

修士論文

全方位画像を用いた広域都市空間の全自動生成

平成13年2月2日提出

指導教官

池内 克史 教授

東京大学大学院 工学系研究科 電子情報工学専攻

96414 高橋 拓二

# 目次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	本研究の背景と目的	2
1.2	本研究の構成	3
<b>2</b>	<b>関連研究</b>	<b>5</b>
2.1	はじめに	6
2.2	CGによる3次元仮想空間を生成する研究について	6
2.2.1	モデル・ベースト・レンダリング	6
2.2.2	イメージ・ベースト・レンダリング	11
2.2.3	イメージ・ベースト・レンダリングの欠点と改善方法	15
2.2.4	データ量(取得画像)と画像品質に関する研究	16
2.2.5	都市などの広域空間を対象とした研究	18
<b>3</b>	<b>任意視点からの画像合成</b>	<b>21</b>
3.1	はじめに	22
3.2	全方位画像の取得	22
3.2.1	全方位画像とは	22
3.2.2	全方位画像取得のためのシステム	23
3.3	仮想視点からの視界合成	25
3.3.1	はじめに (IBR)	25
3.3.2	画像の合成方法	25
3.4	視点合成が可能な領域	25
3.4.1	はじめに	25
3.4.2	2次元上の、合成可能領域	26
3.5	進行方向の光線	28

3.6	任意視界の画像合成結果 . . . . .	30
3.6.1	はじめに . . . . .	30
3.6.2	シミュレーション画像による結果 . . . . .	30
3.6.3	実画像による結果 . . . . .	33
4	生成された画像における問題点の検討	39
4.1	はじめに . . . . .	40
4.2	横方向のひずみ . . . . .	40
4.3	縦方向のひずみ . . . . .	42
5	都市空間の特徴を利用した画像生成	44
5.1	はじめに . . . . .	45
5.2	都市の建物と、それを応用した設定 . . . . .	45
5.3	デプスの推定方法 . . . . .	46
5.3.1	はじめに . . . . .	46
5.3.2	オムニ画像からの自動抽出 . . . . .	46
5.3.3	その他のデプス値を推定する方法 . . . . .	49
5.4	3次元情報によるひずみの補正方法 . . . . .	51
5.5	3次元情報を元に補正した結果 . . . . .	54
5.5.1	はじめに . . . . .	54
5.5.2	シミュレーション画像における、補正した結果 . . . . .	54
5.5.3	実画像における、補正した結果 . . . . .	54
6	結論	59
6.1	結論 . . . . .	60
6.2	応用と今後の課題 . . . . .	62
	謝辞	63
	発表文献	64
	参考文献	65

## 目 次

2.1	佐藤らによるハイライト等の復元 . . . . .	9
2.2	Debevec による 3 次元形状の復元 (その 1) ムービー中の 16 枚の画像からポリゴンモデルを生成する。この例では右側の画像を生成している。また標準的なグラフィック用のハードウェアでリアルタイムでレンダリングが可能である。 . . . . .	10
2.3	Debevec による 3 次元形状の復元 (その 2) 先端付近の画像 . . . . .	10
2.4	4D parameterization in 光線空間 . . . . .	13
2.5	4D parameterization in Lumigraph or Light Field Rendering . . . . .	13
2.6	3D parameterization in Concentric mosaics . . . . .	15
2.7	Minimum sampling curve for IBR . . . . .	17
2.8	図 2.7 の各点における画像生成結果 . . . . .	17
2.9	奥行き情報を利用した 3 次元都市 . . . . .	19
2.10	Mixed Reality における画像撮影用の車 . . . . .	20
3.1	(a) オムニカメラで撮影した画像の持つ光線 (b) 円筒状に配置されたカメラで全方位画像を撮影 . . . . .	23
3.2	実世界での画像撮影用の車 . . . . .	24
3.3	画像の再構築方法 . . . . .	26
3.4	2 次元上の合成可能範囲 . . . . .	27
3.5	(a):singular direction を含む場合の画像生成 (b):singular direction を含まない場合の画像生成 . . . . .	28
3.6	シミュレーション用全方位画像 . . . . .	31
3.7	道路に対して垂直に移動した時の視界の作成結果 . . . . .	31
3.8	道路と平行に走行した際の視界の作成結果 . . . . .	32
3.9	回転運動時の視界の作成結果 . . . . .	32

3.10 (a): HyperOmni camera で撮影したオムニ画像	(b):
(a) から作成した perspective 画像 . . . . .	34
3.11 屋内で撮影したオムニ画像での実験の位置関係図; 図 3.12 に示した生成図は、それぞれ点 (a)、(b)、(c) から見た際の図である。 . . . . .	34
3.12 生成した画像; 画像 (a) では黒板の左端と本棚の右端は完全に分かれているが、画像 (b) では、それらはちょうど重なっている。さらに画像 (c) では、本棚の右端が黒板によって隠されている。 . . . . .	35
3.13 横浜市で撮影した画像を用いて作成した全方位画像 . . . . .	36
3.14 画像作成の位置関係: 方位画像を撮影しながら道路上を走行したが、それを $Y_1$ 軸とする。図 3.15 は仮想走行路-(1) からの眺めであり、すなわち $X$ 軸からの眺めである。図 3.16 は仮想走行路-(2) からの眺めであり、これは $Y_2$ 軸状からの眺めである。 . . . . .	37
3.15 仮想走行路-(1) からの眺め . . . . .	37
3.16 仮想走行路-(2) からの眺め。これらは実際に画像を撮影した走行路以外の別の走行路であるといえる。 . . . . .	38
4.1 横方向のひずみ . . . . .	41
4.2 縦方向のひずみ . . . . .	43
5.1 都市空間におけるデプス値の設定 . . . . .	47
5.2 オムニ画像において特徴点抽出をおこなった結果 . . . . .	50
5.3 連続オムニ画像における特徴点抽出と円の fitting . . . . .	51
5.4 3次元情報による補正方法 . . . . .	53
5.5 実験に用いた、Open GL により作成したパノラマ画像 . . . . .	55
5.6 3次元情報による補正前 . . . . .	55
5.7 3次元情報による補正後 . . . . .	56
5.8 信濃町付近で撮影した画像を用いて作成した全方位画像 . . . . .	57
5.9 3次元情報による補正前 (実画像) . . . . .	58
5.10 3次元情報による補正後 (実画像) . . . . .	58

# 表 目 次

# 第 1 章

## 序論

## 1.1 本研究の背景と目的

近年、映像技術の発展に伴って、高品質な映像をごく一般的に見ることができるようになりつつある。そのため、仮想空間を構築する際に要求される映像の質や技術は非常に高いものになってきており、現在非常に盛んに研究がなされている。なかでもコンピュータグラフィックス (CG) は、現実感の高い仮想空間を視覚的に生成するための最も一般的な手法であり、従来から幅広い分野で研究がなされてきたが、さらに近年では、バーチャルリアリティや GIS, ITS などの最近注目されている技術のベースとして期待され、また計算機の速度や記憶容量の増大により計算機の性能の大幅な向上という計算機側の技術的理由も加わり、このような仮想空間を生成するコンピュータグラフィックスの研究は、コンピュータビジョンの分野においても、現在増々盛んに研究が行われている。

このように盛んに研究が行われた結果、現在では計算機内において大規模かつ精密な三次元幾何形状モデルを扱えるようになり、相当のリアリティを持つ精細な映像を生成できるようになってきている。しかしながらこのような従来のジオメトリベースによるレンダリングでは、非常に高性能なレンダリングエンジンを使用しても、体験者の行動に応じてリアリティの高い画像をリアルタイムに提示することは困難であり、ジオメトリ・ベースのレンダリングにおいてそのリアリティーさにおいて限界があることが認識されてきた。

一方で、そのようなジオメトリ・ベースにより生成された仮想空間よりもリアリティの高い画像を生成しようとした場合、実画像を用いる手法が考えられる。このようなイメージベースによる仮想空間の生成は、しかしながら画像のクオリティは高い反面、これまでの研究においては撮影者の視点と異なる視点から眺めるなどの自由度はほとんどない、もしくは少ない、という問題点があった。しかし計算機において利用できる記憶装置の容量が向上するにつれ、大量の画像を蓄積することが可能になり映像提示の手段として使用されるようになってきており、そのような計算機で映像を扱うことで、撮影された二次元実写画像を撮影された順に再生するだけでなく、それらの実写画像に様々な加工や処理を施し、撮影者の視点以外の映像を作りだすことが容易になると考えられおり、現在このようなイメージベース手法は仮想空間の生成の際の有効な手法の 1 つとされている。

以上のようなコンピュータグラフィックス (CG) の研究に関する流れの中で、特に近年、都市のような大規模な仮想空間を構築しようという試みが多くなされてきて



いる。それはそれまでなされてきた、屋内のような限られた空間の再構築にとどまらず、より広い空間を計算機の中に再構築しようとする要求が出てきたからであり、そのように空間での行動範囲をできるだけ広げようとする、必然的に用意する仮想空間は大きくならざるを得ないためである。

この都市空間全体といった大規模な仮想世界の構築に関しても、上記のようなコンピュータグラフィックス (CG) の流れがより当てはまる状況である。すなわち、これまではその大規模な仮想世界空間の生成手段として、そのほとんどが三次元モデルのレンダリングというジオメトリベース手法を用いられてきた。確かに、仮想空間の規模が比較的小さい場合には、専用のハードウェアなどを使用することで、従来の CG 手法による写真的リアリティの高い高精細な映像を生成することが可能である。しかしながら、都市のような広大な仮想空間を生成するには、膨大なデータ量が必要となり、データの処理能力やデータ生成の労力の問題から、生成しうる映像の写真的リアリティを維持しつつ、可能な視点に高い自由度を持たせることは困難であり、広域仮想空間においては自由な視点移動が特に要求されるにも関わらず、画像を撮影した地点以外からの視界構築は非常に難しいのが実際であった。

そこで本論文では、大量の実写画像を計算機内に取り込み、それをもとに写真的リアリティを持ちながら、都市空間を始めとする広大な仮想空間を構築することを考えている。計算機で映像を扱うことで、撮影された順に再生するだけでなく、それらの実写画像にさまざまな加工や処理を施し、撮影者の視点以外からの映像を生成することを容易に実現することを目的としている。本研究では、このような実写画像を用いた広域仮想空間構築を行うため、データの入力システムとして車載型撮影システムを開発し、それによって得られたデータをもとに任意の視点から、任意方向を見た映像を生成し、簡単に都市空間などの広域仮想空間を計算機内に生成する手法を提案する。さらにシミュレーション画像や実画像を用いた実験を行い、提案した手法の効果を示す。

## 1.2 本研究の構成

第 1 章は本章であり、本研究の背景と目的、本論文の構成について述べた。

第 2 章では、本研究に関連した研究の動向について述べる。

第 3 章では、本研究の主要な目的である、実世界において全方位画像を取得して、それらの画像列を用いて計算機内に広域仮想視界を生成する手法を示す。また本手

法の特徴である、その視界合成の可能範囲などについても合わせて議論を行う。最後に実際に本システムを実装し、シミュレーション画像と実世界画像を用いた実験結果を示す。

第 4 では、本手法の持ついくつかの問題点とその結果生じるひずみについて議論をおこなう。

第 5 章では、画像中に生じるひずみに対して、都市空間における幾何学的特徴を生かした 3 次元情報を導入することで、ひずみを軽減させる手法について述べる。またその 3 次元情報を得る手法についても合わせて述べる。さらに最後に、シミュレーション画像と実世界画像を用いて行った実験結果を示す。

第 6 章では、本論文で述べた研究結果をまとめるとともに、その結果について検討を行う。また、本システムの応用例についてや、今後の研究の課題等についても述べる。

## 第 2 章

### 関連研究

## 2.1 はじめに

コンピュータグラフィックス (CG) は、現実感の高い仮想空間を視覚的に生成するための最も一般的な手法であり、従来から幅広い分野で研究がなされてきた。コンピュータビジョンの分野においても、バーチャルリアリティや GIS, ITS などの近年注目されている技術のベースとして期待され、盛んに研究が行われている。本章ではまず、コンピュータビジョンにおける CG への技術の応用に関して最近注目を浴びているモデル・ベースト・レンダリングとイメージベースト・レンダリングという大きな 2 つのアプローチを中心に紹介をする。さらに本論文の基礎概念となっているイメージベーストについてその特徴を合わせて述べ、最後に、特に広域空間を生成することに関する研究事例について述べる。

## 2.2 CG による 3 次元仮想空間を生成する研究について

### 2.2.1 モデル・ベースト・レンダリング

CG のモデルベースによる仮想空間の生成においては、その生成段階を大きくモデリング段階とレンダリング段階の 2 つにわけて考えることが出来る。前者のモデリング段階では多くの場合、人手によって仮想空間内に存在する物体の 3 次元形状と、反射特性などの材質の光学特性が与えられる。ここでは物体の 3 次元形状としては表面を表現するメッシュモデルが用いられることが多い。また物体の表面の反射特性としては、ある反射モデル関数を仮定しそのパラメタを各メッシュモデルごとに定義するのが一般的である。このような反射モデル関数としては、物体表面における内部反射を近似する Lambert モデルや、ハイライトとして観察される表面反射を近似する Phong モデルなどが用いられ、それらの反射モデルのパラメタは物体の色や表面の滑らかさに関連するつやや具合などを表している。

一旦空間内のすべての物体に対して表面形状を表すメッシュモデルと反射特性などが与えられれば、レンダリング段階、すなわち 3 次元 CG によって光源方向、視線方向、法線方向の三方向の幾何学的関係と物体表面の陰影を計算することにより、そのシーンを任意の光源下において任意の視点から見た場合の画像を生成することができる。このように仮想空間を CG で生成する際に与えられたモデルを利用する手法はよく用いられてきた。

しかし仮想空間の現実性をさらに追求していくにつれて従来型の 3 次元 CG の問

題点が広く認識されてきた。その問題とは、大きく分けて2つの側面で捉えることができる。まず第1番目の問題は、いかにして実際の物体の表面形状とその反射パラメータを取得するかということである。3次元CGによる描画には、描画の対象となる物体の形をあらわすメッシュモデルとそのモデルの反射パラメータが必要不可欠であり、空間内のすべての物体が比較的単純な形状や反射特性をしているならばモデリングツールなどを利用してユーザが入力することも可能であるが、ある程度以上の複雑度を持った仮想空間全体のモデルを生成することは現実には難しい。すなわち人の手による3次元形状や反射パラメータ等の入力によって仮想空間を生成するという手法では、空間内のすべての物体についてそれらの情報を取得してモデルを生成するのは実際には難しい。

第2の問題点は、現実空間内には3次元形状を簡単に記述できない物体が数多く存在するという点である。従来の3次元CGでは、反射モデル関数が光源方向、視線方向、それに表面法線方向により記述されるため、描画される物体に関して表面法線方向が定義できなければならない。しかしながら、現実の物体の中には髪の毛や細かな花びらのように複雑な幾何形状を持つ物体が数多く存在し、これらの物体においてはその表面形状を正確に記述することが困難である。このような物体に対しては、テクスチャマッピングがある程度有効であると考えられている。このテクスチャマッピングでは、あらかじめ準備された画像、例えば実際の物体をカメラで撮影した画像を3次元物体表面上に文字通り貼り付ける。しかしながら、例えば人間の髪の毛に相当する部分をテクスチャとして頭部形状に貼り付けた場合、一見リアリティーの高い画像を生成できるように思われるが、実は視線方向や光源方向を変化させても髪の毛やつやに相当する部分が全く変化しないという問題が生じてしまう。これはテクスチャマッピングでは、視線方向や光源方向の変化に伴う反射の変化を表現できないという点に起因している。そのため、物体形状や表面反射特性が非常に複雑な物体を描画するためには、テクスチャマッピング以外の手法が必要とされる。

3次元CGにより、仮想空間を描画する際の第1番目の問題点、すなわち描画対象となる物体のモデル入力の難しさを解決するものとして、イメージベースド・モデリングと呼ばれるアプローチが近年注目されている。このIBMと呼ばれるアプローチは、簡単に言えば、複数枚の入力画像から対象となる物体やシーンの3次元幾何形状、テクスチャマッピング用画像さらに表面反射パラメータを自動的もしくは半自動で求める手法である。このようなモデルベースによるレンダリングをモデル・

ベスト・レンダリング (MBR) と呼ぶ。入力画像の種類 (カラー画像、距離画像) や推定される情報 (例えば、テクスチャ画像または反射パラメタ)、またユーザの手による手作業の程度の違いがあるが、いくつかの手法について紹介する。

### 1. Object Shape and Reflectance Modeling from Observation[9]

まず代表的な例として、三次元物体の表面形状と反射パラメタを同時に求めるという、佐藤らの研究を挙げる。この手法では、入力画像として物体を回転させながら撮影された一連のカラー画像と距離画像が用いられ、物体の表面形状と反射パラメタが推定される。まず、最初に、対象物体を回転させながら撮影された距離画像を統合することにより、物体の表面形状を表すメッシュモデルを生成する。次に、物体を回転させながら撮影した複数枚のカラー画像において観察される物体各部の色、明るさの変化から、物体表面上の各点において、視線方向と光源方向が変化するにつれ観察される色、明るさがどのように変化するかということが決定される。物体形状そして明るさの変化を用いて、物体表面上の各点における反射モデルパラメタを推定していく。この際のパラメタとしては、物体材質により決定される色を表す内部反射パラメタと、物体表面の滑らかさにより生じるハイライトとしての反射を表す表面反射パラメタがあり、この両方を求めることが可能である。このようにして、物体表面三次元形状と物体表面の内部反射パラメタと表面反射パラメタを求めることにより、対象物体の任意の光源下において任意の視点方向から観察した画像をハイライトなどで表される質感に関しても正確に合成することが示された。つまり簡単にいうと、物体を回転させながら撮影したカラー画像列と距離画像列から物体の表面形状と反射パラメタを推定することで、任意視点からのハイライト等の復元が実現できるのである。またこの手法では一連の処理が自動で行えることも特徴となっている。

### 2. Rendering Architecture from Photographs[8]

: A hybrid geometry-and image-based approach

一方、比較的少数のカラー画像のみから物体やシーンの三次元形状やその見えをインタラクティブに求めていく手法を紹介する。ユーザが対象物体を複数方向から撮影した写真中の幾何形状特徴 (点や線など) の位置を手作業で入力することで、写真測量の原理の基づき対象物体の三次元形状と物体表面のテクスチャマッピング用画像を求める手法が開発されており、それらの中に

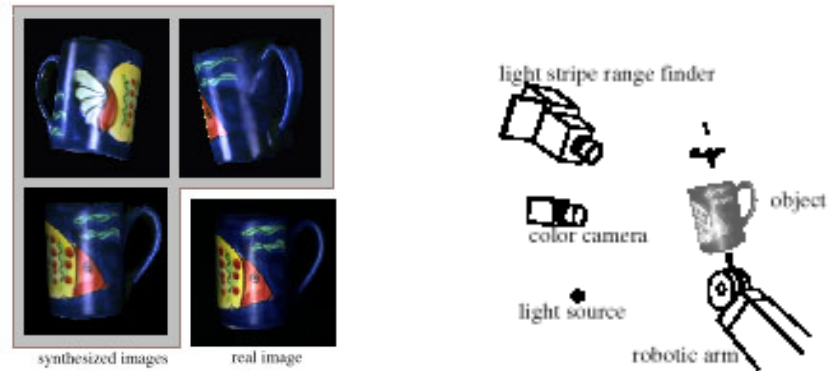


図 2.1: 佐藤らによるハイライト等の復元

は実用化されているものも多い。このようなものは一般的にフォトモデリングツールと呼ばれ、対象物体を撮影した画像から物体形状やテクスチャ画像を求めるといって、IBM 手法の一種として考えることができる。

このようなフォトモデリングツールにおいて三次元物体形状の計算は自動化されているものの、幾何形状特徴の位置の指定はすべて人手に頼るものであり、処理全体の流れは手作業とする部分が非常に多い。また、ユーザにより指定される幾何形状特徴の位置の精度がそのまま三次元形状復元の精度に影響してしまう点などからも、高精度かつ複雑な物体のモデルを生成することは難しい。

このようなフォトモデリングツールの難しさを解決するものとして、最近になり Debevec らにより発表された手法が広く注目を集めた。この手法においても、ユーザが幾何形状特徴の位置を指定する必要があるが、一般的にほとんどの人工物は対称的の形状を持っていることを利用することにより、従来のフォトモデリングツールのように点や線などの幾何形状特徴の全てを入力せずすむようになっている。また、コンピュータビジョンにおける最適化手法などにより、ユーザにより指定された幾何形状特徴の位置に多少の誤差が含まれている場合でも、三次元形状を正確に求めることが可能になっている。このように従来のフォトモデリングツールにコンピュータビジョン手法を用いることにより、Debevec らの手法では複雑な形状を持った物体に対しても半自動的にその形状を復元することができる。また、テクスチャマッピング

ング用画像に関しても、一つの物体表面に対して一つのテクスチャ用画像のみを貼り付けてしまうのではなく、その物体を見る方向に応じて、入力画像中で観察される複数のテクスチャ用画像を内挿することにより、視線方向に応じて見えが変化する様子のある程度再現することも実現されている。



図 2.2: Debevec による 3 次元形状の復元 (その 1) ムービー中の 16 枚の画像からポリゴンモデルを生成する。この例では右側の画像を生成している。また標準的なグラフィック用のハードウェアでリアルタイムでレンダリングが可能である。



図 2.3: Debevec による 3 次元形状の復元 (その 2) 先端付近の画像



### 2.2.2 イメージ・ベースト・レンダリング

また一方で、従来型の三次元 CG により仮想空間を描画する際の第2番目の問題点、すなわち物体形状や反射特性が複雑であるがためにそのモデルを生成することができない物体が多く存在するという点を克服するものとして、近年、イメージベースト・レンダリングと呼ばれるアプローチが注目されている。すなわち、先にも言及したように髪の毛や細かな木の葉などのように、現実には3次元形状や表面反射パラメータなどを正確に記述することが困難である物体が数多く存在する。そこでモデルを推定せずに画像を生成するイメージ・ベースト・レンダリング (IBR) なる手法が提案された。これは物体やシーンの見え方そのものを入力画像として記録しておき、それらを合成することで新しい画像を生成するという手法である。この手法では対象物体の形状や反射特性を考慮する必要がないので、先に指摘した髪の毛などの物体に対しても適用できる。

このような IBR のフレームワークとして Plenoptic function という概念がある。これは物体をある方向から見た場合の見えというのは、その物体が存在する空間内において物体表面から視点位置に到達する光線の束により構成されるという考えである。つまり入力画像を元に光線空間と呼ばれる光線の分布を記述する空間を定義し、物体の見え方をこの空間のパラメータとして記録しておけば、新たな視点からの画像を表示する際には記録された光線空間を参照し再構築するだけで、その視点における物体の画像を生成することができる。

しかし一方でこのような Plenoptic function に基づく手法では、原理的には対象物体をあらゆる方向から撮影した画像が必要であり、その入力画像が膨大となるため現実には不可能である。そこでこの光線の分布を効率良く記述するための入力画像のデータベース化についての研究が盛んになされることになった。その代表的な例が Light Field Rendering、Lumigraph、光線空間法、Concentric Mosaics などである。これらのイメージ・ベーストの研究例を以下に紹介する。

#### 1. Plenoptic Function[14]

空間中に存在する光線は7次元で全て表現することが出来る [14]。すなわち、視点の座標  $(x, y, z)$ 、光線の向き  $(\theta_x, \theta_y)$ 、波長  $\lambda$  および時間  $t$  である。この7次元のデータを用いることで空間中の全ての光線を記述することが可能である。そこで、ある任意視点から任意方向を向いた任意時間の画像を得ようとするれば、その方向に画像平面を想定し、その平面を横切る全ての光線を集め

てプロットすれば所望の画像が得られることになる。これは前述の全ての点から画像を撮影することと基本的には同じであるが、さらに時間の次元を加え、より一般的に分かりやすくパラメタライズしたものと考えられる。また、7次元の情報として記述することで、光線を統一的に扱うことが可能となったことから、画像を効率的にデータベース化し再構築を行なうことが出来る。

## 2. 光線空間法 [13]

このように、空間を光線で記述することで任意視点の画像を効率的に再構築する手法として、実装されたものとして光線空間法が挙げられる。この手法の特徴としては、空間を伝搬する光線は減衰したり変移したりしないことを前提とすることで、7次元の光線情報を4次元に減らし、取得する画像の枚数を実現可能な数まで減らしたことにある。ここでいう4次元とは先程の7次元から  $z$  座標、波長  $\lambda$  および時間  $t$  を差し引いたものである (図 2.4)。

実際の光線の取得方法としては、ある平面を仮定し、これを横切る光線を全て取得して表現するというものである (図 2.4)。画像を再構築する際には逆に光線を任意平面上にプロットすれば良い。

## 3. Lumigraph [17]

先に述べた光線空間法が平面  $(x, y)$  及び光線の向き  $(\theta_x, \theta_y)$  で4次元空間を表現したのに対して、ここで紹介する論文においては、この4次元を以下に述べる方法で表現した点が異なる。

まず、2つの平行した平面を仮定する。空間にある光線はそれぞれの平面を横切るため、ある任意の光線を表現するには、各平面との交点の座標をもつてすれば一意に表すことが可能である。それぞれの平面の座標が  $(u, v)$  および  $(s, t)$  であるとすれば、光線はこれらのパラメータ  $F(u, v, s, t)$  で表現することが出来る (図 2.5)。この論文ではこのような4次元空間を Lumigraph と呼んでいる。

このように表現することで三角関数を用いること無く、単純な計算で画像を合成することが可能になる。また、カメラに近い平面をカメラ面と考えるこ

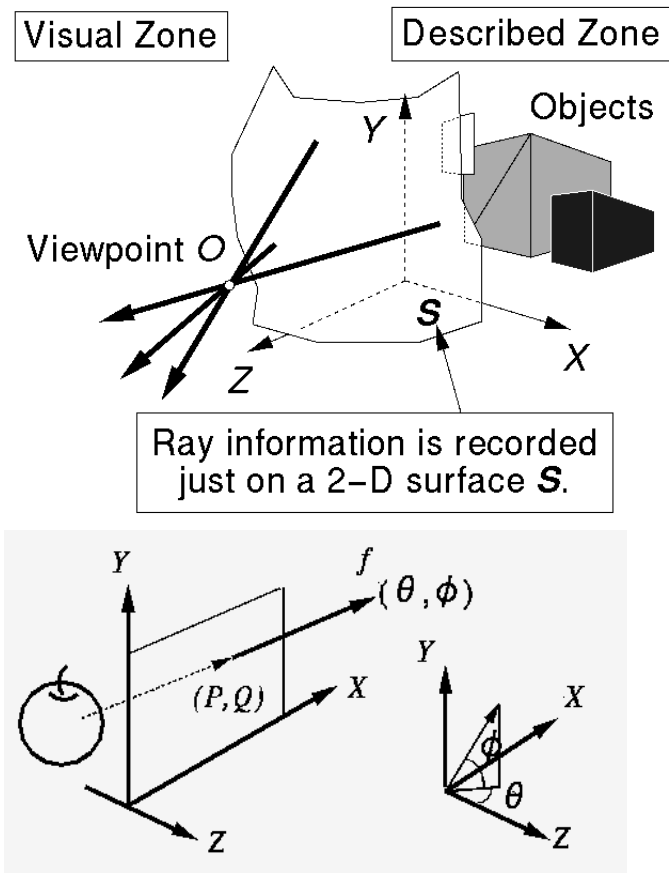


図 2.4: 4D parameterization in 光線空間

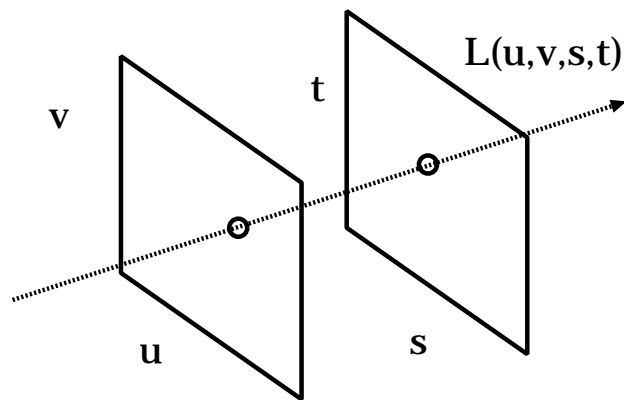


図 2.5: 4D parameterization in Lumigraph or Light Field Rendering

とで、取得したデータから Lumigraph を容易に構築することが出来る。

この論文でさらに注目される点としては、このような取得プロセスを光線空間のサンプリングとして位置付けた分析を行なっていることである。つまり、取得画像を減らしてデータを小さくすることは、粗にサンプリングしていることであり、再構築画像が実際の画像に対し劣化していくことを意味する。これを回避するために、この論文においては物体の 3 次元形状情報を利用する手法を提案している。さらに詳しい分析がのちに行なわれているがこの手法を以下に簡単に述べる。

- (a) 物体の 3 次元形状を推定する。
- (b) 新しい画像生成の際に、必要な光線がサンプリングされていない場合、通常はその光線と、 $(u, v)$  平面及び  $(s, t)$  平面との交点を求めて、必要な光線に最も近い光線を記録された光線空間中から選択する。
- (c) 物体の 3 次元形状が推定できている場合には、必要な光線と物体との交点を計算し、これに最も近い光線を選択することでより正しい補間が実現できる。

この論文では、このように物体の 3 次元形状情報を利用することで、少ないサンプル画像からでもゴーストやダブルイメージの少ないリアルな画像の再構築を実現している。

#### 4. Light Field Rendering[16]

この論文は、[17] と同じ学会で発表された、光線空間を用いた画像の再構築に関する論文である。この研究においても光線を Lumigraph 同様に  $(u, v)$ 、 $(s, t)$  平面を用いた 4 次元で光線空間を表現している。この論文では 3 次元形状を用いた効率的な画像の再構築手法については述べられていない。ただし、この論文では、「近い所を通る光線は連続的に変化することが多いため似ていることが多い」という特徴を利用した光線空間の圧縮を試みている点が注目される。

## 5. Concentric Mosaics[18]

先に挙げた研究がいずれもオリジナルな plenoptic function が 7 次元であることに對して 4 次元に次元を落とすことでレンダリングを実用的に可能にしたの對して、この研究はさらに 3 次元に次元を落としている。具体的には図 2.6 に示すように、1 つの光線を  $(x, y, \theta)$  の 3 つのパラメータで表している。しかしこの手法では、あらかじめ蓄えられる光線は、カメラを回している円周内に限られてしまうので、再構築できる視点の範囲はその円周内に制限されてしまうという欠点がある。

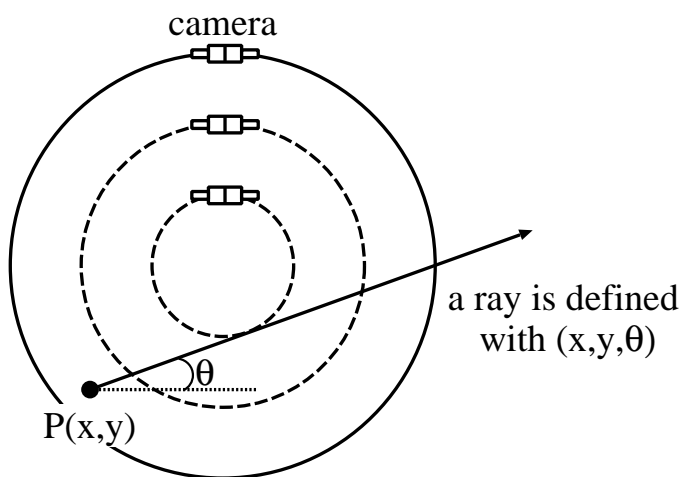


図 2.6: 3D parameterization in Concentric mosaics

以上のように、IBR では光線空間内の光線の分布を記録するための画像が必要であるので非常に大きくなる。そこで以上のように 7 次元情報を 4 次元や 3 次元に減らすことでイメージベースによる画像の再構築を実現可能とする研究が盛んに行われてきた。

## 2.2.3 イメージ・ベースト・レンダリングの欠点と改善方法

このように IBR においては、モデルを利用しないため形状や表面特性が複雑であるため MBR ではモデル化が困難な物体の場合でも、光線を十分密に記録できれば再現することができる。さらに入力画像を効率よくパラダイム化することで高速化の可能性も挙げられる。しかし一方でデータ量は、MBR は物体の形状と反射パラ

メータなどに限定されるのに対して、IBR では前述のように光線空間内の光線の分布を記録するための画像が必要であるので非常に大きくなる。また、モデルベースの場合には光源の変化などに対応して画像を合成することが出来るため、異なった環境に仮想物体を新たに置いて画像の再構築などを行なうことが可能である。これに対してイメージベースの場合には、仮想空間内のオブジェクトをインタラクティブに操作することは原理的に難しい。そこで IBR の手法に 3 次元情報等の付随的な情報を加えることで、より現実感の高い仮想空間を生成する研究やこのようなインタラクティブな操作を実現するための研究が行われている。

#### 2.2.4 データ量 (取得画像) と画像品質に関する研究

節 2.2.2 で既に述べたが、イメージベースとモデルベースとの関係はデータの大きさと表現する画像品質とトレードオフの関係にある。つまり、データのサイズ、言い換えればサンプリングする画像の枚数を減らせば、画像の品質が下がることになる。しかし、サンプリング定理からも、品質が下がらない最低のサンプリング枚数が条件次第では計算可能であると考えられる。ここではこのデータ量、つまり入力する画像の量と、生成する画像の質に関する研究を紹介する。この論文では画像の空間周波数や、実際の空間の 3 次元形状の複雑さを考慮した数学的な分析で、イメージベースの画像構築にとって最低必要なサンプリング枚数の計算を試みている。

この論文では、この最低サンプリング枚数を計算するために次の 3 つの要素を考慮している。

- ・ 空間 (モデル) の複雑さ (ジオメトリ、テクスチャ)
- ・ サンプリング枚数
- ・ 解像度 (画像の品質)

実験では、横軸をモデルの複雑さ (デプスのレイヤ数)、縦軸をサンプリング枚数として、画像の品質をほぼ同程度に保つ曲線を描き、その曲線上のいくつかの点を選び、そのパラメータを用いて実際に画像を再構築し比較している。結果の画像は、それぞれオリジナルとほぼ変わらない品質であった。

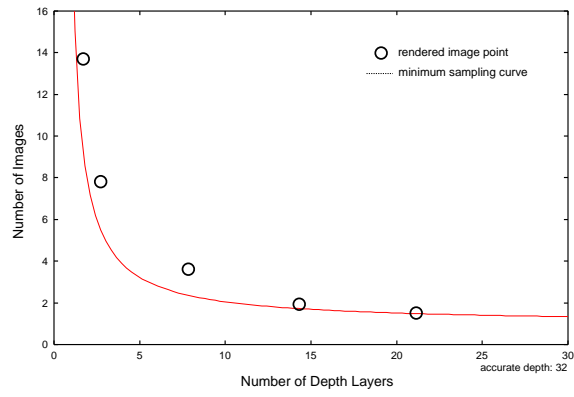


図 2.7: Minimum sampling curve for IBR

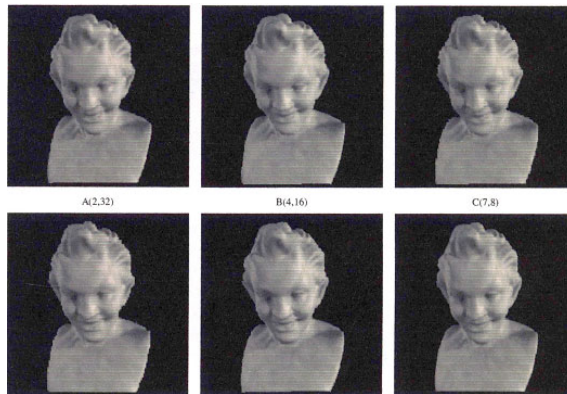


図 2.8: 図 2.7 の各点における画像生成結果

### 2.2.5 都市などの広域空間を対象とした研究

このようなコンピュータグラフィックス (CG) の研究に関する流れの中で、特に近年、都市のような大規模な仮想空間を構築しようという試みが多くなされてきている。それはそれまでなされてきた、屋内のような限られた空間の再構築にとどまらず、より広い空間を計算機の中に再構築しようとする要求が出てきたからであり、そのように空間での行動範囲をできるだけ広げようとする、必然的に用意する仮想空間は大きくならざるを得ないためである。本節ではこのような都市空間に代表される広域仮想空間の生成に関する研究例を紹介する。

#### 1. Construction of a 3d city map using epi analysis and dp matching [20]

都市などの広域仮想空間を構築するアプローチとして、節 2.2.1 で紹介した MBR の考え方に基づいた手法がある。ここではそのような考えに基づき全方位画像の利用した研究を紹介する。

この研究では、ビデオカメラを車載して都市を撮影し、その映像を用いて 3 次元情報を獲得するのだが、その獲得方法として、複数枚の画像間で対応点を正しく得ることが難しいため正確な 3 次元情報が得られにくいステレオ視や因子分解法の利用ではなく、ビデオ画像から得た非常に多量の画像を同時に処理することで安定に 3 次元情報が得られる EPI 解析を利用している。また実世界においては振動、ノイズ、光源環境の変化などの影響があり、撮影した画像列の EPI を直接解析した場合正確な 3 次元情報を得られにくい。そこで本手法ではモデルから EPI を作成し、EPI 同士を直接マッチングさせることで従来の手法よりもよりロバストな解析を行う。具体的にはデジタル地図中の建物ポリゴンデータをモデルとして、これにビデオで撮影した都市の映像とを対応付けることにより、実際の映像データを用いた 3 次元地図を自動生成するというものである。その意味でこの研究はモデルベースの 1 つであるといえる。一方、一般のカメラで撮影した映像では、画角の制限などから撮影できる範囲が限定されているため、特に建物の上方や側面方向の画像の獲得が困難であり、都市の 3 次元モデルというには不完全なものであった。そこでこの研究では、双曲回転体ミラーを用いた全方位カメラ (ここではオムニカメラを使用している) を用いることで、一度に建物全てのテクスチャ情報を獲得し、建物の側面も含めた再構築を試みている。

このようにこの研究ではオムニカメラを用いて都市の自動生成を実現するた



めの時空間画像解析を効率的に行い、オムニ画像から 3 次元情報に基づいたテクスチャ画像の獲得が可能となり 3 次元都市を半自動的に構築する手法を提案している。



図 2.9: 奥行き情報を利用した 3 次元都市

## 2. Mixed Reality[19]

この研究においても、車載型撮影システムにより、実際に都市空間において撮影をした画像を元に仮想空間の生成を試みている。この車載型撮影システムは撮影時の位置情報と姿勢情報が得られる。このようにこの手法においてはイメージ・ベースト・レンダリングの考え方に基づいているが、その画像生成に関しては光線空間などの複雑な手法は用いず、全ての経路における全周画像データをディスクに保存しておくことで、レンダリングする際には必要な画像を選択し視線方向の部分だけを再提示するだけで再構築を実現するシステムである。このため、この手法では基本的にはあらかじめ撮影した経路上しか動くことが出来ない。また撮影経路付近における視点移動に関しては、画像をある程度簡略化された何枚かの書き割りとして表現し、各面の奥行き情報をオプティカルフローなどから求めるという 3 次元レイヤー手法を用いて、画像を生成している。また各々の点で複数台 (この実験では 8 台を使用) のカメラで撮影した画像を 1 枚の全周パノラマ画像に統合し (図 3.1) 全て保存しているため、データサイズは膨大なものとなっている。これをリアルタイムにレンダリングするために実際のシステムは最初に全ての画像データをメモリに読み込むようになっているため、現実には動ける経路も制限されるなど、実用には課題も多い。



図 2.10: Mixed Reality における画像撮影用の車

## 第 3 章

### 任意視点からの画像合成

## 3.1 はじめに

この章では、どのようにして都市を始めとする広域仮想空間において「任意地点から、任意方向へ向いた」画像を生成するのかについて説明をする。本システムは節2.2.2で述べたイメージベースト・レンダリングに基づいており、まず光線空間を記述するために実画像を撮影する。その後撮影した画像列を用いて新しい視界を作成する。本章ではこのような画像の取得方法、画像の生成方法と共に、新しい視界が作成出来る範囲などの本システムのいくつかの特徴について述べる。さらに、シミュレーション画像や実世界で撮影した画像による実験を行い、その効果について検証する。

## 3.2 全方位画像の取得

本論文で提案するシステムは、節2.2.2にも示したイメージベースト・レンダリングの概念に基づいている。IBRに代表されるイメージベース手法では光線空間内の光線の分布を記録するための入力画像が必要であり、また同時にどのようにして効率のよい入力画像にするかは非常に重要である。本研究ではその入力画像として全方位画像を撮影する。ここでいう全方位画像とは画像撮影地点からのすべての光線を含んでいるという意味である。つまり画像撮影地点において、撮影した全方位画像は水平方向の360度すべての光線を持っている、という意味である。

### 3.2.1 全方位画像とは

そのような全方位画像を最も簡単に、最もシンプルに撮影する手段としてオムニカメラの利用が考えられる。このオムニカメラで撮影した画像から容易にperspectiveな画像を作ることができ、ゆえにオムニ画像から通常のパノラマ画像を作ることができる。さらにオムニカメラを使う利点を挙げることができる。すなわち図3.1-(a)の様に、オムニカメラで撮影した画像は水平方向の360度すべてを含んでいるだけでなく、そのview pointを中心とする片半球すべての方向を写すことができる。つまり画像平面よりも上側すべての光線をその画像に持っているのである。

全方位画像を撮影するもう1つの具体的な方法としては、図3.1-(b)の様に幾つかのカメラを、それらすべてのカメラの光軸が1点で交わる様に円筒状に並べることで、各々で撮影したperspectiveな画像を円筒座標系に投影することで、パノラ

マ画像を得ることができる。よってこれ以降、この研究に関して全方位画像とはこれらの方法で撮影した画像を指すこととする。

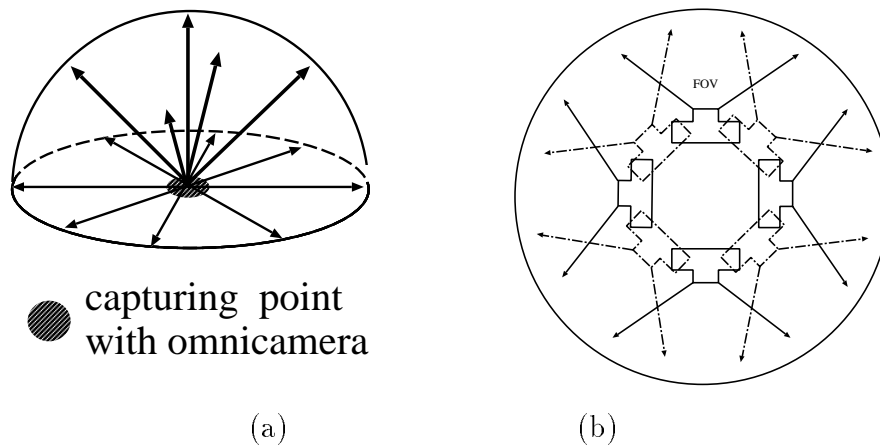


図 3.1: (a) オムニカメラで撮影した画像の持つ光線 (b) 円筒状に配置されたカメラで全方位画像を撮影

### 3.2.2 全方位画像取得のためのシステム

本システムは、都市空間に代表される広域都市空間の IBR 手法による構築を主な目的としている。そのために前もって光線を蓄えておく、つまり実際の都市空間の画像を撮影する必要がある。本システムにおいて実際の画像取得には、図 3.2 のような車載型撮影システムを使用する。この車には以下のような機能を搭載していることを想定している。

- オムニカメラなどの全方位画像を撮影するためのカメラ
- differential GPS
- ジャイロセンサー
- スピードセンサー

以上のような機能を搭載している車載型撮影システムを用いて、都市空間において全方位画像を撮影する。また光線空間法にあるように各光線を記述する必要がある

るので、車載されている D-GPS などを使って各全方位画像の撮影した地点を画像と共に記録しておく。



図 3.2: 実世界での画像撮影用の車

また上述したように、このシステムの目的の1つとして、都市を始めとする広域仮想空間の生成が挙げられる。都市空間においては、その多くに幹線道路などの幅の広い道路が走っている。さらにそのような大きな道路の場合、そのほとんどの部分は直線で構成されていることが多い。また一方で、アプリケーションの1つとしてドライビングシミュレータを考えている。ゆえに都市空間においてそのような直線道路を走行しながら全方位画像を撮影するという設定が、都市空間を生成するにあたって最も妥当性が高い設定であると考えられる。そこでこれより以降は、上記のように道路上を直進走行しながら、全方位画像を撮影し、同時にその撮影地点をGPSなどで記録しておく。そしてそれらの画像列を用いて仮想都市空間を生成することを考える。

## 3.3 仮想視点からの視界合成

### 3.3.1 はじめに (IBR)

これは物体をある方向から見た場合の見えというのは、その物体が存在する空間内において物体表面から視点位置に到達する光線の束により構成されるという考えである。つまり入力画像を元に光線空間と呼ばれる光線の分布を記述する空間を定義し、物体の見え方をこの空間のパラメータとして記録しておけば、新たな視点からの画像を表示する際には記録された光線空間を参照し再構築するだけで、その視点における物体の画像を生成することができる。

### 3.3.2 画像の合成方法

前項で得られた全方位画像列を用いて任意視点からの任意方向の画像を生成する方法を説明する。図 3.3 の様に、 $Y$  軸上を車で走りながら全方位画像を撮影し、点  $P$  からの視界 (仮想的には図の実線) をつくることを考える。光線空間の考え方によれば、点  $P$  まわりの  $R_s$  から  $R_e$  までの光線が必要となる。ここで図の  $\theta_i$  について考える。 $Y$  軸上の任意の点で全方位画像を撮影しているので点  $C1$  においても全方位画像が撮影されており、その全方位画像には点  $C1$  まわりの全方向の光線が含まれていると考えられる。すなわち点  $C1$  における全方位画像には図の  $\theta_i$  なる光線が含まれていると考えられるので、この光線に関しては図 3.3 右側にある様に、点  $C1$  で撮影した全方位画像から  $\theta_i$  に相当するスリットを切り出すことになる。結局撮影した全方位画像列から、各光線に対応するスリットを切り出し、それらを並べ合わせることで、点  $P$  からの視界を作成することができる。

## 3.4 視点合成が可能な領域

### 3.4.1 はじめに

この節では本システムの特徴の 1 つである、広域な空間を生成できることに関して議論を行う。具体的には節 3.2.2 で設定した、都市空間を生成する際に直線道路を走行しながら全方位画像を撮影するという設定のもとで、どのような範囲において新しい画像が生成できるのかについて議論を行う。

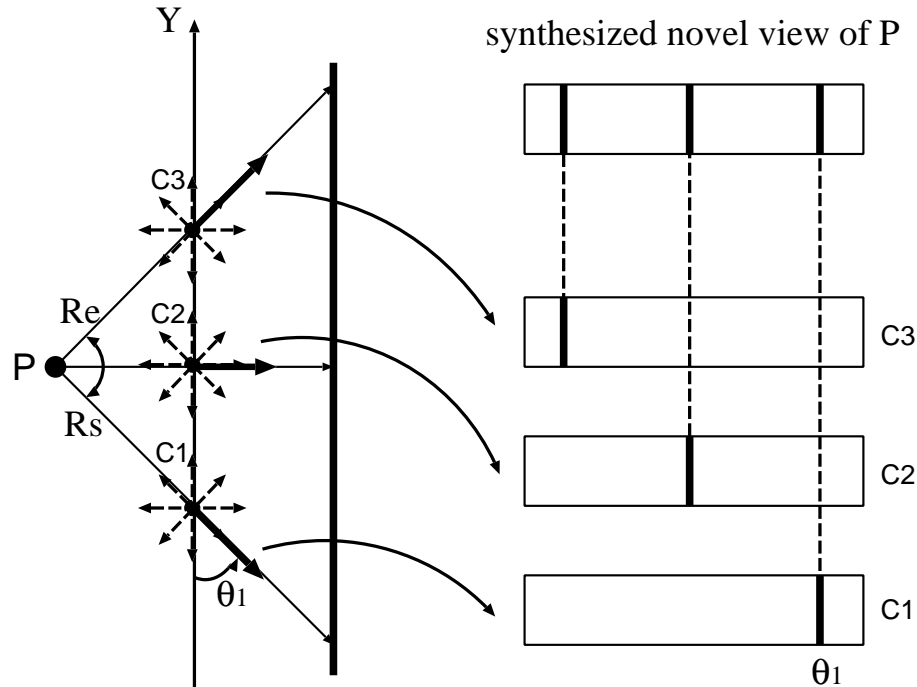


図 3.3: 画像の再構築方法

### 3.4.2 2次元上の、合成可能領域

この節では2次元上、すなわち地平面状において、どのような範囲において新しい視界を生成することができるかについて、議論をおこなう。図 3.4 において、 $xy$  平面を地面と考え、またその  $xy$  平面上においてカメラが点  $C_0$  から  $C_n$  までの間を全方位画像を撮影しながら直線上を移動するとする。ここで視角  $fov$  は一定とする。まず点  $P'$  において生成できる範囲について考察すると、例えば図中の太い波線の方は、その2本の太い波線で挟まれたすべての光線が存在するので完全な画像を構成することができる。一方細い波線の方に関しても同様に必要なすべての光線があるので、画像を構成することが出来る。ところが点  $P'$  よりも画像を撮影した走行路から離れた点  $P$  に関しては、図中の実線の方及びその反対方向しか画像を作成することが出来ない。つまり画像を撮影した走行路上及びその付近ならばどのような方向を向いた視界でも画像を生成することが出来るが、その画像撮影走行路から離れるにつれて方向の自由度は減少してゆき、視角  $FOV$  を円周角とする円周上ではただ1つの方向に関してのみ画像を生成することが出来る。このような偏角に関



しての自由度の違いはあるが、2次元上で新しい画像を生成することが出来る範囲は、図の斜線部にあるような、視角  $FOV$  を円周角とする2つの扇型の内部である。

さらに重要なことは、上の例では点  $C_0$  から点  $C_N$  までの直線上の限られた距離で全方位画像を撮影した場合について考察したが、当然その撮影距離は伸ばすことが出来る。画像を撮影した距離が伸びると、同じ点  $P$  に関して考えた場合、その偏角に関する自由度は増していく。

本手法の大きな目的の1つとして都市をはじめとする広域仮想空間の生成があり、またその実際的なアプリケーションの1つとして、ドライビングシミュレータを考えている。その場合道路上を全方位画像を生成しながら走行し、新しく生成する視界に関して要求される範囲は主に道路上からの視界、つまり画像を撮影した走行路の近くからの視界である。ゆえに画像を撮影する距離は伸ばすことが容易であり、その結果偏角に関する自由度は大きくなり、新しい画像生成に関してはほぼすべての偏角に関して画像を生成することが出来る。

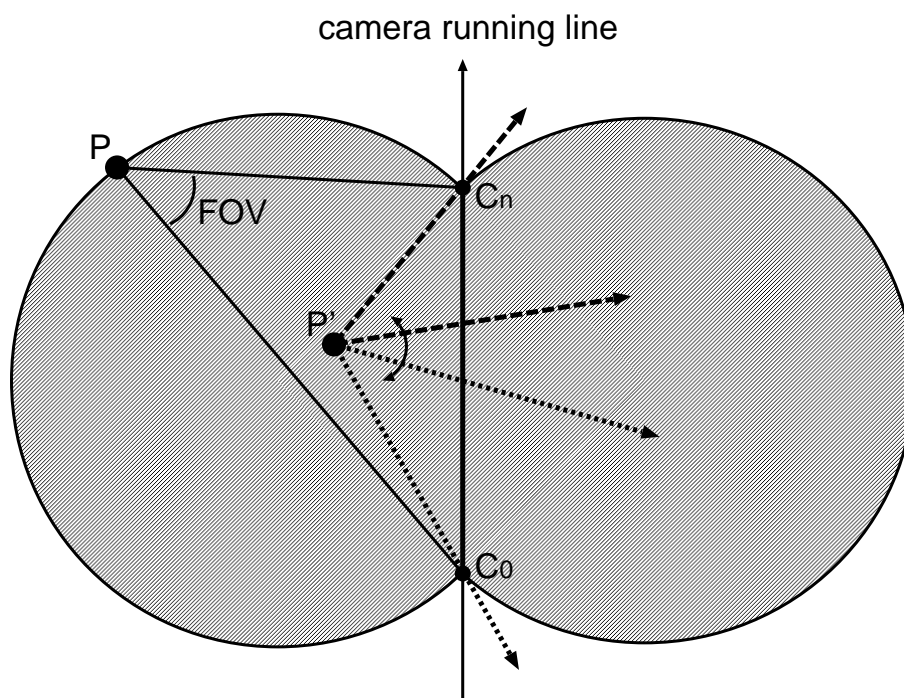


図 3.4: 2次元上の合成可能範囲

### 3.5 進行方向の光線

画像を生成する際に、そのレンダリングについて2種類に分けることができる。すなわち「singular direction」を含む場合と、含まない場合である。ここで「singular direction」とは、画像を撮影した走行路に平行な光線を指す。つまり生成する画像に全方位画像を撮影した走行路と平行な光線を含んでいる場合と含んでいない場合の2つに場合分けされる。進行方向を向いている、例えば運転者の視界の生成の場合などは前者であり、横を向いた場合の視界は後者に分類される。

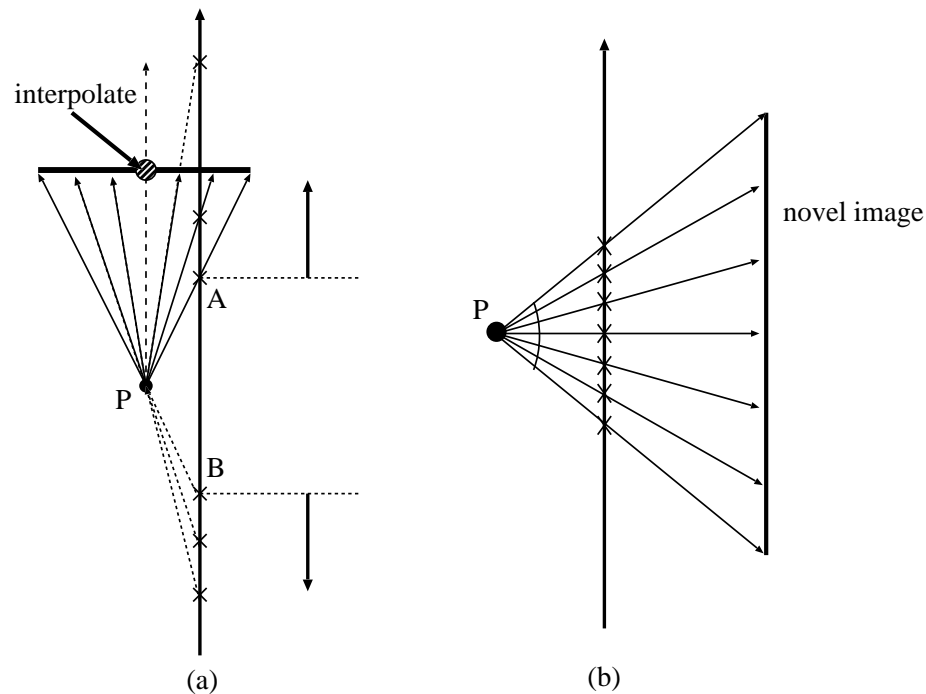


図 3.5: (a):singular direction を含む場合の画像生成 (b):singular direction を含まない場合の画像生成

まず最初に前者の場合について議論を行う。このケースでは、図 3.5-(a) にあるように、スリットを切り出してくる全方位画像について、点 A から前方の全方位画像列と点 B から後方の全方位画像列の2つの画像列から切り出してくることになる。つまりこの図 3.5-(a) では、生成する画像の右側は点 A より前方で撮影された全方位画像からのスリットで構成され、左側の画像は点 B より後方で撮影された全方位画像からのスリットで構成される。この部分の不連続性により、生成される画像にお

いて、singular direction を横切る部分では distortion が発生する場合がある。

加えて、singular direction に平行な光線は無限に全方位画像を撮影しない限り存在しないので、画像を生成する際にはこの光線を補完しなくてはならない。本論文では、その光線に最も近い光線によって代替することを行っている。その理由は、前述したように、都市空間を撮影する際は道路上を走行し、本手法の主なアプリケーションの1つであるITS分野に関しても、画像を生成するのは道路上からの視界である。よって singular direction にそっては通常物体は存在しないか、または非常に遠くに存在する。

一方、singular direction を含まない場合に関しては、生成する画像は singular direction に平行な光線を含んではいない。このケースでは、図3.5-(b)に示すように、必要な光線はすべて撮影した全方位画像に含まれている。さらに言えば、これらの画像生成に光線に用いる光線は、singular direction を含む場合と異なり、連続する画像列に含まれているので、先ほど述べたような distortion は生じない。

## 3.6 任意視界の画像合成結果

### 3.6.1 はじめに

我々はこのシステムについてインプリメントを行い、実際に画像の生成、すなわち画像を撮影した地点以外の場所からの視界を生成した。まずシミュレーションとして、CGで自分で仮想空間を作成し、その中で全方位画像を作り、それらの連続画像列を用いて新しい視界を作成した。また画像生成結果について論を行った。

### 3.6.2 シミュレーション画像による結果

まずシミュレーション用の画像を使って実験を行った。図 3.6 はそのシミュレーション用の画像の 1 枚であるが、このような全方位画像の画像列を用いて以下の画像を生成した。このシミュレーション画像は、道路の幅を  $12m$ 、画像取得頻度を  $50cm$  間隔 (時速  $54km$  で走行しながら  $30$  枚/秒で取得した場合に相当する間隔) として実験を行った。

図 3.7 は、画像を撮影した走行路に対して横切る方向に動いた場合の、横方向、つまり走行路に対して垂直な方向を向いた場合の視界である。この場合、建物の 3 次元情報はないので縦方向には伸縮はされず、その分違和感がある。図 3.8 は走行路と平行に走行した場合の、全方位画像を撮影した走行路に対して垂直な方向を向いた場合の連続画像である。具体的には撮影した走行路に対して  $3m$  建物に対して遠い方向に平行な走行路からの視界である。また図 3.9 は全方位画像を撮影した走行路上ではないある 1 点で回転した際の視界である。

これらの生成した画像は、節 3.5 で説明した singular direction を含んでいるが、新しい視点に最も近い点で撮影した全方位画像から切り出したスリットを代用することで、singular direction の光線を補完している。図に示されているように、その補完を含めて新しい視界の生成はうまくいっているといえる。その一方で生成した画像をよく見ると、スリットの繋ぎ目や補完部分において不連続性が存在している箇所がある。しかしそれらは、morphing の技術を用いることによって、その影響を減らすことが出来る。加えて、これらの画像列が実際に使用される場合、例えば本手法が想定しているアプリケーションの 1 つであるドライビングシミュレータなどにおいては、これらの画像列を動画として使用するが、その場合は上記の不連続性は、人間の目にはその影響はより小さくなる。

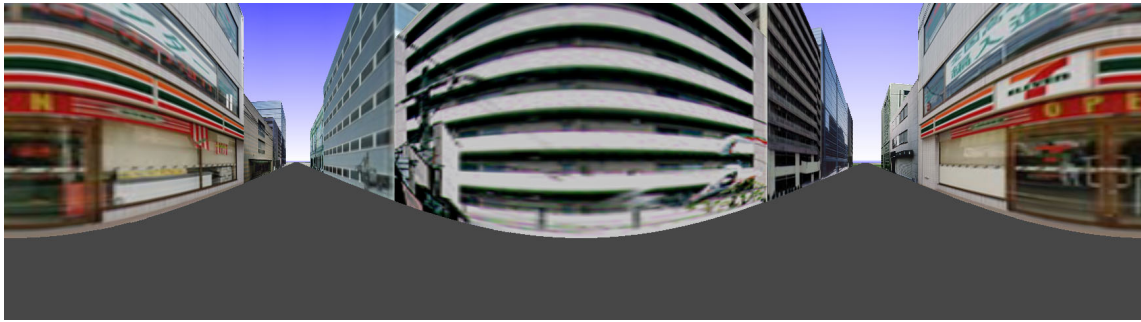


図 3.6: シミュレーション用全方位画像



図 3.7: 道路に対して垂直に移動した時の視界の作成結果





図 3.8: 道路と平行に走行した際の視界の作成結果



図 3.9: 回転運動時の視界の作成結果

### 3.6.3 実画像による結果

我々は、実世界において全方位画像を撮影し、それらを用いて画像を生成した。

まず HyperOmni カメラで撮影した、オムニ画像を使用した実験結果について述べる。図 3.10-(a) はその HyperOmni [21] カメラで我々の研究室を撮影した画像の 1 つである。カメラの特性により、オムニ画像から perspective な画像を作ることが出来る。図 3.10-(b) は図 3.10-(a) から作成した perspective な画像である。

このカメラを使用して、直線上を動きながら、連続したオムニ画像列を撮影した。図 3.11 に表されているように、点  $A$ 、 $B$ 、 $C$  はオムニ画像を撮影した直線上ではない点であり、順に黒板や本棚に近くなっているという位置関係であるが、ここではこれら 3 点から見た視界を生成した。図 3.11-(a)、Fig.3.12-(b)、Fig.3.12-(c) はそれぞれ点  $A$ 、点  $B$ 、点  $C$  から見た視界を生成した結果を表している。ここで注目すべき点は、黒板の左端と、その黒板の後ろにある本棚の右端の関係が、これらの生成した画像間で異なっている点である。もっと詳しく述べるならば、点  $A$  から見た視界を表している図 3.12-(a) では、本棚の右端と黒板の左端は全く重なっていない。一方、点  $A$  よりもそれらの物体に近い点である点  $B$  からの視界である図 3.12-(b) では、黒板の左端は本棚の右端とちょうど重なって見えている。また、点  $B$  よりさらに物体に近い点である点  $C$  からの視界を表している図 Fig.3.12-(c) においては、本棚の右端は完全に黒板によって隠されている。以上のことから、これらの生成された 3 枚の画像は、視点の位置によって変化する occlusion を正確にあらわしており、またその生成された画像は、全方位画像を撮影していない点からの視界である、ということがいえる。

次に、屋外で撮影した全方位画像を用いた実験結果を示す。この実験では、図 2.10 の車載型全方位画像撮影システムを用いて、実際に横浜市中を走行しながらで全方位画像を撮影した。なおこの際カメラは SONY 社製の DCR-VX1000 を用いた。水平画角は約 60 度である。このカメラ 7 台を節 3.2 の図 3.1-(b) で示したように、車上にできるだけすべてのカメラの光軸が 1 点で交わるように円筒状に配置し、それらのカメラで撮影した画像を円筒座標系に投影して作成した全方位画像を使用した。図 3.13 はそのようにして作成された全方位画像の 1 つである。そのような設備を持った車で横浜市街の公道を走行し、秒間 30 フレームで画像を撮影した。また節 3.3 にあるように、画像を撮影しながら、GPS などによって画像の撮影した地点を同時に記録しておく。そのようにして撮影した画像を円筒座標系に投影し、連続全方位画像列を得る。使用した GPS は Trimble 社の 7400 型であり、その精度は  $\pm$

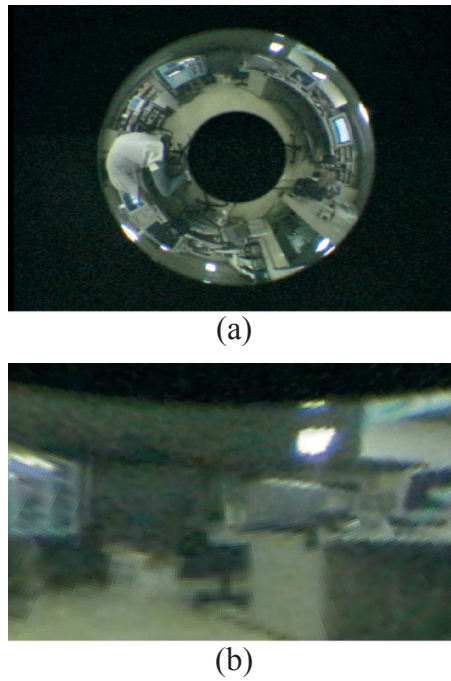


図 3.10: (a): HyperOmni camera で撮影したオムニ画像  
(a) から作成した perspective 画像

(b):

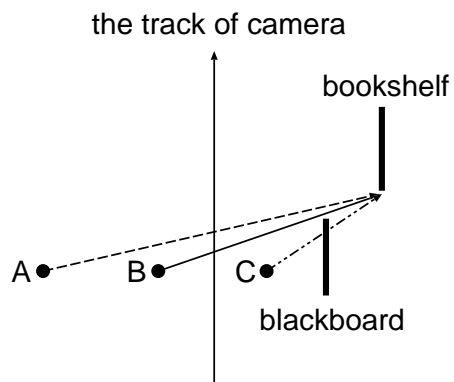


図 3.11: 屋内で撮影したオムニ画像での実験の位置関係図; 図 3.12 に示した生成図は、それぞれ点 (a)、(b)、(c) から見た際の図である。

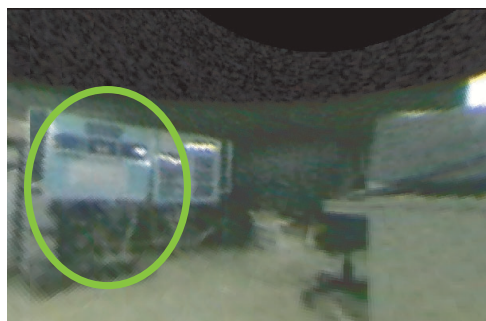




(a)



(b)



(c)

図 3.12: 生成した画像; 画像 (a) では黒板の左端と本棚の右端は完全に分かれているが、画像 (b) では、それらはちょうど重なっている。さらに画像 (c) では、本棚の右端が黒板によって隠されている。

3cm (カタログ値) である。またサンプリングレートは 5Hz である。



図 3.13: 横浜市で撮影した画像を用いて作成した全方位画像

新しく作成した画像について、その位置関係を図 3.14 に示す。注意すべきは、これらの作成した画像について、その視点の位置は、すべて全方位画像を撮影した地点以外の場所であることである。すなわちこれから示すすべての生成した画像は、まったく新しい視点からの画像である。図 3.14 において、直線  $Y_1$  上を画像を撮影しながら走行し、以下の画像を生成した。まず図 3.15 は直線  $X$  上から見た視界を生成した。これは仮想走行路 (1) 上からの視界である。また図 3.16 は直線  $Y_2$  上から見た視界を生成した。同様にこれは仮想走行路 (2) 上を走行した場合の視界であるといえる。すなわちこれらの画像列はいずれも仮想走行路からの視界を再現したものである。

図 3.16-(a) において、中央に存在する白い円形の建築物のエッジが滑らかにつながっている一方、地面にうつっているボールの影は、階段状になっている。このひずみはに関しては 4 章で詳細に説明する。

さらに、スリットの繋ぎ目のところに、不連続な箇所が見受けられるが、このひずみは全方位画像のサンプリング周期に起因している。当然スリットの幅が十分に狭くなれば、この不連続な箇所は消える。言い換えれば、十分に連続して全方位画像を撮影すればよい。加えて、仮想空間中のオブジェクトが視点位置から十分遠方であれば、先ほど指摘したひずみと同様に、この種の不連続性もその影響が小さくなる。ここで街などでは、建築物は通常道路から十分遠方にあるため、その結果これらの不連続性の影響は小さくなる。これらの実験は、街などの広域仮想空間を生成する際には、このシステムがとても十分 effective であることを示している。

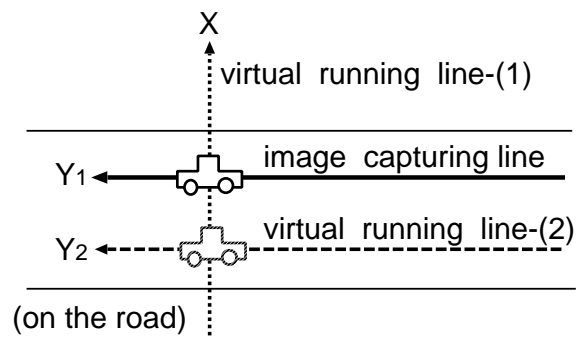


図 3.14: 画像作成の位置関係: 方位画像を撮影しながら道路上を走行したが、それを  $Y_1$  軸とする。図.3.15 は仮想走行路-(1) からの眺めであり、すなわち  $X$  軸からの眺めである。図.3.16 は仮想走行路-(2) からの眺めであり、これは  $Y_2$  軸状からの眺めである。

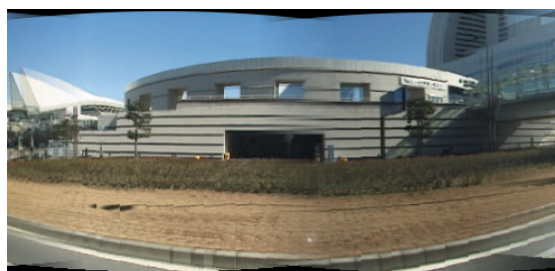


(a)

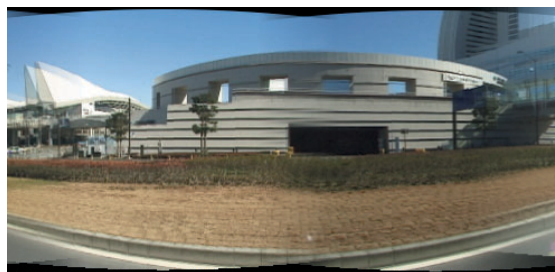


(b)

図 3.15: 仮想走行路-(1) からの眺め



(a)



(b)



(c)



(d)

図 3.16: 仮想走行路-(2) からの眺め。これらは実際に画像を撮影した走行路以外の別の走行路であるといえる。

## 第 4 章

### 生成された画像における問題点の検討

## 4.1 はじめに

節 2.2.2 で説明したように、Plenoptic function に代表される IBR 手法のみで画像を生成することは、その光線空間を記述するためのデータ量が十分であれば、すなわち記録してある光線の量が十分ならば、単にその蓄えられているデータから該当する光線をとってくるだけなので非常に簡単である。しかし実際にはオリジナルな Plenoptic function が示す 7 次元のすべての光線を前もって記録しておくことは不可能であり、実際に画像を生成する場合は 4 次元、3 次元と次元を落とすことでレンダリングを可能にしている。このように Plenoptic function による仮想空間の生成は、次元を落とすことによるひずみ、すなわち光線データの欠落によるひずみが生じる。また一方で、3 次元情報を全く使わずに仮想空間を生成した場合は、画像中に違和感が生じることがある。このひずみは IBR によるレンダリングの際は必ず生じる問題であり、IBR に基づいている本手法においても内包している問題である。この章では以上に挙げた 2 種類のひずみのそれぞれについて、本手法に基づいた検討を行う。

## 4.2 横方向のひずみ

この節ではオリジナルの Plenoptic function から次元を下げたことにより生じるひずみについて述べる。その説明のための図を図 4.1 に示す。この図において、点  $P$  が新しい視点で、太線が生成される画像平面であるとする。図中のオブジェクト  $Q_1$  はその位置関係から点  $P$  からの視界に写っているべきものであるが、レンダリングの際に使用する全方位画像からの切り出されたスリットには写っていないので、生成される画像にも写ってはいない。これは光線空間を記述するための光線を少なくした、すなわち、plenoptic function の次元を落としたからに他ならない。また、図の直線  $Y$  上において全方位画像を撮影するのであるが、理想的には直線上のすべての点で撮影することが望ましいが、当然それは不可能であり、実際はある間隔において撮影することになる。ここでは図のように点  $A_0$ 、 $B_0$  で撮影された全方位画像を使用する状況について考える。今述べたように直線  $Y$  上の任意の点について全方位画像がないので、1 つの全方位画像から取り出す光線 (実際はスリット) は、ある幅をもつことになる。すなわち図の  $\theta_A$ 、 $\theta_B$  である。そこで、図のオブジェクト  $Q_2$  について考える。この物体  $Q_2$  は、本来ならば当然生成される画像に写っていなけれ

ばならない物体であるが、スリット $\theta_A$ 、スリット $\theta_B$ には含まれていないので、合成される画像には写らない。このひずみは3次元情報の欠落によるひずみと考えるよりも、むしろ全方位画像の取得頻度に関係していると考えられる。すなわち直線 $Y$ 上における全方位画像の取得頻度が上げれば上げるほど、この種の横方向のひずみは軽減されると考えられる。しかし実は取得頻度をあげることはデータ量の増大を意味し、節 2.2.4 で述べたように光線空間を記述するためのデータ量と生成される画像の質との、トレード・オフの関係にかんする問題となる。しかしながら、このシステムにおいては、想定している応用範囲を考慮すると、画像の精度はそれほど要求されていないので、生成された画像から考えると、節 3.6.3 で述べた取得頻度で十分であると考えられる。いずれにしてもこのひずみは、Plenoptic function の次元を落としたゆえのひずみではないが、光線が欠落していることが原因であるひずみである。以上のような光線の欠落によるひずみは、生成される画像中において主に横方向に生じるので、本論文ではこのようなひずみを、横方向のひずみと呼ぶことにする。

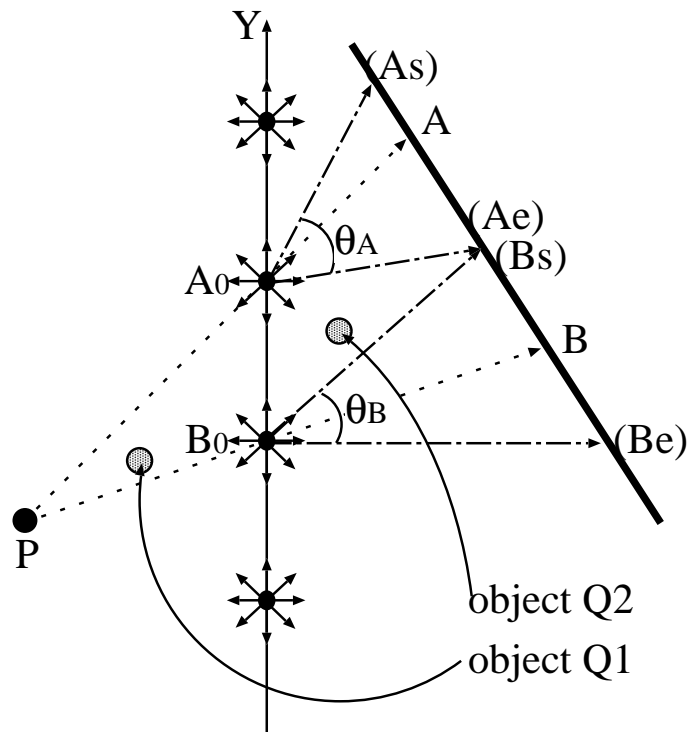


図 4.1: 横方向のひずみ

### 4.3 縦方向のひずみ

この節では縦方向のひずみと呼ばれるひずみに関して説明をする。図 4.2 にその説明のための図を示す。直線  $Y$  上を全方位画像を取得しながら走行し、点  $P$  からの視界を生成することを考える。これまで説明した様に、まず上図を用いて説明をする。上図において、点  $P$  から点  $A_0$  の方向の光線は全方位画像中の光線  $PA_0$  で代用される。同様に点  $P$  から点  $B_0$  の方向の光線は全方位画像中の光線  $PB_0$  で代用される。ここで通常道路から撮影した全方位画像を考えると、画像中のオブジェクトは、ほとんどがビルなどの建造物である。そこで点  $P$  から点  $A_0$  の方向の光線には点  $A$  が写っていると考えると、新しい視点からの光線  $PA$  を全方位画像中の光線  $A_0A$  で代用することになる。これを下図で説明すると、同様に光線  $PA$  を全方位画像中の光線  $A_0A$  で代用するのだが、この下図からもわかるように縦方向の尺度が異なるためひずみが起き、よって伸縮させる必要がある。よってつまり仮想視点と物体までの距離  $PA$  と、その光線を取得した、つまり全方位画像を撮影した地点と物体までの距離  $PA_0$  との相違によって、縦方向の尺度が違うためにおこるひずみである。このひずみを縦方向のひずみと呼ぶことにする。

この種のひずみは、節 2.2.2 で述べた concentric mosaics[18] などにも見られるひずみである。前説で述べた横方向のひずみが、3次元情報が得ることがなくても画像を十分頻繁に取得することができれば、そのひずみを軽減することができた。しかしこの縦方向のひずみに関しては、実世界における距離  $D$  を得ること、またはその近似値を得ることでは、ひずみを軽減することができない。

図??からもわかるように、このような通常の道路からの画像の場合では、この種のひずみは3.5で述べた *singular direction* に近い方向の光線であればあるほど、そのひずみが大きくなる。しかしその *singular direction* の方向は撮影の進行方向であり、通常静的オブジェクトは存在しない。例えばこのシステムの主な応用例である、ITS 分野におけるドライビング・シミュレータを考えれば、*singular direction* 方向である全方位画像の撮影車の前方向には静的オブジェクトは何もなく、動的オブジェクトの車があるだけである場合がほとんどであり、実際の生成画像ではこの種のひずみはそれほど大きなひずみとして表れないと思われる。



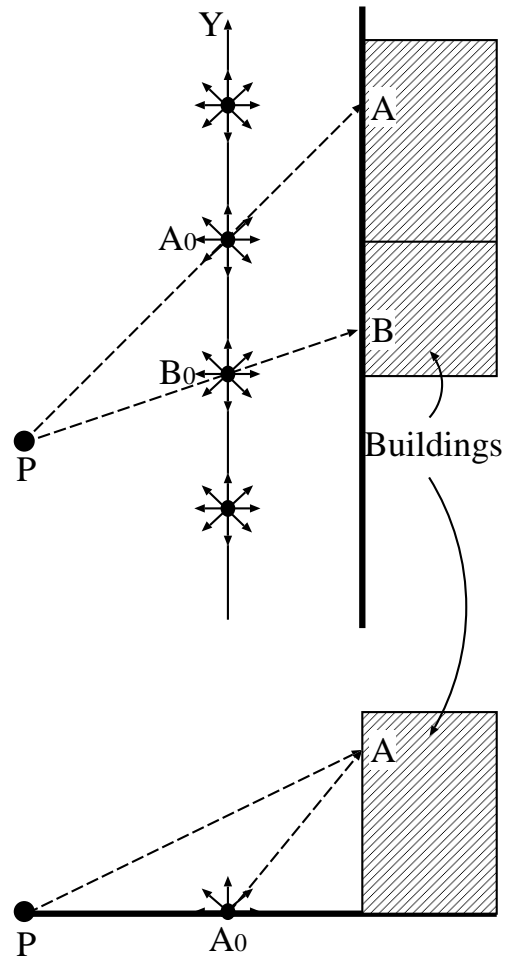


図 4.2: 縦方向のひずみ

## 第 5 章

### 都市空間の特徴を利用した画像生成

## 5.1 はじめに

第 3 章で述べたように、本システムでは、イメージ・ベースト・レンダリングに分類される手法、特に *Plenopticfunction* という概念に基づいて画像を生成しているが、他の実用的な *Plenopticfunction* と同様に、オリジナルな *Plenopticfunction* に対して次元を下げることによって、実際に画像を生成することを可能としているとすることができる。しかし一方でそのように光線空間を記述するためのデータ量を押さえているため、第 4 章で述べたようなひずみが生じる。

そこで節 2.2.3 で述べたように、これらのひずみを軽減するために 3 次元情報を付加することを考える。具体的には、本システムは特に広域仮想空間を生成することを主な目的としているが、その最も適当な対象として、都市や街などがあげられる。この章では上記のような蓄える光線のデータ量を押さえることによって生じたひずみに対して、都市空間に顕著に見られる特徴を生かして得られる簡単な 3 次元情報を用いて、そのひずみを軽減する手法について述べる。加えてその 3 次元情報の取得方法や、その軽減されたひずみの実験結果などを示す。

## 5.2 都市の建物と、それを応用した設定

一般的にビルなどの建造物は直方体をしていることがほとんどである。それは特に都市において主要道路に面している、大型の建造物に関してはよく当てはまる事実である。加えてそのようなある程度幅の広い道路に面しているビルは、多くの場合その道路に面している壁は、隣のビルと並んでいることが多い。つまりこのようなビル群に関しては以下のような特徴があると考えられる。

1. 建造物は多くの場合直方体に近い形であり、道路に面している面は平面に近い形状をしていることが多い。
2. 特に幹線道路などの広い道路に面してある建造物は、その道路に面している面に関して、隣接する建造物のそれと同じ平面上にあることが多い。

都市の建造物の多くに見られる、これらの特徴から、イメージ・ベースト・レンダリングによる都市の仮想空間の構築の際に、節 2.2.3 で述べた付加すべき 3 次元情報について、以下のような設定をすることが出来る。

設定 道路に面している多くの建築物に対して、それらの道路に面する面は、1つの平面に乗っているという近似をすることができる。(5.1 参照)

都市の広域仮想空間の生成の際にこのような設定をすることが出来ると考えられるので、この設定をもとにした3次元情報を導入し、4章で述べたようなイメージ・ベースト・レンダリングによる仮想空間の構築の際の問題点を解決することを考える。

## 5.3 デプスの推定方法

### 5.3.1 はじめに

この節では、前節で述べた設定 5.2 における3次元情報をどのように獲得するかについて述べる。前節で述べた3次元情報とは、具体的には以下の値である。

全方位画像を撮影した地点から、ビルなどの建築物の道路に面している面までの距離

都市の仮想空間の生成のための、全方位画像の撮影においては、基本的には道路を直進しながら撮影するのが最も妥当な撮影方法であることは、節 3.2.2 でも述べた。よって上記の3次元情報の具体的な値は、図 5.1 の距離  $D$  である。これ以降はこの値を「デプス」と呼ぶことにする。

この節ではこのデプスの値を以下に獲得するかについて述べる。最初に画像処理を用いたオムニ画像から取得する方法を述べ、その後それ以外の、レンジセンサーや電子地図を用いた方法などについて述べる。

### 5.3.2 オムニ画像からの自動抽出

節 3.2 でも述べたように、オムニ画像には、半球面のすべての光線の情報がその画像の中に含まれている。これは道路上で撮影した場合を考えると、画像を撮影した点から地上のすべての方向へ伸びる光線の情報を持っていることに等しい。すなわち道路に面しているビルなどの建築物がすべて写っている。このようなオムニ画像の特徴を生かして、デプスの値を簡単に、しかも精度よく獲得する方法を以下に述べる。

図は都市空間を道路上から撮影したオムニ画像である。今述べたように道路に面している建築物がすべて画像中に写っている。このオムニ画像においてカルロート

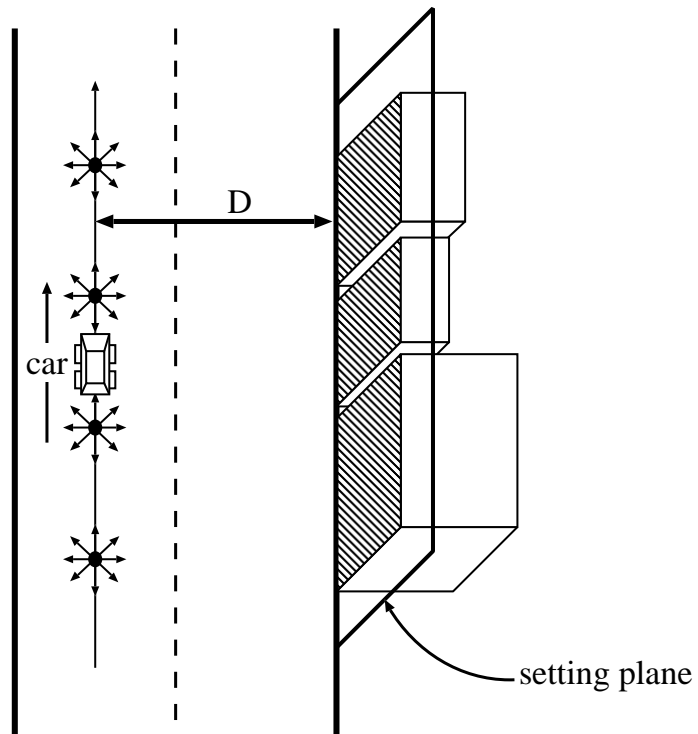


図 5.1: 都市空間におけるデプス値の設定

マシの方法で画像中の特徴点抽出を行ったの結果が図 5.2 である。ビルの角や窓の枠の部分などが抽出されているのが分かる。

ここでオムニ画像の撮影に用いたオムニカメラの特徴を述べる [21]。

まずオムニカメラに使用しているオムニ鏡の形状は放物線面と言われていて、次のような式で表される。

$$z = \frac{h^2 - (x^2 + y^2)}{2h} \quad (5.1)$$

ここで式 5.1 で表した座標系の  $xy$  平面に対して、次のような  $xz$  平面と垂直な直線  $l$  を考える。

$$x = x_0, \quad z = z_0 \quad (5.2)$$

この直線  $l$  と原点  $O(0, 0, 0)$  とを通る平面  $M$  は次の様に表される。

$$z = kx, \quad \left(k = \frac{z_0}{x_0}\right) \quad (5.3)$$

この平面  $M$  とオムニ鏡の式 5.1 の光線を  $xy$  平面に投影したものは次のような式になる。

$$(x + hk)^2 + y^2 = h^2(1 + k) \quad (5.4)$$

これは中心が  $(-kh, 0)$  で半径が  $r = h\sqrt{1+k}$  であるような円の方程式を表しており、実際のオムニ画像に写るのはこの円の弧である。またこの円は  $k$  の値に関わらず常にオムニ画像上の点  $(0, h), (0, -h)$  の 2 点を通っている。これらの 2 点を消失点と呼ぶ。以上のことからオムニ画像中の特徴点の動きについて以下のような性質があることがわかる。

**特徴** オムニカメラ（を乗せている車）が直線運動をしている間については、そのオムニカメラで撮影された連続画像列において、特徴点の軌跡は円の弧になる（図 5.3 参照）。またすべての特徴点において、その軌跡は必ずある 2 点を通るが、その 2 点を消失点と呼ぶ。

つまり例えばあるビルの角は、オム二画像中である円上を動く。また特徴点の軌跡が円上を動き、複数点の特徴点についての軌跡の円が 2 点で交わることで直線状にオム二画像を撮影したことを確認することができる。

次にこれらの特徴点を用いてデプスの値を推定する方法を示す。撮影した連続オム二画像中からある特定の 1 つの特徴点について、そのオム二画像中での動きを自動的に追跡することが出来れば、各オム二画像の撮影した地点が分かっているので、三角測量の手法を用いることでその特徴点の 3 次元位置を測定することができる。このような特徴点追跡による手法は全く人の手を使っていないので、もし可能であればこの手法が最も簡単に全自動にデプス値を測定する手法であると言える。しかし一般に特徴点のトレースは非常に難しく、特にこのオム二画像のような解像度の低い画像に関しては、実際には不可能である。よってここでは次のような手法によってデプスを推定することにする。

まず撮影したオム二画像列に対して特徴点抽出を行い、抽出したすべての特徴点群を図 5.3 のように 1 枚のグラフにプロットする。この図に円弧をフィッティングすることにより 2 つの消失点を得る。ここでビルの角などの 1 つの特徴点を人が決める。その特徴点に関して、異なるオム二画像中に写っているその特徴点を人が選びだし、その特徴点のオム二画像中の座標を記録しておく。ただしその際カメラが直進運動をしていることを確認するために、消失点 2 点と選んだ座標が同じ円上にあることを確認する。このようにして選んだ 2 点から、三角測量の要領でデプス値を算出する。

### 5.3.3 その他のデプス値を推定する方法

前節ではオム二画像において画像処理などを用いてデプスの値を求めたが、この節ではそれ以外の方法について述べる。まずレンジセンサーを用いる方法について述べる。一般的にレンジセンサーは赤外線を対象物に照射し、その跳ね返る時間によってその対象物までの距離を計測する器具である。短時間で精度よくデプスの値を計測することができる。加えて、この場合は設定 5.2 より、デプスの値は 1 つであるので、計測は全方位画像の撮影とはオフラインで、しかも 1 回での計測で済むという利点がある。

また、電子地図を用いる方法が挙げられる。最近では電子化によるペーパーレス化が各分野で進められているが、現在地図に関してもその多くが電子化されていて、比較的精度の高いものが手軽に利用できるようになっている。この電子地図を用い

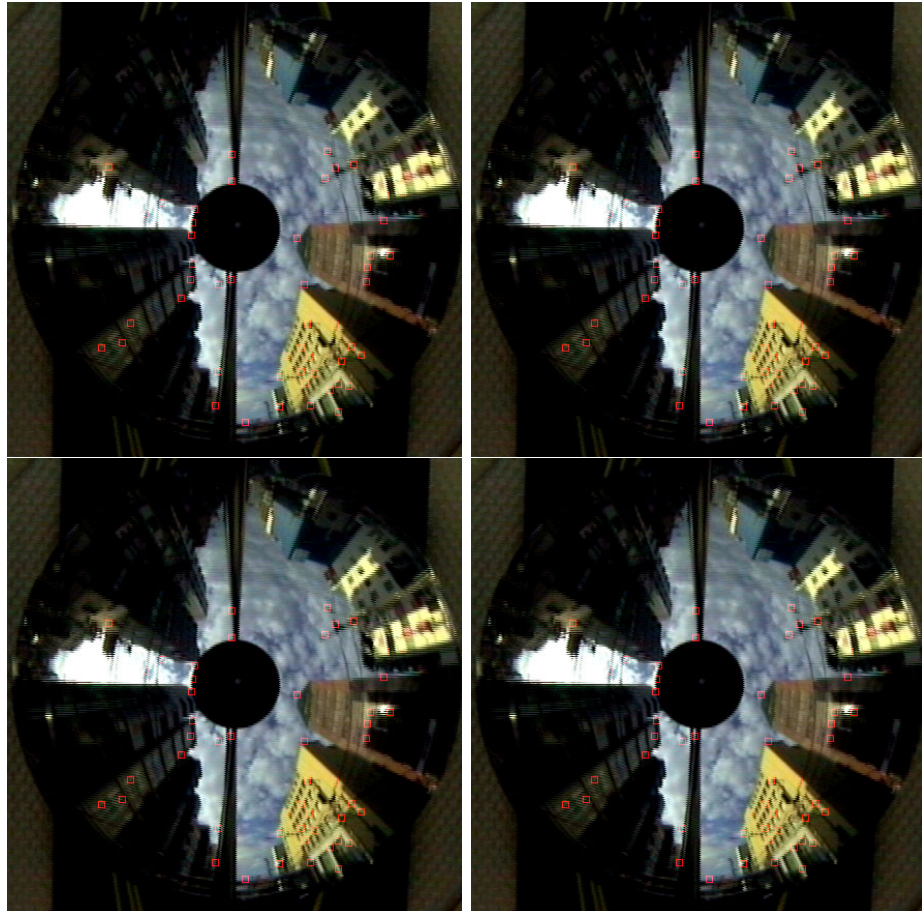


図 5.2: オムニ画像において特徴点抽出をおこなった結果



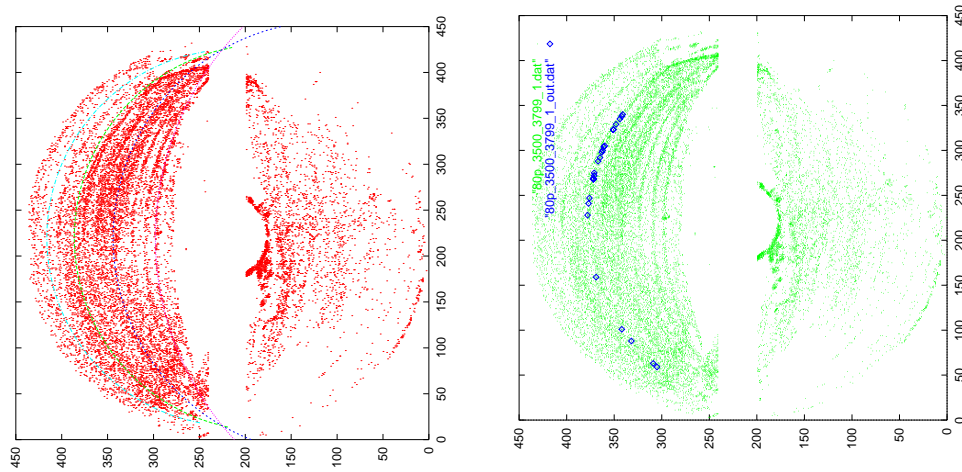


図 5.3: 連続オムニ画像における特徴点抽出と円の fitting

れば、道路ごとに、デプスの値を手軽に、精度よく得ることができる。

他の方法としては以下のような方法が挙げられる。一般に道路の幅というのは道路の種類によって決まっている。よって、走行した道路の車線数は撮影した全方位画像を見れば容易に分かるので、全方位画像中の車線数を照らし合わせることでデプスのおおまかな値を得ることが出来る。

## 5.4 3次元情報によるひずみの補正方法

本節では、節 5.3 で得られたデプス値  $D$  によってどのように生成画像を補正するかについてを述べる。

図 5.4 を用いて説明をする。図中の panoramic image capturing line は全方位画像を撮影する走行路とする。ここでは全方位画像として図 3.6 のような円筒座標系に投影された全方位画像を考える。この条件のもとで点  $P$  から見た視界、すなわち図中の synthesize image を生成することを考える。このとき生成される画像中に全方位画像  $In$  から斜線部のスリットが含まれているとする。ここではこのスリットをど

のように補正するかについて具体的に説明する。

まず点  $I_n$  から撮影した画像を点  $P$  からの視界として用いるのでその距離に応じた補正を行う必要がある。すなわち合成される画像の横方向の各ピクセルに対して、縦方向に次のような補正を行う。

$$prop1 = InE/PE \quad (5.5)$$

その後、このままでは生成された画像は円筒座標系に投影された画像になっているので、これを通常の perspective な画像にするために、次のような補正を縦方向に行う。

$$prop2 = 1/\cos\alpha \quad (5.6)$$

ただし生成される画像と、撮影された円筒座標系に投影された全方位画像の焦点距離は等しいとする。

以上の2つの補正を行い、縦方向のひずみを行う。すなわち生成される画像の横方向の各ピクセルに対して、対応する全方位から切り出してきたスリットと推定したデプス値から、以下のような補正を縦方向に行うことで、縦方向のひずみを軽減する。

$$\begin{aligned} prop &= prop1 \times prop2 \\ &= \frac{InE/PE}{\cos\alpha} \end{aligned} \quad (5.7)$$

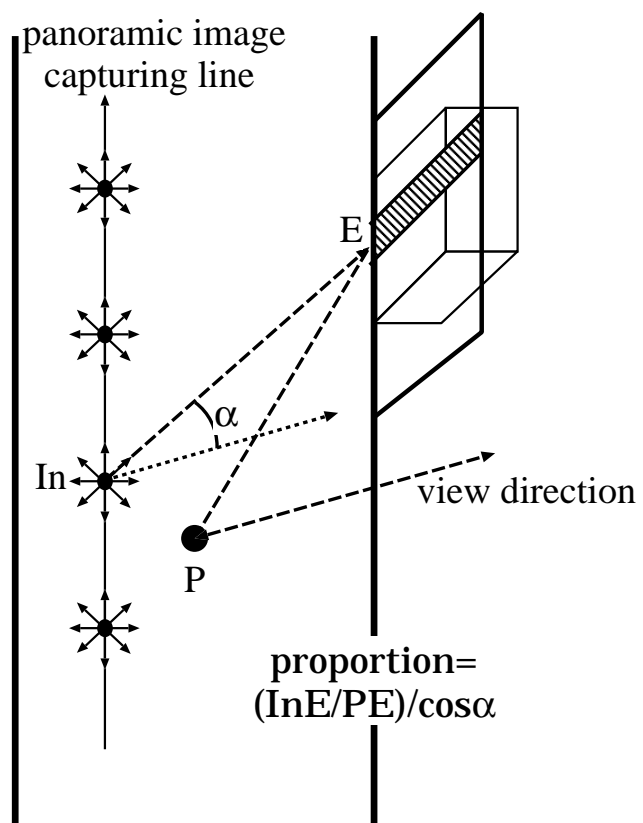


図 5.4: 3次元情報による補正方法

## 5.5 3次元情報を元に補正した結果

### 5.5.1 はじめに

前節 5.3 で述べたような方法でデプスの値を求め、その情報を 3次元情報として、先に述べたイメージ・ベストによる画像生成の結果に付加し、よりリアリティーのある画像を生成することに関して述べたが、本節では以上の補正に対して、シミュレーション画像と実際に実世界で撮影した画像の両方を用いて実験を行い、その実験結果を元に 3次元情報の付加による効果を確認する。

### 5.5.2 シミュレーション画像における、補正した結果

まずシミュレーション画像によって実験を行った。図 5.5 のように、OPEN GL によって作成したポリゴンに、デジタルカメラで撮影したコンビニエンスストアの画像をテクスチャマッピングしてシミュレーション用の画像を作成した。この様なパノラマ画像の連続画像列を用いて実験を行った。

まずこの 3次元情報による補正を行う前の画像を図 5.6 に示す。この図で明かなように、スリットの繋ぎあわせの部分において不連続性が見られる。これに対して図 5.7 は、作成時に設定をしたデプス情報によって、縦方向のひずみを補正したものである。図 5.7 の左は全方位画像を撮影した地点よりも建物に対して遠い地点からの視界を生成したものであり、図 5.6 を縦方向に縮めたものになっている。一方図 5.6 の左側は、建物に対してより近い地点からの視界を作成したものであり、縦方向に伸ばしたものである。この図においても、スリットのつなぎ目に不連続性が軽減されている、もしくは認められないことから、3次元情報による補正の効果が認められた。さらに、補正前では建物のエッジなどの本来直線であるべきところが曲線になっていたが、補正後の画像ではきちんとした直線に補正されていることがわかる。

### 5.5.3 実画像における、補正した結果

次に実画像において実験を行った結果を示す。全方位画像は節 3.2 の図 3.1-(b) のように車載型撮影システムの車の屋根の上にカメラを配置するシステムで作成した。すなわちカメラの光軸が出来るだけ 1点で交わるように配置したカメラで撮影した perspective な画像を円筒座標系に投影することで全方位画像を生成した。この際使用したカメラは SONY 社製の DXC-9000 であり、レンズは 富士写真光機社製の

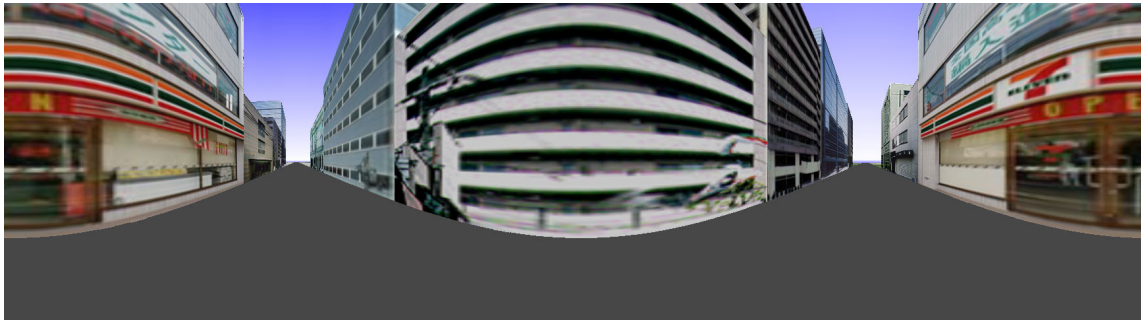


図 5.5: 実験に用いた、Open GL により作成したパノラマ画像

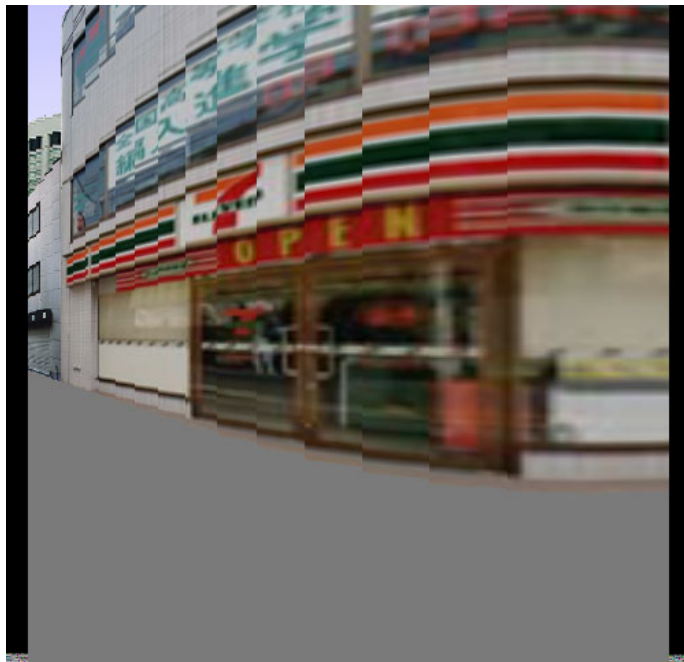


図 5.6: 3次元情報による補正前



図 5.7: 3次元情報による補正後

S12x5BRM-38 を使用した。それらのスペックは、水平画角が  $65^{\circ}14'$ 、垂直画角が  $51^{\circ}17'$ 、対角画角が  $77^{\circ}19'$  である。また Trimble 社の 7400 型 GPS を用いて画像を撮影した地点を記録しておく。またその使用した GPS の精度は  $\pm 3\text{cm}$  (カタログ値) である。以上のようなシステムを用いて秒間 30 フレームで撮影しそれらを用いて全方位画像を生成した。具体的には JR 信濃町駅付近から JR 四谷駅付近までの外苑西通りを走行し、図 5.8 のような全方位画像を生成した。



図 5.8: 信濃町付近で撮影した画像を用いて作成した全方位画像

図 5.9 は節 3.3 で述べた手法で生成した画像である。すなわち 3 次元情報を全く用いないで作成した画像である。この図では建物のエッジの線や窓のエッジ、また歩道と道路の境目の線がきれいにつながっていないことが分かる。すなわちスリットのつなぎ目において不連続性が認められる。これに対してデプス値を用いて補正をおこなったのが図 5.10 である。この図においては図 5.9 で認められたひずみが大幅に軽減されて、建物や窓のエッジ、また歩道と道路の境目の線がきれいにつながっていることがわかる。これらの図の比較から、3 次元情報による補正、つまりデプス値による補正が、縦方向のひずみに対して大きな効果があることがわかる。なお、本節の生成画像において使用した計算機のスペックは、CPU が UltraSPARC-II (400MHz) で、メモリが 1536MB である。また画像生成時間は、各画像とも約 0.5 秒から約 1 秒であった。



図 5.9: 3次元情報による補正前(実画像)



図 5.10: 3次元情報による補正後(実画像)



## 第 6 章

### 結論

この章では、これまで述べてきた本研究の手法やその実験結果などを踏まえて、本システムの特徴などをまとめながら、結論を述べる。また合わせて、本システムの想定している応用例や、今後の課題なども述べる。

## 6.1 結論

本研究において、筆者らはイメージ・ベースト・レンダリング (IBR) 手法に基づいた、新しい仮想空間の生成手法、特に都市空間などの広域仮想空間を生成するのに効率的な手法を提案し、その手法の検討や実験を行った。手法としては、光線空間を記述するための画像として、道路をただ 1 回走行しながら全方位画像を撮影するだけという、非常にシンプルな手法を提案した。またそのレンダリング手法も、光線空間の考え方に基づいて、ストックしてある画像から最も近い光線をとってくるだけ、すなわち全方位画像からスリットを切り出すだけという非常に簡単な手法である。

ここで、本システムの特徴をまとめておく。具体的には次のようなことからである。

1. イメージ・ベースト・レンダリングによる生成
2. 広範囲の仮想空間 (任意視点)
3. photo-realistic な画像
4. プロセスが非常にシンプルである
5. real time rendering の可能性

まず 1. については、イメージ・ベースト・レンダリングによる画像の生成には章 2 で述べたような様々な利点などがあり、本システムにおいてもそれらの利点はそのまま当てはまるといえ、仮想空間を生成するために非常に有効な手法であることが分かった。

次に 2. については節 3.4 で詳しく述べたように、このシステムを用いれば、原理的には地平面上の任意の地点から、任意の方向を向いた場合の視界を生成することができる。すなわち、これまでの主な IBR の各手法に比べて非常にスケールの大きな仮想空間全体を生成することができるのである。

また 3. については、節 2.2.2 でも述べたように、実際に撮影した画像を用いているので、生成する空間は非常にリアリスティックな画像になる。これはポリゴンなどによって生成した画像に比べて非常に現実感がある。

さらに 4. においては、その生成する空間が非常に広いにも関わらず、IBR のための、すなわち光線空間を記述するために撮影する画像は、ただの 1 回だけ全方位画像を撮影しながら走行するだけ、と非常に簡単である。またそのレンダリングのプロセスも非常にシンプルである。その結果、5. に示したように、real time rendering の可能性を十分秘めている。このリアルタイムで画像を生成するという事は、仮想空間を構築する際にその現実性の点から、非常に重要な点であり、特に仮想空間内を walk through することができるというような重要な利点が挙げられる。

本システムには以上のような特徴があるが、完全な IBR による仮想空間の生成では、光線空間を記述するための光線の欠如、3次元情報の欠如などの理由により、生成する画像にひずみが生じてしまうことがある。そこで本研究では、都市空間の特徴を利用したデプスなる 3次元情報を利用することを提案した。この値は非常に簡単に得ることができ、しかも都市空間の多くの場面で適用することができる。

以上のような手法を提案し、シミュレーション画像、および車載型全方位画像撮影システムを構築し撮影した実画像を用いて、以上の手法について実験を行った。その結果以下のような結論が得られた。

1. 提案した手法により、道路上の任意の地点から任意の方向の視界を違和感が非常に少なく生成することができる。
2. 都市空間の特徴を利用したジオメトリ情報であるデプス値は、ある程度精度よく得られ、その得られたデプス値を生成画像に導入することで、ひずみが非常に軽減され、違和感がない画像を生成することが出来た。
3. デプス面近くのオブジェクトや、建物の道路と垂直な面についても、非常に違和感の少ない画像を生成することができる。

以上の結果から、本システムは都市空間を始めとする広域仮想空間を生成するのに非常に有効な手法であることを示すことができた。また実際に撮影システムを構築し、非常に違和感の少ない都市空間を生成することができた。

## 6.2 応用と今後の課題

本節では本システムが想定している、あるいは実装している応用例について述べる。具体的には以下のような事例を考えている。

1. ドライビングシミュレータ
2. 3次元地図の自動更新

まず 1. については、本システムにより静的オブジェクト、つまり都市空間においてはビルなどの建築物を非常に違和感が少なくレンダリングできることが示された。またそのレンダリング過程が非常にシンプルなので将来的にはリアルタイムでのレンダリングが可能であると思われるので、ドライビングシミュレータにおける背景の生成に非常に適していると言える。また 2. については、道路をただ 1 回走行するだけで都市空間を生成することができるので、非常に簡単に 3 次元地図を生成することができる。これは現在主に使用されている 2 次元地図でさえ人の手によって大きな手間をかけて作成していることを考えると、3 次元地図を非常に簡単に生成できることは大きな利点であると考えられる。

一方今後の課題としては、動的オブジェクトのレンダリングがある。すなわち上記のドライビングシステムの際にも問題になったように、本システムでは車や歩行者などの動的オブジェクトを正確にレンダリングすることはできない。そのような動的オブジェクトに対しては以下のような手法を考えている。まず撮影した画像において背景差分などを行い全方位画像から動的オブジェクトを取り除き、静的オブジェクトに関して仮想空間を構築する。その生成した仮想空間の中に、実世界において撮影されている車などを埋め込んでいくという手法が有効であると考えている。

## 謝 辞

本研究を進めるにあたって、毎週打ち合せをして頂き、素晴らしい研究環境と適切な助言・指導をいただいた池内克史教授に心から感謝いたします。研究だけでなく、生活、人生に関しても心づかいをいただき、研究生活の支えになりました。

また研究に関して適時に適切な助言・指導をいただいた佐藤洋一助教授に心から感謝いたします。研究の内容だけでなく、研究の進め方や方向性について、大変勉強になりました。

研究を遂行する上で多くの助言や協力をして頂き、かつ、研究を遂行する上で重要な計算機の管理や保守に常に尽力して頂いた、博士課程2年の西野恒さんに、深く感謝します。

共同研究者として研究に関する様々なアドバイスをして頂き、かつ、学会の際などに大変お世話になった博士課程1年の川崎洋さんに、深く感謝します。

研究に関する助言や、研究で使用する計算機や映像機材の使用に関する様々な助言をいただき影澤影澤政隆助手に、深く感謝します。

研究を進めるに際して、様々な協力やアドバイスを頂いた、松下康之さん、高松淳さん、大石岳史さん、高橋徹さんに、深く感謝します。

学会出席やその他の事務関係についてお世話になりました、松浦芳子さん、元木恵子さん、斉木優子さん、松田真里さんに、深く感謝致します。

研究室の生活を送るにあたりお世話になった、長谷川仁則技官、倉爪亮博士協力員、佐藤いまりさん、大野一さん、小河原光一さん、佐川立昌さん、村尾真洋さん、三枝旭さん、佐藤啓宏さん、橋本謙太郎さん、吉田達哉さん、宮崎大輔さん、増田智仁さん、稲熊伸昭さん、吉藤伸幸さん、園田展人さんに感謝します。

坂内研究室や佐藤研究室の方々には研究の遂行や発表練習などに協力頂き、大変お世話になりました。ここに感謝致します。

最後に、長い間学生生活を支えてくれた家族に感謝します。

平成13年 2月 2日

## 発表文献

- [1] T. Takahashi, H. Kawasaki, K. Ikeuchi and M. Sakauchi, Rendering with panoramic images, *情処研報*, CVIM-119, pp. 33-40 (1999)
- [2] T. Takahashi, H. Kawasaki, K. Ikeuchi and M. Sakauchi, Arbitrary View Position and Direction Rendering for Large-Scale Scenes, Proceedings of Computer Vision and Pattern Recognition Conf, (2000)
- [3] T. Takahashi, H. Kawasaki, K. Ikeuchi and M. Sakauchi, Expanding Possible view point of Virtual Environment Using Panoramic Images, Proceedings of IAPR 15th International Conference on Pattern Recognition(ICPR 2000), Barcelona, Spain, Sep, 2000
- [4] T. Takahashi, H. Kawasaki, K. Ikeuchi and M. Sakauchi, Virtual Driving Sysytem with Real-World Image, 7th World Congress on Intelligent Transport Systems (2000)

## 参考文献

- [5] S. B. Kang and R. Szeliski, 3-D scene data recovery using omnidirectional multibaseline stereo, in *Proceedings of Computer Vision and Pattern Recognition Conf.*, pages 364-370, 1996
- [6] S. Peleg and M. Ben-Ezra, Stereo panorama with a single camera, in *Proceedings of Computer Vision and Pattern Recognition Conf.*, pages 395-401, 1999
- [7] Y. Yu, P. Debevec, J. Malik and T. Hawkins, Inverse Global Illumination Recovering Reflectance Models of Real Scenes from Photographs, in *Proceedings of ACM SIGGRAPH 99*, pages 215-224, 1999
- [8] P. E. Debevec, C. J. Taylor, and J. Malik, Modeling and Rendering Architecture from Photographs: A hybrid geometry-and image-based approach, in *Proceedings of ACM SIGGRAPH 96*, pages 189-198. 1996
- [9] Y. Sato, M. Wheeler, and K. Ikeuchi, Object Shape and Reflectance Modeling from Observation, in *Proceedings of ACM SIGGRAPH 97*, pages 379-387. 1997
- [10] A. Lippman, Movie-Maps. An Application of the Optical Videodisc to Computer Graphics, in *Proceedings of ACM SIGGRAPH '80*, pages 32-43, 1990
- [11] S. E. Chen, QuickTime VR - An image-based approach to virtual environment navigation, in *Proceedings of ACM SIGGRAPH'95*, pages 29-38, 1995
- [12] S. K. Nayar, Omnidirectional Video Camera, in *Proceedings of Computer Vision and Pattern Recognition '97*, page 482 - 488, 1997

- [13] T. Naemura, H. Harashima et al., Ray-Based Creation of Photo-Realistic Virtual World, *Virtual Reality and MultiMedia (VSMM'97)*, pages 59-68, 1997
- [14] E. H. Adelson and J. Bergen, The Plenoptic function and the elements of early vision, in *Computational Models of Visual Processing*, pages 3-20, MIT Press, Cambridge, MA, 1991
- [15] L. McMillan and G. Bishop, Plenoptic modeling: An image-based rendering system, in *Proceedings of ACM SIGGRAPH'95*, pages 39-46, 1995
- [16] M. Levoy and P. Hanrahan, Light field rendering, in *Proceedings of ACM SIGGRAPH '96*, pages 31-42, 1996
- [17] S. J. Gortner, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, The lumigraph, in *Proceedings of ACM SIGGRAPH*, pages 43-54, 1996
- [18] Heung-Yeung Shum and Li-Wei-He, Rendering with Concentric Mosaics, in *Proceedings of ACM SIGGRAPH*, pages 299-306, 1999.
- [19] M. Hirose and E. Takaaki, Building a Virtual World from the Real World, in *Proceedings of International Symposium on Mixed Reality*. (1999-3). 183-197
- [20] H. Kawasaki, T. Yatabe, K. Ikeuchi and M. Sakauchi, Construction of a 3d city map using epi analysis and dp matching, Asian Conference on Computer Vision 2000 (2000).
- [21] Y. Onoe, K. Yamazawa, H. Takemura, and N. Yokoya, Telepresence by real-time view-dependent image generation from omnidirectional video streams, in *Computer Vision and Image Understanding, Vol. 71, No. 2*, pages 154-165, 1998.