

アニメーション付きプレゼンライドの操作可能な Web 動画化とその応用

Converting an Animated Presentation Slide to a Web Video and Its Application

丹羽 佑輔*¹ 白松 俊*² 大冨 忠親*² 新谷 虎松*²
 Yusuke NIWA Shun SHIRAMATSU Tadachika OZONO Toramatsu SHINTANI

*¹ 名古屋工業大学大学院 工学研究科 創成シミュレーション工学専攻

*² 名古屋工業大学大学院 工学研究科 情報工学専攻

Dept. of Scientific and Engineering Simulation, Graduate School of Engineering Nagoya Institute of Technology

Dept. of Computer Science and Engineering, Graduate School of Engineering Nagoya Institute of Technology

We realized a web presentation system enables to display and manipulate presentation slides on web browsers. For realizing to transit among slides including animations and effects on web browsers, we implemented a system to convert presentation slides to videos that have each slide as chapters. The system provides a presentation environment using web browsers supporting HTML5 without any additional plugins. Furthermore, the system enables to create thumbnails of the starts of each slide and animation from the chapter information. The web content slide is applicable to signage contents and HTML snippets that is able to be embedded into web pages.

1. はじめに

既存のプレゼンライドを Web 上で表示/操作可能にするための Web プレゼンテーションシステムを実現した。ここでは、アニメーションを含むプレゼンライド上の視覚効果を再現しつつ、Web 上でページ毎のスライド移動を実現するために、プレゼンライドをチャプター情報付きの動画ファイル(以降、プレゼン動画と呼ぶ)に変換するシステムを実装した。PowerPoint で作成したアニメーション付きのプレゼンテーション資料を Web ブラウザ上で表示することを可能とした。これにより、既存の HTML5 対応 Web ブラウザを利用して、プラグインを必要とせず効果的なプレゼンが可能になる。また、チャプター情報を用いることで、動画中のスライドおよびアニメーション開始時のサムネイルを生成することが可能であり、サムネイルとして利用するための画像ファイルを用意する必要がない。応用として、Web コンテンツ化スライドを HTML スニペットとして既存の Web ページへ貼りこむことが可能であり、迅速な情報発信が可能になり、また、Web サイネージシステムの魅力的なコンテンツとして再利用可能である。

2. プレゼンライドの Web 動画化

プレゼンライドをプレゼン動画へ変換するシステム[1]について説明する。本システムを用いることで、アニメーション付きのプレゼンライドを Web ブラウザ上で閲覧することが可能となる。プレゼンテーションファイルとして PowerPoint プレゼンテーションファイルを対象とする。本システムは、PowerPoint アドインとして開発した。PowerPoint アドインは、PowerPoint の UI に統合することができる。そのため、ユーザーは、PowerPoint の UI からプレゼン動画化プログラムを実行することができ、OS のスタートメニューまたはエクスプローラ上から起動するなどの手間を省くことができる。本システムでは、PowerPoint Object Library (以降、PPOL と呼ぶ。)を用いて、PowerPoint の機能を利用する。また、PPOL は、PowerPoint プレゼンテーションファイルをオブジェクトモデルとして提供する。呼び出し元プログラムは、このオブジェクトモデルを使用して、プレゼンテーションファイルに含ま



図 1. プレゼン動画から生成したサムネイル一覧の表示

れているスライドやアニメーションの情報を取得できる。今回は、PPOL のバージョン 15.0 を使用した。本アドインは、PowerPoint の機能を用いてプレゼンテーションファイルを mp4 形式ファイルへ変換する。プレゼンテーションファイルのオブジェクトモデルを使用して、スライドの画面切り換えアニメーション、およびクリック時に再生開始されるアニメーションの情報を取得する。この情報を mp4 動画にチャプター情報として埋め込むことでプレゼン動画を作成する。チャプター情報とは、動画中のある時刻が、どのスライドのアニメーションに対応しているかを表す情報である。Web ブラウザ上で再生する際は、チャプター情報を用いて、JavaScript によって動画の再生の停止・再生の制御を行うことで、Web ブラウザ上でのスライド操作が可能となる。

3. プレゼン動画のサムネイル生成と表示

Web ブラウザ上でスライドを閲覧する際、プレゼンソフトの様に、サムネイル一覧が表示でき、サムネイルの選択によって、スライドの移動が可能であると便利である。Web ブラウザ上でプレゼン動画から、各スライドおよびアニメーションが開始する時点のサムネイル一覧を表示する方法について説明する。プレゼン動画に含まれるチャプター情報を用いて、スライドまたはアニメーションの開始時刻を取得する。それらの時刻に動画の再生位置をシークし、その時刻のビデオ映像を静止画として記録することで、サムネイルを生成する。実装環境は、HTML5 である。

連絡先: 丹羽佑輔, 名古屋工業大学大学院 工学研究科 創成シミュレーション工学専攻, yusuken@toralab.org

```

function getThumbnails(video, timings, callback, ctx) {
  if (ctx == null) {
    ctx.i = 0; ctx.thumbnails = [];
    ctx.seeked = function() {
      video.removeEventListener('seeked', ctx.seeked);
      var thumbnail = document.createElement('canvas');
      thumbnail.width = video.videoWidth;
      thumbnail.height = video.videoHeight;
      thumbnail.getContext('2d').drawImage(video, 0, 0);
      ctx.thumbnails[ctx.i] = thumbnail;
      ctx.i += 1;
      getThumbnails(video, timings, callback, ctx);
    };
    return getThumbnails(video, timings, callback, ctx);
  } else if (ctx.i >= timings.length) {
    return callback(ctx.thumbnails);
  }
  video.currentTime = timings[ctx.i];
  video.addEventListener('seeked', ctx.seeked);
}

```

リスト 1. WEB PLAY での資料共有

video 要素を用いて動画の再生を行い, JavaScript を用いて動画のシーク処理を行う。canvas 要素を用いて, video 要素に描画された動画の画像を複製し, サムネイルを作成する。実際に, Web ブラウザ上でプレゼン動画から生成したサムネイル一覧の例を図 1 に示す。サムネイル生成の関数 getThumbnails の JavaScript をリスト 1 に示す。引数の video はプレゼン動画が読み込まれた video 要素, timings はチャプター情報の時刻の配列, callback はサムネイルの作成処理が完了した場合に呼び出す関数を, それぞれ指定する。ctx は関数の再帰呼び出しで利用する一時変数である。初回の関数呼び出しでは ctx を省略する。関数の処理の流れについて説明する。初回の呼び出しでは, ctx が生成される。ctx は, 現在取得を試みているチャプターのインデックスを示す変数 i, 複製したサムネイルを保存するための配列 thumbnails, video 要素のシーク完了時に呼び出す関数 seeked によって構成される。ctx が存在する場合は, すべてのサムネイルを複製が完了したかどうかを確認するため, ctx.i の値が配列 timings の要素個数以上であるかを確認する。すべてのサムネイルの複製が完了した場合は, 複製したサムネイルを引数として関数 callback を呼び出す。そうでない場合は, 配列 timings の ctx.i が示す要素の時刻にシークする。動画のシークは, currentTime プロパティに値を設定することで行う。サムネイル取得後, 各サムネイルの click イベントで, そのサムネイルに対応する動画中の時刻へ動画をシークさせる命令を記述する。これにより, サムネイルのクリックでプレゼン動画の表示スライドの切り替えを可能とした。サムネイルの生成時間は, プレゼン動画の動画の長さが 115.55 秒, チャプターの数が 24 の場合, 0.56 秒で完了した。実行環境は, CPU は Core i7-4850HQ 2.3 GHz, メモリは 16.0 GB, ブラウザは Google Chrome バージョン 33.0, である。

4. 資料の共有と HTML スニペットによる配信

図 2 に示す WEB PLAY は, サーバー上で動作するアプリケーションであり, Web 動画化した資料のアップロードを受け付ける。Web クライアントに対しては, 資料の閲覧, HTML スニペットの作成機能を提供する。アップロード時にタグ付けした文字列により, 資料の検索を行うことも可能である。Web 動画化したプレゼンスライドの再生には, HTML5 で記述したプレイヤーを呼



図 2. WEB PLAY での資料共有



図 3. HTML スニペットの生成ダイアログ

び出して行う。WEB PLAY によって共有したプレゼン動画は, WEB PLAY 上で HTML スニペットを生成して, 既存の Web ページへ貼り付けることが可能である。図 3 は, WEB PLAY のプレゼン動画の HTML スニペット作成のダイアログである。このダイアログでは, プレゼン動画の表示サイズ, 自動再生, リピート再生, 自動再生時のスライド送りの間隔を設定でき, 設定内容に応じた HTML スニペットを生成する。

5. おわりに

本研究では, Web プレゼンテーションシステムの開発を行った。本研究は, PowerPoint のプレゼンテーションスライドを対象に, アニメーションを含むプレゼンテーションスライドを動画化するシステムを開発した。既存の PowerPoint プレゼンスライドを, Web ブラウザ上で閲覧できる形式に変換するための PowerPoint アドインを開発した。本アドインを用いることで, アニメーション付きのプレゼンスライドを, 特別なプラグインを入れることなく Web ブラウザ上でも再生することを可能とした。また, チャプター情報を用いることで, 動画中のスライドおよびアニメーション開始時のサムネイルを生成することが可能であり, サムネイルとして利用するための画像ファイルを用意する必要がない利点がある。

参考文献

- [1] 丹羽 佑輔, 白松 俊, 大園 忠親, 新谷 虎松, "既存プレゼン用スライドの操作情報付き動画変換に基づく Web コンテンツ化について", 電子情報通信学会 人工知能と知識処理研究会, Vol.113, No. 332, pp. 53-58, 2013.