

携帯カメラを用いたユーザ視点のロボットコントロール

細井 一弘 新田 亮 屋比久 保史 杉本 雅則

東京大学大学院 新領域創成科学研究科

本研究では、ユーザとロボットが同一空間上にいる環境で、ユーザが自己の視点に基づいて、直感的にロボットに操作・指示を行えるインタフェースを提案する。ユーザはモバイルデバイスに搭載されたカメラで、ロボットを捕らえ、移動させたい方向にモバイルデバイスを振ることで、ロボットを移動させることができる。ユーザはロボットの視点になって考える必要がなく、任意の方向から自由にロボットを操作可能である。

本研究では、提案システムのプロトタイプとして、ノートPCを用いたシステムを構築した。本稿では、飛行船型ロボットを操作対象としたシステムについて、その操作方法、構成について述べる。

Remote Control for Robots from the Viewpoint of Users using Camera Phone

Kazuhiro Hosoi Ryo Nitta Yasufumi Yabiku Masanori Sugimoto

Graduate School of Frontier Sciences, The University of Tokyo

In this paper, we propose a remote control technique for indoor flying robots that allows a user to control them from his viewpoint. By capturing an image of robots with a camera mounted on a mobile phone and moving it in a three dimensional space, a user can intuitively control the robots as he would intend.

1 はじめに

近年、人間と関わりながら活動するロボットに関する研究が、盛んに行われている。ASIMO や AIBO に代表されるようなペットや友人としての役割を果たすエンターテインメント性を持ったロボットや介護ロボット、家事手伝いロボットなど、多種多様なロボットが開発されている。このようなロボットは、従来の産業ロボットとは違い、オフィスや家庭などの日常的な空間で使用されるため、機械・ロボット操作に不慣れな人でも、手軽で直感的に利用できるロボット・インタフェースが要求される。

このような背景の下で、音声やジェスチャーを利用した様々なインターフェースが開発されてきた。音声認識やジェスチャー認識には、ロボットに搭載されたマイクやカメラを用いるものや、環境に埋め込まれたセンサを用いるもの、ユーザに Head Mounted Display(HMD) や特殊なデバイスを装着されるものなどが挙げられる [8]。

ここで、ユーザがロボットを(空間的に)操作する場合、どのような座標系を意識して命令を出すかが問題になる [3]。例えば、遠隔地にあるロボット

に搭載されたカメラ画像を基に操作する場合、ユーザはロボット座標系(ロボットを中心とする座標系)で考え、指示を出すことになる。これに対し、ユーザとロボットが同一空間上において、ロボットに指示を出す場合は、ユーザの視点によるユーザ座標系での指示の方が、ユーザにとってより直感的な操作であると考えられる。

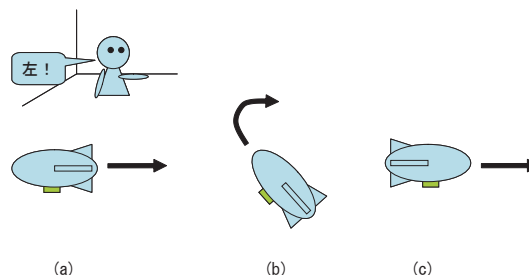


図 1: ユーザ視点の命令とロボットの行動: 3つの飛行船は、それぞれの状態に応じて、ユーザの意図する行動をとるように状態を変化させる。

ユーザ座標系での操作は、ロボットはユーザが意図する行動を採るために、現在の状態に応じて別々の状態変化をしなければならない。(図1参照)。こ

のような操作手法では，ロボットに搭載されたセンサにより，ユーザの位置を特定するよりも，ユーザがロボットの状態を取得できるインタフェースを身につける方が，容易に実現できると考えられる．従来の研究では，ユーザがHMDを装着し，頭部に取り付けられたカメラ画像を基に指示を出すことで，これを実現している [5]．

本研究では，モバイルデバイスに標準装備されたカメラを用いて，ユーザ視点でのロボット操作法を提案する．ユーザはモバイルデバイスのカメラでロボットを捕らえ，移動させたい方向にモバイルデバイスを振ることで，ユーザ視点でロボットを移動させることができる (図2 参照)．カメラ搭載のモバイルデバイスは，携帯電話を始めとして，幅広く普及しており，いつでもどこでも手軽に利用することができる．

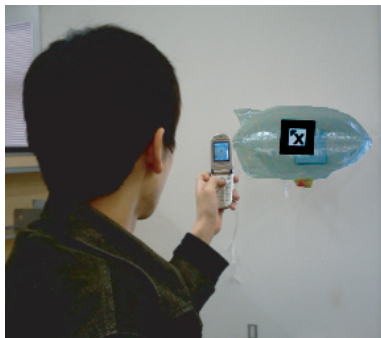


図2: 携帯電話のカメラを用いる操作法

提案システムの利点は次の通りである．

- ロボットの視点を考えないで操作可能．
- 新たにロボットにセンサ群を追加する必要がない．
- 任意の方向からロボットを操作可能．

また，図3のように状態が異なる複数台のロボットが存在する場合，一度の操作で，ロボット群を一斉に移動させることが可能であると考えられる．1台のみを操作したい場合は，特定のロボットだけをカメラで撮影すればよい．さらにヒューマノイドのように複雑な機構を持つロボットに対しては，ロボットの一部分をカメラで撮影することで，例えば，首を振る，腕を上げる，腰を下ろすなどの行動も，直感的に操作できると考えられる．

本研究では，以下のような使用を考えている．

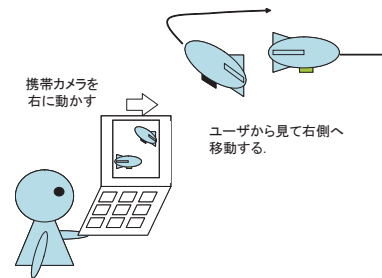


図3: 群ロボットの一括操作

ユーザシナリオ

ユーザとヒューマノイドロボットが同じ部屋の中にいる．ユーザは，部屋にある本棚の手の届かない上段にある本を読みたい．そこで，ロボットが本棚まで移動して，棚の上段から本を取り出し，ユーザの位置まで運んでくる．

このシナリオを本システムで実現すると以下のようなになる．

ロボットを移動させる．

まず，ユーザは，自分の携帯をとりだして，カメラでロボットを撮る．そのまま移動させたい方向（棚がある方向）に携帯を振り，ロボットを移動させる．ロボットが移動中，ユーザは状況に応じてロボットをカメラで撮らえて，携帯を軽く振り，移動方向を修正する操作をする．

ロボットを停止させる．

ロボットが棚の付近まで移動したら，ロボットを静止させるために，カメラをロボットに向けて固定する．するとロボットは，携帯画面の中央に映るように静止する．

ロボットのアームを利用して本を取り出す．

ユーザは，ロボットのアーム部分を拡大して撮影し，アームを移動させたい方向（棚の上段の方向）に携帯を振ることで，アームだけが動き出す．取り出したい本の位置まで，アームが移動したら，先と同様の方法で，カメラをアームに向けて，アームを止める．そして，特定のボタンを押して（あるいは特別なジェスチャーをして），ロボットに本を取り出させる．

本をユーザのところへ運ぶ。

再びロボット全体をカメラで撮影し、携帯を手招きするように振る（手前に引く）。ロボットはユーザに向かって移動する。最後にロボットを停止させたい位置まできたら、カメラをロボットに向けて固定する。

以上のようなシナリオを踏まえて、本研究では、先ずロボットの移動操作に特化したシステムを構築する。操作対象として、小型飛行船型ロボットを用いたシステムを構築する。飛行船型ロボットは、3次元空間を自由に移動でき、また、慣性の影響が大きいので、コントローラによる操作が複雑で比較的難しい。さらに積載重量が非常に小さいので、ジェスチャ・音声認識などのために、複数のセンサ群を搭載することも困難である。本稿では、このような飛行船型ロボットの特徴を踏まえて、提案システムについて述べる。

2 システム構成

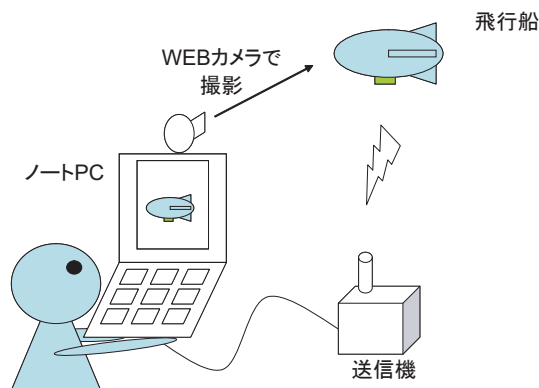


図 4: システム構成

本研究の最終目標はモバイルデバイスによる操作の実現であるが、モバイルデバイスの処理能力の問題から、初期段階としてノートPCを用いてシステムを構築した。システムの構成は図4の通りである。システムは主にノートPC、USBカメラ、送信機の3つから構成される。

ユーザはカメラが固定されたノートPCを上下左右に振ることで、飛行船型ロボットに指示を与える。飛行船は、市販の屋内用ラジコン飛行船(株)タカラドリームフォース02スカイシップ[6]を用いる。飛行船には、カメラから飛行船を認識さ

せるために、両側面にマーカを貼り付ける。このマーカを認識することで、飛行船の相対位置と移動方向を推定する。

3 ソフトウェア構成

ソフトウェアの構成は図5のとおりである。主に画像処理部、行動決定部、命令送信部の3つに分類される。

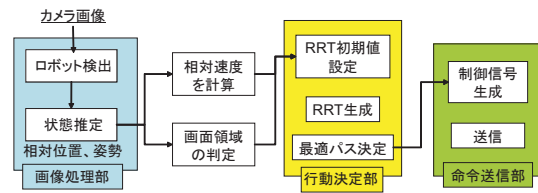


図 5: ソフトウェア構成

カメラから得られた画像は、画像処理部でマーカを検出する（マーカは予め登録しておく）。マーカの形と大きさから現在の相対位置と方向を検出する。この画像処理にはARToolKit[1]を用いる。

3.1 操作の検出

ユーザがモバイルデバイス（ノートPC）を上下左右に振ったときのマーカの相対速度により、ジェスチャ認識を行う。ユーザの振りによるジェスチャは次のルールに従って行う。

1. 飛行船を画面中央に捕らえ、デバイス（ノートPC）を静止させる。
2. 移動させたい方向にデバイスを（速く）振る。

画像処理部で計算された相対位置を時間微分することで、画像中のマーカの相対速度が計算できる。マーカが移動する時とは、1) カメラ（ノートPC）

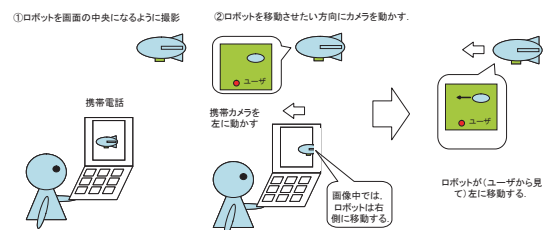


図 6: 操作方法

を移動させたとき、2) 飛行船が移動しているとき、3) 手ぶれによる誤差の3つ場合が考えられる。上記のルールに基づくジェスチャでは、画像中のマーカの相対速度は、図7のような関係を仮定できる。

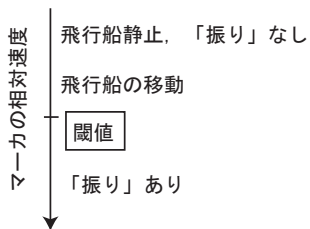


図7: マーカの相対速度の関係

この仮定を利用して、マーカの相対速度がある閾値以下では、ルール1の飛行船の状態を観測している状態であり、閾値よりも大きい場合はルール2のジェスチャーを実行している状態であると判断できる。

さらに操作の認識を簡単にするために、図8に示す画面領域ごとに、それぞれの処理を分ける。

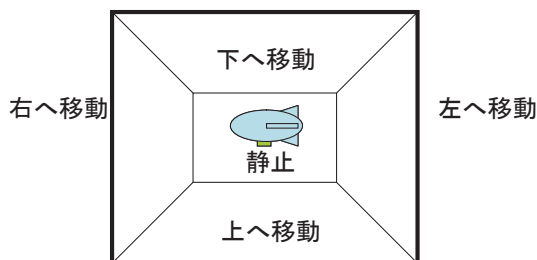


図8: 画面領域によるユーザの操作の判定

画面中央 ルール1の状態。飛行船の状態(位置・方向・速度)を計算する。

画面の端 ルール2の状態。相対速度と画像領域を比較して、ジェスチャであるかを判断する。

4 行動命令の決定

行動命令は、現在の飛行船の状態と、ジェスチャ認識によって得られた移動方向から求まる。

飛行船をある地点からゴールまで導く場合、ゴールに辿り着くまでの経路を求め、その経路を通るように飛行船の行動を逐次変化させる必要がある。

このとき、飛行船は簡単に移動経路を決めることができない。なぜなら、本研究でも用いる飛行船の運動はノンホロノミック系であり、通常の線形制御則によって軌道を生じることができないからである。さらに慣性の影響が大きいことから、その行動は過去の行動命令に大きく依存し、「右に曲がれ」「前へ進め」といった行動命令に対しても、簡単に制御信号を生じることができない。そこで本研究では、Rapidly-exploring Random Tree (RRT)[4]を用いて、飛行ロボットの移動経路とその行動を決めることにする。

4.1 Rapidly-exploring Random Tree (RRT)

RRTは、比較的最近開発された高次元空間でも効率的に探索できるデータ構造である。RRTは、探索空間内の探索木をランダムに伸ばしながらスタートとゴールを結ぶパスを生成する手法である。探索空間の中の全ての点を調べないので、最適解が得られるわけではないが、探索空間が大きな問題には有効である。特にロボットの実時間での経路計画、軌道生成などに使用されている[2]。

表1に基本的なRRT生成のアルゴリズムを示す。

表1: RRT生成アルゴリズム

$RRT_{main}(q_{init})$

1. 初期状態 q_{init} によりツリー G を初期化する。
2. 次のアルゴリズムを1から K まで繰り返す。
3. q_{rand} に $RANDOMCONFIG$ で生成されたランダムな状態を代入する。
4. $EXTEND(G, q_{rand})$ によりツリーを伸ばす

$EXTEND(G, q_{rand})$

1. G 内のノードで q_{rand} に一番近いものを $NEARESTNEIGHBOR(q, G)$ により探し、 q_{near} に代入
2. q_{near} から q_{rand} までの長さが ϵ 以下の時は q_{rand} を q_{new} に、 ϵ 以上の時は q_{near} から q_{rand} に ϵ だけ進んだ方向に q_{new} を作成する。この時障害物などが原因で、 q_{new} が作成できなければ、Trappedを返す。
3. q_{near} と q_{new} を結んで G の一部にする
4. $q_{new} = q_{rand}$ の時は Reachedを、その他の時は Advancedを返す

RRTを飛行船の移動経路生成に用いる場合、各ノードはある時点での飛行船の状態(位置、方向、

速度)をあらわす。初期ノードは、位置を $(x, y, z) = (0, 0, 0)$ とし、移動方向と相対速度は、画像処理部から求められた値をセットする。またゴール地点は、ジェスチャ認識のルール2で求めたマーカの相対速度ベクトルを定数倍した値を用いる。

探索木は飛行船の運動方程式にしたがって、ランダムに枝を伸ばしていく。このとき用いる飛行船の運動方程式は Zwaan の線形近似モデル [7] を用いる。

5 実装と動作検証

本システムを用いて、実際に飛行船の移動操作を行った。スカイシップは図9のように2個のプロペラをそれぞれ独立して回転させることができ、前進、後進、回転をする。また2個のプロペラを支える主軸を回転させることで、上下移動も可能としている。スカイシップ付属のコントローラは、

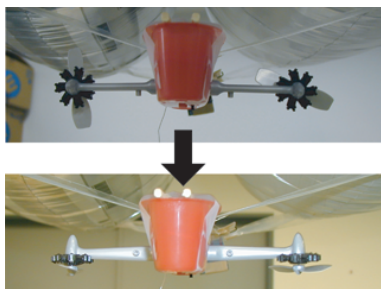


図 9: 試作機

6つの操作ボタンが用意され、それぞれのモータを正回転、逆回転させることができる。本システムではこの6つの入力システムをシリアルコンバータを介して、ノートPCのシリアルポートに接続する。実装したシステムの概観は図10のようになる。

ユーザは図11のようにノートPCを持ち、飛行船をカメラで撮影しながら、操作する。ユーザが飛行船を画面の中央に捕らえると、システムはRRTの初期状態を決定する。さらにこの状態からノートPCを移動させたい方向に振ると、RRTにより移動経路を決定する。図12は、ノートPCを右に振ったときのRRTが生成した移動経路である。現在の状態に応じて適切な経路を生成できていることがわかる。この命令により飛行船はユーザから見て右（飛行船の視点では、斜め右前方）に移動することができた。しかし、移動の“正確さ”に



図 10: 試作機



図 11: 使用の様子

はまだ問題があり、ユーザの意図した“位置”に移動できなかった。また、現状のシステムでは飛行船のスピードを制御できなかったため、思った以上に移動してしまうという問題もあった。

『振り』のジェスチャ認識は、振りの速さや大きさによって、認識し難い場合があった。また飛行船が移動している時のジェスチャ認識は、間違ったジェスチャと認識され易かった。

以上のように解決すべき問題は多々あるが、今回の動作検証から、ユーザの意図した“方向”に、移動させることはできた。これからはユーザの意図した“スピード”で、目的の“位置”に移動・停止させることが課題となる。

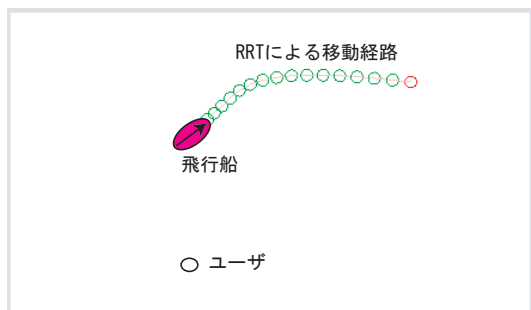


図 12: 右方向への移動命令

6 まとめと今後の展開

本研究では、ロボット操作時の視点の問題に注目して、ユーザが自己の視点に基づいて、ロボットに操作・指示を行えるシステムを提案した。ユーザはモバイルデバイスに搭載されたカメラで、ロボットを捕らえ、移動させたい方向にモバイルデバイスを振ることで、ロボットを移動させるシステムを構築した。本稿では、飛行船を操作対象として、ノートPCを用いてシステムを実装した。ユーザは、本システムを用いることで、飛行船の視点を考慮せずに、任意の方向から直感的に操作することができた。

今後の課題としては、ジェスチャ認識の精度改善、飛行船の制御の改善、応用システムの構築などが挙げられる。ジェスチャの認識に関しては、より柔軟なシステムにするために、オプティカルフローなどを用いて、飛行船の移動量と、カメラの移動量を識別する必要がある。飛行船の制御に関し

ては、特に飛行船が移動している状態で命令を正しく実行できるように、飛行船の運動モデルを修正する必要がある。応用システムについては、他のロボットを操作対象としたシステムの構築や、複数台のロボットを同時制御するシステムの構築を検討している。

また、従来のリモートコントローラによる操作との比較実験を行うことで、提案システムの有効性を検証したい。

謝辞

本研究を行うにあたり、飛行船ラジコン（スカイシップ）を提供して下さった株式会社タカラに感謝の意を表します。

文献

- [1] H. Kato and M. Billingham. Artoolkit. <http://www.hitl.washington.edu/artoolkit/>.
- [2] Jongwoo Kim and James P. Ostrowski. Motion planning of aerial robot using rapidly-exploring random trees with dynamic constraints. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2200–2205, 2003.
- [3] Y. Kuno, M. Sakamoto, K. Sakata, and Y. Shirai. Vision-based human interface with user-centered frame. In *Proceedings of 1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2023–2029, 1994.
- [4] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 473–479, 1999.
- [5] 満上育久, 浮田宗伯, 木戸出正継. 注視指示によるロボットナビゲーション, 2005. 電子情報通信学会 2005 年総合大会, Mar. 2005.
- [6] 株式会社タカラ. ドリームフォース 02 スカイシップ. <http://www.takaratoys.co.jp/skyship/item.html>.
- [7] Sjoerd van der Zwaan, Alexandre Bernardino, and José Santos-Victor. Vision based station keeping and docking for an aerial blimp. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems-IROS'2000*, pp. 614–619, 2000.
- [8] 崎田健二, 小川原光一, 池内 克史, 木村 浩. 視線を利用した人間とロボットの協調作業. 第 21 回日本ロボット学会学術講演会予稿集, Sep. 2003. (東京).