

近傍関係にもとづく情報検索システム

Information Navigation by Neighbor Hopping

増井 俊之 塚田 浩二 高林 哲*

Summary. When a user wants to find information on a computer, he usually has to choose the retrieval method and retrieval keyword before he can start the search operation. This means that a user has to know many things beforehand, including the structure of the data set, efficiency of available retrieval methods, effectiveness of using keywords, etc.

In this paper, we propose an information navigation technique based on the closeness between data items. When a user starts viewing a data item, similar items are listed automatically, and the user can select one of them and continue the operation. Since the similarity of information is calculated from various metrics, users can navigate through the wide information space only by selecting an appropriate data item from the list. For example, a user can start by viewing a picture, find a keyword attached to the picture, find another picture which has the same keyword, find a memo written on the day when the picture was taken, find another memo with similar contents, and so on. In this way, he can browse and find the information he needs only by selecting data items, without worrying about what search method should be used for the navigation.

We have implemented an information navigation system *AkimboFinder* based on this idea, and we show how people can use *AkimboFinder* to find the information they need only by repeatedly selecting related information.

1 はじめに

インターネットや計算機上では様々な検索システムが使われているが、情報の属性や分類のような構造をもとに検索する方法と、情報の内容をもとに検索する方法が主に使用されている。階層型ファイルシステム、Yahoo ディレクトリ、書籍の目次などは階層構造をもとに情報を検索するために使われており、検索ダイアログ、Google などの全文検索システム、書籍の索引などは内容をもとに検索を行なうために使われている。これらの使われ方はかなり異なっているが、カテゴリやキーワードのような検索条件を明示的にユーザが指定して検索を行なうという意味では類似している。

一方、World Wide Web はリンク関係にもとづいた大規模な検索システムであると考えられる。Web 上でブラウジングを行ないながら情報を探す場合は、明示的に検索条件を指定するかわりに、リンクによって関連づけられた情報をたどっていくことにより情報検索が行なわれる。

人間の頭の中は計算機データのように構造化されていないし、索引も存在しないため、条件を指定した検索を行なうのには向いておらず、Web のブラウジングにおける検索のように連想的に記憶をたどる

ことによって、必要な情報検索を行なうことができるようになっていられる。たとえば、何かを昔行なった場所を思い出そうとする場合、直接その場所の情報を思い出すことができなくても、そのときの天気/季節/時刻/会話/人物関係のような関連情報を思い出していくことにより、必要な情報にたどりつくことができることが多い。構造化されていないのに情報を思い出すことができるのは、情報の間に関連が形成されており、連想がうまく働くためだと思われる。

このように、人間の頭の中には膨大な量の雑多な情報から必要な情報を連想によって思い出すことができるのに対し、計算機の場合はファイル名や階層構造のような構造や検索キーワードをきちんと指定しなければ検索を行なうことは難しい。計算機上の情報についても、関連した情報が関連づけられて連想的に検索できるようになっていれば、階層構造や検索キーワードについて深く考えることなく、人間の頭の中の検索に似た柔軟な検索を行なうことができると考えられる。

本稿では、頭の中の情報を連想的に検索するのと同じような方法で計算機内の情報を検索可能な近傍検索システムを提案する。

* Toshiyuki Masui, 独立行政法人 産業技術総合研究所, Koji Tsukada, 慶應義塾大学, Satoru Takabayashi, 独立行政法人 産業技術総合研究所



図 1. 近傍関係の例

2 近傍検索システム

近傍検索システムは、ある情報をもとにしてその近傍の情報を選択するという操作を繰り返していくことにより必要な情報への到達を可能にするシステムである。ある情報に対し、それに内容が近い情報/同じ属性をもつ情報/作成時刻が近い情報/格納場所が近い情報/などをその情報の近傍情報であると定義し、情報を表示するときにはいつもこのような近傍情報を同時に表示して注目点を移動可能にする。ある情報の近傍情報の中からひとつを選択すると、今度は新たに選択された情報を中心とした近傍情報が表示される。この操作を繰り返すことにより、関連情報を連想的にたどって目的の情報に到達することができる。

図 1 のようにイベント/写真/メモ/メールなどの雑多な情報がいろいろな属性によって関連づけられている場合、時刻が同じであるとか内容が近いとかいったリンクをたどっていくことによって、連想的に情報を検索していくことができる。

筆者のひとりが近傍検索システムを活用して情報検索を行なうシナリオの例を以下に示す。

例 1: ある製品を見た展示会を思い出したい

- 会場で誰かとその話を議論したことを思い出す
- 展示会に一緒に行った人の情報を参照する
- その人に関連する写真リストの中から展示会で撮ったものを選ぶ
- 写真を撮った日時から展示会を特定する

例 2: 東大寺南大門の写真を探す

- 奈良案内システムを昔作ったことを思い出す
- 奈良案内システムのディレクトリをさがす
- プログラムの作成日付を調べる

- プログラムを書いたころの写真を格納したディレクトリを参照し、その中から南大門の写真を探す

例 3: 「なぞなぞドア」[14] の写真を探す

- 雑誌に記事を書いたのだから、原稿を作成したころに写真を撮ったはずだと思う
- 雑誌記事のディレクトリを参照する
- なぞなぞ認証の原稿を参照する
- 原稿を作成した周辺の時期の写真から、なぞなぞドアをさがす

例 4: 以前どこかの学会で発表された論文を探す

- その学会の直後に山に登ったことを思い出す
- 山登りの写真を探す
- その写真を撮った日の前後のイベントを調べる
- その中から学会を選び、Web ページに行って原稿を取得する

例 5: 類似システムのソースコードを探す

- 携帯端末用のシステムのプログラムソースを参照する
- 同じディレクトリにある Makefile を参照する
- その Makefile と中身が似た Makefile を参照し、類似システムのソースをみつける

このような連想的な検索は、誰もが頭の中では常に実行していると思われるが、計算機上では簡単には実行できないことが多い。たとえば例 4 の場合、まず山登りに行ったことを思い出し、ファイルブラウザや写真管理ソフトでその写真を探し、ファイルの属性を調べることによって撮影日時を調べ、予定表でその日の周囲の予定を参照して学会を捜し、学会のホームページを検索するといった手間が必要になる。それぞれの検索ステップにおいて必要な操作は異なっており、やりたいことと実際の操作とのマッピングは難しいため、計算機の操作に詳しいユーザでないとこのような連想的な検索を手際よく実行することは難しい。これらのステップがすべてクリックひとつで実行できるようになっていれば、このような連想的な検索を誰でも簡単に実行できるようになる可能性がある。

構造を利用する検索でもキーワードによる検索でも、現在一般的な検索手法では、検索条件を人間が明示的に指定しなければならないのが普通であるし、構造を用いて検索するのか/キーワード検索をするのかなどといった検索手法を指定しなければならない。フォルダウィンドウを開いたり、リストコマンドを発行したりしてファイルの存在を確認するような作業は、計算機ユーザが毎日に行なってい



図 2. PalmWiki のディレクトリを参照

る作業ではあるが、常に計算機内のデータの構造を把握しつつ適切な操作を行なわなければならないため、心理的負担が大きいといえる。また、Google のような全文検索システムを用いて内容検索を行なう場合は、情報を見つけるのに十分なキーワードが存在するか考えたり、検索のクセを理解したりしなければならないため、やはり負荷が高い作業であるといえる。どの方法を使ってどのように検索を行なうかを即座に判断しながら素早く情報にアクセスできるようになるためには習熟が必要であり、計算機の利用をさまたげる大きな原因になっていると考えられる。近傍検索システムでは、見えているリストの中から必要そうなものを選択するというブラウジング操作を繰り返すだけで結果的に目的の情報に到達できることを目標としている。

3 近傍検索システムの使用例

以上のような考えにもとづき、Web ブラウザで使用するののできる近傍検索システム *AkimboFinder* を試作した。*AkimboFinder* は Web アプリケーションとして動作する。

AkimboFinder では以下の基準にもとづいて近傍度を判定し、注目しているファイルの近傍関係にあるファイルを周囲にリストする。

- 注目しているファイルの含まれるディレクトリ構造

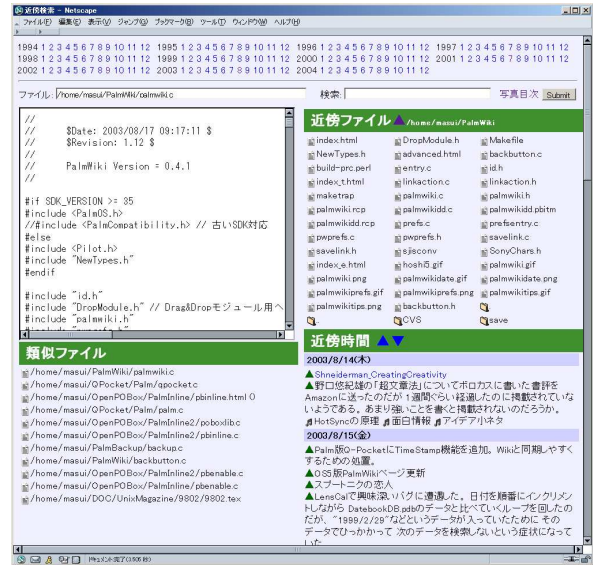


図 3. PalmWiki プログラムを参照

- ファイルの更新日時
- ファイルの内容
- キーワード (登録されている場合)

ファイルの内容の類似度計算には文書間の類似度計算ライブラリ GETA (Generic Engine for Transposable Association)¹[11] を使用している。検索対象となる全てのファイルについてあらかじめインデックスを作成しておき、*tf·idf* 法 [10] にもとづいた類似度が注目ファイルに近いものをリストしている。インデックスの作成には時間がかかるが、ギガバイトオーダのテキストファイルの全文検索実行でも 2~3 秒で終了する。

プログラムの検索 *AkimboFinder* で “PalmWiki” というソフトウェアの開発ディレクトリを参照したところを図 2 に示す。ページの一番上のファイル名入力枠で “PalmWiki” を指定することにより *AkimboFinder* を呼び出すと、ディレクトリ内の “index.html” が左上に表示される。このディレクトリには Palm 上でハイパーテキストを実現するソフトウェアの開発に関する情報が格納されており、同じディレクトリの中のファイルや、このファイルを更新した時刻近辺で作成されたファイル/メモなど関連する情報が画面の右側に表示されている。

ディレクトリ内にある “palmwiki.c” をクリックすると画面は図 3 のように変化する。通常のファイルブラウザでディレクトリをアクセスした場合はそのディレクトリの内容しか参照することができないが、*AkimboFinder* では、palmwiki.c を作成したの

¹ <http://geta.ex.nii.ac.jp/>

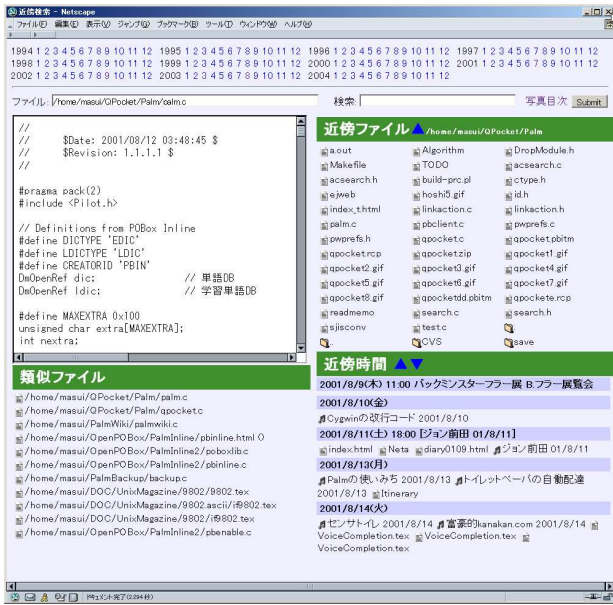


図 4. Q-Pocket プログラム周辺



図 6. 「文京 パーティー」の検索結果



図 5. 近傍ファイルの選択による注目点の遷移

と同じ頃に作成したメモやファイル, palmwiki.c に類似したファイル, その時期に作成したメモ, 写真なども同時に一覧できるようになっている。

ファイルをエディタなどで書き出したときには, その更新時刻が記録されている。また, 筆者の計算機では, アイデアなどのメモは日付/時刻を基にしたファイル名に格納しているため, ある時刻近辺のアクティビティを簡単にリストすることができる。たとえば, palmwiki.c の作成時刻は 2003 年 8 月 17 日なので, その頃に書いた文書が右下に表示されている。

左下には「類似ファイル」として, palmwiki.c に内容が類似したファイルや関連する雑誌記事がリストされている。この中から QPocket/Palm/palm.c を選択すると, 今度はこのファイルが含まれるディレクトリの内容やファイル作成時期に書いた文書が表示される (図 4)。

近傍ファイルの選択による注目点の遷移の様子を図 5 に示す。

写真の検索 AkimboFinder では, キーワードをつけていない写真でも, 近傍時刻のテキスト情報などを利用することによりキーワード検索を行なうこと

ができる。図 6 は「文京 パーティー」というキーワードで検索を行なった結果である。このキーワードに近いメールや文書が左下にリストされており, また右側にはメール送出日近辺で作成された文書や写真がリストされている「近傍時間」の 6 月 4 日のところには文京オフィスで行なわれたパーティーの写真が表示されている。写真にはキーワードは付加されていないにもかかわらず「文京 パーティー」というキーワードでその写真を検索できたことになる。ここで写真のひとつをクリックして近傍検索を行なうと, 画面は図 7 のようになる。図 6 とほとんど画面は変化していないように見えるが, 図 7 は写真のみをもとにして近傍検索を行なった結果であり, ユーザーがキーワードを指定して検索を行なったものではない。この写真にはキーワードが付加されていないが, 写真の撮影時刻からその近傍の文書や予定表情報を抜き出し, それをもとにして全文検索を行なった結果, この写真に関連したメールや文書などの情報を一覧することができる。

添付キーワードにもとづく検索 手間をかけて写真にキーワードを登録すれば, より効果的に関連情報をリンクすることができる。キーワードが登録された写真を選択すると, 同じキーワードをもつ写真をリストすることができるので, 写真を中継として関連した情報を検索することができる。

図 9 は 2000 年 10 月に登った山の写真であるが, これにはキーワードを添付してあるため, 同じキーワードをもつ写真がいくつか下に表示されている。このひとつをクリックすると画面は図 10 のように変化する。場所は同じだが, こちらは 2002 年の写真である。



図 7. 写真のひとつを選択した結果



図 9. 蓼科山頂

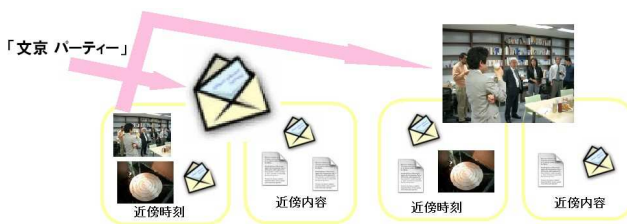


図 8. 注目点の遷移

近傍ファイルの選択による注目点の遷移の様子を図 11 に示す。

4 議論

4.1 近傍を積極的に利用したファイル管理

近傍検索がうまく使えれば、従来とはかなり異なる方法でファイルを管理することができるようになる。

最近広く利用されている計算機では、ファイルはディレクトリの木構造を利用して管理するものが多く、関連のあるファイルは同じディレクトリか近くのディレクトリに置いておくのが普通である。たとえば、文章に関連した写真は、整理のために文章と同じか近くのディレクトリに置いておくのが一般的であるし、そうしておかなければ参照したり整理したりするのに不便である。

しかし、作成時刻や関連情報などをもとに近傍情報に簡単にアクセスできることが確実であれば、関連するファイルを同じディレクトリに置く必要はなくなってしまう。近傍検索システムの場合、時間的に近いファイルや同じキーワードをもつ情報は近くに表示されるから、写真ファイルが全く別の写真ディ

レクトリに置いてあっても問題ない。

このように、いろいろな属性における近傍で検索ができると、属性を使って縦横無尽に関連情報間をジャンプすることができる。内容的な近さ、作者の近さなど、いろいろな属性において近傍に存在する情報を簡単に参照できれば、従来と全く異なるファイル管理が可能になるだろう。

Time-Machine Computing[7] や Lifestreams[2] では時間情報を活用した情報管理手法が提案されているが、AkimboFinder の場合は時刻情報に加えて、内容による分類やファイルシステム構造も同時に利用することができるためさらに効率のよい情報管理が可能になっている。

4.2 デジタル写真の効果的な利用

デジタルカメラが普及し、ハードディスクの容量が飛躍的に増大した結果、何千枚/何万枚もの写真データを誰でも扱えるようになり、各種の写真整理/検索ソフトが提案されている [15]。多くの写真整理システムでは、カテゴリ分類またはキーワードにもとづいた検索を行なうことができるようになっている。

- iPhoto, IrfanView のような画像ビューワや Windows のフォルダ画面などはサムネイル表示をサポートしており、多くの画像をディレクトリ単位で眺めて検索することができる。PhotoMesa[1] は、サムネイル画像を任意の倍率で拡大縮小可能にすることにより、複数のディレクトリにまたがった大量の画像を一覧できるようになっている。Adobe の PhotoShop Album では、撮影時刻にもとづいて容易にブラウジングを行なうための工夫がされている



図 10. 別の時期の写真



図 11. 注目点の遷移

し、時刻にもとづいて自動的にクラスタリングするシステムも提案されている [4] .

- PhotoFinder[5] は、写真の一部に対してキーワードを付加することができる。Aria[6] システムでは、メールに写真が添付されている場合のようにメールテキストと画像に明らかな関連がある場合に画像に自動的にキーワードを付加したり、簡単なユーザ操作によって画像とテキストを関連づけたりしやすいような工夫がされている。また、Vannevar Bush の壮大な Memex 構想を実現しようとしている MyLifeBits システム [3] では、写真や動画などにキーワードを添付する簡単な方法を多数用意することによりキーワードによる検索を可能にする工夫がなされている。

このように、適切なカテゴリ分類やキーワード添付が行なわれていれば効率的に写真を検索することができるが、分類やキーワード付けは非常に手間のかかる作業である。いろいろな機能をもつ写真整理

システムにおいて実際のユーザがどのように機能を利用したかを調査したレポート [9] によれば、タグ付けや類似画像検索のような高度な機能はほとんど使われず、ほとんどのユーザは時間情報を利用したり名前をつけたフォルダに分類したりするだけだったという。

写真をカテゴリ分けしたりキーワードを付加したりすることによって写真検索に成功したとしても、多くの写真整理システムでは検索のフローはそこで途切れてしまう。たとえば、場所や参加者に関するキーワードを使ってパーティーの写真をうまく検索できたとしても、その写真をもとにしてパーティーの案内メールを捜すことはできない。このように、ただ写真を検索しやすくするために日常的に大きな手間をかけてキーワードを付加する作業は見返りが少ないため継続するのが難しいと思われる。

一方、近傍検索の場合は、前述の検索例で示したように、普通に予定表を使ったりメールをやりとりしていれば、写真を分類したりキーワードをつけたりしなくても近傍情報を用いて写真をテキスト検索することが可能であるし、逆に写真をもとにしてそれに関連した写真、文書、メールなどを 1 クリックで検索することができる。近傍検索システムでは写真整理にまったく手間をかけなくても必要な写真を簡単に検索することができるし、写真に関連するテキストやメールを捜すことができる。また、キーワードを登録した場合は情報全体がより有機的に結合され、ナビゲーションの幅が大きく広がるので、キーワードを登録する意欲が持続する可能性が高い。

4.3 自動提示にもとづくインタフェース

一般的な検索システムでは、検索条件や検索手法をユーザが指定してはじめて検索結果を得ることができるが、AkimboFinder の場合はユーザは提示されている候補から必要なものを選択するだけであり、検索手法を指定する必要がない。与えられた選択枝の中から必要なものを選ぶという行為は、何が 필요한のか考えたり、必要なものを得る手段を考えたりすることに比べると負担は小さい。

水口らは、ユーザが何もしない状態でも新しい情報が次々と提示され、自分が欲しい情報が出現したときだけユーザがアクションを起こすことにより必要な情報を取得できるようにする「提示型インタフェース」を提案している [12]。また、渡邊らは、画面上を浮遊するキーワードが衝突により Google 検索を行ない、新たなキーワードが生成されることにより無限にキーワード生成と検索が実行されるのを眺めることができる「眺めるインターフェイス」を提案している [17]。これらのシステムのユーザはシステムに対して受動的に行動することが特徴的であり、ユーザが検索手法の詳細について深く考えなくても結果的に情報検索が行なわれるという点で、近傍検索と

類似している．また，Rhodesらの Remembrance Agent[8]は，ユーザが作成中のテキストに内容が近いファイルをシステムがリアルタイムに検索してユーザに提示することによって関連情報を思い出したり入力作業を効率化したりできるようにしたシステムである．これらのシステムのように，検索条件指定や検索開始/終了処理など，ユーザの能動的な操作を極力少なくし，ユーザはただ選択操作のみ行なうようにしたシステムはユーザの負担を減らすという点で有望であると考えられる．

4.4 近傍検索の拡張

様々な属性の利用 AkimboFinderではファイルの場所，内容，日付などにもとづいた近傍関係を利用しているが，これら以外にも有用な属性はいろいろ考えられる．たとえば写真の場合，撮影場所や画像の類似関係を利用することが考えられるし，HTML文書の場合は内容ではなく外見の類似関係を利用することができるだろう．近傍関係を利用できる属性の追加によりシステムはより有用となると考えられる．

予測入力への利用 あるテキストの近傍にあるテキストは内容や話題が近いものであることが多いため，そのテキストを編集する場合には近傍テキストの情報が参考になることが多いと考えられる．小松らは，日本語の動的補完システム Nanashiki[16]を利用することによって関連テキストを利用した効率的な予測型入力システムを提案している．小松らの予測型入力システムでは，予測型テキスト入力システム POBox[13]と同様に，ユーザの入力に応じて入力単語を予測して候補として提示するが，関連するテキストに含まれる単語を優先して候補として採用するようになっている．「議論」「原稿」のような単語をよく使用するユーザが「g」を入力した場合，通常はこれらの単語が候補になるが「岐阜での学会」に関するメールに返事する場合は「岐阜」や「学会」のような単語が優先的に候補となる．小松のシステムでは，同じ編集バッファ内にある単語のみが優先的に候補として使用されるようになっているが，近傍テキストに含まれる単語や文章を予測に利用することにより，予測入力の効率が向上すると考えられる．

情報の編集 AkimboFinderは近傍関係にもとづいた情報検索のみを行なうものであるが，注目情報を編集可能にすることにより Wiki Wiki Web (Wiki)のような使い方が可能になる．一般的な Wikiではユーザが明示的に指定した名前をもつページのみがリンクになるが，近傍ページが自動的にリンクになるように拡張を行えば，近傍検索システムと Wikiの特長を兼ね備えたシステムを構築できるであろう．

運用の知見 よくできた Web ページは，効率よくリンクをたどっていくことにより効果的な検索を行な

うことができるように設計されているが，近傍検索を利用していると，自分に関係する全情報に関するよく設計された Web ページが自動的に作成されているような印象をもつ．GUI操作でフォルダを開いたりコマンドを発行してディレクトリを移動したりするときには，発見的な楽しみはほとんど存在しないが，AkimboFinderではしばしば思わぬ文書や写真が見つかることがあり，時間を忘れてブラウジングしてしまうことがある一方，もともと何を捜していたのかを忘れてしまうことがあるので注意は必要である．

AkimboFinderではメールやメモなどあらゆるファイルを検索対象にしており，何を検索する場合でもとにかく AkimboFinderで検索キーワードを入力すれば必要な結果が得られるので，手持ちの情報を検索する場合は，とにかく近傍検索システムを開いてキーワードを入力するのが習慣になっている．近年 Web 上で何か検索を行ないたい場合は，Googleでキーワードを指定すると大抵の場合うまく情報がみつかるので重宝であるが，計算機上で何か情報を検索したい場合は，近傍検索を行なえばうまくいくことが多い．何を検索する場合でも同じ検索システムが利用できるのは心理的負担が軽くなる．

AkimboFinderの使用例で示したデータは著者のひとりが使用している実データにもとづいているため，実際には有用な検索が行なわれた場合でも，画面を論文上で公開しにくいケースが多く，実例の選択には苦労した．

5 結論

リンクのクリックを繰り返すことにより情報検索を行なうというスタイルは Web ブラウザの普及により非常にポピュラーなものとなっている．本稿では，近傍にある情報へのリンクを自動生成することにより Web ブラウジングと同じ使い勝手に連想的に情報検索を行なうことができる近傍検索システムを提案し，その特長について述べた．本手法をさらに使いやすく改善することにより，既存の計算機操作の多くを置き換えることを目指していきたい．

参考文献

- [1] Ben Bederson. Photomesa: A zoomable image browser using quantum treemaps and bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST2001)*, pp. 71–80. ACM Press, November 2001.
- [2] Eric Freeman and Scott Fertig. Lifestreams: Organizing your electronic life. In *AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval*, Cambridge, MA, November 1995.
- [3] Gim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits:

- Fulfilling the memex vision. In *ACM Multimedia '02 Proceedings*, pp. 235–238, December 2002.
- [4] Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In *JCDL2002 Proceedings*, pp. 326–335, 2002.
- [5] Hyunmo Kang and Ben Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *Proc. IEEE International Conference on Multimedia and Expo (ICME2000)*, August 2000.
- [6] Henry Lieberman, Elizabeth Rosenzweig, and Push Singh. Aria: An agent for annotating and retrieving images. *IEEE Computer*, pp. 57–61, July 2001.
- [7] Jun Rekimoto. Time-machine computing: A time-centric approach for the information environment. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'99)*, pp. 45–54. ACM Press, November 1999.
- [8] Bradley J. Rhodes and Thad Starner. Remembrance agent: A continuously running automated information retrieval system. In *Proceedings of The First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pp. 487–495, 1996.
- [9] Kerry Rodden and Kenneth R. Wood. How do people manage their digital photographs? In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI2003)*, pp. 409–416. Addison-Wesley, April 2003.
- [10] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, Vol. 24, No. 5, pp. 513–523, 1988.
- [11] 西岡真吾, 今一修. 汎用連想計算エンジン GETA とそれに基づく連想検索システム. 情報処理学会自然言語処理研究会 研究報告, Vol. 2000, No. 137, June 2000.
- [12] 水口充, 浦野直樹. 提示型インタフェースによる情報探索システム. インタラクティブシステムとソフトウェア VI: 日本ソフトウェア科学会 WISS'98, pp. 159–164. 近代科学社, 1998.
- [13] 増井俊之. ペンを用いた高速文章入力手法. 田中二郎 (編), インタラクティブシステムとソフトウェア IV: 日本ソフトウェア科学会 WISS'96, pp. 51–60. 近代科学社, December 1996.
- [14] 増井俊之. なぞなぞドア. *UNIX Magazine*, Vol. 16, No. 8, pp. 185–189, August 2001.
- [15] 増井俊之. 界面駭客日記 (21) - 写真整理地獄. *ASCII*, No. 302, August 2002.
- [16] 小松弘幸, 高林哲, 増井俊之. 日本語動的単語補完方式 Nanashiki を活用した予測入力. インタラクティブシステムとソフトウェア IX: 日本ソフトウェア科学会 WISS2001, pp. 67–74, December 2001.
- [17] 渡邊恵太, 安村通晃. Memorium: 眺めるインタフェースの提案とその試作. 第 10 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 (WISS2002) 論文集, pp. 99–104, November 2002.