

# 図面からの線要素と塊状要素の分離

金子 透 若菜 忠

日本電信電話公社 横須賀電気通信研究所

## [1] まえがき

計算機による図面認識は図面の入力処理から文字認識に至るまでの広範な技術を含んでおり、各個別技術について様々な研究が行なわれている。これら個別技術の一つに、図面からの線要素と文字・記号の分離がある。この分離技術については、例えば連結領域に同一ラベル値を与え、領域の大きさや形状により分離する方法が考えられ、また図形要素の複雑さに着目して分離を行なう近接線密度法が提案されている<sup>1)</sup>。しかしこれらの手法は、図形要素に重なりがある場合に分離が困難になるという問題点があった。

筆者らは、図面中に存在する文字・記号は線要素に比べて大きさの制限されている塊状要素であると仮定し、これを線要素と塊状要素に重なりがある場合でも両者を分離する手法について検討した。検討の際には、境界追跡などの判断処理による原画へのランダムアクセスをなくし、ハードウェアによる高速処理が容易に実現できることを特に考慮した。その結果、新たな手法として、「往復走査分離法」(Shuttle-Scan Segmentation Method, 3S法)を考案し、実験により本方式が基本的に有効であることを検証した。以下、本方式の原理と実験結果について報告する。

## [2] 往復走査分離法の原理

本方式は、画面をラスタ走査しながら各画素一つ一つに特性値を与え、その特性値により各画素が如何なる性質の図形要素に属しているかを基本原則として決定することと基本原則としている。その処理の流れを、図1を用いて、以下に示す。

### ① ラスタ上の各画素

の特性値  $U_{fwd}$  を、既に走査したラスタ上の近傍画素の特性値を参照して決定する。この動作を1画素分走査しながら行なう。(走査経路)

### ② 上記動作を画面を逆戻りする方向で走査しながら実施し、各画素の新たな特性値 $U_{bwd}$ を得る。(走査復路)

### ③ 特性値 $U_{bwd}$ を閾値処理することにより、各々の画素の、副走査方向に関する性質を決定する。

ここで①～③の動作を、それぞれ直交する副走査方向について実施すれば、各

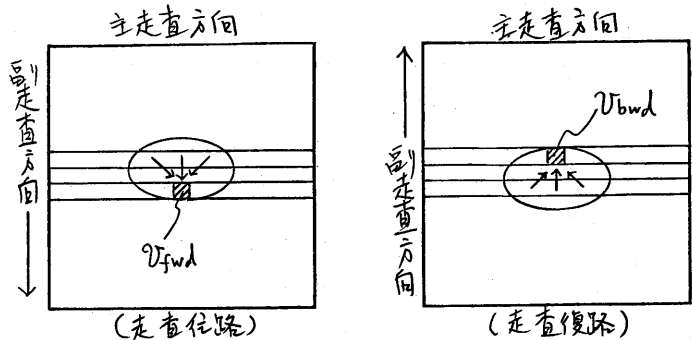


図1. 往復走査分離法の原理

画素の2次元的存在性が得られることにたす。

③ 線要素と塊状要素の分離

往復走査分離法において、画素の特性値の与え方および閾値処理の仕方は、分離すべき図形要素の範囲をどうとるかに依存する。

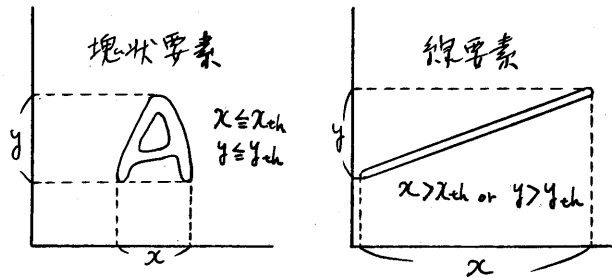


図2. 線要素と塊状要素の定義

ここでは、図2のように、図形要素の画面水平方向および垂直方向への1次元投影像の長さが共にある程度以下に短いものを塊状要素、少くともいずれか一方が長いものを線要素と定義する。ここで画素の特性値に、その画素の属する図形要素の1次元投影像の長さを反映するような値を与えることにする。

具体的な画素の特性値決定法は、線要素と塊状要素が重なりあつた場合、およびノッチ雑音の影響を考慮して、次のようにする。ここにノッチパラメータはノッチ雑音の大きさに応じた適当な定数である。なお原画は白黒2値画像であり、予め黒画素の初期特性値を1としておく。また白画素の特性値は終値0である。(走査経路の黒画素)

直前の走査線上の近傍3画素中、ノッチパラメータより大きい特性値を選び、このうちの最小値に1を加えた値を特性値として与える。近傍3画素の特性値が全てノッチパラメータ以下である場合には、これらの最大値に1を加えた値を特性値として与える。

(走査復路の黒画素)

直前の走査線上の近傍3画素中、ノッチパラメータより大きい特性値を選び、このうちの最小値を往路で自分自身に与えられた特性値の大きい方を、新たな特性値として与える。近傍3画素の特性値が全てノッチパラメータ以下である場合は、これらのうちの最大値を往路で自分自身に与えられた特性値の大きい方を、新たな特性値として与える。

以下に、上記アルゴリズムを数式的に表わす。

まず図3のように、画面上の座標系を、左上を原点として、横軸をx軸、縦軸をy軸と定める。また画面の走査方向に関して表1のような定義をする。ここに

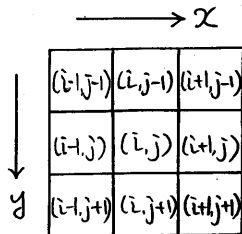


図3. 画面座標系

表1. 画面の走査方向の定義

	特性値	主走査方向	副走査方向
下方走査	$V_{dw}$	$x$	$y$ (正)
上方走査	$V_{up}$	$x$	$y$ (負)
右方走査	$V_{rg}$	$y$	$x$ (正)
左方走査	$V_{rl}$	$y$	$x$ (負)

$V_{dw}, V_{rg}$  は図1の  $V_{fwd}$  に、 $V_{up}, V_{rf}$  は図1の  $V_{bwd}$  に相当する。また主走直方向に  
ついては、軸方向の正負は任意である。

原画上の位置  $(i, j)$  にある画素  $P(i, j)$  の初期特性値を  $v(i, j)$  とする。

$$v(i, j) = \begin{cases} 1 & ; \text{black pixel} \\ 0 & ; \text{white pixel} \end{cases} \quad (1)$$

となる。

画素  $P(i, j)$  の下方向走査後の特性値  $V_{dw}(i, j)$  は、

$$V_{dw}(i, j) = v(i, j) \cdot (V_{dw}^0(i, j) + 1) \quad (2)$$

となる。但しここで

$$V_{dw}^0(i, j) = \begin{cases} \min\{V_{dw}(k, j-1) \mid V_{dw}(k, j-1) > v_n, i-1 \leq k \leq i+1\} & ; \text{if any} \\ \max\{V_{dw}(k, j-1) \mid i-1 \leq k \leq i+1\} & ; \text{otherwise} \end{cases} \quad (3)$$

である。同様にして、

$$V_{up}(i, j) = v(i, j) \cdot \max\{V_{dw}(i, j), V_{up}^0(i, j)\} \quad (4)$$

$$V_{up}^0(i, j) = \begin{cases} \min\{V_{up}(k, j+1) \mid V_{up}(k, j+1) > v_n, i-1 \leq k \leq i+1\} & ; \text{if any} \\ \max\{V_{up}(k, j+1) \mid i-1 \leq k \leq i+1\} & ; \text{otherwise} \end{cases} \quad (5)$$

$$V_{rg}(i, j) = v(i, j) \cdot (V_{rg}^0(i, j) + 1) \quad (6)$$

$$V_{rg}^0(i, j) = \begin{cases} \min\{V_{rg}(i-1, k) \mid V_{rg}(i-1, k) > v_n, j-1 \leq k \leq j+1\} & ; \text{if any} \\ \max\{V_{rg}(i-1, k) \mid j-1 \leq k \leq j+1\} & ; \text{otherwise} \end{cases} \quad (7)$$

$$V_{rf}(i, j) = v(i, j) \cdot \max\{V_{rg}(i, j), V_{rf}^0(i, j)\} \quad (8)$$

$$V_{rf}^0(i, j) = \begin{cases} \min\{V_{rf}(i+1, k) \mid V_{rf}(i+1, k) > v_n, j-1 \leq k \leq j+1\} & ; \text{if any} \\ \max\{V_{rf}(i+1, k) \mid j-1 \leq k \leq j+1\} & ; \text{otherwise} \end{cases} \quad (9)$$

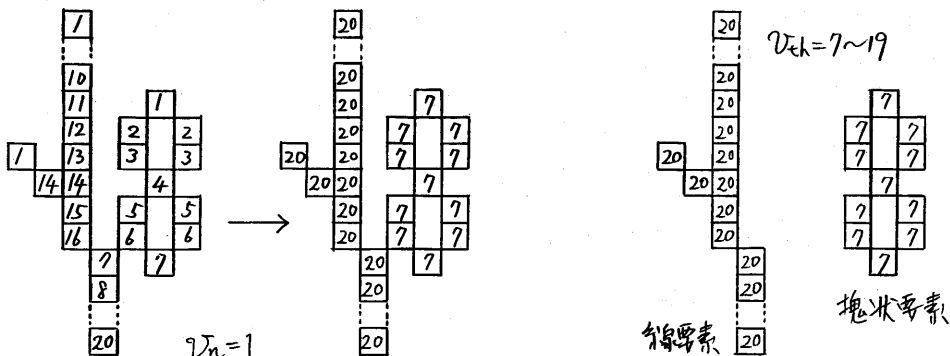
となる。なお、(3)、(5)、(7)、(9)式において  $v_n$  は  $1 \sim 4$  のラック - 1 である。また上記において、画面の境界部分の画素については近傍画素 2 画素のみを参照するこ  
とができる。

以上のごとくして求められた特性値を用いて、画素  $P(i, j)$  の属性を次のように  
定める。

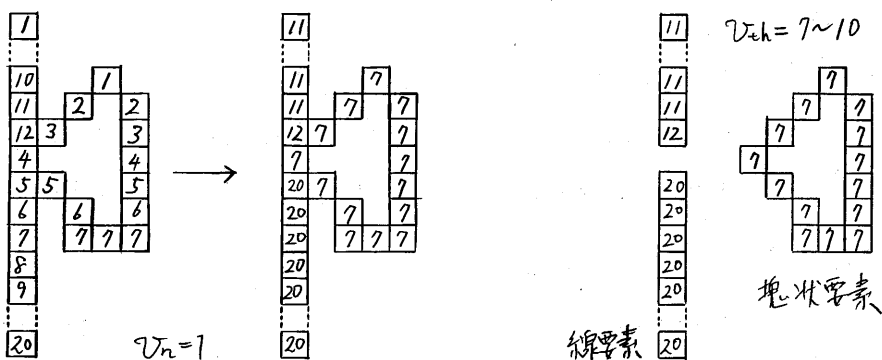
$$P(i, j) = \begin{cases} \text{線要素} & ; \text{if } V_{up}(i, j) > v_{ch}, \text{ or } V_{rf}(i, j) > v_{ch} \\ \text{塊状要素} & ; \text{if } 1 \leq V_{up}(i, j) \leq v_{ch}, \text{ and } 1 \leq V_{rf}(i, j) \leq v_{ch} \\ \text{白地画素} & ; \text{if } V_{up}(i, j) = V_{rf}(i, j) = 0 \end{cases}$$

上式において  $v_{ch}$  は図形要素分離のため、適当な閾値であり、具体的には画面中  
に存在する最大の文字・記号サイズに対応した値である。

本方式により線要素と塊状要素が分離される様子を、図4に模式的に示す。図  
4の(1)は線要素と塊状要素が接している例であり、分離閾値  $v_{ch}$  を  $7 \sim 19$  にこれ



(1) 図形要素が接している例



(2) 図形要素が重なりあっている例

図4. 線要素と塊状要素が分離される様子

ば線要素と塊状要素が完全に分離できる。一方、図4の(2)は線要素と塊状要素が重なりあっている例であり、 $v_{ch}=7\sim 10$ とするこにより塊状要素が分離できる。しかしこのとき線要素の一部が塊状要素がわに切りとられることにある。なお、図4の(1)に、線要素に付随しているノッチ雑音の影響が除去される様子も示されている。

[4] 領域の大きさによる閉領域の分離

往復走査分離法では、白地に着目することにより、領域の大きさを基準とした閉領域の分離が行われる。

画素を走査しながら画素の特性値を決定していくにあたっては、白地画素が互いに斜めのみで連結している場合は両者は元来別々の閉領域に属しているのだから、このような場合は互いに特性値が伝搬しないようにする必要があり。

従ってここでは、図5に示すように、着目している白画素から見て4連結である近傍画素が連続して黒画素である場合、着目白画素から見てこれら近傍黒画素の中間に位置する8連結の近傍画素の特性値は参照しないこととする。

上記条件のもとに、[3]の線要素と塊状要素の分離アルゴリズムを用いる。但し

ここでは白地画素に正の特性値を与え、黒画素の特性値は常に0とする。また図では、線要素と塊状要素が重なりあ、た場合に後者が確実に分離されるように、着目画素の直前走査線上の近傍画素中ノッチパラメータより大きな特性値の最小値をとりこころとしたが、ここでは閉領域同士に重なりあいがないうとして、近傍画素の特性値の最大値を着目画素の特性値にこころこころとする。以下これを数式で表わす。

まず原画上の位置  $(i, j)$  にある画素  $P(i, j)$  の初期特性値  $v(i, j)$  を、

$$v(i, j) = \begin{cases} 1 & ; \text{white pixel} \\ 0 & ; \text{black pixel} \end{cases} \quad (11)$$

とする。

画素  $P(i, j)$  の下へ走査したのうの特性値  $v_{dw}(i, j)$  は、

$$v_{dw}(i, j) = v(i, j) \cdot (v_{dw}^0(i, j) + 1) \quad (12)$$

$$v_{dw}^0(i, j) = \max \{ v_{dw}(k, j-1) \mid i-1 \leq k \leq i+1 \} \quad (13)$$

$$v_{dw}(i-1, j-1) = \begin{cases} v_{dw}(i-1, j-1) & ; \text{if } v(i-1, j) \neq 0 \text{ or } v(i, j-1) \neq 0 \\ 0 & ; \text{if } v(i-1, j) = v(i, j-1) = 0 \end{cases}$$

$$v_{dw}(i+1, j-1) = \begin{cases} v_{dw}(i+1, j-1) & ; \text{if } v(i+1, j) \neq 0 \text{ or } v(i, j-1) \neq 0 \\ 0 & ; \text{if } v(i+1, j) = v(i, j-1) = 0 \end{cases} \quad (13)'$$

となる。

同様にして、画素  $P(i, j)$  の上へ走査、右へ走査、左へ走査したのうの特性値  $v_{up}(i, j)$ 、 $v_{rg}(i, j)$ 、 $v_{lf}(i, j)$  はそれぞれ、次のようになる。

$$v_{up}(i, j) = v(i, j) \cdot \max \{ v_{dw}(i, j), v_{up}^0(i, j) \} \quad (14)$$

$$v_{up}^0(i, j) = \max \{ v_{up}(k, j+1) \mid i-1 \leq k \leq i+1 \} \quad (15)$$

$$v_{up}(i-1, j+1) = \begin{cases} v_{up}(i-1, j+1) & ; \text{if } v(i-1, j) \neq 0 \text{ or } v(i, j+1) \neq 0 \\ 0 & ; \text{if } v(i-1, j) = v(i, j+1) = 0 \end{cases}$$

$$v_{up}(i+1, j+1) = \begin{cases} v_{up}(i+1, j+1) & ; \text{if } v(i+1, j) \neq 0 \text{ or } v(i, j+1) \neq 0 \\ 0 & ; \text{if } v(i+1, j) = v(i, j+1) = 0 \end{cases} \quad (15)'$$

$$v_{rg}(i, j) = v(i, j) \cdot (v_{rg}^0(i, j) + 1) \quad (16)$$

$$v_{rg}^0(i, j) = \max \{ v_{rg}(i-1, k) \mid j-1 \leq k \leq j+1 \} \quad (17)$$

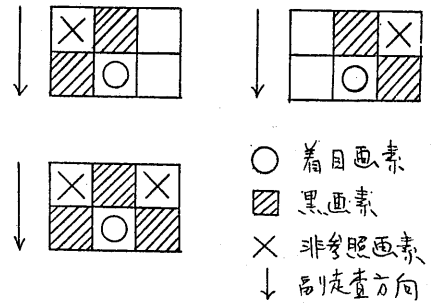


図5. 白地画素に対する非参照近傍画素

$$V_{rg}'(i-1, j-1) = \begin{cases} V_{rg}(i-1, j-1) & ; \text{if } V(i-1, j) \neq 0 \text{ or } V(i, j-1) \neq 0 \\ 0 & ; \text{if } V(i-1, j) = V(i, j-1) = 0 \end{cases} \quad (17')$$

$$V_{rg}'(i-1, j+1) = \begin{cases} V_{rg}(i-1, j+1) & ; \text{if } V(i-1, j) \neq 0 \text{ or } V(i, j+1) \neq 0 \\ 0 & ; \text{if } V(i-1, j) = V(i, j+1) = 0 \end{cases}$$

$$V_{2f}(i, j) = V(i, j) \cdot \max\{V_{rg}(i, j), V_{2f}^0(i, j)\} \quad (18)$$

$$V_{2f}^0(i, j) = \max\{V_{2f}(i+1, k) \mid j-1 \leq k \leq j+1\} \quad (19)$$

$$V_{2f}(i+1, j-1) = \begin{cases} V_{2f}(i+1, j-1) & ; \text{if } V(i, j-1) \neq 0 \text{ or } V(i+1, j) \neq 0 \\ 0 & ; \text{if } V(i, j-1) = V(i+1, j) = 0 \end{cases} \quad (19')$$

$$V_{2f}(i+1, j+1) = \begin{cases} V_{2f}(i+1, j+1) & ; \text{if } V(i, j+1) \neq 0 \text{ or } V(i+1, j) \neq 0 \\ 0 & ; \text{if } V(i, j+1) = V(i+1, j) = 0 \end{cases}$$

以上のごとくして求められた特性値のうち \$V\_{rg}(i, j)\$ および \$V\_{2f}(i, j)\$ を調べることにより、画素 \$P(i, j)\$ が \$x\$ 軸方向および \$y\$ 軸方向にそれぞれ大きさを有する閉領域に属しているかについての情報を得ることが出来る。

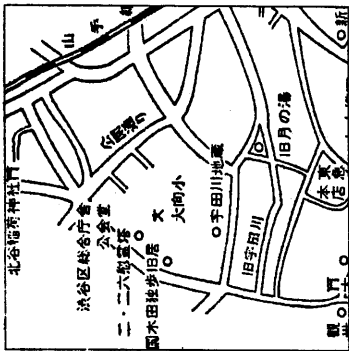
### [5] 実験例

ドラムスキャナが入力した 2, 3 の図面を対象に、本方式による図形要素分離の実験を行った。図 6~8 にその実験結果を示す。

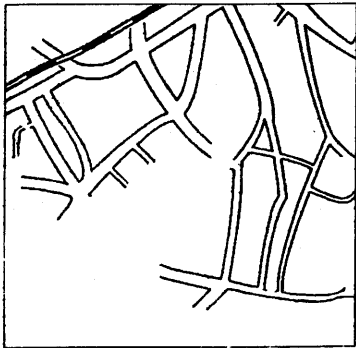
図 6 は、図形要素に重なりが無い場合の線要素と塊状要素の分離例である。分離のノッチパラメータは 31、分離閾値は 25 である。ノッチパラメータに分離閾値よりも大きな値を与えておいて、(3), (5), (7), (9) 式において常に近傍画素の特性値の最大値とし、このことに注意。図に示すようにほぼ良好な分離結果を得ているが、本アルゴリズムでは \$x\$ 軸と \$y\$ 軸の特性値を独立に扱っているため連続した道路でも切れている部分がある。また傾きが \$45^\circ\$ に近い線において、線の縁が一部塊状要素とみなされてしまっている。

図 7 は、図形要素に重なりがある場合の線要素と塊状要素の分離例である。分離のノッチパラメータは 1、分離閾値は 22 である。ノッチパラメータが小さい値のため、(3), (5), (7), (9) 式において強引な場合近傍画素の特性値の最小値とし、このことに注意。従って線要素と塊状要素が重なりあっている部分は塊状要素がわいてしまいがちになり、線要素に何箇所か欠損が見られる。線要素でありながら縦軸の先端についている矢印も塊状要素とみなされている。一方局所的には、その部分の図形的性質により塊状要素が線要素に吸収されているところもある。

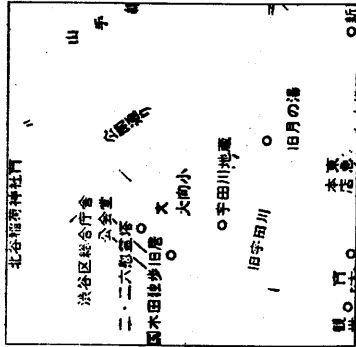
図 8 は、白地に着目して図面中にある一定の大きさの円を分離した例である。円分離の閾値は、下限を 23、上限を 51 とした。図に示すように、円内がほぼ白地である場合は円が良好に分離できる。一方、図の上辺部にある数字 "30" および "20" のそれぞれの間隙に、一部円内と同じ特性値を有する部分が生じたが、分離結果にはさらに経復走査分離法を適用することによりこれを除去することができた。



(1) 原画(林他「渋谷区」歴史, 名著出版) 中尺 512x512画素

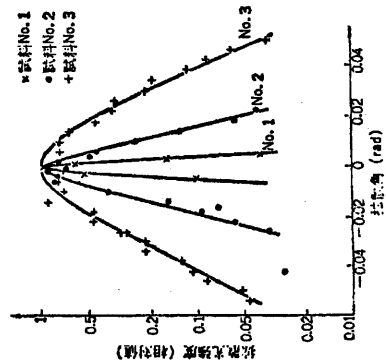


(2) 分離された線要素

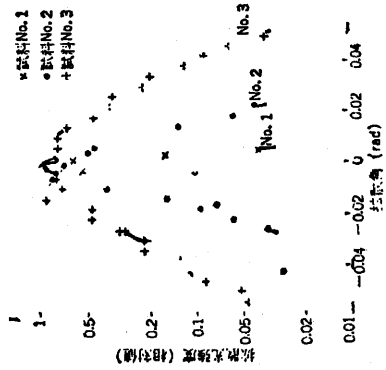


(3) 分離されたブロック要素

図6. 図形要素に集約が無い場合の処理例 ( $\sigma_n=31, \sigma_{th}=25$ )



(1) 原画(大越英敏, 三次元画像工学, 産業図書)



(2) 分離された線要素

図7. 図形要素に集約が存在する場合の処理例 ( $\sigma_n=1, \sigma_{th}=22$ )

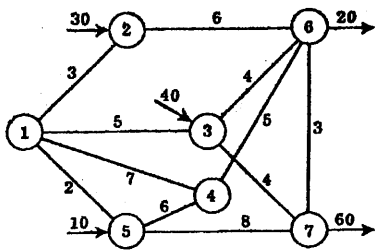
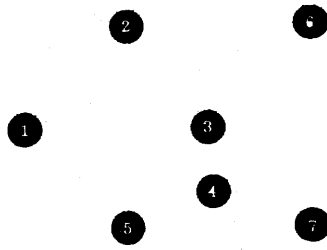


Figure P7-3



(1) 原画 (Au & Steelson; "Introduction to Systems Engineering, Deterministic Models", Addison-Wesley)

(2) 分離された円領域  
( $23 \leq U_{up}, U_{down} \leq 51$ )

図8. 白地に着目した図形要素分離例

[6] 今後の検討課題

図面に混在する線要素と塊状要素を分離する手法として「経路走査分離法」を提案した。本方式は線要素と塊状要素が重なりあっている場合でも有効なことを示し、また一方本方式を開領域の大きさにより図形要素の分離にも利用できることを示した。

今後本方式を図形要素の分離手法として確立するには、図形要素の図形的性質を詳細に調べて種々の場合に対処できるようにアルゴリズムを改良していく必要があるが、具体的には、線に断点がある場合や逆に線が太い場合の処理や、さらには直線の進行方向に重みをつけた特性値をとり入れて直線性の分離基準に付加する方法等を検討して行く予定である。

(謝辞)

本研究に御協力いただいた早稲田大学理工学部の大坪康郎氏に感謝します。また日頃御指導いただき当研究所画像応用研究室の米次室長および同研究室の皆様にも感謝します。

(参考文献)

- (1) 棟上, 佐藤; "図面の認識と理解の現状について", 情文学会, コンピュータビジョン研究会資料, 17-6, 1982年
- (2) 岩城, 久保田, 石井; "近接線密度法による文字・図形切り分け処理の検討", 通信学会技報, PRL81-81, 1981年