仮想環境を用いた視覚移動ロボットのシミュレーションの提案と 画像の記憶に基づく走行手法への適用

松本吉央^{*1} 宮崎 猛^{*2} 稲葉雅幸^{*3} 井上博允^{*3}

Proposal of Mobile Robot Simulator Using VR Technology and Its Application to View-Based Robot Navigation

Yoshio Matsumoto^{*1}, Takeshi Miyazaki^{*2}, Masayuki Inaba^{*3} and Hirochika Inoue^{*3}

Simulation is an important issue in the robotics research field since it is essential for evaluating and predicting the behavior of a robot. However previously developed simulation systems did not have the function of simulating the visual perception process of the robot. This means that a vision-based behavior cannot be simulated. In this paper, we first propose a novel approach using VR technology to simulate the vision-based behavior of a robot. Then the implemented system called "View Simulation System (VSS)", which was designed to simulate the behavior of our view-based navigation method is described. The VSS has a CG generator and a vision processing hardware, and can run the same program as one for our mobile robot. Finally the feasibility of VSS is shown through experiments using the real robot.

Key Words: Simulator, Mobile Robot, Real-time Vision, Computer Graphics

1. はじめに

ロボットの行動生成のアルゴリズムを研究する場合には、様々な条件を任意に設定した上で実験を繰り返し、その性能を定量的に評価することが重要である。しかし、実環境では多くの条件を任意に設定したり、繰り返し実験することは非常に手間がかかるため、通常はアルゴリズムの評価にはシミュレータを用いた仮想的な実験が利用される。

これまでのシミュレータでは、環境を単純化し周囲の状況が完全に分かっていると仮定した上で行動生成のシミュレーションを行っているものが多く、「センサの情報に基づく環境認識」を含めた行動生成アルゴリズム全体のシミュレーションを行えるようなシステムは少ない。しかし、実世界で行動するロボットにおいては、環境認識の結果（どのような種類の情報が得られるのか）は行動生成に大きく依存する。そのため、環境認識アルゴリズムと行動生成アルゴリズムの設計や評価は、分離してそれぞれ別個に行うことは望ましくない。特に、視覚を用いて行動するロボットにおいては、視覚処理の過程を含めた行動生成のシミュレーションが重要であると考えられるが、そのよ

うなシミュレータはこれまでにはなかった。

本論文では、このような問題点を解決するため、コンピュータグラフィックス（以下 CG）による視覚情報と画像処理システムを組み合わせた視覚移動ロボット用のシミュレータ View Simulation System の提案 [1] と、プロトタイプシステムの開発およびその利用 [2] について述べる。以下第 2 章では、仮想環境を用いた視覚を持つ移動ロボットの実時間シミュレーションのコンセプトを提案する。第 3 章では、我々の視覚移動ロボットのナビゲーション手法のために開発したプロトタイプシステムの構成について紹介し、第 4 章ではそのシステムにおける仮想世界のモデリングについて説明する。第 5 章では、開発したシステムを用いて、我々のナビゲーション手法において行動の効率的な定量評価や行動教示を行うことが可能であることを示す。

2. CG を用いた移動ロボットシミュレーション

2.1 CG を用いた画像処理のシミュレーション

CG を画像処理の対象とすることは、コンピュータビジョンの分野では古くから行われてきた [3]。これは、既知の 3D モデルの CG を生成し、ステレオ視やオプティカルフローなど画像処理アルゴリズムの評価に用いるというものである。これらはあらかじめオフラインで生成された画像を用いている。

また、Demetri らはインタラクティブな三次元 CG を画像処理に用いて、仮想世界の魚 (Artificial Life) の視覚による行動アルゴリズムの生成を行っている [4]。この研究では CG を処理し、その結果から行動をシミュレートして、仮想世界を更新

原稿受付 2000 年 10 月 25 日

^{*1}奈良先端科学技術大学院大学情報科学研究科

^{*2}株式会社キーエンス

^{*3}東京大学工学系研究科

^{*1}Nara Institute of Science and Technology

^{*2}KEYENCE Corporation

^{*3}The University of Tokyo

するというビジュアルフィードバックを実現している点が、コンピュータビジョンの分野での CG の利用法より一歩進んでいる。しかし、仮想世界内でのシミュレーションにとどまっておらず、実時間性や実世界での行動への応用に関しては特に考慮されていない。

2.2 CG を用いた移動ロボットシミュレーション

ロボットは、一般にセンサからの情報から外界を認識し、その認識結果から行動を生成し、環境に働きかけるという閉ループを持つ (Fig. 1 (1))。シミュレータは、このうちの認識と動作の部分をモデルを用いて仮想的に行うシステムと位置付けられる。ここでシステム構築の簡単化のために、センサ情報を用いた認識過程のシミュレーションを省略し、直接モデルから認識結果を生成しているシステムが多い (Fig. 1 (2))。例えば、仮想世界を二次元のグリッド状平面として定義し、自己位置や障害物情報などをその上での座標として取り扱うシミュレータ (文献 [5] など) はこれに該当する。このようなシミュレータでは、実際にロボットで得られるセンサ情報とシミュレータ上で得られるセンサ情報の種類や質が違うことがある。特に、視覚を持つロボットのシミュレーションを行う場合、このようなシミュレータと実環境の乖離が問題になる [6]。

近年人工知能のテストベッドとして注目されている、ロボットによるサッカー競技会 RoboCup のシミュレーションリーグ [7] では、モデルから認識結果を得る過程において視野やノイズを設定することで得られるセンサ情報を制限し、現実のロボットに条件を近づける工夫をしている。しかし視覚処理をシミュレーションに含めているわけではないため、実機部門のロボットのプログラムを、そのままシミュレータ上のロボットで用いることはできない。そのため、視覚処理まで含めた実ロボットのプログラムの検証を行えるシミュレータにはなっておらず、シミュレーションリーグと実機リーグはまったく別な競技として位置付けられている。

ロボットのシミュレーションにおいては、CG は単に可視化 (ユーザへの状態提示) のためにだけ利用されることが多く (例えば文献 [8] [9])、CG を画像処理の対象とするものは、これまではほとんど存在しなかった。ロボットの視点からの CG 画像を生成して学習に用いる研究 [10] もあるが、生成される画像は現実感の乏しい単純な幾何的なものであり、それに対して実際のロボットと同じ画像処理を施しているわけではない。

2.3 View Simulation System

従来、ロボットのシミュレーションにおいて画像処理の対象として CG を用いるものがなかった理由は、実環境のモデリングや高品質な CG の高速な生成が技術的に難しかったためであると考えられる。しかし、近年のコンピュータの進歩、特にバーチャルリアリティのための画像表示ハードウェアの進歩により、画像処理対象としての利用に耐え得るリアリティの高い画像を、実時間で生成することが可能となった。そのため、画像処理まで含めたロボットの実時間でのシミュレーションが現実的になってきた。

そこで、これら従来のシミュレータに対し、本論文では環境認識処理 (特に実時間視覚処理) まで含めた移動ロボットのシミュレータ View Simulation System (以下 VSS) を提案する。

(1) Real Robot System

(2) Conventional Simulation System

(3) View Simulation System

(4) Online View Simulation System

Fig. 1 Concept of View Simulation System

Fig.1 (3) にそのコンセプトを示す。Fig.1 (1) の実ロボットと同じようにセンサ情報 (画像) からの環境認識 (画像処理) が行われることが、従来のシミュレータ (Fig.1 (2)) と大きく異なる点である。また、画像の生成をリアルタイムで行うことができれば、処理をオンラインで行うことで Fig.1 (4) に示すように実世界で移動しながら同時に仮想環境内で移動のシミュレーションを行うことも可能となる。

実際の視覚を持つ移動ロボット上のプログラムを、そのままシミュレーションに用いることを可能にするため、VSS は以下の四つの機能が必須である。

● 高品質の画像生成機能

生成される画像は、画像処理の対象として十分なリアリティを持たなければいけない。また、ロボットが搭載するカメラの種類 (単眼, ステレオ, 全方位など) に応じた画像を生成可能である必要がある。

● 実時間での画像生成機能

実時間で画像の生成を行えなければならない。これにより、実ロボットの認識・行動プログラムを、時間軸の変化を考えずにそのまま実時間でのシミュレーションに用いることができるようになる。また、Fig.1 (4) のように現実世界での行動とシミュレーションを同時に行うことも可能になる。

● ソフトウェアの互換性

ロボットに搭載されているものと同じ画像処理ハードウェアを持ち、同じソフトウェアを実行できる必要がある。これにより、実ロボット上の認識プログラムを変更せずに、そのままシミュレーションに用いることが可能となる。

●動作コマンドの互換性

ロボットの動作コマンドを受けて物理世界のシミュレート（視点の移動など）を行うことが可能である必要がある。これにより、実ロボット上の行動プログラムをそのままシミュレータ上で実行し、視覚へのフィードバックを得ることができるようになる。

なお、VSSの実装においては、環境モデルの表現および画像の生成方法は、ロボットが実行するタスクやそこで用いる画像処理に応じて、実装時に選択するべきものである。どんなタスクにも対応できる汎用な仮想環境を作ることは、現時点の技術では困難である。

例えば、ロボットサッカーであれば、色により環境（フィールド、ボール、ゴールなど）を認識することを前提にルールが決められているので、ロボット上では、色認識のための画像処理を行う。そこで、シミュレータ上でも当然カラー画像を生成する必要がある。その際、その画像処理においてテクスチャが重要でなければ、省略しても構わないであろう。

このように、現時点では必要となる画像の質とソフトウェアおよびハードウェアの技術を考慮して、システムを実装する必要がある。

3. 画像列の記憶に基づく走行のシミュレータとしてのVSSの実装

3.1 画像列の記憶に基づく走行

本章では、VSSの実装例として、画像列の記憶に基づく走行手法[11]のシミュレーションを行うシステムを構築し、提案したVSSのコンセプトの有効性を確認する。

画像列の記憶に基づくナビゲーション手法の概要をFig. 2に示す。まず教示走行では、人間の操縦のもとでロボットは自動的に画像列を記憶し、その画像列を用いて環境認識、自己位置認識を行いながら自律走行を行うものである。

3.2 実装するシステムの設計方針

VSSの実装においては、仮想環境は実環境の複雑さを表現する必要がある。しかし、どんなタスクにも対応できる汎用な仮想環境を作ることは困難である。本研究では、VSSの実装にあたり以下のような方針を取ることにした。

- (1) 環境をテクスチャを貼った単純な形状としてモデル化する。
- (2) そのテクスチャは実環境から取り込む。
- (3) 光源は環境光のみ存在。
- (4) 光の反射や影はない。
- (5) 床面の滑り、凸凹等は考慮しない。
- (6) ロボットの慣性は考慮しない。
- (7) ロボットの視覚として単眼カメラを用いる。
- (8) ロボットの運動モデルとしてPWSまたは2W1Sを用いる。実環境の3D幾何モデルを自動的に詳細に獲得しCGに利用するという研究（例えば文献[12]）も行われているが、本研究で実装するプロトタイプでは3D幾何モデルは単純なものにし、実環

Fig. 2 Visual navigation method using View Sequence

境の見え方の複雑さはテクスチャによって補うことにする。このモデルは仮想環境で実環境の複雑さを適度（具体的には、実画像上のパターンと仮想画像上のパターンのテンプレートマッチングをとることが可能な程度）に表現でき、かつ描画に要する時間が短くて済む。仮想現実感の研究でも3D幾何モデルの精度を上げることよりもむしろテクスチャとしての実画像を積極的に用いてリアリティを上げていこうという傾向がある[13]。これは、複雑な3D幾何モデルを用いるよりも少ない手間ではリアリティの高いものが生成可能であると考えられるためである。また今回取ったアプローチは、一般にIBR（Image-based Rendering）と総称される画像生成法[14]と比べるとデータの取得が容易で、データ量が少ないという利点がある。また、テクスチャを貼ったCGにおける光源の計算は現状ではまだ時間がかかるため省略した。ロボットのモデルとしては、我々が実験に用いているロボット[11][15]に合わせたものを用意した。

3.3 システムの概要

Fig. 3に、VSSのハードウェア構成を、Fig. 4にソフトウェア構成を示す。実験システムは実際の視覚移動ロボットとまったく同じユーザプログラムが動作するように構成されており、CG生成部、画像処理部を持つ。このようにCG部と画像処理部の間にループを形成し、ビジュアルフィードバックが実現したことにより、VSS上と実ロボットとでソースレベルで同じプログラムが動作可能となる。

3.4 CG生成部

CGの生成は計算機にとって負荷の高い処理を必要とするため、本システムでは、グラフィックスエンジンと呼ばれる専用のハードウェアを持ったグラフィックスワークステーション、Silicon Graphics社のOnyxRealityEngine2（以後OnyxRE2）を

使用している. OnyxRE2 には RGB 出力の他にビデオ出力を持つので, これを画像処理部への入力として利用している.

CG 生成部は環境とロボットの三次元モデルを持ち, 画像を生成する. また, ロボットの動作命令を画像処理部から受け取り, その三次元モデルを更新していくことで, 実世界の実験環境をシミュレートする. 移動ロボットにおける画像以外のセンサ, 例えばシャフトエンコーダの情報も, 仮想環境生成部のモデルから得ることができ, 必要であれば画像処理部に送られる.

CG 部において, 3D の CG を作成するプログラムを書くための API としては OpenInventor を使用する. OpenInventor は OpenGL 上に構築されており, 現在多くの OS に移植されている. また, OpenInventor のデータファイルは, VRML とほぼ同等のデータ形式であるため, 市販の VRML 用モデリングツールにより作成した様々な仮想環境を実験に利用することができる.

3.5 画像処理部

画像処理部には, 実ロボットに搭載されているものと同一のシステム [16] を用いる. 画像処理部ではユーザプログラムが動作するが, このプログラムは実ロボットとソースコードのレベルで同一のもので, この中には画像処理プログラムとロボットの動作を決定するプログラムが含まれる. ユーザプログラムにおいて生成されたロボットの動作命令は, 実ロボットにおいて

Fig. 3 Overview of VSS

Fig. 4 Software configuration of VSS

はコントローラが受け取り, 直ちに実行されるが, VSS においてはコントローラエミュレータが受け取った後, 仮想環境部へ転送される.

3.6 廊下モデルの作成

3.6.1 幾何モデル

廊下の幾何モデルとしては Fig. 5 に示す, 左右の壁, 天井, 床, 正面の 5 面構成される直方体を用い, 五つの面にそれぞれテクスチャを貼り付ける. Fig. 5 の width, height は実測値を用いるが, depth は実際の廊下の長さよりも短くする. つまり, 廊下のある部分から先は, front の画像 1 枚で代用することで, テクスチャのサイズを減らし, 表示の高速化を図る.

3.6.2 テクスチャの取り込み

テクスチャの取り込みには, パノラミックビュー [17] の画像取り込み手法を利用する. パノラミックビューとは画像中のあるラインに注目し, カメラを動かしたときにその部分から得られる連続的な画像データをつなぎ合わせた画像のことである. ここでは, Fig. 6 上に示すように画像中の 4 スリットをスキャンのためのラインとし, カメラを前方に移動しながら床, 天井, 壁のテクスチャを同時にスキャンしていく.

Fig. 5 Geometric model of corridor

Fig. 6 Texture generation

ただし、この方法はカメラの正確な直線移動を必要とする。実際のテクスチャの取り込みには移動ロボットに搭載したカメラを用いているが、ロボットは正確な直線移動は行えない。ここでは、ロボットの移動誤差による画像の見え方のずれを、画像処理（トラッキング）を用いて補正することにした。まずテクスチャの取り込みを始める前に、廊下の正面中央に見えるパターンとその位置を記憶する。そして、そのパターンをテンプレートマッチングを用いて追跡しつづけることで、ロボットの方向の正面からのずれを検出する。そのずれに合わせてテクス

チャを取り込むためのスリットの位置を上下左右に移動させる。このような方法により、ロボットを廊下で一度走行させるだけで、容易にテクスチャを取り込むことができる。取り込んだテクスチャの例（右壁、天井、正面）を Fig.6 下に示す。テクスチャは解像度が一定であり、また、4 ラインの画像を同時に取るため、奥行き方向に関して 4 面のズレがない。しかし、スリットは一定距離前方を見ているため、凹凸がある場合には正確なテクスチャを取ることができない。

このテクスチャを用いて作成した廊下モデルにおいて、モデル内のロボットの視点から見た画像を Fig. 7 に示す。全体としてはかなりリアルな CG 画像を生成できていることが分かる。

4. 画像列の記憶に基づく走行手法の評価への利用

画像列の記憶に基づくナビゲーションでは、教示走行と自律走行のそれぞれで VSS または実ロボットを用いることができるため、Table 1 に示す四つの組み合わせが考えられる。

VSS で生成する画像列は、完全に真直ぐな走行を行った場合の画像列や、既知のずれを与えた場合の画像列など、実際の記録走行では不可能な任意の経路に沿った画像列を作成することができる [18]。実験には、前章で作成した廊下モデルを用いる。一度作成した仮想環境はある瞬間に固定したデータであるため、繰り返し同じものを用いることができる。これにより対照実験が可能であり、実験条件の変更による結果の変動のうち、この世界固有のデータによるものとそうでないものとを分離することが可能となる。

実画像のテクスチャを用いた廊下のモデルを利用すると、作成した CG は非常にリアルなものになる。さらに、画像列を用いた走行では実際のマッチングに用いる画像は解像度を下げた画像であるため、3D 幾何モデルやテクスチャの細部の誤差は問題にならない。ただし、CG と実画像のマッチングを取る場合には、明るさを実環境で得られる画像と一致させなければならないが、これはモデル内の照明や物体の反射率などのパラメータが複雑に関係するため難しい。そこで、この問題にはマッチングの際に画像全体の明るさを正規化することで対処する。

4.1 実験 1: VSS の再現性の評価

まずはじめに、画像列に基づく走行手法において、VSS を用いた走行が、実環境での走行をどの程度再現できるかを評価するための実験を行った。はじめに実ロボットで教示（意図的に直線からずらした）を行い、得られた画像列を用いて実ロボットと VSS で自律走行実験を行って比較する。これは Table 1 の a, b に当たる。このときの走行軌跡を Fig. 8 に示す。実ロボットの走行（実線）は、教示が正確な直線経路でなかったため、細かくふらつきながら徐々に壁側に寄っている。VSS で同じ画像列を用いて走行を再現すると（破線）、ふらつきはないも

Fig. 7 CG of reconstructed corridor model

Table 1 Four combinations of recording run and autonomous run using VSS

	Rec. Run	Auto. Run	Content of Experiment
a	Real Robot	Real Robot	Normal Navigation
b	Real Robot	VSS	Eval. of ViewSeq.
c	VSS	Real Robot	Virtual Path Generation
d	VSS	VSS	Eval. of Virtual ViewSeq.

Fig. 8 Evaluation of VSS using real View Sequence

Fig. 9 Predicted and real trajectory using virtual View Sequence

の徐々に壁側に寄る傾向を示している。

この実験により、VSSでは実ロボットの走行時の細かなふらつきまでは再現できないが、軌跡のおおまかな傾向は再現できることが示された。VSSで再現できなかった誤差は、モデリングにおける仮定によって起きていると考えられる。再現性をより高めることは、より正確にモデリングを行うための手間とトレードオフの関係である。本実装では、これ以上正確にふらつきを再現することに意味があるとは考えず、本実験で得られた再現性を、画像列に基づく走行手法のシミュレーションとして十分有用であると評価する。

4.2 実験2: 走行経路の予測

前節での走行実験では、実ロボットおよびVSSの両方で経路が壁に寄っていくという傾向が得られた。これは、教示走行における画像列の記憶が正確に行われていなかったためであり、ここでは画像列の誤差が走行経路に与える影響をVSSを用いて調べる。画像列の誤差の原因としては次のものが考えられる。

- 教示の際、カメラの位置が経路から横にずれた。
- 教示の際、カメラは姿勢が進行方向正面からずれた。

ここでは、VSSを用いて意図的にカメラの姿勢だけに既知の誤差を与えた画像列を生成し、走行軌跡にどのような影響を及ぼすかをVSSでの走行に用いて予想した。またこの予想を確認するために、同じ画像列を用いて実ロボットでも走行を行った。この実験はTable 1のc,dに当たる。

画像列としては、スタートから250 [cm] から400 [cm] の間の画像は3 [deg] 右を向いて見える画像、900 [cm] から1,100 [cm] の間は3 [deg] 左を向いて見える画像、その他の区間に関しては正面を向いた画像を与えた。この画像列を用いてVSSで10回の仮想走行を行った結果の走行軌跡の平均をFig. 9の破線で示す。画像に与えた姿勢の誤差はわずかなものであるにもかかわらず、走行軌跡に大きく影響が出ることがこの実験によって分かった。

また、Fig. 9の実線は同じ画像列を用いた実ロボットの走行軌跡であり、VSSによる仮想走行の軌跡予想が、実ロボットで

Fig. 10 Improvement of linearity in autonomous run

Fig. 11 Matching error in autonomous run

の走行軌跡と誤差数 [cm] 程度で一致していることが分かる。

4.3 実験3: 実ロボットでの走行精度の改善

ここでは、VSSを用いて仮想走行を行い、正確に真直ぐに進んだ場合の画像列を生成する。この画像列を用いて実ロボットの走行を行えば、通常の教示走行で作成した画像列を用いるよりもより真直ぐに進むことが期待できる。これは仮想環境を用いたロボットのオフライン教示と見ることができる。

Fig. 10の実線が実ロボットでの教示走行により作成した画像列を用いて、実ロボットで走行した軌跡で、破線がVSSを用いて仮想走行を行って作成した画像列を用いて実ロボットで走行した軌跡である。VSSを用いて教示した場合、走行軌跡の直線性が大きく改善されているのが分かる。このように、VSSを用いることで、実ロボットを用いるよりも正確な教示画像列を作成することが可能であることが分かった。

Fig. 11は、この二つの走行の際のマッチングエラー値を示す。値が小さいほど教示時と自律時の画像が似ていることになる。全般的に、仮想教示を行った場合の値が大きい。この原因としては、実環境と仮想環境の細かなモデルの差以外にも、(1) 前半では、スタート地点で生成できるCGがスリットの位置の関係で画面全体にはならない (Fig. 7 参照)。(2) 後半では、廊下モデルでの正面の1枚の画像が視野中に占める割合が大きくなってきた。

といったものが考えられる。ただし、エラー値が多少高くなってもマッチングの結果として得られる1) 画像列中の自己位置の認識、2) 画像とのずれの検出、に失敗することはなかった。

5. オンラインVSSを用いた走行実験

本章では、前章までに述べたVSSをネットワーク経由で実ロボットと接続して、オンラインでシミュレーションを行えるようにしたオンラインVSS [19] のシステムと、それを用いた走行実験について述べる。

Fig. 12 Overview of Online VSS

Fig. 13 Real views (left) and generated virtual views (right)

5.1 オンライン VSS の実装

オンライン VSS では、Fig. 1 (4) のように二つのループが同時に実行され、ロボットは「自分が動いたら、環境の見え方がどう変化するか」を常に予想しながら行動を行うことになる。このように画像の変化を予想し現実の見え方とマッチングを取りながら走行する移動ロボットの研究は、エッジベースの認識を行う研究ではあった [20] が、ビューベースのものではなかった。

このシステムは、オンラインで VSS 内の環境モデルを更新し、そこで走行シミュレーションを行いながら実環境で移動することを目的としている。まず、オンライン VSS の概要を Fig. 12 に示す。ロボットと VSS は無線 LAN を介して接続されている。ロボットは現在得られる画像を VSS に送り、仮想環境を構築しながら走行シミュレーションを行う。実験に使用するロボットは単眼カメラしか備えていないので環境の更新は簡単化し、環境の形状は既知、テクスチャは 1 枚の画像から逆透視変

換により生成することとする。

これにより生成された画像の例を Fig. 13 に示す。上の列は、テクスチャを作るために画像を取り込んだ位置を基準として、それぞれ 0 [m]、2.7 [m]、4.5 [m] の地点にロボットを進めたときに得られた仮想画像、下の列は実際の廊下における 0 [m]、2.7 [m]、4.5 [m] の地点での画像である。テクスチャの解像度は一定ではなく、進むにつれて徐々に低下するため、4.5 [m] の地点での画像ではテクスチャの粗さが目立っている。しかし、5 [m] 程度までの画像は、実画像とのマッチングを取り自己位置を認識することに問題はなかった。

5.2 走行手順

オンライン VSS を用いたロボットシステムで、走行中にモデルの更新をしながら経路を生成し、自律走行する手順は以下ようになる。上に述べた通り、逆透視変換により生成したテクスチャの解像度はロボットが進むにしたがって徐々に下がるため一度に走行する距離は 5 [m] とし、それを繰り返すことにした。

- (1) ロボットから VSS に 1 枚の画像を転送する。
- (2) 逆透視変換により廊下モデルのテクスチャを生成し、VSS のモデルを更新する。
- (3) VSS で 5 [m] の仮想教示走行を行い画像列を生成する。
- (4) VSS からロボットに画像列を転送する。
- (5) ロボットが画像列に従って自律走行を行い、停止する。
- (6) (1) に戻る。

また、逆透視変換によってテクスチャを生成するとき、ロボットは必ずしも廊下の中心で真正面を向いてるとは限らない。走行に用いる低解像度の画像では位置や姿勢のずれを正確に検出することは困難なため、VSS で生成した画像列を用いた走行では、経路からの位置のずれは最大 5 [cm]、姿勢のずれは 2 [deg] 程度ある。ここで、位置のずれよりも姿勢のずれの方が画像へ大きく影響するので、画像のずれはすべて姿勢のずれによるものとみなし、高解像度の画像を用いて画像のずれを検出し、逆透視変換するときに補正することにした。

VSS は任意の解像度の画像を生成することができるため、ロボットが停止した後に、その位置で見えるはずの画像をより高い解像度で生成してロボット側へ送る。この画像と、実際に見える画像を用いて相関演算を行い、画像のずれを求める。

5.3 経路生成

経路生成は、VSS での仮想教示走行によって行う。通常は、ロボット側から走りたい経路を VSS に送る。VSS はその経路に沿って仮想ロボットを進ませる、仮想教示走行を行うことで画像列を生成する。

また、4 章で述べたように、VSS は生成された画像列を使うとどのような走行軌跡が得られるかをシミュレートする機能を持っているので、必要に応じてあらかじめ VSS で仮想自律走行を行い、走行軌跡を予想、評価してから実際の走行を行うこともできる。

5.4 走行実験

実際の廊下において 5[m] ずつの走行を 4 回繰り返し、合計 20[m] の直線経路の走行を行う実験を行った。テクスチャを更新するときに、ロボットの姿勢のずれを考慮しない場合は誤差が蓄積されて 3 回目の更新時に廊下の中心線から約 40[cm] ずれることもあったが、姿勢のずれを検出して補正した場合には、走行の終了時にも 10[cm] 以内のずれに収まり、安定した走行結果が得られた。

6. 考 察

4 章の実験の結果、本シミュレータで得られる走行軌跡は、実ロボットの慣性や、滑り、床面の凸凹などは考慮していないなど、環境のモデリングに省略した点がいくつかあるため、実世界のものに比べて滑らかになりがちであるが、全体の傾向はよく一致していた。しかし、これは「ロボットが行っている画像処理（全体を粗く見る）と作成したモデルの相性がよかった」ためであるともいえる。我々は、注視に基づく障害物検出手法の研究にも VSS を用いている [21] が、そこでは環境の形状モデルやテクスチャの生成手法には異なるアプローチを取っている。このように、すべての視覚移動ロボットに適した統一的な仮想環境を作ることは現状ではまだ困難である。つまり、シミュレーションの対象となるロボットが利用する画像特徴をよく再現できるよう、場合に応じてモデリング手法を決める必要があるが、その検討については今後の課題とする。

VSS での走行では、画像列以外の要素はほぼ理想的な状態である。このため、実ロボットでの教示によって作成した画像列を用いて VSS で走行を行えば、作成した画像列の影響のみを調べることができる。実際には VSS による実験でも通信の時間遅れが存在するため、VSS による走行も多少のブレがあるが、繰り返しシミュレーションを行うことで、この影響を含めた評価が可能である。

また、5 章の実験の結果、ロボットが仮想環境において走行シミュレーションを行いながら見え方の変化を予想し、その予想と合うように実環境で移動することが可能であることが確認された。今後は、ステレオ視覚やレーザレンジファインダなど廊下の形状を計測するセンサと組み合わせて、実環境を探索し環境モデル（形状やテクスチャ）を更新しながら移動するロボットへの応用が期待できる。

7. お わ り に

本論文では、視覚移動ロボットのための仮想環境を用いたシ

ミュレーション環境 View Simulation System を提案し、プロトタイプシステムの実装およびそれを利用した実験結果について述べた。本システムは 1) CG 生成用コンピュータ、2) ロボットと同一の画像処理装置、を利用することで、視覚処理まで含めたロボットの行動のシミュレーションを行うことができる。また、その実装においては仮想環境のモデルの作成に、テクスチャを用いることで現実世界の複雑さを表現した廊下モデルを作成した。実装したシステムは、我々がこれまでに提案してきた、視覚によるロボットのナビゲーション手法のシミュレーションに用いて、走行誤差の原因の分析と走行軌跡の予測、走行精度の改善を行った。また、VSS と実ロボットをオンラインで接続することで、見え方の変化を予想しながら行動することが可能であることを示した。

現在は限定された仮想環境を用いたシミュレーションしか実現できていないが、バーチャルリアリティの分野でのモデリング技術の発達により、よりリアルな仮想環境の構築が手軽にできるようになれば、視覚を含むシミュレーションがより広い分野に応用できることが期待できる。

また、近年パーソナルコンピュータ用の安価な 3D 画像表示ボードの性能が急速に向上している。これらのボードはゲームをターゲットして開発されてきたが、同時に OpenGL などのプログラミング環境も整備され、三次元 CG が誰もが手軽に使える技術となってきた。比留川ら [22] や Kuffner ら [9] もヒューマノイド型ロボットのための CG を用いた画像処理のシミュレーションを提案しており、今後仮想環境を用いた画像を含めたシミュレーションは、視覚を持つロボットにおける有用な開発ツールとして、広く用いられるようになっていくと期待できる。

参 考 文 献

- [1] 宮崎猛, 柴田智広, 松本吉央, 稲葉雅幸, 井上博允: “仮想ビュー生成器を用いた視覚移動ロボットのシミュレーション環境 “View Simulation System” の構築”, 第 14 回日本ロボット学会学術講演会予稿集, pp.575-576, 1996.
- [2] Y. Matsumoto, T. Miyazaki, M. Inaba and H. Inoue: “View Simulation System: A Mobile Robot Simulator using VR Technology,” Proc. of 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.936-941, 1999.
- [3] D. Marr: Vision. Freeman, 1982.
- [4] D. Terzopoulos, T. Rabie and R. Grzeszczuk: “Perception and Learning in Artificial Animals,” Proc. of ALife V, pp.313-320, 1996.
- [5] 岡田浩之, 山川宏, 大森隆司: “環境同定と報酬獲得のトレードオフを解消する報酬・嫌悪の二次元評価強化学習の提案”, 日本ロボット学会誌, vol.19, no.2, pp.244-251, 2001.
- [6] R.A. Brooks and M. J. Mataric: ‘Real Robot, Real Learning Problems,’ Robot Learning. J.H. Connell et al., Eds, Kluwer Academic Publishers, 1993.
- [7] N. Itsuki: “Soccer Server: a simulator for RoboCup,” Proc. of JSAI AI-Symposium 95: Special Session on RoboCup, 1995.
- [8] 岡田浩之, 関口実, 渡部信雄, 松尾和洋: “移動ロボットのための仮想環境システム”, ロボティクス・メカトロニクス講演会’96 講演論文集, pp.754-755, 1996.
- [9] J. Kuffner, S. Kagami, M. Inaba and H. Inoue: “Graphical Simulation and High-Level Control of Humanoid Robots,” Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp.1943-1948, 2000.
- [10] M. Asada, S. Noda, S. Tawaratsumida and K. Hosoda: “Vision-Based Behavior Acquisition For A Shooting Robot By Using A

- Reinforcement Learning,” Proc. of IAPR/IEEE Workshop on Visual Behaviors, pp.112–118, 1996.
- [11] 松本吉央, 稲葉雅幸, 井上博允: “視野画像列を利用した経路表現に基づくナビゲーション”, 日本ロボット学会誌, vol.15, no.2, pp.236–242, 1997.
- [12] H.Y. Shum, K. Ikeuchi and R. Reddy: “Virtual Reality Modeling From a Sequence of Range Images,” Proc. of IEEE Int’l Conf. on Intelligent Robotics and Systems, pp.703–710, 1994.
- [13] 廣瀬通孝, 宮田亮介, 谷川智洋: “二次元実画像を用いた三次元仮想世界の構築”, ロボティクス・メカトロニクス講演会’96 講演論文集, pp.1047–1050, 1996.
- [14] S.E. Chen: “Quicktimevr—an image-based approach to virtual environment navigation,” Proc. SIGGRAPH, pp.29–38, 1995.
- [15] 柴田智広, 松本吉央, 稲葉雅幸, 井上博允: “人が搭乗しその場その場で行動を指示できるパーソナル視覚移動ロボット HyperScooter の開発”, 日本ロボット学会誌, vol.14, no.8, pp.1138–1144, 1996.
- [16] 井上博允, 稲葉雅幸, 森武俊, 立川哲也: “局所相関演算に基づく実時間ビジョンシステムの開発”, 日本ロボット学会誌, vol.13, no.1, pp.134–140, 1995.
- [17] J.Y. Zheng and S. Tsuji: “Panoramic Representation for Route Recognition by Mobile Robot,” International Journal of Computer Vision, vol.9, no.1, pp.55–76, 1992.
- [18] 松本吉央, 宮崎猛, 稲葉雅幸, 井上博允: “視野画像列の記憶に基づくナビゲーション—View Simulation System を用いた画像列の生成—”, 第 14 回日本ロボット学会学術講演会予稿集, pp.291–292, 1996.
- [19] 松本吉央, 宮崎猛, 稲葉雅幸, 井上博允: “画像列に基づくナビゲーション—View Simulation System を用いたオンライン経路生成—”, 日本機械学会ロボティクスメカトロニクス講演会’97, pp.707–708, 1997.
- [20] A. Kosaka and A.C. Kak: “Fast Vision-Guided Mobile Robot Navigation Using Model-Based Reasoning and Prediction of Uncertainties,” CVGIP: IMAGE UNDERSTANDING, vol.56, no.9, pp.271–329, 1992.
- [21] 柴田智広, 宮崎猛, 松本吉央, 稲葉雅幸, 井上博允: “ハイパースクーターの研究—移動ロボットのための単眼による衝突時間予測器の開発—”, 第 13 回日本ロボット学会学術講演会予稿集, pp.1019–1020, 1995.
- [22] 比留川博久, 喜多伸之, 喜多泰代, 末広尚士, 平井成典, 小原一太郎, 河村進: “HRP 仮想プラットフォームにおける view シミュレータ”, 第 17 回日本ロボット学会学術講演会予稿集, pp.1217–1218, 1999.

松本吉央 (Yoshio Matsumoto)

1970 年 8 月 21 日生。1993 年 3 月東京大学工学部卒業。1998 年 3 月同大学大学院工学系研究科情報工学専攻博士課程修了。同年 4 月オーストラリア国立大学研究員。1999 年 1 月奈良先端科学技術大学院大学情報科学研究科助手, 2001 年 1 月同助教授。ロボットビジョンおよびヒューマンインタフェースの研究に従事。電子情報通信学会会員。(日本ロボット学会正会員)

稲葉雅幸 (Masayuki Inaba)

1958 年 5 月 23 日生。1981 年東京大学工学部機械工学科卒業, 1986 年東京大学大学院工学系研究科情報工学専門課程修了。工学博士。1986 年東京大学講師, 1989 年同助教授, 2000 年同教授, 大学院情報学環所属, 現在に至る。日本機械学会, 情報処理学会, 計測自動制御学会, 人工知能学会, ソフトウェア学会各会員。(日本ロボット学会正会員)

宮崎 猛 (Takeshi Miyazaki)

1972 年 5 月 11 日生。1995 年 3 月東京大学工学部卒業。1997 年 3 月同大学大学院工学系研究科情報工学専攻修士課程修了。1997 年 4 月より株式会社キーエンスに勤務。在学中は, パーチャルリアリティーおよび移動ロボットの研究に従事。

井上博允 (Hirochika Inoue)

1942 年 7 月生。1965 年東京大学工学部卒業。1970 年同大学大学院博士課程修了。工学博士。同年電子技術総合研究所入所。知能ロボットの研究開発に従事。1977 年東京大学工学部機械工学科助教授。1984 年教授。現在, 機械情報工学科教授。ロボット全般, 人工知能, 情報システム工学の研究と教育に従事。(日本ロボット学会正会員)