

Cascaded FAST によるキーポイント検出

長谷川昂宏† 山内悠嗣† 藤吉弘亘† 安倍満‡ 吉田悠一‡

† 中部大学 ‡ 株式会社デンソーアイティラボラトリ

E-mail: tkhr@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp

Abstract

本稿では、画像間の対応付けに不要な点の検出を抑制しながらも高速にコーナーを検出する Cascaded FAST を提案する。高速にコーナーを検出する FAST により複雑なテクスチャ(木の葉、植え込みなどが写り込んでいる領域)を含む画像に、高速なコーナー検出が可能な FAST を適用すると多くのコーナーが検出されるが、これらは、物体認識等のアプリケーションでは必要としない。これは、FAST ではわずか周囲 16 画素の情報に基づいてコーナーを定義しているためである。そこで、提案手法ではより広範囲の周囲 {20, 16, 12} 画素の輝度値の連続性とオリエンテーションの類似性によりコーナーを定義する。また、FAST のアプローチで学習した 3 つの決定木をカスケード状に並べることで、非コーナーを早期に棄却する高速なコーナー検出を実現する。さらに、提案手法ではピラミッド画像を用いることでスケールを獲得し、周囲の画素を参照する枠組みを利用することで高速にオリエンテーションを獲得する。実験により、Cascaded FAST は高速にキーポイントを検出しながらも自然領域からのキーポイント検出を大幅に抑制できたことを確認した。さらに、2 画像間の対応付けの実験結果から従来のコーナー検出法と同等の性能を保ちながら高速に 2 画像間を対応付けることを確認した。

1 はじめに

画像間の特徴点(キーポイント)の対応付け問題は、特定物体認識, 3 次元地図作成, パノラマ画像生成, 画像検索などのアプリケーションにおいて重要な課題である。画像間の対応付けは、画像中からキーポイントを検出した後、キーポイントに対して特徴量を記述し、各キーポイントの特徴量を比較することで実現できる。キーポイント検出は画像間の対応付けの前処理に相当するタスクであるため、可能な限りキーポイント検出が高速であることが望ましい。

画像間のキーポイントを対応付けるためには、見えの変化が生じた複数の画像において同じ点が同じ位置

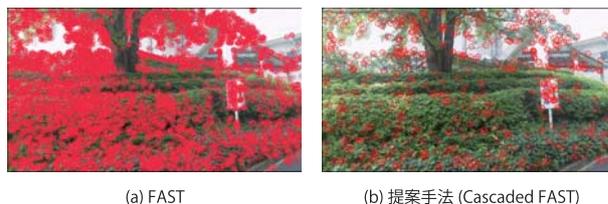


図1 自然画像におけるコーナー検出結果の比較。

で安定して検出される必要がある。そのため、現在までに多くのキーポイント検出法が提案され、これらの手法はコーナー検出法(Harris[2], FAST[8][9]), ブローブ検出法(DoG[10][11], Harris-Affine[6]), 領域検出法(MESR[5])の3つのアプローチに分類される[14]。本研究は、高速なキーポイント検出を目的とするため、コーナー検出法に分類される手法を提案する。コーナー検出は1977年に提案されたMoravecのコーナー検出法[1]以降、その研究の重要性から多くの手法[2]~[7]が提案された。近年では、高速なコーナー検出が可能なFeatures from Accelerated Segment Test (FAST)[8][9]が注目されている。FASTは注目画素と注目画素を中心とする周囲16画素の輝度値の大小関係を用いてコーナーを判定する。そして、この定義に基づいて機械学習により学習した決定木を用いることで効率的に周囲の画素を参照するため、高速にコーナーを検出できる。

しかし、FASTでは図1に示すようなテクスチャが複雑な自然領域(木の葉、植え込み等)から多くのコーナーを検出する問題がある。これは、わずか周囲16画素の輝度情報に基づいてコーナーを判定しているためである。図1(a)のように多くのコーナー点に基づき2画像間を対応付ける場合、2つの問題が発生する。1つ目の問題は、自然領域では視点の変化や、風による葉のゆらぎのような外乱の影響により見えの変化が生じやすいため、複数の画像間で同じコーナーを検出できないことがある。2つ目の問題は、2画像間のコーナーの対応付けには、1枚目から検出された1つのコーナーに対して2枚目から検出された全てのコーナーと比較する。そのため、検出されるコーナー数が多い場合には対応付けに多くの計算コストが必要となる。

そこで、本稿では画像間の対応付けに必要な点のみ

を高速に検出する Cascaded FAST を提案する. 提案手法では FAST で参照する周囲 16 画素に加え, より広範囲の画素を参照する決定木を用いる. そして, 決定木をカスケード状に並べることで高速にコーナーを検出する. そのため, 提案手法では図 1(b) に示すように自然領域からのコーナー点の検出を抑制することができる. また, 提案手法ではピラミッド画像からコーナーを検出することでスケールを得る. さらに, 周囲の画素を参照する提案手法の枠組みを利用することで, コーナー検出と同時にオリエンテーションも得る.

1.1 関連研究

コーナー検出の研究は, コーナーの位置の正確さ (Localization) と 2 画像間におけるコーナーの再現率 (Repeatability) が重要とされ [7], 現在までに多くの研究 [1]~[9] がされている. コーナー検出の初期の研究として Moravec のコーナー検出 [1] が知られている. この方法は, 注目画素を中心とする局所領域をパッチとし, コーナー点を中心とするパッチと近傍領域を中心とするパッチの見えの類似度が低いと仮定した. そして, 類似度としてパッチの差分の二乗和によって求め, 近傍領域で類似度が小さくなる点をコーナーとして検出した. Smallest Univalue Segment Assimilating Nucleus (SUSAN)[4] では, Moravec の 8 方向の輝度の変化強度しか考慮していないという欠点を円形マスクを用いることにより改善している.

一方, コーナー点はエッジの交点とも定義することができ, ある局所領域において x 方向と y 方向の 2 つの異なる方向のエッジが共に強いとき, その局所領域の中心はコーナーと見なせる. Harris[2] らは, このようなコーナー点を検出するために x , y , xy 方向の局所的な勾配を要素に持つヘッセ行列の固有値の関係からコーナーを検出する. さらに, Harris のコーナー検出をベースに, スケール変化やアフィン変形に対して頑健なコーナーを検出する Harris-Affine 検出器 [6] も提案されている.

以上のコーナー検出法は, 冒頭で述べた位置の正確さと再現性が高い一方で, 高速にコーナーを検出できない課題は長らく解決されなかった. この課題にたいして Rosten[8][9] らは, 機械学習を導入することでコーナーを高速に検出する Features from Accelerated Segment Test (FAST) を提案した. FAST では注目画素と注目画素を中心とする周囲 16 画素を比較し, 周囲 16 画素の輝度値が注目画素より明るい, または暗い画素が連続する場合に注目画素をコーナーとして定義する. そして, このコーナーの定義に従い ID3 のアルゴリズムにより学習した決定木をコーナー検出器とする. FAST は検出速度と性能が優れていることから, FAST をベースに改良した手法も幾つか提案されている [12][13]. しかしながら, FAST は注目画素周辺のわずか 16 画素に

基づいてコーナーを定義しているため, テクスチャが複雑な領域から多数のコーナーを検出する問題がある.

また, 多くのコーナー検出法では, 検出したコーナーの座標のみを出力する. しかし, 画像のスケール変化や回転が加わった場合には, 2 画像間の対応付けはできない. そこで, [10][11] では, 画像にスケール変化や回転を加えた場合においても同一の特徴量を得るために, コーナーの座標以外に基準となるスケールとオリエンテーションを獲得している. 本稿では, 座標とスケール, オリエンテーションを持つ点をキーポイントと呼ぶ.

1.2 提案手法の概要

提案手法は下記の 3 つの特徴を持つ.

- 複雑なテクスチャ領域からのコーナー検出の抑制
FAST では注目画素を中心としたわずか周囲 16 画素の輝度情報のみを用いてコーナーを定義するため, 複雑なテクスチャを持つ領域から多数のコーナーを検出する. この問題を解決するために, 提案手法ではより広範囲の画素を参照することでコーナーを定義する.
- 高速なコーナー検出
周囲 {20, 16, 12} 画素を観測する 3 つの決定木を学習し, カスケード状に並べる. これにより, 非コーナー点を早期に棄却できるため, 高速なコーナー検出が実現できる.
- スケールとオリエンテーションの獲得
FAST では, コーナーの座標のみを出力する. 一方, 提案手法は, ピラミッド画像からコーナーを検出することでスケールを得る. さらに, 周囲の画素を参照する提案手法の枠組みを利用することで, 高速にオリエンテーションを得る. これにより, 画像のスケール変化や回転に対して頑健なキーポイントとなる.

2 Features from Accelerated Segment Test (FAST) と問題点

機械学習を導入することで高速にコーナーを検出する FAST について述べる.

2.1 FAST によるコーナー検出

FAST は高速にコーナーを検出するために, 機械学習により学習した決定木を用いてコーナーを検出する.

まず, 決定木の学習に必要なコーナーと非コーナーのラベルを定義する. 学習画像の全ての画素 (学習サンプル) において式 (1) のように, 注目画素 p を中心とする周囲 16 画素を明るい (Brighter), 類似 (Similar), 暗

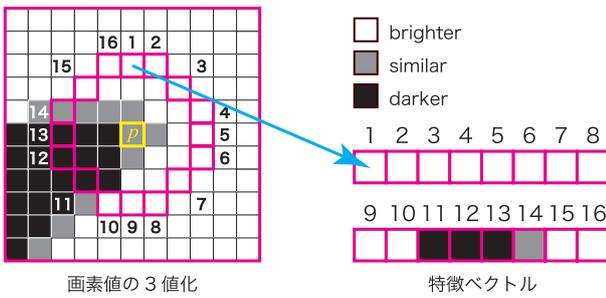


図2 注目画素 p の特徴ベクトル.

い (Darker) の3値に分類する.

$$S_{p \rightarrow x} = \begin{cases} \text{Brighter} & I_p + t \leq I_{p \rightarrow x} \\ \text{Similar} & I_p - t < I_{p \rightarrow x} < I_p + t \\ \text{Darker} & I_{p \rightarrow x} \leq I_p - t \end{cases} \quad (1)$$

ここで, I_p は注目画素の輝度値, $x \in \{1, \dots, 16\}$ は円周上の画素の位置, $I_{p \rightarrow x}$ は周囲16画素の輝度値, t はしきい値を表す. このように3値化した周囲16画素を図2に示すように特徴ベクトルとする. そして, 各注目画素 p のラベルは周囲16画素のうち n 画素以上連続して Brighter もしくは Darker のときコーナー c とし, 条件を満たさない場合は非コーナー \bar{c} と定義する. 次に, 3値化した周囲16画素と注目画素のコーナーのラベルを用いて ID3[15] のアルゴリズムに従って決定木を学習する. 分岐ノードでは円周上の画素の値 $S_{p \rightarrow x}$ を観測し, 式(2)で求められる情報利得 G が最も高い円周上の画素 x を用いて分岐ノードに辿り着いた学習サンプルを分岐する.

$$G = H(P_a) - H(P_b) - H(P_s) - H(P_d) \quad (2)$$

ここで, P_a はノードに辿り着いた画素, P_b は $S_{p \rightarrow x}$ が Brighter と判定された画素, P_s は $S_{p \rightarrow x}$ が Similar と判定された画素, P_d は $S_{p \rightarrow x}$ が Darker と判定された画素である. H はエントロピーを表し, 式(3)より求められる.

$$H(P) = (C + \bar{C}) \log_2(C + \bar{C}) - C \log_2 C - \bar{C} \log_2 \bar{C} \quad (3)$$

ここで, C はコーナー数, \bar{C} は非コーナー数を示す. この処理をコーナーと非コーナーの分類ができなくなるまで, すなわち情報利得が0になるまでノードを分岐する. 情報利得が0となったときのノードを末端ノードとし, 辿り着いたコーナーのラベルを記録する.

決定木によりコーナーを検出する際には, 図3に示すように機械学習により学習した決定木へ注目画素 p を入力し, トラバースする. そして, 到達した末端ノードに記録されたラベル情報によりコーナーもしくは非コーナーと判定する. 決定木を用いることで周囲16画素のうち平均で2.26画素を観測するだけでコーナーを

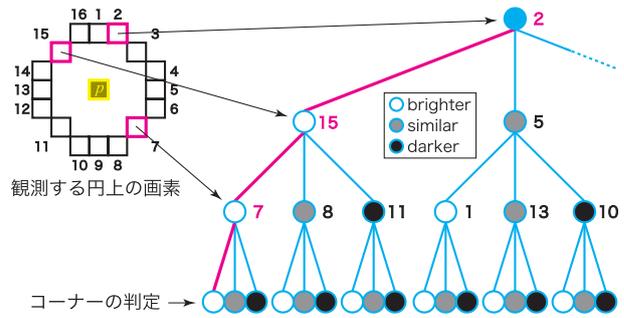


図3 決定木を用いたコーナー検出.

検出することができるため, 高速なコーナー検出が可能である.

2.2 FASTの問題点

高速にコーナーを検出できる一方, 図1のようなテクスチャが複雑な自然領域において多数のコーナーを検出する問題がある. このような点は視点の変化や風による葉の揺らぎにより見えが容易に変化するため, 複数の画像間で同じコーナーを検出できないことがある. また, 多数のコーナーを検出すると2画像間の対応付けに必要な特徴量記述と距離計算の計算コストが大きくなる問題がある. そのため, テクスチャが複雑な自然領域から検出されるコーナーらしくない点を抑制する手法が必要である.

2.3 コーナー点の傾向調査

FASTによって検出されるコーナーらしい点とコーナーらしくない点の傾向を調査する. まず, コーナー周辺領域の見えの傾向調査をする. 見えの傾向調査では, コーナーとして検出された点を中心とするパッチ画像を生成する. このパッチ画像の見えをコーナーらしい点とコーナーらしくない点において比較する. 図4は, 人工物画像と自然画像からFASTにより検出したコーナーを示す. 図4(a)のコーナーらしい点は, どのパッチの見えも類似している傾向が確認できる. 一方, 図4(b)のコーナーらしくない点は, 見えのばらつきが大きい傾向がある.

図4のコーナーの見えの傾向から, コーナーらしい点では周囲16画素の外側と内側の周囲の画素も同じような輝度の変化であることが予想できる. そこで, コーナーを中心とする周囲の画素と注目画素の輝度の差分値の傾向を定量的に解析する. 解析対象である周囲の画素を図5に示す. 解析には人工物のみを含む画像と自然領域のみを含む画像から検出された1,000点のコーナーを使用し, 差分値の平均を解析する. その際, n 画素以上の Brighter もしくは Darker が連続する画素の始点を基準点とし, 注目画素と基準点を結ぶ直線と基準線(角度 0°) とする. 図6に解析結果を示す. 図6のグラフの縦軸は注目画素と周囲の画素の差分の絶対値,

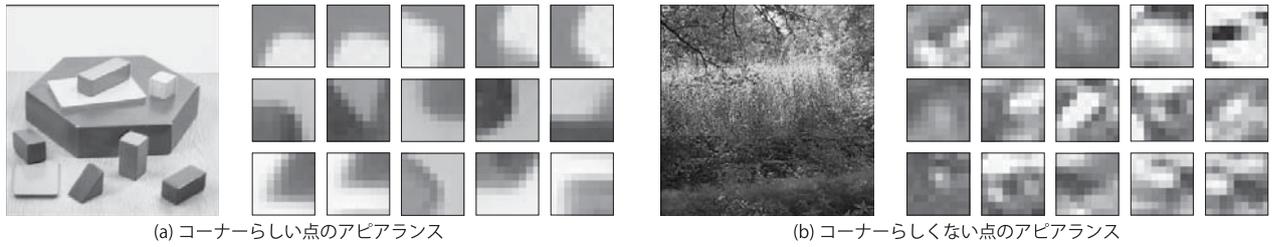


図4 FASTにより検出されたコーナーの見えの違い.

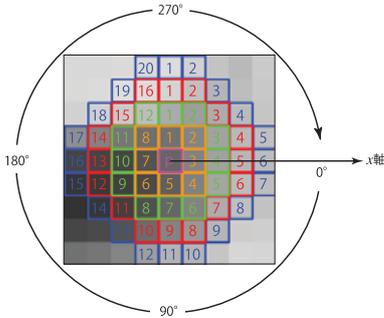


図5 調査対象の領域.

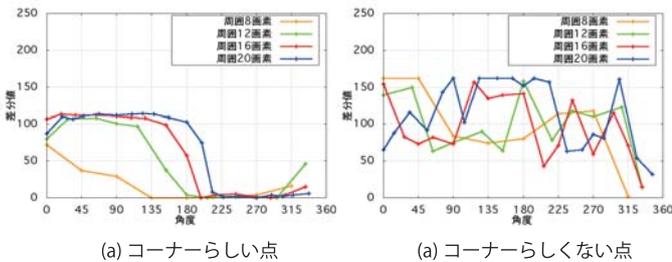


図6 コーナー点の解析結果.

横軸は角度を表わす. コーナーらしい点の周囲 {20, 16, 12} 画素の差分値は大きな値が連続し, グラフの形状が類似していることが確認できる. コーナーらしくない点では周囲 {20, 16, 12, 8} 画素の差分値にばらつきがある. この解析結果から, コーナーらしい点のみを検出するには周囲 {20, 16, 12} 画素の情報に基づきコーナーを定義することが有効であると考えられる.

3 Cascaded FASTによるキーポイント検出

2.3の調査結果から, コーナーらしい点の周囲 {20, 16, 12} 画素において差分値の変化の傾向が似ていることを確認した. そこで, 本研究では周囲 {20, 16, 12} 画素の情報に基づいてコーナーを検出する. また, Cascaded FASTではコーナーの座標に加え, スケールとオリエンテーションも求める. 以下に詳細を述べる.

3.1 コーナーの定義

Cascaded FASTでは周囲 {20, 16, 12} 画素のBrighterまたはDarkerが連続しているかという条件に基づきコーナー候補点を検出する. そして, 検出し

たコーナー候補点に対してオリエンテーションを算出し, 周囲 {20, 16, 12} 画素のオリエンテーションが類似している場合に注目画素をコーナーとして検出する. 以下に各処理の詳細について述べる.

Step1: BrighterまたはDarkerの連続性による条件

周囲 {20, 16, 12} 画素を式(1)によりBrighter, Similar, Darkerの3値に分類する. FASTでは周囲16画素においてBrighterまたはDarkerが9画素以上連続した場合に注目画素をコーナーとするが, 提案手法では周囲 {20, 16, 12} 画素がそれぞれ {11, 9, 6} 画素以上のBrighterまたはDarkerが連続している場合に注目画素をコーナー候補点とする. 周囲 {20, 12} 画素のBrighterまたはDarkerの連続する画素数は, ベースとなるFASTの周囲16画素の比率に合わせて決定した.

Step2: オリエンテーションの算出

Step1で求めたコーナー候補点に対して周囲 {20, 16, 12} 画素のそれぞれのオリエンテーションを算出する. 図7に周囲16画素のオリエンテーションの算出する例を示す. まず, BrighterまたはDarkerが連続する画素の始点から終点までの角度を求める. x 軸に対する注目画素 $p(u_p, v_p)$ と始点の画素 $x_s(u_s, v_s)$ の角度を θ_s , x 軸に対する注目画素 $p(u_p, v_p)$ と終点の画素 $x_e(u_e, v_e)$ の角度を θ_e とすると, 始点と終点の角度 $\theta_{s \rightarrow e}$ は次式により求められる.

$$\theta_{s \rightarrow e} = \begin{cases} 360 - |\theta_s - \theta_e| & \text{If } \theta_s > \theta_e \\ |\theta_s - \theta_e| & \text{Otherwise} \end{cases} \quad (4)$$

$$\theta_s = \text{angle}(x_s, p) \quad (5)$$

$$\theta_e = \text{angle}(x_e, p) \quad (6)$$

ここで, $\text{angle}(\cdot)$ は x 軸を基準とした始点または終点の角度を返す関数である. そして, 始点と終点の角度を2等分する方向をオリエンテーション θ として次式より算出する.

$$\theta = \frac{\theta_{s \rightarrow e}}{2} + \theta_s \quad (7)$$

周囲 {20, 12} 画素のオリエンテーションも図7と同様に算出する.

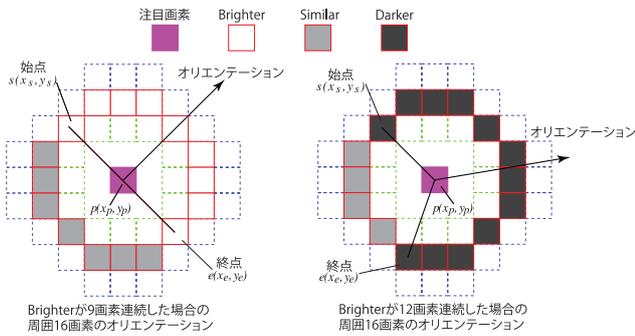


図7 周囲16画素におけるオリエンテーション算出例.

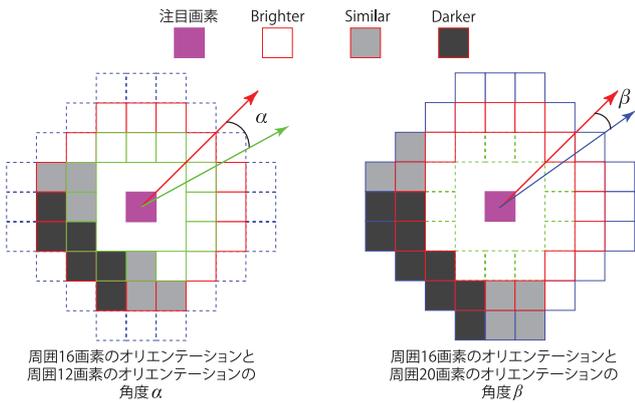


図8 角度差 α と β の算出例.

Step3 : オリエンテーションの類似性による条件

コーナー候補点に対して, 周囲 {20, 16, 12} 画素のオリエンテーションの類似性によりコーナーと非コーナーを判定する. 図8に示すように周囲16画素のオリエンテーションと周囲12画素のオリエンテーションの角度差を α , 周囲16画素のオリエンテーションと周囲20画素のオリエンテーションの角度差を β として求める. そして α と β が式(8)を満たす場合にコーナー候補点である注目画素 $p(u_p, v_p)$ をコーナーとして検出する.

$$p(u_p, v_p) = \begin{cases} c & \text{If } \alpha \leq Th_1 \ \& \ \beta \leq Th_2 \\ \bar{c} & \text{Otherwise} \end{cases} \quad (8)$$

ここで, Th_1 と Th_2 はしきい値を表わす. 周囲20画素のオリエンテーションと周囲12画素のオリエンテーションはそれぞれ分解能が異なるため, α と β に対して別々のしきい値を用いる.

3.2 機械学習による決定木の学習

提案手法においても FAST と同様に機械学習を導入することで高速なコーナー検出が可能である. 提案手法では, まず3.1により学習サンプルをコーナーと非コーナーに分類する. そして, 分類した学習サンプル

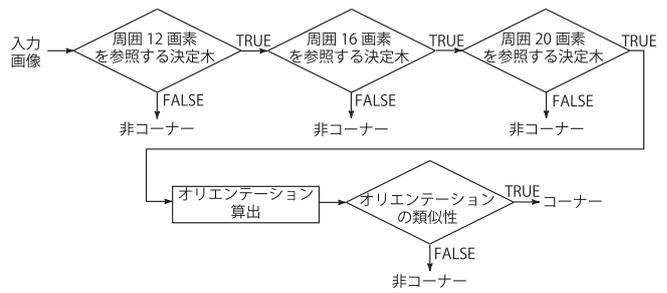


図9 Cascaded FAST によるコーナー検出の流れ.

を ID3 のアルゴリズムに基づき, それぞれ周囲 {20, 16, 12} 画素を参照する3つ決定木を学習する.

3.3 カスケード化による高速化

提案手法ではある画素を入力した際に, 3.2で学習した3つの決定木が全てコーナーと判定した場合にコーナー候補点と出力する. もし, 一つの決定木でも非コーナーと判定した場合には非コーナーとして出力する. これをより高速に処理するために, それぞれの決定木をカスケードに配置する. これにより, 各決定木で非コーナーとして判定された場合は早期に棄却できるため, より高速にコーナー候補点を検出することが可能となる. 図9に決定木のカスケード化を示す. そして, 検出されたコーナー候補点に対して Step2 と同様にオリエンテーションを算出し, Step3 にて述べたオリエンテーションの類似性を測ることでコーナーを判定する. 周囲 {20, 16, 12} 画素を参照する決定木の配置順は, コーナー検出結果に一切影響を与えないが, 検出時間に違いが表れることを事前実験により確認した. 図9に示すように, 周囲 {12, 16, 20} 画素を参照する決定木の配置順が最も高速にコーナーを検出できることを確認した.

3.4 オリエンテーションとスケールの獲得

ここまでに述べた手法では, コーナー点の座標のみを出力するコーナー検出法である. 提案する Cascaded FAST は, コーナー点の座標に加え, スケールとオリエンテーションも出力することが可能である. スケールについては, 複数の解像度で表現される画像ピラミッドから Cascaded FAST によりコーナー点を検出することで, 検出した際に入力された画像解像度のスケールを利用する. オリエンテーションについては, 3.1の Step2 により算出されるオリエンテーションを出力する.

図10は, 画像を $1^\circ \sim 359^\circ$ まで回転させたときのオリエンテーションの誤差を表わした結果である. 例えば, 画像を 45° 回転させた場合, 元画像と回転画像で対応する点のオリエンテーションの角度を差分し, 差分値が

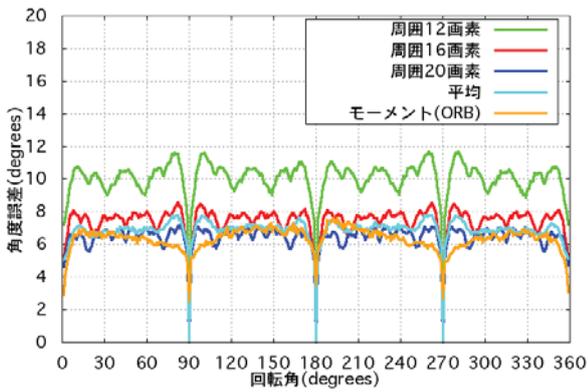


図 10 オリエンテーションの誤差.

45° であるとき誤差は 0 となる. この方法で周囲 {20, 16, 12} 画素のオリエンテーション, 周囲 {20, 16, 12} 画素のオリエンテーションの平均, ORB で用いられるモーメントに基づいて算出したオリエンテーション [17] を比較した. まず, 提案手法である Cascaded FAST の決定木毎のオリエンテーションの誤差を見ると, 周囲 20 画素を参照する決定木のオリエンテーションの誤差が小さいことが確認できる. 提案手法では, 周囲 20 画素を参照する決定木のオリエンテーションを採用する. また, モーメントに基づくオリエンテーションと周囲 20 画素を参照する決定木のオリエンテーションの精度が同等であることがわかる. 図 11 に Cascaded FAST により検出されたキーポイントを示す. 赤色の円の中心はコーナーの座標, 円の大きさはスケールの大きさ, 青色の線はオリエンテーションを表わす. 図 11 から画像を回転させた場合, 検出されたコーナーのオリエンテーションも同様に回転していることが確認できる.

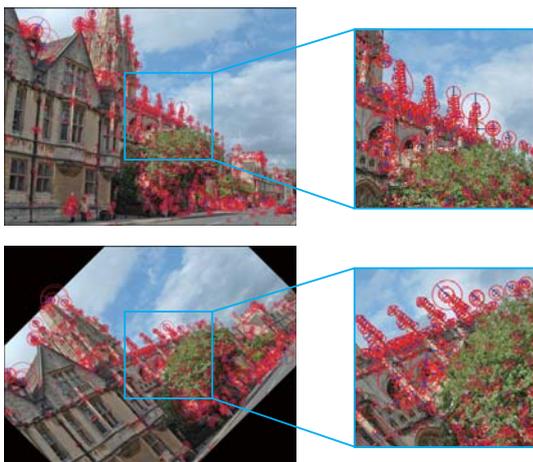


図 11 Cascaded FAST によるキーポイント検出結果の例.

4 評価実験

Cascaded FAST の有効性を確認するために評価実験を行う. 本章では, コーナー検出の評価と 2 画像間の特徴点の対応付けの評価に分けて実験する.

4.1 コーナー検出の評価実験の概要

ここでは, F 値によるコーナー検出法の評価とコーナーの検出時間を評価する. F 値による評価では, Harris のコーナー検出で得られた点を真のコーナーとし, FAST と Cascaded FAST で検出されたコーナー点と真のコーナーが同一の座標であるかを検証する. Harris のコーナー検出によって得られるコーナーはコーナーらしさをレスポンス値として出力できる. このレスポンス値を適切なしきい値で処理することにより, コーナーらしい点を検出し, コーナーらしくない点の検出を抑制できる. そして, FAST と Cascaded FAST の各手法における F 値を比較する.

評価実験にはデータセットとして Oxford Buildings Dataset [18] を使用する. 本データセットの中から人工物と自然領域を含む画像 1,000 枚を使用する.

コーナーの検出時間の評価実験では, Harris のコーナー検出法, FAST, Cascaded FAST のコーナー検出時間を比較する. 本実験では, CPU: Intel(R) Xeon(R) X5470 3.33GHz, メモリ: 32GB の PC を用いる.

4.1.1 F 値による評価

図 12 に各手法における F 値の結果を示す. グラフの縦軸に F 値を示し, 横軸は Harris のコーナー検出により得られるコーナー数を表わす. また, 図中の赤線は, Harris コーナー検出で得られるスコアを適切なしきい値によって処理した際に得られるコーナー数 (950 個) である. 図 12 から Cascaded FAST は FAST より高い F 値であることが確認できる.

図 13 に各手法のコーナー検出結果を示す. FAST では自然領域から多くのコーナーを検出しているが, Cascaded FAST では自然領域からの検出しているコーナーが少ないことが確認できる.

4.1.2 コーナーの検出時間の評価

Harris のコーナー検出法と FAST, Cascaded FAST のコーナー検出時間を比較する. 各手法の処理速度を表 1 に示す. Cascaded FAST の処理時間は FAST に比べて 2.9 [ms] 増加するが, Harris と比較して約 11 倍高速にコーナーを検出できる. また, Cascaded FAST は約 135 [fps] で動作可能である.

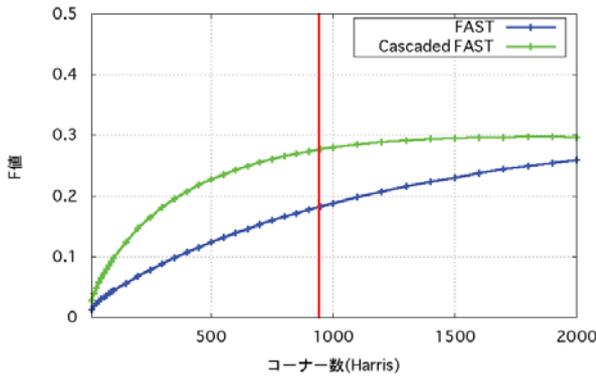


図 12 FAST と Cascaded FAST の F 値の算出結果.



図 13 各手法のコーナー検出結果.

表 1 各手法の検出時間の比較.

	Harris	FAST	Cascaded FAST
検出時間 [ms]	81.1	4.5	7.4
コーナー数	82	7,656	955

4.2 2 画像間の対応付けの評価実験

ここでは、2 画像間のキーポイントのマッチング性能と時間を評価する。比較手法を表 2 に示す。コーナー検出法としては、Harris のコーナー検出と FAST、Cascaded FAST の 3 種類である。スケールの算出には、どの手法においても画像ピラミッドを用いる。オリエンテーションの算出は、[17] で用いられるコーナー点を中心とするパッチ内のモーメントに基づく手法と Cascaded FAST の 2 種類である。特徴量は、回転とスケールの変化を正規化した ORB 特徴量 [17] を用いる。ORB 特徴量はパッチ内の輝度の大小関係を 0 と 1 のバイナリコードで表現した特徴量である。

本研究では、2 画像間の対応付けにアフィン変換によ

り変形した画像を用いる。これにより変換前後の座標位置の同定が容易にできる。

2 画像間のキーポイントを対応付けるためには、1 枚目の画像から得られる 1 つのキーポイントと 2 枚目の画像から得られる全てのキーポイントとの距離を計算し、最も距離が小さい点 p_1 と 2 番目に距離が小さい点 p_2 を求める。そして式 (9) が成立する場合にキーポイントを対応付ける。

$$D(p_1) < D(p_2)T \quad (9)$$

ここで、 $D(\cdot)$ は距離関数、 T はしきい値を表す。また、本実験ではバイナリコードで表現される特徴量を用いるため、ハミング距離により高速な距離計算が可能である。式 (9) が成立する場合、2 点の座標が同一であるかを判定する。座標が $\sqrt{(1+1)}$ 画素以内の位置ずれの場合に正解とする。

キーポイントの対応付け性能の評価は、マッチング率 = (正解数/対応付け数) を用いる。そして、マッチング率と 1 秒間あたりの 2 画像間の対応付けのフレームレート (fps) を比較する。

4.2.1 キーポイントマッチングにおける性能と速度

各手法のマッチングの性能と速度の結果を図 14 に示す。提案する Cascaded FAST は他の手法と比較して、マッチング率はほぼ同等で、フレームレートは高いことが確認できる。図 15 に 2 画像間の対応付けの各処理の計算時間の内訳を示す。Harris のコーナー検出は、コーナー検出の処理時間が占める割合が非常に大きい。一方、FAST はコーナー検出とオリエンテーション算出を高速に処理できるが、特徴量記述と距離計算に多大な計算時間を要する。この理由として、FAST は他の手法と比べて非常に多くのコーナーを検出するためである。提案手法である Cascaded FAST は、他の 2 手法と比べて高速に各処理を実行できていることがわかる。特に、周囲 20 画素の情報を用いてオリエンテーションを算出する場合は、コーナー検出時に計算したオリエンテーションを再利用することができるため、この処理を省くことができる。周囲 20 画素の情報を用いてオリエンテーションを用いた Cascaded FAST は、2 画像間の対応付けを約 43.7[ms] で処理することが可能である。

5 おわりに

本稿では自然領域から画像間の対応付けに不要な点を抑制する Cascaded FAST を提案した。提案手法では、広範囲の周囲 {20, 16, 12} 画素の輝度値の連続性とオリエンテーションの類似性によりコーナーを定義することで自然領域からのコーナー検出を抑制した。ま

表 2 各手法のオリエンテーション, スケール, 特徴量

手法	オリエンテーション算出	スケール取得	特徴量記述
Harris	モーメント	画像ピラミッド	ORB
FAST	モーメント	画像ピラミッド	ORB
Cascaded FAST(モーメント)	モーメント	画像ピラミッド	ORB
Cascaded FAST(周囲 20 画素)	Cascaded FAST	画像ピラミッド	ORB

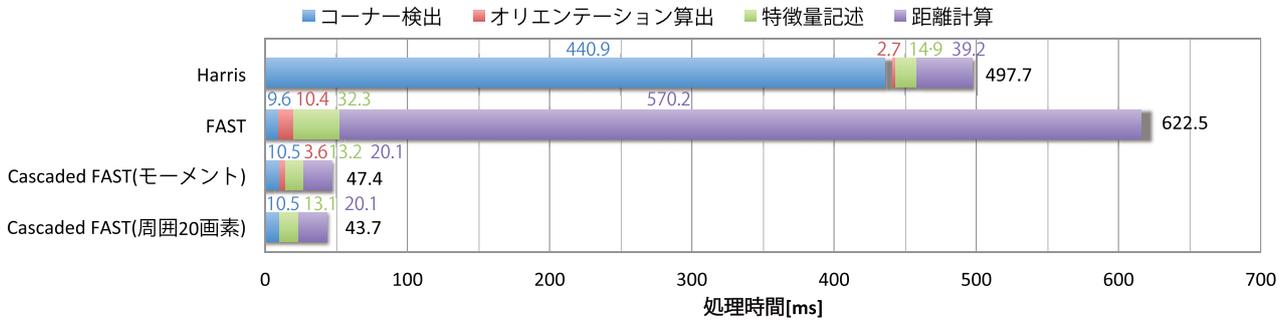


図 15 2 画像間の対応付けの各処理の計算時間の内訳

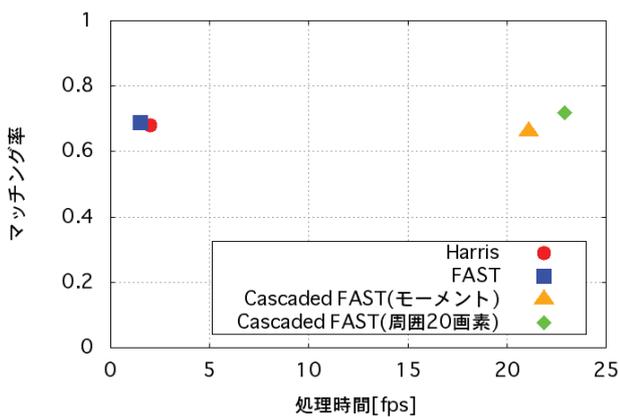


図 14 マッチングの性能と速度

た, FAST のアプローチで学習した 3 つの決定木をカスケード状に並べて非コーナーを早期に棄却することで高速なコーナー検出を実現した. さらに, 提案手法ではピラミッド画像を用いることでスケールを獲得し, 周囲の画素を参照する枠組みを利用することで高速にオリエンテーションを獲得することができ, 対応点マッチングのキーポイントとして有効である. 今後の課題は, 画像ピラミッドから得られるスケールを効率的に獲得することである.

参考文献

[1] H. Moravec, "Towards Automatic Visual Obstacle Avoidance", International Joint Conference on Artificial Intelligence, pp. 584, 1977.
 [2] C. Harris, and M. Stephens, "A Combined Corner and Edge Detector", Alvey Vision Conference, pp. 147-151, 1988.
 [3] J. Shi and C. Tomasi, "Good features to track", Computer Vision and Pattern Recognition, 1994.

[4] S. M. Smith and J. M. Brady, "SUSAN - a new approach to low level image processing", International Journal of Computer Vision, Vol. 23, No. 1, pp. 45 - 78, 1997.
 [5] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", British Machine Vision Conference, pp. 384 - 393, 2002.
 [6] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors", International Journal of Computer Vision, Vol. 1, No. 60, pp. 63 - 86, 2004.
 [7] J. P. Gravel, "Corner Detection", Biological Cybernetics, Vol. 59, No. 4-5, pp. 139 - 153, 1988.
 [8] E. Rosten and T. Drummond, "Machine Learning for High-speed Corner Detection", European Conference on Computer Vision, pp. 430 - 443, 2006.
 [9] E. Rosten, R. Porter, T. Drummond, "Faster and Better: A Machine Learning Approach To Corner Detection", Pattern Analysis and Machine Intelligence, Vol. 32, No. 1, pp. 105 - 119, 2010.
 [10] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", International Conference on Computer Vision, pp. 1150 - 1157, 1999.
 [11] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, No. 2, pp. 91 - 110, 2004.
 [12] E. Mair, G. D. Hager, D. Burschka, M. Suppa and G. Hizinger, "Adaptive and Generic Corner Detection Based on the Accelerated, Segment Test", European Conference on Computer Vision, 2010.
 [13] Cronje, "BFROST : binary features from robust orientation segment tests accelerated on the GPU", Pattern Recognition Association of South Africa, 2011.
 [14] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors: A Survey", Foundations and Trends in Computer Graphics and Vision, Vol. 3, No. 3, 2008.
 [15] J. R. Quinlan, "Induction of Decision Trees", Machine Learning pp. 81 - 106, 1986.
 [16] M. Calonder, V. Lepetit and C. Strecha and P. Fua, "BRIEF : Binary Robust Independent Elementary Features", European Conference on Computer Vision 2010.
 [17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB : An Efficient Alternative to SIFT or SURF", International Conference on Computer Vision, 2011.
 [18] The Oxford Buildings Dataset, <http://www.robots.ox.ac.uk/vgg/data/oxbuildings/>.