二次元エッジ特徴と三次元エッジ形状の統合による物体認識

友 納 正 裕*

Object Recognition by Integration of 2-D Edge Features and 3-D Edge Shape

Masahiro Tomono*

This paper presents a method of recognizing objects and estimating their 3-D poses from a monocular image. This method integrates image edge points and a 3-D edge model into an object model. The method retrieves candidate objects from an object database using image edge points with SIFT feature vectors. Then, the method estimates the 3-D pose of each candidate object by minimizing the re-projection errors of the 3-D edge model. Experimental results show that the method successfully recognized non-textured objects and complex-shaped objects in real environments.

Key Words: Object Recognition, Pose Estimation, Invariant Features, Image Retrieval, Mobile Robot

1. は じ め に

ロボットが環境内を移動して様々な作業を行うためには、物体を認識する能力が不可欠である。ここでいう物体認識とは、あらかじめデータベースに登録しておいた物体を入力画像から見つけ、その三次元空間での姿勢を推定することである（本稿では、位置と方向をまとめて姿勢と呼ぶ）。例えば、ロボットが物体の操作や運搬などの作業を行うには、個々の物体を識別してその姿勢を知る必要がある。多くの場合、操作物体の置き場所も机や棚などの物体であるので、それらも重要な認識対象になる。また、物体認識はロボットの空間理解と密接な関係にあり、物体認識が可能になれば、個々の物体に分節化された抽象度の高い三次元地図が実現できる [19]。

本稿では、ロボットが単眼カメラで撮影した1枚の画像から物体認識を行うことを考える。物体を認識するには、その物体に関する知識が必要であり、その知識を物体モデルと呼ぶ。物体認識の課題として、オクルージョン（物体の一部が隠れること）や照明変化に対するロバスト性、多くの物体が混在する混雑環境での識別性、物体モデル数に対する計算量などがある。近年、Lowe が提案した SIFT (Scale Invariant Feature Transform) 法 [8] により、オクルージョン、照明変化、混雑環境に対処できる物体認識が実現できるようになってきた。しかも、SIFT 法では、平均的には特徴点数の対数に比例する時間で物体モデルを検索できるため、計算量が小さく、目標物体を限定せずに認識するのに適している。

ロボットの物体認識に SIFT 法を用いるには、二つの問題が

ある。一つは、SIFT 法は二次元画像の認識手法であり、物体の三次元姿勢は得られないことである。これは、上述のようなロボットへの応用には大きな問題となる。もう一つは、SIFT 法はテクスチャの少ない物体には必ずしもうまく作用しないことである。屋内環境はロボットの活動場所として重要だが、家具などテクスチャの少ない物体が数多く存在する。そのため、テクスチャの有無によらずに物体認識を行えることが望ましい。

本稿では、SIFT 法に基づく二次元モデルと単眼カメラ画像列から復元した三次元モデルとを統合した物体モデルを用いて、これらの問題に対処する物体認識方式を提案する。本方式のポイント、エッジ点を用いて物体モデルを構成することにある。エッジ点とは画像から抽出されたエッジ上の点である。二次元モデルは1枚の画像内のエッジ点の集合であり、各エッジ点には SIFT 特徴ベクトルが付加される。三次元モデルは、画像上のエッジ点を三次元復元した点（三次元エッジ点）の集合である。本稿ではこれを三次元エッジモデルと呼ぶ。各三次元エッジ点には複数のカメラ視点から見た画像のエッジ点を対応づける。二次元モデルを用いて、入力画像に写っている物体が何であるかを認識し、さらに、対応する三次元エッジモデルを用いて三次元姿勢推定を行う。エッジ点はテクスチャがなくても大量に抽出できるため、テクスチャが少ない物体をロバストに認識するのに適する。本稿の貢献は、SIFT 法をエッジ点に拡張し、さらに、三次元形状に統合することで、テクスチャの有無によらずに、三次元姿勢推定を含む物体認識を実現したことにある。

2. 関 連 研 究

カメラ画像を用いた物体認識には多くの課題がある。まず、カメラ視点によって、同じ物体でも画像上の見え方が変わる。具

原稿受付 2007 年 1 月 13 日

*東洋大学機能ロボティクス学科

*Department of System Robotics, Toyo University

体的には、画像上の位置（並進）、角度（回転）、大きさ（スケール）が変化するし、カメラの透視投影により物体形状は歪む。次に、環境条件も物体の見え方に影響する。照明によって色や輝度に変化するし、オクルージョンがあれば形状が欠ける。また、混雑環境では、背景に多くの物体が混在し、物体の識別が難しくなる。これらの違いを吸収して、同じ物体であると認識しなければならない。

物体認識は、一般には、画像特徴とモデル特徴のマッチングをとり、そのうち物体形状や物体・カメラ間の相対姿勢で決まる幾何制約を満たすものを見つけることで実現される。この際、どのような特徴を利用するかが重要となる。並進、回転、スケール、照明が変化しても認識できるために、これらに対する不変量の特徴として用いることが多い。オクルージョンで物体の一部が欠けることを考慮すると、小領域から抽出できる局所特徴が有利である。さらに、混雑環境で物体を認識するには、個々の特徴自体が高い識別性をもつことが望ましい。

これらの問題に対処するために、種々の方法が提案されてきた。モデルベース・アプローチでは、物体・カメラ間の幾何制約を利用して解を求める。用いる特徴は、コーナ点や線分など、それ自体は識別性の低いものである。Geometric hashing 法は、 n 個の特徴点から基底を作り（弱透視投影では $n = 3$ ）、残りの特徴点をその基底が張る座標系に変換して不変特徴量を求め、物体データベースの索引として利用する [3] [7]。Alignment 法は、画像特徴点とモデル特徴点の 2 個組から弱透視投影パラメータを求め、それを画像エッジとモデルエッジの一致度により検証して解を求める [6]。筆者らは、エッジ線分を用いて、透視投影のもとで三次元物体を認識する方法を開発した [17]。

これに対して、アピアランスベース・アプローチでは、画像特徴の識別性に重点が置かれる。固有空間法は、特徴空間を主成分分析により変換し、識別性の高い少数の特徴で効率的に物体を認識する [13]。さらに、局所領域ごとに固有空間法を適用することでオクルージョンに対処したり、また、照明に対するロバスト性を高める方法も提案されている [14]。

近年、不変性と識別性をもった画像特徴を利用する手法が開発されている [8] [11] [20]。SIFT 法は、画像の相似変換に不変な特徴を用いる [8]。まず、DOG（difference of Gaussians）オペレータの極値として特徴点の位置を決め、特徴点近傍の局所画像の勾配方向ヒストグラムから特徴ベクトルを生成する。この特徴ベクトルは 128 次元あり、識別性が極めて高い。また、ピラミッド画像を用いたスケール解析を行い、特徴点のスケール不変性を得る。認識の際は、この特徴ベクトルを用いて画像特徴点とモデル特徴点を対応づけ、投票によりマッチングの検証を行う。このほか、アフィン不変特徴量も提案されているが [11] [20]、特徴ベクトルの識別性の高さ、および、処理の高速さから、現時点では、SIFT 法が最も実用性が高い。

本稿の提案方式は、SIFT 法の流れに属する。上述のように、提案方式は、エッジ点を用いることでテクスチャの少ない物体にも適用できるようにしたこと、また、物体の三次元姿勢推定を行えるようにしたことが、元の SIFT 法との違いである。SIFT 法をエッジ点に拡張した物体認識方式として [12] があるが、三次元姿勢推定は行っていない。

Fig. 1 Procedure of object modeling and recognition

3. システムの概要

3.1 要件

本稿で考える物体認識は、入力画像中の主要な物体を物体データベースから見つけ、その三次元空間での姿勢を推定する問題である。対象とする物体は剛体であり、変形はしないものとする。その形状は曲面や曲線を含んでもよいが、輪郭あるいは表面から曖昧でないエッジが抽出できることを条件とする。例えば、曲面の輪郭のように、視点によって見かけのエッジが連続的に変化する物体の認識は保証しない。ただし、この場合でも、輪郭以外に有用な情報があれば認識は可能である。

上記条件を満たす物体は多数あり、適用範囲は広いが、本稿では、具体例として家具を考える。家具は操作物体の置き場所になると同時に、ナビゲーションにおいて障害物にもランドマークにもなり、しかも、環境地図の構成要素であることから、ロボットにとって重要性が高い。

3.2 処理の流れ

本システムは、物体モデル生成と物体認識の二段階から成る。Fig. 1 に処理の流れを示す。物体モデル生成では、単眼カメラで撮影した画像列を入力し、Structure-from-motion (SFM) 技術を用いて、カメラ運動（画像列の各画像を撮影したカメラ姿勢の時系列）を推定する。次に、そのカメラ運動をもとに、各エッジ点を三次元復元して、三次元エッジモデルを生成する。さらに、各画像ごとにエッジ点集合から二次元モデルを生成する。各エッジ点には SIFT 特徴ベクトルを付加し、また、三次元エッジ点を対応づける。なお、物体モデル生成の際、画像から対象物体を自動で切り出すのは困難なため、人間が手動で切り出す。物体認識では、1 枚の単眼カメラ画像を入力し、二次元モデルを用いて画像に写っている物体の候補を求める。認識時には物体を手動で切り出す必要はない。次に、その各候補について、三次元エッジモデルを用いて姿勢推定を行う。

本稿では、これらのうち、二次元モデル生成と物体認識、および、姿勢推定に焦点を当てる。カメラ運動の推定や三次元エッジモデル生成の詳細は、他稿 [18] [21] を参照されたい。

4. 物体モデル

4.1 物体データベース

あらかじめ物体モデルを生成して、物体データベースに登録しておく。Fig. 2 に物体データベースの基本構成を示す。1 個の物体モデルは、1 個の三次元エッジモデルと複数の二次元モデル

Fig. 2 Object database

ルから成る。二次元モデルを複数持つのは、様々なカメラ視点から撮影した画像に対応するためである。二次元モデルは SIFT 特徴ベクトルを索引として、kd 木を用いて高速に検索される。

4.2 三次元エッジモデル

三次元エッジモデルは、画像上のエッジ点を三次元復元して生成する。そのために、まず、画像列から SFM を用いてカメラ運動を復元する。KLT トラッカ [15] を用いて、画像間で特徴点を追跡し、その特徴点列から因子分解法 [16] により、カメラ運動と物体形状（特徴点の三次元位置）をアフィン復元する。この復元結果を初期値として、非線形最小化により、投影誤差を最小にするカメラ運動を求める [21]。

カメラ運動を推定した後、エッジ点を三次元復元する。まず、画像列の各画像から、Canny オペレータ [2] を用いてエッジ点を抽出する。次節の二次元モデルと違って、このエッジ点のスケールは固定である。次に、エピソード制約を利用して、画像間でエッジ点の対応をとる。カメラ運動からエピソード線が得られるので、対応点探索はそのエピソード線上に限定できる。そこで、エピソード線とエッジ線分の交点を対応点候補とする。次に、対応候補となるエッジ点近傍の局所画像に対して正規化相関を用いたマッチングにより候補を選ぶ。複数の対応点候補が得られた場合は、各候補を追跡しつつ、カメラの移動につれて候補を絞り込んでいく処理を行う [18]。

4.3 二次元モデル

二次元モデルは、モデル画像、モデル画像を撮影したときのカメラ姿勢、モデル画像から抽出したエッジ点群で構成される。各エッジ点には、二次元認識のために SIFT 特徴ベクトルを付加する。SIFT 特徴ベクトルは局所特徴量なので、透視投影により物体形状が多少変形してもマッチする。そこで、約 30° おきにカメラ角度を変化させて撮影した画像をモデル画像にして二次元モデルを複数生成しておくことにより、種々のカメラ視点で撮影した入力画像に対応できるようにする。

4.3.1 スケールスペース解析を用いたエッジ点抽出

エッジ点は Canny オペレータを用いて次のように抽出する [2]。まず、ガウスオペレータで画像を平滑化する。

$$L(x, y, s) = I(x, y) * G(s) \quad (1)$$

ここで、 $I(x, y)$ は原画像の画素 (x, y) の値、 $G(s)$ は標準偏差 s のガウス関数である。この標準偏差をスケールと呼ぶ。* は畳み込み積分を表す。 $L(x, y, s)$ は、平滑化後の画像である。次に、 $L(x, y, s)$ を一次微分して勾配ベクトル (L_x, L_y) を得る。ただし、

(a) Edges with a constant scale

(b) Edges by scale space analysis

Fig. 3 Edge extraction with/without scale space analysis

$$L_x = \frac{\partial L(x, y, s)}{\partial x}, \quad L_y = \frac{\partial L(x, y, s)}{\partial y}$$

である。そして、各画素 (x, y) の勾配ベクトルを用いて、その勾配方向に沿ってエッジ強度の極大値を求めることで、スケール s におけるエッジ点を求める。エッジ強度は勾配ベクトルの大きさから計算される。

ここで注意すべきことは、スケール s を固定した場合、入力画像とモデル画像で物体サイズが異なると、抽出されるエッジ形状が変わることである。これに対処するために、スケールスペース解析 [10] を行って、物体サイズにできるだけ依存しない安定したエッジを抽出する。スケールスペース解析では、上記の極大値探索を x, y だけでなく、スケール s に対しても行う。まず、勾配ベクトルの大きさをスケール s で正規化する。

$$D = s^{\gamma} \sqrt{L_x^2 + L_y^2} \quad (2)$$

そして、 D を極大にする x, y, s を求める。 x, y については、上記のように勾配方向の極大をとる。 s は、画像中の物体サイズに比例した値となる。このとき、物体サイズによらず同じ位置にエッジ点が得られる。すなわち、エッジ形状は物体サイズに対して不変になる。これをスケール不変なエッジ点と呼ぶ。

本稿では、スケール不変なエッジ点を次の手順で求める。まず、スケール s を少しずつ変化させて Canny オペレータを適用し、各スケールでのエッジ点を抽出する。実装では、 $\gamma = 1$ とし、 s は 0.2 ずつ変化させた。次に、得られたエッジ点のうち、 s について D が極大となるものを選択する。各エッジ点は、位置、スケール、勾配方向の四つ組 (x, y, s, θ) で定義される。

Fig. 3 に、原画像とその 1/2 縮小画像でエッジ抽出の様子を比較した例を示す。Fig. 3(a) は、スケール固定 ($s = 1$) の Canny オペレータで抽出したエッジ点である。原画像のエッジ点数は 8,752 個、そのうち、縮小画像でも同じ位置に抽出されたエッジ点は 39%であった。Fig. 3(b) は、スケールスペース解析により抽出したエッジ点である。原画像のエッジ点数は 11,642 個、そのうち、縮小画像でも同じ位置に抽出されたエッジ点は 58%であった。このように、スケールスペース解析を行うことで、エッジ点位置の再現性が高くなる。それ以上に重要なのは、次項で述べるように、各エッジ点のスケール s が特徴ベクトル

Fig. 4 Scale invariant features (merely 2% are displayed)

のスケール不変性に利用できることである。

4.3.2 特徴ベクトルの計算

エッジ点のマッチングのために、各エッジ点に SIFT 特徴ベクトルを付加する。SIFT 特徴ベクトルは、エッジ点の近傍領域を 4×4 のブロックに分割して、各ブロック内にある画素の勾配方向のヒストグラムを並べて作る [8]。ヒストグラムは 45° おきにとるので、 $4 \times 4 \times 8 = 128$ 次元のベクトルになる。ここで、特徴ベクトルが回転不変となるように、注目しているエッジ点の勾配方向からの相対角度で方向ヒストグラムを作る。また、照明不変性は、微分画素を用いること、特徴ベクトルの閾値処理と正規化により得られる [8]。

特徴ベクトルがスケール不変となるように、特徴ベクトルを生成する近傍領域の大きさを決める。入力画像とモデル画像で写っている物体の大きさが異なる場合、物体に対する近傍領域の相対サイズが両画像で変わるため、固定サイズの近傍領域で上記方向ヒストグラムを作ると、異なる特徴ベクトルが得られてしまう。これに対処するには、近傍領域の半径が物体サイズに比例するように定める必要があるが、エッジ点を用いるため、SIFT 法と同じ方法は使えない。このための方法はいくつかあるが [12] [19]、前項でエッジ点のスケール s が得られているので、本稿ではそれを定数倍して用いる（実装では 2 倍）。

Fig. 4 に例を示す。左が原画像、右がその 1/2 縮小画像である。両画像で位置が一致するエッジ点を一部抜粋し、そのスケールを円の半径で示した（表示は 5 倍に拡大）。両画像での円の半径比が画像サイズの比とほぼ一致していることが分かる。

4.3.3 特徴ベクトルによる索引づけ

二次元認識は、エッジ点の SIFT 特徴ベクトルを用いて入力画像と二次元モデルをマッチングすることで行う。基本的には、入力画像エッジ点とモデル画像エッジ点の各組み合わせに対して、SIFT 特徴ベクトルの一致度を求め、マッチするエッジ点の多い二次元モデルを認識候補とする。だが、これを全数探索で行うと多大な時間を要する。

そこで、SIFT 法と同様に、SIFT 特徴ベクトルを索引として、kd 木を用いて二次元モデルを高速で検索する。すなわち、全二次元モデルの各エッジ点の SIFT 特徴ベクトルから kd 木を構成し、入力画像エッジ点と類似した SIFT 特徴ベクトルをもつモデル画像エッジ点を kd 木で高速に検索して、エッジ点が多くマッチする二次元モデルを求める [8]。この kd 木は、SIFT 特徴ベクトルの 128 個の要素のうち、値の分散が大きいものからデータ空間を二分して構成した二分木である。検索には、Best-Bin-First 法 (BBF 法) を用いる [1]。BBF 法は、検索の

Fig. 5 Correspondence between a 3-D edge point and 2-D edge points

完全性は保証されないが、高速に類似した特徴ベクトルを検索する。これにより、平均としてモデル画像エッジ点数の log オーダーでモデル画像エッジ点を検索できる。

4.4 二次元モデルと三次元エッジモデルの対応づけ

二次元モデルは各モデル画像ごとに設定されたものである。これを三次元エッジモデルと対応づけて、物体の三次元姿勢推定ができるようにする。前述のように、モデル画像は約 30° おきに撮影すればよい。そこで、三次元エッジモデル生成に用いた画像列から約 30° おきに画像を抜粋して、モデル画像として用いる。まず、これらのモデル画像を撮影したカメラ姿勢に基づいて三次元エッジモデルを各モデル画像に投影する。このカメラ姿勢は、4.2 節で求めたものである。そして、各モデル画像に投影された三次元エッジ点のうち、二次元モデルのエッジ点と位置が一致するものを取り出し、二次元エッジ点と三次元エッジ点のペアを作る。例えば、Fig. 5 で、三次元エッジ点 P の画像 I_k への投影点が二次元エッジ点 q と一致するので、 q と P を対応づける。 q' と P も同様である。

なお、各モデル画像から対象物体を切り出す処理が必要であるが、現時点では人間がモデル画像上でマウスを用いて行う。

5. 物 体 認 識

物体認識は、二次元モデルを用いた二次元認識と三次元エッジモデルを用いた三次元姿勢推定の二段階から成る。

5.1 二次元認識

まず、二次元モデル生成と同様にして入力画像からエッジ点を抽出し、各エッジ点の SIFT 特徴ベクトルを生成する。そして、以下の手順で二次元での物体認識を行う。

5.1.1 二次元モデルの選択

前述の kd 木を用いて、入力画像エッジ点とマッチするモデル画像エッジ点を求める。そして、マッチした各モデル画像エッジ点について、その所属する二次元モデルをカウントする。カウント数の多い二次元モデルは、マッチしたモデル画像エッジ点が多いことになるので、その二次元モデルを第一段階の認識候補として選択する。

5.1.2 二次元姿勢推定

5.1.1 項で得たエッジ点対応には多くの誤対応が含まれる。そこで、5.1.1 項で求めた二次元モデルごとに二次元姿勢推定を行い、幾何学的整合性に基づいて誤対応を取り除く。

この二次元姿勢推定では、入力画像とモデル画像の間の相似変換パラメータ（回転角、並進量、スケール比）を、エッジ点対応の投票により求める。まず、回転角とスケール比を投票で

求める。エッジ点の勾配方向に基づいて、式 (3) のように、入力画像エッジ点とモデル画像エッジ点の間の回転角 θ を計算する。また、エッジ点のスケールに基づいて、式 (4) のように、スケール比 s を計算する。ここで、 θ_i, s_i は、それぞれ、入力画像エッジ点の勾配方向とスケール、 θ_m, s_m は、それぞれ、モデル画像エッジ点の勾配方向とスケールである。そして、入力画像エッジ点とモデル画像エッジ点の各対応について θ と s を求めて、回転角とスケール比の二次元空間に投票し、投票の極大値のうち所定の閾値（実装では 50）を超えたものを、回転角とスケール比の候補として採用する。

$$\theta = \theta_i - \theta_m \quad (3)$$

$$s = s_i / s_m \quad (4)$$

次に、候補になった θ と s を満たすエッジ点対応を用いて、並進量を投票によって求める。エッジ点の並進量は式 (5)、(6) で計算される。 x_i, y_i は入力画像エッジ点の画像上の位置、 x_m, y_m はモデル画像エッジ点の画像上の位置である。各エッジ点対応について Δx と Δy を並進量の二次元空間に投票し、投票値が最大となるものを候補として採用する。

$$\Delta x = x_i - s (x_m \cos \theta - y_m \sin \theta) \quad (5)$$

$$\Delta y = y_i - s (x_m \sin \theta + y_m \cos \theta) \quad (6)$$

このようにして求めた相似変換パラメータ値の候補それぞれについて、誤対応除去を行う。すなわち、そのパラメータ値を満たすエッジ点対応だけを取り出し、残りは誤対応として除去する。この時点でマッチしたモデル画像エッジ点を多くもつ二次元モデルとその入力画像エッジ点・モデル画像エッジ点ペアの集合が、二次元認識の出力となる。実装では、全モデル画像エッジ点数の 1% 以上がマッチした二次元モデルを選択した。

なお、SIFT 法では、相似変換ではなくアフィン変換パラメータを求めて、二次元レベルで当てはめ精度の高いマッチングを行う [8] [12]。それに対して、本方式では、この後、完全な三次元姿勢推定を行うので、二次元認識の段階では計算コストが小さい相似変換に留めている。

5.2 三次元認識

5.2.1 三次元姿勢推定

三次元姿勢推定は、三次元エッジモデルを入力画像に投影したときに、その投影像が画像中の物体と一致する物体モデルとカメラの相対姿勢（以後、カメラ姿勢）を計算する問題である。この計算には、三次元エッジ点と入力画像エッジ点の対応が必要である。まず、モデル画像エッジ点と三次元エッジ点の対応づけは、4.4 節によりなされている。また、モデル画像エッジ点と入力画像エッジ点の対応は、前節の二次元認識によりとれている。これより、三次元エッジ点と入力画像エッジ点の対応も得ることができる。このエッジ点対応に基づいて、投影誤差が極小となるカメラ姿勢を計算すればよい。

この解法として様々なものがあるが、本稿ではカメラ回転角についての勾配法を提案する。まず、次のようにして、三次元エッジ点の投影誤差をカメラ回転角 $\tau = (\alpha, \beta, \gamma)$ の関数 $g(\tau)$ で表す。三次元エッジ点 $\mathbf{P}_i = (x_i, y_i, z_i)^T$ と入力画像エッジ点

$q = (u_i, v_i)^T$ が対応する場合、 τ を与えると、カメラの並進ベクトル $\mathbf{t} = (t_x, t_y, t_z)^T$ は式 (7) を満たす。ただし、 f はカメラの焦点距離、 r_{mn} は τ による回転行列 $R(\tau)$ の要素である。 N 個のエッジ点の対応についてこの式を作り、最小二乗法で $\mathbf{t}$ を求める。これで回転と並進が決まるので、そのカメラ姿勢を用いてこの N 個の三次元エッジ点を画像上に投影し、対応する入力画像エッジ点との二乗距離の和を式 (8) のように求め、 $g(\tau)$ とする。

$$\begin{pmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{pmatrix} \mathbf{t} = \begin{pmatrix} A_1 x_i + A_2 y_i + A_3 z_i \\ B_1 x_i + B_2 y_i + B_3 z_i \end{pmatrix} \quad (7)$$

$$A_k = f r_{k1} - u_i r_{k3}$$

$$B_k = f r_{k2} - v_i r_{k3}$$

$$g(\tau) = \sum_{i=1}^N \left\{ \left(u_i - f \frac{X_i}{Z_i} \right)^2 + \left(v_i - f \frac{Y_i}{Z_i} \right)^2 \right\} \quad (8)$$

$$(X_i, Y_i, Z_i)^T = R(\tau) \mathbf{P}_i + \mathbf{t}$$

勾配法による $g(\tau)$ の極小値探索は次のように行う。まず、モデル画像を撮影したときのカメラ姿勢の回転成分を初期値とし、そのときの投影誤差 $g(\tau)$ を式 (7)、(8) より計算する。これは、二次元でうまくマッチングできたのであれば、そのモデル画像のカメラ姿勢の近くに解があると期待できるからである。次に、カメラ回転角 τ を微小量だけ変化させて、そのときの $g(\tau)$ の変化分を求める。これより、 $g(\tau)$ の微分係数を近似的に求め、 $g(\tau)$ を小さくする方向に新たな τ を設定する。これを繰り返して $g(\tau)$ を極小にするカメラ姿勢を求める。

この方法は、独立変数が角度だけなので探索範囲が限定でき、並進を含めた 6 個の独立変数を用いる方法よりも探索の制御が容易で、その挙動も安定しているという特長がある。二次元認識でマッチしたモデル画像は、入力画像とカメラ回転角は近いが、並進量は近いとは限らないので、これは大きな利点となる。ただし、勾配法を用いているので、原理的に極値に陥る可能性がある。経験的には極値に陥ることは少ないが、この対処は今後の課題である。また、この問題は 3 点の対応があれば closed form で解けるが [5]、本稿の方法は解法が単純であり、また、画素誤差に対してロバストであるという利点がある。

5.2.2 RANSAC による誤対応除去

5.1 節の二次元姿勢推定を行っても、エッジ点の誤対応は完全に除去できない。1 個でもエッジ点の誤対応があると、上述のカメラ姿勢計算の誤差が大きくなるため、ロバスト推定の一つである RANSAC [4] で対処する。手順を以下に示す。

(1) エッジ点対応の選出

二次元認識で得られたエッジ点対応のうち、 N 組をランダムに選出する（実装では、 $N = 5$ ）。

(2) カメラ姿勢の計算

ステップ (1) で得たエッジ点対応を用いて、上述の勾配法でカメラ姿勢を計算する。

(3) カメラ姿勢の評価

残りのエッジ点对应のうち、ステップ (2) で得られたカメラ姿勢において三次元エッジ点の投影誤差が閾値（実装では 1 画素）以下であるものの個数 M をカウントする。

(4) ステップ (1)～(3) を所定の回数繰り返し（実装では 100 回）、 M が最大となるカメラ姿勢を解として採用する。

6. 実 験

提案方式を Java 言語で実装し、二次元認識および三次元認識の実験を行った。CPU は Pentium 4 (3.0 [GHz])、メモリは 1 [GB] である。画像はデジタルカメラおよびパンチルトカメラで撮影し、解像度を 320×240 [画素] に縮小して用いた。原画像はカラーであるが、モノクロに変換して用いた。カメラの画角は約 47° である。カメラ校正は焦点距離についてのみ行った。また、閾値等のパラメータ値は実験中は固定した。

6.1 特徴抽出

大量のエッジ点を特徴点として利用することが提案方式のポイントなので、提案方式と Lowe による SIFT 法とで、得られる特徴点数を比較した。SIFT 特徴点は、Lowe が作成したプログラム [9] を用いて抽出した。このプログラムは、入力画像から特徴点の位置、スケール、特徴ベクトルを出力するものである。

両者の比較を Fig. 6 に示す。同図は、机や冷蔵庫などの物体領域をマウスで指定し、そこから抽出した特徴点を数えたものである。画像上の円の中心がマッチした特徴点の位置、半径がスケールを表す。同図上段は、SIFT 法の結果であり、テクスチャが少ない机や冷蔵庫では、数十個から百個程度の特徴点しか得られていない。同図下段は、提案方式の結果であり、テクスチャがなくても 1,000 個以上の特徴点が得られている。

6.2 二次元認識

二次元認識では、単眼カメラで撮影した 1 枚の画像を入力し、マッチする二次元モデルを出力する。目標物体は与えず、画像に写っている物体の二次元モデルを物体データベースから探す。物体データベースには、机 2 種類 (deskW, deskB)、流し台 (sink)、冷蔵庫 (refrig)、小物置き台 (small table)、ロボット (ym2000) の 6 個の物体モデルを登録した。物体モデルは、実環境で物体を撮影した画像列から生成した。物体ごとに、その画像列から 2～5 枚の画像を抜粋して二次元モデルとした。画

Fig. 6 Features with scale extracted from objects (upper: SIFT features, lower: edge points)

像は人間が手で撮影し、カメラの高さは 80～140 [cm] 程度、仰角は下向きに $0 \sim 30^\circ$ 程度である。モデル画像の総数は 19 枚、エッジ点の総数は 41,890 個である。

144 枚の画像を用いて、提案方式による二次元認識の実験を行った。画像は、人間が手で撮影したものと、ロボットに載せたパンチルトカメラで撮影したものが混在している。ロボットのカメラの高さは約 90 [cm] で固定であるが、人間のカメラは高さがまちまちである。カメラの仰角は、どちらも $0 \sim 30^\circ$ 程度で様々である。この実験では、スコア（マッチした特徴点数）が最大の二次元モデルを解として採用した。マッチした特徴点数は、入力画像に依存するが、50～200 個程度であった。認識成功率（スコア最大モデルが正解であった率）は、96.5% (139/144 枚) であった。なお、スコアが最大ではないが 10 位以内の候補に入った率は 100% であった。

認識結果の一部を Fig. 7 に示す。左が入力画像、右がモデル画像であり、画像上の円がマッチしたエッジ点の位置とスケールを表す。白い点はマッチしなかったエッジ点である。Fig. 7 (a) にあるように、オクルージョンのために物体の一部が隠れていても認識できている。Fig. 7 (a) と Fig. 7 (b) にあるように、テクスチャが少ない物体でも認識できている。また、Fig. 7 (c) で顕著のように、入力画像とモデル画像で物体のサイズが違っても認識できている。今回の実験では、二次元モデルに対して入力画像中の物体サイズが 0.4～2 倍に変化しても、認識に成功

(a) deskW

(b) refrig

(c) small table

Fig. 7 Matched points in 2-D recognition by our method (left: input images, right: 2-D models)

した。

性能比較のため、Lowe の SIFT 特徴点を用いた二次元認識の実験を行った。この実験では、上述の Lowe のプログラムを用いて特徴点抽出と特徴ベクトル生成を行い、それ以外の処理はすべて提案方式と同じプログラムを用いた。SIFT 法では全体の特徴点数が少ないため、三次元姿勢推定を想定したときに必要となる 3 点以上がマッチすれば合格とした。実験での認識成功率は、49.3% (71/144 枚) であった。傾向として、small table や ym2000 などのテクスチャが多い物体（特徴点数は、100~300 個）は、認識率がほぼ 100%だが、deskW や deskB、refrig などテクスチャが少ない物体は認識率が 0~30%であった。Fig. 8 に結果の一部を示す。左が入力画像、右がモデル画像である。Fig. 8(a) と Fig. 8(b) に示すように、テクスチャが少ない deskW と refrig では、認識に失敗した。この例では、正しくマッチした点は、deskW で 2 個、refrig で 0 個であった。Fig. 8(c) に示すように、small table はテクスチャが比較的多いため、12 個の点が正しくマッチして認識に成功した。

次に、照明変化に対するロバスト性を調べるために、入力画像の輝度を人為的に変化させて、二次元認識を行った。提案手法による結果を Fig. 9 に示す。同図は、Fig. 7(c) 左の画像に対して、輝度を明るくした画像、および、輝度を暗くした画像での認識結果である。マッチした二次元モデルは、いずれも Fig. 7(c) 右の画像であった。明暗どちらの画像も認識できたが、原画像

(a) deskW

(b) refrig

(c) small table

Fig. 8 Matched points in 2-D recognition using SIFT features (left: input images, right: 2-D models)

でマッチしたエッジ点数が 87 個なのに対して、明るい画像では 49 個、暗い画像では 54 個であった。Lowe の SIFT プログラムを使って同様の実験を行ったが、明暗どちらも認識に成功した。ただし、原画像では 12 個の特徴点がマッチしたが、明暗画像ではどちらも 4 個しかマッチしなかった。

特徴点抽出から二次元姿勢推定まで含めた二次元認識の全処理時間は、入力画像 1 枚につき、提案方式が 5~8 秒程度、Lowe の SIFT 法（マッチング部は提案方式）が 1 秒程度である。提案方式では、このうち、スケール空間解析とエッジ点の SIFT 特徴ベクトル生成に約 3~6 秒かかる。エッジ点は入力画像 1 枚から 5,000~15,000 個生成される。提案方式は大量のエッジ点を扱うため処理時間が多い。この改善は今後の課題である。

6.3 三次元認識

二次元認識で得られた物体モデルを用いて三次元姿勢を推定する実験を行った。Fig. 10 に、sink の三次元認識例を示す。Fig. 10(a) は、三次元エッジモデルをある視点から見た図である。Fig. 10(b) は、二次元認識で求めた相似変換パラメータ値で変換した二次元モデル（白色）を入力画像に重ね合せている。Fig. 10(c) は、三次元認識で求めたカメラ姿勢によって三次元エッジモデル（白色）を入力画像に投影して重ね合せている。

Fig. 11 に、物体データベースに登録された各物体の認識例を示す。SIFT 法で扱いにくい deskW や refrig のようにテクスチャの少ない物体も、エッジ線分を用いる方式 [17] で扱いにくい small table や ym2000 のような細かいエッジから成る物体も、うまく認識できている。

実験では、三次元姿勢推定はすべて成功した。これは、二次元認識で 50~200 個という十分多くのエッジ点がマッチしたことが要因である。処理時間は、RANSAC も含めて 0.1~0.3 秒である。RANSAC の試行を 100 回程度繰り返せば、おおむねよい結果が得られるが、乱数を用いるため、ときおり姿勢のずれた解が生じる。この対処は、今後の課題である。

三次元姿勢推定の精度には様々な要因が関係するため、現時

Fig. 9 Robustness to intensity change

Fig. 10 3-D recognition of a sink

Fig. 11 Examples of 3-D recognition

点では目視レベルの評価のみ行っている。Fig. 10 や Fig. 11 で見られるように、三次元エッジモデルと画像中の物体はよく重なっている。物体モデルの誤差が小さく、かつ、カメラ校正が正確であれば、推定姿勢は数 [cm] 程度の誤差に収まると期待される。これは、近接センサも併用すれば、移動ロボット用には十分な精度と考えられる。詳しい精度評価には、上記の RANSAC のばらつきへの対処、物体モデルの誤差評価、詳細なカメラ校正などが必要であり、今後の課題である。

7. ま と め

本稿では、エッジ点を用いて、1 枚のカメラ画像から不特定物体を認識し、かつ、三次元姿勢を推定する物体認識方式を提案した。本方式は、見かけのエッジが連続的に変化する物体でなければ、家具のようなテクスチャの少ない物体にも、細かいエッジや曲線からなる複雑な物体にも適用できる。今後の課題として、処理時間とメモリ使用量の削減、三次元姿勢推定の精度評価がある。

謝 辞 本研究は、独立行政法人科学技術振興機構の戦略的創造研究推進事業・個人型研究（さがけ）により行われた。実施にあたっては、筑波大学 油田信一教授、坪内孝司教授に、研究環境を提供していただいた。ここに感謝いたします。

参 考 文 献

- [1] J.S. Beis and D.G. Lowe: "Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces," Proc. of CVPR, pp.1000-1006, 1997.
- [2] J. Canny: "A Computational Approach to Edge Detection," IEEE Trans. PAMI, vol.8, no.6, pp.679-698, 1986.
- [3] 出口光一郎: 画像認識論講義, 昭見堂, 2002.
- [4] M. Fischler and R. Bolles: "Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis

and Automated Cartography," Communications ACM, vol.24, pp.381-395, 1981.

- [5] R.M. Haralick, C. Lee, K. Ottenberg and M. Nolle: "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem," International Journal of Computer Vision, vol.13, no.3, pp.331-356, 1994.
- [6] D.P. Huttenlocher and S. Ullman: "Recognizing Solid Objects by Alignment with an Image," International Journal of Computer Vision, vol.5, no.2, pp.195-212, 1990.
- [7] Y. Lamdan, J.T. Schwartz and H.J. Wolfson: "On Recognition of 3-D Objects from 2-D Images," Proc. of ICRA'88, pp.1407-1413, 1988.
- [8] D.G. Lowe: "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. of Computer Vision, vol.60, no.2, pp.91-110, 2004.
- [9] <http://www.cs.ubc.ca/spider/lowe/keypoints/siftDemoV4.zip>
- [10] T. Lindberg: "Edge Detection and Ridge Detection with Automatic Scale Selection," Int. J. of Computer Vision, vol.30, no.2, pp.117-154, 1998.
- [11] K. Mikolajczyk and C. Schmid: "An affine invariant interest point detector," Proc. of ECCV2002, pp.128-142, 2002.
- [12] K. Mikolajczyk, A. Zisserman and C. Schmid: "Shape recognition with edge-based features," Proc. of BMVC2003, pp.779-788, 2003.
- [13] H. Murase and K. Nayar: "Visual Learning and Recognition of 3-D Objects from Appearance," International Journal of Computer Vision, vol.14, no.1, pp.5-24, 1995.
- [14] K. Ohba, Y. Sato and K. Ikeuchi: "Visual Learning and Object Verification with Illumination Invariance," Proc. of IROS'97, pp.1044-1050, 1997.
- [15] J. Shi and C. Tomasi: "Good Features to Track," Proc. of CVPR'94, pp.593-600, 1994.
- [16] C. Tomasi and T. Kanade: "Shape and Motion from Image Streams under Orthography: A Factorization Approach," Int. J. of Computer Vision, vol.9, no.2, pp.137-154, 1992.
- [17] 友納, 油田: "不正確さを許すマップと単眼ビジョンによる物体認識に基づく移動ロボットの屋内ナビゲーション", 日本ロボット学会誌, vol.22, no.1, pp.83-92, 2004.
- [18] 友納: "画像列からの密な物体モデル生成のためのエッジの 3 次元復元", 日本ロボット学会第 23 回学術講演会予稿集, 3B11, 2005.
- [19] 友納: "形状情報と認識特徴を統合した物体モデルによる 3 次元物体マップの構築", 第 11 回ロボティクスシンポジウム予稿集, pp.246-251, 2006.
- [20] T. Tuytelaars and L. Van Gool: "Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions," Proc. of BMVC2000, pp.412-425, 2000.
- [21] 山崎, 友納, 坪内, 油田: "移動ロボット搭載のカメラを用いた物体の三次元形状モデリング", 日本ロボット学会誌, vol.23, no.7, pp.868-877, 2005.

友納正裕 (Masahiro Tomono)

1960 年 5 月 10 日生。1985 年東京大学大学院工学系研究科修士課程修了。同年 NEC 入社。1999 年退社。2002 年筑波大学大学院博士課程修了。2002～2006 年科学技術振興機構さがけ研究者。2006 年東洋大学。現在に至る。ロボットの環境認識、行動制御、ソフトウェアアーキテクチャなどに興味を持つ。情報処理学会、IEEE の会員。(日本ロボット学会正会員)