高速移動ロボットを遠隔操縦するための GUI の開発

Development of Graphical User Interface

for Operating High-speed Mobile Robot Remotely

○小川 明臣 (岡山大学) 正 前山 祥一 (岡山大学) 正 田中 豊 (岡山大学)

Akiomi OGAWA, Okayama University, 3-1-1, Tsushima-naka, Okayama

Shoichi MAEYAMA, Okayama University Yutaka TANAKA, Okayama University

Sarin gas attack on the Tokyo subway (1995), rescue operation had a rough going because there was danger of secondary disaster by chemicals and radiations. Then, it was hoped to realize the system that reduced risk of rescue activities by remotely controlling mobile robot, and understanding the situation in building promptly. For such a background, the project of mobile RT system in the stricken building started and we also participate in it. To control a mobile robot remotely and speedy, the time delay on communication and indication is serious problem. Therefore, we propose the robust GUI to overcome the time delay by giving passing points on the large area map and the camera image combined with the priori-known map data.

Key Words: Rescue, GUI, High-speed robot, time delay.

1. はじめに

地下鉄サリン事件 (1995) や東海村 JCO 臨界事故 (1999) では、化学物質や放射線により二次災害が発生する危険性があるため、救助活動が難航した。そこで、移動ロボットを遠隔操作し、建造物内部の状況を迅速に把握することで、レスキュー活動のリスクを軽減するシステムの実現が望まれた。このような背景のもと、被災建造物内移動 RT システムの開発プロジェクトが発足し、本研究もこのプロジェクトに参画している。

操作用 PC を用いて高速移動ロボットを遠隔操作する場合、通信時間が影響し情報提示が遅延することで、オペレータの操作ミスを招くことが懸念される。そこで、本研究ではこの問題を解決するために、既知の地図情報とカメラ画像の合成画像を提示し、広域地図とカメラ画像上で通過点を与えることによる経路指示・修正を行う GUI の開発を行った。そして、模擬環境においてロボットを遠隔操作する実験を行い、開発した GUI の有用性を検証したので報告する。

2. 被災建造物内移動 RT システムの開発プロジェクト

本研究プロジェクトでは、被災前の環境が既知である地下街 (700[m] 以上) を対象環境としている。この環境では、探索ロボットと操作用 PC 間で直接無線通信を行うことは困難であるため、光ファイバ敷設ロボットを用いて基幹となる有線 LAN を設置し、アクセスポイントを環境内部に 50[m] 間隔で敷設する。これにより、環境深部での通信が可能となる。また、探索ロボット自体にアドホック無線 LAN を搭載することにより、他の通信エリアにおいても操作用 PC と通信を行うことが可能となる。これらにより、複数の探索ロボットが高速移動し、対象とする環境内部の状況を迅速に把握する。目標とするシステムの完成イメージ図を図 1 に示す。

本研究では、高速移動ロボットを遠隔操縦するための GUI の開発を行う。

3. 高速移動ロボットの遠隔操縦システム

本章では、まず、高速移動ロボットを遠隔操縦するためのシステムの概要について説明を行い、次に、構築したシステムにおける経路指示のシナリオについて述べる。

Fig. 1 Mobile RT system in the stricken building

3.1 目標地点への迅速な到達を目的としたシステムの概要

複数のアクセスポイントを介して通信を行った場合、情報提示の遅れによるオペレータの操作ミスが問題となる。従来のジョイスティックやハンドルなどの操縦インタフェースを用いた場合、ロボットに速度や加速度などの時間に依存する情報を与えるため遅延による影響を受け易い。そこで本研究では、ポインティングデバイスを用い、画面上をポイントして経路を指定することとする。この手法は時間に依存しない位置情報を与えるため遅延による影響が少ない。また、オペレータに提示する GUI 画面は、被災前の環境地図から作成した画像をロボットに搭載されたカメラ画像上に合成することで認知性を向上させる。この GUI 画面を提示し、目標地点への迅速な到達が可能なシステムの構築を目指す (図 2)。

Fig. 2 Basic structure of the proposed GUI system

このシステムは大きく分けて、高速移動が可能なロボットと操作用 PC から構成される。まず、ロボットに搭載された外界センサであるカメラから、周囲の環境情報を獲得する。この情報とオドメトリ情報を操作用 PC に送信する。獲得した情報と既知である環境地図をもとに、鳥瞰視点の画面とロボット視点の画面を作成し、オペレータに提示する。オペレータは、提示された GUI 画面上に直接指示を与えることでロボットへの経路指示・修正を行う。

3.2 走行経路指示のシナリオ

本研究プロジェクトで迅速に環境内部の状況を把握するために広域、中域、近域の3つにおける経路指示のシナリオが必要であると考えられる。ここで述べる広域とは対象環境全体、中域とはロボットに搭載されたセンサの視野内、近域とはロボット近傍を指す。

広域においては、ロボットの走行開始前に既知の地図全体から規範となる目標地点までの経路を決定する。しかしながら、被災環境では経路上に被災前の既知環境にはない人や障害物の存在が考えられる。そこで、中域において、オペレータが GUI 画面上を見ながら操縦することにより、ロボットの走行中に経路修正を行う。近域では、環境内部の状況把握のためセンサにより周辺環境の探索を行う。この場合、細かい移動を行うために、従来のジョイスティックを用いて操縦を行う。

本研究では、高速移動での遠隔操縦を対象としているため、近域での操縦は研究対象とせず、走行開始前の全体の地図に対する経路指示と、走行中に得た画像上で経路修正を行える GUI の開発を行う。

4. 情報提示の遅れに頑強な GUI の提案

本章では、情報提示の遅れを補正し、高速移動を行いながら操縦できる GUI を提案する。

4.1 ポインティングによる経路指示

操縦インタフェースとしてマウスやタッチパネルなどのポインティングデバイスを使用し、画像上の任意の位置をポイントする方法を提案する。この手法では、時間に依存しない情報である位置情報をロボットに与える [1]。これにより、情報提示の遅れによる影響を軽減できると考えられる。また、ポインティングによる経路指示は、ジョイスティックやコントローラなどの操縦インタフェースを用いた経路指示に比べ、オペレータの操縦技能を必要としないため、一般的な GUI としても有用であると言える [2]。

経路を指示する手順は、オペレータが鳥瞰視点の地図画像をもとに、地図上を複数点ポイントすることにより、目標地点までの全体の経路を決定する (図 3)。

全体の経路が決定後、走行開始ボタンを押すことによりロボットは走行を開始する。ロボットが走行中は、地図にない人や障害物を避けるため、ロボット視点画面をポイントすることにより、経路変更を行う (図 4)。

Fig. 3 Pre routing on the map

4.2 カメラ画像と三次元地図の合成

高速移動ロボットを遠隔操縦する際、環境情報を文字や数値でオペレータに提示すると、認識に時間がかかり、操作ミスを起こす恐れがある。これに対して、情報の認識に時間がかからないグラフィカルな情報をオペレータに提示すれば、操作ミスを減少させることができると考えられる。そこで、ロボットに搭載されたカメラから獲得したカメラ画像に、操作用 PC が持つ既知の環境地図情報とロボットから獲得したオドメトリ情報から作成した三次元地図を補助情報として合成することとした (図 5)。これにより、オペレータは提示される合成画像に集中できる。合成を行う利点としては、平面情報である画像にカメラ画像に立体的な情報が付加され、距離を把握することが可能となるが挙げられる。また、環境内部が暗いとき、環境を把握する情報として有効である。

カメラ画像と三次元地図を合成する際、オペレータがカメラ画像から得られる情報量の低下を防ぐため、ワイヤフレームで三次元地図を合成することとする。

次節で説明を行う補助情報の提示も、カメラ画像に三次元画像を合成することにより実現を目指す。

4.3 補助情報の提示

指定した経路の位置情報

ポインティングにより指定した複数点の位置に、円錐のマークを表示する。これにより、オペレータはロボットの走行経路を予想し、地図にない障害物を避ける情報として有効である。また、オペレータの認知性を向上させる情報としても有効である (図 6)。

ロボットの推定位置情報

本章の始めて述べたように、情報提示の遅れにより、オペレータが認識している位置と、実際にロボットが走行している位置には誤差が生じる。そこで、オドメトリ情報を獲得した時間と、ロボットの走行速度から、実際にロボットが走行している位置を推定する。この位置を、カメラ画像上にロボットのモデルを描画することにより、オペレータがミスなく操縦を行える情報を提示する (図 6)。

高速移動を維持できる経路情報

ロボットの走行経路の指示・修正を行う際、ポインティングにより指定した経路の中で、回転半径が小さな場所ではロボットは高速移動を維持できない。したがって、高速移動を維持するためには、ある一定値以上の回転半径が必要となる。そこで、回転半径が一定値以上になる経路の範囲を三次元画像で表示し、カメラ画像上に描画することとする (図 6)。

5. カメラ画像と三次元環境モデルの合成表示

前章では、情報提示の遅れを補正し、オペレータが移動ロボットをミスなく遠隔操縦できる GUI を提案した。そこで本章で

Fig. 4 Routing on the image

Fig. 5 Composite image

Fig. 6 Supplemental information

は、まず、GUI 画面の構成について述べ、次に、この画面を実装するために必要な要素について説明を行う。

5.1 GUI 画面の構成

本章で提案した GUI 画面は、鳥瞰視点画面とロボット視点画面を有する (図 7)。画面右側の鳥瞰視点画面は、既知の地図全体を上から見た視点の画面であり、主に全体の経路を指定するために利用する。画面左側のロボット視点の画面は、ロボットの前方をカメラにより写した画面で、環境内部の情報を把握することが可能となる。これにより、ロボットの走行中に経路を変更し、既知の環境地図にない障害物を回避することが可能となる。また、画面上部にメニューバーを配置し、環境地図の読み込みを実現する。

Fig. 7 Developed GUI

ロボット視点の画面は、カメラ画像の上に、壁と床の境界線を緑色のワイヤーフレームで描画することにより認知性の高い画面を構築する。詳しい実装方法は 5.3 節で述べる。

鳥瞰視点の画面は、ロボットの走行している位置を常に中心に表示する。ロボットの座標値は、TCP/IP 通信によりロボットから受信し逐次更新を行う。オペレータは、これらの画面を利用して全体の経路をポインティングにより指示する。経路に指定した座標には、円錐のマークを表示した。

5.2 走行経路指示

全体の経路を指示する場合、カメラキャリブレーションを用いて、オペレータがクリックした任意の位置座標を、画面上の座標系からロボット座標系に座標変換する。この座標を配列に格納し、経路として登録する。

オペレータが走行開始ボタンを押すと、通過点の座標をロボットに送信し、ロボットは与えられた座標に向かって走行を開始

する。ロボットが走行中は、逐次送信されるロボットの位置座標と現在の目標地点との距離を計算し、50[cm] 以内に近づいたとき、次の通過点の座標をロボットに送信する。

5.3 GUI の各構成要素の実装

本研究で開発した GUI のシステム構成を図 8 に示す。図 8(a) では、三次元環境内で実環境におけるカメラの視点と同じ視点で三次元環境をレンダリングした画像を、獲得したカメラ画像の上に描画することで、ロボット視点の画面を作成していた。しかしながら、この手法ではパラメータの調整が難しく、カメラ画像上の壁と床の境界線と合成用画像のワイヤーフレームにずれが生じた。また、計算量が多いという欠点もある。

そこで、これらの欠点を軽減するため図 8(b) に示すシステムを構築した。このシステムは、まず、地図データの座標値を読み込み、ロボット座標系から画面上の座標系に変換する。背景が黒色である画像の上に、変換により得られた座標値をもとにした緑色の線を描画する。次に、この画像の黒色の部分を透明色に指定して、獲得したカメラ画像の上に描画することでカメラ画像と三次元画像の合成を実現する。これにより、情報提示の遅れを補正し、オペレータがミスなくロボットを遠隔操縦できるロボット視点の画面を実現する。

次節より、このシステムを構成する各要素について説明を行う。

(a) Render from 3D CG

(b) Direct draw from map

Fig. 8 Flow to generate the composite image

5.3.1 既知の地図情報

対象となる環境内部の地図は既知の情報として扱うために、環境情報はベクトルマップ表現で表し、外部データとして保持することとする。これにより、鳥瞰視点画面とロボット視点画面を構築する上で利用することが可能となる。

5.3.2 オドメトリ情報

車輪の回転量を測定するエンコード情報から、オドメトリに基づく位置情報を計算し、TCP/IP 通信により、一定間隔でロボットに搭載された制御用 PC から操作用 PC にネットワークを介して送信する。この情報を三次元環境内を走行する三次元ロボットの位置情報や、現在の目標地点から次の目標地点に切替えるための情報として扱う。

5.3.3 カメラ画像

カメラ画像を獲得し、データとして扱うために JMF (Java Media Framework) API を用いることとした。これにより Java プログラム内で音声・映像などのマルチメディアデータを扱うことが可能となる。さらに、JMF API を用いることで、獲得した画像の各画素にアクセスし、処理を加えて画像を表示することが可能となる。

5.3.4 三次元画像の表示

Java で作成した GUI 画面上に、ロボットや環境地図を三次元画像で表示するために Java3D API を用いた。Java3D API は Java の拡張 API であり、Java プログラムの中で三次元グラフィックスや三次元音響を利用できる。

6. 走行実験による GUI の検証

本研究で作成した GUI の有用性を検証するために、模擬環境にてロボットを目標地点まで到達させる走行実験を行った。なお、使用予定の三連クローラ型ロボットが未完成であるため、本研究では、50[cm/s] で走行可能な車輪型移動ロボットを使用し、GUI の評価を行った。

6.1 実験に用いたシステムの構成

今回、車輪型移動ロボット「ビーゴ」(図 9) に USB カメラと制御用 PC を搭載し、実験に用いた。このロボットは、操作用 PC から送信された 2 つの通過点間を直線追従する。このとき、ロボットは 50[cm/s] 程度で走行可能である。また、このロボットは車輪エンコーダからオドメトリに基づき自己位置推定を行うことができる。

ロボットに搭載したカメラは、画素数 30 万画素、視野角約 60[deg] であり、毎秒 15 フレーム間隔でキャプチャすることができる。このカメラを地面から 61[cm] の位置に搭載し、地面と水平方向から下に 26 度傾けた。

オペレータが操作用 PC 上の GUI を操作することで制御用 PC にコマンドが送信され、「ビーゴ」はコマンドに応じた走行を行う。制御用 PC はモータのエンコーダ情報を獲得し、この情報をもとにロボットの位置を計算し操作用 PC に送信する。また、カメラ画像は制御用 PC に接続されたカメラからリアルタイムに配信が行われる。制御用 PC と操作用 PC 間の通信はネットワークを介して行われる。

6.2 対象とする環境

実験を行う上で対象とする環境は、本研究で想定する凹凸の少ない環境である岡山大学総合研究棟 5F 南側廊下とした。この環境は、実測により既知の地図として用いる (図 10(b))。また、対象とする環境に近づけるため、既知の地図にはない障害物を設置した (図 10(a))。

Fig. 10 Target environment

6.3 模擬環境における走行実験

オペレータは開発した GUI 画面のみを見てポインティングにより全体の経路を決定し、「ビーゴ」を初期位置から障害物を回避して目標地点に到達させる走行実験を行った。

6.4 検証実験の結果

開発した GUI を用いて 7[m] 離れた目標地点へ到達できた。しかしながら、ロボットの到達地点と目標地点には 30[cm] 程度の誤差が生じた。これは、カメラキャリブレーションによる座標変換による誤差と、遠距離をポインティングするにつれ 1 画素あたりに占める距離の割合が増えることが原因と考えられる。目標地点においては 1 画素あたり 25[cm] 程度であった。障害物への衝突はなかったため、この誤差は広域、中域の経路指示において問題ない誤差であると言える。

また、ロボット視点の画面におけるカメラ画像と三次元地図の画面合成にはずれが生じた。これは地面の座標から画面上の座標に変換する座標変換によるずれが考えられるため、座標変換に用いる値を正確なものにする必要がある。

6.5 実験で得られた知見

経路指示する情報として位置情報を用いたため、壁や障害物への衝突は起こさず 50[m/s] で走行可能であった。このためロボットが高速移動を行う場合でもこの方法は有効であると考えられる。また、画面のレイアウトは、ロボット視点の画面と鳥瞰視点の画面の 2 つで十分操縦可能であることが確認できた。このレイアウトは複数台のロボットを操縦する場合、ロボットの台数と同数のロボット視点の画面と 1 つの鳥瞰視点の画面で構成することで、1 つの GUI で統合制御を行うことができる拡張性の高い GUI であると言える。

一方、ロボット視点の画面には回転時に、カメラ画像と合成画面のずれによる認知性の低下が問題となった。この問題は、カメラ画像と合成画像、ロボットの位置情報の更新周期を同期させることで解決できると考えられる。

7. まとめ

本稿では、ロボットに搭載されたカメラから得られた画像と、既知の三次元地図を利用して、情報提示の遅れに頑強な GUI の開発を行い、走行実験により有用性を示した。

そこで、今後はロボットが走行中の経路修正と情報提示の遅れを補正する補助情報の提示を実装し、プロジェクトで開発した高速移動ロボットを用いて開発した GUI の有用性を検証する。更に、GUI の操作性を向上させる情報提示システムについて検討していく。

謝辞

本研究は、NEDO (独立行政法人新エネルギー・産業技術総合開発機構) の開発支援により実施されました。

文献

- [1] Jennifer Kay, Charles Thorpe: "Remote Driving with Image Transmission Delays", The Robotics Institute (September 1995)
- [2] Takeshi SEKIMOTO, Takashi TSUBOUCHI and Shin'ichi YUTA: "A Simple Driving Device for a Vehicle - Implementation and Evaluation -", Intelligent Robot Laboratory, Institute of Information Science and Electronics