



修士論文

同期の不完全性と観測誤差を考慮した 複数カメラによる位置推定精度の改善



平成 20 年度修了
三重大学大学院 工学研究科
博士前期課程 情報工学専攻

古川 智久

目次

第1章 序論	1
第2章 ロボットのシステム構成	3
2.1 情報統合までのプロセス	3
2.2 ロボットのハードウェア構成	5
2.3 ロボットのソフトウェア構成	5
2.3.1 map タスク	7
2.4 天井カメラ	7
2.5 時刻同期	7
2.5.1 RBS(Reference Broadcast Synchronization)[13]	8
2.5.2 クロック誤差と関数近似による同期	9
2.5.3 同期実験結果	11
第3章 対象物体の位置推定	14
3.1 Zhang のカメラモデル	14
3.2 対象物体の位置推定手法の提案	16
3.2.1 対象物体方向の推定	16
3.2.2 情報取得タイミングが非同期なシステムの考慮	17
3.2.3 交点の計算	18
3.3 実験	19
3.3.1 実験方法	19
3.3.2 実験結果	21
3.3.3 考察	27
第4章 情報統合	28
4.1 正規分布を用いた統合	28
4.2 提案手法	30
4.2.1 分布の決定	31
4.3 実験	32

4.3.1 実験方法	32
4.3.2 実験結果	32
4.3.3 考察	32
4.4 対象物体が動く環境における実験	35
4.4.1 実験方法	35
4.4.2 実験結果	39
4.4.3 考察	39
 結論	 45
 謝辞	 46
 付録 A カメラキャリブレーション	 49

第1章

序論

近年、危険な環境下における作業を人間の代わりにロボットに行わせること等を目的として、自立行動型ロボットの研究が盛んに行われている。ロボットが自律行動し、期待される仕事を行うためには、その周囲に存在する物体との位置関係等の環境情報を正確に取得する必要がある、非常に重要な課題となっている。現在、これらの必要な環境情報を取得する手段として、主にカメラが用いられている。しかし、単一のセンサや単体のロボットだけでは、センサの観測範囲外に目標物が存在することや、目標との間に障害物が存在するといったこと等により、目標物が観測されないことが起こり、問題となる。また、一般にカメラによる観測情報には、量子化誤差や観測誤差等が含まれるという問題がある。これらの問題を解決する為に、複数の情報を統合することによって環境情報の相互補完を行う情報統合技術がある。本研究は、地理的に位置が異なる複数個のセンサから得られる情報を統合する方法に関して研究するものである。

本研究で扱う統合では、様々なアングル、距離からの情報を得ることが出来るが、各センサが含む誤差も場所により変化する。また、一般に情報統合は各センサが情報を取得した際の時刻情報を元に統合が行われるが、システムのセンサが全て同期していることは稀であり、各センサの情報取得時刻が異なることも重要な問題である。従来研究として、黒田 [2] が本研究室において Elson[13] や Luo[5] らの手法を取り入れ、非同期システムを考慮した情報統合に関する研究を行っていたが、統合後の誤差は大きく改善の余地がある。また、情報の統合手法に関しても、Spanos らの、重みをつけた算術平均による統合方法 [12]、Stroupe らの、センサの観測情報の誤差を正規分布に近似し、統合する手法 [7] や、Sun らの、カルマンフィルタを用いて統合する手法 [11] 等、様々な手法が提案され、研究がなされているが、これらの研究は、各ロボットが独自に対象物体の世界座標位置を推定できるものとしている。しかし、ロボット単体で対象物体の正確な位置推定を可能とするには、幾つものセンサを取り付ける必要がある、これには非常にコストがかかるという問題がある。また、これらの研究では、システムの非同期性についての考慮がされていない。

そこで、本研究では、非同期の、複数ロボットの単眼カメラの画像情報のみを利用した統合法について、より精度よく統合できる手法について研究するものである。

本章では、研究の背景及び目的について述べ、また関連研究についても簡単に述べた。第2章では、本研究室のロボットのシステム構成と同期方法について述べる。また、一般的な情報統合について、情報統合における問題と従来研究についても述べる。第3章では、非同期システムを考慮した複数ロボットの単眼カメラの画像情報のみを利用した対象物体の位置推定法について、実験やそ

の結果を交えて述べる。第 4 章では、第 3 章で述べる位置推定法により得られた情報を統合する方法について、こちらも実験やその結果を交えて述べる。最後に第 5 章で結論を述べる。

第2章

ロボットのシステム構成

本章では、一般のシステムにおいて、情報統合がどのような流れで行われ、各ステップにおいてどのような問題が存在するかについて述べる。また、それらの問題を解決する従来研究と本研究における手法について簡単に述べる。次に本研究で用いたロボットと、天井カメラについて、そのシステム構成について述べ、最後に、分散環境におけるロボット間の時刻同期の方法とその精度について述べる。情報統合は寄せられる情報の取得時刻を元に統合を行っており、各ロボット間において、時刻が同期していることは非常に重要な問題である。ここでは、各ロボットが持つ CPU クロックがそれぞれ固有の誤差を持つことに注目し、その誤差を線形に補正することによって、各ロボットの時刻同期精度を高める。

2.1 情報統合までのプロセス

一般に、情報統合が行われるまでの流れを図 1 に示す。

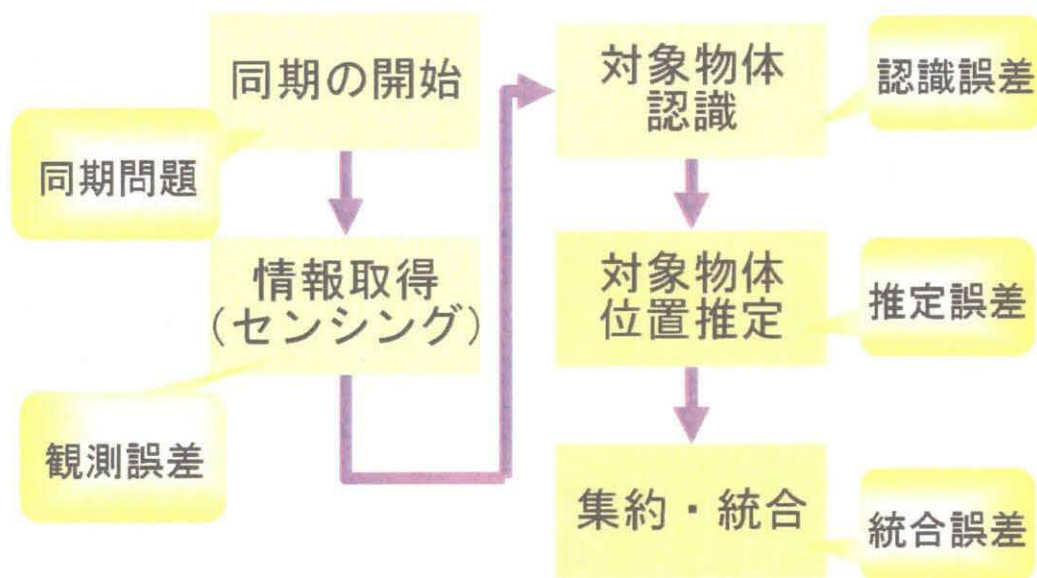


図 1 情報統合までの流れ

■時刻同期 図 1 に示される様に、あるシステムにおいて情報統合を行う際は、まずシステムの時刻同期を行う。これは、情報を統合する際に、時刻情報を基準とし、各センサが情報を獲得した際

の時刻情報を元に、統合を行う為である。その為、情報統合を考える場合には、まず、正確な時刻同期が行われることが必要となる。従来研究として、本研究室の黒田 [2] が行った RBS[13] を用いた同期手法があるが、長期的な同期には問題があったため、これを改善 [1] し、本研究でも用いた。[1] については、本章の最後の節で詳しく述べる。また、同期問題には、センサの時刻同期問題の他に、センサの情報取得タイミングの同期問題が存在する。これについても、本研究室の黒田 [2] が、Luo の手法 [5] を参考に対応がなされているが、そのままでは、本研究で提案する位置推定手法には対応出来ないので、本手法に対応したものに改良した。

■情報取得 (センシング) 時刻同期が行われた後は、システムの各センサにおいて、情報の取得が行われる。この時、例えばカメラでは、画像情報に変換する際の量子化誤差等や周囲の自然環境のブレによる観測誤差等が含まれる為、これらの誤差を考慮する必要がある。これらの誤差は一般的に正規分布で近似される [7], [8]。本研究においても、正規分布で近似することによりこれらの誤差を表現する。

■対象物体認識 センサで取得した情報から対象物体の認識を行う。カメラで情報取得をした場合には、画像処理に当たる。この時、画像処理が正確に行われるかどうかによって誤差が含まれるかどうかが決まる。従って、如何に正確な画像処理を行うかという課題が存在し、本研究室においても研究が行われている [18], [19] が、本研究では扱わないこととした。

■対象物体位置推定 対象物体認識により得られた情報から、対象物体のシステムの世界座表系における位置推定を行う。カメラで情報を取得した場合には、画像上の対象物体位置から、カメラのキャリブレーション結果等を用いて、世界座標系における位置を推定する。しかし、一般にカメラ 1 台のみで画像上の情報から世界座標系における位置情報を推定することは不可能である。対象物体の高さが既知であったり、対象物体がある一定の平面上を移動する様な場合には、未知情報が減る為、推定が可能となる [3], [4], [17] が、誤差は大きい。本研究では、対象物体がある一定の平面上を移動すると仮定し、2 台のカメラを使用して、対象物体とカメラのレンズ中心を結んだ直線の交点から位置推定を行う。

■集約・統合 同一の対象物体について推定された位置情報を集約し、統合を行うことでシステムとしてより正確な対象物体の位置情報を獲得する。また、複数のロボット等で統合を行った場合は、統合結果を同一システム内の全ロボットに送り、共有する。この時に、どのような手法を用いて統合するかによって、統合後の精度が変わる。この時の統合方法は、センサと対象物体迄の距離により重みをつけた算術平均による方法 [2], [12] や、統合に至るまでの誤差を正規分布に近似し統合する方法 [7]、ベイズ則に基づき統合する方法 [9], [10]、カルマンフィルタを用いて統合する方法 [6], [11] 等がある。本研究では、Stroupe らの方法 [7] を参考に、カメラ対による位置推定方

法に対応した統合手法に拡張することで統合を行う。

2.2 ロボットのハードウェア構成

本研究では、本研究室で独自に作成されたロボット 3 台を用いた。ロボットの外観を図 2 に示す。ロボットは、移動機構部、制御部、単眼のビデオカメラで構成されており、3 台ともほぼ同じである。移動機構部は動輪 2 輪、ステアリング輪 1 輪の 3 輪方式である。また、移動機構部の周囲には 8 個の超音波センサが取り付けられ、衝突回避等に利用される。制御部はアルミ板によって作成された筐体の中に、電源、バッテリー、バックプレーン、CPU ボード、画像処理ボード、HDD、が搭載されている。カメラは、御部の筐体の上に載せて固定してある。このカメラは、SONY の EVI-D30 であり、ワイドコンバージョンレンズを取り付ける事によって約 60° の前方視野角を有している。ロボットは、主にカメラからの映像入力として得られる 256×220 pixel の画像を処理することによって周囲の環境情報を取得し、動作を行う。

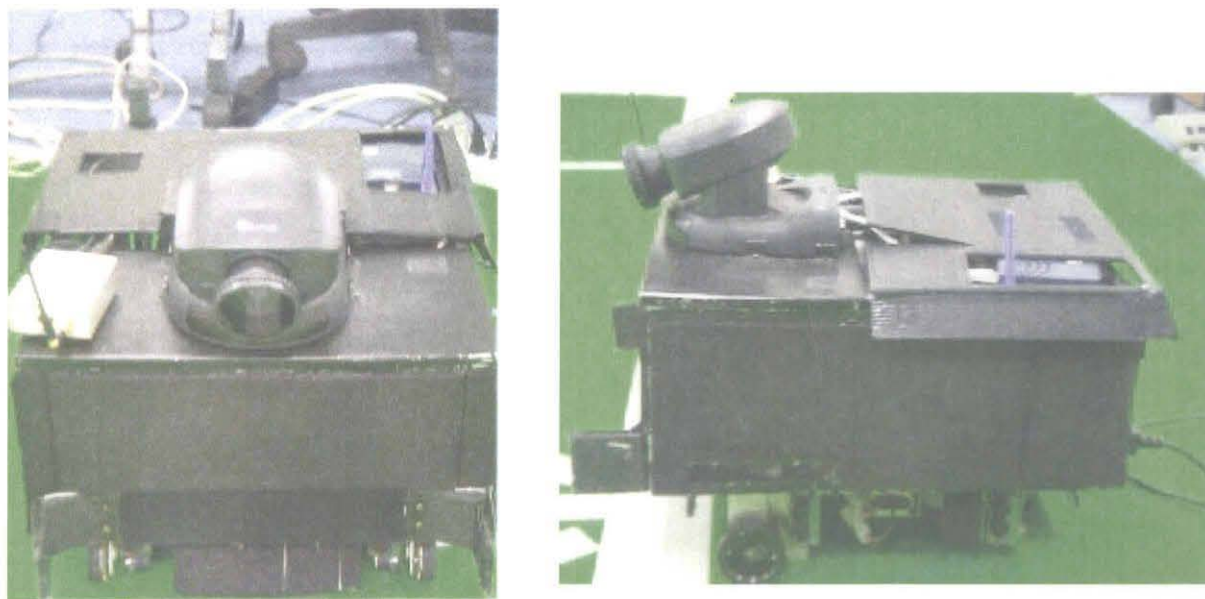


図 2 本研究室のロボット

2.3 ロボットのソフトウェア構成

本研究室のロボットの制御部で使用しているソフトウェアを表 1 に示す。ソフトウェアの構成については、ハードウェア構成と同様に各ロボットとも同じ構成である。

表 1 に示した通り、本研究室のシステムでは、仮想並列コンピュータを実現するソフトウェア

表1 制御部のソフトウェア構成

OS	Vine Linux 3.3.2-0vl18 (kernel version 2.4.31-0vl1.8)
開発環境	gcc version 3.3.2, g++ version 3.3.2
描画処理環境	Gtk version 1.2.10, Glib version 1.2.10
分散並列処理環境	PVM version 3.4.3

PVM[20] を用いてタスクの分散処理を行っている。これによって、オブジェクト指向的な各タスクの並列処理が可能になっている。また、各タスクは固有のデータを管理し、他のタスクからの情報要求メッセージに対して必要な情報のみを通知する為、データの一貫性や独立性も実現されている。したがって、複数のロボットからなるチーム全体をひとつの分散処理システムとして一貫した形態で扱うことが可能となっている。次に、PVM[20] によって実現されている各タスクとその役割を表2に示す。

表2 ロボットのタスク

cog	ロボットが獲得した全ての情報を収集し、対象物体までの相対方向、相対距離、世界座標系における位置の計算や自己位置同定を行う。 map タスクを使用する場合は代表するロボットの map タスクに情報を送信する。
plan	cog タスクから各オブジェクトの位置情報等を受け取り、次の行動を計画する。
transfer	制御部と移動機構部の通信制御を行い、移動機構部からの情報を取得する。
camera	ビデオカメラの動作制御を行い、カメラの pan 角, tilt 角の情報を取得する。
ip	画像処理ボードの制御を行い、6 色のカラー画像処理にてオブジェクトやボールを認識する。 現在、画像 1 フレームに対して処理にかかる時間は、約 100msec である。 尚、ip タスクで得られた情報は、cog タスクに送られる。
map	環境中に存在するすべての味方ロボットがもつ情報を集約して環境 MAP を作成する。 また、その際、同期の処理、情報統合の処理を行う。 このタスクに関しては次の節で詳しく述べる。

本研究では、これらのタスクの内、各ロボットの ip タスクで画像処理された情報を、cog タスクで解析し、map タスクに集約して更なる解析処理を行うことで情報統合を行う。次の節では、map タスクについて述べる。

2.3.1 map タスク

本研究では、時刻同期問題と情報統合を扱うことは先にも述べたが、これらの処理は map タスクで行われる。map タスクの主な処理の流れを以下に示す。

1. タスク開始時刻を取得
2. 初期化处理
3. 各ロボットの cog タスクから環境情報に関する全データを受信
4. 受信したデータを集計
5. 現在時刻の取得及び各データの時刻情報を確認し、同期の処理を行う
6. 集計したデータを元に環境 MAP を作成 (情報統合を行う)
7. 作成した環境 MAP を各ロボットの cog タスクに送信
8. (3) に戻る

map タスクは、同じチームのロボット中の 1 台で集中的に実行され、各ロボットに処理結果を送信する。これによって、map タスクで処理された環境情報が全ロボットで共有され、協調行動等に利用することが可能となる。

2.4 天井カメラ

本研究の実験で使用する天井カメラについて述べる。天井カメラの外観と撮影された画像を図 3 に示す。カメラはロボットと同じく SONY の EVI-D30 であり、システムはキャプチャボードを搭載した PC とカメラとを繋ぎ CMVision^{*1}[22] をライブラリとして使用することによって、30fps で映像を取り込めるものである。この時に取り込まれる画像の解像度は 320×240 pixel である。現在は、取り込んだ画像の取り込み時刻を記録することで 30msec 毎のオブジェクトの位置を知ることが出来、このオブジェクト位置を線形補間することで時々刻々の位置を計算することが出来る。

2.5 時刻同期

本研究室における分散環境での時刻同期システムについて述べる。先に述べた通り、一般に情報統合では情報の取得時刻を元に統合が行われる。これは、本研究室のシステムでも同じである。現在、本研究室のシステムでは、まず、各ロボットが画像取得した時刻の情報を各オブジェクトの情報と共に map タスクに送信する。その後、map タスクが各ロボットの時刻情報を元にオブジェク

^{*1} CMVision については西川の論文 [17] を参照

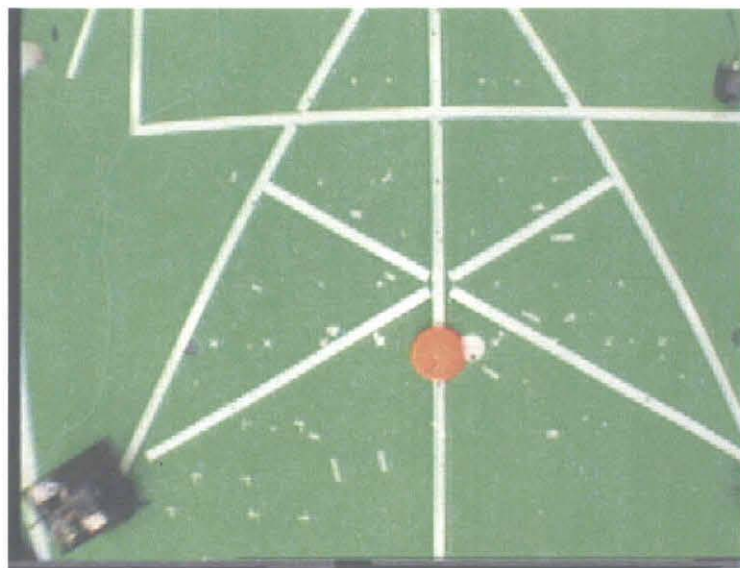


図 3 天井カメラの外観と撮影された画像

トの位置を予測することによって、情報統合が行われる。この為、各ロボットにおいて時刻同期がとれていないと、各情報の取得時点が実際のものとずれる現象が起こり、正確な統合が出来ないという問題になる。このような理由により、情報統合を行うに当たって、各ロボット間での時刻同期が必要である。

従来は、本研究室の黒田 [2] が行ったシステム中のあるロボットに他の全ロボットの時刻合わせを行い、更に RBS(Reference Broadcast Synchronization)[13] を用いて、時刻合わせで合わせきれなかったずれを補正することによって同期を行っていたが、長期的な同期に問題があった。本研究では、卒業研究時にこれを改善した手法 [1] を用いて同期を行う。以下で [1] での同期方法について述べる。

2.5.1 RBS(Reference Broadcast Synchronization)[13]

RBS について述べる。RBS とは、送信側が受信側に対してメッセージをブロードキャストし、その受信時刻を受信側同士が比較することによって時刻のずれを計測する方法である。RBS による同期方法の流れを以下に示す。

1. 送信側が参照パケットを受信側にブロードキャスト (この時に送信時刻はダンプしない)
2. 受信側は送られてきた参照パケットの自分の時計での受信時刻を記録

3. 受信側同士で互いに受信時刻の情報を交換し、時刻のずれを計算

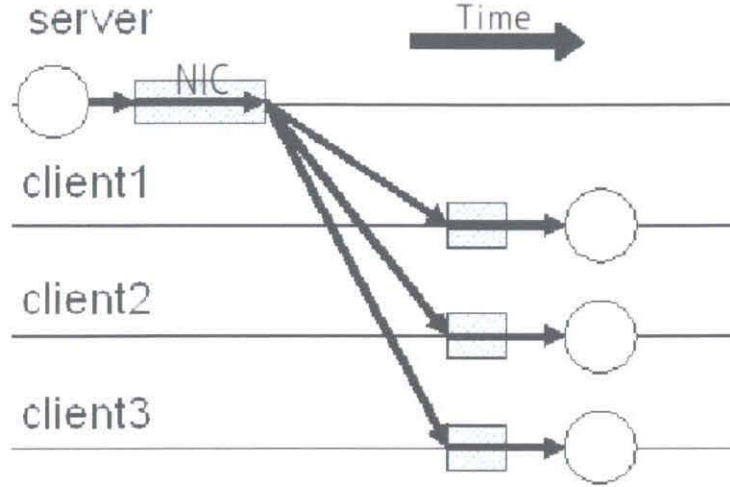


図4 RBS でのメッセージの流れ

この時の受信側 i, j 間の時刻のずれは式 (1) で計算出来る.

$$\forall i \in n, \forall j \in n; Offset[i, j] = \frac{1}{m} \sum_{k=1}^m (T_{j,k} - T_{i,k}) \quad (1)$$

n は受信側の数. m はブロードキャストする参照パケットの数. $T_{r,b}$ はブロードキャストされた参照パケット b を受信した時の受信側 r の時刻を表す.

受信側同士が同期をとる為に、ブロードキャストチャンネルを用いて、参照パケットを送信することによって、一番大きな非決定的なレイテンシの原因がクリティカルパスから取り除ける (図4). その残りの誤差は正規分布となり、また、参照パケットを何回も送信することによって改善が見込める.

2.5.2 クロック誤差と関数近似による同期

各ロボットのクロック誤差と関数近似による同期について述べる. 現在, 本研究室のロボットは全て, 各々の CPU クロックから時刻情報を取得しているが, この CPU クロックには各々固有の誤差が存在する. その為, に示されるように同期開始時点から時間が経過するにつれ, 各ロボット間の時刻情報のずれが大きくなってゆく. その様子を図5に示す. 図5は, 横軸が経過時間, 天井カメラが接続されているデスクトップ PC を基準とし, デスクトップ PC に対する時刻のずれを表す.

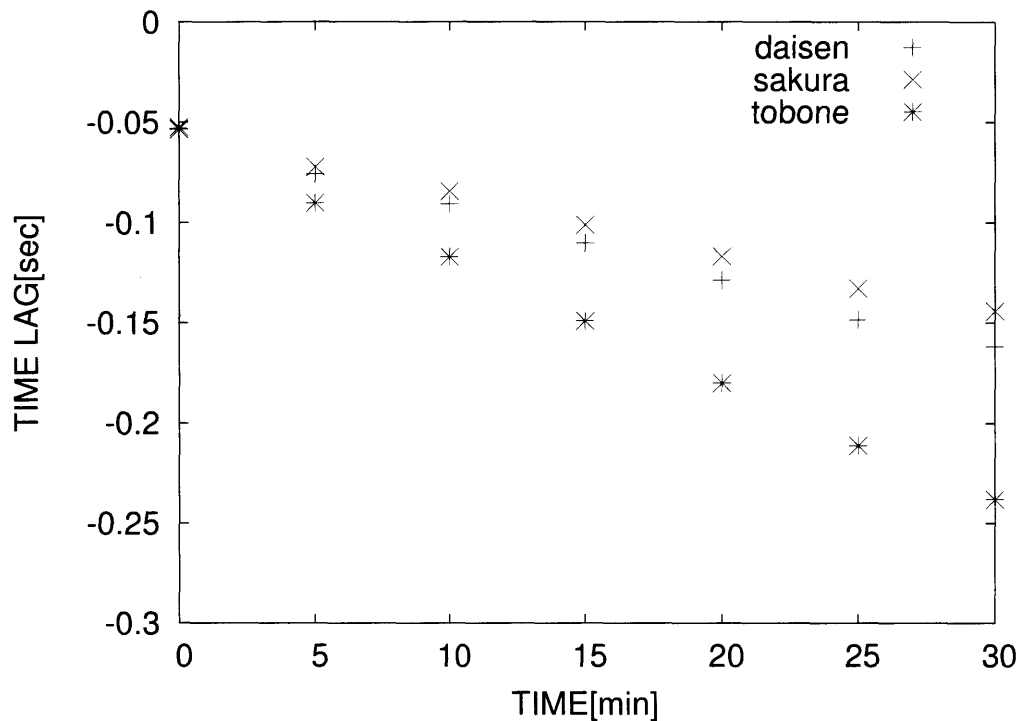


図 5 時間の経過による各ロボット間の時刻差の変化

そこで本研究では、RBS による同期に加え、各クロック固有のクロック精度の誤差統計的なデータから関数近似し、ソフトウェア的に補正することで長期的で精度の高い同期を図る。具体的な同期の流れは以下である。

1. クロック誤差情報を各ロボットが保持
2. ntpdate コマンドにより時刻合わせを行う
3. RBS でずれを計算する際に親となる PC(本研究ではノート PC を使用) が同期開始時刻を記録し、他のロボットに配布
4. RBS によりずれを計算
5. RBS によるずれの情報と、クロック誤差情報によって補正

補正後の各ロボットの時刻は式 (2) で表される。

$$Time = (Error \times t) + Offset + Time_{orig} \quad (2)$$

$Error$ は各ロボット固有のクロック誤差。 t は同期開始時点からの経過時間。 $Offset$ は RBS により求められる同期開始時のずれ。 $Time_{orig}$ は補正前の時刻である。

2.5.3 同期実験結果

本研究で使用したロボット 3 台 (daisen, sakura, tobone) と天井カメラをつないだデスクトップ PC を用い、デスクトップ PC を基準のものとして同期実験を行い、各ロボットの誤差情報の収集及び、同期精度を確かめた。

ロボットのクロック誤差補正の関数作成にはの 200 例のデータを用い、最小二乗法にて関数を算出した (図 6 に例を示す)。この時のクロック誤差は、1 秒間にどれだけクロックがずれるかを表している。尚、1 例のデータにつき、約 1 分間おきに各ロボットの時刻の計測をすることを 30 回行ったので、100 時間分のデータより関数を算出したことになる。作成された関数を用いた同期実験の結果を図 7, 8 に示す。また、比較対象として、[13] に紹介される ntp デーモンと RBS を用いた同期結果を図 9 に示す。

この結果の見方としては、縦軸がデスクトップ PC と各ロボットとの時刻のずれであり、横軸が経過時間である。その為、横軸の各時点において、縦にプロットされるデータが密ならば、ロボット 3 台の間においてより正確な時刻同期が出来ているということであり、横軸の値によらず、縦にプロットされるデータの各間隔が同じならば、安定して同期がとれているということである。また、デスクトップ PC との同期も考えると、各データが横一直線に並ぶと最も理想的な同期が出来ているといえる。

実験結果を見ると、同期開始より約 30 分経過した後でも、各ロボットにおける時間経過によるずれの変化は、1msec 以内となっている。これにより、時間が経過しても、高い精度で同期がとれていることが分かる。また、ntpd を用いた同期手法に比べても、安定して高い精度で同期がとれていることが分かる。

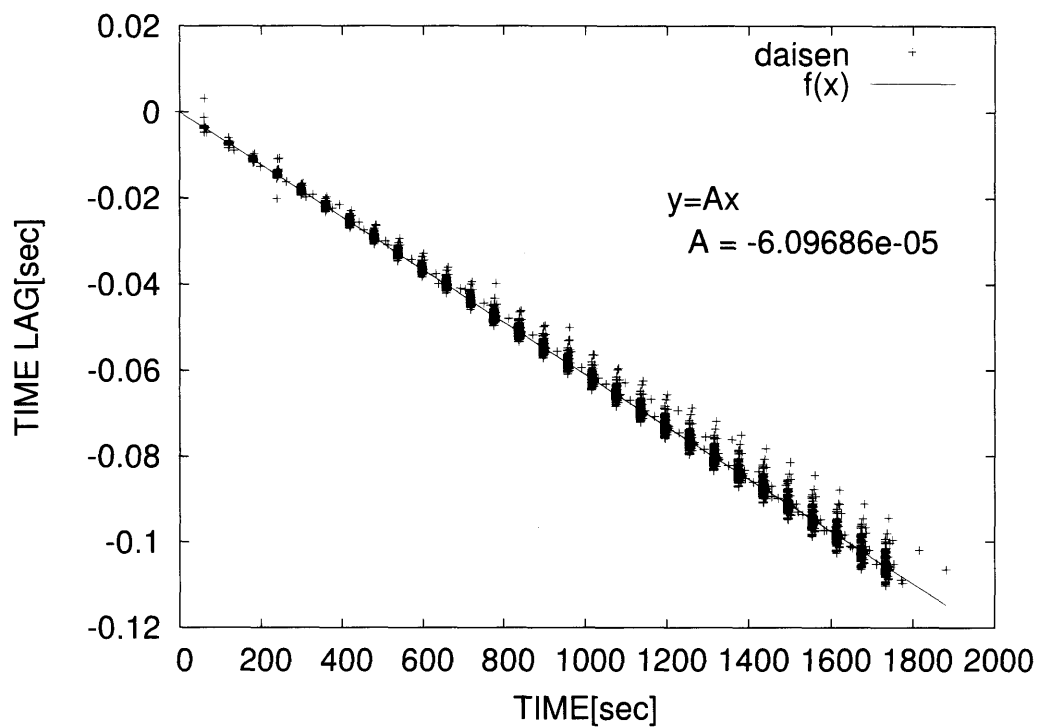


図 6 ロボット “daisen” のクロック誤差

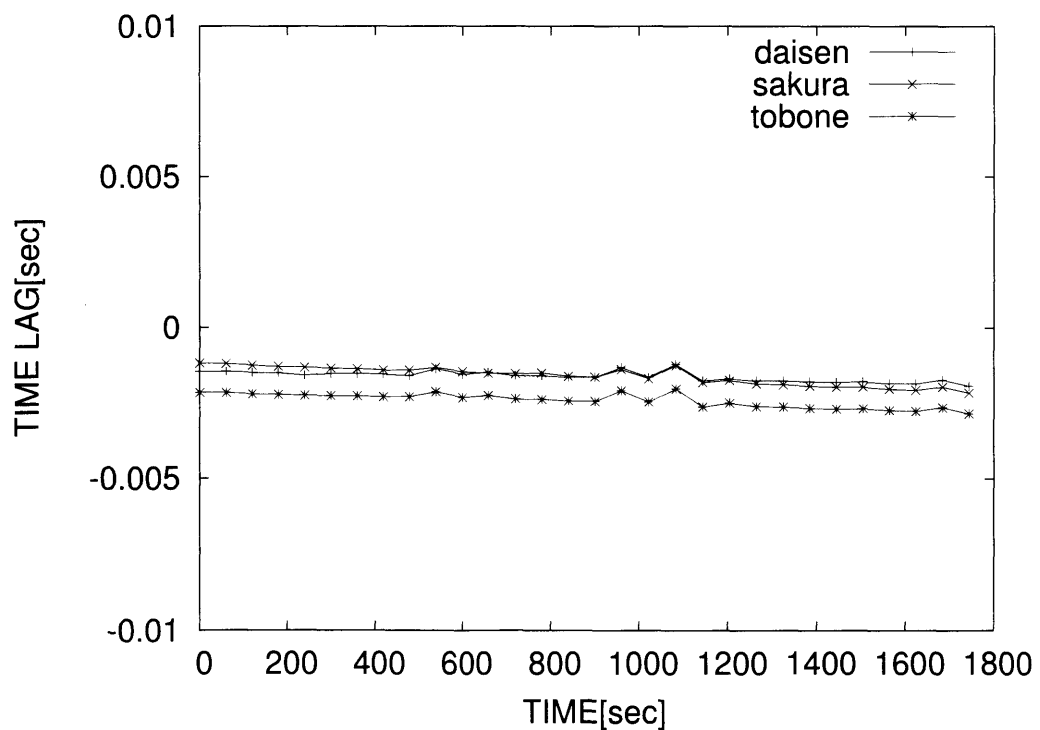


図 7 同期実験結果

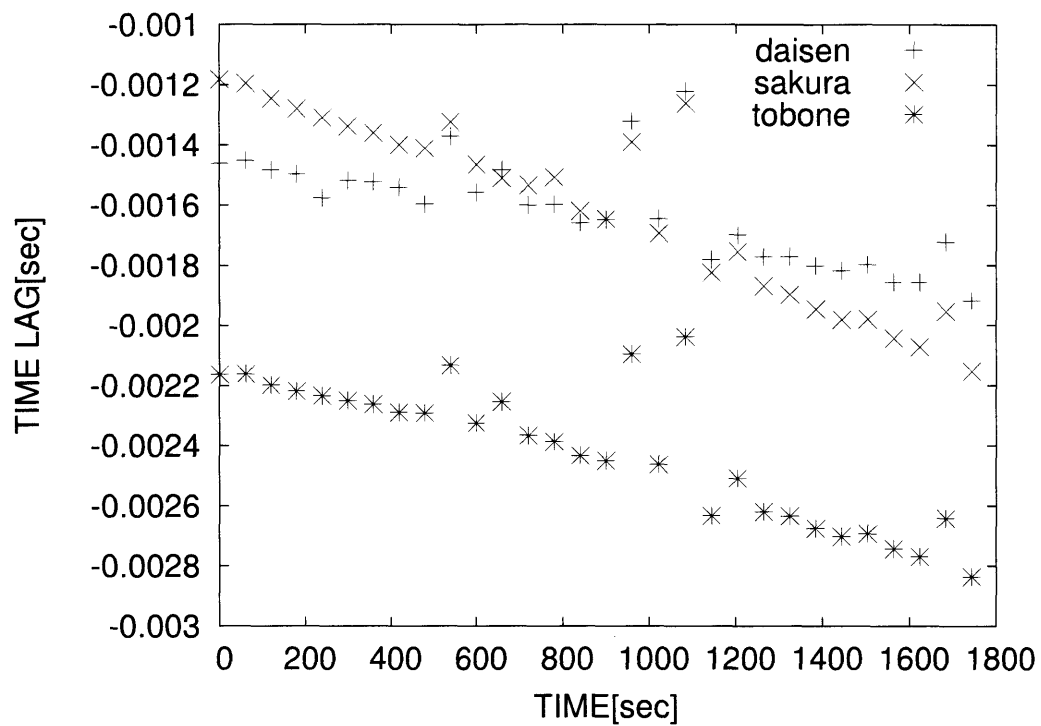


図 8 同期実験結果 (拡大)

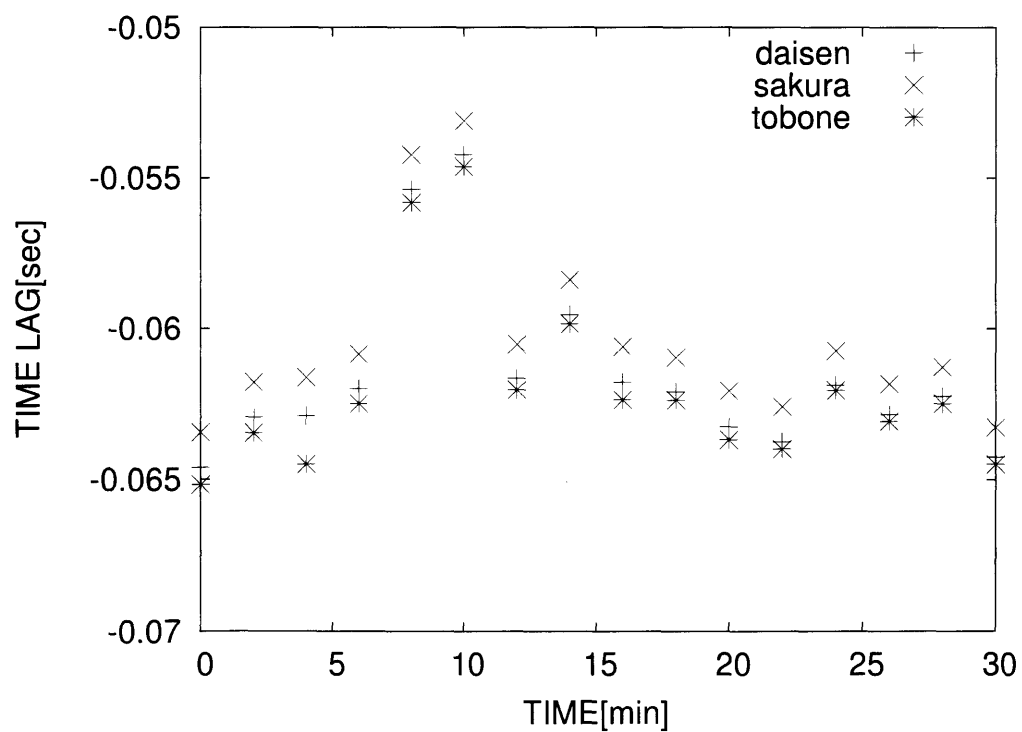


図 9 ntpd を用いた同期実験結果

第3章

対象物体の位置推定

本章では、対象物体の位置情報を獲得する方法について述べる。ある対象物体についての情報統合は、各々のロボットで推定された対象物体の位置情報を統合することにより行われる為、各ロボットにおいて、対象物体の位置情報をより正確に推定することは非常に重要な問題である。

一般に、対象物体の位置を正確に推定するには、複数のセンサを利用する必要がある。特に、本研究室のロボットのようにカメラから得られる画像情報から推定をする場合には、単一の画像情報からカメラと対象物体までの距離を推定することは、一般に不可能である。1台のロボットに複数のカメラを搭載する等の必要がある。その例として、1台のロボットに搭載された複数のセンサ情報を利用し、正確な位置推定を行う研究が行われている [16]。しかし、これではロボット1台当たりのコストが非常に高くなってしまいう問題がある。カメラが単眼のカメラの場合、対象物体の高さが一定で且つある平面上のみを移動すると仮定すれば、消失点を用いた距離算出法 [4] 等により測定可能であるが、誤差は大きい。

そこで、本研究では、2台の単眼カメラを利用し、各カメラの画像情報から、各カメラのレンズ中心と対象物体とを結ぶ直線の交点を計算することによって対象物体の位置推定を行う方法をとる。

3.1 Zhang のカメラモデル

カメラから得られた画像情報より、対象物体の方向を推定する為に、本研究で用いたカメラモデルについて述べる。

本研究室では、ワイドコンバージョンレンズを使用したカメラにより情報取得をしている為、カメラの歪曲収差が無視できない状況となっている。そこで、本研究では、カメラの歪曲収差を考慮した Zhang のモデル [14] を用いる。

Zhang のモデルは、ピンホールカメラモデルを基本として、透視投影変換を用いて、3次元座標を画像平面に投影することで構成されている。この時の3次元座標から画像面座標への変換は、式 (3), (4) で表される。

$$s \cdot m' = A \cdot [R \mid t] \cdot M' \quad (3)$$

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (4)$$

$(X, Y, Z)^T$ はワールド座標系の 3 次元座標であり, (u, v) は $(X, Y, Z)^T$ が画像平面に投影された時の画像座標である. A はカメラ行列, またはカメラの内部パラメータ行列と呼ばれている. c_x, c_y は画像中心である点 (レンズの光学的な中心) であり, f_x, f_y は焦点距離でピクセル単位で表現される. また, R は回転行列, t 平行移動ベクトルであり, $[R|t]$ は 3 次元座標 $(X, Y, Z)^T$ をカメラ座標系に変換する同次変換行列を表す外部パラメータと呼ばれる. 世界座標系 $(X, Y, Z)^T$ からカメラ座標系 $(x, y, z)^T$ への変換は式 (5) のように表される.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \quad (5)$$

更に, カメラ座標系 $(x, y, z)^T$ から, 画像座標系 (u, v) への変換は以下で表される.

$$x' = \frac{x}{z} \quad (6)$$

$$y' = \frac{y}{z} \quad (7)$$

$$u = f_x \times x' + c_x \quad (8)$$

$$v = f_y \times y' + c_y \quad (9)$$

カメラのレンズは半径方向の歪みや, 円周方向の歪みを持っているので, 以上のモデルは, k_1, k_2 を半径方向の歪み係数, p_1, p_2 を円周方向の歪み係数として以下のように拡張される.

$$r^2 = x'^2 + y'^2 \quad (10)$$

$$x'' = x'(1 + k_1 r^2 + k_2 r^4) + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \quad (11)$$

$$y'' = y'(1 + k_1 r^2 + k_2 r^4) + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \quad (12)$$

$$u = f_x \times x'' + c_x \quad (13)$$

$$v = f_y \times y'' + c_y \quad (14)$$

各々の歪み係数は, カメラの姿勢によらないので, 内部パラメータとなる. カメラキャリブレーションにより, カメラの内部パラメータ, 外部パラメータを求めることによって, 世界座標系から $(X, Y, Z)^T$ から画像座標系の値 (u, v) を求めることが可能となる. また, 対象物体の高さが分かっ

ているか、カメラから対象物体までの距離が分かっているならば、画像座標系 (u, v) の値から世界座標系の値 $(X, Y, Z)^T$ も求めることが可能となる。尚、この時のカメラ座標系のとりかたは、奥行きを z 軸の正方向とした、右手系の座標系である。

3.2 対象物体の位置推定手法の提案

本研究で提案する、カメラ対による対象物体の位置推定方法について述べる。前述のカメラモデルにより、対象物体の画像上の位置情報のみからでは、カメラと対象物体との距離は推定することが出来ないが、レンズ中心と対象物体を結ぶ直線は 3 次元のベクトルとして推定することが出来る。よって、同じ対象物体を捉えるカメラが 2 台あれば、各カメラと対象物体を結ぶ直線の交点を計算することによって、対象物体の 3 次元座標を求めることが出来る。

しかし、現実には、カメラから得られる情報には、量子化誤差や観測誤差等が含まれており、同じ物体を 2 台のカメラが捉えていたとしても、それらのレンズ中心と対象物体を結ぶ 3 次元直線が必ず交点を持つとは言えない。それぞれの直線がなす距離が最小となる点を推定位置とする考えもあるが、計算が繁雑となる。

そこで、本研究では、対象物体がある一定の平面上を動くとして仮定し、カメラと対象物体を結ぶ直線を平面上に射影し、2 次元的に解くことで交点を計算し、対象物体の位置推定を行う。

3.2.1 対象物体方向の推定

Zhang のカメラモデルに基づいた、対象物体の画像上の位置情報から、カメラに対する対象物体の相対方向の推定方法について述べる。先に述べた通り、この時、対象物体はある一定の平面の上を移動するものとし、最終的に推定する物体位置もその平面上の 2 次元座標とする。

Zhang のカメラモデルより、画像座標系における 2 次元座標 $C_{obj}(u_{obj}, v_{obj})$ は、カメラ座標系ではカメラのレンズ中心からの 3 次元ベクトルとして式 (15) のように表現出来る。

$$\vec{C}_{obj} = \left(\frac{u_{obj} - c_x}{f_x}, \frac{v_{obj} - c_y}{f_y}, 1 \right)^T \quad (15)$$

この 3 次元ベクトルを元にカメラ座標系における値を求めれば、対象物体の相対方向を求めることが出来る。しかし、画像座標系の値から分かるのは、ベクトルだけであり、対象物体までの距離は分からない。

ここで、対象物体はある一定の平面の上を移動すると仮定する。するとカメラのレンズ中心から対象物体への 3 次元ベクトルは、2 次元平面に射影することが出来る。そこで、物体が移動する平面に垂直でカメラと水平な平面 B を考え (図 10)、平面 B と式 (15) で示されるベクトルとの交点 $(x'_c, y'_c, z'_c)^T$ を求める。

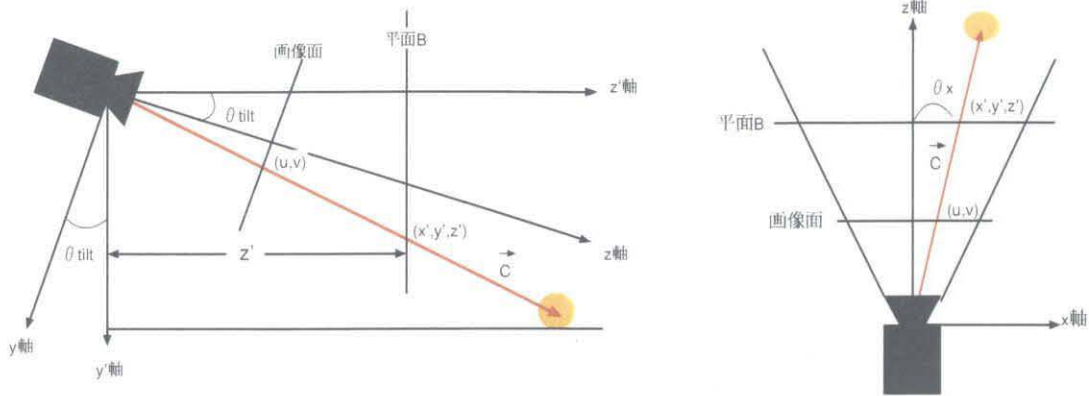


図 10 方向推定のモデル

平面 B はカメラに対してどの距離に置いても、 \vec{C}_{obj} が交わる点 $(x'_c, y'_c, z'_c)^T$ の各値の比は変わらないので、レンズ中心との位置関係から相対的な方向を求めることが出来る。この時の、点 $(x'_c, y'_c, z'_c)^T$ を算出する計算式は、式 (16) より求められる。

$$\begin{pmatrix} x'_c \\ y'_c \\ z'_c \end{pmatrix} = R(-\theta_{tilt})^T \begin{pmatrix} \frac{u_{obj} - c_x}{f_x} \\ \frac{v_{obj} - c_y}{f_y} \\ 1 \end{pmatrix} \quad (16)$$

$$R(-\theta_{tilt}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_{tilt}) & -\sin(-\theta_{tilt}) \\ 0 & \sin(-\theta_{tilt}) & \cos(-\theta_{tilt}) \end{pmatrix} \quad (17)$$

ここで $R(-\theta_{tilt})$ は、カメラのチルト角による回転行列である。また、式 (16) より求められた点 $(x'_c, y'_c, z'_c)^T$ より、カメラから見た対象物体方向 θ_{obj} は式 (18) で求められる。

$$\theta_{obj} = \tan^{-1} \left(\frac{x'_c}{z'_c} \right) \quad (18)$$

以上より、カメラから見た対象物体の方向、つまりカメラのレンズ中心と対象物体がなす相対角を求める。ここで求められた各カメラと対象物体の相対角より、それらを結ぶ直線の求め、カメラ対での交点を計算するが、その前に、本手法では、2 台のカメラの画像情報より位置推定を行う為、画像取得時刻について考える必要がある。次に、情報取得タイミングが非同期なシステムの考慮について述べ、ある時刻での物体方向の推定方法について述べる。

3.2.2 情報取得タイミングが非同期なシステムの考慮

情報取得タイミングが非同期なシステムを考慮し、各カメラで取得した情報のある時刻の情報へと揃える方法について述べる。これは、カメラが情報を取得するタイミングが、処理の周期が一定

でないことによりずれたり、カメラによって取得した情報を情報統合を行うタスクへと送信する際の通信遅延の問題により、カメラの時刻同期が正確に出来ていても、非同期な情報となってしまう問題を考慮するものである。

本研究室では、黒田 [2] が Luo[5] らの方法を取り入れ、直前の情報から対象物体の世界座標系における、対象物体の単位変化量を求め、線形補完を行う方法がとられてきた。しかし、この手法は、世界座標系における単位変化量を求めるものである為、提案するカメラ対を用いた対象物体の位置推定に対応できない。これは、カメラ対を用いた位置推定手法は、2 台のカメラで取得された情報より位置推定を行っていることから、位置推定の処理を行う前に、2 台のカメラ情報間の同期を行う必要がある為である。

そこで、本研究では、各カメラにおいて、対象物体方向の推定を行う際に求めた x_c の値について、直前の値との単位変化量を求め、線形補間をすることによって、ある時刻 t における対象物体の方向を推定する方法をとる。この時の単位変化量は以下の式 (19) で表す。

$$dx_{obj} = \frac{x_{c_k} - x_{c_{k-1}}}{t_k - t_{k-1}} \quad (19)$$

ここで、 x_{c_k} は k 番目の各カメラにおける対象物体のカメラ座標系 x 座標。 t_k は k 番目の情報が時刻 t に得られたことを示す。この単位変化量より、時刻 t から t_p 時間経過した時の対象物体の方向 θ_{t_p} を、以下の式 (20) で表す。

$$\theta_{t_p} = \tan^{-1} \left(\frac{(t_p - t)dx_{obj} + x_{c_t}}{z'_c} \right) \quad (20)$$

ここでの z'_c は方向推定の際に用いた z'_c と同じ値である。

3.2.3 交点の計算

対象物体方向の推定により、各カメラと対象物体がなす相対角が分かった。ここでは、その情報を用いて、対象物体の世界座標系における値を推定する方法について述べる。位置推定におけるモデルを図 11 に示す。

各カメラについて、カメラの世界座標系における位置と向きが分かっているものとする、カメラ 1(x_1, y_1) とカメラ 2(x_2, y_2) に観測される物体の位置 (x_{obj}, y_{obj}) は、以下の式 (22) で求める。

$$x_{obj} = \frac{(y_2 - x_2 \tan \theta_2) - (y_1 - x_1 \tan \theta_1)}{\tan \theta_1 - \tan \theta_2} \quad (21)$$

$$y_{obj} = x_{obj} \tan \theta_1 + (y_1 - x_1 \tan \theta_1) \quad (22)$$

ここで、 θ_1, θ_2 は世界座標系において、カメラと対象物体を結んだ直線が世界座標系 x 軸に対してなす角度であり、前節で求めた相対角とカメラの向きによって求めることが出来る。但し、 θ_1

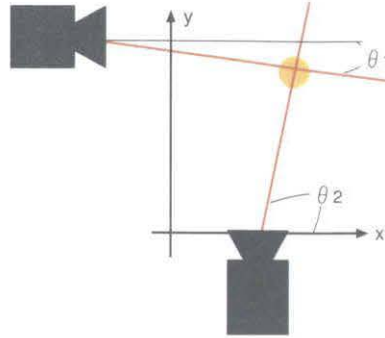


図 11 位置推定のモデル

又は, θ_2 のどちらかが 90° の時は, そのカメラの世界座標系における位置を (x, y) , もう一方のカメラの位置を (x', y') , 対象物体と結んだ直線が x 軸に対してなす角を θ として, 以下の式 (24) により求める.

$$x_{obj} = x \quad (23)$$

$$y_{obj} = x \tan \theta + (y' - x' \tan \theta) \quad (24)$$

尚, 本手法では, 対象物体の位置を推定するのにロボットが最低でも 2 台必要であり, 2 台以上のロボットがある場合では, 各ペアによって位置推定を行うので, n 台のロボットがある場合には, ${}_nC_2$ 個の位置推定結果が得られる.

3.3 実験

本研究室のロボット 3 台を用いて実際に測定した位置と推定結果との誤差を調べることによって精度を確かめた.

3.3.1 実験方法

ロボット 3 台 (sakura, daisen, kaimon) を用いて, 実際に測定した 19 箇所にボールを置き, 観測させ位置推定を行う. この時のロボットは 3 台ともキャリブレーション済^{*2}であり, 画像の歪曲収差はキャリブレーション結果により補正している. 実験におけるロボットの配置を図 12 に示す. 図 12 に示される様に, ロボットは正三角形を形成するように配置し, 三角形の重心方向を向いている. また, 観測対象のボールは, ロボットにより形成された三角形の中 (図 12 の斜線領域) で位置を変えて配置した.

^{*2} Zhang のキャリブレーション手法については付録を参照されたい

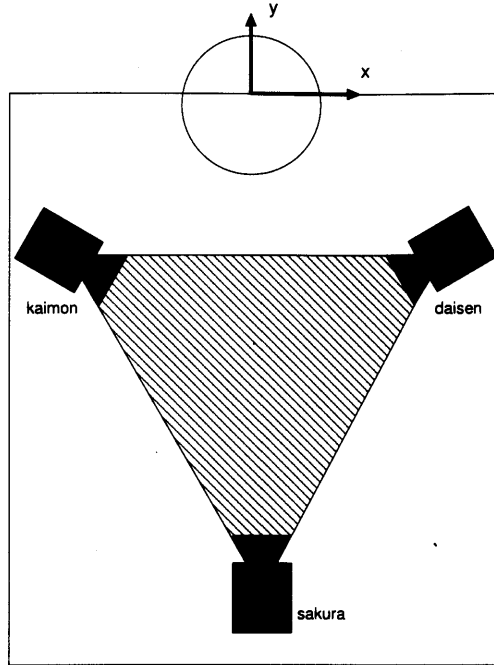


図 12 実験環境

実験を行った位置推定の手法としては、本章で提案した手法に加え、比較対象として、本研究室でこれまで使用されていた、画像上の 2 次元座標値と相対距離との関係を関数で近似して計算する手法と、対象物体の高さが分かっていると仮定し、画像情報から 3 次元座標を計算する手法を用い、結果の比較を行った。尚、この時に使用した距離に関する近似関数は以下の通りである。

距離が近いとき (画像上で対象物体の重心がより下にあるとき (daisen))

$$dist = -5.06912 \times x + 1360.37 \quad (25)$$

距離が遠いとき (画像上で対象物体の重心がより上にあるとき (daisen))

$$dist = 746281 \times x^{-1.45239} \quad (26)$$

ここで x はボールの画像座標での重心位置である。他の 2 台のロボットについても同様に求め、距離と角度から位置を決定する。

また、画像情報から 3 次元座標の計算は、対象物体が高さ h のものであると仮定する。その後、画像情報から得られる対象物体への 3 次元ベクトル $(\frac{u_{obj}-c_x}{f_x}, \frac{v_{obj}-c_y}{f_y}, z_c)^T$ と、地面から高さ h で

地面と平行な平面との交点を計算することによって求める。また、カメラのレンズ中心の高さを y_0 とすると、この時のカメラ座標系における交点位置を $(x''_c, y_0 - h, z''_c)^T$ とすると、その関係は以下の式 (27) で表され、求めることが出来る。

$$\begin{pmatrix} x''_c \\ y_0 - h \\ z''_c \end{pmatrix} = R(-\theta_{tilt})^T \begin{pmatrix} \frac{u_{obj} - c_x}{f_x} \\ \frac{v_{obj} - c_y}{f_y} \\ z_c \end{pmatrix} \quad (27)$$

式 (27) で求まったカメラ座標系の値を、キャリブレーション結果を用い、世界座標系に変換することで、位置推定を行う。

3.3.2 実験結果

実験結果と、それらの位置推定の誤差を距離で表し、プロットしたものを図 13 から図 24 に示す。また、各手法による誤差の平均と標準偏差を示したものを表 3 に示す。尚、各実験結果の誤差を示す図の横軸は、各実験結果のデータがプロットされている点に、左から右へと順に対応している。

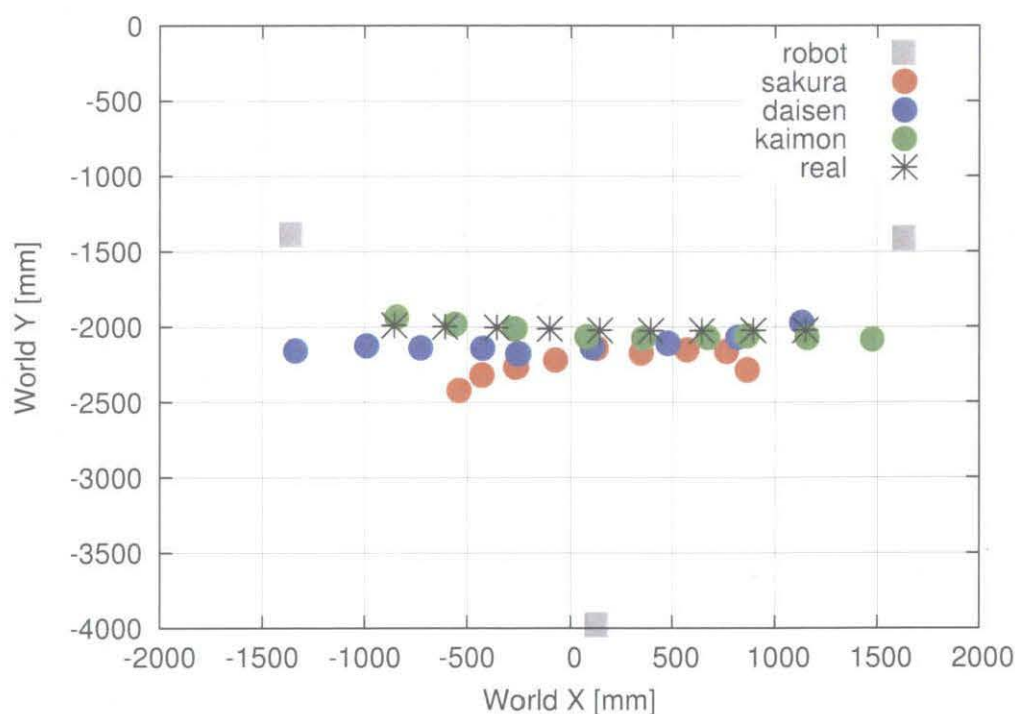


図 13 距離関数を使った時の位置推定結果 1

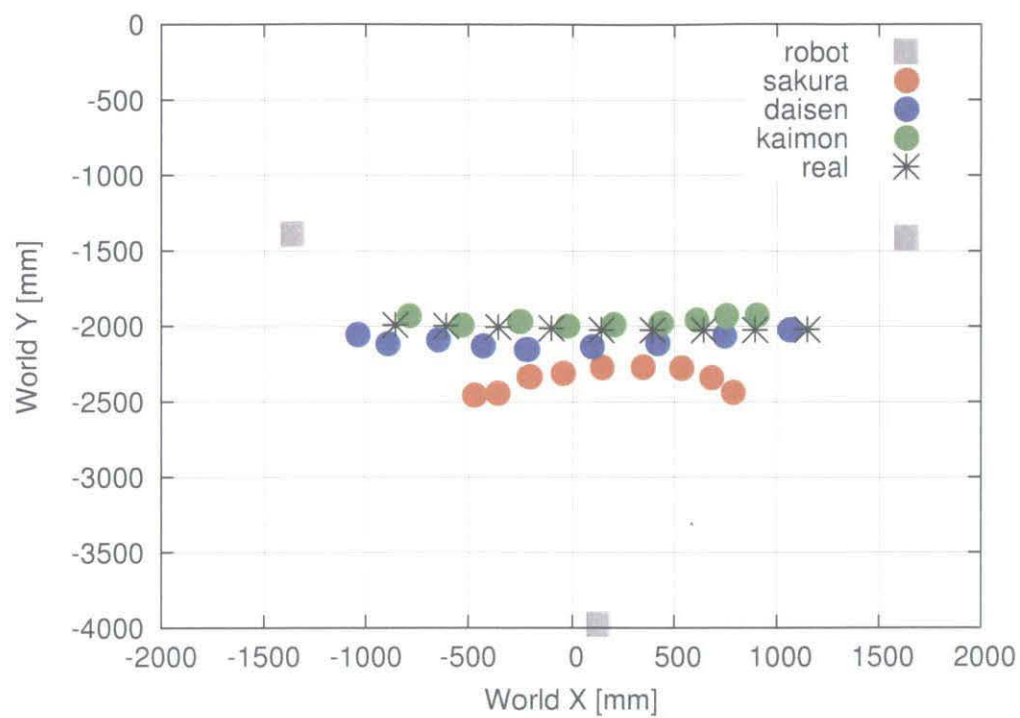


図 14 対象物体方向と平面の交点による位置推定結果 1

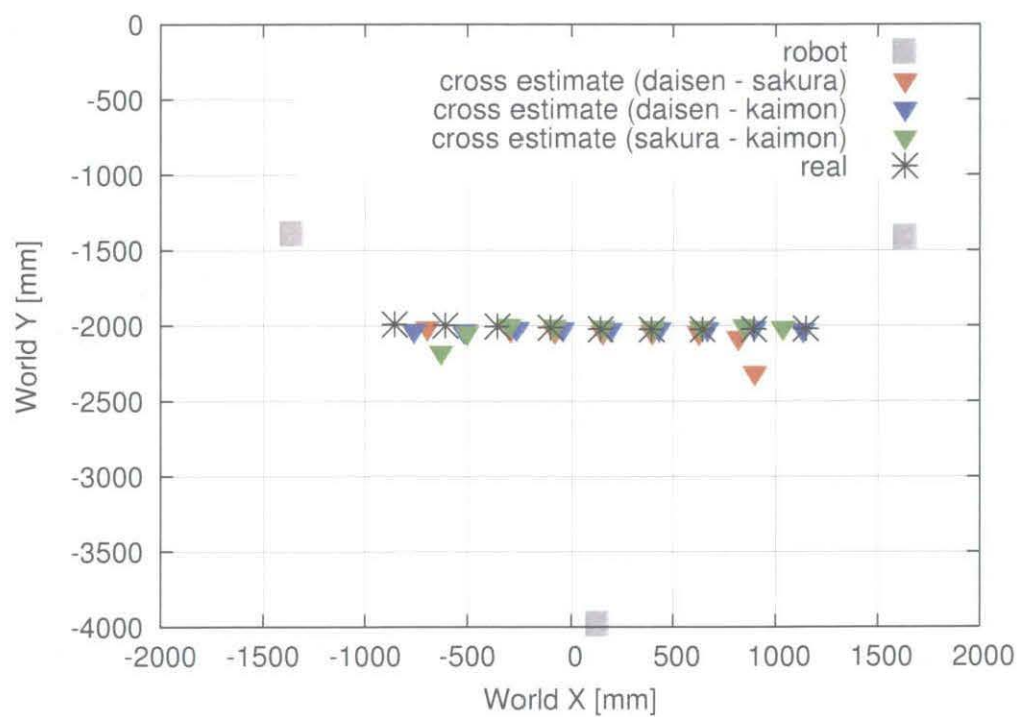


図 15 カメラ対の物体方向の交点を用いた位置推定結果 1

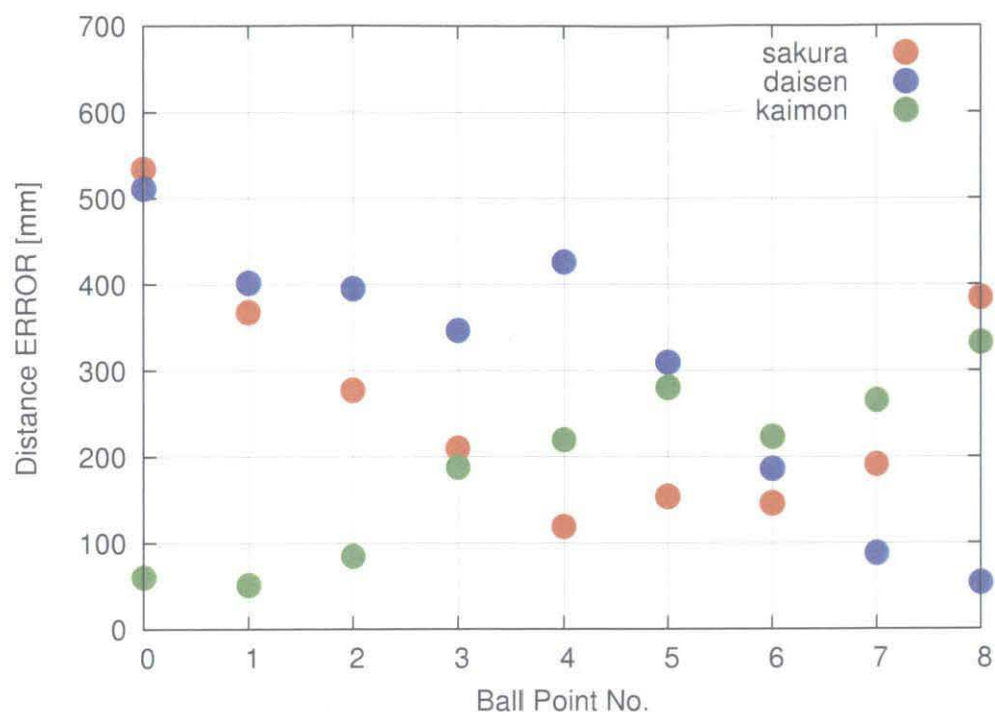


図 16 距離関数を使った時の推定結果の誤差 1

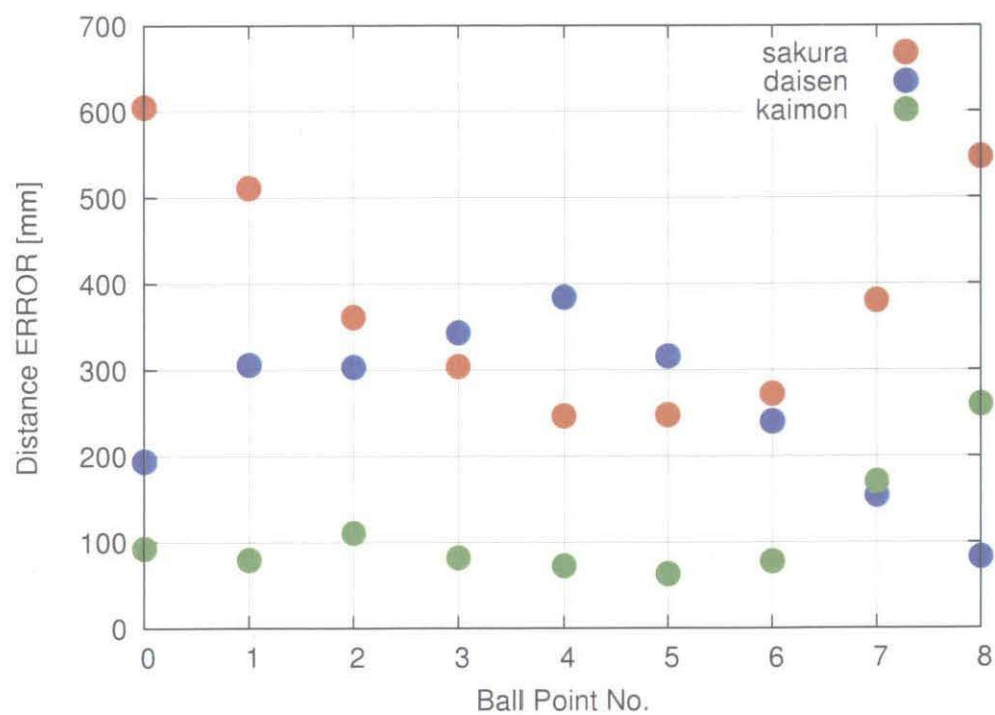


図 17 対象物体方向と平面の交点による推定結果の誤差 1

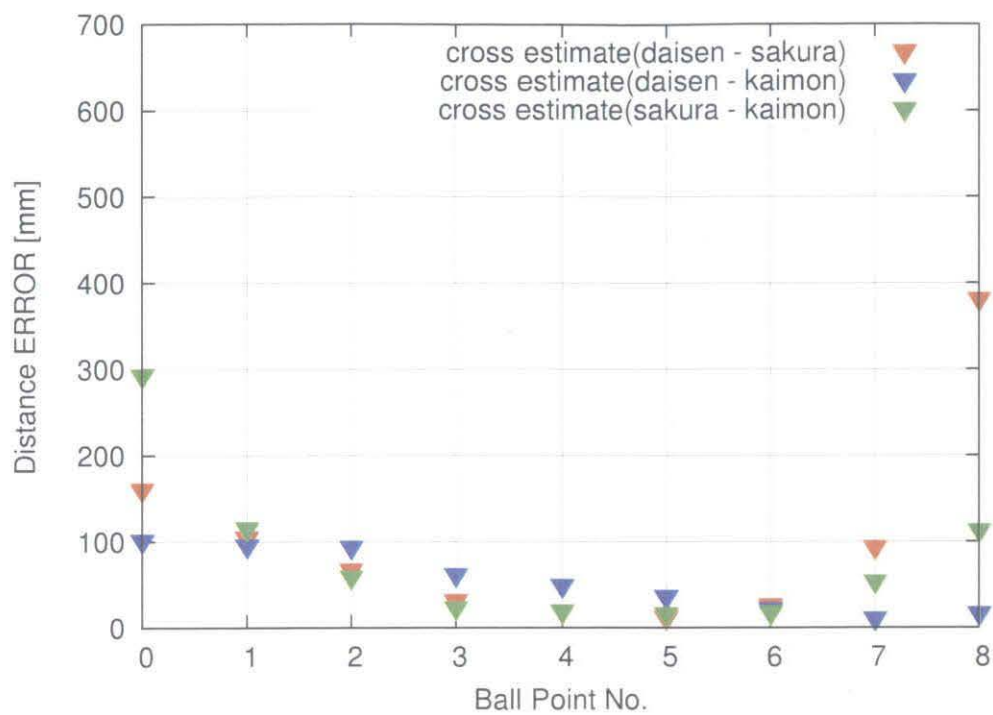


図 18 カメラ対の物体方向の交点を用いた位置推定結果の誤差 1

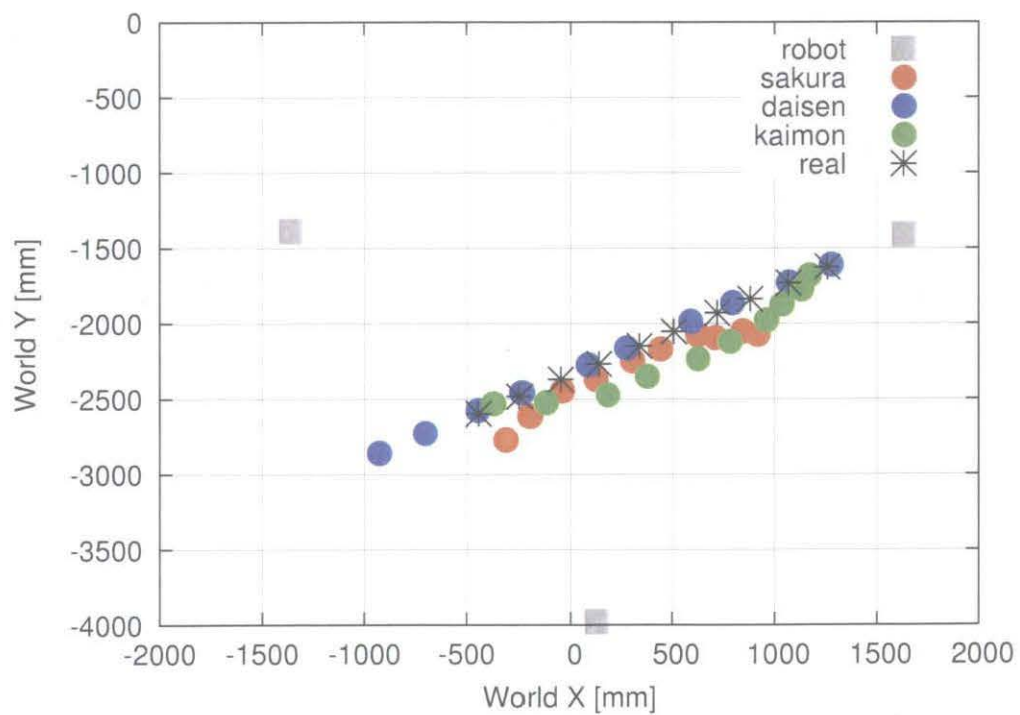


図 19 距離関数を使った時の推定結果 2

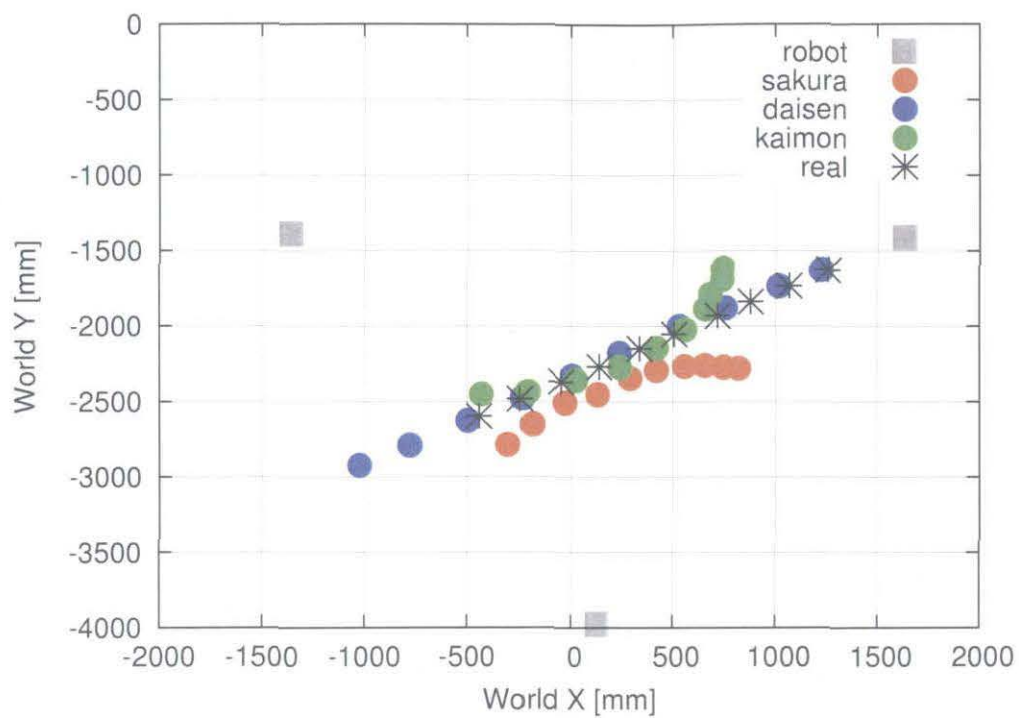


図 20 対象物体方向と平面の交点による位置推定結果 2

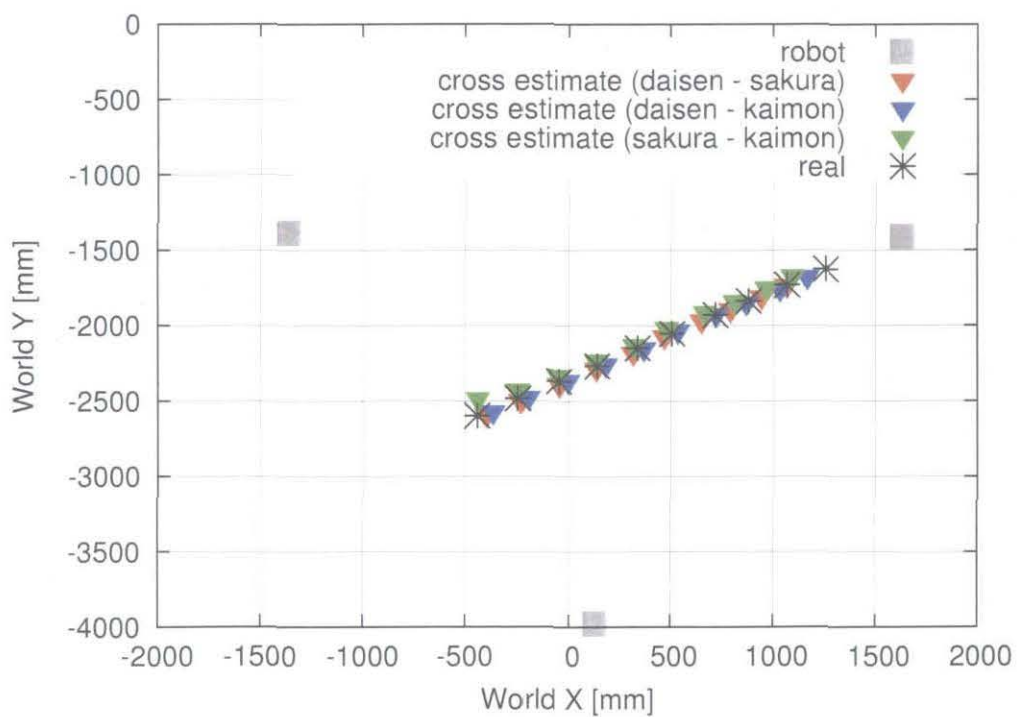


図 21 カメラ対の物体方向の交点を用いた位置推定結果 2

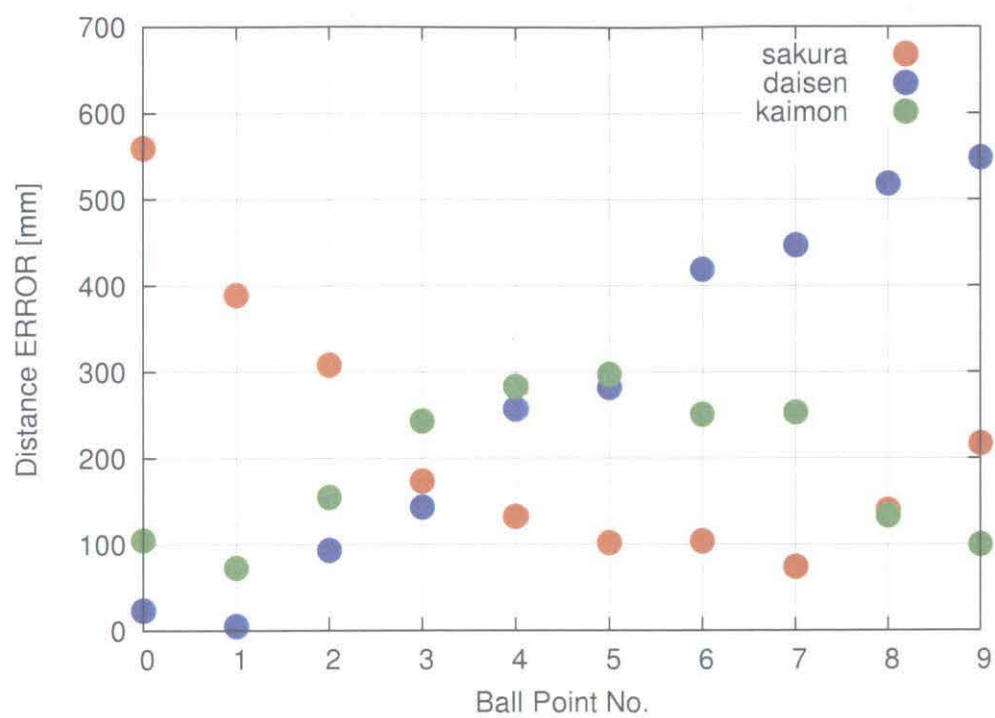


図 22 距離関数を使った時の推定結果の誤差 2

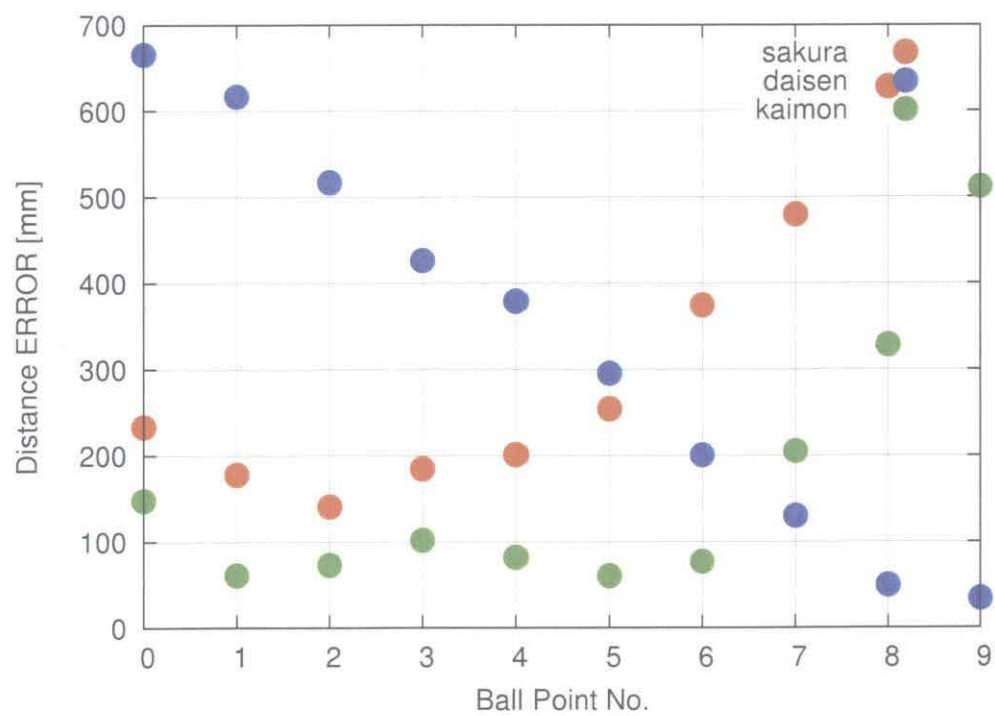


図 23 対象物体方向と平面の交点による推定結果の誤差 1

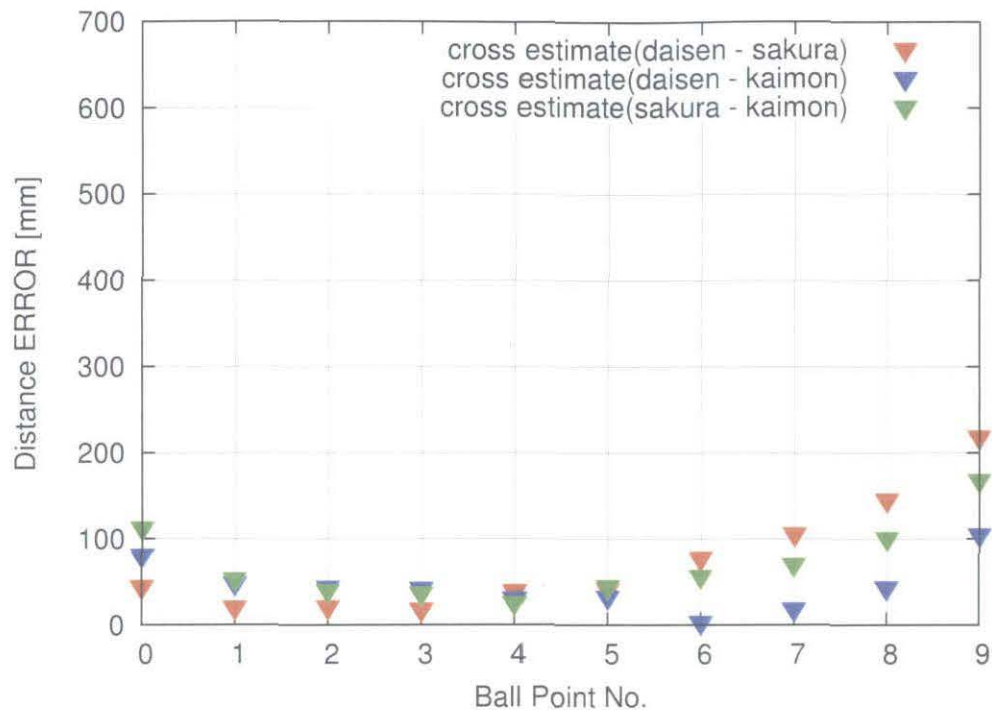


図 24 カメラ対の物体方向の交点を用いた位置推定結果の誤差 1

表 3 誤差の平均と標準偏差 (単位は mm)

実験 1	平均	標準偏差	実験 2	平均	標準偏差
距離関数	252.2378	138.3484	距離関数	227.672	155.4907
平面との交点	252.2361	151.5609	平面との交点	280.8418	211.2159
カメラ対の交点	77.6665	85.9394	カメラ対の交点	63.2122	48.7368

3.3.3 考察

結果から見て分かるように物体方向の交点を用いた推定結果の方が、他の 2 手法に比べても、より正確で安定した精度で推定が出来ている。このことから、各ロボットがカメラを 1 台しか搭載していない場合、本手法は有効な位置推定手法であると言える。但し、実験結果 15 に示されるように、カメラ対の直線がなす角度が小さくなる場合には、精度が落ちることがあるので、これらの場合は対処を考慮する必要がある。また、対象物体が画像の辺縁部に写る場合には、どの位置推定方法においても精度が大きく低下する現象が見られた。これは、キャリブレーション結果を用いてもカメラの歪曲収差を補正しきれていない為であると考えられ、より多くのパターン画像を用いてより正確なキャリブレーションを行う必要があると考えられる。

第4章

情報統合

本章では情報統合について述べる。本研究での情報統合とは、地理的位置が異なる複数センサの情報を統合し、より正確な情報を得ることである。情報統合では、様々なアングル、距離からの情報を得ることが出来るが、各センサが含む誤差も場所により変化する。

従来研究として、センサと対象物体迄の距離により重みをつけた算術平均による方法 [2], [12] や、統合に至るまでの誤差を正規分布に近似し統合する方法 [7], 同じく誤差を正規分布に近似しベイズ則に基づき統合する方法 [9], [10], カルマンフィルタを用いて統合する方法 [6] 等がある。これらの研究では、ロボットが独自に対象物体の世界座標値推定が出来るものとして研究が行われているが、第3章で述べた様に、ロボット単体で正確な世界座標値の推定を行うには、複数のセンサを取り付ける必要がある。

そこで本研究では、統合に至るまでの誤差を正規分布に近似することで考慮し、線形に計算が可能である Stroupe らの方法 [7] を参考に統合を行う。但し、どちらもそのままでは、第3章で述べたカメラ対による位置推定に対応出来ないので、本研究における位置推定に対応した手法へと拡張したものを提案する。

4.1 正規分布を用いた統合

Stroupe らの [7] 正規分布を用いた観測情報の情報統合手法について述べる。論文 [7] によるとカメラによって観測される情報の誤差は、正規分布に近似することが出来る。この時のモデルを図 4.1 に示す。図 4.1 に示されるように、対象物体の推定位置は、距離と方向の誤差により 2 次元の正規分布で表現される。この 2 次元の正規分布は、位置推定された点を中心として分布する。尚、この分布は 2 次元座標でみると、楕円形をしている。

この時の、各標準偏差 σ_{maj} , σ_{min} は、各カメラと対象物体との距離と、各カメラと対象物体との相対角によって決定される。これは、距離が離れることによって観測誤差や量子化誤差が大きくなる為である。

各観測点における分布を表す式を式 (28) に示す。

$$p(X) = \frac{1}{2\pi\sqrt{|C|}} \exp\left(-\frac{1}{2}(X - \bar{X})^T C^{-1}(X - \bar{X})\right) \quad (28)$$

ここで C は共分散行列であり、以下の形をしている。

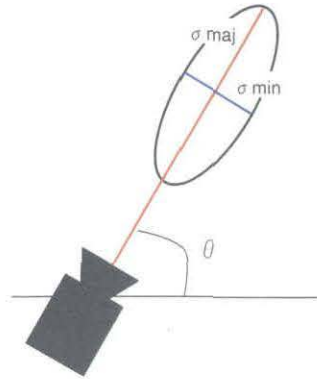


図 25 誤差のモデル

$$C = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \quad (29)$$

ここでの σ_x , σ_y は世界座標系の x 軸, y 軸における対象物体の分布の標準偏差であるので, 各カメラと対象物体との位置関係から求めた標準偏差 σ_{maj} , σ_{min} はそのまま用いることが出来ない. そこで, カメラと対象物体を結ぶ直線を分布の主軸として, 共分散行列 C_L を考えると, C_L は以下のように表される.

$$C_L = \begin{pmatrix} \sigma_{maj}^2 & 0 \\ 0 & \sigma_{min}^2 \end{pmatrix} \quad (30)$$

カメラと対象物体を結ぶ直線が x 軸となす角 θ は分かっているので, θ を利用して, C_L から C への変換が可能となる. C_L から C への変換を以下に示す.

$$C = R(-\theta)^T C_L R(-\theta) \quad (31)$$

これで, 世界座標系の x 軸, y 軸における対象物体の分布の共分散行列が求まった.

次に, 正規分布の統合を考える. これは共分散行列 C と平均値 \bar{X} の統合として以下で線形に計算することが出来る.

$$C_{merge} = C_1 - C_1 (C_1 + C_2)^{-1} C_1 \quad (32)$$

$$\bar{X}_{merge} = \bar{X}_1 + C_1 (C_1 + C_2)^{-1} (\bar{X}_2 - \bar{X}_1) \quad (33)$$

C_1, C_2 は各カメラにおいて対象物体の推定位置の分布における, 世界座標系 x 軸, y 軸に対応した共分散行列であり, \bar{X}_1, \bar{X}_2 は, 各分布の平均値, つまり位置推定結果である. そして, 全ての分布を統合した後の \bar{X} の値を用いて, 統合結果とする. これは, 各 2 次元分布を統合し, 最尤法により対象物体の位置を推定することに相当する.

4.2 提案手法

本研究では, 先に述べた論文 [7] による手法を元に統合を行う. しかし, 本手法における位置推定では, カメラから対象物体への距離情報を用いていない為, 単純に同じ方法とはとれない. そこで, 前章で述べた対象物体方向の交点を利用した位置推定手法によって求められた対象物体位置に対し, 各々のカメラの方向推定の誤差を正規分布によって近似し, そこから得られる 2 次元分布を対象物体の推定位置の分布として表現する手法を提案する. 本手法におけるモデルを図 4.2 に示す.

図 4.2 に示される様に, 対象物体の推定位置の分布は, 位置推定された点を中心として 2 次元に分布する. 尚, この分布は 2 次元座標でみると, 従来手法と同じく楕円形をしている. そのモデルを図 4.1 に示す.

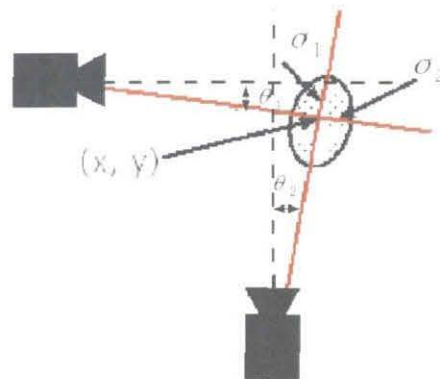


図 26 提案手法における誤差のモデル

尚, この時の各標準偏差 σ_1, σ_2 は, 各カメラと対象物体との距離と, 各カメラと対象物体との相対角によって決定される. 距離に関しては従来手法と同じ原理であり, 相対角については, 画像の端に対象物体が写るほど, キャリブレーションで補正しきれない歪曲収差の影響が出る為である.

各観測点における分布を表す式と, 共分散行列を表す式は従来手法と同じく, 式 (28), (29), (30) で表される.

4.2.1 分布の決定

式 (30) における σ_{min} , σ_{maj} は分布の長軸, 短軸における標準偏差である. しかし, 本研究での位置推定方法では, 2 台の各カメラと対象物体とを結んだ直線が直交することはまず無いので, 推定位置の分布の主軸を求める必要がある.

ここで, ある推定位置における分布の長軸が世界座標系の x 軸となす角を θ , 2 台の各ロボットと対象物体とを結んだ直線がなす角の鈍角を ϕ とすると (図 27), θ は ϕ , σ_1 , σ_2 との関係より, 式 (34) により決定する.

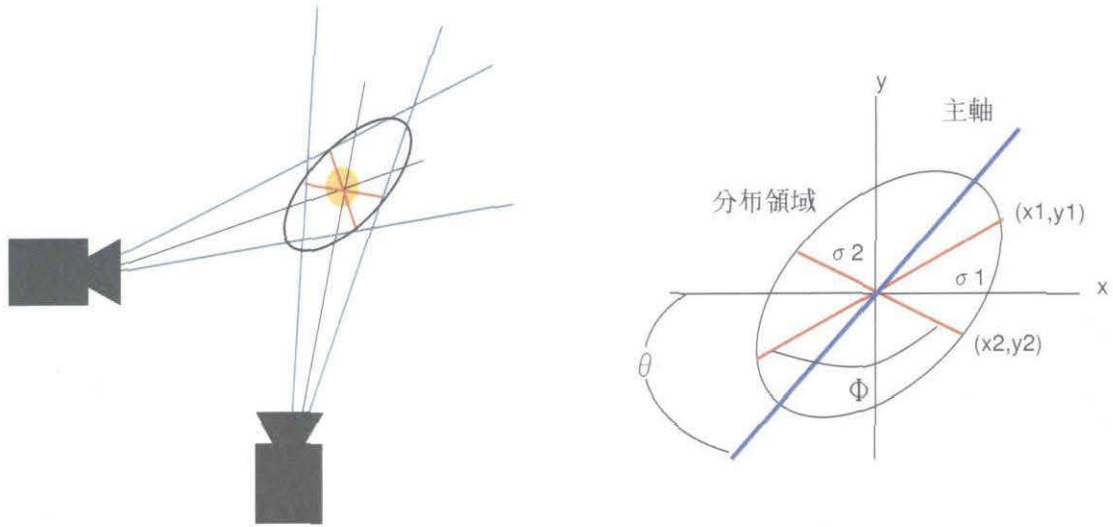


図 27 主軸の決定

$$\theta = \theta_1 + \frac{\sigma_2}{\sigma_1 + \sigma_2} \phi \quad (34)$$

これで θ が求まったので, 楕円の方程式より σ_{min} , σ_{maj} が式 (36) により決定出来る.

$$\sigma_{maj} = \frac{x_1}{\cos(\phi - \theta_1)} \quad (35)$$

$$\sigma_{min} = \frac{y_2}{\sin(\phi - \theta_2)} \quad (36)$$

ここで, θ_1 , θ_2 は, σ_1 , σ_2 の軸が x 軸となす角 (各カメラと対象物体を結んだ直線と直交する直線がなす角) であり, x_1 , y_2 は, σ_1 , σ_2 の軸上の点で, 対象物体の推定位置からそれぞれ σ_1 , σ_2 離れた点の x 座標と y 座標である.

また, θ が求まったことにより, C_L から C への変換が可能となり, 以下は, 従来手法の式 (31) から式 (33) により, 推定位置の統合を行う.

統合後の結果としては, 従来手法と同じく, 全ての分布を統合した後の \bar{X} の値を用いる.

4.3 実験

本研究室のロボット 3 台を用いて実際に測定した位置と推定結果との誤差を調べることによって精度を確かめた。

4.3.1 実験方法

ロボット 3 台 (sakura, daisen, kaimon) を用いて、実際に測定した 19 箇所にボールを置き、観測させ情報統合を行う。この時のロボットは 3 台ともキャリブレーション済である。実験におけるロボットの配置は図 12 と同じである。この時のロボットは 3 台ともキャリブレーション済である。また、本手法によって推定された位置の重心をとったもの、各ロボット独自で対象物体までの距離と世界座標位置を求め、その距離で重み付けをし、算術平均をとったもの、距離の誤差を正規分布に近似し、方向と距離の 2 次元の正規分布を統合したものを比較対象とした。

尚、距離で重み付けを行う算術平均による統合は以下の式で表すことが出来る。

各ロボットの観測に対する重みは、式 (37) で定義される。

$$w_i = \frac{1}{dist_i} \quad (37)$$

$$x_{merge} = \frac{1}{\sum_{i=0}^n w_i} \sum_{i=0}^n w_i \times x_k \quad (38)$$

$$y_{merge} = \frac{1}{\sum_{i=0}^n w_i} \sum_{i=0}^n w_i \times y_k \quad (39)$$

w_i は重みであり、 n はロボットの数。 x_{merge} 、 y_{merge} は統合した結果の世界座標系での位置。 x_k 、 y_k は各ロボットが持つ目標物体の世界座標系での位置である。

4.3.2 実験結果

実験結果と、それらの誤差を距離で表しプロットした図を図 28 から図 31 に示す。また、各手法による誤差の平均と標準偏差を示したものを表 3 に示す。尚、各実験結果の誤差を示す図の横軸は、各実験結果のデータがプロットされている点に、左から右へと順に対応している。

4.3.3 考察

実験結果の誤差の平均と標準を見ると、提案手法が他の 3 手法に比べ、最も誤差が小さく、また、安定した結果を示した。ボールの位置が各カメラにおいて画像の中心部に捉えられる場合は、

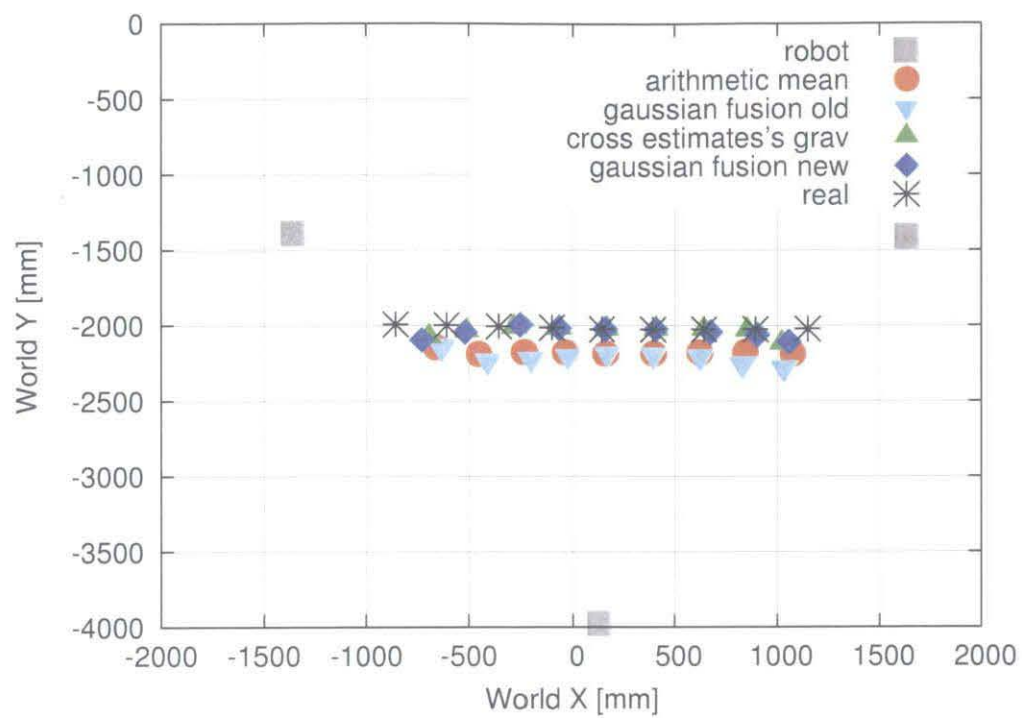


図 28 統合実験結果 1

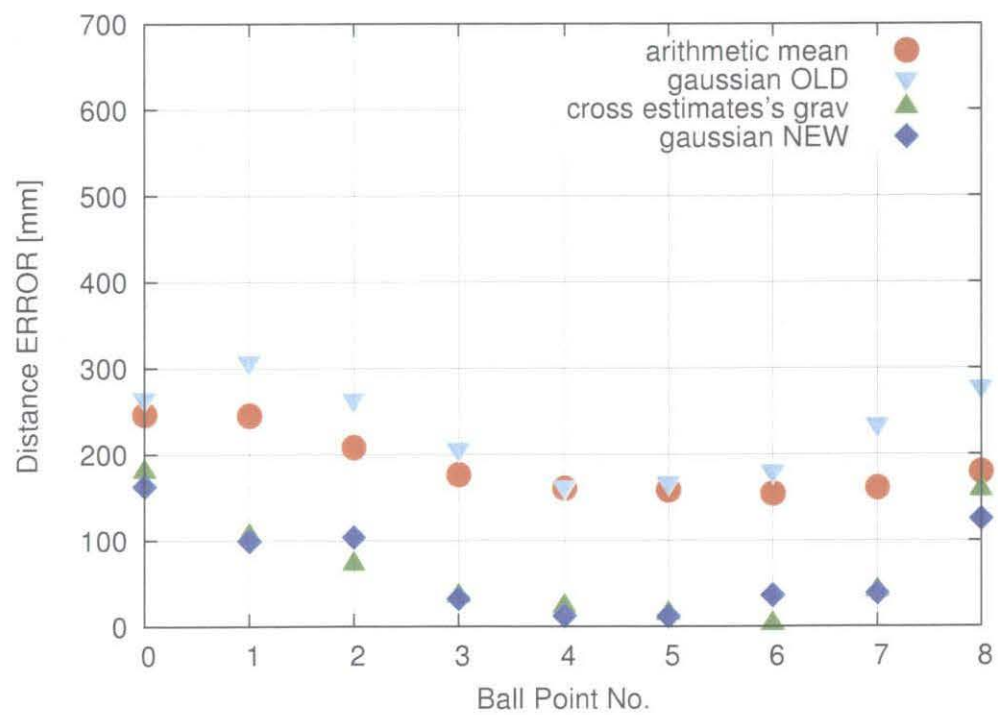


図 29 統合結果の誤差 1

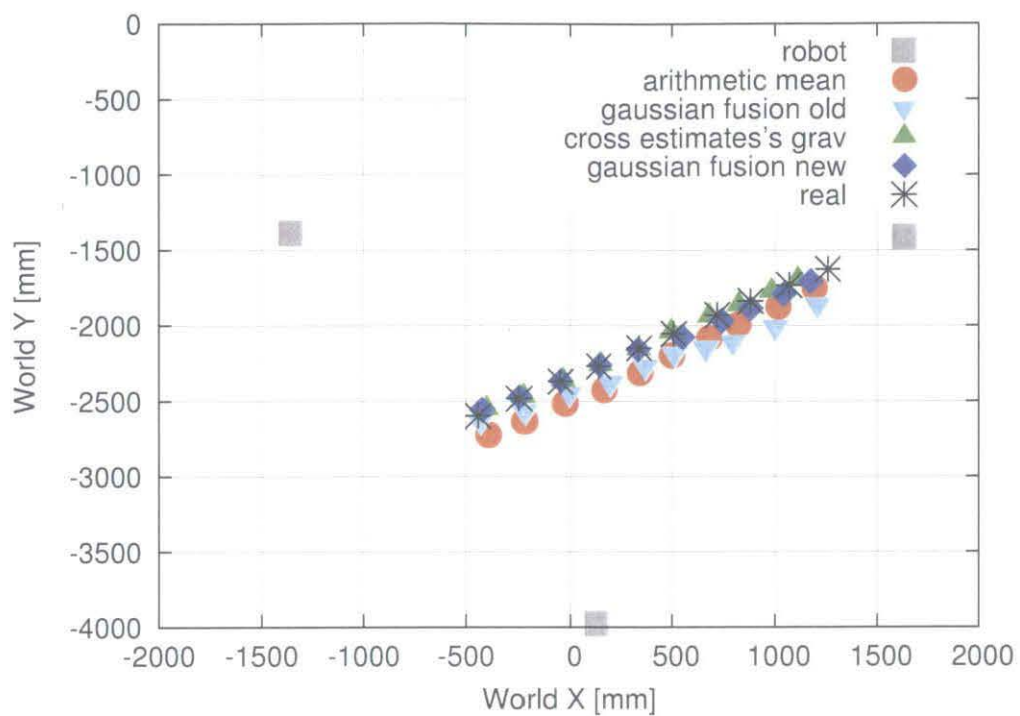


図 30 統合実験結果 2

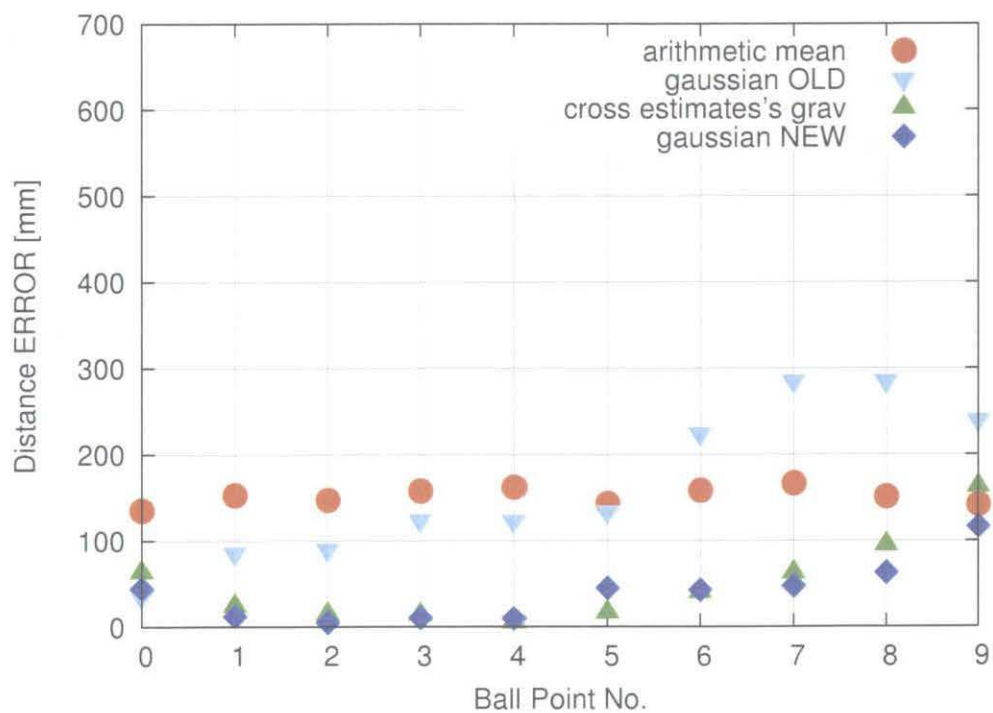


図 31 統合結果の誤差 2

表 4 誤差の平均と標準偏差 (単位は mm)

実験 1	平均	標準偏差	実験 2	平均	標準偏差
算術平均	187.91	36.7563	算術平均	151.6457	9.7938
正規分布 (従来手法)	229.3796	52.8624	正規分布 (従来手法)	162.3271	88.7501
重心	70.2966	64.2046	重心	49.8719	48.7407
提案手法	69.1143	54.512	提案手法	39.7232	33.6713

重心を用いて統合を行う方法の方が精度が高くなったが、画像の辺縁部に捉えられる場合には、提案手法の方が精度が良くなる傾向がみられた。これは、提案手法の、統合する際に誤差の考慮をしていることによるメリットが出た為であると考えられる。このことから、静止環境においては、画像中に散在する複数の対象物体の情報を同時に、且つ正確に取得する必要があるシステムには、本提案手法は有効な方法であると考えられる。

4.4 対象物体が動く環境における実験

これまで、対象物体が静止した環境について、位置推定の精度と情報統合の精度を高めることについて述べてきた。ここで、更に、ロボット 3 台 (sakura, daisen, kaimon) を用いて、ボールを転がし、対象物体が動作する環境における精度を確かめる。

ここで、ボールが動いている状況では、実際の位置を測定することが出来ないため、天井カメラを用いて位置推定することによって、精度の検証を行う。

尚、物体が静止した環境における天井カメラからの位置推定結果は、図 32 から図 35 に、誤差の平均と標準偏差を表 5 に示す。また、この時の統合結果と天井カメラの位置推定結果との誤差を図 36, 37 に示す。本実験において、図 36, 37 の誤差が目標値となる。

表 5 静止環境における天井カメラの誤差の平均と標準偏差 (単位は mm)

	平均	標準偏差
実験 1	17.2007	8.8373
実験 2	26.9532	9.5242

4.4.1 実験方法

実験におけるロボットの配置は図 12 と同じであり、図 38 に示されるように、ボールの転がし方について 2 通りの実験を行った。

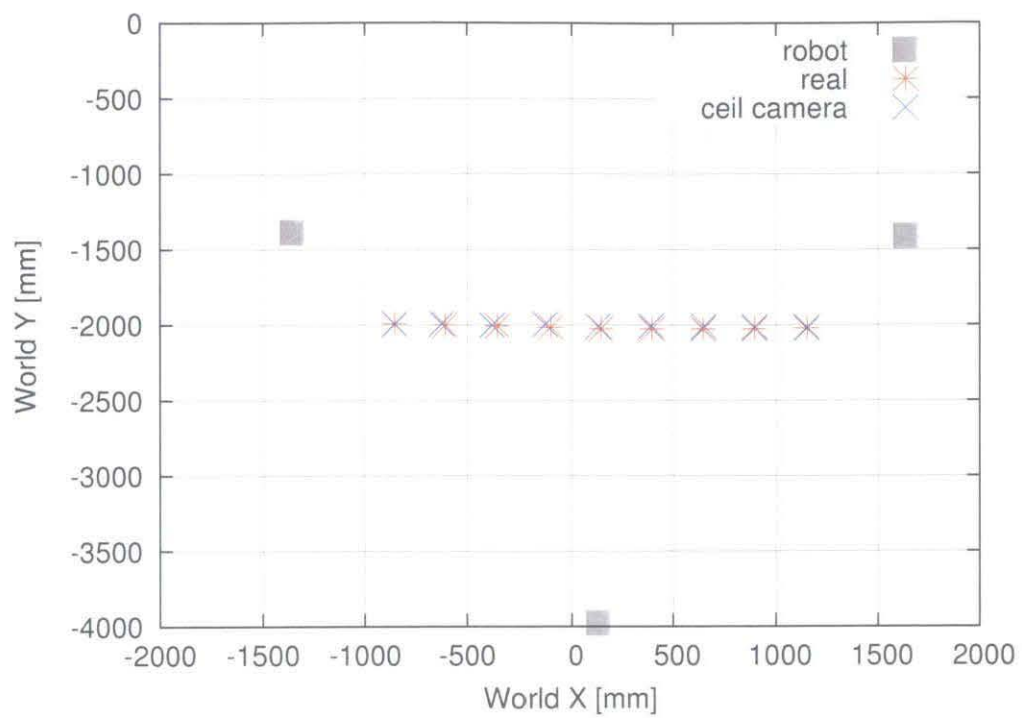


図 32 天井カメラによる位置推定結果 1

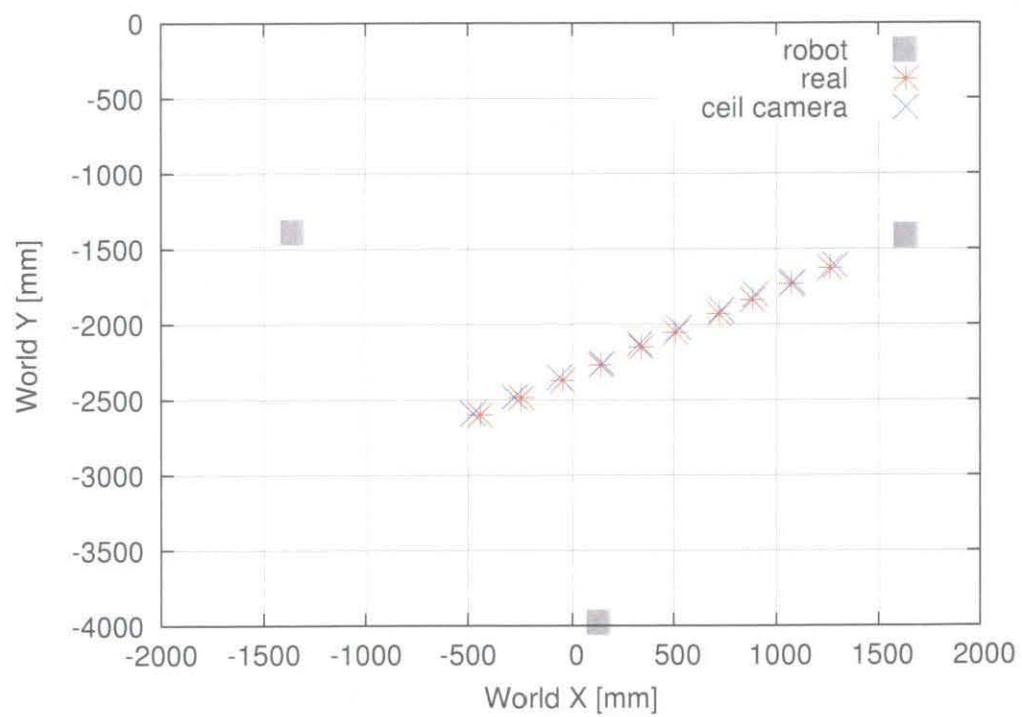


図 33 天井カメラによる位置推定結果 2

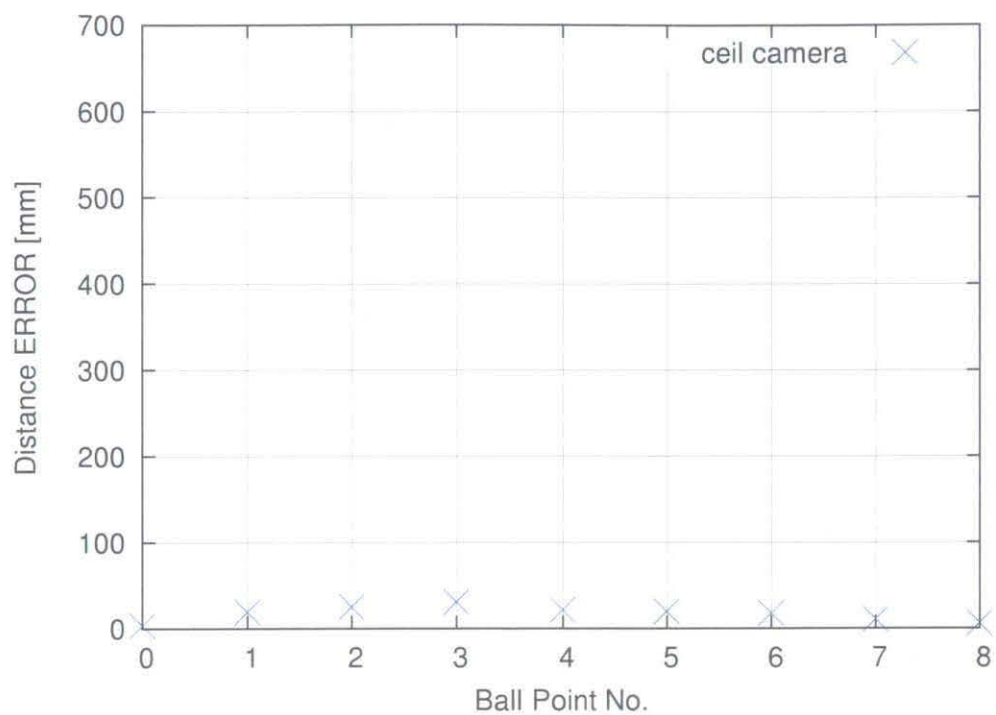


図 34 天井カメラによる位置推定誤差 1

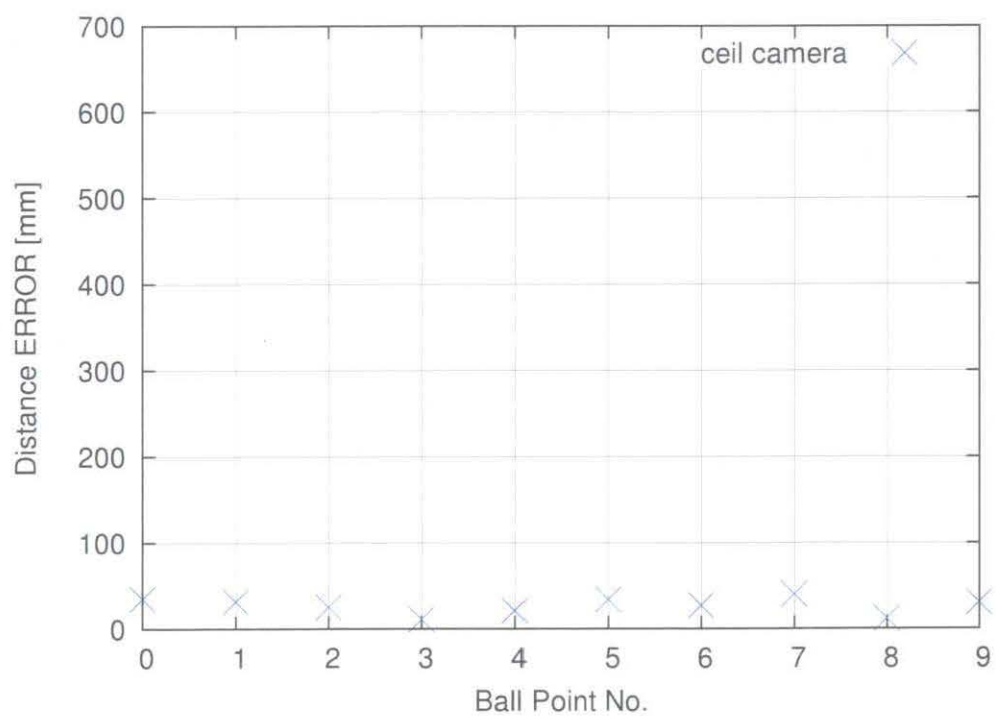


図 35 天井カメラによる位置推定誤差 2

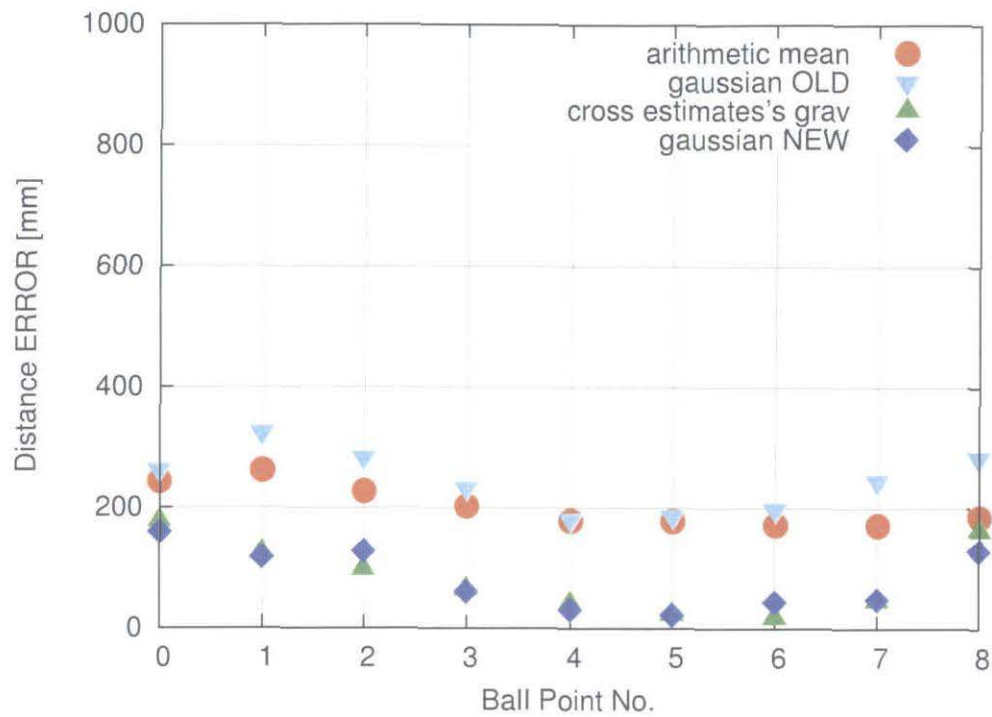


図 36 天井カメラと統合結果の誤差 1

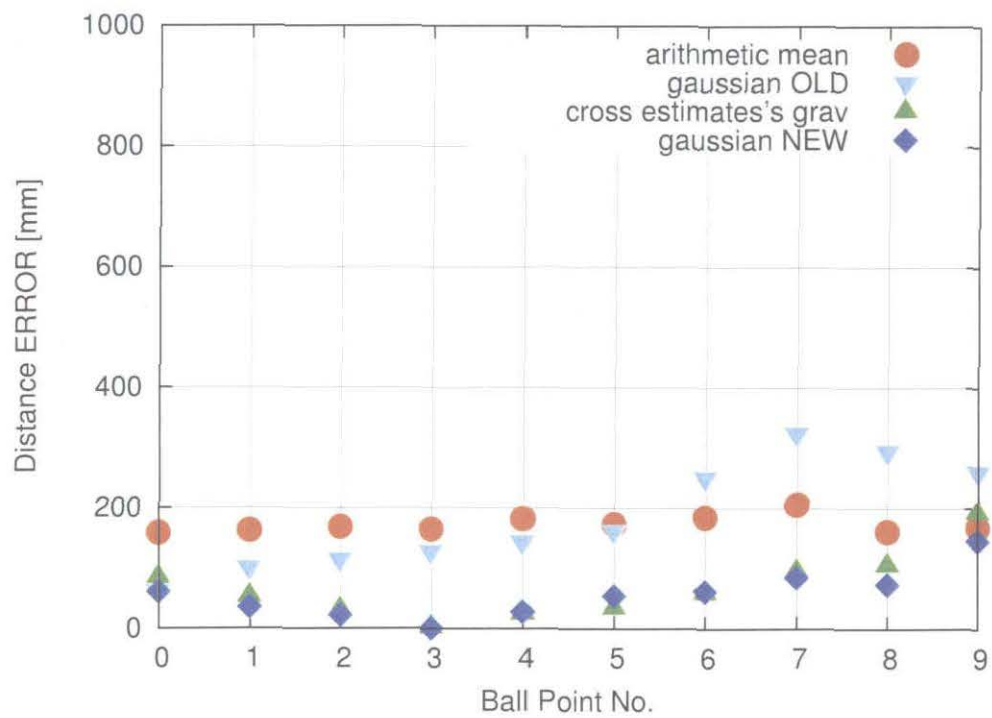


図 37 天井カメラと統合結果の誤差 2

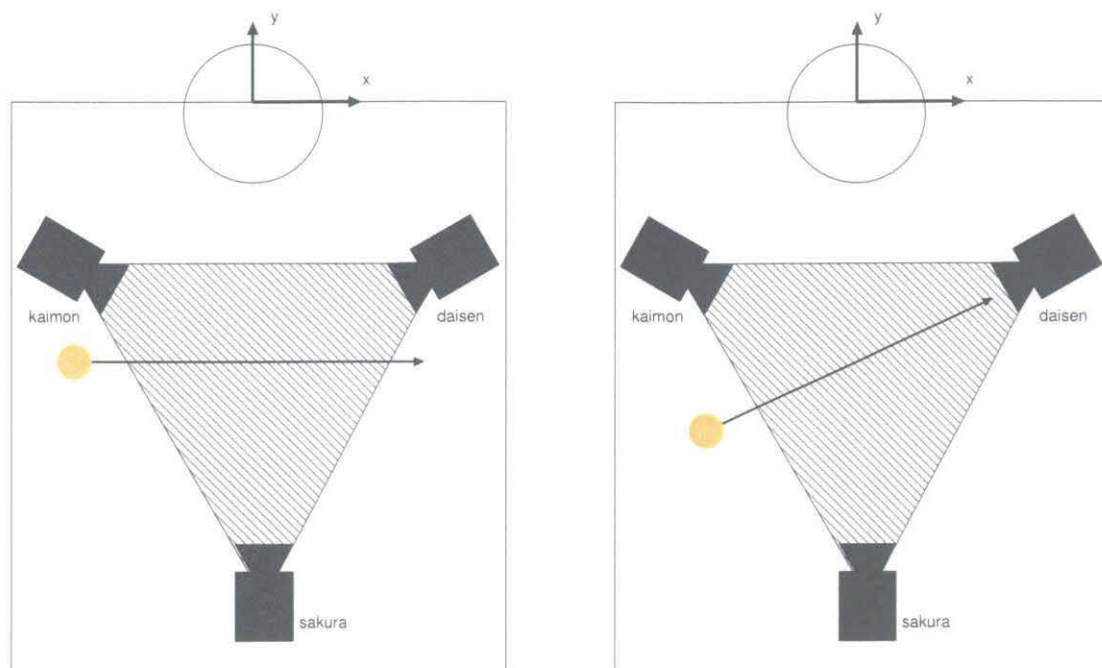


図 38 実験環境 2

4.4.2 実験結果

それぞれの実験結果を図 39, 40 に示す. ここでの天井カメラ (フレームレート 30fps) の値としてとっているものは, 30msec 毎に得られる画像から線形に補間したものである.

4.4.3 考察

実験結果から分かる通り, 対象物体を動作させた環境では誤差が大きく, また不安定であり, 良い結果が得られなかった. ここで, 原因を探る意味も込めて, 非同期システムを考慮して取り入れた, 方向推定時の線形補間をとりやめて, 同様に実験を行った. その結果を図 41, 42 に示す.

これらの実験では, システムの最新時刻と画像取得時刻の差を考慮していない為, 実際の位置より時系列において以前の位置が結果として得られる. その為, 誤差が大きくなる. 図 41, 42 から, その傾向が見受けられる. しかし, これらの結果からは, 実際のボールの動きを追従する傾向が見られ, 図 39, 40 の結果の傾向とは大きく異なっている.

また, 線形補間をした時とそうでない時のカメラ対による位置推定結果を図 43 から 46 に示す.

位置推定の各結果から分かるように, 補間をした場合, カメラ対の対象物体方向の交点を利用した位置推定結果が大きく悪くなっていることから, 原因は, 式 (19), (20) で定義した線形補間方法にあると考えられる. これは, 各カメラがその画像情報から, 対象物体方向の変化量を求め, 未来の時点 t における対象物体方向を推定するものであるが, 画像情報のみから推定している為, 正

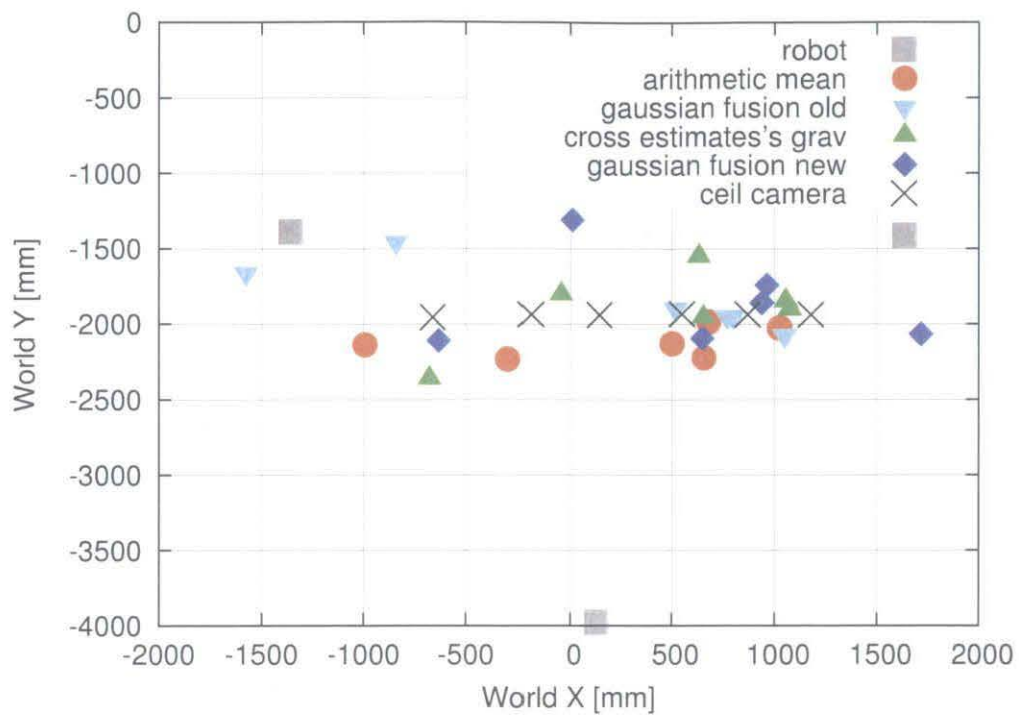


図 39 ボールを転がした時の統合結果 1

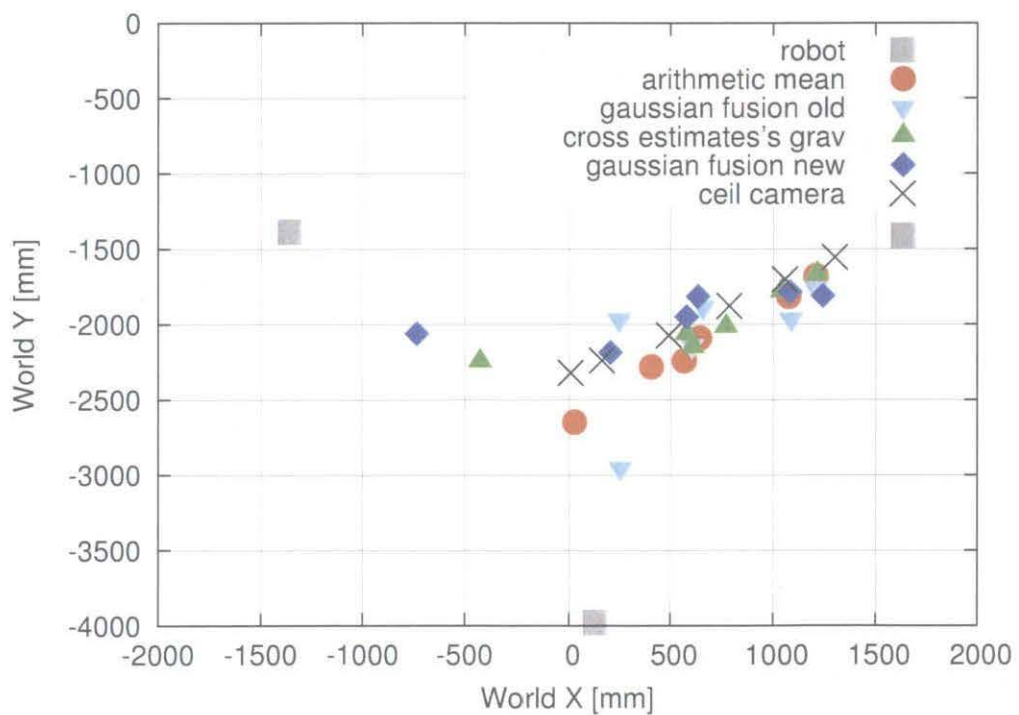


図 40 ボールを転がした時の統合結果 2

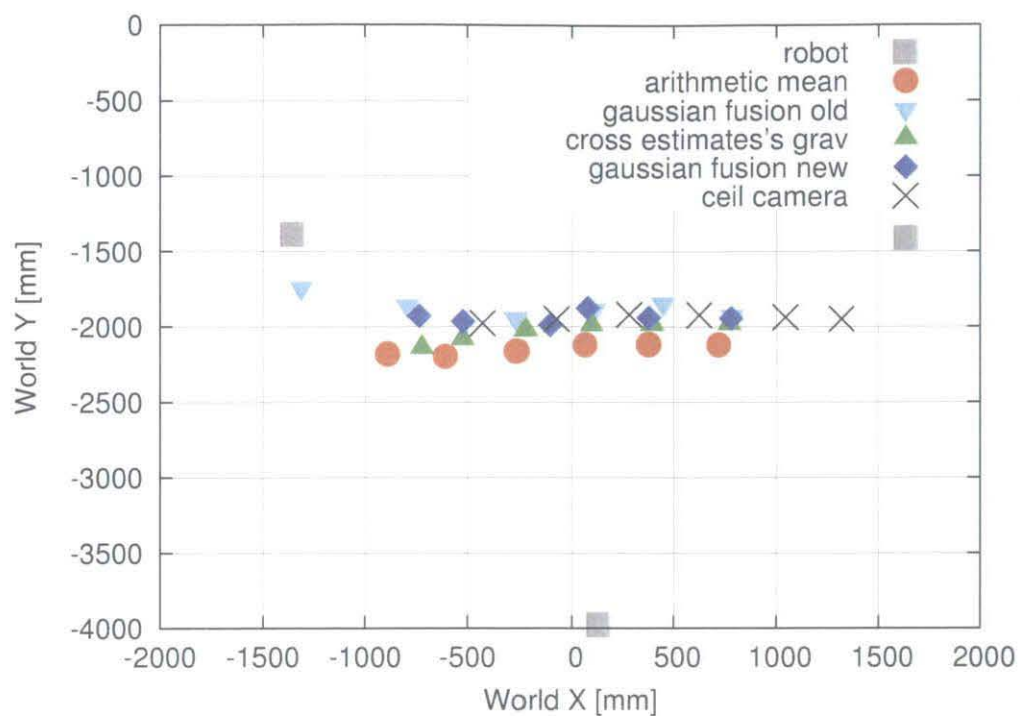


図 41 ボールを転がした時の統合結果 1(補間無し)

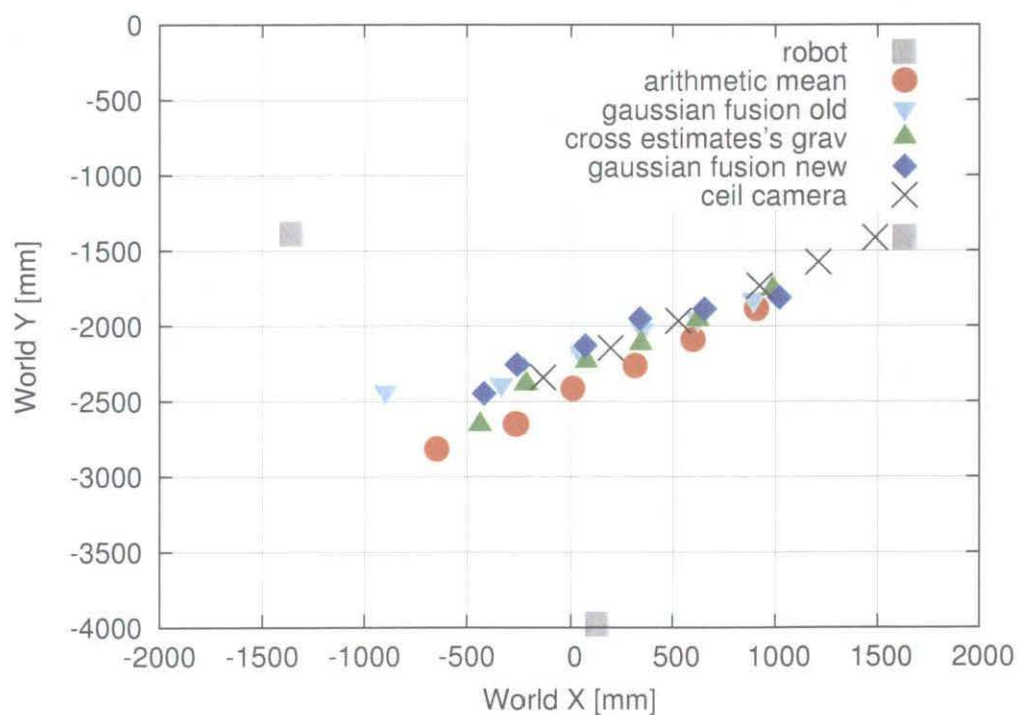


図 42 ボールを転がした時の統合結果 2(補間無し)

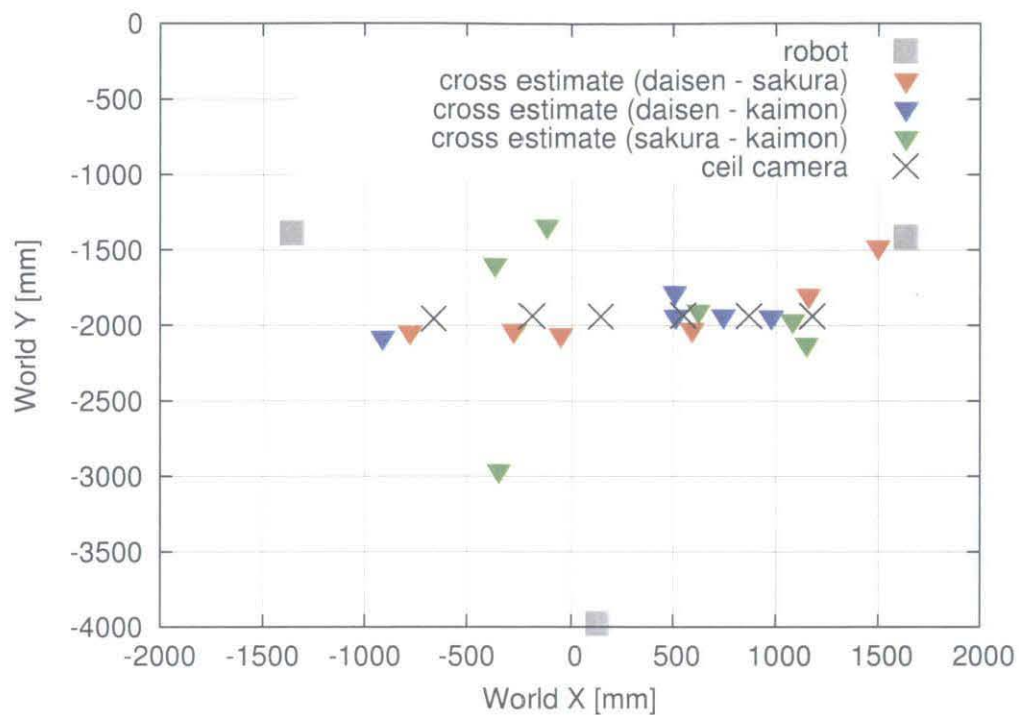


図 43 ボールを転がした時のカメラ対による位置推定結果 1(補間有り)

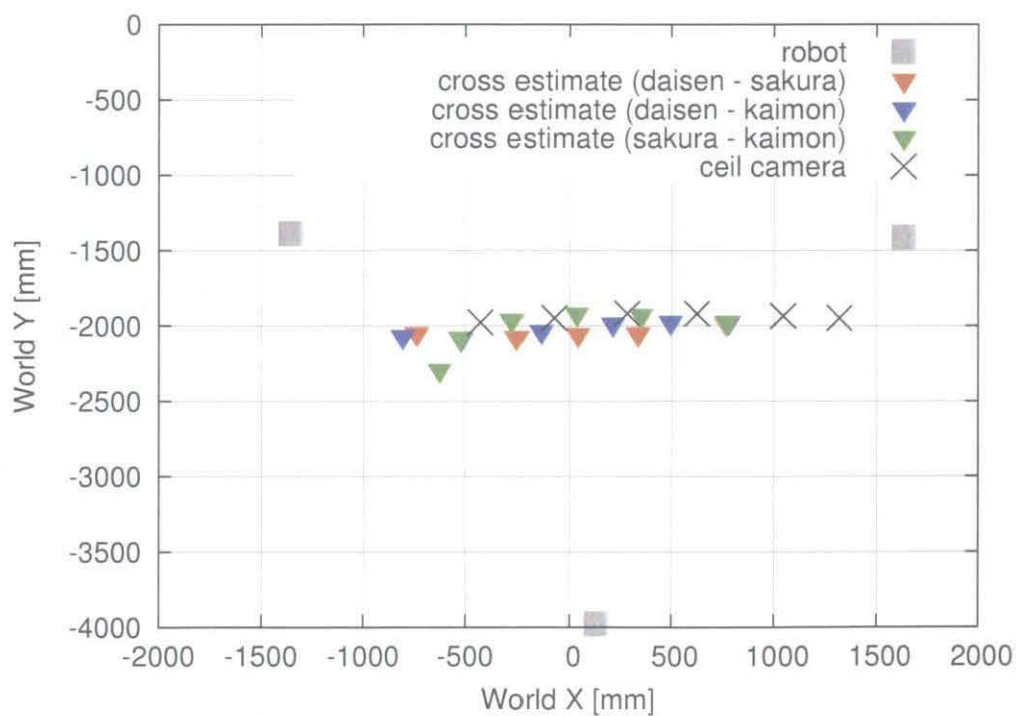


図 44 ボールを転がした時のカメラ対による位置推定結果 1(補間無し)

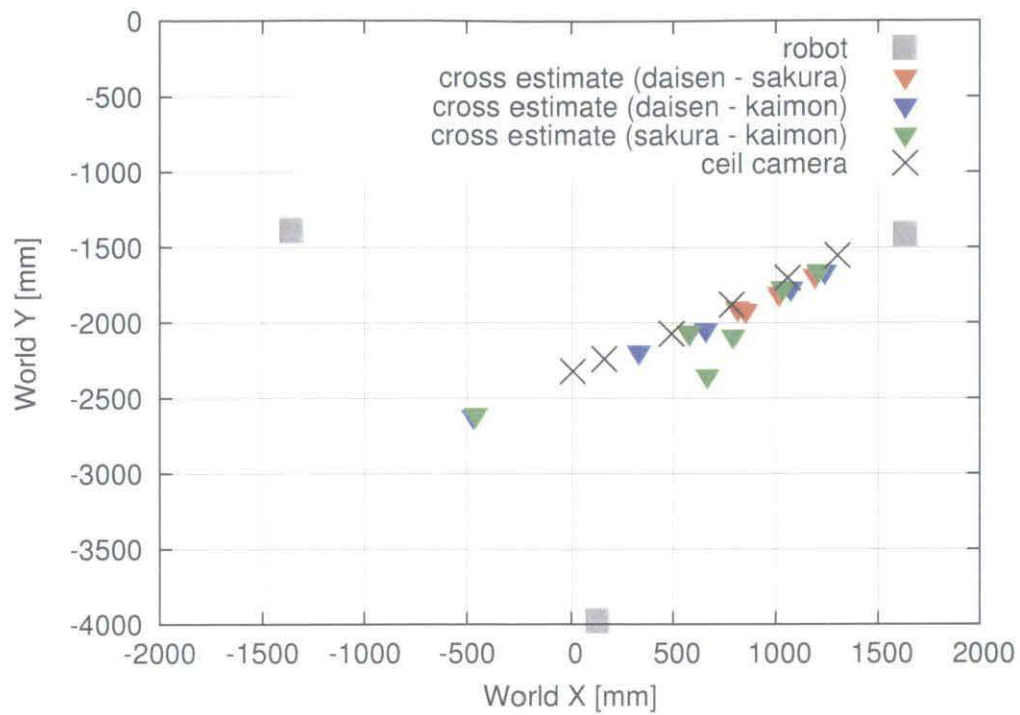


図 45 ボールを転がした時のカメラ対による位置推定結果 2(補間有り)

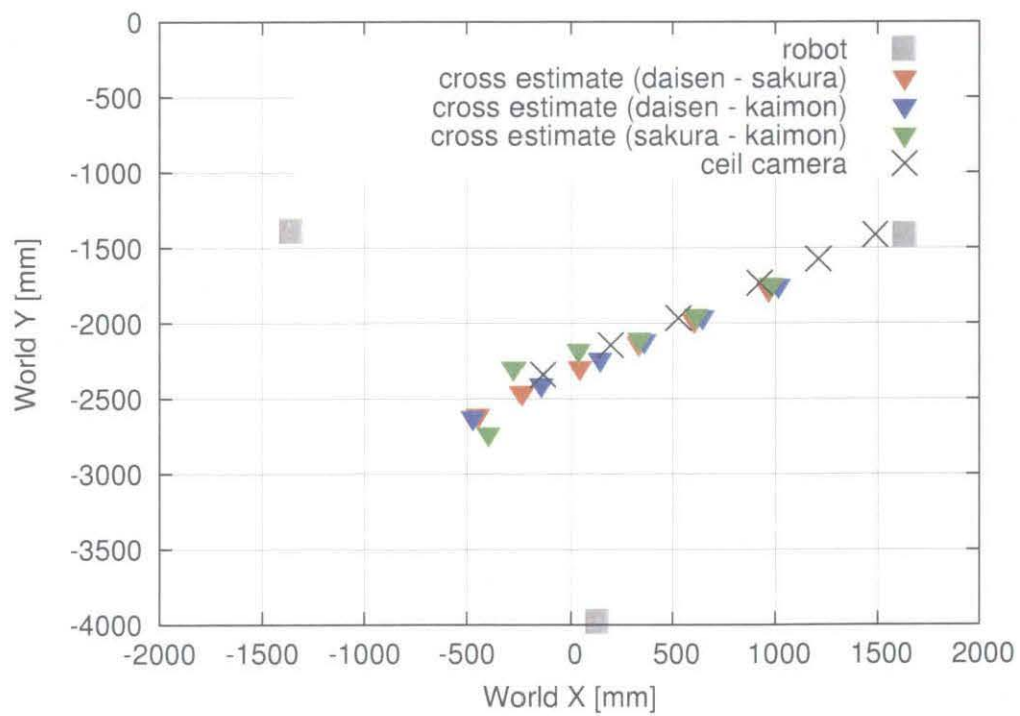


図 46 ボールを転がした時のカメラ対による位置推定結果 2(補間無し)

確な変化量の推定が出来ないことが原因と考えられる。物体が静止している環境では、この影響が少なく、誤差として表れなかったが、物体を動作させた場合には、大きな誤差として影響が出た。

このことから、未来時点における各カメラからの対象物体方向の推定を改善する必要がある。具体的な解決案としては、カルマンフィルタを利用した予測方法を導入することや、以前の統合結果を利用して方向推定を行うこと、消失点や消失線を利用して推定を行うことが考えられ、今後の課題である。

結論

本論文では、非同期の、複数ロボットの単眼カメラの画像情報のみを利用した統合法について、より精度よく統合する方法について述べてきた。第1章では、研究背景と目的について、第2章では、本研究室のロボットのシステムと時刻同期方法について述べた。第3章では、カメラの画像情報からでは対象物体迄の距離が正確に推定出来ないことに注目し、カメラ対を用いたカメラと対象物体を結ぶ直線の交点計算による位置推定方法について実験を行い有効性を確かめた。また、第4章では、カメラの観測誤差を正規分布に近似し表現することで統合を行う今回の手法について説明し、実験を行い有効性を確かめた。

位置推定手法に関しては、誤差平均、標準偏差ともに改善が見られ、これまでよりも正確で安定した手法であることが示せた。統合手法に関しては、重心を用いて統合する手法と比べ、大きく改善出来たとは言えないが、誤差平均、標準偏差ともによりよい結果が得られた。また、提案手法による統合では、対象物体が画像の辺縁部に存在する場合に、より正確な統合が行われることが分かった。これより、静止環境においてのみ言えば、画像中に散在する複数の対象物体の情報を同時に正確に取得する必要があるシステムにおいては、本提案手法は有効な方法であると考えられる。

ただし、カメラ間のなす角度が小さい場合には、本提案手法による位置推定の精度が大きく低下してしまい、同時に統合の精度も大きく低下してしまう為、別の位置推定手法による対処を考える必要がある。また、今回、Zhang の手法によりキャリブレーションを行ったが、レンズの歪曲収差を補正しきれず位置推定の精度が低下することも見られたので、より正確なキャリブレーションを行うといったことが今後の課題である。

また、対象物体が画像中を動作する環境における実験的評価も行った。結果、対象物体が動作する環境においては、本提案手法による改善は示すことが出来なかった。しかし、静止環境では正確な位置推定及び統合が出来ているので、これらとは別の問題であると考えられる。具体的には、本提案手法において、画像取得時刻から統合がなされるまでに通信による遅延などで時間が掛かる為、対象物体の未来位置の線形予測を行い、統合しているが、この予測方法が問題と考えられる。解決策としては、この予測にカルマンフィルタを導入することや、過去の統合結果を利用すること等が考えられ、重要な今後の課題である。

謝辞

日ごろから多くの御指導を頂きました太田義勝教授，鈴木秀智准教授に深く感謝いたします。そして，日頃何かとお世話になりました落合美子事務員に感謝いたします。また，本論文作成にあたって特にお世話になりました鈴木秀智准教授に深く感謝いたします。最後に，日頃から熱心に討論して頂いた研究室の諸氏に感謝いたします。

参考文献

- [1] 古川 智久 “ロボカップサッカーにおける複数エージェントでの時刻同期とマルチセンサフュージョンの検討”, 卒業論文, 三重大大学, 2007.
- [2] 黒田 耕平, “非同期通信と処理を考慮した複数視覚情報の統合に関する研究”, 修士論文, 三重大大学, 2006. 02.
- [3] 中野 博史, “複数台自立移動ロボットによる環境マップ作成に関する研究”, 修士論文, 三重大大学, 2002. 02.
- [4] 能勢 隆, 暮石 芳治, 梅崎 太造, “単眼カメラによる後方斜体との距離計測”, 2001 年度 電気関係学会東海支部連合大会, p.360.
- [5] R.C.Luo and M.H.Lin, “Hierarchical robot multi-sensor data fusion system”, Highly Redundant Sensing in Robotics Systems, Vol. F58, pp. 67-86, 1990.
- [6] A.Ferrein, L.Hermanns, and G.Lakemeyer, “Comparing Sensor fusion Techniques for Ball Position Estimation”, In Proc. RoboCup 2005 Symposium, 2005.
- [7] A.Stroupe, M.Martin, and T.Balch, “Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems”, IEEE International Conference on Robotics and Automation, 2001.
- [8] R.Smith, M.Self, P.Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics” Autonomous robot vehicles Pages: 167-193, 1990.
- [9] P.Pinheiro and P.Lima, “Bayesian Sensor Fusion for Cooperative Object Localization and World Modeling”, In Proc. 8th Conference on Intelligent Autonomous Systems, 2004.
- [10] H.F.Durrant-Whyte, “Sensor Models and Multisensor Integration”, International Journal of Robotic Research, Volume 7, Number 6, 1988.
- [11] Shu-Li Sun, et al, “Multi-sensor optimal fusion fixed-interval Kalman smoothers”, Elsevier Science Publishers B. V., vol. 9, issue.2, pp.293-299, 2008
- [12] D.P.Spanos, R.Olfati-Saber, R.M.Murray, “Distributed Sensor Fusion Using Dynamic Consensus”, IFAC World Congress, 2005.
- [13] J.Elson, L.Girod and D.Estrin, “Fine-grained network time synchronization using reference broadcast”, In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, 2002.
- [14] Z.Zhang, “A Flexible New Technique for Camera Calibration”, Technical Report MSR-TR-98-71
- [15] R.Y.Tsai, “A Versatile Camera Calibration Techniaue for High-Accuracy 3D Machine

- Vision Metrology Using Off-the-shelf TV Cameras and Lenses”, IEEE JOURNAL OF ROBOTICS AND AUTOMATION, VOL. RA-3, NO.4, AUGUST, 1987.
- [16] M.Kam, Z.Zhu, P.Kalata, “Sensor Fusion for Mobile Robot Navigation”, PROCEEDINGS OF THE IEEE, VOL. 85, NO. 1, JANUARY, 1997.
 - [17] 西川 貴之, “CMVision を使ったボールまでの距離の認識”, 卒業論文, 三重大大学, 2005.
 - [18] 中島 圭亮, “ロボカップサッカーにおける対象物体の認識方法の開発と実装”, 卒業論文, 三重大大学, 2007.
 - [19] 福田 剛久, “色情報を利用したハフ変換によるボールの位置推定”, 卒業論文, 三重大大学, 2008.
 - [20] PVM(Parallel Virtual Machine),
<http://www.csm.ornl.gov/pvm/>
 - [21] OpenCV,
<http://opencv.jp/>
 - [22] CMVision,
<http://www.csm.ornl.gov/pvm/>

付録 A

カメラキャリブレーション

本研究で使用したカメラのキャリブレーションについて述べる。キャリブレーションとは、カメラの焦点距離や画像中心といったカメラが各々固有に持つパラメータ (内部パラメータ) と、位置や姿勢といったパラメータ (外部パラメータ) を求めることである。

本研究室ではこれまで、射影行列を用いたキャリブレーションが行われていたが、射影行列によるキャリブレーションでは、カメラの歪曲収差を考慮することが出来ない上、本研究室のカメラはワイドコンバージョンレンズを使用していることもあり歪曲収差の影響が大きく、誤差が大きかった。そこで、カメラの歪曲収差も考慮したキャリブレーション手法でキャリブレーションを行う必要があった。歪曲収差を考慮したキャリブレーション手法には、Tsai の方法 [15] や、Zhang の手法 [14] があるが、本研究では、OpenCV[21] に実装され、画像を用意すれば、自動でキャリブレーションを行うことが出来る Zhang の手法 [14] を用いてキャリブレーションを行った。

A.1 キャリブレーション方法と結果

本研究で行ったキャリブレーションの方法について述べる。まず、白と黒のチェッカーパターンが印刷されたボードを用意し、各々姿勢を変えて 10 枚画像を撮影後、OpenCV[21] のライブラリを使用してキャリブレーションを行った。キャリブレーションに使用したパターン画像の 1 例を図 47 に、キャリブレーション結果の 1 例を表 6 から 8 に示す。

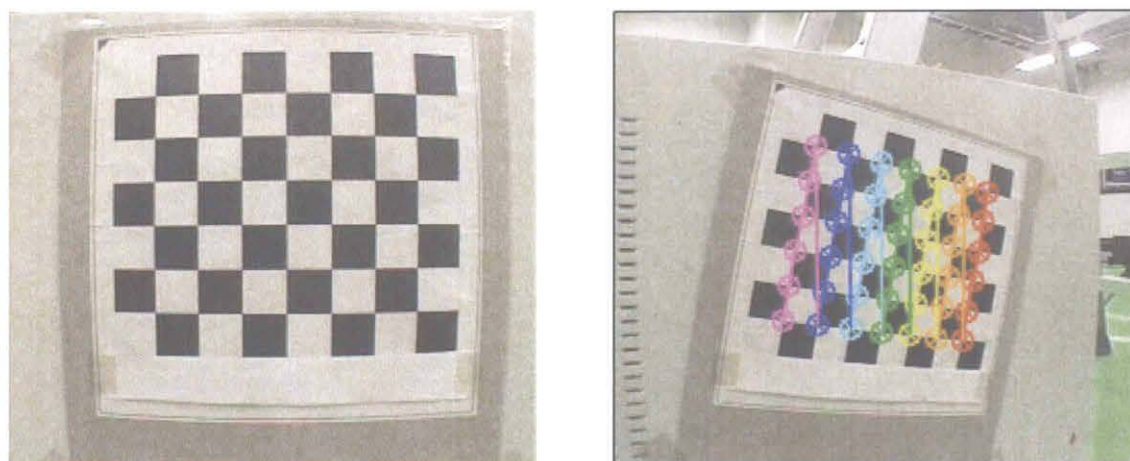


図 47 キャリブレーションパターン画像

また、この時にキャリブレーション用ボードの世界座標系での設置位置を測定器 (ライカ スマートセオビルダー R200S) で測定することにより、キャリブレーション時に世界座標系での回転行列と平行移動ベクトルを求めることが出来る。

表 6 キャリブレーション結果 (内部パラメータ), daisen

cx	127.879
cy	106.708
fx	234.006
fy	233.044
k1	-0.275011
k2	0.222687
p1	0.00246319
p2	0.000432284

表 7 キャリブレーション結果 (内部パラメータ), sakura

cx	134.996
cy	107.801
fx	243.192
fy	242.156
k1	-0.281855
k2	0.206291
p1	0.00310279
p2	-0.00226311

表 8 キャリブレーション結果 (内部パラメータ), kaimon

cx	130.163
cy	103.191
fx	232.522
fy	231.541
k1	-0.278541
k2	0.277961
p1	0.00110409
p2	-0.000620935