Geometric Computations by Broadcasting Automata on the Integer Grid

Russell Martin, Thomas Nickson, and Igor Potapov

Department of Computer Science,
University of Liverpool, Ashton Building,
Ashton St, Liverpool L69 3BX, U.K.,
{Russell.Martin,T.Nickson,Potapov}@liverpool.ac.uk

Abstract. In this paper we introduce and apply a novel approach for self-organization, partitioning and pattern formation on the non-oriented grid environment. The method is based on the generation of nodal patterns in the environment via sequences of discrete waves. The power of the primitives is illustrated by giving solutions to two geometric problems using the broadcast automata model arranged in an integer grid (a square lattice) formation. In particular we show linear time algorithms for: the problem of finding the centre of a digital disk starting from any point on the border of the disc and the problem of electing a set of automata that form the inscribed square of such a digital disk.

1 Introduction

In many cases it is deemed that large numbers of simple robots can achieve certain tasks with greater efficiency than a single complex robot. This leads to the question of how to co-ordinate these robots in their task whilst retaining their simplicity of design, function and the robustness that is inherent to distributed systems. Problems currently being worked on are common subdivisions of more complicated and pragmatic tasks which are found to tackle a large set of problems simply through the coordinated use of such sub problems. These are often, but not restricted to, pattern formation, aggregation, chain formation, self assembly, coordinated movement, hole avoidance, foraging etc.

The problem of swarm (or a set of mobile robots) configuration into regular grids is well known for exploration tasks and environmental or habitat monitoring [5]. However, arranging robots into a regular grid structure has a number of other benefits in terms of self organization and efficient communication. In this paper we illustrate the possibility of employing the property of regular structure for complex geometric constructions via non-oriented broadcasting in the static cluster of robots (broadcasting automata).

We first define a model of swarms of robots arranged on the grid. Each robot is represented by a broadcasting automaton with finite memory that is unable to observe its neighbourhood, but can communicate through the non-oriented broadcasting of messages with its neighbours. As a model broadcasting automata has widely been used for designing communication protocols [1, 4] and provides a realistic abstraction for network interaction.In the broadcasting automata model direct communication between automata is only possible via the broadcast of messages to all neighbouring automata within a certain communication range. In the case of the square lattice topology (as well as triangular and hexagonal lattices), non-oriented broadcasting can be used to efficiently solve a variety of geometric problems by utilizing the effects found in both real physical systems (i.e. waves and interference patterns) and computational systems (i.e. information processing by finite state automata).

The waves generated by activating processes in a digital environment can be used for designing a variety of wave algorithms. In fact, even very simple finite functions for the transformation and analysis of passing information provides more complex dynamics than classical wave effects. We generalize the notion of the standing wave which is a powerful tool for partitioning a cluster of robots on a non-oriented grid. In contrast to classical waves where interference patterns are generated by nodal lines (i.e. lines formed by points with constant values), an automata network can have more complex patterns which are generated by periodic sequences of states in time.

In this paper we introduce and apply a novel approach for self-organization, partitioning and pattern formation on the non-oriented grid environment based on the generation of nodal patterns in the environment via sequences of discrete waves. The power of the primitives are illustrated by giving solutions to two geometric problems: the problem of finding the centre of a digital disk starting from any point on the border of the disc and the problem of electing a set of automata that form the inscribed square of the same digital disk.

2 Broadcasting Automata Model on $\mathbb{Z}^2$

In general, a model of broadcasting automata is defined as a **network of finite automata** which is represented by a pair (G, Λ) where: G is a graph and Λ is a deterministic I/O automaton which is at each vertex, v, of the graph G.

The communication between automata is organized by message passing, where **messages** are from the alphabet Σ and are generated as the output symbols of the automata. The set of vertices connected to the automaton $a \in \Lambda$ at some vertex $v_a \in V$ is given by $\Gamma(a)$ and is the **set of neighbours** for that automaton, a. Messages generated by an automaton, $a \in \Lambda$, are passed to the automaton's **transmission neighbourhood**, which is a subset of its set of neighbours, $\Gamma_T(a) \subseteq \Gamma(a)$. Messages are generated and passed instantaneously at discrete time steps, resulting in synchronous steps. We will assume that if several messages are transmitted to a particular automaton a, it will receive only a set of unique messages, For any multiset of transmitting messages received by a in a single round, the information about the quantity of each type will be lost.

Let us consider two variants of broadcasting automata: **synchronous** and **asynchronous** (or reactive) models. In the case of the synchronous model, every automaton from the moment of activation, which occurs upon the receipt of a signal, follows discrete time steps and reacts on a set of received events (including an empty set, which corresponds to a no event situation). In the case of the asynchronous (reactive) model, every automaton follows discrete time steps but can only react to a non-empty set of events. Although such automata have no shared notion of time, a form of implied synchronicity may be derived from the constant amount of time taken for each round of communication. Both models are computationally equivalent, but the algorithms designed in each model may require different amount of resources such as the size of the messaging alphabet, number of broadcasts and the overall execution time.

Proposition 1. *Any algorithm for broadcasting automata in the reactive model with a message alphabet of size $|\Sigma|$ can be simulated by the synchronous model with a singular alphabet and $|\Sigma|$ slowdown. Any algorithm for broadcasting automata in the synchronous model with a message alphabet of size $|\Sigma|$ can be simulated by the reactive model with an alphabet $|\Sigma'|$ where $|\Sigma'| = |\Sigma| + 1$.*

Broadcasting automata on $\mathbb{Z}^2$. In this paper we consider a model of broadcasting automata on the non-oriented square lattice (integer grid). We note, however, a similar model can be defined on any other lattice or graph structure. On the square lattice it is possible to vary the transmission neighbourhood, Γ_T, by varying the transmission range. In particular, with a transmission radius equal to 1 or 1.5 (which covers those nodes at an exact distance of 1 and $\sqrt{2}$), the so-called **Von Neumann neighbourhood** and **Moore neighbourhoods** are generated, respectively.

In Figure 1 the source of the transmission is shown as the circle at the centre of the surrounding automata, those that are black are within the transmission range and thus in the transmission neighbourhood. The large outer circles represent the transmission range which can be changed to alter the automata that are included in the transmission neighbourhood. If the transmission radius is equal to 1, as in Figure 1 diagram $i)$ then only four of the eight automata can be reached. If the radius is made slightly larger and is equal to 1.5, it can encompass all eight automata in its neighbourhood. As we will show later, iterative broadcasting within Von Neumann and Moore neighbourhoods can distribute messages in the form of a diamond wave and a square wave * .

3 Computational Primitives for Broadcasting Automata

In this section we discuss a number of computational primitives that can be applied to non-oriented broadcasting automata on the grid in order to generate geometrical constructions. In the **synchronous** model waves are passed from automaton to automaton according to the following rules:

1) An automaton a receives a message from an activating source at a time t;

2) At a time $t + 1$, a sends a message to all automata within its transmission neighbourhood $\Gamma_T(a)$;

3) At a time $t + 2$, a ignores all incoming messages for this round.

In the **asynchronous model** simulation we require at least 3 symbols $\{u_0, u_1, u_2\}$:

1) An automaton a receives a message u_i from an activating source;

2) The automaton a broadcasts a message $u_{(i+1) \mod 3}$ to all automata within its transmission neighbourhood $\Gamma_T(a)$;

3) Ignore next incoming message $u_{(i+2) \mod 3}$.

Step 3 in each model prevents a node from receiving back the wave that it just passed

* It is also possible to show, in a more or less straightforward way, that broadcasting automata on $\mathbb{Z}^n$ (for any $n > 0$), with a single initial source of transmission, two radii of broadcasting (1 and 1.5) and a large alphabet of messages, can simulate a Turing Machine.

to its neighbours by ignoring all transmissions received the round after transmission and ensuring that the front of the wave is always carried away from the source of transmission. Waves are passed using two different transmission neighbourhoods referred to as square and diamond waves, which are equivalent to Moore and Von Neumann neighbourhoods respectively, see Figure 1. Whilst within the model an automaton is

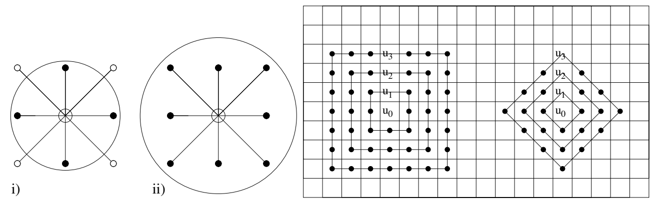

Fig. 1. Diagram i) represents the propagation pattern for a diamond wave (Von Neumann neighbourhood) and diagram ii) shows the propagation pattern for a square wave (Moore neighbourhood). Also wave propagation with the asynchronous model is shown on the square grid: Moore neighbourhood (left) and Von Neumann neighbourhood (right).

unable to directly access the state of its neighbours it is still possible to design many primitives in a similar style as designed for the cellular automata model [6], including synchronization procedures, finding the edge elements on a line (Lemma 1), shortest branch of a tree (Lemma 2), etc. These basic tools will be utilized to illustrate more complex algorithmic methods in Section 4.

Lemma 1. *Given a line of broadcast automata of size n and a single source A at a point on the line it is possible to elect the automaton which is on an edge of the line in $n+3$ time steps, where k is the number of automata from the initiating automaton to its closest side, and with an alphabet, Σ, of size $|\Sigma| = 3$, in the asynchronous model.*

Lemma 2. *It is possible to elect the end point of the shortest branch of a tree in time $3R$, where R is the length of the shortest branch and the alphabet required is $|\Sigma| = 2$.*

3.1 Nodal Patterns of Discrete Interference

In physics, a standing wave is a wave that remains in a constant position. This phenomenon can occur in a stationary medium as a result of interference between two waves travelling in opposite directions. A standing wave in a transmission line is a wave in which the distribution of current value is formed by the superposition of two waves propagating in opposite directions. The effect is a series of nodes (zero displacement) and anti-nodes (maximum displacement) at fixed points along the transmission line. With standing waves in a two dimensional environment the nodes become nodal lines, lines on the surface at which there is no movement, that separate regions vibrating with opposite phase [7].

In this paper we investigate standing waves in a discrete environment, where the original physical system is generalized in several ways. First, the transmitted waves will be discrete, i.e. the values of the wave which are passing a point p will correspond to a sequence of symbols that are observed in p. Assuming that a point in discrete environment has even simple computational power, like finite memory, the idea of nodal points and nodal lines can be extended. Apart from recognizing points with classical standing waves (i.e. having a constant value over time), it is possible to recognize periodic sequences of values over time. This will eventually lead to a richer computational environment where nodal lines of different periodic values can form patterns in $\mathbb{Z}^n$ and where points may react and pass a wave differently in contrast to a physical model.

In order to demonstrate the technique of nodal patterns for geometric computations in non-oriented $\mathbb{Z}^2$ we will start with a simpler abstraction in one dimension line. Let us place two transmitters, T_1 and T_2, on a one dimensional line, each broadcasting words u^*, v^* respectively, where $u, v \in \Sigma$, $|u| = |v| = s$ is the length of the word. The **broadcasting** of symbols begins at transmission points with symbols broadcast away from the source at each time step, cycling through all symbols in the word. We now have two infinite words $(u)^*$ and $(v^R)^*$, the reverse of v, that are shifted towards each other every next time step. In this case any point on the line, which contains two symbols

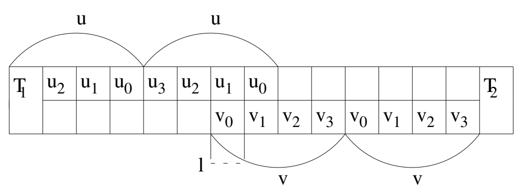

Fig. 2. Shows two transmitters T_1 and T_2 broadcasting words u and v respectively.

from u and v, forms a pair $p_{i,j} = (u_i, v_j)$. So once a point l contains two symbols, we can define a sequence of pairs $p_{(i+t) \mod s,(j+t) \mod s}$ contained in l over discrete time $t = 1, \ldots, \infty$. It is easy to see that such sequence will be periodic with a period less or equal to s, where $|u| = |v| = s$, and it represents a history of symbol pairs from u and v which are meeting at the point l over time.

The **nodal pattern** of a point l is a finite subsequence of s pairs $p_{(i+t) \mod s,(j+t) \mod s}$ over some $t = t'..t' + s$. Let us also assume that nodal patterns are equivalent up to a cyclic shift, so it will not be dependent to the initial time t'. Nodal patterns are labelled using the difference between the indices defined as $P_{|i-j|}$, where i and j are the indices from any two of the pairings of symbols $p_{i,j}$ for some point on the plane at some time. Nodal patterns have s possible labels, $P_0, P_1, \ldots, P_{s-1}$, one for each of the possible pairings. Such indices are now defined as the **node index** of the automaton, which signify the distinct pairing observed by the automata.

Nodal patterns are exemplified in Figure 2 where $s = 4$, $u = u_0, u_1, u_2, u_3$ and $v = v_0, v_1, v_2, v_3$. At point l the pairs $((u_1, v_0), (u_2, v_1), (u_3, v_2), (u_0, v_3))$ form a nodal point corresponding to $p_{(1+t) \mod s,(0+t) \mod s}$ over time t. The nodal pattern is now labelled $P_{|(1+t)-(0+t)|} = P_{|1+t-0-t)|} = P_{|1-0|} = P_1$.

Patterns p_1 and p_2 are called **distinct** if p_1 cannot be constructed from p_2 by applying any cyclic shift to it. Where two patterns are formed by ordered pairs and a word $u = v$ is without repetitive symbols we have s distinct nodal patterns. In the case of unordered pairs such that $(u_i, v_j) = (v_j, u_i)$ the number of distinct nodal patterns is smaller and defines a specific sequence described in the Propositions 2, 3.

Proposition 2. *Given a non-periodic word u, $|u| = s$. The number of possible distinct nodal patterns generated by broadcasting words u and v, where $v = u$, is $s/2 + 1$ if s is even and $(s + 1)/2$ is s is odd.*

Example 1. Given a sequence $u = v = 1, 0, -1, 0$ and so $|u| = |v| = s = 4$ it can be shown that there are $(s/2) + 1 = (4/2) + 1 = 3$ distinct nodal points. Enumerating all possible patterns (modulo removed for clarity) and where t is over time.

$$P_0 = ((1, 1), (0, 0), (-1, -1), (0, 0)); P_1 = ((1, 0), (0, -1), (-1, 0), (0, 1))$$
$$P_2 = ((1, -1), (0, 0), (-1, 1), (0, 0)); P_3 = ((1, 0), (0, 1), (-1, 0), (0, -1)).$$

Patterns P_1 and P_3 are the same assuming that the order of the symbols is not relevant.

Proposition 3. *In the sequence of s nodal patterns $P_0, P_1, \ldots, P_{s-1}$ derived from words $u = v$ where $|u| = |v| = s$, two nodal patterns P_k and $P_{s-k \mod s}$ are equivalent up to a cyclic shift.*

3.2 Nodal patterns with two sequential transmitters

Standing waves and nodal patterns may be formed when u^* ($|u| = s$) is transmitted from one source T_0 and is retransmitted from another source T_1 after reaching it. Any point of a grid having both signals from T_0 and T_1 will have identified its nodal pattern as the "difference" between them.

Upon moving the waves to two dimensions, $\mathbb{Z}^2$, it is important to define the distance between the points in terms of (square or diamond) wave propagation for proper calculation of the nodal patterns, as it is not the same as in Cartesian geometry.

Definition 1. *Square waves adhere to the distance function d_s for distances from x to y where $d_s : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}$ and is defined by the correspondence $d_s((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \max_{1 \le i \le n}\{|x_i - y_i|\}, (x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathbb{Z}^n$.*

Definition 2. *Diamond waves adhere to the distance function d_d for distances from x to y where $d_d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}$ and is defined by the correspondence $d_d((x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n)) = \sum_{i=1}^{n} |x_i - y_i|, (x_1, x_2, ..., x_n), (y_1, y_2, ..., y_n) \in \mathbb{Z}^n$.*

Informally speaking, the distance functions defined above provide the time taken for the diamond/square wave front to reach an automata at coordinate $(y_1, y_2, ..., y_n)$ from a transmitter at $(x_1, x_2, ..., x_n)$.

Let us consider two ways of computing nodal patterns for this process in terms of synchronous and asynchronous cases with transmitters T_0 and T_1. In both cases we aim to get the same distribution of nodal patterns on $\mathbb{Z}^2$ without the need to continuously propagate values from the sources. The local distribution of these patterns will allow us later to locate points in a non-oriented environment.

Definition 3. *An index i of a nodal pattern P_i in a point $\rho \in \mathbb{Z}^2$ is defined as the absolute difference (modulo s) between the shortest time when ρ is reached from T_0 and the sum of times of reaching T_1 from T_0 and ρ from T_1:*

$$i = |d(T_0, \rho) - (d(T_0, T_1) + d(T_1, \rho))| \mod s$$

where d is understood as d_s or d_d for square or diamond wave respectively.

Synchronous model with a single message. In case of transmitting a single message, waves here are activation waves, which the automata use to start internal clocks. Transmissions begin from T_0 where the activation wave propagates through the use of a square (or diamond) wave arriving at T_1 which is activated when reached by the first wave after a constant delay s. Any point that has received the first signal will start its internal clock (which counts modulo s), and then after receipt of the second signal the clock is stopped and the value of clocks corresponds to the index of the nodal pattern for this point which is $|(d(T_0, \rho) \mod s) - ((d(T_0, T_1) \mod s) + (d(T_1, \rho) \mod s))|$.

Asynchronous model with multiple messages. In the case of the asynchronous model, the same distribution of nodal patterns can be simulated by sending a wave from T_0 where, on the wave front, every point that receives a symbol u_i immediately transmits the symbol $u_{(i+1) \mod s}$. The pseudo-synchronization of wave propagation is achieved by assuming that every transmission takes the same constant time. Then transmitter T_1 operates in the same way once reached by u_i, transmitting the next symbol corresponding to u_{i+1} but using a different alphabet $\{v_1, ..., v_s\}$ to avoid problems whereby transmitting in the same alphabet could have a blocking effect on the wave. Each node should now contain a pair of symbols $(u_{i'}, v_{i''})$ which is enough to define the pattern $P_{|i'-i''|}$, where $i' = d(T_0, \rho) \mod s$ and $i'' = (d(T_0, T_1) + d(T_1, \rho)) \mod s$.

Theorem 1. *Let T_0 and T_1 be any two points in $\mathbb{Z}^2$ with coordinates (j_0, k_0) and (j_1, k_1), respectively. Assume that nodal patterns P_i were formed by square waves, i.e. $i = |d_s(T_0, \rho) - (d_s(T_0, T_1) + d_s(T_1, \rho))| \mod s$. For any point $\rho \in \mathbb{Z}^2$ the membership to one of the following sets $\{D_1, D_3, D_5, D_7\}$, $\{D_2, D_4, D_6, D_8\}$ or $\{D_9\}$ is uniquely identified by a number of distinct nodal patterns P_i in the Moore neighbourhood of ρ: $\{D_9\}$ — has two nodal patterns, $\{D_1, D_3, D_5, D_7\}$ — has one nodal pattern, $\{D_2, D_4, D_6, D_8\}$ — has three nodal patterns, see Figure 3:*

$D_1 = \{(x, y)|y \geq -x + (k_1 + j_1), y \geq x + (k_0 - j_0)\}$
$D_2 = \{(x, y)|y \geq -x + (k_1 + j_1), y \leq x + (k_0 - j_0), y \geq x + (k_1 - j_1)\}$
$D_3 = \{(x, y)|y \leq x + (k_2 - j_2), y \geq -x + (k_2 + j_2)\}$
$D_4 = \{(x, y)|y \leq -x + (k_1 + j_1), y \leq x + (k_0 - j_0), y \geq x + (k_0 + j_0)\}$
$D_5 = \{(x, y)|y \leq -x + (k_0 + j_0), y \leq x + (k_1 - j_1)\}$
$D_6 = \{(x, y)|y \leq x + (k_0 - j_0), y \leq -x + (k_0 + j_0), y \geq x + (k_1 - j_1)\}$
$D_7 = \{(x, y)|y \geq x + (k_0 - j_0), y \leq -x + (k_0 + j_0)\}$
$D_8 = \{(x, y)|y \geq x + (k_0 - j_0, y \leq -x + (k_1 + j_1), y \geq -x + (k_0 + j_0)\}$
$D_9 = \{(x, y)|y \leq x + (k_0 - j_0, y \geq -x + (k_0 + j_0), y \leq -x + (k_1 + j_1), y \geq x + (k_1 - j_1)\}$.

Proof (Sketch.). In a system with two transmitters, T_0 and T_1, nodal patterns are formed by broadcasting the square wave from T_0 which, once reached by the wave, will then be broadcast by T_1. The main observation is that broadcasting a square wave generates quadrants defined by the lines $x = y$ and $x = -y$, assuming that the transmitter is the origin, whereby within each quadrant the front of the wave expands such that each element of the orthogonal axis, to the one wholly contained by the quadrant (ie x,-x,y,-y), within the quadrant will contain the same member of the alphabet as all the others. The direction of the waves dictate the structure of the neighbourhoods for the automata. Then by computing an index i of a nodal pattern P_i at each point ρ: $i = |d_s(T_0, \rho) \mod s - (d_s(T_0, T_1) \mod s + d_s(T_1, \rho) \mod s)|$ it is easy to observe that any point

ρ in $D_{\{1,3,5,7\}}$ is surrounded by points with identical nodal pattern. Each area $D_{\{1,3,5,7\}}$ share common direction for both the waves and all waves share a common speed. So the difference between the first and second wave is constant. Hence, any region with arrows that move in the same direction will contain a single nodal pattern.

Those in areas D_9 are waves which are both heading towards each other. This causes the intersection of the waves and their alphabets to only occur in differences of two. If one set of symbols match and generate a nodal pattern (i.e. (u_i, v_j)) then it will not be the next set of symbols that match (i.e. $(u_i + 1, v_j + 1)$) but every other pair (i.e. $(u_i + 2, v_j + 2)$). This generates the pattern of every other nodal pattern being present.

Those in areas $D_{\{2,4,6,7\}}$ have waves that move in orthogonal directions. This leads to diagonal patterns of the same node. If one were to track the intersections of the lines $x = x'$ and $y = y'$ as y' and x' increase, $0 \leq x', y' \leq n$, the result would be a diagonal line of the form $x = y$. The very same happens but with a multitude of the lines represented by the fronts of the wave.

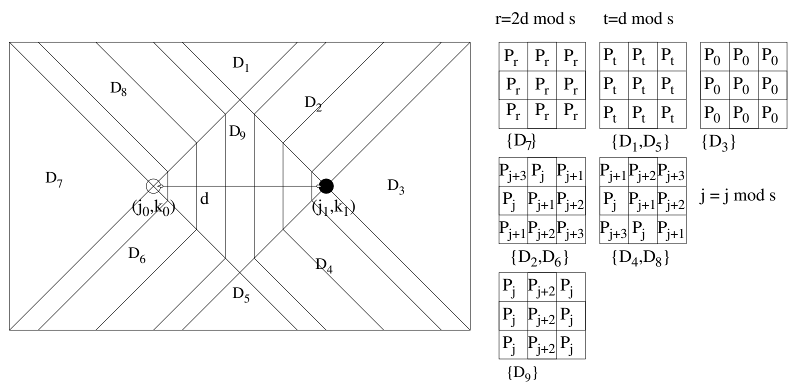

Fig. 3. The distribution of three distinct nodal patterns in case of two “transmitter", where $s = 4$.

Partial Vision of the Neighbourhood. In the current model each element is only aware of its own node index, and information about its neighbours is not directly observable. However, it is possible to organize a partial vision which provides access to a set of distinct values of nodes from the Moore and Von Neumann neighbourhoods via non-oriented broadcast. In order to get this set of distinct values an automaton should send a special message (nodal request) to all neighbourhoods which, in turn, will send their nodal indices on the next step. Alternatively it is possible to initiate an iterative process of such requests through the whole network of broadcasting automata on $\mathbb{Z}^2$ where each next automata will react on this information by transmitting its own nodal value. This type of transmission we call as a **node wave**.

It is possible to test membership to one of the partitions (defined in Theorem 1) generated through waves broadcast by two transmitters. Having partitioned the plane with two transmitters, T_0 and T_1, T_1 waits a constant delay s before broadcasting a nodal wave which allows all automata information about its neighbourhood. After these steps the automaton has enough information to decide to which of the partitions it belongs.

Apart from the above partitioning it is possible to elect vertical, horizontal (set L_0) and diagonal lines (set L_1) starting from a single source by transmitting square and diamond waves after a constant delay.

Proposition 4. *Given a vertex, $T_0 = (j, k)$ on $\mathbb{Z}^2$, it is possible to elect a set of points $L_0 = \{(x, y)|y = k, x \in \mathbb{Z}\} \cup \{(x, y)|x = j, y \in \mathbb{Z}\}$ or the set $L_1 = \{(x, y)|y = x + k - j \in \mathbb{Z}\} \cup \{(x, y)|x = -y + k + j \in \mathbb{Z}\}$ in linear time.*

4 Geometric Problems on the Digital Disc

A set ζ of points in $\mathbb{Z}^2$ is a **digital disk** if there exists a Euclidean circle, with a centre at an integral point, that encloses the pixels of ζ but excludes its complement. Let us consider a model of broadcasting automata on a digital disk which has a diameter D. We define a procedure for finding the centre of the digital disk in linear time using the notion of waves as described in previous sections. It is possible to find the centre of the digital disk as a single point or as a set of two points, depending on whether the radius of the digital disk is odd or even, respectively. In this section we abbreviate the partitions previously mentioned to $a = \{D_9\}$, $b = \{D_1, D_3, D_5, D_7\}$ and $c = \{D_2, D_4, D_6, D_8\}$. The algorithm for finding the centre can begin from any arbitrary point, T_1, on the edge of the digital disc and is implementable in both asynchronous and synchronous models. Depending on the location of the initial point, one of three algorithms is applicable. Finding the correct algorithm to apply is reduced to checking the initial point's neighbourhood to one of three possible sets in the following way.

Definition 4. *Eight points $\{0, 1, ..., 7\}$ on the circumference of a digital circle, ζ corresponds to the following eight angles 0,45,90,135,180,225,270,315.*

Lemma 3. *Given an automaton on the edge of ζ it is possible to check the automaton's membership to one of three sets: $\{0, 2, 4, 6\}$, $\{1, 3, 5, 7\}$ and all other points on the edge of the circle in a time $O(D)$ for both models.*

Proof. From automaton T_1, on the edge of ζ, Proposition 4 is applied via the transmission of a square then diamond wave, resulting in horizontal, vertical lines of elected automata and electing at most two paths. All elected automata on the edge of the disc, ζ, become new points T_N which is a set of points encompassing up to three automata, $\{T_2, T_3, T_4\} \in T_N$. As soon as the automaton or automata on the edge of the disc, ζ, have been elected by T_1, the automata, now denoted as T_N, begin transmission of a square wave. As the transmission of these waves from all automata in T_N, may occur simultaneously on the disc, points at which waves meet each other proceed no further on the disc, due to the automata's inability to receive and broadcast at the same time, cancelling each other. Points of wave cancellation are shown as dotted lines in Figure 4. The partitions formed by the transmissions from new points in T_N can now be detected by the initial point T_1 through the transmission of a neighbourhood detection wave which gives nodal patterns to automata through the transmission of its own square wave and causing neighbouring automata to transmit their states which allows the detection of T_1's neighbouring nodal patterns. By Theorem 1, possible neighbourhood partitions for the initial point T_1 are now be categorised as $\{a\}$,$\{a, c\}$ and $\{a, b, c\}$ which are the points $\{0, 2, 4, 6\}$ and $\{1, 3, 5, 7\}$ and all other points respectively. The procedure requires only three waves of transmissions, each wave require the time that is no more then the diameter of the circle as well as some constant time between transmissions and the constant time for the neighbourhood recognition.

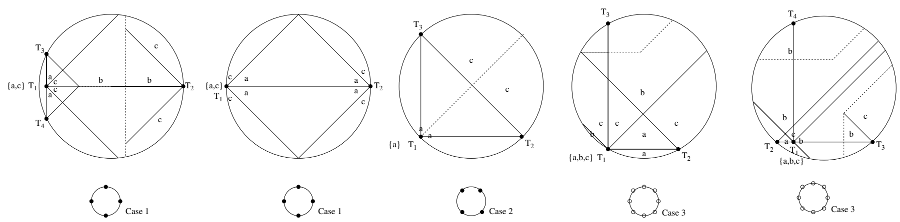

Fig. 4. The above Figure shows the three possible cases stemming from the five possible variants that require differing solutions based on their location. The two differing sets of the 8 points and those points which lay in none of these. The two diagrams for Case 1 correspond to the situations whereby, for case 1, T_1 generates a 3-branched tree with two equidistant branches or a single chord respectively. Whereas for case 3 the situations are those such that T_1 generates a 2- or 3-branched tree respectively.

An Algorithm for Locating the centre of a Digital Disk
1) An automaton on the edge of the disc ζ, T_1, checks its location by the creation of the unique local neighbourhood sets: $\{a\}$, $\{a, c\}$ or $\{a, b, c\}$, see Lemma 3.
2) In case of neighbourhood set $\{a, c\}$ apply the algorithm for case 1. In case of neighbourhood set $\{a\}$ apply algorithm case 2.
3) In the case of neighbourhood set $\{a, b, c\}$, T_1 is the root of a tree with two or three branches. The third branch may appear if the automata, T_1, finds itself on a 'ledge', such that there are automata on three sides of its Von Neumann neighbourhood, formed from the digitization of the circle. The end point of the shortest branch of such tree, placing the automata in a position whereby there are only two automata in its Von Neumann neighbourhood, is found by Lemma 2 which is relabelled T_1 and then apply case 3.

Cases 1 and 2 are basic because the location of the point T_1 is known exactly. The least trivial case is Case 3 where further partitioning is required for locating the centre.
Algorithm for Case 1 (set $\{0, 2, 4, 6\}$)
1) T_1 sends message m_0 to T_2 through the chord which will be sent back from T_2 after some constant delay $k = |\Sigma|$.
2) T_1 sends message m_1 which has a delay of 3 after a constant delay $k = |\Sigma|$.
3) The automata on the chord elected through receipt of both message m_0 and m_1 at the same time will be the centre of the digital disc ζ.
Algorithm for Case 2 (set $\{1, 3, 5, 7\}$)
1) A new point T_4 is generated along the diagonal through the use of diamond neighbourhood detection wave as described in Proposition 4.
2) T_1 sends message m_0 to T_2 through the chord which will be sent back from T_2 after some constant delay $k = |\Sigma|$.
3) T_1 sends message m_1 which has a delay of 3 after a constant delay $k = |\Sigma|$.
4) The automata on the chord elected, through receipt of both message m_0 and m_1 at the same time, will be the centre of the digital disc ζ.

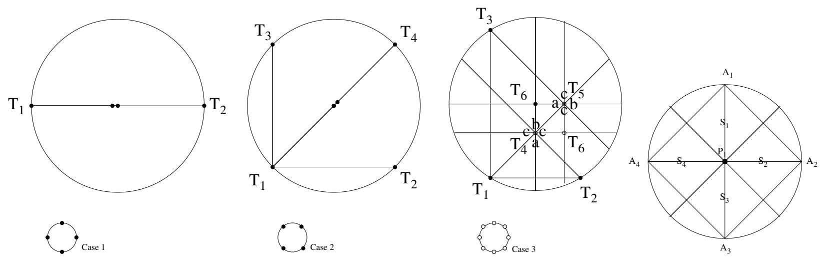

Fig. 5. Constructions required to find the centre of the circle for the three differing cases. Centres are indicated by black dots. Figure on the right corresponds to production of the inscribed square from centre point.

Algorithm for Case 3
1) Point T_2 is identified as the shortest chord of the tree constructed from T_1 via Lemma 2.
2) Point T_3 is identified as the longest chord of the tree constructed from T_1 by broadcasting a signal c only to those automata that have received two a's but no b's after Lemma 2, the longest chord.
3) Transmissions from T_3 followed by T_1 elect the point T_5, the only point on the digital disc that has a neighbourhood containing the nodal patterns $\{a, b, c\}$.
4) Transmissions from T_2 followed by T_1 elect the point T_4, the only point on the digital disc that has a neighbourhood containing the nodal patterns $\{a, b, c\}$.
5) Application of proposition 4 from T_4 and T_5 will generate horizontal and vertical lines. The points at which these two line cross are elected as T_6. Only one of the two points marked T_6 will be in partition a given by the initial construction of T_1 and T_3, this is the centre point of the circle.

Theorem 2. *It is possible to find the centre of a digital disc ζ with diameter D starting from a point on the circumference of ζ, T_0, in both models, in $O(D)$ time* ** .

Theorem 3 directly follows from the construction of the next algorithm and also holds when starting from a point is on the circumference of ζ.

Algorithm for electing elements of the inscribed square:
1) From point P transmit two waves (square and diamond) to elect the four points A_1, A_2, A_3 and A_4 (see Figure 5).
2) Transmit a square wave from points A_1, A_2, A_3 and A_4. The interference pattern from these waves will form four squares, shown as S_1, S_2, S_3, S_4 which will be of the distinct type D_9 according to Theorem 1.
3) A node wave must now be sent from P which informs the automata of their neighbours and allows the automaton to place itself in the set D_9 if it is within the inscribed square. Note that the complement of the square will also have a distinct pattern of type $\{D_2, D_4, D_6, D_8\}$.

** As there is a finite number of passing of waves which all time-bounded by at most $O(D)$, the algorithm cannot exceed linear growth by D.

Theorem 3. *Given an initial transmitting node in the centre of a digital disc. It is possible to elect automata forming an inscribed square in time $O(D)$, where D is the diameter of the digital disk.*

5 Conclusion and Discussion

We have shown that non-oriented broadcasting of messages on the square grid can form stable interference patterns. Such patterns can be used for efficient partitioning, self-location problems and geometric computations on the static cluster (of robots or automata) via transmission of discrete square and diamond waves, where shapes are defined by the radius of broadcasting and the topology of the grid structures. These shapes can be much more complex then square and diamond waves when we can choose larger radii, higher dimensions and other structures of grid topology. For example, broadcasting with a radius three on the square lattice resulting in the octagon shape of the wave which also, in its turn, provides more complex partitioning of the lattice. Based on our ongoing experimental and theoretical work with lager radii we have observed that they can form quite complex shapes and patterns for square, triangular and hexagon grids as well for high dimensional structures.

Moreover, the sequence of the string that is transmitted by a wave can also be increased in complexity (i.e. different periodic or non-periodic strings, numerical sequences, etc.) along with the corresponding aggregation function. For example, in the case of natural wave intersection the aggregation function is simply the addition of amplitudes which we only extended in this paper through access to a finite history.

The proposed algorithms based on digital waves and nodal patterns can also be extended by using iterative application of informational waves on already generated nodal pattern. So in each subsequent round waves can be transmitted with different delays, speed and shapes based on already formed pattern. Currently we are exploring many other generalization and constraints of the proposed approach which will be presented in future publications by the authors.

The current paper extends the existing area of wave algorithms by introducing new methods, framework and models for further theoretical analysis and practical implementation of complex self-orientation and self-organization mechanisms.

References

1. Alderighi, Sechi, Mazzei, Tisato: Broadcast Automata: a Parallel Scalable Architecture for Prototypal Embedded Processors for Space Applications. HICSS (5) 1997: 208-217.
2. Jurek Czyzowicz, Leszek Gąsieniec, Andrzej Pelc *Gathering Few Fat Robots in the Plane* Principles of Distributed Systems, LNCS, Volume 4305/2006, 350-364 pp, 2006.
3. David Peleg, Asaf Efrima, *Distributed algorithms for partitioning a swarm of autonomous mobile robots*. Theoretical Computer Science,Vol. 410 , Issue 14, 1355-1368 pp, 2009.
4. Martijn Hendriks: Model Checking the Time to Reach Agreement. FORMATS 2005: 98-111.
5. Lee, Geunho and Chong, Nak: A geometric approach to deploying robot swarms. Annals of Mathematics and Artificial Intelligence 2008: 257-280.
6. J. Kari, Theory of cellular automata: a survey, Theor. Comput. Sci. 334 (2005)
7. Standing Waves http://en.wikipedia.org/wiki/Standing_wave, January 2010.