Pergamon

Compput. & Graphics. Vol. 18, No. 5, pp. 695–706, 1994

0097-8493/94 $6.00 + .00

0097-8493(94)00094-8

Technical Section

DISCRETE CIRCLES, RINGS AND SPHERES

ERIC ANDRES
Centre de Recherche en Informatique, Université Louis Pasteur 7, rue René Descartes,
67084 Strasbourg Cédex, France, E-mail: andres@dpt-info.u-strasbg.fr

Abstract—This paper presents a new approach to discrete circles, rings, and an immediate extension to spheres. The circle, called *arithmetical circle* is defined by diophantine equations. The integer radius circles with the same centre pave the plane. It is easy to determine if a point is on, inside, or outside a circle. This was not easy to do with previous definitions of circles, like Bresenham's. We show that the arithmetical circle extends Bresenham's circle. We give an efficient incremental generation algorithm. The arithmetical circle has many extensions. We present briefly half-integer centered circles with a generation algorithm, 4-connected circles, and a general ring definition. We finish with the arithmetical sphere, an immediate 3D extension of arithmetical circle. We give elements to build a algorithm for generating the sphere.

1. INTRODUCTION

Computer graphics rests, at the moment, on adaptation of continuous reality to finite computer world[1]. This does not go without conflicts[2]. Algorithms based on reasonings of Euclidean geometry or topology are most often unreliable in the discrete world. Reasonings based on discrete geometry give robust algorithms instead. It's that thought process that we apply to circles in this paper. This approach is defined, presented, and developed in [3]. It has been applied to anti-aliasing[4–5], "Quasi-Affine Maps" have been developed[6], *etc.* Our purpose is to define discrete objects that we can call "discrete circles," as discrete models of the continuous "perfect" circle.

Bresenham's circle is an example of a circle based on an euclidean reasoning. It's formed by the "nearest" points of the Euclidean circle[7]. This definition does not easily define which points are inside, outside, or belong to a circle. Also, if one draws all Bresenham's circles with same center and varying integer radii, then about 10% of the points of the plane are not covered. This is a problem for many applications[8–9].

We will introduce the *arithmetical circle* based on a discrete reasoning. We will see that this circle solves many of the drawbacks of Bresenham's circle, like plane pavement, *etc.*

We first give a new general definition of a ring, and then we introduce the definition of a discrete object that we call the *arithmetical circle.* The circle is defined as a set of points solutions of diophantine equations. We place the *arithmetical circle* in relation to previous definitions of discrete circles. We show, in particular, that the *arithmetical circle* is an extension of Bresenham's circle[10–12]. We give the properties of the *arithmetic circle* like plane pavement. In Section 3, we present an efficient incremental generation algorithm with a proof. Performance analysis with common circle generators is given. In Section 4, we study half-integer centered circles. We give a generation algorithm. We briefly present four-connected circles and rational point-centered circles[13].

We end this paper with a study of the *arithmetical sphere*, an immediate extension of the *arithmetical circle*. We give its definition, properties, and elements to build an efficient generation algorithm.

2. ARITHMETICAL CIRCLE

Many papers have dealt with discrete circles[7, 14–15]. Since the first incremental algorithms[10], many variants have been given. The incremental generation algorithms are the most common[10–12, 16–18], as well as algorithms derived from previous ones and improving their performances[13, 19–21]. But other methods have been developed, from real and discrete differential equations[22–23], splines[24], or polygonal approximations[25], without forgetting generation algorithms using antialiasing[26], or parallel algorithms[27].

Applications of circle generation are various. For example, simple generation of geometric objects like circles, disks or rings[15], wave propagation simulation[9], or generation of thick curves[8]. Even the arithmetical circle, as we called the circle presented in this paper, has already been studied once[15]. But the approach was not based on a discrete definition, and the resulting generation algorithm was not efficient.

We first present the principle of Bresenham's circle generation algorithm[10–12] and study its drawbacks[9]. Then we will examine the arithmetical circle and its advantages in comparison with Bresenham's.

2.1. *Bresenham's circle*

Bresenham's circle[7, 10–12] is constituted by the "nearest" integer points from a continuous circle (Fig. 1). The well-known Bresenham's algorithm is an incremental algorithm working on the circle's upper right quarter, starting at point coordinates $(0, R)$. R is the integer radius of the generated circle. So, if $C(x, y)$ is the last generated point of the discrete circle, then the nearest point $A(x + 1, y)$ or $B(x + 1, y - 1)$ of the continuous circle will be the next generated point. The point $(x, y - 1)$ is not considered. This approach leads to problems (Fig. 1). Many points of plane do not belong to any circle from those with a given centre and varying integer radius. This is a problem for many applications[8–9]. It is not easy to characterize these

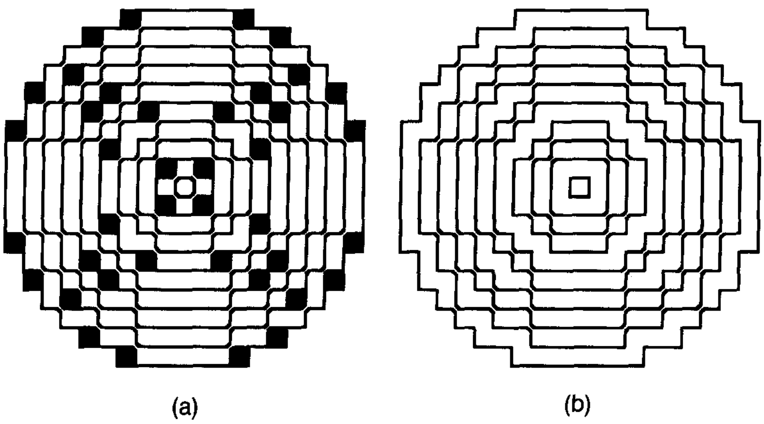

Fig. 1. (a) Bresenham's circles: the black point do not belong to any circle; (b) Arithmetical circles: each point belongs to one and only one circle.

points, and so it is not simple to determine efficiently if a point is outside, inside, or on a Bresenham's circle.

2.2. *Arithmetical circle*

First we will give the general definition of a ring:

Definition 1: A ring $R(A, B, x_o, y_o)$ is defined by: $(x, y) \in R(A, B, x_o, y_o)$ if $(x - x_o)^2 + (y - y_o)^2 \in [A, B[$; where $x, y \in \mathbb{Z}$, and $A, B, x_o, y_o \in R$.

It is very easy to verify if a point is inside, outside or belongs to a given ring.

Definition 2: A point (x, y) is said to be:

- inside $R(A, B, x_o, y_o)$ if $(x - x_o)^2 + (y - y_o)^2 < A$;
- outside $R(A, B, x_o, y_o)$ if $(x - x_o)^2 + (y - y_o)^2 \geq B$.

Definition 3: The arithmetical circle $C(R, x_o, y_o)$ is defined by: $C(R, x_o, y_o) = R((R - \frac{1}{2})^2, (R + \frac{1}{2})^2, x_o, y_o)$; where $R \in \mathbb{N}$.

The arithmetical circle $C(R, x_o, y_o)$ of radius R, and centre (x_o, y_o) is a ring.

The points of the arithmetical circle are the points with integer coordinates of a ring of radius R, centre (x_o, y_o) and thickness equal to 1 in euclidean plane (Fig. 2).

From definition we can deduce: $(x, y) \in C(R, 0, 0)$ if $(R - \frac{1}{2})^2 \leq x^2 + y^2 < (R + \frac{1}{2})^2 \Leftrightarrow R^2 - R + \frac{1}{4} \leq x^2 + y^2 < R^2 + R + \frac{1}{4} \Leftrightarrow R^2 - R + 1 \leq x^2 + y^2 \leq R^2 + R$.

This circle has already been studied once[15]. But the definition and the generation algorithm did not use integer but real numbers. So the definition we propose is much simpler and the deduced algorithm more efficient (see Section 3).

2.3. *Properties of arithmetical circle*

Without loss of generality, we can consider discrete and continuous circles and spheres as centered in $(0, 0)$.

We can easily deduce the following properties from the definition:

$$\left\{\begin{array}{ll} (1) & \text{The radii are integers.} \\ (2) & \text{each digital plane point belongs} \\ & \text{to one and only one circle (Fig. 2)} \\ (3) & \text{The arithmetical circle is a } \textit{good} \text{ approximation} \\ & \text{of an euclidean circle.} \end{array}\right.$$

Bresenham's circle does not have property 2. We will show that the arithmetical circle is an extension of Bresenham's circle.

Theorem 1: Extension of Bresenham's circle: The Bresenham's circle of radius R, centred in $(0, 0)$ is included in $C(R, 0, 0)$.

Proof by induction: We will show by induction that all points of the second octant $(0 \leq x \leq y)$ of Bresenham's circle belong to the arithmetical circle of the same ra-

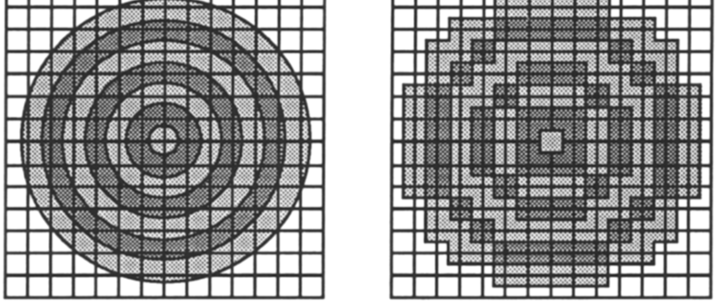

Fig. 2. The arithmetical circles pave the digital plane.

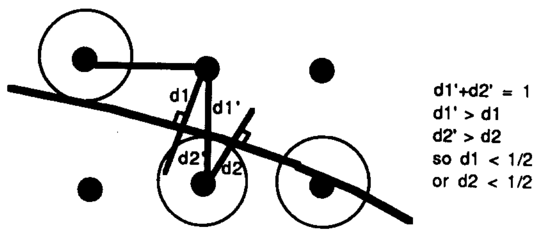

Fig. 3. Bresenham's circle is included in arithmetical circles.

dius. With the 8-symmetry on both circles, it will be enough to prove that a Bresenham's circle is included in the arithmetical circle of same radius. All points we speak about in this demonstration are in the second octant. Point $(0, R)$ belongs to both circles.

Let $D(x, y)$ be a point of Bresenham's circle and of the arithmetic circle. One of the two points $A(x + 1, y)$ and $B(x + 1, y - 1)$ is also a point of Bresenham's circle. Let CC be the continuous circle of radius R centered in $(0, 0)$. Let $P(x + 1, Y)$ be the point of the continuous circle CC having abscissa $x + 1$. There are four possibilities:

1. A and B are inside CC (distance to $(0, 0)$ is less than R). A is the point of Bresenham's circle because A is nearer of CC than B. A is also obviously nearer from CC than D so A belongs to the arithmetical circle.
2. A and B are outside CC. B is the point of Bresenham's circle because B is nearer than A from CC. If the distance from B to CC is greater than $\frac{1}{2}$, then CC has a slope between D and B greater than 1. This is not possible in the second octant. So B is a point of the arithmetical circle and a point of the Bresenham's circle.
3. A is outside CC and B is inside. Let us define $d1' = y - Y$, and $d2' = Y - (y - 1)$; $d1$ is the distance from A to CC and $d2$ the distance from B to CC (Fig. 3). If $d1 = 0$ (resp. $d2 = 0$) then A (resp. B) belongs to the arithmetical circle and Bresenham's circle. Let $d1 > 0$ and $d2 > 0$; then $d1' > d1$, $d2' > d2$ and $d1' + d2' = 1$, so $d1 + d2 < 1$ and $d1 < \frac{1}{2}$ or $d2 < \frac{1}{2}$ (Fig. 3). This proves that the nearest point of CC, the point belonging to Bresenham circle, is a point of the arithmetical circle.
4. A is inside CC and B outside. Same demonstration as in 3). ■

3. CIRCLE GENERATION

The arithmetical circle can be generated by solving the diophantine equations of its definition. The resolution of these equations is long known [18, 28–29].

This mathematical approach of the circle generation reveals a complex underlying arithmetical structure. This algorithm is less efficient, in a nonparallel context, than the incremental one that we present now. We will propose an incremental generation algorithm for the arithmetical circle without solving diophantine equations. This provides an efficient non parallel algorithm.

3.1. *Bases of the incremental algorithm*

The algorithm is based on two theorems. These theorems will allow us to build the incremental generation algorithm and prove its correctness.

The starting idea of this algorithm is the same as Bresenham's, but when Bresenham's chooses between two points the one to be generated next, we choose between three points (Fig. 4). Also, we don't search the point that is the nearest point of continuous circle, instead the point verifying the arithmetical definition.

Let (x, y) be a point of the arithmetical circle $C(R, 0, 0)$ of radius R. Let us examine the conditions for which $A(x + 1, y)$, $B(x, y - 1)$ or $C(x + 1, y - 1)$ (resp. case (a), (b) and (c)) belongs also to the circle. Let us examine the two first cases:

(a) $(x + 1, y)$ belongs to the circle if $(x + 1)^2 + y^2 \le R^2 + R$. So A is a point of the circle if $x^2 + y^2 \le R^2 + R - 2x - 1$

(b) $(x, y - 1)$ belongs to the circle if $x^2 + (y - 1)^2 \ge R^2 - R + 1$. So B is a point of the circle if $x^2 + y^2 \ge R^2 + R + 2(y - R)$.

This brings us to a first important result:

Theorem 2: Case (a) and (b): (Fig. 5). It's not possible to have both cases (a) and (b) together.

Proof by contradiction: Let us suppose that we have both cases (a) and (b): This means that $R^2 + R + 2(y - R) \le x^2 + y^2 \le R^2 + R - 2x - 1$; so $2x + 2y + 1 \le 2R$, which is equivalent to $2x + 2y + 2 \le 2R$, and $R - 1 \ge x + y$. Thus $R^2 - 2R + 1 \ge x^2 + y^2 + 2xy$.

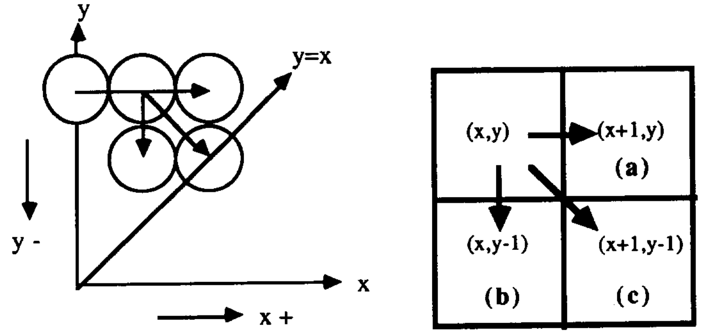

Fig. 4. The algorithm searches the points verifying diophantine definition.

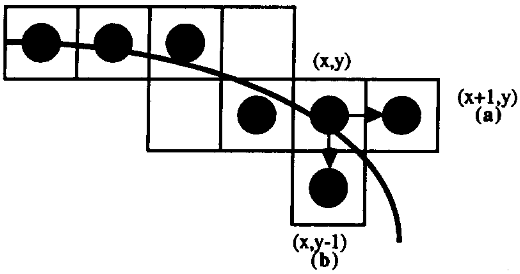

Fig. 5. $(x+1, y)$ and $(x, y-1)$ cannot all two belong to the circle.

But $R^2 - R + 1 \le x^2 + y^2 \le R^2 + R$, because (x, y) is a point of the circle. So $R^2 - 2R + 1 \ge x^2 + y^2 + 2xy \ge R^2 - R + 1 + 2xy$, and $-R \ge 2xy$. This is impossible because $x \ge 0$ and $y \ge 0$. ■

By hypothesis $A(x+1, y)$ is not a point of the circle; so $x^2 + y^2 > R^2 + R - 2x - 1$, and $x^2 + y^2 \ge R^2 + R - 2x$. Finally $x^2 + y^2 \ge R^2 - R - 2x - 2y - 1$, because for radii greater than 1, $R \ge y + 1$. This finishes the proof that $(x+1, y-1)$ belongs to the arithmetical circle. ■

A last result that is important for algorithm optimisation is:

Theorem 4: After (b) comes (a): If (x, y) and $(x, y-1)$ belong to the circle, then $(x+1, y-1)$ belongs also to it.

Proof: A proof by contradiction is simple to obtain. ■

3.2. *Generation algorithm*

With Theorems 2 and 3 it is easy to deduce an algorithm.

```
BASIC ALGORITHM OF CIRCLE GENERATION (x0,y0, R: integer)
{The point (x0,y0) is the centre and R the radius of the circle to be generated}
x := 0
y := R
∂ := R²                         {∂ will always be equal to x² + y²}
While (y ≥ x) do                {Stopping condition of algorithm}
      plot_8_pixels (xo, yo, x, y)   {Plot 8 pixels, one in each octant}
      if (∂ ≤ R² + R −2x −1)
      then                       {case (a) : 4-connected jump on abscissa axis}
        | ∂ := ∂ + 2x +1         {∂ = ((x+1)²+y²) := ∂ + 2x +1}
        | x := x + 1
      else if (∂ ≥ R² + R 2(y−R))
        | then                   {case (b) : 4-connected jump on ordinate axis}
        |   | ∂ := ∂ − 2y +1     {∂ = (x²+(y−1)²) := ∂ − 2y + 1}
        |   | y := y −1
        | else                   {case (c) : 8-connected jump. Using Theorem 3.}
        |   | ∂ := ∂ + 2(x−y+1)  {∂ = (x+1)²+(y−1)² := ∂ + 2(x−y+1)}
        |   | x := x + 1
        |   | y := y − 1
        | endif
      endif
end-while
end-algorithm
```

The second important result:

Theorem 3: Cases (a), (b) and (c): (Fig. 6). If we are neither in case (a) nor in case (b) then we are in case (c).

Proof: Let us prove the first condition for the case (c), namely that $(x+1)^2 + (y-1)^2 \le R^2 + R$. Let us consider a circle of radius greater than 1. For the circles of radii 0 and 1 the theorem is obvious. By hypothesis $B(x, y-1)$ is not a point of the circle, so $x^2 + y^2 < R^2 + R + 2(y - R)$, and $x^2 + y^2 \le R^2 + R - 1 + 2(y - R)$. This gives $x^2 + y^2 \le R^2 + R - 1 + 2(y - R) \le R^2 + R - 1 + 2(y - x - 1)$; because for radii greater than 1, $R \ge x + 1$. Finally, $x^2 + y^2 \le R^2 + R - 2 + 2(y - x)$, and $(x+1)^2 + (y-1)^2 \le R^2 + R$. Let us prove the second condition for the case (c), namely that $R^2 - R + 1 \le (x+1)^2 + (y-1)^2$.

3.3. *Proof of the algorithm*

We will prove the correctness of the generation algorithm for a circle of radius R and centre $(0, 0)$.

(a) Do all generated points belong to the arithmetical circle?

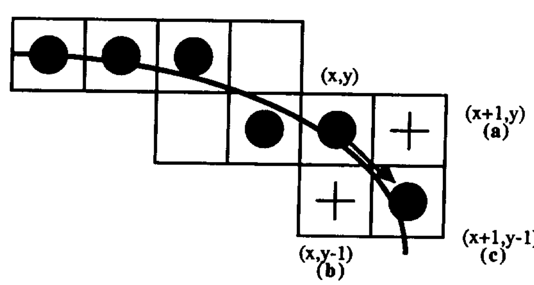

Fig. 6. If neither $(x+1, y)$ nor $(x, y-1)$ belongs to the circle then $(x+1, y-1)$ does.

- Starting point $(0, R)$ belongs to the circle.
- At each moment ∂ is equal to $x^2 + y^2$. We know that if $\partial = x^2 + y^2 \le R^2 + R - 2x - 1$, then $(x + 1, y)$ belongs to the circle of radius R. So case (a) in algorithm preserves belonging to the arithmetical circle of radius R.
- We know that if $\partial \ge R^2 + R + 2(y - R)$ then $(x, y - 1)$ belongs to the circle. So case (b) in algorithm preserves belonging to the arithmetical circle.
- And Theorem 3 says that if we are not in case (a), nor in case (b) then we have to generate the point $(x + 1, y - 1)$, which belongs to the circle. It is exactly what the algorithm does.

  Point $(0, R)$ belongs to the arithmetical circle of radius R and all three cases generate new points belonging to the circle if the current point does. This proves that all points generated by this algorithm belong to the circle.

(b) Are all points of $C(R, 0, 0)$ generated by this algorithm? Theorem 2 and 3 prove that the circle is 8-connected in the octant. Let (x, y) be a point of $C(R, 0, 0)$ in the working octant (Fig. 7).

- Let us show first that $(x + 1, y + 1)$ and $(x - 1, y - 1)$ do not belong to the circle.

  For $(x + 1, y + 1)$ we have $(x + 1)^2 + (y + 1)^2 = x^2 + y^2 + 2(x + y + 1) > R^2 + R$ (because it is easy to show that $x + y \ge R$). This proves that $(x + 1, y + 1)$ does not belong to the circle if (x, y) does.

  For $(x - 1, y - 1)$ we have $(x - 1)^2 + (y - 1)^2 = x^2 + y^2 - 2(x + y - 1) \le R^2 + R - 2x - 2y + 2$ so $(x - 1)^2 + (y - 1)^2 < R^2 - R + 1$ (because $x + y \ge R$).

  So neither $(x + 1, y + 1)$, nor $(x - 1, y - 1)$ belong to the circle.
- The connectivity of the generated points on the working octant means that one of the three points α, β, or γ belongs to $C(R, 0, 0)$ and is generated by the algorithm (Fig. 7).

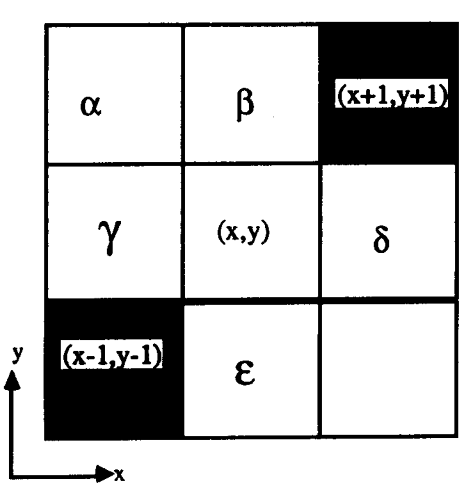

Fig. 7. We try to show that if (x, y) belongs to the circle then it will be generated.

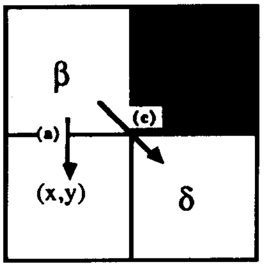

Fig. 8. Case b is tested before case c.

- If α is generated then the algorithm will choose case a, b or c: If it is case c, then (x, y) is generated and the proof is finished. If it is case a, then β is generated. See next. If it is case b, then γ is generated. See next.
- If β is generated (Fig. 8): The algorithm detects that $(x + 1, y + 1)$ does not belong to the circle (test in case a), detects that (x, y) belongs to the circle and detects whether δ is a points of the circle. As the test in case b (for (x, y)) occurs before case c (for point δ), the algorithm will generate (x, y) after β.
- If γ is generated (Fig. 9): (x, y) is generated for the same reasons as before, since case a is detected before case c.

This finishes the proof that (x, y) belongs to the arithmetical circle, and so that the algorithm generates all points of the circle $C(R, 0, 0)$. ■

3.4. *Improvement of the generation algorithm*

Instead of working with $\partial = x^2 + y^2$, and calculating many squares, we will work with a value Δ, which will permit us to avoid these calculations. In fact, it is not the value of ∂ that interest us but its variation.

Theorem 5: control value replacement: With $\Delta = R^2 + R - (x^2 + y^2)$ in place of ∂, only additions, subtractions and multiplications by two are necessary for generating the arithmetical circle.

Proof: At starting point $(0, R)$, instead of starting value $\partial = R^2$, we have the value $\Delta = R^2 + R - R^2 = R$. The condition for case a becomes: $\partial = x^2 + y^2 \le R^2 + R - 2x - 1$, so $\Delta > 2x$. When $\Delta > 2x$, we are in case a.

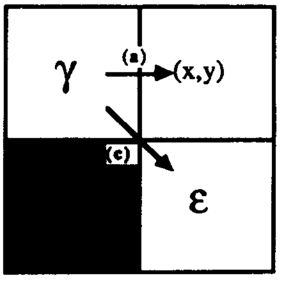

Fig. 9. Case a is tested before case c.

The condition for case b becomes: $\partial = x^2 + y^2 \geq R^2 - R + 1 + 2(y - R)$, so $\Delta \leq 2(R - y)$. When $\Delta \leq 2(R - y)$, we are in case b.

And when we are neither in case a nor in case b, then we are in case c. ■

The factor 2 appearing in the different expressions resulting from Theorem 5 can also be eliminated, as shown in the algorithm. A last improvement results from Theorem 4.

3.5. *Final incremental generation algorithm*

```
CIRCLE (x0,y0,R: integer)
{The point (x0,y0) is the centre and R the radius of the circle to be generated}

begin-algorithm
x := 0
y := R
Δ' := R/2                        {Δ' is equal to Δ/2}
Isodd := R and 1                 {Isodd is equal to Δ and 1, so Δ = 2*Δ' + Isodd}
Plot_4_pixels(xo, yo, R)         {Plot the 4 pixels on axis}
while (y > x)  do                {Stopping condition of algorithm}
  | if(isodd)
  |   | then x,y,Δ',isodd := ODD(x,y,Δ',R,Isodd,xo,yo);
  |   | else x,y,Δ',isodd := EVEN (x,y,Δ',R,Isodd,xo,yo);
  | end-if
  | plot_8_pixels (xo, yo, x, y)  {Plot 8 pixels, one in each octant}
end-while
End-algorithm
```

with the two very similar functions ODD and EVEN:

```
ODD(x,y,Δ',R,isodd,xo,yo: integer)
begin-function
if(Δ' > x)
then
  | Δ' := Δ'-x-1
  | x := x+1
  | Isodd := 0
else
  | if(Δ' <= (R-y))
  | then y := y-1 {case (b) and (c)}
  |      plot_8_pixels (xo, yo, x, y)
  | else y := y-1 {or just case (c)}
  | end-if
  |   Δ' := Δ' + y -x
  |   x := x+1
end-if
return(x,y,Δ',Isodd)
end-function
```

```
EVEN(x,y,Δ',R,isodd,xo,yo: integer)
begin-function
if(Δ' > x)
then
  | Δ' := Δ'-x
  | x := x+1
  | Isodd := 1
else
  | if(Δ' <= (R-y))
  | then x := y-1 {case (b) and (c). Th. 4}
  |      plot_8_pixels (xo, yo, x, y)
  | else x := y-1 {or just case (c)}
  | end-if
  | Δ' := Δ' + y -x
  | x := x+1
end-if
return(x,y,Δ',Isodd)
end-function
```

3.6. *Performance analysis*

To compare the performances of the most common algorithms, we have drawn the circles from radius 0 to 255 and counted all the operations done by each algorithm: incrementation, shift, test, and addition. The sum of these operations divided by the number of points to draw gives a performance rate: the number of operations needed to draw one point.

In Table 1 "points" is for the drawn points. "pts more" is for all the points drawn more than one time by the different algorithms. "ops/pt" is for the performance rate. It's the number of operations needed to generate one point.

"Bresenh." stands for the Bresenham's quadrant algorithm[11]. The same algorithm working on a octant gives a performance rate of ≈1.4 operations per drawn point, and the algorithm described in [12] is equivalent to [19]. "Mcilroy" stands for [7], "Kuzmin" for [19], "Biswas" for [15] and "New algorithm" for the final incremental algorithm given in Section 3.5.

The performance rate is not an absolute value. It will be modified when considering the processors. Here we did as if each operation would take the same time, which is obviously not true in reality. This gives an idea of algorithm performances.

The performance rate of arithmetical circle generation algorithm is equivalent to comparable algorithms[7]. Further study will be to find improvements like in [21]. The algorithm of Biswas and Chaudhuri[15] gives the arithmetical circle, but is very in-

Table 1. Performance analysis.

	Bresenh.	McIlroy	Kuzmin	Biswas	New algorithm
Inc	158.428	33.278	32.895	920.064	32.767
Shift	92.636	512	512	0	512
test	162.777	46.797	46.796	1.050.624	79.363
add	112.013	66.301	33.022	1.184.256	42.696
points	**184.637**	**184.637**	**184.637**	**205.101**	**205.101**
pts more	1.024	1.532	2.548	1.024	1.740
ops/pt	**2.85**	**0.80**	**0.61**	**19.23**	**0.76**

efficient. For the radii 0 to 255, there are 20.000 points not belonging to any Bresenham's circle. Nearby 10% of points are forgotten. This shows that this problem is not negligible. Experience shows that on average it's a rate of $\approx 10\%$ points not belonging to any Bresenham's circle.

4. OTHER CIRCLES

We have seen a definition of circles. This is not the only way to define discrete circles with diophantine equations. We will now examine an arithmetical circle with half-integer centre, and speak briefly about 4-connected circles and circles centered on rational points (see also [13]).

4.1. *Half integer centered circle*

This is another arithmetical circle that finds an use in many applications[8, 13].

Definition 4: The half-integer centered circle $CH(R, x_o, y_o)$ is defined by: $CH(R, x_o, y_o) = R((R - \frac{1}{2})^2, (R + \frac{1}{2})^2, x_o + \frac{1}{2}, y_o + \frac{1}{2})$; where $R \in N$. We assume from now on that the centre is $(\frac{1}{2}, \frac{1}{2})$ (Fig. 10). With what follows, it is easy to find the algorithms for the circles for centre $(0, \frac{1}{2})$ or $(\frac{1}{2}, 0)$.

The circle with half-integer centre has also an 8-symmetry, so we also work on only one octant. The symmetry is shown in the algorithm.

Let us first express the definition in another way:

$$(R - \tfrac{1}{2})^2 \le (x - \tfrac{1}{2})^2 + (y - \tfrac{1}{2})^2 < (R + \tfrac{1}{2})^2$$

So $R^2 - R + \frac{1}{4} \le x^2 + y^2 - x - y + \frac{1}{2} < R^2 + R + \frac{1}{4}$, and $R^2 - R \le x^2 + y^2 - x - y < R^2 + R$. What follows is nearly the same as in Section 3. So for explanations and details, report to Section 3.

To build the generation algorithm, let us examine the 3 cases a, b, and c: Let (x, y) be a point of the circle.

- Case a: Under which conditions does $(x + 1, y)$ belong to the circle?
Here $(x + 1)^2 + y^2 - x - 1 - y < R^2 + R$;
So we are in case a if $x^2 + y^2 - x - y < R^2 + R - 2x$.
- Case b: We are in case b if $x^2 + y^2 - x - y \ge R^2 + R + 2(y - R - 1)$.
- Case c: There exist theorems like Theorem 2 and 3 for half-integer centered circles. So case c occurs when neither case a nor case b occurs.

We will not prove the theorems here. We will just try to find a new control value in place of $x^2 + y^2 - x - y$:
Let us take $\partial = R^2 + R - (x^2 + y^2 - x - y)$ as new control value.

- For a starting point we have: The point $(0, R)$ is no more in the working octant (Fig. 10). The first point of the octant is $(1, R)$. This will be the starting point of the algorithm. Then control value will be at starting: $\partial = R^2 + R - (1^2 + R^2 - 1 - R) = 2R$.
- For case a we have: $\partial = R^2 + R - (x^2 + y^2 - x - y) > 2x$.
- For case b we have: $\partial = R^2 + R - (x^2 + y^2 - x - y) \le 2(R + 1 - y)$

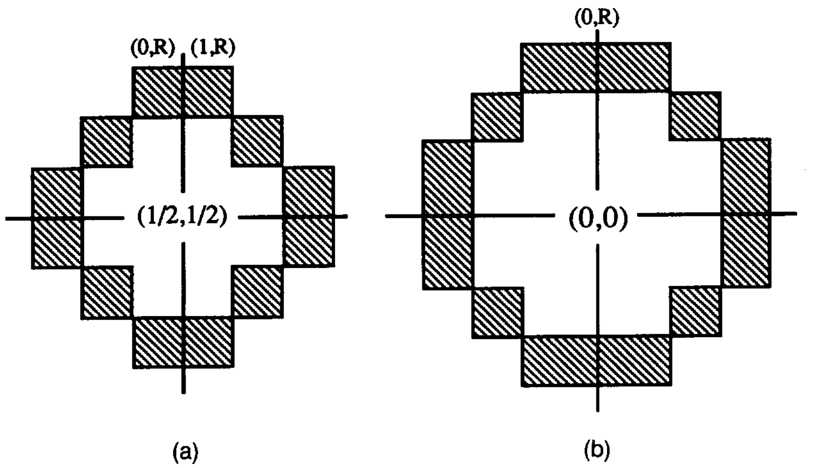

Fig. 10. (a) Circle of half integer centre; (b) circle with integer centre.

We can see that the expressions of the control value have 2 as common factor. As the value of the expressions do not interest us but only their relative values, we will simply eliminate this factor 2.

Finally this gives the following algorithm:

```
HALF INTEGER CENTERED CIRCLE (xo,yo,R : integer)
{The point (x0,y0) is the centre and R the radius of the circle to be generated}
x := S
y := R
∂ := R
While (y ≥ x) do        {Stopping condition of algorithm}
    plot_pixel(xo+x,yo+y)
    plot_pixel(xo+x,yo−y +1)
    plot_pixel(xo−x+1,yo+y)
    plot_pixel(xo−x+1,yo−y +1)        {8-symmetry with centre (1/2,1/2)}
    plot_pixel(xo+y,yo+x)
    plot_pixel(xo+y,yo−x +1)
    plot_pixel(xo−y+1,yo+x)
    plot_pixel(xo−y+1,yo−x +1)
    if (∂ > x)
    then {case (a) : 4-connected jump on abscissa axis}
        ∂ := ∂ − x.
        x := x + 1
        else if (∂ < (R+1−y))
        then {case (b) : 4-connected jump on ordinate axis}
            ∂ := ∂ + y − 1
            y := y − 1
        else {case (c) : 8-connected jump.}
            ∂ := ∂ + (y−x−1)
            x := x + 1
            y := y − 1
        end-if
    end-if
end-while
end-algorithm
```

4.2. *4-Connected circle*

We present briefly the 4-connected circle. This circle, or deduced sphere, can be useful when an ray opaque circle, or sphere is needed, in ray tracing for example[34].

Definition 5: The 4-connected circle $C4(R', x_o, y_o)$ is defined by: $C4(R', x_o, y_o) = R(2R^2, 2(R+1)^2, x_o, y_o)$; where $R' \in N$.

For a desired 4-connected circle of integer radius R, we must take $R' = \left[\frac{R}{\sqrt{2}}\right]$, where $[x]$ is integer part of x.

The points of the 4-connected circle are the points with integer coordinates of a ring of lower radius $\sqrt{2}R$ and upper radius $\sqrt{2}(R+1)$. The length of a pixel diagonal is $\sqrt{2}$, so if a point belongs to the circle, then two of his 4-connected neighbours also belong to the circle.

4.3. *Circle of rational centre*

We present, to complete this survey of circles defined by diophantine equations, the definition of the circles with rational centre.

Definition 6: The circle with rational centre $CR\left(CR, x_o, y_o, \frac{a}{b}, \frac{c}{d}\right)$ is defined by: $CR\left(R, x_o, y_o, \frac{a}{b}, \frac{c}{d}\right) = R\left((R-\frac{1}{2})^2, (R+\frac{1}{2})^2, x_o + \frac{a}{b}, y_o + \frac{c}{d}\right)$; where $a, b, c, d, x, y, x_o, y_o \in Z$ and $R \in N$.

The circle with rational centre has no central symmetry (Fig. 11). This means that there are problems to give an incremental algorithm. But this circle, as the 4-connected circle paves the plane. This important property is always preserved. More studies are necessary to obtain results as in [13], which describes a generation algorithm for similar circles as Bresenham's circles, with non-lattice points centers.

5. THE DISCRETE SPHERE

The sphere is, like the plane[18, 30], a basic object of discrete 3D geometry. We will show how the definition and generation of a discrete arithmetical sphere can be deduced from the definition of the arithmetical circle. After the definition, we will give the properties of the arithmetical sphere. We will finish with giving the elements, permitting design of an efficient generation algorithm[31–32].

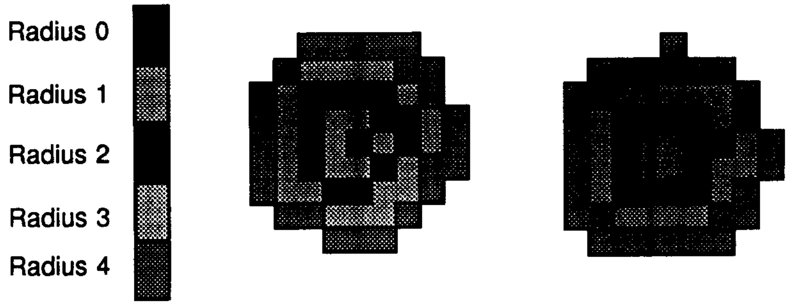

Fig. 11. (a) Circle of centre $(\frac{2}{3}, \frac{1}{5})$; (b) of centre $(\frac{11}{20}, \frac{21}{40})$.

5.1. *Definition of the arithmetical sphere*

The definition of the arithmetical sphere is an immediate 3D extension of the arithmetical circle's definition.

Definition 7: The arithmetical sphere $S(R, x_o, y_o)$ is defined by: $(x, y, z) \in S(R, x_o, y_o, z_o) \Leftrightarrow (R - \frac{1}{2})^2 \le (x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 < (R + \frac{1}{2})^2$; where $x, y, z, x_o, y_o, z_o \in Z$ and $R \in N$.

These diophantine equations are in direct relation with the continuous sphere. The solutions, and points of the discrete sphere, are the integer points that are at a Euclidean distance of $\sqrt{x^2 + y^2 + z^2}$ from the centre of sphere, with $R - \frac{1}{2} \le \sqrt{x^2 + y^2 + z^2} < R + \frac{1}{2}$.

5.2. *Properties of arithmetical sphere*

Compared to the arithmetical circle, the arithmetical sphere has a similar definition and similar properties:

$$\begin{cases} (1) & \text{The radii are integer.} \\ (2) & \text{Each digital point belongs to one and only one sphere} \\ (3) & \text{The arithmetical sphere is a } \textit{good} \text{ approximation of euclidean sphere.} \end{cases}$$

The properties can immediately be deduced from the definition. This means that the arithmetical spheres with same centre and varying integer radii pave the 3D digital space, like arithmetical circles pave the digital plane.

We have seen that the arithmetical circles are connected. A similar property can be given for spheres.

Theorem 6: Holes in arithmetical spheres: A discrete sphere has no 6-connected holes; it means that no 6-connected curve can go through a sphere without intersection.

Proof by contradiction: First the definition permits us to define what is inside or outside of a sphere, and so it is possible to define the notion of "passing through a sphere" clearly and easily.

Let us suppose that there exists a 6-connected hole in $S(R, x_o, y_o, z_o)$. Then there exists a point $A(a, b, c)$ such that $a^2 + b^2 + c^2 < (R - \frac{1}{2})^2$ and a point $B(a + 1, b, c)$ such that $(a + 1)^2 + b^2 + c^2 \ge (R + \frac{1}{2})^2$, with $a, b, c \in Z$ (because the sphere is symmetric).

Thus $a^2 + 2a + 1 + b^2 + c^2 \ge (R + \frac{1}{2})^2$, and so $a^2 + 2a + 1 + b^2 + c^2 + 1 > (R + \frac{1}{2})^2$; $a^2 + 2a + 1 + b^2 + c^2 + 1 - (a^2 + b^2 + c^2) > (R + \frac{1}{2})^2 - (R - \frac{1}{2})^2 = R^2 + R + \frac{1}{4} - R^2 + R - \frac{1}{4} = 2R$.

Hence $2a + 1 > 2R$; and $a \ge R$. This is in contradiction with $a^2 + b^2 + c^2 < (R - \frac{1}{2})^2$. ■

The sphere is based on the diophantine equation $x^2 + y^2 + z^2 = n$. So it is possible, like for circles, to give different other definitions of spheres.

5.3. *Generation of the arithmetical sphere*

We briefly give the elements that will allow the reader to build a sphere generation algorithm based on the circle generation algorithm.

Let us first make some preliminary remarks:

- The arithmetical sphere has an 48-symmetry.
- The sum of three squares does not offer, like sum of two or four squares does, a multiplying structure. It is much more difficult to calculate the solutions of the sphere's diophantine equation.
- On the other hand, the diophantine equation of the sphere can easily be transformed in order to give rise to those of the circle: $x^2 + y^2 = n - z^2$. And we know how to solve this equation for a given $n - z^2$. We just have to solve the diophantine equations $n' = x^2 + y^2$ for $(R - \frac{1}{2})^2 - z^2 \le n' < (R + \frac{1}{2})^2 - z^2$, with z varying from 0 to R. This shows that a slice (z constant) of an arithmetical sphere is not an arithmetical circle, but a general ring (Fig. 12).

The problem with the solution of diophantine equations with a sum of two squares is its slowness. This can be a problem because there are many equations to solve. It is possible to store the solutions of all the equations, but this represents a big file.

Another solution is to use the generation algorithm of arithmetical circles to build the one for spheres: We know a generation algorithm which solves equations

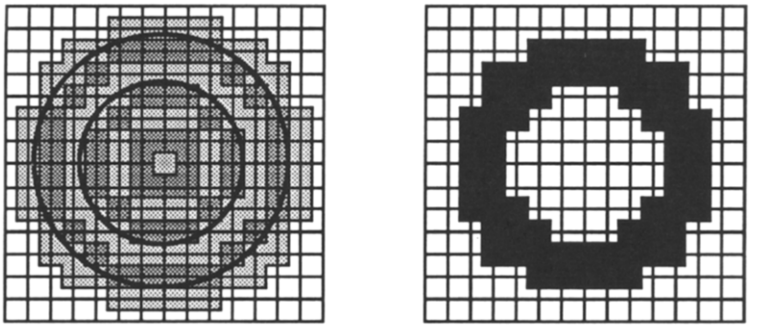

Fig. 12. A slice of an arithmetical sphere is a ring. A ring can be seen as pieces of several arithmetical circles.

with a sum of two squares equations $n' = x^2 + y^2$ for $(R - \frac{1}{2})^2 \le n' < (R + \frac{1}{2})^2$. It is enough to take, for z constant, all the arithmetical circles which have points (x_α, y_α) verifying: $(R - \frac{1}{2})^2 - z^2 \le x_\alpha^2 + y_\alpha^2 < (R + \frac{1}{2})^2 - z^2$. Then point (x_α, y_α, z) is a point of the arithmetical sphere. This means finding all circles of radius u_i such as:

$$(*) [(u_i - \tfrac{1}{2})^2, (u_i + \tfrac{1}{2})^2[\cap [(R - \tfrac{1}{2})^2 - z^2, (R + \tfrac{1}{2})^2 - z^2[\ne \varnothing.$$

```
BASIC-SPHERE ALGORITHM (x0,y0,z0,R : integer)
{The point (x0,y0,z0) is the centre and R the radius of the sphere to be generated}
begin-algorithm
For z := 0 to R do
    | Rmin := Give_min(z,R)                      {Give the smallest ui}
    | Rmax := Give_max(z,R)                      {Give the biggest ui}
    | cstmax := R^2 + R-z^2                      {Outer ring diameter of a sphere slice}
    | cstmin := cstmax - 2*R                     {Inner ring diameter of a sphere slice}
    | For Rcurr := Rmin to Rmax do               {generate pieces of arithmetical circles...}
    |   | circle_sphere(Rcurr,cstmin,cstmax, z)      {corresponding to a slice.}
    | end-For
end-For
end-algorithm

with the functions :

Give min(z,R: integer)
begin-function
rmin := R
Arcmin := R^2-R
mins = Arcmin - z^2
if(mins<0) then rmin := 0
else
    While (mins < Arcmin) do
        | rmin := rmin - 1
        | Arcmin := Arcmin - 2*rmin
    end-while
end-if
return(rmin)
end-function

Give max(z,R: integer)
begin-function
rmax := R
Arcmax := R^2-R
maxs = Arcmin +2*R - z^2
While (maxs < Arcmas) do
    | rmax := rmax - 1
    | Arcmax := Arcmas - 2*rmax
end-while
end-if
return(rmax)
end-function
```

Neither "$(R - \frac{1}{2})^2 - z^2$," nor "$(R + \frac{1}{2})^2 - z^2$" are squares, in general, and there can be many $[(u_i - \frac{1}{2})^2, (u_i + \frac{1}{2})^2[$ included in $[(R - \frac{1}{2})^2 - z^2, (R + \frac{1}{2})^2 - z^2]$. Since all u_i are consecutive, it is just necessary to find the smallest and biggest u_i verifying $(*)$.

With all this remarks it is now possible to design a generation algorithm for the arithmetical sphere. The presented algorithm is a basic one. It can be improved to obtain a more efficient algorithm. Some improvements are obvious, such as improving the function plot_48_voxels, and some are less. This improvements will be part of a forthcoming paper.

And the circle generation algorithm for the sphere: The circle generation algorithm we use is similar to the basic generation algorithm, with just the improvements corresponding to Theorem 5.

```
Circle sphere(Rcurrent,cstmin,cstmax, z : integer)
begin-procedure
x := 0
y := Rcurrent
Δ := Rcurrent
cst_local_min := Rcurrent²+R-cstmax     {with Th.5's changements for this circle}
cst_local_max := Rcurrent²+R-cstmin
while (y >= x) do
    if((Δ>=cst_local_min) and (Δ<cst_local_max))
    then plot_48_voxels (x0,y0,z0,x,y,z)   {there is an 48-symmetry for a sphere}
    end-if
    if(Δ > 2*x)                            {The arithmetical circle generation algorithm ...}
    then                                   {before division by 2 improvement}
      | Δ := Δ-2*x-1
      | x := x+1
    else if(Δ ≤ 2*(R-y))
    then
      | Δ := Δ + 2*y-1
      | y := y-1
    else
      | Δ := Δ + 2*(y-x-1)
      | x := x+1
      | y := y-1
    end-if
end-while
end-procedure
```

6. CONCLUSION

Our diophantine definition and our entirely discrete approach make the link with the general definition of discrete lines[3] and planes[30]. We present a efficient discrete incremental generation algorithm for arithmetical circles, with a proof of its correctness. The arithmetical sphere, natural extension of the arithmetical circle, is presented with properties and a basic generation algorithm.

The diophantine arithmetical definitions permit a precise, strictly discrete definition of the object we deal with. So the algorithms based on these definitions are more robust. But all problems are not solved: What does the intersection of a circle and a line, or of two circles, look like? The intersection between two spheres looks very complicated, is there any way to determine this intersection, like in [3] for digital lines? Also, the arithmetical complexity of the diophantine equations defining the circles and spheres poses problems for studying their properties (Fig. 13).

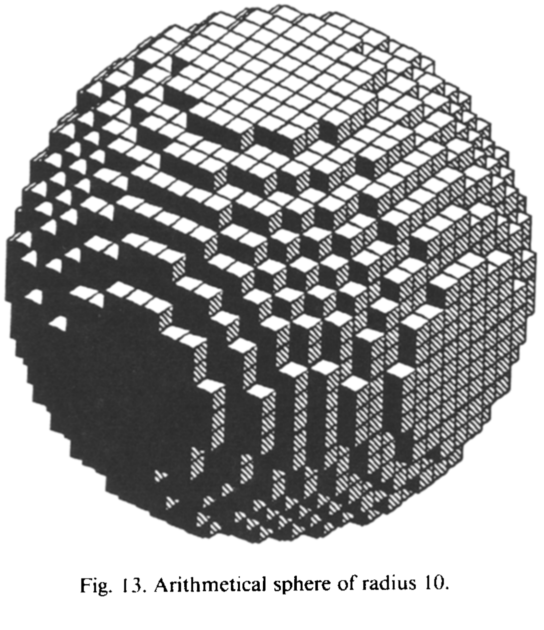

Fig. 13. Arithmetical sphere of radius 10.

Acknowledgements—The author would like to thank Pr. J. Françon and J-P. Reveillès for valuable discussions and help. He also thanks P. Nehlig and M. Tajine for their encouragements.

REFERENCES

1. J. Francon and J-P. Reveilles, De l'imagerie à la géométrie discrète. *Actes 115e Congr. nat. Soc. savantes*. Avignon, 323–339 (1990).
2. J. E. Bresenham, Ambiguities in incremental line rastering. *IEEE Comp. Graph. Appl.* 31–43 (May 1987).
3. J.-P. Reveilles, *Géométrie discrète, calcul en nombres entiers et algorithmique*, Thèse d'état, Université Louis Pasteur, Strasbourg (1991).
4. P. Nehlig and D. Ghazanfarpour, *Affine texture mapping and antialiasing using integer arithmetic*. Eurographics, Cambridge (1992).
5. P. Nehlig, *Applications affines discrètes et antialassage*, Ph.D. Thesis, Université Louis Pasteur, Strasbourg (1992).
6. M.-A. Jacob, *Applications Quasi-affines*, Ph.D. Thesis, Université Louis Pasteur, Strasbourg (1993).
7. M. P. Mc Ilroy, Best approximate circles on integer grids. *ACM Trans. Graph.* **2**(4), 237–263 (1983).
8. K. C. Posch and W. D. Fellner, The circle-brush algorithm. *ACM Trans. Graph.* **8**(1), 1–24 (1989).
9. Z. Kulpa and B. Kruse, Algorithms for circular propagation in discrete images. *CVGIP* **24**, 305–328 (1983).
10. J. E. Bresenham, Algorithm for computer control of a digital plotter. *IBM Syst. J.* **4**(1), 25–30 (1965).

11. J. E. Bresenham, A linear algorithm for incremental digital display of circular arcs. *Comp. Graph. Image Proc.* **20**(2), 100–106 (1977).
12. J. E. Bresenham, *Algorithm for Circle Arc Generation*, Springer-Verlag, Berlin, 213–213 (1985).
13. S. Pham, Digital circles with non-lattice point centers. *The Visual Comp.* **9**, 1–24 (1992).
14. A. Nakamura and K. Aizawa, Digital circles. *Comp. Vision Graph. Image Proc.* **26**, 242–255 (1984).
15. S. N. Biswas and B. B. Chaudhuri, On the generation of discrete circular objects and their properties. *CVGIP* **32**, 158–170 (1985).
16. M. L. V. Pitteway and R. J. Botting, Integer circles, etc. Three move extension of Bresenham's algorithm. *Comp. Graph. Image Proc.* **3**, 260–261 (1974).
17. E. Andres, *Le cercle discret.*, Rapport interne, Centre de Recherche en Informatique, Université Louis Pasteur, Strasbourg (1989).
18. E. Andres, *Cercles discrets et rotations discrètes*, Ph.D. Thesis, Université Louis Pasteur, Strasbourg (1992).
19. Y. P. Kuzmin, An efficient circle-drawing algorithm. *Comp. Graph. Forum* **9**, 333–336 (1990).
20. X. Wu, J. G. Rokne, Double-step incremental generation of lines and circles. *CVGIP* **37**, 331–344 (1987).
21. S. Y. Hsu, L. R. Chow, and H. C. Liu, A new approach for the generation of circles. *Comp. Graph. Forum* **12**(2), 105–109 (1993).
22. G. Reeb, J. P. Réveillès, A. Troesch, and E. Urlacher, *Equations différentielles en nombres entiers*, rapport interne, IRMA, Université Louis Pasteur, Strasbourg (1987).
23. H. Holin, Harthong-Reeb analysis and digital circles. *The Vis. Comp.* **8**(1), 8–17 (1991).
24. L. Piegl and W. Tiller, A menagerie of rational B-spline circles. *IEEE Comp. Graph. Appl.* 48–56 (September 1989).
25. M. Goldapp, Approximation of circular arcs by cubic polynomials. *Comp. Aided Geometric Des.* **8**, 227–238 (1991).
26. D. Field, Algorithms for drawing anti-aliased circles and ellipses. *CVGIP* **33**, 1–15 (1986).
27. W. E. Wright, Parallelization of Bresenham's line and circle algorithms. *IEEE Comp. Graph. Appl.* 60–67 (September 1990).
28. O. Taussky, Sums of squares. *Am. Math. Monthly*, 805–830 (October 1970)
29. G. Hardy and Wright, *Introduction to the theory of numbers*, Oxford Science Publications (1989).
30. E. Andres, *Le plan discret.* Rapport interne, Centre de Recherche en Informatique, Université Louis Pasteur, Strasbourg (1994).
31. A. Kaufman and E. Shimony, 3-*D scan conversion algorithms for voxel based graphics*, Chapel Hill, New York, 45–76 (1986).
32. A. Kaufman, Efficient algorithms for 3-D scan conversion of parametric curves, surfaces, volumes. *Comp. Graph.* **21**(4), 171–179 (1987).
33. B. Vidal, *Vers un lancer de rayon discret*, Ph.D. Thesis, Université des Sciences et Techniques de Lille, Lille (1992).