

I - 5

一般の描画ソフトウェアを利用した
点図作成方法と点図出力支援ソフトウェアの試作

I - 5 一般の描画ソフトウェアを利用した点図作成方法と 点図出力支援ソフトウェアの試作

1. はじめに

前節で記したように、新しく開発した点字プリンタ“NISE Graphic”に対応した点図作成ソフトウェアは存在しない。

これは、開発の元となった点字プリンタESA721 Ver'95と比較しても、前述のように、ピッチ（座標を指定できる最小の間隔）が異なること、及び、座標を指定する制御コードが変更されたことによる。ESA721 Ver'95ではASCIIコード2文字でY座標（この場合、横座標）、次いで同2文字でX座標（縦座標）、計4文字のASCIIコードで座標を指定していたが、“NISE Graphic”では、X座標（縦座標）を2バイト、次いでY座標（横座標）を2バイト、計4バイトのバイナリコードで指定する方式になった。

この“NISE Graphic”に対応した点図作成ソフトウェアを作成するための方針として、前述のように、EDELやDot Draw NISE（当研究所で開発中の「点図くん」の後継版）のような点図作成専用のソフトウェアが、この点字プリンタに対応するようになるという方針もある。また、新たに、“NISE Graphic”用の点図作成ソフトウェアを開発するという方針もある。

一方、通常の描画に用いられている一般の描画ソフトウェアを点図作成用に使えるようにするという方針もある。EDELやDot Draw NISEは、その画面上で、点図を描画して、点図出力までを行うものだが、点図の描画について、一般の描画ソフトウェアで、点図を描画、編集できれば、機能も豊富で、使いやすいという点で、そのほうがよいとも考えられる。

ここでは、Illustrator という一般の標準的な描画ソフトウェアの描画機能を用いて点図を描き、そこから各点の座標値を抽出し、点図印刷プログラ

ムに送って点図を作成するという方針での、“NISE Graphic”に対応した、点図作成・出力手順、および、それを可能にする、点図出力支援ソフトウェアの試作について記す。

なお、EDEL や Dot Draw NISE にも、BMP や JPEG形式の描画ファイルを読み込んで、それらソフトウェアの形式の点図に変換する機能はあるが、ここで示すのは、Illustrator上で、点図を描画、編集、完成させて、その各点の位置データのみを点字プリンタに送る方法である。即ち、点図を作成するという点に関しては、全て、Illustrator上で行う方法である。

以下、Illustrator のバージョンはCSである。なお、Illustrator は、グラフィックデザインの世界では、最も一般的な描画ソフトウェアであるといつてよいものである。

2. 点図作成から点図出力までの手順

1) Illustrator で点図を描く方法

Illustrator で、点図の場合と同様の点線を描くには、以下に記すように [線種] を設定すれば、可能である。

しかし、そのままでは、各点は線の各部分としてつながったままで、各点の座標値を取ることが出来ないため、点字プリンタ出力用の点図としては利用できない。しかし、次のように、オブジェクトに対する透明部分の分割・統合、およびグループ解除、複合パスの解除の機能を使うと、その各点を分離して、別々のオブジェクトにすることができる。

具体的には、以下のようにする。

① [線種] パレットで、次のように、線の種類を、点線を描けるように設定する。(図1参照)

- ・ [線幅] を設定したい点の大きさ（例えば、1.7mm）と同じにする。
- ・ その右にある線端の選択を「丸型線端」（図で



図1 点線を描くための線種の設定

は上真ん中のアイコン) にする。

- ・[破線] をチェックして、線分を「0 mm」、間隔を設定したい点間 (例えば、2 mm) に設定する。

②この設定で、点線を描く。

ここで、描画できる範囲について述べると、“NISE Graphic” で、9×10インチ用紙を用いて出力する場合の最大出力範囲は、横162.6108mm、縦231.1386mm である。最大出力範囲より、少し余裕をもたせるため、ピッチに対応して、縦

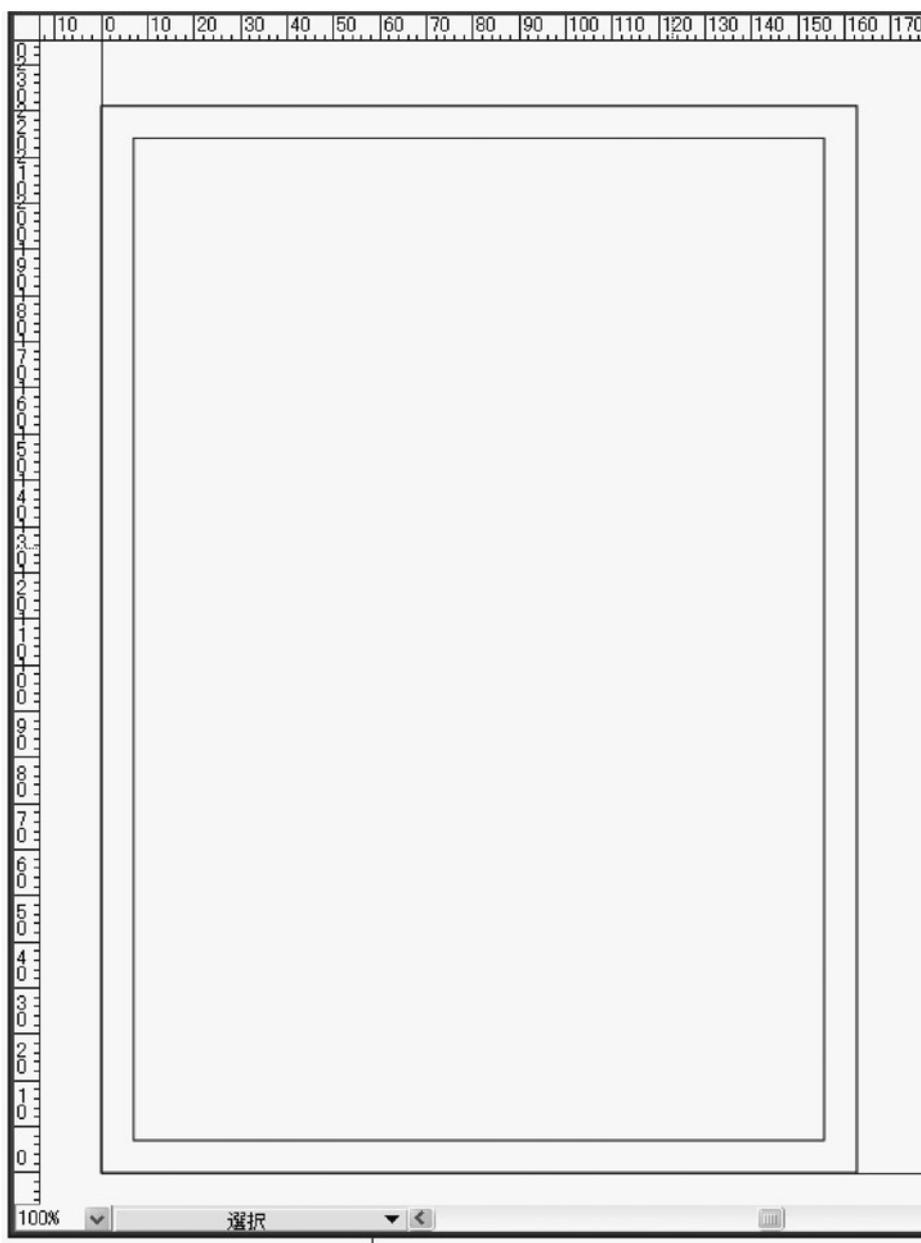


図2 点図の描画範囲

外側の枠が最大描画範囲で、内側の枠が、約7mmの余白をとった範囲。

横とも7mm程度の余白をとると、余白は、横7.0104mm、縦7.0042mmで、出力範囲は、横148.59mm、縦217.1302mmとなる。

Illustratorでは、左下が原点なので、これにあわせて、最大出力範囲と、先の余白をとった範囲を示すと、図2のようになる。

この範囲が、9×10インチ用紙での出力対応範囲となる。

従って、図の、内側の枠内が、描画できる範囲である。

なお、後に述べるJavaScriptでは、この範囲を越えて描画した場合に、実行すると、「描画範囲を越えています」という警告が出るようにしている。

③描いた点線を選択して、その各点を別々のオブジェクトにする。

- ・点線をツールボックスの選択ツールで選択する。
- ・メニューバーで、【オブジェクト→透明部分を分割・統合】を選択し、「OK」をクリックする。
- ・【オブジェクト→グループ解除】を選ぶ。(あるいは、選択した状態で右クリックしても「グループ解除」の項目が出てくる)
- ・【オブジェクト→複合パス→解除】を選ぶ。(あるいは、選択した状態で右クリックしても、「複合パスを解除」の項目が出てくる)

これで、各点が分離され、別々のオブジェクトになる。

なお、点同士が重なってしまった部分や、接近しすぎてしまった部分は修正する。Illustratorでは、角の部分で、そうなることがあるので、その修正は必要である。

2) Illustratorの点図から、各点の座標値をとる方法

各点の座標値は、各点を選択して、[情報]パレットや、[変形]パレットを見れば、知ることが出来る。

自動的に、各点の座標値を得て保存するには、Illustratorで、その作業を自動化するために利用できようになっているJavaScriptで、そのような

Scriptを記述すれば、可能である。^{1) 2)}

ここでは、点図出力したい点図を選択ツールで選択して、実行すると、各点のx座標(横座標)、y座標(縦座標)の各座標値を取得して、点字プリンタ“NISE Graphic”のピッチに対応したX座標(縦座標)、Y座標(横座標)の各座標値に変換して、それぞれ", "区切りでTextファイルに保存するJavaScriptを作成した。なお、今回は、“NISE Graphic”で出力できる、大、中、小の各点のうち、中点に対応するScriptのみを作成した。

なお、座標値の変換については、Math.round関数で、“NISE Graphic”のピッチの座標に換算した際的小数点以下を四捨五入しているため、座標値のずれは、最大でも、“NISE Graphic”の最小ピッチの半分、即ち0.025mm程度となる。従って、Illustrator上で、描いた点図とほぼ同じものを、点字プリンタの点図として出力できると考えられる。逆に、これは、Illustrator上で、描画作業を全て行えるということでもある。

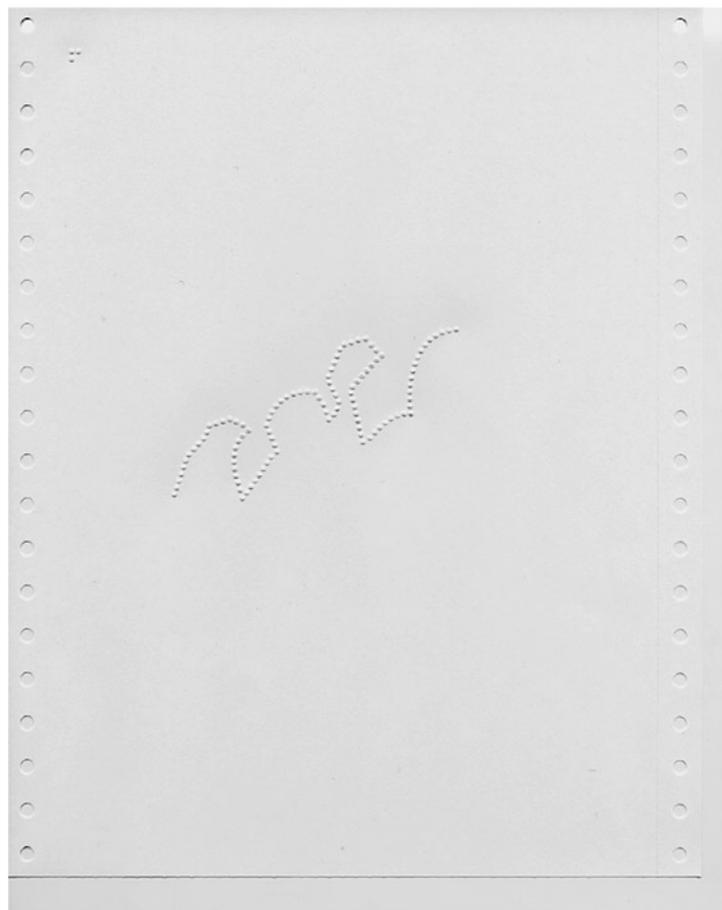
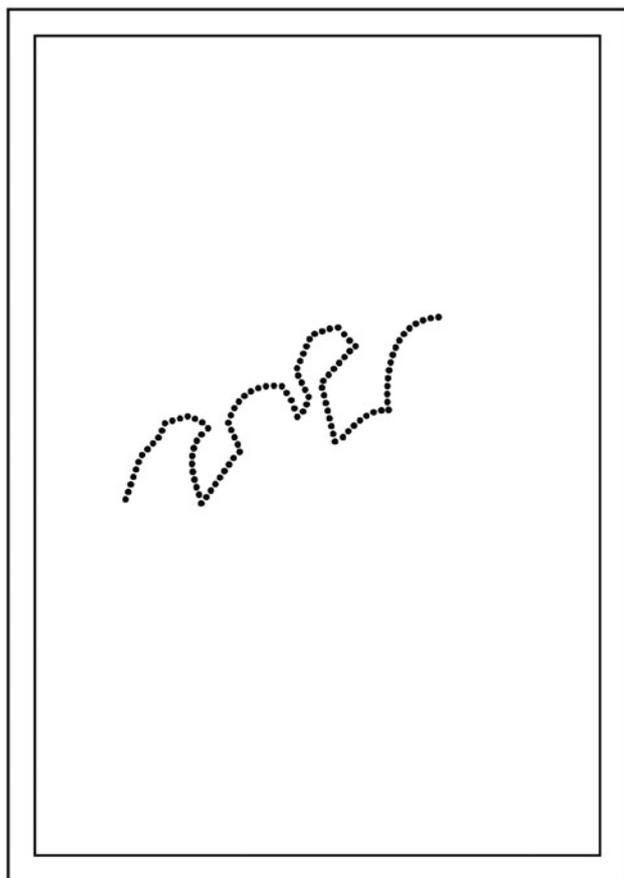
Illustratorで、このJavaScriptを実行するには、点図出力したい点図を選択ツールで選んでから、【ファイル→スクリプト→参照】で、そのファイルを選択し、「開く」をクリックする。

なお、その際、選択した点図が、“NISE Graphic”の出力範囲を越えている点を含んでいる場合は、このJavaScriptを実行した際、「印刷範囲を越えています」という警告を出すようにした。

3) 点図印刷プログラムでの点図出力

現状では、BASIC言語のプログラムで、comポートを開き、グラフィックモードに入り、大・中・小の点種を指定して、座標値をバイナリコードに変換し、comポートに送るというかたちのプログラムがある。具体的には、BASIC/98 for Windows Version 4.0を使用したものである。

これに、2)で作られた、Illustratorの図の各点に対応するX座標、Y座標のファイルを読み込んで、座標の数値を得る部分を追加したプログラムを作成した。



a. 元の図（左）と点図出力したもの（右）（全体）

図3 点図の出力例（1）

日本地図の、房総半島、三浦半島、伊豆半島の部分。

3. 点字の挿入と出力について

点図には、その全体の説明や、その各部の説明などとして、点字が添えられていることも普通である。

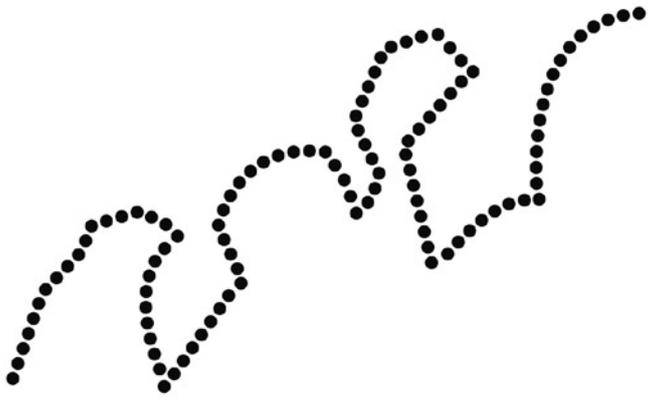
ここで、上記の方法の場合の点図に点字を添える方法として、点字プリンタ自体がもっている点字出力機能を用いる方法もあるが、実は、点字フォント（注）を用いて、点図を描画する際に点字を付加して、それを、上記と同様の方法で、各点を分離すれば、同様に点字プリンタ出力することが可能である。それには、先の【オブジェクト→透明部分を分割・統合】の部分で、「すべてのテキストをアウトラインに変換」にチェックをすれば、可能である。

ただし、この場合の点字出力では、通常の点字出

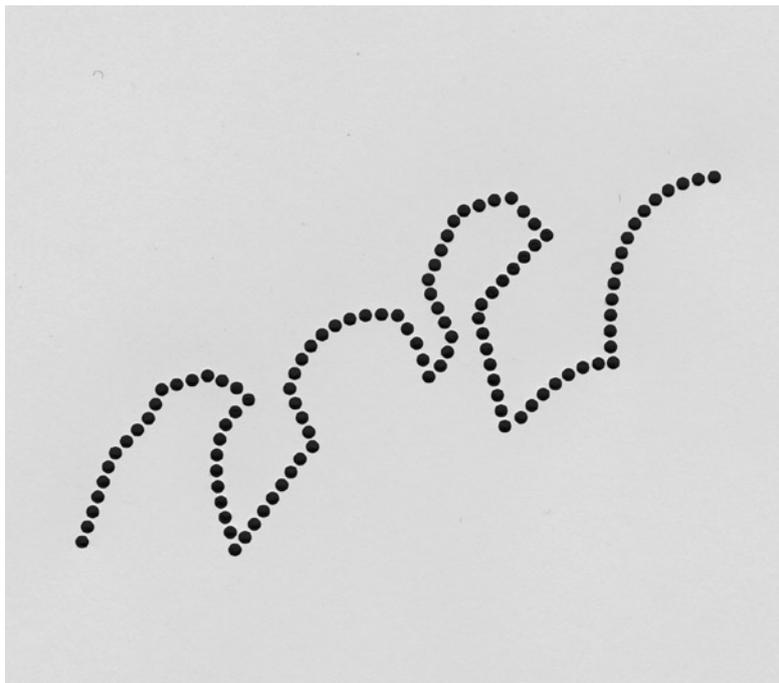
力の場合のように、各文字ごとに打つのではなく、その各文字の各点を一点ごとに打っていくことになる。しかし、“NISE Graphic”の精度であれば、それでも、十分きれいな点字が出力されることが予想される。

なお、点字フォントを90度回転させれば、横向きの図に、同じ横向きで、点字を添えるといったことも容易にできる。

注：点字フォントは、コンピュータ上で使用される、明朝体やゴシック体などと同じものであり、同様に、図に挿入したり、ワープロ上の文字として使えるものである。



b. 元の図（左）と点図出力したもの（右）（実物大）



c. bの両者を重ねてみたもの

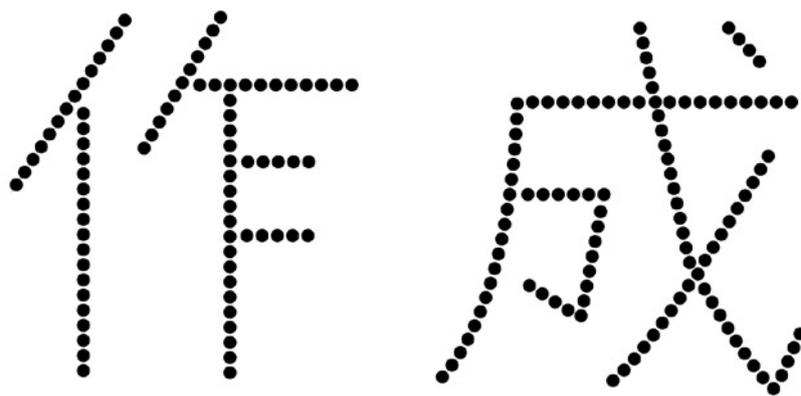
図3 点図の出力例（1）

日本地図の、房総半島、三浦半島、伊豆半島の部分。

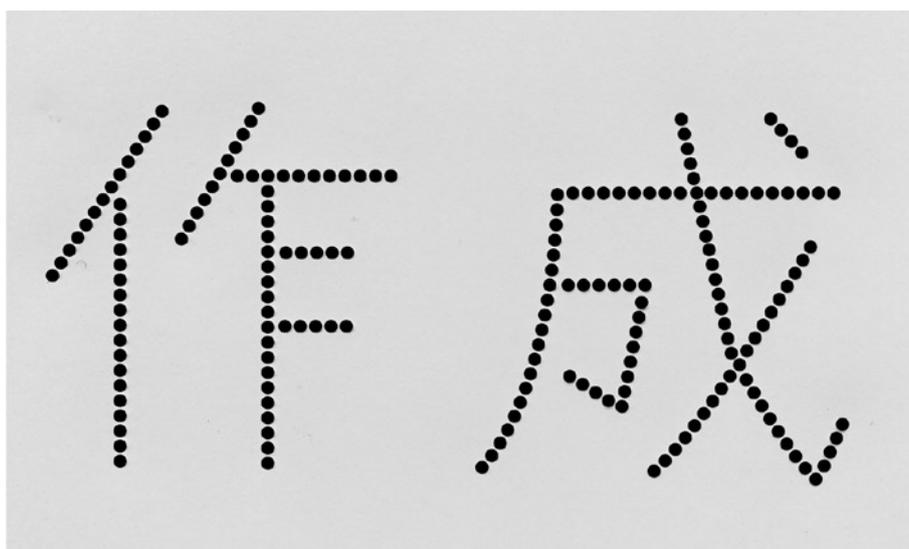
4. 点図の出力結果

以上の手順により、Illustrator で点図を描画し、

“NISE Graphic” で点図出力した結果例を、元となったIllustrator の図とともに、図3～6に示す。また、両者を重ねてみた結果も示す。図5、図6で

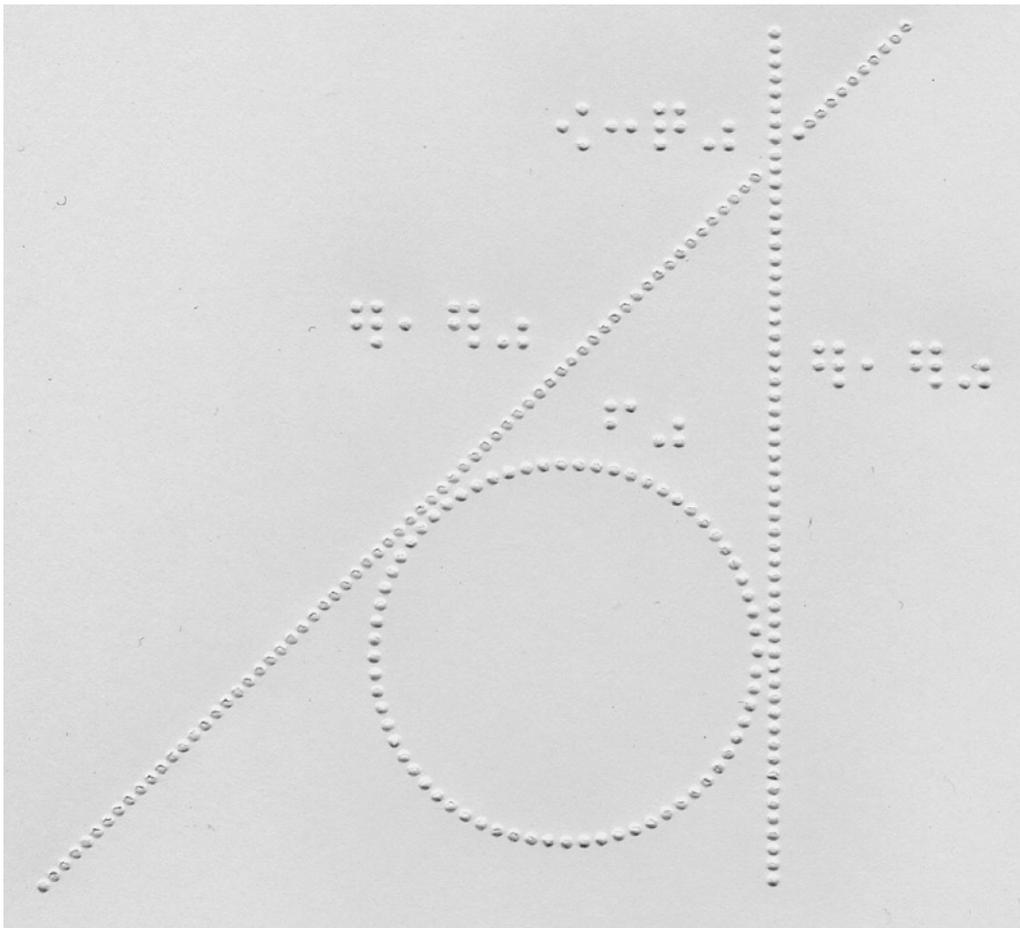
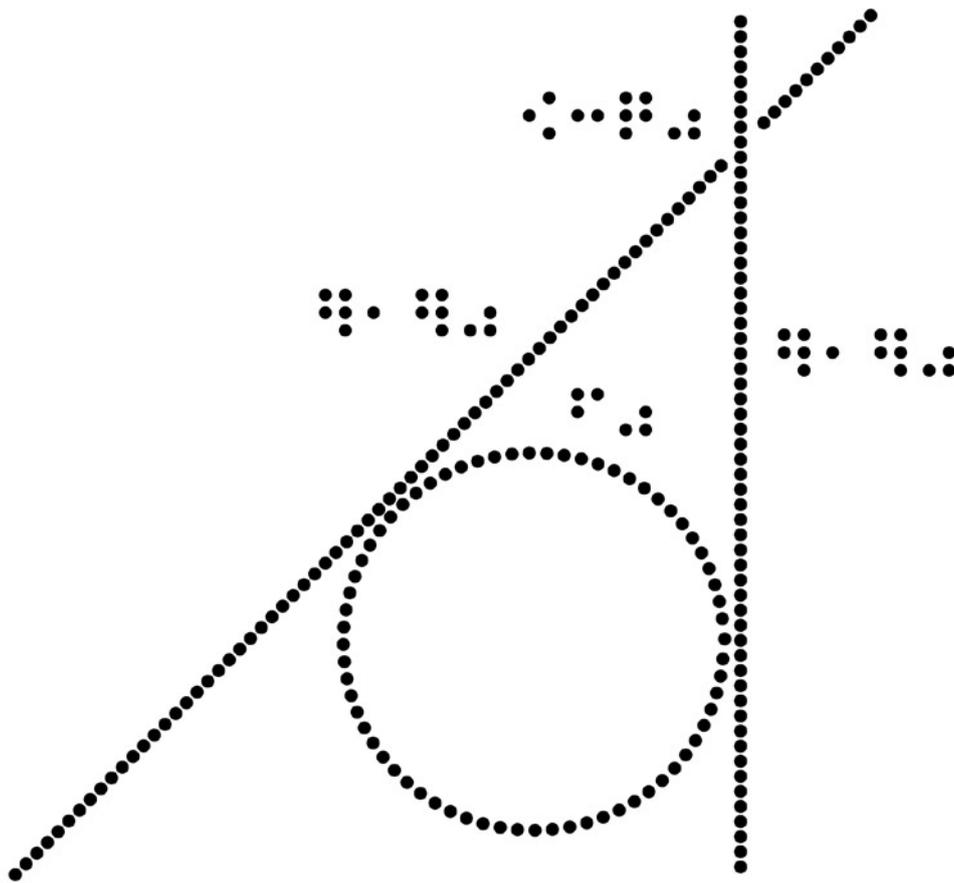


a. 元の図（上）と点図出力したもの（下）（実物大）



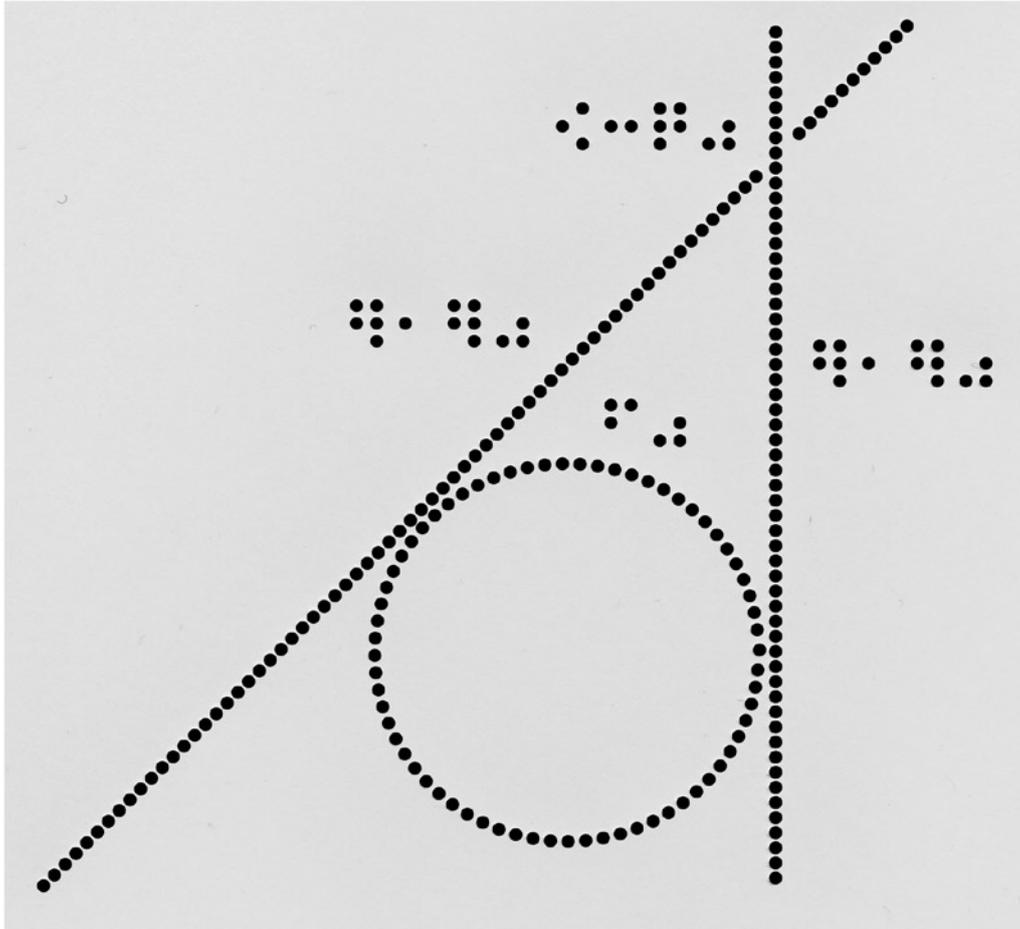
b. aの両者を重ねてみたもの

図4 点図の出力例（2）
漢字の例で、「作成」。



a. 元の図（上）と点図出力したもの（下）（実物大）

図5 点図の出力例（3）
幾何図形の例。点字も出力している。



b. aの両者を重ねてみたもの

図5 点図の出力例(3)

幾何図形の例。点字も出力している。

は、点字を出力した結果も示す。図6は、図、点字とも横向きである。なお、点字について、使用した点字フォントは、日本ライトハウス作成のもの(注)である。

このように、元の図と、そこから位置座標をとって点図を出力した両者の各点は一致していると言ってよいと思われる。

また、点字フォントを元にした点字出力についても、元となった点字フォントの点字と、出力された点字の各点は一致していると言ってよいと思われる。

注：日本ライトハウス点字情報技術センターのWebページ (<http://www.eonet.ne.jp/~tecti/index.html>) からダウンロード可能である。

5. まとめ

新しい点図出力可能な点字プリンタが開発された場合、それ専用の点図作成ソフトウェアを開発したり、既存の点図作成ソフトウェアが、その点字プリンタに対応するようにするという方針があり得る。

しかし、一方、ここで述べたように、Illustratorのような通常の図を描くための、一般的、かつ標準的な描画ソフトウェアで点図を描けば、それが、ほとんどそのまま、点図出力できるということも可能である。

ただし、これが可能であるのも、“NISE Graphic”の精度が、前節でも述べたように、縦、横ともほぼ0.05mmのピッチ(注)で、座標を指定し、点図出力できるものだからである。即ち、先に述べたように、元となる図の各点に対して、その出

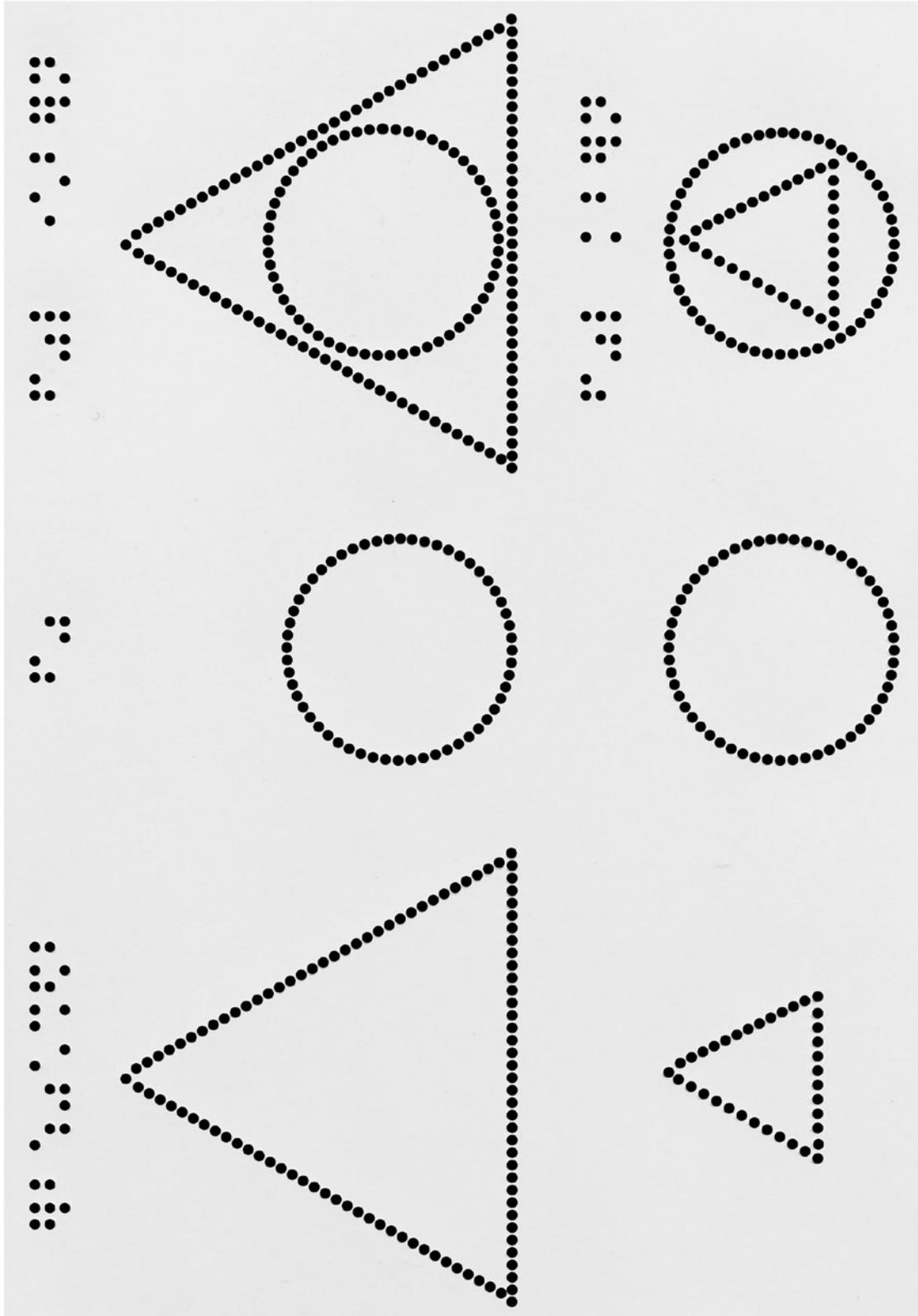


図6 点図の出力例(4)(実物大)
幾何図形の例。図は横向き用で、点字も横向きで出力している。
元の図と点図出力したものを重ねた結果のみ示す。

力結果が、最大でも、0.025mm程度のずれしか起こらないと想定できるからである。

もし、そのずれが、もっと大きい場合、例えば、“NISE Graphic”の開発の元となった、ESA721 Ver'95のように、縦、横とも0.3mm程度のピッチ（注）で座標を指定しなければならない場合、前節での出力評価の結果からも予想できるように、ここで述べたような方法では、元となる図の各点と、出力された各点の位置は、ずれることが考えられる。

精度の高い点図出力可能な点字プリンタを開発することの1つの意義は、ここで示したような、一般の描画ソフトウェアを用いても点図出力を可能にするといった、それに対応する点図作成ソフトウェアを作成する上での利点をもたらすことであると言える。

なお、点字出力についても、精度の高い点字プリンタは、上記のような、点字フォントを用いた方法を可能にすると言える。この方法は、大量の点字を打つことには向かないであろうが、点間を自由に変えられる、また、必要であれば、点字の向きも変えられるという利点もある。前者については、点間を広め取ることで、点字触読の初心者に、より分か

りやすい点字を提供できる。後者については、先に示したように、横向きで使用する用紙に、その向きで点字を添えるといったことは容易である。

注：前節に記した、2台のプリンタのピッチについて、表を再掲する。

2台の点字プリンタのピッチ

	NISE Graphic	ESA721 Ver'95
横	0.05080mm	0.3454mm
縦	0.05003mm	0.3175mm

文 献

- 1) Adobe Developer Support : Adobe Illustrator CS JavaScript Reference.
<http://partners.adobe.com>, 2003.
- 2) 古籬一浩 : Illustrator CS自動化作戦with JavaScript.
<http://www.openspc2.org/book/IllustratorCS/>, 2005.
(金子 健・大内 進)