

仮想環境を用いた移動ロボット用 遠隔操作システムの提案

藤田 正典^{1,a)} 青柳 一之¹ 大橋 修¹ 落合 瑛史¹ 常盤 嘉昭¹ 加藤 由花¹

概要：我々はこれまで、大学キャンパス等の構内に設置されたロボットをインターネット経由で自由に遠隔操作し、キャンパスツアーを体験できる移動ロボット遠隔操作システムを提案してきた。ここでは、ユーザがロボットを操作するインタフェースに、ゲームエンジンを利用して構築された仮想環境を利用する。そのため、仮想環境上に存在する仮想ロボットと実環境上の実ロボットの動作を、インターネット上に設置されたサーバ上でマッピングする必要がある。本稿では、この方式を提案するとともに、評価実験により手法の有効性を検証する。

1. はじめに

近年、携帯端末の普及やクラウド化の進展と共に、情報通信とロボット技術の融合が進み、テレロボティクスやクラウドネットワークロボット [1] など、新たなロボットサービスのニーズが高まっている。とりわけ、テレロボティクス、すなわちロボットの遠隔操作は、現在研究が盛んな分野である [2][3]。従来、ロボットは、主に産業用として利用され、近年では、災害救助、原発での無人作業、宇宙探索などの特殊な用途に向けた研究も盛んであるが、一般ユーザ向けの低価格ロボットの登場とともに、これらの一般ユーザ向けに高齢者見守りや案内サービスなどの幅広い用途の遠隔ロボットサービスのニーズが高まっている [4]。

このような背景から、我々はこれまで、大学のキャンパス等の構内に設置されたロボットをインターネット経由で自由に遠隔操作し、キャンパスツアーを体験できる移動ロボット遠隔操作システム ROCS (Remote Open Campus System) を提案してきた [5][6]。ここでは、ユーザがロボットを操作するインタフェースに、ゲームエンジンを利用して構築した仮想環境を利用し、仮想環境上に存在する仮想ロボットと実環境上の実ロボットの動作を、インターネット上に設置されたサーバ上でマッピングしている。

本稿では、ROCS のサービス要件を明らかにした上で、環境側のランドマーク（蛍光灯の照度）を用いたマッピング手法を提案する。また、評価実験により提案手法の有効性を検証する。



図 1 ROCS のサービスイメージ
Fig. 1 Service image of ROCS.

2. ROCS

2.1 サービスの内容

ROCS のサービスイメージを図 1 に示す。ROCS のユーザは、カメラやマイクなどが搭載されたキャンパス内の移動ロボットを、自宅などの遠隔地からスマートフォンやゲーム端末などからインターネット上のサーバを経由して操作し、キャンパス内を移動しながら、その様子を知ることができる。従来のキャンパス案内サービスは、ユーザが実際にキャンパスに出向く必要があり、時間の都合が悪い、場所が遠いなど、ユーザにとって必ずしも便利なものではない。ROCS は、いつでも、どこからでも、ロボットの遠隔操作が容易に実行でき、サービス全体としての操作性や利便性を高めることができるサービスである。

¹ 産業技術大学院大学
Higashi-Ohi, Shinagawa-Ku, Tokyo 140-0011, Japan
^{a)} a1249mf@aait.ac.jp

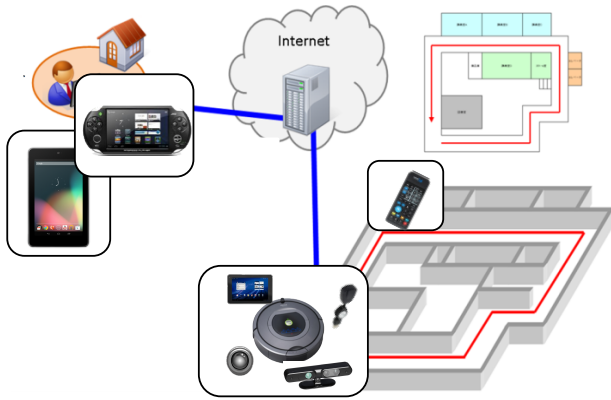


図 2 ROCS のシステム構成

Fig. 2 System architecture of ROCS.

2.2 設計指針

ROCS の設計にあたっては、以下の 3 つの設計指針を策定している。

- ロボット側に複雑な機構を必要とせず、簡易型ロボットを利用できる
- 遠隔操作のための特殊な環境ではなく、大学のキャンパスやショッピングモールなど、一般的な環境での利用が可能である。環境側の制約は可能な限り受けない
- 遠隔地を案内するサービスなので、地理的な前提知識を持たない操作者でも、ロボットを目的地まで遠隔操作できるインタフェースを提供する

これらの設計指針に対応するため、ROCS では、ユーザが遠隔操作する仮想環境と、ロボットが動作する現実世界を分離し、インターネット上のサーバでこれらのマッピングを行う手法を採用している。まず、ロボットとサーバ間で適切に機能分散を行うことで、1 番目の指針に対応する。次に、ユーザインタフェースの記述性に優れ、高い環境シミュレーション構築能力を持つゲームエンジンを活用して遠隔操作の仮想環境を構築することにより、2 番目と 3 番目の指針に対応する。2 番目の指針に対しては、サービス提供前に現実世界の忠実なモデリングが行えるため、環境側へのセンサの設置等の負荷が軽減される。3 番目の指針に対しては、ユーザは仮想環境上での移動、探索、場面遷移等を直感的な操作で行えるようになる。遠隔操作ロボット、サーバ、利用者端末から成る ROCS のシステム構成を図 2 に、仮想環境と実環境のイメージを図 3 示す。

2.3 課題の設定

上記の指針を実現してシステムを実装するために、本稿で解決する課題は以下の 2 点である。

- 課題 1: ロボットの位置情報をどのようにサーバ上にマッピングするか
- 課題 2: サーバ上での仮想的な操作をどのようにロボットの移動命令に変換するか



図 3 ROCS における仮想環境と実環境

Fig. 3 Virtual world and real world on ROCS.



図 4 ロボットのイメージ

Fig. 4 An image of the robot.

3. 提案手法

前章で述べた課題を解決するために、ゲームエンジンを利用して構築した仮想環境（ROCS のユーザインタフェース）と、ロボットの動作する現実世界（ROCS のサービス提供環境）のマッピング方式を提案する。

3.1 想定環境

まず、手法の設計にあたって考察対象となる、ロボットとゲームエンジンについて説明する。

3.1.1 ロボット

ROCS では、入手が容易な簡易型ロボットとして、iRobot 社の Roomba と Android タブレット PC を組み合わせた移動ロボットを想定する。ロボットのイメージを図 4 に示す。ここでは、シリアル通信で接続されたタブレット PC によりロボットをコントロールする。環境側の画像の取得やインターネットを介したサーバへの接続には、タブレット PC 内蔵のカメラや WiFi 機能を利用する。

3.1.2 ゲームエンジン

仮想環境の構築には、ゲームエンジンを用いる。ゲームエンジンは、ユーザインタフェースの記述性に優れ、高い環境シミュレーション能力を持っており、現実の環境を忠実に再現した仮想環境の構築が可能である。また、プログラミングにより仮想環境の柔軟な拡張も可能である。ROCSでは、ゲームエンジンとして、最近のゲームに多用されている Unity を用いることを想定する。

仮想環境の構築にあたっては、ユーザ向けに提供されるサービスの情報（キャンパスにおける図書館の案内情報など）だけではなく、幾何学的な情報（地形や建物内の部屋、廊下の長さ、天井の高さ、窓枠の大きさなど）も設定することにより、実環境と同じスケールで仮想環境が生成される。ユーザ端末上では、仮想環境で操作しやすいカメラアングルを自由に選択することができる。

3.2 マッピング方式

次に、本稿の中心課題である現実環境と仮想環境の処理のマッピング方式について説明する。本稿では、マッピング方式として、環境側のランドマークを用いる方式を採用することにした。具体的には、天井に設置された蛍光灯の光の強弱、及び遠近感を利用し、マッピングを実施する方式を提案する。ここでの前提条件は以下の通りである。

- 蛍光灯の長さと同隔は、ゲームエンジンによるモデリング時に計測し既知である
- ロボットは、天井に蛍光灯が長手方向に対して直角に配置された廊下を移動する
- ロボットは、ある定点を出発点とし、壁と並行に移動する

方式における環境への依存度と、ロボットに要求される機能や自律性の間にはトレードオフの関係がある。本来、マッピングの手法は、可能な限り環境に依存しない方式が望ましいが、今回想定するサービス、ロボットにおいては、事前に環境のモデリングを行う必要があることから、環境への依存度の高い方式を採用することにした。

まず、方式の実現可能性を検証するために、照度データの推移を測定する予備実験を行った。

3.2.1 予備実験

提案手法では、天井に設置された蛍光灯の明るさを計測すると、ロボットの位置の変化に伴って照度データが推移することを利用して、位置推定を行うことを考える。そのため、実際にロボット移動時の照度データを測定し、その実現可能性を検証する。照度データ測定のイメージを図5に示す。

- 実験の内容：ロボットを、あらかじめ決められた走行ルートに沿って移動させ、ロボットに搭載したタブレット PC に内蔵された照度センサを利用して、各地点における照度 (Lux) を測定する。このとき、照度セ

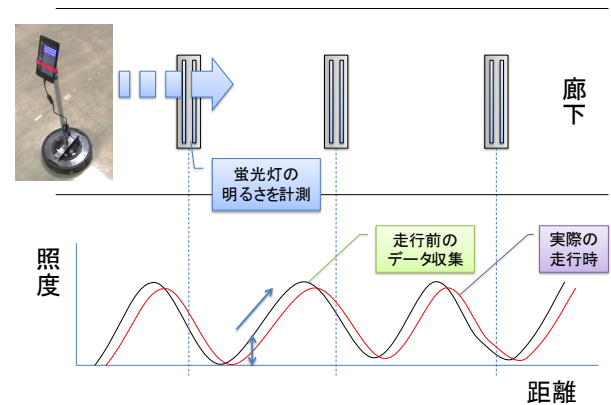


図5 照度データ測定のイメージ

Fig. 5 Measurement of lux data.

ンサによる照度の測定と、遠隔操作の画像データの取得は同一のタブレット PC を用いて行うため、タブレット PC の搭載角度にはある最適な値が存在する。ここでは、事前検証の結果から、この角度を加速度の値として (X: 0.1, Y: 9.2, Z: 3.1) に固定して実験を行うことにした。

- 実験の方法：産業技術大学院大学の5階東側の廊下を測定対象に選定した。外来光や部屋から漏れてくる灯の影響を避けるために、平日夜間19時から22時の間に合計5回の測定を行った。天井には合計5本の蛍光灯が等間隔に設置されているが、これらを全て点灯し、測定者の影響を排除するために、ロボットの移動は離れた場所から遠隔操作で行った。走行ルートは、廊下の端点をスタート地点とし、壁面から67.5 cm の位置 (タイル1枚分) に、壁面に並行になるように設置した。
- 実験の結果：測定結果を図6に示す。X軸は走行ルート上のスタート地点からの移動距離 (タイル番号)、Y軸は照度データ (Lux) である。タブレット PC の搭載角度を固定していることから、照度が極大になるタイル番号は、蛍光灯直下の位置からややずれているが、ロボットの位置と測定される照度の間には相関が見られることから、このデータを利用して、ロボットの位置推定が行えることがわかった

3.2.2 方式の概要

予備実験の結果から、照度データを利用し、ロボットの位置を推定できることがわかった。提案手法では、この推定精度を上げるために、画像処理による蛍光灯の認識を組み合わせることを考える。ロボットを構成するタブレット PC にはカメラが内蔵されているので、ここから取得する画像データを利用できるためである。一方、ロボットの移動時には、姿勢に対する誤差 (仮想環境と現実環境のずれ) が無視できないほど大きいので、この補正が必要であることもわかった。これについては、電子コンパスと傾きセン

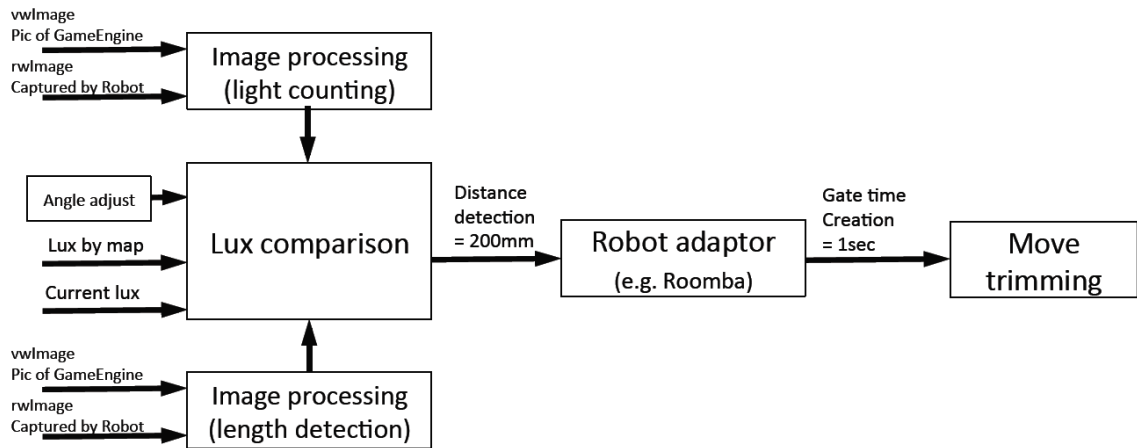


図 7 マッピング方式の概要
Fig. 7 An overview of the mapping method.

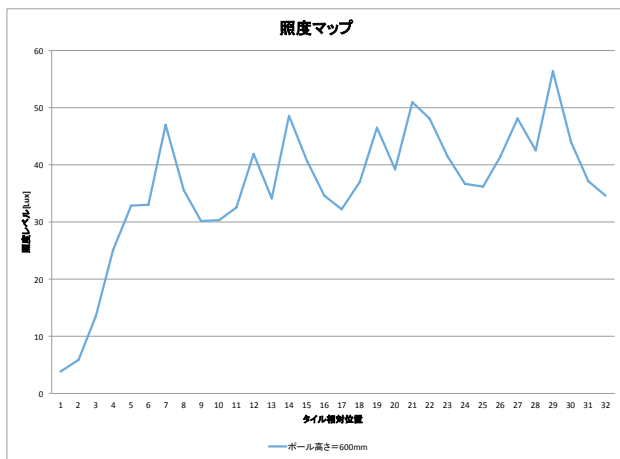


図 6 照度データの測定結果
Fig. 6 A measurement result of lux data.

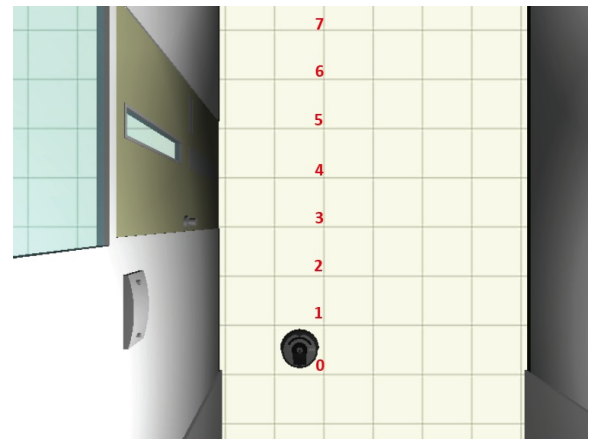


図 8 照度マップテーブル作成のイメージ
Fig. 8 An image of a lux map table.

サを利用し、補正を行う。処理の手順を図 7 に示す。

まず、ロボットは、移動しながら照度センサにより廊下のタイル毎の照度データを測定し、図 8 に示されたように事前に作成されたタイル毎の照度マップと比較することにより現在位置を推定する (Lux comparison)。このとき、電子コンパスによりロボットの姿勢を補正しながら移動を行う (Angle adjust)。位置推定の精度を上げるためには、蛍光管の本数を計測した結果 (Image processing (light counting))、および蛍光管の見え方のデータを仮想環境と実環境間で比較した結果 (Image processing (length detection)) を利用する。これらのデータは、図 9 に示すとおり、ロボットカメラから取得された画像を二値化処理することにより算出する。これらの結果から、仮想環境と現実環境におけるロボット位置の差分を算出し、ロボットを移動させるためのモータ駆動時間を求める (Robot adaptor)。これをロボットに入力してロボットを移動させて、仮想環境と現実環境のロボットの位置を一致させる (Move trimming)。



図 9 ロボットカメラからの取得画像と二値化処理結果
Fig. 9 A robot camera image and the binarized result.

4. 評価

ROCS のプロトタイプシステムを利用し、提案手法の有効性を検証した。

4.1 ロボット側の評価

4.1.1 実装

ROCSの特徴は、インターネット上に設置されたサーバを利用し、ユーザが操作する仮想環境とロボットが動作する実環境を同期させず、操作対象となるロボットが、それぞれの空間で独立に動作する点にある。これにより、ロボットとサーバ間で適切な機能分散を行い、簡易型ロボットの利用を可能にしている。位置推定、ナビゲーション等の処理はサーバ上で行うため、ロボット側では単純な移動機構とセンサ情報の取得機能のみを持てば良い。さらに処理を非同期で行うことにより、仮想環境上の動作と実環境上での動作に時間的なずれが生じることを許容する。簡易型ロボットを利用しながら、位置補正の機能を保持した自律型遠隔操作を実現していることがわかる。

このように、ROCSは適切な機能分散により、簡易型ロボットによる位置推定、ナビゲーションが可能であるが、ROCSに実装されたこれら機能の精度を評価するため、構築したプロトタイプシステムを用いた走行実験を行う。

4.1.2 走行手順

実装したプロトタイプシステムでは、ランドマーク（蛍光灯の照度、蛍光管の本数、蛍光管長）を用いてロボットの現在位置を推定しながら、仮想環境上と実環境上のロボット位置のマッピングを行う。本稿では比較のため、(1) 位置補正を行わない、(2) 照度マップを利用した位置の補正+電子コンパスを利用した姿勢の補正、(3) 照度マップを利用した位置の補正+人手による姿勢の補正、(4) 照度マップを利用した位置の補正+画像を利用した姿勢の補正+電子コンパスを利用した姿勢の補正の4種類の制御方法を実装した。

走行中に位置と姿勢の補正を行いながら、自律走行する手順は以下ようになる。1回の前進処理で走行する距離はタイル3枚分(1,350 mm)とし、目的地に到達するまで移動を繰り返す(ただし初回のみタイル5枚分走行する)。なお、以下の手順は、上記制御手法のうち、4番目の場合の手順である。その他の手法の場合は、実装されない各処理が省略される。例えば、1番目の手順であれば、2、3の処理が省略されることになる。

- (1) 仮想環境上のロボットをスタート地点からタイル5枚分前進させ、停止させる
- (2) 以下のいずれかの手段により、ロボットの姿勢を補正する(進行方向をむかせる)
 - 地磁気センサを利用し、実環境上のロボットの姿勢を補正する
 - 人手でロボットを進行方向に向かせる
 - ロボット上で撮影された蛍光管の画像を利用し、実環境上のロボットの姿勢を補正する。画像に蛍光管が写っていない場合は、地磁気センサで補正する
- (3) 照度マップを利用し、サーバ上で実環境上のロボット

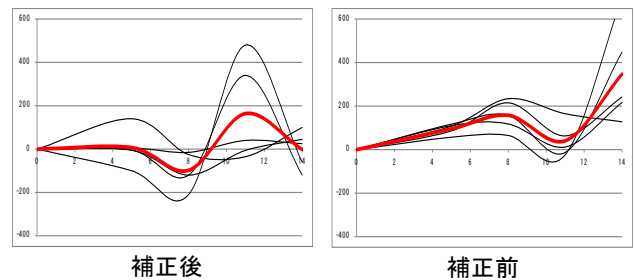


図 10 ロボットの位置のずれ

Fig. 10 An error range of robot position.

の位置を推定する

- (4) サーバ上で補正値を計算し、実環境上のロボットを補正された位置に移動する
- (5) 仮想環境上のロボットを再スタートさせ、タイル3枚分前進させ、停止させる
- (6) 目的地に到達したら終了。それ以外の場合は(2)に戻る

4.1.3 走行実験

実際の廊下においてタイル14枚分(6,300 mm)ずつの走行を5回繰り返し、合計31,500 mmの直線経路を走行する実験を行った。実験は外部光(ノイズ)の影響を避けるために夜間に実施している。なお、今回の実験では、主に位置補正の精度を評価することを目的に、姿勢の補正は人手で行う方式を採用した。提案手法の場合、位置の補正と姿勢の補正は独立に実行されるため、それぞれの手法の精度向上がシステム全体の精度向上につながるためである。

1回の前進処理ごとに、仮想環境上でのロボットの位置と実環境上でのロボットの位置のずれを、サーバ上での位置補正前および補正後に測定した結果を表1に示す。また、これをグラフ化したものを図10に示す。図の横軸はタイルの番号、縦軸は仮想環境上からの位置のずれをmm単位で表したものである。この値は、ずれが全くない場合0であり、仮想環境より前進している場合にプラス、後退している場合にマイナスの値になる。黒い線は各試行ごとの測定値、赤い線は5回の試行の平均値を示す。

ここで、補正前のグラフにおける位置のずれは、タイル3枚分ごとにリセットされていることに注意が必要である。そのため、全く補正を行わずにタイル14枚分ロボットを前進させると、これらのずれは蓄積され、平均641 mm、最大798 mmのずれになる。一方、補正後のグラフにおけるずれは、タイル3枚分ごとにリセットされるので、14枚分前進したときのずれは図10に示されたとおり、平均-2 mm、最大100 mmとなる。これらの結果より、ロボットの移動距離による誤差はあるものの、移動単位ごとに照度マップによる補正を行うことで、誤差が蓄積せず、安定した走行結果が得られることがわかった。

表 1 走行実験の結果
Table 1 An wxperimental result.

No.		5 枚目のずれ	7 枚目のずれ	11 枚目のずれ	14 枚目のずれ
1	補正前	57 mm	65 mm	-24 mm	700 mm
	補正後	-100 mm	-210 mm	480 mm	-60 mm
2	補正前	117 mm	157 mm	-13 mm	448 mm
	補正後	-5 mm	-120 mm	340 mm	-120 mm
3	補正前	86 mm	215 mm	-63 mm	242 mm
	補正後	5 mm	-15 mm	40 mm	25 mm
4	補正前	108 mm	234 mm	165 mm	128 mm
	補正後	140 mm	-25 mm	-35 mm	100 mm
5	補正前	110 mm	118 mm	12 mm	217 mm
	補正後	-5 mm	-120 mm	-6 mm	-45 mm

5. おわりに

本稿では、大学等構内に設置された移動ロボットをインターネット越しに遠隔操作し、自らの自由な操作でキャンパスツアーを体験する ROCS を提案した。ROCS のサービス設計を行うとともに、ROCS を実現するための要素技術として、ゲームエンジンを利用して構築したユーザが操作する仮想世界とロボットの動作する実世界のマッピングを行う方式を提案し、その有効性を検証した。

その結果、まず、ROCS は、ロボットとサーバ間で適切に機能分散を行うことで、簡易型ロボットでの案内サービスをできることがわかった。次に、ゲームエンジンを利用し、現実世界の忠実なモデリングを行うことにより、環境側の制約を限りなく受けないシステムを実現できることがわかった。最後に、仮想環境を利用したインタフェースにより、操作性に優れた遠隔操作システムを実現できることがわかった。

今回、照度マップとロボット画像による単純な補正方式を採用したが、今後、統計データを利用した予測や学習機構を組み込むことにより、マッピング手法の高度化を検討していく予定である。

謝辞 本研究を進めるにあたり、産業技術大学院大学の小川康一さん、三浦彰仁さんから貴重なご意見を頂きました。

参考文献

- [1] 今井倫太, 安西祐一郎: クラウドネットワークロボットへの期待, 電子情報通信学会誌, Vol. 95, No. 12, pp. 1048-1051 (2012).
- [2] 横小路泰義: テレロボティクスフロンティア, 日本ロボット学会誌, Vol. 30, No. 6, pp. 562-564 (2012).
- [3] Tachi, S.: *Telexistence*, World Publishing Company (2009).
- [4] 塩見昌裕, 神田崇行, Glas, D. F., 佐竹 聡, 石黒 浩, 萩田紀博: 複数の案内ロボットが連携してサービスを提供するネットワークロボットシステムの実現, 日本ロボット学会誌, Vol. 29, No. 6, pp. 544-553 (2011).

- [5] 高橋雅彦, 土屋陽介, 成田雅彦, 加藤由花: 移動ロボットを利用した案内サービスの構築を支援するプラットフォーム環境, 情報処理学会論文誌, Vol. 55, No. 2 (2014).
- [6] 大橋 修, 落合瑛史, 加藤由花: ゲームエンジンを用いた移動ロボット遠隔操作手法の提案, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS2013) (2013).