

多視点画像を基にした 任意視点画像の補間生成手法

遠藤利生 後藤 誠 鳥生 隆

(株)富士通研究所

1つの物体を多視点から撮影した画像を基にして、まず物体の3次元構造を復元し、次に任意に指定された視点からの画像を3次元形状の投影によって生成する新しい手法を述べる。一般に structure from motion の原理によって3次元構造を復元する方法は雑音の影響を受けやすいので、復元された3次元形状を再投影する方法は不安定になりがちである。例えば、実際に撮影した視点に再投影して得られる画像は、その視点から撮影した原画像と必ずしも一致しない。本稿では、復元された物体の3次元構造に誤差が含まれていても、実際に撮影した視点からの生成画像がそこでの原画像と一致するような画像生成手法(3次元誤差相殺方式)を提案する。

A Method of Generating the Image at an Arbitrary View Point from Images at Multiple View Points

Toshio Endoh, Makoto Goto and Takashi Toriu

Fujitsu Laboratories Ltd.

We propose a new method of generating the image at an arbitrary view point from images at multiple view points. This method, first, recovers the 3-D structure of an object in the images, then, generates the image at the indicated view point by projecting the 3-D shape of the object. Since recovering 3-D structure from motion is generally sensitive to image noise, generated images by projecting the 3-D shape are often inappropriate. For example, the generated image at an input view point may not coincide with the input image there. We found a method to cancel the 3-D structure error. In this method, the generated image at an input view point should coincide with the input image there.

1 はじめに

コンピュータのディスプレイに映し出された物体をマウスなどを使って自由に動かすことができれば、電子博物館や電子ショッピングが利用者にとってより親しみやすいものになる。このような機能を実現するために、筆者らは、物体を複数個の視点から撮影した画像を基に、それらの画像を補間合成して利用者が任意に指定した視点からの画像をインタラクティブに生成するシステム”MultiView”を開発した。本稿では開発したシステムのアルゴリズムについて述べる。

任意に指定した視点からの画像を生成する手法として、3次元計測によって物体の3次元形状モデルを作成し、それに実写画像から得られるテクスチャをマッピングしたものを任意の視点に投影する方法が提案されている [1]。この方法は蓄積すべきデータ量が少ないという利点があるが、レーザ光に基づくレンジファインダなどの高価な距離計測装置が必要となる。

一方、数多くの視点から撮影した画像を蓄えておき、指定された視点に最も近い視点から撮影した画像を選択して表示する方法が知られている。この方法は実写画像をそのまま表示するので、画像の品質は良好であるが、ディスプレイ上で滑らかに物体を動かすためには予め非常に多数の画像を蓄えることが必要となり、膨大な記憶容量を要する。

データ量を削減するためには、比較的少数個の視点から撮影した画像を基に、視点の間を滑らかに補間して任意の視点からの画像を生成する手法が望まれる。2つの視点から撮影した2枚の画像から任意視点の画像を生成する手法として、透視投影の仮定の基で被写体を覆う四辺形毎の射影変換を利用する方法 [2, 3]、直交投影を仮定すると2枚の画像の特徴点の座標の線型結合によって任意視点における特徴点の座標が定まることを利用する方法 [4] が提案されている。また、エビポーラブレイン画像（平行移動するカメラで撮影した時系列画像を等間隔に並べた3次元画像）において物体上の各点の軌跡が直線であることを利用して、平行移動する複数個の視点から撮影した画像の間を補間する方法 [5] が提案されている。これらの方法はいずれも陽には3次元復元を行わずに、任意視点の画像と実際に撮影した複数視点の画像の間を直接関係付けている。処理の目標が2次元画像を生成することであれば、必ずしも3次元構造を復元する必要がないからである。また、少なくとも実際に撮影した視点では実写画像がそのまま再生されるために良好な画質が得られる

という利点がある。しかし、再生画像を通常のコンピュータグラフィックスによる映像と融合するような場面を考えると、陽に3次元構造を持つておくことの利点も捨てがたい。

一般に、structure from motion の原理によって複数個の視点から撮影した実写画像を基に3次元構造を復元する方法 [6, 7, 8] は雑音の影響を受けやすく、復元した3次元形状を再投影して任意視点の画像を生成する方法は不安定になりがちである。そのため、実際に撮影した視点に再投影して得られる画像はその視点から撮影した原画像に必ずしも一致しない。本稿では、多視点画像を基に物体の3次元構造を復元し、その情報を利用して任意視点における画像を生成する手法で、たとえ3次元構造復元に誤差が含まれていても実際に撮影した視点に投影した場合には原画像に一致するような画像生成手法（3次元誤差相殺方式）を提案する。多視点画像からの3次元構造復元は、一般に、画像間で特徴点を対応付けるステップと特徴点の対応関係から3次元構造を復元するステップの2段階で行われる。画像間の特徴点の対応付けを自動的に行う場合には誤対応が避けがたいので、本稿では、画像間の特徴点の対応付けと3次元復元の間に矛盾が生じないように検証しつつ、これら両者を同時に行う手法についても提案する。

2 アルゴリズム

2.1 概要

1つの物体を多視点から撮影した複数の画像を基に、任意に指定した視点からの画像を生成する基本的な方法として、撮影された入力画像を適当な重みで足し合わせる方法が考えられる。このとき入力画像間の対応をどのように取るかが問題となる。ここでは、物体の一定箇所が画像上に投影されている点（画像特徴点）が入力画像間であらかじめ対応付けされていると仮定する。この仮定の下で、画像を画像特徴点を頂点とする3角形の集合に分割し、各3角形毎に独立にかつ一樣な重みで出力画像を生成する方式を採用する。インタラクティブに利用者に映像を提示することを考慮すれば画像生成を高速化することが望まれる。このような簡単な方式で出力画像を生成することで処理が高速化されることが期待される。このとき次の3点が問題となる。

- (1) 出力画像上の3角形の配置 出力画像上のどこに画像特徴点を置くか
- (2) 入力画像と出力画像の対応 出力画像のある画素を生成する際に参照する入力画像の画素はどれか
- (3) 入力画像を足し合わせる重み 複数の入力画像

の対応する画素の値をどんな割合で足し合わせるか

(1) 出力画像上の3角形の配置

簡単には、指定された視点に近い視点から撮影したいくつかの入力画像の3角形の配置を視点間の距離に応じた重みで平均する方法が考えられる。この方法は、入力画像間で物体の投影像が大きく変化する場合には用いることができないと予想される。ここでは、撮影される物体が剛体であると仮定し、与えられた画像特徴点の対応から各特徴点の3次元位置（物体特徴点）を求め、それを指定された視点に再投影することで3角形の配置を定めるものとする。この方法は、物体特徴点が正確に計算される場合には、正確な位置の3角形を与える。しかし、実際には物体特徴点の計算には誤差が避けられないので、後述するような対策が必要になる。

(2) 入力画像と出力画像の対応

3角形の頂点の対応から定まるアフィン変換で対応付けるものとする。詳細は、後述する。

(3) 入力画像を足し合わせる重み

あらかじめ入力画像の視点位置を3次元的に配置し、その凸包を計算しておく。凸包とは、与えられた点を含む最小の凸多角形のことである。視点が指定されるとその位置と原点を半直線で結び、凸包との交点を求める。その交点が位置する凸包上の3角形の頂点に対応する3視点のみを用い、残りの視点から撮影した入力画像は用いないものとする（すなわち重み0）。3視点の重みは、頂点と交点の位置関係から定める。詳細は後述する。

従って、多視点画像に基づく任意視点からの画像生成は、物体の3次元構造復元と3次元構造に基づく補間生成の2段階に分解される。我々はこの考えに基づき、1つの物体を多視点から撮影した複数の画像を基に、任意に指定した視点からの画像を生成するシステム MultiView を開発した。その全体構成を図1に示す。撮影された多視点画像から物体の3次元構造を復元する MultiView Organizer と物体の3次元構造を基に指定された視点からの画像を補間生成する MultiView Player から構成される。



図1 MultiView 全体構成

2.2 MultiView Organizer の構成

本節では、多視点画像から物体の3次元構造を復元する MultiView Organizer の構成について述べる。入力画像間で画像特徴点の対応が付いていると仮定したので、ステレオ視の原理から各特徴点の3次元位置（物体特徴点）を容易に求めることができる。ここでは、文献 [8] の方法を用いて、画像特徴点の真の位置からの位置ずれが正規分布に従う雑音として観測されるという統計モデルの下での最尤推定量を計算し、物体特徴点の算出とともに視点位置の補正も行った。

3次元構造を復元する際には誤差が避けられないので、物体特徴点を入力視点の1つに投影して得られる画像特徴点は元の画像特徴点とは一般には一致しない。そのため、このままでは指定された視点が入力視点のどれかと一致する場合でも入力画像と異なる画像が生成される。撮影された入力画像は十分高品質であると考えられるので、これは望ましくない性質である。

3次元構造復元誤差の影響は一般には避けられないが、映像生成を主目的とすることによって次のような解決法（3次元誤差相殺方式）が得られる。図2にその考え方を示す。

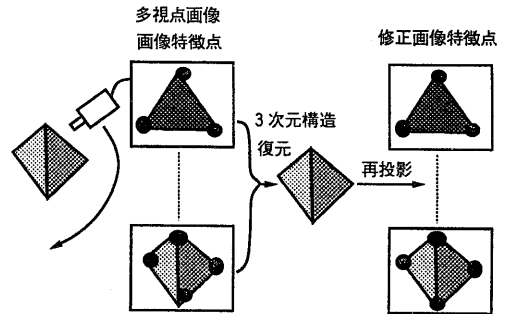


図2 3次元誤差相殺方式

1. 特徴点の再投影 物体特徴点を各視点に再投影して、それを画像特徴点として用いる（修正画像特徴点）
2. 入力画像の扱い 入力画像はそのまま用いる

画像特徴点を修正することにより、3次元構造復元誤差の有無によらず指定された視点が入力視点のどれか1つと一致する場合には、その入力画像が出力されるようになる。視点の移動に対して生成される画像は連続的に変化するので、任意の視点において生成画像が高品質になることが期待される。

なお、物体特徴点間の接続関係も復元する必要はあるが、ここでは物体は円筒でよく近似できると仮定して、物体特徴点を円筒に写像しその凸包の接続関係をそのまま用いることとした。これによって、物体の3次元構造は完全に定められるので、コンピュータグラフィックスによる映像と融合するような用途にも利用できるようになる。

2.3 MultiView Player の構成

本節では、復元された3次元構造から画像を生成する MultiView Player の構成について述べる。基本的には前述したように、指定された視点に物体特徴点を投影して画像特徴点を定め、それらで構成される3三角形毎に独立に、近傍3視点の入力画像を重み付きで足し合わせた画像を出力画像とする(図3)。

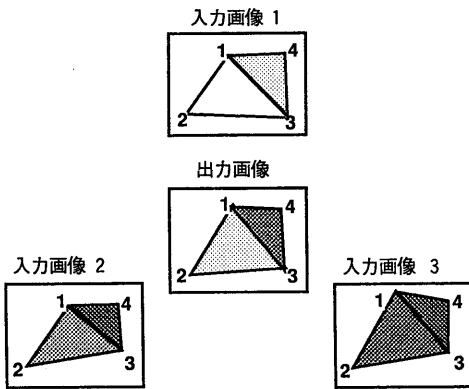


図3 出力画像の生成

入力画像と出力画像の対応は、3三角形の頂点の対応から定まるアフィン変換で対応付ける。つまり、出力画像上の点 (x, y) と入力画像上の点 (x', y') が

$$\begin{aligned} x' &= a_x x + b_x y + c_x \\ y' &= a_y x + b_y y + c_y \end{aligned} \quad (1)$$

という関係にあるとして、3頂点の対応で定まる1次方程式を解いて6つの係数 $a_x, b_x, c_x, a_y, b_y, c_y$ を求める。

出力画像の (x_o, y_o) における画素の値 $I_o(x_o, y_o)$ は、

$$I_o(x_o, y_o) = \alpha_1 I_1(x_1, y_1) + \alpha_2 I_2(x_2, y_2) + \alpha_3 I_3(x_3, y_3) \quad (2)$$

で定める。ここで、 $(x_i, y_i) (i = 1, 2, 3)$ は入力画像 i へのアフィン変換で (x_o, y_o) を変換した点である。重み α_i は次式で定められる。

$$\begin{aligned} \alpha_1 : \alpha_2 : \alpha_3 &= S_1^4 : S_2^4 : S_3^4 \\ \alpha_1 + \alpha_2 + \alpha_3 &= 1 \end{aligned} \quad (3)$$

ただし、 S_i は3視点位置を頂点とする3三角形と指定された視点と原点を結んだ半直線がその3三角形を通過する交点 p を用いて表される図4の領域の面積である。

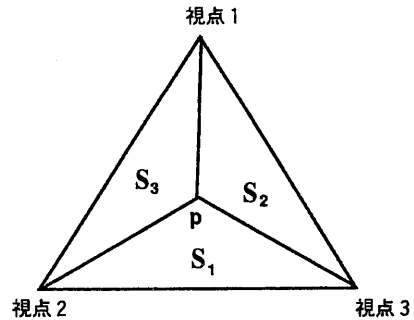


図4 重みの決定

4乗することで、3次元構造復元誤差により発生する可能性がある入力画像間の位置ずれの影響を低減している。

なお、各入力画像をあらかじめ標準3三角形にアフィン変換しておくことにより、出力の3三角形から標準の3三角形へのアフィン変換を1度だけ計算すればよいように処理を高速化できることを注意しておく。

3 MultiView Organizer の自動化

前章では、画像特徴点はあらかじめ対応付けられていると仮定した。特徴点の対応付けは煩雑な処理であり、人手で行うことは困難である。本章では、画像特徴点の対応付けを自動化する一手法について述べる。特徴点を自動で対応付ける場合には誤対応が避けられないので、対応付けと一緒に3次元構造復元を行うことによって対応付けの妥当性を検証する。

ただしここでは、入力視点が球面の赤道に等間隔で配置されている場合のみを扱う。入力視点の配置が一般の場合は今後の課題である。このため、画像生成時には近傍2視点の入力画像を用いるものとし、その重みは入力視点が赤道上に並んでいる場合には経度差の4乗の比で定める。

3.1 問題の定式化

ある固定された軸の回りを等速度で回転している物体を固定したカメラから撮影した時系列画像を基に、物体表面上の特徴点の3次元座標とカメラの姿勢を復元する問題を考える。

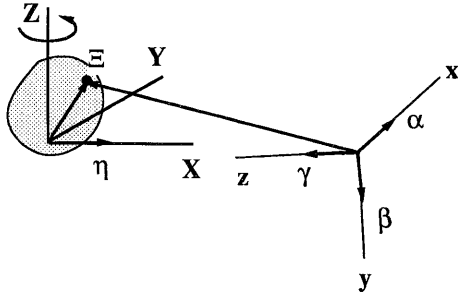


図5 座標系の設定

(1) 座標系

図5に示すように、世界座標系 $(O - XYZ)$ として、物体の回転軸を Z 軸、カメラのレンズ中心から回転軸に降ろした垂線の足を原点、原点とレンズ中心を通る直線を X 軸、原点を通り X 軸と Z 軸に直交する直線を Y 軸とするような座標系を用いる。物体は Z 軸の正の方向に向かって右回りの回転をしているものとし、原点から見てレンズ中心の方向を X 軸の正の方向とする。また、 Y 軸の正の方向はこの座標系が右手系となるように選ぶ。

また、カメラに固定された座標系 $(o - xyz)$ として、レンズ中心を原点、原点を通り投影面(画像平面)と平行な平面を xy 平面、原点を通りこの xy 平面に垂直な直線を z 軸とするような座標系を採る。なお、 xy 平面に対して物体が存在する方を z 軸の正の向きとする。 x 軸と y 軸の正の向きはこの座標系が右手系になるように選ぶ。このようにして定義されるカメラ座標系の x 軸方向、 y 軸方向、 z 軸方向の単位ベクトルをそれぞれ α, β, γ とする。これらの互いに直交する3つの単位ベクトルがカメラの姿勢を表す。また、世界座標系における X 軸方向の単位ベクトルを η 、世界座標系の原点とカメラのレンズ中心の距離を d とすると、レンズ中心の位置ベクトルは $U = d\eta$ で表される。

(2) 基本方程式

物体表面上に K 個の特徴点(以下、物体特徴点と呼ぶ)があるものとし、 k 番目の物体特徴点の時刻 t における位置ベクトルを $\Xi_k(t)$ とする。このとき、この物体特徴点は画像平面において、

$$\begin{aligned} x_k(t) &= f(\Xi_k(t) - U) \cdot \alpha / (\Xi_k(t) - U) \cdot \gamma \\ y_k(t) &= f(\Xi_k(t) - U) \cdot \beta / (\Xi_k(t) - U) \cdot \gamma \end{aligned} \quad (4)$$

に投影される。ここで、 f は焦点距離であり、 $x \cdot y$ は2つのベクトル x, y の内積を表す。物体が Z 軸の回りを回転する角速度を ω とすると、時刻 t にお

ける k 番目の特徴点の座標 $\Xi_k(t)$ は時刻 $t = 0$ における座標 $\Xi_k = \Xi_k(0)$ を使って、

$$\Xi_k(t) = R(\omega t) \Xi_k \quad (5)$$

と表される。ただし、

$$R(\omega t) = \begin{bmatrix} \cos(\omega t) & -\sin(\omega t) & 0 \\ \sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

である。

(4) 問題設定

解くべき問題は次のように表現できる。等しい時間間隔 τ をおいて N 個の時刻 $t_n = n\tau (n = 0, 1, \dots, N-1)$ で撮影された N 枚から成る時系列画像を基に、時刻 $t_0 = 0$ における K 個の物体特徴点の座標 Ξ_k とカメラの姿勢 α, β, γ を推定せよ。ただし、物体の回転軸とカメラレンズの中心との距離 d 、物体の回転速度 ω 、および、焦点距離 f は既知とする。

3.2 処理の流れ

3次元構造復元は、特徴点候補抽出、特徴点の対応付けと3次元座標推定、最尤推定による最適化の3段階で行う。

特徴点候補抽出では、 N 枚の画像のそれぞれにおいて、複数個の特徴点(以下、画像特徴点と呼ぶ)の位置を検出する。この処理によって、 N 枚の画像のそれぞれに対して画像特徴点の座標のリストが得られる。

特徴点の対応付けと3次元座標推定では、 N 枚の画像のそれぞれにおいて抽出された画像特徴点の座標のリストを基に、異なる画像の間で画像特徴点同士を対応付けながら、それらの画像特徴点に対応する物体特徴点の3次元座標を推定する。この処理は仮説と検証の繰り返しによって実行する。具体的には、カメラ姿勢の初期値を設定した後、画像特徴点の対応付けの仮説を次々に生成しながら、その仮説の基で対応付けられた画像特徴点に対応する物体特徴点の3次元座標を算出する。仮説が設定される度に、算出された3次元座標を N 枚の画像上に投影し、投影された位置の近傍に画像特徴点が存在する率があらかじめ設定したしきい値より高いかどうかに基づいてこの仮説を検証する。この仮説・検証の繰り返しはある終了条件を満たすまで繰り返す。

最尤推定による最適化では、カメラ姿勢を表すパラメータと物体特徴点の3次元座標を最尤推定によって最適化する。

なお、以下の議論では、時刻 t_n における回転行列 $R(\omega t_n)$ を R_n と略記する。

3.3 特徴点候補抽出

N 枚の画像のそれぞれに対して、まず、ガウシアンラプラシアン処理を行う。次に、 $L \times L$ の大きさのウィンドウを走査し、ウィンドウ中心の画素値の絶対値が予め定めたしきい値以上であり、かつ、その値がそのウィンドウ内で最大であるときにその画素は画像特徴点であると判断する。 n 枚目の画像から抽出された l 番目の特徴点の座標を (x_{nl}, y_{nl}) と表す。

3.4 特徴点の対応付けと3次元座標推定

まず、カメラ姿勢の初期値 $\alpha_0, \beta_0, \gamma_0$ を設定する。次に、以下の仮説・検証の処理を繰り返して、物体特徴点の3次元座標を推定する。

ステップ1): 物体特徴点の番号 k を1に初期化する。

ステップ2): N 個の時刻からランダムに2つの時刻を選び、それを t_n と $t_{n'}$ とする。また、これら2つの時刻に撮影したそれぞれの画像の画像特徴点のリストからランダムに1つずつ画像特徴点を選び、それぞれ、 (x_{nl}, y_{nl}) 、 $(x_{n'l'}, y_{n'l'})$ とする。

ステップ3): (x_{nl}, y_{nl}) 、 $(x_{n'l'}, y_{n'l'})$ の2点が3次元空間中を回転している同一の物体特徴点が時刻 t_n 、および時刻 $t_{n'}$ に投影されたものであるとの仮説を置き、その物体特徴点の時刻 $t = 0$ における3次元座標 Ξ_k を以下に述べる最小二乗法で推定する。推定残差 $J_{min}^{(1)}$ を算出し、それが予め設定しておいたしきい値 $J_{th}^{(1)}$ より大きいときはこの仮説を棄却してステップ2)に戻る。

●物体特徴点の3次元座標の推定と推定残差の算出

前記の仮説の基で、式(4)の分母を払ったものと式(5)を利用すると、

$$\begin{aligned} x_{nl}(R_n \Xi_k - U) \cdot \gamma_0 &= f(R_n \Xi_k - U) \cdot \alpha_0 \\ y_{nl}(R_n \Xi_k - U) \cdot \gamma_0 &= f(R_n \Xi_k - U) \cdot \beta_0 \end{aligned} \quad (7)$$

および、

$$\begin{aligned} x_{n'l'}(R_{n'} \Xi_k - U) \cdot \gamma_0 &= f(R_{n'} \Xi_k - U) \cdot \alpha_0 \\ y_{n'l'}(R_{n'} \Xi_k - U) \cdot \gamma_0 &= f(R_{n'} \Xi_k - U) \cdot \beta_0 \end{aligned} \quad (8)$$

が得られる。ここで、評価関数 $J^{(1)}(\Xi_k)$ を次式で

定義する。

$$\begin{aligned} J^{(1)}(\Xi_k) &= \\ & \{x_{nl}(R_n \Xi_k - U) \cdot \gamma_0 - f(R_n \Xi_k - U) \cdot \alpha_0\}^2 \\ & + \{y_{nl}(R_n \Xi_k - U) \cdot \gamma_0 - f(R_n \Xi_k - U) \cdot \beta_0\}^2 \\ & + \{x_{n'l'}(R_{n'} \Xi_k - U) \cdot \gamma_0 - f(R_{n'} \Xi_k - U) \cdot \alpha_0\}^2 \\ & + \{y_{n'l'}(R_{n'} \Xi_k - U) \cdot \gamma_0 - f(R_{n'} \Xi_k - U) \cdot \beta_0\}^2 \end{aligned} \quad (9)$$

この評価関数を最小にするような Ξ_k を k 番目の物体特徴点の3次元座標の候補とする。このとき、 $J^{(1)}(\Xi_k)$ の最小値が推定残差 $J_{min}^{(1)}$ である。

ステップ4): 推定された物体特徴点の3次元座標 Ξ_k を式(5)に従って回転させながら式(4)に従って全ての時刻 t_n における画像平面に投影する。次に、ステップ2)で得られた各時刻における画像特徴点のリストの中から投影された点との距離が最小であるような画像特徴点を選択する。このようにして選択された N 個の画像特徴点の中から、さらに、対応する時刻において投影された点との距離が予め定めたしきい値 $dist_{th}$ よりも小さいものを選択し、それらを $(x_{n(i)}, y_{n(i)})$ ($i = 1, 2, 3, \dots, I_n$) とする。ここで、 I_n は選択された画像特徴点の数である。もし、 I_n が予め定めたしきい値 num_{th1} よりも小さければ先の仮説を棄却してステップ2)に戻る。

ステップ5): 評価関数 $J^{(2)}(\Xi_k, \alpha, \beta, \gamma)$ を次式で定義し、この評価関数が最小になるように物体特徴点の3次元座標 Ξ_k 、および、カメラの姿勢 α, β, γ を推定する。

$$\begin{aligned} J^{(2)}(\Xi_k, \alpha, \beta, \gamma) &= \\ & = \sum_{i=1}^{I_n} \left\{ \left[x_{n(i)} - \frac{f(R_{n(i)} \Xi_k - U) \cdot \alpha}{(R_{n(i)} \Xi_k - U) \cdot \gamma} \right]^2 \right. \\ & \left. + \left[y_{n(i)} - \frac{f(R_{n(i)} \Xi_k - U) \cdot \beta}{(R_{n(i)} \Xi_k - U) \cdot \gamma} \right]^2 \right\} \end{aligned} \quad (10)$$

式(10)の最小化は非線型最小化問題になるが、ステップ3)で求めた Ξ_k と $\alpha_0, \beta_0, \gamma_0$ を初期値とするNewton法によって計算できる。

ステップ6): 推定された物体特徴点の3次元座標 Ξ_k とカメラ姿勢 α, β, γ を用いてステップ4)と同様に各時刻における画像平面に物体特徴点の3次元座標を投影し、投影された点との距離が $dist_{th}$ 未満であるような画像特徴点の組 $(x_{n(i)}, y_{n(i)})$ ($i = 1, 2, 3, \dots, I_n$) と画像特徴点の数 I_n を再び算出する。もし、 I_n が予め定めたしきい値 num_{th2} よりも小さければ先の仮説を棄却してステップ2)に戻る。

ステップ7): Ξ_k を k 番目の物体特徴点の座標として登録する。

ステップ8):ある終了条件を満たす場合には終了し、そうで無い場合は物体特徴点の番号 k に1をインクリメントしてステップ2)に戻る。

3.5 最尤推定による最適化

最尤推定の評価関数 $J^{(MLE)}(\mathbf{E}_k, \alpha, \beta, \gamma)$ を次式で定義し、この評価関数が最小になるように物体特徴点の3次元座標 \mathbf{E}_k 、および、カメラの姿勢 α, β, γ を推定する。

$$J^{(MLE)}(\mathbf{E}_k, \alpha, \beta, \gamma) = \sum_{n=0}^{N-1} \sum_{i=1}^{I_n} \left[\left\{ x_{n(i)} - \frac{f(R_{n(i)}\mathbf{E}_k - U) \cdot \alpha}{(R_{n(i)}\mathbf{E}_k - U) \cdot \gamma} \right\}^2 + \left\{ y_{n(i)} - \frac{f(R_{n(i)}\mathbf{E}_k - U) \cdot \beta}{(R_{n(i)}\mathbf{E}_k - U) \cdot \gamma} \right\}^2 \right] \quad (11)$$

式(11)の最小化は前項で求めた \mathbf{E}_k と $\alpha_0, \beta_0, \gamma_0$ を初期値とし、 \mathbf{E}_k と $\alpha_0, \beta_0, \gamma_0$ を交互にNewton法によって最適化する手順を繰り返すことで実行できる。

4 実験システム

開発した画像生成方式の有効性を検証するため、実験システムを構築し利用者がマウスを使用してインタラクティブに指定した視点からの画像を生成する実験を行った。使用した機種はSun SPARCstation 20 SXで、C言語とMotifを用いて開発した。

図6に実験システムを起動したときの画面を示す。マウスの位置に応じた視点からの画像を生成することができる。また、右側のスライダを動かすことによって表示される物体の大きさと距離を変更することができる。

図7は、壺を半球上でほぼ均等に分布する19視点から撮影して手で画像特徴点を指定した多視点画像から、指定された視点からの画像を生成した例である。画像の大きさは320×240である。上側1つと左右2つが補間に用いられた入力画像、中央が生成された画像である。良好な画像が生成されていることが分かる。

図8は、フクロウの置物を赤道上90視点から撮影し特徴点の対応付けを自動的に行い、そのうち19視点からの画像だけを用いて指定された視点からの画像を生成した例である。画像の大きさは320×240である。左右2つが補間に用いられた入力画像、中央が生成された画像である。やはり良好な画像が生成されていることが分かる。生成に必要な時間は人手、自動の場合ともほぼ1枚あたり0.3秒程度であった。

5 むすび

与えられた多視点画像から、任意に指定された視点からの画像を生成する新しい手法について述べた。本手法は、多視点画像から物体の3次元構造を復元し、それを指定された視点に再投影する方式であるが、3次元構造復元誤差の有無によらず入力視点からの生成画像がそこでの入力画像と一致する性質を持つ。利用者がマウスを使用してインタラクティブに指定した視点からの画像を生成する実験システムを構築し、本方式の有効性を検証した。

本稿ではまた、新しい3次元構造復元方式として、画像間の特徴点の対応付けと3次元復元の間に矛盾が生じないように検証しつつ、これら両者を同時に行う手法についても述べた。

参考文献

- [1] 末永康仁, 渡部保日兄: 3D形状と輝度(色)の同時計測が可能なスキャナとその顔画像入力への応用, 情処研報, CV 67-5, 1990.
- [2] 鳥生隆, 遠藤利生, 為清彰: 射影変換に基づく画像間の補間, 信学技報, IE94-14, pp.29-36, 1994.
- [3] 遠藤利生, 鳥生隆, 内田好昭: 画像補間技術, FUJITSU, Vol.46, No.3, 1995.
- [4] 向川康博, 中村裕一, 大田友一: 2枚の顔写真を用いた任意方向の顔画像の生成, 情報処理学会論文誌, Vol.37, No.4, pp.635-644, 1996.
- [5] 片山昭宏, 田中宏一良, 押野隆弘, 田村秀行: 多視点画像の補間・再構成による視点追従型立体画像表示法, 信学論, Vol.J79-D-II, No.5, pp.803-811, 1996.
- [6] Spetsakis, M.E. and Aloimonos, J.Y.: Multiframe Approach to Visual Motion Perception, *Intern. J. Comput. Vision*, Vol.6, No.3, pp.245-255, 1991.
- [7] Tomasi, C. and Kanade, T.: The Factorization Method for the Recovery of Shape and Motion from Image Streams, *Proc. of Image Understanding Workshop*, pp.459-472, 1992.
- [8] Toriu, T. and Endoh, T.: Maximum Likelihood Estimator for Structure and Motion from Multiple Perspective Views, *Proc. of 2nd Asian Conf. Comput. Vision*, Vol.2, pp.707-711, 1995.

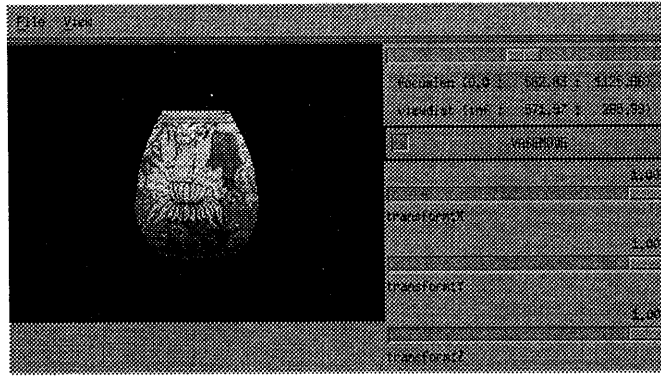


図6 実験システム画面



図7 人手で画像特徴点を指定した例

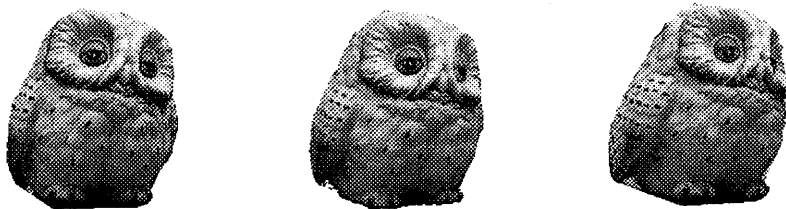


図8 画像特徴点抽出を自動化した例