

LaTeX のためのドット外字パターンエディタ

岩熊哲夫 (2013 年 12 月 8 日)

1 概要

スタイルファイル '`xbmkanji.sty`' を利用すれば、任意の正方形の X-Bitmap ファイルを漢字一文字として使うことができる。その漢字一文字分の X-Bitmap ファイルを作る。システムが持っている漢字のパターンを加工すると便利なのだが、古い Visual BASIC にあった '`Point`' という関数 (?) が無くなったため、`PictureBox` に描いた漢字を一旦画像ファイルに保存して、そのビットマップの色で 64×64 の離散パターン化。システム漢字フォントの完全なコピーではないので、これが著作権の侵害にはなっていないことを祈る。旧版とは異なり、サイズは 64×64 のみ。単に面倒だったから。

2 インストール

普段使いたいディレクトリに、実行ファイル '`XbmGaiji.exe`' とテンポラリの画像ファイル '`Xbmtemp.bmp`' を置いて実行するだけ。この画像ファイルは自動的に生成されないらしい（プログラム開発しているときにどうやって生成されたのか不思議で、開発中にいろいろ問題が生じた。必ずこの画像ファイルを実行ファイルと同じディレクトリに置いておくこと）。ただし、Visual BASIC 2010 で作ったので、ライブラリ '`.NET Framework`' が必須。これはオプションライブラリ（無料）なので

<http://www.microsoft.com/downloads/ja-jp/details.aspx?>

FamilyID=9cfb2d51-5ff4-4491-b0e5-b386f32c0992

にアクセスしてダウンロードしインストール。

アンインストールは、そのディレクトリごと削除するだけで終了。オプションライブラリは、ま、削除してもしなくても・・・

3 ドットパターンの作成

3.1 お絵描き

左の窓が作成窓。カーソル位置座標は画面中央に。その上がプレビュー。ペンや筆でパターンを入力するお絵描き作業で、漢字のようなパターンを作成するのが基本。

左ボタンの押し下げ: カーソル位置に 1 ドット追加。

右ボタンの押し下げ: カーソル位置の 1 ドットを削除。

シフトキーだけを押したまま: 2×2 のドットの追加と削除。

コントロールキーだけを押しのまま: 3×3 のドットの追加と削除。

シフトキーとコントロールキーを押しのまま: 細い直線の追加と削除。

ボタンを押したままマウスを動かすのも可だが、心持ちゆっくり動かすこと。太めにしてマウスを動かせば、ちょうど筆で描いたようなものや、消しゴムで消したようなものに相当する。不思議なことに、最初に何か線を引いたあとの初めての Alt キー押し下げで線が消えるので注意。他のプログラム (CBeam.exe 等) でも同様なので、Visual BASIC 2010 の仕様のようなのだ。理由がわからんのでちょっと困ったものである。もしこれが発生したときは、左右反転等を 2 回すれば元に戻る。そのあとの Alt キー押し下げでは何も発生しない。

3.2 作ったパターンの編集

Start Editing を押す。そのボタンが赤い **End Editing** になり、その状態の間は作成したドットパターンを編集可能。途中でやめるにはその右の **Cancel** ボタンを。編集状態にある間は、右の窓周辺での後述の作業は一切できない。

領域を指定しないままの状態: 左窓で何もしないままでするのは「白黒反転」と「左右反転」「反時計回りの回転」および「リセット」。

Reverse: 押すたびに全体の白黒反転。ネガ・ポジ変換のようなもの。

Mirror: 押すたびに全体の左右反転 (鏡文字) 。

Rotate: 押すたびに 90 度ずつ反時計回りに全体を回転。

Clear All!: プレビューの下のボタンでリセット。一度だけ確認がある。

領域を指定するには: ある領域だけの編集作業も可能。まず左窓で左ボタンを押した位置からドラッグしてボタンを離した領域が矩形で囲まれる。続けて・・・

マウスボタンを領域中央で押し下げてドラッグ: これで矩形をそのまま移動可能。

領域右辺で押し下げてドラッグ: 矩形の右辺を移動させて幅の変更が可能。

領域上辺で押し下げてドラッグ: 矩形の上辺を移動させて高さの変更が可能。

矩形の左と下の辺では何もできない。面倒だったのだ。また、矩形と辺の移動はゆっくりとやること。あるいは 1 ドットずつの矩形の移動や変更は、左窓右下にある上下左右や寸法変更のボタン (**Up** や **Narrow** 等) でも可能。加齢によって、目が疲れやすくマウスの移動がスムーズにはできなくなると、位置や寸法の微調整はこのボタンの方が楽。領域の幅と高さは画面中央に表示。

領域が指定された状態で: 指定された領域内データのための各種編集操作が可能。

Copy: 押すとボタンが赤い **Place** に変わる。これで領域内データを一時的にメモリに保存。そのまま選んだ矩形領域をコピー先の位置に移動して寸法調整。寸法を変えてしまうと、ドットパターンが少し乱れるのは仕方がない。コピー位置に矩形を置いてから赤い **Place** を押せば終わり。コピー作業が終わると**編集モード**は終わってしまうので注意。

Move: コピーと同じ操作と機能。ただし、元の場所のドットがリセット（消去）されるところが、コピーとの唯一の違い。

Erase: 選んだ矩形領域内のみを消去して**編集モード**を終わる。

Reverse: 押すたびに矩形内の白黒反転。ネガ・ポジ変換のようなもの。

Mirror: 押すたびに矩形内のパターンを左右反転（鏡文字）。

Rotate: 押すたびに矩形内の左か上の正方形部分を、90度ずつ反時計回りに回転。領域が長方形であっても回転するのは左上の正方形部分のみ。

最後の三つ「白黒反転」と「鏡文字」および「回転」の場合は何度でも続けて編集可能で、**End Editing** か **Cancel** を押さないと編集状態を終えることができない。

4 システム漢字のドットパターンの利用

4.1 システム漢字のパターン化

参照データとしてシステム漢字を 64×64 のドットに離散化したものを右窓に表示して、その大きさや太さを調整し、さらに鏡文字化や回転および白黒反転し、それを左の作成窓にコピーあるいは重ね書きすることによって、部品を調達できる。

まずは窓上の「参照文字入力欄」と表示されたテキストボックスに、使いたい漢字を左詰めで入力。パターン化されるのは最初（左端）の一文字のみ。それがその右の窓に大きく表示されているはず。フォントは明朝かゴシックのいずれかだけで、右のラジオボタンで選択。この右窓に現れた文字は、窓上の **Discretize** ボタンで右窓にパターン化されて表示される。この右窓は、例えば「魑魅魍魎」のような文字の幅がぎりぎり枠内に入るように、かつベースラインがほぼ一致するように位置を若干調整した。そのため、普通の簡単な文字はなぜか上下左右にある程度の空白ができるがなぜかは不明。普通の漢字相当の外字を作る場合には、試行錯誤が必要。作成窓とプレビュー窓の赤い線がベースラインのおおよその位置。ドットパターンが細いと感じたときは、窓の上方の **Thick[ness]** の数値を 10 側に大きくする。細くしたいときは小さくする。

また参照漢字全体を小さくした部品を作りたい場合には、窓の上方の **Mag[nification]** の数値を小さくする。その都度、表示された漢字も小さくなるが、小さくなるほど細くなるの

で、同時に ‘Thick’ も調整して、判読可能なパターンに調整する。用途は限られるが、一応 85/64 までは拡大もできるようにした。ただし、右と下および右下部分は編集に含めることができない。面倒だったので。

4.2 参照漢字の編集

サイズと太さ以外には以下の変更が可能だが、こちらでは領域は指定できず、64 × 64 全体の変更になる。面倒だったので。

Reverse: 白黒反転。ネガ・ポジ変換のようなもの。

Mirror: 鏡文字。

Rotate: 90 度ずつの反時計回り回転。

4.3 作成画面に反映するには

全体の反映: ボタン **Copy** か **Append** で左の作成窓にパターンを反映。前者は白黒そのままをコピー。後者は黒のみを重ね書きでコピー。これは右窓全体の反映。

部分領域の反映: 漢字の一部分だけを反映させるには、まず右窓内でマウスボタンを押し下げたあとのドラッグでその領域を定義する。移動や寸法の変更操作は、左窓での操作と同じ。右窓での領域が確定したら、左窓に現れている領域を、反映させたい場所に移動および寸法変更して確定させたあと、ボタン **Copy** か **Append** を押す。右窓で領域の位置と寸法を調整している間、左の作成窓の同じ位置にも同じ領域が表示されるが、この右窓での領域を確定させるまでは左窓の領域は変更しないこと。変更すると左窓画面が乱れる。ボタンを一つ省略したかったのがこうなった。領域を定義したものの結局反映作業を途中でやめるときは、中央の **Cancel** ボタンを押す。

5 ファイル入出力

生データは ‘0’ と ‘1’ でドットを表現、アスキーデータは ‘-’ と ‘#’ で表現している。後者はエディタ等で直接眺めること（編集すること）が可能。

Save: パターンの生データを保存。拡張子は ‘patx’。シフトキーを押し下げてこのボタンを押した場合にはアスキーデータの保存。拡張子は ‘txt’。このアスキーデータを左から一列ずつ縦に並べて、‘-’ を ‘0’ に、‘#’ を ‘1’ に変換し、それを 4096 行のデータにすれば、それがパターンの生データ。生データとはいえ ‘0’ と ‘1’ のみのテキスト形式。

Load: 保存した生データ（拡張子は ‘patx’）を読み込み左画面に表示する。シフトキーを押し下げてこのボタンを押した場合にはアスキーデータ（拡張

子は‘txt’)の読み込み。旧バージョンのパターンファイル(ただし64ドットで拡張子が‘pat’のファイル名の拡張子を‘txt’に変更)を読み込みたい(そんな人はいないか)場合には便利かも。

Export X-Bitmap: ドットパターンデータをX-Bitmap形式に変換して保存。拡張子は‘xbm’。シフトキーを押し下げてこのボタンを押した場合には64ドットのX-Bitmap形式ファイルの読み込み。これもテキスト形式ファイル。

Drag & Drop: 3種類のファイルを窓内にドラッグしてドロップすると読み込まれる。

ファイル操作ではすべて、何も確認無しでエラー処理も無い。ファイル読み込みダイアログ窓に現れるファイルリストのアイコンをダブルクリックして読み込んだ場合、そのマウスアイコンの位置とタイミングによっては、その下が作成窓内だった場合にドットが書き込まれたり、選択領域が現れていたりするので注意。「開く」ボタンを利用した方が無難。

Windows XP等で用いた場合、読み込みダイアログ窓のうしろに隠れた窓部分には、今読み込んだはずのパターンが表示されないことがある。そんなときは、一度鏡像にして元に戻す等の操作をするとパターンが現れる。これはVisual BASIC 2010で、他の窓などで隠れた部分の描画データのrefreshが行われないための不具合。

6 どうでもいいこと

仕事を終えるには右上の $\boxed{\times}$ で。終了を右上の \times にしたので、終了直前にファイル保存についていつも質問される。それが面倒だと感じるなら、シフトキーを押しながら \times を押すと強制終了。ただし念のために作業中、パターンが存在しているときは、左窓のデータがプログラムの置いてあるディレクトリの‘xbackupx.patx’に3分毎に保存されている。作りかけのデータを保存し忘れてしまったときや、3分前の状態に戻したいときには、ひょっとしたらデータが残っているかも。プログラムのフローを知っている作者が「想定した」順番以外の作業をしたときに、エラーで終了してしまう等、おかしくなることは多々あるはず。そのときはごめんなさい。領域定義中にときどき枠が消えないことがある(呵呵)ようだが、そのときには、マウスカーソルを画面右上の $\boxed{\times}$ の位置に移動させると、なぜか消える。右下の \boxed{J} を押すと、メッセージのいくつかを日本語表示。

サンプル \LaTeX ソース‘sample.tex’をコンパイルして何ができるか確認のこと。スタイルファイル‘xbmkanji.sty’は同梱配布。画像が別に「意味のある漢字」である必要がないのは、サンプルを見れば明らか。この版にはマインスイーパーは実装しなかった。呵呵。