

## VirtiGeo:衛星写真を自在に高解像度化するための 画像マッピングシステム

三浦 稔 隆<sup>†</sup> 中西 泰 人<sup>††</sup>

かつて GIS 研究者などの専門家にしか利用できなかった衛星写真は、近年の API による拡張可能な Web マッピングサービスの登場により、誰もが簡単に利用できるものとなった。また、球状に衛星写真を貼り巡らせることでバーチャルな地球を構築するソフトウェアも現れ、今や衛星写真は新たな情報表現の基盤になったとも言える。一方で、このようなサービスで提供される衛星写真は、地域によって解像度が異なったり、撮影時期の違いが原因で実際とは異なる風景を映し出すことがある。また、衛星写真では屋根までの情報しか表示されず、屋内の情報は表現しづらい。そこで本研究では、このような問題に対し、表示クライアントに Google Earth を用いたユーザが自ら撮影した画像を高さと共にマッピングできるシステムを構築することで、屋内の表現や衛星写真の高解像度化が行えるかを検証した。

### VirtiGeo: A System for Mapping High-Resolution Images on Google Earth

TOSHITAKA MIURA<sup>†</sup> and YASUTO NAKANISHI<sup>††</sup>

Aerial photos became very popular and easy to use, since the recent web mapping services or virtual earth software allow users to access their aerial photos through API or Software. But sometimes these photos are provided with low-resolution or shows different information due to filmed time. In addition, aerial photos can't show or express inside of the buildings, such as every floor's information. And users can't update or edit these photos. We create the system "VirtiGeo", which allow users to put their own photos on Google Earth with height parameter. In this paper, we propose the way and possibility of express inside of the buildings, and of replace low-resolution photos with users high-resolution photos by using this system.

#### 1. はじめに

近年登場したバーチャル地球儀ソフトウェアと呼ばれる Google Earth では、球状に配置された世界中の衛星写真に様々な情報がマッピングされ、ドラッグアンドドロップによる移動や回転、視点の傾斜機能などを用いて、様々な視点でバーチャルな地球を眺めることができる。加えて、ユーザが KML(Keyhole Markup Language)<sup>1)</sup> と呼ばれる XML ベースのタグを用いたファイルを定義することで、地球儀上の任意の座標に独自の画像やテキスト、3D オブジェクトをマッピングでき、また WMS(Web Map Service) などの本格的な Web マッピングの技術のサポートや、KML ファイルがインターネットを介して容易に共有できることから、旅の思い出や日記といった個人的な内容から気

象現象や都市生活者のアクティビティなどの学術的な内容に至るまで、幅広い情報をマッピングするサービス構築や試みが行われている。

KML を用いたサービス構築は容易である一方で、表示される衛星写真そのものをユーザが更新することは出来ないため、構築するサービスに一定の制限が課せられてしまう。提供される衛星写真は地域によって解像度が異なっており、都心部で提供されている高解像度の画像が、地方でも同じように提供されているとは言えない。また十分な解像度があったとしても、画像は常に最新のものではないため、新築の家があるべき場所が空き地になってしまっていたり、昼や夜といった現在の時間と連動した画像は表示されない。

また衛星写真の性質上、地表に最もズームした状態であっても表示されるのは建築物の屋根までであるため、地下街やビルの各階などの屋内の情報へはアクセスできない。このような「ビルの各階などの屋内の情報」の表現は、各階の画像を異なる高さで縦に積み重ねる手法で実現できると考えられるが、KML による標準の画像表示機能を用いた場合、すべての画像は地

<sup>†</sup> 慶應義塾大学総合政策学部

Faculty of Policy Management, University of Keio

<sup>††</sup> 慶應義塾大学環境情報学部

Faculty of Environmental Information, University of Keio

表に張り付けて表示されてしまうため、画像に高さを持たせられない。また、表示位置の設定には緯度・経度が必要とされる。GPS 機能内蔵のモバイルデバイスの普及によって、画像に関連する位置情報の取得は容易になったが、高さ情報や室内での位置情報取得は依然課題とされており、屋内などの細かな位置の設定には小数点レベルでの正確な緯度・経度の記述が必要のため、位置調整だけでも手間のかかる作業となる。また3Dオブジェクトの表示もサポートされているため、ビルをモデリングしたものに各階の画像をテクスチャマッピングして配置する手法も考えられるが、モデリングや画像の更新にかかるコストを考えると、一般的に普及する手法とは言い難い。このように Google Earth は非常に強力なツールである反面、表示される衛星写真は撮影時期によって実際の情報とは異なるものを含み、また地域による衛星写真の解像度の違いや、表現できる情報に限界があるといった問題がある。

そこで本研究では、Google Earth 上の任意の場所に、ユーザによって撮影された画像を高さのある状態でマッピングできるようにすることで、屋内などの衛星写真には写らない情報の表現を可能にすると共に、それらを組み合わせることで衛星写真の任意の地域を自在に高解像度化できるシステム“VirtiGeo”を提案する。また、面倒な作業が必要とされた画像の位置調整に関しても、視覚的に調整できる機能を提供することで屋内での位置情報取得の問題に対処すると共に、位置調整の簡便化を図る。

## 2. VirtiGeo

本章では、Google Earth 上にユーザが撮影した画像を高さのある状態でマッピングできるシステム:VirtiGeo にて実現される画像、要求される機能と実装、動作手順について述べる。

### 2.1 実現される画像

本システムを用いてビルなどの屋内の画像を、各階の高さに合わせて表示する手法を示す。まず、表示したい屋内の各階の画像を撮影する(図1)。撮影した画像は、本システムの提供する表示位置や高さを視覚的に調整する機能を用いて、適切な位置や高さに設定する(図2)。設定された高さや位置情報を基に本システムが KML を生成することで、Google Earth 上に高さを持った画像が表示される(図3)。一定の高さに画像が固定されているため、地面と垂直に視点を移動することで、最上階から1階までの各階の画像が表示される(図4)。

### 2.2 要求される機能

本システムに要求される機能を以下に示す。

- (1) 画像や位置情報の一元的な管理・蓄積
- (2) デジタルカメラや携帯電話付属のカメラにて撮影した画像の使用

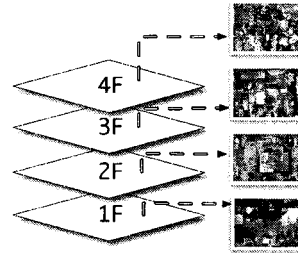


図1 表示したい屋内の各階の画像を撮影

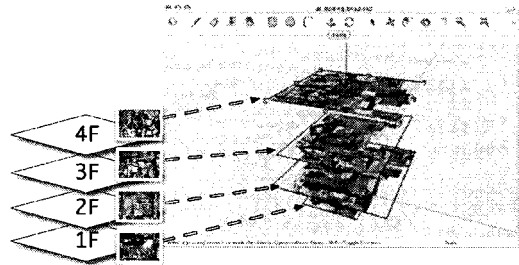


図2 視覚的な画像の位置調整機能

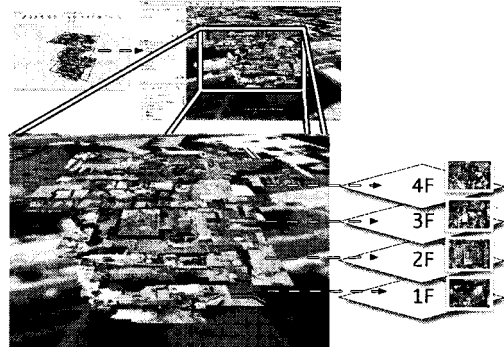


図3 高さに応じて表示された屋内の画像

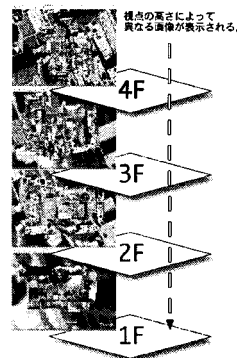


図4 視点の高さによって異なる画像が表示される

- (3) 画像の管理用インタフェース
- (4) 高さを表現できる画像表示機能

表示に用いる画像は、撮影された位置情報などと合わせてシステムが一元的に管理することで、複数の画像の組み合わせや位置情報と画像の関連付けが容易になると考えられる。また、表示する画像はデジタルカメラによって撮影したものだけでなく、位置情報を付加する機能を備えた携帯電話のカメラで撮影したものも利用できれば、位置情報による画像の管理が容易になるとともに、画像更新の即時性の向上が期待できる。 unnecessary 画像の削除といった編集作業が可能な管理用インタフェースも必要である。また、前述の通り KML による標準の画像表示機能では高さを表現できないため、別の方法による表示方法を実装する必要がある。

### 2.3 実装

以上の要求事項を受けて、実装したシステム構成を図 5 に示す。

#### 2.3.1 サーバ・クライアント方式による画像の配信

本システムは、Google Earth をクライアントとしたサーバ・クライアント方式を用いて構築した。サーバ側には Web サーバの他にメールサーバ及び空間データベースを設置した。

これにより、デジタルカメラにて撮影した画像は Web ブラウザとフォームを用いて投稿でき、カメラ付き携帯電話付属にて撮影した場合はそのままメール添付で投稿できる仕組みとした。また空間データベースを用いることで、画像に付随する位置情報の効率的な管理・蓄積を行えるようにした。

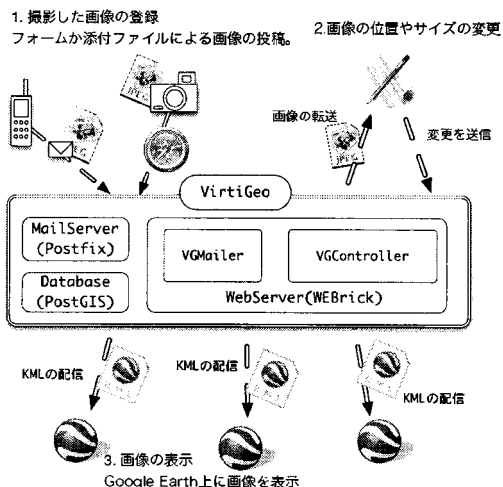


図 5 システム構成

#### 2.3.2 画像の管理用インタフェース

画像の一覧や削除などの簡単な管理作業は、提供する Web ブラウザ用インタフェースを用いて行う。加えて、現在の GPS では屋内の位置情報は正確に取得できないため、屋内の画像が表示対象でもある本システムでは、画像をドラッグアンドドロップすることで表示位置を設定できる機能を実装した。この機能は 3D モデリングツールである Google Sketchup(Sketchup) 用の Ruby プラグインとして実装した。このプラグインを用いることで、本システムのサーバから Sketchup に画像が転送され、登録された大まかな位置情報を基に Sketchup ドキュメント内の 3D 空間上に画像がマッピングされる。それらを任意の位置やサイズに移動・リサイズしたのち、同プラグインを用いて画像の表示位置の変更をサーバに送信すると、Google Earth を通じて見たその画像の位置やサイズも変更される。このような機能は、画像を 3D オブジェクトに埋め込む手法を用いることで実現しているが、詳しくは次節にて述べる。

#### 2.3.3 高さを表現できる画像表示機能

位置やサイズを修正した画像は、表示クライアントである Google Earth を用いて閲覧する。ただし、標準の画像表示機能では高さを表現できないため、別の手法が必要である。本システムでは Google Earth が 3D オブジェクトを表示できるという機能に着目し、画像を単なる画像ではなく 3D オブジェクト内の一部として定義することで、高さを表現することにした。ただし、画像の追加や更新を行う度に手作業でモデリングするのは先に述べたようなコストが伴うため、なんらかの手法を用いて画像の 3D オブジェクト化を自動で行わなければならない。まず、Google Earth 上に任意の 3D オブジェクト (someobject) を表示する際に必要とされるファイル及びディレクトリ構成を図 6 に示す。

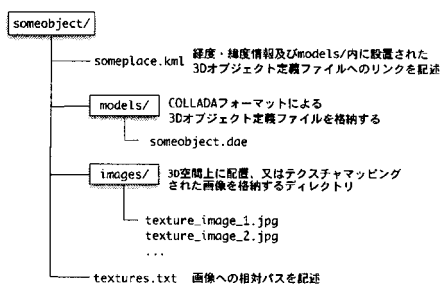


図 6 3D オブジェクト表示に必要なディレクトリ構成

通常の画像表示に必要とされるのが緯度・経度及び画像へのリンクが記述された KML ファイルのみであるのに対し、3D オブジェクト表示にはそれに加え図 6 のような複数のファイルによるディレクトリ

構成が必要とされる。また配置位置を KML ファイル内で緯度・経度を用いて指定するのは同じだが、リンク先は画像ではなく models ディレクトリ内の 3D オブジェクト定義ファイルを指定する。Google Earth では、表示する 3D オブジェクトの規格として COLLADA(COLLaborative Design Activity) フォーマットを推奨しているため、通常この 3D オブジェクト定義ファイルには COLLADA を用いた DAE(Digital Asset Exchange) ファイルが用いられる。COLLADA では 3D オブジェクトを XML スキーマを用いて定義する。例として、1 枚の画像を 3D 空間上に配置した DAE ファイルの一部を図 7 に示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<COLLADA version="1.4.1" xmlns="...">
  <asset>
    ...
  </asset>
  <library_images>
    # 配置する画像のパスを指定
    <image name="texture_image_1" id="texture_image_1">
      <init_from>../images/texture_image_1.jpg</init_from>
    </image>
  </library_images>
  <library_geometries>
    <geometry name="..." id="...">
      <mesh>
        <source id="...">
          # 画像の配置位置を、各4頂点の座標値の
          # 配列(XYZ, XYZ, XYZ, XYZ)で指定
          <float_array count="12" id="...">
            -2.121891 -0.000000 56.985906
            -1.727739 0.000000 53.214934
            -2.065583 0.000000 50.288210
            -2.578577 0.000000 47.330111
          </float_array>
        </source>
      </mesh>
    </geometry>
  </library_geometries>
</COLLADA>
```

図 7 1 枚の画像を 3D 空間上に配置した DAE ファイルの一部

図 7 にあるように、DAE ファイル内において 3D 空間上に配置する画像は <image> タグによって画像へのパスを参照し、<geometry> タグによって 3D 空間内における配置位置の座標値を定義している。また 3D 空間上に配置する全ての画像は images ディレクトリに格納され、DAE ファイルから相対パスによって参照される。textures.txt は図 8 のような各画像への相対パスのみが記述されたファイルである。

```
../images/texture_image_1.jpg<../images/texture_image_1.jpg>
../images/texture_image_2.jpg<../images/texture_image_2.jpg>
../images/texture_image_3.jpg<../images/texture_image_3.jpg>
...
```

図 8 textures.txt に記述された各画像への相対パス

これを受けて、本システムでは以下の作業を自動化することで、手作業によるモデリングを省いた画像の

3 次元オブジェクト化を実現する。

- (1) 図 6 で示したディレクトリ構成の生成
- (2) 配置したい経度・緯度、及び DAE ファイルへのリンクを含んだ KML の生成
- (3) 表示する画像の images ディレクトリへの格納
- (4) 表示する画像に対応した <image> タグと <geometry> タグの DAE ファイルへの書き込み
- (5) textures.txt への相対パスの記述

加えて、図 7 に示したように <geometry> タグでは画像の表示位置を、画像の各 4 頂点の xyz 軸の座標値を用いて指定するため、画像に付随する緯度・経度の値をこの座標値に変換する仕組みが必要である。また、3D オブジェクト化には xyz 軸の支点が必要であるが、全ての画像で 1 つの支点を共有する手法は、広大な 3D 空間内で位置調整を行う必要があるため、適切とは言えない。このため、本システムでは画像や位置情報の蓄積先である空間データベース内に、日本国土地理院の提供する地球地図日本のデータを用いて全国を行政区別に分類したテーブルを作成し、各行政区に xyz の支点を設けた。投稿した画像は付随する位置情報を基にどの行政区で撮影されたものか判別され、自動的に 3D オブジェクトとしての適当なサイズを持った xyz 値が与えられる。このように画像の投稿と共に 3D オブジェクト化が行われることで、編集クライアントである Sketchup による位置調整が可能になる。

また、表示クライアントである Google Earth は「移動が止まったとき」や「一定の高さに到達したとき」といった様々なイベントが KML を用いて利用できるように実装されており、イベント発生時には任意の URL へ指定したフォーマットを用いて現在表示している地域を表す緯度・経度をパラメータとして渡すことができる(図 9)。本システムではこのようにイベントと連動して渡される緯度・経度のパラメータを元にデータベースを検索し、現在表示されている範囲で登録された画像があった場合、登録された行政区に定められた xyz 軸支点を中心とした 3D オブジェクトを DAE ファイルにて定義し、図 6 に示したようなディレクトリ構造やファイルをサーバ側に生成し、そのサーバ上の DAE ファイルを参照する URL を含んだ KML を Google Earth に返すことで、登録された画像の表示を実現する。

```
<Url>
  <href>http://www.example.com/showImages.cgi</href>
  <refreshMode>onStop</refreshMode>
  <viewFormat>BBOX={bboxWest}/bboxSouth}/bboxEast}/bboxNorth}</viewFormat>
</Url>
```

↓

http://www.example.com/showImages.cgi?BBOX=130.21/35.98/132.21/30.64

図 9 KML を用いた指定したイベントとフォーマットによる表示範囲の取得



### 3. 考 察

実際に本システムを用いて丸ビルや4階建てビルの屋内の様子を表現する画像を制作した。4階建てビル制作の際に使用した各階の画像を図10に示す。これらは地面と垂直に撮影した複数の画像から構成され、それらを画像編集ツールなどを用いて一つの画像に組み合わせている。

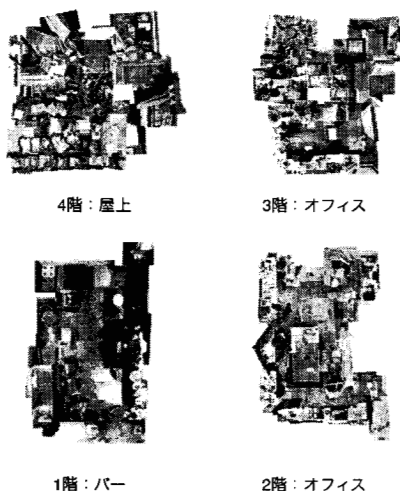


図10 撮影した4階建てビルの各階の画像

組み合わせた画像を経度・緯度とともにシステムへ登録することで、システムによって自動でどの行政区にあるかを検出され、画像に対して適当な3次元座標値が割り当てられる。登録された画像はSketchup向けの編集用プラグインを用いることでサーバからダウンロードされ、図2のように割り当てられた座標値を基にSketchup上に画像がマッピングされる。同プラグインでは画像の撮影された行政区の地図も編集用画面にマッピングされるため、これを参照しながらドラッグアンドドロップで画像の位置やサイズを調整することで、以前の緯度・経度などを用いた数値による調整よりも容易に行えた。変更後は、同じく本システムの提供する変更用プラグインを用いることで、編集画像に対応したサーバ側の画像情報が更新される。画像の合成は手動であるため手間のかかる作業であったが、Sketchupプラグインを用いた画像の位置調整やリサイズを視覚的に行えたことから、以前の数値による調整よりも短い時間で調整できた。編集を終えた画像は、本システムのサーバへアクセスするKMLを開くことでGoogle Earthが起動し、指定した位置と高さに表示されたのが確認できる。表示される画像は図

3のように各階の高さに見合った位置にマッピングされるため、視点を垂直に上げたり下げたりとすることで、屋上や屋根を突き抜けて各階の様子を確認することができた。その他にも、標準の画像表示機能と本システムを組み合わせることで、画像は1地点に固定されたまま、視点の高さによって解像度の異なる画像を次々に入れ替えることでチャールズ・イームズらによるPowers Of Ten<sup>11)</sup>のような表現も可能である。このため、本システムを用いることで、これまで衛星写真では表示できなかった屋内などの表現や、任意の地域にこれを配置することでの衛星写真の高解像度化が可能になったと考えられる。

### 4. 関連研究

本章では、本システムの行った手法や地図上に画像を表示する手法に関連のある、画像による疑似3次元空間の構築の関連研究について述べ、次に画像を用いた地図の有用性について述べる。

#### 4.1 画像による疑似3次元空間の構築

画像を用いて疑似3次元空間を構築する手法は「Image-Based Rendering」と呼ばれ、複数の写真を合成やリンクさせることで、連続的な疑似3次元空間を構築するものである。初期に最も普及したQuickTime VR<sup>2)</sup>では、パノラマ空間構成手法を用いて複数の写真を合成することで、1地点から360度全周を眺めるような効果が実現された。しかし、1地点からの移動ができないことや、360度全周の画像が必要、パノラマを構成する一部の画像の更新ができないなどの問題があったため、画像を1枚ずつ手動によるリンクでつなげて各画像間を移動することで仮想3次元空間を実現するTown Digitizing<sup>3)</sup>やIBNR<sup>4)</sup>、さらに複数人による画像の編集や空間ハイパーリンクを用いたPhotoWalker<sup>10)</sup>といった手法が提案された。これらの問題は表示される画像が1枚であるため、現在地を俯瞰的に把握できないことであった。こうした問題を解決するために、手動によるリンク設定では無く画像解析を用いてカメラの撮影位置を自動算出し、画像間のリンクだけではなくカメラ位置も確認できるようにした手法がJohansson<sup>5)</sup>らによって提案された。これに大量の画像の処理や、異なる画像間で共有されるオブジェクトへの注釈付け機能などを加えたものがPhotoSynth<sup>6)</sup>である。PhotoSynthでは、カメラ位置の抽出によって各写真の撮影時の位置を判定できるので、現在地や各画像の撮影時の位置を俯瞰的に把握できるようになったが、異なる場所で撮影された画像は共存できず、また撮影対象の位置や複数人による編集、もっと広い視野で見た際の位置関係までは原理的に把握できない。

本システムでは、衛星写真の貼り巡らされたGoogle Earth上に高さのある画像を表示することで、このよ

うな疑似3次元空間を構築するシステムだとも言える。加えて本システムは従来研究に見られる異なる撮影場所に対応できないといった問題や、俯瞰図の実現に関する技術的な問題を克服し、また Google Earth によって標準で提供される様々な情報をそれらの画像と同時に表示できることから、より現実空間に近い疑似3次元空間を構築するシステムだとも言える。また、現在の本システムでは空間を俯瞰する2次元の画像を3次元的に配置しているが、Image-Based Rendering などにより制作された室内を表現するオブジェクトを配置することも可能である。

#### 4.2 画像を用いた地図の有用性

地図上に画像を表示する有用性は、主に CyberGuide<sup>7)</sup> に代表される道案内システムにおいて研究されてきた<sup>8)</sup>。A Natural Wayfinding<sup>9)</sup> では曲がり角や迷いやすい場所で撮られた写真を PDA 上の地図に表示し、ユーザが写真と景色を見比べることによって目的地まで移動する実験を行い、写真による道案内は効果的であるという結果が出た一方、システムがユーザによる写真の投稿や更新に対応していないため、天候や季節によって生じる写真と実際の景色との食い違いでユーザが混乱してしまったという結果も出ている。

これらの研究では主に PDA などのモバイルデバイス上の地図を用いている点で本研究とは異なるが、地球全体の任意の地域にユーザによる更新が可能な画像をマッピングできる本システムにおいて、衛星写真では表示されない地下街や迷いやすい建物内のフロアガイド製作といった、道案内製作システムとしての側面を持つ本システムの有用性は十分にあると考えられる。

### 5. まとめおよび今後の課題

本研究では、Google Earth 上の任意の場所に、ユーザによって撮影された画像を高さのある状態でマッピングできるようにすることで、屋内などの衛星写真には写らない情報の表現を可能にすると共に、それらを組み合わせることで衛星写真の任意の地域を自在に高解像度化できるシステム“VirtiGeo”を提案した。また、緯度・経度指定を用いずに視覚的に画像の位置を調整する手法も提案した。本システムは Google Earth を用いて画像の蓄積・編集・配信を行えることから、センサ情報やネットワークカメラなどと連携することで、衛星写真の問題を解決するだけでなく、よりダイナミックなバーチャル地球儀を構築できるシステムだと言える。

今後の課題として、今回手動にて負担の大きかった画像合成作業の改善が挙げられる。画像処理技術などを用いた合成の自動化が必要だろう。また、今回は表示クライアントとして Google Earth や Google Sketchup を用いたため、KML の書き出しや特殊な

ディレクトリ構造の生成、Sketchup を基にした 3D 空間の座標値設定など、実装の汎用性は低いものになっている。WMS などの手法を用いて本システムの画像やデータを配信できるようにすることで、その他の衛星写真を用いる GIS ツールやサービスでの衛星写真の抱える問題への解決策となるだろう。また、今回画像の管理などに日本国土地理院の提供する地球地図日本のデータを用いたため、日本以外の国での管理方法の考案も必要である。

### 参考文献

- 1) KML: <http://earth.google.com/kml/>
- 2) Abowd, G.D., et al., Cyberguide: A Mobile Context-Aware Tour Guide. ACM Wireless Networks, (3): pp. 421-433.(1997)
- 3) Burigat, S, Chittaro, L. Location-aware Visualization of VRML Models in GPS-based Mobile Guides. Proc. 3D Web Technology, pp.57-64(2005)
- 4) S. E. Chen: QuickTime VR An Image-Based Rendering to Virtual Environment Navigation, Proc. ACM SIGGRAPH '95, pp. 29-38 (1995)
- 5) JOHANSSON, B., AND CIPOLLA, R: A system for automatic pose-estimation from a single image in a city scene. In IASTED Int. Conf. Signal Processing, Pattern Recognition and Applications(2002).
- 6) Satoshi Koizumi, Hiroshi Ishiguro: Town Digitizing: Omnidirectional Image-Based Virtual Space, Springer-Verlag, Berlin Heidelberg, pp. 247-258,(2005)
- 7) Ashweeni Kumar Beeharee, Anthony Steed: A Natural Wayfinding - Exploiting Photos in Pedestrian Navigation Systems ACM International Conference Proceeding Series:Vol.159, pp.81-88(2006)
- 8) S NAVE LY N., S E I T Z S., S Z E L I S K I R.: Photo tourism: Exploring photo collection in 3d. In IEEE Trans. on Computer Graphics (SIGGRAPH) (2006).
- 9) M. Tsukamoto: Image Based Pseudo 3-D Visualization of Real Space on WWW, Digital Cities: Technologies, Experiences, and Future Perspectives, Lecture Notes in Computer Science, Springer-Verlag, Vol. 1765, pp. 288-302 (2000).
- 10) 田中浩也, 有川正俊, 柴崎亮介, "空間ハイパーリンクを用いた写真画像群の疑似3次元連携", 情報処理学会誌:データベース, Vol.44, No.SIG3 (TOD17), pp.11-21,(2003)
- 11) フィリップ・モリソン, 村上陽一郎, 村上公子: パワーズ オブ テン—宇宙・人間・素粒子をめぐる大きさの旅 (1983)