

A Comparative Analysis of Detecting Symmetries in Toroidal Topology

Mohammad Ali Javaheri Javid and Wajdi Alghamdi and Robert Zimmer and Mohammad Majid al-Rifaie

Abstract In late 1940's and with the introduction of cellular automata, various types of problems in computer science and other multidisciplinary fields have started utilising this new technique. The generative capabilities of cellular automata have been used for simulating various natural, physical and chemical phenomena. Aside from these applications, the lattice grid of cellular automata has been providing a by-product interface to generate graphical patterns for digital art creation. One notable aspect of cellular automata is symmetry, detecting of which is often a difficult task and computationally expensive. This paper uses a swarm intelligence algorithm – Stochastic Diffusion Search – to extend and generalise previous works and detect partial symmetries in cellular automata generated patterns. The newly proposed technique tailored to address the spatially-independent symmetry problem is also capable of identifying the absolute point of symmetry (where symmetry holds from all perspectives) in a given pattern. Therefore, along with partially symmetric areas, the centre of symmetry is highlighted through the convergence of the agents of the swarm intelligence algorithm. Additionally this paper proposes the use of *entropy* and *information gain* measure as a complementary tool in order to offer insight into the structure of the input cellular automata generated images. It is shown that using these technique provides a comprehensive picture about both the struc-

Mohammad Ali Javaheri Javid
Department of Computing, Goldsmiths, University of London
London SE14 6NW, United Kingdom e-mail: m.javaheri@gold.ac.uk

Wajdi Alghamdi
Department of Computing, Goldsmiths, University of London
London SE14 6NW, United Kingdom e-mail: walgh001@gold.ac.uk

Robert Zimmer
Department of Computing, Goldsmiths, University of London
London SE14 6NW, United Kingdom e-mail: r.zimmer@gold.ac.uk

Mohammad Majid al-Rifaie
Department of Computing, Goldsmiths, University of London
London SE14 6NW, United Kingdom e-mail: m.majid@gold.ac.uk

ture of the images as well as the presence of any complete or spatially-independent symmetries. These technique are potentially applicable in the domain of aesthetic evaluation where symmetry is one of the measures.

1 Introduction

Creating aesthetically pleasing images has been investigated by many researches in the context of evolutionary computing, including the Bimorphs of Dawkins [18], Mutator of Latham [44], and Virtual Creatures of Sims [42]. Although some impressive results have been achieved, there still remains problems with the aesthetic selection. According to [30], first, the subjective comparison process, even for a small number of phenotypes, is slow and forms a bottleneck in the evolutionary process. Human users would take hours to evaluate many successive generations that in an automated system could be performed in a matter of seconds. Secondly, genotype-phenotype mappings are often not linear or uniform. That is, a minor change in genotype may produce a radical change in phenotype. Such non-uniformities are particularly common in tree or graph based genotype representations such as in evolutionary programming, where changes to nodes can have a radical effect on the resultant phenotype. In this study we approach the problem in the framework of dynamical systems and define a criterion for aesthetic selection in terms of its association with symmetry. The association of aesthetics and symmetry has been investigated from different points of view.

In this work, a brief account on cellular automata is presented, followed by a section on symmetry and its significance in aesthetics. Subsequently, the newly proposed measure of information gain is described and the main experiments related to this measure are included. Then a swarm intelligence algorithm – Stochastic Diffusion Search (SDS) – is explained, highlighting its main features, including its unique partial function evaluation aspect. Afterwards, the application of the algorithm in detecting points of symmetry is detailed, illustrating the performance of the method proposed. At last but not least, a comparison is provided between information gain and Stochastic Diffusion Search emphasising their strength and weaknesses. The paper is then concluded with a summary of the finding and the suggested future research.

2 Cellular Automata

Cellular automata (CA) are one of the early bio-inspired systems invented by von Neumann and Ulam in the late 1940s to study the logic of self-reproduction in a material independent framework. CA are known for their capabilities in exhibiting complex behaviour from iterative application of simple rules. In this section mathematical formulations of 2D CA are provided which are specific for the purpose

of this paper and for the rest of the paper all the notions will be referring to these formulations.

Definition 1. A deterministic finite automaton is formally defined [28] as a quintuple of \mathcal{M} such that:

$$\mathcal{M} = \langle Q, \Sigma, \delta, q_0, F \rangle \quad (1)$$

1. Q is a finite set of *states*,
2. Σ is a finite set of symbols as *input alphabet*,
3. $\delta : Q \times \Sigma \mapsto Q$ is the *state transition function*,
4. $q_0 \in Q$ is the *start* or *initial state*,
5. $F \subseteq Q$ is a set of *accepting* or *final states*.

The state transaction function δ determines the transitions from one state to another state. It takes two arguments as $q \in Q$ and an input symbol $a \in \Sigma$ then maps them to a state $q_1 \in Q$ (i.e. $\delta(q, a) = q_1$).

Definition 2. A lattice (L) is a regular tiling of a space by a unit cell.

The *Euclidean* plane is considered so the lattice L is over Z^2 . Lattices can have *square*, *hexagonal* or *triangle* for their unit cells. A lattice can be *infinite* with *open boundary conditions* or *finite* with *periodic boundary conditions*. A finite lattice with periodic boundary conditions where the opposite borders (up and down with left and right) are connected, forms a virtual torus shape (Fig.1).

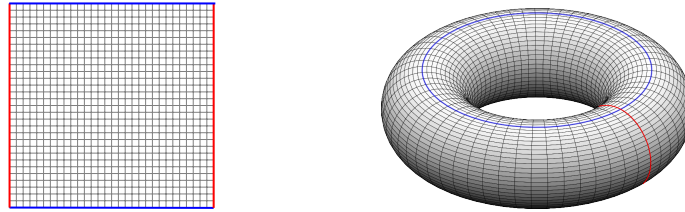


Fig. 1 The formation of virtual torus shape in a lattice with periodic boundary conditions

Definition 3. A cellular automaton is a regular tiling of a lattice with uniform deterministic finite state automata.

A cellular automaton \mathcal{A} is specified by a quadruple $\langle L, S, N, f \rangle$ where:

1. L is a finite square lattice of cells (i, j) .
2. $S = \{1, 2, \dots, k\}$ is set of states. Each cell (i, j) in L has a state $s \in S$.

3. N is neighbourhood, as specified by a set of lattice vectors $\{e_a\}$, $a = 1, 2, \dots, N$. The neighbourhood of cell $r = (i, j)$ is $\{r + e_1, r + e_2, \dots, r + e_N\}$. A cell is considered to be in its own neighbourhood so that one of $\{e_a\}$ is the zero vector $(0, 0)$. With an economy of notation, the cells in the neighbourhood of (i, j) can be numbered from 1 to N ; the neighbourhood states of (i, j) can therefore be denoted (s_1, s_2, \dots, s_N) . Periodic boundary conditions are applied at the edges of the lattice so that complete neighbourhoods exist for every cell in L .
4. f is the update rule. f computes the state $s_1(t+1)$ of a given cell from the states (s_1, s_2, \dots, s_N) of cells in its neighbourhood: $s_1(t+1) = f(s_1, s_2, \dots, s_N)$. A quiescent state s_q satisfies $f(s_q, s_q, \dots, s_q) = s_q$.

Remark 1. There are two common neighbourhoods; a five-cell von Neumann neighbourhood $\{(0, 0), (\pm 1, 0), (0, \pm 1)\}$ and a nine-cell Moore neighbourhood: $\{(0, 0), (\pm 1, 0), (0, \pm 1), (\pm 1, \pm 1)\}$.

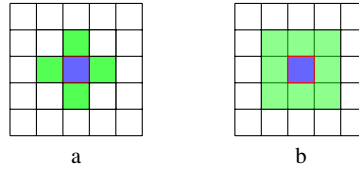


Fig. 2 von Neumann (a) and Moore (b) neighbourhood templates

The collection of states for all cells in L is known as a configuration C . The global rule F maps the whole automaton forward in time; it is the synchronous application of f to each cell. The behaviour of a particular \mathcal{A} is the sequence $c^0, c^1, c^2, \dots, c^{T-1}$, where c^0 is the initial configuration (IC) at $t = 0$.

CA behaviour is sensitive to the IC and to L, S, N and f . The behaviour is generally nonlinear and sometimes very complex; no single mathematical analysis can describe, or even estimate, the behaviour of an arbitrary automaton. The vast size of the rule space, and the fact that this rule space is unstructured, mean that knowledge of the behaviour a particular cellular automaton, or even of a set of automata, gives no insight into the behaviour of any other CA. In the lack of any practical model to predict the behaviour of a cellular automaton, the only feasible method is to run simulations.

3 Symmetry and Aesthetic

Symmetry or having proportionality and balance is an important element of aesthetics. The association of aesthetics and symmetry has been investigated extensively in the literature. A study to investigate the effect of symmetry on interface judgements, and relationship between a higher symmetry value and aesthetic appeal for

the basic imagery, showed that subjects preferred symmetric over non-symmetric images [11]. Further studies found that if symmetry is present in the face or the body, an individual is judged as being relatively more attractive and if the body is asymmetric the face is rated unattractive, even if the person doing the rating never sees the body [37, 20]. Symmetry plays a crucial role in theories of perception and is even considered a fundamental structuring principle of cognition [27]. In the Gestalt school of psychology things [objects] are affected by where they are and by what surrounds them... so that things [objects] are better described as more than the sum of their parts [12].

The Gestalt principles emphasise the holistic nature of perception where recognition is inferred, during visual perception, more by the properties of an image as a whole, rather than its individual parts [23]. Thus, during the recognition process elements in an image are grouped from parts to whole based on Gestalt principles of perception such as proximity, parallelism, closure, symmetry, and continuation [34]. In particular, symmetric objects are more readily perceived [16]. It is not surprising that we humans find sensory delight in symmetry, given the world in which we evolved. In our world the animals that have interested us and our ancestors (as prey, menace, or mate) are overwhelming symmetric along at least one axis [36].

3.1 Evolutionary and Computational approaches

Evolutionary psychologists examine physical appearances like as symmetry, and perceived level of aesthetics as an indirect measure in mate selection [33, 32]. In this view symmetrical faces are examined as more attractive faces. In other words symmetry is positively linked with both psychological and physiological health indicators [38]. In geometry symmetrical shapes are produced by applying four operations of translations, rotations, reflections, and glide reflections. However developing computational methods which generate symmetrical patterns is still a challenge since it has to connect abstract mathematics with the noisy, imperfect, real world; and few computational tools exist for dealing with real-world symmetries [29]. Applying evolutionary algorithms to produce symmetrical forms leaves the formulation of fitness functions, which generate and select symmetrical phenotypes, to be addressed. Lewis describes two strategies in evolutionary algorithms approach for generating and selecting symmetrical forms: “A common approach is to hope for properties like symmetry to gradually emerge by selecting for them. Another strategy is to build in symmetry functions which sometimes activate, appearing suddenly. However this leads to a lack of control, as offspring resulting from slight mutations (i.e., small steps in the solution space) bear little resemblance to their ancestors [26]”.

There are several algorithms designed to detect exact symmetry in images; amongst which, and in the case of a collection of n points in the plane, Atallah describes an algorithm for enumerating all axes of symmetry under reflection of a planar shape [8]. In another work, Wolter et al. give exact algorithms, based on string

matching, for the detection of symmetries of point clouds, polygons, and polyhedra [48]. These algorithms are often impractical due to their sensitivity to noise and high computational expense, because of their restricted nature to exact symmetries.

In terms of approximate symmetry there are two broad categories: the first defines approximate symmetry by an infimum of a continuous distance function quantifying how similar is a shape to its transformed version. One of the works introduced in this category is reported in [49]. The second approach aims to address the computational complexity issue by translating the search into a proxy domain, realising that the set of admissible symmetries is sparse in the transformation space; among the examples of this type of work are [43, 35, 31]. Other variations of symmetry detection have been proposed that fall within the broad above-mentioned categories (e.g. [25] which addresses reflection symmetry detection problem or [50] focusing on boolean functions, which is based on *information theory* [41] that detects symmetry by comparison of two-variable cofactors).

4 Entropy and Information Gain Measure

The *information theory* was an attempt to address a reliable communication over an unreliable channel [39]. Entropy is the core of this theory [17]. Let \mathcal{X} be discrete alphabet, X a discrete random variable, $x \in \mathcal{X}$ a particular value of X and $P(x)$ the probability of x . Then the entropy, $H(X)$, is:

$$H(X) = - \sum_{x \in \mathcal{X}} P(x) \log_2 P(x) \quad (2)$$

The quantity H is the average uncertainty in bits, $\log_2(\frac{1}{p})$ associated with X . Entropy can also be interpreted as the average amount of information needed to describe X . The value of entropy is always non-negative and reaches its maximum for the uniform distribution, $\log_2(|\mathcal{X}|)$:

$$0 \leq H \leq \log_2(|\mathcal{X}|) \quad (3)$$

The lower bound of relation (3) corresponds to a deterministic variable (no uncertainty) and the upper bound corresponds to a maximum uncertainty associated with a random variable. Despite the dominance of entropy as a measure of order and complexity, it fails to reflect on structural characteristics of 2D patterns. The main reason for this drawback is that it only reflects on the distribution of the symbols, and not on their spatial arrangements.

Considering our intuitive perception of complexity and structural characteristics of 2D patterns, a complexity measure must be bounded by two extreme points of complete order and disorder. It is reasonable to assume that *regular structures*, *irregular structures* and *structureless* patterns lie along between these extremes, as illustrated in Fig. 3.

$$\text{order} \xleftarrow{\text{regular structure|irregular structure|structureless}} \text{disorder}$$

Fig. 3 The spectrum of spatial complexity.

A complete regular structure is a pattern of high symmetry, an irregular structure is a pattern with some sort of structure but not as regular as a fully symmetrical pattern and finally a structureless pattern is a random arrangement of elements [22].

A measure introduced in [9, 45, 6] and known as *information gain*, has been proposed as a means of characterising the complexity of dynamical systems and of 2D patterns. It measures the amount of information gained in bits when specifying the value, x , of a random variable X given knowledge of the value, y , of another random variable Y ,

$$G_{x,y} = -\log_2 P(x|y). \quad (4)$$

$P(x|y)$ is the conditional probability of a state x conditioned on the state y . Then the *mean information gain* (MIG), $\bar{G}_{X,Y}$, is the average amount of information gain from the description of the all possible states of Y :

$$\bar{G}_{X,Y} = \sum_{x,y} P(x,y) G_{x,y} = -\sum_{x,y} P(x,y) \log_2 P(x|y) \quad (5)$$

where $P(x,y)$ is the joint probability, $\text{prob}(X=x, Y=y)$. \bar{G} is also known as the conditional entropy, $H(X|Y)$ [17]. Conditional entropy is the reduction in uncertainty of the joint distribution of X and Y given knowledge of Y , $H(X|Y) = H(X,Y) - H(Y)$. The lower and upper bounds of $\bar{G}_{X,Y}$ are

$$0 \leq \bar{G}_{X,Y} \leq \log_2 |\mathcal{X}|. \quad (6)$$

Definition 4. A structural complexity measure G , of a cellular automaton configuration is the sum of the mean information gains of cells having homogeneous/heterogeneous neighbouring cells over 2D lattice.

For a cellular automaton configuration, \bar{G} can be calculated by considering the distribution of cell states over pairs of cells r, s ,

$$\bar{G}_{r,s} = -\sum_{s_r, s_s} P(s_r, s_s) \log_2 P(s_r, s_s) \quad (7)$$

where s_r, s_s are the states at r and s . Since $|\mathcal{S}| = N$, $\bar{G}_{r,s}$ is a value in $[0, N]$.

The relative positions for non-edge cells are given by matrix M :

$$M = \begin{bmatrix} (i-1, j+1) & (i, j+1) & (i+1, j+1) \\ (i-1, j) & (i, j) & (i+1, j) \\ (i-1, j-1) & (i, j-1) & (i+1, j-1) \end{bmatrix}. \quad (8)$$

Correlations between cells on opposing lattice edges are not considered. The result of this edge condition is that $G_{i+1,j}$ is not necessarily equal to $\bar{G}_{i-1,j}$.

In addition the differences between the horizontal (vertical) and two diagonal mean information rates reveal left/right (up/down), primary and secondary orientation of 2D patterns.

So the sequence of generated configurations by a multi-state 2D cellular automaton can be analysed by the differences between the vertical ($i, j \pm 1$), horizontal ($i \pm 1, j$), primary diagonal (P_d) and secondary diagonal (S_d) mean information gains by

$$\Delta \bar{G}_{i,j\pm 1} = |\bar{G}_{i,j+1} - \bar{G}_{i,j-1}| \quad (9)$$

$$\Delta \bar{G}_{i\pm 1,j} = |\bar{G}_{i-1,j} - \bar{G}_{i+1,j}| \quad (10)$$

$$\Delta \bar{G}_{P_d} = |\bar{G}_{i-1,j+1} - \bar{G}_{i+1,j-1}| \quad (11)$$

$$\Delta \bar{G}_{S_d} = |\bar{G}_{i+1,j+1} - \bar{G}_{i-1,j-1}| \quad (12)$$

The advantages of \bar{G} over H in discriminating structurally different patterns is illustrated in Fig. 4 where the 4-state 2D CA configurations with the complete symmetrical (Fig. 4a), the partially structured (Fig. 4b) and the structureless and random (Fig. 4c) patterns are evaluated. As it is evident, the measures of H are identical for structurally different patterns, however, the measure of \bar{G} s and $\Delta \bar{G}$ s are reflecting not only the complexity of patterns but their spatial arrangements (i.e. symmetries) too. Fig. 4 clearly demonstrates the drawbacks of entropy to discriminate structurally different 2D patterns. In other words, entropy is invariant to spatial rearrangement of composing elements. This is in contrast to our intuitive perception of the complexity of patterns and is problematic for the purpose of measuring the order and complexity of multi-state 2D CA configurations for aesthetic evaluation.

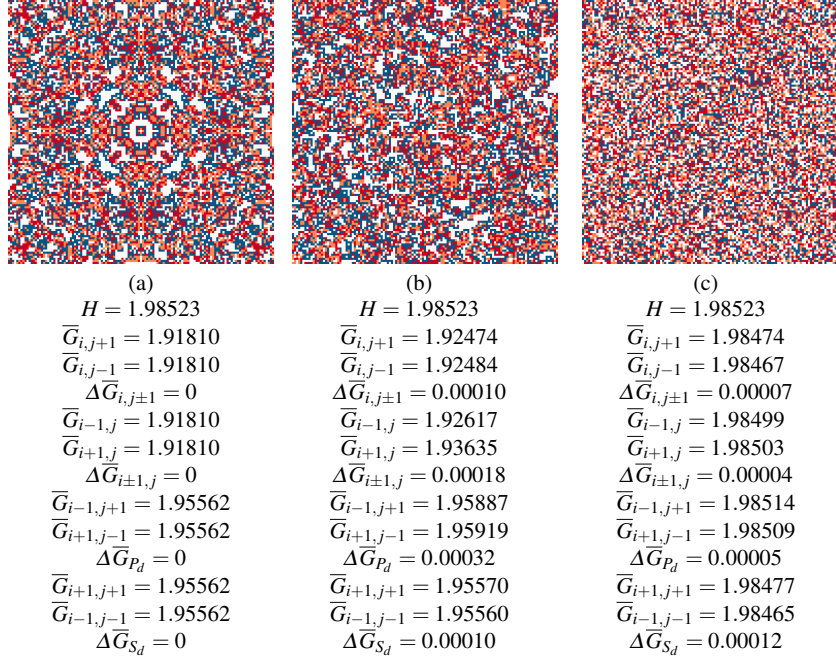


Fig. 4 The comparison of H with measures of $\bar{G}_{i,j}$ for structurally different 4-state CA configurations.

In the research presented in this paper, two modalities of the same algorithms are used to detect absolute symmetry in an input image if present, as well partial (sub-) symmetries.

The next section explains the swarm intelligence algorithm which will be used in detecting symmetrical patterns.

5 Stochastic Diffusion Search

The swarm intelligence algorithm used in this work is Stochastic Diffusion Search (SDS) [13, 3] which is a probabilistic approach for solving best-fit pattern recognition and matching problems. SDS, as a multi-agent population-based global search and optimisation algorithm, is a distributed mode of computation utilising interaction between simple agents. Its computational roots stem from Geoff Hinton's interest 3D object classification and mapping. See [21] for Hinton's work and [14] for the connection between Hinton mapping and SDS. SDS algorithm has been used in various fields including optimisation, generative arts and medical imaging (e.g. [4, 5, 1, 2]). SDS, contrary to many swarm intelligence algorithms, has a strong

mathematical framework describing its behaviour and convergence. The full mathematical model and proof of SDS convergence are elaborated in [3].

5.1 SDS Architecture

Similar to other swarm intelligence algorithms, SDS commences a search or optimisation by initialising its population. In any SDS search, each agent maintains a hypothesis, h , defining a possible problem solution. After initialisation, the two phases of SDS – Test and Diffusion phases – are followed (see Algorithm 1 for a high-level description of SDS).

In the test phase, SDS checks whether the agent hypothesis is successful or not by performing a partial hypothesis evaluation and returning a domain independent boolean value. Later in the iteration, contingent on the strategy employed, successful hypotheses diffuse across the population and in this way information on potentially good solutions spreads throughout the entire population of agents.

In other words, in the Test phase, each agent performs *partial function evaluation*, pFE , which is some function of the agent's hypothesis, $pFE = f(h)$; and in the Diffusion phase, each agent recruits another agent for interaction and potential communication of hypothesis.

Algorithm 1 SDS Algorithm

```

01: Initialising agents
02: While (stopping condition is not met)
03:   Testing hypotheses
04:     Determining agents status (active/inactive)
05:   Diffusing hypotheses
06:     Exchanging of information
07: End While

```

5.2 Standard SDS and Passive Recruitment

In standard SDS, *passive recruitment mode* is employed. In this mode, if the agent is inactive, a second agent is randomly selected for diffusion; if the second agent is active, its hypothesis is communicated (*diffused*) to the inactive one. Otherwise there is no flow of information between agents; instead a completely new hypothesis is generated for the first inactive agent at random (see Algorithm 2). Therefore, recruitment is not the responsibility of the active agents. Higher rate of inactivity boosts exploration, whereas a lower rate biases the performance towards exploitation. Details of the test phase and the fitness function is described later in this paper.

Algorithm 2 Passive Recruitment Mode

```

01: For each agent ag
02:   If ( !ag.isActive )
03:     r_ag = pick a random agent
04:     If ( r_ag.isActive )
05:       ag.hypothesis = r_ag.hypothesis
06:     Else
07:       ag.hypothesis = generate random hypothesis
08:     End If
09: End For

```

5.3 Partial Function Evaluation

One of the concerns associated with many optimisation algorithms (e.g. Genetic Algorithm, Particle Swarm Optimisation and etc.) is the repetitive evaluation of a computationally expensive fitness functions. In some applications, such as tracking a rapidly moving object or generation of CA patterns, the repetitive function evaluation significantly increases the computational cost of the algorithm. Therefore, in addition to reducing the number of function evaluations, other measures can be used in an attempt to reduce the computations carried out during the evaluation of each possible solution, as part of the overall optimisation (or search) processes.

The commonly used benchmarks for evaluating the performance of swarm intelligence algorithms are typically small in terms of their objective functions computational costs [19, 47], which is often not the case in real-world applications (examples of costly evaluation functions are seismic data interpretation, selection of sites for the transmission infrastructure of wireless communication networks and radio wave propagation calculations of one site, etc.).

Costly objective function evaluations have been investigated under different conditions [24] and the following two broad approaches have been proposed to reduce the cost of function evaluations:

- The first is to estimate the fitness by taking into account the fitness of the neighbouring elements, the former generations or the fitness of the same element through statistical techniques introduced in [15].
- In the second approach, the costly fitness function is substituted with a cheaper, approximate fitness function.

When agents are about to converge, the original fitness function can be used for evaluation to check the validity of the convergence [24].

The approach that the standard SDS algorithm uses is similar to the second method. Many fitness functions are decomposable to components that can be evaluated separately. During the test phase of SDS, in partial function evaluation (pFE , which is some function of the agent's hypothesis, $pFE = f(h)$), the evaluation of one or more of the components may provide partial information to guide the subsequent optimisation process.

In other words, instead of evaluating the hypothesis in its entirety, part of it, which is called *micro-feature*, is selected and evaluated accordingly. Therefore, during the test phase, only the randomly selected micro-features of the hypotheses are evaluated and the status of each agent is thus determined. Thus, if the micro-feature of each hypothesis consists of, say, $\frac{1}{10}$ of the entire hypothesis, the computational expense for the evaluation process of each hypothesis would be $\frac{9}{10}$ computationally cheaper.

Next, details of the process through which SDS performs its spatial-independent symmetry detection is presented.

6 Experiments

This section explains the design of the experiments conducted along with the results of applying SDS to identify partial or full symmetries on the cellular automata generated patterns. The inputs to the system are sample patterns used as proof of principle to show the functionality of the method; afterwards, some real world cellular automata generated patterns are fed in the system to evaluate the overall performance of the algorithm in detecting the aforementioned types of symmetries.

In order to adopt SDS to identify symmetries, the following important considerations are taken into account:

- the search space comprises of the entire cells on the grid
- SDS hypothesis is a cell index. For instance, in a 5×5 grid, the coordinate $(2, 2)$ could be the hypothesis and micro-features¹ can be selected by specifying the x_d and y_d distances from the hypothesis; therefore, assuming the (x_d, y_d) distance is $(2, 0)$, this micro-feature should be compared against its corresponding element with $(-2, 0)$ distance from the hypothesis.
- the environment in cellular automata is torus, which means if moving downwards along the search space when we reach the last row, the next row to be visited is the top row. The same is applicable when moving between columns (see Fig. 1 shows the 2D representation of the cellular automata and its real structure as torus).

The patterns in Figs. 5 show the hypothesis $(3, 2)$ and the various possible micro-features, some of which resulting in the hypothesis' status to be *true* while some others lead to the hypothesis' status to be *false*. The hypothesis in these figures are set to be $(3, 2)$ and various micro-features are selected to test the symmetry of the pattern along various axes of symmetry. The torus structure of cellular automata is demonstrated in the choice of some of the corresponding micro-features; see, for example, Fig. 5 top-right corner, where the micro-feature is chosen at $(-1, -1)$ distance. Thus the corresponding cell is chosen at $(1, 1)$ distance from the hypothesis,

¹ Micro-features are used in the test phase of SDS to determine the status of the agent (i.e. active or inactive).

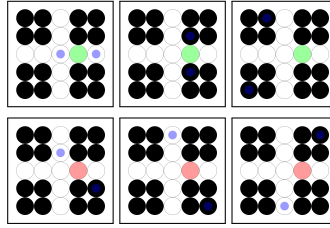


Fig. 5 Sample hypothesis set to be $(3,2)$; active hypotheses are shown in green and the inactive ones are displayed in red; the selected micro-features are highlighted in blue

which means moving out of the 2D canvas from the right border and entering again from the left.

The process through which SDS commences with the initialisation phase and then cycle through the two phases and test and diffusion is explained next.

6.1 Initialisation phase

During the initialisation phase each one of the agents in the population is assigned a hypothesis which is a random (x,y) coordinate from the search space. Additionally, the status of all agents are initially set to *false*.

6.2 Test phase

In the test phase, each agent, which is already allocated an (x,y) hypothesis, picks a random x_d and y_d distances from the hypothesis cell as its micro-feature; the randomly selected micro-feature is then compared against corresponding mirrored cells to checks if the mirror cell has the same value. If the values are the same, the status of the agent is set to *true*, otherwise *false*

6.3 Diffusion phase

The process in the diffusion phase is the same as the one detailed in the algorithm description where each inactive agent picks an agent randomly from the population; if the randomly selected agent is active, the inactive agent adopts the hypothesis of the active agent (i.e. the (x,y) coordinate), otherwise the inactive agent picks a random coordinate from the search space.

After n number of iterations agents converge on the (x,y) coordinates with the most symmetrical quality.

6.4 Experiments and Discussion

One of the main features of SDS is partial function evaluation which here manifests itself in: each time comparing one cell on one side of the symmetrical point to its corresponding cell on the other side. Therefore even when an agent is active, in the next iteration it picks another micro-feature and checks the point from “a different perspective” to ensure that the symmetry still holds. In other words, using this approach, the algorithm allocate its resources “wisely” and repeatedly tests the already maintained points of interest against any asymmetrical discovery.

For the experiments reported in this work, the population size is empirically calculated using the following formula:

$$pSize = \frac{w^2}{4} \quad (13)$$

where pSize is the population size and w is the width of the search space. Using this set-up, the agents land on fourth of the search space; therefore for a 5×5 search space, $pSize = 6$.

As illustrated in the figures, some agents became active on (x,y) coordinates which do not represent the full four-fold symmetry; these agents will eventually pick different micro-features in the next iterations and become inactive; consequently, when they are inactive, they need to choose random agents; given that the number of active agents on the centre of symmetry increases over time (thanks to the diffusion phase), it is likely that an active agent is chosen. This would lead to the inactive agents picking micro-features from the centre of symmetry in their next iterations and become/stay active. Note that in these experiments, alpha is used for the transparency of the agents’ colour; therefore as shown on the figures, the cell with the largest number of active agents can be distinguished from others.

There are occasions when more than one centre of symmetry exists, or there exist some partial symmetries (also called sub-symmetry) in the image along with full centre of symmetry; in this case another flavour of the recruitment strategy is deployed which is called *context-sensitive mechanism*. This strategy frees up some of the agents who are active and share the same hypothesis, and therefore allows the algorithm to constantly check for traces of symmetry in the input pattern.

Algorithm 3 Context Sensitive Mechanism

```

01: If ( ag.activity )
02:   r_ag = pick a random agent
03:   If ( r_ag.activity AND
04:     ag.getHypothesis == r_ag.getHypothesis )
05:     ag.setActivity ( false )
06:     ag.setHypothesis ( randomHypothesis )
07:   End If
08: End If

```

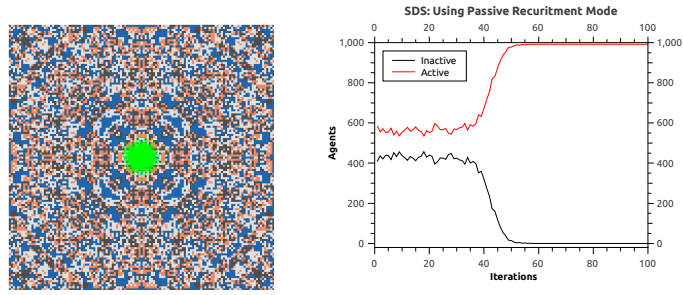


Fig. 6 Passive recruitment mode

In other words, the use of context sensitive mechanism biases the search towards global exploration. Thus, if an active agent randomly chooses another active agent that maintains the same hypothesis, the selecting agent is set inactive and adopts a random hypothesis. This mechanism frees up some of the resources in order to have a wider exploration throughout the search space as well as preventing cluster size from overgrowing; this process goes on while ensuring the formation of large clusters in case there exists a perfect match or good sub-optimal solutions (see Algorithm 3).

The next set of experiments use some complex patterns, generated by cellular automata techniques. Initially an experiment is run that utilises the *passive recruitment mode* without the introduced *context-sensitive mechanism* and later, the impact of *context-sensitivity* is discussed.

The graph in Fig. 6 illustrate the behaviour of the agents' activities; this graph demonstrates that after the initialisation phase, the number of active and inactive agents are balanced; however over time, and due to the presence of a centre of symmetry in the pattern, the number of active agents increases and the number of inactive agents decreases. Therefore, ultimately, once the absolute center of symmetry (where symmetry holds irrespective of which micro-feature is chosen) is identified, the entire agent population becomes active and the number of inactive agents drops to zero.

Using context-sensitive mechanism, the graph in Fig. 7 illustrates the behaviour of SDS algorithm using this mode, where the populations are biased towards global exploration. In this graph, while the increase of active agents and the decrease of inactive agents are visible, it is evident that there are always agents which are released back from the centre of symmetry to the search space to explore the possibility of the *presence of further (sub-) symmetrical points*. This feature is particularly useful in dynamic environment, and where there are more than one absolute points of symmetry (the next experiment uses an input image with a few points of symmetries). The figure shows many active (green) and inactive (red) agents throughout the search space and the graph illustrates that the number of inactive agents never drops to zero.

The next experiment, which uses a more symmetrically complex CA-generated pattern, demonstrates the crucial difference between using SDS with and without

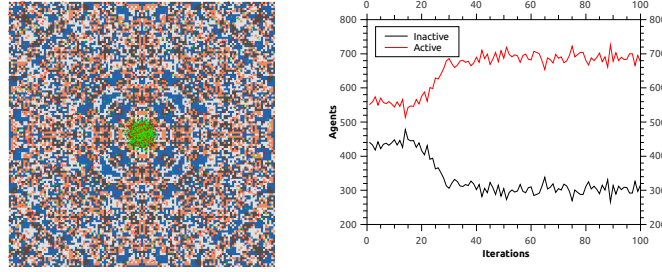


Fig. 7 Context sensitive mechanism

the context-sensitive mechanism. As stated before, context-sensitive mechanism reduces the greediness of the agents and allows the agents to explore the search space for any undetected symmetry, while the pure passive mechanism is greedy and once it finds the absolute point of symmetry (where symmetry holds no matter which micro-feature is picked), it gradually pulls all the agents towards the point and stops them from locating possible partial symmetries in the canvas.

The new input to be used in this experiment has two identically CA grown patterns one on the top-left corner and another on the bottom-left corner. When running the SDS algorithm, it becomes clear that the passive recruitment strategy (see Fig. 8) initially locates two points of symmetry (at $n = 100$ iterations), however later (at $n = 200$ iterations) all agents are drawn towards the absolute point of symmetry (note that the search spaces in cellular automata are torus).

Using the context sensitive approach, the largest partial symmetries are also identified and highlighted (see Fig. 9). The graphs at the bottom of Figs. 8 and 9 clearly show the behaviour of the agents in both modes. As displayed in the graph of Fig. 9, while the number of active and inactive agents are distinguishably far from one another, yet it is shown that the number of active agents does not reach the maximum possible² and the number of inactive agents does not drop to zero. This mechanism insures the identification of other (sub-) symmetrical points in the input. Therefore, depending on the functionalities needed, either of these approaches could be used.

Another observation to be expanded in the future work is the direct proportionality of the agents' activity to the 'strength' of the symmetry. Therefore, while context-sensitive mechanism finds partial symmetries, it is able to 'rank' the various clusters of agents which are formed over the pattern. This could lead to introducing the ratio of active/inactive agents as a measure for *order* and *complexity* along with Shannon's entropy [40] and information gain [10, 7, 46] which are among the very few measures used in cellular automata for measuring symmetry.

The swam intelligence, in neither of the modalities (i.e. passive mode and passive with context-free mode) technique has classified 4 (b) and 4 (c) as unstructured. as they do not possess dominant centre(s) of symmetries.

² Given the size of the search space is $ssSize = 129$, the population size for this pattern is $pSize = \frac{129^2}{4} = 4,160$

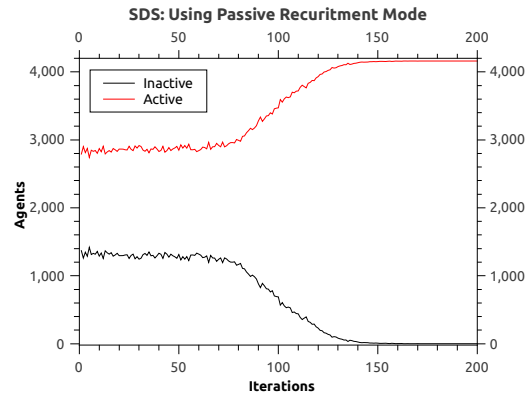
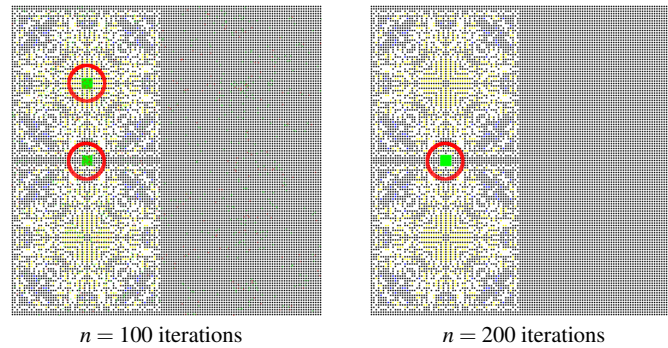


Fig. 8 Passive Recruitment mode: finding absolute symmetry

7 Comparison between information gain and swarm intelligence

In this work, various CA generated images were used as input to both information gain as well as the swarm intelligence algorithm in order to detect the structural characteristics (i.e. order-symmetry and randomness).

It is shown that information gain is able to discriminate structural characteristics ranging from full symmetry, partially structured and fully random patterns (see Fig. 4). One of the main feature of this measure is to quantify the existing structure of the image.

However information gain is unable identify the centre of symmetry and if there are multiple centres of symmetries (see Fig. —), without having a complete centre of symmetry, it classifies the image as completely symmetrical.

The swarm intelligence technique used (Stochastic Diffusion Search or SDS), is capable of identifying all the centres of symmetry irrespective of the structure of the image as evaluated by information gain. On the other hand, SDS is unable to quantify the structure of the input images and shows weakness when fed with images which lack any “visible” centres of symmetries.

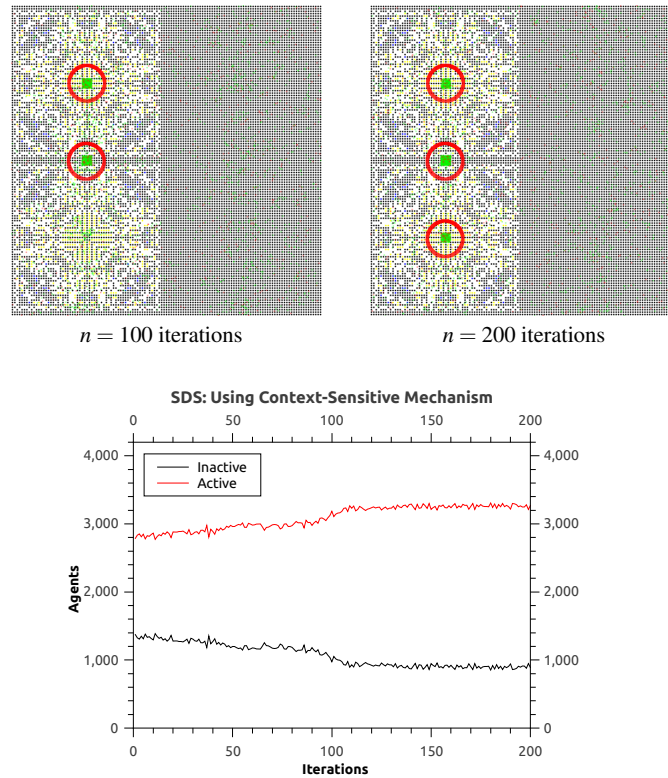


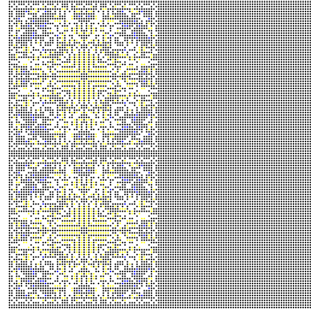
Fig. 9 Context-Sensitive mechanism: finding partial symmetry

These results show the complementary nature of information gain and SDS to work alongside each other offering insight into both the structure of the images as well as the presence of any symmetries.

8 Conclusion

CA provide perspective and powerful tools in generating computer graphics. The multi-state CA rule space is a vast set of possible rules which can generate interesting patterns with high aesthetic qualities. The interaction of CA rules at local level generates emergent global behaviour, that can sometimes demonstrate attractive complexity. Some characteristics of CA, such as the regularity and complexity of the rules that are employed locally, suggest that they could be well suited to generating computer graphics.

This paper demonstrates the capability of a swarm intelligence algorithm – Stochastic Diffusion Search – in detecting absolute symmetries (when present) and



$$\begin{aligned}
 H &= 0.92537 \\
 \bar{G}_{i,j+1} &= 0.86462 \\
 \bar{G}_{i,j-1} &= 0.86462 \\
 \Delta \bar{G}_{i,j\pm 1} &= 0 \\
 \bar{G}_{i-1,j} &= 0.86462 \\
 \bar{G}_{i+1,j} &= 0.86462 \\
 \Delta \bar{G}_{i\pm 1,j} &= 0 \\
 \bar{G}_{i-1,j+1} &= 0.89953 \\
 \bar{G}_{i+1,j-1} &= 0.89953 \\
 \Delta \bar{G}_{P_d} &= 0 \\
 \bar{G}_{i+1,j+1} &= 0.89953 \\
 \bar{G}_{i-1,j-1} &= 1.98465 \\
 \Delta \bar{G}_{S_d} &= 0
 \end{aligned}$$

Fig. 10 Information gain measure applied on the CA generated pattern

the centre of partial symmetrical patterns within the input image. Additionally entropy and information gain were introduced and applied to several CA generated patterns with varying structural characteristics (order-symmetry and randomness).

Evaluating the symmetry of cellular automata generated patterns is often a difficult task partly due to the large size of the search space, and partly due to the constantly changing, dynamic environment in which the cellular automata patterns are generated. These factors contribute to making the detection of symmetrical patterns computationally expensive.

One of the main features of Stochastic Diffusion Search is partial function evaluation which is particularly useful when dealing with large problems with high dimensions and costly evaluation function (e.g. in this case, the expensive computational cost of detecting symmetry in cellular automata generated patterns). The performance of this algorithm is explained in the paper and the results are accordingly demonstrated.

The advantages and weaknesses of the swarm intelligence technique and the information gain measure are presented and it is shown that they can be used as complementary tools to better understand the structural complexity of the input images.

Following the introduction of this novel technique, among the future research topics are: conducting a comparison with other evolutionary and non-evolutionary techniques, computing the correlation between the size of search space and the computational complexity of the process, ranking the quality of the symmetries detected, and applying this method to dynamically evolving cellular automata generated patterns.

References

1. Al-Rifaie, A.M., al-Rifaie, M.M.: Generative music with stochastic diffusion search. In: C. Johnson, A. Carballal, J. Correia (eds.) *Evolutionary and Biologically Inspired Music, Sound, Art and Design, Lecture Notes in Computer Science*, vol. 9027, pp. 1–14. Springer International Publishing (2015). DOI 10.1007/978-3-319-16498-4_1. URL http://dx.doi.org/10.1007/978-3-319-16498-4_1
2. Al-Rifaie, F.M., al-Rifaie, M.M.: Investigating stochastic diffusion search in dna sequence assembly problem. In: *SAI Intelligent Systems Conference 2015*. IEEE (2015)
3. al-Rifaie, M.M., Bishop, M.: Stochastic diffusion search review. In: *Paladyn, Journal of Behavioral Robotics*, vol. 4(3), pp. 155–173. Paladyn, Journal of Behavioral Robotics (2013). DOI 10.2478/pjbr-2013-0021. URL <http://dx.doi.org/10.2478/pjbr-2013-0021>
4. al-Rifaie, M.M., Bishop, M., Blackwell, T.: Information sharing impact of stochastic diffusion search on differential evolution algorithm. In: *J. Memetic Computing*, vol. 4, pp. 327–338. Springer-Verlag (2012). DOI 10.1007/s12293-012-0094-y. URL <http://dx.doi.org/10.1007/s12293-012-0094-y>
5. al-Rifaie, M.M., Bishop, M., Caines, S.: Creativity and autonomy in swarm intelligence systems. In: *J. Cognitive Computation*, vol. 4, pp. 320–331. Springer-Verlag (2012). DOI 10.1007/s12559-012-9130-y. URL <http://dx.doi.org/10.1007/s12559-012-9130-y>
6. Andrienko, Yu. A., Brilliantov, N. V., Kurths, J.: Complexity of two-dimensional patterns. *Eur. Phys. J. B* **15**(3), 539–546 (2000)
7. Andrienko, Yu. A., Brilliantov, N. V., Kurths, J.: Complexity of two-dimensional patterns. *Eur. Phys. J. B* **15**(3), 539–546 (2000)
8. Atallah, M.J.: On symmetry detection. *Computers, IEEE Transactions on* **100**(7), 663–666 (1985)
9. Bates, J.E., Shepard, H.K.: Measuring complexity using information fluctuation. *Physics Letters A* **172**(6), 416–425 (1993)
10. Bates, J.E., Shepard, H.K.: Measuring complexity using information fluctuation. *Physics Letters A* **172**(6), 416–425 (1993)
11. Bauerly, M., Liu, Y.: Computational modeling and experimental investigation of effects of compositional elements on interface and design aesthetics. *International Journal of Man-Machine Studies* **64**(8), 670–682 (2006)
12. Behrens, R.: *Design in the visual arts*. Prentice-Hall (1984)
13. Bishop, J.: Stochastic searching networks. In: *Proc. 1st IEE Conf. on Artificial Neural Networks*, pp. 329–331. London, UK (1989)
14. Bishop, J., Torr, P.: The stochastic search network. In: *Neural Networks for Images, Speech and Natural Language*, pp. 370–387. Chapman & Hall, New York (1992)
15. Branke, J., Schmidt, C., Schmeck, H.: Efficient fitness estimation in noisy environments. In Spector, L., ed.: *Genetic and Evolutionary Computation Conference*, Morgan Kaufmann (2001)
16. Carroll, M., J. (eds.): *HCI Models, Theories, and Frameworks: Toward a multidisciplinary science*. Morgan Kaufmann Publishers, San Francisco (2003)

17. Cover, T.M., Thomas, J.A.: Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience (2006)
18. Dawkins, R.: The blind watchmaker. New York: Norton & Company, Inc (1986)
19. Digalakis, J., Margaritis, K.: An experimental study of benchmarking functions for evolutionary algorithms. *International Journal* **79**, 403–416 (2002)
20. Gangestad, S.W., Thornhill, R., Yeo, R.A.: Facial attractiveness, developmental stability, and fluctuating asymmetry. *Ethology and Sociobiology* **15**(2), 73–85 (1994)
21. Hinton, G.F.: A parallel computation that assigns canonical object-based frames of reference. In: Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2, pp. 683–685. Morgan Kaufmann Publishers Inc. (1981)
22. Javaheri Javid, M.A., Blackwell, T., Zimmer, R., al Rifaie, M.M.: Spatial Complexity Measure for Characterising Cellular Automata Generated 2D Patterns. In: Pereira, F. and Machado, P. and Costa, E. and Cardoso, A. (ed.) Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings., *Lecture Notes in Artificial Intelligence*, vol. 9273, pp. 201–212. Springer International Publishing (2015)
23. Jiang, H., Ngo, C.W., Tan, H.K.: Gestalt-based feature similarity measure in trademark database. *Pattern Recognition* **39**(5), 988–1001 (2006)
24. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. In: *Soft Computing* **9**, 3–12 (2005)
25. Lee, S., Liu, Y.: Curved glide-reflection symmetry detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(2), 266–278 (2012)
26. Lewis, M.: Evolutionary visual art and design. In: J. Romero, P. Machado (eds.) *The Art of Artificial Evolution, Natural Computing Series*, pp. 3–37. Springer (2008)
27. Leyton, M.: *Symmetry, causality, mind*. Bradford Books. MIT Press (1992)
28. Linz, P.: *An introduction to formal languages and automata*. Jones & Bartlett Publishers (2001)
29. Liu, Y.: Computational symmetry. In: CMU Robotics Institute (2000)
30. McCormack, J.: Interactive evolution of l-system grammars for computer graphics modelling. *Complex Systems: from biology to computation* pp. 118–130 (1993)
31. Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)* **25**(3), 560–568 (2006)
32. Møller A., P., Cuervo, J.J.: Asymmetry, size and sexual selection : meta-analysis, publication bias and factors affecting variation in relationships, p. 1. Oxford University Press (1999)
33. Møller A., P., R., T.: Bilateral symmetry and sexual selection: a meta-analysis. *Am. Nat* **151**(2), 174–192 (1998)
34. Park, I.K., Lee, K.M., Lee, S.U.: Perceptual grouping of line features in 3-D space: A model-based framework. *Pattern Recognition* **37**(1), 145–159 (2004)
35. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3d shapes. In: *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 549–559. ACM (2006)
36. Railton, P.: *Aesthetic Value, Moral Value and the Ambitions of Naturalism In Aesthetics and Ethics*, chap. 3. University of Maryland (2001)
37. Randy, T., Steven, G.: Human facial beauty. *Human Nature* **4**, 237–269 (1993)
38. Shackelford, T.K..L.R.J.: Facial symmetry as an indicator of psychological emotional and physiological distress. *Journal of Personality and Social Psychology* **72** (1997)
39. Shannon, C.: A Mathematical Theory of Communication. *The Bell System Technical Journal* **27**, 379–423 & 623–656 (1948)
40. Shannon, C.: A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379–423 & 623–656 (1948)
41. Shannon, C., et al.: The synthesis of two-terminal switching circuits. *Bell System Technical Journal* **28**(1), 59–98 (1949)
42. Sims, K.: Evolving virtual creatures. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 15–22. ACM (1994)

43. Sun, C., Sherrah, J.: 3d symmetry detection using the extended gaussian image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **19**(2), 164–168 (1997)
44. Todd, S., Latham, W., Hughes, P.: Computer sculpture design and animation. *The Journal of Visualization and Computer Animation* **2**(3), 98–105 (1991)
45. Wackerbauer, R., Witt, A., Atmanspacher, H., Kurths, J., Scheingraber, H.: A comparative classification of complexity measures. *Chaos, Solitons & Fractals* **4**(1), 133–173 (1994)
46. Wackerbauer, R., Witt, A., Atmanspacher, H., Kurths, J., Scheingraber, H.: A comparative classification of complexity measures. *Chaos, Solitons & Fractals* **4**(1), 133–173 (1994)
47. Whitley, D., Rana, S., Dzuber, J., Mathias, K.E.: Evaluating evolutionary algorithms. *Artificial Intelligence* **85**(1-2), 245–276 (1996)
48. Wolter, J.D., Woo, T.C., Volz, R.A.: Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer* **1**(1), 37–48 (1985)
49. Zabrodsky, H., Peleg, S., Avnir, D.: Symmetry as a continuous feature. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **17**(12), 1154–1166 (1995)
50. Zhang, J.S., Chrzanowska-Jeske, M., Mishchenko, A., Burch, J.R.: Generalized symmetries in boolean functions: Fast computation and application to boolean matching. In: in *IWLS*. Citeseer (2004)