

領域画像マッチングのための初期探索法

山下 幸彦[†] 若原 徹^{††}

[†] 東京工業大学

東京都目黒区大岡山 2-12-1

^{††} 法政大学

東京都小金井市梶野町 3-7-2

E-mail: tyamasita@ide.titech.ac.jp, ††wakahara@hosei.ac.jp

あらまし 画像マッチングは、画像処理における最も重要な手法の一つである。その中で領域マッチング法に分類される GAT/GPT (Global Affine Transform/Global Projection Transform) 相関法は、高精度なマッチングを実現することができる。しかしながら、GAT/GPT 相関法で直接 whole-to-part のマッチングを実現する場合、多数の大きさのスライディングウィンドウを用いる必要があるため、計算時間がかかるという問題がある。本稿では、スライディング離散フーリエ変換を用いた領域マッチングのための初期位置を探索する高速な手法を提案する。さらに、計算機実験によって、ASIFT と RANSAC を組み合わせた手法との比較を行い、提案手法の有効性を示す。

キーワード 画像マッチング, 領域マッチング, GAT/GPT 相関, 初期探索

An initial search method for region-based image matching

Yukihiko YAMASHITA[†] and Toru WAKAHARA^{††}

[†] Tokyo Institute of Technology

2-12-1, O-okayama, Mekuro-ku, Tokyo, 152-8552

^{††} Hosei University

3-7-2 Kajino-cho, Koganei-shi, Tokyo 184-8584

E-mail: tyamasita@ide.titech.ac.jp, ††wakahara@hosei.ac.jp

Abstract GAT/GPT (Global Affine Transform/Global Projection Transform) correlation methods for image matching that is one of the most important methods in the field of image processing can perform a precise image matching. When GAT/GPT correlation methods are applied to whole-to-part matching, it can be realized by sliding windows with various sizes but it needs much calculation time. In this paper we propose a fast initial search method using the sliding Discrete Fourier Transformation for the region-based image matching. Furthermore, by experiments we compare and evaluate the proposed method against the combination of ASIFT and RANSAC.

Key words Image matching, region-based matching, GAT/GPT correlation, initial search

1. はじめに

画像マッチングは、物体検出、パターン認識などに応用され、画像処理における最も重要な手法の一つである。画像マッチングを実用的なものにするためには、テンプレート画像に対して、入力画像での位置や照明の変化の他、回転を含む変換や、場合によっては物体変形に対して頑健である必要がある。そのために、数多くの画像マッチング手法が提案されている。それらを大きく分ければ、画像から特徴点を抽出し、その特徴点の整合性を評価する特徴点マッチング法と、画像の領域の整合性を直接評価する領域マッチング法の2つが存在する。特徴点マッチング

法の中では、Scale-Invariant Feature Transform (SIFT) [1], [2] に基づく手法が広く用いられており、SIFT を Affine 変換に頑健にした ASIFT [3] と RANSAC [4] の組み合わせは、実用的で頑健な画像マッチングを可能にしている。領域マッチング法は、ガウス平滑化や固有値展開によりパターン変動を吸収した飯島による複合類似度 [5] や、変換を直接求める Lucas-Kanade アルゴリズム [6] などに始まり、現在でもその性能を高めるべく研究が続けられている [7]。その中で、若原らによる GAT/GPT (Global Affine Transform/Global Projection Transform) 相関法 [8]~[10] は、評価基準を微分して得られる非線形方程式を線形近似し、その線形方程式の解で与えられる変換で入力画像を



(a) 原画像 (b) ガウスボケ画像 (c) 横方向微分画像 (d) 縦方向微分画像 (e) LoG 画像

図 1 三角関数のスライディング畳み込み和によるガウス平滑化法

変換することを繰り返すことにより、特徴点に基づく手法よりも高精度なマッチングを可能にしている。例えば、GPT 相関法に同一エッジ方向平均最小距離使った k NN による MNIST の手書き文字認識では、認識率 99.7% という、 k NN に基づく手法としては、非常に高い値が得られている。

MNIST の認識では、入力画像とテンプレート画像とも、文字を切り出した後のマッチングであるため、whole-to-whole マッチングに分類できる。Computer Vision の応用では、入力画像の一部分にテンプレート画像の物体が写っており、それを検出する whole-to-part マッチングが重要である。領域マッチング法でも、適切な初期位置から計算を始めれば、計算時間は特徴点マッチング法とほぼ同程度になる。しかしながら、適切な初期位置がわからない場合に、領域マッチング法で whole-to-part を実現するためには、多数の大きさのスライディングウィンドウから切り出した入力画像に対して、マッチングを試みる必要がある。このようにすることで、マッチングは可能であるが、特徴点マッチング法に比べて計算時間がかかるという問題がある。従って、高速な初期位置探索法を研究する必要がある。

スライディング離散フーリエ変換を用いたガウス平滑化は、その計算量のオーダーが平滑化量に依存せず、画素数に比例したものになる [11]~[14]。本稿では、このガウス平滑化を利用し、SIFT の特徴点での特徴量に基づく領域特徴量とその計算法、そしてその特徴量を使った初期位置の探索法を提案する。さらに、計算機実験によって、ASIFT と RANSAC を組み合わせた手法との比較を行い、提案手法の有効性を示す。

2. 有限区間三角関数の畳み込み和によるガウス平滑化法

2.1 ガウス平滑化の近似実現

1次元離散点上の関数で記述する。ガウス関数は標準偏差の数倍で急速に 0 に近づくため、有限区間 $[-K, K]$ で考え、 $G[n] = e^{-(\pi n/K)^2/(2\sigma^2)}$ による平滑化を考える。 σ は、区間両端を $\pi, -\pi$ と考えたときの標準偏差で、画像上での標準偏差は $\sigma K/\pi$ である。関数 $f[n]$ に対して、ガウス平滑化やその 1 階、2 階微分の近似を求めるためには、次の値が求めれば良い。

$$\sum_{k=-K}^K G[k]f[n-k] \quad (1)$$

$$\sum_{k=-K}^K kG[k]f[n-k] \quad (2)$$

$$\sum_{k=-K}^K k^2G[k]f[n-k] \quad (3)$$

簡単のため、 $\beta = \pi/K$ とおき、次の近似を考える。

$$G[k] \simeq \sum_{p=0}^P a_p \cos(\beta pk)$$

$$kG[k] \simeq \sum_{p=1}^P b_p \sin(\beta pk)$$

$$k^2G[k] \simeq \sum_{p=0}^P d_p \cos(\beta pk)$$

P は計算量を減らすため、DFT を打ち切る次数である。ここでは、 P として 2 から 4 程度を想定している。

有限区間三角関数の畳み込み和を次式で定義すれば、

$$c_p[n] = \sum_{k=-K}^K f[n-k] \cos(\beta pk)$$

$$s_p[n] = \sum_{k=-K}^K f[n-k] \sin(\beta pk)$$

式 (1)-(3) の近似は、次式で表すことができる。

$$\sum_{k=-K}^K G[k]f[n-k] \simeq \sum_{p=0}^P a_p c_p[n]$$

$$\sum_{k=-K}^K kG[k]f[n-k] \simeq \sum_{p=1}^P b_p s_p[n]$$

$$\sum_{k=-K}^K k^2G[k]f[n-k] \simeq \sum_{p=0}^P d_p c_p[n]$$

図 1 に、標準画像 Lenna を、有限区間三角関数の畳み込み和によりガウス平滑化した結果を示す。

2.2 有限区間三角関数の畳み込み和の高速計算

簡単のために、複素指数関数を使って表す。Kober は、カーネル積分

$$u[n] = \sum_{k=1}^n f[k]e^{i\beta pk} \quad (4)$$

を使って、有限区間三角関数の畳み込み和 $c_p[n]$ と $s_p[n]$ を、次

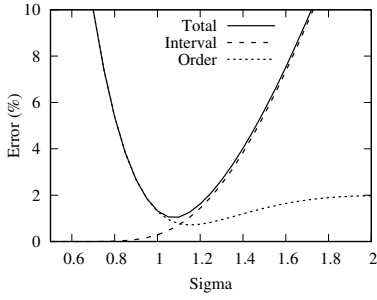
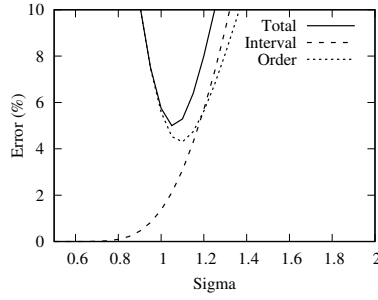
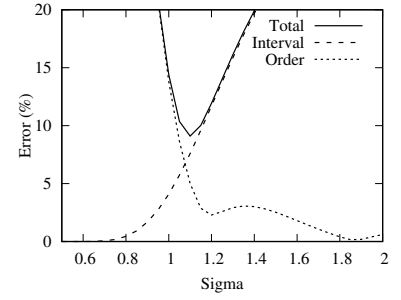
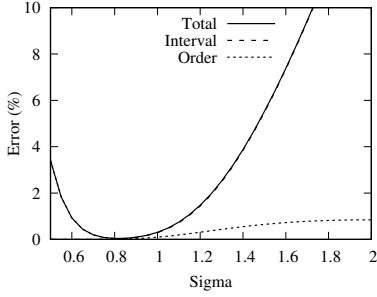
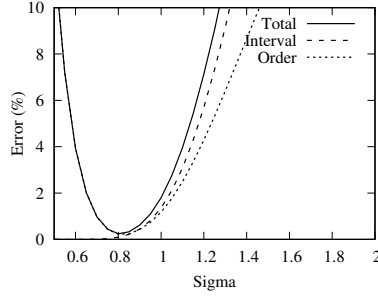
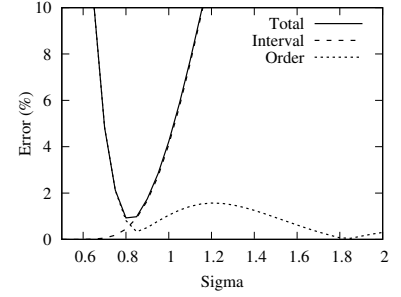
(a) $G[n]$, $K = 256$, $P = 2$ (b) $nG[n]$, $K = 256$, $P = 2$ (c) LoG , $K = 256$, $P = 2$ (a) $G[n]$, $K = 256$, $P = 4$ (b) $nG[n]$, $K = 256$, $P = 4$ (c) LoG , $K = 256$, $P = 4$

図 2 打ち切り誤差 (Interval : 区間打ち切り, Order : 次数打ち切り)

式で求めた。

$$c_p[n] + is_p[n] = e^{-i\beta pn} (u[n+K] - u[n-K-1]) \quad (5)$$

この $u[n]$ は、次の漸化式で計算することができる。

$$u[n] = u[n-1] + f[n]e^{i\beta pn} \quad (6)$$

この式では、三角関数を 1 周期分記憶して、信号にかけていく必要がある。少し変形して、IIR フィルタの形にする。

$$v[n] = e^{-i\beta p} v[n-1] + f[n] \quad (7)$$

とすれば、 $c_p[n]$ と $s_p[n]$ は、次式で求まる。

$$c_p[n] + is_p[n] = (-1)^p (v[n+K] - v[n-K] + x[n-K]) \quad (8)$$

また、積分値でなく、直接、区間和を計算することもできる。こうする方が数値計算精度は向上するが、 K , p を整数倍した場合への、積分値の再利用はできなくなる。

杉本 (2017) らは 2 次のフィルタを使うことを提案している。

$$v[n+1] = 2 \cos(\beta p) v[n] + v[n-1] + f[n+1] - e^{i\beta p} f[n] \quad (9)$$

この式では、実数部と虚数部が分離しているため、 $c_p[n]$ と $s_p[n]$ を別々に計算することが可能である。従って、 $c_p[n]$ だけが欲しい場合などに計算量を削減することができる。しかしながら、2 階の微分方程式を解く形になっており、単精度浮動小数点演算で計算したときに、計算精度に問題が生じる場合が考えられる。

2.3 打ち切り誤差

区間長や P を打ち切るため、ガウス関数の近似に誤差が生じる。画像上での標準偏差は $\sigma K/\pi$ を一定にしても、 K と σ に自由度が生じるが、打ち切り誤差は σ に大きく依存し、 K にはあ

まり依存しない。

図 2 は、 $P = 2$ と $P = 4$ のときの、 $G[n]$, $nG[n]$, LoG の近似誤差を、平均 2 乗誤差ともとの関数の 2 乗平均の比の平方根で示したものである。

$P = 2$ の場合、 $\sigma = 1.1$ とすれば、誤差はガウス関数で 1%、微分ガウス関数で 5%、 LoG で 10% 程度となる。 $P = 4$ の場合、 $\sigma = 0.85$ とすれば、ガウス関数で 0.1%、微分ガウス関数で 0.5%、 LoG で 1% 程度となる。

2.4 単精度浮動小数点演算誤差

倍精度浮動小数点演算を使えば、有限区間三角関数の畳み込み和の計算は十分な精度で計算できるが、今後重要になる安価な GPU は単精度浮動小数点演算器しか積んでいないため、単精度浮動小数点演算による精度評価を行う。標準画像 Lenna と Barbara をそれぞれ、1 次元信号として考え、連結した $524288 (= 512 \times 512 \times 2)$ 次元のベクトルを入力とした。図 3 に、 $K = 256$ のときの、 $c_1[n]$ の区間長 10,000 での最大誤差と、同区間での対象信号の 2 乗平均の平方根との比を示す。

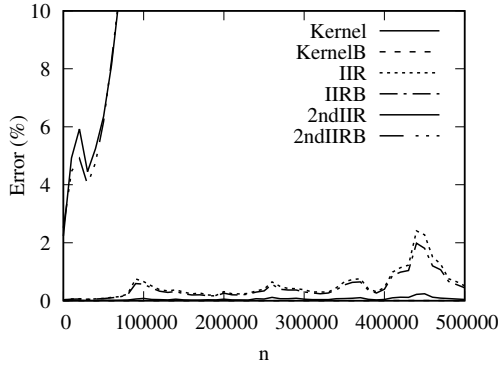
単精度浮動小数点の計算では、2 次 IIR では、初めの 1 万点で 2% 程度の誤差があり、十分とは言えない。

3. 提案手法

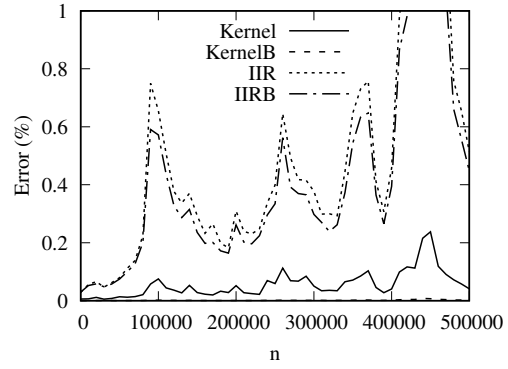
ここでは、提案手法のアルゴリズムを説明する。そのために、まず、提案手法で用いるフーリエ記述エッジ方向ヒストグラムに関して説明する。

3.1 フーリエ記述エッジ方向ヒストグラム

エッジ方向ヒストグラム $D(\theta)$ は、ある領域内の θ 方向のエッジの強さの (重み付き) 和である。これを、 Q 次のフーリエ級数で記述する。



(a) $c_1[n]$, $K = 256$



(b) $c_1[n]$, $K = 256$ (拡大図)

図3 単精度計算誤差 (Kernel : カーネル積分 (式 (6)), IIR : 1次 IIR (式 (7)), 2ndIIR : 2次 IIR (式 (9)). B を付加したものは、区間積分で計算。)

$$D(\theta) = \sum_{q=-Q}^Q d_n e^{iq\theta} \quad (10)$$

ここで、 $d_{-q} = \bar{d}_q$ が成立するため、係数 d_q の実数での自由度は $2Q + 1$ である。

入力画像 $I(x, y)$ をガウス平滑化し、横と縦方向に微分した画像を $I_X(x, y)$, $I_Y(x, y)$ とする。

$$\|I_1(x, y)\| = \sqrt{(I_X(x, y))^2 + (I_Y(x, y))^2} \quad (11)$$

とおけば、点 (x, y) におけるヒストグラムは、ディラックのデルタ関数 $\delta(\theta)$ を使って、 $\|I_1(x, y)\| \delta(\theta - \theta(x, y))$ である。このとき、 $I_X(x, y) = \|I_1(x, y)\| \cos \theta(x, y)$, $I_Y(x, y) = \|I_1(x, y)\| \sin \theta(x, y)$ が成立する。各点のフーリエ係数は、

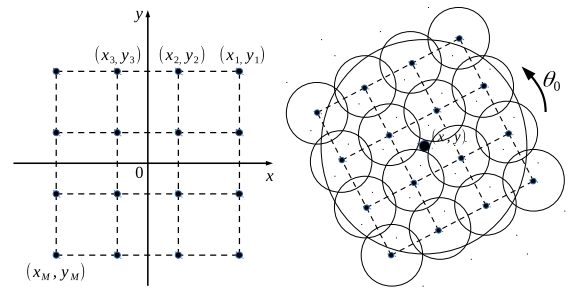
$$\begin{aligned} d_q &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \|I_1(x, y)\| \delta(\theta - \theta(x, y)) e^{-iq\theta} d\theta \\ &= \frac{1}{2\pi} \|I_1(x, y)\| e^{-iq\theta(x, y)} \\ &\simeq \frac{(I_X(x, y) - iI_Y(x, y))^q}{2\pi (\|I_1(x, y)\|^2 + \varepsilon)^{(q-1)/2}} \end{aligned} \quad (12)$$

である。ここで、 ε は正則化定数である (実験では、 $\varepsilon = 10^{-12}$)。この式を使えば、エッジ方向を量子化することなく、フーリエ記述エッジ方向ヒストグラムを求めることができる。また、エッジの基準方向を θ_0 回転したとき、回転したヒストグラムの係数 d'_q は、次式で得ることができる。

$$d'_q = d_q e^{-iq\theta_0} \quad (13)$$

3.2 領域特徴量

領域特徴量に関して説明する。入力画像 I の点 (x, y) において、スケール S の領域特徴量を記述する。 S は、ガウス関数の標準偏差で、その領域のおよその半径である。まず、 $P = 4$ とした有限区間三角関数の畳み込み和によるガウス平滑化法により、 I からスケール $S/8$ でガウス平滑化し、横と縦方向に微分した画像 I_X , I_Y を求める。正弦成分を使うことにより、ガウス平滑化と微分は同時に行うことができる。そして、画像の各点で、式 (12) を使って、フーリエ記述エッジ方向ヒストグラ



(a) 正規化座標 (b) 画像上の座標

図4 ヒストグラムを特徴量とする点の位置

ムを求める。その $2Q + 1$ の係数を、 $P = 2$ とした有限区間三角関数の畳み込み和によるガウス平滑化法により、スケールを S および $S/4$ として平滑化したものをそれぞれ求める。 (x, y) におけるスケール S のフーリエ係数から、一周を L 等分した $\theta = 2\pi l/L$ ($l = 0, 1, \dots, L-1$) におけるヒストグラムの値を求め、それが極大となる方向を、ヒストグラムの値が大きい方から2つ求める。それぞれの角度がその対象領域の基準角度であり、特徴量はその角度で正規化して求める。両者のヒストグラムの値の比が1に近いときには、2つの角度に対する特徴量をそれぞれ求める。比が1から離れているときには、最大の角度に対してだけ特徴量を求める。

(x, y) での基準角度の1つを、 θ_0 とする。提案する領域特徴量は、 θ_0 で正規化したときの、 (x, y) の周りの M 点における、スケール $S/4$ でガウス平滑化したフーリエ記述エッジ方向ヒストグラムを並べたものである。

図4は、特徴量記述に使う原点を中心とした周辺の M 点の正規化座標 (x_m, y_m) ($m = 1, 2, \dots, M$) を図示したものである。実際にヒストグラムを特徴量とする点の画像上の座標 (x'_m, y'_m) ($m = 1, 2, \dots, M$) は、次式で与えられる。

$$x'_m = S(x_m \cos \theta_0 - y_m \sin \theta_0) + x \quad (14)$$

$$y'_m = S(x_m \sin \theta_0 + y_m \cos \theta_0) + y \quad (15)$$

スケール $S/4$ のフーリエ記述エッジ方向ヒストグラムはすべての点で計算済みであるため、その値を抽出し、角度正規化のためにヒストグラムを式 (13) で回転させる。これらの係数を並

Algorithm 1 特徴量計算法

Require: 入力画像 I , 初期スケール S_0 , スケール更新比 r_S , 最大スケール S_{Max} , スケール S において特徴量を求める点の集合 T_S , 特徴量となるヒストグラムを利用する点の特徴量を求める点からの正規化座標 (x_m, y_m) ($m = 1, 2, \dots, M$)

$S \leftarrow S_0$

while $S \leq S_{\text{Max}}$ **do**

I をスケール $S/8$ でガウス平滑化し, 横と縦方向に微分した画像 I_X, I_Y を求める。

画像の各点でフーリエ記述エッジ方向ヒストグラムの係数を求める。

$2Q+1$ 個の係数を, スケール S および $S/4$ でガウス平滑化する。

for $(x, y) \in T_S$ **do**

スケール S で平滑化した係数を使って, 基準方向を (複数) 求める。

for $\theta_0 \in$ (得られた基準方向の集合) **do**

基準方向 θ_0 に対して, 式 (14), (15) を使い, ヒストグラムを利用する点の座標を求める。

それらの点におけるスケール $S/4$ のフーリエ記述エッジ方向ヒストグラムを, 式 (13) で回転させ, 並べて $M(2Q+1)$ 次元のベクトルを得る。

得られたベクトルのノルムを 1 に正規化して, (x, y) における特徴量の 1 つとして記憶する。

end for

end for

$S \leftarrow r_S S$

end while

べ, $M(2Q+1)$ 次元のベクトルが得られる。このベクトルをノルムを 1 に正規化したものが, (x, y) での最終的な特徴量となる。

特徴量を求める点 (x, y) は, 画像中を S に比例した値で, 格子点状に走査する。その比例係数としては 0.2 程度を用いる。

画像から特徴量も求めるアルゴリズムを Alogrithm 1 にまとめる。

3.3 マッチング

マッチングの基本的方法は以下のとおりである。テンプレート画像の抽出したい物体領域の中心を, 特徴量を抽出する点 (x, y) とし, そのおよその半径をスケール S として, 特徴量抽出を行い, その特徴量を記憶する。その特徴量をテンプレート特徴量と呼ぶ。入力画像を複数スケールと格子点走査で特徴量を求め, その中から, テンプレート特徴量との内積が大きい特徴量を出力した座標を, 検出位置とする。また, 両者の基準角の差は, 物体領域の回転角と考えることできる。後段処理のために, 1 点ではなく複数の候補点の抽出を行いたい場合は, 内積が大きい順番に複数個の特徴量を抽出することもできる。

しかしながら, 画像の変形量が大きいときには, 特徴ベクトルが異なってしまい, マッチング精度が下がることが考えられる。そこで, ASIFT の考え方を使って, 画像を Affine 変換し, 特徴量を抽出してマッチングさせることを考える。入力画像を変形させると, 全体の特徴抽出の計算時間が, 変形しない場合の計算時間に変形パターン数をかけた値になってしまう。

そこで, ここでは入力テンプレートだけ変形させることにする。そして, 対象物体に対する特徴量を複数用意し, 入力画像の特徴量との内積を計算し, 最大となるものを選び出す。内積の計算時間が増えることになるが, 全体の計算時間が大きく変わることはない。

また, スケールに関しても, 入力画像のスケールの種類におよそ比例して特徴抽出の計算時間が増加する。そこで, スケールの小さな変化に関してはテンプレートの方で対応することにする。例えば, 実験では, 入力画像に対するスケール変化は 2 倍 ($r_S = 2$) とし, テンプレートでは, S_0 と $\sqrt{2}S_0$ の 2 種類のスケールを用いる。このようにすると, 厳密にはマッチングする領域の大きさが多少異なることになるが, スケールを $\sqrt{2}$ ずつ変化させたマッチングが実現できる。

4. 計算機実験

提案手法の性能を調べるために, 画像から目標とする物体のおよその位置を検出する実験を行った。また, その結果を ASIFT+RANSAC と比較する。実験に用いる画像データセットは, Graffiti (800 × 640 画素) と, Boat (850 × 680) [15] である。Graffiti は, グレイスケール画像に変換して用いた。それぞれ, 6 種類の画像 (img1~img6) からなる。目標領域は, Graffiti では img1 の人物の領域, Boat では img1 のボートの領域とするために, 特徴量抽出のための中心座標およびスケールを, それぞれ, (430, 310) およびスケール 150 とその $\sqrt{2}$ 倍, 中心座標 (425, 340) およびスケール 200 とその $\sqrt{2}$ 倍とした。テンプレート画像を, 検出位置を中心として, 無変換も含めて 9 種類の Affine 変換を施し, 特徴量を求めた。目標を検出する入力画像は, img2~img6 を使った。入力画像に対して, スケールを 2 倍に変化させ, 特徴量の中心点をスケールの 20% の幅で移動させ, 特徴量を計算した。

ASIFT+RANSAC のテンプレート画像は, Graffiti のテンプレートは人物を中心に, img1 から 490 × 368 画素を切り出したもの, Boat では img1 それ自体とした。入力画像は, 提案手法と同じである。ASIFT+RANSAC は, OpenCV の SIFT と RANSAC をライブラリーとして使う python の公開されているプログラム [16] をほぼそのまま利用した。OpenCV は, version 3.4.1 を用いた。

図 5 に Graffiti の, 図 6 に Boat のテンプレート画像となる img1 と, 提案手法によって特徴量の相関が最も大きくなった, 中心点とスケールを図示した。この図から, 提案手法によって正しく物体位置が正しく検出できていることがわかる。図 5 (b)–(f), 図 6 (b)–(f) で中心から円周に伸びる線分は, エッジ方向ヒストグラムが最大になる基準方向を示している。この方向が物体に対して一定していない理由は, 複数種類あるテンプレートの基準方向が物体に対して異なるためである。ASIFT+RANSAC でも, すべての画像に対して, 正しく物体を検出できた。

表 1 に, 計算時間を示す。使用した計算機の CPU は, Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz である。表中の値は, 提案手法, ASIFT+RANSAC とともに, テンプレートの作

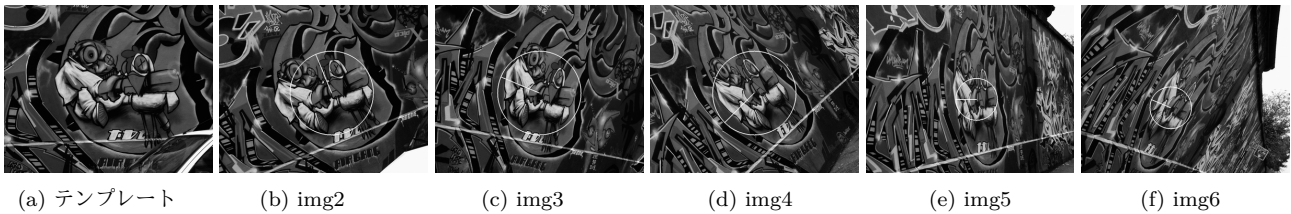


図5 Graffitiの探索結果((b)–(f)の円の中心が検出位置, 半径が検出したスケールを表す。

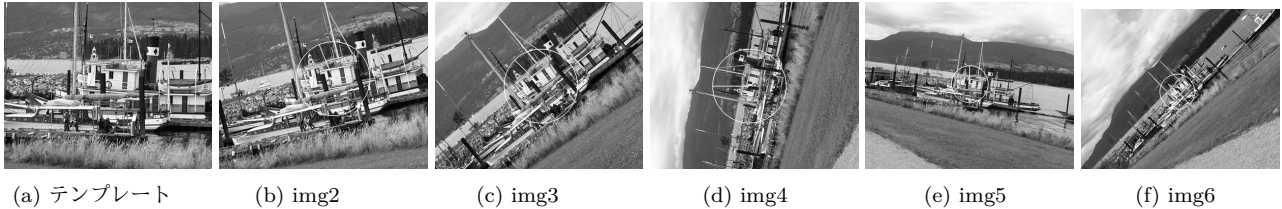


図6 Boatの探索結果((b)–(f)の円の中心が検出位置, 半径が検出したスケールを表す。)

成時間は除いたものである。

ASIFT+RANSACは、プログラムに設定されていたそのままのパラメータで計算しており、さらなる高速化が可能であるが、提案手法も十分に高速に計算できていると考えることができる。そして、スライディングフーリエ変換結果の再利用の実装や、提案手法は単純な浮動小数点演算で計算できるため、GPUによる高速化も容易である。また、本手法では、入力画像のすべての点で、スケールごとにエッジ方向ヒストグラムが計算されているため、その情報を後段の認識などに利用することも可能である。

5. まとめ

GAT/GPT 関連法などの画像の領域マッチング法のための、スライディングフーリエ変換によるガウス平滑化を用いた高速初期探索法を提案した。そして、ASIFT+RANSAC との比較実験により、提案手法の有効性を示した。今後の課題は、計算の高速化と、実際に GAT/GPT 関連法と組み合わせた画像マッチング法を実装することである。

文献

- [1] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol.60, no.2, pp.91–110, Nov. 2004.
- [2] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.506–513, CVPR’04, IEEE Computer Society, Washington, DC, USA, 2004. <http://dl.acm.org/citation.cfm?id=1896300.1896374>
- [3] J.-M. Morel and G. Yu, “ASIFT: A new framework for fully affine invariant image comparison,” *SIAM J. Img. Sci.*, vol.2, no.2, pp.438–469, April 2009.
- [4] M.A. Fischler and R.C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol.24, no.6, pp.381–395, June 1981.
- [5] 飯島泰蔵, *パターン認識理論*, 森北出版, 東京, 日本, 1989.
- [6] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. Comput. Vision*, vol.56, no.3, pp.221–255, Feb. 2004.
- [7] S. Korman, D. Reichman, G. Tsur, and S. Avidan, “FAsT-Match: Fast affine template matching,” *Int. J. Comput. Vision*, vol.121, no.1, pp.111–125, Jan. 2017.
- [8] T. Wakahara, Y. Kimura, and A. Tomono, “Affine-invariant recognition of gray-scale characters using global affine transformation correlation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.4, pp.384–395, April 2001.
- [9] Y. Yamashita and T. Wakahara, “Affine-transformation and 2D-projection invariant k -NN classification of handwritten characters via a new matching measure,” *Pattern Recognition*, vol.52, pp.459–470, April 2016.
- [10] S. Zhang, T. Wakahara, and Y. Yamashita, “Image matching using GPT correlation associated with simplified HOG patterns,” *Proceedings of 2017 Seventh International Conference on Image Processing Theory, Tools and Applications*, vol.1, pp.1–6, November–December 2017.
- [11] E. Elboher and M. Werman, “Cosine integral images for fast spatial and range filtering,” *Proceedings of 2011 18th IEEE International Conference on Image Processing*, pp.89–92, Sept. 2011.
- [12] K. Sugimoto and S.-I. Kamata, “Compressive bilateral filtering,” *IEEE Transactions on Image Processing*, vol.24, no.11, pp.3357–3369, Nov. 2015.
- [13] 杉本憲治郎, 京地清介, 鎌田清一郎, “離散コサイン変換に基づく定数時間ガウシアンフィルタの包括的性能解析,” *信学技報*, vol.117, no.48, IE2017-4, pp.19–24, May 2017.
- [14] V. Kober, “Fast algorithms for the computation of sliding discrete sinusoidal transforms,” *IEEE Transactions on Signal Processing*, vol.52, no.6, pp.1704–1710, June 2004.
- [15] Visual Geometry Group, University of Oxford, “Affine covariant features”. <http://www.robots.ox.ac.uk/~vgg/research/affine/>.
- [16] A. Alekhin, “Affine invariant feature-based image matching sample”. <https://github.com/opencv/opencv/blob/master/samples/python/asift.py>.

表1 計算時間 (second)

Image \ Method	ASIFT+RAMSAC	Ours
Graffiti	21.881	0.699
Boat	168.366	1.132