

移動可能な RGB-D センサを用いた 任意視点からの遮蔽物透視システム*

橘高 達也** 藤井 浩光** 山下 淳*** 浅間 一***

Seeing through Obstacles from Arbitrary Viewpoints by Using Movable RGB-D Sensors

Tatsuya KITAKA, Hiromitsu FUJII, Atsushi YAMASHITA and Hajime ASAMA

This paper presents a system which generates images of objects behind obstacles as if they are seen through the obstacles. For remote operation of robots such as disaster response robots, operational efficiency depends heavily on quality of visual information. Diminished reality, which has been proposed in the field of computer vision, is a technique which generates images in which as if users saw objects through obstacles. The proposed method uses a robot with an arm and two RGB-D sensors attached in front of and on the arm of the robot, acquires three-dimensional information of occluded areas and integrates the images to present see-through image, letting the sensor move along with the arm. In this method, past information of RGB-D images can easily be used to expand the field of view with low calculation cost, if the environment is static. The experiments with a manipulator robot show the ability to see through obstacles from arbitrary viewpoints in real time.

Key words: computer vision, diminished reality, RGB-D sensor, disaster response robot, remote operation

1. 序 論

近年、工事現場や災害現場などでの危険な作業において、人の代わりに作業を行うための遠隔操作ロボットのニーズが高まっている。例えば 2011 年に東日本大震災による津波の被害を受けて発生した福島第一原子力発電所事故の際には、原子炉建屋内の情報収集や瓦礫の撤去などに様々な種類の遠隔操作ロボットが使用された¹⁾。

ロボットの遠隔操作においては、オペレータはロボットに搭載されたカメラからの映像を見てロボットを操作し、作業を行うことが一般的である。よって、作業対象物を映像上で適切に認識することは、効率的に作業を行う上で重要である²⁾。作業対象物が遮蔽物に隠れて見えない場合、オペレータは作業対象物の形や位置を映像上で認識することが困難になるという問題が発生する。従来の遠隔操作では、作業環境にカメラを設置すること、あるいはロボットに複数のカメラを搭載することで視野を広げることが一般的であるが、オペレータは複数の映像を見比べながら作業を行うこととなり、高い熟練スキルと集中力が必要となる³⁾。よって、遮蔽物に隠れた領域をオペレータが見やすいように複数の映像を 1 つの映像に統合することは、有効であると考えられる。

コンピュータビジョンの分野では、隠消現実感 (Diminished Reality, DR) と呼ばれる、映像上で視覚的に不要な物体を隠蔽・消去、もしくは障害となる物体を透過させる技術が提唱されている⁴⁾。この技術は、交通安全のための見通しの悪い道路における建物の透視⁵⁾や、動画の補間⁶⁾、Google ストリートビューの画像中からの歩行者の消去⁷⁾といった様々な分野への応用のために広く研究されている。

ロボットの遠隔操作による実際の作業例として、原子力発電

所内の調査の際に障害となるフェンスの反対側にロボットアームを伸ばし、その先端に取り付けたエンドエフェクタにより反対側の根元部分を切断することで、フェンスを撤去するという作業がある。このとき、DR を適用することで、ロボット本体に近い位置からフェンスの反対側を透視可能であれば、エンドエフェクタの位置合わせが容易になると考えられる。このように、遮蔽物に隠れた作業対象物に対するロボットアームの位置合わせなどを行う際、作業に適した位置・姿勢からの映像をオペレータに提示することが望まれる。そのときに視覚的に障害となる物体の透視を可能にすることは、ロボットの遠隔操作において重要な、作業対象物の形や位置を正しく把握するのに有効である。本研究では、先に述べたような作業を想定し、映像中で遮蔽物に隠れて見えない領域を透視可能なシステムを提案する。

2. 先行研究

以下、映像中で遮蔽物に隠れて見えない領域のことを「背景」と呼ぶ。背景の情報の取得には、背景の形や位置などについての事前知識を用いる方法⁸⁾や、周囲の情報をもとに推定する Inpainting と呼ばれる方法⁹⁾などがある。しかし、これらの方法では災害現場などの未知環境で正確な背景の情報を得られるとは限らない。背景が時間的に変化せず、遮蔽物が動いている場合には、過去の映像を用いて背景情報を取得する方法⁶⁾がある。しかし、背景が時間的に変化する場合や、遮蔽物が動かない場合などにも透視可能にするためには、複数のカメラを用いて異なる視点からの映像を統合する方法⁵⁾⁷⁾¹⁰⁾¹¹⁾が有効である。

先行研究の多くに共通する課題は、背景は十分遠方にあるとして平面で近似している点である。ロボットの遠隔操作に應用することを考えた場合、ロボットは作業対象物の近くで作業を行うため、平面で近似された情報では不十分であり、3 次元の情報が必要となる。環境の 3 次元モデルを生成する研究として、KinectFusion に代表される、移動する 1 台の RGB-D セン

* 原稿受付 平成 28 年 5 月 9 日

掲載決定 平成 28 年 11 月 4 日

** 東京大学大学院 (東京都文京区本郷 7-3-1)

*** 正 会 員 東京大学大学院

サを用いて環境の3次元モデルをリアルタイムで生成・更新するシステムがある¹²⁾¹³⁾。RGB-D センサは、通常のカメラと同様のテクスチャ情報に加えて、センサから画像中の各点までの奥行き情報も取得可能なセンサであり、これによって環境の3次元情報を得ることが可能である。しかし、これらの研究は密な3次元モデルを生成することを目的としており、透視を行うことは想定されていない。また、これらのシステムではセンサを十分に移動させる必要があること、複数のセンサを用いることが想定されていないことなど、本研究で対象とする透視の枠組みにそのまま適用するためには課題がある。

Sugimoto らは、3台のRGB-D センサを用いて背景の3次元情報を取得し、遮蔽物の透視を行うシステムを提案した¹⁰⁾。しかし、このシステムには2つの制約が存在する。

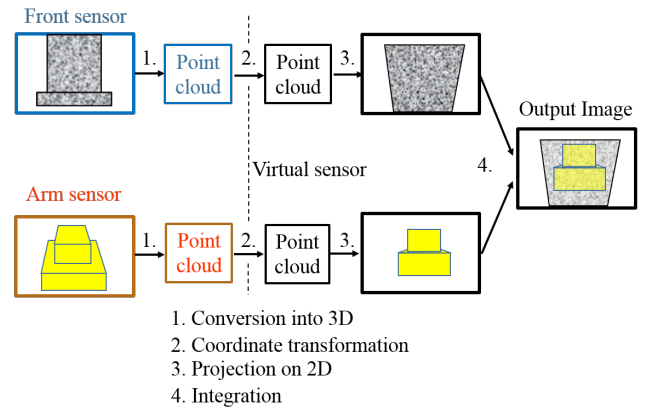
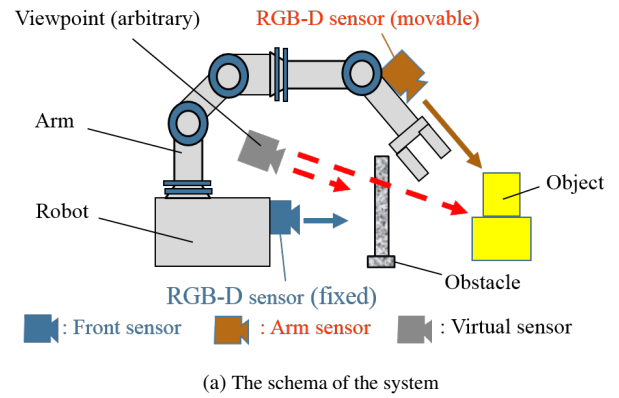
- (1) センサの位置・姿勢が固定されているという制約
- (2) 出力映像を見る視点の位置・姿勢がセンサの位置・姿勢に固定されているという制約

センサが固定されている場合、遮蔽物の位置や形状によっては、複数のセンサを用いたとしても死角が発生し透視が不可能であるような状況が発生するという問題がある。また、出力映像を見る視点の位置・姿勢が固定されている場合、透視が可能であっても映像の奥行き方向の把握が困難な場合があるという問題がある。

Jarusirisawad らは、多数のカメラを用いて3次元の背景を取得した上で、視点を表す仮想的なカメラを定義し、実カメラの間の自由な位置から見た透視映像を生成するシステムを構築した¹¹⁾。このシステムにも、先ほど述べた1番目の制約である、実カメラの位置・姿勢は固定されているという制約が存在する。また、視点を設定可能なのはセンサとセンサの間の限られた領域であり、先ほど述べた2番目の制約である視点の位置・姿勢に関する制約は部分的にしか克服していない。

透視の枠組みにおけるこれらの従来研究で、センサの位置・姿勢や出力映像を見る視点の位置・姿勢が固定あるいは制限されている理由を以下のように分析する。透視映像を生成する際、複数の映像を位置ズレなく統合するためには、センサの位置関係が既知である必要がある。そのため従来研究では、あらかじめセンサの位置・姿勢をキャリブレーションによって求め、その位置・姿勢にセンサを固定することで位置ズレの問題を軽減している。また、従来研究では環境の3次元情報を取得していない、あるいは取得していてもほぼ用いられておらず、実際のセンサの位置・姿勢から大きく離れた視点で透視映像を生成することは不可能である。そのため従来研究では、実際のセンサの位置・姿勢と同じ視点、あるいはそれに近い視点からの透視映像の生成を扱っている。

以上のように、ロボットの遠隔操作のための透視映像生成システムにおいてセンサが移動可能であること、視点が移動可能であることは重要であるが、この2つを満たすシステムは提案されていない。これを踏まえ本研究では、複数の移動可能なセンサを用いて背景の3次元情報を取得し、さらに出力映像の視点も移動可能であるような遮蔽物透視システムを提案する。特にロボットの遠隔操作に適用するためには、透視映像をリアルタイムで生成することが必要である。よって、本研究の目的を「移動可能なセンサを用いた、出力映像の視点を移動可能なリアルタイム遮蔽物透視システムの構築」とする。



(b) The flow chart of the proposed method
Fig. 1 The concept of the proposed method

3. 提案手法

提案手法の概要を図1に示す。本研究では、災害現場などで多く用いられているアームを有するロボットを想定し、図1(a)のようにロボットの正面およびアームにRGB-Dセンサを取り付けたシステムを提案する。基本的な処理の流れを図1(b)に示す。ロボットの正面に取り付けたRGB-Dセンサ（以下、正面センサ）と、アームに取り付けたRGB-Dセンサ（以下、アームセンサ）からの情報を用いて3次元点群を構成し、これを空間中の任意の位置に設定した仮想的なセンサ（以下、仮想センサ）から見た画像を生成する。そして、正面センサ、アームセンサから生成した画像を統合することで、任意視点から遮蔽物を透視したように見える映像を提示する。また、ロボットによる一連の作業において過去に取得したセンサ情報を利用することで、出力映像の視野を拡張することが可能である。

提案手法では、ロボットアームの順運動学を解くことで各センサの位置関係を求めるため、従来の透視の枠組みでは困難であったセンサの移動を可能にする。また、多数の3次元点群の位置合わせなどを行うことなく並列計算によって各点群情報を処理可能であり、これによってリアルタイムでの映像提示を可能にする。これに加え、提案手法は従来手法¹¹⁻¹³⁾で計算の高速化に用いられているGPUを使用せず、CPUのみを用いたリアルタイム処理に適したアルゴリズムとなっている。ロボットの遠隔操作を想定した場合、使用可能な計算機資源には制限がある場合があるため、CPUのみでリアルタイム処理を可能にすることは有用である。

3.1 2次元画像の3次元点群化

事前のキャリブレーションで求めた RGB-D センサの内部パラメータと、センサから得られる距離情報を利用して、3次元点群の情報を生成する。

ピクセル単位で表された画像上の2次元同次座標系 Σ_M (以下、画像座標系) における座標 ${}^M\mathbf{p} = [u, v, 1]^T$ を、mm 単位で表されたセンサを中心とする3次元同次座標系 Σ_S (以下、センサ座標系) における座標 ${}^S\mathbf{p} = [x, y, z, 1]^T$ に変換する。これらの間の関係は、以下の式 (1) のように表される。

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (1)$$

ここで本論文において \simeq は、左辺の0でない定数倍と右辺が等しいことを意味する。また、 f_x, f_y, c_x, c_y は事前キャリブレーションで求めたセンサの内部パラメータである。ここで、点 ${}^M\mathbf{p} = [u, v, 1]^T$ に対応する3次元座標 ${}^S\mathbf{p} = [x, y, z, 1]^T$ のうち、 z は奥行き情報としてセンサから直接取得可能なため、求めるべき未知数は x, y の2つである。よって、式 (1) を x, y について解いた以下の式 (2) によって、画像上の各点の3次元座標を求める。

$$x = \frac{u - c_x}{f_x} z, \quad y = \frac{v - c_y}{f_y} z. \quad (2)$$

3.2 仮想センサ座標系への座標変換

回転行列と並進ベクトルを用いて、正面センサおよびアームセンサ座標系から、仮想センサ座標系へと座標変換する。

座標系 Σ_S から座標系 $\Sigma_{S'}$ へと変換するためには、行列 ${}^{S'}\mathbf{R}_S$ とベクトル ${}^{S'}\mathbf{t}_S$ が必要である。ここで ${}^{S'}\mathbf{R}_S$ は、座標系 Σ_S から $\Sigma_{S'}$ への回転を表す 3×3 の回転行列である。 ${}^{S'}\mathbf{t}_S$ は、座標系 $\Sigma_{S'}$ で見た座標系 Σ_S の原点の位置を表す、 3×1 の並進ベクトルである。

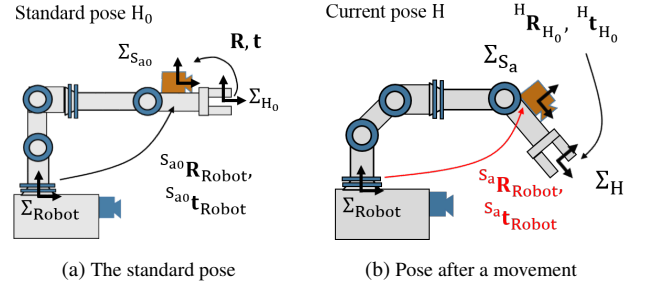
${}^{S'}\mathbf{R}_S, {}^{S'}\mathbf{t}_S$ を求める上で、簡単のためにロボットに固定されたロボット座標系 Σ_{Robot} から任意の座標系 Σ_S への変換を表す回転行列 ${}^S\mathbf{R}_{\text{Robot}}$ と並進ベクトル ${}^S\mathbf{t}_{\text{Robot}}$ を考える。 Σ_{Robot} はロボットアームの根元に設定する、あるいはロボットの外部も含めてロボットに固定された任意の位置・姿勢に設定してもよい。このとき、 ${}^S\mathbf{R}_{\text{Robot}}, {}^S\mathbf{t}_{\text{Robot}}$ は座標系 Σ_S の原点の位置および座標軸の向きに対して一意に定まる。また、 ${}^S\mathbf{R}_{\text{Robot}}, {}^S\mathbf{t}_{\text{Robot}}, {}^{S'}\mathbf{R}_{\text{Robot}}, {}^{S'}\mathbf{t}_{\text{Robot}}$ を用いて、式 (3) および式 (4) により ${}^{S'}\mathbf{R}_S, {}^{S'}\mathbf{t}_S$ を求めることが可能である。

$${}^{S'}\mathbf{R}_S = {}^{S'}\mathbf{R}_{\text{Robot}} ({}^S\mathbf{R}_{\text{Robot}})^T. \quad (3)$$

$${}^{S'}\mathbf{t}_S = {}^{S'}\mathbf{t}_{\text{Robot}} - {}^{S'}\mathbf{R}_{\text{Robot}} ({}^S\mathbf{R}_{\text{Robot}})^T {}^S\mathbf{t}_{\text{Robot}}. \quad (4)$$

さらに、提案手法では変換後の座標系 $\Sigma_{S'}$ は仮想センサ座標系を表し、その位置・姿勢は空間の任意の位置に設定可能である。すなわち、 ${}^{S'}\mathbf{R}_{\text{Robot}}$ と ${}^{S'}\mathbf{t}_{\text{Robot}}$ は任意に設定可能である。従って、正面センサ座標系 Σ_{S_f} およびアームセンサ座標系 Σ_{S_a} から仮想センサ座標系 $\Sigma_{S'}$ への座標変換に必要な回転行列と並進ベクトル ${}^{S'}\mathbf{R}_{S_f}, {}^{S'}\mathbf{t}_{S_f}, {}^{S'}\mathbf{R}_{S_a}, {}^{S'}\mathbf{t}_{S_a}$ を求めるには、 ${}^{S_f}\mathbf{R}_{\text{Robot}}, {}^{S_f}\mathbf{t}_{\text{Robot}}, {}^{S_a}\mathbf{R}_{\text{Robot}}, {}^{S_a}\mathbf{t}_{\text{Robot}}$ を求めればよい。

まず ${}^{S_f}\mathbf{R}_{\text{Robot}}, {}^{S_f}\mathbf{t}_{\text{Robot}}$ について考える。正面センサはロボット本体に固定されており、ロボットに対する位置・姿勢は変化しない。よって、 ${}^{S_f}\mathbf{R}_{\text{Robot}}, {}^{S_f}\mathbf{t}_{\text{Robot}}$ は時間的に変化しない



(a) The standard pose (b) Pose after a movement
Fig. 2 Updating the pose of the arm sensor

め、事前のキャリブレーションによって求めておくことが可能である。

次に ${}^{S_a}\mathbf{R}_{\text{Robot}}, {}^{S_a}\mathbf{t}_{\text{Robot}}$ について考える。アームセンサは、ロボットアームの運動に伴って空間中での位置・姿勢が変化するため、 ${}^{S_a}\mathbf{R}_{\text{Robot}}, {}^{S_a}\mathbf{t}_{\text{Robot}}$ の値は常に更新する必要がある。提案手法では、ある基準となるロボットアームの姿勢 (以下、基準姿勢。図 2 (a) 参照) を定め、そのときのアームセンサ座標系 $\Sigma_{S_{a0}}$ に対して ${}^{S_{a0}}\mathbf{R}_{\text{Robot}}, {}^{S_{a0}}\mathbf{t}_{\text{Robot}}$ をあらかじめ求めておく。また、アームセンサはロボットアームに固定されており、ロボットアームの手先座標系 Σ_H からアームセンサ座標系 Σ_{S_a} への回転行列 \mathbf{R} および並進ベクトル \mathbf{t} は一定であると仮定し、これもあらかじめ求めておく。さらに、ロボットアームの順運動学を解くことで、基準姿勢のときの手先座標系 Σ_{H_0} から現在の手先座標系 Σ_H への移動を表す回転行列 ${}^H\mathbf{R}_{H_0}$ および並進ベクトル ${}^H\mathbf{t}_{H_0}$ を求める (図 2 (b))。このとき、新しいアームセンサの位置・姿勢 ${}^{S_a}\mathbf{R}_{\text{Robot}}, {}^{S_a}\mathbf{t}_{\text{Robot}}$ は、以下の式 (5)、式 (6)、式 (7)、式 (8)、式 (9)、式 (10) と順を追って計算することで求められる。

$${}^{H_0}\mathbf{R}_{\text{Robot}} = \mathbf{R}^T {}^{S_{a0}}\mathbf{R}_{\text{Robot}}. \quad (5)$$

$${}^{H_0}\mathbf{t}_{\text{Robot}} = \mathbf{R}^T {}^{S_{a0}}\mathbf{t}_{\text{Robot}} - \mathbf{t}. \quad (6)$$

$${}^H\mathbf{R}_{\text{Robot}} = {}^H\mathbf{R}_{H_0} {}^{H_0}\mathbf{R}_{\text{Robot}}. \quad (7)$$

$${}^H\mathbf{t}_{\text{Robot}} = {}^H\mathbf{R}_{H_0} {}^{H_0}\mathbf{t}_{\text{Robot}} + {}^H\mathbf{t}_{H_0}. \quad (8)$$

$${}^{S_a}\mathbf{R}_{\text{Robot}} = \mathbf{R} {}^H\mathbf{R}_{\text{Robot}}. \quad (9)$$

$${}^{S_a}\mathbf{t}_{\text{Robot}} = \mathbf{R} {}^H\mathbf{t}_{\text{Robot}} + \mathbf{t}. \quad (10)$$

3.3 出力画像への投影

仮想センサ座標系で表された点群を、仮想センサの位置・姿勢から見たときの画像へと投影する。これは、3.1 節で述べた2次元画像の3次元点群化と逆の処理を行えばよい。仮想センサの内部パラメータを任意に設定し、3次元の仮想センサ座標系 $\Sigma_{S'}$ から2次元の仮想センサの画像座標系 $\Sigma_{M'}$ への変換を行う。

3.4 正面センサおよびアームセンサからの出力映像の統合

正面センサおよびアームセンサから生成した画像をアルファブレンディングによって統合し、透視映像を生成する。なお、正面センサ、アームセンサのうち一方の画像からのみ画素が得られる領域はその画素の色を出力し、正面センサ、アームセンサのいずれの画像からも画素が得られない領域は黒色として出力する。このように、正面センサおよびアームセンサから取得した全ての点群情報を統合し、仮想センサの位置・姿勢から見て重なって見える点群は全て半透明で表示することで透視映像を生成する手法であるため、提案手法では透視対象物の形状などを仮定する必要がない。

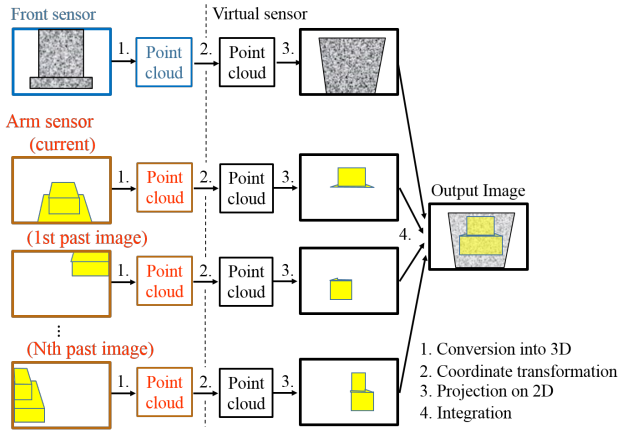


Fig. 3 Superimposing past information of the arm sensor

3.5 過去のアームセンサ情報の重畳表示

3.1 節から 3.4 節までの処理により、正面センサ・アームセンサから得られる現在の環境の情報を統合し透視映像を生成することが可能である。しかし、センサの画角などの制約およびセンサと環境との位置関係によっては、これらの情報だけでは十分な視野を得られない場合がある。そこで、1 章で述べたような作業では環境は短時間で大きく変化しないと考え、ロボットによる一連の作業において過去に取得した環境の情報を用いて視野を拡張する手法を提案する。特に、アームの運動に伴って視野を移動させることが可能なアームセンサからの情報を記録し、現在の仮想センサにおける出力映像に重畳することが有効である。

ただし、過去の全ての点群情報を記録して重畳表示することは非常に計算コストが大きいため、ここではオペレータが出力映像を見ながら点群情報を記録する瞬間を指定するという手法、すなわちオペレータがキー入力などによる指示を出した瞬間における点群情報のみを記録するという手法を用いる。また、センサ情報から生成した 3 次元点群を構成する各点は、3 次元座標 x, y, z とともに色成分 I_r, I_g, I_b の情報も持っており、これら 6 つの値の組の集合として点群を記録するとデータのサイズが膨大になる。よって、点群を構成するために必要な最低限の情報のみを記録することが重要である。さらに、処理の高速化のためには、アルゴリズムを可能な限り並列計算に適した構造にする必要がある。

過去のセンサ情報を重畳する手法の概要を図 3 に示す。オペレータが指定した瞬間において記録する必要があるのは、アームセンサの RGB-D 情報と、ロボット座標系からアームセンサ座標系への回転行列と並進ベクトル ${}^{S_a}\mathbf{R}_{Robot}, {}^{S_a}\mathbf{t}_{Robot}$ の組である。すなわち、3 次元座標 x, y, z と色成分 I_r, I_g, I_b の 6 つの値の組ではなく、色成分 I_r, I_g, I_b と奥行き d の 4 つの値の組として過去の点群情報を記録することでデータのサイズを抑える。これらを用いて、3.1 節から 3.3 節までと同様の処理を行うことで、2 次元の出力画像を生成可能である。3.1 節から 3.3 節の処理の中で一時的に 3 次元点群化および座標変換を経由するものの、3.1 節から 3.3 節を通して「2 次元の入力画像の位置 (u, v) から 2 次元の出力画像の位置 (u', v') へのマッピング」となるように実装することが可能である。これは、点群情報が 3 次元座標 x, y, z と色成分 I_r, I_g, I_b の 6 つの値の組として記録されている瞬間が存在せず、従来手法⁽¹²⁾⁽¹³⁾と比較してデータのサイズ

を抑えることが可能であることを意味する。アームの位置・姿勢が異なる複数の瞬間におけるアームセンサの情報の組を記録しておくことで、任意の枚数の異なる視野を有する出力画像を生成可能である。各センサの位置関係が既知であるため、多数の 3 次元点群の位置合わせなどを行うことなく並列計算によって各画像を生成可能である。また、GPU を用いた高速化手法で用いられる画素単位での並列計算ではなく、画像単位での並列計算であるため、並列化に必要なスレッド数が少なく、CPU での並列計算が容易である。これらの画像を 3.4 節と同様に統合する。ただし、ここでは 3 枚以上の画像をアルファブレンディングによって統合する必要がある。過去の点群情報を記録した回数を N としたとき、現在の正面センサ・アームセンサから生成した RGB 画像に 1 番・2 番、過去の点群情報から生成した RGB 画像に 3 番、4 番、 \dots , $N+2$ 番と番号をつけると、アルファブレンディングの式は式 (11) のように表される。

$$\mathbf{I}_{out}(u, v) = \frac{\sum_{i \in S(u, v)} \alpha_i \mathbf{I}_i(u, v)}{\sum_{i \in S(u, v)} \alpha_i}. \quad (11)$$

ここで、 $\mathbf{I}_i(u, v)$ は i 番目の画像における画素 (u, v) の RGB 値、 α_i は i 番目の画像の不透明度を表すパラメータ、 $\mathbf{I}_{out}(u, v)$ は仮想センサの出力画像の画素 (u, v) の RGB 値を表す。また、 $S(u, v)$ は、画素 (u, v) の RGB 値が得られている画像の番号の集合である。なお、どの画像も画素 (u, v) の RGB 値が得られなかった場合、すなわち $S(u, v)$ が空集合の場合は $\mathbf{I}_{out}(u, v)$ には黒色を割り当てる。

なお、本手法ではロボットによる一連の作業中に環境は大きく変化しないと仮定しており、過去の情報を用いる 3 番目から $N+2$ 番目までの画像については作業の進行に伴う環境変化には対応できない。実際には、一連の作業中に環境は大きく変化しないとしても、作業の結果により作業開始時と作業終了時で環境が異なることは十分考えられるが、この場合には以下のような対応が考えられる。本手法において生成した 3 番目から $N+2$ 番目までの画像データは、任意のタイミングでシステム上から消去可能である。作業の結果により環境が変化する場合、作業を行ったということをオペレータ自身が把握していると考えられるため、オペレータ自身の判断でこれらの画像を全て消去し、必要に応じて再度アームセンサで環境の情報を取得することで、作業後の環境の透視映像を生成可能である。

3.6 アームの運動指令生成

本システムにおいてアームを運動させる重要な目的の 1 つとして、アームとともにアームセンサを運動させることで、運動前の位置・姿勢からでは観測が困難だった領域を観測するという点がある。本節では、環境中のある特定の部分をアームセンサで観測するためのアームの運動指令を生成する手法について述べる。ここでは、センサの画角の制限により観測が困難な領域を観測する場合を想定し、以下のような問題を考える。

- (1) 3 次元空間中の任意の 1 点を観測対象として指定する
- (2) 指定された点をアームセンサの画像の中心に捉えるようにアームを運動させる
- (3) アームセンサは並進運動をせず、ヨー・ピッチの回転のみ行う

図 4 にこの問題設定の概要を示す。すなわち、指定された点がアームセンサ座標系 Σ_{S_a} で $[x, y, z, 1]^T$ と表されるとき、こ

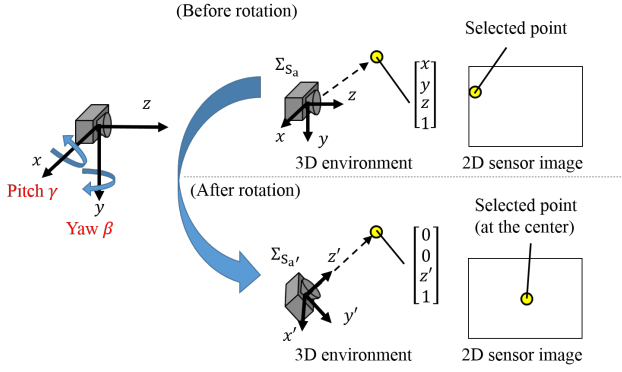


Fig. 4 Rotation of arm sensor to capture the selected point

の点が回転後のアームセンサ座標系 Σ_{S_a} で $[0, 0, z', 1]^T$ に見えるようなアームセンサのヨー角 β 、ピッチ角 γ を求める問題に帰着される。これらは、以下の式 (12) に示す方程式を満たす。

$$\mathbf{X}(-\gamma)\mathbf{Y}(-\beta) \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z' \end{bmatrix}. \quad (12)$$

ただし、 $\mathbf{X}(\theta)$ 、 $\mathbf{Y}(\theta)$ は以下の式 (13) で定義される回転行列である。

$$\mathbf{x}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \mathbf{Y}(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (13)$$

これを解くと $\beta = -\text{atan2}(z, x) \pm \frac{\pi}{2}$ が得られる。ここで $\text{atan2}(y, x)$ は $-\pi$ から π までの値域をもち、座標平面上において x 軸の正の方向とベクトル (x, y) のなす角を表す関数である。また、得られた β を用いて

$$\mathbf{Y}(-\beta) \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (14)$$

とおき、式 (12) に代入することで γ についての方程式である式 (15) が得られ、 γ が求められる。

$$\mathbf{X}(-\gamma) \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ z' \end{bmatrix} \quad (15)$$

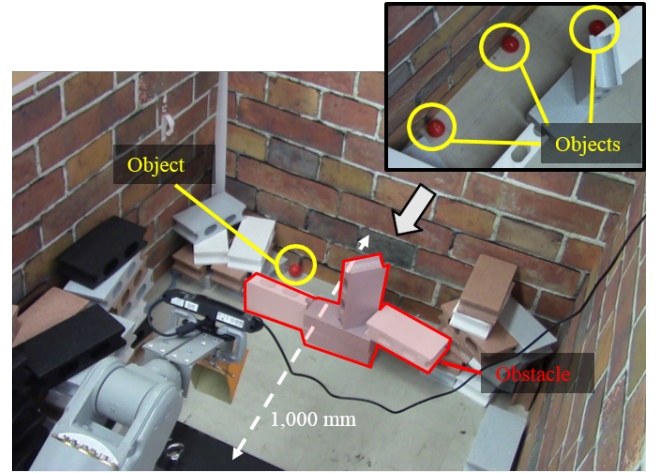
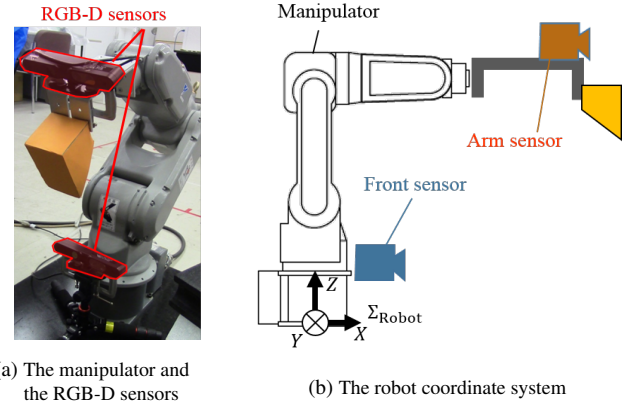
$$\Leftrightarrow \gamma = -\text{atan2}(y'', z'').$$

なお、 β の値が複号で 2 つ得られたのは、ヨー・ピッチをともに π だけ回転すると元の方向に戻る（ただし、元の姿勢に対してロール角 π だけ回転した姿勢となる）ことによるものである。よって、2 つの解のうち絶対値が小さい方を β の値とし、これを用いて γ の値を求める。

4. 実機による検証および定量評価

4.1 実験環境

提案手法の有効性を検証するために実機実験を行った。実験環境を図 5 に示す。本実験では、図 5 (a) 中に示す 2 台の RGB-D センサ (ASUS: Xtion Pro Live) とマニピュレータ (YASKAWA: MOTOMAN-HP3J) を用いた。図 5 (b) に、ロボット座標系の定義を示す。なお、提案手法では全ての座標系がロボットに固定されているため、アームを有するのあれば本実験のように固定式の



(c) An obstacle and the objects seen from different angles

Fig. 5 The experimental environment

ロボットではなく移動式のロボットでも 3.1 節から 3.4 節までの処理によって透視映像を生成可能である。しかし 3.5 節の処理に関しては、環境の変化がないことを前提とするためロボットの移動がない場合のみ適用可能である。実験環境には図 5 (c) 中に示すように、遮蔽物と、透視する対象である直径 50 mm の小球がある。図 5 (c) の左下の画像では小球は 1 個しか見えないが、右上に示した別の角度からの画像で確認できるように、全部で 3 個の小球が置かれている。正面センサから遮蔽物までの距離は約 700 mm であり、正面センサから実験環境の最奥にある壁までの距離 (図 5 (c) 中に点線矢印で示した) は約 1,000 mm である。マニピュレータの関節角の情報の取得レートは 30 Hz であり、RGB-D センサの取得映像のサイズは 640×480 pixel、フレームレートは 30 fps とした。事前のキャリブレーションで求めるべきパラメータのうち、 ${}^{S_f}\mathbf{R}_{\text{Robot}}$ 、 ${}^{S_f}\mathbf{t}_{\text{Robot}}$ 、 ${}^{S_{a0}}\mathbf{R}_{\text{Robot}}$ 、 ${}^{S_{a0}}\mathbf{t}_{\text{Robot}}$ は、一般的にカメラの内部パラメータ推定に広く用いられる Zhang の手法¹⁴⁾ に基づくチェスボードパターンを用いたパラメータ推定により求めた。また、ロボットアームの手先座標系 Σ_H からアームセンサ座標系 Σ_{S_a} への回転行列 \mathbf{R} および並進ベクトル \mathbf{t} は実測により求めた。特に回転行列 \mathbf{R} については、任意の座標軸まわりの 90° 回転といった単純な回転のみでほぼ正確に表せるようにアームセンサを固定することで、各成分が 0, 1, -1 のいずれかであるような回転行列を設定した。仮想センサの内部パラメータは正面センサと同じ値を使用した。また、3.4 節および 3.5 節で述べた画像の統合処理は各画像が均等に統合されるように設定した。

4.2 生成された透視映像

実験の結果、得られた出力画像を図 6 に示す。図 6 (a) は、ある瞬間における正面センサおよびアームセンサからの入力画像である。正面センサおよびアームセンサからの画像では、小球はそれぞれ 1 個および 2 個を確認できるが、全ての小球を一度に確認することは不可能である。図 6 (b) は、この瞬間における仮想センサの出力画像である。正面センサ、アームセンサそれぞれ単独では見えなかった小球 (図 6 (b) 中に円で囲んで示した) が透視できていたことが分かる。

また、仮想センサの位置・姿勢を移動させた場合の出力画像の例を図 6 (c) に示す。このときの仮想センサの位置は、アームセンサの初期位置に対してロボット座標系 Σ_{Robot} で $(X, Y, Z) = (200 \text{ mm}, 200 \text{ mm}, 200 \text{ mm})$ だけ移動した位置であり、仮想センサの姿勢はアームセンサの初期姿勢から Y 軸まわりに 20° 回転した姿勢となっている。仮想センサの位置・姿勢は、マニピュレータの操作中でも任意に移動することができ、様々な視点から透視映像を生成することが可能である。

4.3 過去のアームセンサ情報の重畳表示、アームの運動生成

3.5 節で述べた過去のアームセンサ情報の重畳表示および 3.6 節で述べたアームの運動指令生成を行った結果を図 7 に示す。

図 7 (a) は、現在の正面センサ・アームセンサの情報のみを用いて生成した仮想センサの出力画像の例である。この画像では、情報が得られずに黒く塗りつぶされた領域が目立つ。画像の右側に写っている黄色い点は、アームセンサの運動指令生成のために指定した点を示している。

図 7 (b) は、図 7 (a) におけるアームセンサ情報を記録した後で、図 7 (a) に写っている黄色い点を指定してアームを移動させたときの仮想センサの出力画像である。指定した点を中心とした領域がアームセンサによって捉えられ、出力画像の右側の領域が新たに視野に含まれたことが分かる。また、現在の正面センサ・アームセンサの情報および図 7 (a) のときのアームセンサの情報を使用しているため、図 7 (a) と比較して図中右下部に広い視野が得られている。

図 7 (c) は、さらに図 7 (b) におけるアームセンサ情報を記録した後で、別の点を指定してアームの運動指令を生成したときの仮想センサの出力画像である。この段階では、現在の正面センサ・アームセンサの情報および過去 2 回分のアームセンサの情報を使用しているため、図 7 (b) と比較して図中左上部にさらに広い視野が得られている。

図 8 は、図 7 (c) において仮想センサの位置・姿勢を変えて生成した出力画像である。広い視野を保ったまま、異なる角度から環境を画像上で認識可能である。

4.4 評価

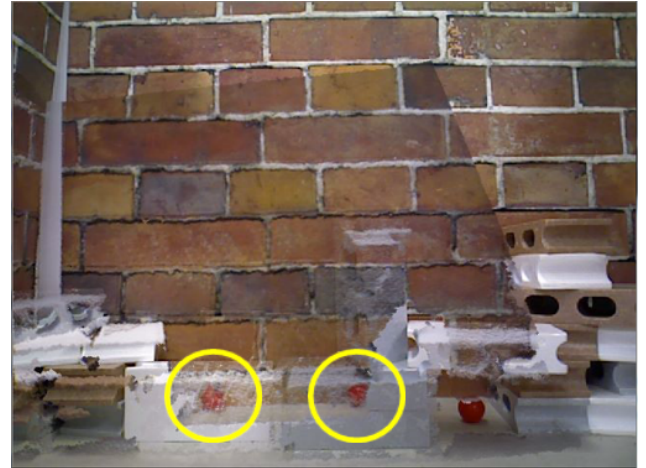
4.4.1 画像の位置ズレ

複数のセンサ情報を統合する際に、互いの画像の位置にズレが生じてしまう場合がある。特に図 7 (c) のように多数のセンサ情報を統合した際にそれが顕著になる。環境を正確に画像上で認識するためには、位置のズレは小さいほどよい。

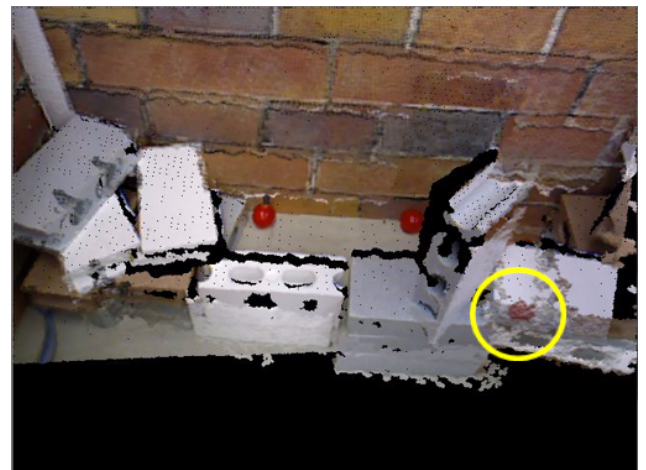
今回の条件における画像の位置ズレを定量的に評価するため、以下のような実験を行った。概要を図 9 に示す。図 9 (a) に示すマーカを、図 9 (b) のように環境中に置き、透視映像を生成した際のその位置のズレを目視で評価した。3 次元的な位置のズレを評価するために、図 9 (c) のように異なる 2 つの視点から透視映像上の位置のズレを評価した。視点 1 は、図 9 (a) に示



(a) Input images from the front sensor (left) and the arm sensor (right)



(b) The output image generated from (a) where the virtual sensor is set at the same position of the front sensor



(c) The output image generated from (a) where the virtual sensor is set at an arbitrary position

Fig. 6 The images acquired by the experiments

す x_1 軸、 y_1 軸と直交する向きから、 x_1 軸、 y_1 軸方向の透視映像上でのマーカの位置ズレを評価した。視点 2 は、図 9 (a) に示す x_2 軸、 y_2 軸と直交する向きから、 x_2 軸、 y_2 軸方向の透視映像上でのマーカの位置ズレを評価した。アームの手先は図 5 (b) に示したロボット座標系の定義において x, y, z の各方向に 150 mm 程度運動してから元の位置・姿勢に戻る、事前に与えた運動を行った。アームの手先の軌跡を図 10 に示す。図 10 (a) は、アームの手先の座標を表す。図 10 (a) は、アームの手先の姿勢を、 $z-y-x$ オイラー角による角度 α, β, γ で表す。また、位置ズレは「正面センサ映像を基準としたときのアームセンサ映像の位置」とし、1 秒ごとに目視およびマーカにより 5 mm 単位で測定した。

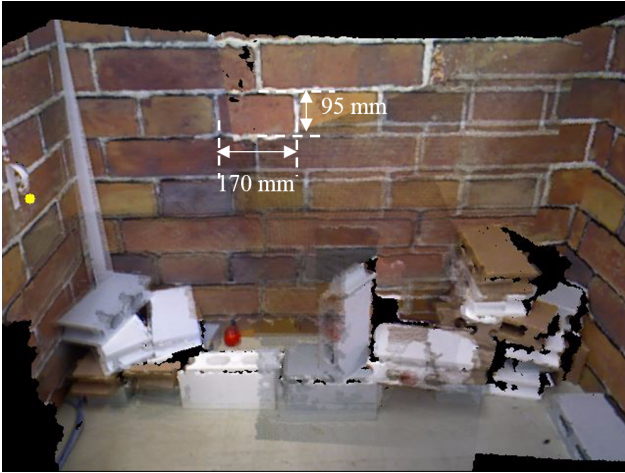
位置ズレの評価結果を図 11 および図 12 に示す。図 11 は、



(a) Before superimposition



(b) After recording the arm sensor's RGB-D image in (a) and moving the arm by assigning the yellow point



(c) After recording the arm sensor's RGB-D images in (b) and moving the arm by assigning another point

Fig. 7 Output images with past RGB-D images superimposed

視点1から見た x_1 軸, y_1 軸方向の位置ズレを表す。これは正面センサから見て上下左右方向のズレであり、最大で 20 mm であった。図 12 は、視点2から見た x_2 軸, y_2 軸方向の位置ズレを表す。こちらは正面センサから見て奥行き方向のズレが y_2 軸方向に含まれており、最大で 35 mm であった。 y_2 軸の 35 mm のズレが全て奥行き方向のズレに起因するもの場合、奥行き



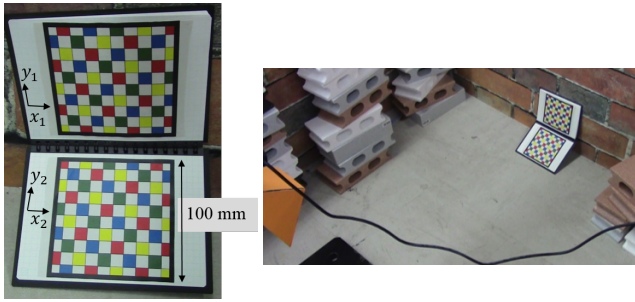
Fig. 8 Superimposed image from a different angle

方向のズレは $35 / \cos 30^\circ = 40$ mm 程度であるといえる。この結果から、位置のズレは 1000 mm 遠方で 50 mm 以内、すなわち小球の直径以内であるといえる。

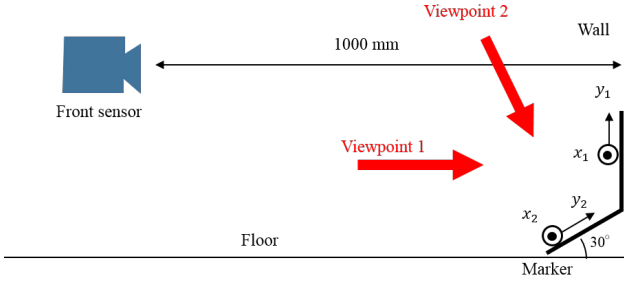
今回の実験では小球に対して遮蔽領域は比較的大きいため、映像上で物体の位置を把握するという観点においてこの位置ズレの大きさは許容できると考えられる。

今回の実験における位置ズレの原因は、大きく分けてアームの絶対位置誤差とセンサのキャリブレーション誤差の2つに分類できると考えられる。アームの絶対位置誤差に関して、図 11 および図 12 に示した位置ズレの評価結果には、アームの運動に起因すると考えられる位置ズレの変化が見られた。時刻 $t=5$ s, 10 s, 15 s, 23 s, 27 s の前後などで、図 11 および図 12 で位置ズレが急激に変化している部分があり、これは図 10 が示すように、アームが静止した状態から移動し始める瞬間およびアームが静止する直前のタイミングとほぼ一致している。特に $t=15$ 前後での y_1 の変化が 35 mm と最も大きく、この変化の大きさは位置ズレ自体の大きさと同程度であるため、アームの絶対位置誤差は画像の位置ズレの支配的な要因の1つであると考えられる。

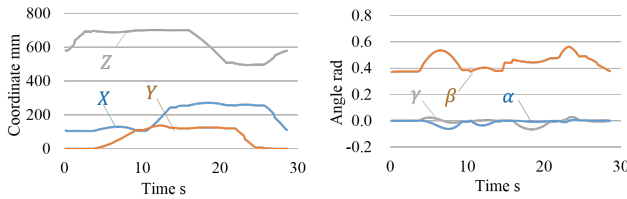
センサのキャリブレーション誤差は、具体的にはキャリブレーションによって推定された各種パラメータの不確かさが位置ズレに影響している。キャリブレーションを繰り返し行って得た各パラメータの平均値・標準偏差を求め、平均値を真値、標準偏差をばらつきと見なすことで、座標変換後の3次元位置の不確かさを求めた。簡単のために、3.2節で述べた基準姿勢におけるアームセンサ座標系 Σ_{S_a} から正面センサ座標系 Σ_{S_f} への座標変換を想定し、内部パラメータ f_x, f_y, c_x, c_y およびアームセンサ座標系から正面センサ座標系への回転行列 ${}^{S_f}R_{S_a}$ と並進ベクトル ${}^{S_f}t_{S_a}$ のみに着目した。これらについて 10 回キャリブレーションを行って得た値の平均値・標準偏差を表 1 および表 2 に示す。なお、回転行列 ${}^{S_f}R_{S_a}$ のばらつきは並進ベクトル ${}^{S_f}t_{S_a}$ のばらつきに対して 3 桁以上小さく、座標変換後の3次元位置への影響は無視できるため省略した。これらのばらつきは微小であるとして、座標変換後の3次元位置のばらつきを各パラメータの1次式の形で定式化したところ、正面センサから 1,000mm 遠方において座標変換後の3次元位置データの約 95%が、35mm 程度以内のズレを持つという計算結果が得られ



(a) The marker and the coordinate for position error evaluation (b) The marker in the experimental environment



(c) The experimental setup and the viewpoint for position error evaluation
Fig. 9 The experimental environment of position error evaluation



(a) The time series data of X, Y, Z (b) The time series data of α, β, γ
Fig. 10 The trajectory of the end effector

た. これは, 図 11 および図 12 で得られた位置ズレの大きさと同程度であったため, キャリブレーション誤差は画像の位置ズレのもう 1 つの支配的な要因であると考えられる.

なお, RGB-D センサによる距離情報の計測結果には歪が発生することが知られており, 近年この歪の補正に取り組む様々な研究が行われている¹⁵⁻¹⁷⁾. しかし本実験では, RGB-D センサから環境までの距離は 1,000mm 程度と小さく, 文献¹⁷⁾によると, この距離における計測歪は十分小さい. よって, RGB-D センサの計測歪の影響は, アームの絶対位置誤差およびセンサのキャリブレーション誤差と比べて十分小さいと考えられる.

4.4.2 フレームレート

また, 仮想センサの出力映像のフレームレートは, 提案システムのリアルタイム性を表す重要な指標である. 一方, 出力画像の中で色情報が得られた領域 (黒く塗りつぶされていない領域) の割合は, 視野の広さを表す指標となる. 図 7 における仮想センサの位置・姿勢において, 透視映像を生成するために使用する RGB-D 画像の枚数を変えた場合のこれらの指標を評価した. なお, 本システムにおける画像処理用 PC が使用しているプロセッサは, Intel Core i7-3770 (クロック数 3.40 GHz, コア数 4, スレッド数 8) であり, 図 3 に示したように各 RGB-D 画像の処理を並列計算した.

出力映像のフレームレートおよび色情報が得られた領域の割合の評価結果を, 図 13 に示す. 提案システムでは正面センサ・

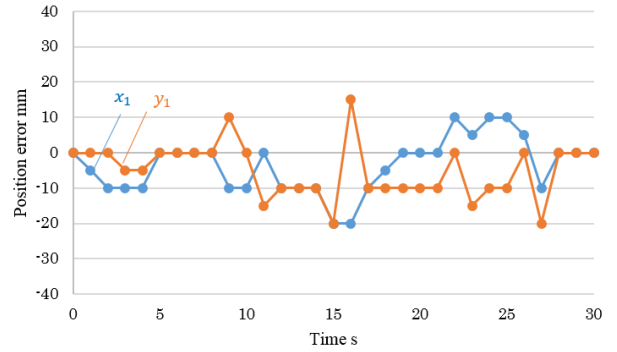


Fig. 11 The position error from viewpoint 1

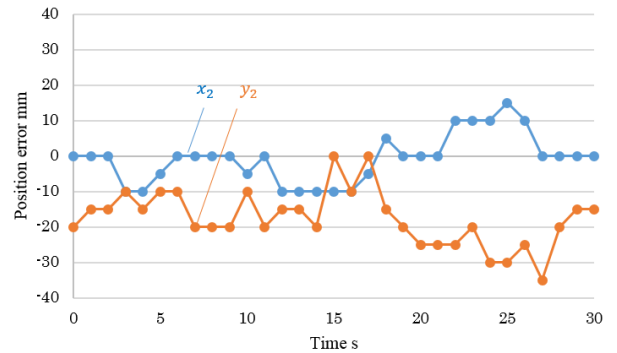


Fig. 12 The position error from viewpoint 2

Table 1 Average and standard deviation of internal parameters

	Avg. pixel	S.D. pixel
f_x	526	9.4
f_y	527	7.8
c_x	307	3.1
c_y	228	7.1

Table 2 Average and standard deviation of translation vector

	Avg. mm	S.D. mm
x component	13.4	7.1
y component	-367	5.0
z component	74.9	2.6

アームセンサからの RGB-D 情報を用いるため, 通常は最低でも 2 枚の RGB-D 画像を使用する. 図 13 では, 参考のために正面センサからの RGB-D 画像 1 枚のみで透視映像を生成した場合の評価結果も示した. また, 並列計算を行わず, 全ての処理を直列で行った場合のフレームレートも比較のために示した.

フレームレートについては 10 秒間にわたって記録したデータの平均を示し, エラーバーはデータの標準偏差を表す. 結果は, 並列計算を行った場合, 使用する RGB-D 画像の枚数が 2 枚 (現在の正面センサ・アームセンサの RGB-D 画像を使用) のときは平均 10.2 fps, 6 枚 (現在の正面センサ・アームセンサの RGB-D 画像と過去のアームセンサの RGB-D 画像 4 枚を使用) のときは平均 7.9 fps となった. 並列計算を行わない場合と比較すると, 使用する RGB-D 画像の枚数が 3 枚以上の場合にフレームレートの向上が見られた. 特に使用する RGB-D 画像の枚数が 6 枚のとき, 並列計算を行わない場合は 3.6 fps であるのに対し, 並列計算を行った場合は 7.9 fps と, 2 倍以上のフレー

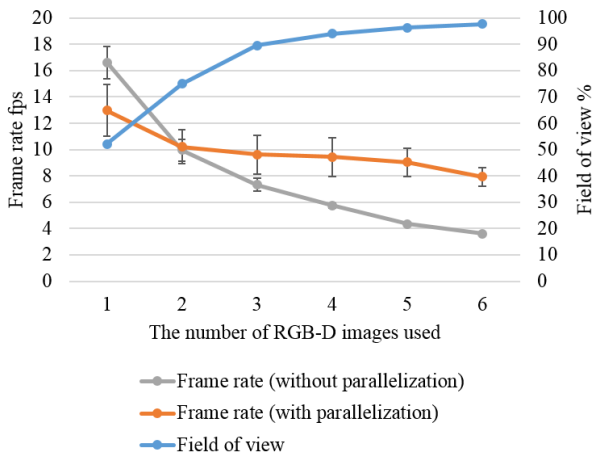


Fig. 13 Relationship between the number of RGB-D images used, frame rate and the percentage of the available field of view

ムレートが得られた。なお、使用する RGB-D 画像の枚数が 2 枚以下のときにフレームレートが向上しなかったのは、並列計算のための準備におけるオーバーヘッドが影響しているためと考えられる。

遠隔操作による作業の例である無人化施工の支援を目的とした俯瞰映像の研究¹⁸⁾で行われた、油圧ショベルの位置決め精度の評価実験では、オペレータに提示する俯瞰映像のフレームレートは平均 9 fps であった。これは、この研究での実験作業において十分使用に堪え得るフレームレートであったことがインタビューの結果から明らかになっている。作業によってリアルタイムの遠隔操作に求められるフレームレートは異なるが、本実験ではこの参考値 9 fps と同程度のフレームレートが得られたことが分かる。

4.4.3 色情報が得られた領域の割合

また、出力画像の中で色情報が得られた領域の割合は、使用する RGB-D 画像の枚数が 2 枚のとき (図 7 (a)) は 75.0 %, 4 枚のとき (図 7 (c)) は 94.1% となった。参考までに、使用する RGB-D 画像の枚数を 6 枚まで増やした場合においては 97.6 % であった。使用する RGB-D 画像の枚数を増やすことで視野が広がり、画像のほぼ全体の色情報を取得可能になることが確認できた。

5. 結 論

本研究では、ロボットの遠隔操作において 1 つの映像中で作業対象を認識可能にすることを想定した、遮蔽物透視システムを提案した。提案手法により、アームとともにセンサを移動させることで広い視野の 3 次元情報を取得し、任意視点からの透視映像を生成することが可能となった。また、過去に取得した環境の情報を利用し映像の視野を拡張する手法、および環境中のある特定の部分をアームセンサで観測するためのアームの運動指令を生成する手法を提案した。提案手法では、ロボットアームの順運動学によって各センサ間の位置関係を求めることで、点群の位置合わせなどの計算コストの大きい処理を避け、さらに並列計算に適したアルゴリズムを提案することで処理の高速化を図った。また、提案手法ではセンサから取得した点群情報を全て透視映像上に表示するため、透視対象の物体の形状などを仮定する必要がない。よって、透視対象が RGB-D センサから見える位置にあり、RGB-D センサで透視対象の奥行き情報を

取得可能であれば、対象物の形状によらず透視可能である。そして、実機による実験を行い提案手法の有効性を確認し、広い視野の透視映像を生成可能であることを実証したとともに、本システムのリアルタイム性を評価した。

今後の展望としては、以下の 3 点が考えられる。

- (1) リアルタイム性の向上
- (2) センサ情報統合時の位置のズレの軽減
- (3) センサ台数を任意の数に増やした場合への拡張

リアルタイム性の向上のためには、より性能の良いプロセッサを使用することに加えて、GPU によるさらなる並列処理などに対応した実装方法を工夫する必要がある。また、画像の位置のズレを軽減するためには、各センサ情報から生成した 3 次元点群の位置合わせを行うことが考えられる。従来手法¹²⁾¹³⁾では、センサの移動推定を行うために 3 次元点群の位置合わせ手法が必須である上、センサの移動量が大きくなるほど位置合わせ手法の計算コストが増大する可能性がある。一方、提案手法ではロボットアームの順運動学によって各センサ間の位置関係を求められるため、点群の位置合わせ手法は原理的には必須ではなく、4.4.1 項で考察した誤差の影響のみを想定すればよい。よって、従来手法と比較して少ない計算コストでの位置合わせ手法を構築することが可能であると考えられる。

提案手法は任意の台数のセンサからなるシステムへと拡張が可能である。センサの台数が多いほど一度に多くの情報を取得可能なため、動的な環境に対しても広い視野を提示可能な遮蔽物透視システムを提案できると期待される。

謝 辞

本研究の一部は、総合科学技術・イノベーション会議により制度設計された革新的研究開発促進プログラム (ImPACT) 「タフ・ロボティクス・チャレンジ」の援助を受けた。また、本研究で使用したマニピュレータは株式会社安川電機からご提供いただいた。

参 考 文 献

- 1) Shinji Kawatsuma, Mineo Fukushima and Takashi Okada: Emergency Response by Robots to Fukushima-Daiichi Accident: Summary and Lessons Learned, *Industrial Robot: An International Journal*, **39**, 5, (2012), 428.
- 2) Masaharu Moteki, Kenichi Fujino, Takashi Ohtsuki and Tsuyoshi Hashimoto: Research on Visual Point of Operator in Remote Control of Construction Machinery, *Proceedings of the 28th International Symposium on Automation and Robotics in Construction*, (2010), 532.
- 3) Akihiko Nishiyama, Masaharu Moteki, Kenichi Fujino and Takeshi Hashimoto: Research on the Comparison of Operator Viewpoints between Manned and Remote Control Operation in Unmanned Construction Systems, *Proceedings of the 30th International Symposium on Automation and Robotics in Construction*, (2013), 772.
- 4) 森尚平, 一刈良介, 柴田史久, 木村朝子, 田村秀行: 隠消現実の技術的枠組と諸問題: 現実世界に実在する物体を視覚的に隠蔽・消去・透視する技術について, *日本バーチャリアリティ学会論文誌*, **16**, 2, (2011), 239.
- 5) Peter Barnum, Yaser Sheikh, Ankur Datta and Takeo Kanade: Dynamic Seethroughs: Synthesizing Hidden Views of Moving Objects, *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality*, (2009), 111.
- 6) Yiping Shen, Fei Lu, Xiaochun Cao and Hassan Foroosh: Video Completion for Perspective Camera under Constrained Motion, *Proceedings of the 18th International Conference on Pattern Recognition*, **3**, (2006), 63.

- 7) Arturo Flores and Serge Belongie: Removing Pedestrians from Google Street View Images, Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, (2010), 53.
- 8) Francesco Cosco, Carlos Garre, Fabio Bruno, Maurizio Muzzupappa and Miguel A. Otaduy: Augmented Touch without Visual Obtrusion, Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality, (2009), 99.
- 9) Jan Herling and Wolfgang Broll: Pixmix: A Real-time Approach to High-quality Diminished Reality, Proceedings of the 11th IEEE International Symposium on Mixed and Augmented Reality, (2012), 141.
- 10) Kazuya Sugimoto, Hiromitsu Fujii, Atsushi Yamashita and Hajime Asama: Half Diminished Reality Image Using Three RGB-D Sensors for Remote Control Robots, Proceedings of the 12th IEEE International Symposium on Safety, Security, and Rescue Robotics, 43, (2014), 1.
- 11) Songkran Jarusirisawad, Takahide Hosokawa and Hideo Saito: Diminished Reality Using Plane-sweep Algorithm with Weakly-calibrated Cameras, Progress in Informatics, **7**, (2010), 11.
- 12) Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges and Andrew Fitzgibbon: KinectFusion: Real-Time Dense Surface Mapping and Tracking, Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality, (2011), 127.
- 13) Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J. Leonard and John McDonald: Real-time large-scale dense RGB-D SLAM with volumetric fusion, The International Journal of Robotics Research, **34**, (2015), 598.
- 14) Zhengyou Zhang: A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, **22**, **11**, (2000), 1330.
- 15) 佐藤雄隆: RGB-D カメラから得られる Depth データの歪み補正, 映像情報 industrial, **46**, **3**, (2014), 66.
- 16) 奥野洋朗, 金井理, 伊達宏昭: RGB-D カメラを用いた屋内環境マッピングの高速化, 精密工学会北海道支部講演論文集, (2015), 29.
- 17) 藤野 慎也, 石井 雅樹, 佐藤 俊太郎: 距離画像センサの精度向上を目的とした補正テーブルの作成, 第 22 回画像センシングシンポジウム講演論文集, (2016), IS2-11.
- 18) 佐藤 貴亮, 藤井 浩光, Alessandro Moro, 杉本 和也, 野末 晃, 三村 洋一, 小幡 克実, 山下 淳, 浅間 一: 無人化施工用俯瞰映像提示システムの開発, 日本機械学会論文集, **81**, 823, (2015), 14.