

4. ロボットビジョンの研究

プロジェクト研究員

山本美子(株式会社エイアールテック)

渡部悠紀(広島大学先端物質科学研究科)

4.1 背景・目的

現在、人間と共存するロボットの開発が盛んに行われており、ロボットが自ら外界の情報を得て、自律的に行動することが強く求められている。

ロボットが空間情報を得るためには視覚が必須である。ロボットに視覚機能を搭載することにより、空間中に存在する障害物や目標物、人間の顔などを認識することが可能となる。視覚処理には検出と認識が必要であり、それに必要な演算量は膨大である。現在のロボットでは、視覚機能として必要な処理をそれぞれ別個の処理として行っており、この場合、得られる情報量が膨大でその中からさらに必要な情報を抽出する処理が必要になり、その結果、処理時間の拡大を招く。人間の場合、視覚から得られる膨大な情報の中から自分の必要としている情報を選択的に処理し、素早く抽出することができる。ロボットビジョンにおいても、視野内に入ってくる情報すべてに一律な処理を施すのではなく、画像領域毎に必要な処理をかけることで情報量を減らすことが重要であると考えられる。そこで、本研究では知能的なリアルタイムロボットビジョンシステムを構築することを目的とする。特に、ステレオビジョンとオプティカルフローによる障害物検出処理の実現を目指す。

4.2 技術の現状

4.2.1 現在のロボット開発

現在、開発されているロボットの視覚機能について表1にまとめる。

表1 ロボットの視覚機能

| ロボット名 | 視覚センサ | 主な視覚機能 |
|-----------------------|------------------------------|---------------------------------------|
| SDR-4X SONY | CMOS センサ x2 赤外線方式測距センサ x3 | ステレオビジョン 顔認識 |
| aibo SONY | CMOS センサ x1 赤外線方式測距センサ x1 | 色認識 動物体検出 |
| asimo HONDA | CCD カメラ x2 | 簡単なパターン認識 |
| SIG 北野共生 プロジェクト | CCD カメラ x2 | ステレオビジョン 顔認識 実時間人物追跡(聴覚センサも用いる) |

| | | |
|---------------|-------------------------|--------------------------|
| PaPeRo NEC | CCD カメラ x2 超音波センサ x5 | ステレオビジョン 動画像処理 顔認識 |
|---------------|-------------------------|--------------------------|

既に開発されているロボットの中にはステレオビジョン、動画像処理など障害物検出機能が盛り込まれているものもあるが、障害物検出を安定して行うことが困難であるため、赤外線センサなどの測距センサも同時に搭載されている場合が多い。カメラから得られる画像は常に最良の状態で得られるとは限らず、環境の影響を強くうけてしまい、正確な処理結果を得ることができない場合が多い。そのため、環境の変化に適応し、安定して処理を行うことが重要な課題となっている。

4.2.2 ステレオビジョンの原理

ステレオビジョンとは、三角測量の原理により、物体の3次元空間における位置を検出する手法である。つまり、異なる視点から同一の対象を観測し、そのそれぞれの画像上への投影位置の違いから、対象の3次元位置情報を得ようとするものである。

ステレオ画像を撮影する際の基本的なカメラの配置を図1に示す。

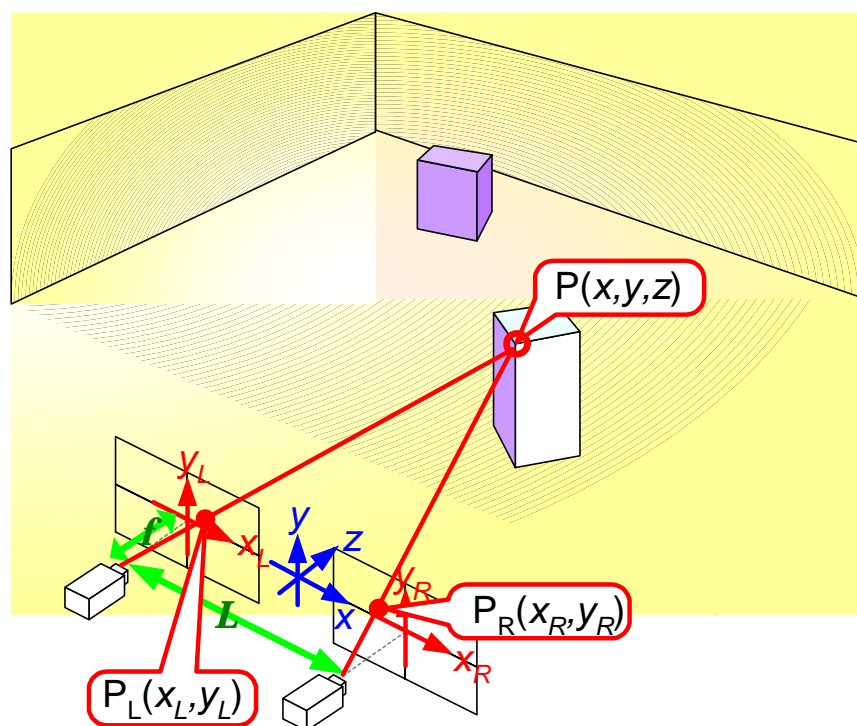


図1 ステレオビジョンの原理

この配置ではカメラの光軸が互いに平行で、かつ各々の画像面が同一平面上にあるようにしている。図中、 f はカメラの焦点距離、 L は2つのカメラの基線長でそれぞれ既知の値

である。空間中の物体上の点 P がそれぞれの画像上の点 P_L, P_R に投影されたとする。ワールド座標系 (x, y, z) と画像座標系 $(x_L, y_L), (x_R, y_R)$ の間には以下のような関係が成り立つ。

$$x = \frac{x_L + x_R}{2} \frac{L}{x_L - x_R}$$

$$y = y_L \frac{L}{x_L - x_R}$$

$$z = f \frac{L}{x_L - x_R}$$

$x_L - x_R$ が左右画像間での投影位置の違いであり、両眼視差と呼ばれる。 f, L は既知の値であることから、両眼視差を求めることにより、左右画像上の投影座標 P_L, P_R から、もとの3次元位置 $P(x, y, z)$ を求めることができる。特に奥行き z に関しては視差のみによって計算される。

ステレオビジョンでは、両眼視差を求める際に対応付けを必要とする。対応付けを行う際、エピポーラ拘束(図2)を利用して対応点探索範囲を限定することができる。エピポーラ拘束とは、一方の画像 I_1 中の点を P_1 とすると、その画像中でのエピポーラ線が1本定まり、そのエピポーラ線と対応する他方の画像 I_2 中のエピポーラ線が定まる。点 P_1 と対応する画像 I_2 中の点 P_2 は計算された線上に存在するという拘束条件である。

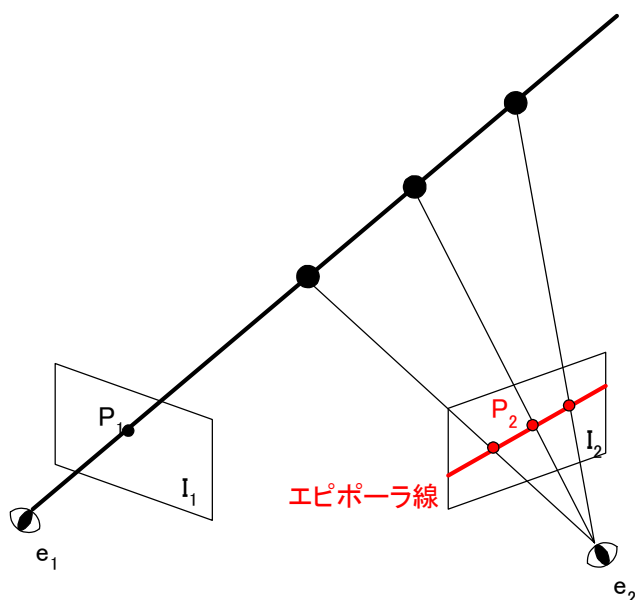
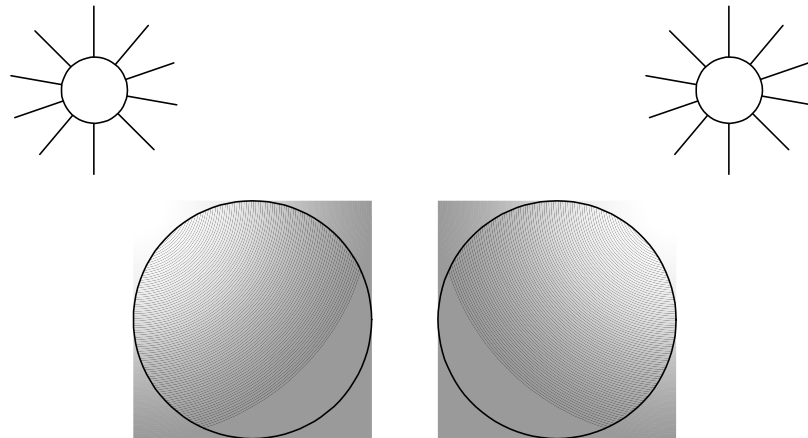


図2 エピポーラ拘束条件

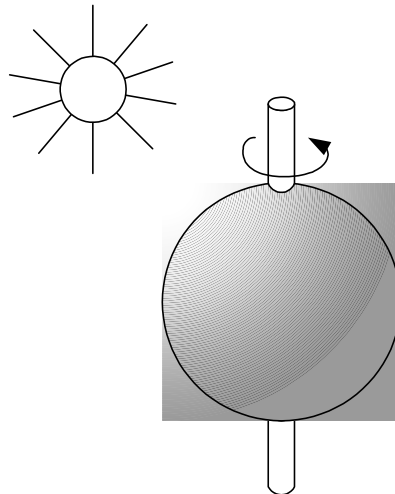
これにより対応点探索問題を 2 次元から 1 次元の問題に落とすことができる。この条件に加え、図 1 に示すようなカメラ配置にすると、対応点探索範囲を同一スキャンラインに限ることができるために計算が簡略化できるので、よく用いられている。本研究でもこのカメラモデルを採用する。

4.2.3 動きベクトル検出の原理

物体、観測者のいずれかが動くと画像中に変化を生じる。動画像解析における基本の要素は、カメラ、物体、光源の 3 つであり、物体もしくは観測者が動くだけではなく、光源が動いても画像中に動きとして変化が現れる。この画像中の明るさパターンが見かけ上動いて見えることをオプティカルフローという。これに対して、画像中の各点に速度ベクトルを対応させたものを動き場という。画像から入手できる情報は動き場ではなく、オプティカルフローである。オプティカルフローはほとんどの場合、動き場と一致するが、図 4 に示すような場合は一致しないので注意が必要である。図 3 に示すような状況を除けば、オプティカルフローは動き場とそれほど変わらないという事実に基づいて動きの情報を得る。本研究ではオプティカルフローと動きベクトルは等しいとして処理を行う。



(a) 静止した球が移動する光源によって照らされている。
画像の陰影は変化するが、動き場は零である。



(b) 滑らかな球が一定の照明の下で回転している。
画像は変化しないが、動き場は零ではない。

図3 オプティカルフローと動き場の関係

オプティカルフローを推定するためには、あるフレームとその次のフレーム間に対応付けを行う必要がある。この対応付け処理は、ステレオビジョンの場合と違い、探索範囲を制限する拘束条件はないので、2次元探索問題となり、一般に演算量が大きい。

オプティカルフローを求めることによって得られた動き場の情報から、障害物の情報を検出してロボットに与える必要がある。動き場の情報から障害物の情報を得る手法の1つにオプティカルフローを延長したときに生じる交点(FOE:Focus of Expansion)を利用する方法がある。

まず、静止した状態で画像を撮影した時を考える。カメラで画像を撮影する際、透視変

換が行われている。透視変換によって三次元空間中の平行線を投影すると画像平面上ではそれらの直線は1点に収束する。この点のことを消失点と呼び、三次元空間中の無限遠に相当する(図4)。

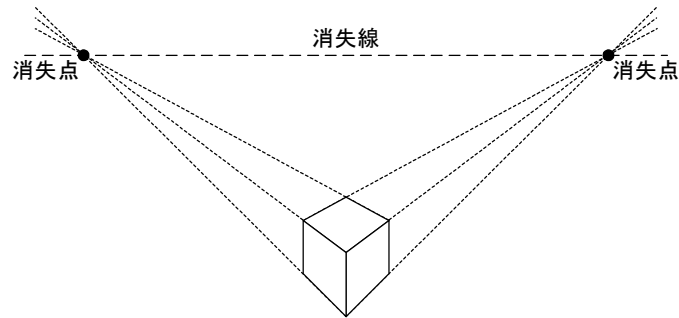


図4 消失点と消失線

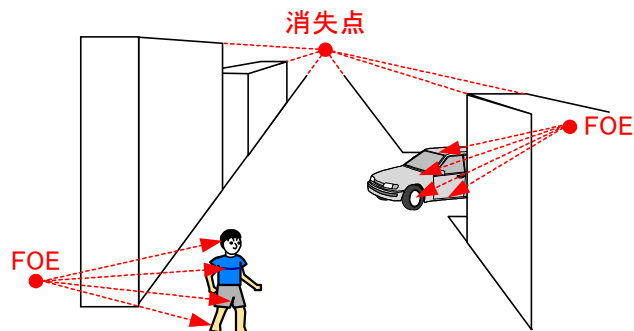


図5 消失点とFOE

観測者が並進運動を行うと、オプティカルフローが生じるが、そのオプティカルフローは延長するとある1点を通過するいろいろな長さのベクトルから成る。この交点をFOE(Focus of Expansion, 拡大焦点)と呼び、これは動きベクトルの消失点と考えられる。FOEは動きの方向ごとに1つ現れるので、観測者と異なる方向へ動く物体がある場合には図5に示すように物体のオプティカルフローから求められるFOEは背景領域の消失点とは異なる場所に出現する。また、進行中の視点が背景領域のFOEとなるため、それ以外の場所にFOEを形成するフローは移動障害物から抽出されたオプティカルフローであると判断することができ、背景領域と移動障害物領域を分離して検出することができる。

ロボットが実際に必要とするのはどのような障害物がどこにあるか、というであり、FOE

を求めることで障害物が移動障害物であるか静止障害物であるかを判断することが可能となる。

図6に消失点とオプティカルフローの現れ方をまとめる。

| | | | | | |
|----------|------------|--------------|--------------|------------|---------------------|
| 上から見た図 | 進行方向が異なる場合 | | | | |
| | | | | | |
| 観測者から見た図 | | | | | |
| | | | | | |
| 抽出されるフロー | | 物体領域のみフローが存在 | 物体領域のみフローが存在 | 全ての点でフローなし | 背景と移動障害物で異なる消失点のフロー |

| | | | | |
|----------|-----------|---------------|-------------|---------------|
| 上から見た図 | 進行方向が同じ場合 | | | |
| | | | | |
| 観測者から見た図 | | | | |
| | | | | |
| 抽出されるフロー | | フローの消失点が背景と同じ | 物体領域: フローなし | フローの消失点が背景と同じ |

図6 オプティカルフローの現れる様子

4.3 ロボットビジョンに要求される性能

ロボットをリアルタイムに制御するためには、視覚処理を高速に行う必要がある。ロボットを制御するためには、視覚から得られた情報に何らかの情報処理を行い、得られた結果に対してロボットの移動経路を求めるなどの後処理が必要となる。その際、後処理に負担をかけないために、不必要な結果は出力しないようにすることも重要な意味をもつ。

また、安全に移動するために処理結果が正確に求められることが重要であり、そのためには環境の変化に対して適応できることが必要である。さらに、視覚はロボットに搭載されるひとつの機能であるから、システム全体が小さいことも要求される。

このような要求に応えるためには、並列処理可能なアルゴリズムと処理を高速に行うことのできるハードウェアが必要とされる。

4.4 アルゴリズムの検討

4.4.1 障害物検出アルゴリズム

障害物は静止障害物と移動障害物の2つに大きく分けることができる。ただし、観測者は移動系とする。本研究ではステレオビジョンとオプティカルフローによって障害物を検出することとする。ステレオビジョンでは、観測者から物体までの絶対的な距離を計測し、障害物の存在位置を知ることが可能で、動きベクトルからは、観測者と物体との相対的な速度を求め、移動障害物検出が可能である。この2つの処理はともにブロックマッチングによって実現が可能で、並列化およびハードウェア化しやすい。

4.4.2 動きベクトルを用いた動物体検出手法の検討

動きベクトルは時系列画像間に対応付けをとって求める。この際、類似などにより画像中全ての点で動きベクトルが正確に得られるとは限らない。室内環境では壁面など明確な特徴のない領域が存在するため、誤った動きベクトルが混入しやすい。誤った動きベクトルが大量に混入すると、正しい消失点を求めることが困難になる。また、処理時間も無駄になる。そこで、エッジ点のみで動きベクトルを推定し、エッジ点で求められた動きベクトルより動物体領域を推定することとした。エッジ点のみの情報が出力されるので面情報が欠落するが、面情報は後処理により復元することができる。

動きベクトルを用いて移動物体を検出するために、背景の FOE と移動物体の FOE を分離する必要がある。FOE を求めるためには複数のベクトルの組み合わせをとり、その交点を求める必要があるが、その演算量は大きい。そこで、組み合わせの数を減少させることによって演算量の減少を図った。以下にその内容を示す。

観測者が直進移動系であり、カメラの光軸は水平であると仮定する。このとき、背景のみかけ上の動きから得られる全ての動きベクトルの交点(FOE)は、理想的には画像中心となる。このことから、画像中心以外に FOE が現れれば、それは移動物体の動きから得られる FOE であると考えられる。そこで、画像中心を原点とする画像座標系の x 軸と動きベクトルとの交点を求めることとし、組み合わせの数を減少させて演算量の低減を図った。 $x=0$ 付近に交点を形成するベクトルは背景領域から得られるものとし、中心以外に交点を形成するベクトルは移動物体から得られるフローと判断する。

入力画像を図7に示す。図7は画像中右にある箱が左に動き、観測者が前へ直進したと

して撮影された画像である。この画像に対して求められた動きベクトルと求められた動きベクトルと x 軸との交点を求めた結果を図 8 に、ヒストグラムに交点の個数をプロットしたものを図 9 に示す。ヒストグラムを演算する際、 x 軸を 16pixel 毎に分割して交点のカウントを行っている。図 10 に示すように、ヒストグラムには画像中心にピークが現れており、これが背景領域から得られる動きベクトルとの交点と考えられる。ヒストグラムに基づき背景領域と移動物体領域を分離した結果を図 10 に示す。図 10 から、移動物体領域と背景領域が分離されていることが確認できる。演算時間は、UNIX WS(Sun UltraSPARK-II 450MHz)上で MATLAB によってシミュレーションを行った結果、もともと約 2600 秒かかっていたものが約 18 秒となり、100 分の 1 以下に短縮された。以上のことから本手法の有効性が確認できた。

この手法では、観測者と同一の方向に移動する物体は図 7 に示すように背景と移動物体の消失点が重なるため、背景領域から分離することができない。この点については、通常、観測者の前に移動物体が現れるときは角度をもって視野に入ってくるため、最初の段階で移動物体領域を検出することができれば、その後も注視することは可能であると考えられる。



(a) 移動前



(b) 移動後

図 7 入力画像

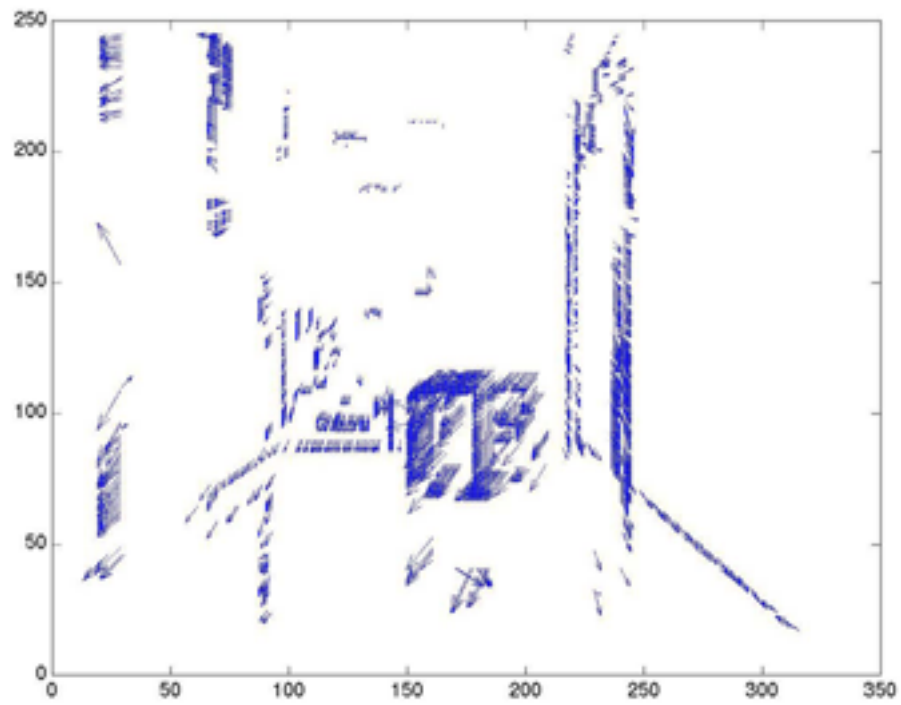


図8 検出された動きベクトル

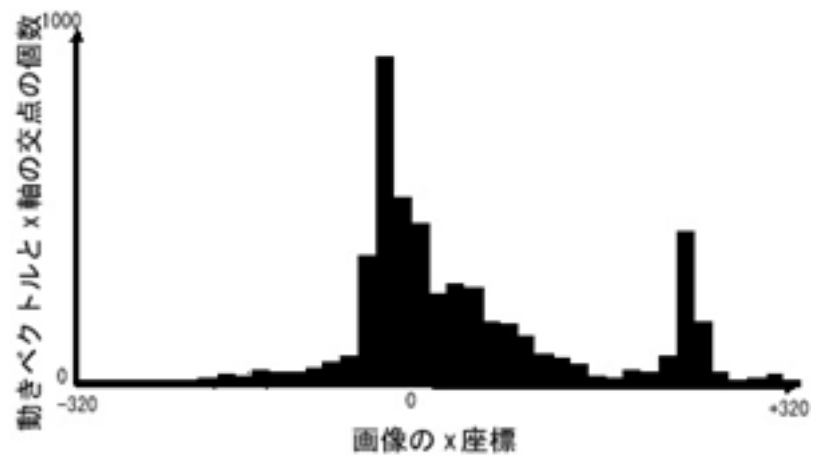


図9 検出された動きベクトルと x 軸との交点のヒストグラム

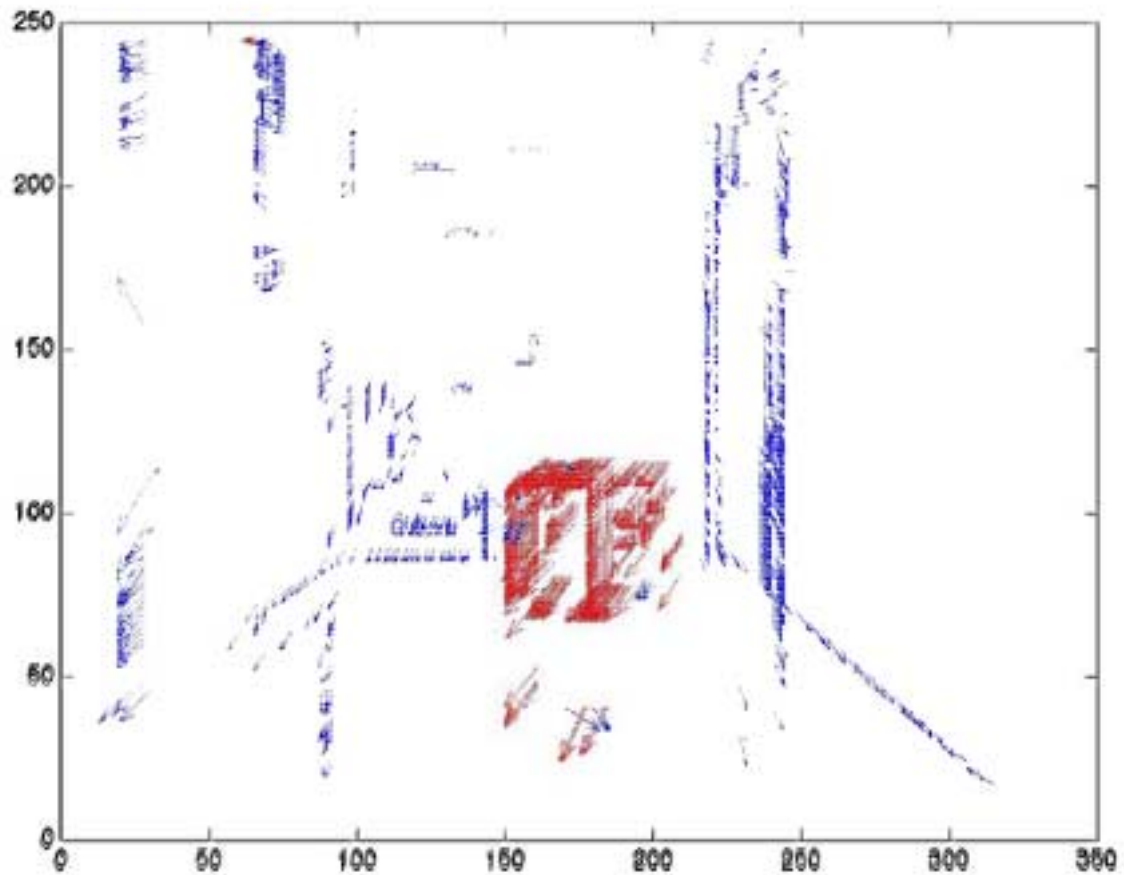


図 1 0 背景の動きベクトルと移動物体の動きベクトルを分離した結果
(青色：背景，赤色：移動物体)

4.4.3 統合処理

人間は、自分が次に行う行動のために必要な情報を様々な処理を組み合わせで効率よく抽出している。ロボットが行動する際にも、自身の行動にあわせて様々な処理を組み合わせる必要がある。ここでは、視覚処理としてステレオ処理と動画処理を対象とし、これらを組み合わせで障害物情報をより効率よく正確に抽出する方法について検討する。

組み合わせ方としては以下の 3 通りが考えられる。

1. ステレオ処理と動画処理を完全並列で行う。

一般に、動画処理はステレオ処理よりも演算量が大いので、2つの処理の同期をどうやってとるかが課題となるが、相互に結果を補完することができ、障害物検出処理の結果が向上することが期待される。さらに、物体の形状モデルやフローの分布モデルと照らし合わせることで物体認識へつなげられる可能性が高い。

2. ステレオ処理を前処理として行い、注目する距離においてのみ動画像処理を行う。動画像処理にける範囲を限定することで演算量の減少が期待される。ロボットが t 時間後に移動する先にのみ着目し、そこに存在する障害物に移動障害物があるのかを検出することで次の時刻に安全に移動できるかどうかを判別することができる。

3. 動画像処理を前処理とし、動画像処理の結果をうけてステレオ処理を行う。ステレオ処理をかける範囲を限定することで演算量の減少が期待される。空間中に存在する移動物体がその時刻でどの距離に位置するのかを検出することで、次の時刻に移動物体がどこへ移動するかを予測することが可能。

移動ロボットに搭載する際、大事なのは衝突を避けることであり、着目すべきなのは衝突危険性のある領域である。衝突危険性のある領域とは、観測者(ロボット)のすぐ近くに存在する物体および観測者に接近してくる物体の占める画像領域である。

ロボットの状態を、「静止中」「移動開始」「移動中」の 3 つにわけ、それぞれの場合について衝突危険性のある領域とそれを検出するために必要な処理を考えた。

静止中のとき、衝突する危険性があるのは移動する障害物のみである。よって、まずは動画像からオプティカルフローを抽出し、移動物体を検出する。移動物体として検出された領域に対してステレオ処理を行い、距離を測る。オプティカルフローから移動物体の相対的な速度は検出できるので、衝突危険性の有無を判定することができる。

移動を開始するときは状態としては静止状態であり、移動障害物に注目する必要があるが、これから移動する先に存在する静止障害物も検出することも重要である。そこで、ステレオ処理を行い、付近に存在する障害物を検出する。ステレオ処理を行うことによって、空間に存在する静止障害物の配置を知ることができる。

移動中は、静止障害物、移動障害物のいずれにも衝突する可能性がある。観測者が移動すると背景領域にもオプティカルフローを生じる。フローの大きさは物体と観測者の相対的な速度と距離に依存する。つまり、フローが大きな領域というのは物体がそばにあるか、早い速度で移動しているかのどちらかである。まず、フローの大きな領域を抽出し、ステレオ処理により距離を測ることで、接近した位置にある物体なのか早い速度で移動してくる物体なのかを判別することができる。

統合処理については、実験を行いながら、演算量と処理速度、実際に起こりうる様々な問題を含めて、今後も検討する必要がある。

4.5 ビジョンシステムの設計

ソフトウェア処理では、演算量の多い視覚処理をリアルタイムで行うことは非常に困難であるため、ハードウェア化が必要である。また、パソコンベースで処理を行うとシステム自体が大きくなってしまい、ロボットの一機能として視覚処理を搭載することが難しい。

ここではロボットに搭載するための小型で高速な処理が可能であるロボットビジョンシステムを実現することを目標とし、検討および設計を行った内容を示す。

4.5.1 機能概要

ビジョンシステムは、画像入力をうけるセンサ部と入力された画像から情報を抽出する視覚処理部の2つに分けることができる。本研究では、視覚処理部に実装するアルゴリズムはステレオビジョンとオプティカルフロー検出処理とし、2つの視覚機能を動的に切り替えながらリアルタイムで障害物検出を行うビジョンシステムの構築を目指す。

4.5.2 実現方式の検討

リアルタイム処理を実現するためには、視覚処理を並列化し、高速に処理を行うことが求められる。一般に、視覚処理で最も演算時間がかかるのが対応付け処理である。

対応付け処理は、ステレオビジョンでは視点の異なるカメラで同時刻に撮られた画像2枚の間で、オプティカルフロー検出では同一カメラで異なる時刻に撮られた画像2枚の間で行う。このとき、一枚の基準画像とそれに対応する参照画像を対応づける必要があるが、対応付け処理を行う間基準画像を保持しておく必要がある。さらに、処理を並列化するためには、保持した画像に対して複数の処理ブロックからアクセスできることが必要である。また、全体の処理時間を短縮するために、画像データへのアクセス時間を短縮することも必要である。

対応付け処理を実装するハードウェアとしては、DSP、FPGA、専用のプロセッサなどが考えられる。視覚処理には、画像を保持するメモリブロックが必要であり、その構成方法が処理速度に大きな影響を与える。メモリブロックをチップ外に設けるとすると、データをやりとりするスピードが落ちてしまい、処理速度も影響を受けてスピードダウンしてしまう。よって、メモリはチップに内蔵することが望ましい。内蔵するメモリ領域は入力される画像サイズに依存するが、8bit グレースケール画像を2frame分と1bit フラグ情報を1frame分保持する必要があるが、CIFサイズ(352x288pixel)ではおよそ1.8Mbit必要である。既存のDSPやFPGAは内蔵メモリを持っているが、現在市販されているものでは1.8Mbitのメモリを内蔵できるものはなく、画像を分割して保持し、数チップで情報をやりとりする必要がある。また、現在発表されているFPGAの中には内蔵メモリサイズが2Mbit以上のものが存在するが、FPGAやDSPなどのメモリブロックの構成には制限があり、処理に対して最適な配置を行うことはできない。また、一般にFPGAやDSPは消費電力が高いという欠点もある。

そこで、メモリ内蔵型の専用プロセッサを開発し、ビジョンシステムに用いることとした。開発するチップ内のメモリ配置は処理にあわせて最適化し、リアルタイム処理、低消費電力化を狙う。また、開発するチップの処理する画像サイズはCIFサイズとする。これはチップサイズの制限から決めた。構成としては、チップサイズをより大きくすれば、同様の構成でより大きな画像サイズを取り扱えるようにする。

4.5.3 システムの構成

ビジョンシステムの構成図を図11に示す。

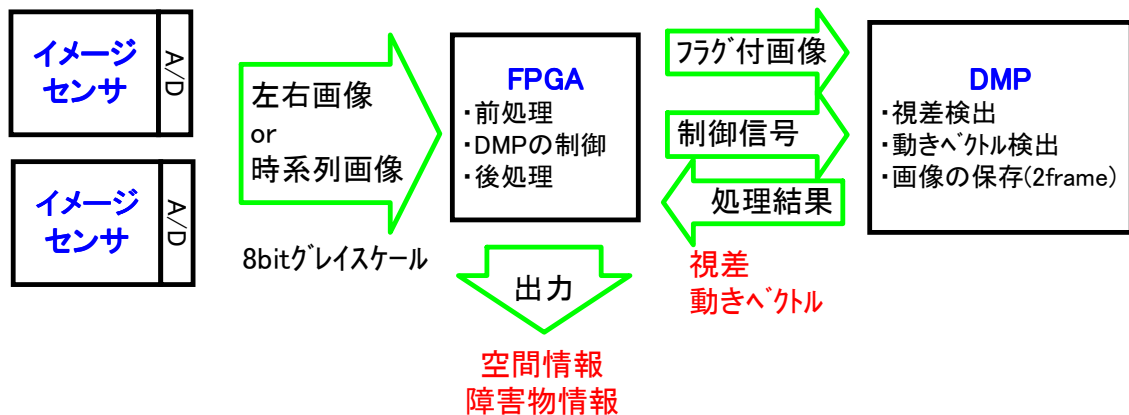


図 1 1 ビジョンシステムの構成

システムは画像センサ、制御チップ、開発するプロセッサ(DMP)の3チップで構成する。開発するプロセッサは、動画像処理およびステレオ処理で演算量の大部分を占める対応付け処理を行うメモリ内蔵型のマッチングプロセッサである。制御部分をFPGAに実装とすることで動画像処理とステレオ処理の統合方法を自由に変えることが可能となり、より柔軟な処理を実現する。また、エッジ検出処理、および対応付け結果を処理する部分はFPGAにおくこととする。エッジ検出方法や対応付け結果から実際に必要な結果を抽出する手法には様々なものがあり、環境によって処理内容を変更する必要があることが予想されるからである。

システムは、画像入力を受けると、FPGAによりエッジ検出を行い、入力画像(8bit, グレイスケール)にフラグ情報を付加した情報をマッチングプロセッサDMPへ送る。DMPは、フラグ情報をもとに指示された対応付け(ステレオビジョンでは1次元探索、動画像処理では2次元探索)を行い、対応付け結果をFPGAに与える。FPGAは対応付け結果を空間に存在する障害物の位置情報や移動情報に変換して出力する。一連の動作をビデオレート(30frame/sec)で処理を行うことを目標とする。

4.6 マッチングプロセッサの開発

視覚処理ではマッチングにかかる時間が全体の処理時間の大部分を占める。そこで、対応付け処理を高速に行うマッチングプロセッサの開発を行った。ここではそのマッチングプロセッサの機能やアーキテクチャについて述べる。

4.6.1 機能概略

マッチングプロセッサ(DMP)の概略を以下に示す。

- ・ 2画像間で対応付け処理を行う。
- ・ 対応付けの評価関数は重み付き差分絶対値和(SAD)とする。
- ・ 評価に用いる小領域のサイズは1x1[pixel]から15x15[pixel]まで可変とする
- ・ 対応点探索範囲は水平方向を32の倍数で垂直方向は8の倍数で可変とする
- ・ 対応付けに用いる画像情報2frame分とフラグ情報1frame分を保持するメモリを内蔵

・画像サイズはCIF(352x288[pixel])サイズとする

対応点探索範囲を1次元方向または2次元方向で選択可能な構造として、ステレオビジョンにおける対応付け処理とオプティカルフロー検出における対応付け処理を実現する。また、画像全体に一律に処理をかけるのではなく、画像のピクセル毎に処理を選択できるようにするため基準画像に対してフラグを設ける。対応付け処理はフラグがたっているピクセルに関してのみ行われる。たとえばエッジ検出処理によって求められたエッジ点についてフラグをたてておけば、エッジ点でのみ対応付け処理を行い、各エッジ点における距離や動きベクトルを求めることが可能である。対応付け処理を行う際、模様のない面のような領域は一般に対応付け結果に曖昧さが残り正しく対応付けをとることが困難である。そのような点を排除するためにフラグをマスクとして用いることができる。

また、対応付けに用いる小領域のサイズはロボットのおかれた環境によって最適な値が異なる。様々な環境で用いることが可能となるよう、ブロックサイズは可変とする。ブロックサイズを可変とすることで、対応付けの曖昧さを取り除くことができる場合もある。

画像メモリは8bit グレースケールを2frame分とフラグビット1bitを1frame分用意する。メモリはデュアルポートとし、分割配置して、メモリブロックを形成して配置する。これにより外部からアクセスできるポート数を増やし、並列度を高める。

4.6.2 アーキテクチャ

(a) 全体

DMP チップ全体のブロック図を図12に示す。DMP チップは、

- ・チップ外部とチップ内部のアドレス変換やコマンドのデコードなどを行うインターフェース部
- ・基準画像とフラグ情報を保持するメモリとそれらの読み出し制御回路を含む基準メモリブロック
- ・参照画像を保持するメモリとそれらの読み出し制御回路、および評価値演算回路、一次元(水平方向)最小値検出回路を含む参照メモリブロック
- ・基準画像と参照画像間のアドレスおよびデータの流れを制御するデータフロー制御部
- ・1次元対応付け結果から、さらに最小値を検出する2次元探索用最小値検出回路ブロック

以上の5つのブロックから構成される。

入出力データを以下に示す。

【入力データ】

チップセレクト，処理選択信号，画像データ，フラグデータ，ブロックサイズ，探索範囲

【出力データ】

処理ピクセルのアドレス，視差値，動きベクトル，内部状態信号

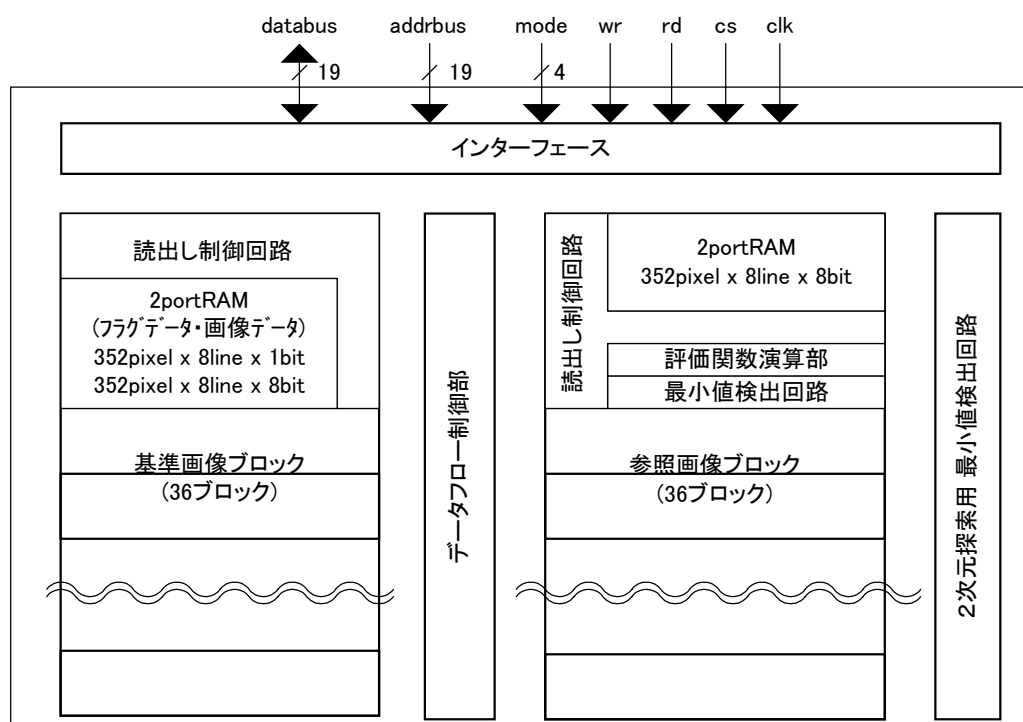


図 1 2 DMP チップ全体ブロック図

基準画像メモリおよび参照画像メモリそれぞれ 8line ごとに 1 つの処理ブロックを形成する。参照メモリブロック内には参照画像 8line ごとにシフトレジスタ、評価関数演算回路、最小値検出回路を配置し、対応付け処理を行う単位ブロックとしている。

ステレオ処理のような 1 次元探索を行う際には、基準画像 1 ブロックと参照画像 1 ブロックを 1 対 1 に対応させて処理を行う。基準画像ブロック、参照画像ブロックはそれぞれ 36 ブロックあるので、36 の対応付け処理が並列的に行われることになる。また、オプティカルフロー抽出のような 2 次元探索を行う際には、基準画像 1 ブロックに対して参照画像ブロックを複数割り当てる。つまり、複数の参照画像ブロックに同一の基準画像を与え、各ブロックで 1 次元の対応付け処理を行い、その結果からさらに最小値を検出することで 2 次元探索における対応付け処理を実現する。探索範囲を縦方向に 64pixel とすると 4 つの 2 次元探索処理が並列で動作する。

(b) 基準画像ブロック

基準画像ブロックは、フラグと基準画像の濃淡値データを保持するメモリブロックとそれらの読み出し制御回路が含まれる。フラグメモリは 1port、基準画像メモリは 2port メモリである。いずれも 352pixel x 8line 分の情報をもつ。

処理の種類としては、外部からメモリブロックへの読み出しと書き込み処理、フラグ探索によるデータの読み出し処理がある。フラグ探索によるデータの読み出し処理は、対応付け処理を行うピクセルを選択的に読み出すための機能である。フラグ位置はマッチング

に用いる小領域の中心画素にあたるため、フラグの位置によっては自分のブロックが持つ以外の画像領域にアクセスする必要がある。そのため、画像メモリの port のうち 1 つは、自身のフラグ探索読み出し用に用いることとし、もう 1 つは他のブロックからの読み出し要求に用いることとする。読み出し要求が複数のブロックから発生した場合には、最初に読み出し要求を出したブロックの読み出しを優先させる。読み出し要求の調停や認可は、読み出し要求調停用の回路で行う。

(c) 参照画像ブロック

参照画像ブロックは、参照画像 352pixel x 8line 分のデータを保持するメモリと基準画像ブロックから出力されるフラグピクセルのアドレスをもとに対応点の候補領域のアドレスを生成して読み出しを行う制御回路から成る。

参照画像メモリは 1 つの基準画像に対する対応付け処理を並列で行えるようにポート数を稼ぐため、352pixel x 8line 分のデータを分割して保持する。各メモリブロックは 2port メモリで構成する。

対応付け処理を行う際は、基準画像ブロックから出力される基準画像データとそのアドレスをうけて、対応点探索範囲とマッチングウィンドウサイズから読み出す画像領域のアドレスを求め、参照画像メモリから読み出す。基準画像ブロックと同様に読み出し要求を調停するための回路ブロックを持つ。

(d) 対応付け処理部

参照画像から読み出された値と基準画像から読み出された値の評価関数を演算する。1 つのブロック内に PE は探索範囲の pixel 数(32)個おき、各 PE が視差 0 から 31 までの評価値を並列に演算する。

評価関数は基準画像と参照画像内の候補領域でそれぞれ対応するピクセルの差分絶対値和である。ピクセル(i, j)について、視差 $disparity$ 離れた位置における評価値は式(1)で示される。

$$E(i, j, disparity) = \sum_{m=-hwin}^{hwin} \sum_{n=-hwin}^{hwin} I_{reference}(i+m, j+n) - I_{candidate}(i+m, j+n+disparity) \quad (1)$$

評価関数は、ピクセルごとの差分絶対値を演算、その結果を重み付けし、マッチングブロックに含まれるピクセル分累算して求める。ピクセルシリアルに評価関数を演算するので、この演算に必要な時間は $(T_{read} + T_{add}) \times hwin$ である。

演算された評価値は、最上位ビットから順に比較が行われ、最小値を検出する。1 つの対応付け処理が終わると、次の入力をまち、基準ブロック内の全ピクセルについて対応付け処理が終わるまで繰り返し動作する。

(e) 2次元最小値検出部

二次元探索は一次元探索で求められた各ブロックでの評価値が最小となる点から、さらに最小値を検出することで行う。二次元方向の探索範囲が可変であるため、値を比較する

範囲もそれにあわせて変化する。そのため、最小値検出回路への入力が可変でなくてはならない。そこで、2次元最小値検出回路は隣接結合を持つビット比較器で構成することとした。設定された探索範囲によって隣接結合のON,OFFを切り替えることで任意の長さの最小値検出回路を構成する。

(f) データフロー制御部

一次元対応付け時は左右に並んだ基準画像ブロックと参照画像ブロックを一対一に対応づけ、2次元対応付け時には探索範囲に応じて一対多に対応づけるための、スイッチ群である。設定された探索範囲によってスイッチのON,OFFを切り替える。このブロックの制御信号は二次元最小値検出部にも与えられる。

(g) インターフェース部

このブロックは、DMPチップ内部と外部の信号を変換する働きを持つ。
 チップ内部はメモリが分割は位置されているために、外部から見えるアドレスとチップ内部でのアドレスは異なっている。そのため、画像全体でのピクセルアドレスと各ブロック内でのピクセルアドレスを変換する。また、入出力されるデータ転送レートの変換や、与えられたコマンドをデコードする機能を持つ。

4.7 今後の予定

今後は引き続きロボットビジョンシステム、およびそれに用いるマッチングプロセッサの論理設計を行い、FPGAを用いたエミュレータで設計検証を行う。さらに、このエミュレータを用いて、デモシステムを試作し、実環境での動作性能を評価、改良、ロボットビジョンシステムの設計資産として完成させる。本プロジェクトで計画していたマッチングプロセッサのチップ試作はこの設計資産を用いて、次期の実用化プロジェクトで実施することとする。また、移動ロボットにカメラを搭載して実験をより詳しく行い、アルゴリズムの改良を行う予定である。

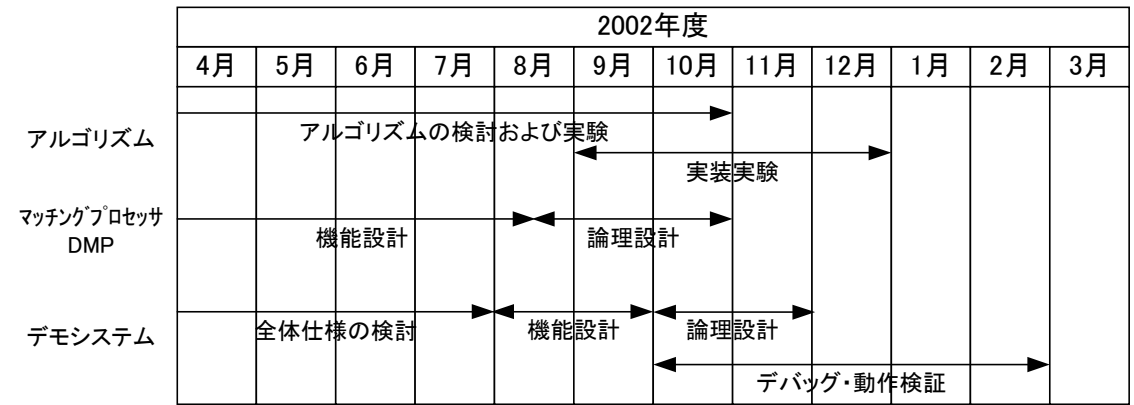


図 1 3 2002 年度計画線表