

多視点画像を用いた誤り検出・訂正機能を持つ高精度対象領域抽出法

大浜 郁[†] 延原 章平[†] 松山 隆司[†]

[†] 京都大学情報学研究科 〒 606-8501 京都市左京区吉田本町

E-mail: †{ohama,nob,tm}@vision.kuee.kyoto-u.ac.jp

あらまし 本論文では対象を複数の異なる視点から撮影した多視点実画像からの対象領域抽出手法を提案する。従来の対象領域抽出問題の多くは単視点情報のみを想定していたが、我々は異なる視点から共通の対象を撮影していることを利用し、複数の視点から共通の対象を撮影したことによって生じる視点間の制約条件、つまり各視点の対象領域は共通の3次元形状を投影した結果であり互いに矛盾しないという制約条件を導入する。これに各視点における制約条件として対象領域は撮影画像を明度に基づいて分割して得られる小領域の集合として表され、対象領域の輪郭は画像上でエッジとして表れるという制約を同時に使用することで誤り検出・訂正機能を実現し、ピクセルやボクセルといった孤立点に対する処理を繰り返すのではなく、ある一定の大きさを持った領域単位の頑健な処理アルゴリズム考案するとともに、実験によってその有効性を検証した。

キーワード 対象領域抽出, 多視点画像, 視体積交差法

Accurate Object Extraction With Error Detection And Correction Using Multi-Viewpoint Images

Iku OHAMA[†], Shouhei NOBUHARA[†], and Takashi MATSUYAMA[†]

[†] Graduate School of Informatics, Kyoto University Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501 Japan

E-mail: †{ohama,nob,tm}@vision.kuee.kyoto-u.ac.jp

Abstract In this paper, we propose an algorithm to extract object silhouettes from multi-viewpoint images based on “intersection” and “projection” constraints. The intersection constraint implies that each silhouettes should be consistent with each other since they are the 2D images of the 3D object shape. The projection constraint implies that each silhouette consists of regions which are given by segmentation based on the texture information, and each contour of silhouettes should be consistent with the edge of the captured image. Based on these two constraints which should be satisfied by multi-viewpoint silhouettes, we introduce region-oriented operation which preserves apparent connectivity and continuity of the object on each image plane, and propose a method which can explicitly detect and correct errors of silhouette extraction. We show some experimental results to demonstrate the advantages of our method.

Key words Object silhouette extraction, Multi-viewpoint images, Shape from silhouette

1. はじめに

本論文では、対象を複数の視点から同時に撮影した多視点画像から、対象領域を抽出する新たな手法を提案する。提案手法のポイントは、(1) 複数の視点間で対象領域が満たすべき制約条件を導入することで、対象と背景の色が近い場合など、単視点での対象領域抽出では抽出が困難であるような場合においても抽出を可能としたこと、(2) 対象領域抽出の誤り検出および訂正を実現したことである。

そもそも画像からの対象領域抽出という問題はコンピュータビジョンの分野で古くから取り組まれているが、いまだ解決に

至っていない。これまでに提案された手法は“対象とは何か”をいかに定義するかという点で分類でき、たとえば背景差分法では対象領域は“対象が存在しなかった場合との差領域”と定義され、SNAKES [1] では“画像の明度勾配が大きい部分を滑らかに繋いだ領域”とされた。しかしながら、ある視点から対象を撮影しその画像中から対象領域を抽出する限り、例えば背景と対象の色が非常に近く区別がつかない場合など、本質的に解決が困難な状況が存在する。

一方、近年の3次元ビデオに関する研究では視体積交差法の使用が広まっている [2]。これは視体積交差法によって複数視点での対象領域情報から一意に対象の概形を得られるからである

が、このような“多視点画像からの対象領域抽出”という状況では、対象領域に関して単視点の場合に加えてさらなる制約条件を課することができる。すなわち、各視点の対象領域はある1つの対象の3次元形状を投影した結果であり、互いに矛盾しないということである。そこで本論文では対象領域を以下のように定義する。(1) 各視点において、対象領域の輪郭は撮影画像のエッジと一致する。言い替えると、対象領域は画像上で見かけ上の連結性を保つ。(2) 視点間において、各対象領域はある1つの3次元形状を投影した結果となっており、互いに矛盾しない。本研究ではこれらの制約条件と、従来の単視点における対象領域抽出手法の結果を使用して誤りの検出・訂正を実現した。このことから本手法は、従来の単視点における対象領域抽出手法に多視点画像間の制約条件を統合することで、誤り検出・訂正機能を持った対象領域抽出を実現したものであるといえる。

以降、第2.節で多視点画像からの対象領域抽出を扱った関連研究に対する本手法の位置づけを述べ、第3.節で上記の対象領域の定義に基づいた領域抽出アルゴリズムを提案する。第4.節で実験によって提案手法の有効性を示し、第5.節で考察と今後の課題について述べる。

2. 関連研究

多視点画像からの対象領域抽出という問題に対して [3] では本研究と同様に、対象領域の見かけ上の連結性と視点間での制約を導入したアルゴリズムが提案されている。しかし誤りの検出に関する処理がなされていないため、対象と背景の色が近い場合に対象領域の見かけ上の連結性の判断を1箇所でも誤ると、処理が破綻するという欠点が存在する。これに対して提案手法では、このような場合に関しても誤り検出・訂正処理を導入することでこの問題の解決を図っている。

一方、多視点対象領域の抽出は対象の visual hull を復元することと等価であるという観点から [4] [5] [6] のように視体積交差法を基本としたアルゴリズムが提案されている。しかしこれらの手法では visual hull をボクセル集合として表し、ボクセルを基本処理単位とするために対象形状の連続性に基づいた頑健な処理を実現できていない。これに対して本研究では、基本処理対象を各視点における連結領域を単位とすることで、対象領域の連結性を利用し、さらに連結領域の見かけ上の形状が視点ごとに異なることを利用して、誤り検出・訂正を実現し、より頑健なアルゴリズムとなっている。

3. 多視点画像を用いた誤り検出・訂正機能を持つ対象領域抽出法

3.1 領域抽出処理モデル

本論文では、十分な精度でキャリブレーションされていて、多視点撮影が可能な環境を想定し、一般的な実環境において、(1) 対象領域の連結性、(2) 多視点間で満たされるべき制約条件、(3) 誤りの検出・訂正の3点を考慮した領域抽出処理を行う。図1に本論文における領域抽出処理の概念図を示す。提案手法は、各視点の初期シルエットを入力として、これを(1) 各視点

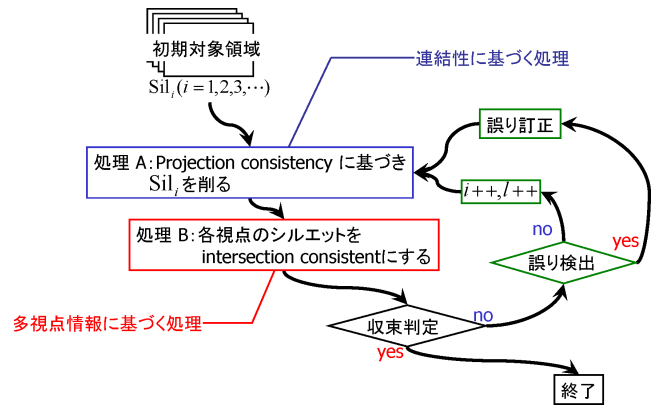


図1 領域抽出概念図

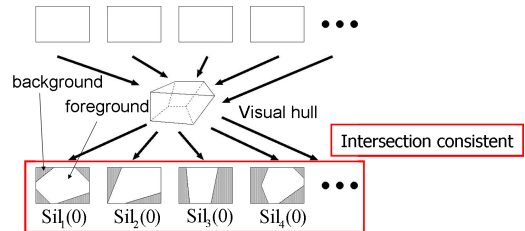


図2 初期対象領域

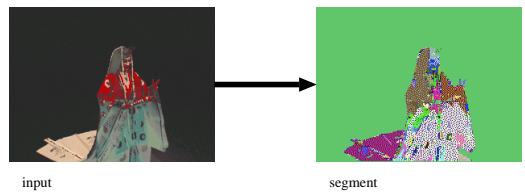


図3 画像の領域分割

における見かけ上の連結性に基づく制約条件（以降 projection 制約と呼ぶ）と (2) 視点間の整合性に基づく制約条件（以降 intersection 制約と呼ぶ）を同時に満たすように交互に更新しながら、最終的に両方の制約を満たす多視点シルエットを出力する。ここで2つの制約条件は以下のように定義する。

Projection 制約: 各視点において、対象領域の輪郭は撮影画像のエッジと一致する。

Intersection 制約: 各視点の対象領域は、ある一つの3次元形状を投影した結果であり、互いに矛盾しない。

以降では、各ステップについて詳細を説明し、提案手法が求める対象領域の定義を明らかにした後、誤り検出・訂正の必要性とその方法を述べる。

なお以降では説明のため下記の記号を使用する。

視点 i n 個の視点のうちのある1つの視点。

ステップ l 反復処理の l 回目のステップ。

対象領域 Sil_i 視点 i の対象領域（シルエット）。

領域分割された対象画像 Seg_i 視点 i の撮影画像を領域分割したものの。

visual hull VH 対象の visual hull。

投影像 VH_i VH の視点 i への投影像。

なおこれらの記号に (l) を加えて、 $Sil_i(l)$ などとした場合、ステップ l の時点での Sil_i を意味するものとする。

3.2 初期状態の定義

十分に対象領域を含むとみなせる各視点の画像領域、言い替え

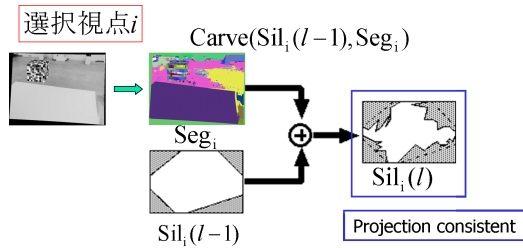


図 4 Projection 制約に基づく対象領域の更新

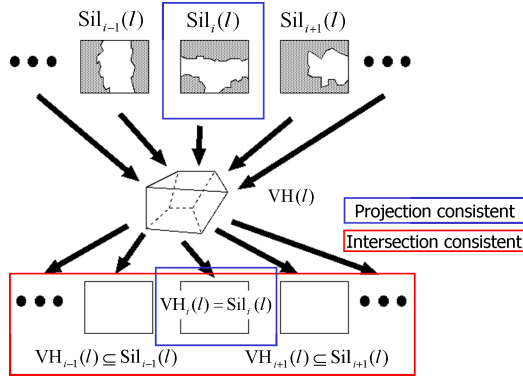


図 5 Intersection 制約に基づく対象領域の更新

ると画像領域全体を対象領域として visual hull を計算し、これを各視点に投影したものを初期対象領域 $Sil_i(0)$, $i = 1, \dots, n$ と定義する．この時点で各視点における対象領域は，intersection 制約を満たしている（図 2）．

また各視点における画像を，明度に基づいて領域分割したものを Seg_i , $i = 1, \dots, n$ と定義する（図 3）．領域分割に際しては何らかの閾値が必要となるが，後述するように誤り検出の際に領域分割に不具合があれば閾値を変更して再分割するので，ここでは領域がある程度大きくなるような閾値で適当に分割するものとする．

3.3 Projection 制約に基づく対象領域の更新

図 1 の処理 A の開始時点では，各視点の対象領域は intersection 制約だけを満たしている，言い替えると対象領域がその時点での visual hull の投影像となっている．このような場合，一般にはその輪郭線が各視点のエッジに一致しているとは限らず，projection 制約は満たされていない．そこでステップ l ではある視点 i だけに注目し，この視点の対象領域を projection 制約を満たすように以下の手順で更新する．

まず Seg_i を構成する各小領域に関して，

Case A: 小領域全体が完全に $Sil_i(l-1)$ に含まれているか，

Case B: 一部でも $Sil_i(l-1)$ の外の領域を含んでいるか，

をチェックする．その結果 Case A である小領域集合をもって，新たな対象領域 $Sil_i(l)$ とする．言い替えると $Sil_i(l-1)$ から Case B となるような領域を削除する．この処理を $Sil_i(l) = \text{Carve}(Sil_i(l-1), Seg_i)$ また， i 以外の視点に関しては変更をせず， $Sil_j(l) = Sil_j(l-1)$, $j = 1, \dots, i-1, i+1, \dots, n$ とする（図 4）．

こうしてから視点 i の対象領域 $Sil_i(l)$ は projection 制約を満たすように更新された．

3.4 Intersection 制約に基づく対象領域の更新

図 1 の処理 B の開始時点では， i 以外の視点の対象領域は

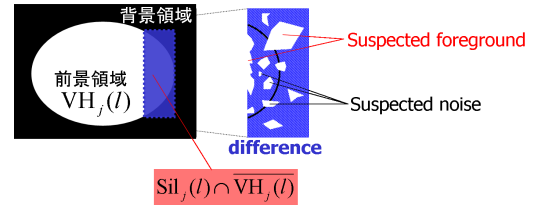


図 6 誤り検出アルゴリズム

intersection 制約だけを満たしており，視点 i の対象領域は projection 制約を満たしている．ここで各視点の対象領域を用いて $VH(l)$ を計算し，それを各視点に投影すると，明らかに視点 i に関しては $VH(l)$ の投影像 $VH_i(l)$ と $Sil_i(l)$ が一致し，それ以外については投影像は一致するかより小さくなり， $VH_j(l) \subseteq Sil_j(l)$ となる（図 5）． i 以外の視点に関して，この $VH_j(l)$ を対象領域候補とみなして，次の収束判定，誤り検出へと進む．

3.5 収束判定

さて図 1 の処理 B が終わった時点で，すべての視点で $Sil_i(l)$ と $VH_i(l)$ が等しくなったならば，各視点における対象領域 $Sil_i(l)$ は intersection 制約と projection 制約を同時に満たしているといえる．このとき $Sil_i(l)$ を最終的な対象領域として，アルゴリズムは終了する．一方いくつかの視点で $VH_j(l) \subset Sil_j(l)$ となったとき，次の誤り検出へと進む．

3.6 誤り検出

前述の収束判定で $VH_j(l) \subset Sil_j(l)$ となったときとは，すなわち projection 制約を満たすように $Sil_i(l)$ を更新すると今度は intersection 制約が満たされていないことが判明したという場合である．このとき [3] の手法では常に projection 制約に基づく更新を正しいものと仮定し， $VH_j(l)$ を新たな $Sil_j(l)$ とした．しかしこれではそもそも projection 制約に基づく対象領域の更新が正しかったか，言い換えると視点 i の領域分割が妥当であったかどうかを検証されていない．そこで本研究では， $VH_j(l)$ を新たな $Sil_j(l)$ とするにあたって削られる領域，つまり $\overline{VH_j(l)} \cap Sil_j(l)$ なる領域を i 以外の各視点 j において対象領域から削ることが妥当かどうかをチェックすることで，視点 i の領域分割と projection 制約による $Sil_i(l)$ の更新が妥当であったかどうかを判断する．

この妥当性のチェックを以下 $\text{IsValid}(Sil_j(l), VH_j(l))$ と呼ぶこととする．この判定は各視点において，対象領域が削られてなくなった領域 $Sil_j(l) \cap \overline{VH_j(l)}$ に関して，該当する背景差分画像上の領域を参照し，前景らしさの評価及び，周囲との連結性を判断することで行う．ここで前景らしさの評価には単視点における対象領域抽出の結果を使用する．具体的にはまず $Sil_j(l) \cap \overline{VH_j(l)}$ に相当する背景差分画像の領域を二値化して対象領域 Sil_{carved} を得る．そして現在の対象領域 $Sil_j(l)$ との和集合 $Sil_{\text{sum}} (= Sil_j \cup Sil_{\text{carved}})$ を得る．この対象領域 Sil_{sum} に対してノイズ除去を行い，その結果 Sil_{carved} の領域が完全になくなれば， $\text{IsValid}(Sil_n(l), VH_j(l))$ は真を返すものと定義する．ノイズ除去の結果 Sil_{carved} の領域がなくならない場合は，その領域にノイズ除去で取り除けない大きさを持った領域が存

一定視点以上で誤り訂正の要求があったとき、誤りが検出された視点 j について、

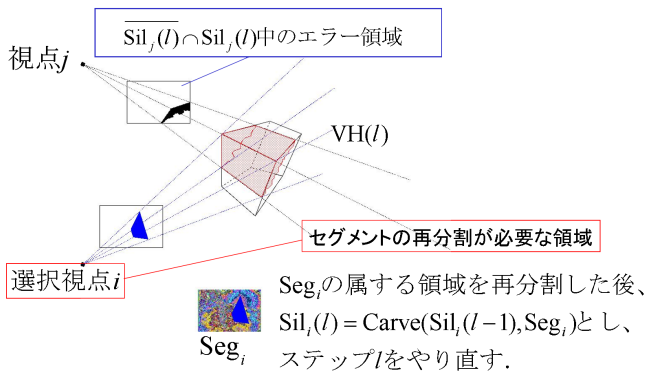


図 7 誤り訂正アルゴリズム

在するか、または、 $\text{Sil}_{\text{carved}}$ 上の対象領域が Sil_j と連結していることが考えられるので、今回削られた領域 $\text{Sil}_{\text{carved}}$ は前景の一部だったと判断する (図 6)。

このように、本手法では従来の単視点における対象領域抽出の結果を使用して誤り検出を行っている。このことから本手法は従来の単視点における手法を置き換えるものではなく、それらを多視点画像間の制約条件を用いた誤り検出・訂正の枠組へと統合するものであるといえる。なお本研究では単純な背景差分法を使用したがる、この部分は他のより高度なアルゴリズムと交換可能である。

3.7 誤り訂正

ステップ l において $\text{IsValid}(\text{Sil}_j(l), \text{VH}_j(l))$ が偽となる視点 j があるとしたら、それはその直前の処理 A の projection 制約に基づく視点 i の対象領域の更新が原因であることは定義から明らかである。そこでこのような視点 j の情報を元に視点 i の対象領域 $\text{Sil}_i(l)$ を更新し直すことで、誤り訂正を実現する。具体的には、そもそも処理 A における $\text{Sil}_i(l)$ の更新は、 Seg_i の領域分割に基づいて行われていたことから、この領域分割がそもそも誤っていた、言い替えると第 3.3 節で Case B として削ってしまった小領域が真の対象領域を含んでしまっていたことを意味している。したがって視点 j の情報を元に視点 i のこのような小領域をより細かい領域へと再分割し、再び処理 A からやり直せば、今度は視点 j で $\text{IsValid}(\text{Sil}_j(l), \text{VH}_j(l))$ となることが期待できる。上記のような小領域を以下のようにして特定する。

- Step 1. まずエラーとなっている部分は、 $\text{VH}_j(l)$ が $\text{Sil}_j(l)$ よりも削れている領域、すなわち $\overline{\text{VH}_j(l)} \cap \overline{\text{Sil}_j(l)}$ という部分に含まれる。
- Step 2. そこでこの領域によって作られる visual cone と、 $\text{VH}(l-1)$ の積領域を得る。
- Step 3. この積領域を直前の処理 A で選択されていた視点 i に投影し、その投影先を含む Seg_i の領域を再分割する。こうして再分割の結果を反映した Seg_i を用いて、ステップ l を処理 A からやり直す。つまり $\text{Sil}_i(l) = \text{Carve}(\text{Sil}_i(l-1), \text{Seg}_i)$ とし、処理 B, 収束判定と進む。

3.8 アルゴリズム

以上をまとめると、先に図 1 に示した提案手法のアルゴリズム

は以下ようになる。

- Step 0 各視点の初期対象領域 $\text{Sil}_i(0)$ および領域分割 Seg_i を作る。また $i = 1, l = 0$ とする (3.2 節)。
- Step l 視点 i を選択しているとして、
 - Step $l.1$ $\text{Sil}_i(l) = \text{Carve}(\text{Sil}_i(l-1), \text{Seg}_i)$ とする。そのほかの視点については、 $\text{Sil}_j(l) = \text{Sil}_j(l)$ とする (3.3 節, 処理 A)。
 - Step $l.2$ 全視点の対象領域を用いて、visual hull $\text{VH}(l)$ をつくり、この $\text{VH}(l)$ の i 以外の視点における投影像 $\text{VH}_j(l)$ を得る (3.4 節, 処理 B)。
 - Step $l.3$ i 以外の全ての視点において、 $\text{Sil}_j(l)$ と $\text{VH}_j(l)$ が等しいならば、このときの対象領域を最終出力として終了。そうでなければ次のステップへ (3.5 節)。
 - Step $l.4$ $\text{Sil}_j(l) \subset \text{VH}_j(l)$ であることから、その差分 $\text{Sil}_j(l) \cap \overline{\text{VH}_j(l)}$ を削ることが妥当かどうかを判定する (3.6 節)。妥当であるならば Step $l.5a$ へ、そうでないならば Step $l.5b$ へ進む。
 - Step $l.5a$ i 以外の視点に関して、 $\text{Sil}_j(l) = \text{VH}_j(l)$ とし、全視点で intersection 制約が満たされる状態にした後、次の視点 $i+1$ を選択し、ステップ $l+1$ へと進む。
 - Step $l.5b$ $\text{IsValid}(\text{Sil}_j(l), \text{VH}_j(l))$ が偽となる視点 j の情報を元に、視点 i の領域分割 Seg_i の該当小領域を再分割し、再度 Step l をやり直す (3.7 節)。

3.9 計算手法

提案手法では各視点の対象領域を intersection 制約に基づき visual hull の投影像として求めている。このため visual hull をボクセル集合として離散的に定義してしまうと、各視点の対象領域画像の解像度との不整合から計算誤差の発生が避けられず、またボクセル空間を画像解像度以上に十分細かなサンプリング解像度で用意することは計算量・必要メモリ量双方の観点から今日の計算機で実現することは現実的ではない。

しかしそもそも提案手法で必要としているのは 3 次元形状としての visual hull 自身ではなく、常にその各視点における投影像だけであることに注目すると、“Image-Based Visual Hulls” [7] のように明示的に visual hull を定義せず、多視点シルエットから直接 visual hull の投影像を求める手法を使用すれば、計算量・メモリ量ともに解決可能である。またこの手法はエピポーラ線に沿った画像の探索を繰り返すので、ステレオ法などと同様に正規化することによる高速化が期待できる。特に本手法の場合は、対象を囲むように撮影する多視点画像を使用することから、カメラが互いに向かい合い、エピポールが画像内に存在する可能性がある。このような状況にも適用可能なものとして、我々は極座標系への正規化を行う手法 [8] を使用した。

3.10 アルゴリズムの停止性

提案したアルゴリズムでは、各ステップにおける誤り検出・訂正を一切行わなかった場合、各視点で最初から正しい領域分割が得られていれば、各視点の対象領域は正しい対象領域へ収束するように削られていくが、途中で偶然 projection 制約と intersection 制約が成立して終了する [3]。また領域分割が正しい

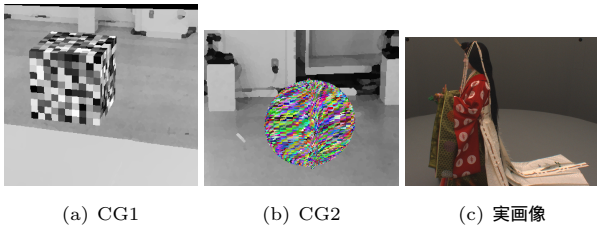
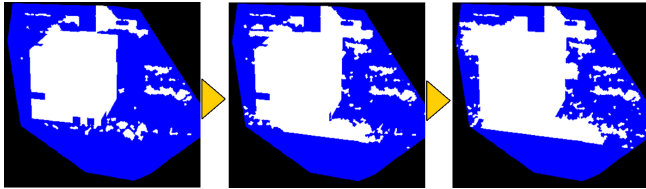
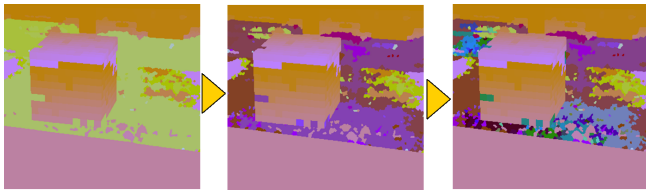


図 8 入力画像



(a) 対象領域の変遷



(b) 領域分割の変遷

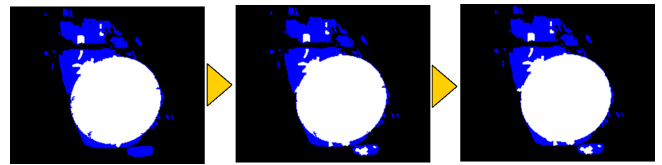
図 9 エラー訂正と領域分割の修正 (CG1)

くない場合には、削る操作が繰り返されて、遂には対象領域がなくなってしまう。つまりこのアルゴリズムは、projection 制約と intersection 制約を満たす対象領域を出力するか、全く出力しないかのいずれかである。このことは各視点における対象領域への操作が「削る」という種類しか存在しないことから自明といえる。一方、どこかで誤りの検出が起きた場合、対象領域は「削った分を戻す」という操作を受けるが、このとき必ず、再度セグメントの更新を行い、必ずより小さい領域へと分割する。したがって、この領域分割がピクセルというある決まった最小単位で停止することを考えると、最終的には対象領域の増減はピクセル単位以下になる。このピクセル単位以下の増減は離散化された計算上の不安定性と見なせるので、やはり停止（収束）しているといえる。つまり本アルゴリズムは、全ての場合において projection 制約と intersection 制約を同時に満たす対象領域を出力して収束、停止するか、全く何も出力せずに終了するといえる。

4. 実験

ここまでに提案したアルゴリズムを用いて、3種類の多視点画像（図 8）に対して領域抽出を行い、本手法の有効性を検証した。3種類の多視点画像のうち図 8(a),(b) は実画像背景に、それぞれ立方体、曲面で構成された仮想物体を投影して作成した画像（各 25 視点）で、図 8(c) は実画像を多視点で撮影したもの（14 視点）である。

それぞれの多視点画像に対し提案手法を適用した際に、誤り検出・訂正が行われる様子を図 9(a), 10(a), 11(a) に示す。各画像は図 1 の処理 A の前後のシルエットの変化を表しており、青色で表されている領域は処理 A により削られたことを意味する。また、領域分割の状態の遷移を図 9(b), 10(b), 11(b) に



(a) 対象領域の変遷

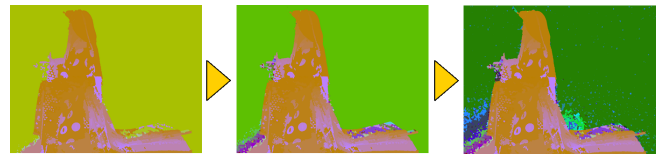


(b) 領域分割の変遷

図 10 エラー訂正と領域分割の修正 (CG2)



(a) 対象領域の変遷



(b) 領域分割の変遷

図 11 エラー訂正と領域分割の修正 (実画像)

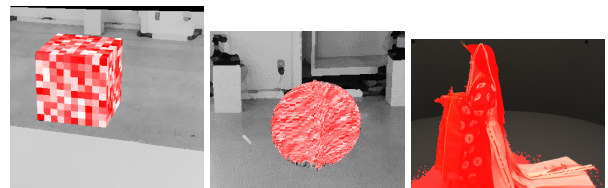


図 12 領域抽出結果 (赤色: 対象領域)

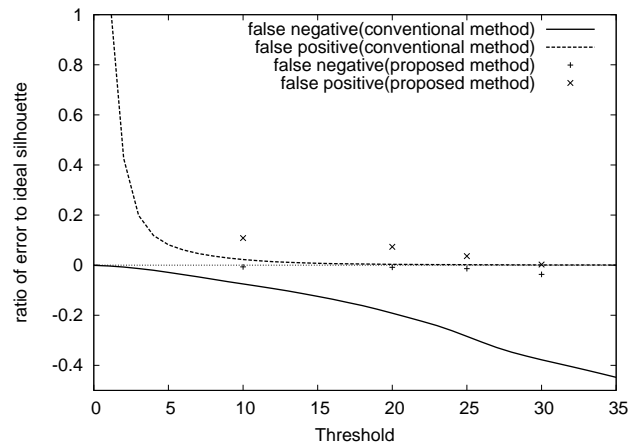


図 13 従来手法との比較 (正解に影を含まない)

示す。画像中色が変化している領域は、誤りが検出されて再分割された部分を表しており、あるステップでの対象領域に欠損が含まれているとき、誤り訂正が働いて該当領域が再分割され、欠損部分が修復されていくのが分かる。各多視点画像それぞれについて反復約 350 回、250 回、400 回で収束し、計算には Xeon 3.6GHz の PC を使用して約 0.5 日、1.0 日、2.0 日の処

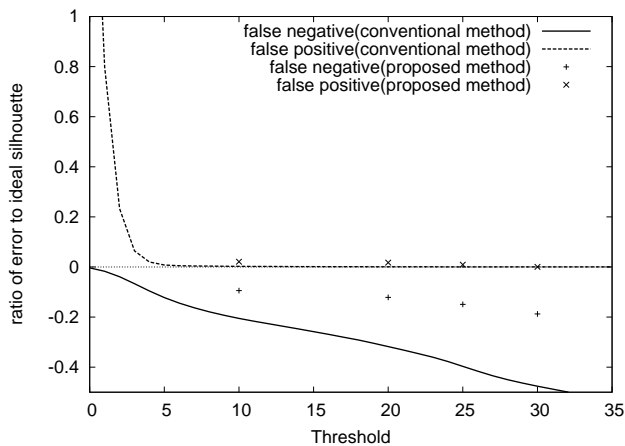


図 14 従来手法との比較（正解に影を含む）

理時間を必要とした。

最終的な対象領域抽出結果を図 12 に示す。図中赤色の部分是对象領域として抽出された領域を表す。CG1 に関しては十分正しく領域抽出が行われている。CG2 に関しても概ね正しい対象領域が得られているが、CG1 の場合と違い、境界部分にわずかに欠損が生じている。この違いは対象物体の形状に因るものと思われる。直方体のような物体は、ある視点で対象領域の輪郭として観測される部分が、他の視点においても輪郭として観測される可能性が高く、ある視点における対象領域の欠損が、他の視点に反映されやすい。一方 CG2 のように曲面で構成された物体の場合、各視点で対象領域の輪郭として観測される部分は、対象表面上では互いに異なる領域である場合が多く、ある視点で生じた欠損を他の視点で確認できずに誤り検出が正常に働かない場合が生じてしまうものと思われる。

実画像の領域抽出結果については、概ね正しい対象領域が得られ、従来手法で問題になるような欠損も生じていないことがわかる。人手によって抽出した対象領域を正解として、背景差分に基づく領域抽出と提案手法の性能を比較したグラフを図 13 に示す。グラフの横軸は背景差分画像を二値化する際の閾値を意味しており、提案手法においては、誤り検出の際に背景差分画像を二値化する際の閾値を指す。縦軸の正方向、負方向はそれぞれ、正解のシルエットの面積に対する、過検出及び欠損の割合を意味する。グラフを見ると、欠損に関しては、提案手法の有効性が十分に現われているが、過検出が従来手法よりも多く生じていることが分かる。この原因としては、提案手法が求めるべき対象領域の定義が、対象の影を含むからであると思われる。なぜならば、対象が存在することによって生じる影もまた、本論文で使用した視点間の制約条件を満たすからである。そこで、対象の影を正解に含んだ場合のグラフを図 14 に示す。正解に影を含んだ場合においては、過検出も従来手法とほぼ同レベルに抑えられており、過検出と欠損を総合的に判断すると提案手法が従来手法に比べて有効であることがわかる。

提案手法で求める対象領域が影を含んでしまう問題に関しては、影によって作られる visual hull の部分を削る視点を追加することで解決できる。例えば床面に影が生じるのであれば、カメラの投影中心が床平面上にあり、視線方向が床平面の法線

と直交するようなカメラをおけばよい。

以上の実験結果より、提案手法では各視点における見かけ上の対象領域の輪郭を、画像上のエッジに沿って選択しながら、視点間で互いに矛盾がなくなる組合せを求めているため、対象表面のうち、複数の視点から輪郭として観測される部分について、本手法の誤り訂正が有効に機能することが分かる。

5. 結 論

本論文では、多視点画像からの対象領域抽出という問題に対して、(1) それらの画像は共通の対象を異なる視点から撮影したものであることから互いに領域形状に制約を受けるという視点間の制約条件と、(2) 対象領域の輪郭は撮影画像上でエッジとして表れるはずであるという単視点上での制約条件を同時に使用する領域抽出手法を提案し、これらの制約条件に基づいた対象領域推定の誤り検出・訂正する枠組を示すとともに、実験によってその有効性を検証した。3.6 節で述べたように、提案した手法は従来の単視点における対象領域抽出手法の結果を誤り検出の手がかりとして利用する。このことから、本手法は従来の単視点対象領域抽出手法を、多視点画像間の制約条件を利用した誤り検出・訂正を含む計算の枠組へと統合することで、さらなる頑健さを実現したものであるといえる。

今後はより高速な計算アルゴリズムについて検討するとともに、対象の輪郭情報だけでなく、その内側のテクスチャ情報を統合することで、提案手法では扱うことのできなかった対象表面部分についても扱い、3次元形状と多視点対象領域の同時復元へと研究を進めることが期待される。

謝 辞

本研究は、文部科学省科学技術振興調整費リーディングプロジェクト「大型有形・無形文化財の高精度デジタル化ソフトウェアの開発」の支援を受けて行った。

文 献

- [1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *IJCV*, Vol. 1, No. 4, pp. 321–331, 1988.
- [2] Takashi Matsuyama, Xiaojun Wu, Takeshi Takai, and Shohei Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation and high fidelity visualization for 3d video. *CVIU*, Vol. 96, pp. 393–434, December 2004.
- [3] Gang Zeng and Long Quan. Silhouette extraction from multiple images of an unknown background. In *Proc. of ACCV*, pp. 628–633, 2004.
- [4] 川本剛司. 多視点画像の統合による視体積交差法の高精度化. 学士論文, 京都大学, 2000.
- [5] 陳蘇. 前景領域と視体積の投影像との整合性を考慮した3次元形状獲得. 学士論文, 京都大学, 2003.
- [6] 豊浦正広, 飯山将晃, 角所考, 美濃彦彦. ランダムパターン背景を用いた視体積交差法のためのシルエット欠損修復. In *Proc. of MIRU*, pp. 1247–1254, 2005.
- [7] Wojciech Matusik, Cheri Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In *Proc. of SIGGRAPH*, pp. 369–374, 2000.
- [8] R. Koch M. Pollefeys and L. Van Gool. A simple and efficient rectification method for general motion. In *Proc. of ICCV*, pp. 496–501, 1999.