Published: Advances in Complex Systems 19(4,5) 1650012-18, 2016.
doi: 10.1142/S0219525916500120

Advances in Complex Systems

WAVES IN ISOTROPIC TOTALISTIC CELLULAR AUTOMATA: APPLICATION TO REAL-TIME ROBOT NAVIGATION

C. CALVO[1,*], J. A. VILLACORTA-ATIENZA[1,*], V. I. MIRONOV[2,*],
V. GALLEGO[1], V. A. MAKAROV[1,2†]

[1] *Instituto de Matematica Interdiciplinar, Universidad Complutense de Madrid, Avda. Complutense s/n, Madrid, 28040, Spain*
[2] *N.I. Lobachevsky State University, Gagarin Av. 23, Nizhny Novgorod, 603950, Russia*

Totalistic cellular automata (CA) are an efficient tool for simulating numerous wave phenomena in discrete media. However, their inherent anisotropy often leads to a significant deviation of the model results from experimental data. Here we propose a computationally efficient isotropic CA with the standard Moore neighborhood. Our model exploits a single postulate: the information transfer in an isotropic medium occurs at constant rate. To fulfill this requirement we introduce in each cell a local counter keeping track of the distance run by the wave from its source. This allows maintaining the wave velocity constant in all possible directions even in the presence of nonconductive local areas (obstacles) with complex spatial geometry. Then we illustrate the model on the problem of real-time building of cognitive maps used for navigation of a mobile robot. The isotropic property of the CA helps obtaining "smooth" trajectories and hence natural robot movement. The accuracy and flexibility of the approach are proved experimentally by driving the robot to a target avoiding collisions with obstacles.

Keywords: Cellular automata; Cognitive map; Cognitive navigation

1. Introduction

Simulation of different wave phenomena using cellular automata (CA) had its origins in the 1950s, when von Neumann, following the proposal made by S. Ulam, applied a fully discrete approach to describe the dynamics of spatially distributed systems. Later, due to the approach simplicity and computational efficiency, CA have become widespread in different fields such as hydrodynamics [1], physical chemistry [2], microelectronics [3], material science [4], and biophysics [5], among others. In the last decades CA have also been proposed as an efficient tool for finding a path among two or more locations in a map or among vertices in a graph, a problem tightly related to studies of the algorithms for robot navigation in complex environments (for short review see, e.g., [6]).

Although important advances have been achieved, one of the main disadvantages of the standard CA resides in their intrinsic anisotropy, which frequently contradicts

*These authors contributed equally.
†Corresponding author. e-mail: vmakarov@mat.ucm.es

the properties of modeled natural media and processes. For example, a roving robot in a free space can move equally in any direction, while a CA may "prefer" some of them depending on the lattice geometry. Figure 1 shows canonical examples of the excitations generated by a point source (single-cell). In all cases the wavefront significantly deviates from a circular shape typically expected in natural systems and thus reveals the anisotropic structure of the CA. Note that the anisotropy comes both from the discrete nature of the lattice and from the local rules defining the cell states at each time step.

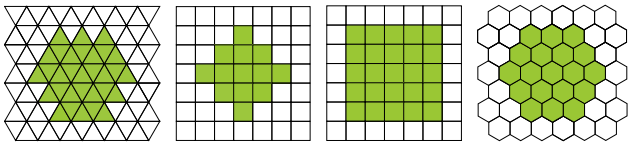

Fig. 1. Canonical forms of wavefronts (left to right): in triangular, square with the von Neumann neighborhood, square with the Moore neighborhood, and hexagonal lattices. All waves deviate significantly from circular shape.

Attempts to overcome the CA anisotropy were made in numerous studies (see, e.g., [2, 7–10]). In particular, Karafyllidis and Thanailakis [9] proposed a computationally efficient CA model for fast simulation of a forest fire, which involved global normalization (i.e., nonlocal knowledge). The model indeed reproduces a circular front, but its linear velocity decreases with time and it may not go through gaps among nonconductive local areas (obstacles). Later, Delorme and colleagues [10] showed that the circular wavefront in CA with binary states can be obtained with high-level algorithms only. Most of the recent works in this direction departure from the use of totalistic CA to automata with continuous state variables. Such models exploit the diffusion process (see, e.g., [2]) and/or the use of time delay [3] or virtually continuous time [11]. Then, the problem becomes similar to modeling reaction-diffusion processes by lattices of ordinary differential equations, known to produce circular fronts. Other methods employ extended Moore neighborhoods thus pushing the borders of local interactions in CA or make use of random variables, such as, e.g., random selection of neighboring cells [7, 12, 13]. Recently, Ortigoza [14] proposed an interesting approach, consisting in the use of a non-uniform triangular lattice. The method splits the space into finite elements and admits complex boundary conditions in problems focused on geographic applications. It has been shown that the model partially neutralizes the CA anisotropy. Nevertheless, designing an accurate and fast simulation by CA of the process of propagation of waves in isotropic media containing obstacles remains an open problem.

The applications of CA to the pathfinding problem range from the use of binary CA driven by simplistic rules [15–17] to modeling the foraging mechanism of slime mold [18]. Simple CA are attractive for the use in real-time tasks, but they usually exhibit significant pitfalls. For example, a non-Euclidian CA metric implies that the

trajectory length cannot be directly assessed [15, 17]. This can cause a situation when a CA finds a suboptimal path. Besides, the commonly considered assumption that the agent has eight possible directions of motion on a square grid leads to unnatural low efficient robot movements [19, 20]. Furthermore, most of the recent works search for the shortest path by starting the excitation in CA from the target position (see, e.g., [15, 19, 20]). This has two drawbacks: i) the target must be *a priori* designated, which sometimes is unfeasible [21] and ii) single solution significantly reduces the agent flexibility. Meanwhile, an advanced cognitive robot capable of interacting with humans will demand diverse criteria to judge the optimality of solutions beyond the trajectory length [22, 23].

Recently, we proposed to use the wave dynamics for building cognitive maps [24] (see also [25, 26] for biophysical details). In this case, a wave front starting from the agent position virtually explores the environment and generates a so-called compact cognitive map [23]. Such an *egocentric* map contains information on possible collisions with obstacles in the Euclidean metric and can be used for planning multiple trajectories to multiple targets. Thus, this approach naturally avoids most of the above mentioned limitations. Its drawback is the relatively high computational load, which stems from the modeling of a rather large 2D neural network. While other methods cannot achieve the desired real-time performance [27], the use of a fast isotropic CA for building compact cognitive maps may significantly improve the dynamic characteristics of robots.

In this work, we propose a model of an isotropic active medium based on traditional totalistic CA. To gain speed we departure from the common approach of discretization of the reaction-diffusion equation. Instead, at the heart of the model there is an analogy with the well-known Huygens' principle, whereby every point of the medium reached by the wave becomes a source of a secondary circular wave. Thus, we use local information representing the distance of each cell to the closest source of excitation. However, as we will show below in the case of a CA, secondary sources may also appear at certain cells of contact of the wave with obstacles. The simplicity and efficiency of the basic equations allow applying this algorithm to real-time tasks. We then illustrate the model on a problem of controlling the navigation of a mobile robot through building cognitive maps. In contrast to other CA methods, our approach provides a variety of smooth (physically plausible) trajectories to the target, and the agent can choose among them according to the particular situation by optimizing the length, safety, or cooperation.

2. Model of isotropic totalistic CA

Let us now introduce a CA model and then provide evidence on its isotropic property, i.e., a constant velocity of the wave propagation in all directions.

2.1. *CA model: Definitions*

We simulate a CA on a two-dimensional square $(L \times L)$-lattice:

$$\Lambda = \{(i,j) \in \mathbb{N}^2 : \ 1 \le i,j \le L\}. \tag{1}$$

The state of each cell (i,j) at discrete time instants $t = 1, 2, \ldots$ is described by two discrete variables: $a_t(i,j) \in \{0, 1, \ldots, L^2\}$ and $z_t(i,j) \in \{0, 1, \ldots, L^2\}^2$ (the upper limit, L^2, corresponds to the number of cells in the lattice). $a_t(i,j)$ accounts for the cell activation (cell (i,j) is activated if $a(i,j) > 0$ and not-activated otherwise) and keeps an information on the wave source that provoked the activity (see below), while $z_t(i,j)$ is an auxiliary counter that stores information on the propagation path to cell (i,j). The first component of $z_t(i,j)$ is the total number of steps performed to arrive to cell (i,j), whereas the second one is the number of diagonal steps among them (Fig. 2).

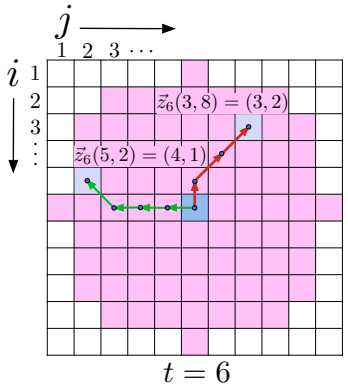

Fig. 2. Example of calculation of the auxiliary variable z_t for $t = 6$ (the wave source is in the center of the lattice). Although several shortest paths may lead to the same target cell, they provide unique counter value.

At $t = 1$ the CA is initialized by activating one or several cells (initial sources):

$$a_1(i,j) = \begin{cases} 1, & (i,j) \text{ is initial source} \\ 0, & \text{otherwise,} \end{cases}$$
$$z_1(i,j) = (0,0), \quad \forall (i,j) \in \Lambda. \tag{2}$$

It allows modeling situations with multiple or/and spatially extended initial sources. Then we denote the set of the activated cells in the CA at time t by:

$$A_t = \{(i,j) \in \Lambda : \ a_t(i,j) > 0\}. \tag{3}$$

A_1 thus corresponds to the set of initial sources. As we will see below, an activated cell cannot be deactivated and the number of the activated cells can only grow, thus $A_t \subseteq A_{t+1}$.

For further calculations we introduce the following sets:

- $B \subset \Lambda$ representing cells occupied by obstacles,
- $\Gamma \subset \Lambda$ corresponding to secondary wave sources (see below),
- $E_t = (\Lambda \backslash (A_t \cup B)) \cup \Gamma$ denoting the set of cells that can be activated ("empty" space).

The Moore neighborhood of a cell (i, j) can be defined as

$$M_{ij} = \{(k, l) \in \Lambda : \|(k - i, l - j)\|_\infty = 1\}. \tag{4}$$

Note that in our case $(i, j) \notin M_{ij}$. Then for each cell (i, j) we define a map[a] $r_t : M_{ij} \to \mathbb{N}^2$

$$r_t(k, l) = \begin{cases} (0, 0), & \text{if } (i, j) \notin E_t \vee (k, l) \notin A_t \\ z_t(k, l) + (1, \mathbb{1}_{D_{ij}}(k, l)), & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathbb{1}_{D_{ij}} : M_{ij} \to \{0, 1\}$ is the characteristic (indicator) function of the diagonal neighborhood:

$$D_{ij} = \{(k, l) \in \Lambda : |k - i||l - j| = 1\} \subset M_{ij}. \tag{6}$$

Note that $\|r_t(k, l)\|_2$ have the meaning of the distance to the corresponding wave source for all activated cells in the Moore neighborhood of (i, j), thus providing possible values for $z_{t+1}(i, j)$. Then, among these values, we will choose an optimal one (see below).

2.2. *CA model: Computational scheme*

Without loss of generality, we can assume that the velocity of a wavefront is equal to one. Then t characterizes also the distance that the wave has overcome at a given time step and the rules for updating the CA state are as follows:

Step I: For each cell $(i, j) \in \Lambda \backslash B$:

(1) Determine the set of cells from its Moore neighborhood that could *potentially* be sources of activation for the given cell (i, j):

$$W_t = \{(k, l) \in M_{ij} : \\ t < a_t(k, l) + \|r_t(k, l)\|_2 \leq t + 1\}.$$

(2) Using W_t determine the cell transmitting excitation to (i, j):

$$(i_t^*, j_t^*) = \begin{cases} \operatorname{argmin}\{\|r_t(k, l)\|_2 : (k, l) \in W_t\}, & \text{if } W_t \neq \varnothing \\ (i, j), & \text{otherwise.} \end{cases}$$

[a]To avoid the notation overloading we omit the index (i, j) in definition (5).

(3) Update the cell state:

$$a_{t+1}(i,j) = \begin{cases} t+1, & \text{if } (i,j) \in \Gamma \wedge (M_{ij} \cap A_t) \neq \varnothing \\ a_t(i_t^*, j_t^*), & \text{otherwise.} \end{cases}$$

$$z_{t+1}(i,j) = \begin{cases} z_t(i,j), & \text{if } (i,j) \notin E_t \backslash \Gamma \vee W_t = \varnothing \\ r_t(i_t^*, j_t^*), & \text{otherwise.} \end{cases}$$

Step II: Using definition (3) obtain the set of activated cells A_{t+1} from $\{a_{t+1}\}$.

Repeat Steps I and II in a loop until $t = T_{\max}$, where $T_{\max}$ is the first time instant when $A_{t+1} = A_t$ is satisfied.

2.3. *Wave propagation in empty space: Critical space scale*

Let us now test the algorithm in an empty space (i.e., $B = \varnothing$). Due to the discreteness of the space, time, and state variables the wavefront in a totalistic CA may differ significantly from the circular one (Fig. 1). We then introduce the following measure of the wave "circularity"

$$C(t) = 1 - \frac{\Delta S(t)}{\pi(t-1)^2}, \tag{7}$$

where $\Delta S(t)$ represents the measure of the symmetric difference between the set of activated cells and a circle of the corresponding radius (Fig. 3, red area in inset; for convenience the cell spatial dimension is (1×1) a.u.). Thus, $C = 100\%$ corresponds to a perfect circle, while low values of C indicate strong deviation of the wavefront from a circular shape.

We then simulated the process of propagation of a wave generated by a point source (single cell). Figure 3 shows the measure (7) and examples of the wave shape obtained by the algorithm for $t = 2, 3, 5$, and 15. It can be observed that for each $t \geq 2$ the centers of all activated cells belong to the closed disk of radius $t - 1$. Thus, our algorithm maximizes the measure (7) for waves propagating with constant velocity and hence the proposed CA indeed models an isotropic medium.

Insets given in Fig. 3 also show that at the beginning the wavefront is far from a circle and consequently the circularity measure takes low values. At $t = 5$, C reaches the value of 87%, i.e., the wave can be considered circular enough, and then we observe some "saturation" ($C(6) = 88\%$). The same critical scale ($t = 5$) also appears when dealing with the secondary wave sources at obstacle boundaries (see below). This observation has an important implication. Let us recall an analogy with the geometric and wave optics. Since for $t < 5$ the wave is qualitatively different from a circle, its interaction with obstacles on such space scales (less than 5 cells) can differ significantly from the process of wave propagation in open space. Thus, we can introduce a characteristic spatial scale for totalistic CA: $\lambda_{\text{ch}} = 5$ cells. In what follows, we will assume that the distance between adjacent obstacles is higher than λ_{ch}, i.e., we omit from consideration the effects of "interference" related to

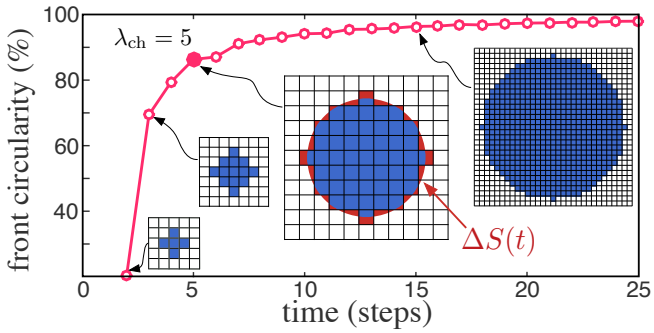

Fig. 3. Evolution of the measure of the front circularity during propagation of a wave initiated by a point source. At $t = 5$ (thick red dot) the wavefront is considered round enough. This time instant defines the characteristic length scale λ_{ch} (see main text for details).

wave optics. This model "granularity" should be taken into account in applications such as, e.g., robot navigation.

3. Secondary wave sources

3.1. *Isotropic CA in free space and front breaking at obstacles*

The algorithm described in Sect. 2.2 includes a set of "extra" secondary wave sources Γ. Before defining this set, let us consider a wave propagating in the CA without them, i.e., we assume $\Gamma = \varnothing$.

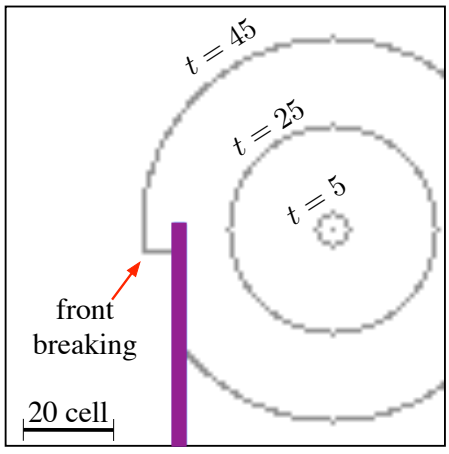

Fig. 4. Example of a circular wave propagating in empty space ($t = 5$ and $t = 25$; for visual clarity only the wavefront was drawn) and the front distortion ($t = 45$) at contact with obstacle (magenta vertical strip) in the absence of secondary sources, i.e., assuming $\Gamma = \varnothing$.

Figure 4 illustrates the process of propagation of a wave in a CA with a single

obstacle (magenta vertical bar). In empty space, i.e., before the contact with the obstacle, the wavefront is circular (measure (7) approaches 100%). Thus, once again the model indeed implements an isotropic medium. Nevertheless, interaction of the wavefront with obstacles may lead to unsatisfactory results. Depending on the obstacle curvature and characteristics of the wavefront at the moment of contact, the front shape may be distorted, causing the front "breaking" (Fig. 4, arrow).

The reason for breaking the front geometry while a wave bends an obstacle lies in the violation of the principle of causality. The updating of the cell states (Step I.3 of the algorithm) involves an implicit calculation of the distance to the original wave source, r_t. Eventually, this distance becomes distorted for certain cells that are "behind" the obstacle. In turn, this leads to a premature activation of these cells and disruption of the wavefront geometry.

3.2. *Interaction of wave with obstacles: secondary wave sources*

To get rid of the effect described in Sect. 3.1, we propose the concept of "additional secondary" wave sources, which is inspired by the Huygens' principle. Such sources appear at contact of the wavefront with certain cells on the obstacle boundary. As we show below the secondary sources prevent front breaking and thus we obtain correct wave propagation even in the presence of obstacles.

In what follows we will assume that each obstacle is a connected set of cells with a minimal "thickness" equal to two cells. To determine the cells corresponding to secondary sources (i.e., the set Γ) we provide the following algorithm:

(1) Determine *potential* secondary sources. Cell (i, j) is a potential secondary source if

$$\mathrm{card}(B \cap M_{ij}) \leq 4, \quad (i, j) \in B.$$

(2) Enumerate all potential secondary sources following counterclockwise along the obstacle boundary $\{(i, j)_q\}$, $q = 1, 2, \ldots, Q$. Then, define vectors:

$$(k_q, l_q) = (i, j)_{q+1} - (i, j)_q.$$

(3) Define secondary sources. A cell $(i, j)_q$ is a secondary source if

$$\lambda_{\mathrm{ch}}(k_{q-1} l_q - k_q l_{q-1}) > \max\{0, k_{q-1} k_q + l_{q-1} l_q\}.$$

Condition (1) selects "vertices" of the obstacles, whereas condition (3) discards among them those cells that involve small enough change in the direction between vectors (k_{q-1}, l_{q-1}) and (k_q, l_q). The critical value is defined by the granularity constant λ_{ch} (see Sect. 2.3). Note, that this algorithm is applied only once. Then the set Γ is used in the computational scheme described in Sect. 2.2.

Figure 5a illustrates the process of identification of the secondary wave sources for two obstacles. On the first step the cells candidates to be secondary sources are determined (condition (1), yellow and orange cells). Then a part of them is assigned as secondary sources in agreement with condition (3) (yellow cells).

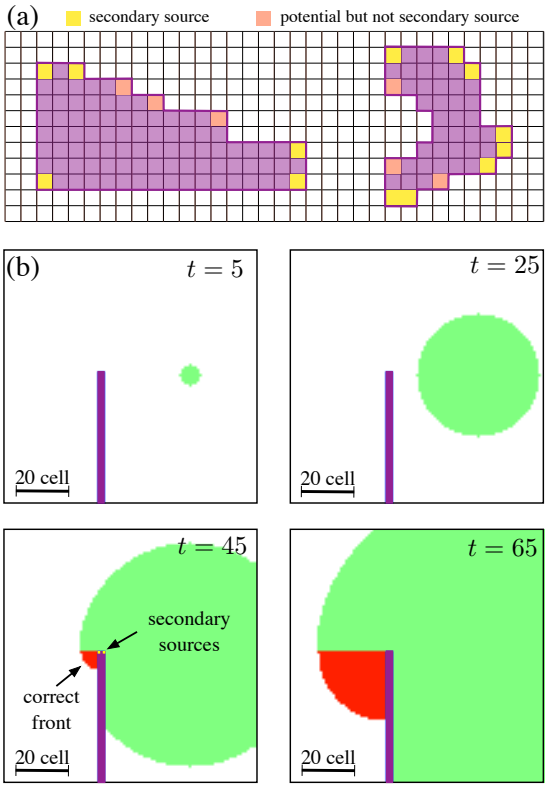

Fig. 5. Secondary wave sources. a) Illustration of the algorithm determining secondary wave sources. Cells on the corners of the obstacle boundary are potential secondary sources (yellow and orange). Only yellow cells satisfy condition (3) and consequently are assigned as secondary sources. b) Propagation of a circular wave and its correct interaction with obstacle (see Fig. 4 for comparison) due to the presence of the secondary wave sources (arrow).

To illustrate the effect produced by the secondary wave sources on the wave propagation, we repeated the numerical experiment shown in Fig. 4. Figure 5b shows the process of wave propagation in the presence of secondary wave sources. At the beginning ($t = 5$, $t = 25$) there is no difference with the previous calculation (see Fig. 4). However, when the wave reaches the tip of the obstacle, one of the secondary sources is activated and generates a "secondary" wave (drawn in red in Fig. 5b, $t = 45$). As a result, the wavefront keeps physically plausible geometry and we obtain a circular wave "behind" the obstacle, as expected.

4. Modeling wave interaction with complex obstacles

Let us now illustrate the process of wave propagation in a CA with multiple obstacles of different spatial geometries.

Figure 6 shows numerical simulation of a CA consisting of (400×400) cells.

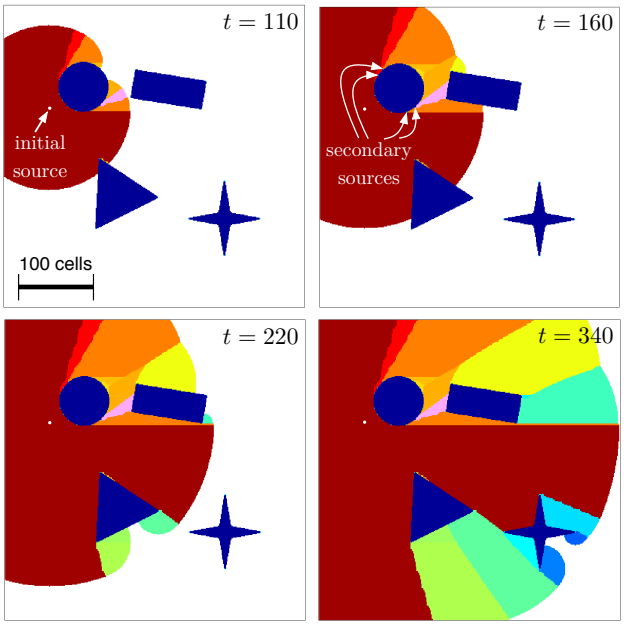

Fig. 6. Wave propagation in an isotropic (400 × 400) totalistic CA with obstacles of complex geometry (painted in dark blue). The color from red to blue corresponds to parts of the wave generated by different secondary wave sources. Circle obstacle generates the highest number of secondary waves (most colorful part of the wave).

A wave, generated by a single initial source (Fig. 6, $t = 110$, arrow), propagates among four obstacles (drawn in blue). When touching obstacles new secondary sources are activated and the state variable, $a_t(i, j)$, of these cells receives the value corresponding to the time instant of activation. Each of the activated secondary sources (Fig. 6, $t = 160$, arrows) generates a new wave. Consequently all cells receiving activation from a certain secondary source inherit this new value of the state variable a. This mechanism of changing $a(i, j)$ enables simple visual tracing of the propagation of excitation in the lattice. We painted the wave in different colors in line with the wave sources generating the corresponding part of the wave. In particular, dark red color corresponds to cells whose activation can be traced back to the initial source, whereas colors from red to blue mark parts of the wave generated by secondary sources (Fig. 6, $t = 220$ and $t = 340$).

Analyzing Fig. 6 we observe that the most complicated thing in terms of the number of secondary sources is the process of bending by the wave of the circle obstacle. This process involves seven secondary sources, while rounding rectangle requires three sources. Such a difference stems from the discrete nature of CA. Indeed, secondary sources appear on convex "corners" in the obstacle boundary (see Fig. 5a). Although in a continuous medium a circle has no corners, its discrete representation in the lattice creates multiple corners (Fig. 3, insets). The other three

obstacles (rectangle, triangle, and star) have simpler structures from the viewpoint of secondary sources. Therefore, the wave rounds such obstacles without exciting many secondary sources and we observe fewer color changes.

5. Application of CA to robot navigation

Let us now provide an example of a sensible application of the proposed CA model. As it has been mentioned in the introduction, the wave dynamics can be used for building cognitive maps, which in turn can be used for robot navigation (for details, see, e.g., [23, 24]).

Figure 7a illustrates an experimental setup consisting of a (255×345) cm arena simulating a corridor with obstacles (painted in orange). A wheeled robot, Pioneer 3DX (Adept Mobilerobotics, linear sizes (l×w×h): 45.5×38.1×23.7 cm), should cross the corridor avoiding obstacles and reach "exit" (blue strip). No direct route to the target is available.

To solve this task the robot was equipped with an onboard computer (NUC, Intel) with customary software packages written in Matlab (Mathworks). The computer was interfaced through WiFi connection to a zenithal camera and through a USB-COM adapter to the robot controllers driving the wheels.

The captured visual information (Fig. 7a) is preprocessed: the target, obstacles, and robot are detected and projected to the floor. Then the obtained abstract image of the situation is mapped into an (80×107) CA network (Fig. 7b). This network size (1 cell $\approx 3 \times 3$ cm) is a compromise between the computational load and navigation accuracy. Indeed, from the one hand side 3 cm is below 10% of the robot size, which is a reasonable tolerance for motion of animated agents. From the other side, taking into account the linear robot velocity of 30-40 cm/s, we obtain that the robot advances 1 cell in less than 100 ms, which is about the time necessary to process visual information. The robot also projects itself into the CA network. However, in the lattice space the robot occupies a single cell only (thick black dot in Fig. 7b). Then, to maintain the balance, its real dimension is properly added to the obstacles (magenta *vs.* orange color), thus proportionally increasing their sizes. Such a procedure facilitates searching for possible robot movements since it allows tracing trajectories regardless the actual robot size and its positioning among obstacles [28].

Once visual information has been properly mapped into the CA, the robot simulates the exploration of the environment by a wave process (Fig. 7c). Note that this exploration is done in internal or "mental" time τ and due to the computational efficiency of the CA model it is fast enough. The wave starts from the robot position (single cell) and finds all possible gaps between obstacles, and finally reaches the target. A quite complex intermediate wave profile at $\tau = 70$ evidences nontrivial possibilities of solving the navigation problem.

Finally, the wave dynamics builds a cognitive map of the given situation (for

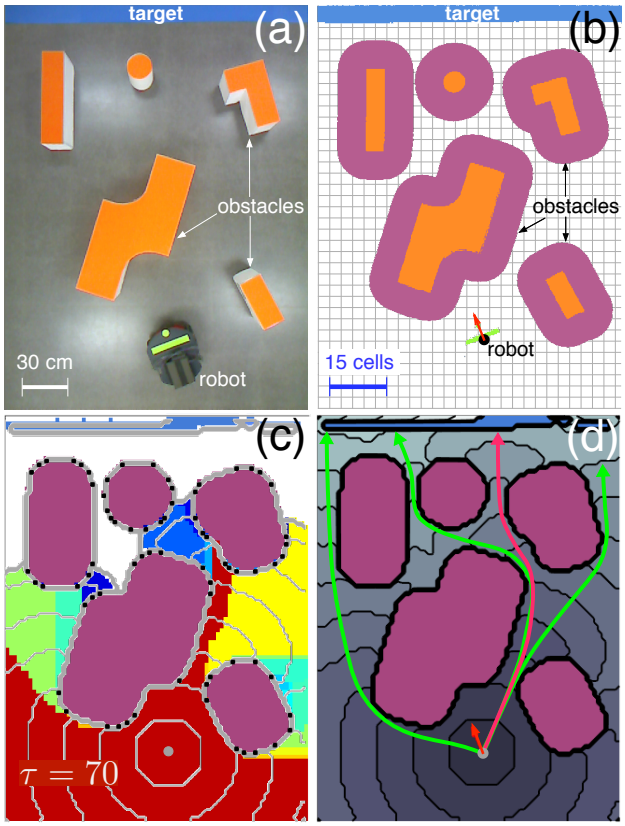

Fig. 7. Application of CA to cognitive navigation. a) Experimental setup (top view). A wheeled robot (Pioneer 3DX) moves along a corridor to an exit (blue stripe) avoiding obstacles (orange). b) Abstract visual perception of the situation shown in (a) and its mapping on the CA lattice. The robot is represented as a black dot (occupies one cell in CA), whereas all obstacles are increased by the robot dimensions. Red arrow indicates the robot head direction. c) A snapshot (at $\tau = 70$, τ is the simulation or "mental" time of the robot) of the process of building a cognitive map in the CA model. Black dots in obstacle boundaries represent secondary wave sources. Colors show the wave origin (as in Fig. 6). d) Cognitive map of the situation built by the CA and feasible trajectories (red one is the shortest). Contour lines show the configuration of the wavefront at different time instants.

details, see [24, 27]). The map is obtained by a local sum over time (Fig. 7d):

$$P_{ij} = \sum_{t=1}^{T_{\max}} \mathbb{1}_{A_t}(i,j). \tag{8}$$

Then, by the gradient descend method adapted to the discrete nature of CA we can trace several trajectories to the exit (target). Figure 7d shows four qualitatively different ways to reach the exit. The trajectory smoothness enables natural movement of the robot, not restricted to piecewise linear paths in eight direction commonly adopted in other works (see e.g., [15, 20]). The selection among different

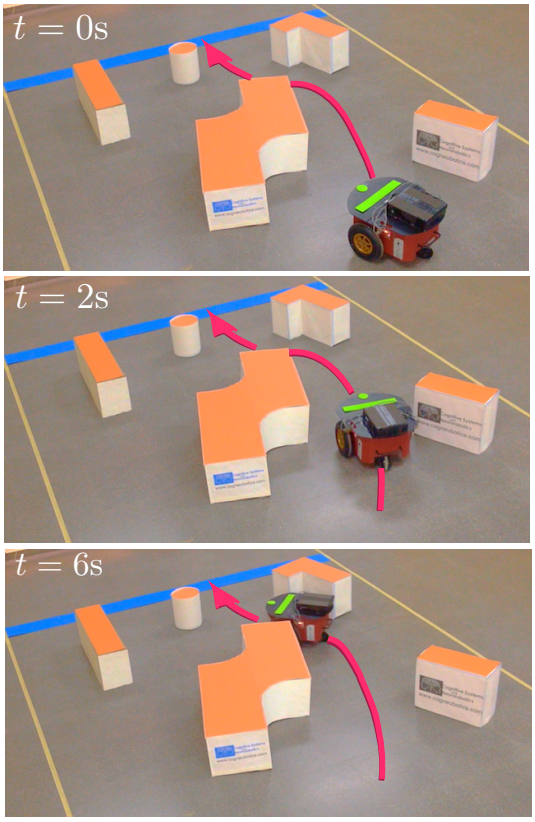

Fig. 8. Three consecutive snapshots of the robot movement. The robot follows one of the trajectories found by the CA (red in Fig. 7d). Thin yellow stripes delimit the "corridor" and blue stripe marks the "exit" (target). The video is available at: http://www.cogneubotics.com/research.html

trajectories requires a motivation criterium and in general can be achieved by using a motivation neural network [29]. Here we choose the shortest way to the exit (Fig. 7d, red curve).

Once a trajectory has been obtained, the robot can perform specific motor commands and, if the cognitive map has been built correctly, it will reach the target avoiding collisions with obstacles. Since our agent is a differential wheeled robot, it requires angular velocities as input [30]. Thus, instead of advancing to the next cell in the path in one of the eight discretized directions, the robot uses a smooth trajectory built up from appropriate arcs. As a result the robot moves along a physically plausible path without sharp changes of the direction requiring "stop-and-go" movements. We note that the Pioneer 3DX embedded controller estimates the position of the robot in space (through odometry), which we used in a feedback loop to ensure precise positioning of the robot. This approach allows reaching an accurate execution of trajectories. To crosscheck the motor accuracy we estimated

the deviation of the robot from a target point after following a trajectory of 6 m long. The error obtained on a circle trajectory was 2.9 ± 1.4 cm, while following an "8" shape trajectory it slightly increased up to 5.7 ± 2.5 cm, which is satisfactory for real-life applications.

Figure 8 shows three consecutive snapshots of the robot moving in the arena following the red trajectory shown in Fig. 7d. It successfully avoids obstacles and reaches the exit, as expected.

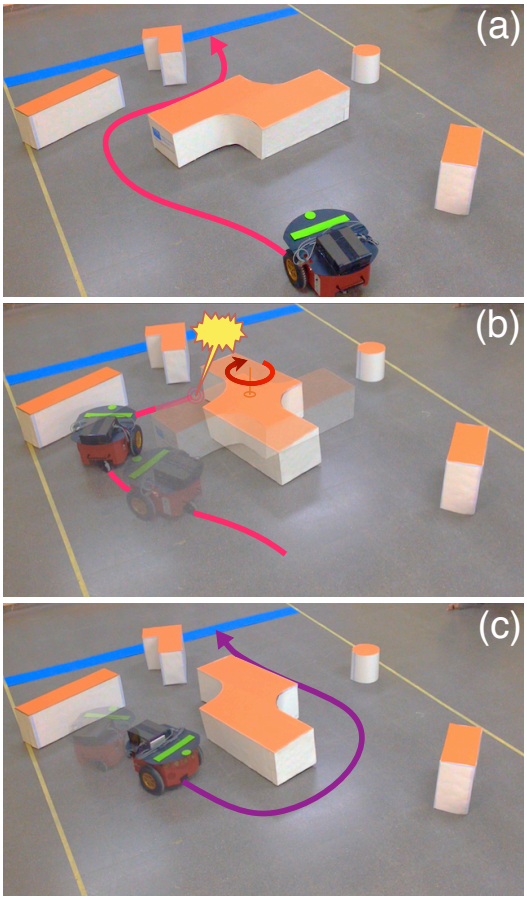

Fig. 9. Robot navigation in drastically and unexpectedly changing environment. a) Initial situation. The robot builds a cognitive map and starts moving to the target (similar as in Figs. 7 and 8). b) On the go the situation changes unexpectedly. The z-shape obstacle in the center of the corridor is rotated and blocks the robot path. The robot slows down and recalculates the cognitive map. c) New trajectory to the target is executed. The video is available at: http://www.cogneubotics.com/research.html

In experimental conditions the process of building a cognitive map takes around

300 ms, which is close to the human reaction time [31]. Taking into account the linear robot velocity 30-40 cm/s (robot displacement < 15 cm), a cognitive map can be obtained in real-time. We then repeated experiments in several different environments including time evolving. Figure 9 shows an example of such a situation. At the beginning the robot constructs a cognitive map corresponding to the initial situation and starts moving to the target (Fig. 9a). However, on the go the situation changes drastically: the z-shape obstacle is rotated unexpectedly (Fig. 9b) and blocks the path to the target. As soon as the robot realizes (< 100 ms) that the situation has been changed and no way to the target is available, it slows down and builds a new cognitive map. New map provides new trajectory to the target and hence the robot makes a turn and successfully reaches the target (Fig. 9c).

6. Conclusions

In this work we have proposed a model of a totalistic isotropic CA. It enables the simulation of the process of wave propagation in media with obstacles (areas not penetrable for waves). We note that our model differs significantly from other approaches. Earlier, similar results have been achieved with the use of CA described by continuous variables, simulating the diffusion equation (see e.g., [11, 32]). Instead of discretizing the Laplacian our model exploits a single postulate: "the information transfer in an isotropic medium occurs at constant rate". To achieve this property in a totalistic CA we introduced in each cell a local counter keeping track of the distance run by the wave from its source. This allowed maintaining the wave velocity equal in all directions. For correct interaction of waves with obstacles we introduced secondary wave sources in accordance with the Huygens' principle. Such sources appear at certain cells on the boundary of obstacles (at certain "vertices") and allow keeping physically plausible wavefront geometry. Numerical simulations have confirmed that indeed the proposed CA enables fast and correct modeling of the process of propagation of waves in isotropic media with complex spatial configurations.

Due to its minimalistic nature (simple local rules *vs.* extended Moore neighborhood used, e.g., in [7, 12]), our model is computationally efficient and it permits relatively simple hardware implementation on a chip, which may further increase the calculation speed [32, 33]. Thus, it can be useful for real-time tasks, such as, e.g., simulation of the heart muscle [34] or building cognitive maps to control mobile robots [23, 27]. To illustrate this ability we implemented the model on an onboard computer of a wheeled robot. The CA was used to build cognitive maps [26]. On the onboard computer this process lasted around 300 ms, independently on the complexity of the environment. This delay falls within the time scale of human reaction [31], which allows the robot to react promptly to unexpected changes in human environments [22, 23].

One of the important differences with other approaches (see, e.g., [15, 19, 20]) is the use of egocentric maps. The cognitive maps built by our model admit several alternative paths (*vs.* the common approach searching for the shortest trajectory)

and thus provide flexibility to accomplish the navigation task. Such an ability is an essential feature of cognitive agents. Then a robot can find optimal trajectories resembling human displacement, which is a must for pursuing the robots deployment in our daily life. In addition the egocentric algorithm does not require any *a priori* knowledge about the target. It just considers the perceived objects as entities, which can be reached or avoided according to the motivation and necessities of the agent. In this way the algorithm enables local and global robot navigation, increasing the versatility of the decision-making.

Real-time algorithms are for real world. To achieve natural, physically plausible robot movements we have payed much attention to the trajectory smoothness. This aspect, frequently overlooked by researchers (see, e.g., [16, 17, 19]), may significantly improve the robot performance in real-life scenarios. The pursued isotropic property of CA makes trajectories "smooth" up to the discreteness of the lattice. Then a complementary smoothing step eliminates the lattice granularity and facilitates the motor execution of trajectories. As a result the robot can advance naturally, avoiding sharp jumps, turns, and stop-and-go type of movement. Our experimental results have shown that: i) The robot equipped with the CA is able to build a cognitive map and when possible to find an optimal trajectory to a target (it may not exist). Then the robot successfully navigates to the target, smoothly avoiding obstacles. ii) If during the robot movement the situation is changed drastically, e.g., some obstacle changes position or another one appears unexpectedly and blocks the path, the robot is able to recalculate the map on the go. Then it selects another way to the target, and accomplishes the initial task (videos and Matlab code are available at: http://www.cogneubotics.com/research.html).

Acknowledgments

This work was supported by the Russian Science Foundation (project 15-12-10018).

References

[1] C. R. Boldea. A particle cellular automata model for fluid simulations. *Ann. Univ. Craiova: Math. Comp. Sci. Ser.* **36**, 35-41 (2009).

[2] I. Karafyllidis and A. Thanailakis. Simulation of the two-dimensional photoresist etching process in integrated circuit fabrication using cellular automata. *Model. Simul. Mater. Sci. Eng.* **3**, 629-642 (1995).

[3] Q. A. Huang, Z. F. Zhou, W. H. Li, and D. W. Xu. A modified cellular automata algorithm for the simulation of boundary advancement in deposition topography simulation. *J. Micromech. Microeng.* **16**, 1-8 (2006).

[4] K. Reuther and M. Rettenmayr. Perspectives for cellular automata for the simulation of dendritic solidification - A review. *Comput. Mater. Sci.* **95**, 213-220 (2014).

[5] A. Nishiyama, T. Tokihiro, M. Badoual, and B. Grammaticos. Modelling the morphology of migrating bacterial colonies. *Physica D* **239**, 1573-1580 (2010).

[6] P. Singhal and H. Kundra. A review paper of navigation and pathfinding using mobile cellular automata. *Int. J. Adv. Comput. Sci. Comm. Eng.* **2**, 43-49 (2014).

[7] M. Markus and B. Hess. Isotropic cellular automaton for modelling excitable media. *Nature* **347**, 56-58 (1990).
[8] I. A. French, D. H. Anderson, and E. A. Catchpole. Graphical simulation of bushfire spread. *Math. Comput. Model.* **13**, 67-71 (1990).
[9] I. Karafyllidis and A. Thanailakis. A model for predicting forest fire spreading using cellular automata. *Ecolog. Modell.* **99**, 87-97 (1997).
[10] M. Delorme, J. Mazoyer, and L. Tougne. Discrete parabolas and circles on 2D cellular automata. *Theoret. Comput. Sci.* **218**, 347-417 (1999).
[11] M. Marek. Grid anisotropy reduction for simulation of growth processes with cellular automaton. *Physica D* **253**, 73-84 (2013).
[12] G. C. Sirakoulis, I. Karafyllidis, and A. Thanailakis. A cellular automaton for the propagation of circular fronts and its applications. *Engin. Appl. Artif. Intell.* **18**, 731-744 (2005).
[13] M. Kobayashi. Isotropic cellular automaton for excitable media with random neighbor selection. *Lect. Notes Comp. Sci.* **8751**, 3544 (2014).
[14] G. M. Ortigoza. Unstructured triangular cellular automata for modeling geographic spread. *Appl. Math. Computat.* **258**, 520-536 (2015).
[15] C. Behring, M. Bracho, M. Castro, and J. A. Moreno. An algorithm for robot path planning with cellular automata. *Proc. 4th Int. Conf. Cell. Autom. Res. Indust.: Theor. Pract. Issues Cell. Autom.* 11-19 (2000).
[16] R. Al-Hmouz, T. Gulrez, and A. Al-Jumaily. Cellular automata based path-planning algorithm for autonomous mobile robots. *Proc. 16th IFAC World Cong.* Paper code: Tu-M04-TP/9 (2005).
[17] K. Charalampous, A. Amanatiadis, and A. Gasteratos. Efficient robot path planning in the presence of dynamically expanding obstacles. *Lect. Notes Comp. Sci.* **7495**, 330-339 (2012).
[18] Y. P. Gunji, T. Shirakawa, T. Niizato, M. Yamachiyo, and I. Tani. An adaptive and robust biological network based on the vacant-particle transportation model. *J. Theor. Biol.* **272**, 187-200 (2011).
[19] U. A. Syed and F. Kunwar. Cellular automata based real-time path-planning for mobile robots. *Int. J. Adv. Robot. Syst.* **11**, 93, doi: 10.5772/58544 (2014).
[20] Y. Tavakoli, H.H.S. Javadi, and S. Adabi. A cellular automata based algorithm for path planning in multi-agent systems with a common goal. *Int. J. Comput. Sci. Net. Secur.* **8**, 119-123 (2008).
[21] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi. Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Trans. Neur. Netw.* **20**, 1724-1739 (2009).
[22] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robot. Autonom. Syst.* **42**, 146-163 (2003).
[23] J. A. Villacorta-Atienza, C. Calvo, and V. A. Makarov. Prediction-for-CompAction: Navigation in social environments using generalized cognitive maps. *Biol. Cybern.* **109**, 307-320 (2015).
[24] J. A. Villacorta-Atienza, M. G. Velarde, and V. A. Makarov. Compact internal representation of dynamic situations: Neural network implementing the causality principle. *Biol. Cybern.* **103**, 285-297 (2010).
[25] J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* **34**, 171-175 (1971).
[26] E. I. Moser, E. Kropff, and M. B. Moser. Place cells, grid cells, and the brain's spatial representation system. *Neurosci.* **31**, 69-89 (2008).
[27] J. A. Villacorta-Atienza and V. A. Makarov. Neural network architecture for cognitive

navigation in dynamic environments. *IEEE Trans. Neur. Net. Learn. Syst.* **24**, 2075-2087 (2013).

[28] T. Lozano-Perez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM* **22**, 560-570 (1979).

[29] M. Schilling, J. Paskarbeit, T. Hoinville, A. Huffmeier, A. Schneider, J. Schmitz, and H. Cruse. A hexapod walker using a heterarchical architecture for action selection. *Front. Comput. Neurosci.* **7**, 126, doi: 10.3389/fncom.2013.00126 (2013).

[30] P. Petrov. Modeling and adaptive path control of a differential drive mobile robot. *Proc. 12th WSEAS Int. Conf. Autom. Contr. Model Simul.* 403-408 (2010).

[31] D. E. Meyer, A. M. Osman, D. E. Irwin, and S. Yantis. Modern mental chronometry. *Biol. Psychol.* **26**, 3-67 (1988).

[32] P. Progias and G. C. Sirakoulis. An FPGA processor for modelling wildfire spreading. *Math. Comput. Model.* **57**, 1436-1452 (2013).

[33] L. Salas-Paracuellos, L. Alba, J. A. Villacorta-Atienza, and V. A. Makarov. FPGA implementation of a modified FitzHugh-Nagumo neuron based causal neural network for compact internal representation of dynamic environments. *Proc. SPIE* **8068**, 80680J, doi: 10.1117/12.886911 (2011).

[34] R. Silva Campos, R. Mendonca Amorim, B. Lino de Oliveira, B. Martins Rocha, J. Sundnes, L. P. da Silva Barra, M. Lobosco, and R. Weber dos Santos. 3D heart modeling with cellular automata, mass-spring system and CUDA. *Lect. Notes Comp. Sci.* **7979**, 296-309 (2013).