Quasi-random Image Transition and Animation

Aneta Neumann[1], Frank Neumann[1], and Tobias Friedrich[2]

[1] Optimisation and Logistics, School of Computer Science,
The University of Adelaide, Australia
[2] Algorithm Engineering, Hasso-Plattner-Institut,
The University of Potsdam, Germany

Abstract. Evolutionary algorithms have been widely used in the area of creativity. Recently, evolutionary processes have been used to create artistic image transition processes using random walks. In this paper, we explore the use of quasi-random walks for evolutionary image transition and animation. Quasi-random walks show similar features as standard random walks, but with much less randomness. We utilize this established model from discrete mathematics and show how agents carrying out quasi-random walks can be used for evolutionary image transition and animation. The key idea is to generalize the notion of quasi-random walks and let a set of autonomous agents perform quasi-random walks painting an image. Each agent has one particular target image that they paint when following a sequence of directions for their quasi-random walk. The sequence can easily be chosen by a user and allows them to produce a wide range of different transition patterns and animations.

Keywords: Quasi-random · computational intelligence · random walks · animation.

Introduction

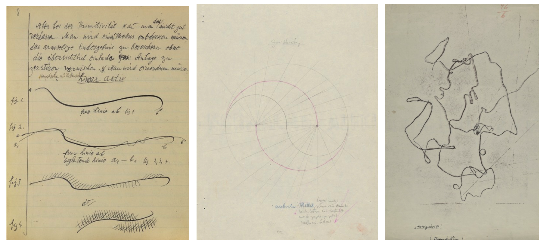
Evolutionary algorithms have been used in various ways to create art and music. In the context of art, various evolutionary computing methods have been used to create artistic images [2, 13, 15, 21, 25, 29, 33, 34]. Recently, the notion of evolutionary image transition [23] has been introduced and the goal of this line of research is to use the evolutionary process itself to create evolutionary processes that are interesting from an artistic perspective and potentially inspired to artists using evolutionary computation. Furthermore, the underlying process of evolutionary image transition has been used for evolutionary image composition taking into account features of the given images [24, 26].

We aim at creating creative work in the context of evolutionary image transition and animation. The key idea in this area is to use *randomness* to create interesting processes of visual effects. Evolutionary dynamics and random walks can be closely linked and random walks are often used to explain evolutionary behavior [3]. Previously, random walks have been used for evolutionary image transition context [23]. Especially random walks biased on the given images can create very interesting effects when it comes to evolutionary image transition. However, the drawback of this is that the behavior is hard to control by the user/designer of the application.

This paper presents a new approach for evolutionary image transition using quasi-randomness and extends the framework to evolutionary animations. Quasi-randomness has already been used in evolutionary computation and it has been shown that using this may lead to better performing and better understandable evolutionary algorithms [30–32]. We replace the classical random walk with a so-called *quasi-random walk*. In this model, the decisions of the walker are not chosen uniformly at random, but deterministically. Instead of moving to a random neighbor, the walker visits all neighbors in a fixed order. This model is well-studied in discrete mathematics for investigating how much randomness is required to achieve similar properties of regular random walks, but with less randomness. The quasi-random walk was rediscovered independently several times in the literature. In order of appearance, it has been called "Eulerian walker" [27], "edge ant walk" [35], "whirling tour" [10], "Propp machine" [7, 19], "deterministic random walk" [6, 8], and "rotor-router model" [11, 12]. To show the relationship to standard random walks, we mostly use the term "quasi-random walk" in the rest of the paper. We will synonymously use "rotor-router model" to sometimes emphasize the inner workings.

Related artistic work

At the begin of the 20th century the well-known artist Paul Klee was part of the famous Bauhaus movement [9]. Klee was a unique university teacher in Weimar and Dessau and an iconic promoter of a theoretical approach to making art. In the Pedagogical Sketchbook [20] the artist shows his innovative approach to artistic expression. One of his important messages about art and design is "an active line on the walk, moving freely, without goal." He describes the line as the most human mark and characterizes the various types of a line. Figure 1 (a) shows "the linear line they are subdivided in some modus in contrast to the primitiveness as a free line, a free line with accompanies of lines". In (b), we see the drawing of "construction



(a) (b) (c)

Fig. 1: Illustrations of Paul Klee methodology on simple strokes. Figures from Zentrum Paul Klee [36]

with middle curve from two centers radiated and in associated region shaft-wide halved". Finally, in (c), we see the drawing with the title Honigschrift as "a flowing line". Paul Klee's notes include 3900 pages and demonstrate in multiple ways how a point can become a line and the line the plane can result in "free irregularity". Quasi-random walks essentially create lines in the space where they perform the walk and we take this as an inspiration for designing our transition and animation methods.

We use quasi-random walks to design algorithms for image transition and animation [3]. In our algorithms, a set of agents perform quasi-random walks where the characteristic of the random walk can easily be determined by a user. The process starts with a given starting image (which may be white or blank). Each agent has an image that it paints. It does so by carrying out its quasi-random walk as determined by the router sequence chosen by the user. The r agents perform their quasi-random walks in parallel painting different images (in the case of animation) or the same target image in the case of transition.

The user can easily determine the behavior of each agent by setting the sequence of directions that is followed at every pixel. An example sequence could be (left, down, right, up, left). Each pixel has its current active direction in the sequence. If the agent visits a pixel p, it paints it with the corresponding pixel of its target image. Afterwards, the agent moves to the next pixel as directed by the current active direction at p in its sequence and updates the active direction at p to the next one in the sequence. The use of different sequences for the agents allows to create various types of interesting transition and animation processes. Analyzing the images created during the animation process with respect to different artistic features, we show that quasi-random animation using different rotor sequences allows to create animation processes with a wide range of different artistic behaviors.

We evaluate the results obtained by the different processes in two ways. Firstly, we carry out a human-based investigation and ask for preferences regarding image created at similar stages of the (quasi-)random processes. Furthermore, we evaluate the quasi-random animation processes in terms of different important features judging the aesthetic of the images [1, 22].

The remainder of this paper is organized as follows. In the next section, we give a brief introduction into quasi-random walks and present our approach for quasi-random image transition and animation. Afterwards, we examine the wide range of image transitions and animations that can be created using various types of agents. We carry out a human user validation and a feature-based analysis of the agents behavior. Finally, we finish with some concluding remarks.

Quasi-random transition and animation

We now describe the concept of quasi-random walks in greater detail and present our method for carrying out quasi-random image transition and animation. A quasi-random walk is best described on the two-dimensional infinite grid $\mathbb{Z}^2$. Imagine a single walker on some arbitrary vertex $(x, y) \in \mathbb{Z}^2$. The walker is allowed to move to one of the four neighbors right, down, left, or up, that is, to $(x+1, y)$, $(x, y+1)$, $(x-1, y)$, or $(x, y-1)$. A random walk would choose independently and uniformly at random one of these four neighbors and move there. In order to use less randomness, the quasi-random walk assigns a permutation of the four directions right, down, left, up to each vertex of the grid. These permutations are fixed initially and deterministically determine the behavior the quasi-random walkers. To store the current status of the permutation, each vertex has a rotor to manage which neighbor to visit next. Each time a walker is leaving a vertex, it moves in the direction of the rotor and updates the rotor according to the fixed permutation of the vertex. When the rotor reaches the last position of permutation, it wraps around and starts again at the beginning of the permutation. The advantage of the rotor-router model is that it doesn't require randomness and the process is completely determined by the sequence used and the starting position of the rotor.

The simplest rotor sequences are $(\text{right}, \text{down}, \text{left}, \text{up})$ or $(\text{right}, \text{ left}, \text{down}, \text{up})$. In both cases the quasi-random walker visits the four neighbors as uniformly as the standard random walk in expectation. Note that the first, clockwise rotor sequence is circular while the second rotor sequence is non-circular. It has been proven that such quasi-random walkers behave very similar to the expected behavior of a standard random walk [8]. In fact, there is even a slight difference between both

[3] Videos are available at `https://vimeo.com/anetaneumann`

Algorithm 1 QUASI-RANDOM ANIMATION

Require: Start image Y of size $m \times n$. For each agent k, $1 \leq k \leq r$, an image I^k of size $m \times n$, sequence S^k and position counters $c^k(i,j) \in \{0, \ldots, |S^k|\}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

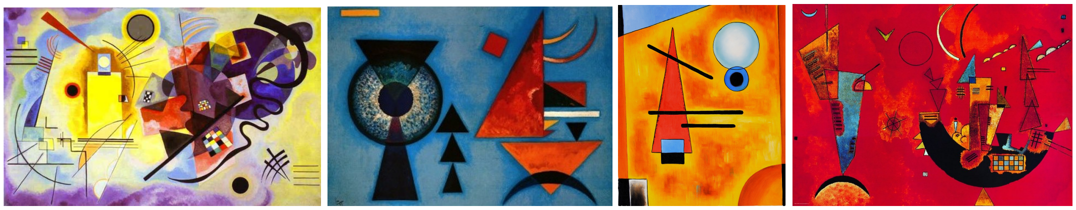
$X \leftarrow Y$
for each agent k, $1 \leq k \leq r$ **do**
    choose $P^k \in m \times n$ and set $X(P^k) := I^k(P^k)$.
end for
$t \leftarrow 1$
while $(t \leq t_{\max})$ **do**
    for each agent k, $1 \leq k \leq r$ **do**
        Choose $\hat{P}^k \in N(P^k)$ according to $S_k(c(P^k))$.
        $X(\hat{P}^k) \leftarrow I^k(\hat{P}^k)$
        $c^k(P^k) \leftarrow (c^k(P^k) + 1) \mod |S^k|$.
        $P^k \leftarrow \hat{P}^k$.
    end for
    $t \leftarrow t + 1$
end while

aforementioned rotor sequences: The second, non-circular rotor sequence is proven to behave even a bit closer to the expected random walk than the first, circular rotor sequence.

Following the idea of a "stack walk" [18], we generalize the concept of the quasi-random walk and allow not only permutations and rotor sequences of length four, but arbitrary sequences of the four cardinal directions. If some directions occur more frequently than others, this results in a bias of the walker in a particular direction. For example, the rotor sequence (right, left, up, down, right, left) will much more explore horizontally than vertically. This bias implies a deviation from the expected behavior of the standard random walk but allows for more interesting effects. We take the technical behavior of quasi-random walks and combine it with Kandinsky's art to create artistic work. *Quasi-random animation* follows the idea of the rotor-router model in order to produce artistic ideas by agents carrying out quasi-random walks. The pseudo-code for quasi-random animation is given in Algorithm 1.

We consider a set of r agents where each of them has the goal to paint one particular image I^k. Each agent k, $1 \leq k \leq r$, works with a sequence S^k which consists of entries from $\{\text{right}, \text{down}, \text{left}, \text{up}\}$. For example, we could have $S^k =$ (right, left, down, up) as in the standard rotor-router model, but also sequences such as S^k = (right, left, up, down, right) which are not symmetric with respect to the different directions. Using such asymmetric sequences creates a bias in the walk performed by the agent and leads to various interesting effects dependent on the choice of S^k. At each time step each agent k moves from its current position P^k to one of its pixel neighbors $\hat{P}^k \in N(P^k)$ where the direction $S^k(c(P^k))$ is determined by the current active pointer at position P^k (wrapping around at the border of the image). Afterwards the current image X gets updated by setting the pixel value at position $\hat{P}^k$ to the value of image I^k at position $\hat{P}^k$. The move of agent k is completed by increasing the sequence counter $c^k(P^k)$ in a modulo fashion and updating the current position P^k to $\hat{P}^k$. *Quasi-random image transition* is a special case of quasi-random animation where all I^k are set to one particular target image I. The key aspects when using this algorithm to create interesting transitions and animations is the number of agents, the images that they are using for painting and the router sequence for each agent. The choice of the router sequence S^k determines the random walk behavior of the agent k. A sequence can be an arbitrary sequence of directions. We mainly study sequences that are rather short in this paper. The reason is that this already gives a large variety of different effects. Furthermore, short sequences are user friendly as they can be easily adapted and understood by the user. The potential of longer sequences (partially explored in our feature-based analysis) allows to design more fine grained animations. In addition, the approach may be extended to have for each pixel its own sequence of directions which would allow to resemble the behavior of biased random walks used for example in image segmentation [14].

A sequence of length 4 containing each direction exactly once such as (right, down, left, up) corresponds to the classical rotor-router model leading to a cover time that is close to the one of a classical random walk which moves at each to a neighboring pixel selected uniformly at random. However, the setting leads to a much smoother transition and animation being generated as it avoids the unbalance in terms of the different directions that occurs in the classical random walk. An unbalanced sequence such as (right, down, left, up, right) where each direction does not appear the same number of times, introduces a bias into the direction that appears more often. Let r be the total sequence length and r_i the number of times direction i appears in this sequence. Assume that the agent has done M steps, where M is large. Then the number of steps it has taken into direction i is roughly $M \cdot \frac{r_i}{r}$. This means that the sequence length r and the number of appearances r_i of direction i can be used to create arbitrary fractions of moves into the different directions. Given that each agent is working



(a) (b) (c) (d)

Fig. 2: Starting and target images

with its own sequence and therefore its own fractions of the different directions, complex behavior can be created by using agents with different types of sequences.

Results for quasi-random image transition

We now show our results for quasi-random image transition where an agent paints the a given image I. We use the pairs of images shown in Figure 2 for our experiments.

To illustrate the effect of the diverse methods presented in this paper, we consider the Yellow-Red-Blue, 1925 (a) and the Cool, 1927 by Wassily Kandinsky (c) as a starting image. We consider the Soft Hard, 1927 (b) and the With and Against, 1929 by Wassily Kandinsky (d) as a target image for image transition. In the experiments, each agent performs $5\,000K$ steps and the images are of size 400 x 400 pixels. This setting holds for all experiments reported on in this paper. We should mention that the use of other starting and target images would show the same algorithmic behavior as the performance of the agents in Algorithm 1 is independent of the choice of the images. However, using different images allows to create different artistic effects and we will use different sets of images to illustrate a wide range of effects we observed.

We consider 1 agent carrying out the image transition. For the experiments with 1 agent, we set the starting point of the agent to the middle of a given image (a) and observe how it transitions into image (b). The results in Figure 3. To exhibit the differences of the symmetric quasi-random walk in comparison to a classical random walk, we show the results of the classical random walk [23] in Figure 3 (top row).

For the first experiment for quasi-random image transition, we consider the symmetric sequence S^1=(right, down, left, up) which resemble a standard roter-router model. Figure 3 (middle row) shows the quasi-random image transition process in a circle-patch style transforming image (a) into image (b). We observe the movement of the agent circulated in right direction over the image. It can be observed that the quasi-random walk (middle row) produces much smoother images than the classical random walk (top row). The quasi-random image transition produces visual pleasing results with bridging the parts of both images in a new abstract composition. In the second experiment, we expand our approach to non standard roter-router settings and use an asymmetric sequence for the transition. We add only one additionally position to the sequence which introduces a bias into the added direction. For the experiment, we consider the asymmetric sequence S^1= (right, down, left, up, right). Figure 3 (bottom row) shows the results. In contrast to the symmetric quasi-random walk, the behavior of the quasi-random image transition with the asymmetric sequences leads to horizontal stripes over the whole image. These stripes gather over time leading to interesting cumulative effects with the target image finally appearing. It also results in a much faster transition process. This is due to the additional direction ”right” in the chosen sequence.

Results for quasi-random animation

We now present results obtained for quasi-random animation. We showcase the results using interesting and characteristic images of the considered animation process. The different animation videos are available online 3. Furthermore, the feature-based analysis in the next section provides an analysis over the whole animation process with respect to artistic features.

2 Agents We use 2 agents starting at positions $[m/4, n/4]$ and $[3m/4, 3n/4]$ and images from Figure 2. Image (c) is used as the starting image and as image I^1 for the first agent, and image (d) is used image as I^2 for the second agent. Firstly, we consider the behavior when using the symmetric sequences S^1 = (right, down, left, up) and S^2 = (up, left, down, right). Figure 4 (top row) shows the results. The second agent starts to paint the image in circle appearances according to the symmetric agent movement. The animation obtains a more dynamic character when the regions painted by the agents meet and the agent operate at a ”larger radius”. This leads to the effect that the agents operate from all direction on the image. It looks as if the agents naturally pervade the image and effortlessly collaborate together.

(a) (b) (c) (d)

Fig. 3: Image transition for 1 agent. Top row: standard random walk [23]. Middle row: symmetric sequences (images after $100K$, $500K$, $2\,000K$ and $3\,500K$ steps). Bottom row: asymmetric sequences (images after $100K$, $200K$, $300K$ and $500K$ steps)

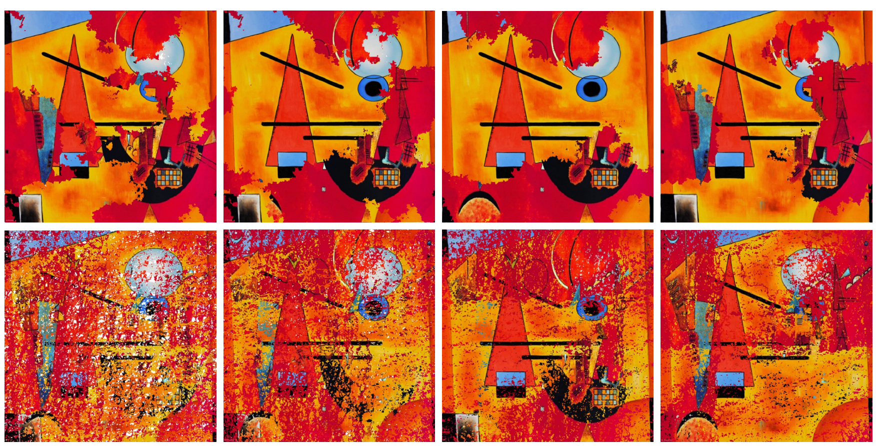


Fig. 4: Quasi-random animation with 2 Agents. Top row: symmetric sequences (images after $3\,600K$, $3\,900K$, $4\,000K$ and $4\,900K$ steps). Bottom row: asymmetric sequences (images after $300K$, $500K$, $1\,600K$ and $3\,700K$ steps)

We now consider 2 agents using the asymmetric sequences S^1 = (right, down, left, up, right) and S^2 = (up, left, down, right, up). Figure 4 (bottom row) shows the results. In contrast to the previous animation, the images obtained are a mixture of the two images of the agents in all parts of the image. This is due to the first agent painting its image in form of horizontal stripes and the second agent painting its image using vertical strips. As the agents move over the image, their current horizontal (for the first agent) and vertical (for the second agent) position determines which image of the agents shines through in which part at any given point in time of the animation.

4 Agents We now consider results that are achieved when using 4 agents. We obtain 3 additional images by changing the color spectrum of the image shown in Figure 2 (c) to blue, green, yellow. We use (c) and these 3 additional images as the images I^k painted by the agents. This means that we can identify the image of an agent by its color. The setting allows us to design animations where the underlying images have the same structure and only differ in terms of their color spectrum. The image in Figure 2 (d) is chosen as the starting image. The 4 agents start at positions $[m/4, m/4]$, $[m/4, 3n/4]$, $[3m/4, n/4]$ and $[3m/4, 3n/4]$, respectively.

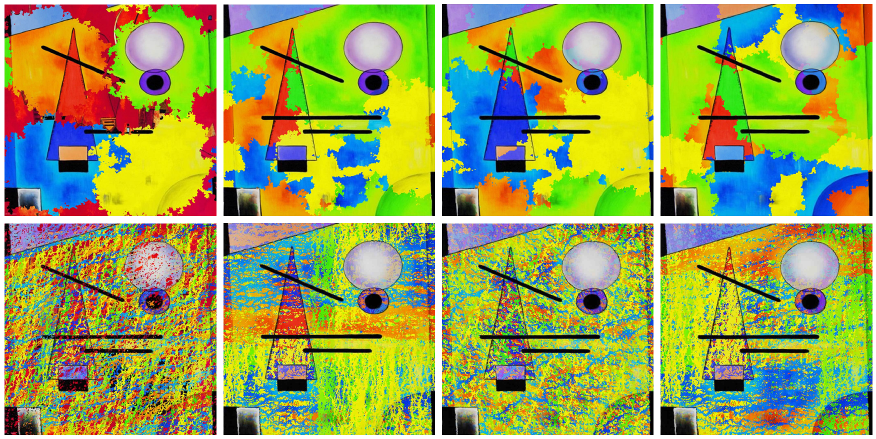


Fig. 5: Quasi-random animation with 4 Agents. Top row: symmetric sequences (images after $500K$, $2\,100K$, $2\,400K$ and $4\,200K$ steps). Bottom row: asymmetric sequences (images after $100K$, $1\,200K$, $4\,000K$ and $4\,300K$ steps)

Table 1: Preference for 1 agent

Image set	CRW	SQRW	AQRW
Column (a)	6	11	3
Column (b)	2	14	4
Column (c)	4	12	4
Column (d)	2	16	2
Total (%)	17.5	66.25	16.25

Table 2: Preferences: 2 and 4 agents

Image set	SQRW	AQRW
Image animation, 2 agents	91.25%	8.75%
Image animation, 4 agents	81.25%	18.75%

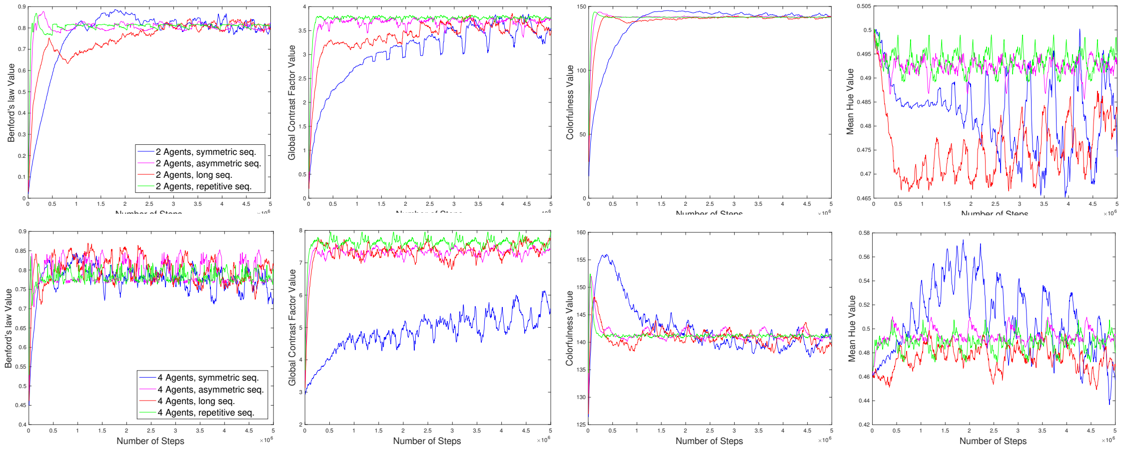
For the symmetric experiment, we use the sequences $S^1 = S^3$ = (right, down, left, up), and $S^2 = S^4$ = (up, left, down, right). The results in Figure 5 (top row) show that the images of the agents appear in circle-patch styles. It can be observed that parts of the different images painted by the agents appear at different time steps at different parts of the image. Thus, the algorithm allows the agents to paint part of their own image. The quasi-random image animation produces a visual pleasing result and combines the different parts of all images in a new abstract composition.

We now consider 4 agents using the asymmetric sequences $S^1 = S^3$ = (right, down, left, up, right), and $S^2 = S^4$ = (up, left, down, right, up). Figure 5 (bottom row) shows the results. In contrast to the symmetric sequences, the different images obtained provide an interesting mixture of the images of the different agents which is due to 2 agents moving horizontally and 2 agents moving vertically. There are no parts that can be clearly attribute to one of the agent's image. Instead of this, the animation produces interesting overlapping effects where the dominance of the colors (corresponding the agent's images) in the different parts of the image change over time.

Human validation

We conduct human subject experiments to evaluate the image results presented in the previous sections. The goal of the experiment is to determine user preferences for images obtained with the same number of agents. We carry out all human-based experiments using Amazon Mechanical Turk [4] with 20 participating users.

For our first investigation, we use four image sets according to the different columns shown in Figure 3. We present each human three images at a time corresponding to the different columns. Each column corresponds to roughly the same number of pixels transferred to the target image, and we want to know whether to prefer the results obtained by the classical random walk (CRW), the symmetric quasi-random walk (SQRW), or the asymmetric quasi-random walk (AQRW). The results given in Table 1 show that 66.25% of the time the human subjects selected image created with quasi-random walk agent with symmetric sequences. In contrast, the human subjects preferred 17.5% of the time images created with standard random walk methods. A similar percentage (16.25%) preferred the images created by the asymmetric quasi-random walk. Interestingly, the result show that the images obtained by the symmetric quasi-random walk are significantly preferred over images created by a standard random walk. We do the same investigations for the images shown in Figure 4 and 5 and compared the results obtained by symmetric quasi-random walk (SQRW) and asymmetric quasi-random walk (AQRW). The results are shown in Table 2. The results of the investigation where the users can choose between images created during quasi-random animation with symmetric and asymmetric sequences, reveal that the users prefer 91.25% of the time the images obtained with symmetric



(a) Benford's law (b) Global Contrast Factor (c) Colorfulness (d) Mean Hue

Fig. 6: Features during quasi-random animation

sequences. The result of the experiment for 4 agents shows that 81.25% of the time users select images generated with 4 agents and symmetric sequences. The experiments indicate of strong preference of the effect of the symmetric sequences in terms of aesthetic appearance of the images. In general, there were small differences in the user preferences, as in all four experiment the users prefer images created with agent/agents with symmetric sequences. The strongest positive response for asymmetric quasi-random walks of 18.75% can be observed for 4 agents.

Feature-based analysis

We now analyze the different animation processes obtained by our quasi-random animation algorithm. As the process creates different images over time, we measure the images created with respect to different artistic features. We consider features from the literature that have been used to measure artistic images [1, 16, 17, 22–24, 28] in order to enable an objective evaluation.

For our analysis, we examine all animation settings with 2 and 4 agents studied in the previous section. Furthermore we investigate the following additional settings for 2 agents: In the *long sequences* experiment, we consider S^1 = (right, down, left, up), S^2 = (right, down, left, up, right, down, left, up, right, down, left, up, right, down, left, up, right, up) and in the *repetitive sequences* experiment we use S^1 = (up, right, right, right), S^2 = (down, right, right, right). For 4 agents we consider the following two additional settings: In the *long sequences* experiment, we use $S^1 = S^3$ = (right, down, left, up), $S^2 = S^4$ = (right, down, left, up, right, down, left, up, right, down, left, up, right, down, left, up, right, up). For the *repetitive sequences* experiment, we use $S^1 = S^3$ = (up, right, right, right), and $S^2 = S^4$ = (down, right, right, right). The results for all animations according to the features *Benford's law*, *Global Contrast Factor*, *Colorfulness*, *Mean Hue* are shown in Figure 6. Here the top row shows the results for animations with 2 agents whereas the bottom row shows the results for 4 agents.

Subfigure (a) shows the results of the different animations for the feature *Benford's law* [5, 16]. It can be observed that the curve for 2 agents with symmetric sequences between $1\,500K$ and $2\,000K$ steps achieve the highest value. High values mean that results are less natural. Furthermore, the 2 agents with long sequences show the most natural results, as the values in comparison to the other agents are the smallest. During the animation, the values increase slowly. The values for 2 agents with repetitive and asymmetric sequences show a more stable behavior during the animation. The feature values for the animations that involves the use of 4 agents with long and asymmetric sequences show regularly appearing differences in values during the animation. They also have the overall the highest feature values among all animations. This means that those animations are less natural and explains the chaotic appearance of the animation. Finally, it can be observed that the curve for 4 agents with asymmetric sequences is clearly lower than those obtained for 4 agents with symmetric sequences.

In Subfigure (b), the results for the feature *Global Contrast Factor (GCF)* show a great difference between symmetric sequences and all other choices. *GCF* measures the richness of contrast and corresponds to the human perception of contrast in terms of visual attention. Matkovic et al. [22] show that humans prefer higher scores for *GCF* if they are looking at images. High values during the animation can be observed for the animations with 2 and 4 agents using repetitive sequences. For the animation with 2 and 4 agents with symmetric sequences, a characteristic zig-zag curve is obtained where the value slowly increases over time. In contrast to this, the animations with 2 and 4 agents using asymmetric, long and repetitive sequences have less variation in term of the feature values.

Subfigure (c) shows the results of the different animations for the feature *Colorfulness*. [16] computed perceived colorfulness to evaluate the effect that processing of the natural images has on their colour. [28] have used measurements of perceived colorfulness of website screenshots to developed computational models. It can be observed that all curves of the *Colorfulness* feature obtain values of around 140-145 after the animation has run for $1\,000K$ steps. Additionally, all curves have a very steady behavior except for the 4 agents with symmetric sequences where we can observe regular changes over time. Such changes make the process more interesting as it shows a larger variety within the animation.

In Subfigure (d), the results for the feature *Mean Hue* are shown. The animation with 4 agents using symmetric sequences reveal the largest variety in terms of this feature value. It can be observe that the curve is alternating between high and low values after approx. $2\,500K$ steps. This expresses the chaotic character of those agents movement during the animation. In contrast to this, the curve obtained during the animation for 2 agents with long sequences has a zig-zag pattern with tendency to increasing those values during the animation. On the other hand, all agents with repetitive and asymmetric sequences tend to have a continuous and regular behavior during the quasi-random animation.

Conclusions

Quasi-random methods allow to create interesting random behaviors and can easily be determined by a user who only has to set a few parameters within a quasi-random algorithm. We have presented a new approach to carry out image transition and animations. Our approach builds on theoretical insights of quasi-random walks and generalizes these concepts to multiple agents performing quasi-random walks for image transition and animation. The approach allows to create different forms of transition and animation by determining a sequence for each agent to be used in the quasi-random walk. Choosing these sequences is very easy and allows an artist to experiment with different complex behaviors resulting in interesting ways of carrying out transition and animation. Our human validation shows a clear preference of quasi-random walks over classical random walks. Furthermore, our feature-based analysis shows that the resulting animations exhibit quite different behavior in terms of important artistic features.

Acknowledgment

This research has been supported by the Australian Research Council (ARC) through grants DP140103400 and DP160102401.

References

1. Acebo, E.D., Sbert, M.: Benford's law for natural and synthetic images. In: Computational Aesthetics in Graphics, Visualization and Imaging. The Eurographics Association (2005)
2. Alexander, B., Hin, D., Neumann, A., Ull-Karim, S.: Evolving pictures in image transition space. In: To appear in proceedings of the International Conference on Neural Information Processing, ICONIP (2019)
3. Allen, B., Lippner, G., Chen, Y.T., Fotouhi, B., Momeni, N., Yau, S.T., Nowak, M.A.: Evolutionary dynamics on any population structure. Nature **544**(7649), 227 (2017)
4. Amazon: Mechanical Turk `https://www.mturk.com/`
5. Benford, F., Langmuir, I.: The law of anomalous numbers. Proceedings of the American Philosophical Society, American Philosophical Society (1938)
6. Cooper, J., Doerr, B., Spencer, J., Tardos, G.: Deterministic random walks on the integers. European Journal of Combinatorics **28**(8), 2072–2090 (2007)
7. Cooper, J., Spencer, J.: Simulating a random walk with constant error. Combinatorics, Probability & Computing **15**, 815–822 (2006)
8. Doerr, B., Friedrich, T.: Deterministic random walks on the two-dimensional grid. Combinatorics, Probability & Computing **18**(1-2), 123–144 (2009)
9. Droste, M.: Bauhaus, 1919-1933. Taschen (2002)
10. Dumitriu, I., Tetali, P., Winkler, P.: On playing golf with two balls. SIAM J. Discrete Math. **16**(4), 604–615 (2003)
11. Friedrich, T., Katzmann, M., Krohmer, A.: Unbounded discrepancy of deterministic random walks on grids. In: 26th ISAAC. pp. 212–222 (2015)
12. Friedrich, T., Sauerwald, T.: The cover time of deterministic random walks. Electr. J. Comb. **17**(1) (2010)
13. Gedeon, T.D.: Neural network for modeling esthetic selection. In: International Conference on Neural Information Processing, ICONIP 2007, Part II. pp. 666–674 (2007)
14. Grady, L.: Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1768–1783 (2006)
15. Greenfield, G.: Robot paintings evolved using simulated robots. In: Workshops on Applications of Evolutionary Computation. pp. 611–621. Springer (2006)
16. Hasler, D., Suesstrunk, S.E.: Measuring colorfulness in natural images. In: Electronic Imaging 2003. pp. 87–95 (2003)
17. den Heijer, E., Eiben, A.E.: Investigating aesthetic measures for unsupervised evolutionary art. Swarm and Evolutionary Computation **16**, 52–68 (2014)

18. Holroyd, A.E., Propp, J.G.: Rotor walks and Markov chains. Algorithmic Probability and Combinatorics **520**, 105–126 (2010)
19. Kleber, M.: Goldbug variations. The Mathematical Intelligencer **27**(1), 55–63 (2005)
20. Klee, P., Glaesemer, J.: Beiträge zur bildnerischen Formlehre. p. 3900. No. v. 2, Schwabe (1979)
21. Machado, P., Vinhas, A., Correia, J., Ekárt, A.: Evolving ambiguous images. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015. pp. 2473–2479. AAAI Press (2015)
22. Matkovic, K., Neumann, L., Neumann, A., Psik, T., Purgathofer, W.: Global contrast factor - a new approach to image contrast. Computational Aesthetics **2005**, 159–168 (2005)
23. Neumann, A., Alexander, B., Neumann, F.: Evolutionary image transition using random walks. In: Computational Intelligence in Music, Sound, Art and Design - 6th International Conference, EvoMUSART 2017. vol. 10198, pp. 230–245. Springer, Cham (2017)
24. Neumann, A., Neumann, F.: On the use of colour-based segmentation in evolutionary image composition. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2018. pp. 1–8. IEEE (2018)
25. Neumann, A., Pyromallis, C., Alexander, B.: Evolution of images with diversity and constraints using a generative adversarial network. In: International Conference on Neural Information Processing, ICONIP 2018, Proceedings, Part VI. pp. 452–465 (2018)
26. Neumann, A., Szpak, Z.L., Chojnacki, W., Neumann, F.: Evolutionary image composition using feature covariance matrices. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017. pp. 817–824 (2017)
27. Priezzhev, V.B., Dhar, D., Dhar, A., Krishnamurthy, S.: Eulerian walkers as a model of self-organized criticality. Phys. Rev. Lett. **77**, 5079–5082 (1996)
28. Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., Gajos, K.Z.: Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In: Conference on Human Factors in Computing Systems CHI. pp. 2049–2058. ACM (2013)
29. Sims, K.: Artificial evolution for computer graphics. In: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991. pp. 319–328 (1991)
30. Teytaud, O.: Quasi-random numbers improve the CMA-ES on the BBOB testbed. In: Artificial Evolution - 12th International Conference, Evolution Artificielle, EA 2015. pp. 58–70 (2015)
31. Teytaud, O., Fournier, H.: When does quasi-random work? In: Proceedings of the Parallel Problem Solving from Nature, PPSN 2008. pp. 325–336 (2008)
32. Teytaud, O., Gelly, S.: DCMA: yet another derandomization in covariance-matrix-adaptation. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2007. pp. 955–963 (2007)
33. Todd, S., Latham, W.: Evolutionary art and computers. Academic Press, Inc. (1992)
34. Unemi, T.: SBArt4 for an automatic evolutionary art. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012. pp. 1–8. IEEE (2012)
35. Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Distributed covering by ant-robots using evaporating traces. IEEE Transactions on Robotics and Automation **15**(5), 918–933 (1999)
36. Zentrum: Paul Klee `https://www.zpk.org/en/`