Mathematical model of stochastic algorithm in digital printing

Dr. Vilko Žiljak, Faculty of Graphic Arts, Zagreb, Croatia
Klaudio Pap, MSc, Faculty of Graphic Arts, Zagreb, Croatia

Keywords: stochastic curves, digital printing, PostScript

Abstract

Generating models of the original stohastic curves are being explored, and their impact on forming the security papers. Differences between PostScript routines are compared regarding limitations of today digital techniques. Discussion is spread on the programming and control of the stohastic vibrations in digital printing. This investigation brings new PostScript routines for versatile halftone screen patterns, as well as suggestions to the development of individual halftone screen shapes.

1. Introduction

Stohastic solution to the individualisation of a print is a marginal part of the digital printing. Computer graphics enable program intervention during printing. First of all, on each printed sheet a new number is feasible, another image or text prepared in separate files. Secondly, individualisation can be generated by an algorithm (usually used in numbering), meaning that data, which does not exist in a computer memory but is created logically according to a program written in C, Pascal or PostScript, is being printed. The highest individualisation levels are random picked line parameters and curves in enclosed group of possible movements or random picked halftone screen elements independently for each pixel.

2. Methods of work

Experiments are based on program solution of assigning random halftone screen shapes to each pixel separately. In order to achieve this, this is created :

1. mathematical models that produce exact shapes of halftone screen elements (Figure 1 - a, b, c, d),
2. procedure which will assign different screen shape and angle to each pixel and
3. PostScript programs which totally realise new solutions to a print individualisation.

This is about pseudorandom numbers with known seed and procedure of generating. That is why it is possible to repeat a series of random numbers, and realisation of a stohastic screened picture. Defining a seed is advised be to putted in function of firm data in the file. For example, if we are dealing with personal document with a photo, a seed can be generated from following data: date of birth and address. Repeating and authenticate a photo could be done only be a person who has full algorithm for a random variable generator. Original PostScript programs and procedures are given as a base to possible further research on stohastic selection of a screen type, frequency and angle on the unique picture.

3. Mathematical models of halftone screen elements

Our basic group contains fifty original shape models of halftone screen elements. Single colour portrait photos on documents usually have 150 kB of memory, so the same screen shape is repeated on an average 3000 times with different level of gray, meaning it appears with a different shape when printing. Figures 1a, 1b, 1c and 1d show 4 different models, each shown in 12 different levels of gray. Associated mathematical relations and PostScript commands are:

Brighter and darker pixels of the same model are different in a great deal, so one could get the impression of halfton screen elements to be originate from different mathematical relations (exclusively present on a model 1b and 1c).

Model 1 (1) $f(x,y) = 1 - \left|(\sqrt{|x|} - y) / 2\right|$

```
{neg exch abs sqrt add 2 div abs 1 exch sub} setscreen
```

Model 2 (2) $f(x,y) = 1 - \left|(\ln |x + 1.1| - y) / 3\right|$

```
{neg exch 1.1 add abs ln add 3 div abs 1 exch sub} setscreen
```

Model 3 (3) $f(x,y) = 1 - \left|(|x|\, 0.5 - |y|) / 2\right|$

```
{abs neg exch abs 0.5 mul add 2 div abs 1 exch sub} setscreen
```

Model 4 (4) $f(x,y) = 1 - \left|(|x| - |y|) / 2\right|$

```
{abs neg exch abs add 2 div abs 1 exch sub} setscreen
```

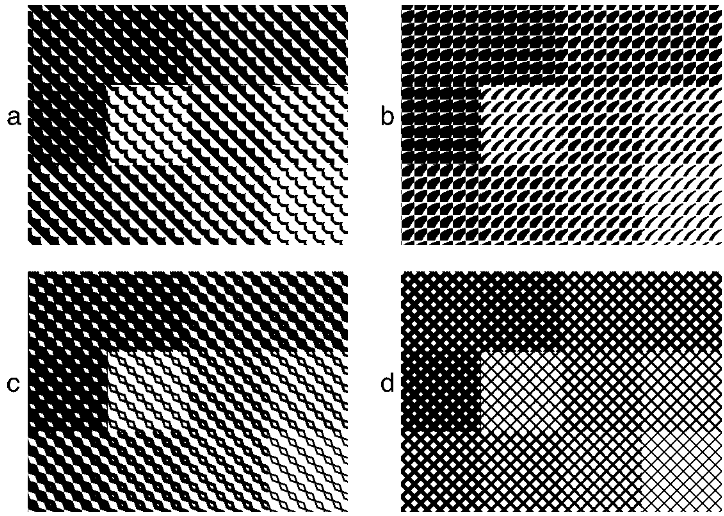

Figure 1 - New shapes of halftone screen elements

```
gsave 10 10 translate
/font1 {/FSHelvetica findfont 10 scalefont setfont} def
/squares <2E2952501EA07099506680B8> def
/r1 {neg exch abs sqrt add 2 div abs 1 exch sub} bind def
/r2 {neg exch 1.1 add abs ln add 3 div abs 1 exch sub} bind def
/r3 {abs neg exch abs 0.5 mul add 2 div abs 1 exch sub} bind def
/r4 {abs neg exch abs add 2 div abs 1 exch sub} bind def
/W 120 def /H 90 def /dx 20 def /dy 10 def

%Figure 1a
0 H dy add translate
font1 -8 H 2 div moveto (a) show
gsave 15 45 {r1} setscreen
W H scale
4 3 8 [4 0 0 3 neg 0 3]
{squares} image
grestore

%Figure 1b
W dx add 0 translate
font1 -8 H 2 div moveto (b) show
gsave 15 90 {r2} setscreen
W H scale
4 3 8 [4 0 0 3 neg 0 3]
{squares} image
grestore

%Figure 1c
W neg dx neg add
H neg dy neg add translate
font1 -8 H 2 div moveto (c) show
gsave 15 45 {r3} setscreen
W H scale
4 3 8 [4 0 0 3 neg 0 3]
{squares} image
grestore

%Figure 1d
W dx add 0 translate
font1 -8 H 2 div moveto (d) show
gsave 15 90 {r4} setscreen
W H scale
4 3 8 [4 0 0 3 neg 0 3]
{squares} image
grestore grestore showpage
```

Figure 2 - Source PostScript program for Figure 1

4. Experiments of stohastic solution to halftone screen, ruling and angle

A portrait on figure 3 is solved with 4 halftone screen models (Figure 1). Pixel is enlarged, the number of pixels reduced, and screen ruling also reduced because of better presentation. Four solutions are compared. Standard halftone screen with circle dots and bigger screen ruling (150 dpi) is used as a control pattern (Figure 3a). Total control of the right way or working different screens, that are repeated one after another gives "shaped moiré" with a period of multiplication product in the size of a pixel in the number of different screen models in program (Figure 3b). Moiré disappears if a selection of screen model is done randomly. Except this, screen ruling can also be randomly picked (Figure 3c), as well as the screen ruling together with screen rotation angle (Figure 3d).

This article encounters the original source of routine and procedure continuity in PostScript. Figures 4a, 4b, 4c, and 4d correspond to figures 3a, 3b, 3c and 3d. Change of a seed value (Figure 4c and 4d) will give totally new order of halftone screen solutions. PostScript solution also includes data of a picture, which otherwise in an operating program of the mass usage have to be taken from an external file (series "Grga" from a program on Figure 4a).

Figure 3b uses the same pixel continuity, but in the way of associating each pixel with a different screen algorithm picked out of the algorithm continuity given on figure 1. Program solution becomes denuded, but it is created in a way which can make each researcher interested in it, continue the development of his own algorithms, procedures and finally apply it in specialised usages .

The third solution includes a random variable generator to choose algorithm of a shape of halftone screen element, with screening angle α equals 0° (Figure 3c and 4c). The seed is set to be fixed (SEED = 12345678). Program shown on Figure 4d generates a contents of Figure 3d, but with a different seed (SEED = 246135). It is also important to say that in this example a random selection of screen α angle is embedded. A new plan of associating the shape of halftone screen element and screen angle of the same pixel continuity, (that is the same picture) has resulted.

a)

b)

c)

d)

Figure 3. a) standard dot halftone screen
b) screening with cyclic continuity of screening algorithm
c) screening with stohastic selection of halftone screen element shape algorithm (SEED = 12345678) i $\alpha = 0^\circ$
d) screening with random selection of halftone screen element algorithm and random selection of screening angle α *(SEED = 246135)*

a)

```
/grga <
2B372A52D6D04A005C4F16232522242521151151E25221F201F1D28362A8CE3D17C100E3316181C1519180C21312A111A20191B1D1A
223441C3C3B07841002918OE0D201A1442757A774F222C1A1B1E1811162B4818110D0E110F0E09034B745B5F4E3B4A78722A211C1916
0E240F020A0E0C0B130F1A2310284C342E2E1E182D6252241A1919155E29050C0E0C0C0F101027271E151114110B151013171B191718
0C0F0B0B0D0C0A0E1B23180A0B0F0B0A0D0F121C1A1C13191915140E0B0A0C0B0A070C131B12100E10110D0F161A1E2729221E1D2413
0E0C0A0C0A0D19080D11121F162027171010121D232027272B38270D0C0B0C062F5D0A0C11181E2524200E100F131D241B1B2633303A
0D0D0C0D0A1530330B161E45564937302216323432B2E212C383D410D0D0C0D0E0612221A2B37646E686D816458695E634C363A48473C
090F0F0D0D0B0903394F5671707A93ADB4B5AC98997C646F6C595E386320080E0B0A083A515A6B70859FBED0DADAD2CBB7B2B0996B6C
304F1E0A0E0C0B07264564556186ACC8DCE7E8E5DAD6D2C1A37760064554130B0D06061A375A4B5B8FB2C8D8E2E8EADDD6CEBD9E693B
0C17230E0B0E23061230545D648FB1C4CFDAE3E4D9D2CCBA9B5C300E0D0B0A0A1F5640202B63736B8EAAA9ACC4D1D3CAC6BAAA9C694D
0E0F0D0A0C53528D7A31698E9796705A7EA7C2CAB7957D7A83746A0E0E0D0C0B0C55A0813E71ABB088560072A4BDCE99520B44628477
0501000507011286BF5C61A6B5C2BFAAB5AFB1B8919D7197A8B68639656833144A083B935A52A3BBC7CBCDC8A6A2BC97AEC6C7CABB85
E5F3F4E2B6931A05586B4A9AB6CEDCE0D79DB6C4A5B4DACFC8AC81EBEBEBEDF06E00061C433C89B0CEDDD8CCBDBDB2ABBFE1DBBE9566
ECE2D2C0DCA50C0007032079A1B6B193B2C9D8DBC5AFBDC9A86B57E5C08280B0BD854005030B499DAC9B694B64666A596492A27E483E
E7DCB97FB2D0EBEDBD49000B51A3B2A2404E76483483A6926B2A07DFC3A3B8CBD9E2EEFAED7905073282BD9C7B8483A3B4AA7E32080E
A09F91928BC3D3E9F3F0F9890804104BBBCCC4C2B3824C1C080B0F91516287A6CDECDFDEE8F0F57B060600318BB3AB702606080C0D0C
5E62777F96DDFAEEE8E6E9F1EF750A08050A1E20315E0E0A0F0B23436C5F527DBAE5F4F9FFFCEEDFE683040A0906040F5711090B0FA7
544F49585B4067838BA8C4EAF2E0B51307090807031F100B0818D05B6097A37332395155514C5988D7D72B040B0B0B0A0B0D0E080EAF
3651999DC3781C2D415D523D37458132010A0B0D0B0A0B0A0A004456606D80471807030F7A886E565B534E1E050A0D0B0A090B0A0143
2A2E33240405090802218366575458726C21070C0B0B0A0A0B00510201000308OA0909090034C04286987260781C090C0C0B0A0B0523
0807080908090909090908056C9BBB85897C46210B0B0B090B0A05180809080708080808090A090625728E7388605F1E090A090B0B1B55
> def
162 240 scale 27 40 8 [27 0 0 40 neg 0 40]
{grga} image showpage
```

b)

```
/ros [ {r1} {r2} {r3} {r4} ] def
/L [ 10 15 20 25 ] def
/K [ 0 15 45 75 ] def
/m 0 def
/pixl 6 def
/dpixl 1 pixl div def
gsave 0 39 pixl mul translate
0 1 39 {/k exch def /i k 27 mul def
gsave
  i 1 i 26 add {/j exch def
  /LIN L m get def /KUT L m get def
  LIN KUT ros m get  bind setscreen
  1 1 8 [dpixl 0 0 dpixl 0 0]
  {grga j 1 getinterval} image
  pixl 0 translate /m m 1 add def
  m 3 gt {/m 0 def} if
  } for
grestore 0 pixl neg translate
} for showpage
```

Fig. 4. a) program for standard dot halftone screen
b) program with cyclic continuity of screening algorithm

c)

```
/ros [ {r1} {r2} {r3} {r4} ] def
12345678 srand
/m {2 31 exp 1 sub } def /rn {rand m div} def
/L  {10 20 rn mul add } def %sluc broj [10,30]
/R {3 rn mul} def             %real sluc broj [0,3]
/IR {R  round cvi } def       %cjeli sluc broj [0,3]
/pixl 6 def /dpixl 1 pixl div def
gsave 0 39 pixl mul translate
0 1 39 {/k exch def /i k 27 mul def
gsave
  i 1 i 26 add {/j exch def
  L 0 ros IR  get  bind setscreen
  1 1 8 [dpixl 0 0 dpixl 0 0]
  {grga j 1 getinterval} image
  pixl 0 translate
  } for
grestore 0 pixl neg translate
} for showpage
```

d)

```
/ros [ {r1} {r2} {r3} {r4} ] def
246135 srand
/m {2 31 exp 1 sub } def /rn {rand m div} def
/L  {10 20 rn mul add } def %sluc broj [10,30]
/K  {0 90 rn mul add } def  %sluc broj [0,90]
/R {3 rn mul} def             %real sluc broj [0,3]
/IR {R  round cvi } def       %cjeli sluc broj [0,3]
/pixl 6 def /dpixl 1 pixl div def
gsave 0 39 pixl mul translate
0 1 39 {/k exch def /i k 27 mul def
gsave
  i 1 i 26 add {/j exch def
  L K ros IR  get  bind setscreen
  1 1 8 [dpixl 0 0 dpixl 0 0]
  {grga j 1 getinterval} image
  pixl 0 translate
  } for
grestore 0 pixl neg translate
} for showpage
```

Fig. 4. c) program with stohastic selection of shape of halftone screen element algorithm (SEED = 12345678) i $\alpha = 0°$

c) program with random selection of screening element algorithm and random selection of screening angle α (SEED= 246135)

5. Conclusion

In documents and post stamps production with high degree of protection and uniformity, introduction of stohastc individualsation on a pixel screening level is recommended. This solution is possible only on digital graphic prepresses and digital printing directed by PostScript. Recommended solution point out the important of studying the computer graphics and programming.

Stohastic screening models successfully solve the protection of print authenticity, therefore it shall be found everywhere, where it is a subject of high importance. In security program we suggest the shapes of halftone screen element which contain lines and curves (shapes on Figure 1c and 1d). First experiences were on simple trigonometric functions (sin, cos). More sophisticated shapes (Figure 2) provide authenticity and characteristics of the printing house that produces this kind of securities.

Researches with multihalftonecreen shapes and stohastc selection of screen ruling and angle are extended to fullcolour originals. This area has specific usage when solving portraits and stamps. Vibrations, caused by different screening of the same pixel provide unrepeatabillity and strong protection against all scanning techniques.

Literature:

Roth S. F.
1988 "Real World PostScript", Addison-Wesley, ISBN 0-201-06663-7

Pap K. and Žiljak V.
1998 "Stohastic Change of Bezier Curve in PostScript for the Security papers", 5th International Design Conference, Product Design Development & Design Education, Dubrovnik, UDK 7.05 (063), ISBN 953-6313-20-0

Žiljak V. and Pap K.
1999 "PostScript" (book), Wiena-Austria & Zagreb-Croatia