

連続カラー画像を用いた双方向投票による 物体形状と色の推定

財満 久義[†] 山本 強^{††}

zaima@cc.hokudai.ac.jp

[†]北海道大学工学研究科, ^{††}北海道大学大型計算機センター

ビデオ撮影画像などを用いて、低次元情報から高次元情報の推定を行う手法はこれまでもいくつか提案されている。多くの手法では、エッジなどの特徴点を抽出し、フレームにまたがって追跡することにより幾何情報の推定を行っている。この場合、特徴点抽出の精度と追跡の精度が重要な問題となる。

本報告では、特徴点の抽出と追跡を行わず、幾何情報を推定する手法について報告する。はじめに、逆方向のレイを利用して各ボクセルの候補色を1次元配列として獲得する。これを順方向のレイを利用してひとつの色に絞り込む。最終的に、色情報を含むボリュームデータを得ることができる。

Estimation of Shape and Color from Bi-directional Voting using a Sequence of Color Frames

Hisayoshi Zaima[†], Tsuyoshi Yamamoto^{††}

[†]Faculty of Engineering, Hokkaido University,

^{††}Computing Center, Hokkaido University

There are many methods for estimating high-dimensional geometric information from low-dimensional information such as using video frames. Many methods extract characteristic elements such as edges and trace their movement for adjacent frames to estimate geometric information. In this case, raising the accuracy of extraction and tracing is a critical and difficult problem.

We describe a new method for estimating geometric information without using the extraction and tracing of characteristic elements. First, we store candidate colors for each voxel in a one-dimensional array using backward rays. We then narrow candidate colors into the final color using forward rays. In this way, we can obtain estimated volume data which includes color information.

1 はじめに

低次元の画像情報から高次元の幾何情報を推定する手法はこれまでも多くの報告がなされ今日にいたっている。代表的な手法としては、ステレオ画像から得られるエピポーラ面を利用した推定手法や、動画像から得られるオプティカルフローを利用した推定手法、カメラ位置の制限が少ない投票方式 (voting method) などが知られている。

従来の手法には数学的には完成度が高い手法が多いが、具体的な実装には困難を伴い、良好な結果を得にくいのが実情である [1]。その原因には次の要因があると考える。

- (1) 特徴点抽出の難しさ
- (2) 特徴点对応づけまたは追跡の難しさ
- (3) カメラキャリブレーションの難しさ
- (4) 精度の問題

従来の手法の多くは上記の全てにおいて問題を抱えている。一部の手法ではカメラの運動や背景色に制限を設けて上記の解決を試みる方法も存在するが、可用性が高いとはいえない。投票方式などは、キャリブレーションとは独立した命題設定をしているが、カメラ運動を制限しない点で高い評価を得ている手法である [2]。ただし、特徴点の抽出と対応づけを行う点で抜本的な解決には至っていない。また、物体の認知 (recognition) に関する研究も盛んであり、対象物の認知と同時にカメラキャリブレーションを行う報告などがある [3]。本報告ではカメラ運動や背景に大きな制限を与えず、特徴点の抽出と追跡を行わない汎用的なモデル推定手法について論じる。

2 推定のパラダイム

2.1 特徴点の抽出と追跡

従来からこの種の問題には特徴点 (feature points) の抽出と追跡を行う手法が用いられてきた。先述のとおり、本質的に難題を多く抱えているが今日の研究では特徴点の抽出法も多くのものが提案され、精度の問題も徐々に

解決されつつある。

また、モデル推定と同時に入力画像のカメラキャリブレーションも行う手法などが実画像を用いたシミュレーションで報告されている [4]。これらの手法でのモデル推定は特徴点の幾何情報を得ることに帰着する。面を構成して着色を行うためには、特徴点間を結んだポリゴンを生成することとなる。この場合、特徴点は相当量の数を必要とし、計算コストや精度の問題とともに、十分な特徴点を抽出できるかという問題にも直面する。特に比較的簡易な形状、比較的簡易なテクスチャをもつ物体であれば問題は顕著に現れる。実装の難しさとともに、対象物体にも考慮が必要となる。

2.2 ボクセル空間内でのボリューム推定

本手法では対象物体がボクセル空間内に存在するものとする。すなわち、各ボクセル値を推定することにより最終的な形状と色の推定を行う。よって、推定結果はボリュームデータとして得ることになる。ただし、各ボクセルに不透明度 (opacity value) をひとつずつ持たせる一般的なボリュームデータとは異なり、各ボクセルは存在の有無 (set/empty) を示す2値のパラメータと、有 (set) である場合はその色情報 (RGB値) を持つこととなる。各ボクセルのとりうる状態を表1に示す。対象物体をボリュームデータとして表現することにより、特徴点の抽出が不要となる。正確に述べるなら、特徴点の追跡とは異なるアプローチで物体形状の推定が可能となる。

	候補色数	候補色		
		R	G	B
Set Voxel(s)	$i (\geq 1)$	r_i	g_i	b_i
Empty Voxel(s)	0	-	-	-

表 1: set voxel と empty voxel

しかしながら、ポリウムデータ表現でも特有の欠点を持ち合わせる。本手法でも対象物体に考慮が必要となる。例えば、対象物体の幾何情報とボクセル空間密度の関係である。対象物の形状や表面色の変化がボクセル空間密度より大きければ問題は顕在化しないが、小さい場合は良好な推定結果が得られない。安易にボクセル空間密度を上げることは計算コストへのしわ寄せが大きい。

しかし、一定の条件を満たせば簡単なアルゴリズムで色情報を含んだ形状の推定が可能となる点で利用価値は大きいものとする。

3 推定のプロセス

3.1 概要

推定のプロセスは表 2 に示す通り 4 ステップに大別され、シーケンシャルな処理を行う。ここでは「各ボクセル毎の候補色」という考え方に着目し、候補色を絞り込む過程を通じて形状の推定を行う。

先述のとおり、各ボクセルは常に set または empty のいずれかの状態にある。ボクセルの状態は set から empty に遷移することはあっても、その反対には遷移しない。このように、処理過程の中で set voxel の候補色を失効させ empty voxel に遷移させることを empty 化と呼ぶことにする。

Forward 処理内の empty 化によって、empty voxels の数は増加し set voxels の数は減少する。この set voxels の減少分は、Backward 処理で過剰に（余分に）評価されている set voxels を指している。これらの過剰な set voxels を extra voxels と称すると、対象物形状を推定するということは、set voxels の中から extra voxels を抽出し、empty 化することに帰着する。

Forward(1) 処理終了時の set voxels が対象物外観形状を表す。

Forward(2) 処理で set voxels の候補色をひとつに絞ることにより、対象物形状と色情報を一意に推定できることとなる。

プロセス	概要
Pre-process	フレーム情報とカメラパラメータの展開
Backward	ボクセル候補色の一次元配列生成
Forward (1)	extra voxels のempty化
Forward (2)	set voxels の色の確定

表 2: 処理ステップ

3.1.1 バックワードレイ処理

本ステップでは、バックワードレイを用いて、各ボクセルの候補色（複数）を推定する。バックワードレイとは、「各ボクセル」から「各視点」へ放たれるレイをいう。この時、レイがスクリーン（投影面）と交われば、そのピクセル色を獲得できる。このピクセル色はそのボクセルがもつ固有の色である可能性がある。すなわち、候補色であるといえる。（図 1 参照）このバックワードレイを全ての視点に対して発することにより、ボクセル色候補を一次元配列として獲得することができる。

しかし、実際の画像では、図 2 に示す通りの遮蔽や背面投影が存在する。よって、この一次元配列中の一部に正規のボクセル色が含まれている、と考えるのが妥当である。必ずしも一次元配列中の最多頻出色がボクセル色になるとは限らない点に注意が必要である。本ステップでは、この一次元配列から候補色を選定している。

まず、正面投影と背面投影を極力分離するためにレイを 6 方向に分類する。各軸 (x, y, z) に沿って正負 $(+, -)$ の計 6 方向である。凸表面上の一点は立体角 180 度で可視状態であることにより、概ね正面からのレイと、背面からのレイに分離することができる。ただし、正面と背面の区別はできないため、この段階では双方から候補色を選出する。この処理を各軸 (x, y, z) について行う。

レイの分類は方向ベクトルの内積によって行う。各軸平行の単位ベクトルを $\vec{direction}_i (i \leq$

6) として、Ray の方向ベクトル \vec{ray} とすると、ある方向 $direction_i$ について

$$\{(\vec{direction}_i, \vec{ray}_j) \geq 0 \mid j \leq frame_num\}$$

が、一定フレーム以上連続して成り立ち、一定フレーム以上の最多連続色が存在する場合、その方向でのボクセル候補色とする。

ただし、一定値以上の連続色が無い場合、その方向では候補色が収束しないことになる。また、異なる方向で同じ色に収束する場合もあるので、6色以内の候補色を持つことになる。6方向全てが収束しない場合（候補色数が0個）は、この時点で empty voxel と推定される。

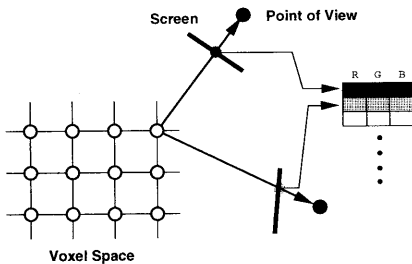


図 1: バックワードレイ

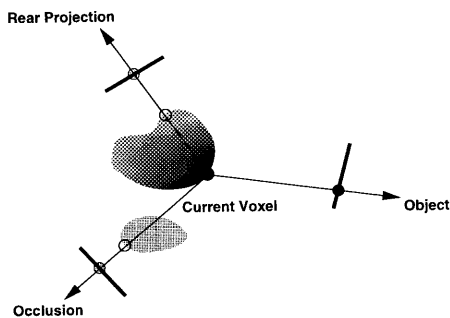


図 2: 遮蔽と背面投影

3.1.2 フォワードレイ処理 (1)

本ステップでは、フォワードレイを用いて、各ボクセルの存在有無の検証を行う。フォワードレイでは、「各視点」から「各ピクセル」を経由してボクセル空間にレイを発生させる。(図 3 参照)

この時、経由するフレームのピクセル色 $C(pixel)$ と、通過するボクセルの候補色 $C(d_i)$ を比較する。ピクセル色 (一色) と候補色 (複数) に同一色がなければその候補色は全て誤りであると推定できる。すなわち、そのボクセルは empty voxel とみなすことができる。換言すれば、候補色数を d_{num} として、

$$(\forall C(d_i) \mid d_i \leq d_{num})(C(d_i) \neq C(pixel))$$

が成り立てば、そのボクセルは empty であると推定できる。

上述のバックワードレイ処理が終了した時点では、対象物の周辺も候補色を持っており、贅肉 (extra voxel) の多い対象物形状が推定されている。図 4 に示すとおり、extra voxel を通過し背景に至るレイにおいて extra voxel の empty 化が可能となる。全スクリーンの全ピクセルについてこの処理を行うと、最終的な対象物形状が浮き彫りになる。

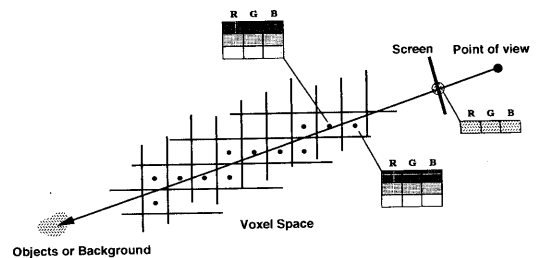


図 3: フォワードレイ / 3DDDA

ただし、実際に empty voxel を推定するにあたっては若干の注意が必要である。フォー

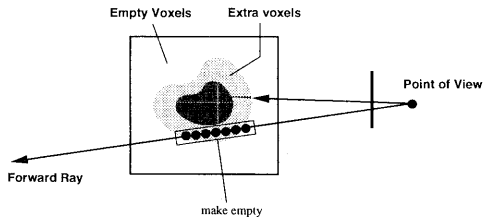


図 4: extra voxels の empty 化

ドレイは 3 D D D A (3 Dimensional Digital Differential Analyzer) を用いている。すなわちボクセル空間は疎であれ密であれ空間的な大きさを持っているという点である。安易な empty 化が傷口を広げ、後の推定結果に多大な影響を与えることがありえる。この典型的な例を図 5 に示す。

図 5 において、任意のボクセルに Ray1 と Ray2 が通過する場合、Ray1 が背景にあたるレイであるとそのボクセルは empty 化されてしまう。この後に、物体表面色をもつ Ray2 がこのボクセルを通過するとき、既に empty となっているとそのまま通過し、他の対象物内部を侵食する結果となる。特に 3 次元空間で比較的小さな物体を扱う時に発生しやすい。これを回避するためには、比較的都度 empty 化を行うのではなく、empty となるボクセルに対してフラグを立てておき(投票しておく)本ステップ終了時に一括して empty 化を行えばよい。これにより、Ray2 は該当ボクセル上での比較において同一色があるとみなしその後の侵食を行わない。

また、扱うカラー画像にもよるが、レンダリング画像などを用いる場合、アンチエイリアシングなどが施されている場合が多い。すなわち、対象物内の色の境界では中間色をもってピクセル色としてしてある場合が多い。この場合、この中間色をピクセル色として、ボクセル候補色との比較を行うと、そのレイの通過ボクセルはことごとく empty になってしまう。これを回避するために、該当するピク

セル色に 8 近傍のピクセル色を含め色の許容色数を広げた。アンチエイリアシングは物体境界の 1~2 ピクセルで処理される事象であるため、8 近傍色を許容することにより回避が可能となる。(図 6 参照)

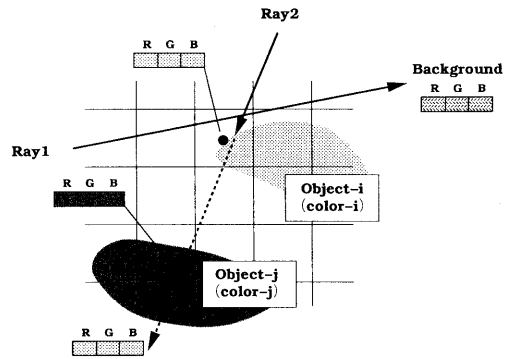


図 5: empty の一括化

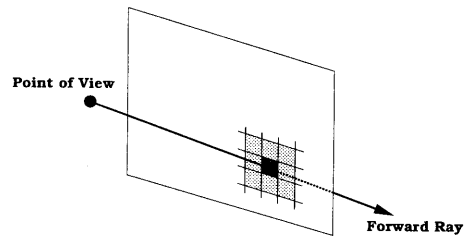


図 6: 8 近傍ピクセル色

3.1.3 フォワードレイ処理 (2)

最終段階となる本ステップでは、フォワードレイを用いて、各ボクセルの候補色をひとつに絞り込む。前段階まででは empty voxel の抽出により、対象物形状は推定できているが、各ボクセルの候補色は一意に求まっていない。このボクセル空間に対して、前ステップと同様に 3 D D D A を用いてレイを通過させ、ピクセル色と候補色の比較を行う。ピクセル色と同色となる候補色があった時点でその候

補色に対して投票を行い、ボクセル探索を中断し、次のレイ処理へ移る。すなわち、対象物体表面に達した時点で投票を行い、物体内部にまでは探索を行わない。このことにより、最終的な候補色1色を推定するとともに、得票のない対象物内部の set voxel を empty 化することができる。結果的に対象物表面のみを抽出した色情報付きのボリュームデータが生成されることになる。

対象物内部を empty 化することは、データ量の大幅な削減が期待できる上に、ポリゴン生成、幾何情報の生成ステップにつながるものと考えられる。

4 実験結果

4.1 推定モデル

本手法についての実験を SGI ONYX (R4400, 256MB) 上にて行った。連続カラー画像は、一般的な CG ツールを用いて作成した。扱う画像は $(320 \times 240 \text{ ピクセル}) \times 50$ 枚とし、ボクセル空間を $32^3, 64^3, 128^3$ として実験を行った。

ここでは、2つの実験結果について報告する。それぞれを、Glass data と Sphere data と称する。

図7に Glass data の連続画像の一部を、図10にその推定結果を示している。ボリューム形式で得られる推定モデルは、set voxel の存在する位置に色付きの小球を配置して表現している。



図 7: 入力フレーム -Glass data-

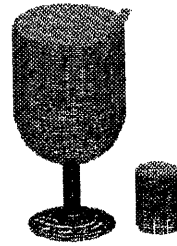


図 8: モデルの推定結果 -Glass data-

また、カメラパラメータや画像枚数をそのままとし、遮蔽の発生を多くした実験モデルが、図9, 10に示す Sphere data である。それぞれのモデルにおいて、良好な推定結果が確認できたと考える。

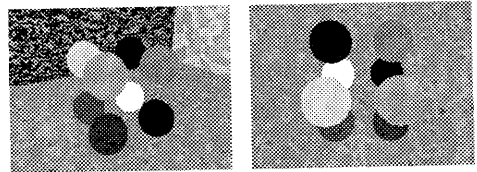


図 9: 入力フレーム -Sphere data-

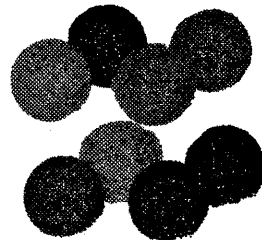


図 10: モデルの推定結果 -Sphere data-

図 11 には、Glass data の推定モデルを Z 軸方向中央で分割した断面図を示している。フォワードレイ (1) 終了時と、フォワードレイ (2) 終了時の比較により、対象物内部のボクセルが empty 化され、表面情報のみが推定されていることが確認できる。

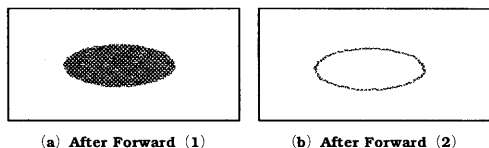


図 11: Glass data の断面形状の変化

4.2 計算コスト

表 3、4 にはボクセル空間密度毎の各ステップ終了時点での empty voxel の数を示している。各ステップを通じて empty voxel が増加する過程を示している。最下段の REMAIN とは、最終的に set を維持したボクセルの数を示している。

	32 ³	64 ³	128 ³
Backward	8,460	69,998	539,836
Forward (1)	31,181	245,816	1,955,894
Forward (2)	32,099	258,785	2,082,064
REMAIN	669	3,359	15,088

表 3: voxel 数の変化 -Glass data-

表 5、6 にボクセル空間密度を変化させた場合の、各ステップでの CPU 利用時間を示す。この中で、Backward に要する時間は、ほぼ正確にボクセル数の変化に比例している。モデルの形状や色、状態等には依存しない。ボクセル数とピクセル数にのみ依存する。すなわち、ボクセル空間密度が、2³倍になれば、Backward 処理時間も 2³倍になっている。

	32 ³	64 ³	128 ³
Backward	4,468	36,134	289,327
Forward (1)	29,733	231,248	1,819,109
Forward (2)	31,355	254,932	2,065,477
REMAIN	1,413	7,212	31,705

表 4: voxel 数の変化 -Sphere data-

一方、Forward に要する計算時間は、ボクセル数とは比例関係をもたない。ここで比例項目として考えられるのは 3 D D D A での探索ボクセル数である。この探索ボクセル数は、ボクセル空間密度が 2³倍になっても、2 倍強にしか増えない。その上、同一色判定処理の回数や探索の途中終了頻度などが、その時のボクセルの状態に依存するため計算時間にもぶれが生じる。実際に本例においても、Forward 処理はボクセル空間密度の 2³倍の変化に対して、1.81 ~ 2.28 倍の間を推移している。この推移を図 12 に示す。

よって、ボクセル空間密度の変化に対して、Backward 処理は常数比例に、Forward 処理はその $\frac{1}{3}$ 乗根 (各軸方向あたりの変化量) に近い値に従ってコストが変化するものと推定できる。

	32 ³	64 ³	128 ³
Pre-process	5.7	6.1	9.0
Backward	30.4	244.1	1,929.8
Forward (1)	303.0	611.5	1,416.6
Forward (2)	146.9	286.6	655.3
TOTAL	486.0	1148.3	4,010.7
	seconds		
	minutes	8.1	19.1
			66.8

表 5: CPU 使用時間 -Glass data-

5 今後の課題

本研究では、背景色やカメラ運動に制限を与えない方向でのモデル推定法について論じてきた。しかし、実験に用いたデータは計算

		32 ³	64 ³	128 ³
Pre-process		5.6	5.9	8.5
Backward		30.3	241.4	1,907.6
Forward (1)		306.7	594.6	1,310.0
Forward (2)		141.6	256.0	580.2
TOTAL	seconds	484.2	1097.9	3806.3
	minutes	8.1	18.3	63.4

表 6: CPU使用時間 -Sphere data-

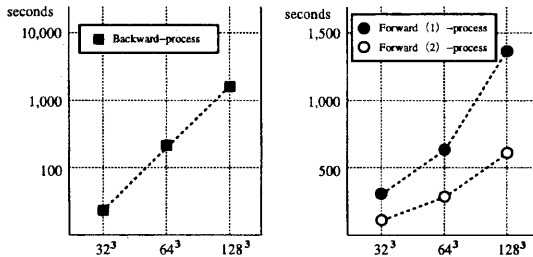


図 12: CPU使用時間の変化 -Sphere data-

機上で生成した画像でありカメラパラメータは正確に求まっていることを前提としている。また、対象物データは原色系のものを用いており、ハイライトや陰影など、自然界に見られる現象を十分に反映しているとは言いがたい。今後はより自然界に近い環境下での実装を検討したい。

また、可用性を高めるためには処理速度の向上も必須であると考えており、並列化などの処理向上策も模索していきたい。

6 むすび

本報告では、対象物体がボクセル空間内に存在すると仮定して、ボリュームデータとしてのモデル推定手法について述べた。カメラパラメータが既知であり、陰影などが少ない場合についての実験を行い、良好な推定結果を得ることができた。

従来手法と比べてカメラ位置の自由度の高い

画像を用いて、従来手法では少例であった色情報を含んだモデルの推定が行えることを確認した。同時に、従来から難題とされている遮蔽問題に対しても比較的口バストである点を確認した。

特徴点の追跡によるモデル推定と、ボリューム空間によるモデル推定は、一概に優劣はつけられないと考える。これらはモデルの種類や状態によって適、不適が決まる。

両者のアプローチが今後も必要であることを認識しつつ、我々はボクセル空間での推定手法についてその可能性を探りたいと考えている。

参考文献

- [1] 出口：'コンピュータビジョンのための幾何学'，情報処理 37(6),pp549-556(1996)
- [2] 浜野他：'空間への voting による 3次元環境情報抽出手法'，信学論 (D-II),J75-D-II,pp.342-350
- [3] H.Murase and S.K.Nayer : Visual Learning and Recognition of 3-D Objects from Appearance, International Journal of Computer Vision, Vol.14,pp.5-24,1995
- [4] C.Tomasi and T.Kanade:Shape and Motion from Image Streams under Orthography : a Factorization Method,International Journal of Computer Vision, Vol.9, No2,pp137-189,1992