

北斎の新形小紋帳 (J8-QT 版) に関するメモ

SHIMURA Masato
JCD02773@nifty.ne.jp

2016 年 8 月 6 日

1

J8-QT 版ではグラフィックス環境が大幅に変更されて、混乱が生じていたが、この度 C.Reiter は *Fractal Visualization and J (4th Edition)* の前半部が出版され、著書のサポートパッケージ *addons/graphics/fvj4* も更新されて上梓された。

先にレポートしたように *fvj4* は *dwin* 本体が全面的に書き直された他は微修正で済ませることが出来たようだ。

1.1 目的

1. 新型小紋帳の図案を *J804-QT* 版で動かす。 不具合のあるものをピックアップする
2. 不具合の修正
3. さらなる改良
4. *Form* を *QT* に移植する。

1.2 ファイルのロード

1. 新型小紋帳の描画関連ファイルのコアの部分をロードする
`load '~addons/graphics/fvj4/dwin.ijs'`

2. 各自の環境で *path* を書いておく

```
hpath=: 'c:/Users/正人/j64-805-user/temp/graphics/hokusai  
/hokusai_J8/dwin/'
```

3. 関連ファイルをロードする。(本格的にはプロジェクトマネージャーを使う)

```
load hpath, 'hokusai_tool_improve.ijs'  
load hpath, 'hokusai_tool_bezier.ijs'  
load hpath, 'hokusai_J8_1.ijs'  
load hpath, 'hokusai_2.ijs'
```

NB. Hokusai_0.ijs is all in one package

2 fvj4

2.1 伊勢型紙

江戸小紋の型紙は今も鈴鹿の白子地区で彫られている。美濃和紙と柿渋を用いた伝統工芸は型紙自体も美しくコレクターもいる。かつてこの地区は紀州藩の飛び地で、型紙商人は紀州藩の高張提灯を宿に掲げて全国を行商をした。中には隠密も混じっていただろう。

2.2 小紋のピース

六つ手万字のピース。べっとりしないために(2枚の型紙を用いて)2つのピースに少し色差を付ける。

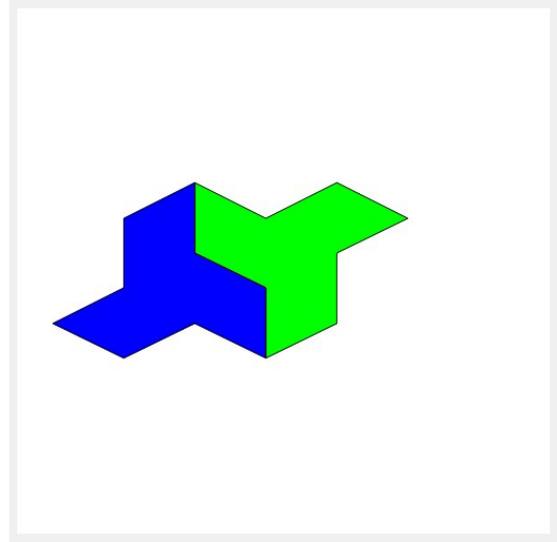
同じ図形を積み重ねる場合と隣り合う異なる図形を組み合わせる場合がある。

- 次は隣り合う2つの小紋の図であり、各々に少し色調を変えた配色をすればよい。
- 同じ図形が並ぶ場合の色調の変化は多少複雑で、プログラムで2パ

ターンに分割する (後述)

MM0;MM1

```
+-----+-----+
| 10 5 | 10 5 |
|  8 4 | 10 3 |
|  8 2 | 12 2 |
|  6 1 | 12 0 |
|  8 0 | 14 1 |
| 10 1 | 14 3 |
| 12 0 | 16 4 |
| 12 2 | 14 5 |
| 10 3 | 12 4 |
+-----+-----+
```



2.3 小紋

小紋を描くには小紋の数だけボックスを積み上げる。左は *MM0* を積み上げたもので型紙 *0* となる。

ボックスを開けるとポリゴンの線が絡まってしまうので、ボックスのまま *L:0* を付けて描く。白抜きは *MM1* の部分で型紙をもう一枚用いる。*CG* は重ね塗りが出来、後塗優先なので型紙より簡単である。

*1

差分は *xxPARAM* で与えている。左下のボックスが *MM0* で、*x* 方向 (横並べ) と *y* 方向 (縦積み) の差分を与えており、指定個数文の差分をボックスで生成する

MM0PARAM=: 10 5;6 _1;_2 5 NB. for automatic

MM1PARAM=: 10 5;6 _1;_2 5 NB. same MM0PARAM

*1 ボックスを積み上げたものは数学の群にあたるが群の演算は何もしないので群を意識しなくともよい

3 4 calc_each_poly (<MM0), <MMOPARAM

この差分計算は次のスクリプトが担っている。

```
calc_each_poly=:4 : 0
'Piece Parameter'=. y
Size=. x
Piece + ("1) L:0 Size mk_diff_sub0 Parameter
)

mk_diff_sub0=:4 : 0
'size_raw size_column'=. x
'base dfx dfy' =. y
X0=.|. { base +"1 (|. i.size_column) */ dfx
tmp=. |: >{ L:0 X0 +"1 L:0 (|. i. size_raw) */ dfy
tmp - L:0 base
)
```

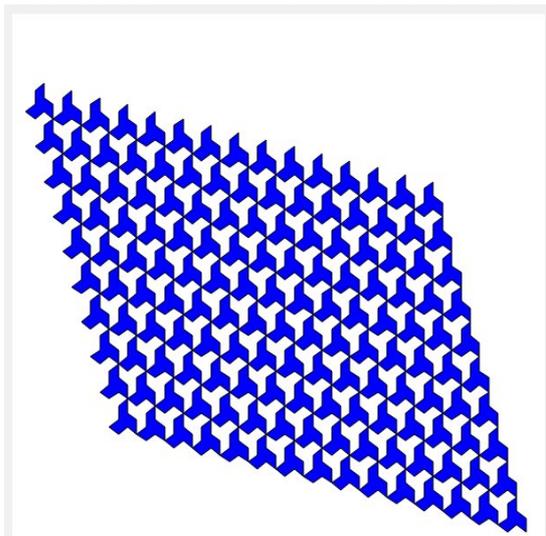
最初にボックスで差分分を計算している。

```
MMOPARAM=: 10 5;6 _1;_2 5 NB. for automatic
          orign xdif ydif
```

```
3 4 mk_diff_sub0 MMOPARAM
+-----+-----+-----+-----+
|_4 10|2 9 |8 8  |14 7 |
+-----+-----+-----+-----+
|_2 5 |4 4 |10 3 |16 2 |
+-----+-----+-----+-----+
|0 0  |6 _1|12 _2|18 _3|
+-----+-----+-----+-----+
```

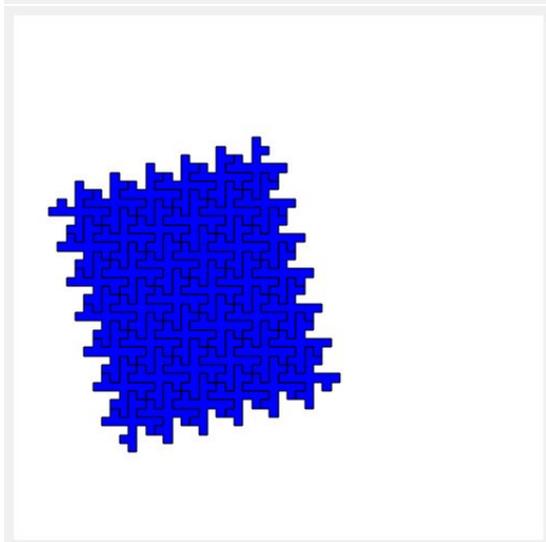
2.4 型紙の成型

ボックスの数字のコンテナはうまく積みあがるが、図形にすると差分分が左上りや右上りの平行四辺形になる。型紙のようにするのに波打つように戻す方法もあるが、中心部を切り出すこととする。



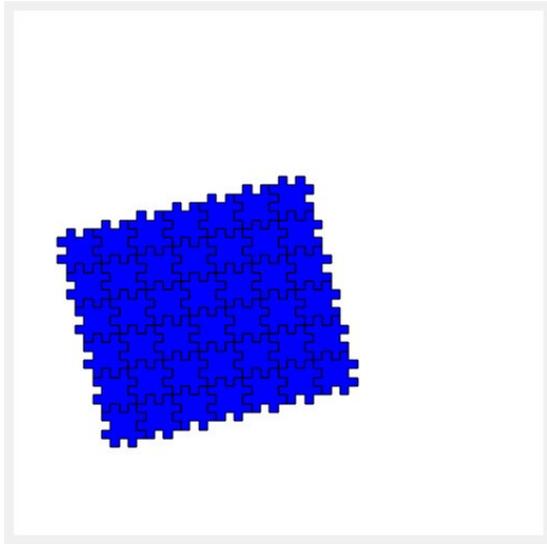
左上がりの四辺形

		ymax
		xmax
xmin		
ymin		



右上がりの四辺形

ymax		xmax
xmin		ymin



右上がりの四辺形

xmax(min)		ymax(min)
ymin(max)		xmin(max)

フィルムのサイズを割り出し *dwin* に与える。取り出すボックスは次のようになる。

- 右か左かは手動とする。
- 指定の 2 点のボックスを取り出し、 $\min(x, y)$, $\max(x, y)$ の値を求める
小さい方を選ぶ $10 < .15$
- 右上がりの場合

NB. `maked 2016/05/21`

```
maxxy_Lpop=: 0 1 &{@;@maxmin@;@{:@ {.
```

```
minxy_Lpop=: 2 3 &{@;@maxmin@;@{.@ {:
```

```
range_Lpop0=: min_Lpop,max_Lpop
```

```
range_Lpop=: 3 : 0
```

NB. `mk square`

```
tmp0=. range_Lpop0 y
```

```
dist=. minxy tmp0
```

```
(2{. tmp0),(2{.tmp0) + dist
```

```
)
```

```
minxy=:<./@(-/)@:|.@(2 2 &$)
```

- 左上がりの場合

```
NB. 2016/05/21
```

```
NB. find each 4 points
```

```
maxy_Rpop=: 1&{@;@maxmin@;@{.@{. NB. top left
```

```
maxx_Rpop=: 0&{@;@maxmin@;@{:@{. NB. topright
```

```
minx_Rpop=: 0&{@;@maxmin@;@{.@{: NB. last-left
```

```
miny_Rpop=: 1&{@;@maxmin@;@{:@{: NB. last right
```

```
range_Rpop0=: >@(minx_Rpop;miny_Rpop;maxx_Rpop;maxy_Rpop)
```

```
range_Rpop=: 3 : 0
```

```
NB. mk square
```

```
tmp0=. range_Rpop0 y
```

```
dist=. minxy tmp0
```

```
(2{. tmp0),(2{.tmp0) + dist
```

```
)
```

- *find_center* は使わなくともよい。range_Lpop で足りる