



## 簡単なデータ表現を用いた両眼視に基づく距離情報の抽出\*

横 矢 直 和\*\* 北 橋 忠 宏\*\* 田 中 幸 吉\*\*

### Abstract

Depth information is important to the three-dimensional recognition of a scene by computers. In this paper, a depth recognition method is described which is based on the binocular stereoscopic vision. Extracting the depth information from stereoscopic images requires algorithmic matching of the left and right images, the so-called correspondence problem.

The proposed algorithm is based on finding boundaries of regions corresponding each other. The regions are defined as the parts with certain gray-levels in each of the images. Thus, the given images are transformed into 2-valued images. This enables the proposed simple algorithm (region-method) to get correspondence of objects in both images.

The region-method is applied to two examples and compared with a conventional correlation-method.

### 1. ま え が き

三次元物体（あるいは自然風景）の認識における有用な情報として距離情報を考えることができる。人間の距離認識においては、両眼の網膜上に生じた像の左右の位置的なずれ（両眼視差）を検出することによる認識——両眼視——、物体像の大小関係に基づく相対的な距離の認識、目の焦点調節による認識、経験に基づく知識からの推論による認識などが総合的に結合されていると考えられる。画像を用いた距離認識において利用できるのは両眼視と推論による認識であるが、本論文では両眼視を利用した方法を扱っている。

一般に、計算機を用いて物体認識を行う場合、物体表面の複雑な模様や表面に映った他の物体の影などは障害になると考えられていたが、両眼視を利用した場合には、安江、白井<sup>1)</sup>が指摘しているように、これらの情報を積極的に表面認識に利用できるという利点がある。

ある。両眼視を利用した距離認識方法は種々提案されてきたが<sup>1)~4)</sup>、これらすべてにとって問題となったのは左右画像内の対応点をいかにして正確かつ効率的に見つけるかということであった。2値の線図形の場合の対応点の決定は比較的簡単であるが、濃淡画像を対象とした場合には、この問題は容易ではない。

従来は、左右画像内の特徴点（空間微分効果として得られる濃淡の変化点）を中心とする矩形を設定し、左右の矩形間の類似度が最大となる特徴点の対を対応点対と決定する方法——相関法——が採用されてきた。しかし、この方法は類似度を求める際に複雑な評価関数の計算を必要とし、しかも1点の対応決定に対して多数の特徴点と比較する必要があるため、全体として評価関数の計算回数が非常に多くなるという欠点をもっている。これは画像があたかもランダム・パターンであるかのように見なして対応づけを行っていることに原因があると思われる。実際には、物体を写した画像はランダム・パターンではなく、画像内の物体固有の構造をもっている。すなわち、左右それぞれの画像内で同一の対象を表わしている部分は同程度の濃淡レベルをもち、その周辺でも類似の濃淡分布をもっていると考えられる。そして、この部分が集まって

\* An extraction of the depth information from the stereoscopic images based on a simple data description by Naokazu YOKOYA, Tadahiro KITAHASHI and Kokichi TANAKA (Department of Information & Computer Science, Faculty of Engineering Science, Osaka University)

\*\* 大阪大学基礎工学部情報工学科

領域を構成している。

このことを考慮に入れ、前記の相関法に替る対応づけの方法として、画像内の領域に注目した方法——領域法——を開発したので、その基本アルゴリズムといくつかの両眼画像に対する実験結果について報告する。本方法では、まず左右画像内の特定の範囲内の濃淡レベルをもつ点を要素とする領域を取り出し、左右でのその領域にそれぞれ異なる重みを与える。そして、両画像を重ね合わせた場合の変化点の特徴によって左右画像内の領域端 (region-boundary) の対応づけを行う。

最後に、従来の相関法との比較および本方法の適用範囲等について考察する。

### 2. 両眼視のカメラ・モデル

筆者らは両眼視におけるカメラ・モデルに関する考察を行ったが<sup>5)</sup>、ここでは基線 (2つのカメラ・レンズの中心を結ぶ線分) に対して垂直な光軸を持つ平行に設置された2台の等価なカメラから両眼写真を得るモデルを考える。基線の長さを  $l$ 、カメラ・レンズの焦点距離を  $f$  とし、光軸に沿ってレンズの中心から  $f$  の距離に画像平面があると仮定する。

Fig. 1 は三次元空間内の基線からの奥行き  $d$  である点  $P$  を左右画像平面に投影した様子を示している。この図で、 $P_L(x_L, y_L)$  は左側画像平面における像を表わし、 $P_R(x_R, y_R)$  は右側画像平面における像を表わしている。このとき、画像平面内の二次元座標系で、 $Y$  座標に関しては式 (2.1) が成立し、奥行き  $d$  に関しては式 (2.2) が成立する。

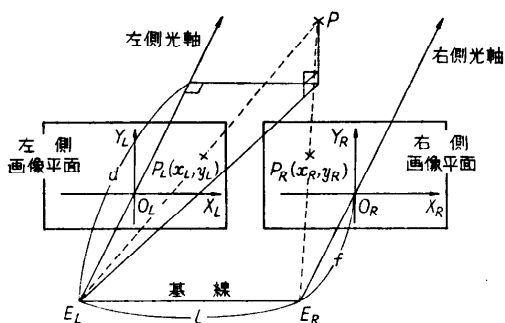


Fig. 1 Camera-model of binocular stereoscopic vision

\* 本論文では、標準化された画像のみを対象としており、画像と画像関数を同義に用いている。

\*\* reference picture: 処理過程において主として引用する画像であり、本論文では右側画像を参照画像とする。本論文で、特に指定のない限り画像内の点は参照画像の点を指している。

$$y_L = y_R \tag{2.1}$$

$$d = \frac{l \cdot f}{x_L - x_R} \tag{2.2}$$

すなわち、左右画像内の対応点は  $Y$  座標が等しく、点  $P$  の奥行き  $d$  は像の  $X$  座標のずれ (両眼視差) に反比例する。したがって、本モデルを用いた場合の対応点探索に際しては  $X$  方向 (行方向) での探索を行えばよい。

### 3. 左右画像内の物体の対応づけ

前章で述べたカメラ・モデルを用いて得られる両眼画像を対象としたときの対応づけの方法として、従来から考えられてきた相関法の概要を示し、ここで新たに提案する領域法について説明する。

左右の画像関数を  $G^L, G^R$  とし、それぞれ  $M \times N$  個に標準化され、各標本点の濃淡は  $0 \sim p-1$  の  $p$  レベルに量子化されているものとする。

#### 3.1 相関法

従来、提案されてきた左右画像内の物体の対応づけアルゴリズムの概要を次に示す。

- 1) 左右画像に対して空間微分処理を行い、閾値  $\theta_1$  に対して  $\partial^2 G(x, y) / \partial y \partial y \geq \theta_1$  を満足する点  $(x, y)$  を求める。この点を特徴点と呼ぶ。
- 2) 参照画像\*\*内の1つの特徴点に対して、他方の画像内で対応点が存在し得る範囲内のすべての特徴点との類似度を調べる。類似度の計算には評価関数を用いるのが一般的であり、左側画像内の特徴点  $(x+r, y)$  と右側画像内の特徴点  $(x, y)$  の類似度評価関数として次のような関数が考えられている<sup>6)</sup>。ただし、特徴点を中心とする矩形の大きさは  $(2m+1) \times (2n+1)$  である。

$$\varphi_1(r; x, y) = - \sum_{\xi} \sum_{\eta} [G^L(\xi+r, \eta) - G^R(\xi, \eta)]^2 \tag{3.1}$$

$$\begin{aligned} \varphi_2(r; x, y) &= \frac{\sum_{\xi} \sum_{\eta} G^L(\xi+r, \eta) \cdot G^R(\xi, \eta)}{\sqrt{\sum_{\xi} \sum_{\eta} [G^L(\xi+r, \eta)]^2} \cdot \sqrt{\sum_{\xi} \sum_{\eta} [G^R(\xi, \eta)]^2}} \end{aligned} \tag{3.2}$$

$$\begin{aligned} \varphi_3(r; x, y) &= \frac{1}{(2m+1) \cdot (2n+1)} \sum_{\xi} \sum_{\eta} G^L(\xi+r, \eta) \cdot G^R(\xi, \eta) \\ &\quad \cdot \sigma_L(x+r, y) \\ &\quad - \frac{\mu(x+r, y) \cdot \mu(x, y)}{L \cdot R} \\ &\quad \cdot \sigma_R(x, y) \end{aligned} \tag{3.3}$$

ただし,  $\sum_{\xi} \sum_{\eta}$  はそれぞれ  $\sum_{\xi=x-m}^{x+m} \sum_{\eta=y-n}^{y+n}$  を意味する.  $\mu_L, \mu_R$  は矩形内での濃淡の平均であり,  $\sigma_L^2, \sigma_R^2$  は矩形内での濃淡の分散である.

- 2)  $r$  を  $r=0, 1, 2, \dots, r_{max}$  (最大視差) と変化させて, 左側画像内の点  $(x+r, y)$  が特徴点であるような  $r$  に対して, 繰り返し  $\varphi(r; x, y)$  を計算し,  $\varphi(r; x, y)$  を最大にする  $r$  の値  $r_0$  を見つける.
- 4) 閾値  $\theta_2$  に対して,  $\varphi(r_0; x, y) \geq \theta_2$  ならば, 左側画像内の点  $(x+r_0, y)$  と右側画像内の点  $(x, y)$  とを対応づけ, 点  $(x, y)$  の両眼視差は  $r_0$  である. 逆に,  $\varphi(r_0; x, y) < \theta_2$  ならば, 右側画像内の点  $(x, y)$  に対して左側画像内に対応点が存在しないと判定する.

### 3.2 領域法

左右画像内の物体の対応づけの方法として領域法を提案する. 本方法では, まず左右画像を各々, 濃淡情報に基づいて2値パターンに変換する. このことは, ある巾の濃度をもつ領域を抽出することを意味する. 次に, このように簡単化された左右の2値画像データを合成し, この合成されたデータの変化の型に着目して画像内の領域端を対応づける.

#### 3.2.1 対応づけに必要な諸定義

対応づけ領域とそれに関する2種類の関数を定義する.

【定義1】 (対応づけ領域) 対応づけを行う領域  $A_{ij}^L, A_{ij}^R$  を次式で定義する.

$$A_{ij}^L = \{(x, y) | 0 \leq i \leq G^L(x, y) \leq j \leq p-1\} \quad (3.4)$$

$$A_{ij}^R = \{(x, y) | 0 \leq i \leq G^R(x, y) \leq j \leq p-1\} \quad (3.5)$$

領域を定めるパラメータ  $i, j$  の決定法としては, (1)  $G^L, G^R$  の値域  $[0, p-1]$  を等分割し, 隣り合う分割点の値を順次  $i, j$  に与える方法と, (2) 濃淡のヒストグラムに基づいて1対の  $i, j$  が1つの“山”をはさむように山の数だけの  $i, j$  の対を設定する方法とが考えられる.

【定義2】 (領域関数) 領域  $A_{ij}^L, A_{ij}^R$  に対して2値の領域関数 (region-function)  $G_{ij}^L, G_{ij}^R$  を次のように定める.

$$G_{ij}^L(x, y) = \begin{cases} 1; & (x, y) \in A_{ij}^L \\ 0; & \text{otherwise} \end{cases} \quad (3.6)$$

$$G_{ij}^R(x, y) = \begin{cases} 1; & (x, y) \in A_{ij}^R \\ 0; & \text{otherwise} \end{cases} \quad (3.7)$$

【定義3】 (合成領域関数) 領域端の対応づけにおいて使用する合成領域関数 (composite region-function)  $G_{ij}$  を次のように定める.

$$G_{ij}(x, y) = \alpha \cdot G_{ij}^L(x, y) + \beta \cdot G_{ij}^R(x, y) \quad (3.8)$$

ただし, 左右の画像における領域端の特徴を保存するためには,  $\alpha \neq \beta$  でなければならない.

#### 3.2.2 領域端の対応づけ

合成領域関数  $G_{ij}$  を用いて領域  $A_{ij}^L$  と  $A_{ij}^R$  の領域端を対応づける方法を考えよう. 式 (2.1), (2.2) に示したように, 対応点の左右画像内での位置的なずれは  $x$  方向であるから,  $G_{ij}(x, y)$  の  $x$  方向での変化に基づいて対応づけを行うことができる.

ここでは, 画像を左端から右方向に走査する (Fig. 2 参照) 場合について考えており, また, 左右画像内で任意の物体対の左右関係が保存されると仮定している.

以下, 式 (3.8) で  $\alpha=2, \beta=1$  の場合について領域法を説明する.

【定義4】 点  $(x, y)$  に関して,  $G_{ij}(x, y) \neq G_{ij}(x+1, y)$  のとき, この点を変化点といい, 変化点の型  $T(x, y)$  を  $(G_{ij}(x, y) \rightarrow G_{ij}(x+1, y))$  で表現する.

【定義5】 変化点の型は12種類存在し, これらを要素とする4つの集合  $C_R^1, C_L^1, C_R^2, C_L^2$  を式 (3.9)~(3.12) によって定める<sup>7)</sup>.

$$C_R^1 = \{(0 \rightarrow 1), (2 \rightarrow 3), (2 \rightarrow 1), (0 \rightarrow 3)\} \quad (3.9)$$

$$C_L^1 = \{(1 \rightarrow 3), (0 \rightarrow 2), (1 \rightarrow 2), (0 \rightarrow 3)\} \quad (3.10)$$

$$C_R^2 = \{(1 \rightarrow 0), (3 \rightarrow 2), (1 \rightarrow 2), (3 \rightarrow 0)\} \quad (3.11)$$

$$C_L^2 = \{(3 \rightarrow 1), (2 \rightarrow 0), (2 \rightarrow 1), (3 \rightarrow 0)\} \quad (3.12)$$

これらの集合要素の意味することは以下のとおりである.

$C_R^1(C_L^1)$ : 画像関数  $G^R(G^L)$  における領域  $A_{ij}^R$  ( $A_{ij}^L$ ) の左端\*が合成領域関数  $G_{ij}$  に現われたときの変化点の型.

$C_R^2(C_L^2)$ : 画像関数  $G^R(G^L)$  における領域  $A_{ij}^R$  ( $A_{ij}^L$ ) の右端\*が合成領域関数  $G_{ij}$  に現われたときの変化点の型.

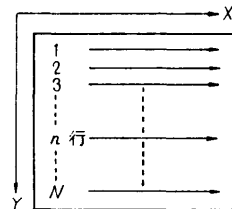


Fig. 2 The way of scanning

\* Fig. 3 参照.

これらの変化点集合は  $G_{ij}$  に関する次のような基本的性質に基づいて定められている。すなわち、領域  $A_{ij}^L, A_{ij}^R$  の左端に相当する部分では、通常  $G_{ij}$  は増加型—— $G_{ij}(x+1, y) - G_{ij}(x, y) > 0$ ——、右端に相当する部分では減少型—— $G_{ij}(x+1, y) - G_{ij}(x, y) < 0$ ——であり、右側画像内の領域端は  $G_{ij}$  の値 1 の変化点として、左側画像内の領域端は値 2 の変化点として現われる。例外は左右の変化点が重複する (0→3), (3→0), (1→2), (2→1) の変化点である。(0→3) は左右画像内の領域左端同士、(3→0) は同右端同士が重複していることを示し、(1→2) は右側画像内の領域右端が左側画像内の領域左端と重複していることを示している。(2→1) は右側画像内の領域左端が左側画像内の領域右端と重複していることを示している。

以上のことから、式 (3.9)～(3.12) の 4 種類の変化点集合のうち、 $C_R^1$  と  $C_L^1$  は互いに対応点を含み、 $C_R^2$  と  $C_L^2$  も互いに対応点を含んでいる。両眼視の原理より、対応する 2 点の座標間には

$$l > x_L - x_R \geq 0 \tag{3.13}$$

の関係が成立することから、 $G_{ij}$  を左端から右方向に走査したとき、まず、現れた  $C_R^1$  の変化点に対してその右  $l$  までの距離に現れる  $C_L^1$  の変化点を領域左端の対応点と見なす。また、 $C_R^2$  の変化点に対しても同様の  $C_L^2$  の変化点を領域右端の対応点と見なす。対応づけの直感的な理解のために、仮想的な両眼画像に対する対応づけの例を領域関数と関連づけて Fig. 3 に示す。

次に対応づけの基本アルゴリズムを説明する。

先に述べたように、対応点は合成領域関数の性質に基づいて決定されるが、実際には、 $G_{ij}^R$  が視差よりも小さな間隔で変化した場合やあるいは一方の画像内に存在している物体が他方の画像内ではかかれていない場合などに誤対応が生じ、その行の以後の対応づけに波及する可能性がある。このような致命的な誤対応を避けるために対応点  $P_L, P_R$  の近傍の類似度評価関数  $\varphi_s$  (式(3.3)参照) による対応の諾否操作を加えることによって、最終的な領域端の対応づけを行っている。

左端対応の候補点列を  $Q_1$ 、右端対応の候補点列を  $Q_2$  とし、各候補点列の書き込み場所を示すポイントとして  $t_1, t_2$ 、また検索場所を示すポイントとして  $b_1, b_2$  を使用する。 $k$  は対応づけの過程で注目している候補点列を示すパラメータであり、 $s$  は参照画像内の 1 変化点に関する対応点探索回数を示すパラメータである。

合成領域関数における点  $(x, n)$  の変化点の型を

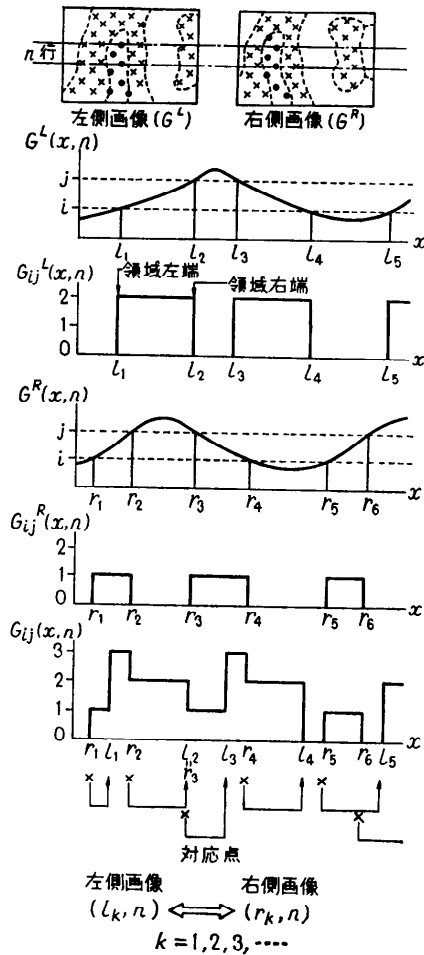


Fig. 3 An illustration of the region-method

$T(x, n)$  で、また参照画像内の点  $(x, n)$  の両眼視差を  $r(x, n)$  で表わす。類似度評価関数の閾値を  $\theta$  とし、設定された画像中の最大視差を  $r_{max}$  で表わす。

〔領域端の対応づけアルゴリズム〕

ステップ 0 (初期行の設定)

$n := 1$ .

ステップ 1 (初期値設定)

$x := 0, t_1 := 0, t_2 := 0, b_1 := 1, b_2 := 1$ , 定義 1 で述べた (2) のヒストグラムに基づく方法によって求められた濃淡レベルの分割両端の値を順次  $(i, j)$  に割り当てる。すべての割り当てを終了していれば、現在の行に対する処理を終了し、 $n := n + 1$  とし次行の処理に移る。このとき  $n = N + 1$  であれば画像全体に対する対応づけの処理を終了する。

ステップ 2 (走査点の移動)

$s=0, x=x+1$ .  $x=M+1$  ならば設定された  $(i, j)$  に関する第  $n$  行の領域  $A_{ij}^L$  と領域  $A_{ij}^R$  との対応づけを終了し, 新たな  $(i, j)$  を設定するためにステップ 1 へ.

### ステップ 3 (変化点の判定)

合成領域関数  $G_{ij}$  に関して点  $(x, n)$  が変化点でなければステップ 2 へ.

### ステップ 4 (変化点の型判定 [ $C_L^1$ ])

$T(x, n) \in C_L^1$  ならば  $k=1$  とし, ステップ 8 へ.

### ステップ 5 (変化点の型判定 [ $C_L^2$ ])

$T(x, n) \in C_L^2$  ならば  $k=2$  とし, ステップ 8 へ.

### ステップ 6 (変化点の型判定 [ $C_R^1$ ])

$T(x, n) \in C_R^1$  ならば  $t_1=t_1+1, Q_1(t_1)=x$  とし, ステップ 2 へ.

### ステップ 7 (変化点の型判定 [ $C_R^2$ ])

$T(x, n) \in C_R^2$  ならば  $t_2=t_2+1, Q_2(t_2)=x$  とし, ステップ 2 へ.  $T(x, n) \notin C_R^2$  ならば, そのままの状態ですステップ 2 へ.

### ステップ 8 (対応点が存在しない場合)

$t_k < b_k$  ならば  $b_k=b_k-s$  とし, ステップ 6 へ.

### ステップ 9 (対応候補点の決定)

$x-Q_k(b_k) \leq r_{\max}$  ならば, 右側画像内の点  $(Q_k(b_k), n)$  に対して左側画像内の点  $(x, n)$  が対応候補点である. 両点の近傍間での類似度評価関数  $\varphi_3(x-Q_k(b_k); Q_k(b_k), n)$  を計算した後に  $b_k=b_k+1$  とする.  $x-Q_k(b_k) > r_{\max}$  ならば  $b_k=b_k+1$  とし, ステップ 8 へ.

### ステップ 10 (対応の可否操作)

$\varphi_3(x-Q_k(b_k); Q_k(b_k), n) \geq \theta$  ならば両点を対応づける. 両眼視差は  $r(Q_k(b_k), n) = x - Q_k(b_k)$  である.  $T(x, n) = (0 \rightarrow 3)$  または  $(2 \rightarrow 1)$  ならば右側画像内の領域左端と左側画像内の領域端との重複を意味する. したがって,  $t_1=t_1+1, Q_1(t_1)=x$  とし, ステップ 2 へ.  $T(x, n) = (1 \rightarrow 2)$  または  $(3 \rightarrow 0)$  ならば右側画像内の領域右端と左側画像内の領域端との重複を意味するため,  $t_2=t_2+1, Q_2(t_2)=x$  とし, ステップ 2 へ.  $T(x, n) = (0 \rightarrow 3), (2 \rightarrow 1), (1 \rightarrow 2), (3 \rightarrow 0)$  ならば, そのままの状態ですステップ 2 へ.

$\varphi_3(x-Q_k(b_k); Q_k(b_k), n) < \theta$  ならば両点の対応を誤対応として棄却し,  $s=s+1$  の後, 両探索のためにステップ 8 へ.

なお, 各ステップにおいて行き先が明確に指定されていない場合には, それに続くステップに進むものとする. 以上の 0~10 のステップによって両眼画像内の領域端が対応づけられるが, 画像全体の対応づけを終

了した後に, 次の操作を加えることによって, さらに誤対応を取り除く. すなわち, 三次元空間における領域端の連続性を仮定し, 画像内の  $X$  軸,  $Y$  軸および視差 ( $R$ ) より構成される三次元空間  $(X, Y, R)$  において孤立した対応点——26 近傍内に他の対応点が存在しないような対応点——を除去する.

ここに示した対応づけアルゴリズムは, いくつかの  $i, j$  の組に対して各組ごとに対応づけを行うというものであり, (1) 未知物体の認識に際して, 画像内の多数の点に関する距離を高速に抽出することを意図している. また, (2) 大体の濃淡レベルが判っている特定の物体の距離を得たい場合には, その物体の濃淡レベルをさむように  $i, j$  を設定することによってある程度まで対処できる. このように距離情報を得る 2 つの目的に対して利用できる点が本方法の相関法と異なる特徴の 1 つである.

## 4. 計算機による実行結果

2. において述べたカメラ・モデルから得られる両眼画像に対して, 領域法によって対応づけを行い, 両眼視差を検出するプログラムを作成し, 計算機で実行した結果を示す. 次に対応づけの部分に相関法に換えたプログラムを作成し, 処理時間に関して領域法と相



Left image Right image  
Fig. 4 Input stereoscopic images (Example 1)  
(size\*: 329×157)

\* この図では出力の都合上, 縦 1/2, 横 1/3 に縮小されており, サイズは 109×78 になっている.



Left image Right image  
Fig. 5 Input stereoscopic images (Example 2)  
(size\*: 393×169)

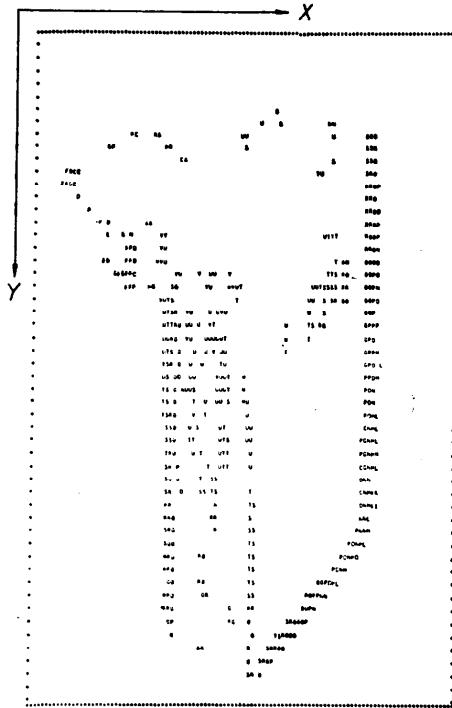
\* この図では出力の都合上, 縦 1/3, 横 1/4 に縮小されており, サイズは 98×56 になっている.

関法との比較を行う。プログラムは画像入力段階ではシンボリック・アセンブリ言語 SABR, 対応づけの部分および以後の出力段階では FORTRAN で書かれている。画像入力装置としてはフライング・スポット・スキャナ (FSS) KOWA-OS 701 を使用し、濃淡 128

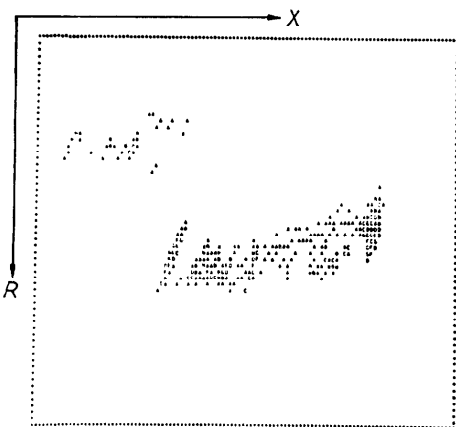
レベル ( $p=128$ ) で入力している。使用計算機は PDP-8/E 相当のミニコン CEC-555 H である。

入力両眼画像 Fig. 4, Fig. 5 (前頁参照) に領域法を適用したときの視差検出結果を Fig. 6(a), Fig. 7(a) に、またそれから得られる奥行き像を Fig. 6(b), Fig. 7(b) に示す。

$i, j$  の決定方法としては、先に定義 1 で述べたように 2 種類考えられ、両方について実験した結果、視差検出結果にはほとんど差異が認められなかった。なお、ここに示した出力結果はヒストグラムに基づいて  $i, j$  を決定した例であり、Fig. 4 に対しては  $(i, j) = (25, 35), (46, 57), (61, 72), (78, 88), (98, 110), (124,$

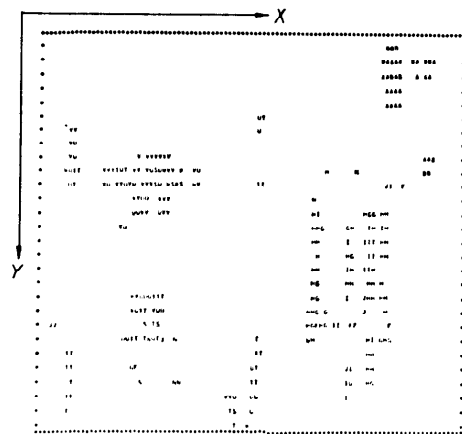


(a) Result of disparity detection

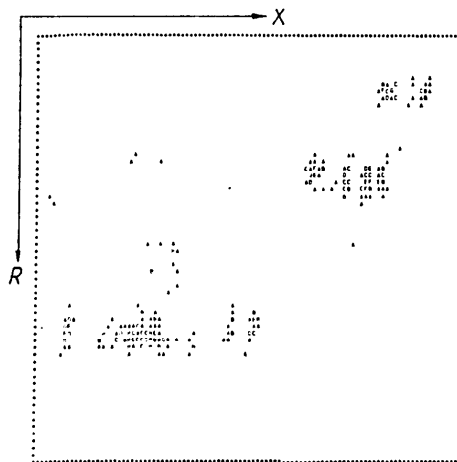


(b) Depth image obtained from (a)

Fig. 6 Application of region-method to example 1  
Similarity threshold  $\theta=0.950$



(a) Result of disparity detection



(b) Depth image obtained from (a)

Fig. 7 Application of region-method to example 2  
Similarity threshold  $\theta=0.950$

127) の6種類, また Fig. 5 に対しては  $(i, j) = (2, 17), (33, 50), (62, 74), (90, 108), (114, 119)$  の5種類設定している.

Fig. 6, Fig. 7 いずれの場合も, 類似度評価の閾値  $\theta$  を経験的に 0.950 に設定している.  $\theta$  を大きく設定すると, 誤対応が除かれると同時に正しい対応点をも消去してしまうという逆効果を伴うため, 画像内の多数の点に関する距離情報を得るという本方法の目的から, ここでは, いくらか誤対応が残る程度に  $\theta$  を設定し, 最後に境界の連続性から, 孤立した対応点を消去するという方法をとっている.

Fig. 6(a), Fig. 7(a) の両眼視差検出結果において, アルファベットは両眼視差の大小を表現している. すなわち, 文字Aは視差が最小の点—カメラから最も速くの点—を, 文字Zは視差が最大の点—最も近く点—を表わしている.

Fig. 6(b), Fig. 7(b) は視差検出点の X-R 平面への投影像であり, 図中の文字は視差検出点の度数を表わしている—例えば, A=1, B=2, ..., Z=26—すなわち, X-R に関する視差検出点の2次元ヒストグラムと考えることができる.

Fig. 4 の入力画像に関するプログラム中の最大視差設定\* ( $r_{max}$  の設定) と処理時間/対応点数との関係を Fig. 8 に示す. Fig. 4 では, 最適な最大視差\*\*は 47 画素であり, 実際に両眼視差を検出するにはこれよりも小さくない値に  $r_{max}$  を設定する必要がある.

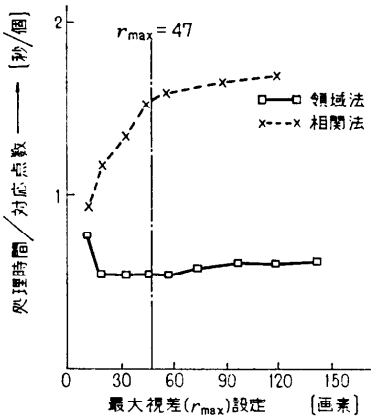


Fig. 8 Relation between the maximum disparity  $r_{max}$  and the computational time / the number of correspondence—points for example 1—

\* 入力両眼画像に対して, プログラム中で指定する仮想的な最大視差であり, 必ずしも実際の最高視差と一致しているとは限らない.  
 \*\* 画像内の点の実際の最大視差.

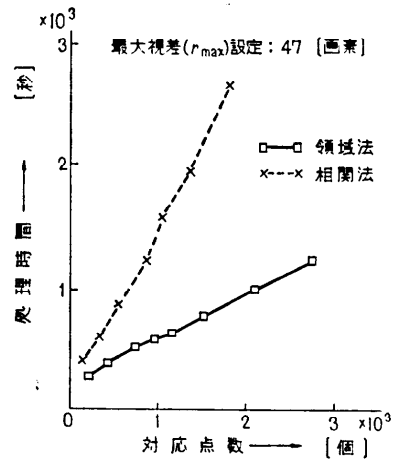


Fig. 9 Relation between the number of correspondence points and the computational time—for example 1—

Fig. 8 から明らかなように,  $r_{max}$  を 47 画素以上に設定すると領域法は相関法に較べて処理時間でほぼ 1/3 に改善されている.

次に  $r_{max}$  を最適な 47 画素に設定した場合の対応点数と処理時間との関係を Fig. 9 に示す. 対応点数の変化は, 領域法では対応づけ領域数 ( $i, j$  の設定回数) の増減によって, また相関法では特徴点抽出における閾値の変化によって起こっている.

### 5. あとがき

両眼視を利用して二次元である2つの画像から三次元情報(距離情報)を抽出する場合に問題となる左右画像内の物体の対応づけの方法として領域法を提案し, 従来の相関法との比較実験を行った.

領域法は各特徴点毎の類似度評価関数の計算回数を減少させることによる処理時間の短縮を意図している. 対応づけの方法は一見複雑になっているけれども, これに要する時間は相関法に較べて改善されており, 初期の目標をほぼ達成していると考えられる. 相関法による処理時間は画像内の点の最大視差が大きくなるに従って増大するのに較べて, 領域法の場合には処理時間が最大視差にほとんど無関係である.

領域法を用いた場合の対応づけの精度は対応づけ領域の切り出しの精度に依存するため, 対応づけ領域の決定に際して, 単なる濃淡レベル  $i, j$  による領域切り出し以外に, カラー情報やテキスト情報を利用することが考えられる. このことによって, より正確な

領域の切り出しが可能となり、誤対応を減らすことができるであろう。しかし、処理時間の増加との兼ね合いを考慮する必要がある。

これまでの実験結果から考えると、領域法では、画像入力装置の劣化および左右画像用写真の状態を考慮に入れても、少なくとも物体を識別するというレベルの距離情報は得られることが判った、しかしながら、直ちに距離の精密測定に応用できるとは言い難い。したがって、画像処理・認識システムの補助手段として次のような利用方法に適していると考えられる。

(1) [物体のセグメンテーションにおいて]

距離情報を用いることによって、部分的に重なって見えている物体の三次元的なセグメンテーションが可能となり、表面模様を別の物体と見なすことなく、物体を完全に識別することができる。Fig. 6 (b), Fig. 7 (b)はこの一例である。

(2) [認識において]

距離情報は、認識対象の一般的な意味構造をも考慮に入れた高次の認識においても、物体のもつ属性の1つとして、推論に利用できると考えられる。

謝辞 最後に、本研究を行うに際し、いろいろと示唆して下さい大阪大学基礎工学部の豊田順一助教授並びに田村進一助教授に深謝します。また、日頃有益な御助言をくださる田中研究室の諸氏に感謝します。

## 参 考 文 献

- 1) 安江利一, 白井良明: 両眼立体視による物体認識の方式, 信学会全国大会予稿集, No. 1618, (1974).
- 2) M. J. Hannah: computer interactive picture processing-Appendix A, Stanford Artificial Intelligence Projects Memo, AIM-166, (1972).
- 3) M. D. Levine, D. A. O'Handley & G. M. Yagi: computer determination of depth maps, computer graphics and image processing, vol. 2, No. 2, pp. 131~150, (1973).
- 4) K. Mori, M. Kidode & H. Asada: An Iterative Prediction and Correction Method for Automatic Stereocomparison, computer graphics and image processing, vol. 2, No. 3/4, pp. 393~401, (1973).
- 5) 北橋忠宏, 横矢直和, 田中幸吉: 両眼立体視に伴う2,3の問題について, 情報処理学会第16回大会講演論文集, No. 176, (1975).
- 6) 白井良明: 物体をどこまで機械で認識できるか, 日経エレクトロニクス, 1975.1.27, pp. 137~165, (1975).
- 7) 横矢直和, 北橋忠宏, 田中幸吉: 簡単なデータ表現を用いた両眼視に基づく三次元情報の抽出, 情報処理学会第16回大会講演論文集, No. 177, (1975).

(昭和51年2月19日受付)

(昭和51年8月2日再受付)