

応用論文

Augmented TV: 携帯端末内蔵カメラを用いて TV の映像を画面外へ拡張するシステム

川喜田 裕之^{*1*2}中川 俊夫^{*1}佐藤 誠^{*2}

Augmented TV: an Augmented Reality System for TV Pictures Beyond the TV Screen

Hiroyuki Kawakita^{*1*2}, Toshio Nakagawa^{*1} and Makoto Sato^{*2}

Abstract --- We propose a new TV system which is able to augment representation of TV programs beyond the TV screen. In the system, which we named augmented TV, since animated 3DCG content interlocked with TV programs is overlaid on live video from the mobile device camera in the mobile device screen by augmented reality techniques, the representation of having a TV character coming out of the screen can be provided. We developed an accurate synchronization method and authoring environment of augmented TV content. We implemented augmented TV, and confirmed frame-accurate synchronization (synchronization error time is about 0.03 seconds or less). And we confirmed that the authoring environment is easy to produce augmented TV content.

Keywords: augmented TV, augmented reality, second screen, synchronization method, TVML

1 はじめに

近年、スマートフォンやタブレットなどの携帯端末をテレビのセカンドスクリーンとして放送番組などの映像コンテンツを楽しむ視聴スタイルが増えている[1-3]。セカンドスクリーンを用いることで、テレビ単体ではできなかった新しい演出が可能である。

一方、カメラ付き携帯端末の普及に伴い、カメラ取込み画像に対して 3DCG 等をオーバーレイ表示するタイプの Augmented Reality (以下、AR) 技術[4]も盛んに研究されている。AR 技術を用いることで、バーチャルオブジェクトを使って対象物を空間的に拡張することができる。

テレビの映像コンテンツに対してセカンドスクリーンを使った AR 技術を適用することで、テレビに映っているキャラクターがスクリーンを通して TV 画面外に飛び出してくるといったポピュラーな演出[5,6]ができるのではないかと考えられる。この演出の最大の魅力は、TV のキャラクターが画面から目の前に飛び出してくるといった視聴者が予想できない事態が起きることにある。AR 技術を使うことにより、そのような驚きやリアリティを TV サービスとして提供することが期待出来る。

そこで、放送番組が映っている TV 画面を携帯端末のカメラ越しに見ることで、TV 番組に連動する 3DCG を

用いて画面外に番組映像を拡張する AR システムを“Augmented TV”と呼ぶことにする。このシステムは、視聴者に印象を与える表現手法の提供が目的であるため、リアリティのある演出を可能とする技術的精度が要求される。本研究では、Augmented TV の実現のために高精度な同期方式と新しい表現のためのコンテンツ制作環境を開発した。なお、本論文は、筆者らによるこれまでの研究成果[7-9]を踏まえ、特に本研究の位置づけを明確にし、さらに光学的な考察を行ったものである。

2 サービスモデルと要求条件

図 1 に、Augmented TV のサービスモデルを示す。視聴者は TV に表示された放送の映像コンテンツ(以下、メインコンテンツ)を携帯端末内蔵のカメラ越しに視聴する。携帯端末では、メインコンテンツに対応する通信コンテンツ(以下、サブコンテンツ)が、例えば TV と携帯端末との位置関係といった視聴者の状況に応じてその表

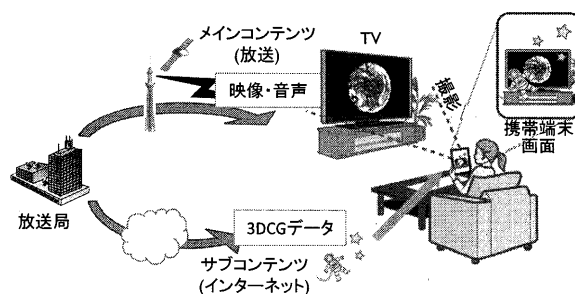


図 1 サービスモデル

Fig.1 Service model

*1 NHK 放送技術研究所

*2 東京工業大学 精密工学研究所

*1 NHK Science & Technology Research Laboratories

*2 Tokyo Institute of Technology Precision and Intelligence Laboratory

示位置等が計算処理されて、オーバーレイ表示される。ここで、サブコンテンツはタイムコードベースのシナリオを持ち、タイムコードを指定する時刻(以下、再生時刻)に基づきメインコンテンツと連動するものを想定している。

Augmented TVを実現するためには、TVと携帯端末とのコンテンツ表示を同期させる仕組みとコンテンツの制作環境が必要になる。各種TVと携帯端末間で汎用的に利用できるように、同期方式の要求条件を以下のように定義した。

- ・ 多様なデバイス間で利用可能であること(ネットワーク非接続端末を含む)
- ・ TV側のメインコンテンツの再生時刻を遅延させないこと
- ・ 携帯端末の画面内において、放送のフレーム単位(誤差0.03秒以下)で高精度に同期すること

コンテンツの制作コストを抑えるために、コンテンツ制作環境の要求条件を以下のように定義した。

- ・ メインとサブの両コンテンツを統合的な環境で制作できること
- ・ コンテンツの素材は汎用的なフォーマットのものが利用できること
- ・ 3DCGキャラクター等は「歩く」や「座る」といった基本動作レベルでモーションを指定できること
- ・ インタラクティブなコンテンツを作れること

3 関連技術

同期方式の関連研究としては、Display-Based Computing[10]という考え方がある。本来人に情報を提示するデバイスであるディスプレイを使って機械に情報を提示することにより、機械を制御するという考え方である。この考え方は、既存インフラの利用を想定しており、新たに連携のための設備を用意するコストがかからない。公共性が求められる放送サービスにおいて、これは大きなメリットとなる。そこで、以下では無線LANなどを用いずに、ディスプレイの情報だけで同期を実現する手法を対象を絞る。

文献[10]はロボットを対象にしたアプリケーションで、今回のようにディスプレイ内のオーバーレイ表示の高精度な同期のための取り組みは行われていない。

前章で定義したAugmented TVのサービスモデルを対象とした技術研究として、山本らは動画の再生時刻を電子透かしにより動画中に埋め込む方法を開発した[11]。この手法を用いれば、TV画面を携帯端末のカメラで捕らえれば動画中に埋め込まれた再生時刻から時刻の同期が可能となる。しかし、電子透かしの制約から一定時間同じ再生時刻を埋め込まなければならず、携

帯端末をTV画面に対して10秒間かざしても0.2秒間程度の同期精度であり、前章で述べた要求条件を満たさない。

筆者らは、データ放送[12]を利用して2次元コードをTVで生成・表示することによりセカンドスクリーンに情報提示する取り組み[13]も行ったが、同様の理由で高精度な同期には利用できない。

仮に電子透かしや2次元コードではなく可視光通信[14]等の技術を用いて各フレームに正確な再生時刻を埋め込んだとしても、実はそれだけでは携帯端末の画面内においてコンテンツ表示の同期が保証されるものではない(次章で詳説する)。

放送分野では、セカンドスクリーンを扱った放送規格が検討されている[15,16]。これらの規格では、主にセカンドスクリーンをTV番組の補足情報提示端末として用いることを想定している。TVと携帯端末との位置関係まで考慮したAugmented TVのように、TVとセカンドスクリーンとが融合することで番組の演出を拡張するような高度な連携手法はまだない。以上から、携帯端末画面内において高精度に同期する新しい方式が必要である。

パワフルな個々の3DCGアニメーション制作環境は、Unity[17]などをはじめに多く存在している。Augmented TVでは、メインとサブの両コンテンツをオーバーレイ表示により合わせて視聴する新しいコンテンツ形態であるため、単に単体の3DCGアニメーションを作るだけでは制作環境として十分ではない。両コンテンツが演出サイドの思い通りになっているか確かめることが可能な統合的なコンテンツ制作環境を用意する必要がある。

4 同期方式

4.1 アプローチと概要

デバイス間の一般的な同期方式としては、各デバイスのクロックを一致させる方式が考えられる。Augmented TVでは、図2に示すように、クロックに合わせて各コンテンツの表示処理を開始し、デコード等の遅延を伴う処理が行われてから画面に表示される。これらの遅延量はデバイスによって異なり、たとえ一致させたクロックに従ってコンテンツを表示しても、携帯端末画面内では表

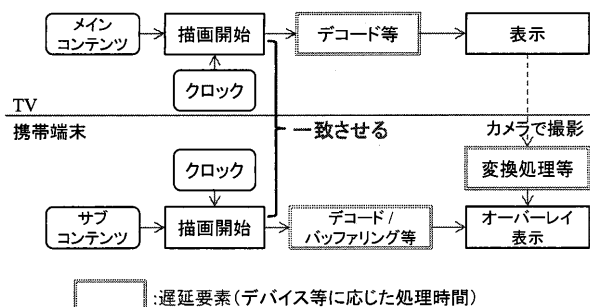


図2 一般的な同期方式

Fig.2 Block diagram of general synchronization method

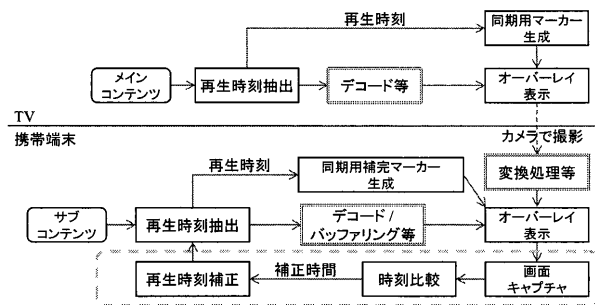


図3 提案する同期方式

Fig.3 Block diagram of proposed synchronization method

示の同期がとれない。

そこで、図3に示すように、各コンテンツの再生時刻を各デバイスの画面上に表示し、携帯端末のカメラ画像内で時刻のズレを比較してサブコンテンツの再生時刻を調整することにより同期する方式を提案する。提案方式では、各デバイスにおいてコンテンツとその再生時刻を遅延要素となる各処理を経ても不可分な形で表示する。再生時刻は、それらの比較を容易にするために動きのある図形(同期用マーカー/同期用補完マーカ、詳細は次節)として表現し、オーバーレイ表示される。ここでは、サブコンテンツおよび図形(同期用補完マーカ)は携帯端末上で同様に演算表示しているため遅延は同じと考えられるので、図形化して表示することによりサブコンテンツとその再生時刻を不可分な形で表示することが可能となる。以上から、デバイスの計算資源等に起因する処理遅延の差異を吸収し、携帯端末上において表示タイミングを一致させることができる。

なお、メインコンテンツの再生時刻を表示する機能は、現行のTV放送においてAugmented TVサービスを行う場合、データ放送を用いる方法が考えられるが、要求する同期精度には達しないことが予想される。この場合は、放送の送出側において本機能を実装し、予め同期用マーカをメインコンテンツの映像の一部とすることでTVの性能差に依存しない同期精度を確保できる。

4.2 マーカのデザイン例と実装方法の詳細

図4に、提案方式によるTV側の同期用マーカの

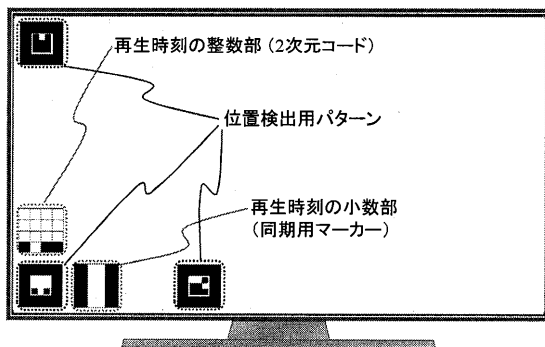


図4 TV画面のレイアウト例

Fig.4 Time marker design example on TV

表示例を示す。画面レイアウトに配慮しつつ携帯端末の処理負荷を抑えながら高精度に同期させるために、TV側の再生時刻は粗密によって表示を分けた(以下、本稿では秒の整数部と小数部で分けた)。粗い再生時刻は2次元コードとして、細かい再生時刻は図5のように連続的かつ周期的に動く同期用マーカとして表示する。

携帯端末は、TV画面と平行に設置し、携帯端末画面の背面のカメラにより位置検出用パターンをもとにそれぞれの位置を検出し、粗い再生時刻を取得して低精度の時刻同期をとる。さらに、画面内の同期用マーカの上に携帯端末で生成した同期用補完マーカ(図5)を位置や向きを合わせてオーバーレイ表示する。このとき、同期用補完マーカは、見かけは同期用マーカと同一平面上に同期用マーカの白い領域に相当するサイズで表示される。同期用補完マーカが白い部分を覆えば同期完了であるが、白い部分が残っている場合は、その幅に応じてサブコンテンツの再生時刻を調整する。この画面上の幅(px)を n 、調整すべき同期誤差(秒)を t とすると、図6に示すように、 n と t の関係式は次式で与えられる。

$$n = \frac{vwt}{2d \tan \frac{\theta}{2}} \quad (1)$$

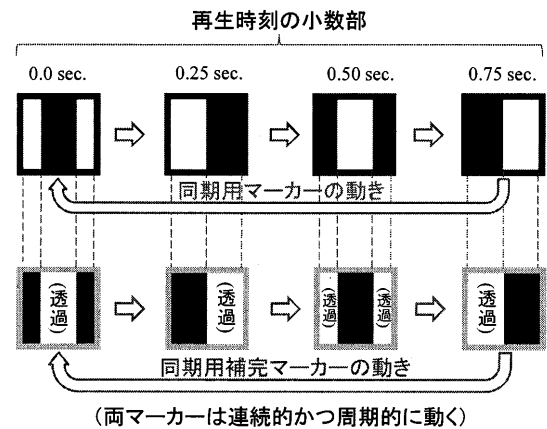


図5 同期用(補完)マーカの動き

Fig.5 Motion of (complementary) time marker

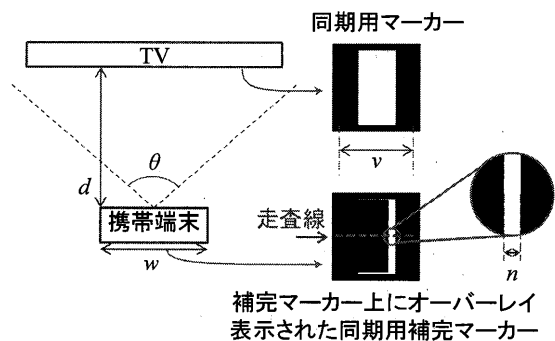


図6 同期精度に関連するパラメータの関係

Fig.6 Parameters related to synchronization accuracy

ここで、 v は同期用マーカーの速さ(m/s) (図 5 のように、白黒の境界がTV画面上を右方向に動く速さ)、 w は携帯端末の画面キャプチャにおけるカメラ画像の横幅(px)、 d は TV と携帯端末との距離(m)、 θ は携帯端末のカメラの水平画角(度)である。なお、 v は、図 5 のように同期用マーカーが再生時刻の小数部を示す場合は、1 秒間でちょうど同期用マーカーの可動領域を動くので、同期用マーカーの可動領域の横幅に相当する。また、 f を焦点距離(m)、 δ を画像座標平面上における 1 画素の大きさ(m/px)とすると、予め f と δ が分かっている場合は次式を用いることもできるが、今回の実験ではパラメータを求める都合から式(1)を利用した。

$$n = \frac{vft}{d\delta} \quad (2)$$

以上の処理を同期完了までループさせることで、コンテンツ間の同期を実現できる。このように、同期用マーカーの上に同期用補完マーカーをオーバーレイ表示させることで同期精度が可視化されて直感的に分かるので、視聴者が同期を確認できる。なお、本方式では、同期精度が位置検出用パターンの位置検出精度に依存するので、高精度に同期するために、図 4 に示した同期用マーカーの両サイドにある位置検出用パターンがともに検出した場合に限り同期判定を行う仕様とする。こうすることで、2 つの位置検出用パターンを 1 つの大きな位置検出用パターンとみなすことができ、その内側に同期用マーカーを配置することにより同期精度に対する位置検出用パターン単体の検出誤差の影響を抑えることができる。同期方式の詳細として、距離に依存しない同期精度を確保するアルゴリズムを以下に示す。

- Step1 同期判定閾値を式(1)から求められる値 n に設定する。(2章の要求条件から t を 0.03 とする)
- Step2 白黒判定閾値を所定の黒領域の画素と白領域の画素との平均値に設定する。(例えば位置検出用パターン内から取得する)
- Step3 再生時刻を 0 とする。この再生時刻は、Step6 での時刻補正を除き、システム時刻と同期して増加する。
- Step4 再生時刻の小数部が 0.5 となったら、図 6 に示す同期用マーカー領域の水平中心線を走査する。
- Step5 走査線上において白黒判定閾値を超える輝度値の画素(白ピクセル)を計数する。
- Step6 白ピクセルの数が同期判定閾値を超えている場合は、式(1)から白ピクセルの数に相当する時間を計算し、その分だけ再生時刻を戻し(時刻補正)、Step4 へ。超えていない場合は、2 次元コードの値を読み取って再生時刻の整数部として同期完了とする。

5 制作環境

5.1 アプローチ

NHK では、これまでに 3DCG ベースの TV 番組を容易にかつ短時間で制作する取り組みとして、番組制作記述言語 TVML(TV program Making Language)を開発してきた[18]。TVML は TV 番組を基本動作レベルで記述するためのスクリプト言語で、TVML プレーヤというソフトウェアに読み込ませることにより汎用的なフォーマットの 3DCG データや音声合成等を使って番組を再生する。また、TVML プレーヤそのものを外部アプリケーションから制御し、ユーザが開発したアプリケーションに取り込んで使えるので、インタラクティブなコンテンツを作ることもできる。

以上から、Augmented TV コンテンツを制作するにあたり、TVML を利用すれば統合的な制作環境という点以外については要求条件を満たすことができる。そこで、TVML を使ったコンテンツ制作環境およびシステムアーキテクチャを検討した。

5.2 制作環境

一般的な AR では実物体を拡張するのに対し、Augmented TV では映像コンテンツを拡張する。映像コンテンツは、もともと 3 次元の空間を 2 次元で表現しているため、本来最適な視聴位置があり、それ以外の位置から視聴すると実際の物の見え方と異なってくる。このことに起因して、初めから任意の位置での視聴を想定している Augmented TV では、サブコンテンツの 3DCG オブジェクトの配置方法は演出次第で幅が出てくる。今回は、TV 画面内のカメラが捕らえた空間をそのまま画面外に延長するように、3DCG オブジェクトの座標系を視聴位置には依存しない形で定義する。

この前提の下、両コンテンツを統合的な環境で作る方法について検討する。TVML では、3 次元のバーチャル空間があり、バーチャルカメラによってその一面を切り出すことで 2 次元の TV 番組を作る。このときに作られる TV 番組をメインコンテンツと捉えれば、バーチャルカメラの投影面より手前側(TV 画面外)の空間をサブコンテンツの空間と捉えることができ、1 つのバーチャル空間を両コンテンツの制作環境とすることができる。

そこで今回、図 7 に示すように、TVML のバーチャル空間内にテレビ画面に相当する境界面(以下、クリッピングプレーン)を導入した。クリッピングプレーンをバーチャルカメラの投影面(図では右側を撮影)と設定すればメインコンテンツが得られ、クリッピングプレーンより手前(図では左)のバーチャル空間をレンダリング領域に設定すれば任意視点でサブコンテンツが検証できる。

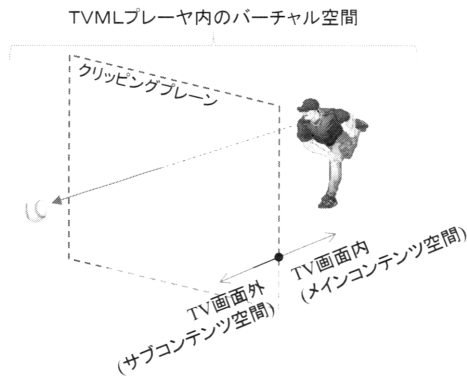


図 7 クリッピングプレーンの導入

Fig.7 Introduction of the clipping plane



図 9 実装したシステムの外観

Fig.9 Appearance of our system

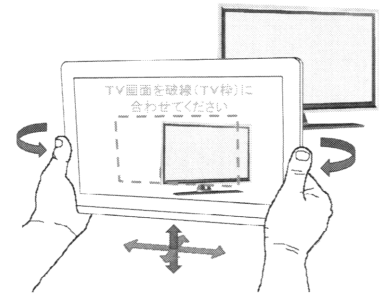


図 11 タブレットの位置合わせ

Fig.11 Position setting of tablet

5.3 携帯端末のシステムアーキテクチャ

前節のクリッピングプレーンを導入した TVML プレーヤは、制作環境として利用できるだけでなく、携帯端末のサブコンテンツ提示環境としてもそのまま活用できる。

図 8 に、携帯端末のシステムアーキテクチャを示す。TVML プレーヤ自体は高精度な時間管理ができないので、両コンテンツの同期精度を確保するために、バックグラウンドで制御アプリケーションが時刻ベースのシナリオに合わせて外部制御を行う構成とした。前章に示した同期方式を図の破線内のように構成することで、高精度に同期することができる。

6 実装評価

提案方式の同期精度等を検証するために、市販の TV とタブレットを用いて実装評価を行った。ただし、同期用マーカーは、現行の TV では生成できないため、メインコンテンツの映像の一部とした。図 9 に、実装した

Augmented TV システムの外観を示す。ここでは、TV から出てきたキャラクターが図のように視聴者と一緒に番組を視聴してくれるコンテンツ例を示した。

表 1-3 に、実装評価に用いたパラメータを示す。図 10 に、TV と携帯端末との距離に応じた同期誤差の測定結果を示す。図 10(a) はタブレットを固定した場合で、図 10(b) はより実際の利用イメージに合わせてタブレットの左右を両手で把持した場合の結果である。(b) の把持は、キャスター付き椅子に座った状態で肘は固定していない。タブレットの位置合わせは、一般の視聴者の利用環境を考慮して、今回の実験では次の手順により行った。測定ごとにタブレットの位置を((b)では椅子も合わせて)いったん別の場所に動かした後に、タブレット画面上にタブレットの空間上の位置を指定するための TV 画面位置を示すガイド線(TV 枠、図 11)を提示し、TV 枠にカメラ取込み画像上の TV 画面を重ね合わせることで手がかかりに位置を合わせた(この TV 枠が指示する厳密な位置については 7.1 節で述べる)。

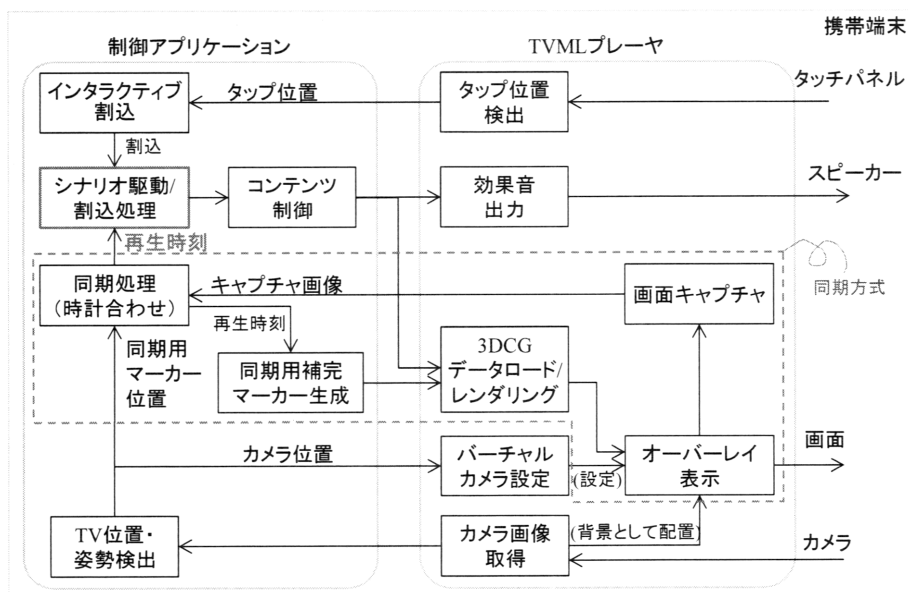


図 8 携帯端末のシステムアーキテクチャ

Fig.8 System architecture in mobile device

表 1 タブレットの仕様
Table 1 Specification of tablet

OS	Windows 8
CPU	Intel Core i3 1.8 GHz
システムメモリ	4 GB
イメージセンサ解像度	1920 x 1080 px
カメラ解像度設定	640 x 480 px
外形	295 x 191 x 11 mm
重量	955g

表 2 TV とマーカーのサイズ
Table 2 Size of TV and time marker

TVの有効画面	886 x 498 mm (40型)
位置検出用パターン・同期用マーカー・2次元コード	200 px (上記TVで92mm) (幅20 pxの余白を含む1辺の長さ)

表 3 同期精度に関わるパラメータの値
Table 3 Parameter values related to synchronization

t	0.03	d	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
v	0.059	n	5.4	3.6	2.7	2.2	1.8	1.5	1.4	1.2
w	1440									
θ	50.4									

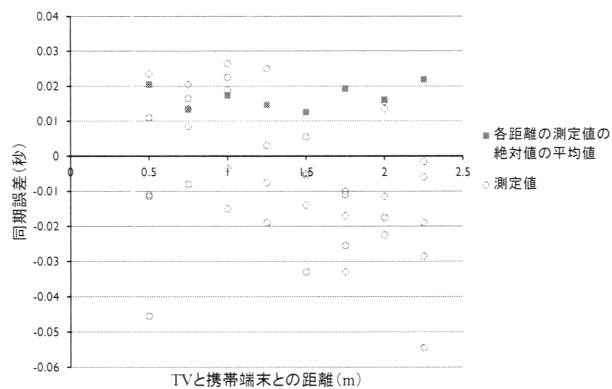
グラフは、携帯端末の画面において、携帯端末の再生時刻から TV の再生時刻を引いた値を測定したもので、各距離について 5 回ずつ測定した値と、それらの値の絶対値の平均値を示している。(a)では、ほとんどのサンプルは 0.03 秒以下の誤差となり、携帯端末の画面内において両コンテンツがフレーム単位で高精度に同期できることを確認した。(b)では、TV と携帯端末との距離が 1.25(m)までは(a)とほぼ同様の結果になったが、距離が大きくなると誤差が大きくなることが分かった。距離が大きくなると手ぶれの影響が大きくなり、同期精度が悪化することがあることを示している。以上から、高精度に同期するためには、距離が大きい場合は何かに固定する必要があることが分かった。

図 12 に、一例として 3DCG キャラクターが画面外に飛び出すシーンにおける携帯端末画面キャプチャを示す。初めメインコンテンツのみに表示されていたキャラクターが頭部からサブコンテンツとして画面外に出てくる様子が分かる。このように、キャラクターが画面をまたぐ際にズレを感じさせない滑らかな出入りを演出できる。

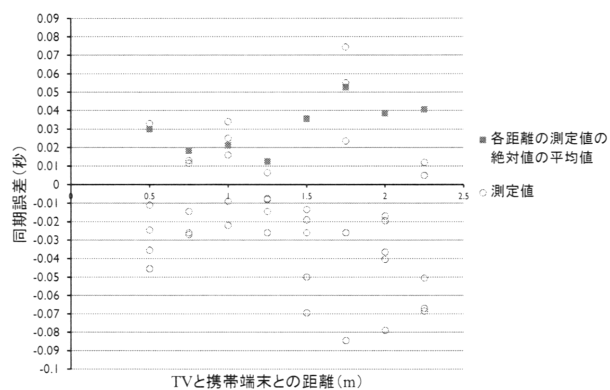
7 考察

7.1 同期方式

提案する同期方式の注意点として、レンズ歪みにつ



(a) タブレットを固定した場合



(b) タブレットを把持した場合

図 10 距離に対する同期誤差

Fig.10 Synchronization error time against distance

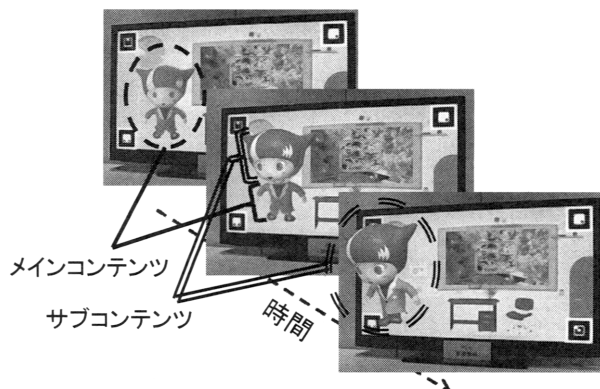


図 12 タブレットの画面キャプチャの変化例

Fig.12 Example of changes of tablet screen captures

いて考察する。本方式は実空間を媒体とする光学的な方式であるため、レンズ歪みによる測定誤差がそのまま同期誤差に直結する。レンズ歪みとは、理想的なピンホールカメラに対して、放射方向に光軸からの距離に応じた非線形的な変位が生じる現象である。本方式では、レンズ歪みが影響する処理は、次の2つである。

- 位置検出用パターンの検出時
- 同期判定時

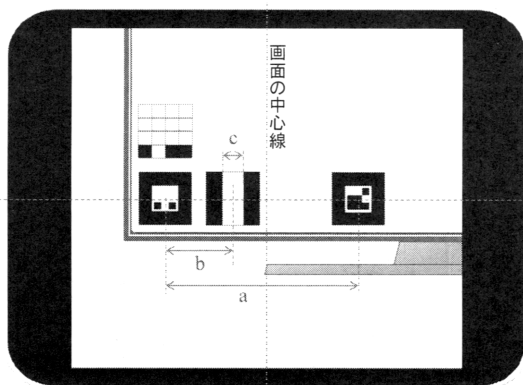


図 13 レンズ歪みを抑えるように配置したタブレット

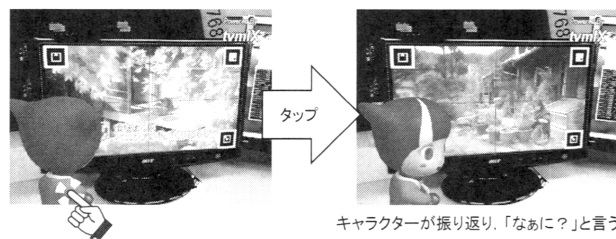
Fig.13 Tablet arranged to reduce the lens distortion

(i)では, 検出した位置検出用パターンから構築するテレビと携帯端末との相対関係を表すための座標系に影響する. この座標系は, 同期用補完マーカの位置と大きさに影響してくる. この影響とは別に, (ii)では, 歪みを持つカメラ取込み画像に対して歪みのない同期用補完マーカをオーバーレイ表示することにより, ここでも同期精度に影響する. 実際には, (i)と(ii)における影響を複合的に考慮する必要がある. 以下では, 今回の実装評価についてレンズ歪みの複合的な影響について考察する.

文献[4]などの一般的な AR システムでは, (i)における影響を抑えるために, ソフトウェア処理によりレンズ歪みの補正を行っている. 図 10 に示した今回の測定では, 歪みの影響を調べるためにあえて補正は行わなかった. 例えば, 距離が 2.25(m)の場合, 同期用マーカを携帯端末画面の左の方で捕らえると, メインコンテンツの再生時刻に比べて, 歪みの影響により 0.1 秒ほど携帯端末の再生時刻が遅れ, 逆に, 右の方で捕らえると 0.1 秒ほど再生時刻が進むことを確認した.

この影響を抑えるためには, レンズ歪みの少ない光軸付近に画像処理対象を捕らえるようにする方法が考えられる. 図 10 の測定では, 図 13 に示すように, 常に携帯端末のカメラの光軸が 2 つの位置検出用パターンの中心を通るように携帯端末を配置した. この時の歪みの影響について考えると, レンズ歪みが非線形的な変位であることに起因して図 13 の a に対する b の歪みと a に対する c の歪みが発生し, これらが同期誤差に直結することが分かる. 今回は, その大きさから同期精度に対してより影響の大きい a に対する b の歪みについて理論値を求める.

表 4 に, 今回の環境におけるレンズ歪み(導出は A.1 に記載)から求めた b と, 同様に求めた a をもとに線形変換により求めた b とのズレ量(px)を示す. また, 表 5 に, 式(1)から求めた 1 ピクセル当たりの同期誤差の理論値を示す. これらの積が今回の実装評価におけるレンズ歪みの影響の理論値である. 表 6 は, この積の値を一



[TVMLスクリプト]

```
sound: open( name=nani, filename="labo¥Sound¥なあに?.wav" )
character: gaze( name=CharacterB, pitch=0, yaw=-120, roll=0, style=witheye, wait=yes )
sound: play( name=nani )
wait( time=1.0 )
character: gaze( name=CharacterB, pitch=0, yaw=0, roll=0, style=witheye, wait=yes )
```

図 14 インタラクティブコンテンツの例

Fig.14 Example of interactive content

覧にしたもので, 補正する場合はこの値だけ携帯端末の再生時刻を進める. この結果から, 距離が長くなるとレンズ歪みの影響は少なくなることが分かる. また, 最も影響の大きい距離 0.5(m)の場合でも, 0.01 秒程度の影響であり, この距離における図 10 の結果を補正しても平均的には 0.002 秒しか改善しない.

以上より, 今回の環境では, マーカが任意の場所にある場合は適切な歪み補正が必要であることと, 携帯端末を適切に配置すれば歪み補正をかけない場合でも同期精度を確保できることが分かった. なお, 今回は広視野な演出を考慮し, 市販のタブレット内蔵カメラにワイドコンバージョンレンズを装着したために比較的歪みの大きい環境であったことを考慮すると, 一般のスマートフォンやタブレットをそのまま利用する環境ではレンズ歪みの影響はより少ないと考えられる.

7.2 インタラクティブコンテンツ

本研究では TVML を実装に用いたので, キャラクターのアクションを基本動作レベルで指定できる. また, キャラクターは, 画面の出入りなど基本的には予め決められたシナリオに沿って動くが, シナリオの中にインタラ

表 4 同期用補完マーカの相対位置のズレ量

Table 4 Distortion values at relative position of complementary time marker

d	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
ズレ量(px)	1.878	0.588	0.253	0.131	0.076	0.048	0.032	0.023

表 5 1 ピクセル当たりの同期誤差

Table 5 Synchronization error time per pixel

d	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
t	0.0055	0.0083	0.0111	0.0138	0.0166	0.0194	0.0221	0.0249

表 6 レンズ歪みによる同期誤差の理論値

Table 6 Synchronization error time by lens distortion

d	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
同期誤差(秒)	0.0102	0.0049	0.0028	0.0018	0.0013	0.0009	0.0007	0.0006

クティブな期間を設けることで柔軟で直感的なコンテンツの構成が可能である。

図 14 に、インタラクティブコンテンツの実装例を示す。この例では、視聴者がタブレット上にサブコンテンツとして表示されたキャラクターをタップすると、キャラクターが振り向いて「なあに？」と聞いてくる。この一連の動作は、図 14 のように「振り返る (gaze)」などのわずか 5 つのコマンドで構成されている。このように、Augmented TV では、一方的な放送サービスに留まらないインタラクティブなコンテンツを提供することができる。

7.3 マーカーのデザイン

4.2 節で示した同期用マーカーや位置検出用パターンのデザイン例の妥当性を示すために、運用上の工夫とデモ展示での来場者の反応について述べる。画面レイアウトや表示の有無などを含むデザインは、メインコンテンツをできるだけ障害しないように、コンテンツの演出要件と技術要件のトレードオフから決定される。また、実際の放送では、携帯端末を利用しない視聴者も想定して、メインコンテンツのみでも完結する放送サービスであることが望ましい。メインコンテンツを障害しない運用方法とデザインについて検討し、意見収集のためにデモ展示を行った。

まず、同期用マーカーについては、同期用マーカーの表示期間を抑える運用上の工夫が考えられる。一般の視聴者が携帯端末のカメラを通して視聴する時の体勢を考えると、Augmented TV のコンテンツは高々数分間と短いことが想定される。また、Mpeg-2 TS[19]を利用した現行の放送には同期信号が多重化されているので、メインコンテンツのジッターはほとんど発生しない。そこで、コンテンツ冒頭部に同期用マーカーを表示する同期期間を設け、その間にのみ同期を行う運用とすることで、メインコンテンツの大部分には同期用マーカーを表示しなくても問題となるようなズレは生じない。この場合は、コンテンツの途中からではサブコンテンツとともに視聴できないが、短いコンテンツであればその影響は少ない。

一方で、位置検出用パターンは、同期期間では同期用マーカーの位置を示すとともに、同期期間以外では TV 画面の四隅に配置することで画面の位置・姿勢を示す役割を果たすので、今回の Augmented TV サービスにおいて常に表示される。そこで、メインコンテンツを障害しない程度のサイズに収める必要がある。今回は、一般的なリビングやカメラ設定を想定して、40 型の TV に対して 1~2.5m 程度の距離で位置検出用パターンを十分に認識できるように、余白部分を含む各パターンを 200(px)四方に決定した。アスペクト比が 16:9 のデジタル放送においてアスペクト比 4:3 の映像を放送する時に、サイドパネル方式[20]にて画面の両側に演出上の意味



図 15 技研公開 2013 の様子

Fig.15 Open house 2013

を持たない静止画(サイドパネル、それぞれ横 240(px)×縦 1080(px))をはめ込むことが多いが、今回のパターンのサイズは、サイドパネル領域に収まることから、メインコンテンツとしての演出領域を十分に確保できる。

携帯端末を利用しない視聴者に配慮した運用について述べる。同期用マーカーは、4.1 節で触れた通り、現行ではメインコンテンツの映像の一部とする運用が想定されるが、位置検出用パターンは、静止画であることから、データ放送を用いることで視聴者のリモコン操作によりその表示の有無を制御することができる。なお、データ放送を用いれば位置検出用パターンのサイズを視聴者のリモコン操作により変更できるように設計することも可能である。

一般の視聴者や番組制作者から主観的な評価に基づく意見を収集するために、技術展示会である技研公開 2013[21]にて 4 日間にわたりデモ展示を行った。技研公開への来場者数は約 2 万人であった。図 15 に展示の様子を示す。この展示では、冒頭 10 秒程度の同期期間を設け、その後は同期用マーカーを表示しない約 4 分間の Augmented TV コンテンツとし、同期期間を含めてループ再生させた。携帯端末では、各ループにおいて非同期の状態スタートし、同期期間中での同期完了をもって初めてサブコンテンツが表示される仕様とした。

ほとんどの来場者は、まず画面デザインよりも TV 画面からボールやキャラクターが飛び出してくる演出に驚き、その非日常感を純粋に楽しんでた。デザインに関しては、特に番組制作者などから、同期期間以外に画面四隅に配置した位置検出用パターンが気になる、という意見があったが、データ放送で表示を制御できることやパターンのサイズは柔軟に変更できることを伝えると納得いただけた。また、同期精度の不足を指摘するような意見はなく、今回検討したデザインは演出面や精度面において十分に実用的であることが確認できた。

8 おわりに

本研究では, AR 技術を用いて TV をカメラ越しに視聴することにより新たな映像表現を可能にする Augmented TV を提案した. Augmented TV を実現するための高精度な同期方式とコンテンツ制作環境を開発し, 実装評価によりその有効性を検証した. 高精度な同期を実現したことにより, キャラクターが TV 画面の内外を自然に出入りするといった演出が可能になった.

Augmented TV は, 視聴者に驚きやリアリティを与えるとともに, インタラクティブなサービスを提供できる新しいメディアである. 今後は, リビングやパブリックビューイングなどの場面において, 空間を活かしたアンビエントメディアとして豊かな表現が活かされていくことだろう.

参考文献

- [1] ニールセン: ニールセン、生活者のマルチスクリーン利用動向を分析;
http://www.netratings.co.jp/news_release/2013/11/Newsrelease20131112.html
- [2] ディズニー: ディズニーセカンドスクリーン;
<http://disney-studio.jp/secondscreen/>
- [3] 日本テレビ: どこでも志村どうぶつ園;
<http://www.ntv.co.jp/zoo/app/ar.html>
- [4] 加藤博一: 拡張現実感システム構築ツール ARToolKit の開発; 信学技報, 101(652), 79-86 (2002.2)
- [5] 水木しげる: テレビくん(漫画); 講談社 (1965.8)
- [6] 鈴木光司原作, 中田秀夫監督: リング(映画); 東宝 (1998.1)
- [7] 川喜田裕之, 西本友成, 遠藤洋介, 藤井亜里砂: 放送通信連携サービスにおけるデバイス間同期方式; 2012 年映像情報メディア学会年次大会, 16-6 (2012.8)
- [8] 川喜田裕之, 金子浩之, 井上誠喜, 藤井亜里砂, 中川俊夫: 番組連動型 3DCG コンテンツの柔軟な制作環境; 2013 年映像情報メディア学会年次大会, 1-5 (2013.8)
- [9] H.Kawakita, T.Nakagawa: Augmented TV: An Augmented Reality System for TV Programs Beyond the TV Screen; International Conference on Multimedia Computing and Systems (ICMCS) (2014.4 掲載予定)
- [10] 杉本麻樹, 小島稔, 中村享大, 新居英明, 稲見昌彦: 画像提示装置で表示した指標画像を用いた位置・姿勢計測; 日本バーチャルリアリティ学会論文誌, Vol.10, No.4, pp.485-494 (2005.12)
- [11] 山本奏, 安藤慎吾, 筒口拳, 片山淳, 谷口行信: モバイル動画透かし技術で実現する映像同期型 AR: Visual SyncAR; 第 41 回画像電子学会年次大会, R6-2 (2013.6)
- [12] デジタル放送におけるデータ放送符号化方式と伝送方式; ARIB STD-B24, Version5.8 (2013.7)
- [13] H. Kawakita, Y. Nishimoto, T. Inoue: Fast Generation Method of 2D Code on DTV Receivers; IEEE International Conference on Consumer Electronics 2011, T08-S07/3, pp.0794-795 (2011.1)
- [14] Visible Light Communications Consortium;
<http://www.vlcc.net>
- [15] H. Ohmata, M. Takechi, S. Mitsuya, K. Otsuki, A. Baba,

K. Matsumura, K. Majima, and S. Sunasaki: Hybridcast: A New Media Experience by Integration of Broadcasting and Broadband; Proceedings of the ITU Kaleidoscope Academic Conference, S5.1 (2013.4)

- [16] Hybrid Broadcast Broadband TV; ETSI TS 102 796 V1.1.1 (2010.6)
- [17] Unity: <http://unity3d.com/>
- [18] NHK TVML Player:
<http://www.nhk.or.jp/str/tvml/japanese/player2/>
- [19] Mpeg-2 Systems: ITU-T Recommendation H.222.0 (or ISO/IEC 13818-1)
- [20] 地上デジタルテレビジョン放送運用規定; ARIB TR-B14, Version5.5 (2014.3)
- [21] 技研公開 2013;
<http://www.nhk.or.jp/str/open2013/index2.html>
- [22] Brown D. C.: Close-range Camera Calibration; Photogrammetric Engineering, Vol.37, No.8, pp.855-866 (1971)
- [23] OpenCV;
<http://opencv.jp/>

A 付録

A.1 レンズ歪みの大きさの導出

キャリブレーションパターンを撮影することにより, レンズ歪みの大きさを求めた. レンズ歪みのモデルは次のように低次多項式で定式化される[22].

$$\begin{aligned}x &= \frac{1}{f}(x_i - c_x) \\y &= \frac{1}{f}(y_i - c_y) \\x_d &= x(1 + \kappa_1 r^2 + \kappa_2 r^4) + c_x \\y_d &= y(1 + \kappa_1 r^2 + \kappa_2 r^4) + c_y \\r^2 &= x^2 + y^2\end{aligned}\quad (3)$$

ここで, (x_i, y_i) は画面における歪みのない座標値, (x_d, y_d) は画面における歪みの発生した座標値, f は焦点距離, (c_x, c_y) は歪み中心の座標値を示しており, $f, (c_x, c_y)$ は, カメラの内部パラメータと呼ばれている. κ_1 と κ_2 は歪曲収差パラメータと呼ばれる歪み度合いを示している. $f, (c_x, c_y), \kappa_1, \kappa_2$ の値は, 今回の実験では OpenCV[23] の `cv::calibrateCamera` 関数によって求め, それぞれ 692(px), (321(px), 256(px)), -0.160, 0.162 であった(樽型歪み).

(2014 年 3 月 10 日受付)

[著者紹介]

川喜田 裕之 (正会員)



2004 年東京工業大学大学院理工学研究科修了. 同年, NHK 入局. NHK 広島放送局にて放送波の受信技術業務に従事し, 2007 年放送技術研究所に配属. 現在, 東京工業大学社会人博士課程在籍. 放送通信連携における

端末連携の研究に従事。

中川 俊夫（非会員）



1989 年東京大学大学院工学系研究科電気工学専攻修士課程修了。同年, NHK 入局, 放送技術研究所に配属。編成局を経て 2012 年より放送技術研究所主任研究員として放送通信連携サービスの研究に従事。

佐藤 誠（正会員）



1973 年 3 月東京工業大学工学部電子物理工学科卒業。1978 年 3 月, 同大学大学院博士課程修了。同年 4 月より同大情報工学科助手。1986 年 3 月東京工業大学精密工学研究所助教授を経て, 現在, 同大学精密工学研究所所長兼教授。博士 (工学)。コンピュータビジョン, パターン認識, ヒューマンインタフェース, VR の研究に従事。ストリング型ディスプレイ SPIDAR を提案, 開発。元日本 VR 学会会長。