Chapter 11
Cellular Automaton Shading for Building Envelopes

Machi Zawidzki

Abstract. This chapter collects the key findings of the pioneering concept of a CA-based shading system for building envelopes, CASS for short. CASS can be considered as a realistic approach to realization of the architect's dream of an adaptive skin of a building which could dynamically change its appearance according to varying external conditions or the inhabitants' will. Since CASS is based on identical modular units it has potential of being relatively robust and inexpensive. Most importantly it takes visual advantage of CA's emergent properties which result in new, intriguing, organic aesthetics. The practical approach focused on tangible and "buildable" results make this material particularly suitable for designers and building engineers — groups that usually are not well addressed in CA research. Additionally, this paper presents an alternative approach to CA which may contribute to the mainstream research in the field.

11.1 Building Envelope and Daylighting

Architectural building envelope (BE) is an example of a real-life engineering design problem, which can be formulated as multi-objective optimization. BE is to serve a number of functions and meet requirements which sometimes are contradictory; the hierarchy of their importance is often ambiguous and strongly user-dependent. BE is an interface between the exterior and interior of a building and usually serves the four main functions:

- protection from external factors, as it improves security and reduces the levels of noise pollution;
- protection from climatic changes (temperature, humidity, glare);
- provision of natural light and visual contact with the environment, or visual isolation from exterior if required;
- aesthetics, as the facade is one of the most articulated visual aspect of a building.

Machi Zawidzki
Massachusetts Institute of Technology, USA
e-mail: zawidzki@mit.edu

P. Rosin et al. (eds.), *Cellular Automata in Image Processing and Geometry*,
Emergence, Complexity and Computation 10,
DOI: 10.1007/978-3-319-06431-4_11, © Springer International Publishing Switzerland 2014

The requirements for the indoor conditions vary among the occupants, however in relatively narrow ranges. For a recent survey of literature on how different factors influence human comfort indoors see [20]. Conversely, the exterior conditions vary substantially both in circadian and annual cycles. Therefore, modern BEs should adapt both to the occupants' requirements and to the variable outdoor conditions. Moreover, in principle, relying exclusively on artificial light indoors is not desirable and in many cases not allowable by building regulations. At humanistic level, according to [50]: *natural light in architecture must be part of a more general philosophy that reflects a more respectful, sensitive attitude in human beings towards the environment in which they live*. The main benefits of daylighting as a design strategy:

- Economical and ecological, as it substantially reduces the energy consumption and greenhouse gases emissions [2] [8];
- Physiological, as it is an effective stimulant to the human visual and circadian systems [10];
- Well-being, as it provides high illuminance and permits excellent color discrimination and color rendering; enables occupants to see both a task and the space well, and to experience some environmental stimulation [9]. Working by daylight is believed to result in less stress and discomfort.
- Social, as those of higher status in organizations are often given spaces closer to/with more windows [9].

Daylight offers a dynamic, changing pattern to stimulate the eye. It also provides a very wide range of illuminance: from 0 to over 25 000 lx, way beyond values commonly required indoors which range from 10 to 1000 lx, that is from the lowest level of color discrimination to the bright appearance [13]. Moreover, its thermal effect and its unique distribution of luminances are desirable in winter and in cold climates and undesirable in summer in hot climates [50]. Moreover, personal preference of illuminance levels and the degree of glare discomfort vary, and desired quantities of additional electric light depend on the type of task and the distance from the window. There are a number of systems for controlling these variances to appropriate levels. For a review see [48]. For a survey of systems that harvest the daylight, usually from the roof, and redistribute it inside the building see [32]. For other related issues in the context of CASS such as daylight design in urban areas, the importance of the outside view, smart windows, etc. see [65]. The complexity of functions performed by this type of a spatial barrier resembles organic skin. A comparison between skin and a building envelope is shown in Tab. 11.1 after [63]. As Tab. 11.1 indicates BE is a multifaceted problem and most importantly, no "conventional" solution that is optimal and universal exits. An alternative idea based on cellular automaton (CA), so called CA-shading system, CASS for short, has been proposed in [66]. In the original CASS the facade is divided into identical units which form a lattice for one-dimensional (1D) CA, and only the edge rows of cells are controlled directly. The pattern on the entire array is the result of local interactions determined

Table 11.1 A comparison between the functions of skin and building envelope

Function	**Skin**	**Building envelope**
Protection	S. is an anatomical barrier from pathogens and damage between the internal and external environment in bodily defense [43] [36].	Outer shell to protect the indoor environment.
Sensation	S. contains a variety of nerve endings that react to heat and cold, touch, pressure, vibration, and tissue injury.	Users' sensory contact with the outside (mostly visual, also audial).
Heat regulation	S. contains a blood supply far greater than its requirements which allows precise control of energy loss by radiation, convection and conduction. Dilated blood vessels increase perfusion and heat loss, while constricted vessels greatly reduce cutaneous blood flow and conserve heat.	Temperature control (insulation, solar gain, heat transfer, thermal mass, etc.)
Evaporation control	S. provides a relatively dry and semi-impermeable barrier to fluid loss [36].	Moisture control (e.g. air conditioning)
Storage and synthesis	S. acts as a storage center for lipids and water.	Heat storage (e.g. thermal mass, Trombe wall [49])
Absorption	Oxygen, nitrogen and carbon dioxide can diffuse into the epidermis in small amounts, some animals use s. for their sole respiration organ [12].	Ventilation (indoor air quality, hygiene and public health).
Water resistance	S. acts as a water resistant barrier so that essential nutrients are not washed out of the body.	Water barrier (water condensation).
Pigmentation	Camouflage, mimicry, UV protection, communication, sexual reproduction, warning etc.	Appearance (aesthetics, communication).
Structure	Other animal coverings such the arthropod exoskeleton or the seashell have different developmental origin, structure and chemical composition.	Structural integrity (shell).

by the update rules, the same for each cell. This system is therefore homogeneous like many biological systems [29], and emerging patterns exhibit certain "organic" integrity as shown in Fig. 11.1.

11.2 Why Cellular Automata to Drive Shading of Building Envelopes?

CA combine certain properties which are particularly suitable for application on building facades, as shown in Tab. 11.2. The first property listed there — *modularity* is mostly relevant to the fabrication of CASS described in Subsect. 11.6. The CA modules, although having uniform topology may have various shapes as in the case

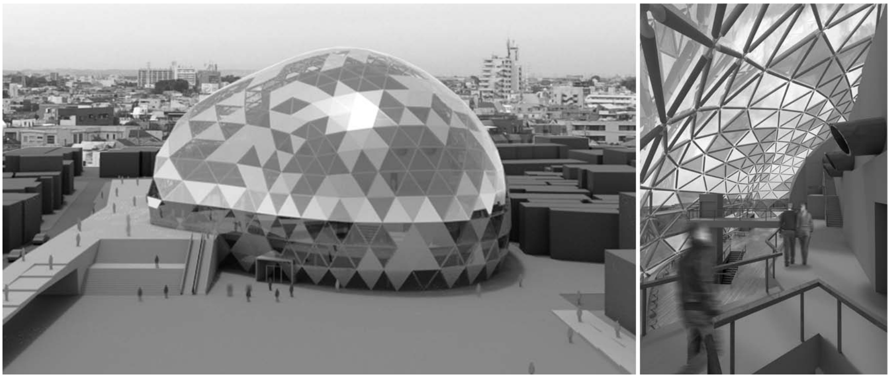


Fig. 11.1 Simulation of CA on a free-form BE comprised of triangular panels of Hita City Auditorium Competition Entry (2003, Oita, Japan, architectural design by Takehiko Nagakura). The 30^{th} time-step of two-color two-dimensional triangular CA rule 9622 is shown (compare with Fig. 11.19).

of CASS on free-form surfaces described in Subsect. 11.4.2. Combination of the last two properties from Tab. 11.2, that is *determinism* with *emergence* is the most challenging. Both derive from the sequence of CA patterns displayed on CASS. These patterns and their transitions must meet the following criteria:

1. Patterns to be visually interesting;
2. Ability of producing patterns of any average density between 0: full transparency, to 1: full opacity;
3. Density transitions to be gradual;
4. Patterns to be evenly distributed.

From these criteria a straightforward constraint can be immediately derived. Namely, only even-number (EN) CA rules should be considered. The shading action starts arbitrarily from full transparency so all cells have value 0. An odd-number rule produces non-zero state from all 0s in the background, so in the first time-step the state of practically entire CASS changes to 1, which precludes the third criterion listed above. In the case of CASS, the visual attractiveness can be attributed to the emergence, defined after [33] as: *a property of a complex system that is not exhibited by*

Table 11.2 Comparison of the key properties of CA and BE

Cellular automaton	**Building envelope**
In principle, all the cells of a particular CA's are identical.	Modularity is highly desirable in building industry, as it allows for mass prefabrication and easy assembly.
Determinism	The control of the state of BE, at least statistical, is crucial.
Capability of strong emergence	Potential of the new, intriguing, organic-like aesthetics

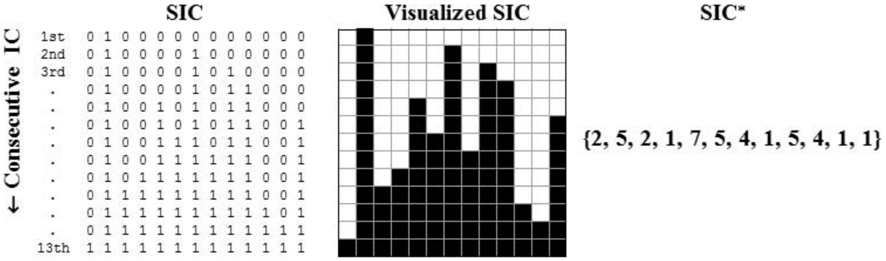
its individual component parts determined from a model of the system. The emergence in CA is manifested by the simultaneous emergence of boundaries between the domains, which take form of boundaries or “particles”, that is small regions of cells separating two domains and persisting for a relatively many time steps [24]. Particular type of particle emerges in the so-called “solitons” — moving persistent structures which pass through one another while preserving their identities [52]. The subsequent sections describe various CA which meet the criteria listed above. The review starts from the simple CA, preceded by a brief explanation of terms specific to CASS. In all the instances, the CA update is synchronous.

11.2.1 The Nomenclature

A few practical terms often used in the context of CASS are briefly explained below. For simplicity they were formulated and illustrated with rectangular, two-color (2C) CA.

The Sequence of Initial Conditions (SIC)
In principle, CA pattern is determined by local transition rules (TRs), initial condition (IC) and the boundary conditions (BC). As a result, so is the average pattern density, which is the ratio between the number of black cells (1s) to the number of all cells in the array. An array with all black cells forms a completely opaque facade (density 1), conversely, an array of all white cells (density 0) corresponds to 100% transparency. For an interactive demonstration see [59]. In CASS the pattern remains still until the next alteration in IC. Such changes to IC form a sequence(SIC), which must meet two arbitrary conditions: 1) for $k > 1$, the k^{th} IC has exactly one black cell more than $(k - 1)^{th}$; 2) all the remaining black cells to be preserved. A single IC is a list of $\{0,1\}$ of length n. SIC is also a n-long list of such lists, forming an $n \times n$ square matrix. Such SIC ensures that the displayed changes of the array will not appear excessively chaotic or disturbing. An alteration of the state of a single cell in IC propagates over the entire array, so the transition from one state of BE to another can be traced, understood and interpreted by the observer. SICs can be conveniently encoded into SIC* using order-based representation(OBR) [23]. A sequence of n ICs is represented by a list of integers of length $n-1$. Each integer is the “distance to the most recently added 1 among the available positions”. Thus SIC* is a “sequence of the leap lengths where consecutive 1s are located”. The positioning is done from left to right. Since in the very last step there is only one available position, is it dropped, as illustrated in Fig. 11.2. For detailed explanation of this encoding method see [67]. For the interpretation of the SIC shown in Fig. 11.2 see Fig. 11.3.

Pattern grayness, in other words pattern density, is defined as a ratio between the number of black (opaque) cells to the total number of cells. The grayness applies to



```
1st   0 1 0 0 0 0 0 0 0 0 0 0 0
2nd   0 1 0 0 0 0 1 0 0 0 0 0 0
3rd   0 1 0 0 0 0 1 0 1 0 0 0 0
 .    0 1 0 0 0 0 1 0 1 1 0 0 0
 .    0 1 0 0 1 0 1 0 1 1 0 0 0
 .    0 1 0 0 1 0 1 0 1 1 0 0 1
 .    0 1 0 0 1 1 1 0 1 1 0 0 1
 .    0 1 0 0 1 1 1 1 1 1 0 0 1
 .    0 1 0 1 1 1 1 1 1 1 0 0 1
 .    0 1 1 1 1 1 1 1 1 1 0 0 1
 .    0 1 1 1 1 1 1 1 1 1 1 0 1
 .    0 1 1 1 1 1 1 1 1 1 1 1 1
13th  1 1 1 1 1 1 1 1 1 1 1 1 1
```

Fig. 11.2 An example of a sequence of 13 initial conditions (SIC) with the corresponding encoded SIC*

any sub-matrix, row of cells or group of cells. A pattern with all white (transparent) cells has grayness 0, with all black (opaque) cells the grayness is 1.

Grayness Function (GF) links the grayness of IC with the grayness of the entire CA array. In other words, GF is a series of graynesses for consecutive ICs. For shading, GF should be monotonic, ideally the grayness of the entire array should be proportional to the grayness of IC. It should also render values from the entire range, that is from 0 to 1. Such GF ensures that the average opacity of the array can be fully controlled. For an interactive demonstration see [41].

Grayness Function Error(GFE) is the difference between the ideal (proportional) GF and the GF of a particular CA at given SIC. For a single CA array $\boldsymbol{A}$ at the given IC it is expressed as:

$$\mathrm{GFE}[\boldsymbol{A}] = \frac{\sum_{w=1}^{W}\sum_{h=1}^{H} \mathrm{a}_{h,w}}{WH} - \frac{\sum_{w=1}^{W} \mathrm{a}_{1,w}}{W} \tag{11.1}$$

Where H and W are the height and width of an array (facade) respectively. GFE for SIC is the result of the summation for all ICs:

$$\mathrm{GFE} = \frac{1}{F}\sum_{i=1}^{F} |\mathrm{GFE}[\boldsymbol{A}_i]| \tag{11.2}$$

Where F is the number of facades' states which equal to the length of SIC, $\boldsymbol{A}_i$ is the i^{th} CA array (for the i^{th} IC of SIC). GFE is usually used for preliminary search for automata suitable for CASS.

Grayness Monotonicity and Pattern Distribution Error(GDE). As discussed in [67], finding the best SIC is a computationally challenging problem. GDE is used for fine-tuning the SIC as it evaluates each CA pattern considering both the grayness function monotonicity and the uniformity of the pattern distribution:

$$\mathrm{GDE}[\boldsymbol{A}, r] = \sum_{k=1}^{\lfloor \frac{W}{r} \rfloor - 1} \frac{\left| \frac{1}{Hr}\sum_{x=kr}^{(k+1)r}\sum_{h=1}^{H} a_{h,w} - \frac{1}{W}\sum_{w=1}^{W} a_{1,w} \right|}{\lfloor \frac{W}{r} \rfloor} \tag{11.3}$$

Where A is a CA array (facade), H and W are the height and width of the array, respectively; r and k are the width and the number of vertical stripes in which the entire array is subdivided, respectively. The summation over all CA patterns produced at each IC from the given SIC:

$$\mathrm{GDE} = \frac{1}{F} \sum_{f=1}^{F} \mathrm{CFA}[\boldsymbol{A}_f] \tag{11.4}$$

Where F is the number of facades' states which equals to the length of SIC, $\boldsymbol{A}_f$ is an f^{th} CA array corresponding to the f^{th} IC in SIC. If each of the r-wide sub-arrays of a given CA pattern have the same average densities, the values of GDE and GFE are equal. For further details see [69]. CA for shading must be both controllable, that is having sufficiently low GDE, and visually appealing. Although the latter criterion is rather arbitrary, in most of the cases, the necessary condition is that the CA pattern is complex, that is of Wolfram class IV behavior [66].

11.3 One-Dimensional Cellular Automata Applied on Surfaces

Although BEs are two-dimensional (2D), the application of one-dimensional (1D) CA for CASS seems the most practical. This is analogous to the common convention for presenting a 1D CA by showing the history of its generation changes, where each row corresponds to a time step in the history. Every row becomes IC for the next row and so forth. This process continues in a cascade and propagates over the entire array. Use of a 2D CA may seem more intuitive because this domain is greater than for one dimensional automata, increasing the chances of finding the best one for CASS. Moreover, the inter-cell wiring seems to be easier to fabricate. Nevertheless, as mentioned in [66], there are major concerns regarding the application of 2D CA. Although their behavior is often intriguing, it is very difficult to control the states of cells. 2D CA continuously updates all the cells until an equilibrium is reached, which almost always leads to an uninteresting, uniform state of the array. All the cells become black or white, often with artifacts (small islands of the opposite state) and usually with locally strobing cells (switching their state at every step forever). In general the exact final state of the 2D CA array is difficult or impossible to predict due to the computational irreducibility. Such 2D array are usually not useful for shading due to strobing. Controlling the state of 2D arrays is difficult, or perhaps impossible. This problem could be solved by freezing the array at a certain step and not allowing it to evolve further, but at present it seems to be a difficult technical problem. With the adopted common convention of displaying 1D CA, this problem does not occur because every row displays the state at a certain step, and once set the state is maintained. The second major problem with 2D CA is the setting of ICs. How is the initial input given to the cells of a 2D array? A possible solution where only cells on the edges of the array are used, as done in the 1D case, has been demonstrated in [63].

11.3.1 Elementary Cellular Automata

There are 256 1D 2C, radius 1 (r1), so called elementary cellular automata (ECA). However, due to symmetries, the number of fundamentally inequivalent ECAs is only 88 [29]. For an interactive demonstration of CASS based on ECAs see [62].

11.3.2 The Original CA for Shading

Since none of ECAs meet both “shading” criteria, for the original CASS 2C1Dr2 CA code {3818817080,2,2} has been introduced in [66]. The first, second and third values in the latter code represent the decimal enumeration of the TR outputs, the number of possible states (colors) and the neighborhood size, respectively. Figure 11.3 shows the best possible, that is the ideal SIC for a 13 × 13 array at fixed boundary conditions with this CA.

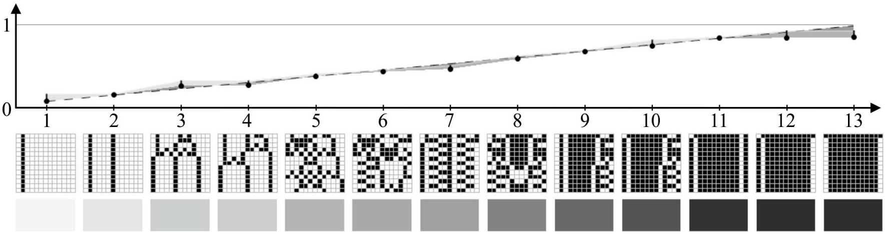


Fig. 11.3 13 × 13 array {3818817080,2,2} CA at SIC*: {2 5 2 1 7 5 4 1 5 4 1 1} renders the opacity transition with the lowest GDE = 0.58 and GFE = 0.42. From the top: 1): Black dots indicate the value of grayness (the average density) of each CA pattern. Dashed line indicates the referential proportionality. Dark gray filling indicates GFE at particular IC: excessive and insufficient densities are shown over and under the reference line, respectively. Similarly, light gray filling indicates GDE at particular IC. 2): The sequence of patterns. 3): The sequence of gray levels equivalent to the graynesses of the CA patterns shown above.

11.3.3 Half-Distance Automata

So called half-distance rules can be created by shifting the successive rows, so the number of input cells becomes even. If an underlying cell is placed between the two cells above, it is called radius $\frac{1}{2}$ (r-$\frac{1}{2}$) CA. For radii $\frac{3}{2}$ (r-$\frac{3}{2}$) and $\frac{5}{2}$ (r-$\frac{5}{2}$), the underlying cell is placed between the corresponding 4 and 6 cells, respectively. For an interactive demonstration see [40]. There are only 16 r-$\frac{1}{2}$ 1D2C such automata, none of them particularly interesting visually. However, the number of corresponding r-$\frac{3}{2}$ automata is 65,536. Half of them, that is 32,768, being EN which is rather manageable for simple analysis. As Fig. 11.4 indicates many of them have low GFE. As Fig. 11.4 indicates, there are many, namely 720, r-$\frac{3}{2}$ CA with very low GFE, that is below 0.05. The six automata whose GF plots are shown on the right have been selected according to the following criteria:

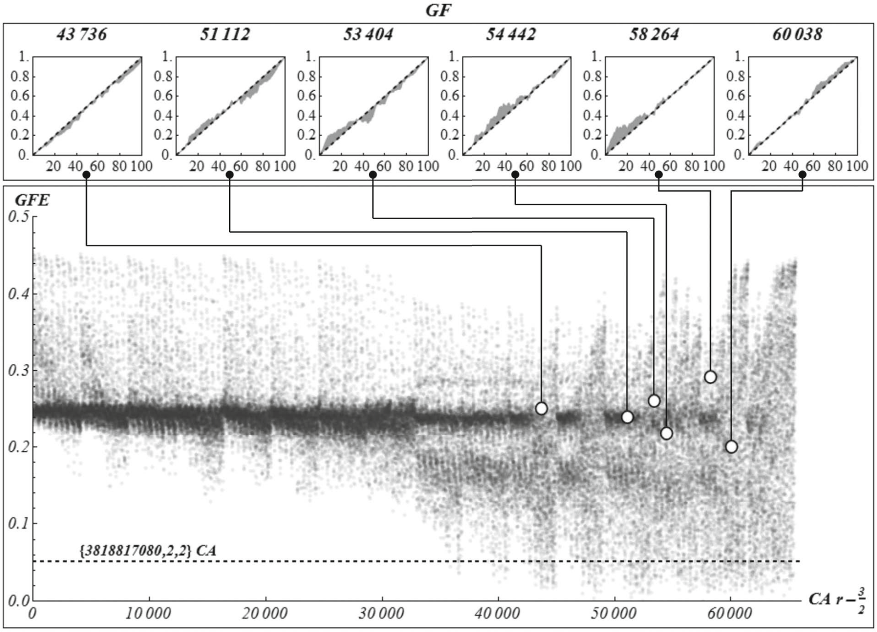


Fig. 11.4 r-$\frac{3}{2}$ CA: On the bottom: the GFE plot for all automata. Some of them with very low GFE (below 0.1). The referential value for {3818817080,2,2} CA is shown with the dotted line. On the top: the GF plots of six selected automata. The referential proportionality and the differences to it are shown with the dotted line and gray filling, respectively.

1. CA Patterns to be vertical, or at least no direction (left or right) should be dominant;
2. Pattens to continue for many steps;
3. Patterns to be distinct, preferably organic; for discussion on "man-made versus organic appearance" see [63].

Evenness of the pattern distribution was not considered. Figure 11.5 shows sample patterns of these automata. Half-distance automata are particularly interesting for CASS, since they can be applied on hexagonal and triangular tessellations, as described in Subsect. 11.3.5.

11.3.4 Higher Order Cellular Automata

A *k*-order cellular automaton is a type of reversible CA where the state of a cell at time *t* depends not only on its neighborhood at time *t* - 1, but on its states at $\{t-1,\ldots,t-k\}$. These automata are particularity interesting due to their intriguing properties [3] and relatively simple implementation for CASS. The number of second order 1D2Cr1 CA, or 2-CA is rather unmanageable for a simple search (total of $2^{2^6} = 1.84 \times 10^{19}$, half of them being EN), however, although there are only

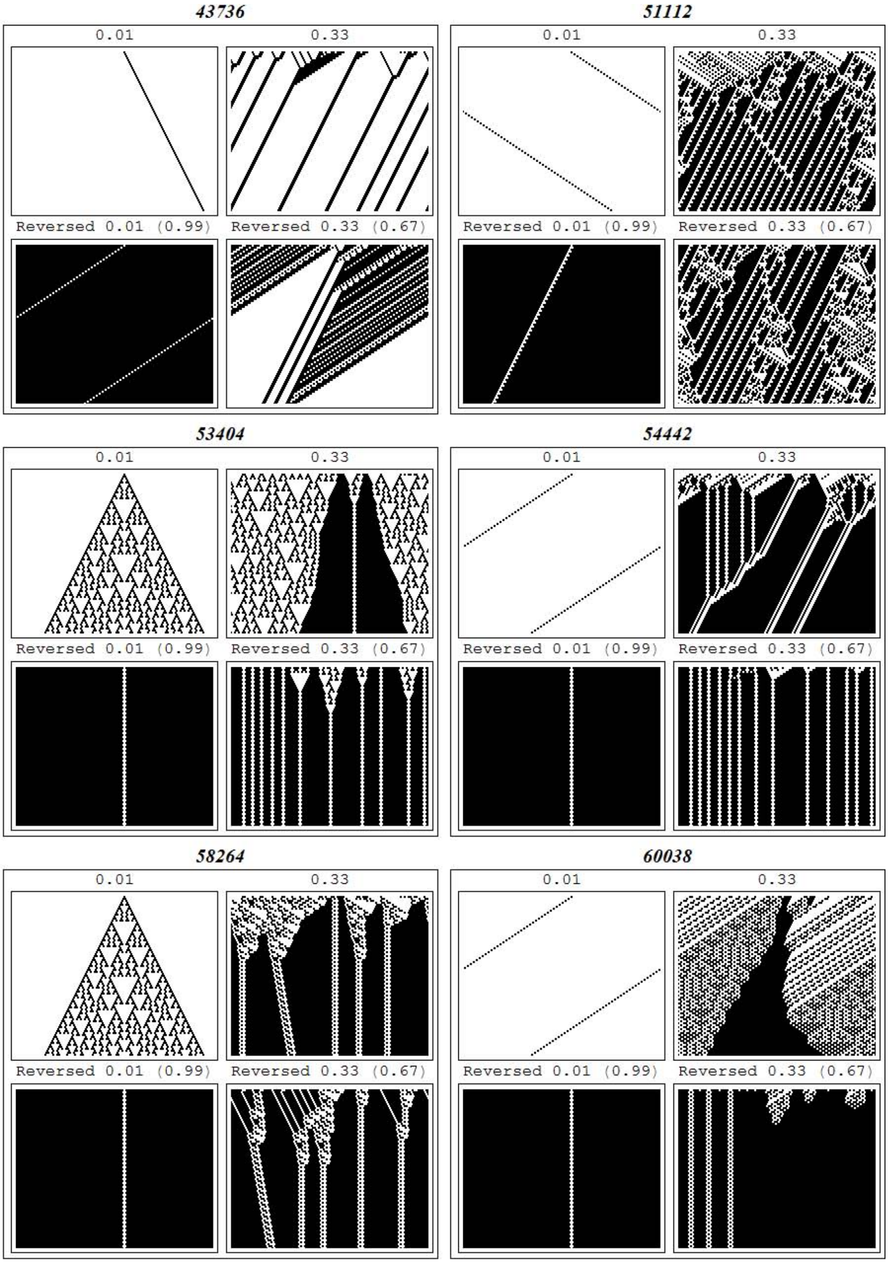


Fig. 11.5 The patterns of r-$\frac{3}{2}$ CA rules: 43736, 51112, 53404, 54442, 58264 and 60038. Patterns at the same ICs are shown for each rule: on the left: a single 1; below: the corresponding reversed IC with a single 0; on the right: $\frac{1}{3}$ rate of 1s and below: the corresponding reversed IC with $\frac{1}{3}$ 0s.

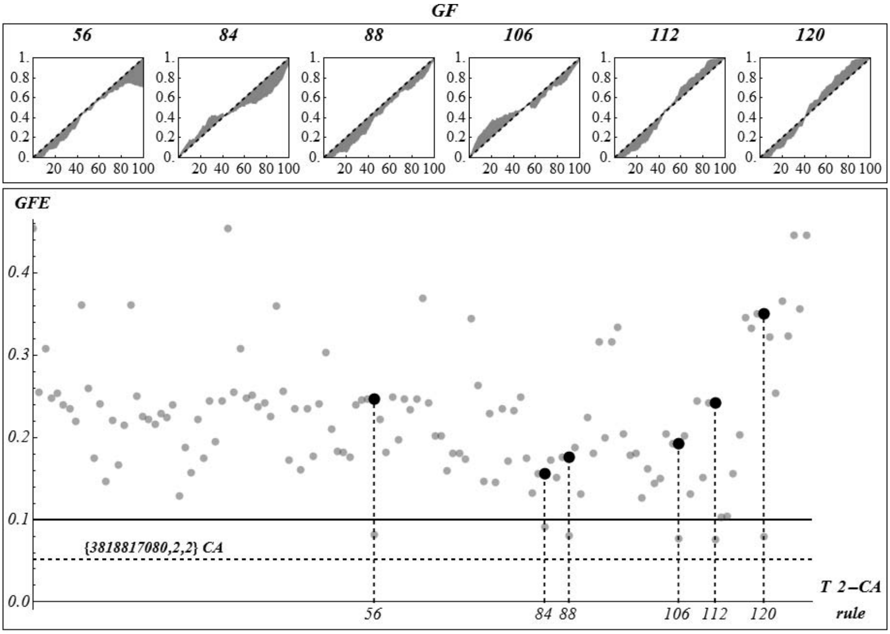


Fig. 11.6 T 2-CA: On the bottom: there are six T 2-CA with relatively low GFE (below 0.1). The referential value for {3818817080,2,2} CA is shown with the dotted line. On the top: the corresponding GF plots. The referential proportionality and the differences to it are shown with the dotted line and gray filling, respectively.

128 totalistic automata of this kind (T 2-CA), of which 64 are EN T 2-CA, some of them have rather good shading properties, as shown in Fig. 11.6. Figure 11.7 shows sample patterns of the selected T 2-CA. Rules 112 and 120 are not shown as they produce overly simple patterns.

11.3.5 Other Regular Tessellations: Hexagonal and Triangular

The square grid is one of three, regular, also called “Platonic” tilings. The remaining two are: triangular and hexagonal. Regular tilings allow edge-to-edge tiling by congruent regular polygons [11]. This property has been widely used in architectural practice since antiquity, and the first systematic mathematical treatment was done in the early XVII century in [30]. The properties of regular tessellations in the context of architectural design and particularly for BE shading are collected in Tab. 11.3. Although the original concept of CASS was based on an opto-mechanical system of square plates made of polarized glass [66], the prototype was based on liquid crystal (LC) technology [68]. The concept of polarized film shading system (PFSS), where shading elements are composed of two polygonal sheets of normally-white polarized films is explored in [65]. One of the parts of PFSS is fixed and the other rotates.

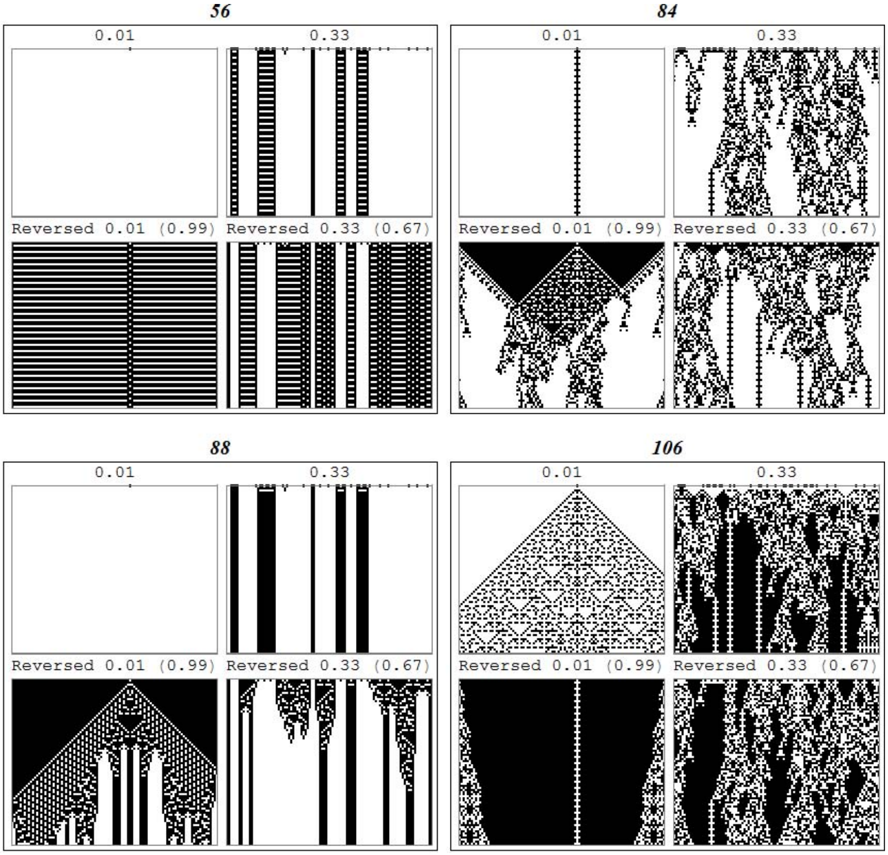


Fig. 11.7 The patterns of T 2-CA rules: 56, 84, 88 and 106. Patterns at the same (coupled) ICs are shown for each rule: a single 1; below: the corresponding reversed IC with a single 0; $\frac{1}{3}$ rate of 1s and the corresponding reversed IC with $\frac{1}{3}$ 0s.

Opacity of the idealized PFSS units is proportional to the angle of rotation: at the parallel position, that is 0°, the opacity is 0 and at the crossed position, that is 90°, it is 1. The minimal and maximal opacities occur at $k \times 180^\circ$ and $k \times 180^\circ + 90^\circ$, respectively; where $k \in \mathbb{N}$. Since the states of a CA are discrete, it is natural to limit the rotation angles of polygonal elements to discrete, so called dihedral rotations (DR). The numbers of DRs for triangle, square, and hexagon are 3, 4 and 6, respectively, as shown in Fig. 11.8. As Fig. 11.8 indicates, PFSS based on triangular, square, and hexagonal tessellations have at maximum 3, 2 and 3 distinct opacities, respectively. However, only the square PFSS renders extreme values from 0 to 1. The triangular and hexagonal PFSS render three shades of gray: 0.16, 0.5 and 0.83. Square PFSS can realize any 2C1D CA, the hexagonal and triangular PFSS can realize both two and three states 1D CA. There are two possible arrangements of cells in the hexagonal tessellation. So called “zigzag” tessellation (hexagon-Z) requires one type of

Table 11.3 The regular tessellations in the context of design and CA shading. *Those two types although have different shapes, are based on identical CA logic.

	Occurrence in nature	**Design**	**Visual attractiveness** **Impact on the pattern**	**CA shading**
□	Extremely rare in macro scale in nature (e.g.: bismuth and galena crystals, cobwebs of *Cyrtophora citricola* [42]).	Rectangular or square meshes are prevalent in built environment.	Not particularly attractive Neutral	The easiest to apply. Each CASS module is identical.
▽	Regular triangular meshes do not occur naturally in macro scale.	Triangles are the only polygons that are rigid in plane [34]. Although this property is particularly useful in architecture and engineering, triangular grid is relatively rarely exposed in the built environment.	Relatively interesting Strong	Some CASS require two types of modules*.
⬡	It is encountered in macro scale in nature more often than other types of regular grids (e.g.: honeycomb, basalt columns). Therefore it carries certain visual organic appeal.	Since it is the only regular tessellation without single points of contact, the patterns appear as the most coherent. Nevertheless, it is relatively rare in human design.	Very attractive Minimal	Some CASS require two types of modules*.

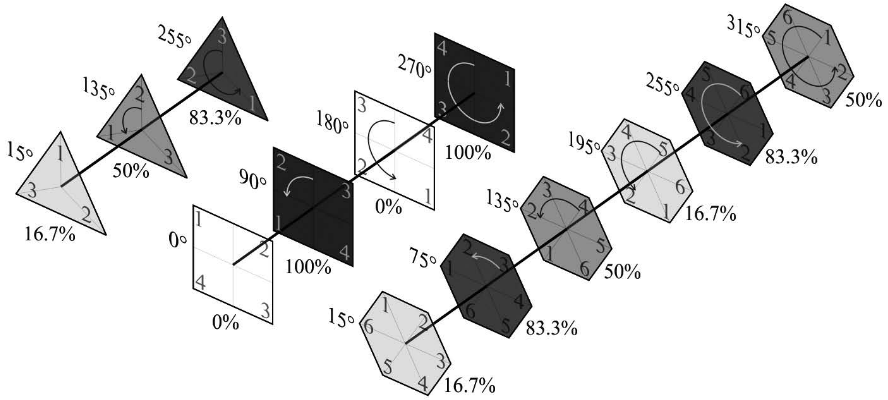


Fig. 11.8 PFSS opacity as a function of dihedral rotation on regular tessellations

module, and the "armchair" tessellation (hexagon-A) requires two modules of different shapes: O and E [6]. For neighborhoods of radii of any positive integer the triangular tessellation also requires two types of modules. Figure 11.9 illustrates the original CASS based on CA rule {3818817080,2,2} implemented by four types of PFSS.

11.3.5.1 PFSS on Hexagonal Grid

PFSS on hexagonal tessellation can execute 2 or 3 state CA. The following examples are based on the former, for 3 state examples see [65]. As mentioned above, depending on orientation, there are two types of hexagonal grids: "zigzag" and "armchair", called hexagon-Z and hexagon-A, respectively.

Hexagon-Z: PFSS$_{HZ}$

Only one type of module is necessary to emulate 1D CA with $PFSS_{HZ}$. However, as Fig. 11.9 indicates, due to the asymmetry of the module, the pattern normally vertical, becomes skewed. It seems possible to find automata whose behavior will compensate for this, however, implementation of half-distance automata is more straightforward as shown in Fig. 11.10.

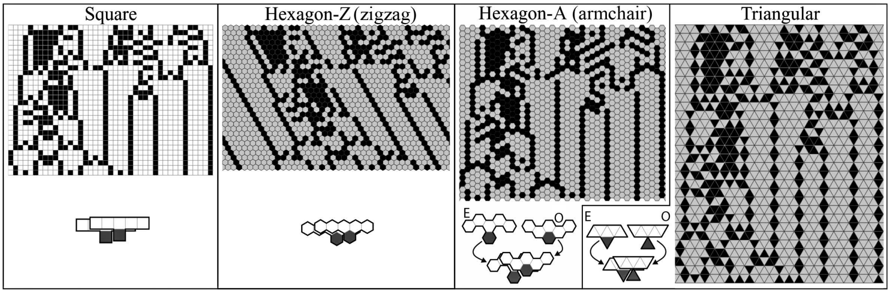


Fig. 11.9 CA rule {3818817080,2,2} realized by PFSS on regular tessellations. Gray levels on arrays represent the opacity values achievable by particular PFSS. For each type two modules for each tessellation are shown schematically. Coupled modules: E and O are shown accordingly.

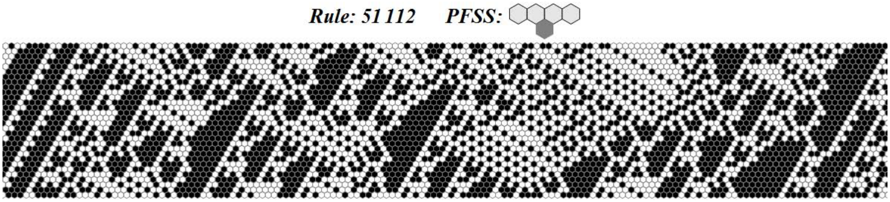


Fig. 11.10 The pattern produced by $PFSS_{HZ}$ driven by 1D2C r-$\frac{3}{2}$ CA rule 51112 from Fig. 11.5.

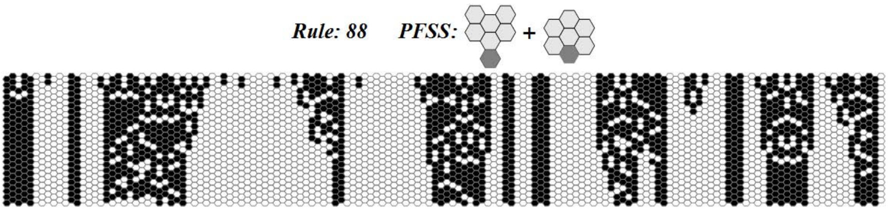


Fig. 11.11 The pattern produced by $PFSS_{HA}$ driven by T2-CA rule 88. IC rate of opacity: 0.5. Two PFSS modules are shown on the top, from the left: E and O.

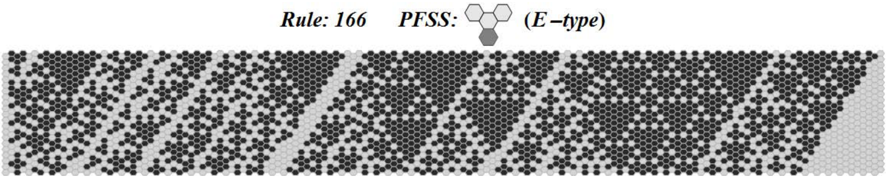


Fig. 11.12 The pattern produced by $PFSS_{HA}$ driven by 2D*2Cr1 $hexagon - A_E$ rule 166. IC rate of opacity: 0.8.

Hexagon-A: $\mathbf{PFSS_{HA}}$

Alternatively, hexagons in a lattice can be arranged so their parallel sides are aligned with the rows of cells. Such arrangement is called hexagon-A, and the corresponding shading system: $PFSS_{HA}$. As mentioned above, in such a case emulating 1D CA requires two types of modules: O and E. Figure 11.11 illustrates the application of totalistic second-order(T 2-CA) rule 88 selected from Fig. 11.7. However, it is also possible to arrange an array of cells using only one type: E or O. Such systems derived from one-dimensional cellular automata, are called two-dimensional cellular automata with specified offsets, and denoted here as 2D*CA. For example there are 256 2D*1Cr1CA analogous to ECA. Although the behavior is also analogous, the patterns produced by the former on $PFSS_{HAE}$ are more visually attractive. For example ECA rule 116 produces very simple patterns at very low GFE. However, the analogous 2D*2Cr1 $hexagon - A_E$ rule 166, although maintains low GFE, produces more intriguing patterns, as shown in Fig. 11.12. For more examples, including $hexagon - A_E$ and 3-state $PFSS_{HA}$ see [65].

11.3.6 PFSS on Triangular Grid: $PFSS_T$

Similarly, in the case of systems based on triangular tessellation ($PFSS_T$), emulating 1D CA with integer-value radii also requires two types of modules, as shown in Fig. 11.9. As noted in Tab. 11.3, the visual impact of the triangular tessellation is very strong. Therefore, it may not be particularly suitable for complex, organic patterns. However, its strong appearance combined with relatively simple CA can produce visually attractive patterns of different nature. As an example, Fig. 11.13 shows $PFSS_T$ based a second-order totalistic automaton (T2-CA) rule 56, the same as in

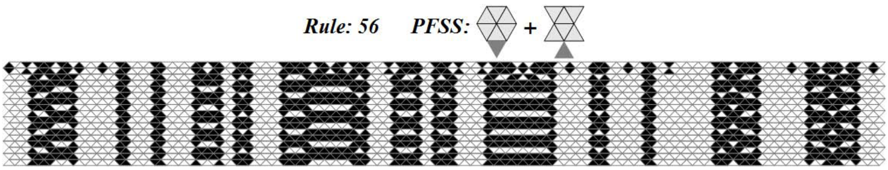


Fig. 11.13 The strong appearance of triangular tessellation can be advantageous. Despite the simplicity of the pattern produced by T2-CA rule 56 it becomes more visually attractive on triangular grid. IC rate of opacity: 0.5. Two PFSS modules are shown on the top, from the left: E and O.

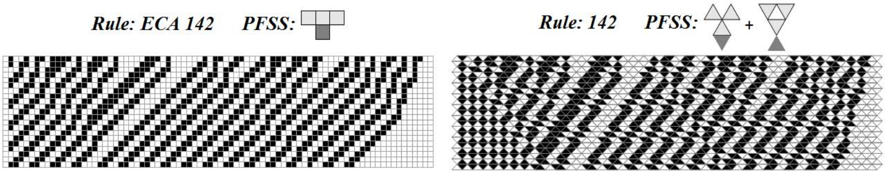


Fig. 11.14 Two patterns showing essentially the same CA at the same IC. On the left: ECA rule 142. On the right: analogous CA realized by combination of two modules (O and E) of $PFSS_T$.

Fig. 11.7. A small modification to the modules shown in Fig. 11.13 allow to emulate $hexagon - A_O$ PFSS, as shown in Fig. 11.14. The logic of both $PFSS_T$ modules shown in Fig. 11.14 is identical to the logic of the corresponding ECA. However, otherwise rather plain patterns gain certain visual attractiveness. The implementation of half-distance CA by $PFSS_T$ also requires two type of modules. However, creating perforated $PFSS_T^*$ by removing every other triangular facet allows to use only one type of module. Moreover, such an arrangement seems particularly interesting for BE, as every void in the tessellation can represent clear glass or solid walls, as shown in figures 11.15 and 11.1. Since the area of $PFSS_T^*$ is perforated in 50%, it should be taken in consideration, that the pattern is not as legible as in other shading systems presented here.

11.4 Two-Dimensional Cellular Automata on Surfaces

As mentioned in the beginning, controlling of the state of a BE surface with 2D CA seems intuitive, but the actual implementation is in fact more challenging than with 1D automata. As previous sections demonstrated, it is rather straightforward to apply any of regular tessellation, that is triangular, square or hexagonal on a planar surface and apply an automaton into it. For the overview of CA, in particular the Game of Life (GOL) in triangular, pentagonal and hexagonal tessellations see [6]; for the corresponding interactive demonstration see [5]. Modest size examples of GOL in triangle, square, and hexagon planar topologies and hierarchical hexagonal grid on the sphere intended for modeling of biogeographical, ecological and

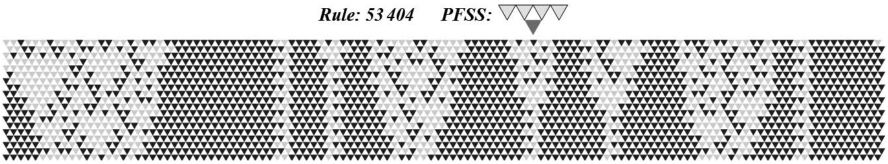


Fig. 11.15 Pattern produced by $PFSS^*_T$ driven by 1D2C r-$\frac{3}{2}$ CA rule 53404 selected from Fig. 11.5

epidemiological processes on the globe have been presented in [31]. Although the sphere is not a particularly practical shape for a building, domes have special position in the history of architecture. More recently, geodesic domes and spheres [47] have gained certain popularity due to their structural and aesthetic properties. A geodesic sphere (GS) is a spherical shell structure or lattice shell based on a network of great circles (geodesics) on the surface of a sphere, which intersect to form rigid triangular elements that also distribute the stress across the structure. Icosahedral geodesic sphere (IGS) is a spherical polyhedron with Euler characteristic $\chi = 2$. All triangular facets of IGS have 6 triangles per vertex, except, according to Euler's polyhedron formula [16], for 12 vertices with 5 triangles, regardless of the recursive subdivisions of the triangles (mesh resolution). The dual polyhedron of IGSs give Goldberg polyhedra [22], which consists of all hexagonal faces, except for also 12 [16], which are pentagons. [57] explores GOL on such a tessellation; for illustrative animation see [56].

The most general case of BE is a free-form surface, which cannot be constructed with regular polygons. Irregular tessellations have been introduced to more realistically model natural phenomena, e.g.: social networks have been simulated on irregular grid structures based on Voronoi-diagrams [18]. A geographic information system (GIS)-based CA on irregularly sized and shaped land parcels, at synchronous and asynchronous development have been used to model land-use change at the land parcel scale in [53]. Incremental matching method for processing of large structures that extend across many neighborhoods intended to enhance the data contained within topographic maps has been proposed in [37]. A cell-based wildfire simulator that uses an irregular grid that produces much faster results comparable in accuracy with traditional fire front propagation schemes has been presented in [28]. Graph-based cellular automata for CA models on irregular lattices have been introduced in [39].

For a free-form BE, however, triangular tessellation is the most suitable. Any 3D surface can be triangulated [45], that is divided into a set of (planar) triangles, with the restriction that each triangle side is entirely shared by two adjacent triangles. Analogous operation is impossible with (planar) quadrilaterals or hexagons. The control of state of such CASS is much more limited than in the previously described 1DCA-based systems. In the examples shown below, there is a single IC cell located arbitrary in the mesh. Alteration of the state of that cell triggers the 2D CA evolution, which propagates over entire mesh until the state of each cell stabilizes. It is imaginable that complex patterns perpetually fluctuating on BE could be desirable and feasible especially considering the number conserving cellular automata (NCCA) [27] [19]. However, here for simplicity, the CA pattern is expected to become still as soon as the propagation reaches the boundary of BE.

11.4.1 Triangular Cellular Automata

Although the most commonly used lattice for CA regardless of the dimension is an orthogonal grid, several studies have been carried out to examine the properties of triangular cellular automata (TCA), e.g. life-like rules on r1 Moore's neighborhood in a triangular lattice have been studied in [4], the computational universality of an 8-state triangular reversible partitioned CA has been demonstrated in [26]. The effect of simple memory on a particular reversible, structurally dynamic CA in the triangular tessellation has been demonstrated in [1]. TCAs has also been proposed for BE shading in [63] and the applications of totalistic and semi-totalistic [21] TCAs have been further studied in [25]. Figure 11.16 shows an example of the evolution of a semi-totalistic triangular CA (stTCA) on a regular triangular lattice with voids. For an illustrative corresponding demonstration see [61]. The history of evolution of a 2D CA can be conveniently shown as a column of consecutive time steps, where unfolded lists of cells are shown as single rows, as shown in Fig. 11.17. However, unlike the 1D case, the information about the adjacencies is partially lost. From such diagrams some fundamental information for CASS can be derived, namely: whether a certain automaton stabilizes, at how many time steps and with what final opacity rate (FOR). Figure 11.18 shows all stable stTCAs. As Fig. 11.18 indicates, there are eight 2D2Cr1stTCAs suitable for CASS: rules: 50, 114, 118, 178, 242, 246, 250 and 254. After stabilization, the average opacities, called here FOR are: 0.51, 0.66, 0.63, 0,55, 0.77, 1, 0.85 and 1, respectively. For

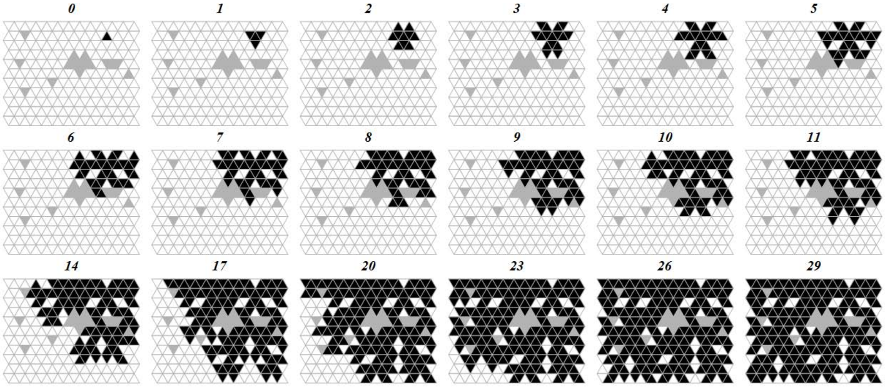


Fig. 11.16 stTCA # 250 at selected time steps starting from a single black cell. The pattern stabilizes at the 29th time step. White, gray and white facets indicate transparent facets, voids in the mesh and opaque facets, respectively.

further investigations including totalistic triangular cellular automata (tTCA) and the stabilization process see [25].

11.4.2 Cellular Automata on Triangulated Free-Form Surfaces

The application of stTCA on a free-form surface has been demonstrated in [63]. However, general rules have potential of allowing more direct control over the CASS mesh than totalistic and semi-totalistic rules. For a corresponding demonstration see [60]. Figure 11.19 shows three arbitrarily selected 2D2C TCAs: 9622, 44862, 65534 and applied on the BE shown in Fig. 11.1. For an interactive demonstration illustrating the application of those TCAs on a free-form surface with voids see [64].

11.5 Application of Evolutionary Algorithms for Optimization of CA Shading

As mentioned in Sect. 11.2.1, the sequence of initial conditions (SIC) is besides the local transition rules (TR) the second most fundamental factor determining the performance of CASS. Finding optimal SIC is much more straightforward than finding an automaton for CASS, nevertheless it is a NP-problem: the number of all possible SICs grows as factorial of the width of a shading array. Since it is impossible to exhaustively explore the entire search space, it is natural to use an meta-heuristics for finding, if not perfect, at least good or competitive solutions. Evolutionary algorithms (EAs) are a well established methodology in this field which has been successfully applied for a number of CA-related problems, e.g.: automated design of CA-based complex systems [54], evolving a non-uniform CA where each cell in the lattice does not use the same rule set [14]; finding Wolfram class IV, that is complex rules [7]; density classification task [38] and the parity problem [58] [35]; discovering and designing cell-state transition functions, where CA are designed to satisfy certain global conditions [15], etc. The application of EA for optimization of SIC for a 100 × 100 cell CASS has been presented in [67]. In that paper the objective is to minimize grayness monotonicity and pattern distribution error (GDE), which is called there the cost function (CF). The core parts of the

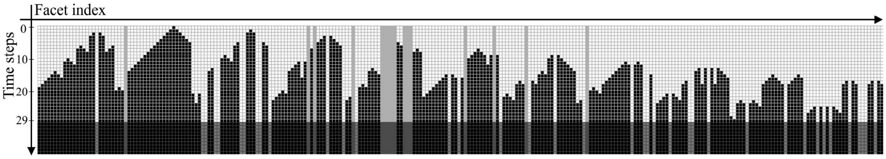


Fig. 11.17 The history of evolution from Fig. 11.16 shown as a single diagram. The shaded part after 29th time step indicates stabilization of the state of the mesh. White, gray and white indicate transparent facets, voids in the mesh and opaque facets, respectively.

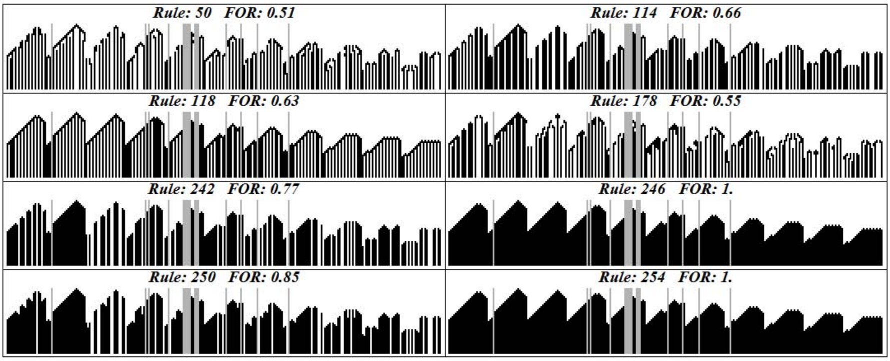


Fig. 11.18 The history diagrams of 40 time steps for stable stTCAs on the same triangular mesh

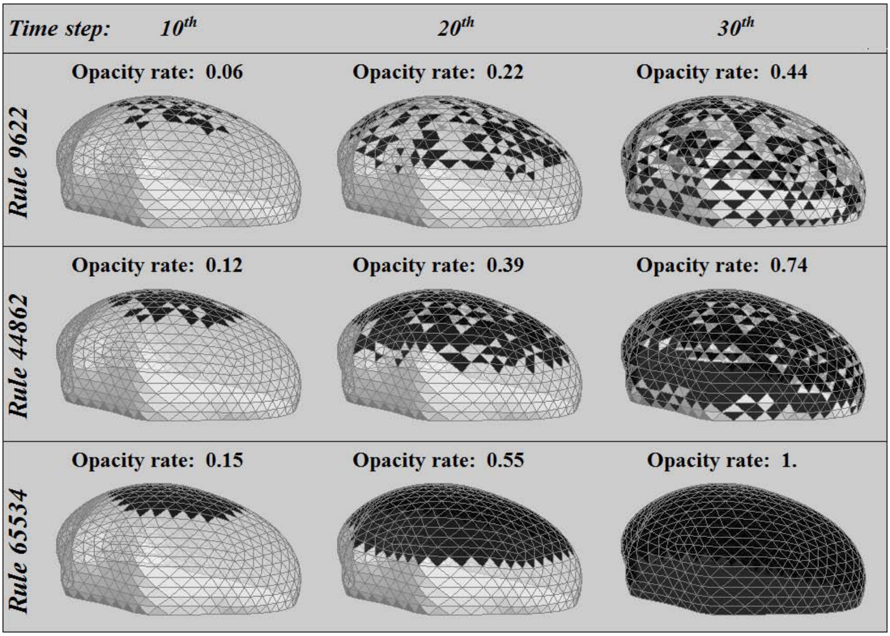


Fig. 11.19 Three general 2C2D TCAs applied on the BE from Fig. 11.1

application of EA for SIC optimization are presented and illustrated below. For further details, as well as the description of the application of evolution strategy [46], that is the simplest EA, where the genetic operations are limited to an intensive mutation and backtracking for the same task see [67]. The genetic operations are performed directly on a SIC* (genotype), that is an OBR-encoded SIC. Figure 11.20 shows examples of mutation and two types of recombination: one-point (OPX) and

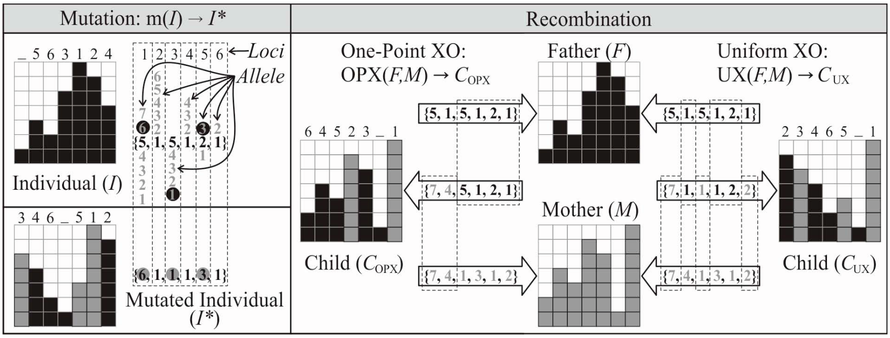


Fig. 11.20 Genetic operations on SIC*: On the left: mutation; on the right: two types of recombination

uniform crossovers (UX). The possible values of the genes, so called *alleles*, in such a genotype are constrained, and depend on the *locus*, that is the position of a certain gene in the genotype. Initially, since there are n available positions in the list, a gene can have n possible values. In the next step, since one position is already occupied, the allele is limited to n - 1, and so forth, up to the last gene in the genotype, which can only have two values: $n-((n-1)-1)=2$, as shown on the left of Fig. 11.20. Mutation is controlled by two parameters: mutation radius m_r the maximal difference of the mutated gene from the original value, and mutation intensity m_i, the maximal number of genes to be mutated, which is expressed as a ratio to the length of a genotype. The values of m_r and m_i in the example shown in Fig. 11.20 are 4 and 0.5, respectively. Figure 11.20 also illustrates two types of recombination: the one-point crossover (OPX) where a single point on both parents' gene strings is selected and the segments of genes are swapped between the two, and the uniform crossover (UX), where each gene is drawn with equal probability from both parents. The major advantage of OBR is that the operation of crossover (XO) is straightforward, and always yields feasible SIC* which represent allowable SIC. The process of adjusting the parameters for various evolutionary techniques for optimization of SIC for a 100×100 cell CASS based on the {3818817080,2,2} CA is described in [67]. Figure 11.21 compares the performance of five methods for an experiment of 10 trials of 200 generations of population of cardinality of 50: evolution strategy (ES), evolutionary algorithm with uniform crossover and mutation rates 0.1 and 0.4 (EA_{UX} 0.1 and EA_{UX} 0.1, respectively), EA with one-point XO and mutation rates 0.1 and 0.4 (EA_{OPX} 0.1 and EA_{OPX} 0.1, respectively). Good results generated by ES indicate that in this problem the operation of mutation is predominant. This influence is also evident in EA_{OPX} where the higher mutation rate produces better solutions. Good results obtained by EA_{UX} can also be attributed to the fact that UX introduces more intensive alterations to a genotype than OPX. However, the best overall result was obtained by EA_{OPX} indicating that the influence of crossover may be an increasing factor in a long run. The selection of the method depends on the designer's strategy. Table 11.4 collects the essential results. In a case of conservative

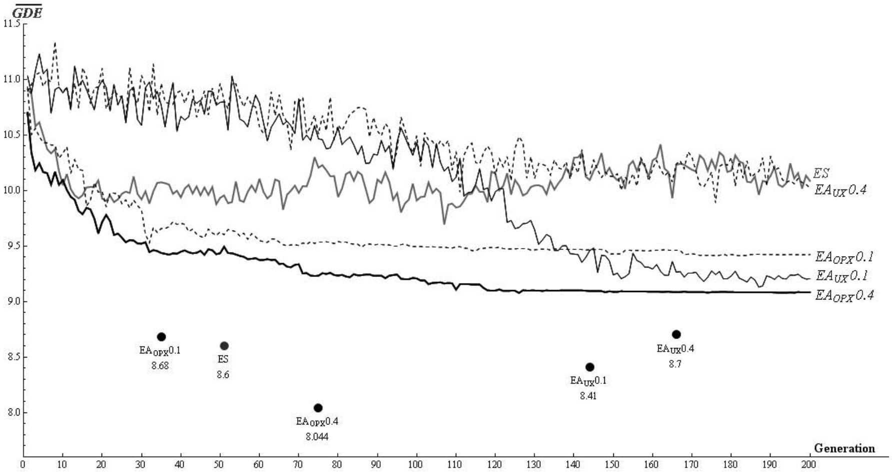


Fig. 11.21 The comparison of the means of the means of GDEs in a population of 50 for each generation in 10 trials for five evolutionary methods. The best solution for each method is indicated by a dot. EA_{OPX} with mutation intensity 0.4 produced consistently the best solutions.

Table 11.4 The results of the experiments for the 100 × 100 shading array

	ES	**EA**$_{UX}$		**EA**$_{OPX}$	
m_i	1	0.1	0.4	0.1	0.4
GDE$_{min}$	8.6	8.41	8.7	8.68	**8.04**
GDE$_{max}$	**9.39**	9.44	9.41	9.58	9.71
$\overline{\textbf{GDE}}_{min}$	9	**8.9**	9.1	9.1	9
σ_T	0.3	0.3	**0.2**	0.3	0.6

strategy, for example if only one trial is to be performed, ES should be chosen, since the worst result in all ES trials is better than in other methods. EA_{UX} with mutation rate m_r = 0.4 is the most predictable, in a sense that the best results in each trial are close and the mean value is competitive. However, EA_{UX} with m_r = 0.1 returns the best results on average, while EA_{OPX} with m_r = 0.4 gives the best overall result. Higher mutation rate (0.4) results in more scattered results, which, however, have the potential of reaching exceptionally good values. Therefore if a larger number of trials can be afforded, the latter method is recommended.

11.6 Prototypes

Although the research on CA commenced already over half a century ago, the physical devices based on them are still very scarce. The completion of the first hardware

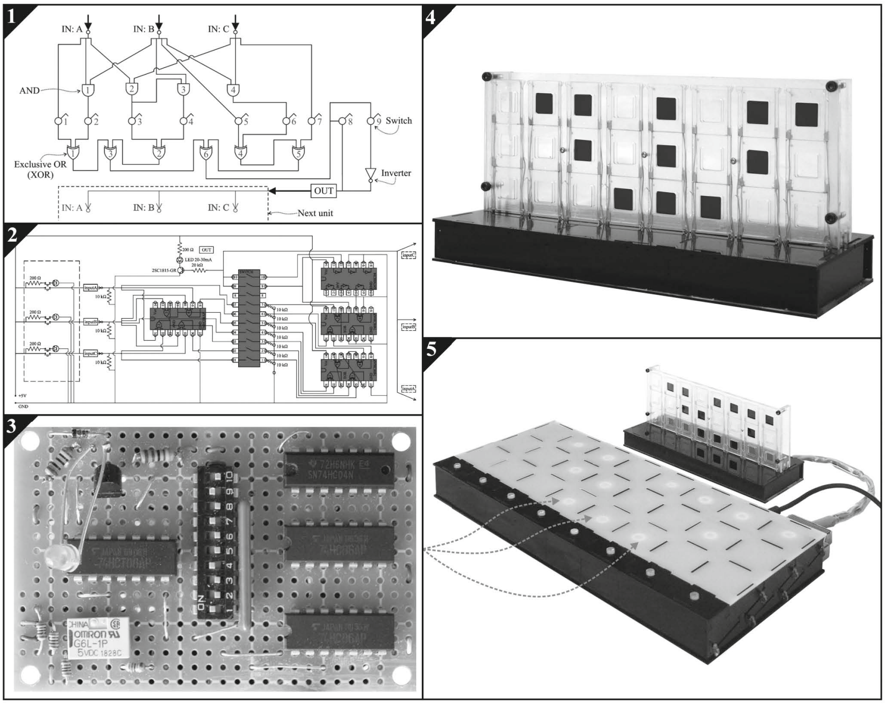


Fig. 11.22 The CASS prototype: 1): Scheme of the logic which emulates 256 ECAs. 2): Electrical diagram of a CA unit. 3): A photograph of a hardware CA unit. 4): SP with some LC elements active (opaque). 5): SP following the pattern produced by CU.

prototype of CASS has been documented in [68]. However, since it is based on ECAs which do not possess good shading properties, it can not be considered as CASS in the strictest sense. Nonetheless, the original motivation was also: to provide a physical educational device driven by CA; as an exercise on electric circuitry; to give students a hands-on experience of CA; to demonstrate the entire process from designing the circuit logic to fabrication of the CA modules; to teach the emergent properties of CA [17]. Therefore the logic was designed to allow emulation of all 256 ECAs by setting manually 8 simple switches at each CA module. Interestingly, this also allows experimentation with non-uniform CA [51], that is automata whose individual cell rules are not the same. Although the concept of CASS is based on interactions among autonomous units, the original prototype, for the reasons mentioned above, is comprised of two components: the CA control unit (CU) and the shading panel (SP) based on liquid crystal (LC) technology. That prototype is fully operational. Since the CA units are based on commonly available integrated circuits (CMOS), they are much larger than intended for the real application, which most likely would combine field programmable gate arrays (FPGA) with printed circuit boards (PCB) [44]. Due to high cost of LC elements, SP is much smaller than intended. Moreover, CU was built also to demonstrate “hardware automaton” [55], therefore each cell is equipped with a light emitting diode (LED) to display the CA action. This is also the reason why the CA circuits are not integrated with the shading elements, as it would be in the actual shading device. Figure 11.22 shows the prototype documentation.

Acknowledgements. This is part of Singapore University of Technology & Design and Massachusetts Institute of Technology Postdoctoral Program (SUTD-MIT PDP). The research is titled: *Effective computational methods for grid and raster-based modeling of practical problems in architectural and urban design.*

References

1. Alonso-Sanz, R.: A structurally dynamic cellular automaton with memory in the triangular tessellation. Complex-Systems 17(1), 1–15 (2007)
2. Aydinli, S., Seidl, M.: Determination of the economic benefits of daylight in interiors concerned with the fulfillment of visual tasks. In: Adepski, M., McCluney, R. (eds.) Proceedings I: 1986 International Daylighting Conference, Long Beach California USA, pp. 145–151 (1986)
3. Baas, N.A., Torbjorn, H.: Higher Order Cellular Automata. Advances in Complex Systems 8(2-3), 169–192 (2005)
4. Bays, C.: Cellular automata in the triangular tessellation. Complex-Systems 8, 127–150 (1994)
5. Bays, C.: Cellular Automata and the Game of Life in the Hexagonal Grid (2001), `http://www.cse.sc.edu/~bays/h6h6h6/`
6. Bays, C.: Cellular Automata in Triangular, Pentagonal and Hexagonal Tessellations. In: Meyers, R.A. (ed.) Computational Complexity, pp. 434–442. Springer, New York (2012)
7. Bilotta, E., Lafusa, A., Pantano, P.: Searching for complex CA rules with GAs. Complexity 8(3), 56–67 (2003)
8. Bodart, M., De Herde, A.: Global energy savings in offices buildings by the use of daylighting. Energy and Buildings 34(5), 421–429 (2002)
9. Boyce, P.: Why Daylight? In: Proceedings of Daylight 1998, International Conference on Daylighting Technologies for Energy Efficiency in Buildings, Ottawa, Ontario, Canada, pp. 359–365 (1998)
10. Boyce, P., Hunter, C., Howlett, O.: The benefits of daylight through windows: Report. Tech. rep., Rensselaer Polytechnic Institute, Troy, New York (2003)
11. Chavey, D.: Tilings by regular polygons II: A catalog of tilings. Computers & Mathematics with Applications 17(1-3), 147–165 (1989)
12. Connor, S.: The book of skin. Cornell University Press (2003)
13. Cuttle, C.: Lighting by Design. Elsevier, Amsterdam (2003)
14. Das, R., Mitchell, M., Crutchfield, J.: A genetic algorithm discovers particle based computation in cellular automata. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 244–353. Springer, Heidelberg (1994)
15. El Yacoubi, S., Jacewicz, P.: A genetic programming approach to structural identification of cellular automata. Journal of Cellular Automata 2, 67–76 (2007)
16. Euler, L.: Elementa doctrinae solidorum. Novi Commentarii academiae scientiarum Petropolitanae 4, 109–140 (1758)
17. Faraco, G., Pantano, P., Servidio, S.: The use of Cellular Automata in the learning of emergence. Computers & Education 47(3), 280–297 (2006)

18. Flache, A., Hegselmann, R.: Do Irregular Grids make a Difference? Relaxing the Spatial Regularity Assumption in Cellular Models of Social Dynamics. Journal of Artificial Societies and Social Simulation 4(4) (2001), http://jasss.soc.surrey.ac.uk/4/4/6.html
19. Formenti, E., Grange, A.: Number conserving cellular automata II: dynamics. Theoretical Computer Science 304(1-3), 269–290 (2003)
20. Frontczak, M., Wargocki, P.: Literature survey on how different factors influence human comfort in indoor environments. Building and Environment 46(4), 922–937 (2011)
21. Garzon, M.: Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks. European Association for Theoretical Computer Science. Springer, Heidelberg (1995)
22. Goldberg, M.: A Class of Multi-Symmetric Polyhedra. Tohoku Mathematical Journal 43, 104–108 (1937)
23. Grefenstette, J., Gopal, R., Rosimaita, B., Gucht, D.: Genetic Algorithms for the Traveling Salesman Problem. In: Proceedings of the 1st International Conference on Genetic Algorithms and their Applications, pp. 160–168. Psychology Press, Pittsburgh (1985)
24. Hanson, J.E.: Encyclopedia of Complexity and Systems Science. In: Meyers, R.A. (ed.) Cellular Automata, Emergent Phenomena, pp. 768–778. Springer (2009)
25. Zawidzki, M., Nishinari, K.: Controlling the opacity of a building envelope by a triangular two-color two-dimensional cellular automaton. In: Sirakoulis, G.C., Bandini, S. (eds.) ACRI 2012. LNCS, vol. 7495, pp. 194–203. Springer, Heidelberg (2012)
26. Imai, K.: A computation-universal two-dimensional 8 state triangular reversible cellular automaton. Theoretical Computer Science 231(2), 181–191 (2000)
27. Imai, K., Fujita, K., Iwamoto, C., Morita, K.: Embedding a Logically Universal Model and a Self-Reproducing Model into Number-Conserving Cellular Automata. In: Calude, C.S., Dinneen, M.J., Peper, F. (eds.) UMC 2002. LNCS, vol. 2509, pp. 164–175. Springer, Heidelberg (2002)
28. Johnston, P., Kelso, J., Milne, G.J.: Efficient simulation of wildfire spread on an irregular grid. International Journal of Wildland Fire (2008)
29. Kari, J.: Theory of cellular automata: A survey. Theoretical Computer Science 304(1-3), 3–33 (2005)
30. Kepler, J.: Harmonice Mundi. Lincii Austriae (1619)
31. Kiester, R.A., Sahr, K.: Planar and spherical hierarchical, multi-resolution cellular automata. Computers, Environment and Urban Systems 32, 204–213 (2008)
32. Kim, J.T., Kim, G.: Overview and new developments in optical daylighting systems for building a healthy indoor environment. Building and Environment 45(2), 256–269 (2010)
33. Konopka, A.K.: Systems Biology: Principles, Methods and Concepts Taylor and Francis. Taylor & Francis, Boca Raton (2006)
34. Laman, G.: On Graphs and Rigidity of Plane Skeletal Structures. Journal of Engineering Mathematics 4, 331–340 (1970)
35. Lee, K., Xu, H., Chau, H.: Parity problem with a cellular automaton solution. Physical Review E 026702(64), 1–4 (2001)
36. Madison, K.C.: Barrier function of the skin: Ąraison d'etre of the epidermis. Journal of Investigative Dermatology 121(2), 231–241 (2003)
37. O'Donoghue, D.P., Mullally, E.C.: Extending Irregular Cellular Automata with Geometric Proportional Analogies. In: Proceedings of the Geographical Information Science Research UK Conference, NUI Maynooth, Ireland, April 11-13 (2007)
38. de Oliveira, P., Bortot, J., Oliveira, G.: The best currently known class of dynamically equivalent cellular automata rules for density classification. Neurocomputing 70(1-3), 35–43 (2006)

39. O'Sullivan, D.: Exploring Spatial Process Dynamics Using Irregular Cellular Automaton Models. Geographical Analysis 33(1), 1–18 (2001)
40. Pegg, E.: Half-Distance Rules with Low Resolution (2000), `http://demonstrations.wolfram.com/HalfDistanceRulesWithLowResolution/` (an interactive demonstration)
41. Pegg, E., Zawidzki, M.: Cellular Shading (2008), `http://demonstrations.wolfram.com/CellularShading/` (an interactive demonstration)
42. Peters, H.M.: Functional organization of the spinning apparatus of Cyrtophora citricola with regard to the evolution of the web (Araneae, Araneidae). Zoomorphology 113(3), 153–163 (1993)
43. Proksch, E., Brandner, J.M., Jensen, J.M.: The skin: an indispensable barrier. Experimental Dermatology 17(12), 1063–1072 (2008)
44. Qasim, S.M., Abbasi, S.A., Almashary, B.: An Overview of Advanced FPGA Architectures for Optimized Hardware Realization of Computation Intensive Algorithms. In: Multimedia, Signal Processing and Communication Technologies (IMPACT 2009), pp. 300–303. Institute of Electrical and Electronics Engineers, Aligarh (2009)
45. Rado, T.: Uber den Begriff der Riemannschen Flache (in German). Acta Szeged 2(2), 101–121 (1925)
46. Rechenberg, I.: Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution. Ph.D. thesis, Stuttgart (1973) (in German)
47. Rothman, T.: Geodesics, Domes, and Spacetime. In: Science a la Mode. Princeton University Press (1989)
48. Ruck, N., Aschehoug, O., Aydinli, S., Christoffersen, J., Courret, G., Edmonds, I., Jakobiak, R., Kischkoweit-Lopin, M., Klinger, M., Lee, E.S., Michel, L., Scartezzini, J.L., Selkowitz, S.E.: Daylight in buildings: A Source Book on Daylighting Systems and Components. Tech. Rep. LBNL-47493, Lawrence Berkeley National Laboratory, Berkeley, California (2013), `http://gaia.lbl.gov/iea21/`
49. Saadatian, O., Sopian, K., Lim, C., Asim, N., Sulaiman, M.Y.: Trombe walls: A review of opportunities and challenges in research and development. Renewable and Sustainable Energy Reviews 16(8), 6340–6351 (2012)
50. Serra, R.: Chapter 6 – Daylighting. Renewable and Sustainable Energy Reviews 2(1-2), 115–155 (1998)
51. Sipper, M.: Co-evolving non-uniform cellular automata to perform computations. Physica D 92, 193–208 (1996)
52. Steiglitz, K., Kamal, I., Watson, A.: Embedding computation in one-dimensional automata by phase coding solitons. IEEE Transactions on Computers 37(2), 138–145 (1988)
53. Stevens, D., Dragicevic, S.: A GIS-based irregular cellular automata model of land-use change. Environment and Planning B: Planning and Design 34(4), 708–724 (2007)
54. Terrazas, G., Siepmann, P., Krasnogor, N.: An evolutionary methodology for the automated design of cellular automaton-based complex systems. Journal of Cellular Automata 2(1), 77–102 (2008)
55. Toffoli, T., Margolus, N.: Cellular Automata Machines. MIT Press, Cambridge (1987)
56. Ventrella, J.: Earth Day 2009 – A Spherical Cellular Automaton (2009), `http://www.ventrella.com/EarthDay/EarthDay.html`
57. Ventrella, J.: Glider Dynamics on the Sphere: Exploring Cellular Automata on Geodesic Grids. Journal of Cellular Automata 6(2-3), 245–256 (2011)
58. Woltz, D., de Oliveira, P.: Very effective evolutionary techniques for searching cellular automata rule spaces. Journal of Cellular Automata 3(4), 289–312 (2008)

59. Zawidzki, M.: Window Opacity Controlled By Cellular Automata (2008), http://demonstrations.wolfram.com/WindowOpacityControlledByCellularAutomata/ (an interactive demonstration)
60. Zawidzki, M.: 2D Triangular Cellular Automata on a Distorted Grid with Holes (2009), http://demonstrations.wolfram.com/2DTriangularCellularAutomataOnADistortedGridWithHoles/ (an interactive demonstration)
61. Zawidzki, M.: 2D2CR1 Cellular Automaton On a Triangular Grid (2009), http://demonstrations.wolfram.com/2D2CR1CellularAutomatonOnATriangularGrid/ (an interactive demonstration)
62. Zawidzki, M.: Delayed CA (2010), http://demonstrations.wolfram.com/DelayedCA/ (an interactive demonstration)
63. Zawidzki, M.: Application of Semitotalistic 2D Cellular Automata on a triangulated 3D surface. International Journal of Design & Nature and Ecodynamics 6(1), 34–51 (2011)
64. Zawidzki, M.: 2D Cellular Automaton on a Triangulated Surface (2013), http://demonstrations.wolfram.com/2DCellularAutomatonOnATriangulatedSurface/ (an interactive demonstration)
65. Zawidzki, M.: Dynamic shading of building envelope based on rotating polarized film system controlled by one-dimensional cellular automata on regular tessellations (triangular, square & hexagonal). Advanced Engineering Informatics (2013) (under review)
66. Zawidzki, M.: Implementing Cellular Automata for Dynamically Shading a Building Facade. Complex-Systems 18(3), 287–305 (2013)
67. Zawidzki, M., Bator, M.: Application of evolutionary algorithm for optimization of the sequence of initial conditions for the cellular automaton-based shading. Journal of Cellular Automata 7(5-6), 363–384 (2013)
68. Zawidzki, M., Fujieda, I.: The prototyping of a shading device controlled by a cellular automaton. Complex-Systems 19(2), 157–175 (2010)
69. Zawidzki, M., Nishinari, K.: Shading for building facade with two-color one-dimensional range-two cellular automata on a square grid. Journal of Cellular Automata 8(3-4), 147–163 (2013)