

# ステレオカメラによる 高分解能距離センサシステム

High Resolution Range Sensor System with Stereo Camera

青山千秋\* 峯田憲一\*\* 岡田泰仕\*  
Chiaki AOYAMA Ken'ichi MINETA Yasushi OKADA

## 要 旨

自動車用障害物検知システムのために、ステレオカメラを使うことで障害物となりそうな複数の物体までの相対距離、速度と物体の幅を検知するシステムを構築した。専用の相関演算ハードウェアと高速なアルゴリズムの開発により実時間で検知できることが実環境において検証された。本論文においては、ハードウェアとソフトウェアの解説とシステムを実行した結果について述べる。

## ABSTRACT

To apply to the obstacle detecting system for automobile, we have developed the system that makes detection of the relative distance from the vehicle to the obstacle which will have possibility of collision, the relative speed of the vehicle and the size of obstacle using a stereo camera. The exclusive correlative operation hardware and the high-speed algorithm are prepared, and the system has been proved for detection of obstacles at real time by testing in real scenes.

In this paper, we show the explanation for the hardware and the software of the system and the results of experiments.

## 1. まえがき

近年のエレクトロニクス技術の進歩により従来以上に自動車におけるインテリジェント化<sup>1)2)</sup>が進み、運転がより安全で、かつ、快適なものになってきている。さらに、レーザレーダにより追従走行するシステムも実用化され、道路にひかれた白線を検知し自動車がレーンを逸脱しないシステムに関しても実用化が近づいている。将来的には自動車が走行する車線(以下、自車線と称す)内に停止している車両を遠方で検出し、減速するようなシステムに発展していくと考えられている。ところが、遠方になるほど画像による白線の位置検知結果とレーザレーダなどによる障害物の位置検知結果との違いが大きくなり、障害物が自車線の内側にあるのか外側にあるのかという判定が困難となってくる。そこで、空間分解能の高い画像を使った単一のセンサにより車線と障害物を検知することで、センサの違いによる誤差を無くし、内外判定を確実にを行う研究が進められ

ている。そして、画像による精度の高い障害物検知が自動車用障害物検知システム構築の課題となっている。ところが、ステレオカメラを利用した障害物検知は報告されているが、画像を使うことで高い分解能を得られるにも関わらず水平方向に対する距離を検出する点数が少ないため検出密度が低く高分解能という画像の特徴が生かせていなかった<sup>3)4)</sup>。

そこで、我々はステレオカメラを使って前走車までの距離や速度および幅をリアルタイムで検出できるセンサを開発した。そして、実験車システムによる実走行テストにより、一つのセンサにより走行領域と障害物の相対位置誤差が低減すること、検出密度が向上すること、障害物間のすり抜けや飛び出しの検知への可能性もあることがわかった。

本稿では、高分解能距離センサのハードウェアとソフトウェアについて解説し、実験結果について報告する。

\*和光基礎技術研究センター

\*\*栃木研究所

## 2. システム概要

本システムは単に画像処理を計算機で行うだけでなく、人間との協調をはかるシステムのために実時間での処理を行うことを要求された。そこで、高速化とレンズのひずみの少ない領域を使うために処理領域を水平面だけとし、さらに一部を専用ハードウェア化することでリアルタイムでの処理を可能とした。

実験車システムにおける処理の流れをFig.1に示す。ステレオカメラより取り込んだ画像について距離とオプティカルフロー(動き)を求め、クラスタリングし、クラスタにIDを付けて、結果を出力する。画像から得られる動きは精度が悪いためIDを追跡しフィルタを通すことで速度情報とする。

このシステムのハードウェアは、ステレオカメラ、相関演算ボード、CPUボード、A/D変換ボード、ジャイロ、車高センサより構成されている。既存のハードウェアとの親和性の点からシステムバスにはVMEバスを採用した。VMEバスは画像データを転送するには十分な帯域を持っていないため、水平面の視差情報に圧縮してCPUボードに送っている。

### 2. 1. ステレオカメラ

ステレオ画像から左右の良い対応をとるためには、左右のカメラの特性が合っていることが必要で、特に本研究で利用している相関演算は、左右の明るさの違いに敏感であるため映像の明るさが同じになるように環境の明るさに応じてカメラの絞リや感度を調整しなくてはならない。ところが絞リは機械的動作により働くため絞リ値を一致させるのは困難である。そこで絞りの代わりに切り替えられるNDフィルタと電子シャッタを利用することにし、再現性良く明るさの調整ができるようにした。

この白黒カメラには10bitのA/Dコンバータが附属しており768×480画素のインタレース画像が毎秒30枚出力される。ただし、この画像のうち処理に使うのは768×240の片フレームだけである。また、画角は約40度に設定している。

### 2. 2. 相関演算ハードウェア

ステレオ画像における左右の画像の対応をとり、動画像における時間の異なる画像の対応をとるために専用のLSIを作ることで2UサイズのVMEボード2スロットとな

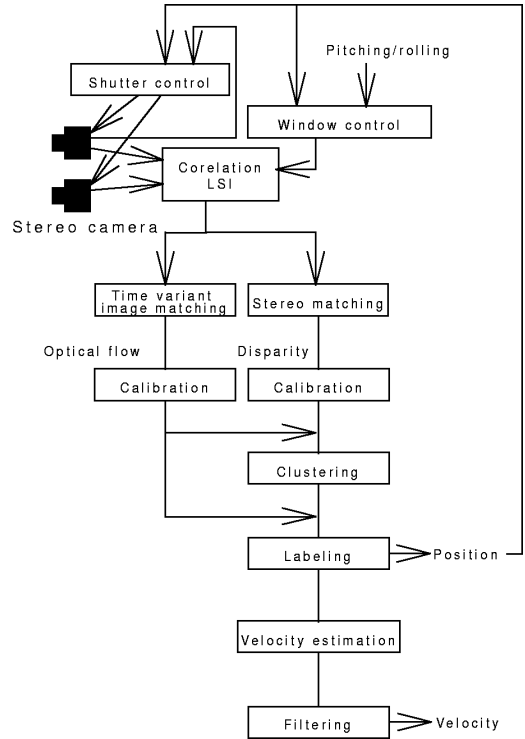


Fig.1 Process flow

り、また、リアルタイムで処理が行えるようになった。

画像の対応点は小領域の相関により求めた。相関計算にはパイプライン処理を可能とするために、左画像と右画像のある量だけずれた対応する画素の明るさ(右の明るさR、左の明るさL)について差の絶対値(以下では一致度と称す)の和Dを計算して利用することにした。そして、これを探索領域(横ずらし量 $dx_{min} \sim dx_{max}$ 、縦ずらし量 $dy_{min} \sim dy_{max}$ )について演算し、Dが最小となる場所を対応点とするアルゴリズムを採用した。

ここで、ある点(x, y)についてのずれ(dx, dy)における相関値は、相関をとる領域の幅がn、高さがm の時以下のように定義する。

$$D_{x,y,dx,dy} = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |R_{x+j,y+i} - L_{x+j+dx,y+i+dy}|$$

対応位置はdxを $dx_{min}$ から $dx_{max}$ まで変え、dyを $dy_{min}$ から $dy_{max}$ まで変えて計算し、最もDの値が小さくなる時の位置(dx, dy)とする。探索は右側の画像から取り出したある小領域について左側の画像で探索するものとする。

このアルゴリズムは自動車の後部などのようなカメラに対して垂直な平面に対しては有効な方法である。また、実際に処理を行った結果から、領域内で距離が変

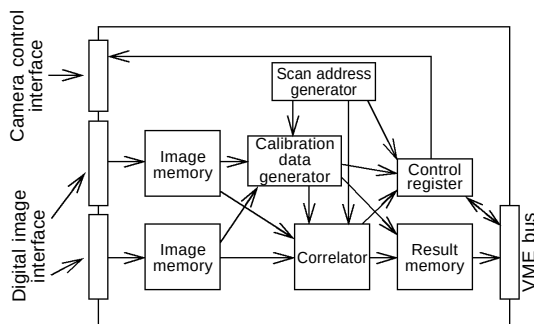


Fig.2 Correlation board

わっている面を持つ自動車の側面や生け垣に対しても対応がとれることがわかった。

相関演算ハードウェアはFig.2の様に画像メモリ、スキャン座標発生器、補正データ発生器、相関演算器、処理結果メモリ、制御レジスタに分かれる。ステレオ処理のためには画像メモリを左右で2枚ずつ使い、動画像処理のためには現在の画像の取り込みに2枚、過去の画像の取り込みに最大8枚メモリが使用できる。つまり、動画像の相関は最大で6枚前の画像と比較ができ33msec×6=198msecの時間差までとれる。

スキャン座標発生器と補正データ発生器の動作により上記の添え字の変化を発生させている。レンズのひずみによる画像の変形をもとの変形していない形に戻す動作を補正データ発生器で発生するメモリアドレスにより行う。

補正データ発生器はレンズの周辺光量低下による信号レベルの低下を補正するデータも発生し相関演算器に供給し乗算により補正させる。

処理結果メモリには相関値と、最小値、最小値のあるアドレスが保存される。

スキャン座標発生器は探索対象領域(ステレオの場合最大横768×縦18画素)の画素データを縦→横の順(dxとdyの添え字のうちdyを先に変化させる順)にメモリから相関演算器に渡すように座標を発生させる。

### 2.3. 相関演算器

従来の相関演算器は対応をとる領域のサイズが小さく(4×4、8×8)、そのため、ノイズに弱いという問題があった。ある実画像を右画像としそれを右方向にシフトさせて左画像とした2枚の画像について、相関をとっ

Table1 Variance of disparity

|          |    | Horizontal |      |      |      |      |      |      |      |
|----------|----|------------|------|------|------|------|------|------|------|
|          |    | 1          | 3    | 5    | 7    | 11   | 15   | 21   | 31   |
| Vertical | 1  | 4.73       | 2.49 | 1.67 | 1.27 | 0.99 | 0.91 | 0.79 | 0.56 |
|          | 3  | 2.46       | 0.97 | 0.86 | 0.86 | 0.67 | 0.56 | 0.09 | 0.08 |
|          | 5  | 1.54       | 0.51 | 0.50 | 0.49 | 0.40 | 0.10 | 0.10 | 0.09 |
|          | 7  | 1.22       | 0.18 | 0.14 | 0.12 | 0.10 | 0.10 | 0.10 | 0.09 |
|          | 11 | 0.59       | 0.20 | 0.12 | 0.11 | 0.09 | 0.09 | 0.09 | 0.09 |
|          | 15 | 0.49       | 0.16 | 0.12 | 0.10 | 0.09 | 0.09 | 0.09 | 0.09 |
|          | 21 | 0.59       | 0.20 | 0.12 | 0.10 | 0.09 | 0.09 | 0.09 | 0.09 |
|          | 31 | 0.28       | 0.21 | 0.11 | 0.09 | 0.09 | 0.08 | 0.08 | 0.08 |

た時のばらつき(Table 1)をみると、5×5画素では、領域ごとのずれの推定量に±0.5画素ぐらいのばらつきがあることがわかる。本システムでは±0.2画素以下の精度が要求されたので、領域サイズを十分大きくする必要があった。そこで、領域サイズは縦横ともそれぞれ最大15まで設定できるようにした。

相関演算器(Fig.3)は、四つの演算器による四つの演算ブロックを有する16並列のパイプライン処理方式を採用した。探索領域 $dx_{min} \sim dx_{max}$ 、 $dy_{min} \sim dy_{max}$ はステレオ視で横方向に0から127、縦方向に-2から+1、動画像で横方向に-8から+7、縦方向に-4から+3取ることが可能である。ステレオ視でも縦方向の探索を可能としたのは左右のカメラの経年変化や振動によるずれを吸収するためである。一つの演算器は処理領域の16分の1を担当するので、ずれ量を16分の1ずつ受け持つことにして、具体的には第一相関器は $dy=0$ 、 $dx=0, 4, 8, \dots, 127$ とずれ量が変わる探索領域を求める計算を受け持っている。

第一演算ブロックの第一相関器について説明する。

a-dの回路は、前の式における

$$f(j) = \sum_{i=0}^{n-1} |R_{x+j, y+1-Lx+j+dx, y+i+dy}|$$

を計算する。

e-hの回路は

$$D_{x=0, y=0, dx=0, dy=0} = \sum_{j=0}^{n-1} f(j)$$

の計算を行う

つまりa-hの回路により $m \times n$ のウィンドの相関処理ができることになる。

つぎに探索元の右側の画像を1画素横にずらした $x=1$ 、 $y=0$ における次の $m \times n$ のウィンドの相関値 $D_{x=1, y=0, dx=0, dy=0}$ の計算を考える。

$x=1$ の場合は、すでに計算されている $x=0$ の相関値と、 $x=0$ の領域の1列目と $x=1$ の領域の4列目が

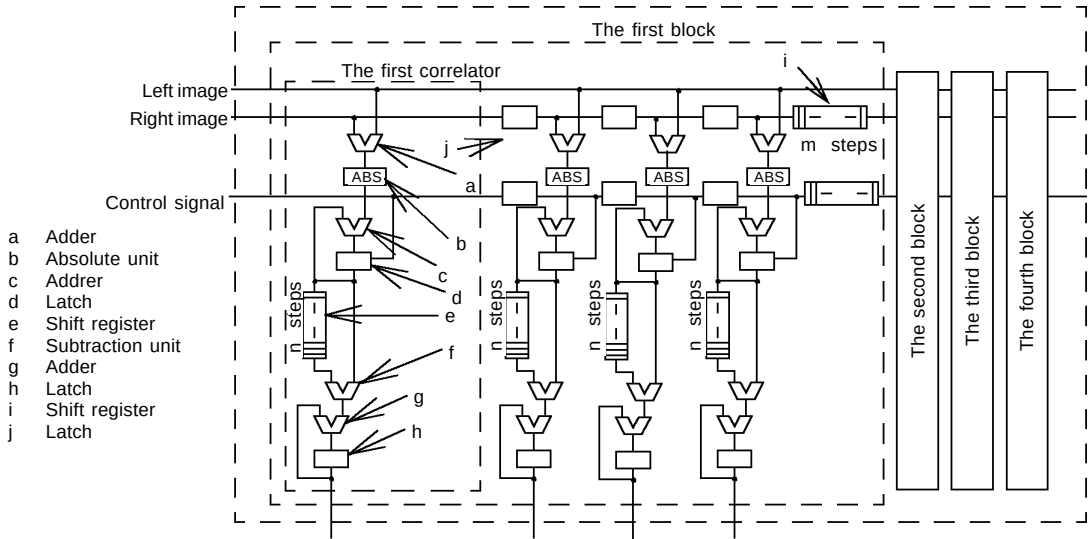


Fig.3 Correlator chip block diagram

違うだけである。そこで、1列目の値を引いて4列目の値を加えるだけで $x = 1$ の場合の領域の和になる。一方1列目の値はシフトレジスタeの出力になっているので $x = 1$ の時の4列目の和から減算器fにより引いて、その値をラッチhの出力と加算機gにより加えればよい。これで残り3列の演算をすることなく $x = 1$ の場合の領域の和が算出できる。

$x$ を0から639まで変化させることで同一のずれ量を持つ640の領域のDの値が算出される。

同様の処理をラッチjにより発生する縦のずれが発生した画像について行うことで、第一ブロック全体としては $dx = 0$ 、 $dy = 0 \sim 3$ の相関処理ができる。

第二ブロックはm段のシフトレジスタiにより画像データを横1画素分ずらすことになるので、 $dx = 1$ 、 $dy = 0 \sim 3$ の相関処理を行うことになる。従って第一～第四ブロック全体では $dx = 0 \sim 3$ 、 $dy = 0 \sim 3$ の相関処理を実行することになる。

この様な処理により領域のサイズが $15 \times 15$ の時に、領域を水平方向に $x = 0 \sim 639$ まで移動し、ずれ量 $dx$ を $dx = 128$ までの各一致度Dを求める際でも、演算回数を $15 \times 15 \times 640 \times 128$ 回から $15 \times 640 \times 128$ 回に減らすことができる。すなわち、小領域の横方向の幅(画素素数)は計算時間に無関係になる。従来はこの様な処理をしていなかったため実時間処理においては時間的制約から領域サイズを大きくすることができなかった。

つぎに、こうして得られた相関値を用いて、それぞれ

の点 $(x, y)$ に対応する最小値 $D_{\min}$ とその時のずれ量 $dx$ 、 $dy$ を求め、処理結果メモリに保存する。

## 2. 4. 補間／信頼度計算

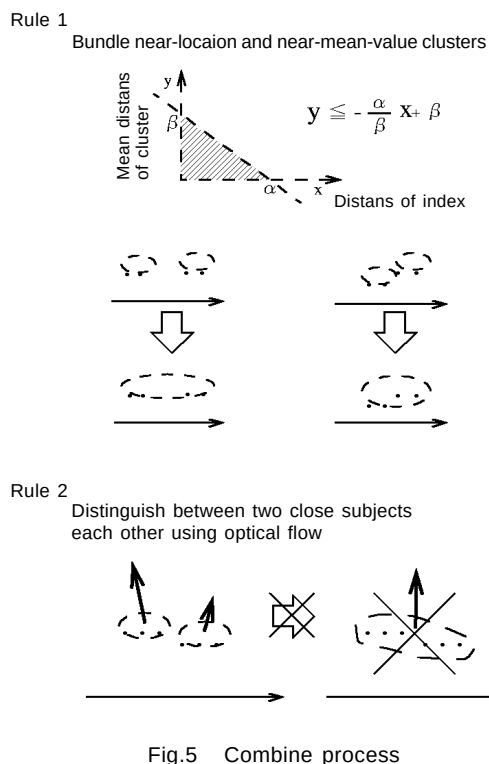
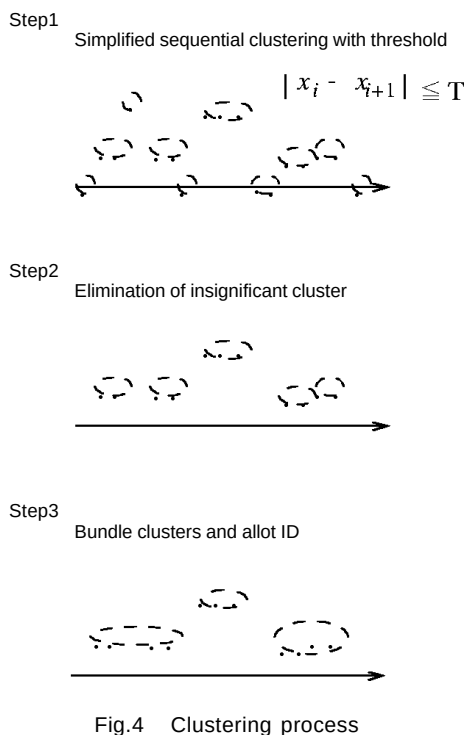
最小値からだけでは1画素単位でのずれしか算出できないため、さらに補間処理をすることにより対応点の分解能を4倍以上向上させている。最小位置 $dx$ は、最小値となるずれ量 $dx_{\min}$ 、最小値 $D_{\min}$ 、そして、その隣の値 $D_{\min-1}$ 、 $D_{\min+1}$ から比例計算により算出している。 $D_{\min-1} < D_{\min+1}$ の場合は以下ようになる。

$$dx = dx_{\min} - \frac{(D_{\min-1} - D_{\min+1})}{2 \cdot (D_{\min} - D_{\min+1})}$$

以上の演算を機械的に行うと、すべての点について対応計算結果が得られるが、画像の性質によってはその結果が正しいとは限らない。そこで確からしさ(信頼度)を求めて、それと距離情報を後段の処理に送るようにする。信頼度としては最小値の谷の周囲に対する相対的深さと、一致度の和の大きさを利用している。信頼度は対応をとるべき領域内のコントラストが低いと最小値の周囲の値が変化がなだらかなるため低いものとなる。

## 2. 5. クラスタリング

つぎに、相関演算結果より得られた視差情報を三角測量の原理により距離に変換し、実平面上の点情報を複数得る。そして、これらの点をそれぞれの近さによりグループ分けする。このグループのことをクラスタ、クラ



スタを作ることクラスタリングという。本研究では以下のような手順によりクラスタリングを行った。(Fig.4)  
Step1: 視差情報をスキャンしてしきい値によりクラスタリングする。

Step2: 無意味なクラスタを除去する。

Step3: クラスタを束ねる。

クラスタを束ねる際には以下の基準を適用する。

(Fig.5)

Rule1: 位置が近く、平均値が近いクラスタは束ねる。

Rule2: オプティカルフローが異なるクラスタは束ねない。

スキャンによりクラスタリングしており、収束計算をしないため短時間で処理を終えている。

## 2.6. ID付け／継承

つぎに、クラスタリングによって得られたクラスタ(検出物体)が、過去の時点で認識していた物体(記憶物)のうちどれに対応するのか調べ物体のIDを付ける。これをID継承と呼ぶ。対応を調べる際には、検出物体と記憶物体の相対位置のみでなく、記憶物体の過去の動きも利

用する。こうすることで複数の物体が交差して動く場合(たとえば、3車線道路の中央を走行しているときに左右を走行する車が前方で交差しながら車線変更をするような場合)でも正しくIDの継承ができるようにした。

記憶物体の過去の動きから現在の位置を予測し、検出物体との距離のみによって対応をとると、記憶物体の過去の速度ベクトルに対して大きく外れた方向に検出物体があっても、速度ベクトルの方向にある検出物体と同じ距離ならば同じ評価値になってしまい、誤対応を起こす可能性がある。つまり、位置情報のみで検出物体と記憶物体の対応をとると物体同士が近づくとIDの入れ替えが起こりうる。また、記憶物体に優先順位を付け順位の高いものから探索を行う場合、順位の高い物体が順位の低い物体の近くで、ノイズなどの影響で一時的に消えると、IDを間違った検出物体に付けてしまうという現象が起こる。

そこで、一回前の記憶物体の位置と検出物体の位置で決まるベクトルを候補ベクトルと呼ぶことにし、この候補ベクトルと記憶物体の速度ベクトルとの類似性を評価して対応の判定をする方法を導入した。類似性の評価は

|                                          |                  |                                                    |                            |
|------------------------------------------|------------------|----------------------------------------------------|----------------------------|
| $P_0$ : $(x_0, y_0)$                     | Past position    | $\hat{P}_0$ : $(x_{p0}^{\wedge}, y_{p0}^{\wedge})$ | Estimated position         |
| $\vec{V}$ : $(x_{\vec{V}}, y_{\vec{V}})$ | Past velocity    | $d$ :                                              | Aberration distance        |
| $P$ : $(x, y)$                           | Detect position  | $\theta$ :                                         | Translation angle          |
| $\vec{P}$ : $(x_{\vec{P}}, y_{\vec{P}})$ | Candidate vector | $k$ :                                              | Constant                   |
|                                          |                  | $C$ :                                              | Coefficient of correlation |



$$\begin{aligned}
 (1) \quad x_{p0}^{\wedge} &= x_0 + x_{\vec{V}} \times \Delta t, \quad y_{p0}^{\wedge} = y_0 + y_{\vec{V}} \times \Delta t & (4) \quad d &= \sqrt{(x - x_{p0}^{\wedge})^2 + (y - y_{p0}^{\wedge})^2} \\
 (2) \quad x_{\vec{P}} &= x - x_0, \quad y_{\vec{P}} = y - y_0 & (5) \quad C &= -(k\theta + d) \\
 (3) \quad \theta &= \arccos\left(\frac{\vec{V} \cdot \vec{P}}{|\vec{V}| |\vec{P}|}\right) = \arccos\left(\frac{x_{\vec{V}} x_{\vec{P}} + y_{\vec{V}} y_{\vec{P}}}{\sqrt{x_{\vec{V}}^2 + y_{\vec{V}}^2} \sqrt{x_{\vec{P}}^2 + y_{\vec{P}}^2}}\right)
 \end{aligned}$$

Fig.6 Evaluated value of ID

以下の手順で行う。(Fig.6)

- (1) 記憶物体の位置 $P_0$ と速度 $\vec{V}$ から、予測位置 $\hat{P}_0$ を求める。
- (2) 記憶物体の位置 $P_0$ と検出物体の位置 $P$ から、候補ベクトル $\vec{P}$ を求める。
- (3) 記憶物体の速度 $\vec{V}$ と候補ベクトル $\vec{P}$ から、候補角 $\theta$ を求める。
- (4) 予測位置 $\hat{P}_0$ と検出物体の位置 $P$ から、ずれ距離 $d$ を求める。
- (5) 候補角 $\theta$ とずれ距離 $d$ から、対応度 $C$ を求める。

この評価値が高いものにIDを引き継ぐようにした。

ところが、この評価を単純にすべての物体に対して素直に行うアルゴリズムは $O(n \log n)$ のオーダーで処理時間がかかる。これでは $n=30$ 以上で十分な処理速度が得られない。そこで、物体の近傍のみを探索することで計算量を大幅に減少させ処理の高速化を図るようにした。具体的には、

- (1) 探索平面を格子状のブロックに分割し、物体が属するブロックの近傍のブロックにあるクラスタに対してのみ距離計算をする。
- (2) 物体の属性として優先度を設け、優先度の高い物体から順に探索を行っていく。一度物体と対応がとれたクラスタはマップから削除する。

この方法により二つの物体が一つのクラスタと対応した場合を考慮する必要がなくなる。処理時間を計測するとFig.7の様にクラスタが40ある時には約30msecかかっていたものが5msec以下になり、フレームレートでの処理が可能となった。

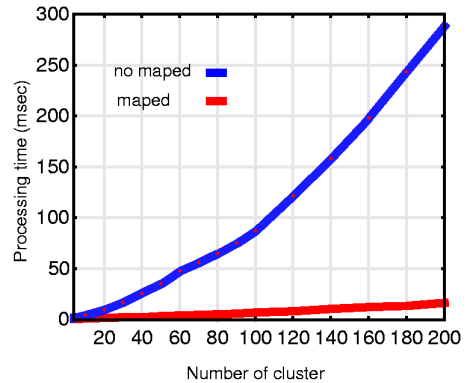


Fig.7 Time of inheritance

## 2.7. ピッチ補正

距離を求める領域を水平面に限定したため、水平面を推定する必要がある。そこで今回は車高センサ、ピッチレートセンサ、ヨーレートセンサにより水平方向を推定することにした。(Fig.8)

ピッチレートセンサを時間的に積分することで車両のピッチ角が原理的には求まるがレートセンサのドリフトにより時間とともに推定水平面が動いてしまう。それを防ぐために長期間的には車両は水平面を向いているものとしてレートセンサの積分値が0になるようにした。

こうすると定常加速の場合は車両が上を向いているにもかかわらず処理領域が車両に対して平行な方向に、つまり、上に向いていってしまう。そこで、車高センサにより定常的傾きを補正するようにした。

車高センサだけでは路面の凹凸による傾斜を反映して

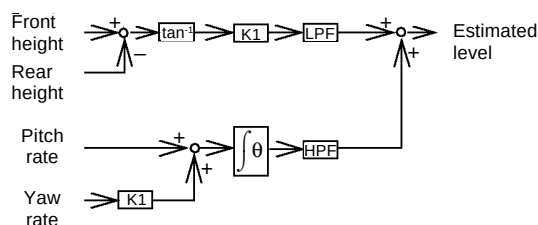


Fig.8 Pitch compensation

しまうため正しい水平面を推定できない。それを避けるためにローパスフィルタを通すと今度は急加速、急減速によるピッチ角変動に追従できなくなり、両センサの結果を利用しなければいけない。ところが、車両が旋回運動をするときは、ロールが発生しピッチ変動がなくてもピッチレートが発生したかのような結果になってしまう。そこで旋回運動時のヨーレートによりピッチレートセンサの出力を打ち消すことにした。

### 3. 実験結果

#### 3. 1. 距離精度

屋内でテストチャートを提示した場合の測定結果をFig.9に示す。黒い点が画素ごとに検出された位置で、下の枠の中にはクラスタリング結果として検出物体のID、位置、幅を示す。カメラの前には幅が435mm (1番)、215mm (2番)のテストチャートと直径が50mmの黒いパイプ (3番)があり、青い線で実測値を示している。この時の基線長は100mm、画角は約40度、水平画素数は768である。

キャリブレーションを特にしていないが位置や幅がほぼ正しく算出できていることがわかる。パイプの幅が細くなっているのはパイプが黒いためよく映っていないためである。

実験結果より単に重心位置が算出できるだけでなく物体の幅までわかるため、従来のミリ波レーダやリフレクタを検知するレーザレーダでは難しかった障害物間のすり抜け判断もできる可能性があることがわかる。

#### 3. 2. 速度精度

X-Yテーブルを利用して3. 1. と同じ設定のカメラに対して左右にターゲットを移動した場合 (Fig.10) と前後に移動した場合 (Fig.11) についての測定結果を示す。速度誤差はカメラの前方約2mの位置で前後方向で

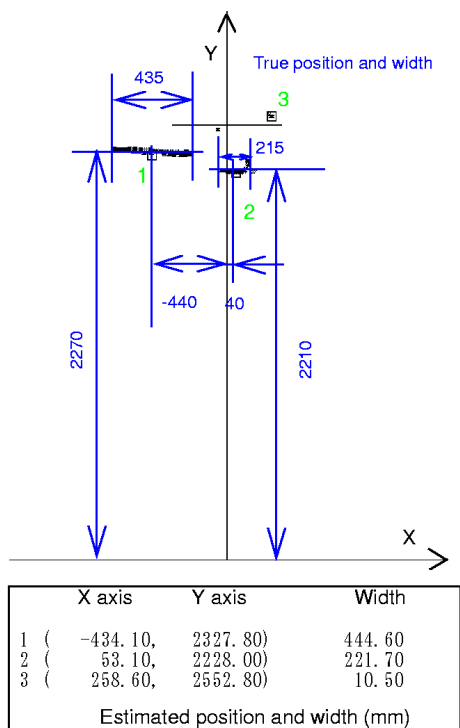


Fig.9 True and estimated position and width

は0.01m/sec、横方向では0.02m/secである。これを基線長1.2mのカメラに換算すると前後方向は60mで5km/h、横方向は20mで0.2km/hである。道路への飛び出しが検知できることがわかる。Fig.11では14秒付近から24秒付近まではターゲットが遠ざかっていくために遠くなるにつれて誤差が増えることで推定速度のばらつきが大きくなっている。立ち上がりがなめらかなのはフィルタが入っているためである。

#### 3. 3. 実環境での検出例

高速道路において走行しながらシステムを動作させた結果についてFig.12～14に示す。

この時のカメラの基線長は約1.2m、画角は約40度、水平画素数は768である。対応をとる処理領域の大きさとしては実画像を水平方向にシフトさせて右と左の画像としたものについて相関をとった時のばらつき (Table 1) から0.1画素以下になるものとして横の精度を上げるために横が7、縦が15を選んだ。

この設定での距離精度は10mから100mの範囲で距離の5%以下 (100mで5m以下)、ばらつきは、3σで距

離の2%以下(100mで2m以下)であった。フロントガラスを含めたキャリブレーションをすることで距離精度の向上が見込まれる。

Fig.12では処理している画像と対応結果を平面上にプロットした表示を示している。小領域の対応をとっているにもかかわらず隣の車線を走行する車両の側面も対応がとれ、後部面と側面が合わさってL字型の距離データがでてるのがよくわかる。これだけの検出範囲をミリ波レーダやレーザーレーダで実現するには多くの素子をならべなければならない。

Fig.13は夜間の画像について処理したもので前方を走行する3台の車が検出できている。ガードレールは検出できないが路肩のリフレクタは良く検出される。また、対向車線を走行する自動車のヘッドライトも高速に接近する点として検出されている。

Fig.14ではクラスタリングした結果を射影変換して処理している画面に重ねて表示したもので、前走車は8.9km/hで遠ざかっていて、側壁は77.1km/hで接近していることを示している。検出結果がほぼ前走車の幅に一致していることと、単一センサのため画像とのずれがないことがわかる。つまり、画像から検出した他の情報、たとえば、走行領域との位置関係にずれが発生しない。さらに、車幅がわかることですり抜け判断ができることもわかる。

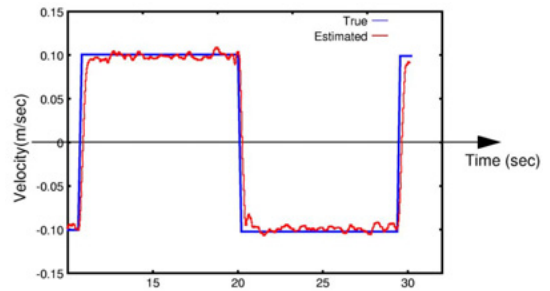


Fig.10 True and estimated lateral velocity

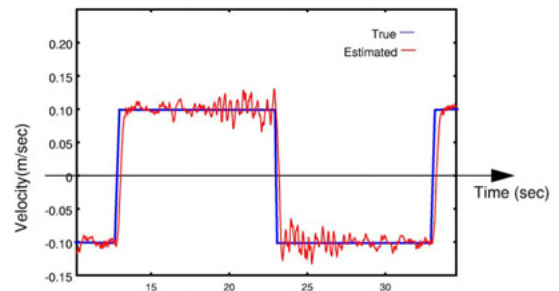


Fig.11 True and estimated longitude velocity

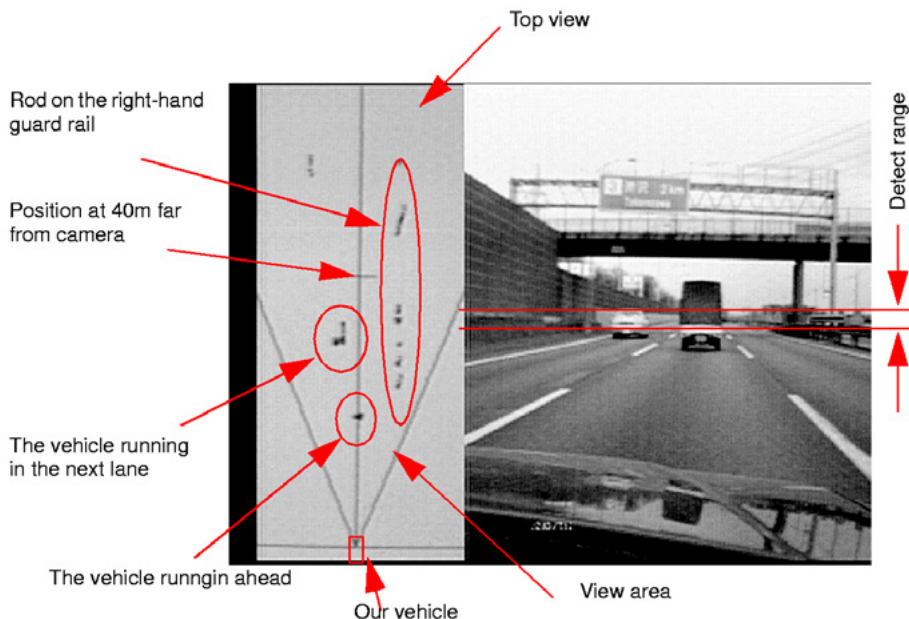


Fig.12 Real image and disparity map in the daytime

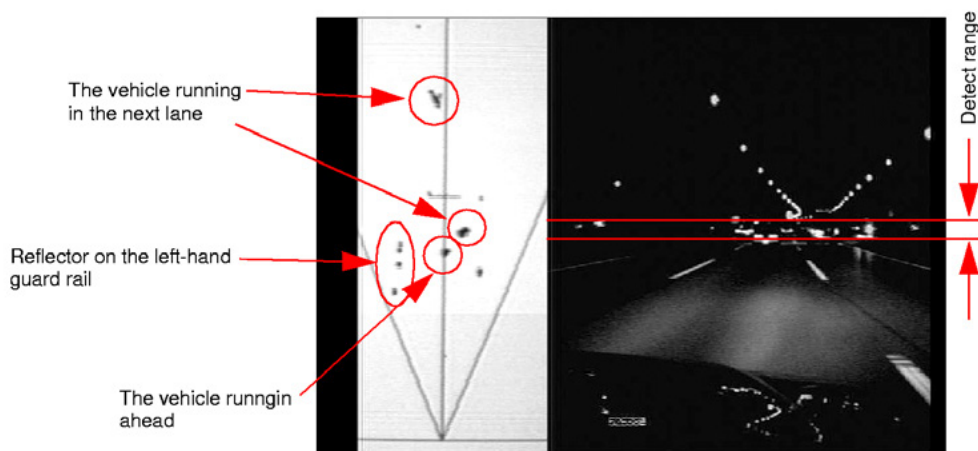


Fig.13 Real image and disparity map at night



Fig.14 Real image and estimated width, distance and velocity of targets

#### 4. まとめ

(1)ステレオ画像処理と動画処理が、相関演算処理の最適化により車載できるボードにおいてリアルタイムで可能であることがわかった。

最大処理領域サイズ 水平640画素

最大マスクサイズ 15×15画素

視差探索範囲 水平方向 0～128画素、  
垂直方向 -2～1画素

動き探索範囲 水平方向 -8～7画素、  
垂直方向 -4～3画素

サイクルタイム 33msec

(2)クラスタリングの高速化によりリアルタイムで前走車の幅、相対位置・速度が検出できた。

(3)開発したシステムを高速道路において動作させ、昼夜を問わず前走車が検知できることが確認できた。

## 参考文献

- 1) 鶴賀孝宏、「ホンダASVの紹介」、HONDA R&D Technical Review Vol.8、1996/7、pp.1-11
- 2) 古川修、「ホンダAHSの概要」、HONDA R&D Technical Review Vol.8、1996/7、pp.12-20
- 3) 太田充彦、「ステレオ視ボード」、画像ラボ、1996.10、pp.54-7
- 4) 実吉敬二、「三次元画像認識技術を用いた運転支援システム」、自動車技術会 学術講演会前刷集 924、1992、pp.169-72

## ■著 者■



青山 千秋



峯田 憲一



岡田 泰仕