

全周形状復元のための不変記述子による 3次元部分形状マッチング

山田 健人^{†1} 金澤 靖^{†1}

本論文では、部分的に復元された物体形状から全周形状を復元するための、各部分形状間のマッチングの方法を提案する。提案法では、多面体で復元された部分形状の頂点の法線方向から見た部分画像を生成し、その部分画像内で複数の keypoint を検出し、それらの SIFT 特徴量により記述子を生成する。このような記述子を用いることで、対応している部分形状間にスケールの差があっても対応可能となる。そして RANSAC を用いて部分形状間の対応する頂点を見つけることにより、部分形状間の位置合わせを行う。提案法の有効性を実験により示す。

3-D Partial Shape Registration using Invariant Descriptor

KENTO YAMADA ^{†1} and YASUSHI KANAZAWA ^{†1}

We propose a registration method for reconstructing an entire shape from its partially reconstructed shapes. For finding matches between partial shapes, we adopt a descriptor which is independent from viewpoints. The proposed descriptor consists of one or more SIFT descriptors obtained from a frontal image to a vertex of a partial shape. Using this descriptor we can compute the similarity transformation between two partial shapes. We show the effectiveness of our method by experiments with simulated data and real images.

^{†1} 豊橋技術科学大学知識情報工学系

Department of Knowledge-based information Engineering, Toyohashi University of Technology

1. はじめに

カメラ画像を用いた全周形状の復元は、物体形状を非接触でコンピュータに取り込むことができるため、実際の人間の頭部モデルや、文化遺産のデジタルアーカイブ化¹⁾、観光地のウォークスルー²⁾、複合現実や拡張現実など、さまざまな分野に応用されている。このような全周形状の復元には、大別すると、カメラ以外のセンサ、例えばレンジファインダなどを用いるものと、カメラ画像のみを用いるものに分けられる。本研究では、誰もが利用可能なカメラ画像のみからの全周形状の復元を考える。

カメラ画像のみからの物体の全周形状の復元においては、物体を撮影した画像列の特徴点群とそれらを撮影したカメラ位置を全てまとめて最適化するバンドル調整²⁾⁻⁴⁾を行うことが主流となっている。バンドル調整は、画像列から3次元形状とカメラ位置を同時に推定する方法であり、原理としては画像列の特徴点群の2次元座標と、復元した3次元形状を復元したカメラで投影した際の差(再投影誤差)を最小化することに相当する⁴⁾。バンドル調整を用いた全周形状の復元の例として、Snavely らの Photo Tourism²⁾ が挙げられる。Photo Tourism は Flickr^{*1}などの画像投稿サイトにおいて、観光地などの多数の人が同じ対象を撮影している画像を用いて、各カメラ位置の復元と形状の復元を行っている。極めて多くの画像が利用できるため、その復元精度は高いが、処理に時間が掛かるという問題がある。

一方、部分的に復元された形状を合成することで全体形状を得るというアプローチにおいては、その位置合わせにおいて ICP (Iterative Closest Point) アルゴリズム^{5),6)} を用いることが多い。この ICP アルゴリズムは、部分的な3次元形状をより大きな3次元形状(モデル形状)に位置合わせする手法であり、位置合わせの初期値が必要ではあるが、その収束が保証されているだけでなく、形状間の対応が未知でも良い。ICP とは異なるアプローチとして、Wu らの VIP (View-point Invariant Patches)⁷⁾ がある。これは、カメラだけでなく、レンジファインダのような直接形状を測定するセンサーを必要とするが、視点に不変な記述子を用いて、重ね合わせを行っている。VIP では、取得した3次元形状から平面を検出し、まず、その平面をその法線方向から見た画像を生成する。そして、その画像内の SIFT 特徴⁸⁾ を検出し、その SIFT 記述子と平面の法線を用いて、部分形状間の位置合わせを行う。したがって、VIP は ICP とは異なり、位置合わせの初期値が不要であるが、カメラ以外のセンサーが必要であるだけでなく、また3次元形状内の平面を用いるため、曲面で構成

*1 <http://www.flickr.com/>

された物体や表面の凹凸が大きい物体には適用が難しい。

本研究では、カメラ画像のみを用いて、その画像列から復元した部分形状の位置合わせを行うことを目的とし、その位置合わせにロバストな不変記述子とそれを用いた対応付け法を提案する。ここでは、復元した部分形状は多面体であるとし、その頂点の法線方向から見た頂点周囲の画像を考える。同じ頂点に対する異なる復元形状であれば、それらの画像はほぼ回転とスケールのみで変換できるはずである。そこで、その画像に対し複数の SIFT 特徴を検出し、それらを用いて頂点の記述子とする。そして投票法を用いて、頂点間の対応を決定する。更に、得られた対応を初期対応とし、RANSAC により合成パラメータを求めることで、ロバストな合成が可能となる。シミュレーションデータおよび実画像による実験により、提案法の有効性を示す。

以下、第 2 節では部分形状の復元方法について、第 3 節では提案不変記述子について、第 4 節ではその記述子に適した対応付け法について、第 5 節では部分形状の合成のためのパラメータ推定法について、第 6 節ではシミュレーションデータおよび実画像による実験結果を示す。最後に、第 7 節でまとめを行う。

2. 部分形状の復元

部分形状の復元には、通常の複数画像からの形状復元法を用いる。ここでは山田らの方法⁹⁾を用いる。以下、山田らの方法について概説する。

山田らの方法は、2 画像のみから高精度に 3 次元復元を行うことを目的としており、カメラに関し、以下の仮定を置いている。

- カメラの光軸点は既知とし、それを画像内の原点とする座標系を用いる。
- 画像に幾何学的な歪みはないか、あるいは既に補正済みであり、画素配列は直交し、アスペクト比は 1 とする。
- 2 台のカメラの焦点距離は等しい、あるいは 1 台のカメラをズームやフォーカスを替えずに撮影する。

そして、高精度な復元形状を得るために、以下の工夫を行っている。

- 基礎行列計算において、拡張 FNS 法を組み込んだ厳密な最尤推定計算法¹⁰⁾を用いている。
- 基礎行列における初期値計算に Taubin 法を用いることで、最適化の収束を速くできる¹¹⁾。
- 基礎行列からのカメラパラメータ (焦点距離) の分解において、高橋ら¹²⁾の方法を用い

ることで、注視条件¹³⁾にも対応している。

- 得られたカメラの運動パラメータから投影行列を計算し、最適な三角測量の方法¹⁴⁾に基づき、3 次元位置を復元する。

この山田らの手法では、入力と出力はそれぞれ次の通りとなっている*1。

入力: 2 画像間の対応点の座標の組 $(x_i, y_i), (x'_i, y'_i), i = 1, \dots, N (> 8)$ 。

出力: 3 次元復元された位置 $(X_i, Y_i, Z_i), i = 1, \dots, N$

一般にカメラ画像のみからの復元では、その復元の基準となる座標およびそのスケールが未知であるため、得られた形状 $((X_i, Y_i, Z_i), i = 1, \dots, N)$ もそれらを決定することは出来ない。そこで、一般には、第 1 画像を撮影したカメラ位置を座標の原点とし、カメラ移動ベクトルの大きさを 1 とするようなスケールを用いて復元することが多い。

以下、得られた部分形状の頂点 $P_i(X_i, Y_i, Z_i), i = 1, \dots, N$ を次のベクトルで表す。

$$\mathbf{x}_i = (X_i, Y_i, Z_i)^\top \quad (1)$$

3. 部分形状マッチングのための不変記述子

前節の方法などで復元した部分形状を合成することを考える。一般にこのような合成には部分形状間で共通の頂点を用いれば良い。ICP^{5),6)}では、このような共通の頂点を予め求める必要はないが、収束するまで繰り返し計算が必要となる。

画像から形状を復元した場合、そのような共通の頂点は必ず複数の画像で特徴点として観測されている。本研究では、それを利用して位置合わせをすることを考える。このとき、その特徴点を含む局所画像をそのまま用いることは難しい。なぜなら、全周形状を復元するような画像を撮影する場合、多くはワイドベースラインでの撮影となり、見た目の画像が大きく変化してしまうためである。そこで本研究では、このような視点に依存しない局所画像を生成し、それを用いて部分形状間の対応を決定する。同様のアプローチとして Wu らの VIP⁷⁾があるが、これはレンジファインダなどで得られた部分形状における平面を求め、その平面の法線方向から見た画像を生成し、その画像内の SIFT 特徴量⁸⁾と平面の法線を用いて位置合わせを行なっている。したがって、平面を含まないような物体に対しては適用できない。

*1 プログラムは以下に公開されている。

<http://www.img.tutkie.tut.ac.jp/programs.html>

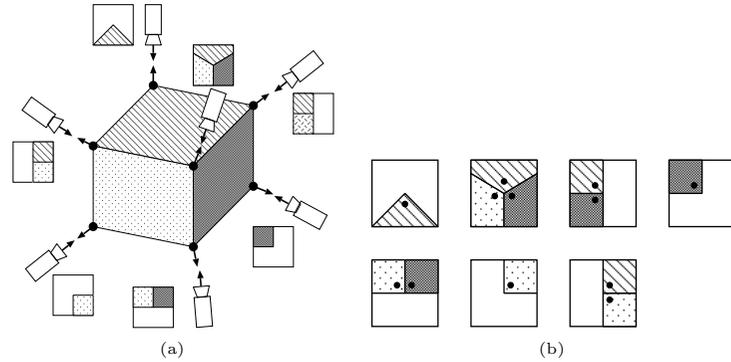


図1 提案不変記述子の概要: (a) 頂点の法線方向. (b) 頂点の正面画像における複数の SIFT 特徴量.

本研究では、復元した部分形状の各頂点に対する法線を求め、その法線方向から見た部分画像を生成して用いる (図 1(a)). しかし、前節の方法で復元した形状は、撮影したカメラ位置を基準とした座標系で復元されること、またその形状に対しスケールの不定性があることから、その生成した部分画像間にも回転やスケールの不定性を含む。そこで、図 1(b) に示すように、部分画像において複数の keypoint とそれらに対する SIFT 特徴量を求め、それらを用いて頂点の記述子を定義する。そして、この記述子を使って共通頂点を見つけることを考える。

以下、それらについて説明する。

3.1 正面部分画像の生成

復元された形状は、ドロネー三角形分割などを用いて多面体で復元されているものとする。このとき、頂点 P に対する法線 \mathbf{n}_P は、その頂点を含む面の法線 \mathbf{n}_k^P , $k = 1, \dots, N_P$ から次のように計算できる。

$$\mathbf{n}_P = N \left[\sum_{k=1}^{N_P} \mathbf{n}_k^P \right] \quad (2)$$

ここで $N[\mathbf{a}]$ は、ベクトル \mathbf{a} の単位ベクトル化とする。もし、頂点が部分形状の縁である場合、その平面の法線を頂点の法線とする。

得られた法線 \mathbf{n}_{P_i} により、以下の手順で頂点 P_i の部分正面画像を生成する。

- (1) 頂点 P_i の法線ベクトル \mathbf{n}_{P_i} を用い、仮想カメラの各軸 \mathbf{x}_{c_i} , \mathbf{y}_{c_i} , \mathbf{z}_{c_i} を次のように

設定する。

$$\begin{aligned} \mathbf{z}_{c_i} &= -\mathbf{n}_{P_i}, \\ \mathbf{y}_{c_i} &= N[\mathbf{z}_{c_i} \times (1, 0, 0)^\top], \\ \mathbf{x}_{c_i} &= N[\mathbf{y}_{c_i} \times \mathbf{z}_{c_i}] \end{aligned} \quad (3)$$

ここで、 $\mathbf{a} \times \mathbf{b}$ はベクトル \mathbf{a} と \mathbf{b} の外積を表す。

- (2) 復元された形状を仮想カメラ座標系に変換する回転行列 \mathbf{R}_{c_i} を次式で求める。

$$\mathbf{R}_{c_i} = (\mathbf{x}_{c_i} \ \mathbf{y}_{c_i} \ \mathbf{z}_{c_i}) \quad (4)$$

- (3) 復元された座標系と仮想カメラ座標系の並進ベクトル \mathbf{t}_{c_i} は、頂点 P_i の位置 \mathbf{x}_i とその法線ベクトル \mathbf{n}_{P_i} を用いて次のように計算できる。

$$\mathbf{t}_{c_i} = \mathbf{x}_i + d\mathbf{n}_{P_i} \quad (5)$$

ここで、 d は任意の実数であり、頂点 P_i と仮想カメラの距離を表す。

- (4) 復元座標系から仮想カメラ座標系への変換 $\{\mathbf{R}_{c_i}, \mathbf{t}_{c_i}\}$ を用い、次のように部分形状の各頂点 P_j の位置 \mathbf{x}_j , $j = 1, \dots, N$ の投影 (x^j, y^j) を求める。

$$s \begin{pmatrix} x^j \\ y^j \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{R}_{c_i}^\top (\mathbf{x}_j - \mathbf{t}_{c_i}) \quad (6)$$

ここで、 s はベクトルの第 3 要素を 1 にするスカラーである。

- (5) 頂点の投影位置 (x^j, y^j) , $j = 1, \dots, N$ を用いて、原画像からテクスチャを貼りつける^{*1}。
- (6) テクスチャが貼り付けられた画像に対し、画像中心 (対象となる頂点 P_i の投影位置) を中心として適当なサイズの正方形に切り出す。

3.2 不変記述子

次に、前節で得られた各頂点の各正面部分画像に対し、keypoint 検出および SIFT 特徴量⁸⁾ の計算を行う。

部分形状 S の頂点 P_i の正面画像 \mathcal{I}_{P_i} に対し、 N_{P_i} 個の keypoint とその特徴量ベクトル $\mathbf{s}_{i\alpha}$, $\alpha = 1, \dots, N_i$ が得られたとき、頂点 P_i の特徴を記述する記述子 \mathcal{D}_i を次のように定義する。

*1 厳密にはテクスチャの貼り付けにも投影が必要であるが、ここでは通常のテクスチャマッピングを用いる。



図 2 部分画像内で検出された keypoint の例

$$\mathcal{D}_i = \{s_{i1}, s_{i2}, \dots, s_{iN_i}\} \quad (7)$$

この記述子は、頂点 P_i によって要素となる SIFT 特徴量の数 N_i が変化するが、このように、一つの頂点に複数の SIFT 特徴量ベクトルを持たせることにより、切出された正面画像のサイズの影響が少なくなり、これらを用いた位置合わせもよりロバストになると期待できる (図 2)。

4. 対応付け

前述の不変記述子を用いて、部分形状間の位置合わせのための頂点の対応付けを行う。各頂点に対する記述子にはそれぞれ複数の SIFT 特徴量が含まれ、その数は一律ではない。そこで、ここでは図 3 に示すような投票法を用いて、その対応を決定する。

対象とする部分形状を \mathcal{S} , \mathcal{S}' とし、それぞれの頂点を $P_i, i = 1, \dots, N, P'_j, j = 1, \dots, M$ とする。また頂点 P_i, P'_j に対する記述子をそれぞれ

$$\begin{aligned} \mathcal{D}_i &= \{s_{i1}, s_{i2}, \dots, s_{iN_i}\}, \\ \mathcal{D}'_j &= \{s'_{j1}, s'_{j2}, \dots, s'_{jM_j}\} \end{aligned}$$

とする。投票法による対応付けの手順は以下の通りとなる。

- (1) 形状 \mathcal{S} のある各頂点 P_i (記述子 \mathcal{D}_i) に対し、以下の処理を行う。
 - (a) SIFT 特徴量ベクトル $s_{i\alpha} \in \mathcal{D}_i, \alpha = 1, \dots, N_i$ に対し、以下の処理を行う。
 - (i) 次の式を最小にするもう一つの形状 \mathcal{S}' に含まれる $s'_{j\beta} \in \mathcal{D}'_j$ を見つける (図 3(a))。

$$E = \|s_{i\alpha} - s'_{j\beta}\| \quad (8)$$
 - (ii) 得られた $s'_{j\beta} \in \mathcal{D}'_j$ を持つ頂点 P'_j に投票する。
 - (b) 以上により、形状 \mathcal{S} の頂点 P_i に対し、最も得票数の多い頂点 P'_k を求める (図 3(b))。もし、同じ得票数の頂点が複数ある場合、式 (8) の最も小さいものを選ぶ。
- (2) 形状 \mathcal{S}' の頂点 P'_j の SIFT 特徴量ベクトル $s'_{j\beta}$ に対して同様のことを行い、最も得票数の多い P_l を求める (図 3(c))。

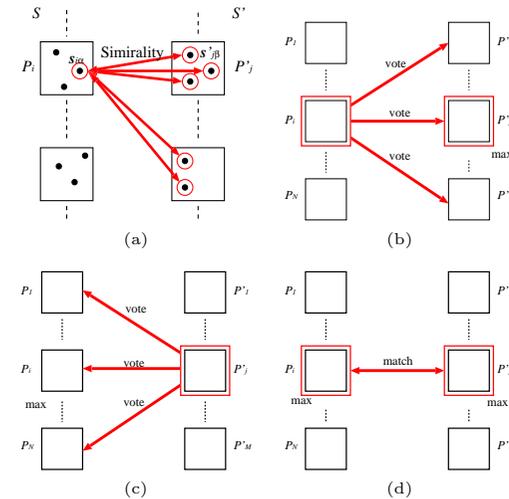


図 3 投票法による対応の決定: (a) $s_{i\alpha}$ と最も類似度の高い $s'_{j\beta}$ の探索. (b) P_i からの投票. (c) P'_i からの逆の投票. (d) 対応の候補の決定.

- (3) もし、 $P'_k = P'_j$ かつ $P_l = P_i$ であるとき、頂点对 (P_i, P'_j) を対応とする (図 3(d))。以上の処理を \mathcal{S} の全ての P_i および \mathcal{S}' の全ての P'_j に対して行い、対応を決定する。

5. 部分形状の合成

第 2 節で説明した通り、カメラ画像のみから復元された形状は、一般にスケールの不定性があるだけでなく、あるカメラを基準とした座標系で復元される。そこで、二つの部分形状を合成するためには、3次元相似変換のパラメータを求めればよい。3次元相似変換の自由度は、回転 3、並進 3、スケール 1 の計 7 であるため、3組の頂点の対応より計算できる (付録 A.1 参照)。

しかし、前節で得られた対応は、局所的な画像に関する記述子を用いているため、誤った対応を含む可能性がある。そこで、得られた対応を初期対応とし、RANSAC¹⁵⁾ を用いて、ロバストに合成のための変換を求めることを考える。その手順は以下の通りとなる。

- (1) 対応の候補の中から、ランダムに 3 組を選ぶ。
- (2) 選んだ 3 組の対応を用い、3次元相似変換のパラメータを計算する (付録 A.1)。
- (3) 求められた 3次元相似変換パラメータにより、片方の部分形状を変換し、対応候補の

表 1 実験環境

OS	Vine Linux 4.2
CPU	Inte Core 2 Duo E6850 3.0 GHz
メモリ	4 GB

表 2 シミュレーションデータの頂点数

部分形状	頂点数
A	179
B	179
C	219
D	219

表 3 対応付けの結果

部分形状対	得られた対応数	正対応数	誤対応数
図 5 (a)-(b)	31	17	14
図 5 (b)-(c)	26	6	20
図 5 (c)-(d)	39	24	15
図 5 (d)-(a)	26	7	19

中で、そのユークリッド距離が閾値未満となる対応の集合（支持集合）とその個数を求める。

- (4) ステップ 1~3 を十分繰り返す、最も多くの支持集合を持つ変換パラメータを求める。
- (5) 最も多くの支持集合の対応を全てを用いて、相似変換パラメータを再計算する。

6. 実験

提案法の有効性を確認するために、シミュレーションデータおよび実画像を用いた実験を示す。

シミュレーションでは、既知の形状データにテクスチャを貼り付けた部分形状を用い、提案記述子による対応付けと合成が可能かどうかを確認する。また、実画像を用いた実験では、実際に撮影した画像から部分形状を復元し、それらを合成することで、全周形状が復元できるかを確認する。実験に用いた計算機環境を表 1 に示す。

6.1 シミュレーションデータによる実験

データは図 4(a) に示す OpenGL の Teapot を用い、図 4(b) のように、Teapot を 45 度の重なりを含むように 4 つに分割して使用した。図中の赤い点が使用した頂点であり、分割した部分形状の頂点数を表 2 に示す。また、この実験においては、各部分形状 A, B, C, D のスケールの比を 1:2:1:2 とした。

表 3 にこれらの各部分形状間の対応付け結果を示す。見てわかる通り、かなりの数の誤対応が含まれていることがわかる。この実験では、正面画像のサイズは 51×51 であり、1 つ

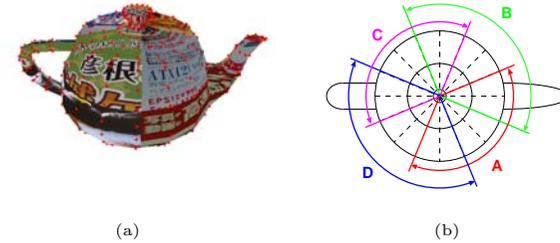


図 4 シミュレーションに用いた形状データ: (a) OpenGL の Teapot. (b) Teapot の分割方法。



図 5 実験に使用した 3 次元形状。

の頂点につき平均 23.7 個の keypoint が検出された。

これを初期対応として、RANSAC により部分形状を合成した結果を図 6 に示す。見てわかる通り、それぞれ正しく合成されており、RANSAC により、誤対応が除去されたことがわかる。更に全ての部分形状を合成した結果を図 7 に示す。これより、全周形状が正しく復元できていることが確認できる。この例において、それぞれ対応付けにかかった時間は平均で 82 秒、合成にかかった時間は 21.5 秒であった。

次に、各部分形状のスケール等倍と 2 倍、重なりを 22.5 度および 45 度とした場合の対応付けの結果を表 4 に示す。見てわかる通り、重なりが小さくなったり、スケールの差が大きい場合、得られる対応の個数が減少することがわかる。ただし、スケールの違いに関しては、生成する正面画像のサイズを大きくすることにより、対応可能であると考えられる。

6.2 実画像による実験

次に実画像による実験を行った。撮影した画像サイズは 1024×768 とした。撮影した画像を図 8 に、これらから復元された部分形状を図 9 に示す。これらの部分形状の復元には、通常の SIFT により得られた対応を用いた。見てわかる通り、これらの部分形状はほぼ正し



図 6 合成した形状: (a) 図 5 (a) と (b) の合成結果. (b) 図 5 (b) と (c) の合成結果. (c) 図 5 (c) と (d) の合成結果. (d) 図 5 (d) と (a) の合成結果.



図 7 復元された全周形状

表 4 重なりおよびスケールの変化と対応数 (正対応/総数)

部分形状対 (右を拡大)	等倍		2 倍	
	45 度	22.5 度	45 度	22.5 度
図 5 (a)-(b)	57 / 71	25 / 37	17 / 31	2 / 23
図 5 (b)-(c)	30 / 40	14 / 27	4 / 21	1 / 20
図 5 (c)-(d)	91 / 102	34 / 51	24 / 39	9 / 24
図 5 (d)-(a)	31 / 41	16 / 28	6 / 24	1 / 15

く復元できていることがわかる。また、この部分形状における頂点数を表 5 に示す。

図 9 の部分形状を用いて頂点の対応付けを行った結果を表 6 に示す。表 6 を見ると、図 9 (c) と (d) の正対応が 0 となっているが、これは図 8 を見てわかる通り、共通な部分なかったためである。このときの各頂点の正面画像における keypoint の数は平均で 35.6 個で



図 8 実験に用いた入力画像対

表 5 部分形状の特徴点の数

部分形状	頂点数
図 9 (a)	1011
図 9 (b)	806
図 9 (c)	785
図 9 (d)	650

表 6 得られた対応の数

部分形状対	得られた対応数	正対応数	誤対応数
図 9 (a)-(b)	113	94	19
図 9 (b)-(c)	140	131	9
図 9 (c)-(d)	35	0	35
図 9 (d)-(a)	29	17	12

あった。

これらを初期対応として、RANSAC により部分形状を合成した結果を図 10 に示す。見てわかる通り、ほぼ正しく合成できていることがわかるが、部分形状の縁の頂点は、その記述子が正面画像のものからではないため、対応が取れておらず、合成できていないこともわかる。このような頂点に対する対応が今後の課題となる。

7. おわりに

本論文では、部分的に復元された物体形状から全周形状を復元するための、各部分形状間のマッチングの方法を提案した。ここでは、多面体で復元された部分形状の頂点の法線方向から見た部分画像はスケールおよび回転に不変な画像で近似できることを利用した。そして、その正面部分画像内において、複数の keypoint を検出し、それらの SIFT 特徴量により、その頂点に関する記述子を定義した。このような記述長の異なる記述子間での対応を決定するために、投票法を用いた対応付け法を示した。そして、シミュレーションデータを用

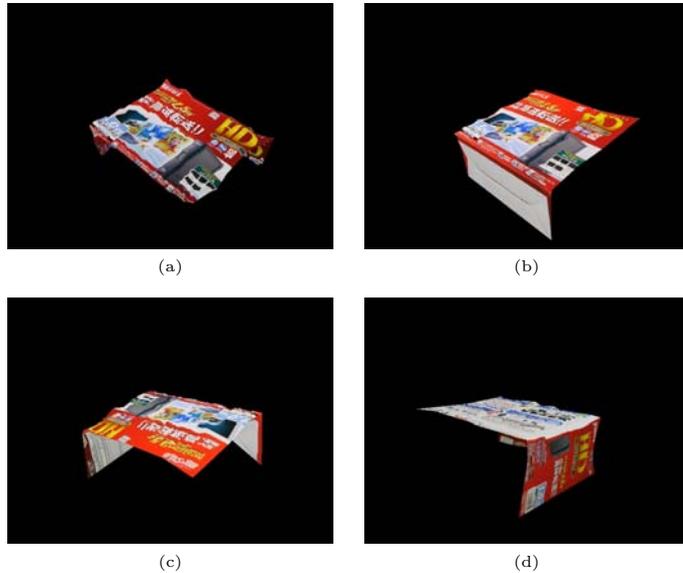


図 9 それぞれ図 8 (a)~(d) から復元した部分形状.

いた実験および実画像による実験により、提案記述子とその対応付け法の有効性を確認した。

今後の課題として、部分形状の縁の頂点に対するロバストな記述方法を検討することが必要となる。また、現在固定している正面画像のサイズに関しても、このサイズにより含まれる SIFT 記述子の数が変化し、対応付けに影響を及ぼすため、この正面画像のサイズの決定方法に関しても何らかの指針を得たいと考える。

謝辞

本研究の一部は、日本学術振興会科学研究費補助金 基盤研究 (C) (No. 21500167) の助成によった。



図 10 合成した全周形状.

参考文献

- 1) 池内克史, 倉爪 亮, 西野 恒, 佐川立昌, 大石岳史, 高瀬 裕: The Great Buddha Project —大規模文化遺産のデジタルコンテンツ化—, 日本バーチャルリアリティー学会論文誌, Vol.7, No.1, pp.103-113 (2002).
- 2) Snavely, N., Seitz, S. and Szeliski, R.: Photo tourism: exploring photo collection in 3D, *ACM Trans. Graphics (SIGGRAPH Proc.)*, Vol.25, No.3, pp.835-846 (2006).
- 3) 棕木雅之, 右田剛史, 天野 晃, 青山正人, 浅田尚紀: 再投影誤差最小化による建物画像列からの全周形状の一括復元, 情報処理学会研究報告, Vol.2004-CVIM-142, pp. 81-88 (2004).
- 4) 岡谷貴之: バンドルアジャストメント, 情報処理学会研究報告, Vol.2009-CVIM-167-37, pp.1-16 (2009).
- 5) Besl, P. and McKay, N.D.: A Method for Registration of 3-D Shapes, *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol.14, No.2, pp.239-256 (1992).
- 6) 増田 健: ICP アルゴリズム, 情報処理学会研究報告, Vol.2009-CVIM-168-24, pp. 1-8 (2009).
- 7) Wu, C., Clipp, B., Li, X., Frahm, J.-M. and Pollefeys, M.: 3D Model Matching with Viewpoint-Invariant Patches (VIP), *Proc. Comput. Vis. Patt. Recog.*, Anchorage, U.S.A., pp.1-8 (2008).
- 8) Lowe, D.: Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision*, Vol.60, No.2, pp.91-110 (2004).
- 9) 山田健人, 金澤 靖, 金谷健一, 菅谷保之: 2 画像からの 3 次元復元の最新アルゴリズム, 情報処理学会研究報告, Vol.2009-CVIM-168-15, pp.1-8 (2009).
- 10) 金谷健一, 菅谷保之: 高ノイズレベルにおける基礎行列の最尤推定, 情報処理学会研究報告, Vol.2007-CVIM-160-9, pp.49-56 (2007).
- 11) 菅谷保之, 金谷健一: 基礎行列の高精度計算法とその性能比較, 情報処理学会研究報告, Vol.2006-CVIM-153-32, pp.207-214 (2006).

- 12) 高橋茂雄, 中辻敦忠, 金谷健一: 未校正 2 画像からの 3 次元復元のための焦点距離計算の安定化, 情報処理学会研究報告, Vol.2003-CVIM-141-12, pp.79-86 (2003).
- 13) Kanatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, the Netherlands (1996).
- 14) 金谷健一, 菅谷保之, 新妻弘崇: 2 画像からの三角測量: Hartley vs. 最適補正, 情報処理学会研究報告, Vol.2008-CVIM-162-54, pp.335-342 (2008).
- 15) Fischler, M. and Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.6, No.25, pp.381-395 (1981).
- 16) 金谷健一: 画像理解 — 3 次元認識の数理 —, 森北出版, 東京 (1990).

付 録

A.1 部分形状の位置合わせの原理

3 次元形状 S と S' の位置合わせを考える. 形状 S の頂点を $\mathbf{x}_i, i = 1, \dots, N$, 形状 S' の頂点を $\mathbf{x}'_j, j = 1, \dots, M$ とする. これらの頂点のうち 3 組 $\{\mathbf{x}_a, \mathbf{x}'_a\}, \{\mathbf{x}_b, \mathbf{x}'_b\}, \{\mathbf{x}_c, \mathbf{x}'_c\}$ が対応しているとき, 形状 S と S' の位置合わせのための 3 次元相似変換は次のように計算できる.

- (1) 3 組の頂点の重心 $\mathbf{x}_g, \mathbf{x}'_g$

$$\mathbf{x}_g = \frac{1}{3} (\mathbf{x}_a + \mathbf{x}_b + \mathbf{x}_c) \quad (9)$$

$$\mathbf{x}'_g = \frac{1}{3} (\mathbf{x}'_a + \mathbf{x}'_b + \mathbf{x}'_c)$$

を計算し, その重心を原点にするよう, それぞれの形状を移動する.

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_g, \quad i = 1, \dots, N \quad (10)$$

$$\hat{\mathbf{x}}'_j = \mathbf{x}'_j - \mathbf{x}'_g, \quad j = 1, \dots, M$$

- (2) 移動後の対応する 3 組の頂点から, 形状間のスケール比 s

$$s = \frac{\|\hat{\mathbf{x}}_a\| + \|\hat{\mathbf{x}}_b\| + \|\hat{\mathbf{x}}_c\|}{\|\hat{\mathbf{x}}'_a\| + \|\hat{\mathbf{x}}'_b\| + \|\hat{\mathbf{x}}'_c\|} \quad (11)$$

を求め, 3 次元形状 S' のスケールを S に合わせる.

$$\tilde{\mathbf{x}}'_j = s\hat{\mathbf{x}}'_j, \quad j = 1, \dots, M \quad (12)$$

- (3) モーメント行列

$$\mathbf{M} = \sum_{i=1}^3 N[\hat{\mathbf{x}}_i]N[\tilde{\mathbf{x}}'_i]^\top \quad (13)$$

を計算し, 次のように特異値分解する.

$$\mathbf{M} = \mathbf{V} \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{U}^\top \quad (14)$$

ここで, $\text{diag}(a, b, c)$ は a, b, c を対角要素に持つ対角行列を表す. そして, 回転行列 \mathbf{R} を以下の式より求める¹⁶⁾.

$$\mathbf{R} = \mathbf{V} \mathbf{U}^\top \quad (15)$$

- (4) 回転行列 \mathbf{R} の行列式が負であれば, 鏡像であるので符号を反転させる.

$$\mathbf{R} = -\mathbf{R} \quad (16)$$

- (5) 形状 S' を得られた回転行列 \mathbf{R} で回転させ, 形状 S と合成する.

$$\mathbf{x}''_j = \mathbf{R}\tilde{\mathbf{x}}'_j, \quad j = 1, \dots, M \quad (17)$$