平成 23 年度　修士論文

物体抽出を用いた一般物体認識の性能改善

早稲田大学基幹理工学研究科　情報理工学専攻

5110B108-5

藤川　哲也

指導　甲藤 二郎　教授

2012 年 1 月 31 日

指導教授印	受付印

目次

第 1 章

まえがき

今日のデジタルカメラの普及に伴い、周囲にはデジタル化された写真が大量に存在している。従来研究されてきた特定の制約下で撮影された画像とは異なり、このような制約のない「一般的」な画像に対して、計算機が画像中に含まれる物体を一般的な名称、たとえば「鞄」「くつ」「いす」などの名称で認識することを一般物体認識(generic object recognition)と呼び、画像認識の研究においてもっとも困難な課題の一つとされる。

近年では、前述のように個人が所有するデジタル画像の量が急激に増加したが、計算機が画像の意味を理解できないために、画像データの分類や検索には人手の介入が不可欠となっている。そのようなセマンティックギャップを解消するために一般物体認識の実現が期待されている。

一般物体認識でしばしば用いられる手法としては Bag-of-Keypoints(Bag-of –Features とも呼ばれる)があげられる。この手法では画像を局所特徴(keypoint)の集合だとみなして、画像をヒストグラムとして表現する手法である。しかし、画像全体から得られるヒストグラムには前景の特徴と背景の特徴が含まれており、前景と背景の共起関係を考慮すると背景の特徴がノイズとなって、認識精度を低下させるカテゴリが存在する場合があると考えられる。

一方で、画像の前景と背景を分離するための領域分割手法としては Graph Cuts などがあげられる。この手法は、ユーザが与えた seed と呼ばれる正解ラベルを付加したピクセルを指定し、それらの情報から前景と背景に分離する手法となっているが、自動で物体を抽出することはできない。そこで、本研究では認識対象の前景物体は視覚的注意を惹きやすいと考え、Saliency Map と呼ばれる視覚注意を表すモデルを用いて自動物体抽出を実現した。

そこで、本研究では Saliency Map と Graph Cuts の組み合わせによって画像を前景と背景に分離し、それぞれの領域から得られる複数の画像特徴を重みづけ統合することで各カテゴリの前景と背景の共起関係を考慮した一般物体認識手法を提案する。

各章の構成

本論文は 6 つの章から構成される。2 章は一般物体認識の歴史を述べ、3 章は関連技術に関して述べる。4 章では、提案手法について、5 章では実験について述べ、最後の 6 章で本

論文のまとめとする。

第 2 章 一般物体認識の歴史

一般物体認識は、画像認識の研究が始まった 1960 年代から研究が行われていたが、最初に成功を見た研究は限定された世界のものであり、その代表例の線画解釈は多くの研究が行われたが、線画そのものや容易に線画が得られる画像のみが研究対象となり、実世界の画像からいかに正しく線画を抽出するかという問題が解決されることはなかった。

その後、あらかじめ用意しておいた物体の形状モデルを知識として与え、画像とモデルの照合を行うことで認識を行うモデルベースト(model-based)物体認識などが提案されたが、それらの方法は、どれも物体の形状を直接認識に利用していた。そのため、認識する対象の形状が完全に既知でなければ正しい認識が不可能であった。また、固有の物体を識別する identification の物体認識には向いていたが、一般的な名前を識別する classification の物体認識（一般物体認識）に適用することは困難であった。

一方、異なるアプローチも提案され、物体の機能を推測して機能から物体を認識する function-based recognition、物体の候補を複数出して物体間の関係により最終的な結果を出力する context-based recognition などが提案されたが、結局ルールベースの認識手法には変わりなく、一般化することは不可能であった。

その後、学習画像を用意して自動的に特徴量を抽出し認識を行う研究が多く行われるようになった。物体の形状を用いない方法として、テクスチャや色を用いる方法が提案された。特徴量が色のみであるため、classification 的な物体認識には向かないが、大量のデータに対する identification にはきわめて有効な手法である。これらの方法では、学習画像を用意すれば認識が可能となるが、認識対象の切り出しによって認識対象のみが写っている学習画像を用意する必要があり、種類を増やすことは容易ではなかった。さらにオクルージョンに対応できないという問題もあった。

近年では、計算機の発展により大量のデータを高速に処理可能になったことにより、統計や機械学習の分野の学習手法が適用できるようになり、人手によるルールやモデル構築に基づく手法から統計的機械学習手法へと移行した。統計的学習手法を用いた研究について代表的な方法を述べる。

（1）領域に基づく手法

領域に基づく手法でもっとも有名な方法が word-image translation model である。これはあらかじめ画像全体に対し数個のキーワードが付加されている Corel 画像データベースを用いて、領域分割された画像の領域への自動アノテーションを行った。Blobworld もしくは Normalized Cuts を用いて領域分割し、領域分割された各画像領域と単語の対応付けを統計的に推定する手法である。

（2）局所パターンに基づく手法

領域分割による方法ではオクルージョンがある場合、形状が複雑で領域分割がうまくいかない場合には対処することが困難であった。そこで C. Schmid らは画像の局所的な特徴の組み合わせによって照合を行う方法を提案した。これは Harris interest point detector によって、画像中から 100 点程度の特徴点を選び出し、各点の特徴を特徴ベクトルとして、それらの集合によって画像を特徴付ける。照合には未知画像に対し、同様に特徴ベクトルを求め、学習画像の特徴ベクトルの中から、それぞれ近い特徴ベクトルを探して投票を行い、最終的に最も多くの投票を得た学習画像にマッチしたとみなす。この研究が局所領域の切り出しによる物体認識の最初の研究である。また D. Lowe も同様の方法によってオクルージョンのあるシーンでの物体認識を実現している。しかし、これらの研究は同一対象を探す identification の物体認識であった。

また、M. C. Burl らは、局所領域の特徴とその位置関係を確率モデルで表現する constellation model（星座モデル）を提案した。この研究では classification の物体認識を実現していたが、学習画像の局所領域はあらかじめ指定しておく必要があった。constellation model では、局所領域の相対位置の情報も確率モデル化していたが、局所領域の特徴量のみで認識を行う方法が提案されており、constellation mode に匹敵する認識結果を出している。その手法が第 4 章で詳細を述べる Bag-of-Keypoints[1]である。

Bag-of-Keypoints では画像を局所特徴の集合とみなしてヒストグラム化する手法であるが、空間的な位置情報や前景と背景などを考慮していない問題があった。空間的な位置情報を付加する手法としては、Lazebnik らは画像を空間的に分割したピラミッドを構築する Spatial Pyramid Kernel という手法を提案した。また、背景はノイズとなるとしたアプローチとして、Marszałek らは背景の特徴の影響を抑制する Spatial Weighting という手法を提案した。その他、前景を抽出する手法として、SVM や AdaBoost を用いてグランドツルース画像と似た領域をテスト画像から探索する手法が提案された。しかし、これらの前景抽出手法は抽出のための学習がさらに必要であり、カテゴリ数の増加に従って処理時間も増加するという問題を抱えている。

また画像特徴に対する研究も進んでいる。アピアランスを表現する SIFT 特徴[2]、人物検出などに用いられる HOG 特徴[3]、形状の類似性に着目したシェイプを表現する Self-Similarity[4]など、さまざまな画像特徴を表現する手法が提案されている。一方で、これらの画像特徴をうまく利用することで精度を改善する研究も盛んである。最近では、これら複数の画像特徴を Multiple Kernel Learning と呼ばれる手法で重みづけて統合することで特に高精度な認識が可能であると報告されている[5]。この Multiple Kernel Learning は最適な重みの推定と学習を同時に行えることもあり、今日非常に多くの一般物体認識の研究で用いられている。

(なお、本節の執筆に当たり[6]を参考にした)

第 3 章 関連技術

3.1 Bag-of-Keypoints とその派生手法

3.1.1 Bag-of-Keypoints

前述の通り、Bag-of-Keypoints[1]は局所領域の特徴量のみで認識を行う手法である。この手法は、統計的言語処理における bag-of-words model のアナロジーで、bag-of-words が語順を無視して、文章を単語の集合とみなすのと同様に、画像を局所特徴の集合と捉える考え方である。実際には図 3.1.1.1 に示すように局所特徴の特徴ベクトルをベクトル量子化することで、画像の特徴点を bag-of-words の words と同様に扱えるようにする。このベクトル量子化された特徴は visual word と呼ばれ、画像はこの visual word の出現頻度ヒストグラムで表現される。L. Fei-Fei らは局所特徴を SIFT 特徴量で表し、全ての特徴量を k-means クラスタリングして codebook（visual word の集合）を作成し、さらにこの codebook を用いて学習画像、テスト画像両方の特徴量をベクトル量子化し、その結果から確率的文書分類手法の LDA (Latent Dirichlet Allocation) を用いて 13 種類のシーンを 64% の精度で分類を行った[7]。しかし、現在では LDA の代わりに SVM (Support Vector Machine) に代表される判別モデル (discriminative model) が利用されるようになった。SVM は高い汎化性能を持ったクラス分類手法であり、現在さまざまな画像認識問題に応用されている。通常、Bag-of-Keypoints を用いた認識の流れは図 3.1.1.2 のようになる。

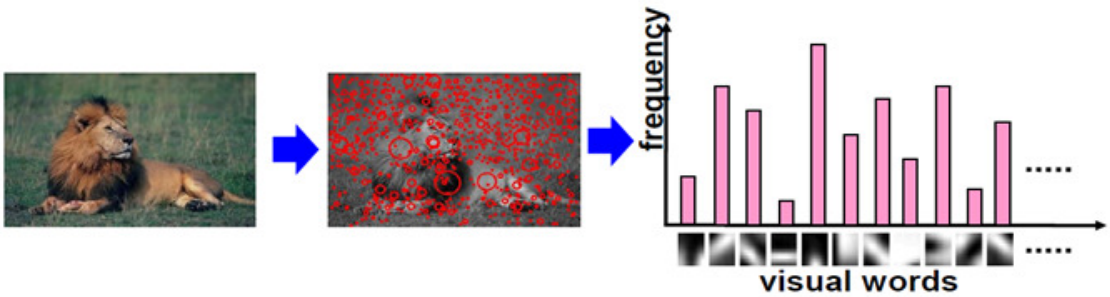


図 3.1.1.1 Bag-of-Keypoints の考え方[6]

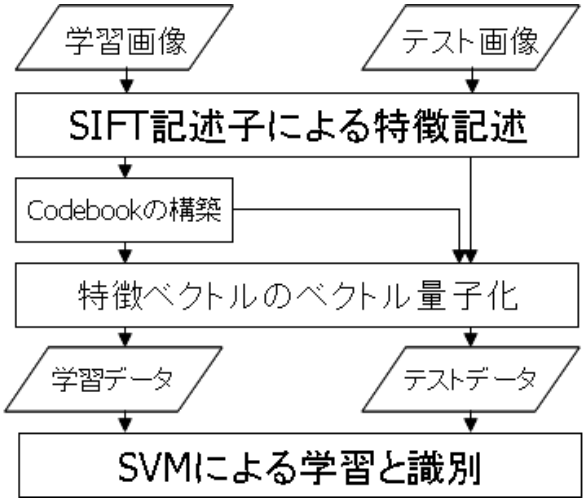


図 3.1.1.2 Bag-of-Keypoints による認識の流れ

3.1.2 Spatial Pyramid Kernel

Spatial Pyramid Kernel は局所特徴の空間的位置情報を無視した Bag-of-Keypoints に空間的位置情報を付加した手法で、Lazebnik らによって提案された[8]。図 3.1.2 に示すように、画像全体でヒストグラムを作成するものが Level 0、1/4 に分割してそれぞれの領域でヒストグラムを作成するものが Level 1、というようにピラミッドを Level 3 まで作成し、これらのカーネルを線形結合することで、空間的位置情報を考慮した認識が可能となっている。また、Lazebnik らは局所特徴量にエッジ上の勾配に関する特徴を 2 スケール×8 方向の 16 次元で表現した”Weak features”と 16×16 ピクセルのパッチで 8 ピクセルごとに Grid Sampling した SIFT 特徴である”Strong features”を用いて、15 種類の風景画像を 81.4%の精度で分類した(前述の Fei-Fei らの手法は 15 種類に対して 65.9%の精度で分類した)。また、Caltech-101 を 64.6%の精度で分類した。

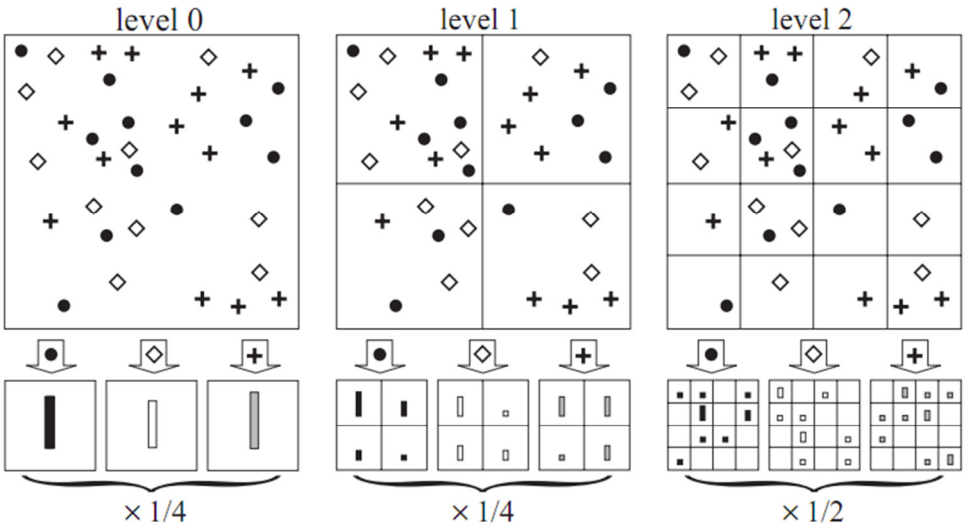


図 3.1.2 Spatial Pyramid Kernel の概要[8]

3.1.3 Spatial Weighting

Spatial Weighting は背景の影響力を抑えるべく重みづけた Bag-of-Keypoints であり、Marszałek らによって提案された[9]。この手法ではあらかじめ作成したグランドツルース(正解画像)の前景とマスク画像の位置関係を学習しておき、入力画像の局所特徴に対して学習済みのマスクの位置関係を適用していくことで、図 3.1.3 の 2 列目に示したような重みマップを作成し、これをヒストグラム構築時の重みとして利用することで背景の特徴による影響を抑制することが可能となっている。

図 3.1.3 Spatial Weighting の概要[9]

3.2 画像特徴量

今日のコンピュータビジョンの世界では、画像の持っている特性を数字で表現し、それらの数字を用いることで、画像認識や画像復元などを行っている。この節では今日のコンピュータビジョンの世界で用いられているいくつかの画像特徴を紹介する。なお、SIFT と HOG の執筆にあたって [10]を参考にした。

3.2.1 SIFT とその派生手法

SIFT(Scale Invariant Feature Transform)[2]は D. G. Lowe によって提案された局所特徴量であり、今日のコンピュータビジョンの研究で最も用いられている画像特徴といえる。この項では SIFT だけでなく、それらを応用した派生手法に関しても述べる。

3.2.1.1 SIFT

SIFT とはスケール変化、回転、照明変化、JPEG 圧縮に頑健な特徴の検出、特徴量記述が可能なアルゴリズムであり、図 4.2 に示すように画像中の勾配に関する特徴を表現することができる。

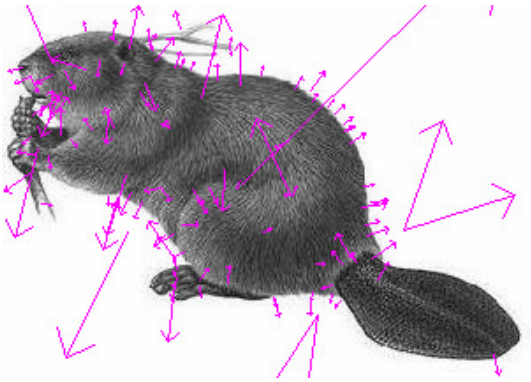


図 3.2.1.1　SIFT 特徴

SIFT の処理は特徴点（キーポイント）の検出(detection)と特徴量の記述(description)の 2 つのステージからなり、以下のような処理を行う。

$$\text{detection}\begin{cases}1.\text{スケールとキーポイント検出}\\2.\text{キーポイントのローカライズ}\end{cases}$$

$$\text{description}\begin{cases}3.\text{オリエンテーションの算出}\\4.\text{特徴量の記述}\end{cases}$$

1.スケールとキーポイント検出では、DoG(Difference of Gaussian)処理によりスケールとキーポイントの候補点を検出し、2.キーポイントのローカライズでは 1.で検出された候補点からキーポイントとして向かない点を削除し、サブピクセル推定を行う。3.オリエンテーションの算出では、回転に不変な特徴を得るためにキーポイントのオリエンテーションを求める。4.特徴量の記述では、3.で求めたオリエンテーションに基づいてキーポイントの特徴記述を行う。以下で詳細を述べる。

3.2.1.1.1 スケールと候補点検出

候補点検出では、DoG 処理を用いてスケールスペースにおける極値探索を行うことで、候補点の位置とスケールを決定する。

3.2.1.1.1.1 LoG によるスケール探索

Koenderink や Lindeberg により、特徴点のスケール探索にはガウス関数が有効であることが証明された。Lindeberg はガウシアンカーネルを用いたスケールスペースとして Scale-normalized Laplacian-of-Gaussian(LoG)を提案している。LoG は、画像にスケール σ を変化させながら次式で示される LoG オペレータ（図 3.2.1.1.1.1）を適用することで、その極大位置を特徴点のスケールとする。

$$LoG = f(\sigma) = -\frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6}\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{1}$$

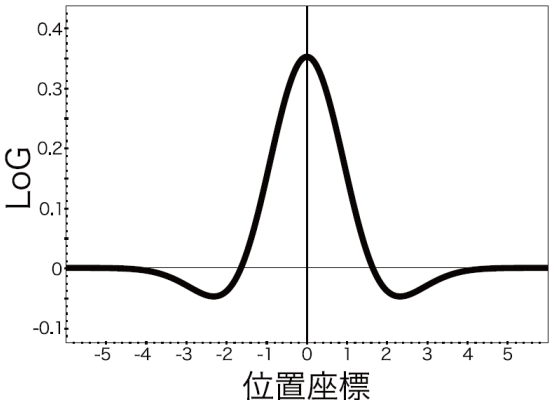


図 3.2.1.1.1.1 LoG オペレータ[10]

ここで、σ はガウシアンフィルタのスケール、x と y は注目画素からの距離である。しかし、LoG は計算コストが高いため、効率的な極値検出法として Lowe によって Difference-of-Gaussian(DoG)を用いる手法が提案されている。DoG と LoG の関係は次式から導かれる。

$$\frac{\partial G}{\partial \sigma} = \sigma\nabla^2 G \tag{2}$$

ここで G はガウス関数、右辺はガウス関数の 2 次微分(LoG)である。この式はさらに次式のように表すことができる。

$$\frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{3}$$

これらの式から、次式がなりたつ。

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{4}$$

この式を書き直すと、次式のようになる。

$$(k-1)\sigma^2\nabla^2 G \approx G(x, y, k\sigma) - G(x, y, \sigma) \tag{5}$$

ここで、左辺は LoG の(k-1)倍であり、DoG が LoG の近似であることがわかる。よって SIFT では計算効率の良い DoG を適用する。

3.2.1.1.1.2 DoG 処理

キーポイント候補点は、入力画像 $I(u,v)$ と各スケールのガウス関数 $G(x,y,\sigma)$ を畳み込んだ平滑化画像 $L(u,v,\sigma)$ の差分(DoG 画像)から求める。それぞれ以下の式により求める。

$$L(u,v,\sigma) = G(x,y,\sigma) * I(u,v) \tag{6}$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{7}$$

DoG の結果の画像 $D(x,y,\sigma)$ は次式で求めることができる。

$$\begin{aligned} D(u,v,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(u,v) \\ &= L(u,v,k\sigma) - L(u,v,\sigma) \end{aligned} \tag{8}$$

この処理を σ_0 から k 倍ずつ大きくした異なるスケール間で行うことで、図 3.2.1.1.1.2 のような複数の DoG 画像を生成する。

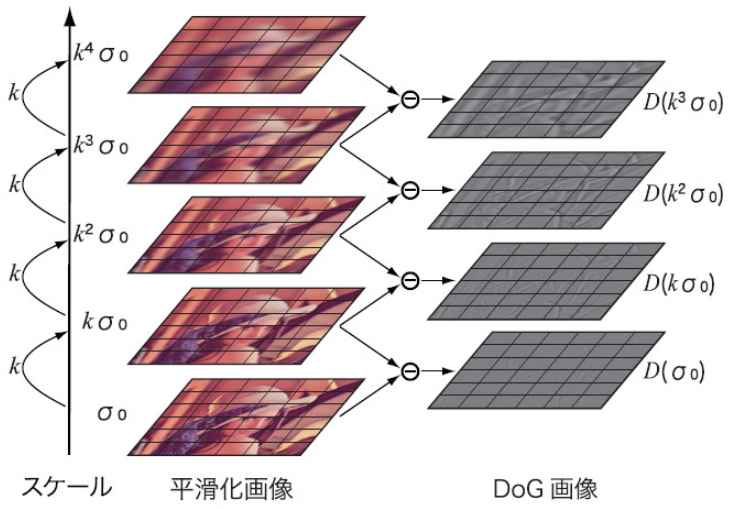


図 3.2.1.1.1.2 DoG 処理の流れ[10]

σ が一定の割合で増加し続けるとガウシアンフィルタのウインドウサイズが大きくなる問題が発生するが、SIFT では画像をダウンサンプリングすることによって σ の変化の連続性を保持した平滑化処理を実現している。

3.2.1.1.1.3 σ の連続性を保持した平滑化処理

σ の連続性を保持した平滑化処理の流れを図 3.2.1.1.1.3 に示す。

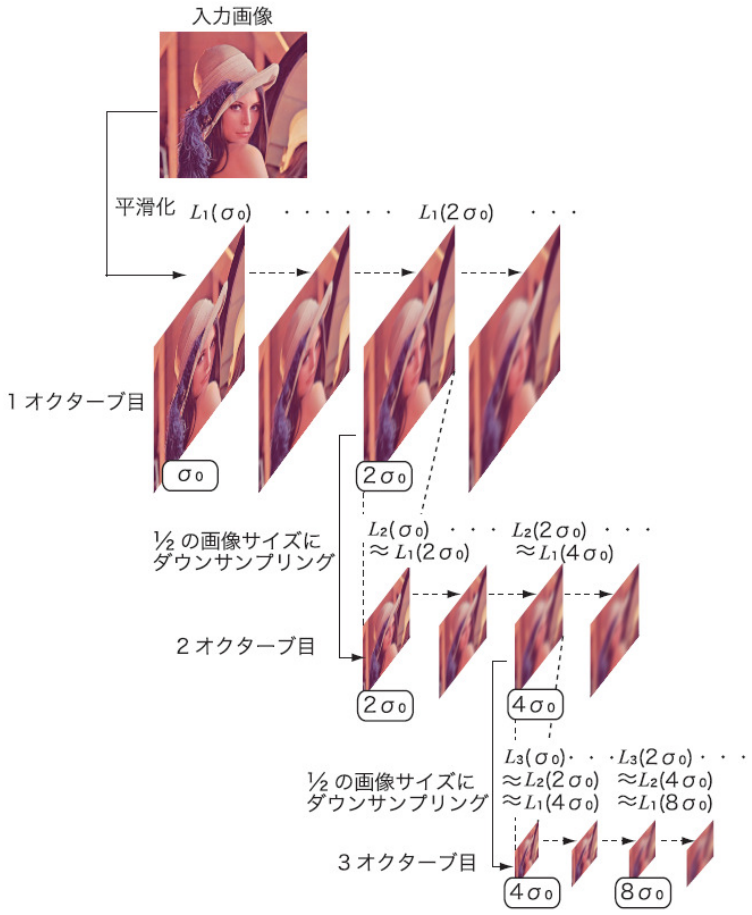


図 3.2.1.1.1.3 σ の連続性を保持した平滑化処理[10]

はじめに、入力画像を初期値である σ_0 で平滑化を行い、平滑化画像 $L_1(\sigma_0)$ を得る。次に k 倍した $k\sigma_0$ で平滑化を行い、$L_1(k\sigma_0)$ を得る。この処理を繰り返すことにより、スケールの異なる複数の平滑化画像を得る。この処理 1 セットを 1 オクターブと呼ぶ。

次に、平滑化画像の中から $2\sigma_0$ で平滑化された画像 $L_1(2\sigma_0)$ を半分のサイズにダウンサンプリングする。ダウンサンプリングされた画像 $L_2(\sigma_0)$ と元画像には次式のような関係が成立する。

$$L_1(2\sigma_0) \approx L_2(\sigma_0) \tag{9}$$

この関係を用いて σ の最大値を制限することで、ウインドウサイズによる計算量の増加を防ぐことができる。

σ の増加率 k は、1 オクターブのスケールスペースの分割数により決定し、1 オクターブでスケールスペースは σ_0 から $2\sigma_0$ まで増加するため、分割数を s とすると増加率 k は $k = 2^{1/2}$ となる。極値探索には DoG 画像を 3 枚 1 組で処理することから、s 枚の極値検出対象画像を得るためには s+2 枚の DoG 画像、つまり s+3 枚の平滑化画像が必要となり、I オクターブにおける平滑化は s+3 回行う。

ダウンサンプリングを行うため、1 枚の入力画像に対するオクターブ数は入力画像のサイズに依存し、画像の一辺の大きさが閾値以下になったときに処理を終了する。

3.2.1.1.1.4 DoG 画像からの極値検出

DoG は異なるスケールの平滑化画像の差分であるため、DoG の値が大きくなる σ では、スケールの変化領域にエッジなどの情報量を多く含んであると言える。よって DoG 画像から極値を検出することでキーポイントとスケールを決定する。極値の検出は前述のように DoG 画像を 3 枚 1 組で行う（図 3.2.1.1.1.4）。注目画素と周囲の 26 近傍を比較し、極値であった場合にはその画素をキーポイント候補点として検出する。この処理を σ の小さい DoG 画像から行う。極値が検出された画素は、それよりも大きいスケールで検出されてもキーポイントの候補点とはしない。この処理をスケールの異なる DoG 画像の全ての画素に対して行う。スケールスペースの極値の性質として、例えば画像サイズが 2 倍になると、DoG の極値探索によりキーポイントのスケールも比例して 2 倍になる。このようにして特徴をもっとも含むスケール σ を自動的に決定するため、空間的に同範囲の領域から特徴量を記述することで、SIFT は拡大・縮小に不変な特徴量となっている。

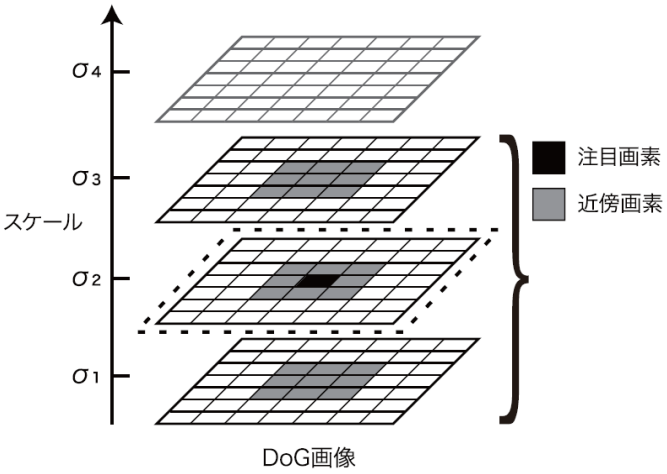


図 3.2.1.1.1.4 極値検出の流れ[10]

3.2.1.1.2 キーポイントのローカライズ

キーポイント候補点の中には、DoG 出力値が小さい（コントラストが低い）点やエッジ上の点が含まれており、これらの点はノイズや開口問題に影響を受けやすく、キーポイントとしては不向きである。これらの点を主曲率とコントラストを用いて絞込み、さらにサブピクセル推定により位置とスケールを算出する。

3.2.1.1.2.1 主曲率によるキーポイントの絞り込み

エッジ上に存在するキーポイント候補点の削除には主曲率を用いる。候補点における 2 次元ヘッセ行列Hを次式により計算し、主曲率を求める。

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{10}$$

行列内の導関数は、候補点の位置での DoG 出力値の 2 次微分から得られる。ここで、ヘッセ行列から求められる第 1 固有値をα、第 2 固有値をβとする。ただし、$\alpha > \beta$とする。この時、対角成分の和$Tr(H)$と行列式$Det(H)$は次式で求められる。

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{11}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{12}$$

さらに、γを第 1 固有値と第 2 固有値の比率とし、$\alpha = \gamma\beta$とすると次式が成り立つ。

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma\beta+\beta)^2}{\gamma\beta^2} = \frac{(\gamma+1)^2}{\gamma} \tag{13}$$

このγはαとβの比率で値が決まることから、固有値を求めることなくエッジ上の点であるかを判別することが可能となる。このγに対して次式の閾値処理を施すことによりキーポイント候補点を削除する。

$$\frac{Tr(H)^2}{Det(H)} < \frac{(\gamma_{th}+1)^2}{\gamma_{th}} \tag{14}$$

[2]では$\gamma_{th} = 10$を採用しており、その場合の閾値は 12.1 となる。

3.2.1.1.2.2 キーポイントのサブピクセル位置推定

前述の通り、サブピクセル推定を用いて位置とスケールを算出する。具体的には 3 変数(x, y, σ)の 2 次関数をフィッティングすることで算出する。ある点$\mathbf{x} = (x, y, \sigma)^T$での DoG 関数$D(\mathbf{x})$をテイラー展開すると次式のようになる。

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{15}$$

この式について $\mathbf{x}$ に関する偏導関数を求め、これを 0 とすると次式が成り立つ。

$$\frac{\partial D}{\partial \mathbf{x}} + \frac{\partial^2 D}{\partial \mathbf{x}^2}\hat{\mathbf{x}} = 0 \tag{16}$$

この時、$\hat{\mathbf{x}}$ はキーポイント候補点のサブピクセル位置を表す。この式を変形すると次式のように表される。

$$\begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x\sigma} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix} \begin{bmatrix} x \\ y \\ \sigma \end{bmatrix} = -\begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix} \tag{17}$$

キーポイント候補点のサブピクセル位置 $\hat{\mathbf{x}}$ を得るために上式を変形すると字式のように表せる。

$$\begin{bmatrix} x \\ y \\ \sigma \end{bmatrix} = -\begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x\sigma} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix} \tag{18}$$

この式を解くことによって、キーポイント候補点のサブピクセル位置 $\hat{\mathbf{x}} = (x, y, \sigma)$ を得る。

3.2.1.1.2.3 コントラストによるキーポイント絞り込み

ここではサブピクセル位置での DoG 出力を算出することでコントラストによるキーポイントの絞り込みを行う。上式は次のように表せる。

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{19}$$

上式を DoG 関数のテイラー展開の式に代入すると、次式が得られる。

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{20}$$

D は DoG 関数であり、$\hat{\mathbf{x}}$ はサブピクセルを表しているため、上式はサブピクセル位置での DoG 出力値を表し、この値を用いてキーポイント削除の判別を行う。文献[2]では、閾値として 0.03 を用いている。出力値が閾値よりも小さい場合、つまりコントラストが低い場合には、ノイズに影響されやすい点であるとしてこの候補点を削除する。

3.2.1.1.3 オリエンテーションの算出

検出したキーポイントに対し、特徴量の記述(description)を行う。まず検出されたキーポイントのオリエンテーションを算出する。オリエンテーションとはキーポイントにおける方向を表し、特徴量記述の際にオリエンテーションにおける方向を表し、特徴量を記述する際に正規化を行うことで、回転に不変となる。キーポイントのオリエンテーションを求めるには、まずキーポイントが検出された平滑化画像 $L(u,v)$ の勾配強度 $m(u,v)$ と勾配方向 $\theta(u,v)$ を以下の式により算出する。

$$\begin{aligned} m(u,v) &= \sqrt{f_u(u,v)^2 + f_v(u,v)^2} \\ \theta(u,v) &= \tan^{-1}\frac{f_v(u,v)}{f_u(u,v)} \end{aligned} \tag{21}$$

$$\begin{cases} f_u(u,v) = L(u+1,v) - L(u-1,v) \\ f_v(u,v) = L(u,v+1) - L(u,v-1) \end{cases} \tag{22}$$

これらを用いて、次式により重み付き方向ヒストグラム h を作成する。

$$\begin{aligned} h_{\theta'} &= \sum_x \sum_y w(x,y) \cdot \delta[\theta', \theta(x,y)] \\ w(x,y) &= G(x,y,\sigma) \cdot m(x,y) \end{aligned} \tag{23}$$

ここで h_θ は、方向を 36 方向に量子化したヒストグラムであり、$w(x,y)$ は画素 (x,y) での重みであり、キーポイントが持つスケールサイズのガウス窓 $G(x,y,\sigma)$ と勾配強度 $m(x,y)$ から算出する。δ は Kronecker のデルタ関数であり、勾配方向 $\theta(x,y)$ が量子化した方向 θ' に含まれるときに 1 を返す。このとき、ガウス窓のスケールにはキーポイントのスケールを適用する。ガウス窓による重み付けを行うことで、キーポイントに近い特徴量がより強く反映される。このようにして算出されたヒストグラムの最大値の 80%以上であるピークをキーポイントのオリエンテーションとする。よって 1 つのキーポイントに対し複数のオリエンテーションが割り当てられる場合も存在する。

3.2.1.1.4 特徴量の記述

検出したオリエンテーションを元に、SIFT 記述子を用いて 128 次元の特徴量を記述する。まず図 4.2.4.1 に示すように、特徴記述を行う領域をキーポイントのオリエンテーション方向に回転する。特徴量の記述にはキーポイント周辺領域の持つ勾配情報を用い、勾配情報はキーポイント中心からそのキーポイントのスケールを半径とする円領域内から求める。次に図 4.2.4.2 に示すように、周辺領域を一辺が 4 ブロックの計 16 ブロックに分割し、ブロックごとに 8 方向の勾配方向ヒストグラムを作成する。このヒストグラムはキーポイントのオリエンテーションのヒストグラムと同様の作成方法で求める。各ブロックで 8 方向

のヒストグラムを作成することから 4*4*8=128 次元の特徴ベクトルとしてキーポイントの特徴を記述する。このようにキーポイントのオリエンテーション方向に座標軸を合わせた領域で特徴記述を行うため、回転に不変な特徴量となる。また、128 次元の各特徴ベクトルの長さはベクトルの総和で正規化を行うことにより、照明変化に対して頑健な特徴量となる。

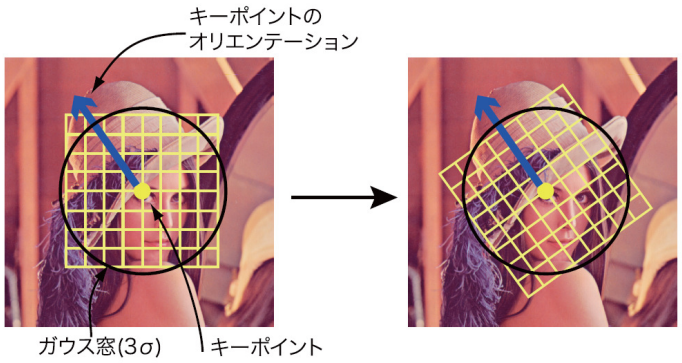


図 3.2.1.1.4.1 特徴記述を行う領域[10]

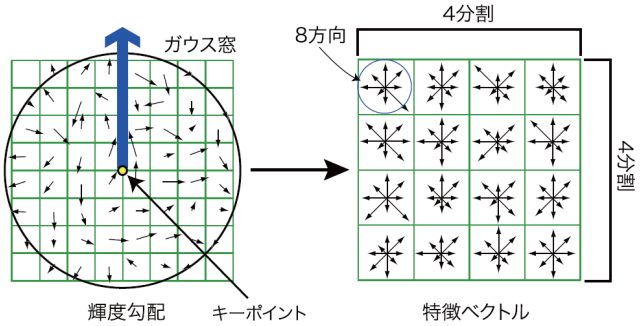


図 3.2.1.1.4.2 ブロックごとの特徴量記述[10]

3.2.1.1.5 SIFT の利用例

Bag-of-Keypoints などで用いられるほか、図 3.2.1.1.5 に示すような異なる画像間での対応点探索が可能である。これは一方の画像で抽出された各キーポイントの SIFT 特徴量と、異なる画像中に含まれる全キーポイントの特徴量とのユークリッド距離 d を算出し、d が最小となる点同士を対応点とすることで検出する。

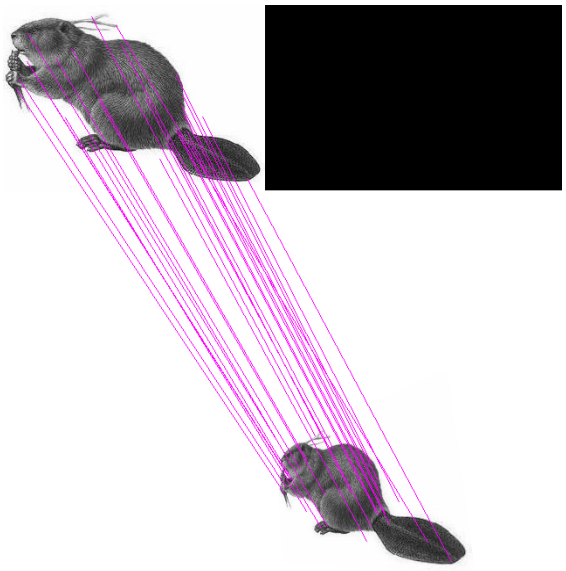


図 3.2.1.1.5 対応点探索例

3.2.1.2 PCA-SIFT

PCA-SIFT[11]は SIFT の勾配情報に対し、主成分分析(PCA)を適用することで性能改善を行った手法である。アルゴリズムとしては、特徴点検出とオリエンテーションの算出は SIFT と同じで、最後の特徴記述のステップが異なる。

SIFT ではキーポイントのスケールに対応した領域を 4*4 のブロックに分割し、各ブロック内で 8 方向の方向ヒストグラムを作成することで、4*4*8=128 次元の特徴量を記述するが、PCA-SIFT ではキーポイントのスケールに対応した領域を 41*41 のパッチにリサンプリングし、パッチ内の水平方向と垂直方向の 2 方向の勾配強度を算出することで、39*39*2=3042 次元の特徴量を得る。この 3042 次元の特徴量に対して主成分分析を適用することで次元圧縮を行う。

ノイズ・回転・スケール変化・アフィン変化・照明変化のある画像を用いた SIFT との比較実験の結果、全ての場合で PCA-SIFT の方が、性能が高いことが報告されており、また特徴量が次元圧縮されているため、高速に処理を行うことができる。

3.2.1.3 CSIFT

SIFT は色情報を含まないが、色はマッチングや物体認識に役立つ情報であり、CSIFT (Colored SIFT) [12]はグレー空間を用いる代わりに色不変空間での SIFT 記述を行う。CSIFT で用いられる色不変モデルは Geusebroek らによって提案されたモデルであり、このモデルでは色不変を実現するために反射スペクトルを次式でモデル化した Kubelka-Munk 理論を適用する。

$$E(\lambda,\vec{x}) = e(\lambda,\vec{x})(1-\rho_f(\vec{x}))^2 R_\infty(\lambda,\vec{x}) + e(\lambda,\vec{x})\rho_f(\vec{x}) \tag{24}$$

ここで、λは波長、$\vec{x}$は画像の位置を表す2次元ベクトル、$e(\lambda,\vec{x})$は輝度スペクトル、$\rho_f(\vec{x})$はフレネル反射率、$R_\infty(\lambda,\vec{x})$は物体反射率、Eは視点方向の反射スペクトルを表す。等しいエネルギー輝度を想定することにより、スペクトルに関する構成要素は波長に不変であり、位置を変化させることができ、実際のケースのほとんどに適用することができるため、Eをλで微分したE_λ、二階微分した$E_{\lambda\lambda}$を用いて、視点やオリエンテーション、輝度方向などに影響を受けない反射特性Hを求めることができる。

$$\begin{aligned} H &= \frac{E_\lambda}{E_{\lambda\lambda}} = \frac{\partial R_\infty(\lambda,\vec{x})}{\partial\lambda} \Big/ \frac{\partial^2 R_\infty(\lambda,\vec{x})}{\partial\lambda^2} \\ &= f(R_\infty(\lambda,\vec{x})) \end{aligned} \tag{25}$$

RGB空間からこれらの不変量の計算を行うために、一般モデルとしてガウシアンカラーモデルが用いられる。このモデルではRGB空間からの線形変換がスペクトルの微分商$(\hat{E},\hat{E}_\lambda,\hat{E}_{\lambda\lambda})$が用いられる。この時、空間の微分商$(\hat{E}_x,\hat{E}_{\lambda x},\hat{E}_{\lambda\lambda x})$はガウシアン微分フィルタと畳み込むことによって得られる。RGBからXYZ(CIE 1964 XYZ表色系)への線形変換と、XYZからガウシアンカラーモデルへの線形変換の結果を用いることで、RGBでのガウシアンカラーモデルの結果が次式のように示される。

$$\begin{pmatrix} \hat{E} \\ \hat{E}_\lambda \\ \hat{E}_{\lambda\lambda} \end{pmatrix} = \begin{pmatrix} .06 & .63 & .27 \\ .3 & .04 & -.35 \\ .34 & -.6 & .17 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{26}$$

Abdel-Hakimらは様々な照明方向、照明強度、視点などの画像を含んでいるデータセットALOI(Amsterdam Library of Object Images)に対して評価実験を行い、多くの場合でCSIFTがすぐれていることを示している。

3.2.1.4 BSIFT

SIFTはキーポイント周辺を含んだ領域の勾配情報の記述を行うため、対象物体以外の背景領域を含んで記述することがある。Steinら[13]は物体と背景の境界情報を用いることで、物体領域のみの情報による特徴点検出と記述するBSIFTを提案している。ただし、BSIFTでは境界情報が重要であり、境界情報が既知であるか、距離画像などから情報を得た画像を対象としている。

BSIFTはDoGのための平滑化としてガウス関数ではなく、以下の式を用いる。

$$I^{k+1}(u,v) \leftarrow I^k(u,v) + \tau\nabla^2 I^k(u,v) \tag{27}$$

$$\nabla^2 I(u,v) = \frac{\partial^2 I}{\partial u^2} + \frac{\partial^2 I}{\partial v^2} \tag{28}$$

kは繰り返し回数であり、式(28)はラプラシアンフィルタ、式(27)は 2 次微分を用いた平滑化である。式(27)を用いることで、物体の境界に注目した平滑化を行う。

SIFT はキーポイントを中心としたガウス分布を重みとして輝度勾配ヒストグラムを作成するが、これでは背景領域も含んでしまうため、BSIFT ではガウス分布と距離情報を用いて距離変換した値を掛け合わせて重み分布とする。これにより、背景領域の影響を抑えた SIFT 特徴量を得ることができる。

3.2.1.5 SURF

Bay らが提案した SURF[14]は SIFT の高速化改良として知られており、若干精度が落ちるものの、AR など高速な処理が必要な技術に取り入れられている。SIFT では複数の DoG 画像に対して処理を加えることでキーポイントを検出していたが、この処理は SIFT アルゴリズムの中でも計算コストが高いという問題を抱えていた。そこで SURF では Hessian-Laplace 検出器を近似した Box Filter を使用し、この出力値の算出に Integral Image[15]を用いることで、高速な処理を可能にしている。

3.2.2 HOG とその派生手法

HOG 特徴量[3]は、SIFT と同様に局所領域の輝度勾配強度とその方向を用いてヒストグラム化した特徴量である。SIFT は局所領域に対して特徴量を記述するが、HOG では一定領域に対して特徴量の記述を行う。これにより、特徴に位置情報が付加され、大まかな物体形状を表現することができるため、人検出や車検出などの物体認識に用いられている。

3.2.2.1 HOG

前述のとおり HOG 特徴は画像の勾配強度と勾配方向のヒストグラムを計算することで得られる。ここでは HOG 特徴の算出方法について述べる。

まず、はじめに各ピクセルの勾配強度mと勾配方向θを SIFT と同様に次式より算出する。

$$m(u,v) = \sqrt{f_u(u,v)^2 + f_v(u,v)^2} \tag{29}$$

$$\theta(u,v) = \tan^{-1}\frac{f_v(u,v)^2}{f_u(u,v)^2} \tag{30}$$

$$\begin{cases} f_u(u,v) = I(u+1,v) - I(u-1,v) \\ f_v(u,v) = I(u,v+1) - I(u,v-1) \end{cases} \tag{31}$$

次に、このように算出された勾配強度と勾配方向を用いて、5×5 ピクセルを 1 セルとした領域の勾配ヒストグラムを作成する。この時、勾配方向は 0°～180°を 20°ずつ 9 方向に量子化する。

さらに各セルで作成した勾配ヒストグラムを 3×3 セルを 1 ブロックとして正規化を行う。

このブロックの正規化は、k 番目のブロックの特徴量(81 次元)をV_kとすると、次式で表現できる。

$$v = \frac{f}{\sqrt{\|V_k\|^2+\epsilon^2}} \tag{32}$$

この正規化は 1 セルずつずらしながら行う。つまり、1 セルが最大で 9 回正規化される。

以上のようなアルゴリズムによって HOG 特徴は一定領域の勾配情報を記述できる。また、HOG 特徴量はセルサイズや画像サイズなどによって次元数が異なり、30×60 ピクセルの画像に対し、HOG 特徴の記述を行った場合、横方向に 4 ブロック、縦方向に 10 ブロックの合計 40 ブロックに対して正規化を行うこととなり、得られる特徴の次元は 40×81 次元=3240 次元となる。

3.2.2.2 PHOG

PHOG(Pyramid Histogram of Oriented Gradients)[16]は HOG 特徴に Spatial Pyramid Kernel 同様、空間的な位置情報を付加すべく、ピラミッド化した特徴量であり、A. Bosch らによって提案された。図 3.2.2.2 に示すように、Spatial Pyramid Kernel 同様、画像を空間的に分割し、それぞれの領域内で HOG 特徴量のヒストグラムを作成することで位置情報を付加した特徴量となっている。本来の手法は 180 度を 9 方向に分割しているが、この手法を提案した論文では 10,20,30,40 で実験を行い、20 の時に最も画像認識のパフォーマンスが高いとしている。また、従来の HOG ではブロック単位の正規化を行いながら高次元の特徴を記述するが、この手法では各セル内の勾配方向に一番近いビンに勾配方向を加算していくことでヒストグラムを作成している。

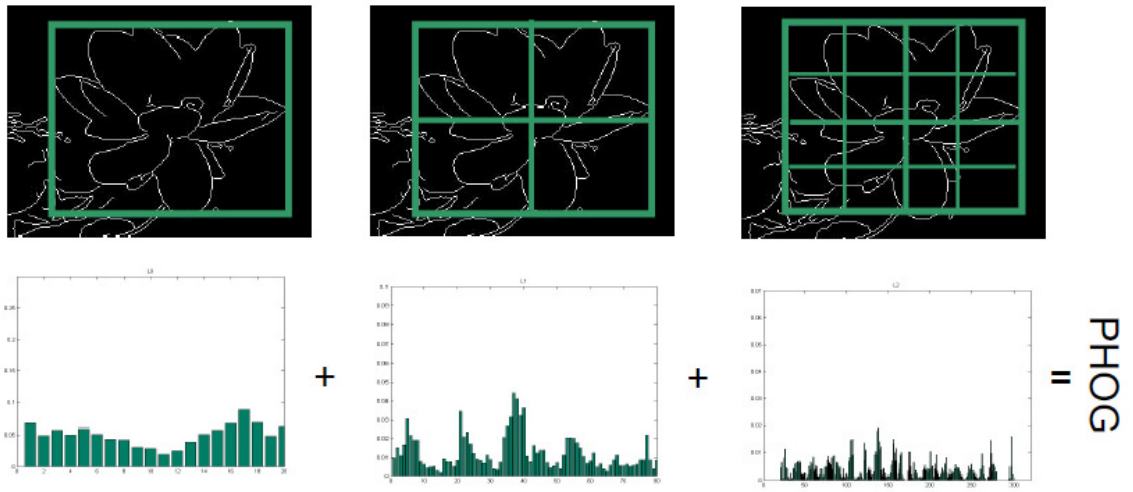


図 3.2.2.2 PHOG のヒストグラム作成方法[16]

3.2.3 Self-Similarity

Self-Similarity[4]は図 3.2.3.1 に示すような相似性に着目した局所特徴量であり、E. Schechtman らによって提案された。

Self-similarity は画像の一部(画像の 5%程度)の特徴を示す局所特徴量となっており、この特徴は図 3.2.3.3、図 3.2.3.4 に示すような画像認識(物体検索やスケッチ検索)、映像認識(動き検出)などに応用できる。

アルゴリズムとしては、各ピクセル q に対して、q を中心とする 5*5 のパッチと、さらにそれを囲む領域(5%程度、80*80 とか)内を CIE L*a*b*空間でテンプレートマッチングのように SSD (Sum of square difference)を用いて比較する。さらに SSD を次式で正規化することで"correlation surface" S_q に変換する(図 3.2.3.2)。

$$S_q(x,y) = \exp\left(-\frac{SSD_q(x,y)}{\max(\mathrm{var}_{noise}, \mathrm{var}_{auto}(q))}\right)$$

ここで、var_{noise} は色や輝度、ノイズ由来などの画素値の分散に相当する項であり、var_{auto} はパッチのコントラストやパターン構造を考慮する項である。var_{auto} は、実装では q に隣接するパッチのうち、最も画素値の差が大きいものの値としている。さらにこの S_q を対数極座標変換し、20 方向、4 つの半径に分割する。各ビンには最大相関値を割り当てる。このようにしてできた 80 次元の特徴を最後に[0,1]で正規化してできた特徴ベクトルが Self-similarity である。

一般物体認識に応用するにおいては、特徴をグリッドサンプリングしたあと、BoK 同様ベクトル量子化をすることでヒストグラム化できると考えられ、実際に Vedaldi らも一般物体認識においてこの特徴を使って(ほかには BoK (dence SIFT)と geometric blur features を用いている)非常によいスコアを出している[17]。

図 3.2.3.1 Self-Similarities[4]

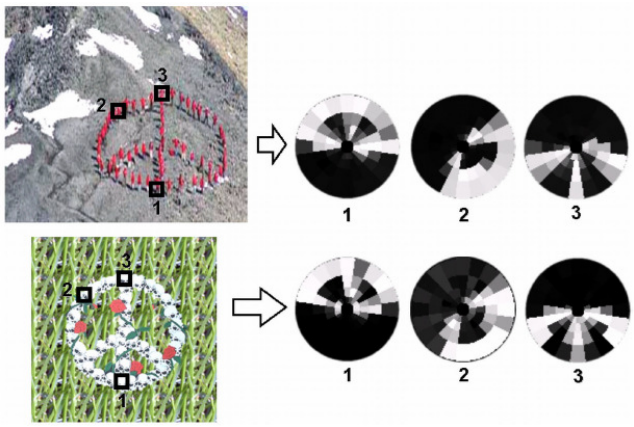


図 3.2.3.2 Self-Similarity descriptors[4]

図 3.2.3.3 物体検出[4]

図 3.2.3.4 スケッチ画像検索[4]

3.2.4 色特徴

色を表現する色空間は通常 3 つ、および 4 つの方向性を備える空間で表現され、RGB をはじめ、HSV や YcbCr など様々な色空間が存在する。この項では、主に用いられている色空間をいくつか紹介する。

3.2.4.1 CIE RGB

RGB は Red、Green、Blue の光の三原色からなる色空間である。光の三原色のため加法混色によって表される。黒の状態にそれぞれ赤緑青をどの程度加えたかによって表現される。一般的なビットマップファイルでは 3 色に 8 ビットずつ割り当て 24 ビットとし、全て 0 の状態を黒、数値が増えるごとに白くなりすべて 255 の状態を白として表す。24 ビットの場合表すことのできる数は 1677 万 7216 色となる。図 3.2.4.1 に加法混色の図を示す。

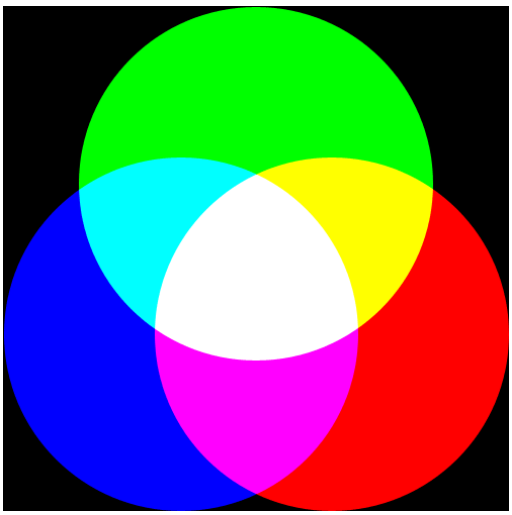


図 3.2.4.1 加法混色

3.2.4.2 CMY

CMY は Cyan、Magenta、Yellow の色の三原色からなる色空間である。色の三原色のため減法混色によって表される。白の状態にそれぞれシアン、マゼンタ、イエローをどの程度加えたかによって表現される。一般的にはプリンタの印刷などに利用される。なおプリンタのインクではすべて混ぜても黒がきれいに表現されないため、別途黒インクを利用している。図 3.2.4.2 に減法混色の図を示す。

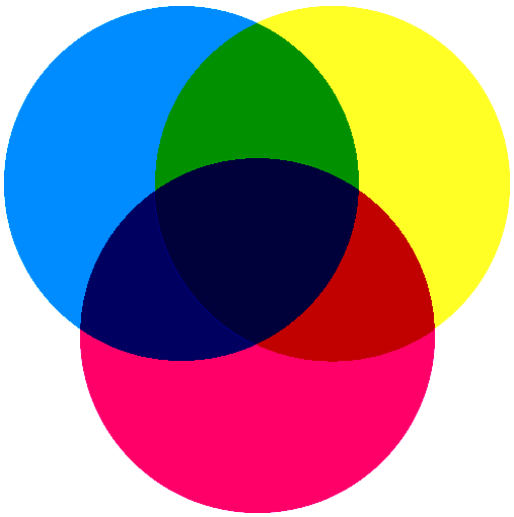


図 3.2.4.2 減法混色

3.2.4.3 HSV

HSV は色相(Hue)、彩度(Saturation)、明度(Value)の 3 つの成分からなる色空間であり 1978 年に Alvy Ray Smith により考案された RGB 色空間の非線形変換である。色相は色の種類を表し、彩度は色の鮮やかさ、また明度は色の明るさを示す。彩度が低下するとくすんだ(灰色がかった)色へと変化する。RGB に比べ、画像の明暗に左右されにくいため照明変化に強く、コンピュータビジョンでは肌色検出などに特に用いられる。図 3.2.4.3 に環状の HSV 色空間の図を示す。

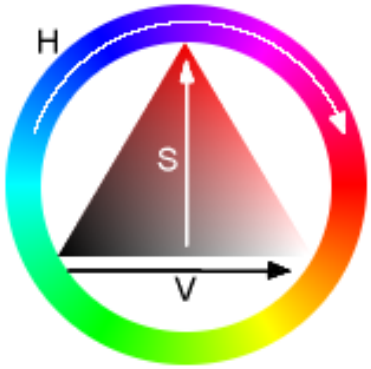


図 3.2.4.3 環状の HSV 色空間[18]

3.2.4.4 L*a*b*

CIE L*a*b*は国際照明委員会(CIE)が策定した色空間であり、L*は色の明度で黒と白の間、a*は緑と赤の間、b*は青と黄色の間に対応しており、3 次元空間でないと正しく表現できない。図 3.2.4.4 に L*a*b*色空間の図を示す。

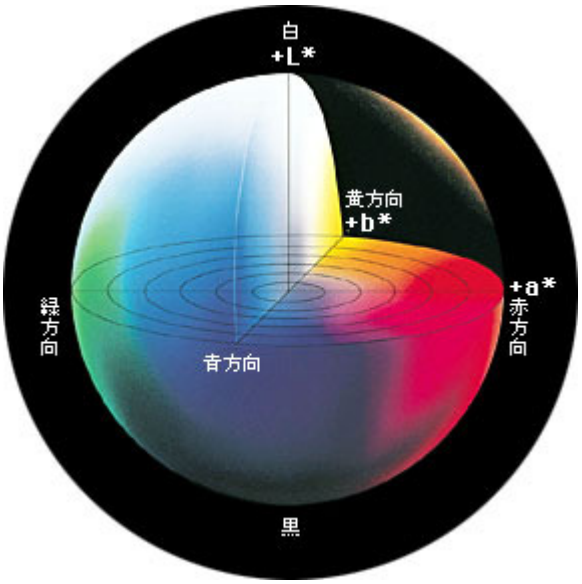


図 3.2.4.4 L*a*b*色空間[19]

3.3 顕著性マップ

顕著性マップとは視覚的注意を表現するモデルであり、物体認識や物体追跡、物体検出などに応用可能であるとされている。ここでは代表的なモデルである Itti らのモデルや物体抽出に応用が可能である Liu らのモデルについて紹介する。

3.3.1 L. Itti らのモデル

Itti らのモデル[20]は顕著性マップの先駆けのモデルとして広く知られており、受容野をシミュレートしたコントラスト差などを用いることで画像中で視覚的注意を惹きやすい領域を示す手法となっている。本節では Itti らの Saliency Map の生成方法とその出力結果について述べる。

3.3.1.1 Itti らの Salieny Map の作成方法

まず入力画像から Gaussian Pyramid によりダウンサンプリングした 9 枚のスケール画像(スケール $c \in \{0...8\}$ 、0 は元画像、8 は 1/256 に縮小)を作成する。次にそれぞれの画像の各ピクセルに対して Center-Surround のスケール間差分を求め、特徴量マップの作成を行う。ここで、求める特徴量は輝度成分 $I(c)$、色成分(赤 $R(c)$、緑 $G(c)$、青 $B(c)$、黄 $Y(c)$)、方向成分 $O(c,\theta)$ であり、RGB(式中では rgb で表記する)を用いてそれぞれ以下に示すように定義する。

$$I = (r+g+b)/3 \tag{33}$$

$$R = r-(g+b)/2 \tag{34}$$

$$G = g-(r+b)/2 \tag{35}$$

$$B = b - (r+g)/2 \tag{36}$$

$$Y = (r+b)/2 - |r-g|/2 - b \tag{37}$$

方向成分$O(c,\theta)$はGabor Filterを用いて$\theta = \{0°,45°,90°,135°\}$の4方向の成分を抽出する。本研究ではGabor Filterの窓幅は8*8を用いた。

また、Centerはスケール$c \in \{2,3,4\}$の画像における画素であり、Surroundとは$\delta \in \{3,4\}$とすると、スケール$s = c + \delta$のスケール画像の画素である。以下の式によりスケール間差分(⊖)で求める。

$$I(c,s) = |I(c) \ominus I(s)| \tag{38}$$

$$RG(c,s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \tag{39}$$

$$BY(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))| \tag{40}$$

$$O(c,s,\theta) = |(O(c,\theta) \ominus O(s,\theta))| \tag{41}$$

$I(c,s)$、$RG(c,s)$、$BY(c,s)$はcの3通りとsの2通りで6マップ、$O(c,s,\theta)$はさらにθの4通りで24マップ存在する。

さらにこれらの特徴量マップを以下の式で結合し、3chのマップを作成する。

$$\bar{I} = \oplus_c \oplus_s N(I(c,s)) \tag{42}$$

$$\overline{C} = \oplus_c \oplus_s [N(RG(c,s)) + N(BY(c,s))] \tag{43}$$

$$\overline{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} \oplus_c \oplus_s N(O(c,s,\theta)) \tag{44}$$

ここで、式中の$N(\bullet)$は各マップの正規化処理を表す。本研究では各マップの画素値を0から1に正規化し、最大値Mとそれ以外の領域での各画素の平均値$\overline{m}$を計算し、全体に$(M - \overline{m})^2$をかけることによって正規化処理を行う。これによって全体的に値の高い(各画素で差が小さい)マップは値が低くなり、局所的に値が高いマップはより値を高くすることができる。

最後に、これらのマップを次式のように線形和を求めることで顕著性(Saliency)を求めることができる。

$$Saliency = \frac{N(\bar{I}) + N(\overline{C}) + N(\overline{O})}{3} \tag{45}$$

3.3.1.2 Ittiらの Saliency Mapの出力例

図3.3.1.2に実際のSaliency Mapの出力例を示す。注意を引きやすい領域が示されているが、色への依存が非常に強く、物体抽出に用いるには困難であることがわかる。

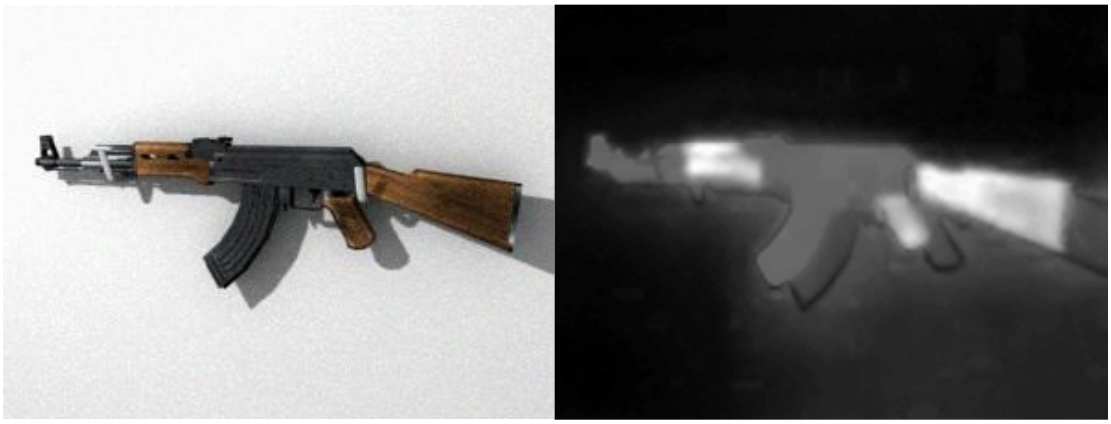


図 3.3.1 Saliency Map の出力例(左図が入力画像、右図が出力画像)

3.3.2 F. Stentiford らのモデル

Itti らの Saliency Map は近傍と特徴が異なる領域を表現していたが、Stentiford らの Saliency Map[21]は他の領域に存在しない特徴を持つ領域を表現する。Itti らの定義では背景が物体のテクスチャよりも複雑で、背景のコントラストが物体のコントラストより高い場合に背景のほうが、顕著度が高くなる可能性があるためである。

Stentiford らの Saliency Map の作成方法の流れを図 3.2.2.1 に示す。

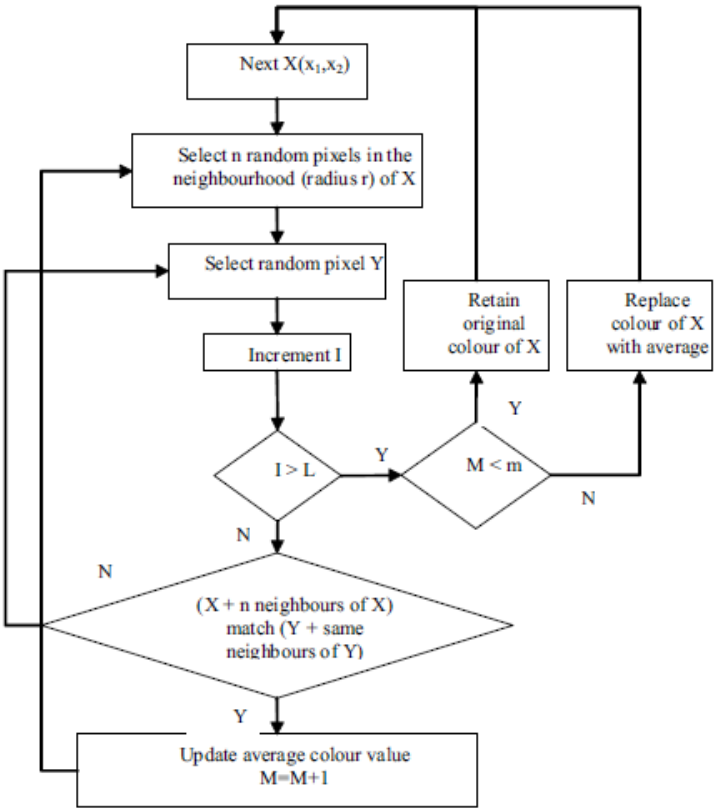


図 3.3.2.1 フローチャート

まず処理対象のピクセル x を定義し、x から半径 r の近傍領域から n 個ランダムにピクセルを選ぶ。またその他の領域からランダムにピクセル y を選択し、x の場合と同様に n 個のランダムなピクセルを選択する。加えてカウンタ I をインクリメントする。I が閾値 L の値を越えていなければ、x と y の近傍ピクセルでのマッチング処理が行われる。このマッチング処理では x の n 個全てのピクセルの色と y の n 個全てのピクセルの色が一致していれば近傍が一致したとみなし、一致していなければあらたに y を選びなおす。一致するたびに平均特徴量 avg を y の特徴量で更新し、マッチカウント M をインクリメントする。さらに新たに x の近傍から n 個のピクセルを選択する。この処理を繰り返し、y を L 回よりも多く選択した時点、つまり I が L よりも大きくなった時点で、マッチカウント M が閾値 m（通常は m=L*0.1 を用いる）を超えていれば avg を x の特徴量で置換し、新たに x を選択する。（この置換は全ピクセルの処理が終了したときに行う。）超えていなければ x の特徴量を保持して新たな x を選択する。これによって顕著な領域（M<m の場合）を保持しつつ、非顕著な領域を平滑化する。全ピクセルの処理が終了すると、非顕著な領域を各時点の avg で置き換えた平滑化画像と顕著性マップを出力する。図 3.3.2.2 のように顕著な領域は緑で、非顕著な領域は赤で出力される。実験では以下のパラメータが用いられている。
L=10,m=1,d=50,r=1,n=3

図 3.3.2.2 カラー画像の出力結果

3.3.3 T. Liu らのモデル

多くの Saliency Map では視覚的注意を引きやすい”領域”を示すことを目的としているが、T. Liu ら[22]は視覚的注意を引きやすい”物体”を示すモデルを提案している。”物体”を示すために、Liu らは center-surround をはじめとする局所的・領域的・大域的な 3 つの特徴を定義し、独自に作成したグランドツルースからそれらの特徴を線形結合するための最適な重みを学習している。

3.3.3.1 T. Liu らの Saliency Map の作成方法

前述のとおり、Liu らは重みを学習し、3 つの特徴をその重みを用いて線形結合することで Saliency Map を作成した。3.3.3.1.1 で重みの学習方法を述べた後、3.3.3.1.2 で結合される 3 つの特徴量について述べる。

3.3.3.1.1 CRF によるグランドツルースからの重み学習

「何が画像中で顕著な物体であるのか」という問題を表現するために、複数のユーザ(論文中では 3 人)によって長方形領域でラベリングされたグランドツルースを作成することで、顕著確率マップ G を作成する。

$$G = \{g_x \mid g_x \in [0,1]\} \tag{46}$$

$$g_x = \frac{1}{M}\sum_{m=1}^{M} a_x^m \tag{47}$$

ここで a_m は m 番目のユーザによるラベリング結果を意味する。

CRF (Conditional Random Field) の枠組みで対象画像 I から得られるラベル $A = \{a_x\}$ の確率を条件付分布 $P(A \mid I) = \frac{1}{Z}\exp(-E(A \mid I))$ でモデル化する。ここで Z は分配関数である。またエネルギー $E(A \mid I)$ を、K 個(論文中では 3 個)の顕著特徴量と隣接ピクセルとの特徴量の線形結合によって次式のように定義する。

$$E(A \mid I) = \sum_x \sum_{k=1}^{K} \lambda_k F_k(a_x, I) + \sum_{x,x'} S(a_x, a_{x'}, I) \tag{48}$$

ここで、λ_k は k 番目の特徴量の重みであり x, x' は 2 つの隣接ピクセルである。

F_k はピクセル x が顕著な物体に属すかそうでないかを示し、全ピクセルに対して 0 から 1 に正規化を行った各特徴マップ $f_k(x, I)$ を用いて次式のように定義する。

$$F_k(a_x, I) = \begin{cases} f_k(x, I) & a_x = 0 \\ 1 - f_k(x, I) & a_k = 1 \end{cases} \tag{49}$$

$S(a_x, a_{x'}, I)$ は 2 つの隣接ピクセルの空間的関係をモデル化したものであり、次式のように定義する。

$$S(a_x, a_{x'}, I) = \left|a_x - a_{x'}\right| \cdot \exp(-\beta d_{x,x'}) \tag{50}$$

ここで、$d_{x,x'} = \left\|I_x - I_{x'}\right\|$ は色差の L2 ノルムであり、$\beta = \left(2\left\langle \left\|I_x - I_{x'}\right\|^2 \right\rangle\right)^{-1}$ である。$\langle\cdot\rangle$ は期待値をあらわす。ただし、この項は隣接ピクセルとのラベルが異なる場合にのみ加算されるペナルティ項である。

CRF 学習の目的は特徴量の最適な線形結合を得るために重み付けベクトル $\vec{\lambda} = \{\lambda_k\}_{k=1}^{K}$ を最尤基準の下で推定することである。与えられた N 枚の学習画像 $\{I^n, A^n\}_{n=1}^{N}$ に対して、最適なパラメータは次式で示す対数尤度を最大化する。

$$\vec{\lambda}^{*} = \arg\max_{\vec{\lambda}} \sum_{n} \log P(A^n \mid I^n; \vec{\lambda}) \tag{51}$$

この対数尤度の λ_k に関する導関数は 2 つの期待値の差に等しい。

$$\frac{d \log P(A^n \mid I^n; \vec{\lambda})}{d\lambda_k} = \left\langle F_k(A^n, I^n) \right\rangle_{p(A^n \mid I^n; \vec{\lambda})} - \left\langle F_k(A^n, I^n) \right\rangle_{P(A^n \mid G^n)} \tag{52}$$

勾配降下方向は、

$$\Delta\lambda_k \propto \sum_{n} \left(\sum_{x, a_x^n} (F_k(a_x^n, I^n) p(a_x^n \mid I^n; \vec{\lambda}) - F_k(a_x^n, I^n) p(a_x^n \mid g_x^n)) \right) \tag{53}$$

ここで、$p(a_x^n \mid I^n; \vec{\lambda}) = \int_{A^n \setminus a_x^n} P(A_x^n \mid I^n; \vec{\lambda})$ は周辺分布であり、$p(a_x^n \mid g_x^n)$ はラベル付けされたグランドツルースから求める。

$$p(a_x^n \mid g_x^n) = \begin{cases} 1 - g_x^n & a_x = 0 \\ g_x^n & a_x = 0 \end{cases}$$

周辺分布 $p(a_x^n \mid I^n; \vec{\lambda})$ の正確の計算は困難であるため、pseudo-marginal を用いて近似を行う。

なお、著者らの学習結果では最もよい重みベクトルは、$\vec{\lambda} = \{0.24, 0.54, 0.22\}$ となっている。

3.3.3.1.2 顕著な物体を定義するために用いる特徴量

顕著な物体を定義するために、局所的、領域的、大域的な特徴量を定義する。

3.3.3.1.2.1 Multi-scale contrast

マルチスケールのコントラスト特徴 $f_c(x, I)$ をガウシアンピラミッドのコントラストの線形結合で定義する。

$$f_c(x, I) = \sum_{l=1}^{L} \sum_{x' \in N(x)} \left\| I^l(x) - I^l(x') \right\|^2 \tag{54}$$

ここで、I^l は(6 枚までの)ピラミッドの l 番目までの画像を表し、$N(x)$ は 9×9 の窓である。

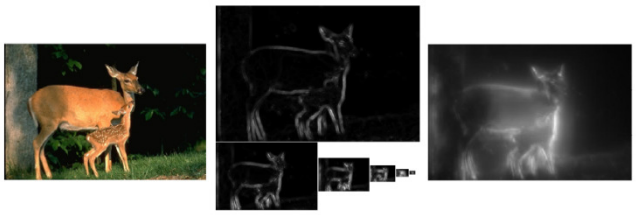


図 3.3.3.1.2.1 Multi-scale Contrast の出力結果[22]

3.3.3.1.2.2 Center-surround histogram

顕著な物体が長方形領域 R で囲まれていたとし、それを取り囲むように領域 R_s を考える。RGB ヒストグラムの χ^2 距離を用いて背景と物体の差の明瞭さを測定する。

$$\chi^2(R,R_s)=\frac{1}{2}\sum\frac{(R^i-R_S^i)^2}{R^i+R_S^i} \tag{55}$$

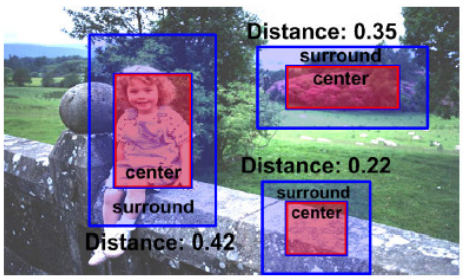
長方形領域のアスペクト比は $\{0.5,\ 0.75,\ 1.0,\ 1.5,\ 2.0\}$ とし、$R(x)$ のサイズは画像の幅と高さのうち、短いほうの長さ×0.1～0.7 で変化させる。このようにして $R(x)$ の形と大きさを変化させ、各ピクセル x を中心とする長方形領域のうち、もっとも明瞭な領域 $R^*(x)$ を探す。

$$R^*(x)=\arg\max_{R(x)}\chi^2(R(x),R_s(x)) \tag{56}$$

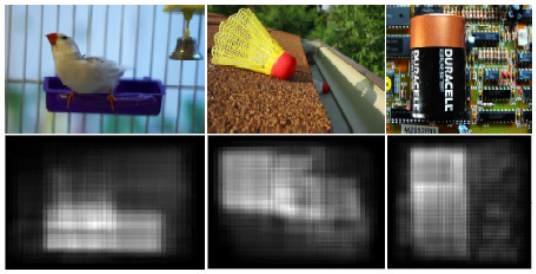
これを用いて center-surround ヒストグラム特徴 $f_h(x,I)$ は空間的距離を考慮した重み付き合計によって定義される。

$$f_h(x,I)\propto\sum_{\{x'|x\in R^*(x')\}}w_{xx'}\chi^2(R^*(x'),R_S^*(x')) \tag{57}$$

ここで、$R^*(x')$ はピクセル x' を中心にしてピクセル x を含む長方形領域である。重み $w_{xx'}=\exp(-0.5\sigma_{x'}^{-2}\|x-x'\|^2)$ は分散 $\sigma_{x'}^2$ に関する Gaussian falloff weight であり、$R^*(x')$ のサイズの 1/3 になるように設定する。最終的に特徴マップ $f_h(\cdot,I)$ も[0,1]に正規化を行う。



(a)



(b)

図 3.3.3.1.2.2(a) 異なるサイズの center-surround histogram

図 3.3.3.1.2.2(b) center-surround histogram feature の出力結果 [22]

3.3.3.1.2.3 Color spatial-distribution

画像のすべての色を GMM $\{w_c, \mu_c, \Sigma_c\}_{c=1}^{C}$ でモデル化する。カッコ内は c 番目の分布の重み、平均、分散共分散行列を表す。各ピクセルは次の確率で色分布に割り当てられる。

$$p(c \mid I_x) = \frac{w_c N(I_x \mid \mu_c, \Sigma_c)}{\sum_c w_c N(I_x \mid \mu_c, \Sigma_c)} \tag{58}$$

各色分布 c に対する空間的位置の水平方向の分散 $V_h(c)$ は

$$V_h(c) = \frac{1}{|X|_c} \sum_x p(c \mid I_x) \cdot |x_h - M_h(c)|^2 \tag{59}$$

$$M_h(c) = \frac{1}{|X|_c} \sum_x p(c \mid I_x) \cdot x_h \tag{60}$$

ここで x_h はピクセル x の x 座標であり、$|X|_c = \sum_x p(c \mid I_x)$ である。垂直方向の分散 $V_v(c)$ も同様に定義する。分布 c の空間的分散は $V(c) = V_h(c) + V_v(c)$ であり、これも[0,1]で正規化を行う。最終的な色空間分布特徴 $f_s(x, I)$ は重み付き合計で定義される。

$$f_s(x,I) \propto \sum_c p(c \mid I_x) \cdot (1 - V(c)) \tag{61}$$

$f_s(\cdot, I)$ も同様に[0,1]で正規化を行う。ただし、画像はさまざまなシーンから切り取られていて、画像の角や境界線付近では色の分散が小さいことを考慮し、画像中心を重く重み付けした場合の空間的分散特徴量は次式のように定義する。

$$f_s(x,I) \propto \sum_c p(c \mid I_x) \cdot (1 - V(c)) \cdot (1 - D(c)) \tag{62}$$

ただし、$D(c) = \sum_x p(c \mid I_x) d_x$ は画像の境界線付近の色はあまり重要でないようにする重みであり、$V(c)$ 同様に正規化を行う。d_x は画像の中心から x までの距離である。

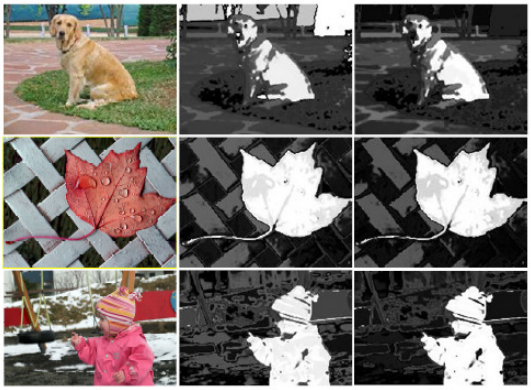


図 3.3.3.1.2.3(a) 入力画像,
(b) color spatial variance feature maps.,
(c)center-weighted, color spatial variance feature maps((62)式)[22]

3.3.3.2 T. Liu らの Saliency Map の出力例

ここでは比較のため、Liu らの Saliency Map の出力に加えて Itti らの Saliency Map も記載しておく。Itti らの手法に比べ、物体全体が顕著とされる傾向にある。1 段目や 3 段目では、Itti らの手法では物体の一部のみが顕著となってしまっているが、Liu らの出力結果では改善されている。

図 3.3.3.2 Itti らの Saliency Map と Liu らの Saliency Map の比較
(左：入力画像、中：Itti らの Saliency Map、右：Liu らの Saliency Map)

3.4 領域分割手法

前景と背景を分割するための領域分割手法としては、従来”動的輪郭モデル”と呼ばれる Snakes[23]や Level Set[24][25]などの手法が用いられてきた。これらの手法では、境界線に対してのエネルギー関数を作成し、エネルギー関数が小さくなるように境界線を変化させる手法であるため、局所解しか求めることができないという問題点を持っていた。一方で、Graph Cuts[26]では各領域からエネルギー関数を定義しているため、大域解を求めることができる。なお、本節の執筆にあたって[27][28]を参考にした。

3.4.1 Snakes

Snakes[24]は Kass らによって提案された動的輪郭モデルの手法である。陽(explicit)な境界のパラメータ表現であり、次式で与えられるエネルギー関数$E(v)$を最小化するように閉曲線 Cが決定される。

$$E(v) = S(v) + P(v) \tag{63}$$

ただし、$v(s) = [x(s), y(s)]$は図 3.4.1.1 に示すような閉曲線 C のパラメータ表現であり、$s \in [0,1]$は閉曲線 Cの弧長パラメータである。

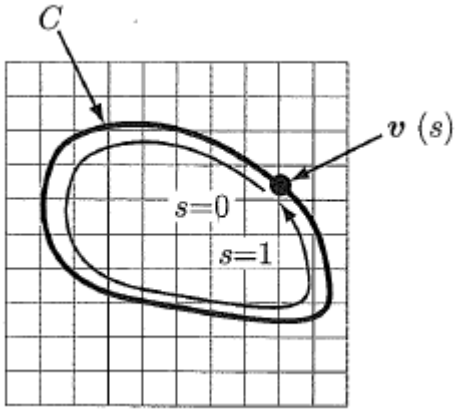


図 3.4.1.1 閉曲線 Cのパラメータ表現[27]

式(63)の$S(v)$は閉曲線 Cの内部変形エネルギーであり、次式により定義される。

$$S(v) = \frac{1}{2}\int_0^1 w_1(s)\left|\frac{\partial v}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 v}{\partial s^2}\right|^2 ds \tag{64}$$

ただし、$w_1(s), w_2(s)$はそれぞれ曲線の張力および剛性を決定する非負の関数である。例えば$w_1(s)$を増やすと、無駄なループを減らし、弧長を短くする効果が生じる。一方、$w_2(s)$を増やすと、全体として曲率の小さい、つまり硬い曲線が生成される。

これに対し、$P(v)$は外部ポテンシャルエネルギーであり、一般に以下のように定義される。

$$P(v) = \int_0^1 P[v(s)]ds \tag{65}$$

ここで$P[v(s)]$は、曲線と画像との適合度を表すポテンシャル関数であり、例えば以下のように定義される。

$$P(x,y) = -c|\nabla[G_\sigma \otimes I(x,y)]| \tag{66}$$

ただし、cは定数、∇は gradient、$G_\sigma \otimes I$はガウシアンフィルタである。これにより、曲線は画像中で濃度勾配の大きな領域で停留するように制御される。

Snakes の数値解法としては、Dynamic Programming を用いたものや greedy アルゴリズムを用いたものが有名であるが、ここでは 8bit グレースケール画像に対する最も基本的

なアルゴリズムについて述べる。

エネルギー関数 Eを局所エネルギーE_iの輪として$E = \sum E_i$のように理参加する。具体的には図 3.4.1.2 のように曲線 C上に離散化した点の集合$\boldsymbol{u} = \{\boldsymbol{v}_i\}, (\boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{N}, \boldsymbol{v_i} = (x_i, y_i))$を考え、曲線 Cを点列$\boldsymbol{u}$を結ぶ多角形で近似する。また、点$\boldsymbol{v}_i$での局所エネルギーE_iを以下のように定義する。

$$E_i = \alpha E_{cont,i} + \beta E_{curv,i} + \gamma E_{image,i} \tag{67}$$

ここで、α, β, γは適当な定数であり、

$$E_{cont,i} = |\boldsymbol{v}_{i+1} - \boldsymbol{v}_i|^2 = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \tag{68}$$

$$E_{cont,i} = \left(\bar{d} - |\boldsymbol{v}_{i+1} - \boldsymbol{v}_i|\right)^2 \tag{69}$$

$$E_{curv,i} = |\boldsymbol{v}_{i+1} - 2\boldsymbol{v}_i + \boldsymbol{v}_{i-1}|^2 \tag{70}$$

$$E_{image,i} = -\frac{I_{v_i} - I_{min}}{\max(I_{max} - I_{min,5})} \tag{71}$$

とする。ただし、$\bar{d}$は頂点間距離の平均値、$I_{\boldsymbol{v}_i}$は点$\boldsymbol{v}_i$での輝度値、I_{max}, I_{min}は点$\boldsymbol{v}_i$の隣接 8 画素のうち最大値と最小値である。また、通常$E_{cont,i}, E_{curv,i}$は、点$\boldsymbol{v}_i$の隣接 8 画素におけるそれぞれの最大値で割ることによって正規化をおこなう。

以上より、具体的なアルゴリズムは次のようになる。

1. (変数の準備) 頂点の座標を格納する変数$\boldsymbol{v}_i = (x_i, y_i), (i = 1, \dots, N)$と頂点の総移動量を表す変数$d_{total}$、平均頂点間距離を表す変数$\bar{d}$、および繰り返し回数を表す変数$n$を準備する。
2. (初期化) $n = 0$とし、対象を囲むように頂点を適当な間隔で配置する。またそのときの頂点座標を$\boldsymbol{v}_i$に格納する。
3. nに 1 を加え、$d_{total} = 0$とする。また平均頂点間距離$\bar{d}$を計算する。
4. ある頂点$\boldsymbol{v}_i$を選び、隣接 8 画素の点jで、式(67)により局所エネルギーE_iを計算する。
5. もっとも局所エネルギーの小さな近傍点新たな頂点として、頂点$\boldsymbol{v}_i$を移動させる。またそのときの頂点の移動量をd_{total}に加える。
6. 4 から 5 をすべての頂点に対して行う。
7. 3 から 6 を頂点の総移動量d_{total}が閾値以下になるか、nがあらかじめ決められた繰り返し回数を超えるまで反復する。

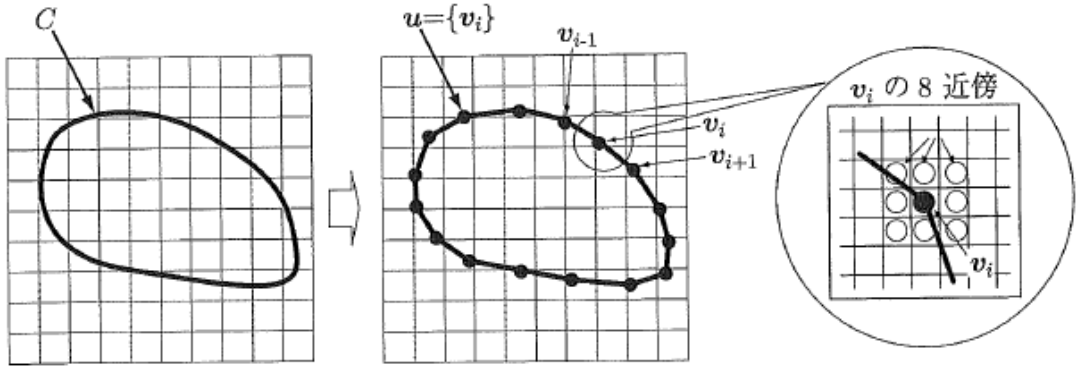


図 3.4.1.2 Snakes の実装法[27]

3.4.2 Level Set

Level Set Method は Osher, Sethian ら[24][25]によって提案された位相変化が可能な動的輪郭モデルであり、Snakes と同じ Active Contour Model であるが、領域の分離や結合を自然な形で表現できる。Level Set では曲線の状態(収縮、膨張、曲率変化など)を偏微分方程式によって表現し、強化の進行を偏微分方程式の解として陰(implicit)に表現するものである。

対象としている空間(画像では 2 次元)に対して、一つ高い次元の仮想空間を設定し、境界をその高次元空間で定義された関数の切断面としてとらえる。時刻tにおいて 2 次元空間上で、境界 Cで囲まれた領域Ωを考えると、Level Set では高次元(3 次元)空間で定義された補助関数ϕのゼロ等高面(zero level set)$\phi = 0$と考える。次に時刻$t + \Delta t$において、この補助関数ϕを、$\phi \leftarrow \phi + \delta$のようにその形状を保ったまま$\phi$の正方向へ移動させ、同様にゼロ等高面を切り出す。この時、補助関数ϕが図 3.4.2 のように双峰的である場合にはゼロ等高面は 2 つの領域Ω_1, Ω_2となり、境界も 2 つとなる。そのように補助関数の形状を領域の特徴に応じて適切に設計し、制御することで、補助関数に対してなめらかな形状を保ちつつ、自然な形で境界の分離、結合が表現できる。補助関数ϕの初期値には初期境界 Cからの符号付距離が用いられる。

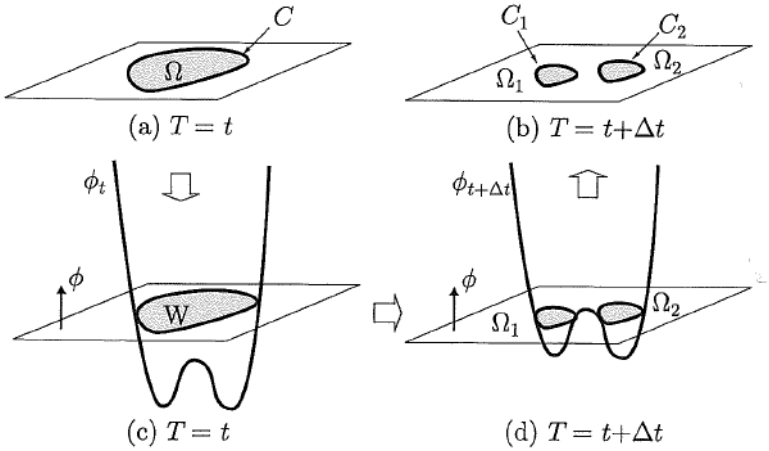


図 3.4.2 Level Set Method の考え方[27]

次に実際の境界線の検出問題を考える。対象となる 2 次元画像を$I(x,y) \in \boldsymbol{R}^2$とし、時刻$t$での境界線を$C(\boldsymbol{p},t)$とする。ただし、$\boldsymbol{p}=(p_x,p_y)$である。この境界に含まれる点$\boldsymbol{p}$は、移動速度$F(\kappa)$で境界線の法線方向$\boldsymbol{N}$に移動していると考える。ここで、$\kappa$はその点での境界線の曲率であり、$F(\kappa)$を成長速度という。これを式で表すと、

$$C_t = F(\kappa)\boldsymbol{N} \tag{72}$$

$$C(\boldsymbol{p},0) = C_0(\boldsymbol{p}) \tag{73}$$

となる。ただし、C_tは境界Cの時間変化、$C_0(\boldsymbol{p})$は初期曲線である。前述の Snakes では差分方程式を利用したラグランジュ法で解くことができるが、トポロジーの変化には対応できないという問題が残る。そこで図 3.4.2 に示したように、補助関数$\phi(x,y,t)$を導入し、境界線$C(\boldsymbol{p},t)$はその関数の一部、すなわち$\phi(x,y,t)=0$を満たすϕで表されると考える。ここで、点$\boldsymbol{p}(t)$が境界線$C(\boldsymbol{p},t)$上の点である場合、これが常に$\phi(x,y,t)$のゼロ等高面上である条件は、

$$\phi[\boldsymbol{p}(t),t] = 0 \tag{74}$$

で表される。これを偏微分すると、

$$\phi_t + \nabla\phi[\boldsymbol{o}(t),t]\boldsymbol{p_t} = 0 \tag{75}$$

となる。また曲線状の単位法線ベクトルは

$$\boldsymbol{N} = \frac{\nabla\phi}{|\nabla\phi|} \tag{76}$$

で表され、さらに成長速度$F(\kappa)$は境界$C(\boldsymbol{p},t)$の法線方向速度であるから、

$$\boldsymbol{p}_t \cdot \boldsymbol{N} = F(\kappa) \tag{77}$$

となる。これにより式(75)は次のように書くことができる。

$$\phi_t = -F(\kappa)|\nabla\phi| \tag{78}$$

$$\phi[C_0(\boldsymbol{p}),0] = 0 \tag{79}$$

このように境界$C(\boldsymbol{p},t)$を直接的に移動する代わりに、補助関数$\phi(x,y,t)$を更新し、$\phi(x,y,t)=0$を満たす線を新たな境界線とすることで、トポロジーの変化に対応した領域追

跡が可能となる。

3.4.3 Graph Cuts

近年、高精度な領域分割手法として注目されている手法に Graph Cuts がある。Graph Cuts は領域分割問題をエネルギー最小化の問題と捉えて解く手法であり、このような手法としては Graph Cuts 以外では前述の Snake などの動的輪郭モデル、Level Sets などが挙げられる。Snake や Level Sets は境界線に対してのエネルギー関数を作成し、エネルギー関数が小さくなるように境界線を変化させる手法であるため、局所解しか求めることができない。Graph Cuts では各領域からエネルギー関数を定義しているため、大域解を求めることができる。

Graph Cuts を用いた領域分割手法として、Boykov らによって Interactive Graph Cuts が提案されている。Interactive Graph Cuts では、ユーザが与えた seed と呼ばれる前景か背景かを示すピクセルと入力画像からグラフを作成し、minimum cut/maximum flow algorithm を用いることでエネルギー関数の最小化を行う。またこの Interactive Graph Cuts を拡張した手法として、繰り返し処理により前景と背景の色分布をセグメンテーション結果から再学習し、繰り返しセグメンテーションを行う GrabCut などが提案されている。

ここでは従来手法の Interactive Graph Cuts について説明する。

画像 P に対する各ピクセル(サイトと呼ばれる)を $p \in P$ 、ラベルを $L = \{L_1, L_2, ..., L_p, ..., L_{|P|}\}$ とし、各 L_p には物体(obj)か背景(bkg)かのラベルが与えられる。また、p の近傍ピクセルを $q \in N$ とする。Graph Cuts ではエネルギー関数を次式のように定義する。

$$E(L) = \lambda \cdot R(L) + B(L) \tag{80}$$

ここで、λ は $R(L)$ と $B(L)$ の比率のパラメータである。$R(L)$ はデータ項と呼ばれる領域に関するペナルティ関数、$B(L)$ は平滑化項と呼ばれる物体と背景の境界に対するペナルティ関数であり、それぞれ以下に示すように定義する。

$$R(L) = \sum_{p \in P} R_p(L_p) \tag{81}$$

$$B(L) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(L_p, L_q) \tag{82}$$

$$\delta(L_p, L_q) = \begin{cases} 1 & if \ \ L_p \neq L_q \\ 0 & otherwise \end{cases} \tag{83}$$

$R_p(L_p)$ は、ピクセル p がラベル L_p である確率が高ければ値が小さくなるような関数として定義し、$B_{\{p,q\}}$ は、p と q の輝度値が似ていれば大きな値を出力する関数として定義する。

$R(L)$ と $B(L)$ により定義したエネルギー関数 $E(L)$ を最小とするようなラベル L を Graph Cuts Algorithm を用いて計算することで領域分割を行う。(80)式のようなエネルギーはマルコフ確率場(Markov Random Field: MRF)の最大事後確率(MAP)推定を行う際によく現れる。

Graph Cuts Algorithm では、画像から図 3.4.3.1 のようなグラフを作成し、min-cut / max-flow algorithm を用いることで分割を行う。グラフ G は、画像の各ピクセルに対応したノードと source と sink と呼ばれるターミナルからなり、各ノード間を接続するエッジを n-link、各ノードと source(S)と sink(T)のターミナルを接続するエッジを t-link と呼ぶ。n-link と t-link のエッジコストは表 3.4.3.1 のように設定する。

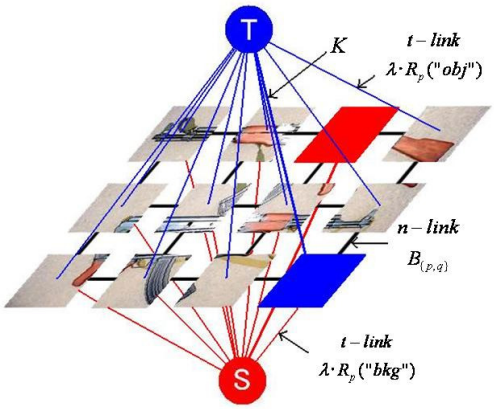


図 3.4.3.1 グラフの作成

表 3.4.3.1 各エッジの重み

edge		*cost*	*pixel*
n-link	$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$
t-link	$\{p,S\}$	$\lambda \cdot R_p("bkg")$	$p \in P, p \notin O \cup B$
		K	$p \in O$
		0	$p \in B$
	$\{p,T\}$	$\lambda \cdot R_p("obj")$	$p \in P, p \notin O \cup B$
		0	$p \in O$
		K	$p \in B$

このとき、

$$R_p("obj") = -\ln \Pr(I_p \mid O) \tag{84}$$

$$R_p("bkg") = -\ln \Pr(I_p \mid B) \tag{85}$$

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)} \tag{86}$$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}} \tag{87}$$

となる。ここでOは物体、Bは背景を表し、I_pはピクセルpの輝度値を表す。$\Pr(I_p \mid O)$、$\Pr(I_p \mid B)$はseed以外のピクセルのt-linkに設定する物体と背景の尤度を表し、$dist(p,q)$はピクセルp、q間のユークリッド距離を表す。また、ユーザは一部のピクセルにseedと呼ばれる物体か背景かを表すO、Bを入力する。このようにして作成したグラフに対し、min-cut / max-flow algorithm[29]を適用し、エッジのコストの総和が最小となるような切断を見つけることで領域分割を行う。文献[26]ではヒストグラムを用いてグレースケールで領域分割を行っていたが、これを改良した文献[30]ではカラー画像で行うために、色分布のモデルとしてガウス混合モデル(Gaussian Mixture Model: GMM)を用いている。

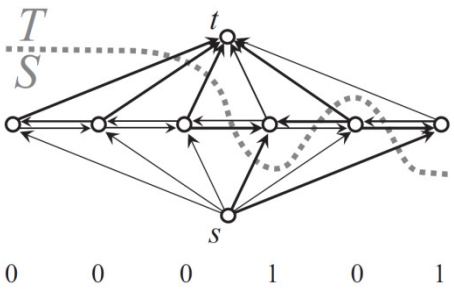


図 3.4.3.2 値 MRF 最小化のグラフ[28]

3.5 クラスタリング手法

クラスタリングとは、データ解析手法の一種であり、教師信号なしに入力データの似たもの同士をグループする手法である。教師なし学習ともよばれ、k-means 法や主成分分析、自己組織化マップなどが該当する。ここでは、一般物体認識に用いられる k-means 法とその派生手法である x-means 法について述べる。

3.5.1 k-means

k-means 法は非階層型クラスタリング手法の 1 つであり、単純なアルゴリズムで計算できることができ、広く用いられている。一般的なアルゴリズムは次の通りである。

1. 各データ $x_i(i=1,...,n)$ に対してランダムにクラスタを割り振る。

2. 割り振ったデータから、各要素の平均を用いて各クラスタの中心(セントロイド) $V_j(j=1,...,K)$ を計算する。

3. 各 x_i と各 V_j との距離を求め、x_i をもっとも近い中心のクラスタに割り当てなおす。

この 2 と 3 の処理を繰り返し行い、全ての x_i のクラスタ割り当てが変化しなくなったときに終了する。k-means 法の欠点としては最初のランダムなクラスタ割り当てに依存することが挙げられ、これにより 1 回で最良の結果が得られるとは限らない。

本研究では SIFT の Visual Word を作成する際、また Self-Similarity の Visual Word を作成する際のベクトル量子化に k-means 法を使用する。

3.5.2 x-means

x-means 法は k-means 法の逐次繰り返しと BIC(Bayesian Information Criterion：ベイズ情報量基準)によってクラスタ数を自動的に決定するアルゴリズムである。一般的なアルゴリズムは次のとおりである。

0. 解析すべきデータとして n 個の p 次元データを用意する
1. 十分に小さなクラスタ数の初期値 k_0 (特に指定しなければ 2)を定める
2. $k=k_0$ として k-means を適用する。分割後のクラスタを
$$C_1, C_2, \ldots, C_{k_0}$$
とする
3. $i=1,2,\ldots,k_0$ とし、手順 4～9 を繰り返す
4. クラスタ C_i に対して $k=2$ として k-means を適用する。分割後のクラスタを
$$C_i^1, C_i^2$$
とする

5. C_i に含まれるデータ x_i に p 変量正規分布

$f'(\theta_i;x)=(2\pi)^{-p/2}\left|V_i\right|^{-1/2}\exp\left[-\frac{1}{2}(x-\mu_i)^t V_i^{-1}(x-\mu_i)\right]$ を仮定し、そのときの BIC (Bayesian Information Criterion: ベイズ情報量基準) を以下により計算する。

$$BIC=-2\log L(\hat{\theta}_i;x_i\in C_i)+q\log n_i$$

ここで、

$\theta_i=\left[\hat{\mu}_i,\hat{V}_i\right]$ は p 変量正規分布の最尤推定値とする。μ_i は p 次の平均値ベクトル、V_i は $p\times p$ の分散共分散行列である。

q はパラメータ空間の次元数で、V_i の共分散を無視すれば $q=2p$ である。共分散を無視しなければ $q=p(p+3)/2$ である。

x_i はクラスタ C_i に含まれる p 次元データとし、n_i は C_i に含まれるデータ数とする。

L は尤度関数であり、$L(\cdot)=\prod\prod f(\cdot)$ である。

6. C_i^1,C_i^2 のそれぞれに対して、パラメータ θ_i^1,θ_i^2 をもつ p 変量正規分布を仮定し、2 分割モデルにおいてデータの従う確率密度を

$$x_i\sim\alpha_i\left[f(\theta_i^1;x)\right]^{\delta_i}\left[f(\theta_i^2;x)\right]^{1-\delta_i}$$

とおく。ここで

$$\delta_i=\begin{cases}1, & x_i\text{が}C_i^1\text{に含まれるとき}\\ 0, & x_i\text{が}C_i^2\text{に含まれるとき}\end{cases}$$

とする。また α_i は基準化定数であり、

$$\alpha_i=1/\int\left[f(\theta_i^1;x_i)\right]^{\delta_i}\left[f(\theta_i^2;x_i)\right]^{1-\delta_i}dx$$

である $(1/2\le\alpha_i\le 1)$。しかし、厳密に α_i を求めようとすると p 次積分が必要となってしまい、計算量が膨大となるため、

$$\alpha_i=0.5/K(\beta_i)$$

により、近似を行う。ここで、$K(\cdot)$ は標準正規分布の下側確率とし、β_i は $f(\theta_i^1;x_i)$ と $f(\theta_i^2;x_i)$ の分離の程度を示す指標で、

$$\beta_i=\sqrt{\frac{\left\|\mu_1-\mu_2\right\|^2}{\left|V_1\right|-\left|V_2\right|}}$$

で示すものとする。

この 2 分割モデルにおける BIC を以下により計算する

$$BIC'=-2\log L(\hat{\theta}_i';x_i\in C_i)+q'\log n_i$$

ここで、$\hat{\theta}_i' = \left[\hat{\theta}_i^1, \hat{\theta}_i^2\right]$は、2 つの p 変量正規分布の最尤推定値である。共分散を無視すれば、パラメータ空間の次元は $q' = 2 \times 2p = 4p$ となる。共分散を無視しなければ $q' = 2q = p(p+3)$ である。

7. BIC>BIC'ならば、2 分割モデルをより好ましいと判断し、2 分割を継続すべく、
$$C_i \leftarrow C_i^1$$
とする。C_i^2 については、p 次元データ、クラスタの重心、対数尤度と BIC を保持し、これらをスタックに積み、手順 4 へ
8. BIC<BIC'ならば、2 分割しないモデルをより好ましいと判断し、C_i^1 についての 2 分割を停止する。手順 7 で作成されたスタックからデータを取り出し、
$$C_i \leftarrow C_i^2$$
とし、手順 4 へ。スタックが空なら次の手順へ
9. C_i における 2 分割が全て終了。手順 4～8 で作成された 2 分割のクラスタが C_i 内で一意になるようにデータの属するクラスタ番号を振りなおす
10. はじめに k_0 分割したクラスタ全てについて 2 分割が終了。全データに対してそれらの属するクラスタ番号が一意になるように番号を振りなおす。
11. 全データの属するクラスタ番号、および各クラスタの重心、各クラスタに含まれるデータ数を出力する

3.6 識別器

識別とは教師あり学習と呼ばれる学習によって得た情報をもとに、未知の入力を分類することを指しており、パターン認識やコンピュータビジョンの世界では非常に重要な今日のパターン認識の世界では、Naïve Bayes や Linear Discriminant Analysis、AdaBoost や Support Vector Machine など、様々な識別器が用いられてきた。本節では、一般物体認識でしばしば用いられる SVM に関して述べる。なお、本節の執筆にあたって[31]を参考にした。

3.6.1 Support Vector Machine

SVM は 2 値分類問題を解くために考えられた、高次元特徴空間において線形空間を用いる学習システムである。基本的には線形の識別器であるが、カーネル関数と最適化法との組み合わせにより非線形の識別器に拡張されている。ニューラルネットワークに比べてパターン認識結果が優れていることが報告されてから多くの研究が行われてきた。ニューラルネットワークは学習アルゴリズムに渡すパラメータの初期値に最終的な解が依存してしまう局所解の問題を抱えていたが、SVM は非線形に識別を扱えるが局所解の問題がないと

いう利点がある。以下では線形 SVM とそれを応用した非線形 SVM の概要について述べる。

3.6.1.1 線形ハードマージン SVM

入力空間 $\chi \in \mathbf{R}^n$ およびデータ集合 $x_1,...,x_r$ が与えられたとすると、線形 SVM の識別関数は次式で表される。

$$f(x) = \mathbf{w}^T x + b \tag{88}$$

ここで、$\mathbf{w}$ は自由度の係数であり線形識別器の重みベクトルと呼ばれる。b はバイアス項と呼ばれる非負のパラメータである。

$f(x) = 0$ を満たす任意の $d-1$ 次元の超平面識別関数は次のように表現される。

$$\{x \in \chi : (\mathbf{w}^T x) + b = 0\} \tag{89}$$

(89)式を図示すると図 3.6.1.1.1 のようになり、これは 2 次元の入力空間 χ にデータが観測された様子を表している。ここで、異なるクラスの学習データが、$n-1$ 次元の超平面で分離できるとする。しかし(89)式からでは 0 でない定数 c を $\mathbf{w}$ および b にかけたもの全てが線形識別関数として導かれてしまい、学習データを完全に識別する超平面は無数に存在することになってしまう。2 値分類問題は、学習データを完全に識別することではなく、将来のデータをできるだけ正しく識別できることである。SVM では、学習データを完全に識別できる超平面の中で最適超平面(図 3.6.1.1.2)は、2 クラスの真ん中を通る超平面であるとする。このような超平面を見つけるために、識別超平面と学習データとの最短距離(マージン)を評価関数とし、この関数を最大にするという手法を用いる。この評価関数を最大化することによって 2 クラスへの距離が自動的にバランスされ、2 クラス間の真ん中を通る最適超平面を見つけることができる。

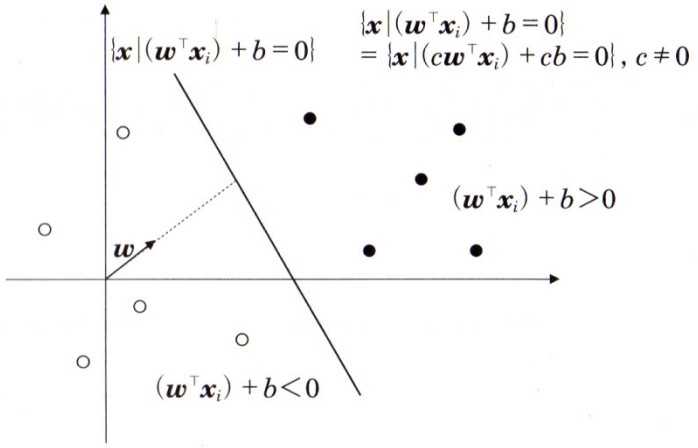


図 3.6.1.1.1　線形識別関数[31]

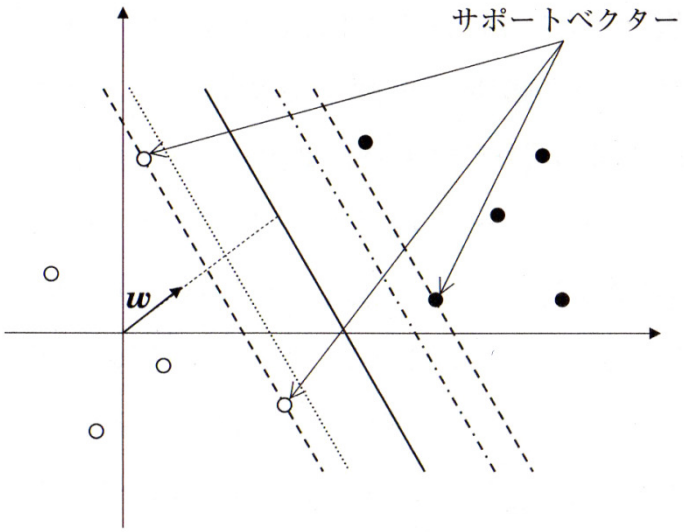


図 3.6.1.1.2　最適超平面の探索[31]

評価関数を最大化し、見つけられた識別超平面に対して最短距離となる学習データは一般的に一つではないが、最適超平面から最短距離にある学習データさえあれば、評価関数を最大にすることによって同じ最適超平面を見つけることができる。

SVM の学習を定式化し、最適超平面の探索が 2 次計画問題に帰着することを示す。上の式で 0 でない定数cを$\mathbf{w}$およびbにかけたものは表現する超平面が変化しないことから冗長性を有していると言え、このような冗長性が有ると学習結果が定まらない。そこで次式で示される制約を加えることによって、識別関数となる超平面を定数cの掛からない$(\mathbf{w},b)\in\chi\times\mathbf{R}$を有する関数になるようにする。

$$\min_{i=1,\ldots,r}\left|(\mathbf{w}^T x_i)+b\right|=1 \tag{90}$$

学習データと識別超平面のマージンは

$$\min_{i=1,\ldots,r}\frac{\left|(\mathbf{w}^T x_i)+b\right|}{\|w\|} \tag{91}$$

と表され、制約条件より$\mathbf{w}$とbは距離

$$\frac{1}{\|\mathbf{w}\|} \tag{92}$$

を持つ識別超平面に最も接近する学習データ点を表現することになる。ここで学習データ

$(x_1,y_1),\ldots,(x_l,y_l),x_i\in\chi,y_i\in\{\pm 1\},i=1,\ldots,l$が与えられ、次式

$$f_{\mathbf{w},b}(x_i)=y_i,\quad i=1,\ldots,l \tag{93}$$

を満たす識別関数

$$f_{\mathbf{w},b} = \mathrm{sgn}((\mathbf{w} \cdot x) + b) \tag{94}$$

を推定する問題を考える。この関数が存在することにより、制約は次式のように表現できる。

$$y_i((\mathbf{w}^T x_i) + b) \geq 1, \quad i = 1,...,l \tag{95}$$

超平面を決定するパラメータ $\mathbf{w}$ と b は、学習データを完全に識別する超平面の中なら、評価関数(マージン)を最大化するように決定するので、(95)式で表現される制約条件の下で、次式を最小化することで推定できる。

$$\tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{96}$$

この目的関数 $\tau(\mathbf{w})$ の最小化は(92)式の最大化を意味し、(95)式の制約条件は、最適化問題の解として得られる超平面が学習データを完全に識別できることを示す。ここでこの凸最適化問題を解くために(96)式のラグランジュ関数を計算する。(95)式は次式のように書き換えられる。

$$1 - y_i((\mathbf{w}^T x_i) + b) \leq 0 \tag{97}$$

この制約条件から、制約関数 $g_i(x), i = 1,...,l$ を $g_i(x) = 1 - y_i \cdot ((\mathbf{w}^T x_i) + b), i = 1,...,l$ とし、

この制約関数を次式で表されるラグランジュ関数

$$L(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{w}) + \sum_{j=1}^{l} \beta_j h_i(\mathbf{w}) \tag{98}$$

に代入すると、次式のように表される。

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((\mathbf{w}^T x_i) + b) - 1) \tag{99}$$

ここで $\alpha_i \geq 0$ はラグランジュ乗数である。このラグランジュ関数を α_i について最大化し、

$\mathbf{w}$ と b について最小化することで最適化問題を解くことができる。最適解においては、パラメータ $\mathbf{w}$ と b についての L の導関数は、鞍点において L の勾配が 0 となることから次式が成立する。

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = 0 \tag{100}$$

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = 0 \tag{101}$$

これより次式が成立する。

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{102}$$

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{103}$$

ここで、クーン・タッカー(Kuhn－Tucker)条件より、鞍点においては以下の条件が満たされる。

$$\begin{cases} \alpha_i \cdot \left[1 - y_i((\mathbf{w}^T x_i) + b)\right] = 0, & i = 1,...,l \\ 1 - y_i((\mathbf{w}^T x_i) + b) \le 0, & i = 1,...,l \\ \alpha_i \ge 0, & i = 1,...,l \end{cases} \tag{104}$$

この条件を満たし、$\alpha_i \ge 0$を有する学習データx_iをサポートベクターという。これよりサポートベクターは次式を満たす。

$$\mathbf{w}^T x + b = 1 \tag{105}$$

つまり、サポートベクター以外の学習データは(103)式の展開項の部分には現れない。
(99)式のラグランジュ関数に(102)(103)式の条件を代入すると凸最適化問題を得ることができる。

目的関数　$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \rightarrow \alpha$ について最小化　(106)

$\alpha_i \ge 0, \quad i = 1,...,l$

制約条件　$\sum_{i=1}^{l} \alpha_i y_i = 0$　(107)

最適なαから$\mathbf{w}$を得るには、(103)式の関係を用いる。bは

$$b = -\frac{1}{2}(\mathbf{w}^T x_{+1} + \mathbf{w}^T x_{-1}) \tag{108}$$

で求められる。ここでx_{+1}, x_{-1}はそれぞれクラス 1,-1 に属するサポートベクターである。

(103)の展開式を識別関数の(93)式に代入することにより、(93)式の識別関数を次式のように書き換えることができる。

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i (x^T x_i) + b \right) \tag{109}$$

この式は分類されるパターンとサポートベクターとの内積で評価される。以上より、(106)(107)式で表現される凸 2 次計画問題を解くことで次式の識別関数

$$f_{\mathbf{w},b}(x) = \mathrm{sgn}((\mathbf{w}^T x) + b) \tag{110}$$

を得ることができる。これが基本となる線形ハードマージン SVM である。

3.6.1.2 線形ソフトマージン SVM

現実問題として、学習データを完全分離できる超平面は存在しないことがほとんどである。そこで、次式で表現される緩和変数(スラック変数とも呼ぶ)を導入して、制約条件を満たさない学習データが存在してもよいようにする。

$$\boldsymbol{\xi}_i \geq 0,\ i = 1, \ldots, l \tag{111}$$

この緩和変数を使って制約条件を次式のように緩和する。

$$y_i\left((\boldsymbol{w}^T\boldsymbol{x}_i) + b\right) \geq 1 - \boldsymbol{\xi}_i,\ \mathrm{i} = 1, \ldots, \mathrm{l} \tag{112}$$

これは図に示すように、$\boldsymbol{\xi}_i$の値によっては$y_i\left((\boldsymbol{w}^T\boldsymbol{x}_i) + b\right)$の値が 0 に近くても制約条件を満たす場合があることを示している。図中の正方形は学習データの中で誤分類されてもよいデータを意味し、ξ_iに値があるデータを表す。図中の記号の色、黒と白はラベルを表す。このように、緩和変数を導入することで、制約条件を満たさない学習データが存在してもよいようにする。

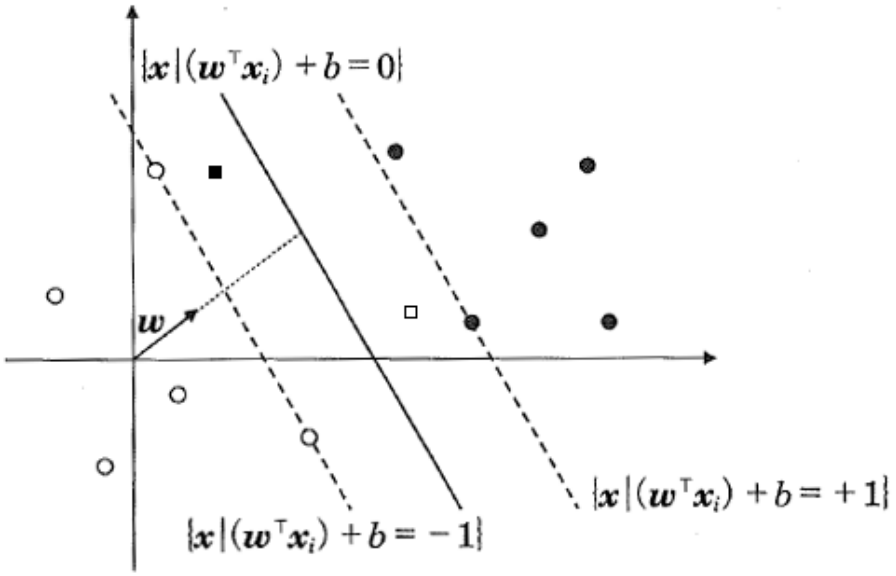


図 3.6.1.2.1 ソフトマージンにおける制約付き線形識別関数[31]

この緩和変数の導入によって、凸最適化問題は次式のようになる。

目的関数 $\tau(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \gamma\sum_{i=1}^{l}\xi_i \rightarrow \boldsymbol{w}, \boldsymbol{\xi}$ について最小化

制約条件 $y_i\left((\boldsymbol{w}^T\boldsymbol{x}_i) + b\right) \geq 1 - \boldsymbol{\xi}_i,\ \mathrm{i} = 1, \ldots, \mathrm{l}$ (113)

$\sum_{i=1}^{l}\boldsymbol{\xi}_i$は学習データ中で誤分類されるパターンの上限値である。

ハードマージン SVM と同様に、この凸最適化問題を解くため、ラグランジュ関数を計算する。制約条件は次式のように書き換えることができる。

$$1 - \boldsymbol{\xi}_i - y_i\left((\boldsymbol{w}^T\boldsymbol{x}_i) + b\right) \leq 0 \tag{114}$$

この制約条件から、制約関数$g_i(\boldsymbol{x}), i = 1, \dots, l$を$g_i(\boldsymbol{x}) = 1 - \xi_i - y_i\big((\boldsymbol{w}^T x_i) + b\big), i = 1, \dots, l$とするとラグランジュ関数は

$$\mathrm{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \gamma \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i \big(y_i\big((\boldsymbol{w}^T \boldsymbol{x}_i) + b\big) - 1 + \xi_i\big) \tag{115}$$

となる。ここで、$\alpha_i \geq 0$はラグランジュ乗数である。最適化問題を解くには、このラグランジュ関数をα_iについて最大化し、$\boldsymbol{w}, b, \boldsymbol{\xi}$について最小化する。

最適解においてはパラメータ$\boldsymbol{w}, b, \boldsymbol{\xi}$についてのLの導関数は鞍点において、Lの勾配が0となることから次式が成立する。

$$\frac{\partial}{\partial b} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = 0 \tag{116}$$

$$\frac{\partial}{\partial \boldsymbol{w}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = 0 \tag{117}$$

$$\frac{\partial}{\partial \boldsymbol{\xi}} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = 0 \tag{118}$$

これらの3つの式からそれぞれ次式が成立する。

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{119}$$

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i \tag{120}$$

$$\alpha_i = \gamma \tag{121}$$

結局、$\mathbf{w}$は学習データの展開式となる。$\mathbf{w}$の解はただ一つに決まるが、ラグランジュ乗数α_iはその必要がない。

クーン・タッカー条件により、鞍点において、以下の条件が満たされる。

$$\left.\begin{array}{c} \alpha_i \cdot \big[1 - \xi_i - y_i\big((\boldsymbol{w}^T \boldsymbol{x}_i) + b\big)\big] = 0, \ \ i = 1, \dots, l \\ 1 - \xi_i - y_i\big((\boldsymbol{w}^T \boldsymbol{x}_i) + b\big) \leq 0, \ \ i = 1, \dots, l \\ 0 \leq \alpha_i \leq \gamma, \ \ i = 1, \dots, l \end{array}\right\} \tag{122}$$

上記の条件を満たし、$\alpha_i > 0$かつ$\xi_i = 0$を有する学習データ$\boldsymbol{x}_i$をサポートベクターと呼び、$\alpha_i = 0$となる学習データは凸最適化問題の解法には関係のないものとなる。つまり、サポートベクター以外の学習データは$\mathbf{w}$で表される学習データの展開項には現れない。

式(99)のラグランジュ関数に式(119)~(120)の条件を代入すると、双対問題である以下の凸最適化問題が得られる。

目的関数$\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \big(\boldsymbol{x}_i \cdot \boldsymbol{x}_j\big) \rightarrow \boldsymbol{\alpha}$について最小化

制約条件$\begin{array}{c} 0 \leq \alpha_i \leq \gamma, \ \ \mathrm{i} = 1, \dots, \mathrm{l} \\ \sum_{i=1}^{l} \alpha_i y_i = 0 \end{array}$ (123)

最適な$\boldsymbol{\alpha}$から$\mathbf{w}$を得るには式(120)の関係を用いる。また、bはハードマージンSVM同様

$$\mathrm{b} = -\frac{1}{2}(\boldsymbol{w}^T \boldsymbol{x}_{+1} + \boldsymbol{w}^T \boldsymbol{x}_{-1}) \tag{124}$$

で求められる。ここで、$\boldsymbol{x}_{+1}, \boldsymbol{x}_{-1}$は、それぞれクラス1、-1に属するサポートベクターである。

(120)の展開式を識別関数の式(93)に代入することによって、識別関数を分類されるパタ

ーンとサポートベクターの内積で評価される次式に書き換えることができる。

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i (\boldsymbol{x}^T \boldsymbol{x_i}) + b\right) \tag{125}$$

以上より、式(113)で表される凸 2 次計画問題を解くことで識別関数

$$f_{w,b}(\boldsymbol{x}) = \mathrm{sgn}\left((\boldsymbol{w}^T \boldsymbol{x}) + b\right) \tag{126}$$

を得ることができる。これがソフトマージン SVM である。

3.6.1.3 非線形ハードマージン SVM

線形 SVM は線形分離可能な場合には高い汎化能力を達成できるが、実際の問題では線形分離可能な場合は多くない。そこでより一般的な識別関数を推定するため、前処理としてベクトル $x_1,...,x_l$ を次式のように高次元特徴空間に写像し、その特徴空間で線形 SVM を行う方法が考えられる。

$$\Phi : x_i \mapsto z_i \tag{127}$$

ここで、z_i は観測された入力ベクトル x_i を高次元特徴空間に写像したものである。z_i による SVM を考えると制約条件を表す(95)式と、目的関数を表す(96)式で表現される特徴空間上での最適化問題は z_i を用いて次式のように表せる。

目的関数　$\tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \rightarrow \mathbf{w}$ について最小化

制約条件　$y_i((\mathbf{w}^T z_i) + b) \geq 1, \quad i = 1,...,l$ (128)

線形 SVM と同様に、この最適化問題を解くために(128)式のラグランジュ関数を計算する。(128)式は次式のように書き換えられる。

$$1 - y_i((\mathbf{w}^T z_i) + b) \leq 0 \tag{129}$$

これより、ラグランジュ関数に制約条件 $g_i(x), i = 1,...,l$ を $g_i(z) = 1 - y_i((\mathbf{w}^T z_i) + b), i = 1,...,l$ として代入するとラグランジュ関数は次式のようになる。

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i (y_i((\mathbf{w}^T z_i) + b) - 1) \tag{130}$$

最適解においては、パラメータ $\mathbf{w}$ と b についての L の導関数は鞍点において L の勾配が 0 になることから次式が成立する。

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{131}$$

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i z_i \tag{132}$$

クーン・タッカー条件から鞍点においては以下の条件が満たされる。

$$\begin{cases} \alpha_i \cdot \left[1 - y_i((\mathbf{w}^T z_i) + b)\right] = 0, & i = 1,...,l \\ 1 - y_i((\mathbf{w}^T z_i) + b) \le 0, & i = 1,...,l \\ \alpha_i \ge 0, & i = 1,...,l \end{cases} \tag{133}$$

上記の条件を満たし、$\alpha_i > 0$を有する高次元特徴空間に写像された学習データをサポートベクターと呼ぶ。サポートベクターは以下の条件を満たす。

$$\mathbf{w}^T z + b = 1 \tag{134}$$

線形 SVM 同様、$\alpha_i = 0$となるサポートベクター以外の学習データは最適化問題の解法には関係のないものとなる。つまり、サポートベクター以外の学習データは(128)式で表現される最適化問題の制約条件を自動的に見たし、学習データの展開項の部分には現れない。

(130)式のラグランジュ関数に(131)、(132)式の条件を代入すると双対問題となる以下の最適化問題を得ることができる。

$$\begin{aligned} &\text{目的関数} \quad \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j (z_i{}^T z_j) \to \alpha \text{ について最大化} \\ &\qquad \alpha_i \ge 0, \quad i = 1,...,l \\ &\text{制約条件} \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \end{aligned} \tag{135}$$

最適なα から $\mathbf{w}$ を得るために(132)式の関係を用いる。

(136)式で表される最適化問題を解くために、高次元特徴空間上で次式で表される内積を計算する。

$$(z^T z_i) = (\Phi(x)^T \Phi(x_i)) \tag{136}$$

これを解くためには膨大な計算が必要となるが、Mercer カーネルの条件を満たす、元の観測空間で定義される次式を満たすカーネル関数を用いることで、高次元特徴空間上へ写像するための膨大な計算を削減できる。

$$(\Phi(x)^T \Phi(x_i)) = k(x, x_i) \tag{137}$$

このカーネル関数を用いることで高次元特徴空間での(109)式に相当する識別関数を導出することができ、(109)式の x に $z = \Phi(x)$ を代入すると

$$
\begin{aligned}
f(z) &= \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot z^T z_i + b\right) \\
&= \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot \Phi(x)^T \Phi(x_i) + b\right) \\
&= \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot k(x, x_i) + b\right)
\end{aligned}
\tag{138}
$$

観測空間であるユークリッド空間の内積に代わって、適切なカーネル関数kを選択することで、非線形の場合でも線形 SVM の特性を全て適用することができる。

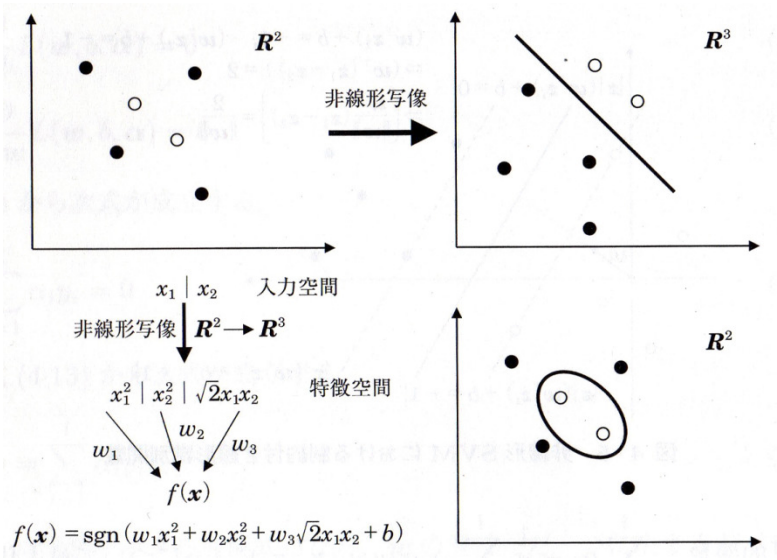


図 3.6.1.3 非線形 SVM の原理[31]

図 3.6.1.3 に非線形 SVM の原理を示す。観測空間(ここでは$\mathbf{R}^2$)上のデータを、非線形写像を用いてより高次元の特徴空間(ここでは$\mathbf{R}^3$)にマッピングし、特徴空間上で分離可能な超平面を作成することで、入力空間では非線形の識別関数になる。

また非線形 SVM では様々なカーネル関数を利用して、多様な学習機械を構成できる。カーネル関数としては

d 次元多項式カーネル　$k(x, x_i) = (x \cdot x_i)^d$ (139)

Radial Basis Function カーネル

$$
k(x, x_i) = \frac{\exp(-\|x - x_i\|)^2}{c} \quad c \text{ はスケールパラメータ} \tag{140}
$$

シグモイドカーネル

$$
k(x, x_i) = \tanh(\kappa \cdot (x \cdot x_i) + \theta) \quad \kappa, \theta \text{ は任意の実数} \tag{141}
$$

などがある。

カーネル関数kは Mercer カーネルの定理を満たす、つまり高次元特徴空間での内積にカーネル関数が一致する必要がある。このことから、$K_{i,j} = \left(y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)_{i,j}$は正値行列となる。

このことは、式(135)で表される最適化問題の目的関数が、凸関数になることを意味する。つまり、非線形 SVM は以下の凸 2 次計画問題

目的関数 $\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \rightarrow \alpha$について最大化

制約条件 $\begin{matrix} \alpha_i \geq 0,\ i = 1, \ldots, l \\ \sum_{i=1}^{l} \alpha_i y_i = 0 \end{matrix}$ (142)

を解き、式(138)で表される識別関数を生成する。式(138)、(142)には高次元特徴空間上での点は現れず、すべてカーネル関数を用いた表現になっていることから、カーネルトリックが利用でき、計算量の大幅な削減が可能となる。

式(132)より、式(142)で表現される最適化問題の制約条件は次のように表現しなおせる。

$\sum_{i=1}^{l} y_i \alpha_i \cdot \boldsymbol{z}^T \boldsymbol{z}_i + b = \sum_{i=1}^{l} y_i \alpha_i \cdot k(\boldsymbol{x}, \boldsymbol{x}_i) + b \geq 1$ (143)

サポートベクターである学習データ$\boldsymbol{x}_j$に対し、上式の統合が成立するので、

$\sum_{i=1}^{l} y_i \alpha_i \cdot k(\boldsymbol{x}, \boldsymbol{x}_i) + b = 1$ (144)

以上より、変数bはα_jを有するサポートベクターに対する次式の平均によって得ることができる。

$1 - \sum_{i=1}^{l} y_i \alpha_i \cdot k(\boldsymbol{x}_j, \boldsymbol{x}_i)$ (145)

3.6.1.4 非線形ソフトマージン SVM

学習データに誤ったデータが含まれる場合、非線形 SVM において適切なカーネル関数を使用しても、学習データを完全に分離できる超平面が存在しない場合がある。そのような場合、線形ソフトマージン SVM 同様に緩和変数を導入して、最適化問題である式(128)の制約条件を満たさない学習データが存在してもよいようにする。この緩和変数を使って、最適化問題である式(128)の制約条件を次式のように緩和できる。

$y_i\left((\boldsymbol{w}^T \boldsymbol{z}_i) + b\right) \geq 1 - \xi_i,\ \ i = 1, \ldots, l$ (146)

これは、ξ_iの値によっては$y_i\left((\boldsymbol{w}^T \boldsymbol{z}_i) + b\right)$の値が、0 に近くても制約条件を満たす場合があることを示している。図 3.6.1.4.1 はこの状態を示したものである。図中の正方形は、学習データの中で誤分類されてもよいデータを意味し、式(146)中のξ_iに値があるデータを表す。図中の円と正方形の色はクラスラベルを意味する。このように、緩和変数を導入することで、式(128)を満たさない学習データが存在してもよいようにするのである。

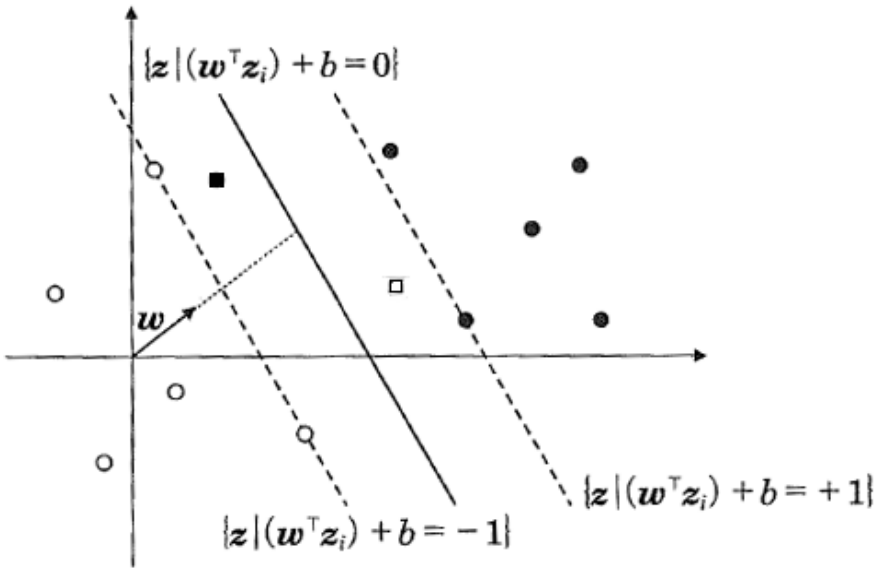


図 3.6.1.4.1 非線形ソフトマージン SVM における制約付き線形識別関数[31]

この緩和変数の導入によって、式(128)で表現される最適化問題は次式のように変形できる。

目的関数 $\tau(\boldsymbol{w},\boldsymbol{\xi}) = \frac{1}{2}\|w\|^2 + \gamma\sum_{i=1}^{l}\boldsymbol{\xi}_i \rightarrow w,\xi$について最小化

制約条件 $y_i\big((\boldsymbol{w}^T\boldsymbol{z}_i) + b\big) \geq 1 - \boldsymbol{\xi}_i,\ \ i = 1,\ldots,l$ (147)

$\sum_{i=1}^{l}\boldsymbol{\xi}_i$は学習データ中で誤分類されるパターンの上限値である。

非線形ハードマージン SVM と同様に、この最適化問題を解くために、ラグランジュ関数を計算する。制約条件である式は、以下のように書き換えることができる。

$1 - \boldsymbol{\xi}_i - y_i\big((\boldsymbol{w}^T z_i) + b\big) \leq 0$ (148)

この制約条件から、制約関数$g_i(\boldsymbol{z}), i = 1,\ldots,l$を$g_i(\boldsymbol{z}) = 1 - \boldsymbol{\xi}_i - y_i\big((\boldsymbol{w}^T\boldsymbol{z}_i) + b\big)$, i = 1, … ,lとすると、ラグランジュ関数は

$\mathrm{L}(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \gamma\sum_{i=1}^{l}\boldsymbol{\xi}_i - \sum_{i=1}^{l}\alpha_i\big(y_i\big((\boldsymbol{w}^T\boldsymbol{z}_i) + b\big) - 1 + \xi_i\big)$ (149)

となる。ここで、$\alpha_i \geq 0$はラグランジュ乗数である。最適化問題を解くには、このラグランジュ関数をα_iについて最大化し、$\boldsymbol{w},b,\boldsymbol{\xi}$について最小化する。

最適化においては、$\boldsymbol{w},b,\boldsymbol{\xi}$についての L の導関数は鞍点において、L の勾配が 0 となるので、次式が成立する。

$$\frac{\partial}{\partial b}L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha}) = 0 \quad (150)$$

$$\frac{\partial}{\partial w}L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha}) = 0 \quad (151)$$

$$\frac{\partial}{\partial \xi} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = 0 \tag{152}$$

これらの 3 つの式からそれぞれ次式が成立する。

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{153}$$

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{z}_i \tag{154}$$

$$\alpha_i = \gamma \tag{155}$$

結局、$\mathbf{w}$は学習データの展開式となる。$\mathbf{w}$の解はただ一つに決まるが、ラグランジュ乗数α_iはその必要がない。

クーン・タッカー条件により、鞍点において、以下の条件が満たされる。

$$\left.\begin{array}{l} \alpha_i \cdot \left[1 - \boldsymbol{\xi}_i - y_i\left((\boldsymbol{w}^T \boldsymbol{z}_i) + b\right)\right] = 0, \ \ i = 1, \ldots, l \\ 1 - \boldsymbol{\xi}_i - y_i\left((\boldsymbol{w}^T \boldsymbol{z}_i) + b\right) \le 0, \ \ i = 1, \ldots, l \\ 0 \le \alpha_i \le \gamma, \ \ i = 1, \ldots, l \end{array}\right\} \tag{156}$$

上記の条件を満たし、$\alpha_i > 0$かつ$\boldsymbol{\xi}_i = 0$を有する学習データ$\boldsymbol{z_i}$をサポートベクターと呼び、$\alpha_i = 0$となる学習データは凸最適化問題の解法には関係のないものとなる。つまり、サポートベクター以外の学習データは$\mathbf{w}$で表される学習データの展開項には現れない。

式(149)のラグランジュ関数に式(153)~(155)の条件を代入すると、双対問題となる以下の最適化問題を得る。

目的関数$\sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \left(\boldsymbol{x}_i \cdot \boldsymbol{x}_j\right) \rightarrow \boldsymbol{\alpha}$について最小化

制約条件$\begin{array}{l} 0 \le \alpha_i \le \gamma, \ \ \mathrm{i} = 1, \ldots, \mathrm{l} \\ \sum_{i=1}^{l} \alpha_i y_i = 0 \end{array}$ (157)

最適な$\boldsymbol{\alpha}$から$\mathbf{w}$を得るには式(154)の関係を用いる。また、bは非線形ハードマージン SVM 同様、α_jを有するサポートベクターに対する次式の平均によって得ることができる。

$$1 - \sum_{i=1}^{l} y_i \alpha_i \cdot k\left(x_j, x_i\right) \tag{158}$$

$$b = -\frac{1}{2}(w^T z_{+1} + w^T z_{-1}) \tag{159}$$

カーネル関数を用いると、高次元特徴空間での式(125)に相当する識別関数を導出することができる。式(125)の$\boldsymbol{x}$に$z = \Phi(x)$を代入して

$$f(z) = \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot \boldsymbol{z}^T \boldsymbol{z}_i + b\right)$$

$$= \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot \boldsymbol{\Phi}(\boldsymbol{x})^T \Phi(\boldsymbol{x}_i) + b\right)$$

$$= \mathrm{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \cdot k(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{160}$$

以上より、式(157)で表現される凸 2 次計画問題を解くことで、式(160)の識別関数を得ることができる。これが非線形ソフトマージン SVM である。

3.6.2 Sequential Minimal Optimization (SMO)

SVM の 2 次計画問題を解くには変数を M 個とすると、$O(M^3)$の時間がかかる。そこで効

率的に得アルゴリズムが必要となるが、SMO(逐次最小最適化アルゴリズム)[32]は 2 つの変数のみを動かす変数として、他の変数は固定した部分問題を繰り返し解いていき、全ての変数が KKT 条件を満たせば終了するという効率的なアルゴリズムを提案している。

3.6.3 Multiple Kernel Learning

3.1 節で述べた Bag-of-Keypoints は一般物体認識で有効な特徴量であるが、カテゴリによっては色やシェイプなど、他の特徴が認識に有効である場合が存在する。例えば、「ライフル銃」や「チンパンジー」などは色の特徴が認識に有効であり、「CD」のように形状が酷似しているカテゴリではシェイプの特徴が認識に有効である。このように、カテゴリによって有効な特徴は異なるため、これらを選択的に利用することで認識精度の改善が可能であることが近年示されている。

そのための特徴統合の手法として、近年用いられているのが Multiple Kernel Learning (MKL) である。この手法では、複数の特徴量のカーネルを重みづけて線形結合することにより統合カーネルを作成し、そのカーネルを SVM に適用することで特徴を選択的に利用することが可能となっている。最適なカーネルの各サブカーネルに対する重みβ_iを学習する問題は MKL 問題と呼ばれ、統合カーネルは以下の式で表すことができる。

$$K_{combined}(x, x') = \sum_{j=1}^{K} \beta_j k_j(x, x') \quad with\ \beta_j \geq 0,\ \sum_{j=1}^{K} \beta_j = 1 \tag{161}$$

この MKL 問題は、全てのβ_jの組み合わせを cross-validation によって解くことも可能であるが、カーネルの数(特徴数と同値)Kが大きくなるにつれ、β_jが取りうる組み合わせは増大し、計算量を抑えるために刻み幅が荒くなることで最適な重みを算出できなくなることが考えられる。そこで、MKL 問題を SVM のフレームワークで解く方法が提案されている。Sonnenburg ら[33]は単一カーネルの SVM 学習の反復により最適なカーネルの重みβ_jを SVM の学習パラメータと同時に求める方法を提案している。その手法を簡単に述べると次のようになる。

1. 最初にβ_jを均一の重みとする。
2. β_jを固定し、統合カーネルを単一カーネルと見なし、通常の SVM 学習を行うことで、サポートベクターの重み$\alpha_i (i = 1, \ldots, N)$とバイアス項$b$を求める。
3. 求めたサポートベクターの重みα_iを固定して、全学習データの識別境界面までの距離が増加するようにβ_jを変化させる。
4. 終了条件に達するまで 2, 3 の手順を繰り返す。

このアルゴリズムは Sonnenburg ら自身によって、機械学習ライブラリである SHOGUN Toolbox[34]として公開されている。

Varma らは、BoK に加えて、テクスチャや色、形状などの多様な画像特徴を画像から抽出して MKL によって統合することで Caltech-101[35]を用いた分類実験での認識精度を、渋滞の単一特徴に比べ 14%ほど向上させ、MKL が非常に有用性の高い手法であることを示

した[5]。

MKL の実装としては前述の Sonnenburg らの SHOGUN Toolbox 以外にも、SimpleMKL[36]や VGG MKL classifer[37]、SpicyMKL[38]などが知られている。

第 4 章 提案手法

4.1 概要

従来の一般物体認識の手法の多くは、画像全体から特徴を記述する手法、あるいは前景領域のみを用いる手法であったが、5 章で述べるが、背景の特徴がカテゴリによっては認識に有効に働くということが確認できた。一方で、前景領域を抽出する物体抽出手法の多くは、抽出のための学習がさらに必要であり、またその学習には正解画像が必要であったため、今後一般物体認識におけるカテゴリ数が増加した際に正解画像の準備が必要となってしまうなどの問題点を抱えていた。

そこで、本研究では学習を伴わない自動物体抽出手法を用い、前景と背景の共起関係を考慮した一般物体認識手法を提案する。図 4.1.1 は提案手法の流れを、図 4.1.2 は提案手法の自動物体抽出の示したものである。具体的には、認識対象となる物体は画像中でも視覚的注意を引きやすいと考えられるため、視覚的注意をモデル化した Saliency Map と高精度な領域分割手法である Graph Cuts を組み合わせ、自動で物体抽出を行う。得られた前景と背景はカテゴリによっては共起関係が変化することが考えられる。そこで、Multiple Kernel Learning と呼ばれる手法を用いて、前景と背景から得られた特徴を重みづけ統合することで、共起関係を考慮した認識を行う。

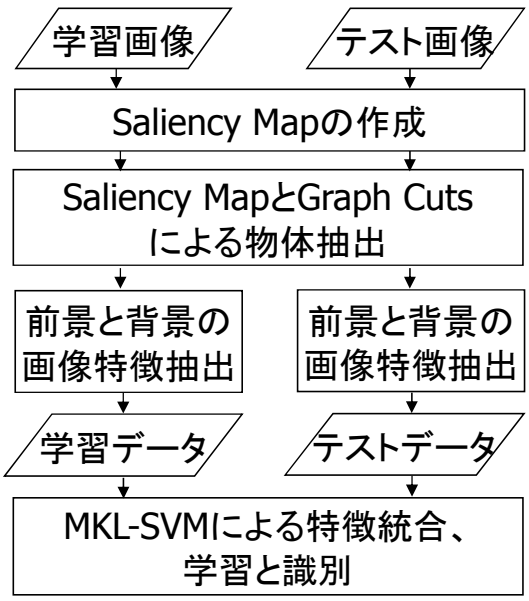


図 4.1.1 提案手法の流れ

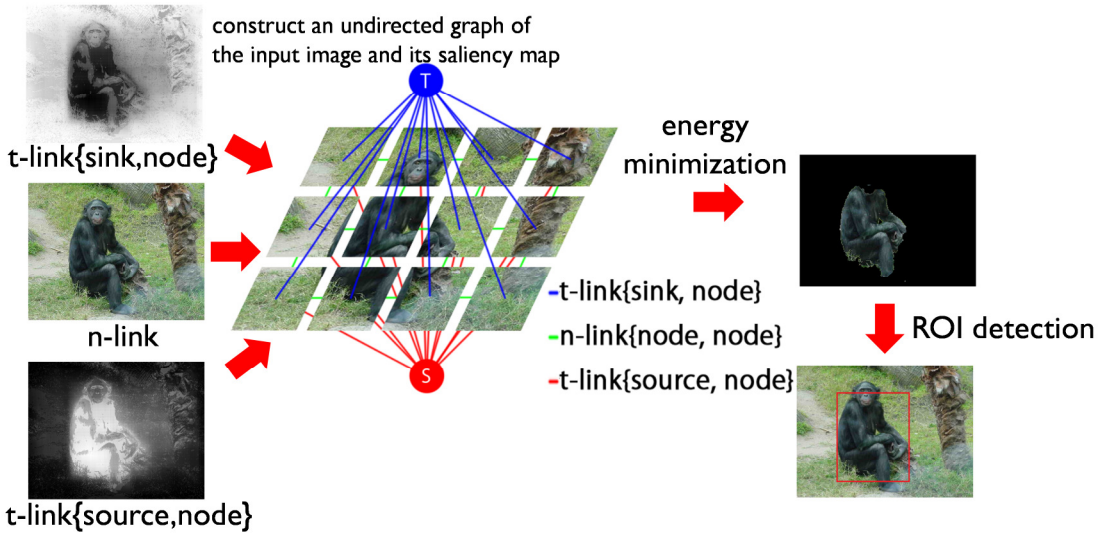


図 4.1.2 自動物体抽出の流れ

4.2 提案手法

前述のとおり、従来の自動物体抽出手法としては、AdaBoost や SVM などの識別器による手法がほとんどであったが、これらの手法は学習が必要であり、カテゴリが増加した場合に処理時間が膨大になり、かつ学習に必要なグランドツルース(正解画像)も増加していくという問題を抱えていた。そこで本研究では、認識対象の物体は視覚的注意を惹きやすいと考え、学習が不要な自動物体抽出手法として Saliency Map を一般物体認識に適用した手法を提案する。本節ではその提案手法の詳細を述べる。

4.2.1 Saliency Map の作成

Saliency Map はいくつものモデルが提案されており[20][21][22][39]、その多くは視覚的注意を惹きやすい「領域」を示したものである。しかし、これらの「領域」を示すモデルはあくまで「領域」を示すものであり、物体全体を抽出する自動物体抽出とは用途が異なるため、これらのモデルによる自動物体抽出は困難であった[40]。そこで、本研究では物体全体を顕著とすることを目的とした Saliency Map である T. Liu らのモデル[22]を用いることで物体全体の抽出を試みる。

Liu らは、局所的特徴である Multiscale Contrast と領域的特徴である Center-surround Histogram、大域的特徴である Color Spatial Distribution を CRF によって色々な画像が入り混じった学習画像群から一度学習し、最適な重みを求めることで、他の画像においても高精度で物体全体を顕著とできるモデルを提案した。その中でも Center-surround Histogram は特に計算量が多く、本研究では、このヒストグラムの距離計算を格子状に行うことで高速化を図っている。

4.2.2 Saliency Map と Graph Cuts による物体抽出

Graph Cuts は高精度な領域分割手法として知られているが、従来の Graph Cuts はユーザが seed と呼ばれる正解ピクセルをいくつかのピクセルに与え、その seed から前景らしさと背景らしさを無向グラフの t-link にコストとして設定する。しかし、自動物体抽出においてはユーザが介入することはできない。そこで、表 4.2.2 に示したようなコストで、4.2.1 項で求めた Saliency Map の値を前景らしさ、背景らしさとして設定する。ここで、$f(p)$は Saliency Map の値を表し、pは対象ピクセル、qはpの隣接ピクセル、Sは無向グラフの Source、Tは無向グラフの Sink であり、Saliency Map の値は[0,1]で正規化している。

表 4.2.2 各エッジのコスト

edge		cost
n-link	{p,q}	$\exp(-\beta d_{p,q})$
t-link	{p,S}	f(p)
	{p,T}	1-f(p)

このようにして得られた無向グラフに対して、Graph Cuts 同様に最小切断アルゴリズムによるエネルギー最小化を行うことで領域分割を行う。さらに、Saliency Map が色などに強く反応し局所的に出た場合には、飛び地や一部欠損した領域が得られることが考えられるため、得られた領域に対してバウンディングボックス(矩形領域)を計算し、その領域を前景領域、それ以外の領域を背景領域として定義することで飛び地や一部欠損を抑制する。この自動物体抽出手法により得られた例を図 4.2.2 に示す。図中の赤い枠内が前景領域、その外側が背景領域である。上段、中段の画像は前景を抽出できているが、下段では大きく枠をとりすぎている、あるいは一部しか抽出できていない場合もいくつか確認できた。また右端の画像はグローブのカテゴリであるが、それよりもボールが目立ってしまっていることにより、適切に抽出できなかったなどの例もあった。

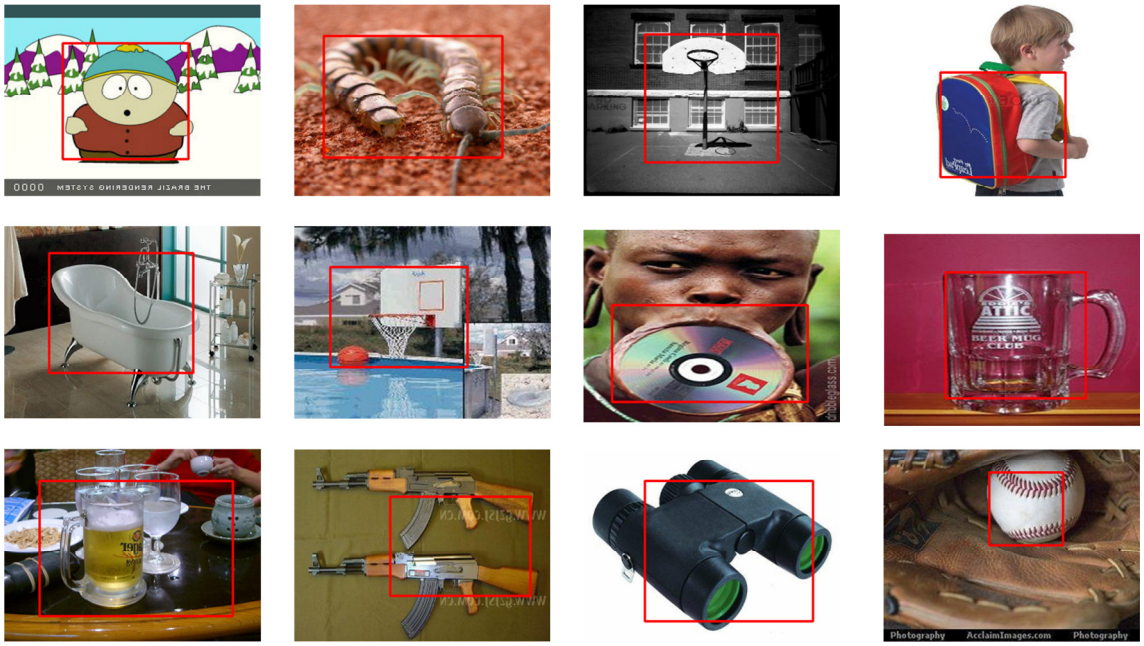


図 4.2.2 自動物体抽出の結果

4.2.3 前景と背景からの画像特徴抽出

得られた前景領域と背景領域のそれぞれから画像特徴を抽出する。この画像特徴としては、今回は SIFT 特徴と色特徴を抽出する。SIFT は 3.1.1 項で述べた Bag-of-Keypoints 表現を用いることで前景と背景のヒストグラム化を行う。色特徴に関しては図 4.2.3 に示すように、前景と背景をそれぞれ空間的に 4 分割し、RGB[0,255]もそれぞれ 0～63,64～127,128～191,192～255 に分割することで、4×4×4×4=256 次元の特徴として抽出する。

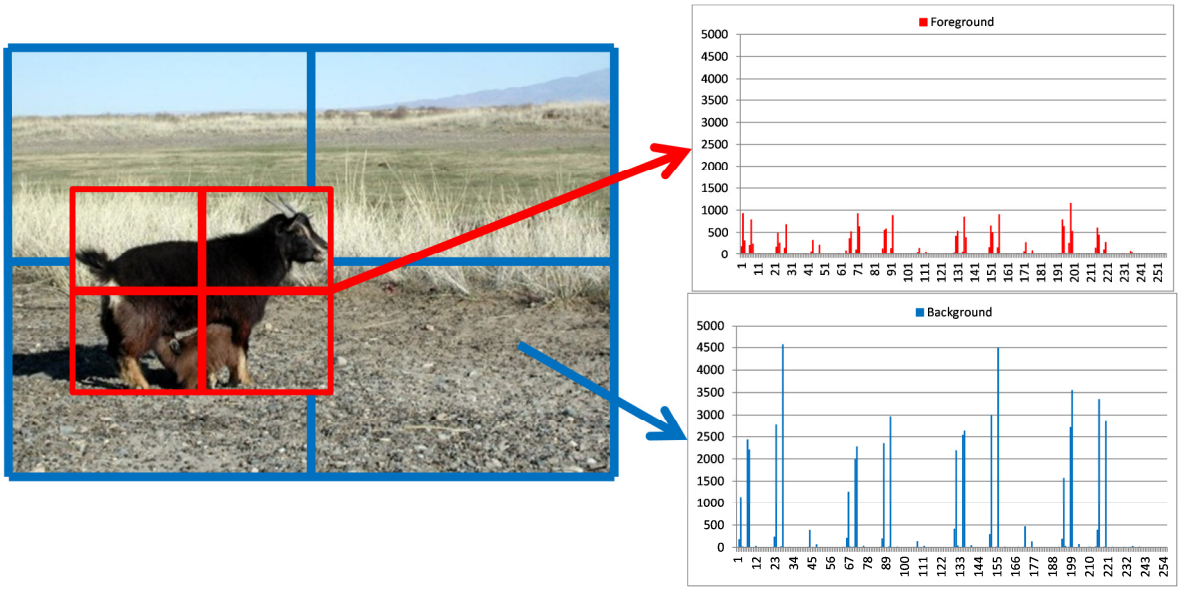


図 4.2.3 色特徴

4.2.4 MKL-SVM による特徴統合と学習・認識

本手法では 3.6.3 項で述べた Varma らの手法と同様に得られた前景と背景の特徴間の最適な重みを MKL-SVM によって求めて統合し、認識を行う。図 4.2.4 はカテゴリ間で前景と背景の重みが異なる様子を示したものである。左はバスケットゴールのカテゴリ、右はバスタブのカテゴリである。バスケットゴールは屋外に設置されている場合や屋内に設置されている場合、またボードが写っているか否か、などの差があり、背景の重みが小さくなっている。一方、バスタブは設置場所の多くが浴室であり、背景のタイルなどから類似した特徴が得られるため、背景の重みが大きくなっている。このようにカテゴリごとに各特徴の最適な重みを求めることによって、前景と背景の共起関係を考慮した認識を行う。

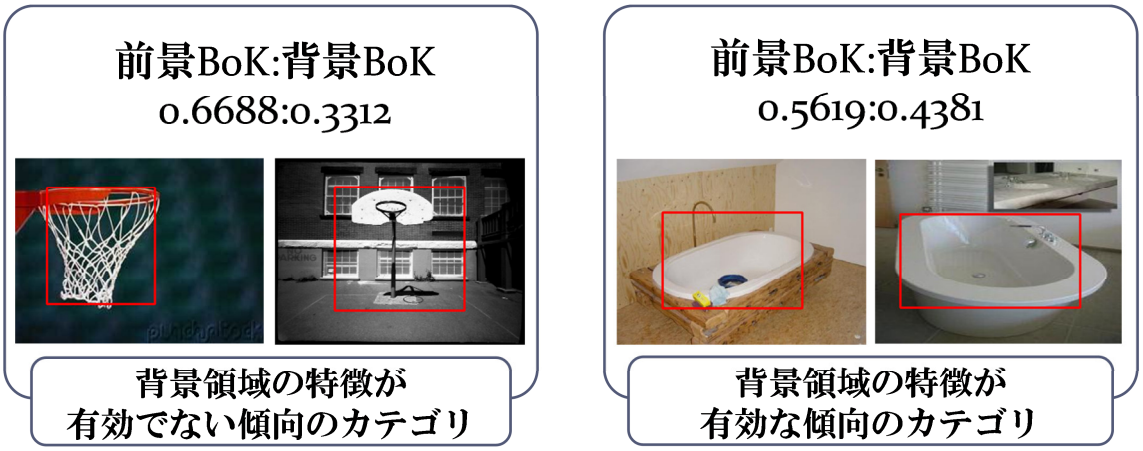


図 4.2.4 MKL-SVM による各カテゴリの特徴間の重み例

第 5 章 実験

5.1 事前実験

提案手法の評価実験を行う前に、物体抽出が実際に一般物体認識に有効であることを確認するための事前実験を行う。事前実験では、物体抽出を行わず、画像全体から特徴を抽出、認識を行う手法(Conventional)と、事前に手作業でグランドツルース([41]からダウンロードが可能)を用いて前景と背景の特徴を分離して抽出し、それらを統合して認識を行う手法(Groundtruth)を比較することで、一般物体認識における物体抽出の有効性を検証する。

ここでは、次の 5 つの項目について評価を行う。

(1) Conventional1：SIFT(BoK)を SVM で認識

(2) Conventional2：SIFT(BoK)+色特徴を MKL-SVM で統合して認識

(3) Groundtruth0：Groundtruth 画像の前景 SIFT(BoK)を SVM で認識

(4) Groundtruth1：Groundtruth 画像の前景 SIFT(BoK)、背景 SIFT(BoK)を MKL-SVM で統合して認識

(5) Groundtruth2：Groundtruth 画像の前景 SIFT(BoK)、背景 SIFT(BoK)、前景色特徴、背景色特徴を MKL-SVM で統合して認識

事前実験の実験環境としては、Caltech-256[42]からランダムに選んだ表 5.1.1 に示した 10 カテゴリを使用する。認識に用いる画像特徴量としてはアピアランスを捉えるために BoK で表現した SIFT 特徴、色特徴の 2 つの特徴を用い、BoK 表現で用いられる Visual Word 数については 100 から 1000 まで 100 刻みで計測し、もっとも精度の高いものをその特徴の Visual Word 数とする。また、各カテゴリの学習画像数は 50 枚とし、学習画像とは別に選出された 40 枚をテスト画像とする。マルチクラス SVM の判定には 1-vs-rest を用いる。性能評価のための評価指標には次式に示す Accuracy を用いる。

$$Accuracy = \frac{\text{正解画像数}}{\text{全画像数}}$$

実験結果を表 5.1.2 と図 5.1 に示す。

表 5.1.1 実験で使用する 10 カテゴリ

ID	Category Name
1	ak47
2	backpack
3	baseball-glove
4	basketball-hoop
5	bathtub
6	beer-mug
7	binoculars
8	cartman
9	cd
10	centipede

表 5.1.2 手法ごとの認識率

Method	Conventional1	Conventional2	Groundtruth0	Groundtruth1	Groundtruth2
Accuracy	0.6475	0.6875	0.705	0.7475	0.7975

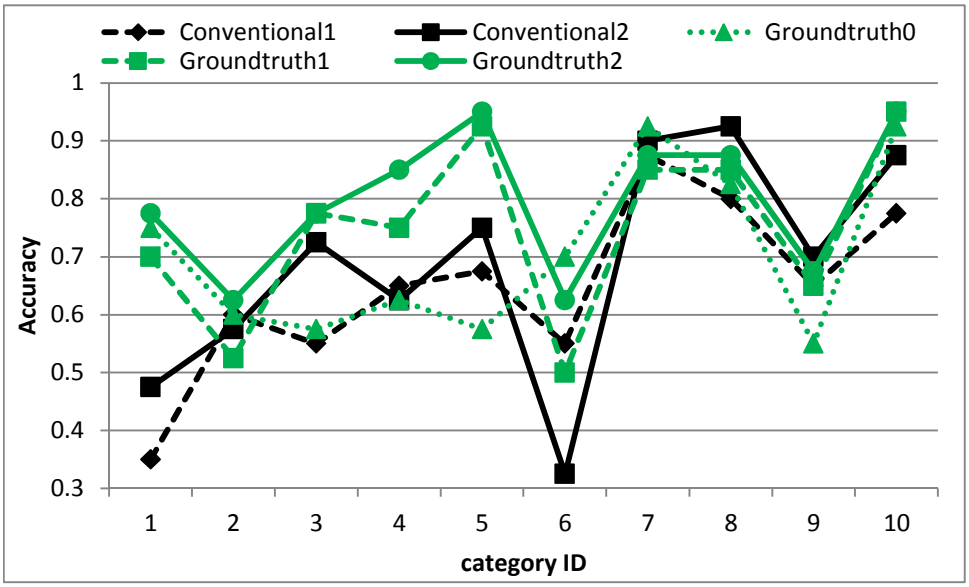


図 5.1 カテゴリごとの認識率

まず、表 5.1.2 の Conventional1 と Groundtruth0 の比較により、画像全体の特徴を用いるよりも前景のみを特徴を用いたほうが有効な傾向にあることがわかる。しかし、図 5.1 のカテゴリごとの認識率をみると、前景を抽出することによって 5 番のカテゴリのように認

識率が低下しているカテゴリが存在する。5 番のカテゴリはバスタブのカテゴリであるが、このカテゴリは4.2.4項で述べたとおりほとんどの背景が浴室のタイルなどであることから特徴が類似しており、背景の情報が認識に有効であったと考えられる。

次に Conventional1 と Groundtruth1 を比較すると、認識率は 10%向上しており、カテゴリごとの認識率を見ても、多くのカテゴリで改善されていることが確認できる。5 番のカテゴリに関しては背景の特徴を統合することで 30%以上改善できていることがわかる。同様に Conventional2 と Groundtruth2 を比較しても認識率は 11%改善されており、多くのカテゴリで認識率が改善できていることがわかる。

以上より、物体抽出を行い、前景と背景を統合することが一般物体認識の性能改善に有効であることがわかる。

5.2 評価実験

提案手法の有効性評価のための評価実験を行う。具体的には以下の 6 手法の比較を行う。

(1) Conventional1：SIFT(BoK)を SVM で認識

(2) Conventional2：SIFT(BoK)+色特徴を MKL-SVM で統合して認識

(3) Proposal1：自動物体抽出後、前景 SIFT(BoK)と背景 SIFT(BoK)を MKL-SVM で統合して認識

(4) Proposal2：自動物体抽出後、前景 SIFT(BoK)、背景 SIFT(BoK)、前景色特徴、背景色特徴を MKL-SVM で統合して認識

(5) Groundtruth1：Groundtruth 画像の前景 SIFT(BoK)、背景 SIFT(BoK)を MKL-SVM で統合して認識

(6) Groundtruth2：Groundtruth 画像の前景 SIFT(BoK)、背景 SIFT(BoK)、前景色特徴、背景色特徴を MKL-SVM で統合して認識

実験に使用する評価用画像データセットには事前実験同様に、表 5.1.1 に示した Caltech-256 からランダムに選んだ 10 カテゴリを用いる。また各カテゴリからランダムに選んだ 50 枚を学習画像とし、その他の画像からランダムに選んだ 40 枚をテスト画像として実験を行う。評価方法は、次式のようにクラスごとの識別率を平均した Accuracy を用いる。

$$Accuracy = \frac{\text{正解画像数}}{\text{全画像数}}$$

また、Visual Word の数は 100 から 1000 までの間で 100 刻みに変化させて、最も認識率の高いものをその手法の Visual Word 数とする。

実験結果を表 5.2.1 と図 5.2.1 に示す。

表 5.2.1 手法ごとの認識率

Method	Conventional1	Conventional2	Proposal1	Proposal2	Groundtruth1	Groundtruth2
Accuracy	0.6475	0.6875	0.7275	0.76	0.7475	0.7975

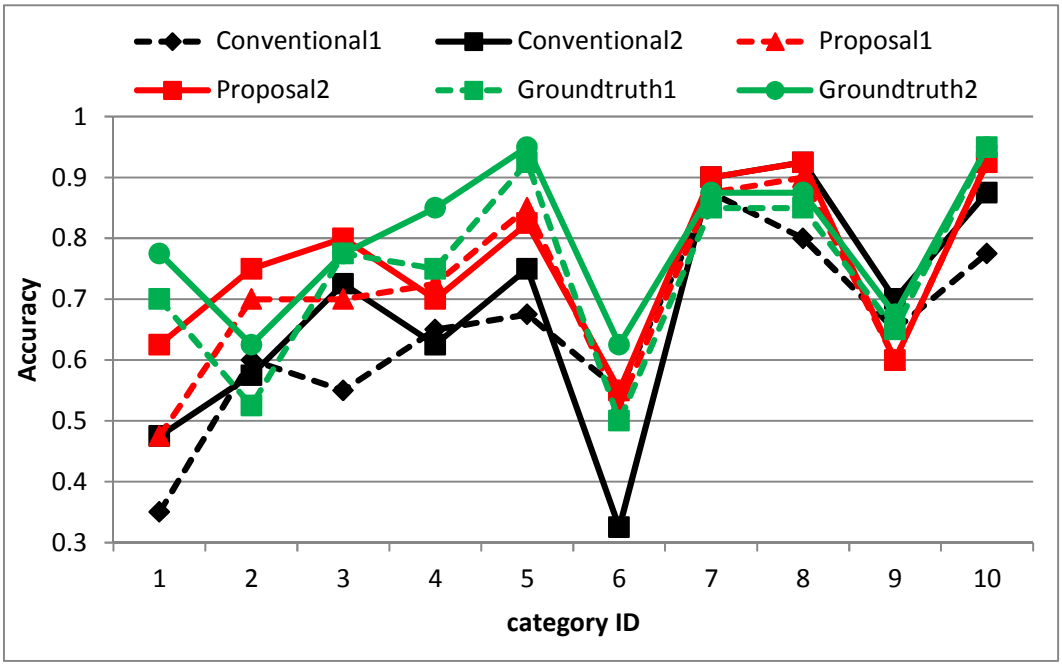


図 5.2 カテゴリごとの認識率

まず表 5.2 を見ると、Conventional1 と Proposal1 を比較すると平均認識率が 8%向上していることがわかる。図 5.2.1 を見るといくつかのカテゴリで精度が落ちているものの、多くのカテゴリで精度が向上している。精度が落ちている原因としては自動物体抽出がうまく行えなかったことが考えられる。

次に Conventional2 と Proposal2 を比較すると平均認識率は 7.25%向上しており、カテゴリごとの認識率を見ても多くのカテゴリで精度が向上している。1～3 番のカテゴリなどでは Conventional1、Proposal1 に比べて精度が向上していることから色特徴が有効に働いていると考えられる。

一方、Proposal1、Proposal2 と Groundtruth1、Groundtruth2 を比較すると Groundtruth を用いる手法が 2～4%程度精度が高いことから、自動物体抽出の精度を改善することで、さらに数%の精度改善が見込めることがわかる。図 5.2.2 は現状で抽出が失敗している画像の例である。上段の画像は元画像の色の影響により一部のみが顕著となり、抽出に失敗している。中段の画像は物体が大きく、Liu らの Saliency Map の Center-surround Histogram の特性上、全体を顕著とできなかったため失敗している。この上段と中段の例は、Saliency Map の特性を改善することで抽出精度向上が期待できる。しかし、下段はバスケットゴールのカテゴリであるにも関わらず、バスケットボールの方が目立っていることから抽出に

失敗した。しかし、このような複数の物体、もしくは認識対象の物体以外が目立っている場合には Saliency Map を用いた抽出は困難であり、画像中に含まれる物体をあらかじめそれぞれ認識した上で分類する必要があると考えられる。

また、6 番や 9 番のカテゴリなど、物体抽出の有無にかかわらず、精度が低いカテゴリが確認できる。これらのカテゴリでは形状(シェイプ)が酷似しており、アピアランスを表現する SIFT や色特徴よりも、シェイプを表現する HOG や Self-Similarity などの画像特徴を用いると精度が改善する可能性があると考えられる。図 5.2.3 は 9 番の CD のカテゴリの物体抽出結果を示したものであり、表 5.2.2 は提案手法の各カテゴリでの各特徴の重みを示したものである。CD のカテゴリでは物体(CD)が画像に大きく写っており、Saliency Map の特性上、全体をうまく抽出できない画像が多く、CD の縁の部分の大部分は背景領域に出てしまっている。表 5.2.2 の CD のカテゴリにおける各特徴の重みを見ると、背景領域の SIFT 特徴が大きな比重を占めていることがわかるが、これは CD の縁の部分を記述した SIFT 特徴によるものと考えられる。このことからも CD のように形状が重要な情報であるカテゴリでは、シェイプを表現する特徴を認識に用いることが重要と考えられる。

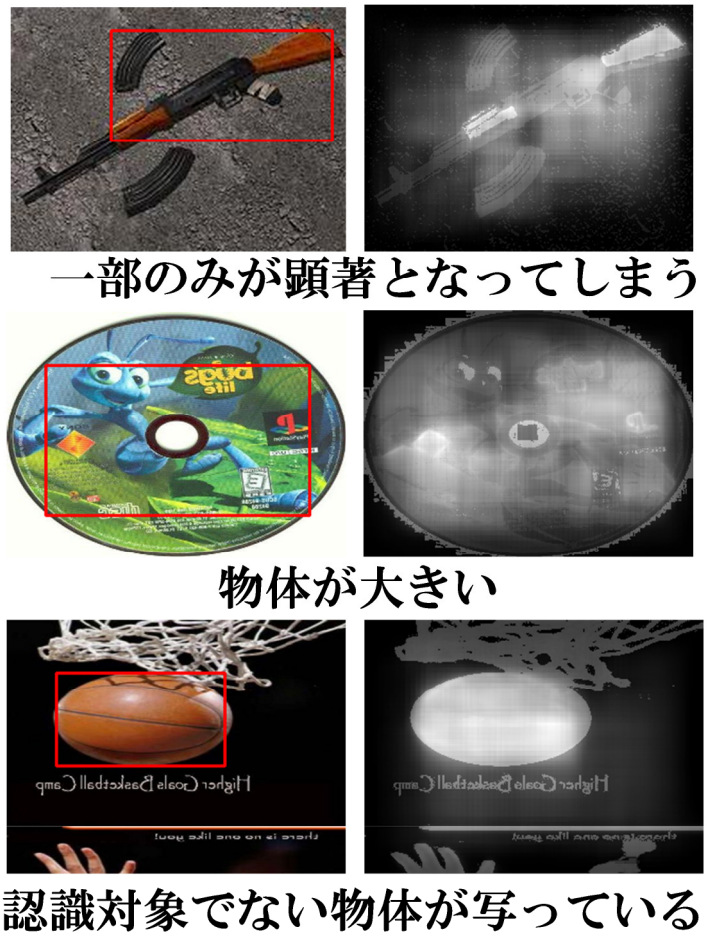


図 5.2.2 自動物体抽出に失敗している画像とその理由

図 5.2.3 CD のカテゴリの画像に対する提案手法による抽出結果

表 5.2.2 CD のカテゴリにおける提案手法の各サブカーネルの重み

category ID	SIFT(fore)	SIFT(BoK)	Color(fore)	Color(back)
1	0.68598	0.16935	0.11932	0.02535
2	0.42505	0.31259	0.23	0.03236
3	0.45809	0.36213	0.1785	0.00129
4	0.62289	0.09036	0.17694	0.10981
5	0.55037	0.19253	0.18442	0.07269
6	0.44524	0.32832	0.15382	0.07262
7	0.45926	0.31586	0.22469	0.00019
8	0.50786	0.07246	0.40685	0.01282
9	0.21444	0.69336	0.08463	0.00757
10	0.69433	0.21281	0.09264	0.00023

第 6 章 むすび

6.1 まとめ

本論文では、Saliency Map と Graph Cuts を用いた自動物体抽出によって抽出した前景と背景の特徴をカテゴリごとに重みづけて統合・認識を行うことで、画像全体の特徴を用いる場合や前景の特徴のみを用いる場合よりも、高い精度で一般物体認識を行うことが可能であることを示した。また、自動物体抽出が理想的に行えた場合にはさらに 2%程度の精度改善が行えることがわかった。

6.2 今後の課題

今後の課題としては、物体抽出の精度を改善するために Saliency Map の特性改善が必要だと考えられる。また、アピアランスを捉える SIFT 特徴だけでなく、シェイプを捉える Self-Similarity や HOG などの特徴を組み合わせることでさらに認識率を向上させられるか検討を行う必要がある。

参考文献

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual Categorization with Bags of Keypoints,” ECCV Workshop on Statistical Learning in Computer Vision, Prague, Czech, pp. 1-22, 2004.

[2] D. G. Lowe, “Distinctive Image Features from Scale- Invariant Keypoints,” International Journal of Computer Vision, 2004.

[3] Dalal N., Triggs B.: “Histograms of Oriented Gradients for Human Detection.” IEEE International Conference on Computer Vision and Pattern Recognition, vol.1, pp.886-893, 2005.

[4] E. Schechtman, M. Irani, “Matching Local Self-Similarities across Images and Videos,” IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA, 2007.

[5] M. Varma, and D. Ray, “Learning The Discriminative Power-Invariance Trade-Off,” IEEE International Conference on In Computer Vision, Rio de Janeiro, Brazil, October 2007.

[6] 柳井啓司, “一般物体認識の現状と今後,” Technical report of IEICE, PRMU 106(229), 2006.

[7] L. Fei-Fei, P. Pietro, “A Bayesian Hierarchical Model for Learning Natural Scene Categories,” IEEE International Conference on Computer Vision and Pattern Recognition, pp.524-531, 2005.

[8] S. Lazebnik, C. Schmid, J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” IEEE Conference on Computer Vision and Pattern Recognition, New York, USA, June 2006.

[9] M. Marszałek, C. Schmid, “Spatial Weighting for Bag-of-Features,” IEEE Conference on Computer Vision and Pattern Recognition, New York, USA, pp. 2118-2125, June 2006.

[10] 藤吉弘亘, “Gradient ベースの特徴抽出 -SIFT と HOG－”, Technical Report of IEICE, PRMU, 107(206), pp.211-224, 2007.

[11] Y. Ke, R. Sukthankar, “PCA-SIFT: A More Distinctive Representation for Local Image Descriptors,” IEEE International Conference on Computer Vision and Pattern Recognition, 2004.

[12] A. E. Abdel-Hakim, A. A. Farag, “CSIFT: A SIFT Descriptor with Color Invariant Characteristics,” IEEE International Conference on Computer Vision and Pattern

Recognition, 2006.
[13] A. Stein, M. Hebert, “Incorporating background invariance into feature-based object recognition,” IEEE Workshop on Applicaions of Computer Vision and Pattern Recognition, pp. 1978-1983, 2006.
[14] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, “SURF: Speeded Up Robust Features,” Computer Vision and Image Understanding, Vol.110, No.3, pp. 346-359, 2008.
[15] P. A. Viola, M. J. Jones, “Rapid object detection using a boosted cascade of simple features,” International Conference on Computer Vision and Pattern Recognition, 2001.
[16] A. Bosch, A. Zisserman, X. Munoz, “Representing shape with a spatial pyramid kernel,” ACM International Conference on Image and Video Retrieval, 2006.
[17] A. Vedaldi, M. Varma, V. Gulshan, A. Zisserman, “Multiple Kernels for Object Detection,” IEEE International Conference on In Computer Vision, 2009.
[18] http://ja.wikipedia.org/wiki/HSV 色空間
[19] http://www.konicaminolta.jp/instruments/knowledge/color/part1/07.html
[20] L. Itti, C. Koch, E. Niebur, “A Model of Saliency-Based Scene Analysis,” IEEE Pattern Analysis and Machine Intelligence, 1998.
[21] F. Stentiford, “A Visual Attention Estimator Applied to Image Subject Enhancement and Colour and Grey Level Compression”, International Conference on Pattern Recognition, 2004.
[22] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, “Learning to Detect A Salient Object,” IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA, 2007.
[23] M. Kass, A. Witkin, D. Terzopoulos, “Snakes: Active contour models,” International Journal of Computer Vision, Vol. 1, No. 4, pp.321-331, 1988.
[24] S. Osher, J. A. Sethian, “Fronts propagating with curvature dependent speed: Algorithm based on Hamilton-Javobi formation,” Journal of Computational Physics, Vol. 79, pp.12-49, 1988.
[25] J. Sethian, “Level Set Methods, 1st ed.” Cambridge University Press, New York, 1996.
[26] Y. Y. Boykov, and M. P. Jolly, “Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images,” International Conference on Computer Vision, Vancouver, Canada, Vol.I, pp. 105-112, July 2001.
[27] 八木康史,斎藤英雄, “コンピュータビジョン最先端ガイド 1”, アドコムメディア.
[28] 石川博, “グラフカット”, 2007-CVIM-158.
[29] Y. Y. Boykov, et al., “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” Workshop on Energy Minimization Methods in

CVPR, 2001.
[30] C. Rother, V. Kolmogorov, A. Blake, " "GrabCut" – Interactive Foreground Extraction using Iterated Graph Cuts", ACM SIGGRAPH 2004.
[31] 小野田崇, "知の科学　サポートベクターマシン", オーム社.
[32] J. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Sch¨olkopf B., Burges C.J.C., and Smola A.J. (Eds.) Advances in Kernel Methods—Support VectorLearning, pp. 185–208, Cambridge, MA, MIT Press.
[33] S. Sonnenburg, G. Rätsch, C. Schäfer, B. Scholköpf, "Large Scale Multiple Kernel Learning," The Journal of Machine Learning Research, Vol. 7, 2006.
[34] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J.Behr, A. Zien, F. d. Bona, A. Binder, C. Gehl, V. Franc, "The SHOGUN Machine Learning Toolbox," Journal of Machine Learning Research 11, pp.1799-1802, 2010.
[35] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[36] http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html
[37] http://www.robots.ox.ac.uk/~vgg/software/MKL/
[38] http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/SpicyMKL/
[39] R. Achanta and S. Susstrunk, "SALIENCY DETECTION USING MAXIMUM SYMMETRIC SURROUND", IEEE International Conference on Image Processing, 2010.
[40] 藤川哲也, 佐藤元昭, 甲藤二郎, "Graph Cuts を用いた Bag-of-Keypoints 法の特性改善に関する検討," 電子情報通信学会画像工学研究会, IE2008, pp. 35-40, 2010.
[41] http://www.katto.comm.waseda.ac.jp/~fujikawa/
[42] G. Griffin, A. Holub, and P. Perona, "Caltech 256 object category dataset," Technical Report UCB/CSD-04-1366, California Institute of Technology, 2007.

謝辞

本論文を作成するにあたり、終始適切なご指導を頂きました甲藤二郎教授に心よりお礼申し上げます。

また、忙しい中さまざまなご指導、助言を頂きました甲藤研究室の先輩方を始め、さまざまなバックアップをして下さった同期に深く感謝いたします。

2012 年 1 月 31 日

藤川哲也

発表文献

1. 藤川哲也, 佐藤元昭, 甲藤二郎, “Graph Cuts を用いた Bag-of-Keypoints 法の特性改善に関する検討,” 電子情報通信学会 総合大会, March 2010.
2. 藤川哲也, 佐藤元昭, 甲藤二郎, “Graph Cuts を用いた Bag-of-Keypoints 法の特性改善に関する検討,” 電子情報通信学会　画像工学研究会, March 2010.
3. 藤川哲也, 甲藤二郎, “物体領域抽出による画像の識別率向上に関する検討,” 電子情報通信学会 画像工学研究会, September 2010.
4. Tetsuya FUJIKAWA, and Jiro KATTO, “A STUDY ON PERFORMANCE IMPROVEMENT OF GENERIC OBJECT RECOGNITION,” Workshop on Picture Coding and Image Processing, December 2010.
5. 藤川哲也, 佐藤元昭, 甲藤二郎, “Graph Cuts を用いた Bag-of-Keypoints 法の特性改善に関する検討,” 日刊工業出版 画像ラボ 1 月号, January 2011.
6. 藤川哲也, 甲藤二郎, “一般物体認識の識別率向上に関する検討,” 電子情報通信学会 総合大会, March 2011.
7. 藤川哲也, 甲藤二郎, “物体抽出を用いた一般物体認識の精度改善の検討,” Image Media Processing Symposium 2011, October 2011.
8. 山崎智章, 藤川哲也, 甲藤二郎, “Bilateral Filter を用いた SIFT による一般物体認識,” Image Media Processing Symposium 2011, October 2011.
9. Tomoaki Yamazaki, Tetsuya Fujikawa, Jiro Katto, “IMPROVING THE PERFORMANCE OF SIFT USING BILATERAL FILTER AND ITS APPLICATION TO GENERIC OBJECT RECOGNITION,” IEEE International Conference on Acoustics, Speech, and Signal Processing 2012, March 2012.
10. 藤川哲也, 甲藤二郎, “Saliency Map を用いた自動物体抽出による一般物体認識の精度改善,” 電子情報通信学会 画像符号化・映像メディア処理 レター特集号, September 2012.(submitted)
11. 山崎智章, 藤川哲也, 甲藤二郎, “Bilateral Filter を用いた SIFT の性能改善と一般物体認識への応用,” 電子情報通信学会 画像符号化・映像メディア処理 レター特集号, September 2012.(submitted)