

平成 29 年度 博士学位論文

多次元属性値可視化による  
対話的メディアコンテンツインタフェース

お茶の水女子大学大学院

人間文化創成科学研究科 理学専攻

齊藤 優理

平成 29 年 9 月

# 要 旨

本論文では、システムがユーザの嗜好を学習し、その学習過程を可視化することで、効果的にコンテンツを推薦、加工するためのユーザインタフェースを提案する。

近年、音楽、映画、本、雑誌、漫画、ゲーム、などのメディアコンテンツに対して、見たい放題、使いたい放題といったコンテンツ配信サービスが展開され、アクセスできるコンテンツの数が膨大に増えている。アクセスできるコンテンツの数が増えれば増えるほど、ユーザの嗜好に合ったコンテンツを見つけることが難しくなる。そこで、ユーザが膨大なコンテンツの中から効率よく自身の嗜好に合ったコンテンツを探しやすくするという課題に対して、推薦アルゴリズムの観点や検索のためのユーザインタフェースの観点などから、様々な研究がなされている。

また、メディアコンテンツを使ったコミュニケーションを可能にするソーシャル・ネットワーキング・サービス (SNS) の拡大により、デジタルカメラやスマートフォンで撮影したスナップ写真や映像などのコンテンツをネットワーク上で簡単に共有することができるようになった。ユーザがネットワーク上にメディアコンテンツを公開する際、コンテンツのパラメータを補正してからアップロードすることがしばしばあるが、手間と時間がかかる。そこで、ユーザがパラメータを手間と時間をかけずにユーザの嗜好に合わせて調節しやすくするという課題に対して、クラウドソーシングの観点やヒューマンコンピュータの観点などから様々な研究がなされている。

本論文では、コンテンツから特徴を抽出し、その特徴量空間において、ユーザの嗜好に合うコンテンツの位置を探索したり、コンテンツの特徴量とコンテンツの補正パラメータの関係を学習したりする際に、可視化によるナビゲーションやユーザクラスタリングを導入することで、コンテンツ推薦やコンテンツ加工の結果に効率よくユーザの嗜好を反映するユーザインタフェースを提案する。著者は、ユーザの嗜好を学習するためのタスクが大きいと実用性が下がるため、タスクを小さくしたい、タスクを小さくすると少数の入力しか得られないため、適用可能な学習手法が限定されるという点に着目した。そこで、上記の2点を考慮したユーザインタフェースを検討する。

まず、音楽コンテンツを対象に、ユーザの嗜好に合わせたコンテンツ推薦のためのユーザインタフェース (MusiCube) を提案する。MusiCube では、まず、楽曲から特徴量を抽出し、それらを並べて多次元の特徴量空間を生成する。次に、その特徴量空間において、対話型進化計算を用いて、ユーザと計算機

の自然なインタラクションによってユーザの嗜好を学習する。この際に、ユーザに自身の嗜好の傾向を気づかせるために、学習過程の可視化を行うことで、少ないタスクからユーザの嗜好を推薦結果に反映することを実現する。被験者 10 名に MusiCube を使ってもらい、有用性を評価した。

続いて、写真コンテンツを対象に、ユーザの嗜好に合わせたコンテンツ加工のためのユーザインタフェース (CrowdRetouch) を提案する。CrowdRetouch では、まずサンプル学習画像をカテゴリ分類し、画像特徴量によって画像クラスタを生成する。次にある程度の人数の初期ユーザに対して、画像カテゴリ、画像クラスタごとに一定数のサンプル学習画像を手動で画像補正させ、その特性によって初期ユーザをクラスタリングする。続いて各ユーザクラスタに対して画像補正パラメータの回帰分析を適用し、その結果を画像一括補正に用いる。一方、新規ユーザは、各ユーザクラスタの回帰式から算出される加工パラメータによって生成された補正結果画像をブレンドし、自身が所属するユーザクラスタを特定する。ここで、各補正結果画像のブレンド率を Parallel Coordinate Plot で可視化することで、ユーザに好みの補正傾向を気づかせることができる。新規ユーザが所属するユーザクラスタが特定できたら、各ユーザクラスタの回帰式と、ブレンド率の学習結果を用いて、他の画像を一括補正する。このように、初期ユーザはユーザクラスタを生成するために、サンプル学習画像に対する手動補正に時間がかかるが、新規ユーザは少数のサンプル学習画像に対する補正結果をブレンドしながら自身が所属するユーザクラスタを探索し特定することで、少ない操作量によってその後の画像一括補正処理を実現する。被験者 26 名に CrowdRetouch を使ってもらい、有用性を評価した。

以上の 2 つのユーザインタフェースの評価結果より、ユーザの操作負担を最小限に抑えつつ、嗜好に合わせて、メディアコンテンツを推薦、加工するためのユーザインタフェースにおいて、可視化によるナビゲーションや、ユーザクラスタリングの導入が有用であることを示すことができた。

# abstract

This thesis presents visual user interfaces for effective media contents recommendation and adjustment. They visualized processes to learn users' preferences for media contents. In recent years, we are able to access variety of media contents easily in a large amount of media contents provided by subscription services which allow unlimited access to music, movies, books, magazines, cartoons, and games. As the amount of media contents increases, it has been more difficult to find suitable media contents which match users' preferences. Therefore, various techniques on this problem have been developed to make easier for users to efficiently search for media contents suited to their preferences from among enormous media contents, from the viewpoint of recommendation algorithms and user interfaces.

In addition, it has been easy to share media contents such as snapshot photos and videos taken by digital cameras and smartphones due to the expansion of social networking service (SNS) that enables communication with media contents. Here, it often takes time and effort to adjust parameters of media contents before users upload media contents on social networking service. Therefore, various techniques on this problem have been developed to make easier for users to adjust the parameters according to the users' preference without taking time and effort, from the viewpoint of crowdsourcing and human computing.

This thesis presents visual interfaces to reflect users' preferences efficiently to media contents recommendation and retouch. They extract features of media contents, and then visualizes the processes of searching for media contents and learning the correction parameters that suit the users' preferences in multidimensional feature spaces. As the scope of this thesis, it is not practical if systems require heavy tasks to learn the users' preferences. Meanwhile, applicable learning methods are limited if only a small number of inputs can be obtained while applying light tasks. Development of effective user interfaces is essential to solve this problem.

The former part of this thesis proposes MusiCube, a visual interface for music selection. MusiCube extracts numeric feature values from acoustic data of tunes, and arranges icons corresponding to the



tunes to generate a multidimensional feature space. Next, in the multidimensional feature space, user's preference is learned by using the interactive evolutionary calculation. MusiCube enables to reflect users' preferences to the recommendation results with light tasks, by visualizing the learning processes in order to make the users aware of their preference tendency. We conducted user evaluation with 10 participants with MusiCube to demonstrate its usefulness.

The latter part of this thesis proposes a new image retouch system "CrowdRetouch" which reflects users' tendency of image retouch for a set of similar photographs. CrowdRetouch firstly asks initial users to manually retouch sample training images, and then divides the initial users based on the image retouch parameters. It then applies a regression analysis to each of user clusters to solve the relationship between the retouch parameters and image features, and automatically retouches rest of similar photographs based on the regression analysis results. After forming the user clusters, CrowdRetouch specifies the clusters of new users with smaller number of training images by visualizing the learning processes, and therefore we do not need to require heavy preprocesses to the new users. CrowdRetouch realizes personalized automatic image retouching to large number of photographs while reflecting preferences of novice users.

This thesis concludes that visualization and navigation of learning process of users' preference are useful while users search for and retouch media contents, based on the evaluation results of the presented two user interfaces.

# 目次

<b>第 1 章 緒言</b>	<b>1</b>
1.1 背景	1
1.2 本論文の貢献	4
1.3 本論文の構成	5
<b>第 2 章 関連研究</b>	<b>6</b>
2.1 はじめに	6
2.2 コンテンツ推薦技術	6
2.2.1 協調フィルタリング	6
2.2.2 内容ベースフィルタリング	7
2.3 コンテンツ検索インタフェース	7
2.3.1 コンテンツのサムネイル作成	7
2.3.2 2次元空間でのコンテンツ提示	8
2.3.3 多次元空間でのコンテンツ提示	9
2.3.4 ユーザインタラクションによる検索	11
2.4 コンテンツのパラメータの最適化	12
2.4.1 クラウドソーシングを用いた解析	12
2.4.2 ヒューマンコンピューテーションを用いた探索	12
2.4.3 クラウドソーシングを用いたヒューマンコンピューテーションを用いた探索	13
2.5 むすび	14
<b>第 3 章 MusiCube の処理手順</b>	<b>15</b>
3.1 はじめに	15
3.2 背景	15
3.3 概要	16
3.4 処理手順	17

3.4.1	システム概要 . . . . .	17
3.4.2	楽曲特徴量の検出 . . . . .	18
3.4.3	MusiCube の表示機能と操作機能 . . . . .	18
3.4.4	対話型進化計算を用いた楽曲提示 . . . . .	22
3.4.5	xy 軸に割り当てる最適な楽曲特徴量の提示 . . . . .	26
3.5	むすび . . . . .	27
<b>第 4 章</b>	<b>MusiCube の実行結果</b>	<b>28</b>
4.1	はじめに . . . . .	28
4.2	ユーザテストの方法 . . . . .	28
4.2.1	実行環境 . . . . .	28
4.2.2	実験方法 . . . . .	28
4.3	実験結果 . . . . .	29
4.3.1	ユーザの嗜好の傾向と類似性 . . . . .	29
4.3.2	最適な楽曲特徴量の提示 . . . . .	32
4.3.3	推薦精度の比較 . . . . .	36
4.3.4	プレイリスト機能の有効性 . . . . .	38
4.3.5	GUI の使いやすさ . . . . .	40
4.4	考察と課題 . . . . .	41
4.5	まとめ . . . . .	43
4.6	むすび . . . . .	43
<b>第 5 章</b>	<b>CrowdRetouch の処理手順</b>	<b>44</b>
5.1	はじめに . . . . .	44
5.2	背景 . . . . .	44
5.3	処理手順 . . . . .	45
5.3.1	画像カテゴリと画像クラスタの形成 . . . . .	46
5.3.2	サンプル学習画像の色補正 . . . . .	46
5.3.3	ユーザクラスタの生成 . . . . .	46
5.3.4	好みの計算モデルの算出 . . . . .	47
5.3.5	新しい画像のクラスタ判別のための学習 . . . . .	49

5.3.6	新しい画像, 新規ユーザに対する画像補正 . . . . .	49
5.4	むすび . . . . .	52
<b>第 6 章</b>	<b>CrowdRetouch の実行結果</b>	<b>53</b>
6.1	はじめに . . . . .	53
6.2	ユーザテスト . . . . .	53
6.2.1	実行環境 . . . . .	53
6.2.2	初期ユーザによるユーザクラスタ生成のための評価実験 . . . . .	53
6.2.3	新規ユーザによるサンプル学習画像の補正方法のための評価実験 . . . . .	54
6.2.4	新規ユーザによるクラスタ判別の評価実験 . . . . .	55
6.3	実験結果 . . . . .	55
6.3.1	キーワードによる画像カテゴリの有効性 . . . . .	55
6.3.2	ユーザクラスタリングの有効性 . . . . .	56
6.3.3	画像特徴量による画像クラスタリングの有効性 . . . . .	57
6.3.4	ユーザクラスタごとの嗜好の傾向 . . . . .	59
6.3.5	初期ユーザのクラスタの精度 . . . . .	59
6.3.6	画像カテゴリごとの嗜好の傾向 . . . . .	60
6.3.7	新規ユーザによるサンプル学習画像の補正方法の比較 . . . . .	61
6.3.8	ブレンド率の可視化結果表示の効果 . . . . .	61
6.3.9	新規ユーザのサンプル学習画像の枚数の比較 . . . . .	62
6.4	考察と課題 . . . . .	62
6.5	まとめ . . . . .	63
6.6	むすび . . . . .	63
<b>第 7 章</b>	<b>結言</b>	<b>68</b>
7.1	本論文のまとめ . . . . .	68
7.2	実世界への適用における課題 . . . . .	70
<b>A</b>	<b>業績</b>	<b>73</b>
<b>B</b>	<b>参考文献</b>	<b>76</b>

# 目 次

1.1	ロングテール減少.	2
2.1	Music Icon の表示例.	8
2.2	MIST の表示例.	8
2.3	Islands of Music の表示例.	9
2.4	MusicMiner の表示例.	9
2.5	PlaySOM の表示例.	9
2.6	Globe of Music の表示例.	9
2.7	MusicGalaxy の表示例.	10
2.8	Parallel Coordinate Plot の表示例.	10
2.9	Scatter plot の表示例.	10
2.10	ImageCube の表示例.	11
2.11	Musiccream の表示例.	12
3.1	MusiCube の GUI 画面.	19
3.2	次元選択パネル.	20
3.3	(左) 楽曲アイコンを表示, (右)xy 軸に割り当てる楽曲特徴量を変化させる回転操作のアニメーション表示.	20
3.4	プレイリスト作成画面.	21
3.5	主成分分析の概念図.	22
3.6	個体群のデータ仕様.	23
3.7	対話型遺伝的アルゴリズムの処理の流れ.	24
3.8	BLX- $\alpha$ に基づく交叉法.	25
3.9	次元選択パネルでの最適な楽曲特徴量の表示.	27
4.1	交叉率 $c$ と突然変異率 $m$ の比較.	29
4.2	嗜好の傾向の MusiCube での可視化結果.	30

4.3	嗜好の類似性のヒストグラム表示. . . . .	32
4.4	嗜好の類似性の各ヒストグラムの中間値. . . . .	33
4.5	被験者 10 人 (A-J) の結果におけるエントロピーの最大値, 平均値, 最小値. . . . .	33
4.6	軸に割り当てる楽曲特徴量を切り替えた回数. . . . .	34
4.7	MusiCube による可視化結果, 線形判別分析による可視化結果. . . . .	36
4.8	ランダム推薦と MusiCube による推薦による推薦精度の比較. . . . .	37
4.9	MusiCube の操作のステップ数. . . . .	38
4.10	被験者 10 人 (A-J) のプレイリストの作成にかかる時間. . . . .	39
4.11	被験者 10 人 (A-J) のプレイリストで目的に合っている楽曲の割合. . . . .	40
4.12	被験者 10 人 (A-J) の結果におけるプレイリストの楽曲ジャンルの割合. . . . .	41
5.1	CrowdRetouch の初期ユーザのための GUI 画面. . . . .	47
5.2	トーンカーブのコントロールポイント. . . . .	48
5.3	画像クラスタとユーザクラスタ. . . . .	48
5.4	説明変数と目的変数の関係. . . . .	49
5.5	CrowdRetouch の新規ユーザのための GUI 画面. . . . .	50
5.6	Parallel Coordinate Plot による補正結果画像のブレンド率の可視化表示. . . . .	51
6.1	画像カテゴリによる分類の効果. . . . .	56
6.2	画像カテゴリによる分類の効果と画像特徴量によるクラスタリングの効果の比較. . . . .	56
6.3	ユーザクラスタリングの効果. . . . .	57
6.4	画像特徴量による画像クラスタリングの効果. . . . .	58
6.5	画像特徴量による画像クラスタ分類なしで, 高い満足度を得た画像の例. . . . .	58
6.6	画像特徴量による画像クラスタ分類ありで, 高い満足度を得た画像の例. . . . .	59
6.7	ユーザの嗜好の傾向の例. . . . .	60
6.8	初期ユーザのユーザクラスタの精度とデンドログラムによるクラスタリングの結果. . . . .	64
6.9	画像カテゴリごとの補正パラメータの補正量. . . . .	65
6.10	補正パラメータの推定方法の比較. . . . .	66
6.11	ブレンド率の可視化結果表示の効果. . . . .	66
6.12	サンプル学習画像に対して補正にかかる合計操作時間の平均. . . . .	66
6.13	サンプル学習画像 1 枚あたりの補正操作回数. . . . .	67

6.14 新規ユーザのサンプル学習画像の枚数の比較. . . . .	67
------------------------------------	----

# 表 目 次

1.1	コンテンツを表す情報. . . . .	4
1.2	本論文で対象とするコンテンツと処理. . . . .	5
3.1	楽曲特徴. . . . .	18
4.1	選曲する時に用いる情報. . . . .	38
7.1	各提案手法の特徴. . . . .	69



# 第1章 緒言

## 1.1 背景

通信技術やネットワークインフラの進展に伴い、最近では、音楽、映画、本、雑誌、漫画、ゲーム、などのメディアコンテンツに対して、見たい放題、使いたい放題といったコンテンツ配信サービスが展開され、アクセスできるコンテンツの数が膨大に増えている。例えば、音楽配信サービスの Apple Music<sup>1</sup>では 3000 万曲もの音楽、書籍配信サービスの Kindle Unlimited<sup>2</sup>では 130 万本もの本や雑誌が配信されている。

インターネット上でコンテンツを配信するサービスの特徴として、ロングテール現象というものがある。Figure 1.1 のように、コンテンツの人気度を、アクセス回数を縦軸に、コンテンツを横軸にして、アクセス回数が多いコンテンツを左側から順に並べると、あまりアクセスされないコンテンツが右側になだらかに長く伸びるグラフとなる。アクセス回数は指数的に減少していくが、最後までほとんどゼロにはならず、ほぼ全てのコンテンツが誰かにアクセスされていることになる。そのため、ユーザがコンテンツを探す時、コンテンツを発見できる範囲を広げておくことが重要である。

このようなサービスでは、ユーザがコンテンツを探すための手段を提供している。例えば、タイトルなどのメタデータを利用する方法、ランキング情報を利用する方法、システムがユーザの嗜好や状況に合わせておすすめのコンテンツを推薦する方法などがある。タイトルなどのメタデータを利用する方法は、ユーザが何らかのキーワードを知っているコンテンツしか探索の対象とならない。また、ランキング情報を利用する方法は、人気のあるコンテンツしか探索の対象とならない。つまり、これらの方法では、アクセスできるコンテンツがたくさんあるにも関わらず、発見できるコンテンツの範囲が限定されてしまう。

システムがユーザの嗜好や状況に合わせておすすめのコンテンツを推薦する方法は、コンテンツ推薦技術によって実現されている。ユーザに適切なコンテンツを推薦する手法に、協調フィルタリングと内容ベースフィルタリングがある。協調フィルタリングは、推薦対象となるコンテンツについての情報や知識を必要とせず、ユーザの行動履歴を収集し、同じ嗜好を持ったコミュニティが共通して好むコンテ

---

<sup>1</sup><https://www.apple.com/jp/music/>

<sup>2</sup><https://www.amazon.co.jp/kindle-dbs/hz/signup>

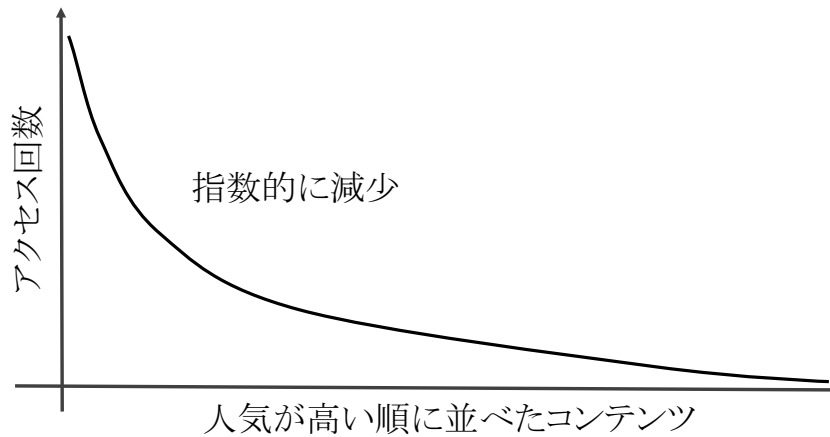


Figure 1.1: ロングテール減少.

コンテンツを提示する手法である。そのため、推薦の対象となるコンテンツは、同じ嗜好を持ったコミュニティの誰かがアクセスしたコンテンツであり、誰にもアクセスされていない新しいコンテンツは推薦の対象とならない。内容ベースフィルタリングは、推薦対象となるコンテンツの特徴とユーザの嗜好情報を比較して、ユーザが好むと判断したコンテンツを提示する手法である。コンテンツの特徴を自動的に抽出できれば、全てのコンテンツが推薦の対象となる。内容ベースフィルタリングでは、コンテンツの特徴を考慮するため、協調フィルタリングに比べて、より細かい単位でユーザの嗜好を推定することが可能となる。一方で、ユーザの嗜好情報は何かの方法で集める必要があり、嗜好情報を収集する際のユーザの操作負担を最小限に抑えることが重要である。

また、ユーザが大量のコンテンツの中から嗜好に合ったコンテンツをストレスなく取得するために、さまざまなユーザインタフェースが提案されている。例えば、コンテンツの印象を短時間で把握するために、コンテンツの特徴をサムネイル画像で表現する手法や、コンテンツ間の関係を把握するために、2次元、3次元、多次元空間上に、コンテンツの特徴に基づいてコンテンツを配置する手法がある。さらに、操作負担を軽減するために、ユーザインタラクションを用いてコンテンツを検索する手法がある。このことから、著者は、ユーザの操作負担を最小限に抑えるためには、コンテンツ群の見せ方を工夫したユーザインタフェースを提供することが重要であると考える。

一方、デジタルカメラやスマートフォンで撮影したスナップ写真や映像、自作曲などのコンテンツを

簡単に共有できるサービスである Facebook<sup>3</sup>や Instagram<sup>4</sup>, YouTube<sup>5</sup>などが広まり, プロ, アマチュア問わず, いつでも, どこでも, 自ら制作したコンテンツを発信できるようになってきている.

コンテンツをネットワーク上で公開する際, パラメータを調節してからアップロードすることがしばしばある. 例えば, 自分で撮影した写真の場合, 写真補正アプリケーションに搭載されている自動補正機能や, 明るさ, コントラスト, 彩度などの画像補正パラメータを手動で補正する機能を使って, 写真をユーザの好みに仕上げる. 自動補正機能には, 多くのユーザが適切だと感じるように自動的に補正できるという利点があるが, 各ユーザの嗜好や各シーンの特性を反映することが難しい. 画像補正パラメータを手動で補正する機能は, 各ユーザの嗜好や各シーンの特性を反映できるという利点があるが, 多数のパラメータが存在する場合には, パラメータの調整に多くの時間と労力が必要となる. また, よく知ったパラメータの場合は, パラメータを調整することによる効果が予測できるため, 余計な調整をしなくて済むことが多いが, 知らないパラメータの場合, 効果が予測できないため, いろいろなパターンで試す必要が出てくる. さらに, 日々の記録として撮影する料理写真のように, 枚数の多い写真群に対して一括補正する場合にも, 1枚ずつ手動操作で画像補正すると手間と時間を割かなくてはならない.

このことから, 著者はユーザの操作負担を最小限に抑えつつ, ユーザの嗜好に合わせて, 加工パラメータを最適化するための技術が重要であると考え. 効率的に最適なパラメータを見つける手法として, クラウドソーシングを用いた解析やヒューマンコンピューティングを用いた探索を適用するものがある. クラウドソーシングを用いた解析とは, クラウドソーシングによって収集したデータをモデル化したり, 機械学習の教師データとして利用したりして, 最適なパラメータを探索していく手法である. この手法は, 多くの人が良いと感じるパラメータセットを見つけることを目標としているため, 必ずしもユーザの嗜好を反映できるとは限らない. また, ヒューマンコンピューティングを用いた探索とは, ユーザがシステムによって提示されるコンテンツを評価することで, ユーザの嗜好に合わせて加工パラメータを最適化していく手法である. この手法は, 新規ユーザや新規コンテンツに対して加工パラメータを最適化するためには, 事前に該当ユーザによる学習データに対する評価データを集めておく必要があり, そのデータの数の増加に伴って, 精度が向上する. そのため, ユーザが手間をかけて評価結果を入力する必要があり, 手間がかかるという問題がある. このことから, 著者は, 評価結果を収集する際のユーザの操作負担を最小限に抑えることが重要であると考えている.

---

<sup>3</sup><https://www.facebook.com/>

<sup>4</sup><https://www.instagram.com/>

<sup>5</sup><https://www.youtube.com/>

## 1.2 本論文の貢献

ユーザの操作負担を最小限に抑えつつ、嗜好に合わせて、メディアコンテンツを推薦、加工するためのユーザインタフェースを提案する。

本論文では、推薦、加工の結果に、ユーザの嗜好を詳細に反映するために、コンテンツを何らかの解析による特徴量で表現することとする。コンテンツを表す情報として、Table 1.1 のようなものが考えられる。

Table 1.1: コンテンツを表す情報。

データ	Pros	Cons
メタデータ	・ コンテンツの特徴抽出不要	・ 詳細な嗜好を反映不可 ・ 人手による付与が必要 ・ 新規コンテンツ対応不可
他者の嗜好情報	・ 十分な評価履歴があれば嗜好を反映可 ・ コンテンツの特徴抽出不要	・ 詳細な嗜好を反映不可 ・ 新規ユーザ対応不可 ・ 新規コンテンツ対応不可 (cold-start 問題)
解析による特徴量	・ 詳細な嗜好を反映可 ・ 自動抽出可 ・ 新規ユーザ対応可 ・ 新規コンテンツ対応可	・ コンテンツの特徴抽出必要

何らかの解析による特徴量を採用することで、推薦、加工の結果に、詳細な嗜好を反映できる可能性が高く、特徴量を自動抽出することが可能で、新規ユーザや新規コンテンツにも対応ができるというメリットがある。

本論文の概要を示す。まず、コンテンツから何らかの解析によって特徴量を抽出し、それらを並べて多次元の特徴量空間を生成する。次に、特徴量空間において、ユーザの嗜好に合うコンテンツの位置を探索したり、特徴量とコンテンツの加工パラメータの関係を学習したりし、その学習過程を可視化する。最後に、学習過程の可視化と学習結果を用いて、コンテンツを推薦、加工することで、ユーザの操作負担を最小限に抑えつつ、嗜好に合わせた推薦結果や加工結果を提供する。

本論文の着眼点は下記の2点である。

- ユーザの嗜好を学習するためのタスクが大きいと実用性が下がるため、タスクを小さくしたい
- 小さいタスクからは少数の入力しか得られないため、適用可能な学習手法が限定される

これらの制約を解決するため、少数の入力からの推薦、加工を効果的に行う仕組みとして、

- 学習過程の可視化
- ユーザクラスタリング

を導入する.

本論文では, Table1.2 で示すように, メディアコンテンツとして, 「音楽」と「写真」を取り上げ, 音楽と推薦, 写真と加工についての提案手法を説明していく.

Table 1.2: 本論文で対象とするコンテンツと処理.

コンテンツ	推薦	加工
音楽	○	
写真		○

### 1.3 本論文の構成

以下に本論文の構成を述べる. 第2章では, ユーザの嗜好に合わせたメディアコンテンツの推薦, 加工についての関連研究を紹介する. 第3章では, ユーザの嗜好に合わせて音楽を推薦するためのユーザインタフェース (MusiCube) の処理手順について説明し, 第4章では, MusiCube の実行結果および評価実験に関する考察を述べる. 続いて, 第5章では, ユーザの嗜好に合わせて写真を加工するためのユーザインタフェース (CrowdRetouch) の処理手順について説明し, 第6章では, CrowdRetouch の実行結果および評価実験に関する考察を述べる. 最後に, 第7章で総括することで, 本論文をまとめる.

## 第2章 関連研究

### 2.1 はじめに

本章では，ユーザの操作負担を最小限に抑えつつ，嗜好に合わせて，メディアコンテンツを推薦，加工するための手法の関連研究を紹介する．メディアコンテンツとして，音楽と写真に焦点を当て，コンテンツ推薦技術，コンテンツ検索インタフェース，コンテンツのパラメータの最適化の事例について述べる．

### 2.2 コンテンツ推薦技術

コンテンツ推薦技術は，膨大な情報の中からユーザの嗜好にあったコンテンツを提供する技術である．基本的な実現方式として，協調フィルタリング [Goldberg92] と内容ベースフィルタリングがある．

#### 2.2.1 協調フィルタリング

協調フィルタリングは，推薦対象となるコンテンツについての情報や知識を必要とせず，ユーザの行動履歴を収集し，同じ嗜好を持ったコミュニティが共通して好むコンテンツを提示する手法である．音楽を対象とした推薦システムにも多く活用されていて，学術研究では，[Shardanand95] [Hayes00] [French01] などがあり，商用システムでは，[Lastfm] [Genius] などがある．

一般に，協調フィルタリングにより精度の高い推薦を行うためには，多くのユーザに推薦システムが利用され，多くのコンテンツに関する行動情報を収集する必要がある．そのため，新規に追加されたコンテンツや新たに推薦システムを使い始めたユーザに対しては行動情報の少なさから類似する嗜好を持つユーザを発見できず，推薦の精度が低くなってしまう（cold-start 問題）．また，推薦対象として膨大なコンテンツが存在し，コンテンツとユーザの関係が希薄である場合，ユーザに関する情報が十分であっても精度の高い推薦を行うことは難しい（sparsity 問題）．さらに，楽しい曲やリラックスした曲というコンテンツの感じ方には個人差や変化があるにも関わらず，そのような嗜好の差や変化が推薦に反映されにくいなどの問題が指摘されている [Matsubara10]．

## 2.2.2 内容ベースフィルタリング

内容ベースフィルタリングは、推薦対象となるコンテンツの特徴とユーザの嗜好情報を比較して、ユーザが好むと判断したコンテンツを提示する手法である。コンテンツの特徴を自動的に抽出できれば、全てのコンテンツが推薦の対象となり、また、新たに推薦システムを使い始めたユーザでもコンテンツの特定の特徴に対する嗜好情報が得られれば精度の高い推薦が可能になる。そして、コンテンツの特徴を考慮するため、協調フィルタリングに比べて、より細かい単位でユーザの嗜好を推定することが可能になる。

音楽を対象とした推薦システムにも多く活用されていて、学術研究では、[Hoashi03] [Cho04] [Rho07] [Sotiropoulos08] などがあり、商用システムでは、[Pandora] [Musicoverly] などがある。音楽を対象とする場合、利用する音楽内容として、楽曲の音響的な特徴が使われることが多い。楽曲の音響的な特徴を用いる場合、音響的な特徴を並べて多次元の特徴量空間を生成し、その空間上で、ユーザの嗜好に合うコンテンツの位置を探索していく。探索の手法として、注目コンテンツに最も類似するコンテンツへの評価値に基づき嗜好を予測する  $k$  近傍法や、好きなコンテンツと嫌いなコンテンツを最もよく分離する識別平面を見つけるサポートベクターマシン、ユーザの嗜好情報から好きなコンテンツの特徴を学習し、学習したモデルに基づいて嗜好を予測する機械学習モデル（適合性フィードバック [Hoashi03]、ニューラルネットワーク [Sotiropoulos08]、進化計算 [Cho04] [Rho07]）を適用した事例がある。

内容ベースフィルタリングにより精度の高い推薦を行うためには、コンテンツの特徴とユーザの嗜好情報を作成する際に、どの特徴を利用するかを適切に判断する必要がある。また、推薦を行うためにはユーザによるコンテンツへの評価値を必要とし、評価値の数の増加に伴い推薦の精度が向上する。そのため、ユーザは推薦システムが利用できるようになるまでに手間をかけて評価値を入力しなければいけないという問題がある。

## 2.3 コンテンツ検索インタフェース

ユーザが、コンテンツ検索のプロセスにおいて、大量のコンテンツの中からストレスなく適切なコンテンツを取得するために、様々なユーザインタフェースが提案されている。音楽コンテンツに関する手法を紹介する。

### 2.3.1 コンテンツのサムネイル作成

ユーザがコンテンツの印象を短時間で把握し、目的のコンテンツを素早く見つけるために、視覚情報を用いてコンテンツを表現する手法がある。

Music Icon は、ユーザごとに、楽曲の印象に合った幾何学模様の画像を自動的に選択することができる [Kolhoff06] (Figure 2.1)．楽曲と画像の対応付けは、ニューラルネットワークを使って、音楽解析情報である Mel-frequency cepstral coefficients (MFCC) のパラメータと画像の幾何学情報の対応付けを学習することにより、自動的に行われる．

MIST は、ユーザごとに、楽曲の印象に合ったアイコン画像を自動的に選択することができる [Oda07] (Figure 2.2)．あらかじめアイコンとして用いる画像を用意し、楽曲と画像の印象を学習結果に基づいて推定し、各楽曲に対して印象の近い画像を自動選択する．

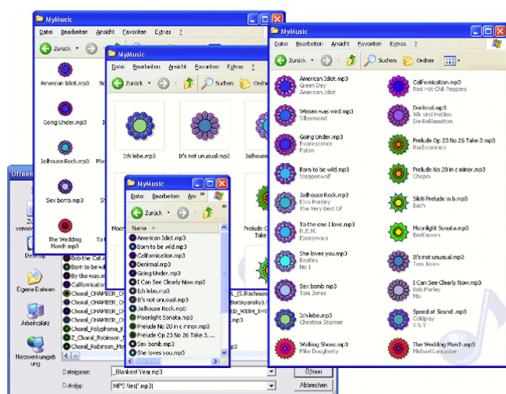


Figure 2.1: Music Icon の表示例.

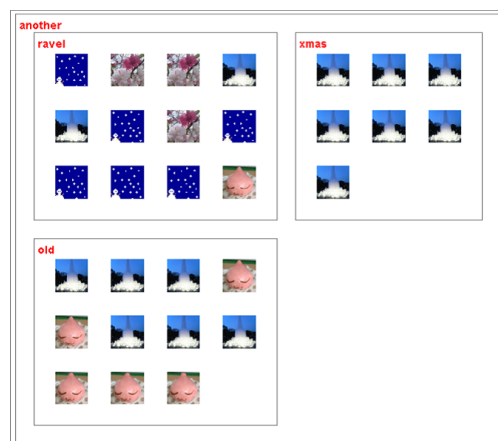


Figure 2.2: MIST の表示例.

これらの手法はコンテンツ単位の印象を視覚的に把握することを目指したものである．各楽曲に付与された画像を比較することで、楽曲が類似するか否かを把握することはできるが、ユーザ自身の嗜好の傾向がどのような特徴に起因しているかを把握することはできない．

### 2.3.2 2次元空間でのコンテンツ提示

検索対象となるコンテンツを2次元空間上に配置する研究が多く行われている．コンテンツの特徴に基づいてコンテンツ群を2次元座標上に配置する際、自己組織化マップ [Kohonen98] や主成分分析 [Jolliffe86]、多次元尺度構成法 [Kruskal78] などの次元削減手法がよく用いられる．

検索対象コンテンツを音楽とし、自己組織化マップを適用した代表的な手法として、Islands of Music [Pampalk01] (Figure 2.3)、MusicMiner [Morchen05] (Figure 2.4)、PlaySOM (Figure 2.5) や PocketSOMPlayer [Neumayer05]、Globe of Music [Leitich07] (Figure 2.6) などがある．

また、主成分分析を適用した代表的な手法として、浜脇らによる楽曲の雰囲気予測する楽曲検索システム [Hamawaki09] などがある．

そして、多次元尺度構成法の一つである Landmark Multidimensional Scaling (LMDS) を適用した手



法として, Stober らによる MusicGalaxy がある [Stober10] (Figure 2.7). MusicGalaxy では, LMDS を用いることで計算時間を軽減し, 平面上に配置された楽曲を対話的に操作することを可能とした. また, ユーザが楽曲の特徴 (音量, リズム, 明るさ) の度合を調節し, それぞれの要素に重みづけをして平面上に投影することができる.

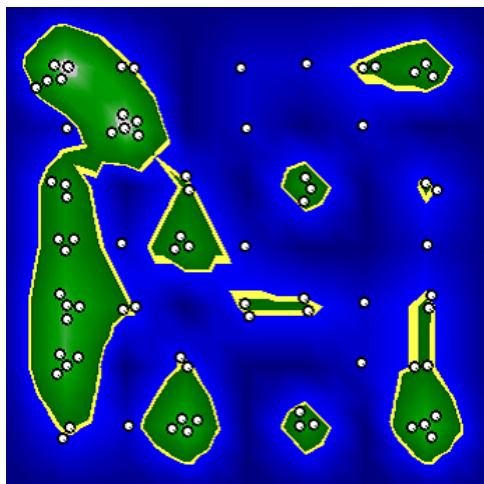


Figure 2.3: Islands of Music の表示例.

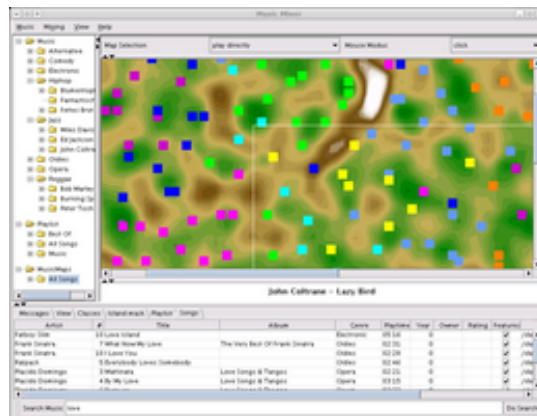


Figure 2.4: MusicMiner の表示例.

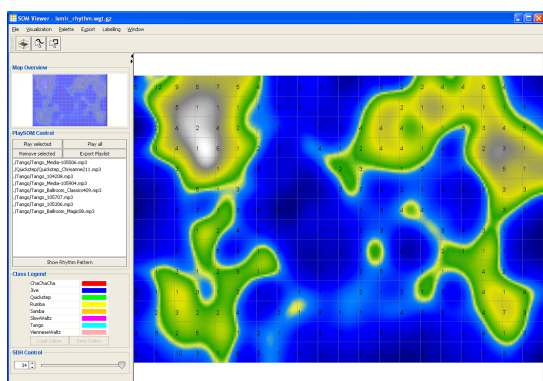


Figure 2.5: PlaySOM の表示例.

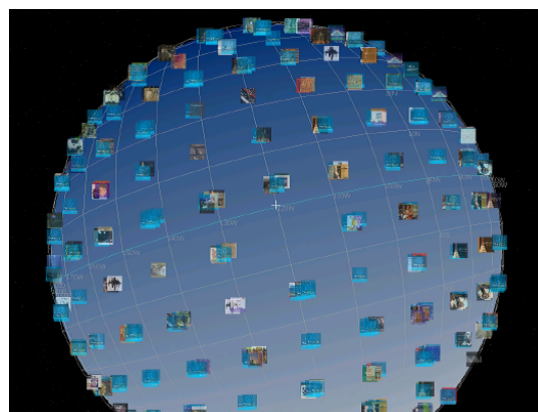


Figure 2.6: Globe of Music の表示例.

ただし, これらの手法は次元削減された特徴を用いてコンテンツを座標上に表示しているため, ユーザ自身の嗜好の傾向がどのような特徴に起因しているかを把握することはできない.

### 2.3.3 多次元空間でのコンテンツ提示

検索対象となるコンテンツを多次元空間上に配置する研究が多く行われている. コンテンツの特徴に基づいてコンテンツ群を多次元空間に配置する際, Parallel Coordinate Plot がある [Inselberg85] (Figure 2.8) や Scatter plot がある (Figure 2.9) がよく用いられる.



Figure 2.7: MusicGalaxy の表示例.

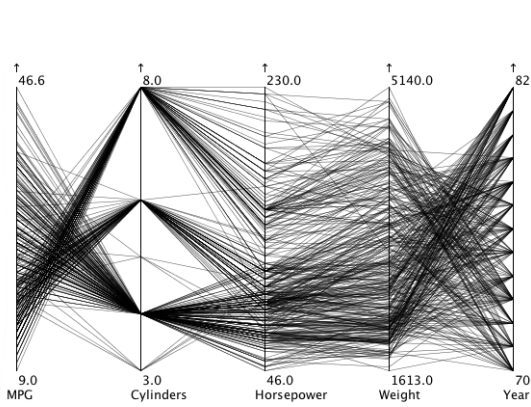


Figure 2.8: Parallel Coordinate Plot の表示例.

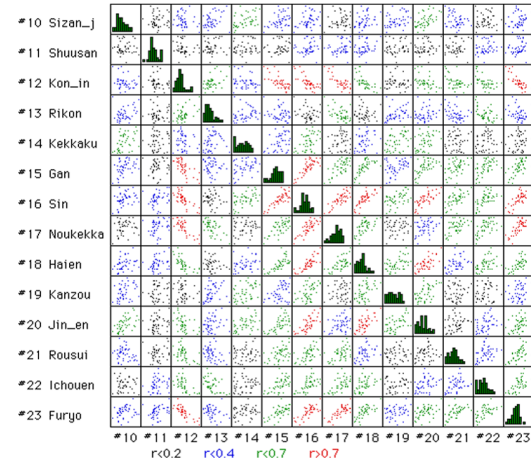


Figure 2.9: Scatter plot の表示例.

Parallel Coordinate Plot は、多くの変数にわたる統計的な傾向を把握しやすいが、表示結果が煩雑になりやすいという問題点がある。また、Scatter plot は、任意の 2 変数の相関性を把握しやすいが、次元の高いデータを適用する場合、表示領域が大きくなってしまいう問題点がある。

そこで、限られた表示面積で有益な可視化結果を得るために、次元削減手法を適用し本質的な情報を保持したまま多次元データを低次元空間で表示する手法がよく用いられてきた。しかし、次元削減手法を適用した手法では、縦軸と横軸に直感的に理解可能な数値を表現できないことが多い。この問題を解決するために、長崎らは、多次元データを特定の次元でクラスタリングして階層型データとして可視化する手法、およびその次元選択のために相関性の高い次元ペアを推薦する手法を提案した [Nagasaki09]。また、N.Elmqvist らは、多次元の中から 2 次元を対話的に選択し、直交座標系の xy 軸に割り当てて表示する Rolling the Dice を提案している [Elmqvist08]。さらに、Rolling the Dice における多次元データの表現形態を画像ブラウザに応用した例として、鄭らによる多次元データを有する大量画像群の一覧のための画像ブラウザ ImageCube がある [Zheng11] (Figure 2.10)。

本論文で提案する MusiCube に搭載された機能のうち、軸に割り当てる最適な楽曲特徴量を提示する

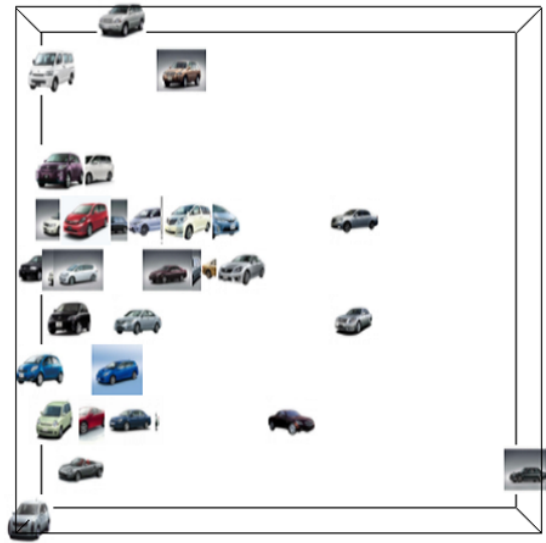


Figure 2.10: ImageCube の表示例.

機能は長崎らの次元推薦を拡張したものであり、次元選択に伴うアニメーション表示は Rolling the Dice における回転表示に類似するものである。

#### 2.3.4 ユーザインタラクションによる検索

ユーザとのインタラクションによって、効果的にコンテンツを探索するための研究が行われている。Musicream は、楽曲の雰囲気の色で表現し、画面上で楽曲を表す円盤を上から下へ流したり、複数の円盤をくっつけたりする操作により、山積みになった CD の中から手に取って聴くような感覚で音楽を楽しむことができる楽曲再生インタフェースである [Goto04] (Figure 2.11)。タイトルやアーティスト名などのメタデータに基づく楽曲検索を越えて、「何か聴きたい」というユーザの漠然とした欲求を満たすことができる。

Songrium は、不特定多数のユーザがコンテンツ間の関係を矢印のタグで自由に定義して共有するインタフェースである [Hamasaki14]。ユーザは、矢印タグが指す先にあるコンテンツを次々と視聴していくことで、潜在的に関心のあるコンテンツに出会う機会を増やすことが期待できる。

これらの手法は、ユーザとシステムのインタラクションによって、次々に新しいコンテンツに出会えるが、ユーザ自身の嗜好の傾向がどのような特徴に起因しているかを把握することはできない。

また、ユーザの嗜好に合うコンテンツを探索する際、進化計算を用いることがある [Ito09]。また、進化計算によって求められた解を可視化し、効率的な解探索を試みる研究もある。山代らの提案する進化計算の探索過程を可視化する手法では、探索過程において評価値空間における個体群の分布を可視化

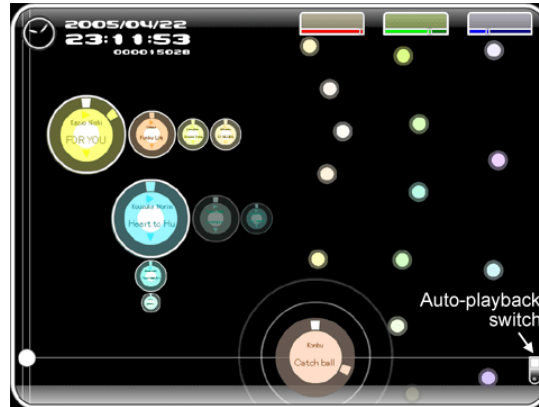


Figure 2.11: Musicream の表示例.

し、ユーザが探索の方向をインタラクティブに決定することによって、求める解を効率的に探索している [Yamashiro07]. このような手法は、本論文で提案する MusiCube の開発において、少ないタスクからユーザの嗜好を推薦結果に反映する方法として非常に有用であると考えられる.

## 2.4 コンテンツのパラメータの最適化

コンテンツのパラメータの最適化とは、パラメータ空間上で、最適なパラメータセットを見つけることである. 効率的に最適なパラメータを見つけるために、クラウドソーシングを用いた解析や、ヒューマンコンピュテーションを用いた探索を適用した事例がある.

### 2.4.1 クラウドソーシングを用いた解析

クラウドソーシングとは、インターネットを通して不特定多数の人に仕事を委託することである [Howe06]. クラウドソーシングを用いて収集されたデータはそのまま利用されることもあれば、機械学習の教師データとして利用する研究も近年多く提案されている. Jaroensri らは、クラウドソーシングによって収集したデータをもとに、画像補正結果に対しての良し悪しを自動的に予測する手法を提案している [Jaroensri15]. その他にも、ポートレート写真の評価 [Zhu14] などにも適用されている.

これらの手法は、多くの人が良いと感じるパラメータセットを見つけることが目標であり、必ずしもユーザの嗜好や意図を反映できるとは限らない.

### 2.4.2 ヒューマンコンピュテーションを用いた探索

ヒューマンコンピュテーションとは、人間とコンピュータを適切に組み合わせて問題解決する試みのことを指し、人間を計算資源と見放してシステムに組み込んでいく [Ahn08]. ヒューマンコンピュテ

ションを画像補正に適用した事例を紹介する。

Marks らは、ランダムに生成されたデザインや画像をユーザに提示し、ユーザはそこから目的に合うものを選択するという作業を繰り返し、パラメータを調整する手法を提案している [Marks97]。Shapira らは、候補の画像をユーザに提示し、混合ガウス分布を使って、色相と彩度を調整する手法を提案している [Shapira09]。これらは、必ずしもユーザの目的とする結果が含まれているとは限らない。また、荒川らは、ユーザに画像補正結果に対して補正操作を行ってもらい、対話型進化計算を用いて画像を補正する顔画像美観化フィルタシステムを提案している [arakawa07]。

これらの手法は、ユーザの嗜好に合わせてパラメータを調整できる可能性はあるものの、1 コンテンツごとにパラメータを調整するため、大量のコンテンツ群に対してパラメータを調整する場合には手間と時間がかかってしまうという欠点がある。

大量の写真群に対して、ユーザの意図や嗜好を効率的に反映した画像補正を実現するために、機械学習を用いた手法が提案されている。Kang らは、システムが複数の補正画像を提示し、ユーザが提示画像を評価することで、システムがユーザの嗜好を学習し、画像補正にユーザの嗜好を反映するシステムを提案している [Kapoor14]。ユーザの評価結果より、画像特徴量空間での画像間距離と、補正パラメータ空間での画像間距離が比例するような計算モデルを算出し、ユーザの嗜好を反映した画像補正パラメータを算出する手法を用いている。また、小山らが提案する SelPh では、ユーザが写真群の色補正パラメータを調整することで、システムがユーザの嗜好を学習し、補正パラメータの調整作業を効果的に支援する [koyama15]。ユーザの嗜好を反映した画像補正パラメータの算出方法は Kang らの手法と同様である。さらに、Bychkovsky らは、事前にプロの写真家に大量のサンプル画像を補正してもらい、ユーザの嗜好とプロの写真家の嗜好を比較し、その差分を画像補正に反映する手法を提案している [Bychkovsky11]。

これらの手法は、新規コンテンツに対して補正を行うためには、ユーザによる学習コンテンツに対する補正結果を必要とし、その補正結果の数の増加に伴って補正の精度が向上する。そのため、ユーザは補正システムが利用できるようになるまでに手間をかけて評価結果を入力しなければいけないという問題がある。

### 2.4.3 クラウドソーシングを用いたヒューマンコンピューテーションを用いた探索

ヒューマンコンピューテーションの実現方法として、クラウドソーシングで収集したデータを用いることが増えている [Greg10]。小山らは、クラウドソーシングを用いたヒューマンコンピューテーションによって収集したデータをもとに、パラメータ空間上で、デザインの「良し悪し」の空間分布を推定し、その空間分布を見ながら対話的にパラメータを最適化する手法を提案している [koyama16]。

この手法は、多くの人が良いと感じるパラメータセットを見つけることが目標であり、必ずしもユーザの嗜好や意図を反映できるとは限らない。

## 2.5 むすび

本章では、提案手法と関連するコンテンツ推薦技術、コンテンツ検索インタフェース、コンテンツのパラメータの最適化の事例について述べた。従来のコンテンツ推薦技術、コンテンツ検索インタフェースは、評価値の入力に手間がかかるという問題や、ユーザの嗜好の傾向がどのような特徴に起因しているかを把握できないという問題がある。また、従来のコンテンツパラメータの最適化技術として、クラウドソーシングや機械学習を用いた手法があるが、クラウドソーシングは多くの人が良いと感じるパラメータセットを見つけることが目標であり、必ずしもユーザの嗜好や意図を反映できるとは限らない。また、機械学習はユーザの嗜好情報を短時間で収集する仕組みがサポートされていなく、ユーザの嗜好の反映に時間がかかるという問題がある。

## 第3章 MusiCubeの処理手順

### 3.1 はじめに

本章では、ユーザの嗜好に合わせて音楽コンテンツを推薦するためのユーザインタフェース (MusiCube) の処理手順について述べる。

### 3.2 背景

通信技術やネットワークインフラの進展に伴い、音楽コンテンツ配信サービスが展開され、誰もがいつでもどこでも世界中の楽曲を聴くことができるようになった。音楽コンテンツ配信サービスの新たな形式として、従来のダウンロード課金型から、月額料金を支払うことで見放題で利用できる定額制 (サブスクリプション) サービスに変わってきている。その結果、1 曲ごとに音楽を買って聴くスタイルから、一定料金を払い、1 曲あたりの単価を気にせずに、気分などに合わせて音楽を自由に選んで聴くスタイルへと変化しつつある。

事前調査として著者は、236 人の学生に対して「気分に合わせて音楽を聴くことはあるか」というアンケートを実施してみたところ、219 人が Yes と答えた。その中でも、「気持ちを落ち着かせてリラックスしたい時」にはクラシック調の曲、「就寝時」には静かで穏やかな曲、「カフェにいる時」にはジャジーな曲、「部屋の掃除をする時」にはアップテンポで明るい曲などのように、その時の気分にあった楽曲を選んだり、YouTube などの類似曲検索を利用して音楽を聴いている人が多かった。

このように、気分などに合わせて音楽を探す場合、タイトルやアーティスト名などのメタデータによる検索、類似曲検索、プレイリストの共有、視聴履歴に応じたレコメンド機能などが使われる。著者が、レコメンド機能が搭載されている数種類の音楽コンテンツ配信サービス (Apple Music <sup>1</sup>, Spotify <sup>2</sup>, Google Play Music <sup>3</sup>) を利用したところ、おすすめの楽曲のタイトルなどのメタデータを提示するのみで、提示楽曲がどのような楽曲なのか、なぜこの楽曲が提示されるのか、わからないという欠点があるように感じた。そこで、著者は、楽曲の特徴を確認しながら一貫性のある選曲ができる楽曲提示インタフェースを開発することで、大量の楽曲の中からユーザの嗜好にあった楽曲を効率よく提示していきたいと考

---

<sup>1</sup><https://www.apple.com/jp/music/>

<sup>2</sup><https://www.spotify.com/jp/>

<sup>3</sup><https://play.google.com/store/music?hl=ja>



えている。

大量の楽曲の中から、ユーザに適切な楽曲を推薦するさまざまな手法が近年提案されている。ユーザに適切な楽曲を推薦する主な手法として、協調フィルタリングと内容ベースフィルタリングが挙げられる。協調フィルタリングは、推薦対象となる楽曲の特徴に関する情報を必要とせず、ユーザの行動履歴を収集し、同じ嗜好を持ったコミュニティが共通して好むコンテンツを提示する手法である。この手法は、大量の嗜好情報が集まればユーザにとって最適な楽曲を比較的精度よく推薦できるが、主な推薦対象曲は多くのユーザに聴かれている楽曲となる傾向にある。また、楽曲の特徴を考慮しないため、推薦結果に嗜好の個人差や変化を反映することが難しい。一方、内容ベースフィルタリングは、推薦対象となる楽曲の特徴とユーザの嗜好情報を比較して、ユーザが好むと判断した楽曲を提示する手法である。つまり、アーティストやジャンルに捉われず、ユーザの嗜好に合った楽曲と類似した曲調を持つ楽曲を推薦する。この手法は、コンテンツの特徴を自動的に抽出できれば、全てのコンテンツが推薦の対象となる。また、ユーザの嗜好情報は何らかの方法で集める必要があり、ユーザの操作負担をなるべく抑えることが重要である。

### 3.3 概要

本章では、対話型内容ベースフィルタリングの一手法として、楽曲の特徴に関するユーザの嗜好を反映させた選曲のための楽曲提示インタフェース "MusiCube" を提案する。例えば、カフェやドライブ、ランニングの BGM (Back ground Music) を選ぶ時のように、ユーザがその時の場面設定に合わせて楽曲を選択する際、ユーザは、アーティストやジャンルに捉われず、自身の嗜好を反映させて一貫性のある選曲をしたい、知っている楽曲やお気に入りの楽曲の他にもメロディーやタイトルさえも知らない楽曲を楽しみたいということがある。そこで、MusiCube では、ユーザと計算機の自然なインタラクションによってユーザの嗜好を学習し、その学習過程や学習結果を可視化することで、自身の嗜好の傾向を気づかせ、一貫性のある選曲を可能にする。このように可視化によるナビゲーションを導入することで、ユーザの操作負担を軽減しつつ、大量の楽曲の中からユーザの嗜好に楽曲を効率よく提示できると期待される。

MusiCube では、まず、楽曲データ (wave 形式) から多次元の楽曲特徴量を抽出し、それらを並べて、多次元空間を生成する。次に、抽出した楽曲特徴量に基づく特徴量空間において、ユーザの嗜好に合う楽曲の位置を対話的に学習する。ここで、ユーザの嗜好を学習する手段として、人間の感性を交えた最適化手法の対話型進化計算を適用する。MusiCube はユーザに提示楽曲を聴かせ、「目的に合う」「目的に合わない」の 2 択で評価させる。この「楽曲の提示」と「ユーザによる楽曲への評価」を繰り返すこと



で、ユーザの嗜好を学習する。さらに、この学習の過程や結果を可視化することで、学習を効率的に行えるように支援する。ユーザは、多次元空間上で、抽出した楽曲特徴量の中から任意の2次元の楽曲特徴量を選択し、それらの楽曲特徴量を Figure 3.1 に示す描画領域の xy 軸に割り当てる。続いて MusiCube は xy 平面上に、楽曲特徴量の値を座標値として、各楽曲をアイコンで表示する。アイコンには「目的に合う楽曲」、「目的に合わない楽曲」、「提示中の楽曲」、「未評価の楽曲」の4種類が用いられる。ここで、座標の2軸に割り当てる次元（楽曲特徴量）を切り替えると、アニメーション表示によってアイコンを再配置することができる。そして、「目的に合っている」と評価された楽曲群が画面上で最も局所集中して表示される次元（楽曲特徴量）を提示することができる。このように、多次元からなる楽曲特徴量を次元削減することなく、システムが自動的に任意の2次元の楽曲特徴量を可視化することで、ユーザにとって重要な楽曲特徴量を気づかせることが可能になる。

対話型進化計算を用いた楽曲推薦の関連研究 [Cho04] [Rho07] と比べて、MusiCube は、特徴量空間における学習過程や学習結果の可視化を積極的に導入することで、ユーザ自身の選曲傾向を気づかせ、選曲にある程度の一貫性を維持できるようなガイドラインを与えている点に特徴がある。著者は、RWC 研究用音楽データベース [RWC] の楽曲データなどを適用事例として、MusiCube の効果を検証した。

## 3.4 処理手順

本節では、MusiCube を構成する各処理の詳細を述べる。

### 3.4.1 システム概要

MusiCube の処理手順は、以下のとおりである。

1. 楽曲データから特徴量を抽出する。
2. 特徴量空間に楽曲をアイコンで表示する。
3. 楽曲を特定のアイコンで提示する。
4. ユーザに提示楽曲を聴かせて評価させる。
5. ユーザの評価に基づいて嗜好を学習する。
6. 3. から 5. を繰り返す。
7. ユーザにとって最適な楽曲特徴量を提示する。
8. プレイリストを作成する。

### 3.4.2 楽曲特徴量の検出

本手法では、数値解析ソフトウェア MATLAB の上に実装された楽曲分析パッケージ MIRtoolbox [MIR] を用いて、wave 形式の楽曲データから 7 種類の楽曲特徴量を抽出した (Table 3.1)。

Table 3.1: 楽曲特徴.

特徴量	説明
RMS energy	音量の平均
Low energy	弱音の割合
Tempo	テンポ
Roll off	85 %を占める低音域の値
Spectral irregularity	音質の変化の大きさ
Inharmonicity	根音に従っていない音の量
Mode	major と minor の音量の差

MusiCube では、これらの特徴量を同等に扱うため、楽曲特徴量  $f$  を正規化した楽曲特徴量  $f'$  を用いる。ここで、 $f' = (f - f_{min}) / (f_{max} - f_{min})$  とし、 $f_{max}$  と  $f_{min}$  は楽曲特徴量の最大値と最小値である。ただし、楽曲には次第に印象が変化するものや複数の印象を持つものがあり、楽曲全体の特徴量を一つに定めることは困難である。現段階では、楽曲における 0 分 30 秒から 0 分 40 秒の 10 秒間から得た特徴量を楽曲全体の特徴量として暫定的に定めている。適用事例として用いた楽曲データベースにおいて、各楽曲のイントロの長さを計測した結果、平均の長さが 28.5 秒だったため、イントロが終了して A メロが始まった 0 分 30 秒から 0 分 40 秒の区間を対象とした。

### 3.4.3 MusiCube の表示機能と操作機能

本節では、MusiCube を構成する各機能（楽曲アイコン表示機能、楽曲提示機能、プレイリスト機能）について説明する。MusiCube の GUI を Figure 3.1 に示す。画面（左側）の描画領域に楽曲群をカラーアイコンで表示し、画面（右側）にパネルを切り替えるためのタブを表示する。

#### アイコンについて

アイコンの色と形状は、以下のように割り当てる。

- 黄色の丸： ユーザがまだ聴いていない楽曲、または評価していない楽曲。
- オレンジ色の星： MusiCube によって提示された楽曲。
- 赤色の丸： ユーザが目的に合っていると評価した楽曲。



Figure 3.1: MusiCube の GUI 画面.

- 青色の丸： ユーザが目的に合っていないと評価した楽曲.

### 楽曲アイコン表示機能

ユーザは、Figure 3.2 に示す次元選択パネル上で、各楽曲が有する楽曲特徴量の中から、任意の2つの楽曲特徴量を選択する。MusiCube はそれらの2つの楽曲特徴量を、Figure 3.3(左) に示すように Scatter plot の xy 軸に割り当てる。この xy 平面上にて、正規化した楽曲特徴量の値を座標値として、各楽曲をアイコンで配置する。

また、xy 軸に割り当てる楽曲特徴量を別のものに差し替える際に、一度 z 軸に楽曲特徴量を割り当ててから 90 度回転表示することにより、シームレスに楽曲特徴量を差し替えながらアイコン群を再配置することができる。この回転操作は Rolling the Dice [Elmqvist08] における Scatter plot の回転表示に類似したものであり、Scatter plot の2軸を切り替える操作によってアニメーション表示で3次元空間を回転させる。Figure 3.3(右) は回転操作によるアニメーション表示の例である。このような回転操作によって、ユーザは次元間の関係を把握しやすくなり、さまざまな楽曲特徴量の組み合わせでアイコン群の分布を容易に見ることができるようになる。

### 楽曲提示機能

MusiCube では、対話型進化計算を用いてユーザの嗜好を学習し、最適な楽曲を提示する。対話型進化計算については、3.4.4 節で詳しく説明する。まず、ユーザはオレンジ色のアイコンで提示された楽曲

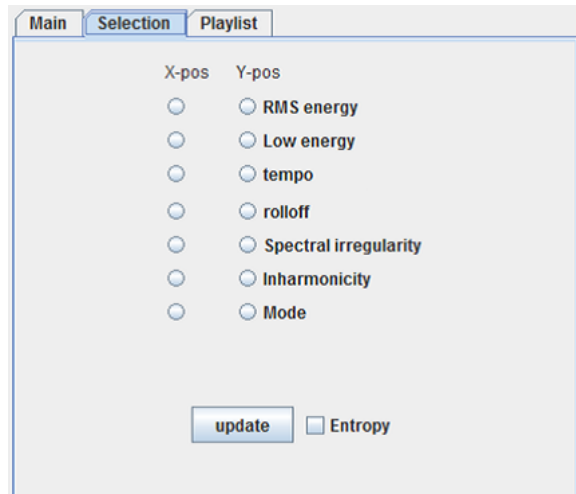


Figure 3.2: 次元選択パネル.

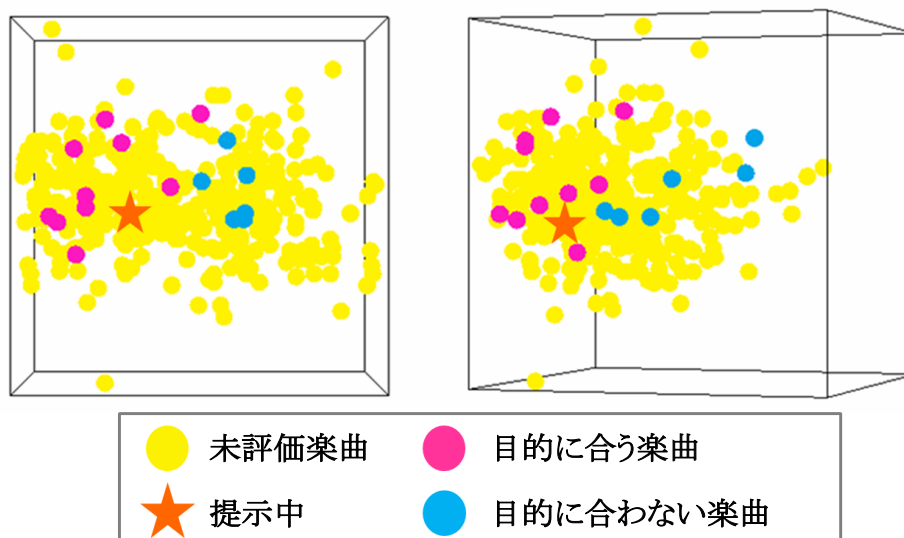


Figure 3.3: (左) 楽曲アイコンを表示, (右)xy 軸に割り当てる楽曲特徴量を変化させる回転操作のアニメーション表示.

を聴き、目的に合っているか否かを評価する。そのようにすると、目的に合っていると評価された楽曲は赤色のアイコンに変わり、目的に合っていないと評価された楽曲は青色のアイコンに変わる。このように、「MusiCube による楽曲提示」と「ユーザによる楽曲評価」を繰り返すことで、MusiCube はユーザの目的に合った楽曲を効率よく提示できるようになる。

「MusiCube による楽曲提示」と「ユーザによる楽曲評価」を繰り返し行い、提示楽曲への評価が集まったら、ユーザは対話型進化計算を一時停止し、立方体領域内における評価済みの楽曲間の相関を見

ることができる。これにより、ユーザは自身の嗜好の傾向を把握することができる。

## プレイリスト機能

MusiCube では、各楽曲に対する評価に基づいてプレイリストを作成する。ここで、プレイリストとは、楽曲をまとめてリスト化したものとする。ユーザは、立方体領域内において、マウスクリックした位置の周辺に配置された楽曲群をプレイリストに追加することができる。本手法ではユーザによって設定された  $xy$  座標平面上におけるユークリッド距離  $S$  を以下の式 3.1 で求め、一定範囲内（現段階では  $S \leq 10$  ピクセル）に属する楽曲を選択楽曲としてプレイリストに追加する（Figure 3.4）。

$$S = \sqrt{(x_{f_1} - x'_{f_1})^2 + (y_{f_2} - y'_{f_2})^2} \quad (3.1)$$

ここで、 $(x_{f_1}, y_{f_2})$  はユーザによって設定された  $xy$  座標平面上でのユーザがクリックした位置の座標値、 $(x'_{f_1}, y'_{f_2})$  は楽曲データの楽曲特徴量とし、 $f_1$  と  $f_2$  はユーザによって選択された楽曲特徴量とする。

ただし、プレイリストに追加できる楽曲は、赤色のアイコン（ユーザが目的に合っていると評価した楽曲）と黄色のアイコン（ユーザがまだ聴いていない楽曲、または評価していない楽曲）に対応する楽曲のみである。

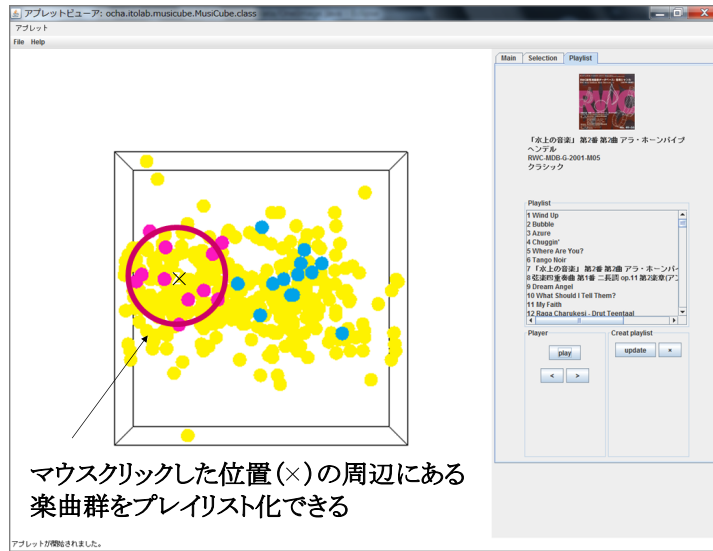


Figure 3.4: プレイリスト作成画面.

### 3.4.4 対話型進化計算を用いた楽曲提示

MusiCube は、正規化した特徴量空間において、対話型進化計算の一つである対話型遺伝的アルゴリズム（interactive Genetic Algorithm）を用いて、ユーザの嗜好を反映させた楽曲提示を行う。

#### 主成分分析による次元削減

MusiCube は、正規化した特徴量空間において、対話型進化計算の一つである対話型遺伝的アルゴリズム（interactive Genetic Algorithm）を用いて、ユーザの嗜好を反映させた楽曲提示を行う。本手法では、任意の2次元の楽曲特徴量を選択し、それらの楽曲特徴量を Figure 3.1 に示す描画領域の  $xy$  軸に割り当てている。しかし、遺伝的操作を行う際に、各楽曲が有する楽曲特徴量をそのまま遺伝子としてしまうと、次元数が多くなり探索性能が悪化してしまうおそれがあるため、対話型遺伝的アルゴリズムの遺伝子として、楽曲特徴量を主成分分析によって次元削減し、上位成分（第1～ $n$ 成分）の値を採用する。

主成分分析とは、相関もしくは分散共分散を利用し、複数の変数を統合して、データ全体の傾向や特徴を表す新たな変数を生成する手法である。

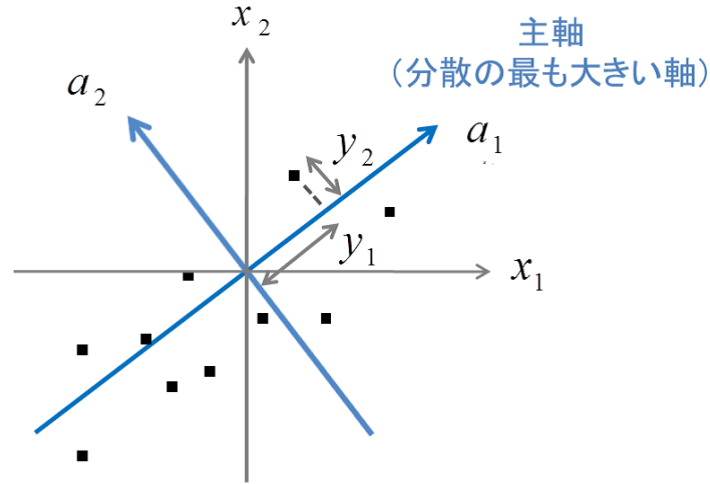


Figure 3.5: 主成分分析の概念図.

Figure 3.5 のように、 $n$  次元における主成分の軸を  $\{a_1, a_2, \dots, a_n\}$  とする。元の変量  $X = \{x_1, x_2, \dots, x_n\}^T$  の各軸への射影  $Y = \{y_1, y_2, \dots, y_n\}^T$  は以下の式 3.2 で与えられる。

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = a_1^T X \\ y_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = a_2^T X \\ \vdots \\ y_n = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = a_n^T X \end{cases} \quad (3.2)$$

変換行列  $A = \{a_1, a_2, \dots, a_n\}$  とすると,  $Y = AX$  となる. ここで, 変換行列  $A$  の単位ベクトルは,  $X$  の分散共分散行列  $S$  の固有ベクトルとして求めることができる. 著者は分散共分散行列が対称行列となることから, ヤコビ法を用いて固有ベクトルを算出している.

## 対話型進化計算のアルゴリズム

本手法では, 実数値遺伝的アルゴリズム (Real-valued Genetic Algorithms) を採用し, 遺伝子表現は, 実数値ベクトルとする. Figure 3.6 に個体群のデータ仕様を示す.

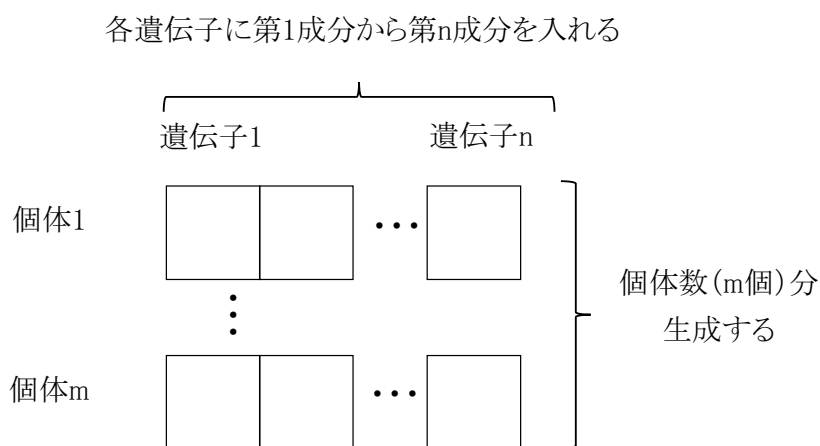


Figure 3.6: 個体群のデータ仕様.

世代交代モデルには, Minimal Generation Gap[Hiroyasu03] を採用する. ただし, 本手法では, 適合度を「目的に合っている」「目的に合っていない」の2値としているため, ユーザによって「目的に合っている」と判定された個体を中心に, 2個の親個体 (楽曲) を抽出して交叉するという処理を反復している. 対話型遺伝的アルゴリズムの処理の流れを Figure 3.7 に, その詳細を 1. から 8. に示す.

### 1. 初期集団生成

MusiCube では, 全楽曲の中から, あらかじめ定めた数の初期個体 (楽曲) を選択する. 初期個体として選択された個体 (楽曲) の数を個体数 ( $m$  個) と呼ぶ.

### 2. 提示

ユーザに対して個体 (楽曲) 群を提示する.

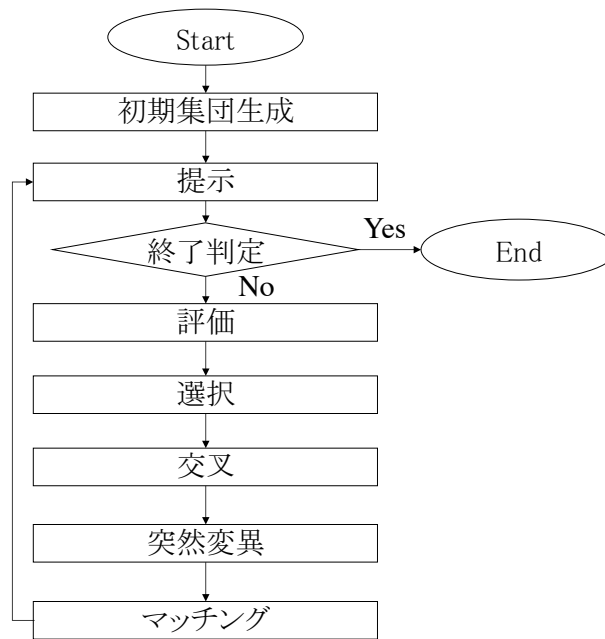


Figure 3.7: 対話型遺伝的アルゴリズムの処理の流れ.

### 3. 評価

ユーザに提示個体（楽曲）を2値（目的に合っているか否か）で評価させる。MusiCubeでは、収束時間を速めるため、一度評価した個体（楽曲）が再提示された場合、自動的に評価値を入力する。

### 4. 選択

ユーザによって「目的に合っている」と評価された個体（楽曲）を親個体（楽曲）とする。ここで、親個体（楽曲）の数は母集団の半数以上と定める。そのため、親個体（楽曲）の数が母集団の半数を満たない場合、母集団の半数を満たすまで、目的に合っていないと評価された個体（楽曲）の中からランダムに個体（楽曲）を選び、親個体（楽曲）とする。

### 5. 交叉

MusiCubeではまず、4.で得られた親個体（楽曲）のうち、総当たりで組み合わせを作る。次に、以下の式3.3によって、各組み合わせで2つの親個体（楽曲）間のユークリッド距離 $d$ を算出し、 $d$ が小さくなるような親個体（楽曲）の組み合わせを母集団の半数分、選ぶ。ここで、 $\alpha$ と $\beta$ は2つの親個体（楽曲）を表し、それぞれ楽曲特徴量を主成分分析によって次元削減して算出された上位成分（第1～ $n$ 成分）の値とし、 $n$ は遺伝子の長さとする。



$$d = \sqrt{\sum_{i=1}^n (\alpha_i - \beta_i)^2} \quad (3.3)$$

さらに，選ばれた2つの親個体（楽曲）の組み合わせで，BLX- $\alpha$ （ただし $\alpha=0$ ）[Hiroyasu03]に基づく交叉法を交叉率に従って適用する．親個体を持つ実数ベクトルの各変数の区間 $d_i$ を両側に $\alpha d_i$ だけ拡張した区間から一様乱数に従ってランダムに子個体を生成する．すなわち，親個体の周辺の各辺からなる直方体の領域が子個体の生成領域となる．Figure 3.8にBLX- $\alpha$ による子個体生成の例を示す．

$$c_{1i}, c_{2i} = u(\min(p_{1i}, p_{2i}) - \alpha d_i, \max(p_{1i}, p_{2i}) + \alpha d_i) \quad (3.4)$$

$$d_i = |p_{1i} - p_{2i}|$$

ここで， $\vec{P}_1 = (p_{11}, \dots, p_{1n})$ と $\vec{P}_2 = (p_{21}, \dots, p_{2n})$ は親個体， $\vec{C}_1 = (c_{11}, \dots, c_{1n})$ と $\vec{C}_2 = (c_{21}, \dots, c_{2n})$ は子個体， $u(x, y)$ は区間 $[x, y]$ の一様乱数を表す．また $\alpha$ はユーザが設定するパラメータである．

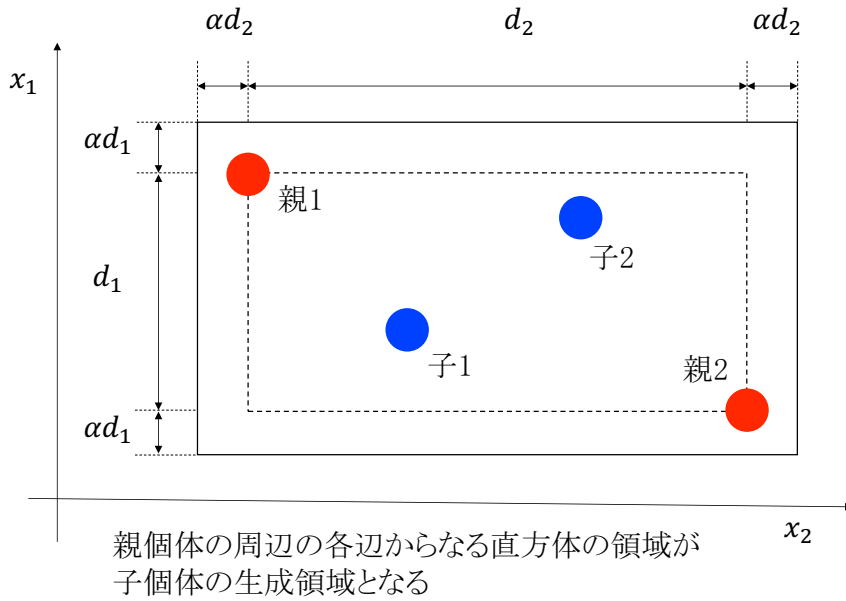


Figure 3.8: BLX- $\alpha$ に基づく交叉法.

## 6. 突然変異

個体群の多様性を維持するため，一様突然変異を適用する．具体的には，突然変異率にしたがって各子個体の遺伝子をランダムに変化させる．

## 7. マッチング

データベースに収録されている楽曲から得られる個体（楽曲）を 6. で得られた子個体とマッチングさせ、データベースから次世代の個体（楽曲）となるものを選択する．具体的には，以下の式 3.5 によって，6. で得た子個体と個体（楽曲）のユークリッド距離  $d$  を算出し， $d$  が最小となる個体（楽曲）を次世代に残す個体（楽曲）とする．次世代に残す個体の数は，あらかじめ定めた個体数（ $m$  個）となる．

$$d = \sqrt{\sum_{i=0}^n (f_i - p_i)^2} \quad (3.5)$$

ここで， $f$  は遺伝的操作で得られた楽曲特徴量， $p$  は楽曲データベースに収録されている楽曲データの特徴量を主成分分析によって次元削減して算出された上位成分（第 1～ $n$  成分）の値， $n$  は遺伝子の長さとする．

## 8. 終了判定

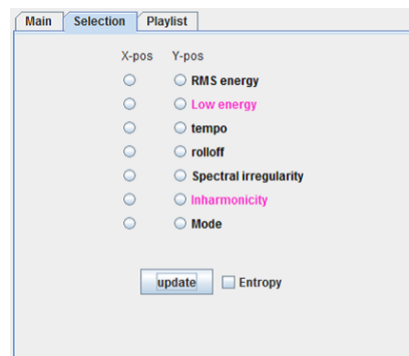
ユーザが求める個体（楽曲）が得られれば操作を終了する．そうでなければ，2. から 7. を繰り返す．

### 3.4.5 xy 軸に割り当てる最適な楽曲特徴量の提示

MusiCube では，赤色のアイコン（目的に合っている楽曲）群の密度がより高くなる楽曲特徴量を自動的に選択することができる．まず，特定の楽曲特徴量を xy 軸に割り当て，その表示領域内を  $N \times N$  個の短形領域で分割する．次に，各短形領域内において赤色のアイコンの数と赤色以外のアイコンの数を算出し，以下の式 3.6 よりエントロピーの合計を算出する．

$$\begin{aligned} E_{sum} &= - \sum_{i=1}^{N^2} (p_{ri} \log p_{ri} + p_{qi} \log p_{qi}) \\ p_{ri} &= r_i / (r_i + q_i) \\ p_{qi} &= q_i / (r_i + q_i) \end{aligned} \quad (3.6)$$

ここで， $i$  番目の分割領域における  $p_{ri}$  と  $p_{qi}$  は赤色のアイコンと赤色以外のアイコンの確率とする．本手法では，全楽曲特徴量を総当たりで xy 軸に割り当て各エントロピーを算出し，エントロピーが最も小さい値となった楽曲特徴量をユーザにとって最適な楽曲特徴量とする．このようにして選択された最適な楽曲特徴量を，次元選択パネルにおいて特定のカラーフォントで提示する（Figure 3.9）．



最適な楽曲特徴量を色付きで表示

Figure 3.9: 次元選択パネルでの最適な楽曲特徴量の表示.

### 3.5 むすび

本章では，MusiCube の処理手順について説明した．まず，3.2 節では MusiCube を提案するに至った背景について，3.3 節では MusiCube の概要について，3.4 節では MusiCube の処理手順について説明した．

## 第4章 MusiCubeの実行結果

### 4.1 はじめに

本章では，MusiCubeの実行結果および評価実験に関する考察を述べる．

### 4.2 ユーザテストの方法

#### 4.2.1 実行環境

楽曲特徴量の抽出には，MIRtoolbox [MIR] を使用した．また，Java JDK 1.6.0 を用いて MusiCube を実装し，Lenovo ThinkPad T510(CPU 2.4GHz, RAM 2.0GB) および Windows7 の上で実行した．システムの評価にあたり，楽曲データ 500 曲を用意した．なお，使用した楽曲データは，RWC 研究用音楽データベース（ポピュラー音楽，ジャズ音楽，音楽ジャンル）に収録されたもので，11 ジャンル（ポップス，ロック，ダンス，ジャズ，ラテン，ワールド，クラシック，ブラスバンド，声楽，邦楽，ア・カペラ）に及ぶ．

#### 4.2.2 実験方法

MusiCube は，ユーザとの自然なインタラクションによって，その時の気分に対応しい楽曲を直感的に選択することを目標としている．そこで，大学で情報科学を専攻する女子学生 10 人を対象に「MusiCube を使って，あるカフェをイメージして，そのカフェで再生したい楽曲を選ぶ」というテーマで実験を行い，実験終了後にアンケートを実施した．実験内容は以下のとおりである．

1. MusiCube によって提示された楽曲を聴く．
2. 提示楽曲に対して，カフェの BGM として合うか否かを評価する．
3. 1. と 2. を繰り返す．
4. プレイリストを作成する．

なお本実験の予備実験として，交叉率  $c$  と突然変異率  $m$  を変化させて，各世代において目的に合っている楽曲の数を検証したところ，交叉率  $c = 0.9$ ，突然変異率  $m = 0.1$  とした場合に，目的に合っている

楽曲がより多く提示された (Figure 4.1). この結果より, 本実験では, 個体数 10, 交叉率  $c = 0.9$ , 突然変異率  $m = 0.1$  と設定した.

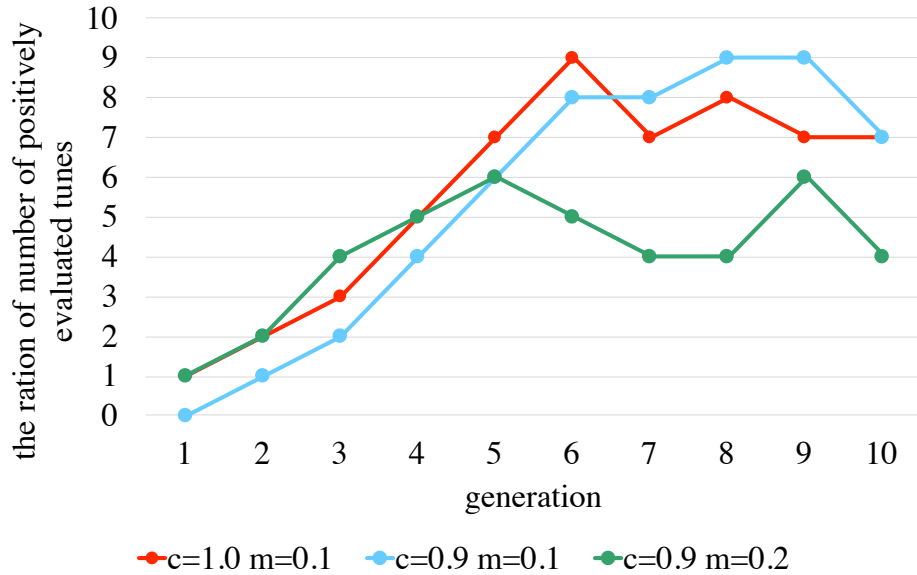


Figure 4.1: 交叉率  $c$  と突然変異率  $m$  の比較.

## 4.3 実験結果

### 4.3.1 ユーザの嗜好の傾向と類似性

被験者 4 人の実験結果に対して, 嗜好の傾向について検証する. 各被験者の可視化結果を Figure 4.2 に示す. 各図は赤色のアイコン (目的に合っている楽曲) 群の密度がより高くなる楽曲特徴量を  $xy$  軸に割り当てた表示結果である.

被験者 A (Figure 4.2(a)) と被験者 B (Figure 4.2(b)) は,  $x$  軸に割り当てられた楽曲特徴量 RMS energy の値が小さく,  $y$  軸に割り当てられている楽曲特徴量 Roll off の値も小さい楽曲を目的に合っている楽曲として評価している. RMS energy は音量の二乗平均の平方根を表す楽曲特徴量である (3.4.2 節). ロックやポップスなどで電子楽器が主導する楽曲の多くは, ほぼ一定の音量で曲が進むため RMS energy の値が大きくなる. 一方, アコースティック楽器で構成された楽曲は, 演奏上の抑揚や楽器の特性をもった音響的变化をもって曲が進み, 相対的に RMS energy の値が小さくなる. また, Roll off は, 85 %を占める低音域の値を表す楽曲特徴量である (3.4.2 節). そのため例えば, 高音域の倍音を多く含む金属楽器音 (例: タンバリン, シンバル) で構成された楽曲は Roll off の値が高くなる. このことから,

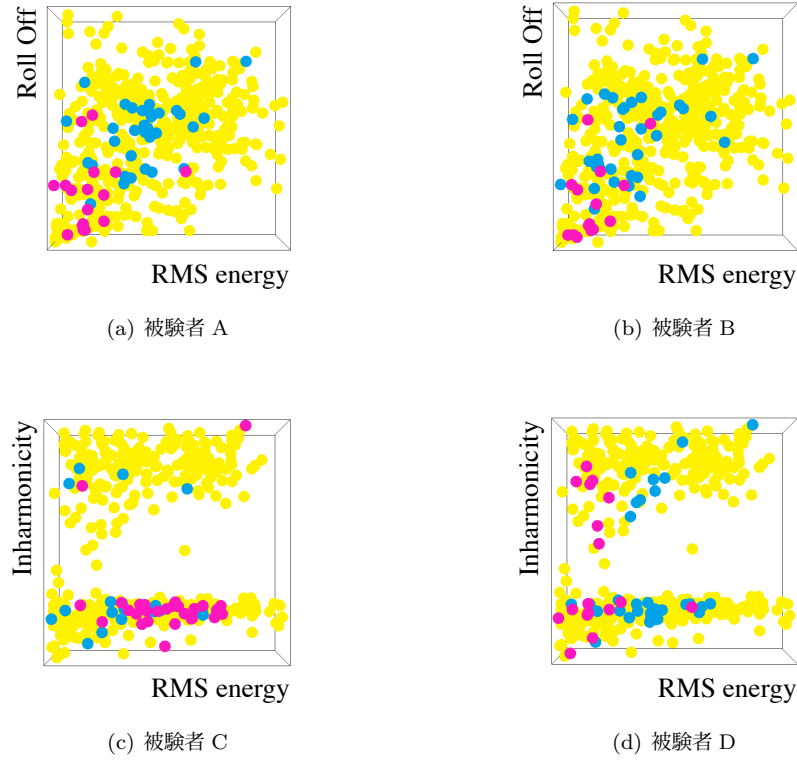


Figure 4.2: 嗜好の傾向の MusiCube での可視化結果.

被験者 A (Figure 4.2(a)) と被験者 B (Figure 4.2(b)) は、カフェの BGM に対して、アコースティック楽器で構成され、高音域の倍音成分が小さい楽曲を選んでいることが示唆される。

一方、被験者 C (Figure 4.2(c)) は、x 軸に割り当てられた楽曲特徴量 RMS energy の値が比較的大きく、y 軸に割り当てられている楽曲特徴量 Inharmonicity の値が小さい楽曲を目的に合っている楽曲として評価している。Inharmonicity は根音に従っていない音の量を表す楽曲特徴量である (3.4.2 節)。そのため、不協和音が多い楽曲は Inharmonicity の値が大きくなり、不協和音が少ない楽曲は Inharmonicity の値が小さくなる。このことから、被験者 C (Figure 4.2(c)) はカフェの BGM として、ロックやポップスなどの電子楽器で構成され、かつ不協和音の少ないシンプルな楽曲を選んでいることが示唆される。

また、被験者 C (Figure 4.2(c)) と被験者 D (Figure 4.2(d)) は、xy 軸に同じ楽曲特徴量が割り当てられているが、被験者 C (Figure 4.2(c)) は x 軸に割り当てられた楽曲特徴量 RMS energy の値が大きく、y 軸に割り当てられた楽曲特徴量 Inharmonicity の値が小さい楽曲を、被験者 D (Figure 4.2(d)) は x 軸に割り当てられた楽曲特徴量 RMS energy の値が小さく、y 軸に割り当てられた楽曲特徴量 Inharmonicity の値が小さい楽曲や大きい楽曲を目的に合っている楽曲として評価している。このことから、被験者 C (Figure 4.2(c)) は、電子楽器で構成されていて、不協和音が少ないシンプルな楽曲を選んだのに対して、

被験者 D (Figure 4.2(d)) は、「アコースティック楽器で構成されていて、不協和音が少ないシンプルな楽曲」と「アコースティック楽器で構成されていて、不協和音が多い（例えばジャズ風の）楽曲」という 2 種類の楽曲をカフェに合う音楽として選んでいることが示唆される。被験者 D (Figure 4.2(d)) において 2 種類の楽曲を選んだのが観察されるように、ユーザの嗜好を学習するアルゴリズムとして対話型進化計算を用いることで、「嗜好の多峰性」にも対応できていることが読み取れる。

次に、被験者 4 人の実験結果に対して、嗜好の類似性について検証する。以下の式 4.1 より任意の 2 人の被験者（被験者 X と被験者 Y）のうち、被験者 X の赤色のアイコン（目的に合っている楽曲）に対して、被験者 Y の全ての赤色のアイコン（目的に合っている楽曲）のユークリッド距離  $d$  を算出し、その平均値をヒストグラムに加算する。この処理を被験者 X の赤色アイコン（目的に合っている楽曲）全てに対して反復する。そして、被験者 Y の赤色のアイコン（目的に合っている楽曲）に対しても同様の処理を行う。

$$d = \sqrt{\sum_{i=0}^n (X_i - Y_i)^2} \quad (4.1)$$

ここで、 $X$  は被験者 X の赤色アイコン（目的に合っている楽曲）の楽曲特徴量、 $Y$  は被験者 Y の赤色アイコン（目的に合っている楽曲）の楽曲特徴量、 $n$  は楽曲特徴量の数とする。

被験者 4 人に対して任意のペアで算出したヒストグラムの結果を Figure 4.3 に示す。また、各ヒストグラムの中間値を算出した結果を Figure 4.4 に示す。ヒストグラムの中間値を算出した結果、被験者 A と B の中間値が最小となった。そのため、被験者間を比較すると、被験者 A と被験者 B の嗜好が類似していることが読み取れる。

また被験者に「その時の気分に対応しい楽曲をユーザの嗜好を反映させて選曲する場合、Figure 4.2 に示すような可視化結果は効果的であるか」というアンケートを実施したところ、5 段階評価で最大値 5、平均値 3.9、最小値 3 という結果を得た。

これらの結果より、「カフェに合う曲」というような特定の目的に対応しい楽曲を選ぶ場合において、MusiCube には可視化技術の搭載によってユーザに自身の嗜好がどのような特徴量に起因しやすいかを気づかせる効果があることがわかる。また、ユーザの嗜好を学習するアルゴリズムとして対話型進化計算を用いることで、被験者 D (Figure 4.2(d)) において 2 種類の楽曲を選んだのが観察されるように、「嗜好の多峰性」にも対応できていることがわかる。

しかし、被験者によっては赤色のアイコン（目的に合っている楽曲）群が局所集中して表示されなかったり、赤色のアイコン（目的に合っている楽曲）群と青色のアイコン（目的に合っていない楽曲）群が混在してしまう結果となった。

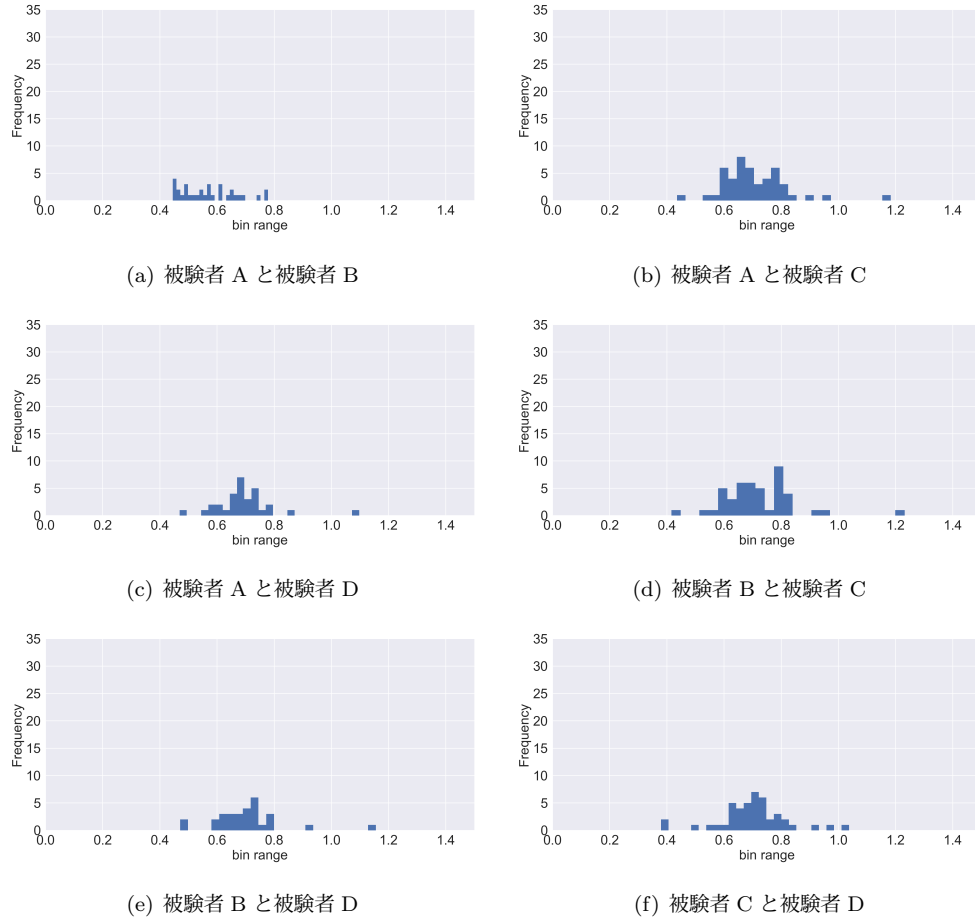


Figure 4.3: 嗜好の類似性のヒストグラム表示.

このような結果となった原因について検討したところ、雅楽などの古典音階からなる楽曲の扱い方や、楽曲から抽出した楽曲特徴量の種類や数に問題があったと考えられる。このような問題を解決するため、MusiCube 本来の特徴を活かしつつ、ジャンルや歌詞などのメタデータを補足的に利用するなど、ユーザインタフェースの見直しを行っていきたい。

#### 4.3.2 最適な楽曲特徴量の提示

各被験者の結果において、楽曲特徴量の任意のペアを総当たりで  $xy$  軸に割り当て、赤色のアイコンのエントロピーを算出した。MusiCube ではエントロピーを赤色のアイコン（目的に合っている楽曲）群が立方体領域内で局所集中して表示されているかを表す指標値として扱っており、実際にこの値が小さいほうが赤色のアイコンが局所集中する傾向にあることを目視確認している。Figure 4.5 にて、10 人の被験者の赤色のアイコンのエントロピーの最小値、平均値、最大値をグラフで示す。なおグラフの横軸は被験者を表し、縦軸はエントロピーを表す。この結果より、すべての被験者において、ランダムに楽曲特徴量を選択するより、エントロピーを最小にする楽曲特徴量のペアを  $xy$  軸に割り当てたほうが、赤



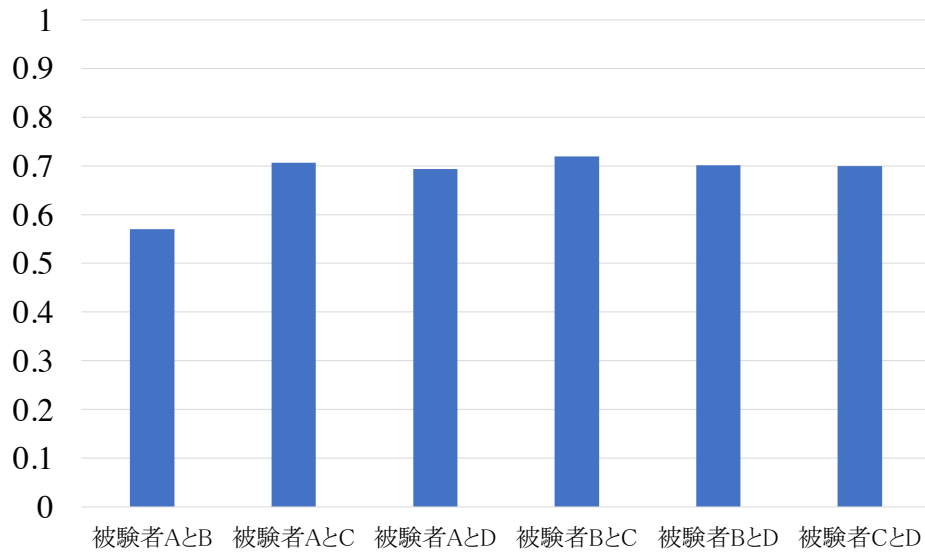


Figure 4.4: 嗜好の類似性の各ヒストグラムの中間値.

色のアイコンの密度が高くなることがわかった. このことから, MusiCube において最適な楽曲特徴量を提示する機能が有効であると言える.

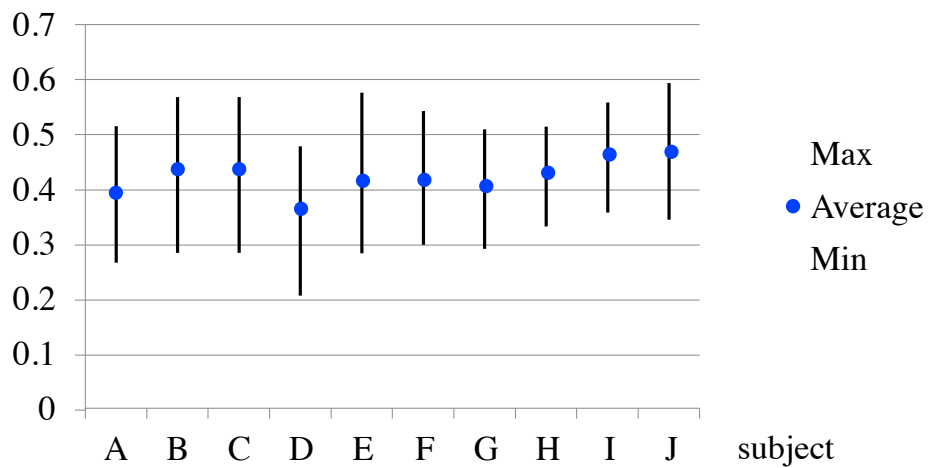


Figure 4.5: 被験者 10 人 (A-J) の結果におけるエントロピーの最大値, 平均値, 最小値.

また, MusiCube を使って可視化結果を表示しながら学習を進め, その学習結果を用いてプレイリス

トを作成する過程で、ユーザがどのくらいの頻度で xy 軸に割り当てる楽曲特徴量を切り替えたかについて検証する。Figure 4.6 に 2 人の被験者の結果を示す。なおグラフの横軸は時間経過を表し、縦軸は被験者を表す。

被験者 A は、対話型進化計算によって嗜好を学習している間に 2 回、xy 軸に割り当てる楽曲特徴量を切り替えていた。被験者 A に対して、最適な楽曲特徴量を提示する機能が有効であるかとヒアリングしたところ、評価した楽曲が増えてきた段階で、赤色のアイコンの密度が高くなるような楽曲特徴量のペアをシステムに提案してもらうことで、自身では気づきにくい嗜好の傾向を観察しながら、一貫性のある選曲ができたというコメントをもらった。一方、被験者 B は、嗜好を学習し始める前に 1 回、学習している間に 3 回、xy 軸に割り当てる楽曲特徴量を切り替えていた。被験者 B からは、自身の嗜好が当初予想していなかった楽曲特徴量に起因していることに気づいたというコメントをもらった。被験者 B は、学習開始前に、自身の嗜好に起因する楽曲特徴量を予想し、x 軸に RMS energy を、y 軸に Roll off を割り当てて、学習を開始した。学習時には複数回 xy 軸に割り当てる楽曲特徴量を切り替え、x 軸に Inharmonicity、y 軸に Roll off を割り当てた結果でプレイリスト作成した。この結果から、ユーザが任意の楽曲特徴量を xy 軸に割り当てることで、自身の傾向を予想しながら学習を進めることができ、さらに、システムで提示する最適な楽曲特徴量を xy 軸に割り当てることで、ユーザ自身では気づきにくい嗜好の傾向を提示できることがわかる。

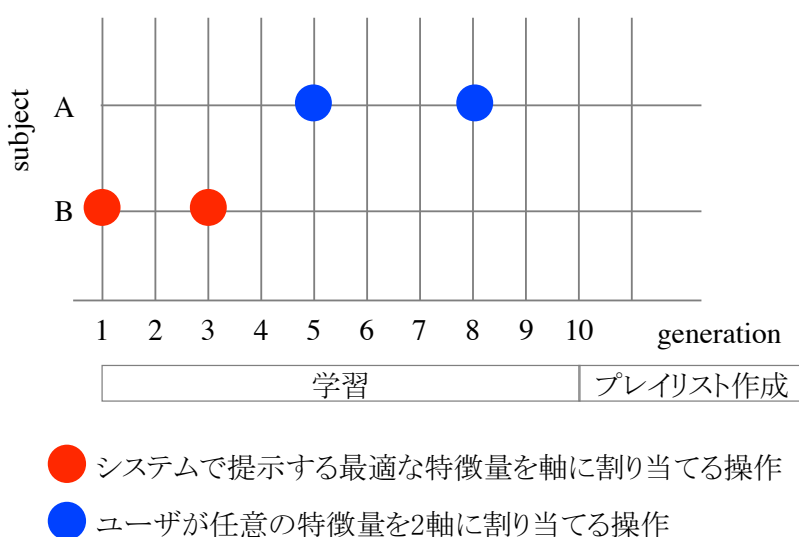


Figure 4.6: 軸に割り当てる楽曲特徴量を切り替えた回数.

ところで、MusiCube は多次元からなる楽曲特徴量を次元削減することなく、任意の 2 つの楽曲特徴量を xy 軸に割り当て、ユーザにとって重要な楽曲特徴量を気づかせるインタフェースとなっている。一方で、楽曲特徴量の次元削減によってアイコン群の位置を決定した方が有用な場合があるかもしれない。そこで、目的に合っている楽曲を表す赤色のアイコンが集まるような結果を表示するために、線形判別分析 [Choo09][Choo10] を適用し、本手法の結果と比較した。

線形判別分析とは、あらかじめ色分けされた点群が画面上でうまく分離するように低次元化する手法である。  $n$  次元における軸を  $\{b_1, b_2, \dots, b_n\}$  とすると、元の変量  $X = \{x_1, x_2, \dots, x_n\}^T$  の各軸への射影  $Y = \{y_1, y_2, \dots, y_n\}^T$  は以下の式 4.2 で与えられる。

$$\begin{cases} y_1 = b_{11}x_1 + b_{12}x_2 + \dots + b_{1n}x_n = b_1^T X \\ y_2 = b_{21}x_1 + b_{22}x_2 + \dots + b_{2n}x_n = b_2^T X \\ \vdots \\ y_n = b_{n1}x_1 + b_{n2}x_2 + \dots + b_{nn}x_n = b_n^T X \end{cases} \quad (4.2)$$

変換行列  $B = \{b_1, b_2, \dots, b_n\}$  とすると、 $Y = BX$  となる。ここで、クラス内共分散行列 (within-class covariance matrix)  $\sum_w$ 、クラス間共分散行列 (between-class covariance matrix)  $\sum_b$  とすると、変換行列  $B$  の単位ベクトルは、 $\sum_w^{-1} \sum_b$  の最大固有値  $\lambda$  に対する固有ベクトルとして求めることができる。著者は、 $\sum_w^{-1} \sum_b$  が対称行列となることから、ヤコビ法を用いて固有ベクトルを算出している。

Figure 4.7 に 2 人の被験者の結果において、「目的に合っている」と評価された楽曲群が立方体領域内で最も局所集中して表示されるような楽曲特徴量を xy 軸に割り当てて表示した場合 (Figure 4.7(a) と Figure 4.7(c)) と線形判別分析を適用し楽曲群を表示した場合 (Figure 4.7(b) と Figure 4.7(d)) のそれぞれの可視化結果を示す。

4.3.1 節でも論じたように、Figure 4.7(a) と Figure 4.7(c) は、xy 軸に同じ楽曲特徴量が割り当てられていて、赤色のアイコン (目的に合っている楽曲) の配置も似ていることから、被験者らの嗜好が同じような特徴に起因していて、類似した嗜好を持っていることが読み取れる。しかし、Figure 4.7(a) の被験者の結果に線形判別分析を適用し楽曲群を表示した Figure 4.7(b) と、Figure 4.7(c) の被験者の結果に線形判別分析を適用し楽曲群を表示した Figure 4.7(d) を比較しても、被験者らがどのような特徴に起因しているかということは読み取れない。つまり、本手法のように「目的に合っている」と評価された楽曲群が立方体領域内で最も局所集中して表示されるような楽曲特徴量を xy 軸に割り当てて表示した場合、ユーザは自身の嗜好がどのような特徴に起因しやすいかを把握することが可能になり、さらに可視化結果を比較することで被験者らの嗜好の差を分析することができる。しかし、線形判別分析によって楽曲群を表示した場合、軸に意味がなくなってしまうことから、ユーザがどのような特徴に起因して楽曲を選曲しているかを判断することが困難になる。この点からも MusiCube に利があるといえる。

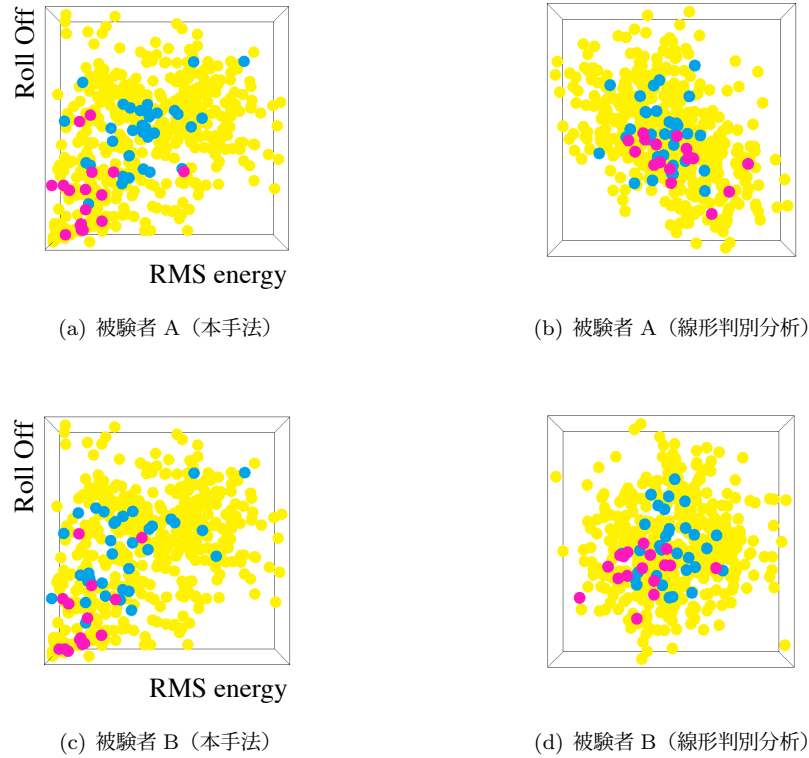


Figure 4.7: MusiCube による可視化結果，線形判別分析による可視化結果。

### 4.3.3 推薦精度の比較

MusiCube の推薦精度を評価するにあたり，まず，対話型進化計算における推薦精度について検証する．音楽推薦の手法として，協調フィルタリングとユーザの嗜好情報のモデリングによる内容ベースフィルタリングが挙げられるが，両者とも推薦の満足度を向上するためには時間を要する．協調フィルタリングでは，システムを利用するユーザ数が増えないと，推薦結果の満足度が上がらないため，時間がかかる．また，ユーザの嗜好情報のモデリングによる内容ベースフィルタリングでは，あらかじめユーザが好む楽曲を一定数集めないと，嗜好の傾向をモデリングすることができない．そのため，1 曲ずつ聴くことにより徐々に推薦精度があがる進化的な効果はなく，推薦を開始するまでに一定の時間がかかる．一方，MusiCube では，進化計算を採用することで，ユーザが 1 人しかいなくても推薦の満足度を上げることができ，また，1 曲ずつ聴いていくうちに徐々に推薦の満足度を上げることもできる．その結果，比較的早期に満足度が高い推薦結果が得られる可能性がある．そこで，評価楽曲数の増加に伴い，推薦精度（被験者 10 人の平均）がどのように変化したかについて，Figure 4.8 に示す．ここで，推薦精度は，各世代（10 個体）における目的に合っている楽曲の割合とする．縦軸は，1 世代（10 曲）中，目的に合っている楽曲と評価した楽曲の割合を，横軸は，世代を示している．評価楽曲数が少ない第 1 世代では，0.423

であるのに対して、評価楽曲数が多くなる第 10 世代では 0.925 となった。この結果より、対話型進化計算を適用することで、評価楽曲数の増加に伴い、各ユーザにとって最適な楽曲を精度よく提示できていると言える。

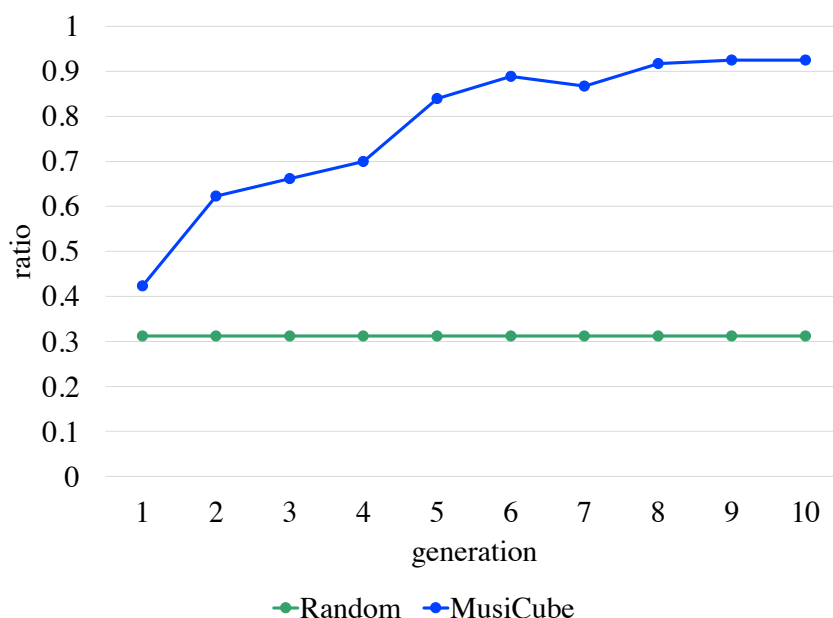


Figure 4.8: ランダム推薦と MusiCube による推薦による推薦精度の比較.

次に、ユーザが目的に合っていると思う楽曲が提示されるまでのステップ数を検証する。MusiCube では、1 世代を 10 曲とし、収束時間を速めるため、一度評価した楽曲が再び提示された場合、自動的に評価値を入力するようにしている。Figure 4.9 に、被験者 A (Figure 4.7(a)) の被験者のステップ数を示す。縦軸は、1 世代 (10 曲) 中の実際に評価した楽曲と前回の評価を使った楽曲の割合を、横軸は、世代を示している。この結果より、ユーザは 14 世代以降、40 曲程度評価した時点で、目的に合っている楽曲の割合が 9 割程度になっていることがわかる。

また、対話型遺伝的アルゴリズムにおける 1 世代の処理時間について検証する。「選択」から「マッチング」までの処理時間を計測したところ、10 回の平均が 31.6 ミリ秒であった。著者の主観では 1 世代の処理時間が 1 秒以内であれば対話的操作に十分であると考えていて、1 世代の個体数が少ないことで MusiCube の処理時間は対話的操作に十分なものとなっている。

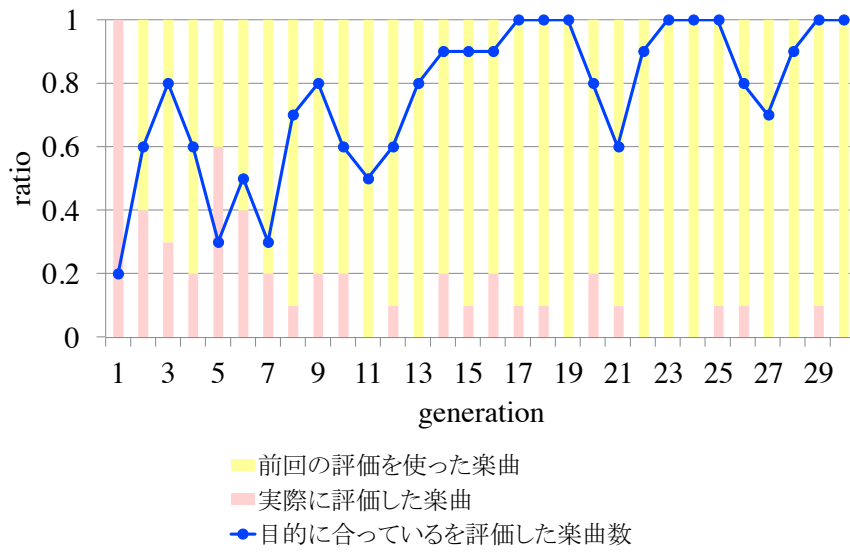


Figure 4.9: MusiCube の操作のステップ数.

#### 4.3.4 プレイリスト機能の有効性

MusiCube のプレイリスト機能の有効性について検証する. MusiCube では 3.4.3 節で説明したとおり, 立方体領域内で赤色のアイコン (目的に合っている楽曲) 群の近くに配置されている楽曲を選択しプレイリストに追加することができる. 被験者には, 評価のために用意した楽曲 500 曲の中からカフェに合う楽曲を 30 曲選ぶというテーマで, Table 4.1 に示すように, 方法 (1) 試聴とメタデータのみを手がかりに選曲, 方法 (2) 試聴, メタデータ, 学習結果の可視化を手がかりに選曲, 方法 (3) 試聴, メタデータ, 学習過程と学習結果の可視化を手がかりに選曲の 3 通りの方法で, プレイリストを作成してもらった.

Table 4.1: 選曲する時に用いる情報.

方法	(1)	(2)	(3)
試聴	○	○	○
メタデータ	○	○	○
学習過程の可視化			○
学習結果の可視化		○	○

まず, プレイリスト作成にかかる時間について検証する. 方法 (1) ~ (3) で, プレイリストの作成にかかった時間を Figure 4.10 に示す. なお, グラフの縦軸は時間を表している. 方法 (1) は, MusiCube を利用した方法 (2) と (3) と比べて, プレイリストの作成にとっても多くの時間がかかっている. また, 方法

(2) と方法 (3) を比較すると、学習過程の可視化を行うことで、プレイリストの作成にかかる時間を短縮できることが読み取れる。

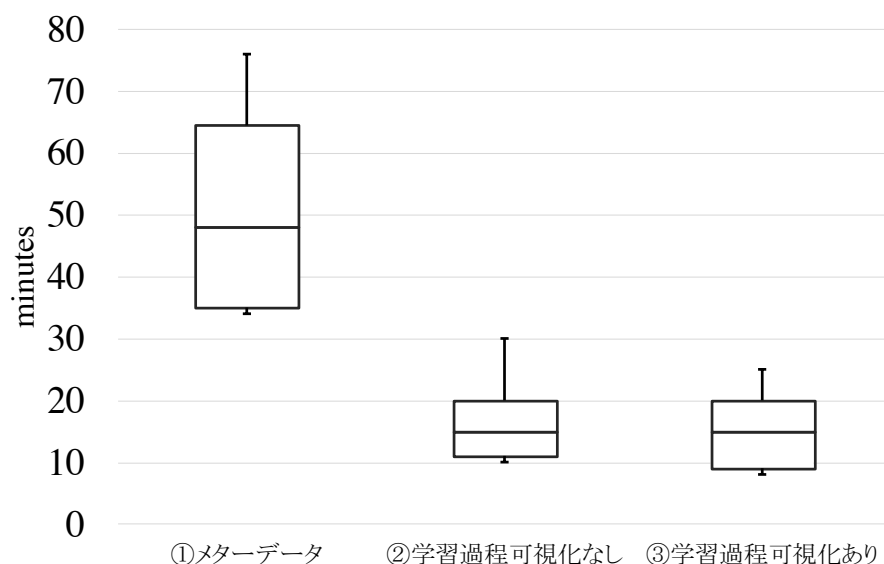


Figure 4.10: 被験者 10 人 (A-J) のプレイリストの作成にかかる時間.

次に、MusiCube のプレイリスト機能を用いて作成したプレイリストの内容について検証する。被験者には、方法 (2) と方法 (3) で作成したプレイリストの楽曲を聴いてもらい、プレイリストに含まれる全曲のうち目的に合う楽曲がどのくらいあるか答えてもらった。その結果を Figure 4.11 に示す。なお、グラフの縦軸はプレイリスト中の目的に合っている楽曲の割合を表している。この結果より、学習過程の可視化を行うことで、プレイリストの満足度が上がっていることが読み取れる。

ユーザテスト後に学習過程の可視化結果を見ずに学習を進める方法 (2) と学習過程の可視化結果を見ながら学習を進める方法 (3)、どちらがよいかとヒアリングしたところ、カフェに合う楽曲を選ぶ時のように、選曲に一貫性があることが重要な時、学習過程の可視化はそのガイドラインとして非常に有益だと感じたという意見が多かった。一方、可視化結果が選曲における先入観になってしまうことを実験前に懸念したが、その点はあまり影響がないという意見が多かった。

音楽に対するユーザの嗜好は、特徴量空間において多峰性を示すことがあるが、対話型進化計算がその複数の最適解領域を同時に探索しているのかはユーザにとってわかりにくい。また、突然変異によって、ユーザの嗜好に全く合わない楽曲を提示されることもあるが、それもなぜ提示されたかユーザにとってわかりにくい。MusiCube による可視化結果は、各楽曲の特徴を理解するのに有効であり、可視化結

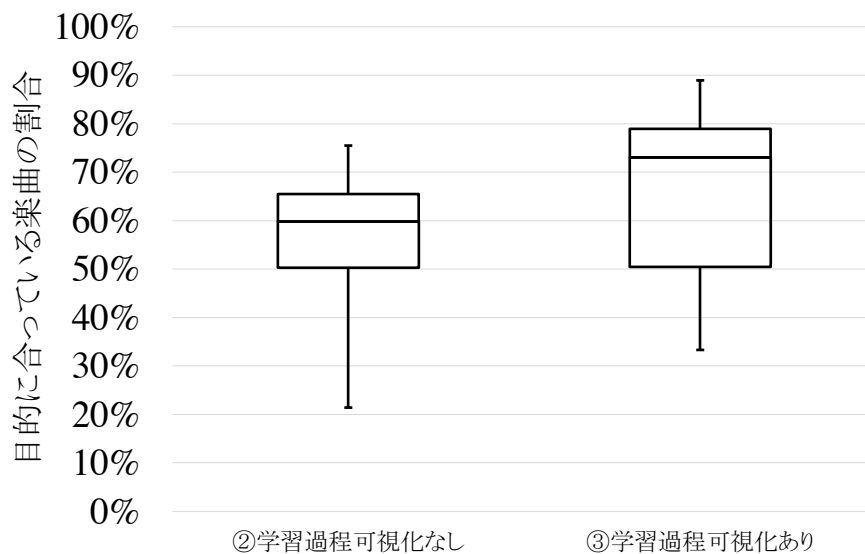


Figure 4.11: 被験者 10 人 (A-J) のプレイリストで目的に合っている楽曲の割合.

果を眺めながらのプレイリストを作成することで、ユーザ自身では気づきにくい嗜好の傾向を観察しながら一貫性のある選曲をすることが可能になったと考えられる.

続いて, Figure 4.12 にて, 10 人の被験者によって作成されたプレイリストにおいて, ジャンル別の楽曲の割合を 100% 積み上げ棒グラフで示す. なおグラフの横軸は被験者を表し, 縦軸はジャンル別の割合を表している.

この結果から, 各プレイリストにはさまざまなジャンルの楽曲が含まれており, MusiCube がジャンルという枠を超えてユーザの嗜好に合う楽曲を選ぶのに有効であることがわかった.

#### 4.3.5 GUI の使いやすさ

最後に, 被験者に「その時の目的に合う楽曲をユーザの嗜好を反映させて選曲する場合, MusiCube の GUI は効果的であるか」というアンケートを実施したところ, 5 段階評価で最大値 5.0, 平均値 4.0, 最小値 3.0 という結果となった.

続いて, アンケートのフリーコメントを紹介する.

- 楽曲特徴量に加えて, ジャンルや歌詞などのメタデータを利用すると, さらに便利になるかもしれない.
- プレイリストを聴きながら, 目的に合っていない楽曲だけを削除する, または GUI 画面 (左側) に



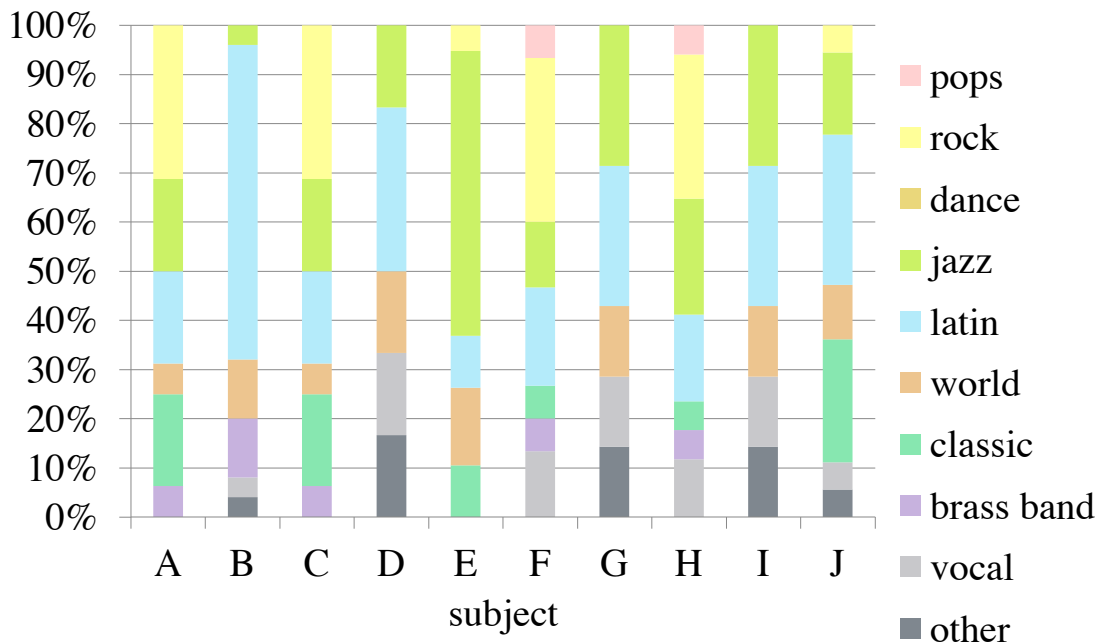


Figure 4.12: 被験者 10 人 (A-J) の結果におけるプレイリストの楽曲ジャンルの割合.

表示されている立方体領域内でプレイリストに追加されなかった楽曲を聴き、目的に合っていたらプレイリストに追加するなど手動で微調整を行う機能があれば、さらに直感的な選曲が可能になるかもしれない。

- 可視化結果がばらついていた場合、自分の定めていた「カフェの BGM」のイメージが曖昧だと認識できる。実際に BGM を選ぶ際、例えばもっとジャズっぽい雰囲気などと統一感のある雰囲気にできると思う。
- 可視化結果を見ながら楽曲評価を行うと、楽曲の感じ方が変わり、評価に影響する可能性がある。

著者は、これらのコメントを参考に MusiCube をより使いやすい楽曲提示インタフェースとして改良していきたいと考えている。

## 4.4 考察と課題

本手法を用いた評価実験から、「カフェに合う曲」というような特定の目的に相応しい楽曲を選ぶ場合において、ユーザによって嗜好が異なることを示し、ユーザごとに嗜好の傾向がどのような楽曲特徴に

起因しているかを示した。また、多次元からなる楽曲特徴量を次元削減することなく、ユーザが任意の2次元の楽曲特徴量を対話的に選ぶことで、ユーザの嗜好がどのような楽曲特徴に起因しているかを提示することができ、ユーザ自身では気づきにくい嗜好の傾向を観察しながら一貫性のある選曲をすることが可能になったことを示した。さらに、対話型進化計算を用いることで、1曲ずつ聴いていくうちに徐々に推薦結果の満足度が上がっていくことや嗜好の多峰性に対応できていることを示した。この結果より、特徴量空間における対話型遺伝的アルゴリズムの学習過程や学習結果を可視化することにより、楽曲の特徴を確認しながら、ユーザの嗜好に合わせて一貫性のある選曲が可能になったと考えている。また、多次元からなる特徴量を次元削減せず、重要な2次元の特徴量をシステムが自動的に選び可視化することにより、嗜好の傾向を読み取ることができるようになったと考えている。

今後の課題として、以下の4点を挙げる。まず1点目として、推薦精度を向上させるために楽曲特徴量をどのように採取するかについて引き続き検討していきたい。現時点では、楽曲における0分30秒から0分40秒の10秒間から得た楽曲特徴量をその楽曲の楽曲特徴量としている。今後は、楽曲全体を10秒ごとに分割しそれぞれから楽曲特徴量を抽出し、サビの部分検出 [Goto02] などを適用することで最も適した楽曲特徴量をその楽曲の特徴量として扱いたい。また、被験者によっては可視化結果において赤色のアイコン（目的に合っている楽曲）群と青色のアイコン（目的に合っていない楽曲）群が混在してしまう結果となった。このような可視化結果となった原因について検討したところ、邦楽などの特殊な音階からなる楽曲の扱い方に問題があったと推測された。著者は3.4.2節で述べた楽曲特徴量に加え、非常に強い印象を与える特別なジャンルや歌詞などのメタデータを補足的に利用することで、MusiCube本来の特徴を活かしつつ、より精度の高い推薦結果を提示できると期待している。

2点目の課題として、自身の嗜好情報と他者の嗜好情報を同時に表示できるような機能を追加し、さらに効率の良い楽曲提示を実現したい。

3点目の課題として、3.4.4節で述べた対話型進化計算を用いた楽曲提示では、抽出した楽曲特徴量に対して主成分分析を行い、上位成分の値を遺伝子として扱っている。このアルゴリズムにおいてユーザの嗜好の傾向に起因しやすい楽曲特徴量に重みをつけることで、より最適な収束結果を導けるかを試したい。また、3.4.4節で述べた対話型進化計算を用いた評価では、収束時間を速めるため、一度評価した個体が再提示された場合、自動的に評価値を入力している。しかし、一度評価した楽曲を常に再提示したり、ある一定の割合で再提示することで、複数回の評価によって、被験者の評価を精密化したり、嗜好の変化に対応できる可能性がある。しかし、これらは、長期間に渡る実験が必要であり、実証に至っていない。

以上について検討した上で、膨大な楽曲データを用いた実験を実施し、MusiCubeの有用性やスケー

ラビリティを再検証したい。

## 4.5 まとめ

ユーザの操作負担を最小限に抑えつつ、ユーザの嗜好に合った音楽コンテンツを推薦するためのインタフェース”MusiCube”を提案し、その有効性を検証した。MusiCube は、ユーザがアクセス可能な全てのコンテンツの中から、嗜好に合った楽曲を精度よく推薦するために、楽曲を特徴量で表現する。その特徴量空間において、ユーザの嗜好に合うコンテンツの位置を対話型進化計算で探索する。続いて、対話型進化計算の学習過程や学習結果を可視化することで、楽曲の特徴を確認しながら、ユーザの嗜好に合わせて一貫性のある選曲を可能にした。また、多次元からなる特徴量を次元削減せず、重要な 2 次元の特徴量をシステムが自動的に選び可視化することにより、嗜好の傾向を読み取ることを可能にした。

## 4.6 むすび

本章では、MusiCube の実行結果について述べた。まず、4.2 節ではユーザテストの方法を述べ、4.3 節では実験結果を説明し、4.4 節では実験結果から導かれる考察と課題を述べた。最後に、4.5 節で MusiCube の総括した。

## 第5章 CrowdRetouchの処理手順

### 5.1 はじめに

本章では、ユーザの嗜好に合わせて写真を加工するためのユーザインタフェース（CrowdRetouch）の処理手順について述べる。

### 5.2 背景

デジタルカメラやスマートフォンの普及、および写真を使ったコミュニケーションを可能にするソーシャル・ネットワーキング・サービス (SNS) などの拡大により、日常生活の中で撮影した写真をネットワーク上で共有することが容易になった。

自分で撮影した画像を SNS 上で公開する際に、各種アプリケーションに搭載されている自動補正機能、フィルター機能、トーンカーブなどの色補正機能を使って、画像の明るさや色合いなどを補正してからアップロードすることもしばしばある。自動補正機能には、多くのユーザが適切だと感じるように自動的に補正できるという利点がある一方で、各ユーザの嗜好や各シーンの特性を反映することが難しいという問題がある。フィルター機能には、アプリケーションによって提供されている効果が簡単にかえられるという利点がある一方で、ユーザの好みが異なるため意図通りの補正にならなかったり、フィルターの種類が多すぎてどれを選んで良いかわからないという問題がある。さらに、トーンカーブなどの色補正には、明るさや色合いなどのパラメータを手動で調整することで、各ユーザの嗜好や各シーンの特性を反映できるという利点がある一方で、画像補正パラメータの調整に不慣れなユーザにとっては、どのパラメータを調整すれば適切に補正できるかわからないことが多い。また、日々の記録として撮影する料理写真のように、枚数の多い写真群に対して一括補正する場合にも、1枚ずつ手動操作で画像補正する手間と時間を割かなくてはならない。

本章では、同じようなシーンで撮影された写真群に対して、各ユーザの画像補正の傾向を反映した画像一括補正を可能とするシステム”CrowdRetouch”を提案する。CrowdRetouch では、まずサンプル学習画像をカテゴリ分類し、画像特徴量によって画像クラスタを生成する。次にある程度の人数の初期ユーザに対して、画像カテゴリ、画像クラスタごとに一定数のサンプル学習画像を手動で画像補正させ、その特性によって初期ユーザをクラスタリングする。続いて各ユーザクラスタに対して画像補正パラメー

タの回帰分析を適用し、その結果を画像一括補正に用いる。一方、新規ユーザは、各ユーザクラスタの回帰式から算出される加工パラメータによって生成された補正結果画像をブレンドし、自身が所属するユーザクラスタを特定する。ここで、各補正結果画像のブレンド率を Parallel Coordinate Plot で可視化することで、ユーザに好みの補正傾向を気づかせることができる。新規ユーザが所属するユーザクラスタが特定できたら、各ユーザクラスタの回帰式と、ブレンド率の学習結果を用いて、他の画像を一括補正する。このように、初期ユーザはユーザクラスタを生成するために、サンプル学習画像に対する手動補正に時間がかかるが、新規ユーザは少数のサンプル学習画像に対する補正結果をブレンドしながら所属するユーザクラスタを探索し特定することで、少ない操作によってその後の画像一括補正処理を利用できる。

CrowdRetouch を利用することにより、画像パラメータの調整に不慣れなユーザであっても、多数の写真群に対して手間と時間を割かずに、各ユーザの嗜好を反映した画像一括補正ができると期待される。著者は、大規模画像セット (ImageNet[ImageNet]) で収集した写真と、著者がデジタルカメラのオート機能で撮影した写真を適用事例として、CrowdRetouch の効果を検証した。

## 5.3 処理手順

本章ではユーザの嗜好を反映した画像一括補正を可能とするためのシステム”CrowdRetouch”を提案する。CrowdRetouch は、同じようなシーンで撮影された写真群を前提として、少数のサンプル学習画像に対する補正操作結果をもとに、ユーザの補正特性傾向を反映した画像一括補正を可能とするシステムである。毎日の記録で撮影する料理写真をまとめて補正する、数十枚ある同じような風景の写真をまとめて補正する、といったように 1 人のユーザが同じような被写体の画像を補正するユースケースを想定している。CrowdRetouch の処理手順は以下の通りである。処理手順は「システムの訓練」と「タスクの実行」の 2 段階で構成される。

### 1. システムの訓練

- (a) サンプル学習画像を画像カテゴリ、画像クラスタに分類する。(5.3.1 節参照)。
- (b) 初期ユーザに少数のサンプル学習画像を色補正させる。(5.3.2 節参照)。
- (c) 1b の結果を用いて、補正の好みによって初期ユーザをクラスタリングする。(5.3.3 節参照)。
- (d) 補正パラメータの回帰式を画像クラスタ、ユーザクラスタごとに算出する。(5.3.4 節参照)。
- (e) 新しい画像がどの画像クラスタに属するかを推定するための処理に対して学習する。(5.3.5 節参照)。

## 2. タスクの実行

- (a) 新規ユーザがどのユーザクラスに所属するかを判定する. (5.3.6 節参照).
- (b) 新しい画像がどの画像クラスに所属するかを判定する. (5.3.6 節参照).
- (c) 所属されたクラスの重回帰分析の回帰式を適用し, 画像を自動補正する. (5.3.6 節参照).

### 5.3.1 画像カテゴリと画像クラスタの形成

CrowdRetouch は「システムの訓練」の準備段階として, まずサンプル学習画像をカテゴリ分類する. CrowdRetouch では, 与えられたサンプル学習画像には対象物を表す概略的なキーワード (例えば「風景」「飲食物」など) が付与されていることを想定する. このようなキーワードが付いていない画像には "other" というキーワードを付与する.

以上の処理によってカテゴリ分類されたサンプル学習画像群に対して前処理として, 画像特徴量によってクラスタリングを適用する. 具体的には, 1 枚のサンプル学習画像に対する画像特徴量をベクトルとして扱い, 画像特徴量ベクトルを主成分分析し, K-means 法を適用することで, 画像クラスタを生成する. 現在の実装では画像特徴量として, R, G, B, RGB(輝度) の各々について平均, 標準偏差, 中間値, 最頻値を算出した合計 16 変数, また画像を分割した各領域の平均輝度値の最小値と最大値の 2 変数, 合計で 18 変数を用いている.

### 5.3.2 サンプル学習画像の色補正

5.3.1 節で示した画像分類の後に, 初期ユーザにサンプル学習画像を提示し, 対話操作による画像補正を課する. CrowdRetouch のシステムの訓練のための GUI 画面を Figure 5.1 に示す. 画面 (左側) に操作パネルを表示し, 画面 (右側) にサンプル学習画像を表示する.

現在の実装では, 画像補正パラメータとしてトーンカーブを用いている. 処理対象のチャンネルは, RGB(輝度), R, G, B の 4 種類である. また, 処理を簡易化するために, 各チャンネルのコントロールポイントをシャドウ側の 1 点 ( $C_s = 64$ ), 中間調の 1 点 ( $C_m = 128$ ), ハイライト側の 1 点 ( $C_h = 192$ ) の 3 点に制限している (Figure 5.2 参照). 結果として現在の実装では, 1 枚のサンプル学習画像に対して初期ユーザは 12 種類の画像補正パラメータを操作することになる.

### 5.3.3 ユーザクラスタの生成

画像カテゴリごとに, ユーザクラスタを形成する. 5.3.2 節で示した対話操作による画像補正結果より, 同一画像カテゴリに属するサンプル学習画像の枚数を  $n_S$ , 画像補正パラメータの個数を  $n_P$  (現状の実装

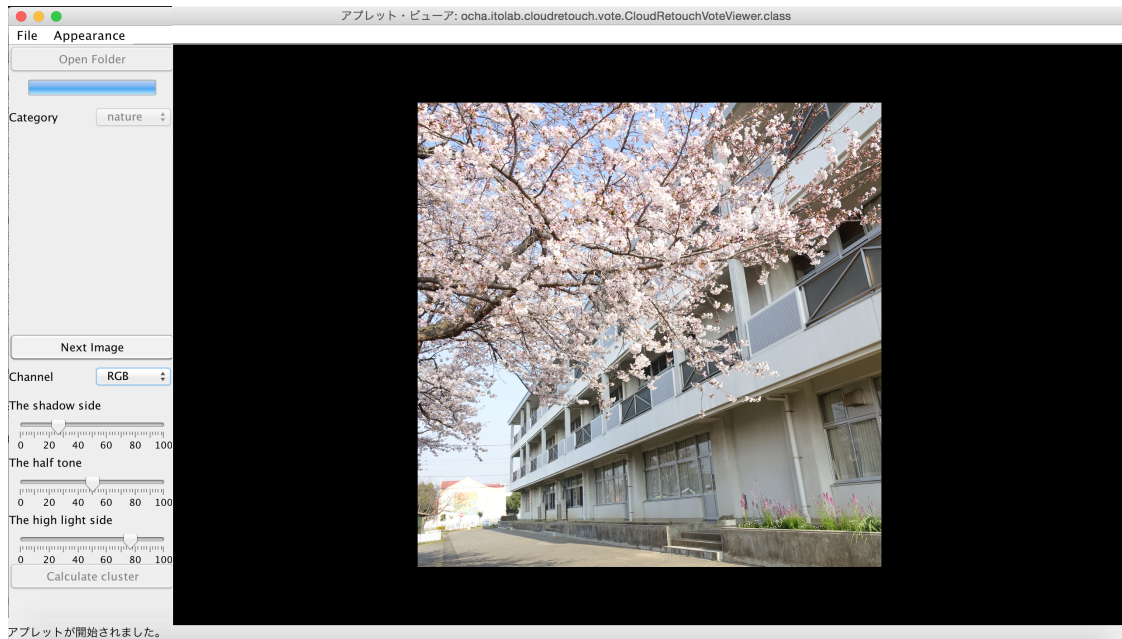


Figure 5.1: CrowdRetouch の初期ユーザのための GUI 画面.

では  $n_P = 12$ ) とすると, 1 人の初期ユーザの画像補正結果を  $(n_S n_P)$  次元ベクトルとして扱うことができる. このベクトルに主成分分析を適用し, 上位の主成分 (現在の実装では第 3 主成分まで) を採用することで初期ユーザの嗜好を表すベクトルとする. このベクトルに対してデンドログラム (ウォード法) を適用し, ユーザクラスタを形成する. 以上の処理によって構成されるデータ構造を Figure 5.3 に示す. ここで, ユーザクラスタは画像クラスタごとに作られる.

### 5.3.4 好みの計算モデルの算出

CrowdRetouch では, 各画像カテゴリにおけるユーザクラスタごとに重回帰分析を適用し, その結果を新しい画像の画像補正パラメータの推定に用いる. 具体的には, 画像特徴量  $X = (x_1, x_2, \dots, x_{n_F})$  を説明変数として,  $i$  番目の画像補正パラメータ  $r_i$  を目的変数としたときに,  $r_i = f_i(X)$  となる  $f_i$  を線形重回帰分析によって求める. 現在の実装では, 画像特徴量として, R, G, B, RGB(輝度) の各々について平均, 標準偏差, 中間値, 最頻値を算出した合計 16 変数, また画像を分割した各領域の平均輝度値の最小値と最大値の 2 変数, 合計で 18 変数を説明変数としている. 重回帰分析で陥りがちな失敗例の一つに, 説明変数が高次元すぎるゆえに却って良質な回帰結果が得られない事例が考えられる. この問題を回避するために現在の実装では, 目的変数 (画像補正パラメータ) と説明変数 (画像特徴量) の Pearson 相

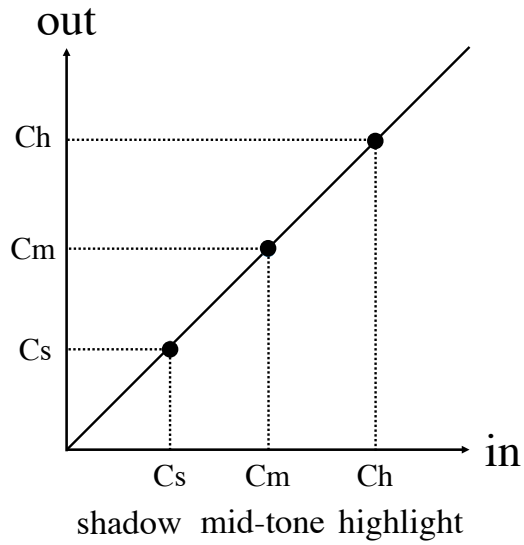


Figure 5.2: トーンカーブのコントロールポイント.

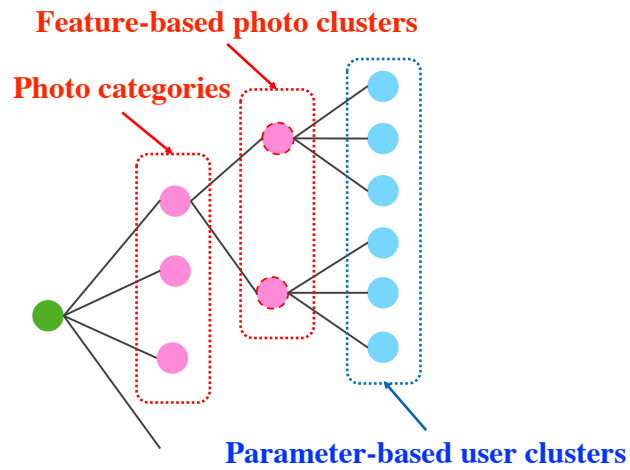


Figure 5.3: 画像クラスとユーザクラス.

関係数を算出し、相関の高い説明変数のみを採用する．また、説明変数どうして相関が高すぎるものが混在することで、回帰結果の信頼性が低下する場合もある．そのため、説明変数どうして相関の高すぎるものがある場合には、1つの説明変数を残して他の説明変数を除外するようにしている．



現在の実装では、目的変数に対して Pearson 相関係数が 0.5 以上の説明変数を採用している。さらに採用された説明変数のペアで Pearson 相関係数を算出し、その絶対値が 0.9 以上の場合どちらかを除外するようにしている。Figure 5.4 に以上の処理を示す。

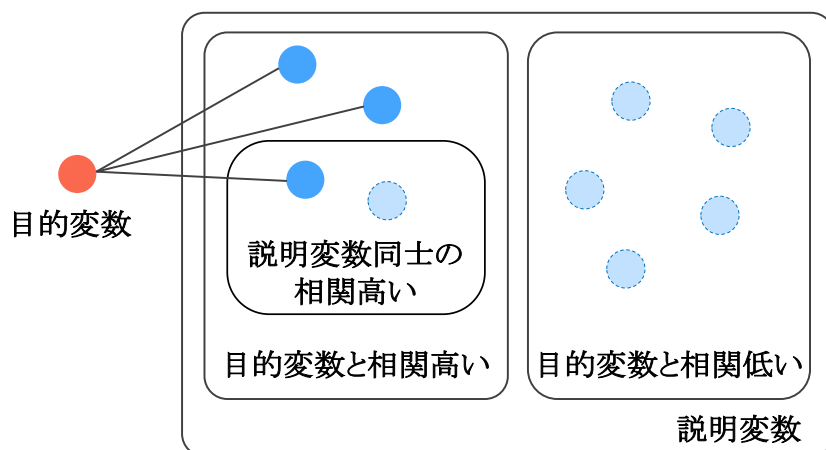


Figure 5.4: 説明変数と目的変数の関係.

### 5.3.5 新しい画像のクラスタ判別のための学習

CrowdRetouch では、新しい画像が与えられると、その画像がどの画像クラスタに分類されるかを判別する。まず、対象物を表すキーワードによって判別し、さらに、そのキーワードごとにサポートベクターマシンを適用している。そこで、「システムの訓練」の一環としてサポートベクターマシンに教師信号を入力する。具体的には、対象物を表すキーワードによって分類されたサンプル学習画像群に関する多次元変数（画像特徴量  $X$ ）およびラベル（所属クラスタ番号）を教師信号として入力する。現在の実装ではサポートベクターマシンによる分類処理に LIBSVM[LIVSVM] を使用している。カーネル関数には RBF カーネルを採用し、 $\gamma$  値と  $C$  値には LIBSVM のデフォルト値を用いている。なお、この処理は多値分類の一種になるが、現在の実装では 1 対 1 分類器を用いて多値分類を実現している。

### 5.3.6 新しい画像、新規ユーザに対する画像補正

前節までに示した方法で「システムの訓練」を完了したことを前提として、新規ユーザが新しい画像を補正するために CrowdRetouch は以下の処理を適用する。まず、与えられた新しい画像がどの画像ク

クラスに属するか、また新規ユーザがどのユーザクラスに属するかを判定する。続いて当該ユーザクラスにおける回帰式を適用することで画像補正パラメータを推定し、その結果に従って画像を補正する。

この処理ではまず新しい画像の所属カテゴリを推定する。対象物を表すキーワードが付与されている画像については、そのキーワードによって所属カテゴリを推定する。キーワードが付与されていない画像については、"other"というキーワードを付与する。続いて、当該画像の画像特徴量  $X$  をサポートベクターマシンに入力することで、最もふさわしい画像クラスを推定する。

そして、新規ユーザがどのユーザクラスに属するかを判別する。新規ユーザにサンプル学習画像を提示し、対話操作による画像補正を課す。CrowdRetouch のタスク実行のための GUI 画面を Figure 5.5 に示す。画面 (左側) にサンプル学習画像に対する補正結果画像を表示し、画面 (中央) に各ユーザクラスの回帰式から算出された画像補正パラメータによって生成された補正結果画像 (以下、フィルターとする) と Parallel Coordinate Plot による補正結果画像のブレンド率の可視化結果を表示し、画面 (右側) に操作パネルを表示する。

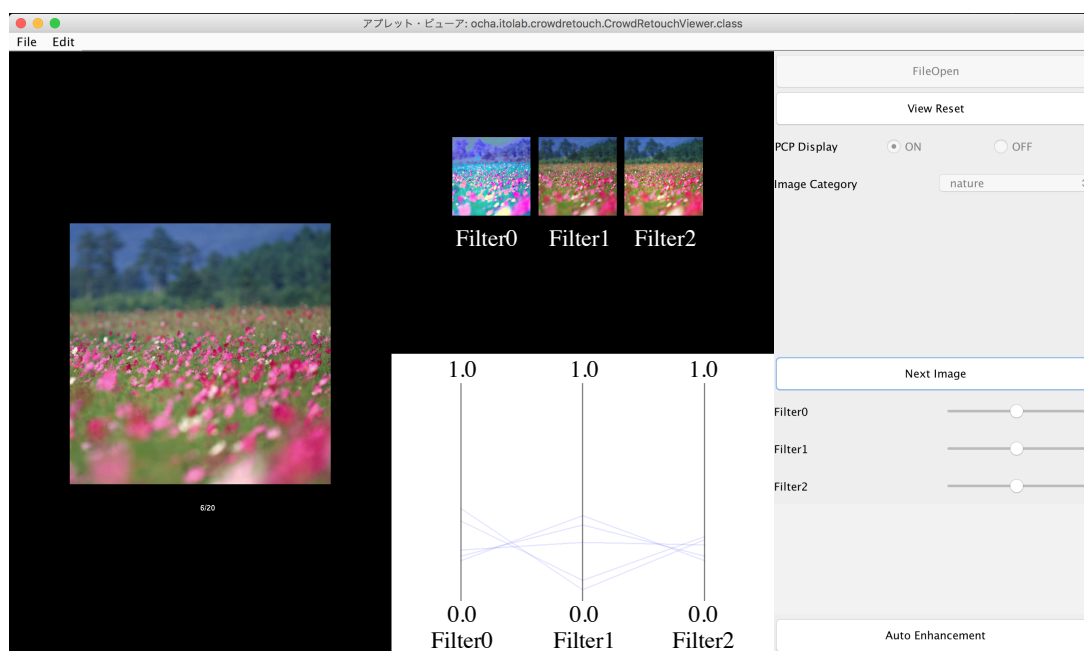


Figure 5.5: CrowdRetouch の新規ユーザのための GUI 画面。

新規ユーザは、サンプル学習画像に対して、フィルターをブレンドしながら画像補正する。この過程において、新規ユーザに画像補正を課するサンプル学習画像の数を減らすため、以下に説明する Davies-Bouldin Index (DB Index) [Davies79] を適用することでユーザクラスタリングに強く貢献している画像を選別する。

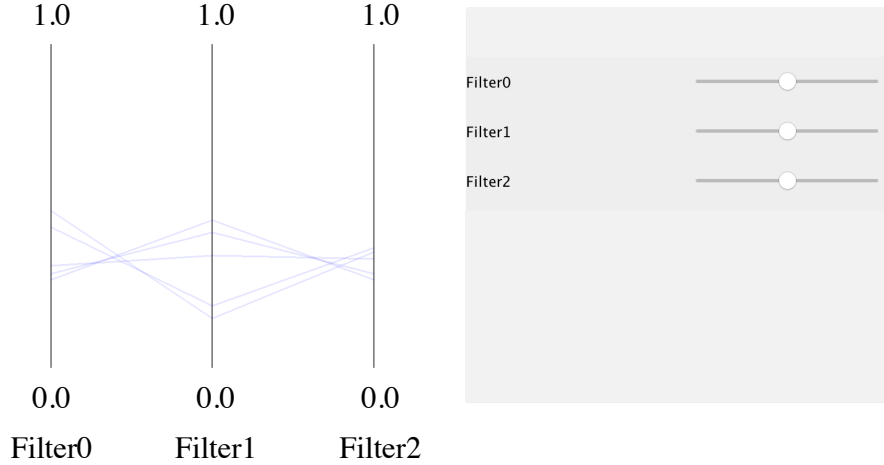


Figure 5.6: Parallel Coordinate Plot による補正結果画像のブレンド率の可視化表示.

具体的には、 $i$  番目のユーザが 1 枚の画像を加工した時、画像補正パラメータが  $n_P$  個あるとすると画像補正パラメータ値の集合は  $n_P$  次元ベクトルとなる．これを  $a_i$  とする．同一ユーザクラスを構成する  $i$  番目のユーザと  $j$  番目のユーザとの間で画像補正パラメータがどれくらい似ているか評価するには、 $a_i$  と  $a_j$  の距離を求める．このユーザクラスを構成する任意の 2 名について距離の平均  $ave|a_i - a_j|$  を求めることで、このユーザクラスがどのくらい密であるかを計算できる．平均が小さければ小さいほど、ユーザクラスが小さいことを意味し、つまりユーザクラスを構成するユーザ群が同様の画像補正パラメータを選ぶ傾向にあることがわかる．

一方で、 $k$  番目のユーザクラスタの中心における画像補正パラメータも  $n_P$  次元ベクトルとなる．これを  $c_k$  とする． $k$  番目のユーザクラスタと  $l$  番目のユーザクラスタの距離の平均  $ave|c_k - c_l|$  を求めることで、ユーザクラスタ間の距離の平均を求めることができる．この平均が大きければ大きいほど、ユーザクラスタごとに画像補正パラメータが明確に異なっていることを意味する．

一般的には  $ave|a_i - a_j|/ave|c_k - c_l|$  の値が小さいほど、そのユーザクラスタは良好であると判断できる．そこで現在の実装では、全てのサンプル学習画像についてこの値を計算し、この値が 2.0 以下であるサンプル学習画像を新規ユーザに提示する．

新規ユーザがサンプル学習画像を補正する際、CrowdRetouch は、Figure 5.6 に示すように、Parallel Coordinate Plot で 1 枚のサンプル学習画像に対するブレンド率を 1 本の折れ線として表示する．これにより、新規ユーザはブレンド率の探索過程の可視化を見ながら、画像補正を行うことで、自身の好みのフィルターの傾向を把握することができる．以上の学習結果により、各フィルターの割合の平均を求め、

その平均が最も高いフィルターを新規ユーザの該当クラスタ (フィルター) とする。最後に、その該当クラスタの回帰式を用いて、画像一括補正を行う。このとき、仮に同一ユーザが新しい写真を大量に用意したとしても、同一カテゴリに属する全ての写真に対して同じ回帰式を適用できるため、ユーザの嗜好を反映させた一括画像補正ができると期待される。

## 5.4 むすび

本章では、CrowdRetouch の処理手順について説明した。まず、5.2 節では CrowdRetouch を提案するに至った背景について、5.3 節では CrowdRetouch の処理手順について説明した。

## 第6章 CrowdRetouchの実行結果

### 6.1 はじめに

本章では，CrowdRetouchの実行結果および評価実験に関する考察を述べる．

### 6.2 ユーザテスト

本実験では，大学で情報科学を専攻している女子学生 26 名を対象に，あらかじめ，初期ユーザ 16 名と新規ユーザ 10 名にグループ分けし，実験を行った．

#### 6.2.1 実行環境

Java JDK 1.7.0 を用いて CrowdRetouch を実装し，MacBook Pro の上で実行した．システム評価にあたり，大規模画像セット ImageNet[ImageNet] からランダムに収集した写真と，著者がデジタルカメラのオート機能で撮影した写真を用意した．また写真共有サイト PHOTOHITO[PHOTOHITO] でのカテゴリ分類の粒度を参考に，画像カテゴリを「food」，「landscape」，「nature」とし，写真の付与されているキーワード情報から写真群を分類した．

#### 6.2.2 初期ユーザによるユーザクラスタ生成のための評価実験

まず，ユーザクラスタの生成に関して，初期ユーザ 16 名に対して以下の内容で実験を行った．

1. 画像カテゴリごとに用意したサンプル学習画像 30 枚に対して，CrowdRetouch のシステムの訓練のための GUI(Figure 5.1) を使って，トーンカーブで画像を補正してもらう．
2. 補正結果を比較するため，下記の (A)～(G) の画像を各被験者に提示し，嗜好に合った補正結果になっているかを評価してもらう．

補正結果を比較するための画像として，画像カテゴリごとに用意した新しい画像 10 枚に対して，下記の画像を用意した．

(A) オリジナル画像．

- (B) Picasa の自動補正機能を使って補正した画像.
- (C) サンプル学習画像群を同一カテゴリ, 同一画像クラスタとし, 被験者を分類せず全員分のデータから推定した補正パラメータを適用した画像.
- (D) サンプル学習画像群を画像カテゴリに分類し, 同一画像クラスタとし, 被験者を分類せず全員分のデータから推定した補正パラメータを適用した画像.
- (E) サンプル学習画像群を同一カテゴリとし, 画像特徴量によって画像クラスタに分類し, 被験者を分類せず全員分のデータから推定した補正パラメータを適用した画像.
- (F) サンプル学習画像群を画像カテゴリに分類し, 同一画像クラスタとし, 被験者をユーザクラスタに分類し, ユーザクラスタごとのデータから推定した補正パラメータを適用した画像.
- (G) サンプル学習画像群を画像カテゴリに分類し, 画像特徴量によって画像クラスタに分類し, 被験者をユーザクラスタに分類し, ユーザクラスタごとのデータから推定した補正パラメータを適用した画像.

### 6.2.3 新規ユーザによるサンプル学習画像の補正方法のための評価実験

次に, サンプル学習画像の補正方法に関して, 新規ユーザ 5 名に対して以下の内容で実験を行った.

1. 画像カテゴリごとに用意したサンプル学習画像 20 枚に対して, CrowdRetouch のタスク実行のための GUI(Figure 5.5) を使って, フィルターの候補をブレンドして画像を補正してもらう.
2. 画像カテゴリごとに用意したサンプル学習画像 20 枚に対して, CrowdRetouch のシステムの訓練のための GUI(Figure 5.1) を使って, トーンカーブで画像を補正してもらう.
3. CrowdRetouch は, 1. と 2. の結果より, 下記の 3 通りの方法で補正パラメータを推定し, 新しい画像に対して画像補正を行う.
4. 3. の補正結果画像を各被験者に提示し, 嗜好に合った補正結果になっているかを評価してもらう.

補正方法を比較するため, 画像カテゴリごとに用意した新しい画像 5 枚に対して, 以下の 3 通りの方法で補正パラメータを推定し, それぞれ補正結果画像を生成した.

- 方法 1. フィルターの候補をブレンドした結果から各ユーザクラスタの割合の平均を求め, その平均が最も高いユーザクラスタを該当ユーザクラスタとする方法.

方法 2. フィルターの候補をブレンドした結果から各ユーザクラスタの割合の平均を求め、その値をもとに各ユーザクラスタの回帰式により算出される補正パラメータに重み付けする方法.

方法 3. トーンカーブを補正した結果から各ユーザクラスタの代表ユーザと新規ユーザの嗜好ベクトルの距離を求め、最小となるユーザクラスタを該当するユーザクラスタとする方法. 各ユーザクラスタの代表ユーザは、ユーザクラスタの重心に位置するユーザとする.

#### 6.2.4 新規ユーザによるクラスタ判別の評価実験

さらに、新規ユーザのクラスタ判別に関して、新規ユーザ 10 名に対して以下の内容で実験を行った。ブレンド率の可視化結果表示あり、表示なしで実施した。

1. 画像カテゴリごとに用意したサンプル学習画像 20 枚に対して、CrowdRetouch のタスク実行のための GUI(Figure 5.5) を使ってフィルターの候補をブレンドして画像を補正してもらう。
2. 1. の結果より、各ユーザクラスタの割合の平均を求め、その平均が最も高いユーザクラスタを該当するユーザクラスタとする。
3. 該当するユーザクラスタの回帰式から補正パラメータを推定し、新しい画像に対して一括画像補正を行う。
4. 3. の補正結果画像を各被験者に提示し、各被験者の嗜好に合った補正結果になっているかを評価してもらう。

ブレンド率の可視化結果表示あり、表示なしの効果を評価するため、それぞれで所属するユーザクラスタを判別するための学習を行い、補正結果画像を用意した。また、クラスタを判別するために必要なサンプル学習画像の枚数を検証するため、新規ユーザがサンプル学習画像 6 枚、9 枚、12 枚に対して補正が終わった時点で、該当するユーザクラスタを求め、それぞれ補正結果画像を生成した。

### 6.3 実験結果

#### 6.3.1 キーワードによる画像カテゴリの有効性

6.2.2 節の被験者に対して以下の実験を行った。

まず、画像群の中から、画像カテゴリを考慮せずにランダムに選んだ新しい画像 10 枚に対して、(A)、(C)、(D) の 3 種類を用意した。被験者に、その中からユーザの嗜好に最も合致する画像を選んでもらい、回答結果を集計した。結果を Figure 6.1 に示す。オリジナル画像である (A) と CrowdRetouch で推定し

たパラメータを適用した (C), (D) を比較すると, CrowdRetouch で指定したパラメータを適用した (C), (D) の方が, 高い満足度を得ることができた. さらに, CrowdRetouch で推定したパラメータを適用した (C), (D) を比較した. 結果として D が最も高い満足度を得る結果になった. この結果より, 画像カテゴリごとにユーザの画像補正傾向が違うことが示唆された.

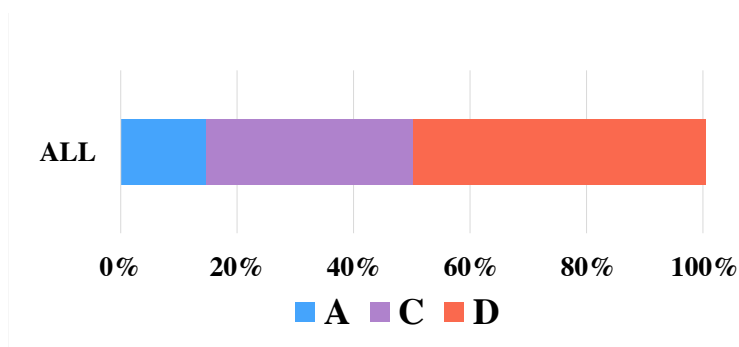


Figure 6.1: 画像カテゴリによる分類の効果.

次に, 画像群の中から, 画像カテゴリを考慮せずにランダムに選んだ新しい画像 10 枚に対して, (A), (D), (E) の 3 種類を用意した. 被験者に, その中からユーザの嗜好に最も合致する画像を選んでもらい, 回答結果を集計した. 結果を Figure 6.2 に示す. オリジナル画像である (A) と CrowdRetouch で推定したパラメータを適用した (D), (E) を比較すると, (D) が最も高い満足度を得る結果になった. この結果より, 画像をキーワードで分類する方が, 画像特徴量で分類するよりも満足度が高いことがわかった. このことから, CrowdRetouch において, キーワードによる画像分類が有効であることが示唆された.

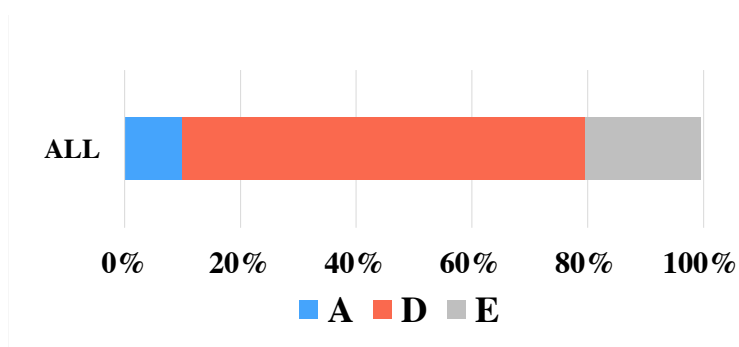


Figure 6.2: 画像カテゴリによる分類の効果と画像特徴量によるクラスタリングの効果の比較.

### 6.3.2 ユーザクラスタリングの有効性

6.2.2 節の被験者に対して以下の実験を行った.



まず、画像カテゴリごとに、新しい画像 10 枚に対して、(A), (D), (F) の 3 種類を用意した。被験者に、その中からユーザの嗜好に最も合致する画像を選んでもらい、回答結果を集計した。結果を Figure 6.3 に示す。オリジナル画像である (A) と CrowdRetouch で推定したパラメータを適用した (D), (F) を比較すると、CrowdRetouch で指定したパラメータを適用した (D), (F) の方が、高い満足度を得ることができた。さらに、CrowdRetouch で推定したパラメータを適用した (D) と (F) を比較した。結果として (F) が最も高い満足度を得る結果になった。この結果より、画像カテゴリごとにユーザをクラスタリングすることで、より高い満足度を得られることが示唆された。

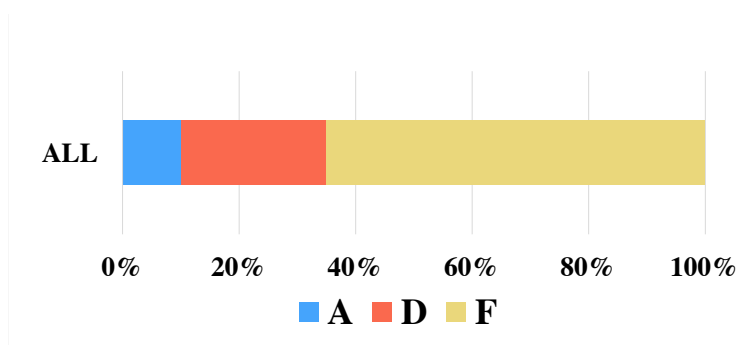


Figure 6.3: ユーザクラスタリングの効果.

次に、画像カテゴリごとに、新しい画像 10 枚に対して、(D) とユーザクラスタごとの (F) を用意し、その中からユーザの嗜好に最も合致する画像を選んでもらい、ユーザが属するクラスタの補正画像を選択する確率を求めた。カイ二乗検定を用いて検定した結果、 $\chi^2 = 5.556$ ,  $p = 0.018$  となり、 $p < 0.05$  で画像選択結果に有意差が認められた。

### 6.3.3 画像特徴量による画像クラスタリングの有効性

6.2.2 節の被験者に対して以下の実験を行った。

まず、画像カテゴリごとに、新しい画像 10 枚に対して、(A), (B), (F), (G) の 4 種類を用意し、その中からユーザの嗜好に最も合致する画像を選んでもらい、回答結果を集計した。「nature」という画像カテゴリにおける結果を Figure 6.4 に示す。

オリジナル画像である (A) と Picasa の自動補正機能を使って補正した (B) と CrowdRetouch で推定したパラメータを適用した (F), (G) を比較すると、どのユーザクラスタにおいても CrowdRetouch で推定したパラメータを適用した (F), (G) の方が高い満足度を得ることができた。さらに、CrowdRetouch で推定したパラメータを適用した (F), (G) を比較した。結果として (G) が最も高い満足度を得る結果に

### Photo category : nature

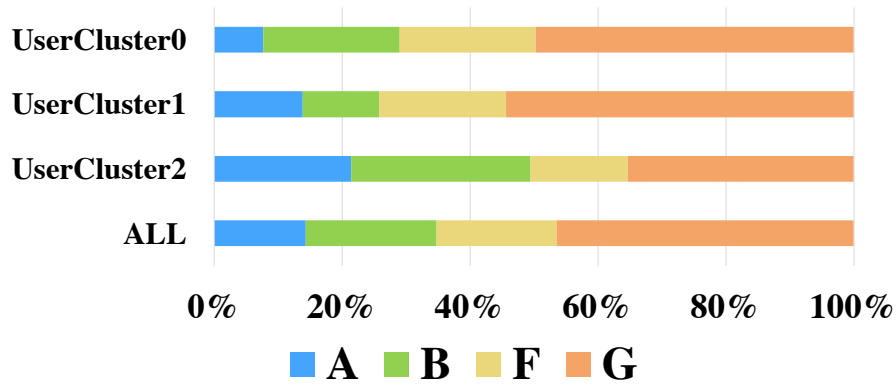


Figure 6.4: 画像特徴量による画像クラスタリングの効果.

なった. これは, 他の画像カテゴリにおいても同様の傾向が見られた.

(F) の結果が高い満足度を得た画像と (G) の結果が高い満足度を得た画像の傾向を分析する. どのユーザクラスタにおいても (F) の結果が高い満足度を得た画像については, 多様な色彩が現れている傾向があることがわかった. 画像の例を Figure 6.5 に示す. 一方, どのユーザクラスタにおいても (G) の方が高い満足度を得た画像については, 単一の色彩が大きな領域を占めることがわかった. 画像の例を Figure 6.6 に示す. つまり, 色彩が単一である画像の場合は特に, 画像特徴量による画像クラスタ分類が効果的であることが示唆される.



Figure 6.5: 画像特徴量による画像クラスタ分類なしで, 高い満足度を得た画像の例.

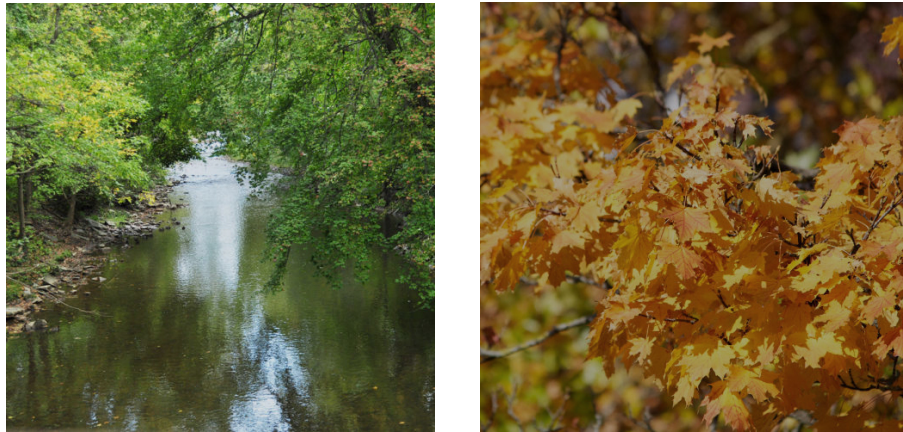


Figure 6.6: 画像特徴量による画像クラスタ分類ありで、高い満足度を得た画像の例.

#### 6.3.4 ユーザクラスタごとの嗜好の傾向

6.2.2 節の実験結果より、ユーザクラスタごとに嗜好の傾向がどのようにわかれたかを分析する．今回は、CrowdRetouch で補正した画像の中で満足度が高かった画像の例を Figure 6.7 に示す．ユーザクラスタ 0 の補正結果は、RGB(輝度) チャンネルのシャドウから中間調にかけて暗くし、トーンカーブの傾斜を強くすることで、コントラストのついた仕上がりになっている．ユーザクラスタ 1 の補正結果は、ユーザクラスタ 0 と同様に RGB(輝度) チャンネルのシャドウから中間調にかけて暗くし、トーンカーブの傾斜を強くすることで、コントラストのついた仕上がりになっている．さらに、G チャンネルのハイライト側を下げて、グリーンを弱く、マゼンタを強くすることで、桜の花びらがピンク色に補正され、春らしい温かみのある仕上がりになっている．ユーザクラスタ 2 の補正結果は、ユーザクラスタ 0 と同様に RGB(輝度) チャンネルのシャドウを少し暗くし、トーンカーブの傾斜を少し強くすることで、コントラストのついた仕上がりになっている．さらに、R チャンネルの中間調を上げて、レッドを強くすることで、全体的に赤色のイメージに近づいた仕上がりになっている．

この結果から、被験者によって補正の好みが変わることが示唆された．結果として、補正の傾向によってユーザをクラスタリングし、各ユーザクラスタに対して個別に補正パラメータを推定した上での一括画像処理に効果があることがわかった．

#### 6.3.5 初期ユーザのクラスタの精度

6.2.2 節の実験結果より、初期ユーザのサンプル学習画像の枚数を重ねるごとに、ユーザクラスタリングの精度が改善されるかを検証した．評価指標は、DBIndex の値とし、デンドログラムによるクラスタ

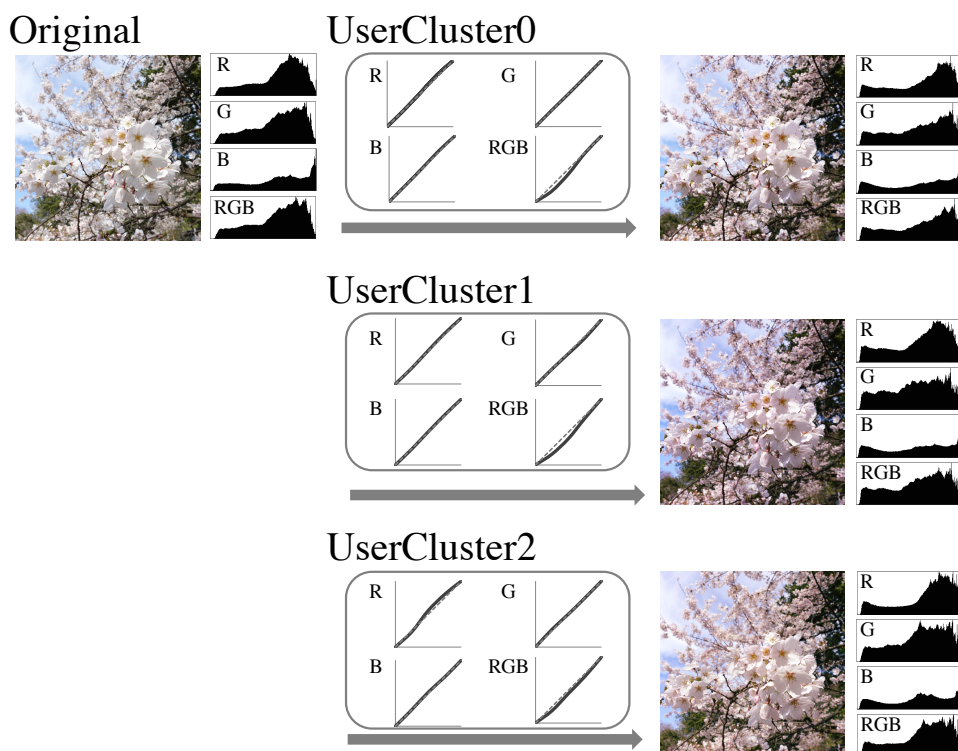


Figure 6.7: ユーザの嗜好の傾向の例.

リングのクラスタ数は DBIndex の値が最善となるように決めた. Figure6.8 に, 各画像カテゴリにおいて, サンプル学習画像の枚数を 10 枚, 20 枚, 30 枚と変えて, DBIndex の値を求めた結果と, デンドログラムの結果を示す. 全ての画像カテゴリにおいて, サンプル学習画像の枚数が増えると, ユーザクラスタのクラスタリングの精度が向上していることがわかる.

### 6.3.6 画像カテゴリごとの嗜好の傾向

6.2.2 節の実験結果より, 画像カテゴリごとの補正パラメータの違いを分析する. 本節では「food」, 「landscape」の画像カテゴリにおいて, サンプル学習画像 30 枚ずつ補正した時の画像補正パラメータの最小値, 平均値, 最大値を, それぞれ Figure6.9(a) と Figure6.9(b) に示す.

なおグラフの縦軸は画像補正パラメータの値を表し, 横軸はユーザクラスタごとのコントロールポイントを表す. 処理対象のチャンネルを RGB(輝度), R, G, B の 4 種類とし, 各チャンネルのコントロールポイントをシャドウ側の 1 点 ( $C_s = 64$ ), 中間調の 1 点 ( $C_m = 128$ ), ハイライト側の 1 点 ( $C_h = 192$ ) としている. Figure6.9(a), Figure6.9(b) の下部の文字列はチャンネルとコントロールポイントの組み合わせを示している. 例えば RGB(輝度) チャンネルのシャドウ側であれば, RGB64 と表している. またグラフの背景色について, 黄色は RGB(輝度) のみを変えたユーザクラスタ, 青色は RGB(輝度) と R, G, B の

いずれかを変えたユーザクラスタ、赤色はすべてのチャンネルを変えたユーザクラスタを示している。

Figure6.9(a) のユーザクラスタ 0 と Figure6.9(b) のユーザクラスタ 0 を比較すると、RGB（輝度）の補正量は類似しているものの、その他のパラメータの補正量は異なる。また、Figure6.9(a) のユーザクラスタ 1 と Figure6.9(b) のユーザクラスタ 2 を比較すると、RGB（輝度）のみ補正している点と同じだが、補正する方向が異なる。

この結果から、画像カテゴリによって補正の傾向が異なることが示唆された。結果として、キーワードによって画像を分類し、各画像カテゴリに対して個別に補正パラメータを推定した上での一括画像処理に効果があることがわかった。

### 6.3.7 新規ユーザによるサンプル学習画像の補正方法の比較

6.2.3 節の被験者に対して以下の実験を行った。6.2.3 節で説明した 3 通りの方法で補正パラメータを推定し、各画像カテゴリにつき評価画像 5 枚に対して、補正結果画像を用意した。被験者に、その中からユーザの嗜好に最も合致する画像を選んでもらい、回答結果を集計した。結果を Figure 6.10 に示す。方法 1 のサンプル学習画像をフィルターの候補をブレンドして補正し、該当クラスタを特定する方法が最も満足度が高い結果となり、カイ二乗検定を用いて検定した結果、 $\chi^2 = 7.0523$ ,  $p = 0.0079$  となり、 $p < 0.05$  で方法 1 の結果に有意差が認められた。

### 6.3.8 ブレンド率の可視化結果表示の効果

6.2.4 節の被験者に対して以下の実験を行った。ブレンド率の可視化結果表示あり、表示なしで、それぞれで所属するユーザクラスタを判別するための学習を行い、補正結果画像を用意した。被験者に、その中からユーザの嗜好に最も合致する画像を選んでもらい、回答結果を集計した。結果を Figure 6.11 に示す。ブレンド率の可視化結果表示ありの方が、満足度が高い結果となり、カイ二乗検定を用いて検定した結果、 $\chi^2 = 3.9729$ ,  $p = 0.0462$  となり、 $p < 0.05$  で可視化結果表示ありの結果に有意差が認められた。

また、ブレンド率の可視化結果表示あり、表示なしで、サンプル学習画像 9 枚に対して補正してもらい、補正にかかる合計操作時間の平均を算出した。結果を Figure 6.12 に示す。ブレンド率の可視化結果表示ありの方が、短時間で学習が終わっていることが読み取れる。さらに、サンプル学習画像 1 枚あたりの補正操作回数を計測した。結果を Figure 6.13 に示す。第 3 四分位点と最大値に差が出ているため、どのような補正にしようか迷ったときに可視化表示が有効であることが読み取れる。

実験後のアンケートでは、ブレンド率の可視化結果を表示しながら学習を行うことで、「操作しやすい」、「好みのフィルターの傾向がわかる」、「偏りから次はこのような編集をしてみようと加工の幅が広

がる」などというポジティブな感想をもらった。一方で、「過去の結果に無意識に引きずられる」というネガティブな感想もあった。

### 6.3.9 新規ユーザのサンプル学習画像の枚数の比較

6.2.4 節の被験者に対して以下の実験を行った。クラスタを判別するために必要なサンプル学習画像の枚数を検証するため、新規ユーザがサンプル学習画像 6 枚、9 枚、12 枚に対して補正が終わった時点で、該当するユーザクラスタを求め、それぞれ補正結果画像を用意した。被験者に、その中からユーザの嗜好に合致する画像を選んでもらい、回答結果を集計した。結果を Figure 6.14 に示す。この結果から、新規ユーザが所属するユーザクラスタを特定するために必要なサンプル学習画像の枚数は、9 枚程度で十分ということが読み取れる。

## 6.4 考察と課題

本手法を用いた評価実験から、同じようなシーンで撮影された写真群に対してユーザの嗜好に合わせて一括補正する場合、画像カテゴリやユーザによって補正傾向が異なることを示した。また、初期ユーザはユーザクラスタを生成するために、サンプル学習画像に対する手動補正に時間がかかるが、新規ユーザは少数のサンプル学習画像に対する補正結果をブレンドしながら所属するユーザクラスタを探索し特定し、少ない操作によってその後の画像一括補正処理を行うことで、ユーザに課するタスクをなるべく最小限におさえつつ、ユーザの好みを反映して画像を補正できることを示した。

今後の課題として、以下の 2 点を挙げる。

1 点目の課題として分析手法の再検討が考えられる。具体的には、新しい画像が所属すべきクラスタの判別分析手法（現時点ではサポートベクターマシン）の再検討、および画像特徴量から画像補正パラメータを推定する回帰分析手法（現時点では線形重回帰分析）の再検討を進めたい。

2 点目の課題として画像補正パラメータの拡張が考えられる。現時点の実装では、色補正パラメータの一種であるトーンカーブを採用し、コントロールポイントを 3 点に絞っている。これに対して、トーンカーブのコントロールポイントを増やす、トーンカーブ以外の画像補正パラメータを用いる、といった拡張を試し、これによって補正結果の満足度が向上するかを検証したい。

以上を検討した上で、大人数の被験者を交えた実験を実施し、CrowdRetouch の有用性を再検証したい。

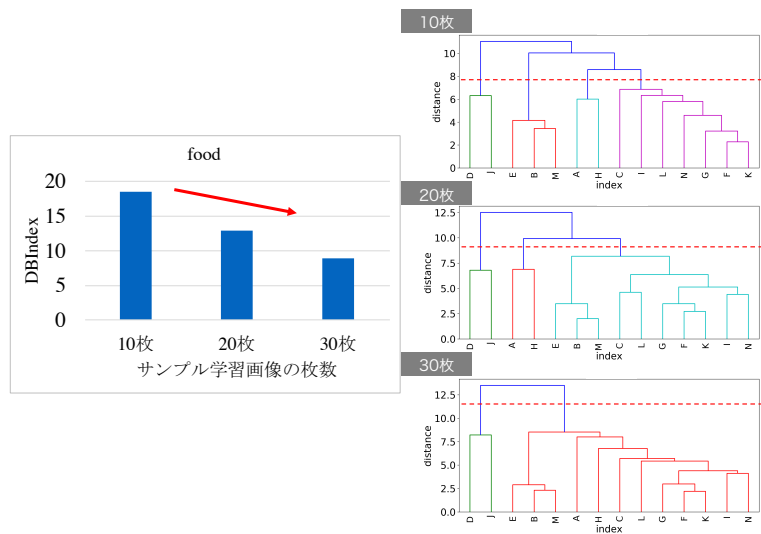


## 6.5 まとめ

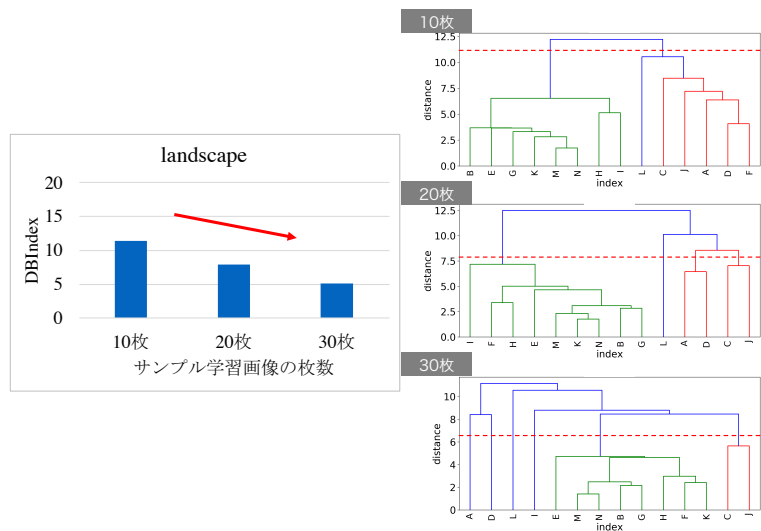
同じようなシーンで撮影された写真群に対して、各ユーザの画像補正の傾向を反映した画像一括補正を可能とするシステム”CrowdRetouch”を提案し、その有効性を検証した。CrowdRetouch では、まずサンプル学習画像をカテゴリ分類し、画像特徴量によって画像クラスタを生成する。次にある程度の人数の初期ユーザに対して、画像カテゴリ、画像クラスタごとに一定数のサンプル学習画像を手動で画像補正させ、その特性によって初期ユーザをクラスタリングする。続いて各ユーザクラスタに対して画像補正パラメータの回帰分析を適用し、その結果を画像一括補正に用いる。一方、新規ユーザは、各ユーザクラスタの回帰式から算出される加工パラメータによって生成された補正結果画像をブレンドし、所属するユーザクラスタを特定する。ここで、各補正結果画像のブレンド率を Parallel Coordinate Plot で可視化することで、ユーザに好みの補正傾向を気づかせることができる。新規ユーザが所属するユーザクラスタが特定できたら、各ユーザクラスタの回帰式と、ブレンド率の学習結果を用いて、他の画像を一括補正する。このように、初期ユーザはユーザクラスタを生成するために、サンプル学習画像に対する手動補正に時間がかかるが、新規ユーザは少数のサンプル学習画像に対する補正結果をブレンドしながら所属するユーザクラスタを探索し特定することで、少ない操作量によってその後の画像一括補正処理を可能にした。

## 6.6 むすび

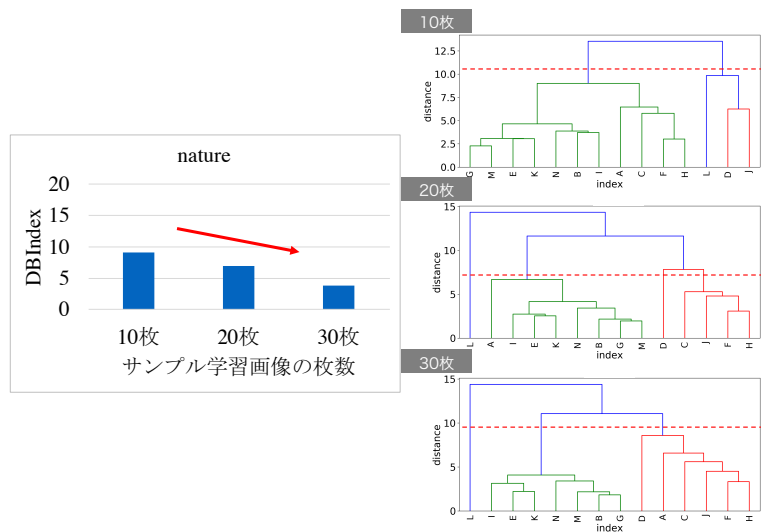
本章では、CrowdRetouch の実行結果について述べた。まず、6.2 節ではユーザテストの方法を述べ、6.3 節では実験結果を説明し、6.4 節では実験結果から導かれる考察と課題を述べた。最後に、6.5 節で CrowdRetouch の総括した。



(a) 画像カテゴリ:food



(b) 画像カテゴリ:landscape

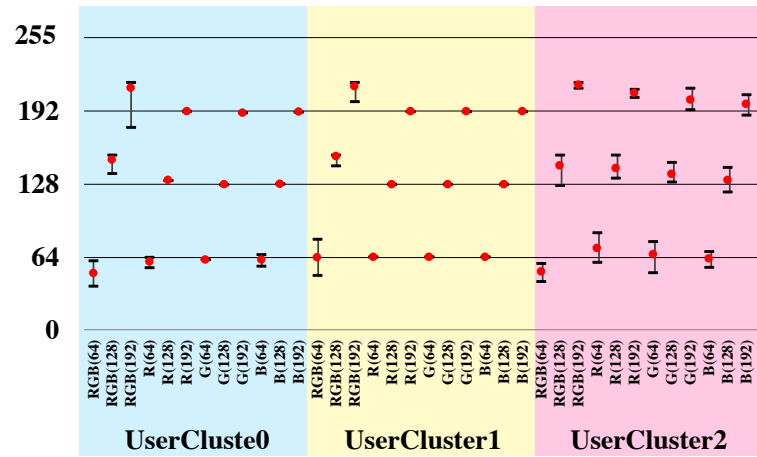


(c) 画像カテゴリ:nature

Figure 6.8: 初期ユーザのユーザクラスタの精度とデンドログラムによるクラスタリングの結果.

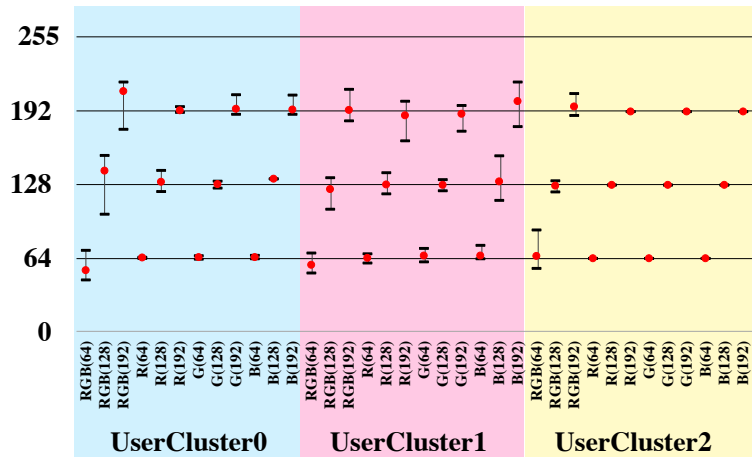


### Photo Category : food



(a) 画像カテゴリ:food

### Photo category : landscape



(b) 画像カテゴリ:landscape

Figure 6.9: 画像カテゴリごとの補正パラメータの補正量.

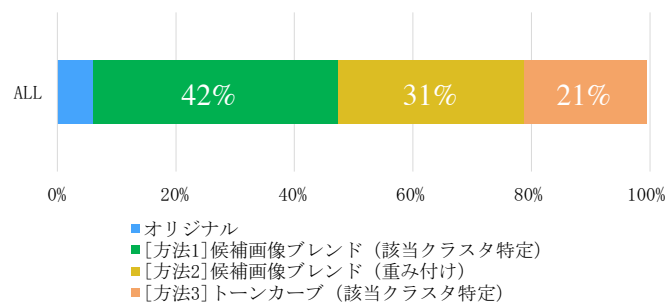


Figure 6.10: 補正パラメータの推定方法の比較.

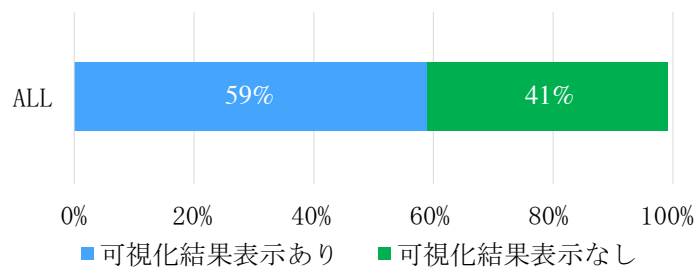


Figure 6.11: ブレンド率の可視化結果表示の効果.

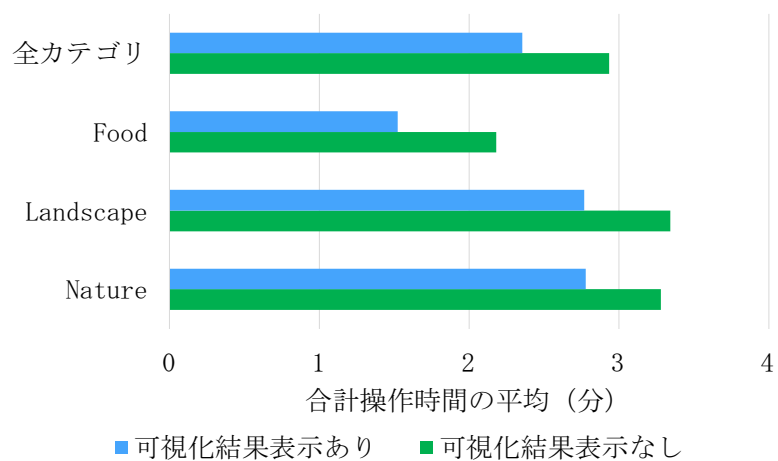


Figure 6.12: サンプル学習画像に対して補正にかかる合計操作時間の平均.

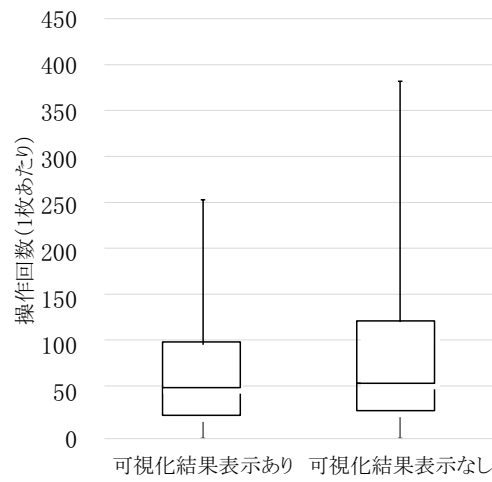


Figure 6.13: サンプル学習画像 1 枚あたりの補正操作回数.

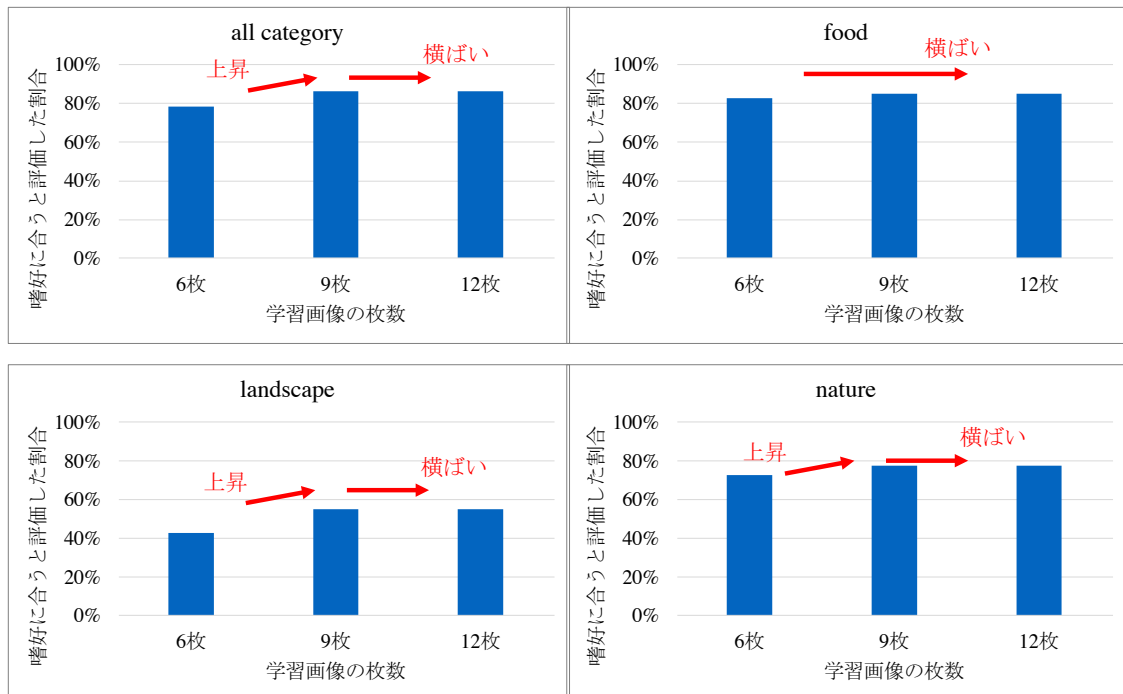


Figure 6.14: 新規ユーザのサンプル学習画像の枚数の比較.

## 第7章 結言

### 7.1 本論文のまとめ

本論文では、システムがユーザの嗜好を学習し、その学習過程を可視化することで、効果的にコンテンツを推薦、加工するためのユーザインタフェースを紹介した。著者の着目点は、コンテンツを推薦、加工する際に、ユーザの嗜好を学習するためのタスクが大きいと実用性が下がるため、タスクを小さくしたい、小さいタスクからは少数の入力しか得られないため、適用可能な学習手法が限定されるという点である。これらの制約を解決するため、少数の入力からの推薦、加工を効果的に行う仕組みとして、可視化によるナビゲーションやユーザクラスタリングが有用であることを示した。以下では本論文の各章で述べた内容を列挙し、内容を総括する。

第1章では、コンテンツ推薦やコンテンツ加工について、現実社会での実用上の課題と学術的な課題を提起した。具体的には、アクセス可能なコンテンツが増えたため、自身の嗜好に合ったコンテンツを探しにくいという課題や、コンテンツの補正パラメータを自身の嗜好に合わせて調節すると手間と時間がかかるという課題があることを述べた。これらの課題に対して、著者は、コンテンツの特徴を抽出し、その特徴量空間において、ユーザの嗜好に合うコンテンツの位置を探索したり、コンテンツの特徴量とコンテンツの補正パラメータの関係を学習したりする際に、可視化によるナビゲーションやユーザクラスタリングを導入することで、コンテンツ推薦やコンテンツ加工の結果に効率よくユーザの嗜好を反映できるのではないかという仮説を示した。

第2章では、第3章～第6章で提案する手法により関連した研究として、コンテンツ推薦技術、コンテンツ検索インタフェース、コンテンツのパラメータの最適化の事例について述べた。

第3章と第4章では、音楽コンテンツを対象に、ユーザの嗜好に合わせたコンテンツ推薦のためのユーザインタフェース（MusiCube）を提案した。MusiCubeでは、まず、楽曲から特徴量を抽出し、それらを並べて多次元の特徴量空間を生成する。次に、その特徴量空間において、対話型進化計算を用いて、ユーザと計算機の自然なインタラクションによってユーザの嗜好を学習する。この際に、ユーザに自身の嗜好の傾向を気づかせるために、学習過程の可視化を行うことで、少ないタスクからユーザの嗜好を推薦結果に反映することを実現した。これらの処理手順について説明し、実行結果と考察を述べた。

第5章と第6章では、写真コンテンツを対象に、ユーザの嗜好に合わせたコンテンツ加工のためのユー

ザインタフェース (CrowdRetouch) を提案した。CrowdRetouch では、まずサンプル学習画像をカテゴリ分類し、画像特徴量によって画像クラスタを生成する。次にある程度の人数の初期ユーザに対して、画像カテゴリ、画像クラスタごとに一定数のサンプル学習画像を手動で画像補正させ、その特性によって初期ユーザをクラスタリングする。続いて各ユーザクラスタに対して画像補正パラメータの回帰分析を適用し、その結果を画像一括補正に用いる。一方、新規ユーザは、各ユーザクラスタの回帰式から算出される加工パラメータによって生成された補正結果画像をブレンドし、自身が所属するユーザクラスタを特定する。ここで、各補正結果画像のブレンド率を Parallel Coordinate Plot で可視化することで、ユーザに好みの補正傾向を気づかせることができる。新規ユーザが所属するユーザクラスタを特定できたら、各ユーザクラスタの回帰式と、ブレンド率の学習結果を用いて、他の画像を一括補正する。このように、初期ユーザはユーザクラスタを生成するために、サンプル学習画像に対する手動補正に時間がかかるが、新規ユーザは少数のサンプル学習画像に対する補正結果をブレンドしながら所属するユーザクラスタを探索し特定することで、少ない操作によってその後の画像一括補正処理を実現する。これらの処理手順について説明し、実行結果と考察を述べた。

以上のように、本論文では、システムがユーザの嗜好を学習し、効果的にコンテンツを推薦、加工することを実現するために、可視化によるナビゲーションやユーザクラスタリングを導入したユーザインタフェースを提案した。

提案手法の特徴を Table 7.1 に示す。

Table 7.1: 各提案手法の特徴.

手法	学習過程の可視化	ユーザクラスタリング
MusiCube	<ul style="list-style-type: none"> <li>一貫性のある選曲ができる</li> <li>ユーザに嗜好の傾向を気づかせる</li> </ul>	
CrowdRetouch	<ul style="list-style-type: none"> <li>ユーザに嗜好の傾向を気づかせる</li> </ul>	<ul style="list-style-type: none"> <li>新規ユーザはクラスタ判別だけで一括画像補正可</li> </ul>

第 3 章と第 4 章で提案した MusiCube では、特徴量空間における対話型進化計算の学習過程や学習結果の可視化を導入することで、楽曲の特徴を確認しながら一貫性のある選曲が可能になる点やユーザに自身の嗜好の傾向を気づかせることができる点が特徴である。

第 5 章と第 6 章で提案した CrowdRetouch では、ユーザの画像補正傾向によって、ユーザクラスタリングを導入することで、初期ユーザによってクラスタリングを構成されれば、新規ユーザはクラスタ判別のみで一括画像補正が可能になる点や、候補のフィルターのブレンド率の探索過程を可視化することにより、好みのフィルターの傾向を見ながら補正できる点が特徴である。

このように、可視化によるナビゲーションやユーザクラスタリングの導入により、ユーザに大きなタスクを課すことなく、効果的にコンテンツを推薦、加工する手法を提案できたと言える。

本論文では、Table1.2 で示すように、メディアコンテンツとして、「音楽」と「写真」を取り上げ、音楽と推薦、写真と加工についての提案手法を説明した。今回の論文で対象外とした音楽と加工については、自動作曲や自動編曲などがあり、さまざまな研究がなされている [Sugimoto07] [Ohshima07]。CrowdRetouch で音楽コンテンツを適用する場合、音楽は写真の加工と違って、音楽理論による制約があるので、CrowdRetouch に制約を課すための仕組みを追加したい。写真と推薦については、ユーザの嗜好に合わせて写真を選択提示する研究は多くされている [Tang12] [Sang12] [Yang10]。MusiCube で写真コンテンツを適用する場合、写真は音楽と違って、1 曲ずつじっくり確認する必要がないので、MusiCube の提示方法に検討の余地があると考えている。

## 7.2 実世界への適用における課題

実世界において、MusiCube や CrowdRetouch を適用するために今後取り組むべき課題を述べ、研究の指針としたい。

まず、大規模なメディアコンテンツのデータベースに対応する必要がある。現在のシステムの開発や検証は、小規模なデータを対象としているため、時間的、計算量的コストの制約がない。しかし、実際に運用する場合には、大規模なメディアコンテンツを対象とするため、各種アルゴリズムの計算量が増え、処理が遅くなってしまうという課題や、学習過程の可視化結果が煩雑になり、重要な情報がユーザに伝わりにくいという課題が出てくる。これらの課題に対して、アルゴリズムの計算量を減らす工夫や、大規模データを要約して可視化する工夫が必須となる。大規模データを要約して可視化する手法として、重要な情報を目立つようにする方法などが考えられる。

次に、メディアコンテンツを表す情報を洗練させる必要がある。現状、MusiCube では、計算量の少ない局所特徴量を用いているが、実際には曲全体を表す特徴量を抽出する必要がある。時系列の動的変化までも捉えられるような特徴量を設計する必要がある。また、CrowdRetouch では、あらかじめ付与されたキーワードと画像特徴量を用いているが、キーワードは自動的に付与される必要がある。さらに、解像度の高い画像に対応するためには、画像解析にかかる計算コストを抑えるために、画像特徴量をあらかじめ選別する必要があるかもしれない。

そして、時間の経過によるユーザの嗜好の変化にも対応する必要がある。システムの評価のために実施したユーザテストでは、時間の経過によって、ユーザの嗜好がどのように変化するかという点に対しての検証ができていない。ユーザの嗜好は、メディアの影響、ユーザ自身の気分の変化などによって

目まぐるしく変化していくため、システムがこのような変化に対して追従、予測できる仕組みが必要である。

さらに、本論文で提案する手法の大きな特徴である学習過程の可視化は、Scatter plot や Parallel Coordinate Plot を採用している。システムの評価のために実施したユーザテストでは、大学で情報科学を専攻している学生を対象としているため、Scatter plot や Parallel Coordinate Plot による可視化結果を読み取ることに抵抗がなく、自身の嗜好の傾向を観察しながら、学習を進めることができた。しかし、一般の人を対象とした場合、Scatter plot や Parallel Coordinate Plot による可視化結果を簡単に読み取れない場合がある。そのため、実際に運用する場合は、可視化結果から嗜好の傾向を知らせるためのガイド文を自動生成できるような仕組みなどを導入し、より使いやすいシステムにしていく必要があると考えている。

最後に、本論文で提案する効果的に解を探索するために可視化によるナビゲーションやユーザクラスタリングを導入するという手法は、人間の心の状況や希望を迅速に表現できるという点で、心の働きのありとあらゆる側面での応用が期待される。心の働きを定量的に分析することは、それが曖昧で複雑なゆえに、非常に難しい課題である。このような課題に対して、心理学、認知学の観点からもアプローチし、新しい展開を生み出していきたい。

# 謝 辞

私が研究活動を始めた 2009 年から今日までの 8 年間、大変多くの方々の御指導を頂きました。博士論文の執筆を終えるにあたり、この場をお借りして心より感謝の意を表します。

本研究を行うにあたり、終始懇切なる御指導、御鞭撻を賜りました指導教員であるお茶の水女子大学の伊藤貴之教授に深く感謝致します。卒業研究、修士論文に引き続き、博士論文の御指導を頂き、研究面・生活面など様々な場面で御助言を頂きました。長きにわたり、暖かく見守って頂いたことを心より感謝致します。また、本論文の審査の過程において、お茶の水女子大学の椎尾一郎教授、小口正人教授、小林一郎教授、萩田真理子准教授には数々の御助言と御指導を賜りました。大変お忙しい中、本論文の副査をお引き受け下さいましたこと深く感謝致します。そして、津田塾大学の小館亮之准教授には、研究活動を始めるにあたり御指導を頂きました。心より感謝致します。

本論文の実験は、伊藤研究室の皆様にご協力して頂きました。また、実験のデータとして、RWC 研究用音楽データベース（ポピュラー音楽、ジャズ音楽、音楽ジャンル）と文京区立図書館に所蔵されている CD に収録された楽曲を使用させて頂きました。

最後になりましたが、陰ながら支えてくれた家族やその他御支援下さいました皆様に心より感謝致します。



## A：業績

### 論文誌

- [1] 齊藤優理, 伊藤貴之, “ MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース, ”可視化情報学会論文集, Vol.34, No.9, pp.17-27, Oct.2014.(学会賞(技術賞), 第26期可視化情報学会).
- [2] 齊藤優理, 伊藤貴之, “ CrowdRetouch: ユーザの画像補正傾向に基づく画像一括補正システム, ”芸術科学会論文誌, Vol.15, No.4, pp.147-156, Nov.2016.(優秀論文賞, NICOGRAPH 2016).

### 国際会議

- [3] Y. Saito, T. Itoh, MusiCube: A Music Selection Interface featuring Interactive Evolutionary Computing in Feature Spaces, IEEE Pacific Visualization Symposium 2011, pp.33-34, March, 2011.
- [4] Y. Saito, T. Itoh, MusiCube: A Music Selection Interface featuring Interactive Evolutionary Computing in Feature Spaces, 15th International Conference on Information Visualization (IV2011), pp.426, July, 2011.
- [5] Y. Saito, T. Itoh, MusiCube: A Visual Music Recommendation System featuring Interactive Evolutionary Computing, Visual Information Communication - Information Symposium (VINCI'11), August, 2011.
- [6] E. Gomi, Y. Saito, T. Itoh, Visualization of Crowd-Powered Impression Evaluation Results, NICOGRAPH International 2015, June, 2015.
- [7] E. Gomi, Y. Saito, T. Itoh, Visualization of Crowd-Powered Impression Evaluation Results, 19th International Conference on Information Visualisation (IV2015), pp.89-94, July, 2015.

- [8] E. Gomi, Y. Saito, T. Itoh, M. Hagita, M. Takatsuka, A Technique for Ranking and Visualization of Crowd-Powered Subjective Evaluations, 21st International Conference on Information Visualisation (IV2017), pp.52-56, July, 2017.

## 国内会議

- [9] 斉藤優理, 小舘 亮之, 伊藤貴之, “対話型遺伝的アルゴリズムを用いたユーザの目的に合った楽曲推薦システム,” 第2回データ工学と情報マネジメントに関するフォーラム, Mar.2010.
- [10] 斉藤優理, 伊藤貴之, “MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース,” 第86回音楽情報科学研究会 (情報処理学会), Jul.2010.
- [11] 斉藤優理, 伊藤貴之, “MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース,” 第18回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2010), pp.119-120, Dec.2010.
- [12] 斉藤優理, 伊藤貴之, “MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース,” 人工知能学会全国大会, Jun.2011.
- [13] 斉藤優理, 伊藤貴之, “MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース,” 第19回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2011), Dec.2011.
- [14] 五味恵理華, 斉藤優理, 伊藤貴之, “遺伝的アルゴリズムを用いた効率的なアンケートの一手法,” 第76回情報処理学会全国大会, 3ZA-2, Mar.2014.
- [15] 斉藤優理, 伊藤貴之, “対話型遺伝的アルゴリズムを用いた画像フィルタ調整に関する初期検討,” 画像の認識・理解シンポジウム (MIRU2014), Jul.2014.
- [16] 五味恵理華, 斉藤優理, 伊藤貴之, “対話型遺伝的アルゴリズムを用いた効率的なアンケート収集の一手法 -女性の装いを例にして-, ” NICOGRAPH 2014, Nov.2014.
- [17] 五味恵理華, 斉藤優理, 伊藤貴之, “対話型遺伝的アルゴリズムを用いたアンケート収集とその可視化,” 第22回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2014), デモ/ポスター発表, Dec.2014.

- [18] 五味恵理華, 齊藤優理, 伊藤貴之, “対話型遺伝的アルゴリズムを用いたアンケート収集とその可視化,” 情報処理学会第 77 回全国大会, 5Z-05, Mar.2015. (学生奨励賞).
- [19] 齊藤優理, 伊藤貴之, “CrowdRetouch: ユーザの好みを反映した画像一括補正システム,” 電子情報通信学会パターン認識・メディア理解研究会 (PRMU) 2016 年 1 月研究会, Jan.2016.
- [20] 五味恵理華, 齊藤優理, 伊藤貴之, 萩田 真理子, 高塚 正浩: “対話型遺伝的アルゴリズムを用いたアンケート収集とその可視化,” 第 8 回データ工学と情報マネジメントに関するフォーラム (DEIM), E5-4, Mar.2016. (学生プレゼンテーション賞).

## B：参考文献

- [Goldberg92] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, Using collaborative filtering to weave an information tapestry, *Communications of the ACM* 35 (1992), no.12, 61-70, 1992.
- [Shardanand95] U. Shardanand, P. Maes, Social information filtering: algorithms for automating “word of mouth”, *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.210-217, May 1995.
- [Hayes00] C. Hayes, P. Cunningham, Smart radio: Building music radio on the fly, *Expert Systems* 2000, pp.2-6 2000.
- [French01] J. C. French, D. B. Hauver, Flycasting: On the Fly Broadcasting, *DELOS workshop: Personalisation and recommender systems in digital libraries*. 2001.
- [Lastfm] <https://www.last.fm/ja/> (閲覧日: 2017 年 9 月 10 日).
- [Genius] <https://genius.com/> (閲覧日: 2017 年 9 月 10 日).
- [Matsubara10] 松原正樹, 李軒, and 斎藤博昭, “音楽への感性を開拓するメタ認知支援ツール,” *日本認知科学会第 27 回大会講演論文集*, No.P3-44 (2010) pp.448-452, 2010.
- [Hoashi03] K. Hoashi, K. Matsumoto, and N. Inoue, Personalization of user profiles for content-based music retrieval based on relevance feedback, *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, pp.110-119, 2003.
- [Cho04] S. B. Cho, Emotional image and musical information retrieval with interactive genetic algorithm, *Proceedings of the IEEE* 92.4 (2004): 702-711. 2004.
- [Rho07] S. Rho, E. Hwang, and M. Kim, Music information retrieval using a GA-based relevance feedback, *Multimedia and Ubiquitous Engineering*, 2007. MUE’07. International Conference on. IEEE, pp.739-744, 2007.

- [Sotiropoulos08] D. N. Sotiropoulos, A. S. Lampropoulos, and G. A. Tsihrintzis, MUSIPER: a system for modeling music similarity perception based on objective feature subset selection, *User Modeling and User-Adapted Interaction* 18.4 (2008): 315-348, 2008.
- [Pandora] <http://www.pandora.com/> (閲覧日: 2017 年 9 月 10 日).
- [Musicoverly] <http://b2b.musicoverly.com/> (閲覧日: 2017 年 9 月 10 日).
- [Kolhoff06] P. Kolhoff, J. Preub, J. Loviscach: Music Icons: Procedural glyphs for audio files, *IEEE SIBGRAPI*, pp.289-296, 2006.
- [Oda07] 小田瑞穂, 伊藤貴之, “MIST: 音楽アイコンの自動選択の一手法,” 第 15 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2007), pp.115-116, 2007.
- [Machida11] 町田和嘉子, 伊藤貴之, “Lyricon: 複数アイコンによるビジュアルな音楽選択インタフェース,” 情報処理学会第 141 回ヒューマンコンピュータインタラクション研究会, HCI-141-8, 2011.
- [Kohonen98] T. Kohonen: Self-organized formation of topologically correct feature maps, in *Neurocomputing: foundations of research*, pp.509-521, Cambridge, MA, USA: MIT Press, 1988.
- [Jolliffe86] I. T. Jolliffe, *Principal Component Analysis and Factor Analysis*, Springer New York, pp.115-128, 1986.
- [Kruskal78] J. B. Kruskal, M. Wish, *Multidimensional Scaling*, Vol.11, Sage, 1978.
- [Pampalk01] E. Pampalk, *Islands of Music: Analysis, Organization, and Visualization of Music Archives*, Master's thesis, Vienna University of Technology, 2001.
- [Morchen05] F. Morchen, A. Ultsch, M. Nocker, C. Stamm, Databionic visualization of music collections according to perceptual distance, In *Proceedings 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pp.396-403, 2005.
- [Neumayer05] R. Newmayer, M. Dittenbach, A. Rauber, PlaySOM and PocketSOMPlayer: alternative interfaces to large music collections, In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pp.618-623, 2005.
- [Leitich07] S. Leitich, M. Tops, *Globe of Music Library Visualization Using GEOSOM*, In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, 2007.

- [Hamawaki09] 浜脇修平, 石先広海, 帆足啓一郎, 滝嶋康弘, “ 楽曲の音響的特徴空間視覚化を利用した音楽検索システム, ” 情報処理学会音楽情報科学研究会, pp.29-34, 2009.
- [Stober10] S. Stober, A. Nurnberger, MusicGalaxy: A Multi-focus Zoomable Interface for Multi-facet Exploration of Music Collections, In Proceedings of 7th International Symposium on Computer Music Modeling and Retrieval(CMMR'10), Vol.6684, pp.273-302, 2010.
- [Inselberg85] A. Inselberg, The Plane with Parallel Coordinates, The Visual Computer, Vol.1, No.2, pp.69-97, 1985.
- [Nagasaki09] 長崎あずさ, 伊藤貴之, 伊勢昌之, 宮下光輔, “ 2 変数詳細度制御を用いた大規模データの可視化, ” 第 17 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2009), pp.135-136, 2009.
- [Elmqvist08] N. Elmqvist, P. Dragicevic, J. Fekete, Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation, IEEE Transactions on Visualization and Computer Graphics, Vol.14, No.6, pp.1141-1148, 2008.
- [Zheng11] Y. Zheng, A. Gomi, T. Itoh, ImageCube: A Browser for Image Collections Associated with Multi-Dimensional Datasets, 15th International Conference on Information Visualization (IV2011), pp.12-17, 2011.
- [Goto04] 後藤孝行, 後藤真孝, “ Musicream: 楽曲を流してくっつけて並べることのできる新たな音楽再生インタフェース, ” 第 12 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2004) 論文集, pp.53-58, 2004.
- [Hamasaki14] M. Hamasaki, M. Goto, and T. Nakano, Songrium: a music browsing assistance service with interactive visualization and exploration of protect a web of music, Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014.
- [Ito09] 伊藤冬子, 廣安知之, 三木光範, and 横内久猛, “ 対話型遺伝的アルゴリズムにおける嗜好の多峰性に対応可能な個体生成方法, ” 人工知能学会論文誌 24.1 (2009): 127-135, 2009.
- [Yamashiro07] 山代大輔, 吉川大弘, 古橋武, “ 遺伝的アルゴリズムの解探索過程の可視化による遺伝的演算効果の把握と解探索の効率化, ” 情報処理学会論文誌, Vol.48, No.SIG 14 (TOM 16), 2007.
- [Howe06] J Howe, The rise of crowdsourcing, Wired magazine 14.6 (2006): 1-4, 2006.

- [Jaroensri15] R. Jaroensri, et al, Predicting range of acceptable photographic tonal adjustments, Computational Photography (ICCP), 2015 IEEE International Conference on. IEEE, 2015.
- [Zhu14] J.Y. Zhu, et al, Mirror mirror: Crowdsourcing better portraits, ACM Transactions on Graphics (TOG) 33.6 (2014): 234, 2014.
- [Ahn08] L. Von Ahn, Human computation, Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. IEEE, 2008.
- [Marks97] J. Marks, et al., Design galleries: a general approach to setting parameters for computer graphics and animation, ACM SIGGRAPH '97, pp.389-400, 1997.
- [Shapira09] L. Shapira, A. Shamir, D. Cohen-Or, Image appearance exploration by model-based navigation, Computer Graphics Forum (EUROGRAPHICS 2009), pp.629-638, 2009.
- [arakawa07] 荒川薫, 野本 弘平, “インタラクティブ進化計算を用いた顔画像美観化システム,” 第21回 ファジィシステムシンポジウム, 8E3-2, 2007.
- [Kapoor14] A. Kapoor, J. Caicedo, D. Lischinski, S. Kang, Collaborative Personalization of Image Enhancement, International Journal of Computer Vision, Vol.108, No.1-2, pp.148-164, 2014.
- [koyama15] 小山裕己, 坂本大介, 五十嵐健夫, “効果的な写真の色補正を実現するための自己強化するシステム,” VC/GCAD 合同シンポジウム, 2015.
- [Bychkovsky11] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, Learning Photographic Global Tonal Adjustment with a Database of Input / Output Image Pairs, IEEE Conference on Computer Vision and Pattern Recognition, pp.97-104, 2011.
- [koyama16] 小山裕己, 坂本大介, and 五十嵐健夫, “ヒューマンコンピュテーションによるパラメタ空間解析を用いた視覚デザイン探索,” コンピュータソフトウェア Vol.33 No.1 63-77, 2016.
- [Greg10] G. Little, et al. TurkIt: human computation algorithms on mechanical turk, Proceedings of the 23rd annual ACM symposium on User interface software and technology. ACM, 2010.
- [RWC] RWC-Music-Database. <http://staff.aist.go.jp/m.goto/RWC-MDB/> (閲覧日: 2017年9月10日).
- [MIR] O. Lartillot, MIRtoolbox, <http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox/> (閲覧日: 2017年9月10日).

- [Hiroyasu03] 廣安知之, et al, “ 遺伝的アルゴリズムにおける実数値ベクトル表現, 世代交代モデル, 母集団分割効果の検討, ” 同志社大学理工学研究報告 44.1 (2003): 25-35, 2003.
- [Choo09] J. Choo, S. Bohn, and H. Park, Two-stage framework for visualization of clustered high dimensional data, In IEEE Symposium on Visual Analytics Science and Technology (VAST 2009), pp.67-74, 2009.
- [Choo10] J. Choo, H. Lee, J. Kihm, H. Park, iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction, In Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2010), pp.27-34, 2010.
- [Goto02] 後藤真孝, “ リアルタイム音楽情景記述システム: サビ区間検出手法, ” 情報処理学会研究報告 音楽情報科学 (MUS) 2002.100 (2002-MUS-047) (2002): 27-34, 2002.
- [ImageNet] ImageNet, <http://www.image-net.org/> (閲覧日: 2017 年 9 月 10 日).
- [LIVSVM] LIVSVM, <http://www.csie.ntu.edu.tw/~cjlin/libsvm/> (閲覧日: 2017 年 9 月 10 日).
- [Davies79] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol.PAMI-1, No.2, pp.224-227, 1979.
- [PHOTOHITO] PHOTOHITO, <http://photohito.com/> (閲覧日: 2017 年 9 月 10 日).
- [Tang12] X. Tang, et al. Intentsearch: Capturing user intention for one-click internet image search, IEEE transactions on pattern analysis and machine intelligence 34.7 (2012): 1342-1353, 2012.
- [Sang12] J. Sang, C. Xu, and D. Lu, Learn to personalized image search from the photo sharing websites, IEEE Transactions on Multimedia 14.4 (2012): 963-974, 2012.
- [Yang10] L. Yang, A. Hanjalic, Supervised reranking for web image search, Proceedings of the 18th ACM international conference on Multimedia. ACM, 2010.
- [Sugimoto07] 杉本知仁, et al, “ 脳波の解析に基づく個人感性獲得による自動作曲, ” 人工知能学会全国大会論文集 21 (2007): 1-4, 2007.
- [Ohyama07] K. Ohyama, T. Itoh, DIVA: An Automatic Music Arrangement Technique Based on Impression of Images., International Symposium on Smart Graphics. Springer, Berlin, Heidelberg, 2007.



Advanced Science,  
Graduate School of Humanities and Sciences,  
Ochanomizu University,  
Yuri Saito