

# 周辺情報の風覚化によるコンテンツ閲覧支援

## Representation of Peripheral Information by Wind for Assistance in Watching Content

水口 充<sup>▼</sup>

Mitsuru MINAKUCHI

風を用いてコンテンツ閲覧を補助する手法を提案する。一般的にコンテンツは視覚および聴覚情報として表現されているので、視聴する際にはユーザは注意を払う必要がある。一方、風は通知性が高く、視覚や聴覚とは別のモダリティであるので、ユーザが何らかのコンテンツを視聴中でも知覚可能であり、視聴を妨げることなく付加的な情報を提示することができる。また、風向がもつ空間性によって情報の存在に気づかせることができ、例えばユビキタスコンピューティング環境において実空間内での情報の所在を指し示すなどの用途が考えられる。本稿では送風ディスプレイとコンテンツの試作について述べ、有効性と応用可能性について議論する。

We propose use of wind to help users watch content. Content generally consist of visual and auditory information. Thus, users have to pay attention content while watching it. In contrast, wind can give users high awareness and is a different modality from visual and auditory senses. Hence, wind can be noticed by users who are watching content and can provide supplementary information without interruption. In addition, wind has spatiality that can indicate directions. Utilizing this characteristic for ubiquitous computing environment, for example, wind can direct users to a location where information is presented in the real world. In this paper, we describe prototypes of blow displays and contents and discuss their usefulness and utility.

### 1. はじめに

人間が視覚から得る情報の割合は8割〜9割と言われるように、我々は日常的な情報取得の大半を視覚に頼っている。

視覚情報は強力である反面、情報の獲得のためには視界に入っている必要があるし、情報の理解のためには意識を集中させなければならないという制約がある。つまり、視覚情報は大量の情報を一度に表現することが可能であるという長所を持つが、人間が意識を集中できる注視点は基本的に1箇所であるので[1]、閲覧者は能動的に視点を制御して情報を取得し理解しなければならない。

従来のコンピューティングにおいては基本的に、ユーザが目的を有して積極的に情報を取得するインタラクション形態を前提としてきたので、この制約が問題となることはなかった。しかし、ユビキタスコンピューティング環境が整備さ

れ、いつでもどこでも情報を利用できるようになると問題が生じてくる。すなわち、ユーザは四六時中画面を見ているわけではないので、システムがユーザに有用な情報を提示してもユーザがそれに気付かないことが起こりうる。

聴覚情報は視覚情報と比べて通知性が高く、特に注意を払っていない場合でも存在に気づきやすいので視覚情報への気付けに利用されることがある。しかし、聴覚情報には環境的な騒がしさの問題がある。また、高度な情報を伝達しようとすると視覚情報と同様の意識の集中の制約もある。

このような視聴覚情報の欠点を補うべく、本稿では風によるコンテンツ閲覧支援を提案する。以下、具体的なシステムについて述べ有効性と応用可能性について議論する。

### 2. 情報メディアとしての風

我々は実世界の中では五感を総動員させて周囲の状態を知覚している。例えば車を運転する際には加速度や振動を通してカーブの度合いやエンジンの状態を感じ取っている。料理をする際には味見をしたり温度や匂いを確かめることは欠かせない。これらの視覚・聴覚以外の知覚情報は、情報源への気づきや注目を促す役割を果たしている。前者の例では振動からエンジンの異常を感じればメータ類に視線を配るし、後者では匂いの変化で調理の過程を確認する。このように、我々は大量の情報源に囲まれながらも、様々な知覚情報によって意識を集中すべき情報源を選択している。

同様に、コンテンツを日常生活の中で利用する際にも、五感的に表現された補助的な周辺情報によってコンテンツの閲覧や取捨選択を支援できると考えられる。

そこで風による触覚の利用を検討した。触覚は視覚や聴覚とは別のモダリティであるので、ユーザが何らかのコンテンツを視聴中でも知覚できるという利点がある。これは、ユーザの活動を妨げないだけでなく、視聴中のコンテンツに関する補助的な情報を提示したり、関連情報の存在に気づかせる、といった用途に使うことができる。また、従来の触覚ディスプレイと異なり非接触型であるのでユーザは器具を装着したり触れたりする必要がない。さらに、温度や匂いといった他の感覚に比べると制御が比較的容易である。

風が表現可能な量は風量と風向である。加えて、断続的な送風と停止のパターンなどの時系列的な変動も利用できる。情報量としては少なく抽象度が高いが、メインの情報の補完や注意を促すといった用途には十分利用できるだろう。

特に、風向で方向を提示できるという特徴は、例えば詳細な情報を表示しているディスプレイを風で指し示すというように、情報自体が提示されている場所や、情報に関連している場所を指し示すなど、実空間を対象としたユビキタスコンピューティングにとって有用である。

### 3. 送風ディスプレイの試作

まず、風量をコンピュータから制御できる送風ディスプレイを試作した。詳細は文献[2]で報告済みであるので、ここでは簡単に概要を説明する。

図1にプロトタイプの概観を示す。左側は直流ファンとPhidgets社[3]製Phidgets Motor Controlの組み合わせによるものである。Phidgets Motor ControlはUSBでコンピュータに接続し、プログラムからパルス幅変調によって直流モータの回転速度を制御することができる。この試作品では最大約40段階で回転速度を制御できる。右側は市販の扇風機の風量スイッチをPhidgets Interface Kitのリレーで切り替える

<sup>▼</sup> 正会員 情報通信研究機構 [mmina@acm.org](mailto:mmina@acm.org)

ように改造したものである。扇風機は小型・縦長のシロッコファンで、3段階の風量調節ができる。



図1 送風ディスプレイのプロトタイプ  
Fig.1 Prototypes of Blow Displays

図2は2方向からの風向を提示するための簡単な設置の例である。2台のファンがユーザーの後方、左右約30度に配置されている。

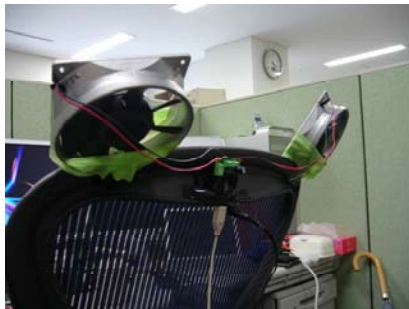


図2 椅子に2台のファンを設置した例  
Fig.2 Two Fans on a Chair

これらの送風ディスプレイで提示される風をどの程度正しく知覚できるかを調べる実験を行ったところ、風量・風向共に十分実用的な誤差内で弁別できることが分かった[4]。

## 4. アプリケーション

情報メディアとしての風の特徴に配慮し、二つの送風ディスプレイの利用形態を考えた。

### 4.1 補助的な情報の提示

閲覧中のコンテンツに関する補助的な情報を風量で表現する例として、3つのアプリケーションを実装した。

#### 4.1.1 重要度の風量化

一つめは、リンク先のコンテンツの重要度を風量で表現する例である。Web ページはリンクで他のページを参照しているが、参照先のページがどのようなコンテンツであるかは開けてみるまで分からない。このため、ユーザは往々にして試行錯誤的にリンク先を開ける必要があり、時間も手間もかかることになる。リンク先のページを先読みしたり、サムネイルなどをポップアップで表示する[5]などの解決方法も提案されているが、やはり能動的にリンク先を閲覧して内容を判断しなければならない。

これに対して、風量で重要度を表現することによって、リンク先のページを開けたりサムネイルを注視しなくても内容の善し悪しを予感し、閲覧時の取捨選択の判断の補助となることが期待できる。

Web ページの重要度としてはページランクを用いた。CGIとして実装したプロトタイプは、パラメータとして与えられたURLのページを取得し、そのページ中のリンクURLのそれぞれに対して、Web上で公開されているサービスを利用してページランクを取得する。そして、リンクをマウスオー

バーしている時に送風ディスプレイをページランクに応じた値で回転させるJava スクリプトを、元のページに埋め込んで出力する。

図3はJava スクリプトを埋め込んだ出力ページの例である。元になったWeb ページはソーシャルブックマークサイトの一つであるdel.icio.us[6]のhotlistである。簡単のために無関係な部分は省略している。

```
...
<script type="text/javascript"><!--
function spin(val) {
  var i = parseInt(val) * 5 + 50;
  httpObj = new XMLHttpRequest();
  if (httpObj != null) {
    httpObj.open("POST", "http://localhost:8080/fanControl/servlet/fanControl?value="+val, true);
    httpObj.send(null);
  }
}
// --></script>
...
<a href="http://www.frozenwarrior.com/~pricewatch/" on
MouseOver="spin(0)">Amazon Price Watch</a>
<a href="http://www.multicians.org/thvv/interviewing.html" onMouseOver="spin(5)">Interviewing and Hiring</a>
<a href="http://www.quackwatch.org/" onMouseOver="spin(7)"> Quackwatch Home Page</a>
...
```

図3 リンク先のWeb ページのページランクを風量化するスクリプトの例

Fig.3 Example Script for Producing Wind According to PageRanks of Linked Web Pages

図3中、function spin(val)はページランクの値を引数とし、その値に応じた風量で送風ディスプレイを駆動する関数である。送風ディスプレイの制御プログラムは別のサブレットとして実装しており、URL 経由で値を渡すことができる。この例では制御サブレットはlocalhostに設置しているものとしているが、実際はアクセス可能な任意のURLを指定可能であるのでネットワーク上の別のホストから送風ディスプレイを制御することもできる。

また、マウスがリンク文字列上にある時にspin(val)を呼び出すために、各リンクにはonMouseOver="spin()"が追加されている。引数はページランクの値である。

del.icio.usのhotlistはユーザによって登録されたWeb ページ中で人気の高いものをピックアップしたものであるが、ページランクはそれらの登録ページの中でも高いものから低いものまで様々であることは興味深い。つまり、ページランクはリンク構造をベースとする重要度であるのに対して、hotlistはどれだけ多くのユーザが重要であると感じたかというように、異なった尺度によるものである。このように異なった基準による指標を同時に提示することはコンテンツの選択の際に有用であると思われる。

#### 4.1.2 賑わいの風量化

二つめは、ネット上の掲示板の盛り上がり度に応じた風量を閲覧中に提示する例である。

例えば、「2ちゃんねるニュース速報+ナビ」[7]では、ニュース記事ごとのスレッドでの平均投稿速度を算出し、「○○res/h」のような形で表示することで人々の反応の高さを示している。これを利用して、4.1.1節と同様に、記事の文字列をマウスオーバーした時に平均投稿速度に応じた風量で送風ディスプレイを駆動するスクリプトを埋め込むCGIプログラムを実装した。スクリプトの出力例は図3と同様になるので省略する。このようにして、ユーザは数字を見

なくてもニュース記事に対する反応の多さを直感的に感じることができる。

他にも例えば、テレビ番組と番組実況チャットの融合において[8]、実況チャット上での発言の多さを風量化することで番組の盛り上がり度を体感するという応用が考えられる。

### 4.1.3 入力値の風量化

三つめは、フォームで入力中の値を風量として提示する例である。

図 4 はテキストエリアに入力した値の桁数に応じた風量で送風ディスプレイを駆動するフォームの例である。  
タグに記述された `onKeyUp="spin()"` によって、キー入力が行われるたびに風量が調節される。

```
<html>
<head>
<script type="text/javascript"><!--
function spin() {
    val = document.valueForm.source.value;
    v = parseInt(val);
    i = 0;
    if (v > 0) i = Math.round((Math.log(parseInt(val))
    / Math.LN10) * 5 + 50);
    if (i > 100) i = 100;
    httpObj = new XMLHttpRequest();
    if (httpObj != null) {
        httpObj.open("POST", "http://localhost:8080/fanControl/servlet/fanControl?value="+i, true);
        httpObj.send(null);
    }
}
// --></script>
</head>
<body>
<form name="valueForm">
input value <input type="text" size="15" name="source"
onKeyUp="spin()" />
</form>
</body>
</html>
```

図 4 入力値を風量化するフォームの例

Fig. 4 Example Script for Producing Wind According to Input Value

この例は閲覧支援ではなく入力支援であるが、昨今生じているようなオンラインショッピングサイトでの値段のつけ間違いや株の売買での株数の間違いといった、数値の入力間違いを未然に防ぐことが期待できる。

### 4.2 情報の所在の指示

液晶モニタの低価格化や PC の高機能化に伴い、個人が複数のディスプレイを同時に使用する作業環境も増えてきている。広大になった画面上には様々な情報ツールを常時表示させておいて随時閲覧したり新着情報を確認している。

これらの継続的な情報ツールはユーザによる気付きを前提としているが、重要な、あるいは緊急性の高い情報などを表示している際には積極的に気付かせたい場合もある。

そこで、ユーザが見るべき方向＝見て欲しい情報を表示しているディスプレイの方向を送風ディスプレイで指し示す実装を行った。

例えば図 5 のように表示ディスプレイと送風ディスプレイを配置する。各ディスプレイ上には、所定の条件で送風ディスプレイを駆動するように改造した Dashboard のウィジェット (MacOSX 上で動作する、HTML と JavaScript で記述されたアプリケーション) を配置する。

改造したウィジェットのスク립ト例を図 6 に示す。この例は、gmail の mailbox を簡易表示するウィジェットに、新着メールが届いた時に送風ディスプレイを駆動するコード

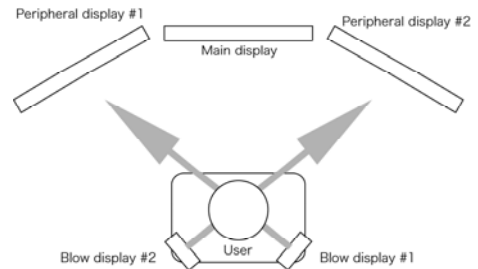


図 5 表示ディスプレイと送風ディスプレイの配置例  
 Fig. 5 Layout of Peripheral and Blow Displays

```
...
Widget.refresh = function(feedDocument) {
    MessagesTable.clear();
    Frontside.hideStatusSpan();
    if (feedDocument != null) {
        var fullCountTagList =
        feedDocument.getElementsByTagName("fullcount");
        if (fullCountTagList != null &&
        fullCountTagList.length > 0 &&
        fullCountTagList[0].firstChild != null) {
            var diff = fullCountTagList[0].firstChild.data -
            Widget.fullCount;
            if (diff > 0) {
                var request = new XMLHttpRequest();
                if (request != null) {
                    if (diff <= 1) {
                        request.open("GET",
                        "http://localhost:8080/doubleFan/servlet/doubleFan?fan
                        1=1&dur=5000");
                    } else if (diff <= 5) {
                        request.open("GET",
                        "http://localhost:8080/doubleFan/servlet/doubleFan?fan
                        1=2&dur=5000");
                    } else {
                        request.open("GET",
                        "http://localhost:8080/doubleFan/servlet/doubleFan?fan
                        1=3&dur=5000");
                    }
                    request.send(null);
                }
            }
        }
    }
}
...

```

図 6 ウィジェットのスク립ト例

Fig. 6 Example Script for a Widget

を追加したものである。前節の実装と同様、送風ディスプレイを駆動するプログラムはサーブレットとして実装されており、URL のパラメータ fan1 および fan2 で各送風ディスプレイの風量を、dur で駆動時間を ms 単位で、それぞれ指定するようにしている。

図 6 のスク립ト例では、新着メールの数に応じた風量で、右側の送風ディスプレイ (ウィジェットは左側の表示ディスプレイ上に表示されていると仮定している) を 5 秒間駆動するように記述している。なお、この例では簡略のためにウィジェットが表示されるディスプレイ＝駆動する送風ディスプレイを固定的としているが、汎用的には表示しているディスプレイを自動認識して駆動する送風ディスプレイを決めるようにすればよい。

## 5. 議論

実装したアプリケーションの有用性に関する定量的な評価は意図的に行っていない。なぜなら、今回の例はいずれも新たな情報表現方法を提案するものであり、それ自体の有用性を定量的に議論する性格のものではないからである。

定性的には、有用である可能性は十分であることを確認した。実装したアプリケーションを数名のユーザに試してもら

ったり暫く使ってみたところ、重要度や入力値の風量化はコンテンツを「体感」という新しい体験を与えるものとして概ね好評であったし、賑わいの風量化は人の気配や臨場感を表現することでオンラインコミュニケーションを支援するために有用であるとの反応を得た。

情報の所在の指示については、今回実装したプロトタイプはデスクワーク環境を対象としたため視界外の情報の存在に気付かせるケースは限定されていたが、それでも何かの作業に集中している時に新着情報の存在を知らせる用途としては十分に機能した。

補助的な情報は従来、視覚化あるいは聴覚化されて表現されてきた。例えば、重要度や賑わい度は数値として表示されたり、人気度はタグクラウドのように文字の大きさで表現されたりしている。他にも色や配置などの視覚情報で表現したり、効果音の大小や高低などで表現することもできるだろう。情報の所在の指示に関しても、ユーザの視界内にある表示装置を介して指し示すことで視線を誘導したり、音を3次元的に定位させて位置を示す方法も考えられる。

これらの表現方法は我々が長年利用してきたものであるから理解しやすいという利点がある一方、補助的な情報の視覚化は本来の情報を読み取りにくくしてしまうし、聴覚化は環境的な騒がしさの問題がある。情報の風覚化は視覚化・聴覚化と排他的に利用するのではなく、相互補完的に利用するのがよいであろう。

今回のプロトタイプは情報風覚化の効果を示すことを目的としてアドホック的に実装している。風覚化の対象とした情報の種類は効果的と思われるものを選定したが、これら以外にもデータ量や情報の新鮮度など、有用と思われる応用は沢山あるだろう。現在はコンテンツを処理して風覚化の対象となる情報を抽出しているが、応用ごとに処理方法を実装するのは効率が悪い。

風を利用する以外にも、温度や匂いなどの、他の五感メディアの利用も考えられる。風は制御が比較的容易である、量と方向を表現できる、という特性により応用が広いと考えられるが、アプリケーションや扱う情報の内容によっては他のメディアの方が直感的に理解できる場合もある。また、ユーザの状況や利用可能な装置などの制約によっても適切なメディアは変化する。さらに、例えば風の温度と強さなどで聴衆の反応の良し悪しと強度を同時に表現するというような、複数のメディアの併用も考えられる。

このような情報の体感化を汎用のものとするためには、コンテンツ自身にメタ情報として体感化可能な付加情報を記述、あるいは、プロキシサーバにて付加情報を抽出するなどのコンテンツ処理の共通化と、付加情報と表現メディアとを対応づけるための割当方法の一般化が望まれる。

## 6. 関連研究

風によって力覚を提示する方法自体は新しいものではない。例えば、鈴木らは風圧による力覚提示方式に関して、風圧と力の関係や衝突感覚などの基礎的な知見を示している[9]。また、小阪らはユーザの周りに3次元的に配置したファンで風を体感できる装置を作成し、風の知覚に関する評価実験を行い、応用を示した[10]。しかし、従来の風の利用は人工現実感において力覚あるいは風を受ける感覚を再現することを目的としており、本稿で提案するようなコンテンツ閲覧や情報の取得を補助する目的とは異なっている。

Pinwheels[11]はネットワークのトラフィックを風車の回

転で表現するアンビエントディスプレイである。これは情報の流れを風に見立てて風車の動きで視覚化したものであり、風自体をユーザに知覚させるものではない。しかし、元々の着想は風で情報を感じさせるところにあり、本研究はそれを具現化したものとも言える。

## 7. まとめと今後の課題

風を用いてコンテンツ閲覧や情報の取得を補助する手法を提案した。風量と風向をコンピュータで制御可能な送風ディスプレイを用いて補助的な情報の提示と情報の所在の指示の二つの利用形態についてプロトタイプを試作し、有効性と応用可能性を示した。

今後、更なる応用可能性を模索するとともに、情報の体感化のためのコンテンツ処理の共通化や身体的な動作による閲覧操作との融合を検討したい。

### [謝辞]

本研究のアイデアについて有益な議論をいただいた京都大学大学院社会情報学専攻の中村聡史氏に感謝致します。

### [文献]

- [1] Raskin, J.: "The Humane Interface", ACM Press (2000).
- [2] 水口 充, 中村 聡史: "送風ディスプレイを用いた協調型アンビエントシステムの提案", 第14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(WISS2006)予稿集, pp.27-30 (2006).
- [3] Phidgets Inc., <http://www.phidgets.com/>
- [4] 水口 充: "風感Web", DEWS2007予稿集 (2007). (to be appeared)
- [5] Kopetzky, T. and Mühlhäuser, M.: "Visual Preview for Link Traversal on the WWW", Proceedings of Eighth International World Wide Web Conference (WWW'99), pp.447-454 (1999).
- [6] del.icio.us social bookmarking, <http://del.icio.us>
- [7] 2ちゃんねるニュース速報+ナビ, <http://www.2nn.jp>
- [8] 宮森 恒, 中村 聡史, 田中 克己: "番組実況チャットを利用したテレビ番組のメタデータ自動抽出方式", 情報処理学会論文誌(トランザクション)データベース, Vol. 46, No. SIG18, pp. 59-72 (2005).
- [9] 鈴木 由里子, 小林 稔, 石橋 聡: "無拘束なインタフェースを目指した風圧による力覚提示方式", 情報処理学会論文誌, Vol. 43, No. 12, pp. 3643-3652 (2002).
- [10] 小坂 崇之, 宮下 芳明, 服部 進実: "没入型三次元風覚ディスプレイの開発と評価", インタラクシオン 2007 予稿集, pp. 105-112 (2007).
- [11] Ishii, H. Ren, S. and Frei, P.: "Pinwheels: Visualizing Information Flow in an Architectural Space", in CHI '01 extended abstracts on Human factors in computing systems, pp. 111-112 (2001).

### 水口 充 Mitsuru MINAKUCHI

情報通信研究機構専攻研究員。2004 京都工芸繊維大学大学院工芸科学研究科博士後期過程修了, 博士(工学)。ヒューマンコンピュータインタラクションの研究に従事。情報処理学会, 日本ソフトウェア科学会, ヒューマンインタフェース学会, ACM, 日本データベース学会各会員。