

タイトル	SIFTを用いた特定物体認識の高速化と移動ロボットによる物体ハンドリングへの応用
著者	塩濱, 教幸; 深谷, 健一; Shiohama, Noriyuki; Fukaya, Ken-ichi
引用	工学研究 : 北海学園大学大学院工学研究科紀要(12): 25-30
発行日	2012-09-28

SIFT を用いた特定物体認識の高速化と 移動ロボットによる物体ハンドリングへの応用

塩濱 教幸*・深谷 健一*

Speed-up of specific object recognition using SIFT
and application to the object handling by Mobile Robot

Noriyuki Shiohama* and Ken-ichi Fukaya*

概要

最近、画像認識により特定の物体を検出する有効な手法として SIFT (Scale-Invariant Feature Transform) が使われている。しかし、大画像から SIFT 特徴量を抽出するには時間がかかるので、物体を検出するのが遅くなるという問題がある。そこで本研究では特定物体認識を高速化するために、画像を分割する方法を考案した。画像分割した理由は、特定物体検出を行うためには大画像全体から SIFT 特徴点を検出する必要はなく、分割画像を順番に特徴点検出し、マッチングすると高速化できると考えたためである。しかし、適当に順番を決めると高速化できない場合があるので、その順番が重要となる。本手法により、その有効性を確認でき、物体検出を高速化することが出来た。応用として移動ロボットによる物体ハンドリングを試みた。

1. はじめに

近年、ロボットの利用は産業用から民生用へと広がっており、人間に代わってロボットにある特定の作業をさせるという研究が行われている。例題として、移動ロボットを使った工具ハンドリングの研究¹⁾や、掃除用ロボット「Roomba (iRobot 社)」、「Smarbo (東芝)」などが商品化されている。本研究では SIFT 画像処理を用いて移動ロボットが物体認識をして対象物を発見し、それをハンドリングさせる実験を行った。

物体認識する際に SIFT を用いると、カメラと対象物との距離が遠い場合、撮影画像に写っている対象物から検出される SIFT 特徴点の数が少なく、認識の成功率が低下する問題がある。撮影する画像の解像度を大きくすると検出される特徴点の数が多くなり、距離の影響は軽減されるが、SIFT の処理時間が長くなってしまい対象物の検出が遅くなる。

対象物の検出を高速化する手法として、撮影画

像を分割することで画像サイズを小さくし、その画像を順番に SIFT 処理しマッチングすることによって対象物検出の際にかかる SIFT 処理の時間を短縮するという方法を提案する。

2. 物体認識

物体認識とは、画像に写っているものが何かを判定する画像処理であり、大きく分けると、一般物体認識と特定物体認識の2種類に分けられる。一般物体認識とは、制約なしに画像を撮影し、その画像に写っている物体は何であるかを一般的な名前でも認識させることである。つまり、色や形などが違う自動車であっても、同じ「自動車」と認識させることである。一方、特定物体認識とは、特定の物体が画像中に写っているかを認識させることである。

今回の研究では特定物体認識を行い、対象物をペットボトルとして、特定のペットボトルを認識させている。

* 北海学園大学大学院工学研究科電子情報工学専攻

Graduate School of Engineering (Electronics and Information Eng.), Hokkai-Gakuen University

2.1 SIFT について

SIFT は、画像中の特徴点の検出と特徴量の記述を行うアルゴリズムであり²⁾、検出した特徴点に対して、画像のスケール変化・回転・照明変化等にロバストな特徴量を記述する。そのため、物体認識や物体検出、画像マッチングなどに用いられる³⁾。最近では、PCA-SIFT などという主成分分析を用いた SIFT があり、これは従来の SIFT に比べると精度が高くなっている。

本研究では SIFT のアルゴリズムを作った David G. Lowe が OpenCV を利用した SIFT のソースコードを公開している⁴⁾ので、それを利用して物体認識を行っている。

ここで、「特徴点の検出」と「特徴量の記述」について、それぞれの内容を以下に示す。

特徴点の検出：

1. スケールとキーポイント検出

DoG (Difference of Gaussian) 画像から特徴点を検出するという処理で、特徴点はスケール変化に対して不変になる。DoG 画像とは、スケールの異なるガウス関数と入力画像を畳み込んだ平滑化画像の差分である。

2. キーポイントのローカライズ

主曲率を求め、しきい値処理をすることによって、1で検出したキーポイントの内、エッジ上の点を削除する。また、サブピクセル位置での DoG 出力値の小さい点、つまりノイズや開口問題に影響を受けやすい点を削除する。

特徴量の記述：

3. オリエンテーションの算出

特徴量記述の際にオリエンテーションにより向きの正規化を行うことで、特徴点は回転に対して不変になる。オリエンテーションを求めるためには、勾配強度と勾配方向を求め、重み付方向ヒストグラムを作成することが必要である。

4. 特徴量の記述

128次元の各特徴ベクトルの長さをベクトルの総和で正規化することで、特徴点は照明変化に対して影響が小さくなる。

SIFT を用いた画像処理の例として、対象物をペットボトルとし、対象物が回転した場合と対象

物が回転し、さらに照明変化した場合の画像マッチングの様子を比較し、図1に示す。

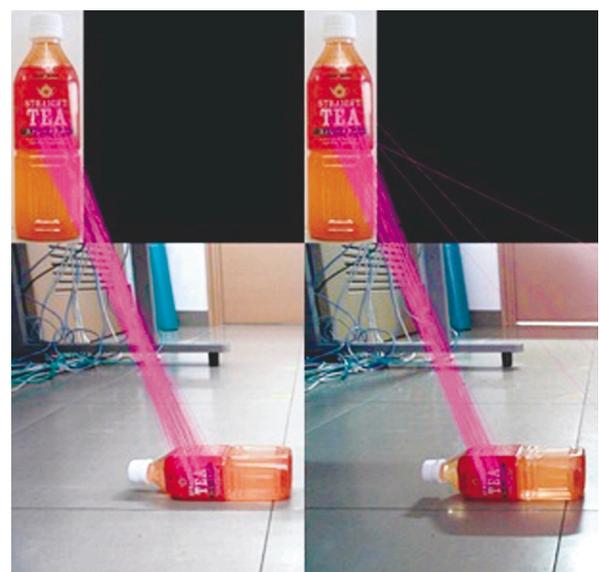
これらを比較した結果、照明変化した場合の方が一致した特徴点の数が少なくなり、また外れ値も出てきてしまったということが分かった。

SIFT の欠点は、例えば、対象物が模様などのないただのボールであった場合、検出される特徴点の数が極めて少なくなるという点である。

2.2 特定物体認識の高速化

前述したとおり、撮影画像をそのまま処理すると対象物の検出が遅いので高速化するために、分割した撮影画像を SIFT 処理しマッチングしていく。まず、どの分割画像から SIFT 処理するか順番を決める必要がある。この順番によって処理の速度が決まるので、その決め方が重要である。順番を決めるために、まず撮影画像を前処理する。以下にその処理の手順を示し、実際に処理した画像を図2に示す。

1. 撮影画像を白黒変換、エッジ抽出し、輪郭線を描画する。
2. オープニング・クロージング処理を使用し、輪郭線を描画した画像のノイズを除去する。
3. ノイズ除去した画像を8分割し、8分割したそれぞれの画像に対して、画素値が白となっている画素の数 (white_pix) を求める。



対象物が回転した例 回転し照明変化した例

図1 SIFT を用いた画像マッチングの例

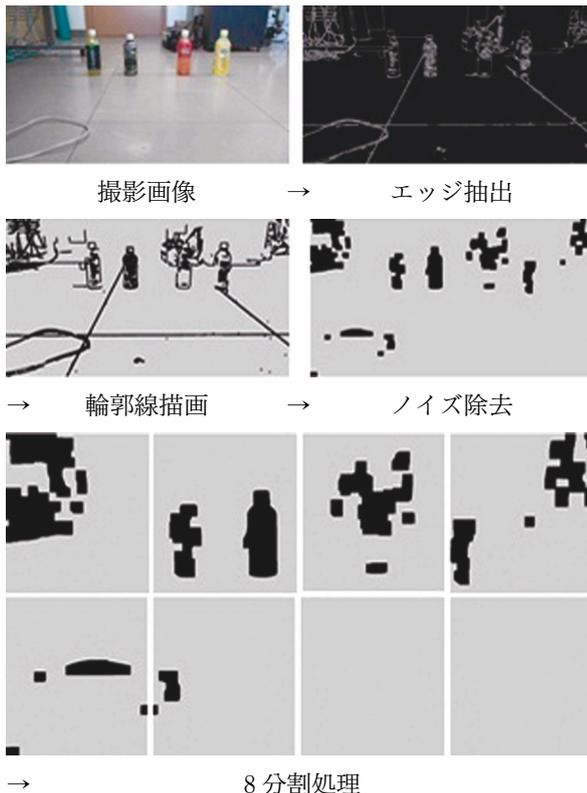


図2 前処理の手順

4. white_pix が最小の画像に対応する分割撮影画像を SIFT 処理する。理由は、SIFT 特徴点が多い分割画像では白画素が少ないと想定されるからである。
5. SIFT 処理済の参照画像 (対象物の画像) とマッチングし、外れ値除去を行った結果、一致した特徴点の数が 15 個以上ならば対象物の検出を成功とする。
6. 一致した特徴点の数が 15 個未満だった場合は、その画像には対象物が見つからなかったとして、white_pix が次に小さかった画像に対応する分割撮影画像とマッチングする。

1～4 の前処理をする理由は、検出される SIFT 特徴点がエッジの近くにあるということが分かったので、エッジを多く含んでいる場所から探していくと高速化できると考えたためである。

5～6 の処理を 8 回繰り返しても対象物が見つからなかった場合は対象物の検出を失敗とする。マッチングしている様子を図 3 に示す。

図 3 は左から順に white_pix が小さかった画像である。左の画像には一致する点が 2 つしかなく、真ん中の画像には一致する点が 3 つしかなかった。右の画像には一致する点が多くあるので、



図3 マッチングの様子

画像に対象物があったという結果が得られる。

2.3 実験結果

ペットボトルを床に 4 つ並べて、撮影画像を 1 分割、4 分割、8 分割の場合の 3 通り、カメラからの距離を 0.5 m、1.0 m、1.5 m の 3 通り、カメラの解像度を 1280×720、1920×1080 の 2 通りで実験し、物体認識にかかった時間を計測した。また同時に一致した特徴点の数も計測した。それらを比較した結果を解像度が 1920×1080 の場合は表 1 と図 4、1280×720 の場合は表 2 と図 5 に示す。使用した機器については後述する。

最初に述べたように距離が遠くなると一致した特徴点の数が少なくなっていくことがわかる。そして、分割しない場合と比べ、8 分割した場合が

表1 マッチングの実験結果 (1920×1080)

カメラからの距離	分割数	SIFT 計算時間 (s)	一致した数 (個)※	高速化 (倍)
0.5 m の場合	1	12.92	105 → 104	1
	4	7.11	82 → 77	1.81
	8	4.17	85 → 80	3.09
1.0 m の場合	1	13.37	42 → 37	1
	4	8.61	45 → 40	1.55
	8	8.22	47 → 46	1.62
1.5 m の場合	1	12.18	29 → 24	1
	4	9.73	27 → 26	1.25
	8	7.54	34 → 29	1.61

※

→前の数は一致した全部の特徴点の数

→後の数は外れ値除去して残った特徴点の数

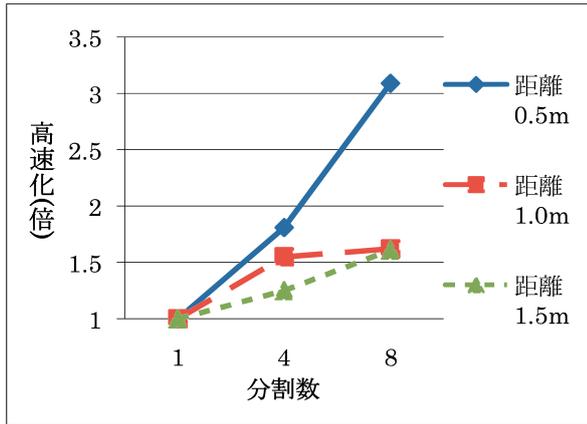


図4 分割による高速化の比較（1920×1080）

物体認識にかかる時間が一番短くなり、一致した特徴点の数が多くなった。

また、撮影画像の解像度を小さくした場合（1280×720）は、SIFT 処理時間は全体的に短くなったが、距離が1.0 m の場合に遅くなってしまった場合もあった。カメラからの距離が1.5 m の場合は一致した特徴点の数が少なかったため対

表2 マッチングの実験結果（1280×720）

カメラからの距離	分割数	SIFT 計算時間 (s)	一致した数 (個)※	高速化 (倍)
0.5 m の場合	1	8.1	56 → 54	1
	4	3.71	45 → 39	2.18
	8	2.25	40 → 39	3.6
1.0 m の場合	1	6.55	31 → 25	1
	4	6.72	29 → 26	0.97
	8	5.35	37 → ?	1.22
1.5 m の場合	1	7.41	10 → 7	
	4	6.91		
	8	7.55		

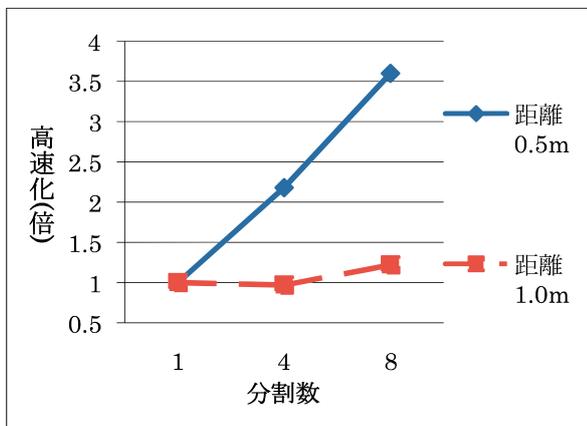


図5 分割による高速化の比較（1280×720）

象物の検出が出来なかった。

2.4 画像分割により起こると予測される問題

画像を分割することによって対象物も分割されてしまう場合がある。この時、どこかの分割画像に、一致する特徴点の数が15個以上あれば検出成功となるが、すべての画像で一致する特徴点の数が15個未満であると検出失敗となってしまいう問題がある。

この問題の対策としては、画像を分割する際にそれぞれの画像を少し大きめに分割するという方法がある。例えば撮影画像のサイズが1920×1080の場合は、それぞれの分割画像の大きさは480×520であるが、それを520×520にするということである。図6、図7に例を示す。これにより、分割されていたペットボトルの分割されない画像を作られる。この場合は、対策をしないと約1.5倍高速化でき、対策をすると約1.4倍処理を高速化できた。処理時間を比較すると少し遅くなっているが、大きな影響がないということがわかる。

今回の実験では対象物の分割によって検出できないということは起こらなかった。その理由は、対象物までの距離が近い場合は全体の特徴点の数が増えるので、特徴点の数が15個未満にならない。



図6 480×480の8分割画像



図7 520×520の8分割画像

よって、その問題は発生するのは、距離が離れている場合と考えれば、それは分割による問題ではなく、距離の問題となるからである。

3. 移動ロボットによる物体ハンドリング

提案した手法を使い、移動ロボットによる物体ハンドリングを行った。使用した機器、実験システムの構成とロボットの全景を表3、図8、図9に示す。

移動ロボットの処理の内容について述べる。まず、指定した位置まで移動し、SIFT 画像処理を行って対象物を検出する。次に、測域センサを用いて対象物までの距離を測り近づき、掴む。最後に、ロボットが最初にいた位置まで対象物を運ぶという流れである。ここで、本実験の対象物はペットボトルである。

3.1 測域センサ

測域センサとは、空間の形状を測定し、そのデータを出力する光走査式距離センサである。今回の実験では単眼カメラを使っているため、画像処理によって対象物を見つけられても、対象物までの

表3 使用機器

自律移動ロボット	品名：	Pioneer3-DX
ノート PC	OS：	Windows7
	CPU：	Core i5-2540M @ 2.60GHz
カラー単眼カメラ	品名：	Life Cam Studio
	解像度：	1920×1080
測域センサ	型式：	URG-04LX-UG01
	検出範囲：	距離 4 m, 角度 240°

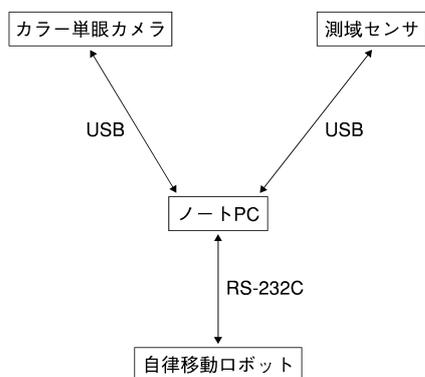


図8 物体ハンドリングの実験システムの構成

距離がわからないので、ハンドリングすることが出来ない。対象物までの距離を測定するために測域センサを用いている。

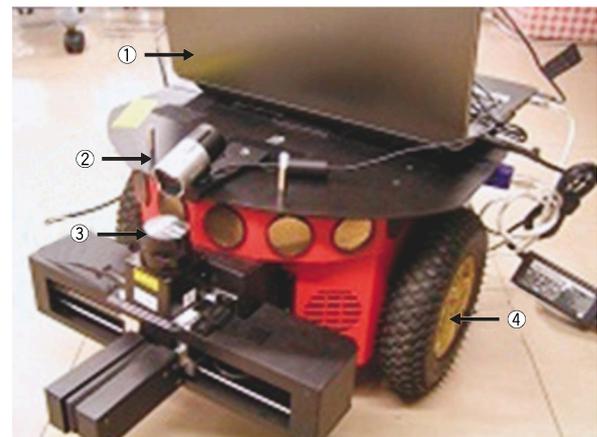
3.2 実験結果

指定したペットボトルの検出に成功し、8分割処理しない場合と比べると、最初にいた位置まで早く運ぶことが出来た。その時間経過を8分割した場合と8分割しない場合とで比較し、図10と図11に示す。

ペットボトルを運び終わるまでの時間の差を比べると、ほぼ画像処理にかかった時間の差と一致する。

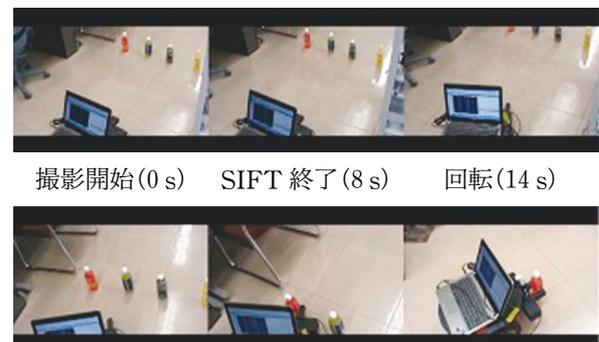
4. まとめ

今回 SIFT を用いた特定物体認識の高速化手法として、撮影画像を分割する方法を提案した。



①ノート PC ②カラー単眼カメラ
③測域センサ ④自律移動ロボット

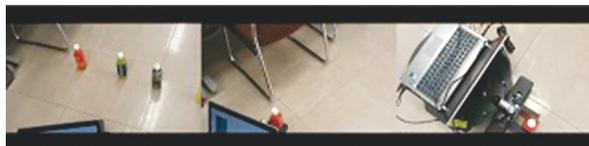
図9 ロボットの全景



移動(18s) 掴む(25s) 戻る(31s)
図10 ハンドリングの時間経過写真 (8分割)



撮影開始 (0 s) SIFT 終了 (15 s) 回転 (22 s)



移動 (26 s) 掴む (34 s) 戻る (40 s)

図 11 ハンドリングの時間経過写真（分割なし）

その結果、実験によりこの手法の有効性を確認できた。

また、本実験では対象物をペットボトルとしているが、ペットボトル以外の対象物でも同じようにハンドリング出来る可能性がある。

今後は測域センサをさらに活用したいと考えている。つまり、距離を測るセンサとして使うだけでなく、例えば、3次元計測を出来るようにし、そのデータを使い対象物の形状を認識することで対象画像の SIFT 画像処理する領域を狭め、画像処理時間を短縮出来るようにする。また、対象物

の存在する位置が正確にわからない場合でも、自己位置認識を併用することで、対象物のおおよその位置を見つけ、その後で SIFT 画像処理によりハンドリングを行うことで機能の汎用性を高めていく。

謝辞

本研究は、文部科学省私立大学戦略的研究基盤形成支援事業（平成 20 年～平成 24 年）の支援を受けて行われた。

【参考文献】

- 1) 木下和樹, 油田信一: 画像中の複数の特徴を用いた片付けロボットのための工具検出, 第 29 回日本ロボット学会学術講演会, 102-7, 2011.
- 2) D. G. Lowe: "Object recognition from local scale-invariant features", Proc. of IEEE International Conference on Computer Vision (IJCV), 60(2), pp.363-370, 1984.
- 3) 藤吉弘亘: Gradient ベースの特徴抽出—SIFT と HOG—, 情報処理学会研究報告 CVIM 160, pp.211-224, 2007.
- 4) <http://blogs.oregonstate.edu/hess/code/sift/>