

講義画面の自動連続キャプチャによる 講義・学習支援

古井 陽之助[†]

PC とスクリーンを使って講義を行う場合について、教員と受講者の双方を支援することを目的として、本研究では新しいシステムを開発している。このシステムは、教員の PC を監視し、あらかじめ指定された条件に合うスクリーンショットを自動的に Web サーバに転送する。受講者はこれを Web ブラウザ上で閲覧することができる。これにより、教員は手間をかけずにノイズの少ない講義の記録を受講者に提供することができる。

Supporting education and learning by taking continuous screenshots of the teacher's PC

Younosuke Furui[†]

This paper introduces a new system that aims at supporting both teachers and students. A client component of the system watches a teacher's PC, captures the screen shots when a certain condition is satisfied, and uploads the image data to the server part of the system. The students can browse the screen shots on the Web. This system allows the teachers to easily create the record of each lesson with less noise data.

1. はじめに

今日、大学等において教員が PC の画面表示を液晶プロジェクタによってスクリーンに投影し、受講者にそれを見せながら講義を進めるというやり方はごく一般的である。本稿では、教員の PC の画面に表示されるスライド資料やソフトウェアの実行状況を自動的に記録して受講者に提供するシステムと、それによる講義・学習支援について述べる。

このシステムは、事前に設定された条件に基づいて記録対象のソフトウェアを自動的に選択し、また画面表示の変化を検出して、変化が生じた瞬間の画面全体を静止画像（スクリーンショット）として記録（キャプチャ）し、その画像を Web サーバに転送する。設定条件を調整して講義内容とは直接関係のない表示を自動的に省くことができるので、教員は手間をかけずにノイズの少ない講義の記録を作ることができる。

受講者は自分の PC で Web ブラウザを実行して Web サーバにアクセスし、次々に蓄積されていくスクリーンショットを閲覧することができる。スクリーンショットの一覧は動画像よりも探索が容易である。

2. 研究の背景

以下では、プロジェクタを用いて PC 画面をスクリーンに投影しながら進めるスタイルの講義や、そうした講義を録画するシステムについて、より詳しく述べる。

2.1 講義における PC とプロジェクタの使用

スクリーンに投影する内容の典型的な例としては、Microsoft Office PowerPoint や OpenOffice.org Impress などのプレゼンテーション用ソフトウェアによるスライドショー、Adobe Reader のフルスクリーンモード、Web ブラウザウインドウなどを挙げるることができる。また、講義の種類に応じて特有のソフトウェアを用いることもある。例えば、プログラミング系の講義において Eclipse や Microsoft Visual Studio などの開発用ソフトウェアを用いて、受講生の目の前で実際にプログラムを動かしてみせる場合などがある。

このような講義の進め方の利点としては、スクリーンの表示内容は活字で清書された状態になっていて読みやすいこと、教員にとっては図解やデモンストレーションを見せながら解りやすく説明できること、また同じ資料に改訂を加えながら使い回すことができることなどが挙げられる。

一方、スクリーンに投影される情報の時間あたりの量は、手書きの板書よりも多く、

[†]九州産業大学 情報科学部 情報科学科

Department of Information Science, Faculty of Information Science, Kyushu Sangyo University

また図解やデモンストレーションなど文字以外の情報が多くなりがちであるため、受講者がノートをとることは困難な場合もある。

このような問題は、教員が受講者に資料を配布することによってある程度解消される。PowerPoint等のプレゼンテーション用ソフトウェアで作成したスライド資料から配布資料を作るのは比較的容易である。しかし、講義中にスライド資料以外に他のソフトウェアを用いる場合、そうしたソフトウェアの実行画面をスライド資料に織り交ぜて配布資料を作るのは却って面倒になることもある。

2.2 講義の記録

近年では講義の様相を録画するシステムの導入が進みつつある。受講者は講義終了後に講義録画を視聴して、復習に役立てたり、欠席等によりリアルタイムでは受講しなかった講義の内容を学んだりすることができる。

筆者らの所属する九州産業大学情報科学部にも「講義記録システム」が導入されている[1][2]。このシステムを学生は予習・復習に、教員は教育改善活動のために利用する。情報科学部の学生全員は大学から貸与されたノートPCを持っているので、これでWebブラウザを実行して講義記録システムのWebサイトにアクセスし、講義録画を視聴することができる。

講義記録システムは、教壇に立つ教員が口頭説明や板書をする様子のほか、プロジェクトに出力されているPCの画面表示を録画することができる。後者の場合、PCの画面に表示された内容は全て録画に含まれる。つまり、スライド資料に含まれないソフトウェアの実行画面等も講義録画には含まれるので、そうした画面表示を受講者はあとで閲覧することが可能となっている。

逆に言えば、教員が受講者に見せることを意図しなかった画面表示（例えば、スライド資料と同じフォルダにたまたま入っていた他のファイルの一覧など）も講義録画には含まれる。この問題は、教員のPCをデュアルディスプレイにすることによってある程度解消できるが、搭載資源の限られているノートPCではその限りでないこともある。

一方、講義録画のような動画像では、全体の一覧性が低く、特定の箇所を頭出しするのが面倒なことがある。講義記録システムの講義録画にインデックスをつける仕組みについては研究[3][4]も行われているが、まだ実用化には至っていない。

2.3 現状の問題点

ここまで述べてきた現状について、その問題点を次にまとめる。

[問題 1] プレゼンテーション形式の講義では受講者はノートをとりにくい状況に陥ることがある。これを解消する上で配布資料は有効ではあるが、プレゼンテーシ

ョン用ソフトウェアで作成したスライド資料と、それ以外のソフトウェアの実行画面を織り交ぜた配布資料を用意するのは面倒である。

[問題 2] 講義記録システムの講義録画は[問題 1]をある程度まで解消するが、講義録画には講義に直接関係のない画面表示も含まれる。

[問題 3] 講義録画は全体の一覧性が低く、講義録画にインデックス付けをする方法もまだ実用化には至っていない。

3. 研究の方針

2.3節で述べた問題点を解決するため、本研究では新たにシステムを開発することにした。以下では、このシステムが備えるべき特徴と、それらの特徴による問題の解決について述べる。

[特徴 1] 教員のPCが特定の条件を満たしている場合にのみ、画面表示の変化を監視する。この特徴は[問題 2]の解決に役立つ。条件の例としては、プレゼンテーション用ソフトウェアがスライドショーを実行中であることや、教員が受講者に実行の様子を見せたいソフトウェアが画面を占有していることなどが挙げられる。

[特徴 2] 画面表示の変化を検出すると、自動的にその瞬間の画面全体を静止画像（以下、**スクリーンショット**）として記録（以下、**キャプチャ**）し、さらに受講者が閲覧できるところに蓄積する。この特徴は[問題 1]の解決に役立つ。

[特徴 3] 受講者は自分のPCを使い、スクリーンショットの一覧を見たり、スクリーンショットそのものを閲覧したりすることができる。この特徴は[問題 1]および[問題 3]の解決に役立つ。

ただし、本稿執筆時点では講義録画のインデックス機能までを実装するには至っておらず、これは今後の課題である。

4. 開発したシステム

本研究において開発したシステム（以下、**本システム**）は教員側クライアント（以下、**Gear**）、サーバ（以下、**Server**）、ビューア（以下、**pViewer**）という三つの部分からなる。以下では本システムの設計、実装等について述べる。

4.1 システムの構成

本システムの構成を図1に示す。Gearは教員のPCにおいて実行する。教員がサー

バへのログイン操作を行い、また PowerPoint のスライドショーを使用するなどしてキャプチャの条件が満たされると、Gear はスクリーンショットをキャプチャし、その PNG 画像をキャプチャ時のタイムスタンプとともに Server に転送する。Server は Gear から送られる PNG 画像とタイムスタンプを受け取り、ファイルシステムに保存する。

一方、受講者が Web ブラウザで pViewer にアクセスすると、Web ブラウザに読み込まれた pViewer が実行を開始し、Server からスクリーンショットとタイムスタンプを取得して Web ブラウザウインドウ内に表示する。pViewer は一定時間ごとに Server から最新のデータを取得し、表示を更新する。

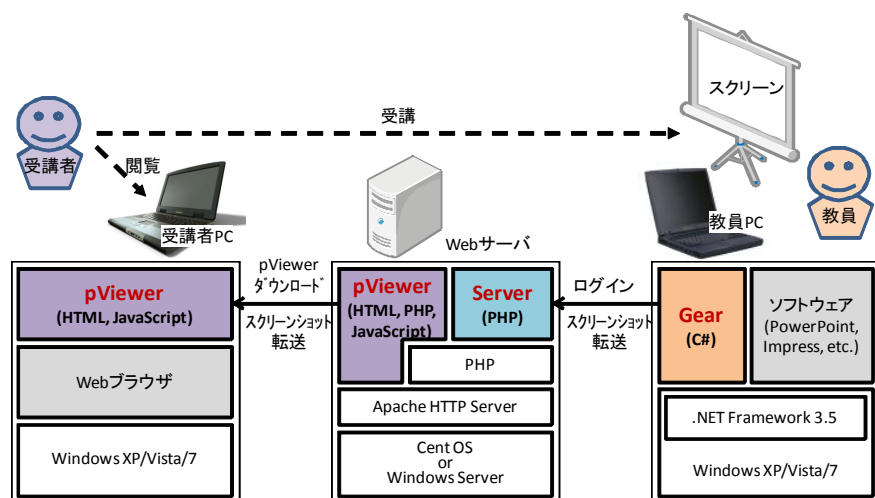


図 1 システム構成図

4.2 使用技術

本研究でシステムを開発するにあたって使用したソフトウェアを表 1~3 に示す。Gear の開発には Visual C#を用い、動作確認には Windows XP と Windows 7 を用いた。Server は PHP スクリプトとして作成した。また、pViewer の主要部分は HTML と JavaScript で記述し、その実行のために Web サーバ側で必要な処理は Server と同様に PHP スクリプトとして作成した。なお、PHP による Web アプリケーション開発では MySQL 等のデータベース管理システムとの組み合わせがよく知られているが、本システムではデータベースを用いていない。

4.3 教員側クライアント (Gear)

教員側クライアント(Gear)が表示するウインドウの概要を 4.3.1~4.3.2 項に示す。また、Gear の実装における主要な工夫点を 4.3.3~4.3.6 項に示す。

表 1 教員側クライアント(Gear)の開発に用いたソフトウェア

ソフトウェア名称	説明
Microsoft Windows XP Professional SP3	教員 PC に用いられることの多い OS の一つ。開発および実行確認に用いた。
Microsoft Windows 7 Professional	教員 PC に用いられることの多い OS の一つ。実行確認に用いた。
Microsoft Visual Studio 2008 Professional Edition	C#の開発環境。
Microsoft .NET Framework 3.5 SP1	C#の実行環境。

表 2 サーバ(Server)およびビューア(pViewer)の開発に用いたソフトウェア

ソフトウェア名称	説明
CentOS 5.3 (Linux 2.6.18)	PHP スクリプトの実行確認のために使用した。下の Windows Server の環境と併用した。
Apache HTTP Server 2.2.13	
PHP 5.3.0	PHP スクリプトの実行確認のために使用した。XAMPP には Apache HTTP Server 2.2.14 と PHP 5.3.1 が含まれる。
Microsoft Windows Server 2003 R2 Standard Edition SP2	
XAMPP (Basis Package) 1.7.3	
TeraPad 0.93	

表 3 ビューアの実行確認に用いたソフトウェア

ソフトウェア名称	説明
Microsoft Windows 7 Professional	受講者 PC に用いられることの多い OS の一つ。
Mozilla Firefox 3.6.13	受講者 PC で用いられることの多い Web ブラウザ。それぞれ異なる HTML レンダリングエンジンおよび JavaScript エンジンを搭載している。
Microsoft Internet Explorer 8.0	
Google Chrome 9.0	

4.3.1 Gear ウインドウの構成

Gear の実行中に表示されるウインドウ (以下、Gear ウインドウ) の構成を図 2 に示す。このウインドウは通常は非表示になっており、マウスポインタが画面の右端にあるときのみ画面上に現れる。デフォルトの表示位置は画面の右端であり、必ず Z 順

の最上位（使用者から見て他のウィンドウよりも手前）である。

Gear ウィンドウ全体は縦長で、下部には使用者が Gear の機能を用いるためのボタンが集まっており、上部にはスクリーンショットの縮小画像が上から順に左端だけ見えている状態で並ぶ。縮小画像のいずれかにマウスポインタをかざすと、その画像だけが左にスライドする（図3）。縮小画像をクリックすると、元のスクリーンショットが表示される（図4）。スクリーンショットの上端にはタイムスタンプ、移動ボタン（[<<]と[>>]）、スクリーンショットの番号、[Close]ボタンが表示され、実際の画面と区別できるようにになっている。

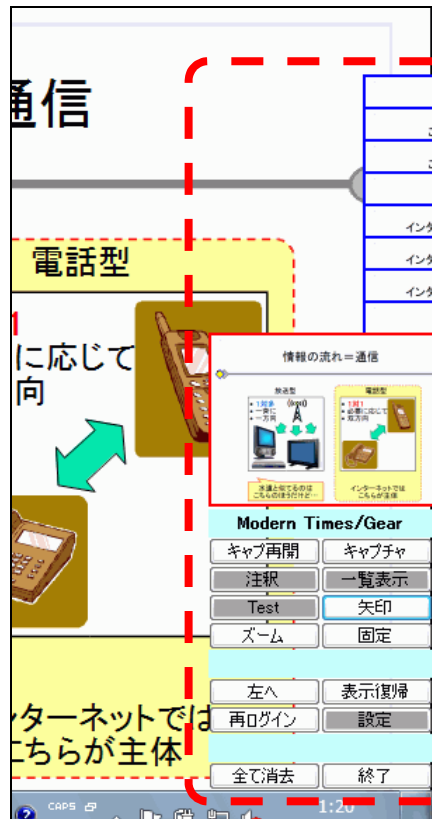


図2 Gearのウィンドウ構成

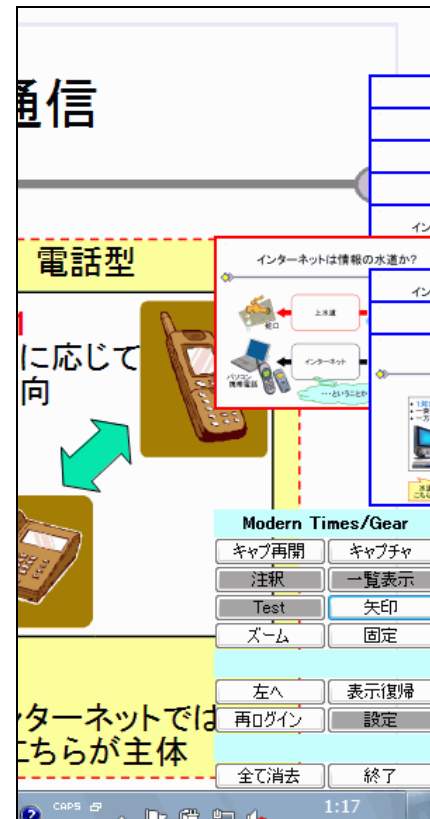


図3 縮小画像の選択

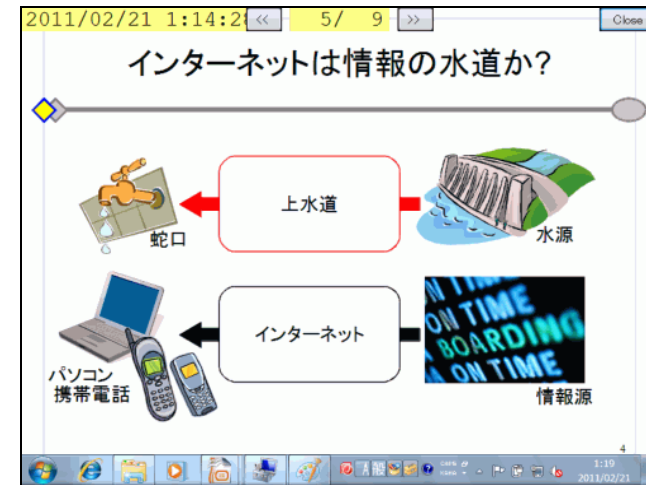


図4 スクリーンショットの表示（上端にボタン等がある）

4.3.2 Gear ウィンドウの機能

Gear ウィンドウの下部に並んでいるボタンのうち、スクリーンショットに関連するものだけを表4に挙げる。

表4 Gear ウィンドウ下部にあるボタンの名称と機能

ボタン名称	機能
キャプチャ	画面を手動でキャプチャする。
キャプ停止	画面の自動連続キャプチャ機能を一時停止する。
ログイン	ユーザ名、パスワード、パッケージ名（4.4節を参照）等を入力してServerにログインする（図5）。スクリーンショットの転送が開始する。
全て消去	Gear ウィンドウ上部の縮小画像を全て消去する。
終了	Gearの実行を終了する。

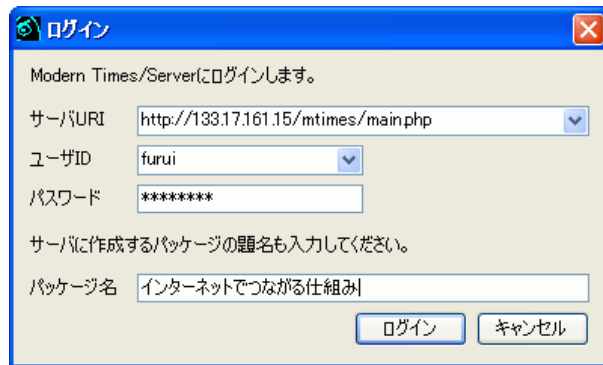


図 5 Gear のログインプロンプト

4.3.3 Gear ウィンドウ自体のキャプチャ抑制

Gear が画面表示をキャプチャする際、Gear ウィンドウが本来の画面表示よりも手前にある状態でスクリーンショットの一部としてキャプチャされることは避ける必要がある。Windows XP では透過色の用いられたウィンドウはキャプチャ画像に含まれないので、これを利用して、Gear ウィンドウには透過色を用い、Gear ウィンドウ自体がキャプチャされるのを防いでいる。

Windows Vista/7 では、Windows Aero[5]を実現する Desktop Window Manager[6]の働きにより、透過色の用いられたウィンドウであってもキャプチャされる。したがって、Windows Vista/7 で Gear ウィンドウのキャプチャを防ぐためには、Desktop Window Manager を無効にしなければならない。具体的には、Desktop Window Manager をサービスの一覧から選んで停止する方法がある。他に、Win32 API を呼び出して一時的に無効にする方法[7]もあるが、この方法は現行の Gear には未実装である。

4.3.4 キャプチャ条件の検査

第 3 章にて[特徴 1]として述べた通り、Gear は「特定のソフトウェアが特定のモードで実行している」といったような複数の条件（以下、**キャプチャ条件**）を常に検査し、その一つでも満たされたときに連続キャプチャを行う。現行の Gear におけるキャプチャ条件を表 5 に示す。これらのキャプチャ条件は、人間が見ればすぐにそれと見分けられる。しかし、プログラムで検査するには、他のソフトウェアや他の状態と機械的に区別できるような特徴がなければならない。以下では、表 5 の各キャプチャ条件を検査するために本研究で Gear に実装している方法について述べる。これらの検査方法はソフトウェアのバージョンに依存しないことが望ましいが、現実には各項目で

述べる通りバージョンによって異なる検査方法を用いている。

表 5 キャプチャ条件

	ソフトウェア	状態
(1)	Microsoft PowerPoint 2000/2003/2007/2010	スライドショー
(2)	OpenOffice.org Impress 2.4	スライドショー
(3)	OpenOffice.org Impress 3.x	スライドショー
(4)	Adobe Reader 9.x	フルスクリーンモード
(5)	Adobe Reader 10.x	フルスクリーンモード
(6)	Eclipse 3.x Java パースペクティブ	ウインドウ最大化

(1) Microsoft PowerPoint 2000/2003/2007/2010 スライドショー

このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。

- 「PowerPoint スライドショー」から始まるタイトルを持つ。
 2000～2010 の PowerPoint はいずれも、スライドショーを実行する際には編集ウィンドウとは別にスライドショーのウィンドウを表示する。このウィンドウは必ず「PowerPoint スライドショー」という文字列から始まるタイトルを持つ。このタイトルは、スライドショーの実行中に[Alt]-[Tab]でウインドウ一覧を表示することによって確認できる。
- (2) **OpenOffice.org Impress 2.4 スライドショー**
 このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。
 - 「OpenOffice.org Impress」を含むタイトルを持つ。
 - 最大化されていない。
 - それにもかかわらず大きさは画面と同じである。

Impress も PowerPoint と同様に、編集ウィンドウとは別にスライドショーのウィンドウを表示する。このウィンドウのタイトルは必ず編集ウィンドウと同じなので、タイトルだけでは編集ウィンドウと区別がつかない。しかし、スライドショーのウィンドウの大きさは画面と同じ大きさであり、なおかつ最大化はされていないので、これを利用する。

なお、Impress 2.4 のスライドショーのウィンドウは Z 順で必ず最上位になるよう設定されている。Gear ウィンドウがこのウィンドウの下になるのを避けるため、Gear は Impress 2.4 のスライドショーを検出するとそのウィンドウの Z 順を変更する。

(3) OpenOffice.org Impress 3.x スライドショー

このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。

- タイトルが空である。
- 最大化されていない。
- それにもかかわらず大きさは画面と同じである。
- ウインドウクラス名が「SALTMPSUBFRAME」である。

OpenOffice.org 3.1 および 3.3 について調べたところ、2.4 と異なり、スライドショーのウインドウにはタイトルがなかった。タイトルの代わりにウインドウを表示しているプロセスの特徴ないし実行ファイルのパス名を調べるといった方法が考えられるが、これらの方法ではウインドウ一覧の検査に加えてプロセスの一覧の検索までも行うことになるので、キャプチャ条件の検査処理に時間がかかるようになることが懸念される。そこで、現行の Gear では Win32 API で規定されているウインドウクラス名を調べるようにした。

なお、ウインドウクラス名の「SALTMPSUBFRAME」は OpenOffice.org 2.4 でも同じであるので、これを利用すれば(2)と(3)のキャプチャ条件は統合することも可能であるが、今後のバージョンにおいてさらに変更がある可能性も考慮して、本稿執筆時点では統合を見送った。

(4) Adobe Reader 9.x フルスクリーンモード

このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。

- 「Adobe Reader」で終わるタイトルを持つ。
- 最大化されていない。
- それにもかかわらず大きさは画面と同じである。

(5) Adobe Reader 10.x フルスクリーンモード

このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。

- 「Adobe Reader」で終わるタイトルを持つ。
- 最大化されていない。
- 大きさは画面より縦に 2 ピクセルだけ大きい。上端は画面より 1 ピクセルだけ上で、下端は画面より 1 ピクセルだけ下である。

(6) Eclipse 3.x Java パースペクティブ ウインドウ最大化

このキャプチャ条件を検査するには、Windows のウインドウ一覧の中に次の条件を満たすウインドウが存在するかどうかを調べる。

- 「Java」を含み「Eclipse」で終わるタイトルを持つ。
- 最大化されている。
- 使用者のキー入力を受け付ける状態(フォアグラウンド)である。

なお、現行の Gear においては、これら(1)~(6)の条件はプログラム中にハードコーディングされている。これをパラメータ化して、Gear の使用者が自由に条件を追加しない変更できるようにすることは今後の課題の一つである。

4.3.5 Gear によるキャプチャの範囲

Gear は次項に述べる仕組みにより画面表示の変化の監視やスクリーンショットのキャプチャを行う。このとき、監視やキャプチャの対象とする画面の範囲は、厳密に画面全体ではなく、Windows のタスクバーを除いた範囲(ワーキングエリア)である。タスクバーとは、通常は Windows の画面の下端に表示されている、横に細長いツールバーである。タスクバーの表示は、使用者がその時点で主に使用しているソフトウェアの状態とは無関係に変わることがあるので、これを監視の対象範囲に含めるのは適切ではない。しかし、キャプチャの対象範囲からタスクバーの部分を外すと、スライドショーをキャプチャした画像は下端が途切れているように見えることがある。これについては今後の検討が必要である。

なお、タスクバーのプロパティで「自動的に隠す」という設定を有効にしておく、タスクバーは画面の下に隠れてしまい、ワーキングエリアと画面全体の範囲は同一になる。

タスクバーとは別の問題として、PowerPoint のスライドショーの実行中には、スライドの左下隅にペンや矢印などのアイコンが表示され、マウスポインタの動きに合わせて表示・非表示の変化を繰り返す。このアイコンの変化を誤って検出することを避けるため、現行の Gear では画面の左下隅を監視の対象範囲から除いている。

4.3.6 連続キャプチャ

第 3 章にて[特徴 1]および[特徴 2]として述べた通り、Gear は画面の変化を監視し、変化を検出したらスクリーンショットをキャプチャし、PNG 画像として保存する。検出方法の基本的な考え方は、短い時間間隔(現行の Gear では 30 ミリ秒)で画面をキャプチャし、その時点のスクリーンショットと最後に保存したスクリーンショットを比較するというものである。しかし、画面全体のキャプチャは負荷の高い処理であり、これを短い時間間隔で繰り返すと、受講者に本来見せるべき他のソフトウェアの実行効率に影響を与える。したがって、Gear が画面の変化を検出する処理を可能な限り負

荷の低く抑えることは重要である。

本研究では、画面全体ではなく縦 1 ピクセルだけの横長の領域を標本としてキャプチャし、これを最後に保存したスクリーンショットと比較するという方式によって負荷を抑えることにした (図 6)。

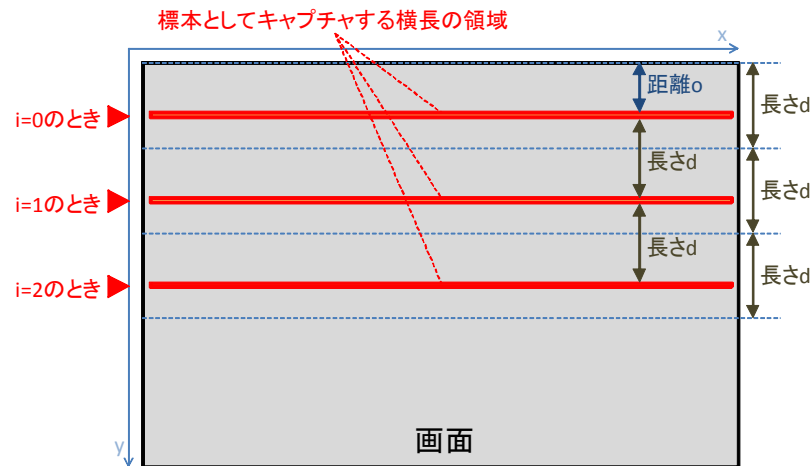


図 6 画面の走査

キャプチャする領域は常に移動させて、一定の時間があれば画面全体をまんべんなく監視できるようにする。この処理の手順は次の通りである。

[手順 1] 画面の縦の長さを一定の整数 c (例えば $c=30$) で割った長さ d を求める。また、長さ d に対して約 3 分の 2 にあたる素数 p を求める。 c と d は互いに素である。

[手順 2] 画面の上端からの距離 o を次式で更新する。ただし、距離 o の初期値は 0 とする。

$$o \leftarrow (o + p) \bmod d$$

[手順 3] 横長の領域の y 座標を次式で計算する。ただし、変数 i の初期値は 0 とする。

$$y \leftarrow o + i \times d$$

[手順 4] 座標 $(0, y)$ を左上頂点とする横長の領域をキャプチャする。

[手順 5] 最後に保存したスクリーンショットの当該領域と比較し、違いがなければ

次の[手順 6]に進む。もし違いがあれば、検出処理を終了して画面全体のキャプチャ処理に移る。

[手順 6] 変数 i を 1 増やす。 $i < c$ ならば[手順 3]に戻って繰り返す。そうでなければ i を 0 にクリアした上で[手順 2]に戻って繰り返す。

この手順により、横長の領域の y 座標は間隔 d だけ増えながら画面の上から下へと移動し、下に到達すると上に戻る。また、素数を用いて画面上端からの距離 o に変化をつけるので、画面全体を走査し終わるまでは同じ y 座標が 2 回以上用いられることはない。

Gear が画面全体を一通り走査するのに要する時間は、画面の縦が 768 ピクセル、時間間隔が 30 ミリ秒の場合に 23 秒である。実際には、スライドショーの実行中にスライドを切り替えると、Gear はほとんど即座にそれを検出する。スライドの一部だけでアニメーション機能などによる変化が生じたときは、ワンテンポ遅れてキャプチャする様子が見られることもあるが、23 秒もかかることはない。

4.4 サーバ (Server)

本システムのサーバ(Server)は、Gear から受信したスクリーンショットの PNG 画像とタイムスタンプを Web サーバコンピュータ内に蓄積する。また、受講生の使用するビューア(pViewer)に PNG 画像やタイムスタンプを送信する。

Server は、一回の講義において Gear から受信するデータをひとまとまりとして扱うため、これを**パッケージ**という単位で扱う。図 5 に示したように、教員は Gear から Server にログインする際にパッケージ名を入力する。ログイン情報とともにパッケージ名を Server が受信すると、Server は空の新しいパッケージを作成する。これ以降、Gear から Server が受信するデータは全てこのパッケージに追加される。追加先を別の新しいパッケージに切り替えるには、Gear 側で再度ログイン操作を行う。

4.5 ビューア (pViewer)

受講者が Web ブラウザで Web サーバ上にある本システムのビューア(pViewer)にアクセスし、ログインおよびパッケージの選択を行うと、HTML と JavaScript で記述された pViewer の主要部分が Web ブラウザに読み込まれて実行を開始する (図 7)。Web ブラウザ上の pViewer は一定時間おきに、Web サーバ上にある PHP スクリプトにスクリーンショットとタイムスタンプの CSV データを要求する。Web ブラウザ上の「Keep seeing the latest」というチェックボックスにチェックが入っていると、pViewer は常に最新のスクリーンショットを表示する。また、受講者が Web ブラウザ上のボタンやアンカーをクリックすると、それに応じてスクリーンショットを切り替える。現行の pViewer はスクリーンショットの一覧をタイムスタンプでのみ表示するが、今後は縮

小画像の表示や、類似度の高いスクリーンショットのクラスタリングなどの機能も実装する予定である。

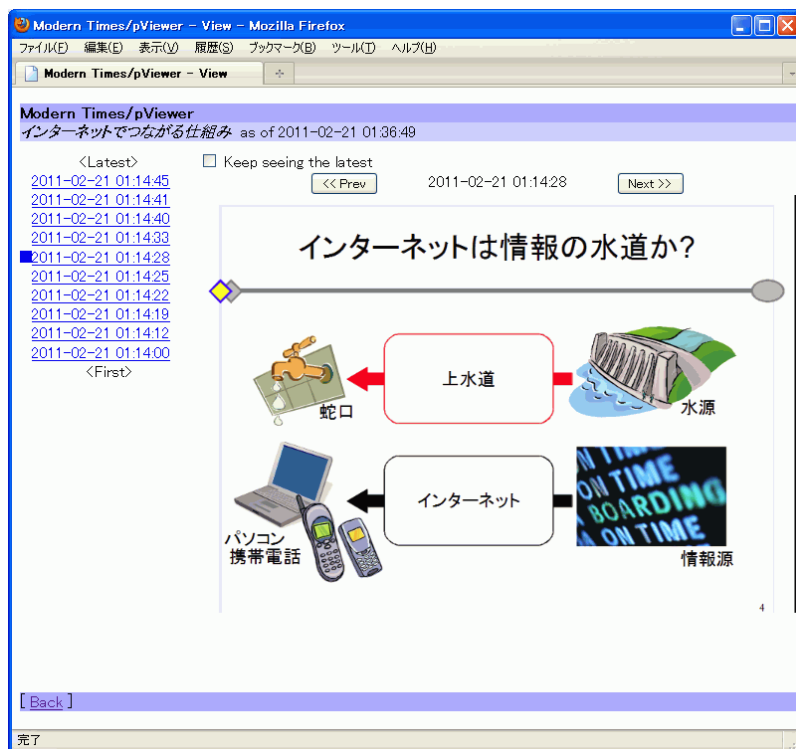


図 7 ビューア(pViewer)の画面構成

5. まとめ

本稿では、大学等において PC とスクリーンを使って講義を行う場合における問題点と、それを解決するための本研究の方針を示し、この方針に基づいて開発したシステムの設計および実装上の工夫について説明した。このシステムは、教員の PC で実行するソフトウェアの実行状況や、画面表示の変化を監視し、あらかじめ指定された条件に合う画面のみ自動的にキャプチャして、そのスクリーンショットをサーバに転

送する。受講者はこれを Web ブラウザ上で閲覧することができる。このシステムを用いることによって、教員は手間をかけずに様々な資料を織り交ぜつつもノイズの少ない講義の記録を受講者に提供することができ、受講者は Web ブラウザ上でスクリーンショットを探索することができる。

記録対象のソフトウェアの指定を変更できる機能を教員側クライアントに実装すること、ビューアの機能を充実させることなどが今後の課題である。また、本稿執筆時点は実現に至っていないが、これらのスクリーンショットを講義録画のインデックスとして活用することや、マウスポインタの移動を記録することによって講義中の PC 操作を紙芝居的に再現することも可能であると考えている。

参考文献

- 1) 牛島和夫: コラム「情報技術と教育」第 7 回, ファカルティ・ディベロプメント, 情報処理, Vol. 45, No. 1, p. 84 (2004).
- 2) 下川俊彦, 牛島和夫: 講義記録システムの利用状況, 九州産業大学情報科学会誌, Vol. 7, No. 1, pp. 23-24 (2008).
- 3) 井上宗徳, 下川俊彦: 講義スライドのフッターを用いたラベル付けによる講義映像のインデックス作成に関する研究, 電子情報通信学会技術研究報告(教育工学研究会), ET2007-55, pp. 1-6 (2007).
- 4) 下川俊彦, 田中宏宗: 講義記録システム映像に対する自動索引作成システムの構築, 九州産業大学総合情報基盤センター広報誌 COMMON, Vol. 29, pp. 12-20 (2009).
- 5) Microsoft: Windows ホーム, Windows Aero とは, <http://windows.microsoft.com/ja-JP/windows-vista/What-is-Windows-Aero>
- 6) Microsoft: MSDN, Desktop Window Manager, [http://msdn.microsoft.com/en-us/library/aa969540\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa969540(v=vs.85).aspx)
- 7) Microsoft: MSDN, DwmEnableComposition Function, [http://msdn.microsoft.com/en-us/library/aa969510\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa969510(VS.85).aspx)