

## 多重サイズ局所記述子の GPU による抽出

市村 直幸<sup>†</sup>

<sup>†</sup> 産業技術総合研究所

〒 305-8568 茨城県つくば市梅園 1-1-1 中央第 2

E-mail: [†nic@ni.aist.go.jp](mailto:†nic@ni.aist.go.jp)

あらまし 近年, 局所不変特徴量は, 画像の対応付けや画像認識における基盤要素として幅広く用いられている. この特徴量は, (1) 画像内での局所領域の設定, (2) 局所領域内部の画像特徴を表す記述子の計算, の 2 段階の処理を通じて抽出される. 局所領域の大きさは, その中心となる特徴点やエッジ等の固有スケールに係数を乗じて決定できる. 通常, その係数は固定されている. しかし, 固定された大きさをもつ 1 つの局所領域から記述子を抽出した場合, 特徴量の隠れへの耐性, 不変性, 識別性の間のバランスをシーン内容に応じて調整することが困難な場合がある. 本論文では, 固有スケールに複数の係数を乗じ, 大きさの異なる複数の局所領域から記述子を抽出する方法を検討する. この方法で抽出される記述子を, 多重サイズ局所記述子と呼ぶ. 多重サイズ局所記述子を用いることにより, 複数の空間的な範囲を用いて, 発生頻度や共起, 相関等に基づく特徴表現が可能となる. このことは, 上記のバランスを制御する有力な手段となると考えられる. 多重サイズ局所記述子の問題点として, 特徴抽出に要する計算量の増加がある. この問題の解決のために, 輝度勾配の方向ヒストグラムに基づく多重サイズ局所記述子を対象とし, GPU(Graphics Processing Unit) による並列処理を用いて特徴抽出を行う. 特に, 方向マップを用いることにより, 多重サイズ局所記述子を高速に抽出できることを示す.

キーワード 局所不変特徴量, GPU, 多重サイズ解析, 対応付け, 物体認識

## Extracting Multi-Size Local Descriptors by GPU Computing

Naoyuki ICHIMURA<sup>†</sup>

<sup>†</sup> National Institute of Advanced Industrial Science and Technology (AIST)

Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki, 305-8568 Japan

E-mail: [†nic@ni.aist.go.jp](mailto:†nic@ni.aist.go.jp)

**Abstract** Local invariant features have been widely used as fundamental elements for image matching and object recognition. Local invariant features can be extracted by the following two steps:(1)detecting local regions, (2)calculating descriptors. The size of each local region is determined by multiplying the characteristic scale of the center pixel, which may correspond to a feature point and an edge etc., by a certain factor. The factor is normally fixed. However, if a descriptor is calculated from only a single local region, it is difficult to adjust the balance among the robustness against for occlusions, the invariance, and the distinctiveness of the descriptor to the contents of scenes. The purpose of this paper is to develop a method for extracting descriptors from multiple local regions with different sizes which are determined by multiplying multiple factors to the characteristic scale. The descriptors obtained from multiple local regions are called *multi-size local descriptors*. Multi-size local descriptors enable us to use various types of representation such as frequency, co-occurrence and correlation based on many different spatial sizes, which is a promising way to control the balance mentioned above. Because multi-size local descriptors increases the computational costs of feature extraction, we develop parallel computational techniques for extracting multi-size local descriptors consisting of the histograms of gradient orientations through the use of a graphics processing unit (GPU). In particular, we demonstrate that orientation maps are useful for efficient extraction of the multi-size local descriptors.

**Key words** local invariant features, GPU, multi-size analysis, image matching, object recognition

## 1. まえがき

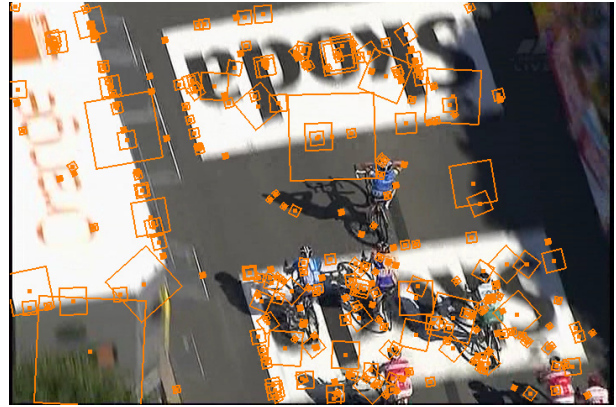
局所不変特徴量は、画像の対応付けや物体認識のために、シーン内容を表現する有力な方法である。この特徴量は、(1) 画像内での局所領域の設定、(2) 局所領域の画像特徴を表す記述子 (descriptor) の計算、の2段階の処理を通じて抽出される [1] ~ [4]。図 1(a) に局所領域の例を示す。図中の正方形が局所領域を表す。このような局所領域で計算された記述子は、局所領域内部の輝度やテクスチャ、エッジ等に基づいて画像特徴を数値化したものであり、多くの場合、ベクトルの形態をとる。

局所不変特徴量には、主として2つの利点がある。1つは、局所領域を用いることによる、視野逸脱や遮蔽による隠れへの耐性である。シーンの一部に隠れが生じて、見えている局所領域の特徴量が使用できる。もう1つの利点は、特徴量に不変性を付与できることである。スケールスペースピラミッドの利用や局所座標系の導入等を通じ、画像の幾何学的変換や輝度変化に対し特徴量が不変になるように、上記 (1),(2) の特徴抽出処理を構成できる。これらの利点から、隠れや視点の移動、照明条件等の違いにより、基準画像からの見えの変化がシーンにおいて生じたとしても、その変化の影響を軽減し、シーンから基準画像と同様の特徴量を得ることができる。

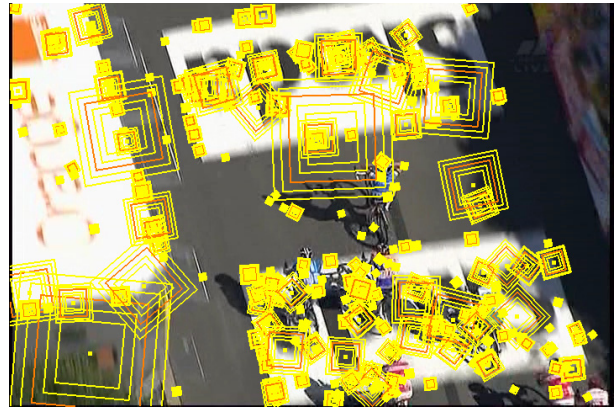
上記の利点を有し、かつ、特徴量としての重要な性質である識別性も、主として画像特徴のヒストグラムにより得ることができる。そのため、局所不変特徴量は、画像の対応付けや物体認識の基盤要素として幅広く用いられている [1] ~ [13]。

局所領域の設定は、スケールスペースピラミッド内部における局所領域の位置と大きさの決定を通じて行われる。局所領域の中心画素の位置は、特徴点やエッジ等の画像特徴に基づき決定する場合と、グリッドやランダム選択等を用い画像特徴とは無関係に決定する場合がある [6], [8], [12]。局所領域の大きさは、中心画素がスケールスペースピラミッドのどのスケールに属するか、つまり、中心画素の固有スケール [15] に基づいて決定される。具体的には、中心画素の固有スケールに係数 (以後、スケール係数と呼ぶ) を乗じて、局所領域の大きさを決定できる。例えば、図 1(a) において局所領域の大きさが異なるのは、その中心画素の固有スケールが異なるからである。このように固有スケールに基づいて局所領域の大きさを決定することにより、シーンでスケール変化が生じたとしても、基準画像と物理的に同じ範囲をカバーする局所領域を設定することが出来る。よって、特徴量はスケール不変性を有することになる。

局所領域の大きさは、特徴抽出に用いる空間的な範囲を決定するため、画像特徴表現に使用可能な情報量を限定する重要な一要素である。よって、局所領域の大きさは、特徴量の隠れへの耐性、不変性、識別性に大きな影響を与える。例えば、図 2 に示す 2 枚の画像から局所不変特徴量を抽出することを考える。図 2(a) の画像 graffiti には、多くの明暗、色、形状の変化が含まれるため、局



(a)



(b)

図 1: 画像 Tour de France [14] における局所領域の設定例。局所領域の大きさは、その中心画素の固有スケールに係数を乗じて決定する。(a) 単一の係数を用いた場合。(b) 複数の係数を用いた場合。この例に示すような大きさの異なる複数の局所領域から抽出される記述子を、多重サイズ局所記述子と呼ぶ。この図では表示の簡明さのため、局所領域は総数の 1/20 のみを示している。

所領域を小さくし隠れへの耐性を高くしても、特徴量に十分な不変性と識別性を付与できる可能性は高い。しかし、図 2(b) の画像 Škoda には、graffiti の有するような変化が乏しいため、より大きな局所領域を使用しないと、特徴量に不変性や識別性を付与することが困難と考えられる。この例からもわかるように、本来、シーン内容に基づいてスケール係数を変化させ、特徴量の隠れへの耐性、不変性、識別性の間のバランスを制御することが望ましいと言える。しかし、ある特定の性質を有するシーンを取り扱うという前提がない限り、局所領域の大きさをシーン毎に制御することは容易とは言えない。現状では、通常、固定された1つのスケール係数を使用し、シーン内容によらず単一の大きさの局所領域から記述子を抽出している。

本論文では、固有スケールに複数の係数を乗じ、大きさの異なる複数の局所領域から記述子を抽出する方法を検討する。この方法で抽出される記述子を、多重サイズ局所記述子と呼ぶ。図 1(b) は、多重サイズ局所記述子を抽出するための局所領域の例である。この例では、それぞれの中心画素において、図 1(a) の局所領域に対し、



図 2: シーン内容が異なる画像の例. (a) 画像 graffiti [16]. (b) 画像Škoda [17]. Graffiti には多くの明暗, 色, 形状の変化が含まれるが, その一方でŠkoda ではそのような変化が乏しい. 局所不変特徴量を抽出する際, 隠れへの耐性, 不変性, 識別性の間のバランスを制御するために, このようなシーン内容の違いに基づき局所領域の大きさを変化させることが望ましい.

より大きさの小さいもの大きなものをそれぞれ 2 つずつ加え, 合計で 5 つの局所領域を設定している. このような複数の空間的な範囲から抽出される多重サイズ局所記述子を用いると, シーンに関する事前情報がない場合にも, 特徴量の隠れへの耐性, 不変性, 識別性の間のバランスの制御が可能となると考えられる. 例えば, 隠れへの耐性を優先にする場合には, 最も小さな局所領域から抽出される記述子のみを利用すれば良い. よって, 単一の大きさの局所領域を用いる場合に比べ, 多重サイズ局所記述子には機能的な優位性があると言える.

上記のような複数の空間的な範囲から記述子を抽出することは, スケール空間ピラミッドの構成そのものでも一部は解決可能である. 例えば, 1 つのスケール係数を使用して, ある中心画素に対し全てのスケールにおいて局所領域を設定すれば, おのずと大きさの異なる複数の局所領域から記述子を抽出することができる. しかし, この場合, 空間的な範囲の変化と共に画像情報のスケールや分解能の変化が伴う. その一方で, 本論文で取り扱う多重サイズ局所記述子では, 空間的な範囲の変化とスケールと分解能の変化は分離可能である点に注意されたい. これにより, ある特定のスケールに属する特徴点やエッジ等を中心画素として局所領域を設定する場合にも, 空間的な範囲の変化を利用した画像特徴表現が可能となる.

多重サイズ局所記述子の関連研究として, Cheng らの研究 [10] が挙げられる. この研究では, 非剛体である物体の形状変化や魚眼レンズ等の光学系, 画像処理等に由来する大きな見えの変化の下においても, 対応付けが可能となるように, 大きさの異なる複数の局所領域から輝度勾配の方向ヒストグラムに基づく記述子を抽出している. 局所領域の位置の決定は, Harris 行列を用いた特徴点抽出により行われている. 特徴点抽出は疎 (sparse) な局所領域設定方法であるため, 特徴点数は多くないものの, 大きさの異なる複数の局所領域を用いるため記述子

の計算量は少ないものとは言えない. Cheng らの論文では, 多くの局所領域から特徴抽出を行うには計算パワーが必要だとの言及はあるが, その対策については述べられていない. しかし, 特徴抽出の高速化に関し, 何らかの対策を立てることは非常に重要な研究課題と言える. 何故ならば, 近年, 対応付けや物体認識において, 特徴点抽出に比べより密 (dense) に局所領域を設定することの有効性が示されているからである [6]~[8], [12], [18]~[22]. 局所領域を密に設定し, なおかつ, 多重サイズ局所記述子を抽出するには, 特徴抽出の高速化を行うことが必須と考えている.

上記の課題を解決するために, 本論文では, 輝度勾配の方向ヒストグラムに基づく多重サイズ局所記述子を対象とし, GPU (Graphics Processing Unit) による並列処理を用いて特徴抽出を行う. 特に, 方向マップと呼ばれるデータ構造 [9] を用いることにより, 効率的に多重サイズ局所記述子が抽出できることを示す. 本研究の新規性として, まず, GPU による多重サイズ局所記述子の高速な抽出方法を具体的に提示することが挙げられる. また, 他の新規性として, 大きさの異なる複数の局所領域を用いることにより, 特徴量の隠れへの耐性, 不変性, 識別性の間のバランスをとる機能が得られることに言及していることも挙げられる. これは, 大きさの異なる複数の局所領域を用いることの有効性が, 物体の変形への対処という点で述べられていた Cheng らの研究 [10] とは違う視点を与えるものと考えている.

以下, 2. 節において輝度勾配の方向ヒストグラムに基づく記述子の説明を行う. 3. 節では, 多重サイズ局所記述子の機能と, その方向マップを用いた計算について述べる. 4. 節で特徴抽出処理の GPU による実装について述べ, 5. 節で計算時間の測定結果を示す. 最後に, 6. 節で本研究をまとめる.

## 2. 輝度勾配の方向ヒストグラムに基づく記述子

本節では, 大きさの異なる複数の局所領域から抽出する, 輝度勾配の方向ヒストグラムに基づく記述子について説明する. 輝度勾配の方向ヒストグラムに基づく記述子には, SIFT [2], GLOH [3], HOG [7], DAISY [9] 等があるが, 基本部分には, 以下に述べる処理が含まれている.

まず, 多重解像度解析を行うために, 入力画像からスケール空間ピラミッドを生成する. 局所領域の設定に特徴点を使用する場合には, ダウンサンプリングと LoG (Laplacian of Gaussian) フィルタ等を組合せ, スケール空間ピラミッドを生成する [2], [15]. そして, スケール空間での極値点を探索し, 特徴点の位置と固有スケールを得る. その特徴点の位置を中心とし, 固有スケールに係数 (スケール係数) を乗じて得られる大きさを有する局所領域を画像内に設定する. 局所領域の設定には, 特徴点以外にエッジ等も利用される [11], [19], [23]. 図 1 が局所領域の設定例である. また,



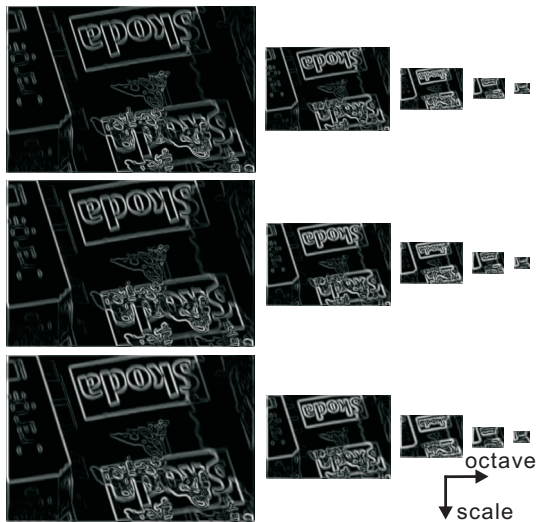


図 3: 多重解像度エッジ画像の例。各画像に対し図 5 に示すような方向マップを作成し、多重解像度方向マップを得る。この図の場合、オクターブ数は 5、各オクターブに 3 つのスケールがある。よって、輝度勾配の方向を 8 つに離散化した場合、多重解像度方向マップの数は、 $5 \times 3 \times 8 = 120$  枚となる。それら全てに対しガウシアンフィルタを適用し、その結果を用いて多重サイズ局所記述子の計算を行う。

輝度勾配が記述子において必要とされるため、ダウンサンプリングとガウシアンフィルタ、微分フィルタを組合せて別途スケール空間ピラミッドを生成する。図 3 にその例を示す。このようなスケール空間ピラミッドにおいて、局所領域のスケールに対応するエッジ画像を用いて記述子を計算する。

図 4 を用いて、記述子の計算方法を説明する。記述子の計算では、まず、局所領域にいくつかのセルを配置する。各セル内部の画素における輝度勾配の方向を得て、そのヒストグラムを生成する。この際、輝度勾配の方向に対応するビンには、セル内部の画素のもつ輝度勾配の大きさに基づいた値を投票する。そして、全セルの輝度勾配の方向ヒストグラムを連結したベクトルを構成する。このベクトルに対しノルムの正規化を行ったものが、その局所領域の記述子となる。輝度勾配の方向はエッジの方向と関連があることから、この記述子が表現しているものは、セルの配置と大きさに依存した形状情報と言える。このような形状情報が、画像の対応付けや物体認識において有用であることは、多くの研究で確認されている [1] ~ [13]。

### 3. 多重サイズ局所記述子とその方向マップを用いた計算

従来、記述子の抽出は、1 つのスケール係数を使用し、単一の大きさの局所領域から行われる。これに対し、本論文では、複数のスケール係数を使用し、大きさの異なる複数の局所領域から多重サイズ局所記述子を抽出する。複数の空間的な範囲より抽出されることから、多重サイズ局所記述子の機能は多様だと思われる。その重要な機能の 1 つとして、図 2 の説明で触れたように、シーンに

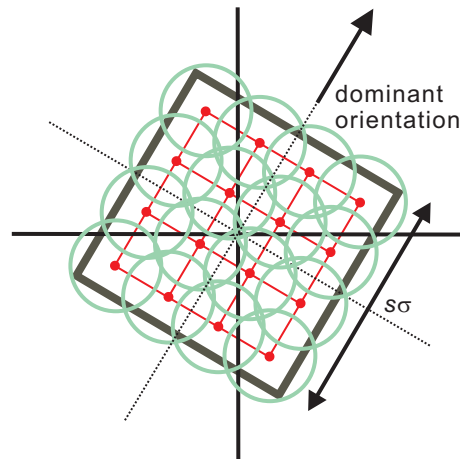


図 4: 正方形の局所領域内部におけるセルの配置。この図では、局所領域に  $4 \times 4$  のセルを配置している。小さな丸はセルの中心を、円はセルの大きさを表す。局所領域の大きさは、中心画素の固有スケール  $\sigma$  にスケール係数  $s$  を乗じて決定する。多重サイズ局所記述子の場合、スケール係数  $s$  を変化させて、図 1(b) に示すような大きさの異なる複数の局所領域を設定する。そして、各局所領域にいくつかのセルを配置し、各セル内部のヒストグラムを連結して特徴量を構成する。

関する事前情報がない場合においても、特徴量の隠れへの耐性、不変性、識別性の間のバランスを制御することがある。例えば、隠れへの耐性を優先にする場合には、最も小さな局所領域から抽出される記述子のみを利用すれば良い。識別性を向上させたければ、複数の記述子を用い、マッチングや新たな特徴量の計算を行うことが有効である。このことは、対応付けや物体認識において利用されている画像特徴の共起 [24] ~ [28] を導入する一手段と考えられる。また、不変性を優先するのならば、全てのサイズ間で総当たりの記述子のマッチングを行い、見えの変化によって生じる局所領域設定の誤差を吸収する機能を持たせることも可能である [10]。これらのことから、単一の大きさの局所領域から抽出される記述子に比べ、多重サイズ局所記述子には機能的な優位性があると言えよう。

多重サイズ局所記述子の問題点の 1 つとして、その抽出に要する計算量の増加が挙げられる。本論文では、多重サイズ局所記述子を効率良く計算するために、方向マップ [9] と呼ばれるデータ構造を利用する。以下に、方向マップを用いた計算方法を説明する。

スケール空間ピラミッドの全画素での輝度勾配の方向と大きさは、微分フィルタを適用した時点で計算できる。よって、多重解像度エッジ画像が生成された時点で、各画素の輝度勾配の方向がヒストグラムのどのビンに投票されるかが定まる。このことから、各エッジ画像に対して、ヒストグラムのビンに対応する複数の 2 次元配列を用意し、離散化された輝度勾配の方向毎に輝度勾配の大きさを保持することができる。このヒストグラムのビンに対応する 2 次元配列を方向マップと呼ぶ [9]。図 5 は、その方向マップの例である。

方向マップに対してガウシアンフィルタを適用するこ

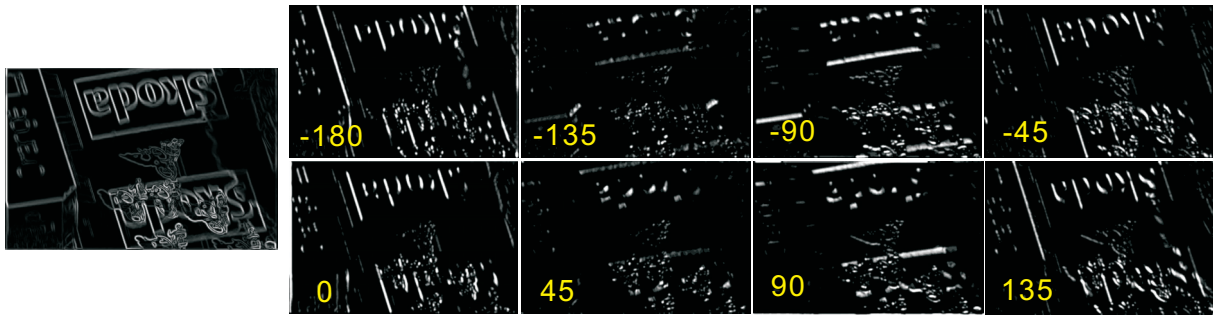


図 5: 方向マップの例. この例では, 左に示す 1 枚の微分フィルタの処理結果から, 右に示す 8 つの輝度勾配の方向に対応する方向マップを生成している. 各方向マップは, 輝度勾配の大きさを保持している. これらの方向マップにガウシアンフィルタを適用することは, ガウシアンフィルタのスケールに応じた大きさをもつセル内部で, 輝度勾配の大きさの重みつき和, つまり, ヒストグラムへの投票値を求めることと等価となる.

とは, フィルタのスケールに応じた範囲で, 輝度勾配の方向の発生頻度を輝度勾配の大きさとガウス関数により定まる重み付きで求めることに相当する. よって, フィルタのスケールを, 記述子を求める際に用いるセルの大きさに相当するように設定すれば, フィルタリングはセル内の画素を用いたヒストグラムへの投票値の計算と等価になる. そのため, 畳み込み済みの方向マップが得られれば, そのマップにおいてセルの中心位置での値を参照するだけで, 輝度勾配の方向ヒストグラムが構成できる.

一般に, 局所領域を密に設定すると, 局所領域の重なり部分が増加する. よって, 従来の方​​法のように, 記述子の計算を行う場合にセルの全画素の輝度勾配の大きさと方向を参照しヒストグラムを構成すると, 局所領域の重なり部分での計算の重複が多数生じる. しかし, 方向マップを用いると, セルの中心画素の投票値のみ参照すればよいため, 局所領域の重なり部分での計算の重複が削減されることになる. よって, 特徴抽出の計算時間が短縮される可能性がある.

上記の方向マップを用いた方法では, 方向マップを用いない場合の局所領域の重なり部分における計算量と方向マップに対するガウシアンフィルタの計算量を交換しているため, その 2 つの計算量のバランスに注意する必要がある. 局所不変特徴量を抽出する場合には, 図 3 に示すような多重解像度エッジ画像から方向マップを生成する. 生成された方向マップを多重解像度方向マップと呼ぶ. 多重解像度方向マップの数は, スケールスペースピラミッドのオクターブ数, 各オクターブのスケール数, および, 輝度勾配の方向ヒストグラムのビン数により定まる. 図 3 の場合, オクターブ数が 5, スケール数が 3 であるので, ビン数を図 5 のように 8 とすると, 多重解像度方向マップの数は 120 枚となる. このように多数の方向マップに対し, ガウシアンフィルタを適用する計算量は少ないものとは言えない. よって, 方向マップを用いない場合の記述子の計算と比較し, このガウシアンフィルタの計算が効率良く実行できるかどうか, 方向マップにより特徴抽出の計算時間を短縮する際の重要な問題となる.

上記の問題の解決方法として, GPU による並列処理の導入がある. GPU 内部には複数の種類のメモリがあるが, メモリレイテンシの短い共有メモリを活用したデータの再利用 (data reuse) により, 多数の方向マップに対し効率的にガウシアンフィルタを適用可能である. よって, 多重解像度方向マップを導入することにより, セルの全画素の値を参照する従来の方​​法と比較し, より高速な特徴抽出が行えることが示されている [21].

方向マップを用いて記述子を計算する方法では, セルの大きさが変化しなければ, セルの配置を変化させてもヒストグラムへの投票値の再計算は不要である. このことから, 局所領域の大きさの変化を, セルの配置の変化として表現する. それによって, 局所領域の大きさを変化させたとしても, 記述子の計算における処理の変化は, 畳み込み済みの方向マップにおける参照画素の位置を変更することだけとなる. よって, 大きさの異なる複数の局所領域を用いても, 記述子の抽出に要する計算量は大きく増加しないものと考えられる. この点が, 本論文で示す多重サイズ局所記述子の抽出方法の大きな特徴である.

次節では, 局所領域の設定や多重解像度方向マップの生成も含めた, 特徴抽出処理全体の GPU による実装について述べる.

#### 4. GPU による実装

本節では CUDA [29] を用いた特徴抽出の実装について述べる. CUDA では, 処理対象となるデータをいくつかのブロックに分け, それぞれのブロックに複数のスレッドを付随させる. ブロック数とスレッド数は execution configuration (以下, EC と記す) と呼ばれる. ブロック数は, ブロックの幅や高さにより間接的に指定する場合が多い.

図 6 は, 実装を行った処理のフローチャートである. 以下に, この図にはない補助的な処理も含めその実装内容を示す. GPU のメモリの使用方法および方向マップの生成の詳細に関しては, 文献 [11], [21] を参照されたい.

Image transfer: ホストコンピュータ (CPU) から GPU への画像データの転送.

Y Component: カラー画像からの輝度画像生成. プ



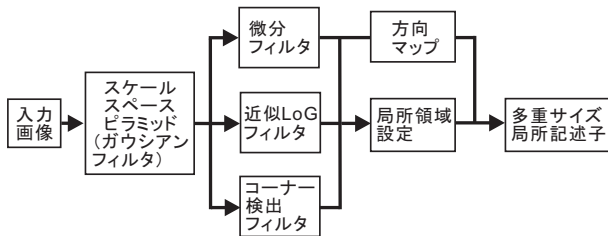


図 6: 局所不変特徴量抽出処理のフローチャート. 密なエッジサンプリングと特徴点抽出に基づき局所領域を設定する [11], [21].

ブロックサイズは  $16 \times 32$ . ブロックの全画素にスレッドを割り当て並列化.

**Down sampling:** スケールスペースピラミッドの生成に必要な画像のダウンサンプリング. EC は Y Component と同じ.

**Gaussian filter:** ガウシアンフィルタを用い, スケールスペースピラミッドを生成. 初期スケールは 1.6. フィルタサイズは, ガウス関数の値が  $10^{-3}$  未満になる点で定義域を打ち切って決定. 等方ガウス関数の変数分離性を利用. 行処理における 1 次元のブロックサイズは 128 とし, 全画素にスレッドを割り当てる. 列処理でのブロックサイズは  $48 \times 16$  とし, ブロックを高さ 8 のサブブロックに分ける. 1 つのサブブロックの全画素にスレッドを割り当て, 各スレッドで他のサブブロックの同一位置にある画素に対し逐次処理を行う. サブブロックを用いて処理を行うのは, 1 つのブロックにおけるスレッド数に上限が存在するためである.

**Gradient filter:**  $5 \times 5$  の微分フィルタ [30] による輝度勾配の計算. ブロックサイズは  $16 \times 16$  とし, ブロックを高さ 8 のサブブロックに分ける. そして, Gaussian filter の列処理と同様にスレッドを割り当てる.

**ALoG-CD filter:** コーナー検出フィルタを併用する近似 LoG フィルタ [31], [32] による, 特徴点抽出のためのスケールスペースピラミッドの生成. EC は Gradient filter と同じ.

**Feature point:** ALoG-CD filter で得られるスケールスペースにおける  $3 \times 3 \times 3$  近傍での極値探索による, 特徴点およびその固有スケールの抽出. 各オクターブ毎に抽出. 近似 LoG フィルタとコーナー検出フィルタの応答に対するしきい値処理により, 特徴点を選択. 各しきい値は, 10 および 100. ブロックサイズは  $16 \times 32$  とし, 全画素に対しスレッドを割り当てる. 特徴点の位置と固有スケールを, 局所領域の位置とスケールとする.

**Edge sampling:** 微分フィルタの処理結果の全スケールにおいて, 微分フィルタの応答がしきい値以上, かつ, 空間  $3 \times 3$  近傍の極大点となる画素をサンプリング [11], [22]. 微分フィルタの応答のしきい値は 10. サンプリングした画素の位置とスケールを, 局所領域の位置とスケールとする. EC は Feature point と同じ.

**Orientation map:** 多重解像度方向マップの生成. 微分フィルタの出力から, 図 5 に示すように輝度勾配の方向を 8 つに離散化した方向マップを生成. EC は Fea-

ture point と同じ. ヒストグラムの境界効果 (boundary effect) を防ぐために, 1 つの輝度勾配の方向に対し, ピンの中心値からの距離に基づいて重み付けを行い, 2 つの方向マップへ輝度勾配の大きさを割り振る [21]. 多重解像度方向マップへ適用するガウシアンフィルタのスケールは, 最大の局所領域でのセルの大きさに相当するように設定する (付録 1. 参照). フィルタリングの処理は, Gaussian filter と同じ.

**Dominant orientation:** 回転不変性を特徴量に付与するための dominant orientation [2] の計算. 局所領域の大きさは, 局所領域のスケールの 5 倍. 30 個の局所領域をブロックとし, 局所領域を単位として並列化. ブロック内の全局所領域にスレッドを割り当てる. Dominant orientation を求めるためのヒストグラムの生成には, 記述子の計算と同様に多重解像度方向マップを利用した.

**Descriptor:** 多重サイズ局所記述子の計算. 図 4 に示すように, 局所領域に  $4 \times 4$  のセルを配置. 複数の局所領域の大きさは, スケール係数  $s_0 = 20$  を基準とし, 付録 2. に示す方法で決定した. EC は Dominant orientation と同じ. 回転不変性付与のため, dominant orientation に基づきヒストグラムを並行移動 [21]. 全セルのヒストグラムを連結後, ノルムの正規化を行い記述子を得る.

## 5. 実験結果

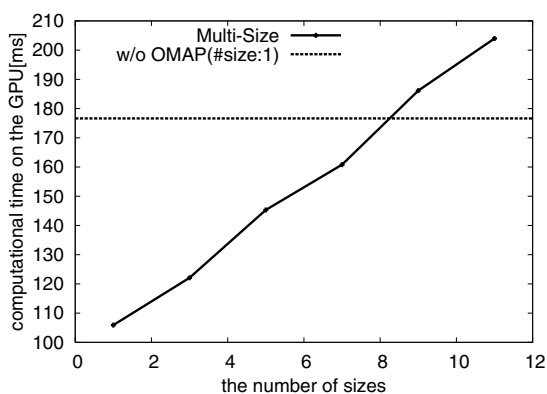
本節では, 4. 節で示した処理を CPU および GPU により実装し, 計算時間を計測した結果を示す. 実験には, 以下の計算機環境を使用した; ホストコンピュータ: HP xw8600 Workstation, OS: Fedora8, CPU: Intel Quad-core Xeon (3.16GHz/12MBL2), メモリ: 8GB DDR2 FBD RAM, グラフィックカード: NVIDIA Quadro FX4600, および, GeForce GTX 280. ディスプレイへの表示は Quadro で行った. 演算専用とした GeForce は 240 個のコアを有し, warp size は 32, Compute Capability [33] は 1.3, 1 つのブロックにおけるスレッド数の上限は 512 である. CPU では単一コアを使用して特徴量抽出を行い, GPU での計算時間と比較した. 浮動小数点演算は, 全て単精度で行った.

図 1 および図 2(a) に示した画像を用い, 計算時間の計測を行った. 画像の解像度は  $720 \times 480$  および  $320 \times 240$  とし, それぞれ, スケールスペースピラミッドのオクターブ数は 5 および 4 とした. また, 各オクターブにおけるスケール数は 3 とした. 輝度勾配の方向は 8 方向で離散化したので, 120 枚もしくは 96 枚の多重解像度方向マップが生成された.

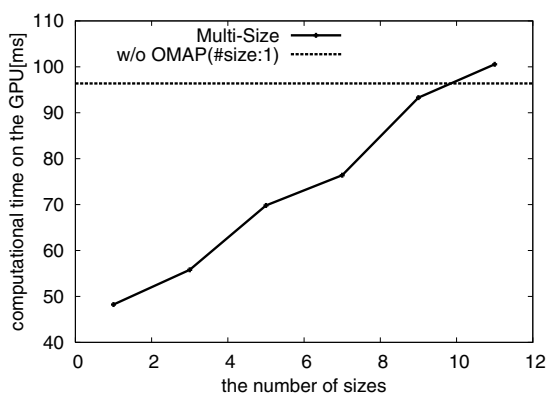
計算時間の計測結果を表 1 に示す. 表 1 の CPU-1 は, CPU で方向マップを使用し 1 つの局所領域から記述子を抽出した場合を表す. GPU- $\{1, 3, 5\}$  は, それぞれ, GPU で方向マップを使用し 1, 3, 5 個の局所領域を用いた場合である. GPU-C は方向マップを使用せずセルの全画素の値を参照し, 1 つの局所領域から記述子を抽出した場合である. また, 図 7 は, 局所領域のサイズ数 (スケール係数の数) に対する計算時間の変化を示す. 表 1 の

表 1: 局所不変特徴量抽出に必要な計算時間．単位は [ms]．OS(Linux) の非リアルタイム性を考慮し，100 回の処理の平均値を示す．総計算時間には，ここに示したタスク以外の処理，例えばメモリアロケーション等も含まれる．

Image	Tour de France					Graffiti				
	720×480					320×240				
Image size										
Task/Implement	CPU-1	GPU-1	GPU-3	GPU-5	GPU-C	CPU-1	GPU-1	GPU-3	GPU-5	GPU-C
Image transfer	N/A	2.571	2.578	2.570	2.563	N/A	0.606	0.603	0.603	0.600
Y component	3.747	0.109	0.109	0.108	0.108	0.938	0.063	0.063	0.062	0.062
Down sampling	0.866	0.141	0.140	0.140	0.140	0.165	0.083	0.084	0.082	0.083
Gaussian filter	277.149	8.888	8.881	8.918	8.766	58.623	4.971	4.999	4.936	4.854
ALoG-CD filter	309.437	13.643	13.642	13.640	13.639	63.805	3.821	3.816	3.813	3.812
Gradient filter	220.617	4.399	4.399	4.409	4.394	45.759	1.364	1.359	1.359	1.358
Feature point	160.817	7.497	7.528	7.473	7.538	35.501	1.959	1.945	1.942	1.957
Edge sampling	12.486	7.347	7.378	7.336	7.312	3.885	1.835	1.827	1.822	1.833
Dominant orientation	92.575	3.295	3.296	3.300	5.977	77.340	1.430	1.467	1.428	3.145
Orientation map	1509.556	46.031	46.935	47.907	N/A	308.565	24.974	25.200	25.338	N/A
Descriptor	51.782	8.356	23.571	45.873	122.483	21.989	4.625	11.938	25.953	76.171
Total	2640.544	105.911	122.111	145.301	176.641	616.980	48.230	55.795	69.812	96.381
#descriptor	5688	5736	5756×3	5786×5	5731	2684	2725	2742×3	2768×5	2706



(a)



(b)

図 7: 局所領域のサイズ数に対する計算時間の変化．(a) 画像 Tour de France に対する結果．(b) 画像 graffiti に対する結果．水平線は，方向マップを使用せず，単一の局所領域から記述子を抽出した場合の計算時間を表す．方向マップを使うことにより，それを使用しない場合と比べ，短い計算時間でより多くの局所領域から記述子を抽出可能ながわかる．

GPU-C と GPU-{1,3,5} を比較すること，および，図 7 より，方向マップを用いることが多重サイズ局所記述子を抽出する処理の高速化に有効であることがわかる．CPU に対する GPU の計算時間における優位性は，表 1 より明らかである．

## 6. ま と め

本論文では，多重サイズ局所記述子の抽出に関し検討を行った．1 つの中心画素に対し，複数の局所領域を設定し記述子の抽出することから，特徴抽出に要する計算量が増加する．その対策として，GPU による並列処理を導入した．特に，方向マップを用いることで，従来の方法よりも高速に多重サイズ局所記述子の抽出が可能であることを示した．また，多重サイズ局所記述子の機能として，シーンに関する事前情報がない場合にも，特徴量の隠れへの耐性，不変性，識別性の間のバランスの制御が可能となることについて述べた．今後は，多重サイズ局所記述子を用いた対応付けや物体認識，および，それらの並列処理による高速化に関する研究開発を推進する予

定である．

## 謝辞

本研究の一部は，科学研究費補助金，課題番号 22500171 「並列処理に基づく物体認識アルゴリズムに関する研究」の助成の下で行われた．

## 文 献

- [1] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Trans. PAMI*, Vol. 19, No. 5, pp. 530–535, 1997.
- [2] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.*, Vol. 60, No. 2, pp. 91–110, 2004.
- [3] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, Vol. 27, No. 10, pp. 1615–1630, 2005.
- [4] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comp. Vis.*, Vol. 65, No. 1/2, pp. 43–72, 2005.
- [5] C. Ssurka, C. R. Dance, L. Fan, J. Willamowski, and

- C. Bray. Visual categorization with bags of keypoints. In *Proc. Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.
- [6] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. Int. Conf. Comp. Vis.*, Vol. 1, pp. 604–610, 2005.
- [7] N. Dalal and B. Triggs. Histograms of orientated gradients for human detection. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, Vol. 1, pp. 886–893, 2005.
- [8] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. European Conf. Comp. Vis.*, pp. 490–503, 2006.
- [9] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, 2008.
- [10] H. Cheng, Z. Liu, N. Zheng, and J. Yang. A deformable local image descriptor. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, 2008.
- [11] 市村直幸. GPU による特徴点とエッジに基づく局所不変特徴量の抽出. 情処研報, Vol. 2009–CG–136, No. 11, 2009.
- [12] 柳井啓司. 一般物体認識. 情処研報, Vol. 2009–CVIM–169, No. 14, 2009.
- [13] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Trans. PAMI*, Vol. 31, No. 12, pp. 2129–2142, 2009.
- [14] J SPORTS において放送された映像を使用している.
- [15] T. Lindeberg. Feature detection with automatic scale selection. *Int. J. Comp. Vis.*, Vol. 30, No. 2, pp. 79–116, 1998.
- [16] Interest Point Test Sequences: <http://lear.inrialpes.fr/people/mikolajczyk/Database/>.
- [17] Škoda: <http://new.skoda-auto.com/>.
- [18] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, Vol. 2, pp. 524–531, 2005.
- [19] X. Ma and W. E. Grimson. Edge-based rich representation for vehicle classification. In *Proc. Int. Conf. Comp. Vis.*, Vol. 2, pp. 1185–1192, 2005.
- [20] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, Vol. 1, pp. 26–36, 2006.
- [21] N. Ichimura. GPU computing with orientation maps for extracting local invariant features. In *Proc. Sixth IEEE Workshop on Embedded Computer Vision (ECVW2010)*, 2010.
- [22] 市村直幸. 密なエッジサンプリングに基づく局所不変特徴量による対応付け. 信学技報, No. PRMU2009-51, pp. 71–76, 2009.
- [23] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proc. British Machine Vis. Conf.*, Vol. 2, pp. 779–788, 2003.
- [24] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Proc. Int. Conf. Comp. Vis. Patt. Recog.*, 2007.
- [25] T. Kobayashi and N. Otsu. Image feature extraction using gradient local auto-correlations. In *Proc. European Conf. Comp. Vis.*, pp. 346–358, 2008.
- [26] T. Mita, T. Kaneko, B. Stenger, and O. Hori. Discriminative feature co-occurrence selection for object detection. *IEEE Trans. PAMI*, Vol. 30, No. 7, pp. 1257–1269, 2008.
- [27] T. Watanabe, S. Ito, and K. Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Proc. Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pp. 34–47, 2009.
- [28] 藤吉弘巨. 局所特徴量の関連性に着目した joint 特徴による物体検出. 情処研報, Vol. 2009–CVIM–166, No. 8, 2009.
- [29] CUDA Zone: [http://www.nvidia.co.jp/object/cuda\\_home\\_new\\_jp.html](http://www.nvidia.co.jp/object/cuda_home_new_jp.html).
- [30] S. Ando. Consistent gradient operators. *IEEE Trans. PAMI*, Vol. 22, No. 3, pp. 252–265, 2000.
- [31] M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, Vol. 16, pp. 75–87, 1998.
- [32] 市村直幸. 近似 LoG フィルタを用いた局所不変特徴量の抽出 – GPU による実装 –. 情処研報, Vol. 2008–CVIM–165, pp. 243–250, 2008.
- [33] *NVIDIA CUDA Programming Guide, Version 2.0, Sec.5.1.2*, 2008.

## 付 録

### 1. 方向マップに適用するガウシアンフィルタのスケールの決定方法

図 4 に示す  $4 \times 4$  のセルを局所領域に配置する場合について, 方向マップに適用するガウシアンフィルタのスケールの決定方法を示す. 局所領域の固有スケールを  $\sigma$  とする. この  $\sigma$  にスケール係数  $s$  を乗じて, 局所領域の大きさを決定する. 局所領域を  $4 \times 4$  に分割すると, その結果得られる正方形の幅は  $s\sigma/4$  となる.

方向マップに対するガウシアンフィルタのスケールを  $\sigma_0$  とする. 局所領域を分割して得られる正方形の外接円の位置で, ガウス関数がどの程度の値を保持するかを変化させ, セル内部の輝度勾配の大きさに対する重み付けを制御する. 本論文では, 外接円の位置でのガウス関数の定義域の値を  $3\sigma_0$  とした. よって, 外接円の直径は  $6\sigma_0$  となる. この直径が正方形の幅の  $\sqrt{2}$  倍と一致することから,  $\sigma_0$  は次式で求められる.

$$\sigma_0 = \sqrt{2}s\sigma/24 \quad (\text{A}\cdot 1)$$

表 1 の計算時間の計測では, 式 (A・2) から得られる最大のスケール係数を用い  $\sigma_0$  を求めた. これは, 最大のセルに対応するガウシアンフィルタを用いること, つまり, 最大のセルを局所領域に配置することを意味する.

### 2. スケール係数の変化方法

まず, スケール係数の基準値  $s_0$  を定める. そして, 次式によってスケール係数  $s$  を変化させ, 基準となる局所領域よりも, 小さなもの大きなもの設定を行った.

$$s = s_0(0.1L + 1.0) \quad (\text{A}\cdot 2)$$

ここで  $L$  は局所領域のサイズ数を  $N$  (奇数) とした場合,  $-\lfloor(N/2)\rfloor \leq L \leq \lfloor(N/2)\rfloor$  である. 例えば,  $N = 3$  ならば,  $-1 \leq L \leq 1$  となり, スケール係数  $s = 0.9s_0$  と  $s = 1.1s_0$  により縮小および拡大された 2 つの局所領域が, 基準となる局所領域に加えられる. 表 1 の計算時間の計測では, スケール係数の基準値  $s_0$  を 20 とした.