情　報　学　基　礎　55－3
デジタル・ドキュメント　19－3
（1999．7．16）

関数化図形表現を用いた紙文書のデジタル化

森 浩一*　和田 耕一**　寅市和男**

*筑波大学 工学研究科
**筑波大学 電子・情報工学系

あらまし

紙文書をデジタル化して利用する場合、一般的にはピクセル画像として扱われる。しかし、ピクセル画像では様々な出力デバイスの解像度を十分に活用した出力を得ることは難しい。これに対し関数化図形を用いた画像は、画像内の領域の輪郭線を近似連続関数で表現しているため小さいサイズの画像ファイルから任意の解像度で出力できる。本論文では、Dynamic Programming を用いて輪郭線の分割と分割された輪郭線の関数近似を行って関数化図形を生成する手法について述べる。いくつかの実験によって提案する手法が紙文書のデジタル化に適している事を示す。

Paper Document Digitization using Shapes Represented by Function Approximation

Koichi Mori*, Koichi Wada**, Kazuo Toraichi**

*Doctoral Program in Engineering, University of Tsukuba
**Institute of Information Sciences and Electronics, University of Tsukuba

Abstract

Although pixel image has been used for digitizing paper documents, pixel image hardly exploits resolution of various output devices. On the other hand, the image represented by function approximation can be output in arbitrary resolution, because boundaries in the image are represented by approximated continuous functions. In addition, The image represented by function approximation has merits of its compact file size and fast drawing. In this paper, a new method, which divides boundaries and approximates the each partial boundary using Dynamic Programming for creating the image represented by function approximated shapes, is introduced. Some experimental results are shown to ensure the efficiency of proposed method for digitizing paper documents.

1. まえがき

近年、デジタル文書の利用が活発化しているが、新聞、雑誌、地図、図面など紙媒体の文書も依然数多く存在している。紙文書をデジタル化することで伝送や保存における利点が得られる上に、すでに紙文書として存在する絵や図などの資産をデジタル文書に埋め込んで活用することが可能となる。

デジタル文書の場合、利用状況により出力デバイスは変化するので、出力デバイスに適した解像度を利用時に決定できなければならない。例えば、ディスプレイは低解像度なので全体を見るには画像を縮小し、詳細を見るには拡大する必要がある。また、プリンタはディスプレイに比べてはるかに解像度が高くなっており、高い精度での出力が要求される。既存の多くの場合、紙文書はスキャナやデジタルカメラで読み込まれピクセル表現の画像（ラスタ画像）ファイルとして扱われている。ラスタ画像ではピクセルの集合で画像を表現しているので、多様な出力デバイスに対してその解像度を十分に活用することは難しくなる。

これに対してベクタ画像のように、画像中のそれぞれの色領域の閉輪郭線で表現する画像では高精度な拡大縮小が可能であり、任意の解像度の出力デバイスに対応できる。輪郭線の集合で画像を表現する方式は、Web 上で用いる画像としても注目されており、W3C において SVG(Scalable Vector Graphics)として標準化が進められている。

輪郭線の集合で画像を表現する方式は、自然画像のように色のばらつきの多い画像には適さない。しかし、多くの文書画像では白黒の２値、あるいは非常に少ない色数で表現できる上に色の境界が明確であるため、紙文書をデジタル化する表現形式として、輪郭線の集合で画像を表現する方式が適している。

輪郭線の集合で表現される画像は、従来では主にアウトラインフォント、図形、図面のデータのように比較的単純な画像を対象に用いられてきたが、文書画像への適用例はほとんどみられなかった。その理由としては、ピクセル表現画像から自動変換された画像の品質が高くないために人間の手による修正が必要となり、輪郭線の量が多く複雑な文書画像には適用が難しいためである。また、自動変換された画像の品質が低下してしまう理由の一つとして、直線近似かスプライン近似のみを用いて輪郭線を表現していることが挙げられる。

我々はこれまでに、複数の近似関数を用いて輪郭線を近似した関数化図形をもとにして画像を表現する手法を提案してきた[1][2]。輪郭線を連続関数で近似表現することにより一つの画像ファイルから任意の解像度で出力することができる。また、輪郭線の部分毎に最適な関数で近似して再生に必要な最低限のデータで構成されているのでファイルサイズも小さくて済む上、再生・表示処理も非常に高速である。

適切な関数近似は再生画質やデータ量に深く関わる重要な処理であるが、[1][2]では輪郭のどの部分をどの関数で近似するかを決定する基準として曲率を用いている。曲率を用いれば理論的には円弧や直線となる部分を抽出できるが、実際の文書画像にはノイズが存在しているため曲率での判定は難しい。また曲率は非常に局所的な特徴であるため大局的に適切な関数近似を行うことができない。大局的にも適切な関数近似を行うには最適化手法を用いて近似関数を決定するべきである。

本論文では、Dynamic Programming(DP)を用いてフィッティングを主とするコスト関数を最小化するという枠組みで近似関数を決定する手法を提案し、スキャナやデジタルカメラで取りこんだ文書画像を自動処理で関数化表現画像に変換してデジタル化する手法について述べる

2. 関数化図形表現を用いた文書画像のデジタル化

2.1. 画像の関数化図形表現

対象とする画像である文書画像では明確に輪郭が存在しており２値化してそのエッジを追跡すれば容易に閉輪郭線が得られる。得られた輪郭点列は、$p_i = (x_i, y_i), i = 1, \ldots, L$ で表されており、前後の点との距離は 1 以下である。得られた輪郭点列は離散的であるが、これを連続関数で近似すれば任意の解像度での輪郭線出力が可能となる。図形関数化処理の流れの概要を図１に示す。

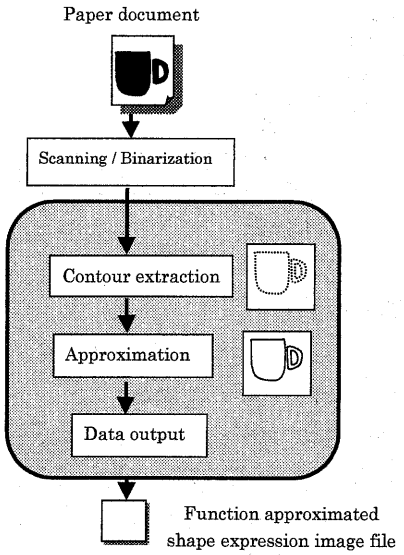


図１　図形関数化処理の流れ

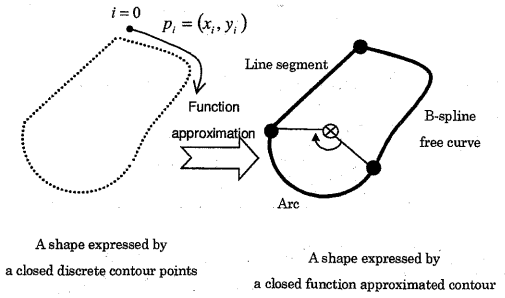


図２　輪郭の関数近似表現

既存の多くの輪郭線を近似する手法では、閉輪郭線１周分の全てを直線の組合わせやスプライン関数（ベジエ関数）などで近似しているが[3][4]、我々の提案する手法では輪郭線を分割しそれぞれの区間で最適な関数を用いて近似している。複数のクラスの関数を用いる事で単一のクラスの関数で近似するより少ないデータで近似できる事が見込まれる。我々の手法では、直線、２次Ｂスプライン曲線、円弧の組み合わせで輪郭線を表現している。図 2 の例では輪郭線が３つの区間に分割されており、それぞれＢスプライン、円弧、直線で近似されている。

画像ファイルとしては、近似関数の情報をもとにして近似輪郭線が再生可能な最低限のデータを保存する。画像全体に共通の情報として画像を構成する輪郭線数、個々の輪郭線に関数情報として輪郭線全体に関する情報と近似区間毎の近似関数に関する情報を保存する。直線区間であれば終点座標、円弧区間であれば中点と終点の座標、自由曲線区間であればＢスプライン係数が必要となる。

2.2. 関数化手法

区間が分かっていればこれまで述べてきたように複数クラスの連続関数を用いて輪郭線を近似することが可能となる。しかも該当区間に対する近似関数が適当でないときには近似に要するデータの増加を招いてしまうので、区間分割と近似関数を決定する処理は近似の精度やデータ量に大きく関わる重要な処理であるといえる。しかし、輪郭線の区分に関するヒントを人間が与えていては要求仕様である自動処理が達成できない。既存の多くの手法では曲率などを用いた角の抽出や近似関数の決定、局所的なフィッテングを用いて大まかな近似関数を決定し、後処理で区間の結合や分割を行って適切な区間分割を達成しようとしている[1][2][5][6]。処理が複数段になるのは、曲率や局所的なフィッティングでは全体的に整合性の取れた区間分割が行えないからである。このような 1 次元系列の信号の区間分割には最適化手法の一つである DP (Dynamic Programming) が有効である[7]。提案する手法では、最適化する関数を分割された区間でのフィッティングの誤差を元にした値の和を用い、輪郭線全体で最適なフィッティングを算出している。以下では DP を用いた輪郭線の区間分割と近似関数決定処理について述べる。

輪郭点列を $p_i=(x_i,y_i)$（$1\le i\le L$：輪郭線長）としたとき、これを $i=i_0,i_1,...,i_h,...,i_H$ で H 個の区間に分割することを考える。$p_{i_{h-1}}\cdots p_{i_h}$ をある関数で近似した時のコストを $d(i_{h-1},i_h)$ とし、全体のコストを区間の近似コストの和で表すと、

$$g(H)=\sum_{h=1}^{H} d\left(i_{h-1},i_h\right) \qquad \cdots(1)$$
$$i_0=1, i_h - i_{h-1} > l_{\min}$$

となる。$l_{\min}$は最小の区間長である。例として、図 2 の輪郭点列のp_{i_3}までを適当な 2 点で 3 分割した場合を図 3 に図示する。区間毎の近似コスト関数は、近似誤差を減らすことだけでなく、なるべく長い近似区間にするということを目標に設定されている。近似誤差を減らすということだけが目的なら、フィッティングの誤差をそのままコスト関数として用いることも考えられる。しかし、スプライン曲線は係数を増やすことで誤差を減らすことができてしまうので、フィッティングの誤差だけで判断すると全ての区間が自由曲線区間となってしまう。そこで、提案する手法では直線と円弧を優先的に決定し、どちらでもない区間を自由曲線区間とするようなコスト関数の設定にしている。ここで用いるコスト関数を式(2)に示す。

$$d(i_{h-1},i_h)=\begin{cases}\dfrac{\varepsilon_{arc}^2}{L_i}, \text{if } \varepsilon_{arc}^2<\varepsilon_{seg}^2 \ \&\ \dfrac{\varepsilon_{arc}^2}{L_i}<d_{thd}\\ \dfrac{\varepsilon_{seg}^2}{L_i}, \text{else if } \dfrac{\varepsilon_{seg}^2}{L_i}<d_{thd}\\ C_{spl}L_i, otherwise\end{cases} \quad \cdots(2)$$

ただし、$L_i=|i_h-i_{h-1}|$は区間長、ε_{arc}^2は円弧近似誤差、ε_{seg}^2は線分近似誤差、rは近似円弧の半径、をそれぞれ示している。d_{thd}は許容最大コストを表す定数で、これを超える場合にはその近似関数を用いないようにしている。現在は実験的に$d_{thd}=0.018$としている。C_{spl}はスプライン曲線近似に対するコストで、直線や円弧に対して優先的に選択されないように比較的大きな定数として$C_{spl}=0.02$に設定している。線分で近似したときの誤差ε_{seg}^2は端点$p_{i_h},p_{i_{h-1}}$を結ぶ直線との平均 2 乗誤差であり、近似直線を$ax+by+c=0$とするとその値は、

$$\varepsilon_{seg}^2=\frac{\sum(ax_i+by_i+c)^2}{a^2+b^2} \quad \cdots(3)$$

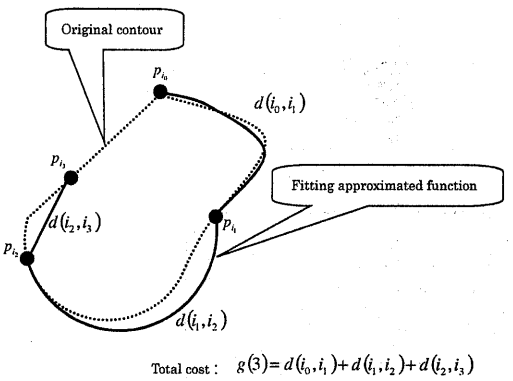


図 3　輪郭線近似コスト関数

となる（図 4(a)）。円弧近似誤差ε_{arc}^2は、部分点列に対する近似円との平均 2 乗誤差である。近似すべき区間の端点が図 4(b)のようにx軸上で対称に位置していれば、近似円の中心座標$(b,0)$は次式で求められる[8]。

$$b=\frac{\sum\left(x_i^2+y_i^2-a^2\right)}{2\sum y_i} \quad \cdots(4)$$

このとき、ε_{arc}^2は円弧の中心から輪郭点への距離から半径を引いたもので表し、

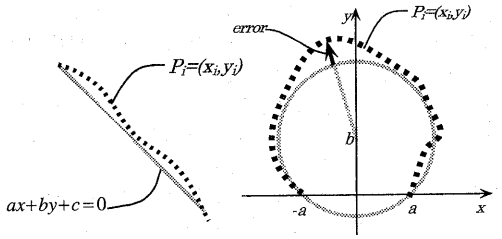
$$\varepsilon_{arc}^2=\frac{\sum\left(\sqrt{x_i^2+(y_i-b)^2}-\sqrt{a^2+b^2}\right)^2}{L_i} \quad \cdots(5)$$

となる。しかし、実際の輪郭点列は両端点がx軸上にのることは非常に稀なので、計算する際には両端点がそのような位置関係になるように点列を移動・回転させる必要がある。

最適な区間分割は、$i_H=L$となるときの全体のコスト$g(H)$を最小化することで得られる。漸化式で表すと、

$$\begin{aligned}g(H)&=\min\left(\sum_{h=1}^{H}d(i_{h-1},i_h)\right)\\ g(h)&=\min\left(g(h-1)+d(i_{h-1},i_h)\right)\\ &\qquad (g(0)=0)\end{aligned} \quad \cdots(6)$$

となる。これまで示したように、ある区間のコストはその区間での近似誤差と区間長をもとに計算されており、前後の区間の状態とは無関係に計算される。した



(a) Line fitting　　(b) Arc fitting

図4　近似関数フィッティング

がって、この最適化計算は DP を用いて効率的に計算することが可能となる。DP を用いたときの特徴は、H までの最適な区間分割を計算すれば、H 未満の最適な分割も同時に計算されることである。H の最大値は、区間の最低長さが l_{min} なので L/l_{min} 以下の最大の整数となる。最終的には、各 H におけるコスト関数を比較し最もコストの少なくなる H を選択し、そのときの区間分割と近似関数を用いてデータを出力する。

このように DP を用いた近似関数決定手法では、最適化手法に基づいて理論的に区間分割と近似関数の決定が行えるが、DP 計算量は長さの 2 乗に比例して増えてしまう。さらに、フィッティングの計算は長さに比例して計算量が増えるため、輪郭線が長くなった場合そのまま DP を適用するのは実用上問題がある。計算量を削減するために、長い輪郭線の場合は区切って処理している。

3. 実験

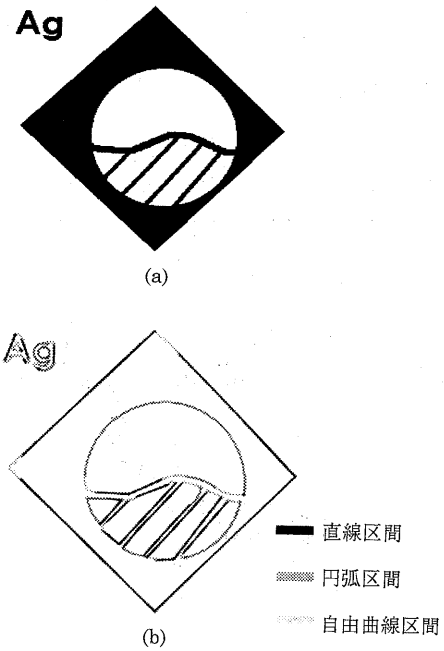
本章では、提案する関数化図形表現手法を用いて紙文書をデジタル化する実験を行い、その結果に対する考察を述べる。実験は全て、PC 上で行い、OS として Windows98 を使用した。

3.1. DP を用いた図形関数化手法の実験

提案する DP を用いた関数化手法が全体的に整合性の取れた区間分割を行えることを確認するために図 5(a)のような比較的シンプルかつノイズの少ない画像に対して関数化処理を行った。図 5(b)が関数化の結果である。近似輪郭線は、近似関数によって色分けしてある。大きな円弧はピクセル表現では部分的に直線になっているが、提案する手法では全体が正しく円弧として近似されている。外側の菱形の輪郭線では部分的に自由曲線で近似されているが、これは処理速度向上のために輪郭線を定間隔に区切って処理しているためこのような結果となった。

3.2. 紙文書デジタル化の実験

次に紙文書をスキャナで読み込みそれを入力画像として関数化図形表現画像ファイルを生成し、そのファイルサイズと出力される画像の品質について評価を行った。実験に用いた画像は図 6 の 5 つの文書画像である。いずれも紙の文書をスキャナで取り込んだもので、300dpi 程度の解像度の白黒 2 値画像である。



(a)

(b)

図5　DP を用いた関数化図形表現

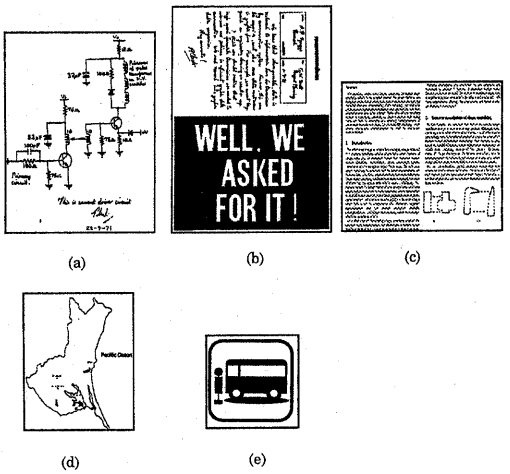


図 6　入力画像

ここではファイルサイズに注目して実験を行った。実験では図 6 の画像を入力として出力される関数化画像ファイルのサイズと同じ画像の GIF ファイルのサイズを比較した。この実験の結果を表 1 に示す。実験の結果から元のビットマップ画像と比べると 1.7%～17.6%程度の圧縮率であり、GIF と比べると 0.2 倍～0.7 倍程度のファイルサイズであり、GIF と比べて十分に小さいファイルサイズであるといえる。他と比べて圧縮率が低い画像(b)では、細かい文字が多いために細かく複雑な輪郭線が増加しファイルサイズの増加を招いている。

図 7 は関数化図形表現画像の出力例である。(b)は(a)の一部分を拡大したもので共にディスプレイ画面への表示例であり、(c)はプリンタへの出力結果である。関数化表現による画像はディスプレイ、プリンタの双方に対してその解像度を十分に生かしていることがわかる。(d)は関数化表現画像と比較するために、もとのピ

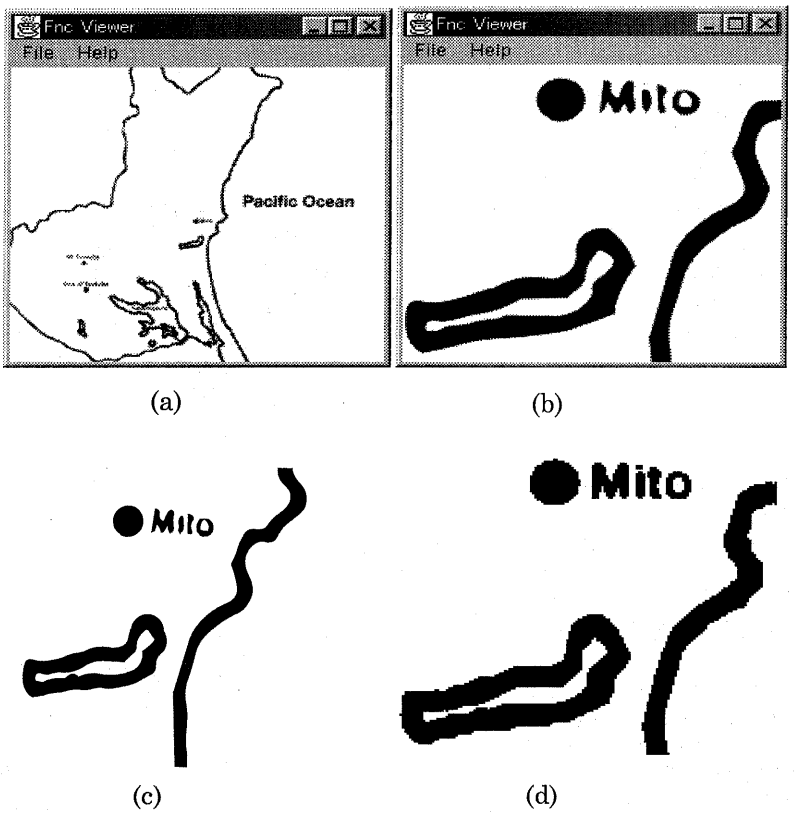


図 7　出力例

表1　ファイルサイズの比較

画像	画像(a)	画像(b)	画像(c)	画像(d)	画像(e)
画像サイズ	2186x2793	2288x3099	2198x1698	1515x1607	1060x1100
原画像ファイルサイズ	746 KB	866 KB	457KB	298 KB	147 KB
関数化画像ファイルサイズ	33 KB	66 KB	81KB	10.7 KB	2.5 KB
圧縮率	4.4 %	7.7 %	17.6%	3.6 %	1.7 %
GIF ファイルサイズ	50 KB	91 KB	124KB	18.9 KB	12.2 KB

クセル画像を拡大表示したものである。拡大した画像では明らかに関数化表現画像の方が高精細な出力が得られている。

以上の実験から関数化図形表現画像は紙文書をデジタル化して用いる画像として、再生品質においてもファイルサイズにおいても適しているといえる。

4. むすび

本論文では、DP を用いて近似関数を決定する手法を提案し、スキャナやデジタルカメラで取りこんだ文書画像を自動処理で関数化表現画像に変換してデジタル化する手法について述べてきた。提案する手法は、近似関数フィッテイングを元にしたコスト関数を DP を用いて最小化しており、全体的に整合性のとれた関数近似が行える。関数化表現図形を用いた画像は、様々な解像度で出力できること、ファイルサイズが小さいといった特長を持っており、提案する手法を用いて自動処理でラスタ画像から生成される。実験ではファイルの圧縮率は 1.7%から 17.6%となり、高圧縮率が達成された。また、出力についても関数化図形表現画像を高精細に表示・印刷できることを確認した。これらの実験から、提案する手法の紙文書のデジタル化手法としての有効性が確かめられた.

今後の課題としては、ファイルサイズのさらなる縮小が挙げられる。関数化図形表現画像ファイルは十分小さいサイズではあるが、基本的には近似関数データをそのまま保存しているため冗長な情報があると思われる。また、一つの画像中にほぼ同じような形状の閉領域があってもそれぞれを別のデータとして保持することになるので、これもまた情報としては冗長である。他には、画質を落としてデータ量を減らせるような表現形式やカラー画像への対応も考慮していきたい。

参考文献

1. 堀内隆彦, 大瀧保広, 寅市和男, :「マルチフォントの自動函数化における接合点の多段階抽出法」, 電気学会論文誌 C, Vol. 113 No.12 , pp. 1136-1143 , 1993
2. 寅市和男, 関田巌, 森亮一 :「高品質文字フォントの自動圧縮」, 電子情報通信学会論文誌 D, Vol. J-70-D, No.6, pp. 1164-1172, 1987
3. John D. Hobby : "Space-Efficient Outlines from Image Data via Vertex Minimization and Grid Constrains" , Graphical Models and Image Processing Vol. 59 No.2, pp. 73-88, 1997
4. Lejun Shao, Hao Zhou : "Curve Fitting with Bezier Cubics", Graphical Models and Image Processing Vol. 58 No.3, pp. 223-232, 1996
5. Wenhua Wan, Jose A. Ventura : "Segmentation of Planar Curves into Straight-Line Segments and Elliptical Arcs" , Graphical Models and Image Processing Vol. 59 No.6, pp. 184-194, 1997
6. Paul L. Rosin, Geoff A.W. West : "Nonparametric Segmentation of Curves into Various Representations" , IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 17 No. 12, pp. 1140-1153, 1995
7. 大津展之, 栗田多喜夫,関田巌 :「パターン認識」, 朝倉書店, 1996
8. 大沢裕, 山川修三, 小田泰充, 新浜継夫 :「図面の認識と理解」, 昭晃堂, 1989