ビデオテックスにおけるプロトコル変換

浜 野 輝 夫[†], 渡 部 保日児[†]
正会員 小 林 幸 雄[†]

あらまし ランク 4 以上の端末を対象とした NAPLPS から CAPTAINPLPS へのプロトコル変換法を提案している。特に、文字を変換する際に、文字フォントを自動的にジオメトリック描画コマンドセットに変換する手法、および NAPLPS 画面をひずみなく縮小する手法(NAPLPS 画面の方が CAPTAINPLPS のそれよりも若干大きい) について述べている。本手法によって、従来、NAPLPS → CAPTAINPLPS 変換で不可能であった形情報、色情報、文字情報の完全変換、画面の縮小、およびオンラインでの実時間変換が可能となる。

1. ま え が き

現在、ビデオテックス通信に適用されている表現レベルプロトコルには、国内で使用されている CAPTAINPLPS、北米を中心に用いられている NAPLPS、そしてヨーロッパを中心に用いられている CEPT の 3 方式があげられる。これら 3 方式は、いずれもが昨年 10 月に CCITT の勧告を受け、国際標準方式となった。

このような状況において、いかにして異方式のシステム間で相互接続を行うかが、国際間ばかりでなく国内的にも大きな課題となっている。

従来、NAPLPS から CAPTAINPLPS へのプロトコル変換法としては、NAPLPS のジオメトリック表示を CAPTAINPLPS のフォトグラフィック表示に変換する手法が知られている¹⁾。しかし、この場合、①文字情報の変換を完全に行うことができない、②CAPTAINPLPS の表示領域が NAPLPS のそれよりも小さいため、画面の端の部分が欠落する、③NAPLPS は着色方式がドット着色であるが、CAPTAINPLPS は 4×4 のブロック着色のため、ブロック内に 3 色以上の色が存在する場合に、その情報が失われてしまう、④NAPLPS の動的な画表示を再現できない、⑤オンラインでの実時間変換が行えない、等の問題点がある。

い、等の問題点がある。

本論では、以上の問題点を解決するために、ランク 4 以上の端末を対象として次のような NAPLPS → CAPTAINPLPS 変換法を提案している。(キャプテン端末は、その機能に応じて 5 つのランクに分類される。ランク 4 とは、文字、記号、モザイクをコード表示し、図形表示はパターン表示とジオメトリック表示の両方の機能を持ち、かつ標準密度画面の端末)

①については、NAPLPS の文字表示方式が CAPTAINPLPS のそれと根本的に異なるため、NAPLPS の文字を変換する際に、文字フォントから自動的にジオメトリック描画コマンドを発生させて描画することにより、CAPTAINPLPS に変換する手法を提案している。

②については、NAPLPS 画面の各頂点をいったん CAPTAINPLPS 表示領域上に写像してから、ジオメトリック描画コマンドのオペランドを算出する手法を提案している。

③と④については、ランク 4 以上の端末はドット単位着色のジオメトリック表示機能を有し、かつ CAPTAINPLPS のジオメトリック図形コマンドが NAPLPS のジオメトリック図形コマンドに準拠している点に着目し、ジオメトリック → ジオメトリック変換を行って解決している。

⑤については、本手法により、パーソナルコンピュータを用いて実験を行い、オンラインでの実時間変換が充分可能であることを確認した。

[†] 日本電信電話株式会社 複合通信研究所

"The Protocol Conversion from NAPLPS to CAPTAINPLPS" by Teruo Hamano, Yasuhiko Watanabe and Yukio Kobayashi (Yokosuka Electrical Communication Laboratory, NTT, Yokosuka)

2. 変換処理系

NAPLPS から CAPTAINPLPS への変換処理系の構成にあたっては、まずオフラインでの変換処理系を作成した²⁾³⁾。オフラインの処理系は、3つのパスから構成される。まず NAPLPS の画面データ（オブジェクトデータ）を逆アセンブルして、NAPLPS のソースファイルを生成する。次に、このソースファイルを CAPTAINPLPS ソースファイルに変換し、最後にこれをアセンブルして、CAPTAINPLPS のオブジェクトデータを生成する。このオフライン変換は、NAPLPS → CAPTAINPLPS 変換を行う際に生じる様々な問題点の抽出とこれに対する解決策の試行をソースレベルで行うことができる。こうして、この解決策の正当性をソースレベルで確認した後、これをオンライン変換系に適用することで、効率良く、かつ確実にオンライン変換の処理系を構築することができる。

オンライン変換処理系では、NAPLPS のオブジェクトデータを、直接 CAPTAINPLPS オブジェクトに変換する。オンライン変換を行う場合、いかに端末に連続的に表示を行わせるかが問題となる。すなわち、端末の利用者にとって、画面表示が断続的になるような状況は、オンライン変換である以上できる限り避けなくてはならない。このため、プロトコル変換の単位を NAPLPS の 1 オペコードとそれに付随する一連のオペランド（1 コマンド）とした。この 1 単位を CAPTAINPLPS に変換して、CAPTAIN 端末に送出する。この 1 コマンドには、複数の CAPTAINPLPS コマンドが含まれるかもしれないが、その各々は“完全な”1 コマンドを形成するはずであり、CAPTAIN 端末に何らかの表示を行わせる、もしくは何らかの内部的な動作を行わせることが保障される最小の変換単位とみなすことができる。

3. CAPTAINPLPS と NAPLPS の比較と変換に伴う問題点

CAPTAINPLPS と NAPLPS の表示機能の比較を表 1 に示す。CAPTAINPLPS では、フォトグラフ図形表示機能ばかりでなく、NAPLPS が母体としているジオメトリック図形表示機能をも、そのプロトコル体系に組み込んでいる。

両者の詳細を比較してみると、共通点として次のようなものがあげられる。

まず、符号体系であるが、これはどちらも国際標準 ISO 2022 に準拠するものである。したがって、両者の符号拡張方式は比較的類似したものになっている。

表 1 CAPTAINPLPS と NAPLPS の比較
CAPTAINPLPS and NAPLPS.

項目 \ 方式	CAPTAINPLPS	NAPLPS
母体となる表示機能	フォトグラフィック	ジオメトリック
着色方式	ブロック着色 ドット単位着色	ドット単位着色
文字・記号表示機能	アルファベット, 数字, 記号, 漢字, 片カナ, 平ガ ナ	アルファベット, 数字, 記号
図形表示機能 フォトグラフィック	あり	なし
ジオメトリック	あり	あり
表示領域(標準)	192×248 ドット	200×256 ドット
表示文字数	(標準) 15 列×8 行(漢字) 31 列×16 行(英・数・ カナ) (高密度) 31 列×16 行 62 列×32 行	(標準) 40 列×20 行

また、ジオメトリック図形を表示する場合には、その画面着色方式はどちらもドット単位着色である。さらに、NAPLPS と CAPTAINPLPS では、ジオメトリックプリミティブの描画コマンドについては完全に一致している。

次に、両者の相違点であるが、これには以下のようなものがあげられる。

まず、表示領域の大きさであるが、NAPLPS が標準で 200×256 ドットであるのに対して、CAPTAINPLPS では 192×248 ドットである。すなわち、縦横とも CAPTAINPLPS の方が 8 画素ずつ少ないわけであり、NAPLPS 画面データと全く同一の座標情報を用いると、この部分が欠落することになる。

また、大きな相違点として、文字表示方式の違いがあげられる。NAPLPS の文字表示方式はドット単位着色であるのに対し、CAPTAINPLPS のそれはブロック着色である。このため、NAPLPS の場合は、文字を画面内の任意の位置に任意の大きさで表示できる他、文字を回転して表示することが可能である。また、キャラクタパスを上、下、左、右どの方向にも設定することができ、さらに、文字間スペーシングや行間隔の変更を行うことも可能である。これに対して、CAPTAINPLPS では、文字は画面内のあらかじめ定められたマス目内にしか表示できないうえ、文字サイズも既定の 6 種類だけが使用できる。文字の回転、キャラクタパス、文字間スペーシング、および行間隔の

変更等は困難である。

1画面に表示可能な文字数は、CAPTAINPLPSでは標準端末で、漢字なら最大15×8文字、英数字、仮名なら最大31×16文字を表示する。一方、NAPLPSの場合は、40×20文字を表示可能である。ただし、NAPLPSでは文字の大きさを任意に設定できるので、判別できるか否かを別にすれば、さらに多くの文字を表示することも可能である。

さらに、NAPLPSではジオメトリック図形の描画点と文字の表示位置を同一にすることが可能であり、両者が同一フレーム上に表示されることから、図形表示と文字表示は、相互に大きな影響を及ぼし合っている。これに対し、CAPTAINPLPSでは図形と文字は全く別のフレーム上に表示され、両者は完全に独立している。

このように、文字表示方式に関して、NAPLPSとCAPTAINPLPSとは根本的に異なるため、文字をコードレベルで変換することは不可能である。たとえ変換できたとしても、ジオメトリックの機能を生かした動的な文字表現（たとえば、文字上から図形を描き、さらにその上に文字を表示する）を行うことは困難である。

NAPLPSとCAPTAINPLPSのその他の相違点としては、いくつかのPDIコマンドや制御コマンド、マクロ定義の方法、色指定、色値設定方法等があげられる。これらは、複数のCAPTAINPLPSコマンドで翻訳したり、変換処理系の内部状態を制御することで、その効果をCAPTAIN端末上に再現することができる。

4. 文字の変換

前章で述べたような文字変換に伴う問題点を解決するために、NAPLPSオブジェクトデータに文字コードが現われるごとに、それに該当する（すなわち、その文字フォントを描くような）ジオメトリック描画コマンドのセットに変換する。たとえば、図1に示すようなAの文字フォントは、5本の直線で構成されるから、5つのLINEコマンドに変換される。

プロトコル変換実行中は常に現描画点と文字表示属

図1 字文フォントのジオメトリック表現
Character pattern drawn by PDI.

性を追跡すれば、各LINEコマンドのオペランドに対して適当な演算を施すことで、NAPLPSの文字表示をCAPTAIN端末上で完全に再現することが可能である。たとえば、スケーリングを施すことで文字を任意の大きさに設定でき、また座標変換を行うことで文字の回転表示を再現できる。また、図形として描かれるので、動的な文字表示も可能となる。

さて、NAPLPSの持つ文字セットのうち、Primary文字セットとSupplementary文字セットに関しては、文字数が少なく（各94文字）、かつ使用頻度も高いので、各文字に対するジオメトリック描画コマンドセットをあらかじめ生成しておき、変換中に必要に応じてこれをアクセスする。しかし、日本国内で使用されている2バイト系文字セットである漢字セットは、きわめて文字数が多く（JIS C 6226 第1水準相当で2965文字程度）、かつ複雑な形状のものが多く、あらかじめ各文字フォントに対応するジオメトリック描画コマンドセットを生成しておくのは非常に困難である。また、処理系自体の規模も大きなものになってしまう。

そこで、NAPLPSデータに漢字コードがある場合には、漢字ROMから該当する文字フォントを読み込み、この文字フォントから自動的にジオメトリック描画コマンドセットを生成する。もちろん、この方法であらかじめジオメトリック描画コマンドセットのファイルを作成しておき、変換実行時に直接これをアクセスしてもかまわない。しかし、どちらが良いかは、実際に処理が行われるシステムの能力に依存するものと思われる。

文字フォントから自動的にジオメトリック描画コマンドセットを生成するアルゴリズムは、次のようなものである。

まず、図2に示すように、文字フォントの左下隅を

--- 走査
— 追跡

図2 文字フォントの走査
Scan of character phon.

起点として、ここから右方向へフォントの走査を行う。

未通過点があれば、ここを始点として文字フォントのパス追跡を行う。パスが分岐している場合は、現在の進行方向を最優先とする。パスがジグザグになることを防ぐため、次の優先方向を水平・垂直方向とし、斜め方向には最低の探索優先順位を与える。追跡パスの方向が変化したら、その曲点座標が、そのまま LINE コマンドのオペランドとなる。

こうして、一通り 1 本のパス追跡が完了したら、当該パスの始点の右隣の点から、文字フォントの走査を再開する。未通過の点があれば、これを新たなパスの始点として、フォントパスの追跡を再開する。

以上の手順を、文字フォント上に未通過点が無くなるまで反復すれば、当該フォントに沿った数本のパスを得ることができる。各パスに対して 1 つの LINE コマンドを割り当て、各パスの曲点座標をそのオペランドにすることで、当該文字フォントを描く LINE コマンドセットを生成する。

しかし、こうして得られたオペランドを拡大すると、文字フォントの連続性が保存されなくなってしまう。

追跡パスが、通過済み領域と未通過領域の両者に隣接した場合、パスは当然未通過領域の方へ進んでいく。この通過済み領域と当該パスとの間に不連続面が発生するが、この両者を結ぶパスは決して存在しない。こうして、パス追跡終了後も、この不連続面は残ってしまう。そこで次のような処理を上述のアルゴリズムに付加することにする。

図 3 文字描画コマンドセット生成アルゴリズム
Conversion algorithm from character pattern to geometric commands set.

●始点付加探索

パスの始点が通過済み領域に隣接する場合に、両者を連結する処理であり、始点をこの通過済み領域上に設定する。(図 4)

●終点付加探索

追跡パスが通過済み領域に隣接して終了した場合に、両者を連結するための処理であり、終点をこの通過済み領域上に進めるものである。ただし、再設定された終点の先に、さらに未通過点があるか否かも探索し、もしあれば追跡は続行される(すなわち、当該通過済み領域を追跡パスが通過する)(図 5)。

●強制接続

追跡パスの曲点が図 6 に示すように通過済み領域に隣接する場合、両者を連結するような LINE コマンドを新たに生成する。

以上のアルゴリズムによって得られるジオメトリック描画コマンドセットは、拡大に対して連続性が保存

図 4 始点の付加探索
Additional serch at start point.

図 5 終点の付加探索
Additional serch at end point.

図 6 強制接続
Forced connection.

されるばかりでなく、1本の直線パスを1LINE コマンドで表現せずに、曲点を含む1本のパスを1LINE コマンドで表現しているの、情報量圧縮の意味でも効率の良い手法である。図7にパスの追跡例を示す。

図7 パス追跡例

Example of character phont pattern tracking.

写真1(a) 始点、終点の付加探索、および強制接続を行った場合のフォントパス追跡例

Example of phont pattern tracking with additional search at both start and end point and forced connection.

写真1(b) 始点、終点の付加探索、および強制接続を行わない場合のフォントパス追跡例

Example of phont pattern tracking without additional search at both start and end point and forced connection.

また、写真1(a)に、始点、終点の付加探索および強制接続を行った場合のフォントパス追跡例を、写真1(b)にこれらの処理を行わなかった場合のフォントパス追跡例を示す。もとのフォントは、図7のものであり、キャラクタフィールドサイズは、どちらも左から順に8×8, 12×12, 16×16, 20×20, 24×24, 32×32, 50×50である。もとのフォントの大きさが16×16ドットなので、これより大きい場合は拡大を、小さい場合は縮小を行っている。写真1(b)では拡大によって文字に不連続部が生じているが、写真1(a)では拡大しても不連続部が生じないことがわかる。

5. 画面の縮小

3章で述べたように、CAPTAINPLPSの標準画面はNAPLPSのそれより縦横とも8ラインずつ小さい。このためNAPLPS画面のこの領域に重要な情報がある場合には、プロトコル変換を行う際にNAPLPSの座標データを縮小する必要がある。いまNAPLPSとCAPTAINPLPSの表示領域を各々 N , $C \subset \mathbb{R}^2$ とすると、 $\forall x \in N$, $x = (x_1, x_2)$, $\exists y \in C$, $y = (y_1, y_2)$

$$f: x \rightarrow y, f(x_i) = ax_i \quad (i=1, 2) \quad (1)$$

なる連続写像 f によって画面は縮小される(ただし、 $a=248/256$ or $192/200$)。しかし、実際は $N, C \in \mathbb{Z}^2$ とみなすことができ、 $[]$ をGauss記号として、

$$g: x \rightarrow y', g(x_i) = [ax_i + 1/2] \quad (i=1, 2) \quad (2)$$

なる写像 g によって縮小される。ただし $\exists y' \in C$, したがって誤差 $e_{x_i} = y_i - y'_i$ ($i=1, 2$), $|e_{x_i}| < 1/2$ が生じる。絶対座標形式コマンドの場合、そのオペランドが示す各点における誤差はこの e_{x_i} ($i=1, 2$)であるが、相対座標形式コマンドの場合、誤差は累積し、オペランドの j 番目の点での累積の誤差 E_{rj} は、

$$E_{rj} = \sum_{l=1}^j e_{x_{il}} \quad (i=1, 2)$$

(ただし、 $e_{x_{il}}$ ($i=1, 2$)は、 l 番目の点での計算誤差)、 E_{rj} は1を超えるかもしれない、その場合、図形はひずむことになる。また、他の絶対座標系コマンドで描かれた図形との位置関係も狂うことになる。

そこで、以下の手順によって縮小を行う。

Step 1: 縮小しようとする図形の各頂点の絶対座標

$$x_{ak} = (x_{ak1}, x_{ak2}) \quad (k=1, 2, \dots, n)$$

(n は頂点数)を求める。

Step 2: $y'_{aki} = g(x_{aki})$ ($i=1, 2; k=1, 2, \dots, n$)

なる $y'_{aki} = (y'_{aki1}, y'_{aki2})$ を求める。

Step 3: 絶対座標形式コマンドの場合には、 y'_{aki} をその縮小時のオペランドとする。相対座標形式コマンドの場合には、 $y'_{ak} - y'_{ak-1}$ ($k=2, 3, \dots, n$) をその縮小時のオペランドとする。

以上の手順を用いれば、相対座標形式コマンドの場合でも、その j 番目の点での累積誤差は $E_{rj} = e_{xij}$ であり、 $|E_{rj}| < 1/2$ である。したがって、縮小後の図形がひずんだり、図形どうしの位置関係が大きく狂うことを防ぐことが可能となる。

図 8 NAPLPS → CAPTAINPLPS オンライン変換実験系
Experiment of on-line conversion from NAPLPS to CAPTAINPLPS.

6. オンライン変換の実験

図8に示すようなオンライン変換の実験系を構成し、NAPLPS端末とCAPTAIN端末に、同時にNAPLPS画面を表示させた。

図形の変換は、ほぼ同時に行えるが、文字表示に若干遅延を生じる（1文字当り約0.05秒）。しかし、ほぼ実時間で表示を行うことが可能である。

写真2(a)にNAPLPS端末に表示した画面を、同(b), (c), (d)にCAPTAIN端末で表示した画面を示す。(b)はNAPLPS画面の縮小を行わなかったものであり、右端と上端が欠落している。同(c), (d)は、各々248/256, 192/200の縮小を行ったものである。縮小によって文字に若干のひずみが生じているが、これについては現在、縮小時専用の小型文字描画セットを適用する手法を検討中である。

文字の形状や大きさが、NAPLPS画面と少し異なって見えるが、これは、この実験に使用したNAPLPS端末の文字フォント形状と変換処理系内部の文字フォント形状が異なるためである。NAPLPSでは、文字フォントの形状や大きさは、それがキャラ

(a) NAPLPS画面

(b) CAPTAINPLPS変換画面

(c) CAPTAINPLPS変換画面

(d) CAPTAINPLPS変換画面

写真 2 NAPLPS → CAPTAINPLPS 変換画面
Converted pictures from NAPLPS to CAPTAINPLPS.

クタフィールドサイズ内におさまるものであれば、何でも良いことになっている。すなわち、文字フォントの形状は NAPLPS 端末の種類によって異なるわけであり、これを特定の NAPLPS 端末について忠実に再現しても無意味である。

なお、現在、オンライン変換の実験で変換可能であることを確認した NAPLPS コマンドは、Primary 文字セット、Supplementary 文字セット、JIS C 6226 セット、インクリメンタルコマンド以外の PDI セットである。マクロセット、モザイクセット、DRCS、インクリメンタルコマンドは未確認であるが、これらに対してオンライン変換を行うことは、原理的に充分可能である。

7. む す び

以上、ランク 4 以上の端末を対象とした NAPLPS

から CAPTAINPLPS へのプロトコル変換法について述べた。本手法によれば、NAPLPS の色情報や形情報、そして漢字も含むすべての文字情報を完全に変換することができる。また、オンラインでの実時間変換も可能である。

したがって、海外（主に北米）ばかりでなく、国内の NAPLPS を用いたプライベートビデオテックスとも、CAPTAIN システムを有機的に接続することが可能である。

(昭和 60 年 3 月 1 日受付, 昭和 60 年 4 月 22 日再受付)

〔参 考 文 献〕

- 1) 宮里, 小花, 中尾, 羽島: “国際ビデオテックス相互接続と変換処理”, テレビ学技報, 8, 39, pp. 13-18 (Dec., 1984)
- 2) 浜野, 渡部, 小林: “ビデオテックスにおけるプロトコル変換”, テレビ学技報, 8, 39, pp. 7-12 (Dec., 1984)
- 3) 小林, 渡部: “ビデオテックスにおけるプロトコル変換”, 昭 59 信学全大, 分冊 3, p. 3-261 (1984)