UDC 004.92:519.216:655
ORIGINAL SCIENTIFIC PAPER
RECEIVED: 22-08-2011
ACCEPTED: 22-11-2011

ACTA GRAPHICA 196

Individual Stochastic Screening for the Development of Computer Graphics

Authors

Maja Turčić[1]*, Vilko Žiljak[2], Ivana Ž.–Stanimirović[2]

[1]*Polytechnic of Zagreb,*
Zagreb, Croatia
**E-mail: mturcic@tvz.hr*

[2]*University of Zagreb,*
Faculty of Graphic Arts,
Croatia

Abstract:

With the emergence of new tools and media, art and design have developed into digital computer-generated works. This article presents a sequence of creating art graphics because their original authors have not published the procedures. The goal is to discover the mathematics of an image and the programming *libretto* with the purpose of organizing a structural base of computer graphics. We will elaborate the procedures used to produce graphics known throughout the history of art, but that are nowadays also found in design and security graphics. The results are closely related graphics obtained by changing parameters that initiate them. The aim is to control the graphics, i.e. to use controlled stochastic to achieve desired solutions. Since the artists from the past have never published the procedures of screening methods, their ideas have remained "only" the works of art. In this article we will present the development of the algorithm that, more or less successfully, simulates those screening solutions. It has been proven that mathematically defined graphical elements serve as screening elements. New technological and mathematical solutions are introduced in the reproduction with individual screening elements to be used in printing.

Keywords:

Stochastic Models, Computer Art Graphics, Screen Shapes, Security Graphics, Design and Print

1. Introduction

Computer technology "sneaked" into human lives in 1960s as a luxury intended for individuals and corporations and not as a tool for general use and expression. Owing to certain people who had opportunity to experiment, various branches of computer art and design have been developed. Computer graphics are responsible for changes in the theory of art, education, entertainment and environmental design (*Franke, 1985*). Computer art was a logical step in developing the means of graphic expression because it offered solutions not achievable by means of analogue methods. A computer has become a new tool that is constantly changing creative processes and completely renewing the concept of artistic expression (*Nierhoff-Wielk, 2007*). Computer graphics tools have created the terms of vector and pixel graphics that present the basis of contemporary printing industry in processing text and images.

Computer graphics do not indicate one author or their style, since programming in various computer languages can result in the same visual work coming from various authors. Each work of art or design is an original, i.e. individual, under the assumption that each image or design is a reflection of individuality and uniqueness of the author and their creative thinking process, i.e. that the author's unique style is recognisable in each image.

Although individuals of the art movement *Nove tendencije* (New Tendencies) fought with numerous critiques of their visual experiments, their ideas have become the foundations for generating new and progressive ideas that are nowadays considered to be common. From the basic computer graphics whose creation, due to slow computers and inadequate computer languages, interface or printers, used to take days, a need for animation, i.e. moving image arose. The integration of the element of time into graphics programming and the combining of more images gave impetus to new ideas and media. In mid 1970s, Tomislav Mikulić could draw only static monochrome lines using a graphical terminal, but with a camera and a film he developed and created animation for TV shows (*Mikulić, 1980*). Soon integration of commuter graphics into movies, cartoons and commercials took place. Nowadays, owing to the wide availability of personal computers and programmes containing routines enabling easier use and expression, animation and 3-D graphics are not a problem to create. However, 3-D graphics are also based on the same or very similar algorithms of two-dimensional graphics. The 3-D technology has become a standard; it has culminated in the film industry and is now available at home as well.

In the same way, due to the availability of personal computers and software, the satiation of ideas has occurred. It is inevitable for those working with programming graphics to use the already existing algorithms that have been proved the most effective and the simplest, for instance circles, sine curves, Bézier curves and similar. The difference lies is the unique procedure for creating forms.

Nowadays, all aspects of graphic industry: layouting and editing, the entire DTP procedures, reprophotography, the preparation and production of printing plates are based on computers, computer software and computer tools. Tools currently used in graphic industry such as *Adobe Photoshop, InDesign* and *Illustrator* have developed from the procedures and experiments of artists/ scientists from 1960s.

2. Nove Tendencije (New Tendencies)

In 1960s, the members of the art movement *New Tendencies* started creating algorithmic works of art, thus becoming the pioneers of such art in Croatia. First they made their works using analogue methods, but as technology developed, they themselves started using machines and programme languages to write in. They blended art and science and introduced the concept of research into their works (*Rosen, 2011*). At that point the viewer could not longer tell the difference between an artist and a scientist. Those works of art contain in themselves recognisable routines that detach them from personality and individuality. The first exhibition was held in 1961 in Zagreb showing experimental works of artists/scientists that experiment with the visual and participate in the creation of a new dominant art movement. New tendencies continued into early 1970s, during which they showed efforts to systematise and study the possibilities, thus making their art primarily different from mere "attempts". Experiments are defined through the methodology and planning of conditions that are then studied. They resulted in repeatable works. This hypothesis is opposed to the traditional concept of an artist as a spontaneous genius expressing themselves through art (*Moles, 1968*).

They build algorithms through systemic research and create functionality by defining boarders (*Klutsch, 2005*). The concept fell apart in 1970 and one of the participants, Frieder Nake, concluded that computer graphics production was repeatable and that new, "really good ideas haven't shown up for a while" *(Nake, 1970)*. On the other hand, the fact is that the entire graphic industry is based on computer graphics; it has changed the industry, resulting in new technologies such as digital printing.

In this article we will present different ways of possible program development providing the same visual results although coming from different authors and programming languages, if that program logics is studied and understood. We create new algorithms resulting in similar solutions to those of other authors in history aimed at determining algorithms and their upgrades.

Many artists have in various ways tried to arrange screening elements and their coverage (*Vladimir Bonačić-1969, Frieder Nake-1965, Georg Nees-1965-1968, A. Michael Noll-1965, Manfred Robert Schroder-1968, Leon D. Harmon/Kenneth C. Knowlton-1966, David R. Garrison-1968/69, Vilko Žiljak-1972*) (*Rosen, 2011*). Technologies of phototypesetting from 1980s had relations in which an entire word could be used as a screening microelement (*Žiljak, 1987*). Artists used to solve the problem with instances of individual expression and without publishing procedures or algorithms for the suggested technique. They did not go beyond the mere presentation of their work of art.

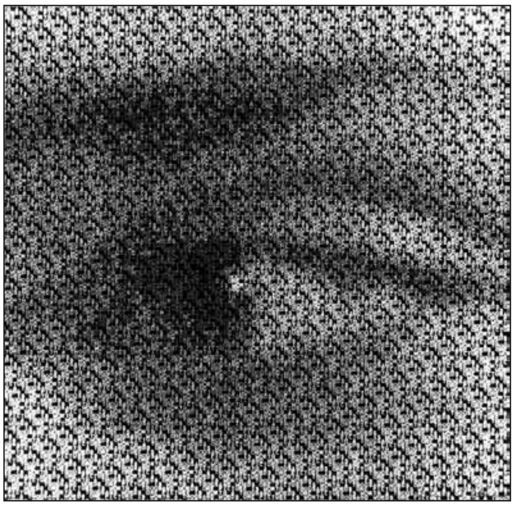

Figure 1. Manfred Robert Schroder, Eye II, 1968

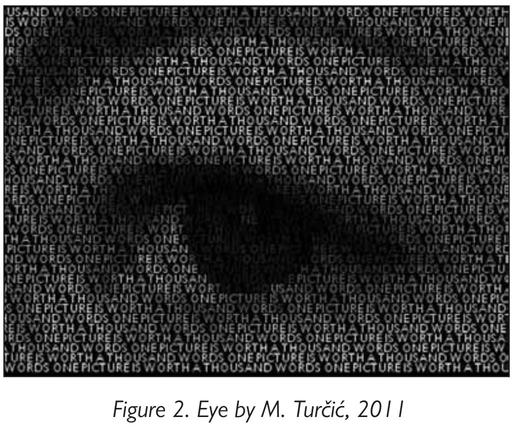

Figure 2. Eye by M. Turčić, 2011

```
PImage img;
int charCounter=0;
String s = "ONE PICTURE IS WORTH A THOUSAND WORDS ";
void draw()
{
  noLoop();
  img.loadPixels();
  int x=0;
  int y=0;
  int rowCounter=0;
  while(y<height)
  {
    while(x<width)
    {
        char currentChar=s.charAt(charCounter);

        color col=img.pixels[y*width+x];
        fill(col);
        text(currentChar,x,y);
        x+=textWidth(currentChar);
        charCounter++;
        if(charCounter>=s.length()) charCounter=0;
    }
    charCounter=0;
    rowCounter++;
    y+=8;
    x=-rowCounter*360;
  }
}
```

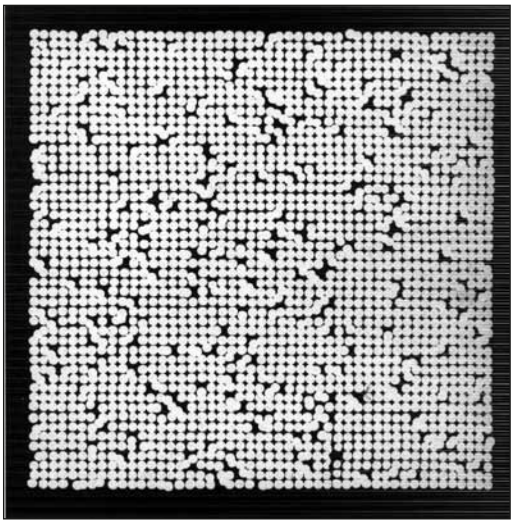

Figure 3. Gotthart Muller 64/6, 1964

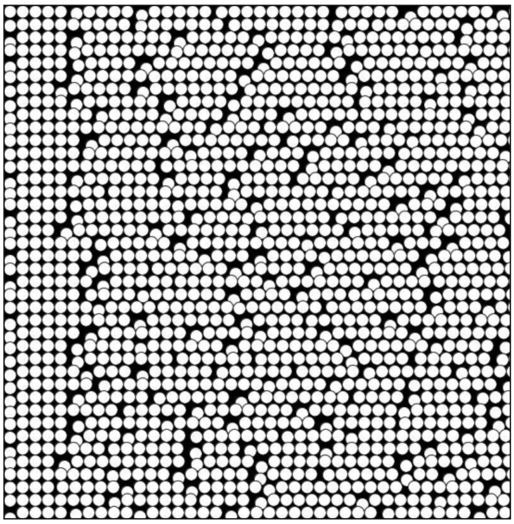

Figure 4. Version by M.Turčić, 2011

3. Individuality and Development

Computer graphics bring about the loss of individuality due to the lack of recognizability of an individual author's style, which is particularly noticeable in its fractal structure. The term 'fractal' was first introduced by Benoit Mandelbrot in 1975 to represent complex structures based on a simple recursion, i.e. repetition. Although each graphic is unique, it does not express individuality because of the possibility that the same result is programmed by another author (*Čerić, 2008*).

However, computer graphics as we know them today and that have been developing since 1960s, prove that an image does not represent an author, because it can be generated and is therefore repeatable. If an author uses fractals or some known algorithms or recognises routines in somebody's work of art, even if they add their ideas to the original setting, they are not an original generator of that art or design. In that way we question the uniqueness of graphic design that is derived from routines and subroutines of computer software and reaches the solutions in the same way. Due to the possibility of repeating a work of art, methods to generate computer graphics destroy the ancient presumption of the recognizability of an author.

After forty years of developing methods and programming languages in computer graphics, the methods have become standardised. The best example of repetition are fractal graphics that have been named (Mandelbrot, Koch, Nova, Julia set, Brown's tree etc.) and in that way standardised and made ready for use by designers and artists. Fractal graphics have a pseudo-stochastic generator making a "stochastic image" with set borders within which it widens, develops, i.e. applies anti-stochastic principles. The problem of Moire was resolved in the printing industry by introducing frequency modular screens. That is a symbiosis of the coverage of an individual pixel through a random disposition of microelements of the coverage of the same pixel *(Žiljak & Pap, 1999*).

Mass application of computer graphics is a consequence of the lack of originality, either voluntary or unconscious. Tools, nowadays used in the applied computer graphics technology, create units and become routines used to create individual solutions for requirements of the advertising industry. If a new graphic solution is created, it evokes the need to create new solutions and that way the hypothesis of exponential development of computer graphics in the world is confirmed. The ultimate result of these processes is the anonymity of authors because individuals will always have avant-garde ideas and their realisation will inspire others to similar processes (either by stealing or by their extension), eventually becoming subroutines, effects and tools in the contemporary graphics software.

The aim is to create one's own base of routines that are upgraded to constantly create new tools, i.e. packages of routines for graphics which are to be individualised. The aftermath of our base is a detachment from standardised work conditions indicating close relations of science and art in the methodology leading to individual digital reproductions.

4. Stochastic and anti-stochastic

Stochastics provides new solutions that are normally not possible due to the human *Gestalt* need for grouping elements. It generates unpredictable shapes, crucial for the vivacity of design, and presents the basis of imitating individuality. Stochastics functions as a simulator of certain processes and implies the planning and its opposites. Anti-stochastics introduces the ability to control, among other, the return into the original condition, thus creating "a flow" of generated elements functioning within them and preventing the work to disperse. Anti-stochastics brings order into distortions that are created by stochastics, imposing predictable borders within which stochastic distorted signs, shapes and elements are generated. In the analogue era, works used to inevitability are different each time due to human errors, but the digital technology has introduced sterile uniformity of products. At the same time, a computer is an ideal tool for generating stochastic variations, hence providing freedom and the richness of expression, impossible for digital solutions. The best examples are handwritten fonts that cannot look authentic without a stochastic variable. This was demonstrated by Donald E. Knuth in his *Metafont Project* that defined letters and signs by mathematical curves and by introducing stochastics, all curves were changed leaving the impression of authenticity (*Knuth, 1986*).

Stochastics in computer programming is based on random number generators. Some programming languages themselves generate parameters of congruence, and in some there is not even an option for the author to define seed, i.e. the seed of generator. But in our attempt to achieve the utter control of a random sequence, we have built our own generators that are congruent due to their speed and simplicity, as well as high quality results. They are based on finding the remainder of the product of the two preceding numbers. Particular attention should be paid to the degeneration, i.e. too frequent repetition of sequences or the transition to one constant. Therefore, the best results are obtained if numbers 'a' and 'm' are prime numbers (*Žiljak & Smiljanić, 1980*).

$$r_{i+1} = (ar_i + b) \bmod m \qquad (1)$$

Congruent sequences always end in a loop; there is always a sequence that is repeated endlessly. If b = 0, that sequence is shorter, but it is still possible to achieve that the sequence is long enough. The sequence cannot be longer than the amount of value m. The best result, i.e. the length of sequence, is obtained by using prime numbers (*Knuth, 1997*).

Figure 5. a) portrait font, the original version and b) a version processed by algorithm

Using the available generator within a programming language (random, math. random, rand...) the same sequence of numbers is always obtained because of their pseudo character. In the attempt to get diverse results, except for control, independent generators are needed to create the rhythm and add new interventions into the existing graphics. In computer graphics, random number generators are used for different purposes (for colours, dimensions, angles…) that do not have the same occurrence period; and that period can also be random. In order to control the parameters of newly introduced segments of programs, we need to ensure the same sequences at the previous pseudo random sequences. The only way to obtain completely controlled solutions is introducing a few independent pseudo random generators. All our generators have a range from 0 to 1, but each one has a different initiator.

The numbers in Table 1 leads to the conclusion that the length of the sequence obtained by a congruent method depends on the choice of numbers. Better results are achieved using prime numbers (especially for 'a' and 'm'); however that is not the only factor determining a generator's efficiency.

An individual test of pseudorandom sequence in the Processing language:

```
// congruent method of
generating random numbers
float a = 2311;
float m = 914189; //the
biggest number
float s = 913373; //sjeme
int i;
float rn;
float f;
//for the range
float x1 = -5;
float x2 = 5;
for (i=1; i<50; i++)
{
 s = (a*s) %m;
 //the range from 0 to 1
rn= s/m;
f = rn * (x2-x1)+x1;
println(f);
}
```

Table 1: examples of generators and their efficiency in relation to the selected numbers

a	S	m	Sequence length	%
7	13	23	10	43
37	19	23	22	95
7	13	24	3	12
2311	41077	96553	32184	33
2311	913373	914189	228546	24
2311	41077	96558	66	0,1
2311	101	37313	9328	25
67	101	193	32	16
401	101	193	192	99
5431	3373	7547	7546	99
5431	3374	7547	5071	67
5431	3373	7542	1254	16
9	25	31	15	48
9	31	73	6	8
9	277	439	73	16
9	53	439	73	16
9	53	331	165	50
227	53	331	330	99
227	56	331	330	99

The backbone of this animation are letters from the *Streetvertizing* font (http://www.cleanvertising.com/) and *Processing library Geomerative* (http://www.generative-gestaltung.de), used to transfer font characters into a certain number of curves, i.e. segments. Afterwards, the end points of these curves are used as starting points of new graphic elements, in this instance lines. The target (finishing point) of these lines is determined using an own random number generator and by each pass through the loop, the letter starts to be drawn again, but the number range that the generator uses is in each pass bigger, resulting thus in the greater length of lines and bigger angles compared to the starting point. Two independent generators are used, one for the x and the other for the y axis. The animation shows alterations of one parameter in this graphic, and that is the range of numbers of the random number generator.

These algorithms for generating graphic elements are targeted at the new area of defining screening technology. The example of a lower case letter 'a' shows coverage with microelements in the shape of lines. New screening elements can already result from the graphic itself and do not have to be created from screening cells. The example showing solutions to the glyph already proves that there is a new field of screening technology. Dissipation and multiplication of these screening elements resolves the problem of Moire in its own way, if the same letter were interpreted in multicoloured print. In order to achieve, for instance, a brown hue of the graphic 6th letter, a 6/6 would preset a yellow channel, letter a 6/4 a magenta channel, letter a 6/3 a cyan channel. Channel K would not be needed. The same principle is used in the spot multicolour system pairing each colour with its own solution. The length of pseudo random generator determining the coverage, and the seed is different for various channels in order to prevent the repetition of the same graphic positions in various channels. Similarly to the way in which these examples generate a screening element as a part of a straight line, this discussion can be expanded to any other graphic element, such as a square, an ellipsis, an arc, a circle or a *Bézier* curve.

```
// Animation of letters thorough segmentation and own random
number generator in Processing
void draw() {
  noFill();
  pushMatrix();
  strokeWeight(0.1);
  stroke(75,0,130);
  beginShape(LINES);
    for (int i = 1; i < pnts.length; i++ ) {
      seed = (mul * seed ) % modul;
      ran = (seed /div)*(minus)-(minus/2);
      seed2 = (mul * seed2 ) % modul;
      ran2 = (seed2 /div)*(minus)-(minus/2);
      vertex(pnts[i-1].x, pnts[i-1].y);
      vertex(pnts[i].x + ran, pnts[i].y + ran2);
    }
  endShape(CLOSE);
  popMatrix();
  div-=1000;
  minus+=1;
  }
```

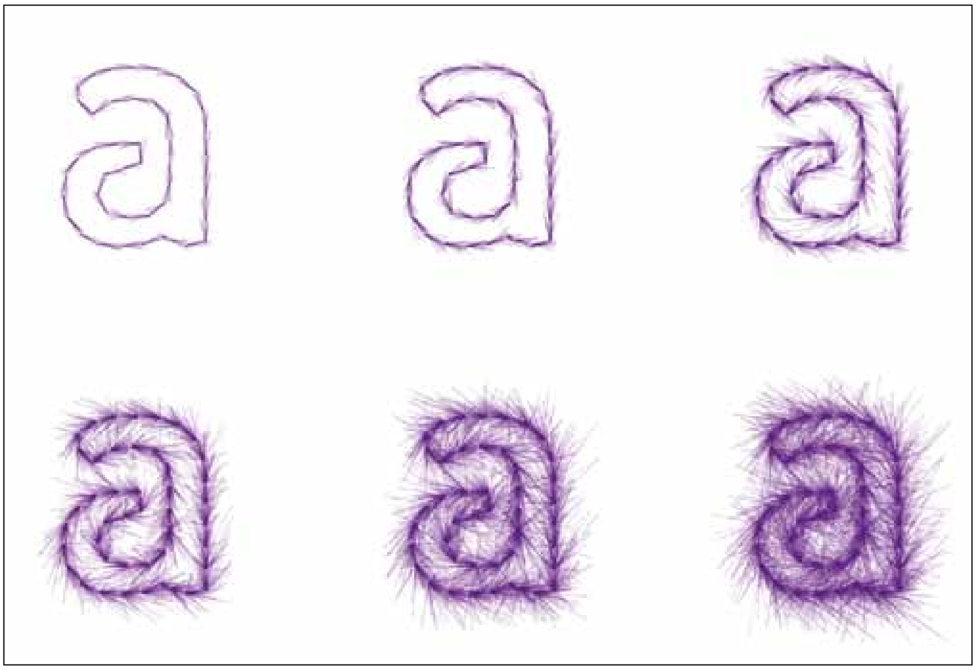

Figure 6. random animation of letter a

Graphic technology is based on solving coverage through various mathematical screening methods. The artists from the age of the earliest computer graphics were the first to come across the problem. The creation of new screening elements for solving coverage is based on computer graphics developed in 1960s. After the use of cameras in reprographics (and net screening) was abandoned, all technology of solving coverage and screening problems was derived from computer graphics. The focus was placed on finding new solutions in design and solving problems of graphic reproduction by means of computer methods.

5. Conclusion

in its beginnings, computer graphics were based on geometrical and mathematical forms and were therefore losing support of critics and audiences. Those beginnings provoked the development of unknown spheres of graphically attractive forms, i.e. new aesthetic area that does not even have to be named art. The way of thinking and human behaviour has also changed. Programming enables the obtaining of a series of images by changing parameters that can lead to animations. New methods of expression such as music or words can be generated by computers and in that way they expand the forms of expression. The acceptance of new methods does not threaten the "classical" art techniques, but it develops new functions of stimulating creative capacities and sensitizing viewpoints. In comparison with traditional techniques, computer offers a different approach and generates results emphasizing cognitive processes, creativity and imagination rather than physical skills and talent. Printing technology has introduced mathematical definitions of screening elements. Authors of those programs have suggested many procedures and forms based on PostScript description of screening cells. We have presented a proposition of screening using simple mathematical forms. Screening solutions of early computer graphics were compared which no analogue mathematical interpretation was possible for. Preconditions for the systematisation of screening elements as a basis for new forms have been created.

References

Nierhoff-Wielk, B., 2007. Ex Machina - The Encounter of Computer and Art. A Look Back. In: W. Herzogenrath and B. Nierhoff-Wielk, eds. 2007. *Ex Machina - Early Computer Graphics up to 1979. The Franke collections and other foundations at the Kunsthalle Bremen. For the 80th birthday of Herber W. Franke 1979*, Bremen: Deutscher Kunstverlag, pp. 22-23, ISBN 978-3-422-06689-2

Klutsch,C., 2005. The Summer 1968 in London and Zagreb: Starting or End Point for Computer Art. In: C&C '05 Proceedings of the 5th conference on Creativity & cognition, London, United Kingdom 12-15 April 2005, NewYork: ACM, ISBN 1-59593-025-6, Available at: <http://www.computerkunst.org/Kluetsch_London_Zagreb.pdf> [Accessed 3 August 2011]

Franke, W.F., 1985. The New Visual Age: The Influence of Computer Graphics on Art and Society, *Leonardo*, 18(2), pp 105-107, Cambridge: MIT Press, Available at: < http://dam.org/essays/franke-the-new-visual-age-1985> [Accessed 3 August 2011]

Čerić, V., 2008, Algorithmic Art: Technology, Mathematics and Art. In: invited lecture at the *Information Technology Interface ITI 2008 30th International Conference*, Cavtat, Dubrovnik 23-26 June 2008, pp 75-81, ISBN: 978-953-7138-12-7, Available at: <http://www.vceric.net/texts/mathmod09_algorithms_mathematics_and_art.pdf> [Accessed 3 August 2011]

Moles, A., 1968, Introduction, Kelemen, B. Putar, R. eds. *Bit Interantional* 3, International Colloquium on Computers and Visual Research, Zagreb August 3-4 1968, pp 3-10, Zagreb: Galerije grada Zagreba

Žiljak V. Smiljanić G., 1980, *Modeliranje i simuliranje sa računalom*, Elektrotehnički fakultet u Zagrebu, Liber

Žiljak, V.,1987. *Računarska Tipografija*, Zagreb: Školska Knjiga

Žiljak V., Pap K. 1999. Mathematical model of stochastic curves in digital printing, In: 26th International Research Conference of IARIGAI, Munchen, Germany september 5-8 1999.

Knuth, D.E. 1986. *The Metafont book*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., ISBN 0-201-13444-6

Knuth, D.E., 1997. *The Art of Computer Programming* 3rd ed Vol 2, pp 10-25, Reading, Massachusetts: Addison-Wesley, ISBN 0-201-89683-2

Rosen, M., 2011. The Art of Programming. In: M. Rosen ed. 2011. *A little known story about a movement, a magazine, and the computer's arrival in art: New Tendencies and Bit International, 1961-1973*, Germany: ZKM Publications, Center for art and media Karlsruhe, pp. 27-40, ISBN 978-0-262-51581-8

Mikulić, T., 1980, *Computer animation*, exibition at: Galerija Nova, centar za kulturnu djelatnost sso, zagreb, mihanovićeva 28/I, 16. june 1980. Available at: <http://tomislav-mikulic.com/1980-animacija-100.pdf> [Accessed 3 August 2011]

Nake, F., 1970, "Statement for PAGE" , *PAGE 8 bulletin of the computer arts society*, may 1970, Gustav Metzger ed., London Available at: <http://www.bbk.ac.uk/hosted/cache/archive/PAGE/PAGE8%20May%201970.pdf> [Accessed 3 August 2011]