

僅か数十バイトの追加情報で実現する簡易 3 次元表示 —グラフィックエンジンを持たない組込機器でも良好な表示—

森川 治[†] 戸田 賢二[†]

[†]産業技術総合研究所 〒305-8562 つくば市梅園 1-1-1

E-mail: [†](morikawa.osamu, k-toda)@aist.go.jp

あらまし 従来の 3 次元 CG では、各オブジェクトは複雑なデータから構成され、描画には多量の計算を必要とする。そのため、実写映像に 3 次元オブジェクトを画像合成する場合でも、実写映像のモデル化を含め多大の準備が必要となる。本報告では画像合成を、背景となる実写映像へのオブジェクト映像の重畳表示に制約することで、非常に軽微な計算による簡易 3 次元表示を実現させた。重畳表示時に必要な伸縮率は、各映像に追加した数十バイト程度の情報から求める。グラフィックエンジンを持たない組込機器にも好適な表示を可能にした。

キーワード HyperStage, 実寸画素密度, 簡易 3 次元表示, 3D-CG

A Simple 3D System Realized by Additional Just Dozens of Bytes —Good Performance in the Embedded Machinery without Graphic Engine—

Osamu MORIKAWA[†] and Kenji TODA[†]

[†]National Inst. of Advanced Industrial Science and Technology 1-1-1, Umezono, Tsukuba, Ibaraki, 305-8562 Japan

E-mail: [†](morikawa.osamu, k-toda)@aist.go.jp

Abstract In usual 3D-CG, each object is made by complex data, and drawing of them needs many calculations. Synthesizing a 3D object and an actual image needs a large preparation such as modeling of the actual image. We will report, synthesizing is limited that the object image is overlaid on the background image, and a simple 3D system by a slight calculation is realized. The expansion and contraction rate of overlaying object image on background image is calculated by just dozens of bytes of information added to each image. This simple 3D system will perform well on embedded machine without graphical engine.

Keyword HyperStage, pix/m, simple 3D, 3D-CG

1. はじめに

一般に 3 次元 CG では、視点を変えて見たり、各オブジェクトを移動したり、加工したりできるなど多くの自由度があるが、描画には多量の計算を必要とする。そのため、快適な閲覧には、グラフィック表示のための専用チップや、高性能のマシンパワーを必要とする。

また、多くの機能を実現するため、各オブジェクトは複雑なデータから構成され、手軽にユーザが作って表示するというわけにはいかない。実写映像に 3 次元オブジェクトを画像合成する場合でも、実写映像のモデル化を含め多大の準備が必要となる。

一方、3 次元 CG の多くの機能が全ての利用者に必要わけではない。例えば、閲覧するだけであれば、加工する必要はないし、逆に閲覧者が勝手に加工したり変形させてしまつては困るような利用状況もありえる。

本報告では、単純化するため、視点の変更やオブジェクトの加工を意図的に省略した。背景となる実写映像の中へのオブジェクト映像の重畳表示に限定するこ

とで、非常に軽微な計算による簡易 3 次元表示を実現させた。重畳表示時に必要な拡大・縮小率は、各映像に追加した数十バイト程度の情報から求める。このため、グラフィックエンジンを持たない組込機器にも好適な表示を可能にした。

2. 3 次元表示

3 次元表示とは、被写体を立体的に知覚できる表示である。立体表示と言われるほとんどは、左右の眼に別々の映像を与えて立体感を出す両眼視差を利用している[1]。実写映像であれば、右眼用と左眼用の映像が必要となり、撮影も 2 台のカメラが必要となる。CG であれば、データは 1 つであっても、右眼用と左眼用 2 つの視点を与えることで 2 つの映像を得ることができる。

しかし 3 次元 CG のデータは複雑で、描画に多量の計算を必要とする。最近ではコンピュータの処理能力は向上したため、3 次元 CG の表示は個人用のパソコンでも可能になってきているが、自分の部屋や自分の映像を素材にデータ化することは、まだ不可能に近い。

また両眼視差による立体視には、視野制限、疑似感覚による生理的不快感などの問題があり、長時間の鑑賞には適していない[2]。

両眼視差ではなく、観察者の移動に同期して表示内容を変更させる運動視差を用いた立体表示もある[3][4]。手前に来れば大きく見え、遠くに移動すれば小さくなって見える。また我々は、手前の事物が奥の事物を隠すことから前後関係を理解できる。すなわち観察者の視点の移動ではなく、被写体の配置を変更することで、立体感を提示する方法もある。小川ら[5]の IBNR では、実写の写真を重畳表示することで立体感を出している。しかし、テーブルとテーブルに載った花瓶のように、動かす2枚の写真に連動性を持たせることができない。

3. 提案する表示方式

本報告では、IBNR と同様に実写の写真を重畳表示し、背景画像の中で各画像を自由に動かし、時に連動させて表示できる表示方式を提案する。

提案する表示方式は、背景写真の作る空間に別の写真の被写体があったと仮定すると、どのような大きさでその被写体が撮影されるかを計算で求め、その計算された大きさに被写体の写真を拡大・縮小して重畳表示するだけのものである。

そのため、3次元CGのように背景写真や重畳表示する被写体の写真の複雑なデータは必要ない。必要なのは、存在位置から表示すべき大きさを求めるためのわずかなデータだけである。これにより、実空間で物を見るときのように鑑賞できる。重畳表示した画像が背景画像内に実際にあるような錯覚が得られる。

以下、**実寸画素密度**という概念を定義し、3次元空間の点 (x,y,z) と写真のピクセルの関係について検討し、表示すべき大きさを計算する手法について述べる。

3.1. 実寸画素密度

写真は3次元空間から2次元平面への射影となっている。すなわち、3次元空間の点 (x,y,z) を与えると写真のピクセル (s,t) が一意に決まる。しかし同じピクセル (s,t) に対応する3次元空間の点は無数存在し、一意には決まらない。射影は3次元から2次元への写像であるので、情報欠落が生じる事は当然の帰結である。

しかし我々は、奥行き情報の欠落したはずの写真から、いろいろな手がかり情報を使って、写真内の被写体の空間位置、特に奥行き情報を読み取ることができる。手がかり情報のひとつに**大きさの恒常性**がある。

近づけば網膜に入る事物の像の大きさが大きくなり、遠ざかれば小さくなる。距離を変えると像の大きさも異なって見えるが、主観的には事物の大きさが変化したとは知覚しない。これが、大きさの恒常性と呼ばれる現象である。人間だけではなく、チンパンジー

にもみられる現象との報告もある[6]。すなわち、我々が知覚する「大きさ」という概念は、観察によって変化しない不変量のひとつであり、網膜像と密接に関係しているが、網膜像に1対1対応する概念ではない。さらに奥行きは、本来の大きさと網膜像の大きさから導かれる知覚といってもよい。

ここで、被写体1mの描画時のピクセル数(pixel/m)を表す「**実寸画素密度**」という概念を定義する。これは、**奥行き知覚を生み出す物理量**と解釈できる。

被写体は距離に反比例した大きさに見えるので実寸画素密度はカメラからの距離に反比例した値を持つ。3次元空間の点 (x,y,z) を与えると写真のピクセル (s,t) における実寸画素密度 n も一意に決まる。具体的には、カメラと原点 $(0,0,0)$ との距離を d 、原点の実寸画素密度を n_0 とすると、 $n = n_0 * d / (d+z)$ となる。本論文ではこれらをまとめて $(s,t;n)$ と記すことにする。 (s,t) から (x,y,z) は一意に決定できなかったが、 $(s,t;n)$ からなら (x,y,z) は一意に決定できることがわかる。

3.2. 実寸画素密度の比較

実寸画素密度は写真の各ピクセル毎に固有の値であり、モニタに表示した時点での拡大・縮小率によっても変化する。同一写真の2点の実寸画素密度を比べることで、奥行きの比較は可能だが、異なる写真の2点の実寸画素密度を比較することには意味がない。逆に、異なる写真があった場合、実寸画素密度を合わせることは意味がある。

実寸画素密度が同じであれば、表示サイズと被写体の実寸は比例する。従って、被写体の実寸での大きさ比較が表示サイズの比較で代用できる。背景となる写真の中へ別の写真を重畳表示した場合、それらの実寸画素密度が一致すれば、重畳表示した画像が背景画像内に実際にあるような表示となる。これにより簡易3D表示が可能になる。

3.3. 追加情報

背景写真に別の写真を重畳表示して簡易3D表示を実現するためには、背景となる被写体空間に3次元座標を定義し、3次元空間の点 (x,y,z) での実寸画素密度が計算できる情報が必要である。

なお本方式では、背景写真の全てのピクセルの実寸画素密度情報を記録する必要は無い。必要なのは3次元空間の点 (x,y,z) の写真内での実寸画素密度である。これは既に写っている被写体の実寸画素密度情報ではない。つまり本方式では、「点 (x,y,z) が与えられたとき、対応する写真内の $(s,t;n)$ がわかる事」は必要だが、「現在の写真内のピクセル (s,t) にある被写体の実寸画素密度」は知る必要がない。

3.3.1. カメラ位置と原点での実寸画素密度

写真に写っている3次元空間の点 (x,y,z) から写真の

ピクセル(s,t;n)を求めるためには、原点(0,0,0)の実寸画素密度 n0、カメラと原点(0,0,0)との距離 d（正確には XY 平面とカメラとの距離）の他に、カメラの位置の情報があればよい。

ここでは、簡単のために、原点(0,0,0)が写真の左下のピクセル(0,0;n0)に撮影されているとする。カメラの位置を(x0,y0,-d)とすると、

$$\begin{aligned} s &= x0 * n0 * z / (d+z) + x * n0 * d / (d+z) \\ t &= y0 * n0 * z / (d+z) + y * n0 * d / (d+z) \\ n &= n0 * d / (d+z) \end{aligned}$$

となる。

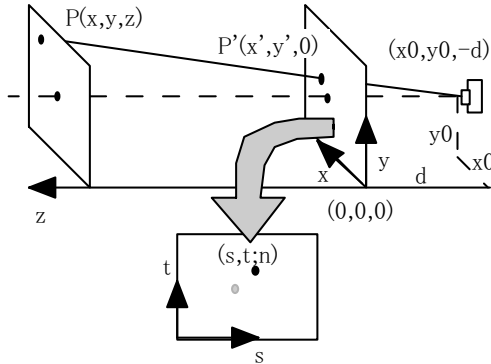


図1 3次元空間の点 P(x,y,z)と写真のピクセル P''(s,t;n)の関係

証明：

z=0のXY平面を考え、撮影を「3次元空間からXY平面への射影」と、「射影されたXY平面から写真への対応の組み合わせ」と捉える。

点 P(x,y,z)が点 P'(x',y',0)に射影されるとすると、

$$\begin{aligned} x' &= x0 + (x-x0) * d / (d+z) \\ &= x0 * z / (d+z) + x * d / (d+z) \\ y' &= y0 * z / (d+z) + y * d / (d+z) \end{aligned}$$

が成り立つ。

点(0,0,0)がピクセル(0,0;n0)で撮影されていることから、点 P'(x',y',0)はピクセル P''(n0 * x', n0 * y'; n0)で撮影される。したがって、写真のピクセル位置(s,t)は、

$$\begin{aligned} s &= n0 * x' \\ &= x0 * n0 * z / (d+z) + x * n0 * d / (d+z) \\ t &= y0 * n0 * z / (d+z) + y * n0 * d / (d+z) \end{aligned}$$

となる。また、被写体は距離に反比例した大きさに見えるので、点 P'では点 P の d/(d+z)倍に縮小される。したがって点 P の実寸画素密度 n は、

$$n = n0 * d / (d+z)$$

となる。

証明おわり。

3.3.2. 消失点

上の証明で、カメラ位置(x0,y0,-d)の情報があれば、3次元空間の点(x,y,z)から写真のピクセル(s,t;n)への変換が可能なが示された。しかしそのためにはカメラ位置を撮影時に計測する必要があり、実用的ではない。そこで、これに代わる、写真上のピクセル(x0*n0, y0*n0)に着目する。このピクセルは、実寸画素密度を無視すると、任意の z に対し空間の点(x0,y0,z)に対応することがわかる。そこで、実寸画素密度が0となるピクセル(x0*n0, y0*n0; 0)を考え、これを消失点(s_i,t_i;0)と書くことにする。なお消失点は、無限遠点(x0,y0,∞)に対応するピクセルであり、写真を見ながら利用者が指定することができる。x0,y0の代わりにs_i,t_iを用いて s,t を表すと、

$$\begin{aligned} s &= s_i * z / (d+z) + y * n0 * d / (d+z) \\ t &= t_i * z / (d+z) + y * n0 * d / (d+z) \end{aligned}$$

となる。

3.3.3. カメラまでの距離 d

カメラとの距離 d も、写真から読み取ることができる。原点での実寸画素密度の丁度半分になるピクセルを見つけ、その奥行き (Z 座標の値) を計測すればよい。たとえば部屋の写真において、テーブルの脚の部分が、原点の実寸画素密度の半分に撮影されていた場合、原点とテーブルの脚との奥行き之差が、求めるカメラとの距離 d になっている。これは実寸画素密度がカメラからの距離の Z 成分に反比例することから明らかである。さらに言えば、実寸画素密度が半分である必要はなく、XY 平面以外の点で実寸画素密度 n とその Z 座標の値 z がわかれば、距離 d は計算により $d = z * n / (n0 - n)$ と求められる。

3.3.4. 3次元原点

上の証明では、計算を簡単にするために被写体の写真左下の位置を原点とした左手系の座標系を使用した。しかし、原点は写真左下である必要はない。たとえば、テーブルの上面の1点を原点とした座標系とすることにより、テーブル上の事物の座標値は y=0 を満たすことになる。また、坂道の背景の場合、坂道の1点を原点とし、Z 軸を坂道の勾配に合わせることで、坂道の上の点はすべて y=0 を満たすような座標系にすることができる。

3.3.5. 描画基準点と、そこでの実寸画素密度

写真を重畳表示する場合、重畳表示する写真の左下を指定するのが常に都合がよいとは限らない。たとえば写真の被写体が、花瓶やコップのように床やテーブルに置くような事物であれば、地面に接する面を指定すると都合がよい。また、絵画や掛け時計など垂直面に設置する被写体写真の場合、被写体の中心位置

や設置する場合の留め金の位置などを使って重畳表示位置を指定したほうが使いやすいことがある。そこで、これら被写体写真を重畳表示するとき便利なピクセル位置を、**描画基準点**と呼ぶことにする。なお定義から明らかなように、この描画基準点には任意性がある。

なお実寸画素密度は、3次元原点と描画基準点が一致する場合には不要であるが、そうでない場合のために一般的には必要である。

3.4. 追加情報の写真への埋め込み

3.3での説明のとおり、追加すべき情報は、

- ・ 描画基準点とそこでの実寸画素密度
- ・ 3次元原点、実寸画素密度、カメラとの距離
- ・ 消失点

である。写真にこれらの情報を追加するだけで、被写体空間に3次元座標を定義でき、3次元空間の点(x,y,z)での実寸画素密度が計算できる。その結果、簡易3D表示を実現できる。

これらの値を写真に埋め込む方法はいろいろ考えられるが、我々は JPEG および PNG ファイルの Exif 規格のコメント文字列として追加した。描画基準点は **origin:**、3次元原点は **origin3d:**、カメラとの距離は **distance:**、消失点は **vp:**の後ろに数値をカンマで区切って記載する「キーワード: 数値,数値,・・」の形式とし、キーワードの順は問わないことにした。数値は、mm、pixel/m での計測値とした。例えば、図3に用いた男性の写真は” origin: 75, 31, 284¥n”、屋外の背景写真は ” origin:846,138,364¥norigin3d:846,138,364¥nvp: 1145, 540¥ndistance:2000¥n”である (¥n は改行コードの意味)。

重畳表示する場合、PNGの透明色を使用することも考えたが、今回は黒(RGB=(0,0,0))を透明色として扱うことにした。

4. ユーザインタフェース仕様

提案する表示方式は、背景写真の作る空間に別の写真の被写体があったと仮定すると得られるはずの映像を表示するものである。存在すると仮定する位置はユーザがマウス操作で指定する。このとき常識では存在しえないナンセンスな位置、例えば人物写真を空中や、水面の上に指定することもできるが、提案する表示方式では表示内容の意味には関与せずに、エラー処理をしない。人物の代わりに風船の写真であれば、空中に存在する事は自然であり、水鳥であれば水面であることは自然だからである。最終的に適切であるか不適切であるかは利用者、あるいは後述するように**自律性をもたせた写真**が判断するものとする。

具体的なユーザインタフェースを以下に述べる。

マウสดラッグにより写真を任意の位置に表示できる。ALT キーを押しながらドラッグすることで拡大・縮小ができる。

他の写真と重ならない位置でダブルクリックすることで、その写真が最前面に移動する。背景の写真と重なった位置でダブルクリックすることで、背景写真のその場所にふさわしい大きさに自動的に拡大・縮小させて表示することができる。このとき対象事物の写真は、背景写真の床に対象事物の写真の足が接地した位置に移動して表示される。本報告ではこの状態を、**対象事物の写真が背景画像にリンクしている**と表現する。また、対象事物の写真をリンクしている子、背景写真をリンクしている親と呼ぶ。

リンクした状態で対象事物の写真をダブルクリックすることでリンクを外すことができる。リンクした状態で対象事物をドラッグすると、対象事物の足が床面についた状態を保持したまま、あたかも対象事物が背景写真の中にあるように、移動先にふさわしい大きさに自動的に拡大・縮小されて表示される。後述するようにこれは、自律性をもった写真がマウスイベントから移動先を、y=0の平面内での移動に制約して決定するためである。対象事物が背景写真から完全に出ると、リンクが外れる。

なおリンクされている場合、対象事物の写真の黒(RGB=(0,0,0))は透明色として表示した。

5. 実装

将来的には、画像合成機能を持つディスプレイ[7]に素材となる複数の映像を接続して閲覧することを想定しているが、今回はパソコン上に閲覧ソフトを作成して実現した。

概念的には、パソコンソフトが対象事物の写真の描画位置と拡大率を計算して表示するのではなく、自律性を持たせた「背景写真」が計算をし、「対象事物の写真」がその計算結果に従い描画する。

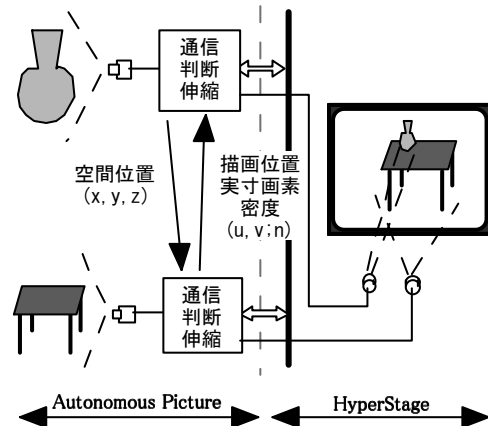


図2 自律型映像とハイパーステージ

5.1. ハイパーステージ

画像合成機能を持つディスプレイ（HyperStage と呼ぶ）に何枚かの写真（自律型映像と呼ぶ）を接続して閲覧することを考える。

HyperStage は、自律型映像同士の通信を媒介し、自律型映像から出される描画要求に従い、モニタ画面の指定位置に、指定した優先度で描画する装置とする。自律型映像は HyperStage を介してお互いに情報交換することにより、表示内容を自律的に決定し、HyperStage に対し描画要求を出す装置とする。マウスやキーボードも HyperStage に接続するので、自律型映像の一種としてみなす。

優先度に従って描画し最終的な表示画面を作成するのは HyperStage が行うので、自律型映像は描画内容、表示位置と優先度だけを決定すればよい。

5.2. 表示位置とサイズの決定

自律型映像は表示位置を、子としてリンクしていない場合、モニタ画面での 2 次元座標 (u,v) で自己管理する。子としてリンクしている場合は、親である自律型映像が作る 3 次元空間での座標 (x,y,z) で自己管理し、描画するときは、親の自律型映像に依頼して 3 次元座標 (x,y,z) に対応する 2 次元座標 $(u,v;n)$ を求めて、それを使用する。

親の自律型映像は、 (x,y,z) に対応する自分の写真でのピクセル位置 $(s,t;n1)$ を求め、そのピクセルのモニタ画面での表示位置 $(u,v;n)$ を求め、子の自律型映像に回答する。

描画内容は実寸画素密度 n に従ってオリジナルの画像を拡大・縮小して作成する。共通の親にリンクしている子の自律型映像同士の描画優先度は z 座標の値によって決定し、値が小さいほど高かった。これにより、後方の写真が前方の写真の背後に隠れて表示されるようになる。

一方 HyperStage は、自律型映像同士の対話を仲介する。また出された描画要求に対しては、今回は Z ソート描画法を採用し、いったん内部バッファに描画データを保存し、適当な時期に優先度の低い順に描画するスケジュールを組んで、モニタに表示した。

5.3. マウスイベントの処理

自律型映像であるマウスのドラッグイベントは、HyperStage を経由してカーソル位置にある自律型映像に対し与えられる。最前面の自律型映像が ALT キーの押し下げ状態を調べ、拡大・縮小、描画位置移動、リンクしている親の空間での移動など、状況にふさわしい処理内容を判断してそれを実行する。描画位置が変更する場合、リンクしている子があれば、全ての子が再描画の対象となる。しかしリンクしている子の描画位置 (u,v) は、親との対話により決定するので、子の自

律型映像は 3 次元座標 (x,y,z) だけを自己管理するだけで良い。

マウスのドラッグイベントから移動先の存在位置 (x,y,z) を求める処理は各自律型映像が行う。処理内容は各自律型映像毎に個性を出してかまわない。例えば、自律型映像が人物なら、移動範囲は同一平面内として決定する。また、自律型映像が絵画や掛け時計の映像の場合は、左右の壁であれば x を一定、正面の壁であれば z を一定に保った移動先に決定すればよい。



図3 2枚の背景写真の間を行き交う対象物（人物）と、テーブルの移動に連動する花瓶の様子（4シーン）

マウスのダブルクリックイベントも同様にカーソル位置にある全ての自律型映像に与えられる。イベント通知のあった自律型映像同士が HyperStage を介して情報交換を行い、背景画像に対象物画像がリンクしたりリンクを切断したりといった処理を実行する。リンクする時点で描画位置 (u,v) から親の自律型映像の 3 次元座標 (x,y,z) に切り替える際に、たとえば人物が背

景画像の地面に接地した位置(x,0,z)を初期値にとることができる(図3B)。

これら自律型映像同士の協調作業の結果、テーブルの上に花瓶をのせてテーブルを手前に移動させると、テーブルが大きく表示され、花瓶も一緒に移動して拡大表示されることになる(図3右半分のB→C→D)。また、背景写真が複数ある場合でも、対象事物の写真を背景写真の間を簡単に行き来させることができる。図3に移動の様子を示す。男性が背景写真を移動し(A→B)、後方にいる女性の近くまで移動し(B→C)、ALT+ドラッグで左写真を拡大している(D)。

6. 基本情報書き込みツール

ユーザが手持ちのデジカメ写真を簡単に扱えるように、基本情報書き込みツールを用意した。

描画基準点を決め、そこにもものさしとなる写真を拡大・縮小して実寸画素密度を指定する。同様に**3次元原点**とそこでの実寸画素密度を指定する(図4左)。さらに別の点とそこでの実寸画素密度を指定すると、**消失点**の候補が表示されるので、そこから消失点を指定する(図4右)。最後に、カメラからの**撮影距離**を数値で入力して終了する。



図4 部屋の写真に基本情報を書き込んでいる様子。ものさし写真(女性)を使って3次元原点の指定(左)、候補から消失点を選択(右)

7. おわりに

背景写真の作る空間に別の写真の被写体があったと仮定すると、どのような大きさでその被写体が撮影されるかを計算で求め、その計算された整合性のとれた大きさに被写体の写真を拡大・縮小して重畳表示する簡易3次元表示方式を提案した。

必要なのは、空間位置から表示すべき大きさを求めるためのわずかなデータだけである。これは被写体の存在する空間に関する情報であり、3次元CGのように被写体に関する情報ではない。そのため、被写体によらず、1枚の写真に1つのデータだけで十分であり、複雑なデータとなる事はない。また動画であれば、各フレーム毎にデータが必要になるが、カメラが動かない限り、被写体の空間は同一であり、省略可能となる。

表示は、自律型映像とハイパーステージの共同作業で実現する。表示内容の拡大・縮小は各自律型映像が同時並行して行い、表示順序はハイパーステージが画

面更新毎に行う。拡大率の計算は、自律型映像同士の対話によって決定する。そのため、自律型映像、ハイパーステージ共にグラフィックエンジンが無くても、快適な描画が実現する。

拡張性は自律型映像の個性で対応できる。例えば背景写真に、指定された位置が水面であるとか空中であるといった属性を返答する機能を与える。人物写真に、その属性を元に移動を判断する機能を与える。これにより「人物写真が水面や空中に浮かぶ」という不自然な構図を回避させることが可能となる。

基本情報を書き込むことで、デジカメで撮影した写真を簡単に3次元表示の素材として扱うことができる。観察時に、好きな画像を好きな位置に追加し、自由に表示位置を動かすことができる。これにより視聴者が主体的に観察することを可能にし、大きさの理解が促進される[8]。

素材となる自律型映像は、ネット経由での接続も可能である。これは新しい商品提示システムとして機能する。例えば、商品映像をネットから取り寄せることで、自宅に購入予定の商品を配置して、購入後の雰囲気確かめしてから購入を判断することが可能になる。

今回はパソコン上に閲覧ソフトを作成して実現したが、最終的には、画像合成機能を持つディスプレイであるハイパーステージを作成し、そこに素材となる自律性を持った複数の映像を接続して閲覧することを想定している。実用化に向けて本方式の研究・開発を進めたい。

文 献

- [1] 鶴飼、田中、杉山、田村、出澤、広瀬：“立体視および立体視装置”、ヒューマンインタフェース田村博(編)、pp.227-230、オーム社、1998
- [2] 奥山文雄：“立体視の影響”、ヒューマンインタフェース田村博(編)、pp.161-163、オーム社、1998
- [3] Johnny Chung Lee: Head Tracking for Desktop Virtual Reality Displays using the Wii Remote (YouTube), <http://jp.youtube.com/watch?v=Jd3-eiid-Uw>
- [4] 森川、片下、堀、戸田：複数入力対応のスマートディスプレイにおける映像フェデレーションの試み、日本バーチャルリアリティ学会第10回大会論文集、287-290(2005)
- [5] 小川、塚本、西尾：静止画を用いたWWW上でのコミュニケーション支援システム、情報処理学会論文誌、Vol.43, No.11, pp.3364-3374 (Nov. 2002)
- [6] 伊村、友永、八木：チンパンジーの大きさの恒常性錯視における線遠近法、影、運動情報の影響、日本動物心理学会第66回大会(2006)
- [7] 戸田、片下、堀、森川：複数のオーバレイフレームを自由に制御できるパーソナルメディアプロセッサマルチディスプレイやコンテンツセキュリティにも対応一、信学技報 CPSY2006-11, DC2006-11, pp.61-66(2006)
- [8] 森川、戸田：任意の画像を任意の位置に重畳表示することで、大きさ理解を促進する簡易3次元表示方式、情報処理学会第98回コンピュータと教育研究会論文集(予定)