

カメラ画像を利用した仮想都市空間の生成と可視化

池内克史

情報理工学系研究科電子情報学専攻

概要

本研究の目的は、図1のような実空間情報収集車両を利用して、仮想都市空間内を車両で自由に走行できるドライビングシミュレータの表示画像を簡易に生成し、その結果として現実の活動を反映した仮想都市空間の可視化を可能とするものである。

1. はじめに

現実活動を反映した仮想都市空間として、実際の交通状況に応じて仮想都市内に車両を生成し、その中をドライビングシミュレータで走行することで可視化するという方法が考えられる。ドライビングシミュレータは大変臨場感溢れ、また、実際の車両を運転席に据えることで、運転者属性（ヒューマンファクタ等）を取得するためのセンサの役割も持つ。

しかしながら、通常のドライビングシミュレータに利用されている道路交通空間の表示はコンピュータグラフィックスを利用した簡易なものであり、やや現実感に欠けるきらいがある。また、ビルなど個々の建物を1つ1つの立体としてデータ化しているため、こうしたモデルの作成に多大な時間と費用がかかるのが現実である。さらに、サグ部のような、人間の錯視が影響すると思われる部分では、こうしたモデルベースの手法よりも、実際の映像に近い表示系を生成する方が、個人の属性をより精度よく取得できると期待できる。

そこでこれらを改善する手法として、車載カメラにより取得された映像を利用して、実際の環境に極めて近く、かつ、既存のドライビングシミュレータに融合可能な道路交通空間の表示システムを構築する。昨年度までに、実写ビデオ画像からの任意視点映像の生成、およびその映像と既存のCGベースのドライビングシミュレータ画像との融合を行ったが、本年度は、この手法を適用する際に発生した2つの問題点の検討と、その解決方法、そしてそのうち1つについてはプロトタイプによる実装を行ったので、これを報告する。



図1. 周辺環境画像収集車両

2. 実写画像を用いた実環境表示手法の開発

2.1 9台ビデオカメラ映像から全方位画像の生成

ユーザに提供する周辺環境画像は実際に対象となる道路を撮影用車両で走行して取得し、処理をして与える。図1に撮影用車両を示す。屋根上には9台のビデオカメラが設置されており、これらの画像を統合することにより全方位画像を生成する。後述の

ように、走行経路上からの見えを全方位画像として残しておくことにより、走行経路外からの見えを画像処理によって合成することができる。したがって、撮影走行は一度だけで充分である。

カメラの光学中心が今回のように一致していない場合、これらの画像を統合すると、一般に対象物の奥行きによって統合画像の境界部分にひずみが発生することが知られている。そこで、個々のカメラを進行方向に対して平行に並べて図2のように時間差をつけて光学中心を一致させることでこの問題を解決した。個々の画像を全方位画像に統合するためには、各カメラ間のキャリブレーションを行う必要がある。このうち回転移動成分については画像中の建物などのエッジから消失点を利用して求めた。また平行移動成分は、各カメラの高さおよび左右位置は並んでおり等しいと仮定する。各カメラの間隔についてはそれぞれのカメラから得られる EPI (画像中のある水平 1 ラインだけを取り出し、フレーム毎に並べた画像) どうしの二次元マッチング (図 3) により取得する。マッチング結果により、図 2 における t_1-t_2 , t_2-t_3 の値がわかるので、その時間だけずらした画像を統合することで、図 4 のような全方位画像が得られる。なお、ここでは、簡略化のため、全方位ではなく前方半方位のみを示している。

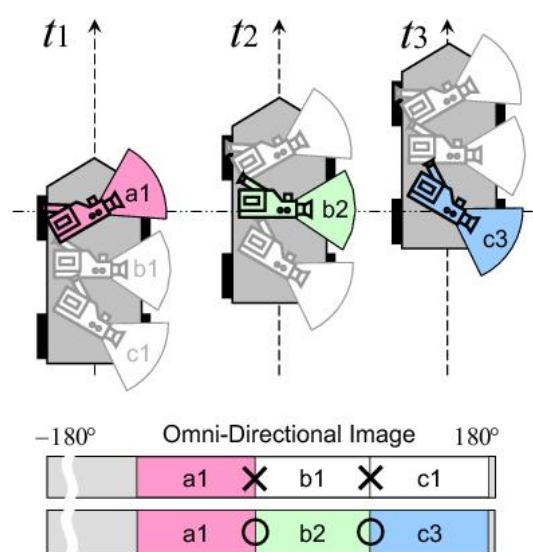


図 2. 時空間的な光学中心の一致

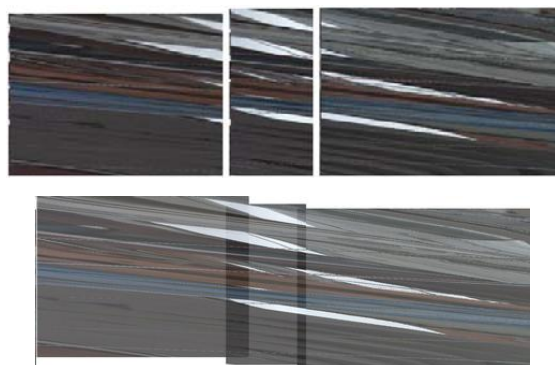


図 3 EPI マッチング(左：3カメラの校正 EPI 画像、右：マッチング結果)



(a) 統合前 (中央上部は地図)



(b) 統合後

図4 全方位画像 (前方半方位画像) の例

2.2 ユーザ視点からの見えの構築とその表示

(1) 任意視点画像合成の基本概念

前節の方法で生成した全方位画像は、計測車両が走行した軌跡の各点における見えを生成しているにすぎない。しかしながら、ドライビングシミュレータへの提示画像においては、被験者は必ずしも計測車両が走行した経路を通るとは限らない。しかしながら、走行経路の全方位画像を用意しておくと、その経路上になかった (例えば他の車線からの) の見えを生成することができる。その原理を図5に示す。

図5において、星 (☆) 印の地点は撮影時には走行していないが、そこからの左方の見えは、それぞれ時刻 t_1 、 t_2 、 t_3 における全方位画像の左前方、左方、左後方の見えから合成することができる。これら t_1 、 t_2 、 t_3 は、前節で述べた光学中心とは異なる。

ることに注意する。ビデオカメラ映像の場合、各全方位画像は 1/30 秒毎に生成されるので、 $t_2-t_1 = 1/30$ 、 $t_3-t_2=1/30$ となっている。

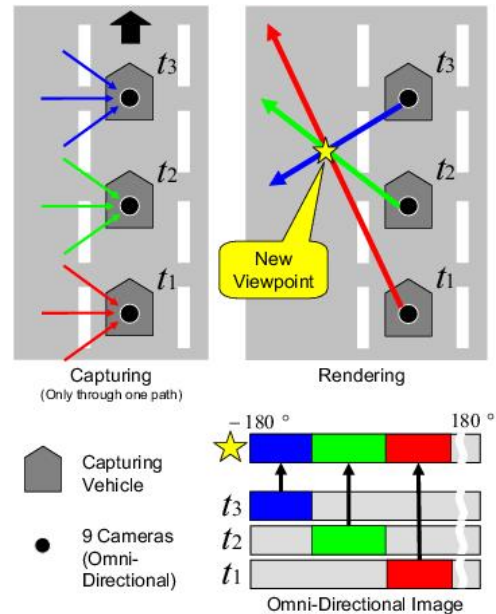


図5 新しい視点からの見え生成

こうして得られた見えは、ビデオ映像を取得した時に撮影された物体がすべて映りこんでいるので、このまま提示すると近景の走行車両 (対向車、先行車等) も含まれている。しかも、ここで得られた映像には、幾何学的情報が一切ないので、生成された見えのどの部分が道路や先行車であり、どの部分が背景かがまったくわからない。しかしながら、背景の建物は道路から一定の距離の部分に垂直に立っていると期待できるので、ここで生成した画像を図6に示すような「板」に投影する。この板は、道路や車両といった近景と背景の建物や樹木といった遠景との境界にあると考えられる。任意視点からみた遠景はすべてこの板の位置にあると仮定することで、生成した見え

をこの面上のテクスチャと考えることができる。すなわち、全方位画像から生成された任意視点からの見えをこの仮想的な板に貼り付けることで背景の提示を実現する。この板は近景より遠方にあるために、道路や車両、標識等を従来どおりの近景として表示すれば、近景部分は従来どおりで、背景部分だけ今回提案の手法で生成された画像が提示できることになる。

なお、この板面は実際は垂直なスリットに分かれており、スリット毎に適当な地点から撮影された全方位画像の一部が貼り付けられる。

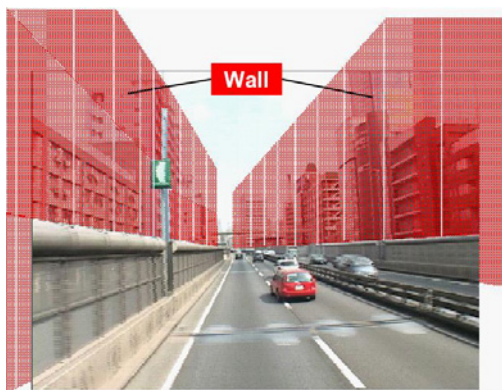


図6 仮想的な板

このテクスチャは視点（ドライブシミュレータの自車両位置）が変わるたびに変わるので、描画毎（1/60秒周期）で動的に更新される。したがって、例えばビルに近づくと、遠方からは見えない側面などがだんだん見えてくるように変化することになる。また、テクスチャの貼り付け面は常に視線方向を向くように回転させる。図7に示すとおり、視点位置に応じて生成した見えを板にはりつけるのであるが、その際、板を一直線に並べたままでは、視線方向と板の法線方向が異なるため見えが

歪んでしまう。そこで、見えが歪まないようにビルボーディングと呼ばれる手法を用いて板を細かく分割し、分割した板の法線方向が常に視点を向くようにしておく。これによって歪みのないきれいな見えを生成することができた。

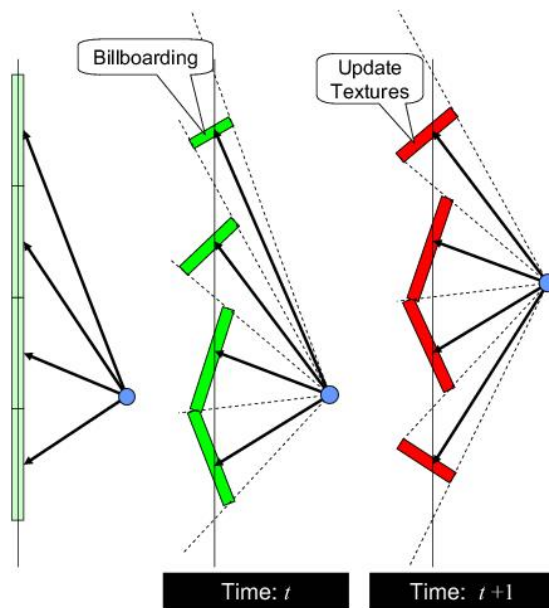


図7 テクスチャの更新とビルボーディング手法

(2) 高速化

上で述べたように、テクスチャの更新は視点位置、すなわち、ドライブシミュレータの自車両位置が変わるたびに行う必要がある。車両位置の更新レートは従来のドライブシミュレータで60Hzであるから、これに合わせる必要がある。しかしながら、この周期で前節の処理を毎回行ってグラフィックハードウェアに読み込ませると時間のロスが非常に大きく、更新レートを60Hzにするのは極めて困難である。

そこで、板面上のスリット毎に、あらゆる方向からの見えをあらかじめデータセットとしてメモリに保持しておき、必要に応じて呼び出す手法をとる。図8にデータセ

ットのイメージを示す。

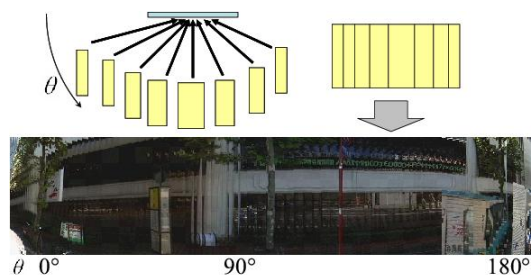


図8 テクスチャデータセットの概念

(3) 品質の向上

テクスチャ貼り付け面上のデータセットにはその面の各方向からの見えが収められているが、この方向は離散的にしか生成することができない。これは、撮影時のビデオカメラのフレームレートが有限であるため、全方位画像が走行系路上で離散的に存在していることによる。データセットが保持している特定の 2 方向の中間に相当する見込み角からの見えが要求された場合は、それらのテクスチャをアルファブレンディングによって補完することにより表示品質を向上させる。

このようにして生成した見えを以下に示す。いずれの場合も、更新フレームレートは 60Hz で表示を行っている映像から 1 枚を切り出したものである。

3. コンピュータグラフィックスと実写画像との融合

前節の方法で、前景と近景を合成することができることがわかった。コンピュータグラフィックス部分との独立性を保つためと高速性を確保するため、ハードウェアにより従来のコンピュータグラフィックス画像と今回提案した実写画像との融合を行うことにした。



(a) 従来のドライビングシミュレータ画像



(b) 走行経路上から正面を見る



(c) 走行経路外から左 45 度を見る



(d) 走行経路外から右 45 度を見る

図9 生成画像の例

図10に今回のシステムのハードウェア構成を示す。イメージベースである遠景部分の処理は必要に応じて複数台のマシン(FarView1,2)によって行う。処理部からはピクセルごとにカラー値と奥行き値が出力される。近景は従来のドライビングシミュレータの出力(コンピュータグラフィックス)を行うマシン(NearView)によって出力される。これらの出力はコンポジットと呼ばれるハードウェア合成装置を経由することにより奥行き値を考慮してピクセル毎のカラー値が決定され、プロジェクタを通じてユーザ前のスクリーンに投影される。

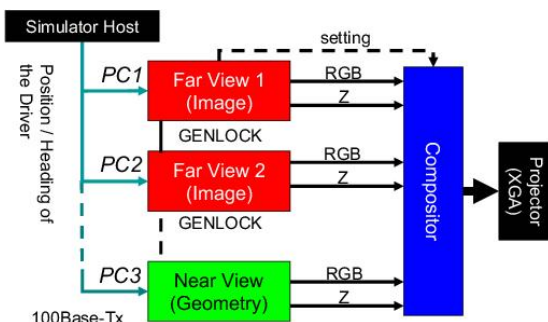


図10 システムのハードウェア構成

なお、今回のシステムで遠景部分の表示処理に用いた FarView マシンのスペックは以下の通りである。

- CPU: Intel Pentium4 2.80C GHz
- グラフィックボード : QuadroFX3000G
- メモリ: 2GB
- OS: Windows XP

3. 2 具体的な事例への適用例

図11に今回の手法と従来手法を合成した結果を示す。図9(a)の従来手法だけの映像と比較すれば、本手法が非常に現実感の高い画像をユーザに提供していることがわ

かる。既に述べたように、仮想板のおかげで、従来手法で描かれている近景(道路周辺映像)と今回提案した手法で描かれた遠景(ビル等の背景)が違和感なく合成されていることがわかる。

また、実際にカメラを搭載した車両を走行させてからこのような映像を作るまでに必要な作業時間は、まだ完全自動化ができないため、多少人手がかかるため1ヶ月くらいである。しかしながら、従来では、背景に現れる町の3次元モデルや見えをグラフィックが手作業で生成しているために、1年程度かける必要があったので、本手法により少なくとも背景に関しては1桁以上早く生成が可能になったと結論できる。



図11 イメージベース(遠景)とジオメトリベース(近景)の合成結果

4. 2つの問題点と解決手法の提案

このようにして、昨年度までに開発したシステムであるが、この表示方法では次のような2つの問題が残る。

1. 撮影した画像中に道路をまたぐ構造物があった場合、これらが側面の仮想壁に「はりつく」形になり、これが違和感として残る。
2. 多くの全方位画像から切り貼りして合成画像を作るため、車両の振動があると、そのままでは合成画像が波打つ形になる。

それぞれについて、解決方法を検討し、一部を実装したので報告する。

4.1 道路横断構造物による違和感の解消

上で述べた違和感とは、図 12 に示すようなものである。この図において、右側に緑色の壁として見えているものは、実は、道路上に設置されている案内標識である。本システムでは、すべての構造物は道路脇のビルと同じ位置に存在すると仮定しているために、このように道路を横断的にまたぐ構造物に関しても無理やり道路脇の位置にあるとして描画する。道路脇の構造物であれば、多少の違和感があっても気にならないが、このように道路上部にある構造物はその実際の位置と描画位置が大きく異なるために違和感が発生してしまうのである。



図 12 道路横断構造物（案内標識）による違和感の例

この問題を解決する方法として、まず最初に、横断構造物の位置を半自動で計算し、それに基づいて元々の全方位画像から横断構造物を消去した画像を新たに生成するという手法を考えた。

すなわち、CG 高速道路の 3 次元形状は既知なので、これは実際の形状とほぼ一致

していると仮定し、そのデータに基づいて画像中から手動で横断構造物の位置を指定する。そして、その指定した位置に横断構造物の後方の画像を貼り付けてしまうという手法である。

ただし、後方の画像は現実には撮像することができないので、ここでは、車両が丁度構造物の真下を通過したときの画像から、適切な画角（位置・姿勢）の部分を取り出して構造物の部分に貼り付けることにした。

首都高 3 号線の場合は、全部で 10 個程度の横断構造物があったので、それらすべてについてこの処理を行い、全方位画像からすべての道路横断構造物を取り除いた。なお、これらの案内標識は CG ベースの既存手法では表示されるため、実写画像から取り除いても何の問題もない。むしろ、2 重に表示されることのないだけ有利である。

なお、実際にこうした処理を行う際には、各構造物についてその大まかな位置（2, 3 のフレームから構造物の 4 つのコーナーを指定）、消す作業を行う区間（どのフレームからどのフレームまで処理を行うか）を手で指定する。すると、その後の修正全方位画像は自動で生成するようにしてある。したがって、横断構造物が多数あると面倒ではあるが、作業自体は単純なものである。

図 13 にこうした処理を施した結果を示す。構造物が見当たらないが、よくみると、画面右側に柱らしきものが立っており、また右側のビルが一部真ん中がえぐれている。このえぐれている部分にももとは横断構造物があったのだが、別の画像で消去しているのである。このようによく見ると若干の違和感は残るものの、図 12 よりは違和感が軽減しているといえる。



図 13 道路横断構造物を消去した例

この手法の問題点は、すべての道路横断構造物を手で指定しなければならないこと、実際の道路形状と CG による高速道路の形状がことなるために、図 13 にあるように、消去した部分が完全に背景に溶け込まずに、どうしても誤差によるズレが発生することがあげられる。そのため、こうした修正を行って見栄えのよい背景画像にするには、かなりの試行錯誤と手作業が要求されることになる。また、交差道路のような本来消去すべきでない構造物まで消去する必要がある。

そこで、より簡易にこうした違和感をなくす方法として、次の図 14 に示す方法を検討している。図 12 のような違和感が発生する理由は、横断構造物に関しては余りにも幾何形状が異なる近似を行っているからである。したがって、道路上部にも仮想壁を用意すれば、幾何形状の近似がより近くなって、違和感が低減されると期待される。

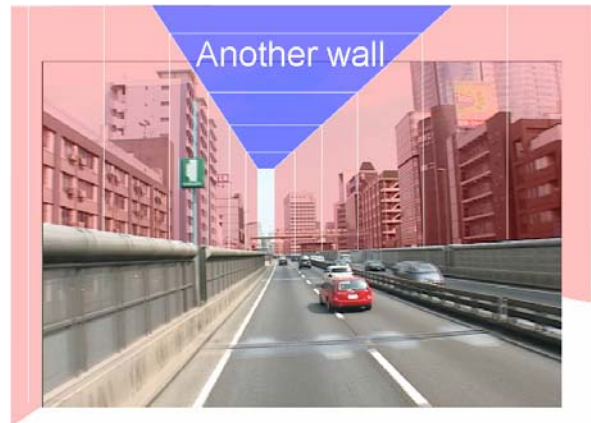


図 14 新たな上部壁の利用



図 15 都市間高速道路における道路横断構造物の例

これについては、現在実装中であるが、例えば東名高速道路大和サグ近辺では、図15に示すような横断構造物が多数あるため、こうした上部壁を利用した表示方法が有効であると思われる。

4.2 計測車両の振動による影響の除去

図8に示したように、ドライビングシミュレータ用の提示画像は、多くの全方位画像からその一部を切り出して張り合わせたものになっている。しかしながら、現実には、車両の振動によって取得した全方位画像の路面からの姿勢が一定ではない。そのため、1つの合成画像だけを取り出せば違和感は少ないのであるが、連続してムービーの形にすると、その姿勢の違いがビル群の高さの違いを引き起こし、まるで津波でも発生しているかのような違和感が発生する。

これを解決するために、全方位画像を取得していることを利用して、各フレームで車両の真横に映っているビルの特徴を頼りに、車両の姿勢を推定し、その推定した姿勢に基づいて全方位画像の位置修正を行うことを提案する。

現状、靖国通りで取得した全方位画像に対して本手法を提案したところ、十分な効果が得られることを確認している。

5. まとめ

本年度は、昨年度までの成果に基づき、次の2つのことを行った。まず、道路横断構造物がある場合に発生する提示画像の違和感の解消を行った。そして、車両の振動による生成画像の揺れについてもその対策を講じプロトタイプでその効果を確認した。

さらに、都市間高速道路における多数の道路横断構造物の違和感解消のために、新たなシステムの提案を行った。今後は、これらの成果を踏まえて、都市間高速道路や一般道など、さまざまな分野への応用を展開していく予定である。

発表文献

[1]小野晋太郎, 平原清隆, 影沢政隆, 池内克史, "車載レンジセンサを利用した路上駐車車両の自動検出," 電子通信情報学会論文誌 基礎・境界, Vol.J88-A No.2, pp.247-256, 2005年2月

[2]小野晋太郎, 小川原光一, 影沢政隆, 川崎洋, 大貫正明, 本多建, 池内克史, ": 画像ベースレンダリングによる高速道路サグ区間における運転シミュレーション映像の生成 (計画概要)," 第4回 ITS シンポジウム, 2005年12月

[3]トウ利洪, 小野晋太郎, 影沢政隆, 池内克史, "City Range Data Analysis," 第4回 ITS シンポジウム, 2005年12月

[4]太田亮, シャミラ モホッタラ, 小野晋太郎, 影沢政隆, 池内克史, "VEHICLE CLASS RECOGNITION OF STREET-PARKING VEHICLES FROM SIDE-VIEW RANGE IMAGES," 第12回 ITS 世界会議, 2005年11月

[5]小野晋太郎, 小川原光一, 影澤政隆, 川崎洋, 大貫正明, 本多建, "Driving View Simulation with a Hybrid Synthesis of Virtual Geometry and Real Image in the Mixed-Reality Traffic Experiment Space," 4th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 05), 2005年10月

[6]Shintaro Ono, Koichi Ogawara, Masataka Kagesawa, Hiroshi Kawasaki, Masaaki Onuki, Junichi Abeki, Toru Yano, Masami Nerio, Ken Honda Katsushi Ikeuchi, "A Photo-Realistic Driving Simulation System for Mixed-Reality Traffic Experiment Space," IEEE Intelligent Vehicles Symposium (IV 2005), 2005年6月

[7]太田 亮, シャミラ モホッタラ, 小野晋太郎, 影沢政隆, 池内克史, "レーザセンサから取得した距離画像からの車種認識," 交通・電気鉄道合同研究会 ITS, 2005年11月

[8]小野晋太郎, 小川原光一, 影沢正隆, 川崎洋, 大貫正明, あべ木順一, 矢野徹, 練尾正美, 本多建, 池内克史, "実写画像の合成による運転シミュレーション映像の構築," 画像の理解・認識シンポジウム (MIRU2005), 2005年7月