


Summary of Bachelor's Thesis
 2014年3月修了卒業論文概要書

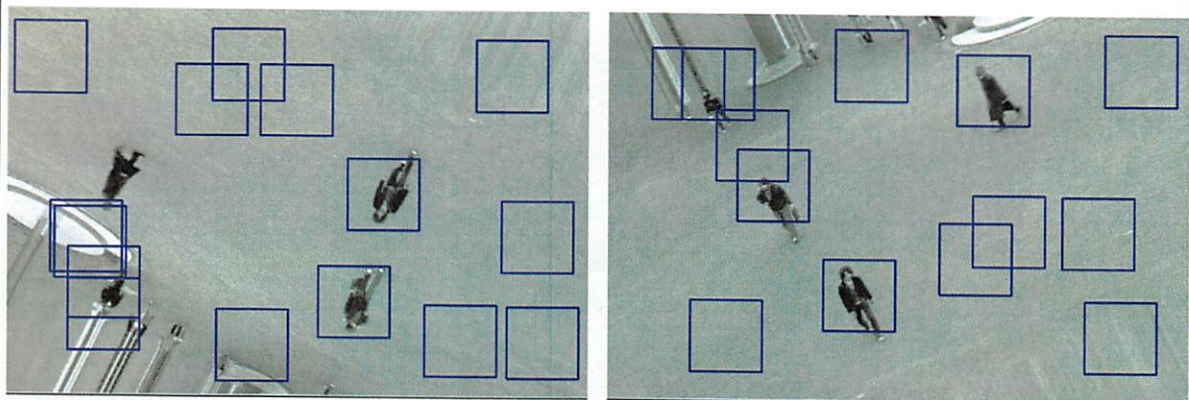
| | | | |
|---------------------------------------|------|---|------------|
| Name 氏名 | 石上 諒 | ID number 学籍番号 | 1W100033-6 |
| 題目 Title (日本語の場合は英文題目も記入) | | Supervisor 指導教員 | 渡辺 裕 |
| 日本語 Japanese | |  | |
| 回転に対して頑健なHOG特徴量の提案 | | | |
| 英語 English | | | |
| A Proposal of Robust HOG for Rotation | | | |

概要 Summary

Histograms of Oriented Gradient(HOG)は、人の顔認識を始めとした一般物体認識の分野で広く用いられている特徴量である。しかし、従来使われているHOGは回転に対して頑健な特徴量ではない。そこで、本論文では回転に対して頑健なHOG特徴量を提案する。

本論文ではまず、従来のHOGの特徴とアルゴリズムについて説明し、従来のHOGが回転に対して頑健な特徴量ではないことを述べる。提案手法は認識対象となる物体画像の主要方向の算出と画像を回転させて認識処理を行うことを特徴とする。従来のHOGはヒストグラムで量子化する角度よりも小さい回転に対応できない。提案手法はこの、より細かい回転への対応を目指す。主要方向は、注目領域内で最も勾配の強い方向と定義し、領域内の全ピクセルの勾配方向の投票によって決定する。主要方向の算出にはガウス関数を使用する。1ピクセルからは1方向分の勾配強度しか得られないため、ガウス関数を使用することによって複数方向に投票することができる。

実験では中部大学藤吉研究室が提供しているデータセット及びHOGのソースコードを使用した。従来のHOGと提案手法における検出精度を同一条件で比較した。その結果、提案手法の回転対象に対する精度が従来のHOGより優れていることがわかった。



提案手法による検出結果例

- 注意 1. 提出期限/ Deadline of Submission 2014年2月6日(木)17:00 /February 6, 2014 17:00
 2. 原本1部、コピー1部の計2部提出 / Must submit one original and one copy

基幹理工学部情報理工学科

Bachelor's Thesis
卒業論文

Title

論文題目

回転に対して頑健なHOG特徴量の提案

A Proposal of Robust HOG for Rotation

| | |
|--------------------|----------------------|
| Student ID 学籍番号 | 1W100033-6 |
| Name 氏名 | Ryo Ishigami 石上 諒 |

| | | |
|--------------------|--------------------------|---|
| Supervisor 指導教員 | Hiroshi Watanabe 渡辺 裕 | 印 |
|--------------------|--------------------------|---|

2014年3月15日

目次

| | |
|---|-----------|
| 第 1 章 序論 | 1 |
| 1.1 研究の背景..... | 1 |
| 1.2 本研究の目的..... | 3 |
| 1.3 本論文の構成..... | 3 |
| 第 2 章 従来の HOG のアルゴリズム | 4 |
| 2.1 まえがき..... | 4 |
| 2.2 HOG のアルゴリズム..... | 4 |
| 第 3 章 提案手法 | 7 |
| 3.1 まえがき..... | 7 |
| 3.2 主要方向算出..... | 7 |
| 3.3 画像の回転..... | 9 |
| 3.4 処理の流れ..... | 11 |
| 第 4 章 従来の HOG と提案手法における検出精度の比較実験 | 12 |
| 4.1 まえがき..... | 12 |
| 4.2 評価指標..... | 12 |
| 4.3 検出精度に関する比較実験..... | 13 |
| 4.3.1 実験条件..... | 13 |
| 4.3.2 実験結果の比較..... | 15 |
| 4.3.3 提案手法の結果に対する考察..... | 17 |
| 第 5 章 結論 | 21 |
| 5.1 総括..... | 21 |
| 5.2 課題..... | 22 |
| 謝辞..... | 23 |
| 参考文献..... | 24 |
| 付録 A 実験で使したソースコード..... | 26 |
| 図一覧..... | 31 |
| 表一覧..... | 32 |

第1章

序論

1.1 研究の背景

物体認識とは、画像の中に何が存在するのかをコンピュータに判別させるための技術である。人間は、視界に入ったものが何であるのかをほとんど無意識に識別することができる。しかしコンピュータの場合は、これと同様の処理を行うために、画像内から「特徴量」と呼ばれる「物体を認識するために有用であると考えられる情報」を抽出しなければならない。この特徴量は数値として人間がコンピュータに与えなければならない。与えられた数値は学習器に渡される。学習器は、与えられた数値群をクラス毎に分ける役割を担う。物体認識では、**Support vector machine(SVM)**、**Real-AdaBoost** 等が主に利用されている[1]。

この、特徴量と学習器を利用して、物体が属するクラスを判別させるのが物体認識の処理手順である。

物体認識の技術は、近年さまざまな分野で利用されるようになってきた。物体認識は計算コストが大きいため、コンピュータの性能の向上によって実用的なものとなり始めた。中でも、人や人の顔といった一般物体の認識を目指す技術は、近年大きな進化を遂げてきた。防犯カメラ、監視カメラ映像からの不審者の発見、ITS（高度道路交通システム）の一端として運転支援を行う等、セキュリティから運転支援システムに至るまで、広い範囲で一般物体認識の技術が期待を集めている[2][3]。現在、人の顔認識を利用してオートフォーカスをおこなうカメラ、スマートフォン等が製品化され、物体認識の技術は比較的生活になじみのある分野となりつつある[4][5]。

第1章 序論

この一般物体認識において利用される特徴量の一つが **Histograms of Orientation Gradients (HOG)** である。HOG は、人の認識にある程度有用であることが先行研究によって示されている。照明変化にもある程度頑健なこの特徴量は、定点カメラにおける人認識から船舶画像認識、漫画のキャラクターの顔認識まで、その判別効果について様々な対象で精度が検証なされている[6][7]。

HOG の特徴の一つに「物体の形状、輪郭を大まかに表現できること」が挙げられる。物体認識には一般物体認識ともう一つ、特定物体認識と呼ばれる分野が存在する。一般物体認識は「人」や「車」など「一般的な名称」を判別する技術であり、特定物体認識はさらに「同一人物であるか？」を判別する技術である。一般物体認識の性質から考えると、「大まかな輪郭情報」に頼る HOG が一定の評価を得ていることも納得できる。一方、特定物体認識には **Scale-Invariant Feature Transform(SIFT)** と呼ばれる特徴量が使われる。SIFT は画像の中から、「キーポイント」となる点を算出し、画像間でのそのキーポイントの一致によって同一物体かどうか判断するものである。SIFT はたとえ、画像間で対象の物体の角度が変わっていても（回転）同一物体を検出することができる[8]。回転に対してロバストな特徴量である。

しかし HOG は、回転に対してはロバストな特徴量でない。HOG をベースに回転不変の特徴量を提案する研究もあるが、良好な検出結果は得られていない。

従来の HOG 特徴量をそのまま回転に対して強い特徴量にするためには

- (1) 従来の学習用画像に、さまざまな角度に回転した物体画像を追加する
- (2) 検出用画像を様々な角度に回転させた画像を作成し、走査していく画像に追加する

等も対策に挙げられる。しかし

(1) に関しては、学習量が膨大になることに加えて、過学習、誤検出の増加が懸念される。角度が異なった画像から特徴を抽出するため特徴量が安定せず、輪郭の再現度が低くなる。

(2) に関しては、認識対象として1枚のみであった画像が 45° 刻みで回転に対応しようとするとも8枚まで増加する。細かく角度に対応しようとするほど、検出の処理時間が遅くなる。ある程度までは、回転に対応できるが、回転にロバストな特徴量とは言い難い。

このような背景から、HOG をベースにした回転にロバストな特徴量の獲得を本研究の目的とする。

第1章 序論

1.2 本研究の目的

HOG をベースにした回転不変な特徴量は、現段階では実現されていない。HOG をベースにした特徴量を作成し、その特徴量が回転不変であることを確認することが本研究の目的となる。そのために従来の HOG と提案手法とで、回転した物体を学習画像に使用し、検出用画像もまた回転している場合に対しての精度を比較する。

1.3 本論文の構成

以下に本章以降の構成を示す。

第1章 本章であり研究の背景、及びその目的について述べている。

第2章 従来の HOG のアルゴリズムがどのように特徴量を算出しているかを示す。また、提案手法においてもこのアルゴリズムを基本として使用する。

第3章 主要方向算出、画像回転の順に、提案手法に関するアルゴリズムを示す。ここで示すアルゴリズムと従来の HOG の算出を併用することで、回転にロバストな特徴量の実現を目指す。

第4章 従来の HOG と提案手法との比較実験を行う。この実験により回転している検出対象に対する精度において、提案手法が優位であることを示す。

第5章 本論文の総括と今後の課題について述べる。

第2章

従来の HOG のアルゴリズム

2.1 まえがき

HOG は輝度（色、明るさ）の勾配強度、勾配方向に着目した特徴量である。物体のシルエットを捉えているため、多少の形状変化には強い特徴量である。しかしその反面、SIFT 等に比べると誤検出も生じやすいという問題がある。一般的に、人間は姿勢変化や服装の変化が伴うため、HOG は主に人物検出や顔検出に用いられることが多い。

2.2 HOG のアルゴリズム

まず画像をセルと呼ばれる単位に分割する。

このセル内の勾配強度 m と勾配方向 θ を、それぞれ式(2.1)、式(2.2)、式(2.3)、式(2.4)から求める。Lはピクセルの画素値を表す。

$$m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (2.1)$$

$$\theta(x, y) = \tan^{-1} \frac{f_y(x, y)}{f_x(x, y)} \quad (2.2)$$

第2章 従来の HOG のアルゴリズム

$$f_x(x, y) = L(x + 1, y) - L(x - 1, y) \quad (2.3)$$

$$f_y(x, y) = L(x, y + 1) - L(x, y - 1) \quad (2.4)$$

このとき勾配方向 $\theta(0\sim 180)$ を 9 方向に量子化する。その後、勾配方向についてのヒストグラムを作成する。ヒストグラムの例を図 2-1 に示す。

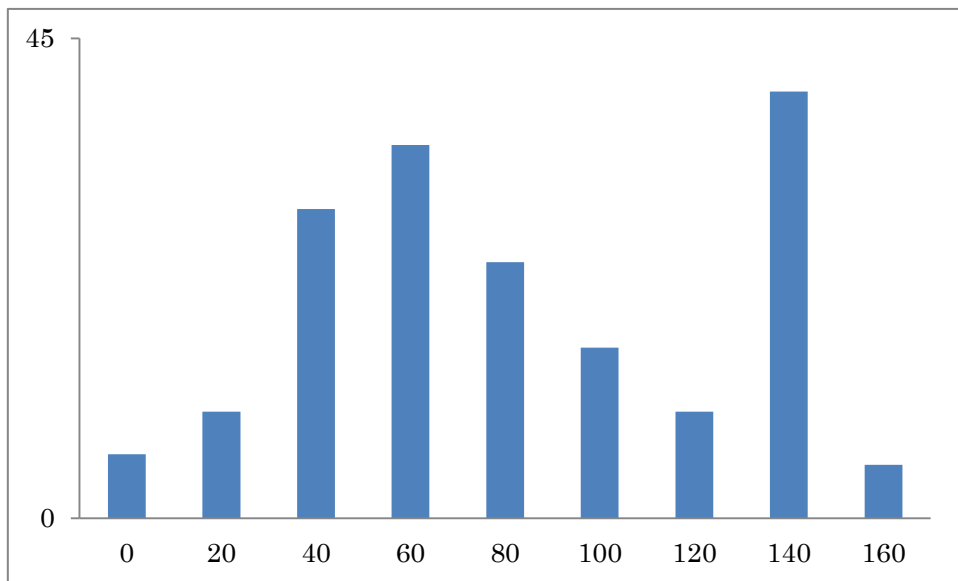


図 2-1 HOG の勾配方向ヒストグラム例

ヒストグラムを作成後、ブロック ($q \times q$ セル) 毎に正規化を行う。この正規化は 1 セルずつブロックを移動させながら行う。

つまり図 2.2 のように 1 個のセルにつき複数回正規化を行うことになる。図 2-2 で 3×3 セルと記述のある部分が、1 ブロックに相当する。ブロックのサイズとセルのサイズは任意に変更することができる。

第2章 従来の HOG のアルゴリズム

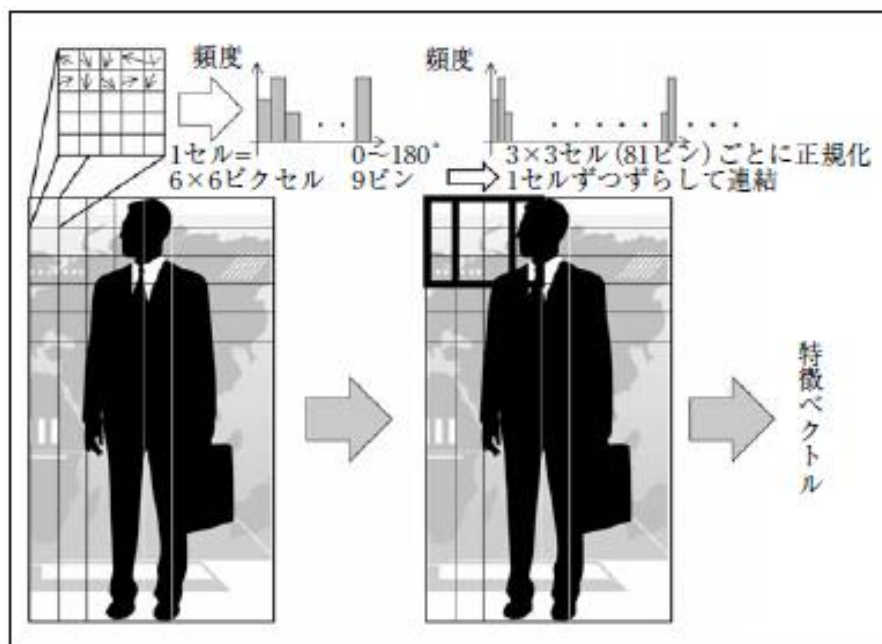


図 2-2 勾配方向ヒストグラムの正規化[9]

$$v'(n) = \frac{v(n)}{\sqrt{(\sum_{k=1}^{q \times q \times N} v(k)^2) + \epsilon}} \quad (2.5)$$

式(2.5)によって正規化されたヒストグラムをすべて結合したものが HOG 特徴量となる。式(2.5)の分母が 1 ブロック内のヒストグラムの総和を表し、分子はそのヒストグラムの一つである。 ϵ はこの式において、分母が 0 になり計算ができなくなることを防ぐ役割を持ち、 N は勾配方向数である。

第3章

提案手法

3.1 まえがき

本研究は、HOG 特徴量を回転不変に扱えるように改良を行った。まず注目領域の主要方向を算出する。次に各注目領域の主要方向が重なるように回転を行う。この2つのステップについて説明を行う。

3.2 主要方向算出

主要方向は、「注目領域内で最も強い勾配方向」と定義する。勾配方向の bin ($0^\circ \sim 179^\circ$ まで 180 方向) に対して勾配強度も加味した重み付け投票を行い、最も値の大きい方向を、その注目領域内の主要方向とする。

注目領域内のあるピクセルに対し、勾配強度 m と勾配方向 θ をそれぞれ式(3.1)、式(3.2)、式(3.3)、式(3.4)から計算する。この部分までは、前章のセル内での勾配計算と同じであるが、勾配方向を 20° ではなく 1° で量子化する。

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \quad (3.1)$$

第3章 提案手法

$$\theta(x, y) = \tan^{-1} \frac{f_y(x, y)}{f_x(x, y)} \quad (3.2)$$

$$f_x(x, y) = L(x + 1, y) - L(x - 1, y) \quad (3.3)$$

$$f_y(x, y) = L(x, y + 1) - L(x, y - 1) \quad (3.4)$$

勾配方向を横軸にとり、図(3.1)のように重み付けを行う。この重み付けにはガウス関数を用いている。ガウス関数を式(3.5)に示す。 μ が中央値を表し、これは式(3.4)までで求められた勾配方向に値する。 x の範囲によって投票を行う範囲を制御することができる。今回は($\mu - 5 \leq x \leq \mu + 5$)を x の範囲としたため、近傍 10° まで投票をおこなった。 σ^2 はガウス関数の勾配を決定する役目を持っている。式(3.5)で算出された重みに、さらに勾配強度をかえて重み付け投票をする。

上記のガウス関数を用いて重み付け投票を行う理由と、勾配方向 1° で量子化する理由を以下に示す。

(1) 従来の HOG にない角度精度の保持

第1章でも述べたとおり、従来の HOG でもある程度の回転であれば対応することができるが、 1° という細かい単位の回転には対応することができない。従来の HOG にない精度を得るために細かい回転にも対応する必要がある。

(2) 投票される対象の多さ

主要方向は $0 \sim 180^\circ$ の範囲で決定する。つまり1個のピクセルに対し180個の投票対象がある。投票対象が膨大なため、投票結果の蓄積が少なくなる恐れがあるため、ガウス関数によって、1個のピクセルが投票する対象を増やしている。

第3章 提案手法

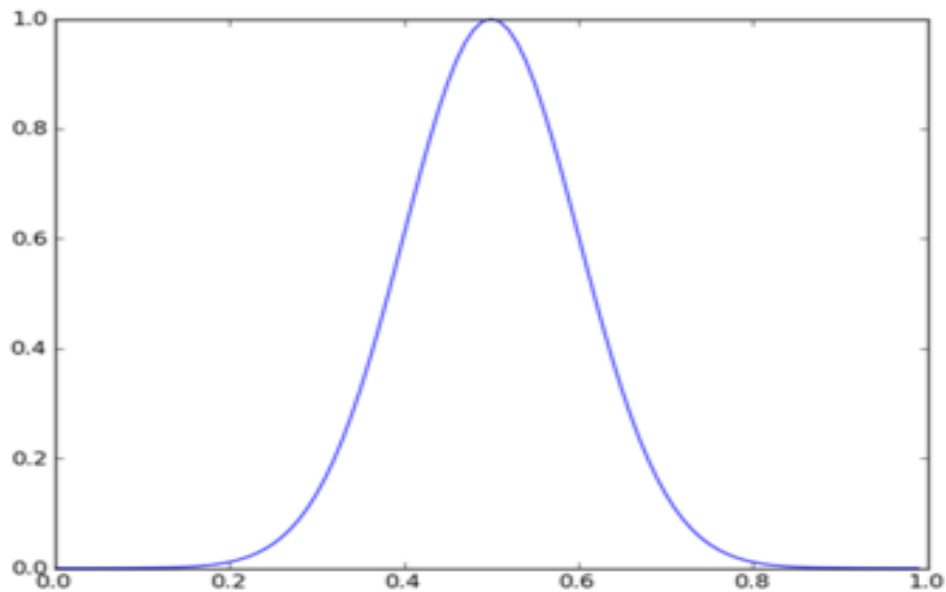


図 3-1 ガウス関数の例示[10]

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \quad (3.5)$$

上記の重み付け投票を、注目領域内すべてのピクセルに対し行い、主要方向を決定する。

3.3 画像の回転

主要方向が決定したのち注目領域の回転を行う。画像の回転角度は、主要方向によって決定される。図 3-2 のように各注目領域の主要方向が同じ角度になるように回転を行う。注目領域内にある物体が同じであった場合、図 3-2 のように物体は高い確率で同じ方向を向くと考えられる。回転にはアフィン変換を利用する。基本的なアフィン変換の式を式(3.6)に示す。また、今回の実験は OpenCV を利用した。OpenCV はコンピュータビジョン向けのオープンソースの画像処理ライブラリである。OpenCV を用いたときの回転の例として図 3-3 を示す。

画像を回転するにあたり、画像境界の外側に存在するピクセルの値が必要になる。今回は、画像の最も外側のピクセル値をそのまま挿入することとした。

第3章 提案手法

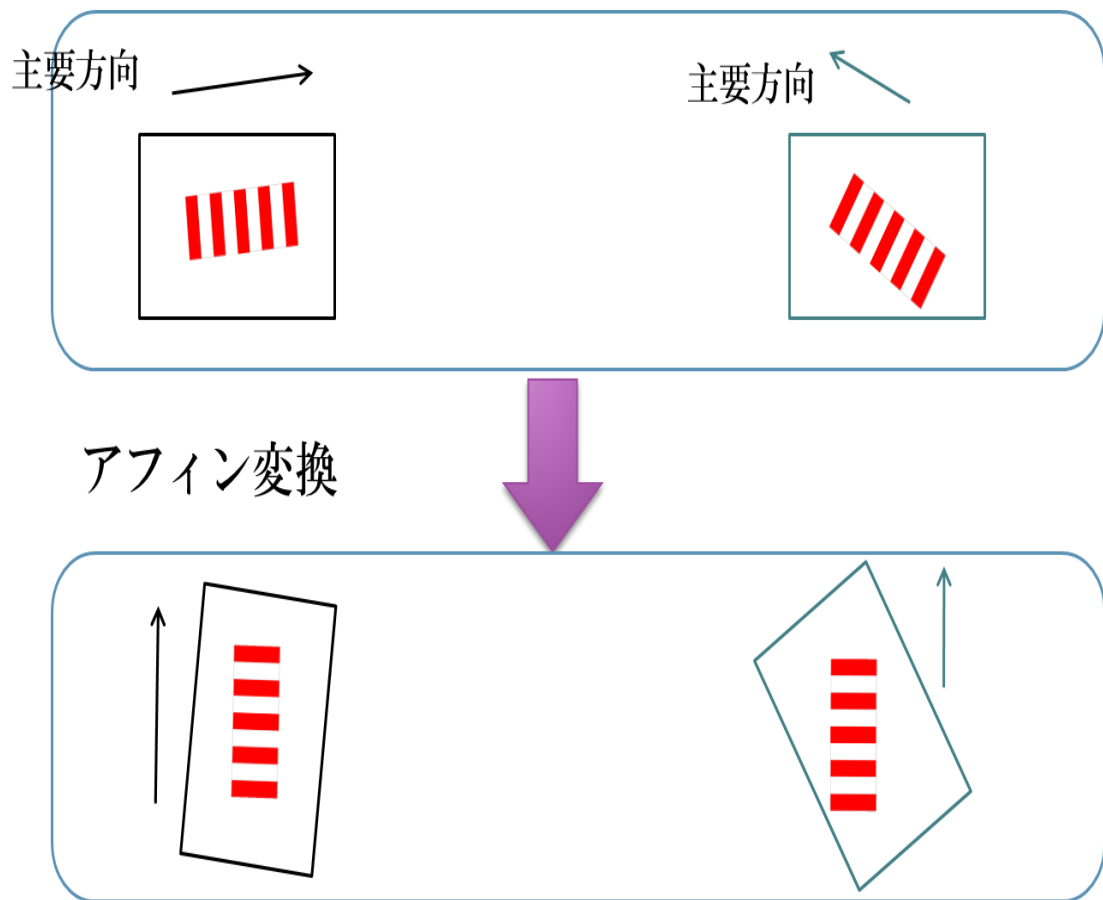


図 3-2 算出した主要方向とアフィン変換による回転の関係

$$[x' \quad y' \quad 1] = [x \quad y \quad 1] \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

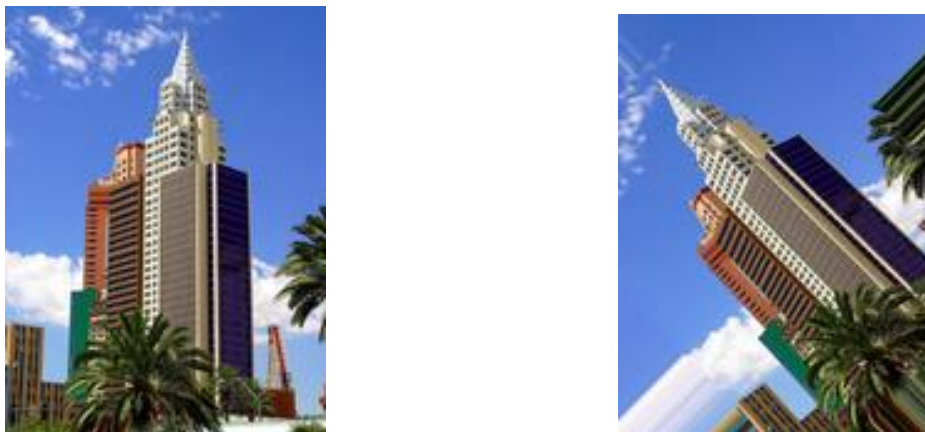


図 3-3 画像回転の例[11]

第3章 提案手法

3.4 処理の流れ

以下に提案手法における特徴量算出処理フローを示す。

- (1) 入力画像に対して、まず主要方向の算出を行う。
- (2) 主要方向を算出したら、それに従い画像の回転を行う。
- (3) 回転した画像に対して HOG 特徴量を算出する。
- (4) すべての画像の特徴量が算出されるまで (1) から (3) を繰り返す。

提案手法では、HOG 特徴量の算出に前処理を付加した形で処理を行う。

第4章

従来の HOG と提案手法における 検出精度の比較実験

4.1 まえがき

前章で述べた提案手法と従来の HOG を用いて、回転している検出対象に対する検出精度について比較を行う。

4.2 評価指標

今回の実験では、検出正解率 P と、認識率 Q を評価指標として用いる。以下に検出正解率 P と認識率 Q の定義を示す。

$$P = \frac{\text{正解検出数}}{\text{全検出数}} \quad (4.1)$$

$$R = \frac{\text{認識できた対象数}}{\text{認識すべき対象の総数}} \quad (4.2)$$

4.3 検出精度に関する比較実験

4.3.1 実験条件

実験には学習用画像として人画像（ポジティブ画像）を 45 枚、非人画像（ネガティブ画像）を 200 枚使用した。これらの画像は、HOG、提案手法ともに同じものを使用した。検出用の画像は 30 枚用意し、この画像の中には、累計 146 人の人物がいる。30 枚の画像の回転範囲は $0\sim 42^\circ$ 、 $135\sim 177^\circ$ となっており、 3° 刻みで回転している。これらの画像を用いて実験を行う。

今回の実験で用いた HOG は、中部大学藤吉研究室のチュートリアル資料にあるソースコードを利用し、画像はチュートリアル資料内のデータセットを使用した。[12]



図 4-1 学習用画像（人画像）の例

第4章 従来のHOGと提案手法における検出精度の比較実験

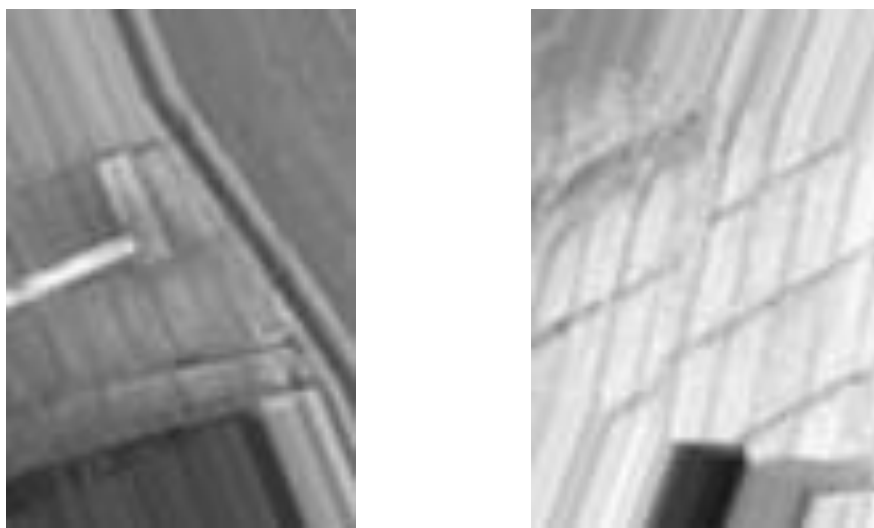


図 4-2 学習用画像（非人画像）の例



図 4-3 検出用画像の例

第4章 従来の HOG と提案手法における検出精度の比較実験

4.3.2 実験結果の比較

表 4-1 に実験の結果を示す。

表 4-1:HOG と提案手法の比較実験結果

| | 総検出数 | 正解検出数 | 平均検出正解率P | 平均認識率R |
|------|------|-------|----------|--------|
| HOG | 52 | 0 | 0 | 0 |
| 提案手法 | 372 | 96 | 0.258 | 0.658 |

表 4-1 の結果からわかるように、HOG では回転対象の検出が不可能である。図 4-4 に実際の検出の例を示す。画像内の青枠が、HOG が人だと判断し検出した部分である。研究背景でも述べたように、やはり HOG は回転対象に対する検出ができないことが確認できた。



図 4.4 HOG の検出結果

一方、提案手法では認識率だけ見ると 65.8% となり、過半数は検出できた。HOG が一人も検出できていないことを考えれば、認識率の面では大幅に改善できたといえる。しかし不正解の検出数が多いため、正解検出数に関しては 25.8% と良好な数値ではない。図 4-5、図 4-6、図 4-7 に提案手法における実際の検出の例を示す。図 4-5 は回転角度が 0° の画像での検出結果、図 4-6 は 30 枚の画像に対して検出を行った結果、最も検出正解率が高かったもの、図 4-7 は最も検出正解率が低かったものである。

第4章 従来のHOGと提案手法における検出精度の比較実験

図 4-5 回転角度 0° の画像での検出結果

図 4-6 提案手法において最も検出正解率の高かった結果

第4章 従来の HOG と提案手法における検出精度の比較実験

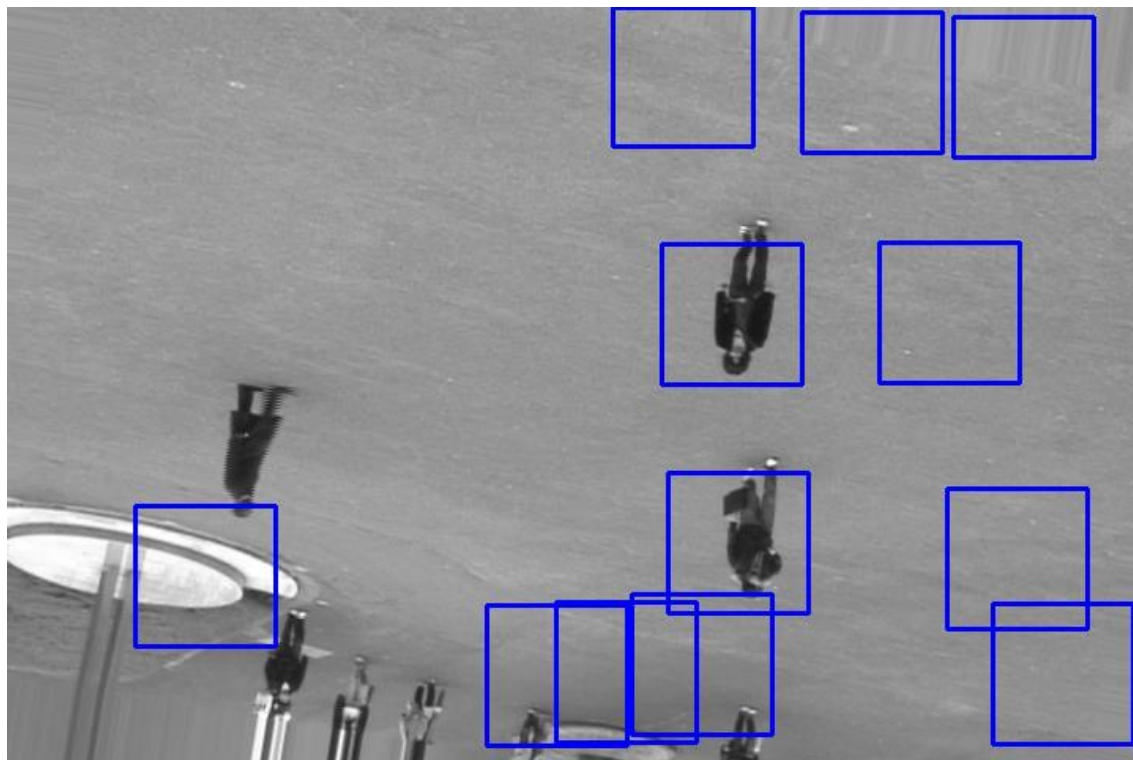


図 4-7 提案手法において最も検出正解率の低かった結果

4.3.3 提案手法の結果に対する考察

提案手法の検出正解率の低さについては、検出ウィンドウの重なり部分が多いことから、検出ウィンドウの統合がうまくいっていないことが一因ではないかと考えられる。検出ウィンドウの統合には従来手法、提案手法ともに **Mean Shift** を用いている。検出ウィンドウの重なりが顕著であった結果を図 4-8 に示す。また図 4-6 の左上、右下などは、画像の回転をおこなう際に画像境界の外側のピクセルを、最も外側のピクセルとした部分である。画像境界外のピクセルの扱いが誤検出の一因とも考えられるが、今回使用した画像は従来存在していた画像に回転を加えたものである。回転に対応することができれば検出用の画像を回転させるとは考えにくいため、現段階では問題としない。学習用画像に関しては、対象の輪郭が画像境界に被っていると回転の際の影響が大きいため、対象が画像の中心に確実にあることが求められる。

表 4-2 に画像ごとの回転角度、検出正解率をまとめたもの、図 4-9 に画像と検出正解率のグラフを示す。ある一定の回転角度で著しく検出率が落ちるということは認められなかった。最も検出率が低かった図 4-7 の結果においても、人物の認識はできていることが確認できる。これは、提案手法が回転に対応できているという証拠になると考えられる。

第4章 従来のHOGと提案手法における検出精度の比較実験



図 4-8 検出ウィンドウの重なりが顕著な例

第4章 従来のHOGと提案手法における検出精度の比較実験

表 4-2 画像ごとの検出正解率 P と回転角度

| 画像No. | 総検出数 | 正解検出数 | 検出正解率P | 回転角度 |
|-------|------|-------|--------|------|
| 1 | 7 | 2 | 0.286 | 135 |
| 2 | 11 | 4 | 0.364 | 138 |
| 3 | 8 | 2 | 0.250 | 141 |
| 4 | 12 | 3 | 0.250 | 144 |
| 5 | 14 | 4 | 0.286 | 147 |
| 6 | 15 | 4 | 0.267 | 150 |
| 7 | 14 | 4 | 0.286 | 153 |
| 8 | 13 | 3 | 0.231 | 156 |
| 9 | 15 | 3 | 0.200 | 159 |
| 10 | 11 | 3 | 0.273 | 162 |
| 11 | 13 | 3 | 0.231 | 165 |
| 12 | 10 | 2 | 0.200 | 168 |
| 13 | 7 | 2 | 0.286 | 171 |
| 14 | 12 | 2 | 0.167 | 174 |
| 15 | 10 | 2 | 0.200 | 177 |
| 16 | 10 | 3 | 0.300 | 0 |
| 17 | 12 | 3 | 0.250 | 3 |
| 18 | 9 | 3 | 0.333 | 6 |
| 19 | 16 | 4 | 0.250 | 9 |
| 20 | 13 | 3 | 0.231 | 12 |
| 21 | 16 | 4 | 0.250 | 15 |
| 22 | 16 | 3 | 0.188 | 18 |
| 23 | 15 | 5 | 0.333 | 21 |
| 24 | 13 | 5 | 0.385 | 24 |
| 25 | 16 | 4 | 0.250 | 27 |
| 26 | 13 | 4 | 0.308 | 30 |
| 27 | 12 | 3 | 0.250 | 33 |
| 28 | 12 | 3 | 0.250 | 36 |
| 29 | 13 | 3 | 0.231 | 39 |
| 30 | 14 | 3 | 0.214 | 42 |

第4章 従来のHOGと提案手法における検出精度の比較実験

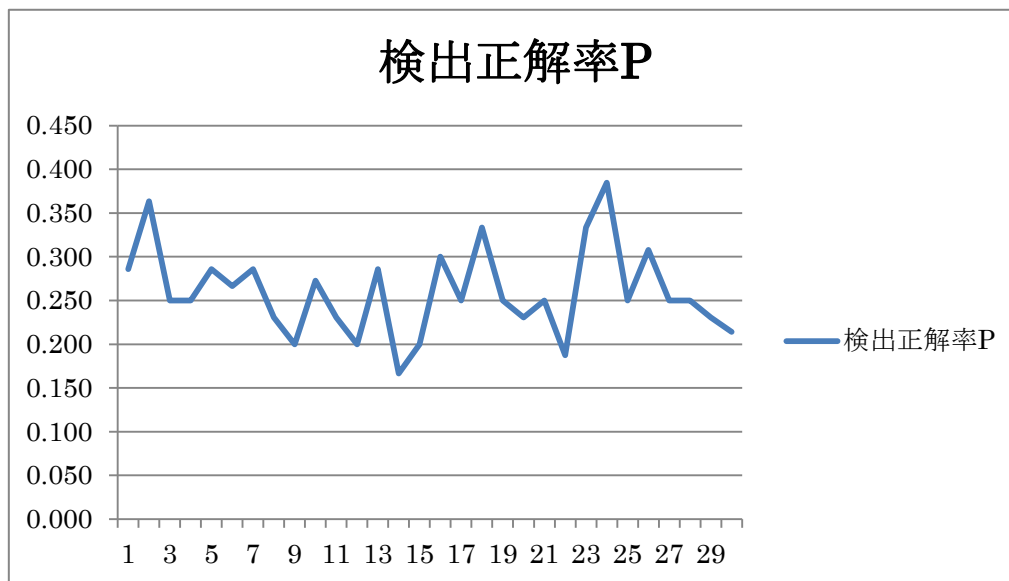


図 4-9 検出正解率 P のグラフ

第5章

結論

5.1 総括

第1章では研究の背景及び、本研究の目的、構成を述べた。

一般物体認識の技術はコンピュータの性能が向上するとともに、セキュリティからマーケティングに至る幅広い分野から注目を集めてきている。この一般物体認識において利用される特徴量が **Histograms of Oriented Gradients (HOG)** であり、一定の効果が認められている。しかし、この特徴量は回転に対して頑健な特徴量ではないため、本研究ではこの特徴量をベースにした回転に頑健な特徴量の実現を目的とした。

第2章では、第1章で述べた **HOG** のアルゴリズム、算出方法について述べた。

HOG は勾配方向と勾配強度に注目した特徴量であり、対象の輪郭を大まかに記述することができる。

第3章では、提案手法のアルゴリズムについて述べた。提案手法の手順は大きく二つに分けられ、「主要方向算出」と「画像の回転」になる。

主要方向は「注目領域内で最も勾配強度の大きい方向」で定義した。**HOG** との差異を出すために投票対象が多くなるので、重み付け投票によって主要方向の決定をおこなうことを述べた。

画像回転に関しては、画像境界の外側のピクセル値について、画像で最も外側のピクセル値が伸びているものと仮定することを述べた。

第4章では、従来手法と提案手法の比較実験の結果について述べた。

従来手法では回転した対象にまったく対応できなかったのに対し、提案手法は、誤検出も多かったものの、従来手法より良い効果が検出で確認できた。

第5章 結論

第5章は結論であり、総括と課題について述べる。

5.2 課題

本研究で、HOGが対応できない回転にある程度対応できる特徴量を実現できた。しかし、検出精度自体は必ずしも良いとは言えず、精度改善のための課題は多い。以下に、今後の課題を示す。

(1) 検出ウィンドウの統合方法

(2) 学習用画像の角度がすべて同じであっても、様々に回転している検出画像に対応できるかどうかの検証

(3) 主要方向の算出方法

(3)については、今回の算出方法である程度の有用性を確認できたが、他の算出方法と比較する必要がある。(2)は、比較実験によって結果を得なければならない。HOGに対しても同様の実験を行い、学習したデータに対して、角度の許容範囲を推定することが想定される。(1)は、今回の提案手法の精度の向上に直結した問題であると考えられる。

謝辞

本研究の機会及び素晴らしい実験環境を与えて下さり研究の方向性等、貴重な時間を割いて御指導頂きました渡辺 裕 教授に心から感謝致します。

本研究のきっかけを与えて下さり、研究の進め方に対して様々なご助言を頂き辛抱強く御指導下さった石井 大祐 助手に感謝致します。

貴重な御意見、様々な御提案を下さったゼミの皆様に御礼申し上げます。

最後に、私をここまで育てて下さった家族に深く感謝します。

2014年2月6日

参考文献

[1]山内 悠嗣、山下 隆義、藤吉 弘亘"[サーベイ論文]統計的学習手法による人検出" 情報処理学会研究報告、Aug.2012

[2]満上 育久、服部 博憲、椋木 雅之、美濃 導彦"固定カメラ映像を対象とした HOG 人物検出器のシーン適応手法"、電子情報通信学会信学技報、Vol.109 No470、Mar.2010

[3]"Hog 人検知"

<<https://sites.google.com/site/onoguchilabathirosakiuni/thema/hog-ren-jian-zhi>>2014年1月30日アクセス

[4]"人物の顔認識-リコー"

<<http://www.ricoh.co.jp/dc/photostyle/knowledge/laboratory/020/lab2.html>>2014年1月28日アクセス

[5]"カメラレビューで顔検出を行う Tech Booster"

<<http://techbooster.jp/andriod/multimedia/10375/>>2014年2月3日アクセス

[6]松本 洋平"HOG 特徴量を用いた船舶画像認識"、日本航海学会講演予稿集1巻1号、Apr.2013

[7]石井 大祐、山崎 太一、渡辺 裕"マンガ固有の特徴を利用したマンガ登場人物識別に関する一検討"、情報処理学会研究報告、Feb.2013

[8]"Gradient ベースの特徴抽出 -SIFT と HOG-"

<<http://www.vision.cs.chubu.ac.jp/04/pdf/TS01.pdf>>2014年2月2日アクセス

[9]山崎俊彦「画像の特徴抽出 Histogram of Oriented Gradients (HoG)」

<https://www.hal.t.u-tokyo.ac.jp/paper/2009/Journal_15.pdf> 2014年1月26日アクセス

[10]技術評論社、中谷秀洋「第 14 回 ベイズ線形回帰を実装してみよう」
<gihyo.jp/dev/serial/01/machine-learning/0014> 2014 年 1 月 24 日アクセス

[11]「OpenCV サンプルコード」
<opencv.jp/sample/sampling_and_geometricaltransforms.html> 2014 年 1 月 20 日アクセス

[12]"CV Tutorial"<http://www.vision.cs.chubu.ac.jp/CVTutorial/>2014 年 2 月 5 日アクセス

付録 A 実験で使用したソースコード

```
//ウィンドウサイズは 64×64
//HOG 特徴量の個数は 1764
//サンプル数、学習回数は適宜変更
//学習用(HOG+RealAdaBoost)への変更点
//CompHist へ追加
    double grad;
    double magnitude;
    double sum_magnitude = 0.0;
    double gaus[180];      //主方向決定用
    for(i=0;i<180;i++){
        gaus[i]=0;
    }
    int rograd=0;  //主方向
    float *m;
    m = new float[6];
    unsigned char* imgSource = (unsigned char*)img->imageData;
    IplImage* img2;
    img2 = cvCloneImage(img);
    CvMat M;
//主方向決定ステップ
    //y 軸方向のボックスの移動
    for(y=0;y<SET_Y_SIZE;y++){
        //x 軸方向のブロックの移動
        for(x=0;x<SET_X_SIZE;x++){
            //差分
            if(x == 0){
```

```

        xgrad = imgSource[y*wStep+(x+0)*bpp] -
imgSource[y*wStep+(x+1)*bpp];
        }else if(x == img->width - 1){
        xgrad = imgSource[y*wStep+(x-1)*bpp] -
imgSource[y*wStep+(x+0)*bpp];
        }else{
        xgrad = imgSource[y*wStep+(x-1)*bpp] -
imgSource[y*wStep+(x+1)*bpp];
        }
        if(y == 0){
        ygrad = imgSource[(y+0)*wStep+x*bpp] -
imgSource[(y+1)*wStep+x*bpp];
        }else if(y == img->height - 1){
        ygrad = imgSource[(y-1)*wStep+x*bpp] -
imgSource[(y+0)*wStep+x*bpp];
        }else{
        ygrad = imgSource[(y-1)*wStep+x*bpp] -
imgSource[(y+1)*wStep+x*bpp];
        }

```

//大きさ

```
magnitude = sqrt(xgrad * xgrad + ygrad * ygrad);
```

//勾配の大きさの計算

```
grad = atan2 (ygrad, xgrad);
```

//ラジアンから角度へ変換

```
grad = (grad * 180)/PI;
```

```
//grad -= 1.0;
```

//マイナスは反転させる

```
if(grad<0.0){
```

```
    grad+=360.0;
```

```
}
```

//0~360度を0~180度にする

```
if(grad>180.0){
```

```
    grad = grad-180.0;
```

```
}
```

```

//cout << grad << endl;
//近傍の 10 (自身含めて 11) だけ反映
double ep = 1.5; //重み

for(i=-10;i<10;i++){
    //cout << grad+ i << " $" << endl;
    if((grad+i)>=180){
        z=grad+i-180;
    }
    else if((grad+i)<0){
        z=grad+i+180;
    }
    else{
        z=grad+i;
    }
    gaus[z]=gaus[z]+magnitude*exp((-pow(i,
2.0))/(2*pow(ep, 2.0)));
}
}

//上記を繰り返したのち、最大値 (主方向) を求める
for(i=0;i<180;i++){
    if(gaus[i]>d){
        d=gaus[i];
        rograd = i;
        //cout << gaus[i] <<"hairetu" << i << endl;
        //cout << rograd << endl;
    }
    else{
    }
}

//アフィン変換
//cout << rograd << endl;

```

```

m[0]= (float) (cos (rograd * PI / 180.0));
m[1]= (float) (-sin (rograd * PI / 180.0));
m[2] = img->width * 0.5;
m[3] = -m[1];
m[4] = m[0];
m[5] = img->height * 0.5;
cvInitMatHeader (&M, 2, 3, CV_32FC1, m, CV_AUTOSTEP);

// 指定された回転行列により, GetQuadrangleSubPix を用いて画像全体を回
転させる
cvGetQuadrangleSubPix (img, img2, &M);

//char* winOriginal = "Image";
//cvNamedWindow(winOriginal,1);
//cvShowImage(winOriginal,img2);
//cvResetImageROI(img2);
//cvWaitKey0;
unsigned char* img2Source = (unsigned char*)img2->imageData;
//今後ヒストグラム計算内の imgSource を img2Souce に変更

//検出用(Detector)への変更点
//学習用で使った HOG をプロジェクトに追加
//IntegralHistogram は用いない。
//Common.h 内の検出ウィンドウの最小スケールは 1.4 に変更した
//検出ウィンドウのラスタスキャン

for(j=0;j<imgSource->height-window_height;j+=STEP_WIDTH){

for(i=0;i<imgSource->width-window_width;i+=STEP_WIDTH){

//ROI で関心領域を切り出す
roi[frame].x=i;
roi[frame].y=j;
roi[frame].width=window_width;
roi[frame].height=window_height;

```



```
cvSetImageROI(imgGray, roi[frame]);
tmp =
cvCreateImage(cvSize(window_width,window_height),8,1);

cvCopy(imgGray,tmp);
cvResize(tmp,tmp2);
cvReleaseImage(&tmp);
//cout << tmp2->width << " " <<
tmp2->height << endl;

//cvShowImage(winOriginal,tmp2);
//cvResetImageROI(imgGray);
//cvWaitKey();
//特征量抽出
ob_HOG.Init(tmp2);
ob_HOG.CompHist(tmp2);
ob_HOG.Getfeature(feature);

//ob_IH.Getfeature(feature,i,j,window_width,window_height);
```

図一覧

| | | |
|-----|-------------------------------|----|
| 2-1 | HOGの勾配方向ヒストグラム例..... | 5 |
| 2-2 | 勾配方向ヒストグラムの正規化([9]より引用)..... | 6 |
| 3-1 | ガウス関数の例示([10]より引用)..... | 9 |
| 3-2 | 算出した主要方向とアフィン変換による回転の関係..... | 10 |
| 3-3 | 画像回転の例([11]より引用)..... | 10 |
| 4-1 | 学習用画像(人画像)の例..... | 13 |
| 4-2 | 学習用画像(非人画像)の例..... | 14 |
| 4-3 | 検出用画像の例..... | 14 |
| 4.4 | HOGの検出結果..... | 15 |
| 4-5 | 回転角度 0° の画像での検出結果..... | 16 |
| 4-6 | 提案手法において最も検出正解率の高かった結果..... | 16 |
| 4-7 | 提案手法において最も検出正解率の低かった結果..... | 17 |
| 4-8 | 検出ウィンドウの重なりが顕著な例..... | 18 |
| 4-9 | 検出正解率Pのグラフ..... | 20 |

表一覧

| | |
|-----------------------------|----|
| 4-1 HOG と提案手法の比較実験結果..... | 15 |
| 4-2 画像ごとの検出正解率 P と回転角度..... | 19 |

