

博士論文

高次元特徴空間の超低次元化に基づく物体認識

Object Recognition Based on Super Reduction
of High-dimensional Feature Space

国立大学法人 横浜国立大学
大学院環境情報学府

齊藤 航太
Kota SAITO

2015年3月

あらまし

近年、Twitter や Facebook などのソーシャルネットワーキングサービスの普及に伴い、我々のアクセスすることができる画像が急激に増加している。そこで、そのような大量の画像を用いることで、画像認識の性能を向上させる研究が盛んに行われている。画像認識では、画像から特徴量と呼ばれる色や輪郭、テクスチャなどを表現するベクトルを算出し、これらの特徴量を用いることで写っている物体の検出や画像がどのようなクラス（人物、自転車、イスなどといった物体のカテゴリ）であるのかの分類などを行う。従来から画像認識に有効な特徴量が数多く提案されており、大量の画像を用いた画像認識において、より特徴量が高次元になる（数千次元以上）ほど認識性能が高くなることが確認されている。しかしながら、特徴量が高次元になると特徴量の算出コスト、認識処理の計算コストが大きく増加してしまう、処理の解析が困難になってしまうといった課題が挙げられる。一方で、画像認識の応用の中でも医療現場における診断支援や工場の部品の疵検査などへの応用においては認識性能だけでなく、なぜそのような認識結果となったのかの説明を求められることも多い。このため、画像認識に求められることとして、“高い認識性能”、“低い計算コスト”、“認識処理のわかりやすさ”の3点が挙げられる。

そこで本研究では、2つのアプローチでこれらの実現を図る。1つ目のアプローチでは、認識対象の性質を用いること、および画像以外の特徴量を利用することで認識性能の向上を目指す。計算コストとわかりやすさについては、ヒトにとって直観的な条件判断を行うネットワークを認識器として用い、認識に有効な特徴量を選択することで低い計算コストとわかりやすさの両立を図る。本論文では、認識対象を空港などの巡回警備において撮影される人物とする。このため、歩行者のような立ち姿の人物に加えて、座っている人物や人や荷物などで一部が隠れてしまっているような人物も検出対象とする。本研究では、身体全体では姿勢によって大きく見た目が異なるが、部分的（手や足、頭など）には見た目の変化が少ないことを人物の性質として仮定する。そして、画像以外の特徴量としてカメラから物体までの距離情報を用いて、画像を人物の一部が切り出されるような過分割を行い、それぞれの領域が人物の候補領域であるかどうかの分類を行う。その後、人物の候補領域を統合することで姿勢変化や隠れに頑健な人物検出を行う。このアプローチによって、認識性能を向上しつつ計算コストと認識処理のわかりやすさを改善することができた。しかしながら、より一般的な画像認識を行う場合は、距離情報が必ずしもあるとは限らない。また、認識処理のわかりやすさを向上させる方法のひとつとして特徴量の可視化が挙げられるが、使用される特徴量が3次元より高くなると特徴量を可視化することができないという課題が残っている。

そこで2つ目のアプローチでは、特徴量を可視化することができる3次元以下へ低次元化（以下、超低次元化と呼ぶ）を行う。もともと特徴量に従来有効性が示されている画像特徴量を用いることで認識性能を保持しつつ超低次元化を行う。特徴量を超低次元化することで計算コストを抑えることができ、3次元以下の特徴空間を見ることで、データの分布や関係性などを解析することができる。また、特徴量の次元数を少なくすることで、決定木のように特徴量の大きさに条件判断を行うような認識器において処理数を少なくすることができ、処理フローの解析が容易になる。本研究では、特徴量の超低次元化手法として、データの自己組織化に基づく超低次元化手法と高次元特徴量の組み合わせによる超低次元化手法を提案する。

まず、データの自己組織化に基づく超低次元化手法を提案する。提案モデルは、球面上にデータをランダムに配置し、データ同士の類似度を引力・斥力として表現して球面上を移動させることで自己組織化を行う。自己組織化を用いることで与えられたデータから有効な特徴量を創発することを期待する。データの自己組織化に基づく超低次元化では、球面上にデータを写像するため、超低次元化された特徴量は球面上の3次元座標と緯度経度の2次元座標である。本論文では、複数のデータセットについて、データ分布の可視化実験を行い、もとの特徴量を用いた場合のクラス分類結果と超低次元化された特徴量を用いた分類結果を比較することで提案手法の有効性を検証する。

次に、高次元特徴量の組み合わせによる超低次元化手法を提案する。高次元の特徴量を用いたとしてもクラス内の多様性が高い画像では識別が困難である場合もあり、それらを識別することができるような新たな特徴量を既存特徴量を組み合わせることで構築し、超低次元化を行う。本研究では、木構造状の関数やプログラムを構築する遺伝的プログラミングのひとつである Cartesian Genetic Programming (CGP) を用いて階層的に特徴量を超低次元化する。第1層となる既存の高次元特徴量を、CGPによって最適化された関数を用いて変換することで新たな特徴量を構築する。このとき、急激に3次元以下にしてしまうと高次元特徴量の情報が失われすぎてしまう恐れがある。このため、階層的に徐々に次元数を落としていくことで最終的に3次元に超低次元化する。本論文ではクラス内の多様性が高い一般画像（自転車、人物）と医用画像（小腸画像）、およびクラス内の多様性が低い一般画像（歩行者）について対象クラスであるかどうかの2クラス分類実験を行い、特徴量の次元数を落としても分類精度を向上させることができることを示す。また、3次元に超低次元化した特徴量を用いてデータ分布を可視化するとともに、ヒトにとって直観的な条件判断を行う決定木を分類器に用い、処理フローの解析しやすさの考察も行う。さらに、使用された既存特徴量の数と、超低次元化した特徴量を用いた分類処理の高速化倍率の2つの観点から、特徴量算出コストと分類計算コストについて評価する。

Abstract

In recent years, we can access large-scale image sets through the Internet. Image recognition methods using large-scale images are expected to improve performance and such methods have been studied. In the image recognition methods, features are vectors calculated by using pixel values to represent the characteristics of colors, edges, textures and so on. The features are used in the image recognition to detect target objects and classify images into classes, which are the object categories for instance person, bike, chair, and so on. Many image features for the image recognition have been proposed in the previous studies. The dimensionality of the previously-proposed image features is mostly over several thousands. The higher the dimensionality of features, the higher the performance of the image recognition. However, the use of these high dimensional features involves high computational costs and the difficulty of analysis of the recognition processing. On the other hand, for applications to the diagnostic support in the medical front or the appearance check in factories, it is necessary that not only the high recognition performance but also the understandability. Thus, needs for the image recognition are “High recognition performance”, “Low computational costs” and “Understandability”.

In this study, we realize the needs by taking following two approaches. The first approach uses the characteristics of the target class and employs not only the image features but also the features other than the image features in order to preserve the high recognition performance. This approach employs the decision network easy to understand for humans to improve the understandability, and selects several features that are effective for recognition in order to reduce the computational costs. In this paper, the target class is “Person” in air ports: pedestrians, sitting people, occluded people and so on. Although the appearances of the whole body change variously, we suppose that the parts, for instance hands, legs, heads and so on, change relatively little. Thus, the proposed method segments depth images into small regions; these regions are then classified as human or nonhuman. The method then merges the regions classified as human into one region per person and outputs bounding rectangles for those regions. Experimental results show that this method is more effective than a previous method in detecting occluded people in various poses. Though the first approach improved the recognition performance, the computational costs and the understandability, there are still problems that the depth images are not always obtained and the selected features are not visualized in case that the dimensionality of selected features is over three.

Therefore, the second approach is to reduce the dimensionality of features to three dimensions or less to visualize the data landscapes. We define the dimensionality reduction to three dimensions or less as “Super Reduction (SR)”. This approach applies SR to high-dimensional feature space with preserving the high recognition performance by using high-dimensional features as input features. The features generated by SR can reduce the computational costs and visualize the data landscapes and relations. Moreover, the understandability of the process flow is improved by using the features generated by SR in case that the decision tree easy understand for humans is employed as the classifier because the number of process is reduced. In this paper, we propose a SR method based on self-organization of data and a SR method based on combination of high-dimensional features.

At first, we propose a SR method based on self-organization of data. The proposed method puts all data points on a spherical surface and each data point moves on the spherical surface under the forces of all the other data points. Since this method employs a spherical surface as the output space, the dimensionalities of the features generated by SR are three dimensions of coordinates in three dimensional space and two dimensions of coordinates of longitude and latitude. In this paper, we performed experiments on several datasets to validate the visualization results and the classification results by comparing results using original features with those using features generated by SR.

After that, we propose a SR method based on combination of high-dimensional features. In this paper, we propose a hierarchical feature dimension reduction method which is based on cartesian genetic programming (CGP). The proposed method generates a predefined number of new features one at a time using CGP per layer. A CGP generates a new feature by combining the previously-proposed image features. The proposed method finally generates three-dimensional features to visualize the data landscapes. We performed experiments on the Graz dataset and the capsule endoscopy images as image sets characterized by high intra-class variation, and the INRIA person dataset as an image set characterized by relatively low intra-class variation. The classification results obtained using the features generated by SR were better than those using the previously-proposed features with regard to the support vector machine, the decision tree and the k-nearest neighbor classifiers. The visualization results of the three-dimensional features generated by SR showed that the object images are broadly separated from the non-object images. The understandability of the process flow is improved by using the features generated by SR in case that the decision tree is employed as the classifier because the number of process is reduced. Moreover, the proposed method reduced the computational costs of the feature extraction and the classification by reducing the dimensionality of features.

目次

第1章	序論	1
1.1	背景と目的	1
1.2	本論文の構成	2
第2章	本研究に関する先行研究	3
2.1	進化計算法	3
2.1.1	遺伝的アルゴリズム (Genetic Algorithm; GA)	3
2.1.2	遺伝的プログラミング (Genetic Programming; GP)	5
2.1.3	Evolutinary Decision Network (EDEN)	6
2.1.4	Cartesian Genetic Programming (CGP)	8
2.2	画像特徴量	9
2.2.1	局所特徴量	9
2.2.2	大域特徴量	10
2.3	データ分布に基づく特徴量の低次元化に関する従来研究	12
2.3.1	主成分分析 (Principal Components Analysis; PCA)	12
2.3.2	自己組織化マップ (Self-Organizing Maps; SOM)	14
2.3.3	多次元尺度構成法 (Multi-Dimensional Scaling; MDS)	15
2.4	既存特徴量を用いた進化計算法による特徴量の低次元化	16
2.4.1	GAを用いた特徴量選択	16
2.4.2	GPを用いた既存特徴量の組み合わせによる特徴量自動構築	17
2.5	まとめ	18
第3章	人物の部分領域に着目した姿勢変化や隠れに頑健な人物検出	19
3.1	はじめに	19
3.2	人物の部分領域に着目した特徴量選択と条件判断に基づく人物検出	19
3.2.1	提案手法の概要	19
3.2.2	Step 1. 距離情報を用いた領域分割	20
3.2.3	Step 2. 過分割された領域の人物・非人物分類	22
3.2.4	Step 3. 人物と分類された領域の統合	24
3.3	人物検出実験	28
3.3.1	実験設定	28
3.3.2	提案手法の各 Step の結果に対する考察	35
3.3.3	提案手法および比較手法の人物検出結果とその考察	40
3.4	まとめ	47

第 4 章	データの自己組織化に基づく特徴量の超低次元化	48
4.1	はじめに	48
4.2	球面上の移動に基づくデータの自己組織化	48
4.2.1	提案モデルの概要	48
4.2.2	データ間の類似度に基づく球面上の移動	49
4.3	特徴量の超低次元化による可視化実験およびクラス分類実験	50
4.3.1	実験概要	50
4.3.2	実験データ	50
4.3.3	実験設定	50
4.3.4	可視化結果の評価指標	51
4.3.5	可視化結果と考察	52
4.3.6	提案した評価指標による評価と考察	53
4.3.7	クラス分類による評価と考察	55
4.4	まとめ	56
第 5 章	既存特徴量の組み合わせによる特徴量の階層的な超低次元化	58
5.1	はじめに	58
5.2	CGP を用いた特徴量の階層的な超低次元化	58
5.3	特徴量の超低次元化および画像分類実験	59
5.3.1	使用した画像セット	59
5.3.2	使用した既存の画像特徴量	61
5.3.3	CGP の設定	62
5.3.4	分類性能による評価	62
5.3.5	処理の解析しやすさによる評価	67
5.3.6	計算コストによる評価	68
5.4	まとめ	69
第 6 章	結論	75
6.1	本論文で得られた成果	75
6.2	今後の課題	76
	謝辞	77
	参考文献	78
	研究業績リスト	82
付録 A	第 4 章における UCI データセットの可視化結果	84
付録 B	第 5 章において最適化された特徴量を組み合わせる関数の例	89

目次

2.1	GA の処理の流れ	4
2.2	GP の木構造を数式とした場合の例	5
2.3	EDEN の構造の例 (表現型)	6
2.4	EDEN の構造の例 (遺伝子型)	7
2.5	EDEN における処理の変更がない場合の例	8
2.6	EDEN における処理の変更がある場合の例	8
2.7	CGP の表現型と遺伝子型の例	9
2.8	SIFT 特徴量の算出方法の概要	9
2.9	LBP および ILBP の概要	11
2.10	BoVW の概要	11
2.11	HOG の概要	12
2.12	PCA の概念図	13
2.13	SOM の概要	14
2.14	MDS の概念図	15
2.15	特徴量選択およびパラメータ最適化に用いる GA の遺伝子型の例	16
2.16	GP を用いた既存特徴量の組み合わせによる特徴量の自動構築結果の例	17
3.1	提案手法の処理の流れ	20
3.2	Step 1 の処理の流れ	21
3.3	距離センサの設置方向と用いる座標系	22
3.4	統合を行う候補の領域の選択範囲	25
3.5	領域の統合による検出ウィンドウの算出の例	25
3.6	対象とした環境の外観	29
3.7	取得したカラー画像の例	29
3.8	正解データの検出ウィンドウの例	29
3.9	Kinect で取得したカラー画像と距離画像の例	31
3.10	各領域の正解クラスの算出方法	32
3.11	考察を行うフレームのカラー画像	36
3.12	距離情報を用いた領域分割結果の例	36
3.13	過分割された領域の人物・非人物分類結果の例	37
3.14	獲得した人物・非人物分類ネットワーク	38
3.15	人物と分類された領域の統合による検出ウィンドウの出力例	39
3.16	獲得した領域の統合の判断を行うネットワーク	40
3.17	検出率・誤検出率による性能比較	42
3.18	比較手法において 1.2÷1.4[m] のウィンドウを用いた検出結果の例	43
3.19	テストデータにおける人物の姿勢・隠れごとの検出率・誤検出率による検出性能	44

3.20	学習データにおける提案手法および比較手法の検出ウィンドウの出力例	45
3.21	テストデータにおける提案手法および比較手法の検出ウィンドウの出力例	46
4.1	提案モデルの概要	49
4.2	各手法の Iris の可視化結果	53
4.3	各手法の Wdbc の可視化結果	54
4.4	SVM におけるクラス分類結果の比較	56
4.5	C4.5 におけるクラス分類結果の比較	57
4.6	k 近傍法におけるクラス分類結果の比較	57
5.1	提案モデルの概要	59
5.2	Graz-01 データセットの画像例	60
5.3	カプセル内視鏡画像の例 (出典: 日比紀文, 『カプセル内視鏡カラーアトラス』, 診断と治療社, 2007, p.10,26)	61
5.4	歩行者画像の例	61
5.5	歩行者画像における ROC 曲線	66
5.6	構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (bikes クラス)	70
5.7	構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (persons クラス)	71
5.8	構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (カプセル内視鏡画像)	72
5.9	構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (歩行者画像)	73
5.10	3 次元空間における bikes クラスの画像の例	74
A.1	UCI データセットの可視化結果 (Bupa, Pima)	85
A.2	UCI データセットの可視化結果 (Ionosphere, Wine)	86
A.3	UCI データセットの可視化結果 (Cleveland, Glass)	87
A.4	UCI データセットの可視化結果 (Ecoli, Yeast)	88
B.1	第 2 層において最適化された関数の例	90
B.2	第 3 層において最適化された関数の例	91
B.3	第 4 層において最適化された関数の例	92

表目次

3.1	色空間および距離画像の特徴量に用いる統計量	23
3.2	点群データの特徴量	23
3.3	領域の形状の特徴量	23
3.4	学習およびテストデータの人物の内訳	28
3.5	Step 1 における各種パラメータ設定	31
3.6	人物・非人物分類を行う EDEN _{s2} の設定	32
3.7	Step 3 の統合とウィンドウ算出の各種パラメータ設定	33
3.8	領域の統合の判断に用いる EDEN _{s3} の設定	33
3.9	学習における算出ウィンドウの統合のパラメータ設定	34
3.10	各サイズのウィンドウのブロック分割における分割数	34
3.11	各サイズのウィンドウの学習データの数	35
3.12	比較手法の各種パラメータ設定	35
4.1	使用した UCI データセット	51
4.2	SOM の設定	51
4.3	提案した可視化結果の評価指標による比較結果	55
5.1	カプセル内視鏡画像分類における層数および次元数	60
5.2	CGP の設定	63
5.3	bikes クラスの分類結果	64
5.4	persons クラスの分類結果	65
5.5	カプセル内視鏡画像の分類結果	65
5.6	主成分分析との比較	67
5.7	画像セットごとの各次元数における決定木の大きさ	68
5.8	テストデータの分類における WEKA の実行時間に関する高速化倍率	68

第1章 序論

1.1 背景と目的

近年、Twitter や Facebook などのソーシャルネットワーキングサービスの普及に伴い、我々のアクセスすることができる画像が急激に増加している。そこで、そのような大量の画像を用いることで、画像認識の性能を向上させる研究が盛んに行われている。画像認識では、画像から特徴量と呼ばれる色や輪郭、テクスチャなどを表現するベクトルを算出し、これらの特徴量を用いることで写っている物体の検出や画像がどのようなクラス（人物、自転車、イスなどといった物体のカテゴリ）であるのかの分類などを行う。従来から画像認識に有効な特徴量が数多く提案されており、大量の画像を用いた画像認識において、より特徴量が高次元になる（数千次元以上）ほど認識性能が高くなることが確認されている。しかしながら、特徴量が高次元になると特徴量の算出コスト、認識処理の計算コストが大きく増加してしまう、処理の解析が困難になってしまうといった課題が挙げられる。一方で、画像認識の応用の中でも医療現場における診断支援や工場の部品の疵検査などへの応用においては認識性能だけでなく、なぜそのような認識結果となったのかの説明が求められることも多い。このため、画像認識に求められることとして、“高い認識性能”、“低い計算コスト”、“認識処理のわかりやすさ”の3点が挙げられる。

そこで本研究では、2つのアプローチでこれらの実現を図る。1つ目のアプローチでは、認識対象の性質を用いること、および画像以外の特徴量である距離情報を利用することで認識性能の向上を目指す。計算コストとわかりやすさについては、ヒトにとって直観的な条件判断を行うネットワークを認識器として使い、認識に有効な特徴量を選択することで低い計算コストとわかりやすさの両立を図る。しかしながら、より一般的な画像認識を行う場合は、距離情報が必ずしもあるとは限らない。また、認識処理のわかりやすさを向上させる方法のひとつとして特徴量の可視化が挙げられるが、使用される特徴量が3次元より高くなると特徴量を可視化することができないという課題が残っている。

そこで2つ目のアプローチでは、特徴量を可視化することができる3次元以下へ低次元化（以下、超低次元化と呼ぶ）を行う。もともとなる特徴量に従来有効性が示されている画像特徴量を用いることで認識性能を保持しつつ超低次元化を行う。特徴量を超低次元化することで計算コストを抑えることができ、3次元以下の特徴空間を見ることで、データの分布や関係性などを解析することができる。また、特徴量の次元数を少なくすることで、決定木のように特徴量の大きさで条件判断を行うような認識器において処理数を少なくすることができ、処理フローの解析が容易になる。本研究では、特徴量の超低次元化手法として、データの自己組織化に基づく超低次元化手法と高次元特徴量の組み合わせによる超低次元化手法を提案する。

まず、データの自己組織化に基づく超低次元化手法を提案する。提案モデルは、球面上にデータをランダムに配置し、データ同士の類似度を引力・斥力として表現して球面上を移動させることで自己組織化を行う。自己組織化を用いることで与えられたデータから有効な特徴量を創発することを期待する。

次に、高次元特徴量の組み合わせによる超低次元化手法を提案する。高次元の特徴量を用いた

としてもクラス内の多様性が高い画像では識別が困難である場合もあり、それらを識別することができるような新たな特徴量を既存特徴量を組み合わせることで構築し、超低次元化を行う。このとき、急激に3次元以下にしてしまうと高次元特徴量の情報が失われすぎてしまう恐れがある。このため、階層的に徐々に次元数を落としていくことで最終的に3次元に超低次元化する。

1.2 本論文の構成

本論文の構成は次の通りである。まず、第2章で本研究に関連する先行研究について述べる。次に第3章では、本研究の1つ目のアプローチである、特徴量選択と条件判断による人物検出手法を提案し、姿勢変化や隠れが存在するような人物の検出実験でその有効性を示す。そして、第4章および第5章で本研究の2つ目のアプローチである特徴量の超低次元化について述べる。第4章では、データの自己組織化に基づく特徴量の超低次元化手法を提案し、クラス分類問題のデータセットに適用することでその有効性を示す。第5章では、高次元特徴量の組み合わせによる特徴量の超低次元化手法を提案し、クラス内の多様性の高い画像セットおよびクラス内の多様性の低い画像セットそれぞれのクラス分類問題に適用することでその有効性を示す。また、低次元化による計算コストの低減の評価および、構築した3次元以下の特徴量を用いたデータ分布の可視化と処理フローの解析を行うことでわかりやすさについても評価を行う。最後に、第6章で本論文のまとめと今後の課題について述べる。

第2章 本研究に関する先行研究

本章では、本研究と関連の深い進化計算法、画像特徴量、データ分布に基づく特徴量の低次元化に関する先行研究、および進化計算法に基づく既存特徴量を用いた特徴量の低次元化に関する先行研究について述べる。

2.1 進化計算法

進化計算法 (Evolutionary Computation; EC) は生物の進化から着想を得た探索・最適化アルゴリズムである。EC の代表的な手法として遺伝的アルゴリズム (Genetic Algorithm; GA) [1-4]、遺伝的プログラミング (Genetic Programming; GP) [5, 6] が挙げられる。本節では、GA と GP および本研究に関連の深い進化計算法の先行研究について述べる。

2.1.1 遺伝的アルゴリズム (Genetic Algorithm; GA)

GA は J.H.Holland によって提案された探索・最適化アルゴリズムである。生物の進化から着想を得たアルゴリズムであり、生物の集団における自然淘汰、交叉、突然変異を模倣した操作を各世代において施すことによって最適解を探索する。GA は関数最小化問題や組み合わせ最適化問題など、さまざまな問題に幅広く適用され、その有効性が示されている。

GA では探索空間中の 1 つの探索点を個体と呼び、個体の性質は遺伝子型 (染色体の集合) によって記述される。遺伝子型は通常、0 または 1 からなる遺伝子のビット列によって表現される。GA の基本的な処理の流れを図 2.1 に示す。

(a) 初期個体の生成

まず、N 個の初期個体集団を生成する。初期個体における各遺伝子のビットは一般的にはランダムに決定される。

(b) 初期個体の適応度の評価

生成された初期個体をあらかじめ設定した適応度関数で評価し、適応度を算出する。適応度は問題に対する解の適合度を表すものであり、適応度関数の設定は GA を用いる上で重要である。

(c) 次の世代の個体の決定

個体集団から、適応度に基づいて劣った個体を淘汰して新しく適応度の高い個体を補う操作を行う。一般的には、個体の選択は重複を許すため、同じ個体が複数回選択される可能性もある。適応度の高い個体を選択されやすくすることで、次の世代に残ることができる確率が高くなるような選択をすることが一般的である。代表的な選択方法のひとつであるルーレット選択では、各個体の選択される確率をその個体の適応度に比例させて設定することで、適応度の高い個体を選択しやすくしている。また、ランダムに選択された個体から最大適応度

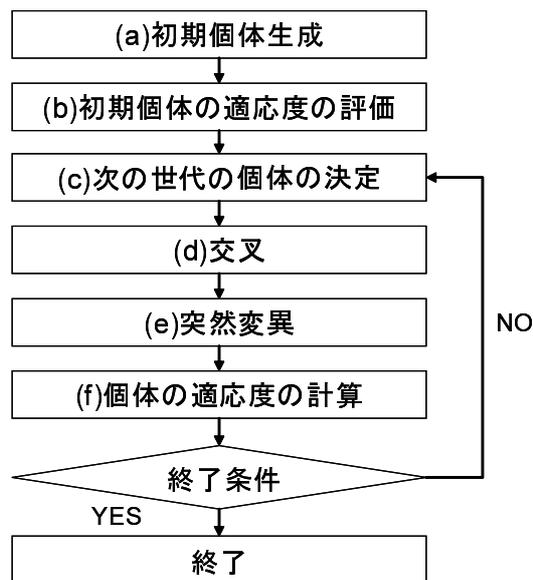


図 2.1: GA の処理の流れ

の個体を選択するトーナメント選択も代表的な選択方法のひとつである。これまでに GA はさまざまな改良がされてきており、淘汰規則の拡張として、その世代の最良個体を次の世代に必ず残すエリート保存戦略も提案されている。エリート保存戦略を適用することで、最大適応度の個体を確実に次の世代に残すことができる。

(d) 交叉

自然淘汰だけを繰り返し適用した場合、個体集団における最良個体の占める割合が高くなり、最終的には最良個体だけとなってしまい、それより適応度の高い個体は得られない。このため、適応度の高い個体を尊重し、それと似た新たな個体を生成する必要がある。これを行う操作が交叉である。まず、個体集団の中から 2 個体をランダムに取り出し、ペアを作る。次に、交叉率と呼ばれる確率に基づいてこの 2 個体の遺伝子列を交叉するかどうかを決定する。交叉することが決定した場合、この 2 個体の遺伝子列を組み替えて新たな 2 個体を生成し、もとの 2 個体と置き替える。遺伝子列を組み替える方法には、染色体の遺伝子列をランダムに決定した点で区切りペアで遺伝子列を交換する 1 点交叉や、区切られた区間の遺伝子列を交換する多点交叉、個体の各遺伝子をランダムに交換する一様交叉などいくつかの交叉方法が提案されている。

(e) 突然変異

全個体の染色体の全遺伝子に対して、突然変異率と呼ばれる非常に低い確率に基づいて突然変異を行うかどうかを決定する。突然変異を行うことが決定した場合、その遺伝子をランダムに変更する。遺伝子が 0 または 1 のビットである場合、突然変異ではビットを反転させる操作が行われる。

(f) 個体の適応度の評価

(c) ~ (e) の操作が行われた後の個体集団の全個体の適応度を求め、あらかじめ設定した終了条件を満たしていれば個体集団の世代交代を終了し、問題に対する最終的な解を決定する。終了条件は、世代交代は最大世代数や適応度の最大値に達することや、最良個体が一定

世代数変化しなかったこととするのが一般的な方法がある。条件を満たしていなければ、さらに世代交代を行い、(c) ~ (e) を繰り返す。

これが遺伝的アルゴリズムの基本的な処理の流れであるが、本研究では Minimal Generation Gap (MGG) [7] と呼ばれる世代交代モデルを用いる。MGG による世代交代ではまず、個体集団の中から親個体となる 2 つの個体を重複を許さずにランダムに選択する。次に、親個体同士をあらかじめ設定した子個体生成数だけ交叉、突然変異を施し、子個体を生成する。親個体と子個体を合わせた個体集団の中から 2 個体を選択してもとの個体集団に戻す。このとき 2 個体は、適応度の最も高いエリート 1 個体と、ルーレット選択によって選択された 1 個体とするのが一般的である。

2.1.2 遺伝的プログラミング (Genetic Programming; GP)

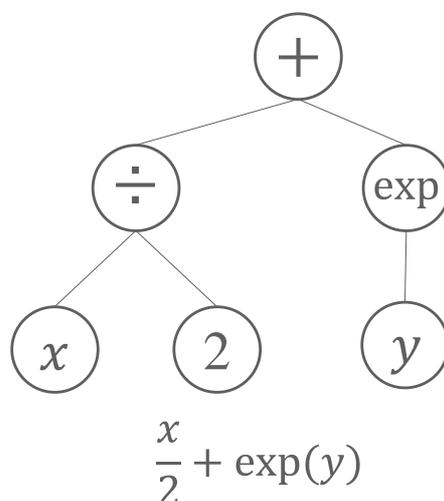


図 2.2: GP の木構造を数式とした場合の例

GP は GA を木構造に拡張した手法である。GP では個体を木構造で表現することで、数式やプログラムを生成することができる。図 2.2 に GP の木構造を数式とした場合の例を示す。GP の処理の流れは基本的に GA と同じである。交叉、突然変異は遺伝子型が異なるため、GA とは異なる操作を施す。一般的な交叉方法として、部分木交換が挙げられる。部分木交換では、2 つの個体でそれぞれランダムに選択した部分木を交換する。また、突然変異では突然変異率に従ってノードをランダムに変更するのが一般的である。GP では、これらの遺伝操作を施すうちに木構造が巨大になるブロートが発生するという問題を抱えている。ブロートが発生すると木が冗長になる、解の探索が非効率になる、計算機のメモリ使用量の大幅な増加などといった問題が発生する。ブロートを防ぐ一般的な方法として、木の深さやノード数を制限し、それらを超えた個体を致死遺伝子とするという方法が挙げられる。しかし、このような単純な制限をする場合、個体の多様性が失われ、効率的な探索が行うことができなくなる恐れがある。そこで、木の大きさに対するペナルティを与えた適応度関数 [8] や、交叉および突然変異の遺伝操作の改良 [9, 10] など、さまざまな手法が提案されている。また、2.1.4 項で述べる GP の手法のひとつである Cartesian Genetic Programming (CGP) [11] では、木構造を構成するノード数をあらかじめ設定し、すべてのノードを遺伝子型に組み込み、その接続関係を最適化することでブロートの問題に対応している。

2.1.3 Evolutionary Decision Network (EDEN)

本項では、第3章においてヒトにとって直観的な条件判断を行う分類器として用いる進化的条件判断ネットワーク (Evolutionary Decision Network; EDEN) [12-14] について述べる。EDEN は、決定木 [15] を拡張した手法であり、入力されたデータの特徴量ベクトルを用いて条件分岐を行い、そのデータのクラス分類を行う手法である。EDEN では分類対象のデータ以外に他のデータを参照入力として入力することができ、これによって処理の変更をすることができる。例えば、画像の領域の分類問題においては、ある領域を分類対象としたときに周囲の領域を参照入力とすることで、周囲の領域の情報を考慮した分類を行うことができる。

EDEN の構造

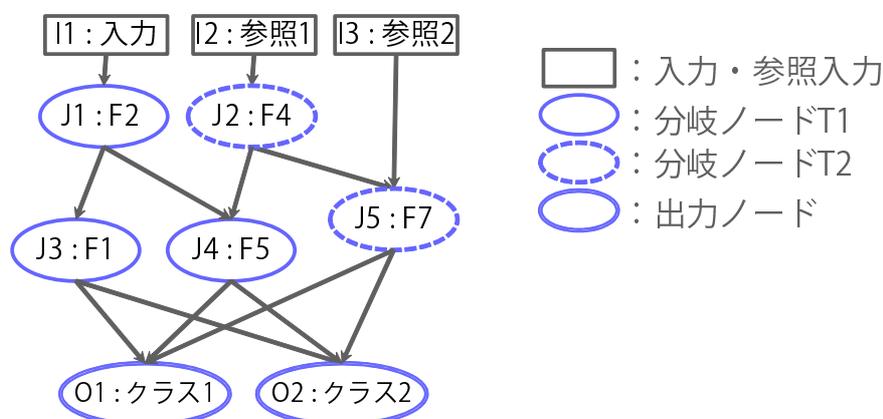


図 2.3: EDEN の構造の例 (表現型)

EDEN の構造の例を図 2.3 に示す。EDEN は表現型にフィードフォワード型のネットワーク構造を採用している。EDEN で用いるノードは次の 3 種類に分けられる。

1. 入力ノード

入力ノードは、入力するデータである特徴量ベクトルに対応しており、分類対象の入力データに加え、処理の変更用いる参照入力データを入力することができる。

2. 中間ノード

中間ノードではデータの特徴量を参照し、しきい値による分岐など、ノードごとの判定内容に応じた分岐を行う。入力データは各中間ノードによって決められた分岐先のノードをたどり、最終的に出力ノードに到達する。

3. 出力ノード

出力ノードは、分類するそれぞれのクラスに対応している。入力データは中間ノードをたどって最終的に 1 つの出力ノードに到達するため、到達した出力ノードに対応したクラスに分類されたことになる。

EDEN では、これらのノードの種類と用いる特徴量、分岐先、および判定内容のパラメータを GA を用いて最適化する。

遺伝子構造と遺伝操作

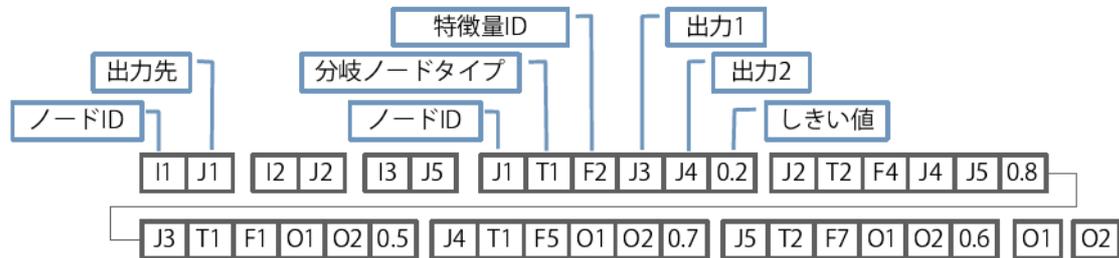


図 2.4: EDEN の構造の例 (遺伝子型)

EDEN の各ノードでは単一の特徴量の大きさによる分岐を行うため、分岐に用いる特徴量とその分岐の基準、およびそれらの組み合わせを最適化することで目的の分類を行う構造を獲得する。EDEN では GA によって構造の最適化を行うため、最適化の際には表現型のネットワークを遺伝子型に変換して遺伝操作を行う。EDEN の遺伝子型は 1 次元の整数および実数の配列で表されており、遺伝子型から表現型のネットワークへ変換することで分類に適用する。表現型である図 2.3 に対応する遺伝子型の一部を図 2.4 に示す。染色体はネットワークを構成する各ノードに対応しており、ノードの種類と用いる特徴量、分岐先、および判定内容のパラメータを保持している。ここで、各ノードの判定内容はノードの種類と用いる特徴量で構成されており、図 2.4 のにおけるノード ID、特徴量 ID にそれぞれ対応している。また、各ノードにはあらかじめノード ID が付けられているものとし、入力ノード、中間ノード、出力ノードの順に割り当てられる。このとき入力ノードと出力ノードの数は決められているが、使用される中間ノードの数は最適化によって決定される。このため、中間ノードは最大数だけ用意しておき、最適化の結果得られるノードの接続の状態によって、使用されるノードと使用されないノードが現れることで、表現型での中間ノードの数が最適化される。なお、分岐先の遺伝操作を行う際はフィードフォワード構造を守るため、分岐先として選択することができるノードを自分の番号より大きい番号に制限している。

本論文では EDEN の世代交代モデルに進化戦略の (1+4) ES を採用し、遺伝操作には突然変異だけを用いている。突然変異の操作は各遺伝子に対して行われ、突然変異率 P_m の確率でノードの種類と用いる特徴量、分岐先、および判定内容のパラメータを一様乱数で変更する。EDEN の世代交代は次のように行われる。

- Step 1.** 世代数を $j = 1$ とする。初期個体を乱数を用いて 1 つ生成し、これを親個体 M として適応度を計算する。
- Step 2.** M に突然変異の操作を施して子個体を 4 つ生成し、その子個体集団を C とし、すべての子個体の適応度を計算する。
- Step 3.** 親個体および子個体の集合 $M + C$ から適応度が最も高い個体を選び、 M と置き換える。ここで、親個体と子個体の適応度が等しい場合は子個体を選択し、 M と置き換える。
- Step 4.** 終了条件を満たす場合は終了する。そうでなければ $j, j + 1$ として Step 2 に戻る。

多入力による処理の変更

EDEN では、多入力によって異なる処理を行うことも可能としている。分類対象の入力データとは別に参照入力データとして、分類対象に関連するデータを別の入力ノードから入力する。処理の変更がない場合の例を図 2.5 に、処理の変更がある場合の例を図 2.6 にそれぞれ示す。はじめに参照入力データをネットワークに入力し、その参照入力データが通過したノードを発火状態にする。そして、すべての参照入力データを遷移させた後、分類対象を入力して分類を行う。分類対象の各ノードの分岐において、発火状態のノードは発火していない状態とは異なる処理が行われる。こうすることで、参照入力データを考慮した分類を行うことができる。処理の変更の例として、各ノードの分岐のしきい値の変更が挙げられる。これは、各ノードの分岐のしきい値に加えて、発火した場合に用いる分岐のしきい値を遺伝子型に組み込むことによって、発火の状態によって用いるしきい値を選択することで実現することができる。

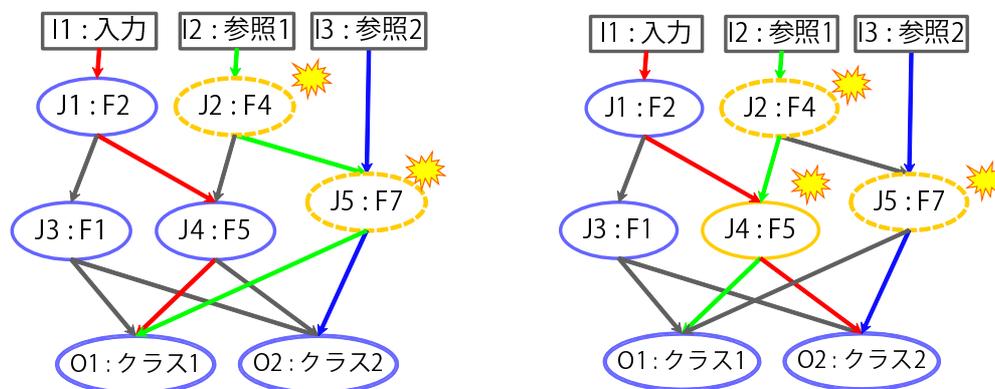


図 2.5: EDEN における処理の変更がない場合の例 図 2.6: EDEN における処理の変更がある場合の例

2.1.4 Cartesian Genetic Programming (CGP)

本項では、第 5 章における高次元特徴量の組み合わせに基づく特徴量の超低次元化手法において用いる CGP について述べる。CGP は GP の手法のひとつであり、あらかじめすべてのノードを遺伝子型に記述しておき、その接続関係を最適化する手法である。こうすることで、プロトの問題に対応し、部分木を再利用することができる構造とすることで、木構造をコンパクトに表現することができる。

図 2.7 に CGP の表現型と遺伝子型の例を示す。本論文では CGP の遺伝子型に 1 次元の整数列を用いる。1 つの染色体は 1 つのノードを表しており、入力ノード、中間ノード、出力ノードの 3 種類のノードで構成される。入力ノードは“ノードタイプ”、“特徴量タイプ”の 2 つの遺伝子で構成され、中間ノードと出力ノードでは、“ノードタイプ”、“演算子タイプ”、“入力 1”、“入力 2”の 4 つの遺伝子で構成される。これらはすべて整数で置き換えられ、それらの整数を用いて最適化される。図 2.7 の例では $O1 = \sin(f9) + (f4 + f16)(f16 - f9)$ と $O2 = 0.1(f4 + f16)(f16 - f9)$ の 2 つの数式を表す。

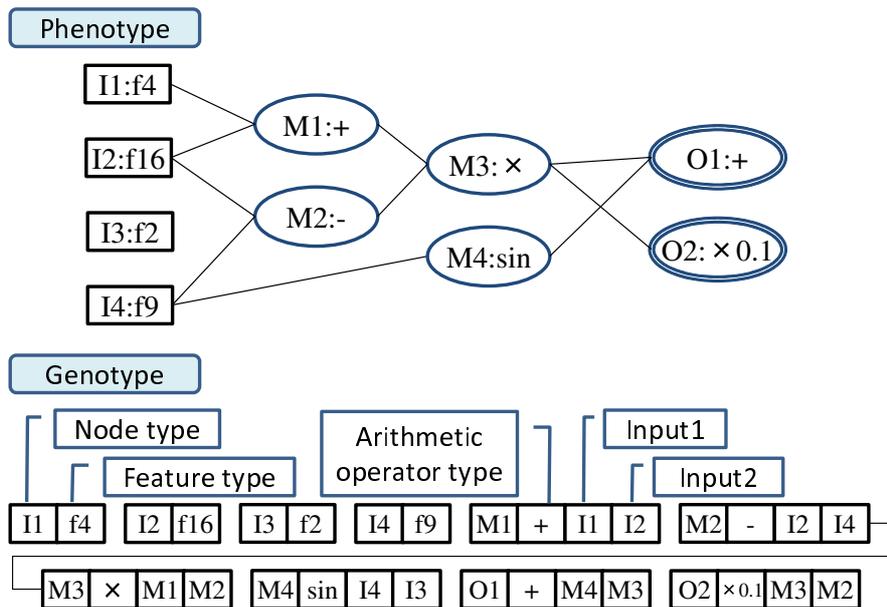


図 2.7: CGP の表現型と遺伝子型の例

2.2 画像特徴量

本節では、従来提案されている画像特徴量について述べる。画像特徴量は、局所特徴量と大域特徴量の2つに大別される。局所特徴量は画像の一部を表す特徴量であり、ある物体が画像の一部に映っている場合や、物体の一部が特徴的であるような画像から物体検出を行う場合に有効である。一方で、大域特徴量は画像全体を表す特徴量であり、画像全体の色やテクスチャの分布や構図などを記述するため、シーン分類や類似画像検索などに有効である。これらの特徴量は、局所特徴量で数百次元、大域特徴量においては数千次元以上の高次元特徴量である。膨大な数の画像を用いた画像認識の研究領域において、このような高次元な特徴量を用いることで、より特徴量の次元数が多くなるほど認識精度が高くなること確認されている [16]。

2.2.1 局所特徴量

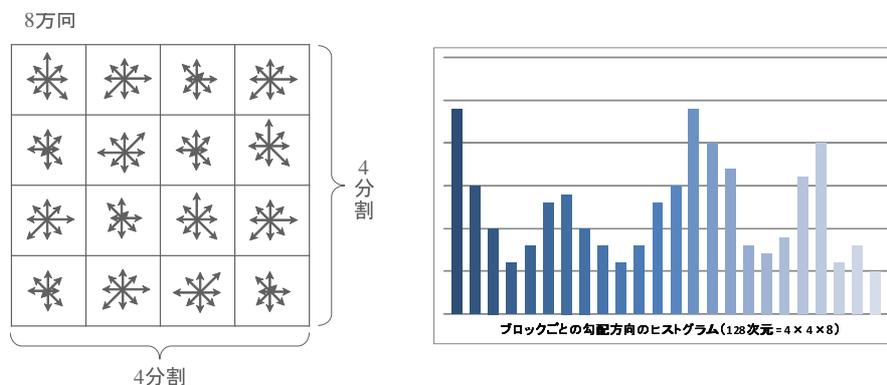


図 2.8: SIFT 特徴量の算出方法の概要

局所特徴量には、色ヒストグラムやエッジ強度など、色やエッジに基づく特徴量がよく用いられる。それらの中でも、Scale Invariant Feature Transform (SIFT) [17]は、画像のスケール変化や回転に頑健な特徴量として有効性が確認されている。SIFTではまず、画像における特徴点の位置(キーポイント)および大きさ(スケール)算出する。キーポイントおよびスケールの算出では、Difference-of-Gaussianの処理を用いて極値探索を行うのが一般的である。そして算出された特徴点の勾配方向ヒストグラムを算出する。このとき、量子化された勾配方向のうち、最大の強度をもつ方向をそろえるように回転させることで回転に頑健な特徴量となっている。また、特徴点を算出する際に、考慮する範囲を表すスケールも同時に算出することで、スケール変化にも頑健である。このため、SIFTは特徴点のマッチングにおいて有効性を示している。本論文では、後述する大域特徴量で用いる局所特徴量としてSIFTを用いるため、本項では特に局所特徴量の算出部分について述べる。図2.8にSIFTの算出方法の概要を示す。SIFTでは、一般的に画像のある大きさのセルを4×4分割し、分割したブロックの8方向(22.5°ごと)の勾配方向のヒストグラムを算出する。このため、SIFTは128次元(=4×4×8)である。勾配方向のヒストグラムを算出する際の、勾配方向 $\theta(x, y)$ および勾配強度 $m(x, y)$ は式(2.1)および式(2.2)で表される。

$$\theta(x, y) = \arctan \frac{f_h(x, y)}{f_v(x, y)}, \quad (2.1)$$

$$m(x, y) = \sqrt{f_v(x, y)^2 + f_h(x, y)^2}, \quad (2.2)$$

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

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

ただし、 $L(x, y)$ は画像の輝度値である。このようにSIFTは、局所的な領域の勾配方向のヒストグラムの分布を表現する特徴量である。

また、局所的なテクスチャのパターンを記述する特徴量としてLocal Binary Pattern (LBP) [18]が提案され、有効性が示されている。図2.9(a)に示すように、LBPは3×3 pixelのブロックを用いた局所特徴量である。図2.9(a)における数値は輝度値を表している。LBPは中央の画素の輝度値を基準として、左上の画素から時計回りに輝度値を比較し、中央の輝度値より低ければ0、そうでなければ1として8ビットのビット列を生成する。これを画像全体もしくは領域の画素で算出し、10進数で表したもののヒストグラムがLBPである。8ビットのビット列であるため、LBPは256次元である。LBPをよりエッジを強調するように改良した手法としてImproved LBP (ILBP) [19]が挙げられる。LBPでは中央の画素の輝度値を周囲の画素と比較する基準に用いているが、ILBPでは図2.9(b)に示すように3×3 pixelのブロックの平均値を周囲の画素と比較する基準に用いる。こうすることで、中央の画素の輝度値が周囲の画素の輝度値と大きく異なった場合でも、良好にエッジを表現することができる。

2.2.2 大域特徴量

大域特徴量では、自然言語処理におけるBag-of-Wordsに基づくBag-of-Visual Words (BoVW) [20, 21]が提案されている。BoVWの概要を図2.10に示す。BoVWでは、色やエッジの局所特徴量がVisual Words (VW)として用いられ、それらが画像全体でどのくらいの頻度で出現するかを特徴量としている。VWとして用いられる局所特徴量として一般的なものはSIFT, LBP, 色ヒストグラムなどである。

BoVWの他に、大域特徴量としてGIST [22]が提案されている。GISTは画像の構図を表す特徴量であり、道路のように奥行きのある画像やオフィス内を写した画像などのように、写っている環

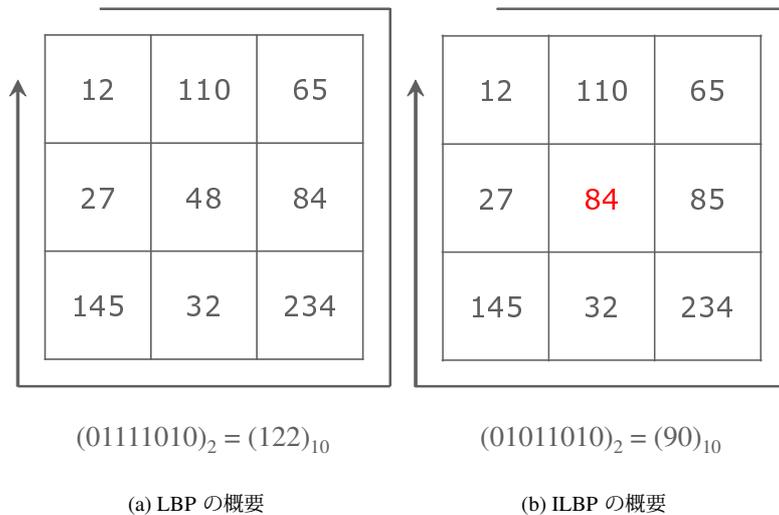


図 2.9: LBP および ILBP の概要

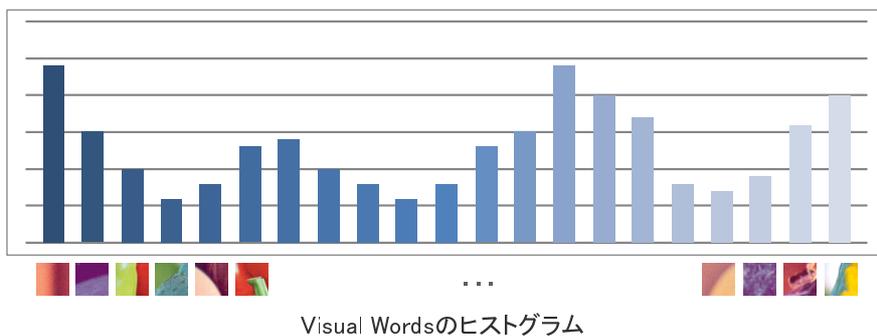


図 2.10: BoVW の概要

境の大まかな構図を表す。GIST はどのような場面を撮影されたかを分類するシーン分類において有効性が示されている。GIST では画像を $4 \div 4$ ブロックに分割し、各ブロック画素をフィルタバンクに適用した応答値が特徴量となる。フィルタバンクは一般的に Gabor フィルタの方向および周波数の組み合わせで 20 種類のフィルタで構成される。3 種類の周波数に対し、最も低周波なものを 4 方向、それ以外を 8 方向として 20 種類とする。RGB の 3 つの色空間に対して行うため、合計で 960 次元である。BoVW は局所特徴量の出現頻度の特徴量であり、画像の向き、被写体の位置に依存しない特徴量であるのに対し、GIST はそれらを考慮するため、被写体のおおよその位置や色、テクスチャが類似した画像に有効であるといえる。

また、GIST と同様に画像の向き、被写体の位置に依存した特徴量で特に歩行者検出において有効性を示している Histograms of Oriented Gradients (HOG) [23] が提案されている。図 2.11 に HOG の概要を示す。HOG は SIFT と同様に勾配方向ヒストグラムに基づく特徴量である。HOG の算出ではまず、画像をあらかじめ定義したサイズのブロックに分割し、それぞれのブロックの勾配方向ヒストグラムを算出する。ブロックの勾配方向ヒストグラムを算出方法は SIFT と同じである。そして、あらかじめ定義したセルサイズでブロックの勾配方向ヒストグラムを式 (2.5) に従って正規

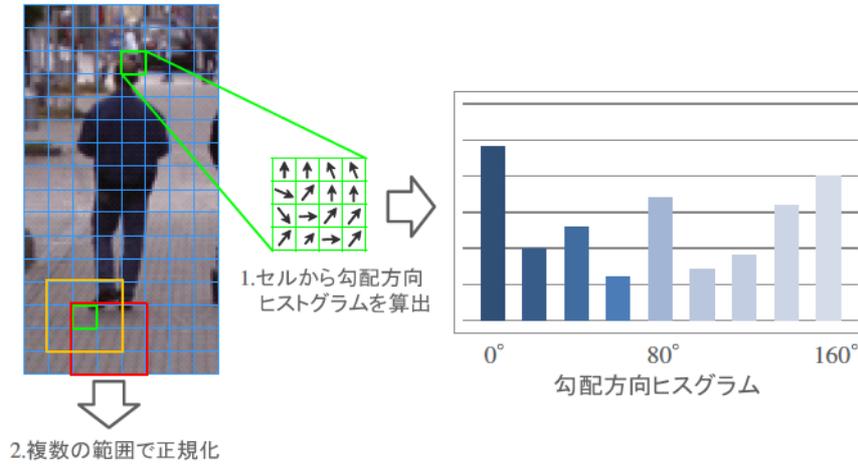


図 2.11: HOG の概要

化する。

$$v(n) = \frac{v(n)}{\sqrt{(\sum_{k=1}^{Nq^2} v(k)^2) + \epsilon}}, \quad (2.5)$$

ただし、 $v(n)$ および $v(n)$ は正規化後および正規化前の勾配方向ヒストグラムであり、 q はセルサイズ、 N は勾配方向数、 ϵ はゼロ除算を防ぐための定数（本論文では 1）である。異なる範囲で正規化を行うことで、周囲の勾配方向ヒストグラムとより詳細に比較した特徴量とすることができる。また、このような正規化方法を行うため、例えば $64 \div 128$ pixel の画像においてブロックサイズが $8 \div 8$ pixel、セルサイズが $2 \div 2$ ブロック、9 方向（ 20° ごと）のとき、HOG の次元数は 3780 次元である。

2.3 データ分布に基づく特徴量の低次元化に関する従来研究

本節では、第 4 章で提案するデータの自己組織化に基づく超低次元化手法と関連の深い先行研究について述べる。データ分布に基づく特徴量の低次元化に関する従来研究の代表的な手法として主成分分析（Principal Components Analysis; PCA）[24]、自己組織化マップ（Self-Organizing Maps; SOM）[25, 26]、多次元尺度構成法（Multi-Dimensional Scaling; MDS）[27]が挙げられる。PCA, SOM, MDS とともにデータ分布やデータ同士の類似度から特徴量を超低次元化することができる手法である。

2.3.1 主成分分析（Principal Components Analysis; PCA）

PCA はデータ分布の広がりが大きくなる新たな特徴量軸を生成する手法である。図 2.12 に PCA の概念図を示す。この例では、もともとなる特徴量が x, y の 2 次元平面であり、斜め右上方向に細長くデータが分布したとする。このときデータ分布の広がり最も大きくなる方向は u 方向の軸で表され、これが第 1 主成分となる。そして第 1 主成分に直交する方向で最も広がり大きくなるのが v 方向の軸となる。分布しているデータをクラス 1 とクラス 2 の 2 クラス分類を行う場合を考える。図 2.12 のように u の軸でみたときに正の方向にクラス 1 のデータが集まり、負の方向にクラ

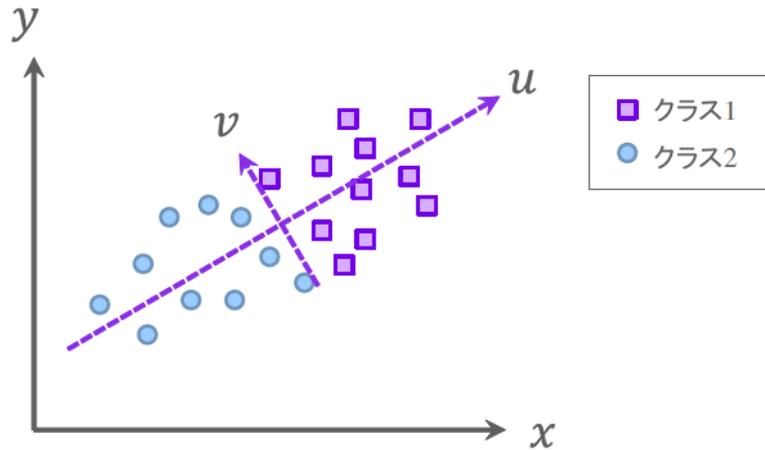


図 2.12: PCA の概念図

ス 2 のデータが集まった場合は第 1 主成分だけを考慮すればよいことになる。しかし、 x, y の軸を用いる場合、これらの軸には依存関係があるため、両方の軸を考慮する必要がある。このように、PCA ではデータの分布の広がり大きい特徴量軸を生成する。

PCA では、もととなる特徴量を線形結合させることで主成分を表現する。このため、PCA は線形なデータに対して有効性を示しているが [28]、非線形なデータには不向きである [29]。図 2.12 の例では、あるデータ (x_1, x_2) の第 1 主成分の値（主成分得点） u_1 および第 2 主成分得点 u_2 は、式 (2.6) および式 (2.7) で表される。

$$u_1 = w_{11}x_1 + w_{12}x_2, \quad (2.6)$$

$$u_2 = w_{21}x_1 + w_{22}x_2, \quad (2.7)$$

ただし、 w_{ij} は第 i 主成分の j 番目の次元の特徴量に対する線形結合の重み係数である。 w_{ij} は共分散行列の固有ベクトルを求めることで得られる。

M 次元のデータ集合 \mathbf{x}_n ($n = 1, 2, \dots, N$) が与えられたとき、 \mathbf{x}_n の平均 $\bar{\mathbf{x}}$ は式 (2.8) で表される。

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n. \quad (2.8)$$

そして、共分散行列 \mathbf{S} は式 (2.9) で表される。

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T. \quad (2.9)$$

\mathbf{S} は $N \div M$ の共分散行列であり、式 (2.9) の定義から、 \mathbf{S} は対称行列である。 $D = \min(N, M)$ とすると、線形代数の定理から、 D 次正方対称行列は D 個の直交する固有ベクトルをもつため、固有ベクトル \mathbf{u}_i ($i = 1, 2, \dots, D$) と、それに対応する固有値 $\lambda_1 > \lambda_2 > \dots > \lambda_D$ が得られる。最大固有値 λ_1 に対応する固有ベクトル \mathbf{u}_1 が第 1 主成分であり、第 i 主成分は \mathbf{u}_i となる。

特徴量を低次元化するには、何次元まで低次元化するのかを決定する必要がある。可視化するために 1~3 次元にするというように決めることも考えられるが、固有値を用いた決定方法がある。固有値は対応する主成分におけるデータの広がりを表すため、固有値が小さいものは近似し

て無視することができる誤差であると考えられることもできる。このため、しきい値 T を設定し、式 (2.10) を満たす最大の d を用いることが一般的である。

$$\frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^D \lambda_i} < T. \quad (2.10)$$

2.3.2 自己組織化マップ (Self-Organizing Maps; SOM)

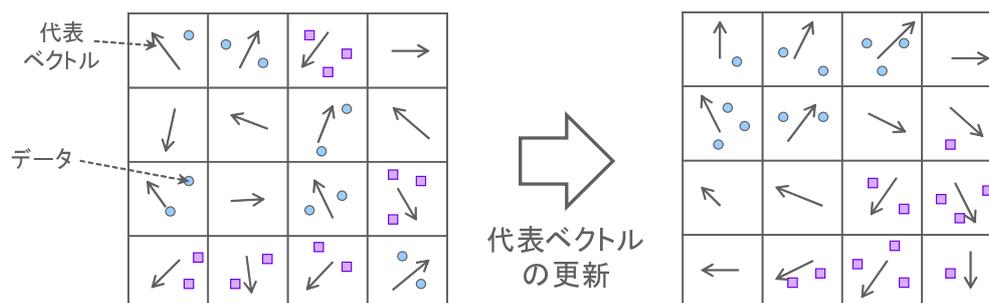


図 2.13: SOM の概要

SOM は Kohonen によって提案された高次元特徴空間の可視化手法のひとつである。図 2.13 に SOM の概要を示す。SOM では、低次元空間に入力特徴量の代表ベクトルをもつユニットを配置し、その代表ベクトルを入力されたデータに近づけるように更新をしていく。代表ベクトルの次元数は入力特徴量の次元数と同じである。図 2.13 は、2 次元空間に 4×4 個の正方形のユニットを配置した例である。ユニットの配置は 1 次元のユニット列にする場合に加え、2 次元では正方形、正六角形のユニットを平面、球面、トーラス状のマップに配置する方法もある。入力されたデータの特徴量を用いることで代表ベクトルの更新を繰り返してデータの自己組織化を行い、類似しているデータが同一もしくは近傍のユニットへと集まる。SOM はこれまでにさまざまな改良がなされており、可視化に重きを置いた従来研究では、代表ベクトルの学習後にユニット間の距離とデータの入力空間での距離に相関をもたせる Visualization-induced SOM (ViSOM) [30, 31] や、2.3.3 項で述べる多次元尺度構成法 (Multi-Dimensional Scaling; MDS) と SOM のハイブリッドな手法である Probabilistic regularized SOM (PRSOM) [32] などが提案されている。

一般的な SOM の代表ベクトルの学習は次の 5 つのステップで行われる。

Step 1. 代表ベクトルの初期化

まず、あらかじめ定めた配置方法で配置したすべてのユニットの代表ベクトルを初期化する。一般的な方法として、代表ベクトルをランダムに初期化する方法、ユニット数だけデータ集合からランダムに選択したデータを代表ベクトルとする方法、マップの端に大きく特徴が異なる代表ベクトルを配置し、その間を線形に補完するように代表ベクトルを決定する方法などが挙げられる。

Step 2. 代表ベクトルの更新に用いるデータの選択

次に、代表ベクトルの更新に用いるデータ \mathbf{x} をデータ集合から選択する。一般的な方法として、データ集合からランダムに選択する方法やデータ集合の順番に従って順に選択する方法がある。

Step 3. 最整合ユニット (Best Matching Unit; BMU) の探索

そして、選択したデータに最も類似したユニット (BMU) c を式 (2.11) を用いて探索する。

$$c = \arg \min_i \sqrt{\sum_{i \in \{1, \dots, M\}} \mathbf{m}_i \cdot \mathbf{x}} \quad (2.11)$$

ただし、 \mathbf{m}_i はユニット i の代表ベクトルであり、 M はすべてのユニット数である。

Step 4. 代表ベクトルの更新

BMU が決定したら、式 (2.12) に従って代表ベクトルの更新を行う。

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) f_{ci}(t) (\mathbf{x} - \mathbf{m}_i(t)), \quad (2.12)$$

ただし、 $\alpha(t)$ は t 回目の繰り返しにおける学習率であり、これは繰り返し回数が増えるにつれて減少する。また、 $f_{ci}(t)$ は t 回目の繰り返しにおける近傍カーネルであり、式 (2.13) はガウシアンカーネルの例を示す。

$$f_{ci}(t) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_c\|^2}{2\sigma(t)^2}\right), \quad (2.13)$$

ただし、 \mathbf{r}_i 、 \mathbf{r}_c はマップにおけるユニット i 、 c の位置である。また、 $\sigma(t)$ は t 回目の繰り返しにおける近傍半径であり、繰り返し回数が増えるにつれて範囲が減少する。

Step 5. 代表ベクトルの更新の繰り返し

Step 2~Step 4 を T_s 回繰り返すことで代表ベクトルを更新する。

2.3.3 多次元尺度構成法 (Multi-Dimensional Scaling; MDS)

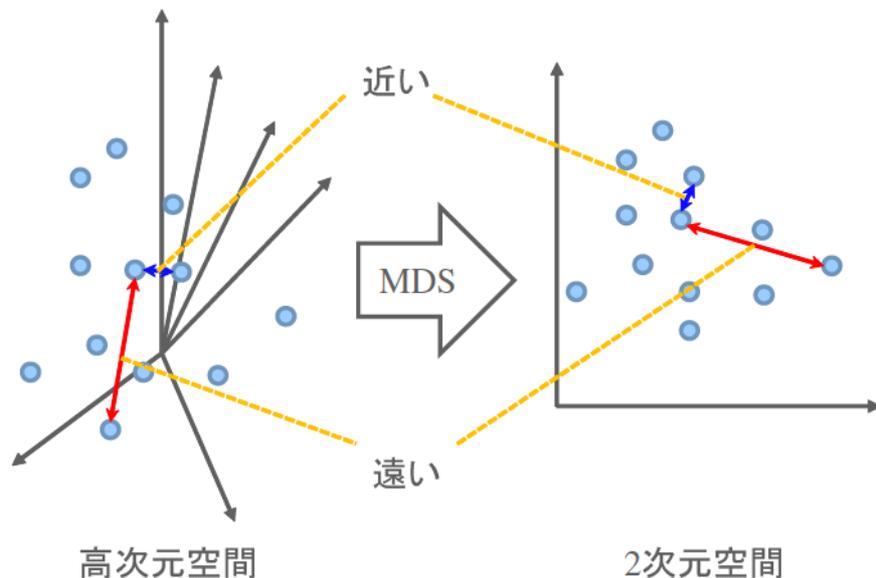


図 2.14: MDS の概念図

本項では、第 4 章において提案するデータの自己組織化に基づく超低次元化手法に関連の深い MDS について述べる。図 2.14 に MDS の概念図を示す。MDS はデータ集合におけるすべてのデー

タのペアについて、その類似性を保ったまま低次元空間にデータを写像する手法である。つまり、低次元空間（一般的には2次元または3次元）において類似したデータほど近くに位置し、そうでないものを遠くに配置する。MDSは、計量多次元尺度法と非計量多次元尺度法の2つに大別される。計量多次元尺度法はデータ間の類似度を間隔尺度で測定し、その類似性を保ったまま低次元空間にデータを写像する。一方で、非計量多次元尺度法ではデータ間の類似度を順序尺度で測定するため、計量多次元尺度法に比べて緩和された写像を行う。計量多次元尺度法のひとつであるサモンマップでは式(2.14)で表されるエネルギー関数 E を最小化することで低次元空間への写像を行う。

$$E = \frac{1}{\sum_{i < j}^N \delta_{ij}} \sum_{i < j}^N \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}, \quad (2.14)$$

ただし、 δ_{ij} および d_{ij} はデータ i, j 間の入力特徴空間における距離、および出力特徴空間における距離である。また、 N は全データ数である。

2.4 既存特徴量を用いた進化計算法による特徴量の低次元化

既存の特徴量を用いた進化計算法による特徴量の低次元化手法には次の2つのアプローチがあげられる。

1. GA を用いた特徴量選択
既存特徴量から与えられたデータ集合に有効な特徴量を選択することで低次元化
2. GP を用いた既存特徴量の組み合わせによる特徴量自動構築
既存特徴量を演算子で組み合わせた関数を構築することで低次元化

次に、それぞれのアプローチの従来研究について述べる。

2.4.1 GA を用いた特徴量選択

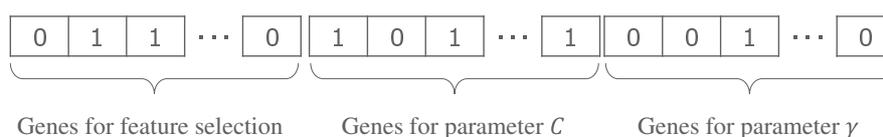


図 2.15: 特徴量選択およびパラメータ最適化に用いる GA の遺伝子型の例

GA を用いた特徴量選択では、選択した特徴量を用いた分類器による分類精度を GA の適応度関数に設定することが一般的である。このとき分類器として用いられるのはクラス分類において特に有効性を示している Support Vector Machine (SVM) [33]が多い。また、SVM はパラメータの設定によって大きくクラス分類の性能が変動することが知られており、使用する特徴量とともに SVM のパラメータを最適化する機会が多い。このように使用する特徴量と SVM のパラメータを最適化することによって、多様なデータセットを提供している UCI データセット [34] に適用した従来研究 [35] や、ネットワークセキュリティにおける侵入検知システムに適用した従来研究 [36]、Eメールのクラス分類に適用した従来研究 [37] など、画像認識以外の分野においても有効性を示している。

図 2.15 に特徴量選択と SVM のパラメータを最適化する場合に用いる遺伝子型の例を示す。遺伝子型は 1 次元のビット列であるが、3 つに区切ることによって特徴量選択、SVM のパラメータである C 、 γ のそれぞれを表現する。特徴量選択のビット列のビット数は入力特徴量の次元数と同じであり、1 のときに対応する特徴量を使用され、0 のときには対応する特徴量を使用されないことを示す。2 つのパラメータは、あらかじめ決めた範囲を用意したビット数で量子化し、それぞれのパラメータとする。このような遺伝子型では、すべての特徴量を用いた場合に分類精度が最も高くなった場合は低次元化することができない。このため、適応度関数に選択された特徴量の数が少ないほど高評価とするような項を導入することで次元数を抑制したうえでより高精度となるような特徴量を選択するという方法もある。しかしながら、特徴量選択で低次元化を行う場合、与えられた既存特徴量の中で分類を高精度にすることができる特徴量が存在しなければならない。特に 3 次元以下へ超低次元化する場合は、必ずしも 3 次元以下の特徴量だけで精度を保つことは困難であると考えられる。そこで、3 次元以下で精度を保つことができる特徴量を、既存特徴量を複数個用いて組み合わせることで構築するアプローチも提案されている。

2.4.2 GP を用いた既存特徴量の組み合わせによる特徴量自動構築

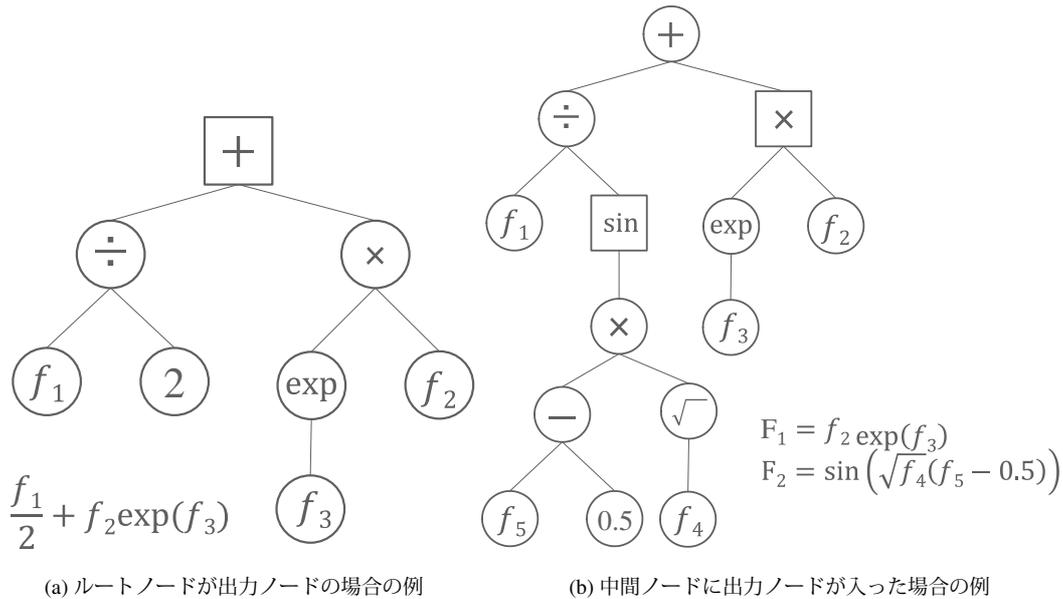


図 2.16: GP を用いた既存特徴量の組み合わせによる特徴量の自動構築結果の例

GP を用いて、既存の特徴量を組み合わせて演算する数式を最適化し、新たな特徴量を構築する手法が提案されている [38–41]。これらの手法では、適応度関数に分類精度を用いることが多く、データのクラスを考慮して特徴量を構築することができる。図 2.16 に、GP を用いて構築される数式の例を示す。図 2.16 では、正方形のノードが構築された特徴量を表す。図 2.16(a) に示すように、基本的にはリーフノードが既存特徴量または定数であり、中間ノードが演算子を表し、ルートノードが数式の出力値として特徴量となる。このため、1 つの木構造で 1 つの特徴量を構築する。一方で図 2.16(b) に示すように、文献 [39] では中間ノードに出力ノードを選択することができるようにすることで複数の特徴量を構築することを可能としている。これらの手法はリーフノードで既存特

微量を選択するため、既存特徴量より使用する特徴量を少なくすることができる。また、GPにおいては構築した特徴量を評価する適応度関数が重要となる。このため、適応度関数にk近傍法を用いた従来研究 [39]や Generalized Linear Machine (GLIM) を用いた従来研究 [40]、決定木 (C4.5) を用いた従来研究 [41]など、適応度関数にはさまざまな分類器が採用されている。

2.5 まとめ

本章では、本研究と関連の深い進化計算法、画像特徴量、データ分布に基づく特徴量の低次元化に関する先行研究、および特徴量の組み合わせに基づく特徴量の低次元化に関する先行研究について述べた。第3章からは、“高い認識性能”、“低い計算コスト”、“認識処理のわかりやすさ”の実現を目的とした2つのアプローチの手法を提案する。

第3章 人物の部分領域に着目した姿勢変化や隠れに頑健な人物検出

3.1 はじめに

本章では，“高い認識性能”，“低い計算コスト”，“認識処理のわかりやすさ”を実現する1つ目のアプローチとして，高い認識性能を保つために認識対象の性質および画像以外の特徴量として距離情報を利用し，ヒトにとって直観的な条件判断を行うネットワークを認識器に採用して特徴量を選択することで，低い計算コストと処理のわかりやすさの両立を図る．本章では，認識対象を空港などの巡回警備において撮影される人物とする．このため，歩行者のような立ち姿の人物に加えて，座っている人物や人や荷物などで一部が隠れてしまっているような人物も検出対象とする．本研究では，身体全体では姿勢によって大きく見た目が異なるが，部分的（手や足，頭など）には見た目の変化が少ないことを人物の性質として仮定する．そして，画像以外の特徴量としてカメラから物体までの距離情報を用いて，画像を人物の一部が切り出されるような過分割を行い，それぞれの領域が人物の候補領域であるかどうかの分類を行う．その後，人物の候補領域を統合することで姿勢変化や隠れに頑健な人物検出を行う．人物検出実験では，乗用移動台車に距離センサおよびカメラを設置し，空港の巡回警備を想定した環境から歩行者，姿勢のさまざまな人物および隠れのある人物の検出を行い，提案手法の有効性を検証する．

3.2 人物の部分領域に着目した特徴量選択と条件判断に基づく人物検出

3.2.1 提案手法の概要

提案手法は次の3つのステップで構成される．図 3.1 に提案手法の処理の流れを示す．

Step 1. 距離情報を用いた領域分割

まず，距離情報を用いて画像の領域分割を行い，人物や物体が数個の領域となるように画像の過分割を行う．過分割は，距離によるクラスタリングを行い，距離が近い画素をまとめることで行う．また，過分割された領域の中で距離を取得することができなかった領域と地面の平面に属する領域は人物ではないとして，それらの領域以外の領域を Step 2 における人物・非人物分類を行う対象の領域とする．

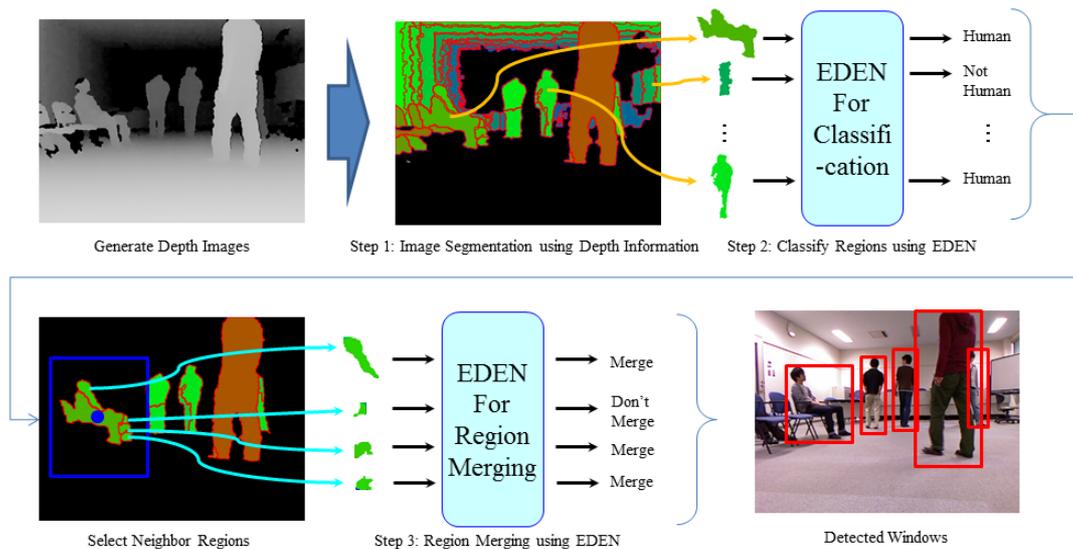


図 3.1: 提案手法の処理の流れ

Step 2. 過分割された領域の人物・非人物分類

次に、分類の対象となった各領域を「人物」、「非人物」の2クラスに分類する。分類には進化的条件判断ネットワーク（Evolutionary Decision Network; EDEN）を用い、領域の特徴量に基づいて周囲の領域を考慮した分類を行う。

Step 3. 人物と分類された領域の統合

最後に、人物と分類された領域を同一人物と考えられる場合は統合し、統合された領域の外接矩形を検出ウィンドウとして出力する。各領域の統合では、人物と分類されたすべての領域に対し、各領域から一定範囲内にある領域をその領域に統合するかどうかの判断を行う。この統合の判断にも EDEN を用い、EDEN における分類するクラスを「統合する」、「統合しない」の2クラスとする。

3.2.2 Step 1. 距離情報を用いた領域分割

Step 1 では、距離情報を用いて画像の領域分割を行う。人物や物体が部分的な数個の領域となるような過分割を目的としており、人物や物体の部分は距離が近いことが考えられるため、距離情報を用いて領域分割を行う。図 3.2 に Step 1 の処理の流れを示す。

まず、カメラで取得されたカラー画像の各画素をカメラからの距離に変換した距離画像から、各階調値のヒストグラムを算出する。そして、このヒストグラムを用いて k 平均法 [42] を用いて階調値を k 個のクラスタに分類する。k 平均法による階調値のクラスタリングは、次の手順で行う。

1. 各階調値にクラスタを割り当てる。本手法では、階調値の範囲である $[0,255]$ を k 等分し、分割されたそれぞれの範囲に $1 \sim k$ のクラスタを割り当てる。
2. クラスタの重心を計算する。本手法では、 n 番目のクラスタの重心 C_n を式 (3.1) を用いて計算する。ここで、 $c(j)$ は j 番目の階調値のクラスタを表し、 $h(i)$ は i 番目の階調値の距離ヒストグラムを表す。

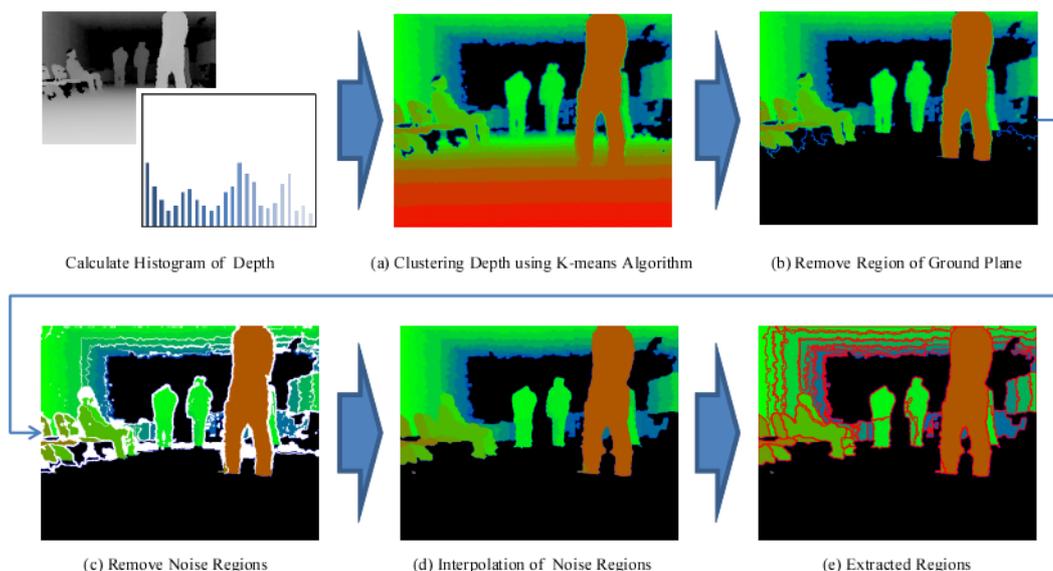


図 3.2: Step 1 の処理の流れ

$$C_n = \frac{\sum_{i,j|c(j)=n} i \mathcal{H}(i)}{\sum_{i,j|c(j)=n} h(i)}. \quad (3.1)$$

3. 各階調値のクラスタを、最も近い重心のクラスタに変更する。
4. クラスタの変更がなかった場合は終了する。そうでない場合は、2. へ戻る。

このようにして、図 3.2 の (a) に示すような距離を k 個に量子化した距離画像を得る。ここで、部分ごとに過分割するように k はある程度大きい数に設定する。

次に、分類の対象領域を絞るために地面の平面に属する領域の抽出を行う。そして図 3.2 の (b) に示すように、地面の平面に属する領域を黒い画素で表される距離を取得することができなかった画素と同じ扱いにする。この領域は人物ではないとして、Step 2 において人物・非人物分類を行わず非人物とする。本手法では、距離画像の各画素を点群データとして 3 次元実空間に写像することで、3 次元実空間における各画素の位置を取得することができる。図 3.3 に示すように、本手法では乗用移動台車に前方を計測するように水平に距離センサを設置する。このため y 軸が高さを表しており、移動中の前後の傾きがほとんどないと仮定することで一定の高さで地面の平面を推定することができる。したがって、 y 軸に対して θ_g [m] 以下の画素を地面の平面に属する領域として抽出する。なお、本実験で用いる乗用移動台車である Segway では前進・後退の移動に体の重心を前後に移動させることで行うため、若干の前後の傾きが存在する。しかし、この傾きはごくわずかであるため、 θ_g を余裕をもって設定することで、乗用移動台車の前後の傾きを測定することなく分類を行う対象の領域を絞ることができる。

そして、 k 平均法によってクラスタリングされた各クラスタの領域および、図 3.2 の (b) における黒い画素で表される領域でラベリングを行い、孤立領域を抽出する。図 3.2 の (c) に示すように、孤立領域のうち面積が θ_n [pixel] 以下のものをノイズとして除去し、画素のラベルを消去する。その後、ノイズとして除去された領域を周囲の領域から成長させることで補完を行う。領域の成長は、左上の画素からラスタスキャンを行い、注目画素の隣の画素がラベルがない場合に注目画素のラベ

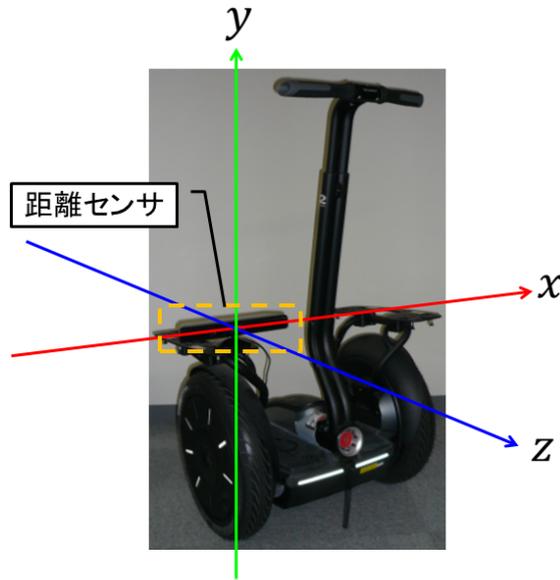


図 3.3: 距離センサの設置方向と用いる座標系

ルを割り当てることで行う。その際、隣の画素として調べる方向を右→下→左→上の順に変化させて成長を行う。領域成長をラベルのない画素がなくなるまで繰り返し、図 3.2 の (d) に示すように、ノイズとして除去した領域の補完を行う。そして、ラベルの付いている領域のうち距離を取得することができなかった領域のラベルを消去し、図 3.2 の (e) のように残った領域を Step 2 で分類を行う対象の領域とする。

3.2.3 Step 2. 過分割された領域の人物・非人物分類

Step 2 では、過分割された領域のうち分類の対象となった各領域を人物・非人物の 2 クラスに分類する。分類は EDEN を用いて行う。Step 2 における EDEN (EDEN_{s2}) の入力ノードは 2 つとし、分類対象の領域の特徴量ベクトルを入力データとして入力し、分類対象の領域に接している周囲の領域を 1 つの領域にまとめ、その領域の特徴量ベクトルを参照入力として入力する。こうすることで、参照入力である周囲の領域を考慮した分類ができることを期待する。また、2 つの出力ノードを用意し、それぞれに「人物」、「非人物」のクラスを割り当てる。

各領域の特徴量の算出

EDEN_{s2} では各領域の特徴量を用いて分類を行うため、まず各領域の特徴量を計算する。本手法ではカラー画像と距離情報を取得することができ、また過分割された領域ごとに分類を行う。このため、特徴量には文献 [12] で用いられている色空間における統計量に加え、距離画像の特徴量と距離画像の各画素を 3 次元実空間へ写像した際に得られる点群データの特徴量、および領域の形状の特徴量を用いる。本手法で用いる色空間および距離画像の統計量と点群データの特徴量、領域の形状の特徴量を表 3.1、表 3.2、表 3.3 それぞれに示す。

色空間は RGB 色空間を用い、色空間の特徴量に用いる統計量を距離画像にも用いて距離画像の特徴量を算出する。点群データの特徴量では、図 3.3 に示すように x 軸はカメラの光軸に直交し、

表 3.1: 色空間および距離画像の特徴量に用いる統計量

名称	内容
Mean	平均値
Max	最大値
Min	最小値
SD	標準偏差
Range	最大値から最小値を引いたもの
1 sigma in rate	平均値 \bullet 標準偏差に収まっている画素の割合
3 sigma out rate	平均値 $\bullet 3\div$ 標準偏差に収まっていない画素の割合
First quartile	値を昇順に並べたとき下から 1/4 にあたる値
Median	値を昇順に並べたとき中央にあたる値
Third quartile	値を昇順に並べたとき下から 3/4 にあたる値
Mode	最も出現頻度の高い値

表 3.2: 点群データの特徴量

名称	内容
Range X,Y,Z	x, y, z それぞれの軸における点群データの幅
SD X,Y,Z	x, y, z それぞれの点群データの標準偏差
1 sigma in rate X,Y,Z	x, y, z それぞれの平均値 \bullet 標準偏差に収まっている点の割合
3 sigma out rate X,Y,Z	x, y, z それぞれの平均値 $\bullet 3\div$ 標準偏差に収まっていない点の割合
Max Y	y の最大値
Min Y	y の最小値
First quartile Y	y の値を昇順に並べたとき下から 1/4 にあたる値
Median Y	y の値を昇順に並べたとき中央にあたる値
Third quartile Y	y の値を昇順に並べたとき下から 3/4 にあたる値

表 3.3: 領域の形状の特徴量

名称	内容
Width	領域の外接矩形の幅
Height	領域の外接矩形の高さ
Aspect ratio	領域の外接矩形のアスペクト比
Area	領域の面積
Filling rate of box	領域の外接矩形に対する充填率

水平方向であるため、 x の値の統計量を用いると人物がカメラに対して左右のどちらに写るかで分類してしまう可能性がある。このため、学習データの x 軸方向の人物の位置に過学習することを抑制するために x の値の統計量を用いない。一方で、 y 軸はカメラの光軸に直交し、鉛直方向であるため、 y の値は点群が存在する3次元実空間の高さを表す。乗用移動台車で移動計測を行った場合、計測される人物の高さは変化が少なく分類に有効に働くと考えられるため、 y の値の統計量は特徴量として用いる。また領域の形状の特徴量については、近い人物は領域の大きさが大きいなどカメラと被写体との距離で相関関係があるため、距離の統計量もしくは z の値と組み合わせることで有効に働くことを期待する。

適応度関数

算出された特徴量を用いて各領域を正しく人物・非人物に分類することができるように進化計算でネットワークを構築する。このとき、最適化の過程で構築されたネットワークの良さを評価する必要がある。本手法では Step 3 において人物と分類された領域の統合を行うため、ここでの誤分類が最終的な検出結果に大きく影響する。そこで、人物・非人物の両方の誤分類を抑制するような最適化を促すために、評価関数には検出性能の評価に用いられる適合率と再現率の調和平均である F 値を用いる。

EDEN_{s2} のネットワークの最適化に用いる適応度関数には F 値を用いる。適応度関数 $fitness_{s2}$ を式 (3.2) に示す。

$$fitness_{s2} = F + \frac{\beta}{n_{e2}}, \quad (3.2)$$

$$F = \frac{2r_h}{n_h + c_h}, \quad (3.3)$$

ただし、 n_h は人物と分類した領域数、 r_h は人物と分類した領域のうち正解した領域数、 c_h は人物と分類すべき領域数を表す。また、 n_{e2} は構築されたネットワークの各ノード間をつなぐエッジの数であり、 n_{e2} の逆数を加算することで、同じ分類結果となる場合は、よりコンパクトなネットワークの方が高い評価としている。なお、 β は定数であり、本論文では実験的に 10^{-4} に設定した。

3.2.4 Step 3. 人物と分類された領域の統合

Step 3 では、Step 2 において人物と分類された領域を同一人物と考えられる場合は統合し、統合された領域の外接矩形を検出ウィンドウとして算出する。各領域の統合では、人物と分類されたすべての領域に対し、各領域から一定範囲内にある領域をその領域に統合するかどうかの判断を行う。この統合の判断にも EDEN を用い、Step 3 における EDEN (EDEN_{s3}) においては分類するクラスを「統合する」、「統合しない」の2クラスとする。人物と分類されたすべての領域において検出ウィンドウが算出されるため、人物ごとに複数の検出ウィンドウが算出される。このため、位置や大きさの近い検出ウィンドウを統合することで最終的な検出ウィンドウを出力する。

領域の統合による検出ウィンドウの算出

領域の統合は、ある領域を注目領域としたとき、注目領域から一定範囲内にある領域を統合する候補の領域として、その領域を注目領域に統合するかどうかの判断を行う。図 3.4 に統合する候補の領域の選択範囲の例を示す。このように注目領域の中心を中心とした一定の大きさの矩形の

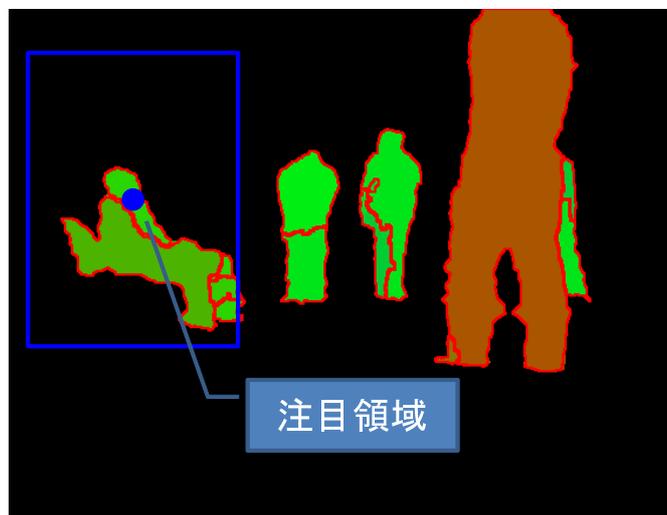
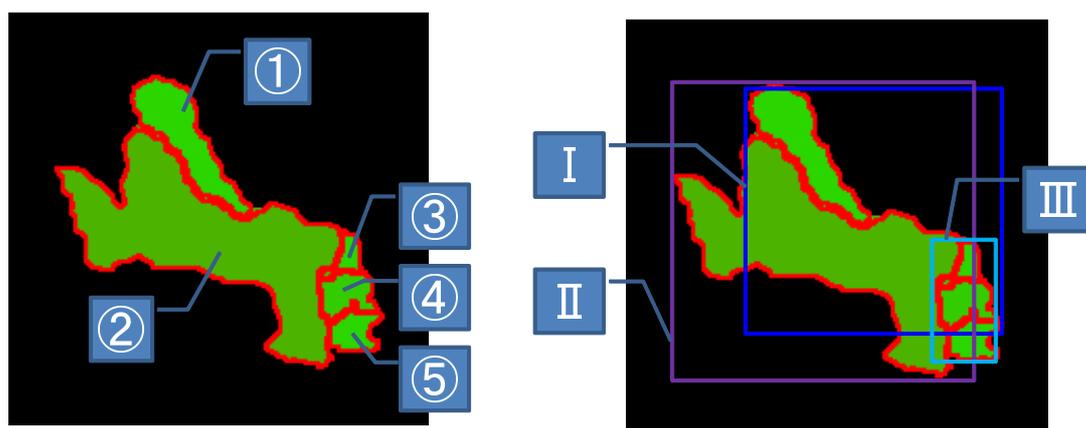


図 3.4: 統合を行う候補の領域の選択範囲



(a) 人物と分類された領域の例

(b) 検出ウィンドウの例

図 3.5: 領域の統合による検出ウィンドウの算出の例

範囲内にある領域を、それぞれ注目領域に統合するかどうかを判断する。図 3.5 に領域の統合による検出ウィンドウの算出の例を示す。図 3.5(a) に示すように 5 つの領域が人物と分類されたとすると、この 5 つのそれぞれの領域に対して統合する候補の領域を統合するかどうかの判断を行う。そして、統合された領域の外接矩形を算出し、これを検出ウィンドウとする。例えば、図 3.5(a) における①を注目領域とした場合に残りの 4 つの領域のうち、③と④の領域を①に統合すると判断された場合、図 3.5(b) における I の検出ウィンドウが得られる。同様に、図 3.5(a) における②、③をそれぞれ注目領域とした場合に、①と③、④と⑤がそれぞれ②、③に統合されると判断された場合は、図 3.5(b) における II、III の検出ウィンドウが得られる。このようにして領域を統合し、検出ウィンドウを算出するため、図 3.5(b) のように同一の人物について複数の検出ウィンドウが算出されることになる。

本手法では距離情報を利用することができるため、統合する候補の領域を選択する範囲を 3 次元実空間における矩形の大きさを用いて算出する。人物の姿勢が変化した場合でも同一人物の手や足などの部分は大きく離れることはないため、3 次元実空間において幅 W_m ÷ 高さ H_m [m] の大きさの矩形から統合する候補の領域を選択する。このとき、この 3 次元実空間の矩形を画像空間に写像することで、統合する領域を選択する範囲の画像空間における矩形の大きさを算出する。なお、この矩形の範囲内にあるかどうかは、各領域の面積に対する、矩形の範囲とその領域の重なっている部分の面積の割合を用いて判断する。これが θ_b 以上であるものを矩形の範囲内にあるものとして、統合する領域の候補に加える。

このように注目領域に対して統合する候補の領域が決定されたら、注目領域ごとに候補の領域を注目領域に統合するかどうかの判断を EDEN_{s3} を用いて行う。EDEN_{s3} では、入力データとして統合する候補の領域を入力し、「統合する」、「統合しない」の 2 つのクラスに割り当てられた出力ノードのどちらに到達するかで注目領域にその領域を統合するかどうかの判断を行う。このとき、参照入力データとして注目領域を入力することで、注目領域の特徴を考慮した統合の判断を行うことを期待する。EDEN_{s3} によって各注目領域ごとに統合する領域が決定したら、各注目領域および統合された領域の外接矩形を算出することで検出ウィンドウを算出する。

また、算出されたウィンドウのうち、3 次元実空間におけるウィンドウの幅と高さを用いてウィンドウの面積を算出する。これらの面積を用いて、次に行う算出されたウィンドウの統合に用いるウィンドウの候補の絞り込みを行う。算出されたウィンドウの面積が $m_h \div W_h \div H_h$ より小さい場合および、 $M_h \div W_h \div H_h$ より大きい場合は、算出されたウィンドウが人物のウィンドウと大きさが大きく異なるとみなして、算出されたウィンドウの統合を行うウィンドウの候補に加えない。ここで、 W_h および H_h は実験で定める基準となる人物のウィンドウの幅および高さであり、 m_h および M_h は人物のウィンドウとみなす最小および最大の割合である。

統合された領域から取得した検出ウィンドウの統合

最後に、各領域の統合によって算出された検出ウィンドウの統合を行う。検出ウィンドウの統合は、各検出ウィンドウの位置と大きさをクラスタリングを行ってまとめることで行う。まず、画像空間におけるウィンドウの中心座標 (x, y) 、幅 w 、高さ h を Mean-Shift クラスタリングを行い、これらの 4 次元空間における点の密度が極値となる点へ移動する。式 (3.4)～式 (3.7) に x, y, w, h それぞれの Mean-Shift Vector を示す。ここで、 i 番目の検出ウィンドウの中心座標、幅、および高さをそれぞれ x_i, y_i, w_i, h_i とし、移動している検出ウィンドウの中心座標、幅、および高さをそれぞれ x, y, w, h とする。また、用いるカーネル関数 $kernel$ は式 (3.9)～式 (3.12) を用いて、式 (3.8) で表さ

れる．ただし，中心座標，幅，および高さのバンド幅をそれぞれ b_x, b_y, b_w, b_h とする．中心座標，幅，および高さそれぞれの Mean-Shift Vector がしきい値 th_{msv} を下回るまで点の移動を繰り返す．

$$m(x) = \frac{\sum_{i=1}^n x_i \times \text{kernel}(i)}{\sum_{i=1}^n \text{kernel}(i)} \quad x, \quad (3.4)$$

$$m(y) = \frac{\sum_{i=1}^n y_i \times \text{kernel}(i)}{\sum_{i=1}^n \text{kernel}(i)} \quad y, \quad (3.5)$$

$$m(w) = \frac{\sum_{i=1}^n w_i \times \text{kernel}(i)}{\sum_{i=1}^n \text{kernel}(i)} \quad w, \quad (3.6)$$

$$m(h) = \frac{\sum_{i=1}^n h_i \times \text{kernel}(i)}{\sum_{i=1}^n \text{kernel}(i)} \quad h, \quad (3.7)$$

$$\text{kernel}(i) = \text{kernel}_x(i) \div \text{kernel}_y(i) \div \text{kernel}_w(i) \div \text{kernel}_h(i), \quad (3.8)$$

$$\text{kernel}_x(i) = \exp\left(\frac{(x - x_i)^2}{b_x^2}\right), \quad (3.9)$$

$$\text{kernel}_y(i) = \exp\left(\frac{(y - y_i)^2}{b_y^2}\right), \quad (3.10)$$

$$\text{kernel}_w(i) = \exp\left(\frac{(w - w_i)^2}{b_w^2}\right), \quad (3.11)$$

$$\text{kernel}_h(i) = \exp\left(\frac{(h - h_i)^2}{b_h^2}\right). \quad (3.12)$$

すべての検出ウィンドウの中心座標，幅，および高さを極値の点へ移動した後，最近傍法によって近隣の検出ウィンドウを統合する．最近傍法では，2つの移動された点がしきい値を下回る場合は統合し，統合される点がなくなるまで繰り返すことで近隣の点を統合する．移動された各点に対して，まず中心座標，幅，および高さのユークリッド距離 d を式 (3.13) で求め， d が最小となる点を探索する．ここで，注目している検出ウィンドウの中心座標，幅，および高さをそれぞれ x, y, w, h とし，探索の際に選ばれた点 i の中心座標，幅，および高さをそれぞれ x_i, y_i, w_i, h_i とする．そして， d がしきい値 θ_a を下回った場合，式 (3.14)～式 (3.17) を用いることで，検出ウィンドウの統合を行う．ここで， d が最小となる点 j の中心座標，幅，および高さをそれぞれ x_j, y_j, w_j, h_j とし，統合されたウィンドウの中心座標，幅，および高さをそれぞれ $x^\epsilon, y^\epsilon, w^\epsilon, h^\epsilon$ とする．これらの処理を統合される検出ウィンドウがなくなるまで繰り返し，統合されて残った検出ウィンドウを最終的な検出結果とする．

$$d = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (w - w_i)^2 + (h - h_i)^2}, \quad (3.13)$$

$$x^\epsilon = \frac{x + x_j}{2}, \quad (3.14)$$

$$y^\epsilon = \frac{y + y_j}{2}, \quad (3.15)$$

$$w^\epsilon = \frac{w + w_j}{2}, \quad (3.16)$$

$$h^\epsilon = \frac{h + h_j}{2}. \quad (3.17)$$

得られた最終的な検出結果から，EDEN_{s3} によるネットワーク構築の際のネットワークの評価を行う．EDEN_{s3} の最適化における統合処理の評価には式 (3.18) の適応度関数を用いる．適応度関数には，検出ウィンドウによる人物検出の再現率 R_m ，全 T フレームにおけるフレーム間の検出ウイ

表 3.4: 学習およびテストデータの人物の内訳

	立ち姿	立ち姿 (隠れあり)	立ち姿以外	立ち姿以外 (隠れあり)
学習	114	54	9	14
テスト	55	39	33	17

ンドウ数の変動のなめらかさ f_w および、ネットワークの各ノード間をつなぐエッジの数 n_{e3} を用いる。ここで、 w_t はフレーム t における検出ウィンドウ数であり、 f_w は式 (3.20) に示すように、連続する 2 フレーム間の検出ウィンドウ数が大きく変動するほど低い値となる。これは、移動計測を行って計測される人物の数はフレーム間で大きく変化することがないという仮定のもと、過検出を抑制するような最適化を促すことを目的としている。

$$fitness_{s3} = \alpha R_m + (1 - \alpha) f_w + \frac{\beta}{n_{e3}}, \quad (3.18)$$

$$f_w = \frac{1}{T-1} \sum_{t=2}^T d_t, \quad (3.19)$$

$$d_t = \begin{cases} \exp\left(-\frac{(w_{t-1} - w_t)^2}{2\sigma^2}\right) & (\|w_{t-1} - w_t\| < 5) \\ 0 & (otherwise) \end{cases} \quad (3.20)$$

3.3 人物検出実験

3.3.1 実験設定

対象とした環境と検出対象および正解データ

対象とした環境は、複数の歩行者や椅子に座っている人物などの存在する屋内環境である。図 3.6(a), 図 3.6(b) に示すように、椅子や机など屋内環境にあるものを配置した。このような環境において複数の人物が歩いたり座ったりしているシーンにおいて、乗用移動台車を用いて移動計測を行ってカラー画像と距離情報を 2 fps で取得した。学習データ、テストデータは、それぞれ異なる 100 フレームのシーケンスである。取得したカラー画像の例を図 3.7 に示す。図 3.7(a) では、歩行者の他に椅子に座っている人物や、人物の重なりでオクルージョンの存在する人物が写されている。図 3.7(b) では、環境内にある椅子でオクルージョンが発生している人物が写されている。このように姿勢が多様である人物や、人物や物体でオクルージョンが存在する人物、画像の端で人物全体が写っていない人物を含む、画像中のすべての人物を検出対象とした。本実験ではこのような人物の検出結果の正解データとして、図 3.8 に示すような検出ウィンドウを作成した。検出ウィンドウは左上の頂点の画像空間における座標および、ウィンドウの幅、高さで構成されている。正解データは、人物として人が見て判断できる部分の外接矩形を人手で算出した。なお、オクルージョンや画像の端で人物全体が写っていない人物に対しては、人物全体のおよそ 3 割以上の部分が見えている場合に人物とみなし、正解データを作成した。表 3.4 に学習およびテストデータの人物の内訳を示す。学習データでは最大 6 人、テストデータでは最大 4 人が同時に 1 フレームに存在し、人物が存在しないフレームは学習で 14 フレーム、テストで 29 フレームであった。



(a) 対象とした環境の外観 1



(b) 対象とした環境の外観 2

図 3.6: 対象とした環境の外観

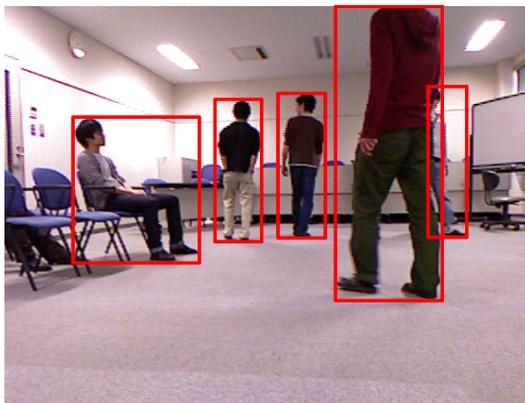


(a) 取得したカラー画像の例 1

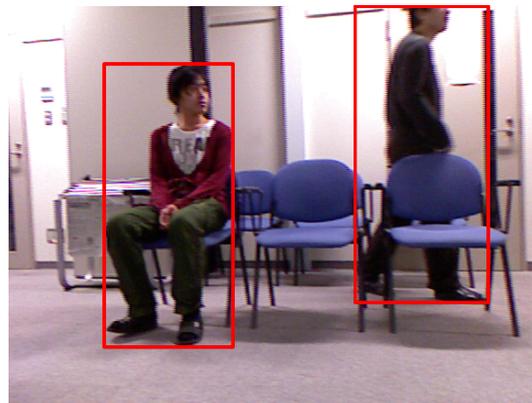


(b) 取得したカラー画像の例 2

図 3.7: 取得したカラー画像の例



(a) 正解データの例 1



(b) 正解データの例 2

図 3.8: 正解データの検出ウィンドウの例

実験に用いた機器

実験では乗用移動台車にカメラと距離センサを取り付けて計測を行った。乗用移動台車には Segway 社の Segway PT i2 (以下, Segway と呼ぶ) を用いた。また, カメラと距離センサにはこれらの両方が搭載されている Microsoft 社の Xbox360 Kinect センサ (以下, Kinect と呼ぶ) を用い, カラー画像と距離情報を取得した。図 3.3 に示したように, 前方を計測するように Kinect を Segway の荷台に設置した。Kinect で取得した点群データの座標系は, Kinect のカメラの焦点を原点とし, Segway の前方で地面と平行に z 軸をとり, z 軸に直交し地面と平行に x 軸をとる。 y 軸は地面に対して垂直にした。 Kinect の設置角度は, Kinect の光軸が z 軸と平行に, また Kinect 本体が x 軸と平行に地面に対して水平になるような角度とした。 Kinect で取得されるデータは, カラー画像とその各画素の 3 次元実空間における位置である。 図 3.9 に Kinect で取得したカラー画像と各画素の 3 次元実空間における位置をカメラからの距離に変換した距離画像の例を示す。 距離画像への変換は, 式 (3.21) に示すように, 距離画像の各画素 $d(u, v)$ の z の値 $z(u, v)$ を 9[m] を最大距離としてグレースケール画像における階調値の範囲の [0,255] に正規化することで行った。 なお, 近いほど白く, 遠いほど黒く示されている。

$$d(u, v) = \begin{cases} 255(1 - \frac{z(u, v)}{9}) & (z(u, v) < 9) \\ 0 & (\text{otherwise}) \end{cases} \quad (3.21)$$

図 3.9(b) の青色の画素は距離を取得できなかった画素および 9[m] より遠い位置にある画素を表している。このように距離の取得できない画素が存在するのには 2 つの理由がある。 1 つ目は, カメラの画角より距離センサの画角が狭いため, 距離センサの画角より外側は距離を取得できないためである。 2 つ目は, Kinect の距離センサの性質によるものである。 Kinect の距離計測はパターン投影方式 [43] を採用しているため, カメラと距離センサの位置の違いでカメラには写っているが距離センサに写っていない部分が生じ, この部分は距離を取得することができない。これは物体の境界付近の背景に見られ, その部分が物体に隠れてしまうために起こる。また, 投影されたパターンの対応をとることで距離が計測されるため, 頭髪の部分などの細かな凹凸部分ではパターンの対応が取れずに距離を取得できない場合もある。さらに, Kinect ではパターンは赤外線で投影されるため, 吸収率の高い黒い部分や反射してしまう鏡や金属などの表面も距離を取得することができない場合がある。

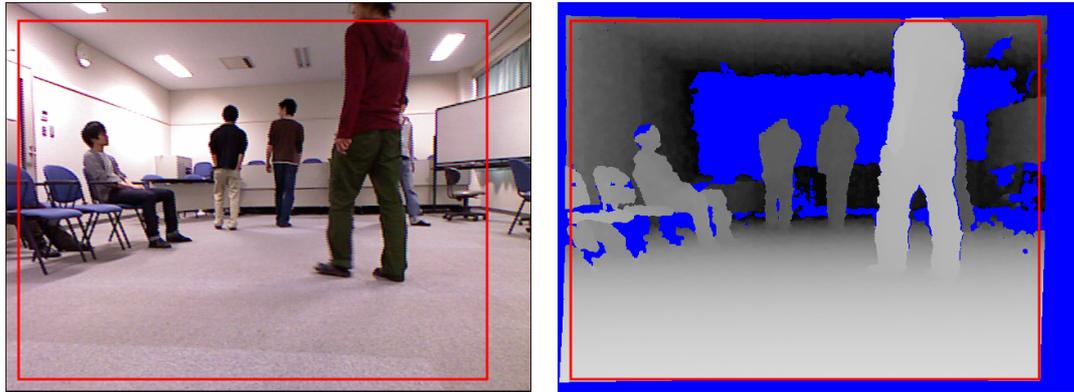
そこで, 1 つ目の理由で取得できない部分は本実験では利用せず, 図 3.9 の赤い枠で囲まれた部分を切り取ってこの範囲を用いた。この範囲は画像の左上を画像空間の原点とすると, 矩形の左上の頂点を (16, 23) とする 570÷440 pixel の範囲である。 2 つ目の理由で取得できない部分は画像全体からすると微小な領域であるため, 提案手法の各 Step でそれぞれ考慮した。

提案手法の各種設定

ここでは, 提案手法の設定について述べる。

Step 1. 距離情報を用いた領域分割

Step 1 における各種パラメータの設定は表 3.5 のとおりである。本研究で対象とする環境ではある程度の数の人物や物体が存在すると仮定している。このため, 各人物・物体を数個の領域に過分割するように, 距離によるクラスタリングにおけるクラスタ数 k を大きめに設定した。ノイズとし



(a) カラー画像の例

(b) 距離画像の例

図 3.9: Kinect で取得したカラー画像と距離画像の例

て除去する領域の面積しきい値 θ_n および、地面として分類の対象領域を絞る地面の高さ θ_g は複数のパラメータを変えて予備実験を行うことで決定した。また、距離を取得できなかった領域は微小な領域であるためノイズ除去の処理を行うことで考慮した。

Step 2. 過分割された領域の人物・非人物分類

EDEN_{s2} の設定を表 3.6 に示す。EDEN_{s2} では、入力ノードを 2 つ用意し、それぞれ入力データとして分類対象の領域を、参照入力データとして分類対象の領域に接する周囲の領域をまとめた 1 つの領域を割り当てた。出力ノードは人物クラス、非人物クラスに対応する 2 つである。また、中間ノードとして用いたノードは、入力データの特徴量をしきい値で分岐する絶対値比較ノードと、入力データと参照入力データの特徴量の差分をしきい値で分岐する差分値比較ノードの 2 種類である。なお、中間ノードの最大数、世代数および突然変異率は複数のパラメータを変えて予備実験を行うことで決定した。使用した特徴量は、表 3.1、表 3.2、表 3.3 に示した合計 66 種類の特徴量を用いた。特徴量は領域ごとに算出されるため、距離を取得することができた領域の画素だけを用いて特徴量を算出することで、距離を取得することができなかった領域を考慮した。

また、学習データにおける過分割された各領域の正解クラスは正解データの検出ウィンドウを用いて決定した。図 3.10 に示すように、各領域の面積に対する検出ウィンドウとその領域の重なっている部分の面積の割合が 0.8 以上である場合にその領域のクラスを人物とした。

表 3.5: Step 1 における各種パラメータ設定

内容	値
距離によるクラスタリングにおけるクラスタ数 k	60
ノイズとして除去する領域の面積しきい値 θ_n	100 pixel
地面として分類の対象領域を絞る地面の高さ θ_g	0.51[m]

表 3.6: 人物・非人物分類を行う EDEN_{s2} の設定

内容	値
入力ノード数	2 (入力：分類対象の領域，参照入力：分類対象の領域に接している周囲の領域)
中間ノードの最大数	100
出力ノード数	2 (人物クラス，非人物クラス)
ノードの種類	2 種類 (絶対値比較，差分値比較)
世代数	1,000,000
突然変異率 P_m	0.02
使用した特徴量	66 種類 (RGB 色成分および距離画像：4 ÷ 11 種類，点群データ：17 種類，領域：5 種類)

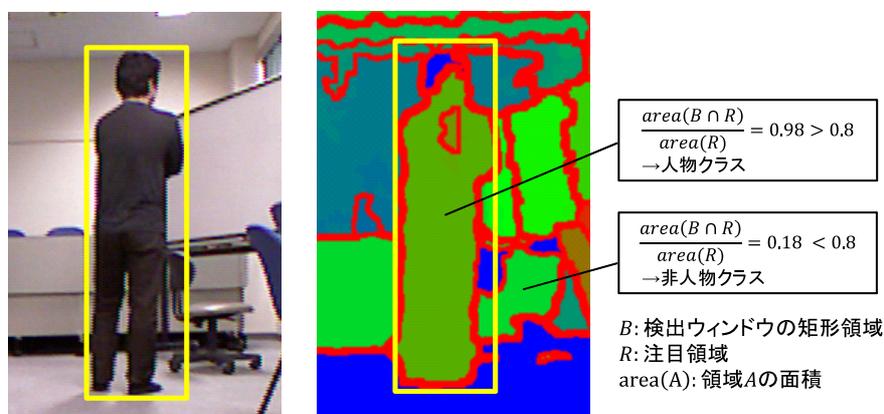


図 3.10: 各領域の正解クラスの算出方法

Step 3. 人物と分類された領域の統合

統合される候補の領域を選択する範囲および，算出されたウィンドウの絞り込みに関するパラメータを表 3.7 に示す．座っている人物は立ち姿の人物より幅が広くなるため，座っている人物も検出できるように W_m を広めに設定した．また， H_m は立ち姿の人物を検出できるように立ち姿の人物の一般的な高さに設定した．同様に，ウィンドウの絞り込みに用いる，基準とする人物の幅 W_h ，高さ H_h にも，それぞれ W_m ， H_m の値を用いた．

また，EDEN_{s3} の設定を表 3.8 に示す．EDEN_{s3} では，入力ノードを 2 つ用意し，それぞれ入力データとして統合する候補の領域を，参照入力データとして注目領域を割り当てた．出力ノードは「統合する」，「統合しない」に対応する 2 つである．また，中間ノードとして用いたノードおよび使用した特徴量は EDEN_{s2} と同じものを用いた．このため，距離を取得することができなかった領域の考慮は Step 2 と同様に行った．なお，中間ノードの最大数，世代数，突然変異率および評価関数における再現率の割合は複数のパラメータを変えて予備実験を行うことで決定した．

学習における算出ウィンドウの統合のパラメータ設定を表 3.9 に示す．Mean-Shift Vector 算出におけるバンド幅は x ， w については取得した画像の幅である 570 pixel で割ることで， y ， h については取得した画像の高さである 440 pixel で割ることで $[0,1]$ に正規化した．なお， b_x ， b_y ， b_w ， b_h および θ_a は複数のパラメータを変えて予備実験を行うことで決定した．

表 3.7: Step 3 の統合とウィンドウ算出の各種パラメータ設定

内容	値
統合する領域の候補を選択する範囲の矩形の幅 W_m	1.35[m]
統合する領域の候補を選択する範囲の矩形の高さ H_m	1.8[m]
範囲内の領域の面積に対するウィンドウと領域の重なる領域の割合のしきい値 θ_b	0.8
ウィンドウの絞り込みの基準となる人物の幅 W_h	1.35[m]
ウィンドウの絞り込みの基準となる人物の高さ H_h	1.8[m]
ウィンドウの絞り込みの最小の面積の割合 m_h	0.25
ウィンドウの絞り込みの最大の面積の割合 M_h	1.25

表 3.8: 領域の統合の判断に用いる EDEN_{s3} の設定

内容	値
入力ノード数	2 (入力：統合の判断を行う領域, 参照入力：統合される側の領域)
中間ノードの最大数	100
出力ノード数	2 (統合する, 統合しない)
ノードの種類	2 種類 (絶対値比較, 差分値比較)
世代数	1,000,000
突然変異率 P_m	0.02
使用した特徴量	66 種類 (RGB 色空間および距離画像：4 ÷ 11 種類, 点群データ：17 種類, 領域：5 種類)
適応度関数における再現率の割合 α	0.6

比較手法の各種設定

ここで、比較手法として用いる距離情報に基づく局所特徴量による人物検出 [44]について述べる。比較手法は距離画像から距離ヒストグラム特徴量を算出し、これを弱識別器として Real Adaboost を用いて特徴量の選択を行うことで人物検出を行う。さらに、オクルージョンに対応した識別を行うために距離情報を用いてオクルージョン領域を抽出し、識別対象の領域の面積に対するオクルージョン領域の割合を用いて識別器の出力を変化させている。また、検出は 3 次元実空間においてウィンドウのラスタスキャンを行う。このため、わずかなずれやスケール変化でも検出され、検出対象付近に検出ウィンドウが多く出力される。そこで、Mean-Shift クラスタリングを行うことで検出ウィンドウを統合する。その際、画像空間におけるクラスタリングだけでは人物の重なりがある場合に誤った統合がされてしまうことがあるため、3 次元実空間でのクラスタリングを行い、検出ウィンドウの統合を行う。このように距離情報を有効に利用することによって、人物の重なりに頑健に検出することができている。

比較手法ではウィンドウのラスタスキャンで人物を検出するため、ウィンドウのサイズが決まったものしか検出できない。比較手法では立ち姿の人物を対象としているため、3 次元実空間における 0.6 ÷ 1.8[m] のウィンドウ 1 種類だけを用いている。しかし、本実験で検出対象とする人物は立

表 3.9: 学習における算出ウィンドウの統合のパラメータ設定

内容	値
Mean-Shift Vector 算出における x のバンド幅 b_x	0.04
Mean-Shift Vector 算出における y のバンド幅 b_y	0.04
Mean-Shift Vector 算出における w のバンド幅 b_w	0.04
Mean-Shift Vector 算出における h のバンド幅 b_h	0.04
Mean-Shift Vector のしきい値 θ_m	10^{-5}
Nearest-Neighbor 法における統合のしきい値 θ_a	0.04

表 3.10: 各サイズのウィンドウのブロック分割における分割数

ウィンドウサイズ	分割数
$0.6 \div 1.8[\text{m}]$	$5 \div 10$
$0.7 \div 1.4[\text{m}]$	$6 \div 11$
$1.2 \div 1.4[\text{m}]$	$10 \div 10$

ち姿の人物に加え、姿勢の異なる人物および人物全体が写っていない人物を含む、すべての人物である。このため、ウィンドウのサイズを立ち姿の人物の1種類だけしか用いないと、姿勢が異なる人物や人物全体が写っていない人物を検出することができないと考えられる。そこで、本実験では立ち姿の人物のウィンドウに加え、 $0.7 \div 1.4[\text{m}]$ および $1.2 \div 1.4[\text{m}]$ のウィンドウを用いた。これらのウィンドウのサイズは、複数のサイズを変えて学習データの抽出を行い、出現頻度の高かったサイズを採用した。

これらのウィンドウから距離ヒストグラム特徴量を算出する際のブロック分割における分割数を表 3.10 に示す。分割されたブロックは、ウィンドウの幅を横の分割数で割った長さを一辺とする正方形である。分割数は、最も遠い距離の人物のウィンドウを画像空間に写像した際の1つのブロックのサイズが $8 \div 8$ pixel 以上になるように設定した。また、 $0.6 \div 1.8[\text{m}]$ のウィンドウでは比較手法においても、人物の上半身およそ6割の特徴を用いているため同様の設定にした。他の2種類のウィンドウでは、姿勢や写っている部分が一定ではないため、ウィンドウのほぼ全体から特徴量を算出した。

これら3種類のウィンドウサイズにおいてそれぞれ識別器を学習し、3つの識別器によって出力された検出ウィンドウから検出率を算出した。それぞれのウィンドウサイズにおける人物の学習データは、適用する際と同様に3次元実空間を奥行きを変えながらラスタスキャンを行い、正解データのウィンドウと切り出されたウィンドウの和集合の面積に対する積集合の面積の割合が $0.6 \div 1.8[\text{m}]$ のウィンドウでは0.85以上、他の2種類のウィンドウサイズでは0.75以上であるものの中からランダムに表 3.11 に示す学習データの数だけ選択した。また、非人物の学習データは切り出されたウィンドウの中で、正解データのウィンドウとの重なりがないものからランダムに選択した。

また、比較手法におけるその他のパラメータ設定を表 3.12 に示す。Real Adaboost における弱識別器の数、3次元実空間におけるラスタスキャンの奥行きのみ幅、および検出ウィンドウからどの程度手前のものをオクルージョンとみなすかの距離は文献 [44] と同じ設定にした。なお、Mean-Shift クラスタリングを行った後のウィンドウの統合には提案手法と同様に最近傍法を用い

表 3.11: 各サイズのウィンドウの学習データの数

ウィンドウサイズ	人物の学習データ数	非人物の学習データ数
0.6 ÷ 1.8[m]	250	750
0.7 ÷ 1.4[m]	250	750
1.2 ÷ 1.4[m]	100	300

表 3.12: 比較手法の各種パラメータ設定

内容	値
Real Adaboost における弱識別器の数 T	500
3次元実空間におけるラスタスキャンの奥行き のきざみ幅	0.3[m]
検出ウィンドウからどの程度手前のものをオクルージョンとみなすかの距離 d_z	0.3[m]
Mean-Shift Vector におけるバンド幅 h	0.3[m] (文献 [44]の統合方法), 0.05 (提案手法と同様の統合方法)
Mean-Shift Vector のしきい値 θ_m	10^{-5}
Nearest-Neighbor 法における統合のしきい値	0.3[m] (文献 [44]の統合方法), 0.05 (提案手法と同様の統合方法)

た. また Mean-Shift クラスタリングは, 文献 [44]と同様に 3次元実空間における検出ウィンドウの3次元座標を用いたクラスタリングと提案手法と同様に画像空間における検出ウィンドウの中心座標, 幅および高さを用いたクラスタリングの2種類を行い, それぞれを比較手法として用いた.

3.3.2 提案手法の各 Step の結果に対する考察

本節では, 提案手法の各 Step の結果を示し, その考察を行う. ここでは, 図 3.11 に示す 2つのフレームのデータを用いて考察する.

Step 1. 距離情報を用いた領域分割

図 3.12 に距離情報を用いた領域分割結果の例を示す. 図 3.12 において, 黄色の線で囲まれた領域が過分割された領域である. また, 黒い領域は距離を取得することができなかった領域, 9[m]より遠い領域, および地面の平面に属する領域であり, Step 2 における人物・非人物分類の対象としない領域である. 距離によるクラスタリングを用いて領域分割を行ったため, 人物と椅子などの物体それぞれが 1つの領域, ないしは数個の領域に過分割されている. 人物において過分割されたものは上半身や片手片足, 頭部付近の領域といった体の部分ごとに分割されることが多かった. これは, 歩いたり手を振ったりしたときに距離が変わることが多いのは体の部分ごとであるためである. また, 壁や天井といった環境の一部は距離でクラスタリングすることで細長い領域となるため, Step 2 において領域の特徴を用いることで非人物と判断しやすいような領域分割が行われている. 距離でクラスタリングを行っているため, フレーム 69 における座っている人物の一部が椅子

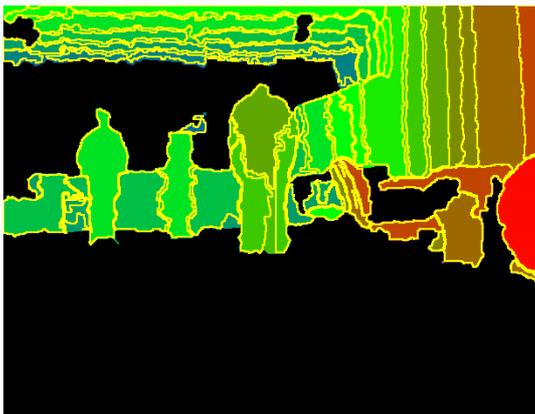


(a) フレーム 0

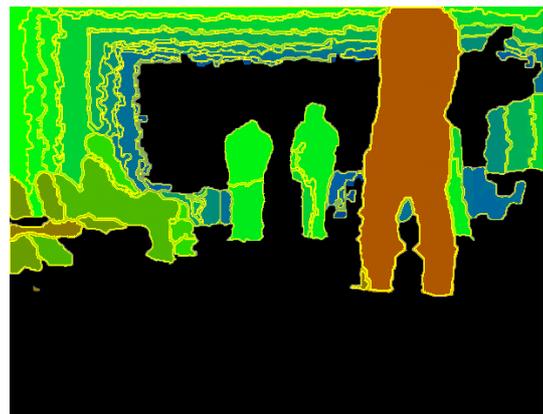


(b) フレーム 69

図 3.11: 考察を行うフレームのカラー画像



(a) フレーム 0



(b) フレーム 69

図 3.12: 距離情報を用いた領域分割結果の例



(a) フレーム 0

(b) フレーム 69

図 3.13: 過分割された領域の人物・非人物分類結果の例

と同じ領域になってしまっており、このような人物の領域と非人物の領域が同一の領域となることを抑制することが課題として挙げられる。

Step 2. 過分割された領域の人物・非人物分類

図 3.13 に図 3.12 と同じフレームの過分割された領域の人物・非人物分類結果を示す。図 3.12 と同様に黄色の線が過分割された領域の境界線である。緑で示された領域は正しく人物の領域を分類することができた領域である。また、赤で示された領域は非人物の領域を人物と誤分類してしまった領域である。フレーム 69 の結果からわかるように、姿勢の異なる人物やオクルージョンの存在する人物に対しても概ね良好な分類結果を得ることができた。しかし、フレーム 0 の赤で示された領域のように、非人物の領域が部分的に人物に特徴が似ている場合に誤分類される結果も見られ、これを抑制することが課題として挙げられる。

図 3.14 に 5 回試行の学習のうち最も分類精度が高かったネットワークを示す。赤、緑、青、および黄色の楕円のノードは、それぞれのノードが RGB 色成分、点群データ、領域、および距離画像の特徴量を用いていることを示す。また、実線の楕円は分類対象領域の特徴量の大きさで分岐する絶対値比較ノードを示し、破線の楕円は分類対象領域と参照入力データである分類対象領域に接している領域を 1 つにまとめた領域との特徴量の差分値で分岐する差分値比較ノードを示している。色の特徴量だけでなく、点群データ、領域、および距離画像の特徴量もそれぞれ用いられており、これらすべての特徴量を有効に活用することができていることがわかる。分類対象領域の特徴量の絶対値だけを用いて分類を行うと、学習データに現れた人物の特徴に過学習してしまう恐れがある。しかし獲得したネットワークでは、分類対象領域とその周囲の領域との特徴量の差分を用いる差分値比較ノードも多く活用されているため、学習データへの過学習が抑制された分類ネットワークを得ることができていると考えられる。また、分類対象領域が通る可能性のあるノードにおいて、参照入力データも通る可能性があるノードが多く見られた。このネットワークを用いて処理の変更を無効にして分類を行ったところ、処理の変更を有効にした場合と比べて分類精度の低下がみられた。このため、EDEN の特徴である多入力による処理の変更が有効に働いていることを確認することができた。

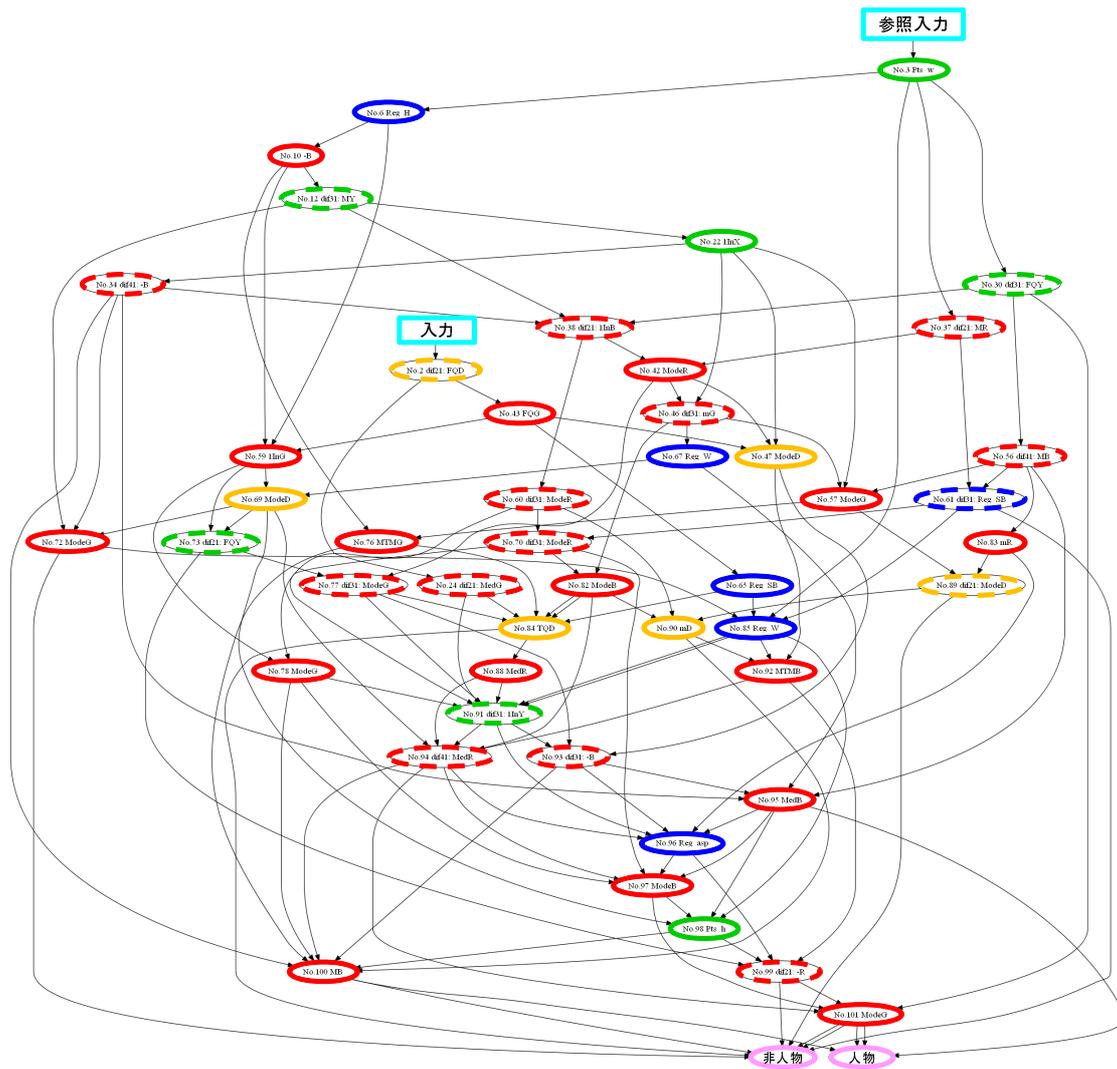


図 3.14: 獲得した人物・非人物分類ネットワーク

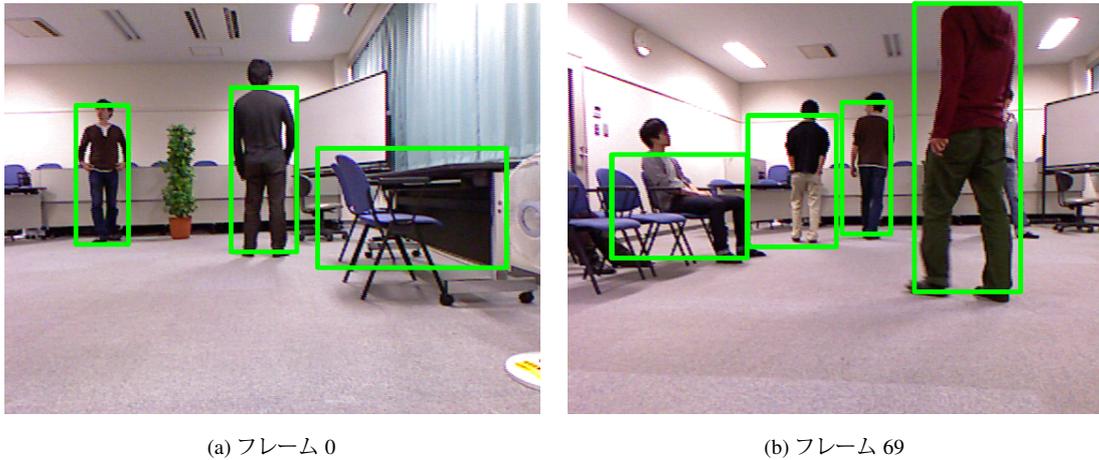


図 3.15: 人物と分類された領域の統合による検出ウィンドウの出力例

Step 3. 人物と分類された領域の統合

図 3.15 に図 3.12 と同じフレームにおいて、人物と分類された領域の統合を行い、検出ウィンドウを出力した結果を示す。立ち姿の人物だけでなく、座っている人物などの姿勢の異なる人物や画像の端や人物・物体によるオクルージョンで人物全体が写っていない人物に対しても、良好に検出することができた。フレーム 0 における左側の人物とフレーム 69 における右から 2 番目の人物は、人物全体が 1 つの領域として人物と分類されたため、他の領域と統合せずに検出ウィンドウを出力している。フレーム 0 における右側の人物とフレーム 69 における左から 2 番目の人物は、複数の領域に分割されているが領域の統合によって良好に検出することができている。また、フレーム 69 における座っている人物については頭部付近の領域を統合することはできなかったが、他のその人物を構成する領域を統合することができ、良好に検出することができた。本手法では、Step 2 において人物と誤分類してしまった領域を、Step 3 の領域の統合で検出性能を補うことを期待している。具体的には誤検出してしまった場合は、その領域は他の領域と統合せずに検出しないようにし、検出漏れはその他の検出することができた領域を統合することで検出することを期待している。フレーム 0 の左側の人物やフレーム 69 の小さな誤検出に対して、この期待通りに検出性能を補うことができた。しかし、フレーム 0 における椅子の部分とフレーム 69 における一番右のオクルージョンの存在する人物については、それぞれ誤検出、検出漏れとなった。これらはフレーム 0 では統合で検出を抑制できなかった例であり、フレーム 69 は逆に検出すべき人物を検出できなかった例である。このような誤検出、検出漏れの抑制が課題として挙げられる。

図 3.16 に 5 回試行の学習のうち最終的な検出ウィンドウの正答率が最も高かったネットワークを示す。各ノードで用いられている特徴量の種類は、図 3.14 における楕円の色と同様であり、ノードの種類も同様に実線が絶対値比較ノードであり、破線が差分値比較ノードである。統合する候補の領域が通る可能性のあるノードは、ほとんどが差分値比較ノードであり、統合する候補の領域と注目領域との特徴量の関係性を用いて判断を行っている。また、大部分が色の特徴量の差分を用いて判断しているが、一部に領域の外接矩形の高さの差分と点群データにおける y の値も用いている。最初に判断を行う領域の外接矩形の高さの差分を用いたノードでは、参照入力である統合される側の領域の方が統合する候補の領域よりも高さ大きい場合、統合しないと判断されている。これは、人物全体が 1 つの領域となっている場合に有効に働いていると考えられる。そのノードで統

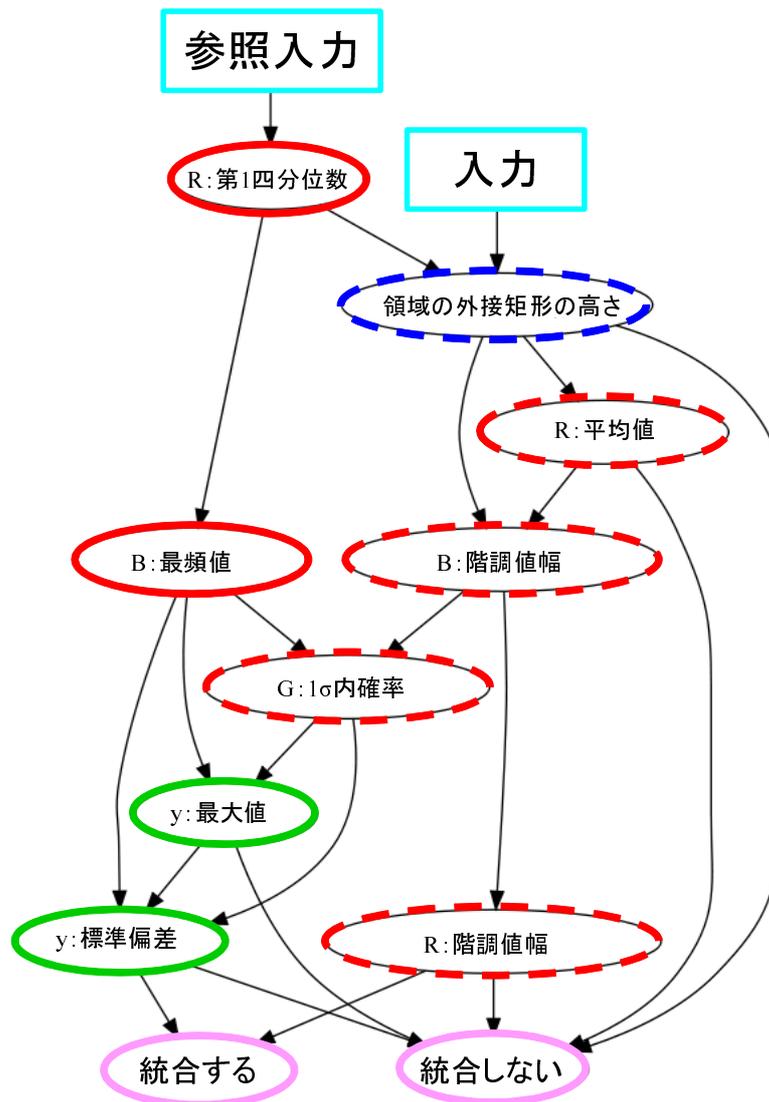


図 3.16: 獲得した領域の統合の判断を行うネットワーク

合しないと判断されなかった場合は、色の特徴量の差分と y の値を用いて統合するかどうかの判断を行っている。

3.3.3 提案手法および比較手法の人物検出結果とその考察

提案手法と比較手法の検出率・誤検出率による性能比較

ここでは、提案手法と比較手法の定量的比較として、検出率・誤検出率による性能比較を行う。本実験では、検出率に再現率 $recall$ を用い、誤検出率に画像 1 枚あたりの誤検出数 (False Positives Per Image; FPPI) を用いた。 $recall$ は、正解した検出ウィンドウ数 R および、すべての検出すべき人物のウィンドウ数 C_a を用いて、式 (3.22) によって算出される。

$$recall = \frac{R}{C_a} \quad (3.22)$$

図 3.17 に、提案手法と比較手法の検出率・誤検出率の推移を示す。提案手法、比較手法ともに、あるパラメータを変化させることで FPPI を変化させた結果を得ることができる。FPPI を変化させるパラメータとして、提案手法では Step 3 における検出ウィンドウの統合のパラメータである θ_m および θ_a を用いた。また比較手法では、Real Adaboost で獲得される識別器の出力が実数であるため、識別器の出力で人物とするしきい値を変化させることで FPPI を変化させた。検出率・誤検出率は一方を上げれば一方が下がるトレードオフの関係にあるため、これらを変化させた推移を見ることで性能を比較することができる。図 3.17 において、縦軸に recall、横軸に FPPI をとる。このため、誤検出を抑えたい一方でより多くの人物を検出するほど高い性能であるといえるため、recall-FPPI のカーブが左上に近づくほど性能が高い。提案手法は学習データ、テストデータともに比較手法をより左上に位置しているため、比較手法より高い性能を示した。比較手法である文献 [44] では、検出ウィンドウの統合において 3 次元実空間での Mean-Shift クラスタリングを行っている。しかし、本実験で用いたデータでは人物同士の重なりが多くは見られなかったため、検出ウィンドウの誤った統合を抑制する働きより距離が異なる複数のウィンドウを検出してしまふことで誤検出が増加してしまった。このため、本実験で用いたデータにおいては学習データ、テストデータともに提案手法と同様の検出ウィンドウの統合方法を用いた方が性能が高かった。また、比較手法はあらかじめ決定したサイズのウィンドウで検出を行う。このため、1 種類のサイズのウィンドウだけでは本実験の検出対象とする姿勢のさまざまな人物を検出することができない。そこで本実験では 3 種類のサイズのウィンドウを用いた。しかし、比較手法では学習データ、テストデータともに 1 種類のサイズのウィンドウを用いた検出結果の方が、3 種類すべてのサイズのウィンドウを用いた結果より性能が高かった。これは、図 3.18 からわかるように、ウィンドウのサイズを変えることでさまざまな姿勢の人物を検出することができるが、一方で誤検出も増加してしまうことが性能を下げた原因として考えられる。人物の姿勢の違いが多様であるため、人物ではない物体の形状にも同様の特徴が見られ、比較手法において誤検出が増加してしまったと考えられる。この結果から、人物の部分的な領域の姿勢の違いによるパターンの変化の方が人物全体のパターンの変化よりも少ないという仮定が当てはまり、提案手法の部分的な領域で人物を検出することが有効に働いたため、検出性能を向上させることができたと考えられる。

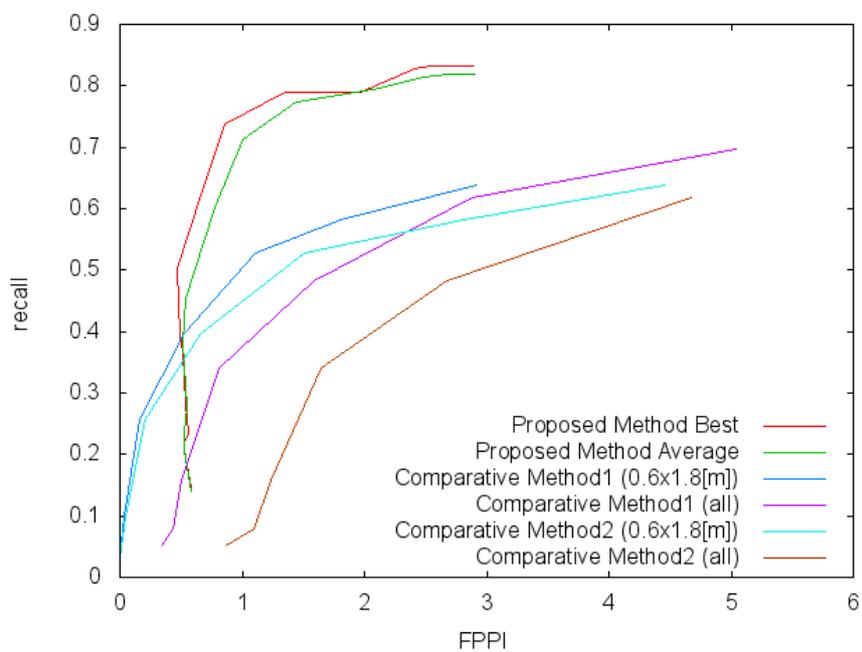
図 3.19 にテストデータにおける人物の姿勢・隠れごとの検出率・誤検出率による検出性能を示す。立ち姿は姿勢の変化が少ないため、立ち姿以外に比べて高い検出性能となった。検出性能が最も高かったのは立ち姿（隠れあり）であったが、これは人物が近い場合に、人物が入りきらず全体が映っていない場合により検出性能を示していた。これは、人物全体が映る場合には距離が遠くなり、距離情報が劣化してしまうため、人物全体の方が検出性能が下がってしまったと考えられる。立ち姿以外は主に座っている人物であったが、人物の部分領域を検出することで、立ち姿には劣るものの、隠れに頑健に立ち姿以外の人物を検出することができたと考えられる。

学習データおよびテストデータにおける検出ウィンドウの出力結果

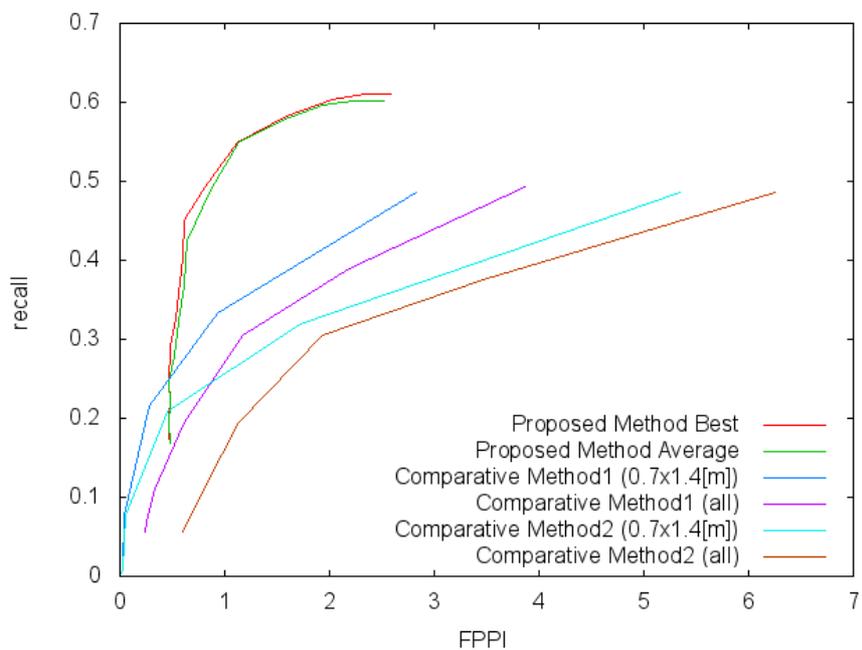
ここでは、出力された検出ウィンドウの結果を示し、その考察を行う。

←学習データにおける検出ウィンドウの出力結果

図 3.20 に、学習データにおける提案手法および比較手法の検出ウィンドウの出力結果例を示す。図 3.20 は提案手法、比較手法ともに FPPI がおよそ 1.0 の場合の結果である。図 3.20(a) は提案手法の学習データにおける結果である。図 3.20(b) は、比較手法の設定の中で最も性能の高かった、 $0.6 \div 1.8$ [m] のウィンドウだけを用いて提案手法と同様の検出ウィンドウの統合を



(a) 学習データに対する結果



(b) テストデータに対する結果

図 3.17: 検出率・誤検出率による性能比較

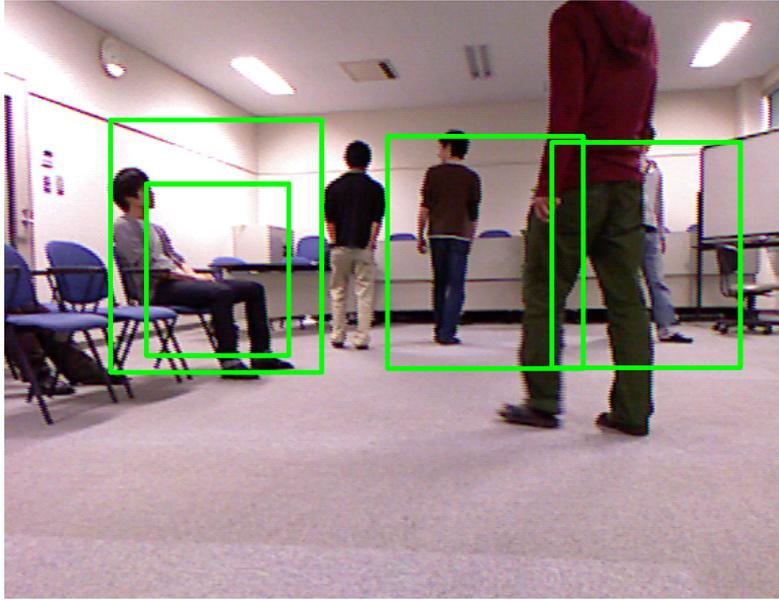


図 3.18: 比較手法において $1.2 \div 1.4$ [m] のウィンドウを用いた検出結果の例

行った結果である。比較手法ではウィンドウのサイズの種類を増やすことでより多くの人物を検出することができたが、一方で誤検出も増えてしまう結果となり、結果的に $0.6 \div 1.8$ [m] のウィンドウだけを用いたものが学習データにおいては性能が高かった。このため、ここでは比較手法として $0.6 \div 1.8$ [m] のウィンドウだけを用いたものの考察を行う。提案手法、比較手法ともに、フレーム 0, 95 からわかるように全体の写っている立ち姿の人物を良好に検出することができている。立ち姿の人物でもフレーム 27, 69 における頭部が写っていない人物のような体の一部が欠けている人物に対して、提案手法では良好に検出することができているが、比較手法では検出漏れが見られた。これは、比較手法の学習において $0.6 \div 1.8$ [m] のウィンドウでは立ち姿の人物を重点的に学習したため、頭部付近の特徴を学習しやすく、頭部が欠けている場合はその部分の特徴量を用いることができずに検出することができなかったと考えられる。また、頭部のかけている人物を検出しようとした場合、ウィンドウの高さが小さくなるためウィンドウのサイズが変化してしまうと検出が困難になってしまっている。一方で、提案手法では頭部の領域の特徴だけでなく、その他の体の一部分でも人物の検出を行っているため、頭部以外の領域を用いることで検出することができたと考えられる。立ち姿の人物と姿勢が異なる人物であるフレーム 64, 69 における座っている人物は、提案手法では座っている人物の大半を囲うウィンドウを検出することができている。比較手法ではウィンドウのサイズが異なるため、フレーム 64 のように座っている人物の上半身だけを検出している。このため、人物全体を囲うウィンドウを正解とした場合は検出漏れとなってしまう。オクルージョンの存在する人物に対しては、比較手法でもオクルージョンの考慮はしているが、どの程度手前のものをオクルージョンとみなすかの設定に依存し、また、オクルージョンの部分が大きくなると検出が困難となってしまう。これに対して提案手法では、見えている部分だけで検出を行うため、フレーム 18 のようにオクルージョンの部分が大きい場合でも、肩から上の部分と足の部分を統合することで人物全体を覆うウィンドウを検出することができている。

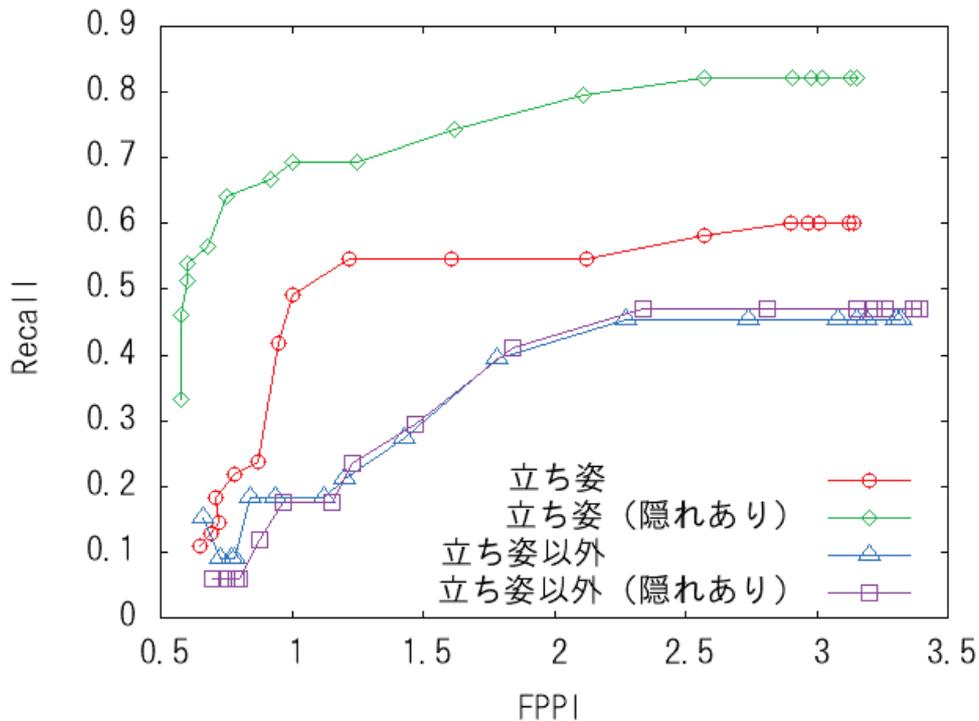


図 3.19: テストデータにおける人物の姿勢・隠れごとの検出率・誤検出率による検出性能

以上のように、提案手法は比較手法より多くの姿勢の異なる人物・オクルージョンの存在する人物を検出することができた。しかし、フレーム 0, 95 のように部分的に人物と検出され、領域の統合でも統合されてしまい誤検出してしまった例もあるため、このような誤検出を抑制することが課題として挙げられる。

←テストデータにおける検出ウィンドウの出力結果

図 3.21 に、テストデータにおける提案手法および比較手法の検出ウィンドウの出力結果例を示す。図 3.21(a) は提案手法のテストデータにおける結果である。図 3.21(b) は、比較手法の設定の中で最も性能の高かった、 $0.7 \div 1.4[m]$ のウィンドウだけを用いて提案手法と同様の検出ウィンドウの統合を行った結果である。テストデータにおける検出結果は提案手法、比較手法ともに誤検出、検出漏れが増加した結果となった。これは、提案手法、比較手法ともに学習ベースの手法であるため、テストデータに学習データとは異なる傾向が表れた場合は検出することが困難であるためである。しかし、現実にはすべてのパターンを学習することが困難であるため、決められた数の学習データでもこのような誤検出、検出漏れを、より抑制することができるということが性能が高いといえると考えられる。本研究では、姿勢が変化した場合の人物全体の画像や距離情報のパターンの変化より、部分的な領域のパターンの変化の方が少ないと仮定しているため、人物の部分的な領域で人物を検出することで姿勢の違いに対応することを期待した。フレーム 54 に写っている人物のような姿勢は学習データにはほとんど現れなかったため、比較手法は検出することができなかったが、提案手法では人物の部分ごとの検出を行うため、この人物を検出することができた。また、フレーム 34 では人物が存

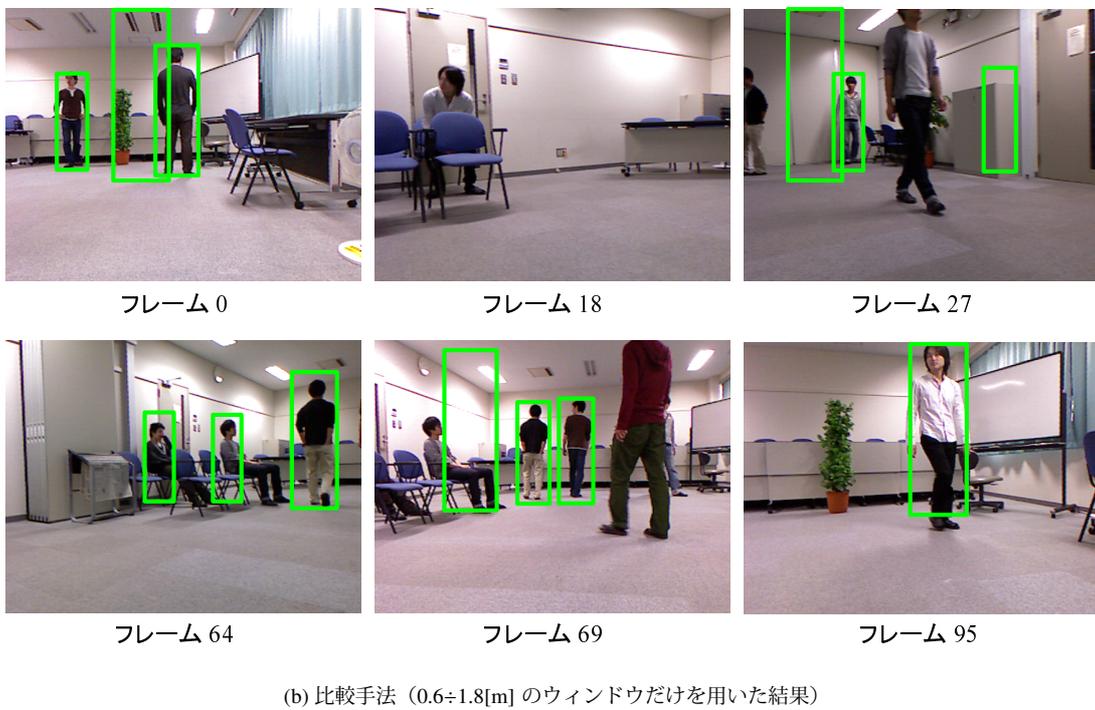
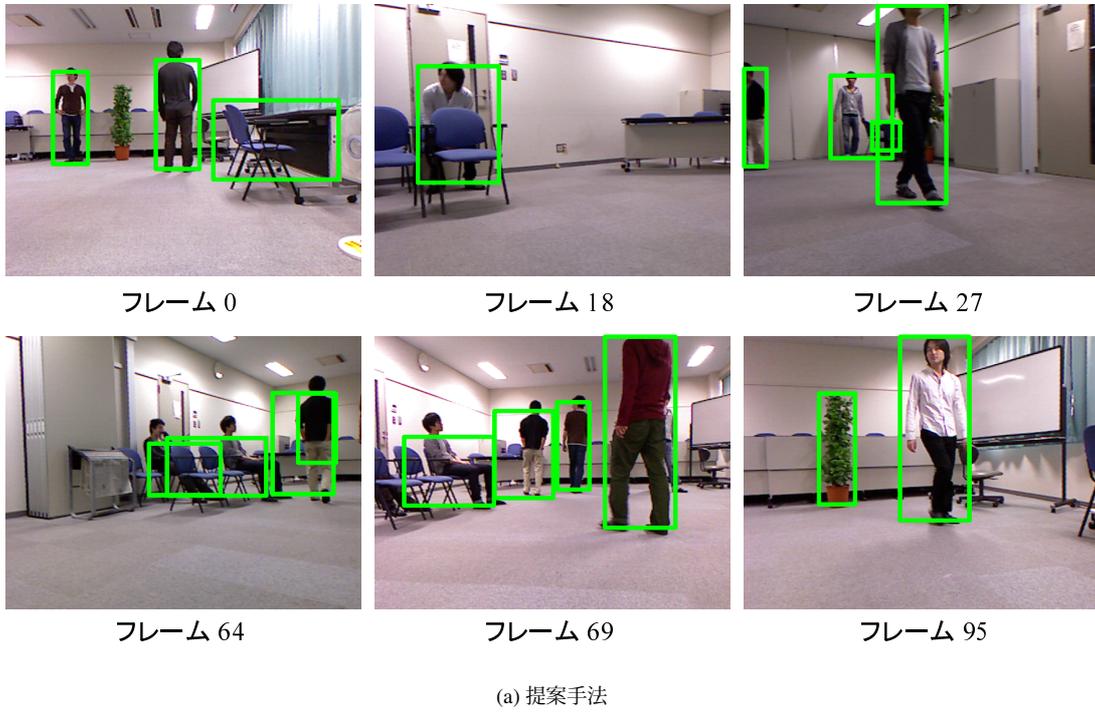


図 3.20: 学習データにおける提案手法および比較手法の検出ウィンドウの出力例

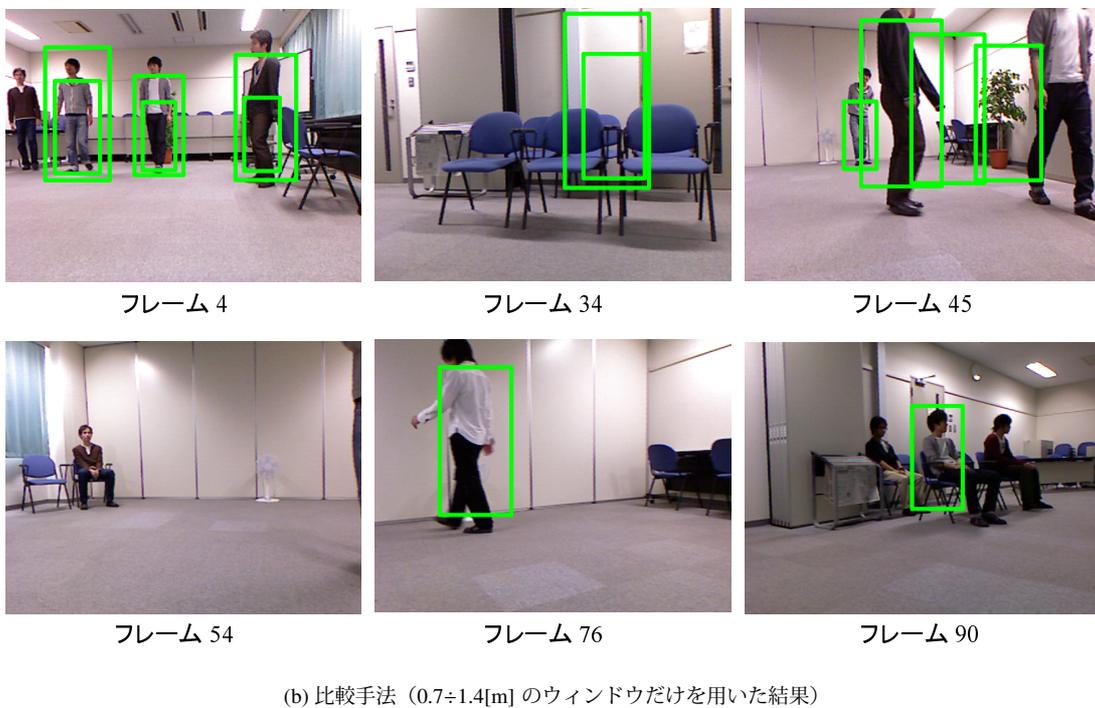
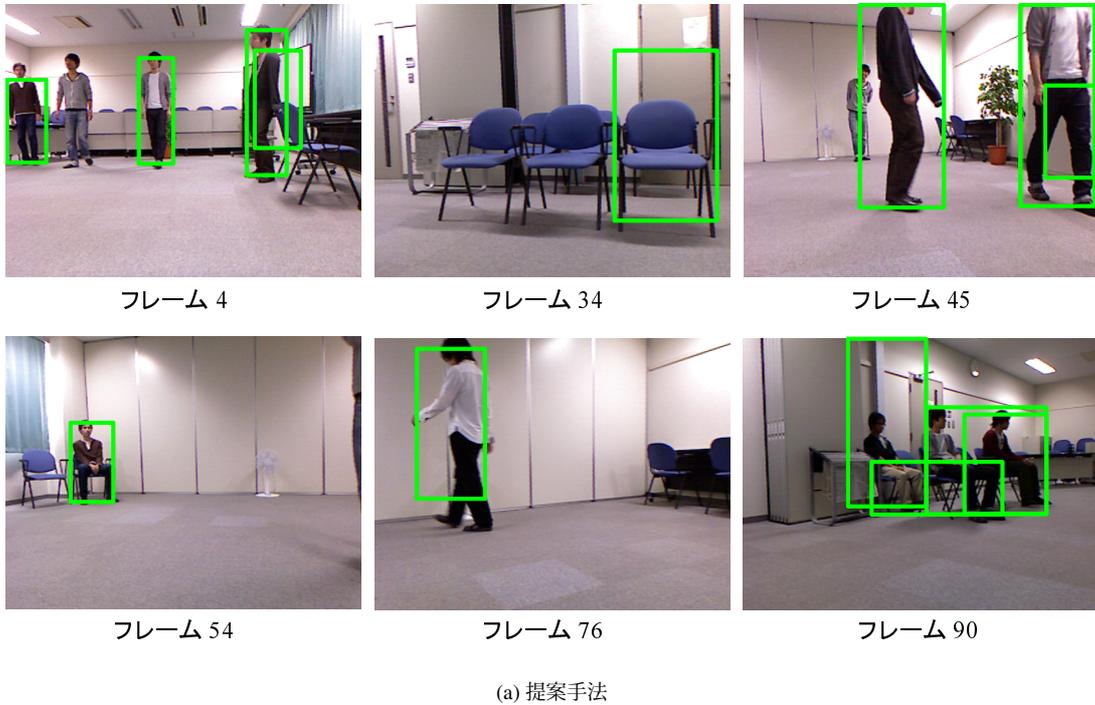


図 3.21: テストデータにおける提案手法および比較手法の検出ウィンドウの出力例

在しないにもかかわらず，提案手法，比較手法ともに誤検出してしまっている．これは，同じ環境を計測した場合でも計測する角度が異なったために人物に似た特徴が表れてしまったことが原因として考えられる．このような誤検出はどちらの手法でもあったが，提案手法の方が抑制することができた．提案手法では部分的に人物であるかどうかを検出するため，部分的に人物に似た特徴の領域を人物と検出してしまっても，領域の統合の際に周囲の領域との関係性で最終的に人物として検出しないようにすることができ，これが有効に働いたと考えられる．

以上のように，提案手法は比較手法より誤検出，検出漏れを抑制することができた．より誤検出，検出漏れを抑制することに加え，フレーム 90 における 2 人の人物の足の領域同士を統合してしまうような，異なる人物の領域を統合してしまうことで現れる誤検出を抑制することが課題として挙げられる．

3.4 まとめ

本章では，距離情報を用いて画像を人物の一部が切り出されるような過分割を行った後，それぞれの領域が人物の候補領域であるかどうかの分類を行い，人物の候補領域を統合することで姿勢変化や隠れに頑健な人物検出を行う手法を提案した．認識対象を空港などの巡回警備において撮影される人物として人物検出実験を行った結果，既存手法と比較して姿勢変化，隠れに頑健に人物を検出することができることを示した．

第4章 データの自己組織化に基づく特徴量の超低次元化

4.1 はじめに

第3章においては、高い認識性能を保つために認識対象の性質および画像以外の特徴量として距離情報を利用した。そして、ヒトにとって直観的な条件判断を行うネットワークを認識器に採用して特徴量を選択することで、低い計算コストと処理のわかりやすさの両立を図った。このアプローチによって、認識性能を向上しつつ計算コストと認識処理のわかりやすさを改善することができた。しかしながら、より一般的な画像認識を行う場合は、距離情報が必ずしもあるとは限らない。また、認識処理のわかりやすさを向上させる方法のひとつとして特徴量の可視化が挙げられるが、使用される特徴量が3次元より高くなると特徴量を可視化することができないという課題が残っている。そこで“高い認識性能”、“低い計算コスト”、“認識処理のわかりやすさ”を実現する2つ目のアプローチでは、特徴量を可視化することができる3次元以下へ低次元化（超低次元化）を行う。

本章では、データの自己組織化を用いることで与えられたデータから認識に有効な特徴量を創発することを期待し、データの自己組織化に基づく特徴量の超低次元化手法を提案する。提案モデルは、球面上にデータをランダムに配置し、データ同士の類似度を引力・斥力として表現して球面上を移動させることで自己組織化を行う。複数のデータセットについてデータ分布の可視化実験を行い、もとの特徴量を用いた場合のクラス分類結果と超低次元化された特徴量を用いた分類結果を比較することで提案手法の有効性を検証する。

4.2 球面上の移動に基づくデータの自己組織化

4.2.1 提案モデルの概要

図4.1に提案モデルの概要を示す。提案モデルではまず、すべてのデータを特徴量に関係なく球面上にランダムに配置する。次に、すべての2つのデータのペアについて、データの類似度を算出する。類似度が高ければそれらのデータがお互いに引き寄せられ、類似度が低ければお互いに遠ざかるように力のベクトルを算出する。すべてのペアについて力のベクトルを算出したら、それぞれのデータごとに合力のベクトルを算出し、合力に基づいて球面上を移動する。このデータ移動をあらかじめ定義した回数だけ繰り返すことで、データの自己組織化を行う。球面上でデータが自己組織化されるため、球面上の3次元座標もしくは緯度経度の2次元座標に超低次元化されたことになる。移動させる空間については、無限平面ではデータが発散してしまい、有限平面ではデータが端に集まってしまう。また、トーラスでは表面の密度が均一でないためにデータが線状に並んでしまう。このため、提案モデルにおいてデータを移動させる空間として球面を採用する。次に提案モデルの詳細について述べる。

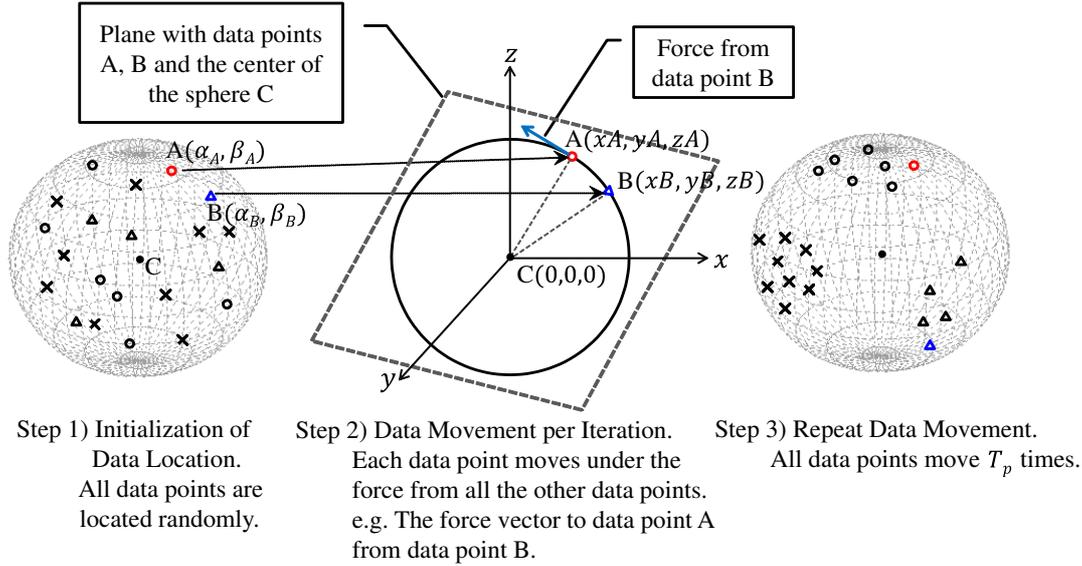


図 4.1: 提案モデルの概要

4.2.2 データ間の類似度に基づく球面上の移動

提案モデルでは、球の大きさを半径 1 とする。データ位置の初期化では、球面上の位置を緯度 α と経度 β で表現し、データのもつ特徴量に関係なく α および β をランダムに初期化する。次に、すべてのデータのペアについてそれらのデータに及ぼす力のベクトルを算出する。 t 回目の繰り返しにおいて、データ j がデータ i に及ぼす力の大きさを f_{ji}^t とすると、 f_{ji}^t は式 (4.1) で表される。

$$f_{ji}^t = |d_{ij} \delta_{ij}|, \quad (4.1)$$

ここで、 d_{ij} はデータ i, j 間の球面上の距離であり、 δ_{ij} はデータ i, j 間の入力特徴空間における距離である。 d_{ij} は式 (4.2) で表される。

$$d_{ij} = \frac{1}{\pi} \arccos(\sin \beta_i \sin \beta_j + \cos \beta_i \cos \beta_j \cos(\alpha_i - \alpha_j)), \quad (4.2)$$

ここで、 α_i, α_j および β_i, β_j はデータ i, j の緯度および経度である。また、提案モデルでは δ_{ij} を重み付きデータ間距離 (Weighted Inter-Point Distance; WIPD) として定義する。 δ_{ij} は式 (4.3) で表される。

$$\delta_{ij} = \frac{1}{1 + \exp(\gamma e_{ij} - \mu)}. \quad (4.3)$$

ここで、 e_{ij} はデータ i, j 間の入力特徴空間における平均二乗誤差であり、 γ は 1 以上の定数、 μ は $[0, 1]$ の定数である。式 (4.3) のように、データ間距離を強調する関数をかけることで、より類似するデータは近い距離となり、より異なるデータは遠い距離となる。

そして、 t 回目の繰り返しにおいて、データ j がデータ i に及ぼす力のベクトルを \mathbf{v}_{ji}^t とすると、 \mathbf{v}_{ji}^t は式 (4.4) で表される。

$$\mathbf{v}_{ji}^t = f_{ji}^t \mathbf{u}_{ij}^t, \quad (4.4)$$

ただし、 \mathbf{u}_{ij}^t はデータ i, j および球の中心で構成される平面による球の断面 (円) のデータ i を始点とする単位接線ベクトルである。ここで、 \mathbf{u}_{ij}^t は単位接線ベクトルであるため、互いに逆方向の 2

つのベクトルで構成される。このため、 $d_{ij} > \delta_{ij}$ の場合は球面上の距離よりも WIPD が小さいため、データ j へ近づく方向のベクトルとし、そうでない場合は、もう一方のベクトルとする。

このようにして、すべてのデータのペアについてそれらのデータに及ぼす力のベクトルを算出する。そして算出した力のベクトルを用いて、力を受けた結果のすべてのデータの位置を更新する。 \mathbf{x}_i^{t+1} を $t+1$ 回目の繰り返しにおけるデータ i の 3 次元空間における初期位置とすると、 \mathbf{x}_i^{t+1} は式 (4.5) で表される。

$$\mathbf{x}_i^{t+1} = \frac{\mathbf{V}_i^t}{\sqrt{y_i^t}}, \quad (4.5)$$

$$\mathbf{V}_i^t = \mathbf{x}_i^t + \frac{1}{N-1} \sum_{j=1, j \neq i}^N \mathbf{v}_{ji}^t, \quad (4.6)$$

ただし、 \mathbf{V}_i^t はデータ i 自身を除くすべてのデータの力のベクトルの合力であり、 N は全データ数である。また、 \mathbf{V}_i^t が球面上に乗るように $\|\mathbf{V}_i^t\|$ で割ることで、最大が球の半径となるようにする。最後に、 \mathbf{x}_i^{t+1} を (α_i, β_i) に変換する。

これらの処理を T_p 回繰り返すことで球面上でデータの自己組織化を行う。なお、これらの処理は並列して計算することができるため、Compute Unified Device Architecture (CUDA) [45] を用いて並列計算することで処理の高速化を図る。

4.3 特徴量の超次元化による可視化実験およびクラス分類実験

4.3.1 実験概要

提案モデルの有効性を示すため、クラス分類のデータセットを用いて特徴量を超次元化してデータ分布の可視化と超次元化した特徴量を用いたクラス分類実験を行った。データ分布の可視化実験では、可視化の先行研究である Self-Organizing Maps (SOM) と Multi-Dimensional Scaling のひとつであるサモンマップとの比較を行う。また、提案モデルにおいても WIPD を用いる場合と用いない場合について比較を行う。さらに、可視化結果の定量評価指標を提案し、定量的な比較も行う。

超次元化した特徴量を用いたクラス分類実験では、用意されているすべての特徴量を用いた場合の複数の分類器の分類性能と、超次元化した特徴量を用いた分類性能を比較する。また、提案モデルではデータを球面上に写像するため、超次元化された特徴量は球面上の 3 次元座標と緯度経度の 2 次元座標である。このため、3 次元座標と緯度経度それぞれの特徴量を用いた実験を行う。

4.3.2 実験データ

実験に用いたデータは UCI Machine Learning Repository [34] のデータセットである。表 4.1 に実験に用いた 10 種類のデータセットを示す。データ数は 150 から 1484、用意されている特徴量の次元数は 4 から 33、クラス数は 2 から 10 とさまざまなデータセットに適用した。

4.3.3 実験設定

まず、可視化実験の設定について述べる。提案モデルにおいては、式 (4.3) における γ を 7.0、 μ を 0.5 とし、繰り返し回数 T_p を 1000 とした。SOM においては、SOM_PAK [46] を用いた。表 4.2

表 4.1: 使用した UCI データセット

Datasets	#Examples	#Categories	#Attributes
Bupa	345	2	6
Pima	768	2	8
Wdbc	569	2	30
Ionosphere	351	2	33
Iris	150	3	4
Wine	178	3	13
Cleveland	297	5	13
Glass	214	6	9
Ecoli	336	8	7
Yeast	1484	10	8

表 4.2: SOM の設定

Parameters	Values
Map size	20 ÷ 20
Topology type of map	Hexagonal lattice
Number of iterations T_s	#Examples ÷ 10
Initial learning rate $\alpha(0)$	0.2
Learning rate function type	Linear function
Neighborhood function type	Gaussian function
Initial neighborhood radius $\sigma(0)$	10

に SOM の設定を示す。また、提案モデルとサモンマップはデータの初期位置をランダムに決定するため、それぞれのデータセットにおいて 5 回試行した。これらのパラメータは、予備実験において複数の組み合わせを試行し、それぞれの手法が最も高性能となったパラメータを採用した。

次に、クラス分類実験の設定について述べる。Support Vector Machine (SVM), 決定木 (C4.5), k 近傍法の 3 種類について WEKA (version 3.6.10) [47] を用いて分類を行った。SVM, C4.5 については WEKA のデフォルトの設定を用い、k 近傍法は $k = 3$ とした。また、クラス分類は交差検証 (3 フォールド) で評価を行った。

4.3.4 可視化結果の評価指標

可視化結果においては、同じクラスのデータがより密集しており、異なるクラスのデータがより離れて位置していることが、良好にデータを分離して可視化できているといえると考えられる。そこで、可視化結果の定量評価指標として、各クラスのデータが密集しているかを評価する Data Cohesion (DC) および、より 1 つのクラスタが 1 つのクラスのデータで構成され、他のクラスタと分離できているかを評価する Cluster Separation (CS) を提案する。DC が小さいと同じクラスのデータが密集しているといえるため、小さいほどよい評価である。CS は大きいと異なるクラスのデータが分離されていることを表すため、大きいほどよい評価である。したがっ

て、 CS/DC が大きいとよい評価である。可視化結果の定量評価は DC 、 CS および CS/DC を比較することで行う。

次に DC と CS の詳細を述べる。 DC は同じクラスのすべてのペアのデータ間距離である。クラスごとにデータ数の偏りが存在するため、 DC はデータ数の偏りを考慮した加重平均とする。 DC を式 (4.7) に示す。

$$DC = \frac{\sum_{k=1}^K w_k c_k}{\sum_{k=1}^K w_k}, \quad (4.7)$$

$$c_k = \frac{1}{N_k C_2} \sum_{i \in I_k} \sum_{j \in I_k} \|\mathbf{v}_{ji}^{T_p}\|, \quad (4.8)$$

$$w_k = N_k / N, \quad (4.9)$$

ただし、 K がクラス数、 N_k がクラス k のデータ数であり、 I_k がクラス k のデータを表すインデックス集合である。

CS は異なるクラスのすべてのデータとの距離である。 DC と同様に、 CS はデータ数の偏りを考慮した加重平均とする。 CS を式 (4.10) に示す。

$$CS = \frac{\sum_{k=1}^K w_k s_k}{\sum_{k=1}^K w_k}, \quad (4.10)$$

$$s_k = \frac{1}{K} \sum_{l=1, l \neq k}^K \left(\frac{1}{N_k} \sum_{i \in I_k} \left(\frac{1}{N_l} \sum_{j \in I_l} \|\mathbf{v}_{ji}^{T_p}\| \right) \right). \quad (4.11)$$

4.3.5 可視化結果と考察

ここでは、10種のデータセットのうち Iris と Wdbc について結果を示し、考察する。その他のデータセットの提案モデル (WIPD あり) の可視化結果を付録 A に示す。図 4.2 に Iris の可視化結果を示す。図 4.2(a), 図 4.2(b), 図 4.2(c), 図 4.2(d) それぞれに、SOM, サモンマップ, 提案モデル (WIPD なし), 提案モデル (WIPD あり) の Iris の可視化結果を示す。SOM では、SOM のユニットのうち 1 つ以上のデータを含むユニットについて、含まれるデータの最も多いクラスを示している。サモンマップおよび提案モデルではすべてのデータについてそれぞれの空間にプロットした。また、SOM とサモンマップについては 2 次元平面のそれぞれの軸において、データの最小値および最大値を用いて $[0, 1]$ の範囲でマップを正規化した。

SOM の結果は、データがマップ全体に広がる結果となった。また、SOM はユニットが等間隔で配置されているため、同じクラスのデータを密集させるためには少ないユニットにデータを含める必要がある。しかし、SOM ではデータをマップ全体に一樣に分布させる傾向があるため、それぞれのクラスのデータが広がりをもった結果となった。サモンマップと提案モデルは類似したアルゴリズムであるため、見た目には類似した結果となった。クラス 2 とクラス 3 が近くに配置され、クラス 1 とは分離された結果となった。特に提案モデル (WIPD なし) の結果はサモンマップと類似した結果となった。一方で、提案モデル (WIPD あり) では、データの類似度を強調しているため、より同じクラスのデータが密集した結果となった。

図 4.3 に Wdbc の可視化結果を示す。図 4.3(a), 図 4.3(b), 図 4.3(c), 図 4.3(d) それぞれに、SOM, サモンマップ, 提案モデル (WIPD なし), 提案モデル (WIPD あり) の Wdbc の可視化結果を示す。SOM では、Iris と同様にデータがマップ全体に一樣に分布した結果となった。また、Iris と比較して 2 つのクラスの境界が近い結果となった。この傾向はサモンマップおよび提案モデルでも見

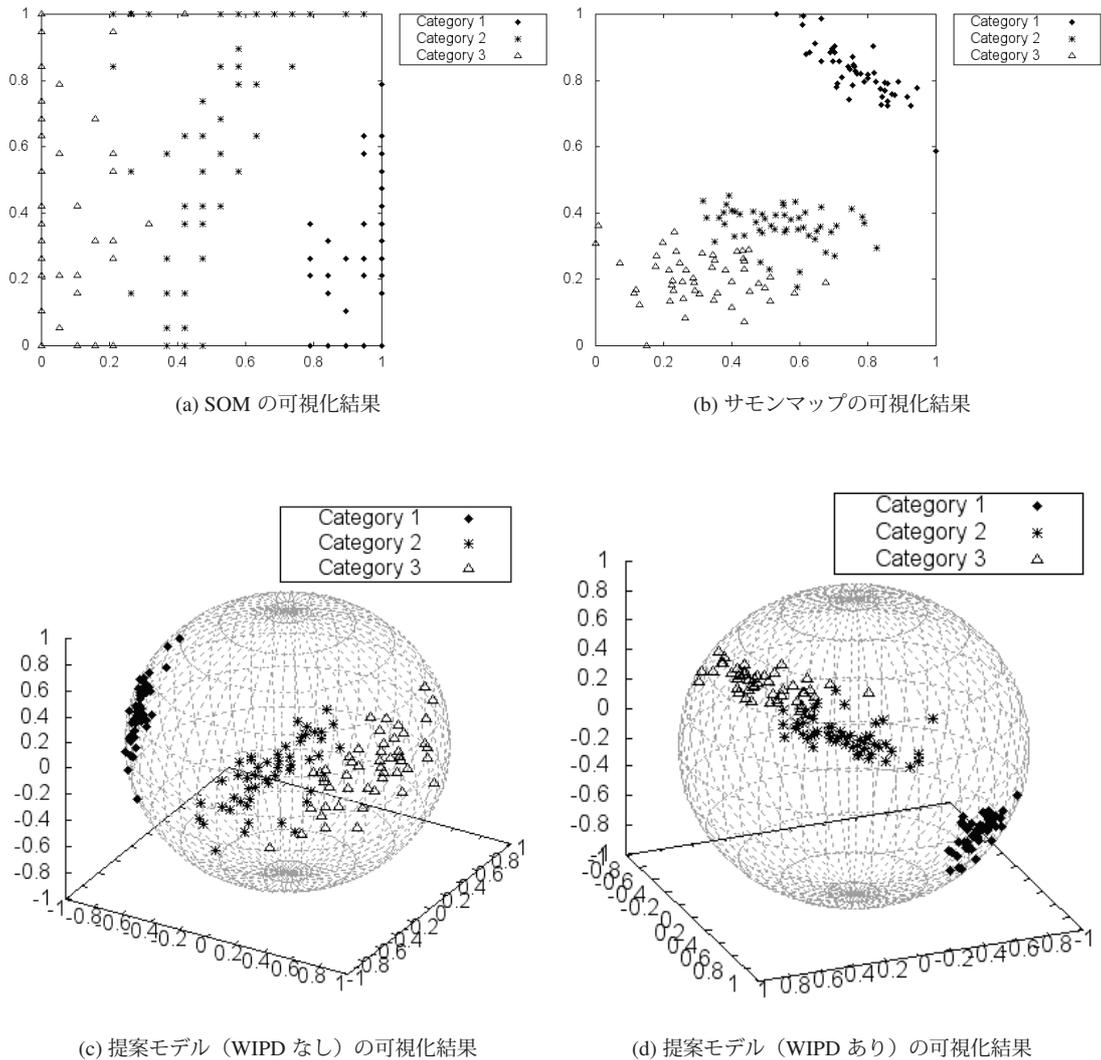


図 4.2: 各手法の Iris の可視化結果

られた。Iris と同様に提案モデル（WIPD あり）の結果が最も同じクラスのデータが密集した結果となった。

4.3.6 提案した評価指標による評価と考察

ここでは可視化結果について、4.3.4 項で提案した可視化結果の評価指標を用いて定量評価を行う。それぞれの手法の距離の範囲が異なるため、それぞれの手法において距離の範囲が $[0, 1]$ になるように最も遠い距離で割ることで正規化する。SOM とサモンマップでは一辺が 1 の正方形の空間であり、対角線が最も遠い距離となるため、 $\sqrt{2}$ で割ることで正規化する。提案モデルでは球面上で最も遠い距離は、球の反対側に位置する場合であり、球の半径が 1 であるため、 π で割ることで正規化する。

表 4.3 に DC 、 CS および CS/DC によるそれぞれの手法の比較結果を示す。 DC は低いほどよい

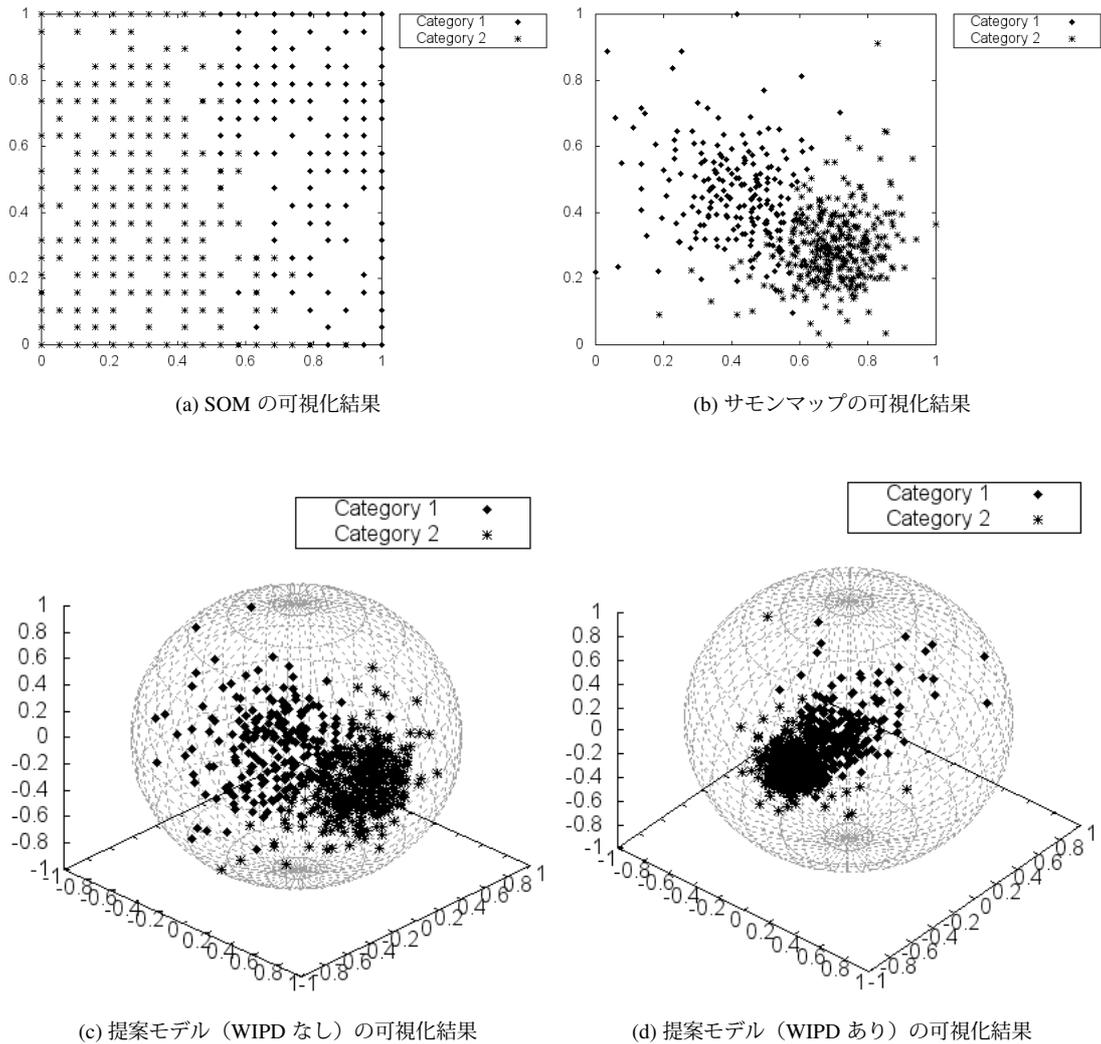


図 4.3: 各手法の Wdbc の可視化結果

性能であり、 CS は高いほどよい性能であるため、 CS/DC が高いほどよい性能である。表 4.3 では、もっともよい性能の数値を太字で示し、2 番目により性能の数値には下線を引いて示す。また、“Average”はすべてのデータセットの DC 、 CS および CS/DC の平均であり、“Ave. Rank”は4つの手法で順位付けをした場合のすべてのデータセットの順位の平均である。このため、“Ave. Rank”が1に近いほどよい性能であるといえる。

DC をみると、ほとんどのデータセットにおいて提案モデル (WIPD あり) が最もよい性能であり、提案モデル (WIPD なし) が2 番目により性能であった。このことから、球面を出力空間として用いることが同じクラスのデータを密集させることに有効に働いたことがいえる。これは、球面上で距離を算出することで空間の位置に依存せず偏りのない距離を算出することができたためであると考えられる。さらに、WIPD を用いることでよりデータを密集させることができたといえる。

一方で CS をみると、SOM がすべてのデータセットにおいて最もよい性能を示した。SOM では出力空間において一様にデータを分布させる傾向があり、これが異なるクラスのデータを分離することに有効に働いたためであると考えられる。2 番目により性能を示したのは、提案モデル (WIPD

表 4.3: 提案した可視化結果の評価指標による比較結果

Datasets	SOM			Sammon's mapping			Proposed_w/oWIPD			Proposed_w/WIPD		
	DC	CS	CS/DC	DC	CS	CS/DC	DC	CS	CS/DC	DC	CS	CS/DC
Bupa	0.414	0.208	0.502	0.189	0.094	0.499	<u>0.164</u>	0.082	<u>0.499</u>	0.104	0.052	0.498
Pima	0.402	0.219	0.546	0.203	<u>0.115</u>	0.569	<u>0.187</u>	0.106	<u>0.568</u>	0.119	0.067	0.566
Wdbc	0.313	0.249	0.796	0.139	<u>0.127</u>	0.916	<u>0.127</u>	0.115	0.910	0.080	0.073	<u>0.915</u>
Ionosphere	0.409	0.218	0.534	0.222	0.130	0.583	0.306	<u>0.180</u>	<u>0.589</u>	<u>0.270</u>	0.163	0.605
Iris	0.266	0.341	1.279	0.131	0.259	1.989	<u>0.127</u>	<u>0.272</u>	<u>2.131</u>	0.092	0.261	2.842
Wine	0.247	0.357	1.443	0.168	<u>0.245</u>	1.460	<u>0.140</u>	0.205	<u>1.464</u>	0.099	0.148	1.501
Cleveland	0.396	0.373	0.941	0.256	0.259	1.012	0.365	<u>0.371</u>	<u>1.016</u>	<u>0.357</u>	0.371	1.040
Glass	0.311	0.325	1.044	0.143	<u>0.179</u>	1.252	<u>0.137</u>	0.165	<u>1.201</u>	0.100	0.120	1.195
Ecoli	0.306	0.421	1.375	<u>0.145</u>	0.240	1.654	0.174	<u>0.302</u>	<u>1.737</u>	0.132	0.259	1.969
Yeast	0.308	0.406	1.318	<u>0.105</u>	0.165	1.572	0.111	<u>0.175</u>	<u>1.590</u>	0.065	0.117	1.791
Average	0.337	0.312	0.978	<u>0.170</u>	0.181	1.151	0.184	<u>0.197</u>	<u>1.171</u>	0.142	0.163	1.292
Avg. Rank	4.0	1.0	3.7	<u>2.4</u>	2.9	2.4	<u>2.4</u>	<u>2.5</u>	2.1	1.2	3.6	1.8

なし)とサモンマップであった。このことから、異なるクラスのデータを離すことについては出力空間を球面上にした場合と平面にした場合とで、データセットごとにどちらが有効であるかが異なるものの、同程度の割合でそれぞれが有効であるといえる。また、提案モデル(WIPDあり)が最も悪い性能となったことから、WIPDは特に同じクラスのデータを密集させることに有効であると考えられる。

DCとCSを合わせて評価するCS/DCについては、提案モデル(WIPDあり)が最もよい性能を示し、提案モデル(WIPDなし)が2番目により性能を示した。このことから出力空間を球面上にすることで、平均的にDCとCSをよくすることができ、WIPDを用いることで特にDCについて性能を向上させることで全体的な性能を向上させることができたといえる。一方で、WIPDの γ, μ についてはデータセットごとに有効な値が異なると考えられるため、 γ, μ の決定方法について検討する必要がある。

4.3.7 クラス分類による評価と考察

すべてのデータセットにおいて用意されているすべての特徴量を用いた場合と、提案モデルで超低次元化した特徴量を用いた場合についてクラス分類実験を行った。提案モデルではデータを球面上に写像するため、超低次元化された特徴量は球面上の3次元座標と緯度経度の2次元座標である。このため、3次元座標および緯度経度のそれぞれの特徴量を用いた結果と用意されているすべての特徴量を用いた結果を比較する。

図4.4, 図4.5, 図4.6それぞれに、SVM, C4.5, k近傍法のクラス分類結果の比較を示す。それぞれの図では、それぞれのデータセットの各クラスのF値のデータ数で重みを付けた平均を用いている。また、もとの特徴量を用いた結果から提案モデルによって超低次元化された特徴量を用いた結果を引いたものも示している。もとの特徴量との差分については、そのままの値と、それを次元数の差分で割った値(1次元あたりの差分)を参考に示している。全体として超低次元化された特徴量を用いた結果はもとの特徴量を用いた結果と比べて劣った結果となった。しかし、その差は少なく、削減した次元数を考慮すると良好な分類精度を得ることができたといえる。

球面上の座標を用いた結果については、図4.4をみると、SVMにおけるクラス分類結果において、およそ半数のデータセットでF値の差が大きくなった結果が得られた。一方で、図4.5, 図4.6をみると、SVMの結果と比べてF値の差が小さくなった結果が得られた。これは、特徴空間を線形に分割するC4.5と特徴空間の距離を用いるk近傍法では、自己組織化して超低次元化した特徴量ともとの特徴量に大きな差が見られなかったことに対し、SVMではカーネル関数をかけるため、より高次元である方が分類がしやすくなるためであると考えられる。

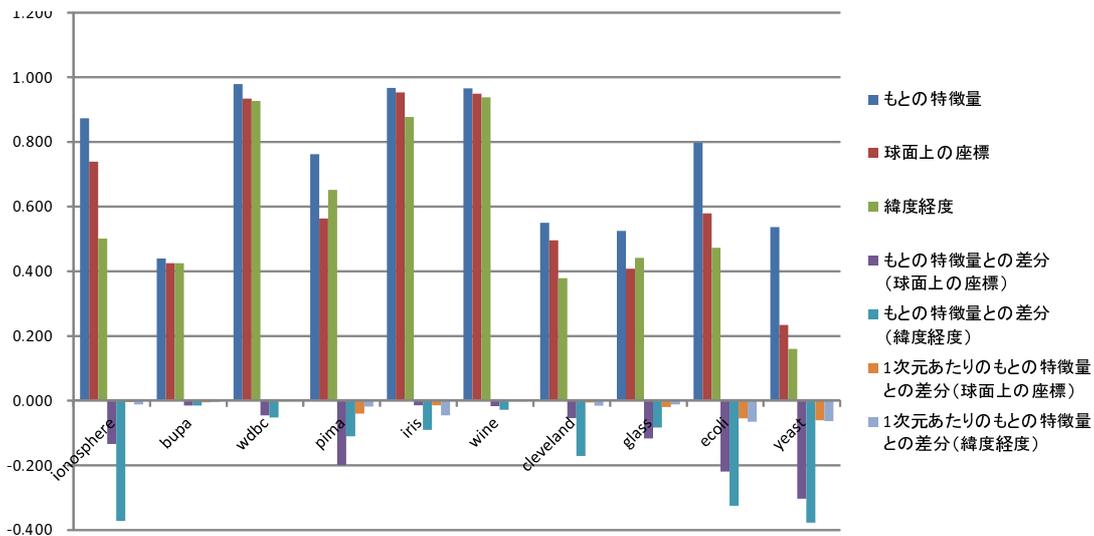


図 4.4: SVM におけるクラス分類結果の比較

緯度経度を用いた結果については、SVM では全体的に球面上の座標を用いた結果より F 値が下がった結果となった。これはさらに次元が下がった結果、より分類が難しくなったことが原因として考えられる。C4.5, k 近傍法については、球面上の座標を用いた結果とあまり変わらない結果となった。球面上の座標を用いた結果と同様の理由であると考えられる。また、いくつかの結果で緯度経度を用いた結果が球面上の座標を用いた結果より高精度になったことについては、球面上の座標で球の裏側同士のデータを x 軸または y 軸で区別することが難しい場合もあり、これを経度にすることで分離しやすくなったためであると考えられる。

最後に、1次元あたりのもとの特徴量を用いた結果との差分については、全体的にあまり変わらない結果となった。この結果から特徴量を減らす数が多いほど分類性能が下がってしまったことが考えられる。データの自己組織化では分類されるデータのそれぞれのクラスの分布を考慮せずに超低次元化するため、よりクラス分類に有効な特徴量に超低次元化するためには、各クラスの分布を考慮することが必要であると考えられる。

4.4 まとめ

本章では、球面上にデータをランダムに配置し、データ同士の類似度を引力・斥力として表現して球面上を移動させることでデータの自己組織化を行い、特徴量を超低次元化する手法を提案した。10種のデータセットを用いて超低次元化を行った結果、従来手法と比較してより同じクラスのデータを集め、異なるクラスのデータを分離させた可視化を行うことができることを示した。また、超低次元化された特徴量を用いたクラス分類実験の結果、用意された特徴量を用いた分類精度には劣った結果が多かったものの、削減した次元数を考慮すると良好な分類精度を得ることができた。

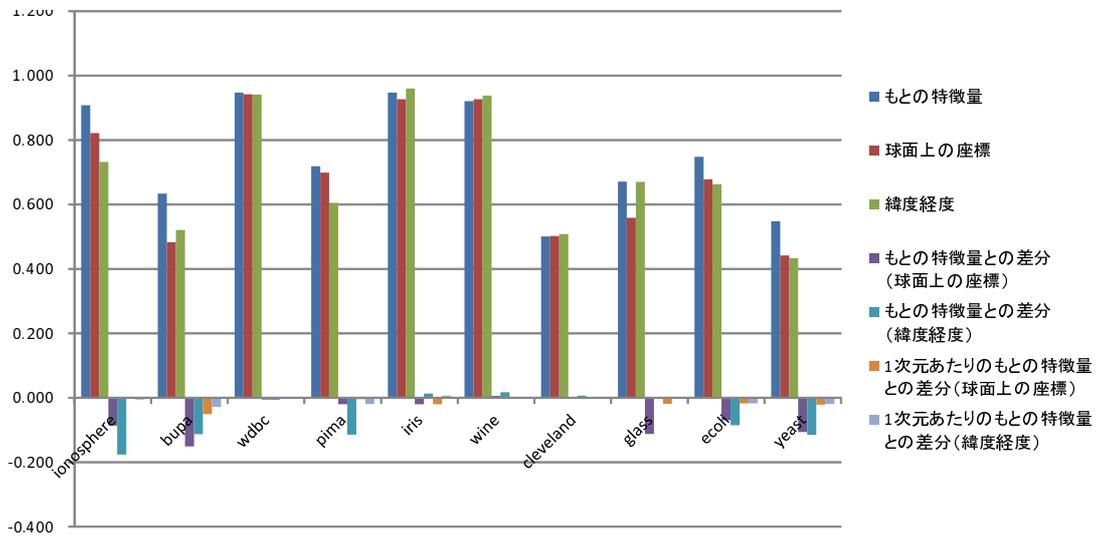


図 4.5: C4.5 におけるクラス分類結果の比較

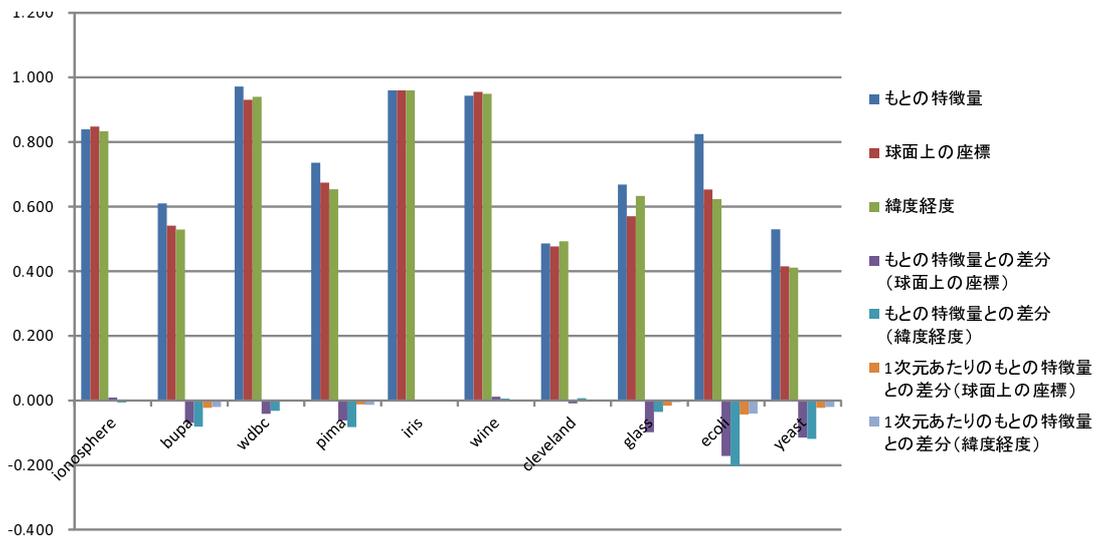


図 4.6: k 近傍法におけるクラス分類結果の比較

第5章 既存特徴量の組み合わせによる特徴量の階層的な超低次元化

5.1 はじめに

第4章では、データの自己組織化に基づく超低次元化手法を提案した。しかし、高次元の特徴量を用いたとしてもクラス内の多様性が高い画像では識別が困難である場合もある。そこで、それらを識別することができるような新たな特徴量を既存特徴量を組み合わせることで構築し、超低次元化を行う。本章では、木構造状の関数やプログラムを構築する遺伝的プログラミングのひとつである Cartesian Genetic Programming (CGP) を用いて階層的に特徴量を超低次元化する手法を提案する。第1層となる既存の高次元特徴量を、CGPによって最適化された関数を用いて変換することで新たな特徴量とする。新たな特徴量を構築する際、急激に3次元以下にしてしまうと高次元特徴量の情報が失われすぎてしまう恐れがある。このため、階層的に徐々に次元数を落としていくことで最終的に3次元に超低次元化する。クラス内の多様性が高い一般画像（自転車、人物）と医用画像（小腸画像）、およびクラス内の多様性が低い一般画像（歩行者）について対象クラスであるかどうかの2クラス分類実験を行い、特徴量の次元数を落としても分類精度を向上させることができることを示す。また、3次元に超低次元化した特徴量を用いてデータ分布を可視化するとともに、ヒトにとって直観的な条件判断を行う決定木を分類器に用い、処理フローの解析しやすさの考察も行う。さらに、使用された既存特徴量の数と、超低次元化した特徴量を用いた分類処理の高速化倍率の2つの観点から、特徴量算出コストと分類計算コストについて評価する。

5.2 CGPを用いた特徴量の階層的な超低次元化

図5.1に提案モデルの概要を示す。提案モデルは複数層の階層構造で構成される。第1層は既存特徴量であり、それらの特徴量をCGPの入力として組み合わせることで第2層の特徴量を構築する。このとき、特徴量は1次元ずつ構築され、それまでに構築された L 次元（本論文では $L=1$ ）の特徴量を用いた分類精度でデータの重みを更新することで、多様な特徴量の構築を目指す。現在の層における t 番目の特徴量のデータ i の重み w_i^t の更新式を次に示す。

$$w_i^{t+1} = \frac{1}{Z_{t+1}} w_i^t \exp(\alpha y_i^t). \quad (5.1)$$

ここで、 y_i^t はデータ i が正しく分類された場合に1、そうでなければ-1である。 $\frac{1}{Z_{t+1}}$ は正規化項であり、すべての重みの和が1となるように正規化する。また、 α は定数であり、本論文では1を用いた。なお、1つ目の特徴量を構築する際の重み w_i^1 はすべて $1/N$ （ N はデータ数）とする。

第2層であらかじめ定義した数の特徴量を構築したら、第3層の特徴量の構築を行う。第3層では、CGPの入力を第2層で構築した特徴量に限定し、それ以外は第2層と同様に特徴量を構築する。入力を前層で構築した特徴量に限定するのは、用いる既存特徴量の数を抑制するためである。これらの処理をあらかじめ定義した層数、次元数だけ繰り返すことで特徴量を超低次元化する。図

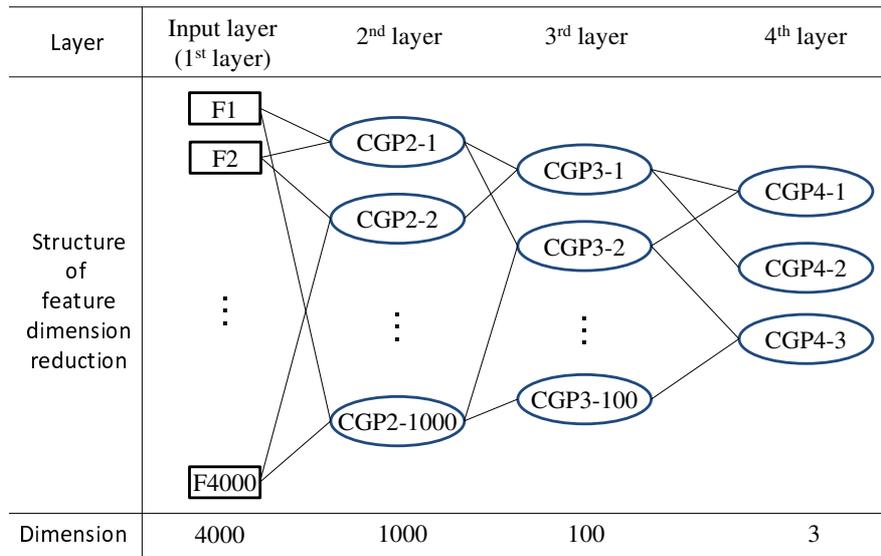


図 5.1: 提案モデルの概要

5.1 の例では、第 1 層の 4000 次元の既存特徴量から第 2 層で 1000 次元に次元を落とし、100 次元、3 次元と 4 層で 3 次元まで超低次元化する。

それぞれの CGP では、入力特徴量およびそれらを組み合わせる演算子の最適化を行う。このため、実際に用いられる既存特徴量の数は第 2 層のすべての CGP で用いられる既存特徴量の合計となり、すべての既存特徴量が用いられるとは限らない。したがって、用いられる既存特徴量が減った場合は特徴量算出コストを抑制することができる。また、CGP の適応度は、それまでに構築した M 次元（本論文では $M = 1$ ）の特徴量を用いた k 近傍法（本論文では $k = 3$ ）によって学習データを分類した場合の分類精度とする。 k 近傍法では、学習データを 50% ずつに分け、一方をサブ学習セットとし、もう一方を検証セットとする。したがって、適応度は検証セットの分類精度である。

5.3 特徴量の超低次元化および画像分類実験

5.3.1 使用した画像セット

本実験で扱う画像分類問題は対象クラスであるのかどうかを分類する 2 クラス分類問題である。本論文では、クラス内の多様性が高い一般画像（自転車、人物）と医用画像（小腸画像）、およびクラス内の多様性が低い一般画像（歩行者）に適用した。

Graz-01 データセット

クラス内の多様性が高い一般画像への適用には Graz-01 データセット [48] を用いた。図 5.2 に Graz-01 データセットの画像の例を示す。Graz-01 データセットは bikes と persons の 2 つのオブジェクトクラスと背景クラスで構成される。bikes, persons, 背景はそれぞれ、373 枚、460 枚、270 枚である。Graz-01 データセットはクラス内の多様性が高く、難しいデータセットであることが知られている。例えば、bikes クラスでは自転車全体が写っているもの（複数、単数）、向きが異なるもの、車輪やサドルだけのものも含まれる。persons クラスも同様に人物の数、向き、スケール、姿



図 5.2: Graz-01 データセットの画像例

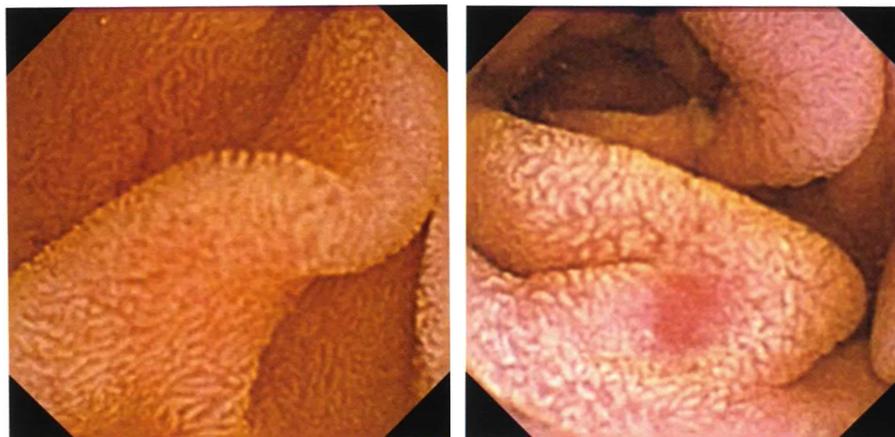
表 5.1: カプセル内視鏡画像分類における層数および次元数

Layer	1st	2nd	3rd	4th	5th
5 layers	2449	1000	500	100	3
4 layers	2449	1000	100	3	-
3 layers	2449	100	3	-	-

勢、手のひらだけといった画像も含まれる。背景クラスでは自動二輪車の画像や人物が普段は存在している通りなどの画像も含まれる。オリジナルの画像サイズは $640 \div 480$ pixel であるが、計算コストを抑えるために $300 \div 225$ pixel に縮小して適用した。本実験では、bikes と persons の 2 つのクラスについて対象画像と非対象画像の分類を行った。学習データとテストデータはそれぞれ合計 200 枚で構成され、対象画像は対象クラスから 100 枚ランダムに選択し、非対象画像はもう一方のオブジェクトクラスから 50 枚、背景クラスから 50 枚をランダムに選択して用いた。学習画像の選択で偏りが生じる可能性があるため、学習データとテストデータの組を 10 組作成し、その分類性能の平均で評価を行う。Graz-01 データセットの分類実験では、提案モデルは 4 層、各層の次元数は第 1 層が後述する既存特徴量で 4000 次元程度、第 2 層、第 3 層、第 4 層の次元数はそれぞれ、1000 次元、100 次元、3 次元とした。

カプセル内視鏡画像

医用画像への適用では、カプセル内視鏡画像を用い、病変があるかどうかの分類を行った。個人情報保護の点から実験に用いた画像を掲載することができないが、実験に用いた画像は図 5.3 ([49] から引用) に示すような小腸内部が撮影された画像である。本実験では、カプセル内視鏡で小腸内部を撮影した画像を用い、4 種類の異常が混在する異常画像と全く異常部位が含まれない正常画像の 2 クラス分類を行った。カプセル内視鏡は、小腸の蠕動運動で移動するため、撮影する角度や絨毛までの距離、異常部位の大きさや見た目などが多様な画像が撮影される。学習データは異常 99 枚、正常 99 枚の合計 198 枚、テストデータは異常 95 枚、正常 95 枚の合計 190 枚で構成される。画像サイズは $256 \div 256$ pixel である。カプセル内視鏡画像の分類実験では、提案モデルの層数および次元数を表 5.1 の 3 パターン用意し、その違いによる影響を考察する。



(a) 正常画像の例

(b) 異常画像の例

図 5.3: カプセル内視鏡画像の例（出典：日比紀文,『カプセル内視鏡カラーアトラス』, 診断と治療社, 2007, p.10,26）



図 5.4: 歩行者画像の例

歩行者画像

クラス内の多様性の低い歩行者画像においては, INRIA Person Dataset [23]を用いた. 図 5.4 に歩行者画像の例を示す. 画像サイズは $64 \div 128$ pixel であり, 歩行者画像として学習データ 2416 枚, テストデータ 1126 枚が用意されている. 非歩行者画像が用意されていないため, $640 \div 480$ pixel の背景画像として用意されている画像からランダムに歩行者画像と同数だけ切り出して用いた. Graz-01 データセットの人物とは異なり, 画像のおよそ中央に立ち姿の人物が映った画像であり, 手やバッグの位置や服の色や柄が異なるものの, おおよその大きさや形は類似している画像セットである. 歩行者画像への適用では, 提案モデルの層数および次元数は Graz-01 データセットと同様に 4 層で, 第 2 層, 第 3 層, 第 4 層の次元数はそれぞれ, 1000 次元, 100 次元, 3 次元とした.

5.3.2 使用した既存の画像特徴量

本実験で用いた特徴量は, GIST, Improved Local Binary Pattern (ILBP) および Bag-of-Visual Words (BoVW) である. GIST は画像の構図を表す特徴量であり, 道路のように奥行きのある画像やオフィス内を写した画像などのように, 写っている環境の大まかな構図を表す. GIST は, 画像を $4 \div 4$ ブロックに分割し, 各ブロックで方向およびスケールの組み合わせで 20 種類のフィルタバンクに適用した応答値が特徴量であり, これを RGB の 3 つの色空間に対して行うため, 合計で

960次元である。ILBPは局所的なテクスチャのパターンを記述するLocal Binary Pattern (LBP)に対してよりエッジを強調する改良が行われている特徴量であり、256次元である。BoVWは自然言語処理におけるBag-of-Wordsに基づいた特徴量であり、色やエッジの局所特徴量がVisual Words (VW)として用いられ、それらが画像全体でどのくらいの頻度で出現するかを特徴量としている。BoVWでは、Colored Scale Invariant Feature Transform (CSIFT) [50], RGBのヒストグラムおよび $L^*a^*b^*$ のヒストグラムの3種類の局所特徴量をVWとして用いた。CSIFTは画像のスケール変化や回転に頑健な特徴量として知られているScale Invariant Feature Transform (SIFT)と比較して色の違い、照明変化に頑健な特徴量である。VWの最大数を1000次元とするため、合計で最大3000次元である。VWは $k = 1000$ としてk-meansクラスタリングを行うことで作成されるため、クラスタリングの結果でVWの数が減る可能性がある。したがって、既存の画像特徴量の最大次元数は4216次元(=960+256+3000)である。本実験では、Graz-01データセットにおいては、第1層の既存特徴量の次元数はbikesクラスで平均4119次元、personsクラスで平均4067次元であった。一方で、カプセル内視鏡画像ではカプセル内視鏡が回転するため、画像の向きの概念がない。このため、画像全体の構図を扱うGISTを用いると学習データに過学習してしまう可能性があるため、カプセル内視鏡画像においてはILBPおよびBoVWの最大3256次元の既存特徴量を用いた。本実験では、カプセル内視鏡画像においては第1層の既存特徴量の次元数は2449次元であった。

VWの設定を次に述べる。CSIFTの次元数はSIFTと同じ128次元である。RGB、 $L^*a^*b^*$ のヒストグラムは各色空間について4ビットに量子化し、それぞれ3つの色空間であるため、64次元である。これらの局所特徴量を $20 \div 20$ pixelのブロックサイズで10 pixelごとにラスタスキャンして算出し、VWごとのヒストグラムを算出することでBoVWの特徴量とする。

また本実験では、歩行者画像において第1層の既存特徴量として、上記の特徴量を用いた実験と、歩行者検出において有効性を示しているHistograms of Oriented Gradients (HOG) [23]を用いた実験の2種類を行った。本実験では、上記の特徴量のうちILBPとBoVWを用いた実験では、第1層の既存特徴量の次元数は3200次元であった。HOGを用いた実験では第1層の既存特徴量の次元数は3780次元であり、HOGの設定はブロックサイズが $8 \div 8$ pixel, セルサイズが $2 \div 2$ ブロック, 9方向(20° ごと)である。

5.3.3 CGPの設定

表5.2にCGPの設定を示す。Graz-01データセットにおいては、コンパクトな式を構築するためにCGPの最大ノード数を21に制限した。一方で、カプセル内視鏡画像、歩行者画像においては、式の表現力を高めて収束を早めるためにCGPの最大ノード数を61を増やして実験を行った。最大世代数は、Graz-01データセットおよびカプセル内視鏡画像においては、上記の設定において予備実験においてそれぞれ十分に収束する世代数に設定した。歩行者画像においてはクラス内の多様性が低いことから学習データに過学習してしまうため、適応度が0.7を超えること、もしくは1000世代に達することを終了条件とした。なお、Graz-01データセットおよびカプセル内視鏡画像においては収束しても過学習するほどの適応度の上昇は見られなかった。

5.3.4 分類性能による評価

Graz-01データセットおよびカプセル内視鏡画像では、Support Vector Machine (SVM), 決定木(C4.5), k近傍法($k = 3$)の3種類についてWEKA (version 3.6.10) [47]を用いて分類を行った。歩行者画像においては処理の解析しやすさの観点から分類器をC4.5に限定し、リーフノードに到

表 5.2: CGP の設定

Parameters	Values
Maximal #generation	30000(Graz) 10000(Capsule) 1000(Pedestrian)
#Population	100
#Children	10
Generation alternation model	Minimal generation gap [7]
Crossover rate	0.8
Crossover model	Two-point crossover
Mutation rate	0.02
Tournament size	2
#Input feature node	10(Graz) 30(Capsule, Pedestrian)
#Arithmetic operator node	10(Graz) 30(Capsule, Pedestrian)
#New feature node	1
Limit of fitness value	1.0(Graz, Capsule) 0.7(Pedestrian)
Arithmetic operators (1 input: x)	$\exp(x), \log(x), \bar{x}$ $\sin(x), \cos(x), \ x\ $
Arithmetic operators (2 inputs: x, y)	$x + y, x - y, x \div y$ $x \pm y, x \bmod y, x^y$

達した学習データの各クラスのデータ数の割合を用いてしきい値を変化させた Receiver Operating Characteristic (ROC) 曲線によって分類性能を評価する。それぞれの分類器のパラメータは WEKA の既定値を用いた。また、比較手法として用いる主成分分析は統計処理ソフトウェアのひとつである R [51]を用いた。

Graz-01 データセット

表 5.3, 表 5.4 に bikes, persons クラスのそれぞれの分類器を用いた結果を示す。分類性能として 10 組の学習データとテストデータの組について再現率と適合率の調和平均である F 値の最良と平均の 2 種類を示し, 10 組における F 値の標準偏差も示す。ここで, テストデータにおける最良の値は, 最良の学習データと組になっているテストデータの値を示している。表 5.3, 表 5.4 では, 学習, テストそれぞれにおいて各分類器ごとにすべての層の中で最良の値を太字で示す。本実験では bikes, persons 両クラスにおいて最良の値が出現する層に同様の傾向が見られた。

まず学習データにおいて, SVM は第 1 層 (既存特徴量), C4.5 は第 2 層, k 近傍法は第 4 層が最も高い精度となった。SVM, C4.5 は高次元であれば境界面を作りやすいために 1000 次元以上の精度が高くなったと考えられる。一方で k 近傍法が 3 次元にもかかわらず第 4 層で最良となったのは, CGP の適応度に k 近傍法における分類精度を採用したためであると考えられる。

表 5.3: bikes クラスの分類結果

		Training				Test			
Layer		1st	2nd	3rd	4th	1st	2nd	3rd	4th
Dimension		4119	1000	100	3	4119	1000	100	3
SVM	Best	1.000	1.000	0.990	0.960	0.865	0.825	0.745	0.810
	Avg.	1.000	1.000	0.965	0.915	0.836	0.821	0.803	0.799
	Std. dev.	0.000	0.000	0.016	0.031	0.162	0.059	0.037	0.025
C4.5	Best	0.990	1.000	0.995	0.970	0.333	0.667	0.770	0.762
	Avg.	0.980	0.982	0.985	0.945	0.525	0.714	0.754	0.774
	Std. dev.	0.006	0.007	0.011	0.022	0.162	0.059	0.037	0.025
kNN	Best	0.920	0.935	0.965	0.965	0.713	0.840	0.800	0.765
	Avg.	0.891	0.916	0.931	0.942	0.784	0.810	0.806	0.781
	Std. dev.	0.017	0.015	0.017	0.019	0.035	0.020	0.020	0.034

そして、テストデータにおいては SVM は第 1 層（既存特徴量）、C4.5 は第 4 層、k 近傍法は第 2 層および第 3 層が高い精度となった。SVM は高次元に強い手法であるため、次元数を落とした場合に精度を向上させることができなかつたと考えられる。しかしながら、第 2 層の 1000 次元とほぼ同等の精度となり、計算コストを考慮するとよい結果であると考えられる。C4.5 では次元数が少なくなるにつれて精度が向上している。k 近傍法についても第 1 層（既存特徴量）の結果より第 2 層以降の方がよい精度となっている。C4.5 や k 近傍法のように過学習しやすい分類器においては、次元数を落としてより抽象的な特徴量にすることで学習データにおける分類器の過学習を抑え、テストデータの分類に有効に働いたと考えられる。また、全体的に標準偏差の値が低いことから安定して同様の性能をもつ特徴量を構築することができていると考えられる。

カプセル内視鏡画像

表 5.5 にカプセル内視鏡画像の異常正常分類の結果を示す。表 5.5 では、F 値の 10 試行平均を表 5.1 に示した 3 パターンについて、次元数を合わせて示している。学習、テストそれぞれにおいて各分類器ごとにすべての層の中で最良の値を太字で示す。また、網掛けの値は、学習、テストそれぞれにおいて各分類器ごとに 3 パターンの中で最良の値である。

学習データにおいては Graz-01 データセットと同様の傾向の結果を得ることができた。一方で、テストデータでは 3 種類すべての分類器において、第 1 層（既存特徴量）の結果を第 2 層以降の結果が上回った結果となった。カプセル内視鏡画像はすべての画像が小腸内部であるため、Graz-01 データセットのように多様な背景がなく、異常部位が bikes や persons のようにある程度決まった形がない。このため、正常と異常が似た画像であるだけでなく、異常部位が多様であるために、高次元の既存特徴量を用いた場合、より学習データに過学習してしまったと考えられる。

層数、次元数の違いにおいては、学習データでは層数にかかわらず、同じ次元数では同程度の精度となった。しかしながら、テストデータでは同じ次元数でも異なる精度となり、次元数にかかわらず第 2 層から第 3 層の結果がよいものとなった。このことから、層数が多すぎると既存特徴量の情報が失われすぎてしまい精度が低下してしまうが、次元数が小さくても浅い層であれば精度を保ちつつ大幅に次元数を落とすことができたといえる。ただし、Graz-01 データセットとテストデー

表 5.4: persons クラスの分類結果

		Training				Test			
Layer		1st	2nd	3rd	4th	1st	2nd	3rd	4th
Dimension		4067	1000	100	3	4067	1000	100	3
SVM	Best	1.000	1.000	0.980	0.915	0.825	0.805	0.794	0.814
	Avg.	1.000	1.000	0.966	0.878	0.819	0.818	0.789	0.764
	Std. dev.	0.000	0.000	0.012	0.035	0.100	0.030	0.022	0.033
C4.5	Best	0.995	1.000	0.995	0.975	0.675	0.718	0.755	0.734
	Avg.	0.986	0.992	0.987	0.952	0.669	0.748	0.742	0.774
	Std. dev.	0.007	0.006	0.007	0.015	0.100	0.030	0.022	0.033
kNN	Best	0.890	0.925	0.970	0.950	0.768	0.734	0.780	0.790
	Avg.	0.861	0.902	0.935	0.943	0.737	0.782	0.789	0.769
	Std. dev.	0.020	0.019	0.018	0.007	0.045	0.030	0.024	0.027

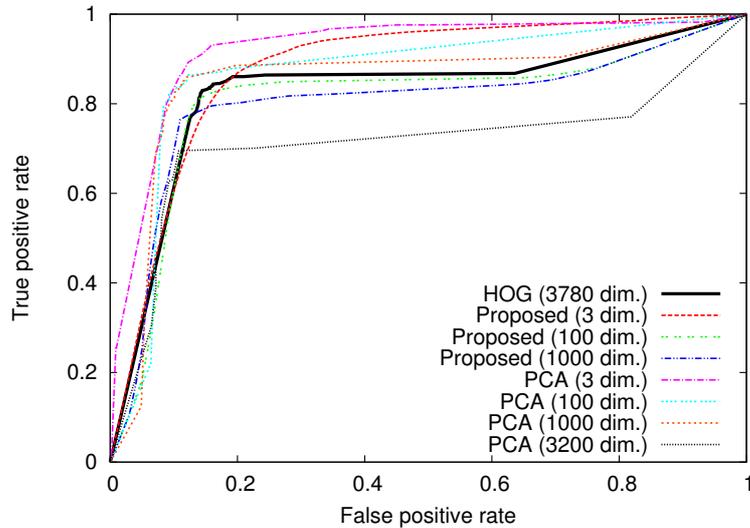
表 5.5: カプセル内視鏡画像の分類結果

		Training					Test				
Dimension		2449	1000	500	100	3	2449	1000	500	100	3
SVM	5 layers		1.000	0.981	0.960	0.939	0.784	0.880	0.871	0.865	0.848
	4 layers	1.000	1.000	-	0.963	0.934		0.881	-	0.876	0.801
	3 layers		-	-	0.958	0.904		-	-	0.880	0.847
C4.5	5 layers		0.994	0.992	0.983	0.968	0.638	0.787	0.820	0.809	0.783
	4 layers	0.995	0.996	-	0.985	0.958		0.822	-	0.784	0.735
	3 layers		-	-	0.988	0.943		-	-	0.812	0.842
kNN	5 layers		0.922	0.945	0.956	0.963	0.639	0.871	0.867	0.866	0.833
	4 layers	0.791	0.927	-	0.946	0.958		0.872	-	0.873	0.796
	3 layers		-	-	0.923	0.944		-	-	0.868	0.841

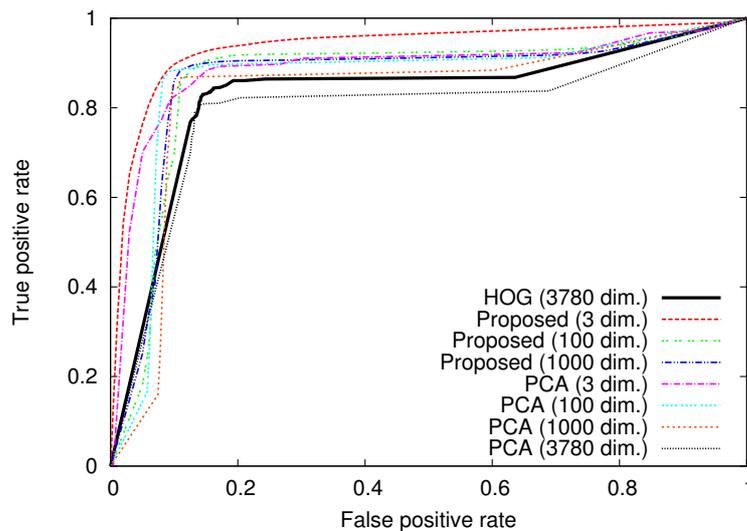
タの結果の傾向が異なることから、適用する画像分類問題によって適切な層数および次元数が異なると考えられ、これらを自動決定することが今後の課題として挙げられる。

歩行者画像

図 5.5 に歩行者画像における ROC 曲線を示す。図 5.5(a) が ILBP と BoVW を第 1 層に用いた結果であり、図 5.5(b) が HOG を第 1 層に用いた結果である。また、構築した 1000 次元、100 次元、3 次元の特徴量を用いた結果は 10 試行の平均である。図 5.5(a) から 3780 次元の HOG を用いた分類性能とすべての層の結果は同程度であるといえる。一方で、図 5.5(b) を見ると、HOG を第 1 層に用いて特徴量を構築することで分類性能を向上させることができた。構築した 1000 次元、100 次元の結果は同程度となったが、3 次元にすることで最もよい性能となり、すべての HOG を用いるよりも有効な特徴量を選択し組み合わせることでより本質的な特徴量にすることができたといえる。また、ILBP と BoVW を用いた結果より HOG を用いた結果がよかったことから、歩行者のようにある程度見え方が決まっている対象に対しては画像の位置に依存した特徴量が有効であるが、それらの中でも有効に働く特徴量を選択し組み合わせることでより分類性能を向上させることができたといえる。



(a) ILBP と BoVW を第 1 層に用いた場合の ROC 曲線



(b) HOG を第 1 層に用いた場合の ROC 曲線

図 5.5: 歩行者画像における ROC 曲線

主成分分析との比較

表 5.6 に Graz およびカプセル内視鏡画像における主成分分析との比較結果を示す。表 5.6 では、それぞれのクラスにおいて 100 次元および 3 次元について、第 100 主成分、第 3 主成分までを用いた結果を用いて、提案手法と次元数を合わせることで比較する。主成分分析と提案手法の結果を比較して F 値が高い数値を太字で示す。表 5.6 からわかるように、すべてにおいて提案手法がよりよい性能を示した。この結果から、Graz-01 データセットおよびカプセル内視鏡画像はクラス内の多様性が高いため、それぞれのクラスを考慮せずに主成分を求めた場合は、2つのクラスを分離す

表 5.6: 主成分分析との比較

Dimension		Training				Test			
		100		3		100		3	
Method		PCA	Proposed	PCA	Proposed	PCA	Proposed	PCA	Proposed
Bikes	SVM	0.954	0.965	0.641	0.915	0.466	0.803	0.425	0.799
	C4.5	0.981	0.985	0.674	0.945	0.471	0.754	0.445	0.774
	kNN	0.856	0.931	0.812	0.942	0.431	0.806	0.445	0.781
Persons	SVM	0.936	0.966	0.761	0.878	0.448	0.789	0.380	0.764
	C4.5	0.982	0.987	0.797	0.952	0.466	0.742	0.432	0.774
	kNN	0.796	0.935	0.836	0.943	0.403	0.789	0.453	0.769
Capsule	SVM	0.929	0.960	0.481	0.926	0.805	0.874	0.444	0.832
	C4.5	0.980	0.985	0.333	0.956	0.585	0.802	0.333	0.787
	kNN	0.750	0.942	0.757	0.955	0.548	0.869	0.555	0.823

るのには有効に働かなかったと考えられる。一方で、クラス内の多様性が低い歩行者画像においては主成分分析と提案手法において同程度の性能が確認された。図 5.5(a) を見ると、主成分分析における 3 次元が最もよい性能を示したが、図 5.5(b) においては、提案手法の 3 次元が最もよい性能を示した。どちらも分類に有効でない特徴量を用いないように超低次元化することで、すべての既存特徴量を用いるより性能を向上させることができたと考えられる。

5.3.5 処理の解析しやすさによる評価

構築した 3 次元の特徴量を用いたデータ分布の可視化

図 5.6～図 5.9 に構築した 3 次元の特徴量を用いたデータ分布の可視化結果の一例を示す。図 5.6, 図 5.7 に Graz-01 データセット, 図 5.8 にカプセル内視鏡画像, 図 5.9 に歩行者画像の可視化結果をそれぞれ示す。bikes, persons 両クラス, カプセル内視鏡画像, 歩行者画像ともに, 学習データとテストデータの各クラスの分布は似たものとなった。Graz-01 データセット, カプセル内視鏡画像では多様性の高い画像データセットにもかかわらず, 良好に対象画像と非対象画像を分離することができた。特にカプセル内視鏡画像では異常画像と正常画像はより明確に分離された結果となった。歩行者画像においても, 歩行者と非歩行者の境界が少し重なるものの, それぞれ別の軸方向に広がるような結果となった。図 5.6～図 5.9 の例では, 3 次元空間においてある程度の広がりをもっているが, 学習データとテストデータの組や試行によってはデータが 3 次元空間内で線状に並ぶような結果も見られた。このため, 適応度で 3 次元空間における広がりや評価する指標を導入するなど, 学習データとテストデータの組や複数試行においても安定して広がりをもたせることが今後の課題として挙げられる。

図 5.10 に 3 次元空間における bikes クラスの画像の例を示す。図 5.10 では, 緑の ÷ マークが bikes クラスの画像であり, 図 5.10 の視点から見て下側に集まっている。密度の高い位置に写像された画像の例は下の 3 枚の画像であり, 似た構図 (横向きの自転車) の画像となっている。一方で, 非対象クラスの画像が集まっているところに写像された画像の例では, 自転車の一部だけが写っているものなど, 学習データで出現頻度が少ない画像が多かった。したがって, この 3 次元空間において似ている画像は近い位置にあり, このように代表的な画像を示すことで, データ分布を直感的に示すことができる。また, このように写像された画像を示すことで, それぞれの特徴量軸に「横向き度」「映っている範囲」などといった人間に直観的な意味をつけることもできると考えられる。

表 5.7: 画像セットごとの各次元数における決定木の大きさ

	1st layer	1000 dim.	500 dim.	100 dim.	3 dim.
Graz	28.4 100.0%	17.7 62.3%	- -	14.6 51.4%	6.3 22.2%
Capsule	21.0 100.0%	16.2 77.1%	11.8 56.2%	7.8 37.1%	4.4 21.0%
Pedestrian	294.0 100.0%	255.7 87.0%	- -	282.1 96.0%	21.3 7.2%

表 5.8: テストデータの分類における WEKA の実行時間に関する高速化倍率

	2nd layer vs 1st layer	3rd layer vs 1st layer	4th layer vs 1st layer
SVM	2.31	4.21	5.23
C4.5	2.12	3.65	4.26
kNN	4.47	16.09	23.25

このような軸の意味を与えられたデータから自動で付与することができれば、より分類の根拠を示すのに有効であると考えられる。

決定木における木の大きさ

決定木においてはそれぞれのノードで特徴量の大小でデータが分岐することで分類を行うため、ノード数が少ないほど処理が解析しやすいといえる。表 5.7 に決定木におけるノード数を示す。表 5.7 は、それぞれの画像セットにおいて、それぞれの次元数ごとの決定木を構成するノード数の平均と第 1 層のノード数に対する割合である。Graz-01 データセット、カプセル内視鏡画像においてはデータ数が小さいため、第 1 層においても 30 ノード以下の比較的小さい木となった。しかし、最終層の 3 次元に近づくと木の大きさは小さくなり、3 次元においては第 1 層の 20%ほどの大きさになった。また、歩行者画像においてはデータ数が大きいため、全体的にノード数が大きくなった。このため、3 次元におけるノード数の割合は 7.2%となり、より小さい木にすることができた。特にデータ数が大きい場合、100 次元の特徴量を用いた場合でもノード数も大きい結果となったため、3 次元にすることでより超低次元化の効果を得ることができたといえる。

5.3.6 計算コストによる評価

画像分類の計算コストは大きく分けて、特徴量算出コストと分類計算コストの 2 つである。提案モデルでは、既存特徴量を組み合わせることで特徴量の超低次元化を行っている。したがって、特徴量算出コストは提案モデルによって使用された既存特徴量の数に依存する。使用される既存特徴量の数は第 2 層の次元数および CGP の入力ノード数によって変化する。Graz-01 データセットにおいては、10 組の学習データとテストデータの組の既存特徴量数の平均は 1352 個である。こ

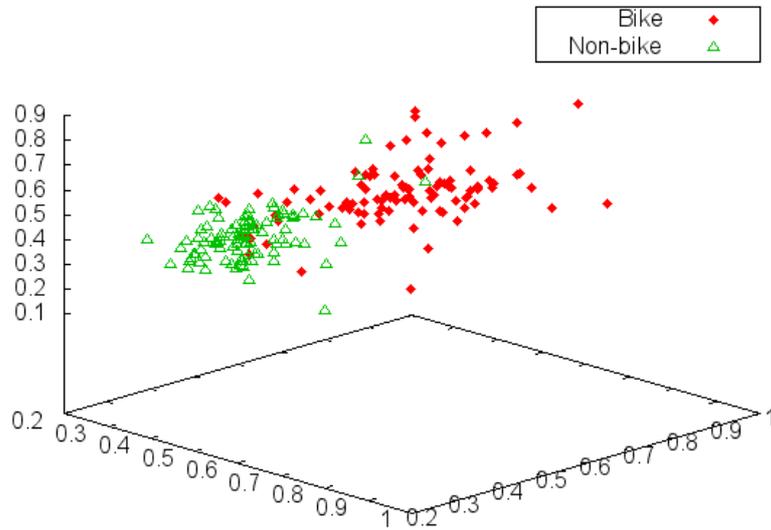
れはすべての既存特徴量の数のおよそ 32 % に相当する。カプセル内視鏡画像については、第 2 層が 1000 次元の場合に平均で 916 個（第 1 層の 37%）、100 次元の場合に平均で 188 個（第 1 層の 7.7%）であり、歩行者画像においては、第 1 層が ILBP、BoVW のときに平均で 1263 個（第 1 層の 39%）、HOG のときに 750 個（第 1 層の 20%）である。全ての既存特徴量の算出コストが同一ではないが、第 2 層が 1000 次元でおよそ 2~4 割程度、100 次元で 1 割以下のコストに抑えることができたといえる。

提案モデルでは、使用される既存特徴量を複数の層において演算することで、新たな特徴量としている。このため、各 CGP で最適化された関数のノード数も特徴量計算コストに関わる。最適化された関数のノード数は、各 CGP の入力特徴量によっても変化するが、関数 1 つあたり数個程度であった。このため、特徴量算出コストにおける演算のコストは微小であるといえる。付録 B に Graz-01 データセットにおいて最適化された各層の関数の例を示す。

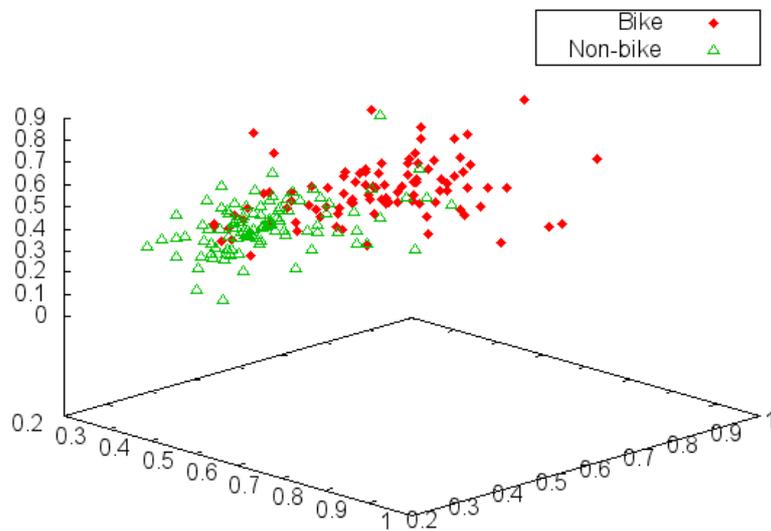
また、分類計算コストは分類器によって処理が異なるが、少なからず特徴量の次元数が計算コストにかかわっている。そこで、Graz-01 データセットにおいて WEKA を用いた特徴量の次元数の違いによる分類計算コストの比較を行った。5.3.4 項で用いた SVM、C4.5、k 近傍法について既存特徴量を用いた分類に対する高速化倍率を算出した。表 5.8 にそれぞれの高速化倍率を示す。WEKA の処理におけるオーバーヘッドがあり、次元数の比率で線形には高速化されなかったが、次元数を落とすことで 2~20 倍以上の高速化を実現した。

5.4 まとめ

本章では、従来有効性が示されている高次元の画像特徴量を演算子で組み合わせ、階層的に徐々に次元数を落とし、最終的に 3 次元に超低次元化する手法を提案した。クラス内の多様性が高い画像の 2 クラス分類実験を行った結果、特徴量の次元数を落としても、既存特徴量をすべて用いた場合および主成分分析で次元数を落とした場合と比較して、分類精度を向上させることができることを示した。また、3 次元に超低次元化した特徴量を用いて画像の分布を可視化し、代表的な画像を示すことで、画像の分布や関係性を解析しやすくなることを示した。処理のわかりやすさの評価として、決定木の木の大きさを既存特徴量をすべて用いた場合と比較して約 80% 以上のノード数を削減することができた。さらに、特徴量の超低次元化を行うことで、特徴量算出コストと分類計算コストを大幅に抑制することができることを示した。



(a) bikes クラスの学習データの分布



(b) bikes クラスのテストデータの分布

図 5.6: 構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (bikes クラス)

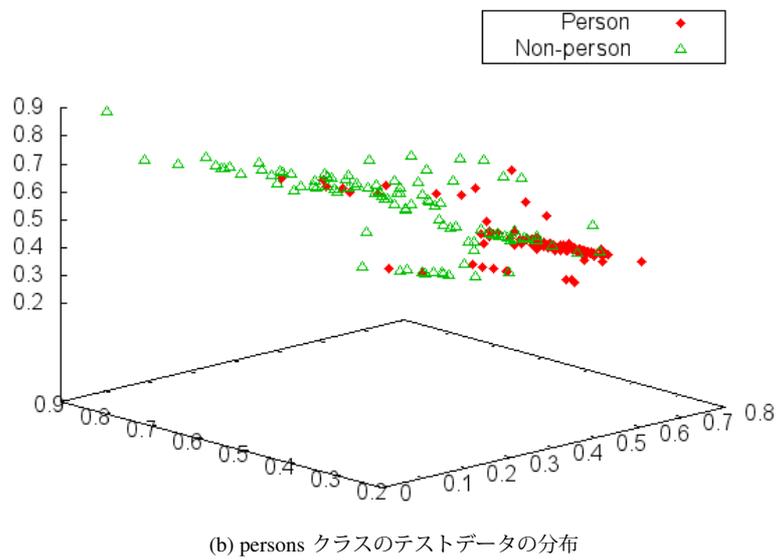
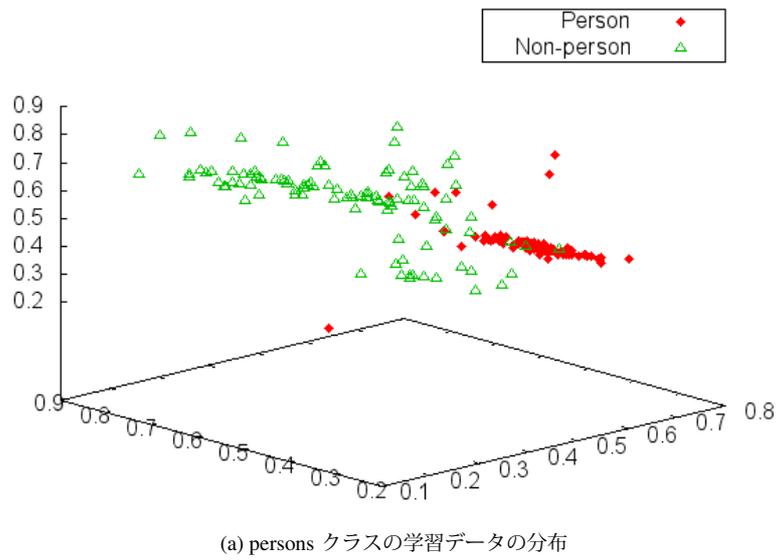
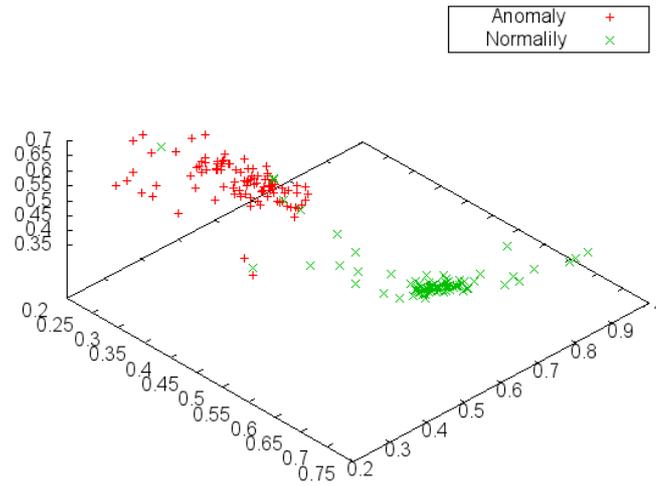
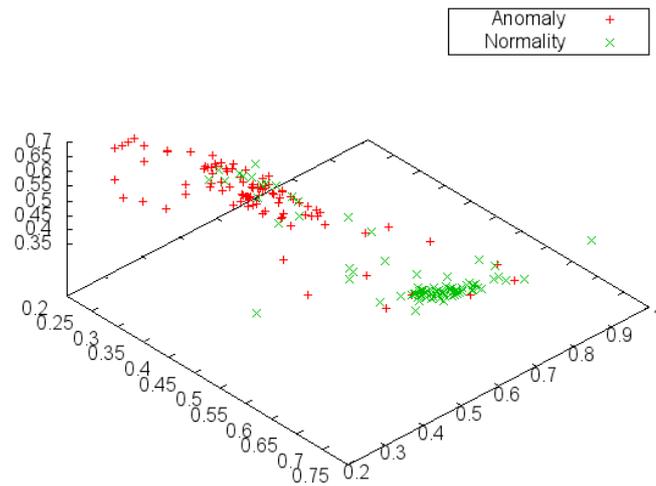


図 5.7: 構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (persons クラス)

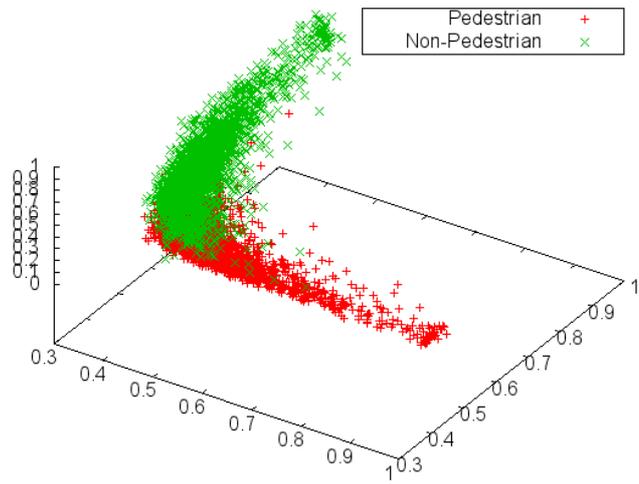


(a) カプセル内視鏡画像の学習データの分布

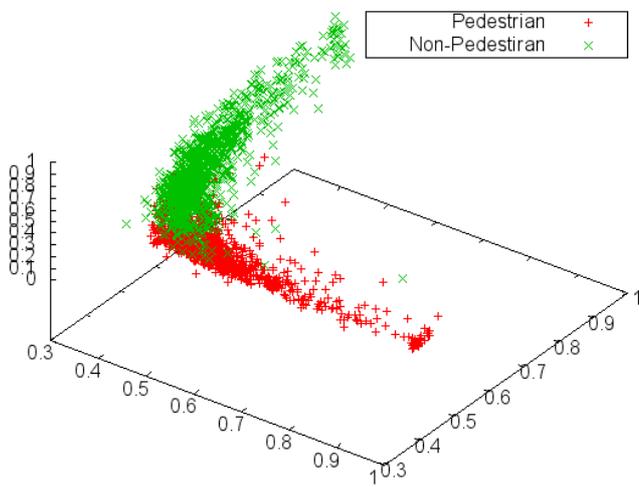


(b) カプセル内視鏡画像のテストデータの分布

図 5.8: 構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (カプセル内視鏡画像)



(a) 歩行者画像の学習データの分布



(b) 歩行者画像のテストデータの分布

図 5.9: 構築した 3 次元の特徴量を用いたデータ分布の可視化結果の例 (歩行者画像)

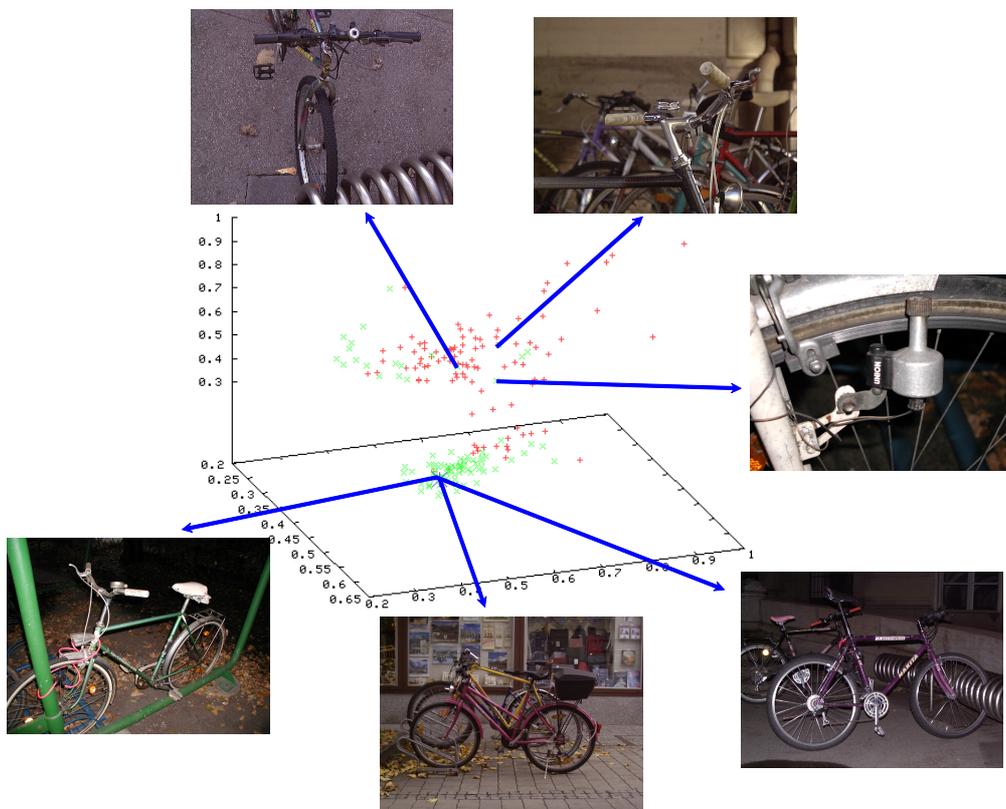


図 5.10: 3次元空間における bikes クラスの画像の例

第6章 結論

6.1 本論文で得られた成果

本論文では、画像認識に求められる“高い認識性能”，“低い計算コスト”，“認識処理のわかりやすさ”を実現するために2つのアプローチの手法を提案した。1つ目のアプローチでは、高い認識性能を保つために認識対象の性質および画像以外の特徴量として距離情報を利用し、ヒトにとって直観的な条件判断を行うネットワークを認識器に採用して特徴量を選択することで、低い計算コストと処理のわかりやすさの両立を図った。2つ目のアプローチでは、従来有効性が示されている高次元の画像特徴量を用いることで、認識性能を保持しつつ低い計算コストと処理のわかりやすさを両立するために特徴量の3次元以下への超低次元化を行った。このアプローチでは、データの自己組織化に基づく超低次元化手法と高次元特徴量の組み合わせによる超低次元化手法を提案した。それぞれの章で得られた成果は次の通りである。

←人物の部分領域に着目した姿勢変化や隠れに頑健な人物検出

距離情報を用いて画像を人物の一部が切り出されるような過分割を行った後、それぞれの領域が人物の候補領域であるかどうかの分類を行い、人物の候補領域を統合することで姿勢変化や隠れに頑健な人物検出を行う手法を提案した。認識対象を空港などの巡回警備において撮影される人物として人物検出実験を行った結果、既存手法と比較して姿勢変化、隠れに頑健に人物を検出することができることを示した。

←データの自己組織化に基づく特徴量の超低次元化

球面上にデータをランダムに配置し、データ同士の類似度を引力・斥力として表現して球面上を移動させることで自己組織化を行う手法を提案した。10種のデータセットを用いて超低次元化を行った結果、従来手法と比較してより同じクラスのデータを集め、異なるクラスのデータを分離させた可視化を行うことができることを示した。また、超低次元化された特徴量を用いたクラス分類実験の結果、用意された特徴量を用いた分類精度には劣った結果が多かったものの、削減した次元数を考慮すると良好な分類精度を得ることができる特徴量に超低次元化することができた。

←高次元特徴量の組み合わせによる特徴量の超低次元化

従来有効性が示されている高次元の画像特徴量を演算子で組み合わせ、階層的に徐々に次元数を落とし、最終的に3次元に超低次元化する手法を提案した。クラス内の多様性が高い画像の2クラス分類実験を行った結果、特徴量の次元数を落としても、既存特徴量をすべて用いた場合および主成分分析で次元数を落とした場合と比較して、分類精度を向上させることができることを示した。また、3次元に超低次元化した特徴量を用いて画像の分布を可視化し、代表的な画像を示すことで、画像の分布や関係性を解析しやすくなることを示した。さらに、特徴量の超低次元化を行うことで、特徴量算出コストと分類計算コストを大幅に抑制することができることを示した。

6.2 今後の課題

本研究をさらに発展させるための課題として、次の3点が挙げられる。

←より多くのクラスおよび大量の画像セットへの応用

近年では、Web上の大量の画像を取得することが容易になっている。このため、より多種多様なクラスの大量の画像を利用することが可能である。このような画像を用いることでより認識精度の向上、クラスの応用範囲の拡大が期待される。一方で、提案手法に有効なクラスの粒度や分類するクラス数に対する検討が必要となる。

←超低次元化に適した特徴量の自動構築

本論文では、従来有効性が示されている高次元特徴量を用いることで高い認識性能を保持したままで特徴量の超低次元化を行った。しかし、従来有効性が示されている特徴量であったとしても超低次元化に適しているとは限らない。このため、超低次元化で組み合わされることでより認識性能を向上させることができる特徴量を画像の画素値から自動構築することで、さらなる認識性能の向上が期待される。このとき、より特徴量算出コストが低くなるような処理とすることで、同時により計算コストを抑制することができると考えられる。

←超低次元化された特徴量への意味づけ

本論文では、3次元以下へ特徴量を超低次元化することでデータ分布の可視化を行った。提案手法では、それぞれの特徴量軸には意味を与えていないため、代表的なデータを示すことでデータの関係性を解析した。しかし、超低次元化された特徴量に意味を自動で与えることができれば、より認識処理のわかりやすさを向上させることができると期待される。本論文では、超低次元化された特徴量はデータの自己組織化結果と、既存特徴量を演算を重ねることで変換する数式である。このため、自己組織化の過程および変換の数式を解析して超低次元化された特徴量の意味づけをするのは困難である。そこで、特徴量を超低次元化する際に用いた学習データを用いることで特徴量の意味づけをすることが考えられる。例えば、学習データの中でも特徴的な画像に“横向きである”、“一部分だけが写っている”などといったラベルをつけ、それらの画像との類似度とそれぞれの特徴量の値の相関から特徴量軸の意味を推測する方法が考えられる。

謝辞

本研究を進めるにあたり親身にご指導と適切なお助言を賜りました長尾智晴先生に深く感謝いたします。長尾先生には、研究だけでなく大学発ベンチャーなどさまざまな活動を経験させていただき、研究室生活で経験したすべてを本研究に活かすことができたと考えております。

本論文をまとめるにあたり貴重なご指導、ご助言を頂きました森辰則先生、富井尚志先生、田村直良先生、岡嶋克典先生、有澤博先生に感謝申し上げます。

長尾研究室の皆様には大変お世話になりました。飛び級で進学した際に温かく迎え、切磋琢磨してくれた同期の皆様、学年が変わってしまっても同期として接してくれた皆様からは大変多くの刺激と活力をいただきました。そして、先に博士課程を卒業された先輩の皆様からは、自分も先輩の皆様のように優れた研究者になりたいと思ふ目標とさせたいいただき、研究への意欲を高めていただきました。また、輪講、サブ輪講などを通じて活発に議論させていただいた後輩の皆様にも、堅実に研究を進めていく姿にやる気をもらい、ときには指導させていただくことで自分もまた多くのことを学ぶことができました。感謝申し上げます。

長い学生生活を、自分を信頼していただき、やりたいことをさせていただいた両親に深く感謝します。また、自分を応援し帰りを待っていてくれた祖母に大変感謝いたします。そして、甥っ子、姪っ子のかわいい姿に元気をもらいました。とても明るい家族の皆様から心から感謝を送ります。

最後に、いつも笑顔で楽しく話をしてくれて研究生生活を支えてくれた大好きな君へ、本当にありがとう。

参考文献

- [1] John H Holland. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press, 1975.
- [2] David Edward Goldberg. *Genetic algorithm in search*. *Optimization and Machine Learning*, 1989.
- [3] 安居院猛, 長尾智晴. *ジェネティックアルゴリズム*, 1993.
- [4] 伊庭齊志. *遺伝的アルゴリズムの基礎*. オーム社, 1994.
- [5] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [6] 伊庭齊志. *遺伝的プログラミング*, 1996.
- [7] 佐藤浩, 小野功, 小林重信. 遺伝的アルゴリズムにおける世代交代モデルの提案と評価. *人工知能学会誌*, Vol. 12, No. 5, pp. 734–744, sep 1997.
- [8] Terence Soule and James A Foster. Effects of code growth and parsimony pressure on populations in genetic programming. *Evolutionary Computation*, Vol. 6, No. 4, pp. 293–309, 1998.
- [9] Riccardo Poli and William B Langdon. On the search properties of different crossover operators in genetic programming. *Genetic Programming*, pp. 293–301, 1998.
- [10] William B. Langdon. Size fair and homologous tree crossovers for tree genetic programming. *Genetic programming and evolvable machines*, Vol. 1, No. 1-2, pp. 95–119, 2000.
- [11] J. F. Miller and P. Thomson. Cartesian genetic programming. In *EuroGP*, pp. 121–132, 2000.
- [12] 中山史朗, 穂積知佐, 矢田紀子, 長尾智晴. 進化的条件判断ネットワーク EDEN による画像分類. *映像情報メディア学会誌*, Vol. 67, No. 7, pp. J278–J285, 2013.
- [13] 中山史朗, 穂積知佐, 矢田紀子, 長尾智晴. 進化的条件判断ネットワーク EDEN による画像分類. 第 38 回知能システムシンポジウム, 2011.
- [14] 特許番号 : 5548990 号.
- [15] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [16] J. Sanchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1665–1672, 2011.

- [17] D.G. Lowe. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157, 1999.
- [18] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, Vol. 29, No. 1, pp. 51–59, 1996.
- [19] H. Jin, Q. Liu, H. Lu, and X. Tong. Face detection using improved LBP under bayesian framework. In *Proceedings of IEEE First Symposium on Multi-Agent Security and Survivability*, pp. 306–309, 2004.
- [20] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *The Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 1470–1477, 2003.
- [21] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22, 2004.
- [22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, Vol. 42, pp. 145–175, 2001.
- [23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, pp. 886–893 vol. 1, June 2005.
- [24] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [25] T. Kohonen. Self-organized formation of topologically correct feature map. *Biol. Cybern.*, Vol. 43, pp. 56–69, 1982.
- [26] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, Germany, 1997.
- [27] R. N. Shepard and J. D. Coaroll. Parametric representation of nonlinear data structures. In P.R.Krishnaiah, editor, *Proc. Int. Symp. Multivariate Anal.*, pp. 561–592, New York, 1965.
- [28] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, pp. 611–622, 1999.
- [29] J. Rubner and P. Tavan. A self-organizing network for principal component analysis. *Europhys. Lett.*, Vol. 10, pp. 693–698, 1989.
- [30] H. Yin. ViSOM: A novel method for multivariate data projection and structure visualization. *IEEE Trans. Neural Netw.*, Vol. 13, No. 1, pp. 237–243, 2002.
- [31] H. Yin. Data visualization and manifold mapping using the visom. *Neural Netw.*, Vol. 15, No. 8-9, pp. 1005–1016, 2002.
- [32] S. Wu and T. W. S. Chow. Prsom: A new visualization method by hybridizing multidimensional scaling and self-organizing map. *IEEE Trans. Neural Netw.*, Vol. 16, No. 6, pp. 1362–1380, 2005.
- [33] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.

- [34] C. L. Blake and C. J. Merz. *UCI Repository of Machine Learning Databases*. Dept. of Information and Computer Science, Univ. of California at Irvine, 1998.
- [35] Mingyuan Zhao, Chong Fu, Luping Ji, Ke Tang, and Mingtian Zhou. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. *Expert Systems with Applications*, Vol. 38, No. 5, pp. 5197 – 5204, 2011.
- [36] Amin Dastanpour and Raja Azlina Raja Mahmood. Feature selection based on genetic algorithm and supportvector machine for intrusion detection system. In *The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013)*, pp. 169–181. The Society of Digital Information and Wireless Communication, 2013.
- [37] Fagbola Temitayo, Olabiyisi Stephen, and Adigun Abimbola. Hybrid ga-svm for efficient feature selection in e-mail classification. *Computer Engineering and Intelligent Systems*, Vol. 3, No. 3, pp. 17–28, 2012.
- [38] H. Guo, L. B. Jack, and A. K. Nandi. Feature generation using genetic programming with application to fault classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 35, No. 1, pp. 89–99, 2005.
- [39] L. Guo, D. Rivero, J. Dorado, C. R. Munteanu, and A. Pazos. Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications*, Vol. 38, No. 8, pp. 10425 – 10436, 2011.
- [40] J. Sherrah, R. E. Bogner, and A. Bouzerdoum. Automatic selection of features for classification using genetic programming. In *Proceedings of Australian and New Zealand Conference on Intelligent Information Systems*, pp. 284–287, 1996.
- [41] K. Krawiec. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, Vol. 3, No. 4, pp. 329–343, 2002.
- [42] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, vol. 28, pp.100-108, 1979.
- [43] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 2012 12(2) pp.1437-1454*, 2012.
- [44] S. Ikemura and H. Fujiyoshi. Real-time human detection using relational depth similarity features. *The Tenth Asian Conference on Computer Vision*, 2010.
- [45] NVIDIA. CUDA C Programming Guide [online], available: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [46] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM_PAK: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

- [47] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *ACM SIGKDD explorations newsletter*, Vol. 11, No. 1, pp. 10–18, 2009.
- [48] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Computer Vision-ECCV 2004*, pp. 71–84. Springer, 2004.
- [49] 日比紀文, 緒方晴彦, 大塚和朗. カプセル内視鏡カラーアトラス: 小腸病変の精細視野を拓く. 診断と治療社, 2007.
- [50] A.E. Abdel-Hakim and A.A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 1978–1983, 2006.
- [51] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, Vol. 5, No. 3, pp. 299–314, 1996.

研究業績リスト

論文

1. 齊藤航太, 長尾智晴: 人物の部分領域に着目した姿勢変化や隠れに頑健な人物検出, 映像情報メディア学会誌, Vol.68, No.2, pp.J99-J107 (2014)

国際会議発表

1. Kota Saito, Noriko Yata and Tomoharu Nagao: Three-dimensional Scene Reconstruction Using Stereo Camera and Laser Range Finder, Proc. of the IEEE SICE 2010 Annual Conference, 18-21 August 2010, Taipei, Taiwan (2010)
2. Kota Saito and Tomoharu Nagao: A Self-organizing Method Using Data Movement on Spherical Surface, IEEE International Conference on Systems, Man, and Cybernetics 2013 (IEEE SMC 2013), pp.1193-1198, October 13-16, 2013, Manchester, UK (2013)
3. Kota Saito and Tomoharu Nagao: Hierarchical Feature Generation Using Cartesian Genetic Programming, the IEEE Symposium on Computer and Informatics (ISCI 2014), 28-29 September, Kota Kinabalu, Malaysia (2014)

国内学会発表

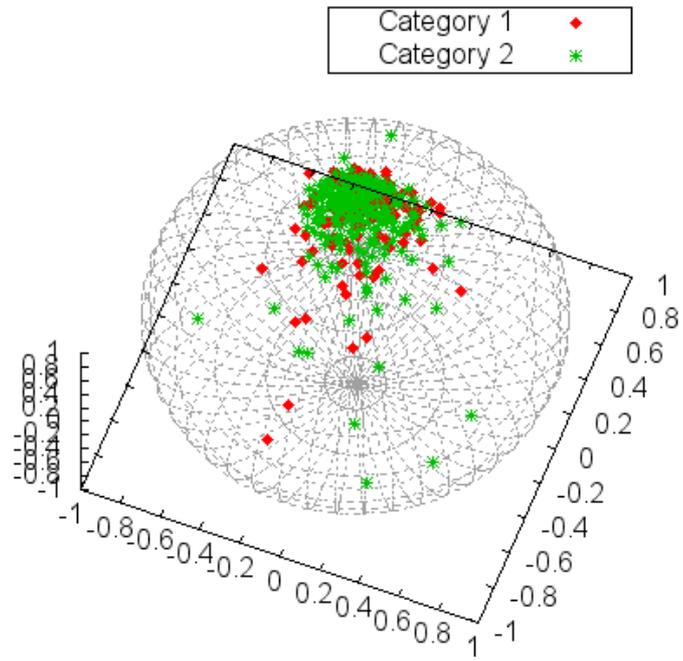
1. 齊藤航太, 矢田紀子, 長尾智晴: ステレオカメラとレーザレンジファインダを用いた三次元環境復元に関する研究, 電子情報通信学会総合大会 (2010)
2. 齊藤航太, 中山史朗, 矢田紀子, 長尾智晴: 立ち乗り電動2輪車を用いた動的環境における3次元データのクラス分類, 電気学会全国大会 (2011)
3. 齊藤航太, 長尾智晴: 乗用移動台車を用いた動的な3次元環境における人検出, 電子情報通信学会総合大会 (2012)
4. 齊藤航太, 長尾智晴: 大規模データへ向けたデータの自己組織化に基づくクラスタリング, 計測自動制御学会 第40回知能システムシンポジウム (2013)
5. 齊藤航太, 長尾智晴: Cartesian Genetic Programming を用いた階層的な特徴量の自動構築と画像分類への応用, 進化計算シンポジウム 2014 (2014)

受賞

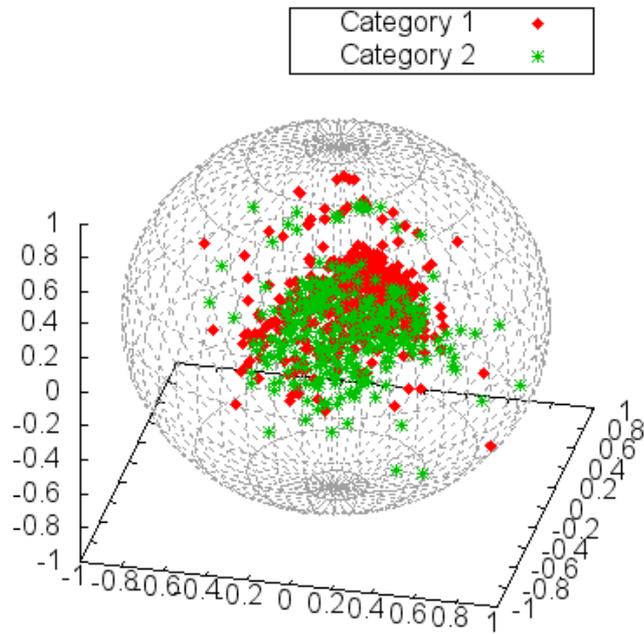
1. IEEE Computational Intelligence Society (CIS) Japan Chapter Young Researcher Award (対象論文：齊藤航太，長尾智晴：Cartesian Genetic Programming を用いた階層的な特徴量の自動構築と画像分類への応用，進化計算シンポジウム 2014)

付録A 第4章におけるUCIデータセットの 可視化結果

第4章におけるデータの自己組織化に基づく特徴量の超低次元化手法によるUCIデータセットの可視化結果（WIPDあり）を図A.1～図A.4に示す。

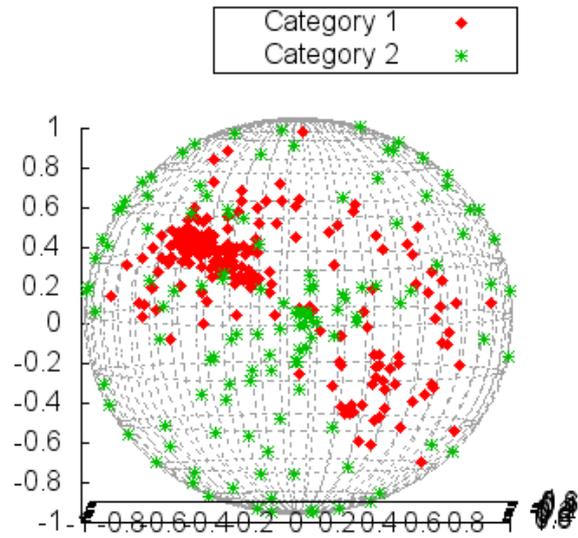


(a) Bupa

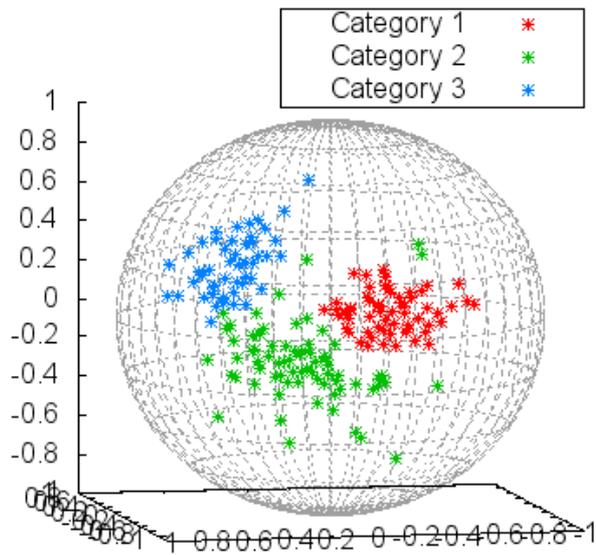


(b) Pima

図 A.1: UCI データセットの可視化結果 (Bupa, Pima)

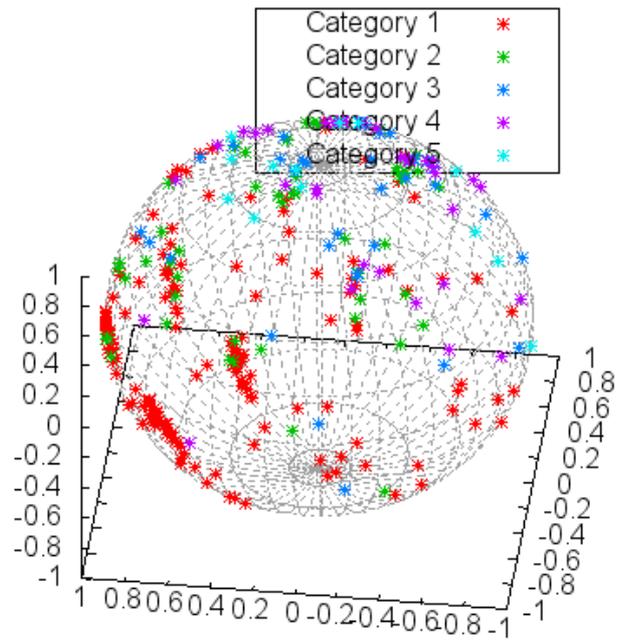


(a) Ionosphere

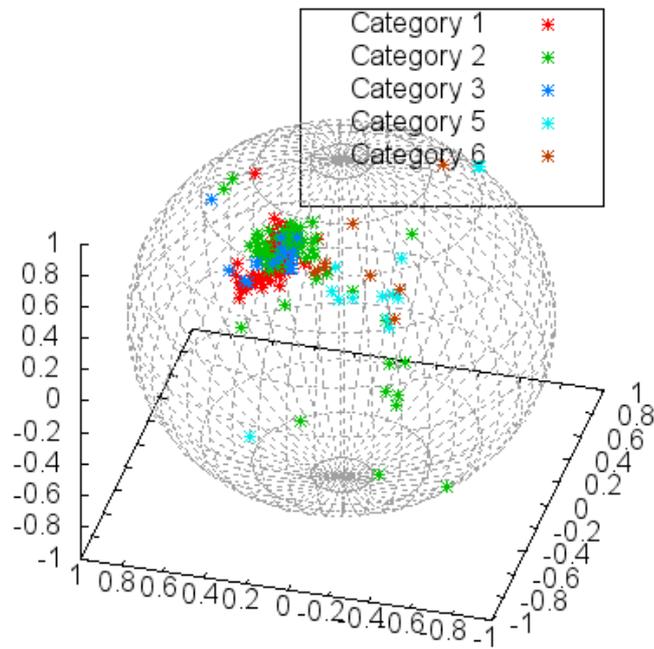


(b) Wine

図 A.2: UCI データセットの可視化結果 (Ionosphere, Wine)

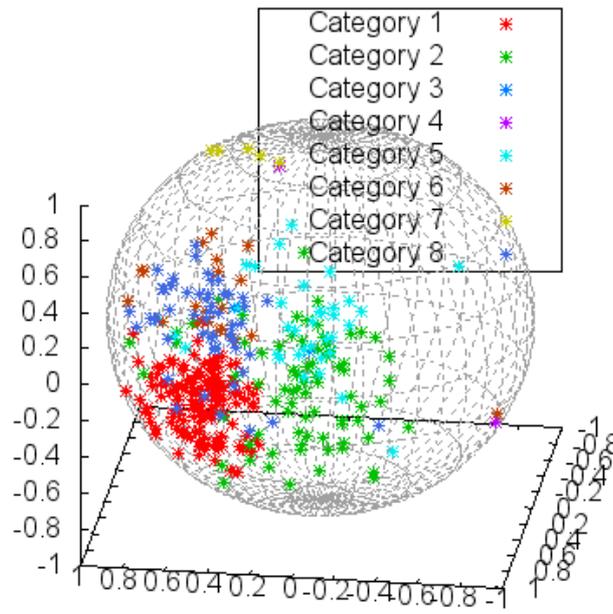


(a) Cleveland

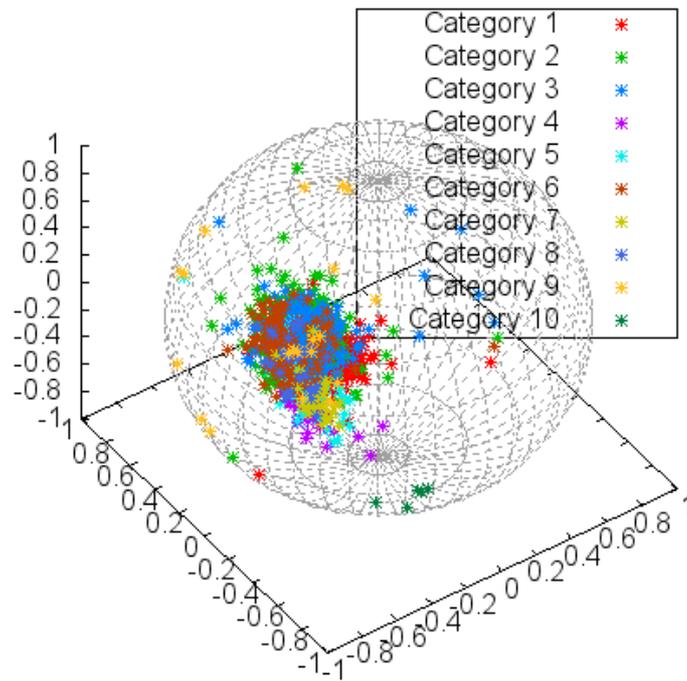


(b) Glass

図 A.3: UCI データセットの可視化結果 (Cleveland, Glass)



(a) Ecoli

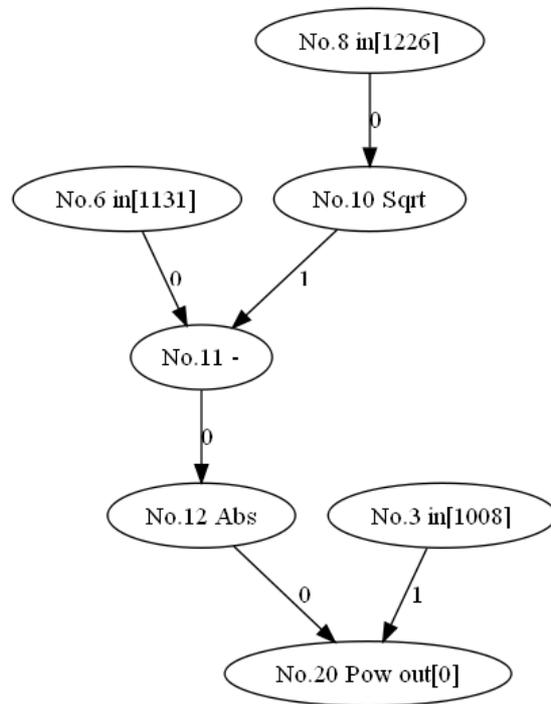


(b) Yeast

図 A.4: UCI データセットの可視化結果 (Ecoli, Yeast)

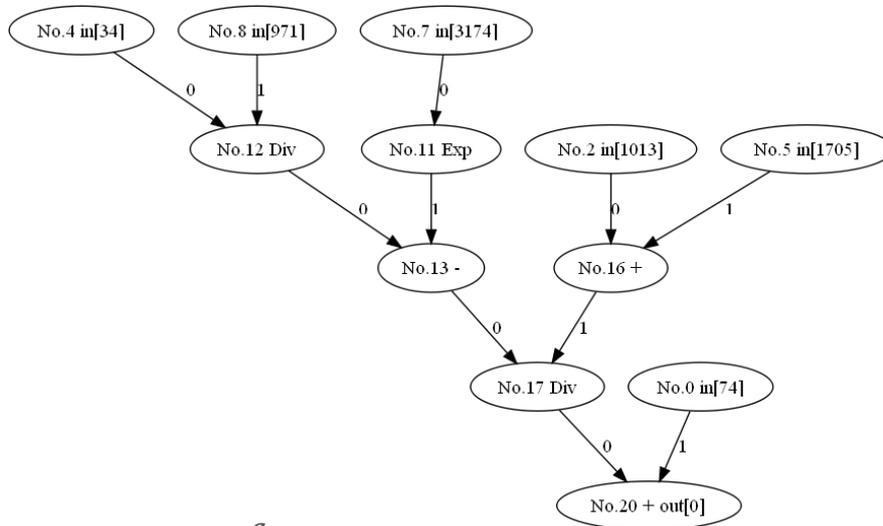
付録B 第5章において最適化された特徴量を組み合わせる関数の例

第5章における既存特徴量の組み合わせによる特徴量の超低次元化手法において、最適化された特徴量を組み合わせる関数の例を示す。なお、図B.1～図B.3の関数の例はGraz-01データセットのPersonsクラスの実験において最適化された関数の例である。また、第2層における結果の入力特徴量は既存特徴量であり、第3層、第4層の入力特徴量は前層で構築された特徴量である。



$$\left| f_{1131} - \sqrt{f_{1226}} \right|^{f_{1008}}$$

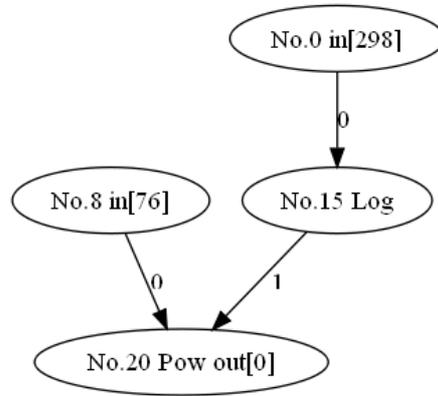
(a) Function1



$$\frac{\frac{f_{34}}{f_{971}} - \exp(f_{3174})}{f_{1013} - f_{1705}} + f_{74}$$

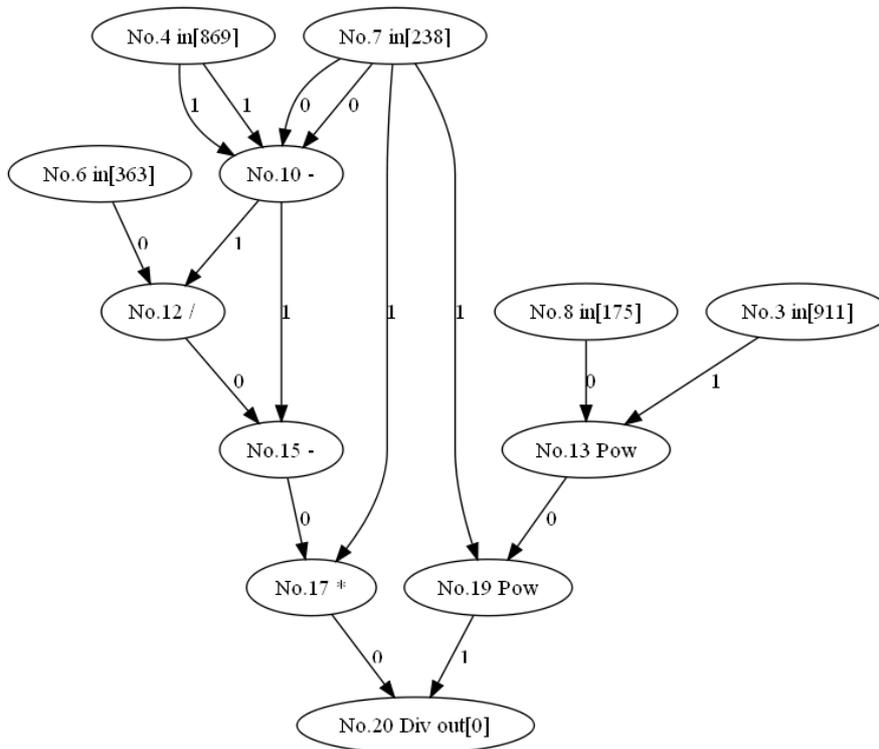
(b) Function2

図 B.1: 第 2 層において最適化された関数の例



$$f_{76}^{\log(f_{298})}$$

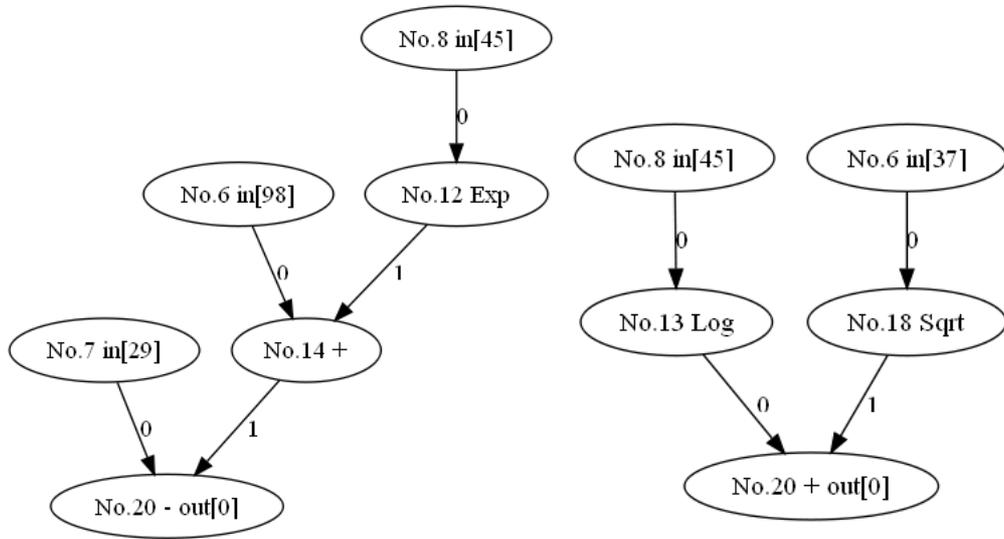
(a) Function1



$$\frac{f_{238} \left(\frac{f_{363}}{f_{238} - f_{869}} - f_{238} + f_{869} \right)}{f_{175} (f_{911} + f_{238})}$$

(b) Function2

図 B.2: 第 3 層において最適化された関数の例

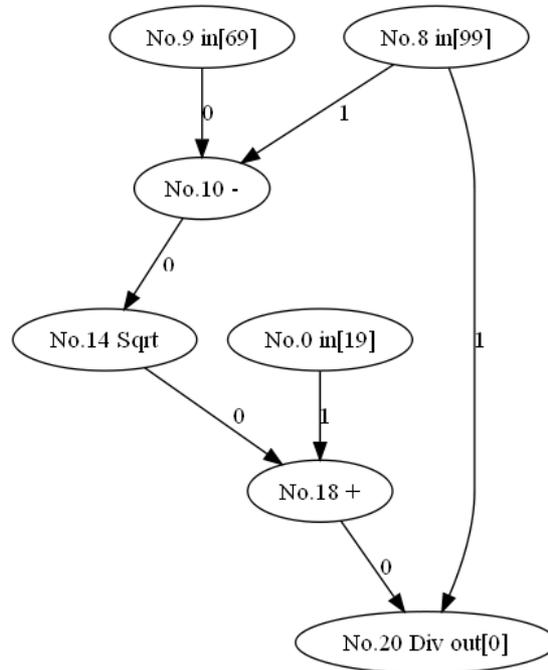


$$f_{29} - f_{98} - \exp(f_{45})$$

(a) Function1

$$\log(f_{45}) + \sqrt{f_{37}}$$

(b) Function2



$$\frac{\sqrt{f_{69} - f_{99}} + f_{19}}{f_{99}}$$

(c) Function3

図 B.3: 第 4 層において最適化された関数の例