

マルチプラットフォーム上の C++による電子音色辞書の構築

山田 祐雅^{†1, a)} 津田 浩利^{†1, b)} 小坂 直敏^{†1, c)}

われわれは、マルチメディアコンテンツ制作の支援ツールとして、環境音データベースに対して様々な方法による音色検索が行える電子音色辞書を構築している。これまでのシステムでは各機能は異なる言語で書かれていたが、これらのモジュールを統合し、処理速度の向上、およびマルチプラットフォーム対応とするため、新たに C++ で再構築を行った。本稿では新たなシステムの検索、および音オブジェクトの一覧表示と 3D 表示による GUI の運動性など、新たな実装法について報告する。

1. はじめに

近年インターネットや高性能コンピュータの普及により、個人での DTM や動画制作などのマルチメディアコンテンツ制作が盛んに行われている。制作の際には、適切な音素材を効率的に選択し使用することが重要である。

われわれは、マルチメディアコンテンツ制作の支援ツールとして、環境音のデータベースに対して音色の登録や様々な音色の検索を行うことができる「電子音色辞書」を構築してきた[1]。同システムでは、Adobe Flex 言語を用いた音色ブラウザ機能や、一つの音オブジェクトを MFCC で表現した後、主成分分析により 3次元に次元圧縮し、これを 3次元空間上に球体で配置する、Adobe Flash 言語で書かれた 3D 音色表示機能[2]などを有す。

これまでのシステム構築は、各モジュールの機能を実装し、その動作確認を主目的としてきた。現在これらの目的はほぼ達成され、次の目標は具体的な音検索に際し、これらの機能を統合してスピーディーにモジュール間を移動したり、連携して操作できるシステムを考える。

これまでのサブシステムは異なる言語で構築されており、別アプリケーションとして動作し、各サブシステムのスムーズな連携ができなかった。また、ActionScript 言語のアップデートによる移植性と保守性が低く、処理速度がネイティブなプログラム言語に比べ遅いということが課題であった。

そこで、本検討では各サブシステムの強い連携を考慮し、その上にさらに処理速度の向上、マルチプラットフォーム対応を可能とする新たな電子音色辞書を提案、再構築することを目的とする。

本稿では、再構築するにあたり新たに導入したクリエイティブ・コモンズの考え方や、検討した開発言語と実装について述べ、ユーザビリティ評価実験結果について報告する。

2. システムの構成

2.1 新機能の設計

以下の機能を新しく実装する。一つはクリエイティブ・コモンズである。本システムは、制作用に音を提供することを目的とし、ダウンロードした音を自由に利用するためのライセンスである。

2.1.1 クリエイティブ・コモンズ

クリエイティブ・コモンズ・ライセンス（以下、CC ライセンス）[3]を提供している国際的非営利組織とそのプロジェクトの総称で、様々な作者が著作物の再利用の条件を CC ライセンスの種類を組み合わせることにより意思表示することができ、著作権を保持したまま作品を自由に流通させることができる。また、受け手は作者が提示したライセンス条件の範囲内で自由に使用することができるという枠組みである。

以下に作品の利用のための条件を挙げる。

- ① 表示
- ② 表示-継承
- ③ 表示-改変禁止
- ④ 表示-非営利
- ⑤ 表示-非営利-継承
- ⑥ 表示-非営利-改変禁止

これらの 4 つの条件を組み合わせた 6 種類の CC ライセンスと、全ての権利を主張する“copyright”，すべての権利を放棄する“public domain”の 8 種類から、作者がどのように流通して欲しいかを選ぶことができる。

2.1.2 音色の登録とアップロード

電子音色辞書は、ユーザは音色のアップロードをする際、音色のデータベース登録に必要な発音体、擬音語、分類等のタグ情報を入力し、上記 8 種類の CC ライセンスからひとつ選ぶことにより、使用者は指定された CC ライセンスを理解の上で制作や再配布を行うことができる枠組みを採用している。ユーザが音データをアップロードすることにより、音色データベースに堆積して行くことにより膨大なデータを他ユーザに提供することができる。

また、アップロードする際に“copyright”を選択した場合、ローカルのデータベースに登録され、ネットワーク上のア

†1 東京電機大学
Tokyo Denki University
a) 08fi130@ms.dendai.ac.jp
b) 12fmi25@ms.dendai.ac.jp
c) osaka@im.dendai.ac.jp

ップロードは行わず、他ユーザは使用できないようになる。

2.2 開発言語の検討

開発言語は、処理速度の向上とマルチプラットフォーム対応を考慮し、様々な OS のネイティブ言語である C++ を採用した。

2.3 3D グラフィックス API の検討

3D 音色表示機能では 3D グラフィックスを用いるため、クロスプラットフォームの 3D グラフィックス API として OpenGL[4]を採用した。OpenGL はハードや OS に依存しない、移植性の高いプログラミングインタフェースである。同じ 3D グラフィックス API として Direct3D があるが、マイクロソフトの DirectX の一部であり Windows 上でしか動作しないため、マルチプラットフォームに対応することができない。

また、OpenGL に基いて設計された C++ のクラスライブラリである GLUT (OpenGL Utility Toolkit) を採用することにより開発コストを抑えることができる。

2.4 C 言語ベースのマルチプラットフォームが可能な GUI ツールキットの検討

C++ 言語で実装すれば異なる OS 上での動作は変わらない。しかし OS 毎に違う低レベルの GUI (Graphical User Interface) ツールキットがあり、それぞれの GUI ツールキットに対応したプログラムを書くのは開発コストが高い。そこで単一のソースコードでマルチプラットフォームを実現するため、C 言語系に基づく以下 5 種類の高次 GUI ツールキットについて比較検討した。

- ① Ultimate++
- ② Gtk+
- ③ Qt
- ④ OpenFrameworks
- ⑤ wxWidgets

また、各項目について検討した結果 wxWidgets[5]を採用することにした。検討結果を表 1 に示す。

表 1 高レベル GUI ツールキットの検討結果

	Ultimate++	GTK+	Qt	OF	wxWidgets
GUI	独自	Unix	native	独自	native
look and feel	△	×	○	△	○
OpenGL	○	×	○	○	○
開発元の更新	最近	2011年7月	最近	最近	最近
License	BSD license	LGPL	GPL	MIT License	wxWindows License
freeware	○	○	△	○	○

2.4.1 GUI デザイン

Qt と wxWidgets のみがネイティブな表示が可能である。各 OS 上でアプリケーションをビルドすると各 OS の低レベル GUI ツールキットで表示される。これにより、ユーザは普段使い慣れた GUI デザインで操作できるため、使いやすさ・ルック・アンド・フィールが良いと言える。

2.4.2 OpenGL のサポート

GTK+ は OpenGL のサポートをしていない。3D 音色表示機能の実装に OpenGL を用いるため、高レベル GUI ツールキット上で OpenGL が動作しなければならない。

2.4.3 ライセンス

wxWidgets の採用している wxWindows License は LGPL 改良ライセンスで、開発したアプリケーションのソースコードの開示義務がなく、商用利用することも可能である。

2.5 wxWidgets の特徴

wxWidgets の主な特徴として以下の 4 点を挙げる。

- (1) 移植性が高い
- (2) ネイティブな低レベル GUI を使って表示可能
- (3) 様々な分野の商用アプリケーションの開発実績
- (4) マルチプラットフォーム対応

単一のソースコードを書けば、OS 毎のビルドにおいて、wxWidgets が自動的に各 OS の低レベル GUI ツールキットを呼び出して表示するため、移植性が高い。

また、wxWidgets を使用した著名なアプリケーションとして“Audacity”[6]がある。

2.6 3DCG について

wxWidgets には wxGLCanvas という OpenGL のラッパークラスが含まれている。これにより OpenGL の描画が wxWidgets で再現できる。また GLUT は OpenGL に準拠したライブラリであるので wxWidget とも親和性が高い。

3. 新しい電子音色辞書の構成

上記の検討結果を取り入れ、新しい電子音色辞書を Windows で実装した。開発言語を C++ とすることにより、処理速度向上、マルチプラットフォームに対応、各サブシステムとの強力な連携を実現した。実装した電子音色辞書の構成を図 1 に示す。

ブラウザ上には MyTimbreList 機能があり、ユーザが制作上有機的にグルーピングしたい音データ群を格納することができる。また、音色の 3D 表示は OpenGL で構築し、ブラウザ上から直接的に表示させることが可能である。この表示は、ユーザが検索の前段階で様々な音色を探索することを狙いとしている。

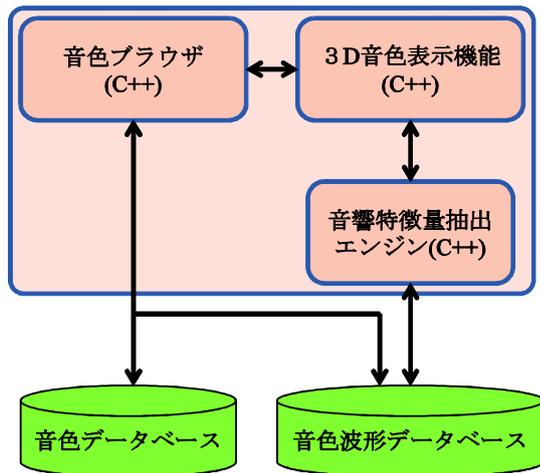


図 1 新しい電子音色辞書の構成

3.1 実装

上述の検討結果をもとに実装した。実装環境の OS は Windows, 開発は Microsoft Visual Studio 2010 Professional で行った。音色ブラウザ機能での変更が、リアルタイムで 3D 音色表示機能に反映されるなど、各サブシステム間の連動を実装することができた。

以下に図 2 から図 4 にそれぞれ音色ブラウザ機能, 3D 音色表示機能, 音色登録・アップロード機能の実装画面を示す。

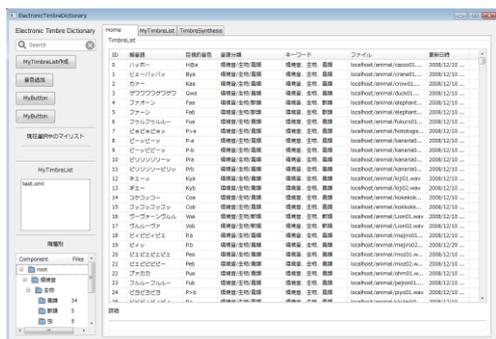


図 2 音色ブラウザ機能

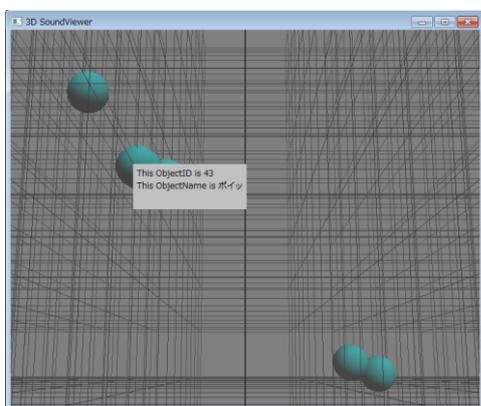


図 3 3D 音色表示機能

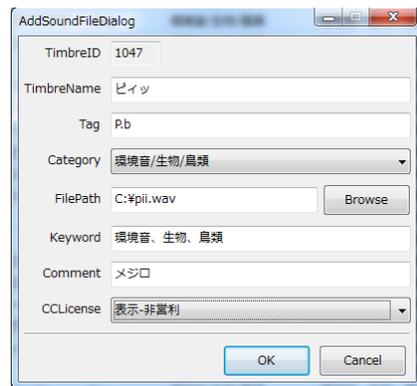


図 4 音色登録・アップロード画面

3.2 音色の検索

旧システムでは Apache Solr[7]で音色検索を行っていたが、本システムでは wxWidgets の find でキーワードや擬音語などのテキストを入力することにより簡易な検索を行う。音データは XML 形式で管理され、OS 上のファイル構造とは別の、音の生成に基づいた階層的な分類体系を持つ。これは、音色ブラウザ上に表示され、ここからの検索を順引きという。また、これ以外に擬音語からの逆引検索や、キーワード、などの音色のタグに基づいた検索も行う。

4. ユーザビリティ評価実験

システムのユーザビリティを評価するため、効果音を検索する場面を 2 種類準備した。発音体とその擬音語を表したものの 13 条件と、これにさらに商業の絵コンテを付して音の出る文脈を与えたものの 5 条件を設定した。これらのテキストを表 2 に示す。

表 2 音検索の条件

条件 No	音色の指示
1	時報の音、ポーン
2	飛行機の音、ゴー
3	ウィンドチャイム、シャラララン
4	波の音、ザザーン
5	鳥、カッコウ
6	缶を空ける音、カシッ
7	シンバル、ジャーン
8	カスタネット、カッカカッカ
9	楽器、ジュアーン
10	jewstrap、琴、ポインーン
11	炭酸の音、シュワシュワシュワ
12	バスドラム、ドン
13	ジュズハーブ、ビョイイーン
14	カッコウ「カッコウ」
15	ヒバリ「ピーチクパーチク」
16	ウシ「ンモー」
17	水がとっくんととっくん注がれる
18	カウベル「カランカラン」

4.1 評価実験方法

被験者は、電子音色辞書の以下のシステムに具備された3種類の検索機能から任意の機能を選んで、音色を検索する。

- (1) 階層・分類からの順引き検索
- (2) 擬音語からの逆引き検索
- (3) キーワード検索

以上の条件に対し、前システムと本システムの2つのシステムに対し評価実験を行った。被験者は前システムの評価実験では7人、本システムの評価実験では18人である。

4.2 評価方法

表2の全18音色条件それぞれに対して、以下の2つを評価した。

- (1) 検索機能の難易度を5段階評価で評価し、平均オピニオン評点(MOS: Mean Opinion Score)を求めた。これをMOS(困難性)と記す。
- (2) 物理評価として、各条件の検索時間(秒)を計測した。一つの音の検索に2分以上の時間を要した場合は検索に失敗したとみなした。検索を成功した率をタスク達成率と呼ぶ。

また、18条件全体に対して、以下の2つを評価した。

- (1) 満足度を0(非常に悪い)から4(非常に良い)までの5段階で評価し、MOSを算出した。
- (2) システムユーザビリティスケール(SUS: System Usability Scale)の評価も行った。

4.2.1 システムユーザビリティスケール

システムユーザビリティスケールでは、以下の指標に対する答えを、ポジティブ評価である奇数番目と、ネガティブ評価である偶数番目の評価値の符号を反転し、それらの総和を0点から100点に正規化したものである。

- (1) このシステムをしばしば使いたいと思う
- (2) このシステムは不必要なほど複雑であると感じた
- (3) このシステムは容易に使えと思った
- (4) このシステムを使うのに技術専門家のサポートが必要とするかもしれない
- (5) このシステムにあるさまざまな機能がよくまとまっていると感じた
- (6) このシステムでは、一貫性のないところが多くあったとおもった
- (7) たいいていのユーザは、このシステムの仕様方法について、素早く学べるだろう
- (8) このシステムはとても扱いにくいと思った
- (9) このシステムを使うのに自信があると感じた
- (10) このシステムを使い始める前に多くのことを学ぶ必要があった

4.3 実験結果

4.3.1 物理評価結果

以下に前システムと本システムにおけるタスク達成率の

比較表を表3に示す。

表3 タスク達成率比較表

	全検索回数	検索失敗回数	タスク達成率
前システム	126回	24回	81.0%
本システム	324回	12回	96.3%

前システムに対する評価では、検索に時間がかかり過ぎて検索に失敗したケースが多く見られた。本システムでは、多くの場合が2分以内に音を見つけることができていた。

以下に前システムと本システムの検索機能別の条件毎の平均所要時間の比較図を図5から図7に示し、図8に全検索機能の平均所要時間の比較図を示す。

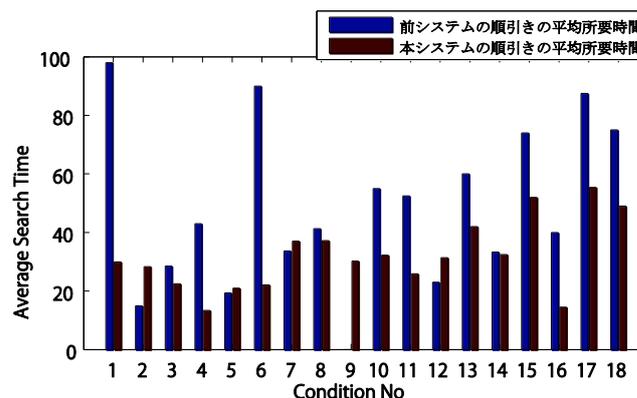


図5 順引き検索の条件毎の平均所要時間の比較

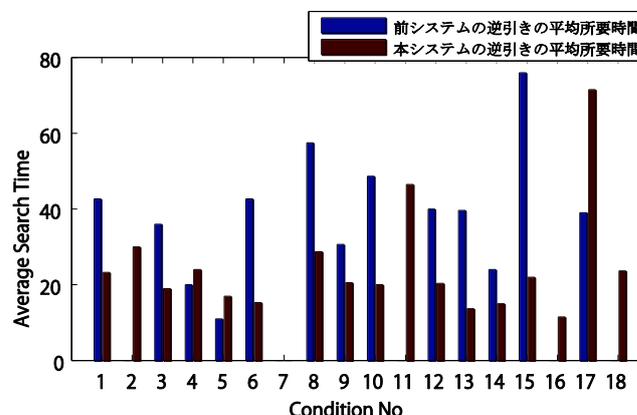


図6 逆引き検索の条件毎の平均所要時間の比較

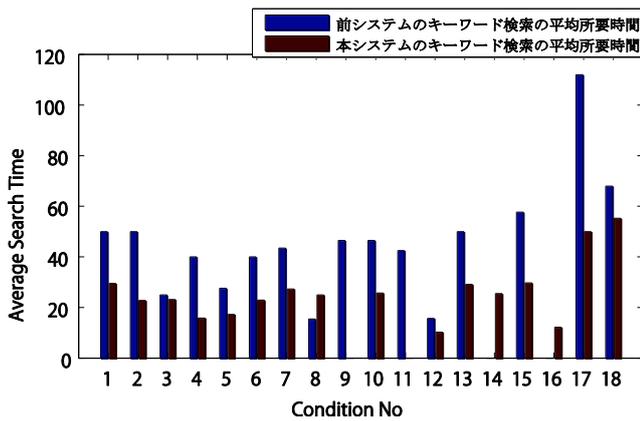


図7 キーワード検索の条件毎の平均所要時間の比較

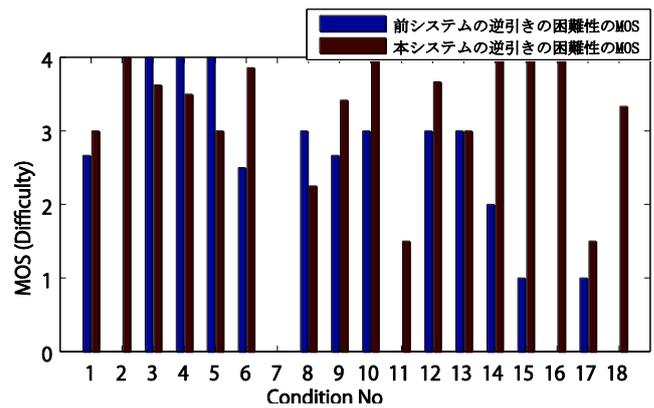


図10 逆引き検索の条件毎の MOS (困難性)

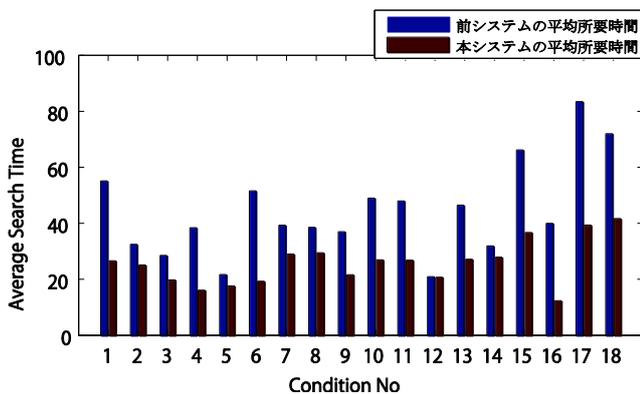


図8 全検索機能の条件毎の平均所要時間の比較

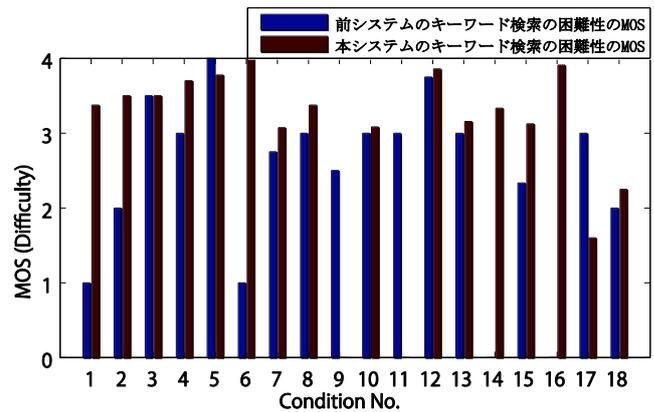


図11 キーワード検索の条件毎の MOS (困難性)

図9から図11に、前システムと本システムに対し、検索機能別に条件毎の MOS (困難性) を比較して示す。また、図12に全検索機能の MOS (困難性) を比較して示す。

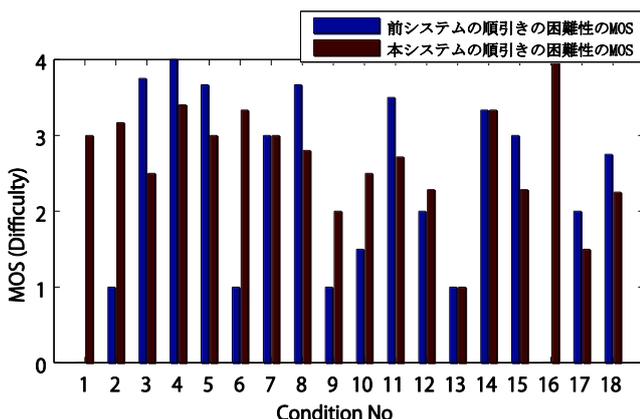


図9 順引き検索の条件毎の MOS (困難性)

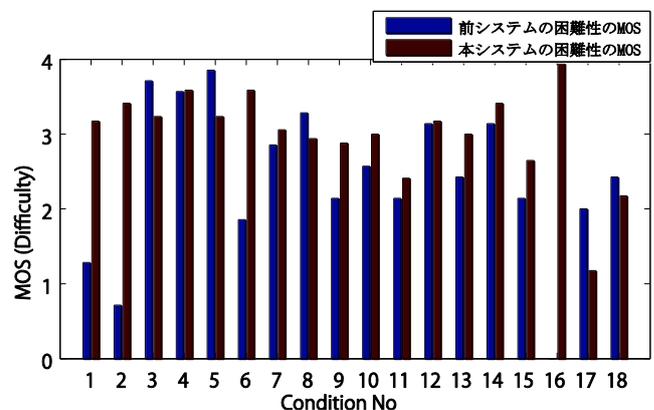


図12 全検索機能の条件毎の MOS (困難性)

全体の平均評価と平均所要時間が前回と比べて改善された。しかし、検索機能別に着目すると、平均所要時間では、音色によっては使用した検索方法により時間がかかり、前システムのデータよりも結果が悪くなったケース (図6条件17) が見られた。

4.3.2 物理評価と主観評価の比較

図13と図14に、前システムと本システムにおける物理評価としての検索時間と各18条件の MOS (困難性) との相関を示す。

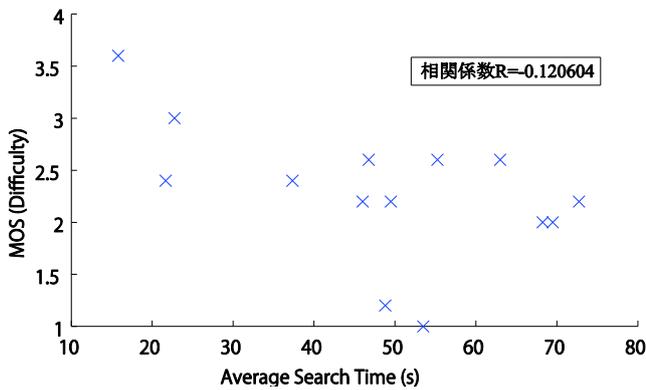


図 13 前システムの物理評価と主観評価の相関

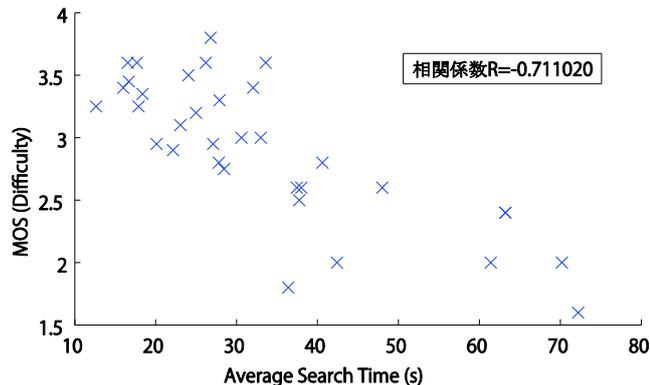


図 14 本システムの物理評価と主観評価の相関

両図より、前システムでは相関が低かったが、本システムでは両評価値の負の相関が高い。これは、前システムでは検索時間の中で GUI 操作に占める割合が多かったのに対し、本システムでは、GUI の効率化により、物理評価の検索時間が実質的な検索に費やされているためと考えられる。

4.3.3 MOS (困難性)

全 18 条件の各難易度の MOS の平均値を求めた。前システムの平均値 2.2 に対し、本システムの平均値は 2.9 となり、評価が向上した。

4.3.4 MOS とシステムユーザビリティスケールの評価

本システムの MOS を評価した。MOS は 3.0 であった。また、本システムではシステムユーザビリティスケールスコアは 79.8 点となり、比較的満足度の高い評価が得られた。

5. 結果の考察

実験結果から以下の 3 点が考えられる。

1. タスク達成率と平均評価と平均所要時間の比較から、提案するシステムの基本的な有効性が確認できた。しかし、音色の数パーセント程は、選定する検索機能で所要時間が多くかかり、ユーザは効率の良い検索の学習が必要であると考えられる。
2. 前システムの評価実験から所要時間が改善した理由が 2 点考えられる。1 つは、wxWidgets の採用により、OS 毎の低次の GUI ツールキットで表示できるため、ユーザは普段使い慣れたネイティブな表示で操作できた

という点である。もう 1 つは、GUI の見直しにより、前システムと比べ操作がシンプルになった点である。

3. 満足度とシステムユーザビリティスケールのスコアから、使いやすいとする評価が得られた。

6. おわりに

電子音色辞書について、C++によるサブシステムの再構築、統合を行った。提案システムでは開発言語を統一したため、システム内部の処理速度が向上した。

また、wxWidgets の採用により、単一のソースコードでマルチプラットフォームに対応することができた。3 種類の検索機能を用いて音色を検索するシステム評価実験を行った結果、タスク達成率 96.3%、困難性を表す MOS2.9 と、前システムより向上し、システムユーザビリティスケールスコア 79.8 点となり、音色を容易に検索できるシステムであることが示された。

今後は、さらに様々なサブシステムの実装、GUI をさらに見直してユーザビリティの向上、Linux や MacOS でビルドを行い、システムの本格リリースに向けて進めていく予定である。

参考文献

- [1] 斎藤 佳紀, 恩田 大河, 根本 翔多, 小坂 直敏, “電子音色辞書のデスクトップアプリケーション化および携帯端末アプリケーションの実装”, 情報処理学会 インタラクシオン 2011 2CR3-11, 2011.
- [2] 吉岡靖博, 石塚慎也, 斎藤佳紀, 高橋奈緒子, 小坂直敏, “電子音色辞書における音色探索システムおよび 3 次元音色表示インタフェース”, 情報処理学会音楽情報科学研究会, MUS-79-11, 2009.
- [3] Creative Commons Japan
<http://creativecommons.jp/licenses/>
- [4] OpenGL - The Industry Standard for High Performance Graphics
<http://www.opengl.org/>
- [5] wxWidgets Cross-Platform GUI Library
<http://www.wxwidgets.org/>
- [6] Audacity: Free Audio Editor and Recorder
<http://audacity.sourceforge.net/>
- [7] Apache Lucene - Apache Solr
<http://lucene.apache.org/solr/>