

# Shepherd: ユーザ視点のマルチロボットコントロールを実現するモバイルインターフェイス

Shepherd: A Mobile Interface for Multi-robot Controls from a User's Viewpoint

細井 一弘      杉本 雅則  
Kazuhiro Hosoi      Masanori Sugimoto

東京大学大学院新領域創成科学研究科  
Graduate School of Frontier Sciences, The University of Tokyo

In this paper, we propose a remote control technique for multi-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.

We have developed the proposed technique by using radio-controlled cars and blimps. The preliminary user studies indicated that the proposed technique worked well for supporting users' intuitive manipulations. Several issues to be solved in the future work are discussed.

## 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。

将来的に、ロボットが社会の中に多数存在するようになった場合、様々な人々がオフィスや家庭などの日常生活の中で、ロボットを利用するようになると考えられる。日常生活でロボットを利用するためには、機械や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・インタフェースが必要である。利用するロボットの数は、1 台とは限らず、状況によって複数台のロボットを同時に扱うことも考えられる。そのようなロボットの中には、ヒューマノイドロボットのように人間と同等のインタラクションをとる事が可能な多機能なロボットもあれば、特別なタスクのみを実行する比較的単機能のロボットもあると考えられる。

単機能のロボットは、コストや実用性を考え、必要最低限の機能のみを持っていると考えられる。例えば掃除ロボットは、自律的に移動し、ゴミを発見した時にそれを回収（掃除）する機能だけがあれば十分である。このような単機能のロボットは、人間を認識したり、部屋全体の状況を把握する等の高度な機能は持っていないと考えられる。

face-to-face の環境下で、このような単機能のロボットを人間が直接コントロールするためには、ロボットに新たに環境認識のための十分なセンサ群を追加するか、外部環境にロボットや人間の指示を認識するための様々なセンサを設置する、あるいはそのロボットに特化した複雑なインタフェースを用意する必要がある。しかし、これらの方法では、コストや使用する環境が限定されるなどの問題がある。また、誰でも簡単に導入ができ、使えるようにすることは難しい。

また、単機能のロボットはそのタスクに応じて最適化された様々な形状を持っており、操作において人間の認知や判断に負担をかける場合がある。例えば、3 つ以上のアームを持ったロボットに対して、ユーザは各アームと操作コマンドの対応付け

を覚えておかなければいけない。また、球や円柱の形状にデザインされたロボット等は、ロボットの前後左右の判断が難しいと考えられる。そのため、例えばユーザから見て右方向に移動して欲しい場合、それがロボットから見てどの方向になるのかを瞬時に考えることは難しい。

以上のことから、我々は多種多様なロボットに対して簡単に導入ができ、人間が日常生活の中で直感的に操作できるインタフェースが必要であると考えた。

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“Shepherd”を開発した。ユーザはモバイルデバイスに搭載されたカメラで、ロボットを撮影し、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移動命令を考える必要がなく、任意の方向から自由にロボットを操作可能である。

また、複数台のロボットを同時に撮影することで、任意の状態のロボットを同時に制御することが可能である（図 1 参照）。特定のロボットのみを操作したい場合は、そのロボットだけをズームアップして撮影し、操作することが可能である。

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、直感的な操作で実現できると考えられる。

ロボットや環境にカメラやセンサを搭載して、ジェスチャ・視線・動作などを認識する手法が数多く提案されている。これらの手法では、センサの認識範囲に限界があるため、ユーザは限られた範囲でしかロボットを操作できない。また、飛行ロボットを考えた場合、ユーザとのコミュニケーションのために新たにセンサを追加することは、積載重量の制約から困難である。Shepherd では、カメラがロボットの状態やユーザのジェスチャを認識するので、ロボットにほとんど変更を加えることなく簡単に導入することが可能である。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。

また、音声やコマンド入力によりロボットに指示を伝える手法と比較した場合、既存の手法は「右へ」「前へ」といったはっきりとした方向の指示には対応できるが、あいまいな方向や場所への移動を指示する場合には、複雑な命令をしなければ

連絡先: 細井 一弘, 東京大学大学院新領域創成科学研究科  
〒 277-8561 千葉県柏市柏の葉 5-1-5, Tel: 04-7136-3891,  
Fax: 04-7136-3891, hosoi@itl.t.u-tokyo.ac.jp



図 1: 提案手法

ならない。Shepherd では、デバイスを動かすことで直感的に 3 次元の移動命令を生成することが可能である。

本研究では、提案システムのプロトタイプとして、モバイル PC を用いたシステムを構築した。本稿では、飛行ロボット（飛行船）を操作対象としたシステムについて、システム構成、機能、評価実験について述べる。

## 2. 関連研究

### 2.1 マルチロボットを操作するインタフェース

Kawamura は、ユーザとロボット間で混合主導型対話が可能な Adaptive User Interface を開発した [3]。このシステムでは、マルチエージェントシステムをベースにしているため、様々なロボットに適用可能である。Trouvain らは、ロボットのコンテキストの切り替えをスムーズに行うためのインジケータについて議論している [8]。彼らは、Visual Indicator と Auditory Indicator を用意して、これらのインジケータがある時とないときで、タスクの達成時間やユーザビリティについて評価を行った。Fong らの PdaDriver[2] では、Collaborative Control によりマルチロボットコントロールをサポートしている。Collaborative Control では、ユーザがスーパーバイザーとなってロボットを操作するのではなく、ユーザはロボットの一機能として働く。ロボットは、状況に応じてユーザに質問したり、認識に違いがないかの確認をとる。このようなユーザとの対話を通して、ロボットはタスクを達成する。

以上の研究は、複数のロボットに同じタスクを与える場合、各ロボットの状態に応じて別々の操作を行い、ロボットの状態の提示方法を工夫することで、ユーザの認知負荷を軽減させる手法を採用している。本研究では、複数のロボットを一括して操作することで、ロボットのコンテキストの切り替えや操作の繰り返しによるユーザの負荷を軽減させられると考える。

### 2.2 ユーザ視点での操作に関する研究

進戸らは、ユーザの頭部にカメラを装着し、床面上を走行するロボットに対して、顔の姿勢による注目点とジェスチャに合わせた指示によりロボットを操作する手法を提案している [7]。満上らは、Head Mounted Display(HMD)、カメラ、眼球計測センサをユーザに装着させ、ユーザの視線によりロボットをナビゲーションするシステムを構築している [6]。

以上の研究では、カメラをユーザの頭部に固定し、視線や顔の姿勢で操作ロボットを決定している。本研究では、モバイルデバイスに搭載されたカメラを用いて、カメラを直接動かすこ

操作対象となるロボットを決定する。カメラを直接動かすことで、ロボットの選択範囲をより広くすることが可能であると考えられる。また、HMD などに比べてユーザの視点をデバイスに集中させることが少なく、外部環境を直接見ながら操作できるので、ロボットの状況を把握し易いと考えられる。

## 3. 直感的な 3 次元移動を実現するインタフェース

飛行ロボットの移動操作を実現するインタフェースには、さまざまな問題がある。我々は、特に 1) どのようにして飛行ロボットに 3 次元的な移動命令を与えるか、2) どのようにして正確に指定した位置まで移動させるかの 2 つの問題に着目した。

多くのユーザは、飛行船、飛行機、ヘリコプターなどの航空機の遠隔操作の経験がない。そのため、初心者でも簡単に 3 次元移動操作が可能なインタフェースが必要である。我々は、この問題を解決するために、ジェスチャーベースの操作手法 (Gesture Mode) を採用した。ユーザは、モバイルデバイスに設置されたカメラを動かすことで、飛行ロボットを移動させる。ユーザのカメラの動きを飛行ロボットの動作に割り当てることで、ユーザは直感的に 3 次元の移動命令をロボットに伝えることが可能である。図 2 は、Gesture Mode を使った操作画面である。中央に表示されている三角錐の 3 D オブジェクトが、ユーザのジェスチャを認識した結果の移動方向を示している。

定位置に飛行船を正確に停止させるのは、比較的難しい制御が必要である。Shepherd では、この問題に対処するため Tracking Mode (図 3) を用意している。Tracking Mode では、ユーザが飛行ロボットをカメラで撮影すると、飛行ロボットがカメラ画像の中央に移動する。そして、中央まで移動するとそこで停止するように静止制御を行う。視点方向の停止位置は、ユーザが Tracking Mode に切り替えた時のユーザと飛行ロボットとの相対的な 3 次元位置を記憶しておき、それを基準として制御を行う。また、ユーザは Tracking Mode 時にカメラを少しずつ移動させることで、簡単に正確にロボットを移動させることができる。

各操作モードは、モバイルデバイスのボタンによって切り替えられる。目標の位置まで距離が長いときは、Gesture Mode で操作し、飛行ロボットが目標地点まで近づいたら、Tracking Mode に切り替えて、正確に所定の位置に停止させる。このように、Shepherd では状況に合わせて 2 つの操作モードを切り替えることによって、簡単に飛行ロボットを操作することが可能である。



図 2: Gesture Mode.



図 3: Tracking Mode.

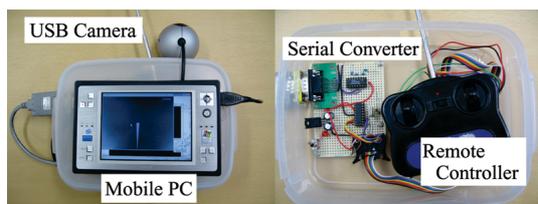


図 4: システム構成

#### 4. システム構成

本研究の最終目標は PDA や携帯電話などの小型のモバイルデバイスによる操作の実現であるが、処理能力の問題から、初期段階としてモバイル 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。システムは主にモバイル 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モバイル PC を上下左右に振ることで、ロボットに指示を与える。ロボットには、ラジコンカーと屋内用ラジコン飛行船を用いる。各ロボットには、図 6 のようにカメラからロボットを認識させるために、上面にマーカを貼り付ける。このマーカを認識することで、ラジコンカーの相対位置と移動方向を推定する。また、飛行船には、図 5 のように側面にマーカを貼り付ける。

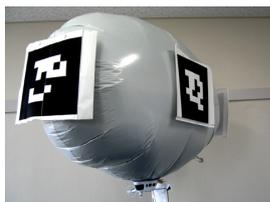


図 5: ラジコン飛行船



図 6: ラジコンカー

ソフトウェアの構成は図 7 のとおりである。主に画像処理部、ジェスチャ認識、行動プランニング、命令送信部の 4 つに分類される。

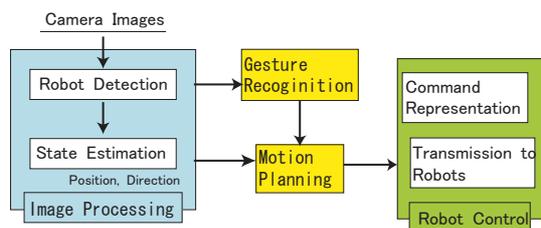


図 7: ソフトウェア構成

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 3 次元の相対位置と方向を検出する。この画像処理には ArTag[1] を用いる。

次に画像処理部から得られたマーカの情報に基づき、ユーザのジェスチャを推定する。この情報とロボットの状態、ユーザとの相対位置関係から、ロボットの行動をプランニングする。命令送信部では、行動プランを基に、モータ制御レベルの命令を生成し、各ロボットに送信する。

#### 4.1 ジェスチャ認識

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. ロボットを画面中央に捕らえ、デバイス（モバイル 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。

ジェスチャを行っている時の（ロボットに添付された）マーカの相対速度を利用して、ジェスチャ認識を行う。マーカが移動する時は、1) カメラ（モバイル PC）を移動させたとき、2) ロボットが移動しているとき、3) 手ぶれによる誤差の 3 つの場合が考えられる。ロボットが低速で移動すると仮定すれば、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、 $1 > 2 > 3$  の関係にあると仮定できる。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 のロボット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 のジェスチャを実行している状態であると判断できる。

#### 4.2 行動プランニング

概要

行動命令は、現在のロボットの状態と、ジェスチャ認識によって得られた移動方向から求まる。

ロボット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ロボットの行動を逐次変化させる必要がある。本研究では、操作対象として飛行船も考えており、また将来的にはヒューマノイドのような複雑なアクチュエータを持つロボットにも対応できるシステムを目指している。そこで、このように様々なタイプのロボットの実時間経路計画に対応できるように、本システムでは、Rapidly-exploring Random Tree (RRT)[5] を用いて、ロボットの移動経路とその行動を決めることにする。

RRT を用いたロボットの移動経路生成法

RRT は、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。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。そのため、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 [4]。

RRT をロボットの移動経路生成に用いる場合、各ノードはある時点でのロボットの状態（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ものをを用いる。新しいノードを作成する際に与える入力値は、ロボットのアクチュエータの制御信号に対応する。飛行船の場合は、プロペラを回転させるモータの ON(正転), ON(逆転), OFF の 3 つの信号に対応し、各プロペラについて、1 つの信号を選択する。

新しいノードの生成には、次の制約を適用させる。あるノードにおいて、離散入力群  $\{U_1, U_2, \dots, U_n\}$  を与えた場合の移動後の状態群  $\{S_1, S_2, \dots, S_n\}$  を推定する。移動後の各状態について、ゴールの方向を向いているかを判定する。ゴールの方向と一致しないものは破棄する。方向が一致する状態群の中から、最もゴールに近づくものを選び、これを新しいノードとして生成する。

#### 4.3 マルチロボットの同時制御

Shepherd では、複数のロボットを撮影することで、対象となるロボットを同時に操作することが可能である。システム

は、得られた画像から、それぞれのロボットの相対位置と状態を認識する。それらを初期状態として、ユーザのジェスチャにあった行動を、それぞれのロボットに対して、プランニングする。以上の処理で得られた行動プランを、それぞれのロボットに送信することで、同時制御が可能となる。

#### 4.4 ズームによる特定ロボットの選択

Shepherd は、1 台のロボットを操作する場合と、複数台のロボットを操作する場合とをシームレスに切り替えるために、ズーム機能を備えている。複数台のロボットがいる環境で、特定のロボットのみを操作したい場合は、そのロボットを拡大表示することで、そのロボットのみを操作することができる。

ロボット全体を撮影し、マウスやカーソルキーを使って、操作対象となるロボットを選択する方法も考えられるが、ロボットの数が多くなると、その切り替え作業が複雑になり、また、ロボットが画面上から外れたときには、再選択する必要もでてくる。

ズーム機能による操作方法では、操作したロボットにカメラを向けて、そのロボットにズームインするだけなので、直感的で簡単な操作方法であると考えられる。



図 8: ズーム機能

## 5. 評価実験

図??に示す試作機を用いて、飛行船の操作の評価実験を行った。提案システムは、Pentium M 1.1GHz Memory 512MB のモバイル PC (Vaio U) を使用し、Visual Studio C++ で開発した。カメラは Logicool 社の Qcam4000 を使用した。キャプチャされた画像は、320x 240 ピクセルサイズで、約 20fps で処理される。

被験者は学生 3 人で、図 1 のようにインタフェースを持ち、飛行船に向けながら操作をしてもらった。また、ラジコン飛行船付属のリモートコントローラを用いて操作してもらい、その操作性について比較してもらった。実験後、被験者に簡単なインタビューを行い、提案システムの操作に対する意見を述べてもらった。実験の結果、被験者は提案システムの使い方をすぐに理解でき、リモートコントローラを用いた場合よりも、簡単に操作できることがわかった。また、問題点としては、「ジェスチャの認識に少し時間がかかる」、「意図してない方向に移動してしまうことがあった」などの意見が得られた。

## 6. おわりに

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、複数台のロボットを同時に操作できるインタフェースを提案した。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。

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。また、応用システムとして、多関節アーム型ロボットを対象としたシステムや、多数の異種ロボットを同時操作するシステムなどの構築を目指す。

## 参考文献

- [1] M Fila. Artag, an improved marker system based on artoolkit. National Research Council NRC/ERB-1111. <http://www.artag.net/>.
- [2] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. In *IEEE Transactions on Industrial Electronics*, Vol. 50, pp. 699–704, 2003.
- [3] K. Kawamura, P. Nilas, and K. Muguruma. An agent-based architecture for an adaptive human-robot interface. In *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2003.
- [4] 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.
- [5] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 473–479, 1999.
- [6] 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.
- [7] 進戸健太郎, 西川敦, 宮崎文夫. ヘッドマウントカメラを用いた指示者の注目点と手振りによる指示を組み合わせたヒューマンインタフェース. *インタラクション 2002 論文集 情報処理学会シンポジウムシリーズ Vol. 2002, No. 7*, 情報処理学会, 2002.
- [8] B. Trouvain and C. Schlick. A study of audio and visual context switch indicators in a multirobot navigation task. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1–6, 2004.