Chapter 10
Convex Hulls and Metric Gabriel Graphs

Luidnel Maignan and Frédéric Gruau

Abstract. The convex hulls construction is mostly known from the point of view of 2D Euclidean geometry where it associates to a given set of points called seeds, the smallest convex polygon containing these seeds. For the cellular automata case, different adaptations of the definition and associated constructions have been proposed to fit with the discreteness of the cellular spaces. We review some of these propositions and show the link with the famous majority and voting rules. We then unify all these definitions in a unique framework using metric spaces and provide a general solution to the problem. This will lead us to an understanding of the convex hull construction as a chase for shortest paths. This emphases the importance of Voronoï diagrams and its related proximity graphs: Delaunay and Gabriel graphs. Indeed, the central problem to be solved is that of connecting arbitrary sets of seeds, in a local and finite-state way, while remaining inside the desired convex hull, i.e by shortest paths. This is exactly what will be made possible by a suitable generalization of Gabriel graphs from Euclidean to arbitrary metric spaces and the study of its construction by cellular automata. The general solution therefore consists of two levels: a connecting level using the metric Gabriel graphs and a level completing the convex hull locally as the majority rule does. Both levels can be generalized to compute the convex hull, when the seeds are moving.

10.1 Notational and Naming Convention

In this chapter, the set of all cells of the space is denoted S. The neighborhood of a cell x is denoted $N(x)$ and does not include the cell x itself. A set of cells

Luidnel Maignan
LACL, Université Paris-Est Créteil, France
e-mail: luidnel.maignan@u-pec.fr

Frédéric Gruau
LRI, Université Paris-Sud Orsay, France
e-mail: gruau@lri.fr

P. Rosin et al. (eds.), *Cellular Automata in Image Processing and Geometry*,
Emergence, Complexity and Computation 10,
DOI: 10.1007/978-3-319-06431-4_10, © Springer International Publishing Switzerland 2014

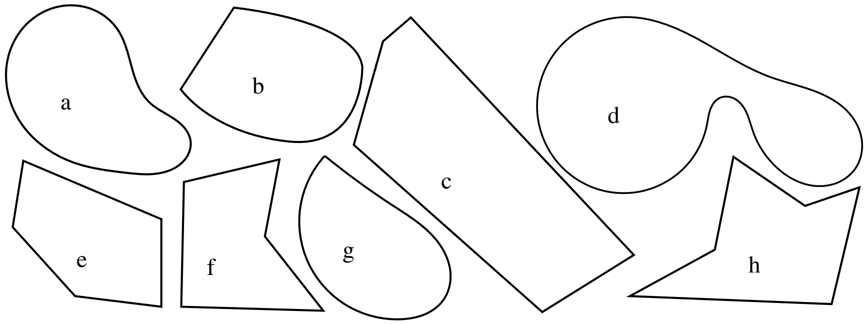


Fig. 10.1 Examples of Euclidean convex and non Euclidean convex shapes

together with a specified neighborhood is called the cellular space. Cells are sometimes called points. The value associated by a configuration c at a cell x is denoted $c(x)$. The configuration obtained at time t by a rule f is denoted f_t, subscripts being used to reduce the impact of repeating the timing information. Hence, $f_t(x)$ is the value associated to a cell x at a time t by a rule f and the transition function of the rule, for example the sum of all the values in the neighborhood, can be denoted as $f_{t+1}(x) = \sum_{y \in N(x)} f_t(y)$. Note that we do not indicate explicitly what is the initial configuration, letting it free to be initialized externally in some sense. Also, rules are often called fields, similarly to electric field and magnetic field. A single cellular automaton can be made of a certain number of fields, in which case they can refers to each other in their transition function, and even to just computed values. For example if there are two rules f and g, $f_{t+1}(x)$ might depend on $g_{t+1}(x)$ as long as there are no circular definition and everything is well-defined. This is why we need to be explicit about the timing information. This is a light notation related to, although a bit more explicit than, the one used in multivariate differential calculus for example.

10.2 Introduction

The convex hull construction is well known in the context of Euclidean geometry. Its definition comes in two steps, one for the word “convex” and one for the added word “hull”. A set of points is **convex** if and only if it contains entirely any segment joining any two of its points. As an exercise to link this definition with its geometrical content, the reader should check, for each shape of Fig. 10.1, whether the set of its inside points is convex or not by trying to exhibit a segment not entirely contained in the shape while its extremities are in the shape. Note that some of the convex shapes are polygons, and others are not.

Convex shapes are simple shapes in some interesting sense. In physics, or physics simulations for example, detecting when two arbitrary shapes collide is difficult but this detection is much easier when the shapes are convex. There are also situations where all the shapes that appear are necessarily convex, such as in a bumble bath, etc. However, there are situations where the shapes are not convex, but we would like to approximate it by a convex shape in order to make life easier. In this case, we have an initial set of points, not necessarily convex, and we look for its **convex hull**, i.e. the smallest convex set containing all the initial points. In the rest of this chapter, these initial points will be called **seeds**. The question is how to compute the convex hull of a set of seeds given as input.

A well known fact is that if there are finitely many seeds, their convex hull is a polygon using these seeds as vertices and many classical algorithms exist to compute this polygon as a list of vertices based on a set of seeds, everything being described using a coordinate system. In contrast, this chapter considers the case where the space is not the usual Euclidean space but a cellular space and the seeds are not represented by coordinates but by a Boolean information holded by each cell and representing whether the cell is a seed or not. So we do not assume any knowledge of the classical algorithms but let us add a few words on why finite sets of seeds have a polygonal convex hull in 2D Euclidean spaces.

The reason is the following: for the set of seeds to be convex, all the segments joining them have to be in the set, so we consider now a set containing the seeds and their pairwise segments. This new set in still not convex in general and we still need to add all the segments joining the newly added points, and so on until no segment is missing. The final result of this iterative adding process is indeed the minimal convex set containing the seeds since only absolutely required segments have been added. But all these additional points are strictly between the firstly added segments, so the border of the convex hull is made of some of the firstly added segment with the seeds as vertices. This is a very basic process, but it worths taking the time to visualize it and to keep it in mind from the start.

In the context of cellular automata, the famous majority and voting rules and some of their variations exhibit behaviors related to Euclidean convex hulls as described in [7]. For instance, let us consider the majority rule where cells have two states, either selected or not selected, and the rule selects a cell if at least 4 or its neighbors are selected. If we denote by P the set of seeds and by the predicate $\mathrm{majo}_t(x)$ the fact that the cell x is selected at time t by the rule majo, this behavior can be written formally as:

$$\mathrm{majo}_{t+1}(x) = x \in P \vee \mathrm{card}\{\, y \in N(x) \mid \mathrm{majo}_t(y) \,\} \geq 4 \tag{10.1}$$

Figure 10.2 shows an evolution of this rule with a dense set of seeds as initial configuration. Here, the Moore neighborhood is used, so each cell has 8 neighbors, and the number 4 is 50% of the neighborhood, hence the name of the rule. So the number would be 2 for a Von Neumann neighborhood, 3 for hexagonal cells which have 6 neighbors, and so on. In all these cases, it is clear that the final result is visually related to Euclidean convex hulls. To make this precise, different approaches have been tried. The goal is not only to understand what sort of convex hull is computed by the majority rule, but also to be able to compute such a convex hull for arbitrary sets of seeds, which is unfortunately not the case for the majority rule.

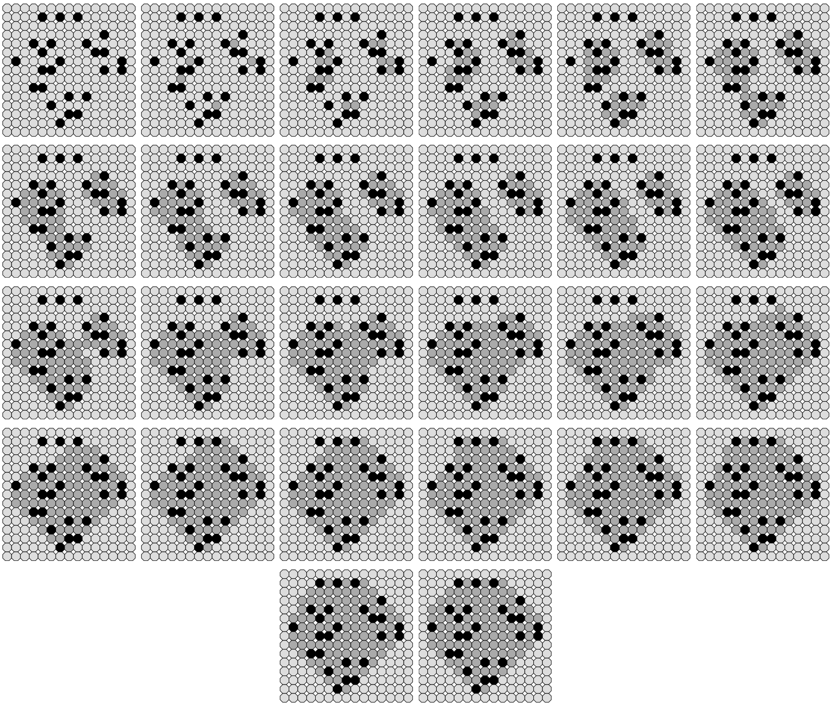
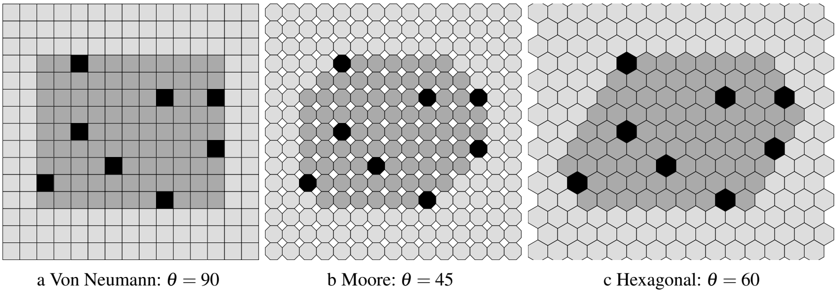


Fig. 10.2 Convergence to one global convex hull

We first describe the early angular point of view to this problem, describing the obtained convex hull in terms of Euclidean convexity with a restriction on the allowed angles in Sect. 10.3. This point of view takes into account the Euclidean positions of the cells. In Sect. 10.4 and Sect. 10.5, we then describe a more general approach describing everything in terms of shortest paths, i.e. in terms of distances, i.e. in terms of metric space. In this case, the focus is on something more intrinsic than the cells position, namely the graph of communication of the cells. We then conclude in Sect. 10.6 by a summary the complete cellular automaton that works for any set of seeds and for any cellular space.

10.3 The Angular Point of View

The first approach is to consider the positions of the cells in Euclidean space. If we take an arbitrary cell and consider the vector leading to each of its 8 neighbors, we can see that they match the directions of the sides of the convex polygon obtained in Fig. 10.2. This idea can be extended to Von Neumann neighborhood with 4 vectors, and to hexagonal cellular spaces with 6 vectors. The trick is therefore to change the definition of convexity and to allow a set to be called convex if and only if it is Euclidean convex polygon and its sides follow one the admitted


a Von Neumann: $\theta = 90$

b Moore: $\theta = 45$

c Hexagonal: $\theta = 60$

Fig. 10.3 θ-convex hulls in cellular spaces

vector directions. The Von Neumann, Moore, and hexagonal cellular spaces thus allow directions whose angles are multiple of 90°, 45°and 60°respectively. This concept is called **θ-convexity**, and examples of θ-convex hull are shown in Fig. 10.3. It might worth noting that, contrary to the Euclidean convex hull, the vertices of the θ-convex hull polygon are not necessarily in the original set of points. Using this definition, let us examine again the behavior the majority rule and see how arbitrary sets of seeds can be handle.

10.3.1 Majority Rule and θ-Convexity

With the definition of θ-convexity, we have an explanation of the results of the evolution depicted in Fig. 10.2. However this behavior is not the general case. When the set of points is not dense enough, we might obtain a disconnected set, corresponding to a collection of partial θ-convex hulls. In fact, it is not really a matter of density, and if we take the initial configuration of Fig. 10.2 and move only one of the six bottom seeds, we obtain the behavior shown in Fig. 10.4. In Fig. 10.2, the moved seed is higher, allowing the partial convex hulls to merge into a bigger convex-hull. So in general, the complete θ-convex hull is not obtained, and it is hard to describe the final result by an other way than saying that this is the fixpoint of the majority rule. But both of these issues will be addressed, firstly with the solution described in the next section, and then further by changing the point of view from a angular to a metrical one.

10.3.2 Complete θ-Convex Hull

In [5, 14], a cellular automaton is proposed that constructs the θ-convex hull of an arbitrary set of seeds, with the little price that the seeds have to be already enclosed in an arbitrary initial connected region. This rule improves this initial region by two stages that eventually transforms it into the θ-convex hull of the seeds. The first stage is to reduce the region to ensure that it is smaller than the wanted θ-convex hull. The second stage is to inflate the region just enough to exactly match

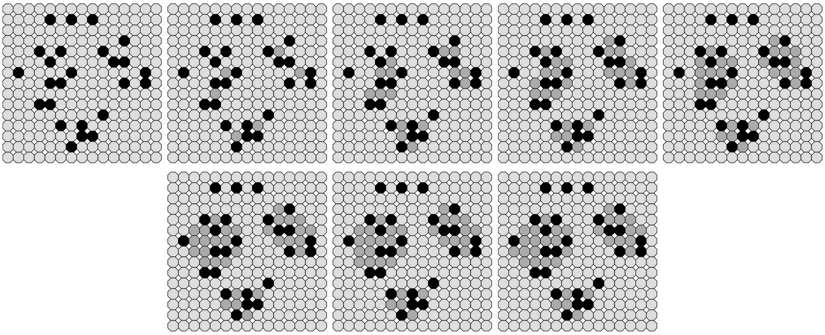
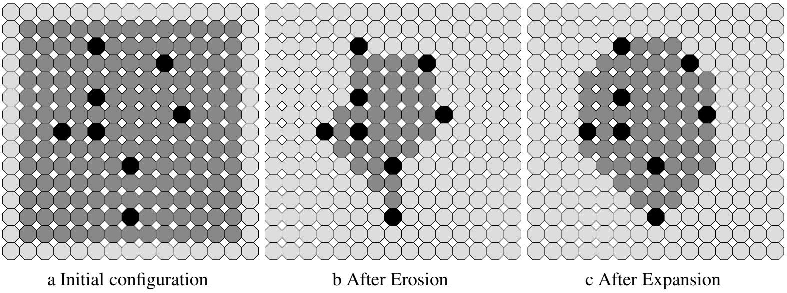


Fig. 10.4 Convergence to many convex hulls

the wanted result. This second stage is simply a modified version of the majority rule described earlier, which is enough because the result of the first stage is a connected region. However, the transition from the first to the second stage requires a mechanism that ensure a coherent global transition to the second stage, as it is not desirable to be in the shrinking and growing stages at the same time. There are of course some little technical difficulties, and the reader is redirected to the original articles for more details. An example of initial configuration is shown in Fig. 10.5, along with the result of the first and second stages on it.

10.4 The Metrical Point of View

The solution described in the previous section is a great improvement on the majority rule since it handles an arbitrary set of seeds. However, one might ask whether it is possible to get rid of the initial region and simply build the region from the seeds. In the case of the Euclidean convex hull, we noted in Sect. 10.2, the definition of the convex hull asks to add all required segments to the initial set of seeds in order



a Initial configuration b After Erosion c After Expansion

Fig. 10.5 Stages from wrapped seeds to their convex hull: The initial wrapping (a) is shrunk into (b) and is then grown to convex hull (c)

to obtain a convex set. This idea is the main anchor of the rest of the chapter, and it is applied with a change of point of view.

In this new point of view, we forget about any position of the cells in some Euclidean space and simply consider the graph of communication of the cells and its structure. In this context, we can not talk about angles, but we can still talk about distances between pairs of cells. Indeed, Euclidean convexity is a particular case of metric convexity. This latter is defined over metric spaces, and replaces the word *segment* in the definition of Euclidean convexity by the more general concept of *shortest path*. Therefore, a set of points is **metric convex** if and only if it contains entirely any *shortest path* joining any two of its points. As everything else in this chapter will depend on this definition, let us be more precise by giving the formal definition of metric spaces, shortest paths and metric convexity.

A **metric space** is a set of points M together with a distance function d, also called metric function or metric, associating to each pair of points $p_0, p_1 \in M$ a real value $d(p_0, p_1) \in \mathbb{R}$ in a certain appropriate way that allows us to think of $d(p_0, p_1)$ as the distance between the points. The requirement to be appropriate is that the function must be symmetric, i.e. $d(p_0, p_1) = d(p_1, p_0)$, the function should assign the value 0 for all and only for distances between a point and itself, i.e. $d(p_0, p_1) = 0 \Leftrightarrow p_0 = p_1$, and whenever we make a detour via a third point, the total distance should be greater or equal than the direct travel, i.e. $d(p_0, p_1) + d(p_1, p_2) \geq d(p_0, p_2)$. This last requirement is the least trivial one. It is often called the **triangle inequality** and is very important.

Given a path in a metric space, one can consider the length of this path by summing the distance between each pair of successive points of the path. Between two arbitrary points p_0 and p_1, one may ask for the length of the shortest paths among all possible linking paths and expect this value to match $d(p_0, p_1)$. However, this does not follow from the three previous requirements, but it is often convenient to have this property. In this case, the metric of the space is said to be intrinsic, and the space is called a **length metric space**. For example, Euclidean spaces and also graphs with their natural notion of distance are both length metric spaces. In this case, the triangle inequality really says something about the length of the shortest paths, detours and direct travels.

In particular, it is possible to express the fact that a point z is on a shortest path between two points x and y as a **triangle equality**: $d(x,z) + d(z,y) = d(x,y)$, which means that we do not make the path lengthier by making a detour via z. We say that z is **between** x and y, or more explicitly that z is on a shortest path joining x and y. We will use this triangle equality and, equivalently, the notion of betweenness many times, and will denote it as $z \in [x,y]$, which means that $[x,y] = \{ z \in M \mid d(x,z) + d(z,y) = d(x,y) \}$. This is called the **interval** between x and y. In the Euclidean case, this set $[x,y]$ is exactly the points of the segment joining x and y, but in our cellular spaces where there might be many shortest paths between two cells, it corresponds to the union of all these shortest paths.

After all these definitions, let us see what a metric convex hull looks like in our cellular spaces. Firstly, in Fig. 10.6, the interval between two points of various cellular spaces are shown. The figure also shows the "Moore-$\sqrt{2}$" cellular spaces which

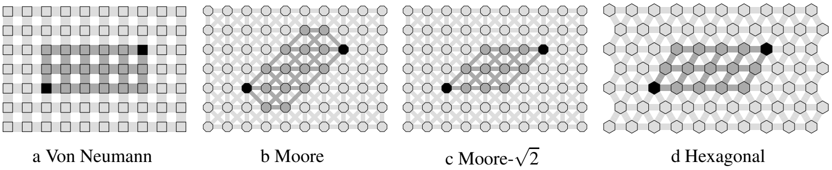


Fig. 10.6 Intervals for different cellular spaces and different metrics

correspond to the communication graph of the Moore cellular space with the length of diagonal assigned the value $\sqrt{2}$ instead of just 1 as graph metric would imply. This difference makes the horizontal and vertical edges shorter, and therefore more preferable than diagonal one. This discards some paths of the Moore cellular space as being shortest. The structure of the intervals dictates the structure of the metric convex hulls. Examples of metric convex hulls are shown in Fig. 10.7. This shows that the definition of metric convex hull is strictly more general than θ-convex hull since the 90, 45, and 60-convex hull corresponds to the metric convexity in Von Neumann, Moore-$\sqrt{2}$ and hexagonal cellular spaces respectively. Note that the Moore-$\sqrt{2}$ can be constructed as the intersection of the Von Neumann and Moore metric convex hull. This allows us to restrict our attention only to Von Neumann, Moore and Hexagonal cellular spaces, i.e. we will always consider just plain graph metric, i.e. the distance between a cell and one of its neighbor is always 1, even for Moore diagonals. This sets precisely what our goal is: to understand the majority rule based on metric convexity and to find a cellular automaton that computes this metric convex hull for arbitrary sets of seeds. Because of the chosen level of generality, the results will also be applicable for a large class of cellular spaces, including 3D and higher dimensional cellular spaces, but those are unfortunately not very good as paper examples.

10.4.1 Majority Rule and Metric Convexity

Coming back to what we already said, computing the convex hull is about adding all the missing shortest paths. In fact, the majority rule can be understood in these terms. To see that, let us consider the neighborhood $N(x) \cup \{x\}$ of the center cell x and the distance function restricted to this neighborhood. If we take a such neighborhood

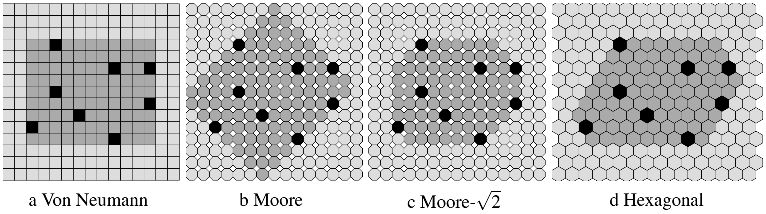


Fig. 10.7 Convex hulls for different cellular spaces and different metrics

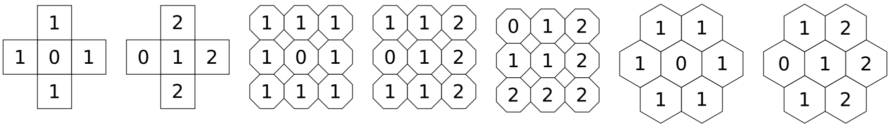


Fig. 10.8 Distances between cells in a neighborhood

and take one cell y in it, we can assign to all cells z its distance $d(y,z)$ to y as depicted in Fig. 10.8. If y is the center x, then all of its neighbors are at distance 1 by definition, as shown in the first, third, and sixth neighborhoods of Fig. 10.8. If y is a neighbor of the center x, then the direct neighbors of y are at distance 1 from y and any other cell z is at distance 2. Indeed, in this last case, the path (y,x,z) is a shortest path of length 2. Now, what is the relation with the majority rule ? The answer is that any time the majority rule selects a cell, it is because half of its neighborhood is already selected as specified in Sect. 10.2. But this implies that there are two selected cells of distance 2 apart from each other. This means that there is a shortest path joining these two cells and passing through the center as we just pointed out. Therefore the center has to be selected for the set of selected cells to have a chance of being metric convex. This explains why the majority rule never produces more than the convex hull, and evolves toward convexity.

Using this understanding, we can replace the majority rule with a more direct one that checks for the existence of two selected cells in the neighborhood such that a shortest path passes through the center:

$$\mathrm{conv}_{t+1}(x) = x \in P \vee \exists y_0, y_1 \in N(x); \mathrm{conv}_t(y_0) \wedge \mathrm{conv}_t(y_1) \wedge x \in [y_0, y_1]; \quad (10.2)$$

Note that there are many ways to write this rule formally, but this one is the more direct one with respect to our intention. Therefore, this formula remains unchanged if we change the neighborhood in size or shape for example, while other ways of writing it might need some adaptation.

With this rule, we obtain for example the evolution depicted in Fig. 10.9. The conv rule has roughly the same global behavior than the majority rule presented in the sense that it generally produces a set of partial convex hulls and that the partial convex hulls can merge to form bigger convex hulls during the evolution. However it has at least two benefits. Firstly, the conv rule is more precise than majo. Indeed, it detects strictly more neighborhoods than the majority rule. As an example, one can note that the initial configuration of Fig. 10.9 is a fixpoint for the majority rule. Secondly, the final result has a clear property that can be expressed without reference to fixpoints: any shortest path of length 2 between any two cells of the selected region belongs to the selected region. The link with the convex hull is far more obvious in this way.

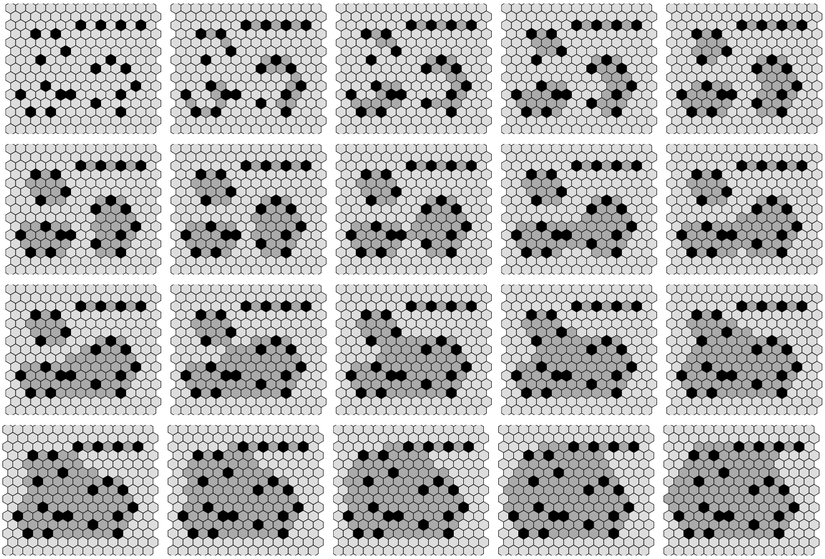


Fig. 10.9 Evolution of the conv rule with neighborhood radius 1

10.4.2 Complete Metric Convex Hulls

Now that we know how to obtain partial metric convex hulls, involving only shortest paths of length at most 2, let us target the detection of shortest paths of arbitrary length. Our first goal is, given two seeds, to select the shortest paths joining them, i.e. all cells of their interval. A first idea might be to compute the distance of all cells to all the seeds and then select the cells where the triangle equality is verified. However we can not handle an arbitrarily large number of integers in each cell in the cellular automata framework. The number of states of the cells has to be finite.

We can reduce this idea and only store in each cell one distance, the distance to the closest seeds, modulo 3. This information is, in fact, enough to detect the middles of the shortest path, and from them, the entire shortest path. A rough intuition for the latter is given by the fact that, along any shortest path connecting two seeds, the distance modulo 3 values stored in the cells is $(0,1,2,0,1,2,0,2,1,0,2,1,0)$ for instance. It is possible to locally distinguish between the middle and the non middle cells, and from these middles, it is possible to join the seeds by following the distance values in decreasing order $(2,1,0)$. This is explained in more details in Sect. 10.4.2.1.

In Sect. 10.4.2.2, we explain that this interval detection is, in fact, enough to solve entirely the problem and we examine why this is the case. In fact, when many seeds are involved, sufficiently many pairs of seeds are connected to allow the conv or majo rules to complete the job. But the investigation of the detection of middles in general cellular spaces of arbitrary dimensions leads to a trip through the notions of

Voronoï diagram, Delaunay and Gabriel graphs in order to make the full connection between the Euclidean and the cellular case, and is therefore postponed to Sect. 10.5.

10.4.2.1 Paths Joining Two Distant Seeds

As just explained, to connect two distant seeds, we need to compute some distance information. The cellular automata framework only allows for a finite number of state for the cells, so we can not store sets of integers, or even single arbitrarily large integers. We are reduced to compute at each cell only one distance value, and to compute it modulo 3: this distance value is the distance of the cell to its closest seeds. We could have called this the distance rule, but we in fact call this the **distance field**.

In the cellular automata case, this distance field corresponds to the computation of the minimal distance modulo k, where k depends on properties of the complete cellular automaton where the distance field is used. It is a very general notion that can be used to solve many problems and can handle the motion of seeds and asynchronicity to some extent for example. Its full description [10] is out of the scope of this book chapter so we present only a reasonable restricted version of it to the case of synchronous static seeds. In this case the value of k is 3.

In this restricted version, the idea is to begin with the value 0 on the seeds, and the value 1 anywhere else as shown in the initial configuration of Fig. 10.10. From this initial configuration, each cell increases its value at each transition until its value becomes greater than one of its neighbor value[1]. This happens exactly when one of the neighbors has a different value since all values only increases then stops at some transition. Formally, the transition function is:

$$\mathrm{dist}_{t+1}(x) = \begin{cases} 0 & \text{if } x \in P \\ \mathrm{dist}_t(x) + 1 \bmod 3 & \text{if } x \notin P \wedge \forall y \in N(x);\ \mathrm{dist}_t(y) = \mathrm{dist}_t(x) \\ \mathrm{dist}_t(x) & \text{if } x \notin P \wedge \exists y \in N(x);\ \mathrm{dist}_t(y) \neq \mathrm{dist}_t(x) \end{cases} \quad (10.3)$$

The meaning of this transition rule can be observed on Fig. 10.10. After the first transition, all values are at 2, except the seeds, which kept their value 0, and the cells around the seeds which kept their value 1 because of the presence of 0 in their neighborhood. After the second transition, all cells values go to 0, except the seeds (because they are seeds), the neighbors of the seeds (because of the presence of the 0 in their neighborhood while their value is 1) and the neighbors of the neighbors of the seeds (because of the presence of 1 in their neighborhood while their value is 2). So circles of cells of same distance value are constructed, and after some transitions, the circles coming from different seeds eventually collide at different cells. The first

[1] Note that because of the modulo three operation, one should only look at the differences between values locally, i.e. in the neighborhood of a given cell. If this cell value is 1, then the order is $0 < 1 < 2$, but if this cell value is 0 or 2, the orders are respectively $2 < 0 < 1$ and $1 < 2 < 0$. So a cell of value v considers, in its local neighborhood, that the value $v - 1 \bmod 3$ is less than its value and $v + 1 \bmod 3$ is greater than its value. This is the way the comparisons have to be understood.

cells where these collisions occur are those of interest: they are exactly at the middle of the shortest paths between the seeds. In Fig. 10.10, this happens after the forth transition. We can see that the distance values in the neighborhood of these middles is special. Each of the middle cells can actually see that the distance pattern in its neighborhood can only appear because of the presence of two seeds at opposite directions.

In Fig. 10.11 theses middle cells are marked in light green. We need to express the general middle detection rule more precisely of course. For a particular grid, it is possible to build this rule by enumerating all local distance patterns corresponding to middles, but expressing this detection rule for general cellular spaces appears to be the main difficulty in the convex hull construction. So let us pretend that we can do it and proceed to the final step. We will come back to the middle detection in the next section.

So if we suppose that we can detect these middle cells, the rest of the evolution is obtained easily and is depicted in Fig. 10.11. Once the middles are detected, we can consider the neighbors of the middles. Let us consider one of them and call it x. It has all required information to know if it belongs to the shortest path or not. Indeed, it knows its distance value $d(s,x) = \text{dist}_t(x)$ to the nearest seed s, the distance value $d(s,y) = \text{dist}_t(y)$ of the marked middle y, and the distance $d(x,y) = 1$. It can thus determine whether it is between the seed and the middle, i.e. whether it is on a shortest path joining s and y, i.e. $x \in [s,y]$. The triangle equality $d(s,y) = d(s,x) + d(x,y)$ to be checked is reduced to $\text{dist}_t(y) = \text{dist}_t(x) + 1$ in this case. So if y is marked as being in the convex hull and this reduced triangle equality is verified, x can deduce that it has to belong to the convex hull too. Formally, we obtain the following transition rule:

$$\text{back}_{t+1}(x) = \text{cent}_{t+1}(x) \vee \exists y \in N(x);\ \text{dist}_t(y) = \text{dist}_t(x) + 1 \bmod 3 \wedge \text{back}_t(y). \tag{10.4}$$

A more simple explanation of the behavior is that to go back to the seeds from the middles, you have to follow the distance value in decreasing order, i.e. the cell x takes the mark from the cell y if $\text{dist}_t(x) = \text{dist}_t(y) - 1$. Both views are simply equivalent, they both describe a trajectory along the shortest possible paths back to the seeds. It is interesting to note that anywhere the notion of shortest path is used, it can somehow be reduced to the triangle equality. Coming back to Fig. 10.11, we can see after 4 transition the detection of the middles of distance value 1, and then, at the next configuration some cells of distance value 0 deduce that they have to be in the convex hull from the presence of a marked cell in their neighborhood and the differences in their distance value, and after that, the same thing happens to some cells of distance value 2 at the next configuration, and so on. It worths checking the distance values on Fig. 10.10 while consulting Fig. 10.11.

10.4.2.2 Pairwise Hull Construction

We now have a way to construct the convex hull of two seeds using one distance field and some detections on it. When considering many seeds, we can obviously not

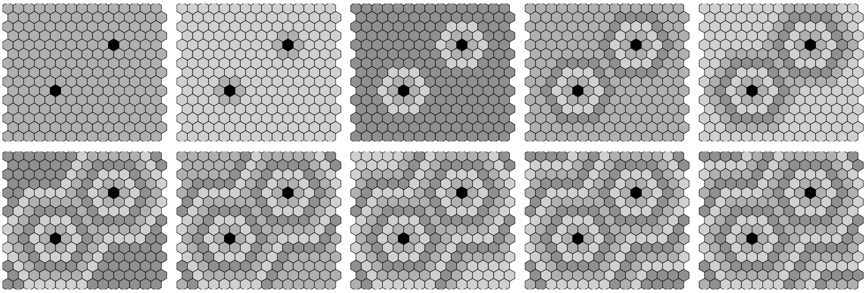


Fig. 10.10 Distance fields from two particles

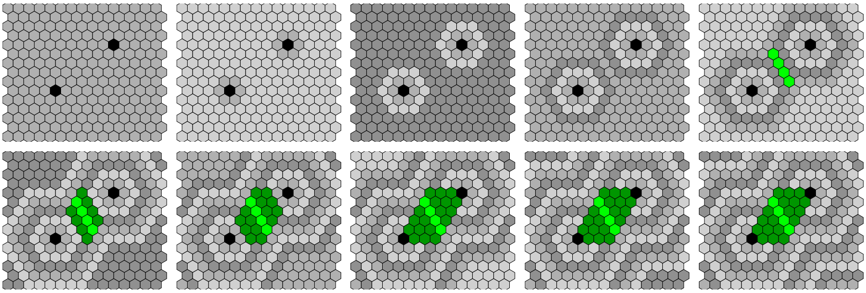


Fig. 10.11 Detected middles and paths back to both particles

build one distance field for each pair of seeds in the space, as we need the number of states of the cells to be finite. However, a single distance field might allow for the detection of middles of more than one pairs of seeds. This can be checked on Fig. 10.12 which shows the evolution of the rules dist and back (using our yet-to-be-defined detection of the middles cent) in the presence of many seeds. In the case of this figure, we can even see that the result is connected. If this is always true that the result of this pairwise construction is connected, then adding conv or majo

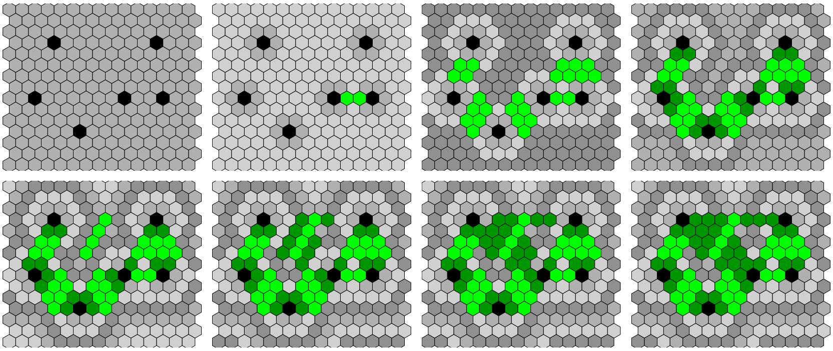


Fig. 10.12 Construction of pairwise convex hull

rule to our previous construction is enough to complete to the complete convex hull. It turns out to be the case from the results of the next section, and we can already say that the final construction of the convex hull is therefore obtain using the dist, cent, back, and conv. Although we used different colors to mark middles, and non middle cells, they can all be marked in the same way is our static case. We therefore use 7 states: i.e. one seed state, and for non-seed cells, there is a binary mark and a modulo 3 value, so 2 * 3 = 6 other states. The binary mark is set to "true" either by the middle detection, the back detection or the conv detection. This is a first possible summary of our metric solution. But understanding why this is indeed a solution in general is the same as understanding how the middles should be detected, which is the subject of the next section. A more detailed summary is given in Sect. 10.6.

10.5 Detectable Middles and Metric Gabriel Graphs

10.5.1 Pairwise Construction in Euclidean Space

In order to understand the detection of the middles, let us look at the pairwise construction described earlier in the context of Euclidean geometry. Figure 10.13 shows very precisely what the result would be in this case, and has to be compared with Fig. 10.12. The distance field would be continuous, and concentric circle would grow similarly as seen in the cellular automata case. When the circles collide, some middles showed in light green are detected, and the corresponding paths from these middles back to the seeds can also be detected. The analogy allows us to use existing tools build from Euclidean spaces to re-expressed the behavior of this construction: we can use now the vocabulary of Voronoï regions and diagrams and two of its associated proximity graphs: Delaunay and Gabriel graphs.

10.5.1.1 Voronoï Diagram and Distance Fields

The Voronoï diagram is a famous construction [1, 3]. It consists of associating to each point of the space its closest seeds. After doing so, each seed has an associated set of points, and this is called the **Voronoï region** of this seed. Some points have many seeds at equal distance. If we highlight these points, the result is called the **Voronoï diagram** and displays the borders of the Voronoï regions, the points where the Voronoï regions are in contact.

Using the distance field implicitly means to have some relation with the Voronoï regions and diagrams since the distance field associates to each cell of the space the distance to the closest seeds. Cells that have more than one closest seeds are those having special patterns of distance value in their neighborhood. This relation is shown in Fig. 10.14a, where the Voronoï regions boundaries, a.k.a. the Voronoï diagram, is displayed in blue. Comparing this with the final configuration of Fig. 10.13 shows the relation with the circles of distance field.

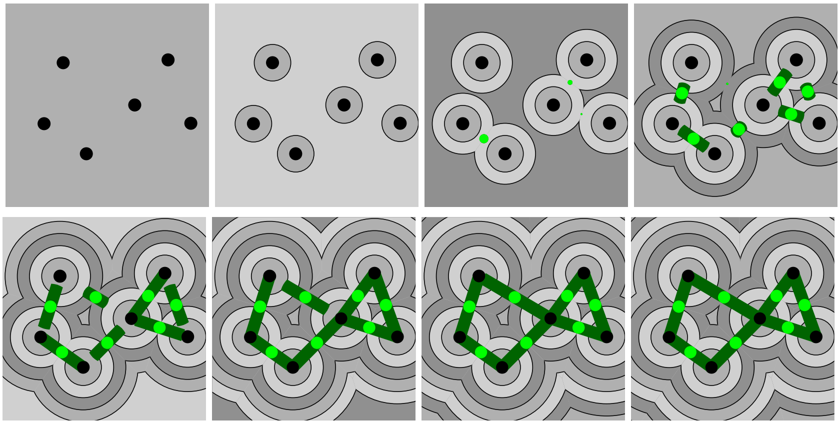


Fig. 10.13 Construction of pairwise convex hull in the Euclidean context

10.5.1.2 Delaunay Graphs as a Tentative

When using Voronoï diagrams, an associated construction is to consider adjacent Voronoï regions. It is formalized in the notion of **Delaunay graph** which has for vertices the seeds, and for edges the pairs for seeds whose Voronoï regions are adjacent. For two regions two be adjacent, there must be at least one point having for closest seeds the seeds of these two regions. This means that the edges of the Delaunay graph can also be described as the pairs of seeds such that there exists a disk containing these two seeds, but having no other seed inside it. This other description has the benefits not to refer explicitly to the Voronoï diagram. Whatever description we use, Delaunay graph edges are strongly related to the pairs of seeds for which we detect the middles in Fig. 10.14b. But if we compare with Fig. 10.13, we can see that there are too many edges.

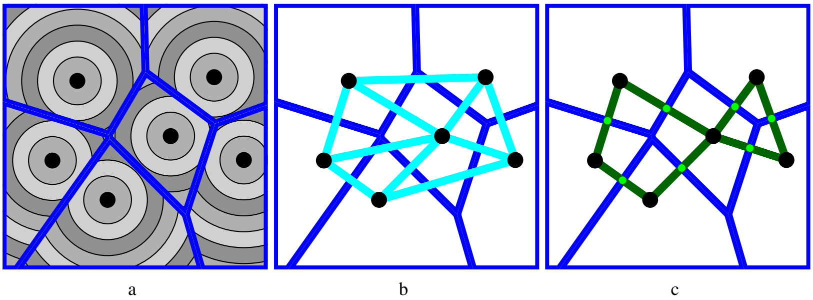


Fig. 10.14 Euclidean distance fields, Voronoï diagram, Delaunay and Gabriel graphs

10.5.1.3 Gabriel Graphs and Middle Detection

In fact, since we detect the edges from their middles and theses middles are detected from special distances patterns on the Voronoï diagram. This means that any edge whose middle is not on the Voronoï diagram is not constructed. Removing those non-detectable edges from Fig. 10.14b reduces it to Fig. 10.14c, and it comes out that this resulting graph is also of great interest in applications relying on graphs in Euclidean spaces: it is the **Gabriel graph** of our seeds. While edges of the Delaunay graphs connect seeds of adjacent Voronoï region, the Gabriel graph retains only edges that do not pass through the Voronoï region of another seed. This can be put in another way which is the most known definition of this construction: there is an edge between two seeds x and y if and only if there is a disc using the segment $[xy]$ as diameter and this disc does not contain any other seed. Given two seeds, such a disc is called the **Gabriel disc** of these seeds and its center is called the **Gabriel center** between these seeds. These Gabriel centers are exactly the middles that are detected in Fig. 10.13, which makes Gabriel graphs very important for our purpose.

Gabriel graphs are proximity graphs introduced in [6] to study sets of geographical points. They are now used in many domains such as wireless [8] and sensors networks for routing and communication management purpose. They also serve as tools to study proximity of points in order to cluster them, in domains like data mining, data and multivariate analysis [2], and machine learning [4, 12]. In computer graphics [9, 13], they can help to convert a set of points into a 3D surfaces and to obtain information about such surfaces. Their wide spread use is due to their numerous properties. Indeed, they are related to Voronoï diagrams, Delaunay graphs as just seen, but also to planar graphs, minimum spanning trees, nearest neighbor graphs, and represent or contain optimal solutions for some classes of energy-minimizing problem [8].

One particularly important property of Gabriel graphs is that *they are always connected*. This is exactly the property what we need to ensure that the construction of the convex hull is always complete, but for the cellular case. So our goal in the next section is to see how this Euclidean construction fits to cellular spaces. In fact, some adaptation will be required. So we will describe a new generalization of Gabriel graphs that we call *metric Gabriel graphs*. The latter carries properties of Gabriel graphs in any metric space, which allows them to be used in most considered cellular spaces in particular. This will provide us with the rule for the detection of middle that we need in general for many cellular spaces.

10.5.2 (Metric) Gabriel Graphs in Cellular Spaces

10.5.2.1 Failure of the Original Definition

In this section, we show that the original Gabriel graph definition does not accommodate the particularities of the cellular space. We highlight the fact that the original definition relies on many uniqueness properties of the Euclidean space, the uniqueness of the segment linking two points for example. Cellular spaces do not

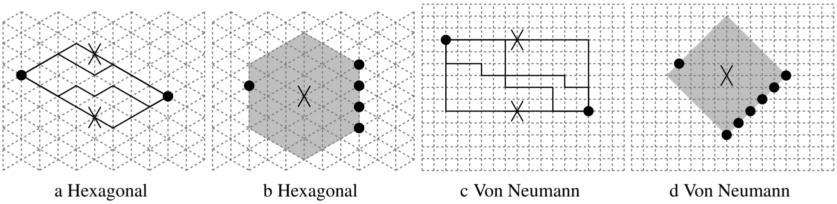


Fig. 10.15 Non-uniquenesses in hexagonal and Von Neumann communication graphs: (a,c) lines indicate possible shortest paths, and crosses indicate many possible centers for the balls, (b,d) the isolated point forms a diameter for the ball with any of the other points

have these uniqueness properties, so a generalization of the original definition is needed.

Indeed, the connectedness of the original definition mainly relies, by definition, on the existence of sufficiently many Gabriel discs and associated centers. In an Euclidean space, for a given arbitrary disc and a given point on the border of this disc, there is a unique diameter segment. Also, for any pair of points, there is a single disc using these points as diameter. Unfortunately, considering cellular spaces, and replacing the notion of disc by the metric notion of balls does not conserve these uniqueness properties as shown in Fig. 10.15. A **ball** is given by a center point and a radius and corresponds to the set of points of the space whose distance to the center point is less or equal to the radius. Looking at (b) and (d), we can see that for a given ball of arbitrary radius r, and one selected cell of its left border, there are many cells on the right forming a diameter for the ball: any of them is at distance $2r$ from the left cell and, as a consequence, has a shortest path joining it with the left cell and passing through the center of the ball. In the same vein, a pair of cells is not a diameter for a unique ball. Indeed, the uniqueness of this ball in Euclidean space is implied by the uniqueness of the shortest path linking two points. When many shortest paths exist between two points, they may have different middles, which leads to different balls, all having the same radius, but each being centered at a different middle as shown in (a) and (c).

Because of these differences between the Euclidean and cellular case, the original Gabriel graph definition in terms of discs does not correspond to a connected graph in the cellular case. Figure 10.16 gives an example of loss of connectedness due to the fact that there might be no ball having only two seeds. In (a), every ball containing two seeds contains an additional third seed, preventing the two points to have a Gabriel edge. In (b), every ball containing two seeds of different lines contains four additional seeds, preventing the lines to be belong to the same connected Gabriel component, and so on.

10.5.2.2 Metric Gabriel Graphs

We therefore need to modify the definition to take into account these particularities in a meaningful way. In [11], an examination of the nature of Gabriel graphs and its transformation for arbitrary metric spaces is given and the following definition

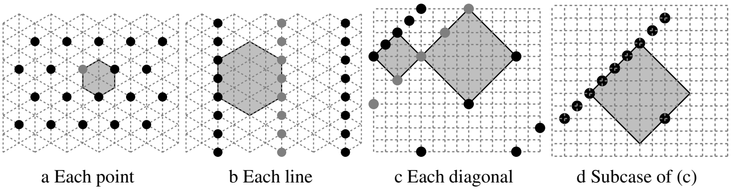

a Each point b Each line c Each diagonal d Subcase of (c)

Fig. 10.16 Hexagonal and square grid: the Gabriel graph is not connected. Grey seeds show one of the connected components and gray balls give the reason of the non-connectedness.

of **metric Gabriel graph** is obtained: its vertices are the seeds as before, and two seeds x and y have an edge connecting them if and only if there is a ball such that the seeds contained in this ball can be partitioned in two sets X and Y with $x \in X$, $y \in Y$ and $d(X,Y) = 2r$, r being the radius of the ball[2]. As before, such a ball if called a **metric Gabriel ball** and its center a **metric Gabriel center**.

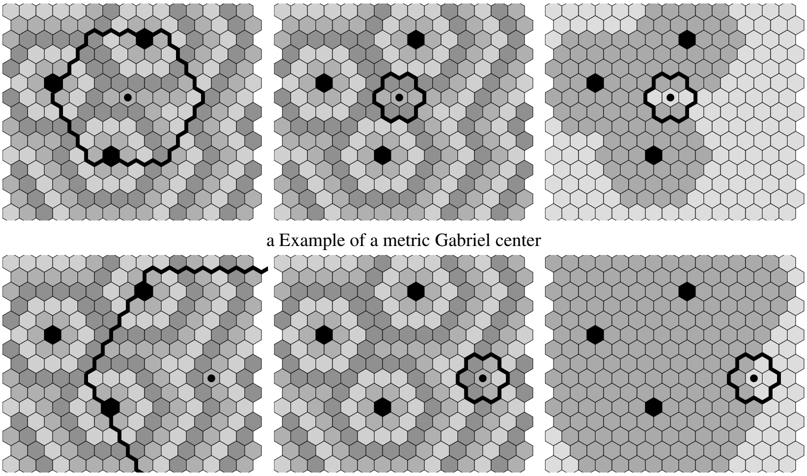
Compared to the original definition of Gabriel graphs, this definition deals with the non-uniqueness of diameters by replacing the requirement of having two seeds x and y such that $d(x,y) = 2r$ (this means precisely that $[xy]$ is a diameter) by the requirement of having two sets of seeds X and Y such that $d(X,Y) = 2r$. The non-uniqueness of balls for a given diameter is managed by requiring only the existence of at least one metric Gabriel ball. This means that whenever diameters and balls are unique, this definition is equivalent with the original one, as it is the case for Euclidean spaces. In fact, metric Gabriel graphs correspond exactly to original Gabriel graphs when the distance function is the Euclidean one. Moreover, *metric Gabriel graphs are always connected* for any set of points in any arbitrary metric space.

So this corresponds exactly to what we want to construct with our detection of middles. More precisely, metric Gabriel centers are precisely the detectable middles. To detect them, we use a very useful relation between metric Gabriel graphs and distance fields: in the same way a ball of radius r is a metric Gabriel ball when the set of seeds that it contains can be separated in two sets of distance $2r$, a distance field neighborhood of radius r corresponds to that of a detectable middle when its set of minimally valued cells can be separated in two sets of distance $2r$. This does not hold in arbitrary metric spaces but is true in cellular spaces.

Figures 10.17a and 10.17b give an example of this correspondence in the case of an hexagonal cellular space with three seeds.

In Fig. 10.17a, the first configuration shows a cell, indicated with a little dot, that is the center of a metric Gabriel ball. Indeed, the two seeds that it contains forms a diameter, which can be checked by exhibiting a shortest path joining them and passing through the center, or simply by checking that they have distance 8, while the ball is of radius 4. When considering only the neighborhood of this cell, in the second configuration, the only available information is the distance values computed

[2] Another way to put it is that there is an edge between two seeds x and y if and only if there is a ball such that x and y are at distance $2r$ and any other seed inside the ball is at distance $2r$ from x or from y.


a Example of a metric Gabriel center

b Example of a cell which is not a metric Gabriel center

Fig. 10.17 Relation between distance fields, metric Gabriel balls, and union of balls

from the seeds, not the actual seeds position. Taking the minimally-valued cells in this neighborhood of radius 1, we can see that they can be partitioned in two sets of distance 2, which means that this partition forms a diameter for the neighborhood. In fact, this neighborhood is a metric Gabriel ball for a *dilation* of the union balls of of radius 3 centered at each of the three seeds, as shown in the third configuration.

In Fig. 10.17b, we can see the converse. The first configuration shows a cell, indicated with a little dot, that is *not* a metric Gabriel center. Indeed, the two seeds do not form a diameter for this ball. When considering only the neighborhood of this cell, in the second configuration, there is no way to partition the three minimally-valued cells of the neighborhood in two sets of distance 2.

To summarize, the distance field values in the neighborhood of a cell allows it to detect whether it is a metric Gabriel center, and using this detection for the detection of middle required in Sect. 10.4.2.1 and Sect. 10.4.2.2, we know that sufficiently many pairs of seeds will be connected to obtained only one connected component and construct the complete convex hull.

10.6 The Complete Cellular Automaton

We are now in position to write the complete cellular automaton constructing the convex hull for arbitrary set of seeds, and for any cellular spaces[3]. It is simply a

[3] For the sake of brevity, some parts are not sufficiently general to be used without modification in all cellular spaces, but we believe that the main "understanding" is in this restriction version. We hope that any reader who needs the extra missing piece of generality will be able to guess it.

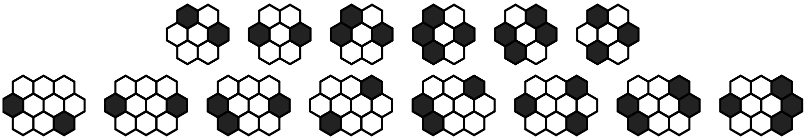


Fig. 10.18 Local configurations of middle cells and middle edges in hexagonal grids

compound of all the building blocks we have introduced throughout the chapter, but we now show explicitly how these building blocks needs to be linked:

$$\mathrm{dist}_{t+1}(x) = \begin{cases} 0 & \text{if } x \in P; \\ \mathrm{dist}_t(x) + 1 \bmod 3 & \text{if } x \notin P \wedge \forall y \in N(x);\ \mathrm{dist}_t(y) = \mathrm{dist}_t(x); \\ \mathrm{dist}_t(x) & \text{if } x \notin P \wedge \exists y \in N(x);\ \mathrm{dist}_t(y) \neq \mathrm{dist}_t(x); \end{cases}$$

$$\mathrm{cent}_{t+1}(x) = \bigvee \begin{cases} \{z \in N(x) \mid \mathrm{dist}_t(z) = \mathrm{dist}_{t+1}(x) - 1\} \text{ diam. } 2\ ; \\ \exists y \in N(x); \{z \in N(xy) \mid \mathrm{dist}_t(z) = \mathrm{dist}_t(xy) - 1.5\} \text{ diam. } 3\ ; \end{cases}$$

$$\mathrm{back}_{t+1}(x) = \mathrm{cent}_{t+1}(x) \vee \exists y \in N(x);\ \mathrm{dist}_t(y) = \mathrm{dist}_t(x) + 1 \bmod 3 \wedge \mathrm{back}_t(y);$$

$$\mathrm{conv}_{t+1}(x) = \mathrm{back}_{t+1}(x)\ \vee\ \exists y_0, y_1 \in N(x);\ \mathrm{conv}_t(y_0) \wedge \mathrm{conv}_t(y_1) \wedge x \in [y_0, y_1].$$

Here the metric Gabriel centers detection is written using two cases: one for diameters of even length, and the other for the diameters of odd length. In the first case, it is sufficient to look at the minimally-valued neighbors (those having value $\mathrm{dist}_{t+1}(x) - 1$) and to check if they can be separated in two sets of distance 2, i.e. the diameter of the neighborhood. For the second case, the center is between two cells x and y, and we need to look at the "neighborhood of the edge (x,y)" of radius 1.5. So $N(xy)$ is a shorthand for $N(x) \cup N(y)$, and $\mathrm{dist}_t(xy)$ is a shorthand for $\min(\mathrm{dist}_t(x) + 0.5, \mathrm{dist}_t(y) + 0.5)$ which is the distance of the edge to the closest seeds. If the minimally-valued neighbors (those having value $\mathrm{dist}_t(xy) - 1.5$) can be separated in two sets of distance 3 (the diameter of the neighborhood), then x is the extremity of an edge-like metric Gabriel center. For concreteness, Fig. 10.18 shows the set of minimally-valued neighbors that can be separated in two sets whose distance is the diameter for the case of the hexagonal cellular space. The first line is for cell-centers, and the second line for edge-centers.

The evolution of this cellular automaton for hexagonal cellular space and without the conv field was already shown in Fig. 10.12. The reader can now recheck the detection of the middles in the light of our explanation of the relation with metric Gabriel graphs, balls and centers. When it comes to the number of states, there are two things to say. Firstly, when restricting to the content of this chapter, namely the construction of the convex hull for a statis set of seeds, it is not useful to distinguish between cells having been marked the cent, back and conv rules. These three detections can be reduced to one Boolean field summarizing the five reasons to be marked as belonging to the convex hull: being a seed, being a cell-like metric Gabriel center, being a edge-like metric Gabriel center, being on the way back from a marked cell and its closest seed, and being between two marked cells in the neighborhood:

$$\mathrm{mark}_{t+1}(x) = \bigvee \begin{cases} x \in P; \\ \{z \in N(x) \mid \mathrm{dist}_t(z) = \mathrm{dist}_{t+1}(x) - 1\} \text{ diam. } 2\ ; \\ \exists y \in N(x); \{z \in N(xy) \mid \mathrm{dist}_t(z) = \mathrm{dist}_t(xy) - 1.5\} \text{ diam. } 3\ ; \\ \exists y \in N(x);\ \mathrm{dist}_t(y) = \mathrm{dist}_t(x) + 1 \bmod 3 \wedge \mathrm{mark}_t(y); \\ \exists y_0, y_1 \in N(x);\ \mathrm{mark}_t(y_0) \wedge \mathrm{mark}_t(y_1) \wedge x \in [y_0, y_1]. \end{cases}$$

This construction therefore requires 7 states: 3 states for the distances modulo 3, multiplied by 2 states for the mark field, plus 1 special state for the seeds that always have dist $= 0$ and mark $= \top$. Its neighborhood radius is 2 because of the edge-like metric Gabriel centers detection. Secondly, if one wants to consider moving seeds for example, it is necessary to keep the distinction between the fields cent, back and conv.

Acknowledgements. The authors would like to thank A. Adamatzky for his important involvement in the development of this work. He was the one who incited them to begin this work and also the one who told them about Gabriel graphs when they were seeking for the appropriate subgraphs of Delaunay graphs which is naturally computed by the pairwise algorithm.

References

1. Adamatzky, A.: Voronoi-like partition of lattice in cellular automata. Mathematical and Computer Modelling 23, 51–66 (1996), doi:doi:10.1016/0895-7177(96)00003-9
2. Aupetit, M., Catz, T.: High-dimensional labeled data analysis with topology representing graphs. Neurocomputing 63, 139–169 (2005)
3. Aurenhammer, F.: Voronoi diagrams - a survey of a fundamental geometric data structure. ACM Comput. Surv. 23(3), 345–405 (1991), `http://doi.acm.org/10.1145/116873.116880`
4. Chen, H., Wei, W.: Geodesic Gabriel graph based supervised nonlinear manifold learning. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCIS, vol. 345, pp. 882–887. Springer, Heidelberg (2006)
5. Clarridge, A.G., Salomaa, K.: An improved cellular automata based algorithm for the 45-convex hull problem. Journal of Cellular Automata 5(1-2), 107–120 (2010)
6. Gabriel, R.K., Sokal, R.R.: A new statistical approach to geographic variation analysis. Systematic Zoology 18(3), 259–278 (1969)
7. Ilachinski, A.: Cellular Automata: A Discrete Universe. World Scientific Publishing Co., Inc., River Edge (2001)
8. Kanj, I.A., Perković, L., Xia, G.: Local construction of near-optimal power spanners for wireless ad hoc networks. IEEE Transactions on Mobile Computing 8(4), 460–474 (2009), `http://doi.ieeecomputersociety.org/10.1109/TMC.2008.132`

9. Lee, C., Kim, D.-U., Shin, H., Kim, D.-S.: Efficient computation of elliptic Gabriel graph. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 440–448. Springer, Heidelberg (2006)
10. Maignan, L., Gruau, F.: Integer gradient for cellular automata: Principle and examples. In: Spatial Computing Workshop at IEEE SASO 2008 (2008)
11. Maignan, L., Gruau, F.: Gabriel graphs in arbitrary metric space and their cellular automaton for many grids. ACM Trans. Auton. Adapt. Syst. (2010)
12. Mukherjee, K.: Application of the Gabriel graph to instance based learning algorithms. Master's thesis, SFU CS School (2004)
13. Park, J.C., Shin, H., Choi, B.K.: Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation. Computer-Aided Design 38(6), 619–626 (2006)
14. Torbey, S., Akl, S.G.: An exact and optimal local solution to the two-dimensional convex hull of arbitrary points problem. J. Cellular Automata 4(2), 137–146 (2009)