

画像からの形状復元に適合した三角網の生成

村田 正和* 中辻 敦忠† 菅谷 保之* 金谷 健一*

* 岡山大学工学部情報工学科 † NEC エンジニアリング第二システムソリューション事業部

特徴点に基づく画像からの3次元復元では通常、それらの特徴点を頂点とする三角網が定義され、シーンが多面体として表示される。しかし、シーン自身が多面体の場合は物理的な辺に矛盾する辺が現れる。本論文ではそのような誤りを検出するフィルタを用いて初期の三角網を物理的な辺に適合するように自動的に変換する新しい手法を提案する。さらに周囲の偽の境界を除去する手法も示す。これらは何のしきい値を設定することなく実行できる。そして実画像を用いて、提案手法が従来のランダムに探索する方法に比べて、誤りを訂正する能力がより高いことを示す。

Generating a Triangular Mesh Adapted for Shape Reconstruction from Images

Masakazu Murata* Atsutada Nakatsuji† Yasuyuki Sugaya* Kenichi Kanatani*

*Department of Information Technology, Okayama University, Okayama 700-8530 Japan

*System Solutions Division II, NEC Engineering, Ltd., Fuchu-shi, Tokyo 183-8502 Japan

In reconstructing 3-D from images based on feature points, one usually defines a triangular mesh that has these feature points as vertices and displays the scene as a polyhedron. However, if the scene itself is a polyhedron, some of the displayed edges may be inconsistent with physical edges. This paper presents a new technique for automatically making a given triangular mesh compatible with the physical shape, using a template that can detect shape inconsistencies. We also present a technique for removing spurious occluding edges. All the procedures do not include any thresholds to be adjusted. Using real images, we demonstrate that our method has higher capability to correct inconsistencies than the conventional methods based on random search.

1. まえがき

画像からシーンの3次元形状を復元するときの大きな問題は、復元した形状をどう表現するかである。ステレオ視を用いれば、カメラ位置や内部パラメータが校正済みなら両方のカメラから見えるどの点も奥行きが計算できるので、画素間補間によってシーンを連続曲面として表現できる。カメラが多数あるときは空間をボクセルに分割し、「空間切除法 (space carving)」[5] によってカメラから見えるボクセルを選んで物体表現とすることもできる。

しかし未校正カメラからの3次元復元では、連続ビデオを用いる因子分解法[8]にせよ、エッジ線幾何学に基づく3次元復元[1]にせよ、画像から特徴点を抽出し、その追跡や対応づけによってそれらの3次元位置を計算する。そのような3次元点の集合からシーンの表現を得るには通常、それらを頂点とする三角網が定義され、テクスチャマッピングされた多面体が表示される。

そのための三角網は、指定したフレーム中の2次元点の集合に、それらを頂点とするドロネー網[7]を定義すればよい。これによって形のバランスのとれ

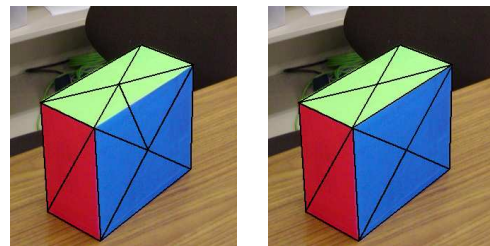


図 1: (a) 物理的な辺に矛盾する三角網。(b) 物理的な辺に適合する三角網。

たメッシュが自動的に生成できる。しかし、シーンが多面体の場合に問題が生じる。

室内や建物などの人工的な環境では物体のほとんどが多面体であり、その頂点が画像の特徴点として選ばれやすい。しかし、隣接する頂点が物理的な辺で結ばれている場合に、単に特徴点の集合にドロネー網を定義すると、そのような物理的な辺に矛盾する辺が定義され、それに基づく多面体表現が真の形状が反映されないことがしばしば起こる。

例えば、図 1(a) の画像の物体頂点が特徴点として選ばれたとする。これから図 1(a) に示すドロネー網生成されるが、得られる多面体表現は物体形状を正しく表していない。一方、図 1(b) の三角網から得られる多面体表現は物体形状を正しく表している。本

*700-8530 岡山市津島中 3-1-1, Tel. (086)254-8173
{murata,sugaya,kanatani}@suri.it.okayama-u.ac.jp
†183-8502 東京都府中市住吉町 5-22-5, Tel. (042)360-4048
a-nakatsuji@pb.jp.nec.com

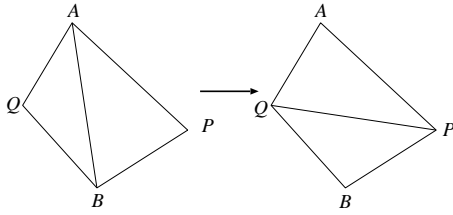


図 2: 対角辺の入れ換え (フリップ)。

論文の目的は図 1(a) のような物理的な辺と矛盾する三角網を図 1(b) のような物理的な辺に適合する三角網に自動的に変換することである。

以下, 2, 3 節でその原理を述べ, 4~7 節で提案手法を説明する。そして, 8 節で実画像を用いて提案手法が従来のランダムに探索する方法に比べて誤りを訂正する能力がより高いことを示し, 9, 10 節でまとめる。付録に, 周囲の偽の境界を除去する手法とその実験結果を示す。提案手法の特徴は, 何らにしい値も設定する必要がないことである。

2. 三角網の適合性

最適な三角網の生成に関して従来からよく研究されているのは, (1) 多数の頂点による密な三角網を形状の記述を損なわないように簡素化する手法, および, (2) 少数の頂点からなる疎な三角網から出発して, 観測データに適合するように辺や頂点を追加して形状の記述を詳細化する方法である。最近では, 前者では例えば Vogiatzis [9] らが真の形状は平面部分が多いという事前確率を用いるベイズ推定にアンニリングを適用し, 後者では例えば Yu [10] らが物体の形状と光反射モデルを推定しながら三角網を詳細化する手順を述べている。

それに対して, 与えられた頂点に対して辺の接続のみを最適化する研究は著者らが知る限り, 飯田ら [2], Morris ら [6], 栗原ら [4] のみである。その基本的な原理は, カメラの撮像をアフィン変換で近似して¹, その物体を写した 2 画像を比較することである。

このとき, ある三角形パッチが物体の平面部分であれば, 一つの画像の三角形パッチは他の画像の対応する三角形パッチにアフィン変換で写像される。したがって, その三角形を他方の三角形にテクスチャマッピングすると輝度値の差が 0 になる。そうでなければその三角形は平面部分ではないから, その三角形のどれかの辺を図 2 のように対角辺と交換 (フリップ) する。これを繰り返せば物意的に適合する

¹ 実際のカメラでは, 厳密には両者は射影変換で結ばれるが, 小さく分割した各々の三角形パッチ同士を考える限り, 射影変換とアフィン変換の差は無視できる。

表現が得られる [6]。

しかし, 実際には特徴点の対応が厳密でなかったり, 物体に凹凸があったり, 明るさが見る向きに依存したりするため, 平面部分の三角形パッチを写像しても輝度差が厳密に 0 になることはない。そこで従来の研究 [2, 4, 6] では, 辺をランダムに選んではその両側の三角形のアフィン変換後の輝度差²を計算し, 対角辺と交換したほうが輝度差が小さくなれば交換し, これをどの三角形パッチも変化しなくなるまで反復している。以下, これをランダムフリップと呼ぶ。

3. 提案手法の原理

以下, 簡単のため三角網の一つの辺はその両側の三角形が共に平面部分のとき, その辺を「正しい辺」, そうでない辺を「誤った辺」と呼ぶ。正しい辺は, 両側の三角形が同一平面上にある場合と, その辺を境界とする異なる平面上にある場合がある。一方, 誤った辺はそれが物理的に異なる面上の 2 点を結ぶ場合である。

従来の方法の問題点は, 三角網が与えられたとき, どれが正しい辺でどれが誤った辺であるかを直接には知ることができず, その判定は辺をその対角辺と交換し, それにテクスチャマップを施した輝度差を計算してみて初めて可能になる。したがって, 次にどの辺を入れ換えてみるかはランダムに決めるしかなく, 不要な入れ換えを何度も繰り返すことになり, 多くの反復回数を要する。また, 誤った辺でも, その近辺の辺がそれより入れ換わると, その影響でその辺を入れ換えても正しくならないことも生じ, 正しい解に収束しないこともある。

このことから, もしあらかじめ各辺の“誤りの程度”を測る方法があって, その値の大きいものを優先して入れ換えれば, 正しい形状に収束しやすくなると予想される。

その誤りの程度を測る原理を図 3 に示す。異なる面は色あるいは明るさが異なるとする。画像中に図 3 の左のような二つの三角形からなる部分があり, その 4 頂点が特徴点として選ばれているとする。この 4 点に三角形パッチを張って多面体を生成しようとするれば, AC を辺とするか BD を辺とするかの二通りの方法がある。

図 3 の中央の上の図は正しい辺 AC を選んだ多面体にテクスチャマッピングしたものを別の方向から見たものである。それに対して, 誤った辺 BD を選

² 以下, 本論文ではカラー画像を考え, R, G, B 値の差の二乗和を「輝度差」, その表示を「差画像」と呼ぶ。

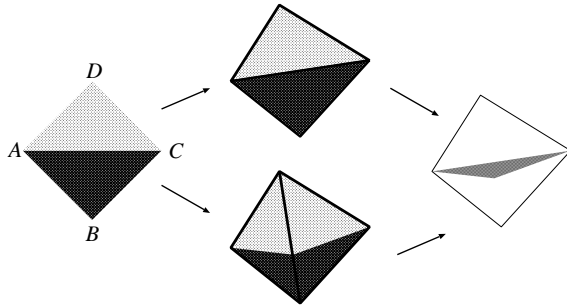


図 3: 正しい辺による 3 次元形状と誤った辺による 3 次元形状の比較 .

んでテクスチャマッピングして別の方向から見ると図 3 の中央の下の図になる . 両者の差をとると図 3 の左のようになり , 誤った辺を横切る偏った三角形領域に差が現れている .

実際に撮影したもう一つの画像には正しいテクスチャが写っているはずであるから , もし誤った辺の両側の三角形の内部をもう一つの画像の対応する三角形の内部に写像して差をとると , 図 3 の左のような三角形領域に差が現れるはずである .

図 4 の画像 (a), (b) は多面体物体を撮影した 2 画像にドロネー網による三角網を重ねて表示したものである . 画像 (c) は画像 (b) の各三角形パッチを画像 (a) の対応する三角形にテクスチャを写像したものである . 画像 (d) は画像 (a), (c) の差画像である . 図 4(d) を見ると , 誤った辺に交差する細い三角形領域に大きな差が現れていることが分かる .

4. テンプレートによる不適合の検出

前節の結果より , テクスチャマップした差画像を作ると , 誤った辺にはそれを横切る細長い三角形領域が現れることがわかる . そこでそれを検出することを考える . このために , 辺の両側の三角形を $l \times l$ 正方形領域³ $ORST$ の $\triangle OSR$, $\triangle OST$ にそれぞれアフィン変換し , それと偏った三角形を検出するテンプレートとの相関 (各画素の積の和) の絶対値を計算する . このテンプレートとして次の関数 $T(x, y)$ を用いる⁴ .

$$T(x, y) = \begin{cases} e^{-\frac{(x+y-l)^2}{2\alpha^2(x-y-l)^2}} & x+y < l, x \geq y \\ T(y, x) & x+y \leq l, x < y \\ -T(l-y, l-x) & x+y > l \end{cases} \quad (1)$$

図 5(a) はこれを濃淡表示したものである . その等高線は R, T を端点とし , 対角線 OS 上に頂点を持

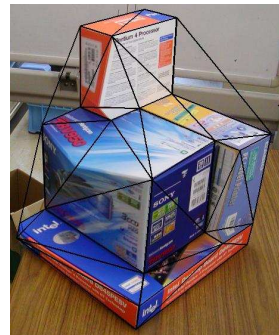
³実験では $l = 100$ とした .

⁴実験では $\alpha = 0.1$ とした .

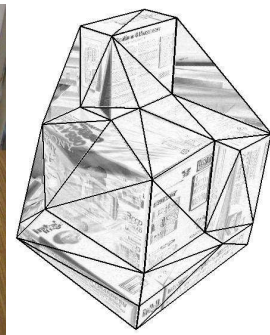


(a)

(b)



(c)



(d)

図 4: (a), (b) 撮影した 2 画像とドロネー網による三角網 (辺数 58) . (c) 画像 (b) の各三角形パッチを画像 (a) の対応する三角形にテクスチャマッピングしたもの . (d) 画像 (a), (c) の差画像 (濃い黒ほど大きい値) .

つ折れ線となっている . そして , 対角線 OS に関して対称であり , $\triangle RST$ は対角線 RT に関して反対称 (対称な値の符号を換えたもの) である . ただし対角線 RT 上の画素の値は 0 とし対角線 OS の上下 2 画素とフレームの周囲の 2 画素の値は 0 とする⁵ .

図 5(b) はその対角線 OS 上の切り口の値のグラフであり , 平均 $l/\sqrt{2}$, 標準偏差 $\alpha l/\sqrt{2}$ の正規分布の片側の符号を換えたものになっている .

反対称なテンプレートを用いるのは , 差の現れる三角形領域がどちら側に偏っているかあらかじめわからないためである . 反対称にすれば , その領域が一方に偏っていても反対側はほとんど 0 であるから , 相関の絶対値を計算することによって偏った領域を検出できる . さらに , テクスチャマッピングのずれのためにパッチ全体に分布するランダムな誤差パターンはこのような反対称なテンプレートによって正負が打ち消される .

⁵三角形パッチの周辺を除外するのは , 特徴点の検出の誤差によって正しい辺であっても物理的な辺と多少ずれている可能性があるからである .

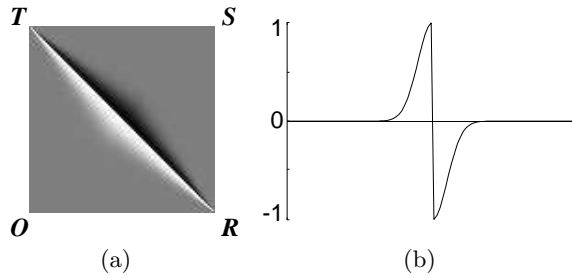


図 5: (a) 誤った辺を検出するテンプレートの濃淡表示 (白が正, 黒が負). (b) 対角線 OS に沿った切り口の値.

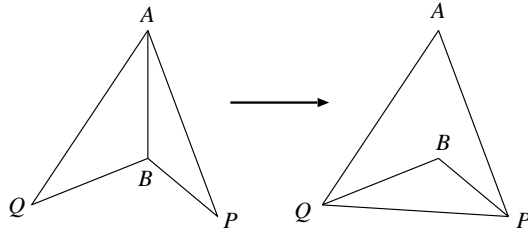


図 6: 四辺形 $APBQ$ が凸でない場合は, 対角辺を入れ換えると裏返しの三角形パッチができる.

5. 辺の評価値の計算

具体的には与えられた辺 AB の不適合の度合いの評価値 $w(AB)$ を次のように計算する.

1. 辺 AB の片側に一つの三角形しかない場合は境界辺であるから, $w(AB) = -1$ を返す.
2. 辺 AB の両側の三角形を $\triangle ABP$, $\triangle ABQ$ とするとき, 四辺形 $APBQ$ が第 1 画像と第 2 画像の両方で凸でなければ $w(AB) = 0$ を返す.
3. そうでなければ, 第 1 画像の四辺形 $APBQ$ をテンプレート領域 $ORST$ に射影変換し, 輝度値を書き込む.
4. 第 2 画像の $\triangle ABP$, $\triangle ABQ$ をそれぞれテンプレート領域の $\triangle OSR$, $\triangle OST$ にアフィン変換し, 輝度値をそこに書かれている値から引く.
5. 第 2 画像の四辺形 $APBQ$ をテンプレート領域 $ORST$ に射影変換し, 輝度値をそこに書かれている値に足す.
6. 第 1 画像の $\triangle ABP$, $\triangle ABQ$ をそれぞれテンプレート領域の $\triangle OSR$, $\triangle OST$ にアフィン変換し, 輝度値をそこに書かれている値から引く.
7. その結果と検出テンプレートとの相関の絶対値を $w(AB)$ とする⁶.

ステップ 2 は, 四辺形 $APBQ$ が凸であれば図 2 のように対角辺を入れ換えることができるが, 凸でなければ図 6 のように裏返しの三角形パッチが生成されるので交換しないという意味である.

⁶R, G, B ごとに別々に相関を計算し, 最後にそれらの二乗和の平方根をとる.

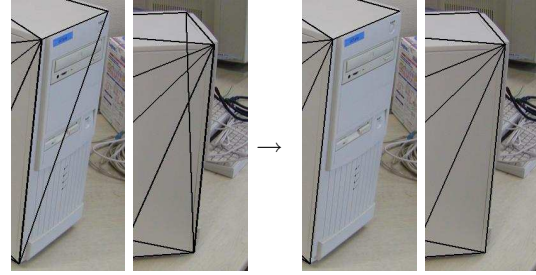


図 7: 2 画像間で反転する三角形パッチ. 境界の辺によって生じる場合はその辺を除去する.

画像座標が $A(a_1, a_2)$, $B(b_1, b_2)$, $P(p_1, p_2)$, $Q(q_1, q_2)$ のとき, 四辺形 $APBQ$ が凸である条件は次のように書ける [3].

$$\begin{vmatrix} p_1 - a_1 & q_1 - a_1 \\ p_2 - a_2 & q_2 - a_2 \end{vmatrix} \cdot \begin{vmatrix} p_1 - b_1 & q_1 - b_1 \\ p_2 - b_2 & q_2 - b_2 \end{vmatrix} < 0 \quad (2)$$

四辺形 $APBQ$ が第 1 画像上で凸であっても対応する四辺形が第 2 画像上でも凸であるとは限らないから, 両方の画像で形をチェックする必要がある.

ステップ 3, 4 は図 4(d) に示す差画像をテンプレート領域に射影変換することに相当する. 射影変換は直線を保存するから, テンプレート領域にも図 4(d) のような三角形領域が現れる. ステップ 4, 5 はこれを第 1 画像と第 2 画像の役割を入れ換えて行うものであり, これにより第 1 画像と第 2 画像に対して対称な操作になる.

6. 初期三角網の生成

まずドロネー網を第 1 画像の特徴点の集合に対して定義し, それに対応する第 2 画像上の各三角形パッチの符号をチェックする. ただし $\triangle ABC$ の符号 $\text{sgn}(ABC)$ とは A, B, C が反時計回りに回転するとき正, 時計回りのとき負, 線分に退化した場合に 0 と定義する. これは $A : (a_1, a_2)$, $B : (b_1, b_2)$, $C : (c_1, c_2)$ とすると次のように計算される⁷.

$$\text{sgn}(ABC) = \text{sgn} \left(\begin{vmatrix} b_1 - a_1 & c_1 - a_1 \\ b_2 - a_2 & c_2 - a_2 \end{vmatrix} \right) \quad (3)$$

右辺の $\text{sgn}()$ は符号関数であり, 引数が正, 負, 0 のときそれぞれ 1, -1, 0 を返す.

符号が異なる三角形パッチがあるときは, それが境界辺を含んでいれば, それらの辺を三角網から除去する (図 7). 内部の三角形の符号が異なるときは, その周りの三角形の符号がすべてそろっていればそ

⁷右辺の sgn の引数は AB, AC を 2 辺とする平行四辺形の符号つき面積である [3].

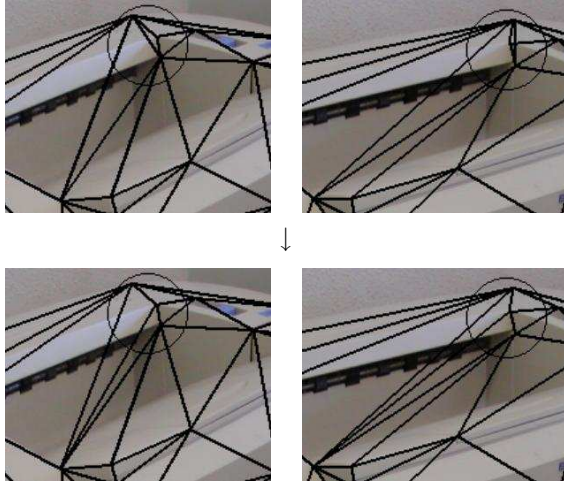


図 8: 内部の三角形で裏返しが生じる場合は辺を入れ換えてこれを解消する。

の三角形のどれかの辺を対角辺に入れ換えることによって裏返しを解消する⁸ (図 8)。

初期三角網の各三角形パッチの符号がそろっていれば、以下で対角辺を入れ換えるのは第 1 画像、第 2 画像ともに図 6 のような凹辺形が生じないときであるから、途中で裏返しの三角形パッチが生じることはない。

7. 三角網の最適化の手順

前節のように生成した初期三角網を 3 次元形状に適合するように次のように再編する。

1. すべての辺の評価値 w を計算する。
2. 評価値が最大の辺 AB の評価値を $w(AB)$ とする。
3. $w(AB) = 0$ なら終了する。
4. 三角網の辺 AB を対角辺 PQ に入れ換え、その三角網に対して辺 PQ の評価値 $w(PQ)$ を計算する。
5. $w(PQ)$ を直前に計算した $w(AB)$ と比較し、 $w(PQ) > w(AB)$ であれば、三角網の辺 PQ を除去し、辺 AB を復活させ、元の三角網に戻し、 $w(AB) = 0$ とする。
6. そうでなければ、辺 PA, PB, QA, QB のうち、現在の評価値が 0 でないものについて、変更した三角網に対する $w(PA), w(PB), w(QA), w(QB)$ を計算し直す。
7. ステップ 2 に戻る。

⁸ 原理的にはより大域的に裏返しが起こり得るが (例えば第 2 画像が第 1 画像に写っている面を裏側から見ているときなど)、本論文ではシーンが符号のそろった三角形パッチからなる凸の多面体で表示できる場合のみを考え、辺の入れ換えで正しい表現が得られ場合のみを対象とする。

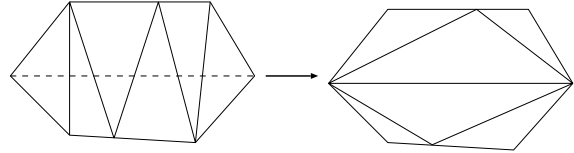


図 9: 望ましい修正には数段階の辺に入れ換えを要する場合。破線は物理的な辺。



図 10: 図 4 の三角網の最適化の結果。(a) 提案手法 (100%, 2 ラウンド, 8.21 秒)。(b) ランダムフリップ (91.5%, 270 回, 9.05 秒)。

この手順では w の絶対的な大きさに意味はなく、大小比較のみを行っているので、何らのしきい値も設定する必要がない。また、一度調べた辺は再びは調べないので、辺の個数だけの回数の計算で終了する。

以上の最適化によって、入れ換えれば正しく修正できるような辺は修正されるはずである。しかし、数段階の入れ換えを必要とする場合、例えば図 9 のように一つの物理的な辺を多くの三角形パッチが横切るような場合は不十分である。そこで上記の手順を「ラウンド」と呼び、それが終了後に改めて辺の評価値 w を計算し直して同じ操作を辺の交換が生じなくなるまで繰り返す。

8. 比較実験

実画像を用いて実験を行ない、2 節に述べたランダムフリップと比較した。栗原ら [4] は照度変化を打ち消す必要性を指摘している⁹、本実験のランダムフリップでは両画像の各三角形パッチごとに各画素の輝度値に線形変換を行い、パッチの輝度値の総和が 0、分散が 1 になるように正規化した¹⁰。そして、辺の個数に等しい回数だけ連続して辺の入れ換えが起こらなくなったところで終了した。

図 10(a) は図 4 の三角網を提案手法によって最適

⁹ 提案手法では反対称なテンプレートを用いるので、三角形内部に一樣に生じる照度変化は打ち消される。

¹⁰ 一つの線形変換によって R, G, B 値の全部の和と分散を 1, 0 に正規化する。この正規化は、辺が入れ替わって三角形パッチが変化することに行う。

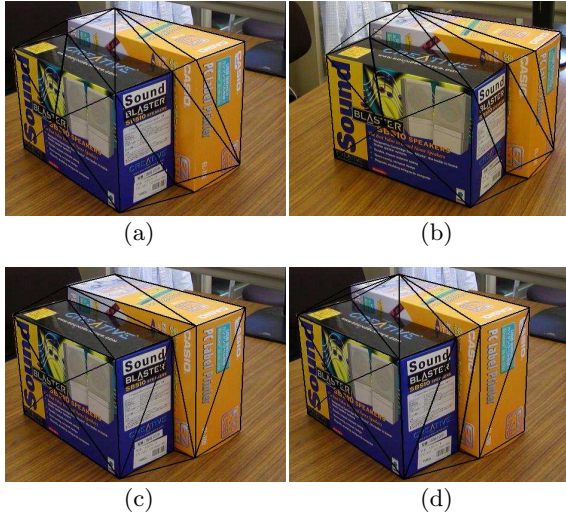


図 11: (a), (b) 第 1, 2 画像と初期三角網 (辺数 31). (c) 提案手法 (100%, 3 ラウンド, 5.29 秒). (d) ランダムフリップ (75.2%, 159 回, 5.43 秒).

化したものである。これは 2 ラウンドで終了した。図説中のカッコ内の数字は正解率 (%) および実行時間 (秒) である。ただし、正解率は辺の総数 (境界辺は除く) に対する正しい (すなわち両端点と同じ平面上にある) 辺の数の割合である。また、CPU には Pentium 4 3.2B GHz, 主メモリ 2GB, OS には Linux を用いた。

図 10(b) は同じ例にランダムフリップを適用した結果である。図説中のカッコ内の数字は正解率, チェックした辺の数, および実行時間である。ただし、辺の選択のランダム性により, 同一例でも試行の度に結果が変化するので 10 回の試行の平均値をとった (図 10(b) は結果の一例)。

図 11~14 は別の画像例である。(a), (b) は第 1, 第 2 画像上に初期三角網を表示したものであり, (c), (d) はそれぞれ図 10(a), (b) に対応する結果である。

9. 考察

これらの実験から, ランダムフリップで正しく修正できる場合もあるが, 探索のランダム性によって結果が必ずしも一定しないことがわかる。そして, 全体的には提案手法のほうがより適合した修正ができています。これは, ランダムフリップが三角形パッチ全体の輝度差を評価しているため, 各面の色や明るさが似ている場合にパッチ全体では差が小さくなって検出しにくいのに対して, 提案手法は見え方の差が誤った辺を横切る細い三角形領域に強く現れることに着目しているため, 誤りに検出能力が高いと考えられる。

実行時間は, ランダムフリップではどこで終了す

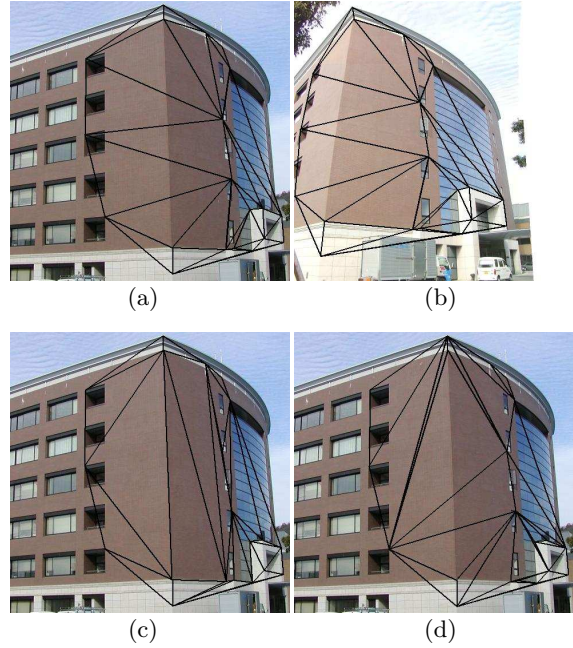


図 12: (a), (b) 第 1, 2 画像と初期三角網 (辺数 47). (c) 提案手法 (100%, 3 ラウンド, 9.69 秒). (d) ランダムフリップ (96.2%, 313 回, 7.62 秒).

るかに依存し, 合理的な基準がないが, 本実験の条件で終了させると辺の数が 50 本程度なら提案手法とほとんど同程度である。しかし, 辺やパッチの数が増えると, ランダムフリップでは比較する一つ一つの三角形パッチの面積が小さくなり, 全体では実行時間があまり増加しないのに対して, 提案方法ではどの三角形パッチも同じ検出プレート領域に写像しているので, 計算時間がパッチ数にほぼ比例している。

10. まとめ

本論文では, 多面体シーンの三角網による 3 次元表現から物理的な辺と矛盾する辺を除去する新しい手法を提案した。これは 2 画像を用いて, 一方の画像を他方へテクスチャマッピングした輝度値の差を検出するフィルタを導入し, 初期三角網を物理的な辺に適合するように変換するものである。これは何のしきい値を設定することなく実行できる。そして実画像を用いて, 提案手法が従来のランダムに探索する方法に比べて, 誤りを訂正する能力がより高いことを示した。

本論文では 2 画像を比較する方法を述べたが, 3 枚以上の画像があれば, それらを互いに比較することによってより正確な表現が得られる。

なお, 物体の頂点を特徴点としてドロネー三角網を生成すると, 物体の境界に凹部があればその外側

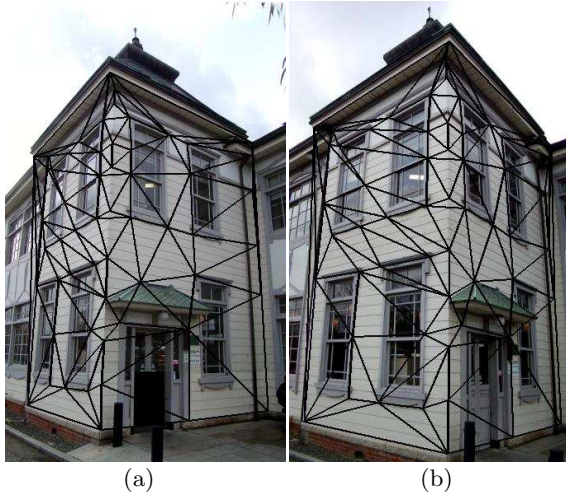


図 13: (a), (b) 第 1, 2 画像と初期三角網 (辺数 157). (c) 提案手法 (98.7%, 7 ラウンド, 83.53 秒). (d) ランダムフリップ (99.3%, 2611 回, 19.60 秒).

に背景を横切る辺ができる．これを「偽の境界」と呼ぶことにする．上記の実験ではこれは無視したが，偽の境界はその両側で画像の性質が同じかどうかを判定して除去することができる．その手順は付録に示す．

謝辞: 本研究の一部は文部科学省科学研究費基盤研究 C (2) (No. 15500113) によった．

参考文献

- [1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K., 2000.
- [2] 飯田亮介, 出口光一郎, 平面を利用したステレオ画像からの任意視点画像の合成, 情報処理学会研究報告, 99-CVIM-114-10, pp. 73-80, January 1999.
- [3] 金谷健一, 「形状 CAD と図形の数学」, 共立出版, 1998.
- [4] 栗原祐介, 日吉久礎, 金谷健一, 太田直哉, 画像の一致度を考慮したサーフェスモデルの構築, 情報処理学会研究報告, 2001-CVIM-127-8, pp. 51-58, May 2001.
- [5] K. Kutulakos and S. Seitz, A theory of shape by space carving, *Proc. Int. Conf. Comput. Vision*, September 1999, Kerkyra, Greece, pp. 307-314.

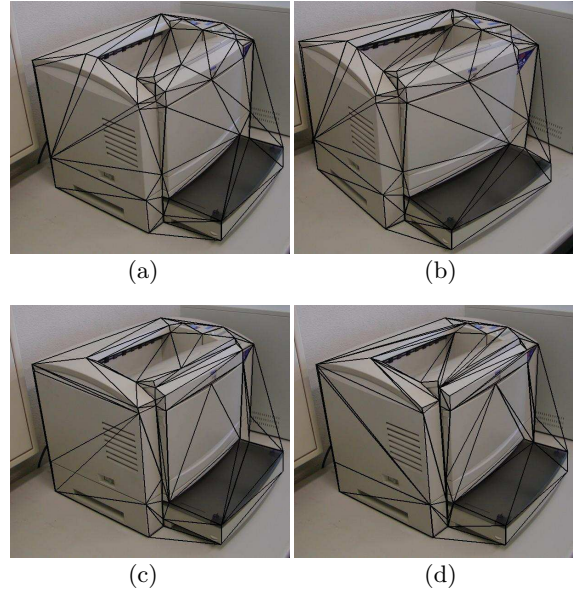


図 14: (a), (b) 第 1, 2 画像と初期三角網 (辺数 114). (c) 提案手法 (96.2%, 4 ラウンド, 34.75 秒). (d) ランダムフリップ (83.8%, 866 回, 19.70 秒).

- [6] D. D. Morris and T. Kanade, Image-consistent surface triangulation, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Hilton Head, SC, U.S.A., June 2000, Vol. 1, pp. 332-338.
- [7] F. Preparata and M. Shamos, *Computational Geometry*, Springer, Berlin, Germany, 1985.
- [8] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography—A factorization method, *Int. J. Comput. Vision*, 9-2 (1992-10), 137-154.
- [9] G. Vogiatzis, P. Torr and R. Cipolla, Bayesian stochastic mesh optimization for 3D reconstruction, *Proc. British Machine Vision Conf.*, Norwich, U.K., September 2003, Vol. 2, pp. 711-718.
- [10] T. Yu, N. Xu and N. Ahuja, Shape and view independent reflectance map from multiple views, *Proc. 8th Euro. Conf. Comput. Vision*, Prague, Czech., May 2004, Vol. 4, pp. 602-615.

付録：偽の境界の除去

偽の境界を 1 辺とする三角形パッチの内部は背景であり，二つの画像で見える部分が異なるから，2 画像を比較して，一方の三角形パッチを他方にアフィン写像して比較して輝度差を検出すればよいように思える．しかし，そのしきい値の設定が困難である．また，背景が白壁のように，どこから見ても同じように見えるものには輝度差が現れない．また，不適切なしきい値を用いて真の境界を除去してしまうと，内部の形状が正しく表せない．そこで何のしきい値も用いない方法を考える．

真の境界は物体の外周に沿うので，エッジ画像を作ればエッジ上にあると期待される．一方，偽の境界は背景を横切るため，エッジ上になかったり背景のエッジを横切ったりしている．そこで，外周辺の各点について，それがその辺の方向のエッジ上にあ

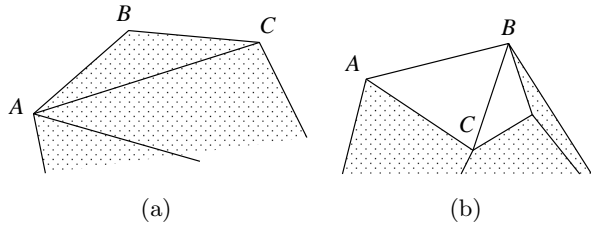


図 15: (a) 辺 AB, BC は真の境界と判定する。(b) 辺 AB が偽の境界と判定されれば辺 AC, CB が外周辺となり、これらば再帰的に判定される。

れば正、その辺に直交するエッジ上であれば負の値をとるような評価値を与え、その边上のその値の平均値が大きいかほどその辺は真の境界である可能性が高いと判定する。

そこで、調べようとする外周辺の方向ベクトルを v とするとき、その边上の各点の境界らしさの評価値を次のように定義する。

$$E = 2(v, \nabla I)^2 - \|v\|^2 \|\nabla I\|^2 \quad (4)$$

ただし $\nabla I = (I_x, I_y)^\top$ はその点の輝度値 I の勾配ベクトルであり、 I_x, I_y はそれぞれ x, y 方向の微分である。これは標準偏差 3 (画素) の 13×13 のガウス平滑化微分フィルタによって計算する¹¹。 $(v, \nabla I)$ はベクトル $v, \nabla I$ の内積であり、 $\|\nabla I\|$ は ∇I のノルムである。

定義より画像の輝度勾配 ∇I が v の方向にあるとき E は最大値 $\|v\|^2 \|\nabla I\|^2$ を、 v と直交するとき最小値 $-\|v\|^2 \|\nabla I\|^2$ をとり、 v と 45° をなすとき 0 となる。これらは v の向きづけに依存しない。

この値を边上で積分し¹²、その長さで割った \mathcal{E} をその辺の境界らしさの尺度とする¹³。そして、この値をその内部の辺の値と比較して除去するかどうかを決定する。このため、しきい値を設定する必要はない。

この方法は本文で述べた三角網の変換と異なり、1 画像のみから判定できる。具体的な手順は以下の通りである。

1. すべての辺の内、その片側にしか三角形が存在しないものに「外周辺」のラベルを付ける。
2. 外周辺の内、それに隣接する三角形の残りの 2 辺のいずれかが外周辺であるような辺を「真の境界」として確定する (図 15(a))。
3. そうでない外周辺 AB と、それに隣接する三角形 ABC の残りの 2 辺 AC, BC の評価値 $\mathcal{E}(AB)$,

¹¹ あらかじめ原画像に平滑化やエッジ検出を行うのではなく、考えている画素ごとにその 13×13 近傍から計算する。

¹² 区間を 100 分割する台形公式によって数値積分を行った。

¹³ R, G, B に対する値の和をとる。

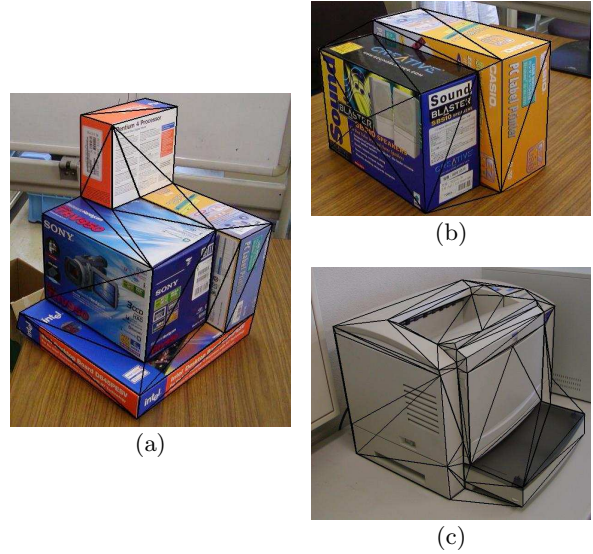


図 16: (a) 図 10(c) の三角網から偽の境界を除去した結果。(b) 図 11(c) の三角網から偽の境界を除去した結果。(c) 図 14(c) の三角網から偽の境界を除去した結果。

$\mathcal{E}(AC), \mathcal{E}(BC)$ を計算する。

4. $\mathcal{E}(AB) \geq \mathcal{E}(AC)$ または $\mathcal{E}(AB) \geq \mathcal{E}(BC)$ なら辺 AB を「真の境界」として確定する。
5. そうでなければ辺 AB を「偽の境界」として三角網から除去し、辺 AC, BC に「外周辺」のラベルを付け、それらに対して同じ操作を再帰的に行う (図 15(b))。
6. 以上をすべての外周辺が「真の境界」として確定されるまで反復する。

2 画像が利用できるときは、上記の手順を両方の画像に対して並行して実施し、両方の画像で判定が一致したときのみ偽の境界として除去する。これは、2 画像で判定が分かれるようなあいまいな辺は除去しないという意味である。その理由は、偽の境界が残っても実際上の応用では大きな支障ではないのに対して、真の境界を誤って除去すると内部の形状が正しく表せなくなる危険があるからである。3 画像以上が利用できれば多数決によって判定はさらに確実となる。

図 16(a), (b), (c) はそれぞれ図 10(a), 11(c), 14(c) の三角網を 2 画像を用いて境界辺を除去した結果である。図 16(a), (b) のような、物体と背景とで色や輝度値にかなり差がある場合は偽の境界辺が正しく除去されている。一方、図 16(c) のような、偽の境界の内部の真の境界の物体側と背景側とで色や輝度値が非常に類似している場合は、判定が安全側に働いて完全には除去されない辺が残っている。これは、それらの辺を除去すると内部の正しい辺まで除去され、正しい物体形状が記述できなくなるためである。