

文書清書システムにおける行組版機能の一実現方式

川崎 敏 治[†] 山 内 司[†]
岩 本 哲 夫^{††} 目 瀬 道 弘[†]

近年、オフィス内での高品位文書作成を目的として、自動組版を行う文書清書システムが注目されている。本論文では、文書清書システムの組版に関し、行を組むための技術である行組について報告する。行組は、文字と文字の相対関係に基づいて文字をバランスよく配置し、文字列を一行内に配置構成する技術である。そのために、日本語の文字をカテゴリライズし、その組み合わせに対して統一した処理で組版を実現する組み合わせマトリクス方式を提案し、本提案方式を用いて、実現できた機能とその特徴を示す。本方式によれば、日本語組版の特徴といえる約物が簡単に扱え、行頭、行末、巻罫・網掛け開始点、終了点を仮想的なカテゴリとして導入することによって、従来、特殊処理としていた行頭末処理及び巻罫・網掛け処理も同様の手法で扱える。また、各カテゴリの組み合わせごとに禁則条件や詰め順位等の属性を付加することで、ジャスティフィケーションも容易に実現できる。さらに、カテゴリライズの仕方、組み合わせの方法および手順、ジャスティフィケーション順位等を入れ換えることで、組版ルールの違いにも対処できる。

A New Method of Line Composition for Electronic Publishing Systems

TOSHIHARU KAWASAKI,[†] TSUKASA YAMAUCHI,[†]
TETSUO IWAMOTO^{††} and MICHIMIRO MESE[†]

Electronic publishing systems have attracted attentions for making documents of high quality in recent years. In this paper, we consider the line composition feature by electronic publishing systems. The line composition is to dispose characters well-balanced on the relations between two characters. Then we explain the combination matrix method by categorizing characters with the roles and their combinations. Using the method, the signs in the Japanese language can neatly be processed. In the same way, headline composing, tailline composing, box-rule and hatching-pattern can easily be handled by instituting some imaginary categories. And the justification can be accomplished by using this matrix. Moreover we can extend this method to other composition rules according to changing the categorization, the parameters in the combination matrix and the order of justification.

1. はじめに

組版とは、美しく読みやすいレイアウトで頁の上に文字や図や画像を配置する技術である。この技術は、活字印刷の時代からデザイナーや印刷会社の技術者が作り上げてきたものである。一般に、書体の選定、字詰、行数、見出しの大きさや行取り、均整のとれた文字間隔（位置計算）、禁則処理、ジャスティフィケーション等を組み合わせて美しい文書を作り出すことを

意味する¹⁾。これらプロフェッショナルな印刷技術者が使用する組版装置として、電算写植機がある。これは、さまざまな印刷要求に応えることができ、文字の配置を定めるパラメータは百数十にも上る。しかし、その値はすべてユーザが設定しなければならないため、組版の知識を身に付け、電算写植機を使いこなすには数年かかるといわれている。それゆえに、自由度を犠牲にはするものの、自動組版の概念が考え出された。

自動組版を行う文書清書システムとして、D.E. Knuth による TeX²⁾、それを改良した LaTeX³⁾等がある。これらは、欧文を基本にしたシステムであるが、日本語も扱えるように改良された日本語 TeX (JTeX)⁴⁾や日本語 LaTeX (JLaTeX)⁵⁾も開発されている。

JTeX および JLaTeX における組版は、TeX が本

[†] (株)日立製作所 マイクロエレクトロニクス機器開発研究所
Microelectronics Products Development Laboratory,
Hitachi, Ltd.

^{††} (株)日立製作所 情報映像メディア事業部映像システム本部
Image System Operation, Information & Image
Systems Division, Hitachi, Ltd.

来持っている box および glue の関係を用い、和文の一字を box とし、文字と文字の間に小さな glue を挿入することで実現されている。すなわち、和文の一字を欧文の一語に対応させている。和文においては、基本的には何処で行を分割しても構わないため、このようなことが可能となる。ただし、和文には、後述するような禁則処理がある。そこで、禁則に該当する箇所には glue を無効にするコマンドを挿入することで対処している^{4),6)}。しかし、もともと欧文用のシステムを TeX のマクロ機能を用いて実現しているため、日本語組版の細かな字間設定やジャスティフィケーションを実現しようとするとその処理は複雑になる。

一方、最初から和欧混植（後述）を行うことを目的としたシステムとして JOSHO^{7),8)}がある。これは、TeX のようにコマンドを用いて（和文コマンドも使える）、組版指示を行うものである。しかし、このシステムは、組版処理方法が固定であり、細かな日本語組版規則にフレキシブルに対応できない。

そこで、新たに、日本語に合わせた組版を実現するためのデータ構造と処理方式を考案することにした。我々は、組版を下記の二つの部分に分けて考えた。

- (1) 文字を一行内に配置構成する部分（行組と呼ぶ）
- (2) 行（および、図、画像等）を頁内に配置構成する部分（頁組と呼ぶ）

本論文では、前者の行組を実現するために、日本語の文字をカテゴリズし、その組み合わせに対して統一した処理で組版を実現する方式を提案し、本提案方式を用いて、実現できた機能とその特徴を示す。

2. 組み合わせマトリクス行組方式

行組は、与えられた行長に対して、美しく読みやすくするために文字列をどのように配置すればよいかを決定する処理のことである。このため、行組は、

- (1) 印刷の分野で決められた文字と文字の相対関係に従って、文字を仮配置して行を作る位置計算。
- (2) 文字列の長さが行長を超えた時点で、禁則条件を満足するように行の区切りを決める禁則処理。
- (3) 文字列の長さが行長に対して過不足を生じた場合、行長と一致するように文字間隔を詰めたり延ばしたりすることで文字配置の微調整を行うジャスティフィケーション。

から構成される^{9),10)}。

従来の位置計算では、各文字ごとに文字幅を属性と

して持たせ処理していた。例えば、TeX においては、文字の位置属性（メトリックスと呼ばれる）と文字形状データ（ビットマップデータ）とを分離して処理を行うため、このような方式が採られている。欧文においては、各文字ごとにメトリックスは異なるため、従来方法で行うほうがよい。しかし、和文においては約物（後述）以外はすべて固定の文字幅を持つため、メトリクスデータを持つことは大部分が無駄となる。

また、前述したように TeX における禁則処理は、glue の挿入および禁則箇所での glue の無効化によって実現しているが、細かな字間設定を行おうとすると glue の値が多岐にわたり、計算が複雑になる。

また、日本語組版のジャスティフィケーションにおいて、字間を拡げる手法（延ばし処理）のみを採ると間延びした感じになる。そのために、電算写植機では、文字の仮想ボディー（TeX の box に相当する）をある程度まで重ね合わせて配置する手法（詰め処理）が採用されている。

したがって、日本語組版における細かな字間設定および日本語特有の禁則処理および詰め処理を含むジャスティフィケーションが課題となる。

まず、これらの課題を解決するために、行組で使用するパラメータを定義する。

位置計算では、

- α : 前文字係数,
- β : 後文字係数,
- γ_1 : 字間率,
- γ_2 : 和欧字間率,
- γ_3 : 巻野・網掛け字間率,

とするとき、前文字に対する後文字の相対位置 d は、これらのパラメータおよび前後各文字の文字サイズ s_1, s_2 および字間 j , 和欧字間 e , 巻野・網掛け字間 b を用いて

$$d = \alpha s_1 + \beta s_2 + \gamma_1 j + \gamma_2 e + \gamma_3 b \quad (1)$$

で与えることができる。なお、字間 j , 和欧字間 e , 巻野・網掛け字間 b は、全体の体裁を決める書式で定められる定数である。

禁則箇所は、前後の文字の組み合わせに対して定まる属性であり、

- “行頭禁則”,
- “行末禁則”,
- “分離禁止”,
- “プロポーション”,

がある。それぞれの属性が設定されている箇所では、

その属性に従って処理を行う。

ジャスティフィケーションのためには、上記の属性“分離禁止”のほかに、“詰め処理を行うための順位”が必要である。

次に、行組を構成する三つの処理を統一的に扱えるように考察した組み合わせマトリクス行組方式について述べる。

文字をその文字の使われ方、意味、形状等を考慮して、上記行組パラメータに従って文字を配置することを想定し、次のように分類することにした。

- 「句読点」 、。、
- 「括弧(受)」 ”)】}]》】】
- 「中黒・コロン・セミコロン」 : ;
- 「括弧(起)」 “([{ [《 [【

「疑問符・感嘆符」

?!

「ダッシュ・リーダー」

— … …

「スペース」

(JIS 句点 01 区 01 点)

「非漢字」

(上記以外の JIS 01 区~08 区)

「漢字」

(JIS 16 区~84 区)

「英数字」

(ASCII コード 21 H~7 EH)

「欧文語間」

(ASCII コード 20 H)

これらを文字カテゴリと呼び、以下、「」付きの名前で参照する。また、一般に、「非漢字」および「漢字」以外を約物と呼び、特に、「句読点」、「括弧(受)」、「中黒・コロン・セミコロン」および「括弧(起)」を半角約物と呼ぶ。

文字カテゴリによる分類は、最初から前記三つすべ

後 前	括弧 読点 (受)	中 黒 コ ロ ン セ ミ コ ロ ン	括 弧 (起)	感 嘆 符 疑 問 符	リ ダ ッ シ ュ リ ー ダ ー	ス ペ ー ス	非 漢 字
句 読 点	$\alpha=0$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
括 弧 (受)	$\alpha=0$ $\beta=0.5$ 詰め2位	$\alpha=0$ $\beta=0.25$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
中 黒 コ ロ ン セ ミ コ ロ ン	$\alpha=0.25$ $\beta=0.5$ 詰め2位	$\alpha=0.25$ $\beta=0.25$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.25$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.25$ $\beta=0.5$ 詰め2位	$\alpha=0.25$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.25$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
括 弧 (起)	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 詰め2位	$\alpha=0.5$ $\beta=0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
疑 問 符 感 嘆 符	$\alpha=0.5$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
ダ ッ シ ュ リ ー ダ ー	$\alpha=0.5$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ 分離禁止	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位
ス ペ ー ス	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.25$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位
非 漢 字	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位

図 1-1 組み合わせマトリクス (その 1)
Fig. 1-1 A combination matrix (part 1).

での処理に共通に利用できるように細分化している。また、位置計算や禁則処理の簡便さのため、さらに、また、位置計算や禁則処理の簡便さのため、さらに、本来文字ではない「行頭」、「行末」、「巻野・網掛け開始点」および「巻野・網掛け終了点」を仮想的な文字カテゴリとして設定する。

次に、以上の文字カテゴリの組み合わせを考える。

図1-1~1-4は、前後の二文字の間の文字間隔の関係をまとめたものである。これを組み合わせマトリクスと呼ぶ。組み合わせマトリクスの各要素は、位置計算に必要な前述の各パラメータおよび禁則条件を示すフラグ類および詰め処理を行うための詰め順位（“詰め1位” “詰め2位”，“詰め3位”）で構成する。なお、図1-1~1-4において、 $\gamma_1, \gamma_2, \gamma_3$ のうち0のものは省略した。

後 前	漢 字	英 数 字	欧 文 語 間	開網巻 始掛野 点け・	終網巻 了掛野 点け・	行 末
句読点	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.25$ 分離禁止	$\alpha=0$ $\beta=0$ 分離禁止
括弧(受)	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.25$ 分離禁止	$\alpha=0$ $\beta=0$ 分離禁止
中黒 コロン セミコロン	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.25$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.25$ $\beta=0.25$ 分離禁止	$\alpha=0.25$ $\beta=0$ 分離禁止
括弧(起)	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 行末禁則
疑問符 感嘆符	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 分離禁止
ダッシュ リーダー	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め2位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 分離禁止
スペース	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 詰め1位
非漢字	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 分離禁止

図1-2 組み合わせマトリクス (その2)
Fig. 1-2 A combination matrix (part 2).

このような組み合わせマトリクスを考えることで、処理とデータを分離でき、したがって、プログラム化が容易になり、しかも、組版ルールの変更にも対処できる。組版ルールは、デザイナーによっても、また、文字種や文字デザイン等によっても異なるため、組版ルールの違いに対処できることは、メリットが大きい。また、仮想的な文字カテゴリを考えることで、従来難しかった行頭行末処理や巻野・網掛け処理も容易に取り扱うことができる。

次章では、この組み合わせマトリクスを用いて、行組を構成する三つの処理を如何に実現するかについていくつかの例を用いて考察する。

3. 提案方式の適用例

3.1 位置計算

3.1.1 位置計算方法

位置計算は、行組の根幹を成す部分である。この部分は、一つ前の文字に対して現在着目している文字をどれだけ離れた位置に置けば体裁よく配置されるかを計算する処理である。これには、約物処理、行頭行末処理、巻野・網掛け処理、欧文のプロポーショナル処理、和欧混植がある。

前述したように、従来の位置計算では、各文字ごとに文字幅を属性として持たせ処理していた。例えば、TeX においては、文字ごとにメトリクスを持っている。また、各文字の組み合わせごとに glue の挿入を制御する。これに対して、本提案方式では、文字をカテゴリ化し、文字カテゴリの組み合わせに対して、メトリクスデータを用意し、また、glue 調整を行うため、データを削減できる。

以降では、位置計算の各処理が、本提案方式を用いて簡単に実現できることについて述べる。

3.1.2 二重約物処理

一例として、コンマの次に小括弧(受)がくる場合を考える。普通

後 前	括弧 読点 (受)	セ ン テ ン ス 中 心 黒	括弧 (起)	感 嘆 問 符	リ ダ ッ シ ユ	ス ペ ー ス	非 漢 字
漢 字	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位
英 数 字	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=1.0$ $\beta=0.5$ 詰め2位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位
欧文語間	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め1位	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位
巻網開 野掛 ・ 点 始	$\alpha=0.25$ $\beta=0.5$ 分離禁止	$\alpha=0.25$ $\beta=0.25$ 分離禁止	$\alpha=0.25$ $\beta=0$ 分離禁止	$\alpha=0.25$ $\beta=0.5$ 分離禁止	$\alpha=0.25$ $\beta=0.5$ 分離禁止	$\alpha=0.25$ $\beta=0.5$ 分離禁止	$\alpha=0.25$ $\beta=0.5$ 分離禁止
巻網終 野掛 ・ 点 了	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位
行 頭	$\alpha=0$ $\beta=0.5$ 行頭禁則	$\alpha=0$ $\beta=0.25$ 分離禁止	$\alpha=0$ $\beta=0$ 分離禁止	$\alpha=0$ $\beta=0.5$ 行頭禁則	$\alpha=0$ $\beta=0.5$ 分離禁止	$\alpha=0$ $\beta=0.5$ 行頭禁則	$\alpha=0$ $\beta=0.5$ 分離禁止

図 1-3 組み合わせマトリクス (その3)

Fig. 1-3 A combination matrix (part 3).

後 前	漢 字	英 数 字	欧 文 語 間	開網巻 始掛野 点け・	終網巻 了掛野 点け・	行 末
漢 字	$\alpha=0.5$ $\beta=0.5$ $\gamma_1=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め3位	$\alpha=0.5$ $\beta=1.0$ $\gamma_1=0.5$ 詰め3位	$\alpha=0.5$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め3位	$\alpha=0.5$ $\beta=0.25$ 分離禁止	$\alpha=0.5$ $\beta=0$ 分離禁止
英 数 字	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め3位	$\alpha=1.0$ $\beta=1.0$ 詰め1位	$\alpha=1.0$ $\beta=1.0$ $\gamma_1=0.5$ $\gamma_2=0$ 詰め1位	$\alpha=1.0$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=1.0$ $\beta=0.25$ 分離禁止	$\alpha=1.0$ $\beta=0$ 分離禁止
欧文語間	$\alpha=1.0$ $\beta=0.5$ $\gamma_1=0.5$ 詰め1位	$\alpha=1.0$ $\beta=1.0$ $\gamma_1=0.5$ $\gamma_2=0$ 詰め1位	$\alpha=1.0$ $\beta=1.0$ $\gamma_1=1.0$ 詰め1位	$\alpha=1.0$ $\beta=0$ $\gamma_1=0.5$ $\gamma_2=0$ 詰め1位	$\alpha=1.0$ $\beta=0.25$ 分離禁止	$\alpha=1.0$ $\beta=0$ 詰め1位
巻網開 野掛 ・ 点 始	$\alpha=0.25$ $\beta=0.5$ 分離禁止	$\alpha=0.25$ $\beta=1.0$ 分離禁止	$\alpha=0.25$ $\beta=1.0$ 分離禁止		$\alpha=0.25$ $\beta=0.25$ 分離禁止	
巻網終 野掛 ・ 点 了	$\alpha=0$ $\beta=0.5$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め3位	$\alpha=0$ $\beta=1.0$ $\gamma_1=1.0$ $\gamma_2=1.0$ 詰め2位	$\alpha=0$ $\beta=1.0$ $\gamma_1=0.5$ $\gamma_2=0$ 詰め1位	$\alpha=0$ $\beta=0$ $\gamma_1=1.0$ $\gamma_2=2.0$ 詰め2位		$\alpha=0$ $\beta=0$ 分離禁止
行 頭	$\alpha=0$ $\beta=0.5$ 分離禁止	$\alpha=0$ $\beta=1.0$ 分離禁止	$\alpha=0$ $\beta=1.0$ 行頭禁則	$\alpha=0$ $\beta=0$ 分離禁止		

図 1-4 組み合わせマトリクス (その4)

Fig. 1-4 A combination matrix (part 4).

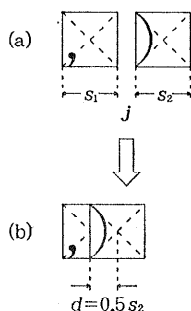


図2 二重約物処理
Fig. 2 Double-signs composing.

に配置すれば、図2(a)のように半角分の隙間や字間が空いてしまい体裁が悪い^{11),12)}。そこで、コンマおよび小括弧(受)が、それぞれいずれの文字カテゴリに属するかを上記の分類に従って決定し、これより組み合わせマトリックスを参照する。組み合わせマトリックス内には $\alpha=0$, $\beta=0.5$ が入っている。また、字間、和歌字間および巻野・網掛け字間が挿入されないように $\gamma_1=\gamma_2=\gamma_3=0$ (図1では省略) となっている。したがって、この二文字間の相対位置 d は、

$$d=0.5s_2 \quad (2)$$

となる。これは、「句読点」であるコンマを半角として扱ったことに相当し、結局図2(b)のように配置され、余分な隙間を詰めることができ、体裁のよい配置結果が得られる。このように、約物が連なる場合の処理を、一般に、二重約物処理と呼ぶ。

二重約物処理の一つに、ダッシュ・リーダ処理と呼ぶものがある。ダッシュまたはリーダが連続しているとき、字間が設定されている場合、普通は図3(a)のように離れて配置されてしまう。しかし、これでは、二全角ダッシュのような長ダッシュや長リーダができない。これを抑制するために、ダッシュやリーダの文字カテゴリである「ダッシュ・リーダ」同士の間には字間を挿入しないように $\gamma_1=\gamma_2=\gamma_3=0$ (図1では省略) を設定する。このようにすることで、二全角ダッシュなどの長ダッシュや長リーダが組めるようになる(図3)。

以上のようにして、文字を文字カテゴリに分類し、組合せマトリックスを参照することによって、二重約物処理を実行することができる。また、組版ルールの違いがある場合でも、このテーブルを差し換えるだけで容易に対処できる。例えば、句点の後と読点の後では、欧文の処理を模倣し(欧文においては、ピリオドとコンマの後とは、欧文語間の長さを変えるのが普

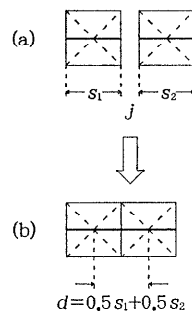


図3 ダッシュ・リーダ処理
Fig. 3 Dash and leader composing.

通である。一般にコンマの後よりピリオドの後のほうが長い) 微妙に処理を変えたい場合など、新たな組み合わせマトリックスを作成することで対処することができる。

3.1.3 行頭行末処理

半角約物が、行頭(headline)または行末(tailline)にきた場合、版面を合わせるためには半角約物の前後の余白を取って組んだほうが体裁がよい^{9),10)}。

例えば、行頭の後に小括弧(起)がくる場合を考えると、普通は図4(a)のように配置されるが、行頭では、小括弧(起)を半角として配置し図4(b)のように組む。

これは、次のように処理すればよい。すなわち、前文字の文字カテゴリである「行頭」の文字サイズを $s_1=0$ と考え、 $\alpha=0$, $\beta=0.5$, さらに $\gamma_1=\gamma_2=\gamma_3=0$ (図1では省略) より

$$d=0.5s_2 \quad (3)$$

とする。

行末に半角約物がきた場合も同様である。例えば、行末にコンマがきた場合を考える。この場合も、普通

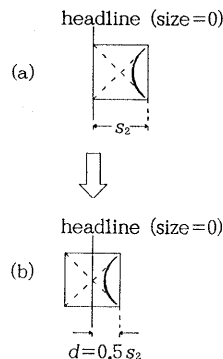


図4 行頭処理
Fig. 4 Headline composing.

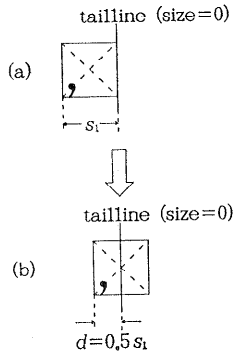


図 5 行末処理
Fig. 5 Tailline composing.

は図 5 (a)のように配置されるが、コンマを半角として配置し図 5 (b)のように組む。

これは、次のように処理すればよい。すなわち、後文字の文字カテゴリである「行末」の文字サイズを $s_2=0$ と考え、 $\alpha=0.5$ 、 $\beta=0$ 、さらに $\gamma_1=\gamma_2=\gamma_3=0$ (図 1 では省略) より

$$d=0.5s_1 \quad (4)$$

とすればよい。

以上のように、行頭行末処理は、行頭および行末を仮想的な文字として扱うことによって、組み合わせマトリクスの一部として組み込むことができるようになる。

3.1.4 巻野・網掛け処理

巻野・網掛けは、文字列に対する属性である。通常、文字列の回りに $1/4\text{ em}$ (em は全角の大きさを 1 em としたときの長さを表す単位で、エムと読む) ほどの余白を取る。さらに、巻野・網掛けの前後は、文字と巻野または網掛けパターンとが干渉しないように巻野・網掛け字間 (通常は $1/4\text{ em}$) の隙間を空ける。

巻野・網掛けは、一種の約物として処理する。したがって、その組み方は約物処理の一部であり、その処理を組み合わせマトリクスの中に組み込むことができる。

実際は、次のように行う。すなわち、「巻野・網掛け開始点」および「巻野・網掛け終了点」を仮想的な文字カテゴリとして導入し、それらの文字サイズを標準文字サイズ s とすることによって、図 6 のように

$$d_1=0.5s_1+j+b, \quad (5)$$

$$d_2=0.25s+0.5s_2, \quad (6)$$

$$d_3=0.5s_2+0.5s_3+j, \quad (7)$$

$$d_4=0.5s_3+0.25s, \quad (8)$$

$$d_5=0.5s_4+j+b \quad (9)$$

が得られる。また、「巻野・網掛け終了点」と「巻野・網掛け開始点」が続いた場合、 $\alpha=\beta=0$ 、 $\gamma_1=1.0$ 、 $\gamma_2=0$ (図 1 では省略)、 $\gamma_3=2.0$ であり、したがって

$$d=j+2b \quad (10)$$

が得られる。これは、巻野・網掛け同士の間、通常の字間に加え巻野・網掛け字間を二個挿入することに相当する。

なお、巻野または網掛けの内部での組み方は通常の組み合わせマトリクス通りであるが、巻野や網掛けの始めと終わりに半角約物がきた場合は、間延びした

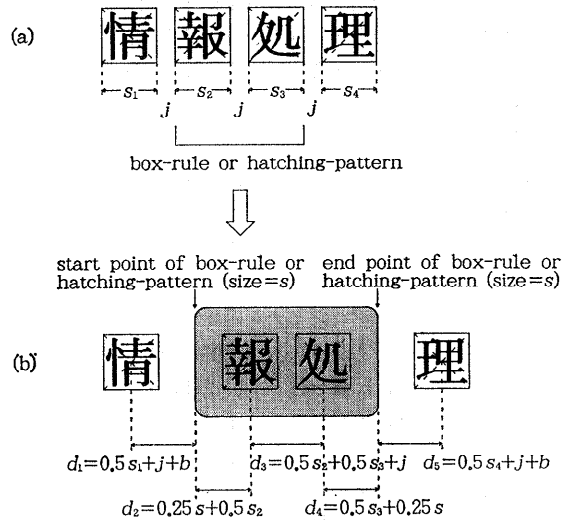


図 6 巻野・網掛け
Fig. 6 Box-rule and hatching-pattern.

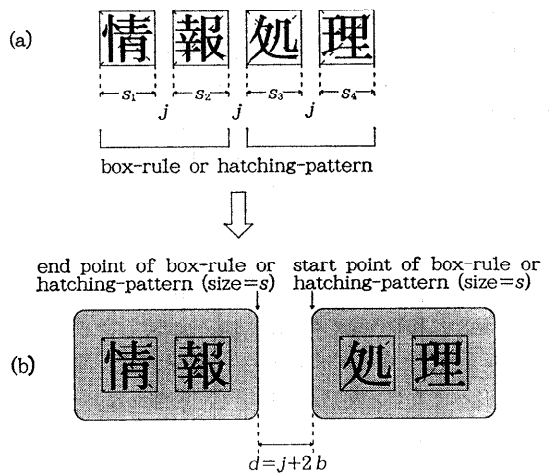


図 7 巻野・網掛けが連なる場合
Fig. 7 Case of two consecutive box-rules or hatching-patterns.

感じを出さないようにそれらを半角として扱う。これは、「巻野・網掛け開始点」および「巻野・網掛け終了点」をそれぞれ「行頭」および「行末」と考え、行頭行末処理と同様に考えればよい（ただし、「行頭」および「行末」は文字幅を持たないのに対し、「巻野・網掛け開始点」および「巻野・網掛け終了点」は文字幅を持つ）。

3.1.5 和欧混植¹⁰⁾

和文においては、半角約物を例外としてすべてが全角の文字として処理すればよい。したがって、和文組においては、約物処理（二重約物処理、行頭行末処理、巻野・網掛け処理等）さえ正しく行えばよい。これはすでに見てきたように、文字のカテゴリライズおよび組み合わせマトリクスを用いることによって、容易に処理ができる。

しかし、英数字においては、各文字が固有の文字幅を持っており、さらに、英数字間でカーニングを起こすために、これを配慮しながらそれぞれの文字位置を計算して行かねばならない。これをプロポーショナル処理と呼ぶが、プロポーショナル処理を行うには、各文字にいくつかのメトリックス（文字幅やカーニング位置等）を持たせ、配置を行ってあげればよい。この方法は、各種の方法が提案されているが、特にカーニングを行うには、各文字の組み合わせをマトリクス状にまとめたカーニングテーブルを用いるなどの方法がよい。英数字の文字数が少ない場合には、これは最もよい方法であり、数種にカテゴリライズしておけば、メモリの節約もできる。すなわち、「英数字」は、「英数字」内でさらにカテゴリライズし、「英数字」用の組み合わせマトリクスを作成し、位置計算を行えばよい。

和文と欧文が同一文書内に現れるような組み方を和欧混植と呼ぶ。欧文の部分は、上記のようにプロポーショナル処理を行い、単語と単語の間は文語間（半角スペースを用いる）によって分離され、組み合わせマトリクスに従い配置する。一つの単語は、分離禁止であり、一種の約物として処理する。すなわち、和文と欧文の間も、組み合わせマトリクスに従い配置する。また、適所に和欧字間を自動的に挿入する。これは、例えば、「漢字」の次に「英数字」が来た場合、他の箇所比べて文字と文字が接近して配置されるような感じがするためである。和欧字間は、人の好みの問題もあるため全く挿入しないほうがよいとする人もいるが、通常は、1/4 cm が取られる^{11),12)}。

例えば、「漢字」の後に「英数字」がくる場合を考

えると、 $\alpha=\beta=1.0$ であり、さらに $\gamma_1=\gamma_2=1.0$ 、 $\gamma_3=0$ （図1では省略）であるから、

$$d=s_1+s_2+j+e \quad (11)$$

が得られる。（「英数字」の文字幅 s_2 は、「英数字」のメトリックスデータより定まる。）

3.2 禁則処理

禁則処理には、行頭禁則処理、行末禁則処理、分離禁止の三つがある^{10),11),12)}。これらの禁則番件は、組み合わせマトリクスの各要素に属性として組み込む。

行頭禁則処理は、行頭に来ては体裁上好ましくない文字を行頭に配置しないようにする処理である。そのためには、行頭禁則文字を前行に押し込むか、または、行頭禁則文字の前の文字とともに次行に追い出すか、どちらかの処理を行う必要がある。行頭禁則文字には、文字カテゴリの「句読点」、「括弧(受)」、「疑問符・感嘆符」、「スペース」および「欧文語間」がある。

行末禁則処理は、行末にきては体裁上好ましくない文字を行末に配置しないようにする処理である。そのためには、行末禁則文字を次行に追い出すか、または、行末禁則文字の次の文字とともに前行に押し込むか、どちらかの処理を行う必要がある。行末禁則文字には、文字カテゴリの「括弧(起)」がある。

分離禁止は、複数の行に跨って配置してはならない文字列を同一行内に配置する処理である。そのためには、分離禁止文字列をすべて前行に押し込むか、または、次行に追い出すか、どちらかの処理を行う必要がある。分離禁止文字列は、すでに見てきた「ダッシュ・リーダー」文字列、「英数字」文字列や巻野等があり、さらに、ユーザによって指定される文字列もある。

以上の禁則条件を満たす行末に最も近い文字と文字の間が、その行の分断点となる。

すなわち、行長（headline から tailline までの距離）を L とするとき、分断点の決定は次のように行う。

[ステップ1]

$$n \leftarrow 1, n_1 \leftarrow 0, E_1 \leftarrow L$$

[ステップ2]

行頭の第1番目の文字との距離 d_1 および第1番目の文字と行末との距離 d を計算する。

[ステップ3]

$$n_2 \leftarrow 1, E_2 \leftarrow |d_1 + d - L|.$$

[ステップ4]

$$n \leftarrow n + 1.$$

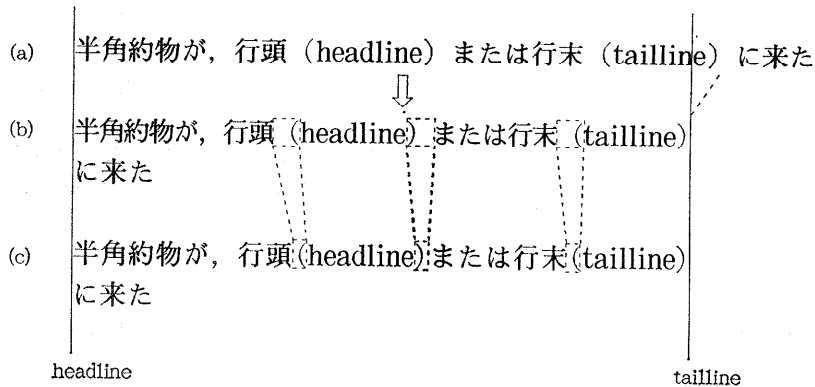


図 9 詰め処理

Fig. 9 The tight setting.

- (3) (1)(2)で対処できない分を、分離禁止でなく、かつ、文字カテゴリ「漢字」との組み合わせになっている箇所をそれぞれ $1/10 \text{ em}$ 以内で詰める。
- (4) 以上で対処できない場合は、詰め処理は不可能である。

(1)からわかるように、詰める場合は、まず最初に行内にあらかじめ存在する空白を詰めることを考慮するわけである。このため、(1)の代わりに次の条件を設けることもある。

(1)行内にある「括弧(受)」の後ろの空白、「括弧(起)」の前の空白、「中黒・コロン・セミコロン」の前後の空白を詰める。ただし、約物処理によってすでに詰めてある箇所には適用できない。

また、(2)と(3)による優先順を設けてあるのは、一般に仮想ボディ（デザインメッシュの大きさ）に対するレター（実際に文字の形状を構成している部分）は、「非漢字」のほうが「漢字」より小さいため、「非漢字」と「非漢字」の組み合わせを他の組み合わせより優先して詰めたほうがバランスがよいからである。

活版印刷の時代には、活字の大きさが固定であるため、ジャスティフィケーションといえば、詰物と呼ばれるスペーサーを挿入する延ばし処理、および、(1)の方法による詰め処理に頼りしかなかった。詰め処理の(2)(3)のような方法は、写植時代に入って初めて可能となり、その歴史はまだ浅い。各種の電算写植システムで統一的な詰めアルゴリズムがなく、(1)と(1)のような手法の流儀にゆらぎがあるのもこの理由によるものかもしれない。

両者の流儀による詰め処理を、それぞれ図9(b)お

よび図9(c)に示す。

以上の詰め処理は、組み合わせマトリクスの各要素に組み込まれている詰め順位によって判定することができる。すなわち、行長 L に対し文字列長 $l (> L)$ であるとき、現在着目している行内に“詰め1位”“詰め2位”“詰め3位”の属性を持つ組み合わせがそれぞれ幾つかあるかカウントしておき、それぞれの文字間の詰め量を計算すればよい。図9(a)の例では、“詰め1位”の属性のものが0箇所、“詰め2位”の属性のものが8箇所、“詰め3位”の属性のものが10箇所ある。したがって、文字の標準サイズを s とするとき

(1) $l-L \leq (8/10)s$ ならば、“詰め2位”の属性の箇所を均等に $(l-L)/8$ 詰めればよい。

(2) $(8/10)s < l-L \leq (18/10)s$ ならば、“詰め2位”の属性の箇所を $s/10$ ずつ詰め、さらに、“詰め3位”の属性の箇所を均等に $(l-L-(8/10)s/10$ 詰めればよい。

なお、 $(18/10)s < l-L$ のとき、詰め処理は不可能である。

以上のように、組み合わせマトリクスを使うことによって、当初の目的であった行組を構成する三つの処理（位置計算、禁則処理およびジャスティフィケーション）を統一した処理で実現することが達成できたものとする。

4. おわりに

以上、文字をカテゴリ化すること、その組み合わせをマトリクス状にして用いることの有効性について述べた。具体的には、文字をカテゴリ化することによって、以下のような特徴が実現できた。

(1) 日本語組版の特徴である約物処理、複雑なスペー

シング処理を必要とする和欧混植。行頭、行末を仮想的な文字として扱うことによる行頭末の半角約物の処理等が、組み合わせマトリクスを用いることによって、パーソナルコンピュータ、ワークステーション等で容易に実現できる。

- (2) 組み合わせマトリクスに分離禁止属性および詰め順位を設定することによって、各文字と文字の文字間がどれほど微調整可能かを決定することができ、詰め処理を含むジャスティフィケーションを自動的に実行できる。
- (3) 組み合わせマトリクス、ジャスティフィケーション順位等を変えることで、容易に組版ルールの違いに対処できる。

謝辞 本研究の機会を与えて頂いた日立マクセル猪瀬文之博士、ならびに本研究を進めるに当たり、熱心に御討論頂いた東京印書館 阿由葉一雄氏、日立印刷 山崎彬雄氏、日立ハイソフト 近藤 晋氏に感謝いたします。

参 考 文 献

- 1) 日経 EP: デスクトップパブリッシングとは何か?, pp. 56-63 (1987).
- 2) Knuth, D. E.: *Computers and Typesetting, Series A, The TeXbook*, Addison-Wesley Publishing Company (1986).
- 3) Lamport, L.: *LaTeX: A Document Preparation System*, Addison-Wesley Publishing Company (1986).
- 4) 斎藤康己: 日本語 TeX, 情報処理学会日本語処理研究会予稿集, 10(3) (1987).
- 5) 大野義夫: TeX 入門, 共立出版 (1989).
- 6) 来住伸子, 山内長承: DP Matching による日本語禁則処理, 第 33 回情報処理学会全国大会予稿集 7X-2 (1986).
- 7) Takahashi, N., Atoda, O. and Manabe, T.: 浄書: Japanese Output Server with HOspitality Proc. of Intl. Conf. on Text Processing with Large Character Set, Tokyo (1983).
- 8) 関口 治, 中川正樹, 高橋延匡: 和欧混合組版機能を持つインテリジェントプリンタ第 1 版の実現, 情報処理学会日本語処理研究会予稿集, 9(1) (1986).
- 9) 岩本哲夫, 山内 司, 目瀬道弘, 川崎敏治, 松田泰昌, 近藤 晋: パソコン DTP における高品位文書処理, 情報処理学会「マイクロコンピュータとワークステーションによる卓上出版のネットワーク」シンポジウム予稿集 (1988).
- 10) 川崎敏治, 岩本哲夫, 山内 司, 目瀬道弘, 松田泰昌, 近藤 晋: 電子編集システムの組版処理における行組の考察, 電子情報通信学会春季全国大会予稿集 D-72 (1989).
- 11) 長谷川文明: 横組組版の原則, 印刷研究社 (1983).
- 12) 写植ルール委員会: 組み NOW, 写研 (1975). (平成 5 年 1 月 22 日受付) (平成 5 年 6 月 17 日採録)



川崎 敏治 (正会員)

1961 年生. 1984 年東京理科大学理工学部数学科卒業. 1986 年同大学院理工学研究科数学専攻修了. 同年(株)日立製作所入社. 現在, マイクロエレクトロニクス機器開発研究所にて, スペースモデル, 日本語電子出版, 3Dグラフィックスの研究に従事. 電子情報通信学会, 日本数学会各会員.



山内 司 (正会員)

1949 年生. 1972 年神戸大学工学部計測工学科卒業. 同年(株)日立製作所入社. 同社日立研究所に入所. 1983 年から同社マイクロエレクトロニクス機器開発研究所にて日本語電子編集システムの研究開発を経て, 現在日本語ワードプロセッサの研究開発に従事. 計測自動制御学会会員.



岩本 哲夫 (正会員)

1946 年生. 1969 年大阪大学基礎工学部制御工学科卒業. 1971 年同大学院修士課程修了. 同年(株)日立製作所入社. 現在, 同社情報映像メディア事業部映像システム本部に勤務. DTP, プレゼンテーション・システムの開発に従事. 電子情報通信学会会員.



目瀬 道弘 (正会員)

1944 年生. 1968 年大阪大学基礎工学部制御工学科卒業. 1970 年同大学院基礎工学研究科物理系修士課程修了. 同年(株)日立製作所入社. 中央研究所にて物体認識, 画像処理, 音声認識などの研究に従事. 現在マイクロエレクトロニクス機器開発研究所第 2 部主任研究員. 同所で図形処理, 日本語ワードプロセッサ, 日本語電子編集システムなどの研究開発に従事. 電子情報通信学会会員.