WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

SBArt4 for an Automatic Evolutionary Art

Tatsuo Unemi
Soka University
Hachioji, Tokyo 192-8577, Japan
unemi@iss.soka.ac.jp
http://www.intlab.soka.ac.jp/~unemi/

***Abstract*—Owing to recent innovations in computer hardware, it has become possible to implement automated evolution to produce interesting abstract images based on computational aesthetic measures for fitness criteria within a feasible time using a personal computer. The present paper proposes extensions of aesthetic measures for the domain of evolutionary art, for both images and animations. Using two small computers connected via Ethernet, we realized an installation of automatic video art that continuously displays a series of new interesting animations on the screen by selecting individuals of higher fitness from an evolving population in real time. The art project also includes automated creation of animations that are accessible over the Internet. The project involves the automatic production of ten 20-second movies everyday and posting the digest movie to a popular web site for movie sharing.**

***Keywords*—video art, automatic art, real-time evolution, computational aesthetic measures.**

I. Introduction

Interest in beauty is an intrinsic psychological characteristics of human beings. One reason why the interactive evolutionary computation (IEC) [1], [2] has generally been applied to artistic productions is that because the evaluation measure of aesthetics is assumed to depend strongly on a subjective criterion for each individual person. Interactive evolutionary computation is a powerful framework for the efficient exploration of a huge search space to find better solutions that fit with the human-subjective criteria. A typical early challenge was the artificial evolution system implemented on a combination of a supercomputer and sixteen graphics workstations by K. Sims in 1991 [3]. A number of researchers and artists have attempted variations of encoding methods, applications for other areas, and improvement of human interface for convenience, for example.

On the other hand, similar to the case in which researchers in artificial intelligence have been struggling to realize functionalities of human intelligence on the machine, the artistic creativity of the machine has also been a target since the early days of computer science. One typical example is AARON, which was developed by H. Cohen. The first painting generated by AARON was presented in 1973 [4]. AARON is an expert system into which the knowledge of how to draw an art piece is embedded.

Computational aesthetic measures have also been examined as a useful technology for finding better images or analyzing the styles of masterpieces in the context of computer graphics and image processing [5]. These methods are also useful for calculating the fitness values in order to automatically select favorable individuals in evolutionary computation so as to obtain beautiful images. As one of the methods by which to reduce user fatigue associated with the IEC approach, a combination of automated evolution was examined by P. Machado *et al.* [6]. Several different types of measures were also examined for automated evolution by E. den Heijer *et al.* [7].

The present paper describes a new approach to embed an automated evolution into a breeding system for abstract images, SBART, developed by the author [8]. The first release in 1994 was runnable on UNIX workstations and was extended and migrated to MacOS 8, MacOS X on Power PC, and then MacOS X on Intel CPU. The recent extensions include a real-time breeding of animation using the power of parallel processing by GPU [9]. Similar to a number of other related studies, the system uses a functional expression as a genotype that calculates a color value for each pixel to paint a rectangular area. Several different types of measures borrowed and modified from previous studies combined with newly developed measures are introduced for fitness evaluation. In addition to measures for a still image, a criterion for animation is also examined for a new functionality in order to evolve a movie.

This extension enabled the installation of a type of automatic video art that continuously displays a series of new and interesting animations on the screen by selecting individuals of higher fitness from an evolving population in real time. The burden for a small computer to draw an animation in high resolution and to calculate the fitness value of each individual animation is heavy. By executing these two processes separately on two machines, smooth real-time playback is possible. The art project also includes the automated creation of animations that are accessible on the Internet. The project involves the automatic daily production of ten 20-second movies and posting of the digest movie to a popular web site for movie sharing.

The following sections describe the aesthetic measures used in this system, the method of generation alternation, the installation setups, and the organization of automatic net-based art project.

II. Aesthetic measure for still image

Aesthetic evaluation by an individual is fundamentally driven by the cognitive system of that individual, the functionality of which depends strongly on personal experience and cultural background. In order to implement such functions via computer, it is necessary to use some type of learning mechanisms to adapt to variations among individuals and to changes over time for an individual. Some researchers are trying to embed such adaptability using algorithms of artificial neural networks [6]. However, we avoid such an adaptive mechanism and consider only common criteria that a majority of people will agree on interestingness at the level of perception. Therefore, abstract painting styles, such as a painting with a single color over the entire area of the canvas, is beyond the scope of the present study, because knowledge related to the historical and cultural context is usually required in order to interpret the piece appropriately.

The following subsections describe three measures for spatial arrangement and three measures for color variation. In order to unify these measures in different dimensions for a final fitness value, a normalization method is introduced, as described in the final subsection.

A. Information theoretic complexity

The most extensively researched measure is information theoretic complexity. Intuitively speaking, simple uniform patterns, such as solid colors, should be eliminated from the candidates for interesting images. Theoretically, Kolmogorov complexity in information theory is an ideal concept to measure the degree of meaningfulness of the information contained by the data. However, the computation of this measure cannot always be correctly completed in a feasible time and space. For example, a pattern of pseudo-random dots is usually describable by a simple algorithm and a few parameters, but it is difficult to find the parameter values from the data even if the algorithm is given. The alternative method by which to approximate this measure is to use an algorithm of data compression, such as JPEG compression for a two-dimensional image. P. Machado *et al.* examined various methods for aesthetic measure based on complexity [6], and J. Rigau *et al.* gave deeper consideration to complexity as an aesthetic measure [10]. In this system, only one simple method has been implemented, namely, to measure the distance between the compression ratio of the given ideal value and the real value measured from the object image, just as E. den Heijer *et al.* examined [7]. The new system was implemented using an API of JPEG compression embedded in the MacOS X Cocoa framework with the image quality parameter as 0.67. Since the ideal value of the compression ratio should be variable depending on the preference of the user, this system allows the user to adjust the ideal value of the compression ratio using a graphical user interface (GUI) described later herein.

B. Global contrast factor

E. den Heijer *et al* used the global contrast factor (GCF) [11] as an alternative measure of the interestingness of a pattern [7], the algorithm of which was originally designed to evaluate *contrast* as closely as possible to the intuitive human measure. Contrast basically refers to the difference in brightness in a single image, but contrast is more than a simple statistical measure of variance among brightness values over the pixels. Contrast should be referred to as high contrast if the image is organized into only black and white pixels without any intermediate gray pixels but should be referred to as low contrast if the allocation of black and white for each pixel is randomly determined. K. Matkovic *et al.* proposed an algorithm by which to calculate a weighted summation among average differences over pixels for different resolutions of a single image [11]. Their original method uses a grayscale image of $1,024 \times 768$ pixels as the original image, and reduces the resolution by half in seven steps until the smallest resolution of 4×3 pixels is obtained. The weight values were statistically induced through psychological experiments involving human subjects.

In order to expand the original method to be applicable to a color image, the difference between brightness is changed into the Euclidean distance between color values in RGB color space. Three component values are scaled in $2:3:1$ for red, green, and blue to adapt to the characteristics of human eyes.

In the current implementation in SBART, the calculation starts from half of the original size, i.e., 512×384, because the computation time must be maintained short enough for the evolutionary process to be efficient. It is hoped that this issue can be resolved by developing a more efficient algorithm using a GPU.

C. One-dimensional brightness distributions

The above two measures consider the placement of colors in an image but do not consider their two-dimensional distribution. In these measures, one-dimensional patterns, such as an image of parallel stripes, are also given a chance to gain high aesthetic score. Two-dimensional Fourier analysis should be useful for evaluating such a pattern expansion over the image but requires a long computing time of $\mathrm{O}(N \log N)$ for each frequency. Another easy method by which to detect a pattern of parallel stripes is to compare the variances of the brightness distribution among rows and columns. If the image is a pattern of horizontal stripes, the variance among rows is large, but the variance among columns is zero.

The algorithm implemented here is used to calculate the distances between distributions of brightness for the ideal distribution and the measured distribution for each angle from $0°$ to $90°$ stepping by $15°$, transform each result value to adjust zero to 1 and the furthest value to 0, then take the geometric mean among the values. Measurement for the variation of different angles is helpful for detecting the orthogonal parallel stripes. The ideal distribution is extracted based on a statistical analysis over 1,000 snapshot photographs, similarly to the case of a color histogram described in the following subsection.

D. Color histogram

In addition to the shape, some statistics on colors are also important in order to index the characteristics of an image. The frequency distribution of brightness was examined in a previous study by E. den Heijer *et al.* using the distance between the ordered frequency distribution and an ideal distribution based on Benford's law [7]. Benford's law states that

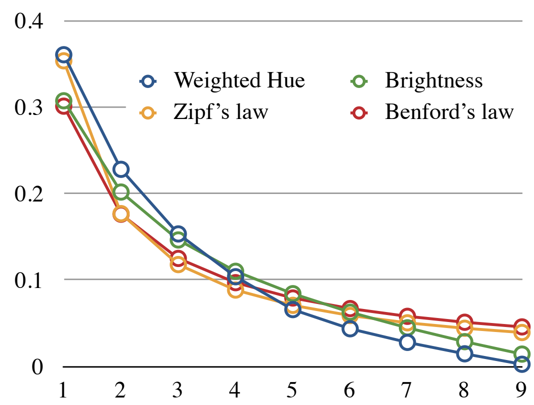


Fig. 1. Comparison among the distributions of the average hue histogram and the brightness histogram extracted from 1,000 snapshot photographs, Benford's law, and Zipf's law.

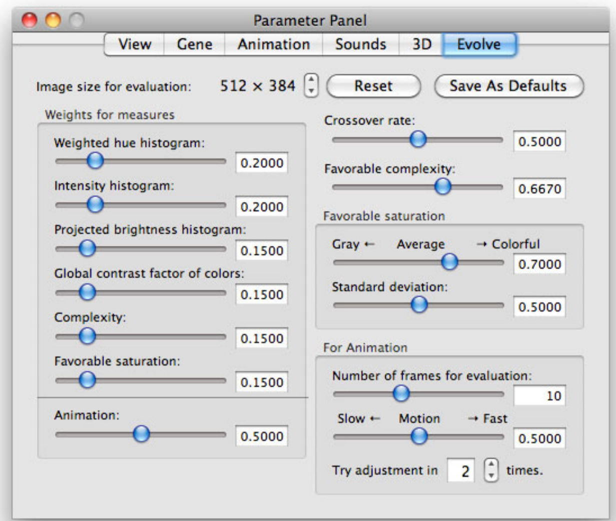


Fig. 2. GUI for parameter settings of automated evolution.

the appearance frequency of digit d at the highest column in prime numbers follows the value of $\log_{10}(1+1/d)$ [12]. The algorithm developed by E. den Heijer *et al.* counts pixels for each span of nine grades of grayscale from black to white over all pixels in a single image, sorts the frequencies calculated from the counted numbers in descending order, and then measures the distance from the distribution of Benford's law. Another distribution found from the frequency of occurrence for each word in a text is given by Zipf's law, which states that the frequency is proportional to the inverse number of the rank in descending order, i.e., the second most frequent word appears half as often as the most frequent word, the third most frequent word appears a third as often as the most frequent word, and so on [13]. Both of these distributions can be found in a number of natural phenomena.

In order to extend the method to be applicable to colors, we use the distribution of hue values as another measure in addition to brightness. The differences from the case of brightness are that the span of possible values is divided into 12 units with one additional unit for gray and that each count value is weighted by the saturation of the color, i.e., the distribution of the results is expressed by 13 real values. Instead of using the above two well-known distributions, we investigated the average distribution among snapshot photographs of natural and urban sceneries and portraits. The results obtained using 1,000 photographs revealed similar shapes to the results provided by Zipf's law, but differ in the lower ranks, as shown in Figure 1. Therefore, as the ideal distribution, we embedded a sequence of frequencies induced from this investigation. The distance is calculated by taking the summation of the absolute differences between the ideal frequency and the calculated frequency for each rank in the manner described by [7].

E. Favorable distribution of saturation

There is no universal aesthetic criteria on the tone of colors. However, the user sometimes has a preference for a monotone or color image. We introduce a user interface that allows the user to indicate the ideal values of the average and standard deviation on saturation values over all pixels as a part of the parameter panel shown in Figure 2. A gray image becomes preferable if both the average and the standard deviation are low, and a colorful image becomes preferable if the average is high.

When we define the range of saturation values within a fixed span, such as $[0,1]$, the possible value of standard deviation is limited to within a range depending on the average value. The standard deviation takes the maximum value when the sample values are restricted in the edge values of the range, i.e., 0 or 1. In this case, the average μ and the maximum standard deviation $\sigma_{\max}$ are:

$$\mu = P_1 \tag{1}$$

$$\sigma_{\max} = \sqrt{P_0 P_1} \tag{2}$$

where P_0 and P_1 are frequencies of saturation values of 0 and 1, respectively, i.e., $P_0 + P_1 = 1$. A detailed derivation of equation 2 is presented in Appendix. For convenience, the user is allowed to input the value within $[0,1]$ as the standard deviation, and the system multiplies the standard deviation with $\sigma_{\max}$ of equation 2 to obtain the actual ideal value. The measure is a two-dimensional Euclidean distance between the ideal values and the actual values extracted from the image.

F. Normalization and unified measure

It is necessary to develop a type of normalization for the six measures described in the above subsections, because these measures have different dimensions and cannot be directly compared. We transform each measure in the range of $[0,1]$, so that 0 indicates the worst measure, and 1 indicates the best measure, and we map each measure to the normalized measure in two stages, so that all of the processed values form distributions of similar shapes. The first stage is a gamma correction $f_1(x) = x^{\gamma}$, so that the average value is equal to

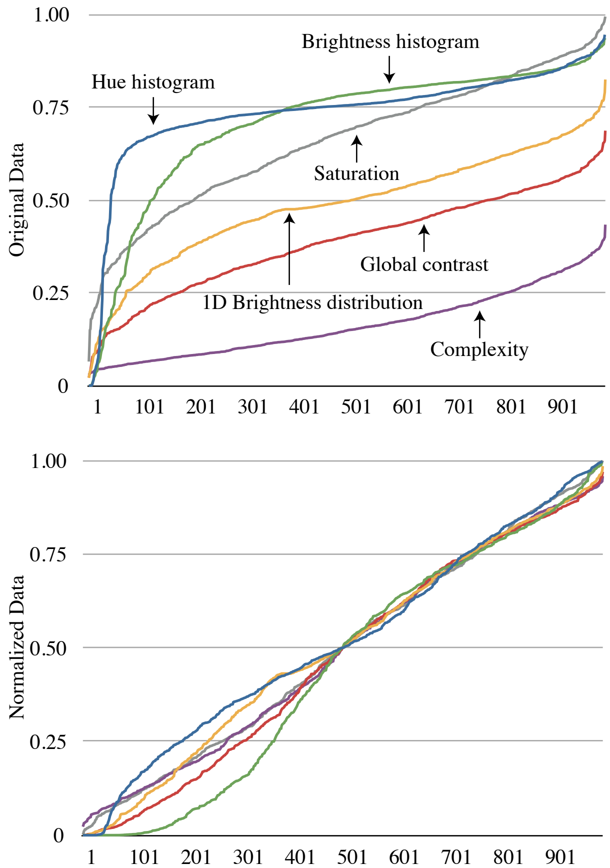


Fig. 3. Distributions of aesthetic measures for 1,000 images produced from random genotypes. Each measure is sorted in ascending order.

the median value, i.e., $\gamma = \log_m \bar{x}$, where m is the median and $\bar{x}$ is the average value of x. The second stage is a linear transformation, so that the average and the standard deviation are adjusted to 0.5 and 0.2, respectively. The equation is $f_2(x) = (0.2/\sigma_2)(x - m) + 0.5$, where σ_2 is the standard deviation among the transformed values by f_1. Combining these two stages, the transformation function f is defined as:

$$f(x) = f_2(f_1(x)) = \frac{0.2}{\sigma_2}(x^{\log_m \bar{x}} - m) + 0.5. \tag{3}$$

If the final value is out of the range of $[0, 1]$, it is forced to be revised to the nearest boundary 0 or 1. In order to determine the coefficients in the transforming functions, we examined 1,000 images drawn with randomly generated genotypes. Figure 3 shows the distribution of each measure and its normalized version.

The final aesthetic measure for a still image is calculated as the weighted geometric mean among these measures, where the weights are adjustable by the user using a GUI in Figure 2. The sliders for each measure allow the user to operate any of the measures at any time. Once one of these values is changed manually, the others are automatically adjusted so as to ensure that the summation of the weights remains 1 and to ensure that the ratio among the other weights stays same if possible.

TABLE I
CORRELATIONS AMONG MEASURES.

	Brt. H	1DBD	GCF	Cmpl.	Sat.
Hue H	0.0133	0.0204	0.0002	0.0126	0.0244
Brt. H		0.0054	0.3292	0.2591	0.0785
1DBD			0.1574	−0.2586	0.0482
GCF				<u>0.5907</u>	0.0854
Cmpl.					0.0579

Hue H: Hue histogram GCF: Global contrast factor
Brt. H: Brightness histogram Cmpl.: Complexity
1DBD: 1D brightness distribution Sat.: Saturation

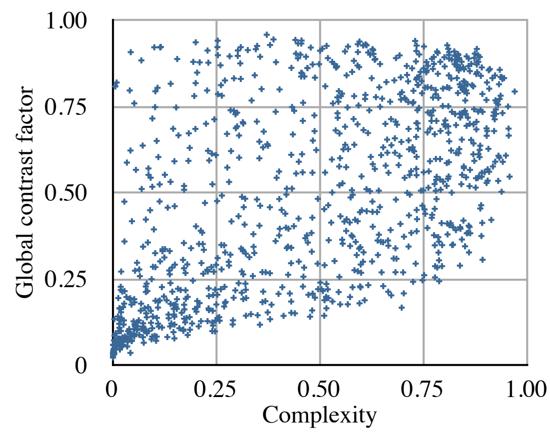


Fig. 4. Scatter plot between global contrast factor and complexity.

G. Correlation between measures

Some of the measures may be useless if they exhibit strong correlation. Table I summarizes the correlation factors between two of the measures described above. The values used for calculation are obtained after normalization. This shows that there is no strong correlation except between the GCF and the complexity. From the scatter plot in Figure 4, the areas of high density are found in both the lower left and upper right corners. However, a high score for one of these two measures does not always mean the score of the other measure will be high. This suggests that both of these measures should be included in the evaluation.

III. AESTHETIC MEASURE FOR ANIMATION

One unique feature of SBArt4, which was released in 2010, is that the user is allowed to breed not only still images but also animations in real time [9]. This used to be difficult because the required computation time to render one frame was much longer than the usual frame rate for smooth animation. However, the recent improvement in GPUs has made this possible. We tried to implement a measure of favorable degree of movement in animation to be combined with the measures for still images. Ideally speaking, this type of measure should be constructed along with total evaluation over the entire duration of the animation, but this seems difficult because of the enormous computation power required. As the first step of minimal functionality for this measure, we implemented an algorithm to calculate the average difference between consecutive frames among samples selected from the entire sequence of frame images. It is theoretically possible to breed an animation of arbitrary duration, while ensuring that the evaluation by the user remains easy, SBArt4 treats relatively short animations, with a default duration of four seconds. As usual, there is a trade-off between quick response and precision. In order to ensure efficient progress of automated evolution, the current implementation uses ten samples by default to estimate the average motion by calculating the distance between color values for pixels of the same position in a sampled frame and the pixels in the next frame. Because the degree of motion should be adjustable based on user preferences, the system provides a slider to allow the user to set up the value at any time. The slider is located at the bottom right of the GUI, as shown in Figure 2.

The measure for a still image is also applied to each frame image of ten samples. The total evaluation is calculated as the weighted geometric mean between the average measure of still images and the average degree of motion in sampled frames. The weight is also adjustable by the user using a slider at the bottom left of the GUI.

IV. Method of generation alternation

From the viewpoint of the main objective of this system, the generational change should be designed in with small-grained alternation in order to ensure that the computation time between generations is short enough for flexible interaction with the playback process. Among the several methods available for this style of genetic algorithm, we select the minimal generation gap (MGG) model [14] because this model provides the smallest grain size and effective exploration of a large search space. In each step of generation alternation, the algorithm performs the following steps:

1) Randomly elect two individuals from the population as parents,
2) organize a family by producing two children by crossover and mutation,
3) select the best individual from four members in the family,
4) randomly select another individual from the family,
5) restore the two selected individuals into the population, and then
6) discard the two individuals in the family that were not selected.

Random selections for both the parents and the second survivor are effective in order to maintain broad diversity in the population, and the selection of the best individual from the family has the same effect as the elitist strategy that guarantees the best solution discovered in the process always remains in the population.

Figure 5 shows a typical trend for the improvement of the fitness values in an evolutionary process. The first 20 steps are used only to calculate the fitness values of the best 20 individuals given as the initial population imported from a field for breeding. The average values in the graph are found from fitness values known among the population until the 20th step and are later based on the 20 best individuals.

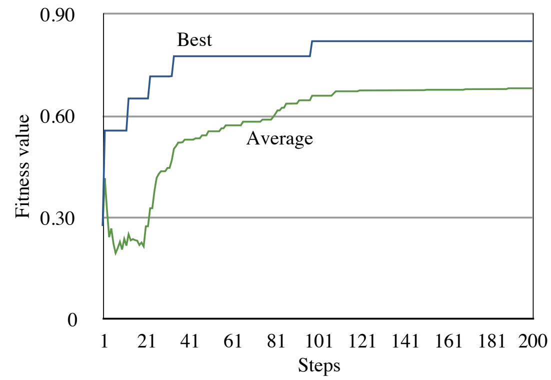
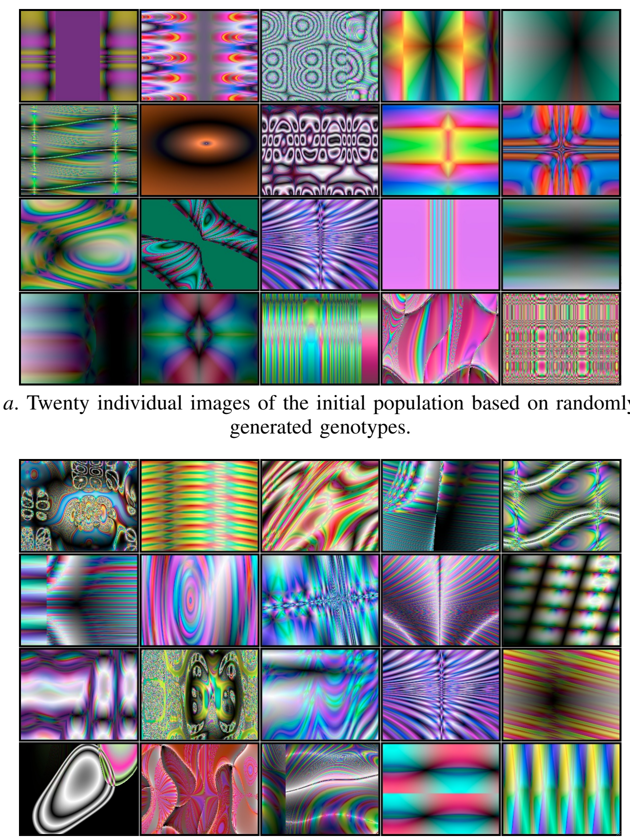


Fig. 5. A typical evolutionary process.



a. Twenty individual images of the initial population based on randomly generated genotypes.

b. The best 20 individual images after 200 steps of generation alternations.

Fig. 6. A comparison between populations of random and evolved individuals.

Figure 6 shows an example of sets of individual images of initial and evolved populations. The population size is 80 but is initialized by 20 randomly selected genotypes and their children. The fitness values of individuals in the initial population in the upper figure are from 0 to 0.65166, the average of which is 0.21530. These of fitness values improved to a range of from 0.61281 to 0.82065, the average of which is 0.68176, where the best individual in the initial population remains.

We designed a GUI, shown in Figure 7, in order to make it easy for the user to monitor the evolutionary process. The

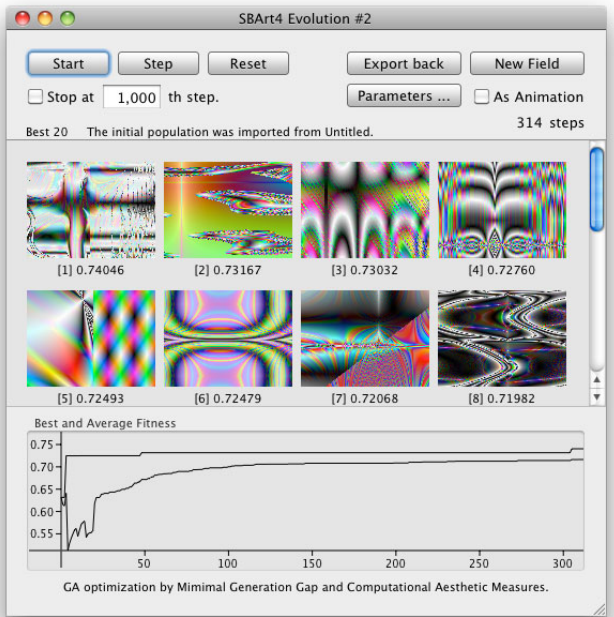


Fig. 7. GUI for automated evolution.

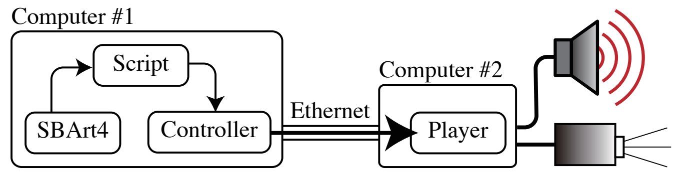


Fig. 8. System setup for installation of automatic art.

middle part of the window displays the best 20 individuals discovered in the process with each fitness value. The bottom part shows a live animation of the trend graph of the best fitness value and the average fitness value among the best 20 fitness values. Each measure of both the still image and the animation is observable by a tool-tip balloon attached to each view of the best 20 individuals that pops up when the mouse cursor remains a number of seconds over the view. The user is allowed to make individuals migrate between the population for evolution and the field for breeding. These functionalities are not necessary for automatic art, but are useful for a combination of evolution and breeding.

V. System Setup for Installation

SBArt4 itself has the functionality to play an individual animation in full-screen mode in order to display the production results. However, this consumes too much computational power to be executed in parallel with the evolutionary process, because of usage conflict in both the GPU and the CPU. From the viewpoint of artistic installation, it is better to keep the frame rate higher than 24 frames per second and to ensure that the resolution is compatible with high-definition TV. In order to achieve this requirement using reasonably priced and sized personal computers, we examined a combination of two machines connected by a LAN cable to execute the evolutionary process and the playback process separately. One machine runs the automated evolution and the other manages the playback of the animation, and the compiled code of individual to be played back is transferred from one machine to the other, as shown in Figure 8.

We developed two new software applications. *Controller* runs on the same machine on which SBArt4 is running, and *player*, which runs on the other machine. The main role of *player* is to receive the text of a code written in shading language in order to draw the animation on the hi-resolution screen. *Controller* is a type of communication bridge between SBArt4 and *player* and receives code from SBArt4 through the general pasteboard, which is usually used to copy and paste between independent applications, and passes the code to *player* by means of a TCP/IP connection through the network. *Controller* has a GUI to control *player* remotely in order to change parameters for timing and sound effects.

For the automatic art, we developed software in AppleScript that supervises both SBArt4 and *controller*. SBArt4 was also extended to accept the scrip commands to be controlled by the script. The limitation of the size of the genotypes was also introduced in order to prevent the infinite growth of a functional expression through long-term evolution. In the current setup, the number of symbols in a single genotype is limited to 200. The script initiates an evolutionary process in SBArt4 starting with a random population and selects 10 individuals after 50 steps in order to copy their codes into a controller in turn. The script then commands SBArt4 to reconstruct the population by genetic combination among the individuals in the current population. Some of the individuals are replaced by new random genotypes in order to maintain diversity in the population. The evolutionary process restarts and continues until playback of all previous 10 individuals is complete. These processes are iterated an arbitrary number of times in order to produce an ever-changing stream of abstract animations. The duration of an animation for each individual is 20 seconds in a typical setting, in which the evolutionary process is allowed to continue for three minutes and 20 seconds in order to produce the next 10 individuals for playback. This is a sufficient time for a small computer, such as Mac mini with a 2.5-GHz Core i5 and an AMD Radeon HD 6630M, to execute more than 50 steps of generational changes in the MGG model described in Section IV.

A synchronized sound effect is also added to the animation. The sound is synthesized with parameters extracted from a statistic analysis of frame images.

VI. Automated Daily Production

The functionality of automated evolution has enabled not only an installation of automatic art but also automated production without the assistance of a human. From October 6th, 2011, the system has been automatically producing 10 movies every day. The production procedure starts at 10 A.M.

Japanese Standard Time, continuing the evolutionary process from a random population until the completion of 200 steps of generation alternation. The procedure selects the 10 best individuals from the final population and generates 20-second movie files for each with a synchronized sound effect.

Each of the produced movie files is compressed in both the H.264 and Ogg formats in order to be adaptable for playback by popular web browsers, such as Safari, FireFox, Google Chrome, and Opera. These movies are accessible from http://www.intlab.soka.ac.jp/~unemi/sbart/4/DailyMovies/. Reorganization of a web site to adapt to the newly generated movies is also performed automatically just after the compressed movie files are uploaded to the web server. The daily and weekly digests of these movies are also posted to a popular site for movie sharing. The daily digest is a sequence of six-second excerpts for each movie, for a total duration of one minute. The weekly digest is a sequence two-second excerpts for each of the 70 movies produced in the last seven days. These digests are accessible at http://www.youtube.com/user/une0ytb/.

The entirety of the daily process is controlled by a program in AppleScript and requires approximately 40 minutes, including evolution, movie file generation, compression, and posting. The daily process consumes an average of 346 MB of the storage in the web server everyday, which means that storing all of the movies produced over a number of years on a hard disk drive is feasible, because 126 GB for one year's worth of movies is not unreasonable, considering the HDD capacity of currently available consumer products.

Figure 9 shows sample frame images of movies produced on January 5th, 2012. These movies are accessible at http://www.intlab.soka.ac.jp/~unemi/sbart/4/DailyMovies/index.html?92.

VII. Conclusion

We implemented an automated evolution based on a combination of aesthetic measures for fitness evaluation and minimal generation gap for generation alternation in a breeding system SBArt4. This type of approach has already been examined by P. Machado *et al.* [6], but we added extended and newly developed measures and a GUI for flexible combination among measures. Evaluation of a favorable amount of motion in an animation is a new challenge in this system. Although there remain points that should be considered in order to improve the effectiveness of supporting the production, we succeeded in building a first trial of feasible implementation for fully automated art using the power of a GPU. The method of combining different measures is a subject for reconsideration. Here, we used a geometric mean instead of a weighted summation because measures should contribute as necessary conditions. The use of fuzzy logic to calculate a membership value between logical conjunction is an alternative method, where the smallest value among the measures is used as the resultant value of the combination. Some of the users might prefer to use more than two measures as sufficient conditions.

The techniques to be examined in the near future are as follows: (1) Two-dimensional Fourier transformation and analysis of the resulting spectra as one of the measures for a still image, and (2) optical flow for animation and analysis of the distribution of flow vectors as an alternative measure for animation, and (3) information theoretical complexity of three dimensional arrangement of boxels as another alternative measure for animation. These ideas should be useful as methods for fitness evaluation but should be checked with respect to how they consume the computation power in order to keep the elapsed time short enough for our purposes. The development of efficient algorithms will be examined in the future.

As applications of automated evolution, we organized an installation of automatic art and a daily automated production of movies for publication on the Internet. Although not examined in the present paper, another artistic application for live performance [15] was realized and was performed at the Generative Art Conference in Rome, 7th of December, 2011. All of these projects remain in the experimental stage. We hope that these projects will be accepted by the widest possible audience.

The binary code of SBArt4, which is runnable on MacOS X 10.6 or newer versions, is available from http://www.intlab.soka.ac.jp/~unemi/sbart/4/.

We hope the present research will inspire something new in human culture, especially concerning the relation between creativities of human and machine.

References

[1] H. Takagi, "Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation," *Proceesings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[2] H. Takagi and H. Iba, "Interactive evolutionary computation," *New Generation Computing*, vol. 23, no. 2, pp. 113–114, 2005.

[3] K. Sims, "Artificial evolution for computer graphics," *Computer Graphics*, vol. 25, pp. 319–328, 1991.

[4] H. Cohen, "The further exploits of AARON, painter," *Stanford Electronic Humanities Review*, vol. 4, no. 2, 1995.

[5] L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer, Eds., *Computational Aesthetics 2005: Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, May 2005.

[6] P. Machado and A. Cardoso, "All the truth about NEvAr," *Applied Intelligence*, vol. 16, pp. 101–118, 2002.

[7] E. den Heijer and A. E. Eiden, "Using aesthetic measures to evolve art," in *WCCI 2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, July 2010, pp. 4533–4540.

[8] T. Unemi, "Simulated breeding: A framework of breeding artifacts on the computer," in *Artificial Models in Software*, 2nd ed., M. Komosinski and A. A. Adamatzky, Eds. London, UK: Springer-Verlag, 2009, ch. 12.

[9] ——, "SBArt4 – breeding abstract animations in realtime," in *WCCI 2010 IEEE World Congress on Computational Intelligence*, Barcelona, Spain, July 2010, pp. 4004–4009.

[10] J. Rigau, M. Feixas, and M. Sbert, "Informational aesthetics measures," *IEEE Computer Graphics and Applications*, vol. 28, no. 2, pp. 24–34, 2008.

[11] K. Matkovic, L. Neumann, A. Neumann, T. Psik, and W. Purgathofer, "Global contrast factor – a new approach to image contrast," in *Computational Aesthetics 2005*, 2005, pp. 159–168.

[12] J.-M. Jolion, "Images and Benford's law," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 1, pp. 73–81, 2001.

[13] G. K. Zipf, *Human behavior and the principle of least effort*. New York: Hafner Pub. Co., 1949.

[14] H. Satoh, I. Ono, and S. Kobayashi, "A new generation alternation model of genetic algorithms and its assessment," *Journal of Japanese Society for Artificial Intelligence*, vol. 12, no. 5, pp. 734–744, 1997.

[15] T. Unemi, "SBArt4 as automatic art and live performance tool," in *14th Generative Art Conference*, Rome, Italy, December 2011, pp. 436–449.

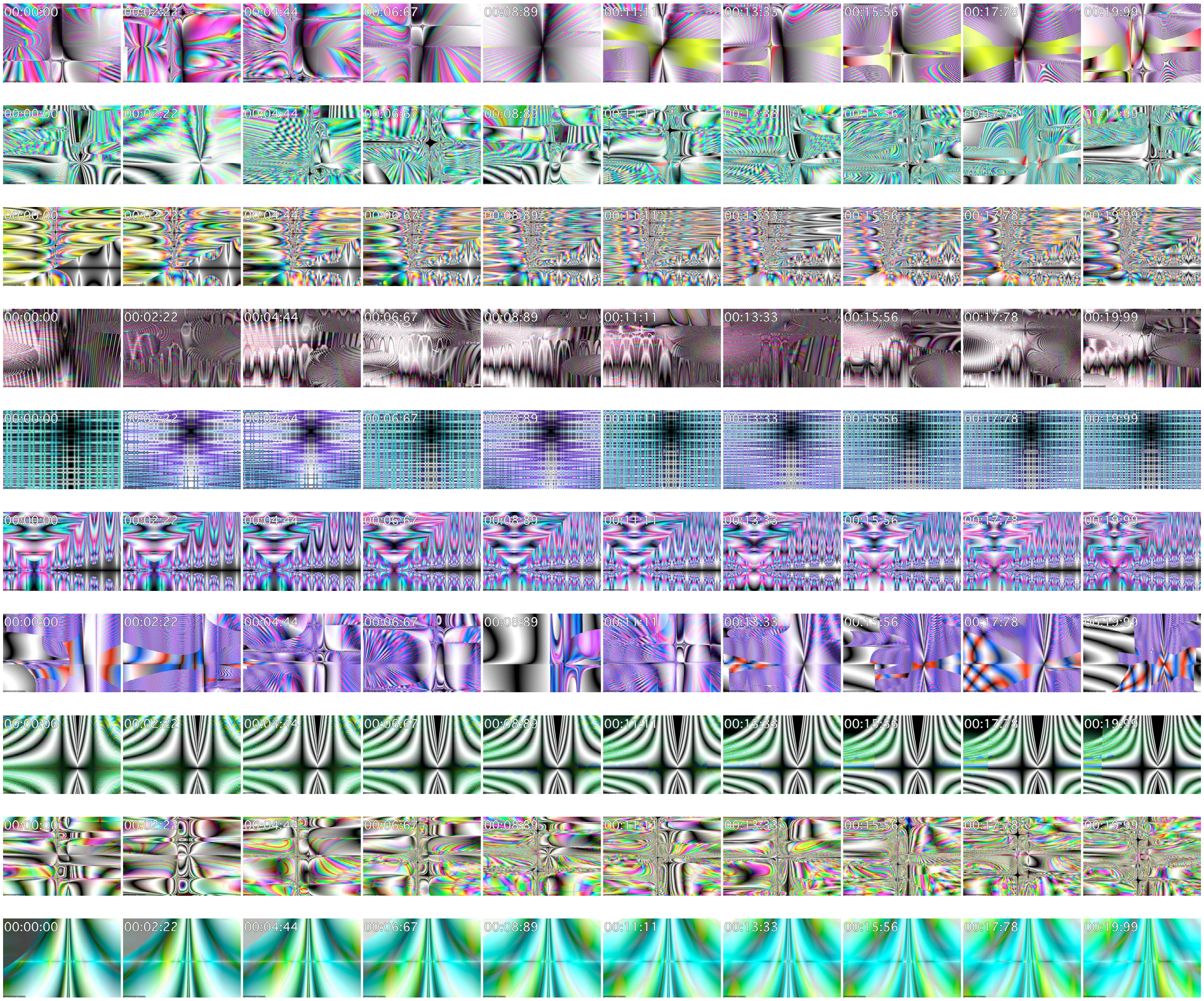


Fig. 9. Sample frames from left to right of each of 10 movies from top to bottom produced automatically on January 5th, 2012.

APPENDIX

From the definition of standard deviation,

$$\sigma_{\max}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \overline{x^2} - \bar{x}^2.$$

Since $x_i = 1$ or 0 and the expected number of x_i having a value of 1 is $P_1 N$,

$$\overline{x^2} = P_1 \quad \text{and} \quad \bar{x}^2 = P_1^2.$$

Therefore,

$$\sigma_{\max}^2 = P_1 - P_1^2 = (1 - P_1)\,P_1 = P_0\,P_1.$$