

東京工科大学 研究プロジェクト報告書

個人適応型ニュース配信プロジェクト

Personal Adaptive News Distribution project

2017 年度～2018 年度

メディア学部

近藤 邦雄

佐々木 和郎

三上 浩司

菊池 司

竹島 由里子

鶴田 直也

兼松祥央 2018 年度

宇佐美 亘 2017 年度

寺岡 丈博 2017 年度

目次

1. はじめに (3)
2. TVML と自然言語処理による CG ニュース映像生成 (4)
3. バーチャルキャラクターとスタジオセットの構築 (22)
4. Wordpress ブログからの TVML 番組自動生成 (37)
5. Android 版アプリ TG プレイヤーの制作 (51)
6. まとめ (59)
7. 本研究に関する発表と展示 (60)

1. はじめに

新しいメディア技術や多様な情報表現，そしてユーザインタラクションを生かしてユーザにわかりやすく正確に伝える報道をデジタルジャーナリズムという．インターネット上には多種多様な情報があり，それらの情報をもとに人が知りたい情報に整理して，発信することが大切である．またメディアの発達により，さまざまな情報が大量に蓄えられビッグデータと呼ばれるようになった．これをわかりやすく表現することは報道にとって重要な課題である．このような課題を解決するために，本研究では，個人対応時代のための新たな報道の価値を作る「個人適応型ニュース配信」の確立を目的とする．

そのために本研究では，次のことを行う．

1. インターネット情報の分析によるテキスト情報の要約制作のための言語処理法を考案する．このためにインターネット上の情報分析と記事作成のための自然言語処理手法の提案と Web ページからニュース記事を制作する手法を検討する．
2. 要約処理で得られた文書情報とユーザが指定した記事情報の画像や映像をもとにニュースコンテンツを自動または半自動で制作し配信するシステムの開発を行う．このシステムの開発では，Web ページを入力して映像を出力するという機能を中心に文書表示・提示方法などを提案する．
3. これらを用いた個人適応型ニュースコンテンツ制作配信システムや個人向け記事制作手法をまとめる．特に，Web 上のさまざまな情報をもとに，CMS (Content Management System) を用いて Web ページ制作を行う．そして CMS で構築する Web 情報を入力して，ニュースコンテンツを出力する「ニュースコンテンツ制作配信システム」を開発する．

本研究プロジェクトによる個人適応型ニュースコンテンツ制作配信システムの特徴は次のとおりである．

- (1) インターネット上の情報を解析して有用な報道情報を抽出し，それらに対して自動的に提供することができる．
- (2) インターネット上の Web データをもとに，ニュースコンテンツとして質的に高度な内容を提供することができる．

2. TVML と自然言語処理による CG ニュース映像生成

本章では、Web 上の記事からニュース番組を模した CG アニメーションを自動生成するシステムについて述べる。本システムは TVML 技術を応用したものであり、Web 上の記事を番組の台本に相当する TVML スクリプトに変換する機能を提供する。

2.1 システム概要

テキスト台本を書くだけでテレビ番組的な CG アニメーションを生成する仕組みである TVML は、インターネット上の様々なテキスト情報をコンピュータで自動的に CG アニメーションに変換する道が開ける可能性がある。しかしながら、Web 上のテキスト情報をそのままアニメーションに変換するには、テキストの量や演出などに関する課題がある。本システムでは、次の 2 点の解決を試みた。

1. 長文テキストを、1 本のニュースとして適切な長さに自動要約する
2. 演出情報(カメラワーク, アートワーク, 映像切り替えなど)や CG キャラクターのセットをまとめて、テンプレートとして利用できるようにする

提案システムの概要を図 2.1 に示す。まず、Web Scraping によってブログなどの Web 記事からテキストと画像 URL を取得する。取得されたテキストは適切な分量に要約され、さらに重要文をスーパーインポーズ(テロップ)として抽出しておく。そして、要約したテキストやスーパーインポーズ、画像 URLなどを合わせて、APE (Automatic Production Engine) スクリプトと呼ぶ中間ファイルに変換する。APE スクリプトは、APE と呼ばれる TVML 変換モジュールによって、演出情報をまとめた TVML テンプレートを参照しつつ TVML スクリプトに変換される。最後に、TVML エンジンに渡すことで CG アニメーションが自動生成される。

以降の節では、上記の処理順に従って、それぞれの詳細を説明する。

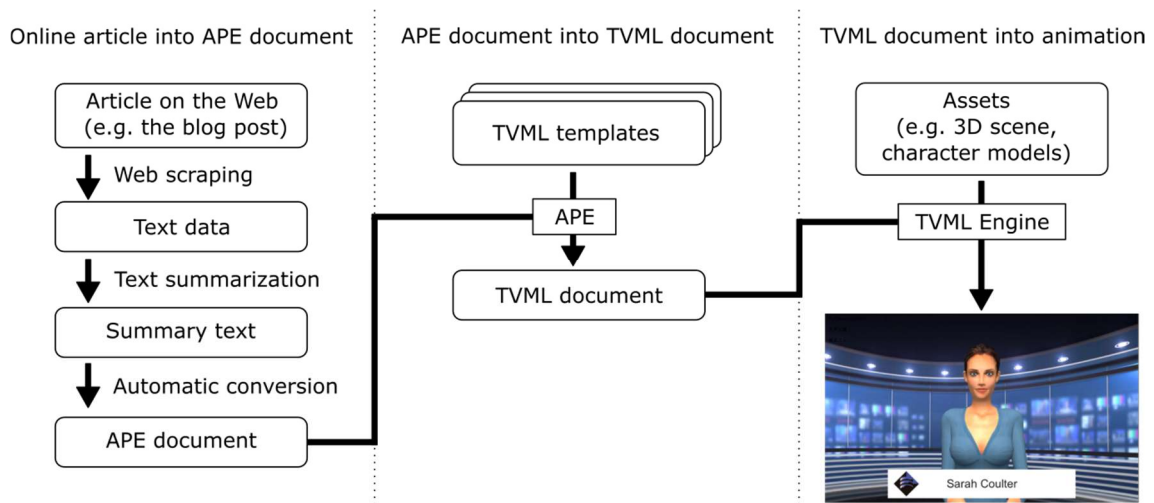


図 2.1 システム概要

2.2 Web から取得したテキストの要約

Web 上の記事に含まれるテキストや画像 URL を取得するために、東京工科大学メディア学部のブログを対象とする簡易的なクローラを実装した。Web サイトによって HTML 構造は異なるので、記事内容を取得するためのクローラは Web サイトごとに作成する必要がある。本節では、メディア学部のブログ記事を例として要約のプロセスを説明する。ブログの文章の分量は記事によって増減するが、ニュース番組では、一つのトピックに対するセグメントの分量はそれほど大きく増減せず、ほぼ一定である。したがって、長すぎるブログのテキストは短くしなければならない。この目的のために自然言語処理における要約文生成アルゴリズムを使って、ブログの記事の分量を 200 ワード程度に短くする。また、ニュース番組では、話している内容を一行に要約して画面にスーパーインポーズする手法が使われる。これについては、ブログ文の中で最も重要度の高いセンテンスを探して、これをスーパーインポーズ用として出力することを試みた。

2.2.1 目的

本研究における要約とは、元のテキストの内容をより短いテキストで簡潔にまとめることを指す。要約の種類は、対象となる文書ならびに要約文の作成方法で分けることができる。要約の対象となる文書が単一の場合は「単一文書要約」、複数の場合は

「複数文書要約」と呼ぶ。また、文書から要約文の元になる文を抽出した後に、文自体を変更せずに出力する形式を「抽出型要約」、我々人間が要約するような新しい文を生成しながら内容を抽象的に出力する形式を「抽象型要約」と呼ぶ。本研究では、スクレイピングしたブログの記事に対してニュース番組を自動で生成することを目的としているため、ここではブログの一記事つまり単一文書を対象とした抽出型要約を行う。

2.2.2 手法

本研究では、Quyangらの Basic Summarization Model (Quyang et al., 2010) を日本語に対応させ、抽出型要約に応用している。具体的には、まず最初にブログの記事内の文書に対して形態素解析¹を行い、文書を構成する文に対して単語ならびにそれらの文書内における出現頻度を求める。次に、式(1)のように各文の重要度 s を文中の単語数 S と文中の単語の文書内出現頻度 w_i を用いて計算する。そして、あらかじめ設定した要約文の単語数(ここでは 200)を超えないよう、重要度の降順で文を抽出する。最後に、抽出した各文を元の記述順に繋ぎ、要約文として出力する。

$$\text{score}(s) = \frac{1}{\|S\|} \sum_i \log w_i \quad (1)$$

上記で出力する要約文は、自動で制作するニュース番組においてアナウンサーが話す原稿となる。この原稿データの他にも、スーパーインポーズで表示する文字列が必要となるが、この文字列に関しても次のように求める。上述した要約文を構成する重要度の高い文を抽出する過程で、まずは最も重要度が高い文を別途抽出する。この抽出した文に対して、構文解析²を行い、述語を同定する。この述語が動詞の場合は原形に、サ変名詞を含んでいる場合は名詞に変換する。次に、この述語と係り先の関係のある文節に対して、述語と最も近い位置にある文節を抽出する。そして最後に、「文節 + 述語」という形式でスーパーインポーズ用の文字列として出力する。この文字列は画面内に一列で表示されることを想定しているため、10文字前後を目安にしている。

以上より、ブログ記事からニュース原稿用の要約文ならびにスーパーインポーズ用の文字列を求めることが可能となる。

¹ 形態素解析器は MeCab 0.996 を使用

² 構文解析器は CaboCha 0.69 を使用

2.2.3 ブログ記事に対する出力例

図 2.2 は、対象とするブログ記事の一つ³である。スクレイピングで得た記事のデータは、一記事につき一ファイルで構成しており、ファイル名は記事の日付で識別している。図中の赤い枠で囲われた箇所が、記事の日付とともに先述した手法で抽出される内容に該当する。

「しぶやでこうか」でサウンド関連3テーマの発表(その1)

2018年3月 2日 (金) 投稿者: メディア技術コース

メディア学部の大淵です。


2月24日(土)に波東シネタワーで開催された「しぶやでこうか」で、私の研究室から3件の発表を行いました。今日はその中から、馬上優太さんの「3次元空間中の音の自動判別とVR技術による可視化」というテーマをご紹介します。

音による情報は、注意をしていなくても気付くというメリットがある一方、一瞬で消えてしまっ後に残らない(専門用語で揮発性といいます)というデメリットがあります。そこでこの研究では、音の情報を目で見えるように変換することを試みました。つい聞き逃した情報を映像として見られるだけでなく、聴覚にハンディキャップのある人にも役に立つ研究です。

音がどちらから来ているかを自動検知するために、複数のマイクを並べたマイクロフォンアレイと呼ばれる装置を使用します。それぞれのマイクに音が到達するタイミングのずれを利用して、音源の方向を推定するのです。ここでは、HARKというオープンソースソフトウェアを活用しました。そして音源の方向がわかったら、それをヴァーチャルリアリティ(VR)の3次元空間中に画像として表示します。VRの実現には、Unityというゲームエンジンを使いました。そして、開発した機能をわかりやすく実演するために考えたアプリケーションが…

VRスイカ割り!

二人一組でプレイするゲームで、一人はVR空間中に現れたスイカの場所を声で伝え、もう一人はそれを聞いてマイクロフォンアレイの周囲で手を叩くというものです。スイカの場所でちゃんと手を叩くと、VRの中でスイカが割れて得点できます。発表では、会場にVRシステムを設営し、実際にVRスイカ割りを体験してもらいました。(スクリーンに映っているのはVR空間の映像ですが、これは本来はVR担当のプレイヤーにしか見えません)



発表では、なかなかスイカが割れずハラハラする場面もありましたが、会場の皆さんには楽しんでいただけたのではないかと思います。やはり実演があると盛り上がりませぬ。

図 2.2 メディア学部のブログ記事例。赤枠の箇所は抽出箇所を表す。

³ <http://blog.media.teu.ac.jp/2018/03/31-c7d5.html>

表 2.1 ブログ記事(図 1)に対する出力内容.

	出力内容
タイトル	「しゅやでこうか」でサウンド関連 3 テーマの発表(その 1)
スーパーインポーズ用文字列	VR スイカ割りを体験
要約文	そこでこの研究では、音の情報を目で見えるように変換することを試みました。それぞれのマイクに音が到達するタイミングのずれを利用して、音源の方向を推定するのです。ここでは、HARK というオープンソースソフトウェアを活用しました。VR の実現には、Unity というゲームエンジンを使いました。スイカの場所でちゃんと手を叩くと、VR の中でスイカが割れて得点できます。 発表では、会場に VR システムを設営し、実際に VR スイカ割りを体験してもらいました。 (スクリーンに映っているのは VR 空間の映像ですが、これは本来は VR 担当のプレイヤーにしか見えません)発表では、なかなかスイカが割れずハラハラする場面もありましたが、会場の皆さんには楽しんでいただけたのではないかと思います。
図	http://blog-media-teu.cocolog-nifty.com/photos/uncategorized/2018/03/01/magami.jpg

表 2.1 は、図 2.2 のブログ記事に対して実際に出力された内容を表している。表の各項目について述べると、まず「タイトル」はブログ記事のヘッダーに該当し、スクレイピングで得られた内容をそのまま出力している。「スーパーインポーズ用文字列」は、「要約文」の中で重要度 s が最も高い文(表中の太字部分)から抽出した文字列である。「要約文」は、単語数が上限の 200 を超えないようになっており、重要度 s の降順で抽出した文を元の記述順に繋いでいる。「図」に関しては、ブログ記事の中に掲載されている図の URL をスクレイピングで取得しており、ニュース制作を自動化する際に必要に応じて画像をオンラインで取得できるよう、URL をそのまま出力している。

2.2.4 考察

本手法で要約文を作成する際、文を構成する単語が文書内にどれくらい出現しているのかを求め、式(1)を用いることで文の重要度 s を計算している。この重要度の降順で抽出した文を元の記述順に繋いで作成した要約文は、表 2.1 からも分かるように、大局的には文脈の整合性がある程度確保できていると考えられる。しかし、元の記述順で文を繋いでいるものの、必ずしも前後に隣接する文が重要度の高い文として抽出されるわけではない。そのため、抽出された文に接続詞や照応詞(代名詞や指示詞など)を含んでいる場合、要約文中の前後に隣接する文脈として不適切な表現に該当することが課題として挙げられる。例えば、表 1 の要約文では、冒頭の「そこでこの研究では、～」における接続詞「そこで」が該当する。これは順接を表す接続詞のため、本来ならばこの冒頭文の前に文意が繋がるような文が必要である。要約文の冒頭文に接続詞から始まる場合は削除することで解決するが、要約文の途中に接続詞から始まる文が位置する場合は、前文との意味関係を正しく解析できなければならない。また、「彼」や「彼ら」などの代名詞や「これ」や「それ」などの指示詞が含まれる文の場合も、要約文の冒頭からその前文までの間に代名詞や指示詞が指している先行詞が含まれている必要がある。そのため、要約文に関しては、文間の意味関係や先行詞の同定が今後の課題として挙げられる。

次に、スーパーインポーズ用の文字列について述べる。本手法では、要約文の中で最も重要度の高い文から、「文節 + 述語」の形で抽出している。表 1 の場合、最も重要度の高い文は「発表では、会場に VR システムを設営し、実際に VR スイカ割りを体験してもらいました。」(太字)である。この重要文に対して、構文解析によって得られた係り受け関係(述語とそれに一番近い文節)から「VR スイカ割りを体験」という文字列を抽出している。この文字列がどの程度正しく抽出できているかどうかを判断するため、2名の学生に要約文に対しスーパーインポーズ用文字列を抽出してもらったところ、1名は「目で見える音『VR スイカ割り体験』発表」、もう1名は「スイカ割りで音の可視化」という結果となった。本手法による文字列は、前者とは「VR スイカ割り」と「体験」で、後者とは「スイカ割り」で重複していたことから、人間に近い文字列の抽出ができていたといえる。

表 2.1 の例以外にも出力した 52 件の要約文に対して、上記の学生 2 名にスーパー

インポーズ用文字列を抽出してもらったところ、本手法が出力した文字列の単語が両者の文字列に含まれていたのは、4件のみであった。これは、そもそも人間同士でも抽出した文字列が大きく異なることが第一の理由として挙げられる。実際に、両者の文字列間で単語が重複したのは 52 件中 29 件しかなかった。そのため、必然的に本手法による文字列との比較対象が少なくなり、その結果、表 1 の例を加えても全体で 30 件中 5 件しか重複しなかった。また、2 名の学生による文字列は「抽象型」のため、要約文において複数の文から単語が抽出されて組み合わされたり、言い換えられたりしていたことも理由として挙げられる。本手法による文字列は重要文の一文のみから抽出される「抽出型」であり、比較対象となる単語の範囲が狭くなったためである。ゆえに、スーパーインポーズ用文字列に関しては、要約文から重要度が最も高い文だけでなく、2 番目(あるいは 3 番目)まで重要度が高い文を抽出し、重要な単語を組み合わせて出力することで、人間が抽出するような文字列に近づけることが期待される。

以上から、本手法による要約文は、文の前後関係による文脈の繋がりに関しては課題があるが、全体の文脈は一定の整合性を再現できている。その一方で、スーパーインポーズ用の文字列に関しては、限定的には人間に近い抽出が可能であるが、抽出対象の範囲と単語を繋げる方法の両面に課題が残された。

2.3 APE スクリプト(台本中間ファイル)の生成

前節の自然言語処理によって、要約されたセリフとスーパーインポーズ用のテキスト、ブログで使われている画像の URL が抽出される。これらの情報から台本用の中間ファイルを生成するための変換スクリプトを実装した。具体的には、図 1 の記事から図 2.3 のような APE スクリプトが生成される。

APE スクリプトは、APE と TVML テンプレートで定義されたタグを持つ XML ベースのファイルである。これらのタグは、次章で説明する TVML テンプレートにおいて定義されており、番組のジャンルによって異なるタグが定義されている。ただし、アナウンサーのしゃべりの<talk>タグなど、共通のものもある。例えば、図 2.3 の冒頭部分はアナウンサーとしてこうかとの 3D キャラクターを使用することを想定したタグや名前のスーパーインポーズが記述されている。<superarticle>タグでは、ブログタイトルをそのまま

ニュースの見出しとして使用した。その後、要約文を 1 文ずつ区切って<talk>タグに入れて、画像やスーパーインポーズを表示しつつ読み上げるようにする。

ただし、画像の取り扱いについては、非常に難しい課題が残っている。図 2.3 の例では画像が 1 枚しか含まれていないため、それを表示すればよいが、複数枚あった場合は、理想的には適切な画像のみを選択して適切な位置に表示する処理が必要となる。しかしながら、要約処理では画像とテキストの関連性を考慮しておらず、また、そもそも画像の本文に対する重要度を評価することは容易ではない。そこで、例題として使用したメディア学部のブログにおいては、1 つの記事に 2-3 枚の画像があることが多かったため、「200 文字前後のセリフのうち冒頭の数文を 50 文字以上になるまで読み上げて、残りを読み上げる時間を等分して画像を表示する」ことにした。画像枚数が多い場合は瞬時に切り替わってしまうことになるが、その場合は一番目の画像だけを用いるなども考えられる。いずれにせよ、記事の特徴と番組の構成を考慮した変換処理が必要である。

ここで、APE エンジンと演出の構造を記しておく。我々は、APE スクリプトに、ある演出を与えるモジュールを一般に APE と呼んでいる。この APE には固有の名前が付いており、名前を指定して利用する。データ構造上では、APE の名前と同じ名前のフォルダ (APE フォルダ) があり、その中に、後述する TVML テンプレートと各種必要データ (CG キャラクタモデルデータ、CG スタジオセットモデルデータ、画像データ、音データなど) が入っている。APE エンジンは、APE スクリプトの<apescript>タグの ape 属性に書かれた APE 名を読み、その名前の APE を使って XML を TVML に変換する。この際、演出に必要なアートワークデータは、この APE フォルダ内にすべて入っている。この APE フォルダはポータブルであり、再生する PC のディスク上でも、サーバー上でも置くことができる。いったん APE フォルダを作れば、どこに持って行っても同じように用いることができる。

```

<apescr ipt ape="Koukaton">
  <supername>こうかとん</supername>
  <pause>1</pause>
  <bow/>
  <pause>2</pause>
  <talk>おはようございます, 東京工科大学ニュースです. </talk>
  <supernameoff/>
  <pause>1</pause>
  <superarticle>「しぶやでこうか」でサウンド関連3テーマの発表(その1)</superarticle>
  <talk>そこでこの研究では, 音の情報を目で見えるように変換することを試みました.
</talk>
  <talk>それぞれのマイクに音が到達するタイミングのずれを利用して, 音源の方向を推定
  するのです. </talk>
  <superarticleoff/>
  <pause>1</pause>
  <superupperright>VR スイカ割りを体験</superupperright>
  <mainimage>http://blog-media-teu.cocolog-
nifty.com/photos/uncategorized/2018/03/01/magami.jpg</mainimage>
  <pause>1</pause>
  <talk>ここでは, HARK というオープンソースソフトウェアを活用しました. </talk>
  <talk>VR の実現には, Unity というゲームエンジンを使いました. </talk>
  <talk>スイカの場所でちゃんと手を叩くと, VR の中でスイカが割れて得点できます.
</talk>
  <talk>発表では, 会場に VR システムを設営し, 実際に VR スイカ割りを体験してもらいま
  した. </talk>
  <talk>(スクリーンに映っているのは VR 空間の映像ですが, これは本来は VR 担当のプレイ
  ヤーにしか見えません)発表では, なかなかスイカが割れずハラハラする場面もありましたが, 会場の
  皆さんには楽しんでいただけたのではないかと思います. </talk>
  <superupperrightoff></superupperrightoff>
  <mainimageoff></mainimageoff>
  <pause>2</pause>
  <talk>おもしろメディアニュースをお送りしました</talk>
  <bow></bow>
  <pause>2</pause>
</apescr ipt>

```

図 2.3 APE スクリプトの例

2.3 TVML テンプレートを参照した TVML スクリプトへの変換

TVML テンプレートは, 図 2.1 の APE モジュールが APE スクリプトを TVML スクリプトに変換する際に参照するデータである. TVML テンプレートは, APE スクリプトの XML タグを置換するためのルールセットである. APE スクリプトの各タグは, TVML テンプレートに書かれた一連の TVML コマンドに置き換えられる.

TVML テンプレートの作成は以下のように行われる。一度このテンプレートを作っておけば、毎回異なる内容の APE スクリプトを与えることで、この演出パターンによる CG アニメーションが量産できる。

- 1) 模倣するテレビ番組を選定する
- 2) TVML スクリプトでテレビ番組をそのまま模倣する
- 3) 得られた TVML スクリプトを人間が編集して TVML テンプレートを作成する

変換の例を図 2.4 に示す。APE スクリプトに含まれる<talk>タグは、TVML テンプレート内の同<talk>タグによって置き換えられる。セリフ内容は APE スクリプトに記述されているテンプレートが使用されるが、挨拶やお辞儀などの一連の行動をテンプレート化しておくこともできる。図 2.3 の例では、冒頭と終わりの挨拶部分は APE 変換スクリプトに含めているが、APE では変化する部分のみを取り扱い、共通部分はすべてテンプレートに入れることが望ましい。

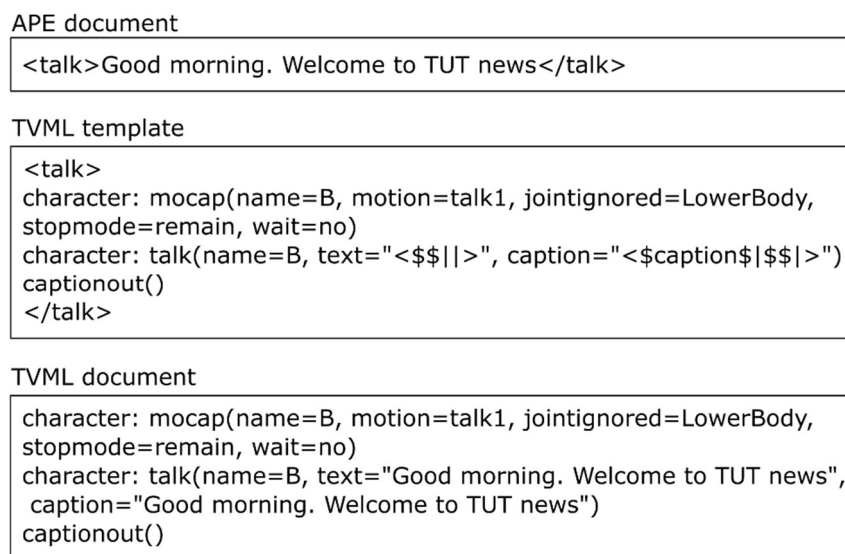


図 2.4 TV テンプレートによる APE スクリプトから TVML スクリプトへの変換

2.4 実験結果と考察

本節では、実装したスクリプトおよびシステムによって生成された CG アニメーションを示しつつ、その結果について考察する。実装は、図 2.1 のすべてのプロセスを統合

したものにはならなかった。現在はブログから HTML データをクローリングして、スクレイピングしたテキストから自然言語処理を行ったテキストをもとに APE スクリプトを生成するまでのプロセスと、APE スクリプトから TVML テンプレートを使って TVML スクリプトを生成して、これを TVML エンジンで再生する機能の 2 つに分かれている。これらを統合しなかった理由は、単に実装上の理由だけであるが、特定の Web サイトの記事の要約をするためには一定数以上の記事を事前にクローリングして重要単語の評価をしておく必要があるので、Web サーバ上に APE スクリプトを置く場合は分離した運用も考えられる。後者の TVML エンジンで再生する部分は、PC をプラットフォームに用い、Unity Game Engine 上に実装された TVML SDK を使用してアプリケーションを作成した。

2.4.1 ブログ記事からのアニメーション生成実験

図 2.5 はアプリケーションの実行画面であり、指定したディレクトリに含まれる APE ファイルの一覧が表示されている。いずれかのファイルを選択すると CG アニメーションの再生が実行される。使用したブログは、前述したように、東京工科大学メディア学部の主に高校生向けの広報用の日本語ブログである。内容は、メディアサイエンス、コンピュータグラフィクス、コンファレンスレポートなどについてである。ブログアーカイブからおよそ 1500 記事を収集し、自然言語処理を行い、一つの記事について、200 ワードのセリフの文章と、スーパーインポーズ用のセンテンス、画像の URL を抽出し、APE スクリプトに変換した。アプリケーションは、APE スクリプトを読み込み、対応する TVML テンプレートを参照しつつ TVML スクリプトを生成して、最終的な CG アニメーションを再生する。

以降では、例題として 3 つの記事を取り上げ、アニメーション再生の様子と自然言語処理の結果を示す。最初はピクトグラムについて紹介した記事であり(図 2.6)、再生するとタイトルは適切に表示されていることがわかる(図 2.7)。画像とともに表示される右上スーパーインポーズ(図 2.8)は適当とは言い難いが、驚きを表現するために疑問形のテロップが出ることも少なくはない。ややスペースが空いているように見えるのは 2 行分を想定しているためである。次の記事はクモの巣と卒業研究テーマの紹介である(図 2.9)。こちらはタイトルが長すぎたため枠外に収まらなかった(図 2.10)。タイトルに

についてはブログ記事で使用されたテキストをそのまま使用しているため、やや長くなってしまふことが多い。これについても単語の重要度によって適切に短くする処理が必要である。この記事には 3 つの画像が含まれているので、後半は画像が連続して切り替わることになるが、1 枚当たりの表示時間は 20 秒ほどであり、大きな違和感はなかった(図 2.11)。最後は、自然言語処理の節で紹介した VR スイカ割りの記事である。前述のスク립トを再生すると図 2.12 のように表示される。

以上の実験から、自動要約したテキストのセリフはおおむね意図に合った処理できたが、タイトルやスーパーインポーズのテキストには改善の余地があることがわかった。また、APE スクリプトへの変換においては、画像の選定と表示タイミングについてもさらに考慮する必要がある。



図 2.5 要約処理を組み合わせた APE スクリプトを再生するシステムの実行画面

Title:おもしろピクトグラムを探せ！
Telop:現れたのが
Summary:いつも目にしている、非常口やトイレのマークのアレです。ピクトグラムは、シンプルなイラストで表現されているため、一目で情報が伝わる、言葉が分からない人にも情報が伝えられるものとして、世界各国で利用されています。まず、最大の問題となるのが、トイレ。これでは、海外の旅行者は分からない・・・そこで、現れたのがピクトグラムです。トイレのピクトグラム、様々な種類があります。この画像は、韓国水原のトイレ博物館に行ったときに撮影してきた、世界のトイレマークの写真です。男女のマークが利用されているということは、どこも共通ですが、そのスタイルは様々です。海外に行くと、トイレだけではなく、交通標識や案内板など、日本にはないおもしろいピクトグラムがたくさんあります。
Figure:http://blog.media.teu.ac.jp/images/2017/10/26/img_2387_5.jpg
http://blog.media.teu.ac.jp/images/2017/10/26/img_1561.jpg

図 2.6 記事 1(2017-10-26)の自然言語処理結果



図 2.7 記事 1(2017-10-26)のアニメーション



図 2.8 記事 1(2017-10-26)の画像とスーパーインポーズの表示

Title:おもしろメディア学：八王子キャンパスの素晴らしい自然環境におけるクモの巣と卒業研究
Telop:びっくり

Summary:1 か月ほど前に、八王子キャンパスで、クモの巣を見つけました。とっても大きくて、1メートルくらいはありました。ただ大きいだけでなく、とてもきれいでした。図鑑でもこのような大きなクモの巣の写真を見たことはありません。クモの糸は細くてきれいに撮影することもむづかしいのですが、この写真はスマートフォンで撮影したのですが、巣の構造が分かるようになっています。この写真を facebook に掲載したところ、友人からメディア学部の卒業論文にクモの巣を CG で描く研究を見つけたというコメントをいただきました。びっくりしました。いろいろなアルゴリズムを考案するなかで、クモの巣を題材にして研究を進めた学生がいたのです。クモの巣をどのようにコンピュータグラフィックスで描くかに興味ある人は、ぜひ卒業論文を読んでください。

Figure:http://blog.media.teu.ac.jp/images/2017/11/19/22815462_1742334712478304_792775576.jpg

http://blog.media.teu.ac.jp/images/2017/11/19/22788848_1742334702478305_450687782.jpg

http://blog.media.teu.ac.jp/images/2017/11/19/22789008_1742334715811637_710629619.jpg

図 2.9 記事 2(2017-10-26)の自然言語処理結果



図 2.10 記事 2(2017-10-26)のアニメーション。タイトルが枠外に飛び出している。

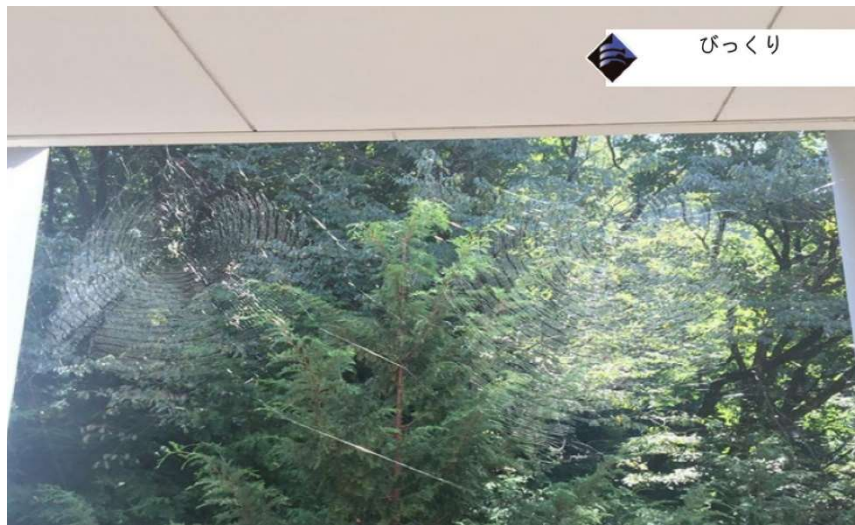


図 2.11 記事 2(2017-10-26)の画像とスーパーインポーズの表示



図 2.12 記事 3(2018-3-2)を再生した様子

2.4.2 ニュース番組制作例

図 2.13～2.15 に TVML を用いた学生によるニュース番組の制作結果の例を示す。これらは、東京工科大学大学院メディアサイエンス専攻で実施した大学院特別講義の課題として、博士課程前期の学生 18 名が課題として制作した。こ

の学生 18 名はいずれも TVML を用いた番組作成の経験はなく、講義として 4 コマ (360 分) 程度のレクチャーを受けた後、課題として 1~2 週間程度の期間で制作した。ニュースの題材としては、東京工科大学メディア学部ブログに掲載されている任意の記事を取り上げた学生が大半であるが、一部学生は web 上にある一般のニュース記事を用いている。また、図 2.13~2.15 をはじめ、TVML による番組作成で用いる映像素材はすべて TVML プレイヤーであらかじめ用意されたキャラクターモデル・背景モデルと、各学生が題材として任意に選定したブログ記事などに掲載されている画像である。したがって、映像制作を専門としていない学生であっても、テキストデータであるテキスト台本の操作のみで、任意の記事をニュース番組形式の動画として生成することが可能であった。



図 2.13 TVML を用いたニュース番組の作成例 1

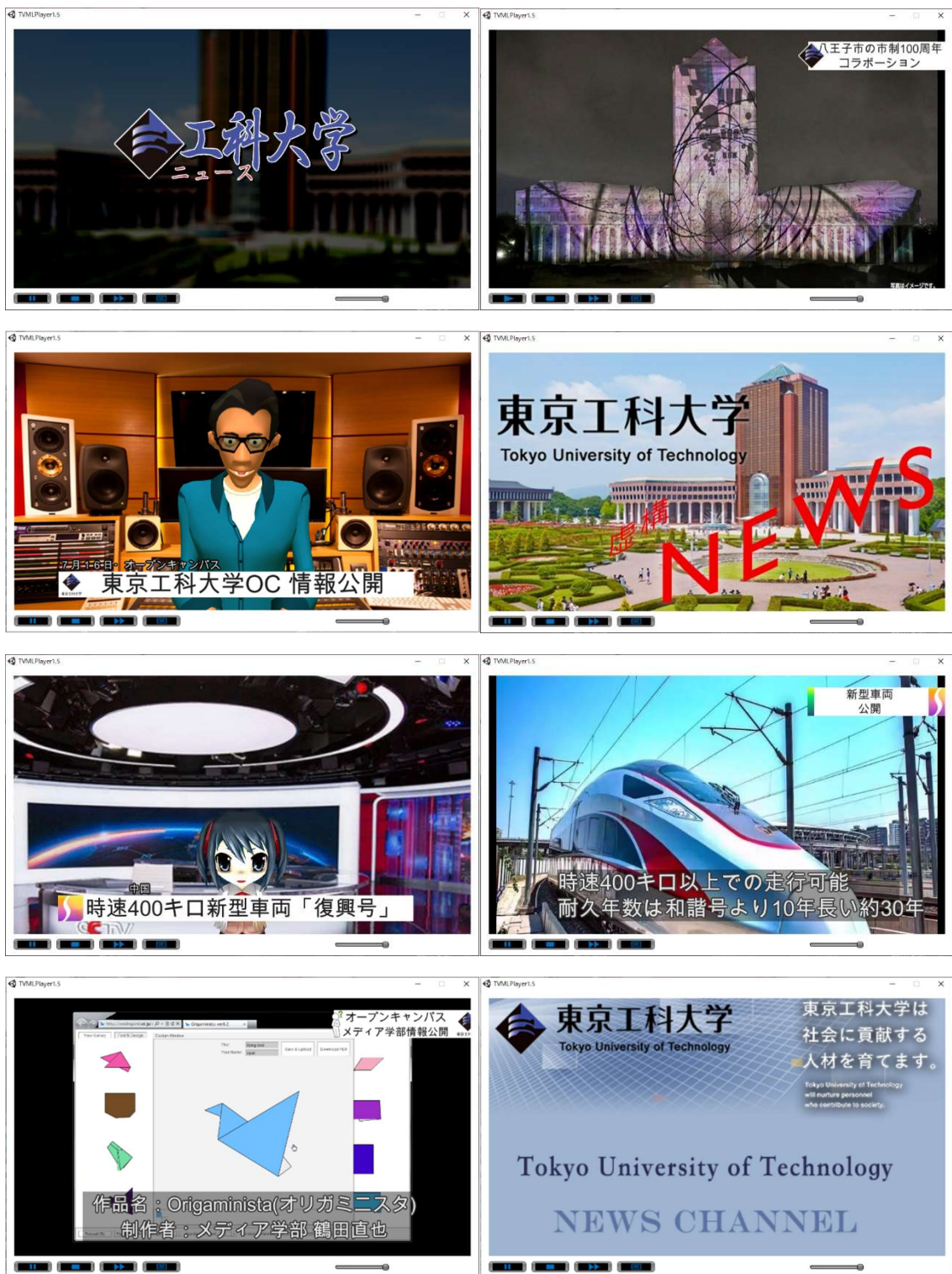


図 2.14 TVML を用いたニュース番組の作成例 2

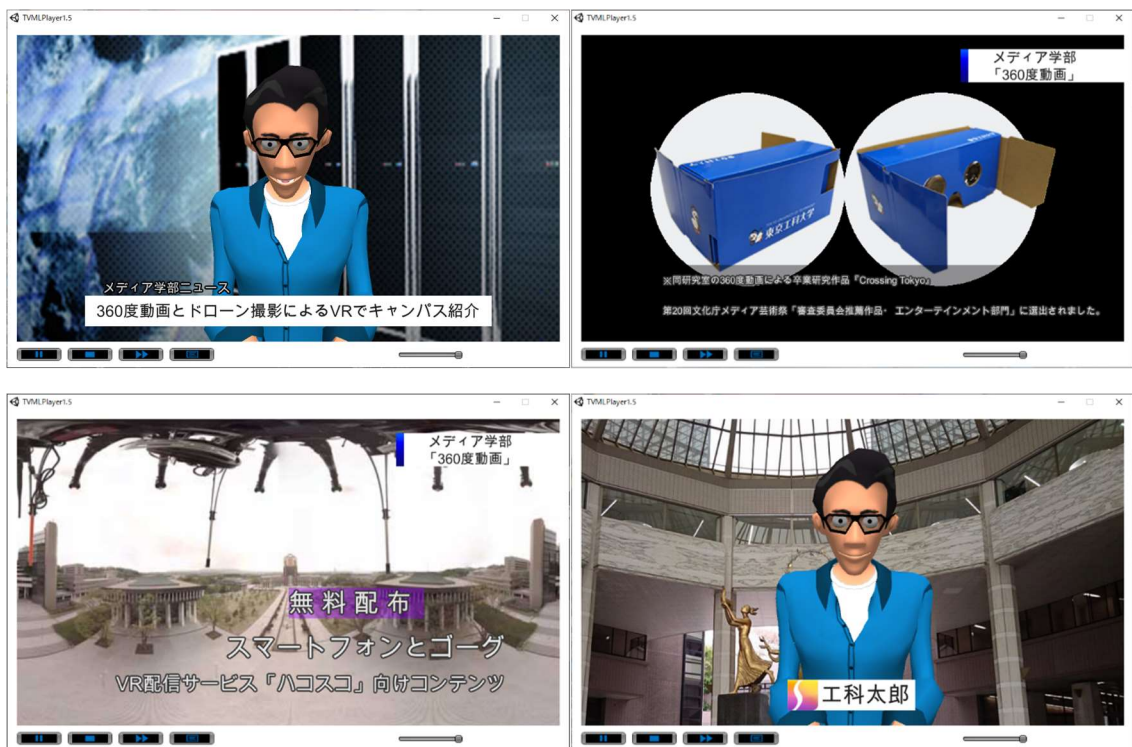


図 2.15 TVML を用いたニュース番組の作成例 3

3. バーチャルキャラクターとスタジオセットの構築

3.1 概要

本節では、ニュース配信をするためのアナウンサーであるキャラクターモデルとニューススタジオ空間の制作について説明する。

(1) 東京工科大学マスコット：こうかとの制作

本学内プロジェクトにおけるニュース番組配信の実験には、メディア学部のブログ記事などを利用する。このため、大学紹介や研究紹介などのニュース番組となるので、本学のマスコットを制作して、アナウンサーとして活用することとした。

(2) アナウンサーキャラクターの制作

本研究では、ニュース映像に登場する基本的な人物として、大人の男女、大学生くらいの若者の男女の4体を制作する。これは、一般的なニュースを配信する場合と、大学、学部の紹介など広報に使う場合で、ニュースを視聴する対象者に、より親しみを持ってもらえるようにするためである。

(3) スタジオ空間の制作

スタジオとして表現するためには、背景画像か、3次元モデルで制作したスタジオが必要である。ニュース番組において、動き回ることが少ない場合は、背景画像を利用することも一つの方法である。本研究では、TVMLで再生される番組のスタジオセット背景として、できるだけ実際のテレビ番組スタジオに近い3次元モデルデータを用意することとした。

3.2 こうかとん制作

こうかとんは、2016年に制定された東京工科大学のマスコットキャラクターである。

大学のノベルティグッズに採用されたり、着ぐるみのキャラクターが製作されイベントなどに登場し大学のPRを行っている。本研究では、広報の許可を得てこうかとんの3Dモデルを制作し、本プロジェクトのアナウンサーとして利用することとした。

(1) こうかとんのデザイン

こうかとんのデザインは、2015年度に学内公募によりコンペ形式により採用された。最終的に採択されたのは、当時デザイン学部 に在籍していた土肥千穂莉さんのデザインである。



東京工科大学 公式キャラクター
「こうかとん」

図 3.1 こうかとんのデザインイラスト

<https://www.teu.ac.jp/information/2016.html?id=75>

(2) 3D モデル作成

3DCG モデル作成は演習講師の中村陽介氏の指導の下、当時 4 年生の柴崎都さんに依頼した。モデル制作には 3dsMAX を使い、その後 MotionBuilder、さらには Maya 用のデータを用意した。

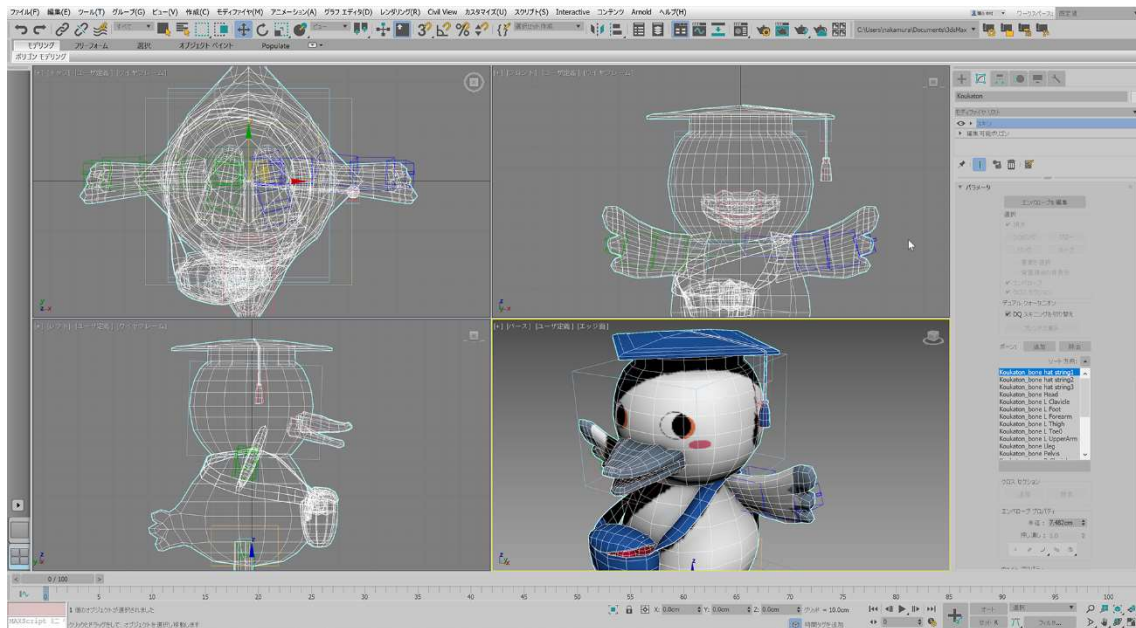


図 3.2 3dsMAX でのこうかとのモデリングの様子 1

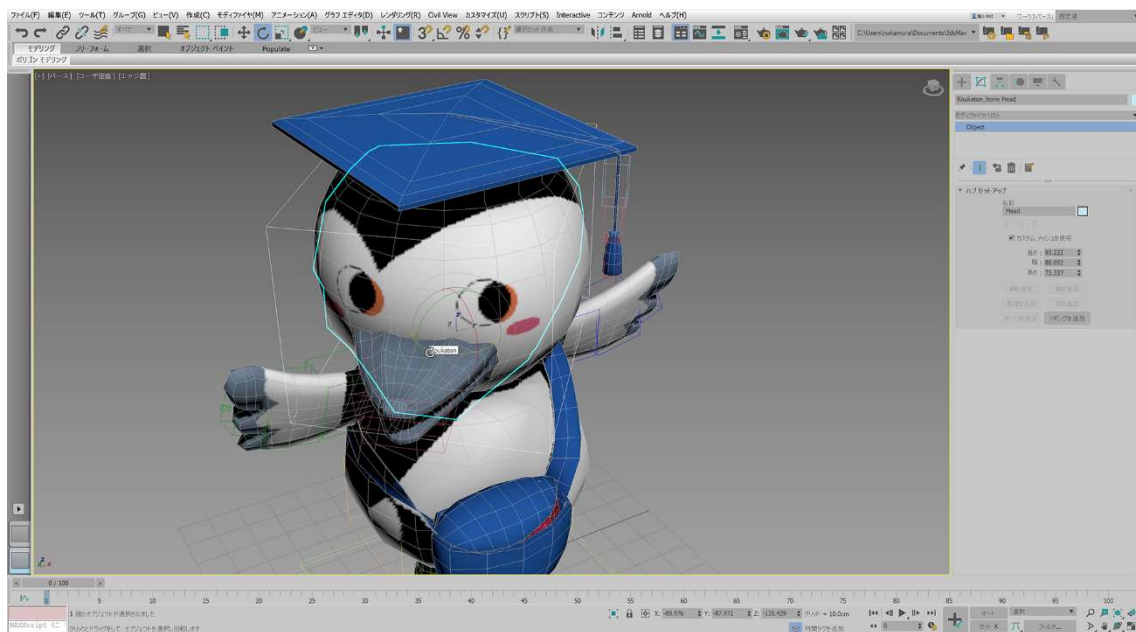


図 3.3 3dsMAX でのこうかとのモデリングの様子 2

アニメーション設定の際にアニメータが使用可能なツールを用いて設定できるように、汎用的なアニメーションソフトウェアに対応したデータを用意した。アニメーション付けしたボーン（ジョイント）オブジェクトを FBX ファイルで書き出し、Unity で読み込むことで、本システムのアニメーションとして利用可能となる。なお、ボーン（ジョイント）オブジェクトの名前を変更すると正しく動作しないため、そのままの名称を利用する必要がある。

また、最終的にはUnity をベースとした本システムにおいてリアルタイムで利用するため、モデルデータやテクスチャなどは軽量なモデルにしている。高品質なモデルのために高解像度テクスチャを別に用意しさまざまな用途に対応できるようにした。

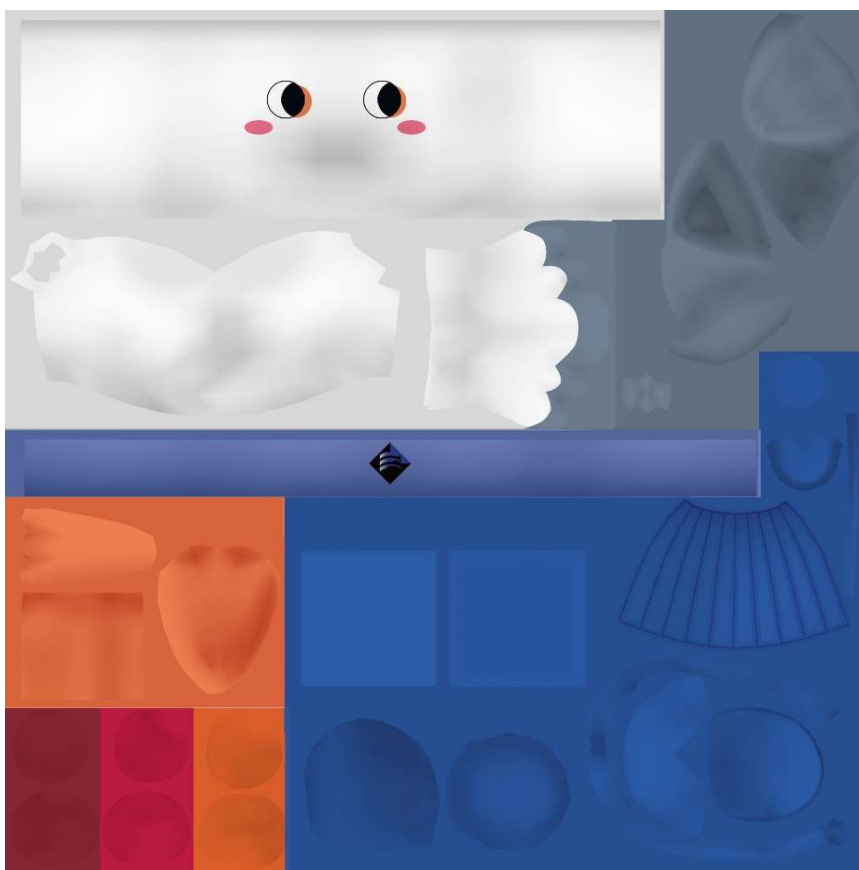


図 3.4 高解像度テクスチャ

(3) システムへの導入

作成したモデルを本システムが利用する Unity で利用可能にするためにパッケージ化した。Unity で利用するためには「Koukaton20170914.unitypackage」を Unity のコマンドの中の「Importpackage」から読み込む

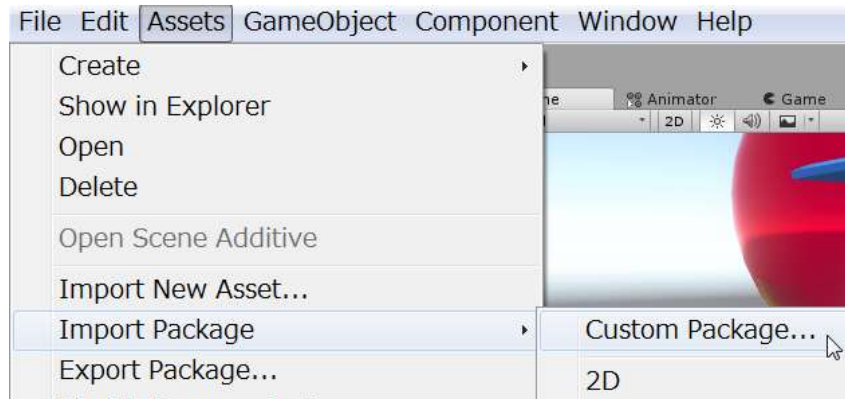


図 3.5 こうかとんモデルのシステム(Unity)への読み込み

その後 Unity の Asset 直下に「Koukaton」がこうかとんの 3D モデルファイルが表示される。このアイコンをシーンやヒエラルキにドラッグアンドドロップすることでシーン上にモデルが表示される。次に Unity 上でこうかとんモデルを表示した例を示す。



図 3.6 システム(Unity)に読み込まれたこうかとんモデル(正面)

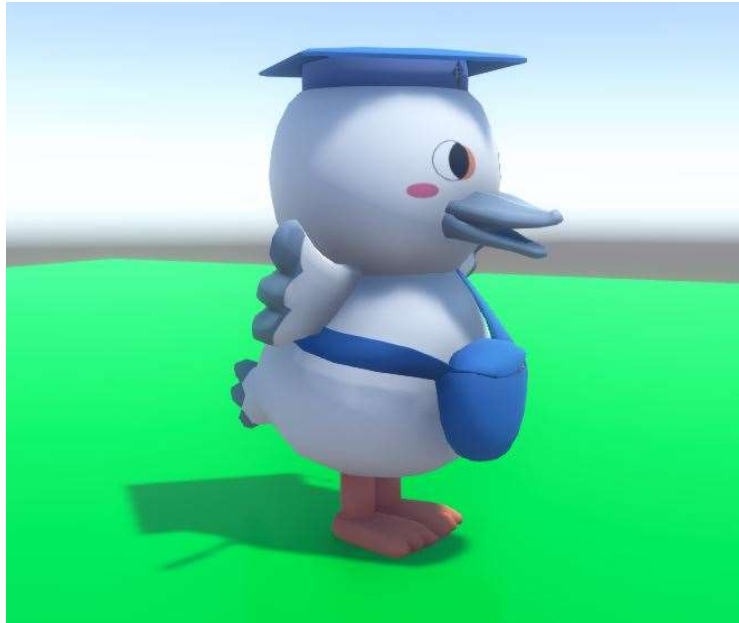


図 3.7 システム(Unity)に読み込まれたこうかとんモデル(側面)

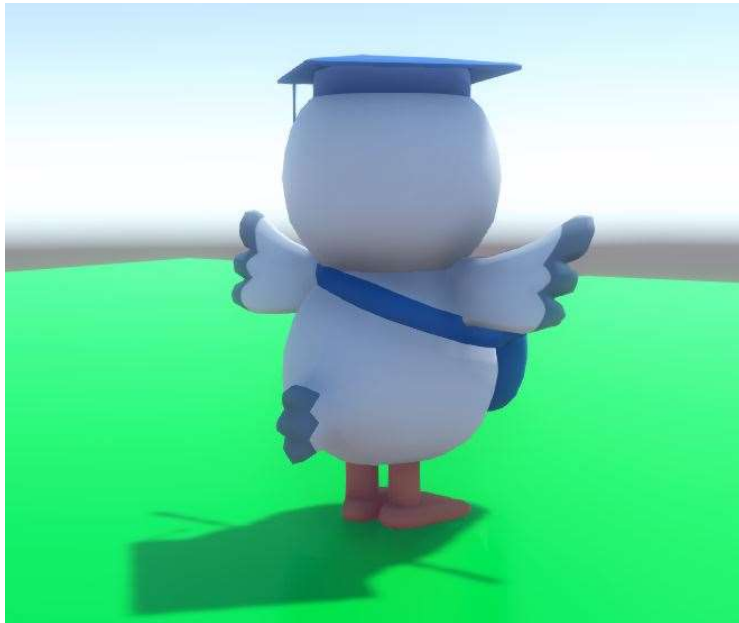


図 3.8 システム(Unity)に読み込まれたこうかとんモデル(背面)

3.3 アナウンサーモデルの制作

本研究では，ニュース映像に登場する人物として，大人の男女，大学生くらいの若者の男女を制作する．これは，一般的なニュースを配信する場合と，大学，学部の紹介など広報に使う場合で，ニュースを視聴する対象者に，より親しみを持ってもらうことができるようにするためである．ここでは，制作したアナウンサーモデルについて説明する．

3DCG モデル作成は演習講師の早川大地氏の指導の下，藤原源誠さんに依頼した．モデル制作には Maya を用いてキャラクターデータを用意した．図 3.9 から図 3.11 にモデル制作結果を示す．図 3.12，図 3.13 にニュース番組の背景と一緒に表示した例を示す．

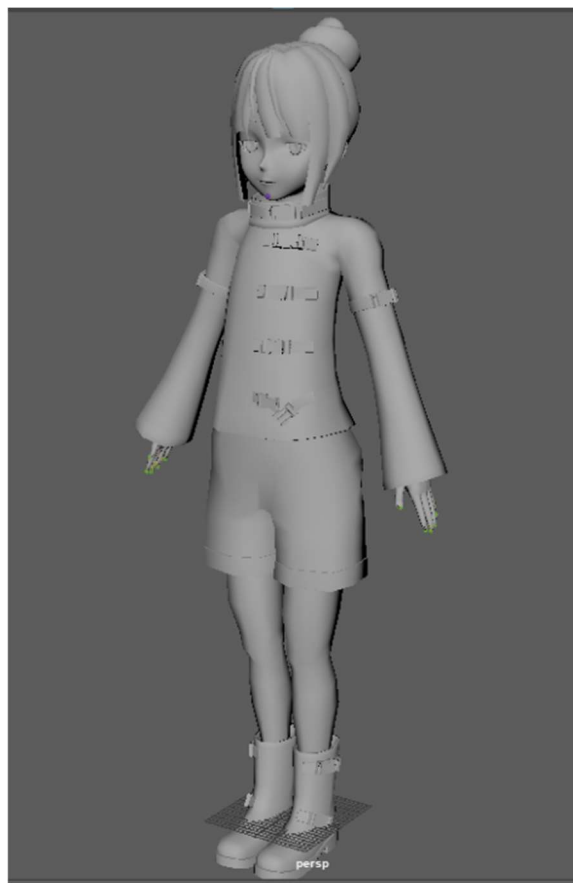


図 3.9 女子学生のキャラクター

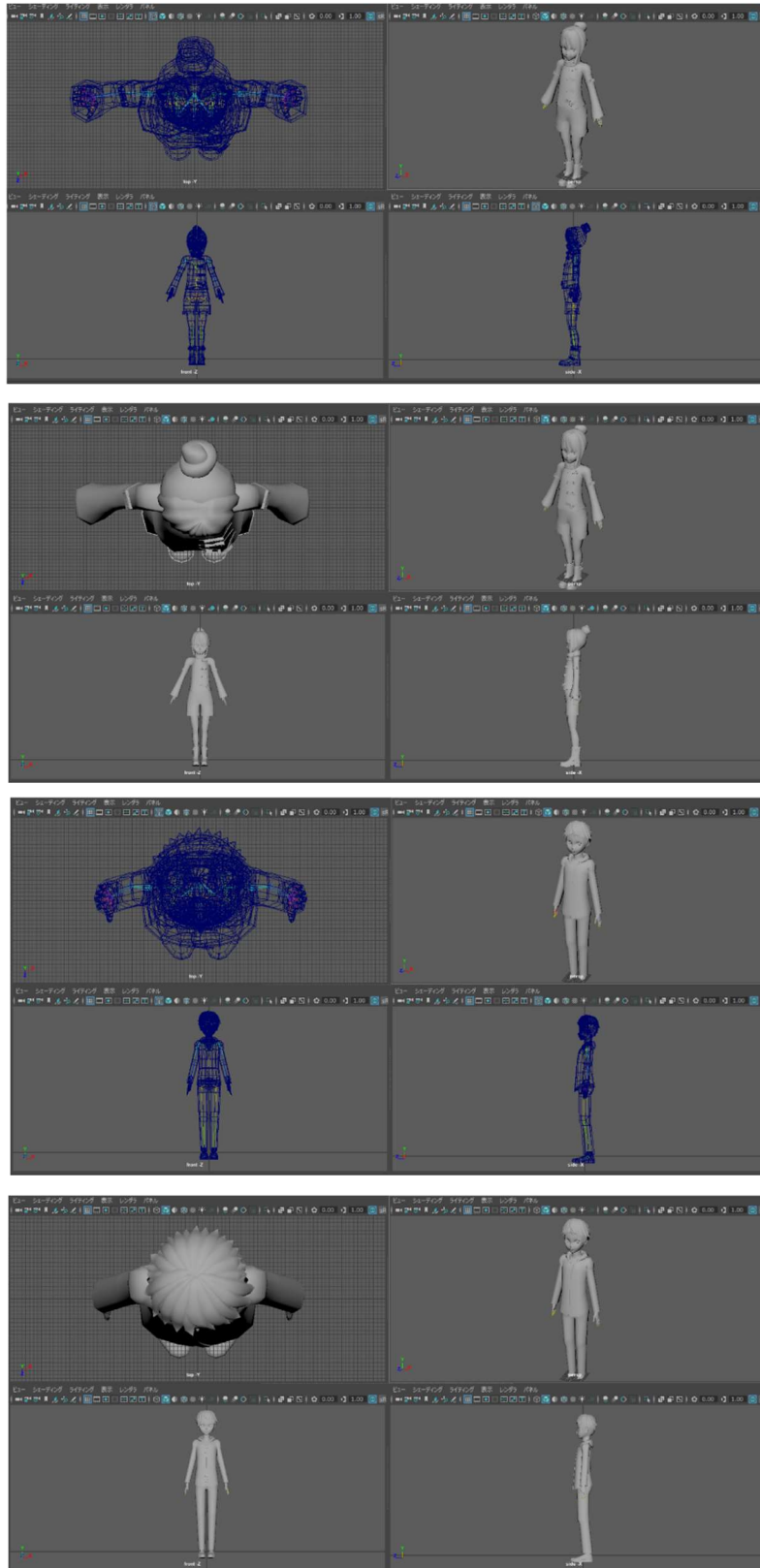


図 3.10 女子学生と男子学生のモデル



図 3.11 男子学生, 女性, 男性のキャラクターモデル



図 3.12 男子, 女子学生のキャラクターモデル



図 3.13 男性, 女性のキャラクターモデル

3.4 TVML におけるスタジオ空間の設計

本研究では、TVML で再生される番組のスタジオセット背景として、できるだけ実際のテレビ番組スタジオに近いデータを用意することとした。CAD の発展に伴って、テレビスタジオの設計デザインにおいても、「ベクターワークス」⁴などの、CAD ソフトが用いられている。これらのソフトが扱う 3D データを変換することで、TVML で再生される番組の空間を完成させることとした。

3.4.1 テレビスタジオ設計における CG の活用

テレビ局における番組制作において、スタジオ空間のデザインは、番組のスタイルや雰囲気を決める重要な要素である。そのために、ディレクターやスタッフが、デザインのイメージ・スケッチをもとにして検討を進める。この時点で、設計データをもとに、リアルな CG 画像があれば、イメージを共有して意見交換を行うことが容易となる。



図 3.14 スタジオセットのシミュレーションの例

⁴ ベクターワークス : MiniCad (ミニキャド) という汎用 CAD をもとに開発された。2D・3D・プレゼンテーション・データベースなどの豊富な機能を使って、空間デザインを総合的におこなうことができる。テレビ局においても、スタジオセットのデザインに用いられている。

3.4.2 3D デザインデータの変換

本研究では、これらのように実際にテレビ局で使用されている、デザインデータを、TVML が現時点で使用している、ゲームエンジン「Unity」が扱える 3D データに変換するテストを行なった。「ベクターワークス」などの CAD ソフトや一般の CG ソフトウェアでは、3D モデルデータをさまざまな形式で表現することができる。自由曲線による形状であっても、実際に最終的なレンダリングの段階で、一度ポリゴンデータに変換して、それを各種レンダリング・プログラムによって最終画像を描画する。それに対して、「Unity」のように、常にリアルタイムに 3D 画像を表示するシステムにおいては、こうした方法では「ひとコマ」の描画に時間がかかりすぎる問題がある。そのため、各パーツのデータは、一度ポリゴンデータに変換した上で、法線ベクトルの向きを外側に揃え⁵る必要がある。

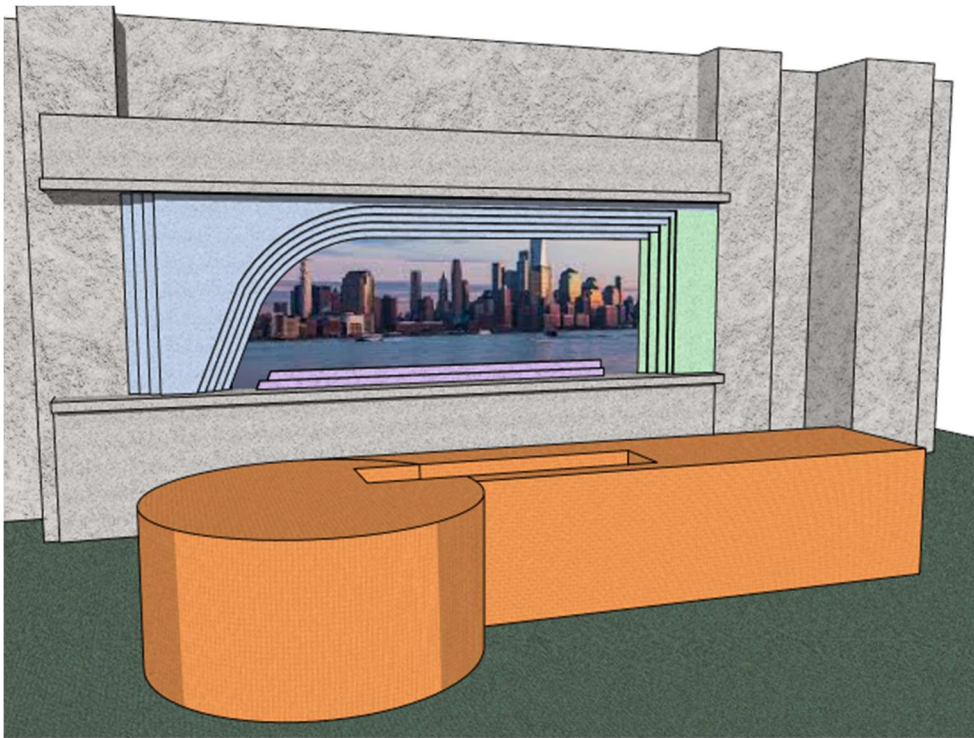


図 3.15 スタジオセットの表示の例

⁵ 一般的な CG ソフトウェアでは、法線ベクトルの向きが内側に向いていても「面」として認識して描画することがある。これに対して「Unity」などのゲームエンジンにおいては、法線が内側を向いた面は認識されないことが多い。

3.4.3 表面素材データ

さらに、変換前の 3D データに、表面の材質を表現するマッピングデータが設定されている場合には、これも事前処理として変換する必要がある。「Unity」などのゲームエンジンでは、表面材質データの計算時間の節約のために、ひとつのポリゴンに対してひとつのマッピングデータにまとめる手法がとられる。さらに言えば、3D ポリゴンにおける、おのこのの面に対して、マッピングデータの部位を厳密に対応させる (UV マッピング) 必要がある。CG ソフトウェアによっては、これらの作業を「ベイク」というプロセスで一括して行うことができる場合がある。しかし「ベクターワークス」においては、この機能がないために、別の手法をとる必要がある。

「ベクターワークス」においては、表面材質にマッピングデータを適用する場合は、以下の三種の方法を選ぶ。1：表面座標系 2：球面座標系 3：立体座標系である。この中で、Unity と互換性がとれる座標系は、最後の立体座標系のみであるということがわかった。それ以外の指定となっている座標系は、マッピングデータを再設定必要がある。

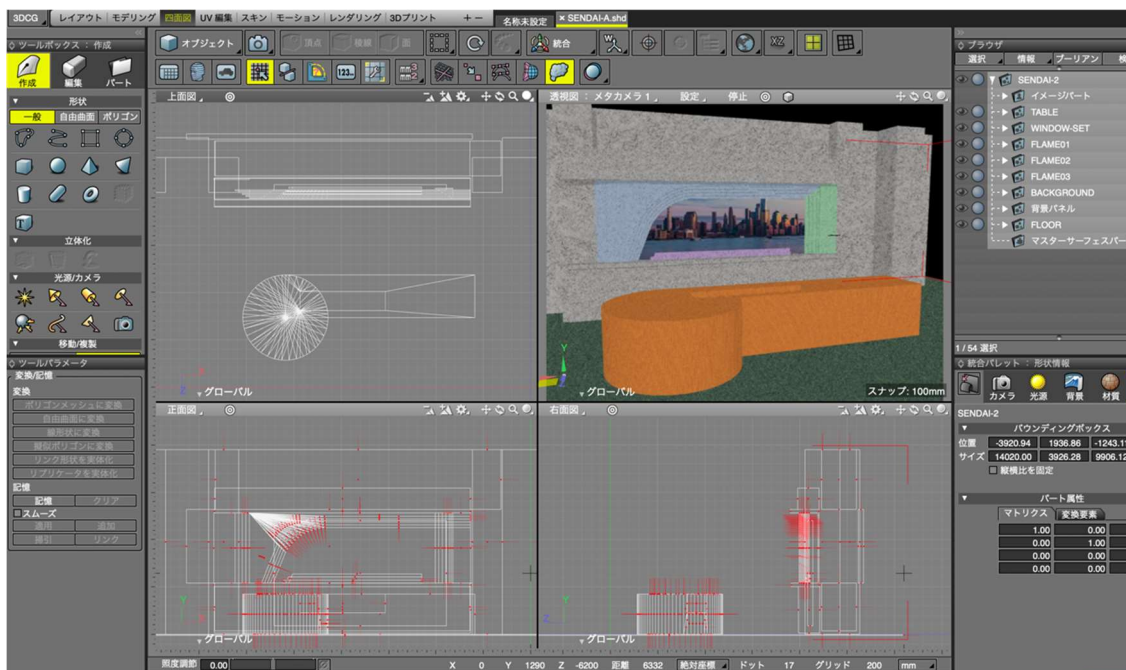


図 3.16 法線ベクトルの向きやマッピングデータの調整画面

3.4.4 スタジオ・デザイン・データの導入

上記のようなプロセスを経ることで、通常のテレビ番組のセットデザインに使用されている、スタジオデザインデータを、本研究の基幹システムである TVML、あるいは「TGPlayer」のデータとして、導入可能であることがわかった。将来の本格的利用のためには、一括して「Unity」などのゲームエンジンに対応する形での変換が必要である。しかし今後は随時、一定の手順にてデータ変換を加えることで、本研究における番組自動生成プロセスに活用することができる。以下は、本研究用に制作されたデータの一覧である。いずれも現時点では、「ベクターワークス」によるデータである。

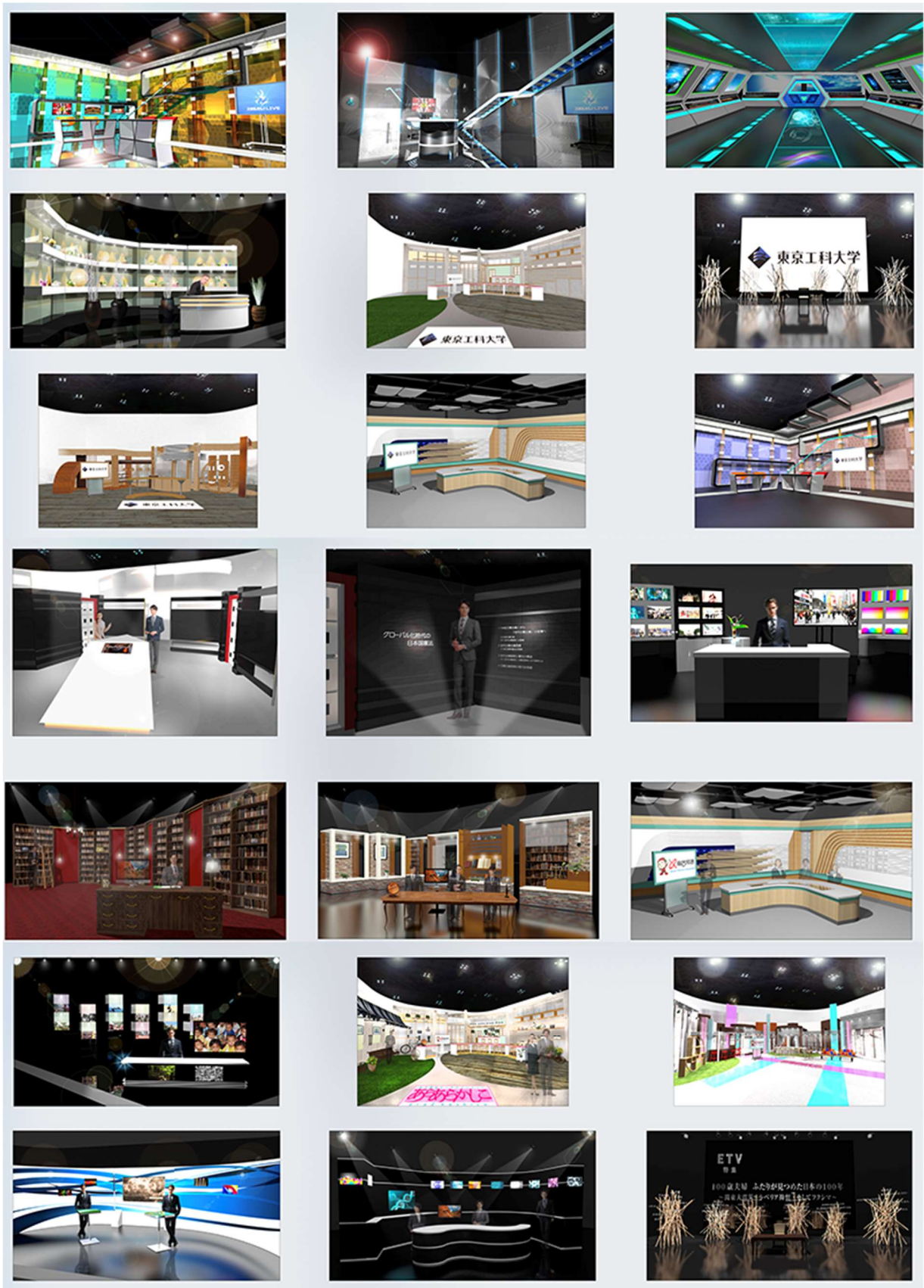


図 3.17 本研究用に制作されたスタジオデータ

4. Wordpress ブログからの TVML 番組自動生成

4.1 はじめに

第 2 章で述べたブログ記事を自動的に CG アニメーションにする TVML システムでは、ターゲットとなるブログ（東京工科大学メディア学部ブログ）からの必要な情報（記事タイトル、本文、日付、画像など）は、メディア学部ブログを記述する HTML に個別にチューニングして作成された Web Scraping（HTML を解析し、必要な情報を抜き出す操作のことをいう）を使って抽出した。こうして抽出された情報に対して自然言語処理をして、その後 TVML に変換して映像化を図った。

一方、メディア学部ブログは、商用のココログ（運営元：ニフティ株式会社）が用いられているが、ココログ側の方でブログシステムにアップデートがかかったとき、これまで使っていた Web scraping ソフトがそのまま使えず、例えば、ブログの本文を取得できない、などといった問題が起きる可能性がある。しかも、すべてはココログ運営側マターであり、それが、いつ、どれぐらいの規模で起こるか予測できない。

ブログからの自動映像生成を実用レベルで完成させるためには、上述のようにいつ誤動作するか分からないシステムでは実用化には無理があるので、なんらかの形でこの問題を解決する必要がある。

そこで我々は、フリーのオープンソースのブログシステムとして、世界的に最も大きいシェアを持つ Wordpress に注目し、Wordpress で構築したブログを対象にして、これを自動的に TVML 映像化するシステムを構築した。本章では、本システムへ至った検討過程と、実際に構築したシステムの詳細、そして実験の結果について述べる。

4.2 ブログ映像化の方法論の検討

前節で述べたような、自動化の一貫性の問題を解決するための方向性は以下の 3 つが考えられる。

- (1) ブログの運営元と提携して、自動映像化の一貫性が保てるようにする。
- (2) 自動映像化アプリケーションに、HTML から TVML への変換プログラムを外部プラグインとしてすぐに入れ替えられる仕組みを設ける。
- (3) HTML 構造が変わらないブログシステムを使ってブログを運営する。

これらのそれぞれに長所と短所があるので、以下に考察する。

1. ブログ運営元との提携

(1)の方法は、今回のメディア学部ブログにおいてはココログの運営元であるニフティ株式会社と何らかの形で提携して、ブログシステムの更新時にも、自動映像化のアルゴリズムに影響がないように配慮する、というものになる。方法として一番確実なことが、ブログの HTML の中に映像化のための特殊なタグやコメントを埋め込んでもらい、アップデートでもこれがきちんと残るようにする。たとえば、本文の HTML のところに下記のようにコメントを入れてもらうなどの方法が考えられる。

```
<!-- 本文：TVML 映像化用，ここから -->  
今日は八王子の東京工科大にお邪魔しています。  
<!-- 本文：TVML 映像化用，ここまで -->
```

あるいはコメントの代わりに XML のタグを定義して、これでもくくっても良い。TVML 映像自動化のために HTML に特別な XML タグを追加する手法は、過去の研究においていくつか見ることができる。

この方法は確実な方法で、自動化の一貫性は保てるが、当然ながらブログ運営元ときちんと提携しておかなければならない。そのためには運営側のメリットを提示し、運営会社を説得できない限り無理である。また、仮にそれができたとしても、運営会社が大きな会社であるほど、現実問題として、将来にわたって安定に提携して、先方のブログシステムの変更を行わせることは容易ではない。

2. アプリ側のプラグイン機能

(2)の方法は、ユーザ側が使う自動化アプリケーションの中の、HTML/TVML 変換部分を固定的にせず、プラグインの仕組みで変更できるようにしておく方法である。ブログ側の HTML 構造の仕様が変わったら、そのつど、新しい変換プラグインを作成し、これを即座に入れ替えて使えるようにする。ユーザ側でプラグインをインストールする手間を省くためには、ネットのサーバー上に最新版プラグインを置いておき、これを参照してアプリが動作するようにしておく。また、ある時、正しく映像化されなかったときは、ユーザがその不具合をアプリ元に即フィードバックする仕組みを組み込んでおき、アプリ提供側はその不具合通知を受けて、プラグインを更新するようにすればよい。完璧な方法ではないが、アプリ提供元がアクティブである限り、現実的な方法である。

さらに、このプラグイン提供の仕組みを一般ユーザに対しても開放して、誰でもプラグインを作れて、それをネットのサーバーに登録することができるようにすることで、コミュニティのボランティアによって、自動映像化できるサイトの数を増やして行くことができる。この場合、ブログだけでなく、ニュースサイトや、情報サイトやその他のサイトの自動映像化を推進できる可能性もある。

一方、この方法は、当のプラグインを更新するデベロッパーがいない限り成り立たず、しかも、いつ起こるかわからないサイトの HTML 構造変更に伴うプラグイン更新を、すみやかに実行してくれる開発元、あるいはボランティアが継続的に存在しない限り、自動化アプリはすぐに陳腐化する。すなわち、以上の機能を果たすコミュニティが常にアクティブな状態になっていないと、自動化アプリの動作が保証されないという欠点がある。

3. 特別なブログシステムの利用

(3)の方法は、ブログ側に固定的なブログシステムを採用する方法で、本研究ではこの方法によって構築した。そのため、方法 1 のように営利団体が運営するブログシステムではなく、ブログシステム自体がユーザによって自由に改変可能な形になっている必要がある。オープンソースのブログシステム（一般に、これを CMS: Content Management System と呼ぶ）であれば、自動映像化する開発者が、ブログシステム自体に手を入れて、前述の例のように、特殊なコメントやタグを埋め込んでおけばよい。オープンソースの CMS には、Wordpress, Joomla, Drupal など多くあるがどれを使っても同じことができる。

一般に、オープンソース CMS の世界では、ブログの外観デザインや機能をパッ

ページにした「テーマ」が複数存在し、これを自分のブログ用に選択して適用する。このテーマは、ソースコードと仕様が一般ユーザに公開されており、テーマ自体を個人が作成し、ネット上に登録して公開することができる。このテーマのソースコードに手を入れて、自動映像化用の新しいテーマとして登録すれば、自身のブログにこのテーマを選択することで、おのずと自動映像化が可能な状態になる。

一方、ブログを始める人間が、特定の CMS の特定のテーマを選択することが必要になることはいうまでもない。一般的なブログを自動映像化するわけにはいかない。

以上、三種類の方法について長所と短所を述べた。本研究では、3の方法を選択した。使用する CMS としては、世界でのシェアが最も高い Wordpress を選択する (<https://www.onblastblog.com/joomla-vs-wordpress-vs-drupal/>)。

Wordpress は PHP をベースに構築されており、特定のテーマの PHP コードの中に、映像化のタグを埋め込むことは比較的容易である。本研究では方法3により、Wordpress を選択して開発を行ったが、前述の三種のハイブリッドなシステムやアプリを構築することはもちろん可能であり、今後、もっとも望ましい状態に向けて検討を進める必要がある。

4.3 WORDPRESS を使った CG アニメーションの自動生成

本章では、Wordpress による自動映像生成システムの詳細を述べる。

図 4.1 に本システムの構成を示す。ブログの記事の HTML をインターネット経由で取得し、この HTML を Web scraping して、タイトル、本文、画像の URL などを抽出する。次に、これらの情報を元 TVML Script Generator により TVML スクリプトを生成し、これを TVML エンジンによって CG アニメーションに変換する。

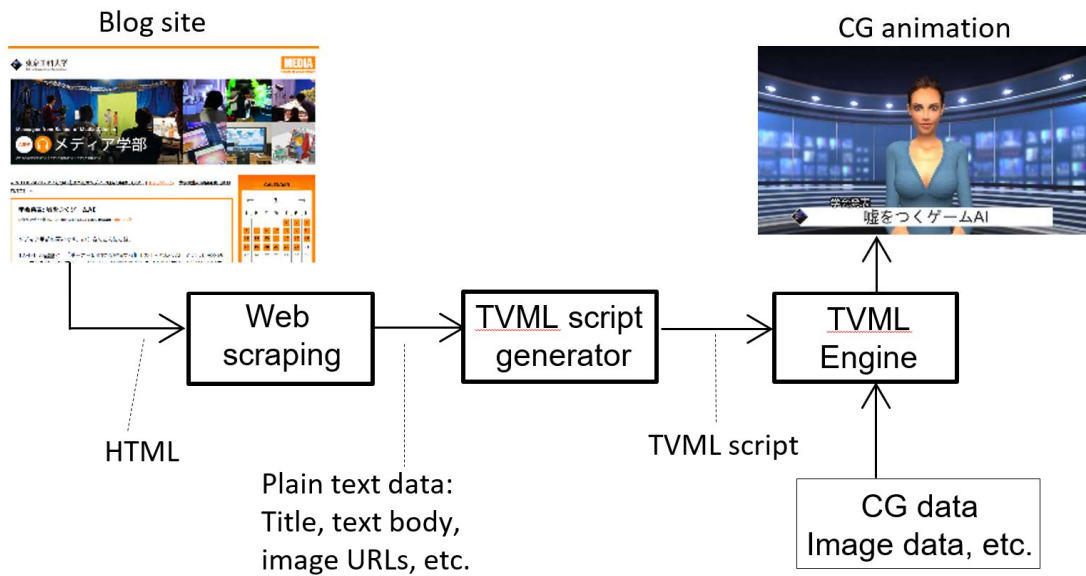


図 4.1 ブログの自動映像化システム

Wordpress のテーマには、Wordpress コミュニティが提供する公式テーマの “Twenty Fifteen” を用いた。前述したように、Wordpress のテーマは PHP で書かれており、このソースコードは、Wordpress の管理画面から変更可能である。2-1 節で述べたように、タイトルや本文など必要な情報を Web scraping で取得するために、特別なコメントやタグを PHP コードに埋め込むこともできるが、本研究では、テーマ “Twenty Fifteen” の PHP ソースを読み、タイトルと本文、および画像を示す特定のタグが使われていることを確認したので、それをそのまま使い、PHP ソース自体の変更は行わなかった。“Twenty Fifteen” の Web scraping のためのタグ一式を表 4.1 に示す。

以上の Web scraping により抽出された、記事のタイトルと、本文と、画像の URL から TVML Script Generator により TVML を生成するが、このモジュールが、TVML 番組映像の演出を決定している。具体的には、オープニングタイトルの付与、スタジオセット、CG キャラクタに何をを使うか、カメラアングル、スーパーインポーズのデザインや表示場所、画像の表示の仕方、などがこのモジュールにより指定されることになる。逆に言うと、このモジュールを変更することで多彩な演出でブログを映像化することができる。今回は試作システムとして、このモジュールは固定的なものを使用しており、このモジュールを変更可能にする仕組みについては今後の課題とする。

表 4.1 テーマ”Twenty-one”の Web scraping

抽出対象	タグ	備考
記事	<article>	複数の記事があり、一つの記事ごとに<article>タグで囲まれている
タイトル	<h2>	
本文ブロック	<div class="entry-content">	まず、この div タグで本文全体を抜き出し、その後、以下の p タグ、a タグ、img タグで、段落ごとのテキストと、画像を抜き出す
本文の各段落	<p>	抜き出されたテキストの中に HTML タグが含まれている場合、これを削除する（多くの場合、本文中のハイパーリンク） また、段落の中の句読点と読点を検出し、テキストを適当な長さに自動分割する
画像	<a>	img タグとセットになっている場合、画像と判断する。href アトリビュートの値の URL を取得する

以上のメカニズムを備えた TGPlayer アプリケーションを Unity ゲームエンジンの上に構築した。アプリケーションを起動すると、初期設定ファイルを読み込み、そこに記載されたブログの URL をアクセスし、前述したプロセスにより記事の取得、TVML スクリプト生成を行い、TVML 番組として再生する。また、テレビ番組的な感じを演出するため、画面の左肩にリアルタイムのデジタル時計表示の機能を組み込んだ。図 4.2 に、Wordpress により構築されたブログサイトと、TGPlayer でこれを再生したところのスクリーンショットを示す。

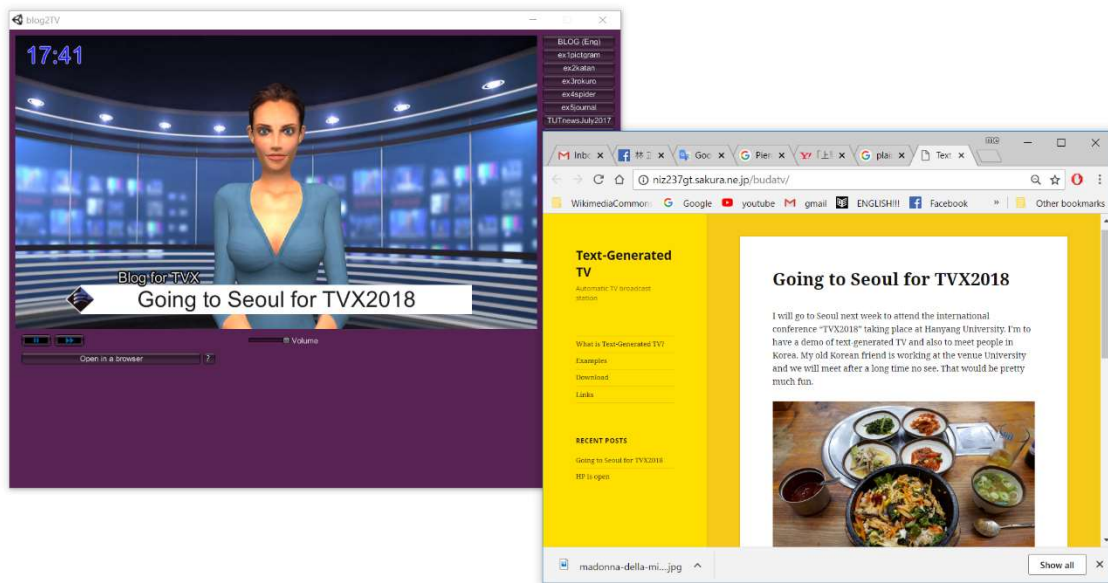


図 4.2 ブログの映像化スクリーンショット
(右が Wordpress ブログ, 左が TGPlayer)

4.4 映像化ブログの制作方法

ここでは、TVML 自動映像化が可能な全体システムの構築の方法について述べる。

(1) Wordpress ブログの構築

まず、Wordpress によるブログの構築を行う。Wordpress は Wordpress.org (<https://ja.wordpress.org/download/>) のダウンロードページからダウンロードできる。本研究は Version 4.8 を用い、実験用のレンタルサーバー（さくらインターネットを使用）にインストールした。以下の URL に英語用と日本語用の二つのブログを構築した。

<http://niz237gt.sakura.ne.jp/budatv/> （英語用）

<http://niz237gt.sakura.ne.jp/budatv/jp> （日本語用）

なお、ブログの日本語／英語の切り替えは、Wordpress の設定／一般／サイトの言語のところ簡単に切り替えることができるので、どちらのサイトも、同一

の Wordpress をインストールすればよい。言語を切り替えると、メニューなどの言語がすべて該当言語に変更される。

次にテーマを選択する。4.3 節で述べたように、Wordpress の設定画面で、テーマとしてここは“Twenty Fifteen”を選択する。本研究では“Twenty Fifteen”のバージョンとして“Version 1.2”を用いている。このテーマのバージョンには注意を要する。テーマのバージョンが更新されると最悪の場合は Web scraping がうまく行かなくなる可能性がある。テーマ自体を更新するときは、PHP のソースコードをチェックして HTML タグ構造が変わっていないかどうかチェックする必要がある。ただし、テーマのバージョンは自動更新されないの、自動映像化の目的で構築した Wordpress ブログは、基本的にテーマの更新を行わないで運営していれば問題は起こらない。また、ブログの機能としても古いバージョンのテーマを使用していても大きな問題は無い。

一方、Wordpress 本体のバージョンだが、本開発では Version4.8 を用いている。Wordpress バージョンの更新は、基本的には自動映像化のための Web scraping には影響しないはずだが、Version5.0 以降で注意を要する。Version5.0 においてブログ記事投稿のエディターが Gutenberg と呼ばれるブロックエディターに変更され、そのため、画像のタグの扱いが変わっており、自動映像化のための Web scraping が誤動作し、画像が出ない現象が発生する。これを避けるためには、Gutenberg を使わず Classic editor を使うか、あるいは TGPlayer 側の対応が必要になる。根本的な対策としては、テーマの“Twenty Fifteen”をそのまま使わず、PHP ソースコードに手を入れて、自動映像化のための独自識別タグを挿入すべきである。このようにして得た新しいテーマに名前を付け、自動映像化用テーマとして、Wordpress のコミュニティに新規登録することで、Wordpress およびテーマのバージョンによる不具合は完全に回避できる。これは今後の課題である。

(2) 記事の制作

ブログの記事の制作は通常の Wordpress の方法と同じである。管理者特権を一人とし、「編集者」特権を持つ複数のユーザを登録し、複数のブログライターがブログ記事を更新できるようにした。編集者特権ユーザは、記事を自由に投稿・更新・修正ができるが、ブログそのものの変更はできない。たとえばテーマ自体を変更したり、Wordpress 本体やテーマのバージョンを更新するなどの操作

はできない。これらは管理者特権ユーザのみ可能で、自動映像化の機能に影響を及ぼすので、管理者と編集者は分離した方がよい。

記事の作成については、タイトルを入力し、本文を入力する。本文中には画像を適宜挿入する。図 4.3 に示すように、画像を掲載した場所の次の段落に記述されたテキストが、CG アニメーションの方では、画像表示のバックで読み上げられる。段落を変えると、その時点で画像が消えてCG キャラが登場するスタジオショットに切り替わるようになっている。

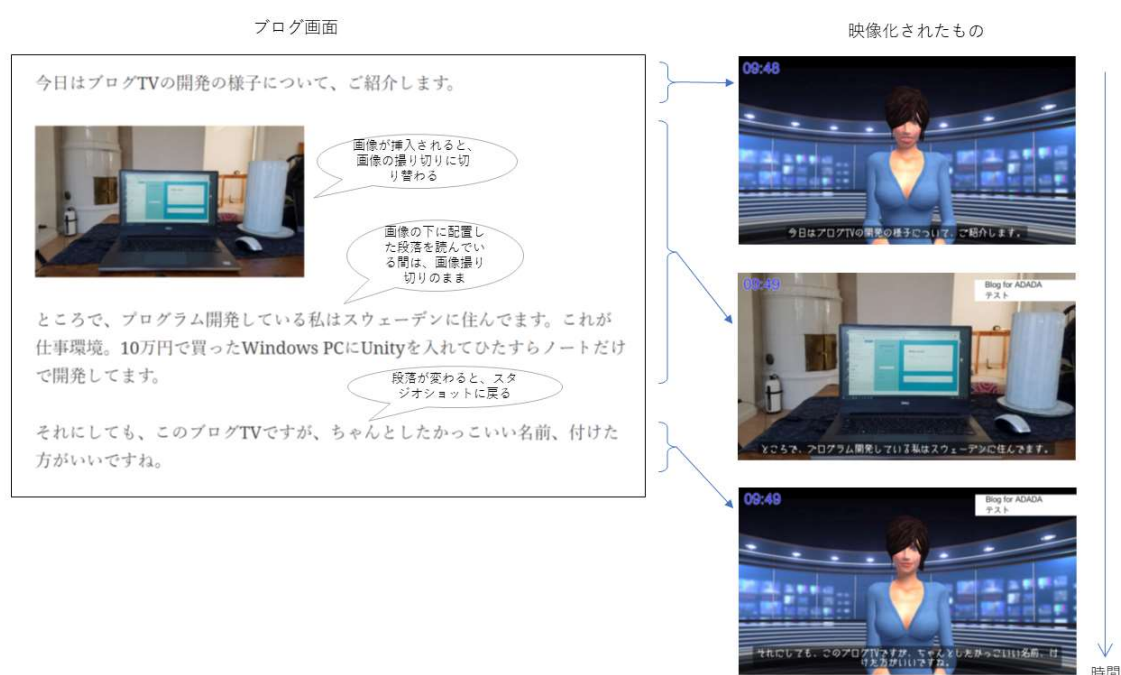


図 4.3 ブログ中の画像の映像化の際の扱い

(3) 記事の再生

再生にはTGPlayerを用いる。TGPlayerを起動して、メニューのブログサイトのアイコンをクリックすることにより、そのブログサイトの最新の記事から、過去の記事に向かって、順次再生が始まる(図 4.3 参照) 本研究では、TGPlayerのメニューのアイコンは、ブログサイト単位で並んでおり、ブログサイトの選択に用いられている。図 4.4 に示すように、日本語用ブログと英語用ブログの二つを構築したので、二つのアイコンが並んでいる。



図 4.4 TGPlayer によるブログ再生結果

4.5 番組制作実験例

ここでは、国際会議 ADADA で発表を行った時のブログ記事例(図 4.5) と映像例(図 4.6) を示す。Wordpress による記事は次の URL で見ることができる。

<http://niz237gt.sakura.ne.jp/budatv/>

ブログ記事には、文章と写真画像が記事には含まれて、それが、ニュース映像では、表示されていることが分かる。簡易的な処理のために、ここでは、文章がそのまま表示されている。

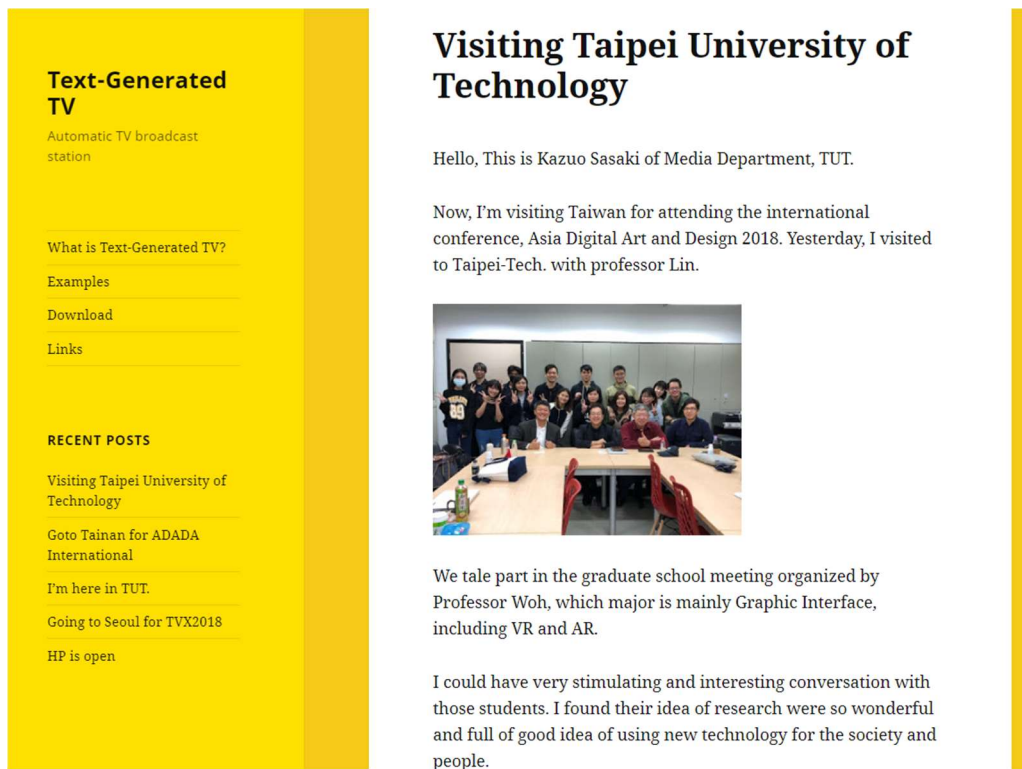


図 4.5 Wordpress の英語版ブログ記事例



図 4.6 図 4.5 のニュース映像例(1)

15:50



Blog(Japanese)

Blog(English)

15:50

Blog for ADADA
Visiting Taipei Unvers



We tale part in the graduate school meeting organized by Professor Woh, which major is mainly Graphic Interface, including VR and AR.

Blog(Japanese)

Blog(English)

15:50

Blog for ADADA
Visiting Taipei Unvers



I could have very stimulating and interesting conversation with those students.

Blog(Japanese)

Blog(English)

図 4.6 図 4.5 のニュース映像例(2, 3, 4)

4.6 評価とまとめ

ニュース番組制作実験を行ったところ、スムーズに運用ができ、TGPlayer による自動映像化も問題なく動いている。国際学会などでデモ発表したところ、「ブログを番組で見る」という発想の新しさに感心する来場者が多く、その大きなポテンシャルについて高評価を得た。

それまでの、一般のブログサイトを対象に行っていた自動映像化システムには、ブログサイトごとに Web scraping の手法を用意しないといけない、ブログの運営者が CMS を予期せずに変えると動かなくなる、といった大きな問題があったが、本研究によって、CG アニメーション自動変換が可能な CMS システムを設計して、提供する方法を採用することで、以下のメリットを得ることができた。

- 1) 自動 CG アニメーション生成が保証される
- 2) 動画再生ボタンなど特別な昨日を CMS に埋め込んで、ユーザの利便を図れる。
- 3) CMS を逆に、CG アニメーション生成ユーザインターフェースとして使うことができる。
- 4) CMS のコミュニティの活性化につながる。

本実験で、我々は CMS として Wordpress を取り上げ、CG アニメーション用の特別なテーマを作成し、これを提供する考え方を示した。特に上述(4)のように、今後、世界的に大規模な Wordpress コミュニティへ訴求することで、認知度を上げ、Wordpress コミュニティへの貢献をして行きたい。

本実験は、ビューワーとしてスタンドアローンの TGPlayer で行ったのでインストール作業が必要である。TGPlayer が Web ブラウザ内で機能するようにするべきであろう。TGPlayer は Unity ゲームエンジンで実装されており、Unity はマルチプラットフォームなので、Web ブラウザでの起動はそれほど困難ではない。また、PC のみならず、Mac や iPad, Android タブレット、スマートフォンなどで使用できるようにすることも重要である。本研究では、PC に加えて、Android で動く TGPlayer も合わせて開発し、Android スマートフォンやタブレットでの TGPlayer の使用も行えるようにした。これについては次章で詳述する。

また、本実験方法では、Wordpress のブログサイトを立ち上げられるユーザしか利用できないが、逆に、これを出発点にして、ネットサービスとして提供されているブログに本機能を導入するように推奨できるであろう。本研究で制作実験をしてみて分かったが、ブログを CG アニメーションで見ることができると同時に、CG アニメーションを制作するツールとして本自動映像化ブログシステムを用いる、という考え方にも可能性を感じた。本手法を使えば、Wordpress のテーマとして、例えば、ニュース番組制作用テーマ、語学番組制作用テーマ、討論番組制作用テーマという形で用途に応じたテーマを複数用意し、提供することができる。CG アニメーション制作の新しい形態として発展させて行くことも検討してみたい。

5. Android 版アプリ TG プレイヤーの制作

5.1 TG プレイヤーの概要

本章では、Blog 記事の自動情報番組化システムのコアアプリケーションである「TG プレイヤー」(TG: Text-Generated) の発展のために必要な、Android 版 TG プレイヤーの開発について述べる。

現在、Windows 版 TG プレイヤーが既に稼働しており、これを改良したうえで Android へ移植する。具体的な開発内容は、現在 Windows 上のローカルで稼働している TG プレイヤーをサーバーデータを参照して起動するように改良、TG プレイヤーの UI の改良、TG プレイヤーの Android への移植、の三点である。

TG プレイヤーは、インターネット上の BLOG の HTML を取得し、これを自動的に TVML 番組に変換して見られるアプリケーションである。また、そのほか、APE スクリプトと呼ばれる番組台本や、TVML スクリプトで直接作られた番組を、同様に再生することができる。

TG プレイヤーにおけるデータの流れを図 5. 1 に示す。

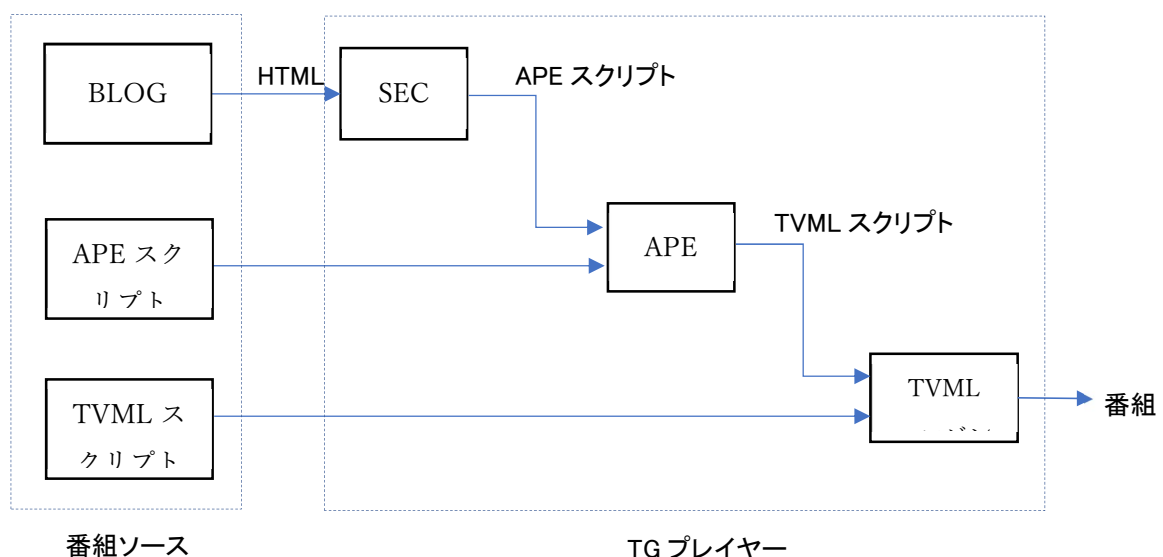


図 5.1 データフロー

図 5.1 において、APE スクリプトは XML で記述された「番組台本」に近い形式で記述された台本であり、TVML スクリプトは TVML 言語仕様に基づいて記述された台本である。SEC は TG プレイヤー内の C#プログラムで、与えられた HTML を APE スクリプトに変換するモジュールで、APE は APE スクリプトを TVML スクリプトに変換するモジュールで TVML テンプレート群を参照することで動いている。すべての番組は、最終的に TVML スクリプトに変換され、TVML エンジンによって CG アニメーションに変換される。

Windows 版 TG プレイヤーでは、図 5.1 のフローにおける番組ソース、SEC エンジン、APE エンジン、TVML エンジンのすべてがローカルのファイルシステムの上で動いている（BLOG の実体はサーバー側だが、その URL はやはりローカル側に書かれている）。ローカルディスク上において、図 5.2 に示すデータ群が定義されており、このデータを TG プレイヤーのアプリケーションが読み取って、番組再生を行っているためである。

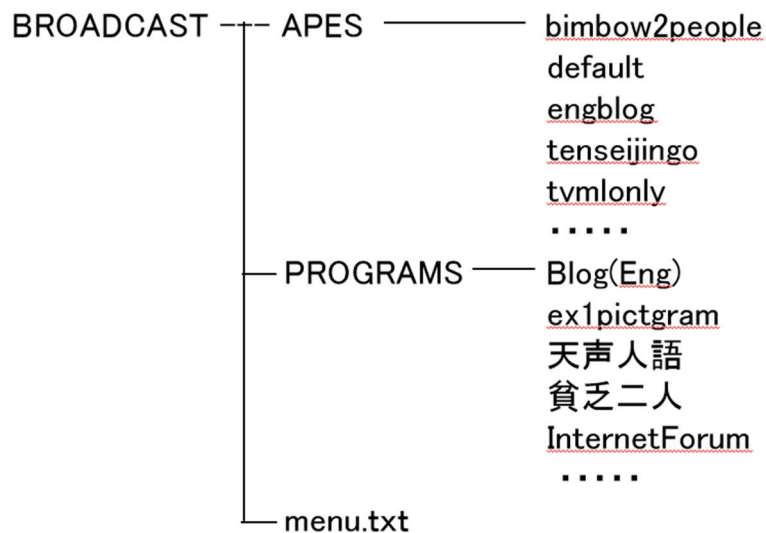


図 5.2 番組関連データのディレクトリ構造

図 5.2 において、APES フォルダの下にある各種フォルダの一つ一つが APE エンジンに相当し、これらフォルダの中には、APE エンジンによる変換に必要な TVML テンプレートと画像や音などの必要データが入っている。また、PROGRAMS

フォルダの下にある各種フォルダが、一つ一つの「番組」に対応し、このフォルダには APE スクリプトと番組再生に必要な画像や音などのデータが入っている。BROADCAST フォルダの下には、APES と PROGRAMS のほかに menu.txt というテキストファイルがあり、これは TG プレイヤーのメニュー画面に表示される番組のリストを記述したものである。

5.2 アプリの仕様と機能

以下の三点についてそれぞれの仕様について述べる。

(1) TG プレイヤーのサーバー対応

- 現在の TG プレイヤーは、ローカルディスク上の BROADCAST フォルダ以下にあるデータを読み込んで動いている。これを、そのままのデータ構造でサーバー上に置き、それを読み込んで動くように改修する。
- ローカルフォルダはそのまま残し、TG プレイヤーは最初にサーバーをアクセスし、アクセスできなかったときにはローカルデータを参照する構造とする。
- サーバー上のデータ特定は、サーバー上の BROADCAST フォルダの URL を指定することで行う。URL はコード中にハードコーディングしてもよい。

(2) TG プレイヤーの UI 改良

- 図 5.3 に示すように、現在の UI は、表示スクリーン、コントロールパネル、番組メニューの 3 つにより成っている。

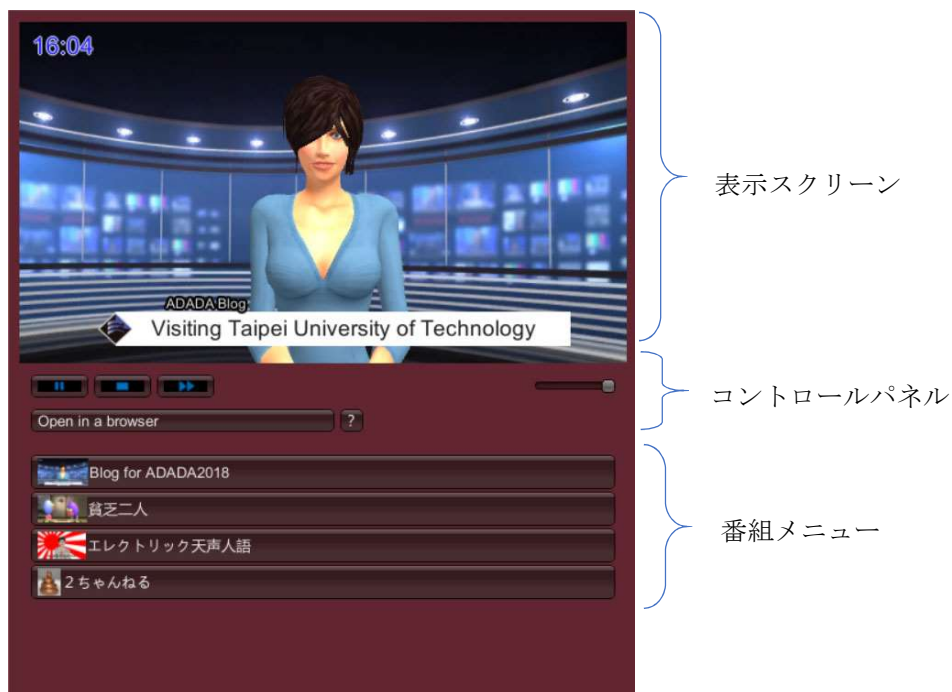


図 5.3 現行の TG プレイヤーの UI 画面

- 以下の二点について改修を行う
 - (1) コントロールパネルを表示スクリーン上に移動し、マウスホバーしたときにコントロールパネルが表示スクリーン上に表示されるようにする。
 - (2) 番組メニューをスクロールタイプに対応し、番組数が増えても問題ないようにする。

(3) TG プレイヤーの Android への移植

- 前項で規定したサーバー対応と、改良した UI を持つ TG プレイヤーを、Android へ移植する。Android 版 TVML SDK を利用する。
- Android 版 TVML SDK のバージョンが、支給する Windows 版 TVML SDK よりバージョンが古いので、まず最初に Android 版 TVML SDK のソースコードを最新版にアップデートしたのち、TG プレイヤーの移植作業に入る。本 SDK のアップデート作業を行う。

- Android 版 TG プレイヤーでは、モバイルのタッチパネル端末を使用するため、以下の機能を有している必要がある。

(1) CG 再生画面において画面をタッチすると、コントロールパネルが全面に浮き出てきて、そこでタッチして操作できる。

(2) 端末の横置きと縦置きに反応してサイズが動的に変わる。

(3) 横置き時は、番組メニューは消え、表示スクリーンが全画面表示される。

以上の機能を TG プレイヤーに実装するが、この際、UI については YouTube のモバイルアプリの UI に倣った形で実装する。

5.3 TG プレイヤーとその実行例

5.3.1 Android 版のスマートフォンとタブレットでの実行方法

実装したアプリは、図 5.4 に示すページからダウンロードできる。

<http://niz237gt.sakura.ne.jp/budatv/jp/>



図 5.4 TG プレイヤーのダウンロード先ページ

このサイトの「ダウンロード」のところで、Windows 版と Android 版のプレイヤー (TG Player という名前になっている。ここからダウンロードして、Wordpress で制作したブログの番組を見ることができる。現在は、試験使用として指定したブログのみであるが、今後、アプリの改良を行い、ユーザが指定した Wordpress で制作したページの番組を表示することができるようにする予定である。

本アプリは、Android のスマートフォンやタブレットで表示できる。図 5.5 にスマートフォンにおける実行例を示す。図 5.6 はタブレットにおける表示例である。タブレットを縦や横にしても表示可能である。

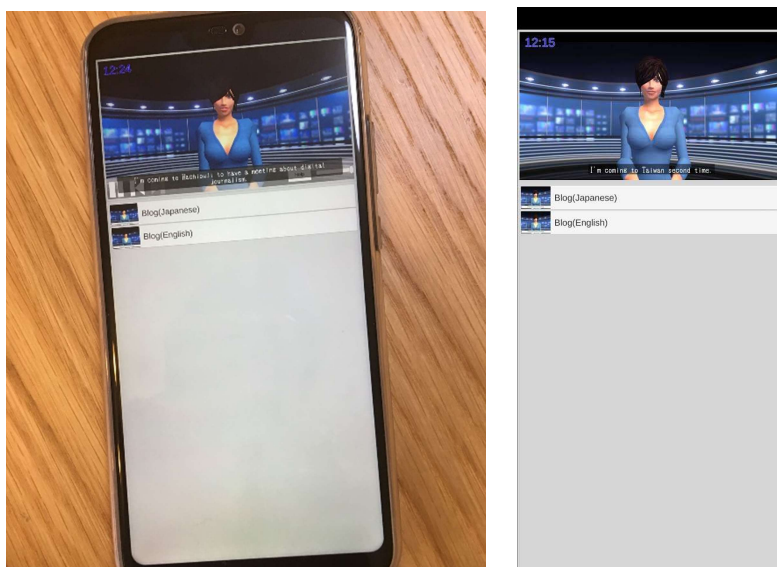


図 5.5 スマートフォンによる実行例

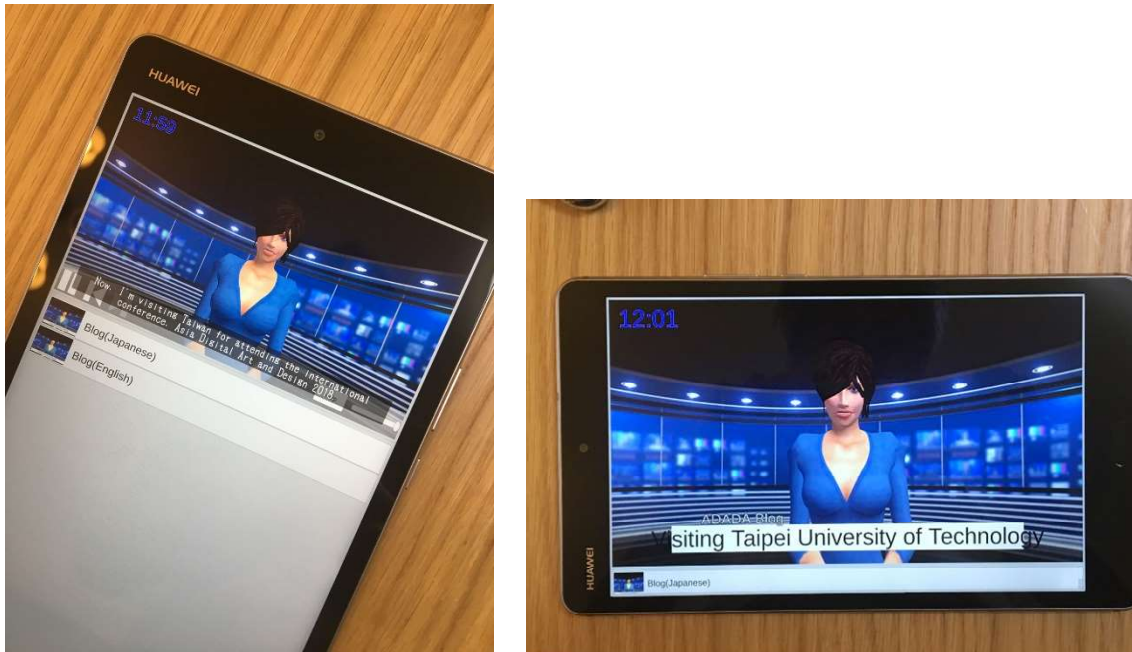


図 5.6 タブレットによる実行例

5.3.2 Wordpress の日本語記事の番組生成例

番組のための記事を Wordpress によって書いたページの URL は次の通りである.

<http://niz237gt.sakura.ne.jp/budatv/jp/?m=201904>

このページの番組表示例を図 5.7, 図 5.8 に示す. アナウンサーの表示, 記事の中の写真画像の表示, さらには, 記事のテロップ表示ができていることが分かる.

このページは, 本プロジェクトの評価用であり, 今後, さまざまな内容に応じて, ページを作成することによって, 実用的なページの公開とその番組表示が可能となる.



図 5.7 タブレットによる実行例（アナウンサーの表示）



図 5.8 タブレットによる実行例（記事のなかにある写真の表示）

6. まとめ

本研究の目的は、個人対応時代のための新たな報道の価値を作る「個人適応型ニュース配信」のシステム開発を目的とする。

そのために本研究では、次のことを行った。

1. インターネット情報の分析によるテキスト情報の要約制作のための言語処理法を提案した。
2. Web ページを入力して映像を出力するという機能を中心に文書表示・提示する映像配信システムを開発した。
3. 個人向け記事制作のために、CMS (Content Management System) のひとつである Wordpress を用いた Web ページを活用できるシステム、アプリを開発した。ユーザは、CMS で構築する Web 情報を入力して、ニュース映像を出力する「ニュースコンテンツ制作配信システム」を開発した。

本研究プロジェクトによるニュースコンテンツ制作配信システムの特徴は次のとおりである。

- (1) インターネット上の情報を解析して要約やテロップなどの報道情報を抽出し、それらを用いたニュース番組が制作できる。
- (2) インターネット上の Web データ、および、ユーザが制作した Web ページを用いて、ニュース映像として提供することができる。また、検索システム、たとえば、Google を使ってキーワードを入力すれば、それに関する Web ページが表示される。この時の URL を入力すれば、個人の意図に一致したニュース番組として表示できる。

本研究で制作した「こうかとん」の 3 次元モデルデータは、今後、利用方法を公開する予定である。

本研究は、TVML の開発者であるウプサラ大学ゲームデザイン学科 林 正樹准教授との国際共同研究として実施した。このような有用な成果を得られたことに深く感謝する。

7. 本研究に関する発表と展示

7.1 研究発表

[1] Naoya Tsuruta, Takehiro Teraoka, Kunio Kondo, Masaki Hayashi, TV Show Template for Text Generated TV, International Workshop on Advanced Image Technology 2018 (IWAIT 2018), 2018.1 (**Best paper Award**を受賞)

[2] Masaki Hayashi, Naoya Tsuruta, Takehiro Teraoka, Kazuo Sasaki, Wataru Usami, Koji Mikami, Tsukasa Kikuchi, Yuriko Takeshima, Kunio Kondo, Automatic Generation of TV Program from Blog Entry, ACM TVX2018, 2018. 6. 26-28

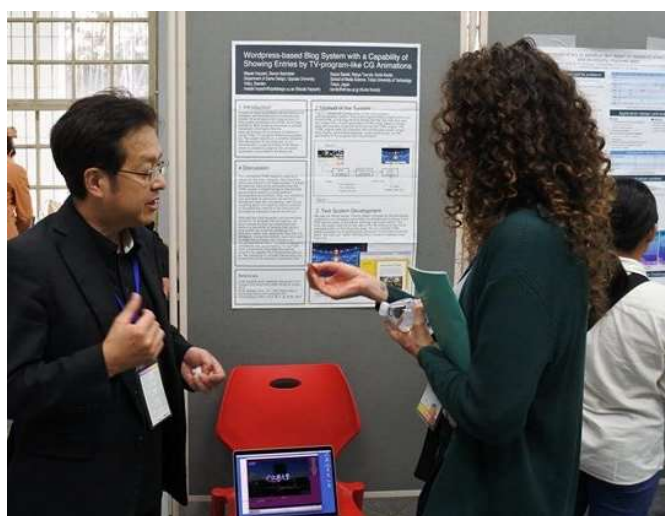
[3] Masaki Hayashi, Steven Bachelder, Kazuo Sasaki, Naoya Tsuruta and Kunio Kondo, Wordpress-based Blog System with a Capability of Showing Entries by TV-program-like CG Animations, The 16th International Conference of Asia Digital Art and Design (ADADA2018), Poster-2D-8, 2018. 11 (Sasaki が発表)



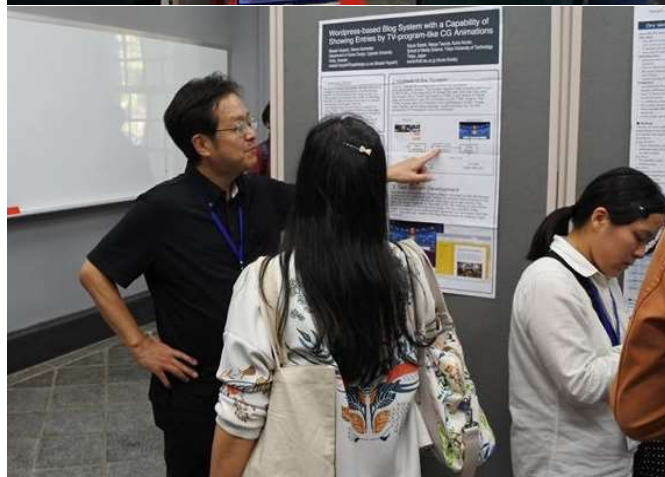
IWAITにおいて受賞した時の鶴田先生



IWAIT の表彰状



ADADA2019 のポスターセッション風景(佐々木先生の発表)

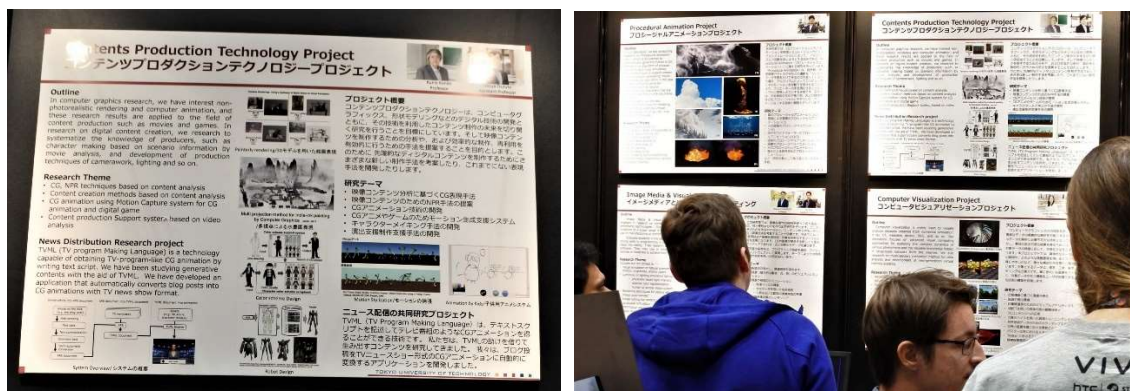


7.2 展示発表

[1] SIGGRAPH Asia2018 :

第11回 ACM シーグラフ, アジアにおけるコンピュータグラフィックスとインタラクティブ技術に関するカンファレンスと展示会は2018年12月4~7日まで東京都の東京国際フォーラムで開催された。この展示会において, メディア学部 of CG 関連研究を展示した。これに合わせて一緒に学内共同研究プロジェクトも展示を行った。世界59カ国より9,735人(内外国人参加者数2,323人)。国内外のデジタルコンテンツ制作者への研究プロジェクトの活動を紹介することができた。

<https://sa2018.siggraph.org/jp/>



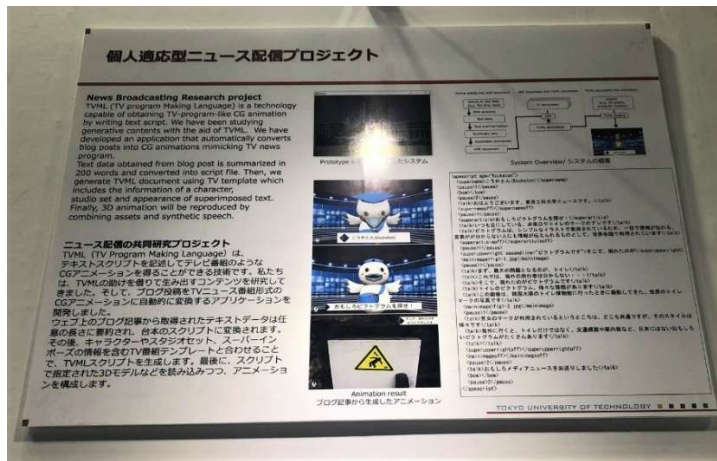
SIGGRAPH Asia2018 におけるプロジェクト紹介

[2] AnimeJapan 2019

東京ビッグサイトにて2020年3月21日(土)~24日(火)に開催された。

「AnimeJapan 2019では, 国内外を代表するアニメ関連企業・団体が多数出展し, アニメ作品展示・グッズ販売, ステージやイベントなどアニメのすべてが集結します」とホームページに書かれている。総来場者数 146,616人であった。

<https://www.anime-japan.jp/>



AnimeJapan 2019 における展示風景

[3] 第 25 回台湾大学芸術祭

メディア学部から三上浩司教授，太田高志准教授が出張し，学部ブログで報告を行った．以下，三上教授のブログ記事から引用する．

「台湾と交流のあった元八王子市長でもある黒須理事のつながりで，第 25 回台湾大学芸術祭に招かれ，日本工学院専門学校，日本工学院八王子専門学校，東京工科大学メディア学部，デザイン学部が招待展示をすることになりました．」

[1] 祝！「令和」初イベント「台湾大学芸術祭」（専門学校，メディア学部．デザイン学部共同出展） 2019 年 5 月 5 日 三上教授執筆
<http://blog.media.teu.ac.jp/2019/05/post-4f1320.html>

[2] いよいよ開幕「台湾大学芸術祭」続報 5 月 6 日 三上教授執筆
<http://blog.media.teu.ac.jp/2019/05/post-98b181.html>

[3] 灼熱の台湾（台湾大学アートフェスティバル，片付け編）5 月 28 日 太田准教授執筆
<http://blog.media.teu.ac.jp/2019/05/post-71b6c1.html>



第 25 回台湾大学芸術祭の展示風景

参考文献

- [1] M. Hayashi, et al., Automatic Generation of a TV Programme from Blog Entries, ACM TVX2018, Poster, 2018.
- [2] M. Hayashi, et al., T2V: New Technology of Converting Text to CG Animation, ITE Transactions on MTA, Vol.2, No.1, pp.74-82, 2014
- [3] 近藤邦雄, 太田高志, 渡辺大地, 三上浩司, ゲーム制作技術とインタラクティブコンテンツデザインの動向, 映像情報メディア学会, 映像情報メディア学会誌, Vol. 68, No. 2, pp.121-124, 2014. 2
- [4] 近藤邦雄, 三上浩司, 柿本正憲, 渡辺大地, 菊池司, 石川知一, 東京工科大学メディア学部デジタルコンテンツ制作技術研究グループ, 画像電子学会学会誌, Vol. 43.No. 3, pp. 420-425 2014. 7
- [5] M. Hayashi, Y. Shishikui, S. Bachelder, M. Nakajima: "An Attempt of Mimicking TV News Program with Full 3DCG - Aiming at the Text-Generated-TV System -", 11th International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB2016), (2016)