

多視点動画像からの実時間 3次元形状復元の研究

武田 淳 岡谷 貴之 出口 光一郎

東北大学大学院情報科学研究科

takejun,okatani,kodeg@fractal.is.tohoku.ac.jp

3次元コンピュータグラフィックスを作成するための、画像による動物体3次元形状の測定(いわゆる、モーションキャプチャ)システムの構成について報告する。対象のまわりを取り囲んで配置した多数台のカメラによる画像から、背景を引き去った後、共通部分を抽出する volume intersection (または、shape from silhouette) の手法は、原理も分かりやすく、高速化による実時間計測の可能性が高い。ここでは、画像から動物体を安定に形状復元することを目的とし、6台のデジタルビデオカメラと1台のPCを用いた実時間形状復元システムの開発を行なった。本報告ではまず始めに、複数の視点から撮影した画像から、実時間で形状を復元する手法について述べる。次に、実際の画像を逆投影して重ね合わせることにより3次元形状を得る際の、実時間で行なうための高速化について述べる。現在の構成では、180cm立方の空間を1cmの分解能で形状復元をするとき、3~4フレーム/毎秒の処理速度であるが、同じ構成で、空間の分解能が2cm程度とすれば、毎秒10フレーム以上の復元が可能である。

A System for Real-time 3D Reconstruction from Multiple Motion Images

Jun Takeda, Takayuki Okatani, and Koichiro Deguchi

Graduate School of Information Sciences, Tohoku University

takejun,okatani,kodeg@fractal.is.tohoku.ac.jp

This article reports a configuration of a system for real-time 3D reconstruction from multiple motion images for 3D motion capturing. We employ the volume intersection (or, so-called the shape from silhouette) technique. In this system, multiple images are obtained by cameras arranged around a moving object. From the respective images back-ground images are subtracted, then, the object images are back-projected into 3D space. Finally, the volume intersection of each projection cones is extracted. This technique has great advantages of the simple and high-speed processing. Our system has 6 digital camera and one PC, and reconstruct 3D object in (180cm)³ space with 1cm resolution at 3 or 4 frames/s speed. It is possible to speed-up to over 10 frames/s when the resolution is reduced to 2cm. We report the techniques to achieve the processing capability as well as the total configuration of the system.

1 はじめに

3次元コンピュータグラフィックスを作成するための、画像による動物体3次元形状の測定（いわゆる、モーションキャプチャ）にはさまざまなものが開発されている[1]。中でも、対象のまわりを取り囲んで配置した多数台のカメラによる画像から、背景を引き去った後、共通部分を抽出する volume intersection（または、shape from silhouette）の手法は、原理も分かりやすく（[2][3]等）、高速化による実時間計測の可能性が高い。

そこで我々は、画像から動物体を安定に形状復元することを目的とし、6台のデジタルビデオカメラと1台のPCを用いた実時間形状復元システムの開発を行なった。本報告ではまず始めに、複数の視点から撮影した画像から、実時間で形状を復元する手法について述べる。次に、実際の画像を逆投影して重ね合わせることでより3次元形状を得る際の、実時間で行なうための高速化について述べる。

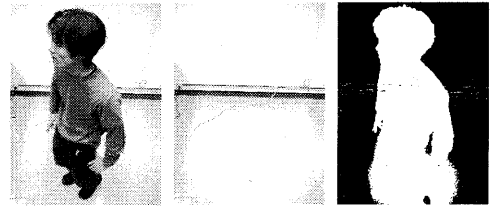


図2: 背景差分で得られる輪郭画像 - 左: カメラ画像 中: 背景画像 右: 2値化した差分画像

2 多視点動画画像からの対象の3次元形状復元

2.1 輪郭画像からの形状復元

多数のカメラを用いて多方向から対象を撮影すると、物体の形状は、各カメラから得られる物体の輪郭画像を逆投影した錐体内部の積で表される（図1）。より多い視点からの画像を用いれば、より正確に形状を復元することができる。ただし、凹部に関しては、形状輪郭を得ることが出来ないため、正確な形状復元をすることが出来ないという欠点があるが、ステレオを始めとする多視点画像からの形状復元で必要とする対応点のマッチングを原則として行わず、複雑な画像処理や演算処理を必要としないため、高速に3次元形状を復元することが可能である。また、カメラの諸パラメータが不正確であることによって復元結果が極端に悪くなったり、形状が大幅に欠落することはない。

本手法では、まず画像から形状復元をしたい対象物を抽出しなければならない。そのためには画像中から背景を取り去る必要がある。ここでは背景差分法を用いて対象物体の輪郭画像を抽出する。カメラから得られる画像と予め用意した背景画像と画素単位で赤青緑それぞれの色成分同士を引き算し、その結果から対象物体の輪郭画像を得る。図2右は、左の前景画像から中の背景画像を差分し、その差

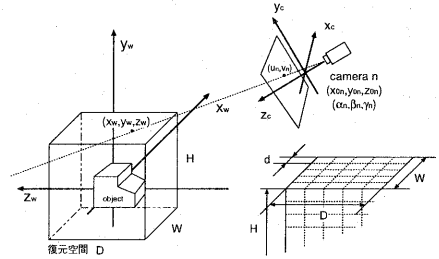


図3: 復元空間とカメラの関係

の結果が閾値より大きい画素は白、小さい画素は黒としたものである。

2.2 投影と逆投影

本報告で行なう形状復元は各カメラからの輪郭画像を逆投影した錐体内部の積として形状を得る。しかしながら、逆投影をするためには、カメラからの距離に比例して画像の拡大、補間処理が必要であり処理が複雑になる。そこで実際の復元計算では、復元をしたい空間中に仮想的な点を考え、各カメラの輪郭画像へ投影をする。投影された先が輪郭画像内かを判断すれば、その点は輪郭画像を逆投影した積の内部であるかを判断できる。

まず復元座標上で、原点を中心とする形状を復元したい範囲を決める。これを、復元空間と呼ぶ（図3）。復元空間の高さ（ y 方向）を H 、幅（ x 方向）を W 、奥行き（ z 方向）を D とする。次に、復元空間に間隔 d の格子を作る。これは、復元をする際の単位となる。これを、復元間隔と呼ぶ。

ある格子上の点 (x_w, y_w, z_w) を、 n 番目のカメラへ投影することを考える。復元空間の座標系に対するカメラの位置 (x_{0n}, y_{0n}, z_{0n}) 、カメラの姿勢が与えられれば、復元座標系に対する角度が点 (x_w, y_w, z_w) は、カメラ座標系へ座標変換ができる。カメラ座標系で表したこの点の座標を (x_c, y_c, z_c) とする。この点はさらに透視変換により、カメラの画像中 (u_n, v_n) へ投影される。もし n 番目のカメラ画像の中で (u_n, v_n) の点が輪郭内であれば、点 (x_w, y_w, z_w) は、このカメラから輪郭画像を逆投影した錐体内部である。その場合、点 (x_w, y_w, z_w) は形状を構成す

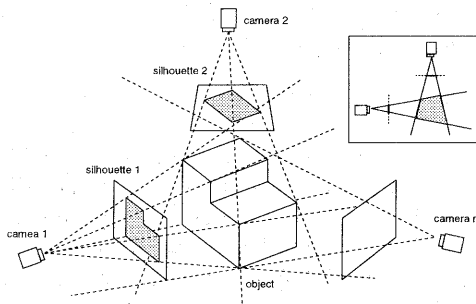


図1: 複数輪郭画像からの形状復元

一部である可能性があるため、続けて他のカメラに対して同様の計算を行なう。全てのカメラにて投影先が輪郭画像内であるとき、この点 (x_w, y_w, z_w) は錐体の積の内部、すなわち物体内部であると判断ができる。

以上の計算を行うと、物体形状は3次元空間内の小格子立方体の集合として表される。これに対し、コンピュータグラフィックスで広く用いられているデータ形式にポリゴンを用いたデータ形式がある。物体形状表面を多角形で近似し、多角形の集合として物体全体を表現する。

本手法では、ポリウムデータで得られた形状を、ポリゴンデータの形式へ変換し、形状の画面表示を行う。ポリウムデータから、ポリゴンデータへの変換には、マーチングキューブ法(文献[5])を用いる。

3 高速化の手法

実時間の形状復元をするためには、計算量を減らすための工夫が必要である。例えば、復元空間全体を計算するのではなく、形状復元に必要な部分のみを計算することなどが挙げられる。以下ではその具体的な手法について述べる。

3.1 復元可能空間

先に便宜上復元空間を定めたが、図4に示すように復元空間と復元が可能な空間は同一ではない。そこで、復元空間内から復元可能空間を予め探し、復元可能空間内でのみ形状復元の計算をする。

復元可能空間は各カメラの画像面を投影した積となり、多面体で表される。まず、復元空間内を荒い間隔でスキャンし、その点を各カメラへ投影することで復元可能空間内であるかを判断する。投影結果が全てのカメラ画像面内であるかを判断し、復元可能空間内であるかを判定する。

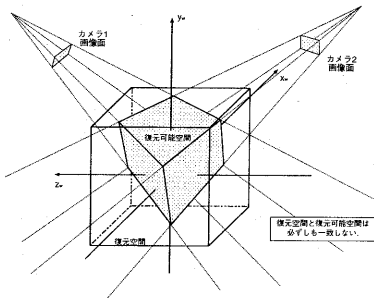


図4: カメラが2台の時の復元可能空間の例

3.2 プレサンプリング

続いて、復元可能空間内の形状の復元を行う。はじめから細かい間隔で復元するのではなく、対象物体外部や物体内部は荒く、復元形状にもっとも関わりのある物体表面近傍は細かく計算を行う。

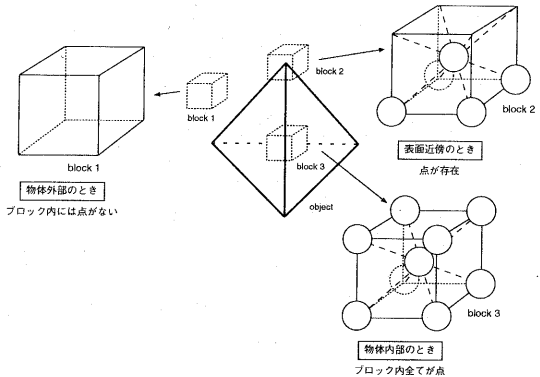


図5: 点の数と物体形状の関係

形状復元空間をいくつかのブロックに分けて、ブロックごとに荒く形状復元を行う。そのブロック内の物体の形状と、復元結果であるポリウムデータの点の数には図5に示すような関係がある。点の数が極端に少ないブロックは物体外部であるため、そのブロックの復元を中止する。点の数が極端に多い場所は物体の内部であり、復元形状に直接の影響は無いため、そのブロックの復元計算を中止する。それ以外は対象物体表面近傍である可能性が高いため復元形状に大きな関わりがあるので、ブロック内で計算をしていない点について追加の復元計算をし、細かい形状を得る。

3.3 形状の補間

実際に復元計算を行うとき、一番の問題となるのが、復元空間のサンプリング間隔である復元間隔である。精度良く形状復元をするためには、復元間隔を小さくする必要があるが、計算量が増えてしまう。冒頭でも述べたように、本手法による輪郭画像からの形状復元では細かい凹凸形状を復元することは難しいため、復元間隔を極端に細かくしても滑らかな形状が得られるわけではない。しかしながら、粗い形状のままではコンピュータグラフィックスなどへ応用は難しい。そこで、本手法では、荒い間隔で形状復元をし、得られた形状を補間することで滑らかな形状を得ることにする。本復元では、補間により良い形状を得るために、補間しやすいデータとなるよう復元空間のサンプリング方法を工夫する。

先にも述べたように、通常は格子間隔 d で格子状の点で形状復元を行なう(図6)。図の場合27点の計算を行う必要がある。

同じ復元空間を図7実線のようにサンプリングする。この場合得られる形状の精度は点と点との最大距離である $\sqrt{3}d$ である。しかし、このサンプリング方法では、図6と比較して、計算量を $1/3$ に減らすことができる。

このサンプリング方法の利点は、補間をする各点の条件が等しいため補間をしやすいつところにある。通常の間引きでは、補間をしたい点が格子の面の上であるのか、格子の辺の上であるのかによって、すでにサンプリングした既知の点との距離、隣接する既知の点の数が異なる。

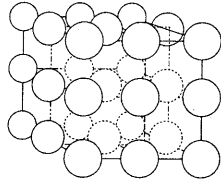


図 6: 格子点でのサンプリング方法 (27 点について計算が必要)

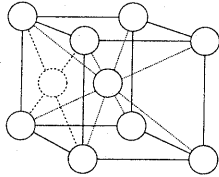


図 7: 高速化を目指した復元空間のサンプリング方法 - 実線部分がサンプリングを行なう点 (9 点について計算をする)

それに対し、本手法のサンプリング方法では、補間をする点は格子の面の上であっても格子の辺の上にあっても周囲を最大 6 個のサンプリングをした既知の点に隣接する。各辺の上の点は図 7 から (d, d, d) だけずれた空間を考えれば良い。面上でも辺上でも隣接する点の条件は変わらないため補間の処理が簡単になり、また各点の条件が等しいためそれだけ補間精度の向上も見込める。

図 7 で、実線部分は図 6 の格子点を本手法でサンプリングすることによって得られる点である。図中点線の点に注目する。この点は、周囲を 5 点に囲まれている。このように 4 点以上隣接する点があるとき、注目している点も物体の一部であるとみなし、補間する。

4 実験

4.1 システムの設定

本システムでの処理の流れを図 8 にまとめる。カメラを設置をする前に、焦点距離を測っておく。その後、カメラの設置をし、カメラ位置を実測する。形状が既知の対象を置き、その画像をもとに各カメラのパラメータの補正を行ない、カメラの正確な位置、角度を求める。

図 8 に示した中で、画像の入手からボリュームデータの出力までを実時間でやる。ボリュームデータの補間、ボリュームデータ形式からポリゴンデータ形式への変換は、形状復元後に行ない、表面形状を VRML 形式で出力する。その後、テクスチャマッピングを行なって 3 次元物体モデルの表面に肌の模様、服の模様などを張り付け、より写実的に 3 次元モデルを表示する。

開発システムでは SONY のデジタルビデオカメラ DFW-VL500 を 6 台用いた (以降、DV カメラと呼ぶ)。

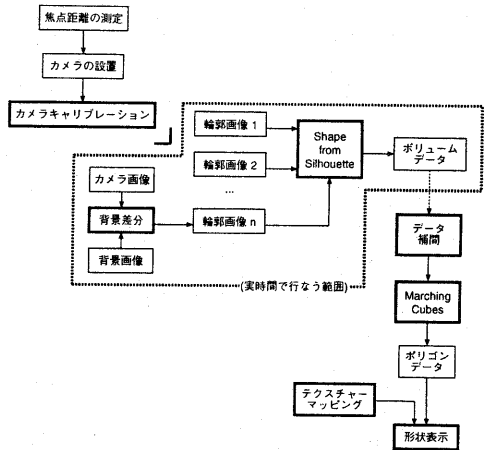


図 8: 実験の流れ

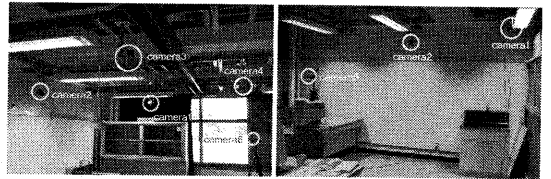


図 9: カメラ配置

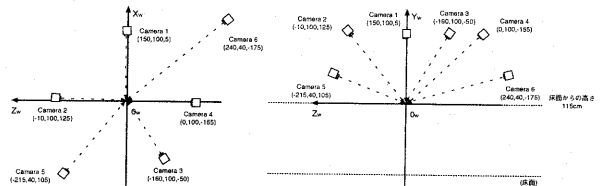


図 10: カメラ位置実測結果 (長さの単位は cm) - 左: 復元空間 x-z 平面図, 右: 復元空間 y-z 平面図

実験では、床からの高さが 115cm のある点を復元座標系の原点として 6 台のカメラを図 9, 図 10 に示すように配置をした。図 10 の座標値は復元座標系でのそれぞれのカメラの位置を実測したものである。全てのカメラは復元座標系の原点の方を向いている。カメラ位置の実測値より角度を算出し、画像を用いて補正を行なう。

復元をする範囲は原点を中心に $W = 180\text{cm}, H = 180\text{cm}, D = 180\text{cm}$ とした。使用したコンピュータは Pentium3-600Mhz, OS は Linux-2.2.14 である。カメラからの入力画像の大きさは $320 \times 240\text{pixel}$ である。



図 11: カメラパラメータの補正に用いた画像



図 12: カメラ画像と背景画像

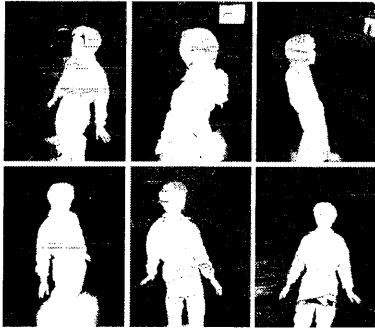


図 13: 輪郭画像

4.2 実験結果

使用した DV カメラには光学ズーム機能がついており、焦点距離とズームの関係も含め、焦点距離は予め測定した。

カメラパラメータの補正には図 11 に示すように 1 辺の長さが 50cm の大きさの立方体を用いた。実験では各カメラで 6 点の組を用いて補正を行なった。

図 12 に示されるカメラ画像と背景画像とで背景差分を行ない、2 値化した結果の例を図 13 に示す。

4.3 復元形状の評価

図 13 の輪郭画像から形状復元を行なった結果を図 14、図 15、図 16 に示す。これらの図は得られた 3 次元モデルを 3 つの異なる視点から表示したものである。左は格子間隔 $d = 1\text{cm}$ で図 6 に示される通常のサンプリング方法で形状復元を行なった結果である。右は同じ画像から、図 7 の

表 1: サンプリング方法の違いと得られる形状の違い

	点の数	ポリゴン数	誤差(点数)
通常の方法	129566	75712	-
提案した方法	25257	-	-
同 (1 回補間)	99875	-	-
同 (2 回補間)	121578	70156	9030

間引き方法でサンプリングを行ない、補間によって通常の $d = 1\text{cm}$ 間隔復元と同レベルの出力をした結果を表示したものである。

得られた形状の表面をより滑らかなものとするため、通常のサンプリング方法で得られた形状に対しても補間をかけた。そこで、本論文のサンプリング方法で得られる形状に対しては、 $d = 1\text{cm}$ 相当の出力をするための補間が 1 回、得られた $d = 1\text{cm}$ 相当の形状表面を滑らかにするための補間を 1 回、計 2 回補間をした。

復元形状のボリウムデータの点の数、マーチングキューブ法によってポリゴン化した後のポリゴン数を以下の表 1 にまとめる。誤差とは、通常のサンプリング方法で得られる形状と、間引きのサンプリング方法および補間で得られる形状とを形状を構成する点の数で比較したものである。

次に、補間によってどの程度元の形状が得られているかを観察する。通常のサンプリング方法で得られる形状図 16 にて、左の形状と補間によって得られる右の形状の間でボリウムデータの差分を計算する。差分をとった結果を図 17 に示す。胴体部など形状変化が緩やかなところは、本論文のサンプリング方法と補間でうまく形状の再現ができていくことがわかる。

4.4 計算高速化の評価

復元の速度性能を測定した。速度は対象物体の形状などその時々条件によって大きく異なるため、図 12 と同じような状況化での各速度の平均値を計測した。

本システムでは、カメラからの画像転送には IEEE1394 ホストアダプタを 2 個用いた。

各カメラ間は厳密には同期がとられてはいない。カメラ入力を切替えながらそれぞれのカメラから画像を取得している。6 台全てのカメラからの画像転送に要する時間は 20 ミリ秒以下であるため、画像の取得はほぼ同時であるとみなした。画像の大きさは $320 \times 240\text{pixel}$ 、YUV 4:2:2 形式でメモリへ転送される。YUV 形式から RGB 形式への変換は CPU で行う。6 台のカメラから画像を転送する速度は、RGB 形式への変換も含め、45 ミリ秒前後であった。

また、より高速な運動物体を計測する場合、もしくは本システムの復元能力が上がった場合には、カメラを切替えるタイムラグが無視できないものとなる。DV カメラの外部同期端子を利用するなどして、カメラ間の同期をとることを考えなければならぬだろう。

$320 \times 240\text{pixel}$ の画像 6 枚の背景差分を行なうのに要する時間は、対象物の大きさなどによらず一定で、68 ミリ秒前後であった (画像転送は含まず)。

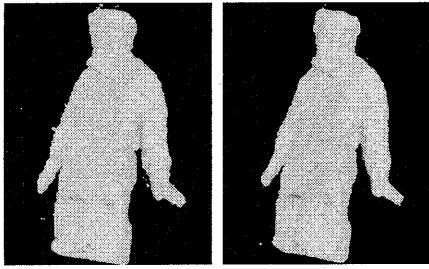


図 14: 復元結果のポリゴンによる表示 (正面図) - 左: $d = 1\text{cm}$ で通常のサンプリングをした形状、右: $d = 2\text{cm}$ で本論文で述べたサンプリングをし、補間をした形状

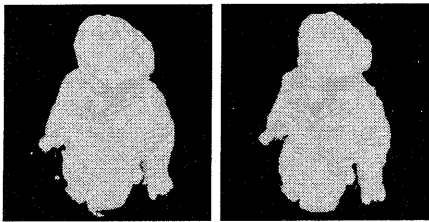


図 15: 復元結果のポリゴンによる表示 (頭周辺) - 左: $d = 1\text{cm}$ で通常のサンプリングをした形状、右: $d = 2\text{cm}$ で本論文で述べたサンプリングをし、補間をした形状



図 16: 復元結果のポリゴンによる表示 (腕周辺の拡大) - 左: $d = 1\text{cm}$ で通常のサンプリングをした形状、右: $d = 2\text{cm}$ で本論文で述べたサンプリングをし、補間をした形状

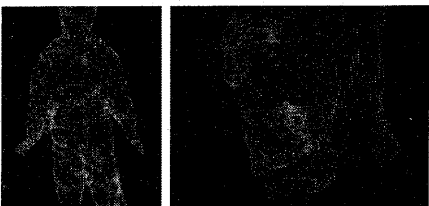


図 17: 形状の差分結果 - 全体と腕周辺の拡大

6枚の画像から形状復元を行なうのに要する時間は、物体の大きさやカメラの位置関係などで計算速度は大きく変化する。ここでは、図14から図16にて右に示した形状を出力する能力を計測した。このときの形状復元のみでの速度は4.4 frames/s程度であった(画像転送、表示、結果のディスク出力は含まず)。

以上を合わせたシステム全体(画像転送、背景差分、形状復元、画面表示)の処理速度は、図14, 図15, 図16右と同じレベルで復元をするとき、対象物の動きによって若干の変動はあるものの、おおよそ2.9 ~ 3.2frames/sであった。

5 まとめ

本論文では、実時間での安定した形状復元を目指し、多数のカメラを用いたシステムを開発し、多視点からの輪郭画像をもとに形状復元を行なった。

多視点からの画像を扱うため、復元座標系とカメラ座標系を考え、その間の座標変換の関係を回転行列と並進成分を用いて示した。カメラの幾何学的な配置関係を工夫し、効率よく対象物体の画像を入手するとともに、カメラの角度も位置の実測より計算によって求めることができた。実カメラパラメータとのずれは画像を用いた補正により小さくすることができた。

実時間での形状復元を目指し、計算量を減らすため様々な手法を用いた。中でも復元空間のサンプリングの方法を工夫することにより補間が容易になり、復元格子の間引きによって計算量を1/3に減らしても補間をせずに直接計算した形状と遜色のない形状を復元できた。

現在の構成では、180cm立方の空間を1cmの分解能で形状復元をするとき、3~4フレーム/毎秒の処理速度であるが、同じ構成で、空間の分解能が2cm程度とすれば、毎秒10フレーム以上の復元が可能である。本手法による形状復元では、計算機の台数を増やすことで容易に高速化が図れる。ここでは復元空間という概念を導入したが、復元空間を細かいブロックに分割し、1つのブロックに1台の計算機を割り当て、それぞれのブロックごとに形状復元を行えば、台数に応じて線形的に計算時間の短縮を行なうことができるであろう。

参考文献

- [1] 吉澤：“光によるヒトの3次元形状計測”，計測と制御，39, 4, p.267-272 (Apr. 2000)
- [2] P.Srivasan ほか：Computational Geometric Method in Volumetric Intersections for 3D Reconstruction, Pattern Recognition, vol.23, no.8, pp.843-857 (1990)
- [3] 松山ほか：“PCクラスタを用いた身体動作の実時間3次元映像化”，日本学術振興会未来開拓学術研究推進事業 分散協調視覚による動的3次元状況理解，1999年度研究成果報告書，pp.190-205 (Mar. 2000)
- [4] 出口：ロボットビジョンの基礎，コロナ社(2000)
- [5] Will Schroeder et al.：“The Visualization Toolkit 2nd Edition”，pp158-164, Prentice Hall PTR(1998)