの機能には、遠近的立体感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。

## 5.3 問題点3(ズームインなのに遠ざかる立体像)

実際のズームングの際には、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、というように感じられる現象がみられる。この現象は、立体写真の光景(コンテンツ)によっても影響を受けるようで、閲覧者によっては、注意深く観察しないと確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、平板に貼られた左右の画像におけるズレ(4.2節で述べた値 $d$ )の画面上での投影(4.2節で述べた間隔 $D$ )が、ズームインの場合には拡大、ズームアウトの場合には縮小していくことにある。これを解消しないと、立体写真のズームングはおかしなものになってしまう。これを防ぐ機能も遠近ファクターに持たせる必要がある。

# 6. 融像式の提案

これまでに、閲覧者が感じる遠近的立体感を遠近ファクターという概念的な言葉だけを用いて、概括的にしか説明してこなかった。そこで本章では、「遠近ファクターの意味する内容を数学的な式として定め、その式を遠近ファクターとして用いると、実際に目標とする位置に立体像を配置することが可能である」ということを示すことにする。この式を本論文では「融像式」と呼ぶが、この式により、第5章で示した2重像の発生(5.1節)、強い飛び出し効果(5.2節)、立体像の逆転した動き(5.3節)などを制御することが可能になる。最初に、融像式の前段であるズレの値を説明する。

## 6.1 融像の解析

### (1) 融像する立体像の位置

「画面上に投影された黒丸点と白丸点の間隔 $D$ 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(その座標系については後述する)」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3は、図2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。左画像と右画像に写っている光景において、それぞれの光景で互に対応する2つの点がダブってディスプレイ画面上に投影されている。このように左右画像には、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1組を取り上げている。左右の画像はそれぞれの平板に貼られているが、この対応する画像の2点間のズレの値を $d$ とする。

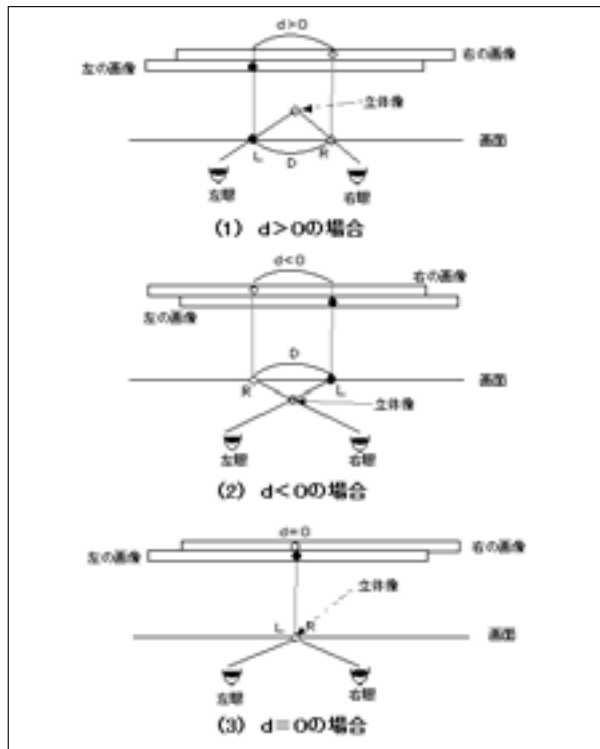


図3 左右画像から再生される立体像

図3では画面への投影について、左画像の点による投影点を黒丸点で、右画像による投影点を白丸点で示している。画面上の丸点間の間隔を $D$ とする。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、その両丸点の間隔 $D$ に基づいて立体像を融像する。

ズレの値 $d$ と間隔 $D$ の決定は、以下の説明する1次元座標系( $x$ 軸)で行う。図3(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左右画像間でのズレの値 $d$ とする。画像の白丸点が右側に移動すれば正の移動とし、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、ズレの値 $d$ は負値とする。従って、左右画像のズレの値 $d$ は正、零、負の値を取る。そして、画像のズレの値 $d$ は、画面上に両丸点間の間隔 $D$ 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、画面上の間隔 $D$ の1次元座標系( $x$ 軸)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、その正負の方向は左右画像におけるズレの値 $d$ の $x$ 軸座標と同じである。

閲覧者は、画面上に投影された黒丸点と白丸点の間隔  $D$  の値によって、立体像が近くに見えるか遠くに見えるかの遠近感を感じるのである。立体像の融像される位置は、図 4 に示す量を用いて幾何学的な比例関係から、以下の式で示すことが出来る<sup>(文献 1)</sup>。立体像の位置は、ディスプレイ画面上を基準位置として、ディスプレイ装置内部に向う方向を正、閲覧者に向う方向を負として、その位置  $u$  を定める。

$$u = \{D / (e - D)\} \times h \quad (e > D) \quad (\text{式 2})$$

ここで、値  $e$  は、閲覧者の瞳孔間隔(一般に人間は 6.5cm)、 $h$  は閲覧者とディスプレイ画面までの距離、 $D$  は画面上に投影された黒丸点と白丸点の間隔である。位置  $u$  の式は、間隔  $D$  に関して単調増加関数である。この式 2 は、図 4 の立体視に関する三角形において比例式を立てることにより、簡単に導くことが出来る。

式 2 から分かることは、閲覧者が融像する立体像の位置  $u$  は、閲覧者の距離  $h$  と瞳孔間隔  $e$  を固定すると、間隔  $D$  から定量的に求めることが出来るということである。間隔  $D$  が 0 から正值の方向に増加すれば、 $u$  は増加し、立体像は閲覧者側から奥まってしまう。間隔  $D$  が正值から 0 の方向に減少すれば、 $u$  は減少し、立体像は閲覧者側に近づいてくる。また、間隔  $D$  が 0 から負値の方向にその絶対値を増加すれば、 $u$  は負方向にその絶対値を増加し、立体像は閲覧者側に近づいてくる。間隔  $D$  が負値から 0 の方向にその絶対値を減少すれば、 $u$  は負値から 0 の方向にその絶対値を減少し、立体像は閲覧者側から奥まってしまう。

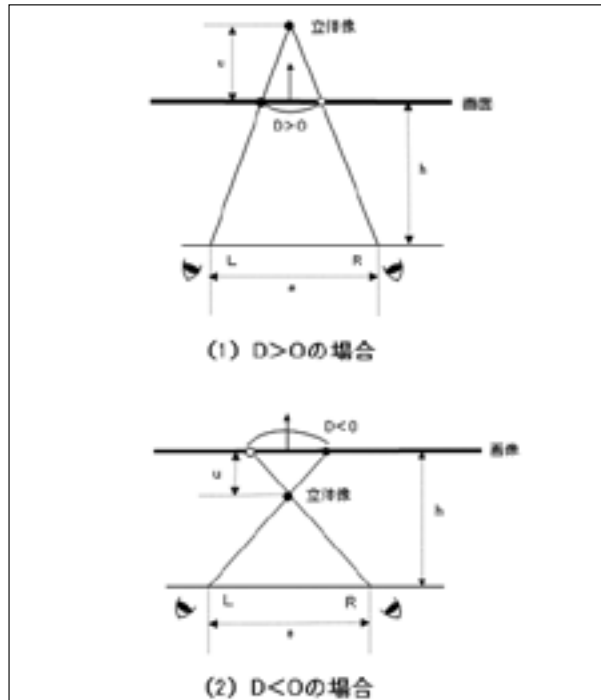


図 4 立体像の位置の推計図

## (2) 融像する立体像の大きさ

視認する立体像の大きさは、式2と同様に幾何学的手続きによって求めることができる。図5では、画面上に左から黒丸、白丸、黒丸、白丸が並んでいるが、2つの黒丸点間の画像部分と、2つの白丸点間の画像部分が、それぞれ左眼用と右眼用に対応し、その画面上の部分的実画像の大きさを  $r$  とする。この部分の実画像は、そのズレの間隔  $D$  によって立体像として視認される。 $u$  は立体像の奥行きである。視認される立体像の大きさを  $v$  とし、簡単な幾何学的な比例関係の計算を行うと、 $v$  は以下の式となる。

$$v = \{r \times (u + h)\} / h \quad (\text{式3})$$

2枚の左右の立体写真から視認される立体像の奥行きと大きさは、式2を借りて式3から決定することができる。

7.4節では、この式2と式3を用いて立体像が融像する位置と大きさをシミュレートすることによって、ズレの間隔  $D$  の立体像への影響の効果を論じることにする。なお、式2と式3は作図と比例式から導出されるが、人が立体視する場合には個人差があるので、実際には眼球の動き－輻輳－なども考慮する必要があると思われる。この論文で示した式2と式3の適用は、概念的なものとして考えて頂きたい。

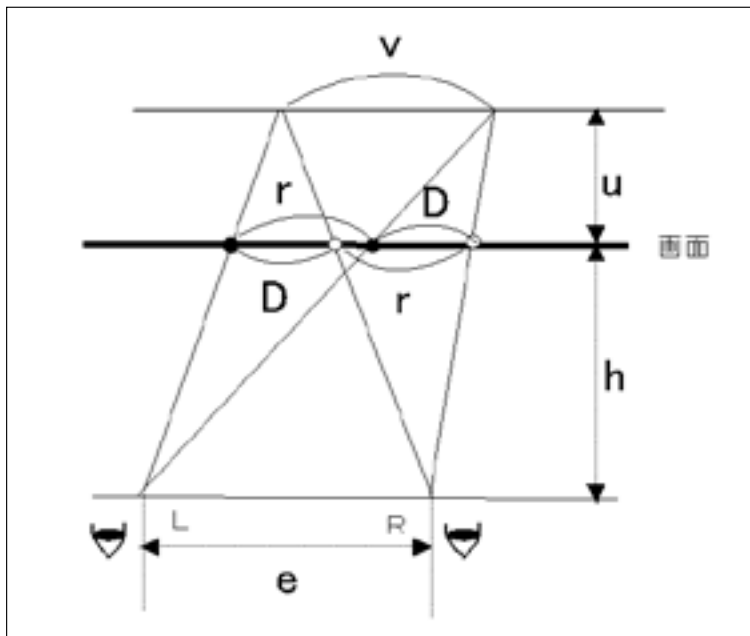


図5 立体像の大きさの推計図

## (3) ズレの値と遠近的感覚の関係

図3(1)に示すように、画像のズレの値  $d$  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、画面上の両

丸点の間隔  $D$  は正の値として増大する。そのため、融像する立体像の位置  $u$  は式 2 から正の値として増大するので、閲覧者は立体像が後退するように感じる。また、図 3(2)に示すように、画像のズレの値  $d$  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、画面上の両丸点の間隔  $D$  は負の値としてその絶対値を増大させる。そのため、融像する立体像の位置  $u$  は式 2 から負の値としてその絶対値を増大させるので、閲覧者は立体像が近づくように感じる。また、図 3(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、画像のズレの値  $d$  は 0 なので、画面上の両丸点の間隔  $D$  の値は 0 となる。そのため、融像する立体像の位置  $u$  は式 2 のから零となるので、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。このようなことは実験でも確認され、周知の事実である(文献 1)。

ここに示したように、画像のズレを増大・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が出来れば、それは遠近ファクターの数学的な式である融像式となるのである。

## 6. 2 融像式の導出

以上のことを踏まえ、「ディスプレイ画面に投影された画像の倍率  $m$ 」と「平板に貼られた 2 枚の左右画像間のズレの値  $d$ 」との量的関係を、以下に 4 段階に分けて議論する。そして、本論文の主題である融像式を導くことにする。

### (1) 第 1 段階( $d$ と $m$ の反比例関係)

まず 5. 3 節の問題点 3 に対応するために、投影された画像の倍率  $m$  を用いて、平板に貼られた左右画像のズレの値  $d$  を定量的な式で表わすことを試みる。4. 2 節の結論①、②で述べたように、ズームインして倍率  $m$  が  $n$  倍となる場合、2DCG における平行投影の原理から、ズレの間隔  $D$  は  $n$  倍の倍率になる。そして、画面上に投影された丸点間の間隔  $D$  は、そのときの倍率  $m$  のもとでズレの値  $d$  に比例して  $n$  倍に拡大される。その結果、6. 1 節で述べたように、ズームインが進行している最中に、閲覧者は立体像が本来は手前に向かって前進するところを、あたかも後退するように感じる。これを防ぐために、まず、立体像を特定の位置に停止させることを考える。それには、図 3 に示す画面上の丸点間の間隔  $D$  を、ズームインによって画像の倍率  $m$  が変化しても、一定とすることが必要である。またズームアウトする場合にも、丸点間の間隔  $D$  を一定とすることが必要である。

画面上に投影された黒丸点と白丸点の間隔  $D$  は、倍率  $m$  が一定のもとでズレの値  $d$  に比例し、正負の符号も同じある。この  $D$  を倍率  $m$  に関わらず一定にするには、 $d$  と  $m$  を反比例関係にすればよい。一定値を  $k_1$  とし、それを反比例定数とする。そして比例定数を  $k_2$  とすれば、

$$D = k_1 = k_2 \times d \times m \quad (k_2 > 0) \quad (\text{式 4})$$

となる。 $k_1 / k_2$  を  $a$  として上式を書き直すと、以下の  $d$  と  $m$  に関する反比例の式、

$$d = a \times (1 / m) \quad (a > 0) \quad (\text{式 5})$$

が得られる。 $D$  は、以下のように倍率  $m$  に関係なく一定値となる。

$$D = k_2 \times a \quad (\text{式 6})$$

なお、図3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  $a$  を正值とする理由は、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、一般に見にくくなる。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にした場合には、 $a$  は負値とする。

## (2) 第2段階(前進／後退ファクター)

式6から丸点間の間隔  $D$  は、画面の倍率  $m$  によらず一定の値を保つことから、ズーミングによって、立体像は後退も前進もしない。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の不自然さはそれほど感じられない。しかし、もう一つ改良を加えることにする。ズームインの場合には立体像が閲覧者に向かって迫ってきて、ズームアウトの場合には遠ざかるようになれば、もっと自然な動きとして閲覧者には感じられるはずである。そこで、以下のような「前進／後退ファクター」と呼ぶものを定義する。

$$f = (m_1 / m)^n \quad (\text{式 7})$$

このファクター  $f$  は、画面の倍率  $m$  が  $\infty$  (無限大) の時は0、倍率  $m$  が  $m_1$  の時は1の値をとる関数である。 $m_1$  は、「縮小限界倍率」と呼ぶべきものであり、この倍率  $m_1$  については第7章で述べる。冪乗の指数  $n$  は、立体像の前進／後退を調節するものであり、 $0 \leq n \leq 2$  の実数値を取る。この指数  $n$  についても第7章で述べる。閲覧可能な限界倍率までズームアウトすると、立体像は徐々に小さくなっていき、立体像全体が画面サイズよりも小さくなる倍率があるが、それが閲覧可能なズームアウトの限界倍率である。倍率  $m$  と縮小限界倍率  $m_1$  は、以下の関係がある。

$$m \geq m_1 \quad (\text{式 8})$$

前進／後退ファクターは、式の定数  $a$  に補正因子として作用するものである。この効果は後で述べる。このファクターを用いて、式5を以下のように補正することにする。

$$\begin{aligned} d &= (a \times f) \times (1 / m) \\ &= \{a / (1 / m_1)^n\} \times (1 / m)^{n+1} \end{aligned} \quad (\text{式 9})$$

これを「理想的なズレの式」と呼ぶ。また間隔  $D$  の式4に式9を代入すると、以下の間隔  $D$  に関する式が得られる。

$$\begin{aligned} D &= k_2 \times d \times m \\ &= k_2 \times (a \times f) \\ &= k_2 \times \{a / (1 / m_1)^n\} \times (1 / m)^n \end{aligned} \quad (\text{式 10})$$

式10から、指数  $n$  が1の場合の間隔  $D$  は、倍率  $m$  に反比例する(倍率  $m$  の逆数に比例する)こ

とが分かる。この式 10 を用いると、ズームイン操作の場合( $m$  が  $m_1$  から  $\infty$  に増大する場合)、その投影された両丸点の間隔  $D$  は正值から 0 に向かい、立体像は閲覧者に向かって前進して来る。また、ズームアウト操作の場合には( $m$  が  $\infty$  から  $m_1$  に減少する場合)、間隔  $D$  は 0 から正值に向かい、立体像は閲覧者から後退していく。前進／後退ファクターを導入し、それを補正因子として式 5 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なるように制御できるのである。指数  $n$  は立体像の動きを倍率  $m$  の逆数について非線形にするために導入された人為的なパラメータであるが、その効果は、第 7 章で述べる。

この段の最後に、式 9 の理想的なズレの式の定数  $a$  は、立体像の遠近感を調整するパラメータに成り得ることを示しておく。式 2 と式 10 から、画像の倍率  $m$  が一定の場合、立体像の奥行き位置  $u$  について、位置  $u$  は間隔  $D$  に関して単調増加関数であり(式 2)、間隔  $D$  は定数  $a$  に比例する(式 10)。そして、定数  $a$  は理想的なズレの式に基づくものなので(式 9)、その結果、立体像の位置  $u$  は理想的なズレの式の定数  $a$  に関しての単調増加関数となる。このことから、理想的なズレの式の定数  $a$  の値が増加すると、立体像の奥行きは増加し、定数  $a$  の値が減少すると、立体像の奥行きも減少することが分かる。なお、定数  $a$  の値は視覚実験から定めるが、これについては後述する。

### (3) 第 3 段階(理想的な融像式)

理想的なズレの式(式 9)を用いて、さらに次のような式を定める。

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

具体的には、次の式である。

$$\begin{aligned} t &= d + b = (a \times f) \times (1 / m) + b \\ &= \{a / (1 / m_1)^n\} \times (1 / m)^{n+1} + b \end{aligned} \quad (\text{式 12})$$

この式は、理想的なズレの式に「2 重像防止定数」と呼んでいる定数  $b$  を付加して拡張したもので、この式を「理想的な融像式」と呼ぶことにする。(この式は 2 重像の発生を防ぎ、閲覧者にとって最適な遠近的立体感を得られるような立体像を融像する役割の式である。このことから“融像式”と命名した)。この融像式が第 2 章で導入した概念的な遠近ファクターの数学的な表現である。

2 重像防止定数  $b$  は、2 重像の発生防止に関係するパラメータである。定数  $b$  は、式 9 の理想的なズレの式の値  $d$  を 0 とするときの、左画像用平板(位置は固定されている)と右画像用平板そのもの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、立体視可能にすることが出来るのであるが、その値は視覚実験から定める。

この定数  $b$  は、画像を撮影したときの 2 台のカメラのレンズ間隔に関係している。理想的なズレの式から算出される値  $d$  は、閲覧者が目視で確認できる画面上の両丸点の間隔  $D$  に対応する、左右画像におけるズレの値であるが、定数  $b$  は、左右画像における両丸点間でズレの値  $d$  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。それゆえ、定数  $b$  を含む融像式  $t$  は、第 10 章で述べる座標変換のための式として用いられる。(なお、撮影カメラのレンズ間隔は、定数  $b$  を視覚実験で決定するときに吸収されてしまう属性であるので、特に拘らなくてもよいものである)。

融像式から得られた最適なズレの値に従って右画像用平板の再配置を行えば、2重像の発生を抑え、閲覧者が最適な遠近的立体感を感じられるような立体像を融像することが可能となるのである。

☆

☆

☆

第3段階で導いた理想的な融像式の定数  $a$ 、 $b$  を決定するには、以下の2つの方法が考えられる。

①個別決定法：画面上に投影された画像の倍率  $m$  を限りなく大きな値に近づけ(ズームインして)( $m$ が正値 $\rightarrow\infty$ )、その投影された左右の画像(黒丸点と白丸点)がズレなく重なるように、視覚実験によって定数  $b$  の値を決定する。その後、定数  $a$  について、ズームアウトして( $m$ が正値 $\rightarrow 0$ )、立体像の奥行きが最良になるように視覚実験によって決定する。この決定法については、第8章で詳し述べる。

②連携決定法：融像式についての倍率  $m$  に関して、ある倍率  $m_a$  を定め、最良に立体像が生成するような変量  $t_a$  を視覚実験によって決定する。再度別の倍率  $m_b$  を定め、そのとき同様に最良の立体像が生成するような変量  $t_b$  を同様な視覚実験によって決定する。そして、融像式の定数  $a$ 、 $b$  を変数とする2元連立方程式を立て、その方程式を解く。しかし、この決定法については、倍率  $m_a$ 、倍率  $m_b$  が任意の倍率であれば、それらの倍率において最良の立体像が確認できても、その組み合わせから2重像を発生しない定数  $b$  の値と最適な遠近感の立体像を生じさせる定数  $a$  の値が得られるかは、試行錯誤となる。

しかし、上述の2つの方法は、ともに問題がある。①の個別決定法については、倍率  $m$  を限りなく上げていくと( $m$ が正値 $\rightarrow\infty$ )、画像の解像度の限界を超えて倍率が理論上無限大となってしまう、立体像の融像が困難となる。②の連携決定法に関しては、方程式の解法をステレオフォトビューア内部で自動的に処理するにしても、閲覧者の実験操作と解法手続きの連携作業が面倒となる。というのは、2つの定数  $a$ 、 $b$  を決定する視覚実験をそれぞれ相互に関連付けて行った後に、連立方程式の解を求めるという手順が、閲覧者に制約を課すことになり、使い勝手のよい立体視ツールにはならない。この場合、定数  $a$ 、 $b$  の融像式における役割の意味付けもし難くなる。なお、倍率  $m_a$  と倍率  $m_b$  を、後述する拡大限界倍率  $m_0$  と縮小限界倍率  $m_1$  と規定すれば、それは個別決定法と同じになってしまう、連携決定法の意味はなくなってしまう。

#### (4) 第4段階(実用的な融像式)

第3段階で述べた問題点を回避するために、閲覧者が簡単に定数  $a$ 、 $b$  を決定できる融像式に改良してみる。この融像式は理想的な融像式の改良式ではあるが、これを用いると、2つの定数  $a$ 、 $b$  の決定において、個別決定法を簡単に適用できるようになる。その理由は、①画像の倍率無限大の問題を回避でき、②視覚実験で定数  $a$  と  $b$  をそれぞれ単独に決定でき、③その決定の際の定数  $a$ 、 $b$  の役割についての意味付けも明確になるのである。また、閲覧者にとっては実験の制約もなく、作業がワンパターンになり軽減される利点がある。そこで、個別決定法を適用できるようにするために、理想的な融像式を改良してみる。

まず、画面上の画像の拡大を最大にする倍率  $m_0$  を導入しよう。これは、「拡大限界倍率」と呼ぶものであり、画像の解像度によって決まる倍率である。この倍率  $m_0$  については第7章で述べる。拡大限界倍率  $m_0$  と倍率  $m$  とは、以下の関係がある。

$$m \leq m_0$$

(式 13)



その理由は、倍率  $m$  が拡大限界倍率  $m_0$  を超えると、画像の解像度の限界を超えているので、生成される立体像は意味を持たなくなるからである。拡大限界倍率  $m_0$  は立体写真のカメラの解像度特性に依存する。

拡大限界倍率  $m_0$  は視覚実験では、縮小限界倍率  $m_1$  の10倍～20倍程度の値として決定される。この拡大限界倍率  $m_0$  を用いて、理想的な融像式(式9)を実験的に取り扱いやすい融像式に変形してみよう。

☆

☆

☆

第1段階で定めた式5において、以下のように倍率  $m$  の逆数  $1/m$  を  $(1/m - 1/m_0)$  で置き換える。この置き換えをするのは、倍率  $m$  が拡大限界倍率  $m_0$  でズレの式の値  $d$  を0とし、6.1節における図3(3)の状況を作り出すようにすることである。これによって、黒丸点と白丸点は画面上で重なり合うことになる。

$$d = a \times (1/m - 1/m_0) \quad (\text{式14})$$

それに伴い、前進/後退ファクター  $f$  (式7) を以下のように変更する。

$$f = \{(1/m - 1/m_0)/(1/m_1 - 1/m_0)\}^n \quad (\text{式15})$$

変更されたファクター  $f$  は、倍率  $m$  が  $m_0$  の時は0、倍率  $m$  が  $m_1$  での時は1の値とる関数である ( $m$  が  $m_1 \leq m \leq m_0$  の範囲で、 $f$  は  $0 \leq f \leq 1$  の範囲にある)。 $m_1$  は、6.2節(2)項で示した縮小限界倍率である。冪乗の指数  $n$  は、立体像の前進/後退を調節するものであり、 $0 \leq n \leq 2$  の実数値を取る。この指数の効果については、第7章で述べる。

そして、以下のように、式14の定数  $a$  に補正因子として式15の前進/後退ファクター  $f$  を作用させる。

$$\begin{aligned} d &= (a \times f) \times (1/m - 1/m_0) \\ &= \{a / (1/m_1 - 1/m_0)^n\} \times (1/m - 1/m_0)^{n+1} \end{aligned} \quad (\text{式16})$$

この式を「実用的なズレの式」と呼ぶが、以降、単にズレの式と言え、この式を指すものとする。そして、第3段階で得たのと同じ方法で、新しい融像式を以下のように求める。

$$\begin{aligned} t &= \text{「実用的なズレの式」} + \text{「2重像防止定数」} \\ &= d + b \\ &= (a \times f) \times (1/m - 1/m_0) + b \\ &= \{a / (1/m_1 - 1/m_0)^n\} \times (1/m - 1/m_0)^{n+1} + b \end{aligned} \quad (\text{式17})$$

新しく定められた式17の融像式は、拡大限界倍率  $m_0$  がかなり大きい場合、式12の理想的な融像式に関数として近づく。この式17を以降、「実用的な融像式」と呼ぶことにする。以降、単に融像式と言え、この実用的な融像式を指すものとする。

実用的な融像式を用いた場合、画面上の両丸点の間隔  $D$  は、式4と式16の上段式と式15

を用いると、以下の式が得られる。

$$\begin{aligned}
 D &= k_2 \times d / (1 / m) \\
 &= k_2 \times (a \times f) \times (1 / m - 1 / m_0) / (1 / m) \\
 &= k_2 \times \{a / (1 / m_1 - 1 / m_0)^n\} \\
 &\quad \times \{(1 / m - 1 / m_0)^n \times (1 - m / m_0)\} \quad (\text{式 18})
 \end{aligned}$$

間隔  $D$  に関する式 18 の中段式は、定数  $a$  に前進／後退ファクターを補正因子として掛けたものとなる。式 18 によって求められる間隔  $D$  は、倍率  $m$  に関して単調減少関数である。また、それは指数  $n$  が 0 のとき、倍率  $m$  に関しての一次式となる。

## 7 実用的な融像式の特徴

### 7.1 融像式の役割

実用的な融像式の役割は、6.2 節の第 1 段階から第 3 段階まで述べてきた理想的な融像式における役割と基本的には同じである。以下にその役割をまとめて述べてよう。実用的な融像式と理想的な融像式の違いは、前者への拡大限界倍率  $m_0$  の導入である。実用的な融像式において拡大限界倍率  $m_0$  の逆数  $(1 / m_0)$  を 0 とおけば、その融像式は、理想的な融像式となる。

#### (1) 2 重像の発生を防ぐ役割は、融像式の定数 $b$ が担う

2 重像の発生を防止する役割は、実用的な融像式の定数  $b$  が担っている。定数  $b$  は視覚実験によって、拡大限界倍率  $m_0$  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 5 章の問題点 1 を解決することが出来る。パラメータとしての定数  $b$  は個々の立体写真毎に定めるが、その決定法は 8.2 節(1)項で述べる。

#### (2) 立体像の位置を調整する遠近ファクターは、融像式の定数 $a$ が担う

式 17 の実用的な融像式の定数  $a$  は、立体像の遠近感を調整するパラメータとなる。式 2 と式 18 から、倍率  $m$  が一定の場合、立体像の奥行き位置  $u$  について、位置  $u$  は間隔  $D$  に関して単調増加関数であり(式 2)、間隔  $D$  は定数  $a$  に関して単調増加関数である(式 18)。そして、定数  $a$  は実用的なズレの式に基づくものなので(式 16)、その結果、立体像の位置  $u$  は実用的なズレの式の定数  $a$  に関しての単調増加関数となる。このことから、実用的なズレの式の定数  $a$  の値が増加するに従って位置  $u$  の値は増加し、それに伴い立体像の奥行きは増加する。また、定数  $a$  の値が減少するに従って位置  $u$  の値は減少し、それに伴い立体像の奥行きも減少する。第 2 章で述べた遠近ファクターの実質的な役割は、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、縮小限界倍率  $m_1$  で立体像の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 5 章の問題点 2 を解決することが出来る。パラメータとしての定数  $a$  は個々の立体写真毎に定めるが、その決定法は 8.2 節(2)項で述べる。

#### (3) 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、実用的な融像式  $t$  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  $t$  と連動する式 18 の間隔  $D$  の式が担っている。従って、この間隔  $D$  の式を用いると、閲覧者に視認される立体像の奥行きや立体像の倍率をシミュレートすることが可能となる。これについては、7.4 節で詳しく論じる。

#### (4) 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 の間の実数を取る定数である ( $n$  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)。実用的な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、その冪乗の指数  $n$  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／後退の効果は倍率  $m$  に反比例するが如く、 $m$  が小さくなればなるほどより強くなる。しかし、立体視感覚は個人差があり、人によっては、前進／後退の効果が強いと感じたり、弱いと感じたりすることがあるかもしれない。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。指数  $n$  の導入によって第 5 章の問題点 3 を解決することができる。

#### (5) 拡大限界倍率 $m_0$ と縮小限界倍率 $m_1$ は、ステレオフォトビューアの定数とみなし得る

拡大限界倍率  $m_0$  と縮小限界倍率  $m_1$  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。前者はズームインの極限での倍率であり、カメラの解像度に関係する値であるが、その値は、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ものと思われる。それ故、個々の立体写真毎に拡大限界倍率  $m_0$  を定めることも考えられる。しかし、事前の視覚実験によれば、一般的に使用されるデジタルカメラの間では同じ値を使用しても差し支えない場合が多い。従ってこの値は、ステレオフォトビューアの開発者によって定める値としてもよい。また後者は、ズームアウトの極限での倍率である。ズームアウトすることによって、立体像は徐々に小さくなっていき、立体像全体が画面サイズよりも小さくなる位置があるが、それが縮小限界倍率  $m_1$  であり、閲覧可能なズームアウトの限界倍率といえる。倍率を云々する場合には、基準の倍率を定めなければならないが、ステレオビューアにおいては、縮小限界倍率  $m_1$  を基準倍率とし、その倍率を 1 と定めることにする。しかし、最小限界倍率は説明の都合上、以降もこの倍率を  $m_1$  と記載する。

ズーミング可能な倍率の範囲は、縮小限界倍率  $m_1$  から拡大限界倍率  $m_0$  までの間の倍率であり、事前の視覚実験によれば、通常、一般的に使用されるデジタルカメラを使用する場合には、拡大限界倍率  $m_0$  は縮小限界倍率  $m_1$  の 10 ～ 20 倍程度の倍率である。

### 7.2 融像式のグラフ表現

式 17 の実用的な融像式の様子をグラフで示そう。画面上の画像の倍率  $m$  が拡大限界倍率  $m_0$  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、

$$\begin{aligned} t_0 &= \{a / (1 / m_1 - 1 / m_0)^n\} \times (1 / m_0 - 1 / m_0)^{n+1} + b_0 \\ &= b_0 \end{aligned}$$

である。また、倍率  $m$  が縮小限界倍率  $m_1$  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、

$$\begin{aligned} t_1 &= \{a_1 / (1/m_1 - 1/m_0)^n\} \times (1/m_1 - 1/m_0)^{n+1} + b_0 \\ &= a_1 \times (1/m_1 - 1/m_0) + b_0 \end{aligned}$$

である。

拡大限界倍率  $m_0$ 、縮小限界倍率  $m_1$ 、および上記の  $t_0$ 、 $t_1$  を用いて、融像式の様子の一例を図 6 にグラフで示す。前進／後退ファクターの冪乗の指数  $n$  を持つ融像式は、点  $(m_0, t_0)$ 、点  $(m_1, t_1)$  を通る関数である。また、前進／後退ファクターの冪乗の指数  $n$  が 0 であるの場合の融像式は、双曲線関数となる。

なお、画面上に投影された画像の倍率  $m$  の取り得る範囲(ズーミング可能な範囲)は、式 8 と式 13 から、

$$m_1 \leq m \leq m_0 \quad (\text{式 19})$$

である。

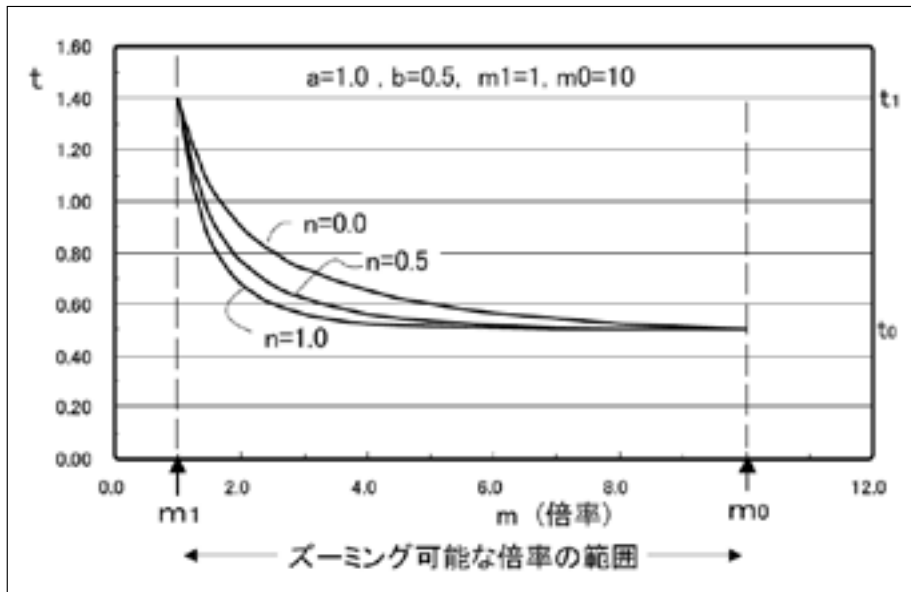


図 6 実用的な融像式のグラフ表現

### 7.3 融像式の働き

実用的な融像式の動作を、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、実用的なズレの式  $d$

$$d = \{a / (1/m_1 - 1/m_0)^n\} \times (1/m - 1/m_0)^{n+1} \quad (\text{式 16 の下段式の再掲})$$

の動作を説明することになる。

#### (1) 値 $d > 0$ の場合

定数  $a$  が正值の場合である。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3(1)のようにディスプレイ画面の後方に融像する。そして、マウスによるズーミング操作によって、倍率を上げる( $m$ を $m_1 \rightarrow m_0$ )と、ズレの式の値  $d$  は正值から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18によって正值から0に近づく。その場合、立体像の位置を定める式2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。逆に倍率を下げる( $m$ を $m_0 \rightarrow m_1$ )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2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

融像式におけるズレの値  $d$  が負値というのは、定数  $a$  が負値である場合である。これまで融像式の定数  $a$  は正值を採用すると述べてきたが、定数  $a$  が負値の場合には、次のような不都合が生じる。画面上に投影された左右画像の黒丸点と白丸点の位置が7.3節(1)項とは逆になり、この場合には、図3(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、マウスのズーミング操作による立体像の動きは、7.3節(1)項とは反対であり、倍率を上げる( $m$ を $m_1 \rightarrow m_0$ )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18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2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。逆に倍率を下げる( $m$ を $m_0 \rightarrow m_1$ )と、間隔  $D$  は0から負値になっていく。その場合、立体像の位置を定める式2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  $a$  は正值を採用する必要がある(ただし、左画像を固定した場合である)。

#### (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、倍率  $m$  は  $m_0$  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ある。この場合、図3(3)のように立体像はディスプレイ画面上で融像する。

### 7.4 立体像のシミュレーション解析

#### (1) 立体像の奥行きと大きさの解析

これまで考察した式を用いると、閲覧者が視認する立体像の奥行き位置と像の大きさを定量的に求めることができる。立体像の奥行き位置の計算には、画面上における左右画像のズレの間隔  $D$  (式18の下段の式)と立体像の位置  $u$  (式2)を用いる。以下にその式を再掲する。

$$D = k_2 \times \{a / (1 / m_1 - 1 / m_0)^n\} \times \{(1 / m - 1 / m_0)^n \times (1 - m / m_0)\} \quad (\text{式18の下段式の再掲})$$

$$u = \{D / (e - D)\} \times h \quad (e > D) \quad (\text{式2の再掲})$$

また、立体像の大きさ  $v$  には、式 3、および上記の 2 つの式を用いる。以下に式 3 を再掲する。

$$v = \{r \times (u + h)\} / h \quad (\text{式 3 の再掲})$$

ただし、ステレオフォトビューアにおいては、ズーミングを行うので、立体像の大きさ  $v$  は、実画像の倍率  $m$  の変化も考慮する必要がある。そこで、実画像の倍率  $m$  の変化に対して立体像の大きさ  $v$  がどのように変化するかを調べることにする。実画像の倍率が  $m_1$  のとき、実画像部分の大きさ（基準長）を  $r_1$  とすると、倍率  $m$  のときその実画像部分の大きさは、

$$r = (r_1 / m_1) \times m \quad (\text{式 20})$$

と、 $r$  の大きさに拡大されるので、実画像の倍率が  $m$  のとき、立体像の大きさ  $v$  を求めるには、式 3 を以下のように書き換えればよい。

$$v = [\{(r_1 / m_1) \times m\} \times (u + h)] / h \quad (\text{式 21})$$

実画像の倍率に対する立体像の倍率を  $b$  とし、その倍率基準を実画像の倍率  $m_1$  のとき、実画像の大きさ（基準長）を  $r_1$  とすれば、立体像の倍率  $b$  は、次式のように、式 21 から求められる立体像の大きさ  $v$  を  $(r_1 / m_1)$  で除算すればよい。

$$\begin{aligned} b &= v / (r_1 / m_1) \\ &= \{m \times (u + h)\} / h \end{aligned} \quad (\text{式 22})$$

計算のパラメータは、閲覧距離  $h$  は 50.0cm、瞳孔間隔  $e$  は 6.5cm、基準長  $r_1$  を 1.0cm、縮小限界倍率  $m_1$  は 1.0、拡大限界倍率  $m_0$  は 10.0 とした。式 18 の定数  $k_2 \times a$  は、17 型立体ディスプレイ (Dimen-G170P) を用いた場合に倍率  $m_1$  の際、画面上でのズレの間隔  $D$  が 1 cm となるように 1.2cm を採用した。また、前進／後退ファクターの冪乗の指数  $n$  は 0.0、0.5、1.0 および 2.0 の 4 通りとした。その計算結果を図 7 に示す。実線が立体像の倍率  $b$  の計算グラフ ( $n$  が 0.0 の場合) であり、破線が奥行き  $u$  の計算グラフ ( $n$  が 0.0、0.5、1.0 および 2.0 の 4 通りの場合) である。なお、シミュレーションとともに、自動ズーミングを行いながら視覚実験も行った。

計算されて求められた立体像の倍率  $b$  は、実画像の倍率  $m$  と大まかには一致しているといつてよい。仔細に精査すると、 $n$  が 0.0 の場合は、非線形性が僅かみられる程度の単調増加の変化をするが、 $n$  が 1.0 に近づくに従って非線形性は非常に弱まり、立体像の倍率  $b$  は実画像の倍率  $m$  に近づく（実画像の倍率が 1.0 のとき、立体像の倍率はやや大きめの 1.20 となるが、実画倍率が 10.0 になるに従って、立体像の倍率  $b$  は滑らかな単調増加で 10.0 になっていく。 $n$  が 0.5、1.0、および 2.0 の場合も同様の变化傾向にある）。4 つの計算値は図全体から眺めると接近していて、4 本の実線はくっついてしまうので、図 7 には  $n$  が 0.0 の場合のみが示されている。なお、シミュレーションによると、実画像の倍率 1.0 に対して、立体像の倍率の計算値がやや大きめとなる（視覚実験でも計算に近い大きさの立体像が視認される）。しかし、立体像の倍率はズーミングの時間進行に沿って滑らかに変化するので、倍率がやや大きめに算定され、立体像がその大きさに視認されることにつ

いては、視覚実験から閲覧者が特に違和感を感じるということとはなかった。

次に奥行き  $u$  であるが、図7に示すように、それは倍率  $m$  の変化に対して非線形性は顕著である。 $n$  が 0.0 の場合には、概略的には線形的であり、立体像の奥行き変化の動きは滑らかに進行する(ただ、ズーム速度が高倍率になるに従って遅くなるという難点がある)。 $n$  が 2.0 に近づくに従って大きく湾曲した曲線となり、非線形性が顕著になってくる。すなわち、ズームインでは閲覧者にとって、倍率  $m$  が 1.0 から 4.0 付近までの間に、立体像は急激に迫ってきて、その後は緩やかに近づいてくるように感じられる。

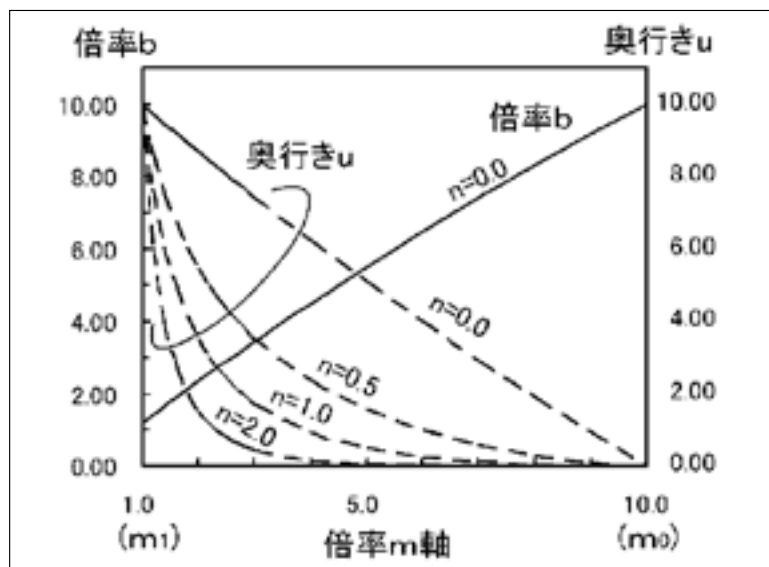


図7 立体像の位置と大きさの計算グラフ

ところで、立体像の倍率  $m$  とその奥行き位置  $u$  の関係は、倍率は2倍になれば、奥行き位置は2分の1となるような反比例の関係がなければならない。そのような関係をほぼ満たす指数  $n$  は、詳細なシミュレーション解析から、0.7 付近の値であることが分かった。この値は上述のように、立体像の倍率の推計にも矛盾無く使用することが出来る。そこで、前進／後退ファクターの指数  $n$  として0.7を用いることとした。しかし、立体像の視認には一般的に個人差があるといわれているので、多数の閲覧者による視覚実験の結果から、心地よい視認を促す平均的な指数  $n$  の値を求めてもよいと思われる。

なお参考のために、計算された立体像の倍率  $b$  と奥行き位置  $u$  を、表1に掲げておく。計算は指数  $n$  を 0.2、0.7、1.5 としている。表中の大文字の  $M$  と  $L$  は、基準とする実画像の倍率  $M$ 、およびそのときの実画像における対象物までの距離(対象物距離)  $L$  である。対象物距離とは、 $M \times L =$  一定値という反比例関係から求められる計算上の仮想距離である。立体像の倍率  $b$  は、実画像の倍率が10倍に対応する立体像の倍率(計算上は10.0)を比較基準とし、また立体像の奥行き位置  $u$  は、実画像の倍率が1倍のときの立体像の奥行き位置を1.0に換算し、その値を比較基準としている。 $L \times M$  の値が10.0となるように計算を行い、指数  $n$  が0.7のとき  $u \times b$  が10前後となる値が得られた。本論文に示した計算モデル(方法)からすると、その結果はほぼ満足すべきものと言えよう。

表1 計算された立体像の倍率と奥行き位置

		n = 0. 2			n = 0. 7			n = 1. 5		
L	M	u	b	u×b	u	b	u×b	u	b	u×b
1.00	10.0	1.00	10.0	10.0	1.00	10.0	10.0	1.00	10.0	10.0
2.00	5.00	3.86	5.32	20.5	1.91	5.10	9.80	1.15	5.02	5.80
3.00	3.33	5.53	3.67	20.3	3.03	3.48	10.5	1.59	3.38	5.40
4.00	2.50	6.65	2.86	18.7	4.10	2.67	10.9	2.24	2.57	5.70
5.00	2.00	7.49	2.29	17.1	5.13	2.18	11.2	3.07	2.09	6.40
6.00	1.67	8.16	1.93	15.7	6.13	1.86	11.4	4.07	1.78	7.20
7.00	1.43	8.72	1.67	14.6	7.11	1.62	11.5	5.26	1.56	8.20
8.00	1.25	9.20	1.48	13.6	8.08	1.45	11.7	6.63	1.41	9.30
9.00	1.11	9.62	1.32	12.7	9.04	1.31	11.8	8.20	1.29	10.6
10.0	1.00	10.0	1.20	12.0	10.0	1.20	12.0	10.0	1.20	12.0

## (2) 前進／後退ファクターの指数nの関数化

図7で説明したように、前進／後退ファクターの冪乗の指数nの定め方によって、閲覧者に視認される立体像の奥行き位置(立体視的に視認される奥行き位置であり、実画像の倍率から推測認知する対象物距離(7. 4節(1)項)ではないことに注意されたい)は変わってくる。特に自動ズーミングを行う場合には、閲覧者に対する立体像の迫り方や後退の仕方は、重要である。そこで、それを簡単に取り扱えるように、指数nを実画像の倍率mの関数として表わすことが出来るようにしてみよう。この関数は、理論から導かれるものではなく、経験的に定められるものである。

図7の解析から分かったことは、①nの値が小さい場合には、実画像の倍率mを $m_1$ から $m_0$ に変化させながらズームインすると、立体像の奥行き位置は、最初は緩やかに閲覧者に向かって近づき、その後に急速に迫ってくるように視認される(ズームアウトの場合には、その逆である)。また、②nの値が大きい場合には、実画像の倍率mを $m_1$ から $m_0$ に変化させながらズームインすると、立体像の奥行き位置は、最初は急速に閲覧者に向かって迫り、その後は緩やかに近づいてくるように視認される(ズームアウトの場合には、その逆である)。

このことから、ズーミングにおいて立体像の迫り方や後退の仕方を変化させるために、以下のような式を導入することができる。そして、その式の様子を図8に掲げる。

$$\begin{aligned}
 &\text{①の場合の指数 } n \text{ の増加関数} && \text{(式 23)} \\
 &n = \{2 / (m_0 - p)\} \times (m - p) && (m \geq p) \\
 &n = 0 && (m < p)
 \end{aligned}$$



②の場合の指数  $n$  の減少関数

(式 24)

$$n = \{-2 / (m_0 - p)\} \times (m - p) + 2 \quad (m \geq p)$$

$$n = 2 \quad (m < p)$$

$p$  は、関数の形状を調整するパラメータで、 $m_1 \leq p \leq m_0$  の範囲の実数をとる。①の場合の式 23 は、図 8 に示すように、指数  $n$  は実画像の倍率  $m$  の変化に従って 0 から 2 に変化し、②の場合の式 24 は、指数  $n$  は実画像の倍率  $m$  の変化に従って 2 から 0 に変化する。

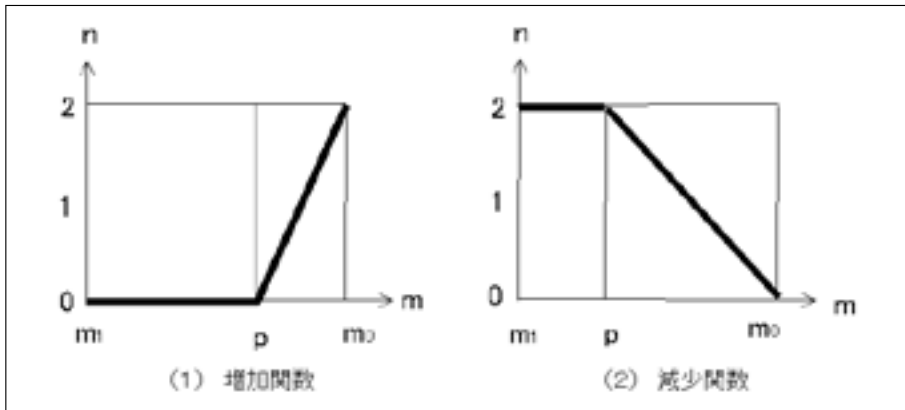


図 8 指数の関数化のグラフ

指数  $n$  を変化させるこれらの式を用いると、 $n$  を固定する場合とは異なり、自動ズーム過程において、立体像の迫り方や後退の仕方を強制的に変化させることが可能となる(パラメータ  $p$  を変化させることによって、その微調整ができる)。事前の視覚実験では、その効果を確認することができた。ステレオフォトビューアにこのような式を実装すれば、例えば立体像が閲覧者の目の前で急に迫ってくるような効果を演出することが可能となるのである。なお、上記の式は一例であり、ここで用いた一次関数以外の関数(正弦関数や余弦関数、あるいは 2 次関数など)を用いることも可能である。

## 8. 実用的な融像式による最適な遠近的立体感の設定

実用的な融像式の定数  $a$ 、 $b$  は、簡単な視覚実験(閲覧者の目視による判定実験)で決定されるパラメータである。これらはステレオフォトビューアにおける立体写真 1 枚 1 枚についての立体視特性を定めるものであり、その値は閲覧者の遠近的立体感を最適にするように決められる。ただし、パラメータ  $a$  は正の値とする。パラメータ  $b$  は正、負、零のいずれかの値をとる。その決定方法を本章で述べる。

### 8. 1 融像式の特性を決定するパラメータ

5. 1 節の問題点 1 のように、倍率  $m$  を大きくして画像を拡大した場合、2 重像が生じるのは、投影された左右画像のズレの間隔  $D$  が大きく拡大されることから(これは虫メガネで微細部分が拡

大されて見えるのと同じである)、融像の崩壊が生じるのである。これを防ぐには、まず、実用的な融像式のパラメータ  $b$  の値を適切に設定して対処する必要がある。それゆえ、このパラメータ  $b$  を「2重像防止定数」と呼ぶ。

また、第5章の問題点2のように、ディスプレイ画面から立体像の全体が飛び出すように感じられる場合は、倍率  $m$  が縮小限界倍率  $m_1$  に近い値であっても、パラメータ  $a$  に0に近い値(ただし  $a > 0$ )が設定されていると、融像式を構成するズレの式の値  $d$  の寄与が小さくなる。その結果、遠近の立体感による立体像が近くに感じられ、目にきつい立体像となってしまうのである。これを防ぐには、パラメータ  $a$  の値を適切に設定する必要がある。それゆえ、この定数を「遠近感制御定数」と呼ぶ。

このようにパラメータ  $a$ 、 $b$  は融像を司る重要なものである。そこで以下に、これらのパラメータをどのように設定したらよいかについて詳しく述べる。

## 8. 2 融像式のパラメータを設定する方法

閲覧者がマウスによるズーミング操作を行って、画面上の画像の倍率  $m$  を変化させると、実用的な融像式に従って右画像用平板の位置が  $x$  軸上で値  $t$  だけ平行移動する機構(ただし左画像平板の位置は固定されているものとする)、および、その時の値を読み取り保存する機構が、ステレオフォトビューアに備わっているものとする。パラメータの決定法は、まずパラメータ  $b$  を定め、その後にパラメータ  $a$  を決定するという、6. 2 節(3)項で述べた個別決定法を実用的な融像式に対して採用する。パラメータの設定は個々の立体写真毎に行う。

なお、パラメータを決定する際には、画像の領域や画像に写っている景観の遠近が関係する。そこで、その領域や遠近について本論文で言及する次の用語を定めておく。「全景領域」、「部分領域」、「近景領域」、「遠景領域」、の各用語である。「全景領域」とは、縮小限界倍率  $m_1$  までズームアウトして画面全体に映し出された画像の全体領域を言い(倍率  $m_1$  がズームアウトの限界であり、その倍率が1.0であることは、7. 1 節(5)で述べた)、写真の全景観が映し出される。「部分領域」とは全体に対する部分であり、画面に映し出された画像の一部分の領域を言う。また写真には、当然のことながら例えば、遠くの山並み景観、近くの木々などの風景、その中間の位置にある家の景色などが混在して、写真全体の景観が構成される。画像の中で遠くの景色が写っている領域を「遠景領域」、近くの風景が写っている領域を「近景領域」と言う。「遠景領域」と「近景領域」は、写真全体の一部を構成するので、「全景領域」に対する「部分領域」である。また「最近景領域」と言った場合には、それは近景とみなされる部分の中で最も近景となる部分領域を指す。例えば町並みを背景として3人が奥から手前に並んだ人物写真においては、画面上の画像では、背景の町並み部分が遠景領域、3人の人物の塊り部分が近景領域、そして近景領域の中で一番手前の人物の写った部分が最近景領域となる。また同様に、「最遠景領域」と言った場合も、遠景とみなされる部分の中で最も遠景となる部分領域を指すものとする。

### (1) パラメータ $b$ の決定

パラメータ  $b$  の定め方を述べよう。ある立体写真  $A$  について、閲覧者は画面に映し出されている画像のズーミング対象とする任意の領域を選択し、その領域に対して画像の倍率を拡大限界倍率  $m_0$  までにズームインする。任意の領域とは、画像における近景領域でも遠景領域でも、あるいは中間の領域でもかまわないが、特に小さく写っている遠方の景観をズームインして立体視閲覧した

いのであれば、希望する最遠景領域を選択する。そして、閲覧者はディスプレイ画面を眺め、左画像の当該領域と右画像のそれに対応する領域との間でズレが生じている場合には、そのズレを解消すべく、右画像用平板を  $x$  軸上で平行移動して、右画像の領域と左画像の領域をズレなく重ね合わせて、図3(3)に示す状態の如く画面上の位置に立体像が視認出来るようにする(第5章の問題点1への対策)。この立体像の確認は閲覧者による目視実験で行う。画面上の位置に立体像が生じるように、パラメータ  $b$  の値を変化させながら目視で調整する。調整されたそのときのパラメータ  $b$  の値を  $b_0$  とし、その値を用いて右画像用平板を  $x$  軸上で平行移動する値  $t_0$  は、融像式を用いて、

$$t_0 = \{a / (1 / m_1 - 1 / m_0)\}^n \times (1 / m_0 - 1 / m_0)^{n+1} + b_0 = b_0$$

となる。この値  $t_0$  を用いて右画像用平板の位置を再配置するのである。

パラメータ  $b$  の値  $b_0$  は、立体写真 A に関する固有のパラメータとなり、ステレオフォトビューアの内部に保存されるものとする。このようにしてパラメータ  $b$  が定められると、画像の拡大限界倍率  $m_0$  では、左右画像に関する(融像式の中の)ズレの式の値  $d$  は 0 なので、その位置では図3(3)に示すように、2重像の発生のない立体像が融像する。

## (2) パラメータ $a$ の決定

続いてパラメータ  $a$  の定め方を述べよう。第5章の問題点2を解決するには、立体像の全体を画面上あるいはそれよりも後方に位置するように設定することである。パラメータ  $b$  がすでに定められているので、拡大限界倍率  $m_0$  においてズレのない状態が実現されている。閲覧者は、この状態から倍率を縮小限界倍率  $m_1$  までズームアウトする。すなわち、画面に画像の全景領域が映し出されるようにする。

この縮小限界倍率  $m_1$  において、パラメータ  $a$  の値を定める目視実験を行う。それは、左右画像によって融像する立体像の具合を、閲覧者が目視実験で判断する。この実験とは、パラメータ  $b$  の定まった融像式において、パラメータ  $a$  の値を変化させながら、閲覧者にとって最適な立体感が感じられる立体像を融像するように(第5章の問題点2への対策)、パラメータ  $a$  の値( $a > 0$ )を目視で調整するのである。このとき閲覧者が感じる立体感が遠近的立体感であり、このパラメータ  $a$  の設定が遠近ファクターの設定ということになる。

具体的には次のように行う。立体写真 A について、縮小限界倍率  $m_1$  までズームアウトした全景領域において、閲覧者は立体ディスプレイ画面を眺めて、パラメータ  $a$  の値を変化させながら(ただし  $a > 0$  である)、左画像の当該領域と右画像のそれに対応する領域から再生される立体像の全体が、立体ディスプレイ画面上の位置あるいはそれよりも後方の位置で視認出来るようにする。これは、当該立体像の立体空間が図1の真ん中の図に示されるような“画面後方の立体空間”にすることである。この作業は「閲覧者が、縮小限界倍率  $m_1$  で縮小表示された全景領域の中で最も近景となる部分領域すなわち最近景領域に注目して、パラメータ  $a$  の値を変化させながら、当該部分領域についての左右の画像をズレなく互いに重ね合わせる」ことによっても実現出来る。パラメータ  $a$  の値を変化させることは、右画像用平板を  $x$  軸上で平行移動させることである。

調整されたそのときのパラメータ  $a$  の値を  $a_1$  とし、その値を用いて右画像用平板を  $x$  軸上で平行移動する値  $t_1$  は、融像式を用いて、

$$\begin{aligned} t_1 &= \{a_1 / (1 / m_1 - 1 / m_0)^n\} \times (1 / m_1 - 1 / m_0)^{n+1} + b_0 \\ &= a_1 \times (1 / m_1 - 1 / m_0) + b_0 \end{aligned}$$

となる。この値  $t_1$  を用いて右画像用平板の位置を再配置する。パラメータ  $a$  の値  $a_1$  は、立体写真 A に関する固有のパラメータとなり、ステレオフォトビューアの内部に保存されるものとする。

パラメータ  $a$  についてさらに詳しく説明しよう。パラメータ  $a$  が小さい値の時には、画面の手前に融像するが、パラメータ  $a$  がある程度の大きさの値であれば、立体像の全体はディスプレイ画面後方に融像するので、見やすい適切な遠近的立体感が得られる。この状態はパラメータ  $a$  に依存し、極端に大き過ぎる値を設定すると、閲覧者によっては遠近的立体感がかなり遠方に感じられ、違和感のある立体像となる。その限界を超えると、両画像の間でズレが大きくなりすぎ、立体像は2重像となって融像しなくなる。パラメータ  $a$  の値を適切に定めることが重要であるが、事前の実験によれば  $a$  の適性値はかなりの幅があるので、その設定は容易である。

☆

☆

☆

実用的な融像式における個別決定法の数学的な背景は以上のような説明となるが、連携決定法は、試行錯誤的に行われるので、このようなすっきりとした説明を行うのは困難である。個別決定法における定数  $a$ 、 $b$  の決め方の具体例については、筆者の開発した SPV(ステレオフォトビューアに基づいて開発された立体写真閲覧ソフト)に基づいて、11. 2 節と 11. 3 節で述べる。

## 9. カメラ撮影で発生する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、ズーミングを行うと、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、ステレオペアの形成を妨げるような左右画像間のズレを補正する機構を取り入れる必要がある。

図 9 を用いて以下に、ステレオペアにおけるズレの補正方法を示す。なお、これらの補正は、9. 2 節の場合を除いて、立体メガネを着用しない非立体視状態で行う方が、作業はやり易い。ズレの補正は、左右画像が完全に重なり合うように目視で調整する。

### 9. 1 上下方向のズレの補正

図 9(1)に示すように、左右画像の間で上下に著しくズレている場合には、立体像は融像しないので、水平に揃える必要がある。このような上下方向のズレが生ずる原因としては、2 台のカメラを雲台にネジで固定する場合、どちらか 1 台の締め付けが緩いような場合に起こる。この補正方法は、図 9(1)に示した右画像を貼り付けた平板を  $y$  軸に沿ってプラス方向に  $p$  だけ平行移動させ、左画像と合わせることである。

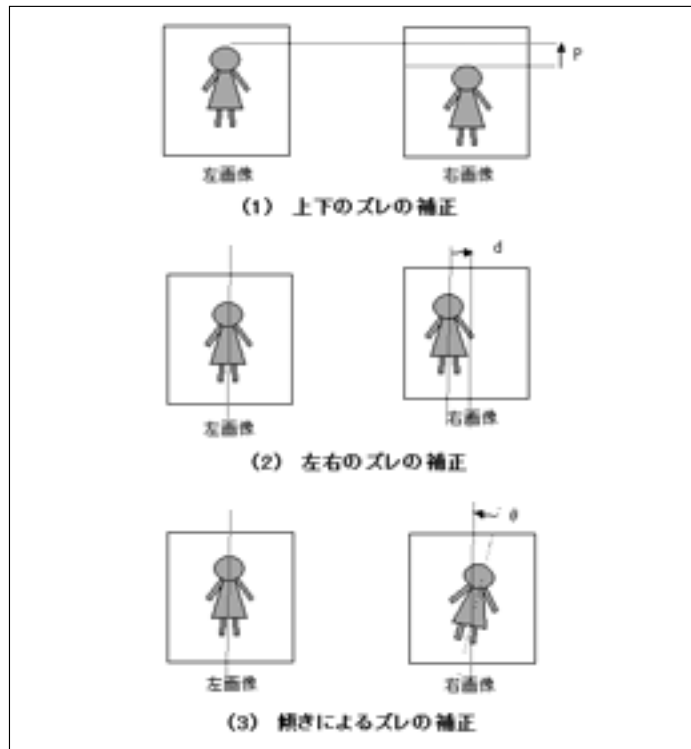


図9 ステレオペアにおけるズレの補正

## 9.2 左右方向のズレの補正

図9(2)に示すように、2つの画像が左右に離れ過ぎる場合である。これは2台のカメラを雲台に取り付ける際にレンズ間の距離を離し過ぎるときに起こり、しかも人物などの近くのを写すときにより起きやすい。このままステレオフォトビューアで閲覧すると、遠近的立体感はかなり近くに感じられ、立体像は飛び出して見える。補正方法としては、図9(2)に示した右画像の平板をx軸に沿って値dだけプラス方向に平行移動させることである。この補正方法は、すでに8.2節(1)項で述べた方法の中に含まれていて、融像式のパラメータbを調節することで対応できる。

## 9.3 傾きによるズレの補正

図9(3)に示すように、右画像あるいは左画像がその原点を中心に傾いて(回転して)しまうことがある。これは2台のカメラを雲台に取り付ける際に、その1台が極僅かではあるが雲台に傾いて設置された場合である。ネジでの締め付け方によって0.5度程度の傾きが生ずることがある。この補正方法は、図9(3)に示したように右画像を貼り付けた平板をその原点を中心に $\theta$ ラジアンだけプラス方向に回転変換させることである。

## 9.4 その他のズレについて

### (1) x軸あるいはy軸の傾きによるズレ

カメラ設置による画像の傾きについては、x軸あるいはy軸の傾き(回転)も考えられるが、この

傾きは通常の撮影作業においては微小角度に収まり、この補正は事前の実験によって他の補正には吸収されることが分かった。というのは、x 軸の傾きの補正は近似的に上下方向のズレの補正、また、y 軸の傾きの補正は近似的に左右方向のズレの補正とみなされ、結果としてそれぞれ上下方向と左右方向のズレの補正操作に反映されるからである。そこで、補正の複雑さを避けることから、ステレオフォトビューアには x 軸と y 軸の傾きによる直接的なズレの補正を取り入れていない。

## (2) z 軸に沿う方向のズレ

2 台のカメラを雲台に設置するとき、1 台のカメラがもう 1 台のカメラに比べて極々僅かではあるが、撮影対象物に近づいている、あるいは離れているというズレが生じるかもしれない。カメラ位置に関して、「1 台のカメラのフィルム面中央を原点とし、その面が xy 平面となる 3 次元座標系」を設定すると、このズレは z 軸方向に沿ったズレと行うことができる。もしこの補正を行うとしたら、右画像を適切な値だけ拡大あるいは縮小させることになる。しかし、雲台に 2 台のカメラを取り付ける際の z 軸方向のズレは一般に 1 ミリメートル以内に収まるので、カメラから撮影対象物までの距離(数十メートル)と比較すると、通常、その誤差は無視することが出来る。そこで、ステレオフォトビューアには、9. 4 節(1)項で述べた x 軸と y 軸の傾きの場合と同様、z 軸に関する直接的なズレの補正を取り入れていない。

## 10. 画像の再配置の方法

### 10. 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 2DCG 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 2DCG はその内部に  $3 \times 3$  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<sup>2)</sup>)。中心が座標系の原点に置かれた右画像用平板の 3 次元行ベクトルを  $[0 \ 0 \ 1]$ 、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  $[x \ y \ 1]$

表 2  $3 \times 3$  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	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & p & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$

とする。なお、そのベクトルの3番目の要素の1は同次座標表現における1である。9.1節、9.2節、9.3節に示したズレ補正の手続きを手順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2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10.3節で述べる。また、それらの間の変換順序は10.4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3次元行ベクトル $[x \ y \ 1]$ は、以下の式25によって座標変換(再配置)が行われるのである。ただし、基準となる左画像用平板に対しては、それは固定されているので、変換行列 $M_{t1}$ (要素 $p$ は零)、 $M_{t2}$ (要素 $t$ は零)、 $M_r$ (要素の三角関数の $\theta$ は零)は単位行列となり、結果として手順4のアスペクト比補正の変換のみが行われる。

$$[x \ y \ 1] = [0 \ 0 \ 1] M_{t1} M_{t2} M_r M_s \quad (\text{式 25})$$

なお、手順1に先立ち、閲覧用の座標変換、すなわちズーミングによる拡大縮小変換と平面内を移動することによる平行移動変換が行われるが、これについては説明を省略する。

## 10.2 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、6.2節(4)項で述べた実用的な融像式を以下に再掲し、それを用いて、「2DCGの持つ座標変換の機能は左右画像の左右のズレの補正を如何にして行うのか」を説明する。

$$t = \{a / (1 / m_1 - 1 / m_0)^n\} \times (1 / m - 1 / m_0)^{n+1} + b \quad (\text{式 17 の下段の再掲})$$

$$(a > 0, m_1 \leq m \leq m_0)$$

右画像用平板の $x$ 軸に対する最適な再配置の値 $t$ は、倍率 $m$ を用いて上記の融像式によって求められる。 $m_0$ は拡大限界倍率、 $m_1$ は縮小限界倍率である。(定数 $a$ と $b$ の設定法は第8章で述べた)。ところで値 $t$ は、表2に示すようにアフィン変換行列 $M_{t2}$ の要素となっている。閲覧者がズレ補正のためのズーミングを行うと、この値 $t$ は、倍率 $m$ 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 $M_{t2}$ が演算され、右画像用平板の3次元行ベクトル $[x \ y \ 1]$ が算出される。そして、この行ベクトルを用いて、2DCGの持つ座標変換機能は、右画像用平板を最適な位置へと再配置するのである。

## 10.3 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

```
//-----
// SPVにおける座標変換
//-----
void paint(double m){

    (ここに画像描画の前処理コード記載：省略)

    AffineTransform at=new AffineTransform(); //Java2Dアフィン変換の利用宣言
    graph=(Graphics2D)g;

    (ここにズーミングのコード記載：省略)

    double t=fusion(m,a,b);          //融像式によるH補正とD補正の算定
    t*=r;                             //A比補正の加味

    myTranslate(0,p,at);              //手続き1：V補正による平行移動
    myTranslate(t,0,at);              //手続き2：融像式の補正による平行移動
    myRotate(theta,w/2,h/2,at);       //手続き3：θ補正による回転
    myScale(r,s,w/2,h/2-p,at);        //手続き4：A比補正による拡大縮小
    graph.setTransform(at);           //Java2Dアフィン変換の設定

    (ここに画像の描画コード記載：省略)
}
```

図 10(1) SPV における座標変換の例

```
//-----
// 座標変換メソッド (Java2D-API)
//-----
void myScale(double sx,double sy,double x0,double y0,AffineTransform at){
    at.translate(x0,y0);
    at.scale(sx,sy);
    at.translate(-x0,-y0);
}
void myTranslate(double x,double y,AffineTransform at){
    at.translate(x,y);
}
void myRotate(double th,double x0,double y0,AffineTransform at){
    at.translate(x0,y0);
    at.rotate(th);
    at.translate(-x0,-y0);
}
//-----
// fusionメソッド (融像式)
//-----
double fusion(double m,double a,double b){
    double m0=15.0; //拡大限界倍率
    double m1=1.0;  //縮小限界倍率
    double n=0.5;
    double t=(a/Math.pow((1/m1-1/m0),n))*Math.pow((1/m-1/m0),n+1)+b;
    return t;
}
```

図 10(2) Java2D-API のメソッド例と融像式のメソッド



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 $1.333 \times 0.8$ 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、 $0.5 \times 0.8$ 、すなわち0.4ということになる。

換算アスペクト比は、横を $r$ 、縦を $s$ で表わすと、 $r/s$ であり、値 $r$ と $s$ はパソコンの対応解像度と立体ディスプレイの画面サイズから計算で求められる。画像用の左右平板の拡大縮小についての座標変換は、値 $r$ と $s$ がアフィン変換行列 $M_s$ の要素になる。

#### 10. 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。10.1節で述べた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。

#### 10. 5 座標変換の事例

上記に示したアフィン変換が実際の2DGGにおいては、どのように記述されるのかを、Java2D-APIを用いて開発されたSPV(ステレオフォトビューアに基づいて開発された立体写真閲覧ソフト)を例として説明しよう。そのプログラムの主要部分を図10に示す。他の2DCG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10(1)のプログラムは、左画像用平板を固定して、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)と左画像の平板の中央を中心とした回転(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。また、手順1の前の部分では、閲覧者のマウス操作によるズーム用の拡大縮小変換とシフト用の平行移動変換を行っている。図10(2)のプログラムは、Java2D-APIの座標変換メソッド(文献2)の使用例と実用的な融像式の例を示す。

このように2DCG(図10ではJava2D-API)の座標変換機能を使用すると、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、掲載したプログラムは、見やすいように整理して体裁を整えたもので、細部は一部省略してあり、完全なものではないこと、また、このJava2D-APIプログラムの事例では、原点は平板(画像)の左上を原点としているので、平行移動変換が多用されていることに注意されたい。

## 11. ステレオフォトビューア方式に基づく SPV の開発

### 11. 1 SPV の実現方法とそのプログラム構造

筆者はステレオフォトビューアを設計する当たり、2DGC の画像処理機能を利用することを着想し、その一般的な概念を第 10 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 の説明である。

SPV では 2DCG として、Java2D-API を採用した<sup>(文献2)</sup>。Java2D は Java プログラム言語が親言語となっている。画像処理および座標変換(アフィン変換)部分は、Java2D が担っている。Java2D-API を用いたソースコードは、Java コンパイラによって翻訳され、Java 仮想マシンによって実行される。Java プログラム部分は、左右の画像を生成し、その 2 つの画像のズーミング等における同期をとっている。従って SPV 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。図 11 に Java、Java2D-API 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 は Java を採用しているため、OS には依存せずに稼動する。

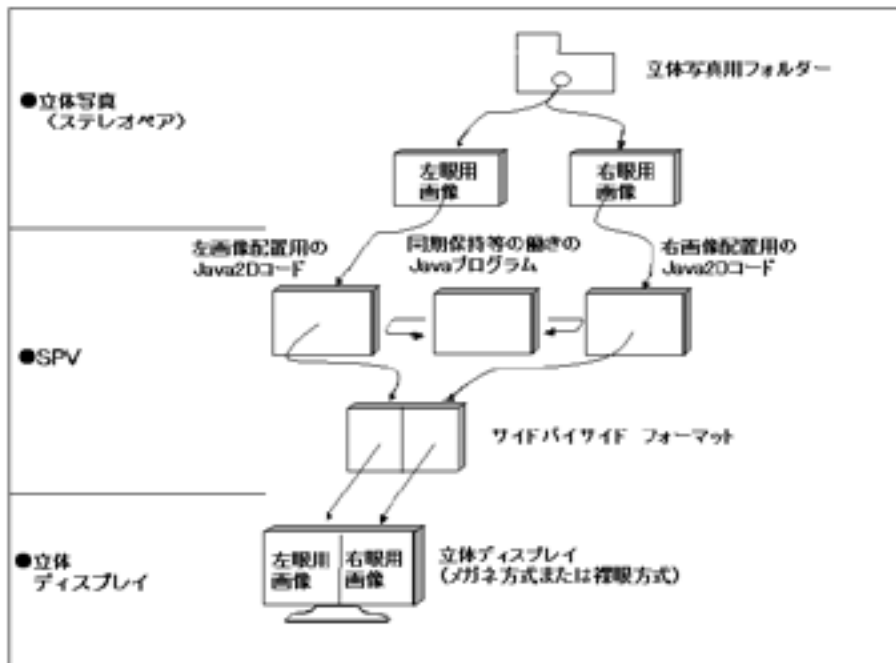


図 11 SPV におけるプログラムの連携構造

#### ● Java2D-API

座標変換を行い、左右画像を適切に再配置する。

- ・変換行列の設定
- ・テクスチャー機能の活用

#### ● Java 言語

Java2D の親言語として働く。

- ・ 左右の 2 画面の設定
- ・ ズーミングのためにスクロール機能の活用
- ・ ズーミングにおける左右画像の同期保持
- ・ 自動ズーミング機能のためにスレッド機能の活用
- ・ 立体写真の情報保存

## 11. 2 SPV の実際

SPV には、各種の操作ボタンが用意されている。その主要なボタンとその働きを Ver.1.3 に即して以下に示す。

- ズレ補正ボタン(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)
  - ・ H ボタン・・・左右のズレを補正する(融像式の定数  $b$  を決定する)。
  - ・ V ボタン・・・上下のズレを補正する。
  - ・  $\theta$  ボタン・・・傾きによるズレを補正する。
- 深度設定ボタン(最適な遠近的立体感を定めるボタン)
  - ・ D ボタン・・・遠近的立体感を定める(融像式の定数  $a$  を決定する)。
- アスペクト比設定ボタン(アスペクト比を設定するボタン)
  - ・ A ボタン・・・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。
- 2 重層化ボタン(左右画像のズレの検知するために左右の画像を重ね合わせるボタン)
  - ・ DBL-on ボタン・・・右画像の上に左画像を半透明で重ねる。
  - ・ DBL-off ボタン・・・上記機能を解除する。
- ズーム機能ボタン(立体写真をズーミングするボタン)
  - ・ ZM-on ボタン・・・立体写真の任意箇所をズーミングする。
  - ・ ZM-off ボタン・・・上記機能を解除する。
- ズーミング領域設定ボタン(ズーミング対象の領域(位置)を設定するボタン)
  - ・ TG-on ボタン・・・設定を行った後、ズーミング対象位置をマウスクリックによって特定する。
  - ・ TG-off ボタン・・・設定を解除する。
- 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)
  - ・ NEXT ボタン・・・次の立体写真を見る。
  - ・ BACK ボタン・・・前に表示した立体写真を見る。
  - ・ AUTO ボタン・・・フォトフレームのように自動表示を行い、複数枚の立体写真を繰り返し見る。(自動ズーミングも行う。)
- 記録ボタン(融像式のパラメータ  $a$ 、 $b$  などを記録するボタン)
  - ・ REC ボタン・・・内蔵するファイルに立体写真のデータを記録保存する。
- 倍率設定バー(画像の倍率を定めるスライドバー)
  - ・ S バー・・・画像を任意の倍率に設定する。

図 12 に実際に開発した SPV の画面を掲げ、図 13 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。

図 13 における最下段の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



図 12 SPV の画面

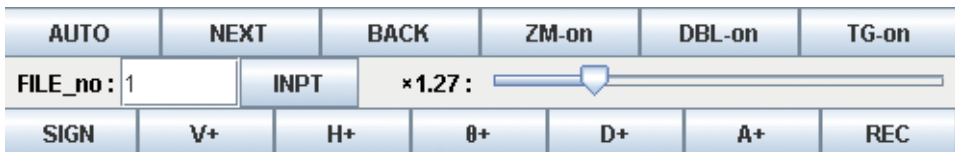


図 13 SPV の操作ボタン

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 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、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。例えば上下補正を行う V ボタン (SIGN ボタンで + と - に切り替わる) の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、V ボタン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. 3 SPV におけるズレ補正等の操作

SPV では、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ディスプレイ装置のために、左右の画像を重ね合わせる機能を持っている。この機能を用いると、立体メガネを装着しないで行うズレ補正を簡単に実行できる。このディスプレイ装置を用いる場合には、画面の左半分には左画像、右半分には右画像がそれぞれ、横に 1 / 2 倍で表示される。そこで、DBL-on ボタンをクリックして右画像の部分に基準の左画像を半透明化したイメージを重ね合わせる。左右画像は重なり合っているので、左右の重なりを見ながら、基準の左画像に合わせて右画像の位置を調整していく。重ねあわせ操作が終了したら、DBL-off ボタンをクリックする。

### (1) 上下のズレ補正

この補正は立体メガネを装着しない補正である。立体写真の撮影者(閲覧者者でもある)は、撮影した立体写真を SPV によって内部に保持されているフォルダーにファイル名を付けて格納する。閲覧機能の NEXT ボタンで写真を表示し、その写真の補正を次のように行う。最初は補正機能の V ボタン(+と-の2種類)のクリックによって、目視判定で左右画像が上下で重なり合ってズレがない状態にする。このズレを補正する  $p$  値は、REC ボタンをクリックすることによりビューア内部に記録保存される。

### (2) 傾きによるズレ補正

次に傾きによるズレ補正を行う。この補正は立体メガネを装着しない補正である。補正機能の  $\theta$  ボタン(+と-の2種類)のクリックによって、目視判定で左右画像が重なりあって、傾きによるズレがない状態にする。このズレを補正する  $\theta$  値は、REC ボタンをクリックすることによりビューア内部に記録保存される。なお、事前の視覚実験においては、立体写真の撮り方にもよるが、この補正を行わなければならない対象の立体写真はごく僅かであった。

### (3) 左右のズレ補正

次に左右のズレ補正を行う。この補正は最初、立体メガネを着用しないで補正を行い、次に融像する立体像を立体メガネを用いて確認する。8. 2 節(1)項で示したように、マウスによるズームイン操作で立体写真の遠景領域のある部分を解像度限界(拡大限界倍率  $m_0$ )まで拡大(ズームイン)する。そしてこの状態のもとで補正機能の H ボタン(+と-の2種類)をクリックして、遠景の拡大部分で左右画像が重なり合ってズレのない状態にする。この判定は目視で行う。そしてその状態で、立体メガネを装着して立体像が画面上の位置に生じるかどうかを確認する。左右のズレを補正するパラメータ  $b$  の値は、REC ボタンをクリックすることによりビューア内部に記録保存される。

### (4) 遠近的立体感の調整

最後に遠近的立体感の調整を行う。この補正は立体メガネを装着した立体視で行う。D ボタン(+と-の2種類)をクリックして、目視判定により立体像全体の遠近感が適切に得られるようにする。まず、8. 2 節(2)項で示した縮小限界倍率  $m_1$  までズームアウトした状態で D ボタンをクリックする。そして、立体像が適切な遠近感のもとで融像できれば、遠近を規定するパラメータ  $a$  の値を REC ボタンをクリックすることによりビューア内部に記録保存する。「遠近的立体感」の補正は、画像のズレ補正というよりも調整というものであるので、適切な(好みの)遠近感が得られれば、それでよい。

☆

☆

☆

上記(1)項、(2)項、(3)項についてのズレ補正は、それが正しく行われないと、完全な立体視を行うことが出来ない。3つのズレ補正と上記(4)項の調整にかかる時間は、慣れれば1枚の立体写真について1分程度である。SPV 内部には、立体写真の任意の枚数分について、上記の  $a$ 、 $b$ 、 $p$ 、 $\theta$  の値を記録保存するファイルが用意されているので、REC ボタンをクリックすることにより、このファイルに各立体写真の立体情報を保存することができる。以後閲覧する際に、このファイルが自動的に使用される。

## (5) アスペクト比の選択と閲覧の方法

閲覧の開始時に、立体ディスプレイ装置の画面サイズと立体写真のアスペクト比を適合させるために、A ボタン(+と-の2種類)をクリックすると、SPV は立体ディスプレイ画面等からアスペクト比を自動的に算出するので、閲覧者は適切なアスペクト比を選択する。実際の閲覧に際しては、AUTO ボタンをクリックすると、フォトフレーム(現在販売されている物の多くが非立体写真用)のように、SPV は複数枚の立体写真を自動的にズーミングしながら表示し続ける。また閲覧者が NECT ボタンあるいは BACK ボタンをクリックすると、特定の立体写真に対して任意のズーミング操作を行うことができる。

## 12. おわりに

### (1) 補足 (3D グラフィックスへの応用について)

本論文で述べた立体視におけるズーミング機構は、3D グラフィックスの3次元仮想空間の技術へ応用することが可能であり、それは簡単であることを以下に示す。3DCG においては、仮想空間内に視点を持った仮想人間(通常アバターと呼ばれる)を置き、そのアバターが仮想空間内を動き回って(ウォークスルーして)眺めた周りの光景をディスプレイ装置に表示する方式を採用している。アバターは2つの眼を持ち、その視点は、仮想空間内に置かれた平板に貼られた画像をある距離  $z$  から眺めているのである。式 17 の実用的な融像式の中で用いられている 2DCG 平面の属性(変量)である画面上の倍率  $m$  を、3DCG 空間の属性(変量)であるアバターの視点から画像までの距離  $z$  に置き換えればよい。この距離  $z$  とディスプレイ画面上に投影された画像の倍率  $m$  は反比例の関係

$$z \times m = k \quad (\text{式 26})$$

にあることから ( $k$  は  $z$  と  $m$  の反比例の定数)、 $z_0 \times m_0 = k$ 、 $z_1 \times m_1 = k$  なので、式 17 に倍率  $m$  を代入すると、

$$t = \{(a/k)/(z_1 - z_0)^n\} \times (z - z_0)^{n+1} + b \quad (\text{式 27})$$

が得られる。また、式 19 の倍率  $m$  の範囲、 $m_1 \leq m \leq m_0$  は、

$$z_0 \leq z \leq z_1 \quad (\text{式 28})$$

と距離  $z$  の範囲に変換できる。ちなみに  $z_0$  は「接近限界距離」、 $z_1$  は「後退限界距離」というべきものである。このように、仮想空間内のアバターの視点から画像を貼った平板までの距離  $z$  に関する  $t$  座標の座標変換式が得られる。式 27 は 3D グラフィックスにおける仮想空間をウォークスルー(画像への接近や画像からの後退)する際の実用的な融像式となり、定数  $k$  を 1 としたときのパラメータ  $a$ 、 $b$  は、第 8 章で述べた視覚実験で決定されるパラメータである。

なお、筆者は、3D グラフィックスである VRML 言語と JavaScript 言語、HTML 言語を用いた SPV と同様な立体写真閲覧ソフトを開発した。仮想空間用の融像式は式 27 を用い、性能的には SPV と全く同じものを作ることができた。その詳しい内容は筆者の文献<sup>(文献 3)</sup>を参照されたい。

## (2) 結論(SPVの有用性について)

本論文では、立体写真をズーミングして眺めることが出来る、2DCGの技術を用いたステレオフォトビューアの新しい方式を提案し、それを実現したSPVを紹介した。2DCGの画像処理技術を単に用いるだけでは、立体写真の左右画像(ステレオペア)における微小なズレがズーミング途中で拡大してしまい、結果的に立体視ができなくなる。これを解決するために、立体視の原理から融像式を導出し、その式のパラメータを立体像の目視による判定実験から決定する方法を考案した。この実用的な融像式により立体写真の解像度限界まで任意の領域をズーミングしても、微細な領域の立体像を2重像の発生を伴わないで綺麗に再生することが可能となったのである。(立体写真の撮り方によっては、遠景と近景が著しく離れて混在する写真となってしまう場合がある。そのような特殊な立体写真に対しては、SPVはすべての2重像の発生を完全に防ぐことが出来るという訳ではない。しかし、本論文で述べたような2重像の発生を融像式で抑制するSPVの開発は、立体視システムの世界では、従来行われなかったものと思われる)。

実際に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を用いて実験したところ、10倍程度までズーミングすることが出来た(画像の粗さを許せば、それ以上の倍率でズームインが可能であった)。使用した立体ディスプレイ装置は、Dimen社製G170P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。

また、融像式を用いない場合には、ズームアウトの状態からズームインを行うと、その進行に伴って本来ならば立体像が閲覧者の方向に拡大して迫ってくるはずであるが、実際には画像が拡大する状態で、立体像は徐々にその位置を後退させるように見えたりする。このような現象は3Dビデオ映像の映写においても起こるのである(文献4)。しかし、融像式を用いることにより、このような現象の発生を抑えることが可能になったのである。

SPVは2DCG技術を駆使しており、立体写真のズレの補正、すなわち左右画像の再配置は、2DCGの持つアフィン変換行列にその計算をさせることによって簡単に行われる。また、2DCGのアニメーション機能を用いて、立体写真の任意箇所に自動ズーミングを行うことができる機構を設けているので、SPVは、立体フォトフレームのソフトウェア・プロダクトとして利用することが可能である。

今回開発したSPVは、①パソコン用立体ディスプレイ装置(立体メガネ方式、裸眼方式とも可)、②3D立体テレビ(市販の家電品、立体メガネ方式、裸眼方式とも可)、③デュアル・プロジェクター方式(プロジェクターを2台使用)の立体投影システム、④ハーフミラー方式(液晶パネルを2台使用)の高品位立体ディスプレイシステムをその表示対象にすることができる。それらの立体表示装置が受け入れる立体フォーマットは、前者2つの装置については「サイドバイサイド・フォーマット」、後者2つの装置については「デュアルチャンネル・フォーマット(左右の2画像が圧縮されずに、それぞれ独立したチャンネルを介して配信に供される方式のフォーマット)」である。それ故、SPVは「サイドバイサイド・フォーマット」と「デュアルチャンネル・フォーマット」の2つの立体フォーマットの取り扱いを可能としている。

## 参考文献

- (1) 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)

- (2) 中山茂：『Java2 ―グラフィックスプログラミング入門―』、技報堂出版(1999)
- (3) 広内哲夫：「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」、情報研究(文教大学情報学部紀要)、pp.57～76、第46号(2012)
- (4) バーバード・メンディブル著、(株)Bスプラウト訳：『3D映像制作』、ボーンデジタル(2010)
- (5) 広内哲夫：「立体写真のズーミングを制御する融像式の提案」、3D映像、Vol.25、No.3、pp.17～30(2011)

## 付記

本論文の内容の一部は、速報の論文(文献5)としてすでに発表されている。速報の論文では、主に本論文の第2章、第4章～7章4節(1)項に記載された融像式に関する理論的な内容を簡明に論じている。本論文では、速報の論文で記述しなかった融像式の定数 $a$ 、 $b$ (パラメータ)の視覚実験による決定方法、ステレオペアの補正方法、および開発されたSPVにおける実際的な機能等を、1章、3章、7章4節(2)項～12章に追加して詳しく論じ、総合報告の形を採っている。