

平成29年度  
中部大学大学院工学研究科情報工学専攻

博士学位論文

局所特徴量の因子分解表現によるキーポイント  
マッチングの高精度化に関する研究

長谷川 昂宏



# 論文要旨

キーポイントマッチングとは、異なる視点から撮影された複数の画像間で、物理的に同一の位置(対応点)を探索する処理であり、物体認識、画像検索、3次元復元などの基礎技術となるため精力的に研究されている。物体認識や画像検索などのアプリケーションにおいて、キーポイントマッチングは低スペックなハードウェア上での高速な動作が必要である。キーポイントマッチングは、(1) キーポイント検出、(2) 局所特徴量記述、(3) 対応点探索、の3つの処理で構成され、それぞれの処理で高速化や高精度化に関する研究が取り組まれている。キーポイントマッチングにおいて、処理時間が低下する原因の1つは、キーポイントを過剰に検出することである。キーポイントとは、画像の局所領域における勾配や輝度情報に基づいて検出されるユニークな点である。入力画像のテクスチャによっては、マッチングに有効でないキーポイントを過剰に検出してしまい、後段処理である局所特徴量記述や対応点探索で計算コストが増加する。

そこで、本研究では不必要なキーポイントの検出を抑制した高速なキーポイントマッチングについて取り組む。キーポイント検出は画像の局所領域において勾配や輝度情報によりキーポイントを検出するため、テクスチャが複雑な領域が多ければ大量のキーポイントが検出されやすい。この問題を解決するために、まず過剰に検出される不必要なキーポイントとマッチングに有効なキーポイントの周辺領域画像を解析する。キーポイント周辺領域を解析することで、不必要なキーポイントが持つ傾向を捉えて有効なキーポイントのみを検出する。さらに、キーポイント検出を決定木のアルゴリズムで解くことで、高速なキーポイント検出を実現することができる。

3次元復元などのアプリケーションでは、キーポイントマッチングは強い視点変化を伴う画像に対しても頑健に対応点を求めることが重要である。視点変化を持つ画像を対応づけるには、キーポイントに対してアフィン不変な楕円領域(アフィン領域)を推定する。推定されたアフィン領域内のみ画像情報から局所特徴量を記述することで、視点変化を持つ画像間の対応点を求めることができる。従来のアフィン領域推定方法は、キーポイントの位置ずれや照明変化等により不正確なアフィン領域を推定する問題がある。この問題については、キーポイントに対して複数候補のアフィン領域を推定することで、高精度なキーポイントマッチングを実現する。キーポイントの複数のアフィン領域推定は、様々な楕円形状の非等方性2次微分フィルタを畳み込み、その応答値を全探索することで複数候補の楕円形状を推定する。この処理は非常に計算コストが高いため、大量の2次微分フィルタを因子分解により低ランクな基底フィルタと重み係数の線形結合により近似する。このようにして、検出されたキーポイントから複数のアフィン領域を効率的に推定することが可能となる。さらに、因子分解による低ランク近似の枠組みを局所特徴量記述にも適用させる。特徴量記述の場合、視点変化に頑健な特徴量を記述するために、入力画像に様々なアフィン変換を施した後に特徴量を抽出する。局所特徴量が画像とフィルタの畳み込みで計算可能な場合、フィルタ側をアフィン変換させ、因子分解により低ランク近似が可能となる。よって、視点変化を考慮した特徴量記述で最も計算コストが高い画像のアフィン変換を事前に計算することができるため効率的である。さらに、アフィン変換によって得られた特徴量群を部分空間に射影することで、より高精度な特徴量を記述する。

最後に、特徴量マッチングの考え方を物流ロボットの物体認識へ応用した手法について述べる。



# 目次

<b>第 1 章 序論</b>	<b>1</b>
1.1 研究の背景	2
1.2 研究目的	3
1.3 本論文の構成	4
<b>第 2 章 キーポイントマッチングのための局所特徴量とその関連研究</b>	<b>7</b>
2.1 キーポイントマッチングについて	8
2.2 キーポイント検出	10
2.2.1 Harris コーナー検出	10
2.2.2 Hessian 検出器	12
2.2.3 Features from Accelerated Segment Test (FAST)	13
2.3 スケールスペースを用いたキーポイント検出	16
2.3.1 Harris-Laplace と Hessian-Laplace	17
2.3.2 Scale-Invariant Feature Transform (SIFT) Detector	17
2.3.3 Speeded-Up Robust Features (SURF) Detector	22
2.3.4 Oriented FAST and Rotated BRIEF (ORB) Detector	25
2.3.5 Spectral SIFT	26
2.4 アフィン領域の推定	30
2.4.1 Harris-Affine と Hessian-Affine	30
2.4.2 Maximally Stable Extremal Regions (MSER)	33
2.5 実数ベクトルによる特徴量記述	33
2.5.1 Scale-Invariant Feature Transform (SIFT) Descriptor	34
2.5.2 Speeded-Up Robust Features (SURF) Descriptor	35
2.5.3 PCA-SIFT	35
2.5.4 Gradient Location and Orientation Histogram (GLOH)	36
2.5.5 Root SIFT	37
2.6 2 値ベクトルによる特徴量記述	37
2.6.1 Binary Robust Independent Elementary Features (BRIEF)	38
2.6.2 Binary Robust Invariant Scalable Keypoints (BRISK)	38
2.6.3 Oriented FAST and Rotated BRIEF (ORB) Descriptor	39

2.6.4	Fast Retina Keypoint (FREAK)	41
2.6.5	Binary Online Learned Descriptor (BOLD)	42
2.6.6	Discriminative BRIEF (D-BRIEF)	43
2.6.7	Bin Boost	45
2.7	視点合成に基づいた多視点特徴量記述	47
2.7.1	Affine SIFT (ASIFT)	47
2.7.2	Affine Subspace Representation (ASR)	49
2.8	まとめ	52
<b>第 3 章</b>	<b>Cascaded FAST によるキーポイント検出</b>	<b>54</b>
3.1	FAST で検出されるキーポイントの傾向調査	56
3.2	キーポイントの検出方法	57
3.2.1	コーナーの定義	58
3.2.2	機械学習による決定木の学習	60
3.2.3	カスケード構造の決定木による高速化	61
3.2.4	スケールとオリエンテーションの獲得	62
3.3	評価実験	63
3.3.1	コーナー検出時間の評価	64
3.3.2	F 値による評価	64
3.3.3	キーポイントマッチングにおける精度と速度	66
3.4	まとめ	68
<b>第 4 章</b>	<b>非等方性 LoG フィルタによる複数のアフィン領域の推定</b>	<b>69</b>
4.1	複数のアフィン領域推定	71
4.1.1	非等方性 LoG スケールスペースの近似	71
4.1.2	非等方性 LoG フィルタの応答値の算出	72
4.1.3	固有関数の連続関数フィッティング	74
4.1.4	複数のアフィン領域の探索	76
4.1.5	単純楕円パターンによるテスト	77
4.2	評価実験	79
4.2.1	データセット	79
4.2.2	Repeatability による評価方法	80
4.2.3	Repeatability による実験結果	81
4.2.4	画像検索タスクにおける認識率	82
4.2.5	処理時間の比較	85
4.2.6	複数のアフィン領域推定の閾値	85
4.3	まとめ	86

<b>第 5 章</b>	<b>因子分解に基づく多視点特徴量と特徴量間距離の下界算出による対応点探索の効率化</b>	<b>87</b>
5.1	因子分解に基づく多視点特徴量	89
5.1.1	線形モデルによる多視点特徴量	89
5.1.2	特徴量記述フィルタの視点合成	90
5.1.3	特徴量記述フィルタのコンパクト化	90
5.1.4	固有関数の連続関数フィッティング	91
5.1.5	連続アフィンパラメータによる多視点特徴量の生成	92
5.1.6	特徴量間距離の下界計算による対応点探索の効率化	94
5.2	評価実験	96
5.2.1	データセット	96
5.2.2	固有フィルタ数 $N_f$ における提案手法の性能	96
5.2.3	上位 $N_l$ 個の下界を用いた提案手法の性能	97
5.2.4	キーポイントマッチング性能の比較実験	97
5.2.5	処理時間	98
5.2.6	まとめ	98
<b>第 6 章</b>	<b>因子分解に基づく多視点特徴量と部分空間表現</b>	<b>99</b>
6.1	多視点特徴量の部分空間表現	100
6.2	勾配方向ヒストグラムモデルへの拡張	101
6.3	評価実験	103
6.3.1	PCA の基底数 $N_p, N_s$ における提案手法の性能	104
6.3.2	固有フィルタ数 $N_f$ における提案手法の性能	104
6.3.3	従来の多視点特徴量記述子との比較	105
6.3.4	HPatches benchmark での評価	107
6.3.5	処理時間の比較	109
6.3.6	まとめ	111
<b>第 7 章</b>	<b>物流ロボットシステムにおける特徴量マッチングを用いた物体認識</b>	<b>112</b>
7.1	ピッキングロボットのための物体認識	113
7.2	把持位置に基づくマルチクラス物体認識	115
7.2.1	把持位置に基づく Convolutional Neural Network の構築	115
7.2.2	学習画像	116
7.2.3	制約付き softmax	117
7.2.4	特徴量マッチングによる未学習物体の認識	119
7.3	評価実験	119
7.3.1	データセット	119
7.3.2	物体認識における精度	120
7.3.3	把持位置検出における精度	123

7.3.4	特徴量マッチングによる認識精度 . . . . .	124
7.3.5	処理時間 . . . . .	124
7.3.6	まとめ . . . . .	126
<b>第 8 章</b>	<b>結論と展望</b>	<b>128</b>
8.1	結論 . . . . .	128
8.2	展望 . . . . .	129
<b>謝 辞</b>		<b>131</b>
<b>参考文献</b>		<b>133</b>
<b>研究業績一覧</b>		<b>140</b>



# 目次

1.1	本論文の構成.	5
2.1	ガウシアンフィルタの1次微分フィルタ.	10
2.2	コーナー, エッジ, フラット領域の微分値の分布と固有値の関係.	11
2.3	2次モーメント行列の固有値の関係性.	12
2.4	ガウシアンフィルタの2次微分フィルタ.	13
2.5	FAST 検出器におけるコーナーの定義.	14
2.6	情報利得による同心円上ピクセルの選択.	15
2.7	決定木によるコーナー検出.	16
2.8	スケールスペースにおける Harris と Hessian のスコア.	17
2.9	DoG 画像からの極値検出.	19
2.10	SIFT のオリエンテーション算出.	22
2.11	Box フィルタによる2次微分ガウシアンフィルタの近似.	23
2.12	Box フィルタを利用した極値探索.	24
2.13	SURF のオリエンテーション算出.	25
2.14	画像ピラミッドによるスケール獲得.	26
2.15	パッチ画像変形に基づくアプローチの処理過程.	32
2.16	画像の2値化によるセグメンテーション.	33
2.17	SIFT 特徴量の記述.	34
2.18	SURF 特徴量の記述.	35
2.19	PCA-SIFT の特徴量記述.	36
2.20	GLOH による特徴量の記述.	37
2.21	2値特徴量のピクセルペアパターン.	38
2.22	BRISK の長距離ペアと短距離ペア.	39
2.23	ORB のピクセルペアの選択例.	41
2.24	BOLD のピクセルペア選択方法.	42
2.25	2値特徴量記述における重みフィルタ.	43
2.26	パッチ画像の positive ペアと negative ペアの例.	44
2.27	矩形フィルタによる重みフィルタの近似.	45
2.28	Bin Boost による2値特徴量の記述.	46

2.29	Bin Boost による学習の流れ.	47
2.30	画像の視点合成.	48
2.31	ASIFT によるキーポイントマッチング.	49
2.32	ASR-naive によるキーポイントマッチング.	51
2.33	ASR-fast によるキーポイントマッチング.	52
3.1	Harris と FAST のキーポイント検出結果の比較.	54
3.2	FAST コーナー検出器により検出されたコーナーのアピアランスの違い.	56
3.3	コーナーパッチ画像の解析対象の同心円.	57
3.4	FAST コーナー検出器により検出されたコーナーの解析.	58
3.5	周囲長 16 ピクセルの同心円におけるオリエンテーションの算出例.	59
3.6	異なる周囲長の同心円のオリエンテーション間の角度.	60
3.7	決定木の有無によるコーナー検出結果の比較.	61
3.8	FAST, Cascaded FAST, Harris のコーナー検出結果の比較.	62
3.9	Cascaded FAST によるコーナー検出の流れ.	63
3.10	オリエンテーションの評価.	64
3.11	Cascaded FAST によるキーポイント検出例.	65
3.12	Cascaded FAST と FAST の F 値の比較.	65
3.13	マッチングスコアと処理時間の比較.	66
3.14	キーポイントマッチングの処理時間の内訳.	67
3.15	各閾値における Cascaded FAST と FAST の性能.	67
4.1	単純楕円パターンによるアフィン領域推定の比較.	69
4.2	行列 $\mathbf{S}$ の対角成分.	72
4.3	SVD による非等方性 LoG フィルタの近似.	73
4.4	固有フィルタと固有関数.	73
4.5	非等方性 LoG フィルタの応答値計算の流れ.	74
4.6	パラメータを固定した際の固有関数 $\rho_5(\cdot)$ の数値.	75
4.7	連続固有関数の次数による非等方性 LoG フィルタの近似誤差.	76
4.8	複数のアフィン領域の探索.	77
4.9	提案手法による複数のアフィン領域の推定結果.	77
4.10	様々な楕円パターンに対するアフィン領域の推定結果.	78
4.11	IEEE Spectrum magazine dataset の例.	80
4.12	overlap error の例.	81
4.13	Oxford matching dataset での各手法の repeatability.	82
4.14	IEEE Spectrum magazine dataset での各手法の repeatability.	82
4.15	Oxford matching dataset での様々な閾値の repeatability.	83
4.16	IEEE Spectrum magazine dataset での様々な閾値の repeatability.	83

4.17	提案手法と Hessian-Affine によるキーポイントマッチング例.	84
4.18	3D 物体データセットの例.	84
4.19	3D 物体データセットを用いた画像検索の認識率.	85
4.20	フィルタ応答値の最大極値の割合を変化させた場合のキーポイントの平均アフィン領域数.	86
5.1	ORB に基づいて設計した特徴量記述フィルタ.	89
5.2	特徴量記述フィルタの視点合成.	90
5.3	SVD によるアフィン変換した特徴量記述フィルタ群のコンパクト化.	91
5.4	ORB の上位 60 枚の固有フィルタ.	92
5.5	多視点特徴量の計算.	93
5.6	アフィンパラメータ非依存行列 $\bar{\mathbf{A}}$ の生成.	94
5.7	多視点特徴量空間における特徴量間の最小距離.	94
5.8	特徴量ペアの下界に基づく対応点探索例.	95
5.9	固有フィルタ数 $N_f$ における平均 matching score.	96
5.10	上位 $N_l$ 個の下界を用いた平均 matching score.	97
6.1	提案手法による多視点特徴量の部分空間表現.	99
6.2	勾配方向ヒストグラムモデルの多視点特徴量記述.	102
6.3	GLOH に基づいて設計した特徴量記述フィルタ.	103
6.4	GLOH の上位 60 枚の固有フィルタ.	104
6.5	PCA の基底数 $N_p, N_s$ を変化させたときの平均 matching score.	105
6.6	固有フィルタ数 $N_f$ を変化させたときの平均 matching score.	105
6.7	異なる視点変化におけるキーポイントマッチングの精度.	107
6.8	HPatches の評価タスク.	108
6.9	HPatches の画像セット例.	109
6.10	HPatches benchmark における特徴量記述子の評価結果.	110
6.11	ASIFT の処理時間を 100% として表示した場合の各多視点特徴量記述子の比較.	110
7.1	ピック&プレースにおける物体認識の流れ.	113
7.2	提案手法の CNN の構造.	115
7.3	学習画像の生成.	117
7.4	学習用パッチ画像の例.	117
7.5	softmax 関数の計算.	118
7.6	Amazon Picking Challenge 2015 の認識対象物体.	120
7.7	Amazon Picking Challenge 2016 の認識対象物体.	120
7.8	APC 2015 データセットの認識率.	121
7.9	APC 2016 データセットの認識率.	122

7.10 Faster R-CNN により検出された物体矩形内の把持位置検出の例. . . . .	123
7.11 APC 2015 データセットの特徴量マッチングによる認識率. . . . .	125
7.12 APC 2016 データセットの特徴量マッチングによる認識率. . . . .	126

# 表目次

3.1	各同心円のオリエンテーションの最大誤差と最小誤差. . . . .	63
3.2	各手法のコーナー検出時間の比較. . . . .	64
3.3	比較手法の詳細. . . . .	66
4.1	楕円回転角の平均誤差 (degree). . . . .	79
4.2	Oxford matching dataset の見えの変化. . . . .	80
4.3	640 × 480 ピクセルの画像における処理時間 [s]. . . . .	85
5.1	各手法の平均 matching score [%] と処理時間 [s]. . . . .	98
6.1	提案手法のパラメータ設定. . . . .	106
7.1	提案手法の CNN 構成の詳細. . . . .	116
7.2	物体矩形内の把持位置検出の正解率 [%]. . . . .	124
7.3	認識の処理時間の内訳 [ms]. . . . .	125



# 第1章

## 序論

本章では，本研究の背景及び目的，本論文の構成について述べる。

## 1.1 研究の背景

キーポイントマッチングは、異なる視点から撮影された複数の画像間で、物理的に同一の位置(対応点)を探索する処理であり、コンピュータビジョンでは物体認識、画像検索、3次元復元など多岐にわたる分野に応用することができる基礎的な技術である。物体認識 [1] では、認識対象物体の画像をリファレンス画像として保持し、リファレンス画像と入力シーン画像との対応点を求めることで対象物体を検出することができる。例えば、車載カメラで撮影されたシーン画像と標識画像のキーポイントマッチングにより、標識を素早く認識してドライバーの支援が可能となる [2]。画像検索 [3] では、撮影された画像からキーポイントと局所特徴量を求めた後、データベースに保持されている大規模な画像特徴量と照合することで、撮影物体の名前や値段、メーカー等を検索結果として取得し、携帯電話端末等を使用したマーケティング分野に利用することができる。このような応用事例は撮影画像の物体と同一物体、すなわち固有名詞を答える問題であり、特定物体認識と呼ばれる。特定物体認識では、車載カメラや携帯電話端末などの比較的低スペックなハードウェアでのアプリケーションを対象としているため、キーポイントマッチングは高速かつ省メモリな処理が求められる。単純に、キーポイント検出や特徴量記述の処理を高速化するだけでなく、物体認識に不必要なキーポイントや特徴量を抑制することで、キーポイントマッチング全体の性能の向上が期待できる。

また、これまで主に特定物体認識の手法として用いられてきたキーポイント検出や局所特徴量記述は、一般物体認識にも応用されている。一般物体認識は、画像中の物体の一般的な名称によるカテゴリを答える問題である。局所特徴量を用いた一般物体認識は Bag-of-Features (BoF) [4] と呼ばれるアプローチにより実現されている。文書を単語の集合と見なして、単語の順序を無視して単語の頻度により文章の分類を行う Bag-of-Words[5] を画像に置き換えたのが Bag-of-Features である。Bag-of-Features は、画像を局所特徴量の集合と見なして、その位置情報(キーポイント座標)を無視して画像のカテゴリを分類する。

物体認識以外には、パノラマ画像生成 [6] や 3次元復元 [7]、Simultaneous Localization And Mapping (SLAM) による自動運転や自律ロボットの自己位置推定や 3次元地図生成等にキーポイントマッチングが用いられる [8, 9, 10]。3次元復元では、異なる視点で撮影された画像間でキーポイントマッチングを行い、対応点の位置関係から 3次元空間上の位置を推定することで 2次元画像から 3次元情報を復元する。しかし、視点変化を伴う画像間には射影変換が発生し、キーポイントマッチングが困難となる。視点変化を伴う画像に対して正確にキーポイントマッチングを行うには、キーポイントのアフィン変換に不変な領域推定、または局所特徴量の多視点表現が必要である。前者の場合、強い射影変化の影響によるキーポイント位置のずれや照明の変化により、正確なアフィン領域推定が困難となるため、そのような状況においても正確にアフィン領域を推定できるようなアプローチが必要である。後者の場合、視点変化にロバストな特徴量を記述するために画像に様々なアフィン変換を施して特徴量を記述する。画像をアフィン変換することで、様々な視点をシミュレートしていることになり、特徴量を多視点で表現できるため高精度なキーポイントマッチングが実現できる。画像のアフィン変換を繰り返すことによる特徴量記述の精度と処理時間はトレードオフの関係にある。これまでに、多視点特徴量の高精度化と高速化の両方の目的を達成した研究は提案されておら



ず、キーポイントマッチングの発展のために多視点特徴量の高精度化と高速化の両方を達成するアプローチが望まれている。

## 1.2 研究目的

本研究では、以下の3つの項目を目的とする。

1. 不必要なキーポイントの検出を抑制しつつ高速なキーポイント検出。
2. 視点変化にロバストなキーポイントの複数のアフィン領域推定。
3. 視点変化にロバストかつ効率的な多視点局所特徴量の記述。

1つ目の目的は、物体認識や画像検索といったアプリケーションを対象とし、このようなアプリケーションでは、車載カメラや携帯電話端末といった低スペックなハードウェア上で動作させることが多く、高速な処理を必要とする。2つ目と3つ目の目的は、3次元復元等のアプリケーションを対象とし、画像間の強い視点変化に対して頑健なキーポイントマッチングが必要である。以下に、各目的の詳細について述べる。

### 不必要なキーポイントの検出を抑制しつつ高速なキーポイント検出

キーポイントマッチングにおいて、処理時間が低下する原因の1つとして、不必要なキーポイントの過剰検出が挙げられる。キーポイントは、入力局所画像の勾配や輝度情報に基づいて検出するため、入力画像内にテクスチャが複雑な領域を多く含むとキーポイントが過剰に検出される。テクスチャが複雑であり、物体認識や画像検索にとって不必要な画像領域の多くは木の葉や植え込み等の自然画像領域である。自然画像領域は、物体認識や画像検索において認識対象とされることなく、多くの場合が背景として扱われる。よって、自然画像領域から検出されるキーポイントは物体認識や画像検索のアプリケーションにおいては不必要である。また、木の葉や植え込み等は風による葉の揺らぎのような外乱の影響を受けやすく、画像間で同一のキーポイントが検出できない問題も発生する。このような理由から、自然画像領域から検出されるキーポイントはキーポイントマッチングに不向きでありながら過剰に検出されやすいため、自然画像領域のキーポイント検出を抑制する。そのために、自然画像領域から検出される不必要なキーポイントと、物体認識等で対象物とされやすい人工物画像から検出されるキーポイントの周辺領域を解析する。それぞれのキーポイント周辺画像を解析して比較することで、自然画像領域のキーポイントと人工物画像のキーポイントの共通する傾向を捉え、その傾向に従ってキーポイントを検出する。また、高速なキーポイント検出を実現するために、決定木のアルゴリズムを用いて不必要なキーポイントを早期棄却するような仕組みでキーポイントを検出する。

### 視点変化にロバストなキーポイントの複数のアフィン領域推定

画像間に強い視点変化を伴う画像に対しても頑健に対応点を求めることは、キーポイントマッチングにおいて重要な課題である。視点変化を伴う画像間を正確に対応付けるには、キーポイントに

対してアフィン不変な楕円領域 (アフィン領域) を推定する必要がある。推定されたアフィン領域内の画像情報から局所特徴量を記述することで、視点変化を伴う画像間の対応点を求めることができる。従来のアフィン領域推定方法は、キーポイントの位置ずれや照明変化等により不正確なアフィン領域を推定してしまう問題がある。この問題を解決するには、キーポイントにおける複数のアフィン領域推定が有効であると考えられる。複数のアフィン領域を推定する手段として、様々な楕円形状の非等方性2次微分フィルタをキーポイント領域に畳み込んで、高い応答値が得られる楕円形状を複数個探索する方法が考えられる。しかし、非等方性の2次微分フィルタは数千種類の形状が存在し、それらのフィルタを検出された全キーポイントに畳み込む処理は計算コストが高い。そこで、2次微分フィルタ群を因子分解により少数の基底フィルタと重み係数の線形結合により近似することで、畳み込み演算の計算コストを削減する。よって、キーポイントの複数のアフィン領域を効率的に推定することができ、キーポイントマッチングの高精度化が期待できる。

### 視点変化にロバストかつ効率的な多視点局所特徴量の記述

キーポイントのアフィン領域推定だけでなく、局所特徴量記述においても視点変化に対するロバスト性を高める研究が取り組まれている。視点変化における従来の特徴量記述方法は、キーポイント周辺画像の多視点画像を生成し、その多視点画像から特徴量を記述することでロバストなキーポイントマッチングを実現している。キーポイントにおける多視点画像は、画像に様々なアフィン変換を施すことで様々な見えを表現し、その画像から特徴量を記述することで、多視点特徴量が生成できる。しかし、入力画像を直接アフィン変換する処理は非常に計算コストが高く、多くの処理時間を必要とする。そこで、本研究では、局所特徴量を入力画像とフィルタの畳み込みで記述することで、計算コストが高いアフィン変換処理を事前計算する。これは、特徴量を記述するフィルタを事前に設計しておくことで、画像ではなくフィルタにアフィン変換を適用することが可能となるため、計算コストを削減できる。この場合においても、アフィン変換した数十万枚のフィルタを畳み込むため、フィルタ群を因子分解により低ランクな基底フィルタで近似させる。特徴量記述のためのフィルタを基底フィルタで近似することで、多視点特徴量を効率的かつ高精度に記述することができる。

## 1.3 本論文の構成

本論文は、図 1.1 に示すように8つの章で構成されている。1章では、本研究の背景と目的を述べた。本研究では、高速なキーポイントマッチングのためのキーポイント検出、視点変化画像におけるキーポイントマッチングの高精度化について、それぞれ新たな枠組みを提案する。

2章では、キーポイントマッチングと関連研究について述べる。キーポイントマッチングシステムの構築方法とキーポイントマッチングの各処理における関連研究を調査してまとめる。

3章では、不要なキーポイント検出の抑制によるキーポイントマッチングの高速化について述べる。キーポイントマッチングの処理全体を高速化するためには、物体認識や画像検索における不必要なキーポイントの検出を抑制させる。また、キーポイントを木構造を用いて検出することで、不必要なキーポイントを高速に棄却することが可能であることを示す。

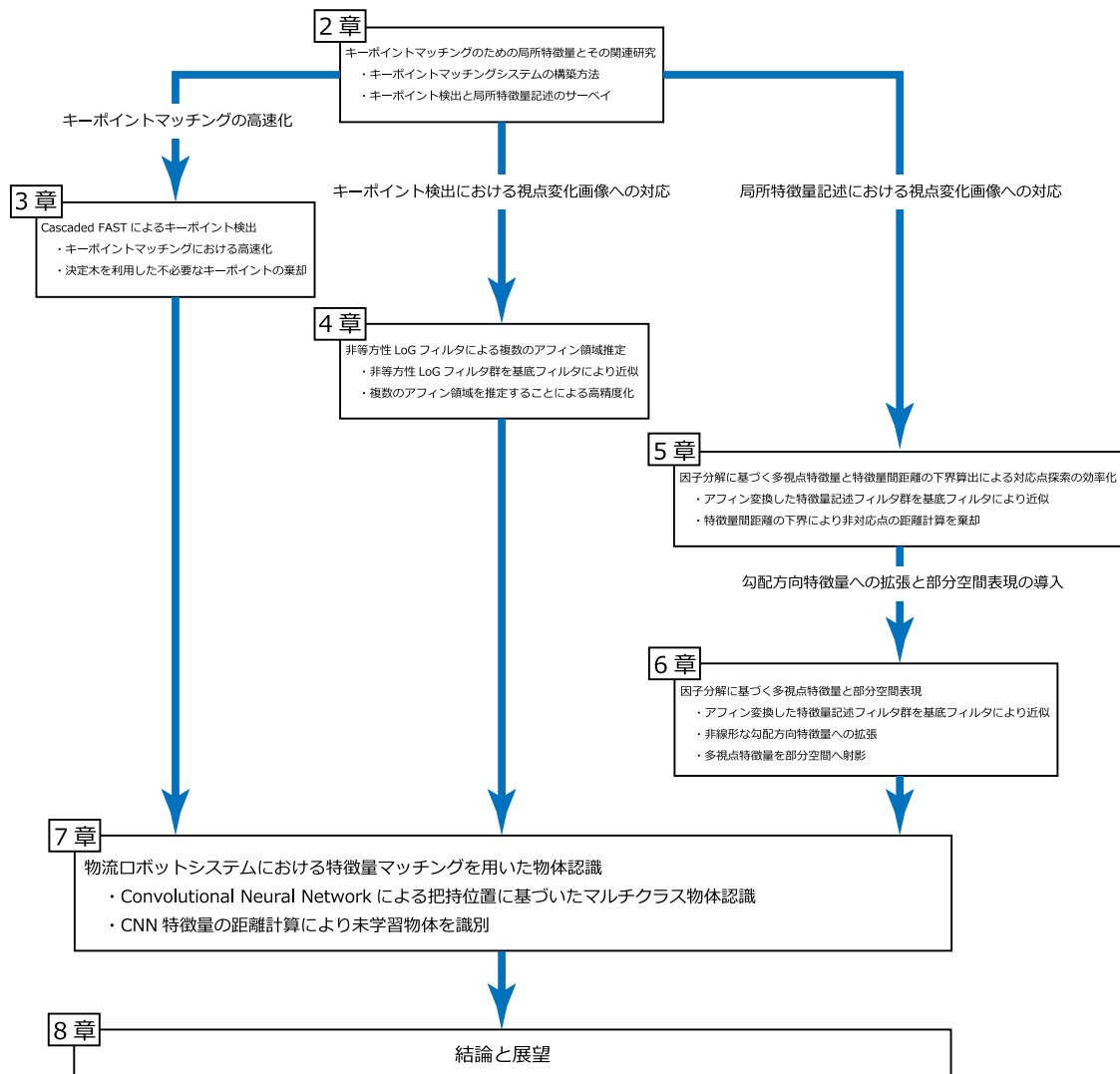


図 1.1: 本論文の構成.

4 章では、キーポイント検出におけるキーポイントマッチングの高精度化について述べる。視点変化を伴う画像において高精度にキーポイントをマッチングさせるために、複数のアフィン領域をキーポイントに対して推定する。アフィン領域を推定するための非等方性フィルタを因子分解により少数の基底フィルタで近似することで、効率的かつ高精度にアフィン領域を推定できることを示す。

5 章では、多視点特徴量の新たな表現方法と効率的な対応点探索について述べる。従来では、入力画像をアフィン変換して多視点特徴量を記述するのに対して、提案手法では特徴量記述をフィルタの畳み込みで設計し、フィルタ自身を事前にアフィン変換しておくことでオンライン時の計算コストを削減する。さらに、特徴量間の距離の下界に基づいて距離計算の棄却を行うことで、効率的な対応点探索が可能であることを示す。

6 章では、5 章で述べた多視点特徴量記述に部分空間表現を導入する。部分空間表現を導入するこ

とで、視点変化に対してより頑健な特徴量を記述することが可能となる。多視点特徴量のさらなる高精度化として、勾配方向特徴量への拡張方法についても述べる。

7章では、物流ロボットにおける特徴量マッチングを用いた物体認識について述べる。Convolutional Neural Network (CNN) による局所的な画像特徴量から、マルチクラス物体認識を実現する。また、特徴量マッチングの考え方を CNN の物体認識に導入することで未学習の物体クラスも識別することが可能となる。これは、物流ロボットシステムの実問題における物体認識を想定したタスクである。

8章では、本論文の結論と展望について述べる。

## 第2章

# キーポイントマッチングのための 局所特徴量とその関連研究

キーポイントマッチングは、物体認識、画像検索、3次元復元等の技術に応用され、古くから取り組まれてきた研究の1つである。そして、幾何学的変化(回転、スケール、アフィン変形)や照明変化等の画像の様々な見えの変化に対して頑健なキーポイントマッチングが研究課題とされている。また、実用化を考慮して高速かつ省メモリな手法も多く提案されている。本章では、キーポイントマッチングシステムの構築方法を述べ、キーポイントマッチングにおける処理を明確にし、各処理における関連研究として、キーポイント検出法や局所特徴量記述について調査してまとめる。

## 2.1 キーポイントマッチングについて

キーポイントマッチングは、(1) キーポイント検出、(2) 局所特微量記述、(3) 対応点探索、の3つの処理で構成され、それぞれの処理で高速化や高精度化に関する研究が取り組まれている。以下にキーポイントマッチングにおける3つの処理について述べる。

### (1) キーポイント検出

キーポイントとは、画像の局所領域における勾配や輝度情報に基づいて検出されるユニークな点である。もし、キーポイント検出を行わずに対応点を求める場合、画像の全ピクセルにおいて局所特微量を記述して大量の特微量間で距離を計算することとなり、非常に高い計算コストが必要となる。画像から、あらかじめキーポイントを検出しておくことで、検出されたキーポイントのピクセルのみに対して特微量を記述して距離を計算すれば良いため、計算コストを大幅に削減できる。また、キーポイントは画像中の特徴的な点を検出するため、キーポイントにおいてはユニークな局所特微量を記述しやすくなり、誤マッチング等による性能低下を避けることができる。キーポイント検出には、検出されたキーポイントの位置の正確さ (localization) と2画像間におけるキーポイント検出の再現性 (repeatability) が重要とされ、様々な方法が提案されている。キーポイント検出の初期の研究として、Moravec コーナー検出器 [11, 12]、Hessian 検出器 [13]、Harris コーナー検出器 [14]、SUSAN コーナー検出器 [15] 等が提案され、これらの手法の多くは画像中のエッジの交わる点、すなわちコーナー点をキーポイントとして検出するように設計されている。また、キーポイント検出にスケールスペース理論 [16, 17] による画像構造の解析を取り入れることで、スケール変化に対応したキーポイント検出器も提案されている [1, 18, 19, 20]。さらに、キーポイントに対して楕円状のアフィン領域を推定することで、画像間に射影変形が発生した場合にも高精度にキーポイントマッチングを行うことができる [21, 22, 23, 24]。

### (2) 局所特微量記述

局所特微量は、画像から検出されたキーポイントの周辺画像の勾配やテクスチャ等の情報を特徴ベクトルとして表現する。画像の輝度値をそのまま特徴ベクトルとして採用した場合、照明変化やノイズの影響を大きく受けるため画像間で正確な対応点を求めることが困難である。そこで、局所特微量記述では勾配方向ヒストグラムやピクセル間の輝度差等の情報を用いて、照明変化やノイズの影響を受けにくい特徴ベクトルを計算する。局所特微量は Scale-Invariant Feature Transform (SIFT) [1] をはじめとして、局所画像の勾配情報等に基づいて実数特徴ベクトルを計算する手法が多く提案されている [18, 25, 26, 27, 28]。しかし、実数ベクトルによる局所特微量はキーポイントマッチングの精度が高い一方で、メモリ使用量の増加や高速に特微量を記述できないという課題がある。この問題の解決策として、局所特微量を  $\{0, 1\}$  または  $\{-1, 1\}$  の2値ベクトルで表現する方法が提案されている [29, 30, 31, 32, 33]。2値ベクトルによる特微量記述の多くは、キーポイント周辺のパッチ画像内のピクセル間の輝度差や、パッチ画像と線形フィルタの内積結果の符号により  $\{0, 1\}$  または  $\{-1, 1\}$  を割り当てるため高速な特微量記述が可能である。さらに、特微量は全て2値で表されるた

め、省メモリなキーポイントマッチングが可能となる。2 値ベクトルによる特徴量記述は実数ベクトルに比べて情報が削減され、キーポイントマッチングの精度が低下することから、実数ベクトルによる特徴量から特徴変換行列により 2 値ベクトルへ変換する間接的な 2 値特徴量記述の研究も取り組まれている [34, 35, 36, 37].

上記で述べた特徴量記述は、キーポイント検出の段階で求められた回転角度、等方性スケール領域あるはアフィン領域に基づいて、固定サイズのパッチ画像に正規化することで、画像変形に不変な特徴量を記述する。一方で、特徴量記述の処理においても画像変形に対してロバストな特徴量を抽出する手法が幾つか提案されている。これは、キーポイントのパッチ画像に対して様々なアフィン変換を適用し、様々な視点をシミュレートしたアフィン変換画像から局所特徴量を計算することで、多視点特徴量を記述する [38, 39]. このように、パッチ画像をアフィン変換させて特徴量を記述することで、様々な視点における特徴量を表現できるため、強い視点変化においても高精度なキーポイントマッチングが可能となる。

### (3) 対応点探索

対応点探索では、式 (2.1), 式 (2.2) のように 2 画像間で検出したキーポイントに対して計算した局所特徴量の  $N_{dim}$  次元ベクトル  $\mathbf{d}, \mathbf{d}' \in \mathbb{R}^{N_{dim}}$  を比較する。特徴量間の比較にはユークリッド距離  $\text{dist}_E$  やベクトル間角度  $\text{dist}_\theta$  が用いられる。

$$\text{dist}_E(\mathbf{d}, \mathbf{d}') = \sqrt{(\mathbf{d} - \mathbf{d}')^\top (\mathbf{d} - \mathbf{d}')} \quad (2.1)$$

$$\text{dist}_\theta(\mathbf{d}, \mathbf{d}') = \cos^{-1} \left( \frac{\mathbf{d}^\top \mathbf{d}'}{\|\mathbf{d}\|_2 \|\mathbf{d}'\|_2} \right) \quad (2.2)$$

また、2 画像間の局所特徴量が  $N_{dim}$  ビットからなる 2 値ベクトル  $\mathbf{d}, \mathbf{d}' \in \{0, 1\}^{N_{dim}}$  であれば、距離計算は式 (2.3) に示されるハミング距離  $\text{dist}_H$  が用いられる。

$$\text{dist}_H(\mathbf{d}, \mathbf{d}') = \text{bitcnt}(\mathbf{d} \oplus \mathbf{d}') \quad (2.3)$$

ここで、 $\oplus$  はベクトルの要素ごとに XOR を計算する演算子であり、 $\text{bitcnt}(\cdot)$  は 2 値ベクトルの 1 が立つビットをカウントする関数である。局所特徴量が 2 値ベクトルの場合は、特徴量間距離を論理演算と単純なビットカウントのみで計算できるため、対応点探索においても高速化が可能である。

対応点を探索する最も単純な方法は 2 画像間の全特徴量ペアにおける全探索 (brute-force search) である。1 枚目の画像から検出されたある 1 点のキーポイントの特徴量と 2 枚目の画像から検出された全てのキーポイントの特徴量間の距離を計算し、最も距離が近い特徴量ペアである 1st nearest neighbor  $\{\mathbf{d}, \mathbf{d}'\}$  と 2 番目に距離が近い特徴量ペアである 2nd nearest neighbor  $\{\mathbf{d}, \mathbf{d}''\}$  を求める。1st nearest neighbor と 2nd nearest neighbor の距離値の比率が閾値  $T_{ratio}$  以下の場合に 1st nearest neighbor  $\{\mathbf{d}, \mathbf{d}'\}$  のキーポイントペアを対応点として採用する。

$$\frac{\text{dist}(\mathbf{d}, \mathbf{d}')}{\text{dist}(\mathbf{d}, \mathbf{d}'')} \leq T_{ratio} \quad (2.4)$$



図 2.1: ガウシアンフィルタの 1 次微分フィルタ.

閾値  $T_{ratio}$  は、アプリケーション等により決定する。例えば、 $T_{ratio}$  が小さい場合、誤対応点数を減らすことができるため、投票ベースの物体認識等に有効である。 $T_{ratio}$  が大きい場合は誤対応点数が多くなるが、全体的な対応点数が増えるため多少のアウトライアを許容できる RANSAC [40] 等を用いた画像間の位置合わせ処理に有効である。

## 2.2 キーポイント検出

ここでは、主にコーナー検出法に焦点を当てる。ここで述べるキーポイント検出法は、スケールスペースや領域推定を行わず、キーポイントの座標のみを出力する。

### 2.2.1 Harris コーナー検出

Harris コーナー検出器 [14] は複数のエッジの交点をコーナーとして定義することで、キーポイントを検出する手法である。まず、入力画像  $\mathbf{I}$  に対して  $x, y$  方向の 1 次微分  $I_x(\mathbf{p}; \sigma_D)$ ,  $I_y(\mathbf{p}; \sigma_D)$  を計算する ( $\mathbf{p} = (x, y)$ )。画像の微分は図 2.1(b), 図 2.1(c) に示すようなガウス関数  $g(\sigma_D)$  を  $x, y$  の各方向で 1 次微分したフィルタを画像に畳み込んで求める。

$$I_x(\mathbf{p}; \sigma_D) = \frac{\partial g(\sigma_D)}{\partial x} * I(\mathbf{p}) \quad (2.5)$$

$$I_y(\mathbf{p}; \sigma_D) = \frac{\partial g(\sigma_D)}{\partial y} * I(\mathbf{p}) \quad (2.6)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\bar{x}^2 + \bar{y}^2}{2\sigma^2}\right) \quad (2.7)$$

$\sigma_D$  は 1 次微分ガウス関数の標準偏差 (スケールパラメータ) であり、 $\bar{x}, \bar{y}$  はガウシアンフィルタの中心からの距離である。そして、次式に示す 2 次モーメント行列  $\boldsymbol{\mu}$  を用いて局所領域における勾配情報を計算する。

$$\boldsymbol{\mu} = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(\mathbf{p}; \sigma_D) & I_x(\mathbf{p}; \sigma_D)I_y(\mathbf{p}; \sigma_D) \\ I_x(\mathbf{p}; \sigma_D)I_y(\mathbf{p}; \sigma_D) & I_y^2(\mathbf{p}; \sigma_D) \end{bmatrix} \quad (2.8)$$



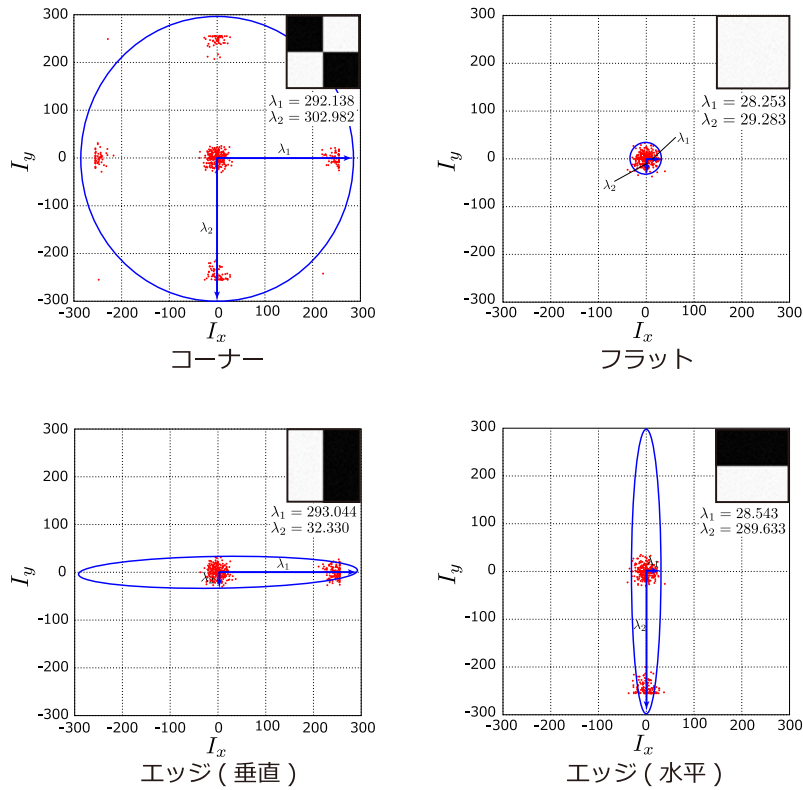


図 2.2: コーナー, エッジ, フラット領域の微分値の分布と固有値の関係。

式 (2.8) は, 単純に画像の 1 次微分を計算するだけでなく, 局所領域  $\sigma_I$  における微分値の総和を求める. これは図 2.2 に示すように, 画像の局所領域における微分値の分布を考慮するためである. 局所領域画像がフラットの場合,  $x, y$  方向の微分値が小さくなるため微分値の分布の分散, すなわち行列  $\boldsymbol{\mu}$  の第 1 固有値, 第 2 固有値が小さくなる. また, エッジ領域の場合は  $x$  方向または  $y$  方向のどちらかの微分値の分布の分散, すなわち行列  $\boldsymbol{\mu}$  の第 1 固有値のみが大きくなる. コーナー領域の場合は  $x, y$  の各方向の微分値の分布の分散が大きくなるため, 行列  $\boldsymbol{\mu}$  の第 1 固有値と第 2 固有値が大きくなる. このような性質を得るために, 式 (2.8) では, 局所領域における微分値の総和を求めている. このとき, 局所領域の微分値の単純な総和でも良いが, ガウス関数  $g(\sigma_I)$  による重み付き和を用いることが多い.  $\sigma_D$  はガウシアン 1 次微分フィルタのカーネルサイズであり,  $\sigma_I$  は微分値の総和を求めるときのカーネルサイズである. これらの 2 つのカーネルサイズは  $\sigma_D = 0.7\sigma_I$  のように設定される [24].

図 2.2 の関係から式 (2.8) の 2 次モーメント行列  $\boldsymbol{\mu}$  の固有値  $\lambda_{e1}, \lambda_{e2}$  は図 2.3 のような関係性が得られる. Shi & Tomasi による最小固有値に基づくコーナー検出法 [41] では行列  $\boldsymbol{\mu}$  の最小固有値  $\min(\lambda_{e1}, \lambda_{e2})$  を閾値処理することでコーナーを検出をしているが, Harris コーナー検出器は以下の

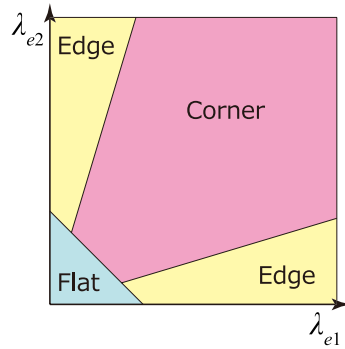


図 2.3: 2 次モーメント行列の固有値の関係性.

コーナースコア  $h_s$  を定義している.

$$h_s = \det(\boldsymbol{\mu}) - k_s \cdot \text{trace}^2(\boldsymbol{\mu}) \quad (2.9)$$

$$\det(\boldsymbol{\mu}) = \lambda_{e1} \cdot \lambda_{e2} \quad (2.10)$$

$$\text{trace}(\boldsymbol{\mu}) = \lambda_{e1} + \lambda_{e2} \quad (2.11)$$

このように Harris コーナー検出器は行列  $\boldsymbol{\mu}$  の固有値問題を実際に解くのではなく、行列  $\boldsymbol{\mu}$  の行列式  $\det$  と対角成分の和  $\text{trace}$  を組み合わせてコーナースコアを計算している。  $k_s$  はコーナースコアの調整パラメータであり、  $k_s = 0.04 \sim 0.06$  が最適値とされている [42, 43, 44]。 検出したコーナーには、  $3 \times 3$  ピクセルのような小領域において最大スコアのコーナーを残し、その他の非最大スコアを持つ近傍コーナーを除去する non-maximum suppression 処理を行う。

コーナー検出時は入力画像の各ピクセルにおいてコーナースコア  $h_s$  を算出し、  $h_s$  の値に対して適切な閾値を設けることでコーナーのみを検出する。 また、式 (2.9) の値を使用して類似度補間手法のように 2 次関数を当てはめることで、コーナーのサブピクセル位置も計算することができる。

## 2.2.2 Hessian 検出器

Hessian によるキーポイント検出 [13] は、画像の画素値が極値を取るような点をキーポイントとして検出する。 画像は座標  $\mathbf{p} = (x, y)$  と画素値  $I(\mathbf{p})$  の 3 次元空間において連続的変化を考えると曲面として見る事ができる。 2 次曲面の座標  $\mathbf{p}$  において極値であるか否かは次式に示す Hessian 行列  $\mathcal{H}$

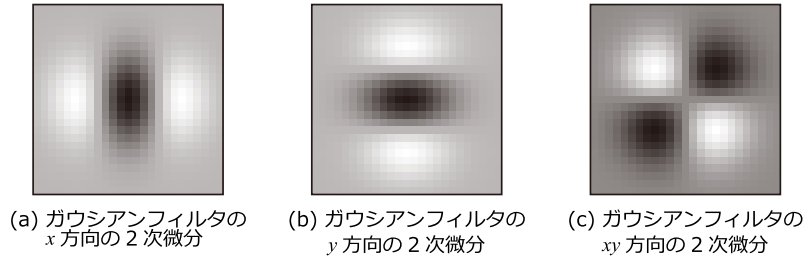


図 2.4: ガウシアンフィルタの 2 次微分フィルタ.

の行列式で判定することができる.

$$\mathcal{H} = \sigma_D^2 \begin{bmatrix} I_{xx}(\mathbf{p}; \sigma_D) & I_{xy}(\mathbf{p}; \sigma_D) \\ I_{xy}(\mathbf{p}; \sigma_D) & I_{yy}(\mathbf{p}; \sigma_D) \end{bmatrix} \quad (2.12)$$

$$I_{xx}(\mathbf{p}; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial x^2} * I(\mathbf{p}) \quad (2.13)$$

$$I_{yy}(\mathbf{p}; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial y^2} * I(\mathbf{p}) \quad (2.14)$$

$$I_{xy}(\mathbf{p}; \sigma_D) = \frac{\partial^2 g(\sigma_D)}{\partial xy} * I(\mathbf{p}) \quad (2.15)$$

$\sigma_D$  はガウシアンフィルタのカーネルサイズであり, 図 2.4 に示すようなガウス関数  $g(\cdot)$  を  $x, y$  の各方向で 2 次微分したフィルタを画像に畳み込むことで画像の 2 次微分を計算し, その値を要素を持つ Hessian 行列の行列式を求めることで以下の判定が可能となる.

- $\det(\mathcal{H}) > 0$  かつ  $I_{xx}(\mathbf{p}; \sigma_D) < 0$  : 座標  $\mathbf{p}$  において極大値を取る.
- $\det(\mathcal{H}) > 0$  かつ  $I_{xx}(\mathbf{p}; \sigma_D) > 0$  : 座標  $\mathbf{p}$  において極小値を取る.
- $\det(\mathcal{H}) < 0$  : 座標  $\mathbf{p}$  において極値を取らない (鞍点).
- $\det(\mathcal{H}) = 0$  : 座標  $\mathbf{p}$  において極値か否かは不明.

この極値判定を画像の全ピクセルに対して行う.  $\det(\mathcal{H})$  の値をキーポイントのスコアとし, スコアが閾値を上回る場合にキーポイントとして検出する. 検出したキーポイントには,  $3 \times 3$  ピクセルのような小領域において最大スコアのキーポイントを残し, その他の非最大スコアを持つ近傍キーポイントを除去する non-maximum suppression 処理を行う.

### 2.2.3 Features from Accelerated Segment Test (FAST)

Harris コーナー検出器や Hessian 検出器は画像の各ピクセルで微分値を計算したり, ガウシアンフィルタの畳み込みが必要であったため, 計算コストが高くなる問題がある. Features from Accelerated Segment Test (FAST) [45] はコーナーを高速に検出するために, あらかじめ定められたコーナーの定

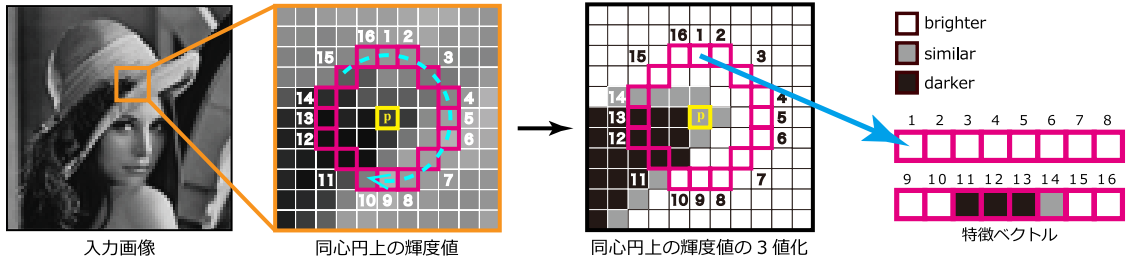


図 2.5: FAST 検出器におけるコーナーの定義.

義に従って画像中のコーナーを決定木で学習する。学習された決定木は、非コーナーを早期判定できるような仕組みになっており、決定木による探索で高速にコーナーを検出することができる。

### ■ コーナーの定義

FAST コーナー検出器では注目ピクセル  $p$  を中心とする周囲長 16 ピクセルの同心円上の画像を参照する。そして、図 2.5 に示すように注目ピクセルの輝度値と比較して周囲長 16 ピクセルの同心円上の輝度値が  $N_{seg}$  ピクセル以上連続して明るい、または暗い場合に注目ピクセルをコーナーとする。著者らは、実験により  $N_{seg} = 9$  の場合に最も repeatability が高くなると報告している [45]。フラットな領域において、 $N_{seg} = 9$  で非コーナーを判定する場合は、同心円上の 16 ピクセルを全て観測する必要はなく、2 ピクセル程度観測すれば非コーナーとして早期判定が可能である。しかし、同心円上の観測点によっては、フラットな領域でも早期判定が困難な場合が発生する。これを改善するために、FAST コーナー検出器では機械学習によりコーナーを学習し、最もコーナーと非コーナーを分離しやすい同心円上の観測点を統計的に決定する。決定した観測点を木構造で表現することで、コーナー検出時は決定木による探索でコーナーの判定が可能となる。

### ■ 機械学習による決定木の構築

FAST のコーナー定義に従ってコーナーを検出する場合は、同心円上のピクセルを効率的に観測するために機械学習により決定木を構築する。決定木の学習には、まず学習画像の全てのピクセル  $p$  において、同心円上のピクセルを明るい (brighter)、類似 (similar)、暗い (darker) の 3 値に分類する。

$$S(\mathbf{p}_c) = \begin{cases} \text{brighter} & I(\mathbf{p}) + T_f \leq I(\mathbf{p}_c) \\ \text{similar} & I(\mathbf{p}) - T_f < I(\mathbf{p}_c) < I(\mathbf{p}) + T_f \\ \text{darker} & I(\mathbf{p}_c) \leq I(\mathbf{p}) - T_f \end{cases} \quad (2.16)$$

ここで、 $I(\mathbf{p})$  は座標  $\mathbf{p}$  における輝度値、 $I(\mathbf{p}_c), c = \{1, 2, \dots, 16\}$  は周囲長 16 ピクセルの同心円上の輝度値、 $T_f$  は 3 値に分類する際の閾値である。図 2.5 に示すように、3 値化した周囲長 16 ピクセルを特徴ベクトルとして生成する。そして、同心円上の 16 ピクセルで  $N_{seg}$  ピクセル以上連続して

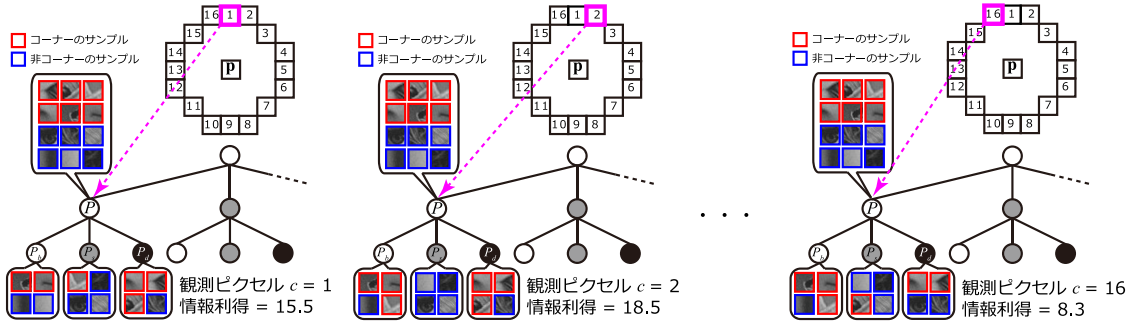


図 2.6: 情報利得による同心円上ピクセルの選択.

brighter または darker の場合にコーナー，そうでない場合は非コーナーとして座標  $\mathbf{p}$  にラベルを与える．次に 3 値化した同心円上の 16 ピクセルの特徴ベクトルと座標  $\mathbf{p}$  の教師ラベルにより，ID3 [46] に基づく決定木構築アルゴリズムに従って決定木を学習する．決定木の分岐ノードでは同心円上の値  $S(\mathbf{p}_c)$  を観測し，式 (2.17) で求められる情報利得  $G_{info}$  が最も高い同心円上のピクセル  $\mathbf{p}_c$  を決定する．

$$G_{info} = H_e(P) - H_e(P_b) - H_e(P_s) - H_e(P_d) \quad (2.17)$$

ここで， $P$  は分岐ノードにたどり着いた  $\mathbf{p}$  の集合， $P_b$  は  $S(\mathbf{p}_c) = \text{brighter}$  と判定された  $\mathbf{p}$  の集合， $P_s$  は  $S(\mathbf{p}_c) = \text{similar}$  と判定された  $\mathbf{p}$  の集合， $P_d$  は  $S(\mathbf{p}_c) = \text{darker}$  と判定された  $\mathbf{p}$  の集合である． $H_e$  はエントロピーを表し，次式より計算できる．

$$H_e(P) = (C + \bar{C}) \log_2(C + \bar{C}) - C \log_2 C - \bar{C} \log_2 \bar{C} \quad (2.18)$$

ここで， $C$  は各分岐ノードにたどり着いたコーナーのラベル数， $\bar{C}$  は各分岐ノードにたどり着いた非コーナーのラベル数である． $H_e(P_b), H_e(P_s), H_e(P_d)$  においても  $H_e(P)$  と同様に，各分岐ノードにたどり着いたラベル数を用いてエントロピーを計算する．この処理をコーナーと非コーナーを完全に分類するまで，すなわち情報利得が 0 になるまで決定木のノードを分岐する．情報利得が 0 となったときのノードを末端ノードとし，たどり着いたラベルを記録し，末端ノードの最終的な出力ラベルとなる．図 2.6 は情報利得による同心円上のピクセル選択の例である．観測ピクセルが  $c = 2$  のとき，学習サンプルのコーナーと非コーナーを最も分類でき，情報利得が高くなるため  $c = 2$  が選択される．

### ■ 決定木によるコーナーの検出

決定木によりコーナーを検出する際には，図 2.7 に示すように学習した決定木へ座標  $\mathbf{p}, \mathbf{p}_c$  の輝度値を入力して分岐させる．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を判定する．決定木を用いることで，同心円上のピクセルを効率的に参照するた

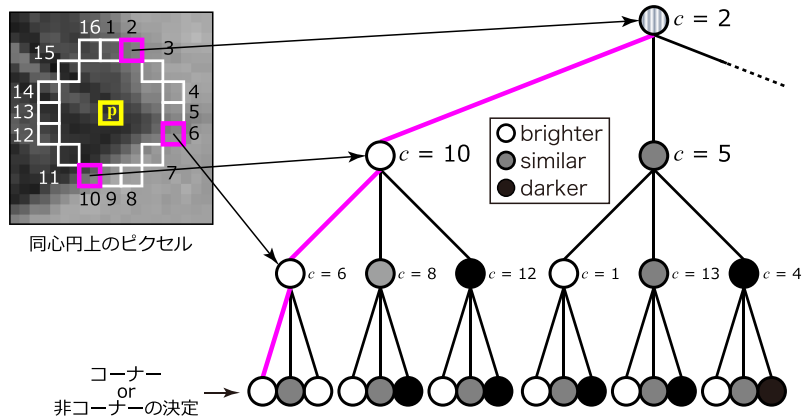


図 2.7: 決定木によるコーナー検出.

め、入力画像のほとんどの画素を早期判定できる。これは、画像のほとんどの局所領域がフラットな領域であるため、同心円上のピクセルを平均で 2.25 ピクセル参照するだけで非コーナーの早期判定が可能となる。

### ■ Non-maximum suppression

FAST コーナー検出器は決定木による探索でコーナーを検出するため、コーナーのスコアが得られない。FAST コーナー検出器においても、Harris や Hessian と同様に小領域において non-maximum suppression を行うには検出したコーナーに対してスコアを別処理で計算する必要がある。FAST コーナー検出器におけるスコアは、式 (2.19) に示すように、コーナー点のピクセルと同心円上の brighter または darker のピクセル間の輝度差の絶対値の合計をスコア  $f_s$  として算出する。

$$f_s = \max \left( \sum_{c \in S_b} |I(\mathbf{p}_c) - I(\mathbf{p})| - T_f, \sum_{c \in S_d} |I(\mathbf{p}) - I(\mathbf{p}_c)| - T_f \right) \quad (2.19)$$

$$S_b = \{c | I(\mathbf{p}_c) \geq I(\mathbf{p}) + T_f\} \quad (2.20)$$

$$S_d = \{c | I(\mathbf{p}_c) \leq I(\mathbf{p}) - T_f\} \quad (2.21)$$

このスコアを用いて小領域における non-maximum suppression 処理を行う。

## 2.3 スケールスペースを用いたキーポイント検出

2.2 節で述べたキーポイント検出法はキーポイントの座標のみを出力する手法である。キーポイントの座標のみの出力の場合、画像の回転や平行移動に対しては頑健なキーポイントマッチングが可能となるが、拡大・縮小といったスケール変化に対応することができない。ここでは、キーポイント検出にスケールスペース理論 [16, 17, 47, 48, 49] を導入した、スケール不変なキーポイント検出法を述べる。

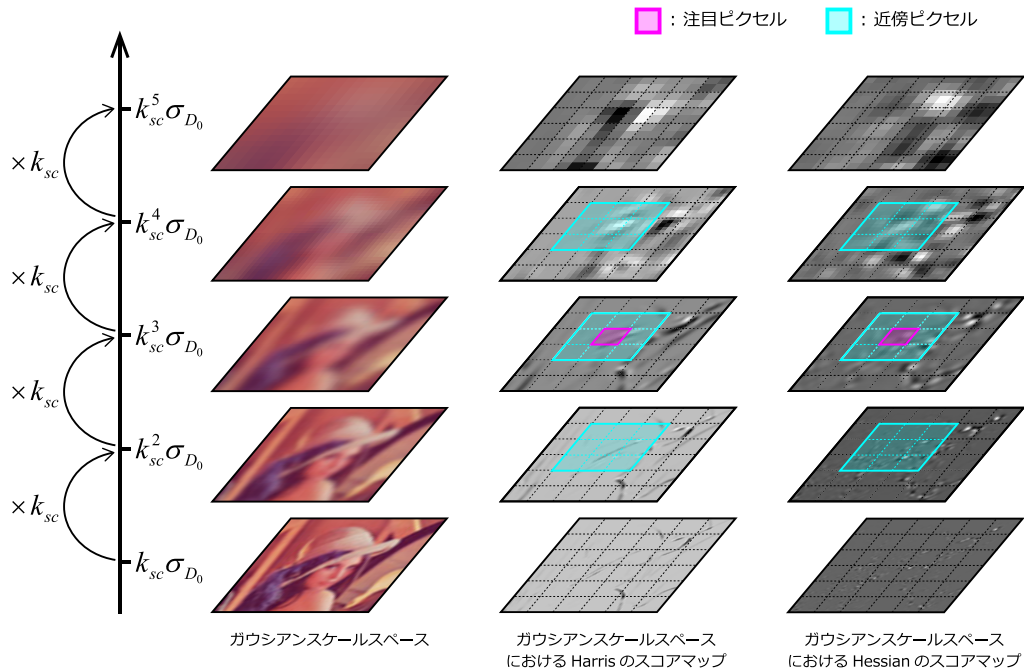


図 2.8: スケールスペースにおける Harris と Hessian のスコア.

### 2.3.1 Harris-Laplace と Hessian-Laplace

Harris-Laplace と Hessian-Laplace [19] は Harris コーナー検出器や Hessian 検出器にガウシアンスケールスペースを導入することで、スケール不変なキーポイントを検出する. Harris-Laplace と Hessian-Laplace におけるスケールスペースはガウシアンフィルタのスケール  $\sigma_D$  を徐々に変化させて式 (2.8) や式 (2.12) のスコアを算出する. そして, 図 2.8 に示すように, スケールスペースの各階層において, 注目ピクセルのスコアが  $[x, y, \sigma_D]$  の 3 次元空間の 26 近傍において極値となる場合にキーポイントを検出する. キーポイントを検出する場合, 注目ピクセルのスケール  $\sigma_D$  をキーポイントのスケールとする. スケールスペースにおける  $\sigma_D$  は  $\{\sigma_{D_1}, \sigma_{D_2}, \dots, \sigma_{D_n}\} = \{k_{sc}\sigma_{D_0}, k_{sc}^2\sigma_{D_0}, \dots, k_{sc}^n\sigma_{D_0}\}$  のように変化させる. ここで,  $\sigma_{D_0}$  は初期スケールであり  $k_{sc}$  はスケールの増加率である. それぞれ,  $\sigma_{D_0} = 1.0, k_{sc} = 1.4$  として設定する [19].

### 2.3.2 Scale-Invariant Feature Transform (SIFT) Detector

Scale-Invariant Feature Transform (SIFT) [1] は Harris-Laplace と Hessian-Laplace と同様にスケールスペースを利用することで, 画像の回転とスケール変化に不変なキーポイントを検出することができる. SIFT のアルゴリズムはキーポイント検出と特徴量記述の 2 つの処理を含んでおり, ここでは SIFT のキーポイント検出について説明する.

### ■ Difference-of-Gaussian (DoG) によるスケールスペース

DoG によるキーポイント検出は、異なるスケールのガウス関数  $g(\sigma)$  と入力画像  $I(\mathbf{p})$  を畳み込んだ平滑化画像  $L(\mathbf{p}; \sigma)$  の差分 (DoG 画像) から求める。

$$L(\mathbf{p}; \sigma) = g(\sigma) * I(\mathbf{p}) \quad (2.22)$$

DoG 画像を  $D(\mathbf{p}; \sigma)$  とすると、DoG 画像を次式で計算することができる。

$$\begin{aligned} D(\mathbf{p}; \sigma) &= (g(k_{sc}\sigma) - g(\sigma)) * I(\mathbf{p}) \\ &= L(\mathbf{p}; k_{sc}\sigma) - L(\mathbf{p}; \sigma) \end{aligned} \quad (2.23)$$

この処理を初期スケール  $\sigma_0$  から  $k_{sc}$  倍ずつ大きくした異なるスケール間で行い、複数の DoG 画像を求める。  $\sigma$  が一定の割合で増加し続けると、ガウシアンフィルタのサイズが大きくなり、処理できない画像の端領域と計算コストの増加という問題が発生する。この問題に対して、画像のダウンサンプリングにより  $\sigma$  の変化の連続性を保持した平滑化処理を行う。

### ■ $\sigma$ の連続性を保持した効率的な平滑化処理

$\sigma$  の連続性を保持した効率的な平滑化処理では、最初に入力画像を初期値である  $\sigma_0$  で平滑化し、平滑化画像  $L_1(\mathbf{p}; \sigma_0)$  を取得する。次に  $\sigma_0$  を  $k_{sc}$  倍した値  $k_{sc}\sigma_0$  で画像を平滑化し、 $L_1(\mathbf{p}; k_{sc}\sigma_0)$  を生成する。同様の処理により、 $\sigma$  の異なる複数の平滑化画像を生成する。ここまでの処理のセットを 1 オクターブと呼ぶ。次に複数生成された平滑化画像の中から  $2\sigma_0$  で平滑化された画像  $L_1(\mathbf{p}, 2\sigma_0)$  を  $\frac{1}{2}$  のサイズにダウンサンプリングする。1 オクターブにおける平滑化の処理回数については増加率  $k_{sc}$  の設定とともに後述する。  $\frac{1}{2}$  のサイズにダウンサンプリングされた画像  $L_2(\mathbf{p}; \sigma_0)$  と、  $2\sigma_0$  で平滑化した画像  $L_1(\mathbf{p}; 2\sigma_0)$  には以下のような関係が成り立つ。

$$L_1(\mathbf{p}; 2\sigma_0) \approx L_2(\mathbf{p}; \sigma_0) \quad (2.24)$$

この関係を利用することで、 $\sigma$  の範囲を制限することができるため、ガウシアンフィルタのサイズによる計算量の増加を防ぐことができる。

### ■ DoG 画像の極値探索

DoG は異なるスケールによる平滑化画像の差分であるため、DoG のスコア (= DoG 画像の各ピクセルの値) が大きい  $\sigma$  では、スケールが変化する領域にエッジ等の情報量を多く含んでいる。そこで、DoG 画像から極値を検出し、キーポイント候補とそのスケールを決定する。図 2.9 のように 3 枚 1 組の DoG 画像から極値を検出する。図 2.9 の赤の破線で囲まれた DoG 画像の注目ピクセル (図 2.9



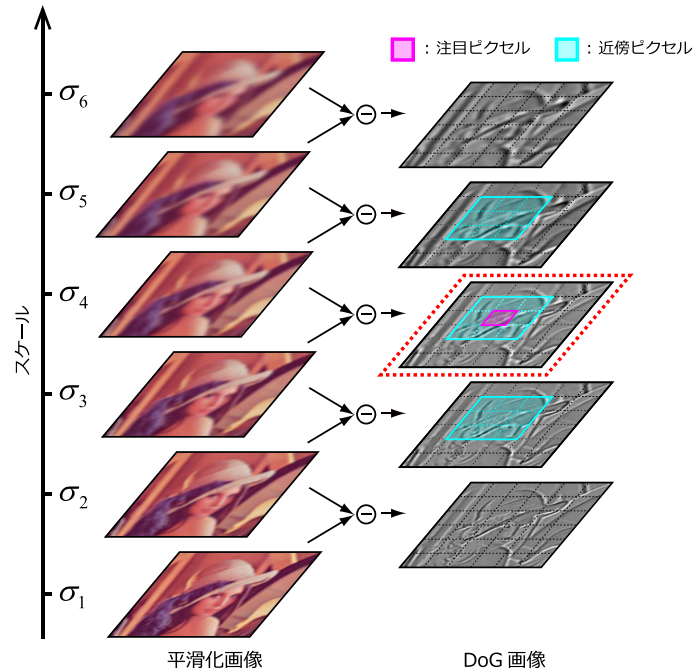


図 2.9: DoG 画像からの極値検出.

マゼンタのピクセル) と、その  $[x, y, \sigma]$  の 3 次元空間における 26 近傍 (図 2.9 シアンのピクセル) の値を比較し、注目ピクセルが極値であった場合に、このピクセルをキーポイント候補として検出する。このようにして、 $[x, y]$  スペースとスケールスペースの両方を考慮したキーポイント候補を検出することが可能となる。極値検出は、 $\sigma$  の小さい DoG 画像から検出し、一度極値が検出されたピクセルは、以降の大きなスケールでは極値探索しない。この処理をスケールの異なる全ての DoG 画像に対して行う。

### ■ エッジ上のキーポイント候補の削除

DoG 画像の極値探索により検出したキーポイント候補の中には、画像のエッジ上に検出されたキーポイント候補が含まれており、キーポイントマッチングの際に開口問題の影響を受けやすい。そこで、キーポイント候補の中からエッジ上に存在するキーポイント候補を削除する。

まず、キーポイント候補における 2 次元 Hessian 行列  $\mathbf{H}_{DoG}$  を次式により計算する。

$$\mathbf{H}_{DoG} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.25)$$

行列内の導関数は、キーポイント候補位置での DoG 出力値の 2 次微分から得られる。ここで、Hessian 行列  $\mathbf{H}_{DoG}$  から求められる第 1 固有値を  $\lambda_\alpha$ 、第 2 固有値を  $\lambda_\beta (\lambda_\alpha > \lambda_\beta)$  とする。このとき、Hessian

行列の対角成分の和  $\text{trace}(\mathbf{H}_{DoG})$  と行列式  $\det(\mathbf{H}_{DoG})$  は次のように計算できる.

$$\text{trace}(\mathbf{H}_{DoG}) = D_{xx} + D_{yy} = \lambda_\alpha + \lambda_\beta \quad (2.26)$$

$$\det(\mathbf{H}_{DoG}) = D_{xx}D_{yy} - D_{xy}^2 = \lambda_\alpha\lambda_\beta \quad (2.27)$$

第 1 固有値と第 2 固有値の比率を  $\gamma$  とし,  $\lambda_\alpha = \gamma\lambda_\beta$  と表記すると次式が得られる.

$$\frac{\text{trace}^2(\mathbf{H}_{DoG})}{\det(\mathbf{H}_{DoG})} = \frac{(\lambda_\alpha + \lambda_\beta)^2}{\lambda_\alpha\lambda_\beta} = \frac{(\gamma\lambda_\beta + \lambda_\beta)^2}{\gamma\lambda_\beta^2} = \frac{(\gamma + 1)^2}{\gamma} \quad (2.28)$$

$\text{trace}^2(\mathbf{H}_{DoG})$  と  $\det(\mathbf{H}_{DoG})$  の比率の閾値を  $\gamma_{th}$  とすると, 次式に示すように  $\text{trace}^2(\mathbf{H}_{DoG})$  と  $\det(\mathbf{H}_{DoG})$  の比率が閾値未満の場合, 第 1 固有値と第 2 固有値の比率が小さいと判定され, キーポイント候補として残す.

$$\frac{\text{trace}^2(\mathbf{H}_{DoG})}{\det(\mathbf{H}_{DoG})} < \frac{(\gamma_{th} + 1)^2}{\gamma_{th}} \quad (2.29)$$

この処理は, 2.2.1 項で述べた Harris コーナー検出器のコーナー判定に類似しており, 実際に行列  $\mathbf{H}_{DoG}$  の固有値問題を解く必要はない. 文献 [1] では  $\gamma_{th} = 10$  を採用しており, 式 (2.29) の右辺は 12.1 となる.

### ■ キーポイントのサブピクセル位置推定

DoG の出力値を  $[x, y, \sigma]$  の 3 変数の 2 次関数フィッティングにより, キーポイントのサブピクセル位置とスケールの補正が可能となる. キーポイントの座標とスケール  $\mathbf{q} = [x, y, \sigma]^\top$  での DoG 関数  $D(\mathbf{q})$  を 2 次のテイラー展開で近似すると次式のように  $D_{apx}(\mathbf{q})$  が得られる.

$$D_{apx}(\mathbf{q}) \approx D + \frac{\partial D^\top}{\partial \mathbf{q}} \mathbf{q} + \frac{1}{2} \mathbf{q}^\top \frac{\partial^2 D}{\partial \mathbf{q}^2} \mathbf{q} = D + \frac{\partial D^\top}{\partial \mathbf{q}} \mathbf{q} + \frac{1}{2} \frac{\partial^2 D}{\partial \mathbf{q}^2} \mathbf{q}^2 \quad (2.30)$$

式 (2.30) において  $\mathbf{q}$  に関する偏導関数が 0 となるような  $\hat{\mathbf{q}} = [\hat{x}, \hat{y}, \hat{\sigma}]^\top$  が, 正確な位置とスケールにおける極値である.

$$\frac{\partial D_{apx}}{\partial \mathbf{q}} = \frac{\partial D}{\partial \mathbf{q}} + \frac{\partial^2 D}{\partial \mathbf{q}^2} \mathbf{q} = 0 \quad (2.31)$$

$$\frac{\partial^2 D}{\partial \mathbf{q}^2} \mathbf{q} = -\frac{\partial D}{\partial \mathbf{q}} \quad (2.32)$$

$$\hat{\mathbf{q}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{q}^2} \frac{\partial D}{\partial \mathbf{q}} \quad (2.33)$$

このとき  $\hat{\mathbf{q}} = [\hat{x}, \hat{y}, \hat{\sigma}]^\top$  はサブピクセル位置を表しており, 式 (2.33) を行列で表記すると次式となる.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\sigma} \end{bmatrix} = - \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial x\sigma} \\ \frac{\partial^2 D}{\partial xy} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y\sigma} \\ \frac{\partial^2 D}{\partial x\sigma} & \frac{\partial^2 D}{\partial y\sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{bmatrix} \quad (2.34)$$

式 (2.34) を解くことにより，キーポイント候補のサブピクセルと正確なスケールを推定することができる．よって，サブピクセル位置推定は位置の補正のみではなく，スケールの補正に対しても有効である．

### ■ 低コントラストのキーポイント候補の削除

極値探索では微小な極値を捉えてしまうため，キーポイント候補に DoG のスコアが低い (= 低いコントラスト) キーポイントが多く含まれている．低コントラストのキーポイントはノイズの影響を受けやすいため，このようなキーポイントを削除する．サブピクセル位置における DoG のスコア  $D(\hat{\mathbf{q}})$  は，式 (2.33) を式 (2.30) へ代入することで次式のように計算することができる．

$$\begin{aligned}
 D(\hat{\mathbf{q}}) &= D + \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} + \frac{1}{2} \hat{\mathbf{q}}^\top \frac{\partial^2 D}{\partial \mathbf{q}^2} \hat{\mathbf{q}} \\
 &= D + \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} + \frac{1}{2} \left( -\frac{\partial^2 D^{-1}}{\partial \mathbf{q}^2} \frac{\partial D}{\partial \mathbf{q}} \right)^\top \frac{\partial^2 D}{\partial \mathbf{q}^2} \hat{\mathbf{q}} \\
 &= D + \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} - \frac{1}{2} \frac{\partial D^\top}{\partial \mathbf{q}} \frac{\partial^2 D^{-1}}{\partial \mathbf{q}^2} \frac{\partial^2 D}{\partial \mathbf{q}^2} \hat{\mathbf{q}} \\
 &= D + \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} - \frac{1}{2} \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} \\
 &= D + \left( 1 - \frac{1}{2} \right) \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} \\
 &= D + \frac{1}{2} \frac{\partial D^\top}{\partial \mathbf{q}} \hat{\mathbf{q}} \tag{2.35}
 \end{aligned}$$

式 (2.35) によりサブピクセル位置における DoG のスコアが計算できる．サブピクセルにおけるスコアの絶対値を  $|D(\hat{\mathbf{q}})| \in [0, 1]$  となるように正規化した後，閾値で処理することによりコントラストが低いキーポイント，すなわち DoG スコアの低いキーポイントを削除する．文献 [1] では，低コントラストのキーポイントを削除する閾値を 0.03 に設定している．

### ■ オリエンテーションの算出

SIFT では，画像の回転に対して不変な特徴量を記述するために各キーポイント位置における主要な方向であるオリエンテーションを算出する．オリエンテーションを算出するには図 2.10 に示すように，キーポイントを中心とする平滑化画像  $L(\mathbf{p}; \hat{\sigma})$  から勾配強度  $m(\mathbf{p})$  と勾配方向  $o(\mathbf{p})$  をキーポイントのスケール  $\hat{\sigma}$  の範囲から求める．

$$m(\mathbf{p}) = \sqrt{g_x(\mathbf{p})^2 + g_y(\mathbf{p})^2} \tag{2.36}$$

$$o(\mathbf{p}) = \tan^{-1} \left( \frac{g_y(\mathbf{p})}{g_x(\mathbf{p})} \right) \tag{2.37}$$

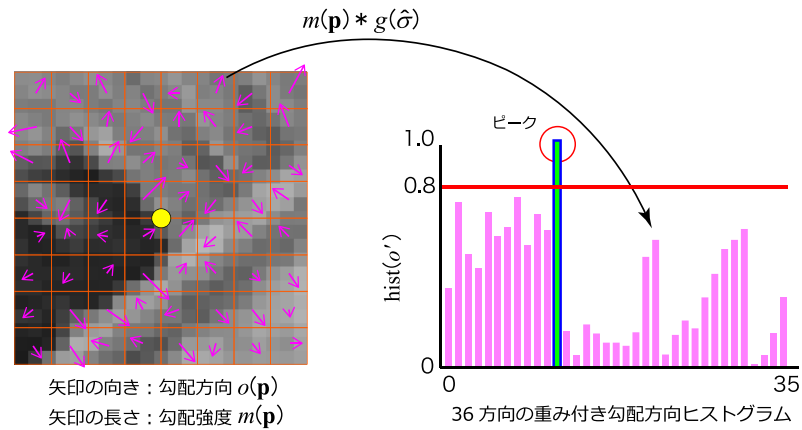


図 2.10: SIFT のオリエンテーション算出.

$$\begin{cases} g_x(\mathbf{p}) = L(x+1, y; \hat{\sigma}) - L(x-1, y; \hat{\sigma}) \\ g_y(\mathbf{p}) = L(x, y+1; \hat{\sigma}) - L(x, y-1; \hat{\sigma}) \end{cases} \quad (2.38)$$

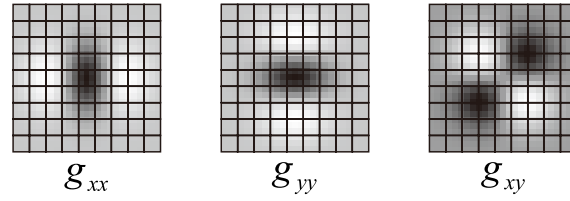
スケール領域における勾配強度  $m(\mathbf{p})$  と勾配方向  $o(\mathbf{p})$  から、重み付き勾配ヒストグラム  $\text{hist}(o')$  を次式より求める.

$$\text{hist}(o') = \sum_x \sum_y g(\hat{\sigma}) * m(\mathbf{p}) \cdot \delta[o', o(\mathbf{p})] \quad (2.39)$$

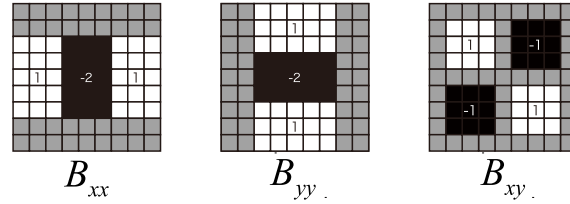
$\text{hist}(o')$  は勾配方向を 36 方向に量子化したヒストグラムであり、キーポイントのスケール  $\hat{\sigma}$  のガウス関数  $g(\hat{\sigma})$  により重み付けした勾配強度を投票する. ガウス関数による重み付けにより、キーポイントに近い勾配強度に大きな重みが与えられる.  $\delta[\cdot]$  は Kronecker のデルタ関数であり、勾配方向  $o(\mathbf{p})$  を量子化した際に、量子化勾配方向  $o'$  に該当する場合に 1 を返す. この重み付き勾配方向ヒストグラムの最大値 (ピーク) から 80% 以上となる勾配方向のビンを全てキーポイントのオリエンテーション  $\hat{\theta}$  として割り当てる. よって、コーナーのような位置から検出されたキーポイントには 2 方向以上のオリエンテーションが割り当てられる. 特徴量記述の際には、各方向に対してそれぞれ特徴量が記述される. さらに、SIFT では勾配方向ヒストグラムに対して 2 次関数の多項式フィッティングを適用することで、オリエンテーションを連続値として算出する. この処理により、正確なオリエンテーションの算出が可能となる.

### 2.3.3 Speeded-Up Robust Features (SURF) Detector

Speeded-Up Robust Features (SURF) [18] も画像の回転とスケール変化に不変なキーポイントを検出することができ、SIFT と同様にキーポイント検出と特徴量記述の 2 段階のアルゴリズムで構成されている. ここでは、SURF のキーポイント検出について説明する.



(a) 2次微分ガウシアンフィルタ



(b) Box フィルタによる近似

図 2.11: Box フィルタによる 2 次微分ガウシアンフィルタの近似.

### ■ Box フィルタによる Hessian の近似

SURF によるキーポイント検出では、Hessian-Laplace の処理に基づいてキーポイントを検出する。しかし、Hessian-Laplace ではスケールスペースにおいて 2 次微分ガウシアンフィルタを畳み込んで極値を検出するため計算コストが高くなる。そこで、SURF は 2 次微分ガウシアンフィルタをシンプルな Box フィルタで近似させることで、キーポイント検出の処理を高速化させている。Box フィルタは積分画像と組み合わせることで、高速にフィルタリングすることが可能となる。図 2.11 に示すように、2 次微分ガウシアンフィルタ  $g_{xx}$ ,  $g_{yy}$ ,  $g_{xy}$  を Box フィルタ  $B_{xx}$ ,  $B_{yy}$ ,  $B_{xy}$  で近似することができる。Box フィルタによって計算される画像の 2 次微分をそれぞれ  $\mathcal{I}_{xx}(\mathbf{p}; \sigma)$ ,  $\mathcal{I}_{yy}(\mathbf{p}; \sigma)$ ,  $\mathcal{I}_{xy}(\mathbf{p}; \sigma)$  とするとき、近似 Hessian 行列  $\mathcal{H}_{apx}$  は次式のように計算できる。

$$\mathcal{H}_{apx} = \begin{bmatrix} \mathcal{I}_{xx}(\mathbf{p}; \sigma) & \mathcal{I}_{xy}(\mathbf{p}; \sigma) \\ \mathcal{I}_{xy}(\mathbf{p}; \sigma) & \mathcal{I}_{yy}(\mathbf{p}; \sigma) \end{bmatrix} \quad (2.40)$$

$$\mathcal{I}_{xx}(\mathbf{p}; \sigma) = B_{xx}(\sigma) * I(\mathbf{p}) \approx g_{xx}(\sigma) * I(\mathbf{p}) \quad (2.41)$$

$$\mathcal{I}_{yy}(\mathbf{p}; \sigma) = B_{yy}(\sigma) * I(\mathbf{p}) \approx g_{yy}(\sigma) * I(\mathbf{p}) \quad (2.42)$$

$$\mathcal{I}_{xy}(\mathbf{p}; \sigma) = B_{xy}(\sigma) * I(\mathbf{p}) \approx g_{xy}(\sigma) * I(\mathbf{p}) \quad (2.43)$$

Box フィルタにおけるスケール  $\sigma$  はフィルタサイズを  $\sigma$  に応じて変化させる。例えば、ガウシアンフィルタのスケールが  $\sigma = \{1.2, 2.0, 2.8, 3.6\}$  と変化する場合、Box フィルタのサイズは  $\{9 \times 9, 15 \times 15, 21 \times 21, 27 \times 27\}$  のように変化する。

Hessian によるキーポイント検出と同様に、近似 Hessian 行列  $\mathcal{H}_{apx}$  の行列式  $\det(\mathcal{H}_{apx})$  を計算することでキーポイントのスコアを求める。

$$\det(\mathcal{H}_{apx}) = \mathcal{I}_{xx}(\mathbf{p}; \sigma) \cdot \mathcal{I}_{yy}(\mathbf{p}; \sigma) - (0.9 \cdot \mathcal{I}_{xy}(\mathbf{p}; \sigma))^2 \quad (2.44)$$

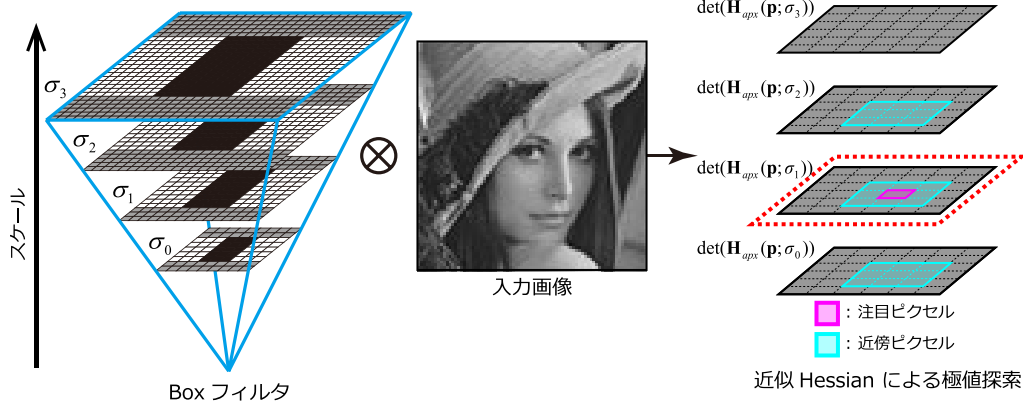


図 2.12: Box フィルタを利用した極値探索.

ここで,  $\mathcal{I}_{xy}(\mathbf{p}; \sigma)$  に 0.9 の重み付けがされているが, これは行列式  $\det(\mathcal{H}_{apx})$  をバランスよく釣り合わせるための相対的な重み係数である. 2 次微分ガウシアンフィルタ  $g_{xx}$ ,  $g_{xy}$  により計算した 2 次微分画像を  $\mathcal{I}_{xx}$ ,  $\mathcal{I}_{xy}$  とすると, 次式に示すような 2 次微分画像のフロベニウスノルムの関係が得られる.

$$\frac{\|\mathcal{I}_{xy}(\cdot; 1.2)\|_F \cdot \|\mathcal{I}_{xx}(\cdot; 9)\|_F}{\|\mathcal{I}_{xx}(\cdot; 1.2)\|_F \cdot \|\mathcal{I}_{xy}(\cdot; 9)\|_F} = 0.912 \dots \simeq 0.9 \quad (2.45)$$

フィルタの計算結果はサイズに関して正規化され, 任意のフィルタサイズに対して一定のフロベニウスノルムが保証されるため重み係数を 0.9 と定めている.

### ■ Box フィルタのスケールスペースによる極値探索

SURF におけるスケール推定は Box フィルタのサイズを変化させることで, スケールスペースにおけるキーポイントスコア (近似 Hessian の行列式) を計算する. 図 2.12 に示すように, Box フィルタによるスケールスペースでスコアを計算した後, SIFT と同様に 26 近傍のピクセルと比較して極値を探索する. 注目ピクセルが極値であった場合に, その位置の座標と Box フィルタのサイズをキーポイントとスケールとして検出する.

### ■ オリエンテーションの算出

SURF のオリエンテーション算出は, 図 2.13 に示すようにキーポイントを中心とした  $6\hat{\sigma}$  の領域から  $x, y$  方向の Haar-wavelet ( $4\hat{\sigma} \times 4\hat{\sigma}$ ) を計算する.  $\hat{\sigma}$  はキーポイントのスケールである. 計算された  $6\hat{\sigma}$  の領域内の輝度勾配をキーポイントを中心として  $\frac{\pi}{3}$  ずつ回転させながら  $x, y$  方向毎に総和を求める. 計算された輝度勾配の総和が最も大きい方向をキーポイントのオリエンテーション  $\hat{\theta}$  として採用する.

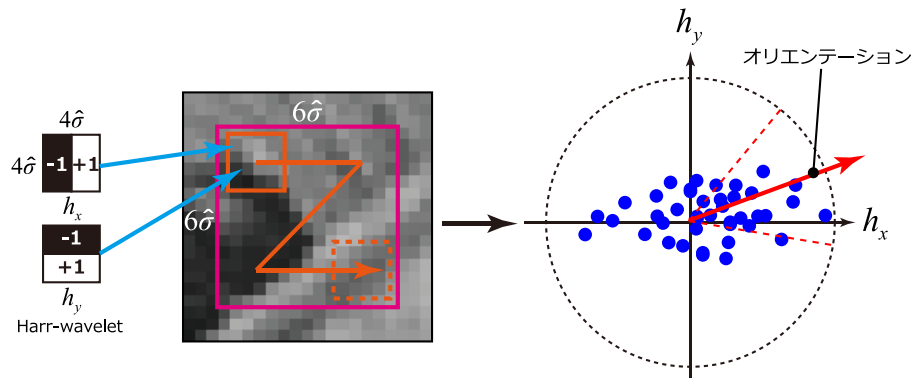


図 2.13: SURF のオリエンテーション算出.

### 2.3.4 Oriented FAST and Rotated BRIEF (ORB) Detector

Oriented FAST and Rotated BRIEF (ORB) は、画像ピラミッドと FAST コーナー検出器を組み合わせることでスケール変化に対応しつつ高速なキーポイント検出を実現している。また、サンプリングピクセルペアの輝度差に基づいた高速かつ省メモリな 2 値特徴量記述も提案している。ここでは、ORB におけるキーポイント検出について述べる。ORB ではキーポイントを高速に検出するために、2.2.3 項で述べた FAST コーナー検出器を使用している。FAST コーナー検出器は高速な処理が可能である一方で、キーポイントのスケールやオリエンテーションを算出しないため、キーポイントマッチング時にはスケール変化や回転に対して不変性が得られない問題がある。そこで、ORB は FAST コーナー検出器を用いてスケールとオリエンテーションを算出する。

#### ■ 画像ピラミッドによるスケール獲得

まず、スケールの不変性を得るために入力画像を多段階にダウンサンプリングした画像ピラミッドを生成する。画像ピラミッドは図 2.14 に示すように入力画像を  $\frac{1}{\sqrt{2}}$  倍ずつダウンサンプリングして生成する。生成した画像ピラミッドの各階層の画像から FAST コーナー検出器によりコーナーを検出する。コーナーが検出された画像の倍率の逆数をそのままスケール  $\hat{\sigma}$  として採用する。また、検出されたコーナー点に対して式 (2.9) に示すような Harris コーナー検出器のスコアを計算する。このスコアが閾値以上のコーナーのみがキーポイントとして検出される。

#### ■ オリエンテーションの算出

オリエンテーションの算出には、画像ピラミッドで求めたスケール範囲のパッチ画像の輝度値から 0, 1 モーメント  $\tilde{m}_{uv}$  ( $u, v = \{0, 1\}$ ) を求める。

$$\tilde{m}_{uv} = \sum_{x,y} x^u y^v I(x, y) \quad (2.46)$$

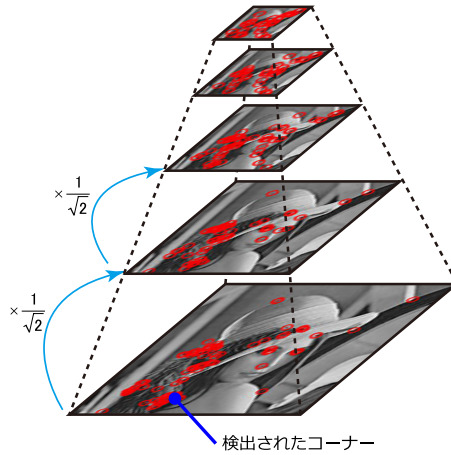


図 2.14: 画像ピラミッドによるスケール獲得.

式 (2.46) から算出したモーメントからパッチ画像の重心位置  $C$  を求める.

$$C = \begin{pmatrix} \tilde{m}_{10} & \tilde{m}_{01} \\ \tilde{m}_{00} & \tilde{m}_{00} \end{pmatrix} \quad (2.47)$$

キーポイント位置からパッチ画像の重心位置までの方向ベクトルがオリエンテーション  $\hat{\theta}$  となり, これは次式によりシンプルに求めることができる.

$$\hat{\theta} = \tan^{-1} \left( \frac{\tilde{m}_{01}}{\tilde{m}_{10}} \right) \quad (2.48)$$

### 2.3.5 Spectral SIFT

Spectral SIFT [20] はガウシアンスケールスペースや LoG スケールスペースに対してスペクトル分解を適用することで, スケールスペースを圧縮させ, 効率的なスケール推定を行う. さらに, スケールスペースのパラメータを連続的な多項式で近似することで高精度なスケールを推定することができる. この項では, 無限次元に拡張させたガウシアンスケールスペースや LoG スケールスペースをスペクトル分解するため, 画像等の畳み込み演算を連続的な積分方程式で表記する.

#### ■ ガウシアンスケールスペースの圧縮

入力画像  $I(x, y)$  に対してガウス関数  $g(x, y; \sigma)$  を畳み込むことで平滑化画像  $L(x', y'; \sigma)$  が得られる.

$$L(x', y'; \sigma) = \int \int g(x, y; \sigma) I(x - x', y - y') dx dy \quad (2.49)$$



ガウス関数のスケール範囲が  $\sigma_1 \leq \sigma \leq \sigma_2$  のように与えられている場合、固有関数  $\varphi_i(\cdot)$  によりスペクトル分解を行うことができる。

$$g(x, y; \sigma) = \sum_{i=0}^{\infty} \left( \int_{\sigma_1}^{\sigma_2} g(x, y; \tau) \varphi_i(\tau) d\tau \right) \varphi_i(\sigma) \quad (2.50)$$

式 (2.50) は無限級数であるため、これを  $N_c$  項で近似する。

$$g(x, y; \sigma) \approx \sum_{i=0}^{N_c} \left( \int_{\sigma_1}^{\sigma_2} g(x, y; \tau) \varphi_i(\tau) d\tau \right) \varphi_i(\sigma) \quad (2.51)$$

式 (2.51) を式 (2.49) へ代入すると次式が得られる。

$$L(x', y'; \sigma) \approx \int \int \sum_{i=0}^{N_c} \left( \int_{\sigma_1}^{\sigma_2} g(x, y; \tau) \varphi_i(\tau) d\tau \right) \varphi_i(\sigma) \cdot I(x - x', y - y') dx dy \quad (2.52)$$

ここで、 $dx dy$  と  $d\tau$  の積分の順序を入れ替えることで次式に展開できる。

$$\begin{aligned} L(x', y'; \sigma) &\approx \sum_{i=0}^{N_c} \left( \int \int \left( \int_{\sigma_1}^{\sigma_2} g(x, y; \tau) \varphi_i(\tau) d\tau \right) \cdot I(x - x', y - y') dx dy \right) \varphi_i(\sigma) \\ &= \sum_{i=0}^{N_c} \varphi_i(\sigma) \cdot \left( \int \int F_i(x, y) I(x - x', y - y') dx dy \right) \\ &= \sum_{i=0}^{N_c} \varphi_i(\sigma) \eta_i(x', y') \end{aligned} \quad (2.53)$$

ここで、 $F_i(x, y)$  は次式のように定義でき、2次元の画像と考えることができるため固有画像と呼ぶ。

$$F_i(x, y) = \int_{\sigma_1}^{\sigma_2} g(x, y; \tau) \varphi_i(\tau) d\tau \quad (2.54)$$

式 (2.53) より、スケール  $\sigma$  のガウス関数による平滑化画像  $L(x, y; \sigma)$  は、入力画像と  $N_c$  枚の固有画像  $F_i(x, y)$  との畳み込み結果  $\eta_i(x', y')$  と固有関数  $\varphi_i(\sigma)$  との積で求めていることが確認できる。

次に固有関数  $\varphi_i(\sigma)$  を積分型の固有値問題により求める。

$$\int_{\sigma_1}^{\sigma_2} K(\tau, \sigma) \varphi(\tau) d\tau = \lambda_s \varphi(\sigma) \quad (2.55)$$

$K(\tau, \sigma)$  は2次モーメントにより定義される積分核であり、次式で求められる。

$$\begin{aligned} K(\tau, \sigma) &= \int \int g(x, y; \tau) g(x, y; \sigma) dx dy \\ &= \frac{1}{2\pi(\tau^2 + \sigma^2)} \end{aligned} \quad (2.56)$$

しかし、式 (2.55) の厳密な解を求めることができないため、固有関数を  $N_c$  次の多項式で近似して解く。

$$\begin{aligned}\varphi_i(\sigma) &= a_{i,0} + a_{i,1}\sigma + a_{i,2}\sigma^2 + \cdots + a_{i,N_c}\sigma^{N_c} \\ &= \begin{bmatrix} 1 & \sigma & \sigma^2 & \cdots & \sigma^{N_c} \end{bmatrix} \mathbf{a}_i\end{aligned}\quad (2.57)$$

固有関数を多項式で近似すると式 (2.55) は  $N_c + 1 \times N_c + 1$  行列の一般化固有値問題に帰着する。

$$\mathbf{K}\mathbf{a} = \lambda_s \mathbf{T}\mathbf{a} \quad (2.58)$$

ここで、行列  $\mathbf{K}$ 、 $\mathbf{T}$  の要素は次式で与えられる。

$$K_{i+1,j+1} = \frac{1}{2\pi} \int \int \frac{\sigma^j \tau^i}{\sigma^2 + \tau^2} d\sigma d\tau \quad (2.59)$$

$$T_{I+1,j+1} = \int \sigma^{i+j} d\sigma = \frac{\sigma^{1+i+j}}{1+i+j} \quad (2.60)$$

式 (2.58) の固有値問題を解くことで、 $N_c + 1$  個の固有値  $\lambda_{s_i}$  と固有ベクトル  $\mathbf{a}_i$  が求められ、これを式 (2.57) へ代入することで多項式で表現された固有関数  $\varphi_i(\sigma)$  を求めることができる。

実際にガウシアンスケールスペース  $\sigma_1 = 1.0, \sigma_2 = 5.0$  の固有関数を 2 次の多項式 ( $N_c = 2$ ) で近似すると、固有値問題から得られる固有値は次数 2 以降で極めて 0 に近い値となる ( $\lambda_{s_0} = 0.0550, \lambda_{s_1} = 0.0070, \lambda_{s_2} = 0.0005$ )。よって、少ない次数でガウシアンスケールスペースを近似することができる。

### ■ Scale Normalized LoG スペースの圧縮

Scale normalized LoG (sLoG) スペースもガウシアンスケールスペースと同様に固有解を求めることができる。sLoG はガウス関数の  $x$  方向に関する 2 次微分と  $y$  方向に関する 2 次微分の和に  $\sigma^2$  をかけることで計算できる。sLoG を畳み込むことで得られる画像  $\mathcal{L}(x', y'; \sigma)$  は次式により求められる。

$$\mathcal{L}(x', y'; \sigma) = \int \int \sigma^2 \nabla^2 g(x, y; \sigma) I(x - x', y - y') dx dy \quad (2.61)$$

$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  であり、拡散方程式の関係から次式が成り立つ。

$$\sigma \nabla^2 g(x, y; \sigma) = \frac{\partial g(x, y; \sigma)}{\partial \sigma} \quad (2.62)$$

式 (2.62) を式 (2.61) に代入すると次式となる。

$$\sigma \nabla^2 g(x, y; \sigma) = \int \int \frac{\partial g(x, y; \sigma)}{\partial \sigma} I(x - x', y - y') dx dy \quad (2.63)$$

ガウシアンスケールスペースの圧縮と同様に固有関数により展開すると、sLoG の積分核は次式となる。

$$\begin{aligned}\mathcal{K}(\sigma, \tau) &= \int \int \sigma \tau \frac{\partial g(x, y; \sigma)}{\partial \sigma} \frac{\partial g(x, y; \tau)}{\partial \tau} dx dy \\ &= \frac{4\sigma^2 \tau^2}{\pi(\sigma^2 + \tau^2)^3}\end{aligned}\quad (2.64)$$

固有関数を多項式で近似することで、式 (2.58) のように一般化固有値問題に帰着させることができる。行列  $\mathcal{K}$ ,  $T$  の要素は次式で求めることができる。

$$\mathcal{K}_{i+1, j+1} = \frac{4}{\pi} \int \int \frac{\sigma^{j+2} \tau^{i+2}}{(\sigma^2 + \tau^2)^3} d\sigma d\tau \quad (2.65)$$

$$T_{i+1, j+1} = \int \sigma^{i+j} d\sigma = \frac{\sigma^{1+i+j}}{1+i+j} \quad (2.66)$$

実際に sLoG スケールスペース  $\sigma_1 = 1.0, \sigma_2 = 2.0$  の固有関数を 3 次の多項式 ( $N_c = 3$ ) で近似すると、固有値問題から得られる固有値は次数 3 以降で 0 に近い値となる ( $\lambda_{s_0} = 0.05513$ ,  $\lambda_{s_1} = 0.00741$ ,  $\lambda_{s_2} = 0.00083$ ,  $\lambda_{s_3} = 0.00004$ )。よって、sLoG スケールスペースにおいても少ない次数で近似することができる。

### ■ Spectral SIFT によるキーポイント検出

2.3.5 項と 2.3.5 項からガウシアンスケールスペースや sLoG スケールスペースが少ない次数で近似できることが確認できたため、このスケールスペースの圧縮を SIFT のキーポイント検出へ応用する。SIFT のキーポイント検出は sLoG  $\approx$  DoG として極値を求めているため、sLoG によるキーポイント検出について述べる。sLoG のスコアは  $N_c = 3$  の 3 次の多項式により次式で表現される。

$$\mathcal{L}(x', y'; \sigma) \approx \sum_{i=0}^3 \eta_i (a_{i,0} + a_{i,1}\sigma + a_{i,2}\sigma^2 + a_{i,3}\sigma^3) \quad (2.67)$$

ここで、sLoG のスコア計算は連続的な多項式で表現されているため微分可能である。よって、極値の位置は  $\sigma$  について偏微分して 0 となる位置を見つけことになる。

$$\frac{\partial \mathcal{L}(x', y', \sigma)}{\partial \sigma} = \sum_{i=0}^3 \eta_i (a_{i,1} + 2a_{i,2}\sigma + 3a_{i,3}\sigma^2) = 0 \quad (2.68)$$

$$\begin{aligned}\equiv & a_e \sigma^2 + b_e \sigma + c_e = 0 \\ \sigma &= \frac{-b_e \pm \sqrt{b_e^2 - 4a_e c_e}}{2a_e}\end{aligned}\quad (2.69)$$

すると、式 (2.68) は 2 次方程式の形となるため、式 (2.69) のように解の公式により極値位置を容易に求めることができる。また、キーポイントを検出する際にはスケールの極値となる位置が  $x, y$  方向に対しても極値であるかを判定する。

## 2.4 アフィン領域の推定

キーポイント検出におけるアフィン領域推定は、スケール推定の一般化と捉えることができる。スケール推定はキーポイントに対して等方性の領域を求めるため、画像間の拡大・縮小には不変な領域が得られるが、射影変化に対しては正確な領域推定ができない。そこで、キーポイントに対して楕円状のアフィン領域を推定することで、アフィン不変な領域を求めることができる。本来、射影変換等是非線形な歪みであるが、キーポイントにおける局所領域においては線形なアフィン変換で近似することができるため、アフィン領域を推定し、その領域内から局所特徴量を記述することで様々な視点変化に頑健なキーポイントマッチングが可能となる。ここでは、これまでに提案されているキーポイントのアフィン領域推定手法を述べる。

### 2.4.1 Harris-Affine と Hessian-Affine

Harris-Affine と Hessian-Affine[24] は、それぞれ Harris-Laplace と Hessian-Laplace で検出したキーポイントの等方性スケールをアフィン領域へ拡張する手法である。Harris-Affine と Hessian-Affine にはパッチ画像変形に基づくアプローチと非等方性ガウシアンフィルタに基づくアプローチがある。以下に各アプローチについて述べる。

#### ■ パッチ画像変形に基づくアプローチ

パッチ画像変形に基づくアプローチは検出されたキーポイントの等方性スケール領域をパッチ画像として抽出する。パッチ画像内の勾配の分布に従ってパッチ画像を繰り返し変形させていくことでアフィン領域を推定する。以下にパッチ画像変形に基づくアプローチの処理過程を示す。

1. Harris-Laplace または Hessian-Laplace によりキーポイントと等方性スケール  $\hat{\sigma}$  を検出。
2. 検出したスケール領域をパッチ画像として抽出し、変形行列  $U^{(i)}$  によりパッチ画像を変形。
3. パッチ画像の 2 次モーメント行列  $\mu$  を算出し、行列  $\mu$  の固有値  $\Lambda_e$  と固有ベクトル  $\Gamma_e$  から楕円領域を推定。
4.  $\Gamma_e^T \Lambda_e \Gamma_e$  により 2 次モーメント行列  $\mu$  を更新。
5.  $U^{(i+1)} = U^{(i)} \mu$  により変形行列を更新。
6.  $1 - \frac{\lambda_{\min}(\Lambda_e)}{\lambda_{\max}(\Lambda_e)} < \varepsilon$  を満たす場合、行列  $U^{(i)}$  をアフィン領域パラメータとする。条件を満たさない場合は 2~5 の処理を繰り返す。

まず、Harris-Laplace または Hessian-Laplace によりキーポイントと等方性スケールを検出する。そして、検出されたスケール範囲をパッチ画像として抽出し、パッチ画像を変形行列  $U^{(i)}$  により変形させる。変形行列  $U^{(i)}$  の初期値には単位行列が与えられる。

次に、パッチ画像から 2 次モーメント行列  $\mu$  を計算する。2 次モーメント行列は、2.2.1 項で述べた Harris コーナー検出と同様の方法で計算する (式 (2.8))。行列  $\mu$  の固有値  $\Lambda_e$  と固有ベクトル  $\Gamma_e$  を

求めることにより楕円領域を推定する。行列  $\boldsymbol{\mu}$  は対称行列であるため、固有値から楕円の長径と短径の大きさが計算でき、固有ベクトルから楕円の長径と短径の方向を計算することができる。そして、行列  $\boldsymbol{\mu}$  の各要素を  $\boldsymbol{\Gamma}_e^\top \boldsymbol{\Lambda}_e \boldsymbol{\Gamma}_e$  に置き換える。  $\boldsymbol{\Lambda}_e$  と  $\boldsymbol{\Gamma}_e$  は行列  $\boldsymbol{\mu}$  の固有値分解で得られる固有値と固有ベクトルである。各要素を更新した行列  $\boldsymbol{\mu}$  を用いて変形行列  $\mathcal{U}^{(i)}$  を次式のように更新する。

$$\mathcal{U}^{(i+1)} = \mathcal{U}^{(i)} \boldsymbol{\mu} \quad (2.70)$$

2次モーメント行列の固有値の比率が式 (2.71) の条件式を満たす場合、変形行列  $\mathcal{U}^{(i)}$  をアフィン領域のパラメータとして採用し、処理を終了する。条件を満たさない場合は  $\mathcal{U}^{(i+1)}$  により、もう一度パッチ画像を変形し、2次モーメント行列を再計算する。

$$1 - \frac{\lambda_{\min}(\boldsymbol{\Lambda}_e)}{\lambda_{\max}(\boldsymbol{\Lambda}_e)} < \varepsilon \quad (2.71)$$

ここで、 $\lambda_{\max}(\boldsymbol{\Lambda}_e)$  と  $\lambda_{\min}(\boldsymbol{\Lambda}_e)$  はそれぞれ行列  $\boldsymbol{\mu}$  の最大固有値と最小固有値である。また、 $\varepsilon$  は閾値であり、文献 [24] では  $\varepsilon = 0.05$  として設定されている。図 2.15 にパッチ画像変形に基づくアプローチの処理過程を示す。変形行列によりパッチ画像の変形を繰り返すことで、2次モーメント行列から求められる楕円領域を正円に近い形、すなわち2次モーメント行列の固有値の比率が1に近くなる。

### ■ 非等方性ガウシアンフィルタに基づくアプローチ

非等方性ガウシアンフィルタに基づくアプローチは検出されたキーポイントと等方性スケール領域をパッチ画像として抽出する。パッチ画像に畳み込む非等方性ガウシアンフィルタを繰り返し変形していくことでアフィン領域を推定する。以下に非等方性ガウシアンフィルタに基づくアプローチの処理過程を示す。

1. Harris-Laplace または Hessian-Laplace によりキーポイントと等方性スケール  $\hat{\sigma}$  を検出。
2. 等方性スケール領域をパッチ画像として抽出し、 $\boldsymbol{\Sigma} = \mathcal{U}^{(i)} \hat{\sigma}$  として非等方性ガウシアンフィルタ  $g(\boldsymbol{\Sigma})$  を生成。
3. 生成した非等方性ガウシアンフィルタを用いてパッチ画像の2次モーメント行列  $\boldsymbol{\mu}$  を算出し、行列  $\boldsymbol{\mu}$  の固有値  $\boldsymbol{\Lambda}_e$  と固有ベクトル  $\boldsymbol{\Gamma}_e$  から楕円領域を推定。
4.  $\boldsymbol{\Gamma}_e^\top \boldsymbol{\Lambda}_e \boldsymbol{\Gamma}_e$  により2次モーメント行列  $\boldsymbol{\mu}$  を更新。
5.  $\mathcal{U}^{(i+1)} = \boldsymbol{\mu}$  により変形行列を更新。
6. 更新後の変形行列  $\mathcal{U}^{(i+1)}$  と更新前の変形行列  $\mathcal{U}^{(i)}$  の差が十分に小さい場合、 $\mathcal{U}^{(i)}$  をアフィン領域のパラメータとする。差が大きい場合は2~5を繰り返す。

まず、パッチ画像変形に基づくアプローチと同様に Harris-Laplace または Hessian-Laplace でキーポイントと等方性スケールを検出する。検出された等方性スケール範囲をパッチ画像として抽出し、

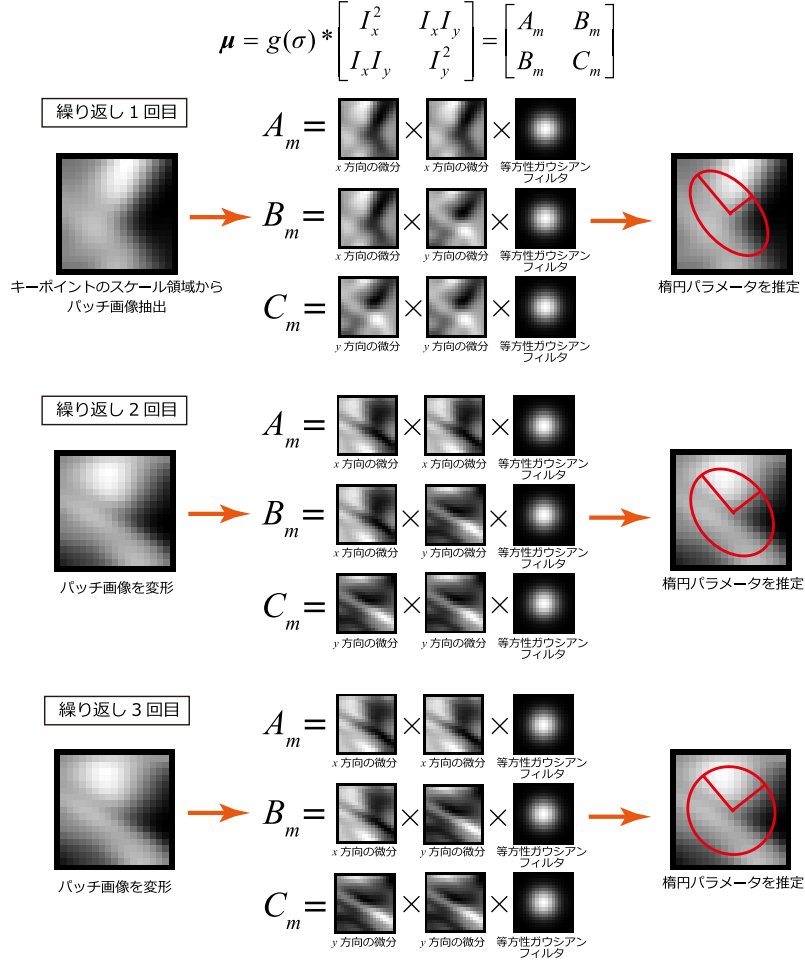


図 2.15: パッチ画像変形に基づくアプローチの処理過程.

$\Sigma = U^{(i)} \hat{\sigma}$  により非等方性ガウシアンフィルタ  $g(\Sigma)$  を生成する. 変形行列  $U^{(i)}$  の初期値には単位行列が与えられる.

次にパッチ画像内の 2 次モーメント行列  $\boldsymbol{\nu}$  を算出する.

$$\boldsymbol{\nu} = g(\Sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.72)$$

$$g(\Sigma) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{\bar{\mathbf{p}}^\top \Sigma^{-1} \bar{\mathbf{p}}}{2}\right) \quad (2.73)$$

ここで,  $I_x$  と  $I_y$  はそれぞれパッチ画像内の  $x$  方向の微分,  $y$  方向の微分であり,  $\bar{\mathbf{p}} = [\bar{x}, \bar{y}]$  はガウシアンフィルタの中心からの距離である. 行列  $\boldsymbol{\nu}$  の固有値  $\Lambda_e$  と固有ベクトル  $\Gamma_e$  から楕円領域を推定し, 行列  $\boldsymbol{\nu}$  の各要素を  $\Gamma_e^\top \Lambda_e \Gamma_e$  に置き換える. そして, 変形行列を  $U^{(i+1)} = \boldsymbol{\nu}$  により更新し, 更新後の変形行列  $U^{(i+1)}$  と更新前の変形行列  $U^{(i)}$  の差が十分に小さくなった場合, 変形行列  $U^{(i)}$  をアフィン領域のパラメータとして採用し, 処理を終了する. 行列間の差が大きい場合は,  $U^{(i+1)}$  によ

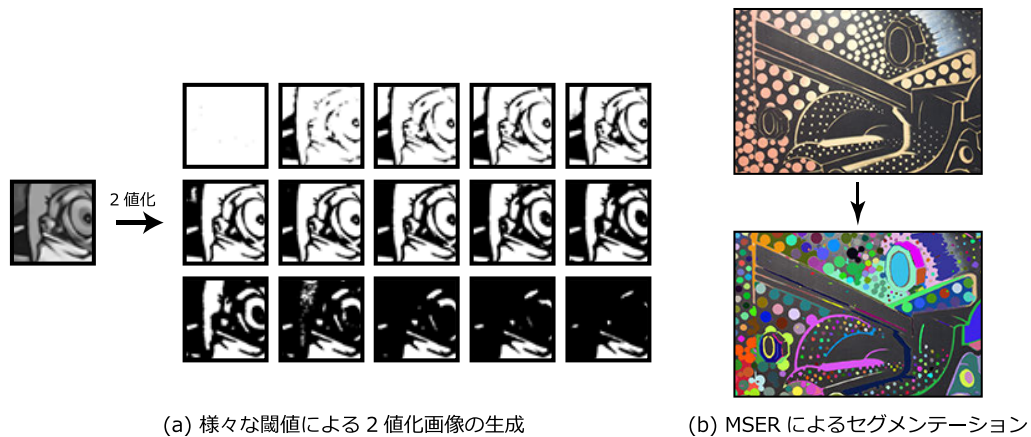


図 2.16: 画像の2値化によるセグメンテーション.

り非等方性ガウシアンフィルタ  $g(\Sigma)$  を更新し、パッチ画像の2次モーメント行列を再計算する.

## 2.4.2 Maximally Stable Extremal Regions (MSER)

Maximally Stable Extremal Regions (MSER) [23] は画像の領域分割に基づく手法であり、分割された領域に対して楕円をフィッティングすることでキーポイントのアフィン領域として利用できる。まず、閾値処理により入力画像の2値画像を生成する。図 2.16(a) に示すように、画像を2値化する閾値を徐々に変化させることで、様々な2値画像を生成する。各2値画像においてピクセルが連結する領域(セグメンテーション)を求め、閾値をある程度変化させても連結領域の変化が緩やかなセグメンテーションを検出する。検出したセグメンテーションは閾値の変化に対して同じような領域であるため、安定領域 (stable region) と言える。そして、検出したセグメンテーションに対して楕円フィッティングを行う。セグメンテーションの中心座標がキーポイントの位置、フィッティングした楕円がアフィン領域となる。図 2.16(b) に MSER により領域分割したセグメンテーションを示す。検出したセグメンテーションはランダムな色で着色している。

## 2.5 実数ベクトルによる特徴量記述

この節では、キーポイントマッチングにおける局所特徴量記述について述べる。局所特徴量記述は、実数ベクトルにより特徴量を表現する方法と2値ベクトルにより特徴量を表現する方法に分けることができる。ここでは、局所特徴量を実数ベクトルにより表現する手法について述べる。

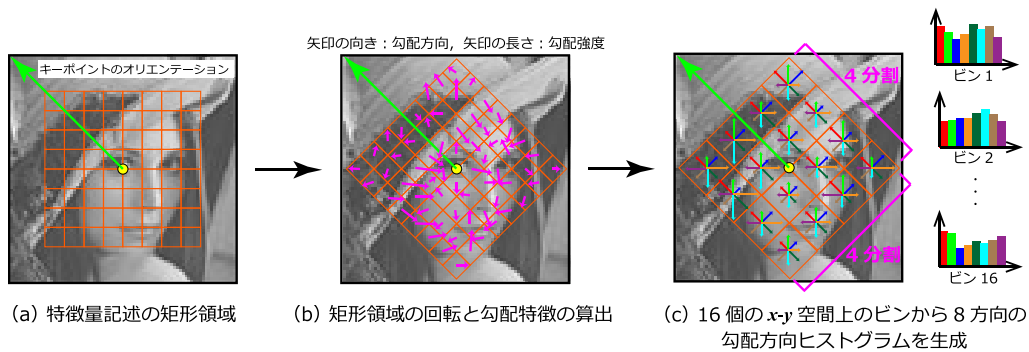


図 2.17: SIFT 特微量の記述.

## 2.5.1 Scale-Invariant Feature Transform (SIFT) Descriptor

Scale-Invariant Feature Transform (SIFT) はキーポイント検出と局所特微量記述の2つの処理で構成されており、キーポイント検出方法は2.3.2項で述べた。ここでは、SIFTの後段処理である局所特微量記述について説明する。

画像から検出したキーポイントの位置  $\mathbf{p}$ 、スケール  $\hat{\sigma}$ 、オリエンテーション  $\hat{\theta}$  を用いて、勾配に基づいた特微量を計算する。まず、図 2.17(a) に示すように特微量を記述する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。特微量の記述には、キーポイントが持つスケールが内接する矩形領域から得られる勾配情報を用いる。矩形領域の一边を均等に4ブロックに分割した計16ブロックで構成される  $x$ - $y$  空間上のビンから、勾配強度と勾配方向を計算する。勾配強度と勾配方向はそれぞれ SIFT のキーポイント検出で定義した式 (2.36) と式 (2.37) により算出する。図 2.17(c) に示すように、各  $x$ - $y$  空間上のビンから8方向 ( $45^\circ$  刻み) の勾配方向ヒストグラムを生成する。この勾配方向ヒストグラムは、SIFT のオリエンテーションの算出時に作成した勾配方向ヒストグラムと同様の方法である。4×4のビンから8方向の勾配方向ヒストグラムを計算するため、最終的に  $4 \times 4 \times 8 = 128$  次元の特微量が記述される。このようにキーポイントが持つスケールとオリエンテーションに基づいて局所特微量を記述することで、拡大・縮小と回転変化に不変な特微量となる。

最後に、画像間の照明変化の影響を低減させるために、次式のように  $i$  次元目の特微量を単位長さで正規化する。

$$d_i = \frac{d_i}{\sum_{j=1}^{N_{dim}} |d_j|} \quad (2.74)$$

ここで、 $N_{dim}$  は特微量の次元数である。照明変化により、画素値に定数の加算または減算が発生しても、特微量は輝度値の差分による勾配で計算されているため影響しない。また、照明変化の影響で画素値に定数の乗算または除算が発生する場合は、ベクトル正規化によりコントラスト変化をキャンセルすることができる。



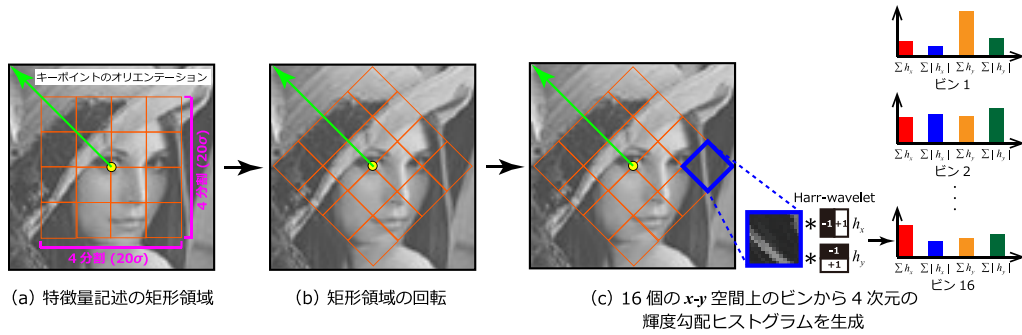


図 2.18: SURF 特徴量の記述.

## 2.5.2 Speeded-Up Robust Features (SURF) Descriptor

Speeded-Up Robust Features (SURF) も SIFT と同様にキーポイント検出と局所特徴量記述の 2 つの処理で構成されており、キーポイント検出方法は 2.3.3 項で述べた。ここでは、SURF の後段処理である局所特徴量記述について説明する。

SURF の特徴量記述はオリエンテーション算出時と同様に Haar-wavelet を用いて輝度勾配を求め、図 2.18(a) に示すように、キーポイントを中心とした  $20\hat{\sigma} \times 20\hat{\sigma}$  の矩形領域を  $4 \times 4$  のグリッドに分割し、 $x$ - $y$  空間上のビンを生成する。矩形領域はオリエンテーション方向に回転され、各  $x$ - $y$  空間上のビンに対して  $2\hat{\sigma} \times 2\hat{\sigma}$  の Haar-wavelet を用いて輝度勾配を計算する。SURF はキーポイント検出時に積分画像を生成しているため、Haar-wavelet を利用することで高速に輝度勾配を求めることができる。  $x$  方向、  $y$  方向の輝度勾配をそれぞれ  $h_x$ 、  $h_y$  とすると、各ビンについて 4 次元の輝度勾配ヒストグラム  $[\sum h_x, \sum h_y, \sum |h_x|, \sum |h_y|]$  が生成される (図 2.18(c))。各ビンにおける輝度勾配ヒストグラムが特徴量となるため、SURF の特徴量は  $4 \times 4 \times 4 = 64$  次元となる。

## 2.5.3 PCA-SIFT

SIFT 特徴量は、高精度な特徴量を記述することができるが各キーポイントに対して 128 次元の高次元な特徴量を求めるため、メモリの消費量が大いという問題がある。PCA-SIFT [26] は、主成分分析 (PCA) を用いることで特徴量の次元を圧縮する。

まず、SIFT により検出したキーポイントのスケール範囲のパッチ画像を生成し、 $41 \times 41$  ピクセルにリサイズする。図 2.19 に示すように、リサイズしたパッチ画像から  $x$  方向および  $y$  方向の勾配を算出し、 $39 \times 39 \times 2 = 3,042$  次元のベクトルを生成する。勾配の計算では、パッチ画像の端領域 1 ピクセルは使用しないため、各勾配画像は  $39 \times 39$  ピクセルとなる。次に、3,042 次元の勾配ベクトルに対して PCA を適用し、PCA 射影行列  $\mathbf{P}_G$  を生成する。PCA 射影行列  $\mathbf{P}_G \in \mathbb{R}^{3042 \times N_p}$  は、大量の学習用画像から検出されたキーポイントの  $x$ 、  $y$  勾配パッチ画像をベクトル化し、その共分散行列の上位  $N_p$  個の固有値に対応する固有ベクトルを並べた行列である。文献 [26] では  $N_p = 36$  を採用している。ここまでの処理は PCA 射影行列  $\mathbf{P}_G$  を求めるための学習である。実際に局所特徴量を計

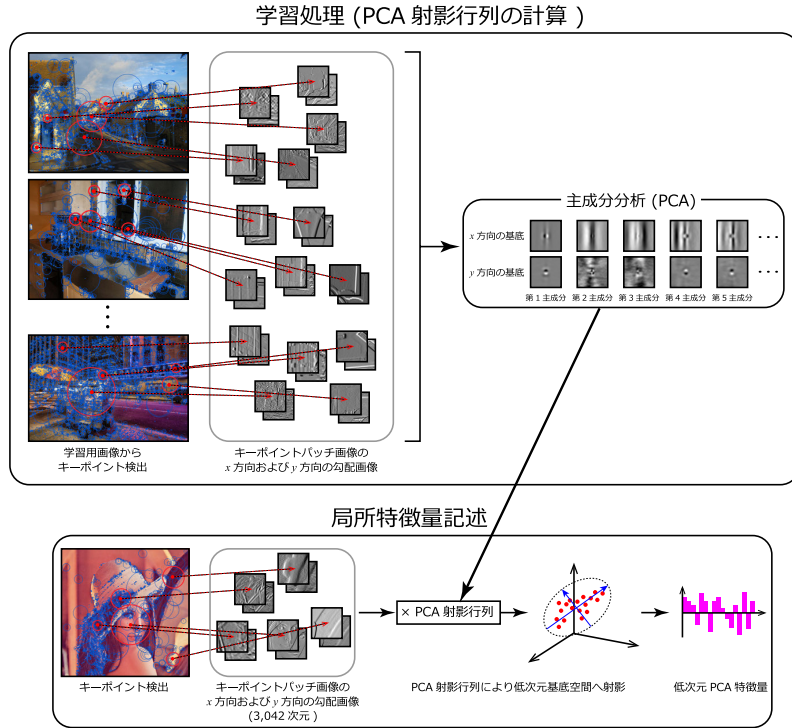


図 2.19: PCA-SIFT の特徴量記述.

算する場合にはキーポイントから求めた  $x, y$  勾配パッチ画像のベクトル  $\mathbf{g}_{x-y} \in \mathbb{R}^{3042}$  に PCA 射影行列  $\mathbf{P}_G$  を掛けることで次元圧縮されたベクトル  $\mathbf{d}_{low} \in \mathbb{R}^{N_p}$  をキーポイントの局所特徴量として使用する.

$$\mathbf{d}_{low} = \mathbf{P}_G^\top \cdot \mathbf{g}_{x-y} \quad (2.75)$$

## 2.5.4 Gradient Location and Orientation Histogram (GLOH)

Gradient Location and Orientation Histogram (GLOH) [50] は SIFT 特徴量を拡張した手法であり、よりロバストな特徴量を記述できるように設計されている. SIFT や SURF といった特徴量は、キーポイント周辺領域における  $x-y$  空間を  $4 \times 4 = 16$  のグリッド状のビンに分割して特徴量を記述する. これに対して、GLOH では特徴量を記述する  $x-y$  空間上のビンを対数極座標 (log-polar) へ変換する. 図 2.20(a) に示すように、半径方向に 3 分割、角度方向に 8 分割した 17 個のビンから勾配方向ヒストグラムを計算する. 各ビンにおいて 16 方向の勾配方向ヒストグラムを計算することで、 $17 \times 16 = 272$  次元の勾配方向ベクトルを算出する (図 2.20(c)). そして、272 次元の勾配方向ベクトルは 2.5.3 項で述べた PCA-SIFT と同様の手順で次元を圧縮し、最終的に 128 次元の勾配方向特徴量を生成する (図 2.20(d)). 次元圧縮に用いる PCA 射影行列は、様々な学習用画像から抽出した 272 次元の勾配方向ベクトルから、あらかじめ計算しておく.

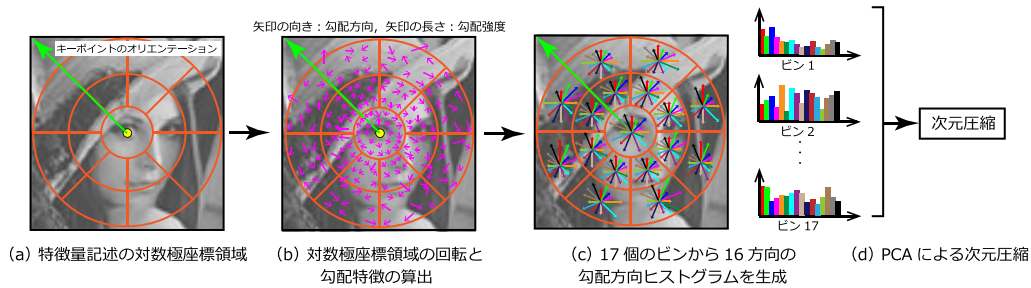


図 2.20: GLOH による特徴量の記述.

## 2.5.5 Root SIFT

SIFT などの実数ベクトルによる局所特徴量では、対応点探索の際に類似度の計算方法として一般的にユークリッド距離が用いられる。Root SIFT [51] では、特徴量間の距離計算においてユークリッド距離の代わりに平方根 (Hellinger) 距離を使用する。テクスチャ分類や画像分類の分野では、ヒストグラム間の類似度としてユークリッド距離を使用すると  $\chi^2$  距離や Hellinger 距離と比較して性能が低下することが知られている。従って、ヒストグラムに基づく局所特徴量によるキーポイントマッチングにおいても Hellinger 距離等を用いることで性能を改善させることができる。実際には、SIFT 特徴量の類似度を Hellinger 距離を使って計算するのではなく、 $N_{dim}$  次元の SIFT 特徴量に対して  $L1$  正規化を行い、特徴量の各要素に対して平方根を計算する。この  $N_{dim}$  次元の特徴ベクトルが Root SIFT であり、 $i$  次元目の Root SIFT 特徴量  $d_i$  は次式のように求められる。

$$d_i = \sqrt{\frac{d_i}{\sum_{j=1}^{N_{dim}} |d_j|}} \quad (2.76)$$

Root SIFT 特徴量を用いたユークリッド距離と通常の SIFT 特徴量を用いた Hellinger 距離は等価となる。通常の SIFT 特徴量を記述した後に式 (2.76) を計算する単純な処理を付け加えるだけでキーポイントマッチングの性能を向上させることができる。そのため、様々なアプリケーションへの導入が容易であり、幅広く使用されてる特徴量である。

## 2.6 2値ベクトルによる特徴量記述

この節では、局所特徴量を2値ベクトルにより表現する手法について述べる。2値特徴量は実数特徴量と比べて精度が劣るものの、高速な特徴量記述と距離計算が可能となる。距離計算においては、式 (2.3) に示すように XOR による論理演算とビットカウントで高速に計算することができる。

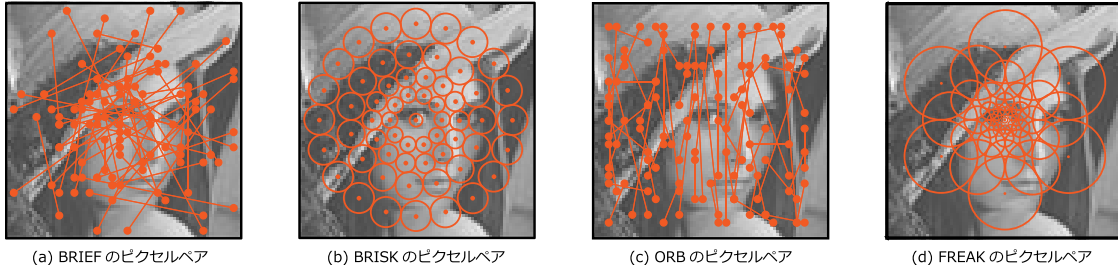


図 2.21: 2 値特徴量のピクセルペアパターン.

## 2.6.1 Binary Robust Independent Elementary Features (BRIEF)

Binary Robust Independent Elementary Features (BRIEF) [29] は、キーポイント周辺のパッチ画像内からランダムに選択されたピクセルペアの輝度差の符号から 2 値特徴量を記述するシンプルな手法である。パッチ画像内からランダムに選択された  $i$  番目のピクセルペアを  $\mathbf{p}_{u_i}, \mathbf{p}_{v_i}$  とし、それぞれのピクセルにおける輝度を  $I(\mathbf{p}_{u_i}), I(\mathbf{p}_{v_i})$  とすると  $i$  ビット目の 2 値特徴量  $d_i$  は次式のように求めることができる。

$$d_i = \begin{cases} 1 & \text{if } I(\mathbf{p}_{u_i}) - I(\mathbf{p}_{v_i}) > 0 \\ 0 \text{ } (-1) & \text{otherwise} \end{cases} \quad (2.77)$$

パッチ画像内から選択されるピクセルペアは、あらかじめ  $N_{dim}$  組用意しておき、これが特徴量の次元数となる。ピクセルペア  $\mathbf{p}_{u_i}, \mathbf{p}_{v_i}$  を選択する方法は複数考えられるが、文献 [29] では図 2.21(a) に示すようにパッチ画像の中心に重み付けされたガウス分布に基づいてランダムにピクセルを選択する。BRIEF では、ノイズに対する影響を低減させるために、パッチ画像をあらかじめガウシアンフィルタで平滑化しておく。このように、パッチ画像内のピクセルペアの輝度差をビット数 (次元数) 分計算するだけで特徴量を記述できるため高速な処理が可能である。

## 2.6.2 Binary Robust Invariant Scalable Keypoints (BRISK)

Binary Robust Invariant Scalable Keypoints (BRISK) [30] は、キーポイント周辺のパッチ画像内に配置された 4 つの同心円上に等間隔にサンプリングされた 60 箇所の輝度値を使用する (図 2.21(b))。2.6.1 項で述べた BRIEF は、特徴量の  $N_{dim} \times 2$  (e.g.,  $N_{dim} = 256$ ) 箇所のピクセルの輝度値へのアクセスが必要となるが、BRISK では 60 箇所の輝度値へのアクセスのみで良いため効率的である。各サンプリング位置は、パッチ画像の中心からの距離に比例する分散を持つガウシアンフィルタにより平滑化されている (図 2.21(b) の各サンプリング位置を中心とする円)。BRISK では、独自のオリエンテーション推定方法を提案している。オリエンテーションは、サンプリング位置の距離が  $\delta_{min}$  以上であるピクセルペア集合  $\mathcal{L}_{pair}$  (長距離ペア) を用いて推定される。オリエンテーションの推定に長距離ペア集合を使用する理由は、パッチ画像内の大局的な輝度勾配方向を捉えるためである。まず、

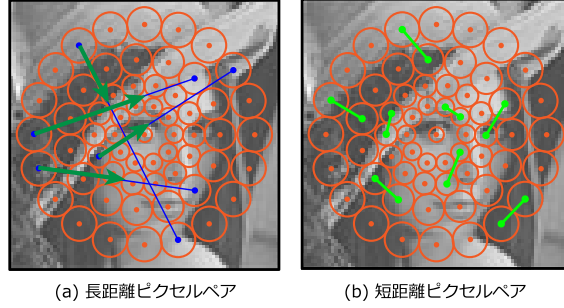


図 2.22: BRISK の長距離ペアと短距離ペア.

長距離ペア  $\mathbf{p}_i, \mathbf{p}_j \in \mathcal{L}_{pair}$  ( $i, j = \{1, 2, \dots, 60\}, i \neq j$ ) において、輝度勾配を次式により求める。

$$\mathbf{g}_p(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_i - \mathbf{p}_j) \frac{L(\mathbf{p}_j; \sigma_j) - L(\mathbf{p}_i; \sigma_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2} \quad (2.78)$$

ここで、 $L(\mathbf{p}_i; \sigma_i)$ ,  $L(\mathbf{p}_j; \sigma_j)$  はスケール  $\sigma_i$ ,  $\sigma_j$  のガウシアンフィルタで平滑化された後の輝度値である。図 2.22(a) に示すように、 $\mathbf{g}_p$  は長距離ペアを直線で結ぶ勾配方向であり、勾配の大きさは長距離ペアの輝度差で与えられる。最後に、長距離ペアで求めた勾配を用いてパッチ画像の大局的なオリエンテーション  $\hat{\theta}$  を次式により推定する。

$$\begin{aligned} [g_{p_x}, g_{p_y}] &= \frac{1}{|\mathcal{L}_{pair}|} \sum_{\mathbf{p}_i, \mathbf{p}_j \in \mathcal{L}_{pair}} \mathbf{g}_p(\mathbf{p}_i, \mathbf{p}_j) \\ \hat{\theta} &= \tan^{-1} \left( \frac{g_{p_y}}{g_{p_x}} \right) \end{aligned} \quad (2.79)$$

このように、BRISK のオリエンテーションは長距離ペア集合の平均勾配ベクトルの角度として定義される。

次に、サンプリング位置の距離が  $\delta_{max}$  以下であるピクセルペア集合  $\mathcal{S}_{pair}$  (短距離ペア) を用いて 2 値特徴量を記述する (図 2.22(b))。特徴量記述に短距離ペア集合を使用する理由は、パッチ画像内の局所的な画像特徴を捉えるためである。 $i$  次元目の 2 値特徴量  $d_i$  は次式により計算できる。

$$d_i = \begin{cases} 1 & \text{if } L(\mathbf{p}_j; \sigma_j) - L(\mathbf{p}_i; \sigma_i) > 0 \\ 0 (-1) & \text{otherwise} \end{cases}, \forall \mathbf{p}_i, \mathbf{p}_j \in \mathcal{S}_{pair} \quad (2.80)$$

BIRSK では、特徴量記述の際に 512 個の短距離ペアを使用するため、最終的に 512 次元の 2 値特徴量が生成される ( $N_{dim} = 512$ )。

### 2.6.3 Oriented FAST and Rotated BRIEF (ORB) Descriptor

Oriented FAST and Rotated BRIEF (ORB) [31] も BRIEF や BRISK といったピクセルペアの輝度差に基づいて 2 値特徴量を記述する手法である。ORB は特徴量記述のみではなく、画像ピラミッドと

FAST コーナー検出器を組み合わせたキーポイント検出法と独自のオリエンテーション推定方法も提案している。ORB におけるキーポイント検出とオリエンテーション推定方法は 2.3.4 項で述べた。ここでは、ORB の特徴量記述方法について述べる。

ORB による特徴量記述方法は BRIEF や BRISK と同様であり、パッチ画像内のピクセルペアの選択方法を工夫している。ORB では、あらゆるピクセルペアに関して統計的に特徴量記述の性能が良くなるピクセルペアを調べて選択する。まず、あらかじめ大量のキーポイントのパッチ画像を学習用画像とし、パッチ画像内で有り得る 205,590 種類のピクセルペア候補を列挙する。全てのピクセルペア候補の輝度差を全学習画像から計算し、パッチ画像間の分散が大きく、かつピクセルペア同士の相関が低くなるようなピクセルペアを採用する。パッチ画像間の分散  $\mathbb{E}$  は次式により計算される。

$$\mathbb{E} = \frac{1}{N_{img}^2} \sum_{m=1}^{N_{img}} \sum_{n=1}^{N_{img}} \text{dist}_H(\mathbf{d}^{(m)}, \mathbf{d}^{(n)}) \quad (2.81)$$

ここで、 $N_{img}$  は学習用パッチ画像の枚数であり、 $\mathbf{d}^{(m)}$  および  $\mathbf{d}^{(n)}$  は各学習用パッチ画像から抽出した特徴ベクトルである。 $\text{dist}_H(\cdot)$  はハミング距離を計算する関数である。パッチ画像間の特徴量の分散が大きいいということは、クラス間分散を最大化しているため、様々な画像においてユニークな特徴量を記述できるピクセルペアと言える。クラス間分散が大きいいピクセルペアのみを採用すると、パッチ画像内で同じような位置のピクセルペアを何度も選択してしまう可能性がある。そこで、ピクセルペア同士の相関が低いペアも選択の条件に加えることで、良い特徴量を記述できるピクセルペアを選択している。 $i$  番目のピクセルペアと  $j$  番目のピクセルペアの相関  $C_{i,j}$  は次式により計算される。

$$C_{i,j} = \left| \frac{2}{N_{img}} \sum_{m=1}^{N_{img}} (d_i^{(m)} \oplus d_j^{(m)}) - 1 \right| \quad (2.82)$$

実際にピクセルペアを選択する処理は、Greedy アルゴリズムを用いて以下の手順で行う。

1. 全てのピクセルペア候補に対して、クラス間分散  $\mathbb{E}$  が大きい順番にソート。
2. 全ピクセルペア候補において、最大のクラス間分散を持つピクセルペアを採用。
3. 採用されたピクセルペアの次に分散の大きいピクセルペア候補に着目し、このピクセルペアの 2 値特徴量と採用済みピクセルペアの 2 値特徴量との相関  $C$  を計算。採用済みの全てのピクセルペアにおいて相関  $C$  が  $T_c$  以下であれば採用。
4. 3 番目の処理を繰り返し、256 組にピクセルペアが採用されたら処理を終了。

相関  $C \in [0, 1]$  の閾値は  $T_c = 0.2$  のように最初は低い値を設定しておく。全てのピクセルペア候補を探索し、256 組のピクセルペアが採用されない場合、閾値  $T_c$  の値を上げてもう一度 3 番目の処理を繰り返してピクセルペアを選択していく。図 2.23 は ORB のピクセルペア選択の例を示した図である。また、図 2.21(c) に ORB で選択されたピクセルペアを示す。選択されたピクセルペアは、縦方向に偏りが発生していることがわかる。これは、学習用パッチ画像を FAST コーナー検出器により

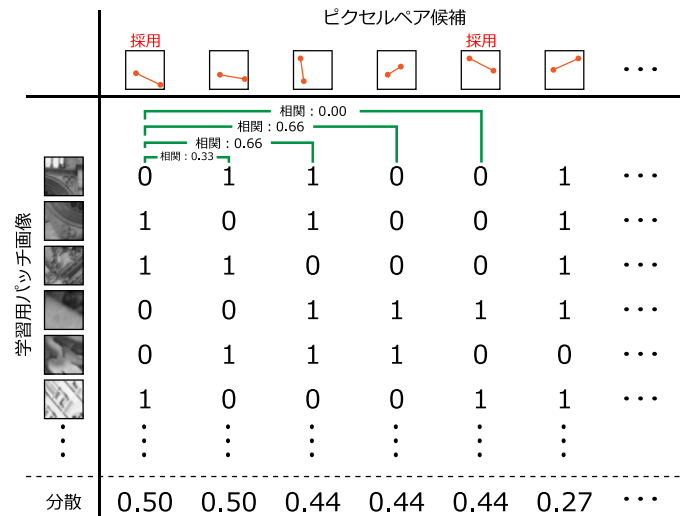


図 2.23: ORB のピクセルペアの選択例.

抽出しており、オリエンテーション方向にパッチ画像を補正しているため、縦方向のピクセルペアの輝度差が特徴量として採用されやすいためである。

## 2.6.4 Fast Retina Keypoint (FREAK)

Fast Retina Keypoint (FREAK) [32] は、生物の網膜構造に基づいてピクセルペアを選択する。眼球内の網膜は、中心に近づくほど錐体細胞の密度が高くなるため、FREAK ではパッチ画像の中心に近づくほど密度が高くなるようなサンプリング位置を用いる。図 2.21(d) に FREAK のサンプリング位置を示す。各サンプリング位置を中心とする円は、ノイズ低減のためのガウシアンフィルタのスケール範囲を表す。ガウシアンフィルタのスケールは、パッチ画像の中心からの距離が長くなるほど指数関数的に大きくなる。

FREAK におけるサンプリング位置は 43 箇所であり、BRISK と比べても少ないが全てのピクセルペアの組み合わせは数千種類となる。そのため、FREAK においても ORB と同様に Greedy アルゴリズムを用いて 2 値特徴量の情報量が大きくなるような 512 組のピクセルペアを選択する。FREAK で選択される 512 種類のピクセルペアは、前半は外側のピクセルペアが選択されていることが多く、後半はパッチ画像の中心付近のピクセルペアが選択されていることが多い。この傾向は生物の視覚システムに類似している。生物の視覚システムでは周辺視野で大まかに観察し、中心視野でより詳細な観察を行う。この視覚システムの構造に基づき、対応点探索で 512 ビットの 2 値特徴量の距離を計算する際に、128 ビットごとに 4 分割し、4 段のカスケード構造を構築する。カスケード構造では、上位の 128 ビットから順番に距離計算し、距離の値があらかじめ設定した閾値を超える場合は計算を打ち切り、早期棄却を行う。実際に FREAK 特徴量により対応点探索を行うと、最初の 128 ビットのみで 90% 以上の対応点候補が棄却されるため、対応点探索においてもより高速な処理が可能である。

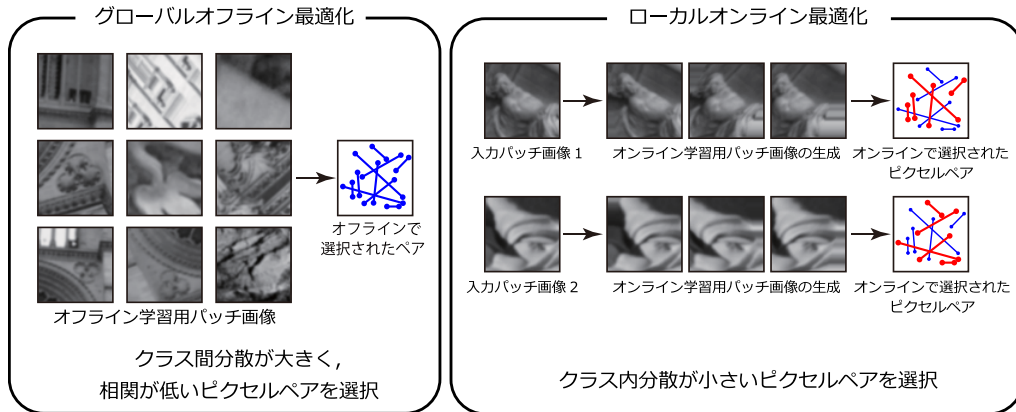


図 2.24: BOLD のピクセルペア選択方法.

## 2.6.5 Binary Online Learned Descriptor (BOLD)

Binary Online Learned Descriptor (BOLD) [52] は, ORB による特徴量記述に基づいた手法である. これまでに述べた BRIEF, BRISK, ORB, FREAK などの特徴量記述は事前に決定したピクセルペアをそのまま使用して 2 値特徴量を記述する手法である. しかし, 事前に決定したピクセルペアは常に固定であるため入力画像のノイズや画像変形によっては大きな性能低下を招くことがある. BOLD では, 2 値特徴量を記述する最適なピクセルペアを入力パッチ画像に応じてオンラインで選択することにより, この問題を解決している.

BOLD のピクセルペアの選択アルゴリズムは, 図 2.24 に示すようにグローバルオフライン最適化とローカルオンライン最適化の 2 段階となっている. グローバルオフライン最適化では, ORB と同様に大量の学習用パッチ画像からクラス間分散を最大化し, ピクセルペア同士の相関が低くなる 512 組のピクセルペアを選択する. グローバルオフライン最適化では, 様々なパッチ画像に対して大域的にピクセルペアを最適化することを目的としている. さらに, BOLD ではパッチ画像の歪み等による性能低下を防ぐためにローカルオンライン最適化によりピクセルペアを選択する. まず, 入力パッチ画像に微小のアフィン変換を施すことで数枚のオンライン学習用パッチ画像を生成する. 次に, グローバルオフライン最適化で選択した 512 組にピクセルペアを用いてアフィン変換したパッチ画像における特徴量の分散, すなわちクラス内分散を計算する. クラス内分散が 0 となるピクセルペア (パッチ画像の歪みの影響を受けにくいピクセルペア) をオンラインで選択することで, 特徴量記述の性能を維持することができる.

ローカルオンライン最適化で選択されたピクセルペアを考慮した 2 値特徴量間 ( $\mathbf{d}, \mathbf{d}'$ ) の距離関数は次式のように定義できる.

$$\text{dist}_M(\mathbf{d}, \mathbf{d}') = \frac{1}{G_{bit}} \mathcal{B} \wedge \mathbf{d} \oplus \mathbf{d}' + \frac{1}{G'_{bit}} \mathcal{B}' \wedge \mathbf{d} \oplus \mathbf{d}' \quad (2.83)$$

ここで,  $\mathcal{B}$  はローカルオンライン最適化で選択されたピクセルペアを 1, 選択されなかったピクセルペアを 0 としたバイナリマスクであり, このバイナリマスクの 1 が立っているビット数が  $G_{bit}$  であ



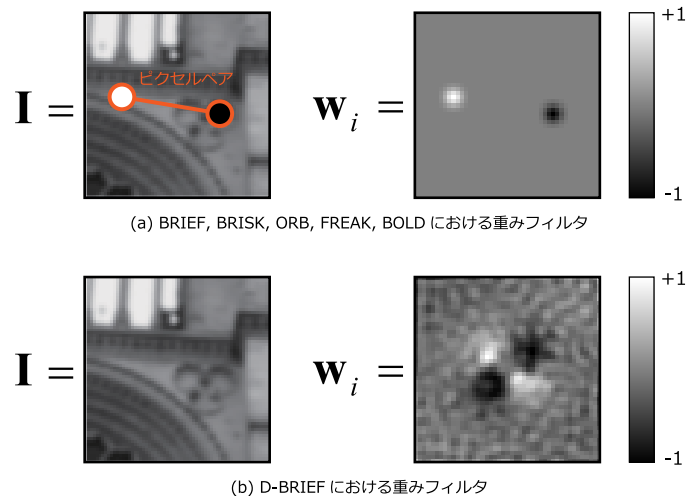


図 2.25: 2 値特徴量記述における重みフィルタ.

る.  $\wedge$  と  $\oplus$  はそれぞれ AND と XOR の演算子を示す. 文献 [52] では, ローカルオンライン最適化でのパッチ画像のアフィン変換は 2 回程度で最も良い性能が得られることが実験的に示されている. また, ローカルオンライン最適化を高速に処理するために, パッチ画像をアフィン変換するのではなく, ピクセルペアの位置をアフィン変換させてクラス内分散を算出する. よって, アフィン変換させたピクセルペアの位置をルックアップテーブルに保持しておくことでオンライン処理でのアフィン変換は必要なくなるため, 低コストなオンライン最適化が実現できる.

## 2.6.6 Discriminative BRIEF (D-BRIEF)

上記で述べた 2 値特徴量記述は, パッチ画像内から選択されるピクセルペアの輝度差の符号により特徴量を表現するシンプルな手法であった. 画像から検出されたキーポイントを中心とするパッチ画像を  $\mathbf{I}$  と表記すると, ピクセルペアの輝度差の演算は画像  $\mathbf{I}$  に対して線形であるため,  $i$  次元目の 2 値特徴量  $d_i$  は次式で表すことができる.

$$d_i = \text{sgn}(\mathbf{w}_i^\top \mathbf{I} + b_i) \quad (2.84)$$

単純にピクセルペアの輝度差の符号で 2 値特徴量を記述する場合, バイアスを  $b_i = 0$  とし,  $\mathbf{w}_i$  は各サンプリングペアに対応する位置に +1 と -1, それ以外に 0 を代入した重みフィルタとすればよい (図 2.25(a)). ピクセルペアの位置にガウシアンフィルタによるノイズ除去を行う場合でも,  $\mathbf{w}_i$  に適切な重み付けをすればよい. 従って BRIEF, BRISK, ORB, FREAK, BOLD の違いは,  $\mathbf{w}_i$  の選び方の違いであると言える. Discriminative BRIEF (D-BRIEF) [53] では, 教師あり学習を用いることで特徴量記述の性能が高くなるような  $\mathbf{w}_i$  と  $b_i$  を求める. 図 2.25(b) は D-BRIEF の教師あり学習で得られた  $\mathbf{w}_i$  を可視化した画像である. 教師データには, キーポイントのパッチ画像の positive ペアと negative ペアを使用する. positive ペアは図 2.26(a) に示すように視点の異なる画像間で物理的に同

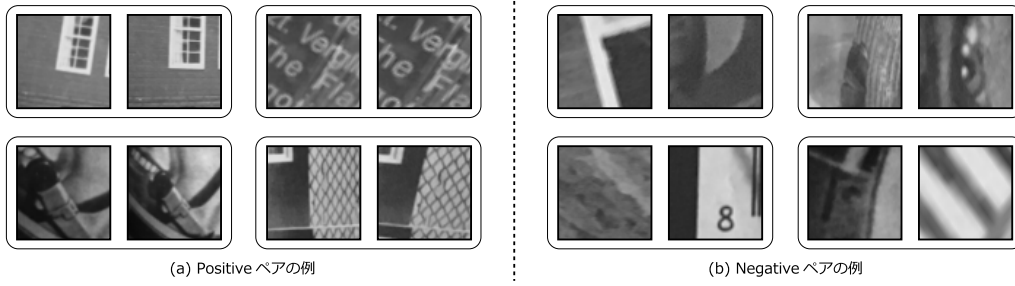


図 2.26: パッチ画像の positive ペアと negative ペアの例.

一の位置から抽出したパッチ画像ペアである. negative ペアは図 2.26(b) に示すように全く異なる位置やシーンから抽出したパッチ画像ペアである. パッチ画像の positive ペアと negative ペアを用いて  $\mathbf{w}_i$  と  $b_i$  を学習することで, パッチ画像ペアの識別に有効な特徴量を記述することができる. しかし, 検出された全てのキーポイントのパッチ画像  $\mathbf{I}$  と重みフィルタ  $\mathbf{w}_i$  の内積を計算するには高い計算コストを必要とする. そこで, D-BRIEF ではさらなる工夫として,  $\mathbf{w}_i$  を矩形フィルタやガウシアンフィルタ等の高速演算可能なフィルタの組み合わせで近似する. 矩形フィルタの場合は積分画像を事前に生成しておくことで高速な処理が可能である. 重みフィルタ  $\mathbf{w}_i$  は辞書行列  $\mathbf{D}_F$  と係数ベクトル  $\mathbf{s}_i$  により, 次式のように近似される.

$$\mathbf{w}_i \approx \mathbf{D}_F \mathbf{s}_i \quad (2.85)$$

辞書行列  $\mathbf{D}_F$  はあらゆる形状の矩形フィルタまたはガウシアンフィルタで構成されており,  $\mathbf{s}_i$  がスパースであれば, 重みフィルタ  $\mathbf{w}_i$  は少ない矩形フィルタの組み合わせで再構成できる.

D-BRIEF の学習処理は, 次式の最小化問題を解くことである.

$$\begin{aligned} \min_{\mathbf{s}_i, b_i} \sum_{\mathbf{I}, \mathbf{I}' \in \mathcal{N}_{pair}} \mathbf{d}^\top \mathbf{d}' - \sum_{\mathbf{I}, \mathbf{I}' \in \mathcal{P}_{pair}} \mathbf{d}^\top \mathbf{d}' + \lambda_r |\mathbf{s}_i|_1 \\ d_i = \text{sgn}((\mathbf{D}_F \mathbf{s}_i)^\top \mathbf{I} + b_i), \quad d_i \in \mathbf{d} \end{aligned} \quad (2.86)$$

ここで,  $\mathbf{I}, \mathbf{I}' \in \mathcal{P}_{pair}$  は positive ペアのパッチ画像,  $\mathbf{I}, \mathbf{I}' \in \mathcal{N}_{pair}$  は negative ペアのパッチ画像である.  $\lambda_r$  は  $\mathbf{s}_i$  をどの程度スパースにするかを定める係数であり,  $\lambda_r$  が大きいほど  $\mathbf{s}_i$  がスパースとなる. 式 (2.86) では, negative ペアの 2 値特徴量同士の内積を小さくし, positive ペアの 2 値特徴量同士の内積を大きくするように  $\mathbf{s}_i$  と  $b_i$  を学習していることがわかる. D-BRIEF における 2 値特徴量は  $\mathbf{d}, \mathbf{d}' \in \{-1, +1\}^{N_{dim}}$  であるため, 内積  $\mathbf{d}^\top \mathbf{d}'$  はハミング距離  $\text{dist}_H(\mathbf{d}, \mathbf{d}')$  により次式で定義できる.

$$\mathbf{d}^\top \mathbf{d}' = N_{dim} - 2 \cdot \text{dist}_H(\mathbf{d}, \mathbf{d}') \quad (2.87)$$

従って, 特徴量間の内積  $\mathbf{d}^\top \mathbf{d}'$  を大きくすることはハミング距離  $\text{dist}_H(\mathbf{d}, \mathbf{d}')$  を小さくすることに相当する.

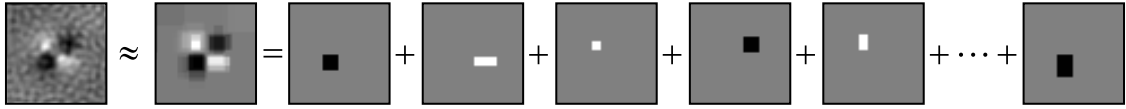


図 2.27: 矩形フィルタによる重みフィルタの近似.

式 (2.86) の最小化問題を解くことで、最適な  $\mathbf{w}_i$  と  $b_i$  が得られるが実際には式 (2.86) を直接解くことが困難であるため、sgn 関数と  $L1$  正則化の項を取り除いて  $\mathbf{w}_i$  についての最小化問題を解く.

$$\{\mathbf{w}_i^0\} = \arg \min_{\{\mathbf{w}_i\}} \sum_{i=1}^{N_{dim}} \left( \frac{\sum_{\mathbf{I}, \mathbf{I}' \in \mathcal{P}_{pair}} (\mathbf{w}_i^\top (\mathbf{I} - \mathbf{I}'))^2}{\sum_{\mathbf{I}, \mathbf{I}' \in \mathcal{N}_{pair}} (\mathbf{w}_i^\top (\mathbf{I} - \mathbf{I}'))^2} \right) \quad (2.88)$$

式 (2.88) の最適化では、バイアス  $b_i$  が取り除かれているが最適化された  $\mathbf{w}_i$  を固定し、 $b_i$  に関する 1 次元探索問題で式 (2.86) を再度解くことで最適な  $b_i$  を決定する.  $\mathbf{w}_i^0$  を少数の矩形フィルタまたはガウシアンフィルタで近似するには、 $\mathbf{w}_i^0$  と  $\mathbf{D}_F \mathbf{s}_i$  との誤差が小さく、 $\mathbf{s}_i$  がなるべくスパースになるように求める. これは、次式で定義する最小化問題で  $\mathbf{s}_i$  を求める.

$$\{\mathbf{s}_i^0\} = \arg \min_{\{\mathbf{s}_i\}} \|\mathbf{w}_i^0 - \mathbf{D}_F \mathbf{s}_i\|_2^2 + \lambda |\mathbf{s}_i|_1 \quad (2.89)$$

図 2.27 に図 2.25(b) の重みフィルタを矩形フィルタの組み合わせで近似した例を示す.

## 2.6.7 Bin Boost

Bin Boost [33] も教師あり学習に基づいて局所特徴量を記述する手法であり、 $N_k$  個の弱識別器の線形和の符号で 2 値特徴量を生成する. Bin Boost による  $i$  次元目の 2 値特徴量  $d_i$  は次式により求めることができる.

$$d_i = \text{sgn}(\mathbf{w}_i^\top \mathbf{h}_i(\mathbf{I})) \quad (2.90)$$

$\mathbf{I}$  はパッチ画像、 $\mathbf{h}_i(\mathbf{I}) = [h_{i,1}(\mathbf{I}), h_{i,2}(\mathbf{I}), \dots, h_{i,N_k}(\mathbf{I})] \in \mathbb{R}^{N_k}$  は  $N_k$  個の弱識別器の出力、 $\mathbf{w}_i \in \mathbb{R}^{N_k}$  は各弱識別器に対する重みである.  $h_{i,j}(\mathbf{I})$  はパッチ画像に関する任意の特徴抽出関数を適用することができる. 重みを掛ける前にパッチ画像を弱識別器に入力する点が D-BRIEF と大きく異なる点である. 文献 [33] では、パッチ画像の矩形領域における勾配方向を弱識別器として使用している. 弱識別器  $h_{i,j}(\cdot)$  は矩形領域、勾配のオリエンテーション、閾値の 3 つのパラメータを持っており、矩形領域内の勾配方向ヒストグラムを計算し、オリエンテーションに該当するヒストグラムの値が閾値以上の場合  $-1$ 、閾値未満の場合  $+1$  を出力する. 特徴量の各次元に対して  $N_k$  個の弱識別器を用いて特徴量を記述するため、特徴量の次元数を  $N_{dim}$  とすると  $N_{dim} \times N_k$  個の弱識別器が必要となる (図 2.28). Bin Boost では、特徴記述に用いる弱識別器  $h_{i,j}(\cdot)$  と弱識別器の重み  $\mathbf{w}_i$  を教師あり

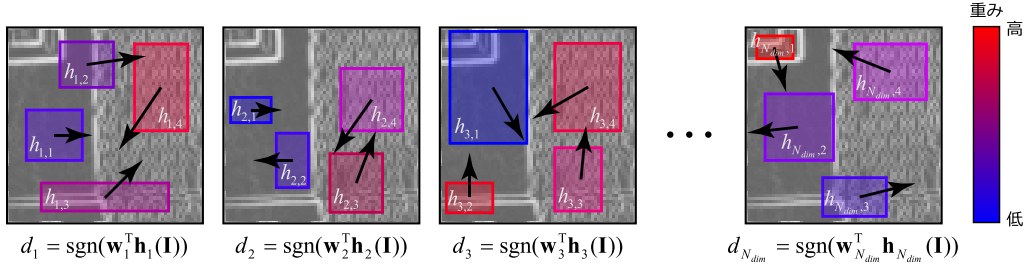


図 2.28: Bin Boost による 2 値特徴量の記述.

学習により決定する.

まず,  $N_{img}$  個のラベル付き学習パッチ画像を  $\{\mathbf{I}_n, \mathbf{I}'_n, l_n\}_{n=1}^{N_{img}}$  と定義する.  $l_n$  は画像ペア  $\mathbf{I}_n, \mathbf{I}'_n$  に対するラベルであり,  $\mathbf{I}_n, \mathbf{I}'_n$  が positive ペアである場合は  $l_n = +1$ , negative ペアである場合は  $l_n = -1$  とする. これらを用いて,  $\mathbf{w}_i$  と  $\mathbf{h}_i$  に関する最小化問題により学習を行う.

$$\min_{\{\mathbf{w}_i, \mathbf{h}_i\}_{i=1}^{N_{dim}}} \sum_{n=1}^{N_{img}} \exp \left( -\vartheta l_n \sum_{i=1}^{N_{dim}} \tilde{c}_i(\mathbf{I}_n, \mathbf{I}'_n; \mathbf{w}_i, \mathbf{h}_i) \right) \quad (2.91)$$

$$\tilde{c}_i(\mathbf{I}_n, \mathbf{I}'_n; \mathbf{w}_i, \mathbf{h}_i) = \text{sgn}(\mathbf{w}_i^\top \mathbf{h}_i(\mathbf{I}_n)) \cdot \text{sgn}(\mathbf{w}_i^\top \mathbf{h}_i(\mathbf{I}'_n))$$

この最小化は, 式 (2.87) の関係から positive ペアに関してはハミング距離を小さくし, negative ペアに関してはハミング距離を大きくするように作用する. AdaBoost [54] の考え方にに基づき,  $i = 1$  から順番に弱識別器  $\mathbf{h}_i$  と重み  $\mathbf{w}_i$  を最適化する. これは, 式 (2.91) の代わりに次式の最大化により達成できる.

$$\max_{\{\mathbf{w}_i, \mathbf{h}_i\}_{i=1}^{N_{dim}}} \sum_{n=1}^{N_{img}} l_n W_i(n) \tilde{c}_i(\mathbf{I}_n, \mathbf{I}'_n; \mathbf{w}_i, \mathbf{h}_i) \quad (2.92)$$

ここで,  $W_i(n)$  は学習サンプルごとに与えられる重み係数であり, 次式により定義される.

$$W_i(n) = \exp \left( -\vartheta l_n \sum_{i'=1}^{i-1} \tilde{c}_{i'}(\mathbf{I}_n, \mathbf{I}'_n; \mathbf{w}_{i'}, \mathbf{h}_{i'}) \right) \quad (2.93)$$

$W_i(n)$  は  $1 \sim i-1$  番目までの次元で正しく識別できていない学習サンプルに対して重みが大きくなり, 既に正しく識別できている学習サンプルに対しては重みが小さくなる. これは, AdaBoost の学習方法と非常に類似している. しかし, 関数  $\tilde{c}_i$  の中にある  $\text{sgn}(\cdot)$  関数は微分不可であるため, 式 (2.92) を解くことが困難である. そこで,  $\text{sgn}(\cdot)$  を取り除いて最適化式を近似する.

$$\max_{\{\mathbf{w}_i, \mathbf{h}_i\}_{i=1}^{N_{dim}}} \mathbf{w}_i^\top \left( \sum_{n=1}^{N_{img}} l_n W_i(n) \mathbf{h}_i(\mathbf{I}_n) \mathbf{h}_i(\mathbf{I}'_n)^\top \right) \mathbf{w}_i \quad (2.94)$$

まず,  $\mathbf{h}_i$  に対して最適化を行う. 弱識別器  $\mathbf{h}_i$  には無数の候補が存在するが, それらの弱識別器候

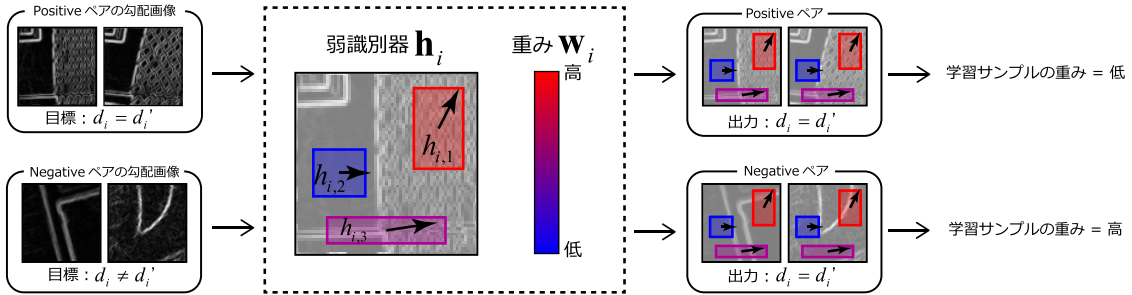


図 2.29: Bin Boost による学習の流れ.

補の中から大きな重み  $W_i(n)$  が与えられている学習サンプルを正確に識別できる弱識別器の集合を求める。つまり、各特徴量次元  $i$  毎に  $N_k$  個の弱識別器を求める特徴選択の問題を解く。そして、 $\mathbf{w}_i$  に関して式 (2.94) を解く。識別に有効な弱識別器  $\mathbf{h}_i$  が決まれば式 (2.94) は次式のように表せる。

$$\max_{\mathbf{w}_i} \mathbf{w}_i^\top M_h \mathbf{w}_i \quad (2.95)$$

$$M_h = \sum_{n=1}^{N_{img}} l_n W_i(n) \mathbf{h}_i(\mathbf{I}_n) \mathbf{h}_i(\mathbf{I}'_n)^\top$$

以上の手順により、 $i$  次元目の特徴量を記述するための  $N_k$  個の弱識別器  $\mathbf{h}_i(\mathbf{I})$  と重み  $\mathbf{w}_i$  を獲得することができる。これを、特徴量の次元数  $N_{dim}$  まで繰り返すことで 2 値特徴量を記述する。図 2.29 に Bin Boost による学習処理の流れを示す。文献 [33] では、 $N_{dim} = 64$  の 2 値特徴量で SIFT の性能を上回ることが報告されている。

## 2.7 視点合成に基づいた多視点特徴量記述

この節では、視点合成に基づいた多視点特徴量について述べる。2.5 節や 2.6 節で述べた特徴量記述は、キーポイント検出器で推定したスケール  $\hat{\sigma}$  やオリエンテーション  $\hat{\theta}$  を用いることで画像間のスケール変化、回転変化に対してロバストな特徴量を記述することができる。また、特徴量を正規化することで照明変化に対してもロバストな特徴量となる。しかし、画像間の視点変化には射影変換等によるスケールや回転以外の歪みが発生する。画像間の射影変換に対してもロバストな特徴量を記述するために視点合成に基づいた多視点特徴量が提案されている。以下に視点合成に基づいた多視点特徴量記述の代表的な手法を述べる。

### 2.7.1 Affine SIFT (ASIFT)

Affine SIFT (ASIFT) [38] は、強い視点変化が発生する画像に対してロバストな特徴量を記述するために、様々なアフィン変換を入力画像に適用する。アフィン変換された全ての画像から SIFT 特徴量 [1] を記述することにより、視点が大きく異なる画像間においても高精度なキーポイントマッチン

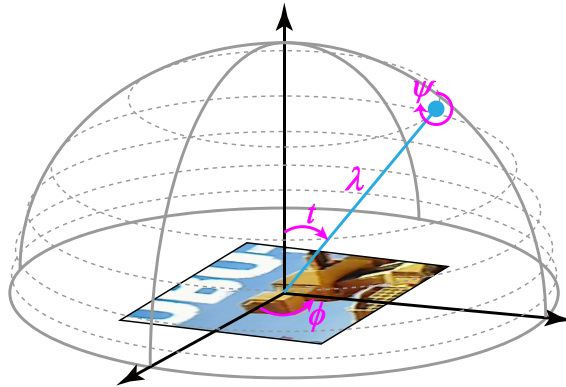


図 2.30: 画像の視点合成.

グを行うことが可能となる. 入力画像をアフィン変換することで, 様々な局所的な視点をシミュレートしていることとなり, 多視点の特徴量が記述される. このように, 様々な視点をシミュレートすることを視点合成と呼び, 視点合成によって記述される特徴量をここでは多視点特徴量と呼ぶ.

画像間の射影変換を表現するホモグラフィ行列  $\mathbf{H}$  は非線形な変換であるが, これは画像の局所領域の変換を仮定するとテイラー展開により線形なアフィン行列  $\mathbf{H}_A$  で近似することができる. スケール行列  $\mathbf{L}_\lambda$ , スキュー行列  $\mathbf{T}_t$ , 回転行列  $\mathbf{R}_1(\psi)$ ,  $\mathbf{R}_2(\phi)$  により, アフィン行列  $\mathbf{H}_A$  は次式のように定義できる.

$$\begin{aligned} \mathbf{H}_A &= \mathbf{L}_\lambda \mathbf{R}_1(\psi) \mathbf{T}_t \mathbf{R}_2(\phi) \\ &= \lambda \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \end{aligned} \quad (2.96)$$

ここで,  $\lambda > 0$  はスケールパラメータ,  $t \geq 1$  は経度に対応する傾きパラメータである.  $\phi \in [0, \pi)$  は緯度に対応する回転パラメータ,  $\psi \in [0, 2\pi)$  は視点軸に対応する面内回転パラメータである. 図 2.30 に視点合成におけるアフィンパラメータの関係を示す. 視点合成のアフィンパラメータは  $\{\lambda, \psi, t, \phi\}$  の 4 パラメータ存在するが, ASIFT では視点合成により画像をアフィン変換した後に SIFT によりキーポイントを検出するため,  $\{\lambda, \psi\}$  はキーポイントのスケール  $\hat{\sigma}$  とオリエンテーション  $\hat{\theta}$  に置き換えることができる. よって, 視点合成に使用する実際のアフィンパラメータは  $\{t, \phi\}$  の 2 パラメータとなる. 視点変化にロバストな多視点特徴量を記述するには, アフィンパラメータ  $\{t, \phi\}$  を適切な間隔でサンプリングした上でアフィン変換画像から特徴量を記述する必要がある. 文献 [38] では, 傾きパラメータを  $t = \{1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}\}$  とし, 回転パラメータのサンプリング間隔  $\Delta\phi$  を傾きパラメータ  $t$  に応じて  $\Delta\phi = \frac{72^\circ}{t}$  と設定することを推奨している.

ASIFT は, 各アフィン変換画像から特徴量を記述するためアフィン変換の回数に応じて特徴量  $\mathbf{d}$  の数が増えるが, これらの特徴量は全て独立した特徴量として対応点探索に用いる. すなわち, 1 つのキーポイントに対して複数の特徴量を持つことになる. 図 2.31 に ASIFT による特徴量記述と対応点探索の流れを示す.

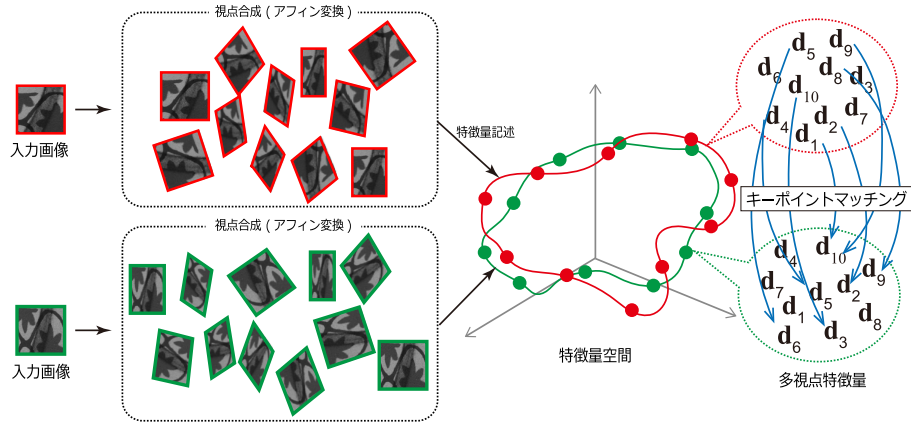


図 2.31: ASIFT によるキーポイントマッチング.

## 2.7.2 Affine Subspace Representation (ASR)

Affine Subspace Representation (ASR) [39] も, ASIFT と同様に画像の視点合成に基づいた特徴量である. ASIFT では, 全てのアフィン変換画像から記述した特徴量を独立したベクトルとして扱っていた. ASR では, アフィン変換画像から記述した特徴量集合をアフィン部分空間に投影することでよりロバストな特徴量を表現している. ASR はパッチ画像を直接アフィン変換することで部分空間特徴量を記述する ASR-naive とアフィン変換画像を基底パッチ画像の線形演算で近似して高速化した ASR-fast を提案している. 以下に ASR-naive と ASR-fast の特徴量記述方法について述べる.

### ■ ASR-naive

まず, ASIFT と同様にキーポイントにおけるパッチ画像をアフィン変換させる. アフィンパラメータ  $\{t, \phi\}$  よりアフィン変換した画像  $\mathbf{I}(t, \phi)$  から特徴量  $\mathbf{d}(t, \phi)$  を記述する. 特徴量は, アフィン変換パッチ画像  $\mathbf{I}(t, \phi)$  に対して PCA 射影行列  $\mathbf{P}_I$  を掛けることで記述する. この特徴量記述方法は, PCA-SIFT と非常に似ており, PCA-SIFT ではパッチ画像の  $x$  方向と  $y$  方向の勾配画像に PCA 射影行列を掛けるのに対して ASR は画像の輝度そのものに PCA 射影行列を掛ける. よって, 全てのアフィン変換から記述した特徴量集合は次式に示すように行列  $\mathbf{D}$  で表すことができる.

$$\mathbf{D} = \mathbf{P}_I^\top \mathbf{I}_A \quad (2.97)$$

ここで,  $\mathbf{I}_A = [\mathbf{I}(t_1, \phi_1) \ \mathbf{I}(t_2, \phi_2) \ \cdots \ \mathbf{I}(t_{N_a}, \phi_{N_a})]$  は, アフィン変換パッチ画像  $\mathbf{I}(t, \phi)$  のベクトルを列に並べた行列である.  $N_a$  はパッチ画像のアフィン変換回数である.  $\mathbf{P}_I$  は, 大量の学習パッチ画像の輝度値から求めた PCA 射影行列である. 文献 [39] では, PCA 射影行列の基底数は  $N_p = 24$  と設定している. これにより, 各アフィン変換画像の特徴量  $\mathbf{D} = [\mathbf{d}(t_1, \phi_1) \ \mathbf{d}(t_2, \phi_2) \ \cdots \ \mathbf{d}(t_{N_a}, \phi_{N_a})]$  が求められる. ここまでは, 特徴量の記述方法が異なるものの ASIFT のアルゴリズムとほとんど同

じであるが、ASRでは特徴量集合  $\mathbf{D}$  をアフィン部分空間へ投影する。特徴量集合  $\mathbf{D}$  をさらに PCA を用いて線形部分空間で表現すると次式が得られる。

$$\mathbf{D} \approx \begin{bmatrix} \hat{\mathbf{d}}_1 & \hat{\mathbf{d}}_2 & \cdots & \hat{\mathbf{d}}_{N_s} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,N_a} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,N_a} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N_s,1} & b_{N_s,2} & \cdots & b_{N_s,N_a} \end{bmatrix} \quad (2.98)$$

ここで、 $\hat{\mathbf{d}}$  はアフィン部分空間における基底ベクトルであり、 $b$  は部分空間座標となる。  $N_s$  はアフィン部分空間における基底数であり、  $N_s = 8$  で十分に元の特徴量を近似できることが報告されている [39]。ここで、  $\hat{\mathbf{D}} = [\hat{\mathbf{d}}_1 \ \hat{\mathbf{d}}_2 \ \cdots \ \hat{\mathbf{d}}_{N_s}]$  と表記すると部分空間上での距離  $\text{dist}_S(\cdot)$  は次式のように定義できる。

$$\text{dist}_S(\mathbb{D}, \mathbb{D}') = \|\sin(\theta)\|_2 = \frac{1}{\sqrt{2}} \|\hat{\mathbf{D}}\hat{\mathbf{D}}^\top - \hat{\mathbf{D}}'\hat{\mathbf{D}}'^\top\|_F \quad (2.99)$$

ここで、  $\mathbb{D}, \mathbb{D}'$  はそれぞれ画像間のパッチ画像  $\mathbf{I}, \mathbf{I}'$  における部分空間、  $\theta$  は部分空間同士の主角度である。この部分空間は基底ベクトルで構成される行列  $\hat{\mathbf{D}}$  を用いることで、ある空間上の点へとマッピングすることができる。マッピングのための射影行列は  $\mathbf{E} = \hat{\mathbf{D}}\hat{\mathbf{D}}^\top$  で表すことができ、行列  $\mathbf{E}$  の対角成分にスケール係数  $\frac{1}{\sqrt{2}}$  を掛け、その上三角行列を取ることで部分空間を1つの特徴ベクトル  $\mathbf{d}_{sub}$  として表現することができる。

$$\mathbf{d}_{sub} = \left[ \frac{e_{1,1}}{\sqrt{2}} \quad e_{1,2} \quad e_{1,3} \quad \cdots \quad e_{1,N_p} \quad \frac{e_{2,2}}{\sqrt{2}} \quad e_{2,3} \quad \cdots \quad \frac{e_{N_p,N_p}}{\sqrt{2}} \right] \quad (2.100)$$

$$\mathbf{E} = \hat{\mathbf{D}}\hat{\mathbf{D}}^\top = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,N_p} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,N_p} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N_p,1} & e_{N_p,2} & \cdots & e_{N_p,N_p} \end{bmatrix}$$

式 (2.100) により、視点合成による多視点特徴量  $\mathbf{D} = [\mathbf{d}(t_1, \phi_1) \ \mathbf{d}(t_2, \phi_2) \ \cdots \ \mathbf{d}(t_{N_a}, \phi_{N_a})]$  を部分空間特徴量  $\mathbf{d}_{sub}$  として表現することができる。  $\mathbf{d}_{sub}$  は部分空間をマッピングした後のベクトルであるため、特徴量間の距離は単純にユークリッド距離  $\text{dist}_E(\cdot)$  で計算できる。

$$\text{dist}_S(\mathbb{D}, \mathbb{D}') = \text{dist}_E(\mathbf{d}_{sub}, \mathbf{d}'_{sub}) = \|\mathbf{d}_{sub} - \mathbf{d}'_{sub}\|_2 \quad (2.101)$$

ASRでは、パッチ画像の輝度をそのまま次元圧縮して特徴量として用いるため照明変化の影響を受けるが、部分空間表現を用いることでパッチ画像の照明変化を吸収することができる。2画像間のキーポイントパッチ画像  $\mathbf{I}, \mathbf{I}'$  の多視点特徴量をそれぞれ  $\mathbf{D} = [\mathbf{d}(t_1, \phi_1) \ \mathbf{d}(t_2, \phi_2) \ \cdots \ \mathbf{d}(t_{N_a}, \phi_{N_a})]$ ,  $\mathbf{D}' = [\mathbf{d}'(t_1, \phi_1) \ \mathbf{d}'(t_2, \phi_2) \ \cdots \ \mathbf{d}'(t_{N_a}, \phi_{N_a})]$  と表記し、パッチ画像間の照明変化が線形であると仮



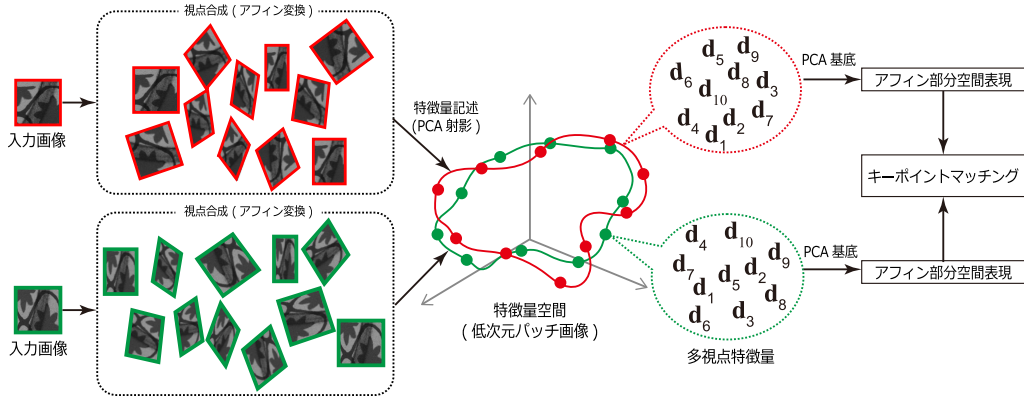


図 2.32: ASR-naive によるキーポイントマッチング.

定すると  $\mathbf{d}(t, \phi) = a \times \mathbf{d}'(t, \phi) + b$  となる.  $a, b$  はそれぞれ線形照明変化の係数である. 多視点特徴量集合の共分散行列をそれぞれ  $\text{cov}(\mathbf{D}), \text{cov}(\mathbf{D}')$  と表記すると, 線形照明変化のパッチ画像間の共分散行列の関係は  $\text{cov}(\mathbf{D}) = a^2 \times \text{cov}(\mathbf{D}')$  となる. これらの共分散行列は同じ固有ベクトルを持ち, ASR では共分散行列  $\text{cov}(\mathbf{D}), \text{cov}(\mathbf{D}')$  の固有ベクトルで特徴量を構成するため, 照明変化の影響を吸収できることがわかる.

図 2.32 に ASR-naive による特徴量記述と対応点探索の流れを示す. ASIFT では, アフィン変換画像から記述した特徴量をそれぞれ対応点探索に用いていたが, ASR ではアフィン変換画像から求めた特徴量を部分空間を用いて 1 つの特徴量として表現する.

### ■ ASR-fast

ASR-naive はアフィン変換画像から得られる多視点特徴量を部分空間表現することで, 視点変化にロバストな特徴量を記述できる. しかし, ASR-naive は ASIFT と同様に入力パッチ画像を直接アフィン変換するため処理時間が増加する問題が解決されていない. そこで, ASR-fast ではパッチ画像自体を PCA の基底画像の線形演算で近似する手法 [55] を導入することで処理を高速化している.

入力パッチ画像  $\mathbf{I}$  は, 様々な画像に PCA を適用することにより求められる PCA 基底画像  $\mathbf{V}$  と係数  $\tilde{a}$  の線形結合で近似できる.

$$\mathbf{I} \approx \bar{\mathbf{V}} + \sum_{i=1}^{N_v} \tilde{a}_i \mathbf{V}_i \quad (2.102)$$

$\bar{\mathbf{V}}$  は平均パッチ画像であり, PCA 基底画像  $\mathbf{V}$  と平均パッチ画像  $\bar{\mathbf{V}}$  に対してアフィン変換を適用することで, アフィン変換画像を再構成することができる.  $N_v$  は, PCA 基底画像の枚数であり文献 [39] では  $N_v = 160$  としている. 式 (2.102) で入力画像に依存するのは投影座標である  $\tilde{a}_i$  のみで, PCA 基底画像  $\mathbf{V}$  と平均パッチ画像  $\bar{\mathbf{V}}$  は一度計算しておけば常に固定であるため,  $\mathbf{V}$  と  $\bar{\mathbf{V}}$  に対して事前にアフィン変換を適用しておくことが可能である. このパッチ画像の近似により, 入力画像のアフィン変換をオンラインで処理する必要がなくなるため高速な特徴量記述が可能となる. 係数  $\tilde{a}_i$  は入力

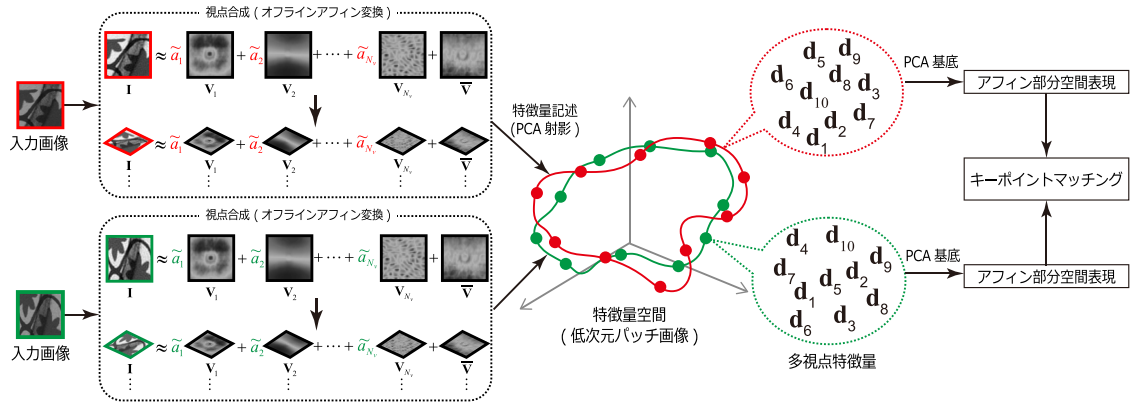


図 2.33: ASR-fast によるキーポイントマッチング.

画像に応じて次式のように計算される.

$$\begin{aligned} \tilde{\mathbf{a}} &= \mathbf{P}_V^\top \mathbf{I} \\ \tilde{\mathbf{a}} &= [\tilde{a}_1 \quad \tilde{a}_2 \quad \tilde{a}_3 \quad \cdots \quad \tilde{a}_{N_v}]^\top \end{aligned} \quad (2.103)$$

$\mathbf{P}_V$  は基底画像  $\mathbf{V}$  に対する PCA 射影行列であり, これも事前に用意した学習画像で計算しておく. 図 2.33 に ASR-fast による特微量記述と対応点探索の流れを示す. ASR-naive では, パッチ画像を直接アフィン変換していたのに対して ASR-fast では PCA 基底画像  $\mathbf{V}$  と平均パッチ画像  $\bar{\mathbf{V}}$  を事前にアフィン変換しておくことで, アフィン変換された画像を高速に再構成する.

## 2.8 まとめ

本章では, キーポイントマッチングの処理の流れについて述べた後, これまでに提案されたキーポイント検出法と局所特微量記述子について述べた.

キーポイント検出の初期の研究 [11, 13, 14, 15, 41, 42] では, 画像間の平行移動や回転変化に対してのみ不変なキーポイントを検出していたが, 画像のスケールスペースを導入することでスケール変化にも頑健なキーポイントを検出できるようになった [1, 16, 17, 18, 19]. また, キーポイントにおけるスケール不変な領域を楕円形状へと拡張することで, アフィン不変な領域を推定することが可能となった [21, 22, 23, 24, 56]. キーポイント検出の高速化という面では, 機械学習により構築した決定木で効率的に探索することで高速なキーポイント検出が達成されている [45, 57, 58]. 様々なキーポイント検出法における性能の比較や評価指標については文献 [25] に詳しく述べられている.

一方, 局所特微量記述ではキーポイント周辺領域の勾配方向ヒストグラムに基づいて特微量を記述する手法により高性能なキーポイントマッチングを実現できる [1, 51]. また, 高次元の特微量の次元圧縮やヒストグラムの簡略化により, 低次元かつロバストな特微量を記述する手法も多く提案された [18, 26, 50]. さらに, 特微量を実数ベクトルで保持するのではなく 2 値ベクトルで保持することで, 省メモリかつ高速なキーポイントマッチングが実現できるため精力的に研究されてきた

[29, 30, 31, 37, 32, 52, 53, 33]. 特徴量記述の視点変化に対するロバスト化という面では, 入力パッチ画像の視点合成により多視点特徴量を記述することで画像間の強い視点変化に対してもキーポイントマッチングが実現できるようになった [38, 39, 59]. 特徴量記述における性能の比較や評価指標については文献 [50, 60, 61] に詳しく述べられている.

以降の章では, キーポイントマッチングの各処理において解決されていなかった問題について取り組んだ研究について述べる. 3章では, キーポイントマッチングに不必要なキーポイントの過剰な検出を抑制しつつ高速にキーポイントを検出する手法を提案する. 4章では, キーポイントに対して複数のアフィン領域を推定することで, 高精度なアフィン領域推定を実現する. 5章と6章では, 多視点特徴量を記述する際に特徴量記述子に対して視点合成を行う効率的な手法を提案し, 因子分解法を用いることで従来よりも効率的に多視点特徴量を記述する. 7章では, 物流ロボットにおける物体認識への応用と特徴量マッチングによる未学習物体の識別を実現させる.

## 第3章

# Cascaded FASTによるキーポイント検出

本章では、キーポイントマッチングに不必要なキーポイントの過剰な検出を抑制することで、高速なキーポイント検出およびキーポイントマッチングが可能な Cascaded FAST を提案する。キーポイント検出は FAST コーナー検出器 [45] により高速な処理が可能であるが、複雑なテクスチャ (木の葉、植え込み等が写り込んでいる領域) を含む画像では過剰にキーポイントが検出される。同じコーナー検出法である Harris コーナー検出器 [14] とキーポイント検出結果を比較すると図 3.1 のような結果となる。FAST キーポイント検出器は、わずか周囲長 16 ピクセルの同心円上の輝度情報のみを用いてキーポイントを検出しているため、テクスチャが複雑な自然領域から過剰にキーポイントを検出してしまう。

このように多くのキーポイントを検出すると2つの問題が発生する。1つ目の問題は、自然領域では視点変化や、風による葉の揺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、画像間で同じキーポイントを検出できないことである。2つ目の問題は、キーポイントマッチングでは1枚目から検出された1つのキーポイントに対して2枚目から検出された全てのキーポイントの特徴量を比較する。従って、検出したキーポイントが多いと特徴量記述や対応点探索の距離計算の計算コストが増加する。例えば、車載カメラや携帯電話端末による物体認識の事例を考える。このような事例の場合、背景に写り込んだ自然領域から多くのキーポイントを検出してしまい、計算コストが非常に高くなる恐れがある。このように物体認識において、認識対象物体以外の自然領域から検出される多くのキーポイントは処理速度の低下を招くため、検出を抑制する必要がある。

そこで、キーポイントマッチングに必要なキーポイントのみを高速に検出する Cascaded FAST を

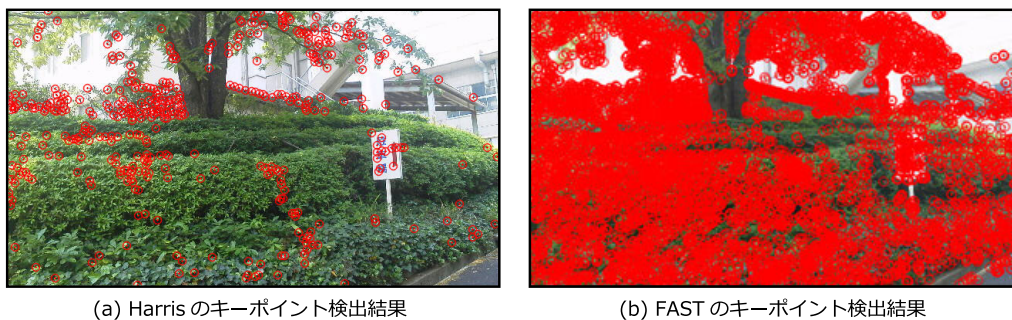
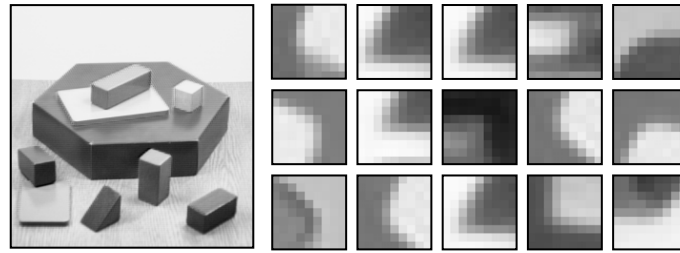
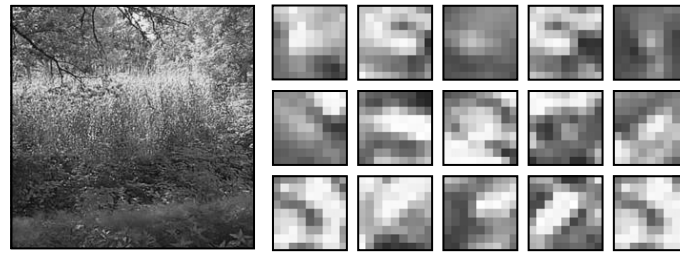


図 3.1: Harris と FAST のキーポイント検出結果の比較。

提案する。提案手法では、FAST コーナー検出器に用いる周囲長 16 ピクセルの同心円上の輝度情報に加え、より広範囲のピクセルを用いた 3 種類の決定木を構築する。そして、3 種類の決定木をカスケード構造に並べることで、高速にキーポイントを検出する。



(a) コーナーらしいアピアランスを持つパッチ画像



(b) コーナーらしくないアピアランスを持つパッチ画像

図 3.2: FAST コーナー検出器により検出されたコーナーのアピアランスの違い。

### 3.1 FAST で検出されるキーポイントの傾向調査

FAST キーポイント検出器は 2.2.3 項で述べた通り，周囲長 16 ピクセルの同心円上の輝度値と注目ピクセルの輝度値を決定木を用いて比較することで高速にキーポイントを検出する。しかし，図 3.1(b) に示すようにテクスチャが複雑な自然領域において過剰にキーポイントを検出する問題がある。ここでは，自然領域から検出されるキーポイントにどのような傾向があるかを調査する。FAST キーポイント検出器は画像中のコーナー点をキーポイントとして検出するため，以降ではキーポイントをコーナーと呼ぶ。まず，検出されたコーナー周辺領域のアピアランスの傾向を調査する。アピアランスの傾向調査では，コーナー点を中心とするパッチ画像を生成し，パッチ画像の見えの違いをコーナーらしい点とコーナーらしくない点で比較する。図 3.2(a) に人工物画像から検出されたコーナーらしいアピアランスを持つパッチ画像，図 3.2(b) に自然画像から検出されたコーナーらしくないアピアランスを持つパッチ画像を示す。人工物画像から検出されたコーナーは類似したアピアランスを持つことが確認できる。一方，自然画像から検出されたコーナーはアピアランスのばらつきが大きい傾向がある。これらのコーナーのアピアランスの傾向から，コーナーらしい点では周囲長 16 ピクセルの同心円の外側と内側の同心円で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異なる周囲長の同心円上の輝度の変化を解析する。図 3.3 に解析対象である同心円上のピクセルを示す。シアン，マゼンタ，青，赤，緑，オレンジの各色で示すピクセルは，それぞれ周囲長 {32, 28, 16, 12, 8} ピクセルの同心円である。パッチ画像の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，パッチ画像の中心ピクセルと各同心円上のピクセルの輝度差を解析する。その際，FAST コーナー検出器のコーナー定義と同様に同心円上のピクセルを brighter, similar, darker の 3 値に分類し， $N_{seg}$  ピクセル以上の brighter も

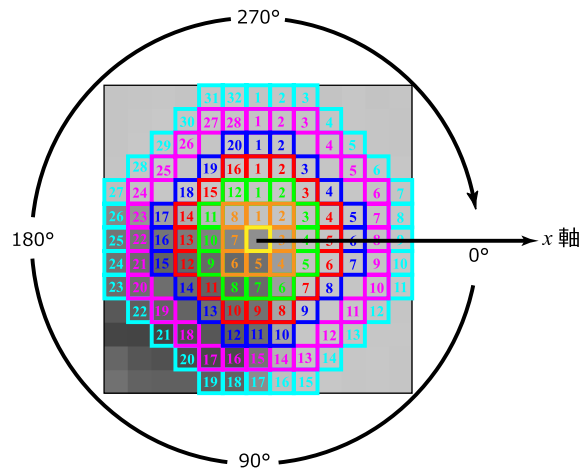


図 3.3: コーナーパッチ画像の解析対象の同心円.

しくは darker が連続するピクセルの始点を基準点とする．そして，注目ピクセルと基準点を結ぶ直線を基準線 (角度  $0^\circ$ ) とする．図 3.4 にコーナーのパッチ画像の解析結果を示す．図 3.4 のグラフの縦軸は注目ピクセルと周囲長の異なる各同心円上のピクセルとの輝度差の絶対値，横軸は基準線からの角度を表す．コーナーらしい点の周囲長  $\{32, 28, 16, 12\}$  ピクセルの同心円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長  $\{32, 28, 16, 12, 8\}$  ピクセルの同心円の差分値にばらつきがある．この解析結果から，周囲長  $\{32, 28, 16, 12\}$  ピクセルの同心円上の輝度情報がコーナーらしい点の検出に有効であると考えられる．しかし，周囲長  $\{32, 28\}$  ピクセルの同心円は，あるコーナー点においては図 3.4(a) のように類似した輝度差が得られるが，同心円の半径が大きいため，図 3.4(b) のように大きな差分値が連続しない場合がある．また，周囲長  $\{32, 28\}$  ピクセルの同心円のように半径が大きくなると画像ピラミッドによりスケールスペースを構築する場合に，小さなサイズの画像からコーナーが検出されなくなる恐れがある．よって，提案手法では周囲長  $\{20, 16, 12\}$  ピクセルの同心円上の輝度情報に基づきコーナーを検出する．

## 3.2 キーポイントの検出方法

3.1 節のコーナーのパッチ画像の調査結果から，コーナーらしい点の周囲長  $\{20, 16, 12\}$  ピクセルの同心円において輝度差の変化の傾向が類似していることを確認した．そこで，提案手法では周囲長  $\{20, 16, 12\}$  ピクセルの同心円の輝度情報に基づいてキーポイントを検出する．さらに，提案手法である Cascaded FAST ではキーポイントの座標に加え，スケールとオリエンテーションも求めることができる．

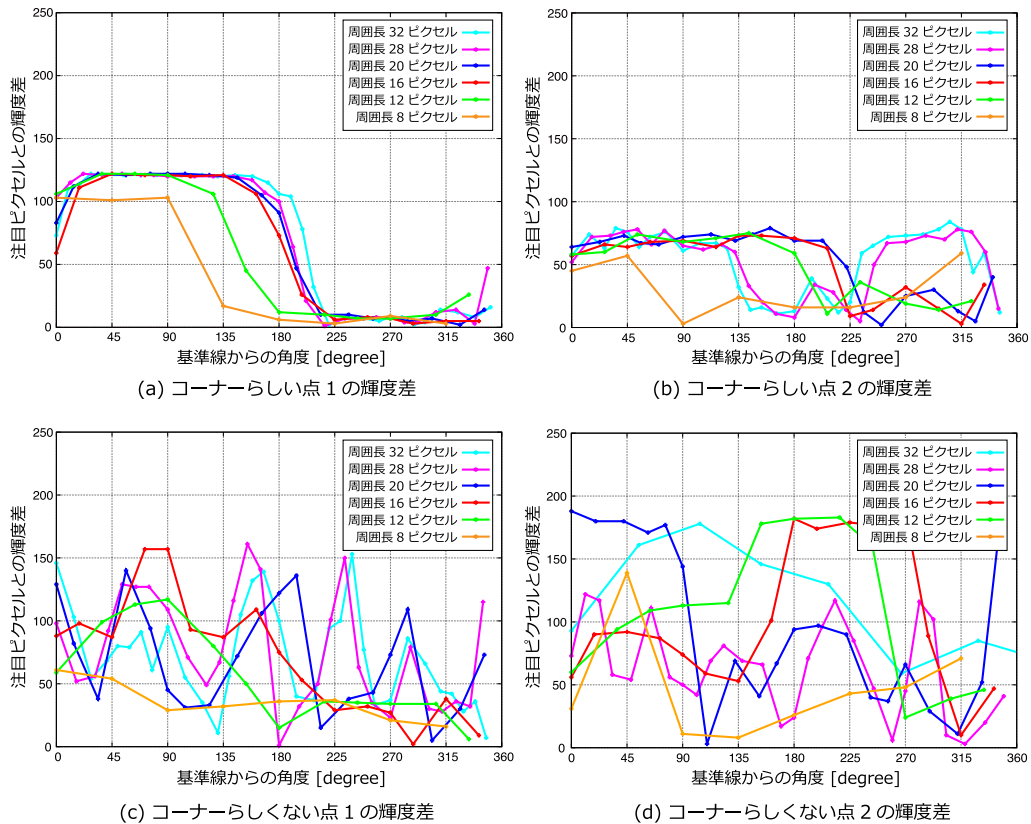


図 3.4: FAST コーナー検出器により検出されたコーナーの解析.

### 3.2.1 コーナーの定義

Cascaded FAST では、周囲長  $\{20, 16, 12\}$  ピクセルの同心円上において brighter または darker に分類されるピクセルが連続しているかという条件に基づきコーナー候補を検出する。そして、検出したコーナー候補に対してオリエンテーションを算出し、周囲長  $\{20, 16, 12\}$  ピクセルの同心円のオリエンテーションが類似している場合に注目ピクセル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

周囲長  $\{20, 16, 12\}$  ピクセルの同心円上のピクセルを FAST コーナー検出器と同様に brighter, similar, darker の 3 値に分類する。FAST コーナー検出器では、周囲長 16 ピクセルの同心円において brighter または darker が 9 ピクセル以上連続した場合に注目ピクセルをコーナーとして検出するが、提案手法では周囲長  $\{20, 16, 12\}$  ピクセルの同心円上において、それぞれ  $\{11, 9, 6\}$  ピクセル以上の brighter または darker が連続している場合に注目ピクセルをコーナー候補とする。周囲長  $\{20, 16, 12\}$  ピクセルの同心円における brighter または darker の連続するピクセル数は、FAST コーナー検出器の周囲長 16 ピクセルの比率に合わせて決定した。



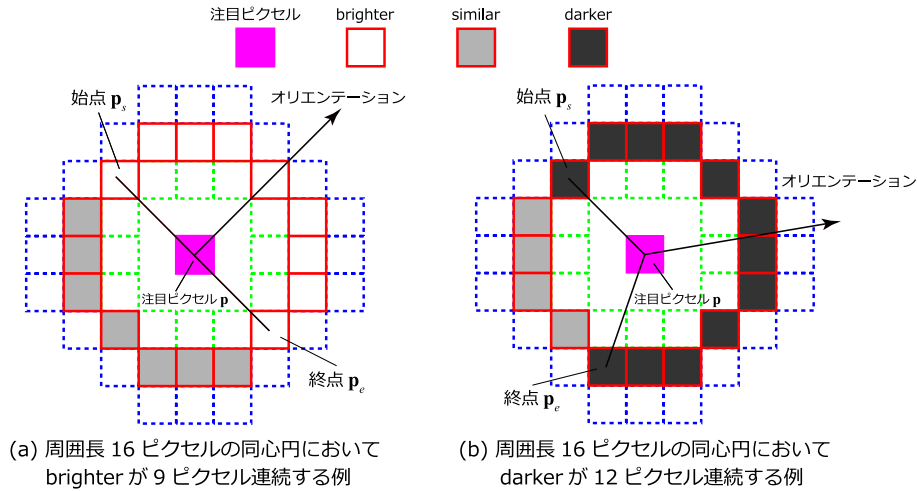


図 3.5: 周囲長 16 ピクセルの同心円におけるオリエンテーションの算出例.

### Step2: オリエンテーションの算出

Step1 で求めたコーナー候補に対して周囲長 {20, 16, 12} ピクセルの同心円におけるオリエンテーションを算出する. 図 3.5 に周囲長 16 ピクセルの同心円におけるオリエンテーションの算出例を示す. まず, brighter または darker が連続するピクセルの始点から終点までの角度を求める.  $x$  軸に対する注目ピクセル  $\mathbf{p}$  と始点のピクセル  $\mathbf{p}_s$  を結ぶ線の角度を  $\theta_s$ ,  $x$  軸に対する注目ピクセル  $\mathbf{p}$  と終点のピクセル  $\mathbf{p}_e$  を結ぶ線の角度を  $\theta_e$  とすると, 始点と終点の角度  $\theta_{s-e}$  は次式により求められる.

$$\theta_{s-e} = \begin{cases} 360^\circ - |\theta_s - \theta_e| & \text{if } \theta_s > \theta_e \\ |\theta_s - \theta_e| & \text{otherwise} \end{cases} \quad (3.1)$$

$$\theta_s = \text{angle}(\mathbf{p}_s, \mathbf{p})$$

$$\theta_e = \text{angle}(\mathbf{p}_e, \mathbf{p})$$

$\text{angle}(\cdot)$  は  $x$  軸を基準とした始点または終点の角度 (degree) を求める関数である. 始点と終点の角度を 2 等分する方向を同心円のオリエンテーション  $\hat{\theta}$  として算出する.

$$\hat{\theta} = \frac{\theta_{s-e}}{2} + \theta_s \quad (3.2)$$

周囲長 {20, 16, 12} ピクセルの同心円において式 (3.2) を用いてオリエンテーションを計算する.

### Step3: オリエンテーションの類似性による条件

検出されたコーナー候補がコーナーらしいアピランスを持つ場合, 周囲長 {20, 16, 12} ピクセルの同心円上の brighter または darker の連続ピクセルは整合性があると考えられる. この整合性を表現するために各周囲長の同心円のオリエンテーションを求め, 各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 3.6 に示すように周囲長 16 ピクセルの同心円のオリエンテ

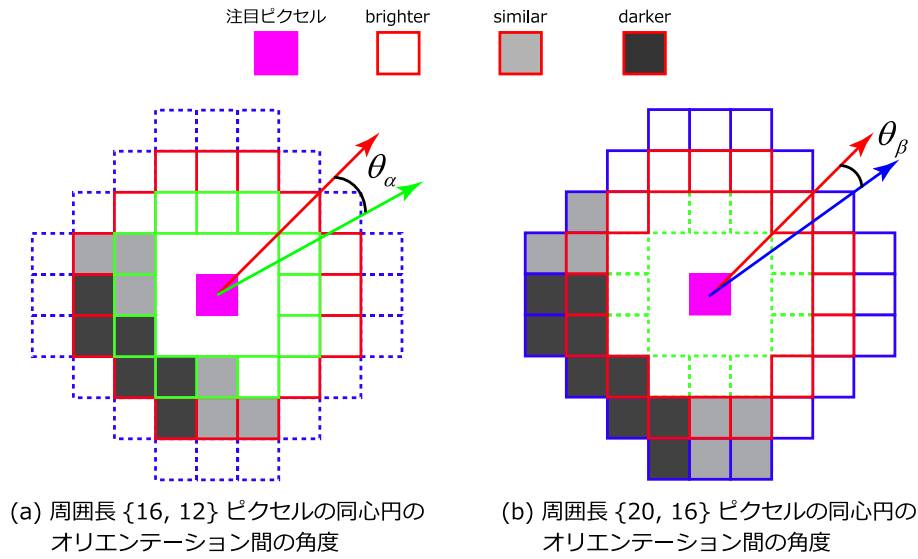


図 3.6: 異なる周囲長の同心円のオリエンテーション間の角度.

ションと周囲長 12 ピクセルの同心円のオリエンテーションの角度差を  $\theta_\alpha$ , 周囲長 20 ピクセルの同心円のオリエンテーションと周囲長 16 ピクセルの同心円のオリエンテーションの角度差を  $\theta_\beta$  として求める. そして,  $\theta_\alpha$  と  $\theta_\beta$  が次式の条件を満たす場合にコーナー候補である注目ピクセル  $\mathbf{p}$  をコーナーとして検出する.

$$\mathbf{p} = \begin{cases} \text{corner} & \text{if } \theta_\alpha \leq T_\alpha \text{ AND } \theta_\beta \leq T_\beta \\ \text{non-corner} & \text{otherwise} \end{cases} \quad (3.3)$$

(3.4)

ここで,  $T_\alpha$  と  $T_\beta$  は各オリエンテーション間角度の閾値である. 周囲長 20 ピクセルのオリエンテーションと周囲長 12 ピクセルのオリエンテーションはそれぞれ分解能が異なるため,  $\theta_\alpha$  と  $\theta_\beta$  に対して別々の閾値を用いる.

### 3.2.2 機械学習による決定木の学習

提案手法においても FAST コーナー検出器と同様に決定木による機械学習を導入することができる. 周囲長 {20, 16, 12} ピクセルの全ての同心円上のピクセルを 1 つの決定木で観測するように学習する方法と, 各同心円上のピクセルを異なる 3 つの決定木で独立して学習する方法が考えられる. 前者の学習方法では観測するピクセル数が増えることにより決定木の階層が深くなり処理速度の低下を招く恐れがある. 一方, 後者の学習方法では観測するピクセル数の少ない決定木を組み合わせることで, 非コーナーを早期棄却することができるため, 周囲長 {20, 16, 12} ピクセルの同心円ごとに独立して決定木を学習させる. 各決定木の学習方法は FAST コーナー検出器と同じで, 学習画像から検

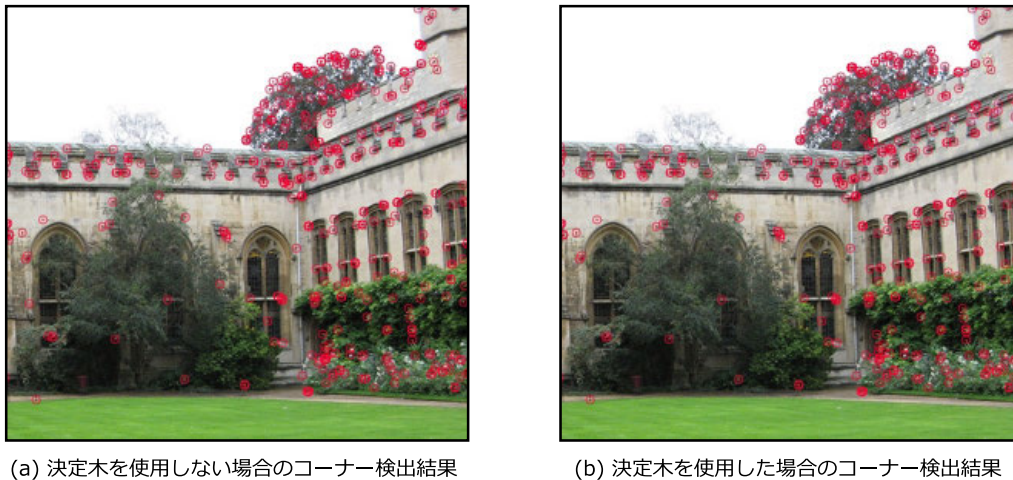


図 3.7: 決定木の有無によるコーナー検出結果の比較.

出された無数のコーナー画像から ID3 のアルゴリズム [46] に基づき、各同心円の 3 種類の決定木を学習する。決定木を学習させるための画像には Caltech-256 Object Category Dataset [62] を使用する。図 3.7 に決定木の有無によるコーナー検出結果の比較を示す。図 3.7(a) は決定木の学習を行わず、周囲長  $\{20, 16, 12\}$  ピクセルの同心円上のピクセルを全て観測することでコーナーを検出した結果である。図 3.7(b) は、周囲長  $\{20, 16, 12\}$  ピクセルの同心円上のピクセルを効率的に観測する決定木を学習させ、学習させた決定木を用いてコーナーを検出した結果である。図 3.7(a) と図 3.7(b) を比較すると、ほぼ同じコーナーを検出できているため、決定木が適切に学習できていることが確認できる。

また、FAST, Cascaded FAST, Harris によりコーナーを検出した結果を図 3.8 に示す。図 3.8(a) に示す FAST コーナー検出器は自然領域からコーナーが過剰に検出されているのに対して図 3.8(b) に示す Cascaded FAST は自然領域からのコーナーの検出を抑制できていることが確認できる。

### 3.2.3 カスケード構造の決定木による高速化

提案手法では異なる周囲長の同心円上のピクセルを観測する 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して出力する。3 つの決定木をカスケード構造に並べ、3 つの決定木のうち 1 つでも非コーナーと判定された場合には早期棄却する。決定木をカスケード構造にすることで、入力画像の約 98% のピクセルを最初の決定木で棄却することができるため、高速な処理が可能である。図 3.9 に Cascaded FAST によるコーナー検出の流れを示す。各決定木における入力ピクセルの棄却率は同等であるが、同心円上のピクセルにアクセスする回数が少ないほど決定木の実行速度が速いため、図 3.9 に示す決定木の配置順が最も高速な処理が可能である。

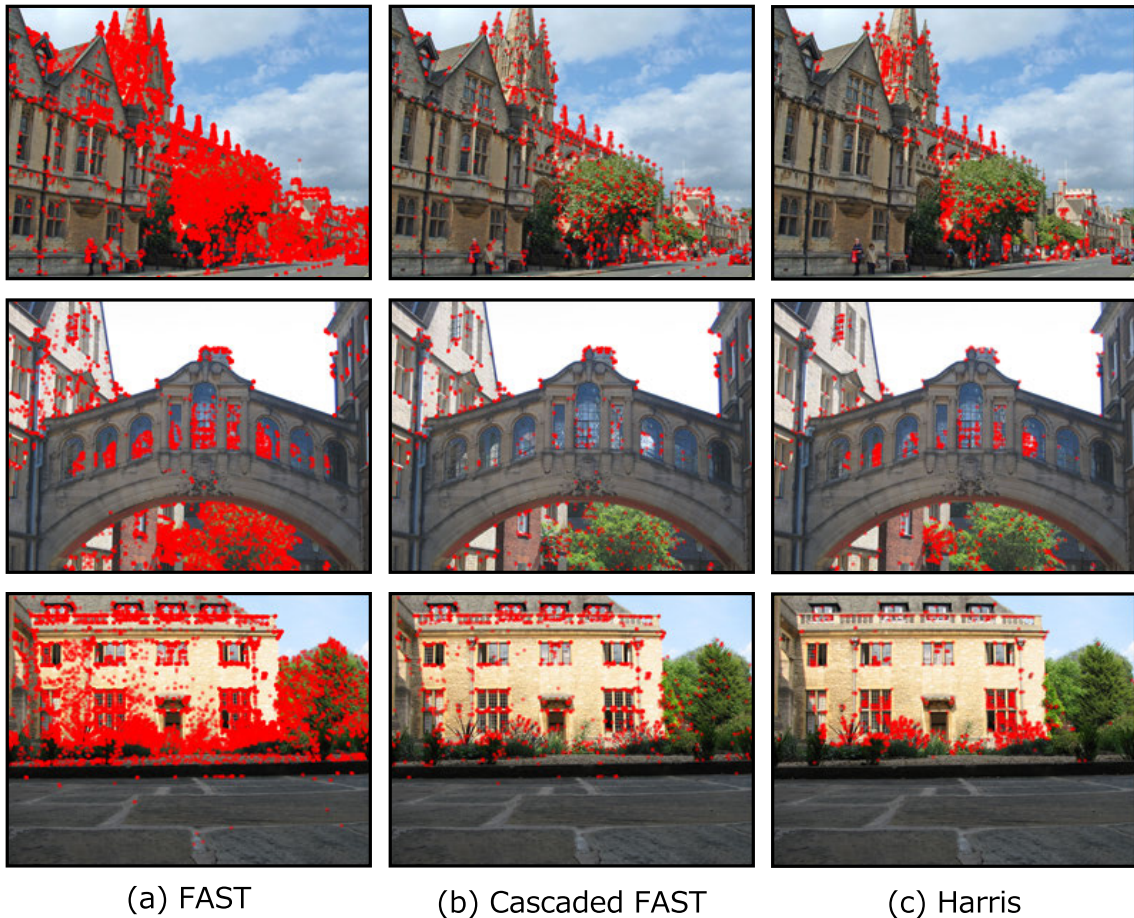


図 3.8: FAST, Cascaded FAST, Harris のコーナー検出結果の比較.

### 3.2.4 スケールとオリエンテーションの獲得

ここまで述べた処理は、コーナーの座標のみしか出力しない。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高速に出力することができる。まず、スケールは複数の解像度で表現される画像ピラミッドから図 3.9 の処理手順でコーナーを検出することで、検出したコーナーの画像解像度をスケールとして利用する。これは、ORB [31] によるキーポイント検出のスケール獲得方法とほとんど同じである。

オリエンテーションについては、3.2.1 項の Step2 により算出されるオリエンテーションをそのまま利用する。図 3.10 は、画像を  $[1^\circ, 359^\circ]$  で回転させたときのオリエンテーションの平均誤差を算出した結果である。例えば、画像を  $45^\circ$  回転させた場合、元画像と回転画像で同一位置のオリエンテーションの角度の差分が  $45^\circ$  であるとき誤差は  $0^\circ$  である。この方法で周囲長  $\{20, 16, 12\}$  ピクセルの同心円の各オリエンテーションと 3 つのオリエンテーションの平均、ORB で用いられているモーメントに基づいて算出されるオリエンテーションを比較した。

各周囲長のオリエンテーションは始点と終点の位置ずれと brighter または darker の連続ピクセル

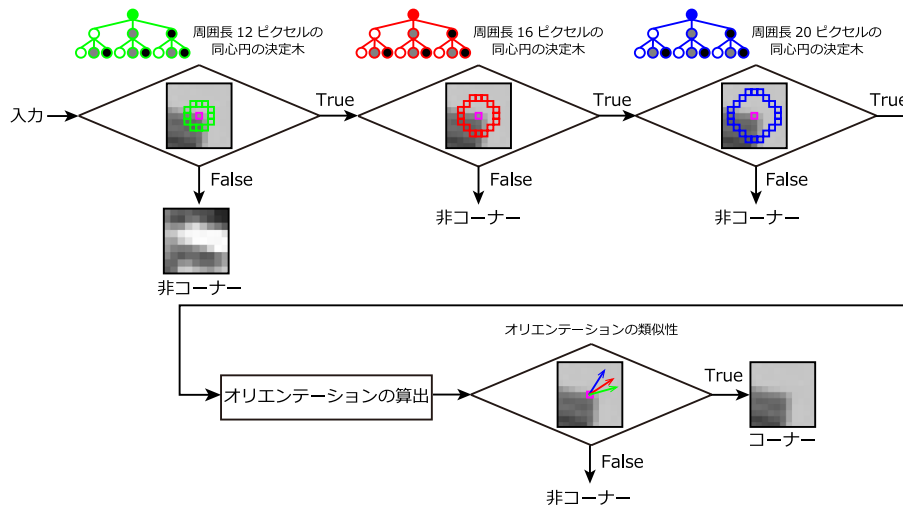


図 3.9: Cascaded FAST によるコーナー検出の流れ.

表 3.1: 各同心円のオリエンテーションの最大誤差と最小誤差.

	周囲長 12 ピクセル	周囲長 16 ピクセル	周囲長 20 ピクセル
最小誤差	13.28°	9.22°	7.02°
最大誤差	36.87°	26.57°	22.62°

数のずれにより誤差が発生する。各周囲長の同心円において始点または終点が 1 ピクセルずれた場合の誤差を表 3.1 に示す。この結果、図 3.10 の各周囲長の同心円のオリエンテーションの平均誤差は各同心円のオリエンテーションの最小誤差より低いため妥当な数値である。また、提案手法である Cascaded FAST の各同心円のオリエンテーションの誤差を見ると、周囲長 20 ピクセルの同心円のオリエンテーションの誤差が最も低いことが確認できる。従って、提案手法では、周囲長 20 ピクセルの同心円のオリエンテーションをキーポイントのオリエンテーションとして採用する。周囲長 20 ピクセルのオリエンテーションは、ORB のモーメントに基づくオリエンテーションと比較して同等以上の精度であるため、キーポイントのオリエンテーションとして十分に利用することができる。図 3.11 に Cascaded FAST により検出されたキーポイントを示す。赤の円の中心はコーナーの座標、円の大きさがスケールの大きさ、青の線がオリエンテーションの方向を表す。図 3.11 から、画像を回転させた場合でもオリエンテーションが同じ方向を推定していることが確認できる。

### 3.3 評価実験

提案手法の有効性を確認するために評価実験を行う。実験では、Harris [14], FAST [45], Cascaded FAST の 3 つのコーナー検出法を比較する。また、実験に使用する計算機の CPU スペックは Intel(R) Xeon(R) X5470 3.33GHz である。

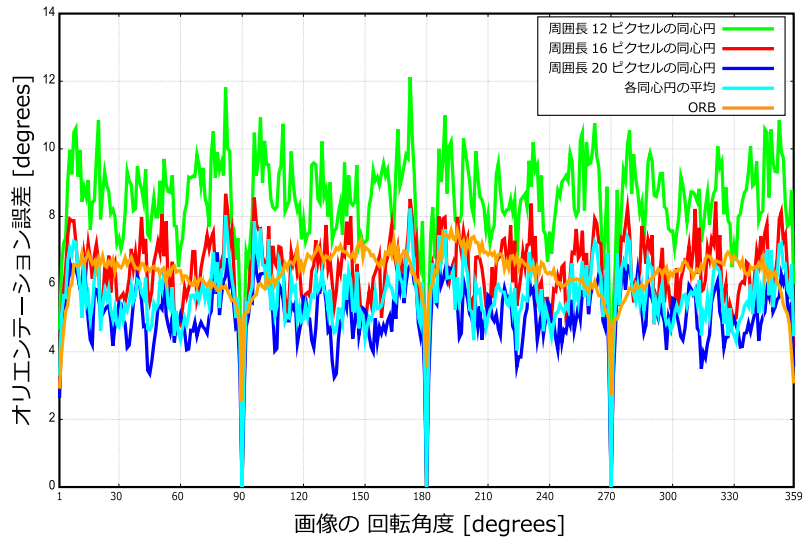


図 3.10: オリエンテーションの評価.

表 3.2: 各手法のコーナー検出時間の比較.

	Harris	FAST	Cascaded FAST	
			決定木あり	決定木なし
コーナー検出時間 [ms]	164.4	4.5	7.4	216.9
コーナー数	1134	8580	1197	1197

### 3.3.1 コーナー検出時間の評価

ここでは、Harris, FAST, Cascaded FAST のコーナー検出時間を比較する。本実験では各手法のコーナー検出のみの時間を比較するため、スケールとオリエンテーションは計算しない。実験画像は、Oxford buildings dataset [63] から人工物と自然領域を含む画像 (1024 × 768 ピクセル) 100 枚を使用する。表 3.2 に各手法のコーナー検出時間の比較を示す。表 3.2 は各手法において画像 1 枚あたりの平均処理時間と検出した平均コーナー数を示している。Cascaded FAST のコーナー検出時間は FAST のコーナー検出時間に比べて 2.9 [ms] 増加するが、Harris と比較して約 22 倍高速にコーナーを検出できる。また、決定木を用いた探索によるコーナー検出は、決定木を使用しない場合と比較して約 29 倍の高速化が可能であり、フレームレートは約 135 [fps] である。

### 3.3.2 F 値による評価

コーナーらしくない点の検出を抑制できているかを確認するため、Cascaded FAST と FAST により検出されたコーナーの F 値を比較する。また、Cascaded FAST のオリエンテーションの類似性による条件の効果を確認するため、オリエンテーションによる判定を使用しない手法を比較する。データセットは Oxford buildings dataset [63] を使用する。検出するべきコーナーと抑制するべきコーナー



図 3.11: Cascaded FAST によるキーポイント検出例.

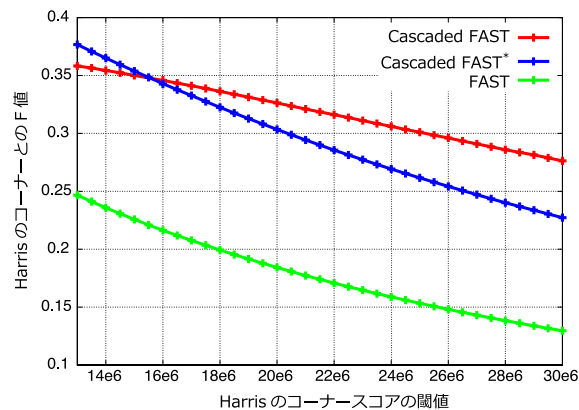


図 3.12: Cascaded FAST と FAST の F 値の比較.

を正確に定義することができないため、ここでは Harris コーナー検出器により検出されたコーナーを真値として利用し、Cascaded FAST と FAST のコーナーの F 値を評価に用いる。

Harris コーナー検出器は注目ピクセルの局所領域内のエッジ情報の分布に基づいてコーナーを定義しているため、適切な閾値によりスコアの高いコーナーのみを出力することで図 3.1(a) に示すように自然領域から検出されるコーナーを抑制し、信頼性の高いコーナーを検出することができる。図 3.12 に Cascaded FAST と FAST により検出されたコーナーの F 値を示す。グラフの縦軸に各手法の Harris のコーナーに対する F 値を示し、横軸は Harris のコーナースコアの閾値を示す。また、青の実線で示す Cascaded FAST\* はオリエンテーションの類似性による判定をしないコーナー検出器である。図 3.12 の実験結果から、Cascaded FAST は FAST より F 値が高いことが確認できるため、コーナーらしい点のみを検出できていると考えられる。また、Cascaded FAST はオリエンテーションの類似性による判定をしない場合、Harris のスコアの閾値を高くするほど F 値が大きく低下する。Harris のコーナースコアの閾値が高いほどコーナーの信頼性が高いといえるため、オリエンテーションの類似性によりコーナーを検出することは有効である。

表 3.3: 比較手法の詳細.

手法	オリエンテーション算出方法	スケール取得方法	特徴量記述子
Harris	ORB (パッチ画像のモーメント)	画像ピラミッド	ORB
FAST	ORB (パッチ画像のモーメント)	画像ピラミッド	ORB
Cascaded FAST <sup>M</sup>	ORB (パッチ画像のモーメント)	画像ピラミッド	ORB
Cascaded FAST	Cascaded FAST (周囲長 20 ピクセル)	画像ピラミッド	ORB

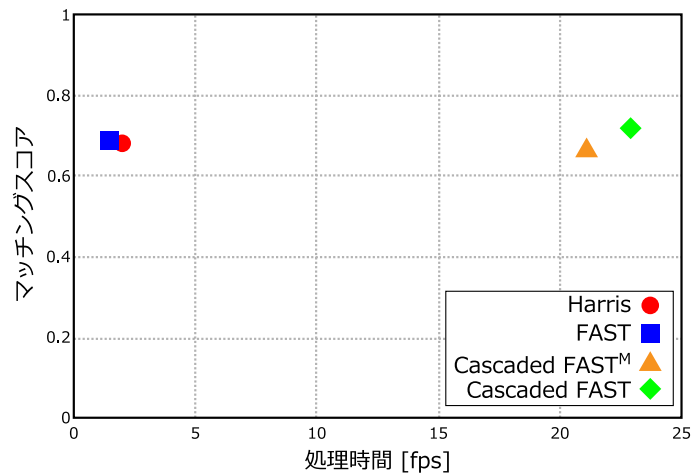


図 3.13: マッチングスコアと処理時間の比較.

### 3.3.3 キーポイントマッチングにおける精度と速度

ここでは、キーポイントマッチングの精度と処理時間を評価する。比較手法を表 3.3 に示す。キーポイントのスケール取得方法は、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方法は ORB [31] で用いられるコーナーパッチ画像のモーメントに基づく手法と Cascaded FAST の周囲長 20 ピクセルのオリエンテーションの 2 種類である。特徴量記述子は全ての手法において ORB [31] を使用する。本実験では、Oxford buildings dataset [63] の画像 130 枚に対してランダムに 10 種類のアフィン変換を施した 1,300 枚の画像を使用する。アフィン変換画像を用いることで、画像間の同一キーポイントの同定が容易にできる。キーポイントマッチングは、2.1 節で説明した方法で行い、式 (2.4) を満たす場合に画像間のキーポイントを対応点とする。距離計算はハミング距離を使用する。画像間のキーポイントが対応点である場合、2 点の座標が物理的に同一の位置であることを判定する。対応点の座標が  $\sqrt{(1+1)}$  ピクセル以内の位置ずれの場合に正解点とする。キーポイントマッチングの精度の評価は、マッチングスコア (= 正解点数 / 対応点数) を用いる。そして、マッチングスコアとフレームレート (fps) を各手法で比較する。

図 3.13 に各手法のマッチングスコアと速度の比較を示す。Cascaded FAST は他の手法と比較してマッチングスコアは同等でありながら、フレームレートが高いことが確認できる。図 3.14 にキーポイントマッチングの処理時間の内訳を示す。Harris コーナー検出器は、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 コーナー検出器はコーナー検出とオリエンテーションの算出は高速に処理することが可能であるが、特徴量の記述と対応点探索に多大な計算時間を必要とする。



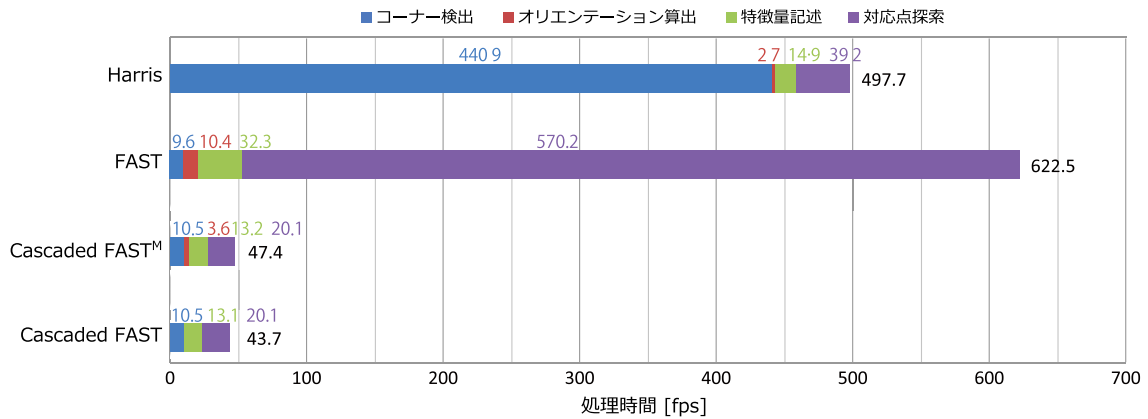


図 3.14: キーポイントマッチングの処理時間の内訳.

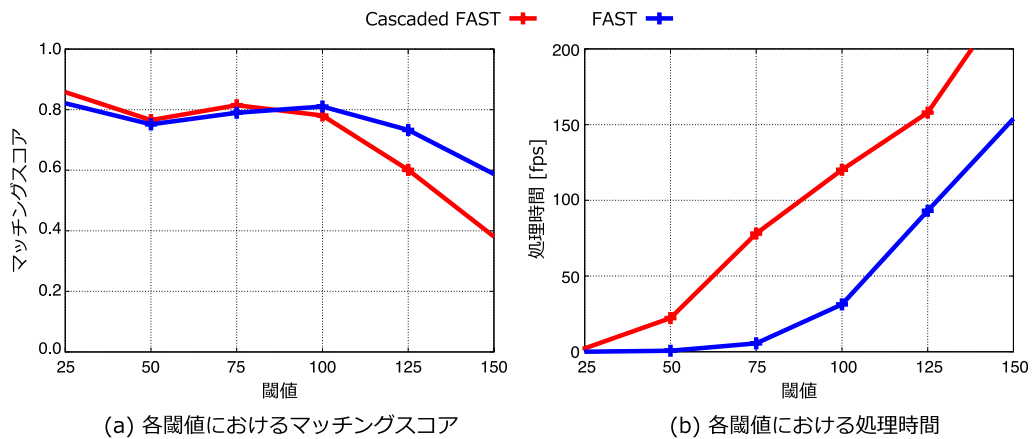


図 3.15: 各閾値における Cascaded FAST と FAST の性能.

この理由として、FAST コーナー検出器は他の手法と比較して自然領域からコーナーを過剰に検出してしまふためである。提案手法である Cascaded FAST では、Harris や FAST と比較してキーポイントマッチングを高速に実行できていることが確認できる。特に、周囲長 20 ピクセルの同心円上の輝度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ため、別処理によるオリエンテーション算出を省くことができる。周囲長 20 ピクセルの同心円上の輝度情報によるオリエンテーションを用いた Cascaded FAST は、2 画像間のキーポイントマッチングを約 43.7 [ms] で処理することが可能である。

また、Cascaded FAST と FAST はコーナー検出の閾値を変化させると性能も変化するため、様々な閾値によるマッチングスコアと処理時間を比較する。この閾値は同心円上のピクセルを brighter, similar, darker に分類する際の閾値であり、式 (2.16) の  $T_f$  である。図 3.15(a) は閾値を変化させたときのマッチングスコアを示す。閾値が 25 ~ 100 の範囲では Cascaded FAST と FAST は同等の精度であるが、閾値をより高くすると検出されるコーナー数が極端に減少するため、マッチングスコアが低下する。図 3.15(b) は閾値を変化させたときの処理時間を比較した結果である。閾値を低く設定

すると検出されるコーナー数が多いため、フレームレートが低くなり、閾値を高く設定すると検出されるコーナー数が少なくなり、フレームレートが高くなる。図 3.15 の結果から、マッチングの精度を維持しつつ高速な処理を行うためには、閾値は 50 ~ 100 の範囲が妥当であると考えられる。

## 3.4 まとめ

本章では、テクスチャが複雑な自然領域からキーポイントの過剰な検出を抑制することで高速にキーポイントマッチングを行う Cascaded FAST について述べた。Cascaded FAST は周囲長 {20, 16, 12} ピクセルの 3 種類の同心円上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自然領域から検出されるキーポイントを抑制した。自然領域から検出されるキーポイントを抑制することで、キーポイントマッチングにおいて計算コストの増加を抑え、Harris や FAST によるコーナー検出法と比較して高速なキーポイントマッチングが可能であることを確認した。Cascaded FAST によるキーポイントマッチングは約 23 [fps] で動作するため、十分にリアルタイム処理が可能である。

## 第4章

# 非等方性LoGフィルタによる複数のアフィン領域の推定

本章では、非等方性 LoG フィルタにより画像から検出されたキーポイントにおいて複数のアフィン領域を推定する手法を提案する。従来手法である Harris-Affine や Hessian-Affine によるアフィン領域推定 [24] では、非等方性ガウシアンフィルタをキーポイントのパッチ画像に繰り返し畳み込むことで、アフィン領域を推定する。

キーポイントにおける局所領域内には複数のアフィン領域が存在する可能性がある。例えば図 4.1 に示すように 2 つの単純な楕円パターンが重なっている場合は各楕円や 2 つの楕円を囲むような複数のアフィン領域が存在することになる。Harris-Affine や Hessian-Affine はキーポイントに対して 1 つのアフィン領域しか推定しないため、画像の変形やキーポイントの位置ずれの影響により異なるアフィン領域を推定してしまうことがある (図 4.1 上段)。また、輝度値に基づいた領域分割によりアフィン領域を推定する MSER では、図 4.1 中段に示すように輝度変化に弱い。

提案手法では様々な楕円形状の非等方性 LoG フィルタを用いてキーポイントに対して複数のアフィン領域を推定する (図 4.1 下段)。しかし、非等方性 LoG フィルタは  $x$  方向のスケール  $\sigma_x$ 、 $y$  方向のスケール  $\sigma_y$ 、フィルタの回転角  $\theta$  の 3 パラメタの組み合わせにより数千種類となる。複数のアフィン領域を推定するには、数千種類の非等方性 LoG フィルタを全てキーポイントの局所領域に畳み込









	単純パターンのテスト	キーポイントの位置ずれ ( $x: +1$ px, $y: +2$ px)	グレースケールパターン
Hessian-Affine			
MSER		—	
提案手法			

図 4.1: 単純楕円パターンによるアフィン領域推定の比較。

む必要があり、膨大な計算コストが必要となる。

そこで、提案手法では Spectral SIFT [20] の考え方を導入することで非等方性 LoG フィルタの畳み込みを効率的に計算する。Spectral SIFT ではガウシアンスケールスペースや LoG スケールスペースに対してスペクトル分解を適用することで、スケールスペースの固有値問題を解き、基底となる少数のフィルタと係数の線形演算でスケールスペースを再構成する。これは、多少の誤差を許すことで離散的な特異値分解 (SVD) で代用することができる。SVD を用いることで数千種類の非等方性 LoG のフィルタ群の固有解を簡単に求めることができ、わずか 14 種類の固有フィルタで数千種類の非等方性 LoG フィルタを近似することができる。そのため、非等方性 LoG フィルタの畳み込みを効率的に計算することが可能となり、複数のアフィン領域を効率的に求められる。

## 4.1 複数のアフィン領域推定

提案手法では、Spectral SIFT の考え方に基づいて複数のアフィン領域を推定する。Spectral SIFT では、次式に示すガウシアンスケールスペースや LoG スケールスペースの固有解を導出している。

$$\begin{aligned} \text{LoG}(\sigma) &= \left( \frac{\partial^2}{\partial x^2} g(\sigma) + \frac{\partial^2}{\partial y^2} g(\sigma) \right) \sigma^2 \\ &= \frac{\bar{x}^2 + \bar{y}^2 + 2\sigma^2}{2\pi\sigma^4} \exp\left(-\frac{\bar{x}^2 + \bar{y}^2}{2\sigma^2}\right) \end{aligned} \quad (4.1)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\bar{x}^2 + \bar{y}^2}{2\sigma^2}\right) \quad (4.2)$$

式(4.1)や式(4.2)は、等方性フィルタであるためアフィン領域の推定には使用することができない。従って、提案手法ではアフィン領域を推定するために非等方性フィルタを利用する。SVDにより非等方性フィルタ群の固有解(固有フィルタと固有関数)を導出することで、効率的に複数のアフィン領域を推定する。

### 4.1.1 非等方性 LoG スケールスペースの近似

まず、非等方性 LoG フィルタを生成する。

$$\text{LoG}(\sigma_x, \sigma_y, \theta) = \frac{\partial^2}{\partial x^2} g(\Sigma) + \frac{\partial^2}{\partial y^2} g(\Sigma) \quad (4.3)$$

$$g(\Sigma) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{\bar{\mathbf{p}}^\top \Sigma^{-1} \bar{\mathbf{p}}}{2}\right) \quad (4.4)$$

$$\Sigma = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

ここで、 $g(\Sigma)$  は非等方性ガウシアンフィルタであり、 $\bar{\mathbf{p}} = [\bar{x}, \bar{y}]^\top$  はフィルタの中心からの距離である。非等方性 LoG フィルタの  $x$  方向のスケール  $\sigma_x$ 、 $y$  方向のスケール  $\sigma_y$ 、フィルタの回転角  $\theta$  は次式のように設定する。

$$\sigma_x = \{1.6, 1.7, 1.8, \dots, 3.2\} \quad (4.5)$$

$$\sigma_y = \{1.6, 1.7, 1.8, \dots, 3.2\} \quad (4.6)$$

$$\theta = \{0^\circ, 5^\circ, 10^\circ, \dots, 175^\circ\} \quad (4.7)$$

LoG フィルタの1辺のサイズを  $N_m = 19$  とし、LoG フィルタの枚数  $N_a$  は  $\sigma_x, \sigma_y, \theta$  の組み合わせにより 4,913 枚となる。4,913 枚の LoG フィルタを SVD により 3つの行列に分解することを考える。

$$\mathbf{W} = \mathbf{USV}^\top \quad (4.8)$$

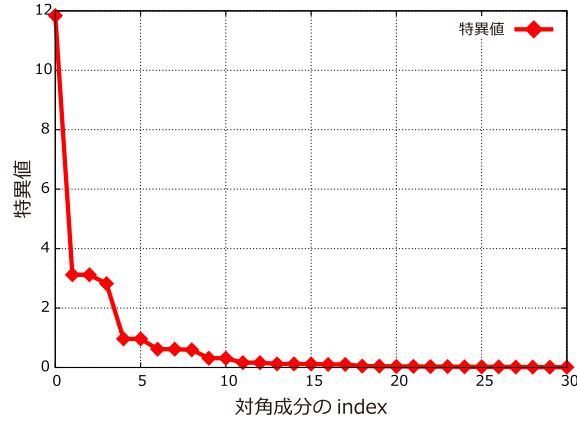


図 4.2: 行列  $\mathbf{S}$  の対角成分.

ここで、行列  $\mathbf{W} \in \mathbb{R}^{N_m^2 \times N_a}$  は、ベクトル化した 4,913 枚の LoG フィルタを各列に並べた行列である。行列  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_{N_m^2}]$  は、行列  $\mathbf{W}$  の固有ベクトルで構成され、各固有ベクトル  $\mathbf{u}$  は 2 次元のフィルタと見なせるためここでは固有フィルタと呼ぶ。行列  $\mathbf{S}$  と行列  $\mathbf{V}^\top$  の積を  $\mathbf{SV}^\top = [\rho_1 \ \rho_2 \ \cdots \ \rho_{N_a}]$  と表記すると、 $\rho$  は各固有フィルタに対する重み係数の役割となり、ここは固有関数と呼ぶ。アフィンパラメータ  $(\sigma_x, \sigma_y, \theta)$  における非等方性 LoG フィルタ  $LoG(\sigma_x, \sigma_y, \theta)$  は次式により近似することができる。

$$\begin{aligned}
 LoG(\sigma_x, \sigma_y, \theta) &= \sum_{n=1}^{N_m^2} \rho_n(\sigma_x, \sigma_y, \theta) \mathbf{u}_n \\
 &\approx \sum_{n=1}^{14} \rho_n(\sigma_x, \sigma_y, \theta) \mathbf{u}_n
 \end{aligned} \tag{4.9}$$

$\rho_n(\sigma_x, \sigma_y, \theta)$  はベクトル  $\rho_n$  のパラメータ  $(\sigma_x, \sigma_y, \theta)$  に対応するスカラー値である。行列  $\mathbf{S}$  は対角成分に特異値を持つ行列であり、対角成分をプロットすると図 4.2 のようになる。行列  $\mathbf{S}$  の特異値は 15 番目以降に極めて 0 に近い値であり、上位 14 個の特異値で累積寄与率が 96.7% となる。従って、非等方性 LoG フィルタ  $LoG(\sigma_x, \sigma_y, \theta)$  を 14 種類の主要な固有フィルタのみで十分に近似することができる (式 (4.9))。図 4.3 に SVD による非等方性 LoG フィルタの近似を示す。また、SVD により得られた固有フィルタと固有関数を図 4.4 に示す。固有関数は  $\sigma_x, \sigma_y, \theta$  の 3 次元空間上の値であるため、図 4.4 では  $\theta = 45^\circ$  に固定した場合の固有関数を示す。

#### 4.1.2 非等方性 LoG フィルタの応答値の算出

固有フィルタ  $\mathbf{u}_n$  と固有関数  $\rho_n(\cdot)$  を用いて非等方性 LoG フィルタの応答値  $R_{LoG}$  を計算する。非等方性 LoG フィルタの応答値  $R_{LoG}$  は式 (4.9) にキーポイント周辺のパッチ画像  $\mathbf{I}$  の畳み込みを行う

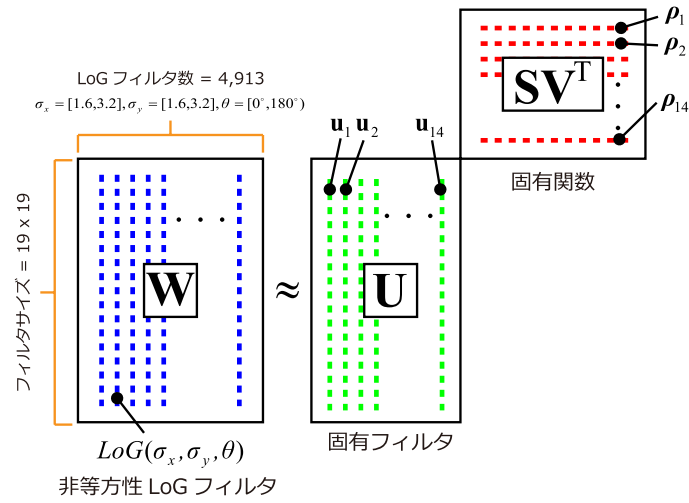


図 4.3: SVD による非等方性 LoG フィルタの近似.

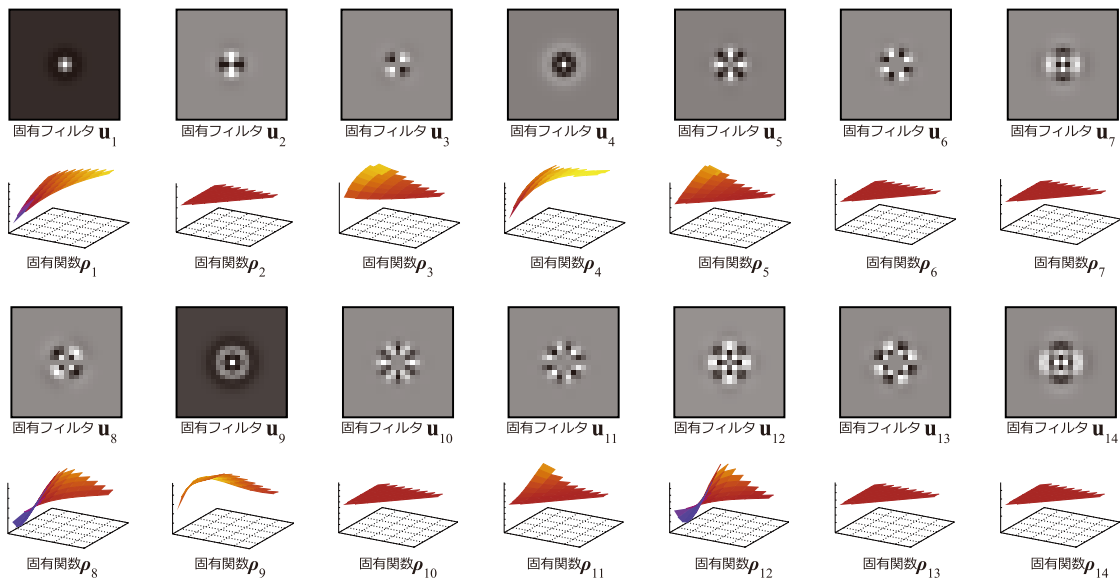


図 4.4: 固有フィルタと固有関数.

ことで計算することができる (式 (4.10)).

$$R_{LoG}(\sigma_x, \sigma_y, \theta) \approx \mathbf{I} * \sum_{n=1}^{14} \rho_n(\sigma_x, \sigma_y, \theta) \mathbf{u}_n \quad (4.10)$$

$$\approx \sum_{n=1}^{14} \rho_n(\sigma_x, \sigma_y, \theta) \eta_n \quad (4.11)$$

ここで、パッチ画像  $\mathbf{I}$  と固有フィルタ  $\mathbf{u}_n$  との畳み込み結果を  $\eta_n = \mathbf{I} * \mathbf{u}_n$  とすると、式 (4.10) は分配法則を適用することができ、式 (4.11) で計算できる。式 (4.11) から、14 種類の固有フィルタとパツ

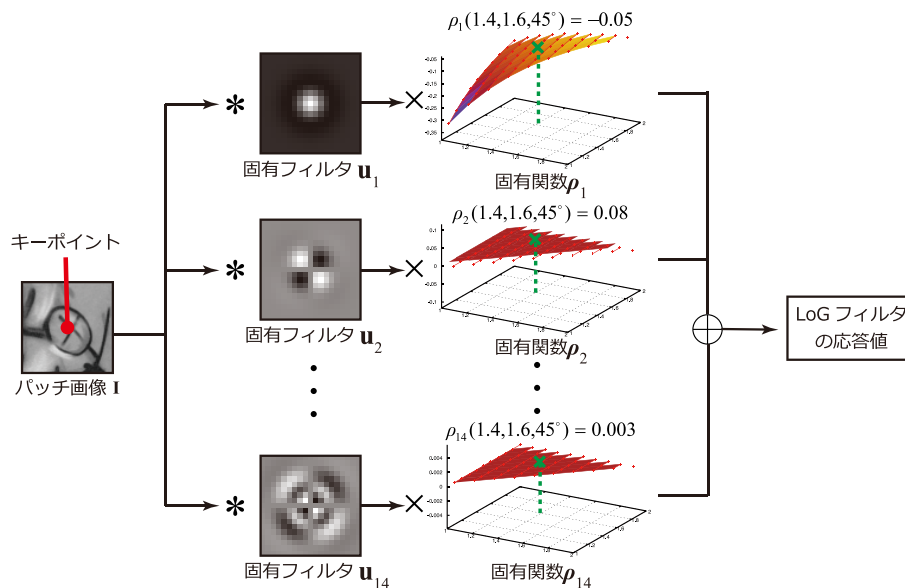


図 4.5: 非等方性 LoG フィルタの応答値計算の流れ.

チ画像はあらかじめ畳み込み計算が可能であることがわかる。事前に  $\eta_n$  を計算しておいた後、求めたいフィルタ応答値のパラメータに応じて固有関数の値のみを変えるだけで容易に応答値を計算することができる。このように、提案手法では計算コストの高い畳み込み処理をわずか 14 回に抑えることができるため、非常に効率的である。図 4.5 に  $\sigma_x = 1.4, \sigma_y = 1.6, \theta = 45^\circ$  の非等方性 LoG フィルタの応答値算出の流れを示す。

### 4.1.3 固有関数の連続関数フィッティング

式 (4.11) から、4,913 種類の非等方性 LoG フィルタの応答値を 14 種類の固有フィルタと固有関数で近似することが可能となった。しかし、固有関数の値は離散的な値であるため、SVD の適用前に生成した 4,913 種類の LoG フィルタのパラメータでしか近似することができない。そこで、固有関数を連続的な関数モデルでフィッティングすることを考える。固有関数  $\rho_n(\cdot)$  は図 4.4 で示したように、3次元空間上の係数である。そのため、SVD により得られた離散的な固有関数  $\rho_n(\cdot)$  と 3変数の連続的な関数モデル  $q_n(\cdot)$  との最小化問題を解く。連続関数モデルはスケールパラメータ  $\sigma_x, \sigma_y$  のべき級数で設計する。しかし、回転角  $\theta$  における固有関数は周期的な波形となる。図 4.6 に固有関数  $\rho_5(\cdot)$  の数値を示す。固有フィルタ  $u_5$  はフィルタの回転角  $\theta$  に不変でないため、固有関数の数値が  $\theta$  に応じて変化していることが確認できる (図 4.6 上段)。そして、スケールパラメータ  $\sigma_x, \sigma_y$  をそれぞれ固定したときの固有関数の数値は周期関数となっていることがわかる (図 4.6 下段)。このような周期関数の場合、べき級数では関数のフィッティングが困難となる。従って、回転パラメータ  $\theta$  に関



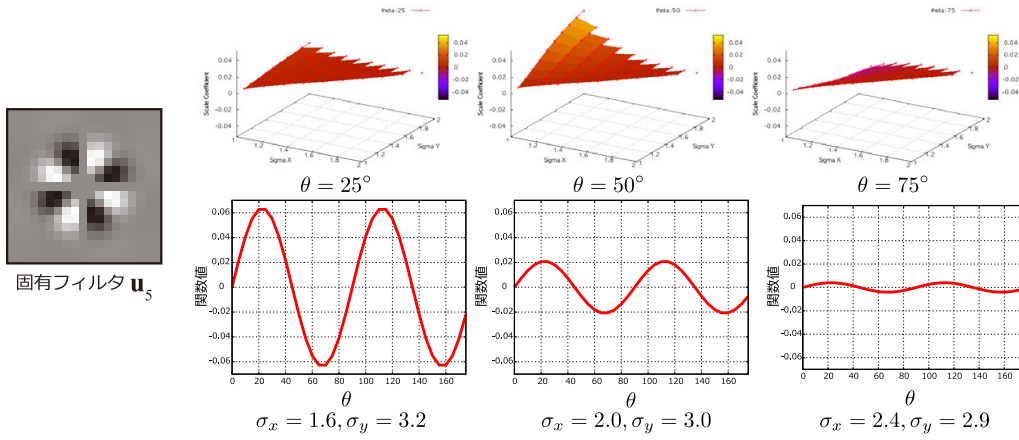


図 4.6: パラメータを固定した際の固有関数  $\rho_5(\cdot)$  の数値.

しては三角関数を用いることで次式のような連続関数モデル  $\varrho_n(\cdot)$  をフィッティングに使用する.

$$\varrho_n(\sigma_x, \sigma_y, \theta) = \sum_{i=0}^{D_I} \sum_{j=0}^{D_J} \sum_{k=0}^{D_K} \alpha_{ijk} \sigma_x^i \sigma_y^j \cos(k\theta) + \sum_{i=0}^{D_I} \sum_{j=0}^{D_J} \sum_{k=1}^{D_K} \beta_{ijk} \sigma_x^i \sigma_y^j \sin(k\theta) \quad (4.12)$$

ここで,  $\alpha, \beta$  は未知係数であり, 次式により離散固有関数  $\rho_n(\cdot)$  と連続関数モデル  $\varrho_n(\cdot)$  との 2 乗誤差が最小となる  $\alpha, \beta$  を決定する.

$$\arg \min_{\alpha, \beta} \left( \sum_{\sigma_x} \sum_{\sigma_y} \sum_{\theta} (\rho_n(\sigma_x, \sigma_y, \theta) - \varrho_n(\sigma_x, \sigma_y, \theta))^2 \right) \quad (4.13)$$

$$\sigma_x = \{1.6, 1.7, 1.8, \dots, 3.2\}$$

$$\sigma_y = \{1.6, 1.7, 1.8, \dots, 3.2\}$$

$$\theta = \{0^\circ, 5^\circ, 10^\circ, \dots, 175^\circ\}$$

連続固有関数の次数  $D_I, D_J, D_K$  は, LoG フィルタを十分に近似できるように設定する. 図 4.7 に様々な次数の連続固有関数で非等方性 LoG フィルタを再構成したときの近似誤差を示す. 連続固有関数の次数が  $D_I = 3, D_J = 3, D_K = 4$  のとき, 連続関数フィッティングをしない場合の近似誤差と同等であるため十分に近似できているといえる. 従って, 連続固有関数  $\varrho_n(\cdot)$  の次数は  $D_I = 3, D_J = 3, D_K = 4$  とする.

このように, 固有関数を連続的な関数で近似することで, 非等方性 LoG フィルタの応答値  $R_{LoG}$  においても次式のように連続的な関数として表現することができる.

$$R_{LoG}(\sigma_x, \sigma_y, \theta) \approx \sum_{n=1}^{14} \varrho_n(\sigma_x, \sigma_y, \theta) \eta_n \quad (4.14)$$

この結果, 任意の連続パラメータ  $(\sigma_x, \sigma_y, \theta)$  における非等方性 LoG フィルタの応答値  $R_{LoG}$  を求めることが可能となる.

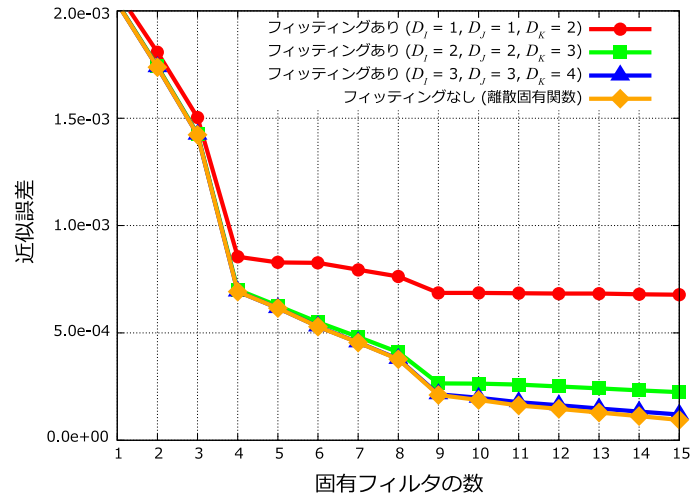


図 4.7: 連続固有関数の次数による非等方性 LoG フィルタの近似誤差.

#### 4.1.4 複数のアフィン領域の探索

ここでは、フィルタ応答値の極値探索により複数のアフィン領域を推定する手順を説明する。入力パッチ画像に複数のアフィン領域が存在する場合、フィルタの応答値が極値となるパラメータ  $(\sigma_x, \sigma_y, \theta)$  も複数存在する。そのため、3次元パラメータ空間上のフィルタ応答値に存在する複数の極値を探索して検出しなければならない。図 4.1 のような単純な楕円パターンで考えると、 $\sigma_x, \sigma_y$  のスケール方向に対してはアフィン領域が1つに決まる。しかし、 $\theta$  方向に対しては複数のアフィン領域が存在すると考えられる。そのため、 $\sigma_x, \sigma_y$  方向に対しては1つのアフィン領域を決定し、 $\theta$  方向に対しては複数のアフィン領域が存在する場合、それらを全て検出する。提案手法における複数の極値探索方法を図 4.8 に示す。まず、 $\theta$  軸を分割することで2次元空間  $(\sigma_x, \sigma_y)$  の極値 (図中 × 印) を各  $\theta$  から検出する。そして、各  $\theta$  から検出した2次元空間上の極値を用いて3次元空間  $(\sigma_x, \sigma_y, \theta)$  の極値 (図中 ○ 印) を求める。3次元空間において複数の極値が得られた場合、最大極値から90%以上の応答値を持つ極値を全てキーポイントのアフィン領域として採用する。また、式 (4.14) に示す応答値の算出は連続固有関数を使用しているため、微分可能な関数である。よって、 $\theta$  軸を分割した2次元空間の極値探索にはニュートン法を用いて高速に処理することができる。図 4.8 の例では、3次元空間上の極値が3つ存在するが、最終的には大きな応答値を持つ2つの極値がアフィン領域として検出される。

図 4.9 に提案手法によるキーポイントの複数のアフィン領域の推定結果を示す。図 4.9 の結果から、提案手法では単一のキーポイントに対して複数のアフィン領域を推定できていることが確認できる。また、提案手法により推定したアフィン領域は視点の異なる画像間で同じ領域を推定していることがわかる。

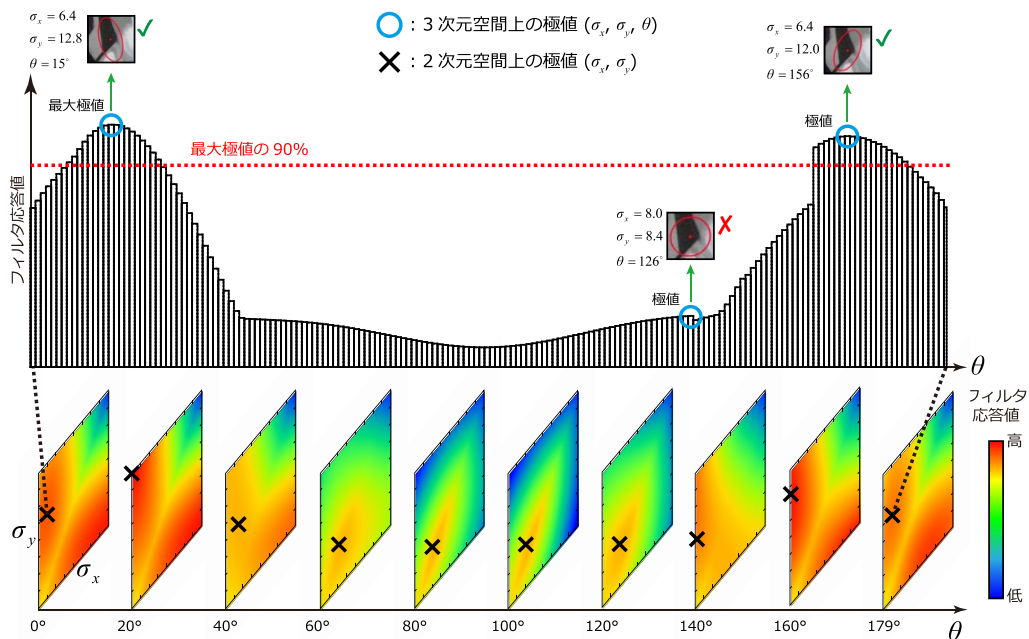


図 4.8: 複数のアフィン領域の探索.

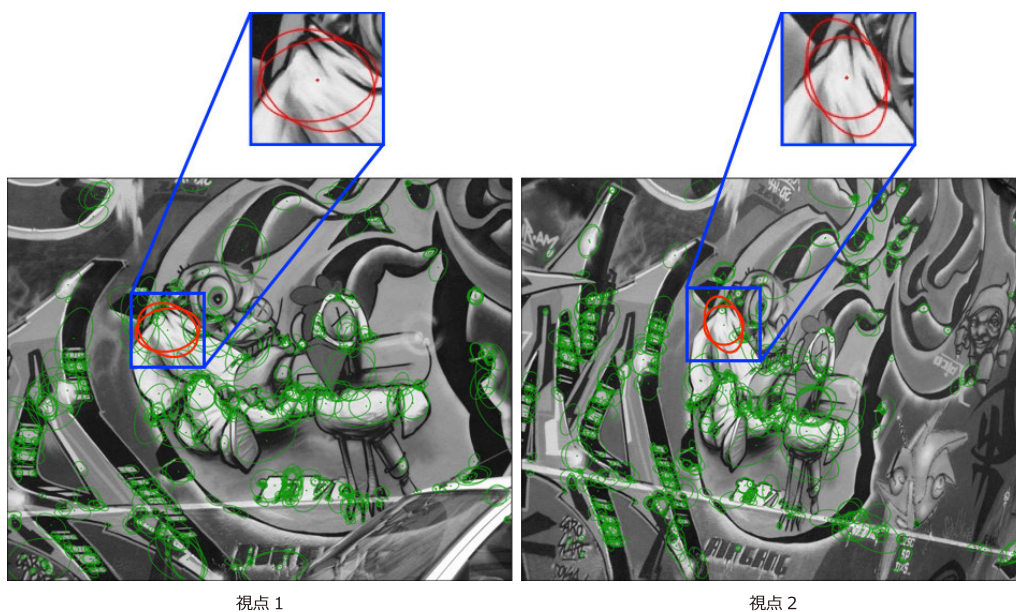
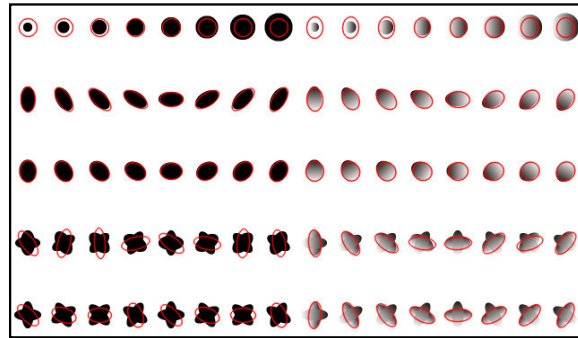


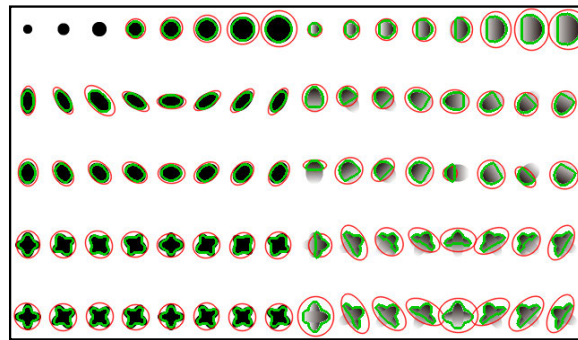
図 4.9: 提案手法による複数のアフィン領域の推定結果.

#### 4.1.5 単純楕円パターンによるテスト

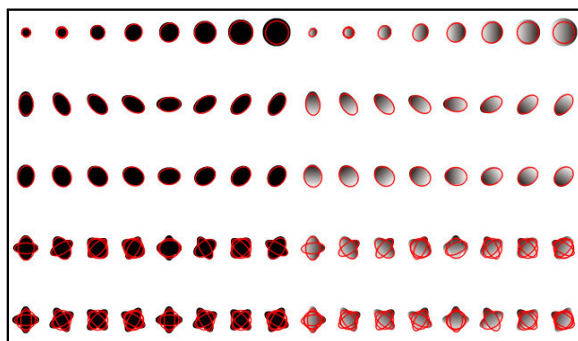
ここでは、単純な楕円パターンを用いてアフィン領域推定の比較を行う。図 4.10 に様々な楕円パターンに対してアフィン領域を推定した結果を示す。楕円パターンの左部分は単一の楕円，交差した楕円，正円を黒で描画した。右部分は左部分と同じ楕円パターンにグレースケールのグラデーション



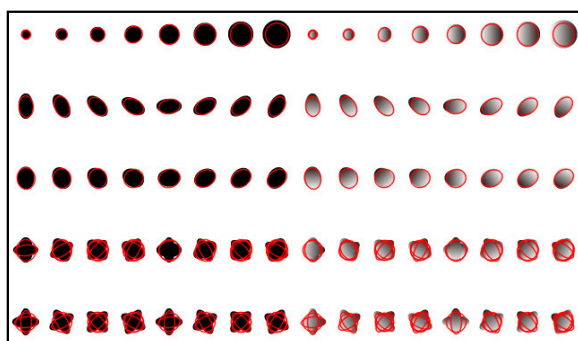
(a) Hessian-Affine



(b) MSER



(c) Original LoG フィルタ



(d) 提案手法

図 4.10: 様々な楕円パターンに対するアフィン領域の推定結果.

表 4.1: 楕円回転角の平均誤差 (degree).

生成したフィルタの回転間隔	10° 間隔	5° 間隔	1° 間隔
Original LoG フィルタ	2.48	1.30	0.53
提案手法	-	0.92	-

処理を施した。このグラデーション処理は照明変化を意味する。各楕円パターンのパラメータの範囲は、長径が [25, 30] ピクセル、短径が [16, 19] ピクセル、回転角が [0°, 60°] である。図 4.10(a) に示す Hessian-Affine によるアフィン領域推定結果は、正円と単一の楕円において正確に推定できているが、交差した楕円パターンでは左部分と右部分で同じアフィン領域を推定できていない。MSER は、輝度値による領域分割に基づいて楕円を当てはめるため、グレースケールパターンではアフィン領域の推定に失敗する (図 4.10(b))。提案手法では、複数のアフィン領域を推定するため、交差した楕円パターンにおいても全て検出することが可能である。また、グレースケールの楕円パターンにおいても単色の楕円パターンと同じアフィン領域を推定することができる (図 4.10(d))。図 4.10(c) は、SVD で分解する前の非等方性 LoG フィルタによるアフィン領域推定結果である。分解前の LoG フィルタのアフィン領域推定結果と提案手法によるアフィン領域推定結果が一致していることから、提案手法は LoG フィルタを十分に近似できていることがわかる。

さらに、連続固有関数によるアフィンパラメータの補間の効果について評価する。定量的な評価をするために、長径 28 ピクセル、短径 16 ピクセルの単純楕円パターンを [0°, 90°] の範囲で 1° 刻みで回転させた画像からアフィン領域を推定する。これらの楕円パターンとアフィン領域推定の回転角の平均誤差を示したのが表 4.1 である。Original LoG フィルタは 1° 間隔で生成してアフィン領域を推定すると誤差は小さくなるが、フィルタの畳み込み回数が増加する。提案手法は分解前の LoG フィルタを 5° 間隔で生成しているが、アフィン領域の回転角の誤差は 1° 以下である。これは、連続固有関数により LoG フィルタ応答値の補間が可能となり、高精度なアフィン領域を推定することができたためと考えられる。

## 4.2 評価実験

提案手法の有効性を確認するために評価実験を行う。実験は異なる視点の 2 画像間で推定したアフィン領域の Repeatability [25] により評価する。評価実験では、提案手法、Hessian-Affine [24], MSER [23], DoG [1] を比較する。

### 4.2.1 データセット

評価実験に用いるデータセットは Oxford matching dataset [50] と IEEE Spectrum magazine dataset を使用する。Oxford matching dataset は 8 シーン { “Graffiti”, “Wall”, “Boat”, “Bark”, “Bikes”, “Trees”, “Leuven”, “UBC” } の画像データセットから構成される。各シーンには見えの変化が生じた 6 枚の

表 4.2: Oxford matching dataset の見えの変化.

見えの変化	シーン
視点変化	Graffiti, Wall
回転・スケール変化	Boat, Bark
ブラー	Bikes, Trees
その他	Leuven, UBC

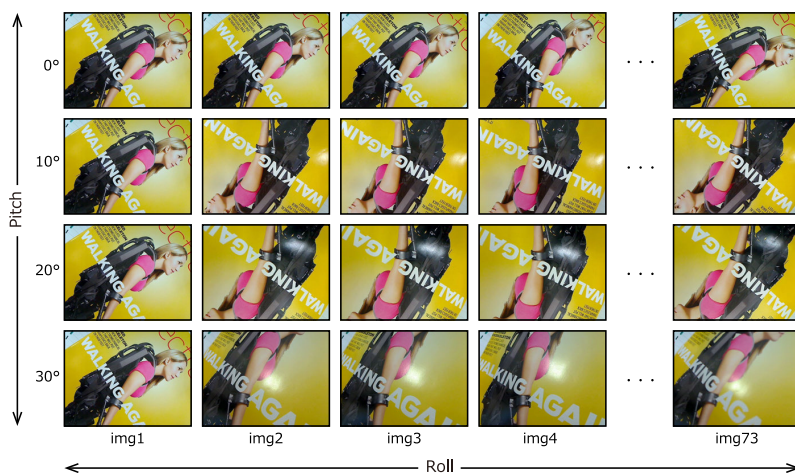


図 4.11: IEEE Spectrum magazine dataset の例.

画像がセットとなっている。各シーンの見えの変化を表 4.2 にまとめる。IEEE Spectrum magazine dataset は 3 種類の雑誌 {“Spectrum 1”, “Spectrum 2”, “Spectrum 3”} の画像データセットである。各種類の雑誌は pitch 角を {0°, 10°, 20°, 30°} 回転させたセットに分かれている。各 pitch 角のセットは roll 角を回転させた 73 枚の画像で構成されている。図 4.11 に Spectrum 1 の画像例を示す。これらのデータセットには各画像間でホモグラフィ行列  $\mathbf{H}$  があらかじめ計算されている。

## 4.2.2 Repeatability による評価方法

Repeatability は異なる視点の 2 画像間から検出されたキーポイント数 ( $\#keypoints1, \#keypoints2$ ) と 2 画像間の対応領域数 ( $\#correspondence\ regions$ ) の割合から算出する。

$$repeatability = \frac{\#correspondence\ regions}{\min(\#keypoints1, \#keypoints2)} \times 100 \quad (4.15)$$

画像間の対応領域を求めるとき、ホモグラフィ行列  $\mathbf{H}$  を用いて 2 画像間で対応するキーポイントのアフィン領域の重なり面積の誤差 (overlap error) を求める。

$$overlap\ error = \left( 1 - \frac{R_A \cap \mathbf{H}_A^\top R'_A \mathbf{H}_A}{R_A \cup \mathbf{H}_A^\top R'_A \mathbf{H}_A} \right) \times 100 \quad (4.16)$$

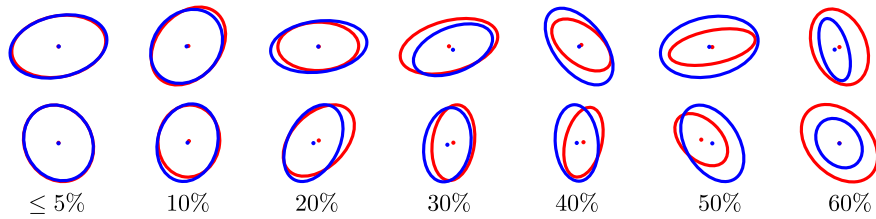


図 4.12: overlap error の例.

ここで,  $R_A, R'_A$  は 2 画像間で検出されたキーポイントのアフィン領域であり,  $\mathbf{H}_A$  は線形化ホモグラフィ行列である. もし, アフィン領域同士の overlap error が閾値  $T_o$  以下であった場合は, 対応領域数 (#correspondence regions) をカウントする. 図 4.12 に 2 つのアフィン領域の overlap error の算出例を示す. まず, repeatability による最初の実験では,  $T_o = 40\%$  に設定し, 提案手法, Hessian-Affine, MSER, DoG を比較する. その後, 閾値を  $T_o = \{40, 30, 20, 10\} \%$  というように変化させた場合の repeatability を提案手法と Hessian-Affine で比較する.

本実験では, 提案手法と Hessian-Affine を対等に評価するために両手法で同じキーポイント (Hessian-Laplace) を使用してアフィン領域を推定する.

### 4.2.3 Repeatability による実験結果

Oxford matching dataset と IEEE Spectrum magazine dataset の repeatability による評価結果を図 4.13, 図 4.14 に示す. これらの実験結果は, overlap error の閾値を  $T_o = 40\%$  としたときの結果である. 提案手法は, 従来法と比較して repeatability が高いことが確認できる. これは, 複数のアフィン領域を推定することで, 画像間のアフィン領域の誤った推定が低減されたためだと考えられる. この結果から, 提案手法は高精度に複数のアフィン領域を推定できていることがわかる. また, Oxford matching dataset の射影変換以外のシーン {“Boat”, “Bark”, “Bikes”, “Trees”, “Leuven”, “UBC”} においても提案手法の repeatability の向上が確認できた. このような結果から, 提案手法は様々な見えの変化に不変なアフィン領域を推定することが可能である.

overlap error の閾値を  $T_o = \{40, 30, 20, 10\} \%$  というように変化させた場合の repeatability による評価結果を図 4.15, 図 4.16 に示す. overlap error の閾値  $T_o$  を 30%, 20%, 10% とした場合においても, 提案手法は Hessian-Affine よりも高い repeatability であることが確認できる. 特に, oxford matching dataset においては, 閾値を下げていくことで Hessian-Affine との repeatability の差が大きくなっていることがわかり, 複数のアフィン領域の効果が有効に働いていると考えられる.

最後に, SIFT 特徴量を用いた提案手法と Hessian-Affine のキーポイントマッチング例を図 4.17 に示す. キーポイントマッチングにおいても, 提案手法は高いマッチング率を得ることができた.

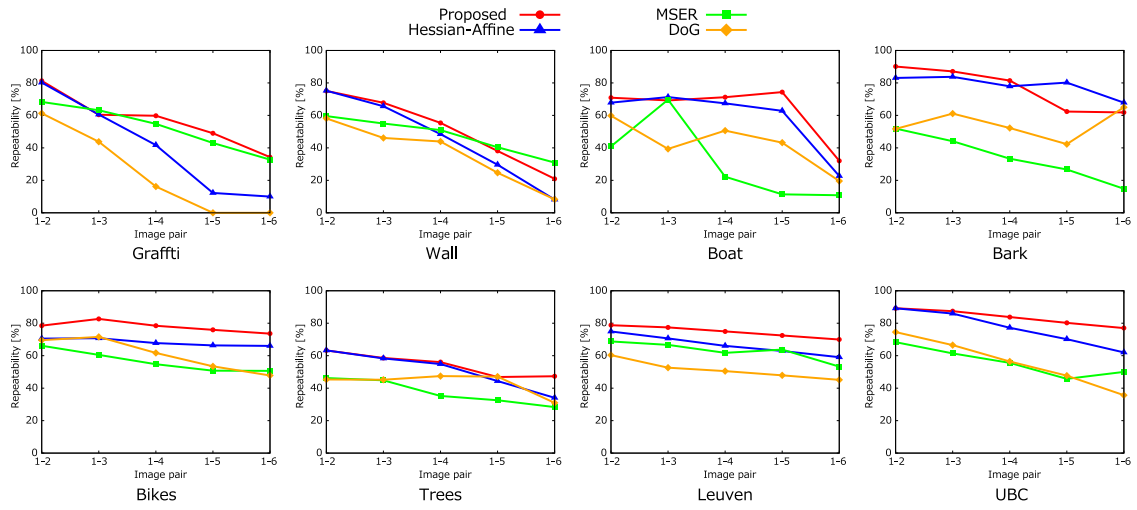


図 4.13: Oxford matching dataset での各手法の repeatability.

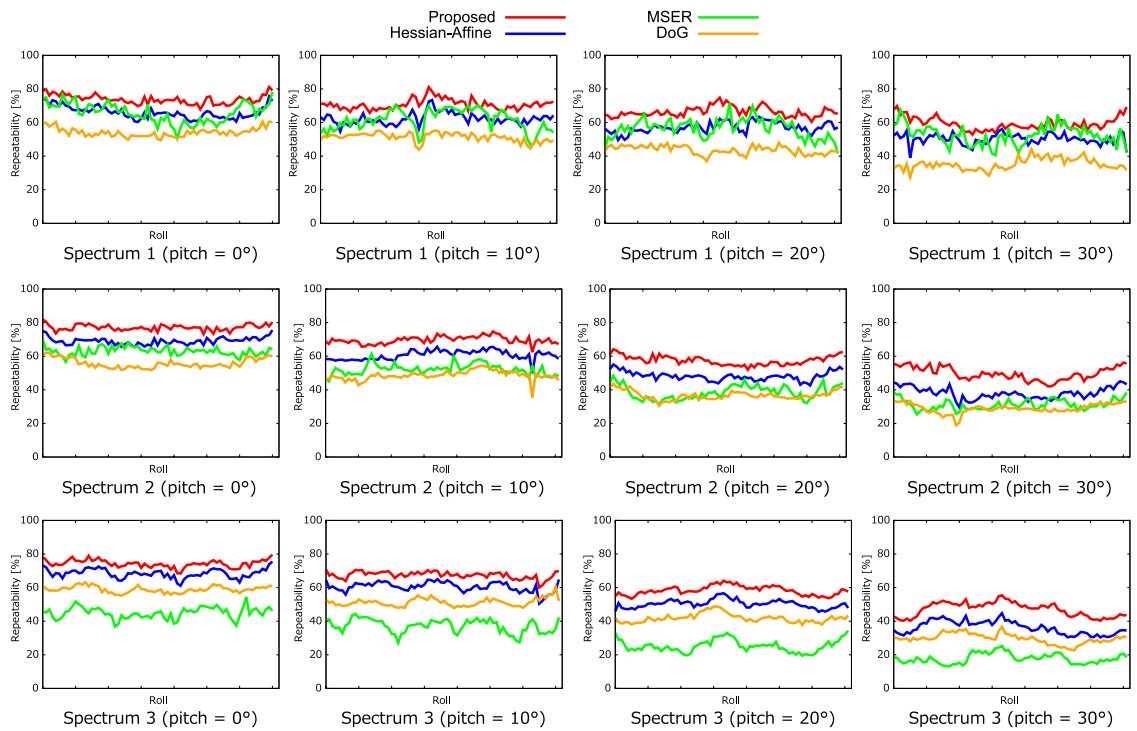


図 4.14: IEEE Spectrum magazine dataset での各手法の repeatability.

## 4.2.4 画像検索タスクにおける認識率

ここでは、3D 物体データセット [64] を用いた画像検索タスクにおける認識率を比較する。実験には、クエリ画像としてターンテーブルにより  $5^\circ$  間隔で回転した 15 種類の物体画像を使用する。そして、回転していない物体画像をデータベース画像としてクエリ画像とマッチングする。図 4.18 に



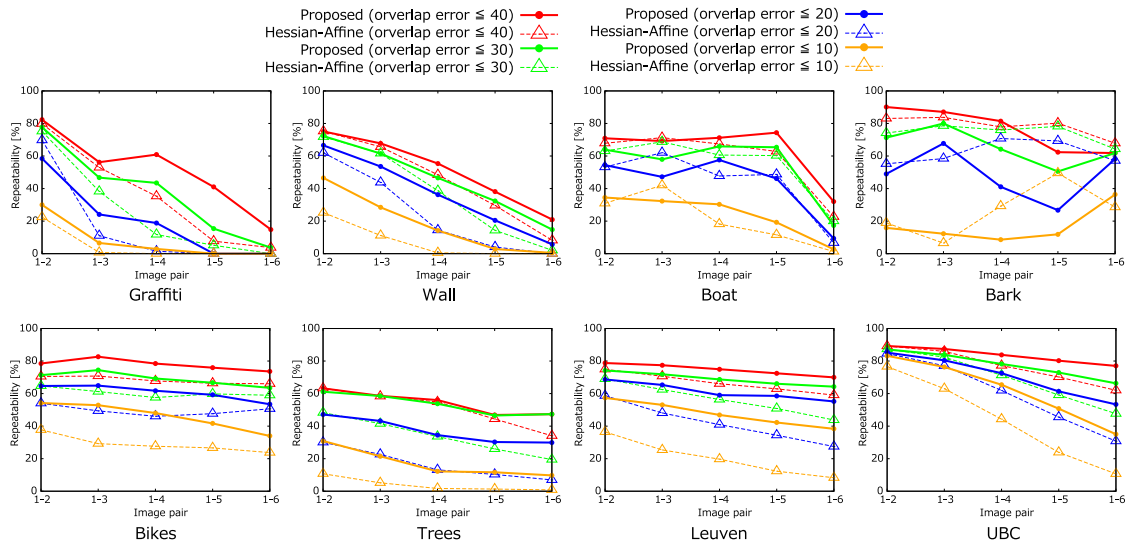


図 4.15: Oxford matching dataset での様々な閾値の repeatability.

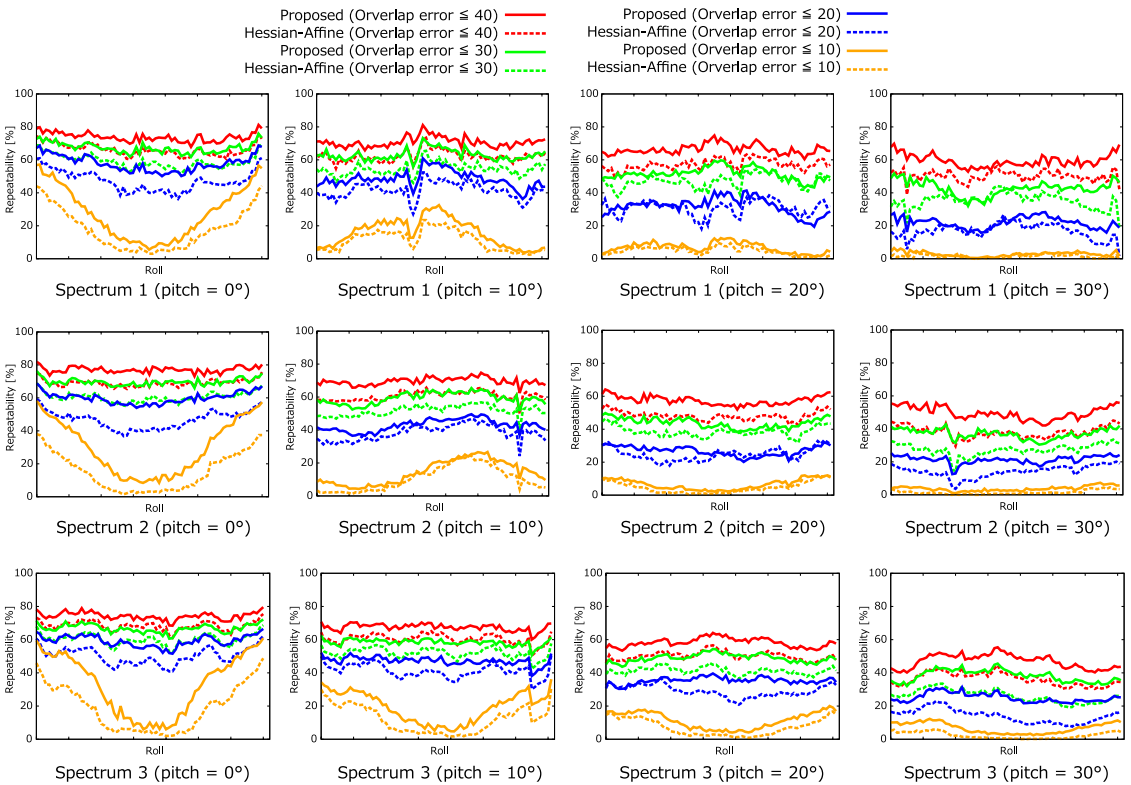


図 4.16: IEEE Spectrum magazine dataset での様々な閾値の repeatability.

評価実験に用いた 3D 物体の画像例を示す。本実験では、クエリ画像を入力した際に、データベース上の全ての検索画像とキーポイントマッチングを行い、最もマッチングスコアの高い画像を検索結果として返す。また、キーポイントマッチングには 128 次元の SIFT 特徴量を使用する。図 4.19 に

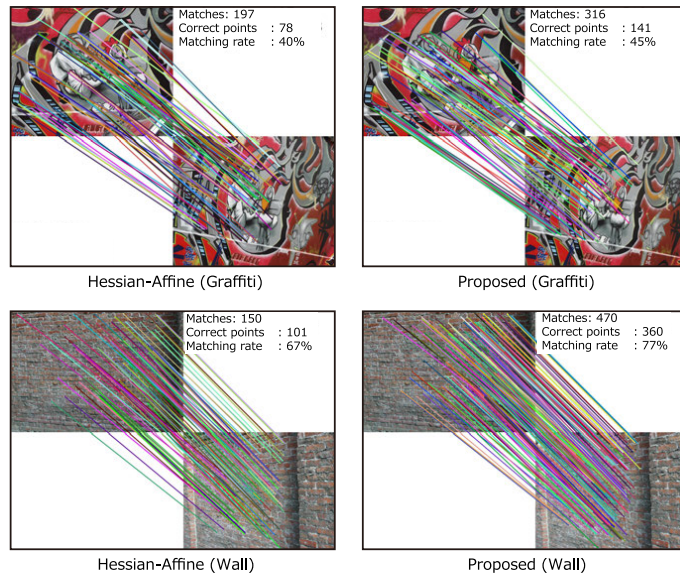


図 4.17: 提案手法と Hessian-Affine によるキーポイントマッチング例.

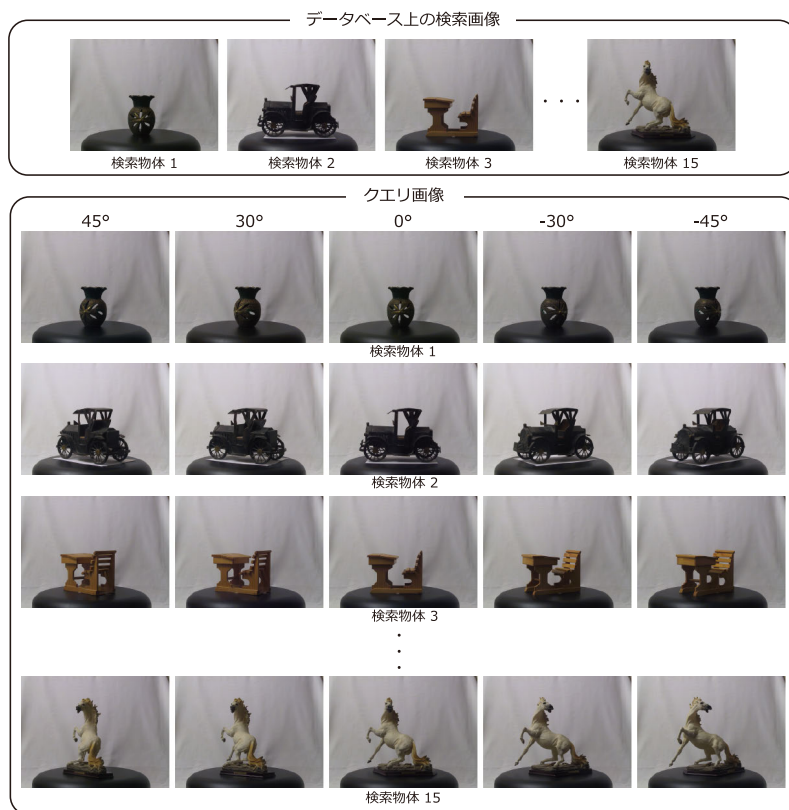


図 4.18: 3D 物体データセットの例.

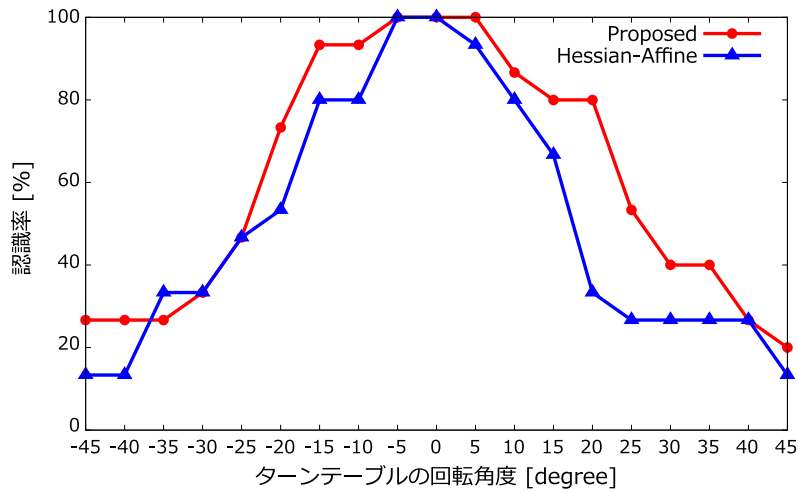


図 4.19: 3D 物体データセットを用いた画像検索の認識率.

表 4.3: 640 × 480 ピクセルの画像における処理時間 [s].

	Hessian-Affine	Original LoG フィルタ	提案手法
処理時間	4.091	198.654	2.277

提案手法と Hessian-Affine の画像検索における認識率を示す。提案手法は、検出されたキーポイントに対して複数のアフィン領域を推定するため、画像検索タスクにおいても高いマッチングスコアが得られ、Hessian-Affine よりも高い認識率であることがわかる。

## 4.2.5 処理時間の比較

提案手法、Original LoG フィルタ、Hessian-Affine のアフィン領域推定における処理時間を比較する。実験に使用する計算機の CPU スペックは Intel Xeon X5470 3.33-GHz である。640 × 480 ピクセルの画像からアフィン領域推定に必要な処理時間の比較を表 4.3 に示す。提案手法は、original LoG フィルタと比較して 87.2 倍高速な処理で複数のアフィン領域を推定することが可能である。これは、4,913 種類の LoG フィルタの畳み込み処理を 14 種類の固有フィルタのみで近似することができるためである。

## 4.2.6 複数のアフィン領域推定の閾値

ここでは、複数のアフィン領域を推定における詳細を述べる。複数のアフィン領域を推定する場合、 $\theta$  軸においてフィルタ応答値の最大極値に近い値を取る他の極値もアフィン領域として推定している (図 4.8)。図 4.20 に複数のアフィン領域の検出処理において、閾値であるフィルタ応答値の最大極値の割合を変化させたときのキーポイントにおける平均アフィン領域数を示す。各キーポイント

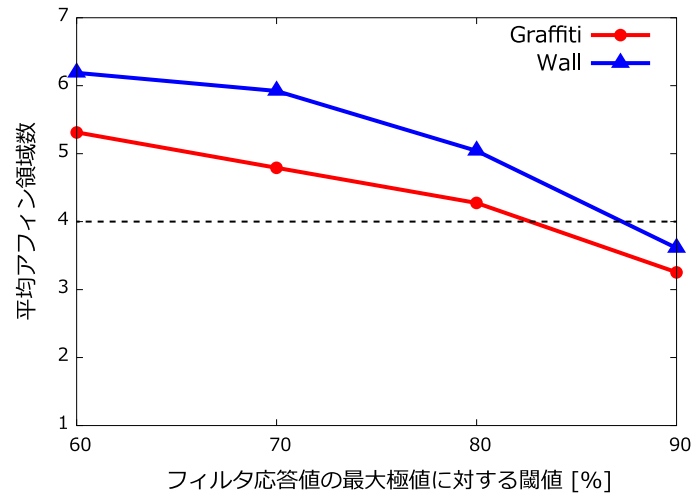


図 4.20: フィルタ応答値の最大極値の割合を変化させた場合のキーポイントの平均アフィン領域数.

トに対してより多くのアフィン領域候補を推定することは、高精度化が期待できるがキーポイントマッチングの高速処理には適していない。キーポイントに対するアフィン領域数は3~4でも十分な精度に達することを実験により確認したため、フィルタ応答値の最大極値の90%以上の応答値を持つ極値をアフィン領域として検出することが望ましい。

### 4.3 まとめ

本章では、非等方性 LoG フィルタを用いた複数のアフィン領域推定方法を提案した。非等方性 LoG フィルタは、SVD を適用することでフィルタの応答値を効率的に算出することが可能となった。また、画像から検出されたキーポイントに対して複数のアフィン領域を推定することで従来法よりも高い repeatability が得られることが確認できた。さらに、提案手法ではフィルタ応答値の計算を連続的な関数として表現することで、任意の連続アフィンパラメータにおける応答値を求めることができ、効率的かつ高精度な複数のアフィン領域を推定することが可能である。

本章では、非等方性 LoG フィルタを低ランク近似することで効率的にアフィン領域を推定する方法について述べたが、この枠組みは後段処理の局所特徴量記述にも応用できると考えている。次の章では、局所特徴量記述子において画像変形に強い多視点特徴量を効率的に記述する方法を示す。

## 第5章

# 因子分解に基づく多視点特徴量と特徴量間距離の下界算出による対応点探索の効率化

本章では、多視点特徴量記述に因子分解法を適用することで、画像間の視点変化に頑健かつ効率的なキーポイントマッチングを実現させる。画像間に強い視点変化を伴う画像のキーポイントをマッチングする場合、Affine SIFT (ASIFT) [38] のように入力画像に様々なアフィン変換を施し、多視点特徴量を記述することが有効である。ASIFTのように、画像  $\mathbf{I}$  に様々なアフィン変換を行ったうえで、特徴量  $\mathbf{d}$  を抽出するモデルは次式のように定義できる。

$$\mathbf{d}(\mathcal{P}) = f(A(\mathbf{I}; \mathcal{P})) \quad (5.1)$$

ここで、 $A(\cdot; \mathcal{P})$  はアフィンパラメータ  $\mathcal{P}$  を用いて画像を変形させる関数であり、 $f(\cdot)$  は与えられた画像から特徴量を記述する関数である。ASIFT の場合は、 $f(\cdot)$  が SIFT 特徴量を記述する関数となる。強い視点変化に対して高精度にキーポイントマッチングを行うには、画像のアフィン変換をオンライン処理で密に行う必要がある。これは、大量のアフィンパラメータ  $\mathcal{P}$  により関数  $A(\cdot; \mathcal{P})$  の実行回数が多くなることを意味しており、計算コストが非常に高くなる。

提案手法では、オンライン処理での画像のアフィン変換を必要としない効率的な多視点特徴量記述を考案する。これは、特徴量記述関数  $f(\cdot)$  を次式に示すように線形演算のみで設計するとシンプルに実現することができる。

$$f(\mathbf{I}) = \mathbf{W}^\top \mathbf{I} \quad (5.2)$$

ここで、 $\mathbf{W} \in \mathbb{R}^{N_m^2 \times N_d}$  は  $N_m \times N_m$  ピクセルの画像  $\mathbf{I} \in \mathbb{R}^{N_m^2}$  から特徴量を抽出する  $N_d$  枚の畳み込みフィルタであり、本研究ではこのフィルタを“特徴量記述フィルタ”と呼ぶ。特徴量記述子の線形モデルでは、パッチ画像  $\mathbf{I}$  の輝度に特徴量記述フィルタを直接畳み込むことで特徴量を計算する。これにより、画像ではなく特徴量記述フィルタにオフライン処理でアフィン変換関数  $A(\mathbf{W}; \mathcal{P})$  を適用することが可能となる。アフィン変換した全ての特徴量記述フィルタとパッチ画像の内積により ASIFT と同様の多視点特徴量  $\mathbf{d}(\mathcal{P})$  を記述することができる。特徴量記述フィルタ  $\mathbf{W}$  はアフィン変換関数  $A(\cdot)$  を適用することにより、大量のフィルタ群が生成されるため、4章で述べたように低ラン

クな基底フィルタで近似する。フィルタ群の近似アルゴリズムとして多くの手法 [20, 49, 65, 66, 67] が提案されているが、提案手法では単純に特異値分解 (SVD) を用いてフィルタ群の低ランク近似を行う。

特徴量記述フィルタを低ランク近似することで、主要な基底フィルタの畳み込みのみで多視点特徴量を効率的に計算することができる。さらに、提案手法では特徴量記述子を連続関数表現することで任意の連続アフィンパラメータにおける多視点特徴量を低コストで記述することができる。

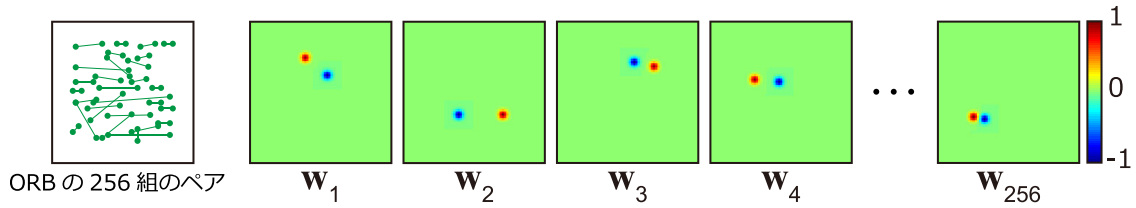


図 5.1: ORB に基づいて設計した特徴量記述フィルタ.

## 5.1 因子分解に基づく多視点特徴量

この節では、線形モデルに基づいた因子分解に基づく多視点特徴量について説明する。

### 5.1.1 線形モデルによる多視点特徴量

提案手法では、式 (5.1) に示すような多視点特徴量を線形モデルにより計算する。線形モデルによる特徴量は式 (5.2) で定義されるように、特徴量記述フィルタ  $\mathbf{W}$  とパッチ画像  $\mathbf{I}$  の単純な内積で特徴量を記述する。これは、ORB [31] や D-BRIEF [53] による特徴量記述方法と類似している。この線形モデルを多視点特徴量記述子へ拡張すると式 (5.3) のように定義できる。線形モデルの場合、アフィン変換関数  $A(\cdot)$  の入力を画像からフィルタへ交換できるため、式 (5.4) で特徴量を定義することができる。

$$\mathbf{d}(\mathcal{P}) = \mathbf{W}^\top A(\mathbf{I}; \mathcal{P}) \quad (5.3)$$

$$= A(\mathbf{W}; \mathcal{P})^\top \mathbf{I} \quad (5.4)$$

特徴量記述フィルタのアフィン変換  $A(\mathbf{W}; \mathcal{P})$  は事前に計算しておくことができるため、線形モデルによる多視点特徴量抽出はオンライン処理によるアフィン変換は不要となる。

特徴量記述フィルタは自由に設計することができるが、提案手法では ORB に基づいてフィルタを設計する。ORB は学習された 256 組のピクセルペアの輝度差により特徴量を記述する。図 5.1 に示すように、ORB の各ピクセルペアの位置に +1 と -1、それ以外に 0 を割り当てた特徴量記述フィルタを生成する。各ピクセルペアにおける特徴量記述フィルタ  $\mathbf{w}_i$  ( $i = \{1, 2, \dots, N_d\}$ ) を列ベクトルとして並べた行列が  $\mathbf{W}$  となる ( $N_d = 256$ )。ノイズの影響を抑えるために、各ピクセルペアの位置はガウス関数により重み付けされている。従来の輝度ベースの特徴量記述は 2 値化関数が適用されるが、提案手法では 2 値化を行わず実数ベクトルを特徴量とする。本研究では、特徴量記述フィルタとして単純に ORB を使用したが、BRIEF や D-BRIEF 等の他の特徴量記述子も提案手法の枠組みに適用することができる。

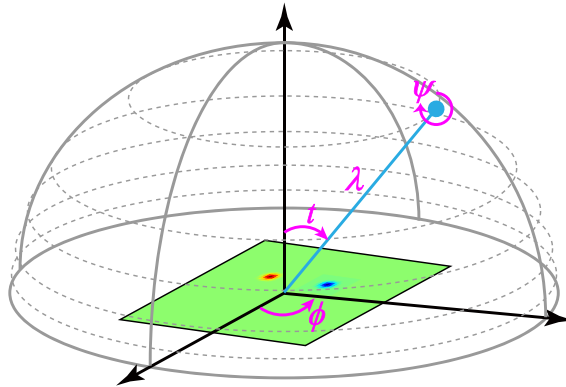


図 5.2: 特徴量記述フィルタの視点合成.

### 5.1.2 特徴量記述フィルタの視点合成

ここでは、特徴量記述フィルタの視点合成に使用するアフィン変換パラメータ  $\mathcal{P}$  について述べる。視点変化を伴う画像間のホモグラフィ行列  $\mathbf{H}$  は、局所領域であることを仮定するとテイラー展開により線形なアフィン行列  $\mathbf{H}_A$  で近似することができる。アフィン行列  $\mathbf{H}_A$  は 2.7.1 項で述べた ASIFT と同じ定義である。アフィン行列  $\mathbf{H}_A$  を用いて、図 5.2 に示すように特徴量記述フィルタ  $\mathbf{w}_i$  をアフィン変換する。アフィン変換パラメータ  $\{\lambda, \psi, t, \phi\}$  は 4 種類存在するが、 $\{\lambda, \psi\}$  はキーポイント検出器により推定されるスケール  $\hat{\sigma}$  とオリエンテーション  $\hat{\theta}$  で置き換えることができる。よって、特徴量記述フィルタのアフィン変換パラメータは  $\mathcal{P} = \{t, \phi\}$  とする。提案手法ではアフィン変換パラメータ  $\{t, \phi\}$  を次式のように定義する。

$$t = \{1.0, 1.2, 1.4, \dots, 4.0\} \quad (5.5)$$

$$\phi = \{0^\circ, 5^\circ, 10^\circ, \dots, 175^\circ\} \quad (5.6)$$

上記のアフィンパラメータにより特徴量記述フィルタをアフィン変換することで 1 枚のフィルタに対して 576 視点のフィルタが生成される。

### 5.1.3 特徴量記述フィルタのコンパクト化

多視点特徴量を記述するためには、アフィン変換された膨大な枚数の特徴量記述フィルタと画像との内積計算が必要となる。アフィン変換された特徴量記述フィルタ  $A(\mathbf{W}; t, \phi)$  の枚数  $N_a$  は 147,456 ( $= 576 \times 256$ ) 枚となる。この  $N_a$  枚のフィルタから構成される行列  $\mathbf{W}_A$  を図 5.3 に示すように SVD を用いてコンパクト化する。

$$\mathbf{W}_A^\top = \mathbf{USV}^\top \quad (5.7)$$



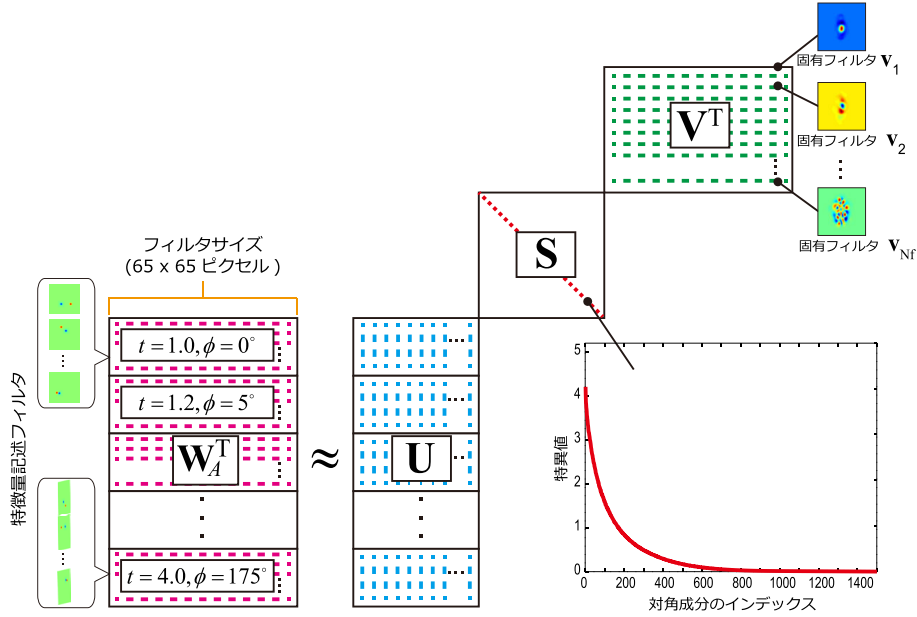


図 5.3: SVD によるアフィン変換した特徴量記述フィルタ群のコンパクト化.

行列  $\mathbf{V} \in \mathbb{R}^{N_m^2 \times N_m^2}$  の列ベクトル  $[\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_{N_m^2}]$  はフィルタとみなすことができるため“固有フィルタ”と呼ぶ。図 5.4 に上位 60 枚の固有フィルタの可視化画像を示す。また、行列  $\mathbf{U}$  と行列  $\mathbf{S}$  の積である  $\mathbf{US} \in \mathbb{R}^{N_a \times N_m^2}$  の列ベクトル  $[\rho_1 \ \rho_2 \ \cdots \ \rho_{N_m^2}]$  は、固有フィルタ  $\mathbf{v}$  の重み係数として作用する“固有関数”である。行列  $\mathbf{S}$  は対角成分に特異値を持ち、図 5.3 に示すように上位の要素のみ大きな値を持ち、下位の要素では 0 に近い値となる。従って、全ての固有フィルタを使用する必要はなく、大きな特異値を持つ上位  $N_f$  枚の固有フィルタを用いてアフィン変換された特徴量記述フィルタを近似することができる ( $N_f \ll N_a$ )。上位  $N_f$  枚の固有フィルタで構成された行列を  $\tilde{\mathbf{V}} \in \mathbb{R}^{N_m^2 \times N_f}$  と表記する。

### 5.1.4 固有関数の連続関数フィッティング

SVD から得られる固有関数  $\rho$  は離散的な値しか持たない。そのため、アフィン変換された特徴量記述フィルタは視点合成で事前に生成したアフィンパラメータ ( $t = \{1.0, 1.2, \dots, 4.0\}, \phi = \{0^\circ, 5^\circ, \dots, 175^\circ\}$ ) でしか再構成することができない。そこで、SVD から得られた離散的な固有関数  $\rho$  を連続関数でフィッティングする。固有関数は  $k$  番目の固有フィルタにおける  $i$  番目の特徴量記述フィルタとして表記すると  $\rho_{i,k}(t, \phi)$  となる。固有関数の連続関数モデルは 4.1.3 項と同様に、以下のような関数モデル  $\varrho_{i,k}(t, \phi)$  を定義する。

$$\varrho_{i,k}(t, \phi) = \sum_{m=0}^{D_M} \sum_{n=0}^{D_N} \alpha_{m,n}^{(i,k)} t^m \cos(n\phi) + \sum_{m=0}^{D_M} \sum_{n=0}^{D_N} \beta_{m,n}^{(i,k)} t^m \sin(n\phi) \quad (5.8)$$

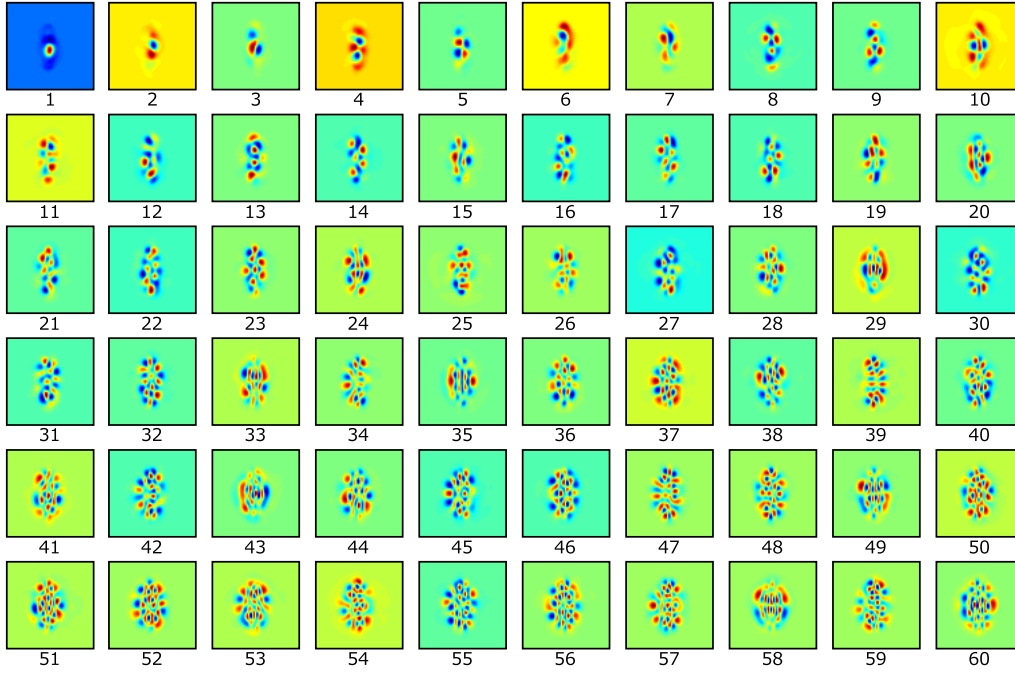


図 5.4: ORB の上位 60 枚の固有フィルタ.

$D_M, D_N$  は連続関数モデルの次数,  $\alpha_{m,n}, \beta_{m,n}$  は未知係数であり, 以下の最小化問題で未知係数を決定する.

$$\arg \min_{\alpha, \beta} \left( \sum_t \sum_{\phi} (\rho_{i,k}(t, \phi) - \varrho_{i,k}(t, \phi))^2 \right) \quad (5.9)$$

$$t = \{1.0, 1.2, 1.4, \dots, 4.0\}$$

$$\phi = \{0^\circ, 5^\circ, 10^\circ, \dots, 175^\circ\}$$

本研究では,  $D_M = 3, D_N = 6$  とすることで元の離散固有関数  $\rho_{i,k}(t, \phi)$  の値を近似できることを確認した. 固有関数を連続関数モデルで表現することで, 任意の連続アフィンパラメータ  $\{t, \phi\}$  によってアフィン変換された特徴量記述フィルタを再構成することが可能である.

### 5.1.5 連続アフィンパラメータによる多視点特徴量の生成

固有関数  $\varrho_{i,k}(t, \phi)$  はアフィンパラメータ  $\{t, \phi\}$  に依存する要素によって構成されるベクトル  $\mathbf{x}(t, \phi)$  と固定係数によって構成される行列  $\mathbf{C}_i$  に分離することができる. よって, 多視点特徴量は図 5.5 のような行列演算で計算することができる. パラメータ  $t, \phi$  における  $i$  次元目の多視点特徴量  $d_i(t, \phi)$

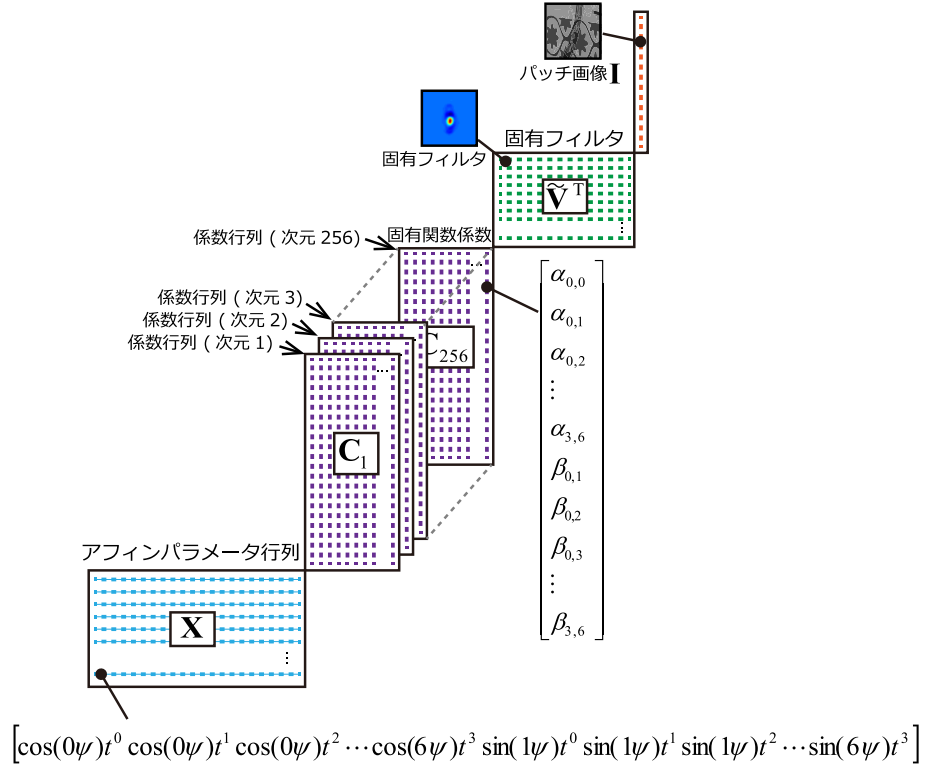


図 5.5: 多視点特徴量の計算.

は以下のように定義できる.

$$A(\mathbf{w}_i; t, \phi) \approx \mathbf{x}(t, \phi)^\top \mathbf{C}_i \tilde{\mathbf{V}}^\top \quad (5.10)$$

$$d_i(t, \phi) \approx \mathbf{x}(t, \phi)^\top \mathbf{C}_i \tilde{\mathbf{V}}^\top \mathbf{I} \quad (5.11)$$

$$\mathbf{x}(t, \phi)^\top = [t^0 \cos(0\phi) \ t^0 \cos(1\phi) \ \cdots \ t^{D_M} \cos(D_N\phi) \\ t^0 \sin(1\phi) \ t^0 \sin(2\phi) \ \cdots \ t^{D_M} \sin(D_N\phi)]$$

$$\mathbf{C}_i = \begin{bmatrix} \alpha_{0,0}^{(i,1)} & \alpha_{0,0}^{(i,2)} & \cdots & \alpha_{0,0}^{(i,N_f)} \\ \alpha_{0,1}^{(i,1)} & \alpha_{0,1}^{(i,2)} & \cdots & \alpha_{0,1}^{(i,N_f)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{D_M, D_N}^{(i,1)} & \alpha_{D_M, D_N}^{(i,2)} & \cdots & \alpha_{D_M, D_N}^{(i,N_f)} \\ \beta_{0,1}^{(i,1)} & \beta_{0,1}^{(i,2)} & \cdots & \beta_{0,1}^{(i,N_f)} \\ \beta_{0,2}^{(i,1)} & \beta_{0,2}^{(i,2)} & \cdots & \beta_{0,2}^{(i,N_f)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{D_M, D_N}^{(i,1)} & \beta_{D_M, D_N}^{(i,2)} & \cdots & \beta_{D_M, D_N}^{(i,N_f)} \end{bmatrix}$$

ここで、行列  $\mathbf{C}_i$  は固有関数  $\mathbf{g}_i(t, \phi)$  の係数  $\alpha, \beta$  で構成される。アフィンパラメータベクトル  $\mathbf{x}(t, \phi)$  に任意の連続アフィンパラメータを与えることで、無数の多視点特徴量を効率的に生成することが可能である。

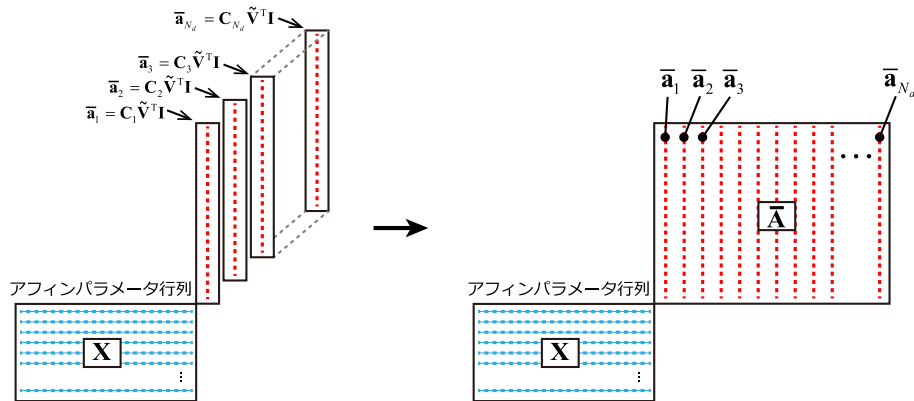


図 5.6: アフィンパラメータ非依存行列  $\bar{\mathbf{A}}$  の生成.

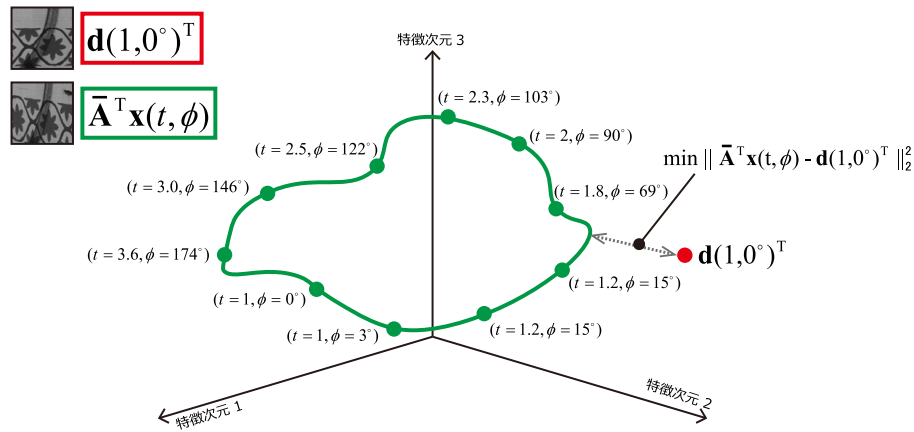


図 5.7: 多視点特微量空間における特微量間の最小距離.

### 5.1.6 特微量間距離の下界計算による対応点探索の効率化

ここでは、因子分解された多視点特微量による効率的な対応点探索について述べる。まず、 $\bar{\mathbf{a}}_i = \mathbf{C}_i \tilde{\mathbf{V}}^T \mathbf{I}$  とすると、 $\bar{\mathbf{a}}_i$  は 1次元のベクトルとなるため、行列  $\bar{\mathbf{A}} = [\bar{\mathbf{a}}_1 \ \bar{\mathbf{a}}_2 \ \dots \ \bar{\mathbf{a}}_{N_d}]$  を生成する (図 5.6)。行列  $\bar{\mathbf{A}}$  を生成することで多視点特微量  $\mathbf{d}(t, \phi)$  は次式で計算できる。

$$\mathbf{d}(t, \phi) \approx \mathbf{x}(t, \phi)^T \bar{\mathbf{A}} \quad (5.12)$$

さらに、様々なアフィンパラメータ  $t, \phi$  における特微量間の最小距離  $Y$  は次式で定義できる (図 5.7)。

$$Y = \min \|\bar{\mathbf{A}}^T \mathbf{x}(t, \phi) - \mathbf{d}(1, 0^\circ)^T\|_2^2 \quad (5.13)$$

ここで、 $\bar{\mathbf{A}}^T \mathbf{x}(t, \phi)$  は視点間の画像ペア  $\mathbf{I}, \mathbf{I}'$  のうち  $\mathbf{I}$  から計算される多視点特微量である。距離計算の問題を簡単にするために、画像  $\mathbf{I}'$  から計算される特微量はアフィンパラメータを  $t = 1, \phi = 0^\circ$  で固

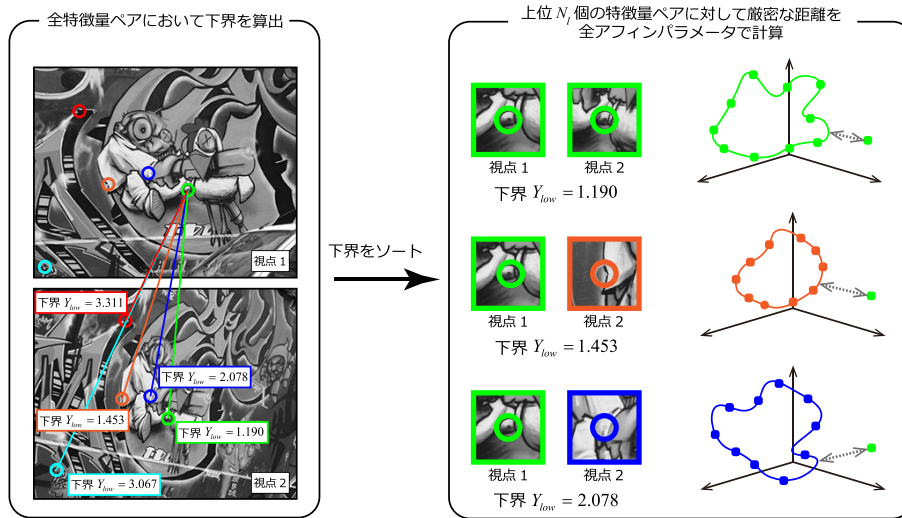


図 5.8: 特徴量ペアの下界に基づく対応点探索例.

定した特徴量とする. 式 (5.13) の行列  $\bar{\mathbf{A}}$  の疑似逆行列を計算することで,  $\hat{\mathbf{x}}(t, \phi) = (\bar{\mathbf{A}})^{-1} \mathbf{d}(1, 0^\circ)^\top$  を計算し,  $\hat{\mathbf{x}}(t, \phi)$  を用いることで次式のように特徴量間の距離の下界  $Y_{low}$  を求めることができる.

$$Y_{low} = \|\bar{\mathbf{A}}^\top \hat{\mathbf{x}}(t, \phi) - \mathbf{d}(1, 0^\circ)^\top\|_2^2 \quad (5.14)$$

下界  $Y_{low}$  は特徴量ペアの全てのアフィンパラメータの距離集合において, どの特徴量ペアの距離値よりも小さな値をとる. そのため, 特徴量ペアで  $Y_{low}$  が大きな値を持つ場合は非対応点といえる. しかし, 下界  $Y_{low}$  は正確な値ではないため, 2 画像間の全特徴量のペアで下界  $Y_{low}$  を算出し, それらをソートした上位  $N_l$  個の特徴量ペアに関して図 5.7 に示すようにアフィンパラメータを変化させて厳密な距離を計算する. 全ての特徴量ペアのアフィンパラメータを総当たりで探索すると多くの処理時間が必要となるため, 距離値の下界に基づいた上位  $N_l$  個のペアに関して正確な最小距離値を探索する. 図 5.8 は視点の異なる画像間でそれぞれ 5 点のキーポイントが検出された場合の下界に基づく対応点探索の例である. 視点 1 の画像から検出された緑のキーポイントに着目した場合, このキーポイントの特徴量と視点 2 の画像から検出された全てのキーポイントの特徴量との下界を求める. 各特徴量ペアにおいて下界に基づいてソートし, 上位  $N_l$  個の特徴量ペアにおいて式 (5.13) を用いて, 多視点特徴量のアフィンパラメータ  $t, \phi$  を変えていきながら正確な特徴量間距離を計算する. 図 5.8 の例では,  $N_l = 3$  とした場合の探索である.  $N_l$  の値が画像間の特徴量の組み合わせ数よりも非常に小さな値であれば, 計算コストを大幅に減らすことができ, 効率的な対応点探索が実現可能である.

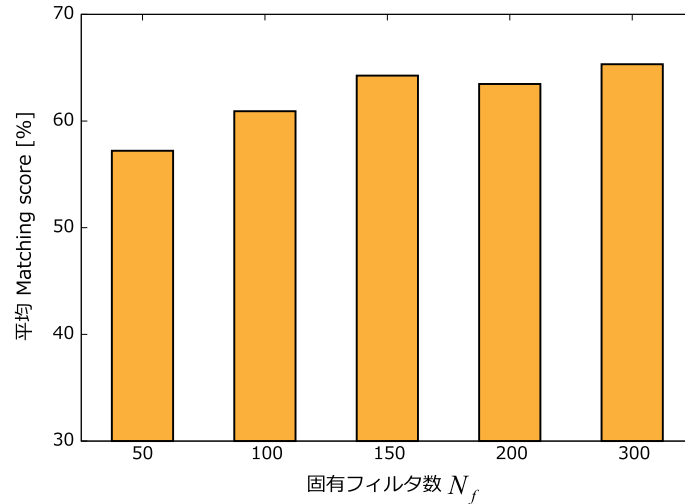


図 5.9: 固有フィルタ数  $N_f$  における平均 matching score.

## 5.2 評価実験

評価実験により，提案手法の有効性を確認する．実験では，次式に示す matching score を評価指標として用いる．

$$\text{matching score} = \frac{\# \text{correct matches}}{\# \text{correct matches} + \# \text{false matches}} \times 100 \quad (5.15)$$

### 5.2.1 データセット

実験では，Oxford matching dataset [50] から 5 シーン {“Graffiti”, “Boat”, “Leuven”, “Bikes”, “UBC”} の画像セットと，RDED dataset [68] から 2 シーン {“Grace”, “Underground”} の画像セットの合計 7 シーンを使用する．各画像データセットは見えの変化を持つ 6 枚の画像で構成されている．

### 5.2.2 固有フィルタ数 $N_f$ における提案手法の性能

提案手法において，固有フィルタ数  $N_f$  を変化させたときのキーポイントマッチングの性能を比較し，性能を維持することができる固有フィルタ数  $N_f$  を決定する．この実験では，{“Graffiti”, “Grace”, “Underground”} の平均 matching score を比較する．使用する 3 シーンのデータセットは画像間に射影変化を伴う画像データセットである．本実験では，下界  $Y_{low}$  を算出せず，全ての特徴量ペアの総当たりにより多視点特徴量との最小距離を探索する．

固有フィルタ数  $N_f$  を変化させたときの平均 matching score を図 5.9 に示す．図 5.9 の結果から， $N_f \geq 150$  で提案手法の性能が維持されていることが確認できる．よって，提案手法の最適な固有フィルタ数  $N_f$  は 150 とする．

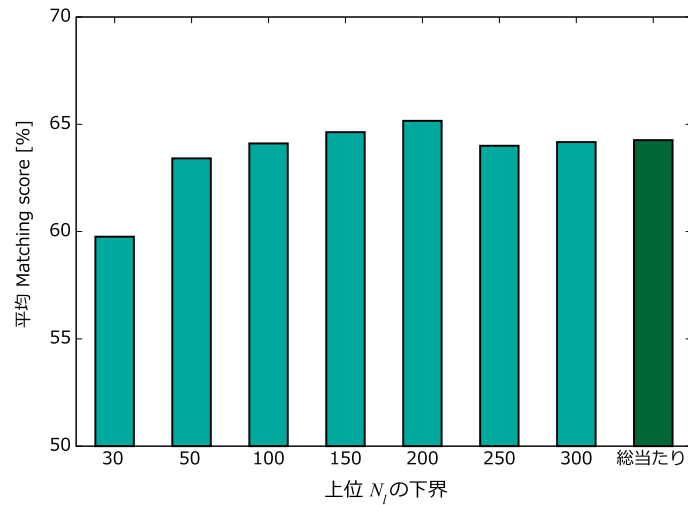


図 5.10: 上位  $N_l$  個の下界を用いた平均 matching score.

### 5.2.3 上位 $N_l$ 個の下界を用いた提案手法の性能

ここでは、5.1.6 項で述べた上位  $N_l$  個の下界を用いた対応点探索の性能を評価する。この実験では、{“Graffiti”, “Grace”, “Underground”} の平均 matching score を比較する。使用する 3 シーンのデータセットは画像間に射影変化を伴う画像データセットである。様々なアフィンパラメータ  $t, \phi$  の多視点特徴量  $\mathbf{d}(t, \phi)$  との最小距離探索を行う  $N_l$  個の特徴量ペアを変化させたときの提案手法の性能を比較する。本実験での固有フィルタ数は  $N_f = 150$  とする。図 5.10 に上位  $N_l$  個の下界を用いたときの平均 matching score を示す。図 5.10 の結果から、 $N_l = 100 \sim 300$  の場合において総当たり探索と同等の性能であることが確認できる。以上の結果より、提案手法では  $N_l = 100$  とする。

### 5.2.4 キーポイントマッチング性能の比較実験

本実験では、提案手法と従来の特徴量記述子との精度を比較する。比較手法は SIFT [1], ORB [31], ASIFT [38], AORB, 提案手法 (brute-force), 提案手法 (top 100) である。AORB は、ASIFT の特徴量記述子を SIFT から ORB に置き換えた手法であり、その他の処理は全て ASIFT と同じである。すなわち、式 (5.1) の関数  $f(\cdot)$  が ORB 特徴量を記述する関数となる。ASIFT と AORB の視点合成のアフィンパラメータは  $t = \{1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}\}$ ,  $\Delta\phi = 72^\circ/t$  とする。提案手法 (brute-force) は下界の算出を行わず、総当たりで対応点を探索する手法であり、提案手法 (top 100) は上位 100 個の下界 ( $N_l = 100$ ) の特徴量ペアのみにおいて、様々なアフィンパラメータの多視点特徴量との最小距離を探索する手法である。全ての手法においてキーポイント検出器は Difference-of-Gaussian (DoG) を使用する。データセットは {“Graffiti” (射影変換), “Boat” (回転 + スケール変化), “Leuven” (照明変化), “Bikes” (ぼかし), “UBC” (JPEG 圧縮), “Grace” (射影変換), “Underground” (射影変換)} の 7 シーンを使用する。

表 5.1: 各手法の平均 matching score [%] と処理時間 [s].

	射影変換	回転 + スケール変化	照明変化	ぼかし	JPEG 圧縮	処理時間
SIFT	63.00	79.93	80.90	83.27	56.13	2.46
ORB	60.46	78.36	79.56	80.01	45.86	2.38
ASIFT	77.90	86.47	85.92	85.66	75.52	187.51
AORB	74.62	83.92	86.26	86.53	72.65	184.90
提案手法 (brute-force)	74.97	85.00	84.64	88.20	68.86	95.24
提案手法 (top 100)	73.90	82.70	82.41	85.98	60.85	43.61

表 5.1 に各データセットの平均 matching score をを示す。射影変換のデータセットにおいて、提案手法は ASIFT よりも matching score が多少劣るが、AORB と同等の精度を達成している。また、その他の見えの変化を伴うデータセットにおいても提案手法は従来法よりも精度が向上していることが確認できる。

## 5.2.5 処理時間

提案手法と従来法のキーポイントマッチングで必要とする処理時間を比較する。表 5.1 に各手法の処理時間を示す。提案手法は ASIFT と比較して約 4.2 倍高速な処理が可能である。また、提案手法の下界を用いた対応点探索を用いることで、総当たり探索よりも約 2.1 倍の処理時間でキーポイントマッチングが可能である。

## 5.2.6 まとめ

本章では、因子分解に基づく多視点特徴量と特徴量間距離の下界算出による対応点探索を提案した。提案手法では、膨大な特徴量記述フィルタを主要な固有フィルタと固有関数で近似することで、効率的な特徴量計算が可能となった。さらに、特徴量間の距離計算において下界を求めることで効率的な対応点探索を実現することができた。

本章で述べた手法は、多視点特徴量を効率的に求めるために線形モデルの特徴量記述子を使用した。これは工夫を加えることで勾配方向ヒストグラム特徴量へ拡張することができる。次の章では、因子分解に基づく多視点特徴量記述を勾配方向ヒストグラムベースの特徴量へ拡張する方法を示す。さらに、多視点特徴量を部分空間表現することで、より高精度な特徴量を記述する。



## 第6章

# 因子分解に基づく多視点特徴量と部分空間表現

本章では、因子分解法を適用した多視点特徴量の部分空間表現を行う。5章で述べた多視点特徴量記述は各アフィンパラメータで計算した特徴ベクトルを独立した特徴として扱っていた。また、下界計算による対応点探索の効率化を行うために、特徴量ペアの一方のアフィンパラメータを固定していた。画像間の強い視点変化に対してより高精度なマッチングを行うには、特徴量ペアの両方を多視点特徴量として記述してマッチングする必要がある。そこで、因子分解法による多視点特徴量をアフィン部分空間へ射影し、部分空間特徴量を記述することで様々なアフィン変換を表現した特徴量ベクトルを生成する。さらに、提案手法ではこれまで線形モデルで扱ってきた多視点特徴量を勾配方向ヒストグラムベースの特徴量へ拡張する。勾配方向ヒストグラムによる多視点特徴量を記述することで、さらなる高精度化が期待できる。提案手法の特徴量記述の流れを図6.1に示す。提案手法では、入力画像に工夫を加えることで、これまで線形モデルしか扱えていなかった因子分解による多視点特徴量を勾配方向ヒストグラムベースの特徴量へと拡張する。さらに、多視点特徴量はPCAを用いて部分空間に投影することで、様々なアフィン変換を表現した特徴ベクトルを求めることができる。

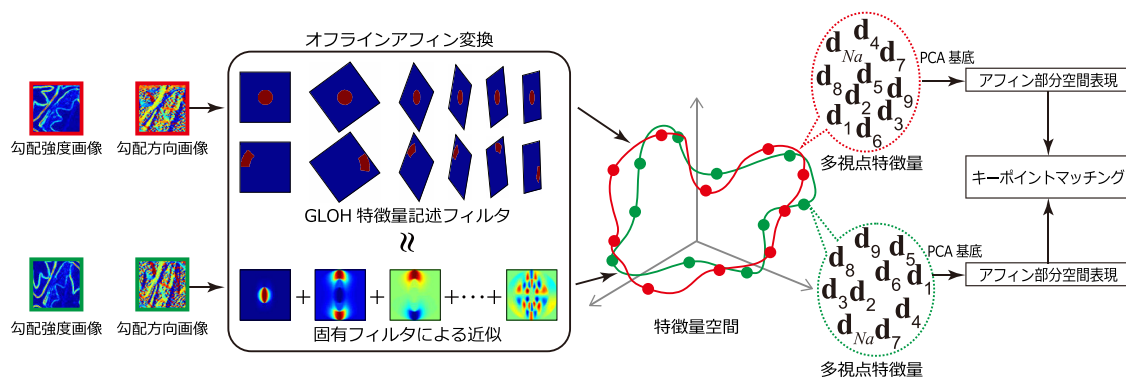


図 6.1: 提案手法による多視点特徴量の部分空間表現。

## 6.1 多視点特徴量の部分空間表現

この節では、因子分解法に基づく多視点特徴量を部分空間特徴量として表現する方法について述べる。5章で述べた因子分解法に基づく多視点特徴量  $\mathbf{d}(t, \phi)$  は各アフィンパラメータ  $t, \phi$  において独立した特徴量として扱っていた。ここでは、様々なアフィンパラメータで得られた多視点特徴量群を部分空間表現する。アフィン部分空間特徴量を記述する ASR [39] では、多視点特徴量を無数に増やすのではなく、PCAにより本質的な要素のみを使用して特徴量を記述している。そこで、提案手法による多視点特徴量も ASR と同様に特徴量の低ランク化を適用する。多視点特徴量の記述方法は 5.1.1 項から 5.1.5 項までに述べた方法と全く同じ方法と同じである。また、ここで示す数式の変数も明記のない限り、5章と同じ変数を用いる。

まず、連続アフィンパラメータにより  $\hat{N}_a$  視点からなる多視点特徴量  $\{\mathbf{d}(t_1, \phi_1), \mathbf{d}(t_2, \phi_2), \dots, \mathbf{d}(t_{\hat{N}_a}, \phi_{\hat{N}_a})\}$  を次式で定義する。

$$\mathbf{D} \equiv \begin{bmatrix} \mathbf{d}(t_1, \phi_1)^\top \\ \mathbf{d}(t_2, \phi_2)^\top \\ \vdots \\ \mathbf{d}(t_{\hat{N}_a}, \phi_{\hat{N}_a})^\top \end{bmatrix} = \mathbf{X}\mathbf{C}\tilde{\mathbf{V}}^\top \mathbf{I} \quad (6.1)$$

$$\mathbf{X} \equiv \begin{bmatrix} \mathbf{x}(t_1, \phi_1)^\top \\ \mathbf{x}(t_2, \phi_2)^\top \\ \vdots \\ \mathbf{x}(t_{\hat{N}_a}, \phi_{\hat{N}_a})^\top \end{bmatrix}$$

$\hat{N}_a$  個の視点からなる多視点特徴量  $\mathbf{D}$  と PCA により算出した射影行列  $\mathbf{P}$  の内積により低ランクな多視点特徴量  $\mathbf{D}_{low}$  を計算する。

$$\mathbf{D}_{low} = \mathbf{P}^\top \mathbf{D} \quad (6.2)$$

$\mathbf{P} \in \mathbb{R}^{N_a \times N_p}$  は大量の学習画像から算出した特徴量データセットに PCA を適用し、その固有ベクトルを並べて生成する。 $N_p$  は PCA 圧縮の基底数である。学習用データには multi-view stereo dataset [69] の “liberty” と “notredame” からランダムで選択した 100k 枚のパッチ画像を使用した。

その後、低ランク多視点特徴量  $\mathbf{D}_{low} \in \mathbb{R}^{N_p \times \hat{N}_a}$  を線形部分空間で表現する。

$$\mathbf{D}_{low} \approx \begin{bmatrix} \hat{\mathbf{d}}_1 & \hat{\mathbf{d}}_2 & \cdots & \hat{\mathbf{d}}_{N_s} \end{bmatrix} \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,\hat{N}_a} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,\hat{N}_a} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N_s,1} & b_{N_s,2} & \cdots & b_{N_s,\hat{N}_a} \end{bmatrix} \quad (6.3)$$

ここで、 $\hat{\mathbf{D}} = [\hat{\mathbf{d}}_1 \ \hat{\mathbf{d}}_2 \ \cdots \ \hat{\mathbf{d}}_{N_s}]$  は部分空間における基底ベクトルであり、 $b$  は部分空間座標となる。 $N_s$  は  $\mathbf{D}_{low}$  の部分空間における基底数である。基底ベクトル  $\hat{\mathbf{d}}$  は  $\mathbf{D}_{low}$  に PCA を適用することで得

られる。最後に、部分空間で表現された特徴ベクトル  $\mathbf{d}_{sub}$  を計算する。

$$\mathbf{d}_{sub} = \begin{bmatrix} \frac{e_{1,1}}{\sqrt{2}} & e_{1,2} & e_{1,3} & \cdots & e_{1,N_p} & \frac{e_{2,2}}{\sqrt{2}} & e_{2,3} & \cdots & \frac{e_{N_p,N_p}}{\sqrt{2}} \end{bmatrix} \quad (6.4)$$

$$\mathbf{E} = \hat{\mathbf{D}}\hat{\mathbf{D}}^\top = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,N_p} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,N_p} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N_p,1} & e_{N_p,2} & \cdots & e_{N_p,N_p} \end{bmatrix}$$

特徴量  $\mathbf{d}_{sub}$  は様々なアフィンパラメータ  $\{t, \phi\}$  で生成した多視点特徴量を表現する部分空間特徴量である。対応点探索において、アフィン部分空間特徴量  $\mathbf{d}_{sub}$  は単純なユークリッド距離で2画像間の特徴量を比較することができる。

## 6.2 勾配方向ヒストグラムモデルへの拡張

ここまで述べた因子分解法による多視点特徴量は、式 (5.1) における関数  $f(\cdot)$  を線形モデルとして扱ってきた。より高性能な特徴量を記述するために、SIFT や GLOH のような勾配方向ヒストグラムモデルの特徴量記述子が有効である。しかし、一般的な勾配方向ヒストグラムモデルは関数  $f(\cdot)$  に非線形演算が含まれるため、提案手法の枠組みへの拡張が困難である。これは、関数  $f(\cdot)$  へ入力する画像  $\mathbf{I}$  を勾配強度画像  $\mathbf{m}$  と勾配方向画像  $\mathbf{o}$ 、量子化行列  $\Omega$  に置き換えることで、勾配方向ヒストグラムモデルの特徴量も計算可能である。

まず、パッチ画像  $\mathbf{I}$  から  $x$  方向および  $y$  方向の勾配画像  $\mathbf{g}_x, \mathbf{g}_y \in \mathbb{R}^{N_m^2}$  を計算することにより、勾配強度画像  $\mathbf{m}$  と勾配方向画像  $\mathbf{o}$  を計算する。

$$\mathbf{m} = \sqrt{\mathbf{g}_x \circ \mathbf{g}_x + \mathbf{g}_y \circ \mathbf{g}_y} \quad (6.5)$$

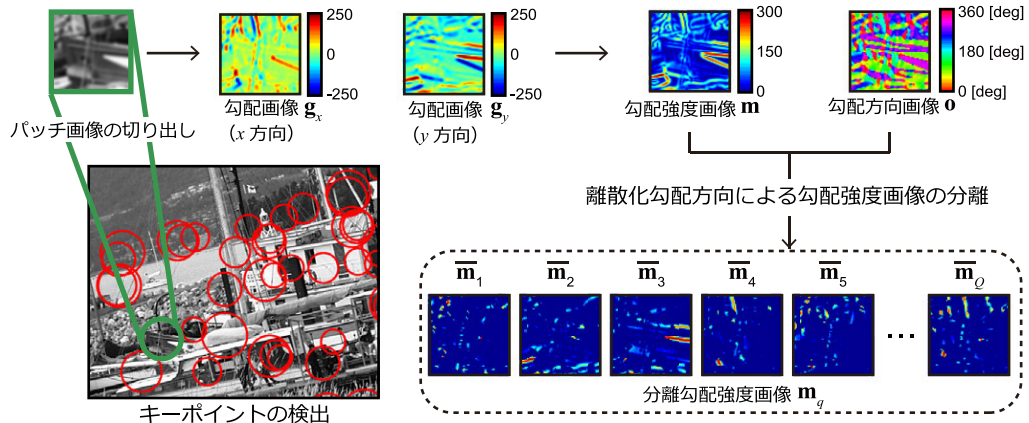
$$\mathbf{o} = \tan^{-1} \left( \frac{\mathbf{g}_y}{\mathbf{g}_x} \right) \quad (6.6)$$

ここで、 $\circ$  演算子はベクトルまたは行列の要素ごとの積を表す。勾配強度画像  $\mathbf{m}$  は、図 6.2(a) に示すように勾配方向ヒストグラムを生成するために勾配方向画像  $\mathbf{o}$  を用いて離散勾配方向  $q = \{1, 2, \dots, Q\}$  に分割させる。よって、各離散勾配方向  $q$  ごとに、分離勾配強度画像  $\bar{\mathbf{m}}_q$  が生成される。

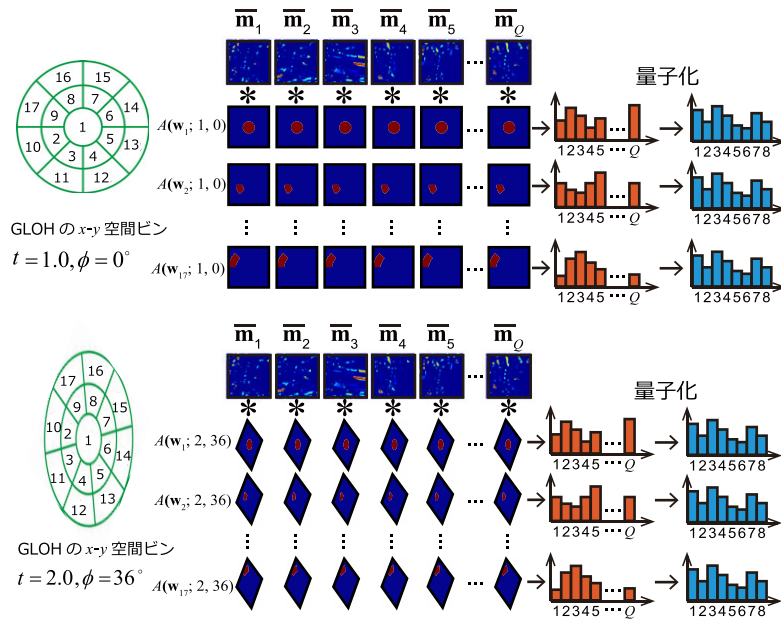
$$\bar{\mathbf{o}} = \text{floor}(\mathbf{o}/\Delta\theta) \quad (6.7)$$

$$\bar{\mathbf{m}}_q = \mathbf{m} \circ \delta[q, \bar{\mathbf{o}}] \quad (6.8)$$

ここで、 $\Delta\theta$  は離散化する勾配方向の間隔 (e.g.,  $\Delta\theta = 15^\circ$ )、 $\text{floor}(\cdot)$  は小数点以下を切り捨てる関数である。 $\delta[\cdot]$  は Kronecker デルタ関数であり、離散勾配方向  $q$  に対応する  $\bar{\mathbf{o}}$  の要素に 1、それ以外の要素は 0 として返す。図 6.2(b) に示すように、分離勾配強度画像  $\bar{\mathbf{m}}_q$  と特徴量記述フィルタ  $\mathbf{w}_i$  の内積を計算することにより、勾配方向ヒストグラムを計算する。特徴量記述フィルタ  $\mathbf{w}_i$  ( $i = \{1, 2, \dots, N_d\}$ )



(a) 分離勾配強度画像の生成



(b) 勾配方向ヒストグラム特徴量の記述

図 6.2: 勾配方向ヒストグラムモデルの多視点特徴量記述.

は、SIFT や GLOH などの  $x$ - $y$  空間上のビンの形状によって設計する。提案手法では、キーポイントの位置ずれなどによるノイズを低減できる GLOH の  $x$ - $y$  空間上のビン形状で特徴量記述フィルタを生成する ( $N_d = 17$ )。GLOH の各  $x$ - $y$  空間上のビンに対応した領域を 1、それ以外に 0 を割り当てることで特徴量記述フィルタを生成する (図 6.3)。勾配方向特徴量は 8 方向に量子化されるため、

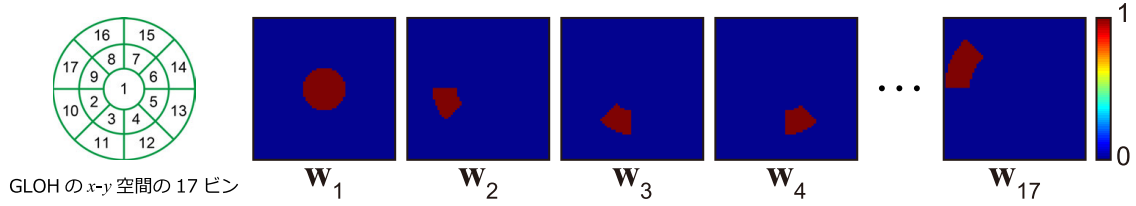


図 6.3: GLOH に基づいて設計した特徴量記述フィルタ.

GLOH 特徴量  $\mathbf{d}$  は次式のように計算する.

$$\mathbf{d} = (\mathbf{W}^T \mathbf{M}) \Omega \quad (6.9)$$

$$\Omega = \begin{bmatrix} \omega & \emptyset & \cdots & \emptyset \\ \emptyset & \omega & \cdots & \emptyset \\ \vdots & \vdots & \ddots & \vdots \\ \emptyset & \emptyset & \cdots & \omega \end{bmatrix}$$

$$\omega \in \{1\}^{\frac{Q}{8} \times 1}, \emptyset \in \{0\}^{\frac{Q}{8} \times 1}$$

ここで,  $\mathbf{W} \in \{0, 1\}^{N_m^2 \times N_d}$  の各列ベクトルは特徴量記述フィルタ  $\mathbf{w}_i$  で構成されており,  $\mathbf{M} \in \mathbb{R}^{N_m^2 \times Q}$  の各列ベクトルは分離勾配強度画像  $\mathbf{m}_q$  で構成されている. 行列  $\mathbf{W}$  と行列  $\mathbf{M}$  の内積により勾配方向ヒストグラムを計算した後, SIFT と同様に勾配方向を 8 方向に量子化する. 勾配方向を量子化するために, 行列  $\mathbf{W}^T \mathbf{M} \in \mathbb{R}^{N_d \times Q}$  と量子化行列  $\Omega \in \{0, 1\}^{Q \times 8}$  の内積を計算する. 量子化行列  $\Omega$  との内積を計算することにより, 行列  $\mathbf{W}^T \mathbf{M}$  の水平方向の近傍が足し合わされ, 8 方向に量子化された勾配方向ヒストグラムを生成することができる. 勾配方向ヒストグラムモデルにおける多視点特徴量  $\mathbf{d}(t, \phi)$  もフィルタ  $\mathbf{W}$  をアフィン変換することで計算することができる.

$$\mathbf{d}(t, \phi) = A(\mathbf{W}; t, \phi)^T \mathbf{M} \Omega \quad (6.10)$$

あとは, 5.1.1 項から 6.1 節で説明した方法を用いることで, フィルタのコンパクト化や多視点特徴量の部分空間表現を行うことができる. GLOH の特徴量記述フィルタを SVD によりコンパクト化した際の上位 60 枚の固有フィルタを図 6.4 に示す.

## 6.3 評価実験

評価実験では, 提案手法のキーポイントマッチングの性能評価, Hpatches benchmark [61] による様々な評価タスクでの比較を行う. 本実験では, 図 5.1 に示す特徴量記述フィルタを用いた多視点特徴量を部分空間表現した特徴量記述子を “ORB-like” と表記する. また, 図 6.3 に示す特徴量記述フィルタを用いた多視点特徴量を部分空間表現した特徴量記述子を “GLOH-like” と表記する.

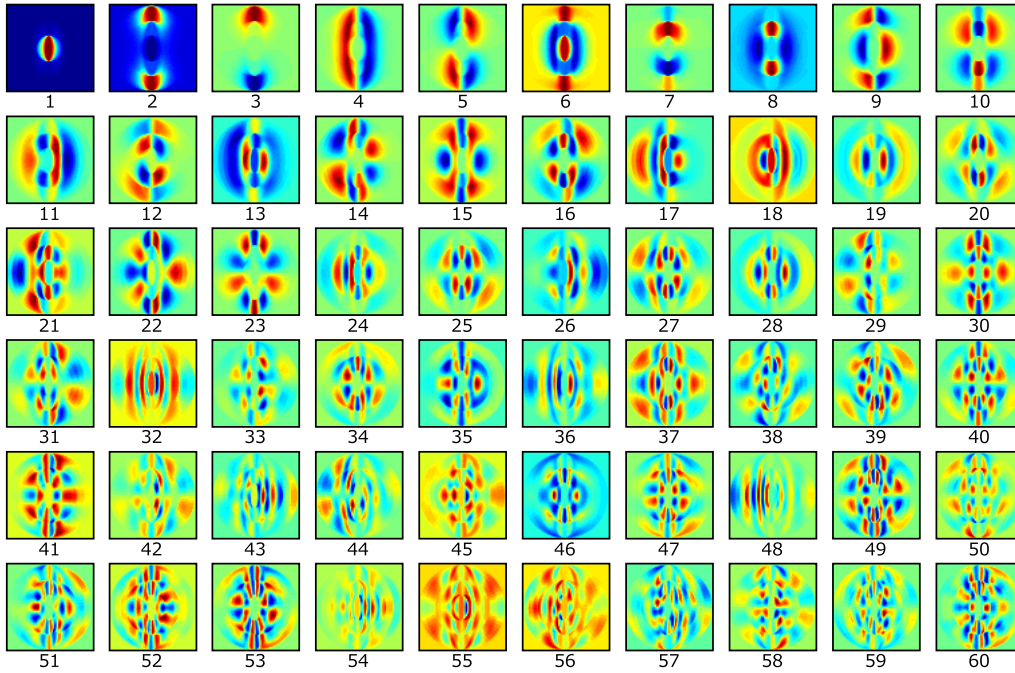


図 6.4: GLOH の上位 60 枚の固有フィルタ.

### 6.3.1 PCA の基底数 $N_p, N_s$ における提案手法の性能

提案手法において、多視点特徴量に適用する PCA の基底数  $N_p, N_s$  を変化させたときのキーポイントマッチングの性能を比較する。  $N_p$  は多視点特徴量を次元圧縮する際の PCA の基底ベクトル数であり、  $N_s$  は低ランク化した多視点特徴量をアフィン部分空間で表現する際の PCA の基底ベクトル数である。本実験では、{“Graffiti”, “Wall”, “Posters”, “Underground”} の 4 シーン (射影変換画像のみ) において PCA の基底数  $N_p, N_s$  を変えたときのキーポイントマッチングの効果を検証する。固有フィルタ数  $N_f$  は、元の特徴量記述フィルタを高精度に近似できるように  $N_f = 1,000$  とする。キーポイントマッチングの性能評価指標には式 (5.15) に示す matching score を使用する。図 6.5 に 4 シーンのデータセットの平均 matching score を示す。提案手法 (ORB-like) では、  $N_p = 106, N_s = 40$  のときに最も高い matching score であるため、特徴量記述にはこれらのパラメータを使用する。また、提案手法 (GLOH-like) では、  $N_p = 42, N_s = 14$  で高い matching score が得られているため、特徴量記述の際にはこれらのパラメータを使用する。

### 6.3.2 固有フィルタ数 $N_f$ における提案手法の性能

提案手法において、多視点特徴量記述に使用する固有フィルタ数  $N_f$  を決定する。より多くの固有フィルタを使用することで、元の特徴量記述フィルタを正確に近似することができる。しかし、大量の固有フィルタとパッチ画像を畳み込むには高い計算コストを必要とする。従って、固有フィルタ数  $N_f$  を変化させながらキーポイントマッチングの性能を評価する。本実験では、{“Graffiti”, “Wall”,

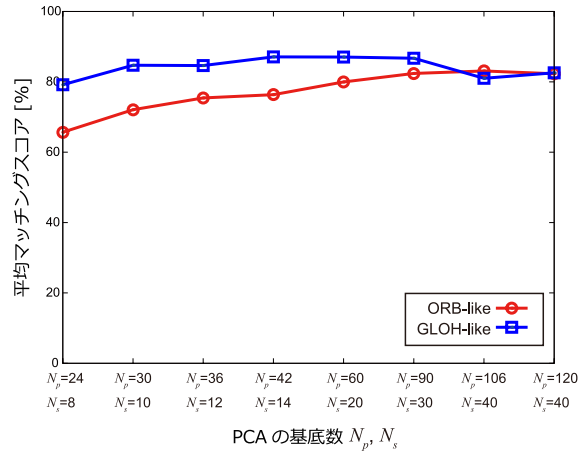


図 6.5: PCA の基底数  $N_p, N_s$  を変化させたときの平均 matching score.

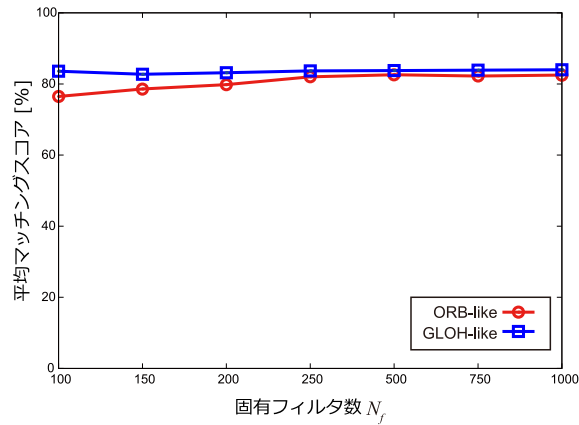


図 6.6: 固有フィルタ数  $N_f$  を変化させたときの平均 matching score.

“Posters”, “Underground” } の 4 シーン (射影変換画像のみ) において固有フィルタ数  $N_f$  を変えたときのキーポイントマッチングの効果を検証する。PCA の基底数  $N_p, N_s$  は、6.3.1 項で決定した値を使用する。キーポイントマッチングの性能評価指標には式 (5.15) に示す matching score を使用する。図 6.6 に 4 シーンのデータセットの平均 matching score を示す。提案手法 (ORB-like) では、 $N_f = 250$  以上で matching score が維持できているため、固有フィルタ数は 250 とする。また、提案手法 (GLOH-like) では、 $N_f = 100$  以上で matching score が維持できているため、固有フィルタ数は 100 とする。

### 6.3.3 従来の多視点特徴量記述子との比較

ここでは、2 画像間のキーポイントマッチングによる従来法との比較実験を行う。比較する手法として提案手法 (ORB-like), 提案手法 (GLOH-like), ASIFT [38], ASR-naive [39], ASR-fast [39] を使用する。全ての手法においてキーポイント検出とオリエンテーション推定方法は SIFT [1] を使

表 6.1: 提案手法のパラメータ設定.

パラメータ	ORB-like	GLOH-like
$t, \phi$	$t = \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\},$ $\phi \in [0^\circ, 180^\circ),$ $\Delta\phi = 5^\circ/t$	$t = \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\},$ $\phi \in [0^\circ, 180^\circ),$ $\Delta\phi = 5^\circ/t$
$\Delta\theta$	–	$15^\circ$
$N_m$	65	65
$N_d$	256	17
$N_f$	250	100
$N_p$	106	42
$N_s$	40	14

用する. 評価用データセットは, Oxford matching dataset [50] と RDED dataset [68] から {"Bark", "Graffiti", "Leuven", "Wall", "Posters", "Underground"} の 6 シーンを使用する. 提案手法のパラメータ設定を表 6.1 に示す. また, ASIFT, ASR-naive, ASR-fast の視点合成のアフィンパラメータは  $t = \{1, \sqrt{2}, 2, 2\sqrt{2}, 4\}, \phi \in [0^\circ, 180^\circ), \Delta\phi = 72^\circ/t$  に設定する.  $\Delta\phi$  は回転パラメータ  $\phi$  のサンプリング間隔である. 図 6.7 に各手法の recall-precision カーブを示す. 各図の凡例には Area under curve (AUC) を示している.

実験結果から, 提案手法は多くのシーン画像において ASIFT より高い精度が得られていることが確認できる. {"Wall 1-5", "Underground 1-5", "Underground 1-6", "Bark 1-5"} の提案手法の AUC は ASIFT よりも 8%以上向上していることがわかる. また, 提案手法は ASR-naive と同等の精度が得られ, {"Underground 1-5", "Underground 1-6"} のシーンにおける提案手法 (GLOH-like) の AUC は ASR-naive よりそれぞれ 4.2%と 8.8%高い結果が得られた. ここで重要な点は, ASIFT と ASR-naive はアフィン変換をオンライン処理で行う手法であり, 提案手法はオンラインのアフィン変換を行うことなく従来法と同等以上の精度を達成することができる.

また, オフラインのアフィン変換で多視点特徴量を記述することができる ASR-fast と提案手法 (GLOH-like) の AUC を比較すると, 多くのシーン画像で提案手法 (GLOH-like) は高い精度が得られた. 特に, {"Underground 1-6", "Wall 1-5"} のシーンにおいては, 提案手法 (GLOH-like) は ASR-fast よりも 9.5%以上高い精度が得られ, "Graffiti 1-4"では約 20%の精度向上が確認できた. 提案手法は, 射影変化を伴う様々なシーンにおいて ASR-naive と同等の精度が得られているため, 射影変化に対して有効であると言える. また, 提案手法 (GLOH-like) は多くのシーン画像で提案手法 (ORB-like) よりも高い精度が得られた. これは, 単純な輝度差で特徴量を記述するのではなく, パッチ画像内の勾配方向ヒストグラムに基づいて特徴量を記述しているため, 提案手法 (GLOH-like) の精度が高かったと考えられる.



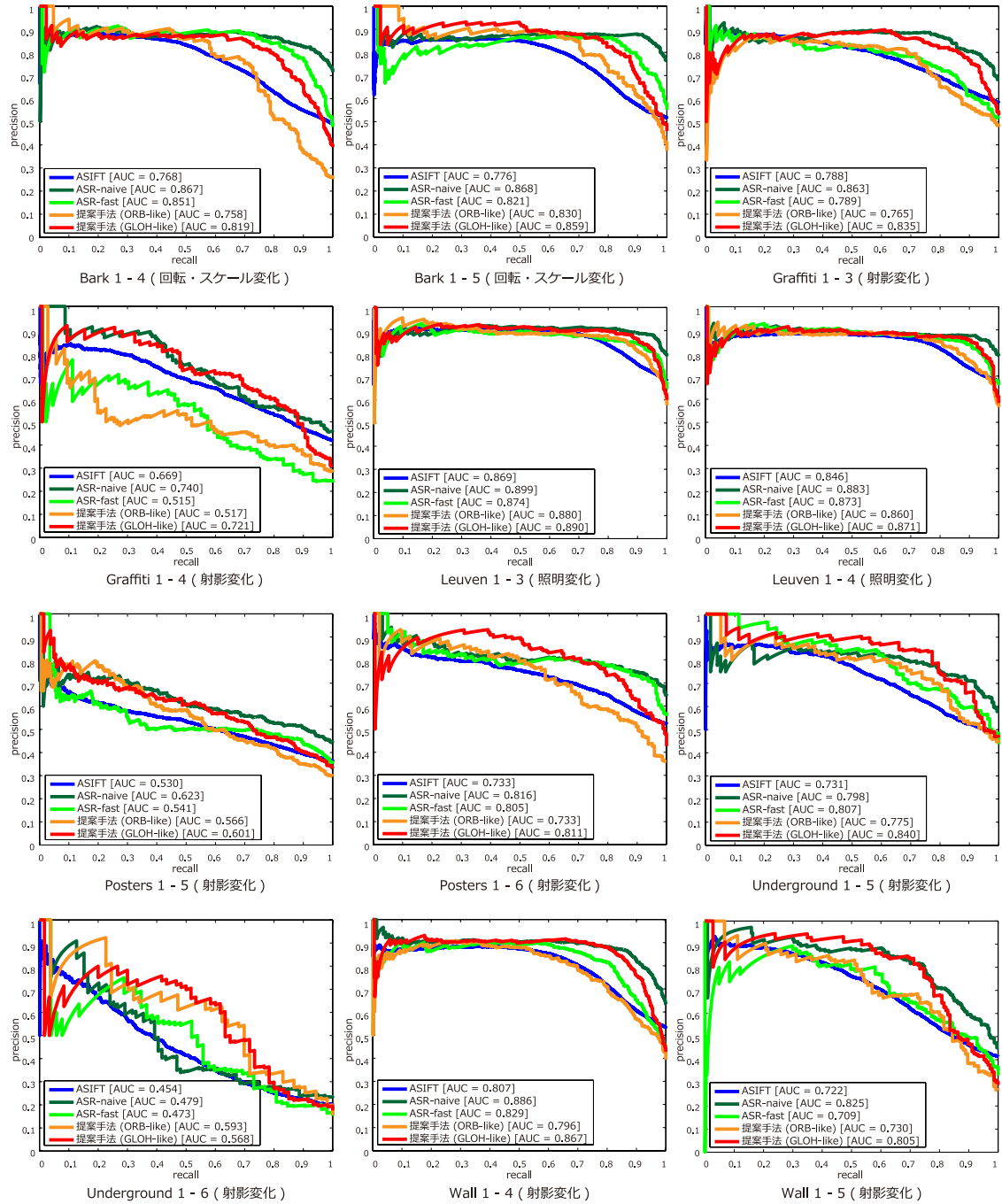


図 6.7: 異なる視点変化におけるキーポイントマッチングの精度。

### 6.3.4 HPatches benchmark での評価

ここでは、局所特徴量記述子の評価ベンチマークである HPatches [61] で公開されている “Patch verification”, “Image matching”, “Patch retrieval” の 3 つの評価タスクで実験する。Patch verification では与えられたパッチ画像ペアが positive ペアであるか negative ペアであるかを局所特徴量で分類し、

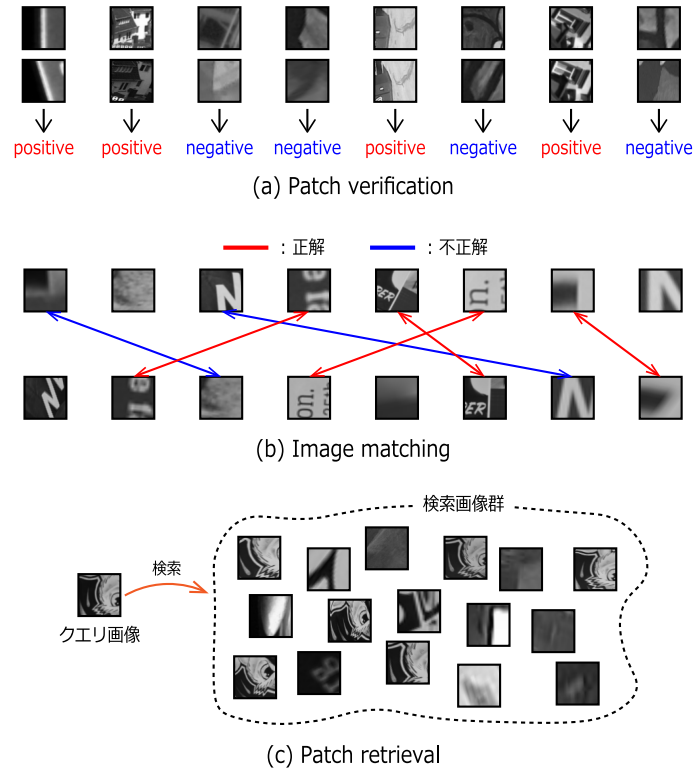
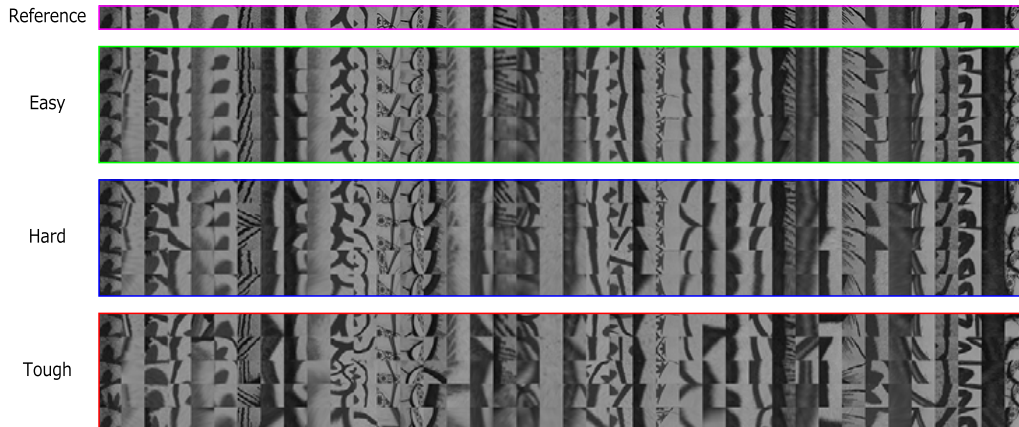


図 6.8: HPatches の評価タスク。

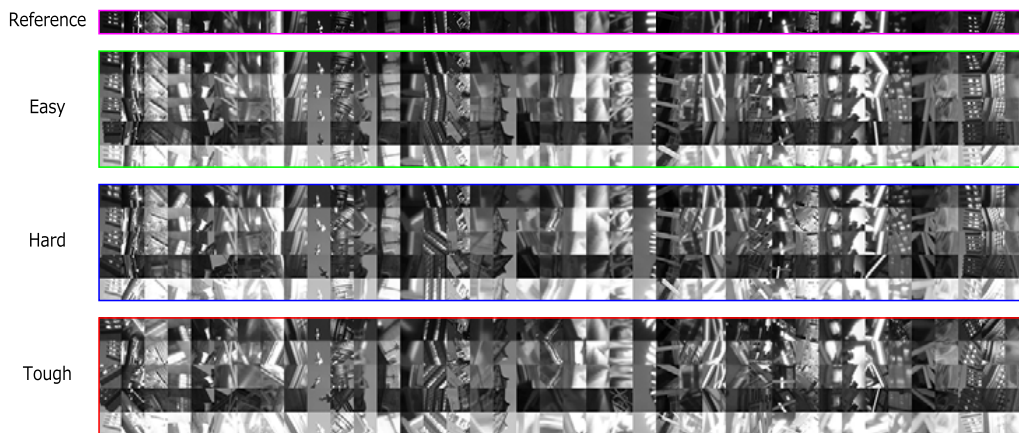
この2クラス分類がどの程度分離できるかを評価するタスクである。Image matching は参照画像から抽出されたパッチ画像とマッチング対象画像から抽出されたパッチ画像で、どの程度正しく対応するパッチ画像を見つけられるかを評価するタスクである。Patch retrieval は大規模なパッチ画像群からクエリパッチ画像と類似するパッチ画像を検索し、どの程度正しい検索結果が得られるかを評価するタスクである。図 6.8 に HPatches の評価タスクの例を示す。

HPatches は、視点変化を伴う画像と照明変化を伴う画像による 116 シーン・696 枚の画像で構成され、画像から検出されたキーポイントの周辺領域を  $65 \times 65$  ピクセルにリサイズしたパッチ画像をデータセットとしている。パッチ画像は、図 6.9 に示すように幾何学的な画像変化の難易度に応じて“Easy”, “Hard”, “Tough”の画像セットに分割されている。HPatches で公開されているベースライン特徴量記述子は、SIFT [1], Root SIFT [51], ORB [31], BIREF [29], Bin Boost [33], Deep Desc [70], TFeat-margin [71], TFeat-ratio [71], DC-siam [72], DC-siam2stream [72] である。これらのベースライン手法と提案手法の性能を比較する。

図 6.10 に Hpatches benchmark で評価した各手法の mean average precision (mAP) を示す。提案手法 (ORB-like) では、従来の輝度差に基づく特徴量記述子である ORB と同等以上の mAP が得られていることが確認できる。提案手法 (GLOH-like) は、patch verification の評価タスクにおいて SIFT や Root SIFT よりも高い性能が得られた。他の評価タスクにおいて、提案手法 (GLOH-like) の mAP は SIFT と同等であることが確認できる。Convolutional Neural Network (CNN) をベースとした特徴量記



(a) 視点変化



(b) 照明変化

図 6.9: Hatches の画像セット例.

述子である Deep Desc, TFeat-margin, TFeat-ratio, DC-siam, DC-siam2stream は本実験においては全体的に高い性能が得られている. CNN ベースの特徴量記述子は, 膨大な学習画像より学習された無数の畳み込みフィルタを使用して最適な局所特徴量を計算するため, 高い精度が得られたと考えられる. また, Hatches benchmark では本来比較すべき手法である ASIFT や ASR 等の視点合成に基づく局所特徴量はベースラインとして公開されていないため, ここでは性能の比較が困難であった.

### 6.3.5 処理時間の比較

ここでは, 視点合成に基づく多視点特徴量記述子のキーポイントマッチングの処理時間を比較する. 処理時間の比較には, Oxford matching dataset [50] と RDED dataset [68] から選択した 5 シーンの画像セットを使用する. 画像から検出された平均キーポイント数は 446 であり, 全ての手法で同じキーポイント検出器を使用した. 実験に使用した計算機の CPU は Intel Xeon 3.33 GHz である. 図 6.11 に各多視点特徴量記述子の処理時間の比較を示す. 図 6.11 の  $x$  軸は ASIT の処理時間を 100% として

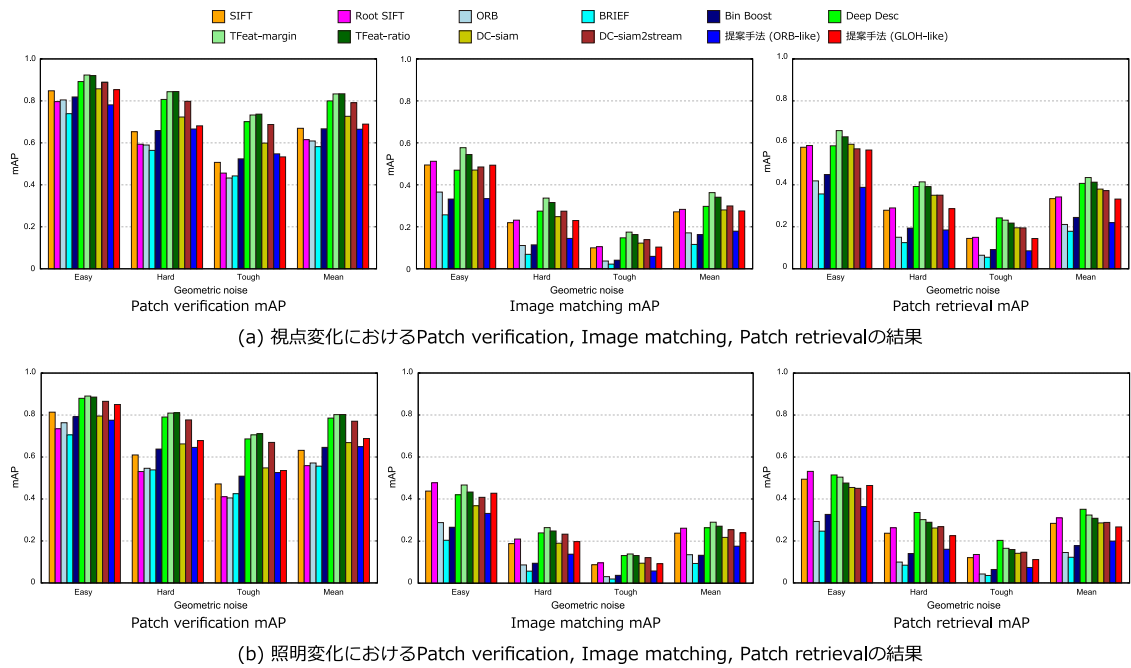


図 6.10: HPatches benchmark における特徴量記述子の評価結果.

表示している。提案手法 (GLOH-like) は ASIFT と比較して 6.6 倍高速な処理が可能である。提案手法は ASIFT のようにオンラインのアフィン変換処理を必要としないため、高速なキーポイントマッチングを実現することができたと考えられる。提案手法では、ASIFT や ASR-naive よりも多くのアフィン変換パラメータを使用しているにもかかわらず、従来法よりも高速かつ効率的なキーポイントマッチングを実現できることが確認できた。また、提案手法 (GLOH-like) は提案手法 (ORB-like) よりも高速であることがわかる。提案手法 (GLOH-like) ではパッチ画像の勾配強度や勾配方向の計算を必要とするが、提案手法 (ORB-like) よりも少ない固有フィルタ数で特徴量を記述するため、このような結果が得られたと考えられる。

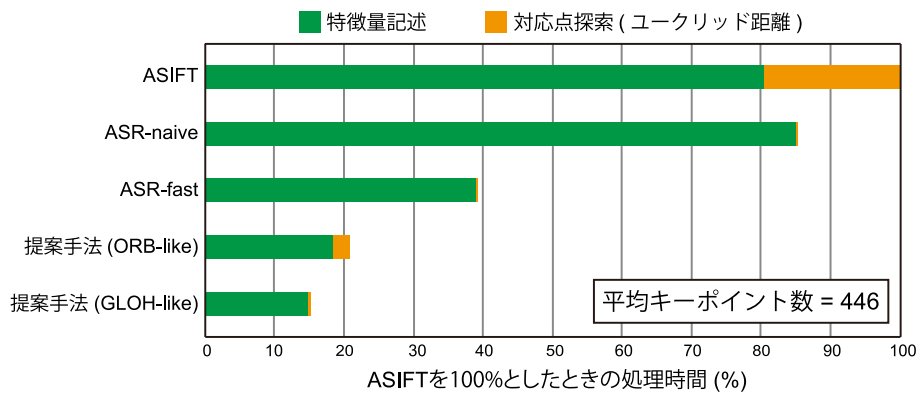


図 6.11: ASIFT の処理時間を 100% として表示した場合の各多視点特徴量記述子の比較.

### 6.3.6 まとめ

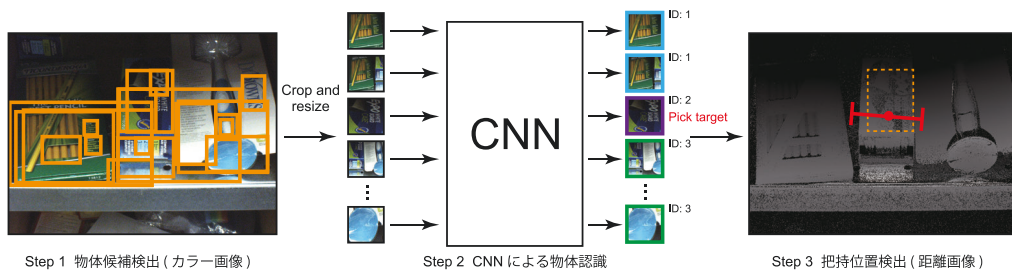
本章では，因子分解法に基づく部分空間特徴量を提案した．提案手法では，特徴量記述フィルタ群をコンパクトに近似することで，効率的に多視点特徴量を記述することが可能となった．さらに，オンラインのアフィン変換が不要であり，任意の連続アフィンパラメータにより様々な多視点特徴量を生成することで高精度なキーポイントマッチングが実現できる．評価実験より，提案手法はアフィン変換に不変なキーポイントマッチングが可能であることを示し，ASIFT や ASR よりも高速な処理を達成した．今後は，提案手法に最適な特徴量記述フィルタの設計や様々な評価タスクにおいて効果的な特徴量を記述することが課題である．

## 第7章

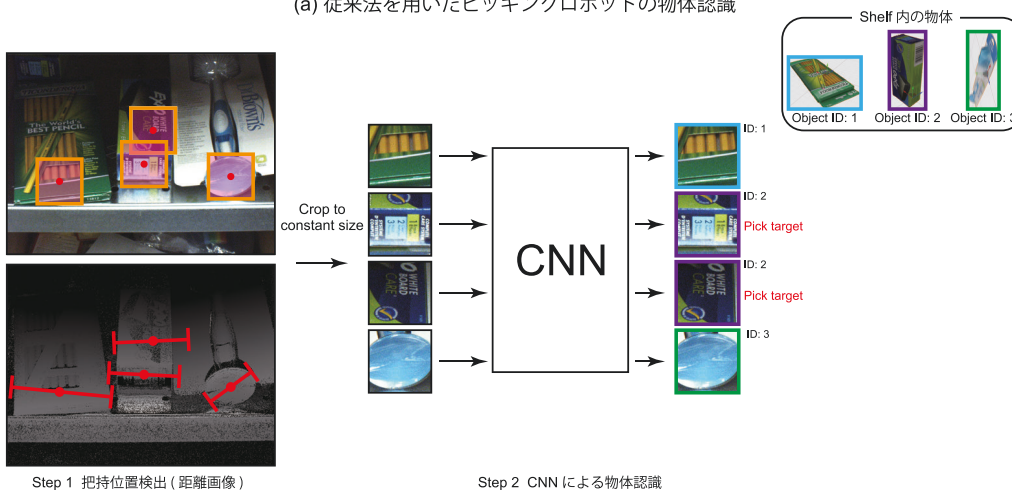
# 物流ロボットシステムにおける 特徴量マッチングを用いた物体認識

本章では、物流ロボットのための特定物体認識への応用について述べる。物流倉庫のピッキングロボットにおいて、棚 (shelf) に収納された多種多様な物体の中から指定された物体をピック&プレースする技術は重要な課題である [73]。このような課題においては、物体の把持可能位置を検出するとともに、物体がどのような物体であるかを特定しなければならない。提案手法では、ロボットの把持位置に基づいた局所画像から特徴を抽出して物体のクラスを識別する。把持位置は、物体上の把持しやすい領域を検出するように設計されており、様々な物体の中で共通するような形状を捉えて把持位置を検出する。識別時には物体の形状情報が失われるが、把持位置周辺領域の物体のテクスチャやカラー情報は得ることができる。よって、把持位置に基づく局所画像の特徴抽出と識別には、Convolutional Neural Network [74, 75, 76, 77, 78] を使用することで高精度な認識が期待できる。しかし、CNN は大量に用意した学習画像を用いて学習させた物体の識別は可能だが、学習画像に存在しない物体は認識することができない。そこで、キーポイントマッチングにおける特徴量間の距離計算の考え方を導入し、CNN から得られる特徴量をデータベース上の特徴量と照合させ、未知の物体を認識する方法も示す。

本章で扱うタスクは、shelf に収納されている物体の名前とピッキング対象物体名が記述された JSON ファイルをロボットシステムで受け取り、ピッキングのための物体認識を実行する。これは現在、Amazon.com の物流倉庫において実用化されている shelf 運搬用の自律移動ロボットによる物流システムを想定したタスクである [79]。Amazon.com での物流システムでは、Kiva と呼ばれる自律移動ロボットが、商品を取込んだ shelf をピッキング作業場所へと運搬する。このとき、運搬動作による shelf の振動により商品の配置が変化する恐れがあるため、商品の配置情報は記録されないが、shelf に収納されている商品名は記録される。このような物流システムの場合、ピッキング時の物体認識処理が実行される前に shelf 内の物体を知ることができる。そこで、物体認識結果は shelf 内の物体のみに制約をつけて出力することができる。この制約を用いることで、shelf に存在しない物体との誤認識を減らすことができ、信頼度の高い結果を返すことができる。



(a) 従来法を用いたピッキングロボットの物体認識



(b) 提案手法を用いたピッキングロボットの物体認識

図 7.1: ピック&プレースにおける物体認識の流れ.

## 7.1 ピッキングロボットのための物体認識

ピッキングロボットにおける物体認識は、画像中にどのような物体がどこにあるかを特定した上で把持位置を検出しなければならない。そのため、従来の物体検出法を利用したピッキングロボットのための物体認識は以下の3段階の処理が必要となる(図 7.1(a)).

**Step 1:** 画像全体からの物体候補検出

**Step 2:** 検出した候補領域を CNN で識別

**Step 3:** 識別した物体領域から把持可能な位置を検出

ここでは、Step 1 と Step 2 が物体検出の処理に相当し、Step 3 がピッキングのための把持位置検出の処理である。物体認識は、CNN を用いることで高い認識精度が得られるが、様々な重みフィルタを繰り返し画像に畳み込むため計算コストが高い。画像全体をラスタスキャンすると数万～数十万単位の検出ウィンドウ画像を CNN で識別するため、膨大な計算時間が必要となる。そこで、region proposal と呼ばれる物体候補領域をあらかじめ検出しておくことで、CNN での識別処理コストの増加を抑えるアプローチが用いられる。Region proposal として検出される領域は、物体が存在する領

域として信頼度の高い領域のみを検出する。そのため、region proposal に基づく CNN の物体検出は、ラストスキャンで画像を網羅的に CNN で識別するよりも、はるかに計算コストを抑えられるため有力な物体検出法である。Region proposal 検出法と CNN を組み合わせた物体検出法として、Regions with Convolutional Neural Network (R-CNN) [80] が提案されている。Region proposal を検出する手法には、類似するセグメンテーションを繰り返し統合する Selective search [81]、画像の勾配やエッジに着目した BING [82]、Edge boxes [83] 等が提案されている。R-CNN は、region proposal 検出法として Selective search を採用している。

R-CNN により、画像中の物体位置検出と識別を高精度に求めることができるが、それでも物体候補領域を繰り返し CNN で識別する計算コストや、Selective search 等による region proposal 検出自体の計算コストが高い理由で、実用的な処理時間で物体を認識することができない。これらの問題を解決するために、Fast R-CNN [84] と Faster R-CNN [85] が提案されている。Fast R-CNN では、CNN の処理時間に関して最もボトルネックである重みフィルタの畳み込み処理を画像全体に 1 度だけ適用する。そして、Selective search により入力画像から検出した region proposal の矩形を、畳み込み処理で得られた特徴マップに射影する。特徴マップ上に射影した各矩形領域を比較的計算コストの低い全結合ニューラルネットワークにより識別することで、物体検出処理を高速化している。Faster R-CNN では、region proposal 検出も CNN のフレームワークで学習させ、より良い物体候補を検出し、Fast R-CNN と同様のアプローチを組み合わせることで R-CNN や Fast R-CNN よりも高精度かつ高速な物体検出を実現している。Fast / Faster R-CNN においても、Step {1, 2, 3} で構成される 3 段階の物体認識に当てはまる。

R-CNN や Fast R-CNN における region proposal 検出は、物体にプリントされている文字や模様、ロゴマーク等のテキストチャを捉えて region proposal を過剰に検出することがある。そのため、ニューラルネットワークによる識別回数が増加し、処理時間が遅くなる。また、Faster R-CNN では region proposal 検出、物体矩形回帰、物体クラス識別の 3 つのタスクを 1 つの CNN で処理するため 18 層の大規模なネットワークを必要とする。さらに、region proposal に基づく手法は、検出した物体矩形内に複数の物体が存在すると、Step 3 において対象物体とは異なる物体から把持位置を検出することがあり、誤った物体を把持する可能性がある。

そこで、提案手法では物体の把持位置に基づいた効率的な物体認識手法を提案し、コンパクトかつシンプルなネットワーク構造での物体認識を実現させる。また、物体の把持位置は画像中の複数の物体上に検出されることを考えると、検出された把持位置領域は物体候補と見なすことができる。よって、Step 1 において把持位置を検出し、検出した把持位置を物体候補とすることで、以下のような 2 段階構造の効率的な物体認識が実現できる (図 7.1(b))。

**Step 1:** 画像全体から把持可能位置を検出

**Step 2:** 検出した把持位置領域を CNN で識別

Step 1 では把持位置検出が物体候補の役割も担うため、Faster R-CNN のように region proposal を含めた CNN を学習する必要はなく、全 8 層のコンパクトなネットワークで認識問題を解くことができる。提案手法では、検出された把持位置の周辺領域を CNN へ入力し、把持位置が検出された物体を



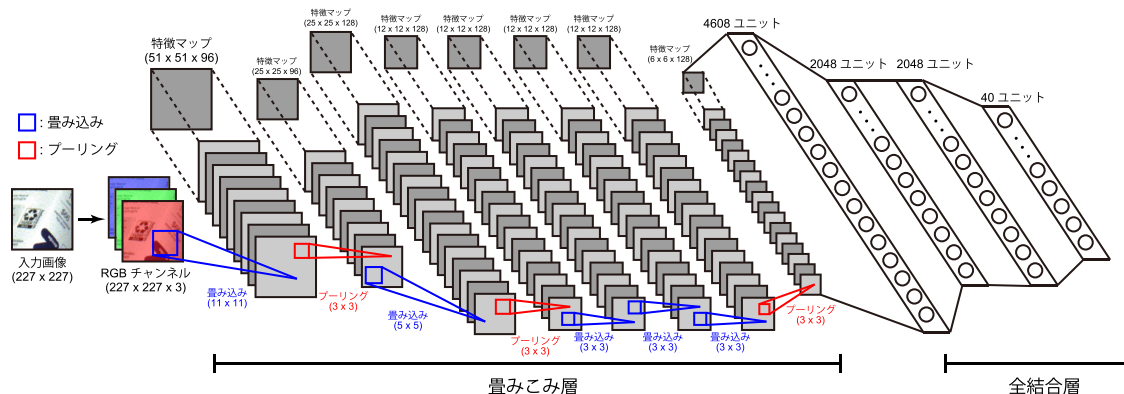


図 7.2: 提案手法の CNN の構造.

識別することで、そのままピッキング動作に移ることができるため、ピッキングロボットのための効率的な物体認識が可能である。

## 7.2 把持位置に基づくマルチクラス物体認識

Convolutional Neural Network による把持位置に基づくマルチクラス物体認識を提案する。提案手法では、まず画像中の全ての物体から把持可能な位置を検出し、検出した把持位置の周辺領域をパッチ画像として切り出した後、CNN で局所的な特徴抽出と識別をする。前処理として把持位置を検出するため、様々な物体の把持位置を高速に検出できる Fast Graspability Evaluation [86] を使用する。Fast Graspability Evaluation はロボットの 2 次元ハンドモデルをテンプレートとして保持し、距離画像から得られた物体セグメンテーションに対して 2 次元ハンドモデルが最適に当てはまる位置を把持位置として検出する。

### 7.2.1 把持位置に基づく Convolutional Neural Network の構築

提案手法は、画像から検出された把持位置に対して CNN により物体を認識する。提案手法の CNN は図 7.2 に示すように畳み込み層と全結合層で構成される。畳み込み層は、重みフィルタを画像または特徴マップに畳み込み、その応答値を活性化関数に通して特徴マップとする。その後、プーリングにより特徴マップを縮小させる。活性化関数には Rectified Linear Unit (ReLU) [87]、プーリングには Max pooling を用いる。また、ミニバッチごとに特徴マップの分布が平均 0、分散 1 となるように正規化する Batch Normalization [88] を導入する。提案手法の CNN 構成の詳細を表 7.1 に示す。提案手法の把持位置に基づいた CNN は、検出された把持位置の周辺領域を固定サイズで切り取った RGB 画像が入力となる。CNN の入力画像サイズは 227 × 227 ピクセルとする。全結合層の出力ユニットはクラス数、すなわち認識対象の物体数+背景クラスとなり、最も応答値の高いユニットが認識クラスとなる。把持位置に基づいた CNN により、検出された把持位置のみに認識処理を実行すれば良い

表 7.1: 提案手法の CNN 構成の詳細.

層	詳細
1 層目 (畳み込み)	フィルタサイズ : $11 \times 11$
	ストライド : 4
	パディング : 0
	正規化 : Batch Normalization
	活性化関数 : ReLU
	プーリング : Max pooling
2 層目 (畳み込み)	フィルタサイズ : $5 \times 5$
	ストライド : 1
	パディング : 2
	正規化 : Batch normalization
	活性化関数 : ReLU
	プーリング : Max pooling
3 層目 (畳み込み)	フィルタサイズ : $3 \times 3$
	ストライド : 1
	パディング : 1
	正規化 : -
	活性化関数 : ReLU
	プーリング : -
4 層目 (畳み込み)	フィルタサイズ : $3 \times 3$
	ストライド : 1
	パディング : 1
	正規化 : -
	活性化関数 : ReLU
	プーリング : -
5 層目 (畳み込み)	フィルタサイズ : $3 \times 3$
	ストライド : 1
	パディング : 1
	正規化 : -
	活性化関数 : ReLU
	プーリング : Max pooling
6 層目 (全結合)	活性化関数 : ReLU Dropout (学習時) : ○
7 層目 (全結合)	活性化関数 : ReLU Dropout (学習時) : ○
8 層目 (全結合)	活性化関数 : Softmax Dropout (学習時) : -

ため計算コストを抑えることができる。また、最初に把持可能な位置を全て検出するため、CNN で認識した後にそのままピッキングが可能であるため効率的である。

## 7.2.2 学習画像

CNN の学習画像は、単一の物体を撮影した画像を使用する。学習画像の撮影は三菱電機社製の産業用ロボット MELFA-3D Vision を用いる。MELFA-3D Vision はカメラとプロジェクタで構成され、アクティブステレオ法により RGB 画像と距離画像を撮影する。画像は、単一物体を様々な姿勢で

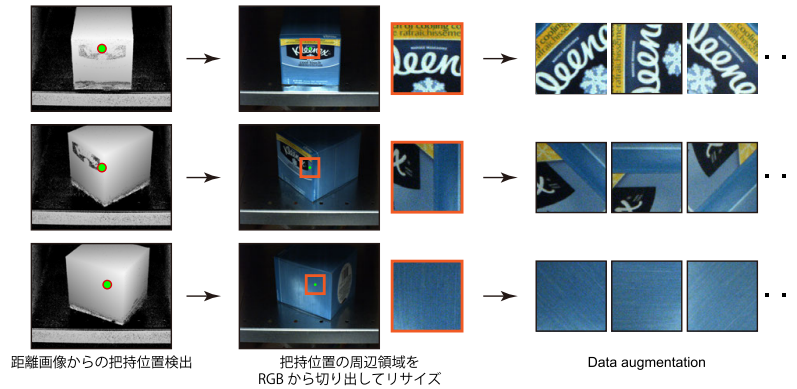


図 7.3: 学習画像の生成.



図 7.4: 学習用パッチ画像の例.

配置して撮影する。撮影した画像から Fast Graspability Evaluation [86] により把持位置を検出し、検出された把持位置の周辺領域を  $227 \times 227$  ピクセルにリサイズして学習用パッチ画像とする。CNN の物体認識は、様々な姿勢や環境下においても認識精度を保つために Data Augmentation により学習画像のバリエーションを増加させる。学習画像生成の流れを図 7.3 に示す。提案手法では、Data Augmentation として回転、平行移動、照明変化、ノイズ付加をランダムで画像に施して学習用パッチ画像を増加させる。Data Augmentation により生成した学習用パッチ画像の例を図 7.4 に示す。

### 7.2.3 制約付き softmax

CNN によるクラス識別では、softmax 関数によりクラス確率を算出する (図 7.5(a))。CNN の出力

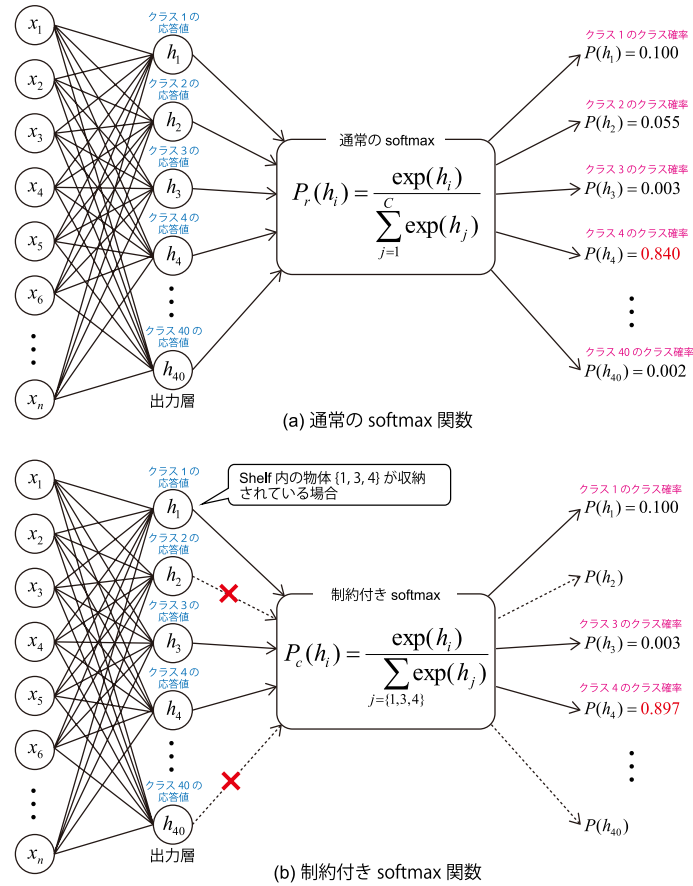


図 7.5: softmax 関数の計算.

ユニットの値を  $h$ , クラス数を  $C$  とすると, クラス確率は式 (7.1) の softmax 関数  $P_r(\cdot)$  により算出される.

$$P_r(h_i) = \frac{\exp(h_i)}{\sum_{j=1}^C \exp(h_j)} \quad (7.1)$$

ここで,  $i$  は CNN の出力ユニットの値  $h$  に対するインデックスである. 本研究では, shelf に存在する物体に対応する出力ユニットのみを用いて計算する制約付き softmax 関数によりクラス確率を計算する (図 7.5(b)). 例えば, shelf に  $\{1, 3, 4\}$  番目の出力ユニットに対応する物体が収納されているとき, 制約付き softmax 関数  $P_c(\cdot)$  は式 (7.2) のように定義できる.

$$P_c(h_i) = \frac{\exp(h_i)}{\sum_{j=\{1,3,4\}} \exp(h_j)} \quad (7.2)$$

制約付き softmax 関数は shelf に収納されている複数の物体が既知である場合において有効な関数であり, shelf に存在しない物体との誤認識を減らすことができる. 制約付き softmax 関数  $P_c(\cdot)$  によ

る物体識別は inference 処理で使用し，学習時には通常の softmax 関数  $P_r(\cdot)$  を用いて CNN を学習させる。

## 7.2.4 特徴量マッチングによる未学習物体の認識

CNN は，出力ユニットのクラス確率を softmax 関数により算出することで，画像の識別問題を高精度に解くことができる。しかし，CNN の出力層のユニット数は学習データの物体クラス数に対応しているため，未学習の物体は認識することができない。そこで，CNN の出力層の手前の全結合層の出力を 2048 次元の特徴ベクトルとすることで，特徴量をマッチングさせる。まず，学習済み物体や未学習物体の画像の特徴ベクトルをあらかじめデータベースに保持する。inference 時には入力画像の特徴ベクトルをクエリとしてデータベース内の特徴ベクトル群との距離を計算し，距離が最も近い特徴ベクトルの物体クラスを認識結果とする。この方法は，単純な特徴ベクトル間の距離計算で物体を認識するため，未学習の物体が存在してもあらかじめ特徴量データベースさえ生成しておけば CNN を再学習することなく未学習物体を認識することができる。

提案手法は検出した把持位置の周辺領域を CNN へ入力して特徴を抽出するため，CNN は局所特徴量記述子とみなすことができる。よって，(1) 把持位置 (キーポイント) の検出，(2) CNN による局所特徴量記述，(3) 距離計算によるマッチング，というようにキーポイントマッチングと同様の処理で物体認識が可能となる。特徴量マッチングによる物体認識の有効性は，7.3.4 項にて実験的に示す。

## 7.3 評価実験

提案手法の有効性を確認するために評価実験をする。本実験では，R-CNN [80]，Faster R-CNN [85] と提案手法である把持位置に基づく CNN の認識精度と処理時間を比較する。R-CNN の region proposal 検出には Selective search [81] を使用する。

### 7.3.1 データセット

本実験で使用するデータセットは国際物流ロボットコンペティションで Amazon Picking Challenge (APC) で使用された商品を使用する。2015 年に開催された APC 2015 では全 25 種類の物体，2016 年に開催された APC 2016 では全 39 種類の物体が使用された。図 7.6 に APC 2015 の認識対象物体，図 7.7 に APC 2016 の認識対象物体を示す。CNN の学習には shelf の中に単一の物体が配置されている画像のみを用いる。APC 2015 の学習画像は 750 枚，APC 2016 の学習画像は 1,709 枚である。評価用画像は shelf の中に複数の物体が配置されている画像を使用する。APC 2015 の評価画像は 594 枚，APC 2016 の評価画像は 200 枚である。



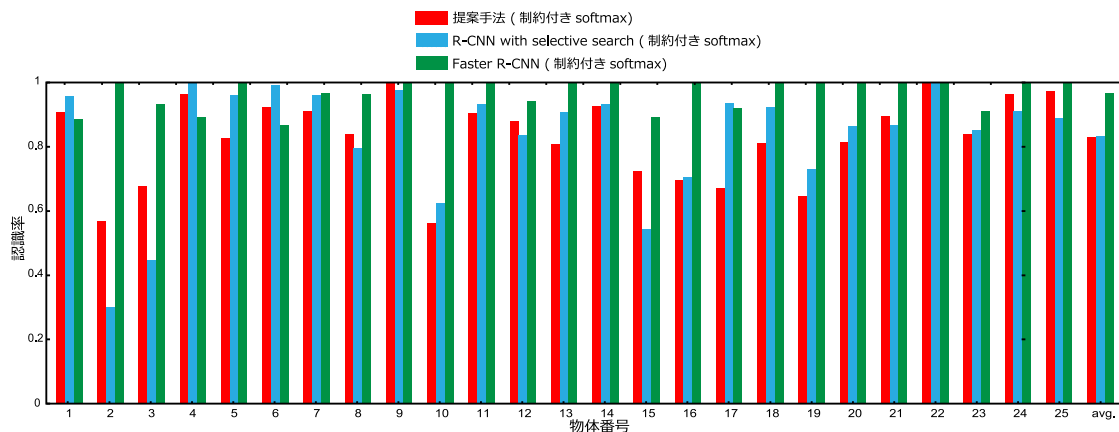
図 7.6: Amazon Picking Challenge 2015 の認識対象物体.



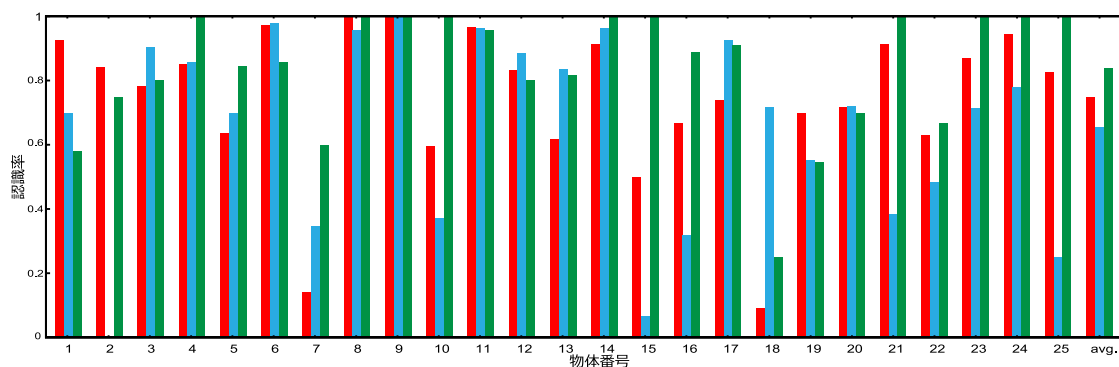
図 7.7: Amazon Picking Challenge 2016 の認識対象物体.

### 7.3.2 物体認識における精度

R-CNN, Faster R-CNN と提案手法である把持位置に基づく CNN の認識精度を比較する. 各手法において, 制約付き softmax 関数を使用して認識結果を出力する. 制約付き softmax 関数は, shelf に収納されている物体クラスのみでクラス確率を計算し, その中での最大値を推定クラスとする. APC 2015 データセットの各物体の認識率を図 7.8 に示す. APC 2015 データセットでは, shelf の中に 2 個



(a) Shelf内の物体数 = 2



(b) Shelf内の物体数 = 3 - 4

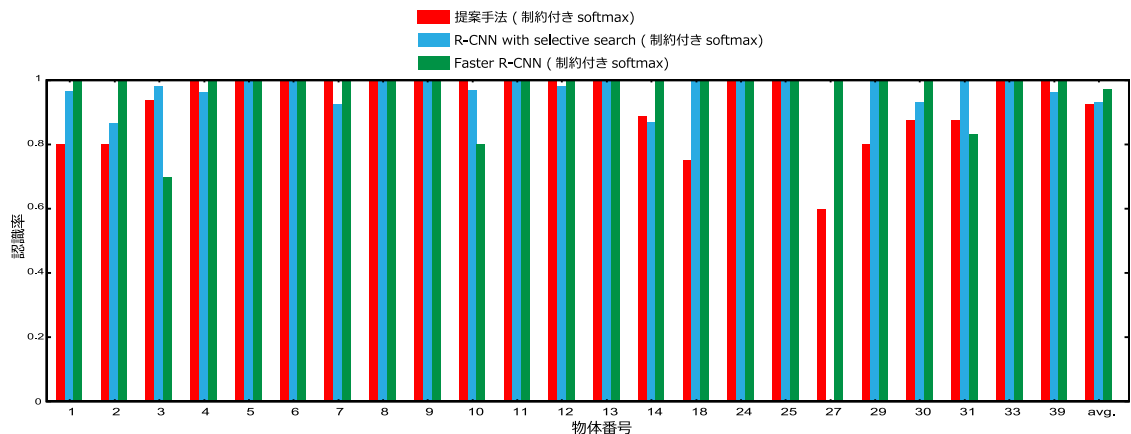
図 7.8: APC 2015 データセットの認識率.

の物体が配置されている場合と 3~4 個の物体が配置されている場合において評価する。グラフの横軸は物体番号であり、図 7.6 に示す番号に対応している。グラフの最終列は全物体の平均認識率である。

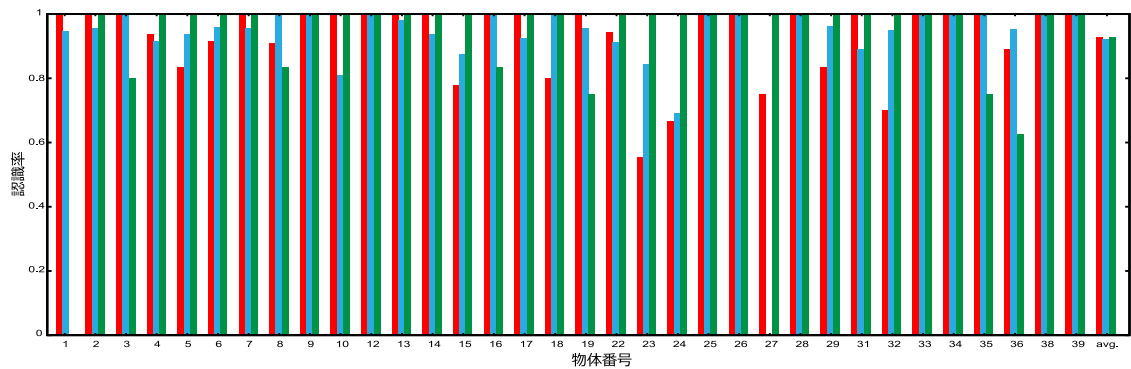
shelf 中の物体数が 2 のとき、提案手法は R-CNN と同等の平均認識率が得られていることが確認できる (図 7.8(a))。shelf 中の物体数が 3~4 の場合において、提案手法は R-CNN よりも平均認識率が 9.2% 向上した (図 7.8(b))。

APC 2016 データセットの各物体の認識率を図 7.9 に示す。APC 2016 データセットでは、shelf の中に 2~3 個の物体が配置されている場合、4~5 個の物体が配置されている場合、6~10 個の物体が配置されている場合において評価する。グラフの横軸の物体番号は、図 7.7 に示す番号に対応している。提案手法は R-CNN と同等以上の平均認識率であり、物体数が 6~10 個の場合においては、Faster R-CNN の平均認識率を 4.3% 上回る結果が得られた。以上の結果から、提案手法は R-CNN と同等以上の認識精度で効率的なピッキングロボットシステムに応用できると言える。

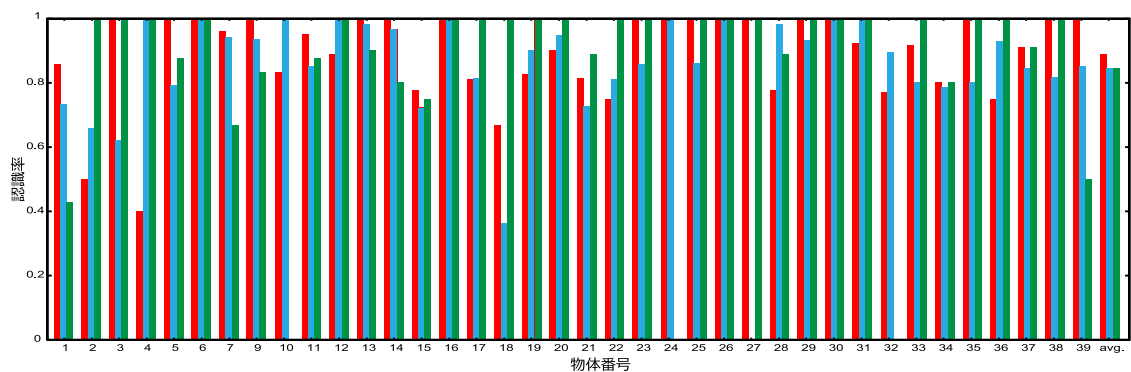
本実験では Faster R-CNN の精度が最も高いが、これは region proposal の検出も CNN によって獲得するため、識別に適した物体候補が検出できていると考えられる。しかし、Faster R-CNN は region proposal 検出、物体矩形回帰、物体クラス分類を全て CNN により処理するため、提案手法の CNN



(a) Shelf内の物体数 = 2 - 3



(b) Shelf内の物体数 = 4 - 5



(c) Shelf内の物体数 = 6 - 10

図 7.9: APC 2016 データセットの認識率.

と比較して大規模なネットワークを必要とする。

提案手法は、APC 2016 データセットに関しては Faster R-CNN と同等の精度であり、R-CNN や Faster R-CNN よりも短い処理時間で効率的に物体を認識することができる。各手法の処理時間については、7.3.5 項で比較する。





図 7.10: Faster R-CNN により検出された物体矩形内の把持位置検出の例。

### 7.3.3 把持位置検出における精度

ここでは、物体認識後の把持位置検出の正解率を比較する。7.3.2 項において、R-CNN や Faster R-CNN は検出した物体矩形に対する認識率を比較した。R-CNN や Faster R-CNN 等の region proposal に基づく手法は、認識物体の位置を矩形領域として検出した後、検出した矩形領域内から把持位置を検出する必要がある。R-CNN や Faster R-CNN は以下の 2 つの要因で誤った物体を把持することがある。

1. 物体矩形の誤認識による誤った物体の把持。
2. 物体矩形内の異なる物体の写り込みによる把持位置の誤検出。

1 つ目の要因については、7.3.2 項の実験で示す認識精度の割合で誤った物体を把持してしまう。さらに、R-CNN や Faster R-CNN では 2 つ目の要因が発生する。検出した物体矩形のクラスが正解であっても、図 7.10 に示すように shelf の中に多数の物体が密集して配置されている場合、物体矩形内の異なる把持位置を検出する場合がある。一方、提案手法は把持位置ごとに物体を認識するため、誤った物体を把持する要因は誤認識した場合のみである。R-CNN や Faster R-CNN により検出した物体矩形から Fast Graspability Evaluation [86] を用いて把持位置候補を検出し、把持位置候補ごとに正解率を求めると表 7.2 となる。提案手法は検出した把持位置から物体を識別する手法であるため、7.3.2 項で示した認識率と同じ精度となる。また、全ての手法において shelf 中の物体のみを対象とする制約付き softmax を適用する。ARC 2015 データセットの shelf 内の物体数が 2 の場合は、物体同士の重なりが少ないため Faster R-CNN の正解率が高い結果となる。しかし、shelf 中の物体数が多くなると R-CNN や Faster R-CNN は複数の物体を 1 つの矩形で検出するケースが多くなり、把持位置の正解率が大幅に低下する。提案手法は、shelf 中の物体数が 3 以上の全てのデータセットにおいて最も良い精度であることが確認できる。

表 7.2: 物体矩形内の把持位置検出の正解率 [%].

データセット	物体数	提案手法	R-CNN with selective search	Faster R-CNN
APC 2015	2	82.88	83.31	90.36
APC 2015	3 - 4	74.71	55.92	68.66
APC 2016	2 - 3	92.72	79.97	91.51
APC 2016	4 - 5	92.69	56.85	82.49
APC 2016	6 - 10	88.89	55.61	74.54

### 7.3.4 特徴量マッチングによる認識精度

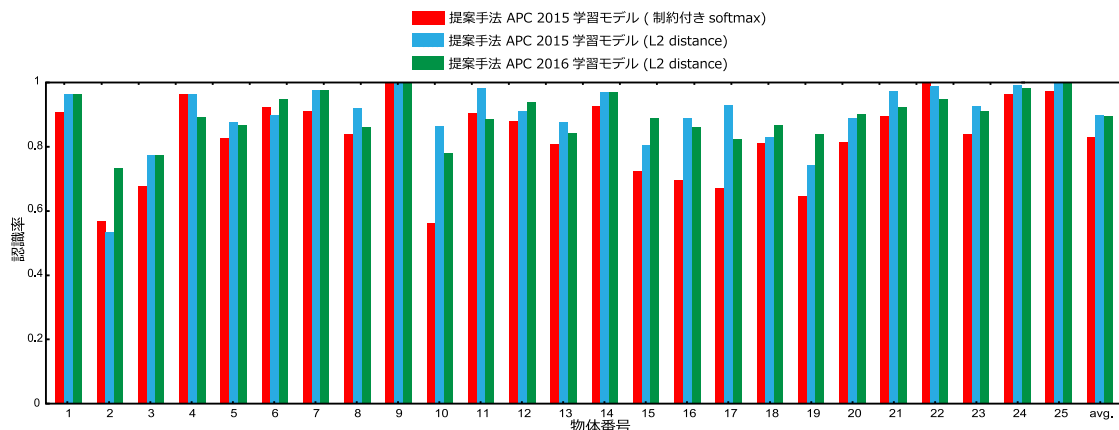
ここでの実験は、特徴量マッチングによる物体認識精度を検証する。また、CNN の特徴量のマッチングによる認識が妥当な精度であるかを確かめるために softmax 関数を使用した提案手法の精度と比較する。特徴量マッチングによる物体認識は、CNN の学習用画像として用意した単一物体の画像から得られた特徴ベクトルをデータベースとして保持しておき、クエリとなる画像の特徴ベクトルとのユークリッド距離でマッチングする。ARC 2015 データセットの各物体の認識率を図 7.11 に示す。凡例に示す“APC 2015 学習モデル”は APC 2015 データセットにより学習した CNN，“APC 2016 学習モデル”は APC 2016 データセットにより学習した CNN を使用したことを示している。すなわち、図 7.11 における APC 2016 学習モデルは全て未学習物体に対して識別した結果である。APC 2016 学習モデルは softmax 関数を用いた APC 2015 学習モデルと同等の精度を達成していることから、特徴量マッチングによる未学習物体の認識は有効であることがわかる。

ARC 2016 データセットの各物体の認識率を図 7.12 に示す。図 7.12 の場合では、APC 2015 学習モデルが未学習物体に対する識別結果となる。shelf 内の物体数が 2~3 の場合、APC 2015 学習モデルは softmax 関数を用いた APC 2016 学習モデルに匹敵する精度が得られていることが確認できる。shelf 内の物体数が 4~5、6~10 の場合においても特徴量マッチングによる手法は 70%以上の認識率が得られている。

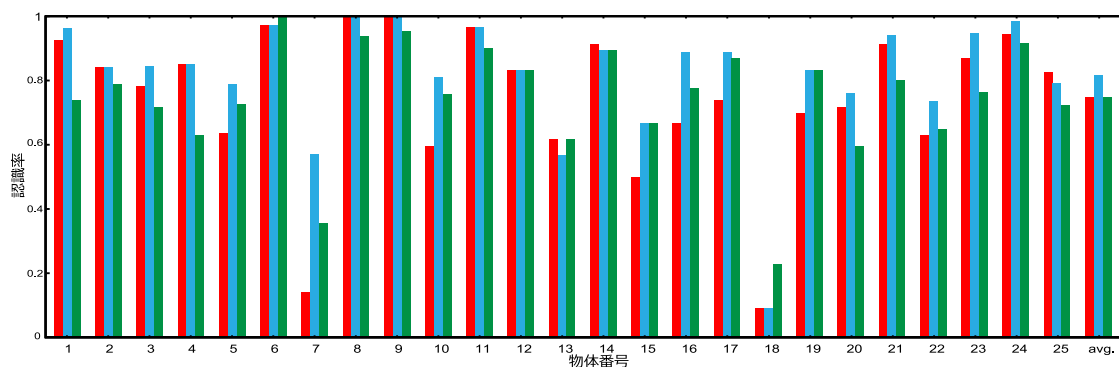
以上の結果から、提案手法である把持位置に基づく CNN は、出力層手前の特徴ベクトルの距離計算に基づいて認識することで、学習されていない物体クラスも識別可能であることがわかる。これは、把持位置に基づいた局所画像の識別問題であるため、物体の形状や大きさに影響を受けない特徴量が CNN により記述されていると考えられる。

### 7.3.5 処理時間

APC 2016 データセットのうち、物体数が 2~3、4~5、6~10 個の画像をそれぞれ 10 枚ずつ用意し、合計 30 枚の画像の平均処理時間を比較する。処理時間は CPU : Intel Core i7-7700 3.6-4.2GHz, GPU : NVIDIA GeForce GTX 1080 Ti を用いて計測する。提案手法の平均把持位置数は 5、R-CNN の平均物体候補数は 14 である。全手法において、把持位置検出は Fast Graspability Evaluation を用いる。表 7.3 に認識に必要な処理時間の内訳を示す。表 7.3 より、提案手法の CPU での処理時間は R-CNN



(a) Shelf 内の物体数 = 2



(b) Shelf 内の物体数 = 3 - 4

図 7.11: APC 2015 データセットの特徴量マッチングによる認識率.

表 7.3: 認識の処理時間の内訳 [ms].

	提案手法		R-CNN with selective search		Faster R-CNN	
	CPU	GPU	CPU	GPU	CPU	GPU
Selective search	-		1662.3		-	
把持位置検出	579.6		570.1		559.4	
CNN による識別	245.1	8.9	622.8	26.8	10551.9	69.2
合計	824.7	588.5	2855.2	2259.2	11111.3	628.6

と比較して約 3.4 倍, Faster R-CNN と比較して約 13.4 倍高速であることが確認できる. R-CNN は, Selective search による region proposal 検出の処理時間の割合が非常に大きい. また, Selective search は物体にプリントされている文字やロゴマーク等のテキストチャに反応して過剰に物体候補を検出するため, CNN の実行回数が多くなり処理時間が遅くなる. Faster R-CNN は, region proposal 検出, 物体矩形回帰, 物体クラス分類の 3 つのタスクを 1 つの CNN で処理するため, 提案手法の CNN と比較して大規模なネットワークが必要となる. GPU を用いた場合においても, 提案手法の処理時間が最も短いため, 提案手法は高速かつ効率的にマルチタスク物体認識が可能であると言える.

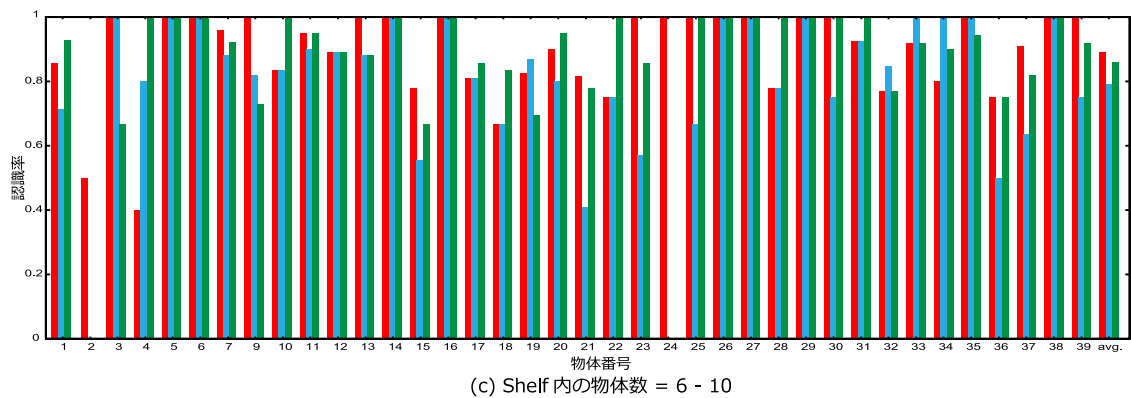
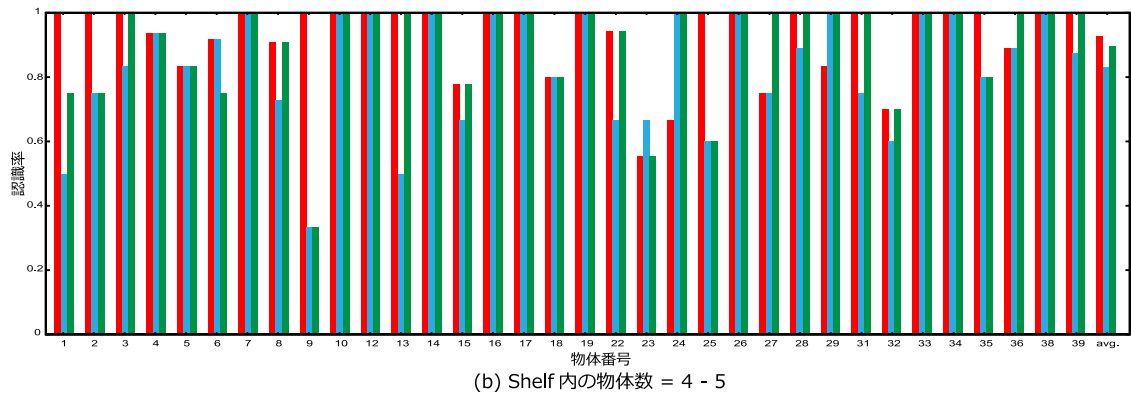
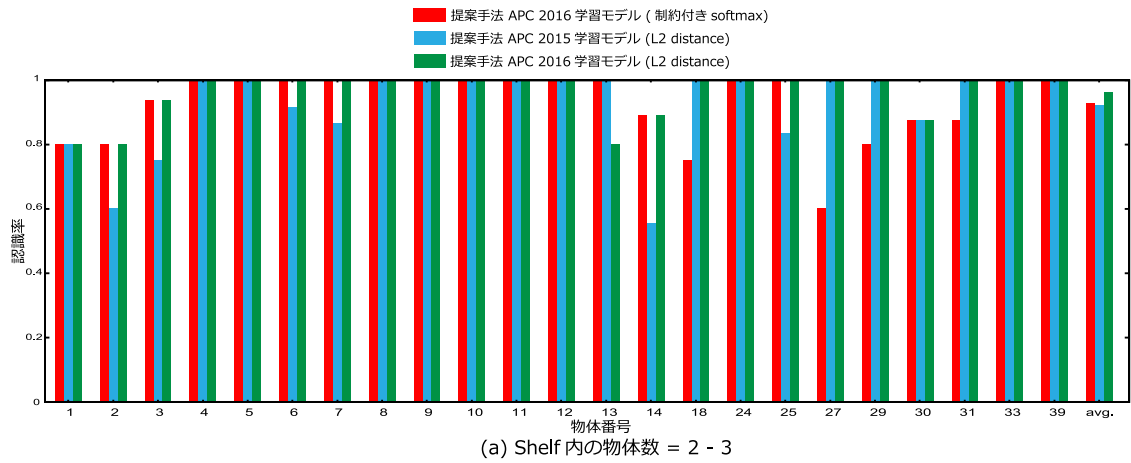


図 7.12: APC 2016 データセットの特徴量マッチングによる認識率.

### 7.3.6 まとめ

本章では、Convolutional Neural Network による把持位置に基づいたピッキングロボットのための物体認識法を提案した。把持位置に基づく CNN による物体認識は R-CNN と同等以上の認識精度が得られ、Faster R-CNN よりも高速な処理が可能である。また、把持位置検出まで含めた認識対象物体の正解率は、提案手法が最も高い精度であることを確認した。さらに、CNN の出力層手前の特徴

ベクトルをデータベース上の特徴ベクトルと照合することで、未学習物体に対しても識別することが可能である。

評価実験では、Amazon Picking Challenge で使用された 25 種類または 39 種類の物体を対象としたが、実際の物流倉庫を想定した大規模な種類の物体において、提案手法の高精度化が望まれる。このような問題に対処するには、物体識別と把持位置検出を単一の CNN の枠組みで実現することが考えられる。物体識別と把持位置検出を同時に解くような CNN を構築することで、把持しやすさと識別に有効な把持位置を同時に学習できると考えており、より高精度な物体識別が期待できる。

## 第8章

# 結論と展望

本論文では、視点変化を伴う画像に対して高精度かつ効率的なキーポイントマッチングを実現させるために、因子分解法に基づく複数のアフィン領域推定や多視点特徴量の記述について述べた。以下に本論文の結論と今後の展望について述べる。

### 8.1 結論

各章のまとめは次の通りである。2章では、キーポイントマッチングの具体的な処理の流れを述べた後、キーポイントマッチングに用いられる処理であるキーポイント検出器と局所特徴量記述子のサーベイを行った。キーポイント検出では、スケールスペースやオリエンテーション、アフィン領域を推定することで、視点変化を伴う画像に対してもキーポイントを対応付けられるようになった。また、局所特徴量記述においても視点変化に対して頑健な特徴量を記述する手法や省メモリ化・高速化に焦点を当てたシンプルな手法など、様々な局所特徴量記述子が提案された。

3章では、検出されるキーポイントがキーポイントマッチングの計算コストにどのように影響を及ぼすかに着目し、不必要なキーポイントの過剰な検出を抑制しつつ高速にキーポイントを検出する Cascaded FAST を提案した。キーポイントは画像の局所領域におけるエッジやテクスチャ情報に基づいて検出するため、テクスチャが複雑な自然領域から多くのキーポイントを検出してしまう。テクスチャが複雑な領域から検出されるキーポイントとそうでないキーポイントの輝度情報にどのような傾向が得られるかを調査し、傾向の違いを捉えるようなキーポイントをカスケード構造の決定木を用いて高速に検出した。この結果、従来のキーポイント検出器と同等の精度を維持しつつ、高速なキーポイントマッチングを実現した。

4章では、視点変化を伴う画像間のキーポイントマッチングを高精度化するために、検出されたキーポイントに対して複数のアフィン領域を推定する方法を提案した。キーポイントに対して複数のアフィン領域を推定するには、大量の非等方性 LoG フィルタを検出されたキーポイント毎に畳み込む必要があるため、高い計算コストを必要とするが、非等方性 LoG フィルタ群に対して因子分解法を適用することにより、効率的に複数のアフィン領域を推定することが可能であることを示した。評価実験により、従来のアフィン領域推定手法よりも高精度であることを確認した。さらに、キーポイントマッチングによる画像検索の問題に対しても提案手法が有効である結果が得られた。

5章では、4章のアプローチを局所特徴量記述に応用した。局所特徴量はパッチ画像内のピクセルペアの輝度差等により特徴量を生成する線形モデルにおいて、畳み込みフィルタの形式で表現する

ことができ、この畳み込みフィルタに様々なアフィン変換を適用することで画像間の強い視点変化に対して高精度な特徴量記述を行う。アフィン変換された大量の畳み込みフィルタは、因子分解法を適用することで、少ない基底フィルタと重み係数の線形演算で近似可能であるため、効率的に特徴量を記述することができる。また、特徴量間の距離計算を最小2乗法の形式で表現することで、特徴量間距離の下界を算出し、効率的な対応点探索を実現した。

6章では、5章で提案した特徴量記述子を線形モデルではなく勾配方向ヒストグラムモデルへと拡張した。勾配方向ヒストグラムに基づく特徴量記述は、非線形処理が存在するため因子分解による特徴量表現が困難であったが、入力パッチ画像の勾配画像に工夫を加えることで、因子分解に基づく勾配方向ヒストグラムモデルの特徴量表現を実現した。さらに、様々な視点で記述した特徴量群をアフィン部分空間へ射影して特徴量を構成することで、より視点変化に頑健な特徴量を獲得することができた。評価実験では、従来のアフィン変換に基づく特徴量記述子と比較して同等以上の精度が得られたことを確認し、特徴量の計算時間も従来法よりも大幅に削減することができた。

7章では、物流ロボットシステムにおける特徴量マッチングを用いた物体認識を実現した。ピッキングロボットの把持位置を利用することで、物体上の局所的な特徴ベクトルをCNNにより算出し、この特徴ベクトルを用いてマルチクラスの物体を実用的な精度で認識した。クラス確率に基づくCNNの物体認識は未学習の物体クラスを識別することができないが、CNNの最終層手前から得られる特徴量を用いてデータベース上の特徴量とマッチングすることで未学習の物体クラスも識別することが可能となった。

## 8.2 展望

本論文では、キーポイント検出の解析に基づくキーポイントマッチングの高速化と視点変化にロバストなキーポイントマッチングのための因子分解に基づく局所特徴量表現を提案した。

因子分解法に基づく特徴量表現において、今後取り組むべき課題は、特徴量記述フィルタのアフィンパラメータ数をより増加させることである。線形アフィン変換におけるアフィンパラメータの中で、スケールパラメータとカメラ軸に対する面内回転パラメータはキーポイントで推定されるスケールとオリエンテーションで代用していたが、より正確な特徴量を記述するには、これらのアフィンパラメータも含めて特徴量記述フィルタをアフィン変換させることが望ましい。さらには、キーポイントの位置ずれを考慮して平行移動を加えたり、最終的には非線形な射影変換で特徴量記述フィルタを歪ませることが必要な可能性がある。しかし、これらの全ての変形パラメータを含めて特徴量記述フィルタの視点合成を行うと爆発的にフィルタ枚数が増加する。このような問題に対しては、テンソル分解等によるフィルタ構造を考慮した基底フィルタの構築や、因子分解法に適した特徴量記述フィルタを設計する必要がある。

また、特徴量の高精度化という点のみに着目するのであれば、Convolutional Neural Network (CNN) による画像の幾何学的変化に対して頑健な局所特徴量記述についても取り組んでいく必要がある。CNNによる高精度な局所特徴量記述は幾つか提案され、高い性能が得られている。より画像間の視点変化に特化した特徴量記述を求めるのであれば、敵対的生成ネットワークによりアフィン変換され

た高品質な特徴量ベクトルを生成する枠組みをつくることで、強い視点変化においてもさらなる高精度化が期待できる。また、特徴量記述を学習ベースにすることで、異なる性質の画像間に対応づけるような特徴量記述も考えられる。例えば、可視光カメラと赤外光カメラで撮影された画像、季節や昼夜などの異なる時間軸で撮影された画像、さらには歴史的建造物等の過去と現在の画像など、全く性質の異なる画像間に対応づける特徴量を実現するのであれば、CNN をベースとした特徴量記述子が有力であると思われる。



# 謝 辞

本研究は、著者が中部大学大学院工学研究科情報工学専攻博士後期課程在学中に、同大学工学部ロボット理工学科 藤吉弘巨教授の指導のもとに行ったものである。研究の遂行にあたり、常日頃ご指導を賜りました中部大学工学部ロボット理工学科 藤吉弘巨教授に深く感謝の意を表します。本論文をまとめるにあたり、有益なご討論、ご助言を賜りました中部大学工学部情報工学科 岩堀祐之教授、中部大学工学部ロボット理工学科 平田豊教授、中京大学工学部機械システム工学科 橋本学教授に謹んで感謝いたします。本研究において、貴重なご意見、ご指導を頂きました中部大学工学部情報工学科 山下隆義准教授、中部大学工学部ロボット理工学科 山内悠嗣助手、熊本大学大学院先端科学研究部 上瀧剛准教授、株式会社デンソーアイティラボラトリ 安倍満氏、吉田悠一氏、石川康太氏に心から厚く御礼申し上げます。最後に、本研究にご協力して頂いた藤吉研究室と山下研究室の皆様感謝致します。



## 参考文献

- [1] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, vol.60, no.2, pp.91–110, 2004.
- [2] 高木雅成, 藤吉弘亘, “SIFT 特徴量を用いた交通道路標識認識”, *電気学会論文誌 C (電子・情報・システム部門誌)*, vol.129, no.5, pp.824–831, 2009.
- [3] D. Nister, and H. Stewenius, “Scalable recognition with a vocabulary tree”, *Conference on Computer Vision and Pattern Recognition*, vol.2, pp.2161–2168, 2006.
- [4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints”, *Workshop on statistical learning in computer vision*, vol.1, no.1-22, pp.1–2, 2004.
- [5] C. D. Manning, H. Schütze, et al., *Foundations of statistical natural language processing*, MIT Press, 1999.
- [6] M. Brown, and D. G. Lowe, “Automatic panoramic image stitching using invariant features”, *International Journal of Computer Vision*, vol.74, no.1, pp.59–73, 2007.
- [7] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: exploring photo collections in 3D”, *ACM transactions on graphics*, vol.25, no.3, pp.835–846, 2006.
- [8] H. Durrant-Whyte, and T. Bailey, “Simultaneous localization and mapping: part I”, *Robotics & Automation Magazine*, vol.13, no.2, pp.99–110, 2006.
- [9] T. Bailey, and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II”, *Robotics & Automation Magazine*, vol.13, no.3, pp.108–117, 2006.
- [10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system”, *IEEE Transactions on Robotics*, vol.31, no.5, pp.1147–1163, 2015.
- [11] H. Moravec, “Towards Automatic Visual Obstacle Avoidance”, *International Joint Conference on Artificial Intelligence*, p.584, 1977.
- [12] H. Moravec, “Rover visual obstacle avoidance”, *International Joint Conference on Artificial Intelligence*, pp.785–790, 1981.

- [13] P. R. Beaudet, “Rotationally invariant image operators”, International Conference on Pattern Recognition, pp.579–583, 1978.
- [14] C. Harris, and M. Stephens, “A combined corner and edge detector”, Alvey Vision Conference, pp.147–151, 1988.
- [15] S. M. Smith, and J. M. Brady, “Susan & mdash; a new approach to low level image processing”, International Journal of Computer Vision, vol.23, no.1, pp.45–78, 1997.
- [16] T. Lindeberg, “Scale-space theory: A basic tool for analysing structures at different scales”, Journal of Applied Statistics, pp.224–270, 1994.
- [17] T. Lindeberg, “Feature Detection with Automatic Scale Selection”, International Journal of Computer Vision, vol.30, pp.79–116, 1998.
- [18] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded-Up Robust Features”, Computer Vision and Image Understanding, vol.110, no.3, pp.346–359, 2008.
- [19] K. Mikolajczyk, and C. Schmid, “Indexing Based on Scale Invariant Interest Points”, International Conference on Computer Vision, pp.525–531, 2001.
- [20] G. Koutaki, and K. Uchimura, “Scale-space Processing Using Polynomial Representations”, Conference on Computer Vision and Pattern Recognition, pp.2744–2751, 2014.
- [21] T. Tuytelaars, and L. V. Gool, “Content-based Image Retrieval based on Local Affinely Invariant Regions”, International Conference on Visual Information Systems, pp.493–500, 1999.
- [22] T. Tuytelaars, and L. V. Gool, “Wide baseline stereo matching based on local, affinely invariant regions”, British Machine Vision Conference, pp.412–425, 2000.
- [23] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions”, British Machine Vision Conference, pp.36.1-36.10, 2002.
- [24] K. Mikolajczyk, and C. Schmid, “Scale & Affine Invariant Interest Point Detectors”, International Journal of Computer Vision, vol.60, no.1, pp.63-86, 2004.
- [25] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, “A Comparison of Affine Region Detectors”, International Journal of Computer Vision, vol.65, no.1-2, pp.43–72, 2005.
- [26] Y. Ke, and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors”, Conference on Computer Vision and Pattern Recognition, vol.2, pp.II–506, 2004.

- [27] S. Wu, and M. S. Lew, “RIFF: Retina-inspired Invariant Fast Feature Descriptor”, ACM international conference on Multimedia, pp.1129–1132, 2014.
- [28] E. Tola, V. Lepetit, and P. Fua, “DAISY: An efficient dense descriptor applied to wide-baseline stereo”, Pattern Analysis and Machine Intelligence, vol.32, no.5, pp.815–830, 2010.
- [29] C. Michael, L. Vincent, S. Christoph, and F. Pascal, “BRIEF: binary robust independent elementary features”, European Conference on Computer Vision, pp.778–792, 2010.
- [30] S. Leutenegger, M. Chli, and R. Siegwart, “BRISK: Binary Robust Invariant Scalable Keypoints”, International Conference on Computer Vision, pp.2548–2555, 2011.
- [31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An Efficient Alternative to SIFT or SURF”, International Conference on Computer Vision, pp.2564–2571, 2011.
- [32] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast Retina Keypoint”, Conference on Computer Vision and Pattern Recognition, pp.510–517, 2012.
- [33] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, “Boosting binary keypoint descriptors”, Conference on Computer Vision and Pattern Recognition, pp.2874–2881, 2013.
- [34] K. Min, L. Yang, J. Wright, L. Wu, X. S. Hua, and Y. Ma, “Compact projection: Simple and efficient near neighbor search with practical memory requirements”, Conference on Computer Vision and Pattern Recognition, pp.3477–3484, 2010.
- [35] D. Achlioptas, “Database-friendly Random Projections: Johnson-Lindenstrauss with Binary Coins”, Journal of Computer and System Sciences, vol.66, no.4, pp.671–687, 2003.
- [36] M. M. Bronstein, C. Strecha, A. M. Bronstein, and P. Fua, “LDAHash: Improved Matching with Smaller Descriptors”, Pattern Analysis and Machine Intelligence, vol.34, pp.66–78, 2011.
- [37] M. Ambai, and Y. Yoshida, “Card: Compact and real-time descriptors”, International Conference on Computer Vision, pp.97–104, 2011.
- [38] J. M. Morel, and G. Yu, “ASIFT: A New Framework for Fully Affine Invariant Image Comparison”, SIAM Journal on Imaging Sciences, vol.2, no.2, pp.438–469, 2009.
- [39] Z. Wang, B. Fan, and F. Wu, “Affine Subspace Representation for Feature Description”, European Conference on Computer Vision, pp.94–108, 2014.
- [40] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm, “USAC: a universal framework for random sample consensus”, Pattern Analysis and Machine Intelligence, vol.35, no.8, pp.2022–2038, 2013.

- [41] J. Shi, and C. Tomasi, “Good Features to Track”, Conference on Computer Vision and Pattern Recognition, pp.593 - 600, 1994.
- [42] J. P. Gravel, “Corner Detection”, Biological Cybernetic, vol.59, no.4, pp.139 - 153, 1988.
- [43] T. Tuytelaars, and K. Mikolajczyk, Local Invariant Feature Detectors: A Survey, Now Publishers Inc., 2008.
- [44] 金澤靖, 金谷健一, “解説コンピュータビジョンのための画像の特徴点の抽出”, 電子情報通信学会誌, vol.87, no.12, pp.1043–1048, 2004.
- [45] E. Rosten, R. Porter, and T. Drummond, “FASTER and better: A machine learning approach to corner detection”, Pattern Analysis and Machine Intelligence, vol.32, pp.105–119, 2010.
- [46] J. R. Quinlan, “Induction of decision trees”, Machine Learning, vol.1, no.1, pp.81–106, 1986.
- [47] P. Perona, “Steerable-scalable kernels for edge detection and junction analysis”, European Conference on Computer Vision, pp.3–18, 1992.
- [48] D. Shy, and P. Perona, “X-Y separable pyramid steerable scalable kernels”, Conference on Computer Vision and Pattern Recognition, pp.237–244, 1994.
- [49] W. Freeman, and E. Adelson, “The design and use of steerable filters”, Pattern Analysis and Machine Intelligence, vol.13, pp.891–906, 1991.
- [50] K. Mikolajczyk, and C. Schmid, “A performance evaluation of local descriptors”, Pattern Analysis and Machine Intelligence, vol.27, no.10, pp.1615–1630, 2005.
- [51] R. Arandjelović, and A. Zisserman, “Three things everyone should know to improve object retrieval”, Conference on Computer Vision and Pattern Recognition, pp.2911–2918, 2012.
- [52] V. Balntas, L. Tang, and K. Mikolajczyk, “BOLD-Binary Online Learned Descriptor For Efficient Image Matching”, Conference on Computer Vision and Pattern Recognition, pp.2367–2375, 2015.
- [53] T. Trzcinski, and V. Lepetit, “Efficient discriminative projections for compact binary descriptors”, European Conference on Computer Vision, pp.228–242, 2012.
- [54] Y. Freund, and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, European Conference on Computational Learning Theory, pp.23–37, 1995.
- [55] S. Hinterstoisser, V. Lepetit, S. Benhimane, P. Fua, and N. Navab, “Learning real-time perspective patch rectification”, International Journal of Computer Vision, vol.91, no.1, pp.107–130, 2011.

- [56] S. Obdrzalek, and J. Matas, “Object Recognition Using Local Affine Frames on Maximally Stable Extremal Regions”, *Toward Category-Level Object Recognition*, pp.85-108, 2006.
- [57] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and Generic Corner Detection Based on the Accelerated Segment Test”, *European Conference on Computer Vision*, pp.183–196, 2010.
- [58] J. Cronje, “BFROST: binary features from robust orientation segment tests accelerated on the GPU”, *Annual Symposium of the Pattern Recognition Association of South Africa*, pp.25–30, 2011.
- [59] D. Mishkin, J. Matas, and M. Perdoch, “MODS: Fast and Robust Method for Two-View Matching”, *Computer Vision and Image Understanding*, vol.141, pp.81–93, 2015.
- [60] J. Heinly, E. Dunn, and J. M. Frahm, “Comparative Evaluation of Binary Features”, *European Conference on Computer Vision*, pp.759–773, 2012.
- [61] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, “HPatches: A benchmark and evaluation of handcrafted and learned local descriptors”, *Conference on Computer Vision and Pattern Recognition*, pp.5173–5182, 2017.
- [62] G. Griffin, A. Holub, and P. Perona, “Caltech-256 Object Category Dataset”, <https://authors.library.caltech.edu/7694/>, 2007.
- [63] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching”, *Conference on Computer Vision and Pattern Recognition*, pp.1–8, 2007.
- [64] P. Moreels, and P. Perona, “Evaluation of features detectors and descriptors based on 3d objects”, *International Journal of Computer Vision*, vol.73, no.3, pp.263–284, 2007.
- [65] P. Perona, “Deformable Kernels for Early Vision”, *Pattern Analysis and Machine Intelligence*, vol.17, pp.488–499, 1991.
- [66] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, “Learning Separable Filters”, *Pattern Analysis and Machine Intelligence*, pp.94-106, 2015.
- [67] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, “TILDE: A Temporally Invariant Learned DEtector”, *Conference on Computer Vision and Pattern Recognition*, pp.5279–5288, 2015.
- [68] K. Cordes, B. Rosenhahn, and J. Ostermann, “Increasing the Accuracy of Feature Evaluation Benchmarks Using Differential Evolution”, *Symposium on Differential Evolution*, pp.1–8, 2011.
- [69] M. Brown, G. Hua, and S. Winder, “Discriminative learning of local image descriptors”, *Pattern Analysis and Machine Intelligence*, vol.33, no.1, pp.43–57, 2011.

- [70] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, “Discriminative learning of deep convolutional feature point descriptors”, *International Conference on Computer Vision*, pp.118–126, 2015.
- [71] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks”, *British Machine Vision Conference*, p.3, 2016.
- [72] S. Zagoruyko, and N. Komodakis, “Learning to compare image patches via convolutional neural networks”, *Conference on Computer Vision and Pattern Recognition*, pp.4353–4361, 2015.
- [73] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, “Analysis and observations from the first amazon picking challenge”, *IEEE Transactions on Automation Science and Engineering*, 2016.
- [74] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, vol.86, no.11, pp.2278–2324, 1998.
- [75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, pp.1097–1105, 2012.
- [76] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.
- [77] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, *Conference on Computer Vision and Pattern Recognition*, pp.1–9, 2015.
- [78] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, *Conference on Computer Vision and Pattern Recognition*, pp.770–778, 2016.
- [79] R. D’Andrea, “Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead”, *Automation Science and Engineering*, vol.9, no.4, pp.638–639, 2012.
- [80] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, *Conference on Computer Vision and Pattern Recognition*, pp.580–587, 2014.
- [81] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition”, *International Journal of Computer Vision*, vol.104, no.2, pp.154–171, 2013.
- [82] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, “BING: Binarized normed gradients for objectness estimation at 300fps”, *Conference on Computer Vision and Pattern Recognition*, pp.3286–3293, 2014.



- [83] C. L. Zitnick, and P. Dollár, “Edge boxes: Locating object proposals from edges”, European Conference on Computer Vision, pp.391–405, 2014.
- [84] R. Girshick, “Fast r-cnn”, International Conference on Computer Vision, pp.1440–1448, 2015.
- [85] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks”, Advances in Neural Information Processing Systems, pp.91–99, 2015.
- [86] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, “Fast graspability evaluation on single depth maps for bin picking with general grippers”, International Conference on Robotics and Automation, pp.1997–2004, 2014.
- [87] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks”, International Conference on Artificial Intelligence and Statistics, pp.315–323, 2011.
- [88] S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, International Conference on Machine Learning, pp.448–456, 2015.

# 研究業績一覧

## 学術論文

- [1] 長谷川昂宏, 山内悠嗣, 安倍満, 吉田悠一, 山下隆義, 藤吉弘亘, “Cascaded FAST によるキーポイント検出”, 電子情報通信学会論文誌, vol. J98-D, no. 4, pp. 560–570, 2015.
- [2] 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, 秋月秀一, 橋本学, 堂前幸康, 川西亮輔, “Convolutional Neural Network による把持位置に基づいたマルチクラス物体認識”, 日本ロボット学会誌, 2018.

## 国際会議発表論文 (査読あり)

- [1] T. Hasegawa, R. Tomizawa, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, “Guided Filtering Using Reflected IR Image for Improving Quality of Depth Image”, Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 3, pp. 33–39, 2016.
- [2] T. Hasegawa, M. Ambai, K. Ishikawa, G. Koutaki, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, “Multiple-hypothesis Affine Region Estimation with Anisotropic LoG Filters”, International Conference on Computer Vision, pp. 585–593, 2015.
- [3] M. Kaneko, T. Hasegawa, Y. Yamauchi, T. Yamashita, H. Fujiyoshi, and H. Murase, “Fast 3D Edge Detection by Using Decision Tree from Depth Image”, International Conference on Intelligent Robots and Systems, pp. 1314–1319, 2015.
- [4] T. Hasegawa, Y. Yamauchi, M. Ambai, Y. Yoshida, and H. Fujiyoshi, “Keypoint Detection by Cascaded FAST”, International Conference on Image Processing, pp. 5611–5615, 2014.
- [5] H. Fujiyoshi, T. Yamashita, Y. Yamauchi, R. Murata, T. Hasegawa, M. Kaneko, Y. Murai, M. Hashimoto, S. Akizuki, M. Nagase, Y. Sakuramoto, S. Takei, S. Itoh, Y. Domae, R. Kawanishi, K. Shiratsuchi, R. Haraguchi, and M. Fujita, “Combined Point Cloud and Appearance-Based Object Detection for Grasping Rigid and Non-Rigid Objects”, International Workshop on Recovering 6D Object Pose at ICCV, 2015.

## 学会口頭発表(査読あり)

- [1] 真野航輔, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, 堂前幸康, 川西亮輔, 関真規人, “固有値テンプレート法による Fast Graspability Evaluation の高速化”, ロボット学会学術講演会, 2017.
- [2] 荒木諒介, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, 堂前幸康, 川西亮輔, 関真規人, “把持のしやすさを考慮した物体把持位置検出の高精度化”, 情報学ワークショップ, 2016.
- [3] 荒木諒介, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, 堂前幸康, 川西亮輔, 関真規人, “Graspability を導入した DCNN による物体把持位置検出”, 日本ロボット学会学術講演会, 2016.
- [4] 長谷川昂宏, Xuanyi Sheing, 荒木諒介, 山内悠嗣, 山下隆義, 藤吉弘亘, “Heterogeneous Learning によるオブジェクトネスと物体把持位置の検出”, 画像センシングシンポジウム, 2016.
- [5] 長谷川昂宏, 富沢凌二, 山内悠嗣, 山下隆義, 藤吉弘亘, “IR 反射強度画像を用いたガイデットフィルタによる距離画像の高品質化”, 画像センシングシンポジウム, 2015.
- [6] 金子将也, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, “決定木を用いた距離画像からの高速な三次元エッジ検出”, ロボティクスシンポジウム, pp. 417-423, 2015.
- [7] 金子将也, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, “決定木を用いた距離画像からの高速なエッジ検出”, 画像センシングシンポジウム, 2014.
- [8] 長谷川昂宏, 山内悠嗣, 安倍満, 吉田悠一, 藤吉弘亘, “Cascaded FAST によるキーポイント検出”, 画像センシングシンポジウム, 2013.

## 学会口頭発表(査読なし)

- [1] 河合康平, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, “特異値分解に基づくコンパクトなアフィン画像特徴記述”, 電気関係学会東海支部連合大会, 2017.
- [2] 長谷川昂宏, 安倍満, 上瀧剛, 山内悠嗣, 山下隆義, 藤吉弘亘, “アフィン変換特徴量記述子と下界算出に基づく距離計算によるキーポイントマッチング”, 画像の認識・理解シンポジウム, 2017.
- [3] 長谷川昂宏, 安倍満, 石川康太, 上瀧剛, 山内悠嗣, 山下隆義, 藤吉弘亘, “画像マッチングのための因子分解による局所特徴量表現”, 画像の認識・理解シンポジウム, 2016.
- [4] 安倍満, 長谷川昂宏, 藤吉弘亘, “対応点探索のための特徴量表現”, パターン認識・メディア理解研究会, vol. 115, no. 388, pp. 53-73, 2015.

- [5] 金子将也, 長谷川昂宏, 山内悠嗣, 山下隆義, 藤吉弘亘, “決定木を用いた距離画像からの多クラスエッジ検出”, 電気関係学会東海支部連合大会, 2014.
- [6] 長谷川昂宏, 山内悠嗣, 藤吉弘亘, “Cascaded FAST による高速なキーポイント検出”, 電気関係学会東海支部連合大会, 2013.
- [7] 長谷川昂宏, 安倍満, 山内悠嗣, 吉田悠一, 藤吉弘亘, “Cascaded FAST と CARD による高速な 2 画像間の対応付け”, 画像の認識・理解シンポジウム, 2013.

## 招待発表

- [1] 川西亮輔, 堂前幸康, 児島諒, 白土浩司, 原口林太郎, 秋月秀一, 橋本学, 長谷川昂宏, 藤吉弘亘, “Amazon Picking Challenge への挑戦”, 精密工学会 第 383 回講習会, 2016.
- [2] T. Hasegawa, M. Ambai, K. Ishikawa, G. Koutaki, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, “Multiple-hypothesis Affine Region Estimation with Anisotropic LoG Filters”, Meeting on Image Recognition and Understanding, 2016.

## 学術表彰

- [1] 2017 年 MIRU 学生奨励賞.  
題目: アフィン変換特徴量記述子と下界算出に基づく距離計算によるキーポイントマッチング
- [2] 2013 年 SSII オーディエンス賞.  
題目: Cascaded FAST によるキーポイント検出