

博士学位論文

シリコン網膜を用いた
知的センシングに関する研究

平成24年3月

九州工業大学大学院生命体工学研究科

眞田 篤

目次

| | |
|---------------------------------|----|
| 第1章 序論..... | 2 |
| 1.1 研究背景..... | 2 |
| 1.2 本論文の目的..... | 5 |
| 1.3 本論文の内容・構成..... | 5 |
| 第2章 視覚システムの開発..... | 11 |
| 2.1 はじめに..... | 11 |
| 2.2 開発する視覚システムのハードウェア..... | 11 |
| 2.2.1 演算装置..... | 11 |
| 2.2.2 カメラ装置..... | 12 |
| 2.3 アルゴリズムの検討..... | 12 |
| 2.3.1 ブロックマッチング法..... | 13 |
| 2.3.2 勾配法..... | 13 |
| 2.3.3 ブロックマッチング法と勾配法との比較..... | 16 |
| 2.3.4 テンプレート領域の検討..... | 18 |
| 2.3.5 探索範囲の決定..... | 19 |
| 2.3.6 演算時間の見積もり..... | 19 |
| 2.3.7 処理の高速化..... | 20 |
| 2.4 FPGAへの実装..... | 21 |
| 2.4.1 システム構成（USB接続）..... | 21 |
| 2.4.2 動作タイミング（USB接続）..... | 23 |
| 2.4.3 システム構成（UART接続）..... | 23 |
| 2.4.4 動作タイミング（UART接続）..... | 23 |
| 2.5 おわりに..... | 24 |
| 第3章 視覚システムを用いたエゴモーシヨン推定..... | 46 |
| 3.1 はじめに..... | 46 |
| 3.2 エゴモーシヨン推定方法..... | 46 |
| 3.3 実験..... | 50 |
| 3.3.1 レールを用いた実験..... | 50 |
| 3.3.2 実機に実装しての実験..... | 51 |
| 3.4 まとめ..... | 52 |
| 第4章 視覚システムを用いたドミナントプレーンの検出..... | 66 |
| 4.1 はじめに..... | 66 |
| 4.2 ドミナントプレーン検出方法..... | 66 |

| | | |
|-----|--------------------|----|
| 4.3 | ドミナントプレーン検出実験..... | 69 |
| 4.4 | まとめ..... | 70 |
| 第5章 | 結論..... | 85 |
| 5.1 | 考察と結論..... | 85 |
| 5.2 | 今後の展望..... | 87 |

謝辞

参考文献

図一覽

| | | |
|--------|--|----|
| Fig.1 | Image of the future robot society..... | 6 |
| Fig.2 | Components for autonomous robot..... | 6 |
| Fig.3 | Technology required for autonomous behaviors..... | 7 |
| Fig.4 | The comparison of sensory organs and robot's devices..... | 7 |
| Fig.5 | Environmental investigation..... | 8 |
| Fig.6 | Sight of the mobile robot..... | 8 |
| Fig.7 | Small omni-directional mobile robot "WITH"..... | 9 |
| Fig.8 | Optical flow..... | 9 |
| Fig.9 | Operation speed - Power consumption of Processor..... | 25 |
| Fig.10 | FPGA board (Atmark Techno "SUZAKU V")..... | 25 |
| Fig.11 | Illuminance in the living environment..... | 26 |
| Fig.12 | Constancy of illuminance[46]..... | 26 |
| Fig.13 | Overview of silicon retina camera..... | 27 |
| Fig.14 | Vision chip..... | 27 |
| Fig.15 | Vision system..... | 28 |
| Fig.16 | Template and Search area..... | 29 |
| Fig.17 | Original Image..... | 29 |
| Fig.18 | 1st frame and 2nd frame..... | 30 |
| Fig.19 | Image shift direction..... | 30 |
| Fig.20 | Result of operation (Block matching method, 1pixel shift)..... | 31 |
| Fig.21 | Result of operation (Block matching method, 5pixel shift)..... | 32 |
| Fig.22 | Result of operation (Horn & Schunck method, 1pixel shift)..... | 33 |
| Fig.23 | Result of operation (Horn & Schunck method, 5pixel shift)..... | 34 |
| Fig.24 | Result of operation (Lucas & Kanade method, 1pixel shift)..... | 35 |
| Fig.25 | Result of operation (Lucas & Kanade method, 5pixel shift)..... | 36 |
| Fig.26 | Result of 1pixel shift (x direction)..... | 37 |
| Fig.27 | Result of 1 pixel shift (y direction)..... | 37 |
| Fig.28 | Result of 5pixel shift (x direction)..... | 38 |
| Fig.29 | Result of 5pixel shift (y direction)..... | 38 |
| Fig.30 | Operating time..... | 39 |
| Fig.31 | Template..... | 39 |
| Fig.32 | Search area..... | 40 |
| Fig.33 | Parallel processing..... | 40 |

| | | |
|--------|---|----|
| Fig.34 | Structure of DSP slice..... | 41 |
| Fig.35 | Input and output data of DSP slice..... | 42 |
| Fig.36 | System configuration (USB IF version)..... | 42 |
| Fig.37 | Operation timing..... | 43 |
| Fig.38 | Improved operation timing..... | 43 |
| Fig.39 | System configuration (UART version)..... | 44 |
| Fig.40 | Operation timing of UART version..... | 44 |
| Fig.41 | Camera image and ground area..... | 53 |
| Fig.42 | Movement of slicer and corresponding points..... | 53 |
| Fig.43 | The cross relationship of the coordinate..... | 54 |
| Fig.44 | Experimental device..... | 55 |
| Fig.45 | Slider..... | 55 |
| Fig.46 | Experimental device..... | 56 |
| Fig.47 | Reference image..... | 56 |
| Fig.48 | Experimental result of change slicer speed($\beta = 0$)..... | 57 |
| Fig.49 | Experimental result of change horizontal angle..... | 57 |
| Fig.50 | Experimental result of change slicer speed($\beta = 30$)..... | 58 |
| Fig.51 | Experimental result of change horizontal angle..... | 58 |
| Fig.52 | Experimental result of change slicer speed($\beta = 45$)..... | 59 |
| Fig.53 | Experimental result of change horizontal angle..... | 59 |
| Fig.54 | Vision system on the WITH..... | 61 |
| Fig.55 | Geometry of the camera installation..... | 62 |
| Fig.56 | Movement of the robot..... | 62 |
| Fig.57 | Trace of the robot..... | 63 |
| Fig.58 | Estimated path and ground truth..... | 63 |
| Fig.59 | Transition of the xy coordinate..... | 64 |
| Fig.60 | Transition of the rotatino angle..... | 64 |
| Fig.61 | Dominant plane..... | 71 |
| Fig.62 | Enkelamann's Dominant plane detection method[72]..... | 71 |
| Fig.63 | Ohnishi's Dominant plane detection method[74]..... | 72 |
| Fig.64 | Ground's Flow and Obstacle's Flow..... | 72 |
| Fig.65 | Detected flow..... | 73 |
| Fig.66 | Projected flow..... | 73 |
| Fig.67 | Movement increase ratio..... | 74 |

| | | |
|--------|---|----|
| Fig.68 | Experiment environment (wall)..... | 74 |
| Fig.69 | Experiment environment (box)..... | 75 |
| Fig.70 | Experiment environment (box offset)..... | 75 |
| Fig.71 | Camera position (distance = 0mm)..... | 76 |
| Fig.72 | Movement increase ratio (distance = 0mm)..... | 76 |
| Fig.73 | Camera position (distance = 500mm)..... | 77 |
| Fig.74 | Camera image(distance = 500mm)..... | 77 |
| Fig.75 | Estimated path (distance = 500mm)..... | 77 |
| Fig.76 | Movement increase ratio (distance = 500mm)..... | 78 |
| Fig.77 | Camera position (distance = 640mm)..... | 79 |
| Fig.78 | Camera image (distance = 640mm)..... | 79 |
| Fig.79 | Estimated path (distance = 640mm)..... | 79 |
| Fig.80 | Movement increase ratio (distance = 640mm)..... | 80 |
| Fig.81 | Camera position (distance = 700mm)..... | 81 |
| Fig.82 | Camera image (distance = 700mm)..... | 81 |
| Fig.83 | Estimated path (distance = 700mm)..... | 81 |
| Fig.84 | Movement increase ratio (distance = 700mm)..... | 82 |
| Fig.85 | near area..... | 83 |
| Fig.86 | Vision system..... | 87 |

表一覽

| | | |
|----------|--|----|
| Table 1 | Specification of FPGA board..... | 26 |
| Table 2 | Result of operation (Block matching method, 1pixel shift)..... | 31 |
| Table 3 | Result of operation (Block matching method, 5pixel shift)..... | 32 |
| Table 4 | Result of operation (Horn & Schunck method, 1pixel shift)..... | 33 |
| Table 5 | Result of operation (Horn & Schunck method, 5pixel shift)..... | 34 |
| Table 6 | Result of operation (Lucas & Kanade method, 1pixel shift)..... | 35 |
| Table 7 | Result of operation (Lucas & Kanade method, 5pixel shift)..... | 36 |
| Table 8 | Travel distance obtained in the moving speed of the camera ($\beta = 0$). .. | 60 |
| Table 9 | Travel distance obtained in the moving speed of the camera ($\beta = 30$) .. | 60 |
| Table 10 | Travel distance obtained in the moving speed of the camera ($\beta = 45$) .. | 60 |
| Table 11 | Error for each installation angle of the camera..... | 61 |
| Table 12 | Movement increase ratio (distance = 500mm)..... | 78 |
| Table 13 | Movement increase ratio (distance = 640mm)..... | 80 |
| Table 14 | Movement increase ratio (distance = 700mm)..... | 82 |
| Table 15 | Detection rate and accuracy rate (all flow)..... | 83 |
| Table 16 | Detection rate and accuracy rate (near flow)..... | 83 |

第 1 章

序論

第1章 序論

1.1 研究背景

現在，ロボットは様々な分野での活躍が期待され，研究・開発が進められている．ロボットの活躍分野も拡大しており，主要な分野である第2次産業はもちろん，第1次産業や第3次産業にも及んでおり，ロボットはいっそう身近なものとなっている．Fig.1に示すのは経済産業省から2008年に発表されたロボット分野に関するアカデミックロードマップ[1]で示されている．人を中心とし，ロボットが人々の生活の場で活躍する未来が描かれている．例えば，農業や林業においての労働力としてのロボット，宇宙などの極限状況で働くロボット，ロボットスーツなど人と一体となるロボット，生活支援・介護など人を直接支援するロボットなどが挙げられている．ロボットが人と共に生活空間で活躍するためには，種々な分野での革新が求められるが，特に重要な課題の1つが自律化である．

ここで，ロボットの自律化にはFig.2に示ように，

- ・ センシング
- ・ コミュニケーション
- ・ マニピュレーション
- ・ 移動

などの要素が複合されて実現される．例えば，自律的に行動するために必要な技術は．Fig.3に示すように，

- ・ 環境認識と把握
- ・ 自己位置認識
- ・ 障害物の認識
- ・ 衝突の回避
- ・ 不整地移動

などが挙げられる．これらのうち，環境認識と把握，自己位置認識，障害物の認識はセンシング技術が大きく関係する．自律的に行動するためには周囲の状況を知ることが重要であるため，本研究ではセンシング技術に着目した．

ここで，ロボットに用いられるセンサには様々な種類のものが開発されており．これらを人の感覚器と対比を行うとFig.4のようになり，人の感覚に対応する様々なセンサがロボットに用いられている．この中でも，人間が得られる外部情報の約80%が視覚からの情報との説もあり[2]，人の感覚器の中でも特に視覚から得られる情報は多く，視覚からの

情報は非常に重要である。人の視覚に相当するカメラシステムからは静止画及び動画が得られ、2次元の静止画の解析から得られる情報は、

- ・色
- ・大きさ
- ・二次元形状
- ・模様
- ・角度

などが挙げられる。また、2つのカメラにおける視差を用いることで、奥行き情報も得られる。動画は一連の連続した静止画の時系列情報を含んでおり画像の時間微分から、

- ・速度
- ・加速度
- ・角速度
- ・角加速度
- ・奥行き

等が得られる。

視覚から得られる情報は時空間情報であり、自律的に行動するために必要な情報の大部分を抽出することが可能である。

自己位置認識に関しては、井上らは単眼カメラを用いて SHIFT 特徴量を計算し、移動前と移動後の特徴点を用いて回転行列と並進移動ベクトルを求めることで、自己の運動を推定し、自己位置推定する手法を提案している [3]。また、岩田らはステレオカメラを用いて3つの高さ領域のマップを作成し、作成したマップとロボットが得た画像情報を比較することにより自己位置を同定する手法を提案している [4]。その他、自身の動きをであるエゴモーション検出に関しても、様々な研究が鋭意進められている [5-13]。

環境認識に関し、山崎らはレーザー測域センサを用いて机などの足を検出し、足の大きさに合わせた家具のモデルをカメラ画像に投影し、モデルとカメラ画像のエッジとの相関から机や椅子などの家具を認識する手法を提案している [14]。

障害物の認識に関しては、久徳らは車載カメラを対象とし、の過去の走行データと、現在の走行データを比較し、それらの差分から障害物を検出している [15]。また、連続画像の差分から注視点の移動ベクトルであるオプティカルフローを算出し、オプティカルフローから自己運動を回転成分と並進成分に分解し、並進成分から障害物予測を行う方法

[16], ロボットの走行可能領域(地面)を平面と近似して, 物体の存在領域と走行可能領域であるドミナントプレーンを検出する方法[16-25]などが挙げられる.

このようにカメラ映像を用いることで, ロボットの自律化に必要な情報が得られる. しかし, カメラの映像には多くの情報が含まれている反面, 画像ノイズや照明条件, 物体が隠れるオクルージョン問題, 対応点判別, 特徴量抽出等の複雑な情報の組み合わせから必要な情報を取捨選択する必要がある. ロボットにおけるリアルタイム処理を想定した場合, 多くのマシンパワーが必要で, 高性能なプロセッサまたは専用のシステムが必要となる. 高性能なプロセッサは多くの場合, 多くの電力を必要とし, 物理的にもシステムが大きくなるため, 視覚システムの高性能化, スタンドアローン化には多くの課題が存在する.

本論文では, これらの問題を考慮し, ロボットへの搭載を前提とした知的センシングを行うための視覚システムを提案する. 知的センシングが可能な視覚システムでは次に挙げる特徴の実現を目標とする.

- ・ スタンドアローン動作可能な自己完結したシステム
- ・ 照明条件の変化への適応が可能
- ・ 移動体の検出・認識
- ・ 低消費電力で長期間にわたる計測

これらの特徴を実現できれば, 例えば Fig.5 に示すように電源とした視覚システムを, 自然環境に設置することで, 生態系の長期観測を行う事ができる. また, Fig.6 に示すように移動ロボットの視覚システムとして搭載することで障害物の回避や走行可能な路面を検出を行えるため, リアルタイムでの行動決定が可能となる. 本論文では石井研究室で開発された全方位移動ロボット WITH (Fig.7) [26]を用いる.

ここで, 視覚システムの動き検出には, オプティカルフローを用いる. オプティカルフローとは物体の動きベクトルのことで, Fig.8 に示すように, 2つの連続画像の比較から得られる物体の移動量・方向である. オプティカルフローは全方位移動ロボット[27, 28]や, ヘリコプター[29]などのナビゲーションや, 動体の検出[30]やトラッキング[31]に用いられ, 非接触での動き検出について報告されている. また, モザイク画像の作成にも導入成果が報告されている[32]. オプティカルフローはロボットや物体の動きを検出するだけでなく, 人の表情を読み取る研究においても使用されており[33, 34], 非常に幅広い分野で応用可能な情報である.

1.2 本論文の目的

知的センシングを行うための視覚システムを開発を行う。開発した視覚システムを用いて移動ロボットの自己運動の推定，走行可能な路面の検出及び障害物の検出を行い，開発した視覚システムの有用性を示す。

1.3 本論文の内容・構成

本論文は全5章で構成してある。

第1章では，研究背景としてロボットの自律化に必要な機能，特に，ナビゲーションに関して述べる。動きの検出システムの有用性，応用範囲の多様さについて述べ，オプティカルフロー演算システムの必要性を示す。

第2章では，視覚システムの開発について述べる。オプティカルフロー演算システムを構成する，カメラ，演算器，及びオプティカルフローの演算方法を検討し，開発を行う。

第3章では，開発した視覚システムを用いて，ロボットの自身の運動であるエゴモーション推定について述べる。

第4章では，開発した視覚演算システムを用いた障害物検出およびドミナントプレーン検出について述べる。

第5章では，結論を示し，考察を行う。



Fig.1 Image of the future robot society

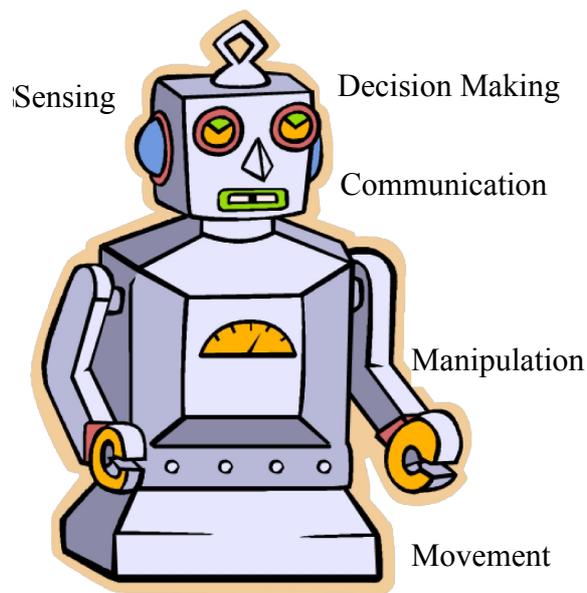


Fig.2 Components for autonomous robot

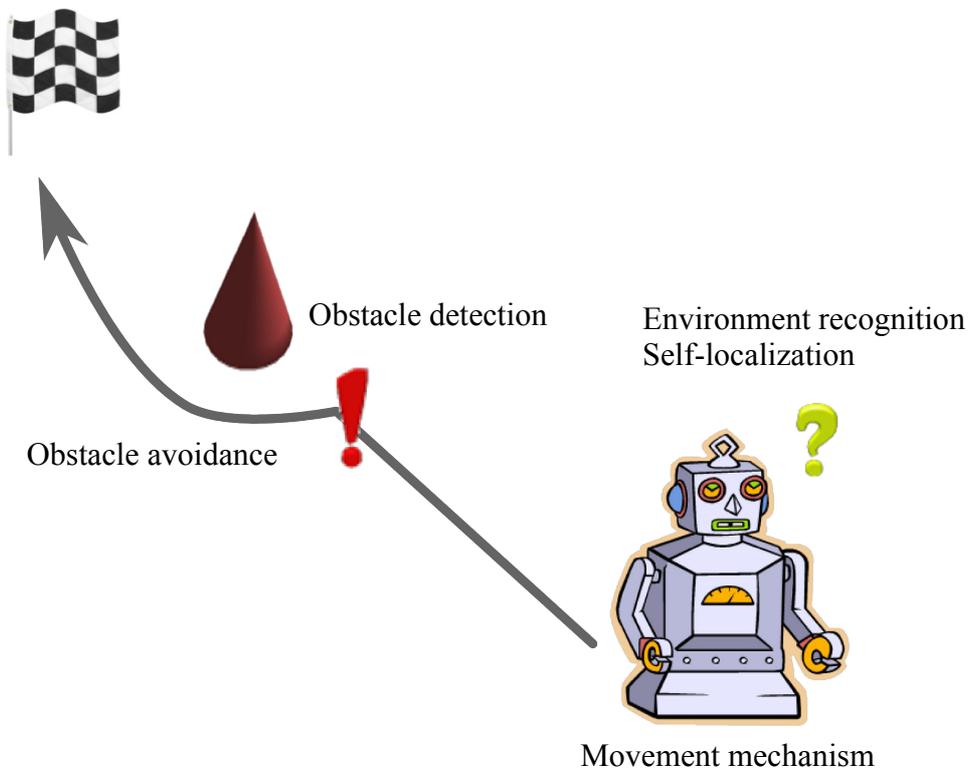


Fig.3 Technology required for autonomous behaviors

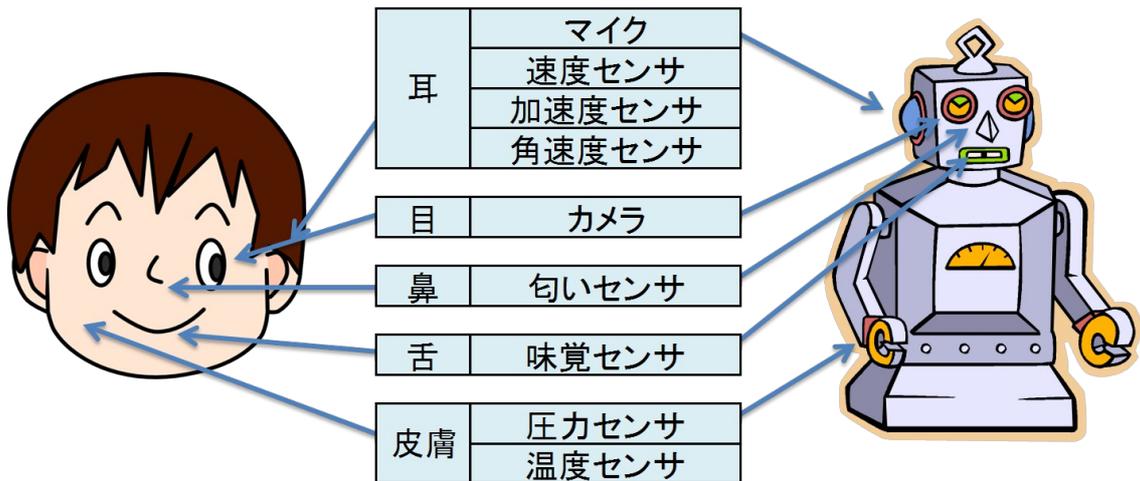


Fig.4 The comparison of sensory organs and robot's devices

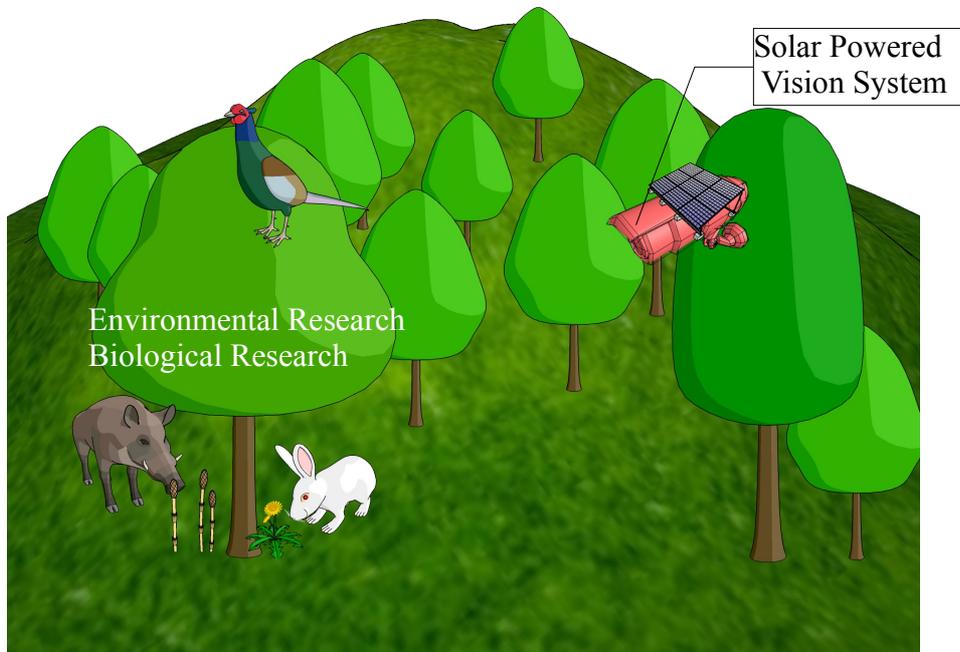


Fig.5 Concept of long-term ecological observation

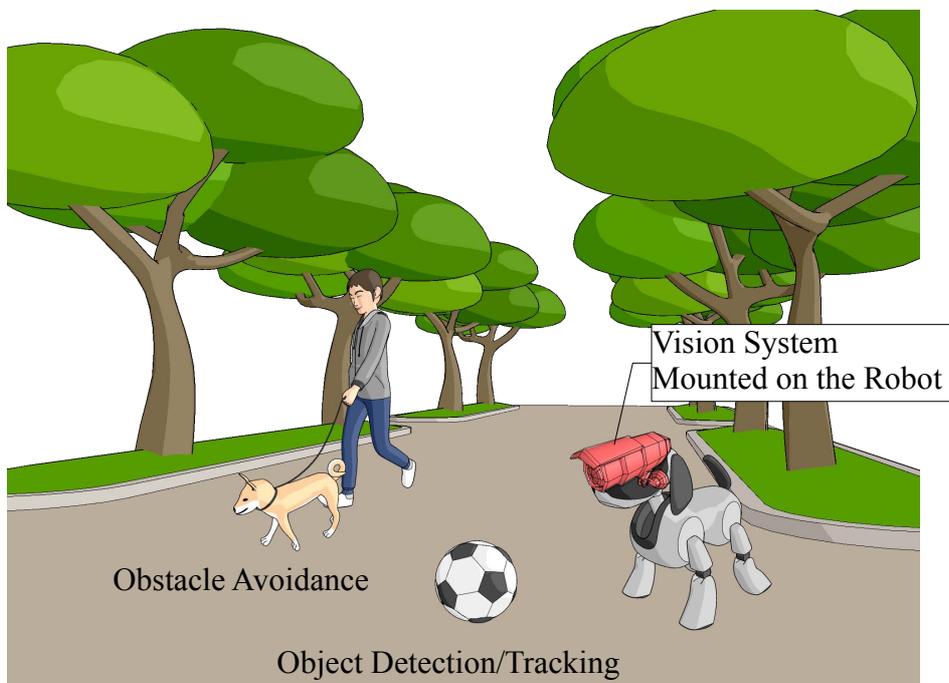


Fig.6 Sight of the mobile robot

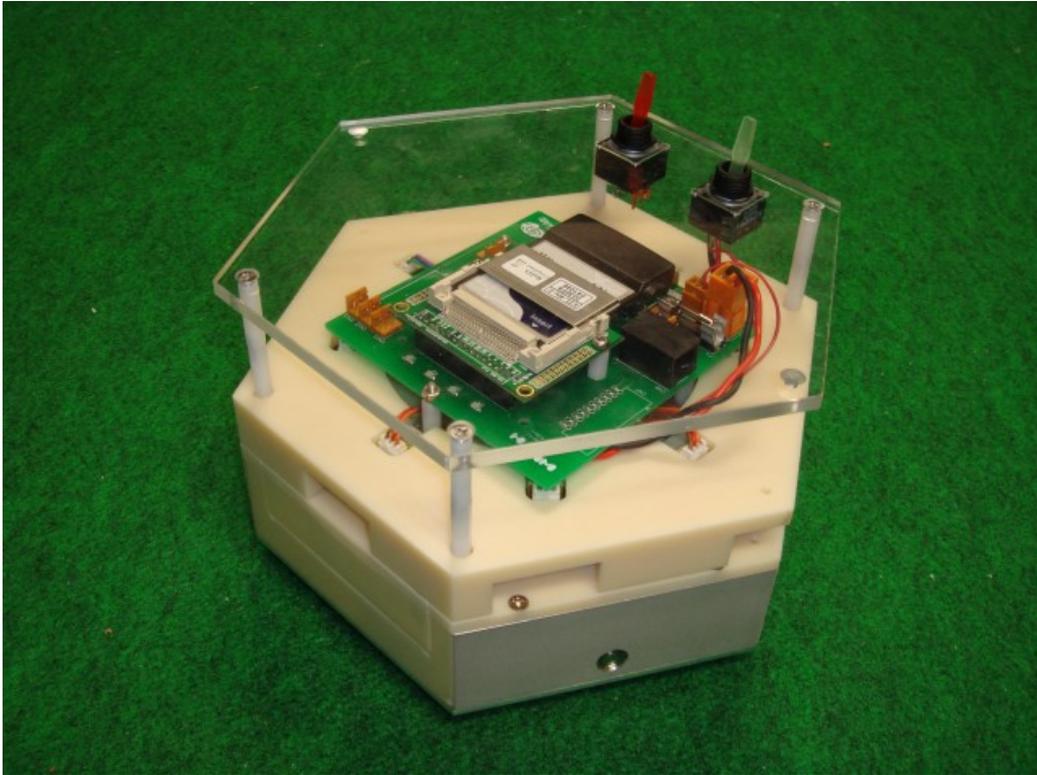


Fig.7 Small omni-directional mobile robot "WITH"

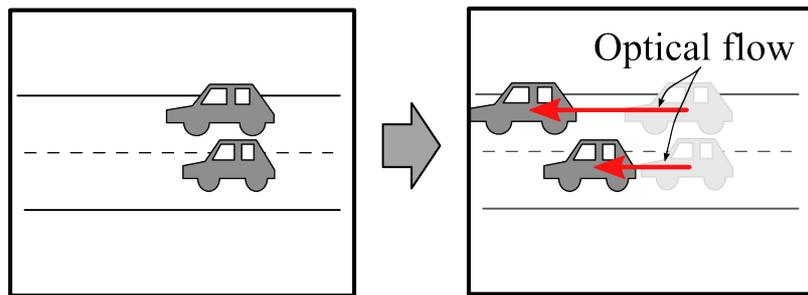


Fig.8 Optical flow

第2章

視覚システムの開発

第2章 視覚システムの開発

2.1 はじめに

第1章において述べたようにロボットの自律化にはセンシング技術が重要であり、様々な種類のセンサが存在するが、視覚からは様々な情報が得られる。本章では、動きの情報であるオプティカルフローを検出する視覚システムの開発について述べる。

まず、オプティカルフローを検出するための主要素として(1)ハードウェア：演算装置とカメラ、(2)ソフトウェア：オプティカルフロー演算アルゴリズム、の検討を行う。次に、決定したアルゴリズムを演算装置に実装し、(3)オプティカルフロー演算の評価実験を行う。

2.2 開発する視覚システムのハードウェア

2.2.1 演算装置

リアルタイムでのオプティカルフロー演算には、大きなマシンパワーが必要であり、これまで多くの研究では高性能計算器や専用の画像処理ボードが採用されている。計算器はスーパーコンピュータに代表されるように物理的にも大きな空間を占有し、消費電力が大きく、専用の画像処理ボードはプロセッサの発展に伴い陳腐化する。これらは、本論文の目標である視覚システムの小型化や小型ロボットへの搭載、知能化には適していない。

演算装置単体の消費電力と演算速度の分布を Fig.9 に示す[35-39]。Intel などの汎用 CPU や、マルチコアプロセッサ Cell 及び GPU は演算能力が高いが、一方で消費電力も大きい。マイコンは消費電力が小さいが演算能力も小さい。本研究では、並列演算による高処理化、単体動作が可能で比較的消費電力の小さい FPGA を用いる。

本システムで使用する FPGA ボードを Fig.10 に、仕様を表1に示す。このボードに搭載されている FPGA は、Xilinx 社の Virtex-4 FPGA(XC4VFX12-SF363)である。この FPGA には PowerPC コアも搭載されており、ロジック回路による並列演算と、CPU による演算を組み合わせることができる。

2.2.2 カメラ装置

我々は様々な照明環境の中で生活しており、どのような環境下でも物体を認識できるように、ロボットも同様の環境での活躍が望まれる。生活空間での照度は、Fig. 11に示すように照度は8桁以上変化する。本研究のカメラは幅広い照明環境の変化にも対応可能なカメラが望ましい。

人間の目は、先述のような幅広く照度に変化する環境でも物体を認識することができおり、非常に優秀な視覚センサである。この生体の網膜の構造と機能を模してアナログCMOS集積回路として実現したものをシリコン網膜と呼び、シリコン網膜を用いた視覚システムの研究・開発が行われている[40-43]。八木、下ノ村らの研究[46-48]によると、八木らが開発したシリコン網膜カメラ[49-51]はフォトダイオードに対するリセット信号を制御することにより、照明環境が変化した場合においても明るさの恒常性が得られ、異なる照度環境下でも同様の映像が得られるとしている(Fig. 12)。

本研究ではオプティカルフローの演算において、照明環境の変化に対応できるようにするため、八木らのシリコン網膜カメラを用いる。Fig. 13にカメラの外観、Fig. 14にシリコン網膜ビジョンチップを示す。

これらのシリコン網膜カメラとFPGAを用いてFig. 15に示すように色、動き、大きさなどの検出システムを目指す。まずは動き検出システムの開発を行う。動きを検出することで、障害物、奥行き、形状などを検出することができる。最終的にはこれらを統合し、物体や状況の認識・判断を行い、対象物の制御まで行えるシステムを目標とする。

2.3 アルゴリズムの検討

代表的なオプティカルフローの演算手法としてブロックマッチング法[52-55]と勾配法[56-63]が挙げられ、これらの演算を用いた研究が盛んに行われている[64-67]。

ブロックマッチング法は、輝度値の急激に変化すると画像においてもオプティカルフローの誤りが小さく、画像中のノイズにも強く安定して演算することができるが、全探索を行うので計算時間が大きくなるという欠点がある。勾配法は対応領域を探索しないので高速だが、輝度値が急激に変化するところではオプティカルフローの誤差が大きく、画像中のノイズに弱いという欠点がある[68, 69]。

ここでは、これらのオプティカルフローアルゴリズムをパソコン上で実際に実行し、FPGAへの実装に適したアルゴリズムについて比較検討を行う。

2.3.1 ブロックマッチング法

ブロックマッチング法は、画像中のある大きさの領域をテンプレートとし、微小時間後の画像を探索領域とし、テンプレートとの対応領域を探索領域中から検出する方法である。

Fig.16 に示すように、テンプレート上の座標 (i, j) における輝度を ${}^T I(i, j)$ とし、探索領域における座標 $(x+i, y+j)$ における輝度を ${}^S I(x+i, y+j)$ とする。ここで、 i, j は画像の左上を座標 $(0, 0)$ とするテンプレート上の画素座標、 x, y をそれぞれ横方向、下方向へのテンプレートのシフト量とする。このとき、テンプレートと探索領域内の対応領域との評価には式(1)に示す差の絶対値の和である SAD(Sum of Absolute Differences)、式(2)に示す差の二乗和である SSD(Sum of Squared Differences)などが使用される。ブロックマッチング法は、この評価値 R_{SAD} 、 R_{SSD} が最小となる x, y を求める方法である。

$$R_{SAD} = \sum_i \sum_j |{}^T I(i, j) - {}^S I(x+i, y+j)| \quad (1)$$

$$R_{SSD} = \sum_i \sum_j \{ {}^T I(i, j) - {}^S I(x+i, y+j) \}^2 \quad (2)$$

2.3.2 勾配法

画像上のある点 (x, y) の時間 t での濃淡値を $I(x, y, t)$ とし、物体が移動したとき微小時間 Δt 後の物体の濃淡値は変化しないと仮定すると、微小時間前後の濃淡値の関係は、

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (3)$$

と表すことができる。

右辺をテイラー展開し、二次以上の項を無視すると

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t \quad (4)$$

となり、ここで $\Delta t \rightarrow 0$ とし、これを整理すると

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (5)$$

と表される。ここで、

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt} \quad (6)$$

とおくと

$$I_x u + I_y v + I_t = 0 \quad (7)$$

と表すことができる、なお I_x, I_y, I_t はそれぞれ

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y} \quad I_t = \frac{\partial I}{\partial t} \quad (8)$$

である。これを勾配法の基本式という。

この勾配法の基本式だけでは、オプティカルフローを一意に決定できないので、拘束条件を付加する必要がある。拘束条件は次に示すような空間的局所最適化法、空間的大域最適化法、時間的局所最適化法、時間的大域最適化法などがある。

・空間的局所最適化法

空間局所領域として $n \times n$ ピクセルの正方形を設定する。局所領域内においてオプティカルフローが一定であると仮定することにより、 n^2 個の観測方程式が立てられる。行列式で表すと、

$$\mathbf{Gf} = -\mathbf{b} \quad (9)$$

となる。ここで、 $\mathbf{G}, \mathbf{f}, \mathbf{b}$ はそれぞれ

$$\mathbf{G} = \begin{bmatrix} I_{1x} & I_{1y} \\ \vdots & \vdots \\ I_{ix} & I_{iy} \\ \vdots & \vdots \\ I_{n^2x} & I_{n^2y} \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} u \\ v \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} I_{1t} \\ \vdots \\ I_{it} \\ \vdots \\ I_{n^2t} \end{bmatrix} \quad (10)$$

である。

この過剰決定系の連立方程式は、最小二乗法によって解くことができる。

$$\hat{\mathbf{f}} = -(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{b} \quad (11)$$

・時間的局所最適法

式(9)において

$$\mathbf{G} = \begin{bmatrix} I_x & I_y \\ I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} I_t \\ I_{tx} \\ I_{ty} \end{bmatrix} \quad (12)$$

とすることにより，空間的局所最適化法と同様に解が得られる。

・空間的大域最適化法

オプティカルフローの空間的滑らかさを条件として付加するために，オプティカルフローの変化率の大きさ

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad (13)$$

を最小にする。

この条件を付加し，式(14)における E の最小化を行う。

$$E = \sum_x \sum_y \left\{ (I_x u + I_y v + I_t)^2 + \alpha^2 \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right) \right\} \rightarrow \min. \quad (14)$$

ここで， α は基本拘束式と滑らかさの相対的な重みを決定する係数である。汎関数 E に変分法を用いることにより，式(15)が得られる。

$$\begin{cases} I_x^2 u + I_x I_y v = \alpha^2 \nabla^2 u - I_x I_t \\ I_x I_y u + I_y^2 v = \alpha^2 \nabla^2 v - I_y I_t \end{cases} \quad (15)$$

式(15)の ∇^2 は式(16)，式(17)のように近似することができる。

$$\begin{aligned} \nabla^2 u(i, j) = & \frac{1}{6} \{ u(i-1, j) + u(i, j+1) + u(i+1, j) + u(i, j-1) \} \\ & + \frac{1}{12} \{ u(i-1, j-1) + u(i-1, j+1) + u(i+1, j+1) + u(i+1, j-1) \} \end{aligned} \quad (16)$$

$$\begin{aligned} \nabla^2 v(i, j) = & \frac{1}{6} \{ v(i-1, j) + v(i, j+1) + v(i+1, j) + v(i, j-1) \} \\ & + \frac{1}{12} \{ v(i-1, j-1) + v(i-1, j+1) + v(i+1, j+1) + v(i+1, j-1) \} \end{aligned} \quad (17)$$

この式(16)，式(17)を解くことで，オプティカルフローを得ることができる。

- ・ 時間的大域最適化法

式(7)に式(18)に示すような時間的滑らかさを条件として付加し，空間的大域最適化法と同様に解くことができる．

$$\left(\frac{\partial u}{\partial t}\right)^2 + \left(\frac{\partial v}{\partial t}\right)^2 \rightarrow \min. \quad (18)$$

2.3.3 ブロックマッチング法と勾配法との比較

ブロックマッチング法はノイズに強いが，計算時間が大きくなり，勾配法は計算時間が短いがノイズに弱いとされている．それらを確認するために，ブロックマッチング法と勾配法の演算時間と精度の比較を行う．

比較は OpenCV2.3 を用いて行う．使用する PC の CPU は PentiumM1.2GHz である．使用する画像は Fig. 17 に示す，橋の画像とする．256×256 ピクセルの元画像から Fig. 18 に示すように 128×128 ピクセルの領域を切り抜いて 2つのフレームの入力画像を作成する．中央から切り出した画像を基準として Fig. 19 に示すように 8方向に領域を移動させて画像を抽出する．領域の移動量はそれぞれ，1ピクセル又は5ピクセルとする．

比較には，ブロックマッチング法(BM)と勾配法として空間的局所最適化法 Lucas&Kanade法(LK)と空間的大域最適化法 Horn&Schunck法(HS)を使用する．

計算条件はそれぞれ以下のようにした．

- ・ ブロックマッチング法

ブロックサイズ : 8×8 pixel

シフトサイズ : 8 pixel

- ・ 空間的大域最適化法(HS)

最大反復回数 : 64

最小変化量 : 0.01

- ・ 空間的局所最適化法(LK)

ウインドウサイズ : 7×7 pixel

Fig. 20～Fig. 25 に OpenCV を用いて計算した結果を示す。中央にテンプレート画像，周囲に 1 ピクセル又は 5 ピクセル領域を移動した画像との計算されたオプティカルフローを示す。表 2～7 には，それぞれの方法で計算したときに得られた 196 領域のオプティカルフローの平均値と標準偏差及び真値を示す。

Fig. 20 と表 2 に 1 ピクセル領域を移動した画像に対してブロックマッチング法を用いてオプティカルフローを計算したときの結果を示す。橋の底面の影になっている部分の一部で正しいオプティカルフローが検出できていない部分があるが，真値と平均値との最大誤差は 0.03，標準偏差の最大値も 0.16 と安定して正確な値が得られた。

Fig. 21 と表 3 には 5 ピクセル領域を移動した画像に対するブロックマッチング法の結果を示す。1 ピクセルずらした場合と同様に，橋の底面の影になっている部分の一部で正しいオプティカルフローが得られていないが，真値と平均値との最大誤差は 0.04，標準偏差の最大値が 0.67 と，安定して正確な値が得られていることがわかる。

Fig. 22 と表 4 に 1 ピクセル領域を移動した画像に対して勾配法の空間的大域最適化法を用いてオプティカルフローを計算したときの結果を示す。結果の画像を見てもわかるように，多くの領域において正しくオプティカルフローが計算できていないことがわかる。このときの真値に対する誤差の最大値は 0.42 (42%)，標準偏差の最大値は 3.62 となっている。

Fig. 23 と表 5 は 5 ピクセル領域を移動した画像に対して同様に勾配法の空間的大域最適化法を用いてオプティカルフローを計算した時の結果である。これも 1 ピクセルの時と同様に多くの領域で正しいオプティカルフローが得られていない。標準偏差の最大値も 50.41 と大きな値となった。

Fig. 24 及び表 6 は 1 ピクセル領域を移動した画像に対して勾配法の空間的局所最適化法を用いてオプティカルフローを計算した時の結果である。Fig. 24 より，橋の欄干に沿って水平方向に誤検出している事がわかる。その他の部分も検出できていない部分がある。このときの真値と平均値との誤差の最大値は 0.79(79%)と大きな値となった。標準偏差の最大値は 1.76 である。

Fig. 25 及び表 7 は 5 ピクセル領域を移動した画像に対する空間的局所最適化法の結果である。1 ピクセルの時と同様に欄干に沿ったオプティカルフローの誤検出が起こっていたり，その他の部分でも正しく検出できていない。標準偏差の最大値は 4.49 である。

Fig. 26 には，領域の移動量が 1 ピクセルの時に，それぞれの方法で検出したオプティカルフローの x 方向の大きさの平均値及び標準偏差を真値と合わせて示す。Fig. 27 は同様に

y方向の結果である。Fig.26, Fig.27より、ブロックマッチング法(BM)が最も真値に近く、標準偏差も小さいことがわかる。空間的大域最適化法(HS)と空間的局所最適化法(LK)は真値との誤差及び標準偏差が大きい。

Fig.28及びFig.29は、それぞれの方法で計算した領域の移動量が5ピクセルの時の結果である。領域の移動量が5ピクセルに増えてもブロックマッチング法が(BM)が真値に近く、標準偏差も小さな値が得られている。

Fig.30はこれらの演算にかかった時間の平均値とその標準偏差である。空間的大域最適化法が最も時間がかかっており約350ms必要とした。ブロックマッチング法の演算時間は、領域の移動量が1ピクセルの時は16msで領域の移動量が5ピクセルの時は31msであった。空間的局所最適化法の演算時間は領域の移動量に関わらず16msとなった。

実験結果より、本オプティカルフロー演算システムには最も誤差が少なく安定しており、演算時間も空間的局所最適化法と大差なかったブロックマッチング法を用いる。テンプレート領域と探査領域の評価関数には、FPGAに実装しやすい差の絶対値の和であるSADを使用する。

2.3.4 テンプレート領域の検討

テンプレートを大きくすれば、ノイズに強くなるが、小さな物体の動きに対して対応できない。また、テンプレートあたりの演算時間はテンプレートの面積に比例して大きくなる。逆に、テンプレートが小さければ、小さな物体の動きもとらえることができるようになるが、ノイズに弱くなり。ノイズ耐性と演算時間はトレードオフの関係にある。そこで、適切なテンプレートの大きさを決定する必要がある。

FPGAへの実装にはテンプレートの一辺の大きさが2のべき乗で表せることが望ましい。使用するカメラの画素数は128×128ピクセルなので、Fig.31に示すようにテンプレートの大きさを8×8ピクセルとし、オプティカルフローの検出領域は、画像の周囲8ピクセルを使用しない14×14の領域とした。

2.3.5 探索範囲の決定

探索範囲を大きくすれば，探索範囲の面積に比例して演算時間が増える．しかし，探索範囲が小さければ大きな運動を検出することができなくなる．テンプレートの大きさと同様に適切な探索範囲を決定する必要がある．

Fig.32 に示すように探索範囲を横方向 x ，縦方向 y ともテンプレートを中心とした ± 7 ピクセルの範囲である 22×22 ピクセルの範囲とした．こうすることで，各方向のオプティカルフローの大きさを 4 ビットで表すことができ，横方向と縦方向合わせて 1 バイトで表すことができ，演算結果を外部に出力する場合に容易に通信できる．

2.3.6 演算時間の見積もり

上述したように，テンプレート領域を 8×8 ピクセル，探索範囲を ± 7 ピクセルと決定したので，この条件での演算時間の指標としてループ回数を推定する．

まずは，SAD を求める式(1)について考える．式(1)では，テンプレートとテンプレートと同サイズに切り取った探索領域の差分の絶対値を求める．よって，テンプレートサイズである 8×8 回のループ計算が必要となる．

次に，テンプレートと同サイズの領域を探索領域内で移動することを考える．探索領域はテンプレートを中央に置いたときに ± 7 ピクセルの範囲としたので， x, y 方向共に 15 ピクセルずらしながら SAD を計算する．これより， 15×15 回のループが必要となる．

最後に，オプティカルフローの探索領域数を考える．探索領域は Fig.31 に示すように 14×14 個配置した．これら全てに上述の x, y 方向にずらしながら SAD 計算を行う必要がある．これより， 14×14 回のループが発生する．

これらを合わせると，総ループ数は以下ようになる．

$$8 \times 8 \times 15 \times 15 \times 14 \times 14 = 2822400 \quad (19)$$

ひとつの計算を 1 クロックで行えると仮定し，100MHz のクロックを用いてこの計算を行った場合，約 28ms 必要となる．式(19)は単純なループ回数だけを求めたが，実際には計算を行うためにメモリからデータを読み出す必要がある．そのため，アドレス指定，読み出しクロック入力が必要となり，実際には約 3 倍の計算時間が必要となる．よって，計算時間は約 84ms となり，約 12fps で動作することとなる．これでは，十分な速度が得られない．よって，オプティカルフロー演算方法を工夫し演算の高速化を行う必要がある．

2.3.7 処理の高速化

演算時間を見積もった結果、100MHzのクロックを用いた演算時間が約84ms(約12fps)となり、十分な速度が得られないことがわかった。これは、逐次処理で演算を行った場合の時間であり、FPGAを用いれば処理の並列化が可能で、演算の高速化が可能である。PCでオプティカルフローを演算する場合はマルチメディア命令を用いた高速化[66]や、テンプレートと比較領域のそれぞれの画素の総和を利用して、比較する領域を削減する方法[67]などが行われている。

そこで、Fig. 33に示すように、探査領域で x, y にテンプレートと同サイズの領域をずらす処理のうち、 x 方向への移動を並列化した。 x 方向には、15ピクセル分ずらしながら計算する必要があるが、これを全て同時に行う。しかし、演算回路を並列化しても画素データの読み出しは並列化できない。複数の画素データを同時に読み出すことは難しい。そこで、並列化した15個の領域の、最初の画素データを読み込み、差分計算を行い、次の探査領域の読み込みを行う際、差分器に入力した探査領域の画素データは、1つ左の差分器にデータを渡すだけでテンプレートの次の画素との差分に再利用できる。そこで、15バイト分のシフトレジスタを用意し、差分器それぞれの探査領域用のデータの保存に使用した。これにより、15画素同時の差分計算を実現した。

差分計算には、Fig. 34に示すようなFPGAに内蔵されているDSPスライス[71]を用いた。DSPスライスを用いることで、加減演算および乗算演算を1クロックで行う事ができる。絶対値の演算にはDSPスライスへの入力であるテンプレートと探査領域の画素データを入れ替えて演算を行い、演算結果が正となる方を選択する。また、差分値の総和演算にもDSPスライスを用いる。

これらのことから、差分計算に15個、逆の入力の差分計算に15個、総和演算に15個の合計45個のDSPスライスが必要になる。

DSPスライスを用いることで、100MHz以上の高速な演算を行う事ができるが、本システムで用いたDSPスライス1つの入力ポート(Cポート)が2つのDSPスライスで共通となっており、DSPスライスは16組32個しか搭載されていない。そのため、DSPスライスが不足し、1つのDSPスライスで1つの演算を行う事はできない。

そこで、SIMD演算[70]を参考に、1つのDSPスライスで複数の画素データの演算を行い、DSPスライスの使用量を削減する。DSPスライスの構成[71]をFig. 34に示す。Aポートは18ビットの入力、Bポートも18ビットの入力、Cポートは2つのDSPスライスで共通の48ビットの入力ポートとなっている。出力は48ビットのPポートである。本システム

では Fig. 34 中の乗算器は使用せずに、加減算器を用いる。加減算器では、C ポートの入力と、A ポートと B ポートとの入力を結合したものの加減算を行う事ができる。

DSP スライスへの入出力データを Fig. 35 に示す。Fig. 35 の上段に示すように、C ポートへの入力には同じテンプレートの画素データを 4 つ入力している。4 つの入力データには 1 ビットの '1' を追加した 9 ビットのデータとして入力している。A ポートと B ポートには探索領域の 4 つの画素データを入力している。この際、1 ビットの '0' を追加した 9 ビットのデータとして入力している。Fig. 35 の下段には上段の C ポートと、A ポート・B ポートの入力データが逆のものを示す。このように入力することで、DSP スライスからの出力ポートである P ポートからは、テンプレート-探索領域と、探索領域-テンプレートの 2 種類の計算結果を得ることができる。この出力データの符号ビットを確認し、'1' となっている計算結果を採用することで、絶対値の演算を行う。

2.4 FPGA への実装

これまでに決定したオプティカルフロー演算アルゴリズムや、周辺のマジュールを FPGA へ実装する。

2.4.1 システム構成 (USB 接続)

Fig. 36 に PC と USB 接続を行う場合のオプティカルフロー演算システムにおけるカメラボードと FPGA ボードのシステム構成を示す。カメラボードにはシリコン網膜カメラの主となるビジョンチップと、ビジョンチップからのデータを FPGA で読み取れるデジタルデータに変換する AD コンバータ、ビジョンチップの抵抗ネットワークや AD コンバータのリファレンス電圧を設定するための DA コンバータが搭載されている。FPGA ボードには FPGA やコンフィギュレーションメモリなどが搭載されている。Fig. 36 には FPGA 内部に製作するモジュール構成を示す。モジュールはカメラボードの各モジュールを制御するためのカメラコントローラと、カメラから読み取ったデータを保存するためのメモリコントローラ、オプティカルフローを計算するためのオプティカルフロー演算器、オプティカルフローの演算結果と映像データを出力するための USB コントローラで構成される。

・カメラコントローラ

カメラボードには先にも述べたようにビジョンチップ，DA コンバータ及びAD コンバータが搭載されている．カメラコントローラはこれら3つを制御する機能として，チップコントローラ，DA コントローラ，AD コントローラの3つのモジュールから構成される．

DA コントローラはカメラボードに搭載されている DA コンバータを制御し，この DA コンバータはビジョンチップの抵抗ネットワークの結合強度を設定し，AD コンバータのリファレンス電圧を作成する．このため，DA コントローラは起動時に一度だけ実行する．

チップコントローラは，ビジョンチップを制御する．ビジョンチップを制御する信号は一定周期ごとに繰り返し出力する必要があるため，あらかじめ決められた時間で遷移するステートマシンによって制御を行った．

AD コントローラはビジョンチップからのデータを読み出すための AD コンバータと読み出す画素を指定するシフトレジスタを制御する．

・メモリコントローラ

メモリコントローラはビジョンチップから読み出された画像データを保存する．画像データを保存するためのメモリは，オプティカルフロー演算時に用いるテンプレートと探査領域用の2つの領域を持つ．メモリコントローラは，ビジョンチップから読み出された画像データを，1フレーム目はテンプレート用のメモリに保存し，2フレーム目は探査領域用のメモリに保存する．

・オプティカルフロー演算器

オプティカルフロー演算器はメモリコントローラ内に保存されている2つのカメラ画像データに対して，ブロックマッチング法を用いて対応点を検出するモジュールである．

・USB コントローラ

USB コントローラは USB インタフェースボードに搭載されている USB インタフェース IC とのデータの受け渡しを行う．USB インタフェース IC は USB2.0 のハイスピードに対応している Cypress 社の EZ-USB FX2 を用いた．PC と EZ-USB FX2 間の転送モードは速度を重視しバルク転送を用いた．

USB コントローラは EZ-USB FX2 に対して，FIFO と同様の方法でデータ転送を行う．

2.4.2 動作タイミング (USB 接続)

カメラコントローラ、メモリコントローラ、オプティカルフロー演算モジュールの動作タイミングを Fig. 37 に示す。ビジョンチップから画像データを読み取ると同時にメモリコントローラは画像データをメモリに保存する。テンプレート用と探査領域用の2つのメモリに画像データを保存した後、オプティカルフロー演算器によってオプティカルフローが演算される。カメラからの画像は 20ms 周期(50fps)で得ることができ、このとき、画像の読み取りにかかる時間は 4.1ms である。また、オプティカルフローの演算にかかる時間は 17.5ms である。このため、オプティカルフローの演算は 3 フレーム毎に行うことになる。このため、Fig. 37 における 2 つ目から 4 つ目のフレーム間のオプティカルフローが得られない。

そこで、メモリコントローラを改良し Fig. 38 のように 3 つのメモリを用いて画像データを保存するようにした。オプティカルフローの演算中には、オプティカルフローの演算に使用していない 3 つ目のメモリに画像データを保存することにより、全てのフレーム間のオプティカルフローを得ることができる。

2.4.3 システム構成 (UART 接続)

UART 接続タイプのシステム構成を Fig. 39 に示す。こちらは USB コントローラの代わりに PowerPC を搭載したモデルである。計測されたオプティカルフローは PowerPC を用いて演算することができる。これにより、後に述べるエゴモーション推定やドミナントプレーンの検出アルゴリズムを実装し、容易に UART を用いてロボットと通信を行う事ができる。PowerPC を用いるためにメモリコントローラが持つメモリを 3 つから 2 つに減らしたが、次に述べるような動作タイミングとすることで、取得した画像データに対して連続的にオプティカルフローを演算した。

2.4.4 動作タイミング (UART 接続)

Fig. 40 に UART 接続タイプの動作タイミングを示す。探査領域として使用した画像データをテンプレートとして使用することで、フレーム間の時間変動するが、画像メモリを 2 つに減らしても連続したオプティカルフローを求めることができる。

2.5 おわりに

本章では、動きの検出システムであるオプティカルフロー演算システムに用いる演算装置とカメラ、オプティカルフロー演算アルゴリズムの検討を行い、それらを用いてオプティカルフロー演算システムの開発について述べた。

演算装置には、低消費電力で並列化することによって演算速度の向上が見込める FPGA を用い、カメラにはダイナミックレンジが広い、網膜を模して作られたシリコン網膜カメラを用いた。オプティカルフロー演算アルゴリズムには、安定して正確な値が得られたブロックマッチング法を用いた。

これらを用いてオプティカルフロー演算システムを開発した。ブロックマッチング法の実装の際には、DSP スライスを用いた並列演算を行う事で高速化を行った。これにより、一般的なカメラのフレームレート 30fps(33ms)より高速な 17ms で動作可能なオプティカルフロー演算システムを開発した。

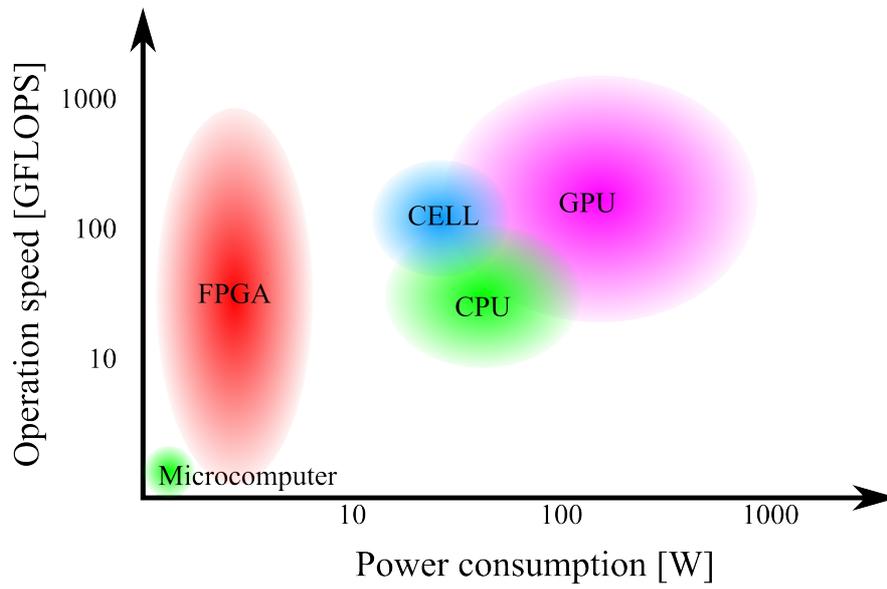


Fig.9 Operation speed - Power consumption of Processor

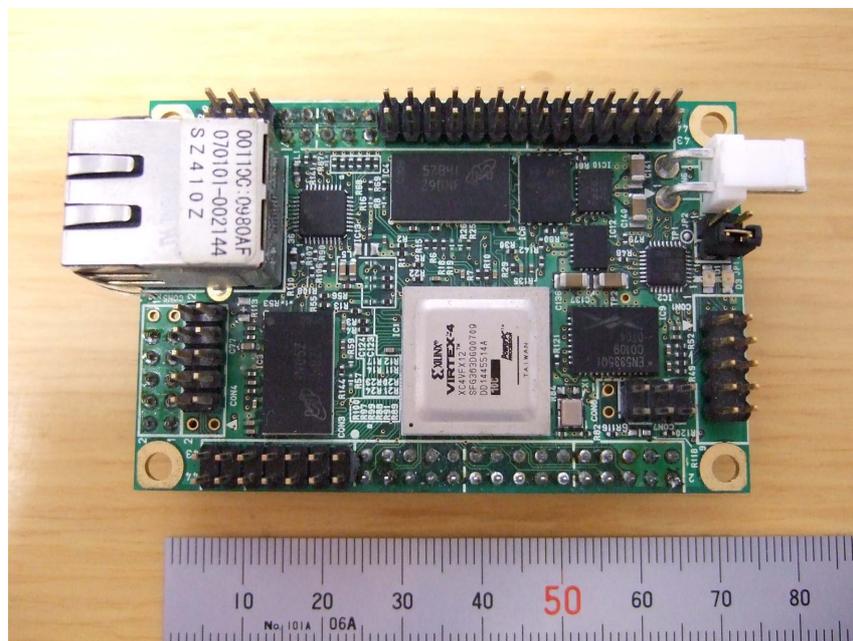


Fig.10 FPGA board (Atmark Techno "SUZAKU V")

Table 1 Specification of FPGA board

| | |
|--------------------------|-----------------------------------|
| FPGA | Xilinx Virtex-4 XC4VFX12-SF363 |
| CPU core | PowerPC405 |
| Size of board | 72 × 47 mm |
| Power consumption (Typ.) | 3 W |

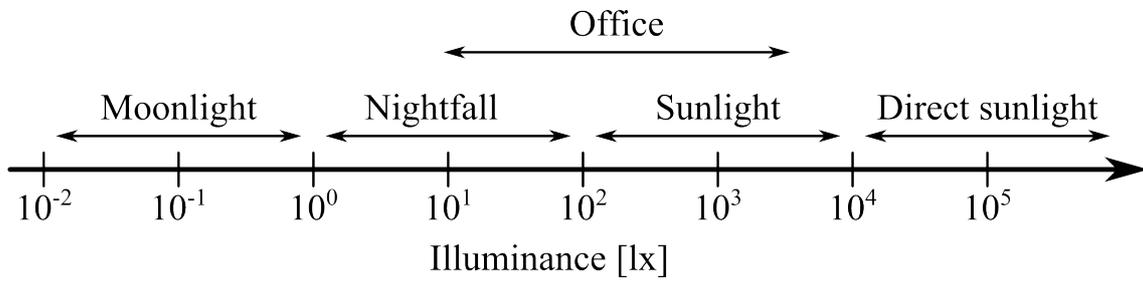


Fig.11 Illuminance in the living environment

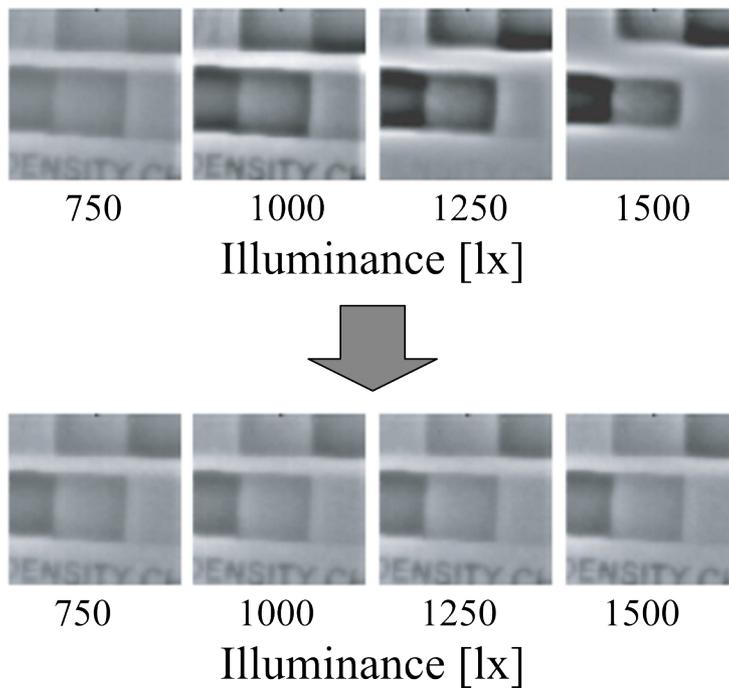


Fig.12 Constancy of illuminance[46]



Fig.13 Overview of silicon retina camera

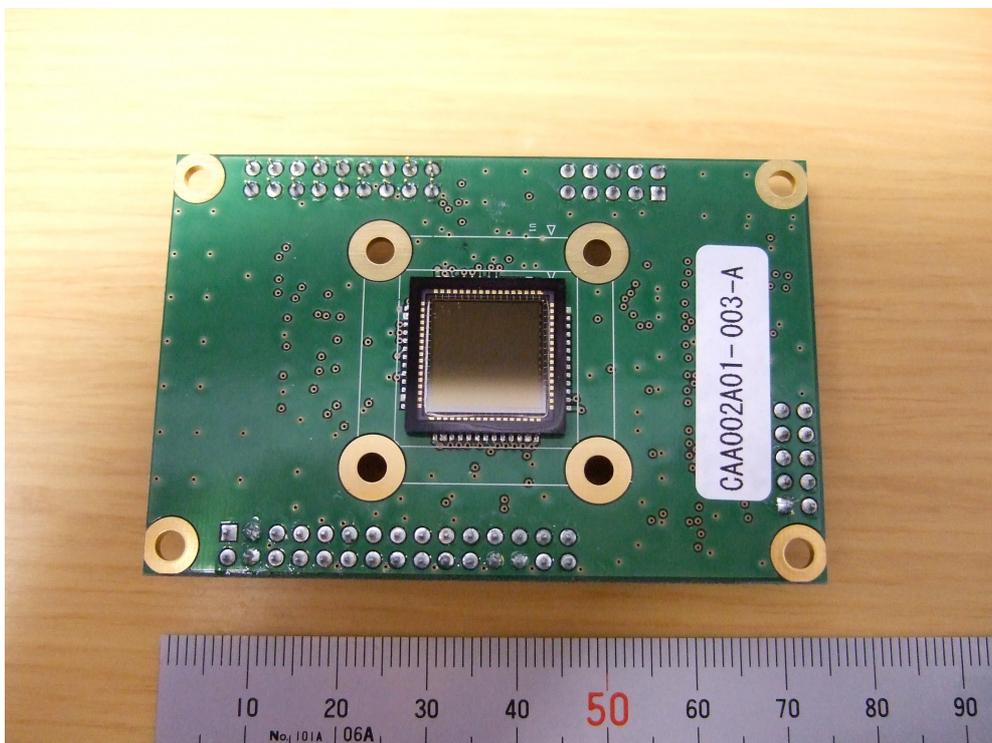


Fig.14 Vision chip

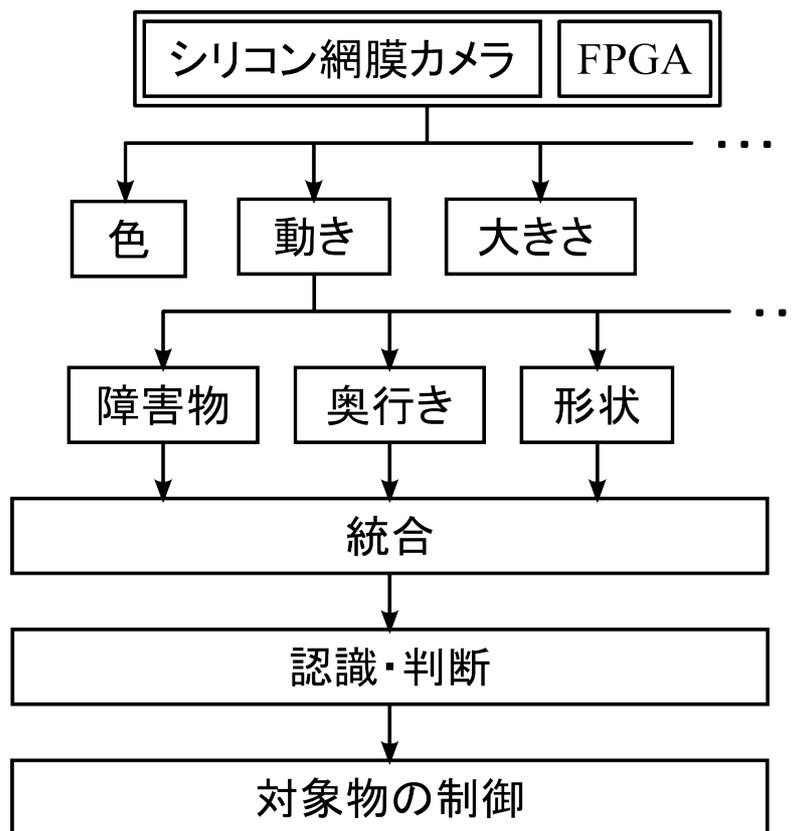


Fig.15 Vision system

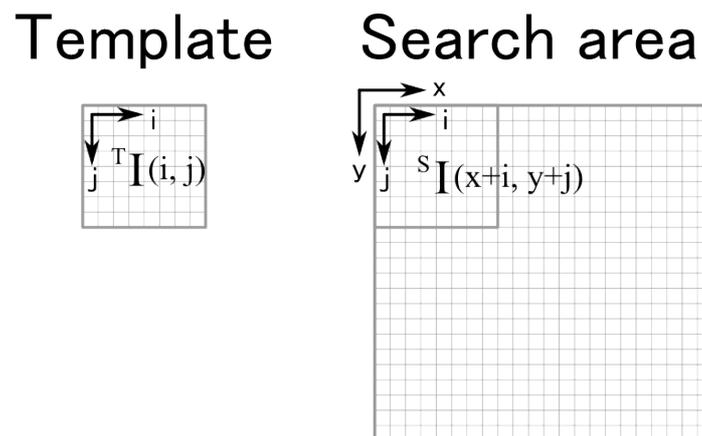


Fig.16 Template and Search area



Fig.17 Original Image

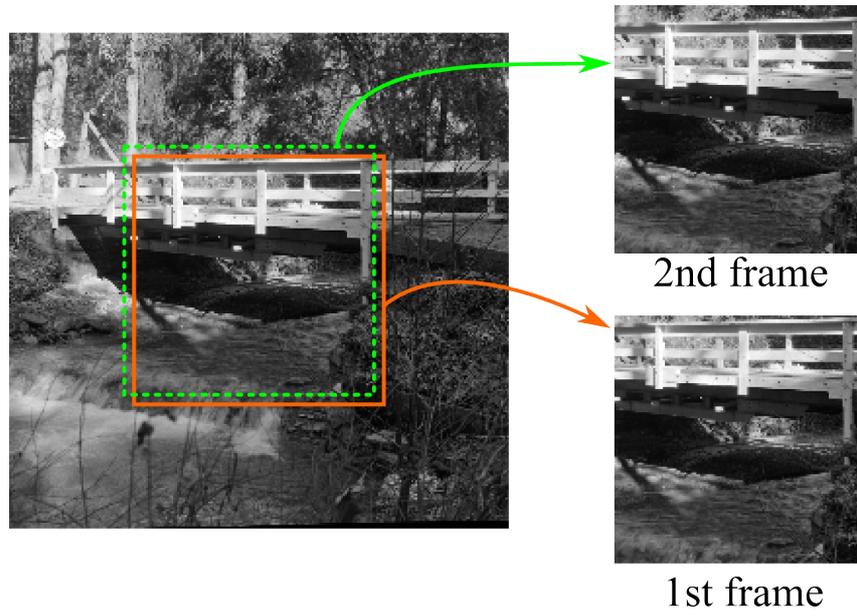


Fig.18 1st frame and 2nd frame

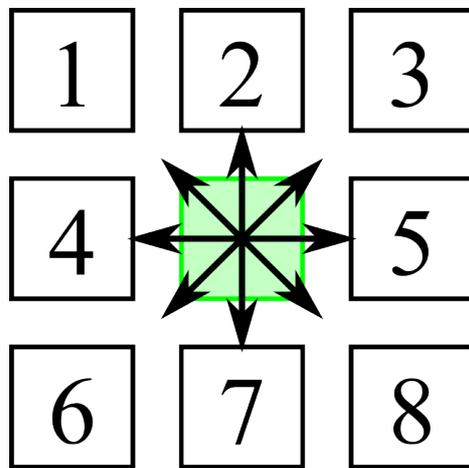


Fig.19 Image shift direction

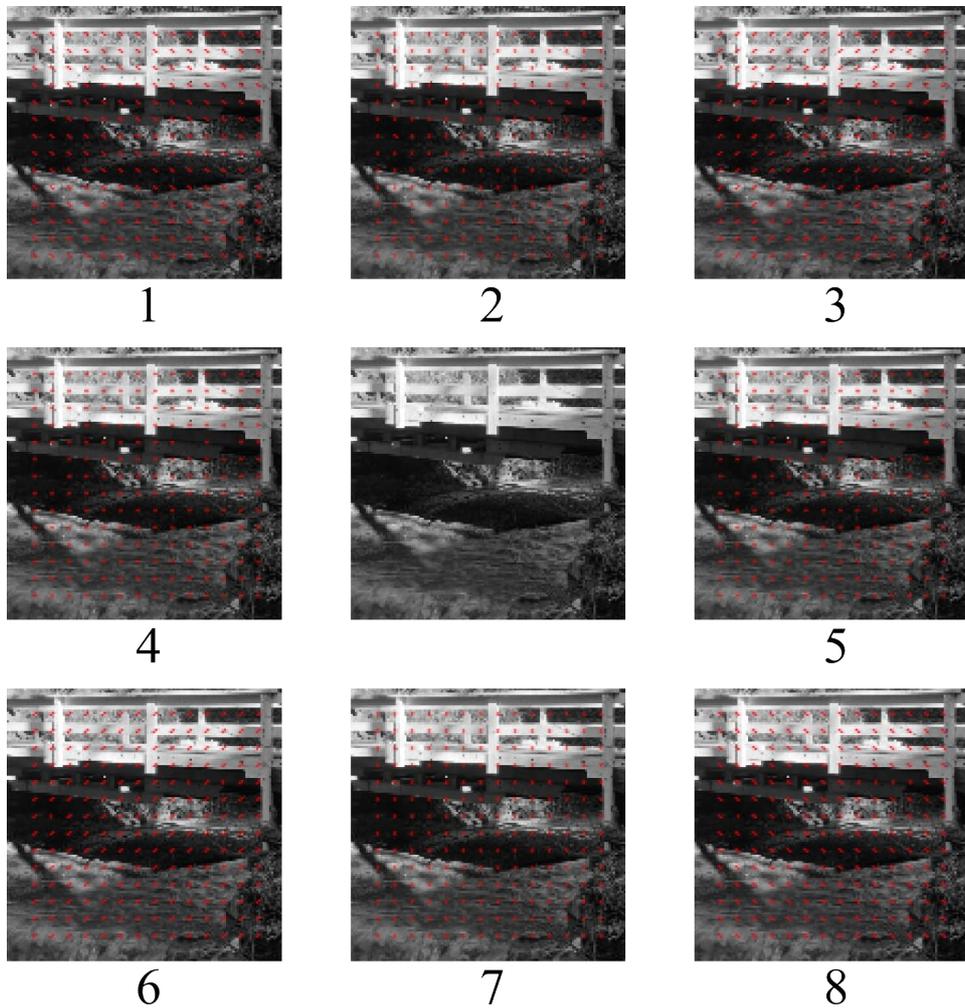


Fig.20 Result of operation (Block matching method, 1pixel shift)

Table 2 Result of operation (Block matching method, 1pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|------|-------|------|------|
| 真値 | x | -1 | 0 | 1 | -1 | 1 | -1 | 0 | 1 |
| | y | -1 | -1 | -1 | 0 | 0 | 1 | 1 | 1 |
| 平均 | x | -0.98 | 0.00 | 0.97 | -0.98 | 0.97 | -0.98 | 0.02 | 1.00 |
| | y | -1.00 | -1.00 | -1.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| 標準偏差 | x | 0.14 | 0.00 | 0.16 | 0.14 | 0.16 | 0.14 | 0.14 | 0.00 |
| | y | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

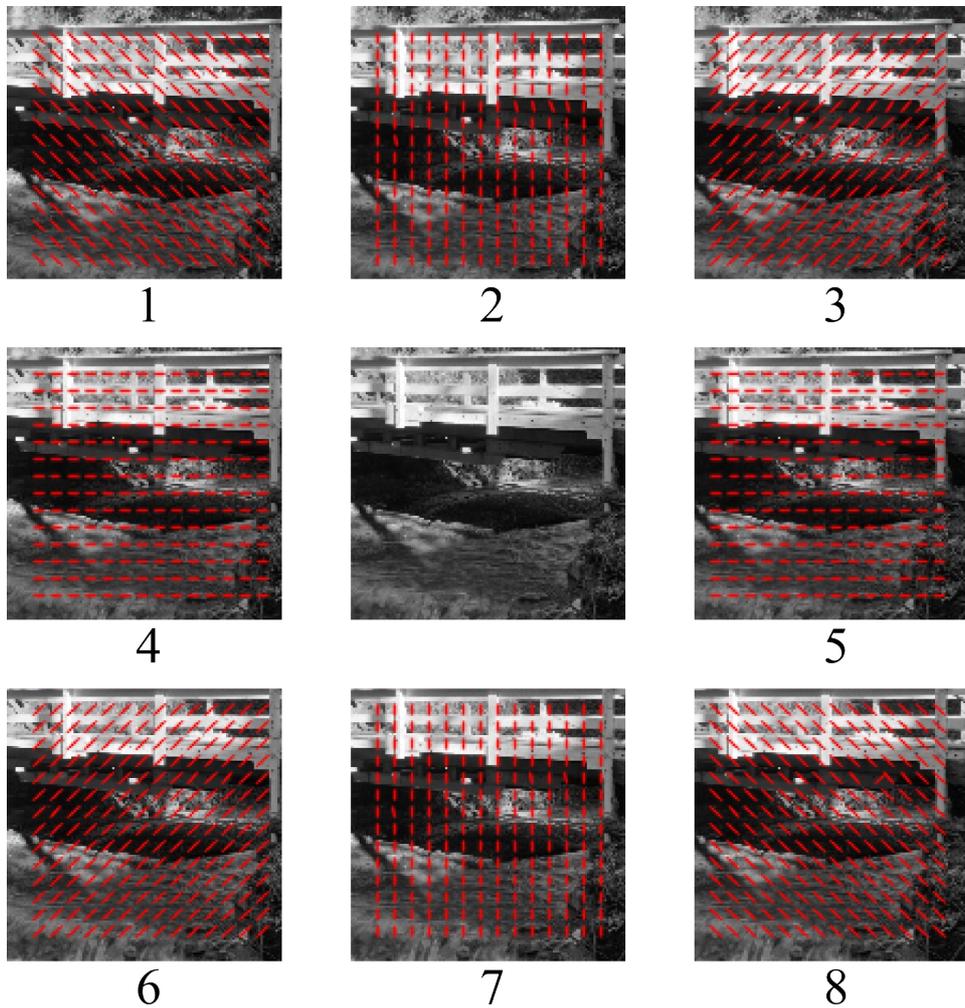


Fig.21 Result of operation (Block matching method, 5pixel shift)

Table 3 Result of operation (Block matching method, 5pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|-------|-------|------|------|
| 真値 | x | -5 | 0 | 5 | -5 | 5 | -5 | 0 | 5 |
| | y | -5 | -5 | -5 | 0 | 0 | 5 | 5 | 5 |
| 平均 | x | -4.97 | -0.01 | 4.96 | -4.96 | 4.93 | -4.97 | 0.01 | 4.99 |
| | y | -4.99 | -4.98 | -4.99 | 0.00 | -0.01 | 5.00 | 4.99 | 4.99 |
| 標準偏差 | x | 0.20 | 0.19 | 0.23 | 0.23 | 0.67 | 0.22 | 0.16 | 0.07 |
| | y | 0.10 | 0.22 | 0.07 | 0.00 | 0.07 | 0.00 | 0.14 | 0.07 |

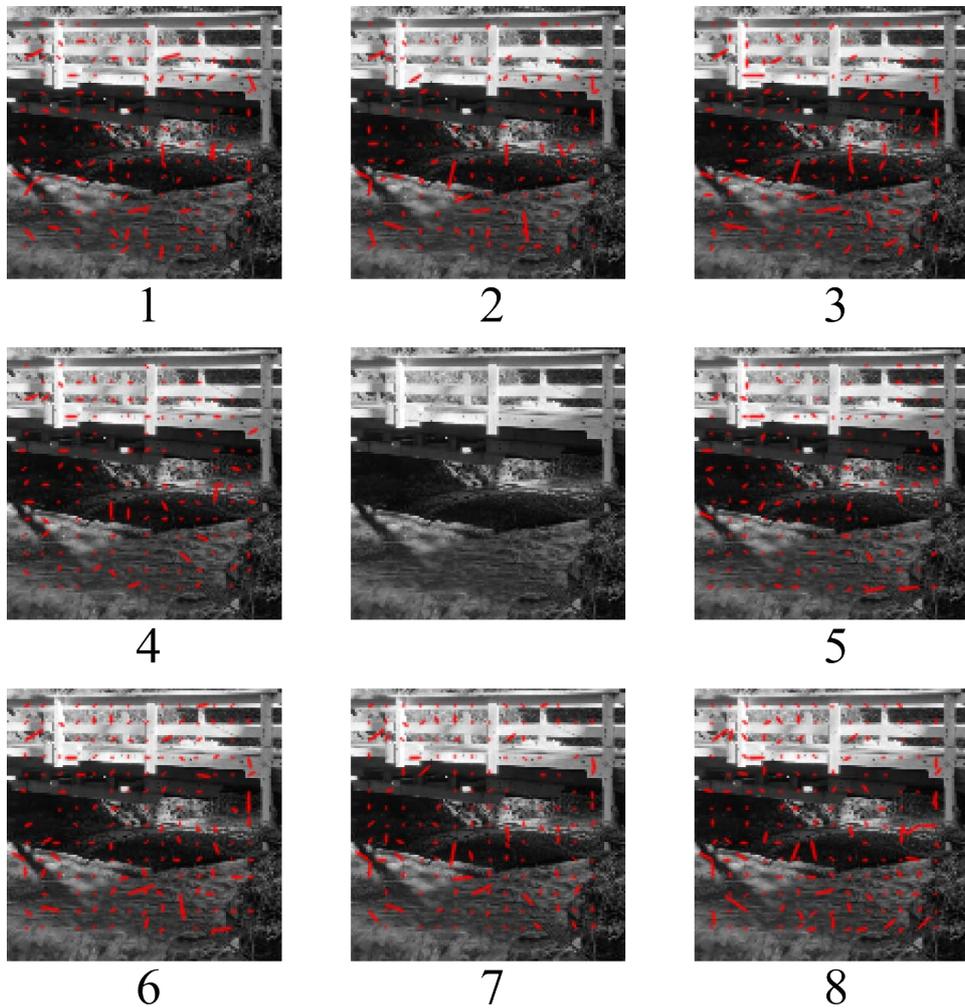


Fig.22 Result of operation (Horn & Schunck method, 1pixel shift)

Table 4 Result of operation (Horn & Schunck method, 1pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|-------|-------|-------|------|
| 真値 | x | -1 | 0 | 1 | -1 | 1 | -1 | 0 | 1 |
| | y | -1 | -1 | -1 | 0 | 0 | 1 | 1 | 1 |
| 平均 | x | -0.64 | -0.23 | 0.57 | -0.65 | 0.74 | -0.31 | -0.01 | 0.68 |
| | y | -0.58 | -0.57 | -0.76 | 0.08 | -0.03 | 0.59 | 1.09 | 0.77 |
| 標準偏差 | x | 2.11 | 2.59 | 2.25 | 1.83 | 1.96 | 2.32 | 2.11 | 3.62 |
| | y | 1.61 | 2.12 | 2.02 | 1.32 | 1.07 | 2.62 | 3.33 | 2.07 |

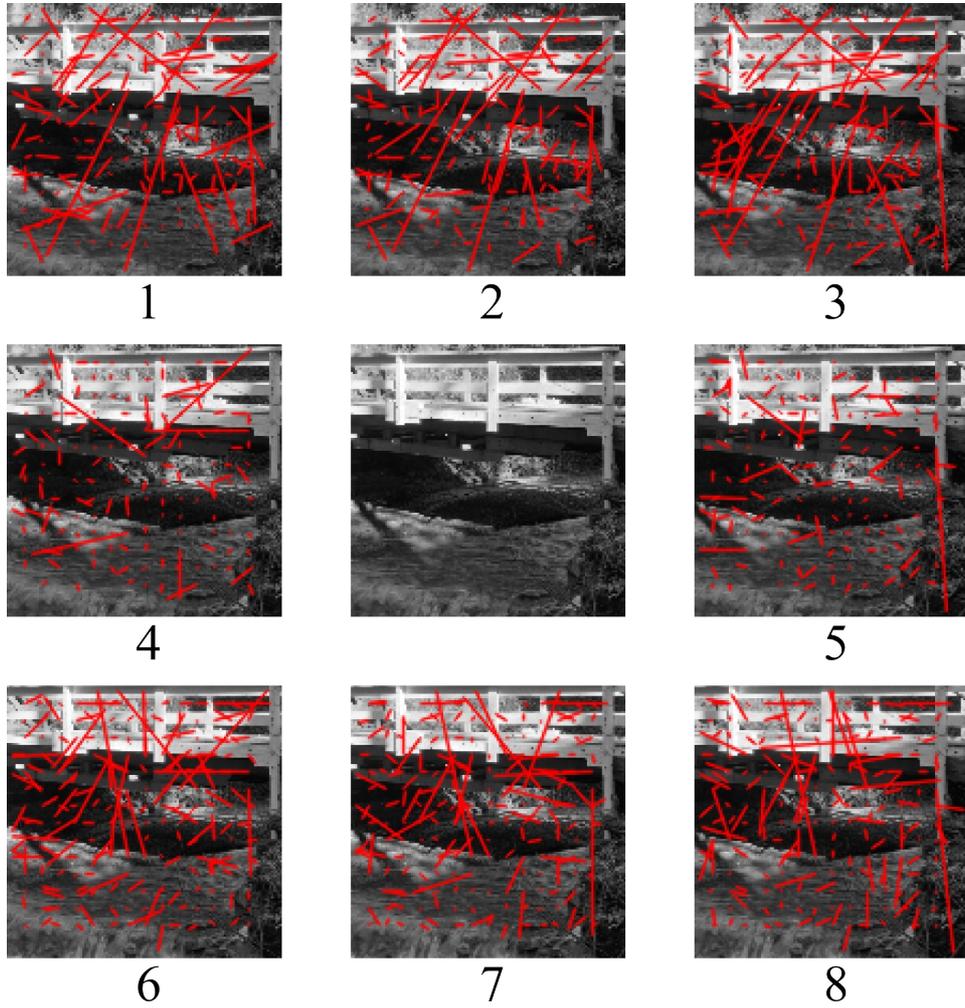


Fig.23 Result of operation (Horn & Schunck method, 5pixel shift)

Table 5 Result of operation (Horn & Schunck method, 5pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| 真値 | x | -5 | 0 | 5 | -5 | 5 | -5 | 0 | 5 |
| | y | -5 | -5 | -5 | 0 | 0 | 5 | 5 | 5 |
| 平均 | x | 0.67 | 0.10 | 0.01 | -1.18 | -0.12 | -1.12 | -1.39 | -0.55 |
| | y | -0.67 | 0.55 | -1.66 | -0.11 | 0.17 | 1.27 | -1.38 | 0.73 |
| 標準偏差 | x | 16.52 | 12.16 | 12.04 | 7.67 | 6.98 | 18.37 | 50.41 | 12.91 |
| | y | 18.46 | 10.91 | 13.86 | 8.18 | 5.26 | 25.30 | 45.88 | 12.45 |

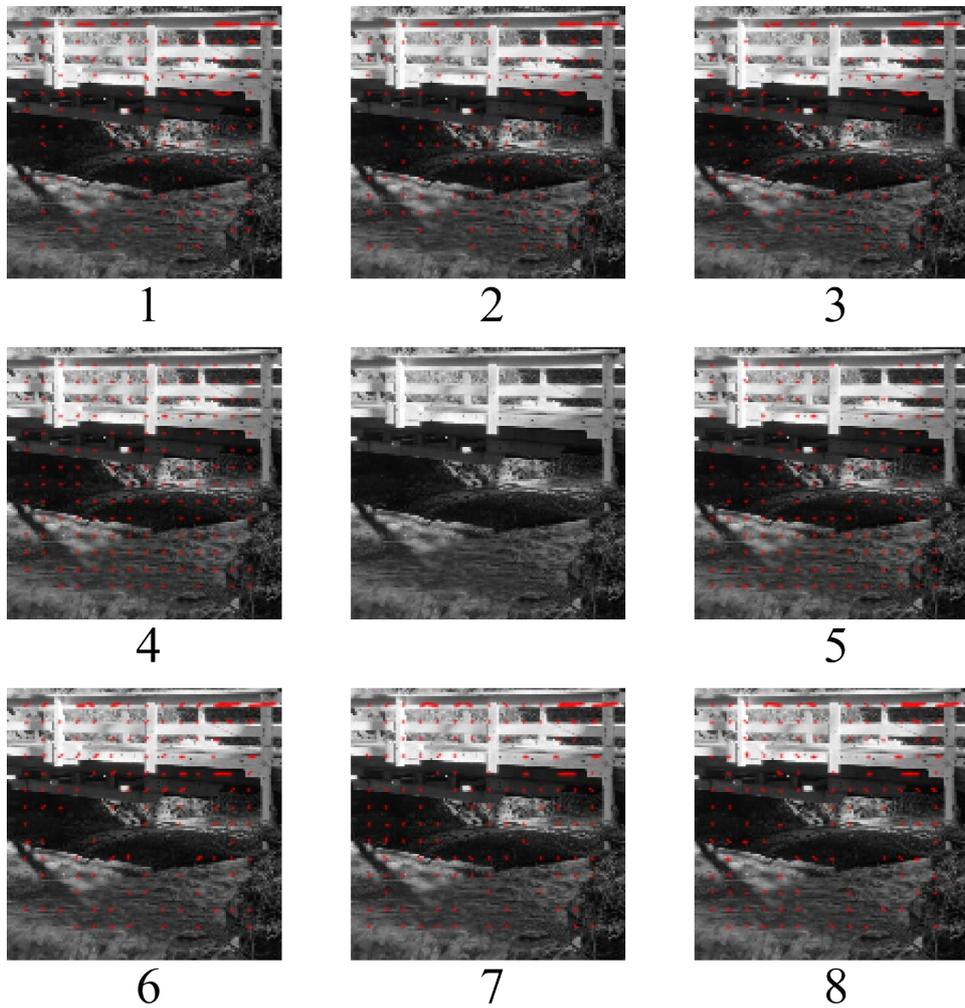


Fig.24 Result of operation (Lucas & Kanade method, 1pixel shift)

Table 6 Result of operation (Lucas & Kanade method, 1pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|------|-------|------|------|
| 真値 | x | -1 | 0 | 1 | -1 | 1 | -1 | 0 | 1 |
| | y | -1 | -1 | -1 | 0 | 0 | 1 | 1 | 1 |
| 平均 | x | -0.21 | 0.12 | 0.42 | -0.72 | 0.77 | -0.21 | 0.12 | 0.34 |
| | y | -0.41 | -0.59 | -0.43 | 0.00 | 0.00 | 0.41 | 0.58 | 0.36 |
| 標準偏差 | x | 1.76 | 1.53 | 1.09 | 0.48 | 0.44 | 1.51 | 1.24 | 0.78 |
| | y | 0.50 | 0.49 | 0.52 | 0.00 | 0.00 | 0.52 | 0.51 | 0.49 |

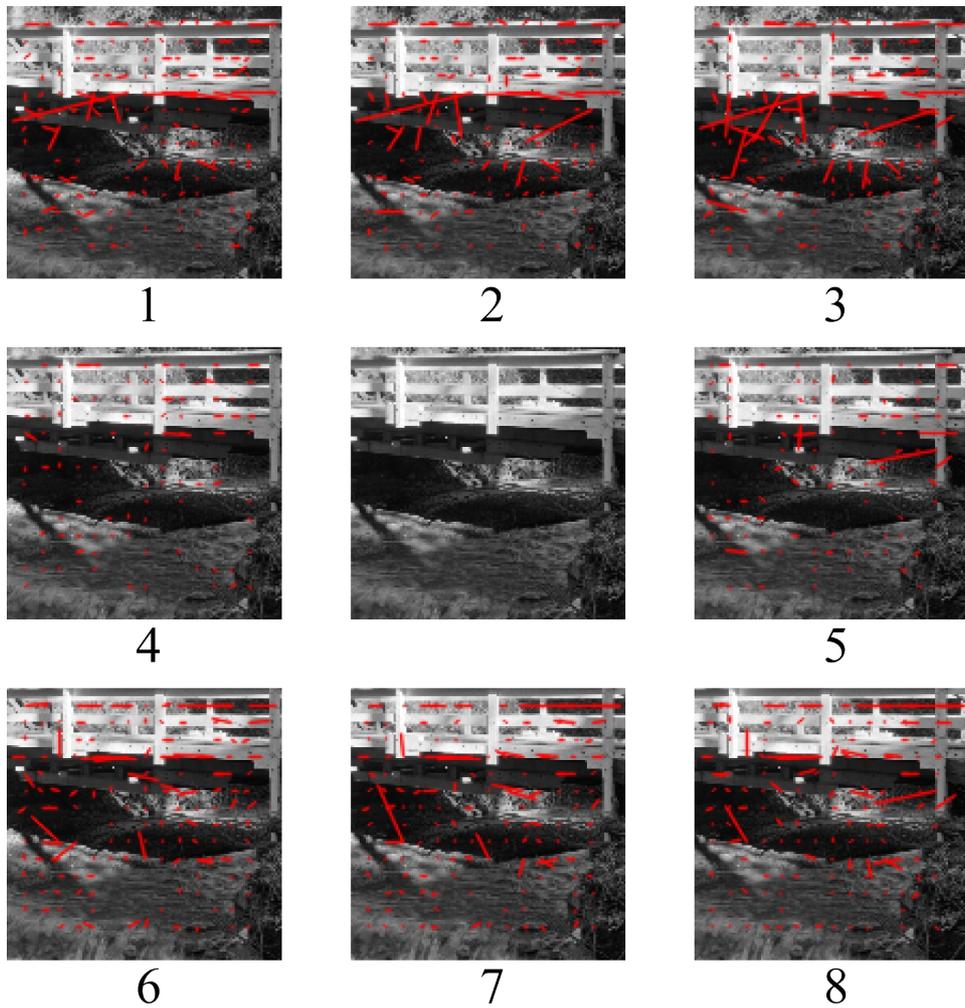


Fig.25 Result of operation (Lucas & Kanade method, 5pixel shift)

Table 7 Result of operation (Lucas & Kanade method, 5pixel shift)

| No. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|-------|-------|-------|-------|-------|-------|-------|-------|
| 真値 | x | -5 | 0 | 5 | -5 | 5 | -5 | 0 | 5 |
| | y | -5 | -5 | -5 | 0 | 0 | 5 | 5 | 5 |
| 平均 | x | -0.56 | -0.03 | -0.04 | -0.52 | 0.40 | -0.91 | -0.36 | -0.37 |
| | y | -0.21 | -0.26 | -0.27 | -0.04 | -0.08 | 0.26 | 0.18 | 0.06 |
| 標準偏差 | x | 4.37 | 4.20 | 3.58 | 1.30 | 1.18 | 4.36 | 4.49 | 3.64 |
| | y | 0.96 | 1.01 | 1.00 | 0.67 | 0.63 | 2.22 | 1.96 | 1.83 |

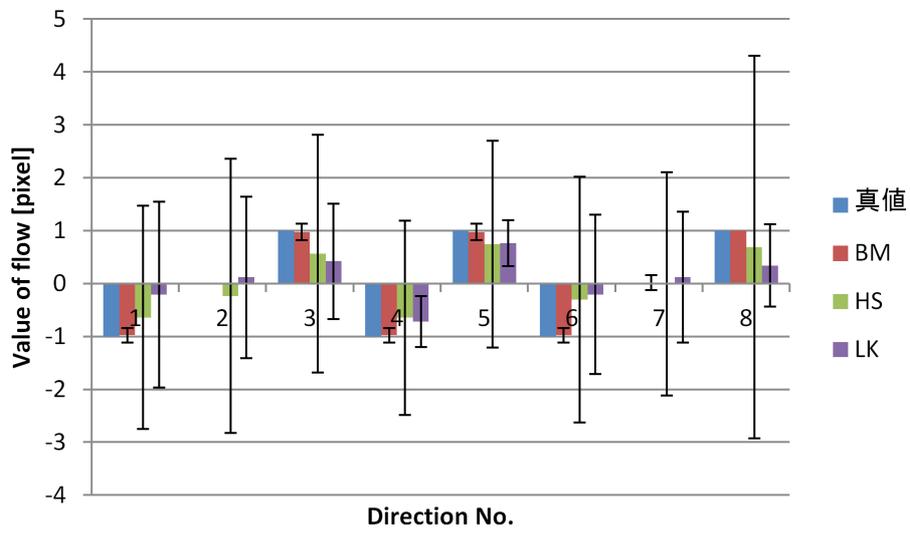


Fig.26 Result of 1 pixel shift (x direction)

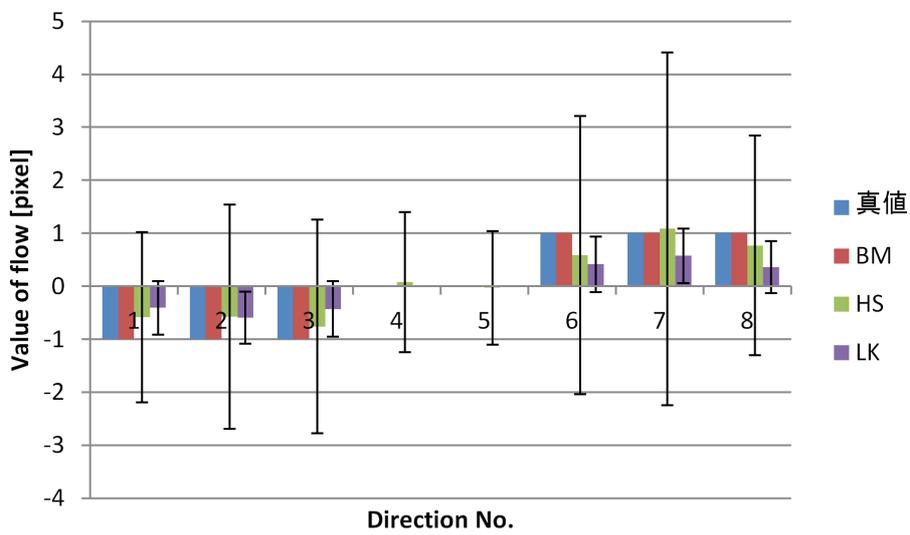


Fig.27 Result of 1 pixel shift (y direction)

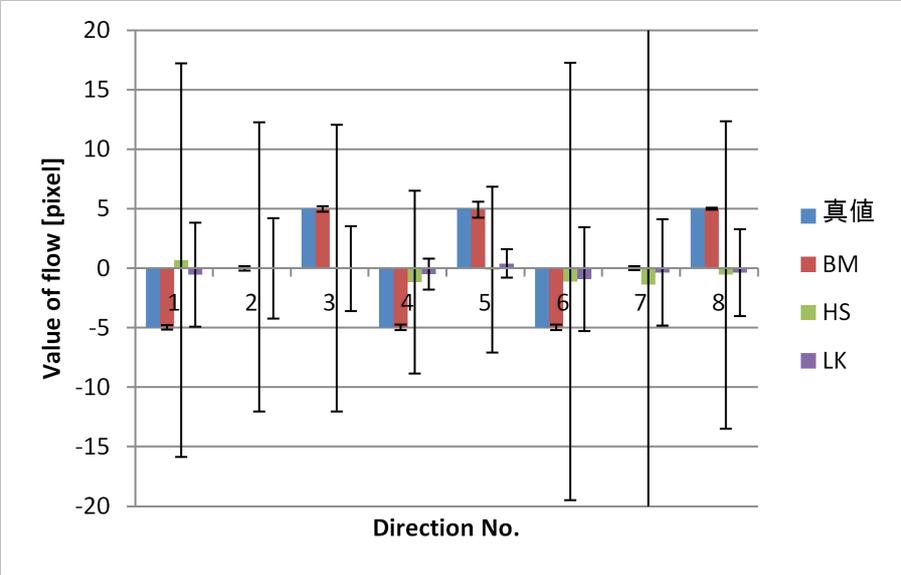


Fig.28 Result of 5pixel shift (x direction)

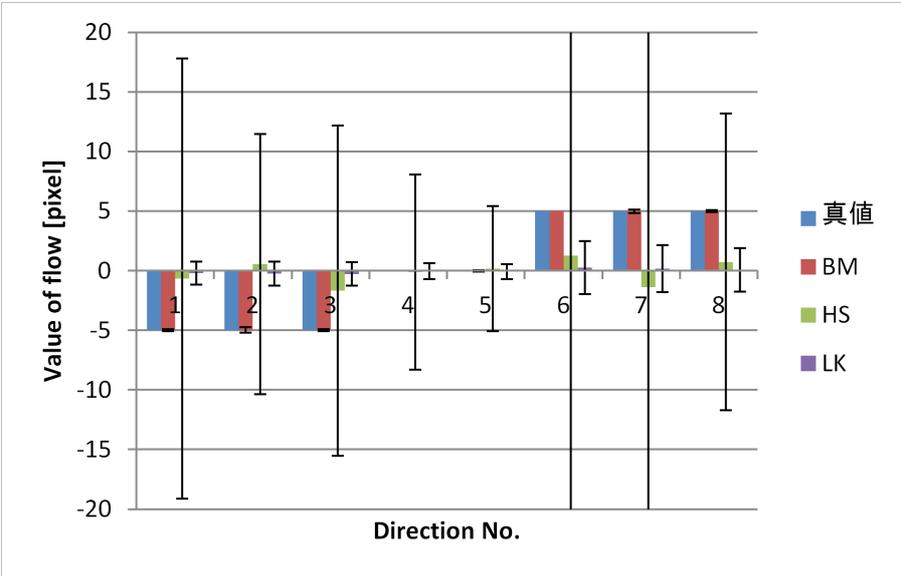


Fig.29 Result of 5pixel shift (y direction)

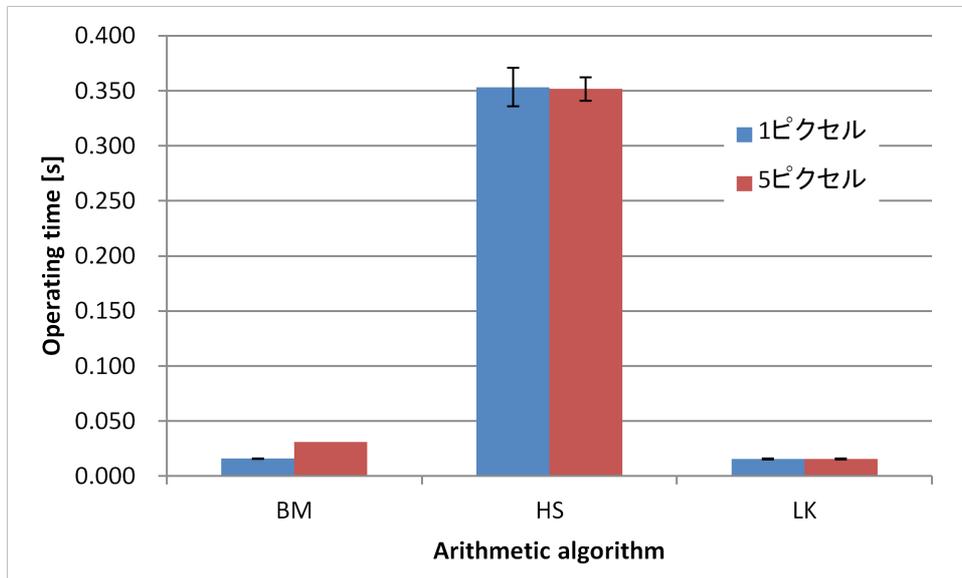


Fig.30 Operating time

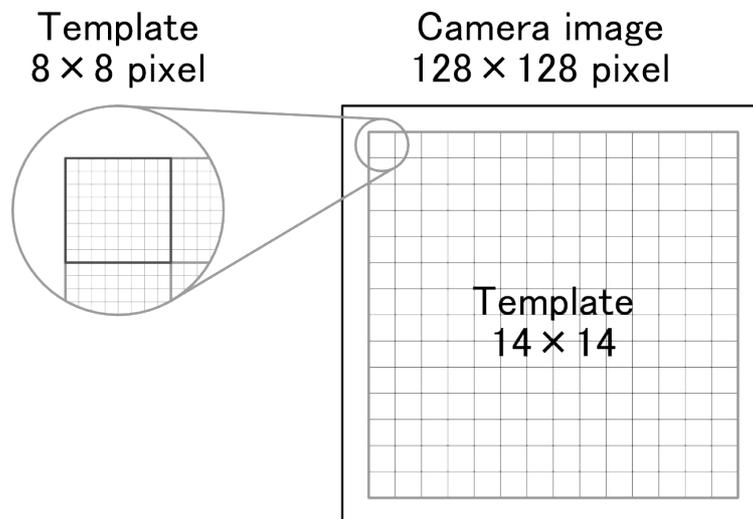


Fig.31 Template

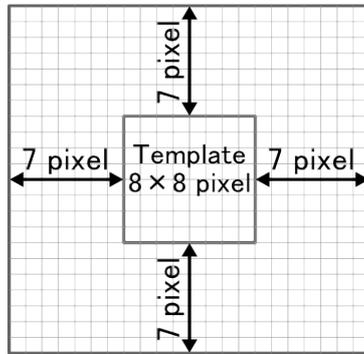


Fig.32 Search area

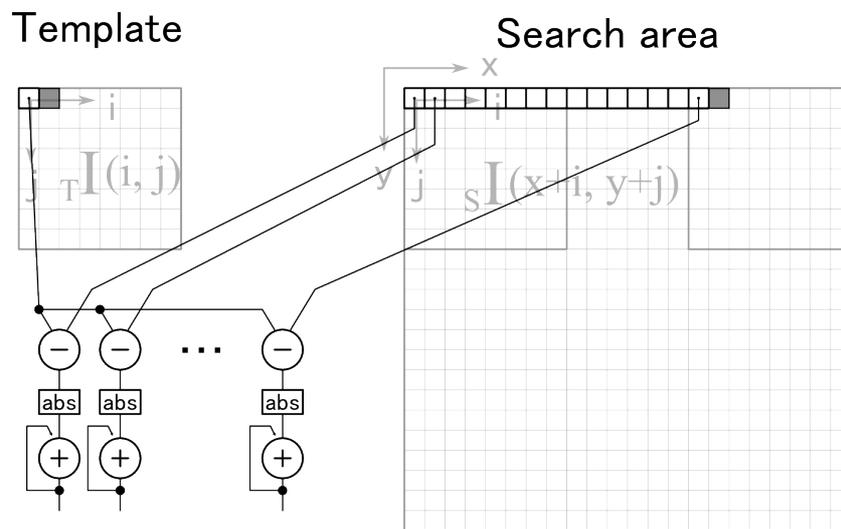


Fig.33 Parallel processing

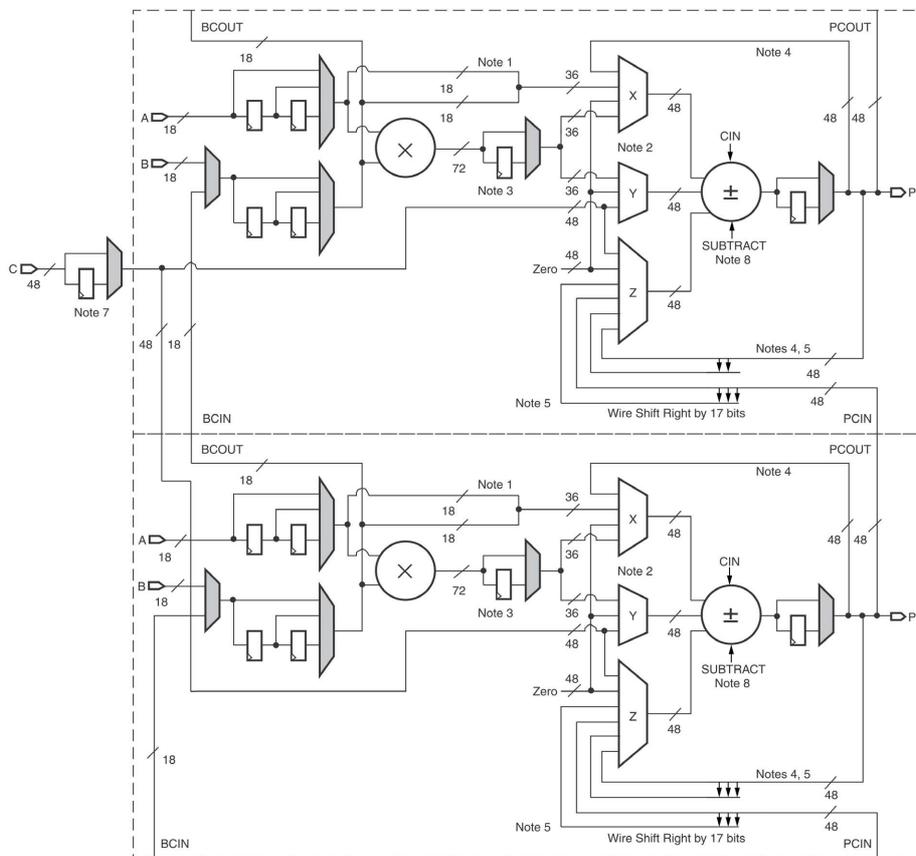


Fig.34 Structure of DSP slice[71]

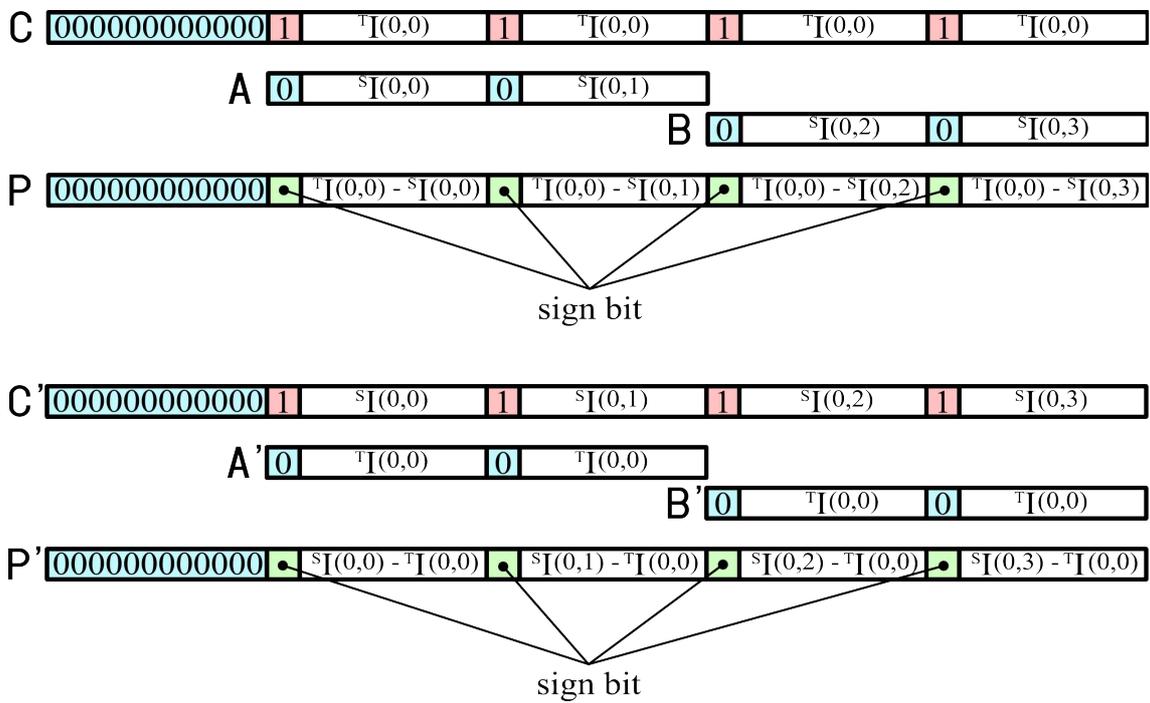


Fig.35 Input and output data of DSP slice

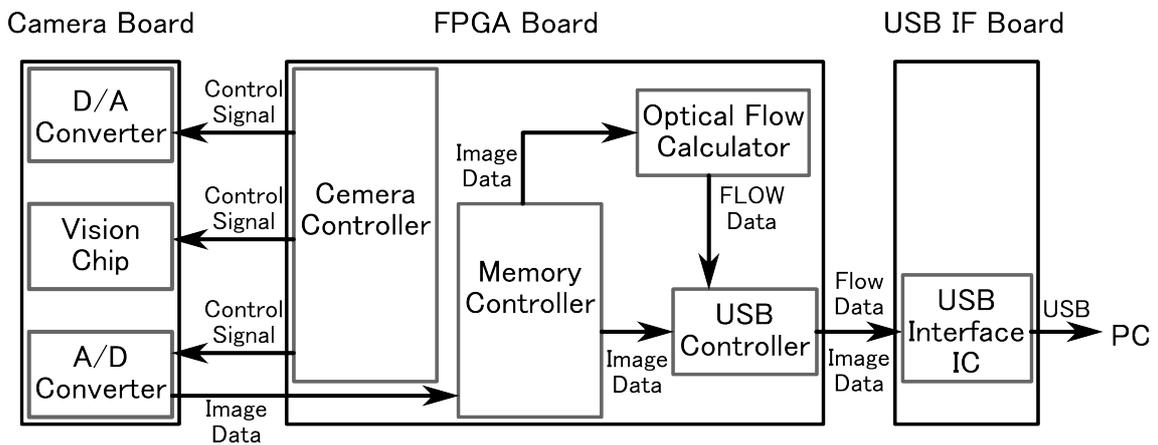


Fig.36 System configuration (USB IF version)

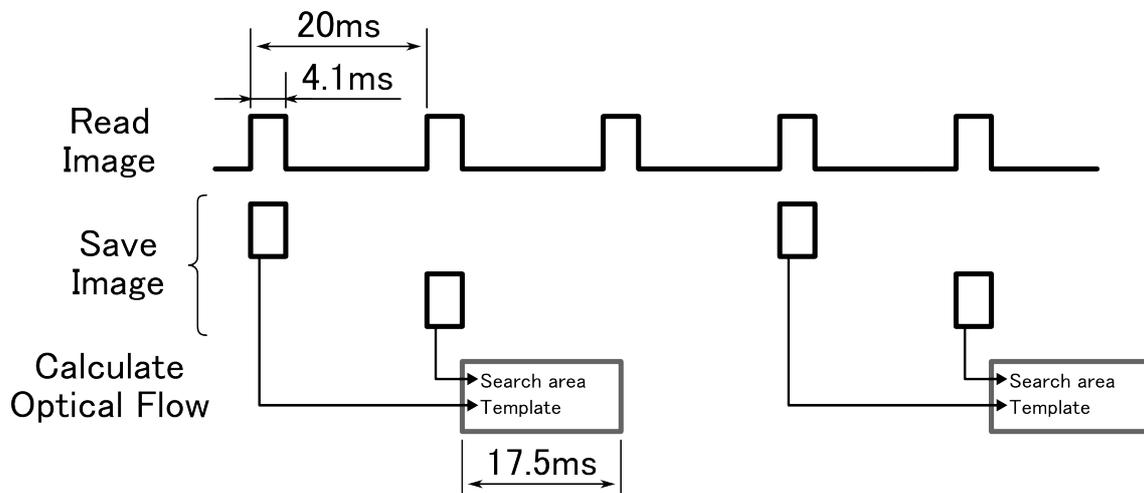


Fig.37 Operation timing

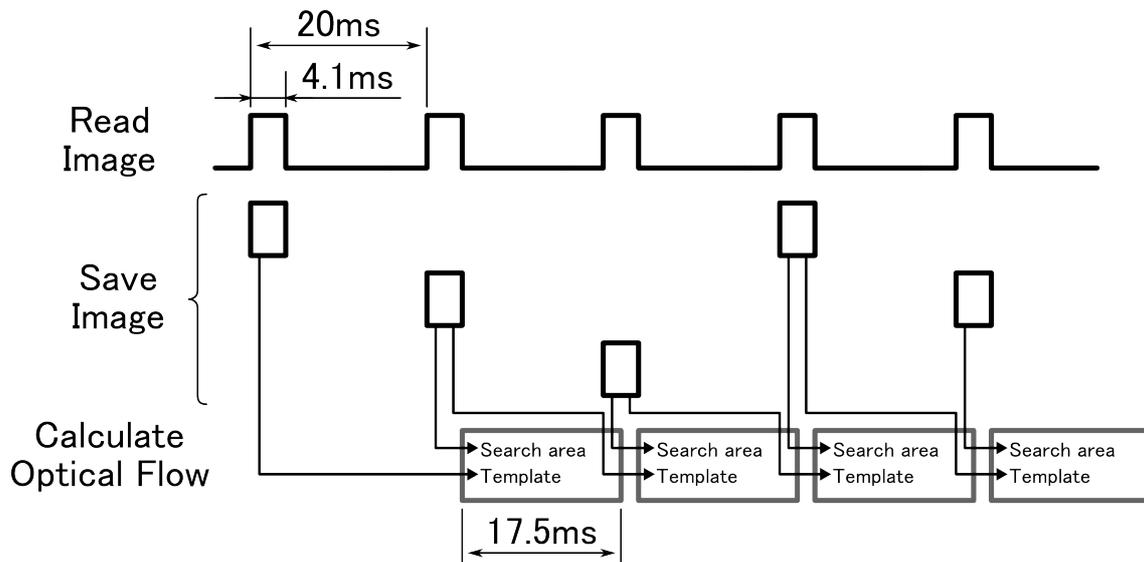


Fig.38 Improved operation timing

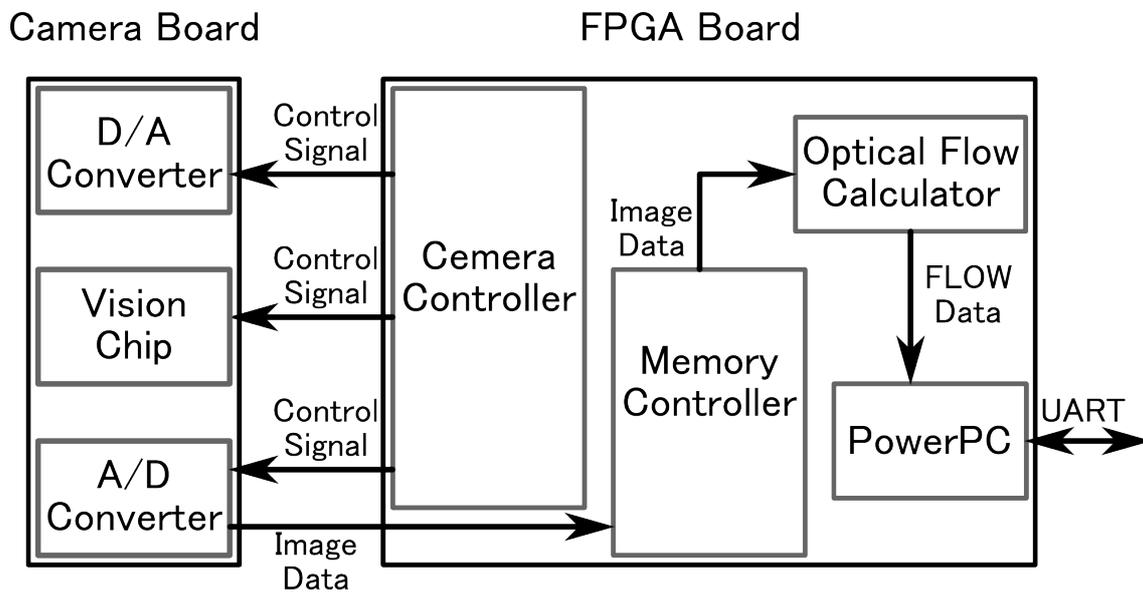


Fig.39 System configuration (UART version)

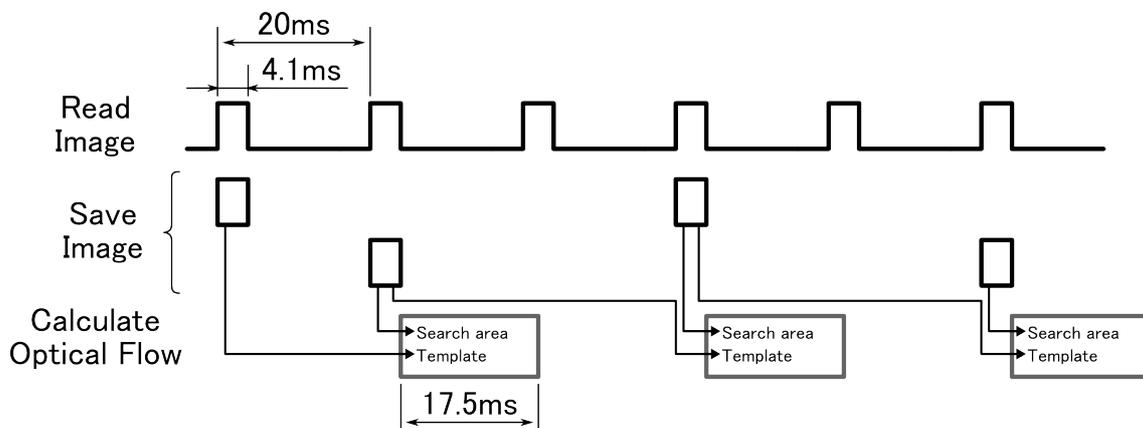


Fig.40 Operation timing of UART version

第3章

視覚システムを用いた エゴモーション推定

第3章 視覚システムを用いたエゴモーション推定

3.1 はじめに

2章において開発したオプティカルフロー演算システムを用いて、移動ロボットの自身の動きであるエゴモーションの推定を行う。

3.2 エゴモーション推定方法

まずは、カメラに映し出された映像と、地面との関係を考える。Fig.41のように原点0にカメラを設置し、このときカメラに映し出される映像と地面との関係を考える。ここでは、中心投影の視点を原点とした視点座標系を考える。Fig.41に示されるように、透視投影面はxy座標面に平行だが、カメラが見ている地面はxy座標とは平行でない。投影面は、カメラの焦点距離fに相当する $z=f$ の位置に置く。地面のz座標は式(20)で与えられるとする。地面の各点を表す座標は地面に平行な座標ではなく、この視点座標系の (x, y) 座標で表す。原点と地面の点とを結ぶ直線を考えて、式(21)に示すように地面の座標 (x, y) は投影面における座標 (x', y') に対応する。

$$z = z_0 x + b_0 y + c_0 \quad (20)$$

$$\begin{cases} x' = \frac{f}{z} x = \frac{f x}{a_0 x + b_0 y + c_0} \\ y' = \frac{f}{z} y = \frac{f y}{a_0 x + b_0 y + c_0} \end{cases} \quad (21)$$

視点座標系の中におかれた地面上の点は別の原点からそこに投影されたと考え、その写像関係を式(22)のアフィン変換を用いて示す、射影変換（ホモグラフィ変換）の一般式が式(23)のように得られる。ただし、焦点距離fは式の係数に含めて考える。

$$\begin{cases} x' = a_1 x + b_1 y + c_1 \\ y' = a_2 x + b_2 y + c_2 \end{cases} \quad (22)$$

$$\begin{cases} x' = \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + c_0} \\ y' = \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + c_0} \end{cases} \quad (23)$$

式(23)で使われる9個の係数 $a_0 \sim c_2$ が得られれば射影変換が決定される。ここで、

式(23)は分子分母を c_0 で割っても一般性が失われないため、式(24)のように未知数は8個になる。

$$\begin{cases} x' = \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + 1} \\ y' = \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + 1} \end{cases} \quad (24)$$

この8個の未知数を得るためには、対応する4点の座標を用いて式(25)の8元連立方程式を解けばよい。

$$\begin{bmatrix} -x_1 x'_1 & -y_1 x'_1 & x_1 & y_1 & 1 & 0 & 0 & 0 \\ -x_1 y'_1 & -y_1 y'_1 & 0 & 0 & 0 & x_1 & y_1 & 1 \\ -x_1 x'_1 & -y_1 x'_1 & x_1 & y_1 & 1 & 0 & 0 & 0 \\ -x_1 y'_1 & -y_1 y'_1 & 0 & 0 & 0 & x_1 & y_1 & 1 \\ -x_1 x'_1 & -y_1 x'_1 & x_1 & y_1 & 1 & 0 & 0 & 0 \\ -x_1 y'_1 & -y_1 y'_1 & 0 & 0 & 0 & x_1 & y_1 & 1 \\ -x_1 x'_1 & -y_1 x'_1 & x_1 & y_1 & 1 & 0 & 0 & 0 \\ -x_1 y'_1 & -y_1 y'_1 & 0 & 0 & 0 & x_1 & y_1 & 1 \end{bmatrix} \begin{pmatrix} a_0 \\ b_0 \\ a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{pmatrix} \quad (25)$$

次に、カメラが地面と水平に移動することを考える。

時刻 t におけるカメラに映し出された点 ${}^s P_t(x, y)$ が Δt 時間後に ${}^s P'_{t+\Delta t}(x+u, y+v)$ に移動したとする。このとき、点の移動距離を実際の移動距離に換算すると、先程のホモグラフィ行列を用いて、それぞれ式(26)、式(27)のように表すことができる。

$$\begin{cases} c x'_t = \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + 1} \\ c y'_t = \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + 1} \end{cases} \quad (26)$$

$$\begin{cases} c x'_{t+\Delta t} = \frac{a_1(x+u) + b_1(y+v) + c_1}{a_0(x+u) + b_0(y+v) + 1} \\ c y'_{t+\Delta t} = \frac{a_2(x+u) + b_2(y+v) + c_2}{a_0(x+u) + b_0(y+v) + 1} \end{cases} \quad (27)$$

地面に対して水平なカメラの移動を考えるので、伸縮率 $\lambda=1$ のヘルマート変換を用いる。ヘルマート変換は、回転、平行移動、拡大縮小を組み合わせた座標変換である。へ

ルマート変換の基本式は式(28)のように表される.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \lambda \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (28)$$

ここで式を次のようにおくと

$$\begin{cases} a = \lambda \cos \theta \\ b = -\lambda \sin \theta \\ c = x_0 \\ d = y_0 \end{cases} \quad (29)$$

$$\begin{cases} x' = ax + by + c \\ y' = -bx + ay + d \end{cases} \quad (30)$$

と表すことができる.

式(30)の4つの未知数 a, b, c, d を求めることでヘルマート変換の式が得られる.

これは, 対応する2点があれば計算することができるが, オプティカルフロー演算システムにより196組の対応点が得られるので, 最小二乗法を用いて計算する.

ここで, 実際に得られた各対応点と, 求める式との誤差を式(31)のようにおく.

$$\begin{cases} Ex_i = ax_i + by_i + c - x'_i \\ Ey_i = -bx_i + ay_i + d - y'_i \end{cases} \quad (31)$$

式(32)に示す, 誤差の二乗和 E が最小になる a, b, c, d を求める.

$$E = \sum_i (Ex_i)^2 + \sum_i (Ey_i)^2 \rightarrow \min. \quad (32)$$

式(32)をそれぞれの未知数 a, b, c, d で偏微分し, それが0になるような a, b, c, d を求める.

$$\frac{\partial E}{\partial a} = (\widehat{x^2} + \widehat{y^2})a + \widehat{x}c + \widehat{y}d - (\widehat{xx'} + \widehat{yy'}) = 0 \quad (33)$$

$$\frac{\partial E}{\partial b} = (\widehat{x^2} + \widehat{y^2})b + \widehat{y}c + \widehat{x}d - (\widehat{xy'} + \widehat{yx'}) = 0 \quad (34)$$

$$\frac{\partial E}{\partial c} = \widehat{x}a + \widehat{y}b + nc - \widehat{x'} = 0 \quad (35)$$

$$\frac{\partial E}{\partial d} = \widehat{y}a + \widehat{x}b + nd - \widehat{y'} = 0 \quad (36)$$

ここで、ハットを付けた変数はそれぞれ次のようにおく。

$$\begin{aligned}\hat{x} &= \sum_i^n x_i & \hat{y} &= \sum_i^n y_i & \hat{x}^2 &= \sum_i^n x_i^2 & \hat{y}^2 &= \sum_i^n y_i^2 & \hat{x}' &= \sum_i^n x'_i & \hat{y}' &= \sum_i^n y'_i \\ \widehat{xx}' &= \sum_i^n x_i x'_i & \widehat{yy}' &= \sum_i^n y_i y'_i & \widehat{xy}' &= \sum_i^n x_i y'_i & \widehat{yx}' &= \sum_i^n y_i x'_i\end{aligned}\quad (37)$$

連立方程式を解くと、 (a, b, c, d) はそれぞれ、

$$a = \frac{\hat{x}\hat{x}' + \hat{y}\hat{y}' - n(\widehat{xx}' + \widehat{yy}')}{\hat{x}^2 + \hat{y}^2 - n(\hat{x}^2 + \hat{y}^2)} \quad (38)$$

$$b = \frac{\hat{x}\hat{y}' - \hat{y}\hat{x}' - n(\widehat{yx}' - \widehat{xy}')}{\hat{x}^2 + \hat{y}^2 - n(\hat{x}^2 + \hat{y}^2)} \quad (39)$$

$$c = \frac{-\hat{x}\hat{y}\hat{y}' - \hat{y}\widehat{yx}' + \hat{y}\widehat{xy}' - \hat{x}\widehat{xx}' + \hat{y}^2\hat{x}' + \hat{x}^2\hat{y}'}{\hat{x}^2 + \hat{y}^2 - n(\hat{x}^2 + \hat{y}^2)} \quad (40)$$

$$d = \frac{-\hat{y}\hat{y}\hat{y}' - \hat{x}\widehat{xy}' + \hat{x}\widehat{yx}' - \hat{y}\widehat{xx}' + \hat{y}^2\hat{y}' + \hat{x}^2\hat{y}'}{\hat{x}^2 + \hat{y}^2 - n(\hat{x}^2 + \hat{y}^2)} \quad (41)$$

となる。

ここで、伸縮率 λ は式(42)のように表すことができるので

$$\lambda = \sqrt{a^2 + b^2} \quad (42)$$

求めるべき伸縮率が1のヘルマート変換式は式(43)となる。

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a/\lambda & b/\lambda & c \\ -b/\lambda & a/\lambda & d \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (43)$$

なお、回転角は式(49)のように表すことができる。

$$\tan \theta = \frac{b}{a} \quad (44)$$

このように、カメラに映し出された映像をホモグラフィ変換を用いて地面上の座標に変換し、地面上の移動前と移動後の点の移動量を求めるためにヘルマート変換行列を求めることによって、オプティカルフローを用いてエゴモーションが推定できる。時間 t と $t+1$ における、カメラに映し出された映像と地面上の座標との相互関係を Fig. 43 に示す。Fig. 43 に示すように、カメラ画像と地面との変換にはホモグラフィ行列を用いる。時

間 t と時間 $t+1$ 間の関係は、オプティカルフローによって得られる。時間 t と時間 $t+1$ における地面上の点の関係はヘルマート変換行列を求めることによって得られる。

3.3 実験

3.3.1 レールを用いた実験

Fig. 44 に実験装置の外観と Fig. 45 にオプティカルフロー演算システムを搭載したスライダを示す。スライダに取り付けられたオプティカルフロー演算システムによって検出されたオプティカルフローはパソコンで記録する。スライダは安定化電源によって任意の速度で動かすことができる。オプティカルフロー演算システムは、Fig. 46 に示すようにスライダに任意の角度で取り付けることができる。スライダの移動量は、スライダを動かすモータのエンコーダ及びレールに取り付けられたメジャーから得る。

スライダのカメラの水平軸の回転角 α はそれぞれ、 0° 、 45° 、 90° として実験を行う。

カメラの仰角 β は、それぞれ 0° （真下方向）、 30° 、 45° として実験を行う。

実験の手順は以下の通りである。

1. カメラをスライダに決められた角度で取り付ける
カメラの注視点の距離を 400mm に保つために、スライダの高さを変更する。
2. Fig. 47 に示すキャリブレーション用画像を用いてホモグラフィ行列を求める
3. 決められた速度でスライダを移動させる
スライダの速度は、モータに印加する電圧により決定される。
4. 得られたオプティカルフローをホモグラフィ行列を用いて実際の距離に変換する
5. 得られた実際の移動量を用いてヘルマート変換行列を求める
6. 1 フレームあたりの移動量が得られるので、積分することで総移動量を求める

Fig. 48 に仰角、水平角共に 0° とし、速度を変化させた時の実験結果を示す。また、Fig. 49 に仰角 0° に固定し、水平角を変更したときの結果を示す。これらをまとめた結果が表 8 である。カメラの水平角度を一定とし、スライダの速度を変更する実験では、総移動距離に対する誤差が最大 4.78% となった。カメラの水平角度を変えた実験では、総移動距離に対する誤差が最大 3.64% となった。

Fig. 50 に仰角 30° 、水平角を 0° とし、速度を変化させた時の実験結果を示す。Fig. 51

に仰角 30° に固定して水平角を変更したときの結果を示す。これらをまとめた結果を表 9 に示す。スライダの速度を変えながらの実験では最大誤差が 4.06%，カメラの水平角度を変えながらの実験では最大誤差が 2.87% となった。

Fig. 52 に仰角 45° ，水平角を 0° とし，速度を変化させた時の実験結果を示す。Fig. 53 に仰角 45° に固定して水平角を変更したときの結果を示す。これらをまとめた結果を表 10 に示す。スライダの速度を変更する実験では，最大誤差が 5.31%，カメラの水平角度を変更する実験では最大誤差が 3.43% となった。

これらの実験全てをまとめた結果を表 11 に示す。誤差が最大となったのは仰角が 45° でスライダ速度が 120mm/s のときで，5.31% の誤差となった。これより，5% 程度の誤差で移動ロボットのエゴモーション推定を行えることが予測できる。

3.3.2 実機に実装しての実験

ルールを用いた実験で，開発した視覚システムを用いることで自己運動を推定することができることを示した。次に，直線運動だけではなく回転運動も含めた自己運動推定を行うために，実機に搭載して自己運動推定実験を行う。Fig. 54 にオプティカルフロー演算システムを搭載した全方位移動ロボット WITH を示す。Fig. 55 にカメラと地面との関係を示す。カメラの設置高さは 400mm とし，カメラから注視点までの水平距離を 350mm とした。実験でのロボットの軌跡を Fig. 56 に示す。まず，①スタート地点から速度 150mm/s で 1m 直進し，②その場で回転速度 0.5rad/s で回転する。そして，③速度 150mm/s で 1m 直進する。これらの動作を繰り返し 1m の正方形を描くように走行させスタート地点に戻る。最後に，スタート時と同じ方向を向いた時点で実験終了とする。このとき，ロボットはジョイスティックを用いて手動で目標軌跡上を動かすものとする。ここで，ロボット自身のデッドレコニングと視覚システムによる自己運動推定を天井に設置したカメラによるモーションキャプチャと比較する。なお，モーションキャプチャによって得られるロボットの軌跡を真値とする。

Fig. 57 にそれぞれの方法で取得したロボットの軌跡を示す。タイヤの路面との滑りによりデッドレコニングが大きく外れた軌跡を推定していることがわかる。Fig. 58 は目標軌道部分を拡大した図である。ここで，視覚システムを用いて推定した軌跡とモーションキャプチャによって得られた軌跡を比較し，総走行区間（約 4m）での平均二乗誤差をとると 39.3mm となる。これは総走行距離に対しての誤差が 0.98% である。

次に Fig. 59 に x 方向，y 方向座標の時間推移を示す。図中の①～③は Fig. 56 の目標軌跡

に対応する。図より、x方向の最大誤差がスタート後14秒の地点で81mmとなり、y方向の最大誤差がスタート後12秒の地点で54mmとなった。総走行距離に対する平均二乗誤差はx方向y方向それぞれ、29.6mm、25.9mmとなり、総走行距離に対して0.74%、0.65%と精度よく自己運動を推定できていることがわかる。Fig.60は回転角の時間変化を示したものであり、図中の①～⑧はFig.56の目標軌跡に対応する。回転角の最大誤差はスタート後47秒の地点で0.24radとなった。総回転角に対する平均二乗誤差は0.097radとなり、総回転角に対して1.54%の誤差となった。回転角に対しても同様に精度よく推定できていることがわかる。

3.4 まとめ

本章では、視覚システムを用いた自己運動推定について述べた。地面における点と、地面に投影したオプティカルフローによって得られた移動先の点との関係を、ヘルマート変換行列を求めることにより、自身の運動を推定した。開発した視覚システムでは196組のオプティカルフローが得られるので、それら全てを用いた最小二乗をを解くことによりヘルマート変換行列を求めた。

レーンを使った実験では、総移動距離に対して最大約5%の誤差で運動を推定できることを確認した。実機に搭載しての実験では、直進運動だけでなく回転運動に対しても高精度な自己運動の推定が行え、1%程度の誤差で自己運動を推定できることを確認した。

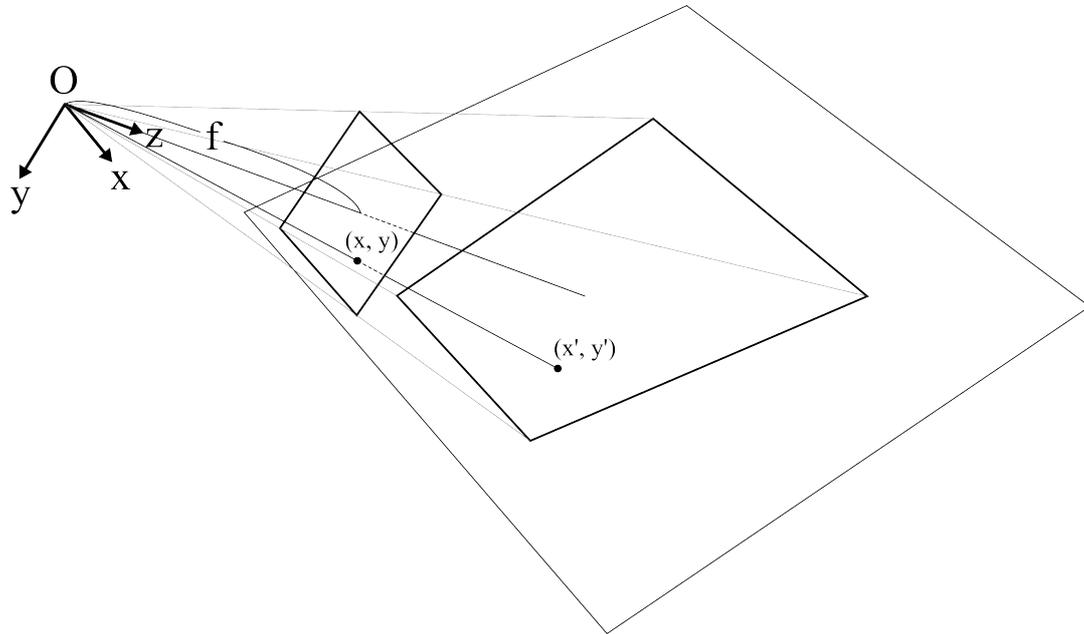


Fig.41 Camera image and ground area

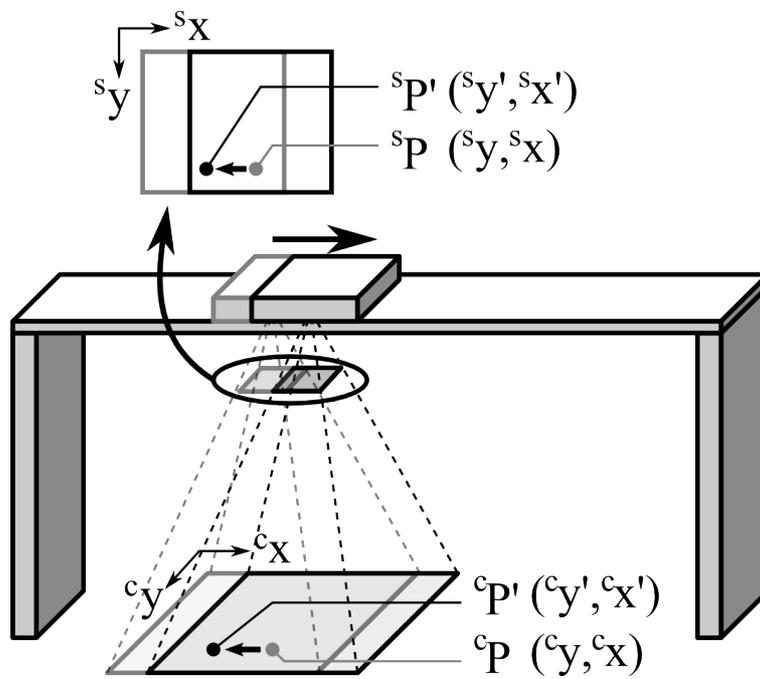


Fig.42 Movement of slicer and corresponding points

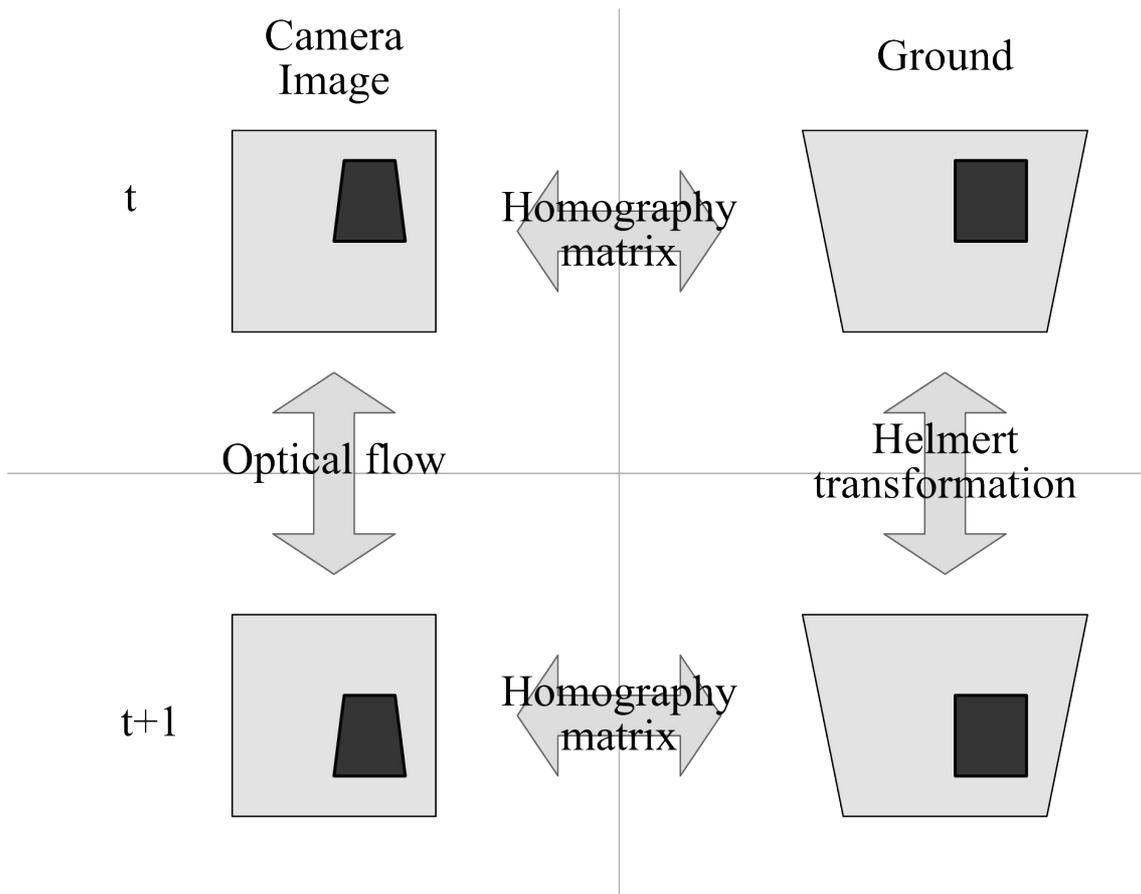


Fig.43 The cross relationship of the coordinate

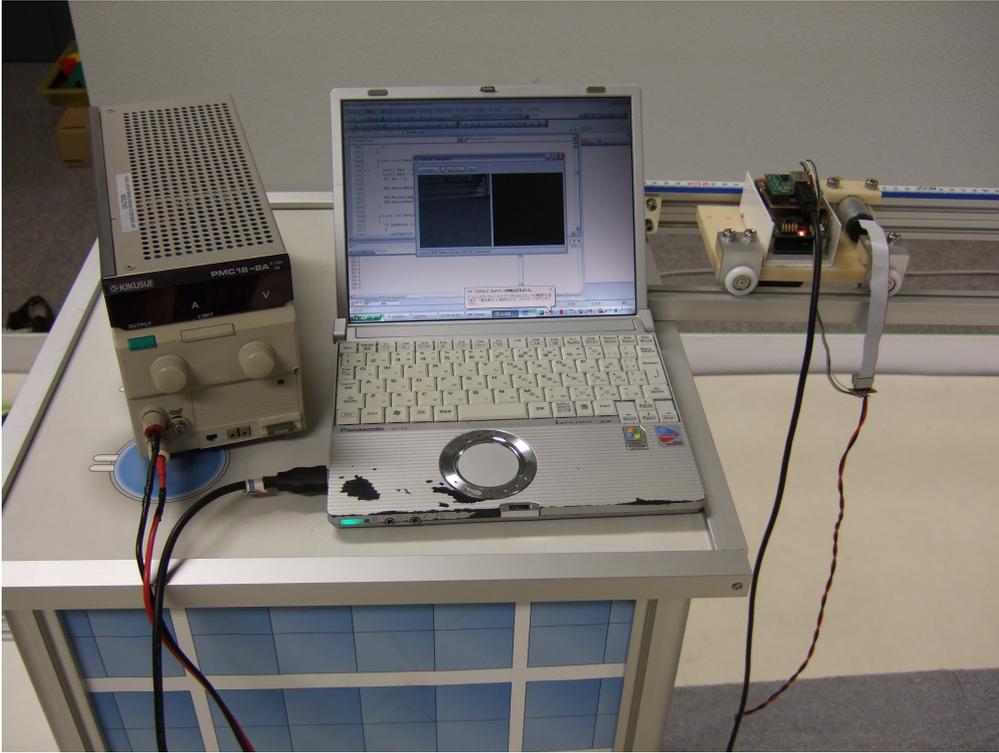


Fig.44 Experimental device

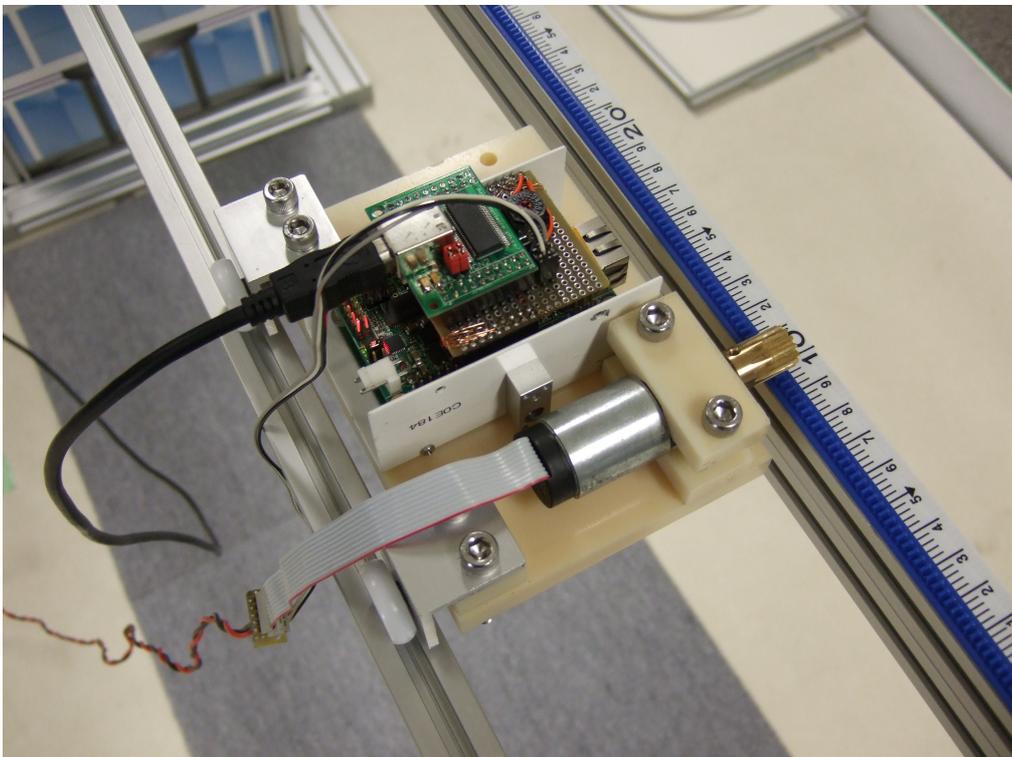


Fig.45 Slider

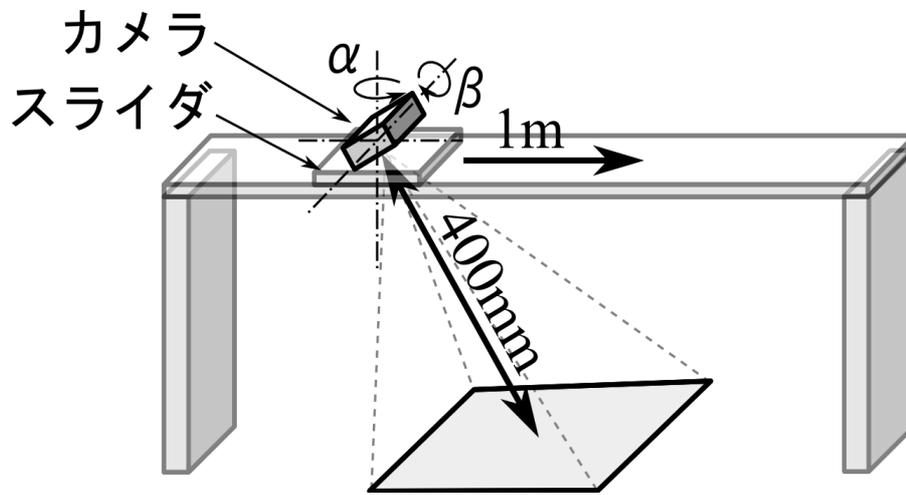


Fig.46 Experimental device

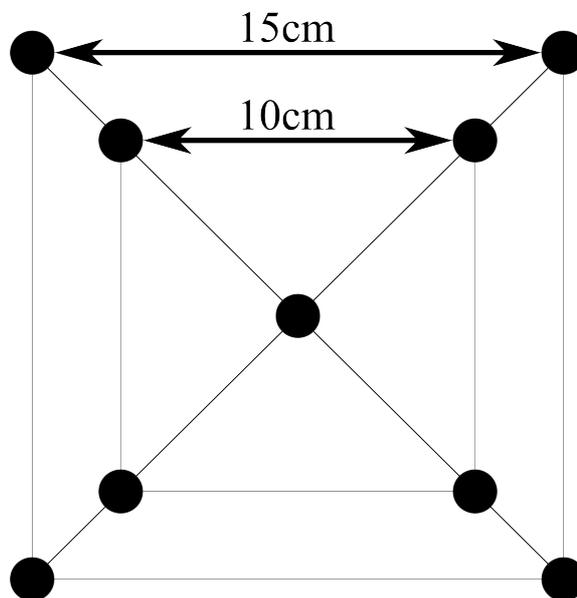


Fig.47 Reference image

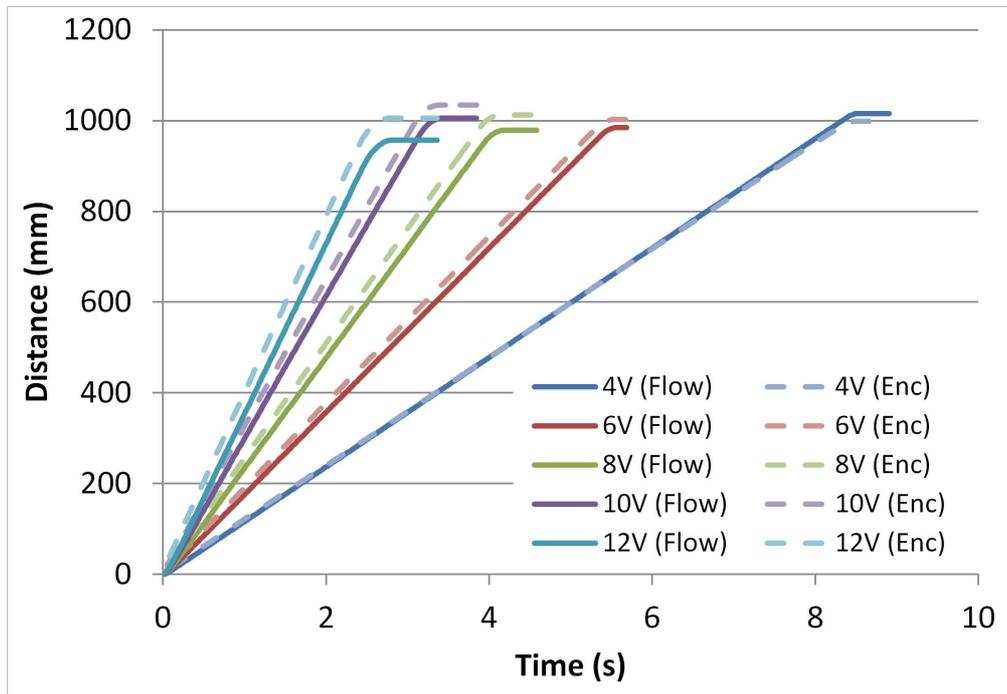


Fig.48 Experimental result of change slicer speed($\beta = 0$)

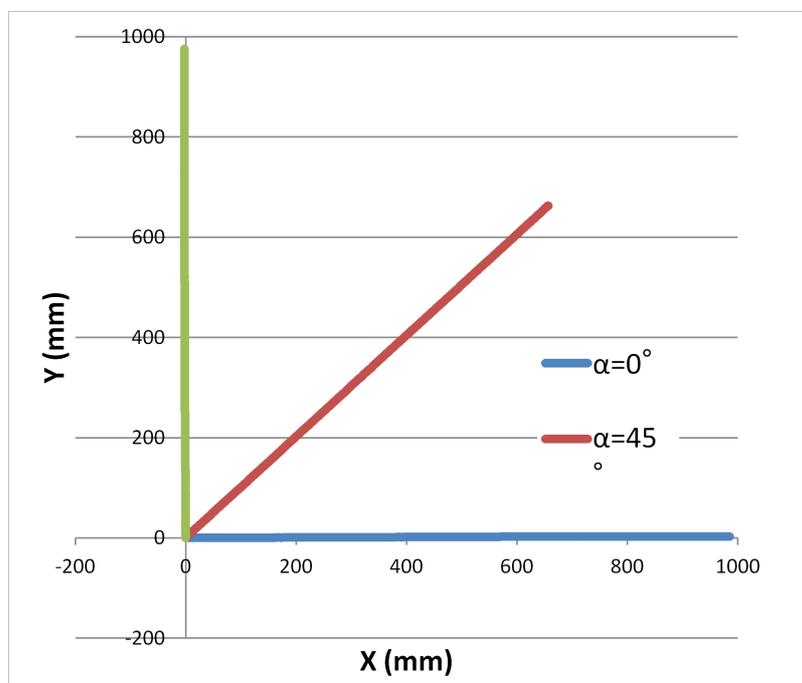


Fig.49 Experimental result of change horizontal angle

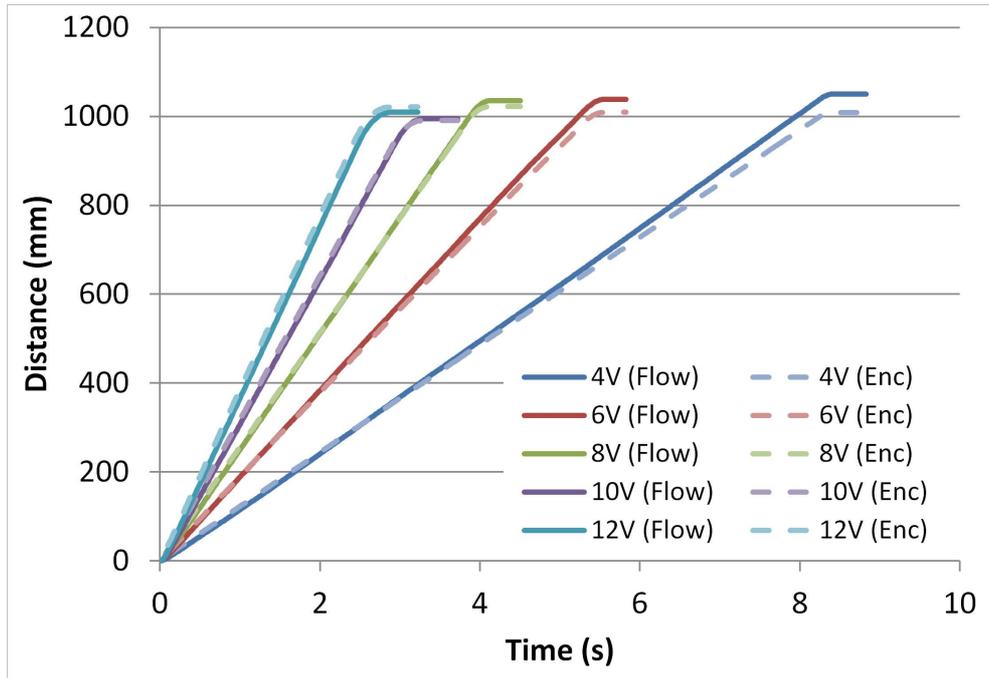


Fig.50 Experimental result of change slicer speed($\beta = 30$)

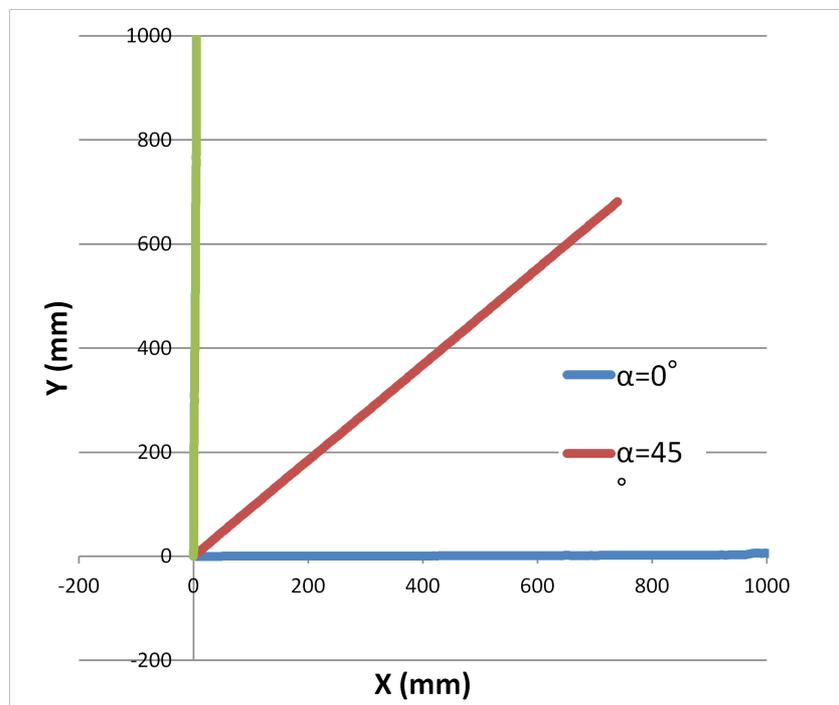


Fig.51 Experimental result of change horizontal angle

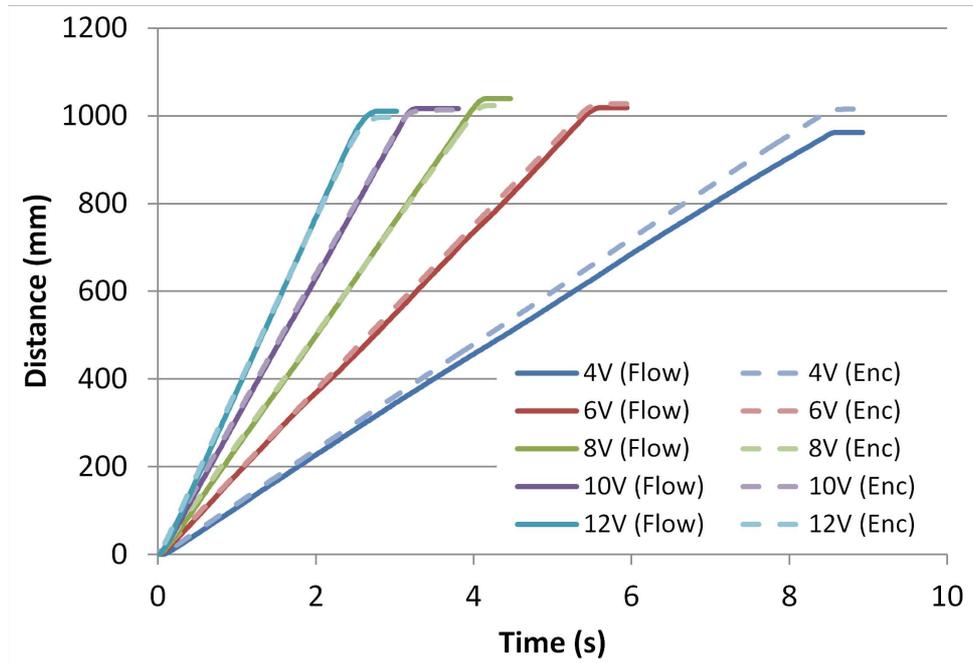


Fig.52 Experimental result of change slicer speed($\beta = 45$)

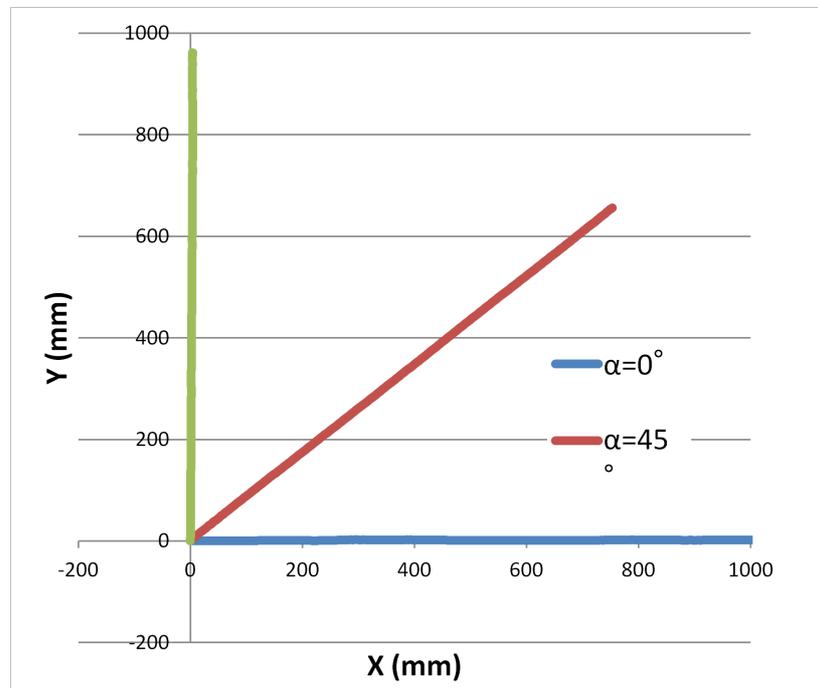


Fig.53 Experimental result of change horizontal angle

Table 8 Travel distance obtained in the moving speed of the camera ($\beta = 0$)

| カメラ水平角度 α (deg) | モータ電圧 (V) | スライダ速度 (mm/s) | 移動距離 (mm) | | 誤差 (mm) | 誤差 (%) |
|---------------------------|--------------|------------------|-----------|-----------|------------|-----------|
| | | | エンコーダ | オプティカルフロー | | |
| 0 | 4 | 120 | 998 | 1015 | 17 | 1.70 |
| 0 | 6 | 188 | 1002 | 985 | -17 | -1.70 |
| 0 | 8 | 253 | 1012 | 979 | -33 | -3.26 |
| 0 | 10 | 320 | 1034 | 1005 | -29 | -2.80 |
| 0 | 12 | 388 | 1005 | 957 | -48 | -4.78 |
| 45 | 6 | 188 | 989 | 953 | -36 | -3.64 |
| 90 | 6 | 188 | 982 | 976 | -6 | -0.61 |

Table 9 Travel distance obtained in the moving speed of the camera ($\beta = 30$)

| カメラ水平角度 α (deg) | モータ電圧 (V) | スライダ速度 (mm/s) | 移動距離 (mm) | | 誤差 (mm) | 誤差 (%) |
|---------------------------|--------------|------------------|-----------|-----------|------------|-----------|
| | | | エンコーダ | オプティカルフロー | | |
| 0 | 4 | 120 | 1009 | 1050 | 41 | 4.06 |
| 0 | 6 | 188 | 1009 | 1038 | 29 | 2.87 |
| 0 | 8 | 253 | 1022 | 1035 | 13 | 1.27 |
| 0 | 10 | 320 | 991 | 995 | 4 | 0.40 |
| 0 | 12 | 388 | 1021 | 1009 | -12 | -1.18 |
| 45 | 6 | 188 | 1003 | 1005 | 2 | 0.20 |
| 90 | 6 | 188 | 1015 | 1025 | 10 | 0.99 |

Table 10 Travel distance obtained in the moving speed of the camera ($\beta = 45$)

| カメラ水平角度 α (deg) | モータ電圧 (V) | スライダ速度 (mm/s) | 移動距離 (mm) | | 誤差 (mm) | 誤差 (%) |
|---------------------------|--------------|------------------|-----------|-----------|------------|-----------|
| | | | エンコーダ | オプティカルフロー | | |
| 0 | 4 | 120 | 1016 | 962 | -54 | -5.31 |
| 0 | 6 | 188 | 1027 | 1018 | -9 | -0.88 |
| 0 | 8 | 253 | 1023 | 1040 | 17 | 1.66 |
| 0 | 10 | 320 | 1013 | 1017 | 4 | 0.39 |
| 0 | 12 | 388 | 996 | 1011 | 15 | 1.51 |
| 45 | 6 | 188 | 1017 | 999 | -18 | -1.77 |
| 90 | 6 | 188 | 962 | 995 | 33 | 3.43 |

Table 11 Error for each installation angle of the camera

| モータ電圧 (V) | カメラ水平角度 α (deg) | 誤差 (%) | | |
|--------------|------------------------------|-------------|---------------|---------------|
| | | $\beta = 0$ | $\beta = -30$ | $\beta = -45$ |
| 4 | 0 | 1.70 | 4.06 | -5.31 |
| 6 | 0 | -1.70 | 2.87 | -0.88 |
| 8 | 0 | -3.26 | 1.27 | 1.66 |
| 10 | 0 | -2.80 | 0.40 | 0.39 |
| 12 | 0 | -4.78 | -1.18 | 1.51 |
| 6 | 45 | -3.64 | 0.20 | -1.77 |
| 6 | 90 | -0.61 | 0.99 | 3.43 |

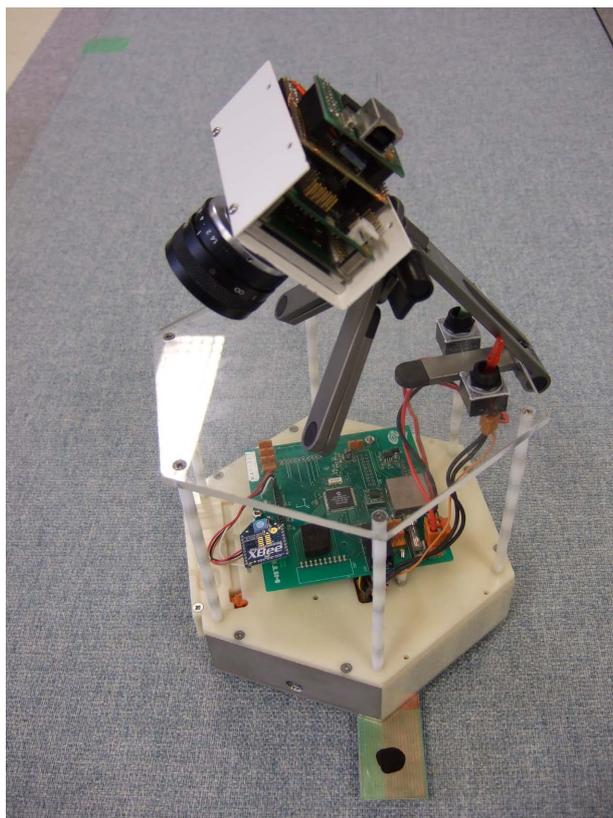


Fig.54 Vision system on the WITH

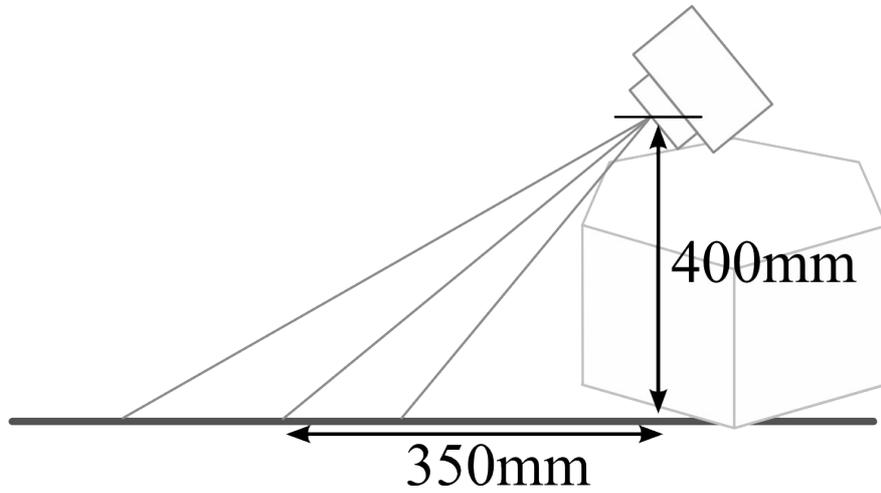


Fig.55 Geometry of the camera installation

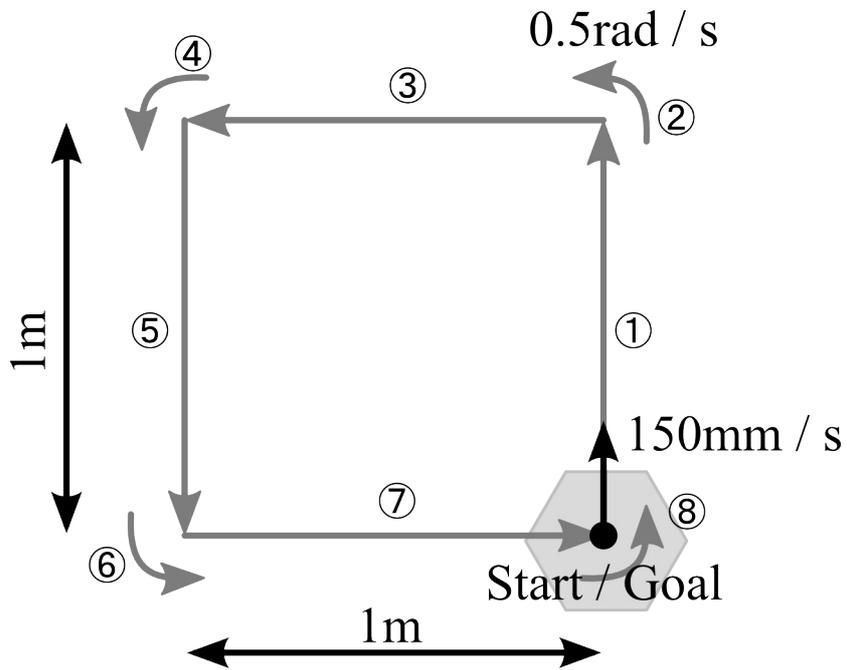


Fig.56 Movement of the robot

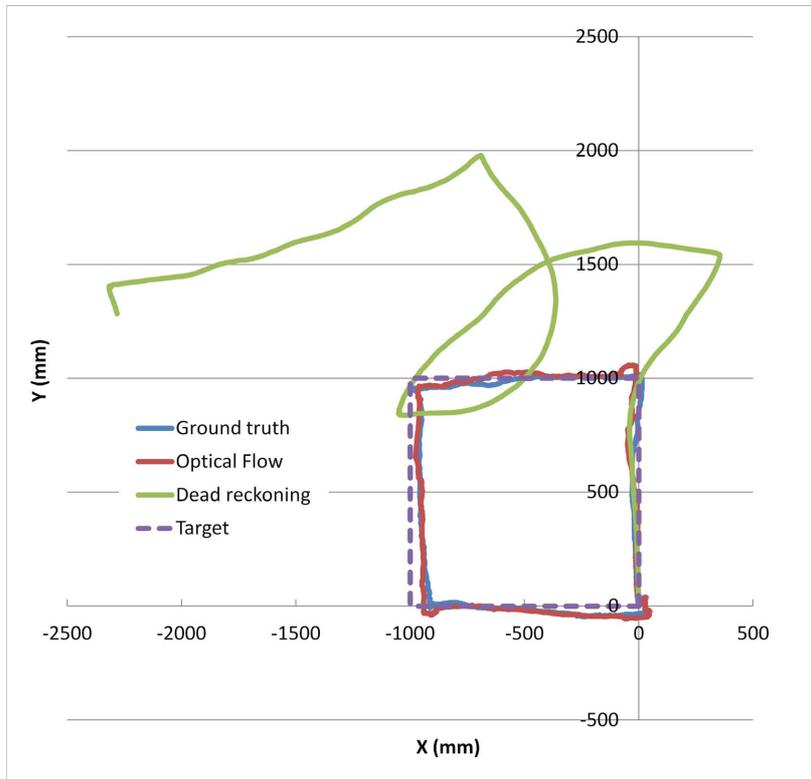


Fig.57 Trace of the robot

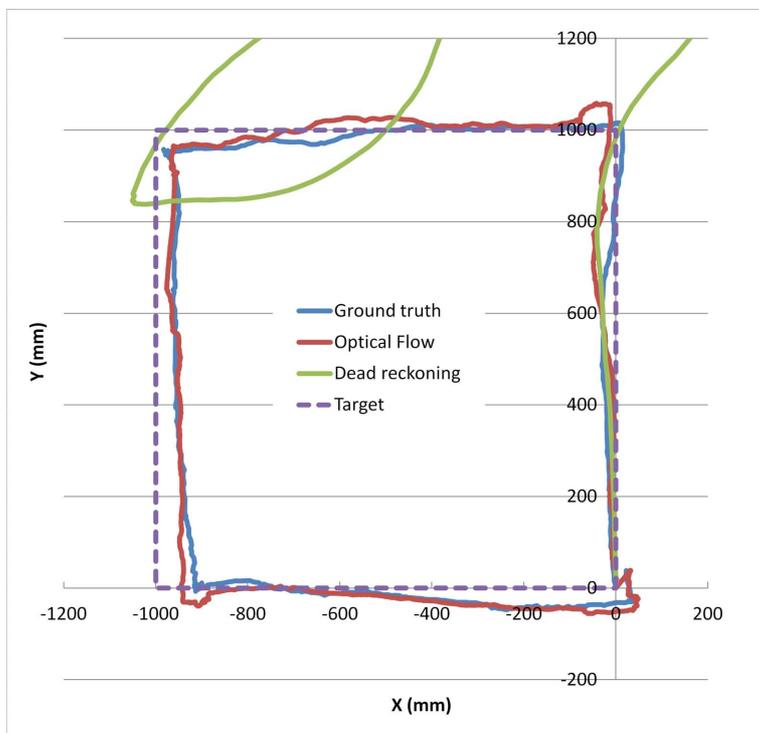


Fig.58 Estimated path and ground truth

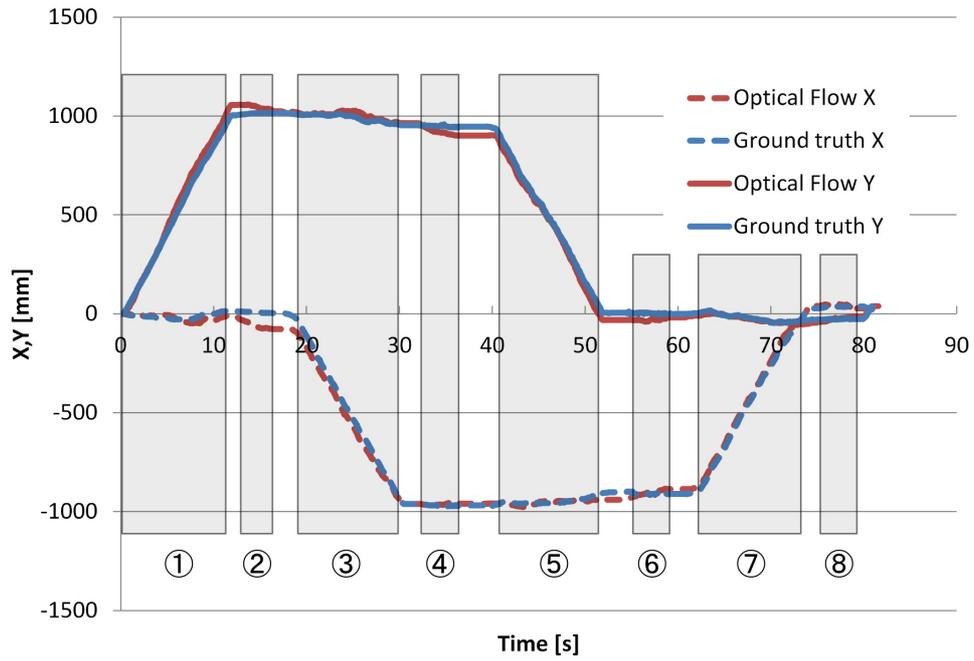


Fig.59 Transition of the xy coordinate

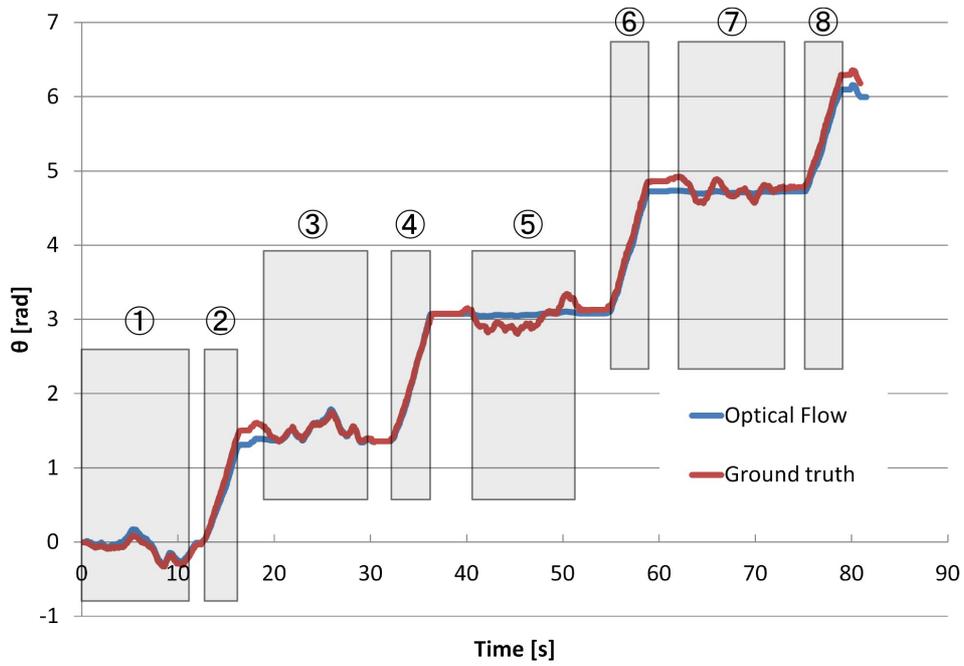


Fig.60 Transition of the rotatino angle

第4章

視覚システムを用いた ドミナントプレーンの検出

第4章 視覚システムを用いたドミナントプレーンの検出

4.1 はじめに

ドミナントプレーンとは Fig. 61 に示すように画像の最大領域を占める平面である。このドミナントプレーンを検出することで、ロボットが移動可能な領域と障害物が存在する領域を切り分けることができる。

ドミナントプレーンの検出に関しては、Enkelmann や Santos らの研究[72, 73]が挙げられる。これらは、Fig. 62 に示すように障害物が無い平面のオプティカルフローを事前に保存しておき、障害物などが存在する環境でオプティカルフローを検出し、保存しているオプティカルフローと比較して異なる部分を障害物がある領域とする方法である。この方法は、前章で行ったエゴモーション推定のようにカメラのキャリブレーションを必要としないという利点があるが、事前に障害物が無い平面でのオプティカルフローを記録する必要があり、実質キャリブレーションに近い。

大西らはこれを改良し[74-76]、障害物が無い平面でのオプティカルフローを必要としない方法を提案した。これは、Fig. 63 に示すように3枚の時間的に連続した画像を用いて、一枚目と二枚目の画像を用いて一時的なオプティカルフローの分布を計算し、二枚目と三枚目の画像を用いて計算したオプティカルフローと比較する方法である。この方法では、一枚目の画像に障害物が含まれていると、ドミナントプレーンの検出が難しいといった問題がある。

本章では、障害物が無い平面でのオプティカルフローを必要とせず、前章で述べたオプティカルフローを用いたエゴモーション推定と同時に行うためのドミナントプレーンの検出について述べる。

4.2 ドミナントプレーン検出方法

Fig. 64 に示すように、カメラを地面に向けて設置し、地面と水平に l_c 移動したとする。このとき、カメラを基準にすると障害物、地面共に l_o 、 l_g だけ動いたことになる。それぞれの移動量は等しく、 $l_c=l_o=l_g$ である。

このとき、検出される障害物のオプティカルフロー f_o と地面のオプティカルフロー f_g の大きさは等しくならず、障害物のオプティカルフローの方が大きい $f_o > f_g$ という関係になる。逆に、段差があり地面より低い部分があると、地面のオプティカルフ

ローの方が大きくなる．このことから，オプティカルフローを用いることで，地面と障害物を分離することができる．しかし，カメラの移動量 l_c が未知の環境では，検出されたオプティカルフローのどちらが地面で，どちらが障害物なのか判断できない．そこで，先に述べたような平面のオプティカルフローをあらかじめ計測する方法が行われた．

本研究で提案するのは，前章のヘルマート変換行列を用いることで，平面のオプティカルフローを必要としない方法である．

路面と高さ h_o の障害物で計測されるオプティカルフローの関係は式(45)のように近似して表すことができる．

$$\frac{F_o}{F_g} = \frac{l_g}{l_o} = \frac{l_g}{l_g - h_o \cos \alpha} \quad (45)$$

l_c : カメラの移動距離[m]

F_g : 地面で検出されるフローの大きさ[*pixel*]

F_o : 物体で検出されるフローの大きさ[*pixel*]

l_g : カメラから地面までの距離[m]

l_o : カメラから物体までの距離[m]

h_o : 物体の高さ[m]

h_c : 地面からカメラまでの高さ[m]

α : カメラの仰角[*rad*]

f : カメラの焦点距離[m]

ここで，障害物（四角で囲った領域）の存在する路面において Fig.65 のようなオプティカルフローが得られたとする．このとき，障害物の存在する領域のオプティカルフローは地面のオプティカルフローより大きい．このオプティカルフローを路面に投影すると Fig.66 における l_{Fi} のように障害物が存在する領域が大きく移動したかのように検出される．この移動量と前章で求めたヘルマート変換行列によって求めた移動量 l_{Hi} を比較することによって物体の存在する領域を推定する．

このとき，式(46)に示す地面の移動量と障害物によって増加した移動量の比を移動量増加比として定める．そして，移動量増加比がしきい値 ε 以下の領域を路面と推定する．このしきい値は以下のようにして決定した．

$$e_i = \frac{|l_{Fi} - l_{Hi}|}{l_{Hi}} \leq \varepsilon \quad (46)$$

e_i : 移動量増加比

l_{Fi} : オプティカルフローとして検出された移動量[m]

l_{Hi} : ヘルマート変換行列から算出した移動量[m]

ε : しきい値

i : フローの番号

移動量増加比はオプティカルフローによって得られた路面の移動量とヘルマート変換行列を求めることによって得られたカメラの移動量を比較し、正規化することで表す。このとき、ドミナントプレーンである領域を決定するためのしきい値は平面の障害物がない路面のオプティカルフローを基準として求めた。

Fig. 67 は障害物のない路面での移動量増加比とその分散である。これにより、分散を加味すると移動量増加比は障害物のない路面では最大約 0.2 程度の値となることがわかる。そこで、ドミナントプレーンを決定するしきい値は 0.2 とした。

この手法を用いてロボットが走行可能な路面及び障害物を認識することは、

- ・従来の方法と比較して、平常時のオプティカルフローを必要としない
- ・エゴモーション推定で算出したヘルマート変換行列を用いるので、ドミナントプレーンの検出を同時に行っても計算コストの増加が少ない
- ・ステレオカメラを用いる手法と比較して、キャリブレーションが容易

などといった利点がある。

また、エゴモーション推定を行ったときに算出したヘルマート変換行列は、画面全体のオプティカルフローを用いて求めたが、ロボット直前の領域に障害物がないと仮定して、ロボット直前の領域だけのオプティカルフローを用いてヘルマート変換行列を求めることもできる。このようにして算出したヘルマート変換行列を用いることで、障害物が画面の大半を占めるような場合でも、障害物を障害物として認識することができる。

4.3 ドミナントプレーン検出実験

前述した手法を用いてドミナントプレーンを検出する。開発した視覚システムを Fig. 68 ~ Fig. 70 のようにスライダに設置し、レール上をスライダを移動させる。移動速度は 150mm/s とし、カメラがスタート位置にあるときの注視点の前方 700mm に障害物を設置する。障害物は壁と箱の 2 種類使用し、箱は画面中央に映し出される位置と、画面右半分に映し出される位置に置く。このとき、評価の基準として、検出率 P_d と正解率 P_c を定める。検出率と正解率はそれぞれ次のように計算する。

$$P_d = \frac{A_d}{A_o} \quad (47)$$

$$P_c = \frac{(A - A_o) - A_e + A_d}{A} \quad (48)$$

P_d : 検出率

P_c : 正解率

A : 全領域数

A_o : 物体が存在する領域数

A_d : 物体が存在する領域内において、物体を検出した領域数

A_e : 物体が存在しない領域内において、物体を検出した領域数

実験結果の一例を示す。Fig. 71 はスタート位置でのカメラ位置と注視点及び視野範囲である。Fig. 72 はこのときの移動量増加比である。カメラが静止しているのでオプティカルフローは検出されず移動量増加比も全域にわたり検出されない。

次に、Fig. 73 はスタート位置から 500mm 移動したときのカメラ位置と注視点及び視野範囲である。Fig. 74 はこのときのカメラ画像で、障害物はまだ視野範囲に入っておらず平らな路面のオプティカルフローのみが検出される。このときの推定した移動量とエンコーダによる正しい移動量は Fig. 75 に示す。Fig. 76 はこのときの移動量増加比である。ノイズにより一部大きな値を示しているが多くの領域はしきい値以下の値である。

さらにカメラが移動して、Fig. 77 はスタート位置から 640mm 移動したときの注視点及び視野範囲である。このときのカメラ画像が Fig. 78 で、視野の上方から 1/3 の所まで障害物が迫ってきている。このときの推定した移動量は Fig. 79 に示す。このときの移動量増加比は Fig. 80 であり、奥側の移動量増加比が大きくなっている。表 13 に移動量増加比の値を示す。表 13 における四角で囲んだ領域は障害物が存在する領域である。このときの検出率は 0.64、正解率は 0.93 である。

カメラが700mm進むとFig.81のように画面1/2の位置まで障害物が近づく。このときのカメラ画像をFig.82に示す。このときの移動量の推定値とエンコーダによる移動量の計測値はFig.83である。移動量増加比はFig.84のようにさらに大きな値となっている。表14に移動量増加比を示す。表14における四角で囲んだ領域に障害物が存在している。このときの検出率は0.59で、正解率は0.90である。

これらの実験を各障害物につき5回行い平均値をまとめたものが表15である。表より物体に近づくにつれ検出率が低下していることがわかる。これは、画面全体のフローを用いて自身の移動量を計算しているためで、カメラ画像における障害物の占める割合が増えると障害物で検出される増加した移動量の影響が大きくなり、自身の移動量が大きく検出されるためである。

そこで、Fig.85に示すようなロボット付近の領域のフローを用いてヘルマート変換行列を求め、同様にドミナントプレーンを推定すると、表16に示す結果が得られる。表15と表16を比較すると検出率及び正解率が向上していることがわかる。これは自身の動きを推定するために障害物の存在しない領域を用いたためである。このように自身の動きを推定する領域を限定することで、その領域と同じ平面を検出することが可能である。

4.4 まとめ

本章では画面全体のオプティカルフローを用いて計算したヘルマート変換行列を用いたドミナントプレーンの検出について述べた。そして、オプティカルフローとヘルマート変換行列を用いることで容易にドミナントプレーンを検出できることを示した。この手法を用いることで前章で述べたエゴモーションの推定から少量の計算を行うだけで障害物の検出を行える。また、任意の領域のオプティカルフローを用いてヘルマート変換行列を計算し、ドミナントプレーンの検出に用いることで、任意に指定した領域と同じ平面を検出できる。これにより、画面の大域を占める平面であるドミナントプレーンだけでなく、小さな領域の検出も可能である。

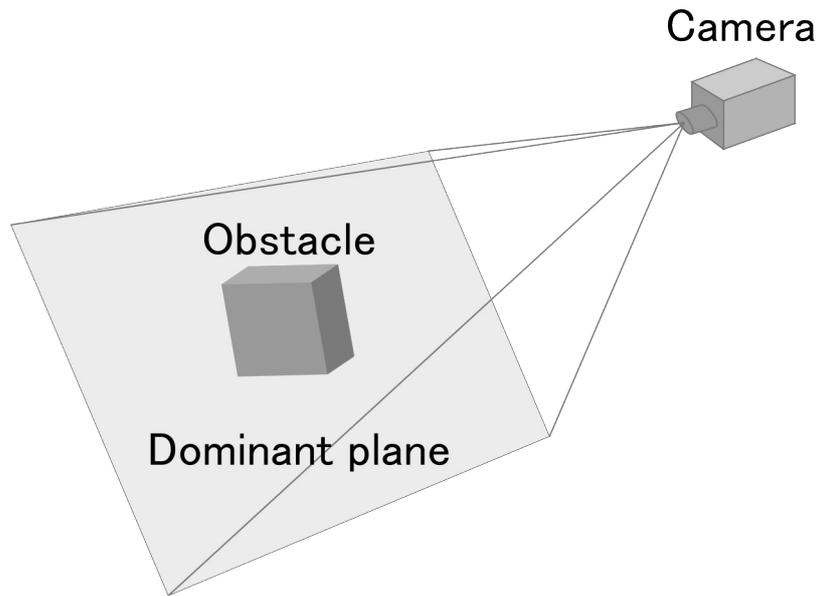


Fig.61 Dominant plane

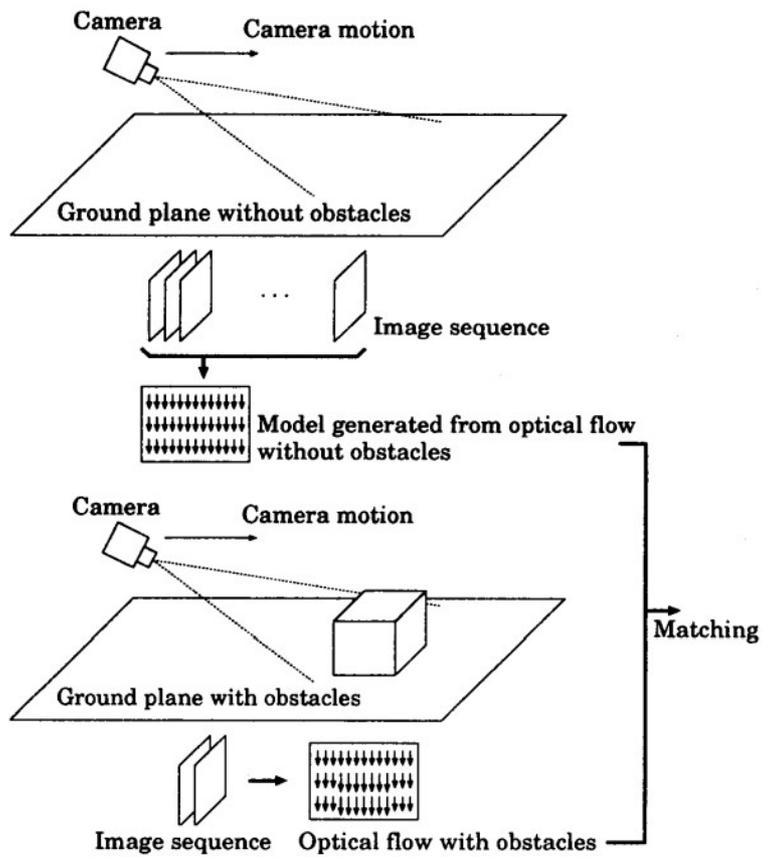


Fig.62 Enkelmann's Dominant plane detection method[72]

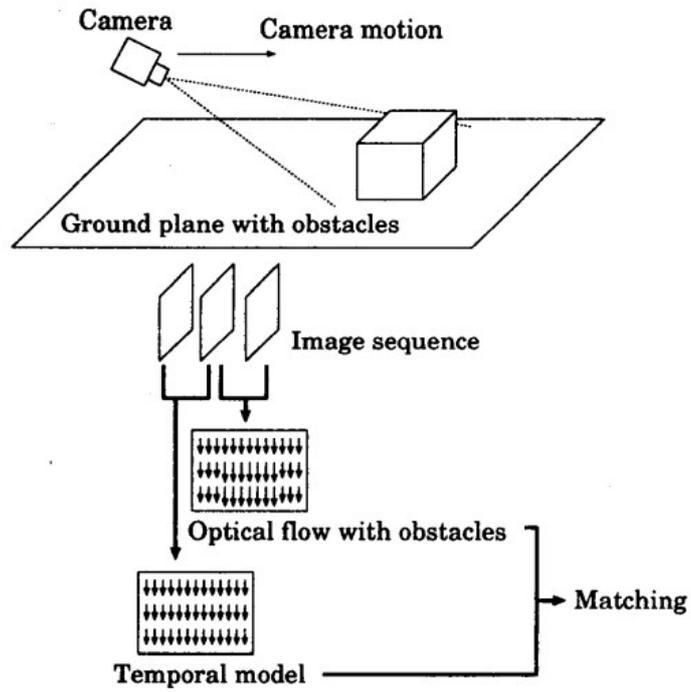


Fig.63 Ohnishi's Dominant plane detection method[74]

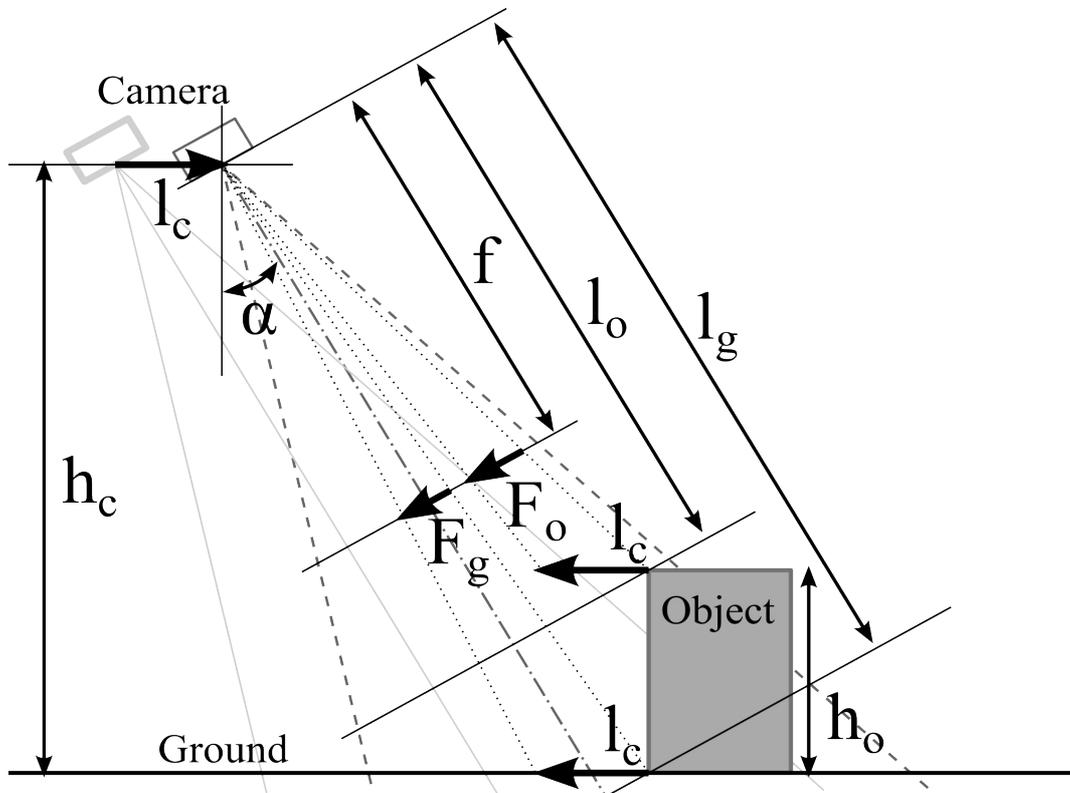


Fig.64 Ground's Flow and Obstacle's Flow

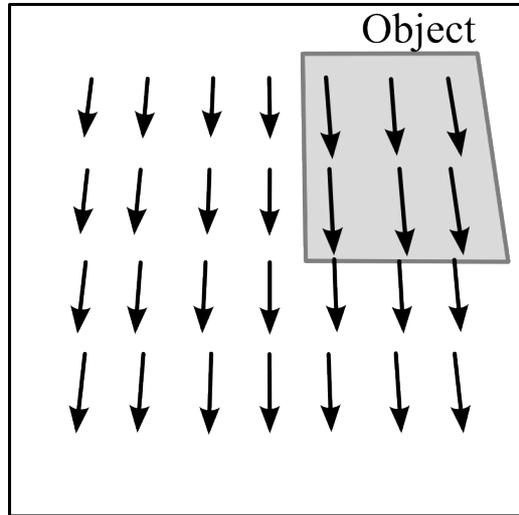


Fig.65 Detected flow

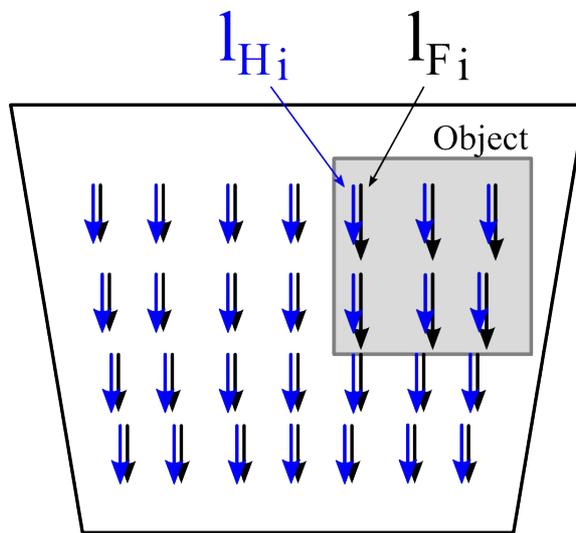


Fig.66 Projected flow

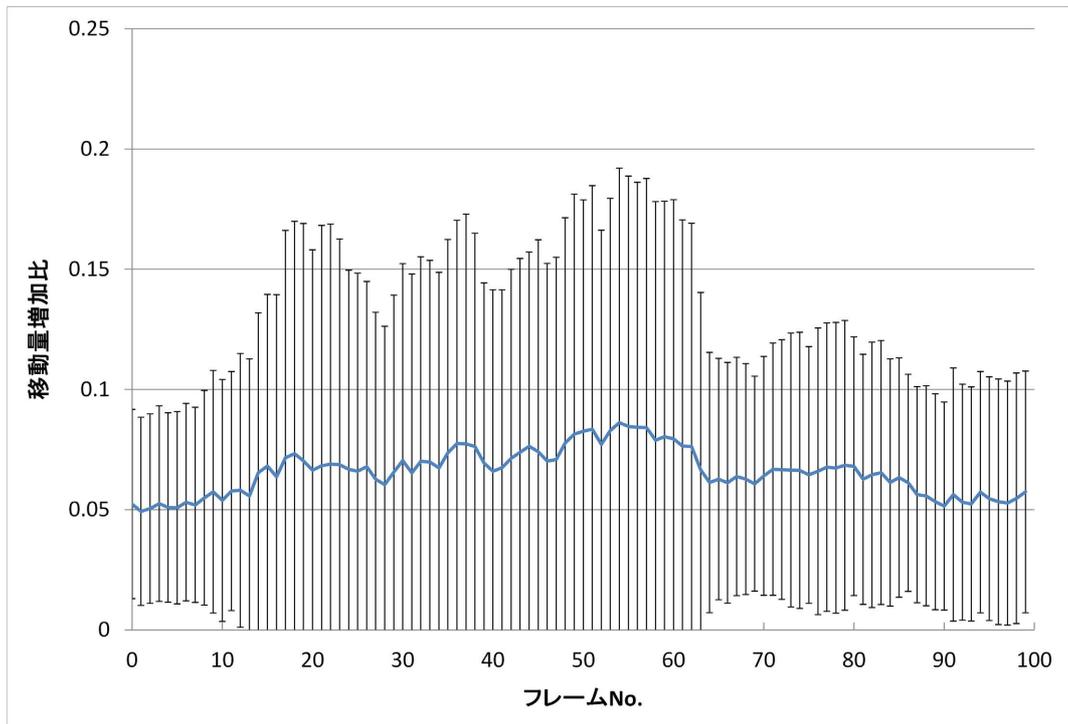


Fig.67 Movement increase ratio

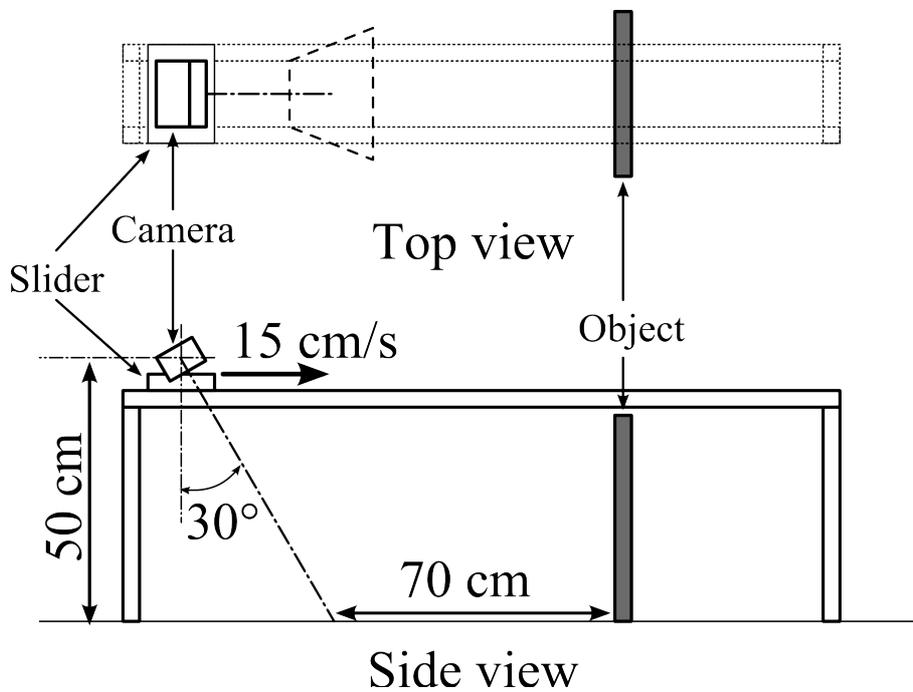


Fig.68 Experiment environment (wall)

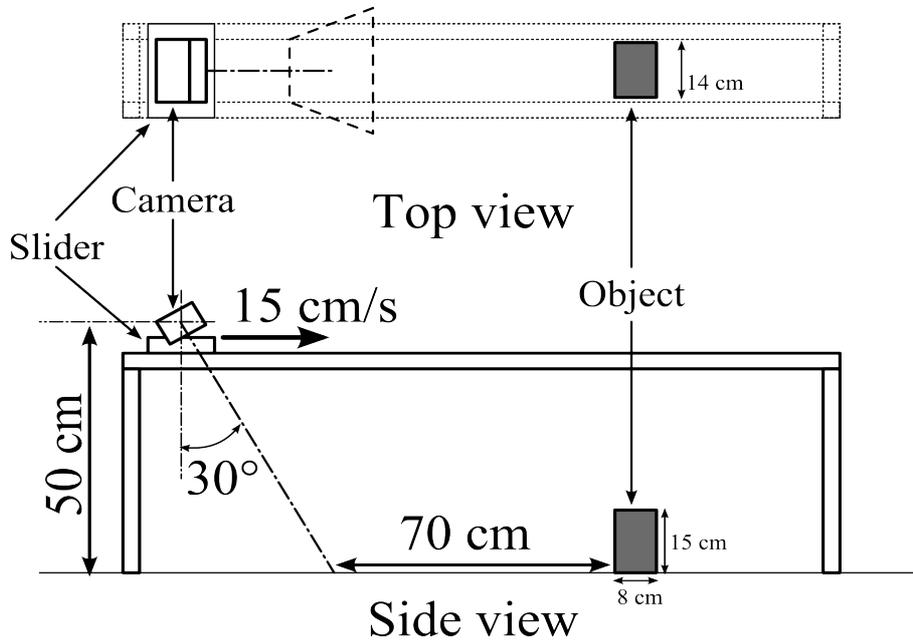


Fig.69 Experiment environment (box)

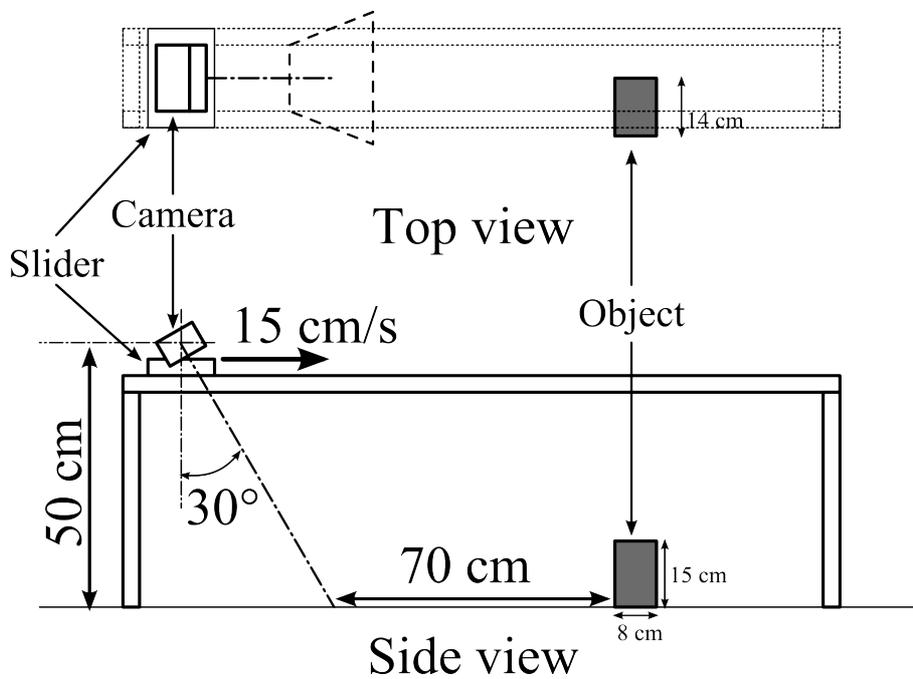


Fig.70 Experiment environment (box offset)

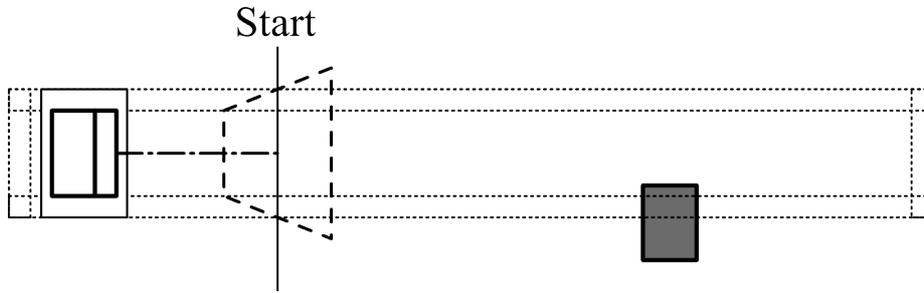


Fig.71 Camera position (distance = 0mm)

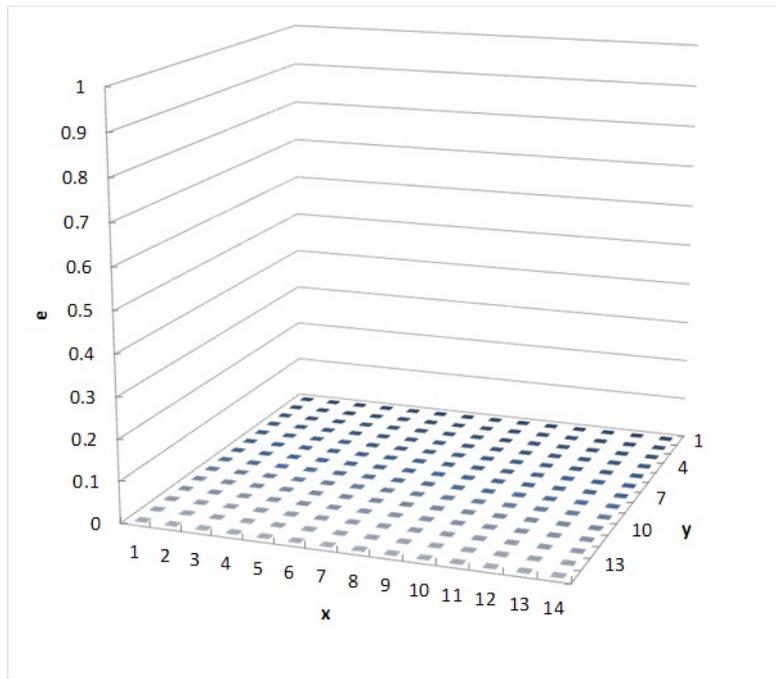


Fig.72 Movement increase ratio (distance = 0mm)

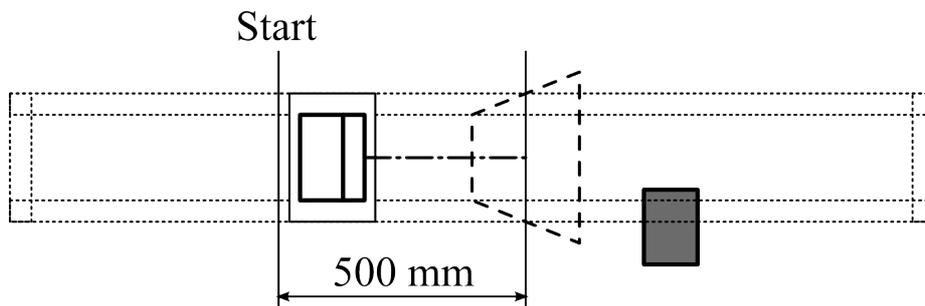


Fig.73 Camera position (distance = 500mm)

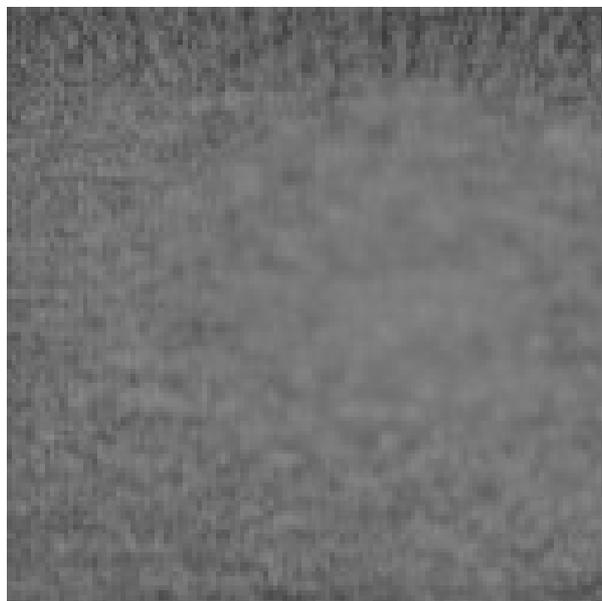


Fig.74 Camera image(distance = 500mm)

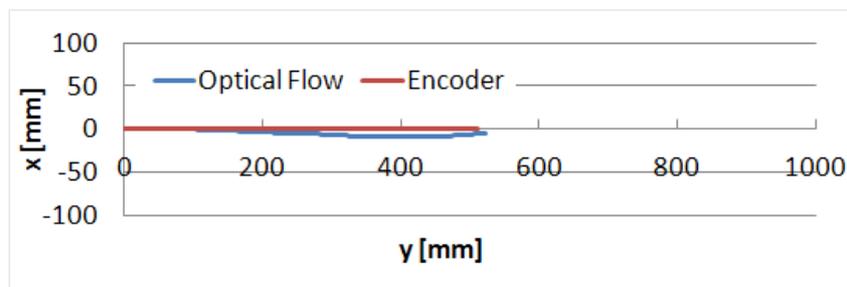


Fig.75 Estimated path (distance = 500mm)

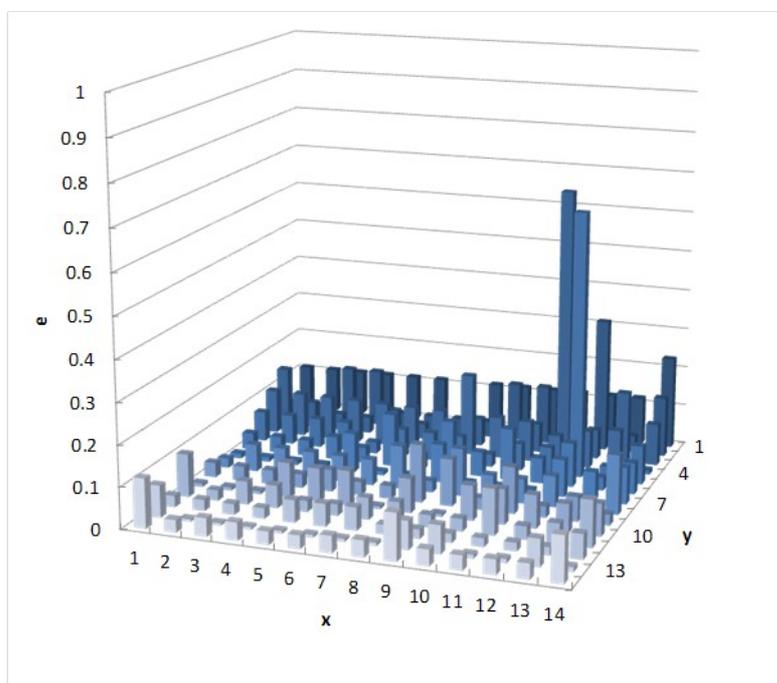


Fig.76 Movement increase ratio (distance = 500mm)

Table 12 Movement increase ratio (distance = 500mm)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0 | 0.16 | 0.16 | 0.16 | 0.16 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.15 | 0.25 |
| 1 | 0.09 | 0.09 | 0.18 | 0.18 | 0.08 | 0.08 | 0.08 | 0.08 | 0.18 | 0.18 | 0.08 | 0.37 | 0.08 | 0.18 |
| 2 | 0.1 | 0.11 | 0.04 | 0.04 | 0.11 | 0.04 | 0.04 | 0.11 | 0.04 | 0.11 | 0.67 | 0.11 | 0.2 | 0.15 |
| 3 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.16 | 0.07 | 0.14 | 0.12 | 0.05 | 0.13 | 0.05 |
| 4 | 0.04 | 0.04 | 0.04 | 0.04 | 0.08 | 0.04 | 0.04 | 0.04 | 0.07 | 0.05 | 0.05 | 0.68 | 0.07 | 0.12 |
| 5 | 0.03 | 0.03 | 0.03 | 0.1 | 0.03 | 0.09 | 0.03 | 0.09 | 0.1 | 0.09 | 0.1 | 0.1 | 0.03 | 0.11 |
| 6 | 0.05 | 0.05 | 0.05 | 0.05 | 0.06 | 0.05 | 0.05 | 0.06 | 0.05 | 0.05 | 0.05 | 0.06 | 0.11 | 0.07 |
| 7 | 0.07 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.05 | 0.07 | 0.15 | 0.04 | 0.04 | 0.12 | 0.07 | 0.12 |
| 8 | 0.03 | 0.03 | 0.03 | 0.03 | 0.1 | 0.03 | 0.03 | 0.18 | 0.08 | 0.03 | 0.08 | 0.03 | 0.09 | 0.03 |
| 9 | 0.05 | 0.05 | 0.05 | 0.05 | 0.12 | 0.05 | 0.05 | 0.05 | 0.05 | 0.12 | 0.12 | 0.05 | 0.11 | 0.06 |
| 10 | 0.15 | 0.07 | 0.03 | 0.03 | 0.07 | 0.07 | 0.03 | 0.03 | 0.07 | 0.07 | 0.14 | 0.07 | 0.03 | 0.14 |
| 11 | 0.03 | 0.03 | 0.03 | 0.03 | 0.1 | 0.1 | 0.1 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.09 |
| 12 | 0.12 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.04 | 0.04 | 0.05 | 0.05 | 0.04 | 0.05 |
| 13 | 0.07 | 0.08 | 0.03 | 0.03 | 0.08 | 0.08 | 0.03 | 0.03 | 0.08 | 0.03 | 0.07 | 0.07 | 0.03 | 0.14 |

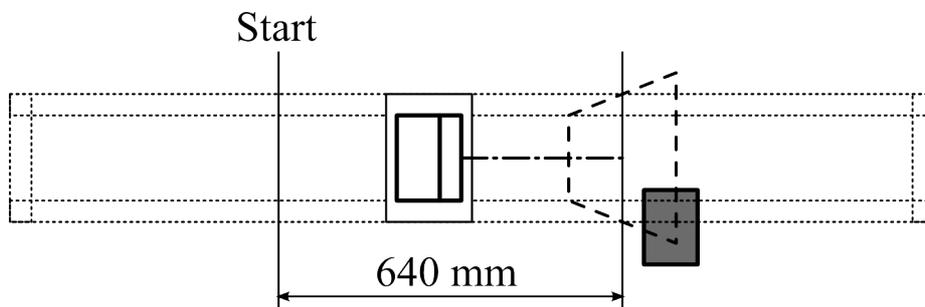


Fig.77 Camera position (distance = 640mm)

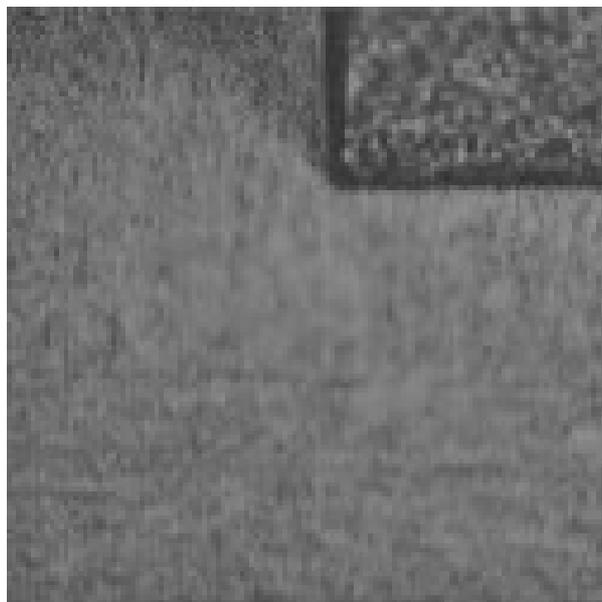


Fig.78 Camera image (distance = 640mm)

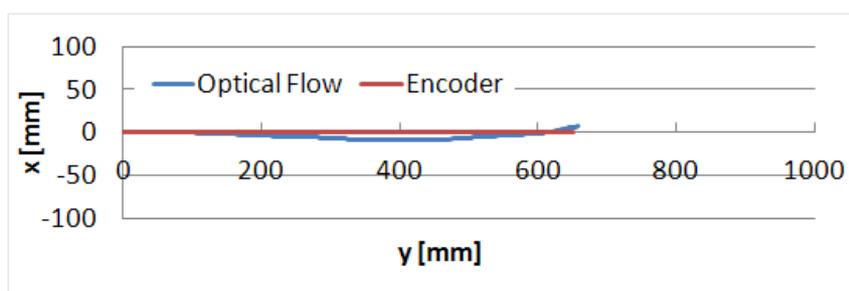


Fig.79 Estimated path (distance = 640mm)

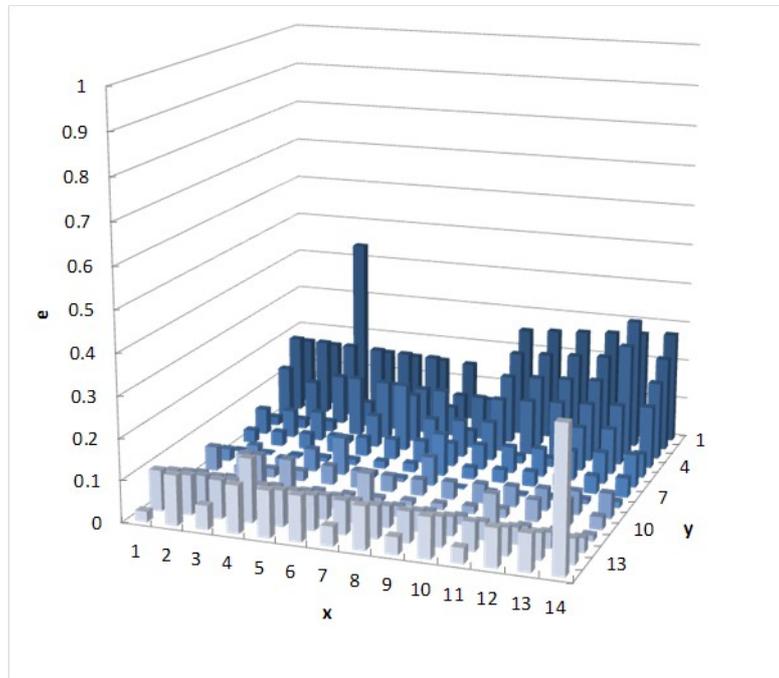


Fig.80 Movement increase ratio (distance = 640mm)

Table 13 Movement increase ratio (distance = 640mm)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0 | 0.11 | 0.11 | 0.4 | 0.1 | 0.1 | 0.1 | 0.1 | 0 | 0.3 | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 |
| 1 | 0.14 | 0.13 | 0.13 | 0.13 | 0.13 | 0.13 | 0.03 | 0.03 | 0.26 | 0.26 | 0.26 | 0.27 | 0.36 | 0.27 |
| 2 | 0.07 | 0.08 | 0.07 | 0.12 | 0.07 | 0.07 | 0.12 | 0.06 | 0.22 | 0.22 | 0.22 | 0.22 | 0.32 | 0.23 |
| 3 | 0.08 | 0.08 | 0.08 | 0.1 | 0.1 | 0.18 | 0.09 | 0.09 | 0.18 | 0.18 | 0.18 | 0.18 | 0.19 | 0.19 |
| 4 | 0.13 | 0.13 | 0.14 | 0.05 | 0.14 | 0.23 | 0.14 | 0.14 | 0.14 | 0.06 | 0.06 | 0.15 | 0.15 | 0.03 |
| 5 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.11 | 0.11 | 0.11 | 0.11 | 0.11 | 0.11 | 0.11 | 0.11 |
| 6 | 0.06 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.16 | 0.07 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| 7 | 0.03 | 0.11 | 0.04 | 0.12 | 0.04 | 0.04 | 0.04 | 0.12 | 0.04 | 0.04 | 0.04 | 0.05 | 0.05 | 0.05 |
| 8 | 0.01 | 0.08 | 0.08 | 0.08 | 0.01 | 0.01 | 0.01 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.1 |
| 9 | 0.05 | 0.05 | 0.03 | 0.03 | 0.05 | 0.05 | 0.06 | 0.06 | 0.06 | 0.06 | 0.06 | 0.02 | 0.08 | 0.06 |
| 10 | 0.02 | 0.02 | 0.05 | 0.02 | 0.03 | 0.03 | 0.05 | 0.03 | 0.03 | 0.03 | 0.05 | 0.03 | 0.03 | 0.03 |
| 11 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| 12 | 0.04 | 0.04 | 0.04 | 0.11 | 0.04 | 0.04 | 0.04 | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| 13 | 0.07 | 0.07 | 0.02 | 0.06 | 0.06 | 0.06 | 0.02 | 0.06 | 0.02 | 0.06 | 0.02 | 0.06 | 0.06 | 0.32 |

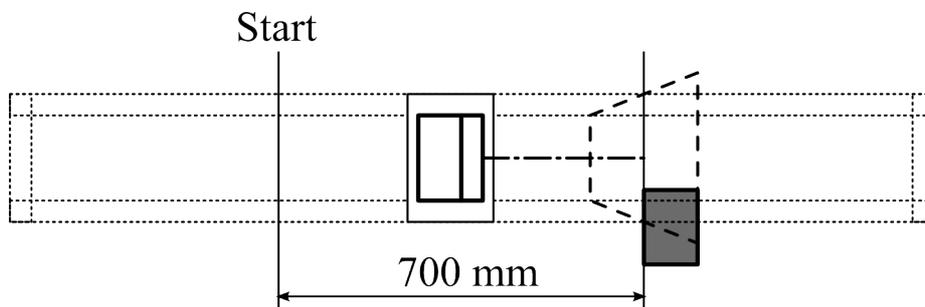


Fig.81 Camera position (distance = 700mm)

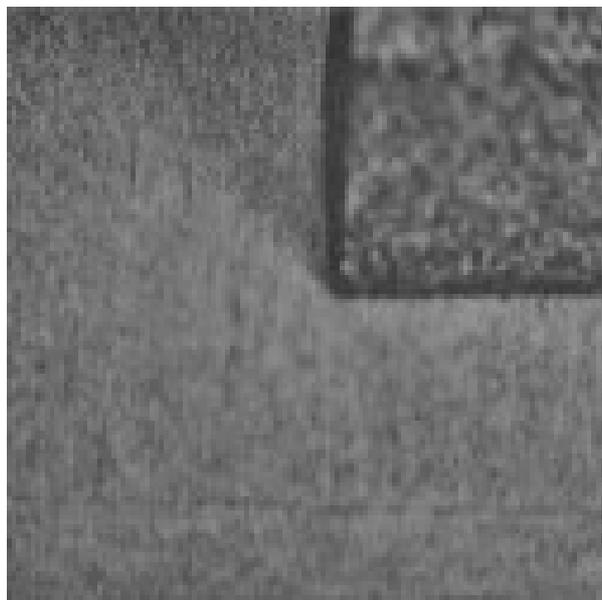


Fig.82 Camera image (distance = 700mm)

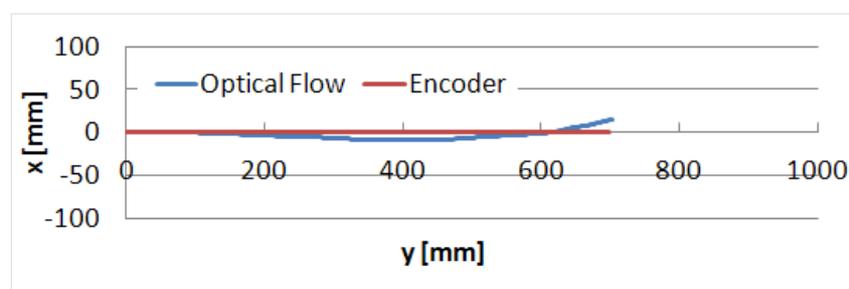


Fig.83 Estimated path (distance = 700mm)

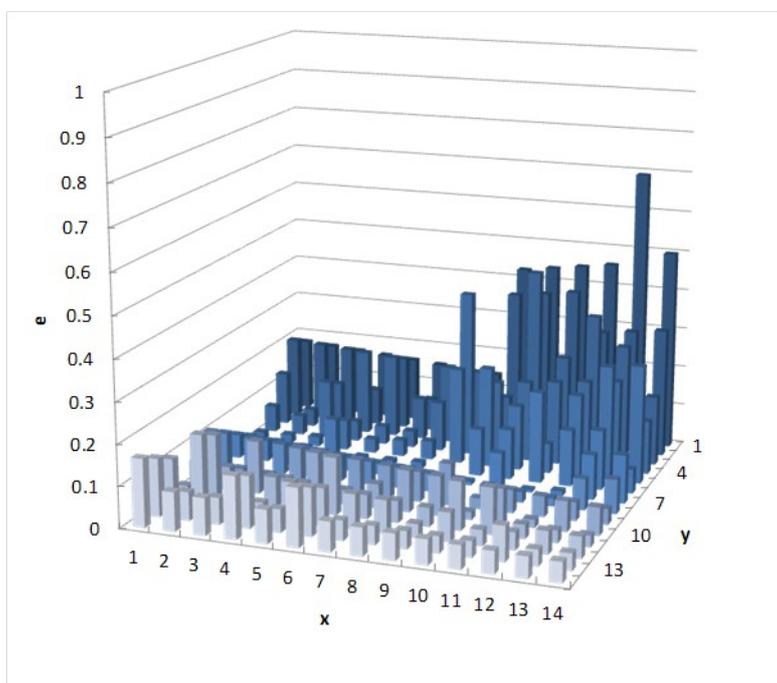


Fig.84 Movement increase ratio (distance = 700mm)

Table 14 Movement increase ratio (distance = 700mm)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0 | 0.1 | 0.1 | 0.09 | 0.09 | 0.09 | 0.08 | 0.08 | 0.08 | 0.5 | 0.51 | 0.51 | 0.52 | 0.75 | 0.53 |
| 1 | 0.13 | 0.13 | 0.12 | 0.01 | 0.12 | 0.01 | 0 | 0.11 | 0.45 | 0.46 | 0.46 | 0.35 | 0.36 | 0.36 |
| 2 | 0.05 | 0.06 | 0.04 | 0.15 | 0.15 | 0.04 | 0.14 | 0.14 | 0.19 | 0.52 | 0.3 | 0.42 | 0.32 | 0.21 |
| 3 | 0.02 | 0.13 | 0.08 | 0.03 | 0.03 | 0.04 | 0.06 | 0.37 | 0.26 | 0.26 | 0.26 | 0.27 | 0.27 | 0.17 |
| 4 | 0.09 | 0.09 | 0.1 | 0.01 | 0.1 | 0.1 | 0.11 | 0.19 | 0.32 | 0.22 | 0.12 | 0.24 | 0.33 | 0.34 |
| 5 | 0.06 | 0.06 | 0.06 | 0.03 | 0.07 | 0.07 | 0.07 | 0.08 | 0.18 | 0.18 | 0.28 | 0.19 | 0.19 | 0.1 |
| 6 | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 | 0.04 | 0.04 | 0.05 | 0.05 | 0.15 | 0.05 | 0.06 | 0.16 | 0.16 |
| 7 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.03 | 0.03 | 0.03 | 0.12 | 0.12 |
| 8 | 0.04 | 0.06 | 0.06 | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.17 | 0.08 | 0.02 | 0.02 | 0.09 | 0.02 |
| 9 | 0.03 | 0.03 | 0.06 | 0.04 | 0.06 | 0.04 | 0.13 | 0.05 | 0.05 | 0.05 | 0.05 | 0.06 | 0.04 | 0.04 |
| 10 | 0.08 | 0.09 | 0.08 | 0.03 | 0.03 | 0.08 | 0.03 | 0.03 | 0.1 | 0.07 | 0.07 | 0.04 | 0.04 | 0.04 |
| 11 | 0.05 | 0.12 | 0.06 | 0.06 | 0.04 | 0.04 | 0.04 | 0.04 | 0.07 | 0.03 | 0.08 | 0.08 | 0.03 | 0.09 |
| 12 | 0.07 | 0.04 | 0.04 | 0.06 | 0.04 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.13 | 0.06 | 0.06 |
| 13 | 0.1 | 0.04 | 0.04 | 0.09 | 0.04 | 0.09 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 | 0.04 |

Table 15 Detection rate and accuracy rate (all flow)

| | 1/3まで接近 | | 1/2まで接近 | |
|---------|-------------|---------------|-------------|---------------|
| | 検出率 | 正解率 | 検出率 | 正解率 |
| 立方体(右) | 20/28 (71%) | 172/196 (88%) | 29/49 (60%) | 161/196 (82%) |
| 立方体(中央) | 36/52 (69%) | 170/196 (87%) | 37/82 (45%) | 144/196 (73%) |
| 壁 | 39/56 (70%) | 161/196 (82%) | 64/98 (65%) | 156/196 (80%) |

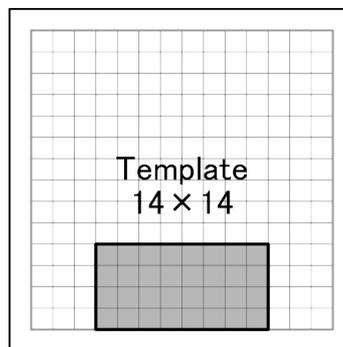


Fig. 85 near area

Table 16 Detection rate and accuracy rate (near flow)

| | 1/3まで接近 | | 1/2まで接近 | |
|---------|-------------|---------------|-------------|---------------|
| | 検出率 | 正解率 | 検出率 | 正解率 |
| 立方体(右) | 24/28 (85%) | 181/196 (92%) | 41/49 (84%) | 184/196 (94%) |
| 立方体(中央) | 43/52 (83%) | 178/196 (91%) | 62/82 (76%) | 173/196 (88%) |
| 壁 | 45/56 (80%) | 182/196 (93%) | 80/98 (82%) | 177/196 (90%) |

第 5 章

結論

第5章 結論

5.1 考察と結論

本論文ではロボットの自律化に際し重要な技術である環境認識，特に動きの認識に焦点をあて、オプティカルフローを用いた研究について述べた。

第2章では、知的センシングシステムを実現するための中核となるオプティカルフロー演算システムの開発について述べた。ロボットは様々な明るさ環境での使用が想定されるため、視覚システムにおけるカメラとして、ダイナミックレンジが広く明るさの恒常性が報告されているシリコン網膜を導入した。オプティカルフローの演算には、低消費電力で高速並列演算が可能なFPGAを用いた。

代表的なオプティカルフローの演算アルゴリズムには、ブロックマッチング法と勾配法があり、ブロックマッチング法は精度は高いが計算時間がかかり、勾配法はノイズに弱いが計算時間が短いといわれていた。そこで、ブロックマッチング法と勾配法の空間的大域最適化法と空間的局所最適化法の3手法の計算精度と演算時間についての比較検証を行った。5ピクセルずらした画像を使用した実験の結果、ブロックマッチング法を用いた場合の誤差が0.8%、標準偏差が0.67となり、勾配法を用いた場合の誤差（空間的大域最適化法、空間的局所最適化法共に100%以上）、標準偏差（空間的大域最適化法：50、空間的局所最適化法：4.5）と比較して優れた値が得られた。また、計算時間の比較ではブロックマッチング法が31ms、空間的大域最適化法が350ms、空間的局所最適化法が16msとなった。ブロックマッチング法の計算時間が空間的局所最適化法の計算時間の倍となっているが、ブロックマッチング法の計算精度の高さと並列演算による計算時間の短縮を見込み、ブロックマッチング法を採用しFPGAへの実装を行った。実装に際し、計算時間に大きく関与するテンプレート領域の大きさと探索範囲を決定した。テンプレート領域の大きさは14×14領域確保できる8×8ピクセルとし、探索範囲はxyの移動量を1バイトで表すことができる22×22ピクセルとした。ブロックマッチング法の演算を高速化するために、テンプレート領域と探査領域の差分計算、並びに差分の和の計算にDSPスライスを使用した。ここで、DSPスライスは十分な数がFPGAに搭載されていなかったためSIMD演算を参考に、1つのDSPスライスで複数画素の同時演算を行った。これにより、約50fpsでオプティカルフローを演算できる視覚システムを開発した。

第3章では、開発した視覚システムを用いて、自身の動きであるエゴモーションの推定について述べた。オプティカルフローによって得られた地面の移動量を用いて、ヘルマー

ト変換行列を求めることにより、視覚システムの移動量を求めた。カメラに映し出される映像と、地面との関係はあらかじめ計測した射影変換によって求めた。開発した視覚システムでは196領域のオプティカルフローが得られるため、最小二乗法を用いてヘルマート変換行列を求めた。レールとスライダを用いた実験では、スライダに視覚システムを搭載し、一定速度でレール上を移動させたときのスライダの移動を検出させた。スライダへの視覚システムの取り付けには、仰角(0°, 30°, 45°)、水平角(0°, 45°, 90°)を変更して実験を行った。この実験により、カメラの取り付け角に関係なく自身の運動が推定できることを確認した。実際に視覚システムを全方位移動ロボットに搭載しての実験では、レールとスライダの実験では行う事ができなかった回転運動に対するエゴモーションの推定と、それらの複合運動に対するエゴモーションの推定を行った。その結果、移動ロボット自身のデッドレコニングが大きな誤差を出す中でも、開発した視覚システムを用いることで、1%程度の誤差でエゴモーションの推定を行えることを示した。

第4章では、視覚システムを用いたドミナントプレーンの抽出について述べた。ドミナントプレーンとは画像中の大半の領域を占める平面のことで、障害物がなくロボットが走行可能な路面であることが多い。ここで、第3章のエゴモーション推定で算出した、ヘルマート変換行列とオプティカルフローを用いてのドミナントプレーン抽出法について提案し、実験によりドミナントプレーンの検出を確認した。これは、ヘルマート変換行列を用いて画面の全体の移動量を算出し、実際にオプティカルフローによって検出した移動と比較することで、画面全体の移動と異なる移動を行っている部分を検出する手法である。この手法は、エゴモーションの推定で算出したヘルマート変換行列と視覚システムによるオプティカルフローの計算結果から得られるため、エゴモーション推定と同時に行っても計算コストの増加が少ないという利点がある。また、従来法では障害物のない平面のオプティカルフローを記憶し、障害物のある環境でのオプティカルフローと比較することでドミナントプレーンを検出していたが、提案手法は障害物のない平面のオプティカルフローを必要としないという利点も持っている。

ドミナントプレーンは先述したように画面の大半の領域を占める平面のことで、ロボットが走行可能な路面を検出しているわけではなく、障害物が画面の大半を占めているときは路面を検出できない場合がある。そこで、ロボット直前の領域に障害物がないと仮定し、その領域で求めたヘルマート変換行列を用いることで、画面の大半に障害物が存在する場合でも、路面の検出を行えることを確認した。なお、この手法では単眼カメラを用いるものなので、ステレオカメラを用いる手法と比較してカメラのキャリブレーションが容易に行えるという利点もある。

以上のことから、開発した視覚システムは、知的センシングを行ううえで目標とした、スタンドアロン動作、移動体の検出、低消費電力で長期間にわたる計測を実現した。本論文ではFig. 86において破線で囲んだ領域を実現し、視覚システム応用例として自己運動の推定、ドミナントプレーン推定および障害物検出を行う事で、開発した視覚システムの有用性を示した。

5.2 今後の展望

本論文では、開発した視覚システムを用いてエゴモーシヨンの推定やドミナントプレーンの検出を行った。視覚システムは、本論文で述べた領域だけではなく、Fig. 86に示すように色や大きさ、奥行きや形状の検出にも用いることができる。そして、検出した情報を統合し対象物の制御まで行える視覚システムの開発が期待できる。

また、エゴモーシヨンの推定やドミナントプレーンの検出において、より精度を高めるためには小さな動きを検出する必要がある。本研究で開発した視覚システムでは、ブロックマッチング法を用いてオプティカルフローの計算を行っており、検出できる運動は画素単位の運動となる。そこで、サブピクセル単位でマッチングを行えるブロックマッチング法を適用することで、さらなる精度の向上が期待できる。

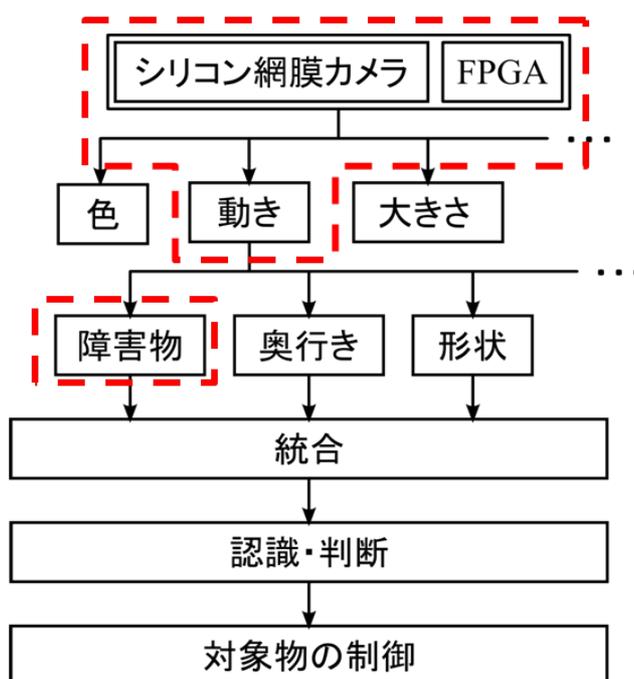


Fig.86 Vision system

謝辞

本論文は九州工業大学大学院生命体工学研究科脳情報専攻における研究生活の中で、多くの方々の御指導，御協力によって完成させることができました。ここに皆様への深謝の意を記します。

九州工業大学大学院生命体工学研究科脳情報専攻 石井和男教授には指導教官として多大なるご指導ご鞭撻を賜りました。石井先生は、自由に研究できる環境を与えて下さり，RoboCup，水中ロボコン，RoboPlus，ロボットデモ・展示会など多くのイベントに参加する機会を与えて下さり貴重な経験を積ませて頂きました。マイペースな私でご迷惑をかけることも多かったと思いますが，常に親身にアドバイスして頂いたおかげで今までやってこられました。ここに深謝の意を表し厚く御礼申し上げます。

本論文を執筆するにあたり，ご指導並びに貴重なご助言を頂きました，九州工業大学大学院生命体工学研究科脳情報専攻 神酒勤教授，宮本弘之准教授に厚く御礼申し上げます。

大阪大学 八木哲也教授，NeuralImageの皆様には，インターンシップおよび研究においてシリコン網膜カメラについてご教授頂き厚く御礼申し上げます。

三ツ和金属株式会社 鈴木徳義社長，森山司さん，会社の皆様には，機械設計・加工について様々なご助言を頂き，本物の設計・加工を知ることができました。ここに感謝の意を示します。

石井研究室において，幸いにも多くの素晴らしい先輩，後輩，同期に恵まれ，多大なるご助言，ご協力を頂きました。

Amir Ali Forough Nassiraei さんには石井研，RoboCup，RoboPlusにおいてご助言，御協力頂きました。特にRoboCupではAmirさんの設計したキッカーの性能には驚かされました。一緒に日本大会で優勝でき，貴重な体験をさせて頂き感謝致します。

佐藤雅紀さんには実験を手伝って頂いたり，多くのご助言を頂きました。また，一緒に旅行に行ったりと，研究・プライベート共にお世話になりました。

石塚誠さんには入学した当初からお世話になりました。入学してすぐに石塚さんの下で研究することで、研究の進め方を学ぶことができました。

西田周平さん、杉山公一さん、大畑智海さん、松尾貴之さんには先輩としてご指導頂き心より感謝致します。

武村泰範君には一緒に RoboCup をやっていく中でいつも助けられてきました。特に、武村君がリーダーをやってくれたおかげで開発に専念できました。そして、日本大会で初優勝したときの興奮は忘れられない思い出です。

神田敦司君とは研究だけでなく RoboPlus で一緒に協力し合い、楽しい研究生活を送ることができました。研究と RoboPlus の仕事を両立でることに驚きでした。

江里口優君が寒い中でも水中の実験を行っているのを見て、自分も頑張ろうと思わせてくれました。ご実家での料理もおいしく、高知旅行は楽しかったです。

白石武尊君とは修士だけの付き合いでしたが、石井研に入ってから初めての学会と一緒にいったのはいい思い出です。

西田祐也君は石井研での後輩としてというより友達としていつも笑いを提供してくれたので楽しい研究生活には欠かせない存在でした。

北住祐一君は RoboCup メンバーとして一緒にロボットの開発を行い、徹夜の作業も乗り越えた戦友です。一緒に初優勝の喜びを分かち合えて幸せでした。北住君が頑張っているのを見て、研究や RoboCup で自分も頑張ろうと奮い立たされていました。

石井研究室の後輩の皆様、先輩らしいこともしてやれずに無理ばかり言っていたと思いますが、協力に感謝致します。

ロボカップメンバーの皆様は、私の作った回路を素晴らしいプログラムで動かしてくれ、作る喜びを実感させてくれました。

最後に、いつも私を温かく見守り続け、経済的・精神的に支援してくれた両親、祖父母、弟に深く感謝致します。

石井研究室において有意義な研究生活をおくれたのは、皆様の多大なお力添えのおかげです。ここに重ねて厚く感謝の意を表し、謝辞と致します。

真田 篤

参考文献

- [1] 経済産業省, “ロボット分野に関するアカデミック・ロードマップ”, URL:http://www.meti.go.jp/policy/economy/gijutsu_kakushin/kenkyu_kaihatsu/19fy-pj/r_1.pdf, 2008
- [2] 教育機器編集委員会編, “産業教育機器システム便覧”, 日科技連出版社, 1972
- [3] 井上 渉, 川西 亮輔, 山下 淳, 金子 透, “単眼カメラ搭載移動ロボットを用いた SIFT 特徴量に基づく特徴追跡と Structure from Motion による SLAM”, 2009 年度精密工学会春季大会学術講演会講演論文集, 2009
- [4] 岩田啓明, 実吉敬二, “ステレオカメラによるビジュアルオドメトリおよび自己位置認識”, 第 11 回計測自動制御学会システムインテグレーション部門講演会, 2010
- [5] Alessandro Bissacco, Soheil Ghiasi, “Fast Visual Feature Selection and Tracking in a Hybrid Reconfigurable Architecture”, Proceedings of the Workshop on Applications of Computer Vision (ACV), 2006.
- [6] Andrew Howard, “Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles”, In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3946-3952, 2008.
- [7] Stephen Se, Piotr Jasiobedzki, “Stereo-Vision Based 3D Modeling and Localization for Unmanned Vehicles”, International Journal of Intelligent Control and Systems, vol. 13, no. 1, pp. 47-58, 2008.
- [8] J. A. Cooney, W. L. Xu, G. Bright, “Visual Dead-Reckoning for Motion Control of a Mecanum-Wheeled Mobile Robot”, Mechatronics, Volume 14, Issue 6, pp. 623-637, 2004.
- [9] Atsushi Sakai, Yuya Tamura, Yoji Kuroda, “Visual Odometry Using Feature Point and Ground Plane for Urban Environments”, Proceedings for the joint conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), pp. 1-8, 2010.
- [10] David Nistér, Oleg Naroditsky, James Bergen, “Visual Odometry”, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Volume 1, pp. 652-659, 2004.

- [11] 岩田啓明, 実吉敬二, "ステレオカメラによるビジュアルオドメトリおよび自己位置認識", 第11回計測自動制御学会システムインテグレーション部門講演会, 2010.
- [12] 田中卓史, "光学マウスセンサーを用いた移動ロボットの制御", 福岡工業大学研究論集, Vol. 40, No. 2, pp. 217-221, 2008.
- [13] 関森大介, 宮崎文夫, "複数の光学マウスセンサ値とシンプルな俯瞰カメラ情報を用いた屋内移動ロボットの自己位置推定", 日本機械学会論文集, C編 71(712), pp. 144-151, 2005.
- [14] 山崎公俊, 稲葉雅幸, "カメラとLRFを併用した日常生活支援ロボットのための家具姿勢認識", 第26回日本ロボット学会学術講演会講演論文集, 3F3-07, 2008
- [15] 久徳遙矢, 出口大輔, 高橋友和, 目加田慶人, 井手一郎, 村瀬洋, "車載カメラ映像の道路面経時差分による前方障害物検出", 画像の認識・理解シンポジウム(MIRU) 2009 論文集, IS1-55, pp.734-738, 2009
- [16] 興梠正克, 村岡洋一, "一次元フロー計算による自己運動の復元と障害物の検出", 情報処理学会研究報告. CVIM, [コンピュータビジョンとイメージメディア] 97(39), pp. 1-8, 1997.
- [17] Naoya OHNISHI and Atsushi IMIYA, "Navigation from optical flow", 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション 104(667), pp. 67-72, 2005.
- [18] Kazushiko Kawamoto, Reinhard Klette, "Dominant Plane Estimation", Communication and Information Technology Research Technical Report 88, 2001.
- [19] Chi-hau Chen, "Handbook of pattern recognition and computer vision", World Scientific, pp. 257-281, 2010.
- [20] Naoya OHNISHI, Atsushi IMIYA, "Model-Based Plane-Segmentation Using Optical Flow and Dominant Plane", Computer vision/computer graphics, collaboration techniques: third international conference, MIRAGE 2007, pp. 295-306, 2007.
- [21] Pär Fornland, Christoph Schnörr, "A robust and convergent iterative approach for determining the dominant plane from two views without correspondence and calibration", Proc. Computer Vision and Pattern Recognition (CVPR'97), pp. 508-513, 1997.
- [22] Andreés Colín-Espinoza, Héctor Alejandro Montes-Venegas, María Enriqueta

- Barilla-Pérez, "Improved Image Dominant Plane Extraction for Robot Navigation", Research and Education in Robotics - Eurobat 2010: International Conference, pp. 1-13, 2010.
- [23] Pär Fornland, "Dominant plane detection for uncalibrated binocular vision", Proc. International Symposium on Intelligent Robotic Systems (ISIRS'98), 1998.
- [24] Kazushiko Kawamoto, Reinhard Klette, "Navigation Using Optical Flow Fields: An Application of Dominant Plane Detection", Communication and Information Technology Research Technical Report 111, 2002.
- [25] Marc Pollefeys, Frank Verbiest, Luc Van Gool, "Surviving dominant planes in uncalibrated structure and motion recovery", Johansen (Eds.) Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Lecture Notes in Computer Science, Vol.2351, 2002.
- [26] 森邦洋, 佐藤雅紀, 武村泰範, 石井和男, "小型全方位移動ロボットによる群知能に関する研究", Robomec2007 CD-ROM, 2A1-D11, 2007
- [27] Nagatani Keiji, Tachibana Satoshi, Sofue Makoto, Tanaka Yutaka, "Improvement of Odometry for Omnidirectional Vehicle using Optical Flow Information", Intelligent Robot and Systems, pp. 468-473, 2000
- [28] 祖父江誠, 永谷圭司, 田中豊, "ビジュアルデッドレコニングによる全方向移動車のオドメトリ精度の改良", 第18回 日本ロボット学会学術講演会予稿集, pp.109-110, 2000
- [29] S. Zingg, D. Scaramuzza, S. Weiss, R. Siegwart, "MAV Navigation through Indoor Corridors Using Optical Flow", Proc. of The IEEE International Conference on Robotics and Automation (ICRA), May 2010
- [30] Luigi Di Stefano, Enrico Viarani, "Vehicle Detection and Tracking Using the Block Matching Algorithm", Proc. of 3rd IMACS/IEEE, pp. 4491-4496, 1999
- [31] 大島直也, 齊藤剛史, 小西亮介, "オプティカルフロー分布を利用した Mean Shift 追跡", 信学論, Vol. J90-D, No.4, pp.1096-1104, 2007.4
- [32] 千葉直樹, 蚊野浩, 美濃導彦, 安田昌司, "画像特徴に基づくイメージモザイク", 電子情報通信学会論文誌 (D-II), Vol. J82-D-II, No. 10, pp. 1581-1589,

1999

- [33] 武田和大, 重留美穂, 小野智司, 中山茂, “オプティカルフローによる読唇の研究”, 2003 PCカンファレンス, 8-F-6, p379, 2003
- [34] 村上翔太郎, 田中秀典, 長島知正, “オプティカルフローを用いた表情解析による心理状態推定”, SVBL年報 Vol.8, pp.67-68, 2009
- [35] Intel, “プロセッサ製品”,
URL:<http://www.intel.co.jp/jp/support/processors/sb/CS-023143.htm>.
- [36] overclock.net, “CPU Gflop Performance Database”,
URL:<http://www.overclock.net/t/586994/cpu-gflop-performance-database>.
- [37] GPUGRD.net, “Nvidia GPU Card comparisons in GFLOPS peak”,
URL:http://www.gpugrid.net/forum_thread.php?id=1150.
- [38] RENESAS, “SH-4 シリーズ機能比較一覧表”,
URL:http://japan.renesas.com/products/mpumcu/superh/sh7750/child_folder/function_comparison_sh.jsp.
- [39] 佐藤一輝, バートルスレンバルス, 関根優年, “FPGA アレイを用いて TFlops を目指したポアソン方程式演算回路の実装と評価”, Vol.108 No.412 RECONF2008-58 pp.19-24, 2009.
- [40] C.ミード, 訳: 臼井史朗, 米津宏雄, “アナログVLSIと神経システム”, トッパン, 1993
- [41] P. Lichtsteiner, C. Posch, and T. Delbruck, A 128x128 120dB 15us latency asynchronous temporal contrast vision sensor, IEEE J. Solid State Circuits, 43(2) 566-576, 2007
- [42] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128x128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change,” presented at 2006 International Solid State Circuits Conference (ISSCC 2006), San Francisco, Feb. 2006
- [43] Shih-chii Liu, “Silicon Retina with Adaptive Filtering Properties”, in Advances in Neural Information Processing Systems 10, pp. 712-718, 1997.
- [44] Becanovic V, and Indiveri G, and Kobialka H-U, and Ploger P, and Stocker A, Mechatronics and Machine Vision 2002: Current Practice; p.13-21, ISBN:0-

- [45] Shih-chii Liu, “Silicon Retina with Adaptive Filtering Properties”, *Advances in Neural Information Processing Systems* 10, pp. 712-718, 1997
- [46] Kazuhiro Shimonomura, Tetsuya Yagi, “Computing lightness constancy with an APS-based silicon retina”, *IEEE Biomedical Circuits and Systems Conference (BioCAS 2008)*, in *Proc.* pp.201-204, 2008
- [47] 奥野弘嗣, 下ノ村和弘, 八木哲也, “エントロピー最大化に基づいた順応型視覚センサ”, *日本機会学会ロボティクス・メカトロニクス講演会 2008 講演論文集*, pp.2P2-B12(1)-(4), 2008
- [48] 奥野弘嗣, 今井快多, 八木哲也, “実時間画像処理機能を備えた広ダイナミックレンジイメージセンサ”, *日本機会学会ロボティクス・メカトロニクス講演会 2009 講演論文集*, pp.1A1-C18(1)-(3), 2009
- [49] Kazuhiro Shimonomura, Tetsuya Yagi, “Neuromorphic VLSI vision system for real-time texture segregation”, *Neural Networks*, vol.21, no.8, pp.1197-1204, 2008
- [50] Kazuhiro Shimonomura, Takayuki Kushima, and Tetsuya Yagi, “Binocular robot vision emulating disparity computation in the primary visual cortex”, *Neural Networks*, vol.21, no.2-3, pp.331-340, 2008
- [51] Jun Hasegawa and Tetsuya Yagi, “Real-Time Emulation of Neural Images in the Outer Retinal Circuit”, *The Journal of Physiological Sciences*, vol.58, no.7, pp.507-514, 2008
- [52] A. Gyaourova, C. Kamath, and S.-C. Cheung, “Block matching for object tracking,” *Tech. Rep. UCRL-TR-200271*, Lawrence Livermore National Laboratory, Livermore, Calif, USA, 2003
- [53] 五味広美, “テンプレートマッチング法による実時間動き検出”, *Technical report of National Aerospace Laboratory 1358*, 1-22, 1998
- [54] 北澤仁志, 李竹, 藪田顕一, “排他的ブロックマッチングによる移動物体の抽出と追跡”, *電子情報通信学会技術研究報告. IE, 画像工学* 108(4), pp. 49-54, 2008.
- [55] Luigi Di Stefano, Enrico Viarani, “Vehicle Detection and Tracking Using the Block Matching Algorithm”, *Proc. of 3rd IMACS/IEEE*, pp. 4491-4496, 1999.

- [56] Berthold K. P. Horn and Brian G. Schunck, "Determining Optical Flow", ARTIFICIAL INTELLIGENCE, vol.17, pp.185-203, 1981
- [57] Bruce D. Lucas, Takeo Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision., 1981, 674-679
- [58] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", Intel Corporation Microprocessor Research Labs, 1999.
- [59] Jean-Yves Bouguet, "Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the algorithm", Intel Corporation Microprocessor Research Labs, 2000.
- [60] Naoya Ohta, "Image movement detection with reliability indices", IEICE TRANSACTIONS on Information and Systems, Vol.E74-D, No.10, pp. 3379-3388, 1991.
- [61] 三池秀敏, 長篤志, 三浦一幸, 杉村敦彦, "一般化勾配法によるオプティカルフローの検出", 情報処理学会論文誌. コンピュータビジョンとイメージメディア 49(SIG_6(CVIM_20)), 1-12, 2008-03-15
- [62] 福井一貴, 安藤隆男, 澤田和明, "勾配法によるオプティカルフロー検出手法", テレビジョン学会年次大会講演予稿集 (31), 171-172, 1995-07-26
- [63] 木村和広, 太田直哉, 金谷健一, 精密なノイズモデルによるオプティカルフローの検出, 情報処理学会研究会報告, 96-CV-99-6, pp.37-42.
- [64] 藪木, 登 and 三木, 成彦 and 前川, 禎男 (1994) カラー動画像からのオプティカルフロー検出に関する検討. 津山工業高等専門学校紀要, 34 . pp. 61-68. ISSN 0287-7066
- [65] 平健介, 柴田昌明, 画像情報とカメラモーションの融合による新しいオプティカルフロー生成手法, J.Fac.Sci, Tech., Seikei Univ. Vol. 43 No.2(2006) pp.87-93, 2006
- [66] 岡田慧, 加賀美聡, 稲葉雅幸, 井上博允, "再帰相関法とマルチメディア命令による高速オプティカルフロー計算法", 情報処理学会 第115回 CVIM 研究会予稿, pp127-132, 1999
- [67] 大脇吉則, 原田雅樹, 藤井俊彰, 木本伊彦, 谷本正幸, 二乗誤差を判定基準とするブ

ロックマッチングの高速化, IPSJ SIG Notes 99(107), 7-12, 1999

- [68] 布施孝志, 清水英範, 堤盛人, “オプティカルフロー推定における光学勾配法の比較分析”, 応用測量論文集, Vol.11, pp.45-52, 2000
- [69] 山本孝之, 橋本秀雄, “輝度勾配法を用いたオプティカルフローによる動き推定の比較”, 映像情報メディア学会年次大会講演予稿集 (1999), 304-305, 1999-08-23
- [70] Intel, “C++ 命令 クラス・ライブラリ リファレンス・マニュアル”, URL:<http://www.intel.com/jp/download/index.htm>
- [71] Xilinx, “XtremeDSP for Virtex-4 FPGAs User Guide”, URL:http://www.xilinx.com/support/documentation/virtex-4_user_guides.htm
- [72] Enkelmann, W. “Obstacle detection by evaluation of optical flow fields from image sequences” Image and Vision Computing, 9, 160-168, 1991
- [73] Santos-Victor, J. and Sandini, G., “Uncalibrated obstacle detection using normal flow”, Machine Vision and Applications, 9, 130-137, 1996
- [74] Naoya OHNISHI, Atsushi IMIYA, “Dominant plane derives dominant motion”, 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション 104(667), 67-72, 2005
- [75] Naoya OHNISHI, Atsushi IMIYA “Dominant plane detection using optical flow and Independent Component Analysis”, Machine Learning and Data Mining in Pattern Recognition 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005
- [76] Naoya OHNISHI, Atsushi IMIYA, “Dominant plane detection from optical flow for robot navigation”, Pattern Recognition Letters archive Volume 27 Issue 9, 1 July 2006
- [77] Naoya OHNISHI, Atsushi IMIYA, “Dominant plane detection from optical flow for robot navigation”, IPSJ SIG Notes. CVIM 2004(91), 49-56, 2004-09-10
- [78] 新井元基, 鷺見和彦, 松山隆司, “画像のブロックマッチングにおける相関関数とサブピクセル推定方式の最適化”, 情報処理学会研究報告. CVIM, コンピュータビジョンとイメージメディア 2004(40), pp. 33-40, 2004
- [79] 清水雅夫, 奥富正敏, “画像のマッチングにおける高精度なサブピクセル推定手法”, 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理 J84-D-II(7),

pp. 1409-1418, 2001.

- [80] 波部斉, 角岡幹篤, 鷺見和彦, 松山隆司, ”画像の2次元空間構造を利用したサブピクセルマッチングの高精度化”, 情報処理学会シンポジウム論文集, IS3-124, 2005.
- [81] 波部斉, 角岡幹篤, 鷺見和彦, 松山隆司, ”画像の空間構造を利用したサブピクセルマッチングの高精度化”, 電子情報通信学科論文誌, Vol. J89-D, No. 2, pp. 393-398, 2006.