

ネイティブ英語発話の日本人風の発音への変換による国際的な意識の促進

西田 健志*

概要. 日本人は英語での読み書き能力の高さに比べて英語での会話を苦手にする傾向があると言われているが、その大きな障壁の一つにはリスニングやスピーキングにおける発音の問題が挙げられる。相手が話していることが聞き取れないことが会話を練習する大きな障害になるのはもちろんのこと、後天的にネイティブと同等の発音が身に付くことが稀であるにも関わらずネイティブの発音を模倣する練習を繰り返させることが自分の発音への自信を失わせ、英語で話すことを恥ずかしいと感じさせてしまっている面がある。一方で、英語が国際的な場でのコミュニケーションのための言語として位置づけられ、世界中で様々な訛りを持つ英語を話す人が増える中で、自分たちと同じような発音しか受け入れず、訛った英語を話す人を低く見がちなネイティブの態度にも改められるべきところがあると考え、そこで我々は、ネイティブ英語発話を日本人風の親しみやすい発音に変換することで、日本人の英語に対する心理的な障壁を緩和するとともに、ネイティブに対しては自分の発話よりも訛った発話の方が日本人には通じやすいという体験を通じて意識の改革を促す手法を提案する。本稿では、国際コミュニケーションを支援する様々な他の方法との関係性を議論するとともに、音声認識と TTS を組み合わせて発音の変換を実装した実験用アプリケーションとそれを利用してもらったときの日・米・中の出身者の感想を報告する。

1 はじめに

高校・大学入試での合格を最優先の目標とする英語教育の結果、文法理解や読解・英作文といった読み書きの能力に比べて、日本人は英会話を苦手に行っているとされている。その反省を踏まえ、ネイティブの英語教師の採用増や小学校からの英会話授業の導入など、会話能力の獲得を中心とした英語教育の改善が議論されている。

英語での会話能力の獲得にあたって特に障害になりやすいのはリスニングや発話における発音の問題である。まず、語彙や文法の知識があつたとしても相手が話していることを聞き取ることができなければ、自分が言いたいことを英語で話す練習をするスタートラインにつくことすら困難である。また、発音についてはネイティブの発音を模倣する練習を重ねるのが通例となっているが、後天的にネイティブと同等の発音に達するのは例外的な少数の者だけであり、そのような練習を当然のように課すのは自信を失わせて学習意欲を削ぐばかりなので改められるべきだという声もある [2]。

英語をあくまで国際的なコミュニケーションの道具として捉える「国際語としての英語 (English as an International Language)」の動きやインド英語やシンガポール英語など世界各地でそれぞれの訛りを持った「世界英語 (World English)」を話す人口が増えている現状を鑑みるに、英米などのネイティブからも、相手が聞き取りやすいように話す、相手

の訛った英語を聞き取れるように努力するといった歩み寄りの精神が求められるのは当然のことと思われる。

発音に寛容で歩み寄りを心がける人は少なくないが、ネイティブらしくない英語に対する忌避的な反応にも根深いものがある。たとえば、ネイティブ発話と同程度に聞き取れる発話であっても訛りがあると聞き取りにくいと評価される傾向 [1] や、訛った英語で話す人が様々な差別を受けていること [6] が知られている。非ネイティブも発音の悪さを認識することだけはネイティブ並みにでき、訛った英語やその話者をネイティブと同様に低く評価する傾向があるが、自分と近い発音については比較的寛容であることが知られている [9]。

本研究では、日本人英語話者とネイティブ英語話者の意識ギャップを埋めるためにネイティブの英語発話を日本人風の訛った英語発話に変換するシステムを利用して、英会話の練習および実践を行うことを提案する。日本人にとっては日本訛りの英語が聞き取りやすい [2] のでリスニングが楽になる分、言いたいことを英語で言うことにより集中することができる可能性がある。また、ネイティブにとっては、訛った発話の聞き取り練習に利用できるだけでなく、自分の元々の発音よりも訛った英語の方が日本人相手に通じやすいという、先入観が覆される体験を通じて、様々な発音の世界英語に対する寛容性を高めることが期待される。

本稿ではまず、提案手法の応用可能性について議論し、続いて様々な実験ができるように実装したアプリケーションについて、利用した日本人、英語ネ

Copyright is held by the author(s).

* 神戸大学大学院国際文化学研究所

イティブ、および中国人の声とともに報告する。さらに広範にわたる関連研究を提案手法との関係性・利害得失・応用場面の違いの観点から整理し、最後に今後の課題をまとめる。

2 考えられる応用例

本稿で提案するネイティブの発話を日本人風の訛った発話に変換する手法には、日本人には聞き取りやすくなることによる効果およびネイティブには聞き取りにくくなることによる効果を活用した様々な応用が考えられる。

2.1 ネイティブと会話するための補助

まず、提案手法はリスニングがあまり得意ではない日本人が英語ネイティブと会話する際の補助として利用することができる。発音が日本人風に変換されるのを聞くことで、訛った発音で話す気恥ずかしさも軽減されるだろう。具体的には、音声翻訳するスマートフォンアプリのようにボタンを押してから話すと話し終わった後に変換される利用形態や、補聴器のように自動的に変換を行う利用形態などが想定される。後者のような利用形態を実現するためには発話部分を精度よく自動的に切り出しリアルタイムに変換できる必要がある。

自転車の補助輪のように、いずれは補助が無くても会話ができるようになりたい人がまずはネイティブと会話できる楽しさを味わいたいという場合にも利用できるだろう。そのような利用のためには、変換の程度を段階的に減らしていくことができる機能など、補助からの卒業しやすさも考慮することが望ましいと思われる。

2.2 ネイティブに対するカルチャーショック体験

様々な訛りを持った「世界英語」を話す人口が増え、TOEICなどの英語能力試験でも様々な発音が出題されるようになる中にも訛った発音で話す人を差別してしまうような人に対して、「発音で人を判断するのは良くない」、「訛った英語を聞き取れるように努力すべきだ」などと言葉で意識改革を求めるのには限界がある。

海外へ旅行や留学に行った際に言葉が相手に通じないという体験をすることは国際的な意識改革を促しやすいと思われるが、英語が国際語としての地位を確立しつつあることが影響して、英語ネイティブの中にはそのようなときにも「英語も通じないのか」と責任を相手に求めてしまう人もあるだろう。

英語ネイティブが日本人と話すときに、最初は自分の発音そのまま発話し、その後それを日本人風の発音に変換した音声をお互いに聞こえるように再生することで「元々の発音よりも訛った発音の方が日本人相手には通じやすかった」という体験ができるものと思われる。この体験を通じて、ネイティブ

の側からも相手に歩み寄るべき余地があることが実感され、様々な発音の世界英語に対する姿勢が改められることが期待される。

もちろん、提案手法はそのような意識改革がなされた後に訛った英語の聞き取りを練習するためにも利用することができる。ネイティブが非ネイティブの発話を聞き取れるようになるという目標は、非ネイティブにネイティブの発音を期待するよりは現実味があると言えるだろう。

2.3 会議での参加格差の是正

リスニング能力が比較的高い日本人にとっても提案手法が有益と思われる状況としてはネイティブとの会議が挙げられる。たとえばオンラインの音声会議においてネイティブの発言を日本人風の発音に変換するとネイティブだけが聞き取りに苦労するようになるため、日本人が議論のペースについていやすくなり、発言もしやすくなる可能性がある。ただし、このような応用を実現するためにはネイティブからの歩み寄りや平等な参加に対する大きな配慮が必要となるので、まずは日本人が聞く音声だけを日本人風に変換するというセットアップも考えられる。また、逆に日本人の発話をネイティブ風に変換する手法を組み合わせることで格差をさらに小さくすることも期待できる。これらの応用を実現するためには発音の変換がリアルタイムに行えることが望ましい。

3 実験用アプリケーション

本章では、提案手法の効果を検証する様々な実験を行えるように実装した実験用のアプリケーションについて述べる。

現時点では一区切り話した後に変換した音声を再生するようになっており、2章で議論した応用例のうち、2.1節で述べた会話中の補助および2.2節で述べたカルチャーショック体験など少人数での利用に対応しているが、2.3節で述べた会議などの多人数での応用には対応していない。

3.1 発音の変換方法

本アプリケーションでは音声認識プログラムとテキスト読み上げプログラム (text-to-speech; TTS) を用いることで日本人風の発音への変換を行う二つの方法 (図 1, 2) を実装した。どちらの方法を用いても普通のノート PC を用いた場合、話し終えた直後に変換後の音声再生される。音声認識には Windows に英語の言語パックをインストールすることで利用できるもの、TTSには Microsoft Speech Platform に含まれるものをそれぞれ利用した。

一つ目の方法は、ネイティブの発話を音声認識して得られた英語テキストを日本語 TTS に読み上げさせることで日本人風の発音に変換する方法である (図 1)。この方法は実装が非常に容易であるが、上

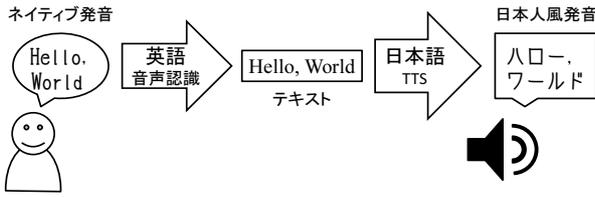


図 1. 変換方法 1: 英語音声認識の出力する英語テキストを日本語 TTS に読み上げさせる方法

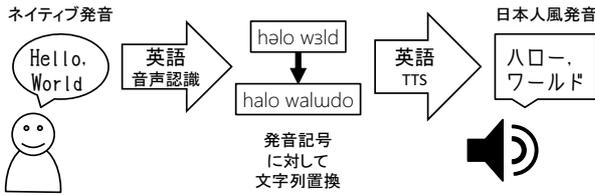


図 2. 変換方法 2: 音声認識の出力する発音記号を文字列置換で変換したものを読み上げさせる方法

述の日本語 TTS を利用することで日本人が各単語をバラバラにカタカナ読みしているような発音に変換することができる。またこの方法では、他言語の TTS を用いることで日本人風以外の様々な訛りの発音に変換できると思われる。

しかしこの方法は英語を日本語 TTS で読み上げる段階で利用している辞書の影響か冠詞の "a" を「エイ」、"who" を「ダブリューヘイチオー」と読むなど、望ましくない変換をしてしまう場合がある。音声認識でも認識率向上のために辞書が利用されるため同様の問題が発生しうる。

そこで、二つ目の方法として、音声認識プログラムの出力する発音記号の文字列を文字列置換 (表 1) によって変換し、変換した発音記号を TTS に指定して読み上げさせる方法を用意した (図 2)。この方法では TTS の辞書に関わらず発音を変換することが可能である。また、音声認識の辞書にない単語や誤認識された語についても発音レベルで大きく外すことは少ないため、一つ目の方法よりは安定して発音を変換できると思われる。

こちらの方法のもう 1 つのメリットとしては、文字列置換のルールを変えることで変換の種類や程度を調節できることが挙げられる。また、日本語 TTS ではなく英語など他の TTS を利用しても日本人風の発音に変換することができるので、英語ネイティブが日本人風の発音で話しているような音声などを得ることができる。その代わりに日本人風以外の発音に変換したい場合には、それぞれの訛りごとに変換ルールを用意する必要がある。

どちらの方法も TTS を利用しているので話者とは違う声になってしまう欠点がある。

表 1. 発音記号の変換ルール例。発音記号の表記には国際発音記号 (IPA) を使用する。

母音			子音		
変換前	変換後	例	変換前	変換後	例
[ɑʌɔ]	a	<u>s</u> un	f	ɸ	<u>f</u> at
ɪ	i:	f <u>i</u> t	v	b	<u>v</u> est
[uʊ]	u	<u>f</u> oot	θ	s	<u>th</u> ink
ɛ	e	<u>b</u> ed	ð	z	<u>th</u> e
ɔ	o	<u>d</u> og	tʃ	tʃ	<u>ch</u> eeper
			r	l	<u>r</u> ight
			m	w	<u>w</u> hen

子音 + 母音			子音の連続と末子音 (母音を後ろに補う)		
変換前	変換後	例	子音	補う母音	例
ɪr, ɛr, ʊr	ɪa, ɛa, ʊa	<u>ear</u>	dt	o	<u>vest</u>
s[i:]	ʃi	<u>see</u>	bfgklmpsvz	u	<u>vest</u>
t[i:]	tʃi	<u>tip</u>			
t[uwɔ]	tʃu	<u>too</u>			

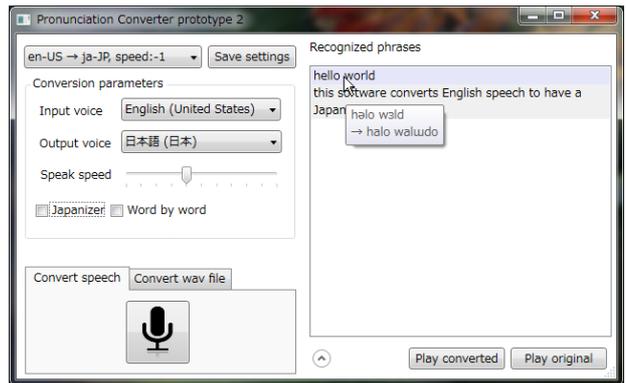


図 3. 実験用アプリケーションのスクリーンショット

3.2 実験用 UI

図 3 は、前節の変換方法を用いて様々な形態の実験ができるように作成した Windows アプリケーションのスクリーンショットである。ウィンドウには、変換パラメータの設定 UI (左上)、マイク入力か音声ファイルを指定する UI (左下)、音声認識結果の履歴 (右) がそれぞれ表示される。

3.2.1 変換パラメータの設定

設定できる変換パラメータには、使用する音声認識と TTS のエンジン、発音変換方法、そして話速度がある。

音声認識と TTS のエンジン 使用する音声認識エンジンを変更することで英語以外の言語を日本人風に変換する実験、TTS エンジンを変更することで様々な発音に変換する実験を行うことができる。

発音変換方法 チェックボックス "Japanizer" にチェックを入れると前章の二つの発音変換方法のうち、発音記号の文字列置換を行う方法が使用される。

話速度 話速度の影響を調べることができるよう、全体的話速度を調節するスライダーと単語間を長くするチェックボックス "Word by word" の二種類の設定を用意した。

3.2.2 入力と出力

本アプリケーションはマイク入力によるその場での変換と録音済み wav ファイルの変換に対応している。マイク入力の場合は、マイクボタンを押してから発話が途切れるまでを1単位として認識し、それを変換した結果が直後に再生される。ファイル入力の場合には全体がまとめて認識、再生される。どちらの方法を利用した場合にも、変換前後の音声と認識結果をファイルに保存することができる。

3.2.3 音声認識結果の履歴

右側には音声認識結果の履歴が表示されており、カーソルを各結果の上に持つて行くことで変換前後の発音記号を確認することができ、クリックすることで発音変換後の音声を再び聞くことができるようになっている。これにより、変換パラメータを変更しながら様々な発音を聞き比べる実験や、音声認識結果や発音記号を相手に見せながら会話するなど他の支援方法と比較する実験が可能である。

3.3 利用者の反応

本節では、実験用アプリケーションを日本人大学生 4 人、日本の大学で英会話の授業を担当しているネイティブ英語教員 1 人、日本の大学への中国人留学生 3 人に簡単に利用してもらった反応をまとめる。

発音変換の質 日本語 TTS をそのまま利用する一つ目の方法で変換した音声については紛れもなく日本人の話す英語らしく聞こえるが、発音記号を変換する二つ目の方法で変換した音声については、日本人らしく聞こえるもの一つ目の方法と比べるとやや聞き取りづらいことがあるとの反応で一致した。

応用可能性 ネイティブ英語教員から、自身の担当している英会話の授業で利用してみたいとの意見が得られた。自身の経験上、日本人訛りの英語が聞き取りづらいと感じるのは最初だけで、少し慣れれば聞き取りやすい部類に入っていると思うが、その割に英会話の授業などで恥ずかしがって話したがない学生が多いので、そのようなときにこれを利用すると話しやすくなりそうだとの意見をいただいた。

他の訛りへの変換 中国語 TTS を利用して変換した英語音声を聞いてもらったところ、ネイティブ英語教員と中国人留学生のうち 1 人が、中国人の話す英語らしく聞こえると答えたが、日本人風の発音よりは微妙であったためか具体的にそのように感じたポイントを指摘した。その同じ音声について日本人大学生からは、かなりネイティブに近く感じられるとの一致した感想が得られた。

4 関連研究

外国語でのコミュニケーションを支援する方法の研究としては、機械翻訳や通訳者を利用する方法、音声の変換を利用する方法に加え、その他様々なインタラクティブシステムによる支援方法がある。本章では提案手法との関係性、利害得失、および応用場面の違いについて議論する。

4.1 機械翻訳を利用する方法

互いに使い慣れた言語を共有しない者どうしが意思疎通を図るときにもっとも手軽に利用できる方法のひとつが機械翻訳である。機械翻訳機能を搭載したテキストチャットに関する研究 [12, 11] に加え、最近では Google 翻訳 [5] や docomo はなして翻訳 [3] といった音声を入力すると翻訳後の音声を出力するスマートフォンアプリも登場している。しかし、機械翻訳の精度が向上しているといってもしばしば誤った翻訳をするという問題があり、特にチャットでの砕けた表現や話し言葉に対応することは現状でも容易ではない。

翻訳精度が高い場合にも、ある言語から別の言語へと翻訳された表現とそれを元の言語へ再翻訳した表現とが一致しないせいで意思疎通が難しくなることが指摘されている [12]。たとえば、相手の使った表現を引用して返事をしたときに元々相手が使っていた表現とは異なる表現に訳されたり、一度お互いに通じると確認しあった表現であっても後にそれを短縮した表現を使ったときに異なる表現に訳されたりするせいで、コミュニケーションの基礎となる表現が安定しないことが問題となる。

機械翻訳を用いる際の問題点を克服するために機械翻訳の使い方を工夫する試みもなされてきた。まず、再翻訳した表現 (back-translation) を表示しながらコミュニケーションを行うとコミュニケーションが円滑になることが実験により明らかにされている [11]。五十嵐は、機械翻訳が失敗しがちな長く複雑な係り受けがある文章を翻訳する際に、文節の区切りや係り受け関係を専用の UI を使って入力し、区切られた文節を別々に機械翻訳して視覚化することで伝わりやすくする手法を提案している [13]。

これらの研究に対して本研究は、翻訳を介するのではない新たな歩み寄りの形を提案するものである。機械翻訳を利用すると相手の話す言語についての知

識がほとんどなくてもコミュニケーションを取ることができるので便利であるが、いずれは補助を卒業したいと考えている人にとっては提案手法の方が向いているだろう。さらに、提案手法には、ネイティブにカルチャーショック体験を与えて意識改革を促すという機械翻訳では実現できない応用がある。

技術的には、機械翻訳に付き物の翻訳ミスに比べると、発音変換の方が想定外の動作による影響を小さくできる可能性がある。特に音声入力の結果を翻訳する場合には、音声入力での認識エラーと翻訳エラーの2種類のエラーが重なって発生することになるので両者の認識精度が極めて高い必要がある。

4.2 人間の通訳を利用する方法

機械翻訳ではなく人間の通訳を利用することで、くだけた表現やその場面の文脈などへのより柔軟な対応が可能になるが、専門の通訳を用意するには高い人的コストがかかり、その場にいる人が対応するにしても高い負担がかかってしまうという問題がある。通訳にかかる負担への対処に関する研究としては、多言語対面会議において少数派言語の参加者を他の参加者が協同で支援することで一人一人の支援負担を軽減する all-for-one 型支援システムが挙げられる [14, 15, 16]。

それに対して本研究で提案するのは、話しているネイティブ自身による歩み寄りや配慮に期待する方法である。

4.3 発音の変換を利用する方法

発話の音響的な性質を変換する研究分野で主流となっているのは、声質、つまり誰の声と認識されるかを変換するものであり、それと比べて発話の発音を変換する研究は多くはない [4]。またそのいずれも、訛りを取り除く、適切な抑揚を付加するなどして、ネイティブらしい発話に近づけることを目的としたものである [4]。

訛りを取り除く方法はネイティブに話が通じやすくなるので便利であるが、時代の趨勢に逆行して「美しい発音」への憧れをさらに強くしてしまう恐れがある。それを防ぐためにはあくまで「相手に一番通じやすい発音」への変換を共通の技術目標として捉え、ネイティブ風への変換と日本人風への変換の両方の研究を進めていくことが適切であろう。

もちろん、ネイティブと非ネイティブがお互いに逆方向の変換を使うという形で組み合わせることもでき、そうした方が非ネイティブの訛りを除去するだけの場合よりもより公平な環境になると考えられる。

技術的には、ネイティブらしい発音の英語に対する音声認識の精度がかなり高くなっているのに対して、非ネイティブの発話を認識するためには様々な訛りの認識に対応するのが難しい [8] のと同様に、発音の変換についてもネイティブ発音を認識して日

本人風に変換する方が容易である可能性が高いと思われる。

4.4 その他のインタラクティブシステムによる支援

単に発話速度が遅くなるように変換したり、音声認識結果を画面に表示するだけでも、リスニングが苦手な人にとっては大きな支援になるものと考えられる。音声訂正 [7] のように音声認識結果を手作業で修正する手法を組み合わせれば、4.1 節で述べた認識エラーと翻訳エラーが重なる問題も大きく解消される。

しかし、これらの方法はどれも利用中の「自分たちは英会話ができない」というネガティブな印象が前面に出てしまいがちなのに対して、提案手法を利用することで「自分たちは訛っているけど英会話ができる」というポジティブさを前面に出すことができるのではないかと考えている。

多人数環境における非ネイティブの支援に着目した研究としては、少人数状況では英語で会話できる非ネイティブであっても会議では議論の速さについていくことが難しく発言が少なくなりがちであることを改善しようとした研究がある [10]。その研究では、オンライン音声会議でネイティブ同士だけ音声伝達を 0.2-0.4 秒程度遅らせることで非ネイティブに時間的な余裕を与える手法が提案・評価され、確かに非ネイティブが議論に参加しやすくなる効果は認められるものの、発話の開始が重なりやすくなって進行が妨げられるという改善すべき点も確認されている。本研究もこの研究と同様に、ネイティブの側から非ネイティブに配慮することを促すことを重視しているが、より意識されやすい形を採ることで歩み寄りへの意識改革を促すことも狙っている。

5 まとめと今後の課題

本稿では英語ネイティブの発話を日本人風の発音に変換して日本人にとって聞き取りやすくする手法を提案し、日本人のリスニング支援や英語ネイティブへのカルチャーショック体験などの応用可能性を議論した。さらに音声認識と TTS を利用して実装した実験用アプリケーションを異なる母国語を持つ人たちに利用してもらい、初期的なフィードバックを得た。

今後の課題としてはまず、日本人風変換を利用した英語発話の聞き取りおよび会話に対する効果について、より多くのネイティブと日本人が参加する実験により検証することが挙げられる。また、さらなる安定性、話者の声質保持、およびリアルタイムな変換などを実現するため、音声認識や TTS を利用するのではなく、音声信号を直接処理することによって発音を変換する手法を模索する。それらの成果を基にして最終的には、ネイティブに対するカルチャーショック体験や会議での格差是正など、他の

応用例についてのシステムを開発し、実証実験を行う計画である。

参考文献

- [1] T. M. Derwing and M. J. Munro. Accent, intelligibility, and comprehensibility: evidence from four L1s. *Studies in second language acquisition*, 19(01):1–16, 1997.
- [2] T. M. Derwing and M. J. Munro. Second Language Accent and Pronunciation Teaching: A Research-Based Approach. *TESOL Quarterly*, 39(3):379–397, 2005.
- [3] N. docomo. はなして翻訳. http://www.nttdocomo.co.jp/service/communication/hanashite_honyaku.
- [4] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna. Foreign accent conversion in computer assisted pronunciation training. *Speech Communication*, 51(10):920–932, 2009.
- [5] Google. Google Translate. <https://play.google.com/store/apps/details?id=com.google.android.apps.translate>.
- [6] M. J. Munro. A Primer on Accent Discrimination in the Canadian Context. *TESL Canada Journal*, 20(2):38–51, 2003.
- [7] J. Ogata and M. Goto. Speech Repair: Quick Error Correction Just by Using Selection Operation for Speech Input Interface. In *Proc. Eurospeech '05*, pp. 133–136, 2005.
- [8] Y. Tsubota, T. Kawahara, and M. Dantsuji. Recognition and verification of English by Japanese students for computer-assisted language learning system. In *INTERSPEECH*, 2002.
- [9] P. Winke, S. Gass, and C. Myford. The Relationship Between Raters' Prior Language Study and the Evaluation of Foreign Language Speech Samples. Technical report, Educational Testing Service, 2011.
- [10] N. Yamashita, A. Echenique, T. Ishida, and A. Hautasaari. Lost in transmittance: how transmission lag enhances and deteriorates multilingual collaboration. In *Proc. CSCW '13*, pp. 923–934. ACM, 2013.
- [11] N. Yamashita, R. Inaba, H. Kuzuoka, and T. Ishida. Difficulties in establishing common ground in multiparty groups using machine translation. In *Proc. CHI '09*, pp. 679–688. ACM, 2009.
- [12] N. Yamashita and T. Ishida. Effects of machine translation on collaborative work. In *Proc. CSCW '06*, pp. 515–524. ACM, 2006.
- [13] 五十嵐 健夫. 文の構造を明示的に指定・表示することによる異言語間コミュニケーション. WISS 2011 予稿集.
- [14] 吉野 孝, 井出 美奈. All for one 型多言語会議支援システムの構築と評価. 情報処理学会論文誌, 51(1):36–44, 2010.
- [15] 宮部 真衣, 吉野 孝. All for one 型対面会議支援システムのためのワークスペースアウェアネスの効果. 電子情報通信学会論文誌. D, 情報・システム, 94(1):27–36, 2011.
- [16] 宮部 真衣, 吉野 孝. 多言語対面会議支援システムのための All for one 型支援の効果. 情報処理学会論文誌, 52(1):90–96, 2011.

未来ビジョン

我々日本の研究者も英語には散々苦い思いをさせられている。論文を読むにも書くにも何倍も時間をかけ、高い金を払って英文添削を受ける。国内の学会で発表された研究に類似した研究を国際学会の査読で見かけても指摘することができない。「英語で発表しなければ意味がない」と発破をかける声が日本人の中からも常に出てくるが、片方に意思疎通の努力責任をすべて押し付ける状況にこそ疑うべき点があるのではなかろうか。

奇しくもインタラクティブシステムの分野では、CHI などの国際会議で文化の相違 (cultural difference) を標榜するセッションが必ずあり、不十分・不安定なインフラや低識字率など様々な問題を抱えた非先進地域に焦点を当てた研究が目立ち始めるなど、他の文化を理解しようという風潮に注目が集まっている。

今こそ、まずは研究者コミュニティの中から、非先進地域も結構だがもっと身近な問題にも目を向けるべきだということ、そして言葉の問題に限らずコミュニケーションでは双方が互いに歩み寄るべきだということを示していくべきときである。

理解さえ得ることができればすぐにでも実現できるアイデアは多い。

- 日本の学会で発表された優れた研究を紹介するセッションを国際学会に設ける
- 採択決定後に一括で英文添削する
- 日本語で投稿と査読を行って、採択が決定してから英訳することを認める

そうしたことを実現していくためには、山下らの研究 [10] や本稿のように英語ネイティブからの歩み寄りを求める研究を積極的に発表し続けることが役に立つと期待している。

SmartVoice: 言語の壁を超えたプレゼンテーションサポーターティングシステム

李 翔 暦本 純一*

概要. 国際的イベントでは、共通言語での講演が求められることが多い。しかし、語学力を短期間に向上させることが難しく、共通言語が母国語ではない講演者にとって負担になる。また、同時通訳などの方法は、コストが高い上、講演者本人の発表を聞いているわけではないため、プレゼンテーションの自然さを損なってしまう。本論文では、言語の壁を超えてプレゼンテーションを可能にするシステム「SmartVoice」を提案する。SmartVoiceは、音声データ化した原稿を講演者の口の動きにあわせてスマートに送り出し、講演者が自分のタイミングで講演できるようにする。音声のイントネーションや強弱も、口の位置と形状に基づいて制御するため、まるで講演者本人が直接喋っているように見える。評価実験を行ない、本人のプレゼンテーションとSmartVoiceによるプレゼンテーションがほとんど区別できないことがわかった。SmartVoiceのアプリケーションは、多言語を用いるプレゼンテーションに限らず、映像に音声・セリフを入れるアフターレコーディングにも適用可能である。本論文では、他分野におけるSmartVoiceの応用についても議論する。

1 はじめに

国際学会などのイベントでは、共通言語、たとえば英語で講演することが求められる。しかし、共通言語が母国語ではない人にとって、それは大きな障壁となる。仮に原稿を書いて読み上げることにしたとしても、正しい発音で流暢に聞こえるためには多くの努力を要する。

言語が異なる講演のために、同時通訳が利用されることがある。しかし同時通訳者を雇うコストや講演者とは異なるタイミングで音声が出る不自然さなどの問題がある。また講演者の感情や非言語情報を伝達することが困難である。予めプレゼンテーション原稿の音読を録音し、プレゼンテーションの時にそれを再生して口パクする方法もあり得るが、講演者は録音のペースに合わせる講演をすることになり、講演の自由さが大きく制約されてしまう。

ボイスシンセサイザーを用いることで、原稿のテキストファイルさえあれば、コンピュータがどの言語の単語でも正確な発音で読み上げてくれる。しかし、文章を朗読する時のイントネーション、抑揚が単調であり、聞き疲れするため、人間の声の代用とすることはできない。例えば通常の講演では、大事なところで間を開けて聴衆の注意をひきつける、などが行われるが、ボイスシンセサイザーではこのような演出をすることは難しい。

本論文では、これらの問題を解決する手段として、「スマートな口パク」に基づくシステムを提案する。スマートな口パクとは、まず原稿をナレーターに読んでもらうか、ボイスシンセサイザーを利用するか

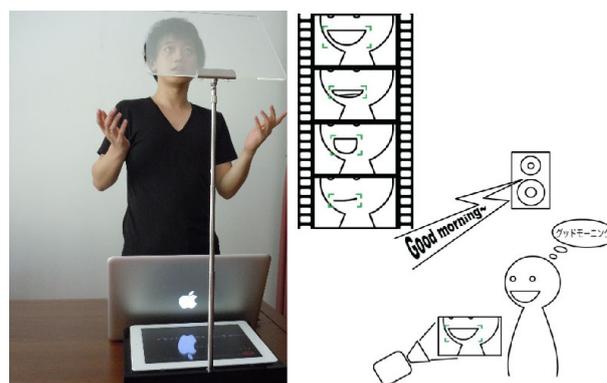


図 1. SmartVoice によるプレゼンテーション

して音声データにしておき、それを講演者の口の動きに合わせて送り出す方式である。講演者は自分のタイミングでプレゼンテーションすることができ、大事なところで間を空けるなど、語速を自由に制御できるようになる。また、イントネーションや音の強弱などを、口の位置や形状から制御するようにしている。この結果、聴衆からすると、講演者本人が直接喋っているように見え、同時通訳のような不自然さがない。この方式での講演は、共通言語でのプレゼンテーションに留まらず、聴衆に合わせた言語での講演（たとえば中国で中国語で講演する）にも利用出来る。この、スマートな口パクに基づくシステムをSmartVoiceと呼ぶことにする。

2 SmartVoice

SmartVoiceは、再生制御およびピッチ、スピード、音量のリアルタイム調整をプレゼンターの口の動きとその他表情にマッピングし、音声口パクのペース

Copyright is held by the author(s).

* Xiang Li, 東京大学大学院 学際情報学府 Jun Rekimoto, 東京大学大学院 学際情報学府, ソニーコンピュータサイエンス研究所

に合わせて自然に流すシステムである。言語の壁を越えたスピーチやプレゼンテーションのサポートを始めるとするさまざまなアプリケーションが考えられる。SmartVoice を用いたプレゼンテーションの様子を、図 1 に示す。

2.1 SmartVoice を用いたプレゼンテーション

実際に SmartVoice を用いたプレゼンテーションの仕方について説明する。

まず、講演原稿を用意し、それに基づいた音声データを準備する。音声データは、ユーザ自身もしくは自分以外のナレーターによって原稿を読み上げた録音か、ボイスシンセサイザーによって自動生成される音声を想定している。用意した原稿テキストファイル (.txt) および音声データをシステムに与えると、SmartVoice は半自動的に原稿と音声のフレーズレベルの対応付けを行う。そして音声と正しく対応付けられる原稿中のフレーズは、ハーフミラープロンプターに表示される。

利用者は、用意した音声データをシステムに与え、顔がカメラ領域に収められ、トラッキングできる状態で、プロンプター画面を見て、原稿中のフレーズを口パクすれば、そのフレーズと対応する音声の流れてくる。一つのフレーズを読み終えて口を閉じると、再生が止まり、プロンプターには次に読むべきフレーズが強調される。再び口パクをすると、今読むべきフレーズと対応する音声の流れるが、それまでに持続時間任意の一時停止ができる。また、イントネーションも前述したように、口パクのペース、口の位置と眉毛の位置によってリアルタイムでコントロールできる。

2.2 SmartVoice におけるスマートな口パク

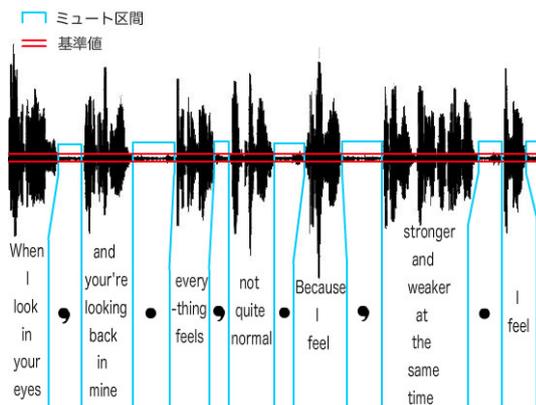


図 2. 音声と原稿からのフレーズ抽出と対応付け

2.2.1 音声の区切りおよび送り出し制御

実際に SmartVoice を使ってプレゼンテーションする時に、ユーザとインタラクションするためにプロンプターを利用している。ユーザがプレゼンター画面を見て、常に現在原稿のどこを読んでいるのかを確認することができる。そのために、音声とテキストは、正しく対応付けていることが必要である。SmartVoice は、この対応付けをシステムが半自動的に処理する。

通常の講演では、呼吸を入れることによって話のテンポを変えたり、一時停止することがある。なお、再生中の区切りに対し、違和感を感じさせないためには、最低限一つの単語の中での一時停止は避けるべきである。本システムでは、原稿テキストをカンマやピリオドなどのパンクチュエーションを区切りとしてフレーズを抽出し、音声データについては、フレーズは音声中に音響レベルが全音声データの平均値の一割以下で持続時間 0.1 秒以上の箇所（以下ミュート区間とする）によって分離された区間とする。一時停止は、ミュート区間内のみを許可する。ただし、特に録音の場合、ミュート区間中に、音響レベルが高いが、持続時間が極めて短いノイズは存在する。これらを見つけ出し、フレーズ抽出プロセスに無視されるよう、ノイズによって分断されるミュート区間が存在しないことを保証するためのアルゴリズムも用意した。(図 2)

原稿と音声の両方からフレーズが抽出されたら、本システムは後述するマッチングプロセスを用いて、原稿テキストと音声データの phrase-to-phrase の対応付けを行う。

プロンプターに表示される原稿テキストは、フレーズ単位で音声と対応し、ひとつのフレーズの音声を送り出したら、プロンプター画面には、自動的に次のフレーズのテキストに切り替わる。講演者はプロンプターに表示されるフレーズを「口パク」で読み上げるか、概ねそのように口を動かすことで音声データが順次送り出されていく。

2.2.2 ピッチ、スピード、音量の制御

音声データは講演者の口の動きによって送り出されていくが、さらに、そのピッチ、スピード、音量も講演者の顔情報から制御できるようにした。例えば、ボイスシンセサイザー音声データを利用した場合でも、このようにして抑揚や感情表現を講演時に付与することができ、より自然な講演になる。

これらの制御は、人間が実際に喋っている時の原理を考えて、方針を決めた。具体的に、ピッチについて、頭を下げる時にピッチを下げ、逆の場合はピッチを上げるようにした(図 3 - b)。次にスピードについて、口の動きが激しければ激しいほど語速を高く設定する。最後に、音量について、目の開き具合によってコントロールしようとしたが、今回は眉毛の

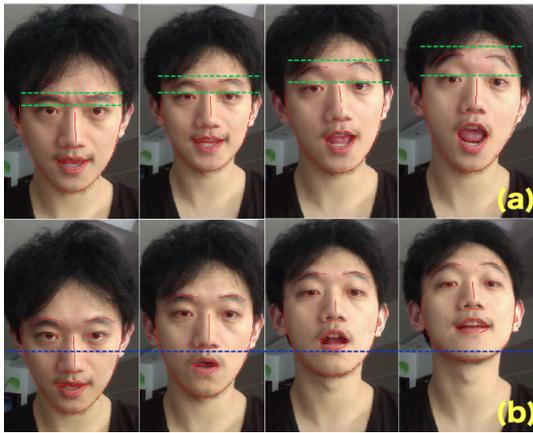


図 3. (a) 音量のコントロール (b) ピッチのコントロール

高さでコントロールすることにした (図 3 - a) . 理由は後述とする.

3 システム構成

本システムは、ホスト側とリモート側の 2 つの部分から構成されている。ホスト側は、ユーザの顔をトラッキングするためのカメラと、音声を出すためのスピーカーが付いている諸処理を担当するノート PC からなり、リモート側は、プロンプターとして使われるタブレット PC とプロンプタースタンドからなる。なお、両部分の間は、無線 LAN 通信でコミュニケーションを取る。(図 4)

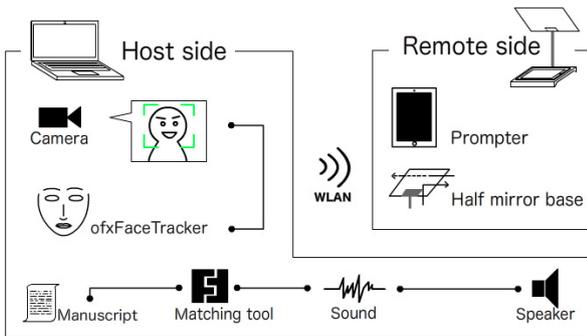


図 4. システム構成図

3.1 ホスト側

本システムは、プレゼンターに負担をかけず、より自然なパフォーマンスを実現するために、使用中に全ての操作は、口の動きや表情のみを使い、ハンズフリーで行う。プレゼンターは、再生の一時停止と再開を決めることが可能のほか、音声のスピード、ピッチそして音量をリアルタイムで変えることによって、ボイスシンセサイザーが生成した音声をより人間的

にすることができる。

3.1.1 顔認識およびフェイストラッキング

SmartVoice は、常時にプレゼンターの顔の位置、口の動きおよび表情をトラッキングする。本システムにおいて、openframeworks[1] 用の Add-on である ofxFaceTracker[2] を用いて、人間の顔およびその傾きや大きさ、さらには目や鼻、口、眉といった顔のそれぞれのパーツの位置や大きさを立体的に検知してトラッキングする。本システムのプロトタイプでは、Web カメラ付きノート PC を用いて、カメラから取得したリアルタイム映像からユーザの顔をトラッキングし、画面にある顔および顔の各パーツの位置、口の開閉パターンおよびそのペース、眉毛の位置などを常時に記録し、表示している。

3.1.2 Phrase-to-phrase の対応付け

前章で言及したように、本システムにおいて、原稿テキストと音声は、フレーズレベルで対応付けられている。この Phrase-to-phrase の対応付けの正しさを保証するために、本システムの一部として半自動的マッチングツールを導入した。

マッチングツールはまず、音声データ中のミュート区間を見つけ出し、フレーズを抽出し、各フレーズの再生時間を配列にキャッシュする。次に、原稿テキストから抽出されたフレーズを、順番にボイスシンセサイザーにかけて、生成された一時的音声データの再生時間をそれぞれ取得して配列にキャッシュする。

ボイスシンセサイザーによって生成された音声の再生時間は、語速の違いによって人間のナレーターによるものと著しく異なる場合があるため、マッチングツールは直接に前述の 2 つの再生時間の配列をマッチングするのではなく、音声データと原稿テキストのそれぞれ各フレーズの再生時間対総再生時間の比をマッチングの対象とする。

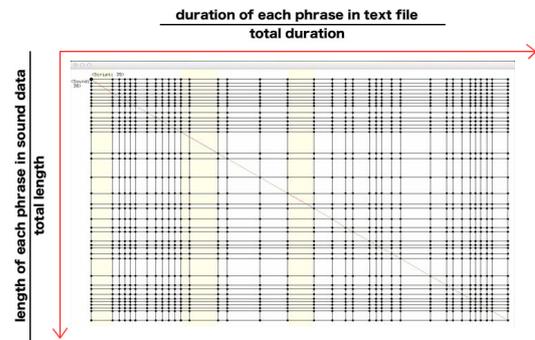


図 5. 自動マッチング結果を示すマッチングツールの画面

マッチングには DP マッチングアルゴリズムを利用している。図 5 のように、黄色に塗りつぶされた

部分は Miss match を表している。Miss match の原因は、原稿と音声から抽出されたフレーズの総数が一致しないことだと考えられ、主に2つのケースがある。一つは、本来は区切りではないところで音声を止めてしまったため、ミュート区間とされた場合。もう一つは、ナレーターが原稿中の区切りを無視して続けて喋った場合である。いずれのケースに対しても、本システムは手動修正をしやすいするための GUI をユーザに提供する。

3.1.3 再生コントロール、感情表現付け

SmartVoice では、一時停止の長さはユーザによって、任意に決めることができる。ミュート区間中に一時停止するたびに、システムはそのミュート区間を丸ごと飛ばすことによって、ユーザが再び口を開くと、音声がすぐ再生し、ディレイ無くプレゼンテーションを再開することが可能である。

前述の通り、ボイスシンセサイザーによるコンピュータボイスは、自然なイントネーションに乏しく、自然に喋っているように聞こえない。そこで、本システムは、再生時にリアルタイムで音声の音量変更、ピッチシフト、ピッチを変えずに再生速度を変えることによって、プレゼンターの感情表現を付けることを可能にした。音声のダイナミックピッチシフトおよびストレッチは、C/C++ベースのマルチプラットフォームで動作するタイム・ピッチマニピュレーションライブラリ Dirac3 を用いて実現している。[3] ピッチシフトについては、0.2 刻みでもとのピッチから上下 1 半音の変動幅に制限し、ストレッチについては、0.7 倍速から 1.6 倍速の幅に制限している。音量について、初期状態では 70 % にセットし、再生時に 50 % から 100 % までの間で変化することができる。

3.1.4 マッピング

本システムは、プレゼンターをプレゼンテーションに専念させるために、すべての操作パラメータは、トラッキングされているプレゼンターの顔のパーツにマッピングしている。

再生の一時停止と再開は、ユーザの口の開閉にマッピングする。フレーズ再生後にプレゼンターが口を閉じ、再び口を開くと、次のフレーズの再生が開始される。また、再生途中、つまりミュート区間以外の位置では、ユーザの口の開閉を問わず、再生し続ける。これによって、音声がスマートにプレゼンターの口の動きに合わせて流れてくることによって、まるで本人が喋っているように見えるという本システムのもっとも基本的な目的が実現される。

本システムでは、ピッチ、音量およびスピードは、再生時リアルタイムに変更を加えることができる。ピッチシフトについては、ユーザの顔位置にマッピングしている。具体的に、再生時にユーザの顔の位

置が初期状態顔の位置との縦方向の差で計算する。初期状態より顔を上げれば、ピッチが高くなり、逆に、頷くとピッチが低くなる（普通に喋る時に、頷くと声帯が圧迫されるため）。音量の変更について、前章で述べたように、今回はユーザの目の開き具合にマッピングしている。しかし、直接目の広き具合を取ると、変化率が小さいため、瞼と連動する眉毛の上下位置を採用した。ピッチシフトと似ており、再生時にユーザの眉毛の位置が初期状態眉毛の位置との縦方向の差で計算し、初期状態より眉毛を上げれば、音量が高くなり、逆の場合は音量が低くなる。スピード変更については、再生時に口パクのペースにマッピングしている。具体的に、今のフレームと直前のフレームにおいて口の面積の変化率を取って、口の動きが激しければ、スピードを上げ、逆にゆっくりと口を動かすと、スピードを落とす事が可能である。

3.2 プレゼンテーション時のインターフェース

3.2.1 プロンプター

本システムにおいて、プレゼンテーション時にユーザと実際にインタラクションをとるユーザインターフェースとして、原稿を提示するためのプロンプターを設けている。なお、本システムのプロトタイプ SmartVoice では、タブレット PC (iPad) をプロンプター画面とし、プロンプタースタンドとしてハーフミラーによるプロンプター台を利用している。ハーフミラー付きプロンプターを使うことによって、プレゼンターは講演会場を自然に見渡しながらかき読むことが可能となる。

3.2.2 ホスト側との協同作業

SmartVoice では、リモート側のプロンプターの稼働環境として iPad を採用し、リモート側とホスト側の間は、無線 LAN 通信を用いてコミュニケーションを取っている。その理由は、なるべくプロンプターの機能をシンプルのままに保ち、プレゼンテーションできるまでの準備作業は、すべてホスト側で完結可能にしたいと考えているからである。そのために、プロンプターに表示すべき原稿は、ユーザが用意したテキストファイル (.txt) をホスト側プログラムで読み込み、講演時に無線通信によりリモート側のプロンプター画面 (iPad) に伝送する。

実際にプレゼンテーションをする際に、ユーザとインタラクションをするのは、プロンプターであるため、ユーザは講演時にはプロンプター画面のみでシステムの挙動を確認できるようになっている。プロンプター画面は、図 6 - c で示しているように、ホスト側でうまく顔を認識しているか否かを、ラベル「DETECTED!」でユーザが知ることができる（認識できてない場合は「???」と表示）。

音声と原稿テキストはフレーズレベルで対応付け

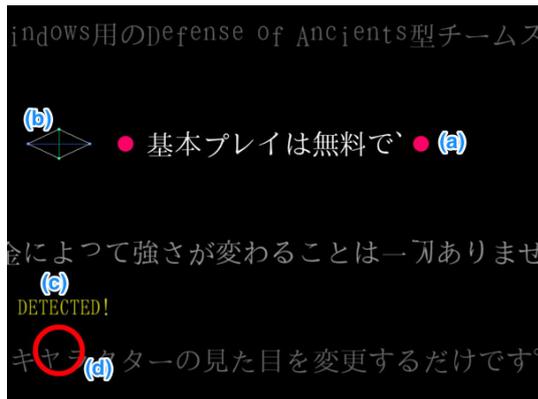


図 6. プロンプター画面

ているため、プロンプター画面にも、フレーズ単位で原稿をユーザに提示する。具体的に、プレゼンターが今読んでいるフレーズだけもっとも見えやすくし、読み終わったフレーズはスライドアップするようにしている。(図 6 - a)

ピッチや音量の変化のようになすぐに分かる変化と異なり、本システムを用いてナレーションの再生スピードを変える場合、現在の再生速度の目安をユーザに提示する。プロンプター画面では、口のイメージ(図 6 - d)が、喋る速度(ナレーションの再生速度)に合わせて、開閉アニメーションの速度が変わる。一時停止の時は閉じるままである。また前述のように、本システムにおいてスピードを決めるのは口の面積の変化率であるため、そのメタファとして、図 6 - b で示す 2 つのバーをプロンプター画面に追加した。この 2 つのバーは、ダイナミックに口のの高さと広さを表し、今取られているパラメータをユーザに提示する。

3.3 性能評価

本システムにおいて、フレーズレベルの音声と原稿テキストとの対応付けは、ホスト側が半自動的にしてくれる。ボイスシンセサイザーが生成するナレーションの場合、パンクチュエーションが遵守されるため、対応付けは正しく行われる。ところが、人声によるナレーションの場合、時々パンクチュエーションを無視したり、フレーズの途中で呼吸を入れたりするナレーションがあるため、対応付けが乱れることがある。その場合、本システムがのマッチングツールが提供するインタフェースを用いて対応付けを手動修正することができる。

本システムに使われるノート PC に付いているカメラは、最大解像度 720p で最大 30fps の撮影機能を備えている。本システム稼働中、プレゼンターによるすべての操作に対し、ホスト側プログラムの FPS は 55fps を上回っており、遅延はほとんど感じられない。

4 ユーザテスト・評価

SmartVoice の効果を確認するために、利用者評価実験を行った。通常のスピーチをしている動画と、SmartVoice を用いてスピーチをしている動画とをビデオ編集によって切り替えたものを実験参加者に見てもらい、どの部分が「口パク」(SmartVoice による箇所)を判定してもらった。判定結果はマウスボタンの押下により入力してもらった。声色の変化による影響を避けるために、この実験では本人が読み上げた音声を、本人が SmartVoice によって再生するというようにした。8 名(コンピュータサイエンス関係学科の大学院生)が実験に参加した。

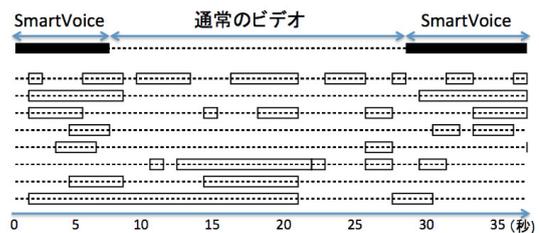


図 7. 生声原稿読み上げ v.s SmartVoice の実験結果

実験結果を図 7 に示す。それぞれの参加者が SmartVoice によるものと判定した部分を白抜きの矩型で示す。この図から明らかなように、SmartVoice ではない部分も口パクと判断した箇所がある一方で、SmartVoice による部分も本人の生のプレゼンテーションだと判断している箇所もある。この実験から、SmartVoice による音声対応は本人によるプレゼンテーションと比較しても自然であることがわかった。

5 議論

5.1 関連研究

フェイシャルアクションを使って、何らかの制御にマッピングする前例として、Lyons らによる The Mouthesizer がある。The Mouthesizer では、ミニヘッドマウント CCD カメラを用いて、口の中の影領域をトラッキングし、MIDI のエフェクト付きや楽器の操作などをユーザの口の動きにマッピングしている。[4] また、口のの高さ、広さの他、舌の動きと位置も使われている。Lyons らは、楽器を演奏するミュージシャンに両手以外のコントローラーを与えた点では、The Mouthesizer はフットペダルと同じだが、

Mouth コントローラーはペダルより直感的で身に着けやすいことを証明し、フェイシャルアクションによるマシンインターフェイスの可能性も言及した。[5]SmartVoice はコンセプト上、The Mouthesizer のとシェアするが、操作中に変わった表情などを取り入れていないため、プレゼンテーション中にユーザは自然な状態を保つことができる。

また、同じくフェイストラッキングを用いるが、音声ではなく、リアルタイムでCG キャラクターの表情を制御する研究もある。[6]

5.2 ボイスシンセサイザーによるビデオ作品の音声付け

従来、デモビデオなどにナレーションを入れたい場合、特にナレーションが母国語ではない時に、原稿のテキストファイルを用意し、ボイスシンセサイザーによって音声化する手法がある。しかし、前述の通り、ボイスシンセサイザーは単語一つ一つをキレイに発音してくれるが、抑揚に欠くため、機械的に聞こえる難点がある。本システムのように、抑揚やタイミングを人間がコントロールすると、ボイスシンセサイザーによる声をビデオのナレーション作成やアニメーションの人声の作成などにも応用できると考えている。

5.3 アドリブの可能性

今まで述べてきたように、本システムに使われる音声は、事前に作っておかないといけない。しかし、プレゼンテーションの場合、最初から最後まで原稿に準じて喋る必要がなく、途中で雑談を挟むことも考えられる。SmartVoice による音声と生声のスイッチによる違和感を隠蔽し、それを気づかせないようにすれば、SmartVoice と生声の間で自在に切り替え、プレゼンテーション中のアドリブも可能と考えている。

5.4 アフターレコーディングの可能性

ドラマや映画の作成は、撮影と音声はそれぞれ単独に収録し、プロの声優のアフターレコーディングによってセリフを映像に入れるのは一般的である。本システムは、リアルタイムにカメラからキャプチャーした映像に限らず、収録済みの映像中のキャラクターの顔をトラッキングして利用することもできるため、出来上がった映像を見ながらアフターレコーディングを行わなくても、SmartVoice を用いれば、タイミング正しく映像に音声を入れることが可能である。

5.5 歌、外国語練習への応用

繰り返しになるが、SmartVoice は「まるでその人が生で喋っているように見える」のようなシステムの実現が目的である。広義的な「人間が喋る」という点から、本システムが支援できるのは、スピー

チやプレゼンテーションに限らないはずだ。

例えば、カラオケと類似するプロンプターを有するため、練習しやすい上、SmartVoice はリアルタイムでピッチシフトとスピード変更もできるため、音感・リズム感の悪い人に、ピッチ感覚とリズム感を身に付けさせ、歌が上手くなることができるかもしれない。

また、外国語学習者の中で、その言語をできるだけネイティブっぽく喋れることを望む人が多い。しかし、発音やイントネーションを真似して、たくさん練習してそれを自分が喋る時の習慣とするのは結構大変である。外国語学習者用に SmartVoice にネイティブ声との比較機能を付けることによって、ネイティブ風のイントネーションを身につけることができるかもしれない。さらにもっと細かな顔筋肉がトラッキングできれば、SmartVoice による発音矯正も不可能ではないと考えている。

6 結論とまとめ

本論文では、言語の壁を超えたプレゼンテーションサポーティングシステム「SmartVoice」を提案した。音声データを分析し、原稿テキストとフレーズレベルの対応付けを実現することによって、ユーザの口の動きに合わせてスマートに音声を送り出すことを可能にした。また、ボイスシンセサイザーによる機械的なナレーションのイントネーションを、ユーザの口の形状・位置によってコントロールすることを実現した。本システムの効果を検証するために、評価実験を行い、SmartVoice による音声対応は本人によるプレゼンテーションと比較しても顕著な不自然がないことがわかった。今後、他の分野における SmartVoice の応用が期待できると考えている。

参考文献

- [1] openframeworks.
<http://www.openframeworks.cc/>
- [2] ofxFaceTracker.
<http://github.com/kylemcdonald/ofxFaceTracker>
- [3] Dirac3L. <http://dirac.dspdimension.com/>
- [4] L. Michael J., H. Michael, and T. Nobuji. The Mouthesizer: A Facial Gesture Musical Interface. In *Conference Abstracts, Siggraph 2001*, pp. 230, 2001
- [5] L. Michael J., T. Nobuji. Facing the Music: A Facial Action Controlled Musical Interface. In *Proceedings, CHI 2001, Conference on Human Factors in Computing Systems*, pp. 309-310, 2001
- [6] W. Thibaut, B. Sofien, L. Hao and P. Mark. Realtime Performance-Based Facial Animation. In *ACM SIGGRAPH 2011 Papers, SIGGRAPH '11*, pp. 77:1-77:10, 2011

視覚障害者のための写真撮影・閲覧・共有インタフェース

浅川 智恵子 原田 丞 佐藤 大介 石原 辰也 高木 啓伸*

概要. 写真は視覚的なメディアではあるものの、多くの視覚障害者が記録や晴眼者とのコミュニケーション手段として興味を持っていることが明らかとなっている。そこで本研究では視覚障害者のニーズ調査の結果に基づき、写真と音声メモ・周辺音を結びつけて記録することで、非視覚的な撮影・閲覧・共有を可能にするスマートフォン・アプリケーション“Accessible Photo Album”(APA)を開発した。日常的な状況での評価のため、5人の視覚障害者被験者が実験者とペアで外出して写真を撮影したうえで、別の実験者に写真を説明するという実験を行った。その結果、すべての被験者が音声メモと周辺音を活用して撮影・閲覧・共有を楽しむことができた。

1 はじめに

視覚障害者にとって写真は本質的にアクセス困難なメディアである。しかし、ソーシャルネットワークの普及によりコミュニケーション手段としての興味が高まってきた[6]。米国では、視覚障害者を対象とした写真教室もすでに実施されている[14]。視覚障害者が写真を撮影するためには、例えば構図の決定・露出・ブレ等の情報を非視覚的にどう伝えるか、撮影した写真を独力でどのように管理するか等の技術的課題が山積みであり、既に多くの研究が報告されている[6, 11, 1]。しかし、写真を共有するインタフェースの研究は例がなく、我々が実施したアンケート調査では、回答者の半数以上が、写真の共有に問題を感じていた[5]。

このメディアのギャップを埋めるためにはアクセス可能な情報、例えば音声を同時に記録する方法が考えられる。既にいくつかの音声録音可能なカメラやカメラ・アプリケーションが実用化されている[8, 9, 13, 16, 17, 18]が、これらは晴眼者を対象としているためアクセシビリティ上の多くの問題がある。ビデオで代替する方法も考えられるが、撮影時に対象物や人に対して一定時間カメラを向け続けなければならない、瞬間的に撮影可能な写真と比較して困難な操作が要求される。

そこで本研究では音声メモ付写真に着目し、視覚障害者による撮影・閲覧・共有を実現するためのスマートフォン・アプリケーション“Accessible Photo Album”(APA)を開発した。APAはスクリーンリーダを用いて操作することができ、撮影時に周辺音と音声メモを録音する機能を備えている。APAでは閲覧時に音声に加えて日付、時刻、GPS情報も非視覚的に参照可能なため、写真を容易に識別するこ

とが可能であり、さらなる解説を加えての共有が可能である。

本論文では関連研究に続いてAPAのデザインと実装を紹介する。その後、評価実験とその結果について報告したのちに、視覚障害者が独力で情報を発信する手段としての可能性について考察する。

2 関連研究

視覚障害者の中で写真撮影に対する興味が高まっている[25, 10, 19, 20]。Jayantらのアンケート調査[6]によると、アンケートに答えた70%のユーザがきれいに写真を撮影することに興味があると答え、ほぼ同数が最近写真を撮影したことがあると答えた。写真の識別には携帯型点字メモ機を併用して写真を撮影した順序に合わせて説明文を記録する、写真のファイル名に意味を持たせる等の方法をとっている。

2.1 音声付き写真

FrohlichとTallynが提案した音声付き写真は、写真とそれに付加された録音音声を1つのメディアとしたものである[3, 4]。この音声付き写真の概念は、特に視覚障害者のために考案されたものではないが、視覚障害者が写真を撮影・閲覧・共有する上で非常に有効であると考えられる。

音声付写真を撮影するためのスマートフォン用アプリケーションは、すでにいくつか実用化されているが[8, 9, 13, 16, 17, 18]、アクセシビリティ上の多くの課題がある。例えば、スクリーンリーダ機能を用いると1枚の写真を撮るために、35回のジェスチャ入力が必要となる。また音声付写真を撮影できるカメラも市販されているが、視覚障害者が操作可能な製品は筆者らの調査では見つからなかった。

2.2 視覚障害者のための写真撮影支援

iPhoneのVoiceOver[15]やAndroidのTalkBack等、スマートフォンのアクセシビリティ向上に伴い、

Copyright is held by the author(s).

* Chieko Asakawa, Susumu Harada, Daisuke Sato, Tatsuya Ishihara, Hironobu Takagi, 日本アイ・ビー・エム株式会社 東京基礎研究所

視覚障害者向けのアプリケーションが各種開発されている。写真関連では、画像認識やクラウドソーシングを用いて、視覚障害者が撮影した写真の内容を認識するサービスも利用が広がっている [2, 22, 24]。これらのサービスを利用することにより、ワイン、飲み物の銘柄、葉の種類、郵便物の内容等を遠隔からのサポートにより知ることができる。写真のブレの検出 [7]、視覚情報を音情報に変換する技術 [23]、写真のブレ、暗さ、傾きを音声で伝達する研究 [12] 等も報告されている。

3 Accessible Photo Album (APA)

視覚障害者が写真を撮影・閲覧・共有することを目的として APA を開発した。APA は以下の 3 つの方針に基づき設計した。

1. 素早く撮影し、音声メモと周辺音を録音できるように、非視覚的なジェスチャ操作を最適化する
2. 閲覧時に素早く写真を識別可能にするために音声メモの再生を効果的に取り入れる
3. 主体的に晴眼者と写真を共有可能にするために専用のモードを設けてインタフェースを最適化する

3.1 実装

APA は iPhone のアプリケーションとして実装し、読み上げには iOS 付属のスクリーンリーダーである VoiceOver を用いるように設計した。APA はカメラモードとアルバムモードの二つのモードを持つ (図 1)。起動直後にカメラモードに遷移し周辺音の録音が始まり、1 本指で下スワイプすることにより写真が撮影される。1 本指でタップして押し続けること (タップ&ホールド) で音声メモの録音が始まり、指が離れるまでの間記録することができる。録音終了直後に録音した音声メモが自動で再生される。

アルバムモードでは日付別にソートされた写真のリストが表示される。リスト間の移動には 1 本指で左右スワイプ、日付間の移動には 1 本指で上下スワイプを用いる。次の写真へ移動する際に、対応する音声メモが自動的に再生される。再生と同時に写真が撮影された日時と位置情報も音声合成により読み上げられる。位置情報については、写真が保存された時点の GPS 情報を地名に変換して読み上げる。晴眼者と写真を共有するときには、リスト上の写真の位置でダブルタップすることでフルスクリーン表示することができる。フルスクリーンビューでは VoiceOver の標準ジェスチャ機能である 3 本指の左右スワイプを用いて写真間を移動することができる。新しい写真が表示されると、音声メモの再生に続いて、写真の向き (縦方向または横方向) が読み上

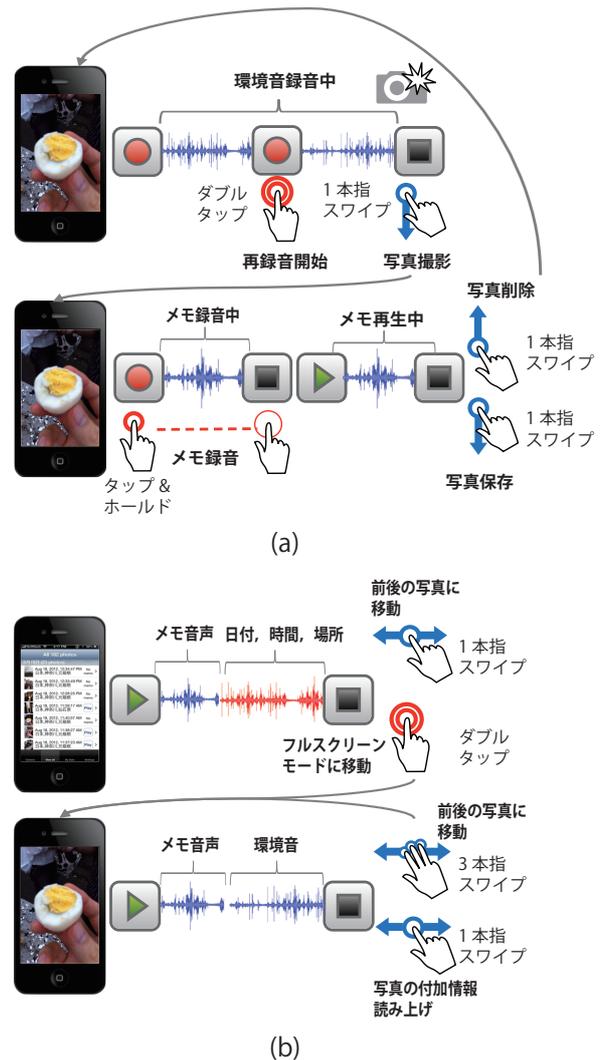


図 1. (a) カメラモードの操作方法, (b) アルバムモードの操作方法

げられるので、デバイスの方向を主体的に変更することが可能である。

4 ユーザ実験

5名の視覚障害者被験者の協力により APA の有効性を評価した。実験では以下の 3 つの点に着目した。

- (a) 写真の撮影、音声メモや周辺音の録音を素早く実行できたか?
- (b) それぞれの写真の識別をしながら、素早く閲覧できたか?
- (c) 晴眼者に対して主体的に写真を共有できたか?

4.1 被験者

被験者は、男性 3 名、女性 2 名、年齢は 25 歳から 36 歳までの 5 名で、謝礼を支払った (表 1)。

表 1. 被験者プロフィール.

	年齢	性別	視力	撮影経験	共有経験
P1	25	女性	弱視	有り	有り
P2	34	男性	全盲	無し	有り
P3	35	男性	全盲	無し	無し
P4	34	女性	弱視	有り	有り
P5	36	男性	全盲	有り	有り

4.1.1 写真撮影の経験

P1, P4, P5 は、カメラ付携帯電話で写真を撮った経験があり、画像のファイル名を変更することで写真を管理していた。P1 と P4 は、その方法に比較的満足していたが、P4 は、処理があまりにも単調であることから、日付や時刻で管理するようになり、結果としてあまり写真を撮らなくなった。P2 と P3 は、実験的に写真を撮った経験があるのみである。P3 は、自分のカメラで友人に写真を撮ってもらった経験があった。全ての被験者が独力で良い写真が撮れるかどうか不安を持っていた。撮影経験のある被験者も全員が晴眼者のサポートを得ていた。

4.1.2 写真共有の経験

P1, P4, P5 は携帯電話あるいは PC から e メール添付により他者と写真を共有したことがあった。P4 は写真をブログに投稿した経験があった。P2 は自分で写真を撮った経験はないが、彼が共有したい写真を友人に撮ってもらいその場で SNS に投稿した経験があった。

4.2 実験手順

実験は 1 人 1 日 (約 8 時間) で行い、以下の 4 つのセッションで構成される。

1) 導入セッション

被験者は、まず筆者らの研究所にて事前インタビュー・APA の操作方法のトレーニング・実験手順の説明を受けた。この導入セッションは平均 1 時間半であった。

2) 写真撮影セッション

第 1 セッション終了後、被験者は実験者と共に写真を撮るためあらかじめ被験者が行きたいと興味を示した場所に出かけた。そのためそれぞれの被験者は異なる場所で写真を撮影した。これは可能な限り日常に近い状況で実験を行うためである。目的地を決定後、実験者によって道順が確定される。これは単に美しい風景だけを写真に撮るのではなく、被験者が興味を持てるもの、例えば触って楽しめたり直接経験できるものに出会えるよう考慮したためである。実験者は、被験者に同行するだけでなく会話をしながら各地点を回り、周辺の環境の説明を行っ

た。あらかじめ決められた訪問地はあったものの、いつ移動するかについては、被験者に決定させた。時には、元々の道順から大きく外れることになったが、被験者が別の興味を見つけたときにはそれを優先した。また写真を撮るタイミングも彼ら自身が決定できるよう考慮した。被験者は好きな場所で自分自身の写真を取ってもらうよう実験者に依頼できることとした。この場合、実験者が写真を撮り音声メモは被験者が録音した。被験者がカメラの向きを合わせる手助けを必要とした時には、実験者が手助けを行った。カメラの向きを手で修正する前に、まず言葉で向きの修正を手助けした。これは、実際に写真を取っているのは被験者自身であると感じられるように考慮したためである。

3) 写真共有セッション

約 5 時間の外での実験終了後、被験者は研究所に戻り、写真共有セッションを行った。写真の撮影経験に関する簡単な質問の後に、被験者は観察者 (別の実験者) と撮影してきた写真を共有するためのセッションを実施した。被験者は、写真を共有しながらその日経験したことを観察者に共有する。音声メモを他人に聞かれることをためらった被験者がいたため、必要であればイヤフォンの使用を認め、被験者は観察者に聞かせたいと思ったときのみ音声を再生できることとした。観察者は被験者と写真を見ながら、その日の経験について自由に会話するよう指示された。被験者と外出した実験者も、写真共有セッションに同席した。観察者は、被験者が写真についての説明を始めるまでは、コメントや質問を待つよう指示された。これは、被験者が晴眼者の助けなしに、一人で写真の識別できるか確認するためである。観察者は、被験者が写真の識別が困難な時のみ、手助けするよう指示された。写真共有セッションは、平均 30 分前後を要した。

4) 実験後のインタビューセッション

1 日の実験の最後に、その日の経験や APA に関する被験者からのコメントを収集した。

4.3 実験結果

それぞれの実験の目的について、ユーザ実験により得られた知見を述べる。

(a) 写真の撮影、音声メモや周辺音の録音を素早く実行できたか?

VoiceOver の経験があったのは P3 のみであったが、他の被験者も、導入セッション中に APA を使用するために必要な基本的ジェスチャを習得することができた。写真撮影セッション中に撮影された写真は、13 枚から 20 枚であった (表 2)。

7 段階のリッカート尺度によるアンケート結果を

表 2. 写真の撮影場所と枚数.

	場所	枚数
P1	ショッピングモール/レジヤ施設	17
P2	ランドマークタワー/レジヤ施設	18
P3	ワインショップ/レジヤ施設	20
P4	史跡/公園	13
P5	中華街/ショッピングモール	15

表 3 に示す。「写真の撮影は容易であったか」との問いに対して、2名の被験者が7を、その他は5で回答した。「音声メモの録音は容易であったか」の問いに対しては、2名が7、2名が6、1名が5と回答した。携帯電話で写真撮影の経験があった3名の被験者に対して、以前の方法と、APA とを比較し、音声メモの詳細度、スピード、使いやすさについて評価を依頼した。その結果、3つの項目全てに対して、全員がAPA が優れていると評価した。

(b) それぞれの写真の識別をしながら、素早く閲覧できたか?

全ての被験者が問題無く閲覧操作を行え、撮影した写真を識別することができた。アンケートにおいても「閲覧は簡単だった」、「目的の写真を見つけることができた」、「撮影した状況を思い出せた」という問いに対して、P2を除くすべての被験者が、「全くそう思う」または「そう思う」と返答した(表3)。

P2だけは2枚の写真を混同したため、「目的の写真を見つけることができた」という問いに対して「どちらかというそう思う」と返答している。この2枚は、シーンが同じで1枚は実験者が撮った被験者とランドマークタワーの写真であり、もう一方は被験者が撮ったランドマークタワーだけであったが音声メモがほぼ同じであったため、P2はその区別ができなかった。

(c) 晴眼者に対して主体的に写真を共有できたか?

「写真共有セッションを主体的に行うことができたか」という問いに対して、P2を除くすべての被験者が、「全くそう思う」または「そう思う」と返答した(表3)。P2は、前述の1組の写真の識別ができなかったことから、「どちらともいえない」と答えた。この判断ミスが、彼のAPA全般に関する評価にどの程度影響を及ぼしたかを判断することは難しいが、他のすべての写真については、問題無く詳細を説明できていた。

4.4 観察

周辺音の有効性

P5が中華街の門で撮影した写真を共有していた時、周辺音として水しぶきが再生された。「自分のすぐ隣でだれかが水しぶきをあげていたことに気づ

表 3. アンケート結果

(1:全くそう思わない, 7:全くそう思う)

	P1	P2	P3	P4	P5
写真撮影セッション					
撮影は楽しかった	7	4	6	7	6
撮影は簡単だった	5	7	5	7	5
メモ録音は簡単だった	6	7	6	7	5
独力で写真を撮りたいと思うようになった	3	4	7	7	4
以前の方法との比較					
より簡単だった	7	-	-	6	7
より早くメモを残せる	7	-	-	6	5
より詳細なメモを残せる	6	-	-	7	6
写真共有セッション					
閲覧は簡単だった	7	7	6	7	7
目的の写真を見つけることができた	6	5	7	7	7
撮影した状況を思い出せた	7	6	6	6	7
写真を見せることは楽しかった	7	7	7	6	6
写真共有セッションを主体的にできた	6	4	6	6	7
実験全体について					
写真共有は楽しい	7	6	6	7	6
同様のアプリを使って写真を共有したい	6	7	6	7	3
実験前より写真を撮りたいと思う	6	6	6	7	7
実験前より写真を共有したいと思う	7	7	6	7	4

いていなかった」と、新たな発見をコメントした。P1とP4は、当初観察者に音声メモを共有することをためらっていたが、2枚の写真について周辺音を共有したところ、その価値に気付いたとコメントした。このときの周辺音は、噴水とブラスバンドの音であった。またP4は、周辺音を利用すれば視覚障害者の友人とも写真を通して経験を共有できるとコメントした。

主要なコメント

すべてのコメントは紙面の制限から掲載できないため、以下に主要なものを紹介する。

P1:「音声メモによって、自分で写真の管理ができるようになった。これはすばらしい進歩である。」

P4:「とても楽しかった。うまく写真がとれているかどうか不安だったため、写真共有についての楽

しさの評価は6としたが、それを除けば素晴らしい経験だった。これまでは自分の経験を言葉でしか伝えられなかったが、写真を使うことで視覚的に情報が伝えられることの楽しさを知った。こんな経験は初めてだった。」

P5:「写真が表示されたとき、すぐにそれについての話ができることは素晴らしい。周辺音に録音された周りの声を聞くと、写真を超えた情報を得ることができたと感じられて、とても良い気分である。」

5 考察

5.1 主体的な写真共有

APAの目的の1つは、視覚障害者が主体的に晴眼者との写真共有を可能にすることであった。その点でP3は最も主体的に、演出も交えて写真共有セッションを楽しんでいた。ある写真を見せる前に、まず観察者に音声メモが聞こえないようイヤフォンを装着し、次に画面が観察者に見えないようiPhoneを裏返した。「多分私がどんなボトルを偶然みつけたか想像できないと思いますよ」と話した後に、画面を元に戻し「コンコルド型のウスキーのボトル」の写真を観察者に見せた。このように手に持つことのできるデバイスだからこそ可能な視覚的演出を、その効果を判断することのできない視覚障害者自身がとったという点で実験者らの予想を超えた行動であった。今後は、このような“演出”も含めた写真共有の表現力向上を目指したインタフェースの工夫や知見の共有が必要となるであろう。

5.2 ユーザインタフェースの改善

音声メモや周辺音をどのタイミングで録音するかは、APAのようなアプリケーションを設計する際に重要な要因となる。実験に使用したAPAでは音声メモと周辺音を別々に取り扱った。シャッターが押されるまでの“周辺音”には晴眼者によるカメラの向きの指示や、周囲の状況説明などの音声が含まれてしまったため、実際には撮影後の“音声メモ”のほうが有用であった。

そこで、実験後にAPAのインタフェースを修正し、シャッター前録音を廃止し、シャッター後にユーザ自身が“タップ&ホールド”ジェスチャにより、周辺音と音声メモを自由に録音できるインタフェースに改良した。この場合周辺音と音声メモの記録の違いはなく一つの音声ファイルとして保存される。これによりシャッターを押す、“タップ&ホールド”でメモと周辺音を録音するというシンプルなインタフェースに変更された。

5.3 音声メモとテキストメモ

これまで写真にテキストでコメントをつけていた被験者も、音声メモは簡単に、素早く、そして周辺

音を含めた詳細な説明が付与できるとコメントした。確かに音声メモは作成が容易であるという大きな利点があるが、テキストには写真の検索が容易になる、ネットワーク上で共有するときに既存サービスと親和性が高いなどの利点がある。今後、音声認識の利用も含めてテキストを音声メディアに取り組み必要がある。

また、二人の被験者が音声メモを後で修正して詳細度を向上したいとコメントした。P5は、観察者からのコメントを音声メモに追加することで、さらに写真の回想が容易になると述べていた。このように音声メモの編集に対するニーズは高く、検討の必要がある。

6 まとめ

本研究では視覚障害者による写真の撮影・閲覧・共有を可能にするスマートフォン・アプリケーション“Accessible Photo Album”(APA)を開発し、評価実験を行った。APAでは写真撮影と同時に音声メモや周辺音を録音し、閲覧・共有時に自動的に再生することで視覚障害者が独力で写真を識別し共有することが可能になる。実験の結果、晴眼者とであっても主体的に写真の共有が可能になるだけでなく、独自の演出表現までも可能になることを観察できた。写真共有は晴眼者だけでなく視覚障害者にとっても価値がある活動になることがこの実験でも裏付けることができた。

今後、より豊かな写真体験を視覚障害者に提供するためにはメディアのギャップを埋めるためにメタデータの拡大が重要な課題の一つとなる。例えばユーザが室内・外どこにいるか、位置情報、コンパス、データベースを組み合わせて写真に写っている建物の詳細な解説などが付与できるであろう。さらにクラウドソーシングのアプローチにより写真の詳細な説明を付加することも可能になってきている。写真は「ソーシャルメディア時代の新たな言語である」とも言われるほど、その重要性が高まっている。視覚障害者もそのような時代に独自の方法論を蓄積してコミュニティ参加できるようにするための技術開発が今後も必要となるであろう。

参考文献

- [1] Baber, C., Cross, J., Khaleel, T., and Beale, R. Location-based photography as sense-making. *Proc. BCS-HCI 2008, British Computer Society (2008)*, 133-140.
- [2] Bigham, J.P., Jayant, C., Ji, H., et al. VizWiz: nearly real-time answers to visual questions. *Proc. UIST 2010, ACM (2010)*, 333-342.
- [3] Frohlich, D. and Tallyn, E. Audiophotography: practice and prospects. *Ext. Abstracts 1999, ACM (1999)*, 296-297.

- [4] Frohlich, D.M. Audiophotography: Bringing photos to life with sounds. Kluwer Academic Pub, 2004.
- [5] Harada, S., Sato, D., Adams, D. W., Kurniawan, S., Takagi, H., and Asakawa, C. Accessible photo album: enhancing the photo sharing experience for people with visual impairment. *Proc. CHI 2013*, ACM (2013), 2127–2136.
- [6] Jayant, C., Ji, H., White, S., and Bigham, J.P. Supporting blind photography. *Proc. ASSETS 2011*, ACM (2011), 203–210.
- [7] Ko, J. and Kim, C. Low cost blur image detection and estimation for mobile devices. *Proc. ICACT 2009*, IEEE Press (2009), 1605–1610.
- [8] Many Ltd. Picle. <http://www.picleapp.com/>.
- [9] Robinson, S. Com-Phone Story Maker. <https://play.google.com/store/apps/details?id=ac.robinson.mediaphone>.
- [10] UCR/California Museum of Photography. Sight Unseen. <http://www.cmp.ucr.edu/exhibitions/sightunseen/>.
- [11] Vazquez, M. and Steinfeld, A. Helping visually impaired users properly aim a camera. *Proc. ASSETS 2012*, ACM (2012), 95–102.
- [12] White, S., Ji, H., and Bigham, J.P. EasySnap: real-time audio feedback for blind photography. *Proc. UIST 2010*, ACM (2010), 409–410.
- [13] Luo Xu. Speature. <http://itunes.apple.com/us/app/speature/id401887412>.
- [14] Blind with Camera School of Photography. <http://www.blindwithcameraschool.org/>.
- [15] Apple - Accessibility - iOS - Voiceover. <http://www.apple.com/accessibility/ios/voiceover/>.
- [16] StoryMark Life. <http://www.storymarklife.com/>.
- [17] Voicepic. <http://voicepic.me/>.
- [18] Storyrobe. <http://storyrobe.com>.
- [19] Photos by Blind Photographers - Photo Essays - TIME. <http://www.time.com/time/photogallery/0,29307,1897093.1883579,00.html>.
- [20] Seeing Beyond Sight: Photographs by Blind Teenagers. Book, video & TED Talk. <http://www.seeingbeyondsight.org/home/>.
- [21] Blind With Camera. <http://www.blindwithcamera.org/>.
- [22] oMoby. <http://www.iqengines.com/omoby/>.
- [23] Seeing with Sound - The vOICE. <http://www.seeingwithsound.com/>.
- [24] TapTapSee. <http://www.taptapseeapp.com/>.
- [25] 伊藤 邦明. 七重, 光をありがとう. 河出書房新社, 2000

未来ビジョン

APA を用いて作成された音声情報群を共有することで、他の視覚障害者の経験を楽しむメディアとして利用できる可能性がある。ユーザ自身が音声情報を写真・地理情報とあわせてソーシャルネットワーク上で共有することで、視覚障害者自身の手でソーシャルなオーディオデータベースを作り出すことができる。このアプローチをジオボイスタグと呼ぶこととする。類似したサービスはこれまでも様々な場面で議論されてきた。しかし、テキストタグやフォトタグと比較してあまり注目されていない。理由の一つはジオボイスタグを利用したアプリケーションに限られるためであろう。視覚障害者にとっては状況が異なる。例えば視覚障害者が晴眼者の友人とジオボイスタグを聞きなが

ら歩けば、視覚的には得られない周辺情報を獲得することができ、より対等な立場から会話を楽しむことができる。晴眼者は意図的に周辺情報の説明をする必要がなくなり、視覚障害者はレストランやショップの情報、それらの評判についても主体的に会話をリードできるようになる。視覚障害者が一人で街中を歩く場合のナビゲーションサポートの可能性もある。このシナリオ実現には周辺音を妨げないでボイスタグを提示する必要がある。視覚障害者は通常、車の音をはじめとした周辺音を音声ランドマークとして街中を移動するためである。共有サービス自体のデザインと並行して、使用状況を考慮したインタフェースや、ソーシャルメディア上の情報との融合、高齢者等の他のユーザ層に向けたユースケースなどを検討する必要がある。

個人的な小さな幸せを実現するブラウザ上での動画編集・共有手法

中村 聡史* 石川 直樹† 渡邊 恵太‡

概要. ニコニコ動画では、動画を一緒に楽しむだけではなく、もともとある動画をもとに、それを改良し、さらなる動画とする N 次創作がそのコミュニティの一体感と、動画の再発掘に有効に働いている。しかし、こうした N 次創作はある程度の創作が前提となっているものであり、例えば音量の変更や、音楽の変更や、部分的な音声の差し替え、音声の追加といった事だけでは創作とはなりえないものである。そうしたちよつとした編集だけを行った動画をアップロードすると、パクリとなじられるであろう。本研究では、そうした微細な編集によって個人的な満足感を達成可能とするだけでなく、その編集を共有することで新たな動画視聴体験を可能とする仕組みを実現する。ここでは、「ブラウザ中で視聴中の動画に対する音量増減や音声の付与などの編集手法」と「付与された動画の共有手法」を実現とする。これにより、部分的に音量を調整したり、音楽を差し替えたり、音声を足したり、部分的にスロー再生やスキップしたいなど個人的な嗜好に基づく動画編集および視聴を可能とする。

1 はじめに

YouTube¹やニコニコ動画²に代表される動画共有ウェブサイトが爆発的に成長している。YouTube では 2012 年 1 月時点で 1 秒あたりに動画長にして 1 時間分の動画が投稿され³ている。また、ニコニコ動画でも、2012 年 11 月 20 日の中村らの調査 [1] によると、1 日あたりに約 5500 本の動画が、1 秒あたりに動画長にして約 47 秒分の動画が投稿されている。YouTube は 2012 年 1 月の時点で 1 日に 40 億視聴、ニコニコ動画では 2012 年第 2 四半期に 1 日あたり 1 億視聴、平均視聴者数は 786 万人で、ユーザの平均視聴時間は 102.5 分であるという⁴。

ニコニコ動画では、先人が創作した動画を活かし、新たな動画を創作するという N 次創作 [2, 3] が多く見られる。N 次創作では、元々 VOCALOID などで作成および投稿された音楽に対して、それにマッチする映像を付けてさらにアップロードしたり、ピアノなどで曲を弾き直してみたり、曲はそのままに歌を自分が歌ったもので差し替えてみたり、踊りを考えて音楽に合わせて踊ってみたりなど、多様な発展が見られる。こうした N 次創作は、ニコニコ動画を発展させ、コンテンツをより良い物へと高めるとともに、愛されるコンテンツを生み出している理由の一つでもある。

ここで、ニコニコ動画で N 次創作に参加するハードルについて考える。オリジナルのコンテンツに対して「歌ってみた」「演奏してみた」「踊ってみた」などタグが付けられているような動画は N 次創作の 1 つである。ちなみに、2013 年 8 月 29 日時点で、ニコニコ動画上で「歌ってみた」で検索される動画は 691,641 件、「演奏してみた」動画は 166,800 件、「踊ってみた」動画は 100,078 件ある。YouTube で「歌ってみた」で検索される動画は約 904,000 件、「演奏してみた」動画は約 586,000 件、「踊ってみた」動画は約 540,000 件ある。なお、海外での「歌ってみた」に相当する（吹き替えも含まれる）「Fandub」で検索すると、その件数は約 1,960,000 件となる。

N 次創作は、オリジナルとなるコンテンツ（動画）を改良するものであるため、参入障壁は、コンテンツを最初から作成するよりは低いように感じられる。しかし、この「歌う」「演奏する」「踊る」という事であっても技量が要求され、ハードルが高いといえる。さらに、アレンジする場合は、その技術力も必要となる。映像をつけるのも同様のハードルの高さが存在するであろう。結果として、N 次創作は行われているものの、そこに参入しているユーザの総数は限られる。また、動画編集ソフトウェアを用いた動画編集や、動画のエンコード、動画のアップロードなどにも障壁は存在する。

ここで、ユーザが動画を視聴している際に、その動画に対して微妙に編集したいと思うことがある。例えば、ユーザが気に入っている楽曲動画なのだけれど、音量が終盤でフェードアウトしておらず、ぶつくと切れてしまうためフェードアウトさせたいと感じることや、作業用 BGM として音楽のメドレーを聴いている際に、ある曲だけは苦手なのでスキップしたいと感じることなどもある。それ以外にも、折角

Copyright is held by the author(s).

* 明治大学 総合数理学部, JST CREST

† 東京農工大学 工学部

‡ 明治大学 総合数理学部, JST CREST

¹ <http://www.youtube.com/>

² <http://www.nicovideo.jp/>

³ <http://jp.techcrunch.com/archives/20120123youtube-reaches-4-billion-views-per-day/>

⁴ <http://dic.nicovideo.jp/a/ニコニコ動画>

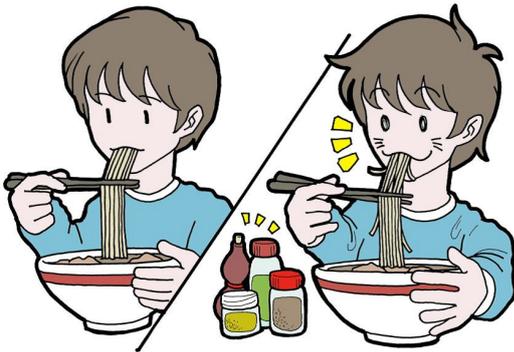


図 1. (左) 調味料をたさずにそのまま食べる (右) 調味料を足して自分好みに調整し食べる

の素晴らしい演奏動画なのだけれど普通に終わってしまうため拍手や歓声などを入れたいなどと感じることや、映像自体は好きなものだけれど BGM があまり好きなタイプではないため、BGM を差し替えたなど様々である。しかし、仮に高価な動画編集ソフトウェアをもっており、編集能力が高いユーザであったとしても、他者が投稿した動画をダウンロードし、音量を微調整だけしたり、部分的に音を追加したり、部分的にスキップしたりといった、微細な変更だけでアップロードするということは世の中から N 次創作として許容されるとは考えにくい。そうした動画をアップロードすると、他の視聴者から「パクリ」と糾弾されてしまうこともあるであろう。

そこで我々は、上記のような編集を、オリジナルの動画に手を加えるのではなく、システムによって再生している動画自体をコントロールすることによってコンテンツを個人化する手法を提案および実装する。また、他者と共有可能にすることによって、編集も 1 つのコンテンツとして楽しむことが可能な仕組みを実現する。これは、ラーメン屋などにラーメンを食べに行った時に、自分の好みに合うように調味料を足すことに似ている (図 1)。

本提案手法によって、N 次創作とは違う、よりカジュアルに動画と係ることが可能な N 次装飾とも言える一つの動画の楽しみ方を提供する。

2 ブラウザ上でのカジュアルな編集と個人化

ニコニコ動画に見られる N 次創作は、その創作のもととなるオリジナルのコンテンツに敬意を払いながら、利用するという点でとても良いコンテンツ生態系を作り出していると言える。しかし、動画の作成者 (N 次創作の作成者も含む) は、動画の視聴者数に比べたら圧倒的に少ない。これは、ニコニコ動画自体にアップロードされている動画の平均的な質が高いために、気後れしてしまっているという点もあるが、そもそも動画作成および N 次創作のハードルの高さも原因の 1 つでもある。

ここで、オリジナリティとクオリティが高い楽曲や映像の動画があり、多くの人から評価されているが、ユーザ個人としては「微妙に気持ち悪いから修正して欲しい」とか、「もう少しこうであつたら良いのに」などと感じることがある。実際、そうしたコメントはニコニコ動画中のコメントで現れていることも多く、不満コメントに対し「あなたが作りなおしてアップロードしたら良いじゃないか」という旨のコメントが付くことも珍しくない。不満を述べるだけというのは実際生産的ではないが、元々の動画の投稿者以外のユーザが、その高い品質の動画をダウンロードし、ユーザの個人的な好みに合わせて微調整だけした動画をアップロードするというのは、N 次創作とは言いがたいものであり、他者からも受け入れられないであろう。

ここで、ユーザ自身がつ、個人のための動画編集のニーズを調べるため、ニコニコ動画や YouTube などに投稿されている不満や要求に対するコメントを分類および整理し、ピックアップすると下記の通りとなる。

- 動画自体は質の高い映像コンテンツではあるものの、音楽が好みのものではないという、ユーザとしては映像と音楽が合っていないように感じている。そこで、映像はそのままに、音楽だけ好みのものに変更したい。
- ユーザが楽しんでいたストーリー性のある動画において爆発音が入っていたが、その爆発音の部分が好みに合わなかったため、他の爆発音と差し替えたい。
- N 次創作として「歌ってみた」や「演奏してみた」「踊ってみた」という動画は、オリジナルの同一楽曲を元に、各投稿者が歌ったり、演奏したり、踊ってみたいものである。このそれぞれの動画について、踊ってみた動画の映像と歌ってみた動画の音声を組み合わせたり、歌ってみた動画の音声とオリジナル動画の映像を組み合わせたり、ギターで演奏している動画とピアノで演奏している動画を組み合わせたりしたい。
- 作業用 BGM とタグ付けられている動画は、複数の楽曲をセットとして 1 動画とされているものであり、ユーザはこの動画を再生するだけで複数の楽曲を連続的に楽しむことができる。その楽曲の一部が気に入らないのでスキップしたり、ある曲だけ音量が低すぎてバランスが悪いため、その楽曲部分のみ音量を調整したい。
- 研究に関するデモビデオが無音であり面白みにかけていた。そこで、そのビデオに追加音声や、音楽を付けて面白くしたい。

- ある素晴らしい出来の動画があるが、音声がついていない。みんなで音声を持ちよって、この無声動画に音を付与したい。

上記のニーズは一部に過ぎないが、その多くはオリジナリティがあるとは言いがたく、独自に動画を投稿した時に許容されるかどうかは怪しい。つまり、上記のニーズを満たすには、動画として再編集およびアップロードするのではなく、個々の編集情報だけを記録し、その編集記録を元に動画を視聴可能な仕組みの実現が必要となる。我々は、こうした動画の個人化（自身の好みに変更すること）をブラウザ上で実現可能とし、その超個人化された編集情報を他者と共有することを可能とするものである。

3 プロトタイプシステム

3.1 必要要件

ユーザ（視聴者）の嗜好に基づき、微妙に動画コンテンツを編集し、個人向けにしたいという欲求を満たすため、我々はウェブブラウザ上で、実際に視聴しながら編集することが可能な編集手法を実現する。システムとして求められる要件を整理すると、下記の通りとなる。

- 視聴中の動画の任意の再生時間に対して、ボリュームを上げる、下げる、消すなどの音量に対するコントロールを可能とする。
- 視聴中の動画の任意の再生時間に対して、一部スキップや、スロー再生などの再生に対するコントロールを可能とする。
- 視聴中の動画の任意の再生時間に対して、任意の音声、音楽追加を可能とする。
- 上記機能を Web ブラウザ上で実現可能とする。
- 動画と編集情報による、新たなる動画編集 ID (URL) で動画を管理する。
- 動画編集 ID をベースとして、他者と共有することを可能とする。
- 共有された編集情報を元に、さらなる編集を可能とする。

3.2 実装方法

本提案システムを、Google Chrome の拡張としてプロトタイプシステムを実装した。本システムは、YouTube やニコニコ動画などのページで動画がロードされると、動画の ID を取得 (YouTube やニコニコ動画上で動画を一意に識別する文字列) し、その ID に合致する編集情報の有無を確認する。編集情報が存在しない場合は、動画編集のためのコントロールを提示し、編集情報が存在する場合は、その情報に基づきコントロール上に編集情報を提示する。こ



図 2. システム導入前 (左) と導入後 (右)

こで、コントロールの表示は、ページ中にコントロール用のコードが含まれる DOM を挿入することで実現している。

音量の変更や、動画のスキップ、再生速度変更といった再生に関するコントロールは、ニコニコ動画や YouTube の動画を再生するコントローラの API を利用して操作している。また、付与される音声については、その音声がある URL 上の音楽/音声ファイルの場合は、そのファイルをロードし、時刻に合わせて同期的に再生させる。さらに、YouTube などの動画を追加音声として利用する場合は、別のタブとして動画をロードし、タイミングを合わせて再生するよう実装している。つまり、システムとしては同時に複数のコンテンツを再生しているだけである。

なお、システムはニコニコ動画および YouTube では、動画プレイヤーをコントロールする API を利用しているだけである。そのため、ラッパーを作成するだけで、他の動画共有サイトなどにも適用可能となる。

3.3 操作方法

図 2 の左は、本プロトタイプシステムを導入する前の YouTube 上のオリジナルのインタフェースで、右が本プロトタイプシステムを導入した後の YouTube 上のインタフェースである。

編集可能インタフェースは、音量コントロールグラフとシークバー、スキップ、再生速度、共有という 3 つのボタン、そして「Drop a media file here」という点線の四角形で囲まれた領域からなる。YouTube では動画下部に編集インタフェースが追加され、ニコニコ動画ではシステムの都合上動画上部に編集インタフェースが追加される。

音量コントロールグラフでは、ユーザはグラフ上の点をマウスのドラッグ・アンド・ドロップ操作によって上下させることにより、その再生時間における音量を変更することが可能となっている。また、シークバーは、通常の動画のシークバーのインタフェースと同じく、動画の再生に応じて自動的に動いていくうえ、ユーザの操作によって任意の再生時間に移動することが可能である (図 3)。

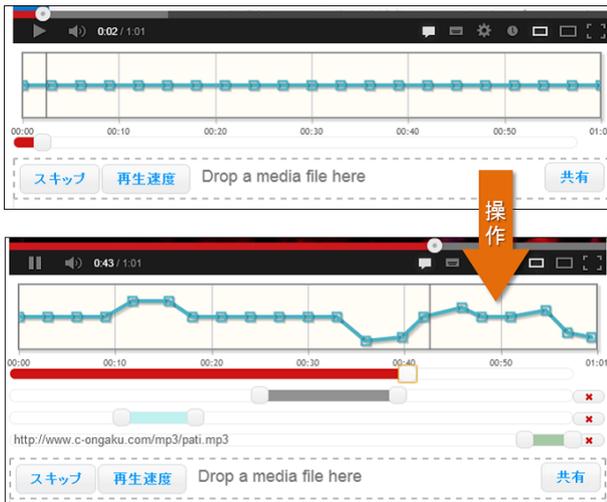


図 3. 編集インターフェースのみを拡大した状態

ユーザがスキップボタンを押すと、編集インターフェースに1つの時間軸が追加され、レンジスライダー（灰色背景の部分）が現在の動画の再生場所を左端に、そこから10秒後を右端にした状態で追加される。このレンジスライダーが指定されている部分が自動的にスキップされる区間となる。ここで、スキップ開始場所はレンジスライダー左端のドラッグ・アンド・ドロップ、スキップ終了場所はレンジスライダー右端のドラッグ・アンド・ドロップによって変更することが可能である。

ユーザが再生速度ボタンを押すと、編集インターフェースに1つの時間軸が追加され、スキップボタンの時と色が違うレンジスライダー（水色背景の部分）が追加される。現在は、この再生速度ボタンによって指定された部分は、0.5倍速で再生される（プロトタイプシステムでは固定）。

「Drop a media file here」という点線で囲まれたスペースに、ユーザがローカルのメディアファイルや、ウェブ上のメディアファイルのURLをドラッグ・アンド・ドロップすると、編集インターフェースに1つの時間軸が追加され、レンジスライダーが追加される。レンジスライダーは、現在の動画の再生場所を左端に、そこからドロップしたファイルの長さ分後を右端にした状態で追加される。ここでは、ユーザがレンジスライダーの左端、右端どちらをドラッグ・アンド・ドロップしても、メディアファイルの開始位置が変更される。また、ドロップされたファイルがどこから取り込まれたのかという情報は、時間軸インターフェース上に合わせて提示される。これにより、リファレンス情報が明確になる。

ボタンの押し下げや、ファイルのドロップなどによる編集内容は、追加された時間軸の右端に登場する×ボタンを押すことにより削除することが可能となっている。また、対象となる動画をユーザがあらかじめ編集している場合は、その編集情報が自動的に

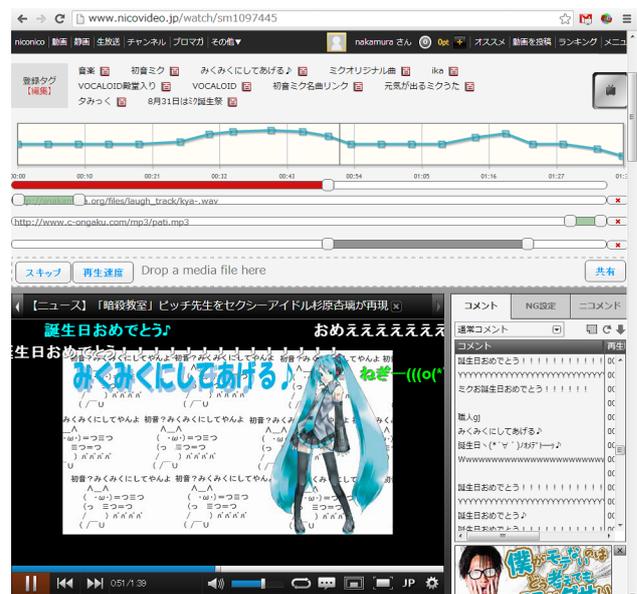


図 4. ニコニコ動画上のインターフェース

にロードされる。

なお、編集情報はローカルに保存されるが、共有ボタンを押すことによって、編集情報のみを他者と共有することが可能となっている。

3.4 利用例

図4は、本システムをニコニコ動画上で利用している様子である。ここでは、動画の最初の部分に大歓声を入れ、通常再生しサビ部分に向けて音量を徐々に上げている。また、1つ目のサビ終了後、2つ目のサビは飛ばしてすぐに音楽の終盤に飛び、音楽の最後では音量を下げるように指定されている。さらに、動画の最終盤に拍手と歓声が再生されるようになっている。

図5は、本システムをYouTube上で利用している様子である。動画自体は、ある研究者が20個のテーマを45秒ずつ（トータル15分）で発表している様子である。ユーザがこの発表の動画を他者に紹介しようと思った際に、10分しか紹介に使える時間がなかったため、紹介相手が興味をもってもらえなさそうなテーマについてはスキップするように指定する一方、あつという間に進んでしまっただけで、再生速度を変更しスロー再生することによって時間を長めにとっている。また、発表者の音量が大きかったり小さかったりとぶれがあったため、なるべく一定になるように音量コントロールインターフェースで変更している。さらに、全体的に音声盛り上がり欠けていたため、笑って欲しい所にラフトラックを入れたり、最後に盛大な拍手を入れることによって、盛り上がり演出している。

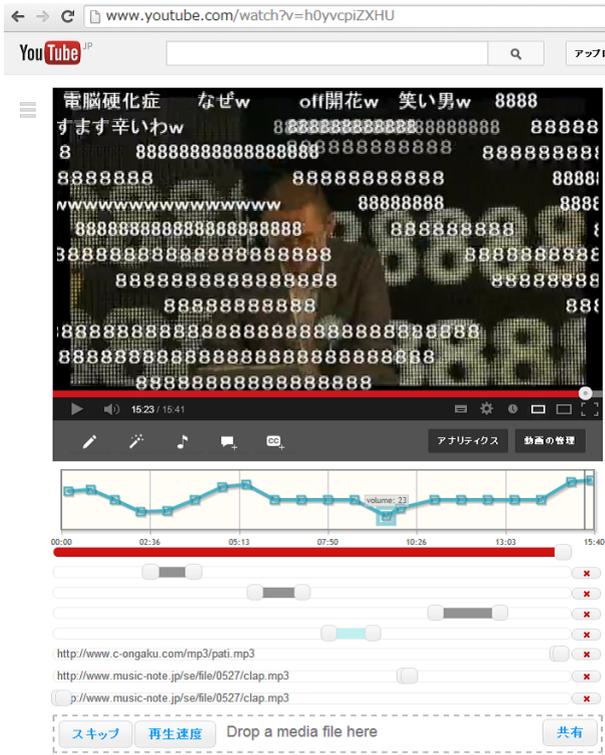


図 5. YouTube 上でのインターフェース

4 考察

本提案手法により、演奏してみた動画の最後に拍手を入れたり、踊ってみた動画の最後に歓声を入れたり、歌ってみた動画の途中途中で合いの手の声を入れたり、科学系の動画の途中途中で解説の説明を音声として入れたり、わかりにくい部分はスロー再生または停止して説明したりといった事が可能となった。この既にある動画に対する編集（装飾）は、その動画コンテンツを変質させる。例えば、ストーリー性のある動画について、途中途中でラフトラック（笑い声）を挿入すると、より面白く感じるという事が実際のコンテンツ制作より観察された。今後は、その有効性をユーザ実験などによって明らかにしていく予定である。

動画の再生時間が長い場合に、スキップや挿入した音声などがどこを指しているのかがわかりにくく操作しづらいという問題があった。この問題については、動画編集ツールにありがちな時間軸を拡大縮小するようなインターフェースを導入することである程度解決可能であると考えられる。

今回提案している手法は、様々な所から集めてきたリソースを1つの動画としてコンテンツ内に取り込むのではなく、部品として外部に存在させつつも1つの動画であるかのように振る舞わせることができるということである。つまり、どの音声、どの音楽がどこから来たのかというリファレンス情報が明確に存在しているということであり、そのリファ

ンス情報を元にオリジナルコンテンツへのアクセスも可能にするものである。ウェブ上の動画などで、リソースの取得元を明記していないケースも多いというえ、仮に明記されていたとしてもアクセスは容易ではない。そういった点で、オリジナルの作成者も尊重しつつも、リファレンスしながら新たなコンテンツを生み出すことができる事が可能となっている。

これまで、ニコニコ動画など動画共有サイトでは、オリジナル動画の作成者と、N次創作をする動画の作成者、動画の視聴者という3つのタイプの利用者が存在していた。ここで、オリジナル動画の作成者やN次創作をする動画の作成者に比べ、動画の視聴者の数の方が圧倒的に多い。我々の手法は、この圧倒的に数が多い動画の視聴者自体に、編集という形で動画に積極的に関与することを可能にするものであり、動画コンテンツを中心とした生態系がより成長するのではないかと期待される。

一方、本提案手法はオンライン上での編集および視聴を可能とするものであるため、公式サイトから配信されているような動画コンテンツも、本手法を使うことによって擬似的に編集し、視聴することが可能となる。例えば、公式のチャンネルとして、アニメなどが配信されているものについて、オープニングやエンディングを飛ばしたり、ダイジェストシーンのみにするよう編集しておき、ダイジェストのみ視聴といったことも可能となる。また、そうした公式の動画に対して、音声を付与したり、別の音楽に差し替えたりといった事も可能となる。また、本手法ではある公式動画の映像と、ある公式動画の音声の組み合わせを動画を違法にダウンロードして編集するのではなく、同期的に再生することで組み合わせとして楽しむことが可能となる。このような動画の使用は、コンテンツの世界を広げるものと期待される。

また、本提案手法を利用することによって、ユーザが好きなアーティストの楽曲をCDから取り込んでmp3化し、ウェブ上のお気に入りの動画の映像に合わせて再生する（その動画の音声自体は消音にする）ということも可能となる。ここでは、ユーザはそうしたmp3ファイルをウェブ上にアップロードして他者と共有するわけではなく、個人的に楽しむだけであるため、著作権の侵害にも当たらず、楽しむことが可能となる。

現在は、YouTubeとニコニコ動画のみに対応しているが、システムの処理としてはYouTubeやニコニコ動画の動画コントロールのためのAPIを利用しているだけである。そのため、そうしたAPIさえ存在し、ラッパーとなるプログラムを書けば、他の動画サービスでも本手法を適用可能となる。例えば、Ustreamなどに保存されている、WISSの登壇発表などに対して、説明の音声を付与したり、発表と無関係なシーンを飛ばしたりといったことが動画

を編集すること無く実現可能となる。

5 関連研究

Seek Rope[4] は、YouTube などの動画共有サイトにおいて、シークバーをロープのように取り出し、ロープを丸めることで繰り返し再生したり、ロープを繋げることで複数の動画をつなげたり、ロープを格子状に配置することでランダム再生を行ったりすることを可能とするシステムである。ウェブ上のコンテンツを活用するという点で類似性があるが、本研究は再生のコントロールのみならず、音声や音楽、他動画などによって装飾可能であるという点で違いがある。

ウェブ上のコンテンツを組み合わせる新たなコンテンツを作り出す試みとして Massh![5] がある。Massh!では、楽曲同士をインタフェース上でつなぐことによってマッシュアップ音楽を作成可能としている。ウェブ上のコンテンツを組み合わせ、利用するという点で類似はあるが、我々の手法は個人的な欲求を満たすというものであり、その表現方法や目的は異なる。

MixBit[7] は、短い動画クリップ (16 秒) を組み合わせることで、新たな動画を作成可能とするシステムである。MixBit では、ユーザ自身が動画を撮影せずとも他者が撮影した動画を活用して動画制作が可能となっており、動画制作のハードルを下げている。ただ、MixBit は閉じたコミュニティの中の動画を再生時間に沿って順に繋ぎ合わせるものである。我々の手法は、コミュニティに閉じることを無くすすべてのオンラインコンテンツを対象としており、また映像と音声などの組み合わせも可能としている。

ラジへえ[8] は、ラジオを聞きながらその音声に対して音声コメントをボタン1つで入れる事を可能にするものであり、コンテンツの他者との楽しみを強調するものである。本研究で提案する手法の一部は、ラジへえの音声付与に似ている。本提案手法は、編集を目的としており、ラジへえのようなリアルタイムでのコミュニケーションは対象としていない。

SMIL[6] はウェブ上でのマルチメディアコンテンツの振る舞いを記述するマークアップ言語である。SMIL では、同期的にマルチメディアコンテンツを再生するといった記述も可能であり、異なるフォーマットの動画ファイルを並べて再生することなども可能である。我々の手法は、SMIL の仕様とは異なり、ブラウザ自体がウェブ上のリソースおよびプレイヤーを直接操作する事が可能となっているため、多様な表現が可能である。ただ、SMIL の記述スタイルは参考になる部分が多いため、今後編集情報の共有などにおいて参考にする予定である。

6 まとめ

本研究では、ウェブ上の動画に対して、ブラウザ上で音量コントロールや、スキップ操作、音声や音楽の付与といった編集を可能とし、その編集情報に基づき視聴可能な仕組みを実現した。また、編集情報を他者と共有可能な仕組みを実現し、他者の編集を参考に視聴したり、他者の編集を利用して再編集したりといった事を可能とした。

今後は、本プロトタイプシステムをウェブ上で公開し、多くのユーザに利用してもらうことでフィードバックを集める予定である。また、現在の操作インタフェースおよび機能は、まだ十分に練られていないため、ユーザからのフィードバックなどを元に改良を行っていく予定である。

ユーザによる編集情報と、ニコニコ動画のコメントにはある程度の相関がある可能性がある。そこで、ユーザの編集情報と、ニコニコ動画のコメントを相互分析することによって、自動編集を可能とする仕組みも実現する予定である。

参考文献

- [1] 中村聡史, 山本岳洋, 後藤真孝, 濱崎 雅弘: 視聴者反応と音楽的特徴量に基づくサムネイル動画の生成手法, 情報処理学会論文誌 (トランザクション) データベース (TOD58), Vol.6, No.3, pp.148-158 (2013-06-28).
- [2] 濱崎雅弘, 武田英明, 西村拓一: 動画共有サイトにおける大規模な協調的創造活動の創発のネットワーク分析-ニコニコ動画における初音ミク動画コミュニティを対象として-, 人工知能学会論文誌, Vol. 25, No. 1, pp. 157-167 (2010).
- [3] 濱野智史: アーキテクチャの生態系 - 情報環境はいかに設計されてきたか, NTT 出版 (2008) .
- [4] 佐藤剛, 宮下芳明: SeekRope: 曲げて切って結べるシークバー, インタラクシオン 2010 論文集, pp. 197-200 (2010).
- [5] 徳井直生: Massh!, <http://www.sonosphere.com/mash/> (2008).
- [6] W3C: Synchronized Multimedia, <http://www.w3.org/AudioVideo/>.
- [7] mixbit: <https://mixbit.com>.
- [8] 加藤由訓, 苗村健: ラジへえ: ラジオ聴取時における感想共有システム, インタラクシオン 2013, pp.32-39 (2013).

HirakuReader : 行間を拡張する電子書籍

中村 将達 西田 健志*

概要.

本を読む時には、書かれている内容を鵜呑みにするのではなく批判的思考を持ち、関連する情報や異なる立場からの意見に触れるなどの学習を通じて広い視野を獲得する「能動的読書」を行うことが理想的だと言われているが、そのような読書習慣を身につけて持続させていくことは容易ではなく、誰しもができるものではない。こうした中、電子書籍は書き込みを読書間で共有できる機能や関連情報が素早く得られるという点から能動的読書を促す効果が期待されている。だが、電子書籍の書き込み共有機能が堅い意見交換の場となってしまうことで堅い場が苦手な人々が避けがちになってしまうという問題があり、能動的読書の習慣のない人々はなおさらそのような傾向に陥る可能性がある。そこで、本研究では娯楽性を考慮したデザインによって堅い意見交換が苦手な人にも本への気軽な書き込みを促し、本に対する積極的な態度を喚起する HirakuReader を開発した。HirakuReader は、行間を拡張するインターフェースを用いることで本の中に存在する行間をメディアとして機能させることができ、読者による投稿によって書籍コンテンツが変容していく電子書籍リーダーである。行間を活用した情報提示によって、読者の投稿を促し、関連情報への素早いアクセスを実現する。本論文では能動的読書を実現するアプローチと、そのアプローチに基づいた HirakuReader のデザインと実装について述べる。

1 はじめに

本を読む時には、本に書かれている内容を鵜呑みにするのではなく批判的思考を持ち、関連する本や情報、異なる立場からの意見に触れるなどの学習を通じて広い視野を獲得する「能動的読書」を行うことが理想的だと言われている [9]。しかし、そのような読書習慣の獲得が高等教育での目標のひとつとされていたり、読書法に関する啓発本が多数出版されていたりすることからもわかるように、そのような良い読書習慣を持つことは容易ではない。

それに対して能動的な読書を促す方法として注目されているのが、読んでいる本とともに様々な情報を提示することができる電子書籍である。たとえば、Kindle や Kobo[6] といった主要な電子書籍リーダーでは、書籍中の特定箇所のハイライトや、特定のフレーズに対するメモや感想を他の読者と共有することができるようになってきている。また、本の感想を共有するための SNS サービスも数多く存在している [15][14]。

しかし、そのような既存のシステムは、ユーザを「変なことを投稿してはいけない」という気持ちにさせるなど堅苦しさを感じさせている可能性がある。電子書籍へ注目が集まる一方で読書よりもテレビやインターネットの手軽なコンテンツを好みがちな若い世代の読書量の低下が問題視されている中¹、電子書籍が堅い意見交換の場となってしまうはまず

まず読書から遠ざかる人が出てしまう恐れがある。

我々は、コメントの重畳表示などに見られる利用者の交流を最優先するニコニコ動画 [13] の精神に電子書籍も倣うべきだと考える。交流最優先の精神が生み出した弹幕やコメント職人といった新たな遊びは堅苦しさを払拭することで多くの人々を巻き込み、その結果として政治のような堅い話題までもが語られる場を作り出すことに成功している。ニコニコ動画自身も、同じようなインタフェースで漫画や小説を楽しめるサービスを提供しているが [12]、文字が主体の情報にコメントを重畳するのはあまりに読みづらいため、電子書籍に対しては同様の精神を受け継ぎながらも異なるインタフェースをデザインする必要があると思われる。

その一試みとして本研究では、書籍中の行間を簡単なタッチジェスチャで広げて空間を作り出し、その空間を利用して関連情報を提示したり様々な情報を書き込んで共有したりできる電子書籍リーダー HirakuReader を開発した。特に、行間に書き込まれる内容や量が変化の様子を徐々に太くなっていく線や燃え上がるエフェクトなどで視覚化することで、コメントの内容うんぬんよりもとにかくコメントする楽しさ、他の読者と気楽にやり取りする楽しさを前面に出し、堅苦しさをなるべく感じずに交流できるように配慮している。読者の投稿によってインタラクティブに変化していく行間は文章そのものの可読性を落とすのが、読者がそのような破壊的

Copyright is held by the author(s).

* Masatatsu Nakamura, Takeshi Nishida, 神戸大学大学院 国際文化学研究科

¹ 読書離れに関しては諸説あり [8][4]、読書離れが起きているとは一概には言えない。

な行動を行うことは本から受けがちな権威的な印象を和らげ能動的読書を促す効果があるものと期待される。本論文では、まずデザイン目標を議論し、その後HirakuReaderの実装および関連研究を紹介し、最後に今後の課題を述べる。

2 デザイン目標

本研究では行間を拡張するインターフェースのデザイン目標として以下のものを掲げている。

1. 本は与えられたまま読むものだという意識の破壊
2. 能動的な読者による書籍そのものの変化
3. 読書時の快適な情報量調節

2.1 本は与えられたまま読むものだという意識の破壊

電子書籍上での意見や注釈の共有機能は、付箋などといった紙の本のメタファーに則ったものとして実装されているものが多く、なるべく中身を汚さないようにするという紙の本の感覚が誘発され、本に対する受け身な態度を助長してしまう。

そこで我々は、読者の働きかけによってコンテンツが変容していき、書き込み行為自体が娯楽性を持つような本を実現することで読者の積極的な読書態度を促すことができるのではないかと考えた。

本研究では行間というスペースを活用することで、読者の働きかけが本のコンテンツを演出するような機能デザインを行った。HirakuReaderでは、本の中にある行間を「開く」ことで書き込みを行うことができ、そうして行われた書き込みは読者間で共有される。書き込みが行われた行間は光ったりアニメーションしたりすることで情報提示を行い、文章の見た目を変容させる。投稿によって変化する行間の表現性や娯楽性によってさらなる投稿を促し、より読者を積極的に本に介入させることを狙った。

2.2 能動的な読者による書籍そのものの変化

本には著作という概念が存在するため、第三者が内容を自由に書き換えることができないが、本研究では行間というスペースを用いて読者の投稿と情報提示を実現しているため、コンテンツそのものの書き換えは行なっておらず、著作物を破壊することなくコンテンツのさらなる充実を可能にしている。

本は時が立つにつれて書き換える必要がある部分などが出てくるが、そういった本は、概ね改訂版で内容が更新されたり、新版のまえがきなどに説明が付け足されたりする。電子書籍では、読者たちの投稿によって改訂版を待たずして内容の訂正などが可能になるが、そのためには読者たちに積極的な情報投稿を促す必要がある。本研究では書き込み行為に娯楽性を持たせることで、読者たちの積極的な態度

を喚起しており、積極的な読者が増えることで内容が日に日に更新されていくような本のあり方を実現することを狙った。

2.3 読書時の快適な情報量調整

本を読んでいる際、元の文章だけでなく関連する情報を閲覧する場面がある。単語の意味、文章内に出てくる用語の詳細な説明といったものに限らず、他の読者たちの投稿した情報を参照することがあるだろう。その際、情報閲覧までにかかるステップが煩雑であるほど読書行為が阻害されてしまうため、なるべく素早く、読書の邪魔をしないような情報提示が求められる。

従来の電子書籍アプリケーションにおけるインターフェースでは、単語部分の長押しや、画面上に現れるアイコンをタッチするなどといった操作を行い、ポップアップウィンドウや画面端の領域に情報を提示するという手法が採用されている。しかし、そういった情報提示が行われる領域は文章表示の邪魔をしないために一定のサイズが保たれており、読者がもっと情報を得たいと感じた時には画面遷移を伴う必要がある。

HirakuReaderのインターフェースでは、二本指で単語や行間を挟み込むことで選択し、そのまま指を広げていくことで文章のレイアウトを変化させて空間を作りだし、そこに様々な情報提示を行うことで関連情報へのアクセスを可能にしている。選択から検索までをスムーズに行うことができ、もっと情報を見たいと感じた際にはそのまま指を広げていくことでさらに多くの情報提示することができるため、閲覧する情報の量を快適に調整することができる。

3 HirakuReader

本章ではタブレット端末向けのアプリケーションとして実装したHirakuReaderの機能について説明を行う。

3.1 行間を広げるジェスチャインターフェース

行間を広げる操作は、二本の指で画面にタッチしている状態からそのまま指を広げるという動作で実行できる。このインターフェースは読書中でも素早く行うことができ、各種情報へのアクセスを素早く、容易にしている。行間を広げると文書のレイアウトが上下に動いていき、ユーザが指を広げれば広げるほどそれに応じて行間が広がっていく。このインターフェースでは、二本指で画面に触れた際の指の真ん中の位置がポインティング位置となっており、二本指の真ん中で単語や行を狙うことで選択を行う(図1)。



図 1. 行間や単語の選択の際、二本指でタッチしている位置の真ん中がポインティング位置となっている。

3.2 行間を広げて関連情報を検索する機能

文章内の単語は二本指で挟み込むことで選択することができ、そのままピンチアウトすることでその単語が含まれる行を挟んで上下の行間を広げられる(図2)。広げられた行間にはその単語に関する情報が提示され、辞書や各種 Web サービスでのその単語による検索結果を表示することができる。単語に関する情報は指をどれほど広げるかによって提示される内容が変化し、少し広げると辞書、もう少し広げると画像検索の結果や Wikipedia のページなどを閲覧することができ、指を広げるほど多くの情報を得ることができる。

3.3 広げた行間に情報を追加し、共有する機能

文章内に存在する行間も単語同様に二本指で挟み込むことで選択することができ、選択された行間を挟んで上下に広げていくことができる。誤って単語の選択が行われてしまわないよう、行間の選択にはページ横の余白のスペースを利用し、その余白の中で任意の行間を二本指の真ん中で狙うという操作を行うことで選択を行えるようにした(図3)。

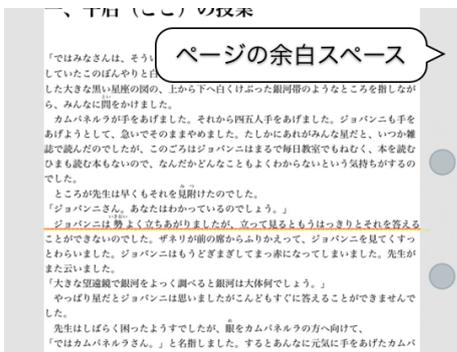


図 3. ページ横の余白領域(グレーの色がかかった領域)の中で任意の行間を二本指の真ん中で挟み込むことで行間の選択を行う。

この広げられた行間には、意見や内容の補足など

といった情報を画像や動画などを交えて投稿することができる。こうして投稿された情報はアプリケーション間で共有され、読者の手によって行間に含まれる情報は充実していく(図4)。投稿の際には文字のサイズや色、表示される際の位置などを指定することができ(図5)、提示される情報に表現の幅を持たせた。



図 5. 投稿用のフォーム。文字の大きさや位置、色を設定することができ、画像も投稿できる。

3.4 行間へ追加された情報を視覚化する機能

読者によって情報が投稿されている行間は、閉じられている状態では線が引かれ、光る・アニメーションするなどの表現を行い、情報が埋め込まれていることを知らせる。投稿が少ない状態では、行間は細い点線で表示され、投稿が増えていくとほかに線が太く点線が細くなっていき、最終的には太い直線になる(図6)。

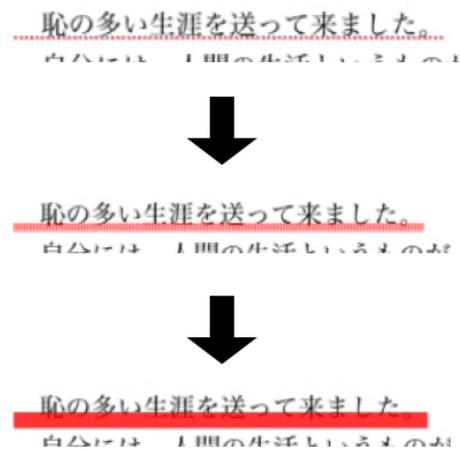


図 6. 行間は投稿が少ない状態では細い点線で表され、投稿が増えるほど点線が細かく、太くなってゆき、行間が強調される

読者が設定できる投稿の情報には、文字のサイズや色、位置などがあり、投稿された書き込みの色に応じて行間の色も変化するなど、設定が直接反映されるような表現を実現している。行間の表現には燃え上がるなどの演出機能も用意しており、こうした強力な表現は本を破壊的に読むメタファーとして機能する(図7)。

これらの機能によって読者の働きかけでインタラクティブに変化する行間を実現し、投稿する行為に



図 2. 文章中の単語が選択され、検索されている様子。文章内の単語は二本指で挟み込むことで選択することができ、選択中の単語は黄色くハイライトすることでわかりやすくした。

娯楽性を持たせることでより積極的に本と関わろうとする読者の態度を誘発することを狙った。

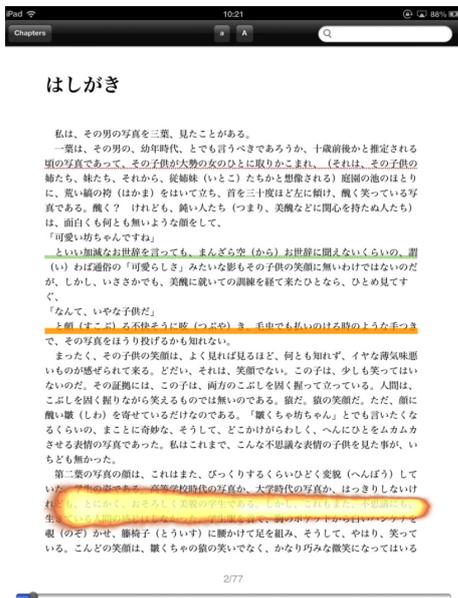


図 7. 情報が投稿された行間の表現。投稿の量や設定によって表現は変化し、行間が燃え上がるようなエフェクトもある。

3.5 実装

本システムはサーバークライアントシステムとして実装を行った。クライアント側では、iPad タブレッ

トを利用し iOS アプリケーションとして実装した。電子書籍のフォーマットには HTML ベースの標準的な EPUB3 を対象としており、CSS や JavaScript を用いて書籍のデザイン調整などを行った。行間の表現などのダイナミックな部分は HTML を用いずに、iOS の API を用いて実現している。サーバー側は Ruby を用いて実装を行っており、データベースには MySQL を用いた。

3.6 ユーザフィードバック

本システムのプロトタイプを同じ研究室の学生に使用してもらい、出てきた声をまとめる。

- 難しい本や退屈な本でも面白い書き込みがあることで読み進めようという気持ちになった
- 一度読んだ本でも、自分の後から読んだ人の書き込みが行われることでまた読みたくなった
- 単語を選択して素早く検索できるため、本を読み進めるのが早くなった、またそれに伴って英語の本が読みたくなった
- 単語を検索して出てくる画像などが面白く、本を読み進める上で参考になる場面があった
- 画面上下の端において、行間を広げる際に狙った箇所とは違う行間を広げてしまうことがあった
- 文字部分を長押しすることで選択する従来の文字選択手法に比べて、二本指で挟む手法は選択部分が指で隠れずに済み素早く選択する

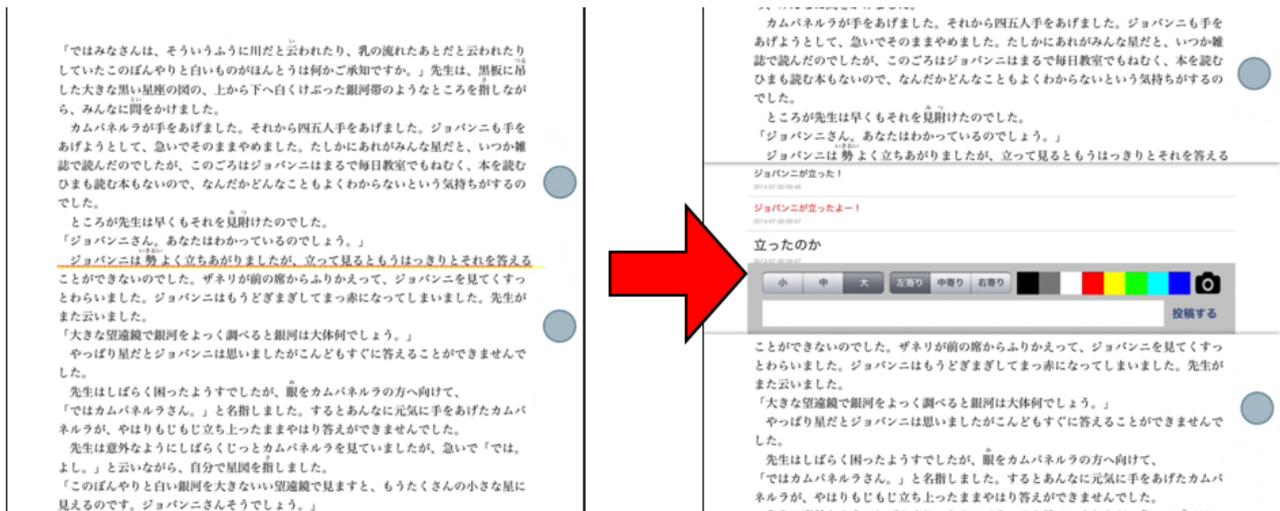


図 4. 情報が投稿されている行間を選択して、情報を閲覧している様子。文章中の行間も単語同様二本指で挟み込んで選択することができ、選択中の行間は黄色くハイライトしている。

ことができたが、文字の細かい選択範囲が操作できないことに不便を感じた

面白い書き込みがあることで読み進めようという気持ちが促進されることや、検索の素早さから関連情報を得やすくなり本の理解度が増すなどの声がある一方で、行間を広げる操作を間違えてしまうなどの声もあり、インターフェースに改善の余地があることを示唆している。

4 今後の課題

本システムは、肝となる行間の表現や、文字選択機能や行間に書き込まれた情報の共有機能に関して課題がある。

4.1 行間の表現

読者の投稿によって変化する行間の表現は娯楽性を引き出す上で重要であるが、どのような情報が投稿されるとどのような行間の演出が現れるかといった点については未だ検討中であり、これからの課題となっている。例えば、燃え上がるなどの表現(図7)は強力な演出となりうるが、いかにして読者の投稿と絡めて効果的に用いるかは検討していく必要がある。

4.2 投稿の共有

電子書籍はデバイスの解像度や画面サイズ、設定された文字サイズによって行の長さが変化してしまうため、特定の行間に書き込んだとしても、他のデバイスや文字サイズで閲覧する際には書き込みが表示される位置が変化してしまう問題がある。

また、行間にしか書き込みを行えない現状のシステムでは文章中の特定の単語や節に対してアノテ

ションしたいというニーズには応えることはできない。書き込み箇所を曖昧にしておくほうが気軽な書き込みを誘発できると我々は考えているが、アノテーションとの兼ね合いを考慮した書き込み箇所の指定方法についても考察する必要があるだろう。

4.3 文字選択

行間を広げるインターフェースは、二本指の真ん中で単語や行間を狙って選択を行うが、文字範囲の選択に関して一つの課題がある。英語などの元々分かち書きされている言語を対象とする場合には、一つの単語を容易にポインティングすることができるが、日本語などの分かち書きされていない言語の場合は困難が生じる。

5 関連研究

Schilit らの [9] は紙の本で行われる注釈やマーキングといった書き込みをデジタルの文書でも行えるようにした。ペンによる自由な形式での注釈やマーキングをつけながらの読書は、文章の理解度を促進する効果があるが [18]、デジタル文書は紙の文書と違ってレイアウトが柔軟に切り替わるためにそうした情報は管理しにくく [3]、また大量の読者による情報共有は実現しづらいため、本研究ではフォームを用いた情報の投稿を採用した。

電子書籍上での注釈情報の共有は Johnson らによって教育上の効果があることが確認されており [5]、電子書籍のこうした機能には教育ツールとしての可能性が期待されている [7]。

このようなユーザ間での注釈情報の共有は、書籍よりも Web ページを対象としたものが積極的に試みられている。その中には Web ページ上に重畳表

示する手法がいくつかあり、ペンデバイスを用いた手書きのもの [17] や付箋メタファーを用いたもの [2] など、本に対して行われる書き込みを模倣した手法が多く、電子書籍での注釈共有を考える上で参考になる。

ニコニコ動画 [13] の動画を見えにくくするようなコメントシステムは、ユーザによるコメント投稿に娯楽性を持たせており、コメントそのものが動画コンテンツの一部と化している。同じ動画共有サイトの YouTube [11] とニコニコ動画では、ユーザの行動パターンに顕著な差が見られ [16]、ユーザの働きかけを積極的に引き出すシステムによって齎されるコンテンツの受容態度の変化が見て取れる。ゲームなどの娯楽性のあるシステムを用いることでユーザから効果的に情報を収集する研究 [10] もあり、娯楽性や表現性といったものがユーザの積極性を引き出すという効果には期待ができる。本研究の電子書籍システムも、行間への書き込み行為が娯楽性を持つような機能やデザインを試みている。

本研究では、行間を広げることで情報提示を行ったが、文章のレイアウトを柔軟に変化させることで効果的な情報提示を行った研究として Chang らの [1] がある。本の脚注などの情報を見るためには、読んでいる部分から離れた位置まで視線を移動させる必要があり、読書の負担となる。Chang らは本文中に情報提示を行う空間を作りだすことでそうした問題を解決している。我々の手法では指を広げるといって単一の動作で、レイアウトを変化させて情報提示空間を作り出す機能と情報提示空間を広げていく機能を実現しており、情報への素早いアクセスが可能となっている。

6 まとめ

本研究ではなるべく多くの人が能動的読書を達成するための試みとして、娯楽性によって読者を引き込んで本への積極的な態度を促すというアプローチを提案し、電子書籍リーダー HirakuReader を開発した。HirakuReader では行間をメディアとして活用することで読者による本への書き込みに娯楽性を持たせ、関連情報へ素早くアクセスできるようにした。今後は正式なユーザスタディを行い、娯楽性が読書態度をどれほど改善するのかといった点や、行間を広げるインターフェースの効果の調査等を行いたい。

電子書籍の浸透がきっかけとなり、今後の本のあり方に関して様々な議論が交わされているが、本研究が本のあるべき姿に一つの方向性を示すことを期待している。

参考文献

- [1] B.-W. Chang, J. D. Mackinlay, P. T. Zellweger, and T. Igarashi. A negotiation architecture for fluid documents. In *Proc. ACM '98, UIST '98*, pp. 123–132, New York, NY, USA, 1998. ACM.
- [2] diigo. <http://www.diigo.com>.
- [3] G. Golovchinsky and L. Denoue. Moving markup: repositioning freeform annotations. In *Proc. ACM '02, UIST '02*, pp. 21–30, New York, NY, USA, 2002. ACM.
- [4] 日本人の読書離れは進んでいるのか. <http://research.goo.ne.jp/database/data/000618/>.
- [5] T. E. Johnson, T. N. Archibald, and G. Tenenbaum. Individual and team annotation effects on students' reading comprehension, critical thinking, and meta-cognitive skills. *Computers in Human Behavior*, 26(6):1496 – 1507, 2010.
- [6] Kobo. <http://kobo.rakuten.co.jp>.
- [7] D. G. Lebow and D. W. Lick. HyLighter: An effective interactive annotation innovation for distance education. In *20th Annual Conference on Distance Teaching and Learning*, pp. 1–5, 2005.
- [8] 若者だけじゃない、各年代で進む読書離れ 30・40代女性の半数は1か月の本代1,000円未満. <http://moneyzine.jp/article/detail/206052>.
- [9] B. N. Schilit, G. Golovchinsky, and M. N. Price. Beyond paper: supporting active reading with free form digital ink annotations. In *Proc. SIGCHI '98, CHI '98*, pp. 249–256, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- [10] L. von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proc. SIGCHI '06, CHI '06*, pp. 55–64, New York, NY, USA, 2006. ACM.
- [11] YouTube. <http://youtube.com>.
- [12] ニコニコ動画. <http://seiga.nicovideo.jp>.
- [13] ニコニコ動画. <http://nicovideo.jp>.
- [14] ブクレコ. <http://bookreco.jp>.
- [15] ブクログ. <http://booklog.jp>.
- [16] 沖松健太郎, 松本淳. 動画共有サイトにおけるユーザー参加型のオープン・クリエーションモデル—YouTube とニコニコ動画の比較分析, 日本社会情報学会 第 23 回全国大会. 2008.
- [17] 浜口拓輝, 加藤直樹, 山崎謙介. Web 上への手書きメモが共有可能なブラウザ PerowserEx の開発. 情報処理学会研究報告, 2009, 2009.
- [18] 浩成 野崎, 彩奈 吉橋, 恭子 梅田, 徹郎 江島. テキストへの自由な書き込み行為が文章理解に及ぼす影響. 日本教育工学会論文誌, 29:49–52, mar 2006.

SuprIME: IME によるテキスト編集機能の統合

中園 翔 増井 俊之*

概要. 計算機上で様々なテキストエディタが利用されているが、システムごとに編集方法や編集機能が異なっているのが不便である。本論文では、各国語入力のために OS に用意されている IME(Input Method Editor) 機能を利用することにより、あらゆるテキストエディタにおいて同じ操作によるテキスト編集を可能にする方法を提案する。我々の手法を利用すると、異なるテキストエディタ上での編集操作が共通化されるだけでなく、テキスト編集時に便利な様々な機能をあらゆるエディタで利用することが可能になる。

1 はじめに

計算機上でテキストを編集するために様々なテキストエディタが利用されている。文書を作成するときはワープロを利用し、メールを書くにはメールクライアントを利用し、文字端末でのプログラム開発には vim や Emacs を利用し、IDE を利用した開発では付属のエディタを利用し、ネット上でテキストを扱うにはブラウザのテキストフォームを利用するといったように、場合に応じて様々なエディタが利用されている。

エディタの機能や操作体系はエディタごとに異なっているのが普通である。ブラウザやワープロでテキストを 1 行消したい場合はマウスで行全体を選択してから削除キーを押せばよいが、vim では「d」キーを 2 回タイプして消すのが普通であり、Emacs では Ctrl-K キーが利用される。Emacs に慣れたユーザがワープロ上でも Ctrl-K で行を消去したいと思っても、そのようなカスタマイズはできないのが普通であるし、機能拡張が可能なシステムを利用している場合でも、操作体系を完全に同じにすることは難しい。

あらゆるエディタの操作を統一することは難しいが、様々なエディタで共通に利用できるソフトウェア層をユーザとアプリケーションの間に置くことができれば、異なるエディタの編集操作をある程度共通化できる可能性がある。現在のパソコンには IME(Input Method Editor) と呼ばれる文字入力機構が用意されており、様々な言語のテキスト入力に利用されている。IME はエディタなどとは独立したソフトウェアであり、ユーザのすべてのキー入力を受け取って各国語に変換した結果をアプリケーションに送出する。IME はあらゆるアプリケーションで共通に利用されるので、たとえば日本語入力用の IME を利用する場合、Emacs でもブラウザでも IDE でも同じ操作で日本語を入力できる。IME は一般には各国語

入力のみのために利用されているが、テキストの挿入/移動/削除といった編集操作も IME が受け持つようにすれば、様々なエディタ上で同じ操作で編集を行なうことが可能になると考えられる。

このような考えにもとづき、Mac 上の様々なテキストエディタにおいて同じキー操作によるテキスト編集を可能にする SuprIME システムを試作した。本論文では SuprIME の実装と利用方法について述べ、柔軟でユニバーサルなテキストエディタの構築について考察する。

2 SuprIME 使用例

2.1 日本語入力

SuprIME は、MacRuby¹ で記述された日本語 IME である「Gyaim²」に様々な編集機能を追加したものである。図 1 は SuprIME による日本語入力の例である。IME はアルファベット以外の文字を入力するときだけ有効にするのが普通であるが、SuprIME は、編集操作を行うために常に有効であることを想定して実装した。そのため、英語入力と日本語入力の両方が可能である。



図 1. SuprIME による日本語入力の例

Copyright is held by the author(s).

* 慶應義塾大学環境情報学部 環境情報学科, 慶應義塾大学環境情報学部

¹ Ruby を Mac 用に拡張したもので、あらゆる Mac の API を Ruby から利用することができる。

² <https://github.com/masui/Gyaim>

2.2 ブロック移動

テキストの一部を別の場所に移動する操作はテキストエディタの重要な機能のひとつであるが、エディタによって操作方法が大きく異なっている。たとえば Emacs でテキストを移動させたい場合は、移動する領域をキー操作によって指定してから削除/コピーし、カーソルを移動してからペーストするという手順を利用するのにに対し、ブラウザの編集領域でテキストを移動させたい場合は、マウスで領域を指定した後で選択領域をドラッグして別の位置に移動することが多い。このように、テキスト移動のような基本操作でもエディタごとに操作が異なっているのは不便であり、操作ミスをしがちであるが、SuprIME を利用するとあらゆるエディタにおいて同じ操作でテキストを移動することができる。

図 2 は Mac OS に標準搭載されている「テキストエディット」でテキストを編集しているところである。

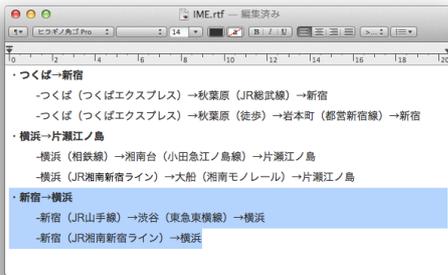


図 2. ブロック移動前の状態

ここで Shift+↑キーを押すとテキストは図 3 のように変化する。

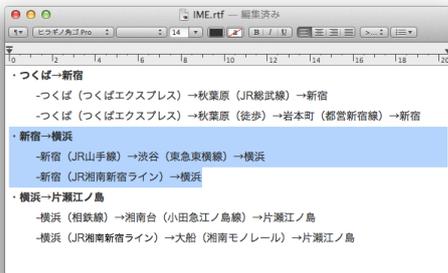


図 3. Shift+↑キーを押した後の状態

図 4 はブラウザ上で Google Docs のテキストを編集しているところである。ここで Shift+↓キーを二度押すと、テキストは図 5 のように変化する。



図 4. ブロック移動前の状態

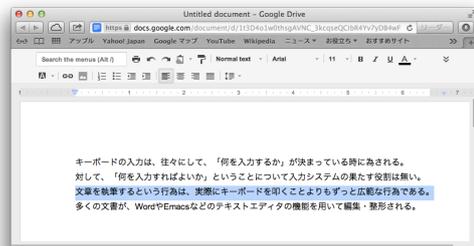


図 5. Shift+↑キーを押した後の状態

このように、あらゆるエディタにおいて Shift+矢印キーという共通の操作でブロック移動を行なうことができる。

2.3 連続インデント

前述したブロック移動は、インデントを調整する機能も備えている。プログラミングを行なっている時だけでなく、普通のテキストを編集している場合でも行頭の空白やタブの量を調整するインデント処理は頻繁に行われるが、自動的にインデントを行うエディタもあれば、コマンドを打ち込まなければならないものもあり、そのような調整機能を持っていないエディタも多い。SuprIME を利用すると、あらゆるエディタや入力フィールドでブロック移動と同時にインデントが自動で行われる。

図 6 は、Xcode 上で Python プログラムを書いている例である。Xcode は、Objective-C や Ruby などの言語で自動インデントの機能を備えているが、Python には対応していない。たとえば、図 6 のようなテキストに対して図 7 のように移動させたいブロックを選択し、Shift+↓キーを入力することで、テキストは図 8 のように変化する。

```

1
2 for i in range(1,101):
3   if i%15 == 0:
4     print 'FizzBuzz'
5   elsif i%5 == 0:
6     print 'Buzz'
7   elsif i%3 == 0:
8     print 'Fizz'
9   else:
10    print i
11
12 class FizzBuzz
13

```

図 6. ブロック移動前の状態

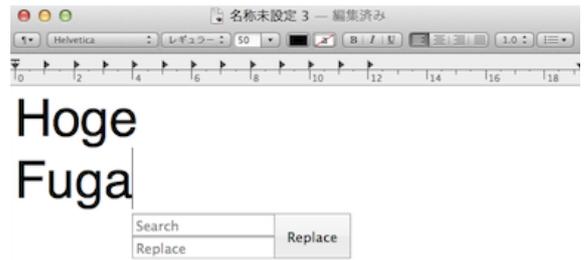


図 9. 単語置換ウィンドウの例

```

1
2 for i in range(1,101):
3   if i%15 == 0:
4     print 'FizzBuzz'
5   elsif i%5 == 0:
6     print 'Buzz'
7   elsif i%3 == 0:
8     print 'Fizz'
9   else:
10    print i
11
12 class FizzBuzz
13

```

図 7. テキストを選択した状態

図 10 のように、入力語と置換語をテキストエリアに打ち込み、Replace ボタンを押すと、

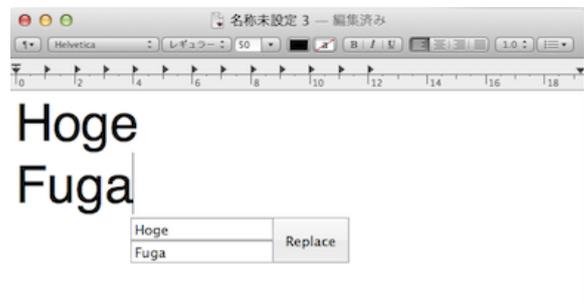


図 10. 置換ウィンドウに単語を入力した例

テキストは図 11 のように置換される。

```

1
2 class FizzBuzz
3
4   for i in range(1,101):
5     if i%15 == 0:
6       print 'FizzBuzz'
7     elsif i%5 == 0:
8       print 'Buzz'
9     elsif i%3 == 0:
10      print 'Fizz'
11     else:
12      print i
13

```

図 8. ブロック移動後の状態

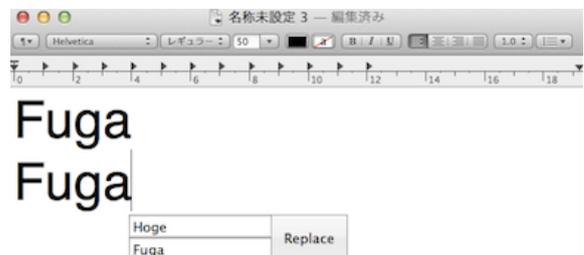


図 11. 単語置換後の状態

このように、あらゆるエディタにおいて、ブロック移動を行う際に適切な空白とタブを自動で補完することができる。

2.4 単語置換

多くのエディタが単語の検索/置換機能を持っているが、その操作方法はシステムごとに異なっている。例えば、コンソール上で検索と置換を行う時は `grep` や `sed` コマンドを用いるが、多くのエディタソフトでは `Control+F` や `Command+F` が用意されている。

SuprIME を利用することでこの操作を共通化する例を挙げる。

あらかじめ割り当てたファンクションキーを押すと、図 9 のように SuprIME の単語置換ウィンドウが起動する。

このように、あらゆるエディタ上で検索と置換が可能である。

2.5 Dynamic Macro

Emacs では lisp プログラムでエディタの機能を拡張することが出来るが、これらのスクリプトを IME 上に実装することで、他のエディタ上でも Emacs と同様の拡張機能を動作させることができる。

以下に SuprIME で Dynamic Macro[7] を実現した例を挙げる。Dynamic Macro は、入力 of 繰り返しを自動化する Emacs 拡張であるが、SuprIME ではこれと同様の機能を Ruby で実装している。図 12 のように、ユーザの入力操作が繰り返しになってい

るとき、任意に設定したファンクションキーを押すと、テキストは図 13 のように編集される。

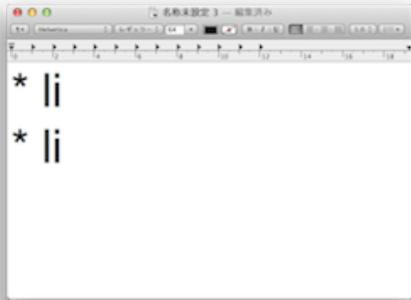


図 12. Dynamic Macro 機能を使用する前のテキスト

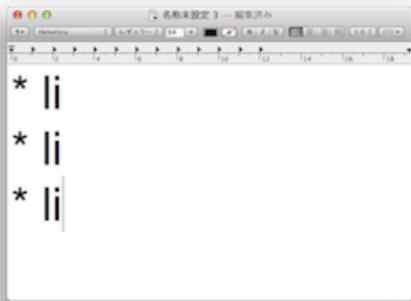


図 13. Dynamic Macro 機能を使用した後のテキスト

以下は SuprIME の Dynamic Macro 機能をブラウザのアドレスバー上で実行した例である。図 14 のように、ユーザの入力に”abc”が繰り返されているとき、任意に設定したファンクションキーを押すと、テキストは図 15 のように編集される。



図 14. Dynamic Macro 機能を使用する前のテキスト



図 15. Dynamic Macro 機能を使用した後のテキスト

Emacs 上に実装された Dynamic Macro は Emacs 上でしか利用できないが、SuprIME 上に実装された Dynamic Macro はあらゆるテキストエディタの上で利用することができる。

3 実装

SuprIME は MacRuby で記述された Mac 用の IME であり、ソースが 500 行程度とコンパクトであるにもかかわらず、他の IME に見られない機能を実装しており、本論文のような実験も容易である。

3.1 MacRuby による実装

MacRuby は、Mac OS 用のアプリケーションを開発するために拡張された Ruby 実行環境であり、Mac OS の Objective-C ライブラリを Ruby で扱うことができる。SuprIME では、Mac OS の IME フレームワークである InputMethodKit Framework を MacRuby から呼び出すことによって基本的な IME の機能を実装している。

3.2 修飾キーの扱い

エディタが編集機能のために用いる、Control や Command などの修飾キーは IME がハンドルしないのが普通であるが、SuprIME では、一部のファンクションキーと修飾キーの入力を受け取り、共通化されたテキスト編集操作を実現している。

SuprIME のように、修飾キーの一部をハンドルする実装を行った場合、修飾キーの入力はエディタに送出されないため、エディタが同じキー入力を編集操作の一部に割り当てている場合、それを利用することは出来なくなる。

3.3 テキストデータの取得

ブロック移動やインデントの処理といったテキスト編集の機能を、あらゆるテキストエリアで行うためには、テキストエリアに入力されている全文を IME が取得する必要がある。InputMethodKit はこの機能を備えていない。SuprIME では、ブロック移動やインデントのコマンドが入力された際に、Mac OS に実装されている、テキストエリアの全文を選択状態にするコマンドを AppleScript により送信している。

Mac OS では、多くのアプリケーションにおいてテキストエリアは NSTextField Class のオブジェクトとして実装されており、この NSTextField に対してテキスト編集や入力を行うための NSTextInput Protocol が存在する。このプロトコルには標準の実装として Command+A キーによる全文選択の機能が備わっている。

SuprIME では、このキー入力を AppleScript によりテキストエリアに送信することにより、テキストエリアの全文を選択状態にしている。SuprIME では、NSTextField の標準実装である Command+C キーによる全文コピーの機能を AppleScript により送信することで、Mac OS のクリップボードにテキストエリアの全文を保存し、それを InputMethodKit

Framework が読み込むことで、テキストエリアの全文を IME が取得している。

3.4 実装手法の問題点

しかし、このような手法で実装できる編集機能や、IME としてのパフォーマンスには問題がある。たとえば、対象とするテキストエリア自体が NS-TextField Class を利用していない場合や、対象とするテキストエリアがテキストの全選択やコピーに割り当てるキーバインドを変更している場合、テキストエリアの全文を取得することが出来ない場合があり、SuprIME の持つ編集機能は有効に機能しない。また、どこまでの編集機能をエディタやテキストエリアから切り離し、どれだけの拡張性を持たせることができるか、という点において、IME の層での実装には問題がある。各国語入力のための枠組みである IME は、テキストエリアよりも先にユーザのキー入力をハンドルすることができるが、エディタが想定するキー入力を横取りして、エディタの機能を打ち消す実装になる可能性がある。前述した Dynamic Macro の例のように、編集操作をマクロ化して行うことは多くあるので、IME とテキストエリアとの間に、テキスト編集のための枠組みを用意し、機能拡張が容易に可能な OS と API 設計のレベルでの実装の修正が求められる。

4 議論

4.1 キー操作の統一の必要性

Xerox PARC の Larry Tesler が 70 年代に発明した [10] 「コピー/ペースト」は現在広く普及しており、ほぼすべてのアプリケーションにおいて同じ操作³でテキストをコピーしたりペーストしたりできるようになっている。つまりコピー/ペーストに関しては 1 章で述べたような問題がほぼ存在せず、ユーザは混乱せずに様々なアプリケーション上でテキスト操作を行なうことが可能になっていることになる。コピー/ペーストの操作がこのように標準化されているのに対し、テキスト移動のような処理が標準化されていないのは問題であるが、それが大きな問題だと認識されておらず、ユーザが様々なエディタの使い方を覚えることをシステムが強制していることはさらに大きな問題だといえるだろう。SuprIME のような手法を利用することによってシステム全体を統一的に利用できるようにする工夫はまだ重要だと考えられる。

4.2 ユニバーサルな入力/編集

パソコンやスマートフォンで利用されている様々なテキスト入力システムは変換方式も使い勝手も全く異なるのが普通になっているが、POBox[6] のよ

うな単純で柔軟な入力方式を利用すると、パソコンでもスマートフォンでもほぼ共通の入力を行なうことが可能になる。Gyaim はこのような思想にもとづいて作成された IME であるが、Gyaim を拡張した SuprIME を利用すると編集操作も共通化することができるため、広い範囲の機器における入力と編集の共通化が期待できる。

4.3 高度な文書編集補助

SuprIME を利用すると、単純な文書編集作業が共通化できるだけでなく、より高度な編集補助を行なうことも可能である。たとえば、単語の言い換えを補助するために、図 16 のように類語を検索して入力することが出来る。

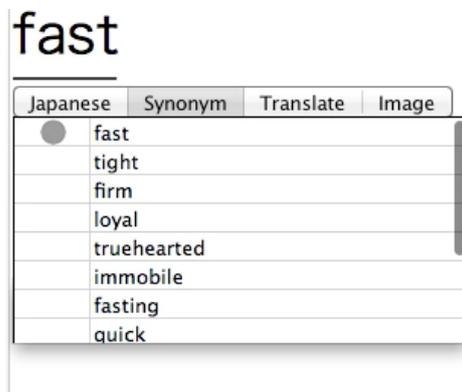


図 16. SuprIME の類語変換機能の例

図 17,18 は SuprIME の画像入力の例である。ここでは、「知らない」という入力語句について画像を検索し、入力候補として表示している。画像が埋め込めるテキストエリアであれば画像を埋め込み、そうでなければ URL が入力される。



図 17. SuprIME で「知らない」という単語を入力した例

³ Mac の場合は Command-C と Command-V



図 18. SuprIME の画像入力ウィンドウ

SuprIME を利用すると、このような高度なテキスト編者操作もあらゆるアプリケーションで共通に利用できる。

4.4 テキスト入力/編集以外への応用

本論文ではテキストエディタに絞った説明を行なったが、IME はユーザのすべてのキー入力を直接受け取る窓口になっているため、キー操作をテキスト編集と関係無い仕事に割り当てることも可能である。たとえばシステム音量や画面の明るさなども SuprIME から制御することが可能であり、システムで用意されたショートカット設定などが不要になる。

4.5 本手法の限界

IME は、テキストエリアが実装されたアプリケーションとは個別に実装されているため、現在の SuprIME の実装ではアプリケーションの内部状態によって動作を変えたり、アプリケーションの振る舞いを制御したりすることはできず、表に出ているテキストの編集操作しかできない。

既存のシステム自体は変更せずに、皮をかぶせる形で機能を拡張する手法はある程度有用ではあるが、問題の根本的な解決が必要な場合には限界がある。テキスト編集の場合は根本的に解決しなければならない問題は多くないので、本論文の手法はとりあえず有効だといえるが、根本的な解決のためには、テキスト入力の枠組みである IME に加えて、各コンピュータがテキスト編集操作を一元化するための枠組みを用意する必要があるだろう。

5 結論

OS に標準装備された IME 機能を活用することにより、様々なテキストエディタにおける入力/編集操作を共通化する手法を提案した。IME 機能はアプリケーションから独立しているため、アプリケーション内のデータを操作することはできないが、様々なアプリケーションにおける入力/編集操作の多くの部分を共通化することができた。テキストの入力や編集は計算機利用における最も重要な仕事のひとつ

であることは間違いないので、操作を簡単化/共通化するための優れた枠組みの開発は重要な課題だと考えられる。

参考文献

- [1] TextEditors.org. <http://texteditors.org/>.
- [2] P. H. Dietz, B. Eidelson, J. Westhues, and S. Bathiche. A practical pressure sensitive computer keyboard. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pp. 55–58, 2009.
- [3] C. Harrison, D. Tan, and D. Morris. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 453–462, 2010.
- [4] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong. The 1line keyboard: a QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 461–470, 2011.
- [5] I. S. MacKenzie, R. W. Soukoreff, and J. Helga. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 471–480, 2011.
- [6] T. Masui. An efficient text input method for pen-based computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pp. 328–335. ACM Press/Addison-Wesley Publishing Co., 1998.
- [7] T. Masui and K. Nakayama. Repeat and predict – two keys to efficient text editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pp. 118–123, 1994.
- [8] T. Murase, A. Moteki, N. Ozawa, N. Hara, T. Nakai, and K. Fujimoto. Gesture keyboard requiring only one camera. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, UIST '11 Adjunct, pp. 9–10, 2011.
- [9] J. Rick. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pp. 77–86, 2010.
- [10] L. Tesler. A personal history of modeless text editing and cut/copy-paste. *interactions*, 19(4):70–75, July 2012.
- [11] D. Wigdor and R. Balakrishnan. TiltText: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pp. 81–90, 2003.

CapStudio: プログラムの実行画面の録画映像を利用した統合開発環境

深堀 孔明 坂本 大介 加藤 淳 五十嵐 健夫*

概要. 通常、プログラム開発は、コードエディタ上でプログラムのソースコードを編集することによって行われる。しかし、ゲーム開発においては、キャラクタ画像やボタンの配置など見た目に関わる部分はゲームの実行画面を見ながらリアルタイムに編集できた方が直感的である。ゲームエンジンのシーンエディタといった、プレビュー画面上でパラメータを編集できるシステムもあるが、フレームワーク上で決められた制約があり、ゲーム画面の描画などのプリミティブな処理をプログラマが編集できず、コーディングの自由度が低くなるという問題がある。そこで、本稿ではゲーム画面の録画映像をベースとしたエディタ（以下スクリーンキャスト）を提案する。これをコードエディタと連携させることで、プログラマはプログラム全体をコードエディタで記述しつつも、ゲーム画面のデザインに関わる定数値やリソースファイルをスクリーンキャスト上でリアルタイムに編集できる。具体的には、定数値やリソースが編集されると即座にその結果がスクリーンキャストに反映される。反対に、スクリーンキャストのゲームオブジェクトから対応するコード片やリソースファイルへアクセスしたり、ゲームオブジェクトの位置やサイズを直接操作してソースコード中の定数値を調整できる。本稿では、スクリーンキャストを組み込んだ統合開発環境 CapStudio を実装し、本手法の可用性を検証した。

1 はじめに

ゲーム開発など、ビジュアルな要素を多く使用するプログラミング開発においては、キャラクタ画像やボタンの配置といった、見た目に関わる部分の調整は非常に重要である。キャラクタ画像は配色を微妙に変えるだけで大きく印象が変わるし、ボタンなどの GUI 部品の配置方法によってゲームの操作性は大きく変化する。しかし従来のコードベースの開発環境においては、このようなビジュアル要素の開発を支援する枠組みはほとんどなかった。このため、たとえばプログラムの編集中にはゲーム画面は表示されないため、キャラクタ画像やボタンの表示位置を修正するたびにプログラム全体をビルド・実行してゲーム画面を表示し、見栄えを確認する必要があった。

一方、Unity [9] などの近代的なゲームエンジンでは、プログラムの編集にゲームのプレビュー画面が常に表示される。プログラマはパラメータの修正結果をリアルタイムに確認したり、画面上のゲームオブジェクトを直接動かすことで位置パラメータを調節することができる。しかし、これらのシステムにはフレームワーク上で決められた制約があり、ゲーム画面の描画といったプリミティブな処理をプログラマが編集することができず、コードベースの開発環境に比べてコーディングの自由度が下がってしまうという問題がある。

本稿では、従来のコードベースの開発環境に対

Copyright is held by the author(s).

* Koumei Fukahori, Daisuke Sakamoto, Jun Kato, Takeo Igarashi, 東京大学

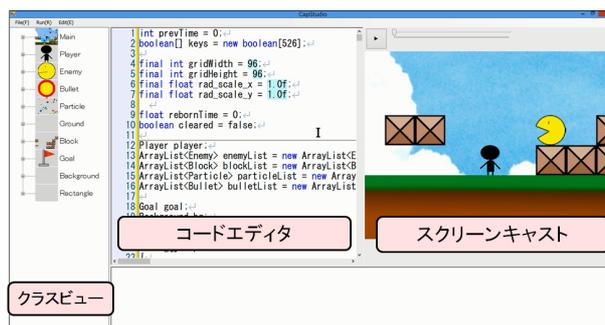


図 1. CapStudio のインターフェース。

してゲームエンジンのような対話可能なプレビュー画面を提供することを目的とする。そのためのシステムとして、対話可能なゲーム画面の録画映像であるスクリーンキャストを組み込んだ統合開発環境 CapStudio を提案する (図 1)。プログラムの実行中、CapStudio はゲーム画面をバックグラウンドで録画する。実行後、録画映像はスクリーンキャストとして表示される。スクリーンキャストは一般的な動画プレイヤーと同様に、映像を再生したり、再生時刻をシークバーで変更できる。

ただし、実際にシステムが記録する情報は、ゲームオブジェクトの描画位置や大きさといった描画履歴であり、システムはこれを用いて録画映像を生成している。また、この描画履歴を用いることで、スクリーンキャストとコードエディタを協調して動作させることができる。具体的には、プログラマがソースコードや画像などのリソースファイルを編集する

と、即座に編集結果がスクリーンキャストに反映される。反対に、スクリーンキャスト上のゲームオブジェクトを選択することで、対応するコード片やリソースファイルへアクセスできる。さらに、ゲームオブジェクトの位置や大きさを直接操作して、ソースコード中の対応する定数値を書きかえることができる。本稿では作成したシステムについて述べ、その実装を説明したあと、3つのサンプルプログラムを用いて本システムの可用性を議論する。

2 関連研究

本研究は2012年のBret Victorによるデモビデオ“*Inventing on Principle*”[1]に強く触発されている。このビデオでは、本システムのようにプログラムの実行画面を開発環境に埋め込み、コードエディタと連携させる手法が紹介されている。ただし、これはあくまでコンセプトビデオであり、システムの具体的な実装については言及されていない。一方で、本稿では、このような実行画面とコードエディタの連携機能を、スクリーンキャストを用いることで実現し、その具体的な実装を示している。

実行画面の録画映像をベースとしたインタフェースを持つシステムとして、Whyline for Java [5] というデバッグツールがある。Whylineは実行時に記録した実行履歴を解析し、プログラムの挙動の原因となるコード片を特定する。たとえば、プログラマが録画映像上のゲームオブジェクトを選択することで、そのオブジェクトの色や位置を決定したコード位置へジャンプすることができる。しかし、Whylineは本システムとは違い、ソースコードの編集を目的としたツールではない。そのため、コードの編集結果を即座に録画映像に反映したり、映像内のゲームオブジェクトを動かしてソースコード中のパラメータを書きかえることはできない。

プログラムを明示的に再実行せずにソースコードの修正結果を計算する手法として、バックグラウンドでプログラム全体を再ビルド・再実行して再計算を行う方法がある[3, 4]。しかしこの方法は、外部プロセスと通信を行うプログラムを開発する場合、暗黙的な再ビルド・再実行によって、意図せずに外部プロセスの状態を変えてしまう恐れがある。一方、本稿で提案する手法では必要な情報をすべて実行時に記録し、記録した情報のみを用いて再計算を行うため、予期しない外部プロセス・リソースの変更は起きない。また、プログラムの編集状態と実行状態を区別しないLive Programming言語[2, 6, 8]も、コード編集の結果を再実行せずに実行画面に反映することができる。しかし、上記の手法と同様に、プログラムの編集中に予期せず外部プロセスに影響をおよぼす恐れがある。

また、本システムともっとも関連のあるシステムとして、Unity [9] や Unreal Engine [10] といった

ゲームエンジンのシーンエディタや Visual Studio [11] などのGUIエディタがある。これらの開発環境では、プログラムの編集時、実行画面のプレビューが常に表示され、パラメータの修正結果をリアルタイムで反映したり、プレビュー画面上のゲームオブジェクトを直接動かして位置パラメータを調節できる。しかし、これらのシステムにはフレームワーク上で決められた制約があり、ゲーム画面の描画といったプリミティブな処理をプログラマが編集できないため、コードベースの開発環境に比べてコーディングの自由度が下がってしまうという問題がある。また、プレビュー画面内の時間の流れは不可逆で、本システムのスクリーンキャストと違い、シークバーを動かして時間を巻き戻すことはできない。

3 提案システム CapStudio

CapStudioの実行画面を図1に示す。プログラマはまずコードエディタにソースコードを記述し、その後、スクリーンキャストを利用してゲーム画面のデザインを修正する。

提案手法の最大のポイントは、ゲームエンジンのシーンエディタのようなパラメータ調整用のエディタを、コードベースの開発環境に対して提供することにある。一般に、描画処理を含めたすべてのソースコードがプログラマによって記述される場合、プログラムの挙動を静的に予測してプレビュー画面を生成することはできない。そこで本研究では、実行時に出力されたゲーム画面を擬似的なプレビュー画面に見立てる。まず、システムはゲーム画面の描画履歴を実行時に記録し、これを利用してゲーム画面の録画映像をスクリーンキャストとして生成する。さらに、記録した描画履歴を利用してスクリーンキャストとコード・リソース間の連携機能を実現する。

3.1 スクリーンキャストの生成

プログラマはまず、コードエディタに Processing 言語 [7] でソースコードを記述する。CapStudioのコードエディタは公式の開発環境 Processing IDE (PDE) と互換性がある。PDEで実行できるコードはCapStudioでも実行でき、その逆も真である。メニューバーの実行ボタンを押すと、プログラムがビルド・実行されゲーム画面が表示される。

プログラムの実行時、システムはバックグラウンドでゲーム画面の描画に関わった関数(以下、描画関数)の呼び出し履歴を記録する。たとえば、図2のようにゲーム画面上に画像がひとつだけ表示されるプログラムでは、ゲーム画面の大きさを指定する `size()`、画像データをファイルから読み込む `loadImage()`、画面をクリアする `background()`、画像を描画する `image()` のいずれかが呼び出されるたびに、そのコード位置、返り値、仮引数の式に含まれる変数の値といった関数呼び出し情報を記録する。これによ

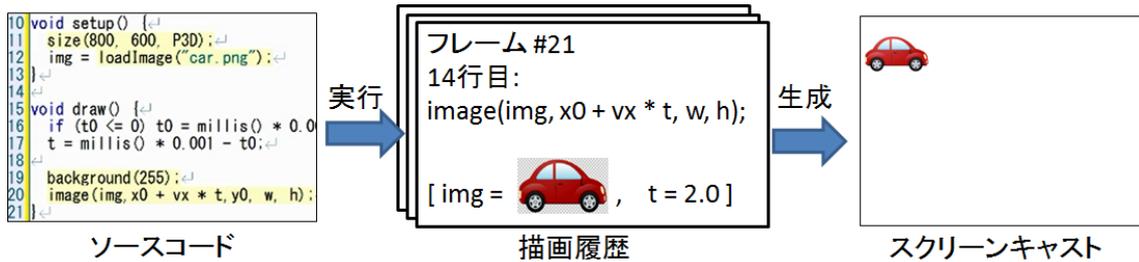


図 2. スクリーンキャストの生成

り、描画されたゲームオブジェクトの位置や大きさ、およびそれを描画したコードが特定できる。ただし、ゲーム画面の描画には Processig が公式に提供している API 関数だけが使われると仮定している。つまり、ソースコード中のどの関数が描画に関わっている、どの引数が位置や大きさを指定するのかという API の仕様はシステムがあらかじめわかっている。もし OpenGL や Java Swing など別の描画 API を使う場合は、各 API 関数の仕様を別途システムに組み込む必要がある。

プログラムの実行が終了すると、システムは記録された描画履歴から、ゲーム画面の録画映像がスクリーンキャストとして生成される (図 3)。スクリーンキャストは動画プレイヤーのように、再生ボタン (▶) を押して録画映像を再生したり、シークバーを動かして再生時刻を変更できる。

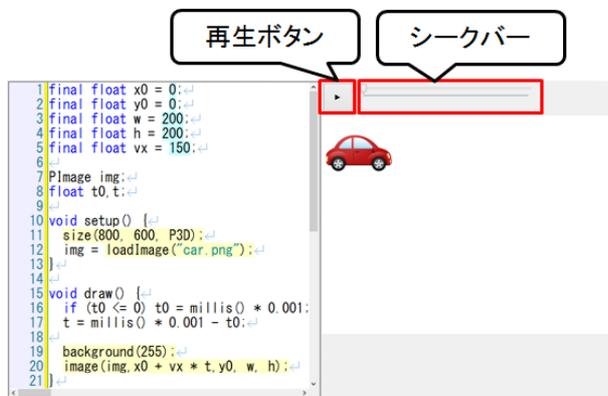


図 3. CapStudio で開発したプログラムの例。車の画像が水平方向に等速移動する。

3.2 コード・リソース編集結果のスクリーンキャストへの即時反映

プログラムの実行後、システムは、ゲーム画面のデザインに関わるソースコードをハイライトする。ゲーム画面のデザインに関わるソースコードとは、本稿では、1) 描画関数呼び出し文の引数、2) 定数宣言式の右辺、の 2 つを指す。たとえば、図 3 のコードでは、`loadImage()`、`image()` といった描画関数

の引数と、`x0`、`y0` といった定数の値がハイライトされる。プログラマがこれらのいずれかを書きかえると、システムは再度ソースコードの構文解析を行い、新しい引数式・定数値および前節で記録した描画履歴にしたがってスクリーンキャストを再描画する (図 4-a)。同様に、プログラマが画像などのリソースファイルを編集すると、システムは編集されたリソースを読み込み直して、スクリーンキャストを再描画する (図 4-b)。

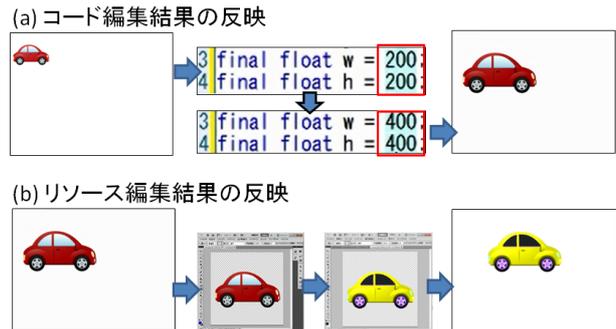


図 4. コード・リソース編集結果のスクリーンキャストへの反映。

3.3 スクリーンキャストを用いたコード・リソースの編集

スクリーンキャスト内のゲームオブジェクトはすべてクリック可能であり、クリックすることで対応するコードやリソースへアクセスできる。左クリックすると、それを描画した関数呼び出しが赤くハイライトされ、そこへカーソルが移動する (図 5-a)。右クリックすると、対応する画像ファイルを開くためのメニューが表示され、選択した画像ファイルが外部の画像編集ソフトで開かれる (図 5-b)。前述したように、これらのコードや画像ファイルが編集されると、即座に結果がスクリーンキャストに反映される。

また、スクリーンキャストからコード内の定数値を調節できる。スクリーンキャスト上のすべてのゲームオブジェクトは位置や大きさをドラッグ操作で変更できる (図 6-a)。こうしてゲームオブジェクトの位置・

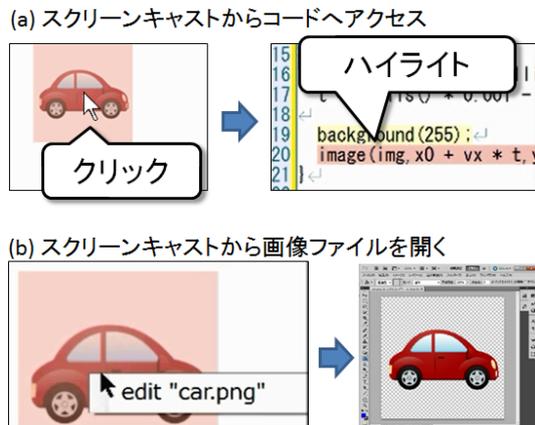


図 5. スクリーンキャストからコード・リソースへのアクセス

大きさを変更すると、同様の結果になるようにソースコードが自動で書きかえられる (図 6-b). デフォルトでは対応する関数呼び出し文の引数に定数値が加えられる. 書きかえられた引数を右クリックすると, 別の書きかえパターンが列挙される (図 6-c). 列挙された各パターンは, 引数の式に登場する定数を書きかえ対象とする. たとえば, `image(img, x0+vx*t, y, w, h)` という関数呼び出しによって描画された画像を, 左端の X 座標が 150 になるように移動したとする. ただし, `x0`, `vx`, `t` の値はそれぞれ 50, 25, 2 で `x0`, `vx` が定数である. このとき, まず第二引数に目的値 (150) と現在の値 (100) との差分 (50) が加算され, `rect(img, x0+vx*t+50, y, w, h)` と書きかえられる. その後, 式 `x0+vx*t+50` が右クリックされると, 定数 `x0`, `vx` を書きかえ対象としたパターンがそれぞれ「`final float x0 = 100`」「`final float vx = 50`」と表示される. たとえば前者のパターンを選択すると, 第二引数が書きかえ前の式 `x0+vx*t` に戻り, `x0` の宣言式が `final float x0 = 100` に書きかえられる.

ただし, 引数に現れる定数の内, 次の 2 つの条件を満たすものが書きかえ対象となる. 1) 描画関数の引数のみに影響を与える. 2) 定数がどんな値になるべきか計算可能. たとえば if 文の条件式に使われる定数は書きかえ対象にはならない. また, 現在の実装では, 引数式が定数に関する線形式の形になっている場合のみ目標値を計算できる. つまり, 引数の式が $a * c1 + b / c2$ で `c1`, `c2` が定数だとすると, `c1` のみが書きかえ対象となる.

4 実装

CapStudio は Processing 2.0 beta 8 の開発環境である. インタフェース部分は C# の WinForm アプリケーションとして実装されており, Windows 7 および Windows 8 で動作している. 以下, 前章で

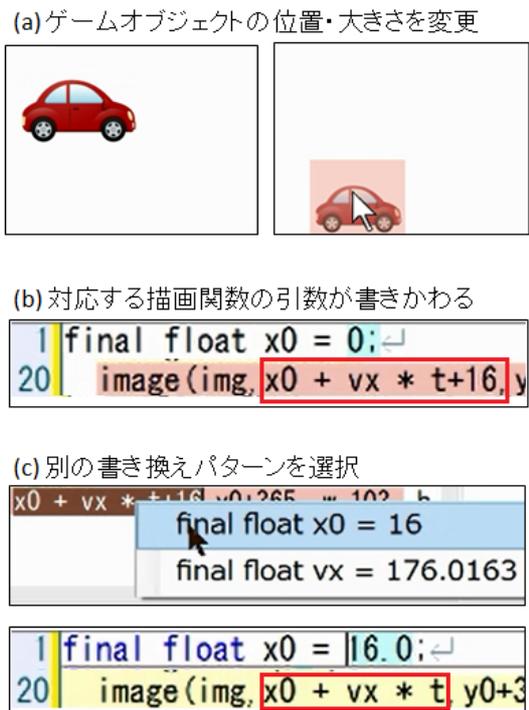


図 6. スクリーンキャストを用いたコード編集

紹介した機能について, 実装アルゴリズムを述べる.

4.1 描画履歴の記録・スクリーンキャストの生成

実行時に描画履歴を記録するため, プログラムがビルドされる前に, 描画関数の呼び出し文に対してコード書きかえが行われる. ソースコード中の描画関数呼び出し文の前後に, その呼び出し文のコード位置, 呼び出し時の返り値および引数の式に登場する変数の値を記録するコードが自動挿入される.

記録された描画履歴を用いて, 現在シークバーが示している時点 (フレーム) のゲーム画面を再現する. まず, プログラムの実行が終わると, システムは実行時に描画関数によって読み込まれたリソースファイルをすべて読み込む. 次に, コードエディタに記述されたソースコードを構文解析し, 定数の値と描画関数の引数式を抽出する. そして, 実行時に描画関数が呼び出されたときの引数値を, 引数式に定数値と実行時に記録された変数値を代入することで求める. その後システムは, 読み込んだリソースと求めた引数値を使って, 目的のフレームの間に呼び出された描画関数を順に再実行することで, スクリーンキャストを描画する.

4.2 コード・リソース編集結果のスクリーンキャストへの即時反映

プログラマーがコードやリソースファイルを書きかえると, スクリーンキャストは実行時に読み込まれたリソースファイルをすべて読み込み直す. その後,

ソースコードの構文解析を再実行し、定数の値と、描画関数の引数式を更新する。その後、新しい定数値・引数式を用いて、各時点における描画関数の引数値を再計算する。その後、現在のフレームで呼び出された描画関数を順に実行しなおして、スクリーンキャストを再描画する。

4.3 スクリーンキャストを用いたコード・リソースの編集

プログラマがスクリーンキャストをクリックすると、現在のフレームで発行された描画関数の呼び出しをすべて走査し、マウスカーソルと接触しているゲームオブジェクトを探す。見つかったら、それを描画した関数呼び出し文をハイライトする。あるいは引数にリソースオブジェクト (PImage 型の画像オブジェクトなど) が与えられている場合は、その元ファイルのファイルパスを表示する。

同様に、プログラマがスクリーンキャスト上のゲームオブジェクトをドラッグすると、まずすべての描画関数呼び出しを走査して、ドラッグされたゲームオブジェクトを探す。次に、マウスの移動量に応じてゲームオブジェクトの新しい座標・大きさを求める。その後、その位置・大きさに描画されるように対応する描画関数の引数の値を求め、現在の引数の値との差分を引数式に加算する。同時に、別の書きかえパターンについて、書きかえ対象とする定数の目標値を、その定数に関する一次方程式を解くことで求める。

4.4 スクリーンキャスト生成手法の応用

2章で解説したコード・リソース編集のための機能に加えて、スクリーンキャストの生成手法を応用して、次の3つの機能を実装している(図7)。1) あるクラスのメンバ関数の中で呼び出された描画関数のみを選択して再実行することで、そのクラスのサムネイル画像を生成する。2) 他のフレームで描画されるゲームオブジェクトを現在のフレームに重ねて描画することで、ゲームオブジェクトの軌跡を残像として表示する。さらに、3) ゲームオブジェクトの描画位置をずらすことで、視点を二次元的に移動させ、画面外に描画されていたゲームオブジェクトを表示させることができる。

5 サンプルプログラム

作成したシステムの可用性を検討するため3つのサンプルプログラムを作成した(図8)。1つ目の例ではゲーム画面のレイアウト、2つ目の例ではリソース編集における、スクリーンキャストの有用性を議論する。3つ目の例では、実行時に外部プロセス・リソースと通信するプログラムの開発において、本システムが有効な性質を持つことを示す。

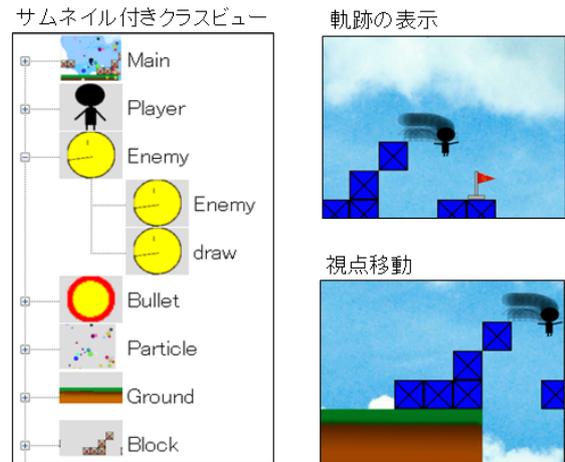


図 7. スクリーンキャスト生成手法の応用

5.1 ゲームのタイトル画面

図 8-a はゲームのタイトル画面に模した GUI プログラムである。タイトル・メニュー項目がそれぞれラベル・ボタンとして表示され、それぞれの位置と大きさはソースコード中で定数として指定されている。これらのオブジェクトを一度実行時に表示させてしまえば、あとはスクリーンキャストでオブジェクトをドラッグして位置・大きさを調節できる。もちろん、既存 GUI エディタやゲームエンジンでも、このように GUI 部品のレイアウトを直接操作することはできる。しかし本研究では、ソースコードがすべてプログラマによって書かれたプログラムに対して同様の操作を可能にしたことが重要であると考えている。

5.2 2D アクションゲーム

図 8-b は簡単な 2D 横スクロールアクションゲームである。ユーザはキャラクタを操作し、道中の敵を弾丸で倒しながらゴールを目指す。キャラクタや敵といったゲームオブジェクトはすべて、外部ファイルから読み込まれた画像を使って描画される。これらの画像を編集するためには、通常はファイルマネージャから画像を保存しているフォルダへ移動し、目的の画像を開く必要がある。一方、本システムでは、スクリーンキャスト上のゲームオブジェクトから直接目的の画像を開くことができ、ユーザにとっての負荷が大きく軽減されている。特に、弾丸のように一瞬しか現れないゲームオブジェクトは、再生時間を巻き戻せない映像内で捉えることは難しく、ゲームエンジンのシーンエディタを利用した場合でも、ゲームオブジェクトから画像を直接開くことは難しかった。しかし本システムでは、シークバーを動かして再生時刻を調整することで、そのようなオブジェクトもスクリーンキャスト上で簡単に捉えることができる。

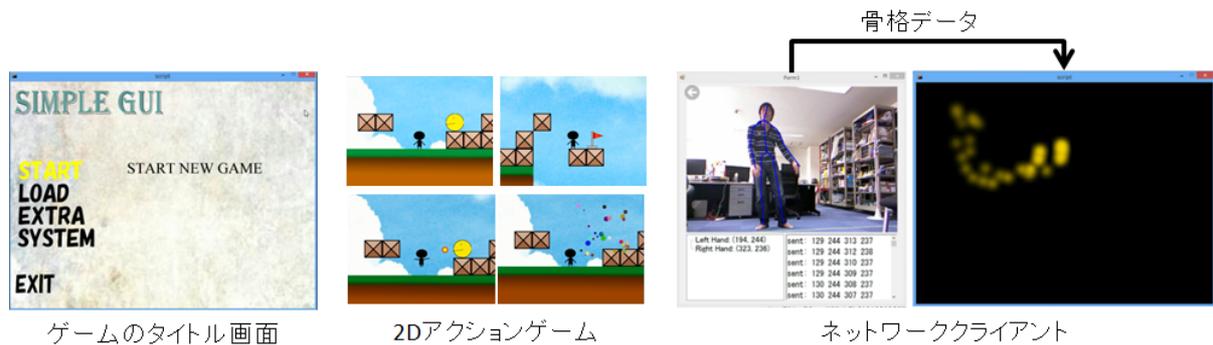


図 8. サンプルアプリケーション。

5.3 ネットワーククライアント

図 8-c はネットワーク通信を行う簡単なメディアアート作品である。Kinect から取得された骨格データを用いて、人間の手からパーティクルを放出する。ただし、Kinect の公式 SDK は Processing に対応していないので、骨格データは外部の C# プロセスに取得させ、TCP 通信で Processing プロセスへ送信している。プログラムの実行を終了すると、C# プロセスとのネットワーク接続が遮断され、Processing プロセスは新たな人間の骨格データを取得できなくなる。このとき、関連研究の章で紹介したバックグラウンドでプログラム全体を再ビルド・再実行する手法では、プログラマがソースコードを編集したときに正しく編集結果を計算できない。一方本システムは、実行時に骨格データの情報が描画履歴として記録されており、これを使って計算を行うため、コードの編集結果がスクリーンキャストに正しく反映される。

6 まとめと今後の課題

本稿では、対話可能なプログラムの録画映像であるスクリーンキャストを組み込んだゲーム開発のための統合開発環境 CapStudio を提案した。スクリーンキャストは従来のコードエディタと協調して動作し、プログラマは全てのプログラムコードをコードエディタで記述しつつも、ゲーム画面のデザインをスクリーンキャスト上で直感的に修正できる。

本提案の問題としては、スクリーンキャストがソースコードの特定箇所の変更結果しか反映できないことが挙げられる。現状の実現方法では、描画関数の呼び出し文と定数の定義式の変化のみトラッキングしている。したがって、ゲームオブジェクト同士の衝突判定といったロジックに関わるコードを書きかえても、スクリーンキャストには反映されない。本システムはあくまでゲームオブジェクトの配置や色合いといった、画面デザインの修正を支援することを目的としている。

現時点では、Processing の描画 API の内、特に現

在は、画像読み込み・画像描画といった、2D ゲームの開発に必要な描画関数のみに対応している。したがって今後は、3D モデルの描画を含めた Processing の描画 API のフルセットへの対応を予定している。それと並行して、スクリーンキャスト上のゲームオブジェクトを三次元的に操作するためのインタフェースも実装する。さらに、OpenGL や Java Swing といった、より低レベルな描画 API への対応も検討している。

参考文献

- [1] Bret Victor - Inventing on Principle. <http://vimeo.com/36579366>.
- [2] J. Edwards. Subtext: Uncovering the Simplicity of Programming. In *Proc. OOPSLA '05*, pp. 505–518, 2005.
- [3] B. Hartmann, L. Yu, A. Allison, Y. Yang, and S. R. Klemmer. Design as exploration: creating interface alternatives through parallel authoring and runtime tuning. In *Proc. UIST '08*, pp. 91–100, 2008.
- [4] J. Kato, S. McDirmid, and X. Cao. DejaVu: integrated support for developing interactive camera-based programs. In *Proc. UIST '12*, pp. 189–196, 2012.
- [5] A. J. Ko and B. A. Myers. Debugging reinvented: asking and answering why and why not questions about program behavior. In *Proc. ICSE '08*, pp. 301–310, 2008.
- [6] S. McDirmid. Living it up with a live programming language. In *Proc. OOPSLA '07*, pp. 623–638, 2007.
- [7] Processing. <http://processing.org>.
- [8] Welcome to Self - Self - the power of simplicity. <http://selflanguage.org/>.
- [9] Unity3D Game Engine. <http://unity3d.com>.
- [10] Game Engine Technology by Unreal. <http://www.unrealengine.com/>.
- [11] Microsoft Visual Studio. <http://msdn.microsoft.com/ja-jp/vstudio>.

順解析と逆解析を相互に利用する打楽器のインタラクティブなデザインインタフェース

山本 和彦 五十嵐 健夫

概要.

本稿では、打楽器の形状と音色のデザインを効率的に行うために、コンピュータによって物体形状からそれを叩いたときの音色を計算する順解析と音色からそれを再現可能な物体形状を推定する逆解析を交互に行い、その支援を行うインタフェースを提案する。本インタフェースでは、ユーザが物体形状を編集した場合には音色を、音色自体を編集した場合には対応する物体形状をそれぞれインタラクティブにユーザにフィードバックする。従来の打楽器を始めとする生楽器の設計では形状データの作成、解析による確認、出力による評価の3段階のプロセス間の非常に煩雑なトライ&エラーを繰り返す。これを効率化するために、形状モデリングと同時にその音を解析して、結果をインタラクティブにフィードバックするインタフェースも過去に提案されているが、提示された情報を基にユーザはさらにどのような操作を加えれば目的とするものに近づけていくことができるのか、ということ判断するのは困難であるという問題がある。本システムでは提示された音色情報自体をさらにユーザが編集してその影響を物体形状に反映させることを繰り返すことによってこの問題を解決する。本稿で提案するような順問題と逆問題を相互に解いてデザインの支援を行うアプローチは楽器のみならず多くの一般的な製品の設計においても役立つ可能性がある。

1 はじめに

現在、打楽器を含む多くの生楽器の設計プロセスは、形状モデリング、解析、出力/評価の3段階から成り立っている。このプロセスでは、まず最初にCADで形状データを作成する。その後、それを有限要素法等で解析してその音の情報(モード周波数、振幅)や実際にそれをものとして出力しても破綻が無いかなどを確認、最後にそれを実際に物体として出力して今度はその実際の音や演奏感を評価する。しかしながらこの3つのプロセスはそれぞれが乖離しており、プロセス間を横断したトライ&エラーは非常に労力がかかる。例えば、解析の結果が気に入らなかった場合には再び形状を修正し直してから再解析を行うということを繰り返す。また、解析プロセスを一度パスできたとしても、その後出力して実際の出音が所望のものでなかった場合には再び形状モデリングと解析のプロセスを繰り返すことになってしまう。これは非常に煩雑な作業であり、設計者はプロセス間を横断したトライ&エラーに労力が割かれてしまい、本来の創造的なデザインに集中することが困難になってしまっているという問題がある。

この煩雑さを解消するために、梅谷ら [1] は形状モデリングと同時にそれを叩いたときの音の基本周波数の情報を音としてユーザにフィードバックすることによって形状モデリングと解析のプロセスを統合した。しかし、これだけではフィードバックされ

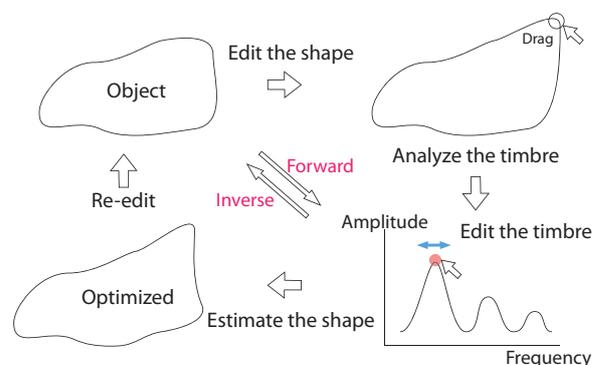


図 1. 順解析と逆解析を繰り返して形状と音色両面から打楽器をデザインする

た情報を基にユーザは、さらにどのような形状変形操作を加えれば目的とする音に近づけていくことができるのか、を判断することが困難であるというさらなる問題がある。

そこで、本稿では、打楽器の形状と音色のデザインを効率的に行うために、コンピュータによって物体形状からそれを叩いたときの音色を計算する順解析と音色からそれを再現可能な物体形状を推定する逆解析を交互に行い、その支援を行うインタフェースを提案する。本システムではユーザが物体形状を編集した場合には音色を、音色自体を編集した場合には対応する物体形状をそれぞれインタラクティブにユーザにフィードバックする。これを使ってユーザは、形状編集操作に応じて提示された音色情報自

体をさらに編集してその影響を物体形状に再び反映させることを繰り返すことによって、形状と音色両面から打楽器をデザインしていくことが可能である(図1)。

本稿で提案するような順解析と逆解析を交互に解いてデザインの支援を行うというアプローチは、ユーザが解析手法独特のノウハウ等の余計な問題に煩わせられることなく常にその時々で着目しているデザイン要素にのみ集中することができるようになるという利点がある。これは音だけではなく例えば剛性や熱伝導性能といった問題にも置き換えることで、楽器のみならず多くの一般的な製品の設計においても役立つ可能性がある。

2 関連研究

コンピュータによって物体の振動音を計算するための最も代表的な手法は、物体を有限要素法で離散化して固有値解析を行い、算出されたモード(固有振動)の線形和としてその振動を近似するものである[3]。固有値解析は非常に計算コストが高く一般的には前計算としてのみ行われる処理であり、インタラクティブな速度で計算を行いデザインツールに利用するためには高速化のアルゴリズムが必要である。

梅谷ら[1]は求める固有値を最低次の固有値のみに限定することで、ユーザが物体形状をモデリングすると同時にインタラクティブな速度で固有値解析を行い、その結果をフィードバック可能にした。この研究ではユーザが鉄琴の形状モデリングを行うと同時にその基本周波数の情報を音としてフィードバックすることによって、複雑な形状の鉄琴であっても容易にその音高を調整することができる。しかし、提示される情報が基本周波数のみであり、楽器としての評価を行うためには不十分であった。C.B.Maxwellら[4]は固有関数が既知である物体形状からの微小な変形による影響をモード空間に射影して自由度縮小を行うことで準インタラクティブな速度で基本周波数だけではなく高次の周波数成分も計算しており、音色の評価をすることも可能である。しかし、元の形状からの変形が大きくなると計算が破綻してしまうという問題がある。

コンピュータによるインタラクティブな物理シミュレーションをデザインツールに利用する研究は近年盛んに行われているが[6][13]、そのほとんどが順解析か逆解析のどちらかを単独で利用するものである。順解析とは、原因を与えてそこから結果を計算する問題であり、物体の運動シミュレーション等多くの問題がこれに属する。その逆が逆解析であり、結果から原因を推定する問題である。本研究で扱う音色からの物体形状の推定もこの逆問題に属する。

Design by Dragging[5]は順解析と逆解析両方を同時にデザインに利用できるように一つの環境に統合した最初の研究である。この研究では医療機器の

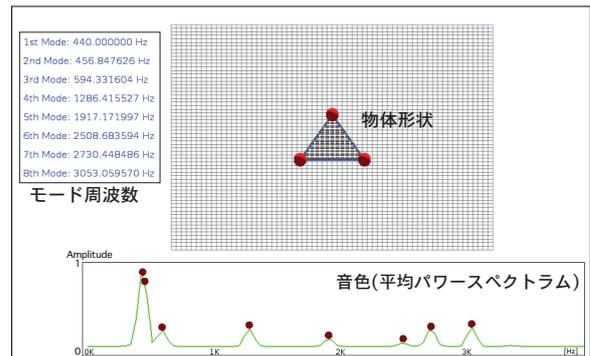


図 2. プロトタイプ画面

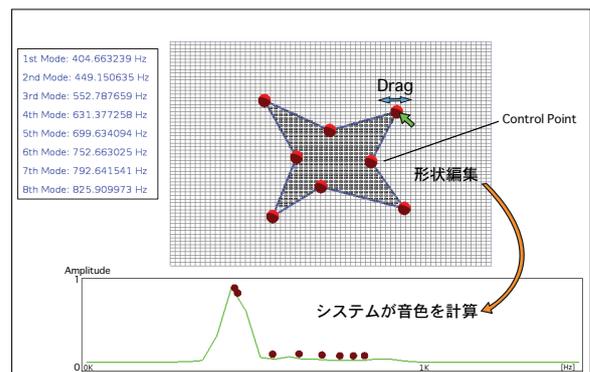


図 3. マウスによる物体形状モデリング

形状を予め細かくパラメータ化しておき、前計算したシミュレーション結果の中からユーザが現在編集している状況に近いものを検索、補間することで、ユーザが物体形状を編集した際には構造への負荷が、結果として表示された構造への負荷状況を編集した場合にはそれによる形状への影響がインタラクティブにフィードバックされる。これによってデザイナーは解析ノウハウを一切気にするこの無く本来のデザイン作業に集中できる。しかし、予め細かくパラメータ化された形状は編集の自由度に乏しく、デザイナーはツール制作者の想定を超えるような形状を作ることができないという問題がある。

3 ユーザインタフェース

図2が本研究で提案するインタフェースのプロトタイプ画面である。これによって打楽器の形状と音色を編集する。中央に編集する打楽器の形状が表示してあり、左側に現在の形状でのモード周波数を低周波数から順に表示、下側には実際に音色を鳴らしたときの音のパワースペクトラムの時間方向の平均値およびその編集画面が用意されている。パワースペクトラムは横軸に周波数(リニアスケール)、縦軸に正規化されたパワーを示している。横軸は一番低い周波数のピークと高い周波数のピークが操作しや

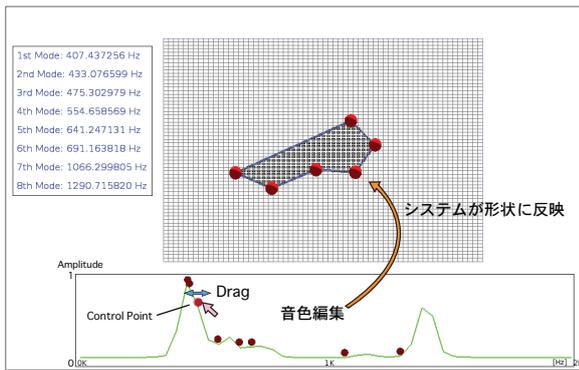


図 4. 周波数領域での音色編集

すいように全体がスケージングされている。

ユーザは中央の画面において物体の周囲の制御点(赤い点)をマウスでドラッグして2次元的に物体形状を変形する(図3)。また、制御点を結ぶ辺の中間をクリックすることで物体周囲の任意の位置に制御点を追加することも可能である。ユーザが物体形状を変更するとそれを叩いたときの音色が再計算され、モード周波数とパワースペクトラムの表示が更新される。一方、下側のパワースペクトラム表示画面には周波数軸上で現在の形状でのモード周波数の位置に制御点(赤い点)が設けられており、ユーザはこれをマウスでドラッグすることによりモード周波数の位置を移動させて音色自体を編集することができる(図4)。音色が編集されるとシステムは現在の形状からできる限り近い形状でその音色を再現できるような形状を計算して画面中央に表示する。音色編集はあまりにも大幅な変更を許してしまうと計算が収束しないか時間がかかってしまうので現在のモード周波数の近傍に編集範囲を制限する。ユーザはこの形状操作と音色操作の2つの操作を繰り返して最終的な目標の形状と音色を同時に作成していく。

ここで、ユーザにとってモード周波数を直接操作することは音色操作の方法として決して最適では無い。しかし音色と形状を結びつけるパラメータであるモード周波数を操作することは、提案手法を実現する上で最も直接的であるためこの方法を採用した。また、モード周波数の位置はパワースペクトラムのピーク位置近傍となる場合が多いが、減衰が早いモードではピークとして表れてこない場合もあり、ユーザにとって操作しにくくなってしまいうこともある。この点については今後の改善が必要である。

さらに、音色の確認についてはモード周波数やパワースペクトラムを視覚的に表示するだけでは不十分であり、実際に音として耳で聴いて評価する必要がある。しかし、楽器としてその出音が音楽的かどうかという評価を行うためには規則的に音を鳴らしたり、マウスでボタンをクリックしたときに音を鳴らす、というだけでは不十分であり、実際にユーザ



図 5. テーブルをあたかも打楽器のように叩くことで現在編集している音色を鳴らして確認する。

がその音を使って音楽を演奏して評価するのが妥当である。そこで本研究では Possessing Drums[10]の手法を音色フィードバックの手段として採用する。これは音響信号処理によって、マイクから入ってきた音(例えばテーブルを叩いた音)から駆動音(テーブルの振動特性を差し引いたもの)を抽出した後に、任意の音響伝達特性を再現できるような共振器に通して音色変換をすることによって身の回りの物体に任意の音を割り当てる研究である。これによって現在編集している物体の伝達特性(それぞれのモード周波数を持つ共振器)にマイクからの音を通すことで、PCを使って設計を行っているユーザは目の前のテーブルを叩いてシミュレーションされた音色で音楽を試奏することでその評価を行うことが可能となる(図5)。

4 アルゴリズム

4.1 音色のシミュレーション

物体の振動音をシミュレーションするためには有限要素法で線形弾性体として離散化して固有値解析を行うのが一般的である。しかし、従来固有値分解によって最低モードだけではなく高次モードも含めて求めることは非常に計算コストが高く、インタラクティブな計算に利用することは困難であった。そこで本研究では部分構造合成法[2]を応用した前処理付きの高速な固有値解析手法を提案する。

有限要素法で離散化された自由度 N の固有値問題は次のような一般化固有値問題に帰着することができる。

$$K\phi = \lambda M\phi \quad (1)$$

K は $N \times N$ の剛性行列、 M は質量行列、 λ は固有値、 ϕ は固有ベクトルである。ここで任意の数 $n(n \ll N)$ だけ固有ベクトルを並べた行列 U によって $K' = U^T K$ 、 $M' = U^T M$ 、 $\phi' = U\phi$ と定義すると式1は、

$$K'\phi' = \lambda M'\phi' \quad (2)$$

と変形することができ、本来の問題に比べて非常に小さな自由度の簡単な問題として解くことができる。

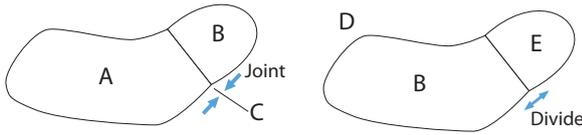


図 6. 部分構造の接合 (左) と分割 (右)

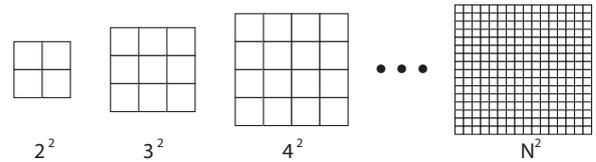


図 7. 前計算を行うグリッド

この問題を部分構造合成法を使って解く。

部分構造合成法 [2] は解析対象の構造体を複数の分系に分割し、分系ごとに自由度縮小して重ねあわせることによって高速に解析を行う手法である。その中でも特に差分モード法 [7] ではある分系の固有関数が既知であった場合にその分系に関しては自由度縮小を行ったまま他の分系と合成することで計算コストを大幅に抑えることができる。

まず、図 6 左のように物体 A に物体 B を接合し、その共有する節点群を添字 C で表すとする。ここで、分系 A の固有関数 U_A と分系 B の固有関数 U_B がそれぞれ既知であるとする、全体の剛性行列 K'_{total} は自由度を縮小することができ、AB 全体の構造を一度に計算するより大幅に省メモリかつ少ない計算コストで処理することができる。

$$K'_{total} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & U_{AC}^T K_{CC} U_{AC} & U_{AC}^T K_{CB} U_{BC} \\ 0 & U_B^T K_{BC} U_{AC} & U_B^T K_{BB} U_B \end{bmatrix} + \begin{bmatrix} U_A^T K_{AA} U_A & U_A^T K_{AC} U_{AC} & 0 \\ U_{AC}^T K_{CA} U_A & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

各行列の添え字は節点の属する分系を表す。また、 U_{AC} は分系 A の固有関数の C 節点要素、 U_{BC} は分系 B の固有関数の C 節点要素である。質量行列に関しては同様の形となるため省略する。

本研究では、図 6 左における部分構造 A はユーザが編集する前の固有関数が既知の形状、B をユーザが編集した後に加わった部分形状と考えることができる。しかし新たに付け加えた形状 B の固有関数は未知であるためこの方法をそのまま利用することはできない。そこで本研究では未知の構造 B の固有関数を高速に求めるために前計算法を導入する。

まず、前処理として計算領域全体を 2 次元の場合には正方形格子の有限要素法で離散化する。これは 3 次元の場合には立方体格子となる。あらかじめ十分な大きさ N をとり、 $2^2, 3^2, 4^2, \dots, N^2$ 節点の大きさの正方形形状それぞれにおいて十分な数の固有ベクトルを求めておく (図 7)。ここで、差分モード合成法 [7] では図 6 右のように固有関数 U_D が既知である構造 D から構造 E を取り除いたときの構造 B の自由度縮小された剛性行列 $K_{B'}$ は、

$$K_{B'} = U_D^T K_B U_A = U_D^T K_D U_D - U_D^T K_E U_D \quad (4)$$

と表すことができる。

つまり、図 6 右における未知の構造 B を覆う構造 D の特性が分かっているならば構造 B の自由度縮小さ



図 8. 変更後の形状は、変形されていない部分 + (変形された部分を覆う正方形領域 D - 変形された部分を除きかつ変形されていない部分を除く D 内領域 (青色)) となる。

れた行列 $K_{B'}$ を構造 B 単独の固有値分解を行うことなく求めることができる。この未知の構造 B を覆う構造 D として前計算した正方形領域の結果を使うと、図 8 のように変形後の形状は、変形されていない部分 + (変形された部分を覆う正方形領域 D - 変形された部分を除き、かつ変形されていない部分を除く D 内領域 (図 8 の青色部分))、となる。この第一項は既知であり、第二項も差分モード合成法の分割処理で計算すれば固有値分解は不要であるのでこれは最終的に自由度が大幅に縮小された後の全体の式 3 の固有値分解を一度解くだけで計算が完了する。これによって従来より高速な固有値解析が可能となる。

ここで、計算精度に関してはそれぞれの部分構造においてどれだけ自由度を残しておくかに依存する。これはユーザが形状をどれだけ大きく変更したかということと計算コストのトレードオフで決定する。

4.2 所望の音色を実現するための形状最適化

いくつかのモード周波数、つまりは与えられた音色からそれに近いモード周波数を実現するような形状を推定する手法について説明する。

本研究では、ユーザが現在編集している形状からできる限り近い形状で所望のモード特性が得られる解をレベルセット法による形状最適化で求める。レベルセット法 [8] とは二つの相で構成される閉じた領域においてそれぞれの相の形態と形状を表現する方法である。本研究の場合は物体表面からの距離関数 $\phi(x)$ がこれに相当し、物体の内部で正、外部で負の値を取り、表面で零になる。レベルセット法に基づく構造最適化手法 [11] は、トポロジー最適化法に代表される他手法 [12] で問題となる形状の波打ち問題やグレースケール (物体と空洞の中間領域) 問題が発生せず、計算コストも少ないという利点がある。ここで、通常、構造最適化問題では、得られる構造が

初期構造に大きく依存してしまうことが問題として挙げられるが、本研究ではユーザが現在編集している形状から最も近い最適形状を得ることを目的としているためにこの点はむしろ利点となる

最適化はレベルセットの関数である目的汎関数 $F(\phi(x))$ を最小化することによって行う。線形弾性体の固有モード最適化問題においては目的汎関数は、現在の固有値と目的とする固有値間の残差となる [11].

$$\text{Minimize } F(\phi) = \sum_{k=0}^{s+1} \left\{ w_k \left(\frac{\lambda_k - \lambda_{target,k}}{\lambda_{target,k}^2} \right) \right\} \quad (5)$$

ここで λ_k は k 番目の固有値, $\lambda_{target,k}$ は目標とする k 番目の固有値, w_k は重みパラメータである. s は目標とする固有値モードの数であるが, $s+1$ まで考慮しているのは、最適化の過程で固有ベクトルの形状が変化して次数が入れ替わってしまうことを防ぐためである. この最適化はレベルセット境界に適切な法線方向速度参照領域 $V_N(x, t)$ を与えて陽解法で移流させることを行う.

レベルセットの時間発展を表すレベルセット方程式の弱形式は参照領域 D において

$$\int_D \frac{\phi(x, t + \Delta t) - \phi(x, t)}{\Delta t} \Psi d\Omega + \int_D V_N(x, t) \Psi d\Omega = 0 \quad (6)$$

となる. ここで, Ψ は試行関数である. 形状最適化問題においては左辺第二項は目的汎関数のレベルセット関数に関する変分

$$\begin{aligned} \int_D f(\phi(x)) \psi d\Omega &= \int_D \sum_{k=0}^{s+1} \left\{ 2w_k \left(\frac{\lambda_k - \lambda_{target,k}}{\lambda_{target,k}^2} \right) \right. \\ &\quad \times (\epsilon(u_k) : \mathbf{D} : \epsilon(u_k) - \lambda_k \rho u_k \cdot u_k) \\ &\quad \left. \times \delta(\phi(x)) H(\phi(x)) \right\} \psi d\Omega \quad (7) \end{aligned}$$

で表される. この $f(\phi(x))$ は目的汎関数の形状感度の被積分関数であり, ϵ はひずみテンソル, u_k は k 番目の固有ベクトル, \mathbf{D} は弾性テンソル, $H(\phi(x))$ と $\delta(\phi(x))$ はレベルセット関数が $-h \leq \phi \leq h$ (h は任意の実数) の領域で値を持つ Heavisible 関数と Dirac Delta 関数 [11] である. 式 6 は有限要素法で離散化して,

$$\phi^{t+\Delta t} = \phi^t - \Delta t \mathbf{E}^{-1} \mathbf{V}_N^t \quad (8)$$

$$\mathbf{E} = \bigcup_{V_e} \int_{V_e} \mathbf{N}^T \mathbf{N} d\Omega \quad (9)$$

$$\mathbf{V}_N^t = \bigcup_{V_e} \int_{V_e} f(\phi(x, t)) \mathbf{N} d\Omega \quad (10)$$

に従って目的汎関数が収束するまで各節点のレベルセット関数を更新することで解く. ここで, \mathbf{N} は形状関数, \bigcup は要素の重ね合わせを表す.

アルゴリズムの手順としてはユーザが現在編集している固有関数が既知の形状を初期条件とし, 式 10 にてレベルセットを更新する. 更新の度に 4.1 章の

方法によって固有値解析をやり直し, 現在の固有値と固有ベクトルを設定し直す. また, レベルセット境界を更新すると, 境界から離れたところのレベルセットの値が無効になるので Fast Marching Method [9] 等を用いて全体を再初期化する必要がある.

ここで, レベルセット法では物体形状を距離関数によって陰的に表現しているため, 計算が収束した後にユーザがさらに形状を編集できるようにするためにはレベルセット法から明示的に物体表面のメッシュを生成してやる必要がある. 本研究では陰関数からのメッシュ生成には Marching Cube 法 [14] を用いた. さらに Marching Cube 法で生成されたメッシュはレベルセット格子と同程度の解像度となっているために細か過ぎてユーザが頂点を操作しにくくなってしまいうという問題があるがスムージングを行い, 荒いメッシュに近似することでこれを解決した.

5 結果

5.1 システムの速度と精度

本稿で提案する手法で形状変形を行いながらモード周波数を算出していった場合と, それと同じ形状で従来の固有値解析を行った場合の節点数とおよその処理時間の関係を表 1 に示す. 括弧内の数字は変形前後で追加された節点数である. 評価に使ったシステムは, CPU: Intel Core i7 Quad 3.4GHz, RAM: 32GB で C++ で実装し, 各形状において最低次から 8 つの固有周波数を求めた. 既存手法では節点数の

表 1. 固有値解析の時間 [ms]

節点数 (追加節点数)	既存手法	提案手法
6588 (1314)	2000	400
9026 (2475)	18000	840
13576 (3081)	37000	1030

増加に応じて計算時間が大幅に増加しているが, 提案手法では僅かな増加で済んでいる.

一方, 形状最適化の精度に関しても評価も行った. 最低次から 8 つのモードを対象として目標値を定め, 初期状態から最適化を行った結果が表 2 である. 結果の 6 次モードに新たなモードが入ってきており, 次数が変わっているが, 目標とする固有周波数群自体はほぼ得られていることが分かる. 8 次の目標値も達成するためには最適化の対象にするモードの数をさらに増やす必要がある.

5.2 エンジニアによる試用

本システムを実際に CAD や解析ソフトウェアの使用経験のあるエンジニア数名に試用してもらった後, インタビューを行った. 被験者には自分の好みの打楽器を作ることを想定してデザインを行うよう

表 2. 固有周波数最適化の結果 (単位 [Hz])

	初期状態	目標値	結果
1st	440	409	410
2nd	674	738	764
3rd	758	967	955
4th	1119	1367	1312
5th	4305	4776	4633
6th	8656	9500	5342
7th	9912	10649	9408
8th	11279	12567	10610

に指示した。

彼らは普段の設計では既に過去に実績のある CAD データを雛型として、それに編集を加えていくことで新しい製品を作っていくことが多いとのことである。そのため、本システムのアプローチは目的となる音色側からも操作を加えていくことができると楽器音に不快な成分が含まれていた場合に抑えるのが容易になったり、または新たな楽器音を探していく際にも非常に便利である、と意見を述べた。また、逆解析の結果得られた形状そのものを再びインタラクティブに修正していけるため、普段使っている解析ツールにも本システムが取り入れられると作業が容易になるとの意見を得た。

しかし、実際の楽器は対称な形状のものがほとんどであるので、対称性を前提にしたインタフェースにしたほうがよいとの意見も得られた。さらに、生楽器では形状を自由に設計するという需要はあまり無く、例えばスピーカ筐体等の設計に利用する方がより有用であるとの意見も多かった。これらは今後の課題とする。

6 まとめと今後の課題

本稿では、物体形状からそれを叩いたときの音色を計算する順解析と音色からそれを再現可能な物体形状を推定する逆解析を相互に行うことで形状と音色両面からの効率的な打楽器デザインを行うことのできるインタフェースを提案し、実際にエンジニアにも試用してもらうことでその創造的なデザインに役立つ可能性があることを確認した。

しかし、本システムの有用性を評価するためにはソフトウェア上での評価だけでは不十分であり、今後は実際に作成した打楽器を実物として出力して評価を行っていく必要がある。さらに、実際の楽器では数多くの部品が組み合わされているため、本稿の手法だけでは実用的とは言えない。複数の部品の組み合わせを考慮することは今後の課題である。また、音色からの形状推定ではユーザの操作次第では物理的に実現不可能な音色が設定されてしまうことがあ

るという問題もある。このため予めユーザに編集可能な範囲をシステム側から提示できることが好ましいが、最適化を行う前の段階ではその音色が実現可能かどうかを知ることは難しく、今後の課題である。

参考文献

- [1] 梅谷信行, 高山健志, 三谷純, 五十嵐健夫. 実時間固有値解析による対話的な鉄琴のデザイン. 情報処理学会論文誌, Vol.52, No.4, pp. 1599–1607, 2011.
- [2] 長松昭男, 大熊政明. 部分構造合成法. 培風館, 1991.
- [3] Changxi Z., and Doug L. J. Toward High-Quality Modal Contact Sound. In *ACM Transactions on Graphics*, 30(4), 2011.
- [4] Cynthia B. M., and David B. Modal Parameter Tracking for Shape-Changing Geometric Objects. In *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07)*, 2007.
- [5] Dane C., Chi-Lun L, Arthur G. E., Daniel F. K. Design by Dragging: An Interface for Creative Forward and Inverse Design with Simulation Ensembles. In *Transactions on Graphics* 27(3), 2008.
- [6] Greg S., Manfred L., Mitani J., and Igarashi T. SketchChair: An All-in-one Chair Design System for End-users. In *the fifth International conference on Tangible, Embedded and Embodied Interaction*, pp. 260, 2011.
- [7] Mochizuki T., and Ogawa M. Efficient Approach for Structural Modifcation Analysis. In *Review of Automotive Engineering*, Vol.27, No.1, pp. 75–82, 2006.
- [8] Osher S., and Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces. In *Springer*, 2002.
- [9] Sethian J. A. Level Set Methods and Fast Marching Methods. Cambridge Monographs on Applied and Computational Mathematics. In *Cambridge University Press*, 2nd edition, 1999.
- [10] Yamamoto K. Possessing Drums: An Interface of Musical Instruments that Assigns Arbitrary Timbres to Personal Belongings. In *Journal of Information Processing*, Vol.21 No.2, pp. 274–282, 2013.
- [11] Yamasaki S., Nishiwaki S., Izui K., and Yoshimura M. A structural optimization method based on the level set method using a new type of re-initialization scheme. In *International Conference on Computational Methods*, pp. 260, 2007.
- [12] Yang R. J., and Chung C. H. Optimal Topology Design using Linear Programming. In *Computers and Structures*, Vol.53, pp. 265–275, 1994.
- [13] Umetani N., Igarashi T., and Niloy J. M. Guided Exploration of Physically Valid Shapes for Furniture Design. In *ACM Transactions on Graphics* 31(4), 2012.
- [14] W.Lorensen, H.Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Computer Graphics*, 21 (4), pp. 163–169, 1987.

SensingFork：フォーク型センサを用いた食行動改善手法の研究

門村 亜珠沙 李 争原 陳 嚴章 塚田 浩二 朱 浩華 椎尾 一郎*

概要. 『ピーマンも食べてね』と小さな子供に言葉で説得するのは難しい。もし、嫌いな食材を食べたときや夕ご飯を全部食べたときに、やる気の出るフィードバックを返したら、好き嫌いの多い子供の食行動も楽しく改善できるかもしれない。そこで我々は、子供の食行動の改善を目的に、食行動（食事中の動作と大まかな食材の種類）を検出するフォーク型デバイス：センシングフォークを開発した。本論文では、センシングフォークのプロトタイプ設計と実装、このセンシングフォークと連動し、子供の食習慣改善を目指すスマートフォン用アプリケーション：腹ペコパンダ、および、本システムの性能評価について述べる。

1 はじめに

バランスの良い食事は健康を維持する上で重要な要因である。現在、世界各国で食事の摂取方法・食材の選び方・食に関する文化など、広い視野から食について教育すること、つまり食育が重要な課題となっている。特に、先進国では、政府がwebサイト上でも食にまつわる多くの情報を公開するなどして、食育を推進している^{1 2}。日本でも、2005年に、バランスの悪い栄養摂取や野菜摂取量の不足などの不健康な食習慣の増加を危惧し、人々が活発で健康的な生活を楽しむことができるように、食育基本法が制定された³。その結果、現在では、多くの幼稚園では食育活動を行っており、例えば、教諭らは子供たちに様々な種類の食材を食べることを促している。しかし、食育活動を引き続き家庭内で行うのは、時間・知識不足などの理由から親子にとって難しい[11]。そこで、我々は小さな子供にとって身近な食器であるフォークに着目し、家庭内での親子の食育支援を行うシステムを構築する。フォークのような身近な日用品をHCI技術で拡張することで、子供たちが楽しく食習慣を自然に改善することを目指す。

本研究では、食事の動作と大まかな食材の種類を検出するために、カラーセンサ／電極／マイコン／Bluetooth／バッテリーなどをフォーク自体に組み込んだデバイス：センシングフォークの設計と実装を行った。さらに、センシングフォークと連動し、子供の食習慣改善を目指すスマートフォン用のアプリケーション：腹ペコパンダを実装した（図1, [4]）。



図 1. システムの外観

2 関連研究

近年、HCIの分野において、食に関する多くの研究が発表されている。Grimesらは、食事をコンピュータで拡張する、Human-Food Interactionの可能性について調査・議論を行っている[3]。Parkerら[9]は、食事のオンライン広告に、栄養素や適切な食べ合わせを併せて示した場合の、子供たちへの影響を調査している。Mansourら[7]は、栄養バランスの良悪を楽しみながら学ぶことができる子供や若者向けのコンピュータゲームを開発している。これらの研究以外にも、食育支援のための様々なサービス^{4 5}が存在するが、その多くはPCやスマートフォン上のみで動作するものだった。

一方、人の食行動に対応したインタラクションを実現するために、食事の場にセンサやコンピュータを組み込むなど、ユーザの食行動に関する研究も多数行われている。Playful Tray[6]は、皿の横に取り付けたディスプレイから食事の進行に合わせて視覚的フィードバックを与えることで、食べるのが遅い子供を支援している。Dining Presenter[8]は、テーブル上方にプロジェクタを設置することで、事

Copyright is held by the author(s).

* Azusa Kadamura and Itiro Siio, お茶の水女子大学, Cheng-Yuan Li, Yen-Chang Chen and Hao-Hua Chu, 国立台湾大学, Koji Tsukada, 公立はこだて未来大学

¹ The U.S. Department of Agriculture:

<http://www.choosemyplate.gov>

² British Nutrition Foundation:

<http://www.nutrition.org.uk/>

³ 日本内閣府: <http://www8.cao.go.jp/syokuiku/index.html>

⁴ Fatworld: <http://www.fatworld.org/>

⁵ Shokuiku Town: <http://www.glico.co.jp/shokuiku/game/>

前に記入した絵やメッセージをテーブルに映し出し、食事を楽しくする効果や食育支援を目指している。Hapifork[1]は、食材を口に運ぶ頻度から早食い行動を感知し、振動で知らせるフォーク型デバイスである。また、本研究に先立って我々は、ユーザが食材を食べた瞬時に音フィードバックを返すことで、食行動改善を目指すフォーク型デバイスを開発してきた[5]。これらのシステムは、少数のセンサにより特定の食行動を検出しているため、アプリケーションが限定されていた。そこで本研究では、人の食行動（食事中の動作と大まかな食材の種類）を検出する多様なセンサを搭載したフォーク型デバイスと、これを用いたアプリケーションを開発した。

3 システム概要

本システムは、センサを組み込んだフォーク：センシングフォーク（図2左）と、フォークと連携して動くスマートフォンアプリケーション（図2右）から構築される。スマートフォン上のソフトウェアには、主に3つのモジュールがある。①フォークのセンサデータからユーザの食行動を推定する「食動作検出部」、②フォークと接触した食材を大まかに判別する「食材判別部」、および、③子供の食習慣の向上を目指した食育ゲーム：腹ペコパンダである。これらを組み合わせて、ユーザが「何を」「どのように」食べているのかを検出し、適切なフィードバックを返すことで食育を支援する。次に、センシングフォークのプロトタイプ的设计と実装について述べる。

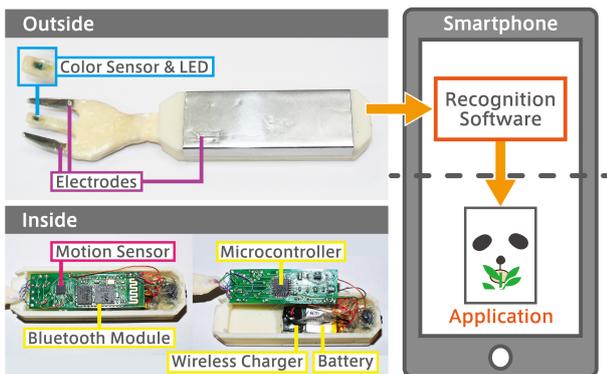


図 2. システム構成

左：センシングフォーク本体のプロトタイプ
右：スマートフォン上のソフトウェア

4 センシングフォーク

図2に示すように、センシングフォーク本体は、様々なセンサやハードウェア部品を内蔵している。得られたセンサデータは、スマートフォン上へ送信され、そこで食動作と食材の検出が行われる。フォークの持ち手サイズはL80×W25×H15mmに収めることで、小さな子供でも容易に握れるように設計した。以下、各モジュールについて詳細に説明する。

の持ち手サイズはL80×W25×H15mmに収めることで、小さな子供でも容易に握れるように設計した。以下、各モジュールについて詳細に説明する。

4.1 ハードウェア

4.1.1 電極

センシングフォークは、先端部分と握り部分に計3つの電極を備える（図2左上）。先端部分の両端2つの電極は、食材とフォークが接触したことを検出し、同時に食材の電気抵抗値を測定している。電気抵抗値は食材により異なるため、食材の推定に利用できる可能性がある。一方、握り部分の電極は、先端部分が食材を介してユーザの口に触れたとき、握り部分→手→腕→口→食材→先端部分と微弱な電流を流す。ここで、握りと先端の電極間の電気抵抗値を測定することで、フォークと食材がユーザの口に接触したこと（＝摂食動作）を検出できる。以上の電極部分の構成と、周辺回路について、図3に示す。

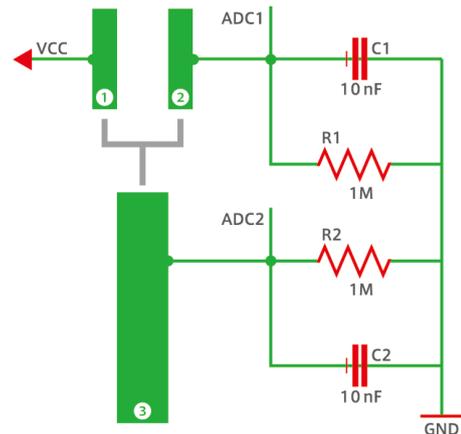


図 3. 電極とその周辺回路図

①と②はフォーク先端部分の電極を示し、
③はフォーク握り部分の電極を示している

4.1.2 カラーセンサ

フォークの先端部分中央には、カラーセンサと白色LEDを並べて組み込んだ（図2左上）。カラーセンサ（Avago ADJD-S311-CR999）はL2.2×W2.2×H0.76mmと超小型であり、フォークに接触した食材の色（RGB各色と輝度）を1024段階で検出し、I²Cを介してマイコンに送信する。フォークが食材の中に差し込まれると、白色LEDが点灯し、食材からの反射光をカラーセンサで測定する。なお、先端部の電極・およびカラーセンサ基板は、白色レジンでフォークの形状に固定した後、透明レジンで表面をコーティングした。レジンは共に歯科用の素材を利用することで、食の安全と防水に配慮した。

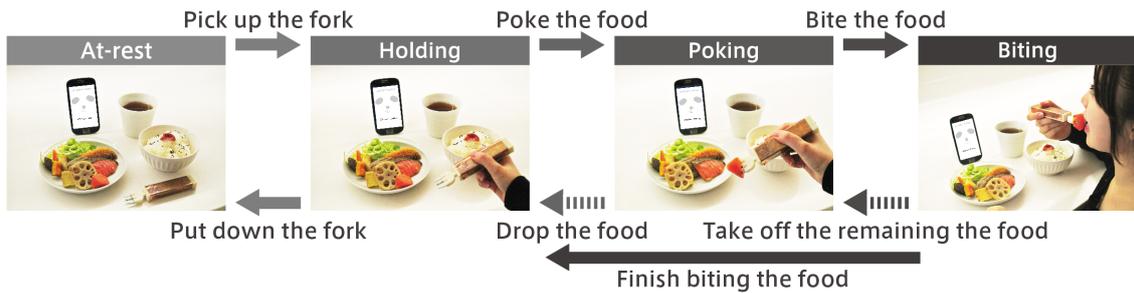


図 4. フォークの状態から導かれる食動作の遷移

4.1.3 モーションセンサ

フォークの握り内部のメイン基板上には、モーションセンサ（3軸加速度+3軸ジャイロセンサ：InvenSense MPU-6050）を設置した。これらのセンサデータはフォークの動きを検出し、I²Cを介してマイコンに送信され、食動作の推定に活用する。食動作検出の詳細は、4.2.1で説明する。

4.1.4 その他のモジュール

握り内部のメイン基板上にはマイコン、Bluetoothモジュール（SeeedStudio Serial port bluetooth module）、リチウムイオン充電電池、およびワイレス充電機構を搭載した（図2左下）。マイコンはAtmel ATmega328を利用し、Arduino互換機として実装した。Bluetoothモジュールは、上述したセンサデータをまとめてスマートフォンに送信する。リチウムイオン充電電池はサイズを考慮して120mAhのものを選択し、毎秒10回の無線送信を行っても2時間以上連続動作することを確認した。バッテリーを内蔵し、無線通信機能を備えることで、通常のフォークとできるだけ同じように扱えるように配慮した。

4.2 ソフトウェア

4.2.1 食動作検出部

食動作検出は、フォークからBluetooth経由で送信されるセンサデータを元に、スマートフォン上のアプリケーションで判定する。本システムでは、フォークの状態から以下の4種類のユーザーの食動作を推定する。図4に食動作の一覧と遷移を示す。以下、食動作検出と推定方法について述べる。

- 休止状態（At-rest）：
休止状態とは、ユーザーがフォークを握っていない（フォークが静止している）状態のことである。ユーザーが使う食器がフォークだけと仮定すると、ユーザーが食事のための動作をしていないと推定できる。フォークのモーションセンサが、一定時間動きを検出しない場合、休止状態にあるとシステムは判定する。

- 把持状態（Holding）：
把持状態とは、食材を刺していないフォークをユーザーが手に持っている状態のことである。この状態からは、ユーザーがフォークを持ち上げていて、何かしらの食材をフォークで食べようと考えていると推定できる。フォークのモーションセンサがフォークの動きや手振れを感知した場合、把持状態にあるとシステムは判定する。
- 刺突状態（Poking）：
刺突状態とは、ユーザーが食材をフォークで刺した状態のことである。この刺突状態からは、ユーザーが食材を刺したが、まだ口に運んでいないことを推定できる。フォーク先端の2つの電極間に一定の電流が流れることで、刺突状態にあるとシステムは判定する。なお、刺突状態をトリガーとして、次項（4.2.2）で説明する食材判別を開始する。
- 摂食状態（Biting）：
摂食状態とは、フォークに刺さっている食材がユーザーの口に接している状態のことである。この摂食状態からは、ユーザーがフォークに刺さっている食材を食べていることが推定できる。フォーク握り部分の電極と先端部分の電極間に微弱な電流が流れることで、摂食状態にあるとシステムは判定する。

これら4つの食動作の判定をより正確にするために、次の2つの仕組みを導入した。まず、各状態の間の遷移に図4のような制約を設けることで、誤認識を軽減する。すなわち、休止状態からは把持状態へしか遷移せず、把持状態からは刺突状態か休止状態へしか遷移しない。例えば、フォークで食材を刺さない（＝刺突状態にならない）限り、把持状態から摂食状態へは移行しない。これにより、例えば子供が食材を刺さずにフォーク先端をなめる動作と摂食動作を区別することができる。次に、摂食状態においては、フォーク握り部分と先端部分の電極に人体を介して微弱な電流が流れるため、フォークの把持者と摂食者が同一であることが判定できる。

4.2.2 食材判別部

本研究では、ユーザにバランスの良い食事摂取を促す目的で、2通りの手法を試作した。1つ目は食材種類判別手法であり、食事前に初期化プロセスを設け、各食材の色と電気抵抗値を登録する。次に、ユーザが食事中、食材を刺す度に、初期データと照合することで食材種類の判別を行う。機械学習アルゴリズムとしては、サポートベクターマシンの利用し、LIBSVM[2]を用いて実装した。2つ目は食材色判別手法であり、初期化は行わず、ユーザが食事中に食材を刺す度に、その食材の色を記録する。なお、ノイズを軽減するために、過去15回の色データを平滑化して利用する。

食材色判別手法では、食事中に取得した色データを随時読み取ることで、色の絶対値から食材の大きな種類を推定する。本手法では厳密に食材を判別することは難しくなるが、見た目の色の違う多くの食材を食べることで食育を支援する食材5色バランス健康法[10]も提案されており、色ベースとした判定手法には一定の有効性があると考えられる。また、初期登録の手間がないことから、導入や運用も容易である。よって、本論文では、食材色判別手法を用いて、後述の食育ゲームを実装した。

5 食育ゲーム：腹ペコパンダ

本章では、健康的な食生活のために様々な色の食材を摂取することを促す、センシングフォークと連動した食育ゲーム「腹ペコパンダ」について説明する(図1, 図5)。腹ペコパンダでは、スマートフォン画面上に表示されたパンダが、食材を把持したり、摂食したりなど、ユーザ(子供)と同じ食行動を行う。ここで、お腹を空かせたパンダに子供が共感することで、実際の子供の行動に影響を与えることを目指した。仮想ペットとしてパンダを題材とした理由は、アジア諸国の子供に親しまれているキャラクターだからである。腹ペコパンダは「偏食」「注意散漫」という子供の一般的な食の問題⁷の改善を目指す。まず、子供が別の色の食材を食べると、ゲームの得点が増える。これにより、高得点を得るために、より多くの食材摂取を動機付ける。次に、子供が食事時間中に一定時間摂食行動を停止したときに、パンダが悲しむことで、子供に食事に戻ることを促すフィードバックを返す。

以下に、フォークの状態と腹ペコパンダの画面(図5)の関係について説明する。

1. まず、お腹を空かせて悲しんでいるパンダが登場し、子供が食べ始めるまで、『僕は腹ペコパンダ、お腹が空いたよ。』と話す。【フォークの状態：休止・把持状態】

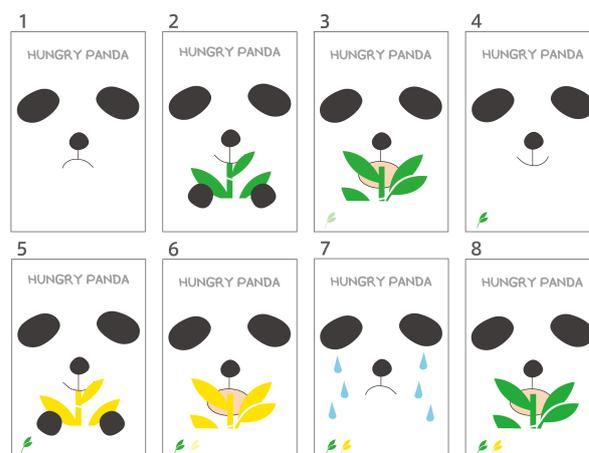


図 5. 腹ペコパンダの画面例

2. 子供が食材を刺すと、パンダも子供が刺した食材と同じ色⁸の箸を持つ。【刺突状態】
3. 子供がその食材を摂食すると、パンダもその箸を食べ『おいしい!』と話す。【摂食状態】
4. フォークの状態が把持状態に移行すると、パンダは箸ポイントを取得し、笑顔になる。【把持状態】
5. 同様に、子供が別の食材を刺すと、パンダは子供が刺した食材と同じ色の箸を持つ。【刺突状態】
6. 子供がその食材を摂食すると、パンダもその箸を食べ、『これもおいしいね!』と話す。【摂食状態】
7. 子供が食事を中断すると(=一定時間、休止・把持状態が続くと)、パンダは泣きながら、『もっと食べたいよ!』と子供がまた食べ始めるまで訴えかける。【休止・把持状態】
8. 子供が再び最初の食材を摂食すると、パンダは最初の箸と同じ色の箸を食べるが、箸ポイントは増えない。『他の食材も食べたいよ!』と話すことで、多様な食材の摂食を促す。【摂食状態】

最後に、終了ボタンを押すと、腹ペコパンダが獲得した箸ポイントの数に対応したアニメーションが表示される。つまり、万遍なく食べることで得られる箸ポイントの数に従って、最後に表示されるアニメーションが変わる。これにより、子供がより多くの食材を摂食する動機付けを行っている。

⁷ 厚生労働省：2005年度乳幼児栄養調査結果

<http://www.mhlw.go.jp/toukei/list/83-1.html>

⁸ 現システムでは、カラーセンサが検出したRGB値を予め設定した閾値で分類して、5色バランス健康法に基づく5色【赤・白・黄・緑・黒】の箸にマッピングしている。

表 1. 各食動作認識の成功回数と各認識に要した平均時間

	User A	User B	User C	User D	User E	User F	各状態の平均
Holding 1【把持状態 1】	5(0.3sec)	5(0.5sec)	5(0.3sec)	5(0.5sec)	5(0.6sec)	5(0.4sec)	5.0(0.4sec)
Poking【刺突状態】	5(0.6sec)	5(0.3sec)	5(0.9sec)	5(0.6sec)	5(0.6sec)	5(0.5sec)	5.0(0.6sec)
Biting【摂食状態】	5(0.3sec)	4(0.9sec)	3(0.9sec)	4(1.2sec)	4(1.1sec)	3(0.7sec)	3.8(0.9sec)
Holding 2【把持状態 2】	5(1.1sec)	5(1.8sec)	5(1.5sec)	5(1.1sec)	5(0.5sec)	5(1.2sec)	5.0(1.2sec)
At-rest【休止状態】	5(1.5sec)	5(1.4sec)	5(1.5sec)	5(1.5sec)	5(1.1sec)	5(1.4sec)	5.0(1.4sec)
各ユーザの平均	5.0(0.8sec)	4.8(1.0sec)	4.6(1.0sec)	4.8(1.0sec)	4.8(0.8sec)	4.6(0.8sec)	

このように、本ゲームはセンシングフォークによって推定・検出されたユーザの食行動を基に、視覚・聴覚的なフィードバックを返し、食事時間中に食育を支援することを目指す。すなわち、子供は楽しく食行動を改善でき、親はパンダの獲得した笹ポイントを見ることで、日々の子供の食行動を継続的に把握することができる。付随的な効果としては、子供の食行動が改善された場合には、本ゲームによるご褒美（笹ポイントと最後のアニメーション）に加えて、親がお菓子などのご褒美を子供に与えるなどして、親子が食育に楽しく取り組むきっかけにつながると期待できる。

6 評価実験

センシングフォークの性能を評価する基礎的な評価実験を行った。本章では、食動作検出と食材判別の精度について述べる。

6.1 食動作検出について

食動作検出の精度を確認するために、6人のユーザ（男女各3名、21～28歳）にセンシングフォークで食材（一口サイズのチーズ）を5回食べてもらった。一連の食動作の認識精度を確認するため、各ユーザには1回食べるごとに、フォークを机に置くように指示した。すなわち、一度食材を食べる度に【把持状態1→刺突状態→摂食状態→把持状態2→休止状態】の順で一連の食動作が行われた。なお、全てのユーザはセンシングフォークを初めて利用し、システムの機能も事前に説明しなかった。

実験では、各ユーザの動作とスマートフォン上の実験用ソフトウェア画面を同時にビデオ撮影した。実験用ソフトウェアには、現在の食動作の状態とセンサデータがテキストで表示されている。実験終了後、実際のユーザの行動とソフトウェア上に表示された食動作をPC上でフレームごとに目視で比較した。なお、ユーザの食動作の遷移は、【把持状態1：フォークが机から離れた瞬間】【刺突状態：フォーク先端が食材に触れた瞬間】【摂食状態：ユーザが食材を噛んだ瞬間】【把持状態2：ユーザの口からフォークが離れた瞬間】【休止状態：フォーク本体が机に置かれた瞬間】というフレームで判定した。

まず、各食動作が認識された回数と、システムが

各認識に要した平均時間を表1に示す。ユーザ毎に25回の食動作（上述した5食動作を5セット）が行われた。その結果、全体としては95%の認識率だった。一方、【摂食状態】において23%の誤認識があり、【刺突状態→摂食状態】と認識されずに【刺突状態→把持状態2】と誤認識されていた。認識されなかったユーザの【摂食状態】をビデオ動画から観察したところ、誤認識された摂食時間の平均は0.1秒であり、正しく認識された場合の平均は0.9秒であった。したがって、フォークを一瞬しか口につけなかった場合に、誤認識が発生した。現在の実装では、バッテリーの消耗を考慮して、スマートフォンとの通信頻度を秒間10回（0.1秒間隔）に抑えており、この結果、ユーザが素早く行動した場合に十分な測定が行えなかったと考えられる。センシング頻度を上げることで、食動作の認識は改善できると考えている。参考として、ユーザCの食動作遷移例を図6に示す。

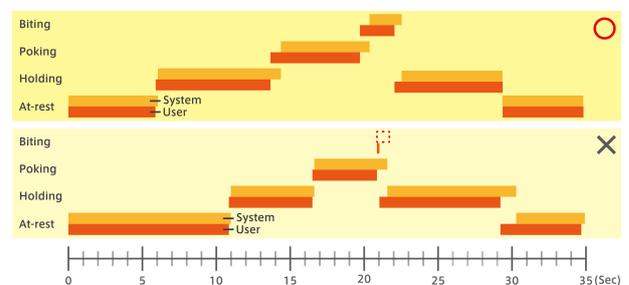


図 6. ユーザ C の食動作遷移の一例

上段が認識成功時、下段が認識失敗時の事例。各2列の横棒は上がシステムの認識結果、下が実際のユーザの食動作を示す。下段では、【摂食状態】が0.07秒と短時間で行われたため、認識されていない。

また、表1に示す通り、各動作認識に要した時間の平均は0.9秒であった。各動作の中でも、【休止状態】の認識は平均1.4秒と時間がかかった。この理由は、1秒以上フォークが静止したときに【休止状態】と判断しているからである。こちらも待機時間を調整することで、認識速度を改善できる可能性はあるが、一般にフォークを机に置く／取る行為が高速で繰り返されることは少ないので、実用上問題ないと考えている。また、【把持状態2】も平均1.2秒

と時間がかかっているが、これは実験プログラムの不具合（摂食時に2.2秒程度のアニメーションが表示され、終了まで状態表示が更新されなかった）が原因であり、実際の認識は【把持状態1】や【刺突状態】と同程度の時間で完了できると思われる。

6.2 食材判別について

4.2.2で述べた食材色判別手法の性能を確認するために、市販のお弁当から色が異なった食材を5種類（ハンバーグ【赤】、米【白】、かぼちゃ【黄】、ブロッコリー【緑】、しいたけ【黒】）用意した。なお、【】内の色は前述の食材5色バランス健康法[10]での分類を示す。実験者は、各食材の同じ面の違う箇所を10回ずつセンシングフォークで刺し、RGB値を記録した。まず、同一食材毎の検出誤差（標準偏差）としては、米が平均0.9、かぼちゃが平均6.5と安定していたのに対し、ハンバーグは平均14、しいたけは平均15、ブロッコリーは平均25と誤差が大きくなった。これは、ハンバーグの上のケチャップや、ブロッコリーの葉部分の隙間、しいたけの表面／内部の色の濃淡の違いといった食材の特性に影響を受けたと考えられる。

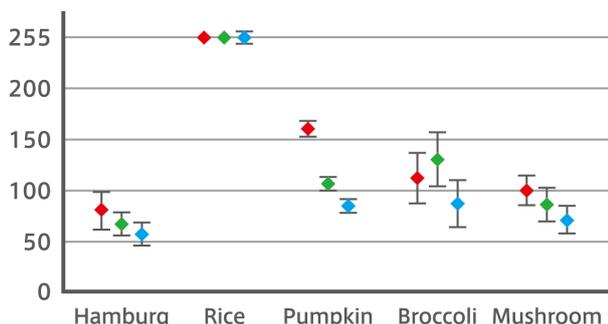


図 7. 各食材の RGB の平均値と標準偏差

次に、現在の腹ペコパンダで採用している、各食材を5色【赤・白・黄・緑・黒】にマッピングする方法を適用したところ、米【白】、カボチャ【黄】は10回中10回、ブロッコリー【緑】は9回認識に成功した。一方、しいたけ【黒】は10回中5回、ハンバーグ【赤】は10回中4回しか認識出来ず、大半が相互に誤認識していた。現在の食材色判別手法は単純な閾値で判定しているため、今後は色空間上の距離や機械学習などを利用して認識率を改良したい。

7 まとめと今後の課題

本論文では、センサ内蔵フォークを用いてユーザの食行動を検出するセンシングフォークの設計と実装について述べた。また、子供の食事中的「偏食」「注意散漫」といった問題行動を改善することを目指した「腹ペコパンダ」ゲームも実装した。さらに、

フォークの基礎的な性能を検証するために評価実験を行い、食動作検出と食材の色による分類について、改良の余地はあるものの、一定の性能があることを確認した。

今後は、実際に子供を対象とした、家庭内での評価実験を計画している。さらに、食事中的更なる問題行動を検出したり、食材の摂食順序を記録したりすることで、新たな食行動改善アプリケーションの可能性を探る。

謝辞

本研究は、JSPS 科研費 25・10205 と科学技術振興機構さきがけプログラムの助成を受けた。

参考文献

- [1] Hapifork. <http://www.hapilabs.com/>.
- [2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3).
- [3] A. Grimes and R. Harper. Celebratory technology: new directions for food research in HCI. In *Proceedings of CHI '08*.
- [4] A. Kadomura, C.-Y. Li, Y.-C. Chen, H.-H. Chu, K. Tsukada, and I. Siiio. Sensing fork and persuasive game for improving eating behavior. In *Proceedings of UbiComp '13*.
- [5] A. Kadomura, K. Tsukada, and I. Siiio. EducaTableware: computer-augmented tableware to enhance the eating experiences. In *Extended Abstracts on CHI '13*.
- [6] J.-L. Lo, T.-Y. Lin, H.-H. Chu, H.-C. Chou, J.-H. Chen, J. Y.-J. Hsu, and P. Huang. Playful tray: adopting UbiComp and persuasive techniques into play-based occupational therapy for reducing poor eating behavior in young children. In *Proceedings of UbiComp '07*.
- [7] A. Mansour, M. Barve, S. Bhat, and E. Y.-L. Do. MunchCrunch: a game to learn healthy-eating heuristics. In *Proceedings of IDC '09*.
- [8] M. Mori, K. Kurihara, K. Tsukada, and I. Siiio. Dining Presenter: Augmented Reality system for a dining tabletop. In *Proceedings of UbiComp '09*.
- [9] A. G. Parker, I. McClendon, C. Grevet, V. Ayo, W. Chung, V. Johnson, and E. D. Mynatt. I am what i eat: identity & critical thinking in an online health forum for kids. In *Proceedings of CHI '13*.
- [10] 杉本恵子. 「生活習慣病は、『食材5色バランス健康法』で予防・改善を」。In *Kellogg's Update*, 第82巻, 2005.
- [11] 鈴木秀子. 子どもから家庭へつなぐ食育～保護者の「学び」からの検討～. 会津大学短期大学部研究年報, 第67巻, 2010.

鍵盤上への演奏補助情報投影機能をもつ ピアノ学習支援システムにおける熟達化プロセスに関する調査

竹川 佳成 椿本 弥生 田柳 恵美子 平田 圭二*

概要. 楽器の演奏技術の向上には多大な時間や労力を必要とするため、敷居の高さに利用を断念したり、習熟効率の低さから挫折してしまう演奏者が多い。この問題を解決するために、筆者らの研究グループは、鍵盤上部に設置したプロジェクタを用いて鍵盤上や鍵盤の周囲に打鍵位置情報など演奏補助情報を投影するピアノ学習支援システムを構築してきた。評価実験から比較手法である「光る鍵盤」の学習方法と比較して課題曲を効率的に学習できることが明らかになったが、「課題曲を効率的に学習できただけか、あるいは、ピアノ演奏そのものの演奏技術が向上したのか」「必要とする提示情報は熟達度に応じて変化するのか」といった提案する学習支援システムを使用した場合における熟達化プロセスは十分調査できていなかった。そこで本研究では、この問題を解決するために、ピアノ学習支援システムの熟達化プロセスの詳細な調査を目的とする。評価実験では、提案する学習支援システムを使いながら5日間かけて6人の被験者に1日30分間課題曲を練習してもらい、実験日ごとの熟達度を調査するためにシステムの補助がない状態で課題曲を演奏してもらった。練習中に記録した打鍵データや視線データなどをもとに熟達化プロセスを検証した。

1 はじめに

ピアノ演奏では、譜読み、指示されている鍵への正確な打鍵、適切な運指（指使い）、リズム感覚、打鍵の強弱、テンポなど、さまざまな技術が求められ、それらの修得には長期間の基礎的な練習を必要とする。ピアノ演奏には多大な時間と労力を必要とするため、敷居の高さに利用を断念したり、習熟効率の低さから挫折してしまう演奏者が後を絶たない。特に初心者にとって、譜面上の音符および運指を見て、音符から鍵盤上の打鍵位置をイメージし、指示された運指で弾くという一連のプロセスは最初に立ち足はかかる難関で、このプロセスに対する労力や精神的負荷の軽減が楽器演奏を楽しく長続きさせる秘訣であるといえる。演奏初期段階（ピアノ初心者が初見の楽曲に対して運指や打鍵位置を覚えるために練習している段階）における敷居を下げるために、筆者らの研究グループは、鍵盤上部に設置したプロジェクタを用いて鍵盤上や鍵盤の周囲に打鍵位置情報など演奏補助情報を投影するピアノ学習支援システムを構築してきた [1]。評価実験から比較手法である「光る鍵盤の学習方法 [2, 3]」と比較して効果的に学習できることが明らかになった。しかし、「課題曲を効率的に学習できただけか、あるいは、ピアノ演奏そのものの演奏技術が向上したのか」「必要とする提示情報は熟達度に応じて変化するのか」といった提案する学習支援システムを使用した場合における熟達化プロセスは十分調査できていなかった。

そこで本研究では、この問題を解決するために、ピアノ学習支援システムの熟達化プロセスの詳細な調査を目的とする。

評価実験では、提案システムを使いながら5日間かけて6人の被験者に1日30分間課題曲を練習してもらい、実験日ごとの熟達度を調査するためにシステムを利用しない状況で練習後に課題曲を演奏してもらった。また、5日間かけて練習してもらった課

題曲とは異なる曲を実験初日と5日目に演奏してもらい、提案システムを使って練習したときのピアノ演奏技術の熟達度を調査した。さらに、打鍵タイミングや打鍵位置といった演奏データだけでなく、視線追跡装置を用いて鍵盤上や鍵盤の周囲のどこを見ていたか計測した視線データにより、熟達化プロセスにおける重要情報の変遷を分析する。

以下、2章で関連研究について説明し、3章で実験計画について述べる。4章で評価実験について述べ、最後に5章で本研究のまとめを行う。

2 関連研究

これまでピアノ学習の支援につながる試みはいくつか行われている。蓄積した演奏データから演奏者の苦手な奏法を割り出し集中的にトレーニングするシステム [4] や、演奏を自動的に評価しアドバイス文や誤りを譜面上に提示 [5] するシステムがある。これらは、打鍵ミス、打鍵の強さなどを主に打鍵情報から評価している。Piano Tutor [6] は演奏追従認識による自動譜めくり機能や、ビデオや音声による模範演奏の提示や、演奏者の演奏データを解析し改善点をテキストなどで指示する機能などをもつ。先生と生徒のレッスン支援 [7] として、音量の変化やテンポ、スタッカートやレガートといったアーティキュレーションの具合等を示すシステムが提案されている。打鍵すべき鍵、運指、手本映像を表示するキーボードやソフトウェア [1, 2, 3] がある。これらはいずれも打鍵情報から演奏を評価し学習目的に必要な情報を提示している。本研究では、視線データと打鍵データを用いて演奏を評価している点で異なる。

学習者の視線の動きを観測する研究 [8] はいくつか行われている。特に、川崎ら [9] はピアノ学習者の視線の動きを観測し、プロフェッショナルのピアニストと初心者との視線の動きの違いを解析している。本研究は初心者を対象としており、対象者という観点では川崎らの研究と類似する部分があるが、本研究のように鍵盤上にアノテーションが投影されるピ

アノではなく、川崎らは一般的なアコースティックピアノを利用しているため、実験環境は大きく異なる。また、川崎らは楽譜上の視線追跡だけであったが、本研究では楽譜および鍵盤の視線を追跡するという点でも異なる。

3 実験計画

筆者らの研究グループが提案するピアノ学習支援システムは、ピアノ初心者を対象としており、五線譜やシステムが生成する補助情報を活用しながら学習者はある楽曲を一から練習し、できるだけ速く習熟し、最終的にシステムの補助なしで演奏できるようになることをめざしている。このために、鍵盤上に次に演奏する打鍵位置や運指番号、譜読みを補助するために鍵盤の上部に提示した五線譜の音符とそれに対応する鍵を結ぶ線など多彩な情報を提示している [1]。1章で述べたように、本研究では提案するピアノ学習支援システムを使用して学習したときにおける熟達化プロセスを調査するために詳細な評価実験を行う。

3.1 実験方針

実験方針として以下があげられる。

ピアノ演奏技能の獲得 提案するピアノ学習支援システムは、30分間の評価実験から比較手法である「光る鍵盤の学習方法」と比較して打鍵ミス数が減り、提案するピアノ学習支援システムを使用しながら練習すれば効率的に課題曲を弾けるようになる。しかし、「新規の楽曲を提案システムの補助なしで演奏できるようになる」といったピアノ演奏の汎用的な技能を修得できたかは検証されていない。

そこで、本研究では提案する学習支援システムを使って練習する前と、提案システムを使った後で、新規の楽曲をそれぞれ演奏してもらう。その演奏精度を計測することで、提案システムが学習者のピアノ演奏技能の向上に貢献するか調査する。

提示情報の重要度の変遷 提案する学習支援システムは演奏を補助するさまざまな情報を提示しているが、例えば、「未熟な段階では打鍵位置が明確にわからないため手元の打鍵位置情報を頼りに演奏するが、熟達するにつれて打鍵位置情報に頼らず楽譜を見る」など熟達度によって情報の重要度は変わってくると思われる。

そこで、より効率的な学習支援システムの提案をめざして、熟達するにつれ必要とされる提示情報の変遷を明らかにする。具体的には、提案システムを使いながら5日間かけて6人の被験者に1日30分間課題曲を練習してもらい、実験日ごとの熟達度を調査するためにシステムを利用しないで課題曲を演奏してもらう到達度テストをうけてもらう。また、打鍵タイミングや打鍵位置といった演奏データだけでなく、視線追跡装置を用いて鍵盤上や鍵盤の周囲のどこを見ていたか視線データを計測する。これらのデータおよび実験中の観測から、熟達化プロセスにおける重要情報の変遷を定量的・定性的に分析する。なお、実験結果を明確にするために、打鍵位置をシステムの補助なしでできるだけ速く習熟できる

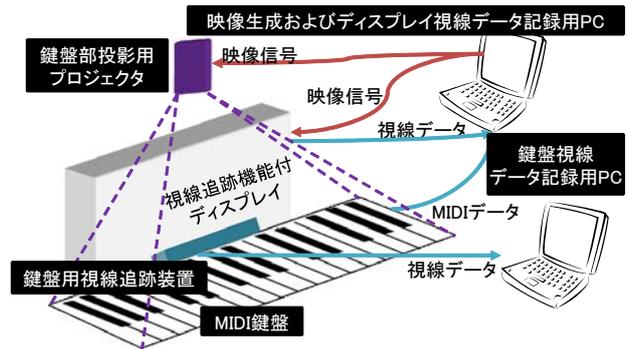


図 1. システム構成

ことを提案システムの目的とし、この目的に特に有効な情報であった「鍵盤上への打鍵位置および運指番号」・「楽譜」・「楽譜上に提示する現在の演奏位置」のみ演奏補助情報として提示する。

3.2 実験システム

実験で使用した学習支援システムのシステム構成を図1に示す。鍵盤上部に設置したプロジェクタを用いて鍵盤上に演奏補助情報を提示する。また、演奏者の前方に視線追跡機能付ディスプレイを配置し、プロジェクタと同様に演奏補助情報を提示している。システムは、MIDI データ（打鍵位置や打鍵強度）を入力とする。さらに、鍵盤部の視線を追跡するために鍵盤上部に視線追跡装置を設置し、演奏の様子を記録するためにビデオカメラを設置した。

映像生成およびディスプレイ視線データ記録用のPCとしてSONY社のVGN-SR94VSを使用し、鍵盤部視線データ記録用のPCとしてSONY社のVPCSAを使用した。また、MIDI鍵盤としてCASIO社のPriviA PX-110を使用した。鍵盤部の視線追跡装置としてTobii社のX1ライトを使用し、ディスプレイ部の視線追跡装置としてTobii社のT60を使用した。プロジェクタとしてBenQ社のMP776STを使用した。プロジェクタの鍵盤投影領域は6オクターブ（72鍵）で、プロジェクタの映像がよく見えるように黒鍵を白く塗った。PC上のソフトウェアの開発は、Windows 7上でMicrosoft社のVisual C++ 2010を用いて行った。

3.3 提示コンテンツ

図2を用いてシステムが提供したコンテンツについて説明する。図中の番号は、以下の箇条書き番号に対応している。

1. 既存の紙媒体の楽譜と同様の楽譜を提示する。
2. 現在の演奏位置を提示する。これにより、学習者は現在どこを演奏しているか直観的に理解できる。正しい鍵を弾いたときのみ演奏位置は進むようになる。
3. 次に演奏する鍵上に色付枠を提示する。運指情報は、運指番号（親指から小指にかけて1から5の番号がそれぞれ割割り当てられている）ごとに対応している輪郭の色や、鍵上に運指番号を提示することで示す。これにより学習



図 2. 提示コンテンツ

者は容易に打鍵位置や運指を把握できる。また、ディスプレイにも鍵盤と同様の打鍵位置情報や運指情報を提示する。ディスプレイには鍵の枠しか提示されないため、鍵盤上に提示された情報と比較して得られる情報は少なく、直観性に欠ける。

4. 楽譜上に表示されている番号付きの黒塗りの四角形は、現在位置を変更するキューポイントである。これは、学習者が集中的に練習したい場合や、途中から演奏したい場合に有効である。キューポイントを切り替えるアイコンを、演奏で使用しない鍵の鍵盤上に投影し、その鍵を打鍵することでユーザが選択的に利用できるようにする。これをキューポイント設定機能と呼ぶ。
5. 提案システムは打鍵位置および運指情報を提示するかどうかを切り替える機能（打鍵位置提示切替機能）をもち、この機能を操作するアイコンを演奏で使用しない鍵の鍵盤上に投影する。打鍵位置提示切替は、割り当てられた鍵を押すごとにトグル式で切り替わる。また、提案システムは模範演奏を再生する機能（模範演奏再生機能）をもち、同様に演奏で使用しない鍵の鍵盤上に模範演奏再生機能に対応するアイコンを投影する。学習者が模範演奏再生機能に割り当てられた鍵を押下すると模範演奏の再生が始まり、再生中にその鍵を再度押下すると模範演奏を途中で止められる。なお、本論文では、キューポイント設定機能・打鍵位置提示切替機能・模範演奏再生機能をまとめて付加機能と呼ぶ。

4 評価実験

評価実験では、演奏初期段階（ピアノ初心者が初見の楽曲に対して運指や打鍵位置を覚えるために練習している段階）における提案システムを用いた際のピアノ演奏に関する熟達化プロセスを、打鍵データや視線データをもとに分析する。

4.1 実験の手順

実験の手順を以下に示す。

被験者 実験に参加した被験者は6名で、五線譜がほとんど読めない鍵盤経験歴のない20代から40代の成人である。なお、各被験者にはあらかじめ楽譜上に書かれている音符の意味や、各種機能の使い方を説明した。

課題曲 熟達化プロセスを評価するために、W. A. Mozart のトルコ行進曲（最初から17小節目まで両手）を演奏してもらった。また、ピアノ演奏技術の熟達度を評価するために、J. S. Bach のメヌエット（BWV Anh. 114）の最初から8小節目までを両手で演奏してもらった。

実験方法 実験では、「トルコ行進曲を30分かけて練習し、到達度テストとして通し演奏（最初から最後まで一通り演奏することを行う）」という試行を1日1回行なった。これを連続5日間かけて繰り返した。

ピアノ演奏技術の上達を検証するために、「初日のトルコ行進曲練習前」および「5日目のトルコ行進曲到達度テスト後」にメヌエットの到達度テストを行った。練習中および到達度テスト中は、視線計測装置が生成する視線データ、MIDI鍵盤が生成する打鍵データをシステムに記録し、演奏中の様子をビデオカメラで記録した。なお、メヌエットの到達度テストは初日と5日目の両方に行ったが、被験者は初日に行ったメヌエットの演奏を覚えておらず、5日目のメヌエットの到達度テストは、初日のメヌエットの到達度テストに影響はない。

到達度テストでは、テスト中で計測している打鍵データをもとに打鍵ミス数を計測した。通し演奏時は、両手法において前面にある楽譜のみ（現在の演奏位置も提示しない）提示した。また、誤打鍵（間違えて打鍵した場合）、未打鍵（打鍵しない場合）、余打鍵（余分に打鍵した場合）を打鍵ミスとみなした。

被験者への指示 30分間の練習では「自然なテンポで譜面をみながらミスなく弾けることを意識して、機能を自由に使ってもらって30分間練習してください。また、この後、到達度テストを行います。到達度テストは打鍵位置の情報などシステムからの補助情報がない状態で最初から最後まで弾いてもらいます。実験中に質問があれば何でも聞いてください」と指示した。また、到達度テストでは「今から到達度テストを行います。最初から最後まで模範演奏にでき

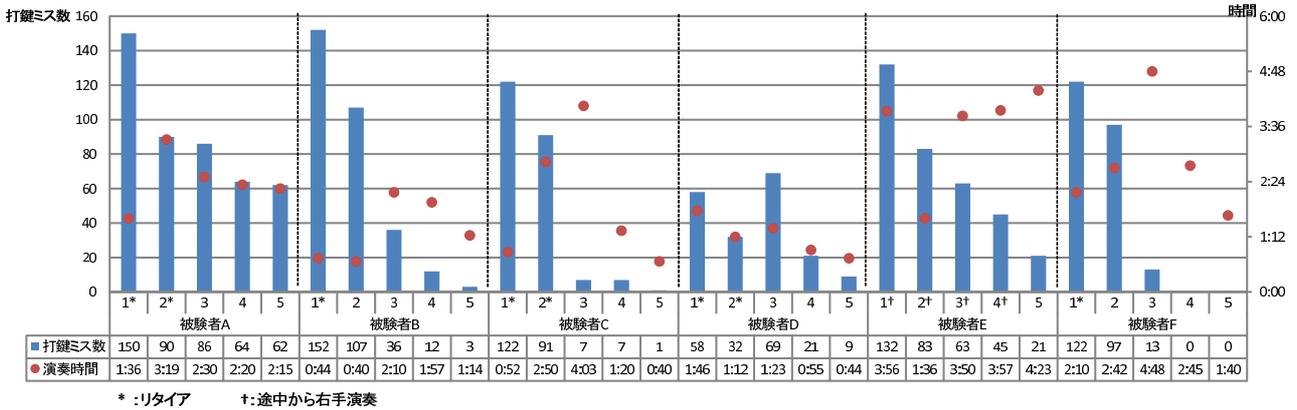


図 3. トルコ行進曲の到達度テスト - 打鍵ミス数および演奏時間

るだけ近いテンポでミスなく弾いてください。制限時間は5分間です。わからないところがあれば飛ばしてもらってもかまいませんし、これ以上演奏できなければ言うてください。たとえ間違っても弾き直しをしないようにしてください」と指示した。なお、難しすぎて練習を放棄した被験者はいなかった。

4.2 ピアノ演奏技能の獲得 - 実験結果と考察

トルコ行進曲の到達度テストにおける打鍵ミス数および演奏時間（課題曲を「最初から最後まで」あるいは「最初からリタイア」するまでにかかった時間）を図3に、メヌエットの到達度テストにおける打鍵ミス数および演奏時間を図4に示す。

全体的に、トルコ行進曲の到達度テストにおける打鍵ミス数は実験が進むにつれて減少し、途中リタイアした場合を除き演奏時間も減っている。また、メヌエットにおいては、打鍵ミス数が減少し演奏時間が短くなった被験者がいた一方、上達が見られなかった被験者もいた。この原因として被験者ごとの学習方略の違いが考えられる。

提案する学習支援システムは、打鍵位置情報を視覚的に提示することで、打鍵位置をできるだけ速く学習できることをめざしている。また、打鍵位置提示切替機能、模範演奏再生機能、キューポイント指定機能といった付加機能を使うことで学習の促進をねらっている。打鍵位置情報はデフォルトで提示されるが、これらの付加機能は選択的に利用される。したがって、提案する学習支援システムは「提示された打鍵位置情報を活用し打鍵位置を学習し、学習効率を高めるために能動的に付加機能を活用する」という学習方略を理想としている。

被験者の学習方略は、「学習支援システムが提供する打鍵位置情報を利用し打鍵位置を学習するものの、付加機能をほとんど使わない学習方略（打鍵位置情報依存型）」「学習支援システムが提供する打鍵位置情報を利用し打鍵位置を学習すると同時に、付加機能を積極的に利用し熟達度に応じて学習方法を柔軟に構築する学習方略（システム方略適合型）」「学習支援システムが提供する打鍵位置情報に頼らず、自身の過去の成功体験をもとに練習する学習方略（システム方略不適合型）」の3つの大きく分類できる。

以降、各学習方略について考察する。

打鍵位置情報依存型 打鍵位置情報依存型の被験者Aは、他の被験者と比較して最も熟達度が低かった被験者である。被験者Aの付加機能を使用回数は、他の方略の被験者と比べて少なく、打鍵位置情報を利用しながら練習していた。実験3日目以降に打鍵位置提示切替機能を使用する様子が観測されるが、30分間の訓練中2分〜3分と短期間に、あるフレーズに対して打鍵位置情報を見ずに弾けるか確認をしただけである。被験者Aは実験が進むにつれてトルコ行進曲の到達度テストの打鍵ミス数は少なくなっている一方、5日目のメヌエットの打鍵ミス数は初日と比べて増えている。被験者Aは「PC用キーボードのタッチタイピングを修得するとき、タッチタイピング練習ソフトウェアで繰り返し練習をして習熟していった過去の成功体験があり、タッチタイピングの練習と類似している実験の学習方法は、学習しやすかった」とコメントしている。また、「タッチタイピングでは、ミスなくタイピングできた単語もできなかった単語も区別なく均等に練習したため、今回の実験においても、間違えやすいところを集中的に訓練するといったことはやらなかった」とコメントしている。したがって、被験者Aは学習支援システムの学習方法に肯定的でストレスなく学習できていたが、学習支援システムを十分使い切れていたとはいえない。

一方、被験者Bは被験者Aより付加機能を使っておらず、システムが提示する打鍵位置を常に使いながら練習していた。被験者Bも、繰り返し練習により新たな技能を習得した過去の成功体験があり、このような学習方略にいたった。被験者Bは被験者Aより受動的な学習方略であったにも関わらず、被験者Aと異なり実験が進むにつれて着実に熟達している。被験者Aの熟達が伸び悩んだ要因の調査は今後の課題である。

システム方略適合型 被験者Cや被験者Dは打鍵位置情報をもとに打鍵位置を学習すると同時に、付加機能を使って難しい箇所を集中的に練習したり、打鍵位置情報を使わずに修得できているか確認しな

鍵盤上への演奏補助情報投影機能をもつピアノ学習支援システムにおける熟達化プロセスに関する調査

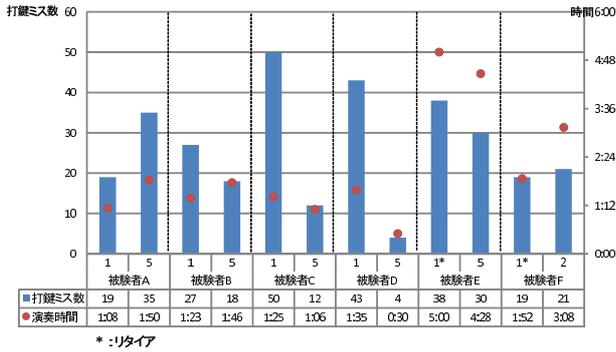


図 4. メヌエットの到達度テスト - 打鍵ミス数および演奏時間

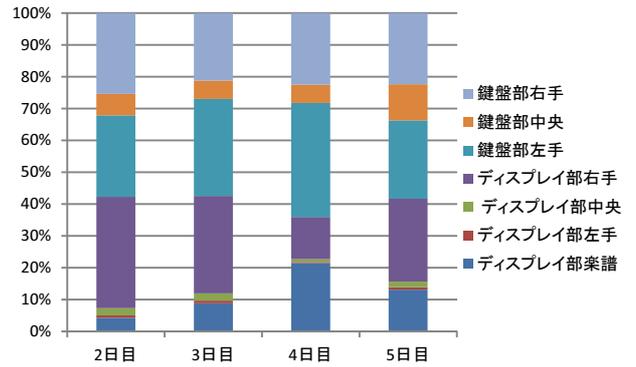


図 5. トルコ行進曲の練習中 - 実験日ごとの視線分布

から練習していた。特に被験者 C は、打鍵位置提示切替機能やキューポイント選択機能を何度も使用し、集中的に部分練習していた。実験が進むにつれて着実に到達度テストの打鍵ミス数が少なくなり、途中リタイアを除き課題曲を弾き通すためにかかった時間も短くなり演奏テンポが速くなっていった。メヌエットにおいても実験初日と実験 5 日目では打鍵ミス数は減少し、演奏テンポは速くなった。これにより、提案する学習支援システムは課題曲を効果的に学習できるだけでなく、ピアノ演奏技術の向上にも貢献できたことがわかる。メヌエットの到達度テストにおいて、システム方略適合型被験者における打鍵ミス数増加率（「実験 5 日目の打鍵ミス数」を「初日の打鍵ミス数」で割った値）が最も小さく、「システム方略適合型被験者における打鍵ミス数増加率」および「システム方略不適合型被験者における打鍵ミス数増加率」間で有意水準 5% で有意差 (p 値=0.047) が観測された。その他の組み合わせにおいて有意差は観測されなかった。

システム方略不適合型 システム方略不適合型の被験者 E や被験者 F の特徴として、実験が進むにつれてトルコ行進曲の到達度テストの打鍵ミス数は減少する一方、メヌエットの到達度テストにおける打鍵ミス数は、実験初日と実験 5 日目と比べて変化は小さい。

被験者 E は、提示された打鍵位置情報にはできるだけ頼らず、模範演奏を聴きこんだり、楽譜をじっくり見ながら打鍵位置を模索していた。被験者 C は、途中まで模範演奏を聴いて練習するという訓練を何度もやっていたため模範演奏再生回数は増えたが、被験者 E は模範演奏を最初から最後までじっくり聴いていた。被験者 E は声楽を学んでいた経験があり、「声楽の新曲に取り組むときその曲を繰り返し聴いて覚える」という成功体験をもつ。今回の実験においても、自身の成功体験を適用し聴覚中心の学習方略で取り組んでおり、提案する学習支援システムの学習方略に適さなかった。

被験者 F も被験者 E と同様に、できるだけ打鍵位置情報を使用しないで訓練していた。被験者 F は、「トルコ行進曲の到達度テストでどうすれば打鍵ミスなく弾けるようになるか」を中心に考えるゴール指向型の学習方略で取り組んでおり、到達度テスト

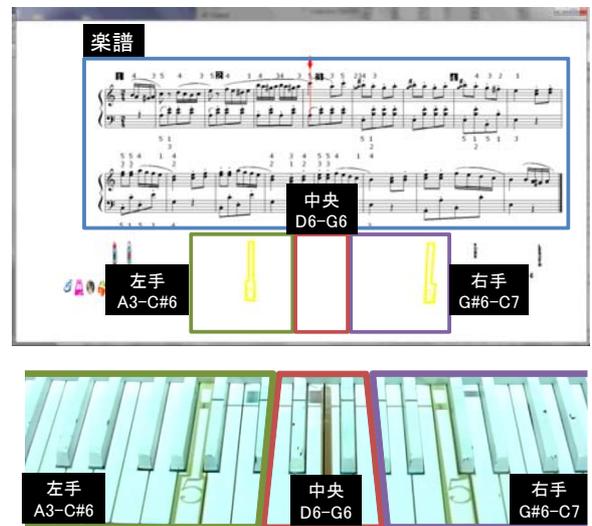


図 6. ディスプレイ部と鍵盤部における視線領域

では打鍵位置情報が提示されないことから、打鍵位置情報を見ずに演奏できるように練習していた。このため、到達度テストにおけるトルコ行進曲の打鍵ミス数は減少しており、被験者 A～被験者 F の中で最初に打鍵ミス数が 0 になったが、楽譜を中心に打鍵位置を考えながら弾いていたため演奏テンポは他の被験者と比べると遅い。また、トルコ行進曲用に組み立てられた学習方略はトルコ行進曲の到達度テストにおいて効果的であったが、5 日間の練習後に演奏したメヌエットの到達度テストでは、5 日間の練習で得られたスキルを活かせず、練習前に演奏したメヌエットの到達度テストと変わらない結果となった。これにより、ピアノ演奏においては楽譜や鍵盤をよく理解する認知的側面の熟達も重要であるが、楽譜上の音符から打鍵位置を正しく演奏できるようにするという身体行動の熟達も重要であるといえる。これらがバランス良く熟達することでピアノ演奏技術そのものも向上すると思われる。

4.3 提示情報の重要度の変遷 - 実験結果と考察

「熟達度」と「提示情報の重要度の変遷」を分析するために、30 分間のトルコ行進曲の練習における

視線の分布を調査した。上述したように学習方略の違いが見られたため被験者ごとに視線分布を分析することが望ましいが、紙面の制約のため、被験者Bに着目して詳細に検討する。被験者Bを選定した理由は、実験日ごとの練習においてシステムが提供する付加機能を使用せず、提示される打鍵位置をもとに打鍵位置を覚える訓練淡々と繰り返しており、どの打鍵に対してどういう視線であるか特定しやすいためである。

図5に実験日ごとの視線分布を示す。実験で得られた視線データを図6に示す視線領域にマッピングすることで図5の視線分布を求めた。ディスプレイには打鍵位置だけを示す仮想的な鍵盤が提示されており、ディスプレイ上の仮想鍵盤と実際の鍵盤を演奏する手を基準に右手・左手・中央に領域をわけた。トルコ行進曲は右手で弾く領域と左手で弾く領域が明確にわかれており、右手と左手の境界線はE6 (Eは音名をおよび6はオクターブ数を意味する) である。視線認識精度の問題から、右手と左手の境界付近は明確に右手左手どちらの領域を見ているか特定できないため、中央という領域を設けた。また、実験日が2日目から始まっている理由はシステムの不具合により初日の鍵盤部の視線がとれていなかったためである。

図5に示すように、熟達度が低い2日目ではディスプレイ部楽譜を見る頻度は少なく、鍵盤あるいはディスプレイ部右手を中心に見ている。4日目になると楽譜を見る頻度はあがり、ディスプレイ部右手や鍵盤部右手を見る頻度は少なくなった。これにより熟達するにつれて、鍵盤に提示された打鍵位置情報や、ディスプレイ上に提示された打鍵位置情報は不要となり、楽譜情報が重要となっていることがわかる。また、右手の打鍵位置に関してはディスプレイ部右手を見ており、左手の打鍵位置に関しては鍵盤上の打鍵位置を直接見ている。被験者Bは「余裕があれば演奏中に楽譜を見つつ練習したい」と思っており、できるだけ視線移動を少なくするために、右手の演奏はディスプレイを見ていた」とコメントしている。また、被験者Bは「右手は利き手であり、左手と比べて鍵盤間の距離が鍵盤を直接見なくても予測できた」と述べており、右手は左手と比べて鍵盤のブラインドタッチ能力が高く、このような視線の動きになったと考えられる。5日目は、4日目とくらべて被験者の調子が悪かったと思われる。このため、5日目の視線は、ミスを避けるために楽譜を見る頻度が低くなった。しかし、5日目におけるトルコ行進曲の到達度テストは4日目の同楽曲の到達度テストより成績が良くなっている。5日目の30分間の練習においても着実に熟達度が向上したことがわかる。

5 まとめ

本研究では、学習者の視線に注目し、ピアノ学習支援システムの熟達化プロセスの詳細な調査を行った。成人した8名の鍵盤楽器初心者に、学習支援システムを使いながら1日30分間トルコ行進曲を5日間かけて練習してもらい、実験日ごとにシステムの補助を使わずに到達度テストを行ってもらった。

実験結果より、成人学習者は過去の成功体験から学習方略をもっており [10]、いずれの被験者も自身

の学習方略を基準にシステムを利用していた。自身の学習方略とシステムの学習方略が適合した被験者は、効率的に習熟でき、ピアノ演奏そのものの演奏技術の獲得までいたった。学習方略が部分的に適合した被験者は学習効果にばらつきが生じ、システムの学習方略に適合しなかった被験者はピアノ演奏技術の獲得まではいたらなかった。

さらに、1名の被験者に対して解析した実験日ごとの視線データの結果から熟達するにつれて被験者が必要とする情報は打鍵位置情報から楽譜情報に遷移していることが明らかになった。

今後は、音符に着目した提示情報の重要度の変遷の調査、他の被験者を対象とした実験データの解析、視線データを考慮した熟達度の算出、熟達度に応じて補助情報を適応的に提示する学習支援システムの構築などがあげられる。

参考文献

- [1] 竹川佳成, 寺田 努, 塚本昌彦: 運指認識技術を活用したピアノ演奏学習支援システムの構築, 情報処理学会論文誌, Vol. 52, No. 2, pp. 917-927 (2011年).
- [2] CASIO: 光ナビゲーションキーボード: http://casio.jp/emi/key_lighting/.
- [3] ヤマハ株式会社: 光る鍵盤 EZ-J210: <http://www.yamaha.co.jp/product/piano-keyboard/ez-j210/index.html>.
- [4] 大島千佳, 井ノ上直己: 不得手要素を克服させるピアノ学習支援システムにむけて, 情報処理学会研究報告 (音楽情報科学研究会 2007-MUS-71), Vol. 2007, No. 81, pp. 185-190 (2007年).
- [5] 森田慎也, 江村伯夫, 三浦雅展, 秋永晴子, 柳田益造: 演奏特徴の強調およびアドバイスメッセージによるピアノ基礎演奏の独習支援, 日本音響学会平成20年度秋季研究発表会, pp. 933-934 (2008年).
- [6] R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul: A Computer-Based Multi-Media Tutor for Beginning Piano Students, *Journal of New Music Research*, 19 (2-3), pp. 155-173, 1990.
- [7] S. Smoliar, J. Waterworth, and P. Kellock: pianoFORTE: A System for Piano Education Beyond Notation Literacy, *Proceedings of the Third ACM International Conference on Multimedia*, pp. 457-465 (1995).
- [8] Wikipedia, Eye movement in music reading, http://en.wikipedia.org/wiki/Eye_movement_in_music_reading
- [9] 川崎智子: ピアノ演奏時における読譜の為の眼球運動, 三重大学教育学部研究紀要教育科学, 第33巻, pp.49-66 (1982年).
- [10] W. W. Lee, D. L. Owens, 清水康敬: インストラクショナルデザイン入門-マルチメディアにおける教育設計, 東京電機大学出版局, p.38.

LumoSpheres: 複数浮遊物体の実時間追跡にもとづく映像投影

山口 裕明 小池 英樹*

概要. 空中を浮遊する大量の粒子を実時間追跡し、映像を投影することで、体積型の立体ディスプレイを実現する構想を提案し、その基礎段階として、運動している少数の球に実時間追跡にもとづいた映像の投影を行った。構築したシステムの基本的な流れは、モーションキャプチャを用いて球の三次元位置を計測し、プロジェクタを用いて球に映像を投影するという処理であるが、これら进行处理している間も球は空中を移動しているため、その時間分だけ正しい位置に映像が投影されない。この問題に対して、統計的に位置を推定するカルマンフィルタと運動モデルを適用し、軌道予測を行うことで安定した映像の投影を実現した。現段階の応用例として、トスジャグリングの球に映像を投影する DigitalJuggling を実装した。そして、パーティクル型立体ディスプレイのプロトタイプを実装し、実現するために必要となる要素を議論した。

1 はじめに

プロジェクションマッピングは、現実の物体の形状に沿ってプロジェクタから映像を投影する技法で、近年イベントやメディアアートとして盛んに用いられている。投影対象は机上、壁、床、建築物、造形物などの様々な形状であるため、その形状に合わせた映像の制作や位置合わせが重要である。そのため、プロジェクションマッピングの投影対象は、静止している物体や回転などの周期運動をする物体が使用されることが多く、自由な運動や高速な運動をする物体には一般的に用いられてこなかった。

一方、ビジョンセンサの性能やコンピュータを用いた物体認識の技術は年々進歩している。物体を追跡するモーションキャプチャシステムは、カメラ数十台を同時に使用することができ、数百 fps というフレームレートを用いて、物体の三次元座標を実時間で計測することが可能である。

このようなプロジェクションマッピングの技術と物体認識の技術を合わせて、運動している物体に対して映像を投影しようというのが本研究の目標である。静止している物体へプロジェクションマッピングを適用すると、投影範囲は物体の表面に限定されるが、運動している物体へ適用すると(物体の表面であることは同様であるが)、物体を動かす範囲だけ投影範囲を空間的に広げることができる。つまり、映像の表現範囲を平面から空間へ拡張することができる。

このような平面から空間へ映像を拡張するという考えから、図1のようなパーティクル型立体ディスプレイを提案する。パーティクル型立体ディスプレイは、大量の粒子を空中に飛散させ、それらをビジョン

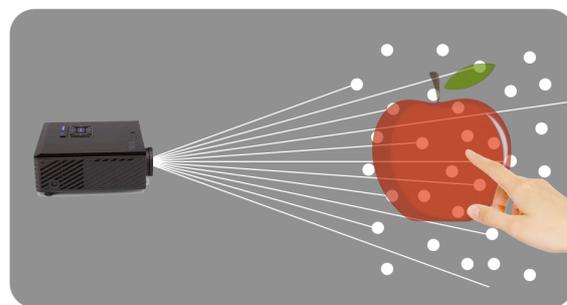


図 1. パーティクル型立体ディスプレイの概念図. 立体像に直接的なインタラクションができる裸眼立体ディスプレイである。

センサで計測し、それぞれの粒子にプロジェクタから任意の色を投影することを繰り返し行うことで実現する体積型の裸眼立体ディスプレイである。立体像が結像される領域を囲うように複数台のプロジェクタを設置すれば、360° どの方向からでも立体像を見ることができる。また、粒子が質量の小さい物質であれば、立体像が結像される空間に直接的なインタラクションが可能である。

パーティクル型立体ディスプレイは、空中に飛散している球の位置の計測と、その物体へ映像を投影するというのが基本的な処理となっている。しかし、それらの処理中も球は空中を移動しているため、計測した球の位置にそのまま映像を投影すると、装置の処理時間やデータの転送時間だけ投影位置がずれる。

本研究ではパーティクル型立体ディスプレイの基礎段階として、そのディスプレイの最小要素(平面ディスプレイであれば pixel に該当)である一つの球を用いて、遅延対策の考案と実装、その評価を行った。具体的には、放物線運動を用いた軌道予測を行うが、単純に運動モデルを用いて軌道予測をすると映像が乱れてしまう現象が発生するため、統計的に

Copyright is held by the author(s).

* Hiroaki Yamaguchi and Hideki Koike, 電気通信大学大学院 情報システム学研究科 情報メディアシステム学専攻

位置を推定するカルマンフィルタと運動モデルを合わせた軌道予測を行い、安定した映像の投影を実現した。

現段階の応用として、トスジャグリングの球に映像を投影する DigitalJuggling をアプリケーションとして示す。そして、パーティクル型立体ディスプレイのプロトタイプを実装し、実現するために必要な要素について探求する。

2 関連研究

Kimura らの “Laser produced 3D display in the air” はレーザの特性を利用した発光を空間中に行うことで、体積的なディスプレイをつくっている [1]。Butler らの “Vermeer” は特殊なスクリーンを回転させることで体積的な投影対象の空間をつくり、そこへプロジェクタから映像を投影をしている [2]。これらのディスプレイは裸眼で 360° どの方向からも立体像を見ることができ、レーザや回転機構により立体像が表れる空間が危険であるため、立体像に対する直接的なインタラクションは工夫が必要である。

FogScreen は空中に霧の層を発生させ、プロジェクタからその層に映像を投影することで映像を結像させている [3]。Nakamura らの “bubble cosmos” は煙を充満させたシャボン玉に映像を投影する [4]。Matoba らの “SplashDisplay” は空中に拭きあげた発泡ビーズに映像を投影する立体的な表現のできるディスプレイである [5]。Barnum らの “A multi-layered display with water drops” は水滴の落下とプロジェクタからの映像を同期させるディスプレイである [6]。霧ではなく水滴を用いることで、時分割に水滴を落とす制御が可能となり、4 層の体積的なディスプレイとなっている。本研究では、制御や計測のできる対象として球を用い、投げ上げや自由落下をしている球に対して映像を投影することで、層ではなく体積的な映像表現を目指す。

プロジェクションマッピング [7] は机上、壁、床、造形物など、物体の形状に合わせてプロジェクタから映像を投影する技術であり、1970 年頃からテーマパークで実用化され、近年盛んにイベントやメディアアートとして用いられる。Parker の “lumarica” は複数のひもに映像を投影する立体ディスプレイである [8]。Okumura らの “Lumipen” は高速視線制御ユニットと高速ビジョンセンサを用いて、運動している球に即時的な映像の投影を実現している [9]。高速な光軸制御により、球に張り付いたような投影ができるが、投影対象は高速視線制御ユニットに対して単数の球であり、奥行き情報を得る場合も工夫が必要である。本研究では、複数の球に対して奥行き情報を考慮した映像を投影することで、体積的な映像の投影範囲を確保する。



図 2. LumoSpheres: 運動している複数の球に映像を投影するシステム。

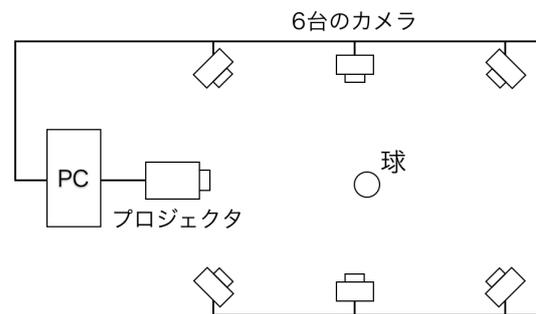


図 3. システム構成 (俯瞰図). 6 台のカメラで球を撮影し、PC で処理をした後、プロジェクタから球に映像を投影する。

3 LumoSpheres

運動している複数の球に映像を投影するシステムを「LumoSpheres」と呼ぶ。図 2 は三個の球に対して動作させている様子である。このシステムは現実空間で運動している球に、任意の映像を投影することができる。平面的なプロジェクションマッピングの空間的な拡張、スポーツの映像的な拡張、ビデオゲームなどのエンタテインメントなどに応用が可能である。図では、1.5m³ の空間の中で三個の球にシステムを動作させているが、カメラからみた球の重なり（オクルージョン）が生じない限り三個以上も可能であり、動作範囲は数十 m³ まで拡張可能である。

3.1 システム構成

LumoSpheres のシステム構成を図 3 に示す。映像の投影対象となる球、その球を動かす空間を囲むように配置した 6 台のカメラ (NaturalPoint OptiTrack S250e)、球の位置計測、追跡を行うソフトウェア (NaturalPoint TrackingTools Ver.2.5)、予測などの処理をする PC (Lenovo ThinkPad T530, Intel Core(TM) i7 2.9 GHz, 8 GB RAM, Windows7 Service Pack 1)、映像を投影をするプロジェクタ (BenQ W1070, 60Hz, 1280 × 720pixel) で構成される。カメラ、PC、ソフトウェア、予測プログラ

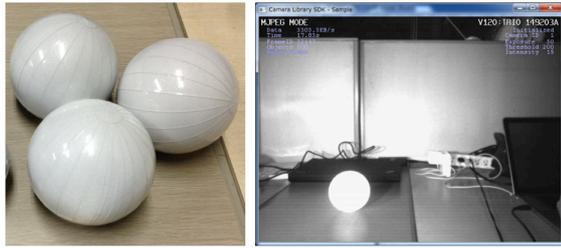


図 4. 再帰反射シートを貼り付けた球 (左) とその反射性能 (右). カメラから照射された赤外線光が再帰反射の性質により球から強く反射される.

ムとのデータ転送は UDP 通信で行われる.

モーションキャプチャの一般的な用途である, フルボディモーションキャプチャとして用いる場合は, 直径 1cm 程度の再帰反射性マーカを身体全体に装着するが, 本研究では, 任意の大きさの球に再帰反射シートを貼ることで, マーカの代わりとしている. 本システムに用いる再帰反射シートを貼り付けた直径 10cm の球と, その反射性能を図 4 に示す. シートは舟形多円錐図 (地球儀の展開図) に切り, 球に貼り付けている. 一つの球は 180g の重さがあるため, 空中を運動する際の空気抵抗は無視できる.

3.2 プロジェクタとカメラのキャリブレーション

現実空間の球に合わせて適切に映像を投影するためには, モーションキャプチャから取得できる三次元座標とプロジェクタから投影する二次元映像の位置合わせが必要である.

プロジェクタはピンホールカメラと同様のモデルとして考える事ができるため, モーションキャプチャから取得できる三次元座標を (X, Y, Z) , プロジェクタから投影する二次元映像の座標を (u, v) とすると, 3×4 の投影変換行列を用いて式 (1) のように表すことができる (式は同次座標系で表している)[10].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

この式は, キャリブレーションにより投影変換行列を求めることができれば, モーションキャプチャ座標 (X, Y, Z) が与えられた時に, プロジェクタから投影する二次元座標のどの位置に球の映像を描けばよいかを示している. 投影変換行列は 12 個の要素からなるため, 座標 (X, Y, Z, u, v) を 6 組以上与えることで求めることができる. 本システムでは, 精度を上げるため座標 (X, Y, Z, u, v) を 12 組与え, 最小二乗法により投影変換行列を求めている. また, 投影変換行列を用いてプロジェクタの外部パラメータを求めることで, モーションキャプチャ座標 $(X, Y,$



図 5. キャリブレーションによる位置合わせ.(左) と (右) は位置が 1m ほど離れているが, 球の形状に沿った映像が投影される.

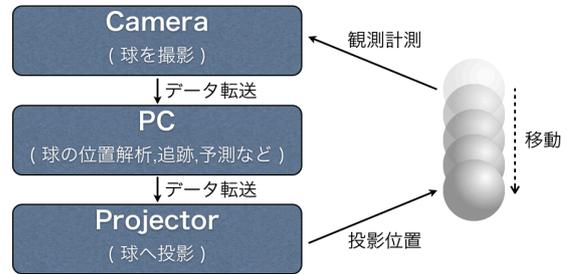


図 6. システムの流れで発生する遅延. 球をカメラで撮影してから, プロジェクタから投影するまでに遅延が発生するため, 軌道予測をする.

Z) からプロジェクタからみた球の三次元位置座標に変換することができる. この変換をすることにより, プロジェクタからみた球までの三次元的な距離がわかり, 映像を描く際のサイズを求めることができる.

キャリブレーション後に映像を投影した様子を図 5 に示す. 現実空間で異なる位置に球を移動しても, 球に合わせた映像が適切に投影される.

3.3 位置推定と運動モデルによる軌道予測

図 6 はシステムの流れを示している. 球をカメラで撮影してから, その球にプロジェクタから映像を投影する過程には, 各装置の処理時間とデータ転送時間が発生する. これらの装置の処理時間とデータ転送時間を合計した時間を, 本研究では遅延時間という. 遅延時間が発生している間も球は運動しているため, 単に取得した三次元位置に映像を投影しても, 球に正しく投影されない. そこで, 運動している球に合うように映像を投影するために, カルマンフィルタを用いて球の状態を統計的に推定し, 運動モデルを適用することで遅延の対策を行う.

カルマンフィルタは, 事前の観測値から現在や事後の確率を最大化するシステムの状態モデルを構築することができる [11]. つまり, 次々と得られる球

の座標データを用いて球の状態 (位置, 速度, 加速度) を統計的に推定することができる. 球の位置ベクトルを式 (2) と表す.

$$p_k = (X_k, Y_k, Z_k)^T \quad (2)$$

このとき, 球の状態を位置, 速度 (一つのドット), 加速度 (二つのドット) で表すと, 状態ベクトルは式 (3) と表す.

$$x_k = (p_k^T, \dot{p}_k^T, \ddot{p}_k^T)^T \quad (3)$$

球の運動は等速度運動と仮定した場合の状態モデルを式 (4) に示す (w はシステムノイズ, I は 3×3 の単位行列, Δt はデータの取得周期).

$$\begin{bmatrix} p_k \\ \dot{p}_k \\ \ddot{p}_k \end{bmatrix} = \begin{bmatrix} I & \Delta t I & \frac{1}{2}(\Delta t)^2 I \\ 0 & I & \Delta t I \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} p_{k-1} \\ \dot{p}_{k-1} \\ \ddot{p}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \quad (4)$$

観測マトリクスは式 (5) とし球の位置のみを観測する.

$$H = [I \ 0 \ 0] \quad (5)$$

これらの式を用いてカルマンフィルタの予測フェーズと修正フェーズを繰り返し行うことで, 球の状態を推定する.

カルマンフィルタにより, 現在の状態を推定したら, システム全体の遅延時間を ΔT として, 式 (6) の運動モデルにより, 球の軌道予測をする.

$$p_{k+1} = \frac{1}{2}\ddot{p}_k\Delta T^2 + \dot{p}_k\Delta T + p_k \quad (6)$$

4 投影精度の評価実験

4.1 評価の尺度

球にどの程度の映像が正しく投影されたかを, システム外部からカメラを用いて評価する. 評価の尺度として, カメラ画像からみた球全体の面積 (半球) に対する, 投影映像の面積の割合を投影精度と定義する. 例えば, 投影された面積が球全体の面積の半分ならば, 投影精度は 50 % となる.

4.2 評価のアルゴリズム

評価はカメラのグレースケール画像を以下の手順で処理することにより行う.

1. 球全体の面積を求める:
Hough 変換を用いて円抽出を行い, 円の面積 (pixel) を算出する.
2. 投影部分の面積を求める:
1 で円抽出をした領域をしきい値を用いて 2 値化し, 投影部分の面積 (pixel) を算出する.
3. 投影精度を求める:
(投影部分の面積 / 球全体の面積) を求めることにより, 投影精度を算出する.

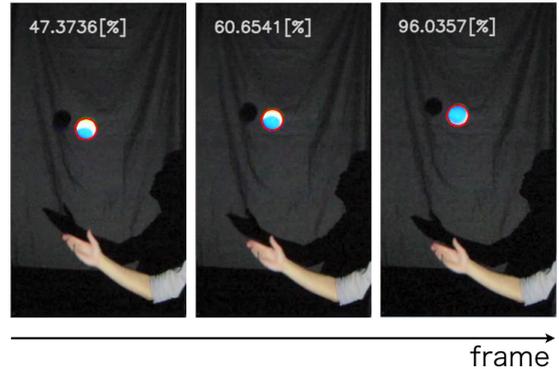


図 7. 投げ上げている最中の数フレームの投影精度. 球全体の面積と投影部分の面積の割合により投影精度が算出される.

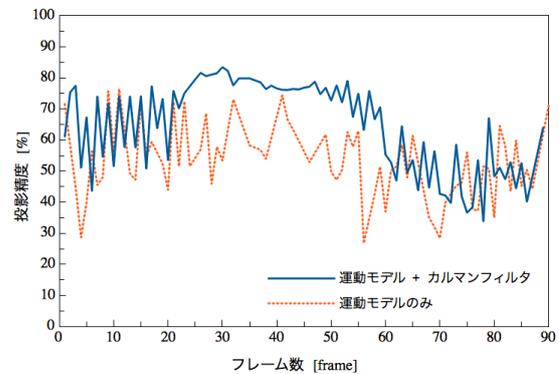


図 8. 投影精度 - フレーム数. カルマンフィルタにより位置推定をし, 運動モデルを適用することで投影精度は安定する.

しきい値は球面積の半分映像が投影されるように設定し, 球部分のヒストグラムから大津の二値化 [12] を用いて事前に決定する.

図 7 は投げ上げ中のある連続したフレームの円抽出と投影精度を表している.

4.3 評価実験の結果

実験は, 120fps のフレームレートで撮影できるカメラを使用し, 球を投げ上げてからつかむまでの一連の運動を記録した. システム全体の遅延時間は 56ms と設定し単に運動モデルを適用した場合と, カルマンフィルタと運動モデルを合わせた場合の軌道予測を比較する. 遅延時間は球の個数により数 ms 変化するが, 今回は定数として任意に決定した.

図 8 は式 (6) の運動モデルを単に適用した場合と, カルマンフィルタで位置を推定した後に運動モデルを適用した場合の投影精度 - フレーム数のグラフである. カルマンフィルタと運動モデルを合わせた軌道予測では, 投げ上げの最高点 (35frame 前後) で球の速度が遅くなるため, 投影精度は 80% 以上となる



図 9. DigitalJuggling: 本システムをトスジャグリングに応用した例. 映像の投影なし (左), 球状の映像を投影 (中), 非球状の映像を投影 (右). トスジャグリング中も動的に映像を切り替えることができる.

が, 速度が速い投げ上げ直後とつかむ直前では投影精度がおよそ 40 %から 75%となる. 単に運動モデルを適用した軌道予測では, 球がどの位置であろうが, 投影精度はおよそ 30 %から 75%となる. カルマンフィルタは過去のデータを統計的に計算するため, 計測時に発生するノイズの影響に強く, 単純に運動モデルを用いた場合と比較して安定した投影となった. 球の速度が速い時に投影精度が振動するのは, 球は時間的に連続して動いているにも関わらず, プロジェクタが 60 Hz で動作しているのが原因である.

5 アプリケーション

本システムは, プロジェクションマッピングやディスプレイの空間的な拡張, スポーツの映像的な拡張, ビデオゲームなどのエンタテインメントなど, コンテンツ次第で様々な応用が考えられる. ここでは応用例として, LumoSpheres をトスジャグリングへ応用した DigitalJuggling, パーティクル型立体ディスプレイのプロトタイプを示す.

5.1 DigitalJuggling

図 9 に DigitalJuggling の様子を示す. システムをトスジャグリングに応用した例である. 映像を投影しない場合は通常のジャグリング球であるが, 投影する映像次第で様々なテクスチャの球でトスジャグリングをしているように見える. テクスチャは, ビリヤードの球のような球状でも, 蝶のような非球状でもよい. 通常のジャグリング球を用いた場合と異なり, 動的なテクスチャの変化が可能である.

5.2 パーティクル型立体ディスプレイ

図 10 にパーティクル型立体ディスプレイのプロトタイプを示す. 小球は直径 10cm の球と同様に, 直径 4cm のピンポン球に再帰反射シートを貼り付けている. 12 個の小球は, 人の手によって空中に飛散

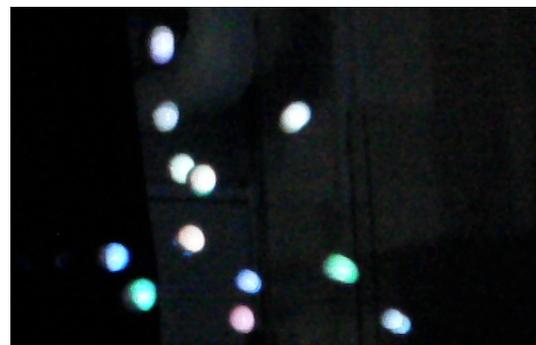


図 10. パーティクル型立体ディスプレイのプロトタイプ. 12 個の小球に割り当てた色を投影している.

させ, ランダムに色を割り振られた映像が投影されている.

6 議論: システムの現状と利点, 問題点

LumoSpheres システムの構築を通して, 運動する少数の物体に対して映像を投影する手法を探索した. ここでは, パーティクル型立体ディスプレイを実現する上で必要な要素を議論する.

6.1 プロジェクタのフレームレート

本システムは球の運動速度が速くなるにつれて, 投影映像のちらつきが生じる. これは, 使用したプロジェクタの映像更新フレームレートが 60 Hz であることが原因である. 現実空間の球の運動は連続的であるのに対し, プロジェクタの映像更新は不連続である. 映像が更新された直後は球に映像が合っていない, 次の更新までに球が移動するため映像がちらついてみえる. これを解決するためには, 単純にプロジェクタのフレームレートを上げる, または, Okumura らの研究 [9] のように高速に光軸制御をする装置を用いることで, プロジェクタからの映像

の更新を連続的に近づける必要がある。

6.2 物体のオクルージョン

本システムは少数の球であれば映像を投影できるが、多数の球になると正確な位置に映像が投影されない。これはカメラの画像で球同士の重なりが生じると、カメラ同士でステレオマッチングができなくなり、位置を正しく計測できなくなるのが原因である。これを解決するためには、単純にカメラの台数を増やす、または、単一の球のみではなく複数の球の相互関係を考慮した予測を行うことで解決できる。これは、球同士の衝突により生じる急激な運動の変化に対しても有効であると考えられる。

6.3 遅延対策としての予測

装置の処理時間やデータの転送時間が生じない理想的な装置であれば、軌道予測をする必要はなく、計測した球の位置に瞬間的に映像を投影すればよい。しかし、今後もコンピュータやビジョンセンサの進歩はあるものの、遅延がなくなることはない。そのような意味で、Okumuraらの研究[9]のようにセンサビジョン部分のハードウェアを工夫して高速にするというアプローチは重要であるが、コンピュータ部分のソフトウェアを工夫した軌道予測によるアプローチも意義のあるものであると考える。

6.4 想定される構成と利点

以上を踏まえて、パーティクル型立体ディスプレイの想定される構成は、連続してパーティクルを空中に飛散させる装置、その装置の周囲を囲むように配置した複数のカメラとプロジェクタ、そしてパーティクル全体をシミュレーションするプログラムが必要である。本構想の最大の特徴は、直接的なインタラクションが可能な裸眼立体ディスプレイであるが、その他の利点として、球に貼り付けた再帰反射シートの性質を利用して、同じディスプレイ空間で複数人が異なる映像を見ることができ、再帰反射の性質により複数台のプロジェクタから投影された映像が混合しないことを利用する。これは、空中で色を発光させるような構造[1]では実現できない。

7 結論と今後の方向性

運動している複数の球に実時間追跡にもとづき映像を投影することで体積型の立体ディスプレイとなる、パーティクル型ディスプレイを提案し、その基礎段階として、運動している少数の球に実時間追跡にもとづいた映像を投影するシステム、LumoSpheresを構築した。そして、LumoSpheresシステムの応用例としてDigitalJugglingを実装し、パーティクル型立体ディスプレイのプロタイプを実装した。

今回の研究では、パーティクル型ディスプレイの基礎段階として、球一つ一つに映像を合わせて投影

する手法を探求した結果、議論で述べた通り、様々な問題が浮かび上がってきた。今後はこれらの問題点を改善しつつ、多くの物体へ安定した映像が投影できるような手法を更に探求していきたい。

参考文献

- [1] Hidei Kimura, Taro Uchiyama, and Hiroyuki Yoshikawa. Laser produced 3D display in the air. In *ACM SIGGRAPH 2006 Emerging technologies*, July-August, 2006.
- [2] Alex B., Otmar H., Shahram I., Steve H., David M., David K., and Danny K.. Vermeer: direct interaction with a 360° viewable 3D display. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, October, 2011.
- [3] FogScreen. <http://www.fogscreen.com/>.
- [4] Masahiro Nakamura, Go Inaba, Jun Tamaoki, Kazuhito Shiratori and Junichi Hoshino. Mounting and application of bubble display system. In *Proceedings of the 2006 ACM SIGCHI international conference*, June, 2006.
- [5] Yasushi Matoba, Taro Tokui, Ryo Sato, Toshiki Sato, Hideki Koike. SplashDisplay: volumetric projection using projectile beads. In *Proceedings of the 2012 Virtual Reality International Conference*, March, 2012.
- [6] Peter C. Barnum, Srinivasa G. Narasimhan, and Takeo Kanade. A multi-layered display with water drops. In *ACM Transactions on Graphics (TOG)*, v.29 n.4, July, 2010.
- [7] Mine, M., Rose, D., Yang, B., Jeroen van Baar, and Grundhfer, A. Projection-Based Augmented Reality in Disney Theme Parks. *IEEE Computer* 45, 32-40, July, 2012.
- [8] Matthew Parker. Lumarca. In *ACM SIGGRAPH ASIA 2009*, December, 2009.
- [9] Kohei Okumura, Hiromasa Oku, and Masatoshi Ishikawa. Lumipen: Projection-Based Mixed Reality for Dynamic Objects. In *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, p.699-704, July, 2012.
- [10] Richard Hartley, and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [11] Greg Welch, and Gary Bishop. *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill, 1995.
- [12] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9, No. 1, pp. 62-66, 1979.

画像照合に基づく多メディア間の話題追跡システム

豊田 正史 伊藤 正彦* Cai-Zhi Zhu 佐藤 真一 宇野 毅明† 喜連川 優‡

概要. 本論文では、Web と放送映像との間の画像の対応関係に基づき話題の追跡を行える新たなシステムを提案する。社会事象を解析する際には、話題がどのメディアから始まったかを同定し、その話題がどのようにメディア間に広がっていったかを分析する必要がある。Web と放送映像との対応を取るため、Web 上のテキストと放送の字幕情報を用いることも可能であるが、字幕の量は十分ではなく、字幕が付けられていない番組も多数存在するため、映像情報の対応関係を解析する事が不可欠となる。提案するシステムでは、ブログ等の Web メディアに投稿された大量の画像から大きな話題になり多数出現した類似画像を抽出し、これが放送映像を起源とするものかどうかを放送映像アーカイブと照合して、話題の時差や露出の差を調査できる手法を実現した。本システムを用いて、大規模なブログアーカイブと放送映像アーカイブを分析した事例を幾つか示すことで本システムの有用性を示す。

1 はじめに

Web はその出現から約 20 年の間に、ニュースサイト、掲示板、ブログ、ソーシャルメディア、コンテンツ共有サイトなど多数のメディアの集合体として進化を続けてきた。現在は、これら Web 上のメディアとテレビや新聞などの従来型マスメディアを包含する多メディアが人々の社会活動を反映し、かつ影響を与えていると言える。こうした状況において、社会事象を分析する際には話題がどのメディアから出現したかを同定し、その話題がどのように多メディアの間に広がっていったかを分析する必要がある。現在においても Web メディアはマスメディア、特にテレビ放送から受ける影響が大きく、Web メディアとテレビ放送の間での話題追跡は重要な課題である。

従来、Web メディアとテレビ放送での話題分析においては、Web 上のブログやソーシャルメディアなどの記事および番組表や字幕など、テキスト情報を用いたものが主であった。しかし、これらのメディアでは積極的に映像・画像を用いることで、文章のみでは伝えることが困難なイベントや事件事故などの様子を伝えており、また字幕の量は十分ではなく、字幕が付けられていない番組も多く存在するため、映像情報の対応関係を解析する事が不可欠となる。

Web 上で話題になった画像が、放送映像中では同じように現れているのか、また、その出現に前後関係はあるのか、話題や画像ごとに出現の仕方に違いや類似性があるのか、逆に特徴のある画像群は具体的にどのような話題を扱っているのか等を調査することにより、以下のような社会事象や多メディア

間で及ぼされる影響の分析が可能になる。

- ニュースが各メディアにおいてどのような画像で扱われていたか、メディアにより現れる画像に違いがあるかを調査する。
- イベントや集会などの模様や賑わいを視覚的に把握することで、メディアによる扱われ方の差があるかどうかを調査する。
- 商品画像の出現数の変化から、人気の度合いやデザインの変化を把握する。

本論文では、Web メディアに投稿された大量の画像から大きな話題になり多数出現した類似画像を抽出し、これが放送映像を起源とするものかどうかを放送映像アーカイブを照合して抽出し可視化することで、話題の時差や露出の差を調査できる手法を提案し、それを実現するシステムを実装する。本システムを、大規模なブログアーカイブと放送映像アーカイブに適用し、実際に観測された社会事象の事例を示すことで本システムの有用性を示す。

以下、第 2 節では関連研究について述べ、第 3 節で提案システムの詳細を説明する。第 4 節において本システムを用いた実験結果を示し、第 5 節で本論文をまとめる。

2 関連研究

従来、Web メディアおよびテレビ放送における話題分析においては、単独のメディアを用いたものが多く、複数メディアを扱ったものでも Web 上のブログやソーシャルメディアなどの記事、および番組表や字幕などテキスト情報を用いたものが主であり、大規模な画像照合を用いて多メディア間の情報追跡を行っているものは筆者の知る限りでは存在しない。

Copyright is held by the author(s).

* 東京大学生産技術研究所

† 国立情報学研究所

‡ 東京大学生産技術研究所・国立情報学研究所

Adarらは、検索エンジンの検索クエリとテレビ番組情報の出現時差の解析を行っており [1], Yangらは、ブログ、新聞、テレビ等の複数メディアにおける話題の時系列パターンを抽出しているが [7], これらは基本的にテキストデータに現れる語やフレーズの出現頻度を用いたもので画像情報は用いられていない。Luoらは、大規模なニュース画像アーカイブの可視化、および話題分析を行っているが [5], 複数のメディアにまたがる分析までは行えていない。

3 多メディア間話題追跡システム

本論文で提案するシステムは、大規模な Web メディアのアーカイブを用いてユーザの指定した調査対象に関連する記事および画像を抽出し、さらに放送映像のアーカイブを用いて画像照合を行い、抽出した Web 画像に対応する放送映像での露出を抽出することで、これら画像の頻度、時間推移、および出現時差を調査可能にするものである。本システムは、Web メディアアーカイブとして、ブログやソーシャルメディアのように、各記事に対して書き手と書かれた時間が紐付けされた文書アーカイブを想定しており、全文検索インデックスによって記事が検索可能となっているものとする。また、放送映像アーカイブに関しては、画像を入力としてその画像に類似する映像が放送されたショットを検索可能とする検索インデックスを実現して用いている。

ユーザは、調査対象に関する複数の検索語を入力としてシステムに与え、システムは、以下のような手法で、画像抽出、分析、可視化を行う。

1. Web メディアアーカイブを検索し、与えられた検索語を含む記事の集合を取得する。各記事はそれが書かれた時間、書き手、および記事本文を含む。
2. 取得した記事の本文に含まれている画像をすべて取得し、各画像の周辺のテキストを抽出する。
3. Web メディア上で話題となり数多くコピーされた画像を抽出するため、ほぼ同一とみなせる画像のペアを列挙しクラスタリングを行う。この結果得られる Web 画像集合は単独の画像が多数出現したことを表し、画像に基づく話題を構成する要素となる。
4. 抽出した各 Web 画像集合を検索クエリとして用いて、放送映像アーカイブを検索し、類似画像が放送されたショットの集合を抽出する。抽出された各ショットの集合を、便宜上 Web 画像集合に対応する放送画像集合と呼ぶ。
5. Web 画像集合を、それらの周辺テキストの類似度を用いてクラスタリングすることで、高頻度で出現する画像の類似集合である Web 画像

トピックを抽出する。さらに Web 画像トピックに含まれる Web 画像集合群に対応する放送画像集合を集約することで、放送映像トピックを抽出する。こうして得られた Web 画像トピックとそれに対応する放送映像トピックの時間推移を 3 次元空間に可視化し、メディア間におけるトピックの推移およびその差を閲覧探索可能とする。

以下では、ほぼ同一画像のクラスタリング、放送映像検索の実現方法、トピックの 3 次元可視化手法についてより詳細に説明する。

3.1 ほぼ同一画像のクラスタリング

Web 上の画像は、オリジナルの画像に対し拡大、縮小、回転や、切り取り、一部変更などの加工を経て公開されるものが多く、ほぼ同一の画像をクラスタリングするための類似度算出にはこうした加工に対し頑健な特徴量を用いる必要がある。本手法ではこうした状況に適した局所特徴量の 1 つである SIFT 特徴量 [4] を用いている。SIFT を用いた画像類似度算出では、各画像から抽出された多数の SIFT 特徴点の類似度を算出し、画像間でマッチする特徴点の数をカウントする必要がある。画像数は通常多数に及ぶため類似する SIFT 特徴点のペアを高速に列挙する手法を開発した。本列挙手法は、宇野による大規模部分文字列マッチングアルゴリズム [6] を基にノイズとなる特徴点の除去処理を加えて改良を施したものである。

各画像からは、サイズの正規化を行った上で、最大 100 個の SIFT 特徴点を抽出する¹。SIFT 特徴点は、128 次元の数値ベクトルで表わされるが、各値を中央値以上か、それ以下かの 2 値に量子化し、128 ビットのベクトルに変換する。さらにこの数値ベクトルは 8 個の値からなる 16 のブロックから構成されることを考慮し、各 8 ビットを 1 文字とみなすことで書く特徴点を 16 文字の文字列と考える。宇野の類似文字列列挙手法は、大量の固定長文字列から、ハミング距離が一定以下の文字列ペアを高速に列挙するもので、本手法ではほぼ同一の画像を抽出するため、ハミング距離 1 以下の類似特徴点のペアをマッチしたものと見做して列挙し、画像間でマッチする特徴点の数を、画像ペアをキーとしたハッシュ表を用いて算出する。2 つの画像の類似度は、画像間でマッチする特徴点の数で表わされる。

SIFT 特徴点を用いた類似画像抽出においては、非常に多数の特徴点とマッチする特徴点があり、ノイズとして類似度の精度に悪影響を与えることが分かっ

¹ ここでは特徴点の最大数を経験的に 100 個に定めたため、これが最適であるとは限らない。しかし最大数をこれ以上大きくすると、ブログ用の縮小画像や携帯電話などで撮影した小さな画像に関して特徴点の数が多くなりすぎノイズが増すことが実験から分かっている。

ており、本手法では、類似ペア列挙の途中段階で、閾値以上の特徴点とマッチする特徴点をフィルタする処理を加えることで、ノイズに対して頑健な類似画像抽出を実現している。ノイズとなる特徴点の数は非常に少ないため、計算量を大きく削減するものではないが、大量の類似特徴点のペアを出力した後フィルタする必要がなく直接類似画像ペアを出力できるため、中間出力を削減することができる。本手法は、4百万画像から抽出した4億のSIFT特徴点の集合を用いて類似画像ペア(閾値1000の場合、約1.35億ペア)を列挙するのに、10GBのメモリと20コアを用いて2時間程度しかかからないスケラビリティを持っている。

類似する画像のペアが列挙できると、画像をノード、類似度をエッジとしたグラフを作ることが可能となる。本手法では、このグラフに対して、スタークラスタリング[2]という簡易なクラスタリング手法を適用することで、画像集合を抽出している。スタークラスタリングは、次数の一番高いノードを中心とし、閾値以上の類似度を持つ隣接ノードを、中心と同じクラスタとして出力するという処理を繰り返す単純なものである。本手法では、あるノードについて後からより類似度の高い中心が見つかった場合、元の中心からその中心へ所属を切り替えるという簡易な改良を施して用いている。スタークラスタリングの計算時間はノード数とエッジ数の和に比例しており、大規模なデータをインタラクティブにクラスタリングするのに適している。また、類似度の低いノードをあえてクラスタに所属させることは無いため、ノイズの多いデータのクラスタリングにも適している。こうした理由から本手法ではスタークラスタリングを用いているが、他のクラスタリング手法を用いても良い。

3.2 放送映像の検索手法

放送映像アーカイブ中のショットを任意の画像により問い合わせ可能な、問い合わせ画像中と同一の物体を含む映像を検索可能なシステムを[8]の手法に基づき構築した。まずオフライン処理として、放送映像アーカイブを構成する各映像クリップから、非常に密にフレーム画像を切り出し(例えば3フレームごとに1フレーム)、各フレーム画像から密にSIFT等の局所特徴量を抽出する(たとえば各フレーム画像から数百点、SIFTもしくはcolor SIFTを使用)。局所特徴量は階層的k-meansにより、100万コードブックというきわめて疎な離散化を行い、各クリップにつき一つのBag of Feature (BoF)ベクトルを得る。次いで、検索時のオンライン処理では、与えられた問い合わせ画像から、同様にBoFベクトルを得る。これとデータベース中のBoFベクトルとの間でL1距離を求め、ランキングリストを得ることにより、検索を実現している。検索は転置

索引により効果的に高速化可能である。

この方法で実際に検索を行うと、大規模なデータベースが対象となった場合、検索結果にノイズがきわめて多くなってしまふことが判明した。そこで、幾何整合性判定によりリランキングを行う技術[3]を導入して、精度の向上を図っている。

3.3 多メディア間のトピック可視化・閲覧手法

本システムでは、Webメディアおよび放送映像から抽出されたトピックの時系列推移を可視化することで、多メディア間の影響分析を視覚的に可能にする新たな可視化手法を提案する。

図1に本可視化手法の概要を示す。平面上に、時間軸とトピックの軸を持ち、各時間軸上にWebメディアアーカイブから抽出されたWeb画像集合、および放送映像アーカイブから抽出した対応する放送映像集合をヒストグラムとして可視化している。画像は単位時間ごと(例えば1日ごと)に集約され、その時間中出现した画像を高さ方向に積み上げることで、画像による棒グラフから構成されるヒストグラムとして表現される。

画像の出現の時間差を認識しやすくするため、Webメディアと放送映像の画像ヒストグラムは、図1のように並列して配置することもでき、また、図3のように上下にヒストグラムを反転させて配置することも可能である。

上記の手法で抽出されたWeb画像集合は、さらにその周辺テキストを用いて意味的にトピックにまとめることも可能である。図1では、各Web画像集合に含まれる画像の周辺テキストをまとめて1文書とみなし、tf-idfで重み付けした単語頻度ベクトルを算出し、コサイン類似度を用いて上記のスタークラスタリングを行い、Web画像トピックにまとめている。Web画像集合のクラスタリングに関しては、スタークラスタリングの閾値をインタラクティブに変更しながらクラスタリング結果を閲覧するシステムを用意し、適切な閾値を選択可能としている(図2)。こうして得られたWeb画像トピックに対し、放送映像側には各Web画像集合に対応する放送映像集合をまとめて放送画像トピックとして表示している。

ユーザは、Web画像数およびショット数の時系列に対する交差相関係数、コサイン類似度等のレンジをパラレルコーディネートを用いたダイナミックエリにより対話的に指定することで、様々な時間差をもつ画像クラスタを抽出し、容易に詳細を探索することが可能である。交差相関係数を用いると、指定したメディアで先行して(もしくは遅れて)現れた画像集合を抽出することができ、放送先行もしくはWebメディア先行で流行した話題を抽出することが可能になる。時系列間のコサイン類似度が高い画像集合を抽出することで出現タイミングが似てい

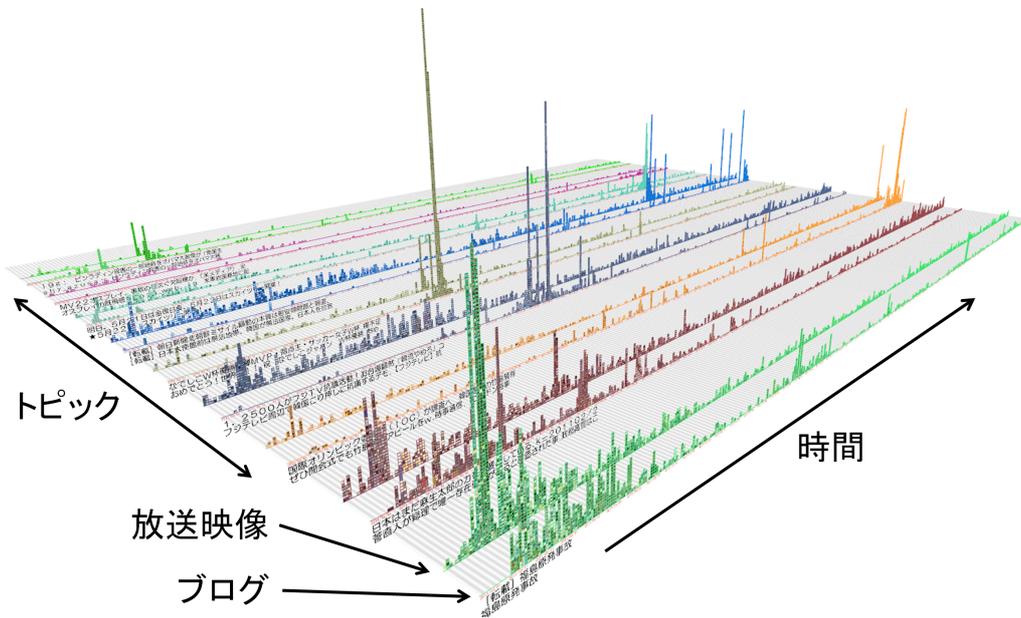


図 1. 複数トピックの推移の概要を可視化

る話題を抽出でき、類似度が低い画像集合からは、放送もしくは Web メディアのみで流行した話題を抽出することができる。

4 大規模ブログアーカイブおよび大規模放送映像アーカイブを用いた実験

4.1 データセット

前節のシステムを実装し、我々が東京大学生産技術研究所において収集している大規模ブログアーカイブおよび大規模放送映像アーカイブを用いた実験を行った。Web メディアとしては、我々が 2006 年から収集している延べ 200 万ブログを対象とした RSS/Atom フィードのアーカイブを用いた。特に、社会分析のターゲットとして 2011 年 3 月から 2012 年 9 月に起きた、東日本大震災およびそれに引き続く原発事故、サッカー女子ワールドカップ、北朝鮮ミサイル問題、ロンドンオリンピックなどを想定して、検索用のキーワードを約 90 個程度選出し、アーカイブから抽出したブログ記事に含まれる約 400 万の画像を実験データとして用いた。

また、国立情報学研究所においてアーカイブした、2011 年 3 月より 2012 年 9 月までに放送された全チャンネルのニュース番組、合計約 6000 時間分を対象として検索インデックスを構築した。検索インデックス構築対象の 6000 時間分のニュースに対

し、連続フレーム間の色ヒストグラム差分に基づくショット分割を行い、約 300 万ショットを検出した。これが検索対象の単位となる。さらに、各フレーム単位での詳細な物体検索を実現するため、1 秒おきに約 2000 万キーフレームを抽出した。局所特徴量数は約 200 億となる。こうして構築したデータベースに対し、ブログから想定したターゲットに関連するキーワードに基づいて抽出した 4000 の Web 画像集合 (46000 画像) を問い合わせとして照合を行い、相互の影響解析につながる根拠となるような画像が数多く検出されたことを確認した。

4.2 事例

本節では上述のデータセットを用い、提案システムを用いて抽出した多メディア間の話題追跡の事例を示す。

図 1 は、震災、原発、オリンピック等の検索語でブログアーカイブから抽出した Web 画像トピック、さらに放送映像アーカイブから抽出した対応する放送画像トピック、それぞれをトピック毎、一日毎に集約し時間軸上に可視化している。全体像を俯瞰 (図 (a)) することで、トピック間の類似性や、ブログ・放送映像間で同時にバーストしている期間などを観測できる。さらに、時系列に対する交差相関係数、コサイン類似度等から様々な反応差をもつ画像集合を対話的に抽出し詳細を探索することが可能である。

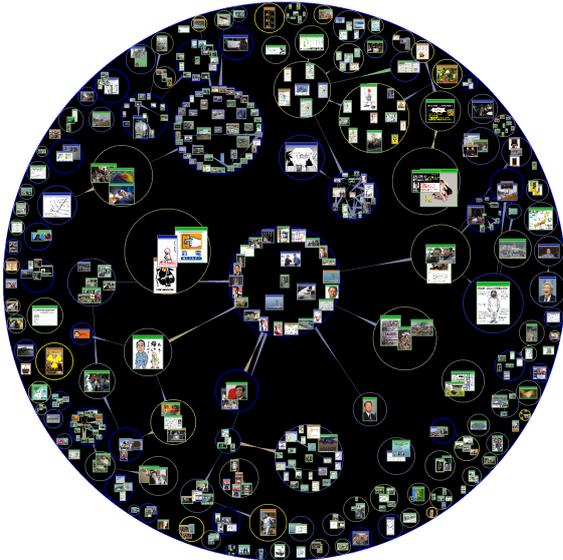


図 2. Web 画像集合のクラスタリング結果の可視化

この中から、東日本大震災の際の福島第一原発事故に関するトピックに着目し、拡大したものが図3である。このトピックに含まれる画像の多くは、事故のあった建屋を撮影したもので、画像ヒストグラムの形状、及び写真の出現順序を見ることで、これらがニュース番組において繰り返し使用され、その後ブログ上でも広まったことが見て取れる。同種の写真は、放送のみならず新聞等のメディアにおいても使用されており、この場合においてはマスメディアが先行して写真を掲示し、それがブログに広まったと解釈することができる。

本システムのダイナミックエリ機能を用いると、このトピックの中からブログが先行して用いた画像があるかどうかを調べることができる。図4は、交差相関係数を用いてブログが先行している画像クラスを抽出したものである。この画像は、MITの研究者によるブログにおいて福島第一原発事故解説に用いられた原子炉の透視図である。本記事が日本のブログにおいて翻訳され話題となり、その後2日遅れてニュース番組で用いられるようになったことが分かる。番組作成者が実際にブログを見てこの画像を発見したかどうかまでは不明であるが、これは画像の照合を用いることで初めて発見できる映像素材の原典追跡の事例となっている。メディア社会学と呼ばれる学問分野においては、映像の原典を探ることが重要なテーマの一つとなっており、本手法はそうした調査に有用である。

テレビとブログとは、同じ話題に対する反応の大きさや勢いが異なり、それは国際問題、政治問題において顕著であることが今回の実験結果から分かっている。図5はその一例を示したものである。2012年のオリンピックにおいて、試合の終了時にある選

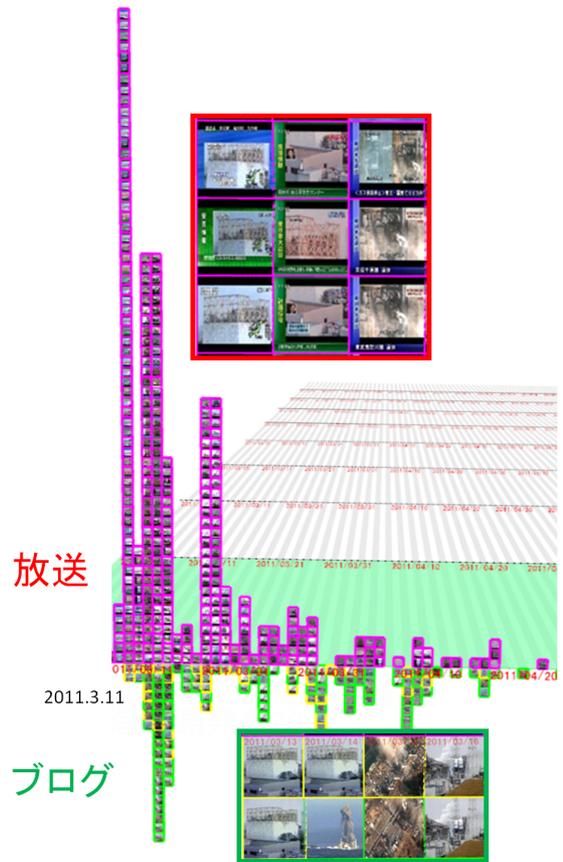


図 3. 福島第一原発事故に関するトピックのメディア比較

手が政治的なアピールをしたことが問題となった。そのニュースは最初テレビで放送されたが、その画像の露出は少なかった。しかし、このニュースは多くのブロガーを刺激しブログ上ではこの画像が大量に使われることとなった。この大きな反応を受けて、テレビのニュースでもこの問題が大きく扱われるようになった。こうした、テレビとブログが相互に反応して話題が大きくなるケースは、センシティブな国際問題や政治問題において良く見られるパターンである。

5 まとめ

本論文では、ブログ等の Web メディアに投稿された大量の画像から大きな話題になり多数出現した類似画像を抽出し、これが放送映像を起源とするものかどうかを放送映像アーカイブを照合して、話題の時差や露出の差を調査できるシステムを実現した。さらに大規模なブログアーカイブと、放送映像アーカイブを用いて実験を行い、様々な社会現象に関する多メディア間の情報追跡を行えることを示した。

本システムを用いることで、過去に起きた様々なイベントに対しての Web メディアと放送映像での反応の違いを学ぶことが可能であり、得られた知見



図 4. ブログがテレビ放送に先行した画像の例

は今後同種のイベントが起きた際にどのような反応が各メディアで起こるかを予想する際の手がかりとなり得る。また、一方のメディアで話題になっているのに、もう一方のメディアでは大きな話題にならないケースも、幾つか本システムで観測されており、その原因を探ることで企業や組織の広報活動に活用することも可能である。ここで示した事例は、本実験で得られた知見のごく一部であるが、本システムを用いることで多メディアに関わる様々な興味深い社会事象を捉えられる可能性を示していると信じる。

現状では、Web 画像集合と対応する放送画像集合が算出された後はある程度インタラクティブに閲覧・探索が行えるが、そのための事前計算には依然としてコストが高い。画像の特徴量抽出や放送映像データベースの索引を事前に作成しておくことはできるものの、画像類似度の算出や、対応画像の検索はインタラクティブに行えるほどではないため、一層の高速化が必要である。

謝辞

本研究は、文部科学省「多メディアWeb解析基盤の構築及び社会分析ソフトウェアの開発」、およびJSPS 科研費 25280111 の助成を受けたものです。

参考文献

- [1] E. Adar, D. S. Weld, B. N. Bershad, and S. D. Gribble. Why We Search: Visualizing and Predicting User Behavior. In *Proceedings of WWW'07*, pp. 161–170, 2007.
- [2] J. A. Aslam, E. Pelekhov, and D. Rus. The Star Clustering Algorithm for Static and Dynamic Information Organization. *Journal of Graph Algorithm and Applications*, 8(1):95–129, 2004.
- [3] K. Lebeda, J. Matas, and O. Chum. Fixing the Locally Optimized RANSAC. In *Proceedings of BMVC*, 2012.
- [4] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [5] H. Luo, J. Fan, J. Yang, W. Ribarsky, and S. Satoh. Analyzing Large-Scale News Video Databases to Support Knowledge Visualization and Intuitive Retrieval. In *Proceedings of VAST'07*, pp. 107–114, 2007.
- [6] T. Uno. Multi-sorting algorithm for finding pairs of similar short substrings from large-scale string data. *Knowledge and Information Systems*, 25(2):229–251, 2010.
- [7] J. Yang and J. Leskovec. Patterns of Temporal Variation in Online Media. In *Proceedings of WSDM'11*, 2011.
- [8] C. Z. Zhu and S. Satoh. Large Vocabulary Quantization for Searching Instances from Videos. In *Proceedings of ICMR'12*, pp. 52:1–52:8, 2012.

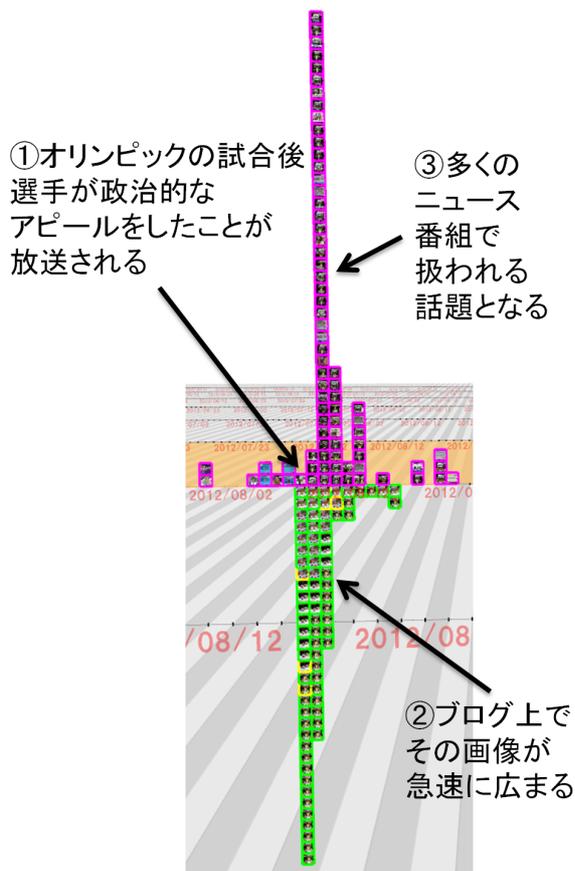


図 5. テレビとブログが相互に影響を与えた例

静止画内物体への変形指示による動画検索

川手 裕太 岡部 誠 尾内 理紀夫 平野 廣美*

概要. 我々は動画から視聴したいシーンを素早く、容易に検索するためのユーザインタフェースを提案する。既存の動画共有サイトで使用されている検索エンジンはテキストを使用したものであるため、視聴したいシーンの様子までは検索することができないといった問題点が存在する。我々のシステムでは、ユーザが動画内の一時停止フレーム等の静止画を入力し、静止画内の物体の変形を矢印2本のスケッチによって指示することができる。その後、システムはユーザが矢印で指定した変形に類似したシーンを検索し、ユーザは目的のシーンを視聴することができる。我々の手法では、ユーザが視聴したい目的のシーンを検索するために2つのステップを採用している。第1ステップでは標準的な画像検索技術を使用し、入力された静止画内の物体と同じ外観のフレームを検索する。第2ステップでは前ステップで選択したフレームから自動的に前後フレームへと早送りと巻戻しを行うことで、目的のシーンを検索する。早送りと巻戻しを行い、ユーザが2本の矢印で指定した変形が発見された場合、早送りと巻戻しを停止させ、目的のシーンを出力する。この2つのステップを行うことにより、静止画内物体が横向きにもかかわらず、外観の異なる正面向きの物体が映るシーンを検索するといったことを可能にした。我々はこの手法がF1カーと馬、飛行機の3種類の物体において有効であることを確認した。

1 はじめに

動画共有サイトの普及により、インターネット上で視聴することができる動画の数は日に日に増加している。ユーザはそれらの動画から視聴したいシーンを迅速かつ容易に検索したいと考えている。ただし、YouTube¹ やニコニコ動画² などの動画共有サイトの動画検索エンジンは、テキストでしか検索することができない。その検索エンジンでは、各動画のタイトルや割り当てられているテキストタグの情報に基づいて検索が行われている。従って、これらの検索エンジンで、検索項目として視聴したい物体の様子に関するテキストを入力しても検索できない。例えば、「左上に飛んで行く飛行機」というテキストに対して、YouTube やニコニコ動画ではいずれも、図1のような左上方向へ飛行する飛行機のシーンを取得することはできない。さらに、そのようなテキストを理解できる動画検索エンジンがあったとしても、ユーザが物体の様子を言葉で正確に記述することは困難であり、面倒である。

このような問題を解決するために、我々はユーザが迅速かつ容易に視聴したい物体の様子を指定することができる新しいユーザインタフェースを提案する。ユーザインタフェースでユーザが検索を行う際の流れは以下の通りである。

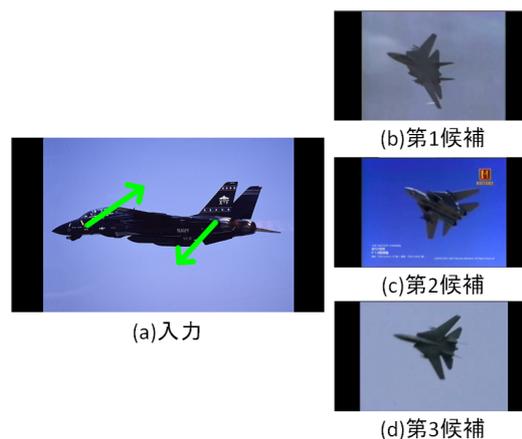


図 1. 提案システム

1. 視聴したい物体の映る動画の一時停止のフレームまたは静止画を入力静止画とする。
2. スケッチインターフェース (図1(a)) にて、2本の緑矢印の描画で入力静止画内の物体の変形を指定する。
3. システムが動画データベースから検索結果の候補のシーンを表示する (図1(b,c,d)).

ユーザは2本の緑矢印で入力静止画内の物体の変形を指示することで視聴したいシーンをインタラクティブに検索することができる。また、この2本の緑矢印は2次元の入力であるが、横向きの入力静止画から正面向きのシーンを検索するといった、3次元で変形させたようなシーンも検索することが可能

Copyright is held by the author(s).

* Yuta Kawate, 電気通信大学, Makoto Okabe, 電気通信大学/JST CREST, Rikio Onai, 電気通信大学, Hiromi Hirano, 楽天株式会社/電気通信大学

¹ <http://www.youtube.com/>

² <http://www.nicovideo.jp/>

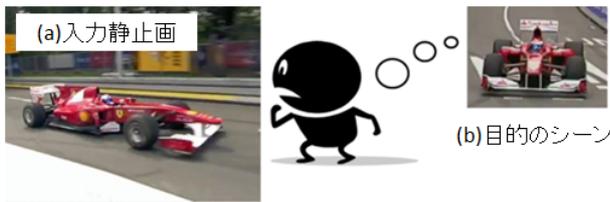


図 2. モチベーション

である。実験において、この方法がF1カー、馬、飛行機といった複数の種類の物体に有効であることを確認した。

2 関連研究

多くの動画からインタラクティブな検索を可能にする外観ベースのインタフェースの提案はなされている。bag-of-features を使用し、比較的次元のベクトルとして各動画のフレームを表すことにより、入力した静止画から静止画や動画の検索が提案された [3, 15]。多くの動画や静止画から構成された3次元のシーンをユーザがインタラクティブに動きまわる事ができるシステムが提案された [1, 5, 16, 17]。これらは structure-from-motion を介してシーンやカメラの位置を3次元再構成を行なっている。しかし、これらの方法は、建物などの静止物体にしか適用されておらず、移動したり、変形したりする物体に適用することは困難であり、非常に計算時間がかかる。

動画内の物体を操作することで、連続した単一の動画内のシーンのインタラクティブなナビゲーションを可能にするシステムが提案された [4, 7, 8, 9, 10]。ユーザは、単一動画内の物体をドラッグによってインタラクティブな姿勢の操作ができ、ナビゲートできる。これらは、3次元再構成ではなく、2次元の画像処理技術が行われているので、比較的計算時間が少ない。しかし、我々のシステムは単一の動画だけではなく、複数の別々の動画から物体を検索することができる。

3 ユーザインタフェース

我々の提案するユーザインタフェースは、ある物体の動画の視聴時に別のシーンが見たいといった、物体の様子を検索するシステムである。例えば、図2(a)のようなF1カーが左から右に走行するシーンの視聴している時、図2(b)のように正面向きに走行するF1カーのシーンを視聴したいという状況を考える。ユーザはそのようなシーンを視聴するために、早送りや巻き戻しを行うか、シークバーを操作する。もし、今視聴している動画にそのようなシーンが含まれない場合、ユーザが他に所持しているF1

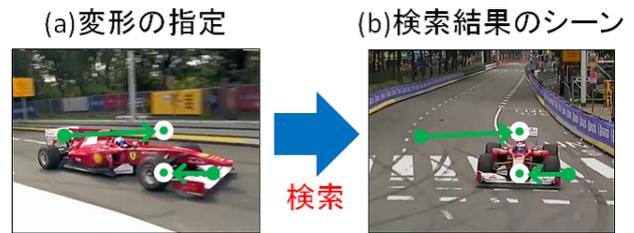


図 3. 提案するユーザインタフェース

カーの動画を一つ一つ見る必要がある。それでも発見できなかった場合、YouTube等の動画共有サイトで見つけ出さなければならない。その作業は非常に面倒である。

我々の提案するユーザインタフェースは目的のシーンを素早く、容易に検索が可能である。ユーザが入力するものは図3(a)のように、視聴したいF1カーが写っている静止画と2本の緑矢印だけである。このユーザインタフェースにおいてユーザは、「視聴したいF1カーが現在見ているF1カーの回転したバージョンである」という考えをもとに2本の緑矢印を入力する。この考えをもとにして、3次元右向きのF1カーを正面向きへ回転させることを考えた場合、正面向きのF1カーを2次元の静止画上で見ると、図2(b)のようにF1カーの前方は下部、後方は上部に配置される。このようにF1カーを回転させたと想定して2本の緑矢印で変形を指定する。ここでは、ユーザは図3(a)のようにF1カーの前方を下方向へ、後方を上方向へと矢印を入力する。我々のシステムはその2本の緑矢印で指定した変形に類似した目的のシーン(図3(b))を検索する。

4 アルゴリズムの概要

動画や静止画検索の技術において、今視聴しているシーン(図2(a))を入力として、目的のシーン(図2(b))を直接探し出すことは困難である。例えば、SIFT特徴量[11]を動画から抽出し、今視聴しているシーンと目的のシーンのマッチングの計算を行う。マッチングにF1カーを使用した場合の結果は図4で示す通りである。

右向きと左向きのF1カーといった外観が類似している左のケースでは正確なマッチングが多く抽出されている。一方で、右向きと正面向きのF1カーといった右のケースでは不正確なマッチングが抽出されているか、マッチングの数が少ない。ここで興味深いことに、図4(a)と(b)のフレームは図5で示す通り同じシーンに存在している。

図6(b)のような入力静止画(図6(a))と外観が類似したフレームを探し出し、そのフレームから動画を巻き戻すことで図6(c)のような目的のシーン(図6(d))を探し出すことができる。巻き戻しで図5(d)

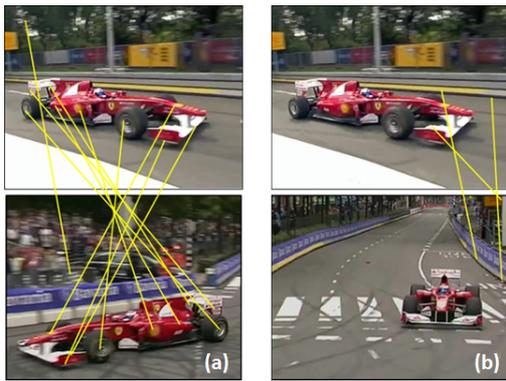


図 4. マッチングの例

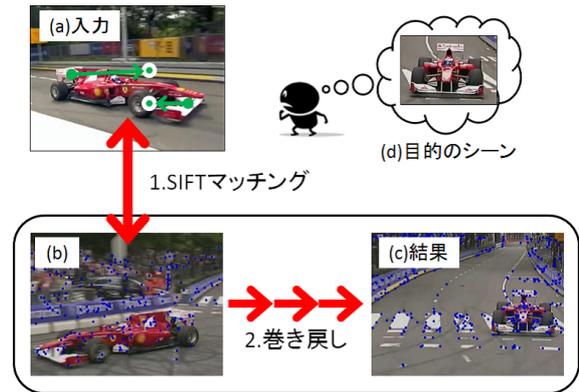


図 6. アルゴリズムの概要

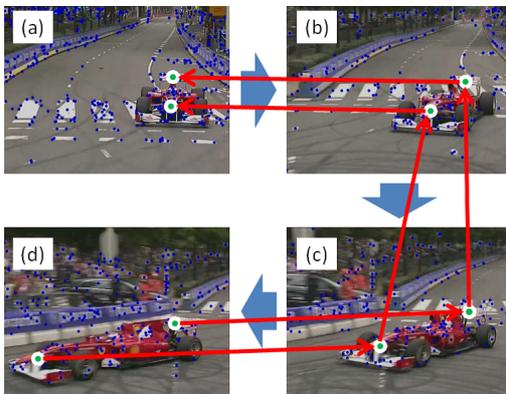


図 5. F1 カーが正面から左向きになる動画

と (a) を関連付ける方法として、図 5 の緑色の点のような物体上に配置された点を追跡する。

検索アルゴリズムは 2 ステップある。

1. フレーム検索ステップ

SIFT 特徴量を使用したマッチングにより、入力静止画内の物体と類似したフレームを検索する。

2. フレーム追跡ステップ

類似フレームから早送り・巻き戻しを行い、ユーザの入力矢印で指定された変形に類似したフレームを出力する。

2 ステップについて 4.1 と 4.2 で記述する。

4.1 フレーム検索

入力静止画と SIFT 特徴量を使用したマッチングを行った結果、マッチング数が最多のフレームを類似したフレームとして選択する。ただし、SIFT 特徴量をそのままマッチングに使用する方法では計算時間がかかる。計算時間を減らすために、Visual Words[15]を使用したマッチングを行う。この計算時間の比較は 5.1 で記述する。

Visual Words を使用したマッチングを行うために以下の前準備を行う。

1. 検索対象の動画の全フレームから SIFT 特徴量を抽出する。
2. 抽出した特徴量をランダムに選択し、クラスタリングを行う。クラスタリングには K-means 法を使用する。
3. それぞれのクラスタの中心のデータ (Visual Words) を辞書と定義する。
4. 辞書を使用し、残りの SIFT 特徴量についてどのクラスタに分類されるかを特定する。

この前準備で求めた動画の各フレームのクラスタの特定が行われた SIFT 特徴量は、ユーザインタフェースのシステム起動時に読み込まれる。ユーザインタフェースで静止画が入力された時、入力静止画と動画の各フレームとのマッチングは以下の流れで行う。

1. 入力静止画から SIFT 特徴量を抽出する。
2. 抽出した特徴量を辞書を使用し、クラスタの特定を行う。
3. 動画内フレームから抽出された各特徴量とクラスタの番号が一致する場合にマッチングとする。

マッチングは多く抽出されるが、その中には外観と比較すると不正確に見えるものが存在する。そのようなノイズを減らすために、RANSAC[6]を使用する。

Visual Words を使用し、入力静止画と外観が類似したフレームをマッチングを行った結果の例が図 7 である。図 7 下のフレームを含む動画において、マッチング数が最多であるので、このフレームをその動画での類似したフレームとして選択する。ただし、Visual Words を使用したことにより SIFT 特徴量をそのまま使用した場合よりも精度が低下している。よって、人間の見た目には類似していないにもかかわらず、マッチング数が動画内で最多のフレームが存在することがある。この問題に対して、マッ



図 7. Visual Words を使用したマッチングの例

マッチングの数が多い順から複数の候補のフレームを類似したフレームとして使用する。

4.2 フレーム追跡

4.1 で選択した類似フレームから前後フレームを探索し、ユーザが矢印で指定した変形にマッチングしたフレームを出力する。前後フレームの探索には図5のようにフレーム上に配置された点を追跡する。追跡点を抽出する処理は予め行い、システム実行時に追跡点データを読み込むようにする。物体の追跡には particle video[13] を使用したアルゴリズムと、SIFT 特徴量のマッチングを使用したアルゴリズムの2種類の方法を行った。前者は精度は高いが計算時間が非常にかかるため、実験では SIFT 特徴量のマッチングを使用する。SIFT 特徴量を使用した物体の追跡は多く行われている [2, 14] が、簡略化した以下のような流れで行う。

1. 動画内の各フレームで SIFT 特徴量を抽出する。
2. $n-1$ と n フレーム, n と $n+1$ フレームの SIFT 特徴量のマッチングをそれぞれ計算する。
3. $n-1$ と $n+1$ フレームとのマッチングが存在する n フレームの特徴点を追跡点とする。

そのアルゴリズムを使用し、追跡は以下のように行う。

1. 4.1 のマッチングの結果を使用し、入力矢印の始点と対応した類似フレーム上の点 (対応点) を求める。
2. 対応点と近傍の追跡点を求める。
3. フレームを移動させ、入力矢印の終点と追跡点が最近似のフレームを求める。

システムは、各動画の最近似フレーム中、上位3フレームを検索結果として出力する。

表 1. 各マッチングの計算時間 (秒)

	SIFT 特徴量 [12]	Visual Words
特徴量抽出	1.23	1.23
クラスタ特定	-	1.84
マッチング	91.11	1.78
合計	92.34	4.85

5 実験

すべての実験は CPU Intel i7-3930k 3.20GHz, メモリ 16GB のパソコンで行った。用意した全ての動画の大きさは 640×360 である。

5.1 計算時間比較

4.1 で記述したフレーム検索ステップにて、SIFT 特徴量をそのまま使用してマッチングを行った場合 [12] と、Visual Words を使用したマッチングを行った場合の計算時間は表 1 である。ただし、動画の SIFT 特徴量は予め抽出しており、それぞれのフレーム数は 761 である。また、ノイズを減らすために RANSAC[6] を使用している。

表 1 の結果から、Visual Words を使用したマッチングは、SIFT 特徴量をそのまま使用した場合より、マッチングを高速に行うことが分かった。

5.2 動画検索

我々は YouTube 上にある F1 カー、馬、飛行機の3種類の物体の動画でシステムの実験を行った。3種類の物体の動画をダウンロードした後、1シーンごとに動画を分割した。各物体ごとに検索対象は 20 本の動画である。それらの動画の全フレームから SIFT 特徴量を抽出し、Visual Words マッチングで使用する辞書とそれぞれの Visual Words データ、SIFT 追跡データを作成した。各物体の動画には 20 種類のシーンが存在する。

図 8 の 1 列目は静止画と矢印を入力したユーザインタフェースの画面、右の 3 列は各入力に対するシステムの検索結果である。実験に使用した動画のフレーム数と検索時間は表 2 に示す通りである。ここで、特徴量の抽出は 1 枚の入力静止画に対して 1 回だけ行えば良いので、2 回目以降の検索は抽出したデータを読み込むようにしているため、1 回目より高速になる。(クラスタ数 F1 カー 2500, 馬・飛行機 500)

図 8 の 1 行目の (a) は F1 カーの動画を検索した結果である。この行では、ユーザが正面向きに移動する F1 カーのシーンを検索しようと、右向きの F1 カーと 2 本の緑矢印を入力した。それを想定して、1 本目の矢印は F1 カーの前方を指定して下に向けて入力する。続けて、2 本目の矢印は F1 カーの後方



図 8. 検索結果 (a) 正面向きに移動する F1 カーのシーン (b) 左向きに移動する F1 カーのシーン (c) 右向きに移動する馬のシーン (d) 正面向きに移動する馬のシーン (e) 右向きに飛行する飛行機のシーン

表 2. 検索時間

	F1 カー	馬	飛行機
フレーム数	3941	7018	3741
特徴量抽出 (秒)	7.08	7.04	1.41
特定 (秒)	3.29	2.86	1.64
追跡 (秒)	0.97	2.29	0.36
合計 (秒)	11.34	12.19	3.41



図 9. 左上へ飛行する飛行機

を指定し上に向けて入力する。検索結果は入力静止画とユーザが指定した矢印に近似した上位3フレームである。上位3フレームの中に、ユーザが検索しようとした正面向きに移動する F1 カーのシーンが含まれている。以降の行でもユーザが視聴したいと思ったシーンが上位3フレームの中に含まれている。

また、図1も飛行機について検索した結果である。これが我々の技術の限界を示している。ユーザは図9のような左上向きの飛行機のシーンを発見しようと2本の緑矢印を描画した。検索された結果は全てで左上を向いており、ユーザの指定が満たされている。

しかし、飛行機はその前方と後方を軸として回転しているシーンばかりである。ユーザが図9のように飛行機が回転していないシーンを望む場合であっても、我々のシステムを使用して、飛行機が回転しているシーンを排除することは困難である。

6 まとめ

本論文では、静止画内の物体に2本の矢印を描画することでシーンの検索ができる新しいユーザインタフェースを提案した。実験において、F1カー、馬、飛行機といった3種類の物体において検索できることを確認した。

今後として、解決すべき課題がある。まず、実験においてユーザが検索しようとしたシーンとは違う意図しないシーンが存在したという問題がある。SIFT特徴量をそのまま使用し、Particle Videoを使用して検索を行った場合 [12] では意図しないシーンはあまり出てこなかった。このような精度低下の原因は、使用するアルゴリズムを変更したからである。しかしながら、フレーム検索では 17 倍、フレーム追跡では 1.5 倍以上高速になったため、より多くの動画を以前よりも時間をかけることなく検索対象にすることができる。より多くの動画を使用することで、シーンの種類が増え、ユーザの意図に近い動画が増えることにより、意図しないシーンが上位に来なくなるのではないかと考える。

次に、色情報を使用していないという点がある。SIFT 特徴量のマッチングや追跡において、色情報を使用していない。そのため、例えば、F1 カーの動画を大量に用意した場合、形は同じだが色が異なる F1 カーが検索結果として表示される可能性がある。この場合のシステムの対応について、色の違う車を許容して検索結果を出すのか、色の違いを許さず色情報を使用する方法を検討するのか等を考える必要がある。

3 点目に、検索する対象が剛体で特徴的な部分が存在しないとうまく働かない点がある。F1 カーや飛行機といった乗り物では色や模様が異なることがあるが、形状についてあまり変わらない。また、タイヤやロゴ、翼等といったように特徴的な部分が存在する。馬は足を動かし形状が変化するが、馬銜や鞍といった馬具を付けられてこれが特徴的な部分となっている。一方で、小動物や魚を検索対象にして実験を行ったが、うまく機能しなかった。その原因は、動画内で形状が変化し、特徴的な部分が存在しないからである。形状が変化すると、SIFT 特徴量を使用したマッチングがうまく機能しない。また、特徴的な部分がないと正確なマッチングも判定するのが困難である。そのような物体に対しても検索可能にするためには、使用する特徴量について検討する必要がある。

参考文献

- [1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski: Building rome in a day. In: ICCV 2009. (2009) 72-79
- [2] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato: Sift features tracking for video stabilization. In Proc. of International Conference on Image Analysis and Processing (2007) 825-830
- [3] R. Datta, D. Joshi, J. Li, J.Z. Wang: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40(2) (2008) 5:1-5:60
- [4] P. Dragicevic, G. Ramos, J. Bibliowicz, D. Nowrouzezahrai, R. Balakrishnan, K. Singh: Video browsing by direct manipulation. In: Proc. of CHI '08. (2008) 237-246
- [5] J. M. Frahm, M. Pollefeys, S. Lazebnik, C. Zach, D. Gallup, B. Clipp, R. Raguram, C. Wu, T. Johnson: Fast robust large-scale mapping from video and internet photo collections. ISPRS Journal of Photogrammetry and Remote Sensing 65(6) (2010) 538-549
- [6] M. A. Fischler, R. C. Bolles: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: Comm. of the ACM 24(6) (1981) 381-395.
- [7] A. Girgensohn, D. Kimber, J. Vaughan, T. Yang, F. Shipman, T. Turner, E. Rieffel, L. Wilcox, F. Chen, T. Dunnigan: Dots: support for effective video surveillance. In: Proc. of ACM Multimedia. (2007) 423-432
- [8] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, S. M. Seitz: Video object annotation, navigation, and composition. In: Proc. UIST 2008. (2008) 3-12
- [9] T. Karrer, M. Weiss, E. Lee, J. Borchers: Dragon: a direct manipulation interface for frame-accurate in-scene video navigation. In: Proc. of CHI '08. (2008) 247-250
- [10] D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, T. Yang: Trailblazing: Video playback control by direct object manipulation. In: IEEE International Conference on Multimedia and Expo. (2007) 1015-1018
- [11] D. G. Lowe: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60 (2004) 91-110
- [12] M. Okabe, Y. Kawate, K. Anjyo, R. Onai: Video Retrieval based on User-Specified Appearance and Application to Animation Synthesis. In: Proc. of MMM (2013)
- [13] P. Sand, S. Teller: Particle video: Long-range motion estimation using point trajectories. In: Proc. of CVPR '06. (2006) 2195-2202
- [14] S. N. Sinha, J. M. Frahm, M. Pollefeys, Y. Genc: Gpu-based video feature tracking and matching. In: Workshop on Edge Computing Using New Commodity Architectures (2006)
- [15] J. Sivic, A. Zisserman: Video google: a text retrieval approach to object matching in videos. In: ICCV. (2003) 1470-1477
- [16] N. Snavely, S. M. Seitz, R. Szeliski: Photo tourism: exploring photo collections in 3d. In: ACM SIGGRAPH 2006 Papers. (2006) 835-846
- [17] J. Tompkin, K. Kim, J. Kautz, C. Theobalt: Videoscapes: Exploring sparse, unstructured video collections. In: ACM Transactions on Graphics (Proc. of SIGGRAPH). (2012)

料理画像をアニメーションすることによる魅力的な料理動画生成システム

崎山 翔平 岡部 誠 尾内 理紀夫 平野 廣美*

概要. 本研究は、1枚の料理画像をアニメーションすることで、より魅力的な料理動画を作成するためのシステムを提案する。泡や湯気、飛沫、具材の揺れなど、料理をより魅力的に見せる要素は多々ある。これらの要素は動きの中に多く現れるため静止画よりも動画の方が表現しやすいが、そうした要素をうまく捉えた動画を撮影することは専門家でも困難である。例えば湯気や泡を発生させるには、実際に火を点火するという方法が挙げられるが、こうした撮影を行うと料理の状態が撮影中に変化してしまうため、撮影のタイミングは極めてシビアであり、状態が変わってしまった料理は二度と撮影に用いることができない。そこで我々は、点火などといった料理の状態を変化させてしまう行為を行わず、料理を見栄えよくセッティングしただけの状態に撮影された静止画に、あとから湯気や泡などの動画を合成することでこれを解決する。こうした動画の合成を行うことのできる既存のソフトウェアは存在するが、非常にパラメータが多く、そのソフトウェアに対する専門的な知識が必要となうに、数秒の動画の作成にも膨大な作業時間が必要となる。そこで本提案では泡や具材の揺れのパラメータ設定を行うためのアルゴリズムを独自に考案することで、ユーザのわずかな入力を基に、短時間で魅力的な料理動画の作成を可能とするシステムを実現した。

1 はじめに

現在、料理店のWEBメニューや電子メニュー、ショッピングサイトの商品イメージには静止画が多く使用されているが、これらの料理静止画を動画に置き換えると、美味しさの表現の幅がさらに広がり、コンテンツ全体をよりリッチにすることができる。料理は「活き」がうまく表現されるとより魅力的に見えるとしており、新鮮な野菜を滴る滴、グツグツと食材を揺らす泡、パチパチと鉄板から弾ける飛沫など、料理の活きを表す要素は数多く存在するが、これらの多くは動きの中に現れるため、静止画よりも動画の方がより豊かな表現が可能である。しかし、料理の活きを思い通りに表現して動画を撮影することは大変難しい。

ここで、料理撮影の専門家が、撮影が困難であるとする「鍋料理」を例に説明する。鍋料理をより魅力的に撮影しようとした場合、鍋に火をかけてグツグツと煮立っている状態を撮影することが最も一般的であるが、実際に鍋に点火してしまうと、肉には火が通りきってしまい黒く変色し、白菜などの菜っ葉は萎れ元の形状が維持できず、多くの具材はやがてスープに沈んでいってしまう(図1-a)。状態が変化してしまった具材は二度と撮影には使用できないため、撮影タイミングは極めてシビアであり、専門家ですら魅力的な動画を撮影することは困難である。

一方で、図1-bのような静止画の撮影は可能である。この静止画は鍋に冷たい水を張り、具材を綺麗

に敷き詰めて、火をつけずに撮影したものである。火をつけていないため、肉には火が通りきることなく明るい色を保っており、春菊などの菜っ葉は萎れず新鮮さを保ち、全ての具材はスープに沈むことなく綺麗に並んでいる。撮影中に具材の状態が変化することがないため、ゆっくりと時間をかけて構図を決めて撮影できるうえ、何度も撮り直しが可能である。しかし、図1-bには泡や湯気、具材の揺れなどといった要素が全く存在していないため、料理の活きを感じられない。



図 1. (a) 実際に鍋に火を点火して撮影された動画の一部。(b) 点火せず冷水に具材を並べただけの状態に撮影された静止画。

ここで、図1-aのように具材の新鮮さを欠いてしまうことなく、図1-bに、図1-aにあったような泡や湯気などといった要素を加えることができれば、魅力的な料理動画の作成が可能である。

そこで我々は図1-bのように見栄えよく撮影された1枚の静止画に、あとから泡や湯気、具材の揺れなどを合成することで、より魅力的な料理動画を作成する手法を提案する。こうした動画作成はAdobe After Effects[1]などの既存ソフトウェアでも可能であるが、ソフトウェアに対する専門的な知識を必

Copyright is held by the author(s).

* Shohei Sakiyama, 電気通信大学, Makoto Okabe, 電気通信大学/JST CREST, Rikio Onai, 電気通信大学, Hiromi Hirano, 楽天株式会社/電気通信大学

要とするうえに、わずか数秒間の動画を作成するために数時間の作業時間を要するという問題がある。そこで本提案では、魅力的な動画作成において特に表現が困難とされる「泡」や「具材の揺れ」を中心に、静止画1枚とユーザの少ない入力を基に、短時間で魅力的な料理動画を生成するシステムを開発した。また、実際にシステムを使用して料理動画を作成し、自然で魅力的な動画が作成できることを確認した。

2 関連研究

静止画をアニメーションする既存研究は数多く存在する [6, 5]。これらは画像の中に入っていきような3次元的な視点の変更やキャラクタアニメーションに便利であるが、料理動画の作成には対応していない。Chuangらは静止画に確率的な動きを与え、水面の揺らぎ等を扱った [4]。水面の揺らぎは泡の表現に似ているが、我々が行う泡の配置では、泡領域同士の関係やユーザが入力した情報によってそれぞれの領域にパラメータの異なった泡を配置するため、この手法を使用することはできない。岡部らは静止面に流体の流れる向きをスケッチすることで流体アニメーションの生成を行った [10]。この手法は、流体動画の配置がユーザのスケッチ技術に依存してしまうため、今回扱うような料理静止画を入力したところで質の高いアニメーションを得られるという保証がない。

また、ビデオテクスチャ合成の研究も多く存在する [13, 12] が、これらで合成した動画の見た目や動きを編集するのは難しい。Bhatらは手書きスケッチに基づいて流体ビデオを編集した [3]。見た目の編集も可能だが画像アニメーションは扱っていない。

さらに、本研究では具材画像をその種類に応じて自動で歪ませることで、具材の揺れを表現する。画像を変形させる研究は以前から多く行われており [2]、五十嵐らはこれをインタラクティブに操作することを実現した [7]。その後もこの手法の拡張などが多く行われている [11, 8]。しかし、本研究のように料理画像に特化し、具材の種類に応じて自動で画像を歪ませ、具材の揺れを表現するといった手法は存在しない。

そこで本研究ではユーザの少ない入力を基に、具材の揺らしアニメーションの生成、各泡領域に対する泡の配置及びパラメータ設定等を行うためのアルゴリズムを独自に考案することで、短時間で料理動画を作成することができるシステムを目指す。

3 システム概要

本手法における動画作成の典型的な流れについて述べる。ユーザはまずマウスを使用して入力静止画(図2-a)に簡単なスケッチを行い、泡を配置する領

域(泡領域)と食材領域の指定を行う(図2-b)。同じスケッチの方法を用いて、図2-bで指定した食材領域を、具材ごとに分割する(図2-c)。次にユーザは図2-cで分割した具材それぞれについて、「変形パターン」と「形状」というもの指定する(図2-d)。変形パターンには「変形、非変形」の2つの選択肢が存在し、その具材が春菊のように柔らかく形が変わりやすいのであれば「変形」、大根のように固く形が変わりにくいのであれば「非変形」を選択する。形状には「平面、球形、棒状、葉類、麺類」という5つの選択肢が存在し、その具材に対してユーザが最も適切であると思うものを1つ選択する。あとは”合成”ボタンを押すだけで、ユーザが指定した領域や具材の情報を基に、システムが自動で泡の位置やサイズ、速さ、具材の揺らし具合などを決定し、動画を合成する(図2-e)。このとき、動画に付与する湯気や音の指定も行うことができる。

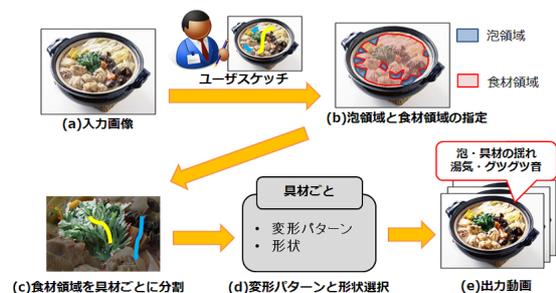


図 2. 動画作成手順.

4 ユーザインタフェース

ここでは作成したユーザインタフェースについて述べる。

4.1 泡領域と食材(具材)領域の指定

ユーザは簡単なスケッチで入力静止画に対し、泡領域と食材領域を指定を行う(図2-a,b)。この指定には、スケッチベースの領域分割器を用いる [14]。泡領域を指定する場合、ユーザは左クリックで泡領域を指定し、右クリックで鍋の縁や背景、食材部分などといった泡領域以外の領域を指定する(図3-a)。この指定は数少ないストロークで実現可能であるため、この操作は数秒で完了することができる。同様の指定を食材領域についても行う(図3-b)。

また、鍋料理のような煮込み料理の場合は、泡の発生と共に「具材が揺れ」が生じていた方がより自然である。そういった料理の場合、図3-bで指定した食材領域を、さらに具材ごとに細かく分割する(図3-c)。この操作も1つの具材に対しておおよそ2ストロークのみで指定を行うことができるため、全ての具材に対しこの指定を行っても、数秒から数分といった短時間で完了することができる。

具材の揺れが存在しない料理（例えば、鉄板の上でジュージューと焼けるハンバーグには具材の揺れは存在しない）については図 3-c の操作は必要ない。

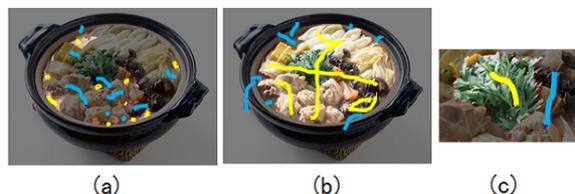


図 3. (a) 泡領域の指定. 黄色のストロークが泡領域, 青色のストロークが泡領域以外. (b) 食材領域の指定. (c) 具材の分割.

4.2 具材情報の入力

ユーザは分割した具材（図 2-c）それぞれに対して、「変形パターン」と「形状」の指定を行う。変形パターンとは、具材の形が変わりやすいか否かを示すものであり、「変形、非変形」という 2 つの選択肢が存在する。春菊のようにひらひらと柔らかく形が変わりやすいものは「変形」、大根のように固く形が変わりにくいものは「非変形」をユーザが選択する。形状については、「平面、球形、棒状、葉類、麺類」という 5 つの選択肢が存在し（図 4）、この中から、その具材に対して最も適切であると思うものを 1 つ選択する。これらの情報は、具材の揺らしアニメーションを生成する際に使用する。



図 4. 具材に対する「形状」の分類例.

4.3 湯気や音の選択

作成する動画に湯気や音を加えることができる。この指定は”合成”ボタンを押す前に行う。湯気を加える場合は、いくつか用意された湯気の素材動画の中から好きなものを選択し、インターフェースに表示された料理画像をクリックすることで、湯気を貼りつける位置を決定する。音を追加する場合は、用意されたいくつかの音素材の中から好きなものを選択するだけでよい。

4.4 泡素材動画の選択

料理動画の合成に使用する泡素材動画を変更することができる。用意した泡素材動画には、水やソース、醤油など複数の液体の泡が存在する。例えば水

とソースでは、粘性について大きな差があるため、泡の膨らみ方に違いがみられる。ユーザは自分の作成したい動画の感触により合った泡素材動画を選択することができる。

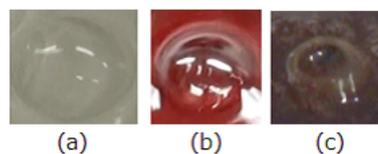


図 5. 泡素材動画. (a) 水 (b) 醤油 (c) ソース.

4.5 火元位置の指定

ユーザは火元の位置を指定することができる。火元とは画像中で最も強く加熱されている点であり、例えば鍋料理において、鍋の中心部を火元として設定したとすると、鍋の中心部に最も熱が加わっているかのように鍋の中心部には激しい泡が発生し、鍋の縁の方では中心部と比較して穏やかな泡が発生する。

この火元の位置の指定は、料理動画視聴者の視線の誘導につながると考えている。一般に、人は動きの大きな部分に着目する。本システムでは火元がその部分に該当するため、動画視聴者に特に注目してほしい部分を火元に設定することで、視聴者の視線をそこへ誘導することができる。

4.6 動画の生成

ここまでの情報を入力すれば、あとは”合成”ボタンを押すだけで料理動画が生成される。ユーザは泡の位置や粘性、具材の揺らし方など、細かなパラメータ設定を一切行う必要がない。計算は数十秒で終了し、料理動画が出力される。ユーザは出来上がった料理動画を見て、泡の状態や具材の揺れ方などに関して気に入らない点があれば変形パターンや形状といった情報を変更して、料理動画を修正することができる。動画作成に必要な情報の入力のほとんど（領域分割以外）はワンクリックで行うことができるため、動画作成はすべて合わせてわずか数分で完了することができる。

5 アルゴリズム

ここでは、泡の配置や具材の揺らし方の決定、泡のレンダリング、音や湯気の追加で用いたアルゴリズムについて述べる。

5.1 泡の配置アルゴリズム

用意した泡データベースには、水、醤油、ソースなど複数の液体それぞれについて、大きさや速度の異なるものが多数含まれている（図 5）。本システムではこのデータベースから適切な泡を選び出し、ユー

ザが図 2-b で指定した泡領域に対して配置する。この配置を行うために独自のアルゴリズムを考案した (表 1)。

表 1. 泡の配置アルゴリズム.

	対象の泡領域		配置する泡	
	位置	広さ	速度	サイズ
手順 1 基本配置	火元付近	狭	極遅	小
		広	遅	小
	火元遠く	狭	遅	極小
		広	速	大
手順 2 重み付け	食材領域との境目及び 食材領域間の泡の速度をより遅く			

この泡配置アルゴリズムは我々が実際に鍋料理や鉄板料理などといった泡の発生しやすい料理の映像を複数観察し、手で料理動画の試作を何度も試みた結果定義したものである。手順 1 では各泡領域に配置する泡の速度、サイズ、座標を決定し、手順 2 では手順 1 で配置した泡に対して重み付けを行う。

我々が調査したところ、発生する泡の特徴の違いには、その泡の領域の火元から距離と、その泡の領域の広さが大きく関係している。基本的に火元に近い部分では、発生から消失までの速度が遅く、縦横のサイズが小さい泡が発生しやすい。火元に近い部分の中でも特に、他の領域と比較して領域の広さが狭い場所では速度がより遅い泡が発生する傾向が見られる。一方、火元から離れた部分では領域の広さによって発生する泡の特徴が大きく異なる。狭い領域では速度が遅く、且つ火元付近で発生する泡などと比較してよりサイズの小さい泡が発生するのに対し、比較的広い領域では速度が速くサイズの大きな泡が発生する。手順 1 はこれらの特徴を考慮している。

手順 1 を本システムにおいて実現するために、各泡領域のピクセル数と、各泡領域の重心から火元までの距離を使用して、各泡領域に配置する泡のサイズ、速度のパラメータを決定する。泡領域 i のピクセル数を $a(i)$ 、重心から火元までの距離を $d(i)$ としたとき、その領域に配置する泡のサイズ $S(i)$ 、速度 $V(i)$ はそれぞれ次のように定義される。

$$S(i) \propto a(i) \cdot d(i) \quad (1)$$

$$V(i) \propto \frac{a(i)}{d(i)} \quad (2)$$

ここで、泡領域 i に対して、サイズ $S(i)$ の泡をグリッド状に並べる。これによりそれぞれの泡の座標が求まる。

さらに、食材との境目に存在する泡や複数の食材領域の間に挟まれて存在する泡は、その周囲の泡と比較して速度が遅くなる傾向が見られるため、手順

2 では、そのような泡に対して手順 1 で決定した速度を 0.7 倍し、これを重み付けとする。

5.2 具材の揺らしアニメーション生成

鍋のような煮込み料理では、グツグツと発生する泡と共に具材が揺れる。本システムではこの揺れのアニメーションを、ユーザが図 2-d で指定した「変形パターン」と「形状」に応じて自動生成する。

揺らし方については 2 パターン用意した。1. 具材画像を変形して歪ませることで揺れを表現するパターンと、2. 具材画像の座標を移動させることで揺れを表現するパターンである。図 2-d において、ユーザが「変形」を選択したい具材についてはパターン 1 の揺らし方、「非変形」を選択した具材についてはパターン 2 の揺らし方を適応する。

パターン 1: 画像変形による揺れの表現

春菊や青葉のような葉類が揺れるとき、具材全体が一様に揺れるのではなく、固い茎のある根本はあまり動かずに、柔らかい葉先だけが揺れていた方がより自然に見える。本提案では、リジット変形 [11] を用いて、春菊の根本のように形の変わりにくい部分を固定したうえで、葉先のように形の変わりやすい箇所を歪ませるように変形することで、具材の揺れを表現する。ここで、リジット変形を行う際、固定する点を固定制御点、歪ませる点を振動制御点という。

各具材に対する固定制御点と振動制御点の決定には、ユーザが図 2-d にて指定した「形状」の情報を使用する。我々は、用意した 5 種類の形状に対して、固定制御点と振動制御点をそれぞれどのように設定するかについて定めた (図 6)。



図 6. それぞれの形状に対する固定制御点 (中抜き円: 赤) と振動制御点 (塗りつぶし円: 黄色) の位置。

ユーザが入力した「形状」情報のみを使用して、システムが自動で図 6 のように制御点を設定するには、具材画像の傾きを考慮しなければならない。例えば図 6 の「平面」の例にある牛肉は、横軸に対して水平ではなく、左下から右上に向かって斜めに横たわっているため、この傾きに応じて制御点の座標を決定しなければ、変形が不自然なものになってしまう。そこで、本手法では主成分分析 [9] によってこの傾きを推定したうえで、制御点の座標を決定する (図 7)。決定された振動制御点は上下左右ランダ

ムに動かすことで具材が揺れているようなアニメーションを生成する。

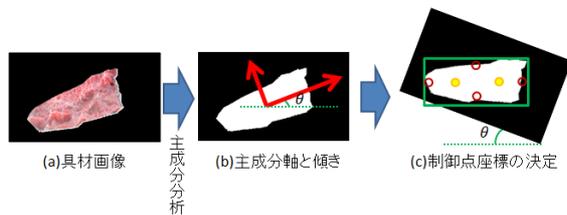


図 7. 主成分分析を利用した具材の傾き推定による制御点座標の決定。

パターン 2: 座標移動による揺れの表現

鶏肉やカットされた大根のような固めの食材が揺れるときは、パターン 1 のように具材の一部が揺れるよりも、具材全体が一様に揺れた方がより自然に見える。そこで、そのような具材に関しては具材画像全体を上下ランダムに座標移動し揺れを表現する。

5.3 レンダリング

泡素材動画と入力静止画の泡領域の色が異なっても泡が自然に見えるように、泡素材動画の泡の動きだけを取り出して泡領域に合成する。これは図 8 のように HSL 色空間の輝度 (Lightness) チャンネルの動きのみを用いることで実現可能である。ただしこのとき、入力静止画の泡領域に存在したエッジが目立ってしまわないよう、泡領域に対してブラーをかけたうえで泡を合成する。

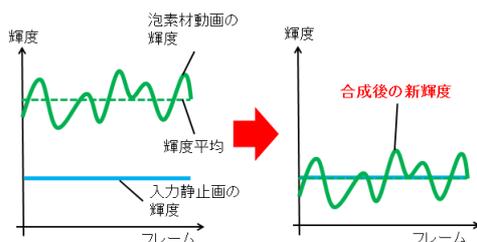


図 8. 輝度の揺らぎを付与するアルゴリズム。

5.4 湯気と音の合成

湯気は泡や具材の揺れと違い、細かいパラメータの設定をしなくても、湯気の素材動画を配置するだけで、動画に馴染んだ適切な湯気に見える場合がほとんどである。そのため、本システムではユーザが指定した位置に湯気の素材動画を貼りつけるだけで湯気の合成を完了する。音に関しても、料理の音はジュージュやパチパチ、グツグツといった白色性の高いものがほとんどであるため、音と映像の同期を考える必要はなく、ユーザが選択した音を動画に付与するだけで合成を完了する。

6 結果と考察

本システムを用いて 4 つの料理動画を作成した (図 9)。料理動画 a, b に使用した泡素材動画は図 5-a, 料理動画 c, d に使用した泡素材動画は図 5-c である。



図 9. 実験に使用した静止画と動画作成に要した時間。

作成した料理動画 a-d について、動画全体や具材の揺れ方などの自然さを確認するためにユーザテストを行った。ユーザテストによって確認したのは Q1-Q4 の 4 項目である。

- Q1 料理動画は静止画よりも美味しそうに見えたか。
- Q2 料理動画は全体的に見て自然であったか。
- Q3 泡は自然であったか。
- Q4 具材の揺れは自然であったか。

4 つの料理動画 (図 9 a-d) に対し、Q1-Q4 それぞれについて、学生 7 人に 5 段階評価を行ってもらった (5:最高評価, 1:最低評価)。

ユーザテストの結果を図 10 に示した。なお、料理動画 d に関しては映像中に具材の揺れが含まれていないため、Q4 への回答は存在しない。

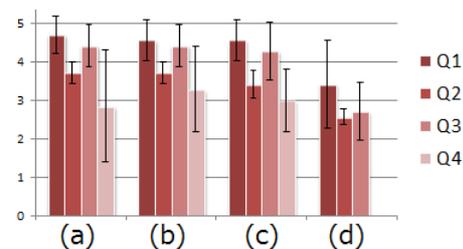


図 10. ユーザテストの結果。色棒は平均点、誤差棒は標準偏差。

ユーザテストの結果、システムで作成した料理動画 a-d すべてにおいて、作成に使用した静止画よりも美味しそうに見えるとの結果が得られた (図 10-Q1)。料理動画 d 以外は標準偏差も小さく、料理静止画をシステムによってより美味しそうな動画にすることができたことには、被験者のほぼ全員が同意したといえる。

動画全体の自然さ、泡の自然さ、具材の揺れの自然さなどを総合的に見て、最も評価が高かったのは料理動画 b であった。料理動画 b に使用した泡素材動画は図 5-a であるが、元画像の赤い色が合成した

泡にうまく反映されており、違和感をほとんど感じさせなかった。具材の揺れに関しても、「棒状」と指定したネギなどが非常に自然に揺れており、他の結果と比較して評価が高かった。一方、最も低い評価となったのは料理動画 d であった。料理動画 d は泡の配置も動画全体としても多少不自然であるという評価を受けた (図 10 d-Q2, Q3)。料理動画 d に対する被験者のコメントには「泡がぼやけて見える」という意見が多かった。これはレンダリング時、入力静止画の泡領域にブラーをかけてから泡の合成を行っていることが大きく影響したと考えられる。入力静止画の泡領域の色の差が激しいときや、指定した泡領域が大きい場合はこうした結果になりやすい。

料理動画 a, b, c のスコアを見ると、どれも同じような形をしている。泡の自然さ (Q3) のスコアの高さが、具材の揺れ (Q4) のスコアの低さに引っ張られた結果、動画全体としての自然さ (Q2) のスコアが下がっているという構図が見て取れるため、具材の揺れが今後改善されれば、動画全体としての自然さも向上することが期待される。

7 まとめと今後の課題

本研究では、ユーザの少ない入力を基に短時間でより魅力的な料理動画を作成するシステムを提案し、ユーザテストによって、その有用性を示した。ここでユーザテストに使用した静止画 (図 9 b, c) に注目したい。これらの静止画にはあらかじめ「料理画像に湯気を合成する」という加工が施されていた。料理画像に湯気を合成するのは、料理をより美味しく見せる方法としてメジャーなものであり、現在多くの料理店で使用されている静止画はこのように湯気が合成されているが、そういった加工を施すよりも (あるいは施した上で) 本システムによってその静止画を動画にした方が、料理をさらに美味しく見せることができる可能性が高いことをユーザテストの結果は示している (図 10 c, d-Q1)。つまり、現在の料理店のメニューなどが本システムにより動画に置き換えれば、料理の美味しさの表現はその幅を広げることにつながると考えられる。

今後の課題としては、具材の揺らしアルゴリズムの改善などが挙げられる。現状として、具材の揺れにおける若干の不自然さが動画全体としての自然さを損なっているため、今後具材の揺れが改善されることで、いまと同じ労力で、より魅力的な動画の作成が可能になると考えている。さらに、今後はユーザインタフェースにも改良を加え、システムによって自動生成された動画をユーザがより自分の感覚にあった動画を直感的に微調整できるようにしたいと考えている。現在レンダリングに数十秒の時間を要しているが理論的には 2, 3 秒程度で可能であるため、ユーザのスケッチを瞬時に結果に反映するなど、よりインタラクティブなシステムにしていきたい。

参考文献

- [1] Adobe After Effects. <http://www.adobe.com/jp/products/aftereffects.html>.
- [2] T. Beier and S. Neely. Feature-based image metamorphosis. In *ACM SIGGRAPH Computer Graphics*, Vol. 26, pp. 35–42. ACM, 1992.
- [3] K. S. Bhat, S. M. Seitz, J. K. Hodgins, and P. K. Khosla. Flow-based video synthesis and editing. In *ACM Transactions on Graphics (TOG)*, Vol. 23, pp. 360–363. ACM, 2004.
- [4] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. H. Salesin, and R. Szeliski. Animating pictures with stochastic motion textures. In *ACM Transactions on Graphics (TOG)*, Vol. 24, pp. 853–860. ACM, 2005.
- [5] A. Hornung, E. Dekkers, and L. Kobbelt. Character animation from 2D pictures and 3D motion data. *ACM Transactions on Graphics (TOG)*, 26(1):1, 2007.
- [6] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 225–232. ACM Press/Addison-Wesley Publishing Co., 1997.
- [7] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *ACM Transactions on Graphics (TOG)*, Vol. 24, pp. 1134–1141. ACM, 2005.
- [8] A. Jacobson, I. Baran, J. Popovic, and O. Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78, 2011.
- [9] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [10] M. Okabe, K. Anjyo, T. Igarashi, and H.-P. Seidel. Animating pictures of fluid using video examples. In *Computer Graphics Forum*, Vol. 28, pp. 677–686. Wiley Online Library, 2009.
- [11] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM Transactions on Graphics (TOG)*, Vol. 25, pp. 533–540. ACM, 2006.
- [12] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 489–498. ACM Press/Addison-Wesley Publishing Co., 2000.
- [13] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.
- [14] 清野達也, 林貴宏, 尾内理紀夫, 三條正裕, 森正弥. 改良領域拡張法による高速画像切抜き手法の提案と評価. *情報処理学会論文誌*, 12(50):3233–3249, 2009.

SyncBlink: 同期刺激によるナビゲーション方式の提案と評価

志摩 駿介 綾塚 祐二 寺田 努 塚本 昌彦*

概要. 近年, スマートフォンのアプリケーションや HMD (Head Mounted Display) などを用いた様々なナビゲーションシステムが開発, 使用されている. ユーザを誘導する情報を提示するシステムにおいて課題となるのは, 提示情報と現実世界とのマッチングである. 例えば, 画面上の地図に進むべきルートや曲がるべき交差点が明示されていても, それらが目の前のどの道, どの交差点に対応するかわからなければユーザは迷うことになる. そこで本研究では, パーソナルデバイスと環境側が連携し, 同期した刺激をユーザに提示することでナビゲーションを行う方式 SyncBlink を提案する. 環境側の各案内表示とユーザ側のデバイスが同期して刺激を発することにより, どの案内表示が目的のものであるかを容易に識別できる. また, 複数のユーザに同じ, または異なった案内を同時に提示することも容易であり, 人の多い環境でも利用できる. 本論文では, ユーザへの刺激として視覚, 聴覚, 触覚刺激を提示するプロトタイプデバイスをそれぞれ実装した. 評価結果から, 提案方式がナビゲーション手法として有効であることを確認した.

1 はじめに

近年, カーナビゲーションやスマートフォン上の歩行者向けナビゲーションなど, 様々な場面でナビゲーションシステムが使用されている. このような道案内を提示するシステムにおいて大きな問題となるのは, 案内情報とユーザの目の前の現実世界とのマッチングである. 例えば, 画面上の地図に進むべきルートや曲がるべき交差点が明示されていても, それらが目の前のどの道, どの交差点に対応するかに自信がもてなければユーザは迷うことになる. また, GPS を用いたユーザの現在位置の確認は, 実際の位置とずれた位置を示すことがあったり, 電波の届かない地下では利用できない.

一方, 現実世界に設置されている案内板などの環境側のナビゲーション設備は, 適切に設置されていれば対応関係のわかりやすい案内となりうる. しかし, この案内は一般的に様々な目的をもつ不特定多数の人々に対して複数の案内情報が近接して提示されるため, あるユーザのための案内がどれであるかを見つけ出すのが難しい. 一種類の案内しかない場合でも, それに気が付くのが困難な場合もある. これらは, 確実にユーザ自身のために提示されるパーソナルデバイス上での案内とは対照的である.

そこで本研究では, パーソナルデバイスと環境側が連携し, 同期した刺激をユーザに提示することにより, これらの欠点を解消したナビゲーション方式 SyncBlink を提案する. この方式はユーザに最低限の刺激を提示できればよく, またパーソナル

デバイス側は視覚的な刺激だけでなく聴覚や触覚の刺激も活用できるため, シンプルで応用範囲が広い.

以降, 2 章で関連研究について述べ, 3 章ではシステム設計について述べる. 4 章で評価実験を行い, 最後に 5 章で本論文をまとめる.

2 関連研究

環境側と同期したパーソナルデバイスを用いたナビゲーションシステムはこれまでも研究されている. 澤野らによる AR (Augmented Reality) を用いたカーナビゲーションシステム [1] では, 車載カメラからの映像を用いて道路端を抽出し, 3 次元 CG を生成して実写画像とマッチングさせている. 高田らによるウェアラブル AR システム [2] は, GPS とジャイロセンサによりユーザの位置と向きを認識し, ユーザが向いている方向にあるレストランの名前やメニューなどの情報を提示する. また, AR を用いたカーナビゲーションは実用化もされている (Pioneer サイバーナビ [3], MapFan eye[4] 等) が, 奥行き方向も含めた実景と提示情報のリアルタイムの表示位置合わせを高精度に行うのは (特に実景をそのまま見せる光学シースルー型の場合) 難しい. 仮に完璧な位置合わせと表示ができたとしても, 人間の奥行き方向の知覚の精度が限られるので, 必ずしもそれが見やすい, 判りやすい表示になるとは限らない. また, 視界上の広い範囲を占める情報を提示するのは安全性の面での懸念が生じる.

デバイスがユーザに刺激を与えることでユーザを案内する方式も研究されている. 塚田らによる Active Belt[5] は, 8 方向に振動モータをつけたベルトを用いて, 振動させる振動モータの位置で目的地の方向を提示し, 振動の強さで目的地との距離を提示する. この他, 水野らによる触覚ナビゲーション

Copyright is held by the author(s).

* Shunsuke Shima and Masahiko Tsukamoto, 神戸大学大学院工学研究科, Yuji Ayatsuka, 株式会社電通国際情報サービス, Tsutomu Terada, 神戸大学大学院工学研究科/科学技術振興機構さきがけ

ン装置 [6] では、両耳裏付近に装着した扁平型偏芯振動モータにより触覚にファントムセンセーションを生起し、位置情報の伝達を試みている。

環境側とパーソナルデバイスを連携させた研究では、松原らによる視覚障害者向け対話型情報提供システム [7] がある。このシステムでは点字ブロックに埋め込まれた位置情報を、ユーザが先端にアンテナのついた杖で読み込み、携帯端末装置に伝送することにより現在地を知ることができる。

3 SyncBlink

本章では、提案方式 SyncBlink の基本的なアイデアを説明してシステムの要求仕様を明確にした後、デバイスのプロトタイプを紹介する。

3.1 基本アイデア

1章で述べたように、環境側の案内情報は、ユーザが案内情報の存在を知覚している場合はどれが必要な情報であるかを判別できる必要がある。また、そもそも案内情報がどこにあるのかをユーザが知覚していない場合も案内情報の存在がユーザに自然に伝わればわかりやすい案内となる。したがって例えば案内板の必要な部分を光らせ、点滅させるなどの方法を用いて、ユーザが必要とする案内だけを目立たせればこれらの目的は達成される。一方、この方法では複数ユーザに対応できず、多くの人が様々なタイミングで利用する案内情報提示には向かない。そこで、「自分に向けての情報であること」を、確実にユーザに伝えるために、ユーザのパーソナルデバイスを活用する。

筆者らは以前「画面上に複数の写真がタイル上に表示されており、それらがランダムに更新される場合、偶然同じタイミングで更新された部分は関連があるように感じられる」という知見を得ている [8]。これを応用すると、パーソナルデバイス上で、そのユーザが必要とする案内板の点滅と同期した点滅を提示すれば、必要な情報をユーザに認識させられるといえる。また、点滅による視覚的な刺激だけでなく、同期して鳴る音や振動による触覚刺激も同じように利用できると思われる。この、「環境側の案内の必要な部分の点滅と、パーソナルデバイス上での同期した刺激」によるナビゲーション方式を SyncBlink と呼ぶ。この方式は、図1のように環境側が1つで同時に複数のユーザに同じまたは異なる案内を行う(一対多)ことも、環境側が複数で複数のユーザに同じまたは異なる案内を同時に行う(多対多)ことも可能である。

3.2 プロトタイプ的设计と実装

提案方式を用いて常時自然な案内を行うために、装着するデバイスは邪魔になりにくいものである必要がある。また、提案方式は環境側デバイスとユー



図 1. 案内板とデバイスの同期の例

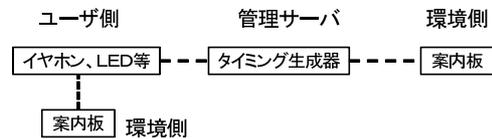


図 2. システム構成

ザデバイスが正確に同期する必要がある。これらの要件を考慮したシステム構成を図2に示す。この図のように、環境側が直接ユーザ側に同期信号を送る場合と、管理サーバが環境側とユーザ側両方に信号を送る場合が考えられる。また、図3に作成した、イヤホン、LED、振動モータをそれぞれ用いたデバイスのプロトタイプと、それらを装着した様子を示す。イヤホンを用いたデバイスでは環境側との同期のタイミングでピープ音が鳴る。LEDを用いたデバイスは眼鏡をかけヘアバンドで頭に装着し、環境側との同期のタイミングでLEDが点灯する。振動モータを用いたデバイスはリストバンドで手首に装着し、環境側との同期のタイミングで振動モータが振動する。イヤホンを用いたデバイスは arduino nano、LEDと振動モータのデバイスは arduino fio で制御し、XBee で無線通信を行っている。環境側が点滅の開始時に、無線通信で点滅の間隔(数値情報)を送信すると、デバイス側は受信した間隔の刺激を提示する。複数のデバイスに異なる情報を提示する場合は、提示したい表示の点滅開始時にそれぞれ点滅間隔を送信する。

4 評価実験

提案方式の有効性を評価した。まず、SyncBlink方式を用いて提示情報の識別を行えるかを確認するため、PC画面上での簡単な実験を行い、次に装着型デバイスを用いた場合の特性を評価する。最後に、実際に案内表示を環境内に設置し、ユーザを目的地



図 3. デバイスのプロトタイプ

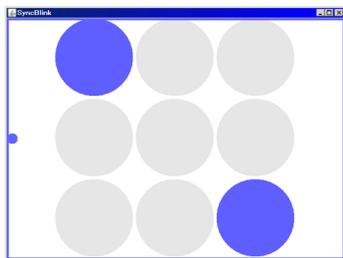


図 4. 実験用アプリケーション 1

に案内する実験を行う。

4.1 PCのみを用いた評価

まず、装着型デバイスは用いずに、視覚刺激と聴覚刺激を個別に、もしくは複合的に用いた場合に提示情報を識別できるかを確認するための実験を行った。実験では異なる周期で点滅する複数の円を PC 画面上に表示し(図 4)、ターゲットとなる円の点滅と同期して「ウインドウの枠を点滅させる(視覚刺激)」「ピープ音を鳴らす(聴覚刺激)」のいずれか、もしくは両方がある状態で、ユーザはターゲットを特定しクリックする。ウインドウの枠を点滅させるのは、注視点以外への刺激を意図している。円はそれぞれが異なる 1~4 秒程度の一定の周期で点滅し、2つ以上の円が同時に点滅することもある。ターゲットをクリックすると円の周囲が赤く、ターゲットで

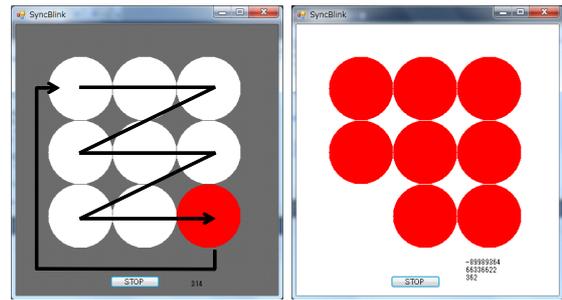


図 5. 実験用アプリケーション 2

ないものをクリックすると円の周囲がグレーになり、被験者は正否を認識できる。クリックすると、その正否に関わらず次のターゲットに進む。

このタスクを、a) 視覚的刺激のみ、b) 聴覚的刺激のみ、c) 視覚的刺激+聴覚的刺激の三種類を各 10 回(これを 1 試行とする)ずつ行わせた。開始前には、c) の条件で説明及び 10 回の練習を行った。被験者は 9 人(男性 6 人、女性 3 人)で、a~c) の順序はランダム(ただし、c) は常に最初か最後)に行った。被験者には、速さよりも正確性を優先するよう教示した。

表 1 が各被験者の各条件下での平均の回答時間、回答までの平均のターゲットの点滅回数、誤答の数である。被験者の番号は、誤答の少ない順にソートした。まず、全ての条件において誤答数が小さいことがわかる。被験者 9 は誤答が多いが、最大でも過半数には達していない。点滅数が 2 回以下になっている(すなわち、ターゲットを見定めたあとに再度点滅を待ち確認していない)試行では誤答が多い。9 名中 4 名は全ての試行を誤答なしで終了し、8 名は誤答なしで終えた試行がある。回答までの時間はターゲットの点滅周期の長さ依存するので、点滅回数で比較すると、全ての被験者において視覚的刺激のみの場合がもっとも点滅回数が多い。

これらの結果から、聴覚的刺激を用いた場合には劣るものの、視覚的刺激のみを用いても同期するターゲットの検出が可能であり、提案方式が有効に使えることが推察される。また、誤答の多かった被験者 8, 9 は回答までの平均のターゲットの点滅回数が少ないことから、誤答は確認不足から起こることが示唆されるので、ユーザに対し確認を促すようなインターフェースの設計をした方がよいことがわかる。

4.2 ユーザデバイスを用いた評価

前節における PC 上での評価結果をふまえ、3 章で述べた装着型デバイスを用いて評価実験を行った。前節の実験と同様に、点滅する複数の円を PC 画面上に表示する。規則的、またはランダムな順序で円を点滅させ、図 3 に示したデバイスのいずれかを装着して、ターゲットとなる円の提示のタイミングとデバイスの刺激提示の同期の識別を確認した。規則

表 1. PC のみを用いた評価の結果

	視覚的刺激のみ			聴覚的刺激のみ			視覚的刺激 + 聴覚的刺激		
	時間 (s)	点滅数	誤答数	時間 (s)	点滅数	誤答数	時間 (s)	点滅数	誤答数
被験者 1	6.52	2.9	0	5.56	2.2	0	5.98	2.9	0
被験者 2	7.38	2.6	0	4.76	2.5	0	4.10	2.0	0
被験者 3	8.45	3.3	0	6.93	3.3	0	5.87	2.9	0
被験者 4	10.8	3.8	0	7.80	3.3	0	5.72	2.6	0
被験者 5	8.73	3.4	1	6.94	2.8	1	8.20	3.2	0
被験者 6	5.85	2.1	0	4.23	2.1	0	4.32	1.7	2
被験者 7	12.0	5.4	3	7.44	3.4	0	4.59	1.8	0
被験者 8	5.00	2.1	0	4.24	1.4	2	5.08	1.8	2
被験者 9	6.11	2.3	3	2.20	1.1	2	2.46	1.3	4

表 2. ユーザデバイスを用いた評価の結果

		イヤホン			LED			振動		
		0.3(秒)	0.2(秒)	0.1(秒)	0.3(秒)	0.2(秒)	0.1(秒)	0.3(秒)	0.2(秒)	
間隔無	規則的		0.75	0.75	0.50	0.75	0.75	0.50	1.00	1.00
	ランダム	点灯	0.83	0.75	0.33	0.83	0.75	0.25	1.00	0.92
		消灯	0.92	0.75	0.58	0.83	0.75	0.50	1.00	0.58
間隔有	規則的		0.92	0.83	0.67	1.00	0.83	0.67	1.00	1.00
	ランダム	点灯	1.00	1.00	0.83	1.00	1.00	0.67	1.00	1.00
		消灯	1.00	1.00	1.00	1.00	1.00	0.92	1.00	1.00

的な順序で点滅する表示では、図 5 左の矢印の順序で点滅する。ランダムな順序で点滅する場合のみ、図 5 右のように提示方法を点灯ではなく消灯にした場合の実験も行った。これは、規則的な順序で点滅する場合は、次に点滅する円がわかっているため点灯と消灯で認識に差がないと判断したためである。規則的、ランダム共に、点滅と次の点滅の間に点滅時間と同じ長さの間隔がある場合とない場合（例えば点灯との同期なら、すべての円が消灯している時間がある場合とない場合）の 2 パターン行った。点灯時間は施行中一定で、0.3 秒、0.2 秒、0.1 秒とした。ただし、振動デバイスは 0.1 秒間隔では振動が起らなかったため 0.1 秒は省略した。点滅順序が規則的な場合では、同期場所をランダムに決定した後 3 周期点滅させる動作 3 回を 1 試行とした。点滅の順序がランダムな場合では、同期場所の点滅周期が長いとその場所が予測できてしまい、同期回数が多いほど予測できる可能性が増すので、規則的な場合より同期回数を 1 回減らし、2 回同期するまで点滅を行う動作 3 回を 1 試行とした。このように、前節の実験と異なり正解が分かたらすぐに回答させるのではなく一定期間の点滅の後回答させるようにした理由は、前節の実験結果から確認不足による誤回答が多く、また実際の環境でも複数回の点灯期間が存在することが一般的であると考えたためである。被験者は 4 人（男性 3 人、女性 1 人）で、図 5 左の矢印の順に左上の円から 1~9 の番号を割り当て、同期した円の番号を答えてもらった。また誤認識の傾向を調べるため、ランダムの場合には同期場所の点滅の前後に点滅した円の場所を記録した。

表 2 に実験の結果をまとめた。数値は被験者 4 人の平均正答率である。この結果から、0.3 秒と 0.2 秒の結果で比較すると、振動モータを用いたデバイスでの認識率が他の 2 つのデバイスでの認識率と比べやや高いが、各デバイス共にある程度高い正答率が出ているため、点滅間隔を長くすれば確実な認識が可能だと推察できる。また、点滅間隔が 0.1 秒の場合、振動以外のデバイスで点灯との同期よりも消灯との同期の方がわずかに認識率が高かった。このことから点灯との同期より消灯との同期の方が認識率が高いと推察できるが、点滅間隔を長くすれば点灯、消灯ともに確実な認識が可能である。また各デバイス共に点滅と点滅の間に間隔がある場合の方が間隔がない場合に比べ認識率が高かった。

この他、ランダムな順序の場合の平均認識率は、規則的な順序の場合に比べやや高かった。このとき、規則的に点滅させた場合の誤認識の 78.4% がターゲットの点滅の前に点滅した円との誤認識であった（ランダムの場合には 47.3%）。このことより、規則的な順序で点滅させた場合、前後の点滅と誤解した場合のリカバリが難しくなることが分かる。ランダムな場合は、前後の点滅が毎回異なるため、そのミスが起こりにくかったと考えられる。よって、点滅の順序を規則的にする場合は、誤認識をなくすため点滅の間隔を長くする必要がある。

4.3 実際の案内による評価

環境側の表示と 3 つのプロトタイプを用いて、実際にユーザを案内できるかの実験を行った。前節の評価より、点灯との同期と消灯との同期では、消灯



図 6. 案内表示



図 7. 案内実験の風景

との同期の方が認識率が高かったため、実験は消灯との同期で行った。また、ランダムな順序での点滅では、ユーザが表示を注視する必要があり、間隔の長い規則的な順序での点滅での認識は十分確実だと推察できるので、実験は規則的な順序での点滅で行った。点滅の間隔は、ユーザが容易に識別できかつ次の提示までの待ち時間が長いと感じないように、筆者の判断で0.7秒とした。実験は以下に示す3種類行ったが、環境側が1つの場合は図6左に示す案内板を用い、複数の場合は図6左の案内板と右のPC画面での表示とを併用した。また複数のユーザを同時に案内できるかを調べるため、各実験2人または3人1組で行い、同じまたは異なる案内を提示した。実験風景を図7に示す。この図では被験者は直進を提示されている。

円滑に案内できるかの評価

歩行中のユーザが立ち止ることなく円滑に案内された方向へ進めるかを調べるため、実験を行った。被験者に分岐を示す十字路から約16m手前にスタートラインを設け、十字路の1m手前の道端に直進または右左折を提示する案内板を設置し、歩行中の被験者が立ち止ることなく十字路を案内された方向に進めるかを調べた。被験者には始め進行方向とは反対を向いてもらい、システムを起動して最初の同期を感じてから、進行方向に進んでもらった。被験者は12人で各被験者それぞれ3つのデバイスで1回ずつ3回実験を行った。実験の結果、歩行速度には個人差があったものの、どのデバイスを用いても立ち止まった被験者はいなかった。

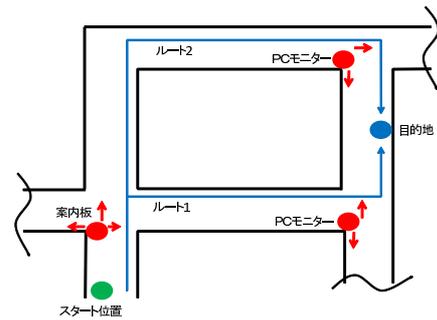


図 8. 目的地案内実験の簡易図

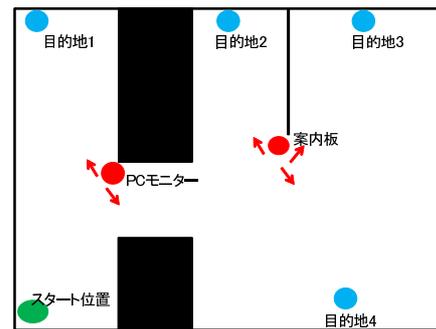


図 9. 複数表示実験の簡易図

目的地への案内の評価

実際にユーザを目的地へ案内できるかを調べるため、被験者に2つの分岐を進んでもらう実験を行った。図8のように環境側の表示は、案内板とPCモニターを用い、始めに案内板を用いて3方向の分岐を、次にモニターを用いて2方向の分岐を進んでもらった。スタート位置から目的地までには2通りのルートがあり、始めの分岐でどちらかのルートを提示し、2つのルートそれぞれに次にある分岐にモニターを設置した。案内板とPCモニターの間地点付近では、デバイスは両方からの信号を受け取り、意図した案内とは別の刺激を提示するが、案内板とモニターが同時に視界に入ることなく、どちらかの表示がはっきり見える距離まで近づくともう一方からの提示は届かなかった。被験者は10人である。

実験の結果、すべての被験者を意図した場所へ案内できた。中間地点付近での両方からの刺激の提示について違和感を覚えた被験者もいたが、ほとんどの被験者は表示の見えていない地点での刺激なので気にならなかったと答えた。表示がはっきり見えるところでは片方からの刺激の提示しかないことから、間違った方向への案内は起こらなかった。また、この違和感を感じる問題は、次に示す実験の手法を用いることで解決できる。

複数の案内表示が同時に見える場合の評価

複数の案内表示が同時にユーザの視界に入ると、ユーザを正しく案内できるかを調べるため、短

い間隔で2つの分岐点がある環境に案内板とPCモニターを設置して被験者を意図した場所に案内できるかの実験を行った。始めに右または左方向への分岐があり、左方向には新たな分岐はなく、右方向へ進んだ先にはさらに3方向への分岐を設けた(図9)。最初の2方向への分岐へ近づくと、次の3方向への分岐を示した案内版が見えるようにした。2つの案内表示が同時に見えても正しい案内を行えるようにするため、それぞれの案内板の要素数2と3の公倍数6を1周期とした点滅を行わせることで、すべての表示の組み合わせに対応する。例えば案内する場所が図9の目的地1の場合、ユーザ側のデバイスは始めの案内の左の表示で常に刺激を提示するが、目的地2, 3, 4の場合、始めの案内が右を表示し、次の案内がユーザの目的地を指す組合せの場合にのみ刺激を提示する。このようにすることで、刺激提示間隔は延びるが、ユーザが間違えた情報を受け取ることがなくなる。

この実験の結果、案内場所が図9の目的地2, 3, 4の場合には最初の分岐での情報提示に多少時間はかかったが全ての被験者意を意図した目的地に案内できた。一方、視界内の案内版数および各案内板における選択肢数が増加すればその組合せも膨大になるため、刺激提示の周期が実用的なレベルに保てなくなる可能性がある。そのような場合、ユーザに最も近い案内表示からの信号とのみデバイスを同期させ、かつ後方の案内表示からの信号を受け取らない機構や、同時に見える複数の案内表示は点滅の周期を早くするといった工夫が必要となる。

4.4 考察

以上の実験より、提案方式は持ち運べるデバイスを用いて確実に円滑な案内ができることが確認できた。また、複数のユーザに同じまたは異なる案内が可能であることも確認できた。目的地への案内の評価において、被験者に同期のわかりやすかったデバイスについて聞いたところ、イヤホンを用いた音との同期が一番わかりやすいと答えた被験者が最も多く6人だった。またそのうち5人が、振動についても音と同じくらいわかりやすかったと答えた。光が一番わかりやすかったと答えた被験者は1人で、それ以外の被験者は光が一番わかりにくかったと答えた。これは、案内板の表示が視覚刺激なのに対し、同じ視覚刺激を用いていることから、案内板の表示とデバイスの光提示両方を意識するのが難しいからだと推察でき、ユーザ側パーソナルデバイスに視覚刺激は不向きだと考えられる。一方、この結果から音声案内等に対して視覚刺激によりユーザをナビゲートするといったシステムが有効になる可能性がある。

このように、提案方式を用いて歩行中のユーザを円滑に目的地に案内できることを確認できたため、

案内方式を用いると長時間装着型デバイスに注意を向けることなく提示を識別できる可能性が高い。したがって、パーソナルモビリティや自動車などの運転中の使用も可能だと考えられる。

5 まとめ

本研究では、パーソナルデバイスと環境側が連携し、同期した刺激をユーザに提示することでナビゲーションを行う SyncBlink を提案した。実験結果より、環境側の各案内表示を独立したタイミングで点滅させ、ユーザが必要とする案内に同期した刺激を端末から受け取ることにより、適切に案内が行えることを確認した。ユーザ端末側の刺激としては視覚や聴覚、触覚を活用でき、歩行中だけでなく、パーソナルモビリティの運転中等、さまざまな状況に対応することが可能だと考えられる。同じまたは別々の案内を複数のユーザに同時に行うことも容易であり、人の多い環境でも活用できる。

今後は、さまざまな形態での実装を進める。その上で実環境での実験を行い、有効性を検証する予定である。

参考文献

- [1] 澤野弘明, 岡田 稔: 拡張現実感技術を用いた次世代カーナビのための道路端抽出法の検討, 情報処理学会研究報告(グラフィクスとCAD研究会), Vol. 2004, No. 121, pp. 103-108 (Nov. 2004).
- [2] 高田大輔, 小川剛史, 清川 清, 竹村治雄: 身体動作に基づき提示情報を切り替えるコンテキストウェアなウェアラブルARシステム, ヒューマンインタフェース学会論文誌, Vol. 12, No. 1, pp. 47-56 (Feb. 2010).
- [3] CyberNavi: <http://pioneer.jp/carrozzeria/cybernavi/>.
- [4] MapFan eye: <http://www.mapfan.com/iphone/ar>.
- [5] 塚田浩二, 安村通晃: Active Belt: 触覚情報を用いたベルト型ナビゲーション機構, 情報処理学会論文誌, Vol. 44, No. 11, pp. 2649-2658 (Nov. 2003).
- [6] 水野統太, 内田雅文, 井出英人: 振動刺激によるナビゲーション情報の認知評価, 電子情報通信学会技術研究報告(MEとバイオサイバネティクス研究会), Vol. 106, No. 592, pp. 41-44 (Mar. 2007).
- [7] 松原 広, 深澤紀子, 後藤浩一: 視覚障害者向け対話型情報提供システムの開発, 情報処理学会研究報告. ITS, [高度交通システム], Vol. 2001, No. 83, pp 75-82 (Sep. 2001).
- [8] 綾塚祐二: 描き足していくことを強調した手書きコミュニケーションツール, 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(WISS 2004), pp. 9-12 (Dec. 2004).

Traxion: 仮想力覚提示デバイス

暦本 純一*

概要. 仮想力覚を作り出す新しい触覚アクチュエータの構成について報告する. このデバイスは人間の振動刺激とその知覚の非線形な関係に基づき, 振動波形を制御することで押したり引いたりといった「力」を感じさせる. 既存研究と比較してアクチュエータのサイズや重量が非常にコンパクト (35.0 mm × 5.0 mm × 7.5 mm, 5.2 g) で, 力覚提示方向を迅速に変更することが可能である. 振動アクチュエータへの入力信号と, 実際にアクチュエータに発生している振動や加速度をレーザ測距装置で計測し, 仮想力覚の発生原因を解析した. 評価者実験の結果, すべての実験参加者が仮想力覚を感じることを確認した. 仮想力覚の強さを評価するために, 現実の錘と仮想力覚を比較する評価実験を行い, このアクチュエータが約 30 g の仮想力覚を作り出していることを推定した. 本研究は多様な場面でインタラクティブシステムに応用することが可能であると考えているが, 本論文では歩行者ナビゲーションと, DLP(Digital Light Processing) プロジェクター光による可視光通信で空中に力覚場を作り出す応用例についても報告する.

1 はじめに

触覚はわれわれが生活する上でとても重要な感覚である. 触覚として知覚することで, われわれは現実世界の物体の形状を把握し, 物体の操作を精密に, 正確に, そして素早く行うことができる. コンピュータなどの電子機器の操作においても, 触覚フィードバックを利用している研究事例は多い.

ここで, 触覚によるフィードバックを2つに分類する. ひとつめは力覚フィードバックと呼ぶもので, 実際に「力」を発生させるものである. 機構部品 (ロッドなど) により操作部に力を与えているもの [7, 9], ワイヤーにより力を与えるもの [5] などが提案されている. もうひとつは, 触覚ディスプレイあるいは振動フィードバックとよばれるもので, 力は発生させずに, フィードバックとして振動を利用するものである. たとえばタッチパネル上のボタンをタッチしたときに, クリック感を振動によって提示するものなどが提案されている [4, 11].

これらの二種類はそれぞれの目的・用途があり, また長所・短所がある. 力覚フィードバックは機構部品やワイヤーなどで固定物と接続する必要があり, 装置が大掛かりになってしまうので, モバイル用途などに適用することができない. 一方, 触覚ディスプレイでは本当の「力」を提示することができないので, 現実での力覚を再現するような用途には向かない. また, フィードバックとして用いる場合も, 利用者や操作部を特定の方向に導くために力の方向を利用することができない.

機構部品によらずに力覚を提示するいくつかの研究事例がある. 2ないし3個の偏心モーターによって慣性モーメントにより力覚を提示する装置を提案されている [14, 10, 3]. 雨宮らは刺激の認知の非

線形性にもとづき, 回転体とクランク機構によって非対称的な加速度パターンを生成し, 力覚が感じられることを示している [2, 16]. これらの研究により, 実際に物理的な力を発生させなくても, 知覚としての力覚の提示が可能になってきた. しかし, 従来提案されている機構はいずれも回転体やクランク機構などの機械部品からなっており, 全体のサイズや重量の面で改良の余地がある. たとえば雨宮らのデバイスは単方向の力覚提示に 70 mm × 200 mm × 48 mm の装置を要し, 重量も 250 g であった [2]. これらを組み合わせる2次元ないし3次元の力覚提示装置を構成するためには, さらに重量が増すことが予測される.

回転体やクランク機構による力覚提示のもう一つの問題は反応性である. 回転方向を急に変化させたり逆転することは難しいので, インタラクティブなシステムにおいて力覚の量を迅速に提示することができない. クランク機構は, 構造上, 回転方向を逆にしても非対称な加速度のパターンは逆転しないので, 一次元かつ単方向への力覚提示しか行えない. 回転体ではモーターの回転方向を変えることで力覚の提示方向の変化が可能であり, 小型モーターによってアクチュエータの小型化も可能ことが報告されている [10]. しかしモーターの回転を急激に停止・開始することは依然困難である.

本論文では, これらの問題を解決し, 従来よりも飛躍的に小型軽量化された仮想力覚提示デバイス, Traxion について報告する. Traxion の外形は 7.5 mm × 35.0 mm × 5.0 mm で重量は 5.2 g であり, 双方向一次元の仮想力覚を創り出すことができる. また提示方向を変化させるのに要する時間も数ミリ秒程度である.

2 Traxion の動作原理

Traxion はアルプス社のゲームパッド用振動アクチュエータ (Force Reactor [1]) の入力信号を制御す

Copyright is held by the author(s).

* 東京大学大学院情報学環・ソニーコンピュータサイエンス研究所

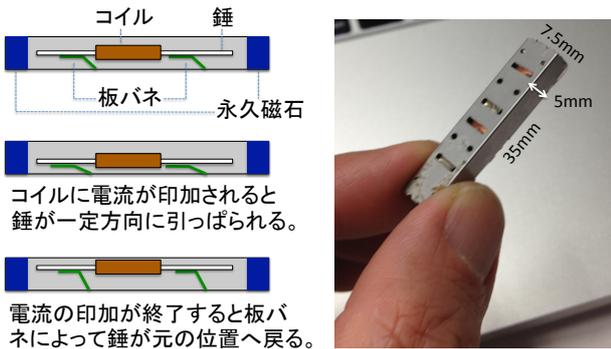


図 1. 振動子 (フォースリアクター) の構成

ることによって仮想力覚を作り出している。Force Reactor は、コイルが板バネで支えられた振動体が格納されている。ケースの両端には永久磁石が設置されている (図 1)。この構造でコイルに電流を加えると磁界が発生し、コイルが一定方向に移動し、加速度を生み出す。印加を停止すると板バネによってコイルが元の位置に戻る。この構造をゲームパッドに組み込むと、様々な振動パターンを作り出すことができる。

実際にアクチュエーターに発生する加速度をレーザー測距装置 (キーエンス LK5000) によって測定した結果を図 2 に示す。指で保持した場合と、スポンジ、シリコン、ゴムの台上にアクチュエータを置き計測した場合を比較した。図では指で保持した場合と比較的加速度パターンが似ているスポンジ台での測定結果を示している。加速度計測周期は $50 \mu s$ である。

図 2(1) は、コイルに直流電圧を印加したときのアクチュエーターに発生する加速度である、印加後コイルの移動にともない加速度が発生し、それが減衰していく様子がわかる。信号の印加/停止を周期的に繰り返すことで定常的な振動を作り出すことができる。図 2(2) は 125 Hz の正弦波を、(3) は同じく 125 Hz の矩形波 (デューティー比 5:5) を加えた場合の加速度である。コイルに発生する磁界による運動と板バネによる減衰効果が組み合わさっており、周期運動ではあるものの、入力信号がそのまま単純には加速度変化に反映しているわけではないことが観測される。この (2) と (3) の場合、アクチュエーターを指で保持しても、振動は感じられるものの、力を感じることはなかった。

次に、アクチュエータにデューティー比の異なる矩形信号を印加してみると、仮想力覚が感じられることを発見した。図 2 (4) がそれで、周波数 125Hz でデューティー比 2:6(2 ms on, 6 ms off) の信号を与えると、アクチュエータが一方向に引っばられるような力を感じるようになった。図からも明らかのように、このときアクチュエータが非対称な加速度運動をしていることがわかる。また、デューティー比を変化させることで、仮想力覚の方向も制御することがわかった。図 2(4) では下向きの力を感じ

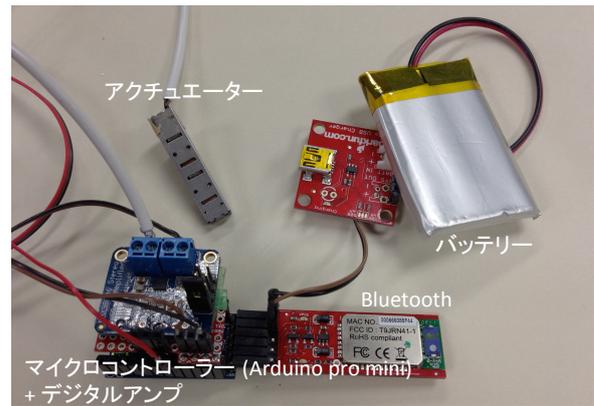


図 3. 試作システムの構成: マイクロコントローラー (Arduino) が Bluetooth 経由で命令を受信しアクチュエーター用の駆動波形を生成する。信号増幅にはオーディオ用デジタルアンプを利用している。

るが、(5) のデューティー比 6:2(6 ms on, 2 ms off) では逆に上向きの力を感じる。以上の場合でも、コイルに印加された信号が単純に振動に反映されているわけではなく、コイルと板バネの効果が複合していることを指摘しておく。

ここで、仮想力覚の発生原因について考察する。アクチュエーターは外部と機構部や糸などで支えられている訳ではないので、振動加速度を一定時間積分すればその値はゼロになり、現実には力は発生していない。したがって感じられる力は人間の知覚機構によるある種の錯覚であると推定される。仮想力覚の発生原理については Stevens の幂法則 (Stevens's power law) [13] と呼ばれる、刺激とそれに伴う感覚の強さの関係によるものが仮説の一つである。Stevens の幂法則は、刺激 I による感覚の強さ S が $S = kI^\alpha$ となるというものである (ここで k は比例定数、 α は刺激の種類により決まる係数)。測定された加速度の積分値はゼロになるが、このようにして知覚された感覚 S の積分値はゼロにはならず、力として感じられるのではないかと推測している。仮想力覚の発生原因として考えられる他の要因は、刺激のマスク効果である。強い刺激の直後の弱い刺激がマスクされるという現象である。図 2(4),(5) では大きな加速度の直後に (反対方向への) 小さな加速度運動が連続しているので、この部分の知覚がマスク効果によって抑制されれば、やはり力覚を感じるようになる。

今回のデバイスでは、アクチュエーターをデューティー比の異なる矩形信号で駆動した場合に、コイルと板バネによる構成によりその駆動信号が変形された結果、力覚を感じる非対称な加速度を生み出していると推定される。

システム構成

本研究で利用しているシステム構成を図 3 に示す。駆動波形を発生させるマイクロコントローラー

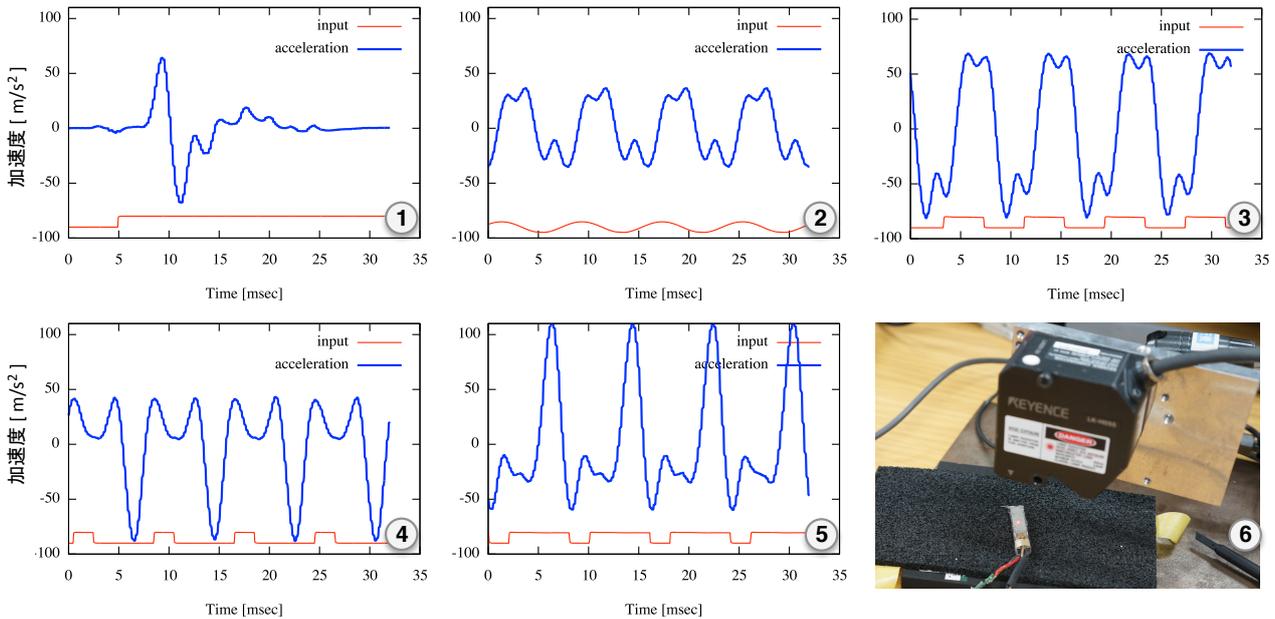


図 2. Traxion によって駆動される振動形状：デューティー比が異なる入力信号により異なる加速度が形成される。(1) 単一パルス印加と対応する加速度 (2) 正弦波 (3) 矩形波 (デューティー比 1:1) (4) デューティー比 2:6 (5) デューティー比 6:2 (6) 加速度計測環境

(Arduino) が Bluetooth 経由で指令を受け取りアクチュエーターを駆動する。信号増幅のためにオーディオ用デジタルアンプを利用している。前章の実験により、仮想力覚を発生させるための波形は矩形で充分なことが分かったので、マイクロコントローラーの出力端子 (0V/5V を切り替え可能) を直接オーディオアンプの入力端子に接続し、DA 変化を省略することで構成の簡素化を実現している。

3 評価

3.1 実験 1: 一次元仮想力覚デバイス

Traxion の効果を確認するために利用者評価実験を行った。実験参加者は 10 名で、全員コンピュータサイエンス関係学科の大学院生である。参加者はアクチュエーターを指で保持し、仮想力覚が感じられたときにその方向を回答する。アクチュエーターを駆動する時間は 640, 320, 160, 80, 40 ms で、正方向と逆方向の仮想力覚を作り出す 2 つの振動パターン (2 ms:6 ms と 6 ms:2 ms のデューティー比) との組み合わせを提示する。振動時間、振動パターンの組み合わせはランダムに選択されている。各参加者は、それぞれの振動時間に対して 5 回の刺激提示・回答のタスクを行う。回答はゲームコントローラーのスティックによって入力する。振動以外の情報を遮断するため、参加者はアイマスクとノイズキャンセルヘッドフォンを装着している (図 4:左)。

評価結果を図 4(右) に示す。グラフに示されているように刺激時間が短縮されると若干正答率が落ちるものの、いずれも高い正答率を示しており、この

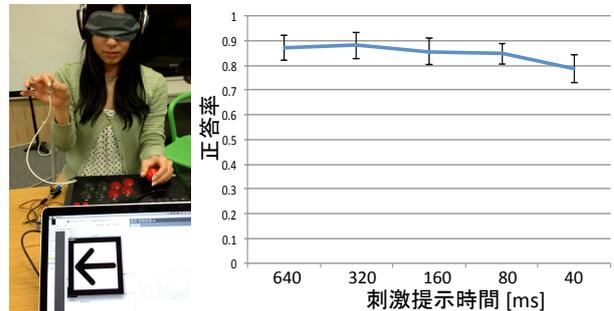


図 4. 実験 1: 一次元仮想力覚の評価実験環境および結果

仮想力覚が複数の実験参加者が同様に感ずる感覚であることが確認できた。

実験後のインタビューによると、多くの実験参加者が、指で保持しているアクチュエーターを動かしてみると仮想力覚をより明確に感じられるという意見が多く聴かれた。とくに力覚の方向にアクチュエーターを動かした時にこの効果は顕著だった。この効果に対する仮の説明は以下のようである。現実世界での牽引力、たとえばゴムやスプリングによるものは、牽引する方向に物体を動かすことで力が弱くなるものが多い。一方、仮想力覚の場合、移動させても感じられる力はとくに減らないので、それを「逆算して」元の力が強くなったと錯覚するのではないか、という可能性が考えられる。

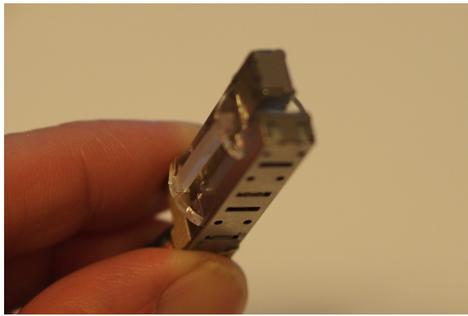


図 5. 2次元仮想力覚デバイスへの拡張

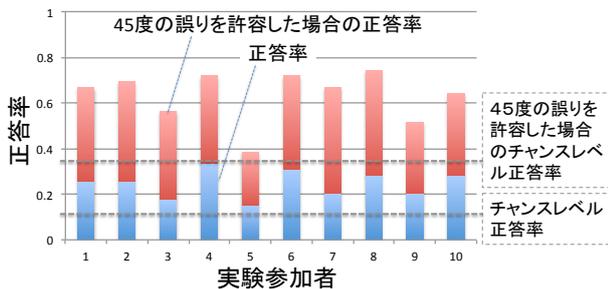


図 6. 実験 2: 二次元仮想力覚提示実験結果

3.2 実験 2: 二次元仮想力覚デバイス

次に、本デバイスを組み合わせることで多次元の仮想力覚が提示可能か検証した。図 5 のように、2つのアクチュエーターを、それぞれの牽引方向が直角になるように接合した。このデバイスで、同時に加速力覚を提示した場合に、その合成方向に力覚が感じられるかを評価した。

実験 1 と同じ 10 名が評価実験に参加した。アクチュエータを指二本で保持し、力が感じられた方向を答える。この実験では 45 度ごとに 8 方向からいずれかを回答するようにした。刺激提示の時間は 640 ms で固定し、デューティー比は 2 ms:6 ms を用いた。縦横方向の仮想力覚はどちらか一つのアクチュエーターをその方向の力覚が提示されるように駆動し、斜め方向は両方のアクチュエーターを駆動した。参加者は実験 1 と同様、アイマスクとノイズキャンセルヘッドフォンを装着し、ゲームスティックによって回答した。

この実験では、図 6 で示すように正答率はチャンスレベルを上回るものの、正しく力覚が感じられているとは言いがたい状況だった。複数のアクチュエーターを同時に駆動しても、全体の振動が斜め方向を力覚を感じさせるような振動としては合成されないことが主な理由であると考えているが、複合アクチュエータでの実際の振動の解析を今後実施する予定である。

3.3 実験 3: 仮想力覚量の推定

では、発生している仮想力覚の強さはどのくらいだろうか。これを評価するために実験 3 を計画・実施

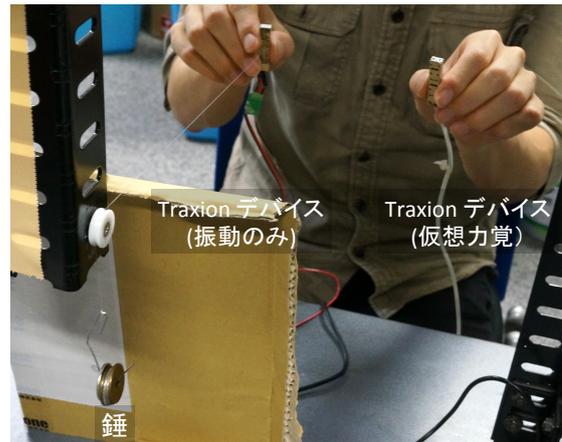


図 7. 実験 3: 仮想力覚量の評価

した。実験参加者は、仮想力覚を発生するアクチュエータと、振動はしているが仮想力覚が発生しないように印加信号を調整したアクチュエーターを両手で保持する。力覚が発生していないアクチュエーターにテグスと滑車を介して錘を取り付け、牽引力が発生するようにしている。それぞれのアクチュエーターを交互に駆動し、錘の重さを変え、ちょうど同じ位の牽引力だと感じたときの重さを記録する。実験参加者には、錘の重さを感じるために、それぞれのアクチュエーターをゆっくりと前後に動かすことをお願いした。

それぞれの錘に対して、「錘の方が重い／軽い／同じ位」と回答してもらい、同じ位と答えた値を実験結果として記録した。実験 1,2 と同じ 10 名が実験に参加した結果、平均値は 29.8 g (0.292 N) 標準偏差は 8.5 g (0.083 N) だった。この実験により、発生している仮想力覚がおおむね 30 g 程度であると推定される。

ただし、この実験条件の場合、片方の手は錘の重量を常に支えているが、仮想力覚は振動しているときしか発生しない。したがってこの二つを単純に比較して正しいかどうかは疑問が残る。今後実験条件を変更するなどして仮想力覚の推定精度を上げていきたい。

4 応用

4.1 方向の提示

仮想力覚は、歩行者を誘導するために有効だと考える。たとえば指輪型のアクチュエータに仮想力覚を発生させて、誘導した方向へ牽引することができれば、視覚情報によらず空間ナビゲーションができるかもしれない。また、たとえば物の位置を何らかの手段で認識できるとすると、その物に向かって指先を牽引するデバイスなどが考えられる。また視覚障害者のナビゲーション手段としても利用できるのではないかと考えている。

こういった位置誘導応用の可能性を検証するため

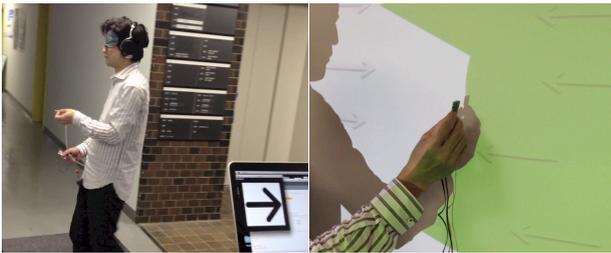


図 8. Traxion アプリケーション例: (左) 歩行者の誘導, (right) 仮想力覚つき入力装置. プロジェクターの照射によって力場を作り出す. DLP プロジェクターにより力の方向を光の点滅パターンとして照射する. 仮想力覚つき入力デバイスでその光をデコードし, 力覚に変換する.

に, 簡単な実験を行った. 利用者はアイマスクを装着し, 指で保持したアクチュエーターの牽引する方向に曲がるようにする. たとえば廊下の角に来た時に右に曲がるか左に曲がるかを仮想力覚によって支持する. 今回の実験では, 直進する場合には力覚は提示しなかった. また方向の支持はウィザードオブウォズ方式で, 実験参加者を目視して必要なナビゲーション支持をワイヤレスで与えるようにした (図 8(左)). 実験の結果, 視覚情報にたよらずとも歩行が可能で, 適切な場所で曲がれることが分かった.

4.2 仮想力覚つき入力デバイス

他の応用可能性として, 仮想力覚による入力デバイスが構想できる. もしスタイラスなどに仮想力覚を組み込むことができれば, アプリケーションによって牽引力や斥力を発生させることが出来る (8(右)). 従来の力覚付き入力デバイスはスタイラス全体をロッドによって保持してため作業範囲に制約があったが, 仮想力覚ではその制限がない.

図の例では, 作業空間に力の場を動的に作り出している. DLP プロジェクターの照射光で力場を定義する. DLP プロジェクターは, 色ごとに光の点滅パターンが異なるので, 簡易な可視光通信装置として使うことができる. 入力デバイスのフォトトランジスターでこの点滅を受信し, 色ごとにことなった力の方向を発生させれば, プロジェクターで照射する図形によって力場を定義することができる [12].

たとえば, 手術をする際のメスの動きを仮想力覚によってガイドするような応用が考えられる. 患者の体表に力場をプロジェクションし, メスに光センサーと仮想力覚アクチュエーターを装備すれば, 切開すべき場所を力覚として感じるができるようになる.

5 関連研究

他の仮想力覚デバイスとの比較

前述したように, 回転体やクランク機構によって仮想力覚を生み出すデバイスが提案されている [14,

2, 3] が, 本論文が提案するデバイスと比較すると重量やサイズが大きなものとなっている. また, 回転体は急激に回転数を変化させたり逆転させることができない. このことにより, これらの装置では単方向の一定量の仮想力覚を作り出すことができても, その方向を逆転させたり, 力覚量を制御することが難しい. これに対して, 本論文で提案しているデバイスはより単純な構造ながら仮想力覚を提示することが可能で, 既存事例から大幅に小型軽量化されている (重量比で約 40 分の 1) 点が利点と言える.

また, [2] では評価実験の結果, 周期運動が 40 Hz 以上になると仮想力覚が消失していくことを報告している. しかし, 本論文のデバイスでは振動数が 150 Hz 付近でも安定して仮想力覚が感じられた. したがって従来の回転体に基づく方式と本研究の構成では仮想力覚の発生原理が異なっている可能性もある. これについては今後さらに精密な検証をすすめていきたい.

力覚によるナビゲーション

ActiveBelt はベルト状の装着型デバイスで, 複数のアクチュエーターの振動によって, 利用者を誘導することができる [15]. 上述のように, Traxion では方向を仮想力覚で提示することで, 利用者を特定の方向に誘導することを試みている.

力覚提示素子

触覚によるフィードバックは, タッチパネルに組み込まれたものが多く提案, 実用化されている [4, 11]. われわれが利用している素子は, ゲームコントローラー用に開発されたものである [1] が, 他の原理による力覚提示素子でも仮想力覚が発生するかどうかは興味深い課題である. ワイヤは機構を利用せずには触感フィードバックを発生させるものとしてアレイ状の超音波トランスデューサーによるものがある [6] が, 本論文で示しているような力覚をつくりだすものではない.

6 結論・今後の課題

本論文では仮想力覚を生成する触覚アクチュエーター Traxion について報告した. 振動子の駆動パターンを工夫することにより, 仮想的な「力」を生成することが可能になった. 評価実験によりこの効果を複数の利用者によって確認した. また, 仮想力覚の大きさも評価し, 約 30 g 相当の力が生成されていると推定した. このデバイスは既存事例の力覚装置と比較して重量比で約 1/40 であり, インタラクションデバイスとしての利用可能性が高まった.

仮想力覚は人間の刺激と知覚の関係に基づいて力覚を「錯覚」として感じさせるものであり, 物理的な力は発生していない. また, 従来の機構部品による力覚フィードバックのように, 利用者の行動を束縛するほどの大きな力を作り出すことはできない. 一方, 振動フィードバックとは異なり, 明確に力の方向を感じることができる. したがって, 従来の力

覚フィードバックとも振動フィードバックとも異なるタイプの触覚フィードバックだということができる。例えば、歩行者ナビゲーションで移動方向に牽引するアプリケーションを振動フィードバックで実現しようとする、「右」「左」などの方向を特定の振動パターンと対応づける必要があり、利用者がこの対応づけを記憶理解しなければならない。仮想力覚ではこのようなマッピングが不要で直接方向を感じさせることができる。

現状では既存の特定のタクトイルアクチュエーターの駆動パターンによって力覚が発生することを確認した段階だが、他の振動子でも同じ効果が発生するかどうかは興味ある課題である。実際、tactor [8] と呼ばれるコイル型の振動子でも力覚発生量は弱いながらも類似の仮想力覚が発生することを確認した。また、今回利用したアクチュエーターは、上下方向に非対称の構造を持っており、上方向の力覚と下方向の力覚の量が若干異なるように感じられる。今後は各種のアクチュエーターの駆動信号と加速度を計測し、仮想力覚の発生原理をより厳密に確認していきたい。

一方、評価実験の結果、一次元の仮想力覚デバイスを単純に組み合わせても、合成力覚を作り出すことができないことが分かった。これについても振動子の運動や加速度を解析し、可能性を探求していきたい。

アプリケーションとして、DLP プロジェクターによる可視光通信により、プロジェクションされた平面に容易に力場を形成できることを示した。これを仮想力場と呼ぶ。この発想を発展させると、各種のサーフェスコンピューティングにおいて、投影されている物体と現実のインタラクティブデバイス間に引力や斥力を作り出すようなインタフェースが可能になる。また、現実の物体に投影した場合には、その物体の解析結果と仮想力場が対応するような応用が可能になる。たとえば、サーモカメラで撮影した温度映像から、温度勾配を力覚として感じるようなインタフェースや、手術の際のメスの力覚によるナビゲーションなどが考えられる。

参考文献

- [1] ALPS. Alps Electric Press Release Force Reactor™. http://www.alps.com/e/news-release/2005/0608_01.html, 2008.
- [2] T. Amemiya, H. Ando, and T. Maeda. Phantom-DRAWN: direction guidance using rapid and asymmetric acceleration weighted by nonlinearity of perception. In *Proceedings of the 2005 international conference on Augmented tele-existence*, ICAT '05, pp. 201–208, 2005.
- [3] A. Badshah, S. Gupta, D. Morris, S. Patel, and D. Tan. GyroTab: a handheld device that provides reactive torque feedback. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 3153–3156, 2012.
- [4] M. Fukumoto and T. Sugimura. ActiveClick: Tactile Feedback for Touch Panels. In *CHI 2001 summary*, pp. 121–122, 2001.
- [5] Y. Hirata and M. Sato. 3-dimensional Interface Device For Virtual Work Space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 889–896, 1992.
- [6] T. Iwamoto, M. Tatezono, and H. Shinoda. Non-Contact Method for Producing Tactile Sensation Using Airborne Ultrasound. In *Proc. EuroHaptics 2008*, Vol. 5024 of *LNCS*, pp. 504–513. Springer, 2008.
- [7] H. Iwata. Artificial reality with force-feedback: development of desktop virtual space with compact master manipulator. *SIGGRAPH Comput. Graph.*, 24(4):165–170, 1990.
- [8] T. Labs. Haptuator Mark II Product specification. <http://www.tactilelabs.com>.
- [9] M. Minsky, O. young Ming, O. Steele, J. Frederick P. Brooks, and M. Behensky. Feeling and seeing: issues in force display. *SIGGRAPH Comput. Graph.*, 24(2):235–241, 1990.
- [10] N. Nakamura and Y. Fukui. Development of Fingertip Type Non-grounding Force Feedback Display. In *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*, pp. 582–583, 2007.
- [11] I. Poupyrev, S. Maruyama, and J. Rekimoto. Ambient touch: designing tactile interfaces for handheld devices. In *ACM UIST 2002*, pp. 51–60, 2002.
- [12] J. Rekimoto. SenseableRays: Opto-Haptic Substitution for Touch-Enhanced Interactive Space. In *CHI '09 extended abstracts*. ACM, 2009.
- [13] S. Stevens. On the Psychophysical Law. *Psychological Review*, 64(3):153–181, 1957.
- [14] Y. Tanaka, M. Sakai, Y. Khono, Y. Fukui, J. Yamashita, and N. Nakamura. Mobile Torque Display and Haptic Characteristics of Human Palm. In *ICAT 2001*, pp. 115–120, 2001.
- [15] K. Tsukada and M. Yasumura. Activebelt: Belt-type wearable tactile display for directional navigation. In *Lecture Notes in Computer Science. Springer-Verlag GmbH, Oct 2004*, pp. 384–399. Springer, 2004.
- [16] 雨宮 智浩, 杉山 久. ”牽引力錯覚を利用した牽引式羅針盤による視覚障がい者の歩行誘導の評価”. *信学技報*, 109(27, WIT2009-43):215–220, 2009.

HandyScope : 引き出しジェスチャを用いたテーブルトップ用遠隔地操作手法

栗原 拓郎 三田 裕策 大西 主紗 志築 文太郎 田中 二郎*

概要. 大型のマルチタッチテーブルトップでは、ユーザはその大きさのため遠隔地に手が届かず、操作が困難である場合がある。そのような場合、ユーザはその領域に手が届く位置まで移動することを求められる。そこで我々は遠隔地操作手法である HandyScope を示す。HandyScope では、ユーザは手元から遠隔地を操作することができる。また、ユーザは手元と遠隔地の間にてオブジェクトを移動させることができる。加えて、HandyScope の起動、操作には引き出しジェスチャを用いることにより、ユーザは従来のマルチタッチ操作と競合することなく素速くポインティングを行うことができる。我々は HandyScope の性能を測るために従来のタッチ操作と比較実験を行い、遠隔地を選択する場合 HandyScope は有用であること及び、オブジェクトまでの距離が遠くなると HandyScope がより有用な手法であることを確認した。

1 はじめに

大型のマルチタッチテーブルトップを使用する際、ユーザはテーブルトップの周りに立ち、その位置からタッチ操作を行う。このタッチ操作を行う範囲について、ユーザは自身の位置から 34cm 以内の位置においてその 90%を行っていると Toney らは述べており [11]、それ以上遠くに位置する手の届かない領域（遠隔地）に対してタッチ操作を行うことは困難である。

そこで我々は遠隔地操作手法である HandyScope を示す。HandyScope では、ユーザは 2 個の円形のウィジェットを用いて遠隔地を操作することができる。2 個のウィジェットとは、遠隔地へ送り操作範囲を決定するサークル (scope) と手元において操作するためのサークル (handler) である。scope 内の領域は handler にも表示され、ユーザは handler に表示された領域を操作することにより遠隔地を操作することができる。また、ウィジェットを経由することにより、ユーザは手元と遠隔地の間にてオブジェクトを移動させることができる。加えて、本手法では Control-Display (CD) 比を動的に変更できるため、ユーザは素速く目標領域まで scope を移動させること (ポインティング) ができる。

本手法では吉川らの提案する、両手を用いた引き出しジェスチャ [13] を本手法の起動及び scope の位置を決定するために使用する。引き出しジェスチャでは、ユーザは別のデバイスを使用せずに、タッチ操作のみにて使用することができる。また、引き出

しジェスチャは従来のマルチタッチ操作と競合しないため、複数ユーザが同時に使用できる。テーブルトップでは複数ユーザが同時に作業することも多く、本手法はその際にも有用である。

2 関連研究

テーブルトップ環境における遠隔地ポインティング手法については多くの研究がなされている。Parker らは、スタイラスの先端の影を用いてポインティングを行う手法を示した [9]。また Banerjee らは片手の指を指す動作によってポインティングを行う手法を示した [2]。これらの研究は、ポインティング手法を実現するために追加のデバイスが必要となる。一方、我々の手法は追加のデバイスを用いる必要がない。また、マルチタッチテーブルトップ環境における仮想マウスを提案した研究もある [4, 8]。これらの研究では手の接触形状を認識するマルチタッチテーブルトップが必要である。一方、我々の手法では手の接触形状を認識する必要がないため、複数のタッチ点を検出可能なマルチタッチテーブルトップのみにより実現できる。

また、両手を用いたポインティング手法の研究も行われている。I-Grabber [1] はマルチタッチインタラクションにより操作するウィジェットである。両手を用いたポインティングを採用している点において我々の手法と同様であるが、我々の手法では CD 比を動的に変更でき、かつポインティング開始からポインティング先の決定までを一連のジェスチャにて行うことができる。そのため、ユーザは素速く正確にポインティングを行うことができる。所らは 2 つの加速度センサを用いた両手によるポインティング手法を示した [10]。また、Malik らは画像処理を用いた両手によるポインティング手法を示した [7]。こ

Copyright is held by the author(s).

* Takuro Kuribara and Yusaku Mita, 筑波大学大学院システム情報工学研究科 コンピュータサイエンス専攻, Kazusa Onishi, 筑波大学 情報学群 情報科学類, Buntarou Shizuki and Jiro Tanaka, 筑波大学 システム情報系.

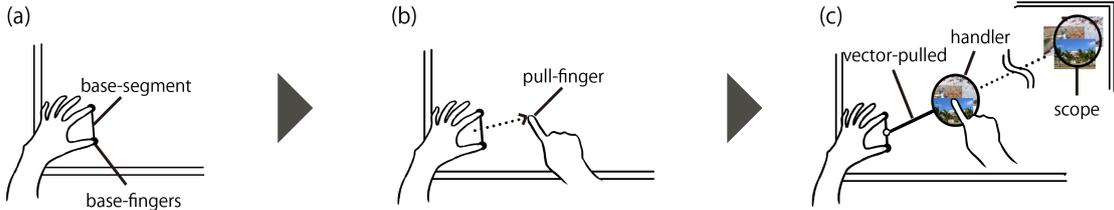


図 1. HandyScope の起動及び操作方法. a) 2本の指を置く, b) 1本の指にて2本の指の間を横切ることにより起動, c) HandyScope の操作

これらの研究と異なり, 我々の手法ではテーブルトップを対象に, タッチを用いた両手によるポインティングを行う. そのため他のデバイスを必要としない.

3 HandyScope

本節では, HandyScope の操作手法及び, HandyScope を用いたオブジェクトの操作手法について述べる.

3.1 起動及び操作方法

HandyScope の起動及び操作方法を図 1 に示す. ユーザは最初に非利き手の 2本の指 (base-fingers) をテーブルトップ上に置く (図 1a). 次に, 利き手の 1本の指 (pull-finger) により base-fingers の間 (base-segment) を横切る操作を行う (図 1b) と, handler が pull-finger を中心とした位置に表示される. 同時に, base-segment の中心点と pull-finger の間 (vector-pulled) の半直線上に scope が表示される. ユーザが vector-pulled を変更すると, その変更に応じて scope の位置が変更される. 図 1c に HandyScope を用いた操作の様子を示す. HandyScope は base-fingers をテーブルトップから離すことにより終了される.

3.2 scope の位置の決定

base-segment が生成されてから i フレーム後における scope の位置 P_i は以下の式により決定される. ここで S_0, S_i は図 2 に示すように base-segment の長さであり, S_0 は base-segment が生成された時の長さ, S_i は base-segment が生成されてから i フレーム後の長さである. また, G_i は base-segment の中心であり, V_i は G_i から pull-finger までの vector-pulled である. また, α は定数であり, k_i は CD 比を表す. base-segment を拡大すると CD 比を表す k_i が大きくなり, 逆もまた同様である. すなわち, 利き手もしくは非利き手を移動させることにより V を変化させると, その変位である ΔV_i に基づき, scope の位置が変化する.

$$P_i = G_0 + \sum_j^i k_j \Delta V_j,$$

$$\Delta V_i = V_i - V_{i-1},$$

$$k_i = \alpha \frac{|S_i|}{|S_0|}. \quad (1)$$

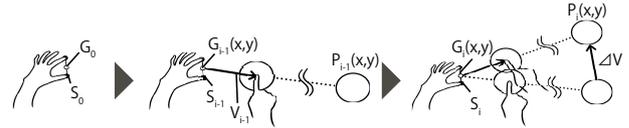


図 2. サークルの位置

3.3 CD 比の動的な変更

CD 比は base-segment の長さにより変更される. 図 3 に CD 比と base-segment の長さの関係を示す. ユーザは CD 比の変更を行いながらポインティングを行うことができるため, 大きな CD 比による大きな操作と小さな CD 比による細かい操作を動的に切り替えて操作することができる. 例えば, 図 4 に示すように大きな CD 比により素早く scope を移動させ, 小さな CD 比により微調整を行うことができる.

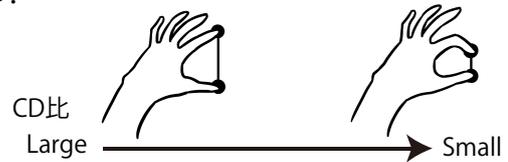


図 3. CD 比と base-segment の長さの関係

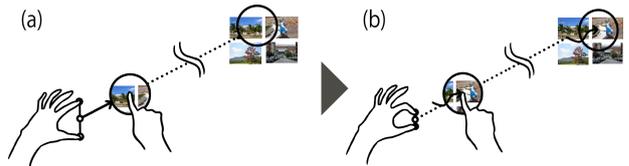


図 4. CD 比の動的な変更による scope の操作. a) 大きな移動によるポインティング先の決定. b) 小さな移動による精密なポインティング先の決定

3.4 サークルを経由した遠隔地の操作

図 5 に示すように, handler 内には scope 内の領域が表示されている. また, handler 内での操作は全て scope 内に適用される. 図 5a に遠隔地のオブジェクトを拡大縮小している様子を, 図 5b にオブジェクトを回転している様子を示す. 遠隔地に対する操作を全て handler 上にて行うことができるため, ユーザは自身が遠隔地に移動して操作することや, 遠隔地のオブジェクトを手元に移動させることなく, 遠隔地のオブジェクトを操作することができる.

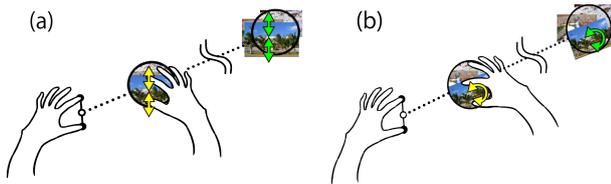


図 5. 遠隔地のオブジェクトの操作. a) オブジェクトの拡大縮小, b) オブジェクトの回転

3.5 サークル間のオブジェクトの移動

handler 内のオブジェクトをタッチしてサークル外までドラッグ操作を行うと、遠隔地のオブジェクトが手元に移動する。実際に移動している様子を図 6a に示す。同様に、図 6b に示すように手元にあるオブジェクトを handler 内にドラッグすることにより遠隔地にオブジェクトを移動させることができる。これにより、遠隔地と手元の間においてオブジェクトの移動を素早く行うことができる。

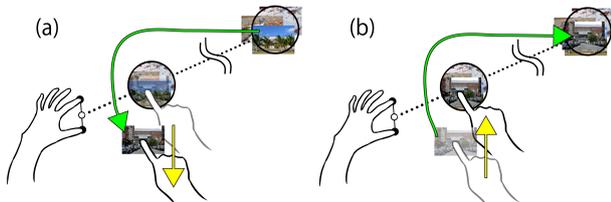


図 6. サークル間のオブジェクトの移動. a) 遠隔地から手元にオブジェクトを移動, b) 手元から遠隔地にオブジェクトを移動.

3.6 サークルの再移動

HandyScope の起動時は pull-finger を中心としてサークルが表示されるが、一度 pull-finger をテーブルトップから離すと、handler 内に対する操作は scope 内に対する操作として扱われる。そのため、図 7 に示すように再度サークルを移動させる場合には handler の中央ではなく、枠をドラッグする。これにより、scope の位置を移動させることができる。

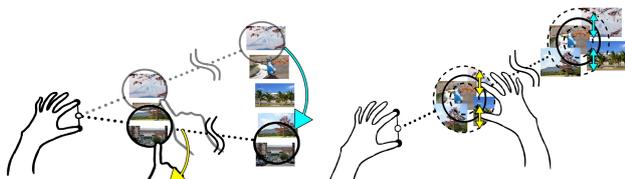


図 7. サークルの再移動

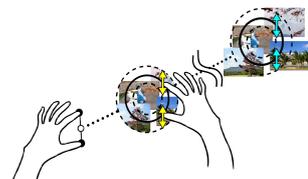


図 8. サークルの大きさの変更

3.8 HandyScope のメリット

HandyScope では、ユーザは非利き手の 2 本の指により CD 比を変更し、この 2 本の指の中心点と利き手の指の間の相対位置によりポインティング先を決定する。ポインティング先には scope を、手元には handler を表示し、ユーザはこのサークルを介して遠隔地に対するインタラクションを行う。これにより Frisbee[6] 及び Dynamic Portals[12] と同様に遠隔地を手元から操作することが可能になる。また、Frisbee では任意の位置から起動できず、Dynamic Portals では操作対象としたい遠隔地の指定に他者の協力を必要とするが、HandyScope では、ユーザは自身の操作のみにて任意の位置から起動し、動的な CD 比を用いて操作対象としたい遠隔地を素速く決定することができる。

4 実験

HandyScope の性能を調査するために、比較実験を行った。比較実験では HandyScope (HandyScope 条件) 及び従来の直接タッチ (タッチ条件) の性能をテーブルトップ上の典型的な操作である選択、回転、拡大縮小の 3 種類のタスクにおいて比較した。

4.1 被験者及び実験環境

被験者は HandyScope を使用したことのない 10 名 (20 歳から 24 歳の大学生及び大学院生) であり、右利き 9 名、左利き 1 名であった。実験環境を図 9 に示す。実験に用いたテーブルトップは、画面サイズが 147cm × 80cm の 60 インチディスプレイ (パイオニア社, PDP-607CMX¹) に PQLab 社のマルチタッチフレーム (PQ Lab, Multi-Touch G³²) を装着することによりマルチタッチ機能を付したものである。なお、テーブルトップに関する幾つかの研究 [3, 5, 14] において、テーブルトップの高さを 91cm から 105cm としていたため、本実験ではテーブルトップの高さをその範囲内である 93cm とした。また、初期の CD 比にてテーブルトップの端までポインティング可能となるように式 1 における α を 12 とした。

4.2 実験手順

被験者には、選択タスク、回転タスク、拡大縮小タスクをこの順に課した。タスクは、テーブルトップ上でのポインティング手法を提案した研究 [2] に做ったタスクである。各タスク前には、本番と同じタスクを本番の 1/4 の量だけ練習タスクとして課した。各タスクを行う際には、操作条件間の順序効果の打ち消しを狙い、被験者を HandyScope 条件を先

¹ <http://pioneer.jp/biz/karte/PDP-607CMX.html>

² http://multi-touch-screen.com/product_g3.html

に行う者とタッチ条件を先に行う者の2組に分けた。被験者には全てのタスクを終了した後にアンケートに回答して貰った。なお、被験者1人あたりの実験時間が約1時間半となったので、拘束時間に対する謝礼を渡した。

4.3 選択タスク

被験者には様々な位置に表示されるオブジェクトを選択して貰った。

まず、被験者は各試行の開始前にテーブルトップの短辺の中心（すなわち図9中の被験者の足元の床に、黒色のビニールテープにて示されている場所）に立つ。この状態から、テーブルトップ上のいずれかの提示位置に表示されるオブジェクトを選択する。なお、試行の開始前に開始点及びオブジェクトは表示されている。

テーブルトップ上に表示される開始点及びオブジェクトの提示位置を図10に示す。

HandyScope条件では、被験者は開始点上においてHandyScopeを起動させることにより試行を開始する。次にオブジェクトにscopeを合わせ、オブジェクトをタップする。オブジェクトがタップされると1回の試行が終了し、フィードバックとしてビープ音が発生する。タッチ条件では、被験者は開始点をタップすることにより試行を開始する。次に表示されたオブジェクトに手が届く位置まで移動して、オブジェクトをタップする。



図9. 実験環境

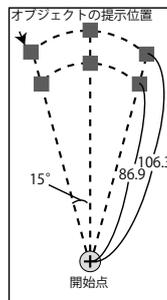


図10. 開始点及び提示位置

本実験における独立変数は、開始点からオブジェクトまでの距離（86.9, 106.3cm）、開始点からオブジェクトまでの角度（15, 0, 15度）、オブジェクトの大きさ（3.9, 5.8, 7.7cm）、操作条件（HandyScope, 直接タッチ）である。被験者は各独立変数の組み合わせにおいて試行を3回ずつ、合計108試行（2 3 3 2 3）を行った。また、操作条件以外の試行の順序はランダムであった。

実験結果

両操作条件における1試行の所要時間を図11の左2つのグラフに示す。HandyScope条件では所要時間は1715ms、タッチ条件では1943msであった。

被験者毎の所要時間において対応のあるt検定を行った結果、HandyScope条件における操作が有意に速かった（ $t(9) = 2.72, p = .011 < .05$ ）。

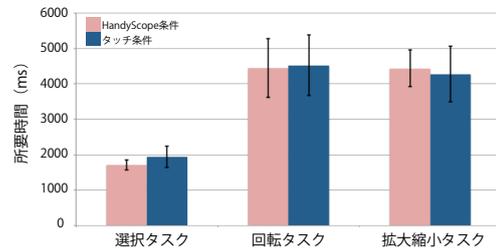


図11. 各タスク毎の1試行の所要時間

4.4 回転タスク

被験者には様々な提示位置に様々な角度にて表示されるオブジェクトを回転させて、目標（ドック）に合わせて貰った。ドックの表示位置をオブジェクトと同位置、表示角度をオブジェクトと異なる角度とした。開始点及び提示位置、試行の開始方法は選択タスクと同じである。

HandyScope条件では、被験者はHandyScopeを使用してオブジェクトを回転させてドックに合わせる。オブジェクトとドックの角度が等しく（誤差±5度以内）なるとオブジェクトの縁が赤色になる。この状態において操作を終了すると1回の試行が終了し、フィードバックとしてビープ音が鳴る。タッチ条件では、表示されたオブジェクトに手が届く位置まで移動して操作を行う。

本実験における独立変数は、開始点からドックまでの距離（86.9, 106.3cm）、開始点からドックまでの角度（15, 0, 15度）、ドックの大きさ（5.8, 7.7cm）、回転角度（45, 45度）、操作条件（HandyScope, 直接タッチ）である。被験者は各独立変数の組み合わせにおいて試行を2回ずつ、合計92試行（2 2 3 2 2 2）を行った。また、操作条件以外の試行の順序はランダムであった。

実験結果

両操作条件における1試行の所要時間を図11の中央2つのグラフに示す。HandyScope条件では所要時間は4444ms、タッチ操作では所要時間は4520msであった。被験者毎の所要時間において対応のあるt検定を行った結果、操作手法による所要時間に有意差はなかった（ $t(9) = .267, p = .397 > .05$ ）。

4.5 拡大縮小タスク

被験者には提示位置に表示されるオブジェクトを拡大縮小させて、ドックに合わせて貰った。ドックはオブジェクトと同位置に異なる大きさで表示される。開始点及び提示位置、試行の開始方法は選択タスクと同じである。

HandyScope条件では、被験者はHandyScopeを使用してオブジェクトを拡大縮小してドックに合わ

せる。オブジェクトとドックの大きさが等しく（誤差 ± 4.8mm 未満）なるとオブジェクトの縁が赤色になる。この状態において操作を終了すると1回の試行が終了し、フィードバックとしてピープ音が鳴る。タッチ条件では、表示されたオブジェクトに手が届く位置まで移動して操作を行う。

本実験における独立変数は、開始点からドックまでの距離（86.9, 106.3cm）、開始点からドックまでの角度（15, 0, 15度）、ドックの大きさ（5.8, 7.7cm）、ドックに対するオブジェクトの拡大・縮小（1.5倍, 0.67倍）、操作条件（HandyScope, 直接タッチ）である。被験者は各独立変数の組み合わせにおいて試行を2回ずつ、合計92試行（2 × 2 × 3 × 2 × 2）を行った。また、操作条件以外の試行の順序はランダムであった。

実験結果

両操作条件における1試行の所要時間を図11の右2つのグラフに示す。HandyScope条件では所要時間は4438ms、タッチ条件では所要時間は4278msであった。被験者毎の所要時間において対応のあるt検定を行った結果、操作条件間における所要時間に有意差はなかった（ $t(9) = .935, p = .187 > .05$ ）。

4.6 実験結果の考察

上述のように、選択タスクにおいてはHandyScopeを使用した操作が有意に速く、回転及び拡大縮小タスクにおいては有意差が現れなかった。この結果より遠隔地を選択する場合、HandyScopeは有用であると言える。

一方、回転及び拡大縮小タスクにおいて操作条件間に有意差が見られなかった原因として、HandyScopeの再起動に時間がかかることが挙げられる。実験において、被験者がオブジェクトをドックに合わせたつもりでHandyScopeを終了したが、実際には合っていないため試行が終了せず、再度HandyScopeを起動してポインティングを行う様子が見られた。直接タッチ操作ではユーザがオブジェクト付近に移動して入力を行うため、再度入力を行う際に時間がかからないが、HandyScopeは一度終了してしまうとポインティングを再び行う必要がある。これにより、HandyScopeによる回転及び拡大縮小タスクに時間がかかったと考える。

また、HandyScopeは遠隔地を操作することを想定して設計されているため、より遠くの位置に対する操作では大きな差が出ることが予想される。HandyScopeが遠距離の操作に対して有用な手法であることを確認するため、それぞれのタスクにおけるオブジェクトまでの距離毎の所要時間を用いて、操作条件における距離による特性を調べた。この結果を図12に示す。また、各タスクの各距離において、操作条件による有意差を確かめるため、対応

のあるt検定を行った。選択タスクにおいて、オブジェクトまでの距離が106.3cmの試行（ $t(9) = 3.13, p = .006 < .01$ ）は、距離が86.9cmの試行（ $t(9) = 2.16, p = .029 < .05$ ）と比較して、操作条件間における所要時間の差がより大きかった。回転タスクにおいて、オブジェクトまでの距離が106.3cmの試行（ $t(9) = .772, p = .230 > .05$ ）と距離が86.9cmの試行（ $t(9) = .357, p = .364 > .05$ ）は共に操作条件による有意性が現れなかったが、所要時間は距離が106.3cmの試行ではHandyScope条件の方が速く、距離が86.9cmの試行ではタッチ条件のほうが速かった。拡大縮小タスクにおいて、オブジェクトまでの距離が3.9cmの試行（ $t(9) = 1.48, p = .086 > .05$ ）と距離が106.3cmの試行（ $t(9) = .321, p = .378 > .05$ ）は共に操作条件による有意性が現れなかった。これらの結果から、オブジェクトまでの距離が遠くなるとHandyScopeが有用な手法であることが確認できた。

本実験のタッチ操作において、遠隔地のオブジェクトを操作する際、ユーザはテーブルトップの外周に沿って2, 3歩の移動を行っていた。オブジェクトまでの距離が遠くなると歩数が増えるため、操作条件間の有意差がより大きくなると思われる。

また、本実験ではHandyScopeの初期のCD比を、テーブルトップの端までポインティング可能となるように設定したため、CD比の変更を行う被験者はほぼ見られなかった。このため、CD比の動的な変更について評価を得るためには、改めて詳細な調査を行う必要がある。

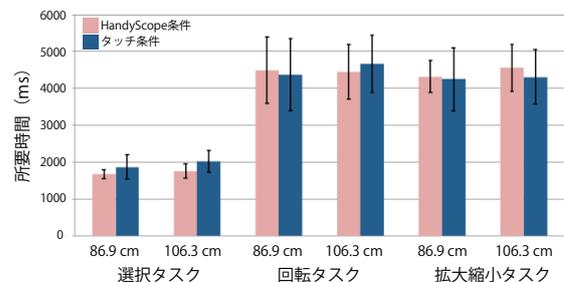


図12. 各距離毎の1試行の所要時間

4.7 アンケート結果とその考察

それぞれのタスクにおける、HandyScopeとタッチ操作のどちらの手法が好みであったかのアンケート結果を図13に示す。

選択タスクにおいては全ての被験者がHandyScopeが好みであると答えた。また、拡大縮小タスクにおいては8名の被験者がHandyScopeが好みであると答えた。この内全ての被験者がその理由として、タッチ操作ではオブジェクトに手が届く位置まで移動する必要があるが、HandyScopeはその場にて操作を行うことができるためと答えた。

回転タスクにおいては好みが別れた。2名の被験者は回転操作に両手を用いることができるため、タッ

チ操作のほうが好みであると答えた。また、別の2名の被験者は base-fingers を常にテーブルトップ上に接触させることが大変であったと答えた。別の被験者1名は試行の失敗時における HandyScope の再起動は手間であると答えた。

拡大縮小タスクにおいてタッチ操作が好みと答えた被験者2名は、その理由として常に base-fingers をテーブルトップ上に接触させることが大変であったと答えた。

そのため、今後は base-fingers をタッチパネル上から離しても遠隔地の操作を続けられるように、サークルを固定させる機能を実装する。

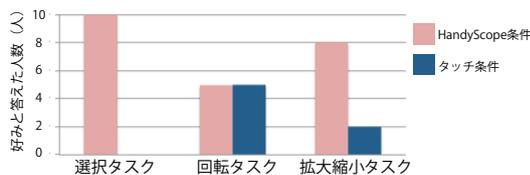


図 13. 手法の好みのアンケート結果

5 議論：引き出し方向

引き出しジェスチャは図 14 に示すように 2 種類の引き出し方向が考えられる。そこでどちらの引き出し方向が HandyScope に適しているかを調べるために、4.3 節に挙げた選択タスクを著者らが行った。その結果、図 14b の引き出し方向による操作が図 14a の引き出し方向による操作よりも所要時間が短くなった。所要時間に違いが生じた理由として、図 14a の引き出し方向では handler と scope 間の視線の移動距離が大きくなり、ポインティングが困難になることが考えられる。このため、HandyScope の引き出し方向を図 14b の方向に決定した。今後は被験者実験を行い、この妥当性を評価する予定である。

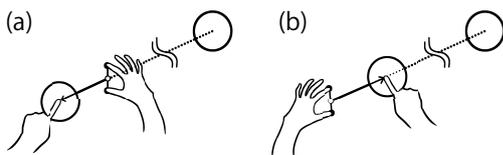


図 14. 引き出し方向

6 まとめ

我々は新たな遠隔地操作手法として HandyScope を示した。HandyScope を用いることにより、ユーザは 2 種類の円形のウィジェットを用いて遠隔地を素早くポインティングできる。ポインティング後は、遠隔地の直接操作や遠隔地と手元間におけるオブジェクトの移動が可能である。

評価実験の結果、遠隔地を選択する場合 HandyScope は有用であること及び、オブジェクトまでの距離が遠くなると HandyScope が有用な手法となることが確認できた。アンケートからは HandyScope がユーザに好まれることを確認した。今後は、base-fingers をテーブルトップから離しても遠隔地の操作

を続けられるように、サークルを固定させる機能を実装する。また、CD 比を動的に変更できることの詳細な評価や、オブジェクトの移動について評価を行う。さらに、複数のディスプレイを連携させた際の本手法の有用性についても調査していきたい。

参考文献

- [1] M. Abednego, J.-H. Lee, W. Moon, and J.-H. Park. I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. ITS '09, pp. 61–64.
- [2] A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. ITS '11, pp. 11–20.
- [3] N. Banovic, F. C. Y. Li, D. Dearman, K. Yatani, and K. N. Truong. Design of unimanual multi-finger pie menu interaction. ITS '11, pp. 120–129.
- [4] T. Bartindale, C. Harrison, P. Olivier, and S. E. Hudson. SurfaceMouse: supplementing multi-touch interaction with a virtual mouse. TEI '11, pp. 293–296.
- [5] G. Furumi, D. Sakamoto, and T. Igarashi. SnapRail: a tabletop user interface widget for addressing occlusion by physical objects. ITS '12, pp. 193–196.
- [6] A. Khan, G. Fitzmaurice, D. Almeida, N. Burtynk, and G. Kurtenbach. A remote control interface for large displays. UIST '04, pp. 127–136.
- [7] S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. UIST '05, pp. 43–52.
- [8] J. Matejka, T. Grossman, J. Lo, and G. Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. CHI '09, pp. 1073–1082.
- [9] J. K. Parker, R. L. Mandryk, and K. M. Inkpen. TractorBeam: seamless integration of local and remote pointing for tabletop displays. GI '05, pp. 33–40.
- [10] Y. Tokoro, T. Terada, and M. Tsukamoto. A pointing method using two accelerometers for wearable computing. SAC '09, pp. 136–141.
- [11] A. Toney and B. H. Thomas. Applying reach in direct manipulation user interfaces. OZCHI '06, pp. 393–396.
- [12] S. Voelker, M. Weiss, C. Wacharamanotham, and J. Borchers. Dynamic portals: a lightweight metaphor for fast object transfer on interactive surfaces. ITS '11, pp. 158–161.
- [13] T. Yoshikawa, B. Shizuki, and J. Tanaka. HandyWidgets: local widgets pulled-out from hands. ITS '12, pp. 197–200.
- [14] H. Zhang, X.-D. Yang, B. Ens, H.-N. Liang, P. Boulanger, and P. Irani. See me, see you: a lightweight method for discriminating user touches on tabletop displays. CHI '12, pp. 2327–2336.

手法を実現する。また、ライフログ写真を長期記憶化するための受動的閲覧手法を実現する。さらに、長期的な利用からライフログ写真における人間関係がどの程度利用可能かといったことについて議論を行う。

2 写真ライフログ

ライフログにデジタル写真を使うメリットは、静止画であるため一覧化しやすく、現在発売されているデバイスを利用するだけで手軽にライフログを実践することが可能であることなどである。ここで、写真ベースのライフログは、主観的なライフログ（主観写真ライフログ）と客観的なライフログ（客観写真ライフログ）に大別される [2]。主観写真ライフログとは、撮影時においてユーザの明確な意図があり、ユーザのタイミングと視点で空間を切りとるものである。

客観写真ライフログとは、システムにより自動記録されるものであり、SenseCam などウェアラブルカメラを利用した自動記録がそれに相当する。また、ユビキタスホームなどでのユーザの記録も客観的なライフログといえる。SenseCam は首から下げて利用する客観写真ライフログ用のカメラであり、1分間に2回または、センサ（照度、地磁気、加速度センサなど）で取得した外界の状況に応じて自動撮影されるようになっている。客観写真ライフログは自動撮影されるため、自然に撮影できることや、何らかの作業中に撮影できることなどメリットがある。一方、撮影するだけではユーザの視点が抜け落ちるため、後に活用しようと思った際に何をキーとして探したらよいかわからないなどの問題がある。

主観写真ライフログと客観写真ライフログの特性の違いについて詳細に分析した研究は存在しないが、Gurrin らは SenseCam を用いた客観写真ライフログによる1年間の記録（100万枚）から無作為に1000枚を抽出および分析している [5]。その結果、客観写真ライフログには腕や手がよく写っており、作事中（24.6%）や運転中（16.2%）、会話中（13.3%）などの写真が多く、人を自然に記録できている。また、インドアの写真が73.4%と多く、風景写真は1.1%とかなり少ないことを報告している。さらに、29.9%の写真に人が写っていることを報告している。一方、著者の主観写真ライフログの中から1000枚を無作為に抽出して調査したところ、作業中の写真はほとんどないが、作業の後などに撮影したものや、他人が作業している様子や、他の人が会話している様子を撮影していたものが多かった。また、65.6%がインドア、34.4%がアウトドアの写真であり13.2%が風景写真であった。さらに、44.0%の写真に人が写っていた。単純な比較はできないが、主観、客観ともに人が写っている傾向が高いといえる。

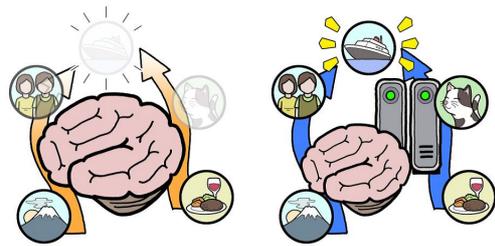


図 2. ライフログ（外部記憶）により記憶拡張

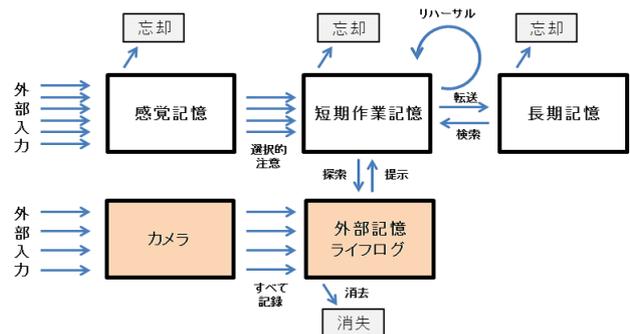


図 3. ライフログによる記憶のモデル

ここで、基本的に人間は画像を膨大かつ長期的に記憶することが可能である [7] ものの、Isola らの写真の覚えやすさに関する研究 [6] によると、写真のタイプごとに記憶の容易性が異なり、人を含んだ写真を最も覚えやすく、次に人の大きさのオブジェクトが被写体となっている写真を覚えやすいのに対し、風景のような写真は綺麗であっても覚えにくい事がわかっている。つまり、写真ライフログでは、時空間をベースとしつつも、人間関係を効果的に利用した探索手法および、人を考慮した長期記憶化手法が有効になるのではと考えられる。

そこで、本稿ではこのような人間関係を中心としたライフログ探索を可能とする仕組みを実現する。ここでは、ライフログ写真から人間関係を抽出および可視化し、対話的に探索可能とする。

3 ライフログの外部記憶化手法

先述の通り、短期作業記憶はすぐに失われてしまうものであり、長期記憶への定着も難しい。ここで、主観写真ライフログを記憶として使う（図2）と、図3のように消去しない限り短期作業記憶へ何度も呼び出すことが可能となる。そこで、本稿ではライフログを外部記憶として扱うために、短期作業記憶へと呼び出すための探索手法と、何度もユーザに提示し、リハーサルを行えるようにすることで長期記憶に定着させるための受動的閲覧手法を提案する。

3.1 人間関係を利用した探索

デジタル写真を撮影すると、撮影日時やカメラの機種名、各種撮影情報などが EXIF 領域に自動記録される。また、最近では GPS 搭載のデジタルカメラも増えており、撮影場所情報も自動的に記録されるようになってきている。さらに、Google Picasa や Apple iPhoto などのように、写真ライブラリ内の写真に対する顔画像認識を行うソフトウェアも近年広がりを見せている。我々は、こうした撮影日時、撮影場所、被写体情報を写真とともに取り込み、データベースに格納し、時間、空間、人間での検索を可能とする。

しかし、人の記憶は曖昧であり、数ヶ月、数年前のこととなると、目標とするライフログ写真をいつ撮影したのか、どこで撮影したのか、誰と撮影したのかといった事も正確に覚えていない。つまり、目標とするライフログ写真への到達も一筋縄ではいかない。ライフログ写真の検索および探索においては、それを思い出すきっかけとなる柔軟な絞り込みと周辺情報提示が必要となる。

ここで、ライフログ写真として蓄積されているものは、ユーザ（撮影者）が見て切り取ったものであり、それぞれの写真に写っている人物を認識すると、ユーザからみた人物関係（主観的視点におけるソーシャルネットワーク）が見えてくる。つまり、人を1つのノードとし、同一の写真に写っている人には関係があるとみなしてその人物間（ノード間）にエッジをはり、そのエッジがどのようにはられているかということを利用してクラスタリングを行うことで、コミュニティを検出することが可能となる。そこで、本稿ではある時空間での写真集合からすべての人間関係を抽出し、その人間関係を Newman 法 [3] によってハードクラスタリングした結果とともに提示することにより、人間関係を考慮した探索を可能とする。

なお、ハードクラスタリングした結果の人間関係をバネモデルを用いて提示し、提示されている人間関係をもとに検索する手法を実現する。ここでバネモデルにおけるノード間の引力はユーザ間の親密性の高さに基づき、強さを変更している。また、人と人との間の親密性については、「同じ写真に共起していること」「同じ時分に共起していること」「同じ時間に共起していること」「同じ日に共起していること」という4パターンを用意し、その共起頻度によって計算している。

3.2 受動的閲覧による長期記憶への格納

近年、横長の液晶ディスプレイが広く普及している。横長のディスプレイは、動画を視聴する場合や、ゲームをする場合などには効果的であるが、作業を行なっている場合には、2つのウィンドウを表示するには少し狭すぎ、1つのウィンドウを表示する

は広すぎる。ここで、1つのウィンドウのみを表示するのであれば、画面の両端に余裕がある。

受動的閲覧による、外部記憶からの長期記憶化では、こうした画面の両端のスペースを有効活用し、利用者とライフログ写真との遭遇を促進し、その事自体が短期記憶のリハーサルとして働くようにすることによって、長期記憶化を狙うものである。受動的閲覧において、システムは特に人間にフォーカスを当て、任意の人間を取り出し、その人間を中心とした写真（その人間と一緒に居た場所や、居た時間を利用）中心とした n 件のライフログ写真をデータベースから取り出し、1枚ずつ写真を上から下にスクロールさせながらアニメーション提示するものである。

4 実装

提案システムを Microsoft Visual C++ を利用して、データベースに SQLite を利用して実装した。また、サムネイル画像を必要に応じて自動生成することにより、20万枚以上でも問題なく動作する仕組みを実現している。さらに、データベースやサムネイルはバックグラウンドで自動的に構築および生成されるため、初めて利用する際にも特に負荷なく利用可能である。システムは LifelogViewer という名前で Web 公開している¹。

図4は、カレンダーインタフェース、地図インタフェース、人間関係ビューを利用して時間および空間を指定しながら目的とする食事のライフログ写真を探索している様子である。カレンダーインタフェースで表示月を変更している際に提示される人間関係ビューから、まずどの月かを限定し、次に地図インタフェースを用いてズームし表示領域を限定することで提示される人間関係のクラスタを限定している。さらに、そこから目的とする人が含まれるクラスタを指定して検索し、最終的に目的の写真にたどり着くことができている。図5は、ライフログの受動的閲覧手法を利用して、作業中にライフログを提示している様子である。

5 考察

時間、空間、人間関係を考慮して20万枚の写真ライフログを扱える環境は限られている。Apple の iPhoto は1操作辺りに数十秒かかるため利用できず、Google の Picasa も頻繁に固まってしまうという問題が生じた。速度面は無視して、著者がこれまでに必要とした様々なライフログ探索について提案システムと Picasa の比較を行った。大きな違いは、Picasa ではユーザは明確に場所や時間、被写体を選べる場合は効果的に目標とするライフログ写真を探しだすことができるものの、少しでも記憶が曖昧な

¹ <http://calendar2.org/>

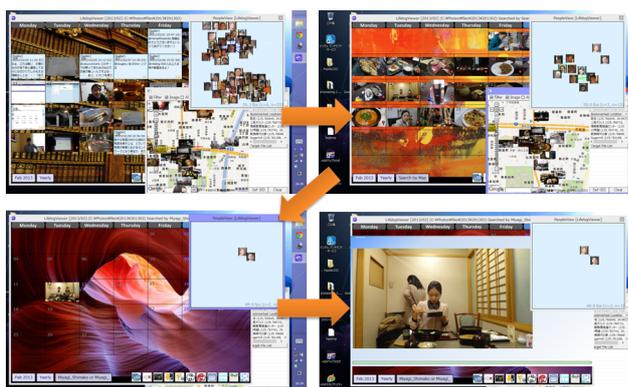


図 4. 時空間と人間関係を利用した写真探索



図 5. ライフログの受動的閲覧 (画面右端)

場合には探すことが困難であるという点である。一方、提案システムでは、ユーザの記憶が曖昧な場合であっても対象となる写真の発見が可能であり、検索達成率は高かった。

本手法によってある特定の月の人間関係がどうなるかを示したものが図6である。上段は、私が京都大学から明治大学へと転職前(2013/02)と、転職後(2013/04)で大きな違いが現れており、2013/04では転職先の大学での学生が登場し密なネットワークを作っている事がわかる。また、2013/03では、転職元や転職先の方々、京都の友人の方々などの交流があったため、様々な広がりのあるコミュニティができています。下段は、WISSが開催される12月について、2010、2011、2012年を比較したものである。2010は右上に、2011は左下に、2012は下にそれぞれWISSのコミュニティが形作られている。また、2010年はWISSのコミュニティは独立しているが、2011、2012年については他のコミュニティとの関係が深まっていることが分かる。これは、2011年、2012年の12月にそれぞれ開かれたニコニコ学会にWISS関係者が関与しているためである。

クラスタリングにおいて、知人と解釈する度合いを「同じ写真での共起」「同じ時分での共起」「同じ時間での共起」「同じ日での共起」と4つ用意し比較した。様々な時空間でのクラスタリング精度について実際に運用することで比較してみたところ、2005年までは「同じ日での共起」、2006年～2008年に



図 6. 人間関係の変化

については「同じ時間での共起」、2009年以降については「同じ時分での共起」と「同じ写真での共起」が良い結果となっていた。これは、撮影頻度と、被写体として人間を選ぶ度合いとに深く依存しているのではないかと考えられる。実際、2005年までは撮影頻度は高くなかったが、2006年から2008年については撮影頻度があがり、2009年以降も撮影頻度はこれまでと大きく変化していないものの、著者の撮影スタイルが変化していき人が被写体となっている量が増加したことも理由の1つとして考えられる。撮影頻度や撮影場所によってイベントを分離することが可能になれば、本問題は解決する可能性を秘めている。

受動的閲覧が作業の邪魔になるかどうかを調べるため、1年以上の長期的な利用を実践した。その結果として、受動的閲覧の仕組みは、作業中はほとんど目に入らず、ふと気が緩んだ時に目に入る程度であり、著者自身は問題がなかった。むしろ、集中して作業をし、気を抜いた時に、知人の楽しい写真や、娘や妻などの写真が目に入り、休憩時の癒やしとなっていた。また、受動的閲覧で提示される写真の中のグラフなどから着想を得て、講義資料に取り込んだこともあった。以上のことより、長期記憶に定着させるための方法として有効であったかは不明であるが、少なくとも受動的閲覧は有効であった。

この1年間でライフログが役に立った事例は多数あるが、そこから選別したものは下記の通りである。

- 出産育児一時金請求書を記入する際に医療施設の名称&住所が必要となった。妻と沖縄で一緒だった時に出生届を書いた事を思い出し、地図UIを開いて「沖縄」を指定。表示される人間関係UIから妻の顔をダブルクリックし、カレンダーUIに提示された写真の中から妻がベッドに座っている写真を発見。そういえばこの時だったと、その日の写真を一覧化し出生届&出生証明の写真とその情報を発見。

- 転居先の水道代自動引落しの手続き時に、銀行への届け出サインが必要に、銀行での契約時以降サインを使った覚えがないため日本語か英語か記憶が無い。写真のGPS情報がずれているのか契約した場所で検索しても写真が見つからない。口座を使い始めたのは入籍時である事を思い出し、入籍日あたりを探索。契約後に、妻と飲みに行ったお店の店長が人間関係ビューに提示され、ふとその事を思い出して周辺を探索。書類を発見し、日本語だったことを確認して記入。
- 耐震工事のためのオフィスの一時的な引っ越しをしていて、1年後に戻ることになった時、「前のオフィスに時計ってありましたっけ?」「あったとしたらどんなのでしたっけ?」という話になり、数名の同僚と一緒に時計が写っている写真を受動的閲覧で過去に見かけたことを思い出し、場所を人間関係UIを提示しながら過去に遡ることによって、木の枠のこんなダサイ感じの時計であったことを発見。

他にも、契約解除時にすでに忘れてしまったIDを調べたり、田舎で運行本数が極端に少ないバス停でカメラに運行表を記録させ後で活用したり、著者の妻が著者の友人に会った時にどのような服を着ていたかを探したりなど、その活用方法は様々である。

こうした探索は、時空間や人間関係があったからこそ可能となっているものが多い。また、受動的閲覧によって、何度も任意の人が被写体となっている写真と、その人の周辺の写真（一緒に作業した記録や、一緒に摂った食事、その前後のスケジュールに関するものなど）が連続的に提示されているため、自然と長期記憶に定着し、その定着した「人」と「周辺の写真」から目的とする写真を探索可能となっているケースも多い。

一方、長期的な利用から得られた問題としては、まず忘れることができないという事が大きな理由としてあげられる。例えば、ある時まで恋愛関係にあった人との別れは、その別れの前までは幸せな記録として残っていたのに、別れの後では辛いものとしてただ存在するだけである。ライフログ写真の受動的閲覧では、こういった辛い過去もどんどん提示されてしまう。こういった、忘れることができない何かをどうするかというのは、今後の大きな課題の1つである。

また、美術館内や映画館内など、撮影できない場所は多数ある。こういった状況において、本人としては記録した覚えがあるのだけれど、実際には撮影および記録しておらず、探索しても見つからないということがある。そうした、存在しないライフログ（見つからない記録）をどう提示するかというのは課題の1つである。

ライフログを長く続けていくと、ライフログが無いと困るというライフログ依存症をといえるのではと思えることが多々ある。例えば、すぐにデジタルカメラにて記録することができるため、無理覚える必要がなく短期作業記憶が劣化しているように感じることや、受動的閲覧で何度でも見るため、ある出来事が遠い過去のはずなのにいつ最近であるかのように感じる事などである。こういった問題は、今後ライフログが世の中に広まると、社会問題となるのではと考えている。

先日、著者のパソコンのSSDが突如読み込めなくなってしまう、最後にバックアップを取った日から、20日間分のライフログデータが失われてしまった。このようなデータの損失は、現代の記憶喪失といえるかもしれない。こういった戻せない記録をどう保管するか、また補完するかといったことも大きな課題となるのではないだろうか。

6 関連研究

SmartWrite/SmartCalendar[9]は、写真・メモベースのライフログであり、時空間をベースとした写真の探索を可能としているが人間関係は考慮されていない。我々の手法は時間のみならず、空間や人間関係を考慮した閲覧および検索を可能としており、対象の柔軟な探索が可能である。

五味らは、時間、空間、人間関係の3者を利用したライフログ写真の閲覧手法を提案している[10]。しかし、五味らの手法では、あらかじめ時間ベースで全ての写真のクラスタリングを行なってイベントを抽出し、イベントにおける共起度をもとに人物のクラスタリングを行なっているため、任意の時間や場所を指定した動的なクラスタリングは行えない。

捧ら[8]も、時間、空間、人間関係を利用してライフログ写真を閲覧する手法を提案している。ここでは、写真における共起性とその前後の写真から、人物同士の親密度を計算し、クラスタリングを行うというものである。ただし任意の時空間を対象とはしていないうえ、計算コストが高く、リアルタイムの分析には不適である。

Facebook, mixiなどのSNSの友人関係からソーシャルグラフを作る試みは多数研究されている[4]。しかし、こうした研究では、グラフは明示的なリンク（フォロー、友達として登録など）を元としている。一方、ライフログの中には明確な友人関係に関する情報は存在していない。また、こうした人間関係は日々変化するものであり、関係が強まったり（他機関の知人から同僚へ、赤の他人から人生の伴侶へなど）、関係が弱まったり（交際相手からただの知人へ、クラスメートから他大学の友人へなど）する。我々は、ライフログ写真の中から2者間の関係性の強さを計算し、人間関係の変化を考慮しつつ、探索を可能とする手法を実現するものである。

7 まとめ

本稿では、主観写真ライフログの能動的探索および受動的閲覧手法を提案および実装し、その有効性について検討を行った。主観的な評価にすぎないが、長期にわたって継続利用しており、今まで到達できなかった探索を可能となり、主観写真ライフログの記憶拡張としての可能性を明らかにした。

システムは Web 公開および雑誌などで紹介されており、多数のダウンロードがある。今後は、主観写真ライフログを長期にわたって継続しているユーザーに依頼し、システムの継続的な評価を行う予定である。

謝辞

本研究は文部科学省科学研究費補助金若手研究(A) (研究代表者: 中村聡史, #23680006) の助成を受けたものです。

参考文献

- [1] Atkinson, R.C., Shiffrin, R.M: Human memory: A proposed system and its control processes, The psychology of learning and motivation (Volume 2), pp.89-195 (1968).
- [2] 中村聡史: 身近になったライフログ ~パソコンに眠る数万枚の写真をどう活用すればよいか~, 情報処理 54(2), pp.142-149 (2013).
- [3] M.E.J. Newman: Fast algorithm for detecting community structure in networks. In Physical Review. E 69 (2004).
- [4] Ken Wakita, Toshiyuki Tsurumi: Finding community structure in megascale social networks, Proceedings of the 16th international conference on World Wide Web, pp. 1275-1276 (2007).
- [5] Cathal Gurrin, Alan F. Smeaton, Daragh Byrne, Neil O'Hare, Gareth J. F. Jones, Noel O'Connor: An examination of a large visual lifelog, Proc. of AIRS'08, pp. 537-542 (2008).
- [6] Isola, P., Xiao, J., Torralba, A., and Oliva, A: What makes an image memorable? Proc. of the 24rd IEEE Conference on Computer Vision and Pattern Recognition, pp. 145-152 (2011).
- [7] Lionel Standing: Learning 10,000 Pictures, Quarterly Journal of Experimental Psychology, vol.25, pp.207-222 (1973).
- [8] 捧隆二, 佃洗撰, 中村聡史, 田中克己: 時間・空間・人物情報に基づくインタラクションによるライフログ画像の探索手法の提案, DEIM Forum 2012 D9-4.
- [9] 美崎薫: SmartWrite/SmartCalendar 手軽に書けるメモとメモと写真を見続けるカレンダー環境の提案, 情報処理学会ヒューマンインタフェース研究会報告 2005(71), 71-76 (2005).
- [10] 五味愛, 伊藤貴之: 「何時, 何処で, 誰と」 3つのメタ情報に基づく個人写真ブラウザ, 芸術科学会論文誌, Vol. 10, No. 1, pp. 36-47 (2011).

未来ビジョン

人の記憶は曖昧なものであり、過去に見聞きしたものを思い出せないことは多い。忘れることは必ずしも悪ではなく、痛ましい出来事や、友人知人との別れ、恐怖を覚えた出来事や、昔の彼女との辛い思い出など、思い出したくないことも多々あるであろう。しかし、思い出せないことによって失敗をしてしまうことも少なくない。

今後、技術の進歩により、ウェアラブルコンピューティングが今の携帯電話のように人々にとってありふれた普通のものとなり、また生活の中での視線検出技術の進歩や脳波測定技術の進歩などにより、主観的なライフログ写真の記録がより簡単になり、ライフログ写真のインデックスもより深いものになると期待される。将来、見聞きしたものを全て外部記憶に記録

し、記録したものを検索などにより取り出すことも可能となるであろう。

つまり、忘れることができなくなる時代が到来することになる。もちろん、そうしたデバイスを使わないという選択しはありえるだろう。しかし、記憶拡張された人と記憶力という面で圧倒的に差が広がってしまい、嫌々ながらもそうしたデバイスを使うユーザーが増えるのではと考えられる。

本研究はこうした忘れることができなくなることは、どういったメリットがあり、どういったデメリットがあるのかということ、実践的に取り組むものである。また、人の様々な記憶の仕組みとの関係を考えていくことにより、ライフログによる記憶拡張がどのように記憶を変えていくのか、社会を変革させていくのかについて考えるものである。

EpisoPass: エピソード記憶にもとづくパスワード管理

増井 俊之*

概要. 忘れる可能性が低いエピソード記憶にもとづく秘密の質問を使って強力なパスワードを生成/管理するシステム「EpisoPass」を提案する。EpisoPassは、ユーザが作成した秘密の質問への回答にもとづいてシード文字列を換字することによってパスワードを生成する。シード文字列や回答のバリエーションにより異なるパスワードが生成されるので様々なサービスに対して異なるパスワードを生成できることに加え、シード文字列を逆計算することにより既存のパスワードの管理もできる。適切な運用により、パスワードに関連するあらゆる情報を秘密にすることなく強力なパスワードの生成/管理が可能である。

1 はじめに

個人認証のためにパスワードが現在広く利用されている。パスワード認証には多くの問題があることが知られているが [21], 今後も長期にわたって利用され続けると予想されるため [8], 問題点を認識しつつ適切に運用するための工夫が必要である。

パスワードの長期的記憶が難しいことはパスワード認証の大きな問題点のひとつである。安全に運用するためにはパスワードはランダムで長い文字列であることが望ましいが、そのようなものを頭の中に記憶しておくことは難しい。また複数のサービスを利用する場合、サービスごとに異なるパスワードを利用することが望ましいが、すべてのパスワードを記憶しておくことはほとんど不可能である。Florêncioの2007年の大規模な調査によれば、ユーザは平均25個のサイトで6.5個のパスワードを利用しており、3ヶ月間にユーザの4.28%がパスワードを忘れていた [5]。また2011年の野村総研の調査によれば、一般的なユーザがパスワード認証を行なうサイトは平均19.4個で、利用しているパスワードは平均3.1個であった [23]。多数のパスワードを記憶することが困難であるため、多くのユーザが同じパスワードを複数サイトで使い回しているのだと思われる。

異なるパスワードをすべて記憶することは不可能なのでどこかに記録しておく必要があるが、パスワード文字列をそのまま記録するのは危険なので、複数のパスワードを秘密情報として扱うためのパスワード管理システムが利用されている。パスワード管理システムはひとつの「マスターパスワード」を利用して他のすべてのパスワードを管理するもので、暗号化されたデータベースにパスワードを格納するもの [1][3][12] が多いが、サービス名をもとにマスターパスワードを変換することによって複数のパスワードを生成するシステム [18] もある。両者ともに

マスターパスワードの記憶が必須であり、マスターパスワードを盗まれたり忘れたりする危険が常に存在する。

一般にユーザはパスワードを忘れがちであるため、多くのサービスにおいてパスワードを復元したり初期化したりする手段が用意されている。ユーザが秘密の質問に対する答を登録し、質問に正しく回答することによってパスワードを復元したりリセットできるサービスは多いし、秘密の質問に答えることによってパスワード管理システムのマスターパスワードを復元するシステム [25] も提案されている。

新しく覚えた情報や新しく考えた情報はどうしても忘れてしまう可能性があるため、新しく作成したパスワード文字列を記憶して認証に利用することは本質的に無理がある。一方、既知で忘れることがないエピソード記憶を秘密の質問として認証のために直接利用することができれば、認証に必要な情報を忘れてしまうことがないはずである。多くの画像認証システム [2][19] は秘密の質問に対して適切な操作を行なうことによって認証を行なっているためパスワードのような特種な情報を記憶する必要がない。画像認証システムはまだ普及しておらず、利用できる環境は限られているが、忘れないエピソード記憶を利用した秘密の質問への回答を強力なパスワードに変換するシステムがあれば、通常のパスワード認証を用いた現在の様々なサービス上でも、認証方法を忘れる心配なく安全に認証を行なうことができるようになる。本論文ではこのようなシステム「EpisoPass」について述べる。

2 EpisoPass

2.1 EpisoPassの原理

EpisoPassは、ユーザが忘れることがない個人的なエピソード記憶を文字列に変換することによって安全なパスワードを生成するシステムである。パスワード文字列は以下の手順で生成される。

Copyright is held by the author(s).

* Toshiyuki Masui, 慶應義塾大学 環境情報学部

1. パスワード生成の「種」となる文字列を用意する。以下ではこれを「シード文字列」と表現する。
2. 忘れることがない個人的なエピソード記憶にもとづく秘密の質問を複数作成し、それぞれについてひとつの正答と複数の偽答を用意する。
3. 質問と回答の組にもとづいてシード文字列に換字操作を行なう。すべてに正しく回答したとき生成される文字列をパスワードとして利用する。

2.2 EpisoPass 利用例

2.2.1 ブラウザでの利用

筆者が twitter のパスワードを生成するためにブラウザで EpisoPass を利用している例を図 1 に示す¹。シード文字列として「Twitter123456」という文字列を指定しており、4 個の秘密の質問に対する回答選択に応じて「Mfveabn574923」のようなパスワード候補が生成される。異なる答を選択したり異なるシードを指定すると全く異なる文字列が生成される。シード文字列の 8 文字目が数字である場合はパスワードの 8 文字目も数字になるなど、シード文字列の文字種に対応したパスワード候補が生成される。パスワードとして大文字/小文字英数字と記号をすべて利用しなければならないサービスの場合はシード文字列に「PassWord123!@#」のような文字列を指定すればよい。

最初の秘密の質問は筆者の小学校の同級生に関するもので、最後の質問は数年前の体験に関するものである。これらの質問は古いエピソード記憶にもとづいており、筆者が将来答を忘れることはほとんど考えられないが、本人以外がこのような質問に答えることは難しいので正しいパスワードを得ることはできない。

秘密の質問と答はブラウザで編集でき、右上の「サーバにセーブ」ボタンを押すことによりシード文字列、秘密の問題、答のリストがサーバにセーブされる。「ファイルにセーブ」ボタンを押すと JSON データをパソコンにダウンロードでき、パソコン上の JSON データをブラウザにドラッグ&ドロップするとサーバにアップロードできる。ユーザはどれが正答かを指定するわけではないので問題データを見てもユーザのパスワードはわからない。

シード文字列を「Facebook123456」に変更すると、生成されるパスワードは図 2 のように変化する。このように、サービスごとに異なるシード文字列を利用することによって様々なパスワードを簡単に生成できる。

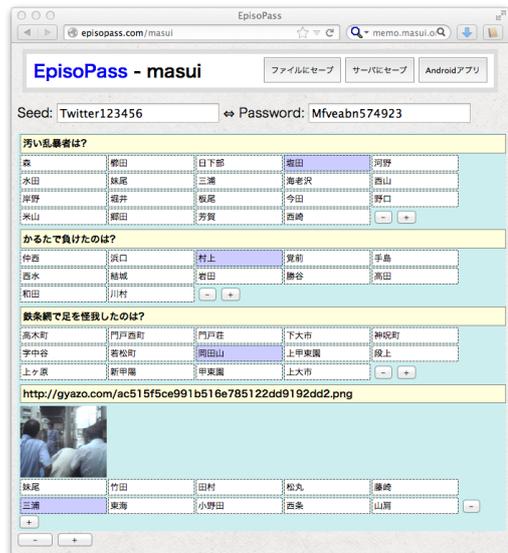


図 1. ブラウザ上で Twitter のパスワードを生成。



図 2. FaceBook のパスワードを計算。

2.2.2 既存パスワードの利用

現在「Masui1234」のようなパスワードを利用している場合、図 3 のようにパスワード欄に現在のパスワードを入力すればそれを生成するシード文字列が自動生成されるので、生成されたシード文字列を記録しておけばよい。



図 3. 既存のパスワードを利用。

既存のパスワード管理システムは、利用中のパスワードを記憶するもの [1][3][12] と新しいパスワードを生成するもの [18] に分類されるが、EpisoPass はこの両方をサポートしている。

¹ <http://EpisoPass.com/masui>

2.2.3 Android アプリ

Web サービスを利用する場合、ブラウザとサーバとの間の通信を記録されたり盗み見されたりされる心配を完全に払拭することはできない。前述の例において、パスワードはブラウザ内部でJavaScriptにより生成されるので、一度ページを表示した後はネットワークを遮断してもパスワード計算を行なえるようになっているが、最初から全く通信を行わずにパスワードを作成できる方がより安心であろう。このため、通信を全く行わずにマシン単体でパスワード計算を行なうための Android アプリを用意した。ページの右上の「Android アプリ」ボタンを押すと、現在表示している秘密の問題と答を内蔵した Android アプリがサーバ上でビルドされてダウンロードされる。

シード文字列を設定して Android アプリを実行すると図4のように質問がひとつずつ表示され、ボタンを押してすべて回答するとパスワードが計算され表示される。

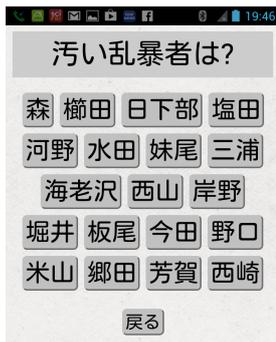


図 4. Android アプリ。

回答入力とパスワード計算は Android 端末で実行されるため、端末を「機内モード」に設定するなどの方法でネットワーク接続を遮断した状態でもパスワードを計算することができる。EpisoPass をインストールした Android 端末を持っていれば常に各種のパスワードを計算できるので、他人のマシンや公共の場所に設置されたパソコンなどでも容易にネットサービスを利用することができる。

前述の方法で EpisoPass アプリをサーバからダウンロードする場合は、ブラウザ上で秘密の問題をサーバに登録する必要があるが、秘密の問題を全くネット上に露出することなくアプリを利用することもできる。秘密の問題を含まない EpisoPass アプリを Google Play で公開²しているのので、これを端末にインストールした後、ローカルマシンで作成した秘密の問題を端末に転送すれば EpisoPass.com からダウンロードしたアプリと同様に利用できる。こ

² https://play.google.com/store/apps/details?id=com.pitecan.episo_pass

の手法を使うと秘密の質問が通信路を通ることがないので安全であるが、アプリのセットアップの手間は増える。

2.3 パスワード文字列の計算方法

問題と回答から文字列を生成し、その MD5 値によってシード文字列を換字することによりパスワードを生成している。パスワード文字列の計算方法は附録に示す。

3 議論

3.1 運用形態と安全性

パスワード管理システムにおいて最も重要なのは安全性である。EpisoPass は様々な運用方法が可能であり、運用形態によって安全性の評価が異なるので、利用例にもとづいて EpisoPass の安全性を考える。

3.1.1 EpisoPass の利用を秘密にする場合

パスワード管理に EpisoPass を利用していることを公開せず、単なるパスワード生成機として利用する場合は、普通にパスワードを利用する場合に比べて安全度が低下することは無い。パスワードとして充分強力な文字列をシード文字列として利用すれば EpisoPass によって換字された文字列もパスワードとして強力だと考えられるし、推測しやすい文字列をシード文字列として設定した場合であってもランダムな換字操作によってより強力なパスワードが生成されるので、EpisoPass を利用するデメリットは無く、通常のパスワード管理システムと同様の方法で利用できる。

3.1.2 シード文字列を秘密にする場合

EpisoPass の秘密の質問を公開した場合でも、シード文字列を通常のパスワードと同じレベルで秘密にしておけば SuperGenPass[18] と同じレベルの利便性と安全性が確保できる。

3.1.3 すべて公開する場合

秘密の質問を解くことが不可能であれば、シード文字列と秘密の質問をすべて公開しても安全である。この場合、ユーザは秘密の質問とシード文字列を通常のテキストデータと同じように管理できるし、パスワード管理のために新しく記憶しなければならない情報が皆無なので、ユーザはパスワード管理について注意を払う必要がなくなり気が楽になる。

秘密の質問を解くことを困難にするためには質問の数と偽答の数が多くなければならないし、自分だけが知っているエピソード記憶をうまく秘密の質問にするにはコツが必要である。良い秘密の質問を作る方法に関しては 3.3 で議論するが、すべての質問を公開するのが心配な場合や、秘密の質問の数や質

かが充分でないと感じられる場合は3.1.1や3.1.2の手法で運用し、徐々に運用方法を変えていけば良いだろう。

3.2 秘密の質問の強度

パスワードは長年利用されているため強度や実際の運用に関して多くの研究が存在するが[7][11], 秘密の質問の強度に関しては十分な研究が行なわれていない。EpisoPassの運用実績は長くないが安全性などについて考察を行なう。

EpisoPassで選択枝が10個の秘密の質問を8個使用する場合、総当たりでパスワードを生成するには1億(10^8)通りの試行が必要であり、エントロピーは26.6ビットとなる。英字からランダムに8文字を並べて作成したパスワードのエントロピーは37.6ビットになるが、“pmvixuzq”のように全く意味のないパスワードを記憶して利用することは少ないため、実際に利用されるパスワードのエントロピーは20ビット程度と考えられているので[14], 秘密の質問と選択枝の数を10個程度用意すれば通常のパスワードと同程度の強度が期待できることになる。総当たり攻撃が可能なオフライン運用ではエントロピーの大きさは重要であるが、オンラインサービスではパスワード入力を何度か間違えるとサービスがブロックされるのが普通なので、それほど長いパスワードを用意する必要は無いと考えられている[6]。

一方、秘密の質問を利用する認証の脆弱性を利用した攻撃が近年問題になっている。パスワードを忘れたときのために、あらかじめ設定した秘密の質問に答えることによってパスワードをリセットできるサービスがあり、「母親の旧姓は?」や「最初に飼ったペットの名前は?」のような質問に対してユーザが答を登録するようになっている。このような問題は他人が調べたり推測したりすることが容易であるうえに秘密の質問の数は一般的に少なく、パスワードよりも脆弱だといえる[15]。ユーザが作成した秘密の質問を使えばこのような問題はなくなるはずであるが、他人に解かれにくい問題をユーザが作成することは難しく、またユーザ自身が答を忘れてしまうことも多いと考えられている[10][16]。

EpisoPassでは、他人には解くことが難しく自分では忘れないような秘密の質問を自由にいくつでも利用できるようになっている。問題作成に慣れていないユーザには有効な秘密の質問を作成することは難しいかもしれないが、次節で述べるように、適切な質問を選ぶことによりこの問題を解決できるはずである。

3.3 秘密の質問の選択

EpisoPass利用において秘密の質問の選択は非常に重要である。他人が推測することが難しく、自分が決して忘れないようなエピソード記憶を秘密の質

問として利用すべきであり、以下のような性質をもつ記憶は秘密の質問として利用すべきではない。

- 自慢になるもの(何かの機会にうっかり他人に話してしまう可能性がある)
- ネット上に記録が残っているもの
- 他人と情報を共有しているもの
- 趣味や嗜好に関連するもの(他人に推測されやすいうえに嗜好が変化する可能性がある)

このようなものではなく、「わざわざ人に話すことはないが自分の記憶に強く残っているような無難なエピソード記憶」を秘密の質問として利用するのが良いであろう。具体例としては以下のようなものがある。

- 昔のちょっとした怪我の場所や種類
- 昔のちょっと悔しい思い出
- 昔何かを見つけた場所

たとえば図1の3問目のような経験は他人に話したことが無いが、痛い思いをしたことは忘れないし、偽答の地名を並べるのも簡単なので、認証のための秘密の質問として適切であると考えられる。

3.4 偽答の作成方法

秘密の質問の種類によっては偽答を用意するのが難しい場合があるが、正答として人名や地名を利用する場合、正答に似た人名や地名をリストすることは難しくない。「世田谷」が正答であるとき、「目黒」「杉並」のような偽答を用意するのは簡単である。正答と同じカテゴリに属する単語を自動的にリストすることができれば正答をもとにして簡単に偽答のリストを生成することができる。ひとつの単語もしくは単語の集合と同じカテゴリに属する単語を検索する手法は「同位語検索」と呼ばれ、Webのデータを利用した様々な同位語検索システムが提案されている[17][24]。

人名や地名の偽答を作成したい場合は人名や地名のデータベースを利用して偽答を生成することができる。市町村の人口ランキングや位置関係のデータなどを利用すれば似た地名を偽答としてリストすることが可能であるし、人名ランキングを利用すれば似た苗字を偽答とすることができる。たとえば日本の名字ランキングの40位近辺に「長谷川」「近藤」「石井」「斉藤」「坂本」「遠藤」「藤井」などの名字があるので、「石井」が正答のときこれらの名字を偽答にすればよい。しかし、「小学生のとき〜だった同級生は誰?」という秘密の質問の正答が「石井」であるときこの方法で偽答を生成すると、「長谷川」「近藤」などの同級生の存在を確認することにより正答が「石井」であることが判明してしまう可能性がある。なので注意が必要である。

3.5 パスワード漏洩時の問題

秘密の質問を公開している場合、シード文字列とパスワードの対応がひとつでも漏洩してしまうと、総当たり計算でチェックすることにより、すべて秘密の質問の正答が判明してしまう。秘密の質問の正答を知っていればシード文字列からパスワードを計算することができるので、漏洩した秘密の質問は利用不可能になってしまう。3.1.1 や 3.1.2 のような運用をしている場合はパスワードがひとつ漏洩しても他のパスワードは安全だが、3.1.3 のような運用をしている場合はひとつでもパスワードが漏洩するとあらゆるパスワードが漏洩してしまうことになる。

通常の Web サービスなどのパスワードが漏洩することは考えにくい、「自転車の鍵番号」のように家族などで共有する可能性があるものに対して 3.1.3 のような運用を行なうと、鍵番号を知っている人物が総当たり攻撃を行なうことによって秘密の質問の答が判明してしまう。シード文字列や秘密の質問を秘密情報として扱わない場合はパスワードを他人と共有しないように注意する必要がある。

3.6 画像認証の利用

忘れにくいエピソード記憶を利用する認証手法として様々な画像認証システム [2][9][19] が提案されている。複数の画像の中から正答を選択するもの (Cognometric 方式)、ひとつの画像の中の特定の場所を指定するもの (Locimetric 方式)、画像の上で描画操作を行なうもの (Drawmetric 方式) が広く利用されているが [2][9]、EpisoPass は図 1 のように文字列のかわりに画像 URL を秘密の質問として利用する点が異なっている。Cognometric 方式はエピソード記憶を効果的に利用できるが、多数の偽答画像が必要だという問題がある。Locimetric 方式はエピソード記憶を効果的に利用できないことに加え、クリックしやすい「ホットスポット」は限られているため十分なエントロピーを確保できないことが問題になる [4]。また Drawmetric 方式もエピソード記憶を効果的に利用できないし、ユーザは似た傾向のストロークを選びがちであるため十分なエントロピーを確保しにくいことが知られている [13]。EpisoPass のように画像を秘密の質問として利用する場合、通常の秘密の質問の場合と同様に偽答を増やすことが容易であることに加え、画像に関連したエピソード記憶を有効に利用できるという利点がある [22]。

図 5 は本棚.org[20] のユーザのひとり³ が利用している画像認証問題である。このユーザ以外には正答は見当もつかないが、本人にとっては忘れることがないエピソード記憶と結びついた画像だということであった。



図 5. エピソード記憶と深く結びついた画像。

4 結論

エピソード記憶に結びついた秘密の質問を利用してパスワードを生成/管理できるシステム EpisoPass を提案した。EpisoPass は単純な原理にもとづいており柔軟な利用形態が可能であり、強力な秘密の質問を用意することにより秘密情報を全く覚えることなく安全な認証を行なうことができる。強力な秘密の質問を作成して安全に運用が可能かどうかを長期的に評価したいと考えている。

参考文献

- [1] AgileBits Inc. 1Password. <https://agilebits.com/onepassword>.
- [2] R. Biddle, S. Chiasson, and P. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.*, 44(4):19:1–19:41, Sept. 2012.
- [3] Dashlane, Inc. Dashlane. <https://www.dashlane.com/>.
- [4] A. E. Dirik, N. Memon, and J.-C. Birget. Modeling user choice in the PassPoints graphical password scheme. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pp. 20–28, 2007.
- [5] D. Florêncio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pp. 657–666, 2007.
- [6] D. Florêncio, C. Herley, and B. Coskun. Do strong web passwords accomplish anything? In *Proceedings of the 2nd USENIX workshop on Hot topics in security*, HOTSEC'07, pp. 10:1–10:6, Berkeley, CA, USA, 2007. USENIX Association.
- [7] E. Hayashi and J. Hong. A diary study of password usage in daily life. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2627–2630, 2011.
- [8] C. Herley, P. C. Oorschot, and A. S. Patrick. Passwords: If We're So Smart, Why Are We Still Using Them? In R. Dingledine and P. Golle eds., *Financial Cryptography and Data Security*, pp. 230–237. Springer-Verlag, 2009.

³ <http://hondana.org/Leiko>

- [9] Internet Safety Project. Graphical Passwords. <http://www.internetsafetyproject.org/wiki/graphical-passwords>.
- [10] M. Just and D. Aspinall. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pp. 8:1–8:11, 2009.
- [11] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2595–2604, 2011.
- [12] LastPass.com. LastPass. <https://lastpass.com/>.
- [13] D. Nali and J. Thorp. Analyzing user choice in graphical passwords. Technical Report TR-04-01, School of Computer Science, Carleton University, Ottawa, 2004.
- [14] National Institute of Standards and Technology. Electronic Authentication Guideline. NIST Special Publication 800-63-1. <http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf>, 2011.
- [15] A. Rabkin. Personal knowledge questions for fallback authentication: security questions in the era of Facebook. In *Proceedings of the 4th symposium on Usable privacy and security*, SOUPS '08, pp. 13–23, 2008.
- [16] S. Schechter, A. J. B. Brush, and S. Egelman. It's No Secret. Measuring the Security and Reliability of Authentication via 'Secret' Questions. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, SP '09, pp. 375–390, 2009.
- [17] R. C. Wang and W. W. Cohen. Language-Independent Set Expansion of Named Entities Using the Web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pp. 342–350, 2007.
- [18] C. Zarate. SuperGenPass. <http://supergenpass.com/>.
- [19] 小池 英樹, 増井 俊之, 高田 哲司. 画像を用いた個人認証手法. 情報処理, 47(5):479–484, May 2006.
- [20] 増井 俊之. 本棚通信: 控え目なグループコミュニケーション. インタラクシオン 2005 論文集, pp. 135–142, February 2005.
- [21] 増井 俊之. マイ認証. *Unix Magazine*, 21(3), March 2006. <http://www.pitecan.com/UnixMagazine/PDF/if0603.pdf>.
- [22] 増井 俊之. パスワードとの闘い - パスワードなし認証システムの運用報告. コンピュータセキュリティシンポジウム 2009 論文集, 2009.
- [23] 野村総合研究所. 利用者登録する商品・サービスを選別する傾向が強まった生活者と顧客情報の鮮度維持を望む事業者～生活者と事業者を対象とした ID に関する実態調査～. <http://www.nri.co.jp/news/2012/120208.html>, February 2012.
- [24] 大島 裕明, 小山 聡, 田中 克己. Web 検索エンジンのインデックスを用いた同位語とそのコンテキストの発見. 情報処理学会論文誌. データベース, 47(19):98–112, December 2006.
- [25] 平野 亮, 森井 昌克. パスワード運用管理に関する考察および提案とその開発. 電子情報通信学会技術研究報告. ISEC, 情報セキュリティ, 111(285):129–134, November 2011.

附録: パスワード計算アルゴリズム

パスワードは, 秘密の質問と回答の組合せにもとづいてシード文字列を換字することによって計算される. 換字は文字種ごとに行なわれる. たとえばシード文字列の 1 桁目が数字のときはパスワードの 1 桁目は数字に変換され, シード文字列の 1 桁目が記号のときはパスワードの 1 桁目は記号に変換される.

数字の換字の場合, シード文字列内の数字 A は, 換字関数 $f_N()$ によってパスワード内の数字 B に変換される. $f_N()$ は以下のような関数である.

$$f_N(x) = (10 + N - x) \bmod 10$$

ここで N はシード文字列と回答の組み合わせから計算される自然数で, 答の選択により変化する. たとえば $N = 5$ のとき, $f_5(x) = (10 + 5 - x) \bmod 10$ となるので, x と $f_5(x)$ の対応は以下ようになる.

$$f_5(0) = 5$$

$$f_5(1) = 4$$

...

$$f_5(8) = 7$$

$$f_5(9) = 6$$

$f_N()$ は N により変化するので, N がわからなければ $f_N()$ もわからない. N はシード文字列と回答の組み合わせから計算される自然数で, 秘密の質問の答とシードを知らなければ N を計算することはできない. EpisoPass では以下のようにして $f_N()$ を計算している.

1. 問題文字列と選択した答の文字列の組を連結した文字列 S を生成
2. S の MD5 ハッシュ値 M (16 進 32 桁の文字列) を計算
3. シード文字列の k 桁目の文字に対し, M の $(k-1) \times 4$ 文字目から $(k-1) \times 4 + 3$ 文字目までの部分文字列 (16 進 4 桁) を取得し, それを N とする. たとえばハッシュ値 M が 12345678... だったとき, $0x1234 = 4660$ なので, 1 桁目を計算する換字関数は $f_{4660}(x)$ となる.