Computing with liquid crystal fingers: Models of geometric and logical computation

Andrew Adamatzky

Unconventional Computing Centre, University of the West of England, Bristol, BS16 1QY, UK

Stephen Kitson

Information Surfaces Lab, HP Labs, Bristol, BS34 8QZ, UK

Ben De Lacy Costello

Centre for Analytical Chemistry and Smart Materials, University of the West of England, Bristol, BS16 1QY, UK

Mario Ariosto Matranga and Daniel Younger

HP Labs, Bristol, BS34 8QZ, UK

When a voltage is applied across a thin layer of cholesteric liquid crystal, fingers of cholesteric alignment can form and propagate in the layer. In computer simulation, based on experimental laboratory results, we demonstrate that these cholesteric fingers can solve selected problems of computational geometry, logic, and arithmetics. We show that branching fingers approximate a planar Voronoi diagram, and nonbranching fingers produce a convex subdivision of concave polygons. We also provide a detailed blueprint and simulation of a one-bit half-adder functioning on the principles of collision-based computing, where the implementation is via collision of liquid crystal fingers with obstacles and other fingers.

I. INTRODUCTION

Liquid crystals (LCs) are fluids composed of anisotropic (usually rod-shaped) molecules that exhibit long-range order [1,2]. In the simplest phase, known as the nematic, the molecules tend to align in a common direction, called the director n. In this paper we use a cholesteric phase in which the LC contains at least one component which is chiral so that the director adopts a helical structure through the LC. In a display, a thin layer of LC is enclosed between transparent substrates which contain electrodes to enable the application of a voltage.

LCs behave as elastic media and will adopt the director configuration that minimizes the elastic energy of the system. In addition the molecules have anisotropic dielectric properties so that applying an electric field will tend to cause them to rotate. The configuration adopted by the LC director will therefore be determined by the combination of the boundary conditions imposed by the inside surfaces of the substrates, as well as the interaction between the electric field and the LC molecules. In the devices considered in this paper the inside surfaces of the substrates are designed to induce homeotropic alignment, that is, with the director orthogonal to the substrates. The thickness of the LC layer is chosen to be close to the natural pitch of the cholesteric LC, which is the distance over which the director undergoes a full rotation. Under these conditions the natural tendency of the LC to twist is suppressed and the LC adopts a uniform untwisted configuration with the director orthogonal to the substrates. However, small perturbations to the system can upset this equilibrium and can cause the director to collapse into complex localized structures known as cholesteric fingers. These fingers consist of extended domains of twisted LC, and a rich variety of structures and phases has been reported [3–6]. Four distinct types of fingers have been identified [7].

In our system we perturb the equilibrium by applying a voltage. The LC that we use has a negative dielectric anisotropy, that is, the dielectric constant is greatest for electric fields directed orthogonal to the long axes of the rod-shaped molecules. The field, which is applied across the layer, tends to rotate the director away from the initial vertical state, and as the director becomes more planar the natural chirality of the LC dominates, resulting in the formation of cholesteric fingers. The anisotropic optical properties of the LC means that these fingers are clearly visible under an optical microscope. In this paper we enhance the contrast by adding dichroic dyes [8]. The fingers nucleate at defects or particles [3] in the LC and grow as the applied voltage exceeds a threshold and retract as it reduces below the threshold. For larger voltages the fingers start to branch, and as they fill the space the fingers start to repel each other.

Fingers have also been reported in other LC phases [9,10] and there have also been studies of isolated fingers which can propagate as a result of electroconvection in nematic LCs [11,12]. These propagating localizations in liquid crystal first attracted the attention of the unconventional-computation community in the early 2000s in the frame of collision-based computing. It was demonstrated in cellular automata models that worms in liquid crystals can implement basic logical gates when they collide with each other and either annihilate or deflect as a result of the collision [13]; however, no extended results were obtained at that time. Recent work in HP Labs has focused on the interaction between LC materials and engineered microstructures [14,15] and some of these systems have shown the potential for engineering the nucleation and propagation of cholesteric fingers. These results inspired us to reconsider the concept of collision-based computing in liquid crystals and to expand a range of problems solved by the propagating and interacting fingers.

The paper is structured as follows. In Sec. II we describe how we make our samples and measurements and in Sec. III we show how to imitate space-time dynamics of LC finger propagation using mobile automata on lattices. The classical problem of the planar Voronoi diagram is tackled by LC fingers in Sec. IV. Subdivision of a concave shape in convex shapes using LC fingers is demonstrated in Sec. V. We describe a design of a one-bit half-adder implemented via collision between fingers and obstacles, and between pairs of fingers, in Sec. VI. Final thoughts of future LC finger computing devices are outlined in Sec. VII.

II. FABRICATIONS AND MEASUREMENT OF DEVICES

Devices were constructed using glass substrates coated with a transparent conductor layer made from indium tin oxide (ITO). To control the nucleation of the cholesteric fingers, we patterned polymer structures onto one surface. Fingers tend to nucleate at the sharpest points on any structures. Figure 1 shows a typical set of features: 40-μm-diameter rings with four protrusions to seed the nucleation of fingers. The structures are 5 μm high and cover the ITO on one substrate. They are made from SU-8 (MicroChem Corp.) using photolithography.

The surfaces of the two substrates are then coated with a thin polymer layer to align the LC homeotropically. To assemble the cell, the substrates are gently placed in contact so that the SU-8 structures set the cell spacing, and the substrates are sealed on two edges with UV curing glue (NOA73 from Norland Products Inc.). The cell is then capillary filled with the LC mixture, with the cell heated to above the isotropic or nematic phase transition temperature to avoid the flow artifacts that can occur when filling with the LC in the nematic phase. The LC used was MLC2037 (Merck KGaA) which was chosen because it has a negative dielectric anisotropy, and the filling was carried out at 95 °C. It was doped with an additive (1.29% by weight of zli811, Merck) to impart a chiral pitch to the mixture. The concentration of the chiral dopant was chosen so that the pitch was very close to the cell spacing, so that the homeotropic state was just stable. The mixture was then doped with a blend of dichroic dyes (1% by weight G232, G241, and G472 from Hayashibara Biochemical Laboratories, Inc.). These enhance the contrast and make the fingers visible, without the need to use polarizers [8]. Dichroic dyes are rod-shaped molecules that only absorb light polarized along their long axis. They align with the LC, so in the quiescent state the
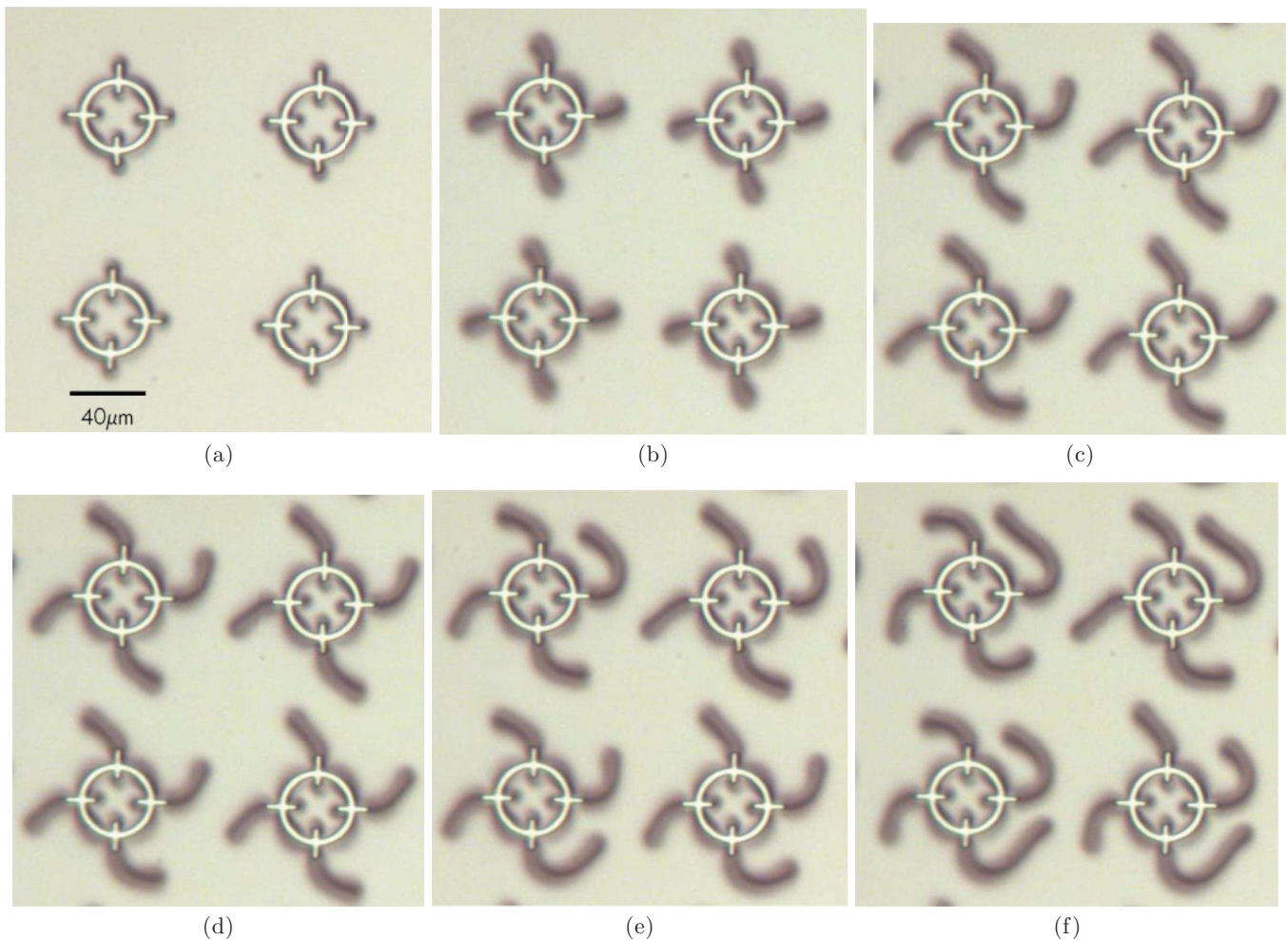

FIG. 1. (Color online) Experimental laboratory snapshots (optical microscopy) of colliding LC fingers, taken with an applied 1-kHz ac voltage with an amplitude of 1.5 V. Each photo is taken at a different time: (a) before the voltage is applied, and (b) 4 s, (c) 25 s, (d) 40 s, (e) 55 s, and (f) 80 s after the voltage is applied.
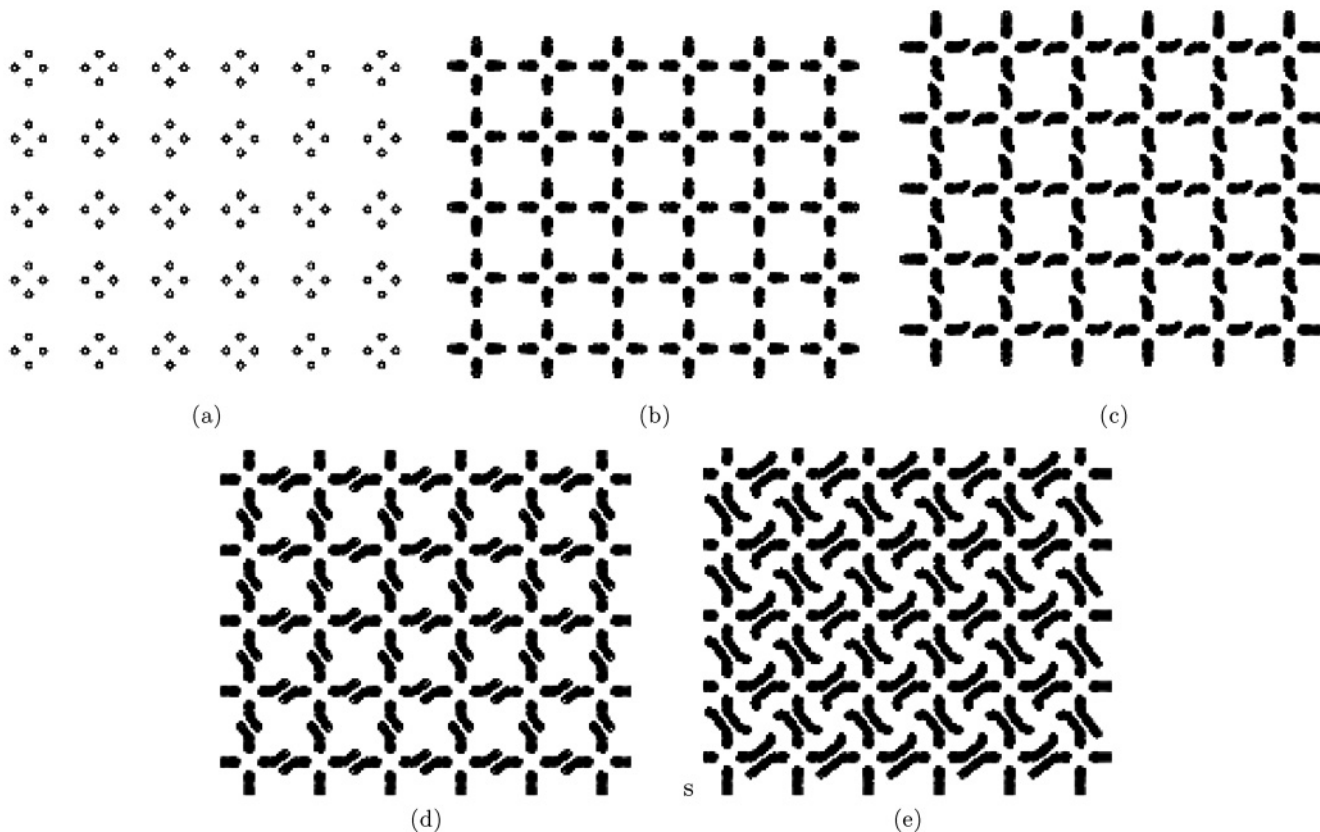
FIG. 2. Simulation of experimental results, shown in Fig. 1, using mobile automata.

LC and dyes are both aligned vertically. Thus the dyes absorb very little light. When the fingers form, the LC and dyes at least partially align at a more planar angle, so the dyes absorb more light and the fingers appear black. Applying a voltage across the LC layer causes the fingers to nucleate from the sharp points of the structures. These then slowly extend while the voltage is maintained (Fig. 1), and then retract when the voltage is removed. We use a 1-kHz sinusoidal voltage with an amplitude
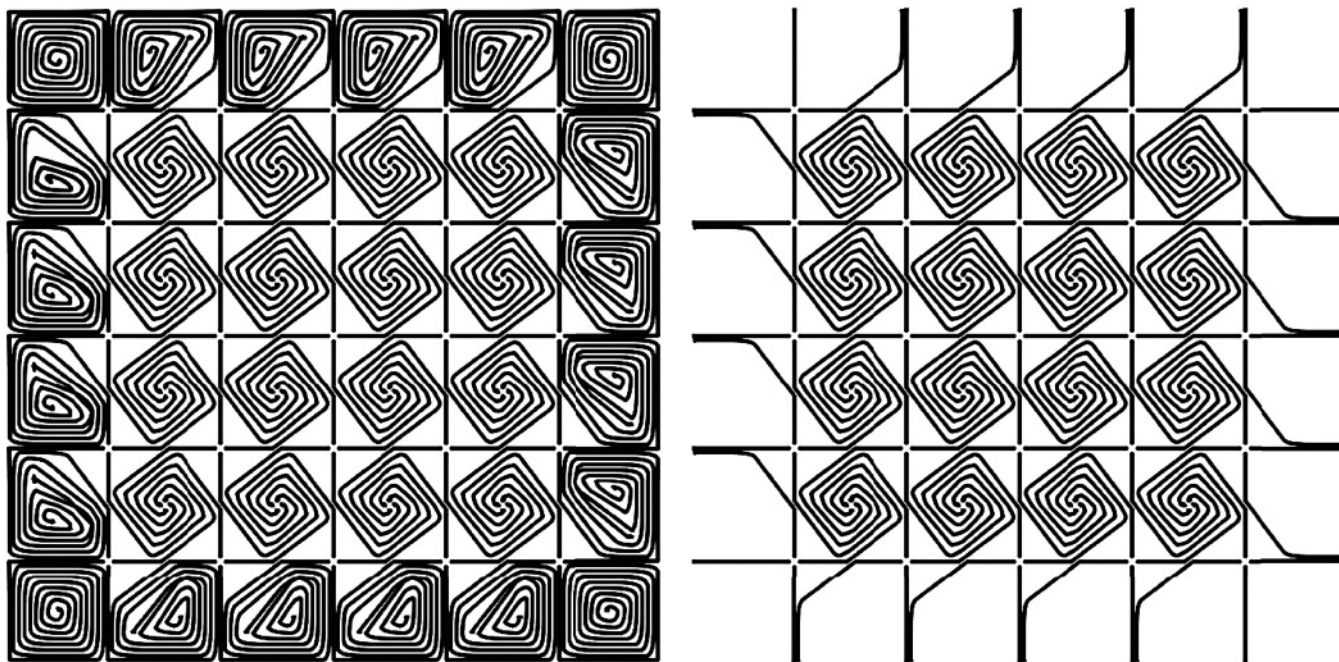

FIG. 3. Simulated interaction between fingers when distance between initial seeds is large enough to allow prolonged movement. (a) Solid (impassable) boundaries. (b) Boundary-less space.

of around 1.5 V. In the next section we describe a model that captures some of the behavior of the cholesteric fingers.

III. MOBILE-AUTOMATON MODEL OF LIQUID CRYSTAL FINGERS

A mobile automaton is a tuple $A = \langle \mathbf{c}, s, f, \alpha, d, \rangle$, where $\mathbf{c} \in \mathbf{R}^2$ is a Euclidean coordinate of the automaton, $\mathbf{c} = (x, y)$; function $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$ transforms coordinates as $x^{t+1} = x^t + d \sin \alpha$ and $y^{t+1} = y^t + d \cos \alpha$; in models presented here $d = 0.5$. The automaton moves on a lattice L. All nodes of L are in state "empty" (0) initially. At each step t of evolution time the nodes are updated as follows: for every $u \in L$: $u^t = 1$ if there is an automaton A with coordinates u. The "occupied" state is absorbing. Obstacles are also defined as domains of occupied states "1".

Figure 1 shows a sequence of microscope photos of fingers propagating away from a dense array of nucleation sites. With no applied voltage [Fig. 1(a)] the director will be distorted around each nucleation site [16]. As the voltage is increased the fingers nucleate and grow away from the polymer structures [Figs. 1(b)–1(f)]. As fingers approach each other they deviate from a straight path to avoid contact. We find that the fingers always tend to turn in the same direction, left in the case of Fig. 1. We believe this is due to the natural chirality of the LC mixture. The model is based on these observations. An automaton A chooses where to move as follows. If node $(x^t + d \sin \alpha, y^t + d \cos \alpha)$ is empty (0), then A moves into this node; otherwise it rotates left: $\alpha = \alpha + \pi/360$ and its state s is incremented. The exact angle of scattering is not known and may depend on many factors beyond our present knowledge. Thus we adopted increment $\pi/360$ whose α is increased until a "free" site for the next position is found. The automaton stops when s exceeds a certain threshold of attempts.

The mobile-automaton model is phenomenological. A mobile automaton moving in a two-dimensional discrete space very roughly imitates the tip of a LC finger propagating in a quasi-two-dimensional space being squashed between two electrodes. Our model in no way competes with existing numerical models of LC fingers, see, e.g. [17], but must rather be considered as a fast-prototyping tool in the design of LC finger computing algorithms.

We qualitatively verified our model using experimental observations. For example, if we consider Fig. 1, the discs with four singularities or protrusions are arranged in a regular grid so that the protrusions are aligned between neighboring discs (the protrusions of four neighboring discs describe a square array) [Fig. 1(a)]. When a voltage of 1.5 V is applied, LC fingers nucleate near the protrusions [Fig. 1(b)]. The fingers propagate along the original axis determined by the protrusions. When two fingers come into a head-on collision, each of them turns left [Figs. 1(c)–1(f)]. To imitate this experimental finding we regularly and uniformly distributed clusters of mobile automata in a two-dimensional space (Fig. 2). Initially the four automata of each cluster have their velocity vectors angles α oriented north, south, west, and east [Fig. 2(a)]. When finger f' moving north collides with finger f'' moving south the finger f' turns south-east, and finger f'' north-west [Figs. 2(b)–2(d)]. Similarly, the finger initially traveling west deviates south-west and the finger traveling east

turns north-east. Further changes of the fingers' trajectories are attributed to subsequent collisions [Fig. 2(e)]. If the distance between the initiation sites of the fingers is large enough to allow prolonged movement of the fingers, the four neighboring fingers form spirals (Fig. 3).

IV. APPROXIMATION OF VORONOI DIAGRAM

Let $\mathbf{P}$ be a nonempty finite set of planar points. A planar Voronoi diagram of the set $\mathbf{P}$ is a partition of the plane into such regions that, for any element of $\mathbf{P}$, a region corresponding to a unique point p contains all those points
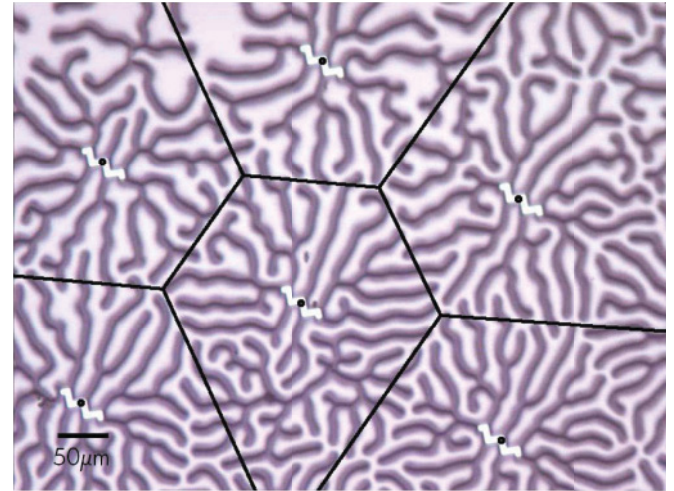
(a)
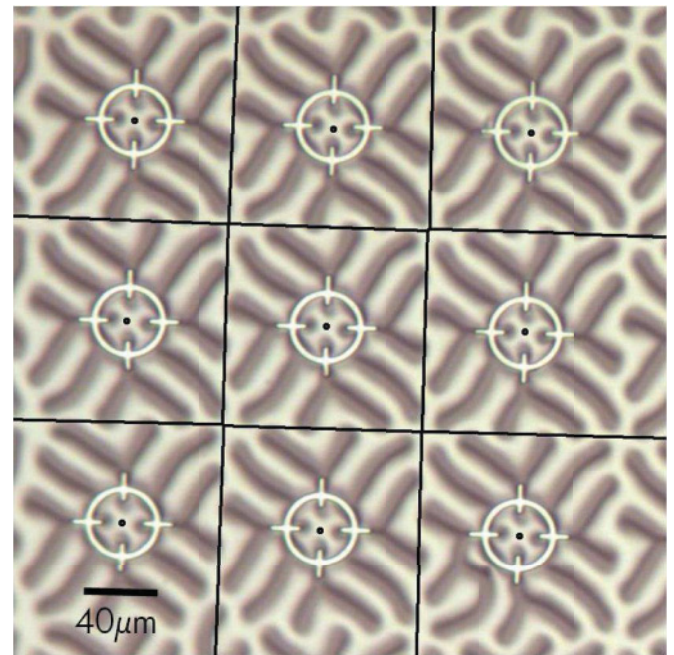
(b)

FIG. 4. (Color online) Experimental image of interaction between branching LC fingers originating from several seeds. The central point of the seeding structures are marked by small black discs. The edges of the Voronoi diagram are calculated using a classical sweep-line algorithm [31] and are drawn as solid lines superimposed on the experimental image.

of the plane which are closer to p than to any other node of **P**. A unique region $\mathrm{vor}(p) = \{z \in \mathbf{R}^2 : d(p,z) < d(p,m) \,\forall\, m \in \mathbf{R}^2, m \neq z\}$ assigned to the point p is called a Voronoi cell of the point p. The boundary of the Voronoi cell of the point p is built up of segments of bisectors separating pairs of geographically closest points of the given planar set **P**. A union of all boundaries of the Voronoi cells determines the planar Voronoi diagram: $\mathrm{VD}(\mathbf{P}) = \cup_{p \in \mathbf{P}} \partial\mathrm{vor}(p)$ [18]. Voronoi diagrams are applied in many fields of science and engineering. A few books and conference proceedings are available on the theory and applications of the Voronoi diagram [19,20].

Construction of a Voronoi diagram is a classical problem of unconventional-computing devices. This was the first problem ever solved in a reaction-diffusion chemical computer [21]. The basic concept of constructing Voronoi diagrams with reaction-diffusion systems is based on an intuitive technique for detecting the bisector points separating two given points of the set **P**. If we drop reagents at the two data points, the diffusive waves, or phase waves if the computing substrate is active, travel outward from the drops. The waves travel the same distance from the sites of origin before they meet one another. The points where the waves meet are the bisector points; see the extensive bibliography in [22,23]

and mechanisms of bisector formation in chemical media in [24–28]. The Voronoi diagram is also approximated in crystallization-based processors [29] and colonies of acellular slime mold *Physarum polycephalum* [30]. Thus it is intuitive to ascertain how Voronoi diagrams could be approximated with LC fingers.

The behavior of the fingers is voltage dependent. When the voltage exceeds a critical voltage (in this case 1.7 V), the fingers grow much more quickly and start to branch (Fig. 4). Recursively branching fingers form fronts or a phase boundary between a free space and the domain filled by fingers. Fronts originating from different sources compete for space. When two or more fronts meet, they stop propagating. Thus to approximate a Voronoi diagram we place the seeding structures of fingers in positions of space corresponding to planar points from **P** and increase the voltage to cause finger branching. The locus of space not occupied by any fingers represents the edge of the Voronoi diagram $\mathrm{VD}(\mathbf{P})$. From experimental observation it can be concluded that the accuracy of Voronoi diagram construction is improved when fingers are highly branched (approximating a continuous expanding front emanating from a defined site). This can be achieved by controlling the voltage or maximizing the density of fingers initiated by altering the
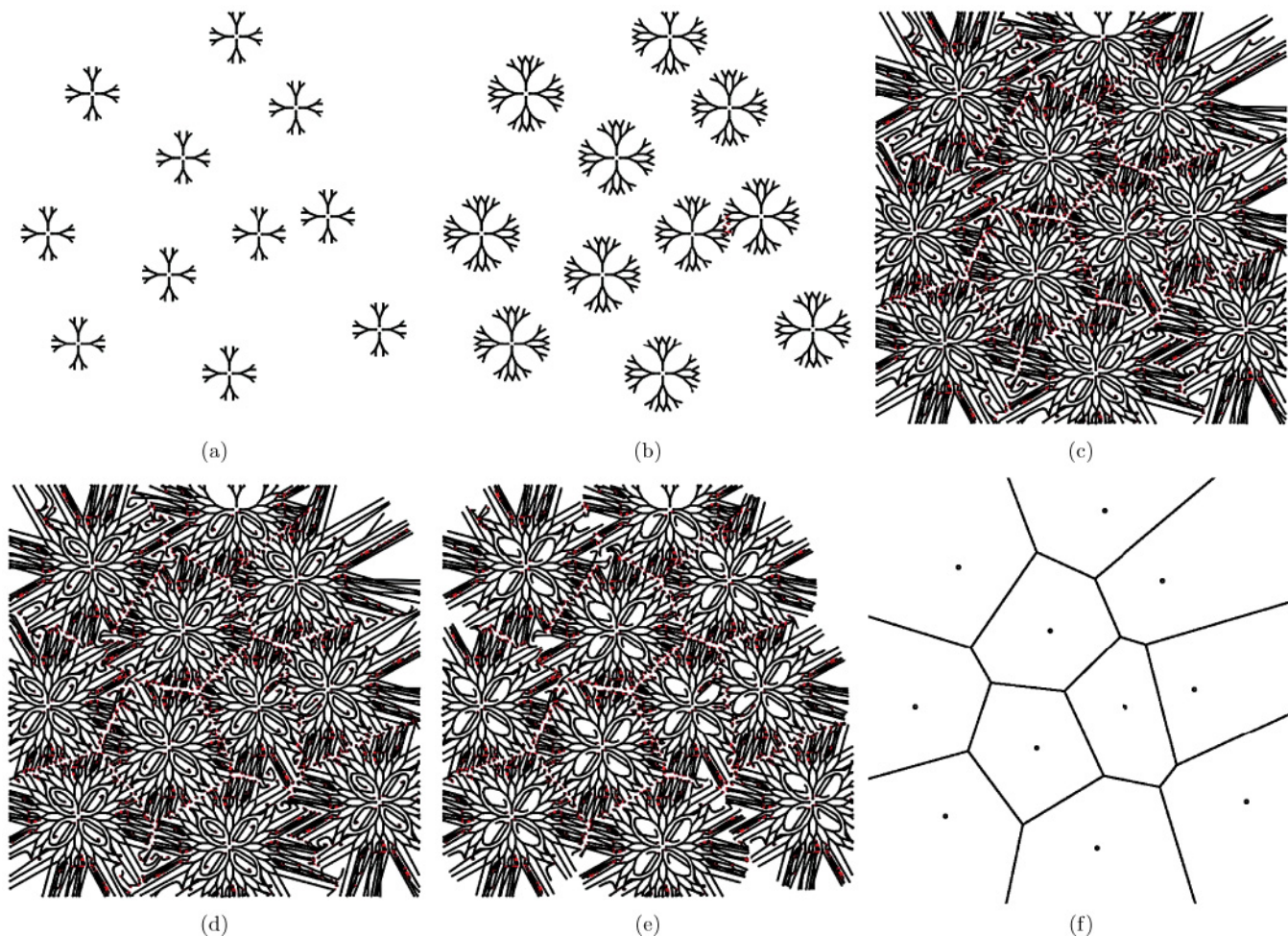

FIG. 5. (Color online) Simulated approximation of Voronoi diagram. (a)–(e) Snapshots of the simulated dynamics of branching LC fingers. Edges of the Voronoi diagram calculated using a classical sweep-line algorithm [31] are drawn by solid red (gray) lines in (a)–(e) and shown explicitly in (f).

geometry and the positioning of the seeding structures (for example, compare the constructions in [Figs. 4(a) and 4(b)]. The approximation of the Voronoi diagram of 11 planar sites in the mobile-automaton model of LC fingers is shown in Fig. 5.

V. CONVEX SUBDIVISION OF CONCAVE POLYGONS

A concave polygon is a shape comprised of straight lines with at least one indentation, or angle pointing inward. The problem is to subdivide the given concave shape into convex shapes. The problem is solved by LC fingers as follows. Fingers are generated at the singular points of indentations and the fingers' propagation vectors are co-aligned with medians of the corresponding inward angles. Given a concave polygon, every indentation initiates one propagating finger. A finger turns left, relative to its vector of propagation, when it collides with another finger or a segment of the polygon. Given a polygon with n indentations, n fingers will be generated. By following their "turn-left-if-there-is-no-place-to-go" routine and also competing for the available space with each other, the fingers fill $n-1$ convex domains. At least one convex domain will remain unfilled.

In the example shown in Fig. 6(a), there are three indentations. They initiate fingers propagating, approximately, south (a), west (b) and north-north-east (c). The finger traveling south (a) collides with a finger traveling north-north-east (c). In the result of the collision, finger a turns left and moves south-east. Finger c turns left and moves north-west-west [Fig. 6(b)]. At some stage, finger b collides into finger c and finger a collides into finger c. When fingers collide into the "walls" of the polygon, they turn left and more or less accurately follow the wall until the next collision [Fig. 6(c)]. Each finger forms a spiral and becomes stuck inside the spiral at some point of its growth.

The computation of the convex subdivision of a concave polygon is completed when all fingers cease moving. The result of the computation is a set of convex polygons, each polygon is filled by a unique finger and at least one unfilled convex domain. In the example provided, a concave polygon is subdivided into four domains. The north-eastern domain is constructed by finger a, the south-eastern domain by finger c and the south-western domain by finger b. The north-western domain remains empty [Fig. 6(d)]. In the case of regular polygons, e.g., star-shaped in Fig. 6(e), boundaries between
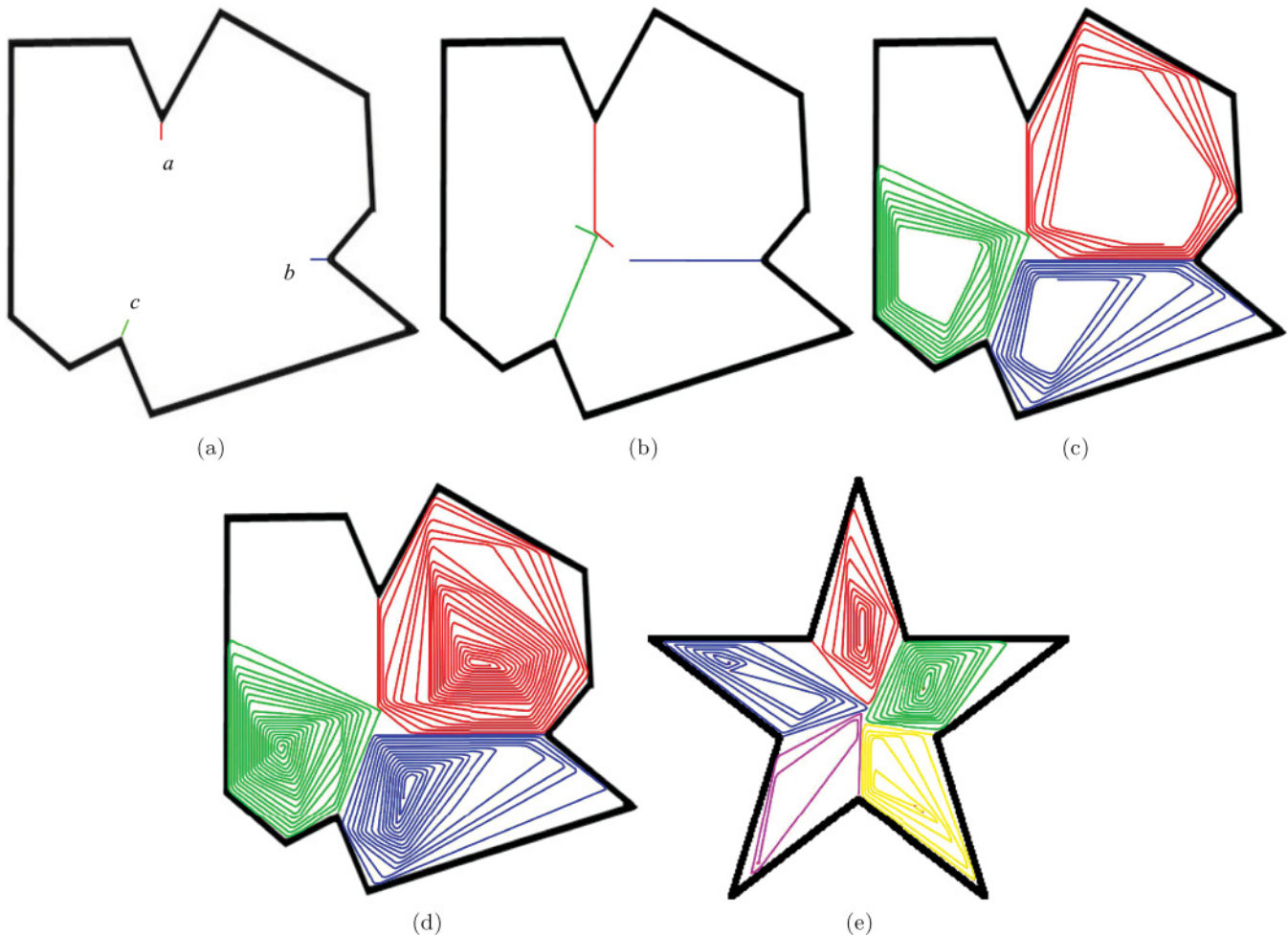

FIG. 6. (Color online) Convex subdivision of concave polygons in mobile-automaton model of LC fingers. Snapshots of the space after the propagation of all fingers has stopped. (a)–(d) Snapshots of LC finger model subdividing a concave polygon with three indentations. (e) Convex subdivision of star-shaped polygon.

finger-filled convex domains correspond to the skeleton of this planar polygonal shape.

VI. ONE-BIT HALF-ADDER

The half-adder presented is based on the paradigm of collision-based computing, which originates from conservative logic and the billiard-ball model by Fredkin and Toffoli [32] and logical computation by colliding gliders in Conway's game of life [33]. A collision-based computer employs mobile compact finite patterns and traveling localizations to represent quanta of information in nonlinear media. Information values, e.g., truth values of logical variables, are given by either absence or presence of the localizations or other parameters of the localizations. The localizations travel in space, and when collisions occur the result can be interpreted as computation. There are no predetermined stationary signal channels (wires)—however, stationary localizations, or reflectors, are allowed—and a trajectory of the traveling pattern is considered a transient wire. Almost any part of the medium's space can be used as a wire. Localizations can collide anywhere within the sample space; there are no fixed positions at which specific operations occur, nor location-specified gates with fixed operations. The localizations undergo transformations, form bound states, annihilate, or fuse when they interact with other mobile patterns. Information values of localizations are transformed as a result of collision [34].

In the design discussed, we use reflectors, which are stationary structures or defects. They are "artificial" in a sense that they do not belong to the liquid crystal medium but are externally and intentionally introduced to deflect propagating fingers. We assume the obstacles are rectangular. In real-world implementations, corners of the rectangular reflectors would generate additional propagating fingers; thus obstacles must be smooth, e.g., circular or oval. However, our model is coarse-grained and therefore any oval reflector would be composed of small rectangles and "perceived" as a group of rectangles by the fingers.

Basic gate $\bar{x}y$ is illustrated in Fig. 7. The gate consists of three obstacles o_1, o_2, and o_3 and two input fingers [Fig. 7(a)]. The fingers represent values of boolean variables x and y: presence of a finger x or y corresponds to $x = 1$ or $y = 1$ (TRUE), absence of a finger corresponds to 0 (FALSE). Obstacles o_1 and o_2 are used to deflect finger x, while obstacle o_3 is for deflecting finger x after collision with finger y.

Let us consider input $x = 0$ and $y = 1$ [Fig. 7(b)]. Finger x is not present. Finger y travels south undisturbed. There are no obstacles in its way, thus it continues along its original trajectory which represents output $\bar{x}y$. For input $x = 1$ and $y = 0$ only finger x enters the gate [Fig. 7(c)]. Finger x turns east after colliding with obstacle o_1. It then collides with obstacles o_3 and turns north. Both outputs are nil. When both inputs are TRUE both fingers x and y enter the gate [Fig. 7(d)]. Finger x enters the gate earlier than finger y. Thus by the time finger y passes southward along obstacle o_2, finger x moves northward. The fingers x and y collide with each other. As a result of this collision, finger y turns east and finger x turns west. The deflected trajectory of finger y propagating east represents output xy. Deflected finger x collides with obstacle o_2, is deflected south, and eventually becomes trapped and propagation ceases [Fig. 7(d)].

To implement a binary half-adder, one must realize calculation of a sum and carry on results: $x \oplus y$ and xy. An architecture of a LC finger half-adder can be implemented
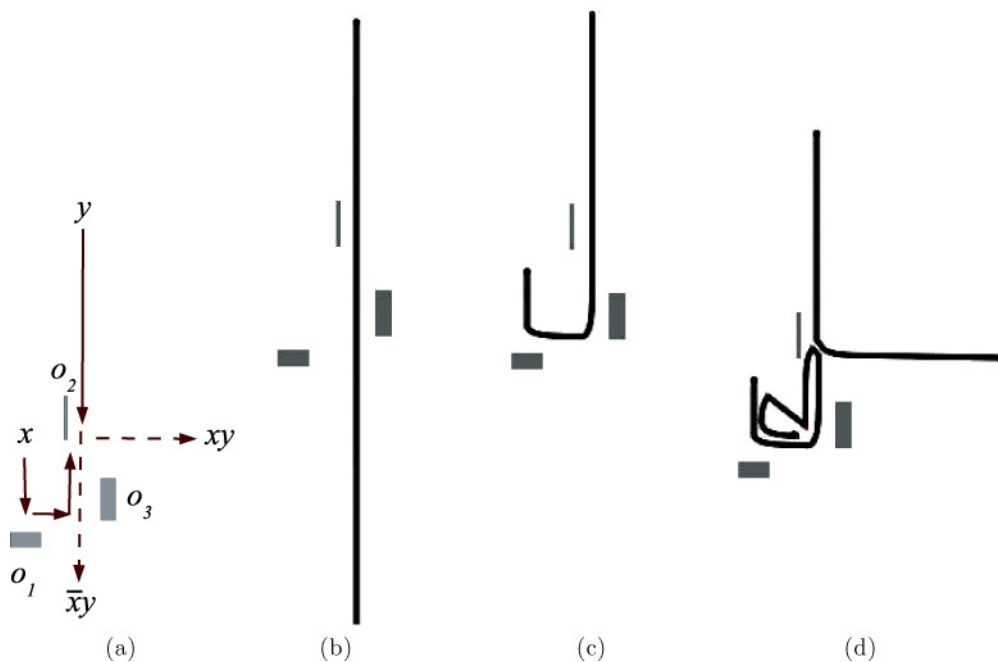

FIG. 7. (Color online) LC finger gate $\langle x,y \rangle \rightarrow \langle \bar{x}y, xy \rangle$. (a) Scheme: trajectories of fingers x and y entering the gate are shown by solid lines, trajectories of fingers after collision are shown by dotted lines, obstacles are shown by grey rectangles and tagged o_1, o_2, and o_3. (b)–(d) Configurations of two-dimensional mobile-automaton model. Fingers and obstacles are represented by states of cells. (b) Input $x = 0$, $y = 1$. (c) Input $x = 1$, $y = 0$. (d) Input $x = 1$, $y = 1$.

with three gates $\langle x,y \rangle \rightarrow \langle \overline{x}y, xy \rangle$ and five additional obstacles
(Fig. 8) (see also [35]). There are seven input trajectories
and two output trajectories. Inputs include three fingers
representing constant TRUTH, two fingers representing values
of x, and two fingers representing values of y. In the present
model we do not tackle multiplication or splitting of signals,
therefore we assume copies of variables x and y are presented
a priori. All input fingers travel south, output $x \oplus y$ travels
south, and output xy travels east. Note that all input fingers
enter the half-adder at different moments of time [Fig. 8(a)].
In total we have seven fingers labeled a to g and 14 obstacles
labeled o_1 to o_{14} [Fig. 8(b)].

Simulation of the one-bit half-adder in the mobile-
automaton model is illustrated in Fig. 9. When both inputs x
and y are FALSE only three fingers corresponding to constant
TRUE propagate southward [Fig. 9(a)]. Finger c collides, is
deflected by obstacle o_4, turns east, is deflected by obstacle
o_{11}, and turns north. It collides with finger f. As a result of
this collision, finger c turns north-west and is "self-trapped"
while finger f turns south-east. The body of finger f prevents
further propagation of finger g. Finger f and g are deflected
by obstacle o_{14} and propagate north. Thus for input $x = 0$ and
$y = 0$ no output fingers appear along dedicated trajectories
[Fig. 9(a)]: $\langle 0,0 \rangle \rightarrow \langle 0,0 \rangle$.

For the input combination $x = 0$ and $y = 1$, the situation
develops as follows [Fig. 9(b)]. Fingers c, f, and g, represent-
ing constant TRUE enter the the adder as usual. Fingers a and
e, representing x, are absent. Fingers b and d, representing
$y = 1$, enter the adder. Finger b collides with obstacle o_3 and
turns east, then collides with obstacle o_7 and heads north.
Thus fingers b and c come into a head-on collision. Finger b
turns west and is self-trapped. Finger c turns east, collides with
obstacle o_7 and then o_6, and gets trapped. Finger d collides with
obstacle o_8, heads east, collides with obstacle o_{12}, and travels
north. Fingers f and g continue their travel undisturbed. Thus
operation $\langle 0,1 \rangle \rightarrow \langle 1,0 \rangle$ is implemented.

When $x = 1$ and $y = 1$ [Fig. 9(c)], finger a (representing
first copy of x) is turned north by obstacles o_1 and o_5. Finger
c (representing second copy of x) is turned north by obstacles
o_{10} and o_{13}, and collides with finger f. Fingers c and f
stop propagating as a result of the collision. Finger c is
turned east by obstacle o_4 and then north by obstacle o_{11}.
Finger g continues its propagation undisturbed and thus
operation $\langle 1,0 \rangle \rightarrow \langle 1,0 \rangle$.

Most interactions between fingers take place in situation
$x = 1$ and $y = 1$ [Fig. 9(d)]. Finger a collides with finger
b. Finger b turns east as a result of the collision. The
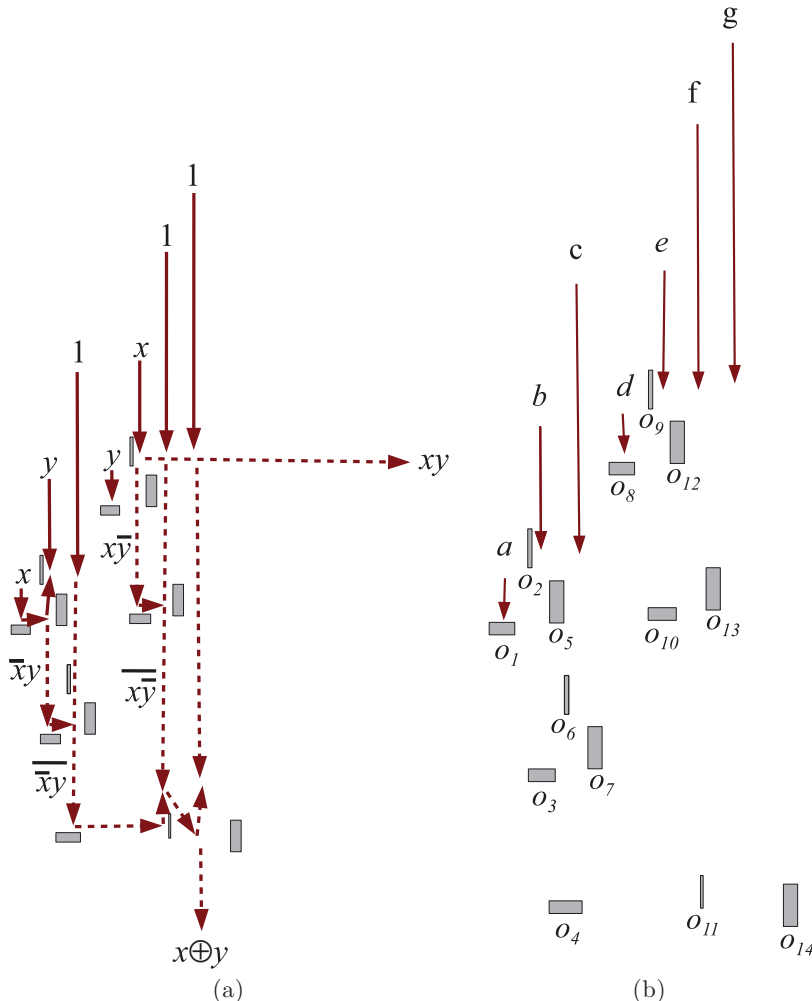body of finger b prevents finger c from propagating north.

FIG. 8. (Color online) Scheme of one-bit
half-adder implementable with (a) LC fingers
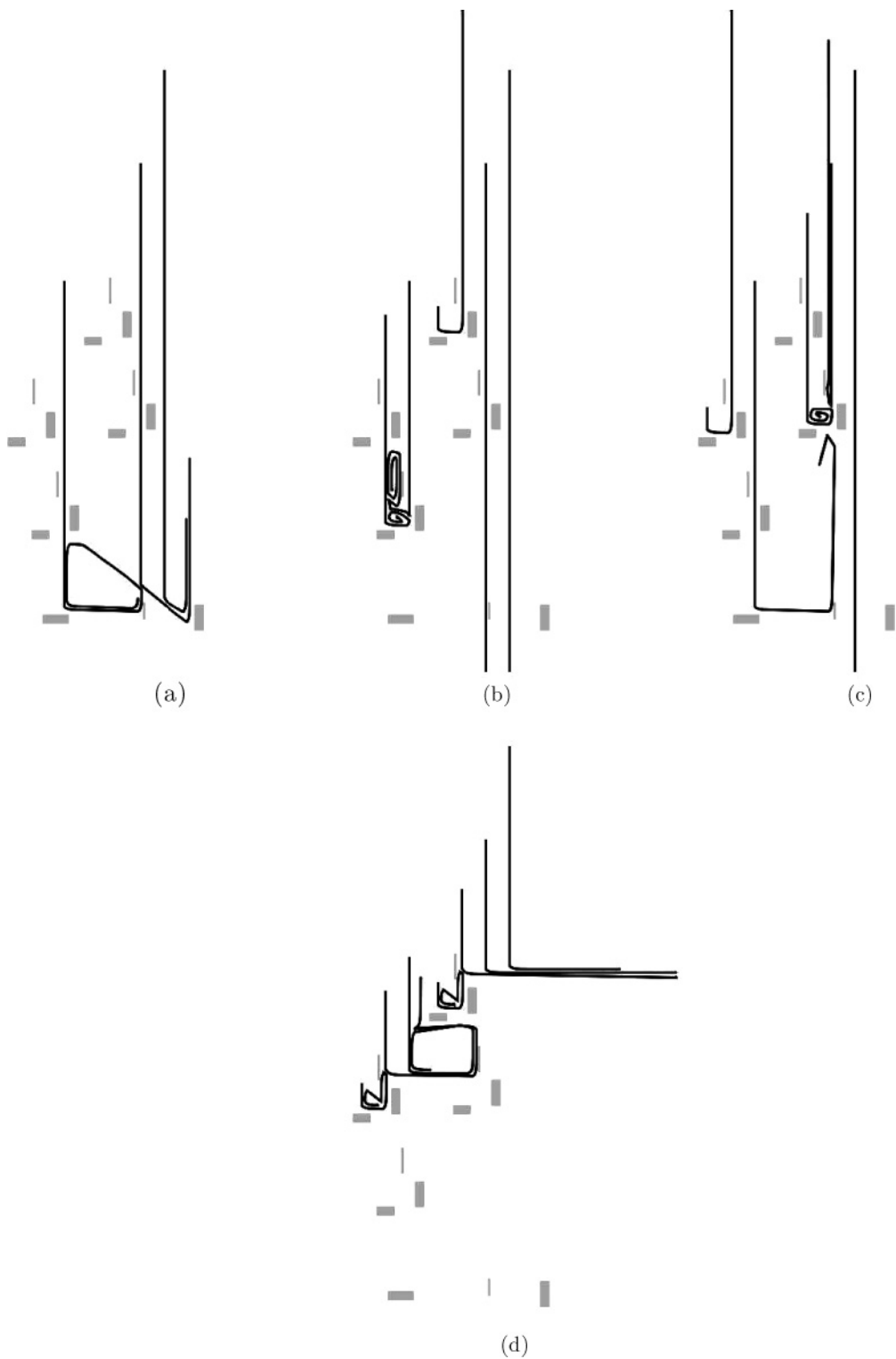and (b) labels of key elements of the adder.

FIG. 9. Configurations of two-dimensional mobile-automaton model. Fingers and obstacles are represented by states of cells. (a) $x = 0$, $y = 0$, (b) $x = 0$, $y = 1$, (c) $x = 1$, $y = 0$, and (d) $x = 1$, $y = 1$.

Finger d collides with finger e and becomes self-trapped. Finger e turns east as a result of the collision and exits the adder. Final trajectory of finger e represents $xy = 1$ [Figs. 8(a) and 9(d)]. Propagation finger g is blocked by the body of fingers e and f, it does not exit the adder along its original trajectory, and thus $x \oplus y = 0$.

VII. DISCUSSION

Cholesteric liquid crystals exhibit growth of localized phase defects—fingers—in response to the application of an ac electric field. The fingers show (almost) deterministic behavior when they collide with other fingers and obstacles,

thus they are suitable for implementation of collision-based computing schemes. For higher values of voltage applied, fingers show branching. Wave fronts of branching fingers stop their propagation when they collide with other fronts. Thus branching fingers can approximate a Voronoi diagram, a plane subdivision based on proximity criteria. For low applied voltages, the fingers remain solitary and thus each finger can represent a quantum of information and be an

elementary unit of a collision-based computing device. To illustrate the feasibility of the approach we provided the design of a one-bit binary half-adder and proved the correctness of its functioning using a mobile-automaton model. Collision-based computing prototypes presented in the paper are based on computer imitations of finger propagation, and further work is required to implement the design in physical laboratory conditions.

[1] P. G. De Gennes and J. Prost, *The Physics of Liquid Crystals* (Clarendon, Oxford, 1995).

[2] P. Oswald and P. Pieranski, *Nematic and Cholesteric Liquid Crystals: Concepts and Physical Properties Illustrated by Experiments* (CRC, Boca Raton, FL, 2005).

[3] I. I. Smalyukh, B. I. Senyuk, P. Palffy-Muhoray, O. D. Lavrentovich, H. Huang, E. C. Gartland, V. H. Bodnar, T. Kosa, and B. Taheri, Phys. Rev. E **72**, 061707 (2005).

[4] S. Pirkl and P. Oswald, Liq. Cryst. **28**, 299 (2001).

[5] P. Ribiere, P. Oswald, and S. Pirkl, J. Phys. II (France) **4**, 127 (1994).

[6] J. Baudry, S. Pirkl, and P. Oswald, Phys. Rev. E **59**, 5562 (1999).

[7] J. Baudry, S. Pirkl, and P. Oswald, Phys. Rev. E **57**, 3038 (1998).

[8] D. L. White and G. N. Taylor, J. Appl. Phys. **45**, 4718 (1974)

[9] J.-F. Li, X.-Y. Wang, E. Kangas, P. L. Taylor, C. Rosenblatt, Y. Suzuki, and P. E. Cladis, Phys. Rev. B **52**, R13075 (1995).

[10] X. Y. Wang, J.-F. Li, E. Gurarie, S. Fan, T. Kyu, M. E. Neubert, S. S. Keast, and C. Rosenblatt, Phys. Rev. Lett. **80**, 4478 (1998).

[11] M. Dennin, D. S. Cannell, and G. Ahlers, Phys. Rev. E **57**, 638 (1998).

[12] H. Riecke and G. D. Granzow, Phys. Rev. Lett. **81**, 0333 (1998).

[13] A. Adamatzky, in *Collision-Based Computing*, edited by A. Adamatzky (Springer, New York, 2003), p. 411.

[14] S. Kitson and A. Geisow, Appl. Phys. Lett. **80**, 3635 (2002).

[15] S. C. Kitson, E. G. Edwards, and A. D. Geisow, Appl. Phys. Lett. **92**, 073503 (2008).

[16] C. P. Lapointe, T. G. Mason, and I. I. Smalyukh, Science **326**, 1083 (2009).

[17] T. Nagaya, Y. Hikita, H. Orihara, and Y. Ishibashi, J. Phys. Soc. Jpn. **65**, 2713 (1996).

[18] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction* (Springer, New York, 1985).

[19] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tesselations* (Wiley, New York, 2000).

[20] F. Anton, *Proceedings of the Sixth IEEE International Symposium on Voronoi Diagrams, ISVD '09* (IEEE, New York, 2009).

[21] D. Tolmachiev and A. Adamatzky, Adv. Mater. Opt. Electron. **6**, 191 (1996).

[22] A. Adamatzky, *Computing in Non-linear Media and Automata Collectives* (IOP Publishing, Bristol, UK, 2001).

[23] A. Adamatzky, B. De Lacy Costello, and T. Asai, *Reaction-Diffusion Computers* (Elsevier, Amsterdam, 2005).

[24] B. P. J. De Lacy Costello, Int. J. Bifurcat. Chaos **13**, 1561 (2003).

[25] B. De Lacy Costello and A. Adamatzky, Int. J. Bifurcat. Chaos **13**, 521 (2003).

[26] B. P. J. De Lacy Costello, P. Hantz, and N. M. Ratcliffe, J. Chem. Phys. **120**, 2413 (2004).

[27] B. P. J. De Lacy Costello, A. Adamatzky, N. M. Ratcliffe, A. Zanin, H. G. Purwins, and A. Liehr, Int. J. Bifurcat. Chaos **14**, 2187 (2004).

[28] B. P. J. De Lacy Costello, I. Jahan, A. Adamatzky, and N. M. Ratcliffe, Int. J. Bifurcat. Chaos **19**, 619 (2009).

[29] A. Adamatzky, Phys. Lett. A **374**, 264 (2009).

[30] A. Adamatzky, *Physarum Machines* (World Scientific, Singapore, 2010).

[31] S. Fortune, in *Proceedings of the 2nd Annual Symposium on Computational Geometry* (ACM, New York, 1986), p. 313.

[32] E. Fredkin and T. Toffoli, Int. J. Theor. Phys. **21**, 219 (1982).

[33] E. R. Berlekamp, J. H. Conway, and R. L. Guy, *Winning Ways for Your Mathematical Plays*, Vol. 2 (Academic, New York, 1982).

[34] A. Adamatzky, in *Collision-Based Computing*, edited by A. Adamatzky (Springer, New York, 2003), p. 411.

[35] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.84.061702 for a video of a simulation of one-bit half-adder.