

# GAによるブロック・マッチング探索を用いた オプティカル・フロー推定法の一提案

# 1. 背景と目的

我々は屋内走行用自律移動ロボットの開発に取り組んでいる。

電動車椅子を  
改造し、各種  
センサーをお  
よび走行制御  
のためのPC  
を搭載

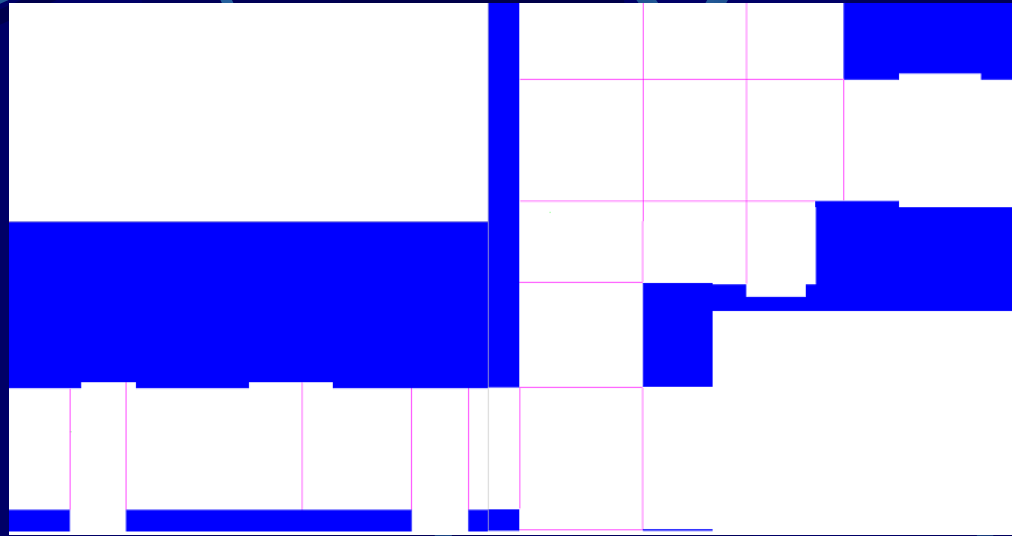
この他、  
CCDカメラ  
姿勢角センサ  
を搭載



超音波距離  
センサ

ライン検出  
リーダー

- 走行環境の地理的情報をPC上の環境マップとして保持
- 各種センサ出力および環境マップを参照しながら、走行する。



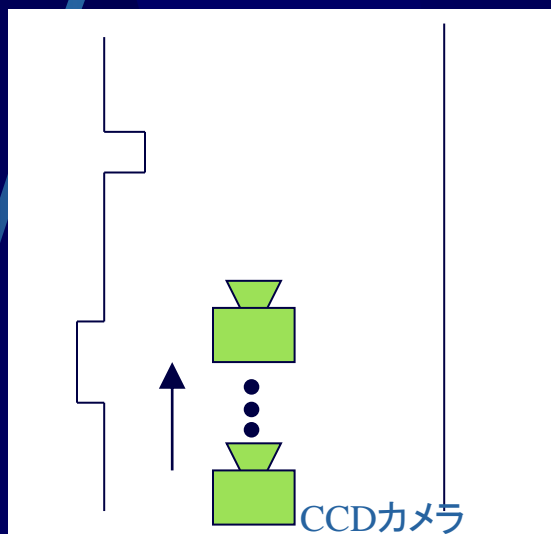
環境マップの例

### 地図情報獲得システム

環境マップを予め作成するために、CCDカメラにより走行路を撮影し、地図情報を獲得する。

## 2. 地図情報獲得システム

- (1) CCDカメラを移動させながら連続画像を取得
- (2) 各フレームの直線要素を抽出
- (3) 各フレーム間でのオプティカル・フロー推定
- (4) フレーム間での直線要素の対応・追跡



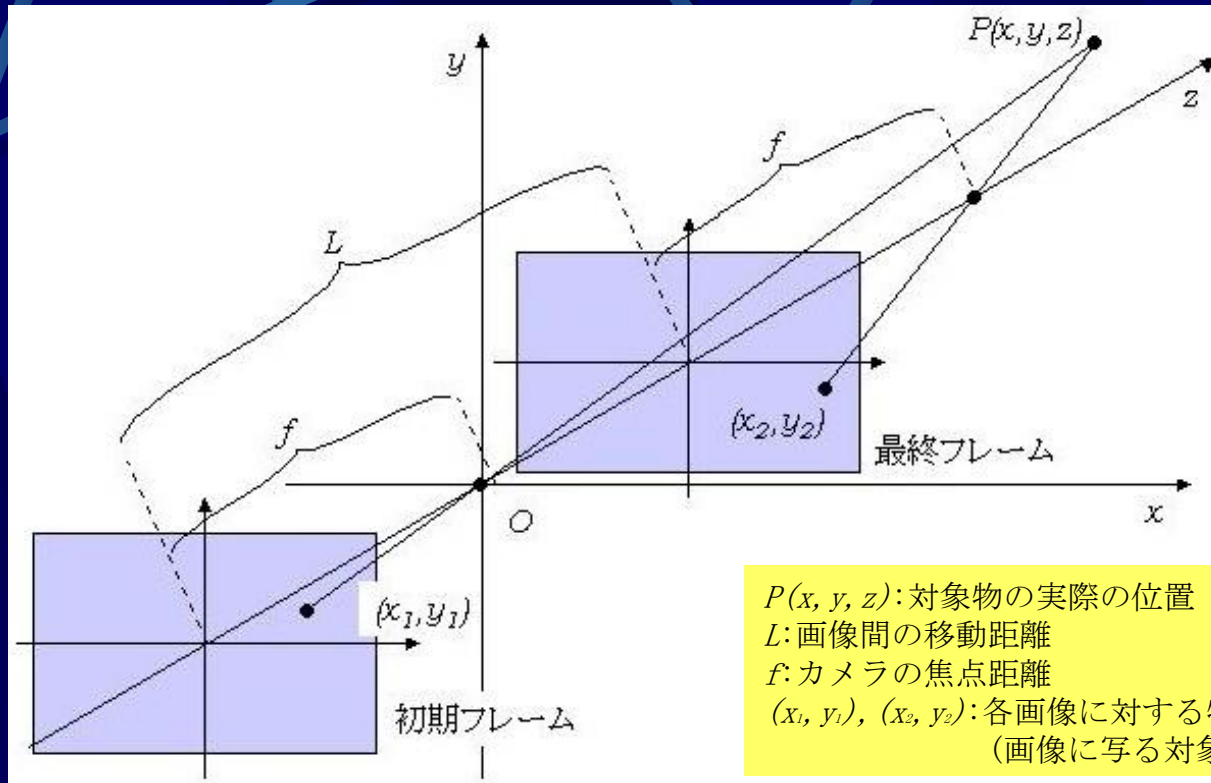
移動させながら撮影



走行路を撮影した例

初期フレームと最終フレームの各直線要素の対応

対応の取れた初期フレームと最終フレームの間で、ステレオ  
立体視により、地理的情報を抽出し、環境マップを作成する。



$P(x, y, z)$ : 対象物の実際の位置  
 $L$ : 画像間の移動距離  
 $f$ : カメラの焦点距離  
 $(x_1, y_1), (x_2, y_2)$ : 各画像に対する特徴点の座標  
 (画像に写る対象物の座標)

$$x = \frac{-x_1 x_2 (L + f)}{f(x_2 - x_1)} \quad y = \frac{-x_1 y_2 (L + f)}{f(x_2 - x_1)} \quad z = \frac{-x_1 (L + f)}{x_2 - x_1}$$

地図情報獲得システムでは、...

- 連続するフレーム間の対応をとるためのオプティカル・フロー推定を用る。

本報告では、...

- 種々のオプティカル・フロー推定法を比較検討
- ブロック・マッチング法にGAを適用した手法の検討

## ■ 各フレームに対する画像処理

◆ 濃淡化

◆ エッジ成分抽出

▪ sobelフィルタ

▪ 二値化

◆ ラベリング

◆ 端点抽出



## ■ 各フレームに対する画像処理

◆濃淡化

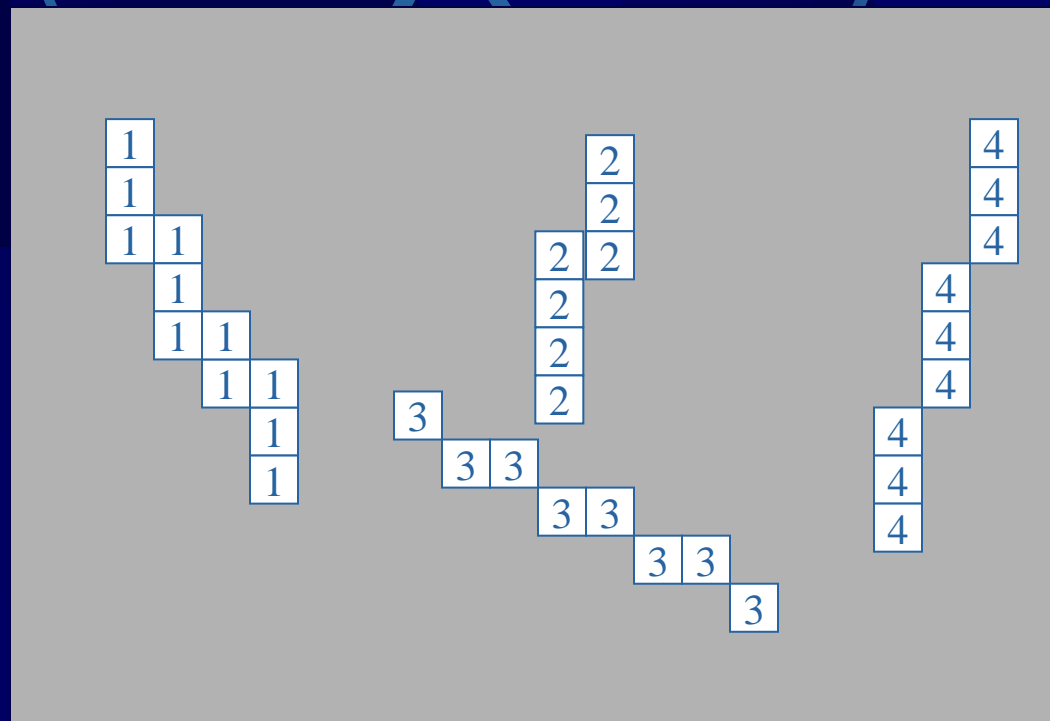
◆エッジ成分抽出

▪sobelフィルタ

▪二値化

◆ラベリング

◆端点抽出



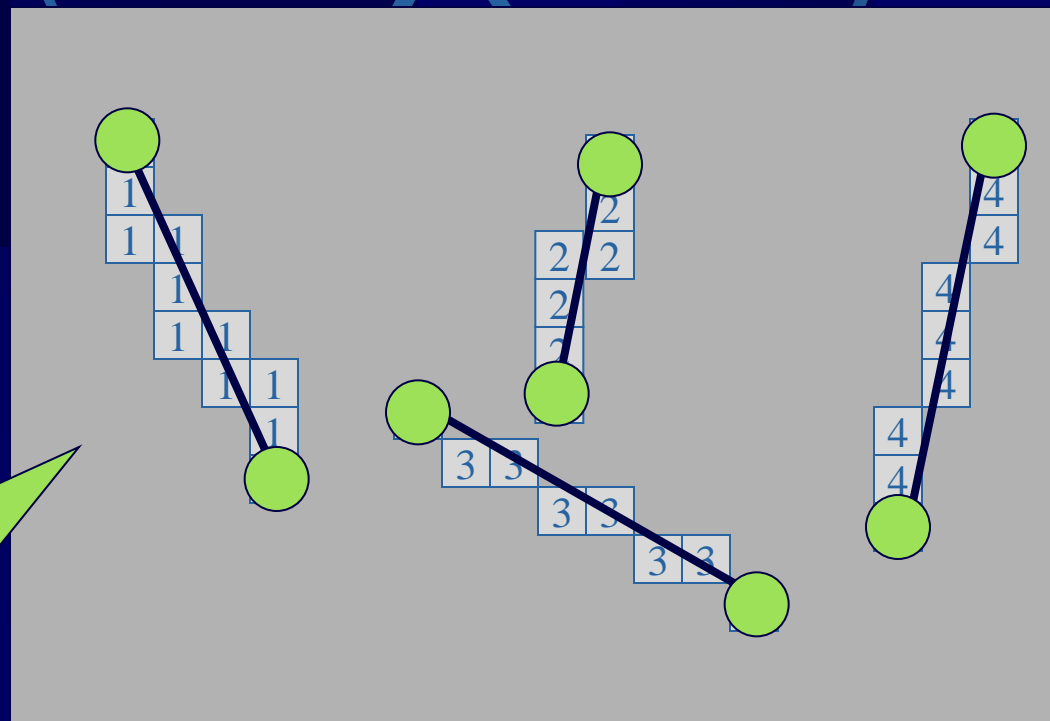
ラベル付けられた画素の例



## ■ 各フレームに対する画像処理

- ◆濃淡化
- ◆エッジ成分抽出
  - sobelフィルタ
  - 二値化
- ◆ラベリング
- ◆端点抽出

これらの端点を  
特徴点とし、フ  
レーム間で追  
跡する。



端点抽出の例

### 3. GAを用いたオプティカル・フロー推定法

#### ■ 種々のオプティカル・フロー推定法

■ 光学勾配法: オプティカル・フロー (OF) を含む拘束方程式を過剰決定連立方程式として解く手法。  
(過剰決定連立方程式の解き方により、種々の手法が提案されている。)

■ ブロック・マッチング探索法: 局所領域の類似度により移動量であるオプティカル・フローを算出する手法。

## ■ 光学勾配法

### ■ 画像の時空間微分による拘束方程式を解く方法

仮定:  $I(x, y; t) = I(x + \Delta x, y + \Delta y; t + \Delta t)$

テーラー展開

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

整理すると

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0$$

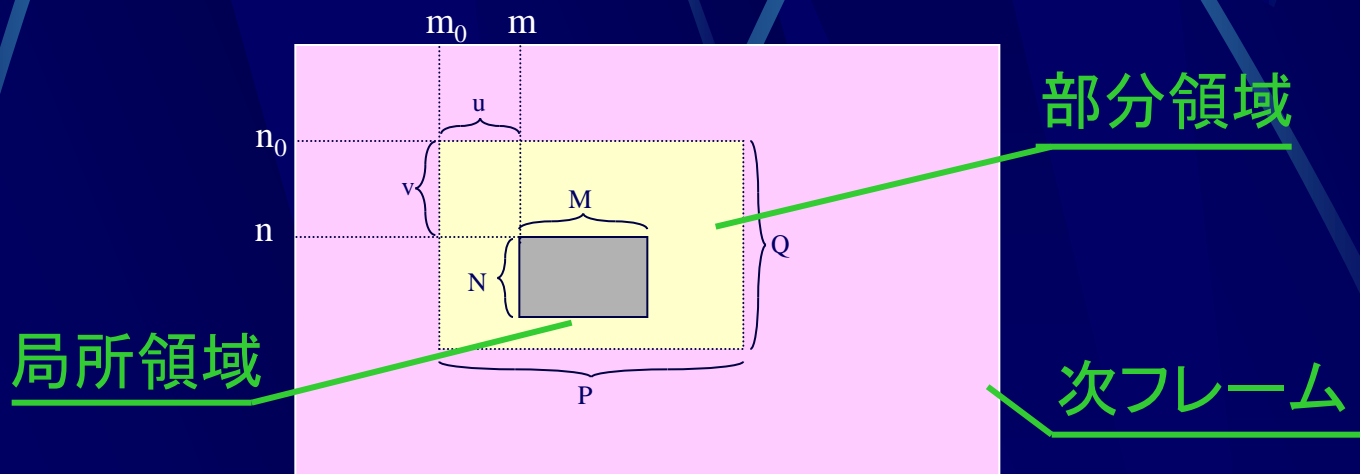
拘束式

### ■ 信頼度を別途保障しなければならない

## ■ ブロック・マッチング探索法

- 画像中の局所領域をテンプレート・ブロックとする。
- 次フレーム中の部分領域を全探索する。
- 局所領域の評価は差分評価する。

$$D(u, v) = \sum_{n=n_0}^{n_0+P-1} \sum_{m=m_0}^{m_0+Q-1} |I(x+m, y+n; t) - I(x+m-u, y+n-v; t + \Delta t)|$$



## ■ それぞれの特徴

### ■ 光学勾配法(種々の手法を試みた)を用いた場合

- ◆ 計算時間は比較的短い。
- ◆ 推定結果の精度は低い。

### ■ ブロック・マッチング探索法を用いた場合

- ◆ 推定精度は比較的高い。
- ◆ 計算時間は長い。(本研究への応用は困難)

それぞれの手法による結果は、後述する。

## ■ GAを用いたブロック・マッチング探索によるOF推定(提案手法)

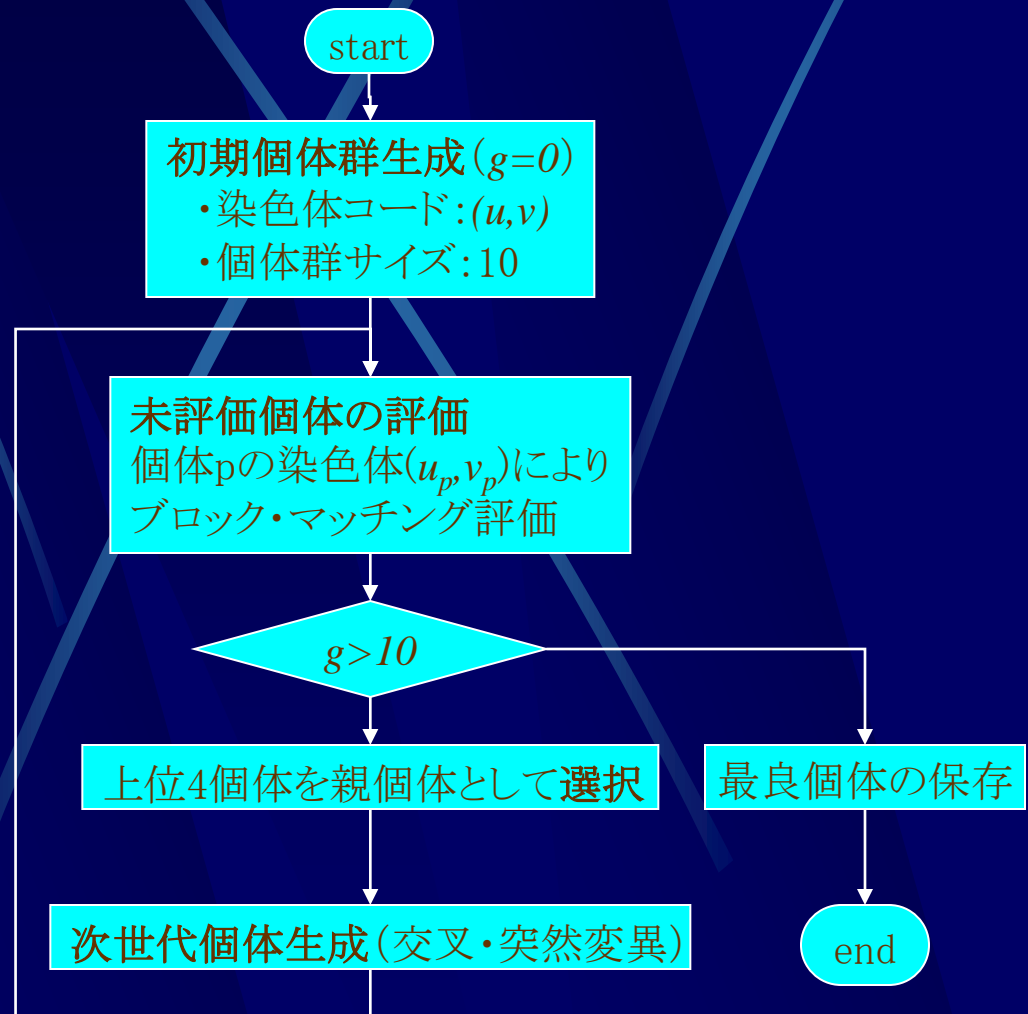
■ ブロック・マッチング探索における、 $(u, v)$ をGAにより探索する。

■ 計算時間の縮小を目的とするため、簡易なアルゴリズムが望ましい。

■ 遺伝子コード

```
int u | int v
```

■ 個体群サイズ:10, 上位4個体選択, 終了世代数:10

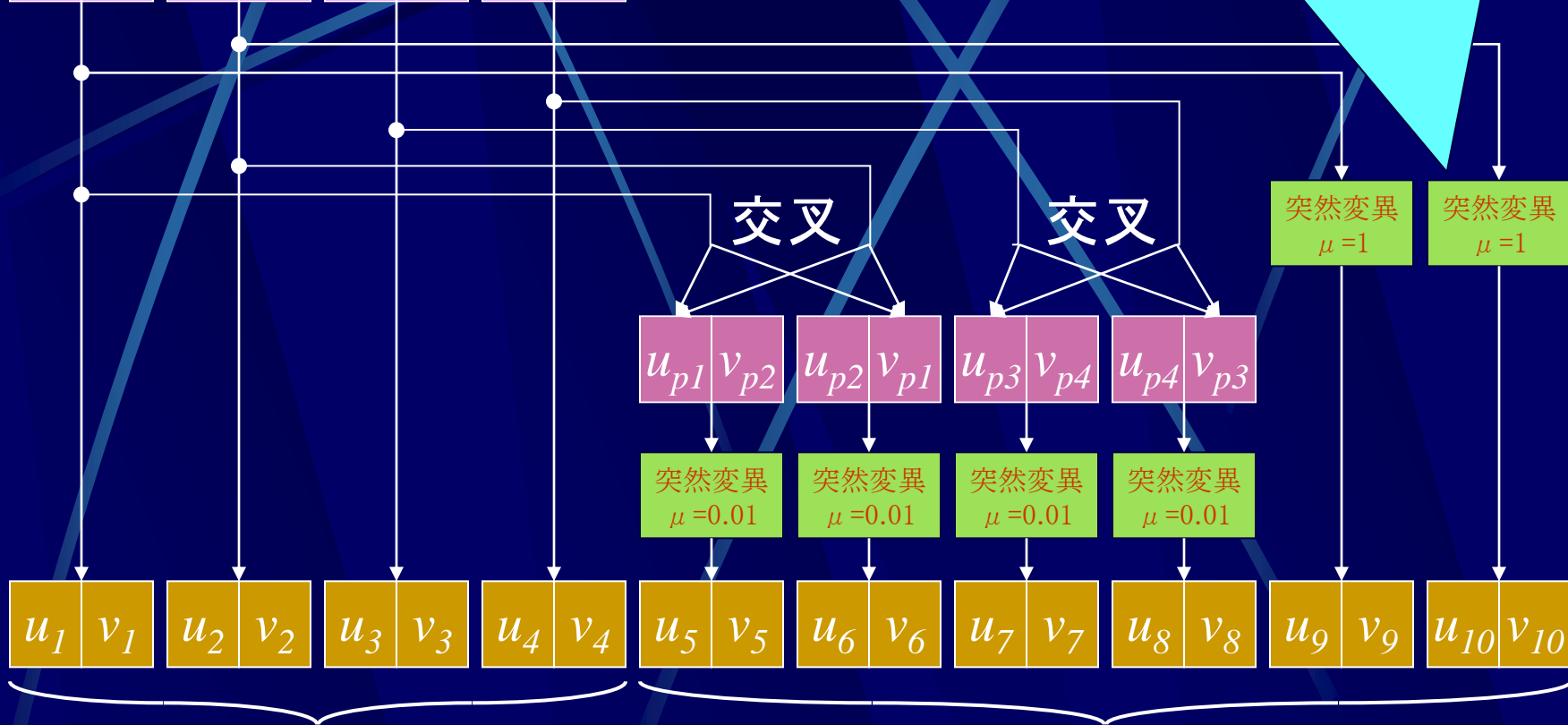
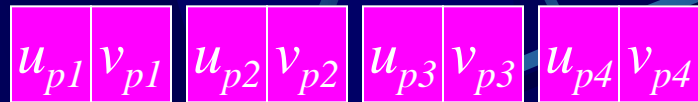


## GAでの処理の流れ

# 突然変異の演算

$$\langle u_p/v_p \rangle + \delta, \delta: \text{random number} \in [-2, 2]$$

選択された上位4個体(親)



既評価(4個体)

未評価(6個体)

## 次世代への個体群の生成手順

## 4. 実験と検討

### ◆ 試行したOF推定法

#### ● 光学勾配法

- 空間局所最適化法

- 二階微分法

- 時空間フィルタリング法

- Lucas-Kanade法

- ブロック・マッチング探索法

- GAを用いたブロック・マッチング探索法

### 諸条件:

対象画像のサイズ:  $640 \times 480$ [pix<sup>2</sup>], 局所領域サイズ:  $10 \times 10$ [pix<sup>2</sup>]

部分探索領域サイズ:  $20 \times 20$ [pix<sup>2</sup>]

使用したPC: Sony, Model PCV L-520(pentium 500MHz, 512MB)

処理系: Visual C++6.0, MFC AppWizard(exe)ダイアログベース



## ◆ 使用したフレーム(原画像)



第1フレーム



第2フレーム

## ◆ 空間局所最適化法の試行結果



- ある程度、エッジ付近にOF推定値が得られている。
- 全体的に実際の変異より小さすぎる値が多い。

## ◆ 二階微分法



- 極端に大きな値のOF推定値が得られたり、エッジが存在しない点でもOF推定値が検出された。
- 全体的に適切にOF推定値が得られていない。

## ◆ 時空間フィルタリング法



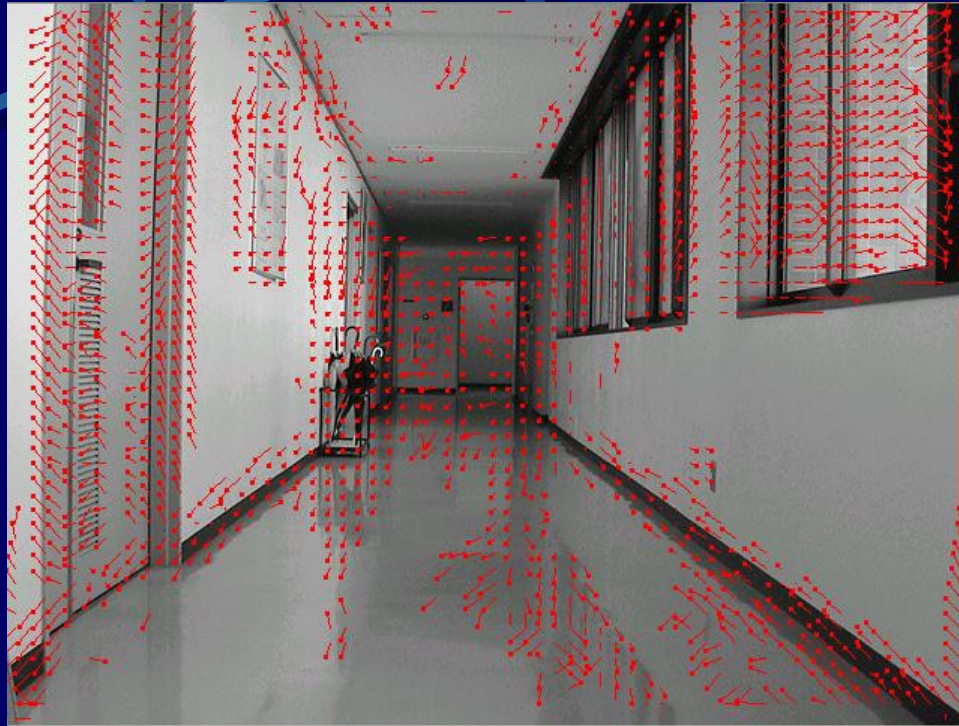
- 極端に大きな値のOF推定値が得られたり、エッジが存在しない点でもOF推定値が検出された。
- 二階微分法と同様に、適切にOF推定値が得られなかった。

## ◆ Lucas-Kanade法



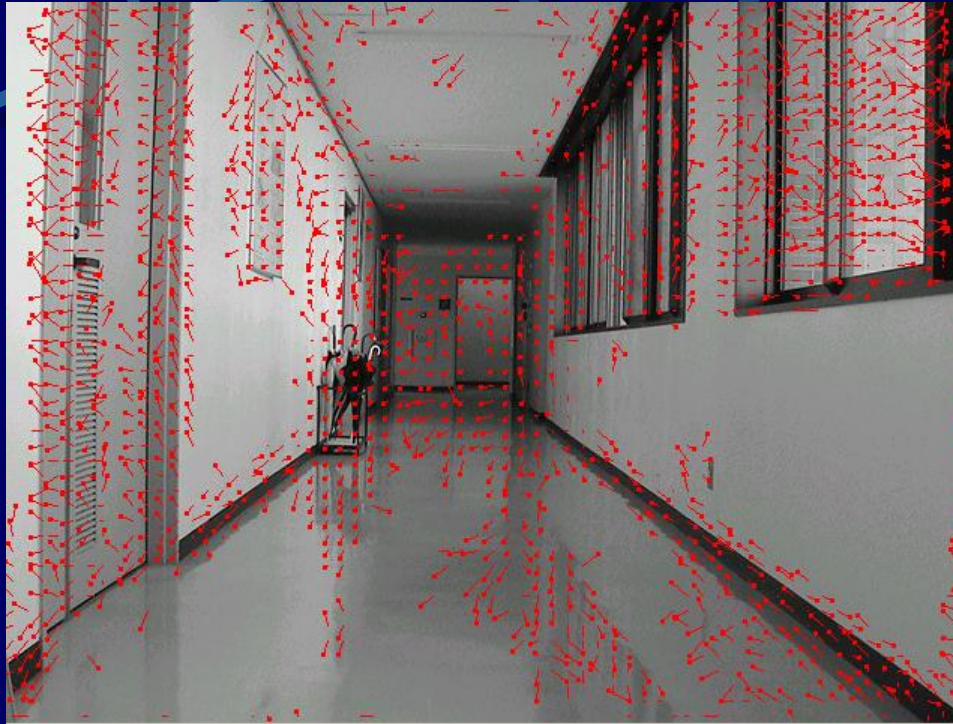
- エッジの上で、ある程度適切にOF推定値が得られている。
- OF推定値が得られなかったエッジ部分もある。
- 全体的にOF推定値が、実際の変異より小さい。
- 全体的には、試行した光学勾配法の中では最も良好な結果が得られた。

## ◆ ブロック・マッチング探索法



- エッジおよびエッジ付近に適切にOF推定値が得られた。
- 今回試みた手法の中では、OF推定値だけを比較すると、最も良好な結果が得られた。

## ◆ GAを用いたブロック・マッチング探索法(提案手法)



- OF推定値が得られた点は、ブロック・マッチング探索法とほぼ同等である。
- 得られたOF推定値の精度はやや劣る。

## ◆ 処理時間の比較

手法	処理時間[sec]
空間局所最適化法	0.66
二階微分法	0.82
時空間フィルタリング法	0.05
Lucas-Kanade法	0.11
ブロック・マッチング探索法	28.79
提案法	9.34



## 5. まとめ

●自律移動ロボットが移動するには環境マップが必要

環境マップ作成にオプティカルフロー推定で直線要素を対応付けし、地理情報を算出する。



そのオプティカル・フロー推定の手法として

種々の光学勾配法、ブロックマッチング探索法、GAを用いたブロックマッチング探索法で比較・検討を行った。

### 結果

提案手法は比較的短い処理時間で、良好な精度のオプティカル・フロー推定値を与えた。

### 今後

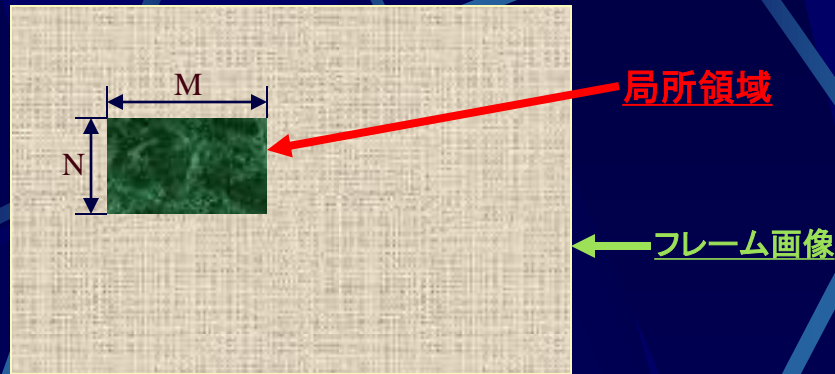
- さらに精度を確保しつつ処理時間の短縮をはかる。
- 地図情報獲得システムを開発する。

これにて発表を  
終わらせていただきます。

御静聴、ありがとうございました。

# 補足1: 空間局所最適化法

ある空間領域において各点のOFは一定であると仮定することで得られる手法。



空間局所領域 $M \times N$ を設定すると局所領域内におけるオプティカルフロー $(u, v)$ は一定であると仮定し、各画素ごとに拘束方程式を用いる。拘束方程式を行列で下の様に表現する。

$$Gf = -b$$

このとき、過剰連立方程式を右のように設定し、上式を最小二乗法で解くと次式のようになる。

$$f = -(G^T G)^{-1} G^T b$$

これによりオプティカルフロー  $f(u, v)$  を求める。

$$G = \begin{pmatrix} \frac{\partial I_1}{\partial x} & \frac{\partial I_1}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I_i}{\partial x} & \frac{\partial I_i}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I_{M \times N}}{\partial x} & \frac{\partial I_{M \times N}}{\partial y} \end{pmatrix}$$
$$f = \begin{pmatrix} u \\ v \end{pmatrix}$$
$$b = \begin{pmatrix} \frac{\partial I_1}{\partial t} \\ \vdots \\ \frac{\partial I_i}{\partial t} \\ \vdots \\ \frac{\partial I_{M \times N}}{\partial t} \end{pmatrix}$$

## 補足2、二階微分法

各画素において濃淡値の二階微分を求める手法。局所領域内全ての画素位置について、濃淡値の空間勾配をもとめ、過剰連立方程式を組み立てる。拘束方程式を下の様に表現する。

$$Gf = -b$$

このとき、過剰連立方程式をこのように設定し、空間微分(空間勾配)及び空間の二階微分を求める。

$$G = \begin{pmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \\ \frac{\partial}{\partial x} \bullet \frac{\partial I}{\partial x} & \frac{\partial}{\partial x} \bullet \frac{\partial I}{\partial y} \\ \frac{\partial}{\partial y} \bullet \frac{\partial I}{\partial x} & \frac{\partial}{\partial y} \bullet \frac{\partial I}{\partial y} \end{pmatrix}$$
$$b = \begin{pmatrix} \frac{\partial I}{\partial t} \\ \frac{\partial}{\partial x} \bullet \frac{\partial I}{\partial t} \\ \frac{\partial}{\partial y} \bullet \frac{\partial I}{\partial t} \end{pmatrix}$$

空間局所最適法と同様、最小二乗法で解くことにより次式のようになる。

$$f = - (G^T G)^{-1} G^T b$$

これによりオプティカルフロー  $f(u, v)$  を求める。

## 補足3、時空間フィルタリング法

光学勾配法の基本拘束式を解くため、二種類のフィルタを用いる手法。  
フィルタをそれぞれ $g, h$ とし、適用した結果について空間勾配を求め $\mathbf{G}, \mathbf{b}$ を得る。

$$\mathbf{G} = \begin{pmatrix} \frac{\partial}{\partial x} (g * \mathbf{I}) & \frac{\partial}{\partial y} (g * \mathbf{I}) \\ \frac{\partial}{\partial x} (h * \mathbf{I}) & \frac{\partial}{\partial y} (h * \mathbf{I}) \end{pmatrix}$$
$$\mathbf{b} = \begin{pmatrix} \frac{\partial}{\partial t} (g * \mathbf{I}) \\ \frac{\partial}{\partial t} (h * \mathbf{I}) \end{pmatrix}$$

“\*”は畳み込み積分である。拘束方程式

$$\mathbf{G}f = -\mathbf{b}$$

から下式が得られ、これよりオプティカルフローを検出する。

$$f = -\mathbf{G}^{-1}\mathbf{b}$$

## 時空間フィルタリング法のフィルタに関しては下のような特徴が要求される。

- 拘束式での画像の微分可能性を保障する連続性
- 拘束式のテイラー展開の高次項無視による影響の抑制
- 二つの拘束式が独立となる

このような条件を満たすためには

- 平滑効果をもつフィルタである
- フィルタにそれぞれ方向性をもたせる → **ガウスフィルタを用いる**

$$g(x, y) = \frac{\partial}{\partial x} \left[ \frac{1}{2\pi\sigma_1\sigma_2} \exp \left( -\frac{1}{2} \left\{ \left( \frac{x}{\sigma_1} \right)^2 + \left( \frac{y}{\sigma_2} \right)^2 \right\} \right) \right]$$

← X方向に分散大きく、かつ微分されたフィルタ

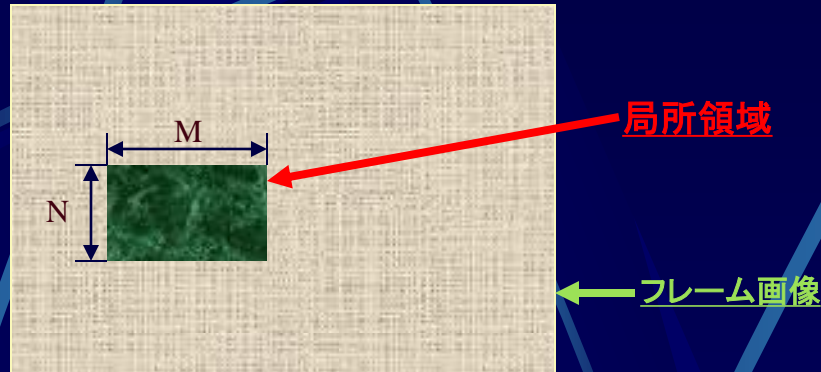
$$h(x, y) = \frac{\partial}{\partial y} \left[ \frac{1}{2\pi\sigma_1\sigma_2} \exp \left( -\frac{1}{2} \left\{ \left( \frac{x}{\sigma_2} \right)^2 + \left( \frac{y}{\sigma_1} \right)^2 \right\} \right) \right]$$

← y方向に分散大きく、かつ微分されたフィルタ

( $\sigma_1, \sigma_2$ は、分散を表し、 $\sigma_1 \gg \sigma_2$ )

## 補足4、Lucas-Kanade法

空間局所最適化法の一つ。下図のように局所領域を仮定すると、



局所領域のオプティカルフロー  $u, v$  は次式で決定される。

$$u = \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \frac{\partial I(x, y; t)}{\partial x} \{I(x, y; t + \Delta t) - I(x, y; t)\}}{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \left\{ \frac{\partial I(x, y; t)}{\partial x} \right\}^2}$$
$$v = \frac{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \frac{\partial I(x, y; t)}{\partial y} \{I(x, y; t + \Delta t) - I(x, y; t)\}}{\sum_{y=0}^{N-1} \sum_{x=0}^{M-1} \left\{ \frac{\partial I(x, y; t)}{\partial y} \right\}^2}$$