

歩行可能な簡易 VR システムを用いた 街中情報配信システムの設計と実装

木村 眞 吾^{†1} 新井 イスマイル^{†2} 寺田 直 美^{†1}
河合 栄 治^{†3} 西尾 信 彦^{†4} 砂原 秀 樹^{†5}

バーチャルリアリティ (Virtual Reality : VR) の研究が進歩し, シミュレータやエンターテイメントなど幅広い分野で高い成果が得られている. これらを実現するシステムの一つとして没入型装置が用いられている. しかしながら, 既存の没入型装置はコスト, 設置場所などの面で大きな制約を受けるため, 簡易に導入することが困難であった. 本研究では, 一般に販売されている機器であるプラズマディスプレイ, バランス Wii ボード, PC を用いて簡易的な没入型装置を構築した. そのシステム上に, Google StreetView を用いて現実の街並みを投影し, 歩行による操作を実現した. 同時に, Web 上にある地理位置情報に基づくデータを重ね合わせて表示することにより, 街を歩行しながら情報を得られるようにした. 本システムを用いて「大阪・北ヤードナレッジキャピタルトライアル 2009」イベントにてデモンストレーションを行った.

Design and Implementation of Real-World-oriented Contents Distributing System with Simplified Immersion System Controlled by Walking Interface

SHINGO KIMURA,^{†1} ISMAIL ARAI,^{†2}
NAOMI TERADA,^{†1} ELJI KAWAI,^{†3}
NOBUHIKO NISHIO^{†4} and HIDEKI SUNAHARA^{†5}

Virtual Reality(VR) Technology has made remarkable progress and its applied technology are used in a wide sphere, especially in the field of developing simulators or entertainment equipment. One typical VR apparatus is a large-scale immersion system. However, those systems are not to easy to install because those are mostly expensive and space-consuming. In this paper, we implement simple immersion system using consumer products: plasma display panels(PDPs), Wii Balance Board and personal computers. Our implementation includes walking interface, panoramic view of Google Street View and mapping

additional information program. User obtains diverse information just walk in our VR systems, using contents mapping program which plots data over Street View images based on its geographical location information. We also show the detail of our system demonstration at "Knowledge Capital Trial 2009 at Osaka" event in March 2009.

1. はじめに

災害, 航空, 建築, 医療, 化学, 自動車, エンターテイメントなどの分野でバーチャルリアリティ (Virtual Reality : VR) システムの研究が行われている. VR を実現するシステムのひとつとして, 没入型装置がある. これは視界あるいは身体を完全に覆う形で映像を投影するもので, 高い臨場感を得ることが可能である. 既存の没入型装置は画像処理や CG をはじめとする専門知識や没入型ディスプレイなど巨大な装置, さらにそれを設置するためのスペース, 操作するための専用アプリケーションやデバイス, 表示用の特殊な画像といった要素が必要となる. そのため, 多くのコストがかかり, 一般に普及しづらいという問題点がある. 本研究では, 一般に販売されている機器を用いて簡易的な没入型装置を構築した. この装置は安価に設営可能であるという利点から, イベント会場など一時的な場所での利用, また, 歩行訓練など医療・福祉方面での利用, 教材など教育方面での利用, 安心安全情報など地域情報の共有などに応用も可能である. 本システムでは, 現実の街並みを投影し, 歩行による操作を実現可能なアプリケーションを実装した. 付加的な情報を提示させるために, Web 上の地理位置に関連付けされた大量の情報を収集して表示させた. これにより, 検索という手段を用いなくても, 街並みを歩く動作だけでそのエリアに対する情報を受動的に得ることを目指した. 本論文では 2 章においてシステム概要について, 3 章では簡易型没入歩行システムの実装について述べる. 4 章ではイベントで行ったデモについて述べ, 5 章で本論文をまとめ, 6 章では今後の課題について述べる.

†1 奈良先端科学技術大学院大学 情報科学研究科

†2 立命館大学 総合理工学研究機構

†3 情報通信研究機構

†4 立命館大学 情報理工学部

†5 慶応義塾大学大学院 メディアデザイン研究科

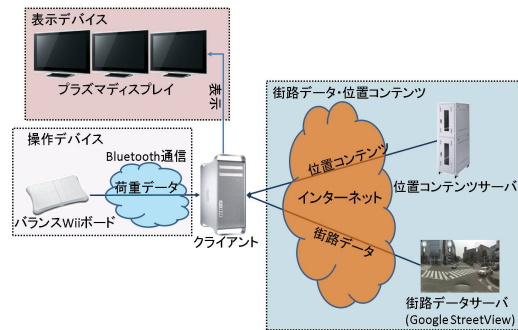


図 1 システム全体構成



図 2 実際のシステム

2. システム概要

2.1 システム全体構成

システム全体構成を図 1 に示す．システムは表示デバイス，操作デバイス，位置コンテンツサーバ，街路データサーバの 4 つで構成される．

表示デバイスは大型のプラズマディスプレイを用いる．動きの自由度を確保しつつ没入させる必要がある．本システムでは 180 度の視野を覆うために，50 インチディスプレイを 3 枚用いて 120 度ずつ開いて配置した．ユーザはその中央位置で操作を行う．

操作デバイスに関しては，歩行による動作で直感的に操作可能で入手が容易かつ安価なものとしてバランス Wii ボードを用いた．通信方法に関しては Bluetooth を用い，クライアントには Mac OS を使用した．

ディスプレイに表示させる街並み画像は，全方位カメラでの撮影画像に比較的近く，既に多くの街路データを使用可能な Google StreetView を用いた．

位置コンテンツに関しては新井，西尾らによって開発中の実世界指向 CMS¹⁾ を利用した．データベースには Web 上から取得した地理位置情報と Google StreetView に表示可能な形式を持ったコンテンツ (例：ぐるなびなど) を登録してある．

本システムはユーザがバランス Wii ボード上で行った動作を解析することで歩行や片側に荷重を置いているだけなのかを判断し，その動作に基づき表示している街路データや位置コンテンツを随時変更していくシステムである．その結果，実世界に近い感覚で街中を散歩している感覚が得られる．本システムは，まず，システムの利用者であるユーザがバラ

ンス Wii ボード上で行った動作を荷重データとして Bluetooth 通信によりクライアントに送信する．そのデータに対して，クライアント上で動作解析を行う．次に，その動作が片側に荷重を置いているだけなのか歩行なのかを判断し，片側に荷重を置いているだけならば，方向転換を行い，位置コンテンツがもっている地理位置情報に基づき，現在表示している街路データ上に表示可能な情報を随時表示させる．歩行ならば歩行速度に合わせて街路データ (Google StreetView) と位置コンテンツがもっている地理位置情報に基づき，現在表示している街路データ上に表示可能な情報を随時表示させる．実際のシステムは図 2 に示す．

2.2 アプリケーション構成

2.2.1 表示系アプリケーション構成

表示ソフトウェアとしてプラットフォーム非依存，情報のマッシュアップが可能，Google StreetView が利用可能であるといった点から Web ブラウザを使用した．表示系である Web ブラウザは，バランス Wii ボードの動作解析の結果にアクセスする必要があるが，Web ブラウザからローカルホスト上にあるファイルを直接参照することはセキュリティの観点から困難である．そのため，ローカルホスト上で Web サーバを立ち上げ，その Web サーバ上にバランス Wii ボードの動作解析結果を逐次アップロードし，Web ブラウザからその結果を定期的に確認することで解決した．街路データを Web ブラウザに表示する方法として Google StreetView をコントロール可能な API である Google Maps API を用いた．表示系アプリケーション構成は図 3 に示す．

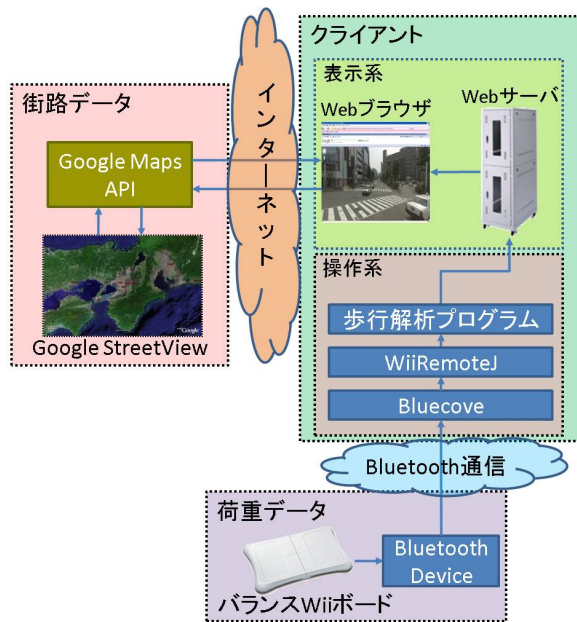


図3 アプリケーション構成

2.2.2 操作系アプリケーション構成

バランス Wii ボードとクライアント間の通信には Java API である WiiremoteJ²⁾ を使用するため、ドライバとして Java API の BlueCove³⁾ を使用した。操作系アプリケーションについては図 3 に示す。WiiremoteJ はバランス Wii ボードデバイスの検出、プロトコル変換、信号情報の受信などが行える。この機能を利用し、バランス Wii ボードから歩行データを受け取り、歩行解析プログラムを作成する。歩行解析プログラムは、WiiremoteJ から受け取った一定時間の信号情報をもとにバランス Wii ボード上で行った動作を解析し、歩行動作している、止まっている、片側へ荷重を置いているといった結果を返す。

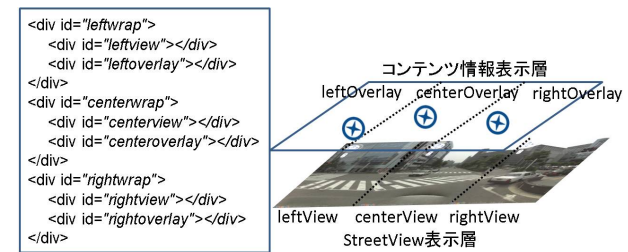


図4 画面分割方法

3. 簡易型没入歩行システムの実装

3.1 3面表示システム

3.1.1 表示システム

本システムでは街路データとして東京や大阪などの主要都市、さらに、海外の都市まで画像が提供されている Google StreetView を利用した。3面表示には図 4 に示すようにフル HD 横 3 面分の広さのブラウザに、フル HD1 枚分の StreetView 映像を再生する Google Maps オブジェクトを 3 つの映像が繋がるように調整配置した。また、StreetView では表示部をクリックすることにより前進や方向転換などの操作が可能となっているが、3 つの StreetView 映像の表示角度や表示位置のバランスが崩れてしまうという問題点がある。そのため、StreetView 映像とコンテンツを図 4 で示すように別々の表示層とし、重ね合わせて表示させることにより、誤って StreetView 映像をクリックした場合であっても、クリック不可能になるようにした。コンテンツについてはクリックにより詳細情報を表示する。この際のクリックの問題については前述した。また、モニタサイズに対応するため、3 枚の StreetView 映像の境目がディスプレイの境目になるように画面サイズ調節が必要となる。画面サイズの調整には Web ブラウザの幅の情報を Javascript 用いて取得し、取得した幅の 1/3 を基準に微調整が可能なパラメータを設けた。さらに、単純に 1 地点から左右それぞれ 120 度ずらした映像を表示させた場合、正面部と左右に映し出される StreetView 映像のスケールが若干異なってしまい、連続的に見えないことが判明した。具体的には、正面映像がやや小さく表示された。その結果、StreetView 映像の視野角は 90 度程度であることが予想されたため、図 5 に示すように視点調整を行った。本来は正面映像を若干大きく表示させる必要があったが、仕様上倍率の微調整が困難であったため正面・左・右映像をズーム表示

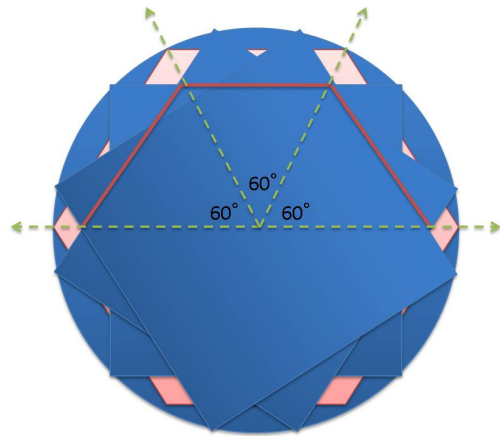
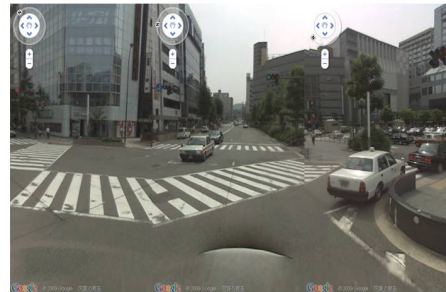


図 5 視野角の調整



図 6 (a) 調整前



(b) 調整後

させることで、誤差の範囲を小さくし、連続的に表示されるように微調整した。調整後と調整前実際の映像を図 6 に示す。

3.1.2 道路方向と表示連動

本システムでは前進と方向転換の 2 種類の操作の実装を行った。前進については歩行の場合に実行され、次の進行方向の映像に移動するというものである。直進道路や直行する交差点などでは問題なく 3 つの画面が連動動作する。方向転換については交差点で左右に回りたい場合や周囲を見回したい場合に使用する。この動作についてはどのような道路でも連

動動作が可能である。

前進動作の 3 画面連動動作に関しては、道路がカーブしている場合と直行していない交差点 (例：三叉路など) の場合に問題が発生する。以下にその問題点と解決法を示す。

道路がカーブしている場合 図 7(a) のように、道路がカーブしている場合は、前進時に、視線方向が道路のカーブに追従せず、異なる向きを表示するという問題点が生じる。解決法として、正面映像の進行方向を取得し、映像が切り替わった際に、取得した進行方向とは異なる向きで表示されていた場合に、映像が進行方向と同じ向きになるように調整する。そうすることで、常に正面映像は道路に沿った状態の映像を表示させることが可能となる。また、左右映像については正面映像に対して 120 度の角度でディスプレイが配置されているため、方向をずらして表示しなければならない。よって正面映像の進行方向に対し、ずらす分だけの修正値を加えることで左右映像を得ることが可能である。以上の方法からカーブしている道路に関してはスムーズに進むことが可能である。

直行していない交差点 図 7(b) のように、直行していない交差点の場合、三叉路を例にとると、左右どちらの道路を次の進行方向とするのかを選択する必要がある。また、交差点とは違い、三叉路では次の進行方向の方角が非常に近い。直交している交差点の場合は、右折が東の場合、直進が左、左折が西と方角に最低でも 90 度以上の差がある。これに対し、三叉路では右側の道路が北で、左側の道路が北北西などの場合も考えられる。この場合、方角の差は 22.5 度しかない。そのため、正面映像が右側の道路を進行方向として選択し、左映像が左側の道路を進行方向と選択する可能性がある。これらの問題点の解決法として、正面映像が向いている方角を取得し、さらに、正面映像が進むことが可能な方向をすべて取得する。これらのデータを元に、正面映像が向いている方角に最も近い方向を選択する。また、左右映像については正面映像が向いている方向に進み、道路がカーブしている場合と同様にずれた分だけの修正値を加える。以上の方法から直行していない交差点をスムーズに進むことが可能である。

3.2 歩行解析プログラム

荷重データの解析結果はローカルホストに置かれているので Web ブラウザによるアクセスが不可能である。そのため、クライアント PC 上に Web サーバを立ち上げ、その Web サーバ上にデータをアップロードしていく。アップロードするデータは扱いやすいように JSON 形式とした。アップロードされたデータを、体格や性別、年齢の違う 4 人での歩行解析実験を繰り返した結果、Web ブラウザからのバランス Wii ボードデータへのアクセスは 1 秒に 1 回というタイミングで問題なく動作した。

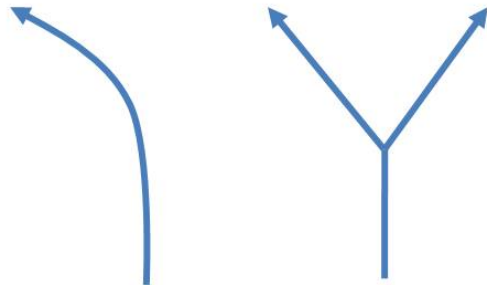


図 7 (a) 道路がカーブしている場合 (b) 三叉路の場合

バランス Wii ボードからのデータを受け取るためには、Bluetooth からの信号を受け取り、それを解析する機能が必要になる。本システムでは、JSR082 と呼ばれる Java で Bluetooth にアクセスするインタフェースを提供している API を実装している Bluecove を利用した。また、バランス Wii ボードとの通信部分にはバランス Wii ボードにアクセス可能で、開発が容易な API である WiiremoteJ を使用した。Bluecove は WiiremoteJ で使用されており、コーディングする際には意識する必要はない。WiiremoteJ は Bluecove から受け取った信号の解析やバランス Wii ボードの検出などのイベントを受け取ることが可能な API である。

本システムでは、荷重データを解析するプログラムを参考にしたサイト⁴⁾ から利用した。以下にそのプログラムの概要について述べる。

バランス Wii ボードには 4 つの荷重センサが存在している。これらのセンサから WiiremoteJ を利用し、データ受信を行う。受信したデータから以下に示す任意の位置の荷重データを取得するには getMass() メソッドの引数に位置を代入することで可能となる。センサの設置された右上、右下、左上、左下の 4 領域の荷重データを得るには TOP, BOTTOM, LEFT, RIGHT が示す位置の荷重データを以下のように getMass() メソッドを使用することで取得することができる。

```

左上：getMass(TOP,LEFT);
右上：getMass(TOP,RIGHT);
左下：getMass(BOTTOM,LEFT);
右下：getMass(BOTTOM,RIGHT);

```

このプログラムでは前への荷重や後ろへの荷重が取得することが困難なため、上下左右の荷重データを取得できるように改良した。上の荷重データを取得するには、RIGHT と LEFT

の中点の位置と TOP を引数として与える必要がある。中点の位置は (RIGHT+LEFT)/2 を計算すること取得可能である。これらを getMass() メソッドの引数とすることで上の荷重データを得ることができる。同様の方法で、右、左、下も得ることができる。以下にプログラムを示す。

```

上：getMass(TOP,(RIGHT+LEFT)/2);
右：getMass((TOP+BOTTOM)/2,RIGHT);
左：getMass((TOP+BOTTOM)/2,LEFT);
下：getMass(BOTTOM,(RIGHT+LEFT)/2);

```

これを基に、上、右上、右中、右下、中下、左下、左中、左上、中で示される 9 つの位置を推定するアルゴリズムを利用し、バランス Wii ボードのどの面を踏んでいるかを推定する。

- (1) 4 方向の力 (f_1, f_2, f_3, f_4) の割合 r_n を求める ($r_n = f_n \sum_{i=1}^4 f_i$)
- (2) r_n の最小値 $m_n = \min(r_n)$ を求める
- (3) m_n 以外の r_i が、全て閾値 th を超えていた場合、 r_n の方向の真逆を踏んでいたこととする m_n が左の場合、上、右、下方向にかかる力が閾値を超えていた場合は右側を踏んでいたことと見なす
- (4) 閾値を超えていなかった場合は、真ん中を踏んでいたことと見なす

以上が参考にしたプログラムの概要である。

本システムでは、歩く動作とどちら側に体重が乗っているかという動作しか行わないため、右側と左側の 2 分割に変換した。歩く動作の解析には、体格や性別、年齢の違う 4 人で歩行を行い、実験を繰り返した。その結果、動作するごとにアップロードされる荷重データを 100 個並んだ配列に順次格納し、右側と左側へ荷重がかかっているデータが 1 ずつ以上あれば、歩行と認識する。右側へ荷重がかかっているデータが 30 個以上かつ左側へ荷重がかかっているデータ 0 個の場合が右側への方向転換と認識する。左側へ荷重がかかっているデータが 30 個以上かつ右側へ荷重がかかっているデータ 0 個の場合が左側への方向転換と認識する。それ以外に関しては止まっていると認識する。以上の解析結果をクライアント PC で立ち上げてある Web サーバ上にアップロードし、そのデータに対し、Web ブラウザが 1 秒毎にアクセスする。

3.3 コンテンツ表示システム

3.3.1 システム概要

飲食店情報やイベント情報、ある地点で撮影された写真、中継中の動画など Web 上には地理位置情報に基づいたさまざまなコンテンツが存在する。これら位置に関連した情報を直

感的に取得するために、Google StreetView 上にマッピングされるシステムが実装されている¹⁾。本研究では、このコンテンツ表示システムと、3.1 で述べた 3 面表示システムと連動させることで、没入して歩行しながら情報を得られるシステムを実現した。

3.3.2 システム構成

このシステムは GPS などを用いて現在位置を取得し、その場所における StreetView 映像を獲得する。ユーザの位置をその都度サーバシステムにフィードバックすることでコンテンツを選び出し、サーバシステム上でマッシュアップを行い、ユーザに適した情報を提供する。Web コンテンツだけでなく、地理位置情報に関係するものであれば様々な情報を Google StreetView 上にオーバーレイ表示させることが可能である。例えば、飲食店のレビュー記事、その場所で撮影された写真や動画、その場所に関係のある動画(動画広告)、その地点にいる人とのリアルタイム中継などがある。

3.3.3 表示方法

今回用いたシステムではサーバ上もしくはインターネット上にデータの実態が存在するコンテンツを収集し、地理位置情報と飲食店などレビューといった内容をデータベースに登録する。そして、現在表示している映像の位置に基づいて、近隣の情報をアイコンで StreetView 映像にマッピング、オーバーレイ表示させるものである。アイコンをクリックすることにより、詳細情報を表示させる。このとき、図 8 で示すように、遠くの情報はアイコンを小さくすることで遠近感を出して表示させる。情報収集部分に関してはモジュール化しており、さまざまなコンテンツを容易に取り入れ可能な設計となっている。このシステムを利用することで Google StreetView 上でもさまざまな情報を表示させることが可能となった。

4. デモ

本システムは 2009 年 3 月 12, 13 日に実地される「大阪・北ヤードナレッジキャピタルトライアル 2009」にて「ジツセカイ三面鏡お散歩」としてデモを行った。会場では、50 インチのプラズマディスプレイと操作系にバランス Wii ボード及びジャイロセンサ内蔵の Bluetooth マウスを用いて来場者が実際に操作するデモンストレーションを行った。システムは図 9 で示す構成になっており、Google StreetView 映像は会場付近の大阪市内エリアを中心に京都市内、神戸市内など複数箇所を用い、来場者がシステム上でステップを踏みバーチャル上の散歩をするというものである。同時に、会場に別途設置した臨場感中継システムと連動させ、Google StreetView 上のカメラアイコンをクリックすることにより、上部に別途配置した中継用モニタがアクティブになり、その地点に置いたカメラと中継が始まるよう

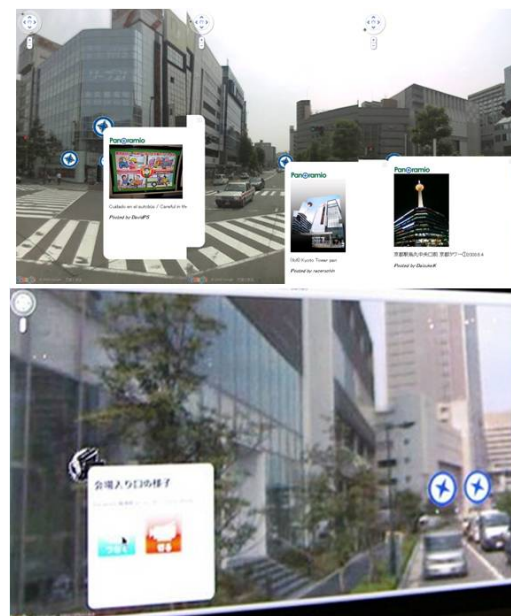


図 8 コンテンツ表示システム

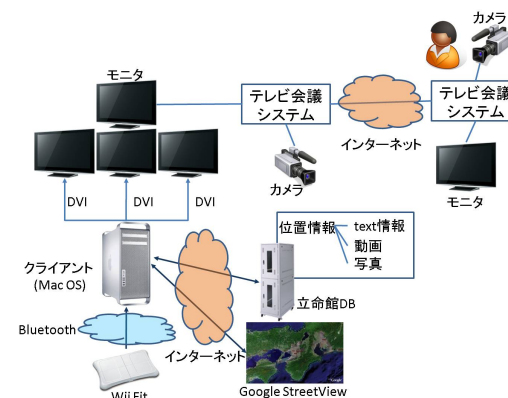


図 9 デモで行ったシステム図

に設計している。なお、モニタは表示用 PC からシリアル通信によって制御し、アイコンクリックを検出してアクティブ化する。コンテンツはぐるなび、Panoramio、用意していた動画 (地理位置情報を付与したもの) といった文字情報コンテンツに加え、動画コンテンツも登録しておき、その位置に関連した動画が再生されるようにした。

5. ま と め

本研究では、簡易的な没入を実現するために、装置として 3 枚のプラズマディスプレイと PC、バランス Wii ボードを用い、Google StreetView 上をバランス Wii ボードで歩いて操作し、位置に応じてコンテンツがオーバーレイ表示されるシステムを構築した。表示系では、Web ブラウザを利用し、Google StreetView を表示させた。3 つの StreetView 映像を表示し、さらに、移動があった場合も 3 つの映像が同期し、連動表示させるための機構を実装した。操作系では、クライアントとバランス Wii ボード間で Bluetooth 通信を行い、受信した荷重データをもとに、歩行解析を行う機構を実装した。コンテンツ系では、Web 上の地理位置情報を持ったコンテンツだけでなく、飲食店のレビュー記事、その場所で撮影された写真や動画といったコンテンツも提供可能なシステムを利用した。このシステムをデモで展示した。

今後は Web 上の情報収集部分の改善や、実験的に行った中継などのリアルタイムデータとの連動を考慮していく必要がある。また、今回のデモンストレーションの結果をもとに、入力操作部分の改善を行う予定である。

謝 辞

本研究およびデモンストレーションにあたり、サイバー関西プロジェクトより全面的に支援いただきました。関係者各位に深謝いたします。

参 考 文 献

- 1) 西尾 信彦, 坂本 憲昭, 新井 イスマイル, "Google Street View とマッシュアップした実世界指向 CMS", インタラクション 2009 論文集, pp.77-78, 2009 年 3 月.
- 2) WiiremoteJ, <http://www.wiili.org/WiiremoteJ>
- 3) Bluecove, <http://www.bluecove.org/>
- 4) バランス Wii ボード, <http://code.google.com/p/tokyo-jogging/>