

# PC 操作画面をリアルタイムに配信する Web システムのための伝送量圧縮方式

市村 哲<sup>†</sup> 間下 直晃<sup>††</sup> 伊藤 雅仁<sup>†</sup>  
 宇田 隆哉<sup>†</sup> 田胡 和哉<sup>†</sup> 松下 温<sup>†</sup>

本論文では、PC の操作画面やアプリケーション画面をほぼリアルタイムに配信する Web システムのための伝送量圧縮手法「Interframe Compression Method for HTML Push-out」(ICMHP)を提案する。著者らが実装した QuickBoard は、PC のデスクトップ画面をスナップショット画像として画像ファイルに保存し、これを HTTP で Web サーバから Web ブラウザに擬似プッシュ転送する Web システムであるが、ICMHP はこのような情報プッシュ型 Web システムに適した伝送量圧縮手法である。ICMHP は、プラグインや ActiveX 等のプラットフォームに依存したコンポーネントを用いることなく実装できることが特徴である。このため、汎用的な Web ブラウザや、PDA 等に搭載されている簡易 Web ブラウザに PC 画面を配信する際に用いることが可能である。本論文では、実際の教育現場の要求を観察した結果について報告し、この観察結果に基づいて作成したプロトタイプシステムについて述べる。

## A Compression Scheme Suitable for Sending Out Desktop Screen to Web Browsers

SATOSHI ICHIMURA,<sup>†</sup> NAOAKI MASHITA,<sup>††</sup> MASAHITO ITO,<sup>†</sup>  
 RYUYA UDA,<sup>†</sup> KAZUYA TAGO<sup>†</sup> and YUTAKA MATSUSHITA<sup>†</sup>

We developed "Interframe Compression Method for HTML Push-out" (ICMHP), a method integrating interframe compression mechanism used in video compression technology with web-server-side technology, delivering computer-screen images to standard web browsers, or even PDA portable web browsers. This paper describes a QuickBoard web-server system that implemented ICMHP scheme. The system allows users to deliver computer-screen images of any application to more than 100 web browsers running on various devices in quasi-real-time. That no client software except a standard web browser is needed for the user's PC is an important feature of the system. This paper describes the implementation and evaluation of the prototype we developed.

### 1. はじめに

PC の普及、または、常時接続およびブロードバンドの浸透とともに、E-ラーニング（遠隔学習）が重要なアプリケーションの 1 つとしておおいに注目を集めている。E-ラーニングの形態としては、すべての学習コンテンツをサーバにあらかじめ蓄積しておくオンデマンド学習の形態も存在するが、遠隔大学講義や、遠隔セミナー配信、遠隔英会話教育等に関しては、リアルタイムに講義をしたいという要求が強い。リアルタイムに講義を配信することで、従来の集合教育のよ

うな場が構築され、講師と学習者または学習者どうしが密にインタラクションできるようになるという効果がある。リアルタイムの遠隔講義では、講師の顔映像と音声に加えて、講師が用いているアプリケーションの表示画面をタイミングよく配信することが重要である<sup>1),2)</sup>。

一方、現在 Web システムが情報システムの主流となっている。システムの初期導入やバージョンアップの際にでも、サーバ側のプログラムを変更するだけで済み、システムの管理コスト・開発コストを大幅に削減することができるメリットがある。また、特殊なネットワーク環境や特別なクライアントソフトが不要という利点がある。このように利点の多い Web システムであるが、HTTP の特性上、情報をリアルタイムにユーザに配信するようなサービスに向かないという性

<sup>†</sup> 東京工科大学  
 Tokyo University of Technology  
<sup>††</sup> 株式会社バイキューブ  
 V-cube Inc.

質がある。たとえば、Web サーバから複数の Web ブラウザに向けて情報をプッシュ配信することができず、このことがリアルタイム型の Web システムを設計する際の大きな制約となっている。

そこで著者らは、Web システムの利点を活かしつつリアルタイムに情報を配信できるシステムを構築することが重要と考え、PC の表示画面を、PC、PDA 等の汎用 Web ブラウザにほぼリアルタイム（以下、「準リアルタイム」）に配信できる Web システム QuickBoard<sup>3)</sup>を開発してきた。本システムは、PC の表示画面をスナップショット画像として画像ファイルに保存し、これを Web サーバから HTTP で Web ブラウザに転送することを特徴としており、PC に表示された任意の画面を複数の Web ブラウザにプッシュ配信する機能を有している。具体的には、Web ブラウザが定期的に Web サーバに対して問合せを行い、送信するデータに変更があった場合にだけ Web サーバから Web ブラウザにデータを転送するように実装されている。このようにして、プッシュ型配信をサポートしない HTTP 通信において擬似的なプッシュ配信（以下、「擬似プッシュ配信」）を可能としている。

しかしながら、現状の QuickBoard システムは、画面の一部領域が変更されただけで、全領域を転送するようになっていたため<sup>4)</sup>、データ転送容量が多くなってしまいう問題があった。たとえば講師は、伝送したいデスクトップ画面領域をマウスによって指定する必要があるが、「テキストエディタ全体」というような、一般的には広い画面領域を指定することが普通である。この場合、エディタ上の数文字分しか画像が更新されていなかったとしても、テキストエディタ全体のスナップショット画像が作成されて全 Web ブラウザに伝送されることとなる。

本論文では、PC のデスクトップ画面やアプリケーション画面を擬似プッシュ配信する Web システムのための伝送量圧縮手法「Interframe Compression Method for HTML Push-out」（以下、「ICMHP」）について提案する。ICMHP は、送信すべき領域のうち、変化のあった領域だけを差分伝送することを Web システム上で実現する方式である。ICMHP は、プラグインや ActiveX 等のプラットフォームに依存したコンポーネントを用いることなく実装できる方式である。本論文では、構築したシステムの機能および評価の結果について述べる。

## 2. Web システムのための伝送量圧縮方式

### 2.1 要求分析

著者らは、実際の現場を対象として要求分析し、そこで発見された課題を解決することが重要と考えている。本件では、リアルタイムな講義資料配信が必要となる現場として、著者らが所属する大学の「UNIX システムプログラミング演習」の授業を選定した。本授業の受講学生数は 100 名弱であり、各自、FreeBSD + XFree86 がインストールされた PC を用いて演習授業を行っている。従来は、書画カメラと大型テレビとの組合せによって資料提示を行っていた。学生からは資料の文字が見にくい等の問題点が指摘されており、これらの問題を改善することが課題となっていた。授業観察した結果、以下のような必要要件が明らかとなった。

- A) 講師が使うアプリケーションを限定しない。
- B) 受講者は Web ブラウザ以外必要としない。
- C) 同時に 100 人以上が接続できる。

A) に関し、リアルタイムに情報を Web 配信する遠隔講義システムとしては、講義資料をあらかじめサーバにアップロードし、サーバ上で画像ファイルに変換して各 Web ブラウザに配信する例がある。田中らの遠隔講義支援システム<sup>5)</sup>は、PowerPoint ファイル等をサーバにアップロードし、サーバ上で画像ファイルに変換して各 Web ブラウザに配信している。また、Web ブラウザに表示する URL を遠隔地間で交換することで、複数人間で同期 Web ブラウジングするシステムも提案されている<sup>6),7)</sup>。しかしながら、計算機実習のような授業では、講師の使用ツールを限定することはできず、また、必要に応じてテキストエディタ上にタイプした文字を見せたり、デバッグツールの動きを見せたりしながら講義を進めたいという要求がある。よって、サーバにあらかじめ資料をアップロードしておかなければならない従来の方法ではこれらの要求に対応できない。

B) に関し、MPEG による配信やストリーミングビデオによる配信（Windows Media Screen<sup>8)</sup>と Windows Media Encoder<sup>8)</sup>の組合せ等）は、現状では専用のアプリケーション、または、OS に依存した専用のプラグインをユーザの環境に導入する必要があり、今回対象とする使用環境においては利用できない。また、リアルタイムに任意のアプリケーションの画面を遠隔伝送するための手段としては、Netmeeting<sup>9)</sup>や VNC<sup>10)</sup>等の遠隔デスクトップ共有ツールが存在し、これらのツールも PC 画面をお互いに送りあう機能を

備えている。しかしながら、これらのツールを利用するためには、利用するユーザ全員が専用のクライアントソフトウェアをインストールして用いる必要があり、使用できるオペレーティングシステムも限定されてしまうという問題があった。これらの制約から、クライアント側は Web ブラウザだけで動作するように構成することを要件とした。

C) に関し、Netmeeting や VNC 等のデスクトップ画面共有ツールは、基本的に 1 対 1 通信のためのツールであることから、多人数または多拠点への同時情報配信に用いることができないという問題がある。このため、100 名程度の受講生にリアルタイム配信しなければならないという今回の用途に用いることはできない。

以上の要求分析に基づき、著者らは過去において QuickBoard システムを構築し、従来の書画カメラ + 大型テレビによる資料表示方法を、Web システムを用いた資料表示方法によって置き換えた。QuickBoard は、PC の表示画面をスナップショット画像として画像ファイルに保存し、これを Web サーバから HTTP で Web ブラウザに転送することによって、準リアルタイムな PC 画面の配信を行う。具体的には、Web ブラウザが定期的に Web サーバに対して問合せを行い、PC 画面が変更されたことが検知された場合にだけ、Web サーバから Web ブラウザに画像データを転送するようになっている。

## 2.2 従来技術の問題点

前述のように QuickBoard では、PC 画面が変更された場合に画像データを転送するようになっているが、現状では、画面の一部領域が変更されただけで、全領域を転送するようになっているため<sup>4)</sup>、ネットワーク転送量が多くなるという問題があった。

ネットワーク転送量を減少させて画像を転送する技術としては、MPEG2 や MPEG4 等のデジタル映像圧縮技術<sup>11)</sup>が存在する。連続する画像フレーム間の相関を利用し、動き補償フレーム間予測およびフレーム間差分転送により、前のフレームの画像ピクセルのうち、変更があった画像ピクセルの情報だけを転送できるようになっており、少ないデータ量で映像情報を転送できる。しかしながら、絶えず画像が書き換わる映像の圧縮を対象とした技術であるために、プレゼンテーションスライドの転送には適さない部分が多いのが実情である。著者らがネットワーク帯域に関する実験を行った結果、パワーポイントスライドで普通に用いられる文字フォント(文字サイズ 16 ポイント)がはっきり読める画質で MPEG2-TS over IP 伝送しようとす

ると、約 4.5 ~ 6 Mbps のネットワーク帯域が必要となることが分かった。仮に 100 名にユニキャストで同時配信したとすると、450 Mbps から 600 Mbps のネットワーク帯域を占有する必要がある、一般的なネットワーク環境でのリアルタイムスライド配信はきわめて困難となることが明らかである。

この原因として、MPEG2 over IP 等の映像圧縮伝送技術は、基本的にフルモーション (30 frame/sec) の自然画ビデオ映像を伝送するための技術であり、ある画素の輝度や色情報が変化した場合に、その変更をただちに遠隔地に伝送するようになっていることがあげられる。自然画像(動画像)のようなアナログ情報はランダムな画素が時間の経過とともに変化することが普通であり、つねにピクセル単位の粒度で差分比較することが必要である。このため、差分情報としてきわめて多くの情報をネットワーク伝送しなければならない。PC の VGA 出力をデジタルキャプチャした映像であっても、A/D 変換特性のばらつき等により微妙に輝度や座標が変わるため、自然画をソースとした映像とさほど変わらないほどのネットワーク帯域が必要となるのが実情である。

一方、PC のデスクトップ画面を転送する場合には、自然画像を転送する場合とは異なり、輝度や物体の位置がわずかに変化するという現象は発生しない。このため、MPEG2 のような微小ブロック単位の映像圧縮技術を適用するのはネットワーク帯域の無駄使いであることが多い。また、MPEG に採用されている動き補償フレーム間予測は、被写体がある程度の動きをとると仮定し、被写体の動き量だけずらして次のフレームの画像を予測する方法であるが、これは自然画像を前提としたものであって、PC のデスクトップ画面のように被写体に動きがない画像には効果を発揮しない。さらに、パワーポイントスライドや Web ページ等の画像を遠隔地に伝送するような場合には、厳格なリアルタイム性は要求されず、伝送量を増やして高速に差分情報を送りあうことを特徴とする映像伝送技術は必ずしも適さない。たとえば、講師がパワーポイントスライドを次の画面に進めたとき、その表示が 2、3 秒遅れたとしても、実質的な問題は発生しないことが普通であり、ゆるやかなリアルタイム性が確保されていればよい。

以上のように、従来の映像圧縮技術が少数人に高速に自然画動画像を伝送することを目的としているのに対し、本論文で提案する圧縮伝送方式は、デジタル情報を比較的低速で伝送すればよい場合に適した方式であり、使用ネットワーク伝送帯域を極力抑えてより多

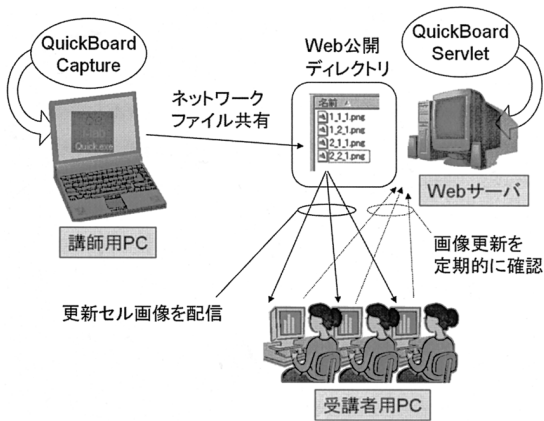


図 1 システム構成

Fig. 1 The system architecture.

くの人に必要なる情報を配信することを目的としている。なお、本論文の焦点は、QuickBoard システムに限らず、PC のデスクトップ画面やアプリケーション画面を準リアルタイムに配信する Web システムにおいて適用可能な伝送量圧縮手法を提案することにある。

### 2.3 提案方式

QuickBoard システムでは、講師用 PC のデスクトップ画面から生成されたスナップショット画像は、いったん、Web サーバ用 PC の公開ディレクトリに配置され、その後、受講者用 PC 上で動作する Web ブラウザに配信されるようになっている。このシステムにおいて、提案する伝送量圧縮方式「ICMHP」は、以下の基本機能を備えている。

- 講師用 PC のデスクトップ画面を複数の矩形領域（以下、「セル」）に画面分割し、それぞれのセルのスナップショット画像（以下、「セル画像」）を作成する。
- 画像更新があったセルのセル画像のみをネットワーク転送する。
- 受講者用 PC 上の Web ブラウザは、ネットワーク転送されたセル画像を HTML のテーブルに配置して表示する。

## 3. プロトタイプシステム

構築したプロトタイプシステムの全体構成を図 1 に示す。図 1 に示されるように、ICMHP の基本機能は、講師用 PC 上で動作する「QuickBoard Capture」アプリケーションと、Web サーバ（本実装においては Tomcat）上で動作する「QuickBoard Servlet」とが協調動作することによって実現されている。QuickBoard Capture は、Windows PC 上で動作するアプ

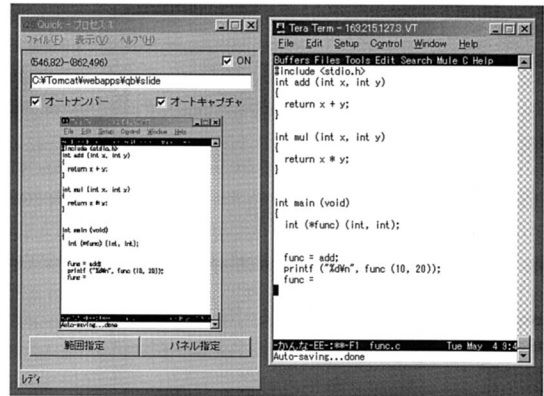


図 2 QuickBoard Capture の使用例

Fig. 2 An usage of QuickBoard Capture.

リケーションとして、また、QuickBoard Servlet は Java サブレットとして実装されている。以下、それぞれのコンポーネントについて説明する。

### 3.1 QuickBoard Capture

QuickBoard Capture は、講師用 PC のデスクトップ画面のスナップショットを、画像ファイルとしてファイル保存するソフトウェアである。図 2 は、QuickBoard Capture (左) と、講師が操作している Emacs ウィンドウ (右) とが並んで表示されている図である。

送出したいデスクトップ画面範囲を講師があらかじめ指定しておくこと、QuickBoard Capture は、その指定された画面範囲を複数のセルに自動分割し、それぞれのセルのスナップショットをセル画像として個別のファイルに書き出すようになっている。講師は、任意ウィンドウをマウスクリック指定する方法、または、任意矩形領域をラバーバンド指定する方法によってデスクトップ画面範囲を指定することができる。

いったん画面範囲が指定されると、QuickBoard Capture のオートキャプチャ機能によって、画面更新の有無（ピクセル色情報変化の有無）が定期的（本実装では 500 ms 間隔）に自動監視される。そして QuickBoard Capture は、画面更新があったと検知したセルのセル画像ファイルを選択的に作成する。

本実装の場合、セル画像を更新するタイミングにおいて値が 1 増えるカウンタを設け、セル画像のファイル名にこのカウンタ値を付加するようになっている。ファイル名は、< セル列番号 - セル行番号 \_ カウンタ値 .png > という規則で付けられている。すなわち、画面更新があったセル画像のファイル名は変化し、画面更新がなかったセル画像のファイル名は変化しない。なお、PNG (Portable Network Graphics) 画像圧縮<sup>12)</sup> は、画像の質を低下させない可逆圧縮でありな

```

<html><body>
<table cellSpacing=0 cellPadding=0 border=0>
<tr>
<td><img src='./1-1.png' name='_1_1' ></img></td>
<td><img src='./2-1.png' name='_2_1' ></img></td>
</tr><tr>
<td><img src='./1-2.png' name='_1_2' ></img></td>
<td><img src='./2-2.png' name='_2_2' ></img></td>
</tr>
</table></body></html>

```

図 3 自動生成された HTML ソース  
Fig. 3 Generated HTML source.

がら、PC 画面のスナップショットのような画像データをきわめて効率的に圧縮できるという特徴を有しており、Internet Explorer (以下「IE」) やネットスケープで表示できる画像フォーマットの 1 つである。

そして QuickBoard Capture は、各セル画像を HTML のテーブルにレイアウトするための HTML ファイルをあわせて自動作成する。図 3 は、QuickBoard Capture によって作成された HTML ファイルの例 (縦 2 分割 × 横 2 分割の例) である。<img>タグの src 属性によって各画像領域の画像ファイル名が示され、name 属性によってその画像領域のセル名 (列番号と行番号の組) が示されている。すべてのセル画像のカウンタ値は 1 に設定されている。HTML ファイルが作成されるのは、送出したいデスクトップ画面範囲を講師が指定したタイミングのみであり、画像更新のつど作成されるセル画像とは条件が異なる。

QuickBoard Capture によって作成された画像ファイルおよび HTML ファイルは、Windows のネットワークファイル共有の機能によって、即座に Web サーバに配置され、そして後述する QuickBoard Servlet によって、受講者の Web ブラウザに送信される (図 1 参照)。図 4 は、ICMHP 手法を用いて Web ブラウザに表示されたデスクトップ画面の例である。HTML のテーブルの属性が、<table cellSpacing=0 cellPadding=0 border=0> というようにテーブル要素間に隙間がないようになっているため、隣り合ったセル画像どうしがお互いに縫い目なく配置され、あたかも 1 枚の画像のように表示されている。領域分割されている様子を分かりやすく説明するために、<table>タグの border 属性を 2 に変更して Web ブラウザに表示させた例を図 5 に示す。

### 3.2 QuickBoard Servlet

QuickBoard Servlet は、Web サーバ上の Web 公開ディレクトリに存在するイメージデータを受講者用 PC にプッシュ送信する Web アプリケーションである。ただし実際は、HTTP プロトコルがプッシュ型配

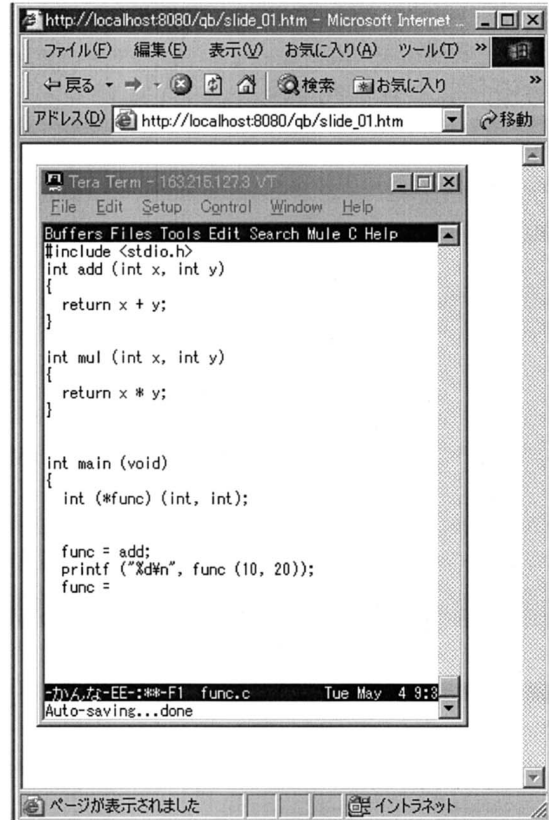


図 4 ICMHP 方式により伝送された講師の PC 画面  
Fig. 4 Lecturer's PC screen image sent with ICMHP method.

信をサポートしていないために、QuickBoard Servlet が能動的に複数の Web ブラウザにデータを送りつけることはできない。そこで QuickBoard では、画像更新があったかどうかを QuickBoard Servlet に定期的 (5 秒間隔) に問い合わせる小さな JavaScript プロシジャを受講者の Web ブラウザにダウンロードさせ、そのプロシジャによって Web ブラウザにポーリングさせるようになっている (図 1 参照)。このポーリング間隔は、画像更新遅延が 5 秒以下の場合には、ほぼ受講者全員が、講師が話している内容と表示画面とが一致していたと感じたという実験の結果に基づいて決定されている<sup>4)</sup>。

本実装においては、受講者の Web ブラウザが QuickBoard Servlet に画像更新の有無を問い合わせたときに、QuickBoard Servlet は画像更新があったセル画像のファイル名のリストを作成し、それに基づいて Web ブラウザにダウンロードさせるべき JavaScript を作成する。たとえば、図 3 の状態に対してセル (1,1) とセル (2,1) に画像変化があった場合には、以下のよう

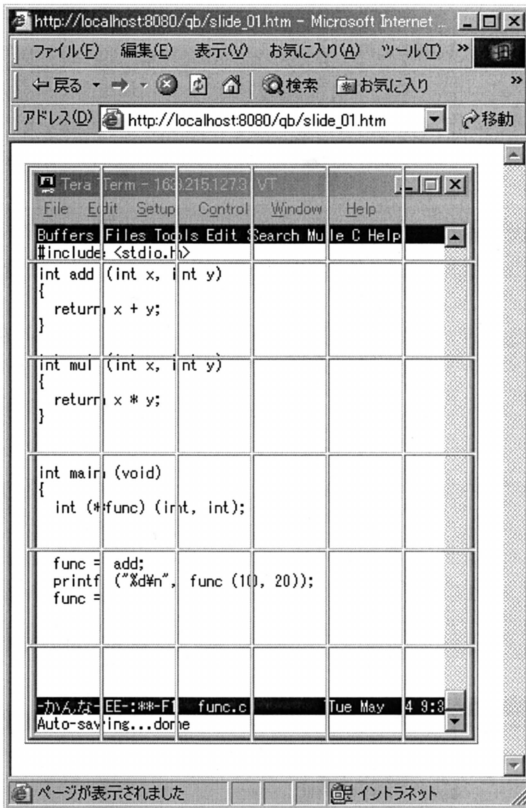


図 5 講師の PC 画面を border=2 に設定して表示

Fig. 5 Lecturer's PC screen image with the attribute of border = 2.

な JavaScript が作成される (画像更新がまったくなかった場合には, JavaScript は作成しない).

```
<script language="JavaScript">
  mainwindow.document.images["_1_1"].src
    = "/1-1_2.png";
  mainwindow.document.images["_2_1"].src
    = "/2-1_2.png";
</script>
```

そしてこの JavaScript をダウンロードした Web ブラウザは, HTML テーブル (mainwindow というウィンドウ内に表示されている) にレイアウトされて表示されているセル画像のうち, mainwindow 内の "\_1\_1" という名前を持ったイメージオブジェクトの内容を "/1-1\_2.png" の内容に差し替え, また, "\_2\_1" という名前を持ったイメージオブジェクトの内容を "/2-1\_2.png" の内容に差し替えるという働きをする. この操作は, Web サーバから新たに 1-1\_2.png という画像ファイルをダウンロード (図 1 参照) して, 座標 (1, 1) のセルの画像として表示し, 2-1\_2.png という画像ファイルをダウンロードして, 座標 (2, 1) の

セルの画像として表示するという DOM (Document Object Model) 操作に相当している.

## 4. 評価

### 4.1 性能評価実験

標準的な 640×480 ピクセル (VGA サイズ) のアプリケーションウィンドウ操作画面を, ICMHP 方式を用いて Web ブラウザに伝送する場合について, 必要伝送容量に関する評価を行った. VGA サイズのウィンドウ内で Emacs を起動し, C 言語のプログラムを表示させておき, ウィンドウを 36 個 (縦 6 分割×横 6 分割) のセル (1 セルの大きさは 106×80 ピクセル) に分割して ICMHP 方式で伝送する実験を行った. なお, ウィンドウ内に存在するオブジェクトは文字がほとんどであり, PNG 画像形式でスナップショット画像を保存するとファイルサイズはおよそ 30 KB である. なお, 36 個のセルに分割した理由については後述する.

ファイルサイズ 30.0 KB の PNG 画像を用意して実験した結果, 36 個のセルに分割した場合, 36 個の画像ヘッダが必要になることや画像圧縮効率が落ちる等の理由から, トータルのファイルサイズは 35 KB 程度 (1.167 倍) になった. この条件において画素変化が発生し, その画素変化が 1 つのセル領域内に限られたと仮定すると, 35 KB/36 (= 0.972 KB) だけの情報を伝送すればよいこととなる. 本システムの場合, 画像伝送の頻度は多くても 5 秒に 1 度であるので, 必要ネットワーク帯域はわずかに 0.194 KB/sec (1.56 Kbps) で済むことになる (ただし実際の HTTP 伝送では, 通信のたびに 50 バイト前後の HTTP ヘッダ情報が付加される). この結果によると, 標準的な MPEG2 over IP (6 Mbps) と比較すれば 3900 分の 1 以下のきわめて少ないネットワーク帯域でおさまることが分かる. 仮に, 画素変化が隣接する 4 つのセル領域にまたがっていたと仮定すると転送データ量は約 4 倍になるが, この場合にでも 6.24 Kbps のネットワーク帯域で済む. さらに, 本方式によって転送されるセル画像データは可逆圧縮されたデータであるため, MPEG2 等で送信した場合は異なり, 画像劣化はまったくない.

ただし, 画像をセル画像に分割して圧縮した場合の合計ファイルサイズは, 画像を分割せずに圧縮した場合と比較して増大すると予測される. そこで著者らは, 分割数を変化させてトータルファイルサイズを計測する実験を行った. 表 1 に実験結果を示す.

表 1 の実測データから分かるように, 画像をあまりにも多くのセル画像領域に分割することは, トータル

表 1 領域分割数とトータルファイルサイズの対応

Table 1 A comparison between the number of images and the total file size.

分割数(個)	トータルファイルサイズ (KB)
1	30.0
4	30.6
9	30.8
36	35.0
144	43.6
576	71.6
3600	241.7
14400	819.3
90000	4602.0

ファイルサイズの大幅な増加を引き起こす結果となる。このことから、MPEG のように微小ブロック (8×8 ピクセル) ごとに画像を分割し、それを HTML で記述したと仮定した場合には、伝送に必要な帯域がきわめて大きくなることは明らかである。

なお、本実装で述べた 36 分割という値は、分割数を変化させて Web ブラウザの処理遅延時間を測定した実験結果から得られた値である。ここで、この実験について述べる。著者らは、画像分割数を大きくすると、Web ブラウザの処理速度が低下する現象に遭遇した。Web ブラウザの画像読み込みスレッドの数が急増し、スレッド管理のためのオーバヘッドが増加したためと考えられる。そこで、分割数を徐々に変化させて、Web ブラウザの処理負荷による処理遅延時間を計測した。実験は、Pentium4-2.4 GHz PC 上で動作する Web サーバと Web ブラウザとを 100 Mbps LAN 内に設置して行った。VGA サイズのアプリケーションウィンドウ画面を転送するようにし、ICMHP を用いない方法 (無分割) で伝送した場合、サーバから画像を取得して Web ブラウザに表示するまでに 150 ms (10 回試行の平均値) かかることをあらかじめ調べておいた。そして、分割数を 100 分割, 200 分割, 400 分割, 1000 分割と変化させて、同時間を計測した (実験ステップ 1)。実験の結果を表 2 (a) に示す。この結果から、ここで実験したどの分割数においても、許容できる処理時間内に処理が終了しないことが分かった。受講者が許容できる遅延時間は、前記ポーリング間隔と処理遅延時間を足して 5 秒程度である。そこで次に、画像分割数を減らして実験を行った (実験ステップ 2)。実験の結果を表 2 (b) に示す。42 分割であれば、受講者が許容できる遅延時間内に収まる可能性がある。しかし、Web ブラウザを動かす PC の性能が、実験に用いた PC の性能より低い場合があることを考慮して、より安全な 36 分割を選定することとした。

表 2 領域分割数と処理遅延時間の対応

Table 2 A comparison between the number of images and the processing delay.

分割数(個)	処理遅延 (msec)
(無分割) 1	150
100	3,000
200	7,000
400	15,000
1000	30,000

(a) 実験ステップ 1

分割数(個)	処理遅延 (msec)
(6 行 x6 列) 36	200
(7 行 x6 列) 42	450
(7 行 x7 列) 49	750

(b) 実験ステップ 2

36 分割の場合の 200 ms という遅延時間は、無分割の場合の 150 ms と比較して 50 ms しか変わらず、問題とならない値である。

#### 4.2 サーバ負荷測定実験

続いて、Web サーバにかかる負荷を測定する実験を行った。通常 Web サーバには、スレッド同時実行数の上限を制御する機能 (最大スレッド数制限機能) が備わっており、最大スレッド数を超えた要求が到着した場合は、その要求は待ち行列に入らなくなる。そして、最大スレッド数を超えた接続要求が到着した場合は、その接続要求は待ち行列に入り、待ち行列の中のスレッドは同時実行スレッド数が減った瞬間に実行開始される。本実装で用いた Tomcat の場合もデフォルトの状態では並行実行スレッド数の上限は 50 と定められており、サーバに過剰な負荷がかからないようになっている。さらに、本システムの構成の場合、個々の Web ブラウザは別々のタイミングで 5 秒に 1 回 Web サーバ (QuickBoard Servlet) にアクセスするため、Web サーバへのアクセス要求は 5 秒間の間にほぼ均等に分散されると考えられる。よって、スレッド同時実行数の上限に到達する可能性はきわめて少ない。

想定システムレベルでの Web サーバ負荷を調べるために、PC 1 台あたり 10 個の Web ブラウザプロセスを動作させ (Web ブラウザのローカルキャッシュ機能はオフに設定)、10 台の PC を用いて QuickBoard Servlet に同時アクセスする実験を実施した (用いた PC はすべて Pentium4-2.4 GHz マシン)。この条件において、合計 100 の Web ブラウザプロセスが、QuickBoard Servlet に対して、画像更新があったかどうかをポーリングによってチェックする実験 (実験 A) と、画像が更新された場合にそのセル画像をサーバから取

得して表示するという実験（実験 B）を行い，Web サーバマシンの CPU 使用率を WindowsXP のタスクマネージャツールのログを用いて解析した．なお，実験開始前の状態における CPU 使用率の平均は 3% であった．

実験 A を 1 分間実施した結果，CPU 使用率は平均値 7.4%（最大値 11%）となった．平均値および最大値のいずれも Web サーバの稼動に支障をきたす値ではないことが確認できた．続いて実験 B の結果について述べる．実験 B は，C 言語のプログラミングを教えるという状況を模擬し，VGA サイズのウィンドウ内で Emacs を使用する条件で行った．そしてアプリケーションウィンドウを 36 個のセルに分割し 3 分間のデータを採取した．この結果，CPU 使用率は平均値 12.1%（最大値 17%）となり，平均値，最大値ともに Web サーバの負荷が問題になる値ではないことを確認した．

#### 4.3 実利用実験

続いて，実際の講義において ICMHP 方式の効果を測定する実験を行った．C 言語のプログラミングを教えるという場면을測定対象とし，前述した実験と同様に，講師は VGA サイズのウィンドウ内で Emacs を利用して講義を行った．QuickBoard Capture のオートキャプチャ機能を使い，講師はキーボードタイピングに集中できるようにした．また，前述の実験と同じく，VGA サイズのウィンドウは 36 分割されるように設定した．そして，この講義中において，Web ブラウザに画像転送が行われた 30 回分のデータを QuickBoard Capture のシステムログから採取した．

この実験の結果，36 分割された画像領域について，一度の伝送で Web ブラウザに送られた更新セル画像数は，平均 5.3 個（標準偏差 = 4.7）であった．1 セル画像のファイルサイズは，平均 972 バイト（= 35.0 KB/36）であったことから，一度の伝送で Web ブラウザに送られたセル画像の合計ファイルサイズの平均は 5.1 K バイトとなった．

ICMHP 方式を用いずに伝送した場合の平均は前述したとおり 30 KB であることから，ICMHP 方式の導入によって約 1/6 にデータ転送量が削減されたこととなる．なお，システムログから採取した 30 回の事象の中には，画面スクロール操作，または，複数行にわたるカットアンドペースト操作が 7 回含まれており，これらが平均更新画像領域数を 5.3 個に押し上げる結果となった．普段の講義においてこのように頻繁にカットアンドペースト操作が行われることはまれであり，このような条件下でありながら，データ転送量

を約 1/6 に削減できたことから，ICMHP の効果が示されたと考えられる．

#### 5. おわりに

本論文では，Web ページ伝送における画像伝送容量を減らすために，HTML のテーブル要素を用いて画像分割し，動画圧縮伝送技術と静止画圧縮伝送技術とを組み合わせる方式を提案した．この技術は，デジタル情報を比較的低速で伝送すればよい場合に適用できる方式であり，使用ネットワーク伝送帯域を極力抑え，より多くの人に必要な情報を配信するために適した技術と位置づけられる．

本論文で提案した圧縮伝送方式は，HTML 標準規格のテーブルタグを使用するため，プラグイン等の追加クライアントソフトウェアを導入する必要がないことが特徴である．また，きわめて標準的な HTTP 通信のみで Web ブラウザに情報を配信し，かつ，Web ブラウザに専用プラグインを必要としないため，PDA に搭載されている簡易な Web ブラウザ等にも情報配信が可能である．非常に安価に準リアルタイムな Web サービスを構成することが可能であるため，E-ラーニングをはじめ，遠隔セミナー，オンラインヘルプデスク，遠隔ヘルスケア，遠隔故障診断，監視カメラ等の幅広い分野に適用可能と考えられる．

#### 参考文献

- 1) Stefik, M., Foster, G., Bobrow, D.G., et al.: Beyond the Chalkboard: Computer Supported for Collaboration and Problem Solving in Meetings, *Comm. ACM* (Jan. 1987).
- 2) Crowley, T., Milazzo, P., Baker, E., Forsdick, H. and Tomlinson, R.: MMConf — An Infrastructure for Building Shared Multimedia Applications, *Proc. ACM CSCW'90*, pp.329–342 (1990).
- 3) 市村, 宇田, 伊藤, 田胡, 松下: QuickBoard: 汎用 Web ブラウザのためのリアルタイムスライド配信サービスの試作, 情報処理学会グループウェアとネットワークサービス研究会報告, GN-46-7, pp.37–42 (2003).
- 4) 市村, 中村, 伊藤, 宇田, 田胡, 松下: QuickBoard: 任意のアプリケーション画面をリアルタイム配信する Web システム, 情報処理学会論文誌, Vol.45, No.1, pp.244–254 (2004).
- 5) 田中, 勅使河原: Web ページやその部分構成要素のリアルタイム共有・記録機能の設計と開発, 情報処理学会第 65 回全国大会第 4 分冊, pp.21–22 (2003).
- 6) Greenberg, S. and Roseman, M.: Group-



Web: A WebBrowser as Real-Time Groupware, *Proc. ACM SIGCHI '96 Conference Companion*, pp.271-272 (1996).

- 7) Kobayashi, M., Shinozaki, M., Sakairi, T., Touma, M., Daijavad, S. and Wolf, C.: Collaborative Customer Services Using Synchronous Web Browser Sharing, *Proc. ACM CSCW'98*, pp.99-108 (1998).
- 8) <http://www.microsoft.com/windowsmedia/>
- 9) <http://www.microsoft.com/windows/netmeeting/>
- 10) <http://www.uk.research.att.com/vnc/>
- 11) Rao, K.R. ほか: デジタル放送・インターネットのための情報圧縮技術, pp.200-237, 共立出版 (1999).
- 12) <http://www.libpng.org/pub/png/>

(平成 16 年 5 月 7 日受付)

(平成 16 年 11 月 1 日採録)



市村 哲 (正会員)

1989 年慶應義塾大学理工学部計測工学科卒業。1994 年同大学大学院理工学研究科博士後期課程修了。博士 (工学)。同年富士ゼロックス (株) 入社。1997 年～1999 年富士ゼロックスパロアルト研究所駐在。2002 年 4 月より東京工科大学助教授。グループウェア, ネットワークサービス, 生体情報活用等の研究に従事。『IT TEXT 基礎 Web 技術』, 『IT TEXT 応用 Web 技術』(オーム社)。DICOMO 2003 優秀論文賞受賞。ACM, 電子情報通信学会各会員。



間下 直晃 (正会員)

2000 年慶應義塾大学理工学部情報工学科卒業。2002 年同大学大学院理工学研究科前期博士課程修了。現在同大学院後期博士課程在学。1998 年株式会社ブイキューブ設立。2004 年株式会社ブイキューブブロードコミュニケーション設立。現在両社の代表取締役兼最高経営責任者。大学・ビジネスの両方で, 遠隔教育や遠隔コミュニケーションについての研究に従事。



伊藤 雅仁 (正会員)

1998 年慶應義塾大学理工学部計測工学科卒業。2000 年同大学大学院理工学研究科修士課程修了。2003 年同大学院理工学研究科後期博士課程修了。博士 (工学)。現在, 東京工科大学コンピュータサイエンス学部専任講師。モバイル・コンピューティング, 情報家電, デジタル放送の研究に従事。2000 年情報処理学会高度交通システム研究会優秀研究報告賞受賞。IEEE, 電子情報通信学会各会員。



宇田 隆哉 (正会員)

1998 年慶應義塾大学理工学部計測工学科卒業。2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了。2002 年同大学大学院理工学研究科開放環境科学専攻後期博士課程修了。2003 年 4 月より東京工科大学コンピュータサイエンス学部専任講師。博士 (工学)。ネットワークセキュリティの研究に従事。2002 年 IFIP/SEC2002 Best Student Paper Award 受賞。電子情報通信学会会員。



田胡 和哉 (正会員)

1986 年筑波大学大学院工学研究科博士課程修了。工学博士。筑波大学電子情報工学系助手, 東京大学工学部助手, 日本 IBM 東京基礎研究所を経て 2003 年より東京工科大学コンピュータサイエンス学部助教授。オペレーティングシステムの構成方式に興味を持つ。1984 年情報処理学会論文賞受賞。ACM 会員。



## 松下 温 (フェロー)

1963 年慶應義塾大学工学部電気工学科卒業, 1968 年イリノイ大大学院コンピュータサイエンス専攻修了, 1989 年より 2002 年 3 月まで慶應義塾大学理工学部教授, 2002 年 4

月より東京工科大学教授および慶應義塾大学理工学部客員教授, 2003 年 4 月より東京工科大学コンピュータサイエンス学部長, 教授および慶應義塾大学理工学部客員教授. マルチメディア通信, コンピュータネットワーク, グループウェア等の研究に従事, 情報処理学会理事, 同学会副会長, マルチメディア通信と分散処理研究会委員長, グループウェア研究会委員長, 電子情報通信学会情報ネットワーク研究会委員長, MIS 研究会委員長, パーチャルリアリティ学会サイバースペースと仮想都市研究会委員長, 情報処理学会 ITS 研究会委員長等を歴任. 郵政省, 通産省, 農水省, 建設省, 都市基盤整備公団, 行政情報システム研究所等の委員長, 座長, 委員を多数歴任. 特に国土交通省, 住宅情報化標準策定委員会委員長, 経済産業省総合エネルギー調査会電子計算機と磁気ディスク委員会委員長を務める. 現在, 経済産業省総合エネルギー調査会 “ルータ装置基準委員会” 委員長, 最高裁判所専門委員. 『やさしい LAN の知識』(オーム社), 『201x 年の世界』(共立出版) 等著書多数, 1993 年情報処理学会ベストオーサ賞, 1995 年および 2000 年情報処理学会論文賞, 2000 年 10 月 20 日情報処理学会 40 周年記念 90 年代学会誌論文賞, 2000 年 10 月 2 日電子情報通信学会フェロー, 2000 年 10 月 VR 学会サイバースペース研究賞, 2001 年 5 月情報処理学会功績賞, 2002 年 3 月情報処理学会フェロー, 電情報通信学会, 人工知能学会, ファジイ学会, IEEE, ACM 各会員.

---